**LQR从入门到**

**0 前言**

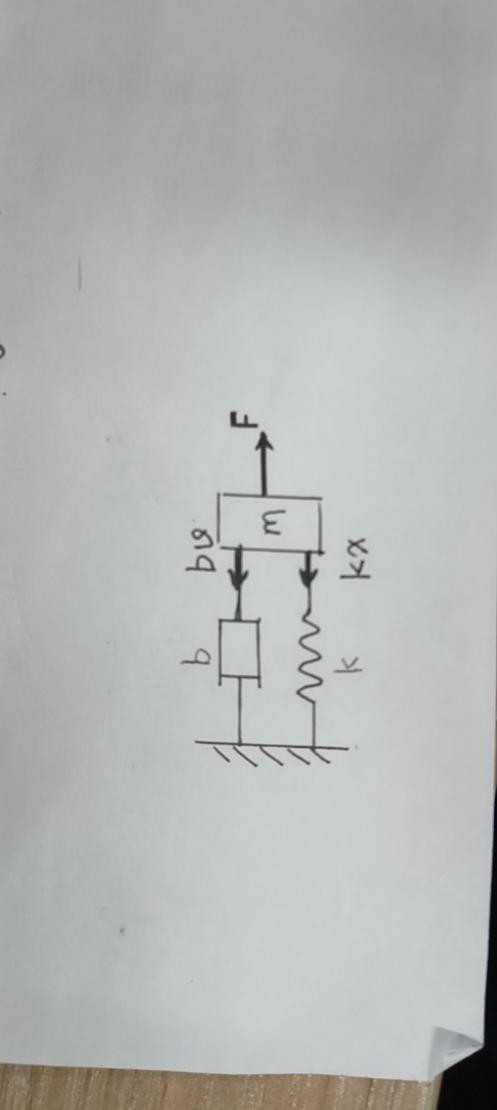
请确保您在阅读此教程前，已学过《代数与几何》、《高等数学》。

*本文用了较大篇幅进行数学推导与讲解，实际应用从第5部分起叙述。*

**[关注DR\_CAN喵，关注DR\_CAN谢谢喵](https://space.bilibili.com/230105574)**

**1 动力学建模与状态空间方程**

我们首先来看看这个朴实无华的弹簧系统：



(上面那个b是阻尼器，阻力与速度成正比，类似弹簧阻力与伸长量成正比)

我们可以列出动力学方程：， 也就是

(1)

再加上***x***与***v***的关系：  (2)

这里用了两个“参量”，也就是***x***和***v***来完全表达整个系统的状态，我们称之为**状态变量**。

当然，状态变量并不是唯一的。

同时，方程中还有一个外力***F***，这是你的**输入**。

上面这两个式子，我们可以用**矩阵形式**，表示成一阶微分方程：

(3)

*“不觉得这很酷吗？很符合我对高级控制算法的想象”*

类似地，对于我们所要控制的云台，我们大致可列出如下方程：

（此处以yaw轴举例，因为不用考虑重力的影响）

(4)

(5)

***Te***为输入转矩，b为电机阻尼，电机摩擦暂不考虑。

同样地，用矩阵表示以上方程：

(6)

其中***θ***和***ω***是状态变量，***Te***是输入。

观察(3)式与(6)式，我们可以发现其**共性**：

不妨记状态变量为x1,x2,…,xm，输入为u1,…,un，

记**X** = ，**U** = ，则有 (7)

实际上系统还有输出表达式 (8)

（为便于叙述，(8)式中的**DU**项略去；对上面的弹簧系统，**Y** = **[ *x* ]** ）

(7)与(8)称为系统的**状态空间方程**。

**2 收敛是啥？收敛去哪儿？收敛要干什么？**

试想一下，对于一个已经稳定的云台，它的位置（“位置”是以你的目标作为参考）、速度以及你给的电机输出都是0。

也就是，t→+∞时，有**X** = **0**, **U** = **0**。

先考虑系统的**可控性**，即是否能通过**U**来影响所有的xi：

系统可控判据：（不给出证明） (9)

