**基于LQR的平衡小车**

**1.LQR控制器**

1.1概述

LQR即线性二次型调节器，其基于贝尔曼最优原则，是一种用于线性系统的最优控制器。

设一个系统完全能控的状态方程为：

通常情况下，最基本的控制系统是需要全状态反馈控制的，而目标一般是设计一个状态反馈控制器，即u=-Kx来控制系统的表现：



1.2最优控制

然而K并不是唯一的，那么什么情况下K才是最优的呢？

在这样的前提下需要引入一个评价标准，对于控制系统，一般使用代价函数（Quadratic Cost Function）来作为其的评价标准，代价函数越小，说明性能指标越好，反之则越差。

代价函数的一般形式如下：

其中Q为n\*n的半正定的状态加权矩阵，R为n\*n的正定的控制加权矩阵。Q,R在工程中常为兑成对角矩阵，其对角线上的元素表示对矩阵中对应分量的重视程度，越大则对的重视程度越大，其在代价函数中便会更快趋向更小的值。

LQR的思路是设计一个反馈控制器u=-Kx,使得代价函数最小，即, K矩阵的最小值的表达形式是这样的

已知有状态方程和代价函数：

所以对于最优的K矩阵可以表示为如下：

其中P是如下方程的解

在实际应用中，出于效率的考量，往往不使用人工的方法计算K，一般使用计算机软件辅助计算K，在matlab中一般使用库函数来计算

**2.小车的数学模型**

两轮小车的结构由车体和双轮组成，可以简化的看作一个移动倒立摆，分别对小车的车轮和车体进行力学分析，可以通过其建立系统的状态空间表达式。

2.1小车的状态方程

在对小车进行物理分析后可以建立小车的状态方程，因为在小车的运动全过程中小车的倾角一直很小，所以可以认为=1,,=0，所以有如下状态方程：

= +

其中状态变量从上至下依次表示 小车的位移，前进速度，车体的倾角，车体的角速度，小车的转向角，转向速度

电机的扭矩和车轮的加速度有线性关系如下，因此可以转化为：

=/I

=/I

其中

左轮无摩擦力时的线速度大小（rad/s）

右轮无摩擦力时的线速度大小（rad/s）

故而，对于系统的状态空间表达式

其中各参数的值如下

A= B=(I/r) C= D=0

x= u=

=

=

=

=

=

=

=

=

=

m 车轮的质量（kg）

r 车轮的半径（m）

车轮的水平位移（m）

车轮收到地面的摩擦力（N）

车轮收到车体作用力的水平方向的分力（N）

车轮电机输出的转矩（N\*m）

I 车轮的转动惯量（kg\*m^2）

车轮的角速度（rad/s）

车轮的质量（kg）

质心距离底盘中心的距离（m）

车体绕质心转动时的转动惯量（kg\*）

车体与竖直方向的夹角（rad）

轮距（m）

车体绕y轴转动时的转动惯量（kg\*）

小车的偏航角（rad）

左轮无摩擦力时的线速度大小（rad/s）

右轮无摩擦力时的线速度大小（rad/s）

�

**3.matlab计算K矩阵simulink仿真以及keil5的C代码**

3.1 Matlab代码和Simulink仿真

反馈矩阵很难通过手动计算来得出，因此往往利用matlab的函数来进行计算，程序如下：

Clc

m = 0.035;

r = 0.0672/2;

i = 0.5\*m\*r^2;

M = 0.737-2\*m;

L = 0.5\*0.0903;

J\_p = (1/12)\*M\*(0.0903^2+0.0530^2);

d = 0.1612;

J\_delta = (1/12)\*M\*(0.0930^2+0.0530^2);

g = 9.8;

Q\_eq = J\_p\*M+(J\_p+M\*L^2)\*(2\*m+2\*i/r^2);

A\_23 = -(M^2\*L^2\*g)/Q\_eq;

A\_43 = M\*L\*g\*(M+2\*m+2\*i/r^2)/Q\_eq;

B\_21 = (J\_p+M\*L^2+M\*L\*r)/(Q\_eq\*r);

B\_22 = B\_21;

B\_41 = -(M\*L/r+M+2\*m+2\*i/r^2)/Q\_eq;

B\_42 = B\_41;

B\_61 = 1/(r\*(m\*d+i\*d/r^2+2\*J\_delta/d));

B\_62 = -B\_61;

A = [0 1 0 0 0 0; 0 0 A\_23 0 0 0; 0 0 0 1 0 0; 0 0 A\_43 0 0 0; 0 0 0 0 0 1; 0 0 0 0 0 0];

B = (i/r)\*[0 0; B\_21 B\_22; 0 0; B\_41 B\_42; 0 0; B\_61 B\_62];

