

系統整合設計期中報告

指導教授：嚴家銘 教授

課程名稱：系統整合設計

班級：四設四甲

學生：劉肇烜


學號：49623149

二〇一一年四月二十三日

目錄

作者簡介	3
前言	4
內容章節	5
心得	17
參考資料	17

作者簡介

個人基本資料		
姓名	劉肇烜	
E-mail	zxd780815@hotmail.com	
學歷	國立秀水高級工業職業學校機械科	
興趣	跑步、閱讀	
專業證照	丙級證照—電腦輔助機械製圖	
熟悉軟體	Microsoft Office、Solid Work、Auto CAD	

前言

系統整合設計係指在特定目的之驅使下，以整體考量作為前提，令範圍內之多數事物，得以依一定秩序，相互溝通連結，並渾然自成一體之方法。

系統整合設計之目的：

1. 納入跨領域或較大範圍的設計考量，期許達成廣域的設計最佳化
2. 完成能因應局部環境快速變化的永續設計與能高度重用之系統化設計

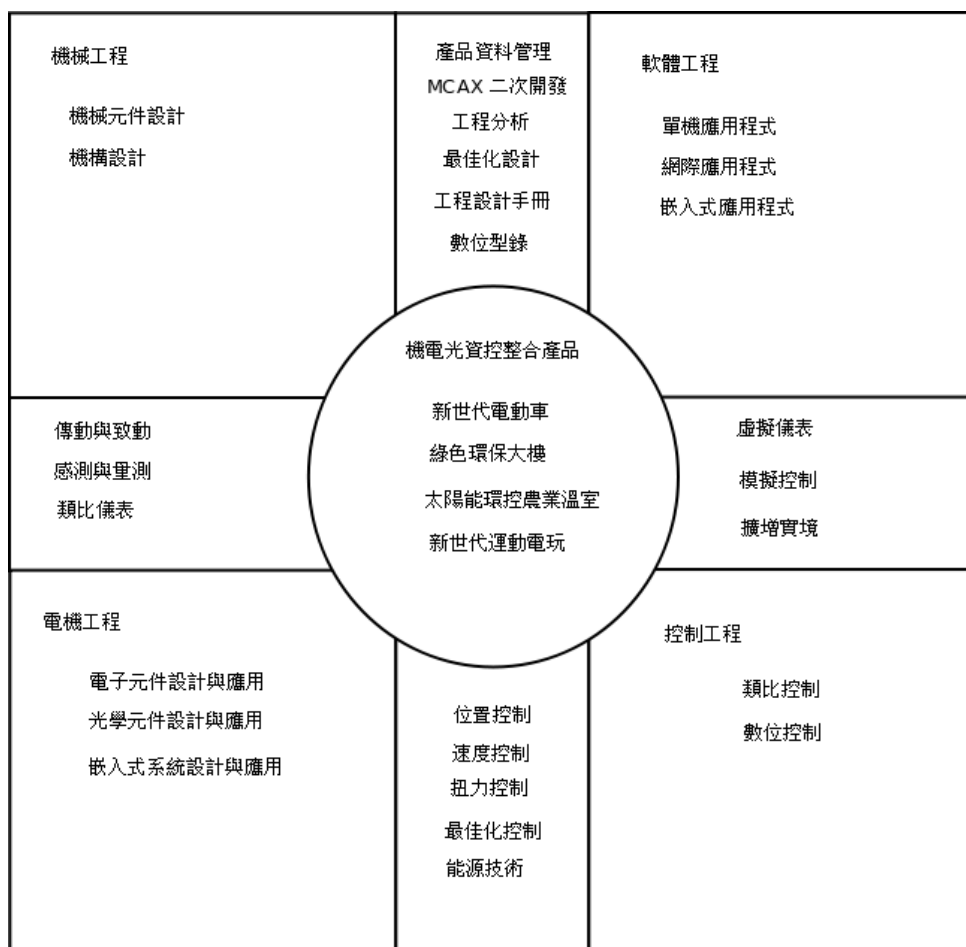


圖 1

內容章節

根據下列的動態系統 (所有質量與對應係數皆設為 1):

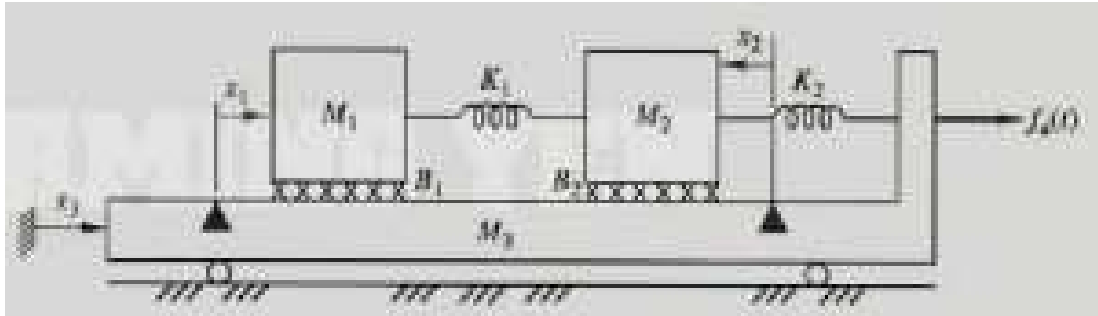


圖 2

若 $f_d(0)=0$, 在 $x_2=-1, x_1=x_3=0$ 的起始條件下，模擬系統的動態運動結果。

1. 以 Scicos 類比計算器解 Rossler attractor 微分方程式, $x(0)$
 $= 1$, 類比元件配置:

