二、MATLAB 动力学分析:四自由度弹阻质系统

(一)状态空间法

1. 状态空间法概述

经典控制理论中常用到传递函数来描述系统，但仍有一些不足之处：

- 系统模型为单输入单输出系统
- 忽略初始条件的影响
- 不包含系统的所有信息
- 无法利用系统的内部信息来改变系统的性能

近年来，机械系统的现代趋势是向高度复杂化发展，对于复杂的时变、非线性、多输入-多输出系统的问题，需要用对系统内部描述的新方法——状态空间分析法。这是一种对系统的完全描述，不同于传递函数的“黑箱”，状态空间法能够完全表征系统的所有动力学特征，属于内部描述，实现了各种不同的系统（单变量、多变量、时变、时不变、线性、 非线性等）描述形式的统一。适合描述复杂的动态系统。

现代控制理论中的状态空间理论发展迅速，以状态空间法为基础,分析和设计控制系统。近年来, 状态空间法逐步推广并成功地运用到军事、生物医学、社会经济及人类生活等诸多领域 , 并且有着广阔的发展前景[1]。它的出现，推动了控制理论的发展，实现了由古典控制理论向现代控制理论的过渡。

了解状态空间法首先需要清楚一些概念：

状态变量

能够完全表征系统动力学特征的一组独立变量成为系统的状态变量，是系统的内部变量。由状态变量构成的列向量 $x(t)=[x_1(t),x_2(t),\cdots x_n(t)]^T$ 称为状态向量。状态向量的取值空间称为状态空间。

状态模型

如果系统是线性定常的，并且可以用 n 个状态变量，r 个输入变量和 m 个输出变量描述。设一个 n 维线性控制系统的状态向量为 $x(t)$，输入向量为 $u(t)$，输出向量为 $y(t)$，状态模型=状态方程+输出方程。

状态方程具有的形式为

$$\begin{cases} \dot{x}_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_{11}u_1 + b_{12}u_2 + \cdots + b_{1r}u_r \\ \dot{x}_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + b_{21}u_1 + b_{22}u_2 + \cdots + b_{2r}u_r \\ \qquad\qquad\qquad\qquad \vdots \\ \dot{x}_n = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n + b_{n1}u_1 + b_{n2}u_2 + \cdots + b_{nr}u_r \end{cases} \tag{2.1}$$

它的输出方程具有的形式为

$$\begin{cases} \dot{y}_1 = c_{11}x_1 + c_{12}x_2 + \cdots + c_{1n}x_n + d_{11}u_1 + d_{12}u_2 + \cdots + d_{1r}u_r \\ \dot{y}_2 = c_{21}x_1 + c_{22}x_2 + \cdots + c_{2n}x_n + d_{21}u_1 + d_{22}u_2 + \cdots + d_{2r}u_r \\ \qquad\qquad\qquad\qquad \vdots \\ \dot{y}_n = c_{n1}x_1 + c_{n2}x_2 + \cdots + c_{nn}x_n + d_{n1}u_1 + d_{n2}u_2 + \cdots + d_{nr}u_r \end{cases} \tag{2.2}$$

式中，系数 $a_{ij},b_{ij},c_{ij},d_{ij}$ 为常数，其中有一些可能为零

用矢量矩阵表示为

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{2.3}$$

其中

$$x=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},\quad A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix},\quad B=\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1r} \\ b_{21} & b_{22} & \cdots & b_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nr} \end{bmatrix},\quad u=\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_r \end{bmatrix},$$

$$y=\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},\quad C=\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix},\quad D=\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1r} \\ d_{21} & d_{22} & \cdots & d_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nr} \end{bmatrix}$$

矩阵 *A,B,C,D* 分别称为状态矩阵、输入矩阵、输出矩阵和直接传递矩阵。矢量 *x,u,y* 分别称为状态矢量、输入矢量和输出矢量。状态矢量元素是状态变量，输入矢量元素是输入变量（如果系统仅包含一个输入变量，那么 *u* 就是一个标量）。输出矢量的元素是输出变量。

2. MATLAB 编程

在 MATLAB 中一个以状态空间形式表示的机械系统可以用语句

sys=ss(A,B,C,D)

来定义。状态矩阵 A、输入矩阵 B、输出矩阵 C 和直接传递矩阵 D 可以通过 MATLAB 的矩阵定义方法来定义。

对于状态空间定义的系统在单位阶跃输入下的响应，MATLAB 命令

step(sys)

将生成其单位阶跃曲线。对于多输入多输出的系统，step 命令产生一定数量的阶跃响应曲线，每条曲线都是系统每个输入与输出的组合。

命令 lsim 产生线性定常系统对任意输入的响应，如果以状态空间形式给出的初始条件为零，则

lsim(sys,u,t)

产生系统对任意的具有用户指定时间 t 的输入 u 的响应。如果初始条件在状态空间模型中不为零，则命令 lsim(sys,u,t,x_0)产生一个在输入为 u 和初始条件为 x_0 的系统响应。这些命令中的 sys 均以状态空间形式定义。

下面我们以一个案例展开，并再次基础上延伸拓展。

案例 如图 2- 1 所示的带有分支的质阻弹振动系统，已知系统的所有刚度系数为$1N/m$，阻尼 $c_1=2, c_2=c_3=1N\cdot s/m$， $m_1=4kg, m_2=1kg, m_3=3kg, m_4=2kg$；初始条件为：$t=0, x_1(0)=x_2(0)=x_3(0)=x_4(0)=1; \dot{x}_1(0)=\dot{x}_2(0)=\dot{x}_3(0)=\dot{x}_4(0)=10;$

用状态空间法进行 MATLAB 编程，探究振动规律。

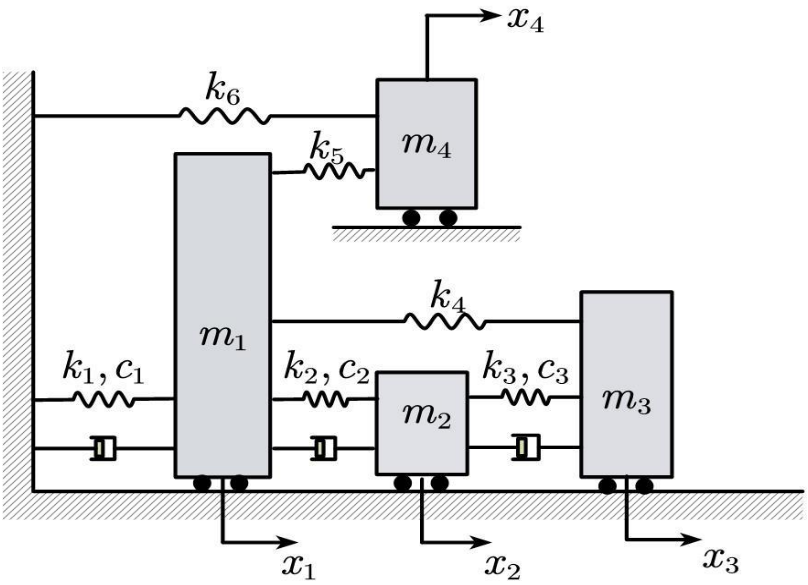


图 2- 1 带有分支的质阻弹系统

Step1. 设定系统参数

设定分析步的时间间隔为 0.01s，分析 50s

分析此模型，为四自由度系统。为了方便表述出矩阵 A、B、C、D，我们运用影响系数法提前计算刚度矩阵$[K]$和质量矩阵$[M]$,求解过程如下：

先只考虑静态，

令 $X=[1\quad 0\quad 0\quad 0]^T$,可得 $k_{11}=k_1+k_2+k_4+k_5, k_{21}=-k_2, k_{31}=-k_4, k_{41}=-k_5$

令 $X=[0\quad 1\quad 0\quad 0]^T$,可得 $k_{12}=-k_2, k_{22}=k_2+k_3, k_{32}=-k_3, k_{42}=0$

