二级倒立摆系统分析和设计

基本信息

姓名：蒋梓轩　班级：人工智能2302　学号：2213212344

实验环境：Matlab R2024a

实验原理

实验背景

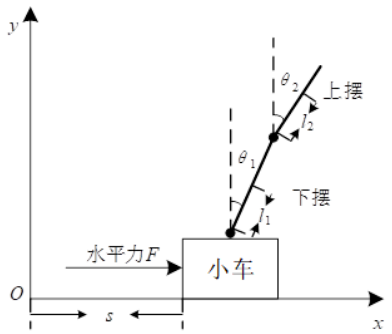
倒立摆系统是一种典型的**高阶次、不稳定、多变量**的非线性系统，因其控制原理在飞行器、机器人等高端装备中的广泛应用，长期以来受到国内外学者的高度关注。

倒立摆系统的控制方法在多个领域具有实际意义和广泛应用，例如：军事工业，航天技术，机器人领域等。因此，对该系统的建模、控制与稳定性分析具有重要的研究价值。

系统描述

对倒立摆系统可描述如下：给小车底座施加一个外力F，使小车停留在预定的位置，并使摆杆不倒下，即摆杆的摆动范围不超过某个预先设定的垂直偏离角。二级倒立摆系统由上摆、下摆和在导轨上运动的小车组成，如下图所示。

易见，**这是一个不稳定的系统**，若不加以控制，双摆将会立即倒下。为此需要由三个电位器分别将小车位置s、下摆角θ_1位移以及上摆角位移与下摆角位移之差$\theta_2-\theta_1$三路信号送入控制器。控制器发出适当的信号经功率放大器去控制伺服电机（产生外力F），通过传送皮带，电机将驱动小车在水平方向上不断进行运动以保持双摆不倒。



如上方示意图所示，s为小车偏离平衡位置的位移；l_1为下摆质心到轴心的距离；l_2为上摆质心至轴心的距离。设L_1为下摆轴心至上摆轴心距离；M_0为小车及驱动系统的等效质量；M_1为下摆质量；M_2为上摆质量；I_1为下摆转动惯量；I_2为上摆转动惯量；f_0为小车系统的摩擦系数；f_1为下摆转轴处的摩擦力矩系数；f_2为上摆转轴处的摩擦力矩系数；G_0为力与控制电压之比；g为当地重力加速度；U为直流放大器的输入电压。二级倒立摆的参数取值如下表所示：

符号	取值	符号	取值
$l_1(m)$	0.304	$I_1(kg\cdot m^2)$	0.04962
$l_2(m)$	0.226	$I_2(kg\cdot m^2)$	0.00482
$L_1(m)$	0.49	$f_0(N\cdot m\cdot s)$	22.947
$M_0(kg)$	1.32822	$f_1(N\cdot m\cdot s)$	0.007056
$M_1(kg)$	0.22	$f_2(N\cdot m\cdot s)$	0.002646
$M_2(kg)$	0.187	$G_0(N/V)$	11.88
$g(m/s)$	9.8		

系统建模

利用能量相关计算，拉格朗日方程，并在平衡点进行如下近似：

$$cos(\theta)\approx 1, sin(\theta)\approx \theta$$

可以得到如下结果：

$$\mathbf{M}(0,0)\begin{bmatrix}\ddot{s}\\ \dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}=\mathbf{F}(0,0,0,0)\begin{bmatrix}\dot{s}\\ \dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}+\mathbf{N}(0,0)\begin{bmatrix}s\\ \theta_1\\ \theta_2\end{bmatrix}+GU \tag{1}$$

其中：

$$\mathbf{M}(0,0)=\begin{bmatrix}M_0+M_1+M_2 & M_2L_1+M_1l_1 & M_2l_2\\ M_2L_1+M_1l_1 & I_1+M_1l_1^2+M_2L_1^2 & M_1L_1l_2\\ M_2l_2 & M_1L_1l_2 & I_2+M_2l_2^2\end{bmatrix} \tag{2}$$

$$\mathbf{F}(0,0,0,0)=\begin{bmatrix}-f_0 & 0 & 0\\ 0 & -(f_1+f_2) & f_2\\ 0 & f_2 & -f_2\end{bmatrix} \tag{3}$$

$$\mathbf{N}(0,0)=\begin{bmatrix}0 & 0 & 0\\ 0 & (M_1l_1+M_2L_1)g & 0\\ 0 & 0 & M_2l_2g\end{bmatrix} \tag{4}$$

$$\mathbf{G}=\begin{bmatrix}G_0\\ 0\\ 0\end{bmatrix} \tag{5}$$

经计算，$\mathbf{M}(0,0)$是可逆矩阵，故公式（1）可写为：

$$\begin{bmatrix} \ddot{s} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = [\mathbf{M(0,0)}]^{-1}\mathbf{F}(0,0,0,0) \begin{bmatrix} \dot{s} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + [\mathbf{M(0,0)}]^{-1}\mathbf{N}(0,0) \begin{bmatrix} s \\ \theta_1 \\ \theta_2 \end{bmatrix} + [\mathbf{M(0,0)}]^{-1}\mathbf{GU}$$

实验要求

要求如下：

（1）定义状态状态向量$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6]^T = [s, \theta_1, \theta_2 - \theta_1, \dot{s}, \dot{\theta}_1, \dot{\theta}_2 - \dot{\theta}_1]^T$，输出向量$\mathbf{y} = [x_1, x_2, x_3]^T = [s, \theta_1, \theta_2 - \theta_1]^T$，建立状态空间模型；

（2）判别系统的稳定性（通过系统矩阵的特征多项式来判定）；

（3）判别系统的能控性和能观性；

（4）当系统状态的初态为$\mathbf{x}(0) = [x_1, x_2, x_3, x_4, x_5, x_6]^T = [0.1, 0, 0.15, 0, 0, 0]^T$，输入$U = 0$时，求系统的状态响应$\mathbf{x}(t), t \in [0, 1]$，并画出状态响应曲线。

（5）构造一个状态反馈控制器，使系统在平衡点渐近稳定，即在状态反馈控制作用下使两个摆杆处于垂直状态。利用*Simulink*对系统的动态过程进行仿真，给出系统的仿真结构图；画出控制量和闭环系统状态变量随时间变化的曲线。

