微小型固定翼无人机飞行控制设计与实践

第9章 图像伺服对接控制实验

全 权
qq_buaa@buaa.edu.cn
自动化科学与电气工程学院

北京航空航天大学
BEIHANG UNIVERSITY

可靠飞行控制研究组
RELIABLE FLIGHT CONTROL GROUP

大纲

1. 实验原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 硬件在环仿真实验
6. 本章小结

实验原理

常见的空中回收方法

从母舰下放缆绳和缆绳末端的伞套，无人机被控使得机上锥管插入伞套完成锁定

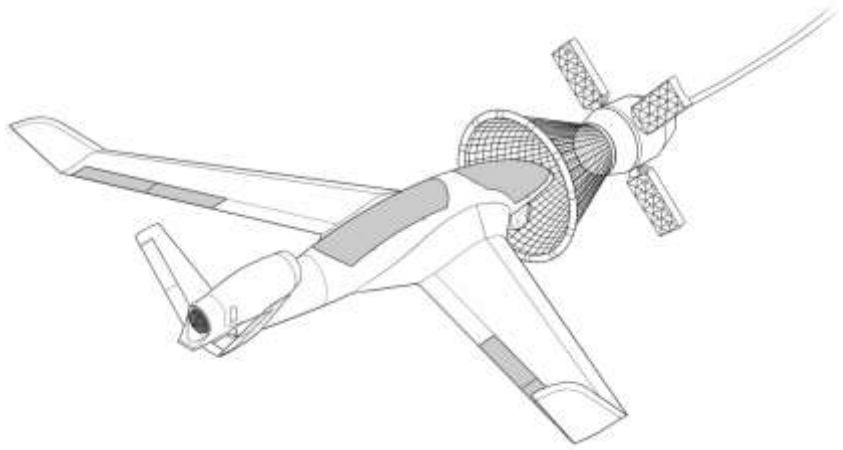


图. 无人机空中回收装置

缆绳回收到指定位置使得无人机能被母舰机械臂抓回母舰

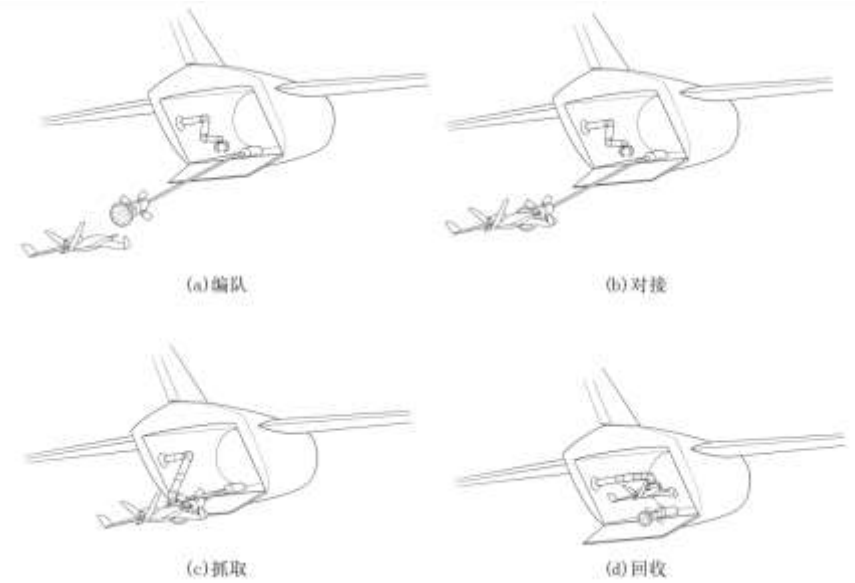


图. 无人机回收过程

实验原理

图像伺服控制基本原理

图像伺服控制是利用相机数据来控制机器人运动的方法。相机固定安装在机器人上，从而使得相机随着机器人的运动而运动，进而产生变化的图像。

图像伺服控制的目的是使得图像跟踪误差最小，而**图像跟踪误差** $\mathrm{e}(t)$ 定义为：

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(t), \mathbf{a}) - \mathbf{s}^* \tag{1.1}$$

- $\mathbf{m}(t)$：图像测量信息（例如图像坐标系中的目标点坐标或目标物体质心的坐标）。
- $\mathbf{a}$：用来表征系统的先验知识的参数（例如相机的内参矩阵或者物体 3D 模型。
- $\mathbf{s}^*$：表示这些视觉特征的期望值。

实验原理

图像伺服控制基本原理

首先给出视觉特征 s 与相机运动速度 $\bar{\mathbf{v}}_c = [\mathbf{v}_c^T, \boldsymbol{\omega}_c^T]^T$ 的关系：

$$\dot{\mathbf{s}} = \mathbf{L}_s \bar{\mathbf{v}}_c \tag{1.2}$$

式中 L_s 被称为与 s 相关的特征雅可比矩阵，v_c 是相机坐标系原点的瞬时线速度，ω_c 是相机坐标系的角速度。

根据式 (1.1) 和 (1.2)，可得误差 e(t) 与相机运动速度 v_c 的关系：

$$\dot{\mathbf{e}} = \mathbf{L}_e \bar{\mathbf{v}}_c \tag{1.3}$$

式中L_e=L_s。将 v_c 作为机器人的控制器输入，如果希望使误差 e(t) 按照指数收敛（即 $\dot{\mathbf{e}} = -\lambda \mathbf{e}$），则可以令

$$\bar{\mathbf{v}}_c = -\lambda \mathbf{L}_e^+ \mathbf{e} \tag{1.4}$$

式中 $\mathbf{L}_e^+ = (\mathbf{L}_e^T \mathbf{L}_e)^{-1} \mathbf{L}_e^T$ 。

实验原理

图像伺服控制基本原理

在实际应用中，需要推导或估计出 L_s，进而完成图像伺服控制器的设计。接下来进行经典的图像伺服控制问题中 L_s 的推导。

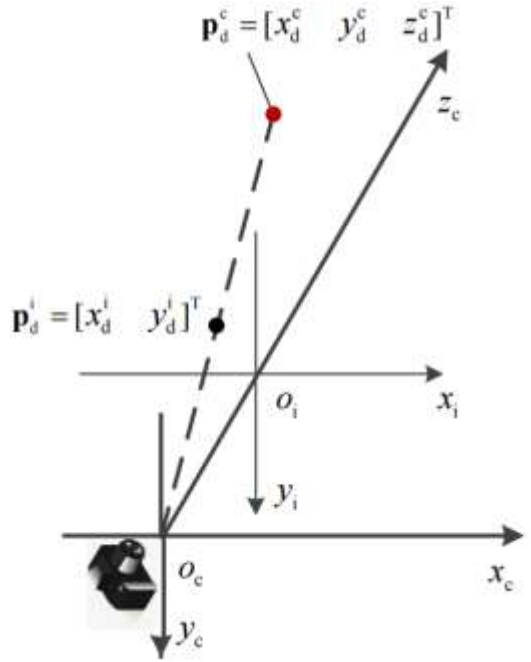


图. 图像伺服控制涉及的坐标系

如图所示，$o_c x_c y_c z_c$ 是相机坐标系，记为“c”，$o_i x_i y_i z_i$ 是图像坐标系，记为“i”。

将目标点（本章指锥套圆心，因此下标记为“d”）在相机坐标系中的坐标定义为 $\mathbf{p}_d^c=[x_d^c \; y_d^c \; z_d^c]^T$。

它在图像中被投影为一个二维点，坐标为 $\mathbf{p}_d^i=[x_d^i \; y_d^i]^T$。

实验原理

图像伺服控制基本原理

图像中的图像跟踪误差 e 定义为

$$\mathbf{e}=\begin{bmatrix}e_x\\e_y\end{bmatrix}=\begin{bmatrix}x_{\mathrm{d}}^{\mathrm{i}}-x_{\mathrm{o}}^{\mathrm{i}}\\y_{\mathrm{d}}^{\mathrm{i}}-y_{\mathrm{o}}^{\mathrm{i}}\end{bmatrix} \tag{1.5}$$

式中 $(x_o^i,\ y_o^i)$是图像中的一个固定点，称为图像的**收敛点**。

对上式求导可得：

$$\dot{\mathbf{e}}=\begin{bmatrix}\dot{e}_x\\\dot{e}_y\end{bmatrix}=\begin{bmatrix}\dot{x}_{\mathrm{d}}^{\mathrm{i}}\\\dot{y}_{\mathrm{d}}^{\mathrm{i}}\end{bmatrix} \tag{1.6}$$

根据小孔成像原理可得

$$\begin{cases}x_{\mathrm{d}}^{\mathrm{i}}=\dfrac{x_{\mathrm{d}}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}}\\[2ex]y_{\mathrm{d}}^{\mathrm{i}}=\dfrac{y_{\mathrm{d}}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}}\end{cases} \tag{1.7}$$

实验原理

图像伺服控制基本原理

对式子微分得：

$$\begin{cases} \dot{x}_{\mathrm{d}}^{\mathrm{i}} = \dfrac{\dot{x}_{\mathrm{d}}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} - \dfrac{x_{\mathrm{d}}^{\mathrm{c}}\dot{z}_{\mathrm{d}}^{\mathrm{c}}}{{z_{\mathrm{d}}^{\mathrm{c}}}^2} = \dfrac{\dot{x}_{\mathrm{d}}^{\mathrm{c}} - x_{\mathrm{d}}^{\mathrm{i}}\dot{z}_{\mathrm{d}}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} \\ \dot{y}_{\mathrm{d}}^{\mathrm{i}} = \dfrac{\dot{y}_{\mathrm{d}}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} - \dfrac{y_{\mathrm{d}}^{\mathrm{c}}\dot{z}_{\mathrm{d}}^{\mathrm{c}}}{{z_{\mathrm{d}}^{\mathrm{c}}}^2} = \dfrac{\dot{y}_{\mathrm{d}}^{\mathrm{c}} - y_{\mathrm{d}}^{\mathrm{i}}\dot{z}_{\mathrm{d}}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} \end{cases} \tag{1.8}$$

又有如下关系

$$\dot{\mathbf{p}}_{\mathrm{d}}^{\mathrm{c}} = -\mathbf{v}_{\mathrm{c}}^{\mathrm{d}} - \omega_{\mathrm{c}}^{\mathrm{d}} \times \mathbf{p}_{\mathrm{d}}^{\mathrm{c}} \tag{1.9}$$

即

$$\begin{cases} \dot{x}_{\mathrm{d}}^{\mathrm{c}} = -v_{\mathrm{d},x}^{\mathrm{c}} - \omega_{\mathrm{d},y}^{\mathrm{c}} z_{\mathrm{d}}^{\mathrm{c}} + \omega_{\mathrm{d},z}^{\mathrm{c}} y_{\mathrm{d}}^{\mathrm{c}} \\ \dot{y}_{\mathrm{d}}^{\mathrm{c}} = -v_{\mathrm{d},y}^{\mathrm{c}} - \omega_{\mathrm{d},z}^{\mathrm{c}} x_{\mathrm{d}}^{\mathrm{c}} + \omega_{\mathrm{d},x}^{\mathrm{c}} z_{\mathrm{d}}^{\mathrm{c}} \\ \dot{z}_{\mathrm{d}}^{\mathrm{c}} = -v_{\mathrm{d},z}^{\mathrm{c}} - \omega_{\mathrm{d},x}^{\mathrm{c}} y_{\mathrm{d}}^{\mathrm{c}} + \omega_{\mathrm{d},y}^{\mathrm{c}} x_{\mathrm{d}}^{\mathrm{c}} \end{cases} \tag{1.10}$$

将式 (1.10)代入式 (1.8)，再结合式 (1.6)，可得$\dot{e}$ 和 $\boldsymbol{v}_{\boldsymbol{d}}^{\boldsymbol{c}}$，$\boldsymbol{\omega}_{\boldsymbol{d}}^{\boldsymbol{c}}$ 之间的关系为

$$\dot{\mathbf{e}} = \underbrace{\begin{bmatrix} -\frac{1}{z_{\mathrm{d}}^{\mathrm{c}}} & 0 & \frac{x_{\mathrm{d}}^{\mathrm{i}}}{z_{\mathrm{d}}^{\mathrm{c}}} & x_{\mathrm{d}}^{\mathrm{i}} y_{\mathrm{d}}^{\mathrm{i}} & -(1+{x_{\mathrm{d}}^{\mathrm{i}}}^2) & y_{\mathrm{d}}^{\mathrm{i}} \\ 0 & -\frac{1}{z_{\mathrm{d}}^{\mathrm{c}}} & \frac{y_{\mathrm{d}}^{\mathrm{i}}}{z_{\mathrm{d}}^{\mathrm{c}}} & 1+{y_{\mathrm{d}}^{\mathrm{i}}}^2 & -x_{\mathrm{d}}^{\mathrm{i}} y_{\mathrm{d}}^{\mathrm{i}} & -x_{\mathrm{d}}^{\mathrm{i}} \end{bmatrix}}_{\mathbf{L_s}} \begin{bmatrix} v_{\mathrm{d},x}^{\mathrm{c}} \\ v_{\mathrm{d},y}^{\mathrm{c}} \\ v_{\mathrm{d},z}^{\mathrm{c}} \\ \omega_{\mathrm{d},x}^{\mathrm{c}} \\ \omega_{\mathrm{d},y}^{\mathrm{c}} \\ \omega_{\mathrm{d},z}^{\mathrm{c}} \end{bmatrix}$$

实验原理

目标识别及定位

目标识别及定位解决的是“**目标在哪儿**”的问题，当前常用的方法有基于标志点检测的识别定位和基于神经网络的识别定位等。

基于标志点检测的视觉导航需要提前在目标上布置标志点，在空中加油对接场景中，标志点可以采用 LED，或者采用 850nm 窄波红外光源。结合光学滤波与算法滤波可以将标志点很好地提取出来。

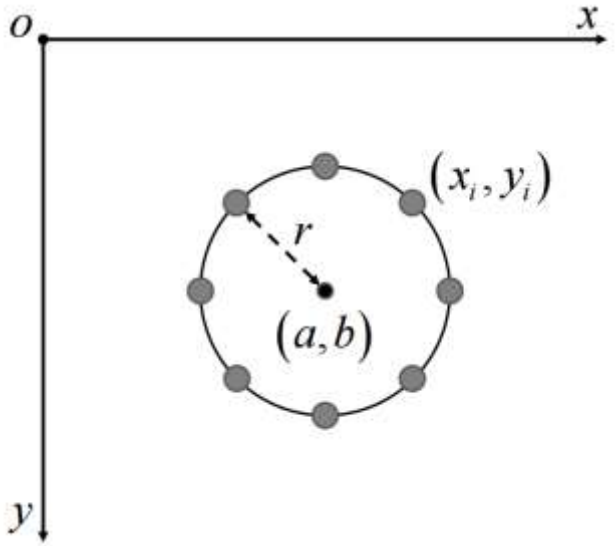


图. 锥套标志点布置

设锥套标志点被布置为圆形，如图所示，其在图像上的投影满足表达式

$$(x-a)^2+(y-b)^2=r^2$$

式中坐标 (a,b) 表示锥套的圆心位置，r 为锥套半径。

假设可以检测到 N 个标志点 $p_i = (x_i, y_i) \in \mathbb{B}$，式中 $\mathbb{B}$ 为所有圆标志点的坐标集合。因此相应的残差$\boldsymbol{\varepsilon_i}$可以有

$$\boldsymbol{\varepsilon_i} = (x_i-a)^2+(y_i-b)^2$$

实验原理

目标识别及定位

取残差的平方作为这里拟合的代价函数，即

$$J=\sum_{i=1,\cdots,N}{\varepsilon_i}^2=\sum_{i=1,\cdots,N}\left((x_i-a)^2+(y_i-b)^2-r^2\right)^2$$

由最小二乘原理可以得到，当拟合结果为最优时应该有代价函数偏导为 0，即

$$\frac{\partial J}{\partial a}=\frac{\partial J}{\partial b}=\frac{\partial J}{\partial r}=0$$

计算化简并消去高阶项有

$$\begin{cases}(\overline{x}^2-\overline{x^2})a+(\overline{x}\cdot\overline{y}-\overline{xy})b=\frac{1}{2}(\overline{x^2}\cdot\overline{x}+\overline{x}\cdot\overline{y^2}-\overline{x^3}-\overline{xy^2})\\(\overline{x}\cdot\overline{y}-\overline{xy})a+(\overline{y}^2-\overline{y^2})b=\frac{1}{2}(\overline{x^2}\cdot\overline{y}+\overline{y}\cdot\overline{y^2}-\overline{x^2y}-\overline{y^3})\end{cases}$$

式中

$$\overline{x^m y^n}=\frac{\sum_{i=1}^{N}x_i^m y_i^n}{N}\qquad m,n\in 0,1,2$$

实验原理

目标识别及定位

综合以上各式可以得到锥套圆心坐标即半径：

$$\begin{cases} a = \dfrac{(\overline{x^2}\cdot\bar{x}+\bar{x}\cdot\overline{y^2}-\overline{x^3}-\overline{xy^2})(\overline{y^2}-\overline{y^2})-(\overline{x^2}\cdot\bar{y}+\bar{y}\cdot\overline{y^2}-\overline{x^2y}-\overline{y^3})(\bar{x}\cdot\bar{y}-\overline{xy})}{2(\overline{x^2}-\overline{x^2})(\overline{y^2}-\overline{y^2})-2(\bar{x}\cdot\bar{y}-\overline{xy})^2} \\ b = \dfrac{(\overline{x^2}\cdot\bar{y}+\bar{y}\cdot\overline{y^2}-\overline{x^2y}-\overline{y^3})(\overline{x^2}-\overline{x^2})-(\overline{x^2}\cdot\bar{x}+\bar{x}\cdot\overline{y^2}-\overline{x^3}-\overline{xy^2})(\bar{x}\cdot\bar{y}-\overline{xy})}{2(\overline{x^2}-\overline{x^2})(\overline{y^2}-\overline{y^2})-2(\bar{x}\cdot\bar{y}-\overline{xy})^2} \\ r = \sqrt{a^2-2\bar{x}a+b^2-2\bar{y}b+\overline{x^2}+\overline{y^2}} \end{cases}$$

之后，基于小孔成像原理以及锥套的先验知识，即可计算出深度信息。

基于神经网络的识别定位可以采用 YOLO 神经网络直接进行锥套的识别，不需要布置标志点，但是需要提前进行采集大量的训练数据，并消耗较长的时间和较多的计算资源进行训练，使用时也需要边缘计算平台具有较强算力。基于 YOLO 神经网络进行视觉导航，可以基于小孔成像原理以及锥套先验知识进行深度信息计算，也可以直接融合深度图信息得到距离。

本章给出的例程均使用 YOLO 神经网络进行视觉导航。

实验原理

对接控制

对接控制解决的是如何利用视觉导航给出的信息完成对接的问题，对接控制算法的开发需要进行模型建立和控制器设计。

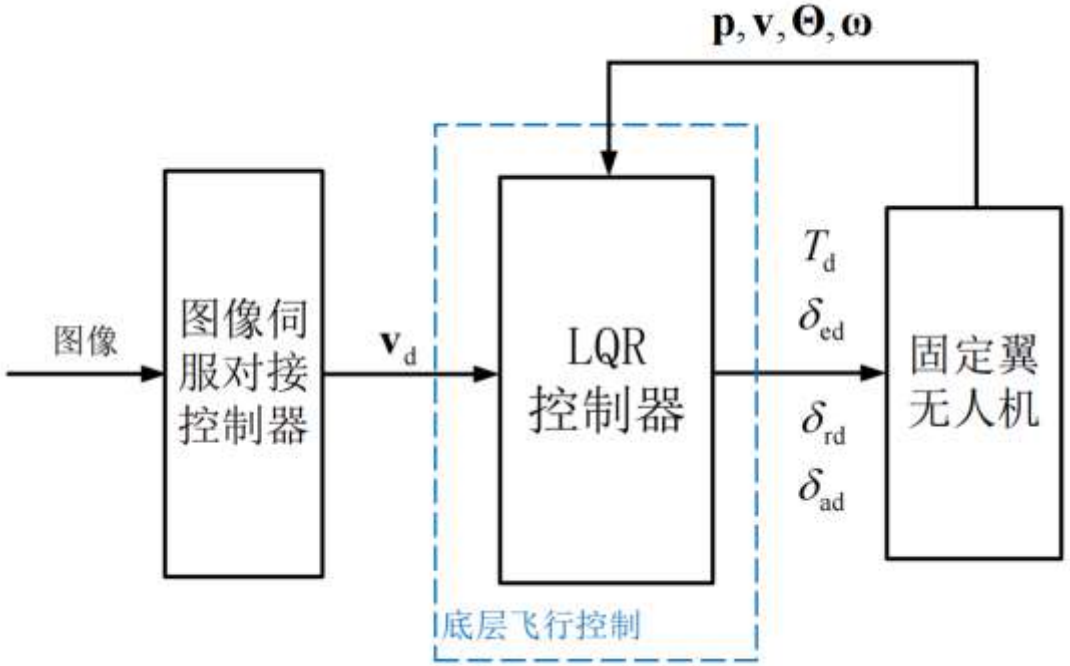


图. 图像伺服对接控制结构

实验原理

对接控制 - 模型建立

1.坐标系

在建立对接阶段系统模型时，需要定义五个坐标系：地面坐标系 $(o_g x_g y_g z_g)$，受油机坐标系 $(o_r x_r y_r z_r)$，加油机坐标系 $(o_t x_t y_t z_t)$，相机坐标系 $(o_c x_c y_c z_c)$ 和相对坐标系 $(o_{re} x_{re} y_{re} z_{re})$。