令 $X=[0\quad 0\quad 1\quad 0]^T$,可得 $k_{13}-k4, k_{23}=-k_3, k_{33}=k_3+k_4, k_{43}=0$

令 $X=[0\quad 0\quad 0\quad 1]^T$,可得 $k_{14}=-k_5, k_{24}=0, k_{34}=0, k_{44}=k_5+k_6$

因此刚度矩阵$[K]$为

$$K=\begin{bmatrix} k_1+k_2+k_4+k_5 & -k_2 & -k_4 & -k_5 \\ -k_2 & k_2+k_3 & -k_3 & 0 \\ -k_4 & -k_3 & k_3+k_4 & 0 \\ -k_5 & 0 & 0 & k_5+k_6 \end{bmatrix} \tag{2.4}$$

只考虑动态

令 $\ddot{X}=[1\quad 0\quad 0\quad 0]^T$,可得 $m_{11}=m_1, m_{21}=0, m_{31}=0, m_{41}=0$

令 $\ddot{X}=[0\quad 1\quad 0\quad 0]^T$,可得 $m_{12}=0, m_{22}=m_2, m_{32}=0, m_{42}=0$

令 $\ddot{X}=[0\quad 0\quad 1\quad 0]^T$,可得 $m_{13}=0, m_{23}=0, m_{33}=m_3, m_{43}=0$

令 $\ddot{X}=[0\quad 0\quad 0\quad 1]^T$,可得 $m_{14}=0, m_{24}=0, m_{34}=0, m_{44}=m_4$

因此，质量矩阵$[M]$为：

$$M=\begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix} \tag{2.5}$$

在计算矩阵 A、B、C、D 时，我们运用矩阵运算进行简化，基于刚度矩阵

$[K]$和质量矩阵$[M]$将矩阵 A、B、C、D 化为如下形式：

$$A=\begin{bmatrix} [0]_{4\times4} & [1]_{4\times4} \\ -[m]^{-1}*k & -[m]^{-1}*[c] \end{bmatrix} \tag{2.6}$$

$$B=[0]_{8\times1} \tag{2.7}$$

$$C=\begin{bmatrix} [1]_{4\times4} & [0]_{4\times4} \end{bmatrix} \tag{2.8}$$

$$D=[0]_{4\times1} \tag{2.9}$$

Step2. 定义初始条件

由题目可知该系统为有初始条件的有阻尼自由振动系统。

则初始条件可写为

$$x_0=\begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \\ \dot{x}_1(0) \\ \dot{x}_2(0) \\ \dot{x}_3(0) \\ \dot{x}_4(0) \end{bmatrix}=\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} \tag{2.10}$$

激励

$$u=[0]_{1\times5001} \tag{2.11}$$

Step3. 模型求解

运用状态空间函数 ss 定义系统，运用 lsim 函数求解系统响应。

Step4. 图形可视化

运用 plot 和 subplot 函数表示各个自由度的响应曲线。

在此基础上本组又创新地运用 *drawnow* 函数，将结果做成动态效果。

源程序如下：

源程序 ***(my_code4.m)***

```
%% 三自由度，任意激励（含动画）
%% 设定系统参数
t=0:0.01:50;
t=t';
m1=4; m2=1; m3=3; m4=2;
k1=1;k2=1;k3=1;k4=1;k5=1;k6=1;
c1=2;c2=1;c3=1;

k=[k1+k2+k4+k5,-k2,-k4,-k5;
          -k2,      k2+k3,    -k3,      0;
          -k4,       -k3,      k3+k4,  0;
          -k5,       0,         0,     k5+k6];

c=[c1+c2,-c2,0,0;
   -c2,c2+c3,-c3,0;
   0,-c3,c3,0;
```

```
    0,0,0,0];

m=[m1,0,0,0;
    0,m2,0,0;
    0,0,m3,0;
    0,0,0,m4];

dof=size(m,1); %判断有多少自由度

A=[zeros(dof,dof),eye(dof,dof);-inv(m)*k,-inv(m)*c];
B=[0;0;0;0;0;0;0;0];
C=[eye(dof),zeros(dof)];
D=zeros(dof,1);

sys=ss(A,B,C,D);
%% 定义初始条件
x0=[1;1;1;1;10;10;10;10];  %初始条件 z1,z2,z3,z4,z1_dot,z2_dot,z3_dot,z4_dot
u=zeros(1,5001);
% u1=[0:0.01:1];
% u2=[0.99:-0.01:-1];
% u3=[-0.99:0.01:0];
% u4=0*[4.01:0.01:10];
% u=[u1,u2,u3,u4];
% u=ones(1,2001); %也是单位阶跃
% u=sin(0.05*t);%可以任意设定其他激励
%% 求解
y=lsim(sys,u,t,x0); %含初始条件、激励
% y=step(sys,t);
y1=y(:,1);
y2=y(:,2);
y3=y(:,3);
y4=y(:,4);

%% 绘图
figure(3);
x1=y1;
subplot(4,2,1);
plot(t,z1);
grid on;
title('x1,x2,x3,x4 响应');
ylabel('x1/m');


x2=y2;
subplot(4,2,3);
plot(t,x2);
grid on;
ylabel('x2/m');

x3=y3;
subplot(4,2,5);
plot(t,x3);
grid on;
ylabel('x3/m');

x4=y4;
subplot(4,2,7);
plot(t,x4);
grid on;
xlabel('t/s');
ylabel('x4/m');

%将 z1,z2,z3 放在一张图中对比
```

```
subplot(1,2,2);
plot(t,x1,'r',t,x2,'b',t,x3,'m',t,x4,'c');
hold on
title('x1,x2,x3,x4 对比图');
legend('x1','x2','x3','x4');
xlabel('t/s');
ylabel('x/m')

hold on
p1=plot(t(1),z1(1),'o','MarkerFaceColor','r');
p2=plot(t(1),z2(1),'o','MarkerFaceColor','b');
p3=plot(t(1),z3(1),'o','MarkerFaceColor','m');
p4=plot(t(1),z4(1),'o','MarkerFaceColor','c');
hold off
axis manual
for k = 2:length(t)
    p1.XData = t(k);
    p1.YData = z1(k);
    p2.XData=t(k);
    p2.YData=z2(k);
    p3.XData=t(k);
    p3.YData=z3(k);
    p4.XData=t(k);
    p4.YData=z4(k);
    drawnow
end
```

