第3章

四旋翼飞行器的原理

3.1 基本原理

四旋翼飞行器通过 4 个螺旋桨的相互配合，利用力的合成和分解原理，可以实现各种飞行动作，四旋翼飞行器通常被设计为两种模式，如图 3.1 中的“×”字模式和“十”字模式。

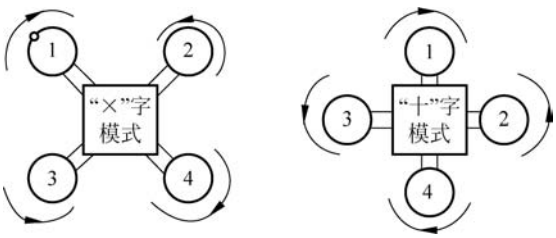

图 3.1 “×”字模式和“十”字模式

为了抵消旋翼在旋转过程中产生的反扭矩，例如 1 号位置的旋翼逆时针旋转时，空气会产生推动飞行器逆时针旋转的反扭矩，飞行器的相邻旋翼需要有不同的转向，即 1、4 号电机顺时针旋转而 2、3 号电机逆时针旋转，这样空气推动飞行器旋转的反扭矩刚好抵消，使得飞行器能够平稳飞行。由于飞行器的 4 个桨在旋转的过程中都要提供向上的升力，而相邻电机转向相反，因此在相邻电机上还要使用形状不同的螺旋桨，分别称为正桨和反桨，通常 1、4 号电机顺时针旋转搭载正桨，2、3 号电机逆时针旋转搭载反桨。

3.2 四旋翼飞行器的 6 个基本飞行动作

四旋翼飞行器可以沿着升降方向、前进后退方向和左右方向平移运动，以及沿着 3 个轴旋转运动，共 6 个基本飞行动作。

3.2.1 升降运动

升降运动的实质就是四旋翼飞行器沿着 Z 轴的两个方向运动，如图 3.2 所示。只要 4 个电机转速同时增加，使飞行器的升力大于重力和阻力的和之后，就可以上升，反之，4 个电

机的转速同时下降，就可以使飞行器下降，当升力和重力达到平衡之后，飞行器就能实现空中悬停，在实际飞行中，悬停是一个动态稳定的过程，即使在悬停中，升力也是不断进行调整的。

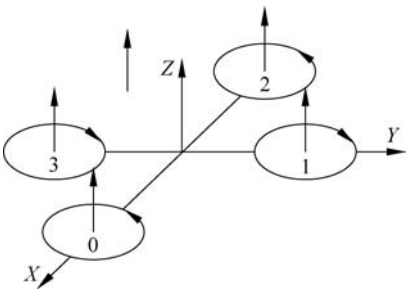

图 3.2　升降运动

3.2.2　俯仰运动

假定 X 轴是飞行器的正方向，俯仰运动就是绕 Y 轴的旋转运动；而若飞行器在空中自由飞行，那么俯仰运动就伴随着前进和后退的运动；若飞行器绕 Y 轴正方向逆时针旋转，飞行器就会向前运动，反之就会向后运动。保持 1、3 号电机的转速不变，增大 0 号电机的转速，减小 2 号电机的转速，飞行器就会绕 Y 轴顺时针旋转，这个动作称为仰；反之，飞行器会绕 Y 轴逆时针旋转，称为俯。俯仰运动如图 3.3 所示。

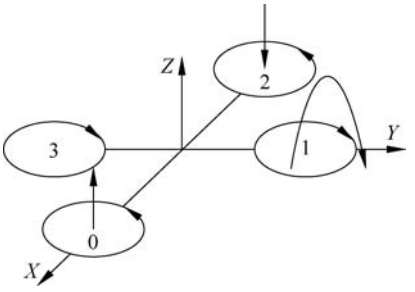

图 3.3　俯仰运动

3.2.3　横滚运动

横滚运动的原理和俯仰运动的原理相同，即保持 0 号和 2 号电机的转速不变，改变 1、3 号电机的转速，使飞行器绕 X 轴进行旋转，称为横滚运动，如图 3.4 所示。

