

1. 实验名称及目的

1.1. 实验名称

飞控状态数据读取实验

1.2. 实验目的

通过 RflySim 的底层开发接口，可获取的关键状态部分数据。本实验将进行载具姿态、电池、传感器融合、GPS、IMU 的部分数据的获取，以此思路获取其他各种数据。

1.3. 关键知识点

本实验硬件在环仿真数据流：Simulink 控制器算法生成代码下载到飞控中，然后用 USB 实体信号线替代 Simulink 中的虚拟信号线。CopterSim 将传感器数据（例如，加速度计、气压计、磁力计等）通过 USB 数据线发送给飞控系统；飞控系统内的 PX4 自驾仪软件将收到传感器数据进行滤波和状态估计，将估计的状态信息通过内部的 uORB 消息总线发送给控制器；控制器再通过 USB 数据线将“rfly_px4”的 uORB 消息回给 CopterSim，CopterSim 通过 UDP 的方式以 40100 系列端口将该数据转发出去，启动 Simulink 的 PX4ExtMsgSender.slx 即可实时监听到所订阅的各种传感器信息，从而形成一个硬件在环仿真闭环。

本实验实飞数据流：在 Simulink 中搭建各种传感器数据的获取文件，该文件内部均是一 uORB 消息订阅的方式订阅各种传感器的数据；自动代码生成的过程中会在在 PX4 软件系统中生成名为 px4_simulink_app 的模块应用，通过烧录到飞控中与其他模块形成并行运行。在 QGC 中通过监听“rfly_px4”的 uORB 消息即可接收所自定义的消息。

2. 实验效果

实现飞控中关键状态部分数据的读取。

3. 文件目录

文件夹/文件名称	说明
Exp1_VehicleAttDataGet.slx	载具姿态数据获取文件。
Exp2_BatteryMesDataGet.slx	电池数据获取文件。
Exp3_SensorCombDataGet.slx	传感器融合数据获取文件。
Exp4_VehicleGPSDataGet.slx	GPS 数据获取文件。
Exp5_IMUDataGet.slx	IMU 获取文件。
PX4ExtMsgSender.slx	飞控 uORB 消息监听程序

4. 运行环境

序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1

2	RflySim 工具链	Pixhawk 6X 或 Pixhawk 6X mini ^②	1
		遥控器 ^③	1
		遥控器接收器	1
		数据线、杜邦线等	若干

①：推荐配置请见：<https://doc.rflysim.com>

②：须保证平台安装时的编译命令为：px4_fmu-v6x_default，固件版本为：1.12.3。其他配套飞控请见：<http://doc.rflysim.com>

③：本实验演示所使用的遥控器为：天地飞 ET10、配套接收器为：WFLY RF209S。遥控器相关配置见：<https://rflysim.com/doc/zh/B/3.1ET10.html>

5. 实验步骤

5.1. Exp1_VehicleAttDataGet.slx—载具姿态数据获取文件

多旋翼载具的姿态数据通常包括以下几个重要的参数：

姿态角（Euler 角）：

1、滚转角（Roll）：载具绕其前进方向的轴旋转的角度。

2、俯仰角（Pitch）：载具绕其侧向轴旋转的角度。

3、偏航角（Yaw）：载具绕其垂直轴旋转的角度。

四元数（Quaternions）：四元数是一种用于描述旋转的数学工具，通常用四个实数表示。

角速度（Angular Rates）：载具绕各个轴的旋转速率。

加速度（Acceleration）：载具在各个轴上的加速度。

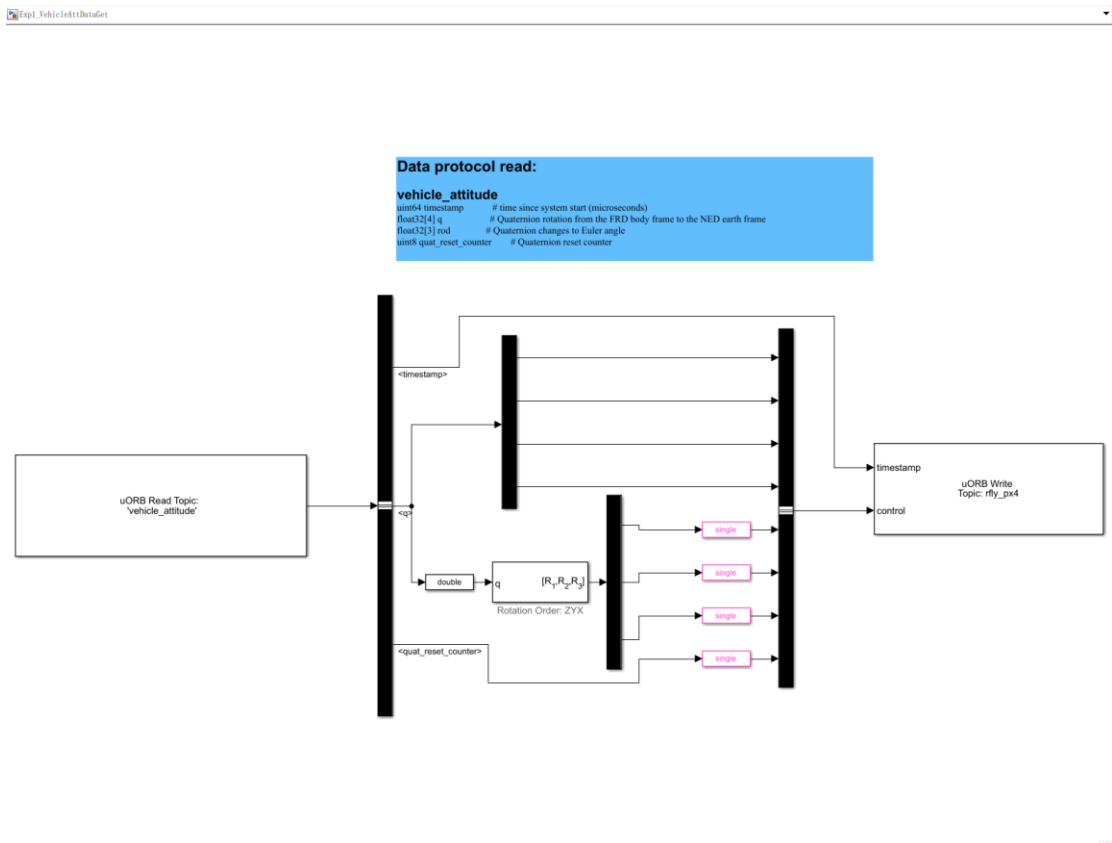
角加速度（Angular Acceleration）：载具在各个轴上的角加速度。