在对接阶段主要考虑加油机与受油机之间的相对位置和姿态，加油机坐标系 (运动惯性坐标系) 比地面坐标系 (固定惯性坐标系) 更便于建模和分析。

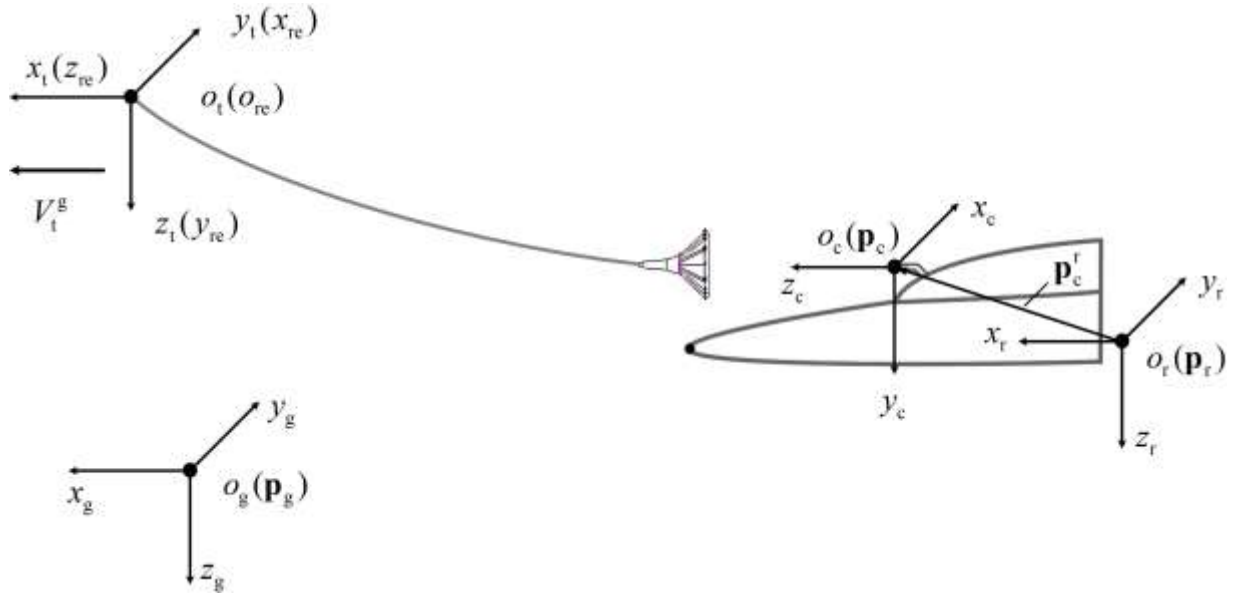


图. 坐标系

实验原理

对接控制 - 模型建立

为了进一步简化不同坐标系之间的坐标转换，地面坐标系的坐标轴方向也选择与加油机的水平移动方向相同。此外，$\boldsymbol{R}_{\mathrm{t}}^{\mathrm{r}}$ 表示用于描述受油机坐标系 'r' 相对于加油机坐标系 't' 的角度关系的旋转矩阵。这里的符号运算规则定义为

$$\begin{cases} \mathbf{R}_j^i \mathbf{x}_k^j = \mathbf{x}_k^i \\ \mathbf{x}_i^j - \mathbf{x}_k^j = \mathbf{x}_k^i \end{cases}$$

式中，$i, j \in \{\mathrm{g}, \mathrm{r}, \mathrm{t}, \mathrm{c}, \mathrm{re}, \mathrm{d}\}$表示它所在的坐标系，$k \in \{\mathrm{g}, \mathrm{r}, \mathrm{t}, \mathrm{c}, \mathrm{re}, \mathrm{d}\}$ 表示它属于哪个对象。例如，$V_{\mathrm{t}}^{\mathrm{g}}$是加油机在地面坐标系中的前向飞行速度，$\mathbf{v}_{\mathrm{d}}^{\mathrm{c}}$ 是锥套在相机坐标系中的速度，$\mathbf{p}_{\mathrm{c}}^{\mathrm{r}}$ 是从受油机坐标到相机坐标的差矢量，即图中的

$$\mathbf{p}_{\mathrm{r}}^{\mathrm{c}} = \mathbf{p}_{\mathrm{c}}^{\mathrm{t}} - \mathbf{p}_{\mathrm{r}}^{\mathrm{t}}$$

实验原理

对接控制 - 模型建立

2.受油机模型

在对接阶段，受油机的俯仰角变化范围很小，基准状态为等速平飞，其姿态满足水平侧滑条件，即滚转角和侧滑角满足 $\boldsymbol{\phi}=\boldsymbol{\beta}=0$，迎角 $\boldsymbol{\alpha}$，航迹角 $\boldsymbol{\gamma}$ 和俯仰角 θ 满足 $\boldsymbol{\theta}=\boldsymbol{\gamma}+\boldsymbol{\alpha}$。在这种情况下，受油机在加油机坐标系下的运动方程可以被解耦成**横侧向通道运动**和**纵向通道运动**。

根据第四章，可以将受油机的运动状态方程简化表示为

$$\dot{\mathbf{x}}_{\mathrm{r}}=\mathbf{f}\left(\mathbf{x}_{\mathrm{r}},\mathbf{u}_{\mathrm{r}}\right)$$

式中，$\mathbf{x}_{\mathrm{r}}=[\,x_{\mathrm{r}}\ y_{\mathrm{r}}\ h_{\mathrm{r}}\ \phi\ \theta\ \psi\ V_{\mathrm{r}}^{\mathrm{g}}\ \alpha\ \beta\ \omega_{x_{\mathrm{b}}}\ \omega_{y_{\mathrm{b}}}\ \omega_{z_{\mathrm{b}}}\,]^{\mathrm{T}}$ 表示受油机的状态，分别包括位置、欧拉角、气动角和角速率。向量 $\mathbf{u}_{\mathrm{r}}=[T\ \delta_{\mathrm{e}}\ \delta_{\mathrm{a}}\ \delta_{\mathrm{r}}]^{\mathrm{T}}$ 表示控制输入，分别包括油门、升降舵、副翼和方向舵。

实验原理

对接控制 - 模型建立

类比第四章得到

$$\Delta\dot{\mathbf{x}}_{\mathrm{r}} = \mathbf{A}\Delta\mathbf{x}_{\mathrm{r}} + \mathbf{B}\Delta\mathbf{u}_{\mathrm{r}}$$

式中 $\mathbf{x}_{\mathrm{r}}^{*}$ 为配平状态，$\Delta\mathbf{x}_{\mathrm{r}} = \mathbf{x}_{\mathrm{r}} - \mathbf{x}_{\mathrm{r}}^{*}$ 为偏离配平状态的小扰动状态量。在此基础上进而得到解耦的小扰动状态方程

$$\begin{cases} \Delta\dot{\mathbf{x}}_{\mathrm{rlon}} = \mathbf{A}_{\mathrm{rlon}}\Delta\mathbf{x}_{\mathrm{rlon}} + \mathbf{B}_{\mathrm{rlon}}\Delta\mathbf{u}_{\mathrm{rlon}} \\ \Delta\dot{\mathbf{x}}_{\mathrm{rlat}} = \mathbf{A}_{\mathrm{rlat}}\Delta\mathbf{x}_{\mathrm{rlat}} + \mathbf{B}_{\mathrm{rlat}}\Delta\mathbf{u}_{\mathrm{rlat}} \end{cases}$$

式中 $\Delta\mathbf{x}_{\mathrm{rlon}} = [\Delta x_{\mathrm{r}}\ \Delta h_{\mathrm{r}}\ \Delta\theta\ \Delta V_{\mathrm{r}}^{\mathrm{g}}\ \Delta\alpha\ \Delta\omega_{y_{\mathrm{b}}}]^{\mathrm{T}}$ 为纵向通道状态向量，$\Delta\mathbf{u}_{\mathrm{rlon}} = [\Delta T\ \Delta\delta_{\mathrm{e}}]^{\mathrm{T}}$ 为纵向通道输入，$\mathbf{A}_{\mathrm{rlon}}$、$\mathbf{B}_{\mathrm{rlon}}$ 为相应的纵向通道状态方程矩阵。

相似的 $\Delta\mathbf{x}_{\mathrm{rlat}} = [\Delta y_{\mathrm{r}}\ \Delta\phi\ \Delta\psi\ \Delta\beta\ \Delta\omega_{x_{\mathrm{b}}}\ \Delta\omega_{z_{\mathrm{b}}}]^{\mathrm{T}}$ 为横侧向通道状态向量，$\Delta\mathbf{u}_{\mathrm{rlat}} = [\Delta\delta_{\mathrm{a}}\ \Delta\delta_{\mathrm{r}}]^{\mathrm{T}}$ 为横侧向通道输入，$\mathbf{A}_{\mathrm{rlat}}$、$\mathbf{B}_{\mathrm{rlat}}$ 为相应的横侧向通道状态方程矩阵。

实验原理

对接控制 - 模型建立

3. 图像伺服模型

右图中$o_t x_t y_t z_t$是加油机坐标系，记为“t”；$o_c x_c y_c z_c$是相机坐标系，记为“c”，从“t”到“c”的转换矩阵为 Rct；$o_i x_i y_i z_i$是图像坐标系，记为“i”。可得

$$\dot{\mathbf{e}} = \underbrace{\begin{bmatrix} -\frac{1}{z_d^c} & 0 & \frac{x_d^i}{z_d^c} & x_d^i y_d^i & -(1+x_d^{i\,2}) & y_d^i \\ 0 & -\frac{1}{z_d^c} & \frac{y_d^i}{z_d^c} & 1+y_d^{i\,2} & -x_d^i y_d^i & -x_d^i \end{bmatrix}}_{\mathbf{L}_s} \begin{bmatrix} v_{d,x}^c \\ v_{d,y}^c \\ v_{d,z}^c \\ \omega_{d,x}^c \\ \omega_{d,y}^c \\ \omega_{d,z}^c \end{bmatrix} \tag{1.22}$$

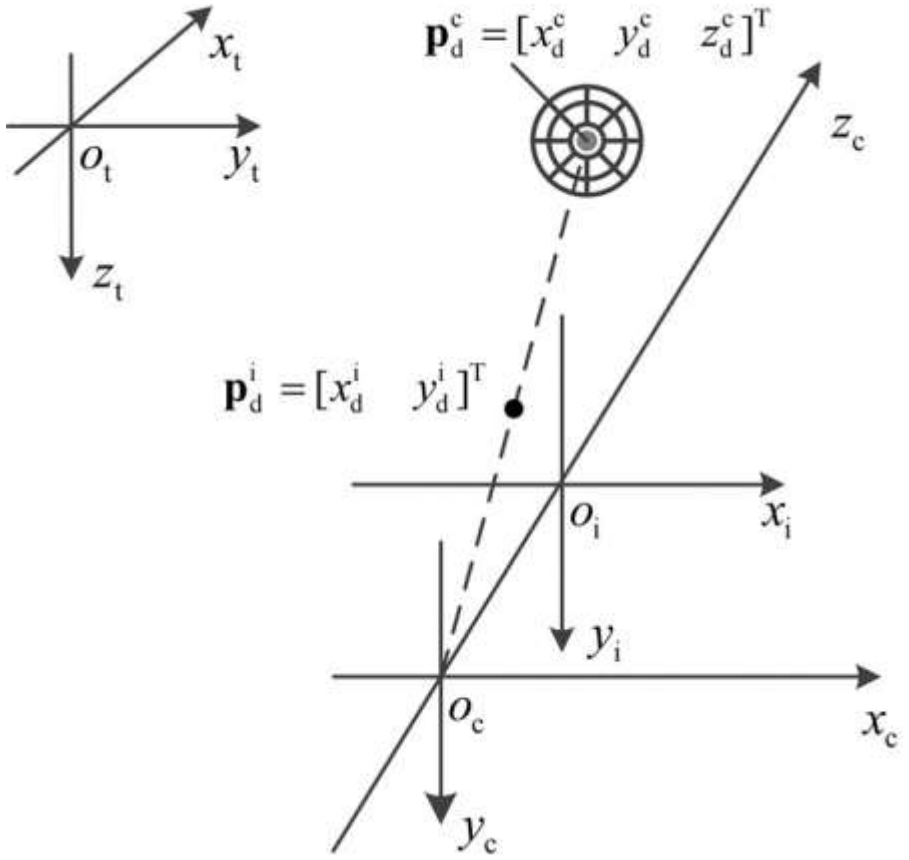


图. 坐标之间的关系

实验原理

对接控制 - 模型建立

其中：

- $\mathbf{v}_{\mathrm{d}}^{\mathrm{c}}=[v_{\mathrm{d},x}^{\mathrm{c}}v_{\mathrm{d},y}^{\mathrm{c}}v_{\mathrm{d},z}^{\mathrm{c}}]^{\mathrm{T}}=\mathbf{R}_{\mathrm{t}}^{\mathrm{re}}(\mathbf{v}_{\mathrm{c}}^{\mathrm{t}}-\mathbf{v}_{\mathrm{d}}^{\mathrm{t}})=\mathbf{v}_{\mathrm{c}}^{\mathrm{re}}-\mathbf{v}_{\mathrm{d}}^{\mathrm{re}}$是相对线速度，
- $\omega_{\mathrm{d}}^{\mathrm{c}}=[\ \omega_{\mathrm{d},x}^{\mathrm{c}}\ \omega_{\mathrm{d},y}^{\mathrm{c}}\ \omega_{\mathrm{d},z}^{\mathrm{c}}]^{\mathrm{T}}$是相对角速度。
- $\mathbf{R}_{\mathrm{t}}^{\mathrm{re}}=\mathbf{R}_{\mathrm{t}}^{\mathrm{c}}$是从加油机坐标系到相对坐标系的转换矩阵
- $\mathbf{v}_{\mathrm{c}}^{\mathrm{re}}=[\ v_{\mathrm{c},x}^{\mathrm{re}}\ v_{\mathrm{c},y}^{\mathrm{re}}\ v_{\mathrm{c},z}^{\mathrm{re}}\ ]^{\mathrm{T}}$和 $\mathbf{v}_{\mathrm{d}}^{\mathrm{re}}$是相对坐标系中相机和锥套的速度。

z_d^c 是相机与锥套中心平面 (安装LED 的位置) 在相机坐标系下沿着 z_c 轴的距离，也叫深度。式 (1.22) 与原理部分的式(1.3)对应，为对接控制的图像伺服模型。

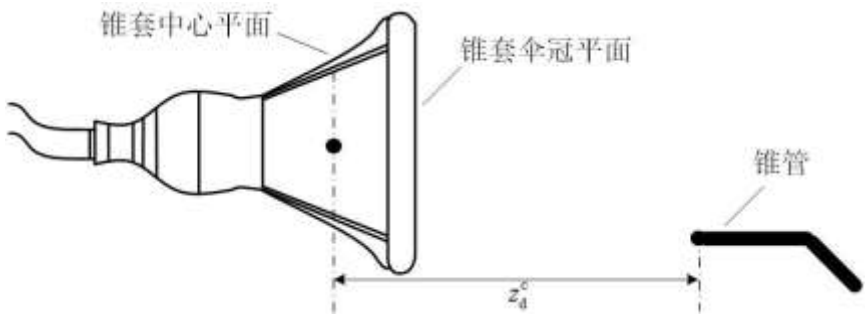


图. 深度

$$\dot{\mathbf{e}}=\underbrace{\begin{bmatrix}-\frac{1}{z_{\mathrm{d}}^{\mathrm{c}}} & 0 & \frac{x_{\mathrm{d}}^{\mathrm{i}}}{z_{\mathrm{d}}^{\mathrm{c}}} & x_{\mathrm{d}}^{\mathrm{i}}y_{\mathrm{d}}^{\mathrm{i}} & -(1+x_{\mathrm{d}}^{\mathrm{i}\,2}) & y_{\mathrm{d}}^{\mathrm{i}} \\ 0 & -\frac{1}{z_{\mathrm{d}}^{\mathrm{c}}} & \frac{y_{\mathrm{d}}^{\mathrm{i}}}{z_{\mathrm{d}}^{\mathrm{c}}} & 1+y_{\mathrm{d}}^{\mathrm{i}\,2} & -x_{\mathrm{d}}^{\mathrm{i}}y_{\mathrm{d}}^{\mathrm{i}} & -x_{\mathrm{d}}^{\mathrm{i}}\end{bmatrix}}_{\mathbf{L}_{\mathrm{e}}}\begin{bmatrix}v_{\mathrm{d},x}^{\mathrm{c}}\\ v_{\mathrm{d},y}^{\mathrm{c}}\\ v_{\mathrm{d},z}^{\mathrm{c}}\\ \omega_{\mathrm{d},x}^{\mathrm{c}}\\ \omega_{\mathrm{d},y}^{\mathrm{c}}\\ \omega_{\mathrm{d},z}^{\mathrm{c}}\end{bmatrix}. \tag{1.22}$$

$$\dot{\mathbf{e}}=\mathbf{L}_{\mathrm{e}}\overline{\mathbf{v}}_{\mathrm{c}}. \tag{1.3}$$

实验原理

对接控制 - 模型建立

为了简化计算，通过忽略一些不重要的变量，并将式 (1.22) 解耦为横侧向通道和纵向通道，得到如下公式。

（1）在 $x_c - z_c$ 平面，三个自由度为 $v_{d,x}^c, v_{d,z}^c, \omega_{d,y}^c$，而 $v_{d,y}^c = 0, \omega_{d,x}^c = 0$，由此得到

$$\dot{e}_x = -\frac{v_{\mathrm{d},x}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} + \frac{e_x v_{\mathrm{d},z}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} - (1+e_x^2)\omega_{\mathrm{d},y}^{\mathrm{c}}$$

（2）在 $y_c - z_c$ 平面，三个自由度为 $v_{d,\boldsymbol{y}}^c, v_{d,z}^c, \omega_{d,\boldsymbol{x}}^c$，而 $v_{d,x}^c = 0, \omega_{d,y}^c = 0$，由此得到

$$\dot{e}_y = -\frac{v_{\mathrm{d},y}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} + \frac{e_y v_{\mathrm{d},z}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} - (1+e_y^2)\omega_{\mathrm{d},x}^{\mathrm{c}}$$

实验原理

对接控制 - 模型建立

将受油机看作是一个刚体，在加油机坐标系下，相机的线速度等于受油机质心的线速度与相机绕质心旋转的速度的矢量和:

$$\mathbf{v}_{\mathrm{c}}^{\mathrm{t}} = \omega_{\mathrm{r}}^{\mathrm{t}} \times \mathbf{p}_{\mathrm{c}}^{\mathrm{r}} + \mathbf{v}_{\mathrm{r}}^{\mathrm{t}}$$

- $\mathbf{v}_{\mathrm{c}}^{\mathrm{t}} = [v_{\mathrm{c},x}^{\mathrm{t}}\ v_{\mathrm{c},y}^{\mathrm{t}}\ v_{\mathrm{c},z}^{\mathrm{t}}]^{\mathrm{T}}$ 是相机的速度，
- $\mathbf{v}_{\mathrm{r}}^{\mathrm{t}} = [v_{\mathrm{r},x}^{\mathrm{t}}\ v_{\mathrm{r},y}^{\mathrm{t}}\ v_{\mathrm{r},z}^{\mathrm{t}}]^{\mathrm{T}}$ 是受油机的线速度，
- $\omega_{\mathrm{r}}^{\mathrm{t}} = [\omega_{\mathrm{r},x}^{\mathrm{t}}\ \omega_{\mathrm{r},y}^{\mathrm{t}}\ \omega_{\mathrm{r},z}^{\mathrm{t}}]^{\mathrm{T}}$ 是受油机的角速度，
- $\mathbf{p}_{\mathrm{c}}^{\mathrm{r}} = [\ x_{\mathrm{c}}^{\mathrm{r}}\ y_{\mathrm{c}}^{\mathrm{r}}\ z_{\mathrm{c}}^{\mathrm{r}}\ ]^{\mathrm{T}}$ 是相机在受油机坐标系下的坐标。

