

系統整合設計

期中報告

指導教授：嚴家銘

班級：四設四甲

學號：49623127


姓名：胡育霖

完成日期：2011/4/18

目錄

目錄.....	1
作者簡介.....	2
前言.....	3
內容.....	4
心得與討論.....	15
參考資料.....	15

作者簡介

個人基本資料		
姓名	胡育霖	
班級	四設四甲	
學號	49623127	
學歷	國立虎尾科技大學-機械設計系(在學) 國立彰化師大附工-機械木模科(已畢)	
證照	丙級木模、丙級電腦輔助立體製圖	
<p>學習歷程：高中時，學習 SolidWorks、AutoCAD，懂得基本操作及繪製一般工程圖及立體圖。</p> <p>大一時，複習 SolidWorks、AutoCAD，另學習 Pro/E、Java</p> <p>大二時，學習 Solid Edge</p> <p>大三時，學習 Inventor、Ansys、COSMOSWorks、Processing、Arduino</p> <p>大四時，學習 RecunDyn、Excel、以及本課程所使用的 20-sim、Scicos</p>		

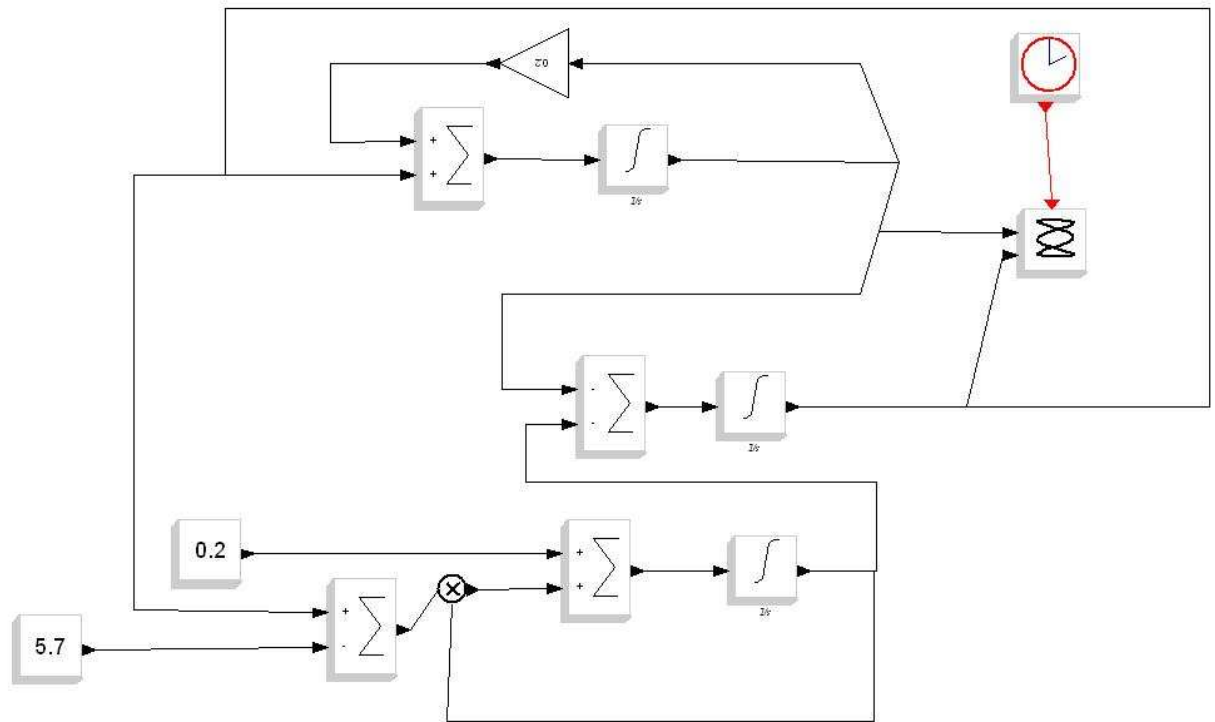
前言

系統整合設計，在建立不同系統及設備間的資料交流管道，積極的意義可縮短作業時間、減少人力資源的浪費、維持資料一致性、增加系統效率。

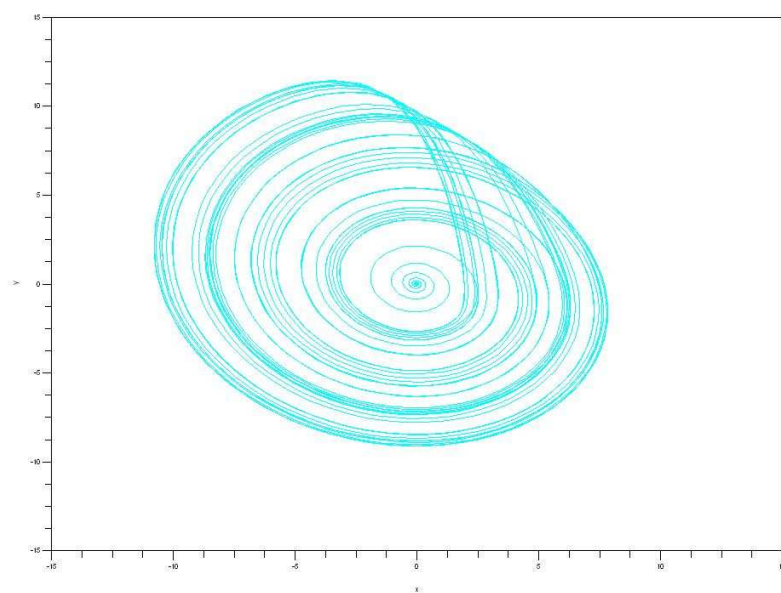
本文藉由機械動力系統利用各種的解題法求解，以達到系統整合的練習。

題目一

使用 Scicos 類比計算器解 Rossler attractor 微分方程式

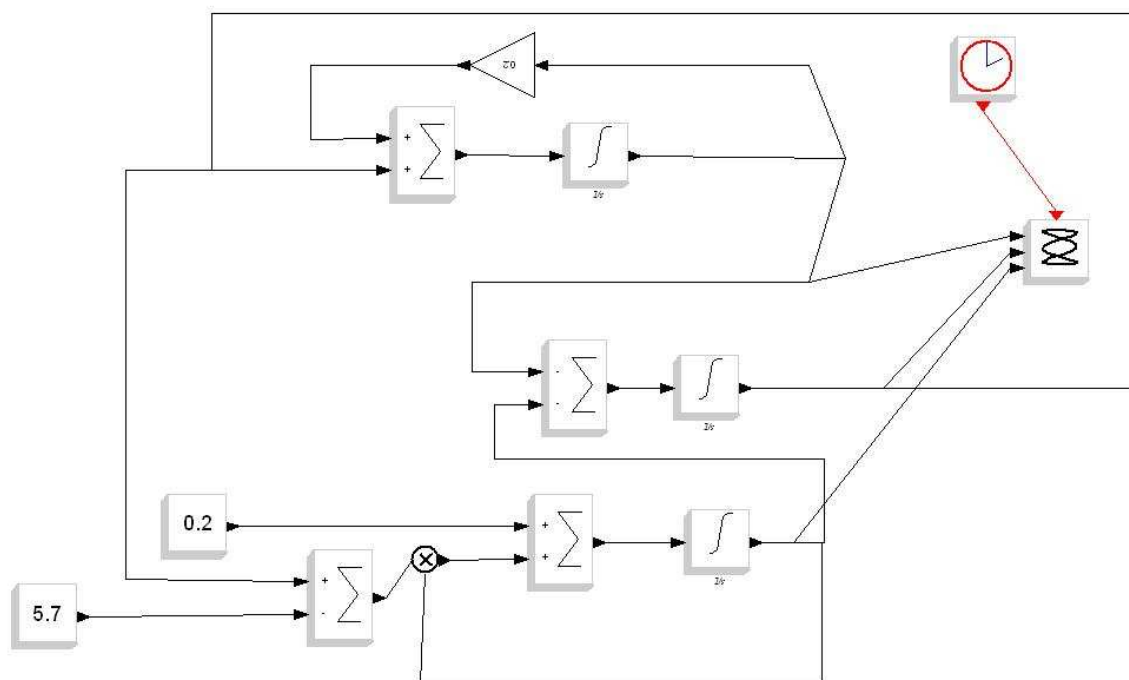


