**实验一 MATLAB基础**

1. **对于矩阵，以下四个指令的结果是什么？并解释。**

（1）

解释：使对矩阵A中各变量求取0.5次方

（2）

解释：是对矩阵A求取开方

（3）

解释：是对矩阵A各变量求取算术平方根

（4）

解释：是对矩阵求取开方

**2. 分别用MATLAB窗口命令、文本文件和函数文件三种方式求以下函数在(-2,-3,4)的值。**

****

1. MATLAB窗口命令

|  |
| --- |
| >> x=[-2 -3 4]  x =  -2 -3 4  >> y1=3\*x(1)^2+abs(x(2))+sqrt(x(3))  y1 =  17  >> y2=3\*x(1)^2-x(2)-x(3)  y2 =  11 |

1. MATLAB文本文件

**程序如下：**

|  |
| --- |
| x=[-2 -3 4];  y1=3\*x(1)^2+abs(x(2))+sqrt(x(3));  disp(y1)  y2=3\*x(1)^2-x(2)-x(3);  disp(y2) |

**程序输出：**

|  |
| --- |
| >> sy1\_021  17  11 |

1. MATLAB函数文件

**程序如下：**

|  |
| --- |
| x0=[-2 -3 4];  [y1, y2]=solve(x0);  function [y1, y2]=solve(x)  y1=3\*x(1)^2+abs(x(2))+sqrt(x(3))  y2=3\*x(1)^2-x(2)-x(3)  end |

**程序输出：**

|  |
| --- |
| >> sy1\_022  y1 =  17  y2 =  11 |

**3. 某城市的人口数量（单位是百万）如表所示，用两种不同的插值方法对没有进行人口统计年份的人口数量进行预测。**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **年代** | **1910** | **1920** | **1930** | **1940** | **1950** | **1960** | **1970** | **1980** | **1990** | **2000** |
| **人口** | **75.995** | **91.972** | **105.711** | **123.203** | **131.669** | **150.697** | **179.323** | **203.212** | **226.505** | **249.633** |

1. 一维线性插值

**程序如下：**

|  |
| --- |
| x=[1910 1920 1930 1940 1950 1960 1970 1980 1990 2000];  y=[75.995 91.972 105.711 123.203 131.669 150.697...  179.323 203.212 226.505 249.633];  x1=1910:1:2000;  y1=interp1(x,y,x1,'linear');  plot(x,y,'pk',x1,y1,'-b') |

**对应图像输出：**



1. 三次样条插值

**程序如下：**

|  |
| --- |
| x=[1910 1920 1930 1940 1950 1960 1970 1980 1990 2000];  y=[75.995 91.972 105.711 123.203 131.669 150.697...  179.323 203.212 226.505 249.633];  x1=1910:1:2000;  y1=interp1(x,y,x1,'spline');  plot(x,y,'pk',x1,y1,'-r') |

**对应图像输出：**



**4. 已知，求 满足 有 。画出曲线形状。**

**程序如下：**

|  |
| --- |
| syms t;  syms x;  y=exp(-t)\*cos(10\*t);  y1=@(x)abs(exp(-x).\*cos(10\*x))-0.05;  [ts]=fzero(y1,5)  fplot(y,[0,5],'-k');  hold on  ys=exp(-ts)\*cos(10\*ts);  plot(ts,ys,'pr')  plot([0,5],[0.05 0.05],'-.b',[0,5],[-0.05 -0.05],'-.b');  hold off |

**程序输出：**

|  |
| --- |
| >> sy1\_04  ts =  2.8755 |

**对应图像输出：**



**实验二 控制系统的数学模型**

1. **求以下传递函数的状态空间表达式，并说明其为何种形式。**



**程序代码如下：**

|  |
| --- |
| num=[1 1 1];  den=[1 6 11 6];  G=tf(num,den);  [A,B,C,D]=tf2ss(num,den) |

