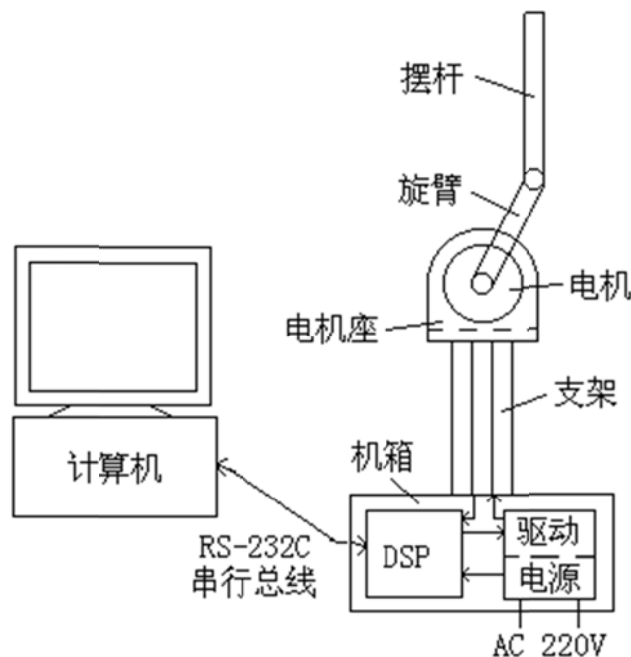


§9 旋转式倒立摆

XZ-II 型旋转式倒立摆系统，是一个典型的机电一体化系统，采用内置 DSP 作为控制器和力矩电机作为执行元件，使摆杆倒立。

9.1 系统组成

XZ-II 型旋转式倒立摆系统采用 DSP 作控制器，由直流力矩电机直接驱动，能够独立执行实时控制算法，脱离计算机直接运行；也可以通过 RS-232C 串行通讯接口用计算机控制，进行在线控制算法调试。它的 DSP 部分、电源与电机驱动部分全部安装在机箱内。下图为系统的总体结构示意图。

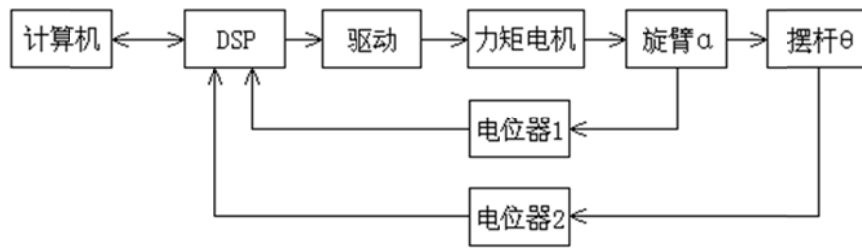


系统总体结构图

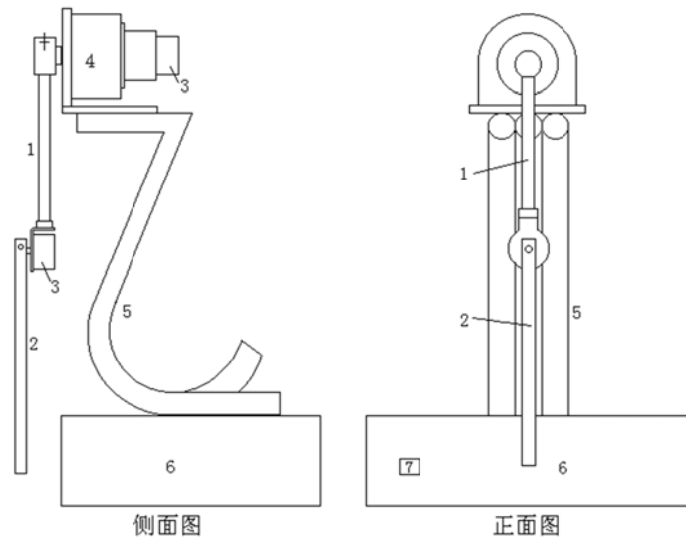
DSP 控制器是核心控制器件，它完成数据传送，A/D、D/A 转换，运算，数据处理等功能。系统的执行机构是直流力矩电机，由专门的驱动电路驱动，控制倒立摆的运动。系统的测量反馈元件是两个塑料电位计，用于反馈旋臂和摆杆的角度信号。