结果展示

（含有动态效果）

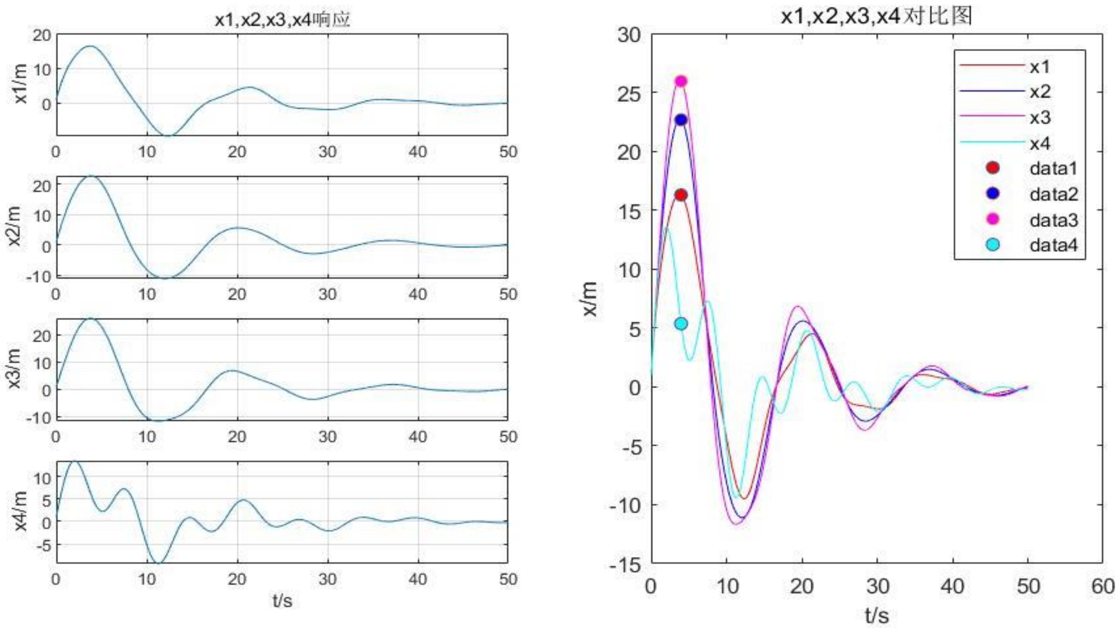


图 2- 2 四自由度 自由振动的响应曲线（动态）

案例拓展：

基于 MATLAB 强大的计算能力，我们不满足于案例给予的条件，将案例扩展为受迫振动，假设在 m4 物体上施加正弦激励 $f(t)=\sin(t)$，如图：

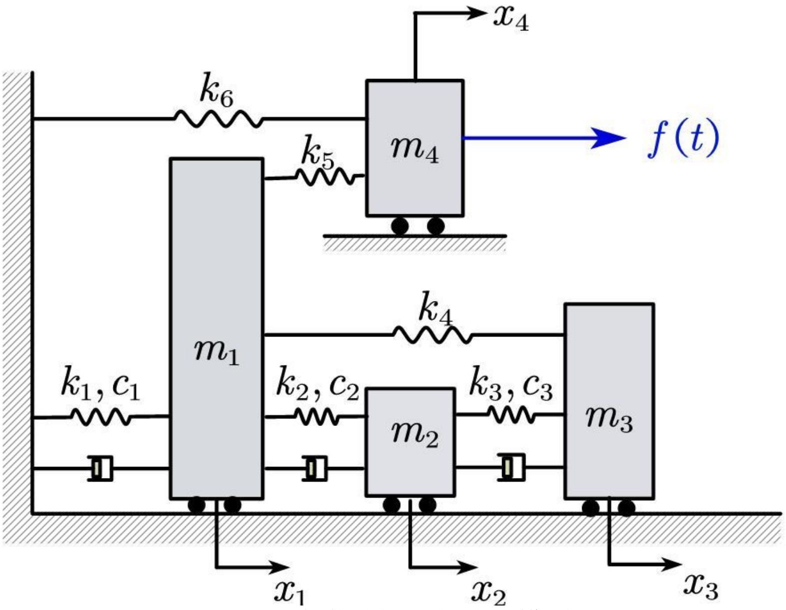


图 2- 3 含激励的系统物理模型

这时，只需将激励 u 改为 *u=sin(1*t)*

将 $[B]$ 矩阵改为

$$B=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dfrac{1}{m_4} \end{bmatrix} \tag{2.12}$$

即可在 m4 上施加激励。事实上，只要改动矩阵 $[B]$，就可对任意自由度施加激励。

源程序 *(my_code4_2.m)*

```
%% 四自由度，任意激励（含动画）
%% 设定系统参数
clc;clear;
t=0:0.01:50;
t=t';
m1=4; m2=1; m3=3; m4=2;

k1=1;k2=1;k3=1;k4=1;k5=1;k6=1;
c1=2;c2=1;c3=1;
```

```
k=[k1+k2+k4+k5,-k2,-k4,-k5;
        -k2,      k2+k3,     -k3,      0;
        -k4,        -k3,       k3+k4,  0;
        -k5,        0,          0,      k5+k6];

c=[c1+c2,-c2,0,0;
   -c2,c2+c3,-c3,0;
   0,-c3,c3,0;
   0,0,0,0];

m=[m1,0,0,0;
    0,m2,0,0;
    0,0,m3,0;
    0,0,0,m4];

dof=size(m,1); %判断有多少自由度

A=[zeros(dof,dof),eye(dof,dof);-inv(m)*k,-inv(m)*c];
B=[0;0;0;0;0;0;0;1/m4];
C=[eye(dof),zeros(dof)];
D=zeros(dof,1);

sys=ss(A,B,C,D);
%% 定义初始条件
x0=[1;1;1;1;10;10;10;10];  %初始条件 z1,z2,z3,z4,z1_dot,z2_dot,z3_dot,z4_dot

u=sin(1*t);%可以任意设定其他激励
%% 求解
y=lsim(sys,u,t,x0); %含初始条件、激励
% y=step(sys,t);
x1=y(:,1);
x2=y(:,2);
x3=y(:,3);
x4=y(:,4);

%% 绘图
figure(1);
subplot(4,2,1);
plot(t,x1);
grid on;
title('x1,x2,x3,x4 响应');
ylabel('x1/m');

subplot(4,2,3);
plot(t,x2);
grid on;
ylabel('x2/m');

subplot(4,2,5);
plot(t,x3);
grid on;
ylabel('x3/m');

subplot(4,2,7);
plot(t,x4);
grid on;
xlabel('t/s');
ylabel('x4/m');

%将 z1,z2,z3 放在一张图中对比
subplot(1,2,2);
plot(t,x1,'r',t,x2,'b',t,x3,'m',t,x4,'c');
```

```
hold on
title('x1,x2,x3,x4 对比图');
legend('x1','x2','x3','x4');
xlabel('t/s');
ylabel('x/m')

hold on
p1=plot(t(1),x1(1),'o','MarkerFaceColor','r');
p2=plot(t(1),x2(1),'o','MarkerFaceColor','b');
p3=plot(t(1),x3(1),'o','MarkerFaceColor','m');
p4=plot(t(1),x4(1),'o','MarkerFaceColor','c');
hold off
axis manual
for k = 2:length(t)
    p1.XData = t(k);
    p1.YData = x1(k);
    p2.XData=t(k);
    p2.YData=x2(k);
    p3.XData=t(k);
    p3.YData=x3(k);
    p4.XData=t(k);
    p4.YData=x4(k);
    drawnow
end
```

响应如图所示，可以看出在受迫振动的情况下，表现出与自由振动不同的响应曲线。

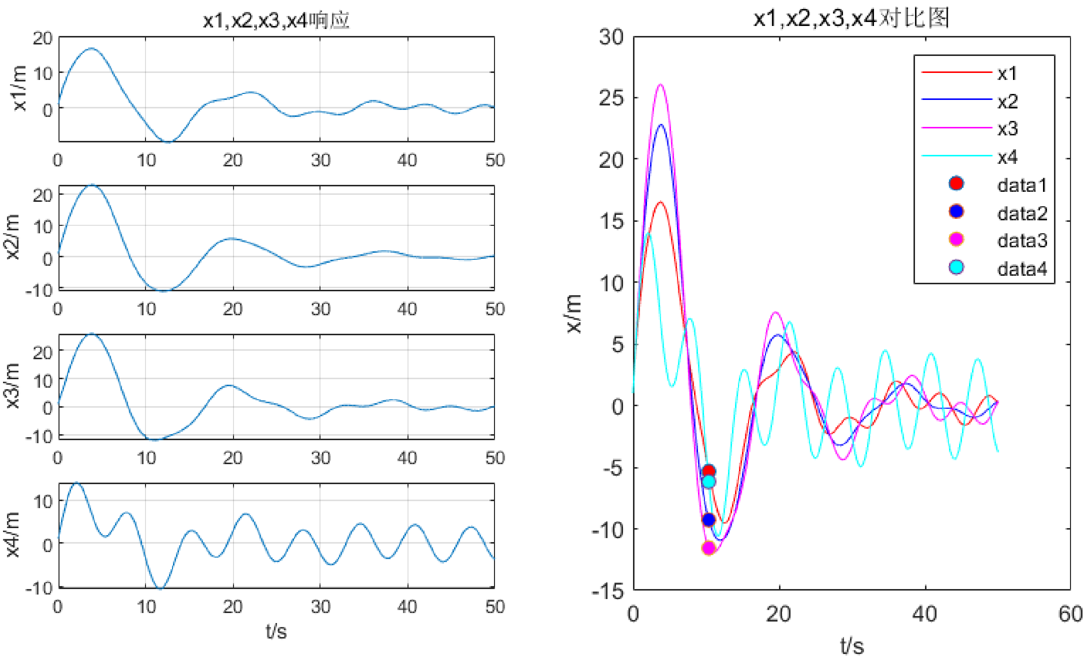


图 2- 4 含激励的受迫振动响应

(二)Simulink 仿真

1. Simulink 简介

Simulink 是 MATLAB 的一个交互式动态系统建模、仿真和分析工具，它具有高性能的科学计算能力，被广泛应用于线性、非线性、离散、连续及多变量系统的仿真和分析。通过 Simulink 模块库提供的各种模块，建立仿真模型，再选取适当信号模块作为信号输入，选取输出模块来观察系统响应，分析系统动态特性。

