

原 四轴PID讲解

2015年04月19日 19:44:24

Finding_Nemo

阅读量: 41038

标签:

四轴PID

PID算法

四轴串级PID

四轴起飞飘

PID整定

更多

个人分类: 四旋翼飞行器

51

12

插播一条DJI招聘信息【常年有效】：包括控制算法、软件功能开发、SDK、嵌入式开发、GNSS接收机设计、测试开发（包括CI持续集成）。请各位对空中机器人，地面机器人，教育机器人有兴趣的同学将简历发送至nemol1990@gmail.com

正文开始：这篇文章分为三个部分：

- PID原理普及
- 常用四轴的两种PID算法讲解(单环PID、串级PID)
- 如何做到垂直起飞、四轴飞行时为何会飘、如何做到脱控？

PID原理普及

1、对自动控制系统的基本要求：

稳、准、快：

稳定性（P和I降低系统稳定性，D提高系统稳定性）：在平衡状态下，系统受到某个干扰后，经过一段时间其被控量可以达到某一稳定状态；**准确性（P和I提高稳态精度，D无作用）**：系统处于稳态时，其稳态误差；**快速性（P和D提高响应速度，I降低响应速度）**：系统对动态响应的要求。一般由过渡时间的长短来衡量。

2、稳定性：当系统处于平衡状态时，受到某一干扰作用后，如果系统输出能够恢复到原来的稳态值，那么系统就是稳定的；否则，系统不稳定。

3、动态特性（暂态特性，由于系统惯性引起）：系统突加给定量（或者负载突然变化）时，其系统输出的动态响应曲线。延迟时间、上升时间、峰值时间、超调量和振荡次数。

通常：上升时间和峰值时间用来评价系统的响应速度；

超调量用来评价系统的阻尼程度；

调节时间同时反应响应速度和阻尼程度；

4、稳态特性：在参考信号输出下，经过无穷时间，其系统输出与参考信号的误差。影响因素：系统结构、参数和输入量的形式等

5、比例（P）控制规律：具有P控制的系统，其稳态误差可通过P控制器的增益Kp来调整：Kp越大，稳态误差越小；反之，稳态误差越大。但是Kp越大，稳定性会降低。

$$m(t) = K_p e(t)$$

由上式可知，控制器的输出m(t)与输入误差信号e(t)成比例关系，偏差减小的速度取决于比例系数Kp：Kp越大，偏差减小的越快，但是很容易引起振荡向通道中存在较大的时滞环节时；Kp减小，发生振荡的可能性小，但是调节速度变慢。单纯的P控制无法消除稳态误差，所以必须要引入积分I控制。参考输入信号，Kv为开环增益)

$$\lim_{t \rightarrow \infty} e(t) = \frac{R}{K_v}$$

当参考输入信号R不为0时，其稳态误差只能趋近于0，不能等于0。因为开环增益Kv不为0。

6、比例微分（PD）控制规律：可以反应输入信号的变化趋势，具有某种预见性，可为系统引进一个有效的早期修正信号，以增加系统的阻尼程度，而的稳定性。（tau为微分时间常数）

开发者调查

Python学习路线！

Office 365商业协作版 5折钜惠！

登录

注册

×

$$m(t) = K_p e(t) + K_p \tau \frac{de(t)}{dt}$$

如果系统中存在较大时滞的环节，则输出变化总是落后于当前误差的变化，解决的方法就是使抑制误差的作用变化“超前”，增强系统的稳定性。

7、**积分 (I) 控制规律**：由于采用了积分环节，若当前误差e(t)为0，则其输出信号m(t)有可能是一个不为0的常量。需要注意的是，引入积分环节，可以使得系统可以跟踪更高阶次的输入信号，以消除稳态误差。

$$m(t) = K_I \int_0^t e(t) dt$$

8、**比例积分 (PI) 控制规律**：在保证系统稳定的前提下，引入PI控制器可以提高它的稳态控制质量，消除其稳态误差。（TI为积分时间常数）

$$m(t) = K_p e(t) + \frac{K_p}{T_I} \int_0^t e(t) dt$$

积分调节可以消除静差，但有滞后现象，比例调节没有滞后现象，但存在静差。

PI调节就是综合P、I两种调节的优点，利用P调节快速抵消干扰的影响，同时利用I调节消除残差。

9、**比例积分微分 (PID) 控制规律**：除了积分环节提高了系统型别，微分环节提高了系统的动态性能。

$$m(t) = K_p e(t) + \frac{K_p}{T_I} \int_0^t e(t) dt + K_p \tau \frac{de(t)}{dt}$$

观察PID的公式可以发现：Kp乘以误差e(t)，用以消除当前误差；积分项系数Ki乘以误差e(t)的积分，用于消除历史误差积累，可以达到无差调节；微分以误差e(t)的微分，用于消除误差变化，也就是保证误差恒定不变。由此可见，P控制是一个调节系统中的核心，用于消除系统的当前误差，然后，I控制控制余留的静态误差而辅助存在，对于D控制，所占的权重最少，只是为了增强系统稳定性，增加系统阻尼程度，修改PI曲线使得超调更少而辅助存在。

