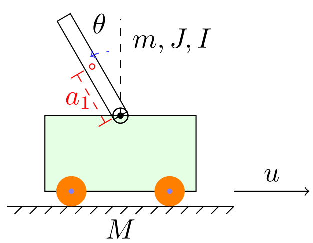
系统建模问题背景

问题 4

试用 Simulink 搭建如下倒立摆的 PID 和 LQR 控制器，假设倒立摆的状态空间模型为：

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du = \begin{bmatrix} \theta \\ x \end{bmatrix}$$

状态矩阵 $A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 41.63 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -0.6099 & 0 & 0 & 0 \end{bmatrix}$,$B = \begin{bmatrix} 0 \\ -2.7584 \\ 0 \\ 0.6898 \end{bmatrix}$,$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$,$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$，状态向量 $x(t) = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix}$，控制输入 $u(t)$，输出 $y(t)$。



其中，M 为摆杆的质量，m 为小球的质量，J 为摆杆的转动惯量，I 为小球的质量矩 (这里两种模型的 J, I 可以看为一种东西)，u 为输入力，θ 为摆杆与水平线的夹角，l 为摆杆重心与水平线的距离。这里给出假设：$M = 1.42, m = 0.12, b = 0.1, l = 0.188, J = I = 0.0014$，均为国际单位制 (kg,m,s)。

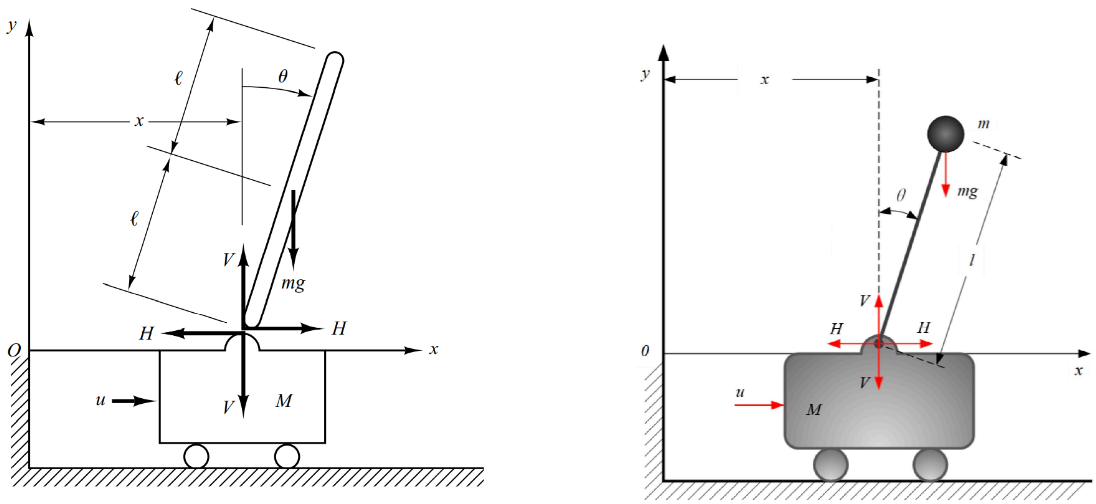
数学推导与模型构建

如下图所示，我们拥有两种倒立摆系统的模型假设：

- 在杆模型中，我们假设杆的质量是均匀分布的，并且不考虑系统的摩擦。

- 在球模型中，我们胡烈摆杆质量和地面摩擦，并且假设杆的质量集中在杆的末端。

这种假设影响的是重心位置，但是我们可以通过巧妙地让重心到连接点的长度为 l 来寻找统一的表达式。



(a) 倒立摆杆质量模型 (b) 倒立摆球质量模型

图 1: 两种倒立摆模型

倒立摆杆质量模型

严格来说：假设 u 为外部作用力，M 为小车质量，m 为摆杆质量，摆杆长度为 $2l$ 且摆杆质量均匀分布。设 x 为小车位置，θ 为摆杆与竖直方向的夹角。系统不考虑空气阻力的影响。那么我们根据牛顿第二定律，考虑摆杆重心的水平运动和垂直运动，然后在考虑小车的水平运动和摆杆的转动 (绕重心的转动运动)。

$$\begin{cases} m\frac{d^2}{dt^2}(x + l\sin(\theta)) = H \\ m\frac{d^2}{dt^2}(l\cos(\theta)) = -mg + V \\ M\frac{d^2}{dt^2}(x) = u - H \\ I\frac{d^2}{dt^2}(\theta) = Vl\sin(\theta) - Hl\cos(\theta) \end{cases}$$