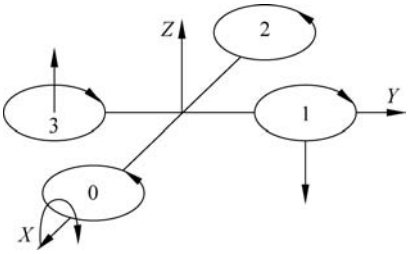

图 3.4　横滚运动

3.2.4 偏航(自旋)运动

偏航运动又被称为自旋运动,是飞行器绕 Z 轴旋转的运动,如图 3.5 所示。在一个螺旋桨旋转时,飞行器会受到一个反扭矩的影响,导致飞行器沿螺旋桨转向的相反方向旋转,而四旋翼飞行器采用了 4 个螺旋桨,其中 2 个顺时针旋转,另外 2 个逆时针旋转抵消了这种反扭矩。那么,为了让飞行器能够产生自旋运动,只要让由于 4 个电机旋转受到的反扭矩无法自相抵消即可。当 0、2 号电机的转速上升,1、3 号电机的转速下降,飞行器就会沿 Z 轴顺时针旋转,反之,则会沿 Z 轴逆时针旋转。通常在进行偏航运动时,为了保证飞行器升力不变,其中两个电机下降的转速和另外两个电机上升的转速相同。

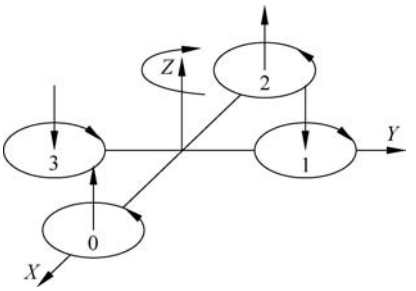

图 3.5 偏航(自旋)运动

3.3 姿态解算

姿态解算是飞行器正常飞行的基础,只有飞行器能够正确掌握飞行姿态,飞行器才能够通过动态平衡调节保持飞行的稳定性。要进行姿态解算,需要对传感器获取的多种不同的数据进行滤波和融合,选择合适的方式描述飞行器的姿态,以及用一种方法让飞行器的姿态与姿态数据一一对应。

3.3.1 姿态表示

欧拉角是一种最为常用的描述旋转的办法,它用三次旋转来表示一个物体的姿态,分别是章动角 φ、旋进角 θ 以及自旋角 ψ,三次旋转的顺序有多种选择,飞行器的姿态描述中,多用 zyx 的旋转顺序进行描述。

zyx 的取法又被称为航空次序欧拉角,分为 Roll(横滚角)、Pitch(俯仰角)和 Yaw(偏航角),如图 3.6 所示。坐标系首先绕 Z 轴旋转 ψ,将 ψ 称为偏航角再绕 Y 轴旋转 θ,将 θ 称为俯仰角;最后绕 X 轴旋转 φ,将 φ 称为横滚角。

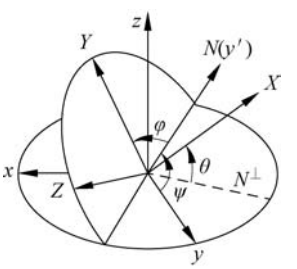

图 3.6 航空次序欧拉角

为了保证旋转描述的一致性,俯仰角被限制在 $-90°\sim 90°$,其他两个角度则被限制在 $-180°\sim 180°$。

欧拉角描述法存在明显的不足。首先在俯仰角大小接近 90°时会出现奇点,具体的现象有当俯仰角在 90°时横滚

角与俯仰角会发生突变，以及在奇点附近，可能出现某一自由度的角速度分量很大甚至无穷大的问题。另外，欧拉角仅可以用于描述一次旋转，当进行第二次旋转时，很有可能产生万向节死锁问题，具体表现为，考虑第一次旋转时，俯仰角旋转了 90°，另外两个角度均为 0°，如果在此基础上实现第二次旋转，绕最初的 Z 轴转动一个角度，这是无法实现的，因为第一次旋转后，横滚角与偏航角的转轴重合，即系统丢失了一个自由度，只剩下两个自由度。鉴于欧拉角姿态描述中这种无法回避的奇异现象的存在，采用四元素与欧拉角的相互转换构建姿态控制率比较适合工程应用。

