最优控制：
基于状态空间法的四旋翼
飞行器的LQR控制

2022.6.4

目 录

第一章 研究动机

LQR 是一种流行的线性系统的控制方法。对于四旋翼飞行器这样的非线性动力学系统，通常会限制工作区域，并在平衡点处对系统进行线性化[4]。这降低了非线性系统随时间线性化的计算成本，但同时也降低了控制性能与精度。

本文研究了与状态变量有关的 LQR 四旋翼飞行器控制。为了获得更好的性能和精度，我们不再限制四旋翼飞行器的工作区域和应用小角度近似，而是随时间更新状态矩阵 A 并计算最优控制信号。

为了通过实时求解 CARE（连续时间代数李卡提方程）来计算最佳增益，我们探索了 “结构保持加倍算法”（SDA）。仿真结果表明，SDA 比 MATLAB 的内置函数 care 快了 5.1 倍，且具有很高的精度。

第二章 符号

符号	含义
$[\varphi,\theta,\psi]^T$	欧拉角
$R_i \in SO(3)$	从固定坐标系到惯性坐标系的旋转矩阵
$[x,y,z]^T \in R^3$	惯性坐标系中的位置
$v_B=[u,v,w]^T \in R^3$	固定坐标系中的速度
$\Omega=[p,q,r]^T \in R^3$	固定坐标系中的角速度
$M=[\tau_x,\tau_y,\tau_z]^T \in R^3$	固定坐标系中的力矩
$f_t \in R$	四旋翼飞行器的总推力
$m \in R$	四旋翼飞行器的质量
$J \in R^3$	相对于固定坐标系的惯性矩阵
$T \in R^3$	将固定坐标系角速度映射到欧拉角速度的矩阵
$x \in R^{12}$	LQR 控制器的状态变量
$u \in R^4$	LQR 控制器的控制变量
$A \in R^{12\times12}$	LQR 控制器的系统矩阵
$B \in R^{4\times10}$	LQR 控制器的输入矩阵
$C \in R^{12\times12}$	LQR 控制器的观测矩阵
$Q \in R^{12\times12}$	LQR 控制器的状态惩罚矩阵
$R \in R^{4\times4}$	LQR 控制器的控制惩罚矩阵
$K \in R^{12\times12}$	LQR 控制器的反馈增益矩阵
$X \in R^{4\times10}$	CARE（连续时间代数 Riccati 方程）的唯一解

第三章 数学模型

3.1 欧拉角和旋转矩阵

欧拉角是由 Leonhard Euler 引入的用来描述刚体方向的三个角度。为了在三维欧式空间中描述这样一个方向，需要三个参数。欧拉角被用来描述一个参照系和另一个参照系之间的位置关系，它们将一个参照系中的一个点的坐标转换为该点在另一个参照系中的坐标。欧拉角通常表示为 $\varphi\in[-\pi,\pi]$，$\theta\in[-\frac{\pi}{2},\frac{\pi}{2}]$ $\psi\in[-\pi,\pi]$。我们可以从欧拉角生成三个变换矩阵：

以 φ 角度旋转 x 轴：

$$R_x(\varphi)=\begin{bmatrix}1 & 0 & 0\\ 0 & c(\varphi) & -s(\varphi)\\ 0 & s(\varphi) & c(\varphi)\end{bmatrix} \tag{3-1}$$

以 θ 角度旋转 y 轴：

$$R_y(\theta)=\begin{bmatrix}c(\theta) & 0 & s(\theta)\\ 0 & 1 & 0\\ -s(\theta) & 0 & c(\theta)\end{bmatrix} \tag{3-2}$$

以 ψ 角度旋转 z 轴：

$$R_z(\theta)=\begin{bmatrix}c(\psi) & -s(\psi) & 0\\ s(\psi) & c(\psi) & 0\\ 0 & 0 & 1\end{bmatrix} \tag{3-3}$$

其中 $c(\varphi)=\cos(\varphi)$，$s(\varphi)=\sin(\varphi)$，$c(\theta)=\cos(\theta)$，$s(\theta)=\sin(\theta)$，$c(\psi)=\cos(\psi)$，$s(\psi)=\sin(\psi)$。以 x-y-z 的乘法顺序组合三个矩阵：

$$\begin{aligned}R_{zyx}(\varphi,\theta,\psi)&=R_z(\psi)\cdot R_y(\theta)\cdot R_x(\varphi)\\ &=\begin{bmatrix}c(\theta)c(\psi) & s(\varphi)s(\theta)c(\psi)-c(\varphi)s(\psi) & c(\varphi)s(\theta)c(\psi)+s(\varphi)s(\psi)\\ c(\theta)s(\psi) & s(\varphi)s(\theta)s(\psi)+c(\varphi)c(\psi) & c(\varphi)s(\theta)s(\psi)-s(\varphi)c(\psi)\\ -s(\theta) & s(\varphi)c(\theta) & c(\varphi)c(\theta)\end{bmatrix}\end{aligned} \tag{3-4}$$