**程序输出如下：**

|  |
| --- |
| A =  -6 -11 -6  1 0 0  0 1 0  B =  1  0  0  C =  1 1 1  D =  0 |

**说明：**

A矩阵为的矩阵

B矩阵为的矩阵

C矩阵为的矩阵

D矩阵为的零矩阵

1. **已知系统的状态空间表达式，求其传递函数矩阵。**



**程序如下：**

|  |
| --- |
| A=[0 1;-2 -3];B=[1 0;1 1];  C=[1 0;1 1];D=zeros(2,2);  [num1,den1]=ss2tf(A,B,C,D,1)  [num2,den2]=ss2tf(A,B,C,D,2) |

**程序输出：**

|  |
| --- |
| num1 =  0 1.0000 4.0000  0 2.0000 2.0000  den1 =  1 3 2  num2 =  0 0 1  0 1 1  den2 =  1 3 2 |

1. **求以下传递函数的零极点模型及其最小实现，并说明两者区别。**



**程序如下：**

|  |
| --- |
| %零极点模型  num=[5 50 155 150];  den=[1 11 41 61 30];  [z,p,k]=tf2zp(num,den)  %最小实现  [Zm,Pm]=minreal(z,p) |

**程序输出：**

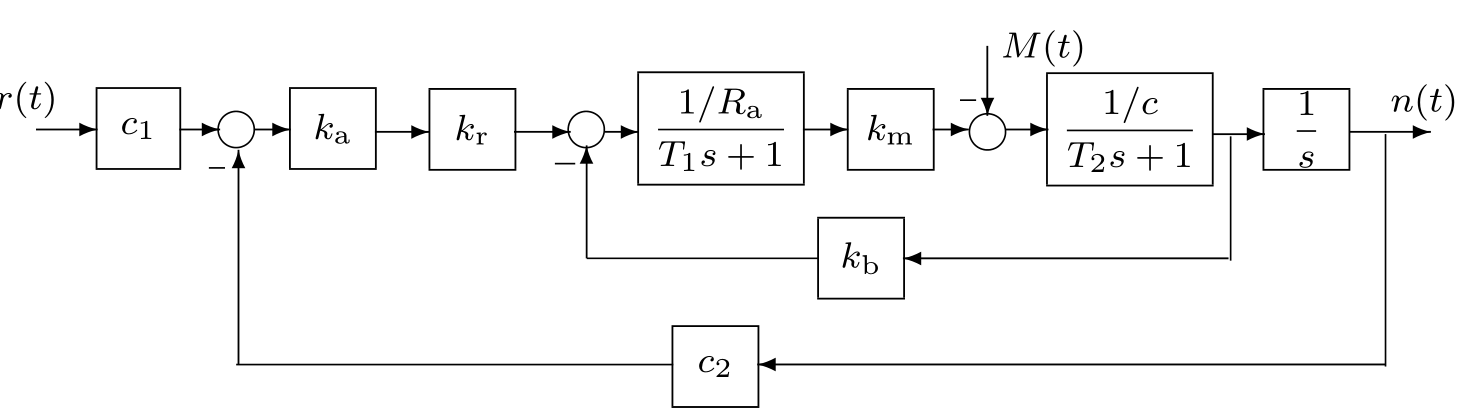
|  |
| --- |
| %零极点模型  z =  -5.0000  -3.0000  -2.0000  p =  -5.0000  -3.0000  -2.0000  -1.0000  k =  5  %最小实现  3 pole-zero(s) cancelled  Zm =  空的 0×1 double 列向量  Pm =  -1.0000 |