1) Simulink 建模三个特点

①可视化：simulink 仿真采用交互式开发的方法，操作简单、直观、用户只需要拖拽鼠标即可实现动态系统的仿真。图形化的界面可以避免过多的编程，同时又可以直观的反映仿真的过程。

②扩展性强：simulink 有较强的扩展性，用户可以根据自己的需求来编写自己的模块库，建立子系统，封装子系统。

③灵活性强：simulink 是一个非常灵活的仿真建模工具，虽然 MATLAB 为用户提供封装了大量的模块，但是用户在使用的时候也可以修改里面的参数。近年来在各大领域的得到了大量的应用。

2) Simulink 数据类型

Simulink 支持 MATLAB 内置的所有数据类型，绝大多数的模块都默认 double 的数据类型。在 simulink 模型窗口中选择 Display——>signal&ports——>port Date Types 命令，可以查看信号的数据数据类型和输入输出的数据类型。

3) Simulink 模块库

Simulink 模块库提供了各种领域的基本模块，按照实际应用及功能组成若干子库。MATLAB 在储存这些子库时都按照功能都分门别类以便查找，每一类就是一个模块库。

Commonly Used　Blocks 模块库 常用的模块库
Sources 模块库，为仿真提供各种信号源
Sinks 模块库，为仿真提供输出设备原件
Continuous 模块库，为仿真提供连续系统
Discrete 模块库，为仿真提供离散元件
Math 模块库，提供数学运算功能元件
Function&Tables 模块库，自定义函数和线形插值查表模块库
Nonlinear 模块库，非连续系统元件
Signals&System 模块库，提供用于输入、输出和控制的相关信号及相关处理
Subsystems 模块库，各种子系统

4)利用 Simulink 进行仿真的主要步骤为[2]：

①对系统进行分析，建立数学模型；
②在 Simulink 中建立系统的仿真模型并设置仿真参数
③执行仿真得出结果并加以分析

2. 仿真案例

在 Simulink 中，机械振动分析可以采用下面两种方法建立仿真模型：①利用

积分模块建模；②利用状态空间模块建模。本文对这两种方法均进行了建模，并将两种不同的方法进行对比，得出结论。采用的例子与前文相同，以便更好地对不同结果进行验证。

案例 如图 2-1 所示的带有分支的质阻弹振动系统，已知系统的所有刚度系数为$1N/m$，阻尼 $c_1=2, c_2=c_3=1N\cdot s/m$，$m_1=4kg, m_2=1kg, m_3=3kg, m_4=2kg$；初始条件为：$t=0, x_1(0)=x_2(0)=x_3(0)=x_4(0)=1; \dot{x}_1(0)=\dot{x}_2(0)=\dot{x}_3(0)=\dot{x}_4(0)=10;$
用 Simulink 进行 MATLAB 仿真，探究振动规律。

2.1 积分模块建模

分析系统的受力情况，便于建模，如**图 2-5** 所示。

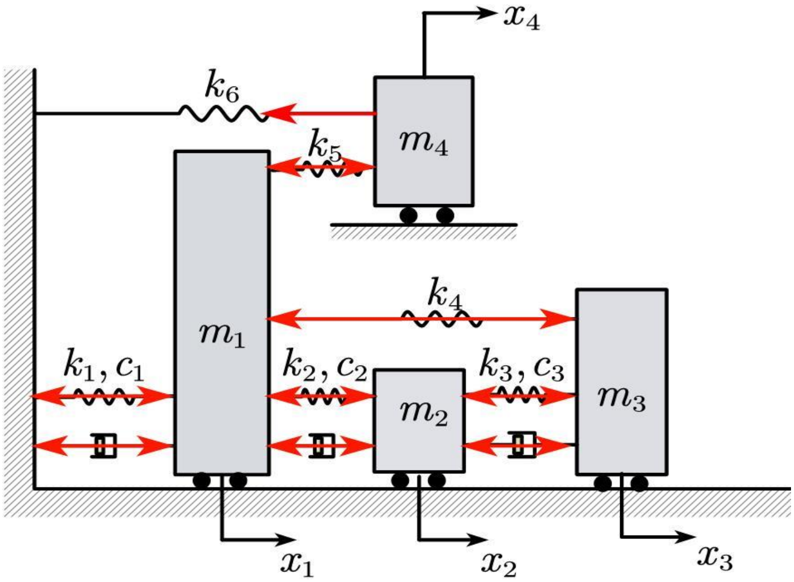


图 2-5 带有分支的质阻弹系统 受力情况

由牛顿第二定律列出如下的动力学方程：

$$\begin{cases} k_1x_1+c_1\dot{x}_1+k_2(x_1-x_2)+c_2(\dot{x}_1-\dot{x}_2)+k_4(x_1-x_3)+k_5(x_1-x_4)+m_1\ddot{x}_1=0 \\ -k_2(x_1-x_3)-c_2(\dot{x}_1-\dot{x}_2)+k_3(x_2-x_3)+c_3(\dot{x}_2-\dot{x}_3)+m_2\ddot{x}_2=0 \\ -k_4(x_1-x_3)-k_3(x_2-x_3)-c_3(\dot{x}_2-\dot{x}_3)+m_3\ddot{x}_3=0 \\ k_6x_4-k_5(x_1-x_4)+m_4\ddot{x}_4=0 \end{cases} \tag{2.13}$$

初始条件为 $x_1(0)=x_2(0)=x_3(0)=x_4(0)=1; \dot{x}_1(0)=\dot{x}_2(0)=\dot{x}_3(0)=\dot{x}_4(0)=10$

并将公式(2.13)进行如下变形得到公式(2.14)

$$\begin{cases} \ddot{x}_1=-\dfrac{1}{m_1}\left[(k_1+k_2+k_4+k_5)x_1+(c_1+c_2)\dot{x}_1-k_2x_2-c_2\dot{x}_2-k_4x_3-k_5x_4\right] \\ \ddot{x}_2=-\dfrac{1}{m_2}\left[-k_2x_1-c_2\dot{x}_1+(k_2+k_3)x_2+(c_2+c_3)\dot{x}_2-k_3x_3-c_3\dot{x}_3\right] \\ \ddot{x}_3=-\dfrac{1}{m_3}\left[-k_4x_1-k_3x_2-c_3\dot{x}_2+(k_3+k_4)x_3+c_3\dot{x}_3\right] \\ \ddot{x}_4=-\dfrac{1}{m_4}\left[-k_5x_1+(k_5+k_6)x_4\right] \end{cases} \tag{2.14}$$

根据式(2.14)可方便地建立可视化模型。详细建模过程如下

Step1. x1

根据(2.14)的第一行公式可得

$\ddot{x}_1 = -\dfrac{1}{m_1}\left[(k_1+k_2+k_4+k_5)x_1+(c_1+c_2)\dot{x}_1-k_2x_2-c_2\dot{x}_2-k_4x_3-k_5x_4\right]$，这是关于 x1 的二阶微分方程。中括号中一共有 6 项，可知我们需要运用 Add 模块进行求解。其中对增益模块分别指定为 k_1,k_2,k_3,k_4,k_5，c_1,c_2 或 $1/m_1$，积分模块 integrator1、integrator2 的初值 *(initial condition)* 分别指定为 $\dot{x}_0$、x_0（如**图 2-7** 所示），在运行前对 k_1,k_2,k_3,k_4,k_5，c_1,c_2 或 $1/m_1$ 进行赋值。两个示波器模块分别用来实时输出质点的位移 x_1 和速度 v_1 曲线。

