

可以。下面是这篇《倒立摆：采用状态空间方法设计控制器》的中文整理翻译。为了便于理解，我保留关键公式、参数和 MATLAB 代码含义，对重复代码进行了压缩。[原始教程](#)

倒立摆：状态空间控制器设计

本教程主要使用以下 MATLAB 命令：

```
ss      % 创建状态空间系统
eig      % 计算特征值/系统极点
lsim     % 仿真系统响应
lqr      % 设计LQR控制器
ctrb     % 构造可控性矩阵
obsv     % 构造可观性矩阵
place    % 配置系统极点
```

一、倒立摆状态空间模型

系统状态定义为：

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix}$$

其中：

- (x)：小车位置
- ($\dot{x}$)：小车速度
- (ϕ)：摆杆偏离竖直位置的角度
- ($\dot{\phi}$)：摆杆角速度
- (u)：施加在小车上的控制力

摆杆实际角度满足：

$$\theta = \pi + \phi$$

也就是说，($\phi=0$) 表示摆杆处于竖直向上的平衡位置。

系统方程是：

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

其中：

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -0.1818 & 2.6727 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.4545 & 31.1818 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1.8182 \\ 0 \\ 4.5455 \end{bmatrix}$$

系统输出为小车位置和摆杆角度：

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}u \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

二、控制目标

给小车输入一个向右移动 (0.2 m) 的位置指令，希望达到：

- 小车位置和摆杆角度的稳定时间小于 5 秒；
- 小车位置上升时间小于 0.5 秒；

- 摆杆偏离竖直方向不超过 (20°) ，约为 (0.35 rad) ;
- 小车位置和摆杆角度的稳态误差小于 2%。

这里不只是为了要让摆杆保持竖直，还要控制小车移动到指定位置。

由于需要同时控制小车位置和摆杆角度，因此状态空间方法非常合适。

三、全状态反馈

采用控制律：

$$[u = -K\mathbf{x}]$$

展开以后：

$$[u = -(k_1x + k_2\dot{x} + k_3\phi + k_4\dot{\phi})]$$

这意味着控制器同时根据以下四个量决定电机输出：

- 小车位置；
- 小车速度；
- 摆杆角度；
- 摆杆角速度。

闭环系统为：

$$[\dot{\mathbf{x}} = (A - BK)\mathbf{x}]$$

四、开环极点

系统参数为：

```
M = 0.5;      % 小车质量
m = 0.2;      % 摆杆质量
b = 0.1;      % 小车摩擦系数
I = 0.006;    % 摆杆转动惯量
g = 9.8;      % 重力加速度
l = 0.3;      % 摆杆质心到转轴的距离
```

通过：

```
poles = eig(A)
```

得到开环极点：

```
0
-0.1428
-5.6041
5.5651
```

其中 (5.5651) 是正数，位于复平面的右半平面，所以开环系统不稳定。

这符合直觉：如果不进行控制，竖直向上的摆杆会倒下。

五、检查系统是否可控

在设计状态反馈控制器之前，要先确认输入 (u) 能否控制全部状态。

可控性矩阵为：

$$[\text{matrix} C = \begin{matrix} B & AB & A^2B & \cdots & A^{n-1}B \end{matrix}]$$