**说明：**

**零极点模型**是分子分母分别为零点、极点形式的传递函数模型；

**最小实现**是将传递函数中彼此相等的零极点对消后得到的传递函数模型。

1. **求电机拖动系统速度输出和两个输入信号间的传递函数。**



**程序如下：**

|  |
| --- |
| clear,clc  syms c1 ka kr Ra T1 T2 c km kb c2 ;  Ra=1;T1=1;T2=1;c=1;c1=1;c2=1;ka=1;kr=1;km=1;kb=1;  nblocks = 12; %12个环节  n1=1;d1=1;  n2=c1;d2=1;  n3=ka;d3=1;  n4=kr;d4=1;  n5=1/Ra;d5=[T1 1];  n6=km;d6=1;  n7=1/c;d7=[T2 1];  n8=1;d8=[1 0];  n9=-kb;d9=1;  n10=-c2;d10=1;  n11=1;d11=1;  n12=-1;d12=1;  blkbuild;%构建a,b,c,d  Q=[1 0 0; 2 1 0;3 2 10 ;4 3 0;5 4 9;6 5 0;7 6 12;8 7 0;9 7 0;10 8 0; 11 8 0;12 0 0];  INPUTS=[1,12];%1,12为输入  OUTPUTS=11; %11为输出  [A,B,C,D]=connect(a,b,c,d,Q,INPUTS,OUTPUTS)  [num,den]=ss2tf(A,B,C,D,1);  G1=tf(num,den)  [num,den]=ss2tf(A,B,C,D,2);  G2=tf(num,den) |

**程序输出：**

|  |
| --- |
| State model [a,b,c,d] of the block diagram has 12 inputs and 12 outputs.  A =  -1 -1 -1  1 -1 0  0 1 0  B =  1 0  0 -1  0 0  C =  0 0 1  D =  0 0  G1 =  1  ---------------------  s^3 + 2 s^2 + 2 s + 1  Continuous-time transfer function.  G2 =  -s - 1  ---------------------  s^3 + 2 s^2 + 2 s + 1  Continuous-time transfer function. |

**实验三 控制系统的仿真方法**

1. **分别用欧拉法和四阶龙格-库塔法求微分方程的数值解，并与解析解比较， ，。**

****

1. 利用**欧拉法**求解微分方程的数值

**程序如下：**

|  |
| --- |
| clear,clc  y1(1)=1/3;  h=0.1;%步长h=0.1  k=1;  for t=h:h:1.5  k=k+1;  y1(k)=y1(k-1)+(-30)\*y1(k-1)\*h;  end  x=0:h:1.5;  plot(x,y1,'k') |

**程序输出：**

当h=0.1，0.075，0.05时，输出图像依次为：







1. 利用**四阶龙格-库塔法**求解微分方程的数值

**程序如下：**

|  |
| --- |
| clear,clc  f=inline('-30\*y','t','y'); %微分方程表达式  h=0.1; %步长h  n=1+1.5/h; %t点的个数  y(1)=1/3;t(1)=0;  for i=2:n  k1=f(t(i-1),y(i-1));  k2=f(t(i-1)+h/2,y(i-1)+k1\*h/2);  k3=f(t(i-1)+h/2,y(i-1)+k2\*h/2);  k4=f(t(i-1)+h,y(i-1)+k3\*h);  y(i)=y(i-1)+h\*(k1+2\*k2+2\*k3+k4)/6;  t(i)=t(i-1)+h;  end  plot(t,y) |