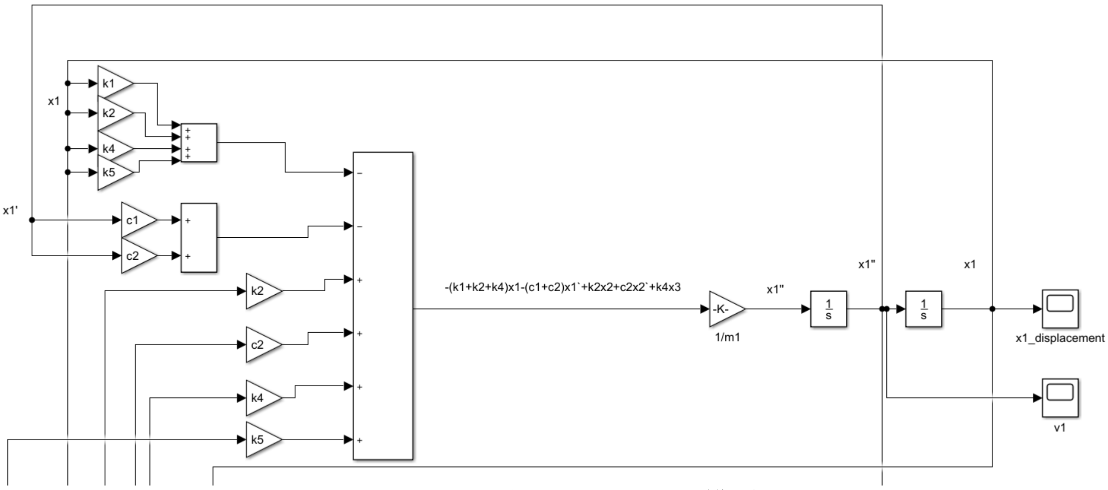


图 2-6 质点 1 的 simulink 模型

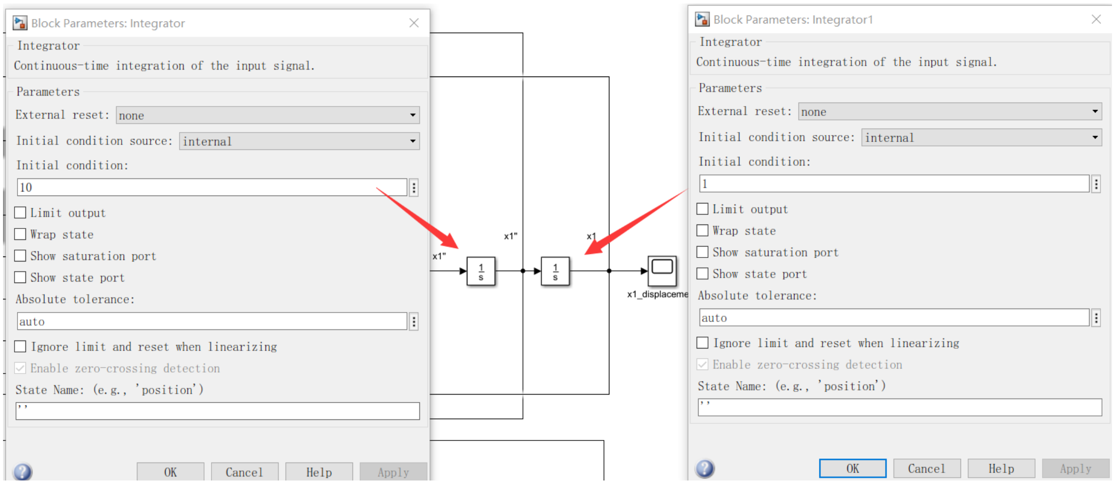


图 2-7 积分模块参数设置

Step2. x2

根据(2.14)的第二行公式可得

$\ddot{x}_2 = -\dfrac{1}{m_2}\left[-k_2x_1-c_2\dot{x}_1+(k_2+k_3)x_2+(c_2+c_3)\dot{x}_2-k_3x_3-c_3\dot{x}_3\right]$，这是关于 x2 的二阶微分方程。中括号中一共有 6 项，可知我们需要运用 Add 模块进行求解。其中对增益模块分别指定为 k_2,k_3，c_2,c_3 或 $1/m_2$，积分模块 integrator3、integrator4 的初值 *(initial condition)* 分别指定为 $\dot{x}_0$、x_0（同上），在运行前对 k_2,k_3，c_2,c_3 或 $1/m_2$

进行赋值。两个示波器模块分别用来实时输出质点的位移 x_2 和速度 v_2 曲线。

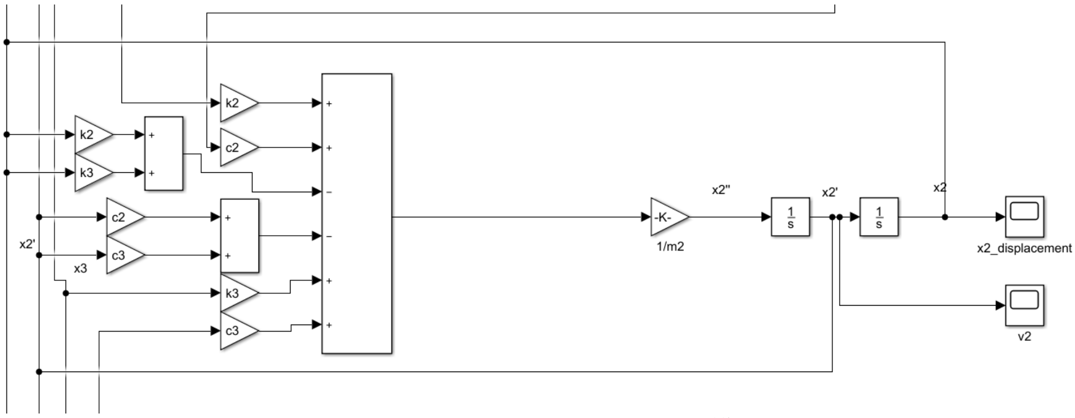


图 2- 8 质点 2 的 simulink 模型

Step3. x3

根据(2.14)的第三行公式可得

$\ddot{x}_3 = -\dfrac{1}{m_3}\left[-k_4x_1 - k_3x_2 - c_3\dot{x}_2 + (k_3+k_4)x_3 + c_3\dot{x}_3\right]$，这是关于 x3 的二阶微分方程。中括号中一共有 5 项，可知我们需要运用 Add 模块进行求解。其中对增益模块分别指定为 k_4, k_3， c_3 或 $1/m_3$， 积分模块 integrator5、integrator6 的初值*(initial condition)*分别指定为 $\dot{x}_0$、x_0（同上），在运行前对 k_4, k_3， c_3 或 $1/m_3$ 进行赋值。两个示波器模块分别用来实时输出质点的位移 x_3 和速度 v_3 曲线。

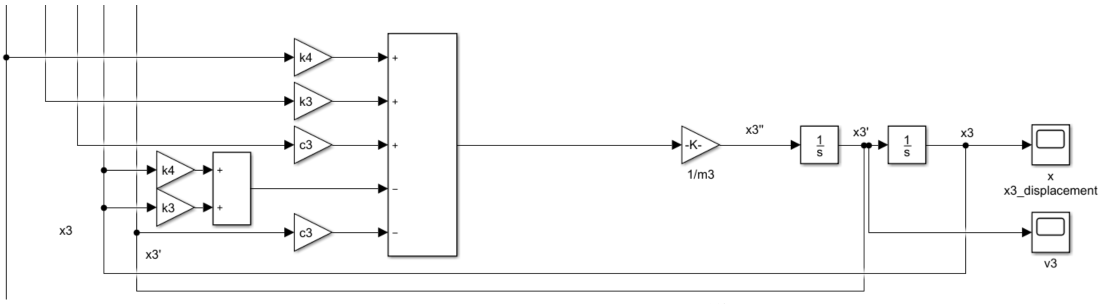


图 2- 9 质点 3 的 simulink 模型

Step4. x4

根据(2.14)的第四行公式可得

$\ddot{x}_4 = -\dfrac{1}{m_4}\left[-k_5x_1 + (k_5+k_6)x_4\right]$，这是关于 x4 的二阶微分方程。中括号中一共有 2 项，可知我们需要运用 Add 模块进行求解。其中对增益模块分别指定为 k_5, k_6，$1/m_4$， 积分模块 integrator7、integrator8 的初值*(initial condition)*分别指定为 $\dot{x}_0$、x_0（同上），在运行前对 k_5, k_6， $1/m_4$ 进行赋值。两个示波器模块分别用来实时输出质点的位移 x_4 和速度 v_4 曲线。