系统**可观测性**这里不展开叙述。

接下来是系统**收敛**的条件：**A**的所有**特征值实部**为负。

系统**不产生振荡**的条件：**A**的所有**特征值虚部**为0。

**请牢记这个结论！！**

*以下为数学推导过程，需要电路or复变or自控理论基础，可先跳过。当然也可以观看DR\_CAN的《现代控制理论》等视频进行了解。*

对于一个系统，令其传递函数

其中F(s)解析，此时系统有极点s1,…,sn。对其进行Laplace逆变换后可得

P(t)为关于t的多项式。当t→∞，h(t) = 0当且仅当Re(s1),…,Re(sn)均小于0。并且由于si为复数时可表示为，此时 ，可知**虚部会引起系统振荡**，且**绝对值越大振荡频率越大**。

对于状态空间方程，进行Laplace变换可得到

消去X(s)，得到传递函数

很明显，**H**(s)的极点就是**A的特征值**，因此系统收敛的条件即**A的特征值实部为负**，不振荡的条件即**A的特征值虚部为0**。

此结论也可从以下内容理解：

**状态矩阵A的特征值与系统平衡点的关系**（可以理解为A的特征值对系统收敛性的影响，以两个状态变量为例）：

我们令 , , 令A的特征值为k1, k2。

k1, k2均为实数时：

k1, k2均为负数时，平衡点为稳定平衡点；

k1, k2一负一正时，平衡点为鞍点；

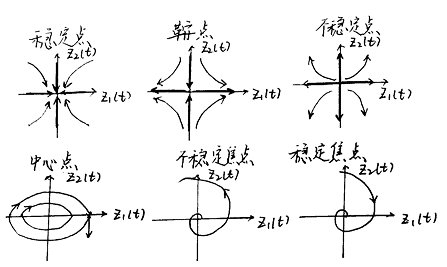
k1, k2均为正数时，平衡点为不稳定点。

k1, k2均为复数时：

k1, k2均为纯虚数时，平衡点为中心点；

k1, k2均实部大于0时，平衡点为不稳定焦点；

k1, k2均实部小于0时，平衡点为稳定焦点。



**3 由反馈设计控制器（极点配置）**

事实上，有很多系统是无法“自然收敛”的，比如经典的系统**一阶倒立摆**，这就需要我们去外加输入**U**来控制。

对于一个完全可观测的系统，我们会想到**让输出与系统的状态相关**(也就是**闭环控制**)，即**U** = f(**X**)。

我们最容易想到的是**线性反馈**，于是有

**U** = -**kX** (14)

其中**k**是一个m\*n矩阵。

那么此时，（7）式变为 ，如果**特征值实部**为负，则系统收敛，所以我们要找到合适的k，使**特征值实部**为负。

**4 何为LQR控制算法**

我们现在面临这样的问题： **k**矩阵的参数该如何确定？

与此同时，还要考虑以下问题：系统将以什么“路径”收敛？是要不顾一切，以最快速度收敛？还是稳扎稳打，用比较小的能量消耗，达到既定的目标？

实际上，所有的x与u是**对立统一**的，稳定时都需要趋于0，但几乎不可能让所有量都以最快速度趋于0。我们需要进行一定的“权衡”：

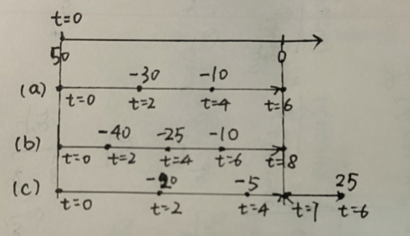
先来看以下表达式：

(15)

**系统收敛时此式为0，未收敛时恒为正数**。

如何理解这个表达式的作用？先来看这个例子：

季夏到来，Boris也迎来了他的体测。这回50m的老师想整点活，稍微修改了成绩标准：从起跑开始每过2s，老师把**学生到终点的距离的平方**加到成绩上（t=0时不计分），学生最终要停在终点线上，最终成绩越小越好。



Boris前面3个人的情况如下：小a以较快的速度到达了终点，他的成绩是1000。小b跑得稍慢一些，成绩为2325。旁边的小c跑得非常快，可是一下子没刹住车，跑过了终点线，最后用了差不多3秒才跑回来，最终成绩为1050。

可见，这位老师的特殊计分方式反映了学生的“控制”水平：既要快速到达终点，又要防止越过终点线造成的折返。

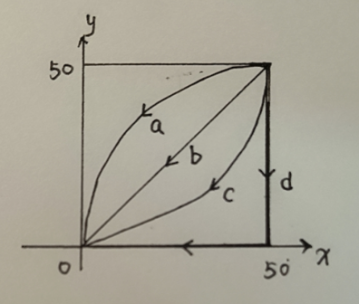
对应到控制理论中，就是**既要快速收敛，又要防止超调振荡**。因此，我们构造**代价函数**