9.2 系统的工作原理

旋臂由转轴处的直流力矩电机驱动，可绕转轴在垂直于电机转轴的铅直平面内转动。旋臂和摆杆之间由电位器的活动转轴相连，摆杆可绕转轴在垂直于转轴的铅直平面内转动。由电位器测量得到的 2 个角位移信号（旋臂与铅直线的夹角，摆杆和旋臂之间的相对角度），作为系统的 2 个输出量被送入 DSP 控制器。由角位移的差分可得到角速度信号，然后根据一定的状态反馈控制算法，计算出控制律，并转化为电压信号提供给驱动电路，以驱动直流力矩电机的运动，通过电机带动旋臂的转动来控制摆杆的运动，工作原理如图所示。



系统结构框图



1. 旋臂 2. 摆杆 3. 电位器 4. 直流力矩电机 5. 支架 6. 机箱 7. 电源开关
机械结构图

§10 旋转式倒立摆的状态反馈控制器和状态观测器设计

一、实验名称：旋转式倒立摆的状态反馈控制器和状态观测器设计

二、实验目的：

1. 认识 XZ-II 旋转式倒立摆系统，掌握系统构成、工作原理、使用方法、注意事项和软件操作平台使用方法。

2. 掌握一阶旋转式倒立摆系统建立数学模型，并利用 Matlab 对其进行仿真研究。

3. 掌握一阶旋转式倒立摆系统的状态反馈控制（极点配置算法），在 MATLAB 平台上进行仿真研究。

4. 掌握实际系统的调试方法，对一阶旋转式倒立摆系统进行在线控制。

5. 掌握旋转式倒立摆状态观测器设计方法，并利用 Matlab 对其进行仿真研究。

6. 研究引入状态观测器对系统性能的影响

三、实验类型：综合设计

四、实验环境：计算机，XZ-II 型旋转式倒立摆

五、实验内容与实验步骤：

1. 机理法建模

系统建模和参数测量是控制算法设计的第一步，建立比较精确的数学模型是控制系统设计的基础。下面，用牛顿力学对模型进行一个简单的分析：

如图 7.1 所示，根据牛顿力学，在非惯性系 S2 中，对摆杆有：

$$J_2 \ddot{\theta}_2 + f_2 \dot{\theta}_2 = M_{12} + m_2 g L_2 \sin \theta_2 \quad (10.1)$$

其中， M_{12} 为旋臂对摆杆的作用力矩，

$$M_{12} = m_2 L_2 [R_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - R_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2)] \quad (10.2)$$

在惯性系 S2 中，对旋臂，有：

$$J_1 \ddot{\theta}_1 + f_1 \dot{\theta}_1 = M_0 + M_{21} + m_1 g L_1 \sin \theta_1 \quad (10.3)$$

其中， M_0 为电机输出转矩。

$$M_0 = K_m (u - K_e \dot{\theta}_1) \quad (10.4)$$

M_{21} 为摆杆对旋臂的作用力矩，利用反作用规律，有：

$$\begin{aligned} M_{21} &= m_2 \left[g + \frac{d^2}{dt^2} (R \cos \theta_1 + L_2 \cos \theta_2) \right] \cdot R \sin \theta_1 - m_2 \left[\frac{d^2}{dt^2} (R \sin \theta_1 + L_2 \sin \theta_2) \right] \cdot R \cos \theta_1 \\ &= m_2 g R \sin \theta_1 - m_2 R^2 \ddot{\theta}_1 - m_2 R L_2 \ddot{\theta}_2^2 \sin(\theta_1 - \theta_2) - m_2 R L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) \end{aligned} \quad (10.5)$$