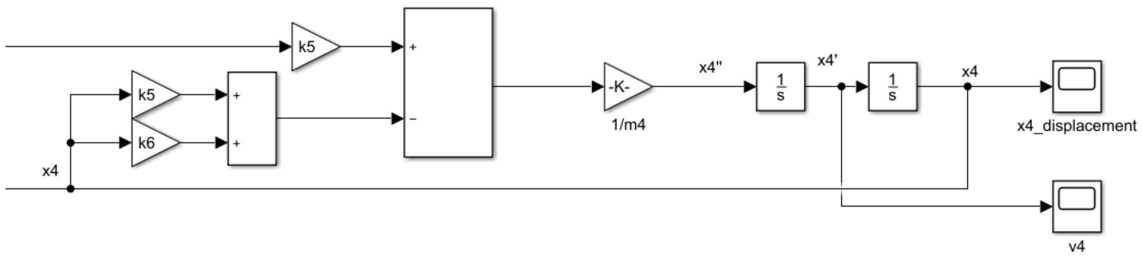


图 2- 10 质点 4 的 simulink 模型

运行 *parameter_dof4.m* 文件后，得到工作区 $k1 \ldots k6, m1 \ldots m4, c1 \ldots c3$ 的值。

parameter_dof4.m

```
clc;clear;
m1=4; m2=1; m3=3; m4=2;
invm1=1/m1; invm2=1/m2; invm3=1/m3; invm4=1/m4;
k1=1;k2=1;k3=1;k4=1;k5=1;k6=1;
c1=2;c2=1;c3=1;

k=[k1+k2+k3+k4,-k2,-k4,-k5;
        -k2,      k2+k3,    -k3,     0;
        -k4,       -k3,      k3+k4,  0;
        -k5,       0,         0,     k5+k6];

c=[c1+c2,-c2,0,0;
   -c2,c2+c3,-c3,0;
   0,-c3,c3,0;
   0,0,0,0];

m=[m1,0,0,0;
    0,m2,0,0;
    0,0,m3,0;
    0,0,0,m4];

dof=size(m,1); %判断有多少自由度
A=[zeros(dof,dof), eye(dof,dof);-inv(m)*k,-inv(m)*c];
B=[zeros(dof,dof);inv(m)];
C=[eye(dof),zeros(dof)];
D=zeros(dof,dof);
```

然后启动 Simulink，新建模型文件，打开空白编辑窗口，建立如图 2- 11 所示的总的仿真模型。其中各 Gain(增益)皆取自工作区的变量，Scope（示波器）用来显示各自由度的位移动态变化。双击即可显示结果。

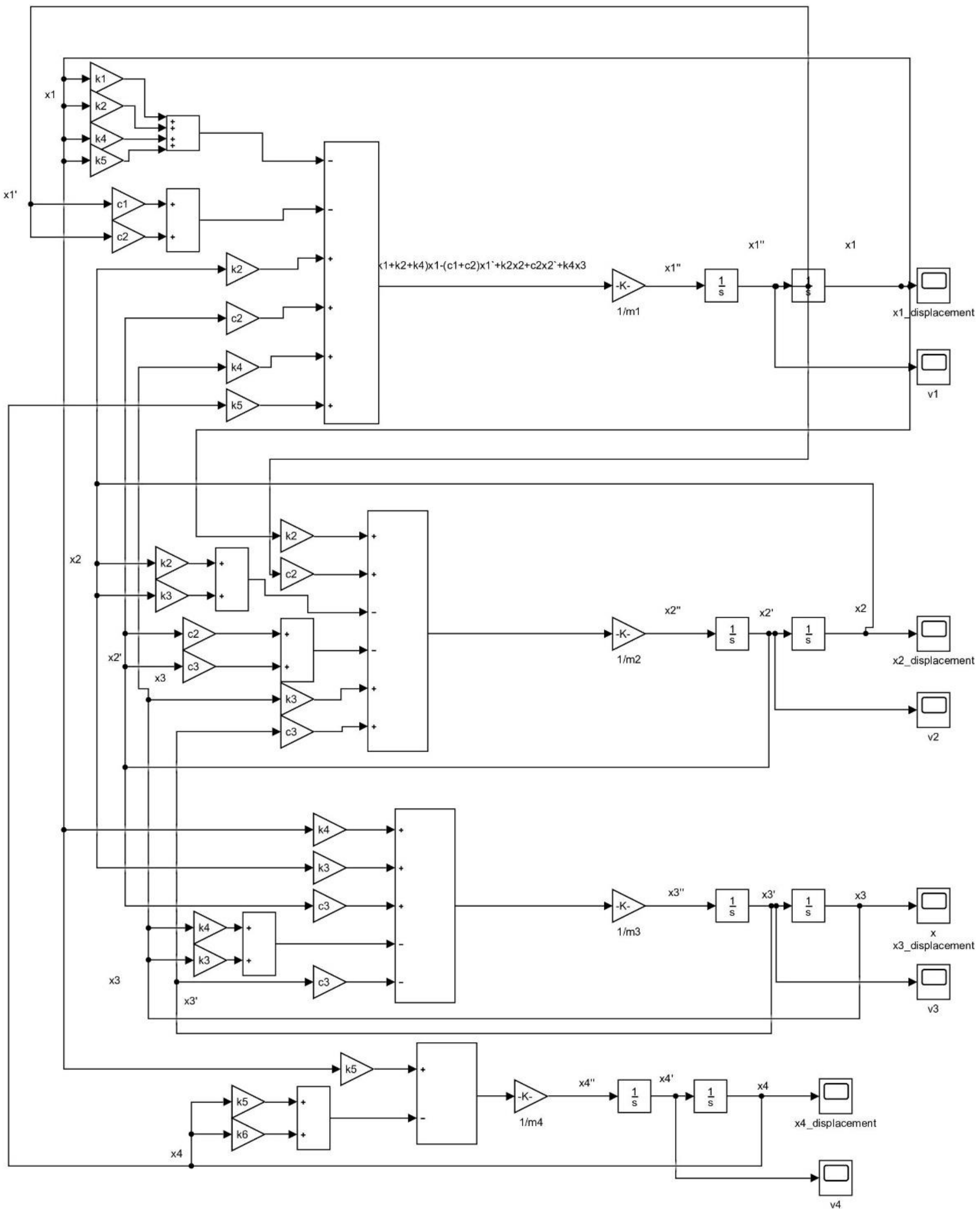
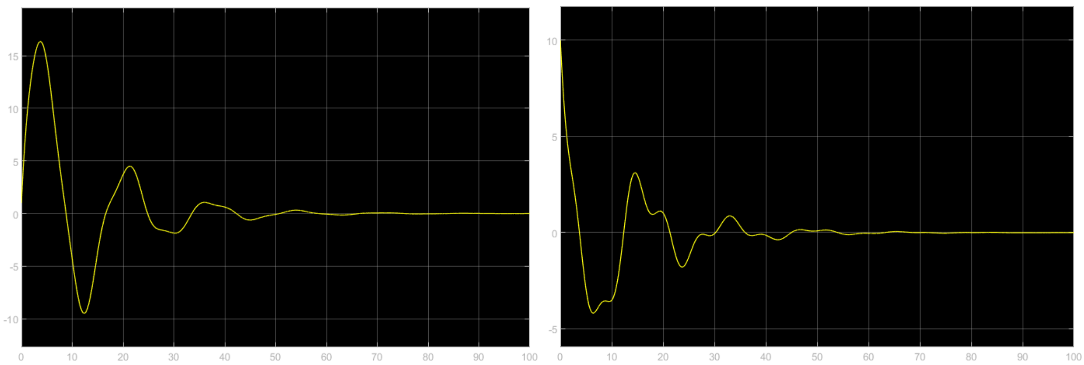
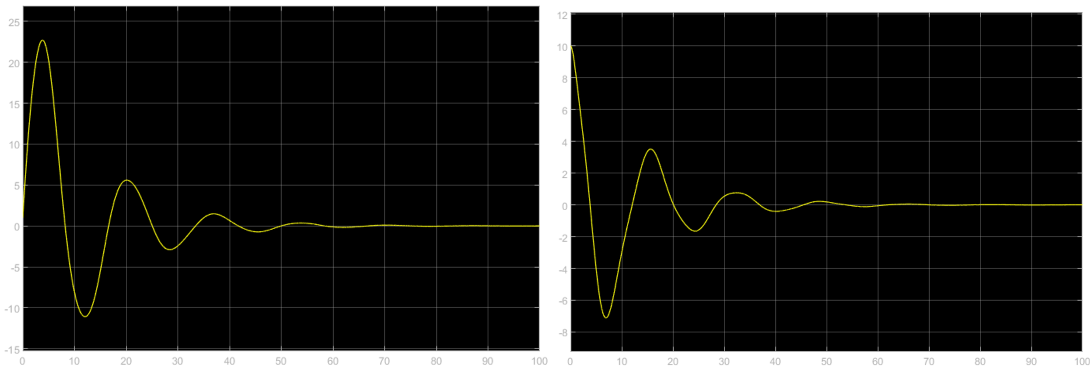


