
Paparazzi User's Manual

Ecole Nationale de l'Aviation Civile, Toulouse, France

October 2007

Updated: February 3, 2008

https://wiki.paparazziuav.org/w/images/0/0a/Users_manual.pdf

翻译: houfang hfhoufang@tom.com April 16, 2014

摘要

这是一个如何使用 Paparazzi 系统的快速指南。它分为两部分。第一部分是一个飞机准备、UAV 无人机系统操作和飞行结果分析的初略过程，也可当成操作清单使用。该过程的描述适用于一个给定了飞行方案、已经预编程和调试好的飞行器。第二部分对于系统的不同组件和工具进行了说明。

Paparazzi 项目处于持续演进中，最新的文档可以从项目 wiki 地址中得到：paparazzi.enac.fr。

重要声明：本手册基于一种特定的多发喷气发动机机体、特定的飞行方案（在 basic.xml 文件中描述）和特定的设置编写。其他配置下，会存在若干细节变化。

目录

Chapter 1.	快速指南.....	3
1.1.	飞行准备.....	3
1.1.1.	硬件.....	3
1.1.2.	模拟.....	3
1.1.3.	携带到场地的项目列表.....	6
1.2.	场地飞行.....	6
1.2.1.	安装.....	6
1.2.2.	飞行前准备.....	6
1.2.3.	起飞.....	7
1.2.4.	巡航.....	8
1.2.5.	着陆.....	10
1.2.6.	着陆后.....	11
1.3.	飞行后数据分析.....	11
1.3.1.	数据分析.....	11
Chapter 2.	参考手册.....	13
2.1.	Paparazzi Center.....	13
2.1.1.	飞行器.....	13
2.1.2.	编译和烧写.....	13
2.1.3.	模拟, 飞行和回放.....	14
2.1.4.	其他代理.....	14
2.2.	地面控制站 GCS.....	15
2.2.1.	控制块 Strips	15
2.2.2.	地图 Map	16
2.2.3.	飞行手册 Notebook.....	17
2.2.4.	控制台 Console.....	18
2.3.	飞行计划.....	19
2.3.1.	Flight Plan 文件结构.....	19
2.3.2.	Waypoints 文件结构.....	19
2.3.3.	Sector 块	20
2.3.4.	Blocks 区域.....	20
2.3.5.	Expressions 表达式	21
2.3.6.	Exception 异常.....	21
2.3.7.	Navigation Primitives 航线基元.....	21
2.3.8.	Edition 编辑.....	25
2.4.	机体配置文件.....	25
2.5.	设置.....	26
2.6.	调整.....	27
2.6.1.	Actuators Layer 作动器层.....	27
2.6.2.	Stabilization Layer 稳定层	27
2.6.3.	Navigation Layer 航线层.....	29
Chapter 3.	附录.....	30

Chapter 1. 快速指南

UAV 的操作可以分为以下三个部分：准备、飞行和报告。

1.1. 飞行准备

飞行准备可能会需要数小时，特别是在要对电池充电的情况下。

1.1.1. 硬件

电池

检查以下设备的电池：

1. 飞行器
2. RC 收发器
3. 笔记本电脑


机械部件

在每次飞行前进行以下检查：

1. 检查所有固定和活动的表面。检查所有可移动的部件是否被妥善安置。
2. 检查所有舵机的齿轮和摇臂（或通过舵机摇臂实现的控制面，例如升降舵或副翼）。
3. 调节各部分平衡：当飞行器支持 CG 标记时，必须水平稳定。如果需要，可以调整电池位置来配平。
4. 检查螺旋桨是否紧固。

1.1.2. 模拟

为了便于场地操作，事先进行任务模拟是一个好的方法。在具备 Internet 连接的情况下，可以下载地图来进行模拟操作。

在 GCS 电脑上，从  图标运行 Paparazzi Center（如 Figure 1.1）程序。（程序安装、运行方式参见 <https://wiki.paparazziuav.org/wiki/Installation>）

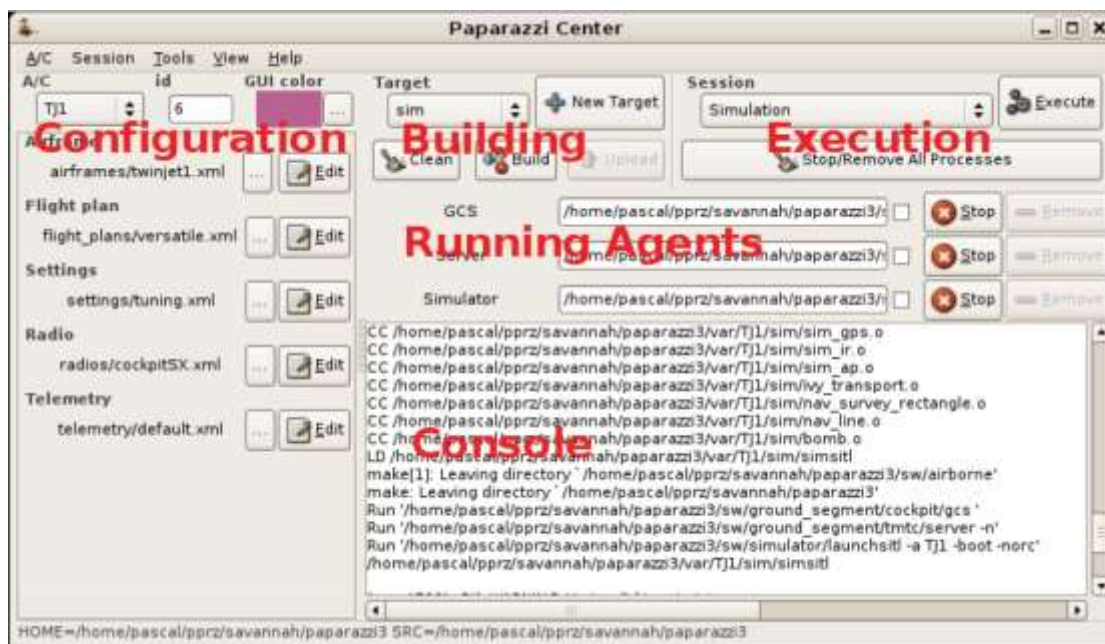




Figure 1.1

1. 在 A/C 组合框（界面左上）中，选择目标飞行器。
2. 在 Target 组合框（界面中上）中，选择 sim，点击 Build 按钮。（顺便提一句，如果飞行器的配置未改变，该操作无效）
3. 在 Session 组合框（界面右上），选择 Simulation，点击 Execute 按钮。一个用于飞行器的小窗口和一个用于 GCS 的大窗口会出现在屏幕上。在 Paparazzi Center 中，显示三个当前运行的代理进程（分别是 GCS，服务器和模拟器）。
4. 如果你打算改变飞行任务的默认地址，你需要重启模拟器代理：
 - （a） 停止模拟器代理（点击右侧的 Stop 按钮）。
 - （b） 编辑模拟器代理的命令行，移除 -boot 项。
 - （c） 重启模拟器代理（点击右侧的 Redo 按钮）。
 - （d） 在飞行器窗口中（此时会自动出现），点击 Set Pos 按钮，然后输入目的地址（使用 WGS84 十进制数据）和相对地面高度（单位：m）
5. GCS 窗口显示飞行器的飞行状态。可以通过 aircraft 菜单的 Center A/C 项使飞行器在地图上居中。通过 Nav 菜单的 Fit to window 项调整各显示元素（飞行器和路径点）。通过鼠标滚轮、PageUp 键，PageDown 键或者右上侧的滚动条缩放。GCS 的详细信息在 2.2 节中给出。
6. 如果你想要为飞行任务下载地图数据：
 - （a） 连接计算机到 Internet。
 - （b） 地图数据可以用以下方法下载：
 - 鼠标右键单击，在弹出菜单中选择 Download one tile，下载当前位置地图
 - 通过 maps 菜单的 Google Maps Fill 项为当前窗口下载所有地图
 - maps 菜单的 Google Maps auto 项可以在窗口移动缩放时自动下载所需地图数据

下载的地图数据会存放在缓存中，下次 GCS 运行时，可以从缓存读取。

7. 接下来在 GCS 中进行任务模拟飞行的操作就和在 1.2 节中描述的真实飞行一样了。经过 GPS 的初始化阶段（10s 左右），航线位于 Holding Point 区域。点击 Takeoff  转

换到起飞区，在飞行器窗口点击 Launch 或者 GCS 窗口的 .