这个矩阵描述了从机身参考系到惯性参考系的旋转过程。

旋转矩阵是特殊正交群 SO(3)的成员，它不是乘法换元的，而且还具有 $R^{-1}=R^{T}$ 和 $\det(R)=1$ 的特性。

请注意，有几种方法可以生成旋转矩阵。然而对于欧拉角，奇点发生在 $\theta=\pm 90^{\circ}$，这被称为“万向节锁定”。

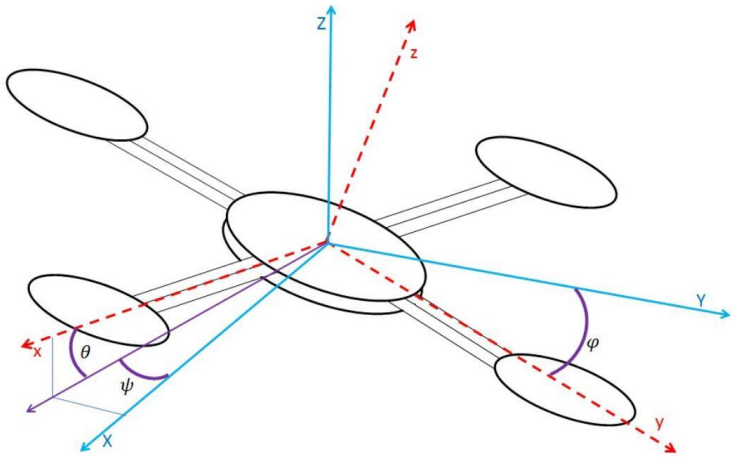


图 3-1 欧拉角

3.2 欧拉角速度与角速度

与角速度矢量不同，欧拉角速度 $\dot{\varphi}$，$\dot{\theta}$，$\dot{\psi}$ 位于不同的坐标系中。为了计算固定坐标系的角速度，我们需要将它们分别转换为同一坐标系中，如下所示：

$$\Omega=\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix}+R_{z}\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix}+R_{z}R_{y}\begin{bmatrix}\dot{\varphi}\\0\\0\end{bmatrix} \tag{3-5}$$

$$\Omega=J\begin{bmatrix}\dot{\varphi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix} \tag{3-6}$$

矩阵 J 被称为 Jacobian 矩阵。反雅可比矩阵 J^{-1} 可以帮助我们从固定坐标系中的角速度计算出欧拉角速度。

$$\begin{bmatrix}\dot{\varphi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix}=J^{-1}\Omega \tag{3-7}$$

$$T \equiv J^{-1} = \begin{bmatrix} 1 & s(\varphi)t(\theta) & c(\varphi)t(\theta) \\ 0 & c(\varphi) & -s(\varphi) \\ 0 & \frac{s(\varphi)}{c(\theta)} & \frac{c(\varphi)}{c(\theta)} \end{bmatrix} \tag{3-8}$$

3.3 四旋翼飞行器的动力学

四旋翼飞行器动力学的简化形式如下[5]。

$$\dot{x} = v \tag{3-9}$$

$$m\dot{v} = mge_3 - f_t R e_3 \tag{3-10}$$

$$\dot{R} = R\hat{\Omega} \tag{3-11}$$

$$J\dot{\Omega} + \Omega \times J\Omega = M \tag{3-12}$$

虽然这与下一节控制器设计中推导的略有不同，但我们使用这些公式来更新仿真的动力学特征。

3.4 推导四旋翼无人机的状态空间模型

在本节中，我们将推导出 12 个四旋翼无人机的动力学方程，用于后面章节的控制器设计。

如同上一节中的推导，固定坐标系中的角速度到欧拉角速度的变换如下：

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T \cdot \Omega \tag{3-13}$$

$$T = \begin{bmatrix} 1 & s(\varphi)t(\theta) & c(\varphi)t(\theta) \\ 0 & c(\varphi) & -s(\varphi) \\ 0 & \frac{s(\varphi)}{c(\theta)} & \frac{c(\varphi)}{c(\theta)} \end{bmatrix} \tag{3-14}$$

展开后得到转速的动力学方程：

$$\begin{cases} \dot{\varphi} = p + r(c(\varphi)t(\theta)) + q(s(\varphi) + t(\theta)) \\ \dot{\theta} = q(c(\varphi)) - r(s(\varphi)) \\ \dot{\psi} = r\dfrac{c(\varphi)}{c(\theta)} + q\dfrac{s(\varphi)}{c(\theta)} \end{cases} \tag{3-15}$$

从固定坐标系到惯性坐标系的平移速度的变换为：

$$v_I = R \cdot v_B \tag{3-16}$$

展开后得到速度的动力学方程：

$$\begin{cases} \dot{x} = w(s(\varphi)s(\psi) + c(\varphi)c(\psi)s(\theta)) - v(c(\varphi)s(\psi) - c(\psi)s(\varphi)s(\theta)) + u(c(\psi)c(\theta)) \\ \dot{y} = v(c(\varphi)c(\psi) + s(\varphi)s(\psi)s(\theta)) - w(c(\psi)s(\varphi) - c(\varphi)s(\psi)s(\theta)) + u(c(\theta)s(\psi)) \\ \dot{z} = w(c(\varphi)c(\theta)) - u(s(\theta)) + v(c(\theta)s(\varphi)) \end{cases} \tag{3-17}$$

描述力矩、角速度和角加速度之间关系的欧拉方程如下：

$$J\dot{\Omega} + \Omega \times J\Omega = M \tag{3-12}$$

其中

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \in R^{3\times 3}$$

展开得到以下方程。稍后将对它们进行重新排序，以得到角加速度的动力学方程：

$$\begin{cases} \tau_x = \dot{p}I_x - qrI_y + qrI_z \\ \tau_y = \dot{q}I_y + prI_x - prI_z \\ \tau_z = \dot{r}I_z - pqI_x + pqI_x \end{cases} \tag{3-18}$$

根据牛顿第二定律，力的方程如下：

$$m(\Omega \times v_B + \dot{v}_B) = f_B \tag{3-19}$$

类似地，展开后得到三个方程。同样，我们稍后将对它们重新排序，以得到转换后的加速动力学方程：

$$\begin{cases} -mg(s(\theta)) = m(\dot{u} + qw - rv) \\ mg(c(\theta)s(\varphi)) = m(\dot{v} - pw + ru) \\ mg(c(\theta)c(\varphi)) - f_t = m(\dot{w} + pv - qu) \end{cases} \tag{3-20}$$

其中，

$$f_B = Rm\dot{v} = R\left(mge_3 - f_t Re_3\right) = Rmge_3 - f_t e_3 \tag{3-21}$$