圖一 類比元件配置

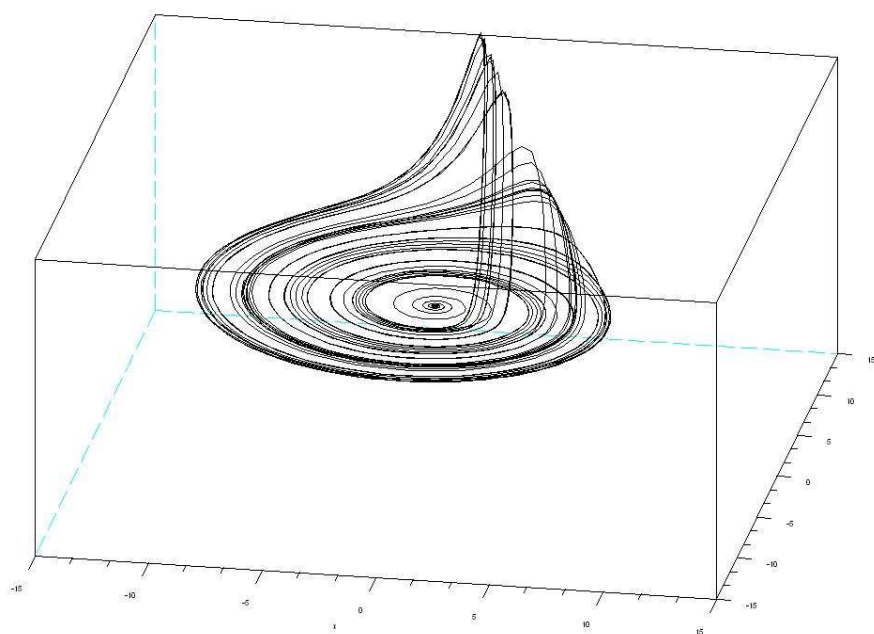


圖二 模擬結果

繪製 3D 模擬結果



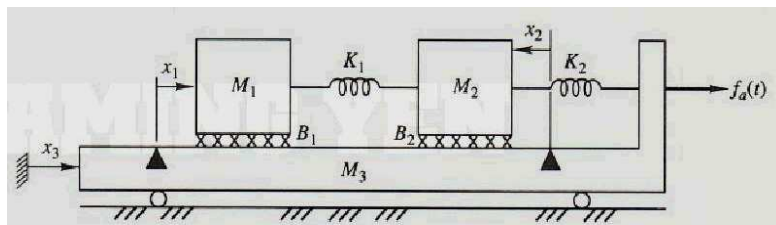
圖三 類比元件配置



圖四 3D 模擬結果

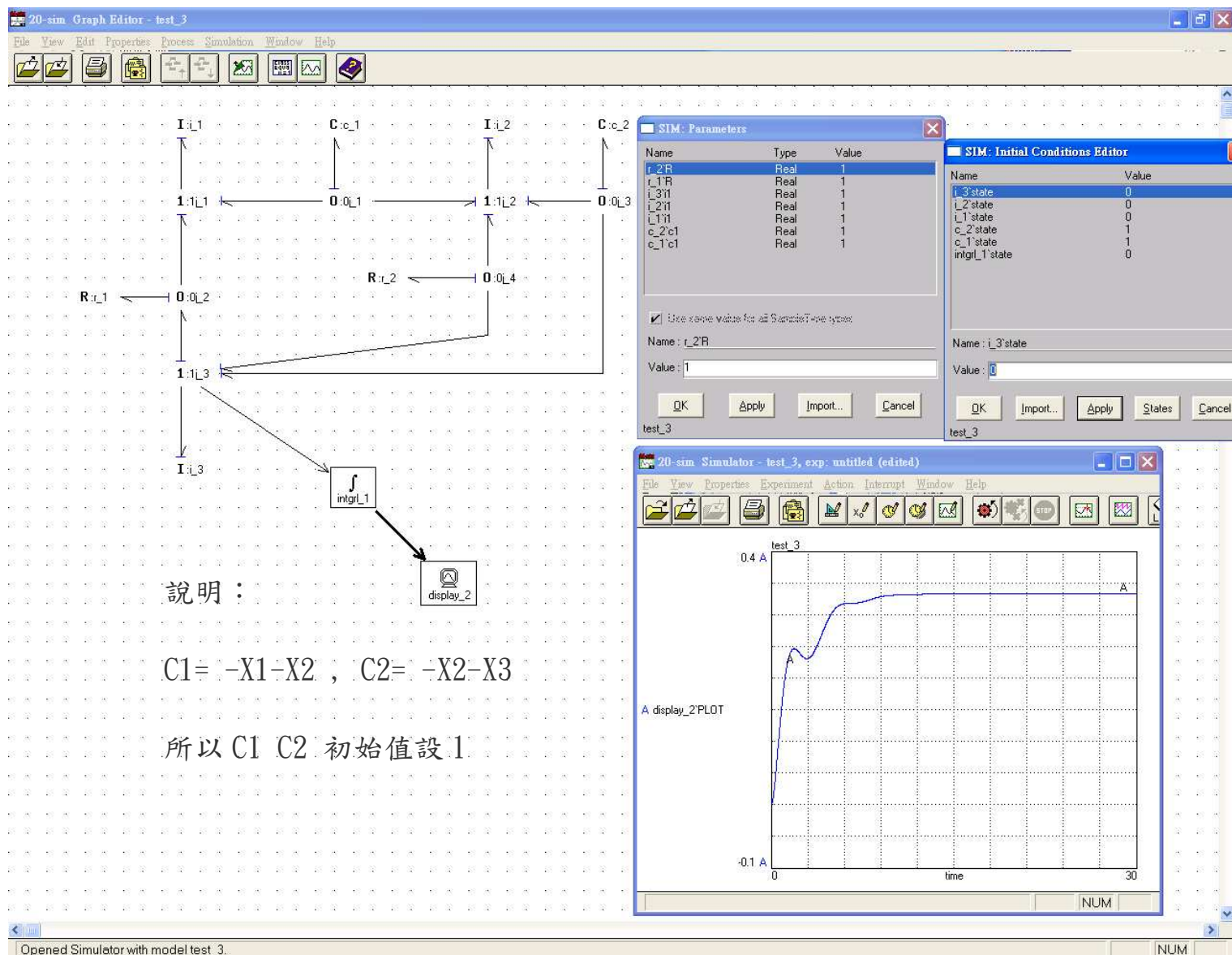
題目二

利用 20-sim 鍵結圖模擬分析



摘自 <http://blog.kmoli.info>

圖五 題目二

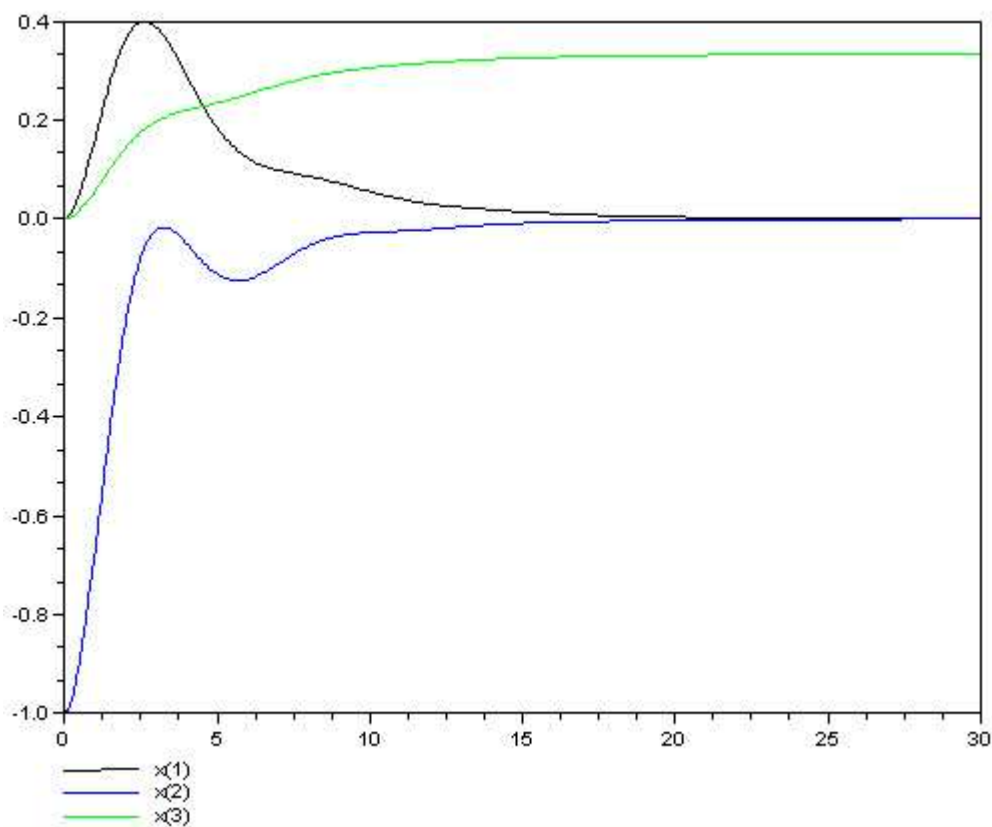


圖六 模擬結果

利用 ScicosLab 模擬分析

```
1 function dx = f(t,x)
2 b1=1;k1=1;k2=2;m1=1;m2=1;