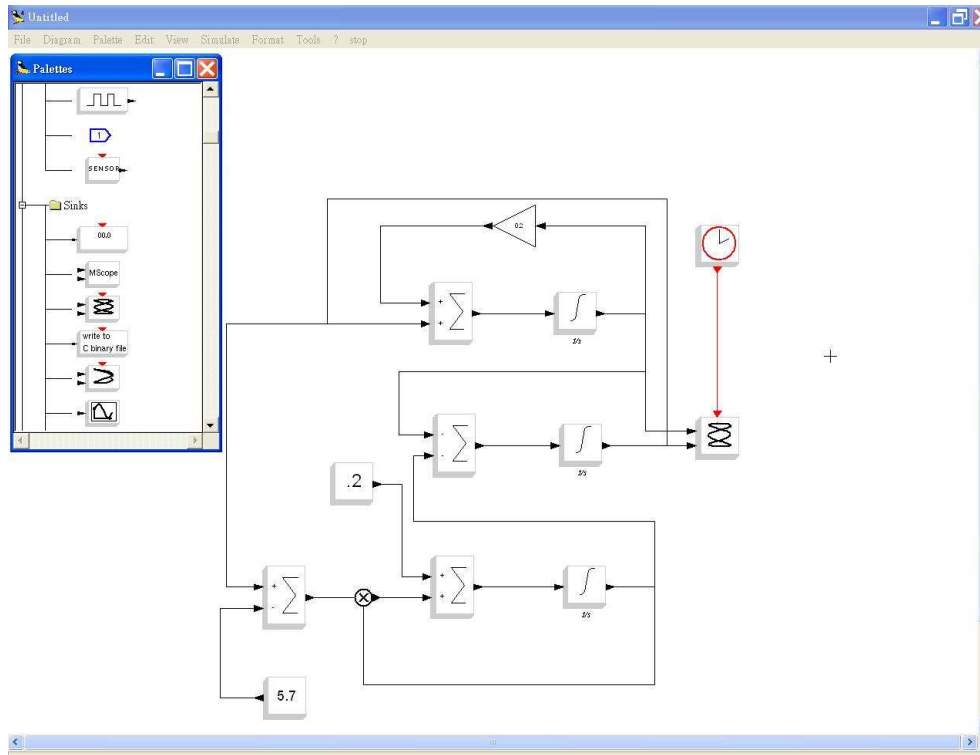


圖 3

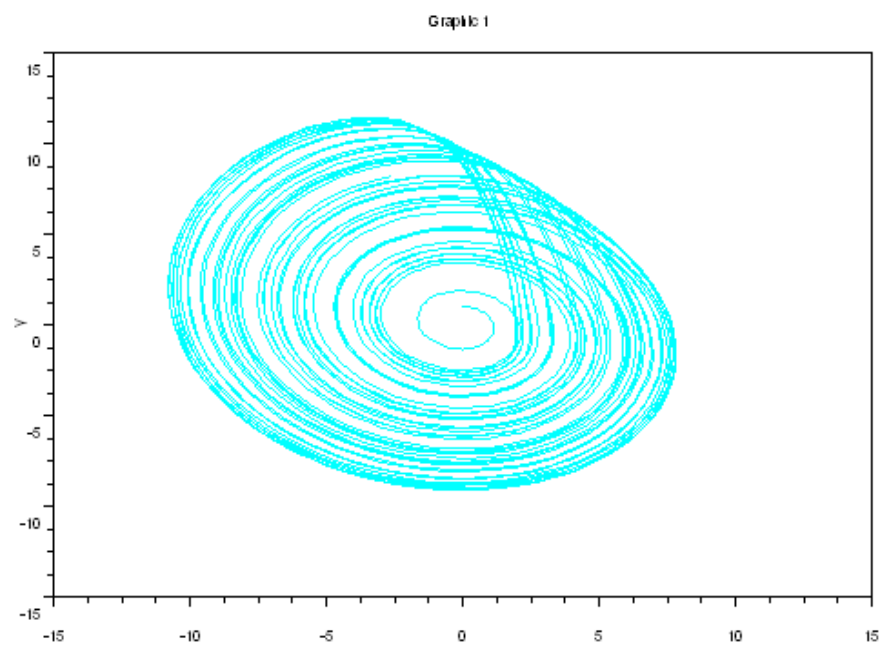


圖 4

2. 以 C Runge-Kutta 程式解譯系統

```
2.c - Sci
File Edit Search View Tools Options Language Buffers Help
11.c 22.c
1  /* Runge Kutta for a set of first order differential equations */
2
3  #include <stdio.h>
4  #include <math.h>
5
6  #define N 3 /* number of first order equations */
7  #define dist 0.1 /* stepsize in t */
8  #define MAX 300.0 /* max for t */
9
10 FILE *output; /* internal filename */
11 // 利用 pipe 呼叫 gnuplot 繪圖
12 FILE *pipe;
13
14 void runge4(double x, double y[], double step); /* Runge-Kutta function */
15 double f(double x, double y[], int i); /* function for derivatives */
16
17 -void main() {
18
19     double t, y[N];
20     int j;
21
22     output=fopen("osc.dat", "w"); /* external filename */
23
24     y[0]=1.0; /* initial position */
25     y[1]=0.0; /* initial velocity */
26     y[2]=0.0;
27
28     //fprintf(output, "%0\t%f\n", y[0]);
29
30     for (j=1; j*dist<=MAX; j++) /* time loop */ {
31
32         t=j*dist;
33         runge4(t, y, dist);
34         fprintf(output, "%f\t%f\n", y[1], y[0]);
35     }
36
37     fclose(output);
38 }
```

lin91 co=2 INS (CR+LP)