（6）分别设计一个全维观测器和降维观测器，要求观测器极点配置为离虚轴的距离是闭环极点的2-3倍。

（7）结合（5）和（6），利用*Simulink*对带有状态观测器（可以选择（6）中的一种）的状态反馈控制系统的动态过程进行仿真，给出系统的仿真结构图；画出控制量和闭环系统状态变量随时间变化的曲线。

实验内容&结果

一、状态空间模型的建立（Task 1）

根据给定的状态向量与输出向量定义，引入坐标变换矩阵 ($\mathbf{T}$)，将原始状态变量与给定状态向量关联起来，即：

$$\hat{\mathbf{x}} = \mathbf{T}\mathbf{x}$$

基于系统动力学特性，构造系统矩阵、输入矩阵、输出矩阵以及传递函数矩阵：

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ [M(0,0)]^{-1}N(0,0) & [M(0,0)]^{-1}F(0,0,0,0) \end{bmatrix},$$
$$\hat{\mathbf{B}} = \begin{bmatrix} \mathbf{0}_{3\times1} \\ [M(0,0)]^{-1}G \end{bmatrix}$$

利用坐标变换关系，求得最终的状态空间模型矩阵：

$$\mathbf{A} = \mathbf{T}^{-1}\hat{\mathbf{A}}\mathbf{T}, \quad \mathbf{B} = \mathbf{T}^{-1}\hat{\mathbf{B}}, \quad \mathbf{C} = [\mathbf{I}_{3\times3} \quad \mathbf{0}_{3\times3}], \quad \mathbf{D} = \mathbf{0}_{3\times1}$$

具体数值结果如下：

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \\ 0 & -1.4740 & -0.2577 & -15.2134 & 0.0055 & -0.0004 \\ 0 & 13.3152 & -9.2419 & 17.9677 & -0.1025 & 0.0975 \\ 0 & -2.7338 & 54.4863 & -3.6891 & 0.2599 & -0.4456 \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 7.8762 \\ -9.3022 \\ 1.9099 \end{bmatrix},$$
$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$
$$\mathbf{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

二、系统稳定性的判别（Task 2）

为判断系统的稳定性，需计算系统矩阵 (\mathbf{A}) 的特征值。通过数值计算方法，求得特征值如下：

$$\text{eig_A} = [0,\ 7.1743,\ 3.3494,\ -15.3362,\ -3.2740,\ -7.6748]$$

分析可知，特征值中存在实部为正的值（如 (7.1743) 和 (3.3494)），因此根据稳定性判据，该系统在无控制作用下不是渐近稳定的。

三、系统能控性和能观性的判别（Task 3）

构建能控性矩阵和能观性矩阵：

$$\mathbf{Qc} = [\mathbf{B},\ \mathbf{AB},\ \mathbf{A}^2\mathbf{B},\ \mathbf{A}^3\mathbf{B},\ \mathbf{A}^4\mathbf{B},\ \mathbf{A}^5\mathbf{B}], \quad \mathbf{Qo} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \mathbf{CA}^4 \\ \mathbf{CA}^5 \end{bmatrix}$$

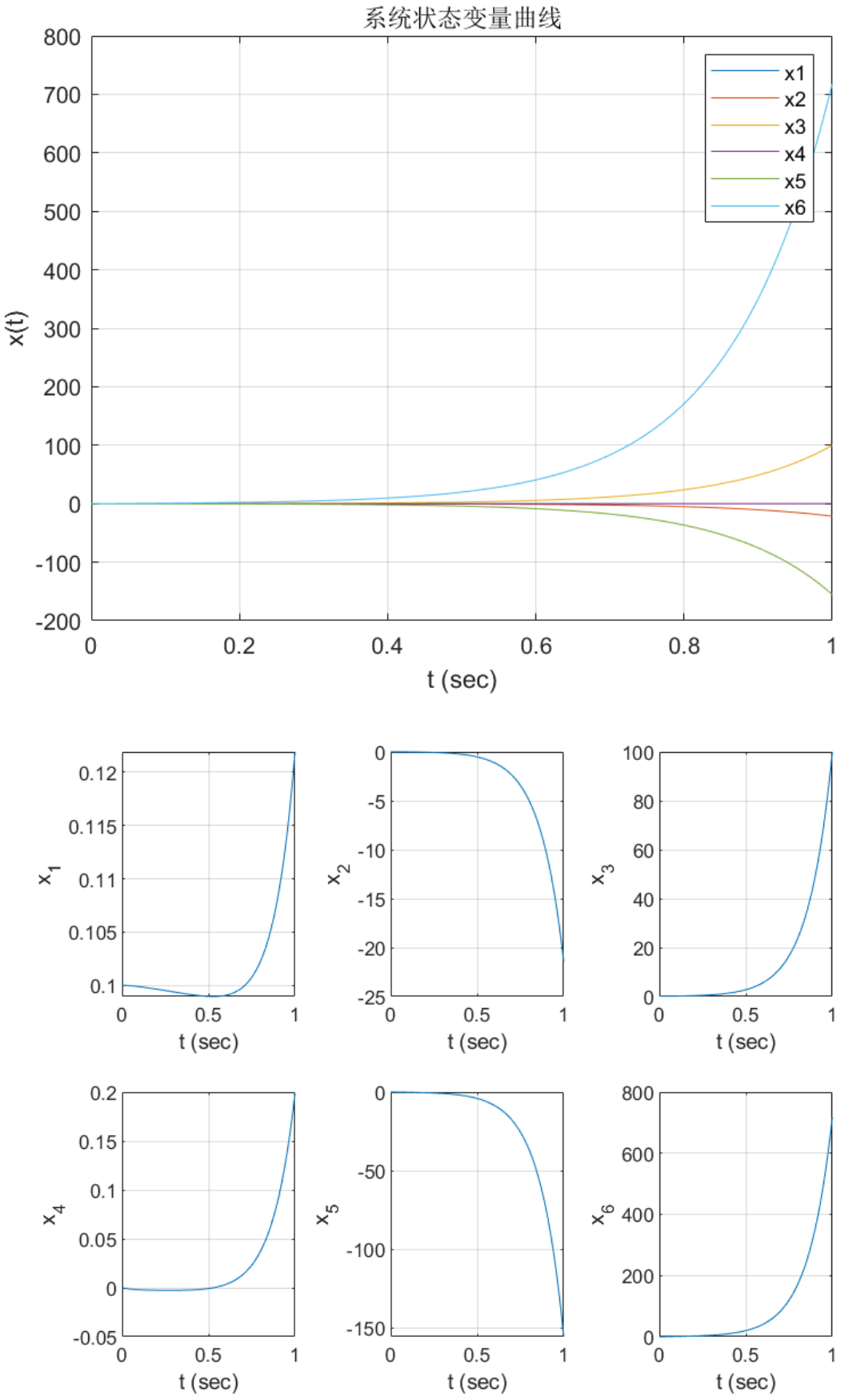
通过计算矩阵的秩，发现 $(\mathrm{rank}(\mathbf{Qc}) = 6)$ 且 $(\mathrm{rank}(\mathbf{Qo}) = 6)$，说明系统是完全能控且完全能观的。这为后续设计状态反馈控制器和观测器提供了基础。