**Q**和**R**是**正对角阵**（不建议部分元素给0，防止可能出现的无法求解情况），元素就是(15)中的**q与r**。

我们要做的，就是寻找**k**使得J的值最小。

这些q,r都是**我们自己给出的系数**，但我们要如何确定呢？接下来的例子可以帮助你理解：

Boris在体育选课抽中了“控制理论中的体育基础”，这天老师又更改了规则：场地为一个50m\*50m的正方形，起点与终点分别位于正方形的对角。老师同样每2s加一次成绩，但增加值为**x2+10y2**。同学们很快会想到了以下4种路径：



很快，4位同学就走上了跑道，他们的体能都差不多，速度大致相当。

最终他们以差不多的速度前进，但成绩却有明显差别。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学生 | 2s | 4s | 6s | 8s | 10s | 成绩 |
| a | (32, 44) | (16, 34) | (6, 18) | (2, 6) | (0, 0) | 35840 |
| b | (35.9, 35.9) | (21.7, 21.7) | (7.57, 7.57) | (0, 0) | (0, 0) | 19987 |
| c | (44, 32) | (34, 16) | (18, 6) | (6, 2) | (0, 0) | 16652 |
| d | (50, 30) | (50, 10) | (40, 0) | (20, 0) | (0, 0) | 17000 |

为什么c的成绩这么好？实际上，c**兼顾了x与y方向距离的减小**，但他的“策略”主要还是先减小y，因为老师成绩的判断标准中，**y的系数比x更大**。

也就是说，老师把y方向的“成绩系数”增大，能让学生尽可能地按照“先减y后减x”的轨迹行动，但两者也做到了兼顾。 *“抓主要矛盾”*

这是否与我们控制的系统很类似？例如云台的yaw轴，我们要让角度尽可能快地到达预定目标，同时也要让速度在可控范围内（防止超调）。至于耗能方面，则显得不那么重要。因此，在确定**Q**和**R**的时候，对应**期望快速收敛的状态变量**，对应的系数可以适当给大一些（实际应用中一般是**2~4个数量级**）。 *“极限犹可突破，至臻亦不可止！”*

以英雄yaw轴为例，笔者令**Q** = diag[40000, 200]，**R** = [1]。

现在，我们有了以下4个矩阵：**A**, **B**(系统建模得到), **Q**, **R**(自己给的)，利用它们可以求出所需的**k**。我们直接使用**Matlab**进行求解（~~手算二阶都花了我一节马原课）~~：

>> lqr(A, B, Q, R)

ans =

200.0000 2.8184

这里的200.0000和2,8184就是k1和k2。**好了，您已经学会LQR了，接下来请调好这个云台吧！**

**5 代码分析与参数获取**

有句话说得好，**理想臣服实践，勤恳铸就巅峰**。接下来讲讲LQR算法的实际应用。

在单片机中，我们仅需要提供**k**的各参数。可使用以下代码：

void LQR\_Calc(lqr\* obj){

obj->x1\_err = obj->x1 - obj->x1\_ref;

float i\_ref = -(obj->x1\_err \* obj->config.k1 + obj->x2 \* obj->config.k2) / 0.741;

obj->output = 1111 \* i\_ref - 236;

    if(obj->output > obj->config.outputmax) obj->output = obj->config.outputmax;

    else if(obj->output < -obj->config.outputmax) obj->output = -obj->config.outputmax;

}

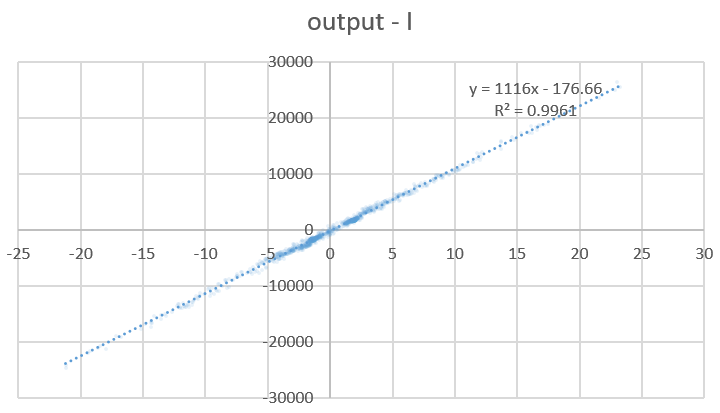
第1句是将传入的角度与预期相减，得到角度误差作为状态空间方程定义的x1(即代码中**x1\_err**)。接下来的几行代码都是确定电机的输出，因为我们无法直接读取或给出电机转矩的值。