圖 5

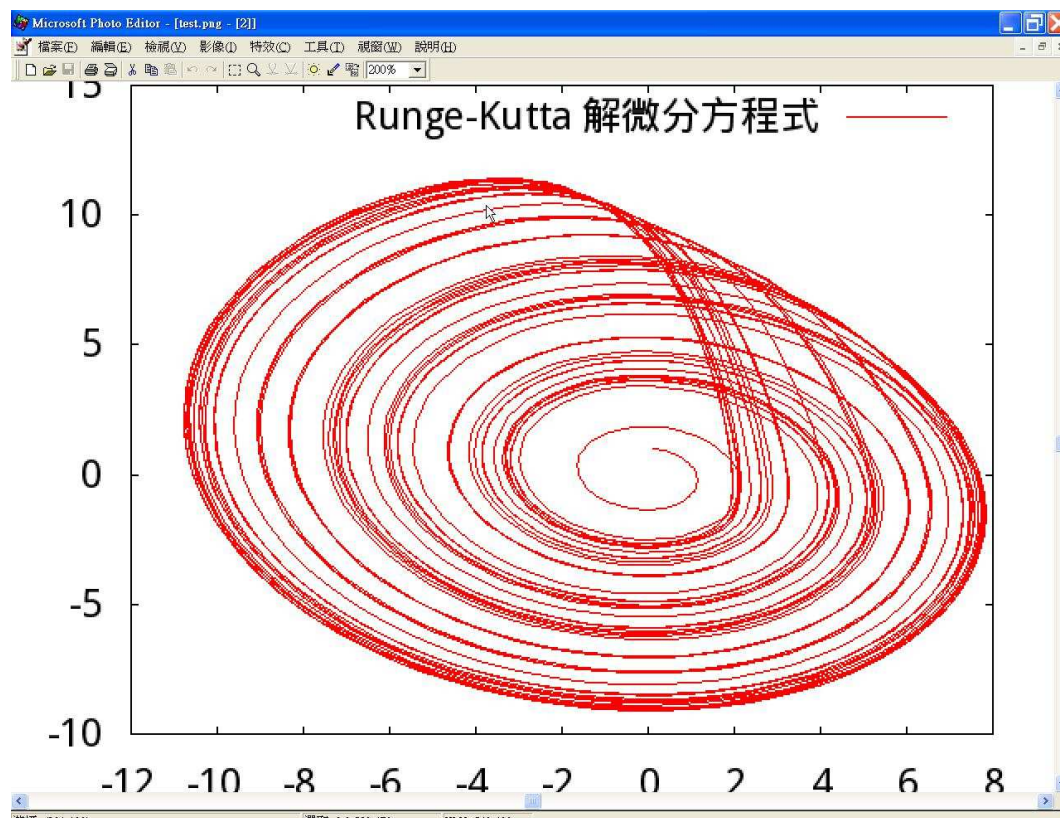


圖 6

3. 利用鍵結圖進行模擬：

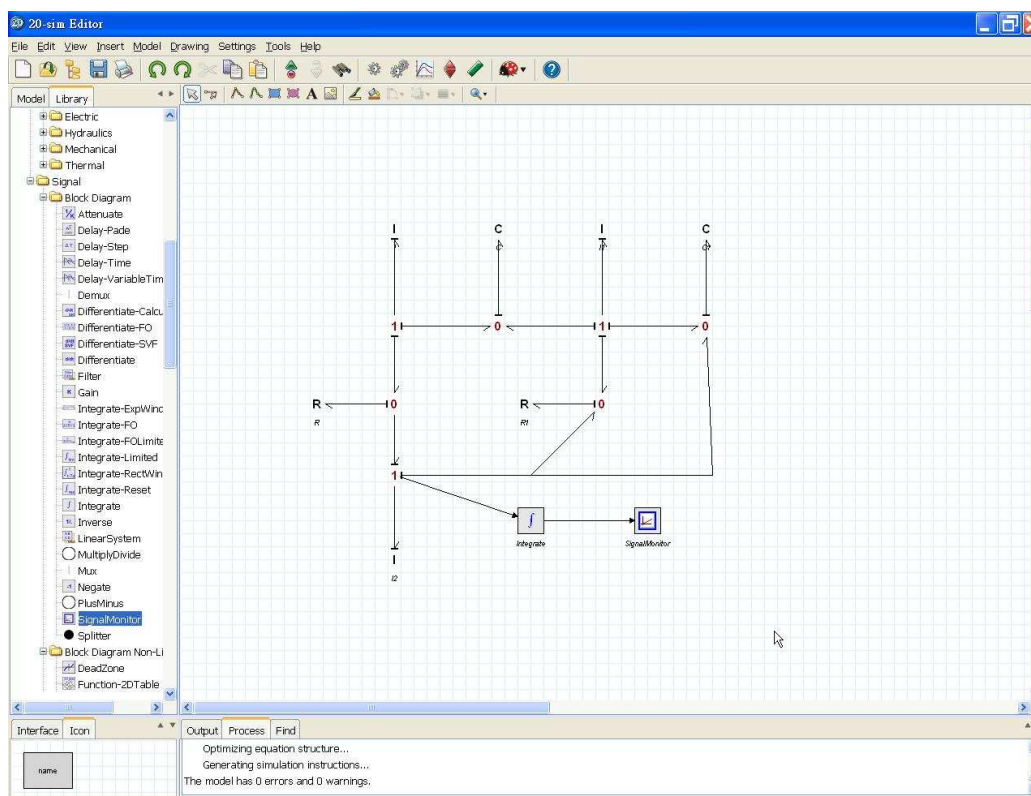


圖 7

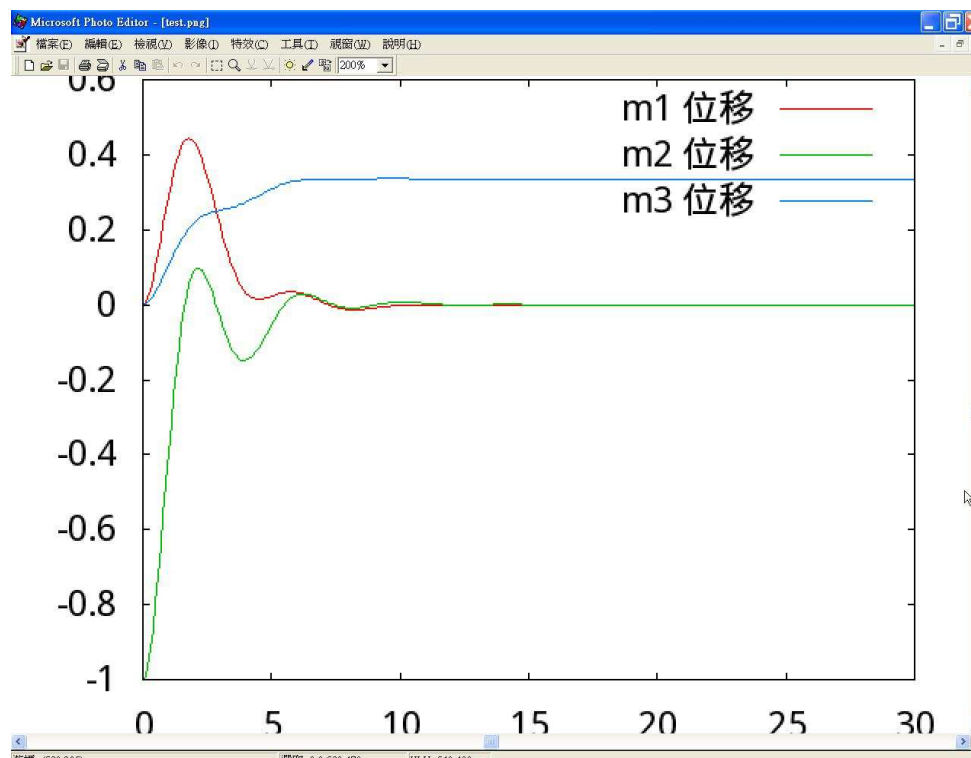


圖 8

課堂練習

題目一

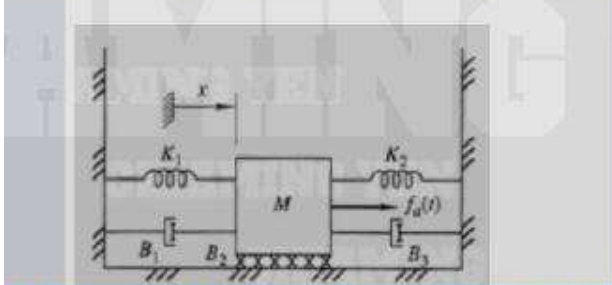


圖 9

以 scicoslab 模擬

```
1 function dx = f(t,x)
2 b1=1;b2=1;b3=1;k1=1;k2=1;m=1;
3 dx(1)=x(2);
4 dx(2)=-(b1/m)*x(2)-x(1)*((1/k1)/m)-(b2/m)*x(2)-x(1)*((1/k2)/m)-(b3/m)*x(2);
5 endfunction
6
7 t0 = 0
8 x0 = [-1;0]
9 t = 0:0.1:30;
10 x = ode(x0, t0, t, f);
11 plot2d(t',x(1,:),leg="x(1) ")
12
13
```