最后，四旋翼飞行器的所有动力学方程如下：

$$f = \begin{cases} \dot{\varphi} = p + r(c(\varphi)t(\theta)) + q(s(\varphi)t(\theta)) \\ \dot{\theta} = q(c(\varphi)) - r(s(\varphi)) \\ \dot{\psi} = r\dfrac{c(\varphi)}{c(\theta)} + q\dfrac{s(\varphi)}{c(\theta)} \\ \dot{p} = \dfrac{I_y - I_z}{I_x} rq + \dfrac{\tau_x}{I_x} \\ \dot{q} = \dfrac{I_z - I_x}{I_y} pr + \dfrac{\tau_y}{I_y} \\ \dot{r} = \dfrac{I_x - I_y}{I_z} pq + \dfrac{\tau_z}{I_z} \\ \dot{u} = rv - qw - g(s(\theta)) \\ \dot{v} = pw - ru + g(s(\varphi)c(\theta)) \\ \dot{w} = qu - pv + g(c(\theta)c(\varphi)) - \dfrac{f_t}{m} \\ \dot{x} = w(s(\varphi)s(\psi) + c(\varphi)c(\psi)s(\theta)) - v(c(\varphi)s(\psi) - c(\psi)s(\varphi)s(\theta)) + u(c(\psi)c(\theta)) \\ \dot{y} = v(c(\varphi)c(\psi) + s(\varphi)s(\psi)s(\theta)) - w(c(\psi)s(\varphi) - c(\varphi)s(\psi)s(\theta)) + u(c(\theta)s(\psi)) \\ \dot{z} = w(c(\varphi)c(\theta)) - u(s(\theta)) + v(c(\theta)s(\varphi)) \end{cases}$$

（3-22）

第四章 LQR 控制

4.1 四旋翼飞行器的线性控制

设计的线性控制器的状态变量为：

$$\mathrm{x}=[\varphi,\theta,\psi,p,q,r,u,v,w,x,y,z]^T \tag{4-1}$$

设计的控制器的控制输入为：

$$\mathrm{u}=[f_t,\tau_x,\tau_y,\tau_z]^T \tag{4-2}$$

最后，线性系统的状态空间表示为：

$$\begin{cases}\dot{\mathrm{x}}=A\mathrm{x}+B\mathrm{u}\\ y=C\mathrm{x}\end{cases} \tag{4-3}$$

能观测性矩阵由以下公式给出：

$$\mathcal{O}=\begin{bmatrix}C\\ C\cdot A\\ C\cdot A^2\\ M\\ C\cdot A^{11}\end{bmatrix}\in\mathbb{R}^{144\times 12} \tag{4-4}$$

能控性矩阵由以下公式给出：

$$\mathcal{C}=\begin{bmatrix}B & A\cdot B & A^2\cdot B & \cdots & A^{11}\cdot B\end{bmatrix}\in\mathbb{R}^{12\times 48} \tag{4-5}$$

4.2 线性化

如上所述，为了进行线性化，需要一个平衡点。这样一个平衡点可以是：

$$\mathrm{x}_e=\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{x} & \bar{y} & \bar{z}\end{bmatrix}^T\in R^{12} \tag{4-6}$$

平衡点（4-4）可由恒定的输入值得到：

$$\mathrm{u}_e=\begin{bmatrix}mg & 0 & 0 & 0\end{bmatrix}^T\in R^4 \tag{4-7}$$

使用上一节得出的动力学方程，我们可以通过围绕平衡点做线性化得到 A 和 B 矩阵。

$$A=\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_e,\mathbf{u}=\mathbf{u}_e}=\begin{bmatrix}\frac{\partial \dot{\varphi}}{\partial \mathbf{x}}\\ \frac{\partial \dot{\theta}}{\partial \mathbf{x}}\\ \frac{\partial \dot{\psi}}{\partial \mathbf{x}}\\ \frac{\partial \dot{p}}{\partial \mathbf{x}}\\ \frac{\partial \dot{q}}{\partial \mathbf{x}}\\ \frac{\partial \dot{r}}{\partial \mathbf{x}}\\ \frac{\partial \dot{u}}{\partial \mathbf{x}}\\ \frac{\partial \dot{v}}{\partial \mathbf{x}}\\ \frac{\partial \dot{w}}{\partial \mathbf{x}}\\ \frac{\partial \dot{x}}{\partial \mathbf{x}}\\ \frac{\partial \dot{y}}{\partial \mathbf{x}}\\ \frac{\partial \dot{z}}{\partial \mathbf{x}}\end{bmatrix} \tag{4-8}$$

$$\frac{\partial \dot{\varphi}}{\partial \mathbf{x}}=\left[-rs(\varphi)t(\theta)+qc(\varphi)t(\theta),rc(\varphi)\sec^2\theta+qs(\varphi)\sec^2\theta,0,1,s(\varphi)t(\theta),c(\varphi)t(\theta),0,0,0,0,0,0\right]$$

（4-9）

$$\frac{\partial \dot{\theta}}{\partial \mathbf{x}}=[-qs(\varphi)-rc(\varphi),0,0,0,c(\varphi),-s(\varphi),0,0,0,0,0,0] \tag{4-10}$$

$$\frac{\partial \dot{\psi}}{\partial \mathbf{x}}=\left[-r\frac{s(\varphi)}{c(\theta)}+q\frac{c(\varphi)}{c(\theta)},rc(\varphi)\sec\theta t(\theta)+qs(\varphi)\sec\theta t(\theta),0,0,\frac{s(\varphi)}{c(\theta)},\frac{c(\varphi)}{c(\theta)},0,0,0,0,0,0\right]$$