3.3.2　数据滤波

数据滤波是去除噪声、还原真实数据的一种数据处理技术。

1. 维纳滤波

维纳滤波是一种平稳随机过程的最佳滤波理论，换句话说就是在滤波过程中系统的状态参数(或信号的波形参数)是稳定不变的。维纳滤波仅在理论上有意义，在实际应用中有局限性，其表现在：不适用于非平稳随机过程的滤波；要用到所有时刻的采样数据，需要的数据存储容量大；求解维纳-霍夫方程时要用到高阶数矩阵的求逆运算，计算量大，而且实际数据下的维纳-霍夫方程可能无解。

2. 卡尔曼滤波

卡尔曼滤波不仅适用于平稳随机过程，也适用于非平稳随机过程。它将系统的状态迁移用状态方程来表述，并用固定维数的矩阵运算递推式代替了维纳滤波解维数大的线性方程组。克服了维纳滤波的一系列局限性而获得了成功应用。在应用中，卡尔曼滤波区别于递归加权最小二乘法的关键是考虑了系统噪声和测量噪声，建立了包括状态方程和观测方程在内的准确加权融合系统模型。

3.3.3　数据融合

在传感器获取一系列数据后，通过数据融合一系列具有不同特性的、存在误差的数据经过计算，得到最接近实际的值。

对于姿态解算而言，需要将陀螺仪获得的三轴角速度数据和加速度计获得的三轴加速度数据进行融合，获取飞行器的姿态角数据。

陀螺仪测得的数据是三轴的角速度数据，将角速度进行积分，就可以获取三轴的角度数据。陀螺仪受飞行器振动的影响较小，噪声较低，但由于使用的是积分的方法，时间久以后容易产生累积误差，且无法修复，对飞行器的姿态控制影响较大。

加速计测得的数据是重力加速度和飞行器本身的加速度在三轴上的分量。当飞行器静止时，利用测得的三轴加速度数据可以唯一确定角度，求出的是一组绝对量，不存在累积误差。但在飞行器实际飞行的过程中，电机的振动会让加速度计的数据包含大量的噪声，难以滤除，同时，由于飞行器自身的加速度造成的角度计算偏差，也是加速度计本身无法排除的。

因此，可见通过陀螺仪计算出的角度数据瞬时比较准，长时间会产生累积误差，而通过加速度计算出的角度值瞬时误差较大，长时间则比较稳定，因此，需要通过一定的算法将这两种数据进行融合，以获得更接近真实角度的角度值。

常用于加速度计和陀螺仪数据融合的算法有一阶互补滤波算法、二阶互补滤波算法和

卡尔曼滤波算法。

1. 一阶互补滤波

一阶互补滤波指的是对加速度计和陀螺仪计算而得的角度数据进行简单加权。实现一阶互补滤波的程序代码如下：

```
K = 0.075;     //对加速度计取值的权重
float A = K /(K + dt);
Com_angle = A * (Com_angle + omega * dt) + (1 - A) * angleA;
//Com_angle 为融合角度值,omega 为角速度,dt 为采样间隔,angleA 为加速度计求得角度
```

2. 二阶互补滤波

二阶互补滤波则是将加速度信号作为参考值，将融合计算得到的角度与加速度计算出的角度的偏差量加入积分之中。代码实现如下：

```
K = 0.5;
float x1 = (angleA - Com2_angle) * K * K;
y1 = y1 + x1 * dt;
float x2 = y1 + 2 * K * (angleA - Com2_angle) + omega;
Com2_angle = Com2_angle + x2 * dt;
```