圖 10

模擬結果

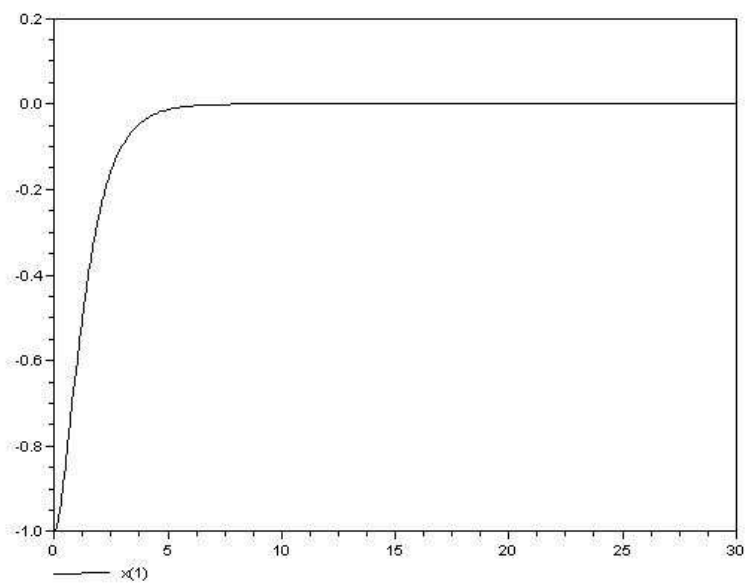


圖 11

以 scicos 模擬

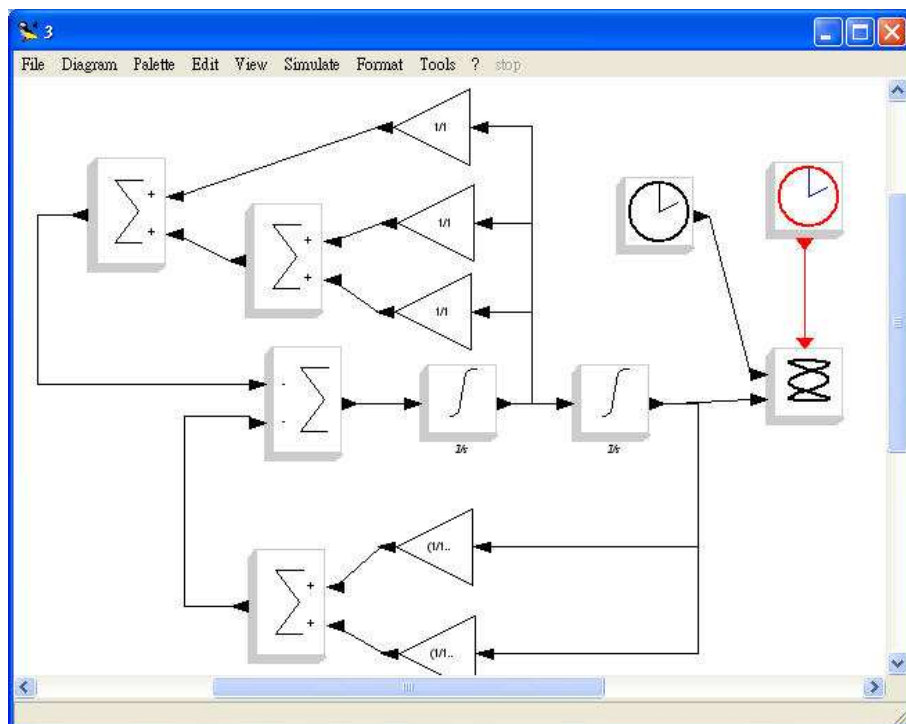


圖 12

模擬結果

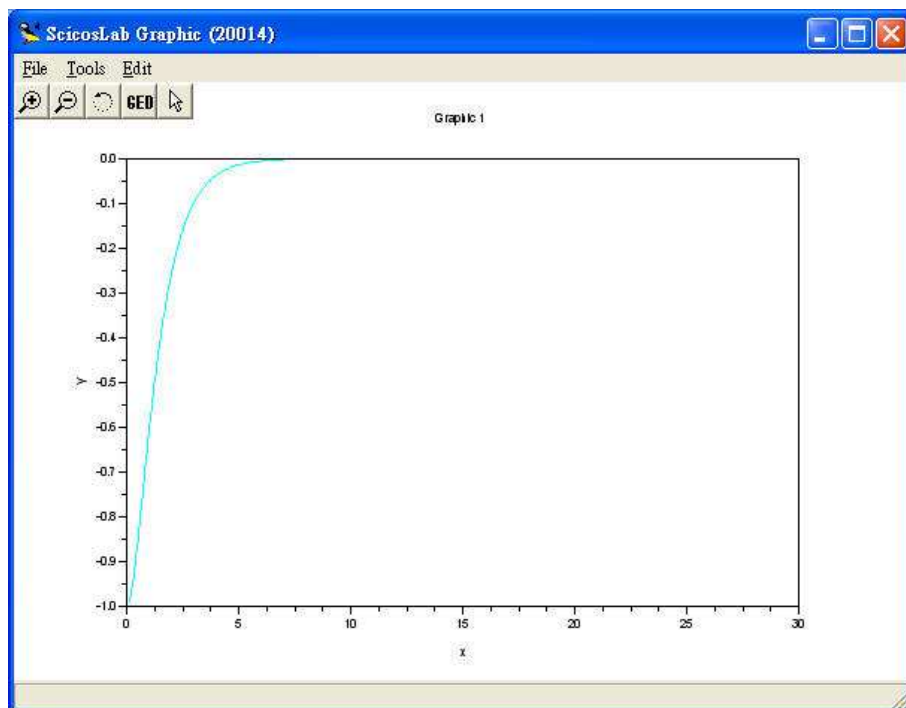


圖 13

以 C Runge-Kutta 模擬

```
81     }
82 }
83
84 - double f(double x, double y[], int i){
85     double b1=1.0, b2=1.0, b3=1.0, k1=1.0, k2=1.0, m=1.0;
86     if (i==0)
87         x=y[1];
88
89     if (i==1)
90         x=-(b1/m)*y[1]-y[0]*(k1/m)-(b2/m)*y[1]-y[0]*(k2/m)-(b3/m)*y[1];
91
92     return x;
93 }
```

```
>Exit code: 0
>tcc dynamic3system.c
>Exit code: 0
>tcc -run dynamic3system.c
gnuplot> set term png enhanced font "v:/wqy-microhei.ttc" 18
Terminal type set to 'png'
Options are 'nocrop enhanced font v:/wqy-microhei.ttc 18 size 640,480 '
gnuplot>
gnuplot> set output "test.png"
gnuplot>
gnuplot> plot "osc.dat" title "m1 位移" with lines
gnuplot>
gnuplot> quit
>Exit code: 0
```