（4-11）

$$\frac{\partial \dot{p}}{\partial \mathbf{x}}=\left[0,0,0,0,\frac{I_y-I_z}{I_x}r,\frac{I_y-I_x}{I_y}q,0,0,0,0,0,0\right] \tag{4-12}$$

$$\frac{\partial \dot{q}}{\partial \mathbf{x}}=\left[0,0,0,\frac{I_z-I_x}{I_y}r,0,\frac{I_z-I_x}{I_y}p,0,0,0,0,0,0\right] \tag{4-13}$$

$$\frac{\partial \dot{r}}{\partial \mathbf{x}} = \left[0,0,0,\frac{I_x - I_y}{I_z} q,\frac{I_x - I_y}{I_z} p,0,0,0,0,0,0,0 \right] \tag{4-14}$$

$$\frac{\partial \dot{u}}{\partial \mathbf{x}} = [0,-gc(\theta),0,0,-w,v,0,r,-q,0,0,0] \tag{4-15}$$

$$\frac{\partial \dot{v}}{\partial \mathbf{x}} = [gc(\varphi)c(\theta),-gs(\varphi)s(\theta),0,w,0,-u,-r,0,p,0,0,0] \tag{4-16}$$

$$\frac{\partial \dot{w}}{\partial \mathbf{x}} = [-gc(\theta)s(\varphi),-gs(\theta)c(\varphi),0,-v,u,0,q,-p,0,0,0,0] \tag{4-17}$$

$$\frac{\partial \dot{x}}{\partial \mathbf{x}} = \begin{bmatrix} w(c(\varphi)s(\psi) - s(\varphi)s(\psi)s(\theta)) + v(s(\varphi)s(\psi) + c(\psi)c(\varphi)s(\theta)) \\ w(c(\varphi)c(\psi)c(\theta)) + v(c(\psi)s(\varphi)c(\theta)) - u(c(\psi)s(\theta)) \\ w(s(\varphi)c(\psi) - c(\varphi)s(\psi)s(\theta)) - v(c(\varphi)c(\psi) + s(\psi)s(\varphi)s(\theta)) - u(s(\psi)c(\theta)) \\ 0 \\ 0 \\ 0 \\ c(\psi)c(\theta) \\ -c(\varphi)s(\psi) + c(\psi)s(\varphi)s(\theta) \\ s(\varphi)s(\psi) + c(\varphi)c(\psi)s(\theta) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4-18}$$

$$\frac{\partial \dot{y}}{\partial \mathbf{x}} = \begin{bmatrix} v(-s(\varphi)c(\psi) + c(\varphi)s(\psi)s(\theta)) - w(c(\psi)c(\varphi) + s(\varphi)s(\psi)s(\theta)) \\ v(s(\varphi)s(\psi)c(\theta)) + w(c(\varphi)s(\psi)c(\theta)) - u(s(\theta)s(\psi)) \\ v(-c(\varphi)s(\psi) + s(\varphi)c(\psi)s(\theta)) + w(s(\psi)s(\varphi) + c(\varphi)c(\psi)s\theta) + u(c(\theta)c(\psi)) \\ 0 \\ 0 \\ 0 \\ c(\theta)s(\psi) \\ c(\varphi)c(\psi) + s(\varphi)s(\psi)s(\theta) \\ -c(\psi)s(\varphi) + c(\varphi)s(\psi)s(\theta) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4-19}$$

$$\frac{\partial \dot{z}}{\partial \mathbf{x}}=\begin{bmatrix} -w(s(\varphi)c(\theta))+v(c(\theta)c(\varphi)) \\ -w(c(\varphi)s(\theta))-uc(\theta)-v(s(\theta)s(\varphi)) \\ 0 \\ 0 \\ 0 \\ 0 \\ -s(\theta) \\ c(\theta)s(\varphi) \\ c(\varphi)c(\theta) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4-20}$$

$$A=\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\Bigg|_{\mathbf{x}=\mathbf{x}_e}^{\mathbf{u}=\mathbf{u}_e}=\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4-21}$$

$$B=\left.\frac{\partial \mathrm{f}(\mathrm{x},\mathrm{u})}{\partial \mathrm{u}}\right|_{\mathrm{x}=\mathrm{x}_e,\mathrm{u}=\mathrm{u}_e}=\begin{bmatrix}\frac{\partial\dot{\varphi}}{\partial \mathrm{u}}\\ \frac{\partial\dot{\theta}}{\partial \mathrm{u}}\\ \frac{\partial\dot{\psi}}{\partial \mathrm{u}}\\ \frac{\partial\dot{p}}{\partial \mathrm{u}}\\ \frac{\partial\dot{q}}{\partial \mathrm{u}}\\ \frac{\partial\dot{r}}{\partial \mathrm{u}}\\ \frac{\partial\dot{u}}{\partial \mathrm{u}}\\ \frac{\partial\dot{v}}{\partial \mathrm{u}}\\ \frac{\partial\dot{w}}{\partial \mathrm{u}}\\ \frac{\partial\dot{x}}{\partial \mathrm{u}}\\ \frac{\partial\dot{y}}{\partial \mathrm{u}}\\ \frac{\partial\dot{z}}{\partial \mathrm{u}}\end{bmatrix} \tag{4-22}$$

$$\rightarrow B=\begin{bmatrix}0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&\frac{1}{I_x}&0&0\\ 0&0&\frac{1}{I_y}&0\\ 0&0&0&\frac{1}{I_z}\\ 0&0&0&0\\ 0&0&0&0\\ -\frac{1}{m}&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\end{bmatrix} \tag{4-23}$$

4.3 线性二次型最优控制

最优控制的目标是最小化一个代价函数（它代表物理约束），然后通过确定控制输入来稳定系统。

定义的线性二次型调节器的代价函数为：

$$J(\mathbf{x},\mathbf{u})=\int_0^{\infty}\left(\tilde{\mathbf{x}}^T Q\tilde{\mathbf{x}}+\tilde{\mathbf{u}}^T R\tilde{\mathbf{u}}\right)dt \tag{4-24}$$