10、P控制对系统性能的影响：

- 开环增益越大，稳态误差减小（无法消除，属于有差调节）
- 过渡时间缩短
- 稳定程度变差

11、I控制对系统性能的影响：

- 消除系统稳态误差（能够消除静态误差，属于无差调节）
- 稳定程度变差

12、D控制对系统性能的影响：

- 减小超调量
- 减小调节时间（与P控制相比较而言）
- 增强系统稳定性
- 增加系统阻尼程度

13、PD控制对系统性能的影响：

- 减小调节时间
- 减小超调量

• 增大系统阻尼，增强系统稳定性

开发者调查

Python学习路线！

Office 365商业协作版 5折钜惠！

登录

注册

×

- 增加高频干扰

51

14、PI控制对系统性能的影响：

12

- 提高系统型别，减少系统稳态误差
- 增强系统抗高频干扰能力
- 调节时间增大

15、P调节、I调节降低系统稳定性

D调节增强系统稳定性

所以PI调节器的P比P调节器的P要小一些，PD调节器的P比P调节器的P要大一些

16、位置式PID表达式（数字PID）：



$$P(n) = K_p \left[e(n) + \frac{T_s}{T_i} \sum_{i=0}^n e(i) + \frac{T_d}{T_s} (e(n) - e(n-1)) \right]$$

P(n)为第n次输出，e(n)为第n次偏差值，Ts为系统采用周期，Ti为积分时间常数，Td为微分时间常数

17、消除随机干扰的措施：

- 几个采样时刻的采样值求平均后代替本次的采样值
- 微分项的四点中心差分 $(e(n)-e(n-3)+3e(n-1)-3e(n-2))*1/(6Ts)$
- 矩形积分改为梯形积分 $\sum_{i=0}^n e(i) \rightarrow \sum_{i=0}^n [e(i) + e(i-1)]/2$

18、PID调试一般原则

- 在输出不振荡时，增大比例增益P
- 在输出不振荡时（能消除静态误差就行），减小积分时间常数Ti
- 在输出不振荡时，增大微分时间常数Td

19、描述比例Kp的性能：**比例带**。比例带就是Kp的倒数：比例带越大，Kp越小，无超调，稳态误差大，调节时间长；比例带越小，Kp越大，系统会有发散，稳态误差减小，调节时间缩短

20、描述积分Ki的性能：**积分时间常数Ti**。与积分系数Ki也是倒数关系：积分时间常数Ti越大，积分系数Ki越小，系统稳定性增加，但是调节速度变慢；数Ti越小，积分系数Ki越大，系统稳定性降低，甚至振荡发散。**无论增大还是减小积分时间常数Ti，被调量最后都没有静差。**

21、描述微分Kd的性能：**微分时间常数Td**。主要用于克服调节对象有较大的时滞。Td越大，微分作用越强，系统阻尼程度增加。

22、比例P调节作用：

系统一旦出现偏差，比例调节立即产生调节作用用以减少偏差

比例作用大，可以加快调节，减少调节时间，减少稳态误差

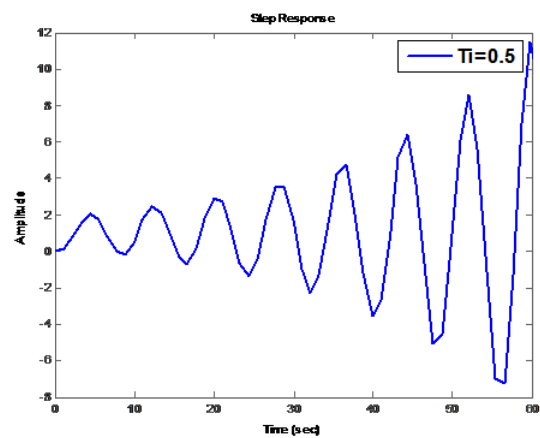
过大的比例作用，使系统的稳定性下降



广告

广告

广告



减少超调量，减少调节时间



×

- 超调趋势随着 K_p 增大、积分时间 T_i 减小而增大

26、积分分离的措施：（在系统启动，结束或者大幅度增减时，短时间系统输出会有很大偏差，造成PID运算的积分积累，致使控制量超过执行结构可能动作范围，引起系统较大的超调，甚至振荡）

- 当误差大于某个阈值时，采用PD控制，当误差在该阈值范围内，采用PID控制

注意：阈值的选取，过大，则达不到积分分离的目的；过小，则会导致无法进入积分区

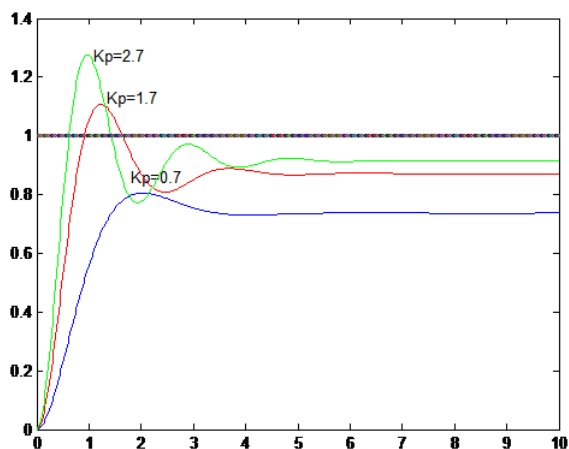
27、抗积分饱和的措施：

- 限制PI调节器的输出（这样有可能在正常操作中不能消除系统的余差）
- 积分分离法：误差在某个范围内开启积分调节（既不会积分饱和又能在小偏差时利用积分作用消除偏差）
- 遇限削弱积分法：调节器输出大于某个值后，只累加负误差（可避免控制量长时间停留在饱和区）