3. 卡尔曼滤波

卡尔曼滤波针对各种随机过程都十分有效。相比维纳滤波，卡尔曼滤波能够用于非平稳随机过程，克服了其局限性。

卡尔曼滤波器被称为最优化自回归数据处理算法，对于很多应用，它是最好的方法，效率可以说是最高的。它是一个迭代预测器，每次只需输入上一时刻的数据，就能预测下一时刻的数据，在利用加速度值计算出角度后得到后验估计误差，从而不断迭代修正，得出最优解。

从理论上来说，卡尔曼滤波器能得到最好的结果，但对于飞行器而言，由于在飞行过程中振动十分剧烈，会对卡尔曼滤波器的预测产生很大的影响，从而导致在手持飞行器进行测试时，卡尔曼滤波器具有最好的表现，一阶互补滤波效果较差，而二阶互补滤波器数据比较平滑，但收敛较慢，角度跟踪性能差。但在实际飞行时，二阶互补滤波器飞行最为稳定。

3.3.4 姿态解算

对于飞行器的姿态解算，通常采用获取陀螺仪和加速度计的数据，利用这些数据进行四元数运算，直接求出航空次序欧拉角。四元数相对于其他各种旋转描述法有着明显的优势。

(1) 巧妙的设计避免了处理缓慢的三角函数计算，运算效率提升。

(2) 四元数在全角范围内无奇点，没有万向节死锁问题。

常用的旋转描述有矩阵法、欧拉角以及四元数。

1. 矩阵法

(1) 绕 Z 轴旋转，如图 3.7 所示。

$$x' = x\cos t - y\sin t$$

$$y' = x\sin t + y\cos t$$

$$z' = z$$

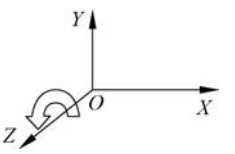

图 3.7 绕 Z 轴旋转

矩阵表示为：

$$(x' \quad y' \quad z' \quad 1)=(x \quad y \quad z \quad 1)\begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{绕 } Z \text{ 轴旋转}$$

(2) 绕 X 轴旋转，如图 3.8 所示。

$y'=y\cos t-z\sin t$

$z'=y\sin t+z\cos t$

$x'=x$

矩阵表示为：

$$(x' \quad y' \quad z' \quad 1)=(x \quad y \quad z \quad 1)\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{绕 } X \text{ 轴旋转}$$

(3) 绕 Y 轴旋转，如图 3.9 所示。

$z'=z\cos t-x\sin t$

$x'=z\sin t+x\cos t$

$y'=y$

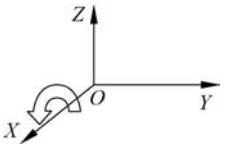

图 3.8　绕 *Y* 轴旋转

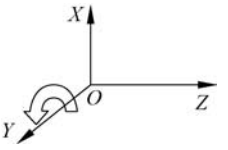

图 3.9　绕 *Z* 轴旋转

矩阵表示为：

$$(x' \quad y' \quad z' \quad 1)=(x \quad y \quad z \quad 1)\begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{绕 } Y \text{ 轴旋转}$$

矩阵法计算量最大，既要计算三角函数，计算三维旋转要用到四阶的矩阵运算，在实际使用中并不常用。

2. 四元数

欧拉角是一种最直观的描述旋转的方法，它以三次旋转的合成来描述空间任意旋转。但如 3.3.1 节所述欧拉角姿态描述中存在无法回避的奇异现象，需要结合四元素法构建姿态算法。