3 dx(1)=x(3);
4 dx(2)=x(4);
5 dx(3)=-(k1/m1)*x(1)-(x(1)+x(2))*(k2/(m1+m2))+x(3)*(b1/(m1+m2));
6 dx(4)=-(b1/(m2+m1))*x(4)-(x(1)+x(2))*(k2/(m1+m2))+sin(t);
7 endfunction
8
9
10 t0 = 0
11 x0 = [0;0;0;0]
12 t = 0:0.1:30;
13 x = ode(x0, t0, t, f);
14 plot2d(t',[x(2,:)'],leg="x(1)")
```

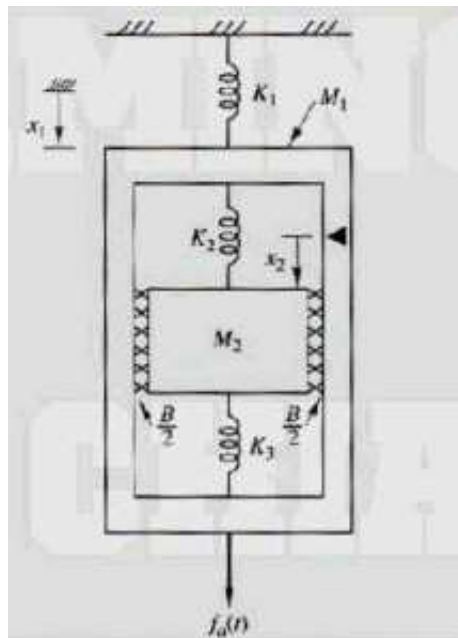
圖七 (題目二_ScicosLab)



圖八 模擬結果(題目二_ScicosLab)