**程序输出：**

当h=0.1，0.075，0.05时，输出图像依次为：







1. 利用**解析法**求微分方程：

**程序如下：**

|  |
| --- |
| t=0:0.00001:1.5;  y=exp(-30\*t)/3;%解析表达式  plot(t,y) |

**程序输出：**



**说明：**

由图像分析可得，龙格库塔法和欧拉法的仿真结果均不理想，与真实情况有较大的出入；但是龙格库塔法精度更高，在步长较小时，具有一定的精度，更加接近真实值。

1. **利用input() 函数修改例3-2的程序，使其变为连续系统面向结构图的通用仿真程序。**

**程序如下：**

|  |
| --- |
| r = input('输入=');  P = input('传递函数p=');  W = input('W=');  W0= input('W0=');  Wc= input('Wc=');  Tf=input('输入仿真时间Tf：');  h=input('输入计算步长h：');  A1=diag(P(:,1));  B1=diag(P(:,2));  C1=diag(P(:,3));  D1=diag(P(:,4));  H=B1-D1\*W;  Q=C1\*W-A1;  A=inv(H)\*Q;  B=inv(H)\*C1\*W0;  x=[zeros(length(A),1)];  y=[zeros(length(Wc(:,1)),1)];  t=0;  for i=1:Tf/h  k1=A\*x+B\*r;  k2=A\*(x+h\*k1/2)+B\*r;  k3=A\*(x+h\*k2/2)+B\*r;  k4=A\*(x+h\*k3)+B\*r;  x=x+h\*(k1+2\*k2+2\*k3+k4)/6;  y=[y,Wc\*x];  t=[t,t(i)+h];  end  plot(t,y) |

**程序输出：**

|  |
| --- |
| 输入r=10  传递函数p=[0.1 1 0.5 1;0 1 1 0;2 1 2 0;10 1 10 0]  W=[0 0 0 -1;1 0 0 0;0 1 0 0;0 0 1 0]  W0=[1;0;0;0]  Wc=[0 0 0 1]  输入仿真时间Tf：15  输入计算步长h：0.1 |



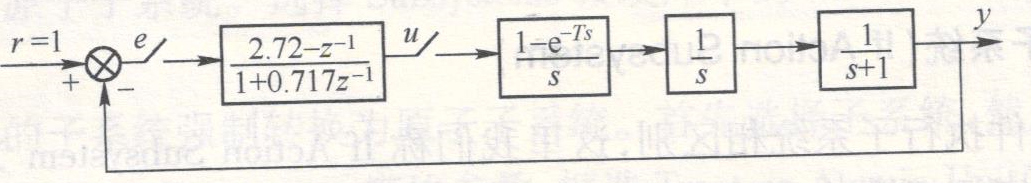
图3.2仿真曲线（Tf=15,h=0.1）

1. **编写滞环非线性特性的M函数文件。**

**程序如下：**

|  |
| --- |
| function [x,u1] = backlash1(u1,u,x1,s) %滞环非线性函数  if (u>u1) %u1和u分别为k-1和k时刻的输入量  if((u-s)>=x1)  x = u-s;  else  x = x1; %x1和x分别为k-1和k时刻的输出量  end  else if(u<u1)  if((u+s)<=x1)  x = u+s;  else  x=x1;  end  else  x=x1;  end  end  u1=u; |