从受油机坐标系到加油机坐标系的转换矩阵为 $\mathbf{R}_{\mathrm{r}}^{\mathrm{t}} = \mathbf{I}_3$

因此，可得

$$\begin{aligned}\mathbf{v}_{\mathrm{c}}^{\mathrm{t}} &= \omega_{\mathrm{r}}^{\mathrm{t}} \times \mathbf{p}_{\mathrm{c}}^{\mathrm{r}} + \mathbf{v}_{\mathrm{r}}^{\mathrm{t}} \\ &= \mathbf{R}_{\mathrm{r}}^{\mathrm{t}} \omega_{\mathrm{r}}^{\mathrm{r}} \times \mathbf{p}_{\mathrm{c}}^{\mathrm{r}} + \mathbf{R}_{\mathrm{r}}^{\mathrm{t}} \mathbf{v}_{\mathrm{r}}^{\mathrm{r}} \\ &= \omega_{\mathrm{r}}^{\mathrm{r}} \times \mathbf{p}_{\mathrm{c}}^{\mathrm{r}} + \mathbf{v}_{\mathrm{r}}^{\mathrm{r}}\end{aligned}$$

$\omega_{\mathrm{r}}^{\mathrm{r}} = [\ \Delta\omega_{x_{\mathrm{b}}}\ \Delta\omega_{y_{\mathrm{b}}}\ \Delta\omega_{z_{\mathrm{b}}}\ ]^{\mathrm{T}}$

是 ω_r^t 在受油机坐标系的各坐标轴上的角速度分量

$\mathbf{v}_{\mathrm{r}}^{\mathrm{r}} = [\ \Delta u\ \Delta v\ \Delta w\ ]^{\mathrm{T}}$

是 v_r^t 在受油机坐标系的各坐标轴上的速度分量

实验原理

对接控制 - 模型建立

相机的线速度可以表示为

$$\begin{aligned}\mathbf{v}_{\mathrm{c}}^{\mathrm{re}} &= \mathbf{R}_{\mathrm{t}}^{\mathrm{re}}\mathbf{v}_{\mathrm{c}}^{\mathrm{t}} \\ &= \mathbf{R}_{\mathrm{t}}^{\mathrm{re}}(\omega_{\mathrm{r}}^{\mathrm{r}} \times \mathbf{p}_{\mathrm{c}}^{\mathrm{r}} + \mathbf{v}_{\mathrm{r}}^{\mathrm{r}})\end{aligned}$$

由于

$$\begin{cases}\Delta u = \Delta V_{\mathrm{r}}^{\mathrm{g}} \cos\Delta\alpha\cos\Delta\beta \\ \Delta v = \Delta V_{\mathrm{r}}^{\mathrm{g}} \sin\Delta\beta \\ \Delta w = \Delta V_{\mathrm{r}}^{\mathrm{g}} \sin\Delta\alpha\cos\Delta\beta\end{cases}$$

则上式可化为

$$\begin{cases}v_{\mathrm{c},x}^{\mathrm{re}} = \Delta V_{\mathrm{r}}^{\mathrm{g}} \sin\Delta\beta + x_{\mathrm{c}}^{\mathrm{r}}\Delta\omega_{z_{\mathrm{b}}} - z_{\mathrm{c}}^{\mathrm{r}}\Delta\omega_{x_{\mathrm{b}}} \\ v_{\mathrm{c},y}^{\mathrm{re}} = \Delta V_{\mathrm{r}}^{\mathrm{g}} \sin\Delta\alpha\cos\Delta\beta + y_{\mathrm{c}}^{\mathrm{r}}\Delta\omega_{x_{\mathrm{b}}} - x_{\mathrm{c}}^{\mathrm{r}}\Delta\omega_{y_{\mathrm{b}}} \\ v_{\mathrm{c},z}^{\mathrm{re}} = \Delta V_{\mathrm{r}}^{\mathrm{g}} \cos\Delta\alpha\cos\Delta\beta + z_{\mathrm{c}}^{\mathrm{r}}\Delta\omega_{y_{\mathrm{b}}} - y_{\mathrm{c}}^{\mathrm{r}}\Delta\omega_{z_{\mathrm{b}}}\end{cases}$$

忽略高阶项，化为

$$\begin{cases}v_{\mathrm{c},x}^{\mathrm{re}} = x_{\mathrm{c}}^{\mathrm{r}}\Delta\omega_{z_{\mathrm{b}}} - z_{\mathrm{c}}^{\mathrm{r}}\Delta\omega_{x_{\mathrm{b}}} \\ v_{\mathrm{c},y}^{\mathrm{re}} = -x_{\mathrm{c}}^{\mathrm{r}}\Delta\omega_{y_{\mathrm{b}}} \\ v_{\mathrm{c},z}^{\mathrm{re}} = \Delta V_{\mathrm{r}}^{\mathrm{g}} + z_{\mathrm{c}}^{\mathrm{r}}\Delta\omega_{y_{\mathrm{b}}}\end{cases}$$

实验原理

对接控制 - 模型建立

结合方程

$$\begin{cases} \Delta\dot{\mathbf{x}}_{\text{rlon}} = \mathbf{A}_{\text{rlon}}\Delta\mathbf{x}_{\text{rlon}} + \mathbf{B}_{\text{rlon}}\Delta\mathbf{u}_{\text{rlon}} \\ \Delta\dot{\mathbf{x}}_{\text{rlat}} = \mathbf{A}_{\text{rlat}}\Delta\mathbf{x}_{\text{rlat}} + \mathbf{B}_{\text{rlat}}\Delta\mathbf{u}_{\text{rlat}} \end{cases}$$

可以得到：

（1）纵向通道受控模型为

$$\begin{cases} \Delta\dot{\mathbf{x}}_{\text{rlon}} = \mathbf{A}_{\text{rlon}}\Delta\mathbf{x}_{\text{rlon}} + \mathbf{B}_{\text{rlon}}\Delta\mathbf{u}_{\text{rlon}} \\ \mathbf{v}_{\text{rlon}}^{\text{re}} \triangleq \begin{bmatrix} v_{\text{c},y}^{\text{re}} \\ v_{\text{c},z}^{\text{re}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -x_{\text{c}}^{\text{r}} \\ 0 & 0 & 0 & 1 & 0 & z_{\text{c}}^{\text{r}} \end{bmatrix}}_{\mathbf{C}_{\text{rlon}}} \Delta\mathbf{x}_{\text{rlon}} \end{cases}$$

（2）横侧向通道受控模型为

$$\begin{cases} \Delta\dot{\mathbf{x}}_{\text{rlat}} = \mathbf{A}_{\text{rlat}}\Delta\mathbf{x}_{\text{rlat}} + \mathbf{B}_{\text{rlat}}\Delta\mathbf{u}_{\text{rlat}} \\ v_{\text{rlat}}^{\text{re}} \triangleq v_{\text{c},x}^{\text{re}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & -z_{\text{c}}^{\text{r}} & x_{\text{c}}^{\text{r}} \end{bmatrix}}_{\mathbf{C}_{\text{rlat}}} \Delta\mathbf{x}_{\text{rlat}} \end{cases}$$

控制目标：对图像伺服模型和，以及受控模型和，设计合适的输入信号 Δu，使得随着时间 t 趋于无穷，图像误差 e 收敛至 0 的同时深度收敛至期望值 0。

实验原理

对接控制 - 控制器设计

1. 外环控制

外环控制器设计是设计期望的相对速度 v_d^c，使图像误差 e 收敛至 0 的同时深度 z_d^c 收敛至期望值 0。具体包括**横侧向通道控制器设计**和**纵向通道控制器设计**。

(1) 横侧向通道控制器设计

考虑到对接过程中 $\omega_{d,y}^c$ 的变化很小，可以忽略不计，则 $v_{d,x}^c$ 的期望速度设计为

$$v_{d,xdes}^c = k_1 e_x$$

如果 $v_{d,x} = v_{d,xdes}^c$，则有式子

$$\dot{e}_x = -\lambda_1 e_x$$

式中 $\lambda_1 = \frac{k_1 - v_{d,z}^c}{z_d^c}$，$k_1$为可调节的参数，满足 $k_1 > max(v_{d,y}^c)$，这时有 $\lim_{t\to\infty}|e_x(t)| = 0$。

实验原理

对接控制 - 控制器设计

(2) 纵向通道控制器设计

考虑到对接过程中 $\omega_{d,x}^c$ 的变化很小，可以忽略不计，则 $v_{d,y}^c$ 的期望速度设计为

$$v_{d,ydes}^c = k_2 e_y$$

如果 $v_{d,y} = v_{d,ydes}^c$，则有式子

$$\dot{e}_y = -\lambda_2 e_y$$

式中 $\lambda_2 = \frac{k_2 - v_{d,z}^c}{z_d^c}$，$k_2$ 为可调节的参数，满足 $k_2 > max(v_{d,z}^c)$，这时有 $\lim_{t\to\infty}|e_y(t)| = 0$。

深度方向相对速度 $v_{d,z}^c$ 的期望速度设计为

$$v_{d,zdes}^c = -k_3 z_d^c$$

此时，如果 $v_{d,z}^c = v_{d,zdes}^c$ 且 $k_3 > 0$，则有 $\lim_{t\to\infty}|z_d^c(t)| = 0$。

除此之外，为了防止受油机的速度过快而错过锥套，在上式中添加项“$-k_4|e_x| - k_5|e_y|$”来调节受油机的速度

$$v_{d,zdes}^c = -k_3 z_d^c - k_4|e_x| - k_5|e_y|$$

如果 $|e_x|$ 和 $|e_y|$ 较大，则该项可以减慢速度以避免超调。

实验原理

对接控制 - 控制器设计

综上，在外环控制器中，v_d^c的期望速度为

$$\mathbf{v}_{\mathrm{d,des}}^{\mathrm{c}} = \begin{bmatrix} v_{\mathrm{d},x\mathrm{des}}^{\mathrm{c}} & v_{\mathrm{d},y\mathrm{des}}^{\mathrm{c}} & v_{\mathrm{d},z\mathrm{des}}^{\mathrm{c}} \end{bmatrix}^{\mathrm{T}}$$

实际上，在外环控制中，由于$v_d^c = v_c^{re} - v_d^{re}$,而只有$v_c^{re}$可以控制，因此将$v_d^{re}$视为扰动，从而得到$v_{c,des}^{re} = v_{d,des}^{re}$，将其解耦表示，得到

$$\mathbf{v}_{\mathrm{c,des}}^{\mathrm{re}} = \begin{bmatrix} \mathbf{v}_{\mathrm{rlon,des}}^{\mathrm{re}} \\ v_{\mathrm{rlat,des}}^{\mathrm{re}} \end{bmatrix}$$

式中 $\mathbf{v}_{\mathrm{rlon,des}}^{\mathrm{re}} = [v_{\mathrm{d},y\mathrm{des}}^{\mathrm{c}} \; v_{\mathrm{d},z\mathrm{des}}^{\mathrm{c}}]^{\mathrm{T}}$, $v_{\mathrm{rlat,des}}^{\mathrm{re}} = v_{\mathrm{d},x\mathrm{des}}^{\mathrm{c}}$。

实验原理

对接控制 - 控制器设计

2. 内环控制

内环控制器直接输出发动机油门操控量和舵面操控量对受油机模型进行控制。内环控制器的设计是设计合适的期望输入信号 Δu，使 v_c^{re} 跟踪 $v_{c,des}^{re}$，并抑制复杂的干扰。具体包括横侧向通道控制器设计和纵向通道控制器设计。内环控制器采用最优控制方法，即线性二次型调节器（LQR）。

LQR 小知识：

对于线性定常系统：

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}. \tag{1.43}$$

式中 $\mathbf{x} \in \mathbb{R}^n$ 为 n 维状态，$\mathbf{u} \in \mathbb{R}^m$ 为 m 维输入，系统可控或至少是可镇定的。定义性能指标为：

$$J = \frac{1}{2}\int_0^\infty (\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \mathbf{u}^{\mathrm{T}}\mathbf{R}\mathbf{u}). \tag{1.44}$$

式中 $\mathbf{u}$ 不受约束，$\mathbf{Q}, \mathbf{R}$ 一般为常数对称正定阵（一些情况下可以放宽为对称半正定阵）。使 J 取极小值的最优控制可以表示为：

$$\mathbf{u} = -\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}. \tag{1.45}$$

式中 $\mathbf{P} \in \mathbb{R}^{n\times n}$ 为 $n \times n$ 维正定对称阵，满足以下的里卡蒂矩阵代数方程（Riccati equation）：

$$-\mathbf{P}\mathbf{A} - \mathbf{A}^{\mathrm{T}}\mathbf{P} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P} - \mathbf{Q} = 0. \tag{1.46}$$

该方程可以使用 MATLAB 中的 “lqr” 函数方便地解得结果。

实验原理

对接控制 - 控制器设计

(1) 纵向通道控制器设计

将跟踪误差定义为 $\mathbf{e}_{\text{rlon}} = \mathbf{v}_{\text{rlon}}^{\text{re}} - \mathbf{v}_{\text{rlon,des}}^{\text{re}}$ 这里使用积分器来抑制复杂干扰，其定义为

$$\mathbf{q}_{\text{rlon}} = \int_0^t \mathbf{e}_{\text{rlon}} = \int_0^t (\mathbf{v}_{\text{rlon}}^{\text{re}} - \mathbf{v}_{\text{rlon,des}}^{\text{re}})$$

则有式子

$$\begin{bmatrix} \Delta\dot{\mathbf{x}}_{\text{rlon}} \\ \dot{\mathbf{q}}_{\text{rlon}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\text{rlon}} & \mathbf{0}_{6\times2} \\ \mathbf{C}_{\text{rlon}} & \mathbf{0}_{2\times2} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_{\text{rlon}} \\ \mathbf{q}_{\text{rlon}} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{\text{rlon}} \\ \mathbf{0}_{2\times2} \end{bmatrix} \Delta\mathbf{u}_{\text{rlon}} - \begin{bmatrix} \mathbf{0}_{6\times2} \\ \mathbf{I}_{2\times2} \end{bmatrix} \mathbf{v}_{\text{rlon,des}}^{\text{re}}$$

控制器设计为

$$\Delta\mathbf{u}_{\text{rlon}} = -\mathbf{K}_{x1}\Delta\mathbf{x}_{\text{rlon}} - \mathbf{K}_{e1}\mathbf{q}_{\text{rlon}}$$

式中 $\mathrm{K}_{\mathrm{x1}} \in \mathbb{R}^{2\times6}$，$\mathrm{K}_{\mathrm{e1}} \in \mathbb{R}^{2\times2}$。为了计算 K_{xl} 和 K_{el}，设计如下的代价函数

$$J(\Delta\mathbf{u}_{\text{rlon}}) = \underset{\mathbf{K}_{x1},\mathbf{K}_{e1}}{\operatorname{argmin}} \int_0^{\infty} \left(\begin{bmatrix} \Delta\mathbf{x}_{\text{rlon}}^{\mathrm{T}} & \mathbf{q}_{\text{rlon}}^{\mathrm{T}} \end{bmatrix} \mathbf{Q}_{\text{rlon}} \begin{bmatrix} \Delta\mathbf{x}_{\text{rlon}} \\ \mathbf{q}_{\text{rlon}} \end{bmatrix} + \Delta\mathbf{u}_{\text{rlon}}^{\mathrm{T}} \mathbf{R}_{\text{rlon}} \Delta\mathbf{u}_{\text{rlon}} \right)$$

式中 $\mathrm{Q}_{\mathrm{rlon}} \geq 0$ 和 $\mathrm{R}_{\mathrm{rlon}} \geq 0$ 为加权矩阵。

实验原理

对接控制 - 控制器设计

(2) 横侧向通道控制器设计

相似的，将跟踪误差定义为 $e_{\mathrm{rlat}}=v_{\mathrm{rlat}}^{\mathrm{re}}-v_{\mathrm{rlat,des}}^{\mathrm{re}}$ 这里使用积分器来抑制复杂干扰，其定义为

$$q_{\mathrm{rlat}}=\int_0^t e_{\mathrm{rlat}}=\int_0^t\left(v_{\mathrm{rlat}}^{\mathrm{re}}-v_{\mathrm{rlat,des}}^{\mathrm{re}}\right)$$

则有式子

$$\begin{bmatrix}\Delta\dot{\mathbf{x}}_{\mathrm{rlat}}\\ \dot{q}_{\mathrm{rlat}}\end{bmatrix}=\begin{bmatrix}\mathbf{A}_{\mathrm{rlat}} & \mathbf{0}_{6\times2}\\ \mathbf{C}_{\mathrm{rlat}} & \mathbf{0}_{2\times2}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}_{\mathrm{rlat}}\\ q_{\mathrm{rlat}}\end{bmatrix}+\begin{bmatrix}\mathbf{B}_{\mathrm{rlat}}\\ \mathbf{0}_{2\times2}\end{bmatrix}\Delta\mathbf{u}_{\mathrm{rlat}}-\begin{bmatrix}\mathbf{0}_{6\times2}\\ \mathbf{I}_{2\times2}\end{bmatrix}v_{\mathrm{rlat,des}}^{\mathrm{re}}$$

控制器设计为

$$\Delta\mathbf{u}_{\mathrm{rlat}}=-\mathbf{K}_{x2}\Delta\mathbf{x}_{\mathrm{rlat}}-\mathbf{K}_{e2}q_{\mathrm{rlat}}$$

式中 $\mathrm{K_{x2}}\in\mathbb{R}^{2\times6}$，$\mathrm{K_{e2}}\in\mathbb{R}^{2\times2}$。为了计算 $\mathrm{K_{x2}}$ 和 $\mathrm{K_{e2}}$，设计如下的代价函数

$$J(\Delta\mathbf{u}_{\mathrm{rlat}})=\underset{\mathbf{K}_{x2},\mathbf{K}_{e2}}{\operatorname{argmin}}\int_0^{\infty}\left(\begin{bmatrix}\Delta\mathbf{x}_{\mathrm{rlat}}^{\mathrm{T}} & q_{\mathrm{rlat}}^{\mathrm{T}}\end{bmatrix}\mathbf{Q}_{\mathrm{rlat}}\begin{bmatrix}\Delta\mathbf{x}_{\mathrm{rlat}}\\ q_{\mathrm{rlat}}\end{bmatrix}+\Delta\mathbf{u}_{\mathrm{rlat}}^{\mathrm{T}}\mathbf{R}_{\mathrm{rlat}}\Delta\mathbf{u}_{\mathrm{rlat}}\right)$$

式中 $\mathrm{Q_{rlat}}\geq0$ 和 $\mathrm{R_{rlat}}\geq0$ 为加权矩阵。

实验原理

对接控制 - 控制器设计

3. 控制器综合

综上，图像伺服控制器可以综合为

$$\begin{cases} v^{c}_{d,xdes} = k_1 e_x \\ v^{c}_{d,ydes} = k_2 e_y \\ v^{c}_{d,zdes} = -k_3 z^{c}_{d} - k_4 |e_x| - k_5 |e_y| \\ \Delta \mathbf{u}_{rlon} = -\mathbf{K}_{x1} \Delta \mathbf{x}_{rlon} - \mathbf{K}_{e1} \mathbf{q}_{rlon} \\ \Delta \mathbf{u}_{rlat} = -\mathbf{K}_{x2} \Delta \mathbf{x}_{rlat} - \mathbf{K}_{e2} q_{rlat} \end{cases}$$