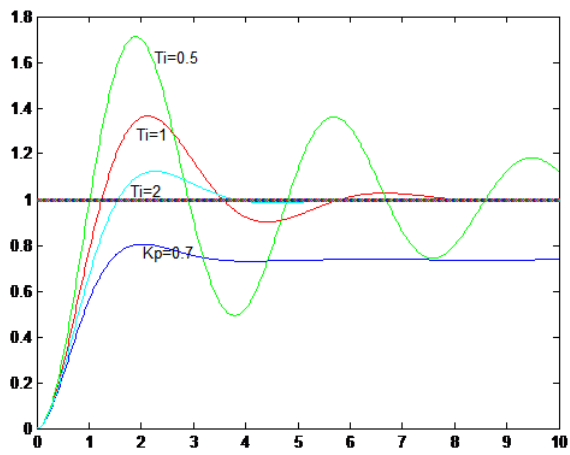
28、比例微分PD调节器

- 仍然属于有差调节
- 提高了系统的稳定性，**因为微分D的作用总是试图抑制被调量的振荡**，所以也增加了系统的阻尼程度
- 因为提高了系统的稳定性，所以可以适当加大 K_p
- **D只是辅助作用，主要还是P控制**

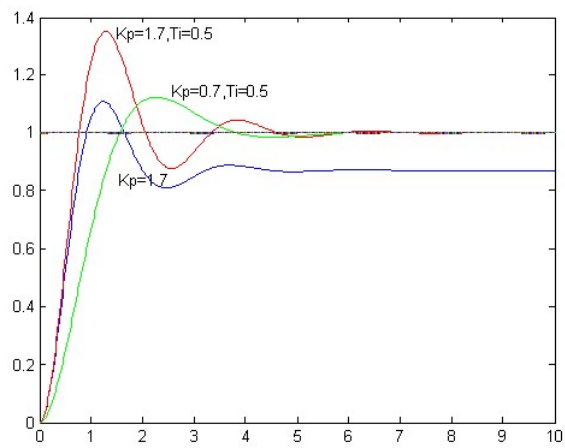
29、MATLAB仿真



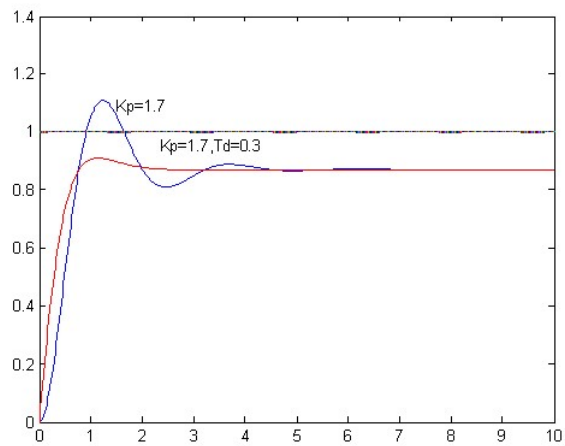
纯P调节（ K_p 大，稳态误差小，响应快，但超调大）



PI调节 (Ti小, 响应速度加快, 超调大, 系统振荡加剧)



PI调节 (在同样积分常数 T_i 下, 减小比例增益 K_p 可减小超调, 增加系统的稳定性)



PD调节 (引入微分项, 提高了响应速度, 增加了系统的稳定性但不能消除系统的余差)

现象：最终整定的系统，其调节效果应该是被调量波动小而平缓。在一个扰动过来之后，被调量的波动应该呈现“一大一小两个波”（波形高度差4:1）

注意2：如果看不到这种被调量的周期特征，那说明参数整定的很好。即满足快速性，也不会超调

整定积分时间：

方法：主调的作用是为了消除静态偏差，当比例作用整定好的时候，就需要逐渐加强积分作用（调小积分时间Ti或者增大积分项系数Ki），直到消除静态偏差。是说，积分作用只是辅助比例作用进行调节，它仅仅是为了消除静态偏差。

整定微分作用：

方法：逐渐加强微分作用（增加微分时间Td或者增加微分项系数Kd），直到PID输出毛刺过多

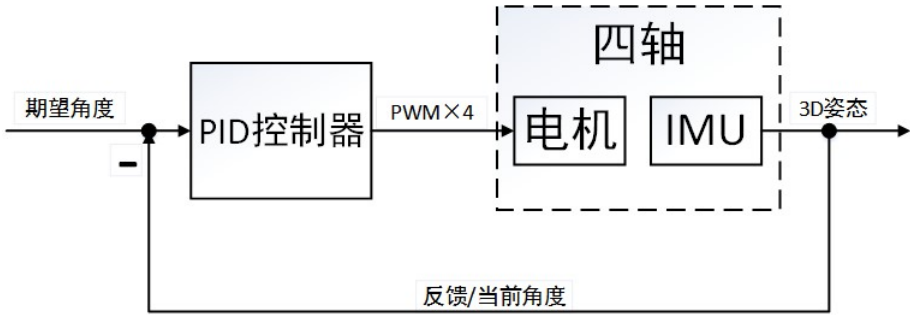
34、串级调节系统，一般而言，主调的比例弱，积分强，以消除静差；副调的比例强，积分弱，以消除干扰。但是不绝对！