图 2- 11 总的积分模块建模

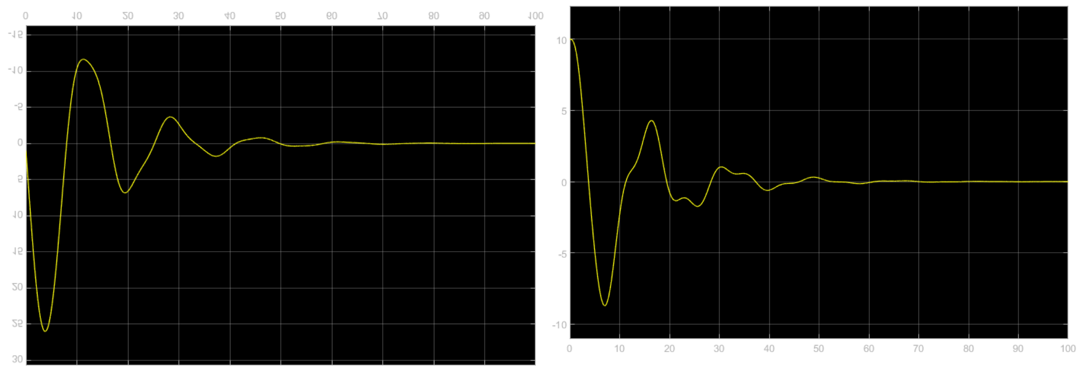
运行结果如**图 2- 11** 所示



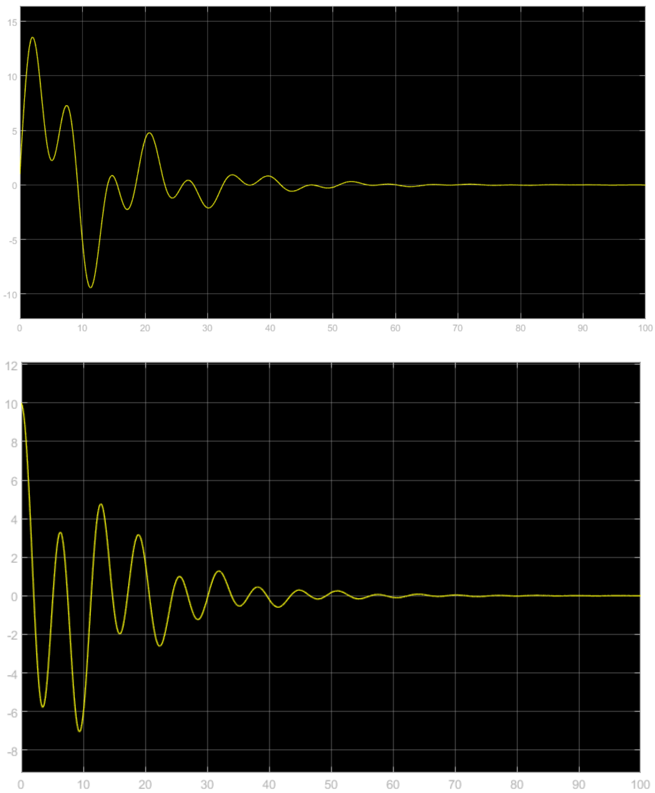
(a) x1，v1 响应曲线



(b) x2, v2 响应曲线



(c) x3, v3 响应曲线



(d) x4,v4 响应曲线

图 2- 12 积分模型得到的各自由度位移曲线

2.2 State-Space 模块建模

上述方法是基于微分方程建立的，可以看出过程比较复杂，灵活性不强。因此我们改进方法，将前文所述的状态空间法应用于 Simulink 仿真模块中，可以更方便快捷地建立仿真模型。

模块说明：

State-Space 模块可实现具有定义的如下行为的系统：

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

$$x|_{t=t0} = x_0$$

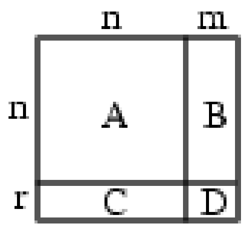
其中 x 是状态向量，u 是输入向量，y 是输出向量，x 是状态向量的初始条件。矩阵系数必须具有以下特征：

A 必须是 n×n 矩阵，其中 n 是状态的数量。

B 必须是 n×m 矩阵，其中 m 是输入的数量。

C 必须是 r×n 矩阵，其中 r 是输出的数量。

D 必须是 r×m 矩阵。



一般情况下，模块有一个输入端口和一个输出端口。C 或 D 矩阵中的行数与输出端口的宽度相同。B 或 D 矩阵中的列数与输入端口的宽度相同。

建模过程：

启动 MATLAB- Simulink 之后, 新建模型文件,打开一空白的编辑窗口.采用 Continuous 库中的 State-Space 模块进行建模，采用输出子模块库中的 Scope(示波器)模块来分别显示 *displacement*(位移),*velocity*(速度),*acceleration*(加速度)的仿真结果.建立仿真模型如图 2- 13 所示。