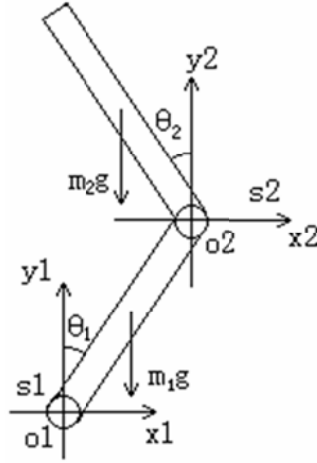


图 10.1 模型分析

联立式(10.1)~(10.5), 消去中间变量 M_{12} 和 M_{21} , 并将 M_0 代入, 得矩阵形式的系统非线性数学模型。

系统非线性模型:

$$\begin{bmatrix} J_1 + m_2 R_1^2 & m_2 R_1 L_2 \cos(\theta_1 - \theta_2) \\ m_2 R_1 L_2 \cos(\theta_1 - \theta_2) & J_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} f_1 + K_m K_e & m_2 R_1 L_2 \sin(\theta_1 - \theta_2) \dot{\theta}_2 \\ m_2 R_1 L_2 \sin(\theta_2 - \theta_1) \dot{\theta}_1 & f_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} K_m \\ 0 \end{bmatrix} u + \begin{bmatrix} m_1 g L_1 \sin \theta_1 + m_2 g R_1 \sin \theta_1 \\ m_2 g L_2 \sin \theta_2 \end{bmatrix} \quad (10.6)$$

根据非线性模型式(10.6), 令 $\theta_1 \rightarrow 0$, $\theta_2 \rightarrow 0$, 可将非线性模型线性化。

线性模型:

$$\begin{bmatrix} J_1 + m_2 R_1^2 & m_2 R_1 L_2 \\ m_2 R_1 L_2 & J_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} f_1 + K_m K_e & 0 \\ 0 & f_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} (m_1 L_1 + m_2 R_1)g & 0 \\ 0 & m_2 g L_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \begin{bmatrix} K_m \\ 0 \end{bmatrix} u \quad (10.6)$$

$$\text{令 } J = \begin{bmatrix} J_1 + m_2 R_1^2 & m_2 R_1 L_2 \\ m_2 R_1 L_2 & J_2 \end{bmatrix}, \quad F = \begin{bmatrix} f_1 + K_m K_e & 0 \\ 0 & f_2 \end{bmatrix}, \quad M = \begin{bmatrix} (m_1 L_1 + m_2 R_1)g & 0 \\ 0 & m_2 g L_2 \end{bmatrix},$$

$$K = \begin{bmatrix} K_m \\ 0 \end{bmatrix}。$$

则有

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = -J^{-1}F \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + J^{-1}M \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + J^{-1}Ku \quad (10.7)$$

系统的状态方程:

$$\begin{cases} \dot{x} = Ax + Bu \\ Y = Cx \end{cases}, \quad \text{其中 } x = [\theta_1 \quad \theta_2 \quad \dot{\theta}_1 \quad \dot{\theta}_2]^T, y = [\theta_1 \quad \theta_2]^T$$

$$A = \begin{bmatrix} 0_{2 \times 2} & I_{2 \times 2} \\ J^{-1}M & -J^{-1}F \end{bmatrix}, B = \begin{bmatrix} 0_{2 \times 1} \\ J^{-1}K \end{bmatrix}, C = [I_{2 \times 2} \quad 0_{2 \times 2}]$$

系统的主要机械参数及变量如表 10.1 所示。

表 10.1 机械参数及变量

旋臂质量 m_1	0.200kg	摆杆质量 m_2	0.052kg
旋臂长度 R_1	0.20m	摆杆长度 R_2	0.25m
旋臂质心到转轴距离 L_1	0.10 m	摆杆质心到转轴距离 L_2	0.12 m
电机力矩—电压比 K_m	0.0236 N·m/V	电机反电势—转速比 K_e	0.2865 V·s/m
旋臂绕轴转动摩擦 力矩系数 f_1	0.01 N·s	摆杆绕轴转动摩擦 力矩系数 f_2	0.001 N·s
旋臂绕轴转动惯 J_1	0.004 kg·m ²	摆杆绕轴转动惯量 J_2	0.001 kg·m ²

2. 系统分析

将参数值代入方程，在 MATLAB 中键入如下函数，可以求出系统的能控性、能观性、特征根，可看出系统是能控、能观、开环不稳定。

```
rank(ctrb(A,B))
```

```
rank(observ(A,C))
```

```
eig(A)
```