MATLAB 代码：

```
co = ctrb(A,B);  
controllability = rank(co)
```

计算结果：

```
controllability = 4
```

系统共有四个状态，可控性矩阵的秩也是 4，因此系统完全可控。

这说明理论上可以通过小车受到的控制力，把系统状态控制到需要的位置。

六、LQR控制器

LQR 是线性二次型调节器，用来寻找一个状态反馈增益矩阵 (K)。

它优化的目标可以写成：

$$[J = \int_0^{\infty} \left(\mathbf{x}^T Q \mathbf{x} + u^T R u \right) dt]$$

其中：

- (Q)：状态误差权重；
- (R)：控制量权重；
- (J)：需要最小化的总代价。

直观理解：

- 增大 (Q)：更重视控制误差，控制会更积极；
- 增大 (R)：更重视节省控制力，控制会更温和。

两者存在折中：响应越快、误差越小，通常需要越大的电机力矩。

第一组权重

开始设置：

```
Q = C' * C;  
R = 1;  
K = lqr(A,B,Q,R);
```

此时：

$$[Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}]$$

这里：

- $(Q(1,1))$ 是小车位置的权重；
- $(Q(3,3))$ 是摆杆角度的权重。

得到：

$$[K = \begin{bmatrix} -1.0000 & -1.6567 & 18.6854 & 3.4594 \end{bmatrix}]$$

这个控制器可以稳定系统，但是响应不够快，而且小车最终没有到达指定位置，甚至移动方向也不正确。

调整权重

将位置和角度权重提高：

```
Q(1,1) = 5000;
Q(3,3) = 100;
R = 1;

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

此时：

$$[Q = \begin{bmatrix} 5000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}]$$

得到：

$$[K = \begin{bmatrix} -70.7107 & -37.8345 & 105.5298 & 20.9238 \end{bmatrix}]$$

控制律就是：

$$[u = -K\mathbf{x}]$$

也就是：

$$[u = -(-70.7107x - 37.8345\dot{x} + 105.5298\phi + 20.9238\dot{\phi})]$$

增大 (Q) 中的位置和角度权重后：

- 小车位置响应更快；
- 摆杆偏转更小；
- 稳定时间缩短；
- 但控制力和能源消耗也会增加。

继续增大 (Q) 可能进一步减小误差，但会要求电机输出更大的力。

七、添加前置补偿

现在控制器已经满足瞬态响应要求，但仍然存在稳态位置误差。

全状态反馈的基本形式是：

$$[u = -K\mathbf{x}]$$

它天然是把状态调节到零，并不会自动保证小车跟踪 (0.2 m) 的位置指令。

因此在参考输入前加入前置补偿系数 $(\bar{N})$ ：

$$[u = -K\mathbf{x} + \bar{N}r]$$

其中：

- (r) ：目标位置；
- $(\bar{N})$ ：前置补偿系数。

MATLAB 中计算得到：

$$[\bar{N} = -70.7107]$$

加入前置补偿后，小车能够稳定在目标位置附近，同时满足：

- 上升时间要求；
- 稳定时间要求；
- 摆杆最大偏角要求；
- 稳态误差要求。

前置补偿的缺点

$(\bar{N})$ 是根据数学模型计算的，而且它位于反馈环之外。

因此：

- 如果模型准确，补偿效果很好；
- 如果实际质量、摩擦或电机参数与模型不同，仍可能出现稳态误差；
- 它也不能很好地抵抗未知的持续干扰。

可以加入积分控制来消除由模型误差和恒定扰动引起的稳态误差。

两种方法可以组合：

$$[u = -K\mathbf{x} + K_i \int (r - y) dt + \bar{N}r]$$

前置补偿根据模型提前给出大致正确的控制量，积分项再负责消除残余误差。

八、基于观测器的控制

前面的 LQR 假设四个状态全部可以测量：

$$[x, \quad \dot{x}, \quad \phi, \quad \dot{\phi}]$$

但实际机器人可能只有：

- 编码器测量位置；
- IMU 测量角度或角速度。

有些状态不能直接测量，因此需要状态观测器估算它们。

可观性

可观性矩阵为：

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

MATLAB 代码：

```
ob = obsv(A,C);
observability = rank(ob)
```

结果：

```
observability = 4
```

因此系统完全可观，可以根据系统输入和测量输出估算全部四个状态。

需要注意：如果只能测量摆杆角度，而没有小车位置，则这个系统不能完全估计全部状态。

九、状态观测器原理

观测器方程为：

$$\dot{\hat{\mathbf{x}}} = A\hat{\mathbf{x}} + Bu + L(\mathbf{y} - \hat{\mathbf{y}})$$

其中：

- $\mathbf{x}$ ：真实状态；
- $\hat{\mathbf{x}}$ ：估计状态；
- $\mathbf{y}$ ：传感器实际输出；
- $\hat{\mathbf{y}} = C\hat{\mathbf{x}}$ ：估计输出；
- L ：观测器增益。

关键部分是：

$$L(\mathbf{y} - \hat{\mathbf{y}})$$

它根据“实际测量值与估计值之间的差别”不断修正状态估计。

定义估计误差：

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$$

误差动力学为：

$$\dot{\mathbf{e}} = (A - LC)\mathbf{e}$$

只要 $(A - LC)$ 的所有极点都在左半平面，估计误差就会逐渐趋近于零：

$$\hat{\mathbf{x}} \rightarrow \mathbf{x}$$

十、选择观测器极点

LQR闭环系统的极点为：

$$[-8.4910 \pm j7.9283] \quad [-4.7592 \pm j0.8309]$$

最慢极点的实部是 (-4.7592) 。

一般让观测器比控制系统快约 4 ~ 10 倍，所以教程选择：

```
P = [-40 -41 -42 -43];
L = place(A',C',P)';
```

得到观测器增益 (L)。

观测器不能设置得无限快，因为观测器带宽越高，对传感器噪声也越敏感。

这对实际机器人尤其重要：IMU 本身存在振动噪声，观测器过快可能把噪声放大到电机输出中。

十一、控制器与观测器组合

实际控制时，不再使用全部真实状态，而使用观测器估计的状态：

[$u = -K\hat{\mathbf{x}} + \bar{N}r$]

工作过程如下：



完整系统同时包含：

- LQR状态反馈；
- 参考输入前置补偿；
- 状态观测器。

仿真结果与能够测量全部状态时几乎一致，满足设计要求。

十二、对应到两轮平衡机器人

这个小车倒立摆模型可以对应到两轮平衡机器人：

教程中的变量	两轮平衡机器人中的含义
(x)	车轮累计位置
($\dot{x}$)	车轮前进速度
(ϕ)	机器人机身倾角
($\dot{\phi}$)	IMU测得的俯仰角速度

教程中的变量	两轮平衡机器人中的含义
(u)	左右电机共同输出的力矩
(K)	四个状态对应的反馈增益

因此平衡机器人的 LQR 控制可以写成：

$$[u = -(k_{xx} + k_v \dot{x} + k_{\phi} \phi + k_{\omega} \dot{\phi})]$$

它和你之前看的“速度 PI + 直立 PD”串级控制很相似，只是组织方式不同：

$$[\text{LQR输出} = \text{位置反馈} + \text{速度反馈} + \text{角度反馈} + \text{角速度反馈}]$$

LQR根据模型和 (Q、R) 权重统一计算这四个增益；串级 PID 则分别调整速度环和直立环参数。