联立化简即可，我们为了方便后续设计，假设倒立摆角度很小，近似认为 $\sin\theta = \theta, \cos\theta = 1$; 以及倒立摆绕重心的转动惯量很小，近似认为 $I = 0$。那么可以得到：

$$\begin{cases} (M + m)\ddot{x} + ml\ddot{\theta} = u \\ ml^2\ddot{\theta} + ml\ddot{x} = mgl\theta \end{cases}$$

定义状态变量 X:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix}$$

那么可以得到状态方程:

$$\dot{X} = \begin{bmatrix} \dot{\theta} \\ \dot{\dot{\theta}} \\ \dot{x} \\ \dot{\dot{x}} \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{(M+m)g}{Ml} x_1 - \frac{1}{Ml} u \\ x_4 \\ -\frac{mg}{M} x_1 + \frac{1}{M} u \end{bmatrix}$$

可以写出以为倒立摆系统的状态空间方程表达式:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(M+m)g}{Ml} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{M} & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{Ml} \\ 0 \\ \frac{1}{M} \end{bmatrix} \cdot u$$

$$y = CX + D\hat{u} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot \hat{u} = \begin{bmatrix} \theta \\ x \end{bmatrix}$$

倒立摆球质量模型

严格来说：假设 u 为外部作用力，M 为小车质量，m 为摆杆末端球质量，摆杆长度为 l 且摆杆质量集中在杆的末端。设 x 为小车位置，θ 为摆杆与竖直方向的夹角。系统不考虑空气阻力的影响。

同上述过程，我们考虑倒立摆摆杆重心的水平-垂直运动与小车水平运动和摆杆转动:

$$\begin{cases} m\frac{d^2}{dt^2}(x + l\sin(\theta)) = H \\ m\frac{d^2}{dt^2}(l\cos(\theta)) = -mg + V \\ M\frac{d^2}{dt^2}(x) = u - H \\ I\frac{d^2}{dt^2}(\theta) = Vl\sin(\theta) - Hl\cos(\theta) \end{cases}$$

整理一下:

$$\begin{cases} (M+m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = u \\ ml\ddot{x}\cos\theta + (I + ml^2)\ddot{\theta} - mgl\sin\theta = 0 \end{cases}$$

那么我们同样假设在系统平衡点附近，摆杆的角度、角速度可以忽略不计，即 $\cos\theta = 1, \sin\theta = \theta; \dot{\theta} = 0$，$\sin\theta\dot{\theta} = 0$，则:

$$\begin{cases} (M+m)\ddot{x} + ml\ddot{\theta} = u \\ ml\ddot{x} + (I+ml^2)\ddot{\theta} = mgl\theta \end{cases}$$

定义状态变量 X:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix}$$

那么同理我们可以得到系统状态空间方程:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{m^2gl^2}{I(M+m)+Mml^2} & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{ml}{I(M+m)+Mml^2} \\ 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \end{bmatrix} \cdot u$$

$$y = CX + D\hat{u} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot \hat{u} = \begin{bmatrix} \theta \\ x \end{bmatrix}$$

更一般的形式

我们这里还要考虑近来阻尼系数 b, 那么我们现在的方程就变成了:

$$\begin{cases} (M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta} = u \\ ml\ddot{x} + (I+ml^2)\ddot{\theta} = mgl\theta \end{cases}$$