圖 14

模擬結果

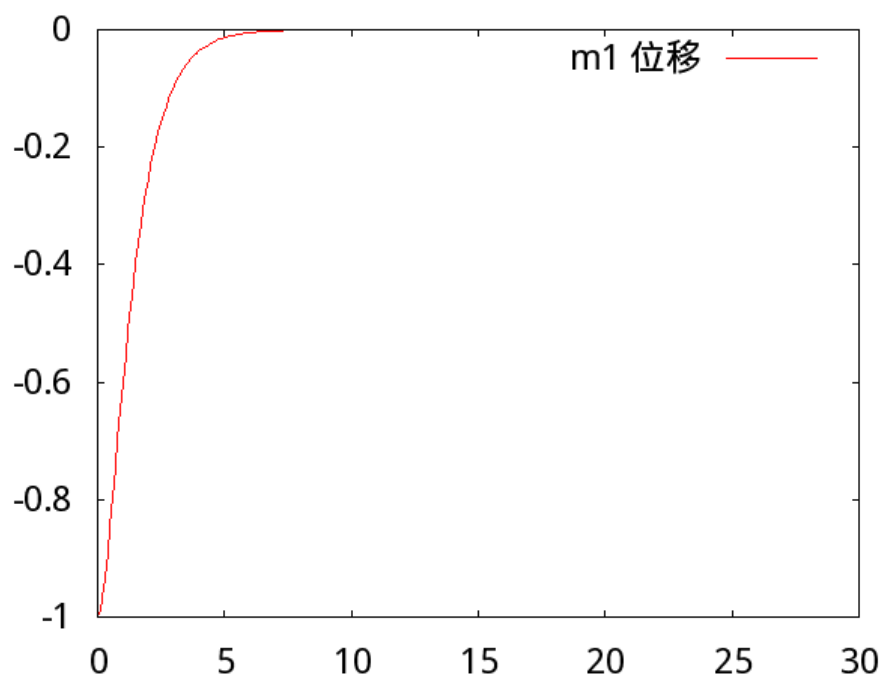


圖 15

以 20-sim 模擬

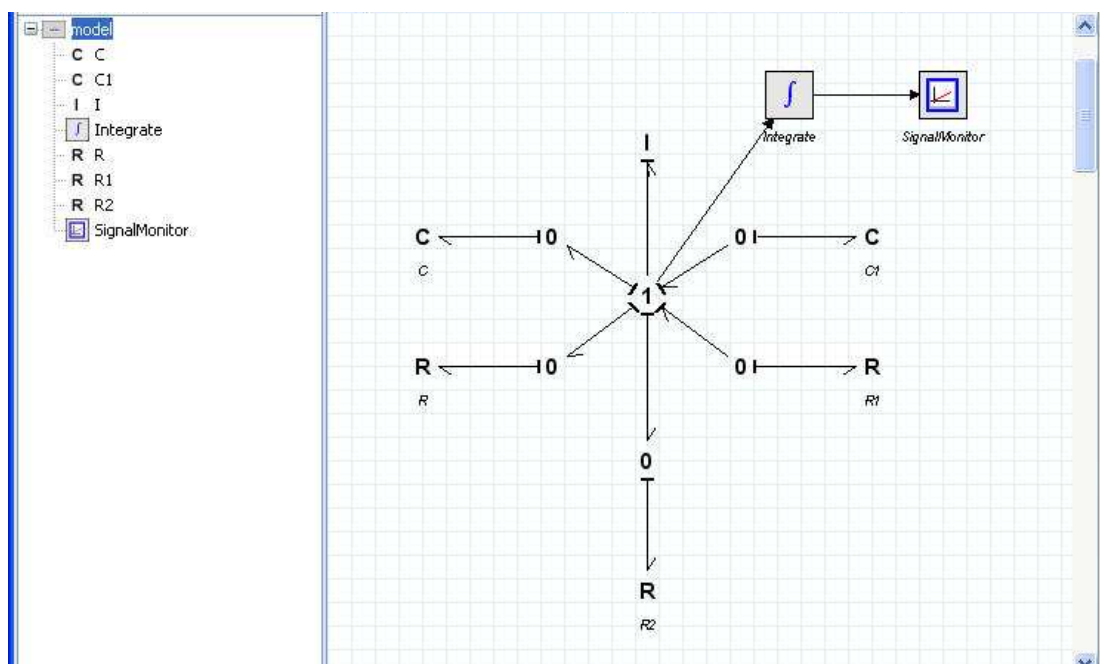


圖 16

模擬結果

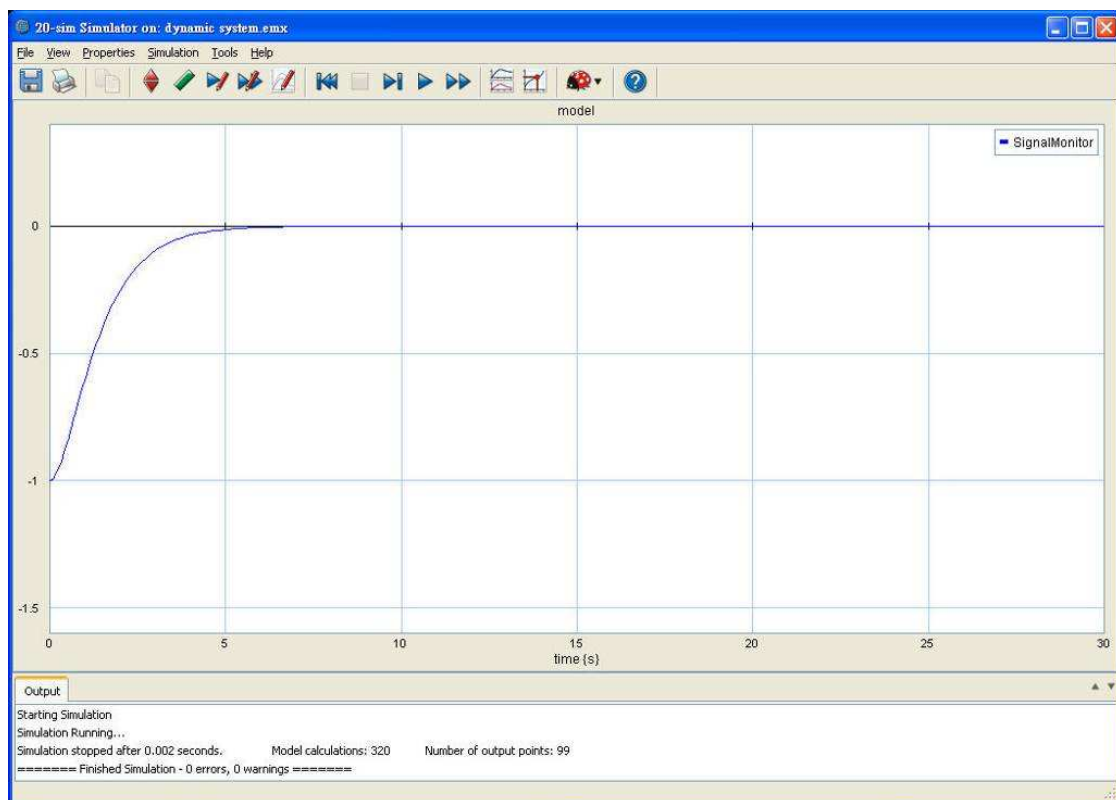


圖 17

題目二

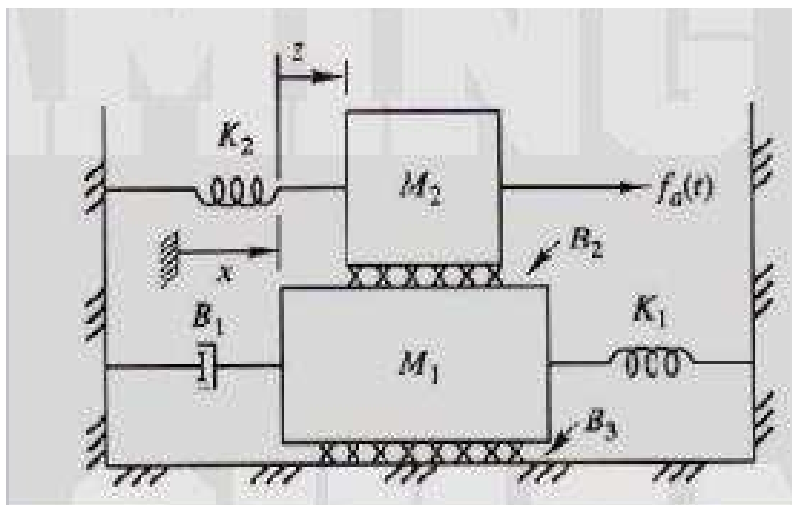