35、不完全微分PID控制：微分控制对高频干扰非常敏感，为了避免在误差扰动突变时的微分控制的不足，在微分项通道前加一个低通滤波器可以使得其改善

36、微分先行PID控制：只对被调量进行微分，而不对给定值进行微分。这样的处理在改变给定值时，输出不会改变，被控量的变化通常比较缓和，适用于频繁升降的场合。

常用四轴的两种PID算法讲解(单环PID、串级PID)

这里主要讲解的PID算法属于一种线性控制器，这种控制器被广泛应用于四轴上。要控制四轴，显而易见的是控制它的角度，那么最简单，同时也的一种控制策略就是角度单环PID控制器，系统框图如图所示：



角度单环PID原理框图 [/blog.csdn.net/nemoli1990](http://blog.csdn.net/nemoli1990)

或许有些朋友看得懂框图，但是编程实现有一定困难，在这里笔者给出了伪代码：

```
当前角度误差      =      期望角度      - 当前角度

单环PID_P项       =      Kp              * 当前角度误差

当前角度误差积分及其积分限幅

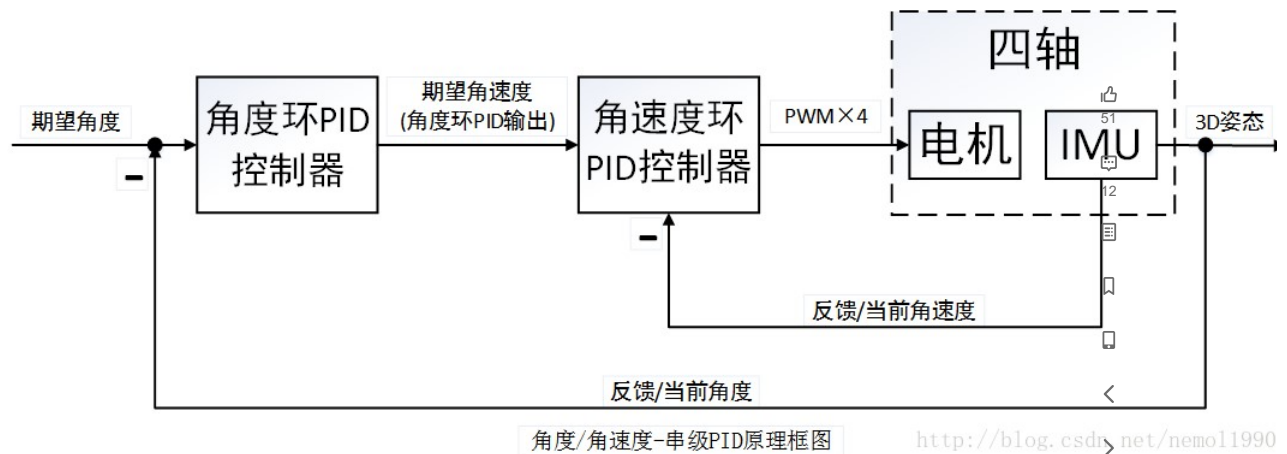
单环PID_I项       =      Ki              * 当前角度误差积分

当前角度的微分
(原理上为当前角度误差-上次角度误差,实际上角度的微分就是角速度,恰好由陀螺仪输出)

单环PID_D项       =      Kd              * 当前角速度的微分(直接用陀螺仪输出)

单环PID_输出      =      单环PID_P项 + 单环PID_I项 + 单环PID_D项
```

上述角度单环PID控制算法仅仅考虑了飞行器的角度信息，如果想增加飞行器的稳定性(增加阻尼)并提高它的控制品质，我们可以进一步的控制它的角速度/角速度-串级PID控制算法应运而生。在这里，相信大多数朋友已经初步了解了角度单环PID的原理，但是依旧无法理解串级PID究竟有什么不同。其实就是两个PID控制算法，只不过把他们串起来了(更精确的说是套起来)。那这么做有什么用？答案是，它增强了系统的抗干扰性(也就是增强稳定性)，因此器控制飞行器，它会比单个控制器控制更多的变量，使得飞行器的适应能力更强。为了更为清晰的讲解串级PID，这里笔者依旧画出串级PID的原理框图示：



同样，为了帮助一些朋友编程实现，给出串级PID伪代码：

```

当前角度误差    =    期望角度        -    当前角度
外环PID_P项      =    外环Kp          *    当前角度误差

当前角度误差积分及其积分限幅

外环PID_I项      =    外环Ki          *    当前角度误差积分
外环PID_输出     =    外环PID_P项    +    外环PID_I项

当前角速度误差  =    外环PID_输出    -    当前角速度(直接用陀螺仪输出)
内环PID_P项      =    内环Kp          *    当前角速度误差

当前角速度误差积分及其积分限幅

内环PID_I项      =    内环Ki          *    当前角速度误差积分

当前角速度的微分(本次角速度误差-上次角速度误差)

内环PID_D项      =    内环Kd          *    当前角速度的微分
内环PID_输出     =    内环PID_P项 + 内环PID_I项 + 内环PID_D项
  