式中$k_i > 0$，$i = 1,2,\ldots,5$为控制增益。

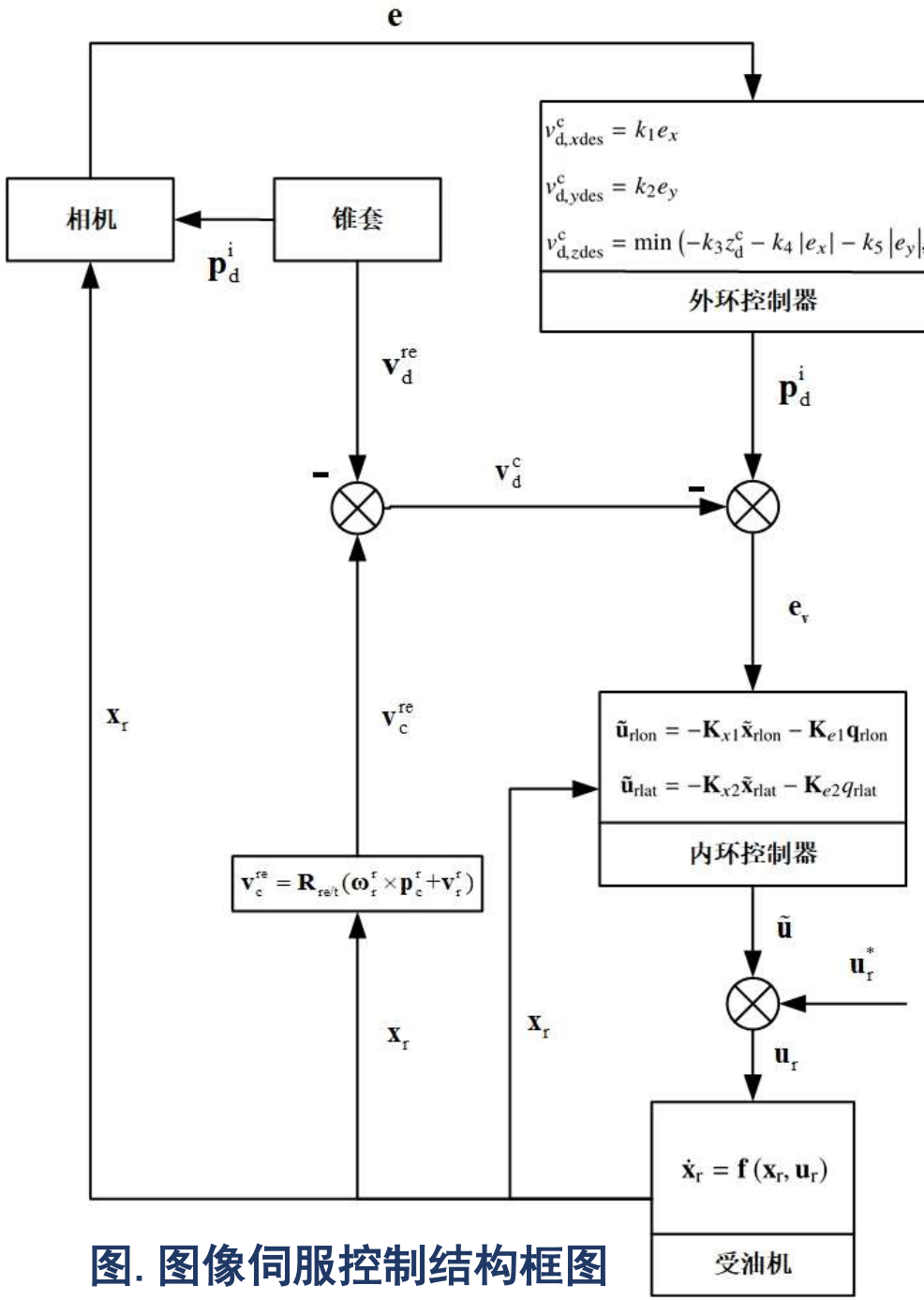


图. 图像伺服控制结构框图

实验原理

相关发展

无人机利用搭载的摄像头进行环境感知，并基于图像信息进行精确的运动控制，这使得无人机在进行复杂任务时获得更丰富的环境信息和更高的控制精度。基于图像的视觉伺服主要依赖摄像机作为视觉反馈信息源，这减少了对全球定位系统（GPS）和其他外部导航辅助设备的依赖，从而增强了环境适应性和系统的可靠性。

- **多旋翼无人机：**视觉伺服技术的应用广泛，涵盖了精确着陆、物体跟踪、空中目标拦截以及自主避障等关键任务。
- **固定翼无人机：**基于图像的伺服控制同样在自主打击和空中对接等领域发挥重要作用。

无论是无人机回收对接还是空中加油对接，由于各种大气扰动，对接控制都非常困难，因此近年来利用图像伺服进行自动对接控制成为了研究的热点，这是因为采用基于图像的伺服控制进行空中加油对接具有若干显著优点，特别是在自动或半自动操作环境中，图像伺服控制可以实现高精度的视觉跟踪，这对于完成精细的对接操作至关重要。

大纲

1. 实验原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 硬件在环仿真实验
6. 本章小结

基础实验

实验目标

■ 已知

(1) 软件：MATLAB R2022b 或以上版本，Python，AARSim 仿真平台。

(2) 程序：实验指导包“e7/e7-1”。实验指导包中主要有“Simulink”和“Python”两个文件夹，其中“Simulink”文件夹中主要有：基于图像伺服的空中加油对接仿真程序“AAR_IBVS.slx”，初始化文件“F16_Init.m”以及其他平台中调用的函数等。“Python”文件夹中主要有：锥套识别脚本“detect-realtime.py”以及其调用的其他函数等。

■ 目标

了解掌握基于图像伺服的空中加油对接仿真平台的原理，熟悉各模块功能，编写图像伺服控制代码，运行仿真完成空中加油对接实验，分析实验结果。

基础实验

实验步骤

步骤一：了解图像伺服的空中加油对接仿真平台的构成

①加油机“Tanker System”模块

设定加油机加速度为零，成等速平飞状态，其状态通过积分器来实现，

输出：“Tanker States”，表示加油机的状态。

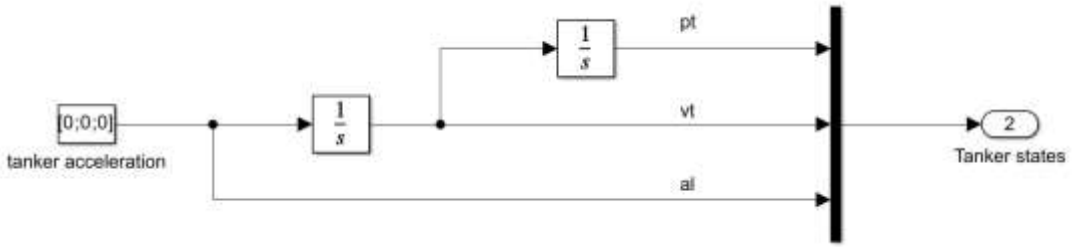


图. 加油机状态的 Simulink 实现，来源于“e7/e7-1/Simulink/AAR_IBVS.slx”

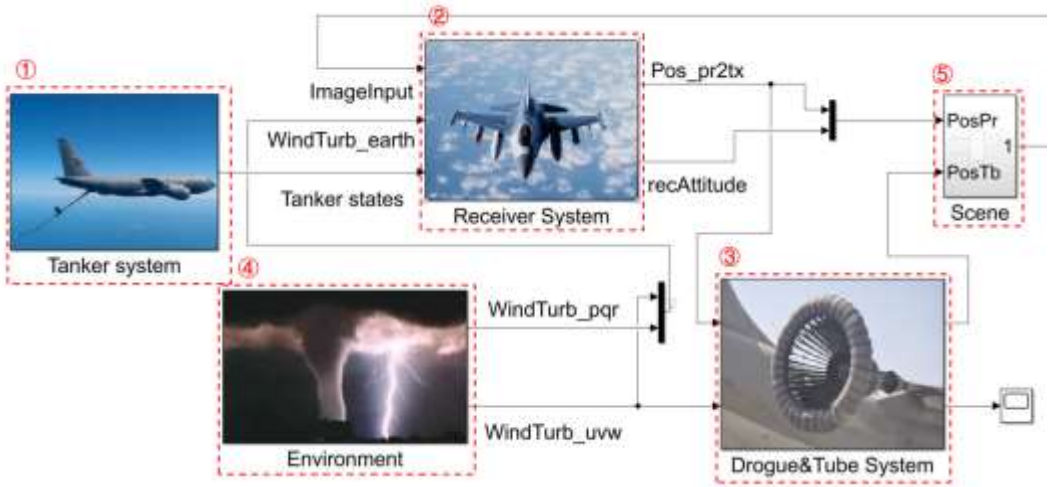


图. 空中加油对接仿真平台，来源于“e7/e7-1/Simulink/AAR_IBVS.slx”

实验步骤

步骤一：了解图像伺服的空中加油对接仿真平台的构成

②受油机“Receiver System”模块

- **输入：**
  - “ImageInput”:图像信息输入，包括被估计出的锥套位置、图像误差和深度
  - “WindTurb_earth”，表示地面坐标系下的风扰动
  - “Tanker States”，表示加油机的状态
- **输出：**
  - “Pos_pr2tx”，表示锥管相对于加油机的位置
  - “recAttitude”表示受油机的姿态
- **三个部分：**
  - 受油机模型
  - 控制器
  - 坐标转换模块

③锥套锥管模块“Drogue&Tube System”

- **输入：**
  - 锥管相对于加油机的位置“Pos_pr2tx”
  - 风扰动的线速度“WindTurb_uvw”
- **输出：**
  - 各节软管和锥套的位姿“PosTb”
- **目的：**
  - 计算对接装置锥套的位姿

④环境模块“Environment”

- **输出：**
  - 风扰动的线速度“WindTurb_uvw”
  - 风扰动的角速度“WindTurb_pqr”
- **目的：**
  - 根据空速和高度计算风速等大气数据

基础实验

实验步骤

步骤一：了解图像伺服的空中加油对接仿真平台的构成

⑤视景模块 “Scene”

- **目的：**
  - 完成对接场景的视景显示
- **输入：**
  - 各节软管和锥套的位姿 “PosTb”
  - 受油机的位姿 “PosPr”
- **实现原理：**
  - 通过 UDP 广播与 RflySim 以及 Python 终端进行通讯
  - 将各节软管、锥套、受油机和加油机的位姿信息发送给 RflySim3D 进行显示
  - 接收来自 Python 返回的视觉处理结果

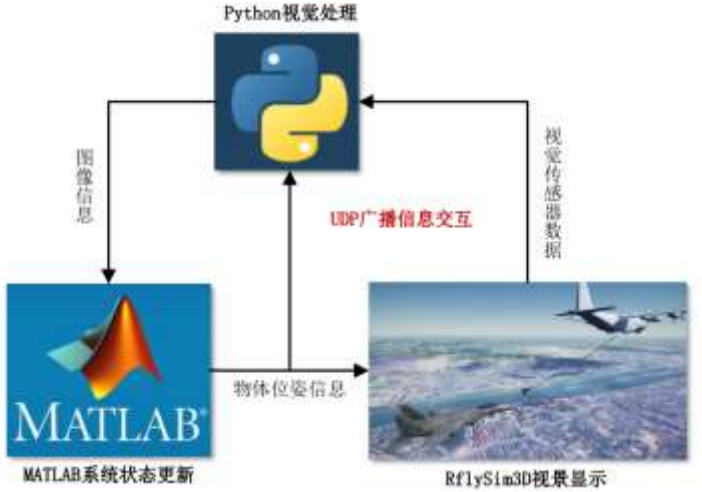


图. 信息交互结构

图. 对接视景效果

实验步骤

步骤二：实现图像伺服控制算法

在“AAR_IBVS\ReceiverSystem\Controller\IBVS Controller”路径下预留出了外环控制器的 Matlab Function 模块。

- 输入：图像误差、深度估计结果以及受油机的状态
- 输出：相对坐标系下相机的速度期望

```
ey = -ey; % reverse ey
Z_d = 1.0; % compensation Z
kz = 0.5; % set control parameters
ky = 3;
kx = 1;
k1 = 3; k2 = 1;
u_vcz = kz*(Z-Z_d)-k1*(abs(ex))-k2*(abs(ey));
u_vcy = ky*(ey-0);
u_vcx = kx*(ex-0);
```

将 y 轴方向的图像误差进行反向

设置深度误差的补偿值，用于控制插入锥套的深度

设置控制参数

计算期望的相机相对于锥套的速度

图像伺服控制器，来源于“e7/e7-1/Simulink/AAR_IBVS.slx”

实验步骤

步骤二：实现图像伺服控制算法

内环控制器为 LQR 控制，以纵向通道为例：

```
A_lon=A([1,3,5,7,8,11,13,14],[1,3,5,7,8,11,13,14]);
B_lon=B([1,3,5,7,8,11,13,14],1:2);
C_lon_xd_h=[1 0 zeros(1,6);
           0 1 zeros(1,6)];
% augment the system
AA_lon=[A_lon,zeros(8,2);-C_lon_xd_h,zeros(2,2)];
BB_lon=[B_lon;zeros(2,2)];
% set lqr parameters
Q_lon=diag([4 10 10 1 10 10 0 0 1 3]) %
R_lon=diag([50 50]);
% calculate K of lqr
[k_lon,p_lon,e_lon]=lqr(AA_lon,BB_lon,Q_lon,R_lon);
Kp_lon=k_lon(:,1:8);
Ki_lon=-k_lon(:,9:10);
```

设置式中的 A_{rlon}, B_{rlon}, C_{rlon}

进行系统增广，将跟踪误差的积分引入到系统中

设定 LQR 的 Q 矩阵和 R 矩阵

使用 MATLAB 中的 “lqr” 函数求解 LQR 问题

将增广系统的反馈增益矩阵分为式中的 K_{x2}, K_e

图像伺服控制器，来源于 “e7/e7-1/Simulink/AAR_IBVS.slx”

实验步骤

步骤二：实现图像伺服控制算法

计算得到“Ki”和“Kp”，在 Simunlink 中搭建实现。

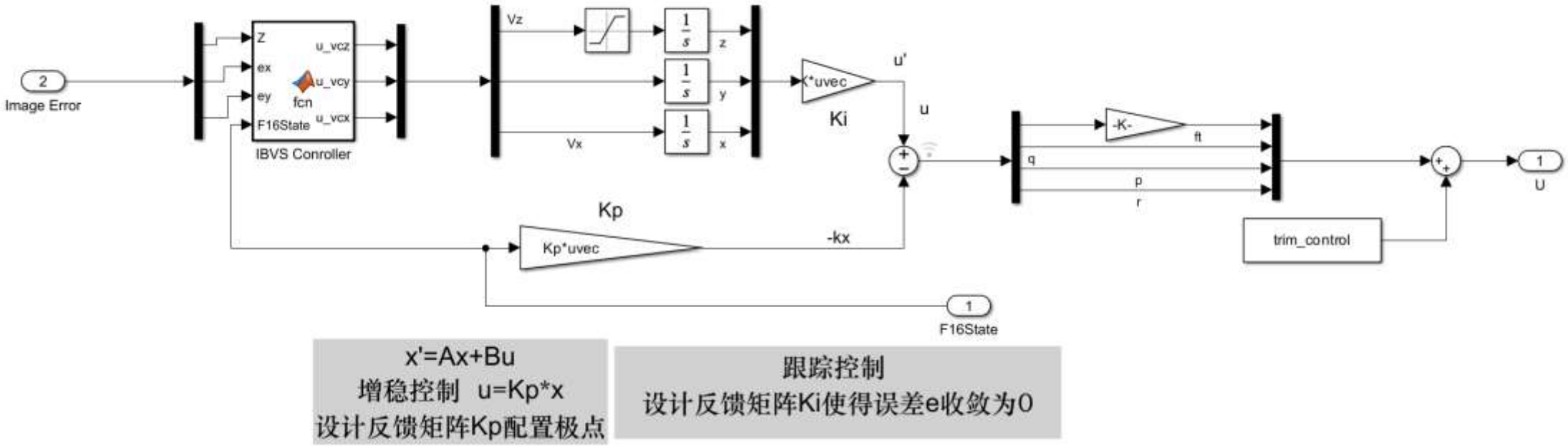


图. 图像伺服控制器实现，来源于“e7/e7-1/Simulink/AAR_IBVS.slx”

基础实验

实验步骤

步骤三：运行仿真，分析空中加油对接实验结果

运行 Simulink 仿真:

- 打开 RflySim3D;
- 启动 Simulink 文件并在 5s 内暂停;
- 运行 “detectrealtime.py”;
- 等待识别图窗弹出后，继续运行 Simulink文件（如果 Python 报文件路径错误，则需要把平台复制到 C 盘下再运行）;
- 等待仿真运行完成，运行“AnalysisVisualError.mlx”，计算得到对接过程中的对接误差并绘图，可以得到对接实验结果。

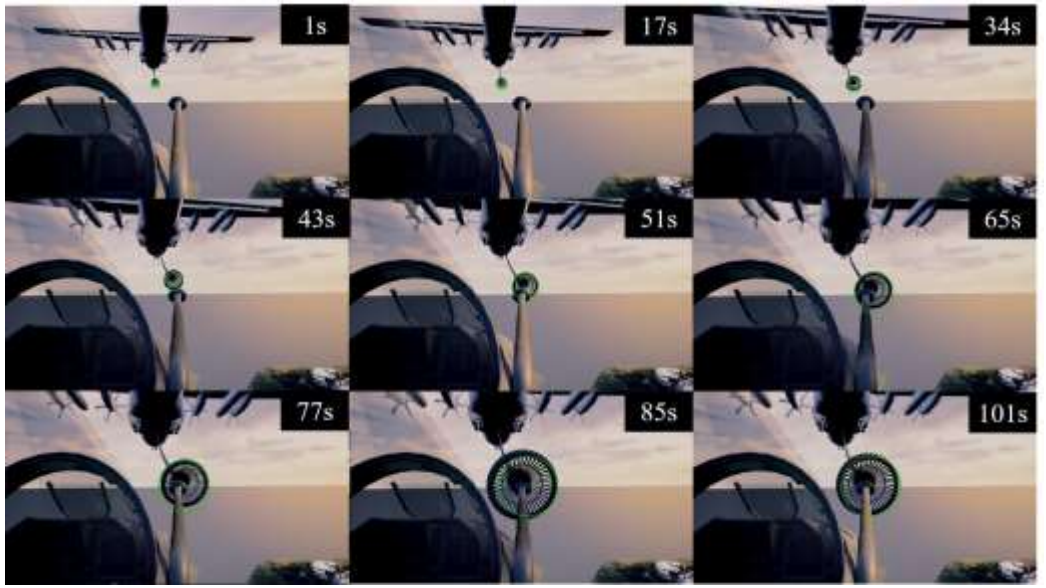


图. 对接过程视景显示

从对接视景中观察，图像伺服算法完成了对接任务。

基础实验

实验步骤

步骤三：运行仿真，分析空中加油对接实验结果

“图像误差”：对接过程中图像误差曲线，表示在图像坐标系中，锥套中心点与图像坐标系原点的差。

- 图像误差在 100 秒内收敛至零。

“深度误差”：对接过程中深度误差曲线，锥管顶点与锥套平面的距离。

- 深度误差越小，深度误差衰减速度越小，符合设计的相应通道的外环控制器式。

“对接位置误差”：对接过程中相对坐标系下三轴对接误差，即锥管末端与锥套中心之间的位置差。

- 横侧向方向和纵向方向：锥管末端以较快的速度靠近锥套中心。
- 深度方向：锥管末端靠近锥套中心的速度随着深度误差的衰减也逐渐减小。三轴对接误差均收敛至零，对接成功。