圖 18

以 20-sim 模擬

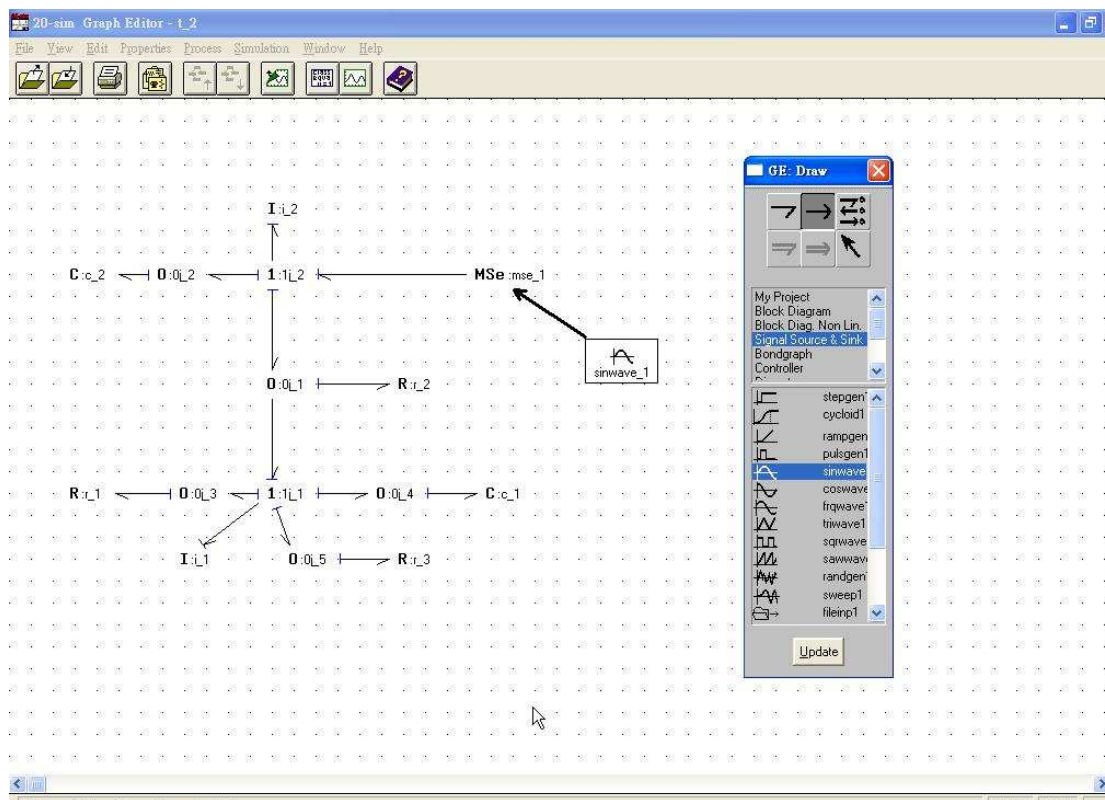


圖 19

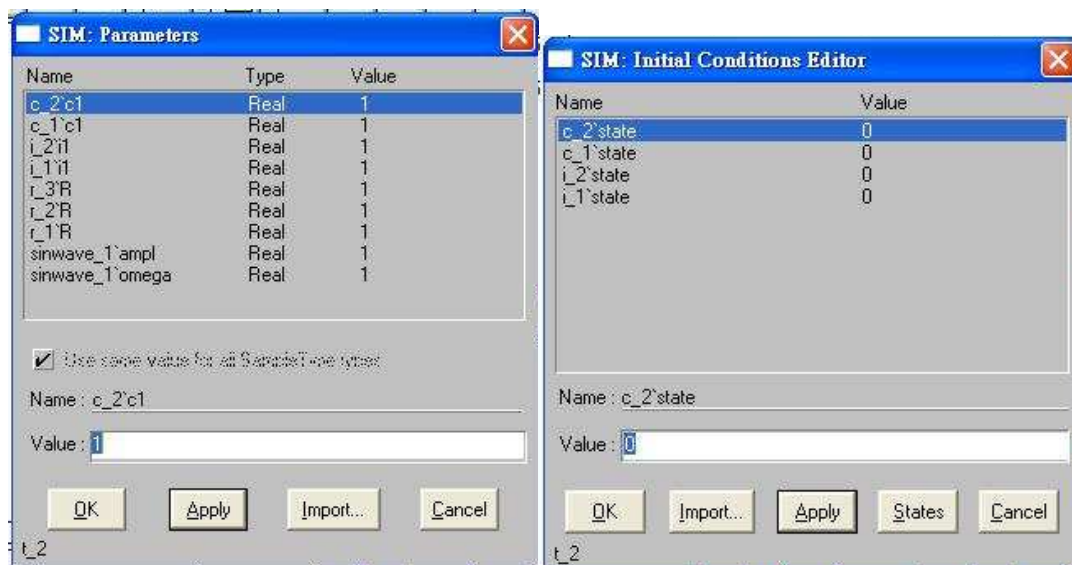


圖 20

模擬結果

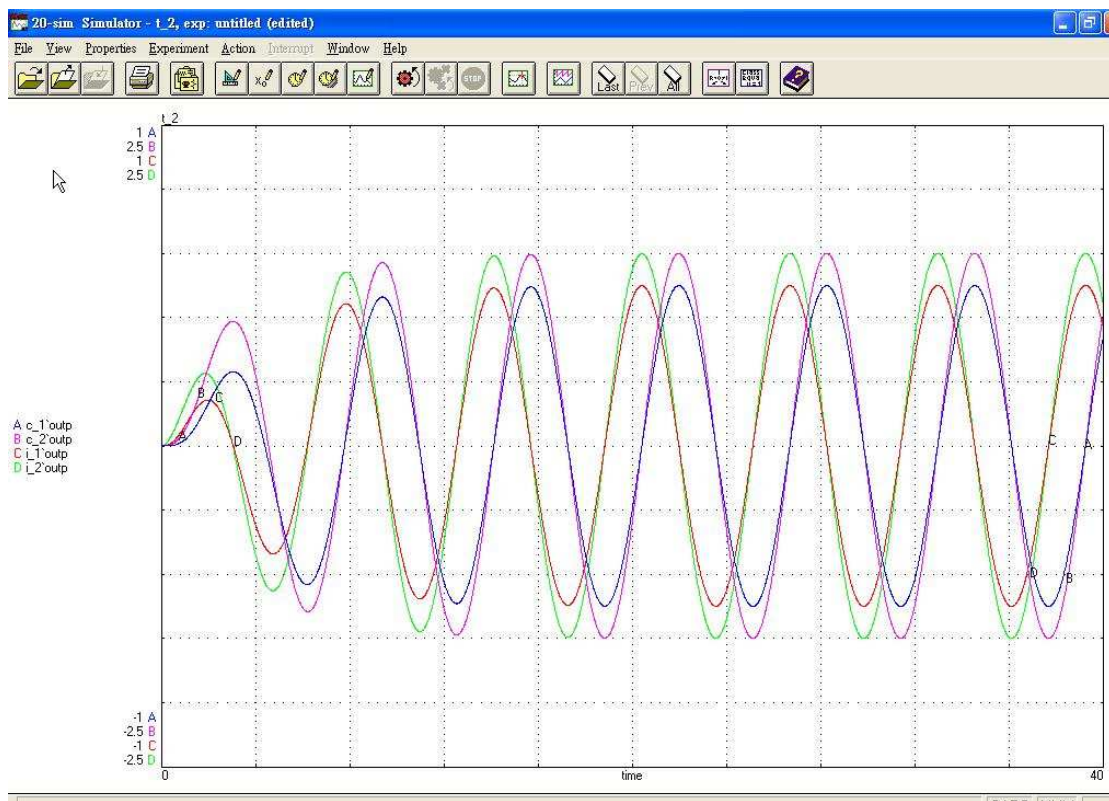


圖 21

題目三

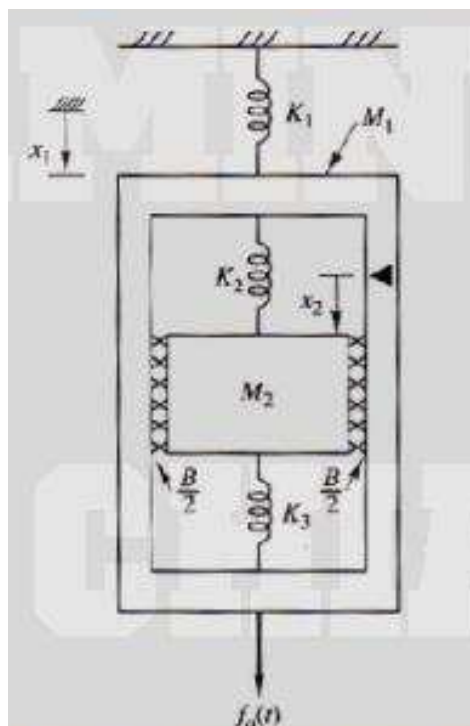


圖 22