那么在添加了阻尼系数之后，我们更一般的表达式应该是:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mgl(M+m)}{p} & -\frac{(I+ml^2)b}{p} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{m^2gl^2}{p} & -\frac{mlb}{p} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{ml}{p} \\ 0 \\ \frac{I+ml^2}{p} \end{bmatrix} \cdot u$$

$$y = CX + D\hat{u} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot \hat{u} = \begin{bmatrix} \theta \\ x \end{bmatrix}$$

其中,$p=I(M+m)+Mml^2$，当然还有 $q=(M+m)(I+ml^2)-(ml)^2$，在后面 pid 调节的时候可以使用。

控制环节

LQR 控制

已知系统状态空间模型:

$$\dot{X}=AX+Bu$$

其中 X 为状态向量，u 为控制输入，A 为状态矩阵，B 为输入矩阵。

LQR 控制考虑最佳控制向量的矩阵 K:

$$\hat{u}=u-KX$$

其中 K 为增益矩阵，X 为状态向量。

那么 LQR 控制的目标转换为确定下列最小化二次型函数:

$$J=\frac{1}{2}\int_0^T(X^TQX+u^TRu)dt$$

此时不妨选择 $Q=I,R=I$，使用 matlab 代码求解得到增益矩阵 K:

```
% 倒立摆系统参数
M = 1.42;    % 小车质量
m = 0.12;    % 摆球质量
l = 0.188;   % 摆杆长度
I = 0.0014;    % 摆球转动惯量
g = 9.8;     % 重力加速度9.8m/s^2
q = (M+m)*(I+m*l^2)-(m*l)^2;
p = I*(M+m)+(M*m*l^2);
% 倒立摆系统状态空间矩阵定义
A = [0              1 0 0
   (M+m)*m*g*l/(I*(M+m)+(M*m*l^2)) 0 0 0
   0              0 0 1
   -m^2*g*l^2 /(I*(M+m)+(M*m*l^2))    0 0 0];
B = [0 -m*l/(I*(M+m)+(M*m*l^2)) 0 (I+m*l^2)/(I*(M+m)+(M*m*l^2))]';
Q=eye(4);
R=eye(1);
K=lqr(A,B,Q,R);
```

此时可以输出得到 $k = [-36.9690, -5.8463, -1.0000, -2.3376]$。

那么我们拥有了这个 k，就可以搭建 LQR 控制器了，首先考虑如下理论模型，我们注意到，在 LQR 控制器中，反馈 K 到输入 u 中，然后在乘以 B，与 A 反馈相加之后。

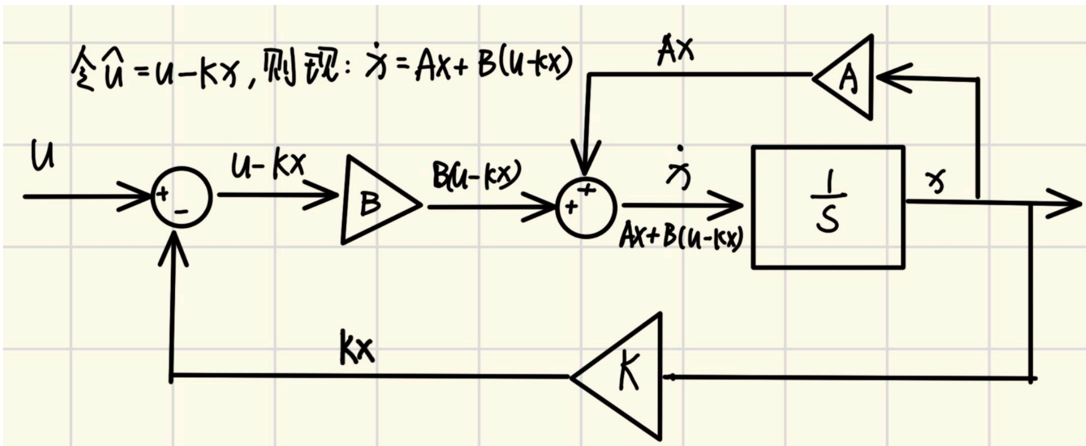


图 2: LQR 控制器

PID 控制

进行 laplace 变换, 同时假设 $\theta, \dot{\theta}$ 很小, 根据上述推导不难有如下式子:

$$\begin{cases}(M+m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta + b\dot{x} = u \\ ml\ddot{x}\cos\theta + (I+ml^2)\ddot{\theta} - mgl\sin\theta = 0\end{cases}$$