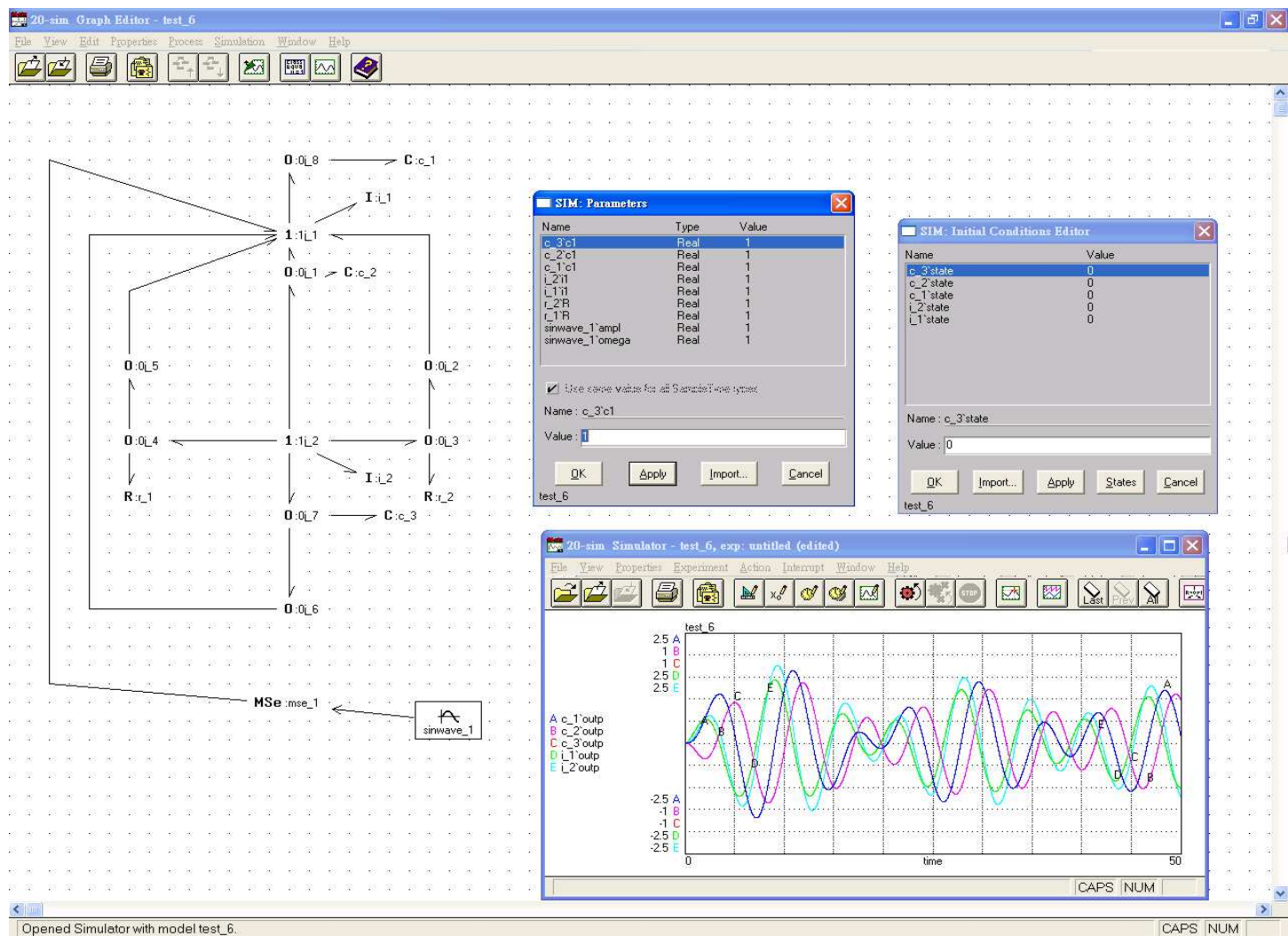
題目三

利用 20-sim 鍵結圖模擬分析



摘自<http://blog.kmol.info>

圖九 題目三

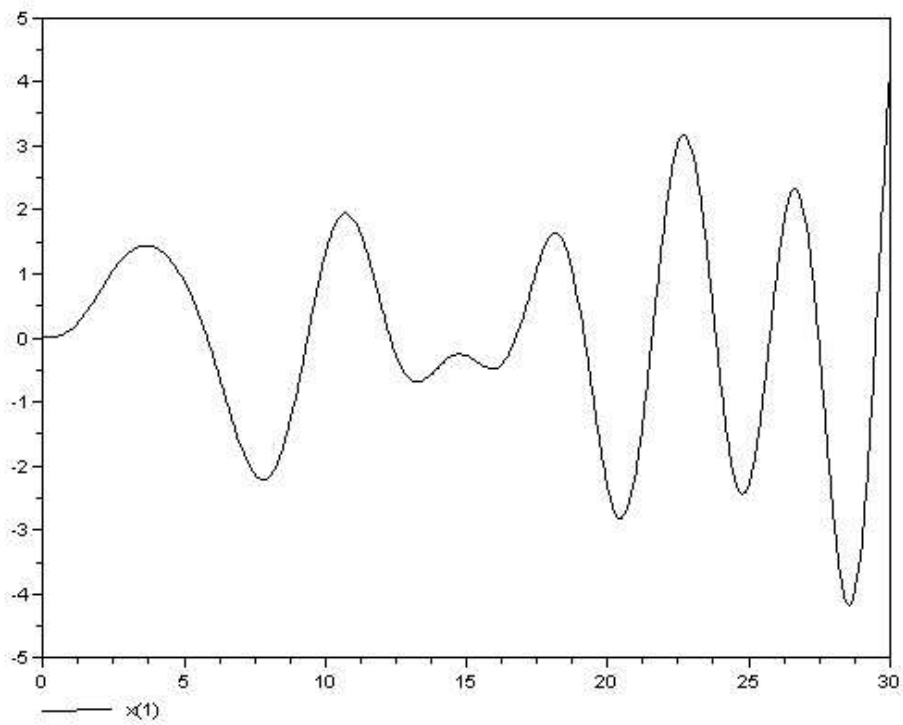


圖十模擬結果(題目三_20sim)