8. 如果你想在模拟中考虑风的因素：

- (a) 在 Paparazzi 控制中心的 Tools 菜单中，运行 Environment Simulator 代理。代理会出现在运行的代理列表中，一个名为 Gaia 的窗口会显示。如 Figure 1.2 所示。



Figure 1.2

- (b) 在该窗口中，可以设置风向(从北向出发，顺时针方向的度数值)和风速(m/s)。

9. 模拟过程可以被加速，最大加速因子为 10 倍。时间尺度可以在 Environment Simulator 代理中设置。

10. 要启动一个新的模拟，可以从 Paparazzi Center 中进行。

- 仅仅重启模拟器代理，可以点击 Stor 和 Redo。
- 重启所有代理，点击 Stop/Remove all processes 和 Execute。

11. 所用的运行代理可以通过 Session 菜单被保存为一个会话，以备下次继续运行。

要退出，点击 Stop/Remove process，然后关闭 Paparazzi Center。

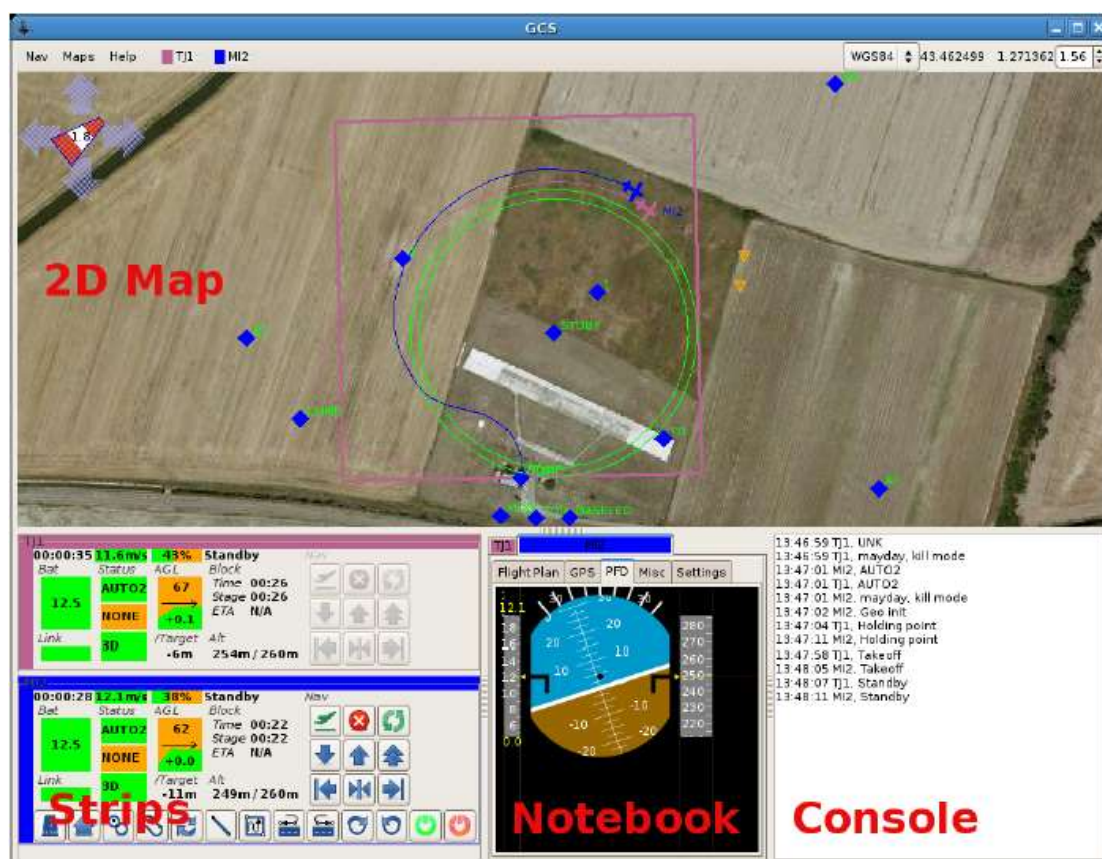


Figure 1.3

1.1.3. 携带到场地的项目列表

1. 包含所有可拆卸部件的完整飞行器。
2. RC 收发器。
3. 地面数据连接 modem。
4. 运行 GCS 的计算机，USB 连接线。
5. 为多次飞行准备的充电设备和电源。

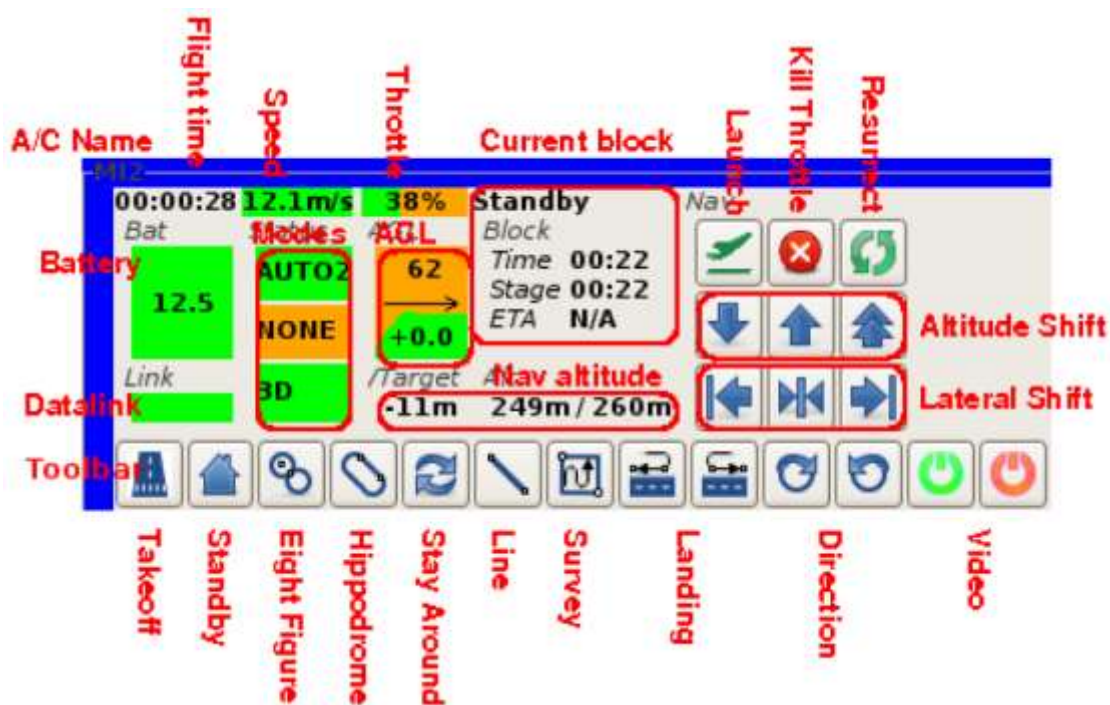


Figure 1.4

1.2. 场地飞行

1.2.1. 安装

1. 启动 Linux 操作系统电脑中的 GCS，运行 Paparazzi Center。
2. 将数据 Modem 插入 USB 接口。将天线调整为竖直方向。

1.2.2. 飞行前准备

按照以下步骤操作：

1. 在 Paparazzi Center 中选择并执行 Flight 对话。
2. 打开 RC 收发器，油门关闭，MANUAL 模式。
3. 飞行器装入电池，关闭电池仓。
4. 检查 GCS 的上行和下行连接。
5. 等待 GPS 信号俘获。

6. 载入地图（如果上次对话已经被下载缓存）。

飞行器：

1. 检查遥控器分别在 MANUAL 和 AUTO1 位置的升降翼行程和方向。
2. 检查遥控器分别在 MANUAL 和 AUTO1 位置的引擎工作情况（全油门）。
3. 检查 AUTO1 位置时的水平稳定性。