```

关于如何整定单环PID与串级PID的问题，请原谅笔者的能力有限，无法给出标准而可靠的整定流程，这里我给出三个链接，第一个为阿莫论坛的一位同PID整定现象与思考，个人觉得参考价值很大；第二、三两个分别为APM与PX4的串级PID整定现象说明，大家可以参考他们的网页说明。

www.amobbs.com/thread-5554367-1-1.html

copter.ardupilot.com/wiki/configuration/tuning/

pixhawk.org/users/multicopter_pid_tuning

而笔者在整定串级PID时的经验则是：先整定内环PID，再整定外环P。

内环P：从小到大，拉动四轴越来越困难，越来越感觉到四轴在抵抗你的拉动；到比较大的数值时，四轴自己会高频震动，肉眼可见，此时拉扯它，它不动，过几秒后稳定；继续增大，不用加人为干扰，自己发散翻机。

特别注意：只有内环P的时候，四轴会缓慢的往一个方向下掉，这属于正常现象。这就是系统角速度静差。

内环I：前述PID原理可以看出，积分只是用来消除静差，因此积分项系数个人觉得没必要弄的很大，因为这样做会降低系统稳定性。从小到大，四轴会不动，不再往下掉；继续增加I的值，四轴会不稳定，拉扯一下会自己发散。

特别注意：增加I的值，四轴的定角度能力很强，拉动他比较困难，似乎像是在钉钉子一样，但是一旦有强干扰，它就会发散。这是由于积分项太大，拉速度快，给的补偿非常大，因此很难拉动，给人一种很稳定的错觉。

内环D：这里的微分项D为标准的PID原理下的微分项，即本次误差-上次误差。在角速度环中的微分就是角加速度，原本四轴的震动就比较强烈，引起较大，此时微分就更易引入噪声。因此一般在这里可以适当做一些滑动滤波或者IIR滤波。从小到大，飞机的性能没有多大改变，只是回中的时候更

较大，此时微分就更易引入噪声。因此一般在这里可以适当做一些滑动滤波或者IIR滤波。从小到大，飞机的性能没有多大改变，只是回中的时候更

较大，此时

微分就更易引入噪声。

因此一般在这里可以适当做一些滑动滤波或者IIR滤波。

从小到大，飞机的性能没有多大改变，只是回中的时候更

较大，此时微分就更易引入噪声。

因此一般在这里可以适当做一些滑动滤波或者IIR滤波。

从小到大，飞机的性能没有多大改变，只是回中的时候更

较大，此时微分就更易引入噪声。

因此一般在这里可以适当做一些滑动滤波或者IIR滤波。

续增加D的值，可以肉眼看到四轴在平衡位置高频震动(或者听到电机发出滋滋的声音)。前述已经说明D项属于辅助性项，因此如果机架的震动较大，DI加。

外环P：当内环PID全部整定完成后，飞机已经可以稳定在某一位置而不动了。此时内环P，从小到大，可以明显看到飞机从倾斜位置慢慢回中，用手拉手，它会慢慢回中，达到平衡位置；继续增大P的值，用遥控器给不同的角度给定，可以看到飞机跟踪的速度和响应越来越快；继续增加P的值，飞机变感，机动性能越来越强，有发散的趋势。

如何做到垂直起飞、四轴飞行时为何会飘、如何做到脱控？

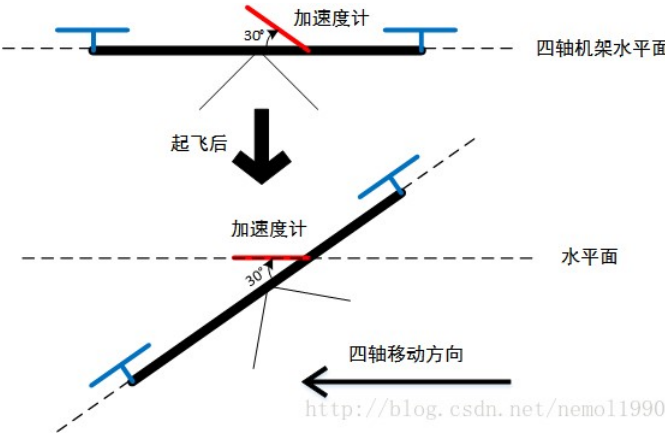
眼看这三个问题是三个不同的问题，其实就原理上讲，他们的原因在绝大多数情况下都是由于**加速度计**引起的。如果飞机可以垂直起飞，说明你的置地很水平，同时也说明你的PID控制算法参数找的不错，既然可以垂直起飞，那么飞行过程中，只要无风，四轴几乎就不会飘，自然而然就可以脱控。见，加速度计是个十分重要的器件。

在姿态解算中，或者说在惯性导航中，依靠的一个重要器件就是惯性器件，包括了加速度计和陀螺仪。陀螺仪的特性就是高频特性好，可以测量高动；而加速度计的低频特性好，可以测量低速的静态加速度。无论是何种算法(互补滤波、梯度下降、甚至是Kalman滤波器)，都离不开对当地重力加速分析。惯性导航利用的就是静态性能好的加速度计去补偿动态性能好的陀螺仪漂移特性，得到不飘并且高速的姿态跟踪算法，因此基于惯性器件的姿态计是老大，它说了算。