有初始状态为 0，而且 $\sin\theta = \theta, \cos\theta = 1, \dot{\theta} = 0$:

$$(I+ml^2)\Phi(s)s^2 - mgl\Phi(s) = mlX(s)s^2 - ml\Phi(s)s^2 + bX(s)s = U(s)$$

那么可以给出两种传递函数:

$$P_{\text{pend}}(s) = \frac{\Phi(s)}{U(s)} = \frac{mls}{((M+m)(I+ml^2)-(ml)^2)s^3 + b(I+ml^2)s^2 - (M+m)mgls - bmgl}$$

$$P_{\text{cart}}(s) = \frac{X(s)}{U(s)} = \frac{(I+ml^2)s^2 - mgl}{((M+m)(I+ml^2)-(ml^2))s^4 + b(I+ml^2)s^3 - (M+m)mgls^2 - bmgls}$$

当然，我们还有另外一种传递函数:

$$\begin{cases}\ddot{x} = \frac{(I+ml^2)u + ml(I+ml^2)\dot{\theta}^2\sin\theta - m^2l^2g\sin\theta\cos\theta}{(I+ml^2)(M+m) - m^2l^2\cos^2\theta} \\ \ddot{\theta} = \frac{mlu\cos\theta + m^2l^2\dot{\theta}^2\sin\theta\cos\theta - (M+m)mlg\sin\theta}{m^2l^2\cos^2\theta - (I+ml^2)(M+m)}\end{cases}$$

这个时候带入 $\theta \to 0$ 求解，有:

$$\begin{cases} \ddot{x} = \frac{0.0022u - 0.005\theta}{0.0082} \\ \ddot{\theta} = \frac{0.0226u - 0.3405\theta}{-0.0082} \end{cases}$$

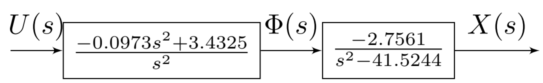
所以在平衡点附近的参数就线性化为:

$$\begin{cases} \ddot{x} = 0.2683u - 0.6098\theta \\ \ddot{\theta} = -2.7561u + 41.5244\theta \end{cases}$$

然后 laplace 变换，得到:

$$\begin{cases} X(s) = \frac{-0.0973s^2 + 3.4325}{s^2}\Phi(s) \\ \Phi(s) = \frac{-2.7561}{s^2 - 41.5244}U(s) \end{cases}$$

我们根据这个可以画出来系统框图如下:



这样关于角度和关于小车位移的传递函数就有了，可以进行 pid 控制仿真了。

```
%Modeling
%%Transfer Function
mCart = 1.42;
mPend = 0.12;
b = 0.1;   % 阻尼系数
I = 0.014;   % 转动惯量
g = 9.8;
L = 0.188;
q = (mCart+mPend)*(I+mPend*L^2)-(mPend*L)^2;
s = tf('s');
P_cart = (((I+mPend*L^2)/q)*s^2 - (mPend*g*L/q))/(s^4 + (b*(I + mPend*L^2))*s^3/q -
    ((mCart + mPend)*mPend*g*L)*s^2/q - b*mPend*g*L*s/q);
P_pend = (mPend*L*s/q)/(s^3 + (b*(I + mPend*L^2))*s^2/q - ((mCart + mPend)*mPend*g*L)*s/q
    - b*mPend*g*L/q);
sys_tf = [P_cart ; P_pend]
inputs = {'u'};
outputs = {'x'; 'phi'};
set(sys_tf,'InputName',inputs)
set(sys_tf,'OutputName',outputs)
% PID环节
Kp = 100;
Ki = 1;
```

```
Kd = 30;
C = pid(Kp,Ki,Kd);
T1 = feedback(P_pend,C);
T2 = feedback(1,P_pend*C)*P_cart;
```