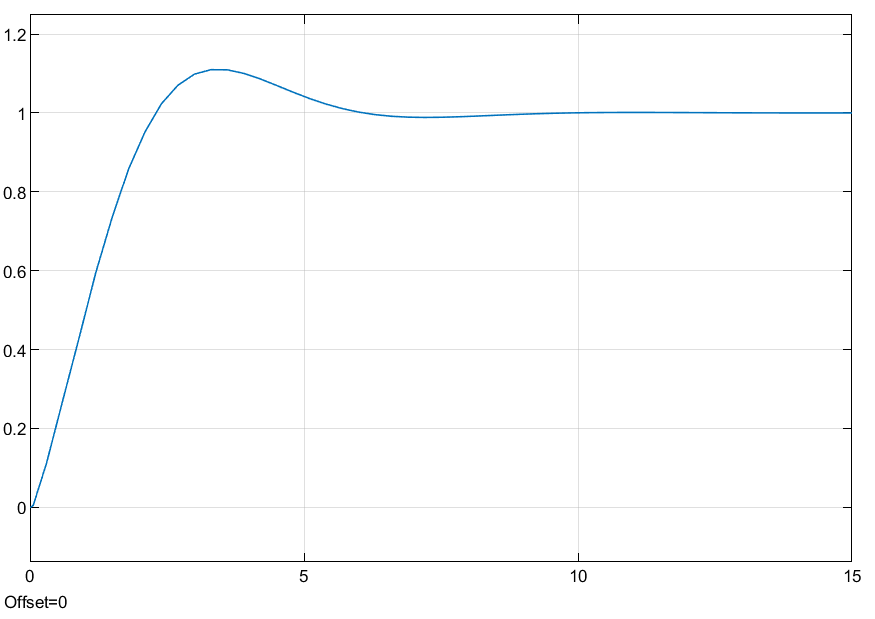
1. **用Simulink求采样控制的单位阶跃响应。**



**Simulink系统控制框图如下：**



**仿真输出的单位阶跃响应图像如下：**



**实验四 控制系统的分析与设计**

1. **分别用M文件和Simulink求微分方程的解曲线和相平面曲线。**

1. **M文件的求取**

**程序如下：**

|  |
| --- |
| clear,clc  [t,x]=ode45(@odefun,[0,20],[1;0]);  plot(t,x(:,1),t,x(:,2));%微分方程的解曲线  legend('x','x的一阶导')  plot(x(:,1),x(:,2));%微分方程的相平面曲线  legend('x','x(t)')  function dx=odefun(t,x)  dx=[x(2);(1-x(1)^2)\*x(2)-x(1)];  end |

**程序输出：**



图4.1.1微分方程的解曲线



图4.1.2微分方程的相平面曲线

**（2）Simulink的仿真**



图4.1.3系统Simulink的仿真框图

**仿真结果**（与M文件所得结果近似相同）：

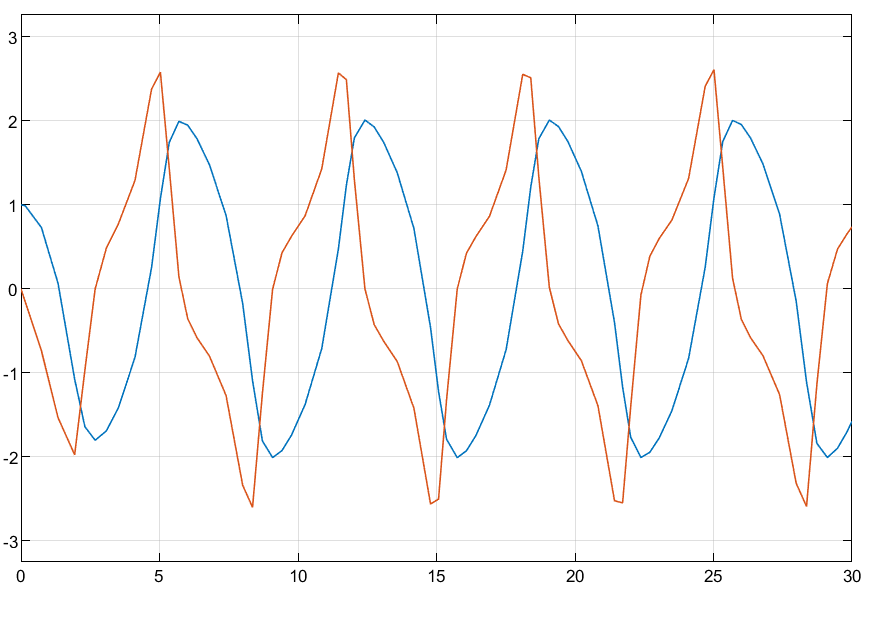
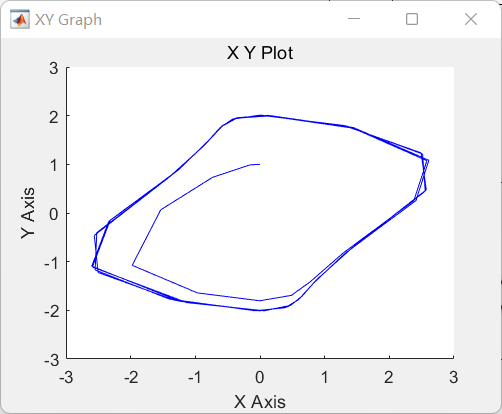
 