使函数最小化的控制输入$u(\cdot)$是一个状态线性反馈。有了期望状态向量x_0和前馈控制向量u_0，可以得到反馈控制信号和最优增益：

$$\mathbf{u}=\mathbf{u}_0-K(\mathbf{x}-\mathbf{x}_0) \tag{4-25}$$

$$K=R^{-1}B^T X \tag{4-26}$$

最优增益可以通过连续代数 Riccati 方程（CARE）的唯一解来计算，如果它存在的话。

$$A^T X+XA-XGX+H=0 \tag{4-27}$$

$$G=BR^{-1}B^T \tag{4-28}$$

$$H=C^T QC \tag{4-29}$$

第五章 求解 CARE 的数值方法

5.1 求解 CARE 的传统方法

我们得出了 Hamiltonian 矩阵 $\mathcal{H}=\begin{bmatrix} A & -G \\ -H & -A^T \end{bmatrix}$，并将其与 LQR 控制器的最优控制联系起来。在矩阵 A 可镇定和可观测的条件下，证明了 CARE 有唯一的对称半正定解 X 。

研究人员已经提出了几种求解 X 的算法。由 Laub 发表的 MATLAB 使用的算法[2]，应用 QR 算法求解 X ，并将问题改为特征值问题 $\mathcal{H}x=\lambda x$。不过 QR 算法既没有保留 Hamiltonian 结构，也没有保留相关的特征值。

其他方法：（1）定点迭代法：利用 DARE（离散代数 Riccati 方程），用$\{X_k\}$序列来求解收敛的 X ；（2）牛顿法的应用很广泛。

$$X_{k+1}=\hat{A}^T X_k(I+\hat{G}X_k)^{-1}\hat{A}+\hat{H} \qquad (5\text{-}1)$$

5.2 结构保持加倍算法

一种新的算法：结构保持加倍算法[1]能够产生序列为$\{X_{2^k}\}$的稳定解 X 而不是产生序列$\{X_k\}$。同时，由于 Hamiltonian 矩阵 $\mathcal{H}$ 的结构保全性，能够产生高精度的解。因此，SDA 具有快速性和准确性。

算法 1：结构保持加倍算法

输入：$\mathcal{H}=\begin{bmatrix} A & -G \\ -H & -A^T \end{bmatrix}\in \mathrm{H}$，其中 $\sigma(\mathcal{H})\bigcap \mathrm{Im}=\Phi;\in$

输出：对 CARE 的稳定解 $X=X^T\geq 0$

1 计算

$\hat{A}_0 \leftarrow I+2\hat{\gamma}\left(A_{\gamma}+GA_{\gamma}^{T}H\right)^{-1};$

$\hat{G}_0 \leftarrow 2\hat{\gamma}A_{\gamma}^{-1}G\left(A_{\gamma}^{T}+HA_{\gamma}^{-1}G\right)^{-1};$

$\hat{H}_0 \leftarrow 2\hat{\gamma}\left(A^T\hat{\gamma}+HA_{\gamma}^{-1}G\right)^{-1}HA_{\gamma}^{-1};$

$j\leftarrow 0;$

2 计算到收敛为止

计算

$\hat{A}_{j+1} \leftarrow \hat{A}_j\left(I+\hat{G}_j\hat{H}_j\right)^{-1}\hat{A}_j;$

$\hat{G}_{j+1} \leftarrow \hat{G}_j+\hat{A}_j\hat{G}_j\left(I+\hat{H}_j;\hat{G}_j\right)^{-1}\hat{A}_j^T;$

$\hat{H}_{j+1} \leftarrow \hat{H}_j+\hat{A}_j^T\left(I+\hat{H}_j;\hat{G}_j\right)^{-1}\hat{H}_j\hat{A}_j;$

If $\left\|\hat{H}_j=\hat{H}_{j-1}\leq \grave{o}\hat{H}_j\right\|,$

Stop;

End

3 设置$X\leftarrow \hat{H}_j$

第六章 更新旋转矩阵的数值方法

这章将介绍参考文献[3]，应用角速度更新旋转矩阵的方法。

6.1 应用角速度更新旋转矩阵

线速度与角速度的变换公式为：

$$v = \frac{dr}{dt} = \omega(t) \times r(t) \tag{6-1}$$

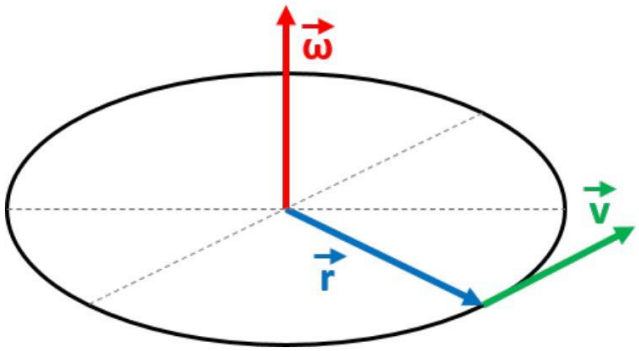


图 6-1 线速度、角速度和旋转矢量

其中 $\omega(t)$ 是转速矢量。我们可以通过整合上述运动学方程来更新位置矢量。

$$\begin{aligned} r(t) &= r(0) + \int_0^t d\omega(\tau) \times r(\tau) d\tau \\ &= r(0) + \int_0^t d\theta(\tau) \times r(\tau) \end{aligned} \tag{6-2}$$