利用 ScicosLab 模擬分析

```
1 function dx = f(t,x)
2 b1=1;k1=1;k2=2;m1=1;m2=1;
3 dx(1)=x(3);
4 dx(2)=x(4);
5 dx(3)=-(k1/m1)*x(1)-(x(1)+x(2))*(k2/(m1+m2))+x(3)*(b1/(m1+m2));
6 dx(4)=-(b1/(m2+m1))*x(4)-(x(1)+x(2))*(k2/(m1+m2))+sin(t);
7 endfunction
8
9
10 t0 = 0
11 x0 = [0;0;0;0]
12 t = 0:0.1:30;
13 x = ode(x0, t0, t, f);
14 plot2d(t', [x(2,:)'], leg="x(1)")
```

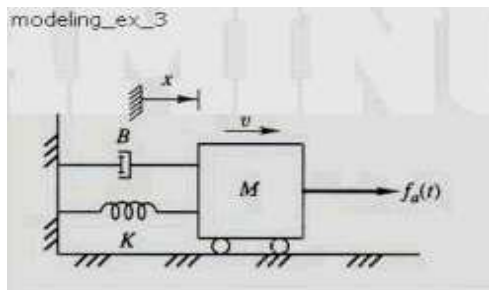
圖十一 (題目三_ScicosLab)



圖十二 模擬結果(題目三_ScicosLab)