下面是代码中数值常量的含义：

通过LQR算法，我们可以获得所需要的转矩***Te***，而我们知道**转矩与电机电流是线性关系**。在6020的手册上，可得知**转矩常数**为741 mN·m/A（其它电机的转矩参数同样可以在手册中得到），也就是 ***Te*** = 0.741I。

然而，给6020发送的CAN信号值（output）控制的是电压而非电流。不过可以测得output与电流之间的关系：让电机保持运行（期间静止与运动均持续一段时间），在**Excel散点图**中绘制趋势线，大致得到如下关系：

output = 1111 I – 236 (17)



（绘制趋势线时已去除残差较大的点；此关系进行了多组测量，以上为其中一组）

对于LQR的使用，有如下代码：

LQR\_SetConfig(&yaw\_controller\_config.lqr\_config, 200, 11.25, 30000);

此函数传入了参数k1,k2与输出上限。k1,k2的计算在Matlab中代码如下：

>> J = 0.05 %云台转动惯量

J =

0.0500

>> b = 0.01 %电机阻尼

b =

0.0100

>> A = [0 1; 0 -b/J]

A =

0 1.0000

0 -0.2000

>> B = [0;1/J]

B =

0

20

>> Q = diag([40000, 200])

Q =

40000 0

0 200

>> R = [1]

R =

1

>> lqr(A,B,Q,R)

ans =

200.0000 14.8224

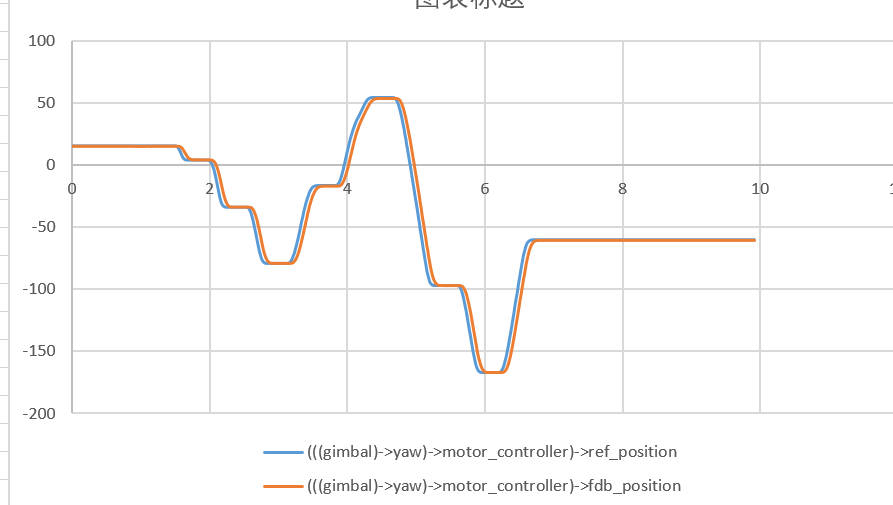
其中的**A**与**B**矩阵，对yaw轴我们需要知道云台的转动惯量***J***与电机阻尼b，***J***可通过**Solidworks**计算（不需要特别精确，因为实际系统也会有各种无法估计的误差），而b给0.01就行（求**k**的时候发现b从0.01给到10都没什么区别）。