系统是完全可以控和完全可观测的，因此可以根据状态反馈确定反馈控制律，使系统闭环稳定。

3. 状态反馈控制的 MATLAB 仿真

为方便起见，令 $a=J_1+m_2R_1^2$, $b=m_2R_1L_2$, $c=J_2$, $d=f_1+K_mK_e$, $e=(m_1L_1+m_2R_1)g$, $f=f_2$, $h=m_2gL_2$ 在 MATLAB 中，将各个参数值代入非线性数学模型，进行仿真，利用 ODE 函数求解微分方程。

(1) 极点配置，求取状态反馈增益矩阵以及状态观测器增益矩阵

选择主菜单 file→new→m-file 建立文件 fang_daolibai.m:

```
clear;
```

```
clc;
```

```
A=[      0      0      1      0
      0      0      0      1
    65.8751 -16.8751 -3.7062  0.2760
   -82.2122  82.2122  4.6254 -1.3444]; % System matrix
```

```
B=[0;0;5.2184;-6.5125]; % Control Matrix
```

```
C=[1,0,0,0;0,1,0,0]; % output matrix
```

```
P=[-4+3i,-4-3i,-30-6i,-30+6i]; % expect pole
```

```
rank(ctrb(A,B)) % controllability
```

```
rank(observ(A,C)) % observability
```

```
eig(A) % stability
```

```
K=place(A,B,P) % state feedback gain matrix
```

```
P1=[-20+3i,-20-3i,-80-1i,-80+1i]; %观测器极点
```

```
L=place(A',C',P1)' %状态观测器增益矩阵
```

利用极点配置的方法求反馈矩阵 K。任取一组期望极点 P，在 MATLAB 中利用 place(A,B,P)函数求得 $K=[K_a, K_o, K_{va}, K_{vo}]$ 。

(2) 建立旋转式倒立摆的非线性模型的微分方程

运行 MATLAB6.0 以上的版本，选择主菜单 file→new→m-file 建立文件 dlfun.m:

```
function xdot=dlfun(t,x);
```

```

m1=0.200; m2=0.052; L1=0.10; L2=0.12; r=0.20; km=0.0236; ke=0.2865;
g=9.8; J1=0.004; J2=0.001; f1=0.01; f2=0.001; % value of parameter
a=J1+m2*r*r; b=m2*r*L2; c=J2; d=f1+km*ke; e=(m1*L1+m2*r)*g; f=f2; h=m2*L2*g;
K=[ * * * *];% State feedback gain matrix obtained according to the desired pole;
u=-K*[x(1);x(2);x(3);x(4)]; % control variable
xdot=zeros(5,1); %derivative of x
xdot(1)=x(3);
xdot(2)=x(4);
xdot(3)=((-d*c).*x(3)+(f*b*cos(x(2)-x(1))).*x(4)+b*b*sin(x(2)-x(1)).*cos(x(2)-x(1)).*x(3).*
x(3)-b*c*sin(x(1)-x(2)).*x(4).*x(4)+e*c*sin(x(1))-h*b*sin(x(2)).*cos(x(2)-x(1))+km*c*u)/(a*c-b
*b.*cos(x(1)-x(2)).*cos(x(2)-x(1)));
xdot(4)=((d*b*cos(x(1)-x(2))).*x(3)-(a*f).*x(4)-a*b*sin(x(2)-x(1)).*x(3).*x(3)+b*b*sin(x(1
)-x(2)).*cos(x(1)-x(2)).*x(4).*x(4)-e*b*sin(x(1)).*cos(x(1)-x(2))+a*h*sin(x(2))-b*cos(x(1)-x(2))
*km*u)/(a*c-b*b.*cos(x(1)-x(2)).*cos(x(2)-x(1))); % differential equations to describe the
nonlinear model

```

```

xdot(5)=-K*[xdot(1);xdot(2);xdot(3);xdot(4)]; % derivative of control variable

```

(3) 求解非线性模型的微分方程

选择主菜单 file→new→m-file 建立文件 daolibai.m:

Close all;

t0=0;tf=20;%simulation time

```

K=[ * * * *];% State feedback gain matrix obtained according to the desired pole;

```

```

r=[-0.1;0.05;0;0];%initial value of x1,x2,x3,x4

```

```

u0=-K*r;

```

```

x0=[r;u0];%initial value

```

```

[t,x]=ode45('dlfun',[t0,tf],x0); % solve differential equations

```

```

figure;

```

```

plot(t,x(:,1)*180/pi,'r',t,x(:,2)*180/pi,'b',t,x(:,5),'g');

```

```

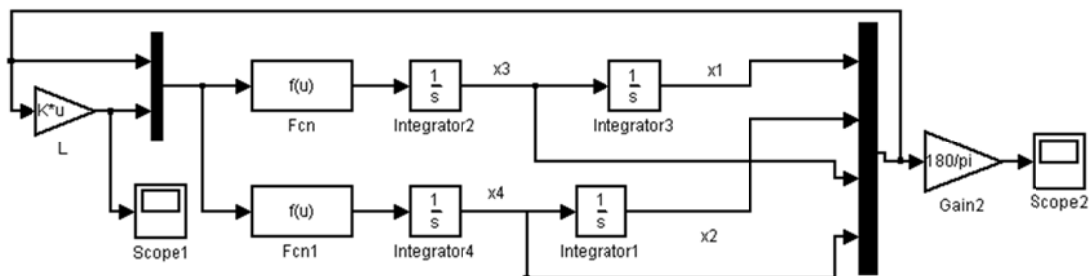
legend('spiral arm angle','the pendulum angle','control value');

```

(4) 仿真步骤:

- 选择期望极点 P，运行 fang_daolibai.m 求出状态反馈矩阵 K；
- 将状态反馈矩阵 K 代入到 dlfun.m 和 daolibai.m 中，点击保存；
- 在 daolibai.m 中，设置系统的初始值，运行并观察响应曲线。
- 选择多组极点，重复以上步骤。

4. 状态反馈控制的 Simulink 仿真



Fcn:

$$((-0.0168*0.001)*u[3]+(0.001*0.0012*\cos(u[2]-u[1]))*u[4]+0.0012*0.0012*\sin(u[2]-u[1])* \cos(u[2]-u[1])*u[3]*u[3]-0.0012*0.001*\sin(u[1]-u[2])*u[4]*u[4]+0.2979*0.001*\sin(u[1])-0.0612 *0.0012*\sin(u[2])* \cos(u[2]-u[1])+0.0236*0.001*u[5])/ (0.0061*0.001-0.0012*0.0012*\cos(u[1]-u[2])* \cos(u[2]-u[1]))$$

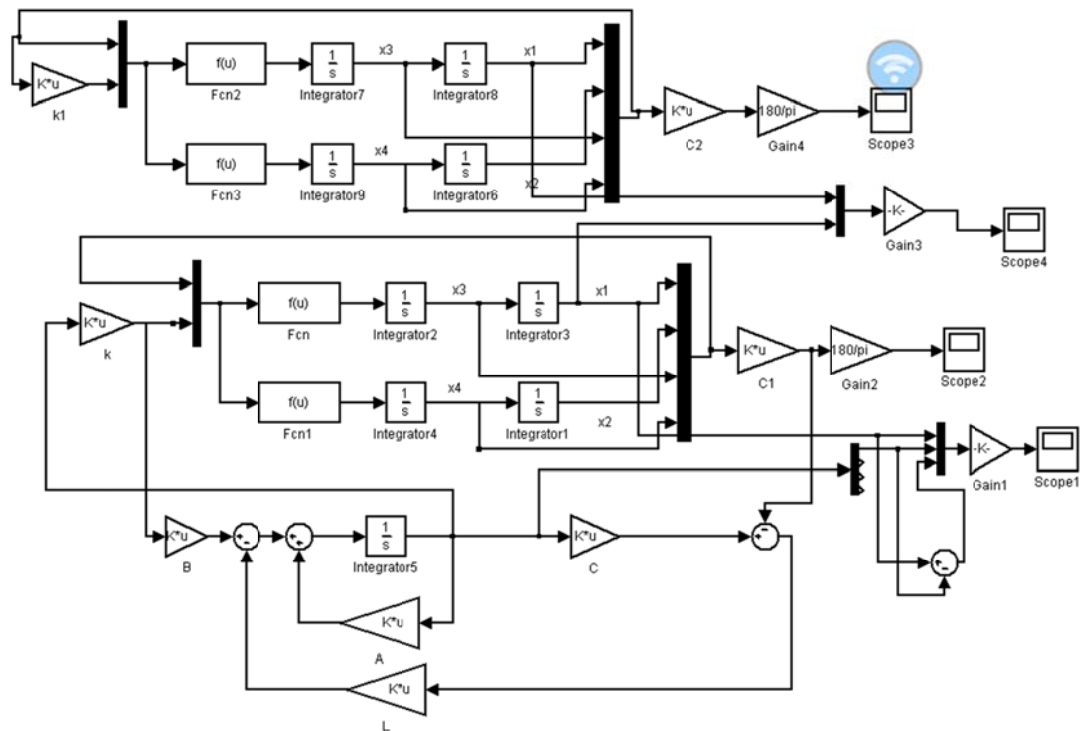
Fcn1:

$$((0.0168*0.0012*\cos(u[1]-u[2]))*u[3]-(0.0061*0.001)*u[4]-0.0061*0.0012*\sin(u[2]-u[1])*u [3]*u[3]+0.0012*0.0012*\sin(u[1]-u[2])* \cos(u[1]-u[2])*u[4]*u[4]-0.2979*0.0012*\sin(u[1])* \cos(u [1]-u[2])+0.0061*0.0612*\sin(u[2])-0.0012*\cos(u[1]-u[2])*0.0236*u[5])/ (0.0061*0.001-0.0012*0. 0012*\cos(u[1]-u[2])* \cos(u[2]-u[1]))$$

仿真步骤:

- 选择期望极点 P，运行 fang_daolibai.m 求出状态反馈矩阵 K；
- 将状态反馈矩阵 K 代入图中的 L 模块中，点击保存；
- 双击 integrator3 设置 x_1 (θ_1) 的初始值，双击 integrator1 设置 x_2 (θ_2) 的初始值，运行并观察响应曲线。

5. 状态观测器的 Simulink 仿真



仿真步骤:

- 选择倒立摆的期望极点 P 和观测器的七万极点 P1，运行 fang_daolibai.m 求出状态反馈矩阵 K 和观测器增益矩阵 L；
- 将状态反馈矩阵 K 代入图中的 k 和 k1 模块中，将 L 带入图中 L 模块中点击保存；
- 双击 integrator3 设置 x_1 (θ_1) 的初始值，双击 integrator1 设置 x_2 (θ_2) 的初始值，运行并观察响应曲线。

6. 实时控制

打开 dsp.exe，选择“控制模式”。在参数设置中，按设计好的反馈参数，设置 K_a , K_o , K_{va} , K_{vo} (见图 6.5 所示参数设置对话框)。点击“OK”并进行联机控制。根据倒立摆的实际运行情况，可以修改这些参数，改善控制效果。

六、实验报告要求

1. 对一阶旋转式倒立摆系统进行动力学分析，并推导其数学模型。
2. 编写 MATLAB 仿真程序，记录输出的数据、曲线，并进行理论分析。
3. 在设计状态反馈参数中，给定多组极点进行实验，观测仿真结果，比较超调量，调节时间等性能指标。
4. 分析引入状态观测器对系统性能的影响。

七、思考题

1. 以倒立摆系统为例，简述极点配置方法设计系统控制器的基本思路。
2. 非线性系统进行如何局部线性化？
3. 描述期望极点的位置对闭环系统控制律的幅值的影响，说明原因。
4. 倒立摆系统的非线性方程式如何推导的，简单说明其过程
5. 为什么说闭环控制后的系统为非最小相位系统？
6. 描述极点位置与系统的动态性能、稳态性能和稳定性的关系。
7. 观测器的极点和倒立摆的极点，哪个应离虚轴更近一些，为什么？