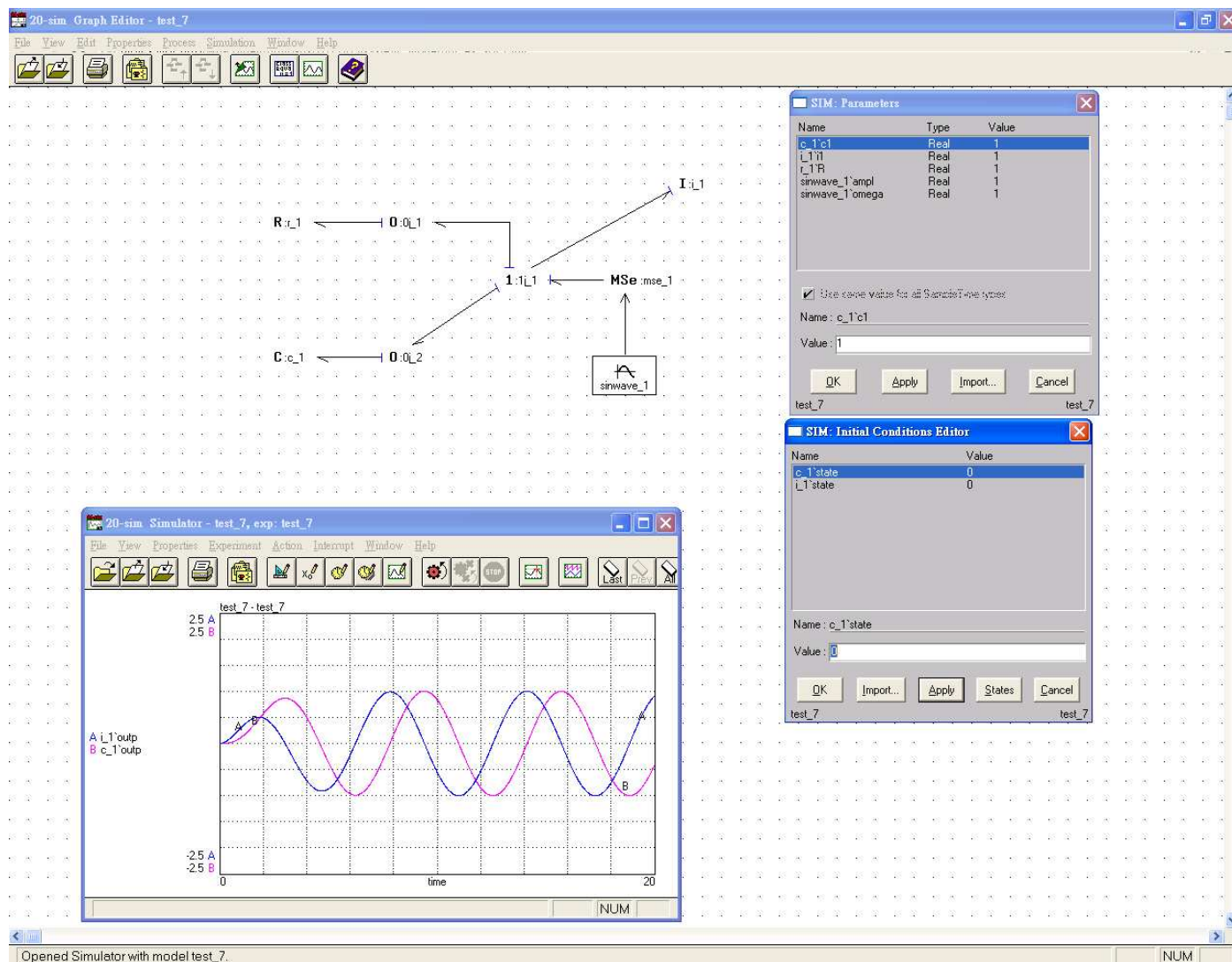
題目四

利用 20-sim 鍵結圖模擬分析



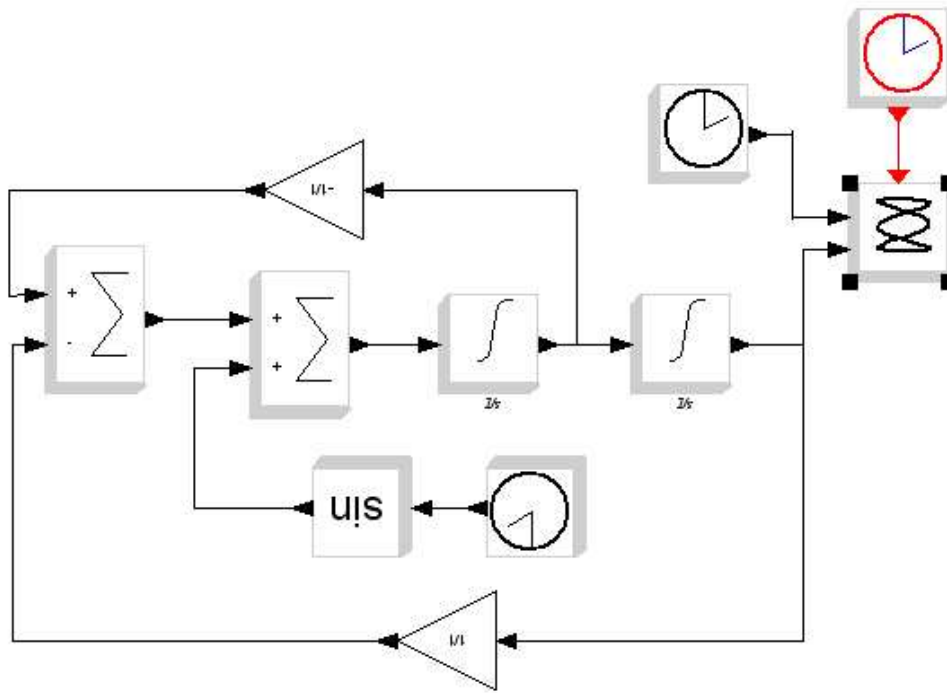
摘自<http://blog.kmol.info>

圖十三 題目四

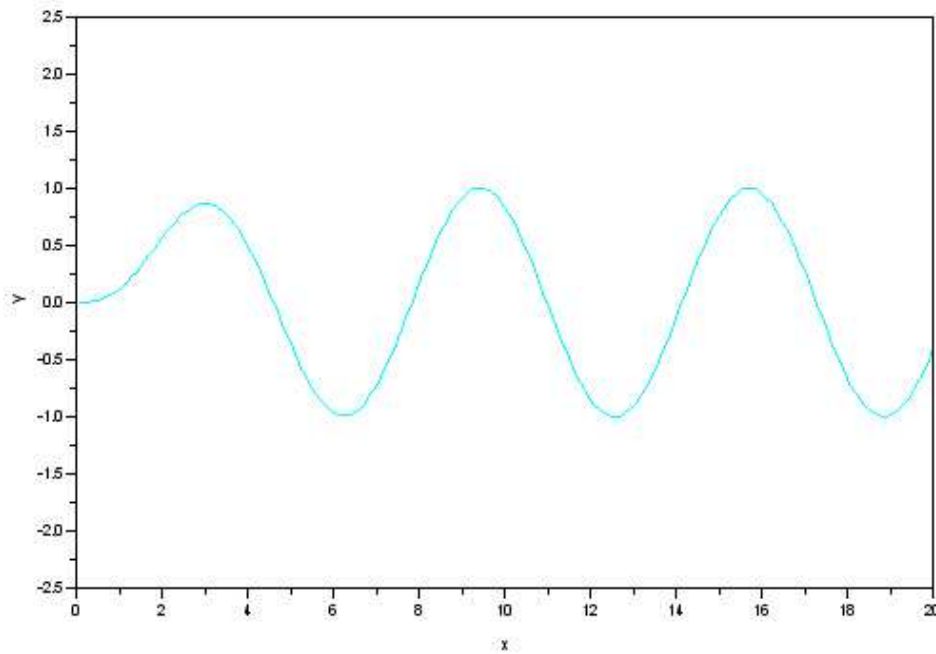


圖十四 模擬結果(題目四_20-sim)

利用 Scicos 模擬分析



圖十五 (題目四_Scicos)



圖十六 模擬結果(題目四_Scicos)