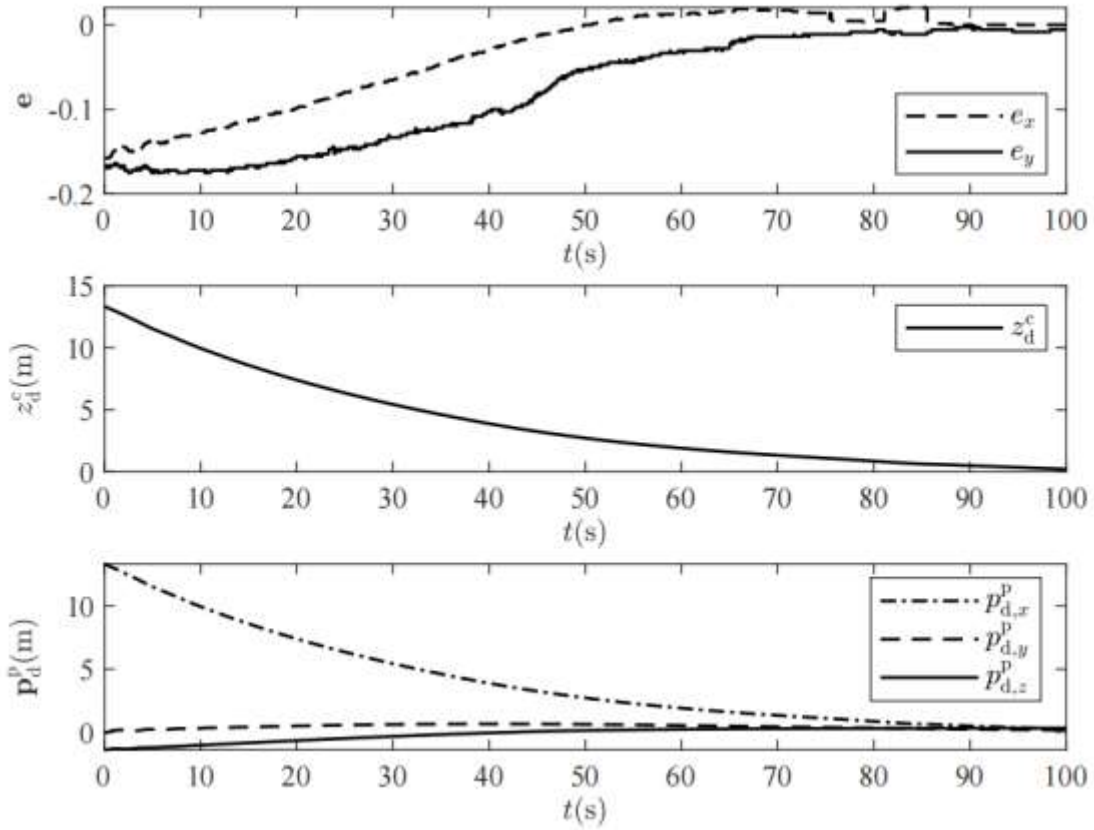
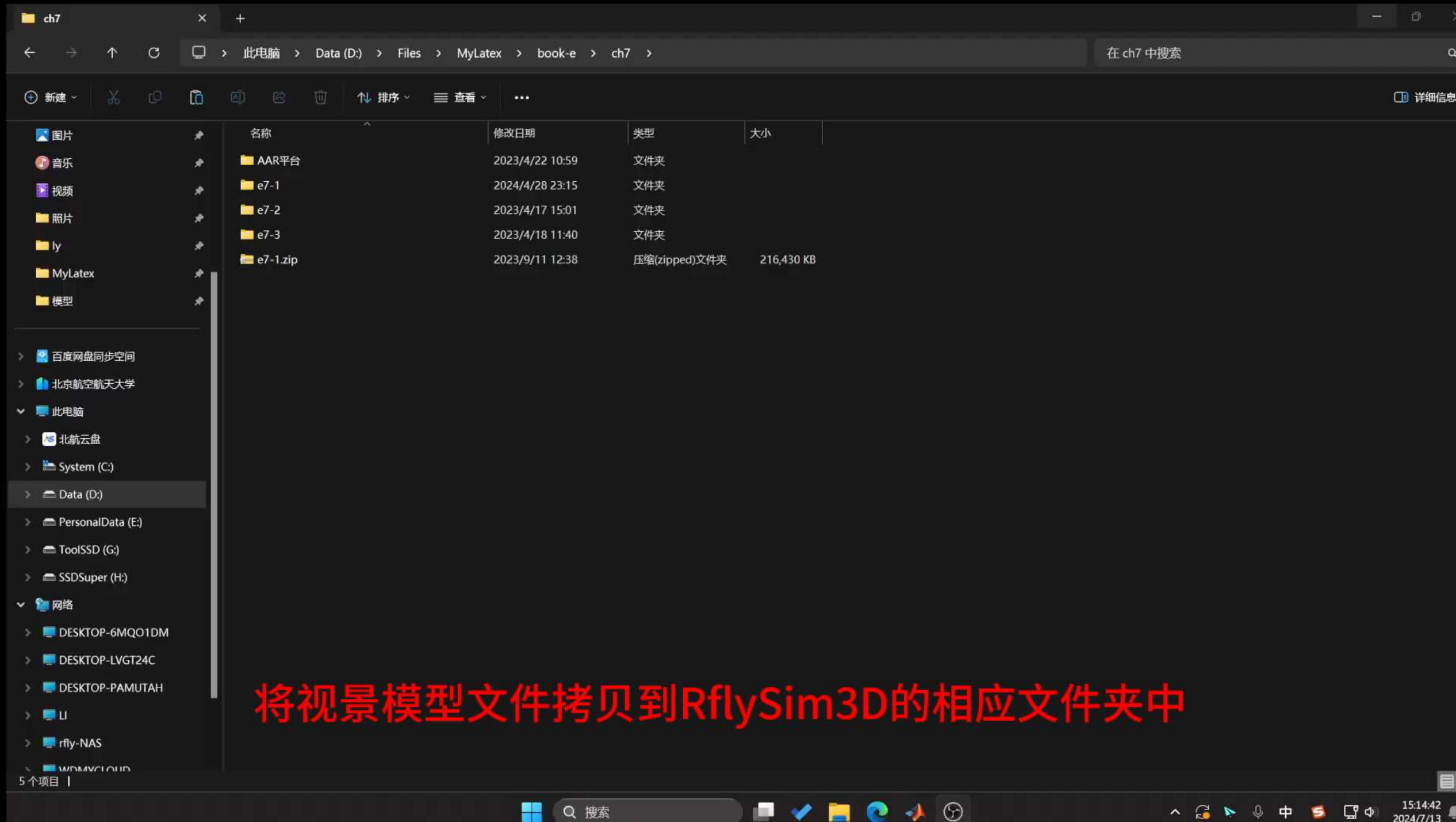


图. 对接误差


ch7
此电脑 › Data (D:) › Files › MyLatex › book-e › ch7 ›
在 ch7 中搜索
新建
排序
查看
详细信息
图片
音乐
视频
照片
ly
MyLatex
模型
百度网盘同步空间
北京航空航天大学
此电脑
北航云盘
System (C:)
Data (D:)
PersonalData (E:)
ToolSSD (G:)
SSDSuper (H:)
网络
DESKTOP-6MQO1DM
DESKTOP-LVGT24C
DESKTOP-PAMUTAH
LI
rfly-NAS
名称
修改日期
类型
大小
AAR平台 2023/4/22 10:59 文件夹
e7-1 2024/4/28 23:15 文件夹
e7-2 2023/4/17 15:01 文件夹
e7-3 2023/4/18 11:40 文件夹
e7-1.zip 2023/9/11 12:38 压缩(zipped)文件夹 216,430 KB
将视景模型文件拷贝到RflySim3D的相应文件夹中
5 个项目
搜索
15:14:42

大纲

1. 基本原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 硬件在环仿真实验
6. 本章小结

分析实验

实验目标

■ 已知

(1) 软件：MATLAB R2022b 及以上版本，Python，AARSim 仿真平台。。

(2) 程序：实验指导包“e7-2”。实验指导包中主要有基于图像伺服的对接控制程序“IBVS”文件夹和基于位置伺服的对接控制程序“PBVS”文件夹。以“IBVS”文件夹为例，其中主要有“Simulink”和“Python”两个文件夹。“Simulink”文件夹中主要有：基于图像伺服的空中加油对接仿真程序“AAR_IBVS.slx”，初始化文件“F16_Init.m” 以及其他平台中调用的函数等。“Python” 文件夹中主要包含锥套识别脚本“detect-realtime.py”以及其调用的其他函数等。注意“IBVS”文件夹和“PBVS”文件夹中的“detect-realtime.py”并不完全一样，区别在于前者返回的是锥套中心在图像坐标系中的像素位置和深度，后者返回的是估计出的锥套的三维位置。

■ 目标

对比分析不同强度紊流对基于图像伺服的空中加油对接算法的影响，以及在不同紊流下不同的控制参数。并与基于位置伺服的空中加油对接算法进行对比，分析二者在相机安装误差上的抗扰能力的区别。

分析实验

实验步骤

步骤一：对比不同紊流强度下对接效果

紊流强度是高度和超越强度概率的函数，在高度一定的前提下，紊流强度与超越强度概率是负相关的。

在 Simulink 的 Aerospace 工具箱中有名“Dryden Wind Turbulence Model”的相应模块，在“Probability of ex_x0002_ceedance of high-altitude intensity”选项中可设置超越强度概率。一般情况下：

- 超越强度概率为 10−1 对应“I级紊流”；
- 超越强度概率为 10−2 对应“II 级紊流”；
- 超越强度概率为 10−3 对应“III 级紊流”。

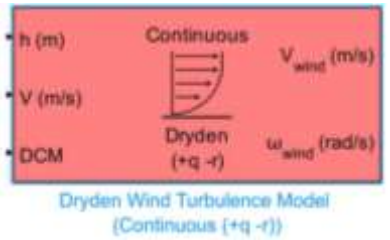


图.“Dryden Wind Turbulence Model” 模块

修改超越概率强度“Probability of exceedance of high-altitude intensity”即可修改紊流强度

图. 紊流强度设置

分析实验

实验步骤

步骤一：对比不同紊流强度下对接效果

将紊流强度分别设置为 0、I、II 并运行仿真，且不改变控制参数，观察仿真结果。

（1）随着风扰动强度的增加，虽然在对接过程中，受油机的运动轨迹出现了一些波动，但是图像误差 e、深度误差 $z_{\mathrm{d}}^{\mathrm{c}}$和对接误差 $p_{\mathrm{d}}^{\mathrm{p}}$ 依旧在 100s 内基本收敛为 0。

这说明：

- 在 I 级紊流下对接仍然是成功的，并且对接误差曲线依然比较平滑
- 控制器对于风扰动具有一定的抑制功能，在较小的风扰动下具有良好的对接鲁棒性

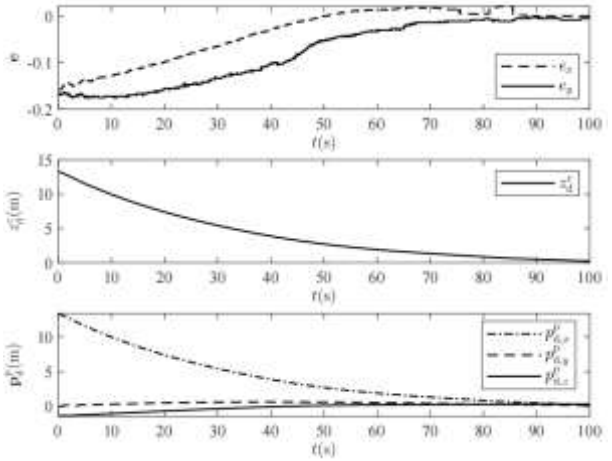


图. 无紊流对接误差

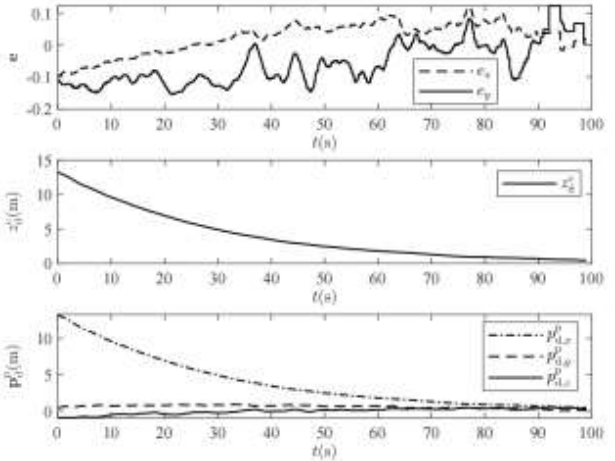


图. I 级紊流对接误差

分析实验

实验步骤

步骤一：对比不同紊流强度下对接效果

（2）可以看出：

- 在 II 级紊流下，**图像误差曲线波**动更加剧烈，特别是 40s 以后。
  - 当相机距离锥套越近时，锥套在加油机坐标系的移动单位距离反映在图像中的移动距离越大。
- **深度误差曲线**也有了较为明显的波动。
- **对接误差曲线**经过 110s 也未能同时收敛至 0 附近：$p_{\mathrm{d},x}^{\mathrm{p}}$ 方向上的最终误差大约为 1m，$p_{\mathrm{d},y}^{\mathrm{p}}$ 和 $p_{\mathrm{d},z}^{\mathrm{p}}$ 大约为 0.5m。
  - 锥管未能插入锥套，因此对接是失败的。
  - 与较强的风扰动作用有关。实际上在当前的无人机空中加油对接中，对于大气情况也有规定，当风扰动超过一定限度之后，对接任务就因安全原因不允许被执行了。

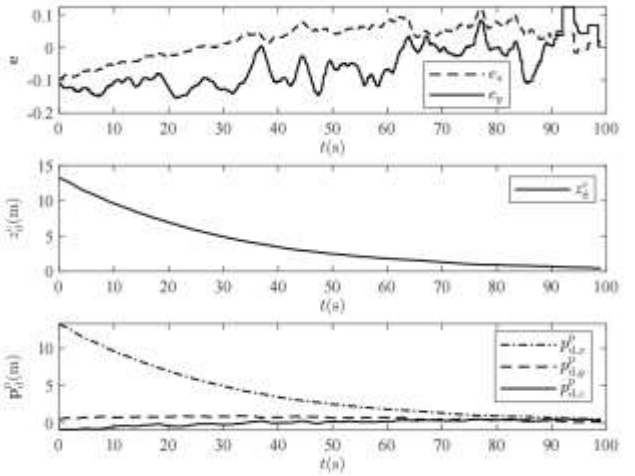


图. I 级紊流对接误差

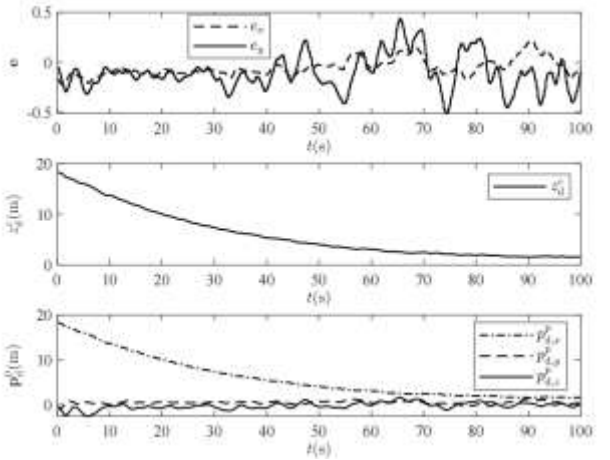


图. II 级紊流对接误差

分析实验

实验步骤

步骤二：对比图像伺服与位置伺服对接效果

基于图像伺服的空中加油对接平台“AAR_IBVS.slx”与基于位置伺服的空中加油对接平台“AAR_PBVS.slx”的区别在于受油机的控制器部分。

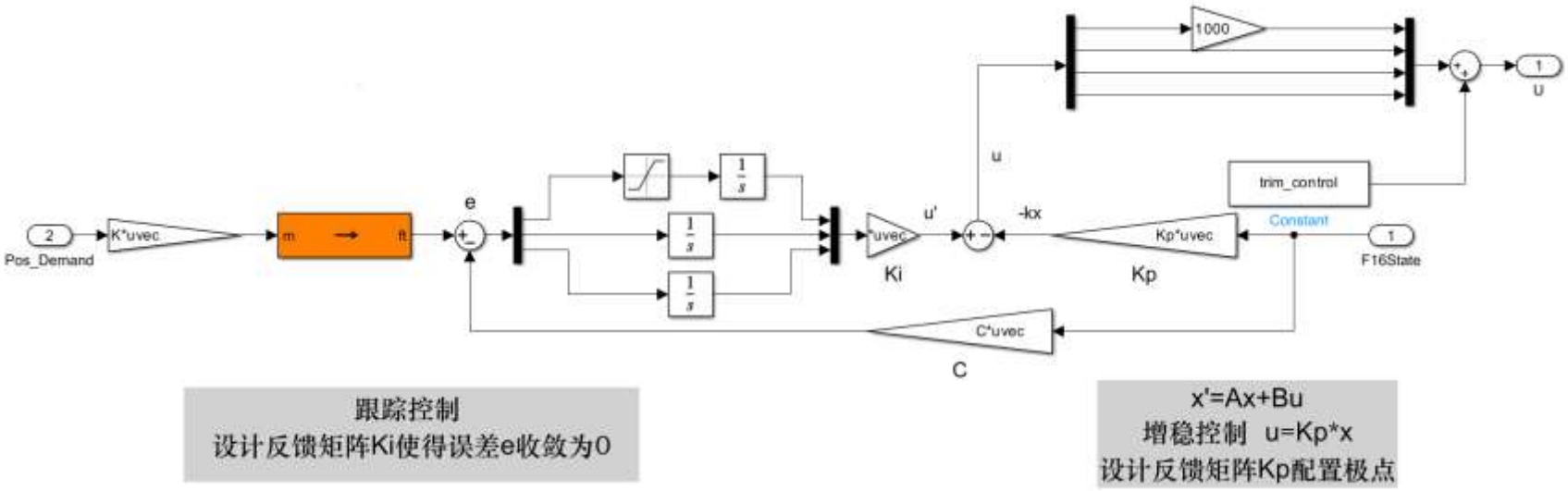


图. 位置伺服控制器，来源于“e7/e7-2/PBVS/Simulink/AAR_PBVS.slx”

分析实验

实验步骤

步骤二：对比图像伺服与位置伺服对接效果

假设标定出现问题等，导致了相机安装位置出现了一定的偏差 $[1 \quad 0 \quad -0.5]^T$。

- 在 AARSim 平台的PBVS 仿真初始化文件“F16_Init.m”中进行修改：

```
%Without camera location error
% d_cam_rc = [4.7;0.54;-1.45];

%With camera location error
d_cam_rc = [4.7+1;0.54;-1.45-0.5];
```

修改之前即没有相机安装位置测量误差的情况

加入了相机安装位置误差的情况

PBVS 平台中注入相机安装位置测量误差，来源于“e7/e72/PBVS/Simulink/F16_Init.m”

分析实验

实验步骤

步骤二：对比图像伺服与位置伺服对接效果

- 在 AARSim 平台的 IBVS 仿真初始化文件“F16_Init.m”中进行修改：

```
%Without camera location error
% Lx = 4.7;
% Ly = 0.54;
% Lz = -1.45;

%With camera location error
Lx = 4.7+1;
Ly = 0.54;
Lz = -1.45-0.5;
```

修改之前即没有相机安装位置测量误差的情况

加入了相机安装位置误差的情况

IBVS 平台中注入相机安装位置测量误差，来源于“e7/e7-2/IBVS/Simulink/F16_Init.m”

分析实验

实验步骤

步骤二：对比图像伺服与位置伺服对接效果

- 分别运行：
  - 基于**图像伺服**的空中加油对接平台“AAR_IBVS.slx”和相应的“detect-realtime.py”仿真
  - 基于**位置伺服**的空中加油对接平台“AAR_PBVS.slx”和相应的“detect-realtime.py”仿真
- 运行完仿真后运行绘图脚本“AnalysisError.mlx”，即可得到对接误差曲线。

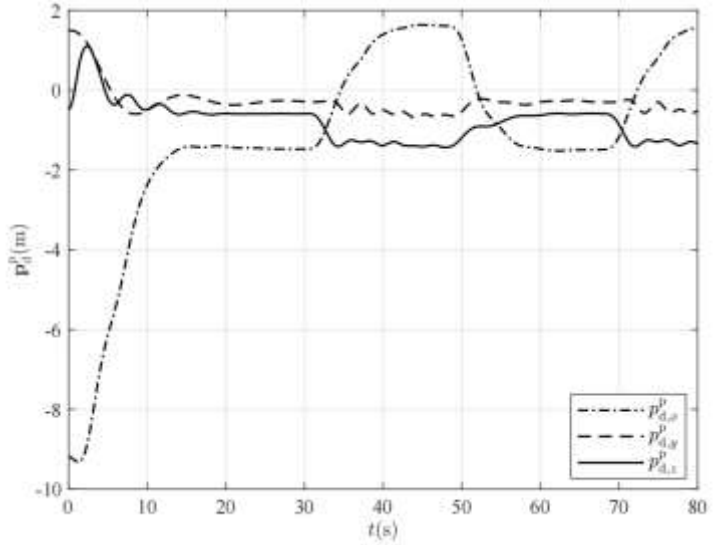


图. 有相机位置测量误差时位置伺服对接误差

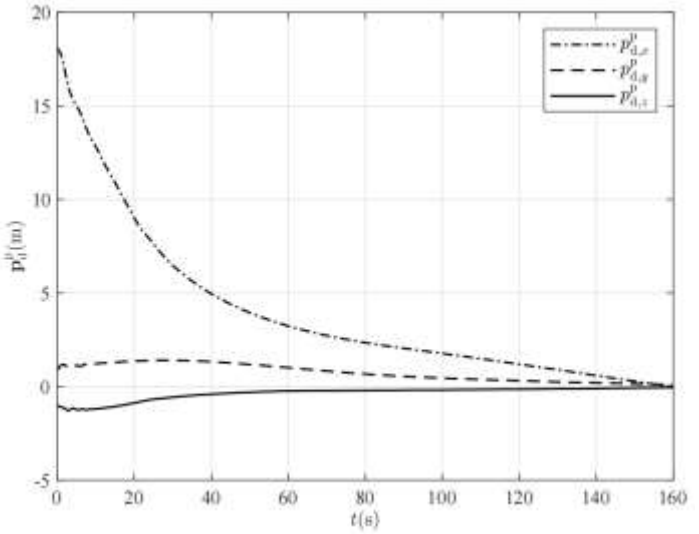


图. 有相机位置测量误差时图像伺服对接误差

分析实验

实验步骤

步骤二：对比图像伺服与位置伺服对接效果

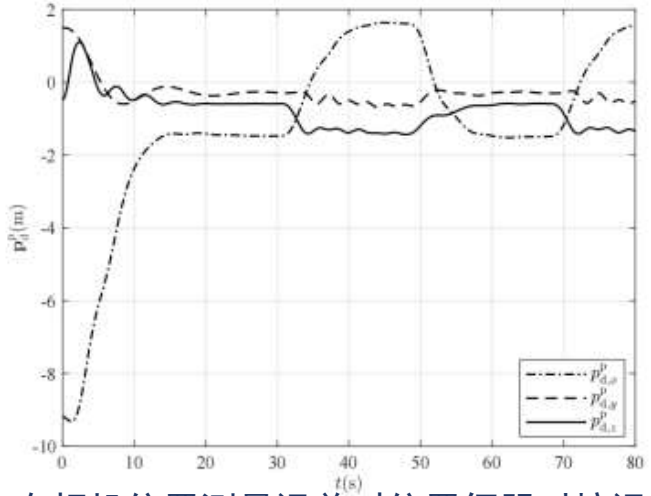


图. 有相机位置测量误差时位置伺服对接误差

- 0-15s 为靠近阶段，锥套与锥管的距离大概在 2m 左右；
- 15-30s 为位置保持并估计锥套位置的过程；
- 30-50s 为第一次尝试对接，显然由于相机位置测量误差，对接误差并未收敛至 0 并且相差很远，特别是 $p_{d,x}^p$ 方向；
- 50-70s 为对接失败后退回等待位置重新估计锥套位置；
- 70-80s 为第二次尝试，但是依旧失败。

