实验2 Crazyflie无人机单机运动控制

**一、实验目的**

1. 熟悉动捕系统Optitrack在无人机上的应用

2. 掌握Crazyflie单机运动控制

**二、实验设备**

1. Ubuntu系统电脑

2. Crazyflie无人机 \* 1

3. Crazyradio \* 1

4. 动捕系统Optitrack

**三、实验原理**

 

图2-1 装有四个动捕标记球的Crazyflie 图2-2 Crazyradio PA、

Crazyflie是一个多功能的开源飞行开发平台，重量仅27g。其可以通过蓝牙与手机或电脑通信。也可以通过Crazyradio PA能实现更低延迟的通信，在计算机上显示数据与使用计算机远程控制。

Crazyflie有一个9轴惯性测量单元（IMU），由陀螺仪、加速度计和磁力计组成。可以使用压力传感器来估计高度。大部分处理在主微控制器（STM32）上完成。它运行FreeRTOS作为其操作系统，状态估计和姿态控制以250Hz执行。第二微控制器（nRF51）用于无线通信并用作电源管理器。这两个微控制器可以通过syslink交换数据，这是一种使用UART作为物理接口的协议。扩展端口允许添加附加硬件。

机载传感器足以稳定姿态，但不能稳定位置。在本实验中，我们将使用完全自主飞行的动作捕捉系统。通过crazyswarm功能包提供的SDK编写Python脚本可以实现crazyfl­ie的飞行控制。进一步地，还可以使用ROS编写crazyflie飞行控制器，设计无人机姿态控制器。

**四、实验准备**

实验所需的软件环境：ubuntu20.04 + ROS Noetic + crazyswarm功能包

1. Ubuntu系统安装

建议安装Windows+Ubuntu双系统，参考网络教程。

1. ROS安装

[下载官方链接](http://wiki.ros.org/cn/noetic/Installation/Ubuntu)、[安装过程bug解决参考](https://blog.csdn.net/qq_44339029/article/details/120579608?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2~default~BlogCommendFromBaidu~default-1-120579608-blog-106899359.pc_relevant_baidufeatures_v6&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2~default~BlogCommendFromBaidu~default-1-120579608-blog-106899359.pc_relevant_baidufeatures_v6&utm_relevant_index=2)。

Tips：确保Ubuntu apt换为国内镜像源

1. Crazyradio配置

以下参考[crazyradio权限设置参考](https://www.bitcraze.io/documentation/repository/crazyflie-lib-python/master/installation/usb_permissions/)

1. sudo groupadd plugdev
2. sudo usermod -a -G plugdev $USER
4. sudo gedit /etc/udev/rules.d/99-bitcraze.rules
5. # Crazyradio (normal operation)
6. SUBSYSTEM=="usb", ATTRS{idVendor}=="1915", ATTRS{idProduct}=="7777", MODE="0664", GROUP="plugdev"
7. # Bootloader
8. SUBSYSTEM=="usb", ATTRS{idVendor}=="1915", ATTRS{idProduct}=="0101", MODE="0664", GROUP="plugdev"
9. # Crazyflie (over USB)
10. SUBSYSTEM=="usb", ATTRS{idVendor}=="0483", ATTRS{idProduct}=="5740", MODE="0664", GROUP="plugdev"
11. sudo udevadm control --reload-rules
12. sudo udevadm trigger

crazyradio（PA）固件更新:

1. cd ~/crazyflie\_docs
2. git clone https://github.com/bitcraze/crazyradio-firmware.git
3. sudo apt-get install sdcc binutils
5. #编译
6. cd ~/crazyflie\_docs/crazyradio-firmware/firmware
7. make CRPA=1
9. #烧写前先安装python3-usb包
10. sudo apt install python3-usb
12. #烧写
13. python3 ../usbtools/launchBootloader.py
14. sudo python3 ../usbtools/nrfbootload.py flash bin/cradio.bin
16. #重新插拔crazyradio，检查版本应为version 99.55
17. lsusb -d 1915:7777 -v | grep bcdDevice
18. crazyflie客户端安装