利用 C Runge-Kutta 模擬分析

```
void main(){

    double t, y[N];
    int j;

    output=fopen("osc.dat", "w"); /* external filename */

    y[0]=0.0; /* initial position */
    y[1]=0.0; /* initial velocity */

    fprintf(output, "0\t%f\n", y[0]);

    for (j=1; j*dist<=MAX ;j++) /* time loop */{

        t=j*dist;
        runge4(t, y, dist);
        fprintf(output, "%f\t%f\n", t, y[0]);
    }

    fclose(output);

    pipe = popen("gnuplot -persist","w");
    //fprintf(pipe,"set term png enhanced font \"v:/fireflysung.ttf\" 18\n");
    fprintf(pipe,"set term png enhanced font \"v:/wgy-microhei.ttc\" 18\n");
    //fprintf(pipe,"set yrange [68:70]\n");
    fprintf(pipe,"set output \"test.png\"\n");
    fprintf(pipe, "plot \"osc.dat\" title \"m1 位移\" with lines\n");
    fprintf(pipe,"quit\n");
    pclose(pipe);
}

void runge4(double x, double y[], double step){

    double h=step/2.0, /* the midpoint */
    t1[N], t2[N], t3[N], /* temporary storage arrays */
    k1[N], k2[N], k3[N], k4[N]; /* for Runge-Kutta */
    int i;

    for (i=0;i<N;i++){

        t1[i]=y[i]+0.5*(k1[i]=step*f(x,y,i));
    }

    for (i=0;i<N;i++){

        t2[i]=y[i]+0.5*(k2[i]=step*f(x+h, t1, i));
    }

    for (i=0;i<N;i++){

        t3[i]=y[i]+ (k3[i]=step*f(x+h, t2, i));
    }

    for (i=0;i<N;i++){

        k4[i]= step*f(x+step, t3, i);
    }

    for (i=0;i<N;i++){

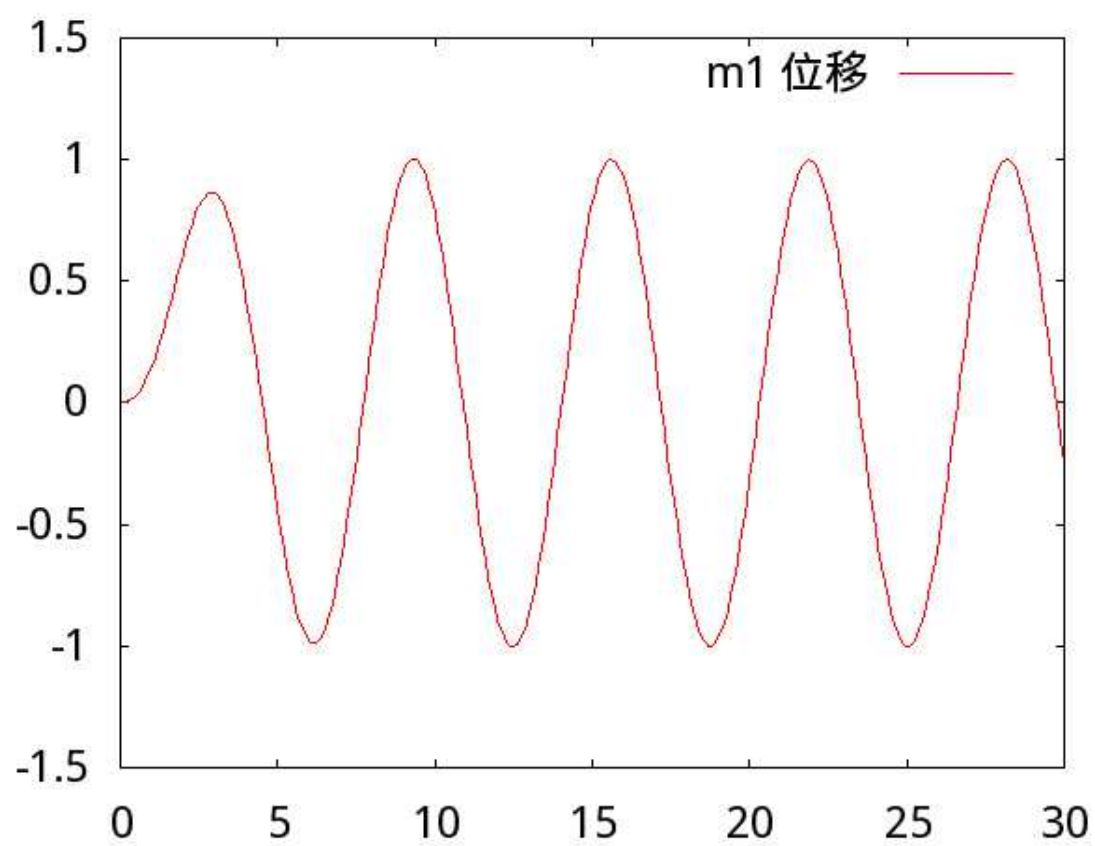
        y[i]+=(k1[i]+2*k2[i]+2*k3[i]+k4[i])/6.0;
    }
}

double f(double x, double y[], int i){
    double b1=1.0, b2=1.0, k1=1.0, m=1.0;
    if (i==0)
        x=y[1];

    if (i==1)
        x=-(b1/m)*y[1]-y[0]*(k1/m)+sin(x);

    return x;
}
```

圖十七 程式碼(題目四_C Runge-Kutta)



圖十八 模擬結果(題目四_C Runge-Kutta)

心得與討論

使用多種模擬方法來驗證模擬結果是否一致，20-sim 較容易上手，只需”看圖說故事”的方式來求解，不過要有能量守恆的觀念，相較於其他三種工具，都是需要推導公式，不過只要理解其中觀念，就可以很輕易的去求解。

參考資料

1. <http://blog.kmol.info>