四、系统状态响应的求解与分析（Task 4）

在给定初始状态：

$$\mathbf{x}(0) = [0.1,\ 0,\ 0.15,\ 0,\ 0,\ 0]^T$$

以及零输入（$\mathrm{U} = 0$）的条件下，求解状态方程，得到系统在时间区间 $(t \in [0, 1])$ 内的状态响应曲线，结果如下图所示：



从图中可以看出，在无控制输入时，系统状态呈现发散趋势，进一步验证了系统不稳定性的结论。

五、状态反馈控制器的设计与仿真（Task 5）

1. 控制器设计

为使系统在平衡点渐近稳定，采用极点配置法设计状态反馈控制器。给定期望的闭环极点为：

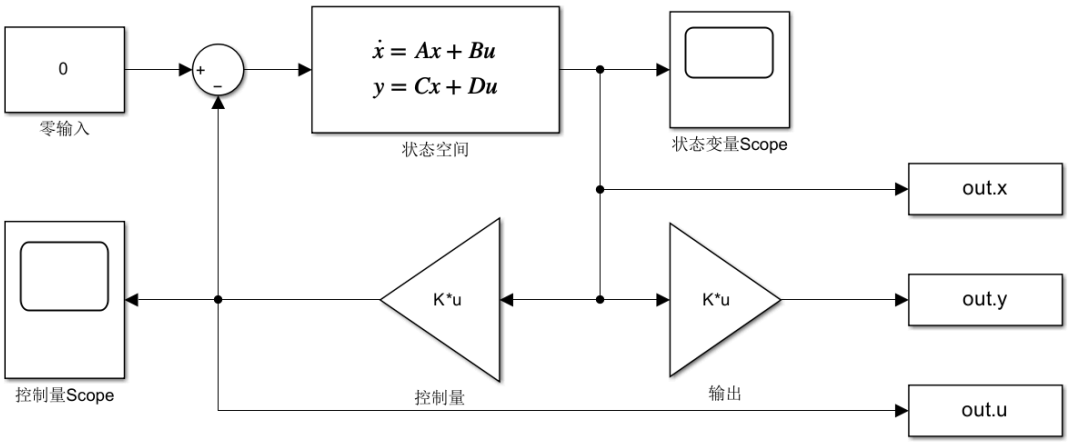
$$P = [-1,\ -1+1i,\ -1-1i,\ -2,\ -2+1i,\ -2-1i]$$

通过调用极点配置函数，计算得到状态反馈增益矩阵：

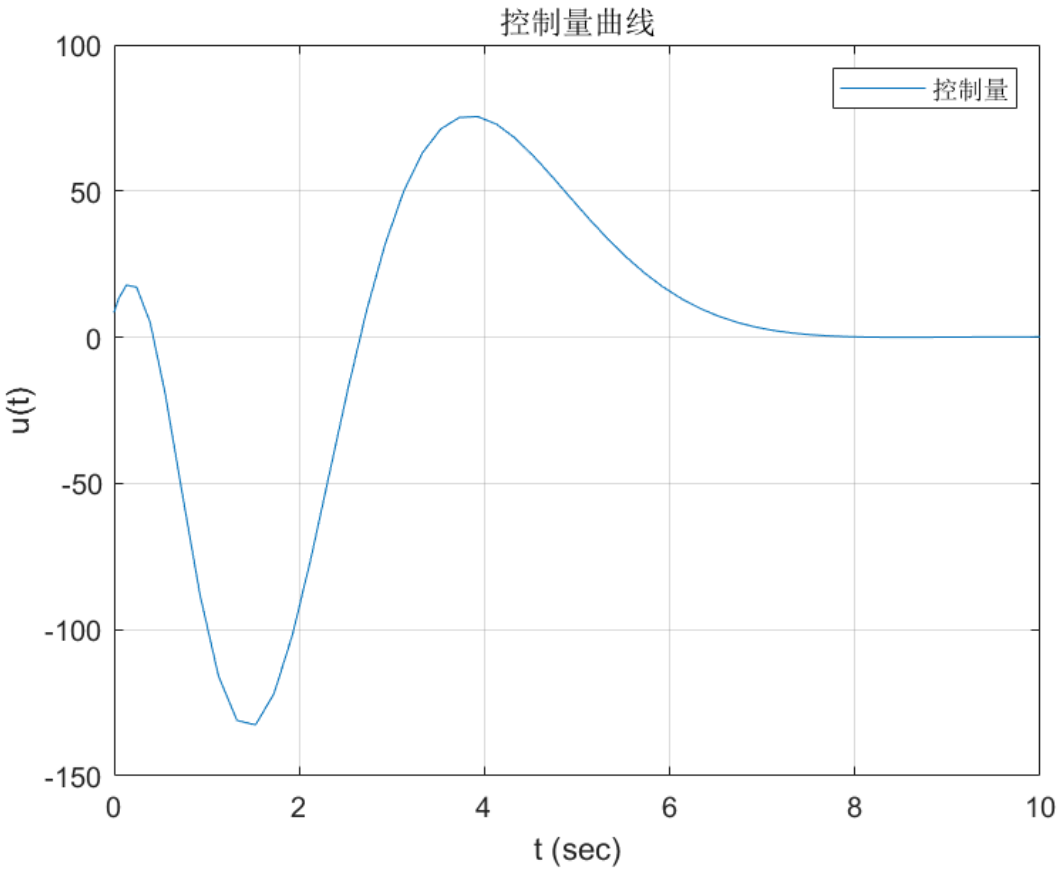
$$K = [0.0042,\ -1.2331,\ 56.1819,\ -1.9177,\ 0.4313,\ 6.4690]$$