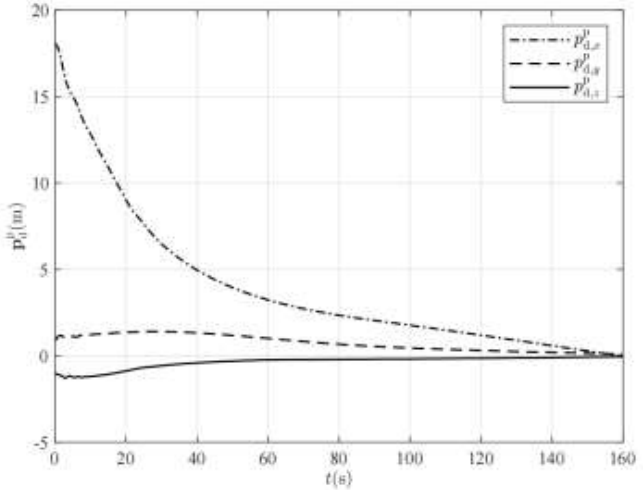


图. 有相机位置测量误差时图像伺服对接误差

对接误差在 160s 内平稳地收敛至 0，虽然用时较长，但是对接是成功的

在有相机位置测量误差时，**图像伺服**控制器可以成功对接但是**位置伺服**不能成功对接。

分析实验

实验步骤

步骤二：对比图像伺服与位置伺服对接效果

（1）图像伺服控制器：

- 相机测量位置只会用于解耦后的纵向图像伺服模型以及横侧向图像伺服模型，而这二者又只会应用纵向和横侧向通道的内环状态方程增广，最终纵向与横侧向的增广状态方程则会用于内环增稳控制器的设计。
  - 相机位置的测量结果只会对于图像伺服的内环控制器有影响。
- 相机的三维模型重构误差、位置测量误差、相机校准误差以及相机的安装位置误差等**误差**会导致图像误差和深度误差。
  - **图像误差**对以上的各种误差是不敏感的且可以被外环控制器消除
  - **深度误差**对以上各种误差是敏感的，但是深度只决定对接时的前向速度，对于对接成功与否没有横侧向误差影响那么大。

（2）位置伺服控制器：

- 内环控制器需要跟踪的是外环控制器利用位置伺服获取的加油机坐标系下锥套位置，内环控制器只需要跟踪上外环给定的锥套位置，那么就会成功对接。
  - 因此这里对于位置伺服，相机测量位置直接作用外环控制器中对锥套的精确定位。
- 相机的三维模型重构误差、位置测量误差、相机校准误差以及相机的安装位置误差等误差会导致三维的位置估计误差。
  - 位置估计误差对以上各种误差都很敏感，位置估计误差直接决定了对接的成功与否。

MATLAB R2022b

HOME PLOTS APPS

New Script | New Live Script | New | Open | Find Files | Compare | Import Data | Clean Data | Variable | Save Workspace | Clear Workspace | Favorites | Analyze Code | Run and Time | Clear Commands | Simulink | Layout | Preferences | Set Path | Parallel | Add-Ons | Help | Community | Request Support | Learn MATLAB

FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES

D: ▸ Files ▸ MyLatex ▸ book-e ▸ ch7 ▸

Current Folder

Name

AAR平台
e7-1
e7-2
e7-3
e7-1.zip

Details

Select a file to view details

Command Window

```
  Figure (2) with properties:

      Number: 2
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [573 438 560 420]
       Units: 'pixels'

  Show all properties


h3 =

  Figure (3) with properties:

      Number: 3
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [573 438 560 420]
       Units: 'pixels'

  Show all properties


h4 =

  Figure (4) with properties:

      Number: 4
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [573 438 560 420]
       Units: 'pixels'

  Show all properties

fx >>
```

第一部分：对比不同扰动下的对接情况

Workspace

Name	Value
A	18x18 double
A_la	8x8 double
A_lon	8x8 double
AA_la	9x9 double
AA_lon	10x10 double
airSpeed	393.7200
alpha0	6.6239
alt	[0,0,0]
altitude	9843
B	18x4 double
B_la	8x2 double
B_lon	8x2 double
b_ref	30
BB_la	9x2 double
BB_lon	10x2 double
beta	0
bowSinks	10227x6 double
BowWave	1x1 struct
C	3x18 double
C2m	3x18 double
C_la_yd	[1,0,0,0,-1.4500,4.7...
C_lon_xd_h	2x8 double
d_pr_rc	[8.2900,0.4400,-1.7...
d_ref	11.3200
data	1x1 double timeser...
dLEF	9.7444
DroguePos	1x1 struct
e_la	[-0.3255 + 2.2908i;...
e_lon	[-0.3902 + 0.5513i;...
Euler_0	[0,0.1156,0]
ex_image	16001x1 double
ey_image	16001x1 double
fi_flag	1
figurename1	'ImageError'
figurename2	'DockingError'
figurename3	'DepthError'
figurename4	'Error-1-2-4'
filename	'Analysis\Data_202...
h1	1x1 Figure
h2	1x1 Figure
h3	1x1 Figure
h4	1x1 Figure

搜索

15:31:24
2024/7/13

大纲

1. 基本原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 硬件在环仿真实验
6. 本章小结

分析实验

实验目标

■ 已知

(1) 软件：MATLAB R2022b 及以上版本，Python，AARSim 仿真平台。。

(2) 程序：实验指导包“e7/e7-3”。实验指导包中主要有速度模式下的基于图像伺服的空中加油对接仿真程序“AAR_IBVS_Vmode.slx”，初始化文件“F16_In_new.m”以及其他平台中调用的函数等。Python 锥套识别脚本使用基础实验中的即可。

■ 目标

1. 设计以“受油机相对地面的速度”作为顶层控制接口的 IBVS 控制器，并在 AARSim 仿真平台中完成软件仿真对接控制验证。

2. 进一步，使用 CubePilot/Pixhawk 自驾仪，以“受油机的航点”作为控制接口，完成硬件在环仿真对接控制验证。

设计实验

实验设计

步骤一：建立速度控制模式下的制导模型

速度控制模式的底层飞行控制可以用线性二次型控制方法（LQR）、模型预测控制和反馈线性化方法（MPC+FL）或微分平坦度方法来实现。具有速度控制模式的接收机的制导模型可以描述如下：

$$\dot{\mathbf{p}}_{\mathrm{r}}^{\mathrm{g}} = \mathbf{v}_{\mathrm{r}}^{\mathrm{g}}$$

$$\begin{bmatrix} v_{\mathrm{r},x}^{\mathrm{g}} \\ v_{\mathrm{r},y}^{\mathrm{g}} \\ v_{\mathrm{r},z}^{\mathrm{g}} \end{bmatrix} = \begin{bmatrix} G_x & & \\ & G_y & \\ & & G_z \end{bmatrix} \begin{bmatrix} v_{\mathrm{r},x\mathrm{ref}}^{\mathrm{g}} \\ v_{\mathrm{r},y\mathrm{ref}}^{\mathrm{g}} \\ v_{\mathrm{r},z\mathrm{ref}}^{\mathrm{g}} \end{bmatrix}.$$

$\mathbf{p}_{\mathrm{r}}^{\mathrm{g}} = \left[p_{\mathrm{r},x}^{\mathrm{g}}\ p_{\mathrm{r},y}^{\mathrm{g}}\ p_{\mathrm{r},z}^{\mathrm{g}}\right]^{\mathrm{T}}$

受油机在地面坐标系下的位置

$\mathbf{v}_{\mathrm{r}}^{\mathrm{g}} = \left[v_{\mathrm{r},x}^{\mathrm{g}}\ v_{\mathrm{r},y}^{\mathrm{g}}\ v_{\mathrm{r},z}^{\mathrm{g}}\right]^{\mathrm{T}}$

受油机在地面坐标系下的速度

从 $v_{r,ref}^{g}$ 到 v_{r}^{g} 的传递函数

$\mathbf{v}_{\mathrm{r,ref}}^{\mathrm{g}} = \left[v_{\mathrm{r},x\mathrm{ref}}^{\mathrm{g}}\ v_{\mathrm{r},y\mathrm{ref}}^{\mathrm{g}}\ v_{\mathrm{r},z\mathrm{ref}}^{\mathrm{g}}\right]^{\mathrm{T}}$

受油机在地面坐标系下的参考速度

实验设计

步骤二：建立速度控制模式下的相机运动模型

为了在速度控制模式下充分考虑姿态和角速度对图像误差的影响，与基础实验和分析实验不同，此处进行较为完整的推导，得到:

$$\dot{e}_x = -\frac{v_{\mathrm{d},x}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} + \frac{e_x v_{\mathrm{d},z}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} + e_x e_y \omega_{\mathrm{d},x}^{\mathrm{c}} - (1+e_x^2)\omega_{\mathrm{d},y}^{\mathrm{c}} + e_y \omega_{\mathrm{d},z}^{\mathrm{c}}$$

$$\dot{e}_y = -\frac{v_{\mathrm{d},y}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} + \frac{e_y v_{\mathrm{d},z}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}}} + (1+e_y^2)\omega_{\mathrm{d},x}^{\mathrm{c}} - e_x e_y \omega_{\mathrm{d},y}^{\mathrm{c}} - e_x \omega_{\mathrm{d},z}^{\mathrm{c}}.$$

实验设计

步骤二：建立速度控制模式下的相机运动模型

推导得到受油机在地面坐标系下的速度与相机相对于加油机的速度之间的关系（考虑受油机速度、姿态和角速度）：

$$\begin{cases} v_{c,x}^{re} = v_{r,y}^{g} + x_c^r \Delta\omega_{z_b} - z_c^r \Delta\omega_{x_b} \\ v_{c,y}^{re} = v_{r,z}^{g} + y_c^r \Delta\omega_{x_b} - x_c^r \Delta\omega_{y_b} \\ v_{c,z}^{re} = v_{r,x}^{g} - V_t^g + z_c^r \Delta\omega_{y_b} - y_c^r \Delta\omega_{z_b} \end{cases}.$$

实验设计

步骤三：图像伺服控制器设计

设计受油机相对地面的期望速度为：

$$\begin{cases} v_{\mathrm{r},x\mathrm{d}}^{\mathrm{g}} = -k_3 z_{\mathrm{d}}^{\mathrm{c}} - k_4 \left|e_x\right| - k_5 \left|e_y\right| + V_{\mathrm{t}}^{\mathrm{g}} - z_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{y_{\mathrm{b}}} + y_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{z_{\mathrm{b}}} \\ v_{\mathrm{r},y\mathrm{d}}^{\mathrm{g}} = k_1 e_x + z_{\mathrm{d}}^{\mathrm{c}} e_x e_y \Delta\omega_{y_{\mathrm{b}}} - z_{\mathrm{d}}^{\mathrm{c}} (1 + e_x^2) \Delta\omega_{z_{\mathrm{b}}} + z_{\mathrm{d}}^{\mathrm{c}} e_y \Delta\omega_{x_{\mathrm{b}}} - x_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{z_{\mathrm{b}}} + z_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{x_{\mathrm{b}}} \\ v_{\mathrm{r},z\mathrm{d}}^{\mathrm{g}} = k_2 e_y - z_{\mathrm{d}}^{\mathrm{c}} e_x e_y \Delta\omega_{z_{\mathrm{b}}} + z_{\mathrm{d}}^{\mathrm{c}} (1 + e_y^2) \Delta\omega_{y_{\mathrm{b}}} - z_{\mathrm{d}}^{\mathrm{c}} e_y \Delta\omega_{x_{\mathrm{b}}} - y_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{x_{\mathrm{b}}} + x_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{y_{\mathrm{b}}} \end{cases}.$$

设计实验

实验步骤

步骤一：实现速度控制模式下的图像伺服控制器

在“AAR_IBVS_Vmode\Receiver System\Controller”路径下用“Matlab Function”实现速度控制模式下的图像伺服控制器。

- 输入：图像误差、深度估计结果以及受油机的状态
- 输出：地面坐标系下受油机的速度期望

```
d_pr_rc = [6.06;0.54;-0.86]; % the distance between the camera and the receiver
ey = -ey; % reverse ey
Z_d = 1.0; % compensation Z
kz = 0.5; % set control parameters
ky = 3;
kx = 1;
k1 = 3;
k2 = 1;
vgz = kz*(Z-Z_d)-k1*(abs(ex))-k2*(abs(ey))-d_pr_rc(3)*q+d_pr_rc(2)*r;
vgy = ky*(ey-0)+Z*ex*ey*q-Z(1+ex^2)r+Z*ey*p-d_pr_rc(1)*r+d_pr_rc(3)*p;
vgx = kx*(ex-0)-Z*ex*ey*r+Z(1+ey^2)q-Z*ex*p-d_pr_rc(2)*p+d_pr_rc(1)*q;
```

→ 将 y_i 轴方向的图像误差进行反向

→ 设置深度误差的补偿值，用于控制插入锥套的深度

→ 设置控制参数

→ 计算期望的相机相对于锥套的距离

速度控制模式下的图像伺服控制器，来源于 “e7/e7-3/AAR_IBVS_Vmode.slx”

设计实验

实验步骤

步骤一：实现速度控制模式下的图像伺服控制器

- 在“AAR_IBVS_Vmode\Receiver System\Controller”路径下用“Matlab Function” 实现速度控制模式下的图像伺服控制器，该模块的输入为图像误差、深度估计结果以及受油机的状态，输出为地面坐标系下受油机的速度期望。
- 底层控制器采用 LQR 方法实现速度控制模式，与基础实验中的 LQR 控制器的区别在于：
- 不需要对系统进行关于视觉跟踪误差的增广，而将受油机在地面坐标系下的速度直接作为控制目标。

设计实验

实验步骤

步骤二：运行仿真，分析空中加油对接实验效果

完成以上的图像伺服算法实现后，即可运行 Simulink 仿真。下图为加油机受油机对接视景显示。

图. 对接画面

设计实验

实验步骤

步骤二：运行仿真，分析空中加油对接实验效果

右图为无扰动情况下对接过程中误差曲线，图像误差表示在图像坐标系中，锥套中心点与图像坐标系原点的差。

- **图像误差：**在 67 秒内收敛至零。
- **深度误差：**深度误差越小，深度误差衰减速度越小。
- **对接误差：**横侧向方向和纵向方向相对距离以较快速度衰减，而深度方向由于控制器设计，深度误差衰减速度与深度误差负相关。
- 最终三轴对接误差均收敛至零，对接成功。

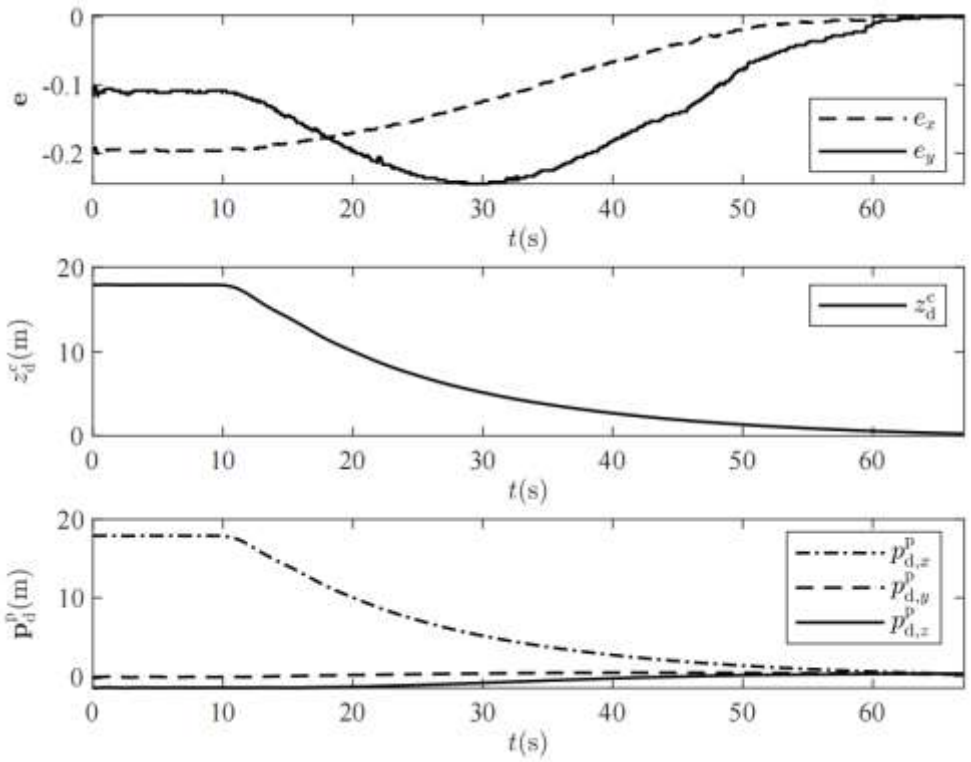


图. 对接画面

设计实验

实验步骤

步骤二：运行仿真，分析空中加油对接实验效果

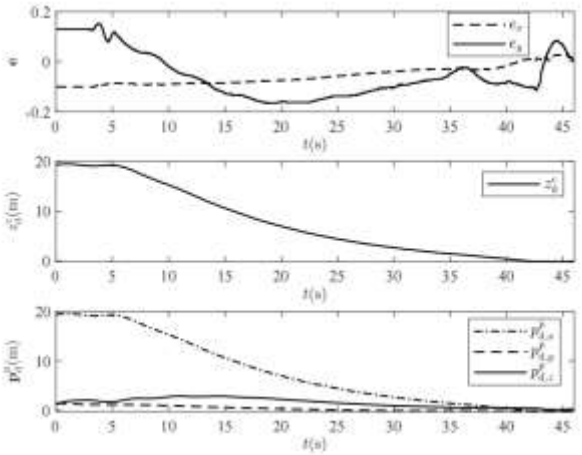


图. I级紊流、头波效应和加油机尾流干扰

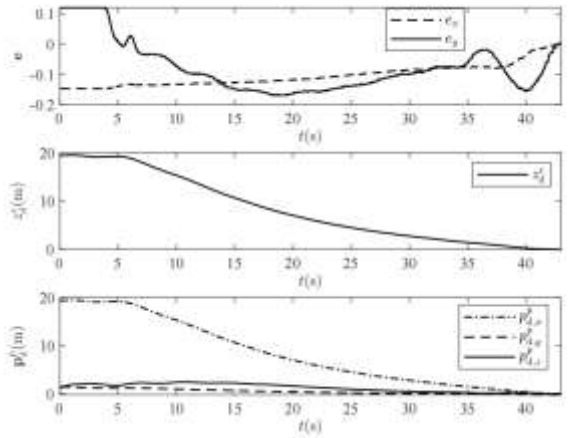


图. I级紊流、头波效应、加油机尾流和离散阵风

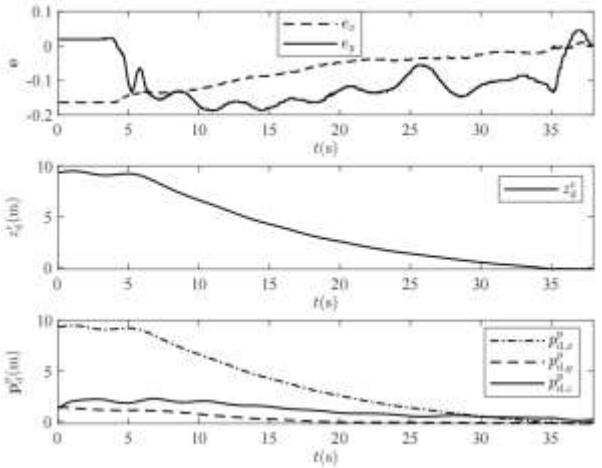


图. II级紊流、头波效应、加油机尾流和离散阵风

离散阵风对于对接结果没有明显的影响。

→ 本节设计的图像伺服控制器可以抵抗离散阵风的干扰。

虽然随着干扰的逐渐增加，对接过程中的图像误差和对接误差曲线的波动逐渐增大，但是最终均可以完成成功的对接。

→本节设计的控制器具有较好的鲁棒性和抗扰能力。

MATLAB R2022b

HOME PLOTS APPS

New Script | New Live Script | New | Open | Find Files | Compare — FILE
Import Data | Clean Data | Variable | Save Workspace | Clear Workspace — VARIABLE
Favorites | Analyze Code | Run and Time | Clear Commands — CODE
Simulink — SIMULINK
Layout | Preferences | Set Path | Parallel | Add-Ons — ENVIRONMENT
Help | Community | Request Support | Learn MATLAB — RESOURCES