下面，我给大伙推理一下四轴如何平稳飞行的思路，欢迎各位批评指出：-)

首先，为了让四轴平稳的悬停或飞行在半空中，四个电机必须提供准确的力矩->假设力矩与电机PWM输出呈线性关系，也就是必须提供准确的4路PWM由遥控器输入(期望角度)、PID算法及其参数和姿态解算输出(当前角度)组成，假设遥控器输入不变(类似脱控)、PID算法及其参数也较为准确(PID不精确，但只要在一个合理的范围内，控制品质差不了多少)，也就是姿态解算的输出必须是十分准确的，可以真实反应飞行器的实际角度->姿态解算的结果和陀螺仪给出，根据前述惯性导航的描述，加速度计补偿陀螺仪，因此要得到精确的姿态解算结果，务必要求加速度输出精确的重力加速度g->这里仍行，因此忽略掉额外的线性加速度(事实证明，在四轴强机动飞行过程中，线性加速度必须要考虑并消除)，假设加速度计输出重力加速度g，这个重力加速度分“精确”。

我先写到这里，总结一下：精准力矩->精准PWM->精准姿态->加速度计输出“精确”重力加速度g。这里的“精确”打了引号，意思不是说加速度的性能输出精确的当地加速度g，而是说它能够准确反应机架的角度。为了达到悬停、平稳的飞行效果，控制算法输出的PWM会让加速度计输出的重力加速度内的分量就可能少，也就是说：**PID控制算法控制的不是机架水平，而是加速度计水平，PID不知道机架是什么东西，它只认加速度计，它的使命就是让平。**我现在假设加速度计与机架存在某个角度，比如右倾30°，四轴主视图如图所示。



上图中，加速度计(红线)与四轴机架的水平面(虚线)呈30°。起飞后，PID控制算法会尝试将加速度计调整至水平位置，因此四轴就会往图中左边飘，倾斜30°。这就是为什么飞机无法垂直起飞，或者飞行过程中往一个方向飘的原因：加速度计和机架没有水平。**因此在加速度计的机械安装时，尽量保证加速度计水平。**如果有些朋友已经将加速度计固定在飞控板上，可以通过遥控器的通道微调功能设置悬停时的期望角度，软件上校正这种机械不水平。

除了上述讨论的加速度计安装水平问题，也需要对加速度计进行零偏置校正，具体的方法叫做6位置标定法：即将加速度计沿着6个方向放置，分别测出加速度计g在6个方向上的最大输出值，然后取平均，得到圆球的中心点(这里假设g投影为球，实际上为椭球，需要进行最小二乘法拟合求三轴标定系数)。

最后提一点，如果加速度计和遥控器均做了调整，飞机可以做到垂直起飞，并且飞行效果还行，但是飞机的回中速度较慢，感觉就像是在抬轿子一述：悬停时，猛往一个方向打摇杆后立即放手，飞机会往摇杆方向走很远才停下。这是因为飞机过于稳定，也就是**内环的作用过强或者外环作用过弱**，导致是降低内环P或者加大外环P。

以上内容，均为笔者学习过程中的体会和想法，难免有错误之处，还请诸位批评指出，共同学习进步。

APM飞控学习之路：3 APM系统介绍与开发环境搭建 - 岳小飞Fly的博客

“工欲善其事，必先利其器”。在进行无人机飞控开发时，选择一个合适的软硬件平台以及IDE是十分重...

3.7万

来自：岳小飞Fly的博客



kiti1013

关注

28篇文章



蓝白天际线

关注

385篇文章



神秘路人甲

关注

17篇文章

pid基础介绍 - IMBeGooD的博客

这是网上看到的，讲的不错生动 总所周知，PID算法是个很经典的东西。而做自平衡小车，飞行器PID...

7526

来自：IMBeGooD的博客

PID 简介 - asszz的博客

PID是比例、积分、微分的简称，PID控制的难点不是编程，而是控制器的参数整定。参数整定的关键...

2500

来自：asszz的博客

PID算法详解（1） - HandsomeHong的博客

一、首先介绍一下PID名字的由来： P：Proportion（比例），就是输入偏差乘以一个常数。I：Integ...

3.6万

来自：HandsomeHong的博客

一个退役操盘手肺腑之言，写给无数正在亏钱的散户

集升商贸·顶新

手把手带你玩串级PID仿真实验 - Rick_Grimes的博客

写在前面的话：由于在设计四旋翼飞行器的时候用到了串级PID控制，因此花了一点时间搞了搞，真...

3147

来自：Rick_Grimes的博客

PID控制原理详解（一） - 花山熏

PID的理解 关于理解PID控制算法最典型的一个例子就是一个漏水的水缸的问题。网上有很多讲解...

1.4万

来自：花山熏

PID基本原理 - 魂淡林的博客

1.PID基本原理

350

来自：魂淡林的博客

飞控中的PID算法理解 - zouxu634866的博客

前言：这篇博客是对一位博主的博客的转载，地址是：https://www.cnblogs.com/kinson/p/9651434.ht...