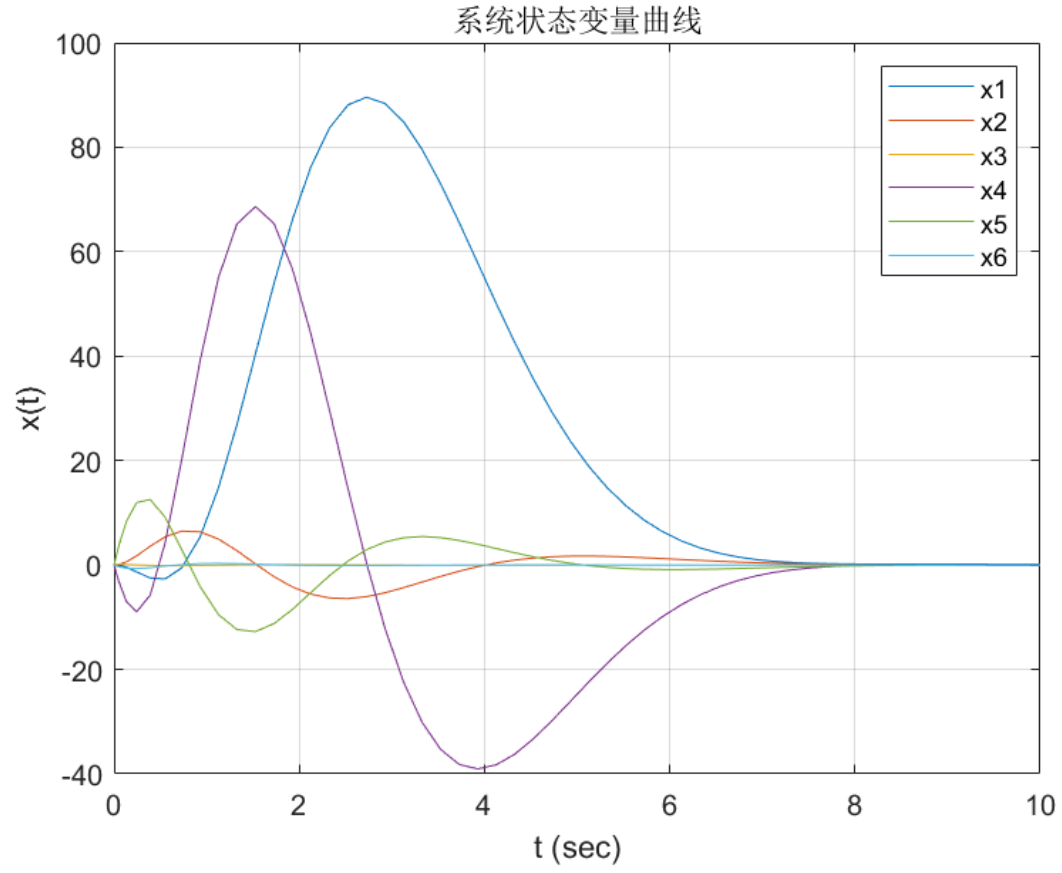
2. 仿真验证

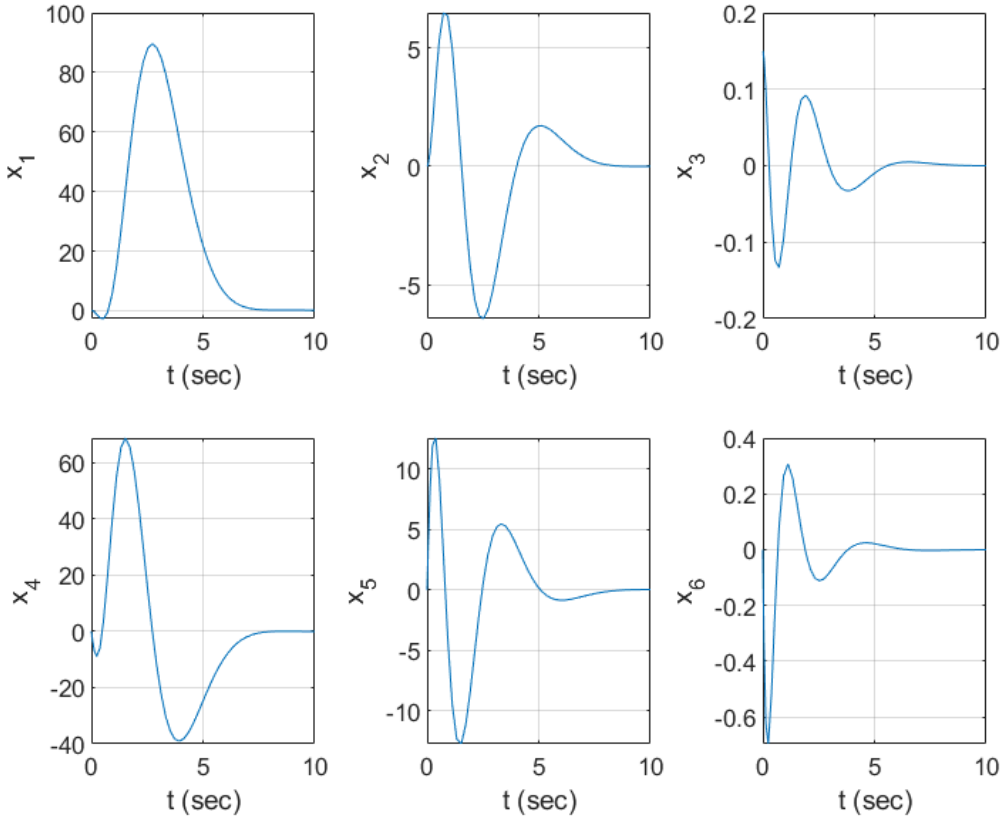
利用 Simulink 搭建包含状态反馈控制器的系统仿真模型，结构如下图所示：



对系统进行 10 秒时长的仿真，得到控制量和状态变量随时间变化的曲线：


控制量曲线
控制量
u(t)
t (sec)
100
50
0
-50
-100
-150
0
2
4
6
8
10


系统状态变量曲线
x1
x2
x3
x4
x5
x6
x(t)
t (sec)
100
80
60
40
20
0
-20
-40
0
2
4
6
8
10



从仿真结果可以看出，加入状态反馈控制器后，系统状态逐渐收敛至平衡点，表明控制器设计有效，系统实现了渐近稳定。

六、状态观测器的设计（Task 6）

1. 全维状态观测器

鉴于系统能观性，设计全维状态观测器。给定观测器极点为：

$$P1 = [-4 + 2i,\ -4 - 2i,\ -5 + 1i,\ -5 - 1i,\ -6 + 1i,\ -6 - 1i]$$

通过计算，得到观测器增益矩阵 ($\mathbf{M_1}$)，以及观测器系统矩阵 ($\mathbf{A_1}$)：

$$\mathbf{A_1} = \begin{bmatrix} 0.3829 & -2.4103 & 0.6400 & 1.0000 & 0 & 0 \\ -9.4031 & -5.4760 & -2.2825 & 0 & 1.0000 & 0 \\ 1.9906 & -0.7225 & -9.1455 & 0 & 0 & 1.0000 \\ -58.1696 & 23.7011 & -7.4865 & -15.2134 & 0.0055 & -0.0004 \\ 55.0354 & -45.8967 & 0.2943 & 17.9677 & -0.1025 & 0.0975 \\ -11.8771 & 6.4160 & -23.3607 & -3.6891 & 0.2599 & -0.4456 \end{bmatrix},$$

$$\mathbf{M_1} = \begin{bmatrix} -0.3829 & 2.4103 & -0.6400 \\ 9.4031 & 5.4760 & 2.2825 \\ -1.9906 & 0.7225 & 9.1455 \\ 58.1696 & -25.1751 & 7.2287 \\ -55.0354 & 59.2119 & -9.5362 \\ 11.8771 & -9.1498 & 77.8470 \end{bmatrix}$$

观测器方程为：

$$\dot{\hat{x}} = \mathbf{A_1}\hat{x} + \mathbf{B}u + \mathbf{M_1}(y - \mathbf{C}\hat{x})$$

2. 降维状态观测器

考虑到系统输出矩阵 ($\mathbf{C}$) 的秩为 3，设计维度为 3 的降维状态观测器。给定观测器极点为：

$$P2 = [-5 + 1i,\ -5 - 1i,\ -6]$$

进行降维状态观测器设计的流程代码如下：

```
r = length(A)-rank(C);
P2 = [-5+1i, -5-1i, -6];
disp('rank(C)')