红外：持机鼻向右侧翘起 30° ，右升降翼应该升起，左升降翼下降。持一侧机翼使机鼻下倾，两侧升降翼应该同时升起。相同方法检查另一侧。

陀螺仪（可选）：持机鼻快速倾斜斜飞行器，升降翼应该反向移动。

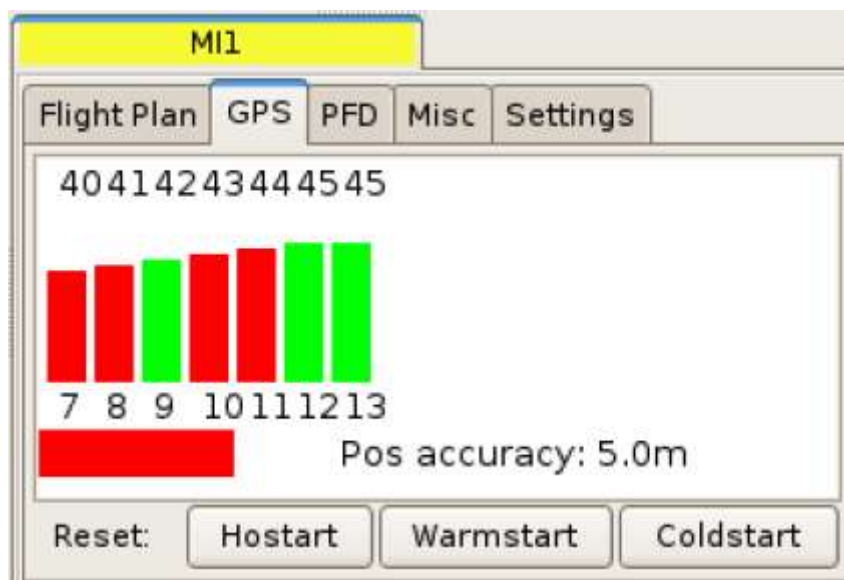




Figure 1.5

1.2.3. 起飞

1. 检查飞行时间（在 Figure 1.4 的 Stripe 区域中）。如果值不是空，点击重置。（需要确认）。
2. 移动 CLIMB 路径点到需要的起飞方向。转到 Takeoff  区域。
3. 在 RC 遥控器上，转到 AUTO1 或者 AUTO2 模式：
 - 在 AUTO1 模式起飞：
 - (a) 半开升降舵。
 - (b) 油门全开。等到引擎全速运转（启动延迟效应消失），抛出。
 - (c) 检查左右转向操作。
 - (d) 转到 AUTO2 模式。
 - 在 AUTO2 模式起飞
 - (a) 在 GCS 中启动油门，（launch, ）。
 - (b) 等到引擎全速运转（启动延迟效应消失），抛出。
4. 在初始的油门全开爬升阶段检查电池电量。
当到达一个安全高度后，航线进入 Standby 飞行区域。

1.2.4. 巡航

在每次飞行任务中，GCS 的原则是：既保证飞行器本身能安全飞行，又监控其遵循既定飞行航线。

检查清单

以下参数在飞行过程中必须持续监控：

- **Datalink 状态：**在 Figure 1.4 中的指示灯变成红色，并对上次接收到电池信息以来的时间以 s 为单位计数。如果连接丢失，操作员可以尝试在 Paparazzi Center 中重启 Data Link 代理，或者通过插拔重启 modem，并重启 Data Link 代理。
- **飞行时间与电池电量：**如果电量持续走低，则必须尽快降落飞行器。
- **高度，AGL 和倾角：**除非是在没有树木的地域飞行，推荐飞行高度不得低于 50m。请注意，在当前地点的 terrain 模式 (/data/srtm/目录中) 不可用的情况下，0m 是指地面高度。
- **速度与油门：**会显示对飞行速度的估计值。
- **活动航线区：**检查当前航线区域是预设的区域。当航线区域切换时，会有信息显示在 console 中。
- **目标高度：**确认目标高度适用于地表高度和指派的飞行空域。
- **Console：**随时读取信息










Figure 1.6

飞行任务

在飞行过程中，操作员可以通过以下手段改变飞行任务：

- **移动 Waypoint：**点击并拖动路径点。对话框 Figure 1.6 用于确认改动。
- **改变高度：**通过 按钮每次升降 5m，或者 每次上升 30m。
- **左右移动 ()：**每次航线移动 5m。可用于直线飞行（飞行轨迹不再依据原定航线）和绕圈飞行（飞行圆半径被修改）。此设置每个飞行段初始化时被重置。
- **进入下一区域飞行模式：**
 - **Standby：**围绕 STDBY 飞行。方向可以通过环绕箭头 改变。飞行半径可以在 Setting/nav 页面中修改（nav_radius 参数）。
 - **Eight figure：**飞经路径点 1，然后环绕路径点 2 飞行。当该区域为活动区域

时,路径点 2 自动移近路径点 1,以免飞行路径图案过大。环绕半径由 `nav_radius` 控制。

-  **Hippodrome**: 沿路径点 1 和 2 的椭圆路径飞行。   改变环绕方向。
-  **环绕 (救生)**: 通过当前的 A/C location 设置 MOB 路径点。无人机进行环绕飞行。
-  **Line**: 在路径点 1 和 2 之间飞行。转向时使用平滑的 U 形转向圆弧。这个区域收到保护,不会受到数据连接丢失的影响: 如果飞行器超过 22s 未接收到 GCS 的任何信息,航线转到 Standby 区域。注意,地面站以 10s 为周期向飞行器至少发送一条信息 (风速估计信息)。
-  **Survey**: 在路径点 S1 和 S2 确定的矩形区域内,进行南北向的扫描飞行。S1 和 S2 可以随时移动。
-  **Landing**: 飞出航迹以完成从路径点 AF 到 TD 的滑行。路径点 BASELEG 被自动放置,来实现绕圈飞行并抵达 AF。

  用于绕圈,8 字形和椭圆形飞行的方向改变。


HOME 模式


HOME 模式是一种错误保护模式。当正常飞航线被中止时,飞行器以一个安全高度围绕 HOME 路径点飞行。这种模式被以下事件触发:

- 在 MANUAL 或者 AUTO1 模式时,飞行器丢失了 RC 的上行连接。
- 距离 HOME 路径点的距离超出飞行计划中设置的阈值 (在 GCS 地图中用一个圆圈表示)。

要退出该模式返回到 AUTO2 模式,点击  指示器。

Kill Throttle

Kill Throttle 状态阻止自动驾驶仪启动引擎。这是飞行方案初始化时的状态,并在进入起飞区域  时重置。它在油门指示上用红色表示。该状态被以下事件触发:

- 电池电量下降到灾难性水平 (在 `airframe` 文件中定义)。
- 飞行器飞出预定义的 HOME 阈值 1.5 倍。
- 操作员触发  (需要确认操作)。

该状态可以使用  重置。

Plotting Data

可以在飞行过程中实时对数据绘图。

在 Paparazzi Center 的 Tools 菜单中, 启动一个报文代理。这个代理将显示所有飞行器送出的报文。如 Figure 1.7 所示。选择 BAT 报文。



Figure 1.7

在 Paparazzi Center 的 Tools 菜单中, 启动一个 Real-time plotter 代理(如 Figure 1.8 所示)。拖动 Messages 窗口的 BAT 报文的 Voltage 数据域到 Real-time plotter 代理中。一条数据曲线被绘制出来并周期性地向左滚动。

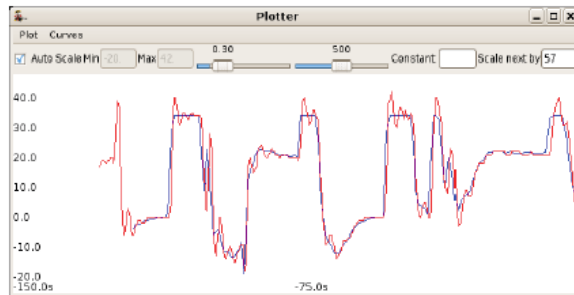





Figure 1.8

1.2.5. 着陆

出于较低的高度精度和下降斜率, 全自动的降落需要一个很大的平坦场地。即使在手动降落方式下, 自动驾驶仪计算和跟踪的降落过程会对初学的 RC 操作员有极大的帮助。

1. 在预定场地上设定 TD 路径点为地面高度用于降落。
2. 在顺风方向设定 AF 路径点(风向估计显示在 2D 地图的左上角), 该点距 TD 约 250m 远, 30m 高 (相对 TD 点)。
3. 飞往右侧 (若最后的转向是右转 ) 或者左侧 (若最后的转向是左转 ) 降落区域。
4. 最终决策:
 - 降落于 AUTO2 模式: 降落后, 在 GCS 中关闭油门 .
 - 在降落前转换到 AUTO1 模式: 控制降落, 然后在 RC 中关闭油门。
 - 绕飞: 如果无法降落, 飞回到 Standby 区域。

1.2.6. 着陆后

1. 卸下电池
2. 停止 GCS 对话（在 Paparazzi Center 中点击 Stop/Remove Processes），退出 Paparazzi Center。

1.3. 飞行后数据分析

1. 检查飞行器在降落时有否损坏。
2. 如果需要，保存日志文件的备份。（var/logs/目录中的.log 和.data 文件）

1.3.1. 数据分析

飞行中收到的所有数传报文数据都可以被存储和分析。

飞行回放

在 Paparazzi Center 的 Tools 菜单，开始一个 Replay 对话。在 Replay 代理中，打开 log 文件，开始重放（菜单操作，如 Figure 1.9 所示）。重放可以在时间线的任意位置暂停或者重新开始

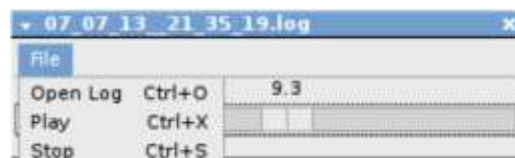


Figure 1.9

所有记录的数传报文数据可以象在模拟或者真实飞行中一样，被同步产生并被所有代理使用（GCS， 报文等等）。当然，GCS 操作员可以在回放中仅进行观察而不做任何修改。