98

来自：zouxu634866的博客

开源四轴飞行器CC3D的稳定模式和PID算法简介 - 小华同学的学习笔记

本文主要介绍一下CC3D的两大飞行模式的原理：1. Rate 2. Attitude. 以及PID的基本原理。至于AxisL...

5674

来自：小华同学的学习笔记



马上公测的网游

百度广告

四轴飞控源码 - dsy

PID算法程序 基于四轴飞行器 CPU: STM32F103CB2.4G: NRF24L01电子罗盘: HMC5883陀螺仪+加速度计: MPU-6050 固定的传感...

飞控控制方法学习 - InfiniteYuan

飞控控制方法学习“飞控”是指无人机的飞行控制器，用于自动化保持飞行器处于一个特定的状态（悬停...

5644

来自：InfiniteYuan

机器人巡线算法优化方案 - qq_39188087的博客

引言 大赛机器人的运行环境是大赛的场地,比赛时场地的情况将非常复杂,因此,机器人必须知道自己当...

4559

来自：qq_39188087的博客

基于STM32的四轴飞行器项目总结 - 无知人生,记录点滴

STM32F103RCT6 32位处理器 STM32F103RCT6 程序存储器为256K, STM32F103C8T6 程序存储...

797

来自：无知人生,记录点滴

四轴PID控制算法 - WELLSTONE_CHENG

四轴控制原理 四轴飞行器的螺旋桨与空气发生相对运动，产生了向上的升力，当升力大于四轴的重力...

1368

来自：WELLSTONE_CHENG



激光测距传感器

百度广告

第一版小四轴总结 - kiti1013的专栏

0.第一次打板 没有注意重心问题，由于720电机路无力，在PID参数上出现了较严重问题，最后飞行效...

673

来自：kiti1013的专栏

四轴自适应控制算法的一些尝试开源我的山猫飞控和梯度在线辨识自适应等算法一(转) - 蓝精...

1825

来自：蓝精灵的专栏

本文的最主要目的在于抛砖引玉，同时该论坛真的是非常好的一个论坛，没有这个论坛，没有那么多这...

开发者调查

Python学习路线!

Office 365商业协作版 5折钜惠!

登录

注册

×

匿名带你从零开始做**四轴-第二期--四轴的硬件组成** - weixin_35703797的博客 2305
四轴飞行器的硬件，一般分为两大类，一类是使用空心杯电机的微型四轴，一种是使用无刷电机的四... 来自: weixin_35703797的...

烤四轴 - qinxiaxiao的博客 359
烤四轴方法，先盗图一张先调试内环后外环（不明白自行搜索串级PID）1、在飞机的起飞油门基础上... 来自: qinxiaxiao的博客

匿名**四轴上位机波形绘制软件所需的下位机报文协议编写** - jirryzhang的博客 1561
extern float ZGyroModuleAngle;//Yaw轴角度 float ZGyroModuleAngleMAX; float ZGyroModuleAngle... 来自: jirryzhang的博客



少儿编程,到底该学什么
百度广告

四旋翼飞行器Quadrotor飞控之 PID调节（参考APM程序） - super_mice的专栏 4.1万
做四轴也有一段时间了，最近一直在做PID方面的工作。现在四轴基本可以实现室内比较稳定的飞行，... 来自: super_mice的专栏

两种四轴PID讲解 - Mr.A的专栏 1357
参考资料：爱无人机论坛 单级PID：PID算法属于一种线性控制器，这种控制器被广泛应用于四轴上... 来自: Mr.A的专栏

我的四轴专用PID参数整定方法及原理 - 蓝精灵的专栏 2374
给四轴调了好久的PID，总算是调好了，现分享PID参数整定的心得给大家，还请大家喷的时候手下留... 来自: 蓝精灵的专栏

四轴之第一炸 - Andrew的博客 442
三天前买花了240块钱在咸鱼上买了一个二手f330四轴。昨天把他和我的遥控器调试完之后，准备今天... 来自: Andrew的博客

5种常用的四轴飞行器PID算法讲解集合 - 瘦子 1.1万
在某莫上看到的，makeflyeasy大神的作品就分享给大家科普一下先分享一些算法的效果 三角函数直... 来自: 瘦子

一个退役操盘手肺腑之言，写给无数正在亏钱的散户！
陕西信息科技 · 熾焱

四旋翼无人机PID控制 - 哈哈哈哈哈的博客 2850
一、建立四旋翼无人机动力学方程 二、确定PID控制律 1、姿态控制回路 2、位置... 来自: 哈哈哈哈哈的博客

Pixhawk参数调整 - 大话开源飞控 6789
Pixhawk copter固件的默认参数是为3DR QUAD COPTER设计的。如果你想获得一个最佳的飞行表现... 来自: 大话开源飞控

PID参数的调整方法 - u010988991的博客 3824
1.PID算法演示软件1 （1）P太小，实际值和设定值时间有很大的偏差，实际值偏小或偏大。长时间回... 来自: u010988991的博客

PID代码及注释 - kaka的博客 5847
/*****所谓PID即为P（比例）+I（积分）+D（微分*****/ ***(1)P... 来自: kaka的博客

px4官网调参指南 多旋翼无人机PID调参指南 - chaibubble 5363
译文部分：多旋翼无人机PID调参指南不用碳纤维或增强碳纤维桨调整多轴，不使用损坏的桨片。... 来自: chaibubble