下载crazyflie-lib-python和crazyflie-client-python包

1. git clone https://github.com/bitcraze/crazyflie-lib-python.git
2. git clone https://github.com/bitcraze/crazyflie-clients-python.git
3. #分别在这两个目录下，安装和配置依赖
4. cd crazyflie-lib-python
5. pip3 install -e .
6. cd crazyflie-clients-python
7. pip3 install -e .
8. #在crazyflie-client-python目录下输入下列命令可以启动客户端
9. python3 -m cfclient.gui
10. #客户端GUI的作用一般是更新固件和配置ID
12. #crazyflie ID说明
13. #默认地址为E7E7E7E7E7
14. #crazyflie命名修改最后两位，并在crazyflie上做好标记防止忘记地址
15. #uri1:=radio://0/100/2M/E7E7E7E701  uri2:=radio://0/100/2M/E7E7E7E702
16. Crazyflie配置
17. #设置环境变量
18. export CSW\_PYTHON=python3
19. #安装依赖
20. sudo apt install -y ros-noetic-tf ros-noetic-tf-conversions ros-noetic-joy
21. sudo apt install -y libpcl-dev libusb-1.0-0-dev
22. sudo apt install -y swig lib${CSW\_PYTHON}-dev ${CSW\_PYTHON}-pip
23. ${CSW\_PYTHON} -m pip install pytest numpy PyYAML scipy
24. ${CSW\_PYTHON} -m pip install vispy
26. git clone https://github.com/USC-ACTLab/crazyswarm.git
27. cd crazyswarm
28. git submodule update --init --recursive
29. git submodule update --recursive
30. #编译
31. ./build.sh
32. #验证安装
33. cd ros\_ws/src/crazyswarm/scripts
34. source ~/crazyswarm/ros\_ws/devel/setup.bash
35. $CSW\_PYTHON -m pytest

可参考[crazyswarm官方安装链接](https://crazyswarm.readthedocs.io/en/latest/installation.html)

**五、实验步骤**

可安装使用VSCode方便代码编写。

1. **Crazyswarm中动捕系统与局域网设置**
2. motion\_capture\_type: "optitrack".        # 选择动捕的版本optitrack or vicon
3. motion\_capture\_host\_name: "192.168.2.17" # 设置局域网的Ip
4. object\_tracking\_type: "libobjecttracker" # 默认参数 one of motionCapture, libobjecttracke
5. **Motive中设置**

● Up axis: Z

● 若object\_tracking\_type:libobjecttracker,关闭all assets和打开labeled or unlabeled markers

1. **Crazyflie标记点文件配置**
2. **crazyflies.yaml文件设置，每次飞行需要设置初始点。**

路径：ros\_ws/src/crazyswarm/launch/crazyflies.yaml

crazyswarm服务器在启动时会读取配置文件crazyflies.yaml

● 单个标记排列，crazyflies.yaml条目的initialPosition字段将被忽略，但仍然应该设置它，因为解析器需要它。

● 多个标记排列，initialPosition必须正确。在运动捕捉设备的坐标系中，位置以米为单位指定。不要求CF精确地从这些位置开始，几厘米的变化是可以的。

1. **crazyflieTypes.yaml设置，若crazyflie标记点更改位置需要设置。**

● 单个标记点排列中，markerConfiguration字段不需要。

● 多个标记点排列中，每个排列需要用markerConfiguration字段，默认的标记排列为0。

新增标记点类型需要： 1.在运动捕捉空间的原点放置一个具有所需排列的CF。Crazyflie的正面应指向运动捕捉坐标系的x方向。2.获得标记点的坐标，roslaunch crazyswarm mocap\_helper.launch。 3.修改crazyflieTypes.yaml

1. **简单悬停**

单标记点，每次飞机起飞时正方向应朝x轴正方向。

多标记点的，在标记点参数配置时，也应朝x轴正方向，但以后起飞可以朝任意方向。

1. #启动crazyswarm\_server
2. source ~/crazyswarm/ros\_ws/devel/setup.bash
3. roslaunch crazyswarm hover\_swarm.launch
4. #新开一个终端，执行脚本，起飞悬停两秒后降落
5. python3 hello\_world.py
6. **实操任务**

**在无人机悬停实验的基础上实现无人机圆形飞行**

**案例：**

/home/uav2/crazyswarm/ros\_ws/src/crazyswarm/scripts/hello\_world.py

1. """Takeoff-hover-land for one CF. Useful to validate hardware config."""
3. from pycrazyswarm import Crazyswarm
5. TAKEOFF\_DURATION = 2.5
6. HOVER\_DURATION = 5.0
8. def main():
9. swarm = Crazyswarm()
10. timeHelper = swarm.timeHelper
11. cf = swarm.allcfs.crazyflies[0]
13. cf.takeoff(targetHeight=1.0, duration=TAKEOFF\_DURATION)
14. timeHelper.sleep(TAKEOFF\_DURATION + HOVER\_DURATION)
15. cf.land(targetHeight=0.04, duration=2.5)
16. timeHelper.sleep(TAKEOFF\_DURATION)
18. **if** \_\_name\_\_ == "\_\_main\_\_":
19. main()