数据绘制

Log Plotter 代理（在 Paparazzi 的 Tools 菜单中启动）使用 log 文件中记录的数据，提供与实时绘图当中一样的功能。

打开一个 log 文件后，在 log 菜单中可以选择某一个报文域（如 Figure 1.10 所示）。轨迹可以被输出到一个 KML 文件并在 Google Earth 中打开。

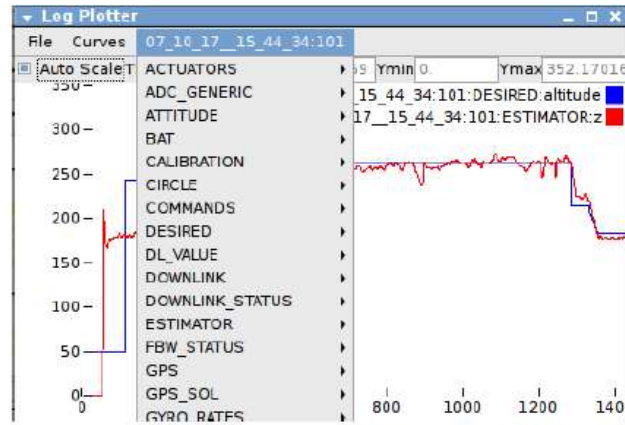


Figure 1.10

Chapter 2. 参考手册

系统的部分特性没有在此文档中描述，它们可以在系统的文档页面中访问：
paparazzi.enac.fr

2.1. Paparazzi Center

Paparazzi Center 是用于 Paparazze 系统的初始图形界面。它用于：

- 将机体配置文件、飞行计划文件、设置文件整合为（能完成飞行的）一件飞行器。
- 编译代码和模拟器。
- 运行一个模拟、飞行或回放的对话。

2.1.1. 飞行器

一个飞行器被定义为以下元素的集合：名字、数字标识，GCS 窗口中的颜色以及一组 XML 文件：

- Airframe 文件（参见 2.4），包括所有硬件设备（伺服机构，传感器等）以及自动驾驶仪的调整（控制环路增益，**中性值**，等）。
- Flight plan，包含了飞行任务的路径点集合和航线区域集合。例如：basic.xml，versatile.xml。
- Setting 文件，包含了在飞行过程中可以调整的参数集合。几个 XML 文件可以同时选择。例如：basic.xml，tuning.xml（用于扩展调整），light.xml（**光照持续时间设置**），switch.xml（**用于视频电源切换**）。
- Radio 文件，描述了 RC 收发器的参数（频道数量和名称，PPM 信号范围）。
- Telemetry 文件，列出了飞行器周期性发送的信息。仅熟练用户和开发用户需要除了 default.xml 以外的信息。

在以上文件的界面上可以载入一个编辑器。提供一个图形界面用于编辑飞行文件。（参见 2.3）

2.1.2. 编译和烧写

控制嵌入式自动驾驶仪的（Target 目标为 ap）和模拟器（Target 目标为 sim）的代码需要通过配置文件生成。在组合框中选择合适的编译目标（只有熟练用户才需要其他 new 编译目标）。在编译期，控制台会输出编译信息的细节。编译错误通常会被定位并高亮显示。

通过 Upload 按钮来烧写编译生成的代码。飞行器必须先通过 USB 电缆连接用于烧写的电脑后启动。

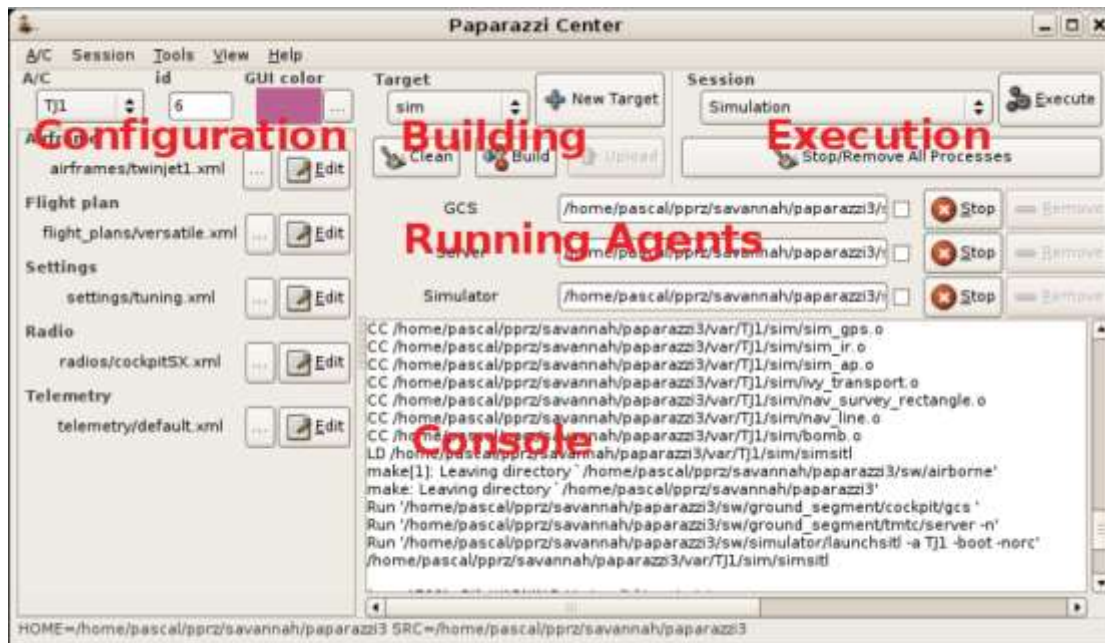


Figure 2.1

2.1.3. 模拟，飞行和回放

系统通过一个由多个相互通讯的代理组成的会话来完成运行。可以使用预定义的会话或者用户自行创建新会话。所有的会话都在组合框内显示。

运行的代理在 Paparazzi Center 中列出，可以停止和重启。自动复位标志会将停止的代理立即重启。

Simulation 会话启动一个服务器，一个 GCS 界面和选定飞行器的模拟器。它要求该飞行器的 sim 目标已经被成功编译。

一个 flight 会话至少包含一个 Modem Link 代理，一个服务器和一个 GCS。

2.1.4. 其他代理

Paparazzi 代理可以在 Tools 菜单中启动。其中有一些需要额外的选项来工作。

飞行计划编辑器：它通过一个飞行计划树（参见 2.3）和一个配置了 2D 地图的 GCS 来运行。编辑项可以从 Edit 菜单运行。

模拟器：它要求在待模拟的飞行器的名字后面加入 -a 选项来运行：

- -boot：飞行器自动启动。
- -launch：飞行器自动起飞。
- -noar：不显示 RC 收发器的图形界面。

数据连接：它驱动一个 Modem。其他重要参数：

- -d<port> 连接 Modem 的串口设备（通常是 /dev/ttyUSB0）。
- -s<baudrate> 波特率（通常是 9600 或者 57600）。
- -transport<mode> 如果某些飞行器使用的是 Maxstream Modem，则必须将其设为 xbee。

报文：电传报文调试（参见 Figure 1.7）

环境模拟器：它允许在模拟中改变风速，时间缩放尺度和 GPS 状态（该代理过去命名为 Gaia）。

实时数据绘制器：将飞行或者模拟中受到的电报报文数据绘制成图表。

日志数据绘制器：将日志文件中的电报报文数据绘制成图表。

2.2. 地面控制站 GCS

GCS（如 Figure 1.3 所示）是 Paparazzi 系统的主图形界面。它显示了飞行、模拟或者回放过程中一个或几个飞行器的关键参数。它同时允许操作员通过移动路径点，改变航线模式或者设置自动驾驶仪的参数来和 UAV 进行交互。

信息显示窗口的 4 个区域中：一个 2D 地图，一个（操作）控制块 Strips，一个（飞行）手册 Notebook，一个（消息）控制台 Console。（请参见 Figure 1.3 确定区域的名称。）

2.2.1. 控制块 Strips

每个 A/C 都有一个与之相应的控制块用于显示该 A/C 的信息并提供常见命令按钮。当同时监控多个飞行器时，只有一个处于激活状态。对于其他非激活飞行器，航线按钮为灰色，其工具条被隐藏。点击控制块（不是按钮！）来激活对应的飞行器。控制块的说明在图 Figure 1.4 中。



飞行参数


主要飞行参数显示于控制块的左侧。用颜色来区分不同指标：颜色越绿越好。如果 kill throttle 被设置为 on，则油门 Throttle 为红色。如果经过预定时间 5s 后未收到电池电量信息，则数据连接 datalink 变为红色。速度 speed 提示信息显示速度估计值。电池 battery 和地面高度 AGL 显示之前一小段时间的历史信息。AGL 显示了用箭头表示方向的垂直速度。飞行时间 Flight time 和航线模式 Navigation mode 可用鼠标触发：鼠标点击 Flight time 会重置，点击 mode 会重置到 AUTO2 模式（可用于退出 HOME 模式）。