以 20-sim 模擬

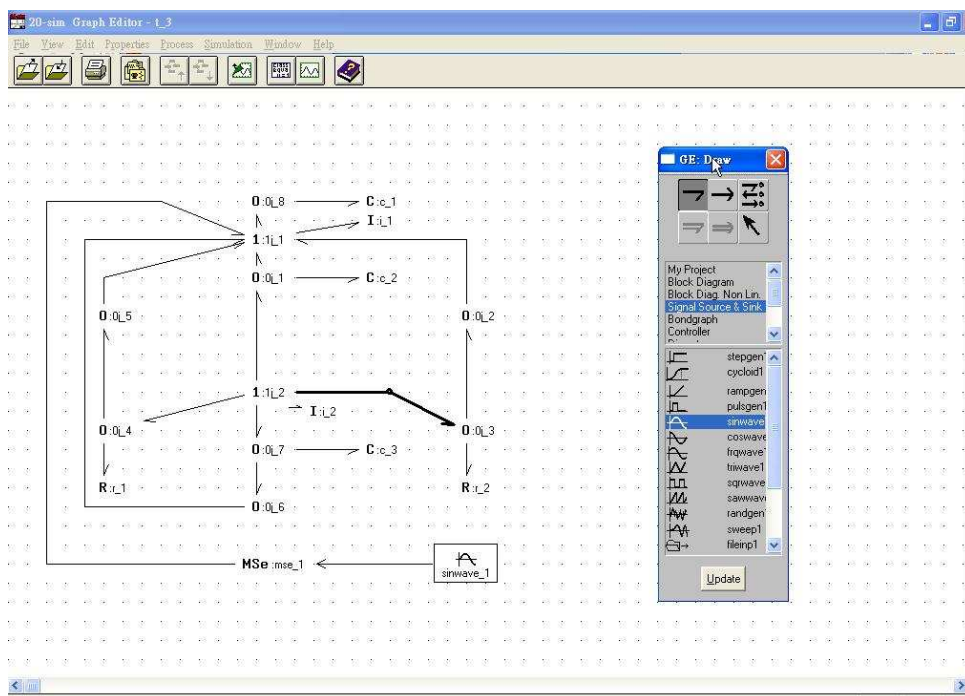


圖 23

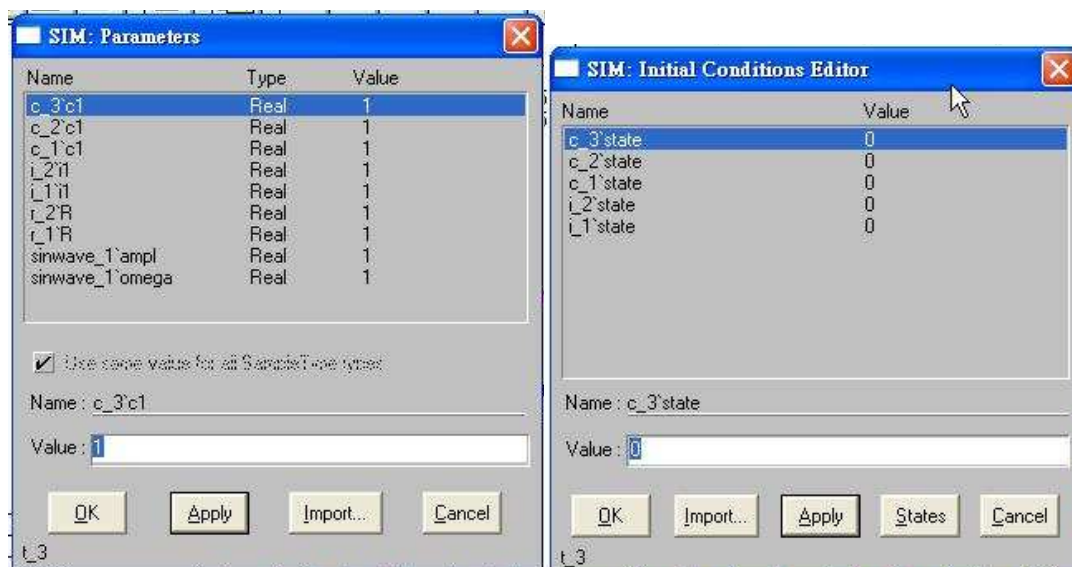


圖 24

模擬結果

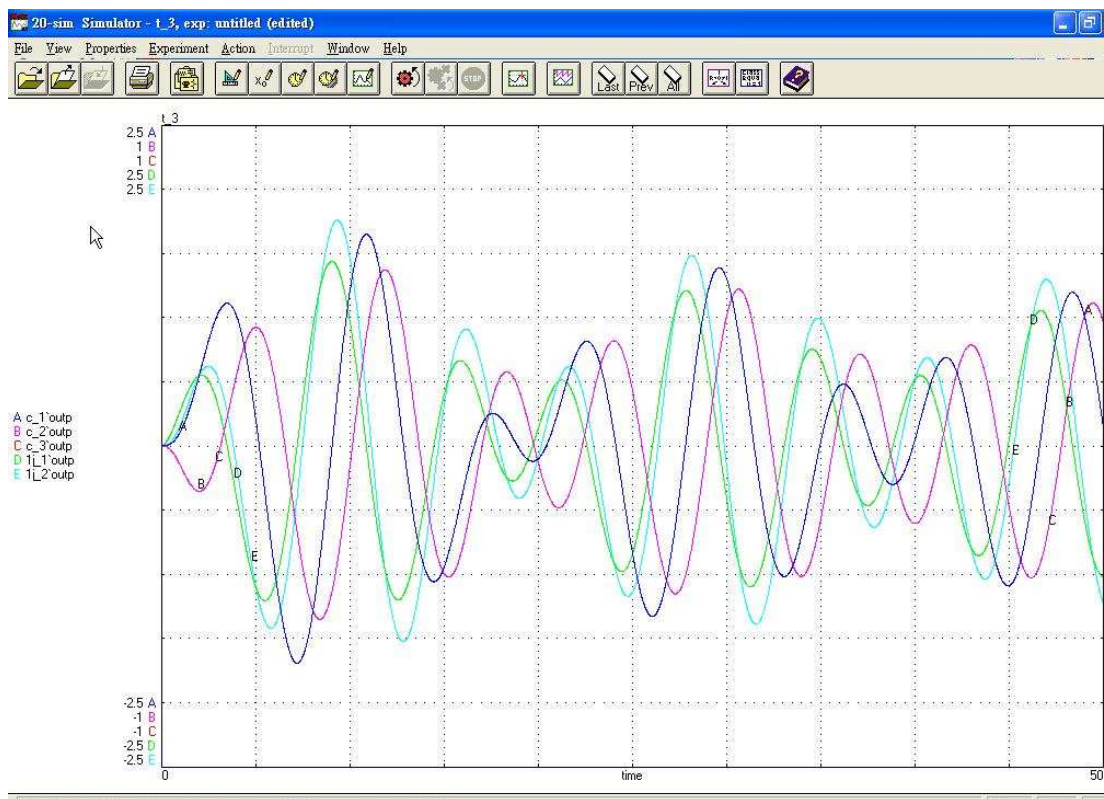


圖 25

心得

課堂練習的部分是採用鏈結圖模擬法，而使用 20-sim 必須考慮到共流結(1)，共勢結(0)與儲位能元件(C)，儲動能元件(I)，阻尼元件(R)之間的能量流方向，還有設定的初始條件，才能使圖形與其他測量方法的結果相同，對於程式的部分須完成微分方程式等步驟，因此必須演算公式，使 scicoslab、C Runge-Kutta 的模擬結果與 20-sim 鏈結圖完全相同。

實作的過程中，以程式編譯的難度最大，因對於推導方程式需耗費許多時間，但這卻是最佳的驗證方法之一，故須熟練程式編譯能力，這是目前學習的重點。

參考資料

網路連結：blog.kmol.info