方法是将方程（6-2）应用于 R 矩阵的行或列，将它们视为旋转矩阵。

我们遇到的第一个障碍是我们要跟踪的向量和旋转向量未在同一参照系中测量。理想情况下，我们希望在地球参照系中跟踪飞机的轴线，但陀螺仪的测量是在机身参照系中进行的。通过识别旋转中的对称性，可以轻松解决问题。在地面参照系中，地球坐标系的旋转与地球坐标系中地面的旋转相同。因此，我们可以通过翻转陀螺仪信号的符号来跟踪地球轴线。为了方便起见，我们可以将符号翻转过来，并交换叉积中的因子：

$$
\begin{aligned}
&r_{earth}\ (t) = r_{earth}\ (0) + \int_0^t r_{earth}\ (\tau) \times d\theta(\tau) \\
&d\theta(\tau) = \omega(\tau) d\tau \\
&r_{earth}\ (t) = \text{one of the earth axes, as viewed from the plane}
\end{aligned}
\qquad (6\text{-}3)
$$

公式（6-3）中的向量是公式（3-4）中 R 矩阵的行。接下来的问题是如何方便地实现公式（6-3）。我们采取与 Mahoney[1]使用的相同的矩阵方法。首先回到公式（6-3）的微分形式：

$$
\begin{aligned}
&r_{earth}\ (t+dt) = r_{earth}\ (t) + r_{earth}\ (t) \times d\theta(t) \\
&d\theta(t) = \omega(t) d\tau
\end{aligned}
\qquad (6\text{-}4)
$$

将比例积分漂移补偿反馈控制器得出的校正转速与陀螺仪的测量值相加，以得到真实转速的最佳估计值：

$$
\begin{aligned}
&\omega(t) = \omega_{gyro}\ (t) + \omega_{correction}\ (t) \\
&\omega_{gyro}\ (t) = \text{三个陀螺仪的测量} \\
&\omega_{correction}\ (t) = \text{陀螺仪校正}
\end{aligned}
\qquad (6\text{-}5)
$$

GPS 和加速度计参考矢量被用来计算旋转误差，该误差通过反馈控制器和公式（6-5）进行旋转矩阵的更新。

因为旋转矩阵的列向量表示空间中三个正交方向的向量。应用上述公式来更新旋转矩阵：

$$
\begin{aligned}
&R(t+dt) = R(t) \begin{bmatrix} 1 & -d\theta_z & d\theta_y \\ d\theta_z & 1 & d\theta_x \\ -d\theta_y & d\theta_x & 1 \end{bmatrix} \\
&d\theta_x = \Omega_x dt \\
&d\theta_y = \Omega_y dt \\
&d\theta_z = \Omega_z dt
\end{aligned}
\qquad (6\text{-}6)
$$

6.2 旋转矩阵的重正交化

由积分产生的数值误差会逐渐降低旋转矩阵的正交性，因此需要在每个积分步骤后进行正交化和重正交。在每一个步积分后进行重正交化。

假设我们有一个旋转矩阵 R：

$$R=\begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} \tag{6-7}$$

计算 X 和 Y 的点积，它应该为零。其结果衡量了 X 和 Y 的行向对方旋转的程度。

$$X=\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix} \tag{6-8}$$

$$Y=\begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix} \tag{6-9}$$

$$error = X \cdot Y = X^T Y = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \end{bmatrix} \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix} \tag{6-10}$$

我们将一半的误差分摊到 X 和 Y 行，并通过交叉耦合将 X 和 Y 行大致旋转到相反的方向。

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}_{orthogonal} = X_{orthogonal} = X - \frac{error}{2} Y \tag{6-11}$$

$$\begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}_{orthogonal} = Y_{orthogonal} = Y - \frac{error}{2} X \tag{6-12}$$

下一步是调整矩阵的 Z 行，使其与 X 和 Y 正交，我们可以简单地让新的 Z 是 X 和 Y 行的叉积：

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}_{orthogonal} = Z_{orthogonal} = X_{orthogonal} \times Y_{orthogonal} \tag{6-13}$$

最后一步是进行矢量归一化，这可以通过使用 X 、Y 和 Z 各自的范数重新缩放来完成：

一种方法是将每一行的各个元素除以该行元素平方之和的平方根。

$$X_{normalized} = \frac{X_{orthogonal}}{\left\| X_{orthogonal} \right\|} \tag{6-14}$$

$$Y_{normalized} = \frac{Y_{orthogonal}}{\left\| Y_{orthogonal} \right\|} \tag{6-15}$$

$$Z_{normalized} = \frac{Z_{orthogonal}}{\left\| Z_{orthogonal} \right\|} \tag{6-16}$$

另一个更简单的方法是意识到幅值永远和 1 很接近，因此我们可以使用泰勒展开，由此产生的行向量的幅值调整方程为：

$$X_{normalized} = \frac{1}{2}\left(3 - X_{orthogonal} \cdot X_{orthogonal}\right) X_{orthogonal} \tag{6-17}$$

$$Y_{normalized} = \frac{1}{2}\left(3 - Y_{orthogonal} \cdot Y_{orthogonal}\right) Y_{orthogonal} \tag{6-18}$$

$$Z_{normalized} = \frac{1}{2}\left(3 - Z_{orthogonal} \cdot Z_{orthogonal}\right) Z_{orthogonal} \tag{6-19}$$

第七章 仿真结果

7.1 轨迹跟踪

四旋翼无人机跟踪以下圆形轨迹进行仿真。

$$\left[p_x, p_y, p_z\right] = [0.5\cos(0.25\pi t), 0.5\sin(0.25\pi t), -0.1t] \tag{7-1}$$

$$\left[v_x, v_y, v_z\right] = [-0.5\sin(0.25\pi t), 0.5\cos(0.25\pi t), -0.1] \tag{7-2}$$