老中医说：男人多吃它，性生活时间延长5倍
新方向 · 熾焱

小四轴编程入门教程 - qq_15063463的博客 80
小四轴编程入门教程之一：陀螺仪和加速度计 在小四轴中，陀螺仪是一种用于测量小四轴旋转速度的... 来自: qq_15063463的博客

四轴飞行器原理与双闭环PID控制 - lmonkey000的博客 5167
相关专业术语：IMU—(Inertial measurement unit, 简称 IMU) 是测量物体三轴姿态角(或角速率)以及... 来自: lmonkey000的博客

下载

四轴PID学习笔记 - mercer_of_cleaner 06-05
四轴PID学习笔记

基于树莓派的四轴[仅基本功能] - Nonikka的小方舟 3406

开发者调查

Python学习路线!

Office 365商业协作版 5折钜惠!

 来自: Nonikka的小方舟

51
12
注册



注册

下载 ANO-MR-F1工程

ANO-MR-F1 四轴 Keil4 工程 PID算法.....

04-21

51

12

<

>

几乎没人知道！微信新出的这个赚钱功能简直赚疯了！

娜娜彩·熾熾

四轴飞行器的串级PID参数整定经验 - RhythmWANG 的博客

1403

串级PID即将两个PID控制器按照串联的方式连接起来，前一个的输出作为后一个的输入两者共同控制... 来自: RhythmWANG 的博客

飞控算法之PID - 廖旭 廊坊师范学院信息技术提高班第十一期

2382

本篇文章中，我们将粗略的了解一下PID过程控制算法，涉及到一下概念和几个分立算法。PID是什么... 来自: 廖旭 廊坊师范学院信...

PID控制器开发笔记之七：微分先行PID控制器的实现 - foxclever的专栏

461

前面已经实现了各种的PID算法，然而在某些给定值频繁且大幅变化的场合，微分项常常会引起... 来自: foxclever的专栏

PID控制器开发笔记之六：不完全微分PID控制器的实现 - foxclever的专栏

560

从PID控制的基本原理我们知道，微分信号的引入可改善系统的动态特性，但也存在一个问题，... 来自: foxclever的专栏

PID教程 - 张云博的专栏

3.7万

PID教程 介绍 本教程将向您展示了比例每个比例项(P)的特点，积分项(I)和微分项(D)控制，以及如... 来自: 张云博的专栏

高档海参怎么挑？揭秘黑商暴利赚差价，几乎没有营养价值

帝农·熾熾

Pid控制算法-PID算法的离散化 - 三月，有人呼唤你的名字

1.3万

PID控制算法的C++语言实现 二 PID算法的离散化 上一节中，我论述了PID算法的基本形式，并对其... 来自: 三月，有人呼唤你的...

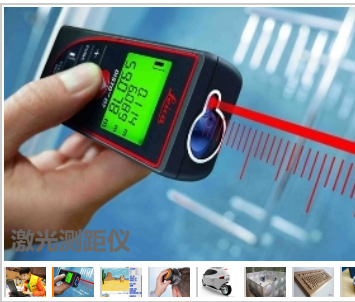


Finding_Nemo

关注

原创	粉丝	喜欢	评论
21	583	311	168

等级: 博客 4 访问: 19万+
积分: 1802 排名: 3万+



最新文章

uCOS-III学习笔记

Kalman论文笔记

四轴加速度计滤波

基于四元数的姿态解算算法图解

四元数姿态的梯度下降法推导和解读

个人分类

开发者调查

Python学习路线！

Office 365商业协作版 5折优惠！

登录

注册

×

学习笔记 12篇

一些好玩的小程序 3篇

归档

2015年5月 1篇

2015年4月 2篇

2014年12月 1篇

2014年9月 1篇

2014年4月 1篇

展开

热门文章

四轴PID讲解

阅读量：40982

四元数姿态解算中的地磁计融合解读

阅读量：23555

四旋翼飞行器的姿态解算小知识点

阅读量：22767

四元数姿态的梯度下降法推导和解读

阅读量：21545

微型四旋翼飞行器的设计与制作

阅读量：15620

最新评论

STM32单片机学习笔记

qq_38405680: 厉害

四轴加速度计滤波

qq_41373510: 楼主，能不能分享下你对加速度和角速度原始数据进行滤波的代码啊

四旋翼飞行器的姿态解算小知识点

xiangfengzhuo: 博主您好，当载体运动起来以后，加速度计所测得为动态加速度，此时如何根据加速度计的测量值估计姿态？望...

四元数姿态的梯度下降法推导和解读

xajh_wk: [reply]qq_27967579[/reply] 我也疑惑这里

四轴加速度计滤波

u013082827: 博主有没有写有没有相关的论文？我希望参考一下。



100M

100M宽带+电信电视

电信宽带每月仅需60元！



联系我们



微信客服



区块链大本营

QQ客服 kefu@csdn.net
客服论坛 400-660-0108

工作时间 8:00-22:00

开发者调查 招聘 Python学习路线! Office 365商业协作版 5折钜惠!

51

12

12

12

12

12

12

12

12

登录

注册

×