disp(rank(C))
R = [zeros(3, 3), eye(3)];
T_inv = [R;C];% Q
T = inv(T_inv);% P
[A2, B2, C2, d2] = ss2ss(A, B, C, D, T_inv);
A11 = A2(1:rank(C), 1:rank(C));
A12 = A2(1:rank(C), rank(C)+1:6);
A21 = A2(rank(C)+1:6, 1:rank(C));
A22 = A2(rank(C)+1:6, rank(C)+1:6);
B11 = B2(1:rank(C), :);
B21 = B2(rank(C)+1:6, :);
K2 = place(A11', A21', P2);
M2 = K2';
A3 = A11-M2*A21;
U3 = B11-M2*B21;
Y3 = A12-M2*A22 + (A11 - M2*A21)*M2;
```

如上，求得降维观测器的系统矩阵 ($\mathbf{A_3}$)、输入矩阵 ($\mathbf{U_3}$) 和输出矩阵 ($\mathbf{Y_3}$)：

$$\mathbf{A_3} = \begin{bmatrix} -5 & 1 & 0 \\ -1 & -5 & 0 \\ 0 & 0 & -6 \end{bmatrix}$$

$$\mathbf{U_3} = \begin{bmatrix} 7.8762 \\ -9.3022 \\ 1.9099 \end{bmatrix},$$

$$\mathbf{Y_3} = \begin{bmatrix} 70.0345 & 8.3959 & -0.1581 \\ -84.6254 & -10.1780 & -9.7288 \\ 22.1343 & -4.2933 & 21.1596 \end{bmatrix}$$

观测器方程为：

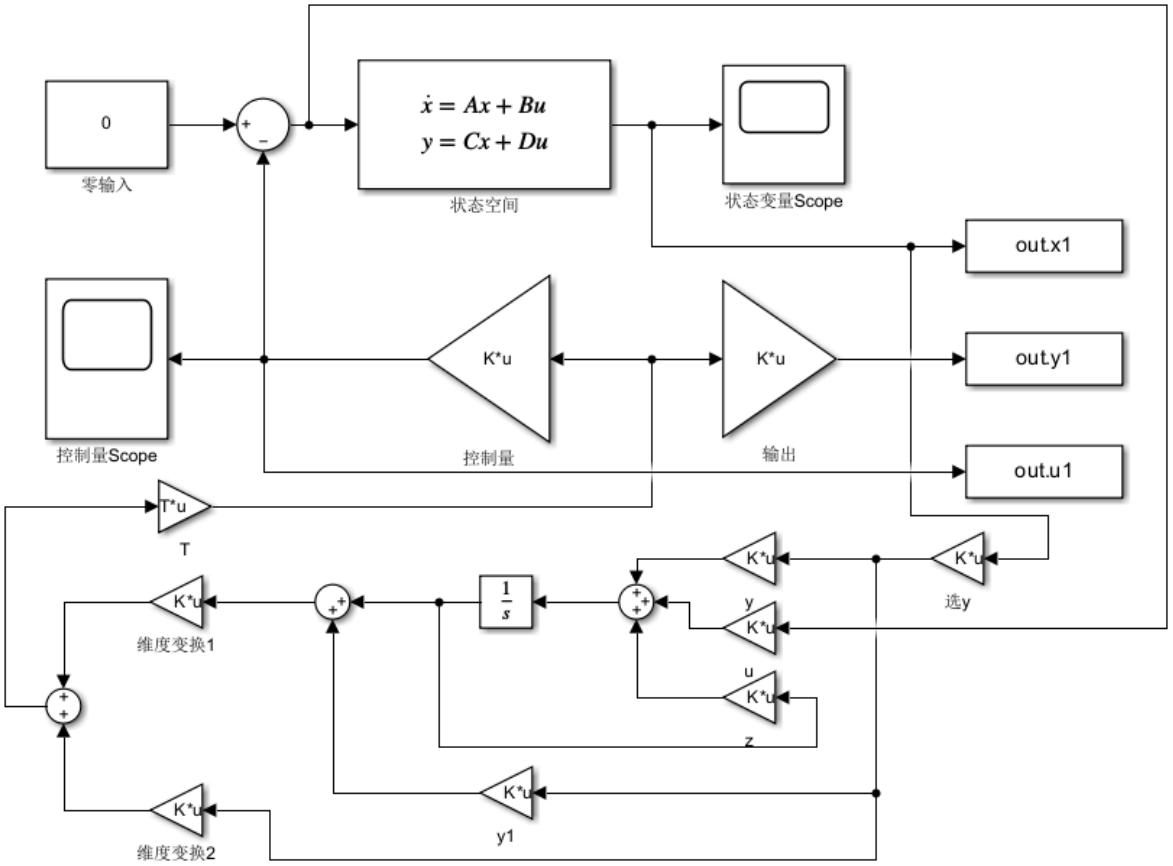
$$\begin{aligned} \dot{\mathbf{z}} &= (\overline{\mathbf{A}_{11}} - \mathbf{M}\overline{\mathbf{A}_{21}})\mathbf{z} + (\overline{\mathbf{B}_1} - \mathbf{M}\overline{\mathbf{B}_2})\mathbf{u} + \\ &[(\overline{\mathbf{A}_{12}} - \mathbf{M}\overline{\mathbf{A}_{22}}) + (\overline{\mathbf{A}_{11}} - \mathbf{M}\overline{\mathbf{A}_{21}})\mathbf{M}]\mathbf{y} \\ &= \mathbf{A}_3\mathbf{z} + \mathbf{U}_3\mathbf{u} + \mathbf{Y}_3\mathbf{y} \\ \hat{\mathbf{x}} &= \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} \begin{bmatrix} \mathbf{z} + \mathbf{M}\mathbf{y} \\ \mathbf{y} \end{bmatrix} \end{aligned}$$