我们现在有输出如下:

$$P_{\mathrm{cart}} = \frac{1.388\times10^{-5}s^2-0.0001682}{2.098\times10^{-5}s^4+1.388\times10^{-6}s^3-0.000259s^2-1.682\times10^{-5}s} \quad \text{从输入 U(s) 到输出 1X(s)}$$
$$P_{\mathrm{pend}} = \frac{1.716\times10^{-5}s}{2.098\times10^{-5}s^3+1.388\times10^{-6}s^2-0.000259s-1.682\times10^{-5}} \quad \text{从输入 U(s) 到输出 } 2\Phi(S)$$

随后我们转化:

$$\mathrm{T1} = \frac{1.716\times10^{-5}s^2}{2.098\times10^{-5}s^4+0.0005163s^3+0.001457s^2+3.433\times10^{-7}s}$$
$$\mathrm{T2} = \frac{2.912\times10^{-10}s^6+1.926\times10^{-11}s^5-7.125\times10^{-9}s^4-4.669\times10^{-10}s^3+4.357\times10^{-8}s^2+2.829\times10^{-9}s}{4.404\times10^{-10}s^8+1.086\times10^{-8}s^7+2.586\times10^{-8}s^6-1.321\times10^{-7}s^5-3.862\times10^{-7}s^4-2.46\times10^{-8}s^3-5.774\times10^{-12}s^2}$$

这个时候我们可以直接运算 T1,T2 的脉冲响应即可。

而同时我们也有这样的 PID 控制器框图如下:

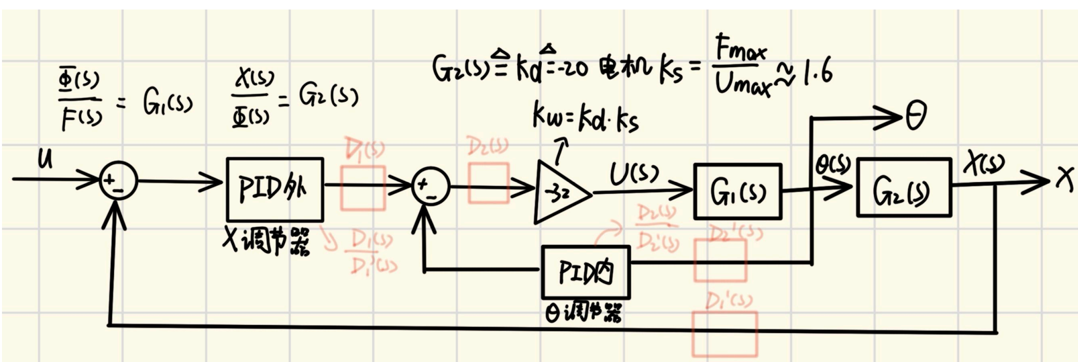


图 3: 双 PID 控制器

Matlab 代码与 Simulink 仿真测试

LQR 控制

根据上述操作，我们已经大概知道了怎么计算，现在我们进行 matlab 仿真测试。

```
% 倒立摆系统参数
M = 1.42;    % 小车质量
m = 0.12;    % 摆球质量
```

```
l = 0.188;  % 摆杆长度
I = 0.0014;   % 摆球转动惯量
g = 9.8;    % 重力加速度9.8m/s^2
q = (M+m)*(I+m*l^2)-(m*l)^2;
p = I*(M+m)+(M*m*l^2);
% 倒立摆系统状态空间矩阵定义
A = [0              1 0 0
   (M+m)*m*g*l/(I*(M+m)+(M*m*l^2)) 0 0 0
   0              0 0 1
   -m^2*g*l^2 /(I*(M+m)+(M*m*l^2))    0 0 0];
B = [0 -m*l/(I*(M+m)+(M*m*l^2)) 0 (I+m*l^2)/(I*(M+m)+(M*m*l^2))]';
C = [1 0 0 0
   0 0 1 0];
D = 0;
% 倒立摆系统状态空间模型
G = ss(A, B, C, D)
```