航线参数




中间区域重组显示了航线状态。当前航线区域的名称显示在顶部。从当前区域飞行开始经过的时间，从当前航段（航段 stage 定义参见 2.3.7）开始经过的时间，预计到达下一路径点的时间 ETA（处于 go 航段时）。底部显示了当前高度，目标高度，和两者之差（左侧，AGL 下方）。

航线控制

右侧面板是 9 个按钮的集合。第 1 行控制油门。启动  用于起飞阶段。终止  用于设置

kill throttle 状态（在确认后。参见 1.2.4），该状态组织自动驾驶仪启动引擎。重设  将重置 kill throttle 状态。

第 2 行控制航线目标高度。每次+5m ()，-5m ()，或者+30m ()。

第 3 行控制水平航线。每次左移 5m ()，右移 5m ()，或者居中回到初始航迹 ()。

工具条

工具条是一个定义在飞行计划（参见 2.3.1）中和设置文件中的定制按钮集合。因此可以被这两个文件修改。在 basic.xml 的飞行计划和设置文件中定义的按钮的细节请参见 1.2.4。

2.2.2. 地图 Map

显示的 2D 地图包含下列信息：

- A/C 轨迹：可以通过 A/C 菜单的 Clear track 项擦除轨迹。
- A/C 标签包括 A/C 名字，高度和地速。默认选项为关闭。可以通过 A/C 菜单的 A/C label 项激活显示。
- 导向块（黄色三角）：这是 A/C 自主飞行时方向跟随点。
- 路径点在 flight plan 中定义（菱形块）。
- 飞行轨迹用绿色显示。
- 默认背景色为黑色。Google 地图块或者用户地图可以加载用于航线参考（通过 map 菜单设置）。
- 照相机足迹（灰色多边形）表示当前被记载照相机拍摄到的地表带状区域。默认选项为关闭。可以通过 A/C 菜单的 Cam footprint 项激活。
- 鼠标位置的 WGS84 格式坐标显示在右上角。
- 一个公制的通用横轴墨卡托 UTM 坐标网格可以叠加在背景上。通过 Nav 菜单的 UTM grid 项激活。
- Nav 菜单的 SRTM 项可以在靠近位置坐标信息的右上角显示鼠标位置的当前地表高度。SRTM 数据文件（.hgt.zip 或者 .hgt.bz2 文件）需要拷贝到 data/srtm 目录，可以从 <ftp://e0srp01u.ecs.nasa.gov/srtm/version2/srtm3> 下载。

航线

你可以通过以下方法移动/缩放地图：

通过地图上的蓝色箭头按钮或者键盘上的方向键移动。

通过鼠标滚轮，键盘上的 page up/down 或者地图右上方显示了缩放比例的 up/down 按钮来进行缩放。

为了显示所有的 A/C 和路径点，通过 f 快捷键或者 Nav 菜单的 Fit 项来调整地图适应窗口大小。

通过 A/C 菜单的 Center A/C 项来将当前 A/C 在地图居中。

Google 地图块

可以使用从 Google 服务器下载相应的地图照片来自动填充默认的黑色背景。

路径点编辑

当前加载的飞行方案中的任一路径点可以通过以下 2 中方法修改：

- 拖拽路径点到一个新的地点（会出现一个确认对话框）。
- 左键单击某一路径点会打开一个对话框用于编辑路径点的坐标和高度。

路径点修改在经过对话框确认后被立即送往飞行机。GCS 会反复发送修改数据，同时闪烁显示该路径点，直到收到飞行器发回确认信息为止。飞行中不能添加新的路径点。

2.2.3. 飞行手册 Notebook

手册框架为每一个运行的飞行器准备了一个显示页面。每一个页面被分成若干子页面用于显示电传报文数据以及微调自动驾驶仪参数。

当前选中的飞行器属于活动飞行器（参见 2.2.1）

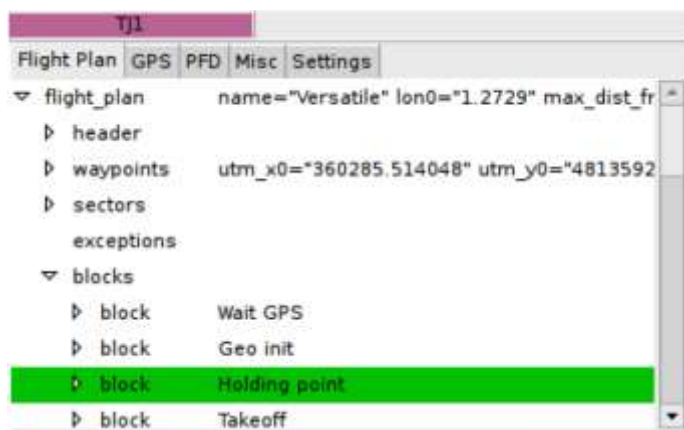


Figure 2.2

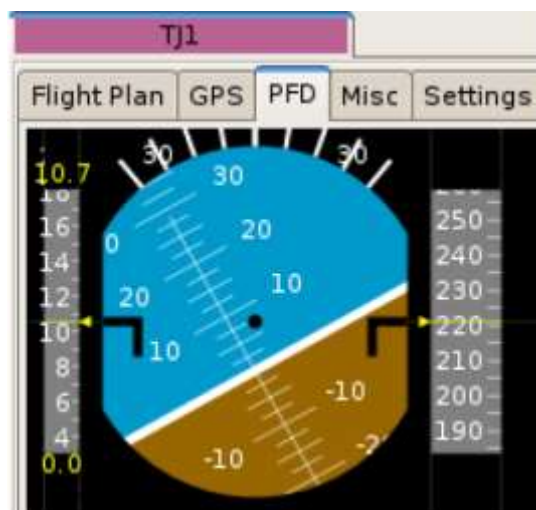


Figure 2.3

飞行计划

Figure 2.2 显示了飞行计划的详细信息。活动区域和航段被高亮显示。在某一区域双击会将航线切换到该区域。

GPS

GPS 页显示了收到的卫星信号强度和估计的位置精度（如 Figure 1.5 所示），并提供了重置 GPS 接收器的手段。

PFD

PFD 页面显示飞行器的高度，地面速度。最小（/最大）速度显示在速度指示的下方（/上方）（通过鼠标单击重置这些值）。

设置

setting 页允许操作员在飞行中改变参数值。页面是按照 settings.xml 文件中的 dl_settings 段来布局的。每个段和字段和一个制表位联系在一起。从左到右每一行依次显示参数名，当前值（由 A/C 周期性送回），一个用于用户输入的滑块或者单选按钮，一个提交/撤销按钮。

控制块中包含的按钮（参见 Figure 1.4）与主控二进制设置关联。

2.2.4. 控制台 Console

在字符界面控制台显示航线块的改变和警告信息。

2.3. 飞行计划

飞行计划文件的正式描述在 DTD 中给出。这个 DTD 必须在飞行计划的 XML 文件头部使用以下语句行加以引用：

```
<!DOCTYPE flight_plan SYSTEM "flight_plan.dtd">
```

飞行计划文件保存在 /conf/flight_plans 文件夹中。飞行计划编辑器可用于在 GUI 下创建基本飞行计划。

2.3.1. Flight Plan 文件结构

从 DTD 文件中析取：

```
<!ELEMENT flight_plan ( header?, waypoints, sectors? include*, exceptions?, blocks)>
```

一个飞行计划有 2 个必备元素：（飞行）路径点 waypoints 和（飞行）区域 blocks，典型情况下还包含可选的 include's 和 exceptions。

根 flight_plan 元素由以下属性描述：

```
<flight_plan name lat0 lon0 ground_alt security_height alt max_dist_from_home>
```

name: 任务名字（字符串）。

lat0, lon0: 定义了 WGS84 格式下的点（0, 0）的坐标度数。

ground_alt: 地面海拔（以 m 为单位）。

security_height: 用于围绕 HOME 点做失效保护飞行的海拔高度。

alt: 路径点的缺省海拔高度。

max_dist_from_home: 距离 HOME 点的最大允许距离（以 m 为单位）。

下面是一个飞行计划第 1 行的例子：

```
<flight_plan      name="Example      Muret"      lat0="43.46223"      lon0="1.27289"
max_dist_from_home="300" alt="250" ground_alt="185" security_height="25">
```

2.3.2. Waypoints 文件结构

路径点是用来定义航迹的地理坐标点。一个路径使用名字和相对坐标来描述：

```
<waypoint name x y [alt]/>
```

这里的 x 和 y 是相对于点（0, 0）的坐标（以 m 为单位）。alt 项是可选项，当该点的海拔高度与全局的 alt 高度不同时设置。记住一个名为 HOME，用于失效保护模式的坐标点不可少的。