图4.1.4解曲线输出图像 图4.1.5相平面输出图像

1. **已知单位负反馈系统的开环传递函数**



**绘制系统的根轨迹和伯德图，并求时系统的相位裕量和幅值裕量。**

**程序如下：**

|  |
| --- |
| clear,clc  num1=[16]; %开环传递函数分子多项式洗漱，假定K=16  den=conv(conv([1,0],[1,1]),[1,10]); %开环传递函数分母多项式  sys1=tf(num1,den); %系统传递函数模型  rlocus(sys1); %绘制系统的根轨迹  axis([-15 5 -15 15]); %设置坐标范围：实轴 [-8,2]，虚轴[-6,6]  [Gm,Pm,Wcg,Wcp] = margin(sys1) %Gm幅值裕度，Pm相角裕度  margin(sys);%绘制bode图，并标出裕度值 |

**程序输出：**

|  |
| --- |
| Gm =  6.8750  Pm =  36.6174 |

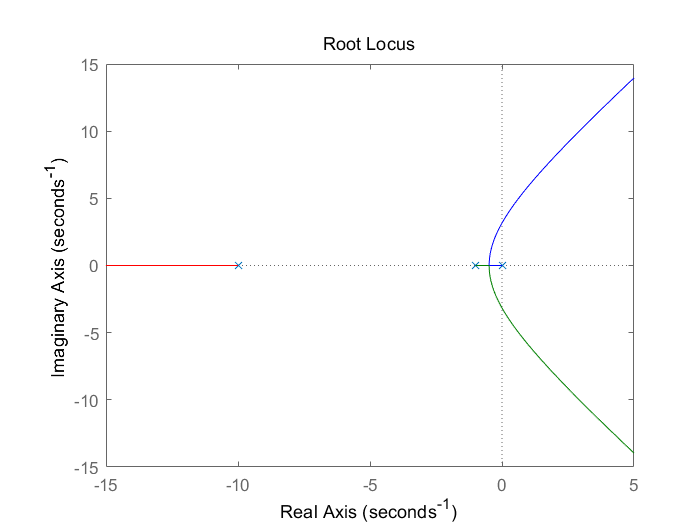


图4.2.1系统的根轨迹图

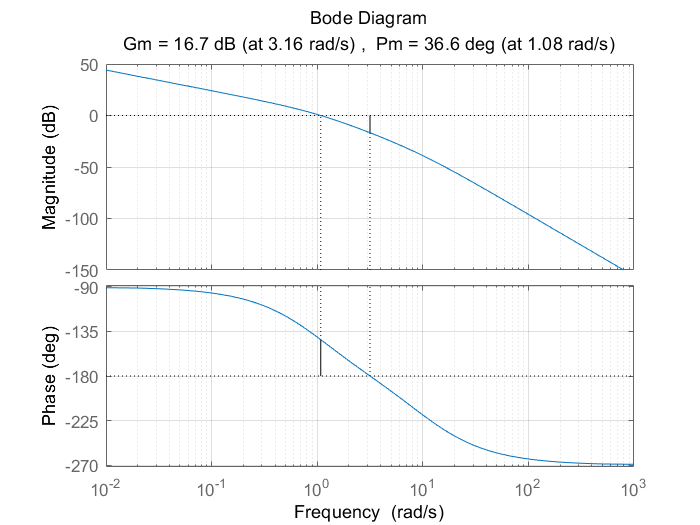


图4.2.2 系统的Bode图

1. **已知单位负反馈系统的开环传递函数 ，串联校正装置 ，求校正前后系统的性能指标对比。**

**程序如下：**

|  |
| --- |
| %程序一：校正前后系统bode图比较  clear,clc  num11=[4];  den11 =conv([1 0],[1 0.5]);  sys1=tf(num11,den11);%校正前系统传函  num21=[25 2.5]; %串联校正装置后传函的分子系数  bode(sys1);  hold on  den21=conv(conv([1 5],[1 0.01]),[1 0]);%校正后装置传函分母系数  sys2=tf(num21,den21);%校正后系统传函  bode(sys2);  legend('校正前','校正后')  %程序二：校正前后系统阶跃响应曲线图比较  clear,clc  num11=[4];  den11 =conv([1 0],[1 0.5]);  %校正后幅值相角幅值裕度和相角裕度  [Gm,Pm,Wcg,Wcp]=margin(num11,den11);  [num12,den12]=cloop(num11,den11);%校正前闭环传递函数  