我们有输出如下，可以看出，尽管我们定义了阻尼系数 b，但是这道题这样给出来的状态矩阵就是没有计算阻尼系数的形式，但是与题目所给形式完全一致:

```
G =
  A =
          x1      x2      x3      x4
   x1      0       1       0       0
   x2  41.63       0       0       0
   x3      0       0       0       1
   x4 -0.6099      0       0       0
  B =
         u1
   x1      0
   x2 -2.758
   x3      0
   x4 0.6898
  C =
      x1  x2  x3  x4
   y1  1   0   0   0
   y2  0   0   1   0
  D =
       u1
```

```
y1   0
y2   0
```
连续时间状态空间模型。

那么根据之前讨论过的 (主要参考控制环节-LQR 控制)，这里主要使用了图 2 中的理论系统形式, 我们可以搭出 LQR 控制器的 simulink 文件:

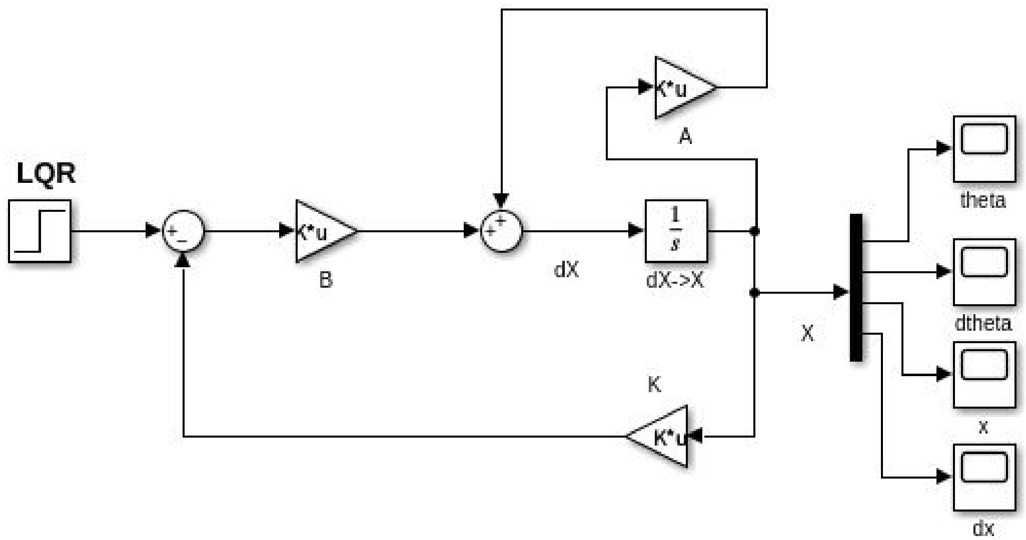


图 4: simulink 仿真文件

完整代码

```
% 定义状态矩阵 A 和控制矩阵 B 观测矩阵 C 和输入矩阵 D
A = [0, 1, 0, 0;
    41.63, 0, 0, 0;
    0, 0, 0, 1;
    -0.6099, 0, 0, 0];
B = [0;
    -2.7584;
    0;
    0.6898];
C = [1, 0, 0, 0;
    0, 0, 1, 0];
```

```
D = [0;
    0];
G = ss(A, B, C, D);
% 定义权重矩阵 Q 和 R
Q = eye(4); % 状态权重矩阵，这里使用单位矩阵
R = 1;     % 控制权重矩阵，这里使用 1
[K, S, E] = lqr(A, B, Q, R);
B
A
K
A_cl = A - B * K;
B_cl = B;
C_cl = C;
D_cl = D;
Gclose = ss(A_cl, B_cl, C_cl, D_cl);

% 计算并绘制阶跃响应
t = 0:0.01:10;
[y, t, x] = step(Gclose, t);

figure;
% 摆角与角速度响应
subplot(2, 1, 1);
plot(t, x(:, 1), 'b-', 'LineWidth', 2);
hold on;
plot(t, x(:, 2), 'r--', 'LineWidth', 2);
grid on;
xlabel('time(s)', 'FontSize', 20);
ylabel('{\theta and \dot \theta}', 'FontSize', 20);
title('摆角与角速度响应', 'FontSize', 25);
legend('摆角', '角速度');
% 位移与速度响应
subplot(2, 1, 2);
plot(t, x(:, 3), 'g-', 'LineWidth', 2);
hold on;
plot(t, x(:, 4), 'm--', 'LineWidth', 2);
grid on;
xlabel('time(s)', 'FontSize', 20);
ylabel('{x and \dot x}', 'FontSize', 20);
```

```
title('位移与速度响应', 'FontSize', 25);
legend('位移', '速度');

set(gcf, 'Position', [100, 100, 800, 600]);
```