四元数是简单的超复数，一个实部，三个虚部。

四元数一般定义如下：

$$q=w+x\mathrm{i}+y\mathrm{j}+z\mathrm{k}$$

其中，w、x、y、z 是实数。同时，有

$$i * i = -1$$
$$j * j = -1$$
$$k * k = -1$$

四元数也可以表示为：

$$q = [\boldsymbol{v}, w]$$

其中，$\boldsymbol{v} = (x, y, z)$是矢量，$w$ 是标量，虽然$\boldsymbol{v}$ 是矢量，但不能简单地理解为三维空间的矢量，它是四维空间中的矢量。

通俗地讲，一个四元数(quaternion)描述了一个旋转轴和一个旋转角度。创建这个旋转轴和这个角度可以通过 Quaternion::ToAngleAxis 转换得到，它可以返回一个绕轴线 axis 旋转 angle 角度的四元数变换。当然也可以随意指定一个角度一个旋转轴来构造一个四元数。这个角度是相对于单位四元数而言的，也可以说是相对于物体的初始方向而言的。

四元数项对于其他各种旋转描述法有着明显的优势。

(1) 虽然较欧拉角表示法多了一个参量，但是四元数的运算却巧妙地绕开了复杂耗时的三角函数运算，有更高的运算效率。

(2) 四元数在全角范围内无奇点，不会出现死锁问题。

3.3.5 PID 平衡算法

PID 是在自动控制领域非常常用的一种算法。在飞行器的控制中，PID 控制非常重要，使用在对飞行器姿态角的控制、飞行器的定高控制、定点悬停控制等。

1. PID 算法的理解

如果要让飞行器从地面起飞在空中某一高度悬停，就要有向上的升力来克服地球引力，这个力是需要控制的量，飞行器高度有它现在的“当前值”，也有期望的“目标值”，当两者差距较大时，就让电机开足马力，尽快让飞行器到达目标高度附近，而当飞行器接近目标高度时，就让电机稍稍用力即可。这就是 P 的作用，P 就是比例的意思。实际写程序时，就让偏差(目标值减去当前值)与调节装置的“调节力度”建立一个一次函数的关系，就可以实现最基本的“比例”控制了。P 越大，调节作用越激进，P 调小会让调节作用更缓慢。

有了 P 的作用后会发现，只有 P 好像不能让飞行器稳定，飞行器总是在某一高度处上下波动。设想一个弹簧挂一重物，现在在平衡位置上拉它一下然后松手，这时它会振荡起来。因为阻力很小，它可能会振荡很长时间，才会重新停在平衡位置。要是它浸没在水里，同样拉它一下，那么重新停在平衡位置的时间就短得多。所以需要一个控制作用，让被控制的物理量的“变化速度”趋于 0，即类似于“阻尼”的作用。当比较接近目标时，P 的控制作用就比较小了，越接近目标，P 的作用越小。但还是有惯性让飞行器冲过目标高度，D 的作用就是让物理量的速度趋于 0，只要这个量具有了速度，D 就向相反的方向用力，尽力抑制这个变化。D 参数越大，向速度相反方向刹车的力就越强。

飞行器加上 P 和 D 两种控制作用，如果参数调节合适，飞行器就应该可以悬停在空中了。但看了一眼返回的高度值可能会发现一个不好的情况，稳定的高度值实际还没达到目标值，也许飞行器太重，这时上升的动力和重力已经相等了，这可怎么办？P 认为和目标已经很近了，只需要轻轻加力就可以了，D 认为高度没有波动，好像没我什么事了，于是高度永远也到不了目标值。根据常识知道，应该进一步增加上升动力，于是就要引入参数 I 了。增

加一个积分量，只要高度偏差存在，就不断地对偏差进行积分（累加），并反映在调节力度上。这样一来，即使高度相差不太大，但是随着时间的推移，只要没达到目标高度，这个积分量就不断增加。系统就会慢慢意识到：还没有到达目标高度，该增加功率了。到了目标高度后，假设高度没有波动，积分值就不会再变动。这时，上升动力仍然等于重力，但高度是稳稳地定在了目标值上。I 的值越大，积分时乘的系数就越大，积分效果越明显。所以，I 的作用就是减小静态情况下的误差，让受控物理量尽可能接近目标值。I 在使用时还有个问题：需要设定积分限制，以防止在刚开始上升时就把积分量积得太大，之后难以控制。