Tc = ctrb(A,B);

if (rank(Tc)==6)

fprintf('此系统是可控的！\n');

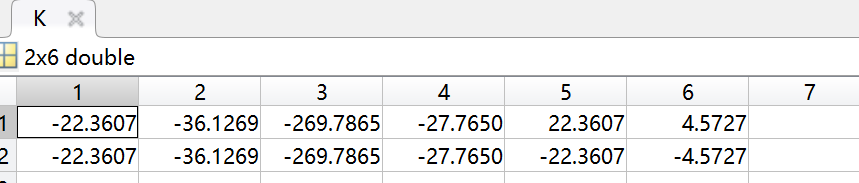
Q = [1000 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 1000 0 0; 0 0 0 0 1000 0; 0 0 0 0 0 0];

R = [1 0; 0 1];

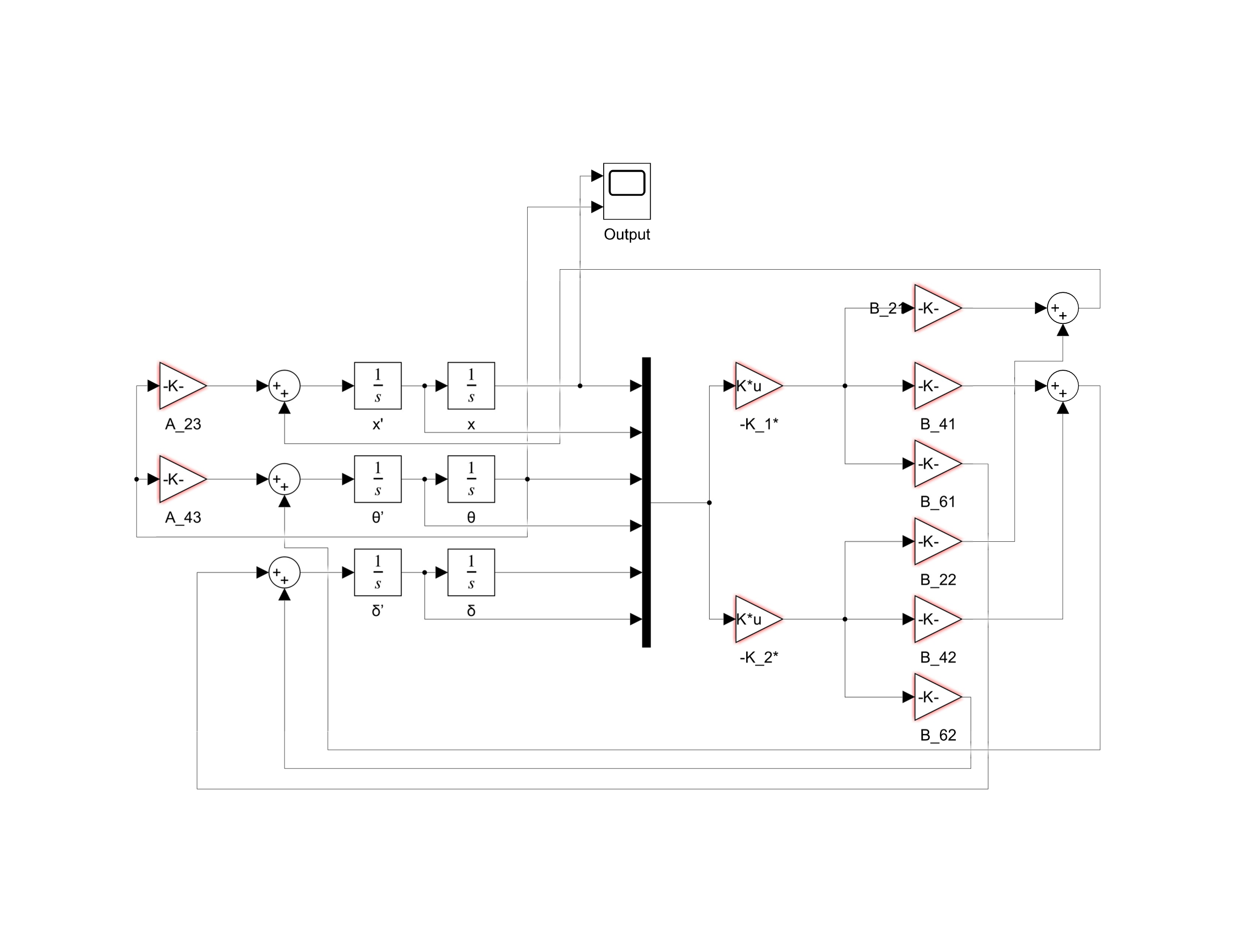
K = lqr(A,B,Q,R);