其中，$\mathbf{M}$(代码中的$\mathbf{M}_2$) 为降维观测器增益矩阵。

七、带状态观测器的控制系统仿真（Task 7）

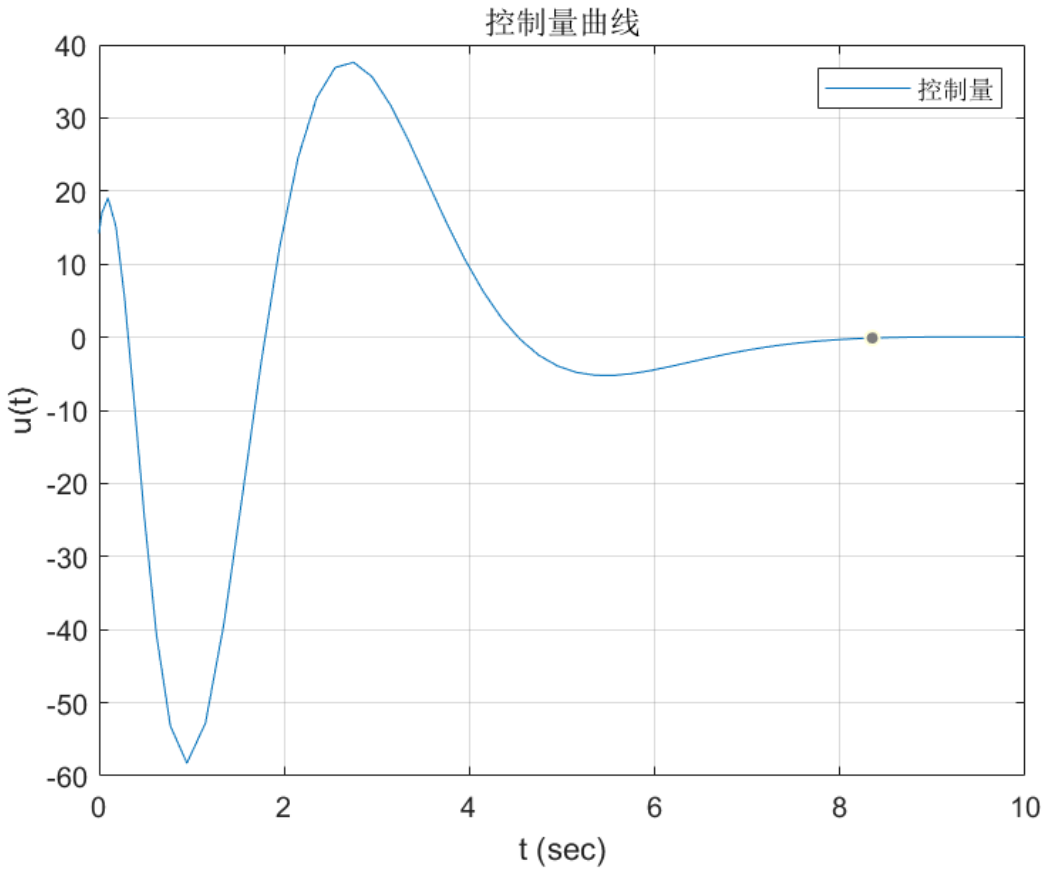
1. 仿真模型搭建

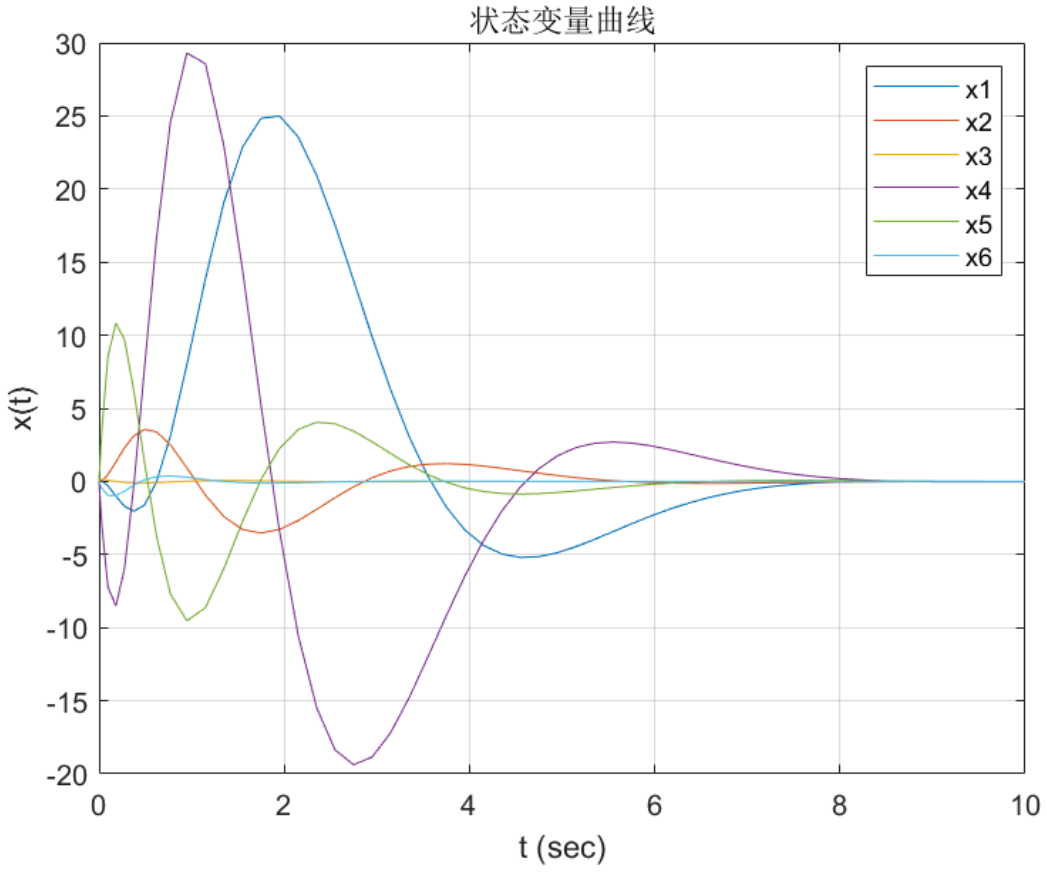
结合任务五和任务六的设计结果，在 Simulink 中构建带状态反馈和降维状态观测器的控制系统仿真模型，结构如下图所示：