示例：

```
<waypoints>
```

```
  <waypoints name="home" x="0" y="0"/>
```

```
  <waypoints name="1" x="-100" y="60" alt="270"/>
```

```
  <waypoints name="2" x="-130" y="217" alt="300"/>
```

```
</waypoints>
```

路径点可以在飞行计划编辑器（参见 2.3.8）中方便地编辑。以下划线开始的路径名除了在编辑模式下是不可见的。

2.3.3. Sector 块

将一个路径点列表作为拐角坐标，可以描述出一个平面块。这个区域可以在 GCS 上显示。有一个函数可以检测坐标点（通常就是飞行器坐标）是否位于该多边形内。目前此功能要求该多边形是凸的，各拐角按顺时针方向放置。对于一个名为 Sector 的块，构造出来的函数是：

```
bool_t InsideSector(float x, float y);
```

这里的 x 和 y 以东向和北方向为坐标增长方向，单位为 m，以飞行计划中的地理坐标（HOME）为参照点。注意：如果飞行计划经过了动态调整，此区域也将被调整，但是目前在 GCS 中的显示不会被更新。

示例：

```
<sectors>
  <sector name="Muret">
    <corner name="_1"/> <corner name="_2"/>
    <corner name="_3"/> <corner name="_4"/>
  </sector>
</sectors>
```

据此可以写出一个停留在该区域的异常：

```
<exception cond="! InsideMuret(estimator_x, estimator_y)" deroute="Standby"/>
```

2.3.4. Blocks 区域

区域元素是飞机计划中的主要部分：它们描述了任务的不同单元。他们由多个基元构成，分别称为航段和异常。你可以将一个接一个的放置在一起。当一个航段（或者一个飞行区域）完成时，自动驾驶仪自动转到下一个。飞完最后一个区域的最后一个航段的行为未定义。

DTD 中如下定义：blocks 是由多个 block 元素构成，而一个 block 元素又是多个航段构成的序列：

```
<!ELEMENTS blocks (block+)>
<!ELEMENTS block (exception| while| heading| attitude| go| xyz| set| circle| deroute| stay|
follow| survey_rectangle| call)*>
```

示例：

```
<block name="circlehome">
  <circle radius="75" wp="HOME"/>
</block>
```

如果你的系统配置了数据连接，你可以使用属性 strip_button 在飞行器控制块中加入一个按钮：

```
<block name="descent" strip_button="Descent">
...
</block>
```

该按钮可以激活此区域。

可以定义一个图标来显示该按钮，strip_button 标签内容是此图标的工具栏提示文本。这个图标必须是在 data/pictures/gcs_icons 目录下的一个可用图像文件：

```
<block name="Takeoff" strip_icon="takeoff.png" strip_button="Takeoff">
```


2.3.5. Expressions 表达式

几乎所有的航段的数字属性都按照 C 表达式来解析。这里的 C 表达式的语法被限定于：

- 数字常量
- 自动驾驶仪的内部变量（未完全文档化，参见示例程序）
- 2 元运算符：>, <=, >=, <>, ==, +, -, /, *;
- 部分工具函数

下一节中有部分表达式的使用范例。

2.3.6. Exception 异常

飞行管理器可以处理异常。它通过周期性的条件检查（与航向控制步调一致），运行控制跳转一个给定的区域。以下是异常的语法：

```
<exception cond="..." deroute="...">
```

这里 cond 是一个表达式，deroute 是一个我们预设的，一旦表达式为真就立即前往的区域名。

一些异常的范例：

```
<exception cond="10>PowerVoltage()" deroute="go_down"/>
```

```
<exception cond="(groud_alt+10>estimator_z)" deroute="go_up"/>
```

```
<exception cond="(estimator_flight_time>840)" deroute="quick_land">
```

通过<exception>元素，异常可用于一个块的内部，或者飞行计划的全局。在下面的例子中，从控制地面站接受到上一条报文的时间被记录，如果超过 22s 未接受任何报文，则航线转到 Standby 区域。该异常适用于任何飞行区域：

```
<flight_plan...>
```

```
  <waypoints>...</waypoints>
```

```
  <exceptions>
```

```
    <exception cond="datalink_time>22" deroute="Standby"/>
```

```
  </exceptions>
```

```
</blocks>...
```

2.3.7. Navigation Primitives 航线基元

航线状态以 3D 形式描述了目标航迹。水平方向上的状态由航段定义，垂直方向的控制则有航段的若干属性值来定义（即航段与航段之间有不同的水平投影轨迹，而航段内部的高度可以有若干不同值？）。以下是可用的航线航段：

- attitude：保持固定海拔高度飞行；
- heading：保持一个给定航向；
- go：飞往给定路径点；
- circle：围绕给定路径点飞行；
- follow：跟随另一架飞行器；

不同航段的垂直控制通过使用 vmode 属性来完成。可用的值包括：

alt（默认）：自动驾驶仪保持在路径点（如果存在的话）的海拔高度，或者是 alt 参数

设定的海拔高度：

climb: 自动驾驶仪以 climb 参数设定的爬升率（单位 m/s）爬升；

throttle: 自动驾驶仪以 throttle 参数设定值（在 0~1 之间）飞行；

glide: 自动驾驶仪在两个路径点间以设定斜率飞行。

默认设定为 throttle。但是，如果设定 pitch 属性为 auto，throttle 属性为一个常量，则垂直控制仅取决于 A/C 所控制的飞行姿态。当垂直控制模式为 throttle 时，pitch 参数也可以设置为任意值（单位为度）：这通常会影响到飞行器的空速。

不同的航线模式的细节参见下一节。

Circle 绕飞

circle 基元是主要的航线模式：航迹被定义为以指定半径围绕一个给定路径点的圆飞行：

```
<circle wp="home" radius="75"/>
```

一个正的半径意味着 UAV 顺时针飞行，负值相反。

有个 until 参数可用于控制航段的结束。下例定义了一个固定油门大小，机头上升，不断绕圈直到电池电量耗尽的上升航迹：

```
<circle wp="wp1" radius="50+(estimator_z-ground_alt)/2" vmode="throttle" throttle="0.75"
pitch="0.3" until="10>PowerVoltage()"/>
```

Go 飞往

go 有很大可能是最有用的基元。基本上，自动驾驶仪尝试加入一个指定的路径点（wp 是唯一的必须参数）。所以最简单的例子是：

```
<go wp="HOME"/>
```

该指令将 HOME 路径点设置为预设目标位置。注意，因为 vmode="alt" 是默认值，所以目标路径点的海拔高度将会被考虑进去。当到达目的地后，航线将会立即切换到下一个航段。

一般来说，加入一个路径点却不指明一个具体的航迹（比方说一条给定直线）不是一个好的控制方式。将 hmode 设置为 route，航线将会飞经一个连接 2 个路径点的段：

```
<go from="wp1" wp="wp2" hmode="route"/>
```

飞行器的目标海拔高度就是目标路径点的海拔高度，这个也可以通过 alt 参数来修改。

以下是一个用固定油门来保持高度的示例：

```
<go from="wp2" wp="wp3" hmode="route" pitch="auto" throttle="0.75"
alt="ground_alt+100"/>
```

与垂直控制相关的参数也可以用来取代默认的海拔高度模式：

```
<go from="wp1" wp="wp2" hmode="route" vmode="climb" climb="1.5"/>
```

最后，approaching_time（单位 s）接近时间参数用于决定目标何时到达。它可以被设定为 0（直接在目标路径点上方）。默认值是 CARROT 时间，该值在 airframe 配置文件中设定。（就是指飞机与指示光标 CARROT 之间的飞行时间？）

```
<go from="wp1" wp="wp2" hmode="route" approaching_time="1"/>
```

Attitude 飞行姿态

attitude 参数的航线模式对应 horizontal 模式的当前最底层控制回路。通过它，自动驾驶仪保持固定的飞行姿态。roll 参数为必须参数。（单位为度，正值表示降低右翼）。

使用固定空速飞离的示例：

```
<attitude roll="0" vmode="throttle" throttle="0.5"/>
```

保持姿态，绕圈飞行的实例：

```
<attitude roll="30" alt="ground_alt+50"/>
```

注意，因为没有对飞行器的地理位置进行控制，因此该模式不是安全的航线模式。但是它对于调试飞行器横滚控制回路是有用的。

Heading 航向

heading 基元是与自动驾驶仪的 horizontal 模式的第 2 层控制回路相关的必要参数（单位为度，顺时针，北=0，东=90）。通过它，自动驾驶仪保持跟定航向。

一个用于起飞区域，跑道航向，80%油门大小，机头上升直到爬升 30m 的示例：

```
<heading course="QFU" vmode="throttle" throttle="0.8" pitch="0.3" until=
="(estimator_z>ground_alt+30)"/>
```