End

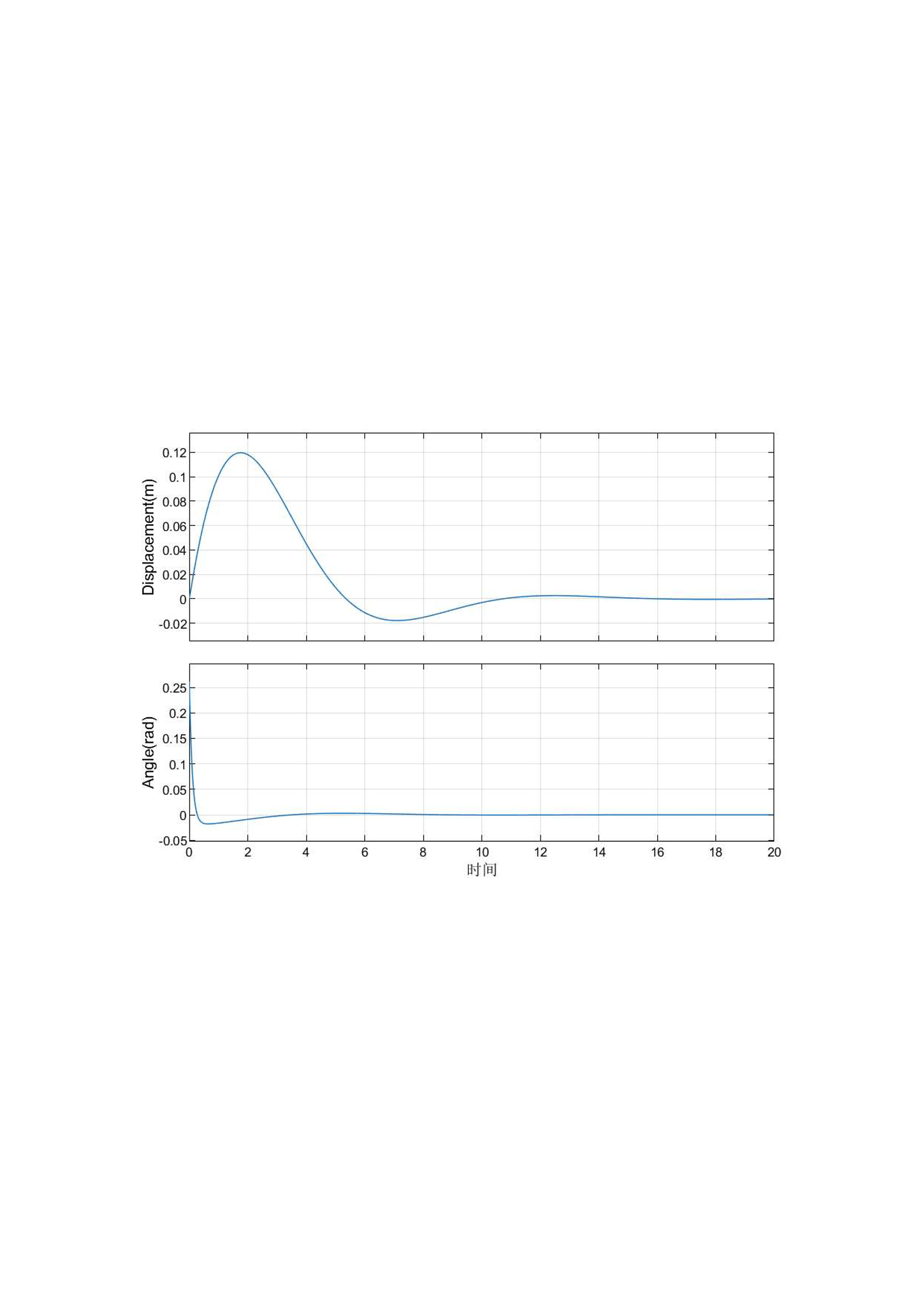
最终可以得出与之相对应的K矩阵：



之后使用simulink进行仿真，系统如下



仿真结果：



3.2keil5代码（c语言）

L\_accel=-(K1\*x\_pose + K2\*(x\_speed-Target\_x\_speed) + K3\*(angle\_x-Target\_angle\_x)+K4\*gyro\_x + K5\*angle\_z + K6\*(gyro\_z-Target\_gyro\_z));

R\_accel=-(K1\*x\_pose + K2\*(x\_speed-Target\_x\_speed) + K3\*(angle\_x-Target\_angle\_x)+K4\*gyro\_x - K5\*angle\_z - K6\*(gyro\_z-Target\_gyro\_z));

LQR控制器的效果是使得所有的状态变量都变为0， LQR控制器的效果是使得所有的状态变量都变为0，那么，可以把“x\_speed-Target\_x\_speed”作为新的状态变量，使它变为 0 即等价于使得 x\_speed=Target\_x\_speed。）