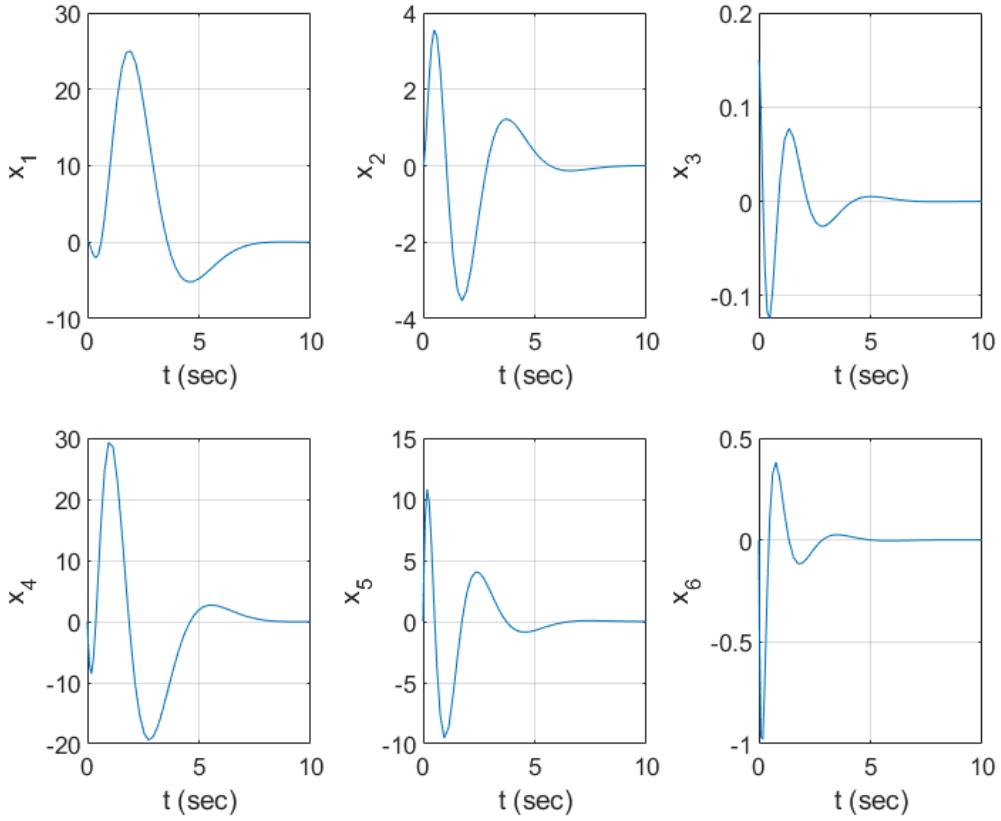


2. 仿真结果与分析

对系统进行 10 秒时长的仿真，得到控制量和状态变量随时间变化的曲线：


控制量曲线
控制量
u(t)
t (sec)
40
30
20
10
0
-10
-20
-30
-40
-50
-60
0
2
4
6
8
10


状态变量曲线
x1
x2
x3
x4
x5
x6
x(t)
t (sec)
30
25
20
15
10
5
0
-5
-10
-15
-20
0
2
4
6
8
10



仿真结果显示，带状态观测器的控制系统能够有效实现状态估计，并通过状态反馈使系统状态收敛至平衡点。控制量曲线表明控制器输出在系统稳定过程中起到了关键作用，验证了所设计观测器和控制器的有效性。

实验总结

本次实验围绕二级倒立摆系统的建立、分析以及控制设计展开，主要完成了以下工作：

1. 成功建立了系统的状态空间模型，并对其稳定性、能控性和能观性进行了全面分析，为后续的控制设计奠定了基础。
2. 设计了状态反馈控制器，通过合理的极点配置实现了对系统稳定性的有效控制，并通过仿真验证了控制器的性能。
3. 分别设计了全维和降维状态观测器，能够准确估计系统状态，解决了系统状态无法直接测量的问题，其中降维观测器的设计在保证性能的同时降低了观测器的维度，提高了设计效率。
4. 将状态反馈控制器与降维状态观测器相结合，构建了完整的控制系统，并通过仿真验证了整个系统的有效性和可行性。

整个实验过程涉及系统建模、稳定性分析、能控能观性判别、控制器设计以及观测器设计等多个关键环节，全面地运用了现代控制理论的基本方法和工具，加深了对控制系统设计与分析原理和流程的理解，同时也提升了运用相关理论知识解决实际控制问题的能力。