Deroute 航路

deroute 基元是飞行计划的 goto（飞往）指令。它将航线转换到指定区域：

```
<deroute block="landing"/>
```

注意，该基元不能用于执行下一节给出的循环指令。

Loops 循环

使用 while 基于可用于编写无限次的循环。其 cond 参数为一个布尔表达式。while 语句内部是航段：

```
<while cond="TURE">
  <go wp="A"/>
  <go wp="B"/>
  <go wp="C"/>
  <while cond="5>stage_time"/>
</while>
```

在这个例子中，我们执行一个无限循环：依次飞往路径点 A，B，C，等待 5s 后，重复。

有限循环使用 for：

```
<for var="i" from="0" to="3">
  ...
</for>
```

该循环的循环体语句将会执行 4 次。

for 循环的控制变量可以用在循环语句航段控制的表达式中：

```
< for var="i" from="0" to="5" >
    <circle wp="HOME" radius="75" alt="ground_alt+50*$i" until="stage_time>10"/>
</for>
```

本例中，我们围绕 HOME 路径点以 50m 飞行 10s，100m 飞行 10s...直到飞达 250m。不允许在一个区域中两个有限循环使用同样的循环控制变量。

Follow 跟随

follow 是一个用于控制一台 UAV 以指定距离（单位米）和指定高度差（单位米）跟随另一台 UAV（真实或者模拟的，使用其 ac_id 命名）飞行的特殊基元。

下例中，自动驾驶仪将在后方 50m 差和上方 20m 跟随序号为 4 的 A/C：

```
<follow ac_id="4" distance="50" height="20"/>
```

set

set 是一个危险的基元，仅限于专家用户使用：它直接操作自动驾驶仪的内部变量。例如，你可以修改用于 HOME 模式失效保护过程的默认地面海拔高度（也可以在你自己的飞行计划文件中修改）：

```
<set var="ground_alt" value="ground_alt+50"/>
```

此参数功能过于强大因而很有可能带来灾难性错误，请慎用。

Call 调用

call 允许用户用 C 语言定义自己的航线过程。它的值是一个对布尔函数的调用，该函数当航段未完成时返回 TRUE（一个只被调用一次的函数将会立即返回一个 FALSE）。这个性质用于下面的 line 模式：

```
<call fun="nav_line_init()"/>
<call fun="nav_line(WP_1,WP_2,nav_raduis)"/>
```

本例中 nav_line_init() 返回 FALSE，而 nav_line() 一直返回 TRUE（此航段不会终止）。这样的函数一般定义于一个附加 C 文件中，而且需要在机体文件中说明(在 makefile 段，参见 2.4)。

```
ap.srsc += nav_line.c
```

```
sim.srsc+=nav_line.c
```

这些函数同样需要在头文件中申明，头文件必须在飞行计划的 header 部分包含：

```
<header>
```

```
#include "nav_line.h"
```

```
</header>
```

这些 C 源文件和头文件存放于 sw/airbone 目录中。

2.3.8. Edition 编辑

使用基于图形界面接口的编辑器是一个最方便的编辑飞行计划的方法。从 Paparazze Center 的 Tools 菜单 Flight Plan Editor 项可以启动它（如 Figure 2.4 所示）。通过 Edit 菜单可以打开和关闭一个飞行计划。编辑器的上半部分显示了标准 2D 地图，地图上的路径点可以移动和编辑。可以通过 CTRL 键+鼠标左键在地图上创建一个新的路径点。下半部分用 XML 树结构显示了飞行计划。树的元素可以进行修改：

- 元素可以鼠标右键菜单删除，拷贝，插入。
- 按下鼠标左键，可以进行元素拖放。
- 选中元素的属性在右侧的列中显示。属性值可以编辑修改。通过右键菜单，可以添加某些属性。

注意，这是一个语义编辑器：属性和元素项目必须符合 DTD 规定，才能提交。

飞行计划也可以使用任何文本编辑器编辑，常见的编辑器（例如 gedit）可以识别 XML 文件的语义并提供符合使用者喜好的高亮显示。

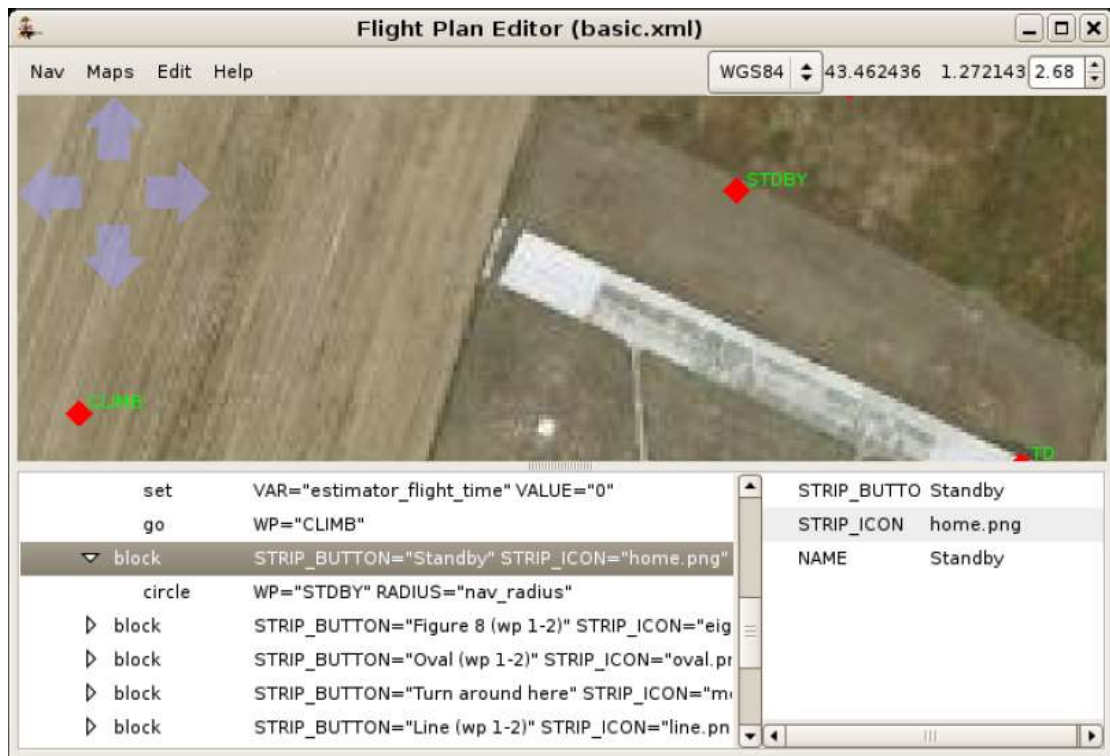


Figure 2.4

2.4. 机体配置文件

机体配置文件存放于 con/airframes 文件夹，保存了一个飞行器所需的所有硬件和软件配置。这是一个包含了 Makefile 底层代码的 XML 文件。所有的增益、配平和形状都使用标准的 XML 元素进行定义。硬件定义，例如处理器型号、modem 协议、舵机驱动等，位于 makefile 的原始段中。该文件可使用任何标准文本编辑器编辑。

文件的完整描述可以从项目的 wiki 文档获得。这里我们仅给出一些经常需要修改的属性。

Servos Neutrals 舵机中性点

舵机的中性点使用 `servos` 元素定义（通常在文件头部）。

```
<servos>
  <servo name="AILEVON_LEFT" no="1" min="2000" neutral="1490" max="1000"/>
  ...
</servos>
```

这里的 `neutral` 必须是 `MANUAL` 模式的中性点配平位。可以从飞行期间 `ACTUATORS` 的电传报文中得到（在 `Messages` 代理中，参见 Figure 1.7）。

Infrared Neutrals 红外中性点

`INFRARED` 段包含了侧滚和航向倾角的中性点。这些中性点通常在 `AUTO1` 模式中用于配平。

```
<section name="INFRARED" prefix="IR_">
  <define name="ROLL_NEUTRAL_DEFAULT" value="5" unit="deg"/>
  <define name="PITCH_NEUTRAL_DEFAULT" value="5" unit="deg"/>
  ...
```

注意：这些值在文件中用度为单位，而在 `GCS` 设置中用弧度显示。所以 `GCS` 的调试结果需要乘以 $180/\pi \approx 57$ 才能在文件中使用。

Complementary C Module 附加 C 模块

`makefile` 段有时需要修改以增加其他的 C 语言模块。例如，趋风搜索策略没有被包括在一个标准的机体文件中。要使用它，可以将 `anemotaxis` 模块加入机载目标（`ap target`）或者模拟器目标（`sim target`）进行编译：

```
ap.srsc+=anemotaxis.c
sim.srsc+=anemotaxis.c
```

2.5. 设置

设置文件的语法在 `con/settings/Settings.dtd` 中定义。它是一个有命名变量构成的树结构。每个变量被设置了 `min`, `max` 和 `step` 属性。这些属性用于在 `GCS` 中创建 `Settings` 图形界面。它的格式：

```
<dl_setting max="2" min="0" step="1" var="pprz_mode">
```