2. PID 算法的使用

在飞行器的控制中采用数字式 PID，数字 PID 控制算法主要有位置式 PID 和增量式 PID 两种。位置式 PID 控制的基本算法框图如图 3.10 所示。

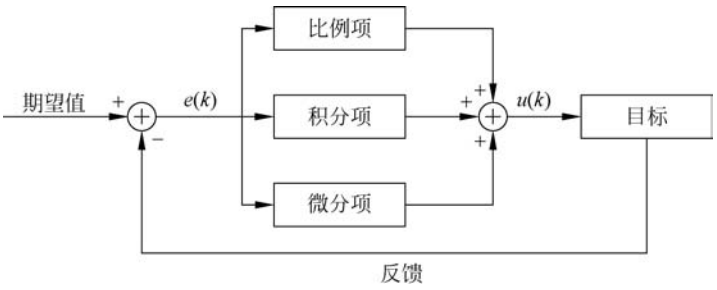

图 3.10　PID 基本算法框图

计算公式为：

$$u(k)=K_{\mathrm{P}}\left[e(k)+\frac{T}{T_i}\sum_{j=0}^{i}e(j)+\frac{T_{\mathrm{d}}}{T}(e(k)-e(k-1))\right] \tag{3.1}$$

其中，$e(k)$代表的是设定目标和当前值的偏差值，若 $e(k)$是设定当前横滚角和设定横滚角的差值，那么 $u(k)$指的就是横滚方向的电机输出量。

将式(3.1)展开，可以分为比例项、微分项和积分项。

比例项控制是十分直接的控制，根据差值立即对油门输出量作出改变，将飞行器向差值的反方向调整就行。通过调整比例项的参数，可以让飞行器的角度在一定范围内等幅摆动。

微分项使用的变量是上一次和本次的差值之差，以横滚为例就是横滚方向的转动速度，在比例项的基础上加上微分项参数，主要是为了限制飞行器旋转的速度，通过比例项和微分项的联合作用，可以让飞行器在该角度上形成动态稳定的结果。

积分项的存在是为了消除稳态误差，如果飞行器存在重心的偏差，或者飞行器的电机动力有所不同，那么飞行器通过比例项和微分项维持稳定之后，有可能和设定目标有一个固定的误差，通过加入积分项，可以消除这种稳态误差。

$$u(k)=K_{\mathrm{P}}\left[e(k)+\frac{T}{T_i}\sum_{j=0}^{k}e(j)+\frac{T_{\mathrm{d}}}{T}(e(k)-e(k-1))\right] \tag{3.2}$$

$$u(k-1)=K_{\mathrm{P}}\left[e(k-1)+\frac{T}{T_i}\sum_{j=0}^{k-1}e(j)+\frac{T_{\mathrm{d}}}{T}(e(k-1)-e(k-2))\right] \tag{3.3}$$

数字式 PID 调节还有一种增量式 PID 调节，由上面两式，根据递推原理可以得到：

$$A = K_P\left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right)$$

$$B = K_P\left(1 + \frac{2T_d}{T}\right)$$

$$C = K_P\frac{T_d}{T}$$

增量式 PID 求出的是油门的增量，与当前的油门值叠加，就可以实现对飞行器的控制，增量式 PID 相对于位置式 PID 具有以下优点。

（1）因为 PID 的输出是油门增量，因此短暂出现错误动作、错误数据时影响较小，需要时还能够用逻辑判断消除。

（2）计算公式中没有积分量，输出量只和最近几次采样的数据有关，方便程序进行处理，没有积分的累积误差。