附录（代码）

```
clear;
clc;
diary log.txt

disp('Init')

l1 = 0.304;
l2 = 0.226;
L1 = 0.49;
M0 = 1.32822;
M1 = 0.22;
M2 = 0.187;
I1 = 0.04962;
I2 = 0.00482;
f0 = 22.947;
f1 = 0.007056;
f2 = 0.002646;
G0 = 11.88;
g = 9.8;

disp('Init done!')

%%
disp('Task 1')
M_0_0 = [M0 + M1 + M2, M2*L1 + M1*l1, M2*l2;
        M2*L1 + M1*l1, I1 + M1*l1^2 + M2*L1^2, M1*L1*l2;
        M2*l2, M1*L1*l2, I2 + M2*l2^2];

F_0_0_0_0 = [-f0, 0, 0;
             0, -(f1 + f2), f2;
             0, f2, -f2];

N_0_0 = [0, 0, 0;
        0, (M1*l1 + M2*L1)*g, 0;
        0, 0, M2*l2*g];

G = [G0;
    0;
    0];

M_0_0_inv = inv(M_0_0);

A = [zeros(3, 3), eye(3);
    M_0_0_inv*N_0_0, M_0_0_inv*F_0_0_0_0];
```

```
B = [zeros(3, 1);
    M_0_0_inv*G];

T = [1, 0, 0, 0, 0, 0;
    0, 1, 0, 0, 0, 0;
    0, 1, 1, 0, 0, 0;
    0, 0, 0, 1, 0, 0;
    0, 0, 0, 0, 1, 0;
    0, 0, 0, 0, 1, 1];

A = inv(T)*A*T;

B = inv(T)*B;

C = [eye(3), zeros(3, 3)];

D = zeros(3, 1);

disp("A: ");
disp(A);
disp("B: ");
disp(B);
disp("C: ");
disp(C);
disp("D: ");
disp(D);

sys = ss(A, B, C, D);
disp('状态空间模型')
disp(sys)

disp('Task 1 done!')
%%
disp('Task 2')

eig_A = eig(A);
is_stable = all(real(eig_A) < 0);

disp('eig_A:')
disp(eig_A)

if all(real(eig_A) < 0)
    disp("系统是渐近稳定的")
else
```

```
    disp("系统不是渐近稳定的")
end

disp('Task 2 done!')
%%
disp('Task 3')

Qc = [B, A*B, A^2*B, A^3*B, A^4*B, A^5*B];
Qo = [C; C*A; C*A^2; C*A^3; C*A^4; C*A^5];
rank_Qc = rank(Qc);
rank_Qo = rank(Qo);
disp('rank_Qc:')
disp(rank_Qc)
if rank_Qc == size(A, 1)
    disp("系统是能控的")
else
    disp("系统是不能控的")
end
disp('rank_Qo:')
disp(rank_Qo)
if rank_Qo == size(A, 1)
    disp("系统是能观的")
else
    disp("系统是不能观的")
end
disp('Task 3 done!')
%%
disp('Task 4')
x0 = transpose([0.1, 0, 0.15, 0, 0, 0]);
t = 0:0.005:1;
u=0*t;
[~, T, x] = lsim(sys, u, t, x0);

figure;
for i = 1:6
    subplot(2, 3, i);
    plot(T, x(:, i));
    xlabel('t (sec)');
    ylabel(['x_', num2str(i)]);
    grid on
end

figure;
plot(T, x);
legend('x1', 'x2', 'x3', 'x4', 'x5', 'x6');
```

```
xlabel('t (sec)');
ylabel('x(t)');
title('系统状态变量曲线');
grid on;

disp('Task 4 done!')
%%
disp('Task 5')
P = [-1, -1+1i, -1-1i, -2, -2+1i, -2-1i];
K = acker(A, B, P);
disp('K:')
disp(K)

% 运行Task5.slx

simout_x = out.x;
simout_y = out.y;
simout_u = out.u;
simout_t = out.tout;

figure;
for i = 1:6
    subplot(2, 3, i);
    plot(simout_x.Time, simout_x.Data(:, i));
    xlabel('t (sec)');
    ylabel(['x_', num2str(i)]);
    grid on
end

figure
plot(simout_x)
legend('x1', 'x2', 'x3', 'x4', 'x5', 'x6');
xlabel('t (sec)');
ylabel('x(t)');
title('系统状态变量曲线');
grid on

figure
plot(simout_u)
legend('控制量')
xlabel('t (sec)');
ylabel('u(t)');
title('控制量曲线');
grid on
```

```
disp('Task 5 done!')
%%
disp('Task 6')

disp("全维状态观测器")
if rank_Qo == size(A, 1)
    disp("系统是能观的")
else
    disp("系统是不能观的")
    return
end

P1 = [-4 + 2i, -4 - 2i, -5 + 1i, -5 - 1i, -6 + 1i, -6 - 1i];
K1 = place(A', C', P1);
M1 = K1';
A1 = A - M1*C;
disp('A')
disp(A1)
disp('M')
disp(M1)

disp("降维状态观测器")
r = length(A)-rank(C);
P2 = [-5+1i, -5-1i, -6];
disp('rank(C)')
disp(rank(C))
R = [zeros(3, 3), eye(3)];
T_inv = [R;C];% Q
T = inv(T_inv);% P
[A2, B2, C2, d2] = ss2ss(A, B, C, D, T_inv);
A11 = A2(1:rank(C), 1:rank(C));
A12 = A2(1:rank(C), rank(C)+1:6);
A21 = A2(rank(C)+1:6, 1:rank(C));
A22 = A2(rank(C)+1:6, rank(C)+1:6);
B11 = B2(1:rank(C), :);
B21 = B2(rank(C)+1:6, :);

K2 = place(A11', A21', P2);
M2 = K2';

A3 = A11-M2*A21;
U3 = B11-M2*B21;
Y3 = A12-M2*A22 + (A11 - M2*A21)*M2;
disp('A3')
```

```
disp(A3)
disp('U3')
disp(U3)
disp('Y3')
disp(Y3)

disp('Task 6 done!')

%%

disp('Task 7')

simout_x1 = out.x1;
simout_y1 = out.y1;
simout_u1 = out.u1;
simout_t1 = out.tout;

figure;
for i = 1:6
    subplot(2, 3, i);
    plot(simout_x1.Time, simout_x1.Data(:, i));
    xlabel('t (sec)');
    ylabel(['x_', num2str(i)]);
    grid on
end

figure
plot(simout_x1)
legend('x1', 'x2', 'x3', 'x4', 'x5', 'x6');
xlabel('t (sec)');
ylabel('x(t)');
title('状态变量曲线');
grid on

figure
plot(simout_u1)
legend('控制量')
xlabel('t (sec)');
ylabel('u(t)');
title('控制量曲线');
grid on


disp('Task 7 done!')
```