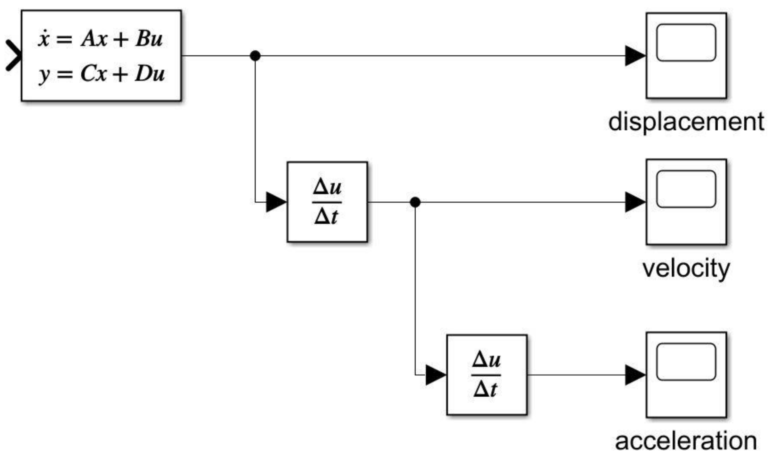


图 2- 13 State-Space 模块建模模型

其中 A,B,C,D 的参数同样需要运行 *parameter_dof_4.m* 文件后得到。参数设置如下图所示：

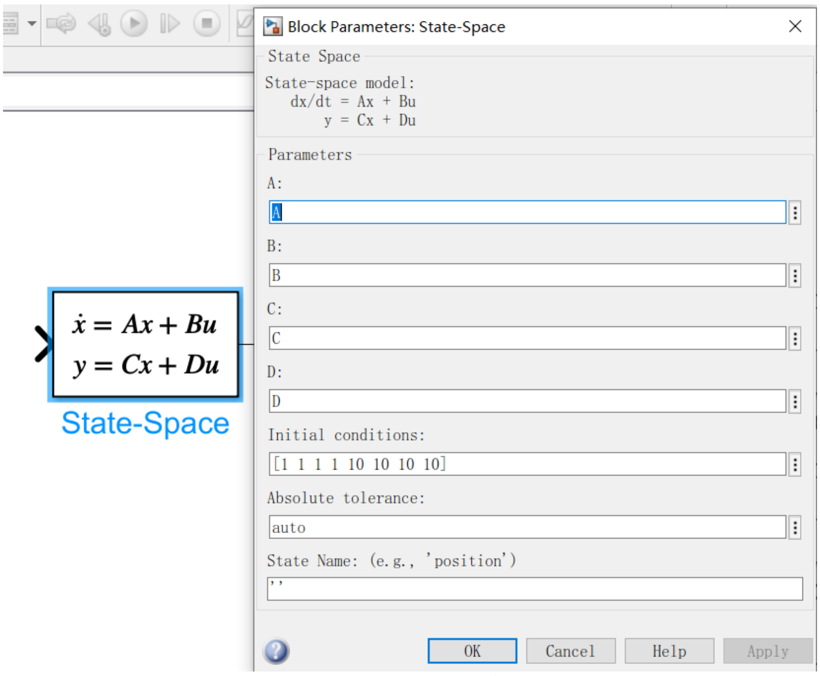


图 2- 14 State-space 模块参数设置

单击 *Run* 按钮运行结果，输出结果如**图 2- 15**，**图 2- 16**，**图 2- 17** 所示

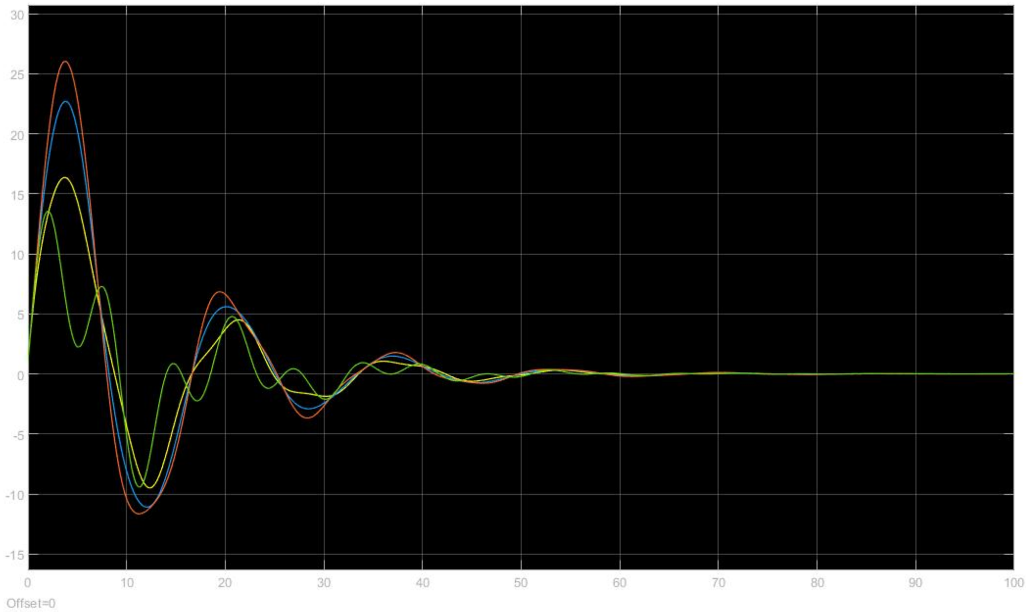


图 2- 15 state space-Simulink 模型得到的位移

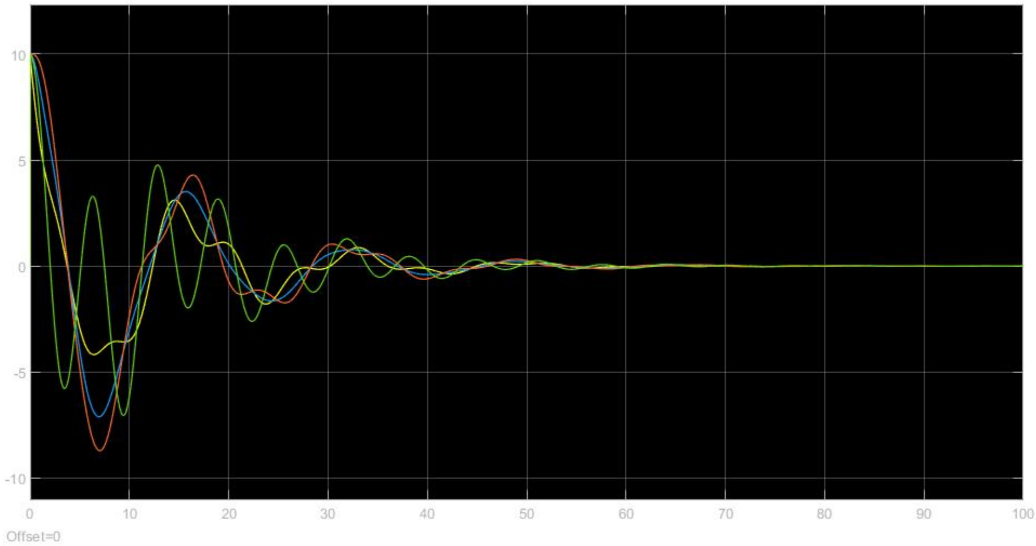


图 2- 16 state space-Simulink 模型得到的速度

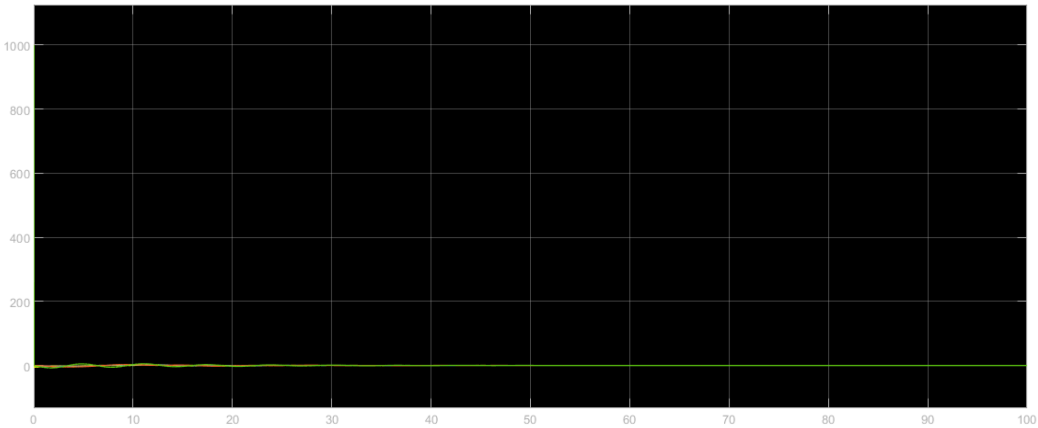


图 2- 17 state space-Simulink 模型得到的加速度

可以看出结果与积分模块建模得到的结果相同，可以验证结果的正确性。