LQR 控制器的期望设定值给定为：

$$\mathbf{x}_0 = \left[0,0,0,0,0,0,v_x,v_y,v_z,p_x,p_y,p_z\right] \tag{7-3}$$

请注意，所需的零姿态 $(\varphi,\theta,\psi) = (0^\circ,0^\circ,0^\circ)$ 实际上违反了四旋翼飞行器的动力学，因为只有在倾斜角度后才能改变位置。这意味着四旋翼飞行器需要很大的倾角来跟踪快速变化的轨迹。在这种情况下，平衡点可能远离 $(0^\circ,0^\circ,0^\circ)$，使得线性化不合理并破坏了稳定性。解决方案是规划遵循四旋翼飞行器动力学约束的轨迹。

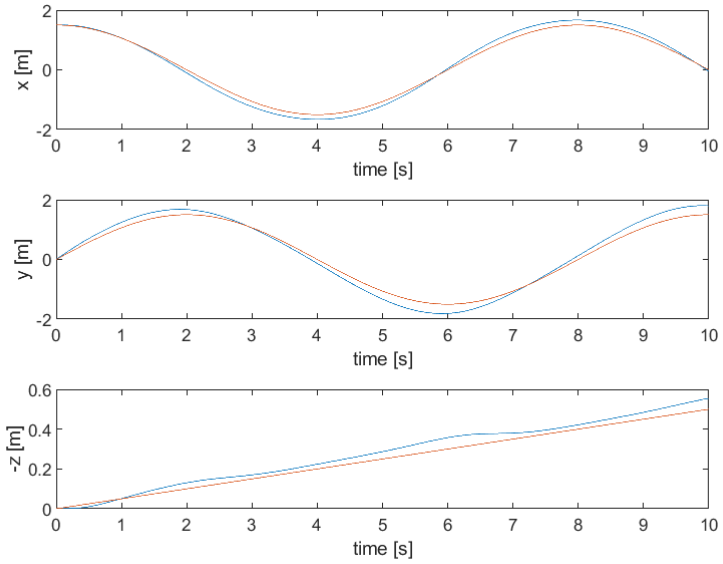


图 7-1 位置跟踪（蓝：真实状态，橙：期望状态）

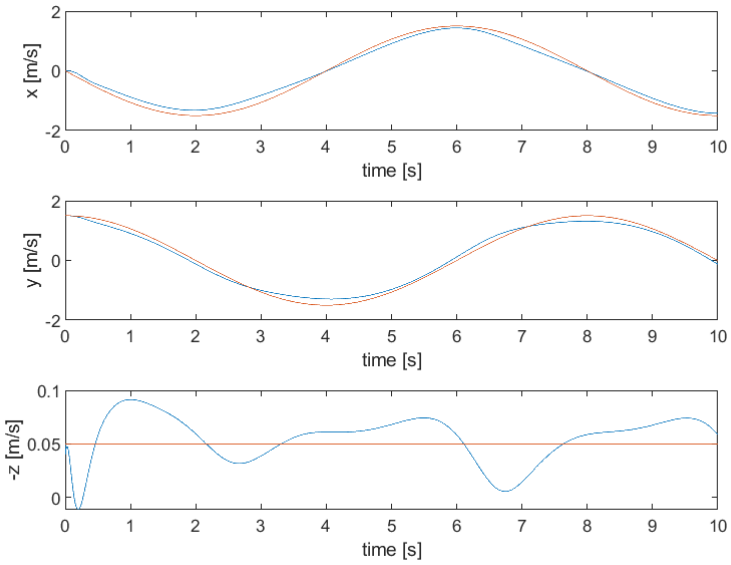


图 7-2 速度跟踪（蓝：真实状态，橙：期望状态）

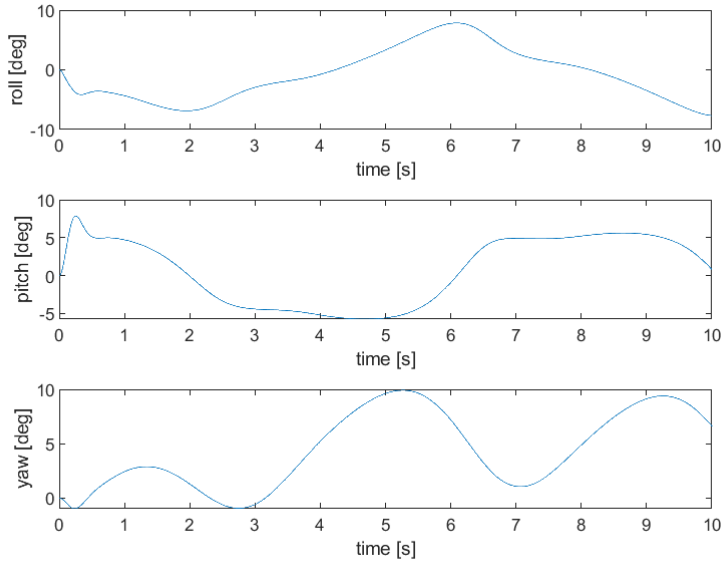


图 7-2 姿态

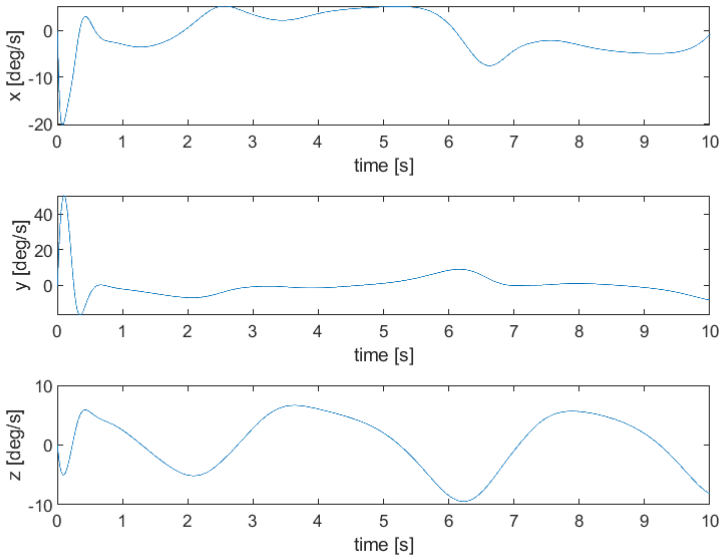
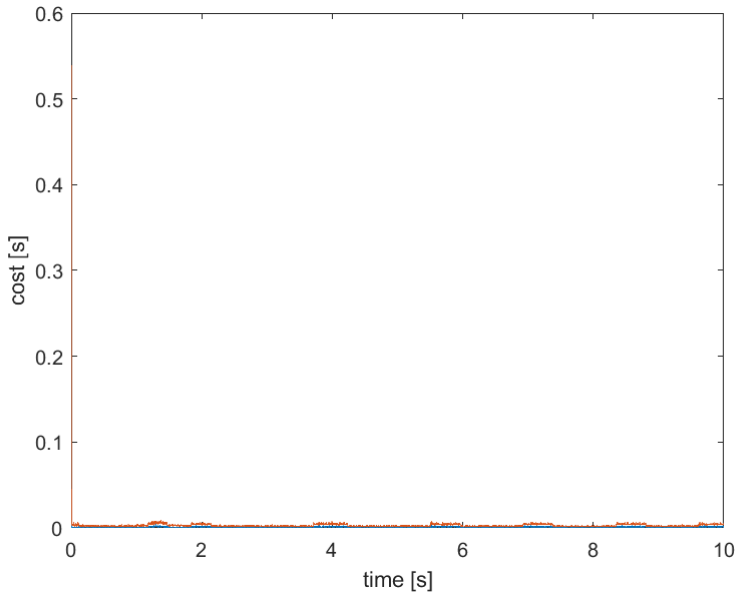


图 7-4 角速度

7.2 结构保持加倍算法的性能结构

我们用 MATLAB 的内置函数 care 和 SDA 进行了仿真和求解 CARE。仿真结果显示，SDA 比 MATLAB 的 care 函数快 5.1 倍，而且精度略高，如图 7-5 和图 7-6。