可以添加更多的属性：

- `shortname="s"`：s 将取代 `GCS` 标签中的变量名。
- `module="m"`：它标记了该变量从哪个文件中得到。相应的 `#include "m.h"` 语句将被产生在相应的 C 代码中。
- `handler="h"`：标记了进行设置操作的宏。这个宏通常命名为 `m_h()`。

在控制块(strip)中封装的按钮也可以与变量关联。他们在 `dl_setting` 元素的 `strip_button`

子元素中描述:

```
<dl_setting max="1" min="0.0" step="0.05" var="v_ctl_auto_throttle_cruise_throttle"
shortname="cruise throttle" module="fw_v_ctl" handler="SetCruiseThrottle">
```

```
<strip_button name="Dash" value="1"/>
```

```
<strip_button name="Loiter" value="0.1"/>
```

```
</dl_setting>
```

为了美化控制块，可以为按钮使用图标:

```
<dl_setting max="1" min="0" step="1" var="power_switch" module="autopilot"
handler="SetPowerSwitch">
```

```
<strip_button name="Switch on" icon="on.png" value="1"/>
```

```
<strip_button name="Switch off" icon="off.png" value="0"/>
```

```
</dl_setting>
```

图像文件需要存放在 data/pictures/gcs_icons 目录中。

存放在 conf/settings 目录下的 basic.xml 和 tuning.xml 文件提供了设置文件的诸多范例。

2.6. 调整

Paparazzi 是一个高度可定制化的系统。调整自动驾驶仪，使之适用于另一种全新机体的完整过程，超出了本文档的范围。不过，我们在此将强调一些适用于微调一架已预设置过的飞行器的过程，例如进行一次深度维护，或者是更换一个作动器。

2.6.1. Actuators Layer 作动器层

为了优化性能，飞行器的控制面需要配平以实现在无指令干预下的中立飞行。为了进行飞行器配平，需要执行以下过程。请注意，该过程需要全手动飞行，这需要一个有经验的遥控飞行操作员。

- 确认遥控器没有实施任何配平。
- 让遥控飞行操作员手动控制飞机起飞。
- 让操作员象操作一架传统飞机一样进行配平。使用期望的自动飞行模式下的最小速度进行水平飞行的手动配平。
- 一旦达到满意姿态，降落飞机，保持配平偏移量。
- 载入 Message 代理，查找 ACTUATORS 报文信息。
- 按照 ACTUATOR 报文信息修改在 airframe 配置文件(见 2.4)的 servos 段的 neutral 域。通过使用 servos 段元素的 no 域来建立关联。
- 编译，烧写飞行器程序，**重做一遍手动遥控配平**。

对控制面中性点位置的任何调整有可能影响到自动飞行模式下的行为，因此需要进行相应调整。

2.6.2. Stabilization Layer 稳定层

高度稳定层包括 2 个控制回路:

- roll 横滚回路包括 2 个增益和 1 个中性值。

- **pitch** 航向倾角回路包括 1 个增益和 1 个中性值。

尽管航向倾角回路看起来比较容易调整，但在实际操作过程中并不是这么简单，因为长时间保持固定的航线倾角姿态是很困难的。

Pitch loop 航线倾角回路

航线倾角控制是一个简单的比例回路（参见 Figure 2.5）。

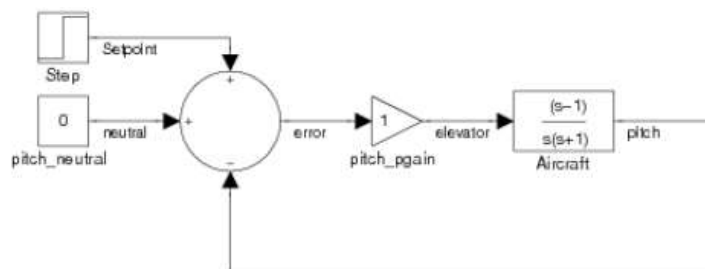


Figure 2.5

- **调整点** `pitch_neutral` **从 0 开始调整**以达到预期的空速。
- 调整 `pitch_pgain` 来得到一个满意的航线倾角步进指令反馈速度。此值过低则可能无法达到调整点（`pitch_neutral?`）的值。过高则会引发振荡。一旦得到了一个合适的增益值，请确保在高速下对其高速飞行下的波动进行测试。对飞行器进行俯冲和失速测试，确保自动驾驶仪能够改出姿态。

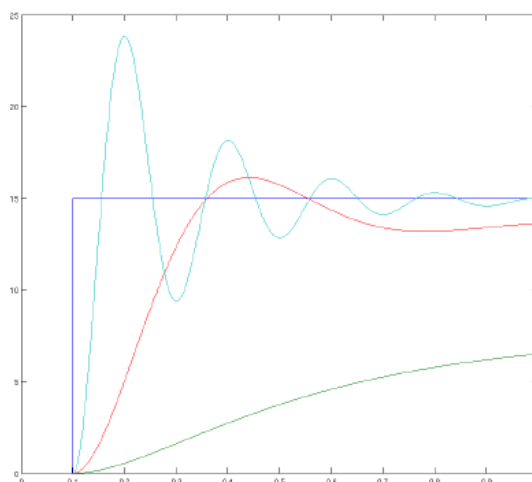


Figure 2.6

注意：为了能够成功的调试中性值，增益参数应该有一个合理有效的值。请小心不要强制设置一个较低增益配合一个较高的中性值，然后增大增益。这样的话可能会导致飞行器执行一个极端的倾角姿态进而导致失速或过速。

Roll loop 横滚回路

横滚控制基于一个稍微复杂些的 P-D 比例微分回路。与 `pitch` 类似，`roll_neutral` 也应该设置为能实现飞行器在无指令干预下的直线飞行。同样，中性点的值也应该是配合一个能实现有效控制的增益值的合理值。

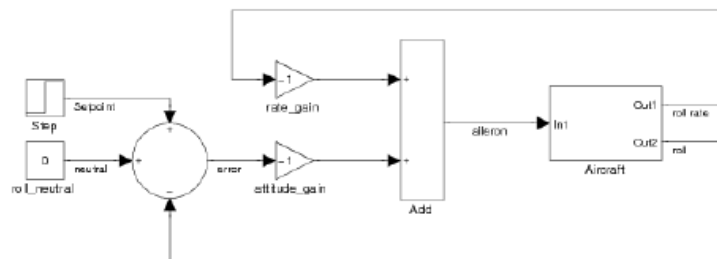


Figure 2.7

横滚控制回路试图完成一对看似矛盾的目标：实现一个给定滚转角，同时保持一个 0 滚转角速度(机身保持固定的倾斜角度，但是不持续翻滚？)。roll attitude pgain 和 roll rate gain 分别控制着两个目标的权重。增大 roll attitude pgain，则飞行器会更快地实现倾斜角，但可能引发过调或者振荡。而增大 roll rate gain，会抑制振荡，但以较长的姿态响应时间为代价。

- 你可以在 AUTO1 模式下用遥控器操纵杆或者 AUTO2 模式下的飞行计划文件来实现这些调整步骤。
- 可以通过目视观察对飞机进行粗略调整，然后使用实时图形数据进行精细调整。
- 你可能会倾向使用一个在控制良好前提下尽可能低的增益值。它可以在鲁棒(强健)性和电池效率上带来好处。
- 调整和飞行姿态的估计的精确性有密切联系。如果你的姿态误差放大了 2 倍，那么 roll attitude pgain 也会放大 2 倍作为补偿。这不是一个好状况，它会对上层的航线控制带来混乱。

2.6.3. Navigation Layer 航线层

Navigation 航线层很大程度上基于一个和 pitch 回路类似的比例回路。一个航线误差(例如航向没有考虑风的影响)会被乘上一个名为 course pgain 的系数，然后送到一个转向调整点。



Figure 2.8

Chapter 3. 附录

缩写、术语对照表

UAV: Unmanned Aerial Vehicle 无人机

RC: Remote Control 遥控

CG: Center of Gravity 重心

AGL: Above Ground Level 离地高度

throttle: 油门、油门

MOB: Man Over Board 救生

TD: Touch Down

A/C: Autopilot/Controller

EAT: Estimate Time of Arrival

Flight: 飞行

Navigation: 航线（少数地方翻译成飞行）

carrot: 导向块

UTM: Universal Transverse Mercator 通用横轴墨卡托(坐标网)

DTD: Document Type Definition 文档类型定义, XML 规范的一部分

waypoints: （飞行）路径点

block: （飞行）区域

trajectory: 航迹（为与 navigation 航线区别）

Primitives: 基元

stage: 航段

QFU: 跑道航向

roll: 横滚

pitch: 航向倾角

P-D loop: Proportional Differential 比例微分回路