Search Documentation

D: ▸ Files ▸ MyLatex ▸ book-e ▸ ch7 ▸

Current Folder

Name
- AAR平台
- e7-1
- e7-2
- e7-3
- e7-1.zip

Command Window

```
    Position: [573 438 560 420]
       Units: 'pixels'

  Show all properties


h3 =

  Figure (3) with properties:

      Number: 3
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [573 438 560 420]
       Units: 'pixels'

  Show all properties


h4 =

  Figure (4) with properties:

      Number: 4
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [573 438 560 420]
       Units: 'pixels'

  Show all properties

>> load('D:\Files\MyLatex\book-e\ch7\e7-3\Analysis\Data_2023-41-18-10-04III_Turbence_0倍_Bowwave.mat')
>> load('Data.mat')
>> AnalysisVisualError_NEW
>> load('D:\Files\MyLatex\book-e\ch7\e7-3\Analysis\Data_2023-41-18-10-04III_Turbence_0倍_Bowwave.mat')
>> AnalysisVisualError
>> AnalysisVisualError
>>
```

打开“e7/e7-3/AAR_IBVS_Vmode.slx”

Workspace

Name	Value
ex_image	7801x1 double
ey_image	7801x1 double
figurename1	'ImageError'
figurename2	'DockingError'
figurename3	'DepthError'
figurename4	'Error'
h1	1x1 Figure
h2	1x1 Figure
h3	1x1 Figure
h4	1x1 Figure
left	1100
PError	8933x3 double
right	8900
t	1x7801 double
VisualSignal	8933x3 double
x_pd	7801x1 double
y_pd	7801x1 double
z_pd	7801x1 double
z_pd_image	7801x1 double

Details

Select a file to view details

搜索

15:59:38
2024/7/13

大纲

1. 基本原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 硬件在环仿真实验
6. 本章小结

硬件在环仿真实验

实验目标

■ 已知

(1) 软件：MATLAB R2022b 或以上版本，Python，AARSim 仿真平台，RflySim 仿真平台。RflySim 仿真平台提供了微小型固定翼无人机硬件在环仿真环境，可以实现使用 Python 终端通过 MAVLINK 向固定翼无人机发送“offboard”模式下的指令，指令支持“空速 + 高度 + 航向”控制模式。

(2) 程序：实验指导包 “e7-4”。实验指导包中主要有：包含空中对接硬件在环仿真程序的“CopterSimulink_HITL” 文件夹，进行母舰和锥套运动仿真的 “Simulink” 文件夹，存放母舰、无人机及锥套 RflySim3D 模型的“module” 文件夹。其中 “CopterSimulink_HITL” 文件夹主要包含无人机控制脚本文件“Mytrajectory.py”、锥套识别脚本文件 “detect-realtime.py”、一键启动 RflySim 批处理文件“PX4SimHITL.bat”、Python 环境配置列表文件 “requirements.txt”、装有 Python 环境安装包的文件夹“packages”、装有飞控固件的文件夹 “Firmware”、保存图像误差的文件 “error_data.mat” 和误差分析绘图文件 “AnalysisVisualError.m” 以及调用的其它函数等。“Simulink” 文件夹主要包含运行母舰和锥套运动仿真文件 “AirRefueling_Platform.slx” 和其它平台中调用的函数等。“module” 文件夹主要包含母舰模型“WestTransportC130J”，无人机模型 “MQ-9Reaper” 和锥套模型“ProbeLightScene”。

（3）硬件：CubePilot/Pixhawk 自驾仪。

■ 目标

在设计实验的基础上，使用 CubePilot/Pixhawk 自驾仪，以“空速 + 高度 + 航向”作为控制接口，完成硬件在环仿真对接控制验证。

实验设计

步骤一：建立“对地速度”制导模型与“空速 + 高度 + 航向”制导模型的关系

无人机在地面坐标系下的位置与无人机在北东地坐标系中的位置相对应，可表示为：

$$\begin{bmatrix} p^{\mathrm{g}}_{\mathrm{r},x} \\ p^{\mathrm{g}}_{\mathrm{r},y} \\ p^{\mathrm{g}}_{\mathrm{r},z} \end{bmatrix} = \begin{bmatrix} p_{x_{\mathrm{e}}} \\ p_{y_{\mathrm{e}}} \\ H \end{bmatrix}$$

$$\mathbf{p}^{\mathrm{g}}_{\mathrm{r}} = \left[p^{\mathrm{g}}_{\mathrm{r},x} \; p^{\mathrm{g}}_{\mathrm{r},y} \; p^{\mathrm{g}}_{\mathrm{r},z} \right]^{\mathrm{T}}$$

无人机在地面坐标系下的位置

- p_{xe}：无人机在北东地坐标系中的北向位置
- p_{ye}：无人机在北东地坐标系中的东向位置
- H：无人机相对于地面的实际高度

实验设计

步骤一：建立“对地速度”制导模型与“空速 + 高度 + 航向”制导模型的关系

在本章设计实验中已经推导出了速度控制模式下的制导模型，结合空速、高度和航向角控制模式下制导模型，可以得到两者之间的关系。

$$\mathbf{v}_{\mathrm{r,ref}}^{\mathrm{g}} = \left[v_{\mathrm{r},x\mathrm{ref}}^{\mathrm{g}}\ v_{\mathrm{r},y\mathrm{ref}}^{\mathrm{g}}\ v_{\mathrm{r},z\mathrm{ref}}^{\mathrm{g}}\right]^{\mathrm{T}}$$

受油机在地面坐标系下的参考速度

$$\begin{cases} v_{\mathrm{r},x\mathrm{ref}}^{\mathrm{g}} = \dfrac{1}{G_x}\left(V_{\mathrm{a}}\cos\psi + v_{\mathrm{wx}}\right) \\ v_{\mathrm{r},y\mathrm{ref}}^{\mathrm{g}} = \dfrac{1}{G_y}\left(V_{\mathrm{a}}\sin\psi + v_{\mathrm{wy}}\right) \\ v_{\mathrm{r},z\mathrm{ref}}^{\mathrm{g}} = \dfrac{1}{G_z}\dot{H} \\ \psi = \chi + f\left(v_{\mathrm{wx}}, v_{\mathrm{wy}}\right) \\ V_{\mathrm{a}} = G_{V_{\mathrm{a}}} V_{\mathrm{ad}} \\ \chi = G_{\chi}\chi_{\mathrm{d}} \\ H = G_H H_{\mathrm{d}} \end{cases}$$

G_x, G_y, G_z：从 $v_{r,ref}^{g}$ 到 v_r^g 的传递函数

在硬件在环实验中，因风速较小，v_{wx}, v_{wy} 和 $f(v_{wx}, v_{wy})$ 均可以忽略

实验设计

步骤一：建立“对地速度”制导模型与“空速 + 高度 + 航向”制导模型的关系

实验中无人机的对接速度较慢，其动态过程可忽略，$G_x, G_y, G_z, G_{Va}, G_\chi, G_H$ 均视为 1。因此，可化为：

$$\begin{cases} v^{\mathrm{g}}_{\mathrm{r},x\mathrm{ref}} = V_{\mathrm{ad}} \cos \chi_{\mathrm{d}} \\ v^{\mathrm{g}}_{\mathrm{r},y\mathrm{ref}} = V_{\mathrm{ad}} \sin \chi_{\mathrm{d}} \\ v^{\mathrm{g}}_{\mathrm{r},z\mathrm{ref}} = \dot{H}_{\mathrm{d}} \end{cases}$$

由两种制导模型之间的关系，及简单的三角函数关系，可以用“相对地面速度”接口表示出“空速 + 高度 + 航向”接口，得到：

$$\begin{cases} V_{\mathrm{ad}} = \sqrt{(v^{\mathrm{g}}_{\mathrm{r},x\mathrm{d}})^2 + (v^{\mathrm{g}}_{\mathrm{r},y\mathrm{d}})^2} \\ \chi_{\mathrm{d}} = \arcsin \dfrac{v^{\mathrm{g}}_{\mathrm{r},y\mathrm{d}}}{V_{\mathrm{ad}}} = \arcsin \dfrac{1}{\sqrt{(\frac{v^{\mathrm{g}}_{\mathrm{r},x\mathrm{d}}}{v^{\mathrm{g}}_{\mathrm{r},y\mathrm{d}}})^2 + 1}} \\ H_{\mathrm{d}} = \int_0^t v^{\mathrm{g}}_{\mathrm{r},z\mathrm{d}} + H_0 \end{cases}$$

H_0：无人机进入图像伺服控制前的初始高度

硬件在环仿真实验

实验设计

步骤二：设计空速、高度和航向角控制模式下的图像伺服控制器

在硬件在环仿真实验中，使用的“空速 + 高度 + 航向”控制接口是 Python 中封装的顶层接口。与设计实验中不同的是：

- 无法通过设计内环的 LQR 控制器对底层控制接口实现增稳控制；
- 仅通过设计外环的比例控制器难以抑制干扰，无人机在对接时容易产生较大幅度机身摆动或振荡。

→ 改进：在外环控制中加入微分阻尼，从而抑制对接时的机身摆动及振荡。

加入微分阻尼后横向通道和纵向的期望速度分别为：

$$v_{\mathrm{d},xd}^{\mathrm{c}} = k_{xp} e_x + k_{xd} \dot{e}_x + z_{\mathrm{d}}^{\mathrm{c}} e_x e_y \omega_{\mathrm{d},x}^{\mathrm{c}} - z_{\mathrm{d}}^{\mathrm{c}} (1 + e_x^2) \omega_{\mathrm{d},y}^{\mathrm{c}} + z_{\mathrm{d}}^{\mathrm{c}} e_y \omega_{\mathrm{d},z}^{\mathrm{c}}$$

$$v_{\mathrm{d},yd}^{\mathrm{c}} = k_{yp} e_y + k_{yd} \dot{e}_y - z_{\mathrm{d}}^{\mathrm{c}} e_x e_y \omega_{\mathrm{d},y}^{\mathrm{c}} + z_{\mathrm{d}}^{\mathrm{c}} (1 + e_y^2) \omega_{\mathrm{d},x}^{\mathrm{c}} - z_{\mathrm{d}}^{\mathrm{c}} e_x \omega_{\mathrm{d},z}^{\mathrm{c}}$$

- k_{xp}，k_{yp}：比例过程系数
- k_{xd}，k_{yd}：微分过程系数

实验设计

步骤二：设计空速、高度和航向角控制模式下的图像伺服控制器

- 若$v_{\mathrm{d},x}^{\mathrm{c}} = v_{\mathrm{d},x\mathrm{d}}^{\mathrm{c}}$，则可以得到式子

$$\dot{e}_x = -\frac{k_{xp} - v_{\mathrm{d},z}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}} + k_{xd}} e_x$$

选取 $k_{xp} > max(v_{\mathrm{d},z}^{\mathrm{c}})$，即可保证 $\lim_{t\to\infty} |e_x(t)| = 0$ → **横向通道**的图像误差收敛至 0。

- 若$v_{\mathrm{d},y}^{\mathrm{c}} = v_{\mathrm{d},y\mathrm{d}}^{\mathrm{c}}$，则可以得到式子

$$\dot{e}_y = -\frac{k_{yp} - v_{\mathrm{d},z}^{\mathrm{c}}}{z_{\mathrm{d}}^{\mathrm{c}} + k_{yd}} e_y$$

选取 $k_{yp} > max(v_{d,z}^{c})$，即可保证 $\lim_{t\to\infty} |e_y(t)| = 0$ → **纵向通道**的图像误差收敛至 0。

实验设计

步骤二：设计空速、高度和航向角控制模式下的图像伺服控制器

深度方向上的期望速度设计：

- 与设计实验中一致，同样忽略锥套的角速度，认为 $\omega_d^c = \omega_c^{re} = R_{re/t}\omega_c^t \approx [\omega_{yb} \quad \omega_{zb} \quad \omega_{xb}]^T$
- 将 v_d^{re} 视为扰动，令 $v_{c,d}^{re} = v_{d,d}^c$，$v_d^c = v_{d,d}^c$

得到改进后无人机相对地面的期望速度，即 $v_{r,d}^g$

$$\begin{cases} v_{\mathrm{r,xd}}^{\mathrm{g}} = -k_z z_{\mathrm{d}}^{\mathrm{c}} - k_1 |e_x| - k_2 |e_y| + V_{\mathrm{t}}^{\mathrm{g}} - z_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{y_{\mathrm{b}}} + y_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{z_{\mathrm{b}}} \\ v_{\mathrm{r,yd}}^{\mathrm{g}} = k_{xp} e_x + k_{xd} \dot{e}_x + z_{\mathrm{d}}^{\mathrm{c}} e_x e_y \Delta\omega_{y_{\mathrm{b}}} - z_{\mathrm{d}}^{\mathrm{c}} (1 + e_x^2) \Delta\omega_{z_{\mathrm{b}}} + z_{\mathrm{d}}^{\mathrm{c}} e_y \Delta\omega_{x_{\mathrm{b}}} - x_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{z_{\mathrm{b}}} + z_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{x_{\mathrm{b}}} \\ v_{\mathrm{r,zd}}^{\mathrm{g}} = k_{yp} e_y + k_{yd} \dot{e}_y - z_{\mathrm{d}}^{\mathrm{c}} e_x e_y \Delta\omega_{z_{\mathrm{b}}} + z_{\mathrm{d}}^{\mathrm{c}} (1 + e_y^2) \Delta\omega_{y_{\mathrm{b}}} - z_{\mathrm{d}}^{\mathrm{c}} e_y \Delta\omega_{x_{\mathrm{b}}} - y_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{x_{\mathrm{b}}} + x_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{y_{\mathrm{b}}} \end{cases}$$

k_z：深度方向上的比例系数

实验设计

步骤二：设计空速、高度和航向角控制模式下的图像伺服控制器

综合式子，可得空速、高度和航向角控制模式下的图像伺服控制器：

k_1, k_2：调节无人机速度的反馈系数，在图像误差过大时减慢速度以避免超调

$$
\begin{cases}
v_{\mathrm{r},xd}^{\mathrm{g}} = -k_z z_{\mathrm{d}}^{\mathrm{c}} - k_1 |e_x| - k_2 |e_y| + V_{\mathrm{t}}^{\mathrm{g}} - z_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{y_{\mathrm{b}}} + y_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{z_{\mathrm{b}}} \\
v_{\mathrm{r},yd}^{\mathrm{g}} = k_{xp} e_x + k_{xd} \dot{e}_x + z_{\mathrm{d}}^{\mathrm{c}} e_x e_y \Delta\omega_{y_{\mathrm{b}}} - z_{\mathrm{d}}^{\mathrm{c}} (1 + e_x^2) \Delta\omega_{z_{\mathrm{b}}} + z_{\mathrm{d}}^{\mathrm{c}} e_y \Delta\omega_{x_{\mathrm{b}}} - x_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{z_{\mathrm{b}}} + z_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{x_{\mathrm{b}}} \\
v_{\mathrm{r},zd}^{\mathrm{g}} = k_{yp} e_y + k_{yd} \dot{e}_y - z_{\mathrm{d}}^{\mathrm{c}} e_x e_y \Delta\omega_{z_{\mathrm{b}}} + z_{\mathrm{d}}^{\mathrm{c}} (1 + e_y^2) \Delta\omega_{y_{\mathrm{b}}} - z_{\mathrm{d}}^{\mathrm{c}} e_y \Delta\omega_{x_{\mathrm{b}}} - y_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{x_{\mathrm{b}}} + x_{\mathrm{c}}^{\mathrm{r}} \Delta\omega_{y_{\mathrm{b}}} \\
V_{\mathrm{ad}} = \sqrt{(v_{\mathrm{r},xd}^{\mathrm{g}})^2 + (v_{\mathrm{r},yd}^{\mathrm{g}})^2} \\
\chi_{\mathrm{d}} = \arcsin \dfrac{v_{\mathrm{r},yd}^{\mathrm{g}}}{V_{\mathrm{ad}}} = \arcsin \dfrac{1}{\sqrt{(\frac{v_{\mathrm{r},xd}^{\mathrm{g}}}{v_{\mathrm{r},yd}^{\mathrm{g}}})^2 + 1}} \\
H_{\mathrm{d}} = \int_0^t v_{\mathrm{r},zd}^{\mathrm{g}} + H_0
\end{cases}
$$

H_0：无人机进入图像伺服控制前的初始高度

实验步骤

步骤一：完成 Pytorch 环境配置

在 Python 中运行 YOLO 图像识别需要配置 Pytorch 及一系列依赖包：

- 在所在系统中安装 Anaconda。
- 如图，可以打开命令行输入“conda -V”检验是否安装以及查看当前 conda 的版本。
- 使用“conda create -n AAR_plat python=3.9.6”命令创建 Python 版本为 3.9.6、名字为“AAR_plat”的虚拟环境。“AAR_plat”文件夹可以在 Anaconda 安装目录“envs”文件夹下找到。

图. Python 环境配置

硬件在环仿真实验

实验步骤

步骤一：完成 Pytorch 环境配置

- 使用“activate AAR_plat”命令激活名为“AAR_sim”的虚拟环境，并通过“cd Anaconda/envs/AAR_plat/Scripts”命令在命令行中打开用于配置虚拟环境的文件夹。
- 将实验包“e7-4”中“CopterSimulink_HITL”文件夹的“requirements.txt”文件和“packages”文件夹放入虚拟环境的“envs/AAR_plat/Scripts”文件夹下，并在命令行输入“pip install --no-index --findlinks=packages -r requirements.txt”命令，虚拟环境会自动离线安装“requirements.txt”文件中所有的包。

➢ 仅需要 CPU 完成锥套识别程序的计算 → Pytorch 环境配置已完成

➢ 需要使用 Nvidia 系列 GPU 完成锥套识别程序计算 → 进一步完成 CUDA 配置，以获得更佳的识别效果

实验步骤

步骤一：完成 Pytorch 环境配置

CUDA 的配置：

- 如图，查询本机显卡支持的 CUDA 版本，在命令行中输入“nvidiasmi” 命令即可得到如红框内所示的显卡最高支持的 CUDA 版本。
- 在 Nvidia 官网中下载并安装对应版本的CUDA。
- 进入 Pytorch 官网查询对应 CUDA 版本的 Pytorch 下载链接。在命令行中激活“AAR_sim” 虚拟环境后粘贴链接并安装 Pytorch 包。