至于**Q**和**R**，凭感觉给即可。如果发现云台出现振荡，可适当将q1减小，或直接对k1进行操作。

q1要大于q2，否则会振荡。（经验之谈）

注意，**在传入角度和角速度的时候**，一定要**输入弧度制**（否则相当于k变大了57倍）！

**6 实际效果**

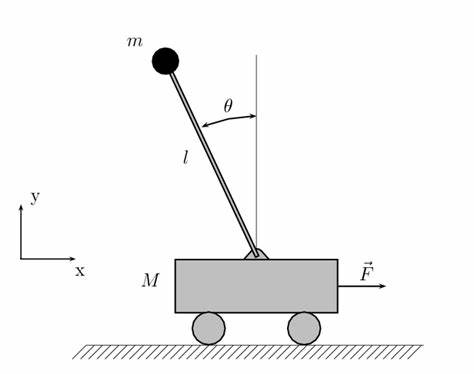


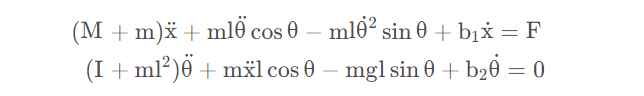
(英雄yaw轴响应效果)

可以看到LQR比起PID，参数确定过程中**减少了盲目性**，能通过已有数据计算出增益系数。同时，对于多输入输出系统如一阶倒立摆、平衡车，LQR能更方便地控制多个状态变量但这就要求我们能够**获得相关物理量**，并且**正确系统建模**。*此处@机械组*

**7 从建模到仿真——倒立摆建模**

1. **建立倒立摆物理模型**





M – 小车质量；

m – 摆杆质量；

***x*** – 小车位移坐标；

l – 杆长一半；

***θ*** – 摆杆与竖直向上夹角（注意：**顺时针为正，和图片相反**）；

b1 – 小车与地面摩擦系数，与速度成正比；

b2 – 摆杆与小车连接处阻尼系数，与角速度成正比；

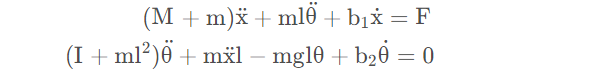
***F*** – 施加在小车上的外力；

***g*** – 重力加速度；

***I*** – 摆杆的转动惯量。

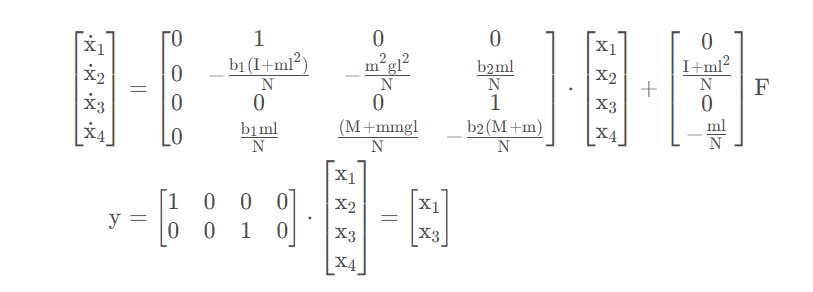
1. **对非线性系统进行线性化（这点极为关键，需要补充系统线性化相关知识）**

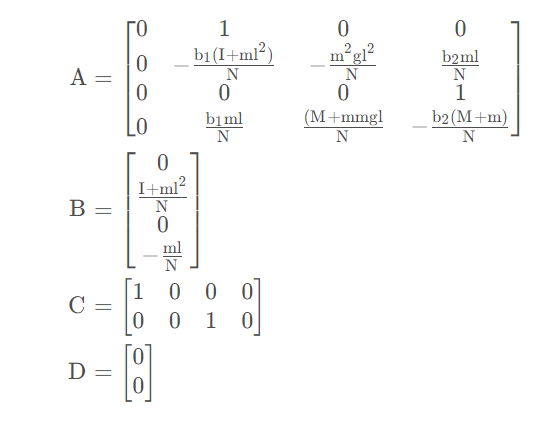
假设角度变化***Δθ*** 极小，则cos***θ*** ≈ 1 , sin***θ*** ≈ θ



1. **建立状态空间**







1. **matlab计算lqr**

源码见附件《dao\_li\_bai\_LQR\_cunting》

**五、Matlab进行Simulink仿真**

见附件《dao\_li\_bai\_LQR\_simulink》

参考：**[【Advanced控制理论】8\_LQR 控制器\_状态空间系统Matlab/Simulink建模分析](https://www.bilibili.com/video/BV1RW411q7FD/?spm_id_from=333.999.0.0)**

1. **Matlab进行Simscape Multibody仿真（可视化建模）**

见附件《dao\_li\_bai\_LQR\_simscape\_multibody》

参考：**[用Simscape实现三维物理仿真（四）——用PID控制倒立摆系统](https://readair.blog.csdn.net/article/details/104702335)**