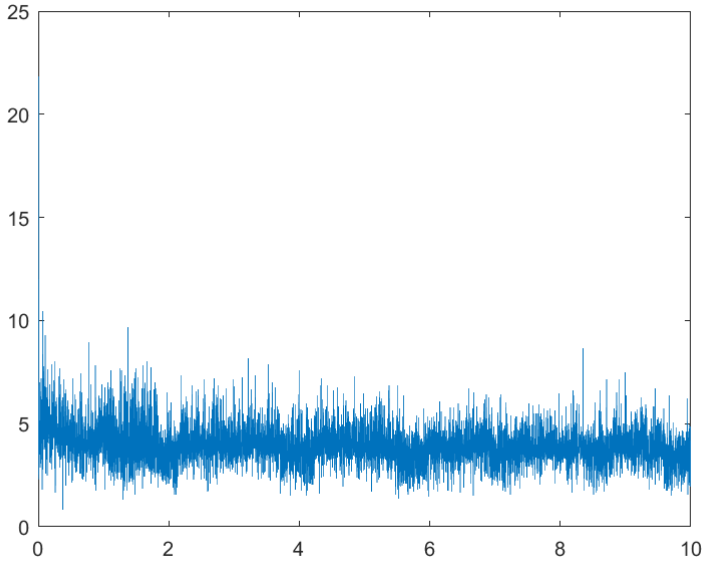


图 7-5 时间成本比较（蓝：SDA，橙：MATLAB 函数）

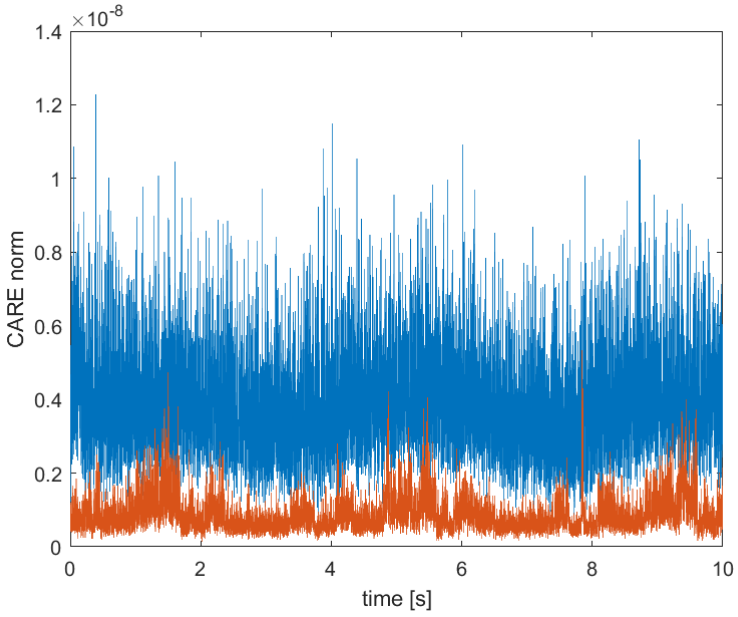


图 7-6 精度比较（蓝：SDA，橙：MATLAB 函数）

参考文献

[1] W.-W.Lin E.K.-W.Chu, H.-Y.Fan. A structure-preserving doubling algorithm for continuous-time algebraic riccati equations. Linear Algebra and its Applications.

[2] Alan J. Laub. A schur method for solving algebraic riccati equations. IEEE Transactions on Automatic Control.

[3] William Premerlani and Paul Bizard. Direction cosine matrix imu: Theory.

[4] Francesco Sabatino. Quadrotor control: modeling, nonlinear control design, and simulation.

[5] N. Harris McClamroch Taeyoung Lee, Melvin Leok. Geometric tracking control of a quadrotor uav on se(3). IEEE Conference on Decision and Control.