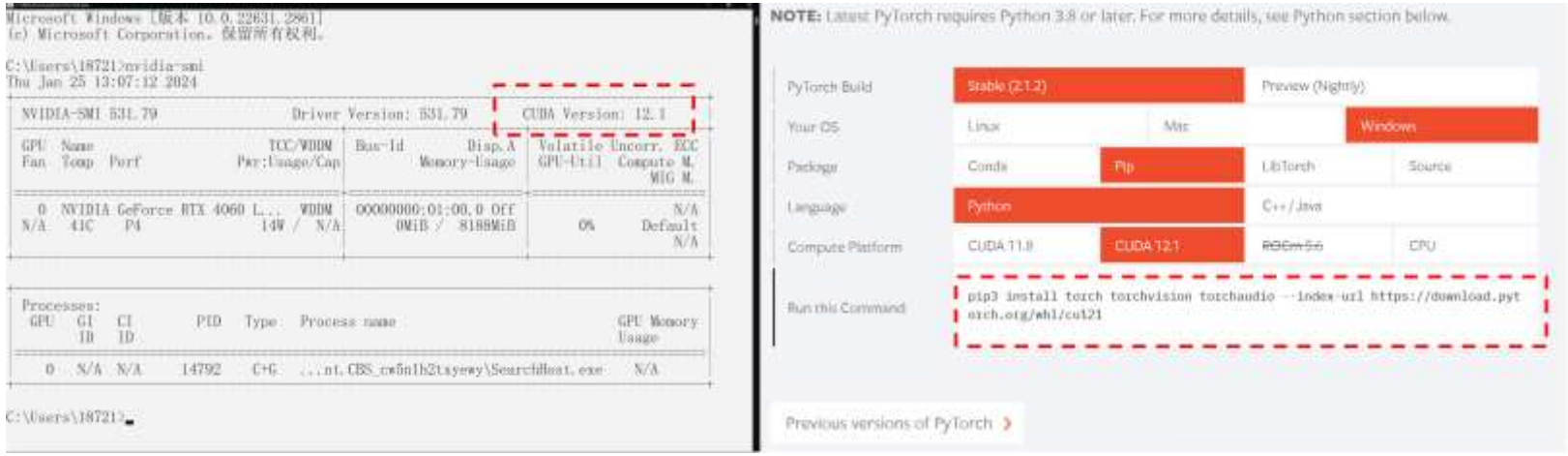


图. CUDA 版本查询

实验步骤

步骤二：实现“空速 + 高度 + 航向”控制模式下的图像伺服控制器

控制无人机的代码位于“CopterSimulink_HITL”文件夹中“Mytrajectory.py”文件的“SimpleRoute()”函数中。“SimpleRoute()”函数主要分为五个阶段，用标志位“flag”表示。

- 在第一阶段，即“flag=0”时，无人机解锁；
- 第二阶段，即“flag=1”时，无人机起飞并进入 Offboard 模式；
- 第三阶段，即“flag=2”时，进入姿态控制模式，稳定爬升姿态；
- 第四阶段，即“flag=3”时，进入速度高度偏航控制模式，定高直线巡航并追及母舰；
- 第五阶段，即“flag=4”时开启 YOLO 图像识别，实现“空速 + 高度 + 航向”控制模式下的图像伺服控制器。

硬件在环仿真实验

实验步骤

步骤二：实现“空速 + 高度 + 航向”控制模式下的图像伺服控制器

在第五阶段中，

- 输入：图像伺服控制器将图像误差、深度估计结果以及无人机的角速度
- 输出：无人机的空速期望、航迹偏角期望和高度期望

→ 由函数“SendVelYawAlt”即“空速 + 高度 + 航向”顶层控制器执行。

```
ex = (obj_xx - px) / 640 # unitisation of image error
ey = (obj_yy - py) / 360
ez = obj_zz
if flagIII == 1:
    ex_last = ex # first difference is zero
    ey_last = ey
d_pr_rc = [6.5, 0, -2.1] # the distance between the camera and the receiver
Vgt = 14.7 # mothership speed
k1 = 0.002 # set control parameters
k2 = 0.002
kx_p = 3.6
kx_d = 0.12
ky_p = 0.1
ky_d = 0.1
kz = 0.03
```

对图像误差进行单位化

设置初始微分过程的差值为 0

相机相对无人机中心的位置

母舰飞行速度

设置控制参数

“空速 + 高度 + 航向”控制模式下的图像伺服控制器

硬件在环仿真实验

实验步骤

步骤二：实现“空速 + 高度 + 航向”控制模式下的图像伺服控制器

```
if abs(obj_xx - px) <= 30 and abs(obj_yy - py) <= 30 and obj_zz <= 0.5:
    Va = 15 # docking success when less image error and depth
    Chi = -mav.uavPosNED[1] / 50
    H = -92.2
    flag = 5
else:
    Vgx = kz*(ez)-k1*abs(ex)-k2*abs(ey)+Vgt-d_pr_rc[2]*q+d_pr_rc[1]*r # calculate desired velocity
        of receiver
    Vgy = kx_p*(ex)+kx_d*(ex-ex_last)+ez*ex*ey*q-ez*(1+ex**2)*r+ez*ey*p-d_pr_rc[0]*r+d_pr_rc[2]*p
    Vgz = ky_p*(ey)+ky_d*(ey-ey_last)-ez*ex*ey*r+ez*(1+ey**2)*q-ez*ey*p-d_pr_rc[1]*p+d_pr_rc[0]*q
    Va = math.sqrt(Vgx**2 + Vgy**2) # calculate desired velocity, angle and height of receiver
    Chi = math.asin(Vgy/Va)
    H = Vgz + H
mav.SendVelYawAlt(Va, Chi, H) # controller send commands
ex_last = ex # calculate the difference
ey_last = ey
```

“空速 + 高度 + 航向”控制模式下的图像伺服控制器

判断无人机是否与母舰对接成功，成功则无人机加速撞击回收装置

先得到对地期望速度，再计算期望空速、期望航迹偏角和期望高度

“空速 + 高度 + 航向”顶层控制器执行期望指令

记录上一次的图像误差，计算微分过程

- 程序采用了位置式 pd 控制，微分过程可表示为 $k_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}$。而每次程序执行指令的间隔相等，即 dt 为常数，可以将微分过程的系数 k_d 与时间间隔 $\mathrm{d}t$ 合并为一个系数

→ 程序中的常量“k_d”。

硬件在环仿真实验

实验步骤

步骤二：运行硬件在环仿真，分析空中对接实验结果

模型导入：

- 将“module”文件夹中的母舰模型“WestTrans_x0002_portC130J”，无人机模型“MQ-9Reaper”和锥套模型“ProbeLightScene”复制到“PX4PSP_RflySim3D_RflySim3D_Content”文件夹下

修改飞控固件，重新通过 QGroundControl 地面站烧录到自驾仪中：

- 在“Copter_x0002_Simulink_HITL/Firmware”文件夹中存放有修改后对应型号的飞控固件
- 自驾仪： CubePilot 自驾仪、使用“PX-4_fmu-v5”固件的自驾仪

硬件在环仿真实验

实验步骤

步骤二：运行硬件在环仿真，分析空中对接实验结果

以CubeOrange 自驾仪为例：

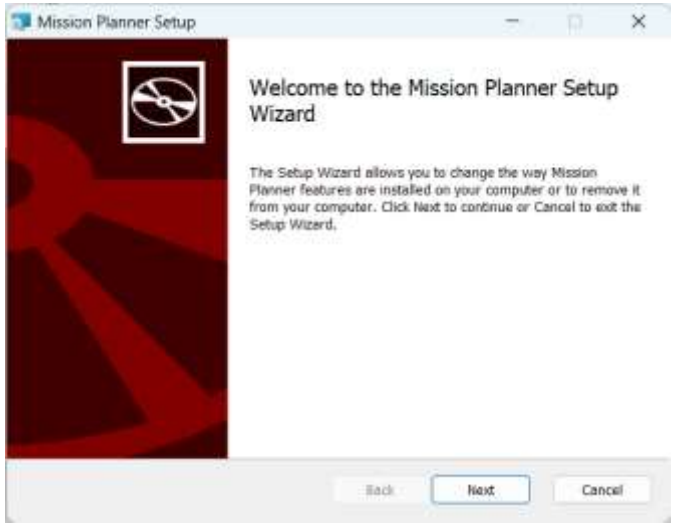


图. 驱动软件安装

安装 CubePilot 自驾仪的驱动软件。

图. 飞控固件烧录

通过 QGroundControl 地面站将飞控固件烧录至自驾仪中。

图. 固件烧录成功

QGroundControl 地面站进行飞控固件烧录，提示“升级完成”即完成烧录。

硬件在环仿真实验

实验步骤

步骤二：运行硬件在环仿真，分析空中对接实验结果

完成固件烧录后即可进行硬件在环仿真实验：

图. 硬件连接

在电脑上接入 CubePilot/Pixhawk 自驾仪，在出现打开实验包 “e7-4” 中的 “CopterSimulink_HITL” 文件夹，运行 “PX4SimHITL.bat” 批处理文件；

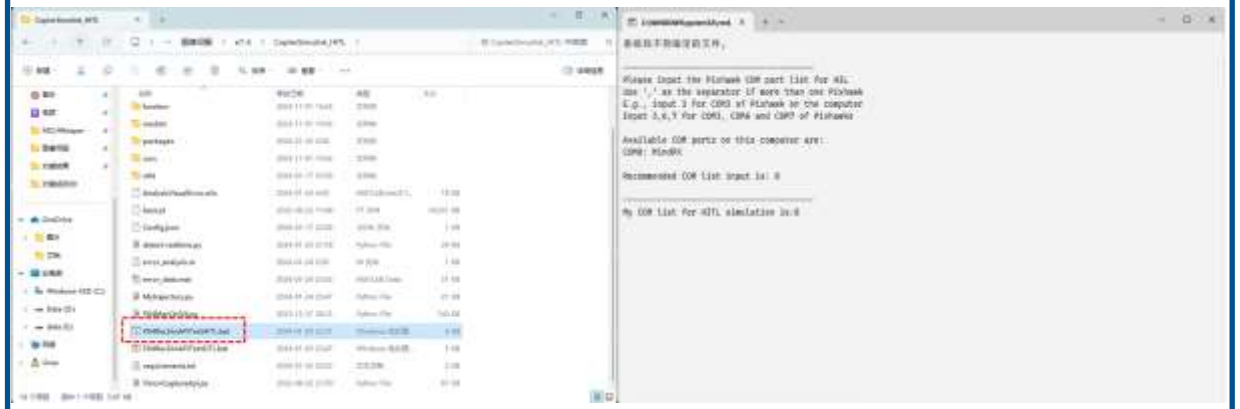


图. RflySim 硬件在环仿真平台设置

弹出如图所示界面，在终端窗口内输入推荐的串口号，按下回车键，一键启动 RflySim 硬件在环仿真平台。

硬件在环仿真实验

实验步骤

步骤二：运行硬件在环仿真，分析空中对接实验结果

图. 准备起飞

切换到 QGC 地面站界面，当地面站左上角显示无人机处于“Ready to fly”状态后，运行“Simulink”文件夹中母舰运动仿真平台“AirRefueling_Platform.slx”。

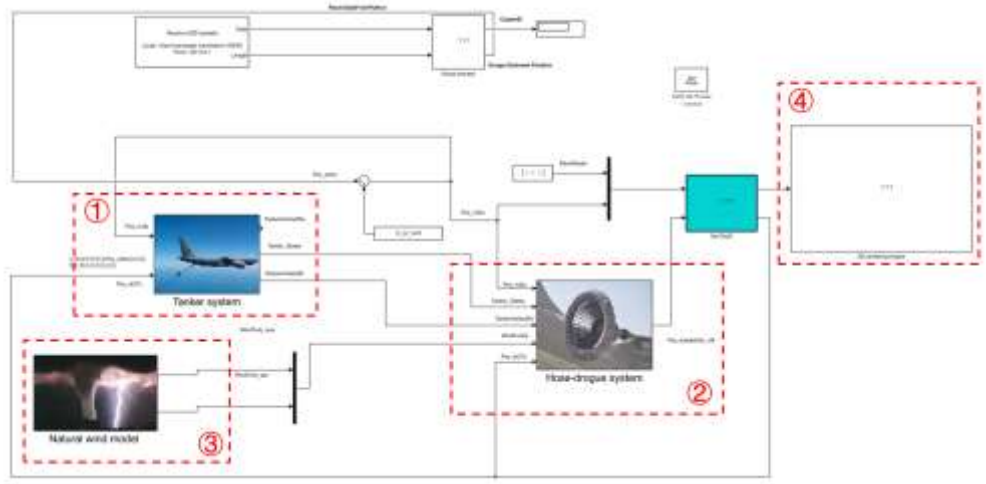


图. 母舰运动 Simulink 仿真

母舰运动仿真平台主要由四大部分组成：

①加油机模块 “Tanker system”

②锥套模块 “Hose-drogue system”

③环境设置模块 “Natu_x0002_ral wind model”

④母舰轨迹规划模块 “3D rendering engine”

硬件在环仿真实验

实验步骤

步骤二：运行硬件在环仿真，分析空中对接实验结果

母舰的运动速度及运动轨迹可以在母舰轨迹规划模块即“AirRefueling_Platform/3Drendering engine”路径下用“MATLAB Function”进行设置。

- 输入：时间
- 输出：母舰的位置

```
function y = fcn(t)
    y = [-100 -119 -100]; % initial position
    v1 = 15; % velocity
    y(1) = y(1) + v1 * t; % trajectory
end
```

→ 母舰的起始位置
→ 母舰的速度
→ 母舰的运动轨迹

母舰速度及运动轨迹设置

硬件在环仿真实验

实验步骤

步骤二：运行硬件在环仿真，分析空中对接实验结果

在本实验中母舰设置为定高直线飞行，起始坐标为 (−100,−119,−100)，速度为 15m/s。

- 此处坐标系为北东地坐标系，因此母舰飞行方向为正北，定高 100m。
- 运行“CopterSimulink_HITL” 文件夹中的无人机控制程序 “Mytrajectory.py”。
- 等待编译器终端中输出 “Waiting forclient to connect to named pipe...” 后，再运行锥套识别程序 “detect-realtime.py”。
- 为节省计算机内存，设定无人机离母舰足够近时才会开始进行图像处理，在前四个阶段不会开启机载摄像头。

硬件在环仿真实验

实验步骤

步骤二：运行硬件在环仿真，分析空中对接实验结果

在无人机飞行过程中：

- CopterSim → 实时观察无人机的坐标、三维速度和姿态角
- RflySim3D 和 “detect-realtime.py” → 开启机载摄像机实时观察无人机运动

图. 对接画面

硬件在环仿真实验

实验步骤

步骤二：运行硬件在环仿真，分析空中对接实验结果

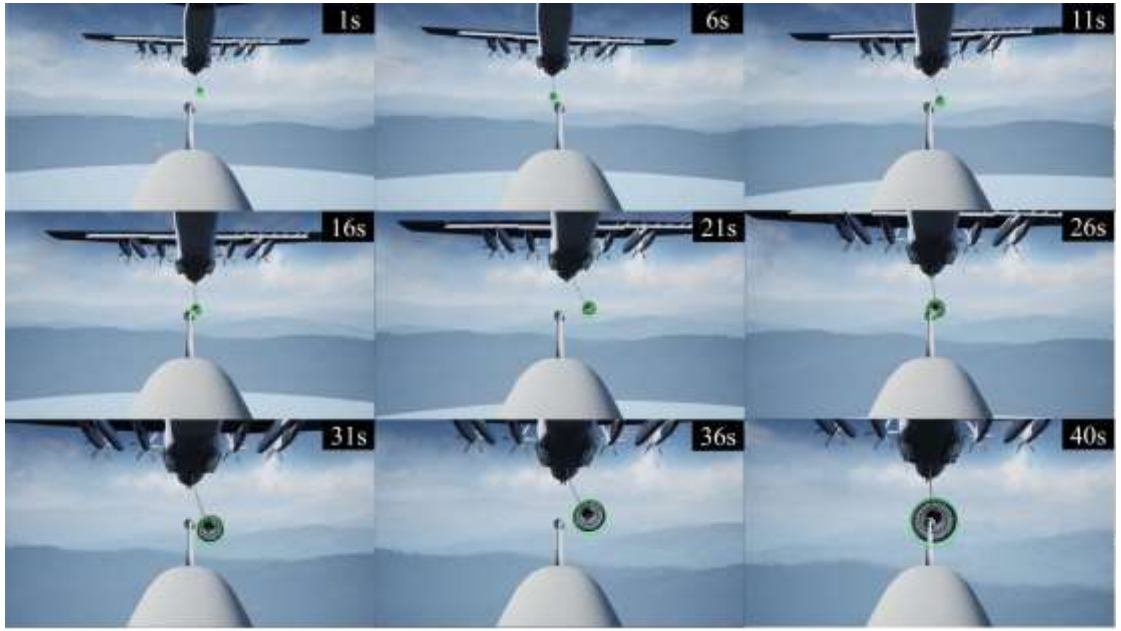


图. 对接过程

无人机与母舰能成功完成对接任务

- 在无人机与母舰对接的过程中，“Mytrajectory.py”程序会不断记录图像误差、深度误差和位置误差信息，并保存为“error_data.mat”文件
- 运行误差分析绘图程序“AnalysisVisualError.mlx”→ 得到对接过程中的误差曲线

硬件在环仿真实验

实验步骤

步骤二：运行仿真，分析空中加油对接实验结果

“图像误差”： 在图像坐标系中，锥套中心与无人机锥管位置的差。

- 在 40s 内收敛至零。

“深度误差”： 无人机锥管顶点与锥套平面的距离。

- 深度误差越小，深度误差衰减速度越小，与深度方向的图像伺服控制器设计相一致。

“位置误差”： 在世界坐标系中，锥套与无人机锥管绝对位置的差。

- 在 40s 内收敛至零

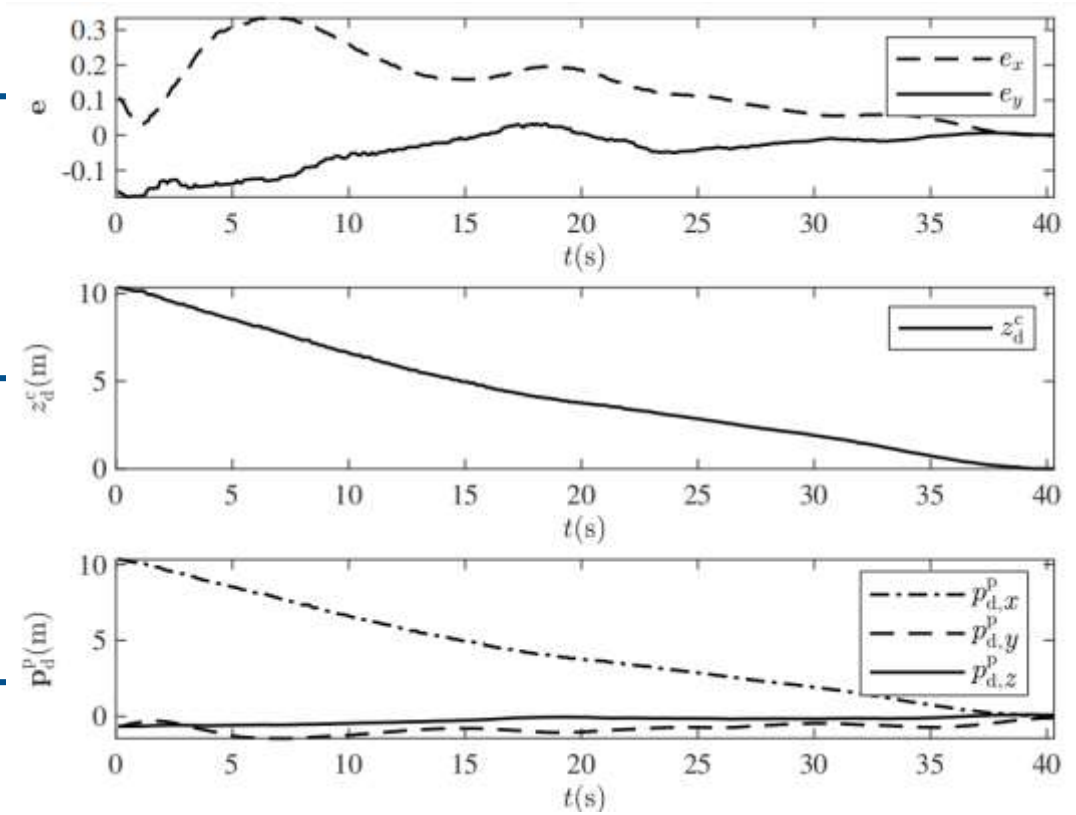


图. 硬件在环仿真误差曲线

硬件在环仿真实验

实验步骤

步骤二：运行硬件在环仿真，分析空中对接实验结果

图像误差和位置误差曲线虽然出现了振荡，

但振荡不断衰减，最终误差趋于零

改进的横侧向和纵向方向上 PD 控制器能有效抑制机身振荡和摆动，有较好的抗干扰能力，在缺少内环 LQR 方法增稳的情况下依然能使对接误差收敛至零，成功完成对接任务。


将CubeOrange 自驾仪
与计算机连接

致谢

- 感谢刘润潇，陈汉励为本讲课程准备做出的贡献

- 更多信息请访问公众号和网站

rfly.buaa.edu.cn