step(num12,den12)%校正前阶跃响应曲线  hold on  num21=[25 2.5]; %串联校正装置后传函的分子的系数  den21=conv(conv([1 5],[1 0.01]),[1 0]); %校正后装置传函的分母系数  %校正后幅值相角幅值裕度和相角裕度  [Gm1,Pm1,Wcg1,Wcp1]=margin(num21,den21)  [num,den]=cloop(num21,den21);%校正后闭环传递函数  step(num,den)%校正后阶跃响应曲线  legend('校正前','校正后') |

**校正前后性能比较：**



图4.3.1校正前后Bode图对比

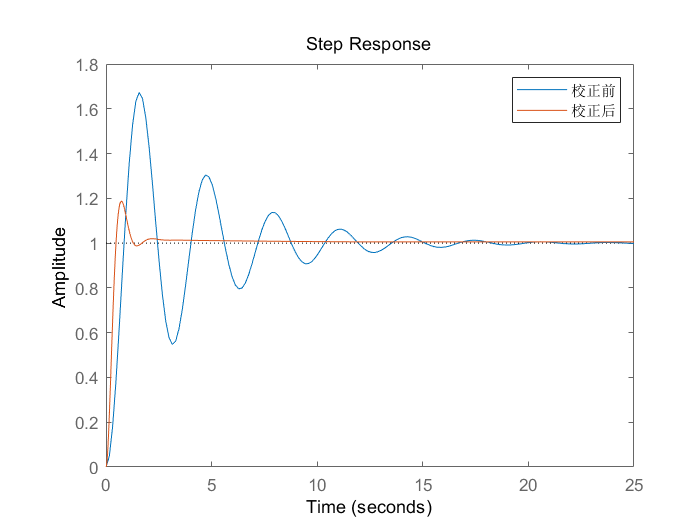


图4.3.2校正前后系统阶跃响应对比

**具体性能指标比较：**

**校正前：**

|  |
| --- |
| Gm =  Inf(表示幅值裕度无穷大)  Pm =  14.2485 |

**校正后：**

|  |
| --- |
| Gm1 =  Inf(表示幅值裕度无穷大)  Pm1 =  50.5097 |

1. **已知开环系统**



**设计状态反馈，使闭环极点为；设状态不可量测，设计状态观测器使其闭环极点为 -8，-8。**

**程序如下：**

|  |
| --- |
| clear,clc  A = [0 1;20.6 0];  b = [0;1];  C = [1 0];  %判断能控性和能观性  disp('The Rank of Controllability Matrix');rc = rank(ctrb(A,b))  disp('The Rank of Obstrabilaty Matrix');r0 = rank(obsv(A,C))  P=[-1.8+2.4\*j -1.8+2.4\*j]; K=acker(A,b,P)  %设计状态观测器  A1 = A';b1 = C';C1 = b';  P1 =[-8 -8];  K1 = acker(A1,b1,P1);L=K1' |

**程序输出：**

|  |
| --- |
| The Rank of Controllability Matrix  rc =  2  The Rank of Obstrabilaty Matrix  r0 =  2  K =  18.0800 3.6000  L =  16.0000  84.6000 |