PID 控制

那么根据之前讨论过的 (主要参考)，我们可以搭出 PID 控制器的 simulink 文件:

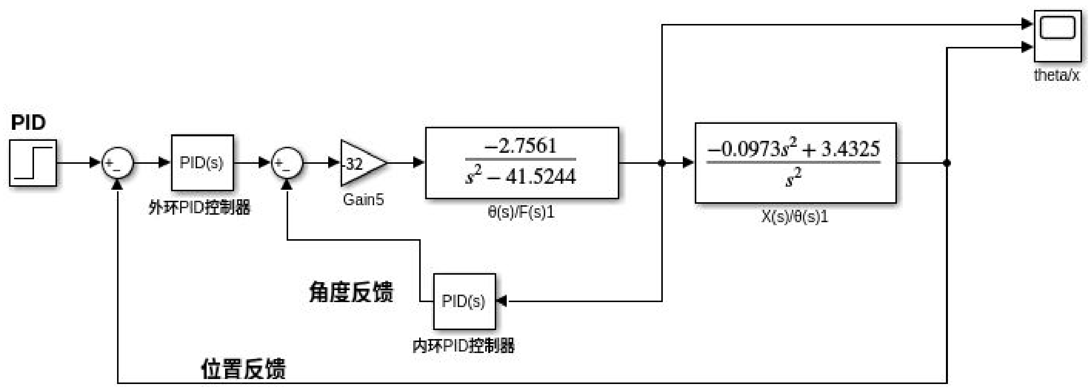


图 5: simulink 仿真文件

完整代码

```
%Modeling
%%Transfer Function
mCart = 1.42;
mPend = 0.12;
b = 0.1;    % 阻尼系数
I = 0.014;    % 转动惯量
g = 9.8;
L = 0.188;
q = (mCart+mPend)*(I+mPend*L^2)-(mPend*L)^2;
s = tf('s');
P_cart = (((I+mPend*L^2)/q)*s^2 - (mPend*g*L/q))/(s^4 + (b*(I + mPend*L^2))*s^3/q -
    ((mCart + mPend)*mPend*g*L)*s^2/q - b*mPend*g*L*s/q);
```

```
P_pend = (mPend*L*s/q)/(s^3 + (b*(I + mPend*L^2))*s^2/q - ((mCart + mPend)*mPend*g*L)*s/q
    - b*mPend*g*L/q);
sys_tf = [P_cart ; P_pend]
inputs = {'u'};
outputs = {'x'; 'phi'};
set(sys_tf,'InputName',inputs)
set(sys_tf,'OutputName',outputs)
t = 0:0.05:5;
u = ones(size(t));
[y,t] = lsim(sys_tf,u,t);
figure ;
plot(t,y) ;
title('Open-Loop Step Response') ;
% axis([0 3 0 50]) ;
legend('x','phi');
% PID环节
Kp = 100;
Ki = 1;
Kd = 30;
C = pid(Kp,Ki,Kd);
T1 = feedback(P_pend,C);
t=0:0.01:4;
figure;
subplot(2,1,1);
impulse(T1,t)
title('Close-loop impluse Response')
legend('phi');
% figure;
T2 = feedback(1,P_pend*C)*P_cart;
t = 0:0.01:5;
subplot(2,1,2);
impulse(T2, t);
title('Close-loop impluse Response')
legend('x');
```