这些姿态数据可以通过惯性测量单元（Inertial Measurement Unit, IMU）或其他惯性传感器来获取。IMU 通常包括陀螺仪（Gyroscope）用于测量角速度，加速度计（Accelerometer）用于测量加速度，有时也包括磁力计（Magnetometer）用于测量地磁场以辅助定向。

在实际的飞行控制系统中，这些姿态数据通常以数字方式进行传输和处理，以便控制算法对载具的姿态进行精确的控制。

如下图所示，打开[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\11.StateDataGatAPI](#)中的 Exp1_VehicleAttDataGet.slx 文件。该系统主要由 uORB Read and Function-Call Trigger 和 uORB Write Advanced 模块构成。

uORB Read and Function-Call Trigger 和 uORB Write Advanced 具体用法和功能请参考：[API.pdf](#)



此外，该模型也用到了 Data Type Conversion（数据类型转换）模块。

如下图所示，Data Type Conversion（数据类型转换）模块用于将输入信号的数据类型转换为所需的输出数据类型。这个模块在处理不同数据类型之间的转换时非常有用，例如将整数转换为浮点数，或者将固定点数转换为双精度浮点数等。

主要特点和用法包括：

数据类型转换：Data Type Conversion 模块可以将输入信号的数据类型转换为指定的输出数据类型。常见的数据类型包括整数、浮点数、固定点数等。

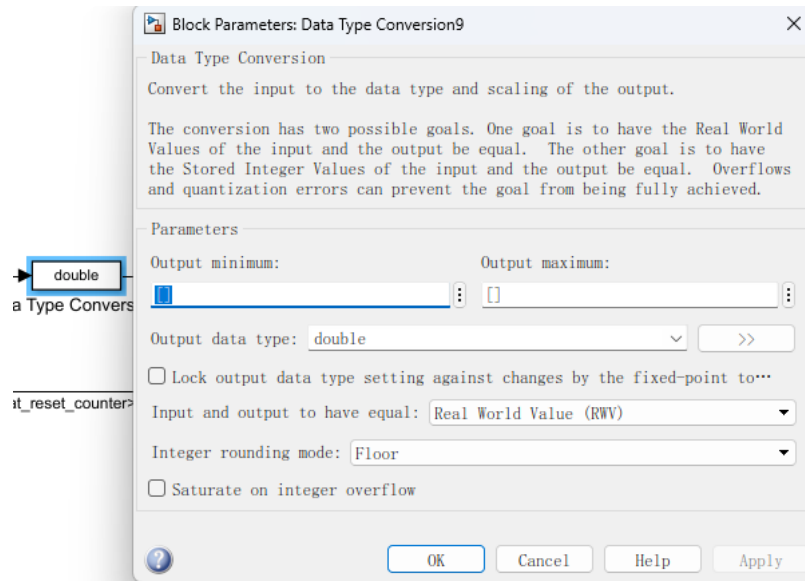
精度控制：通过 Data Type Conversion 模块，可以控制信号的精度，例如从高精度转换到低精度，或者从低精度转换到高精度。这对于在系统中平衡性能和计算成本非常有用。

饱和和截断：在进行数据类型转换时，可以选择是否进行饱和或截断操作。饱和操作会将超出目标数据类型范围的值限制在目标范围内，而截断操作则会直接舍弃超出目标数据类型范围的部分。

舍入模式：在进行浮点数或固定点数转换时，可以选择不同的舍入模式，包括向上舍入、向下舍入、向零舍入等。这有助于控制在转换过程中的舍入误差。

输出端口数目：Data Type Conversion 模块可以有一个或多个输出端口，取决于输入信号的数量和转换类型。你可以根据需要选择合适的输出端口。

通过 Data Type Conversion 模块，可以灵活地控制信号的数据类型和精度，以满足系统建模和仿真的需求。这在处理不同数据类型的信号、接口以及在不同精度下进行计算时非常有用。



5.2. Exp2_BatteryMesDataGet.slx—电池数据获取文件

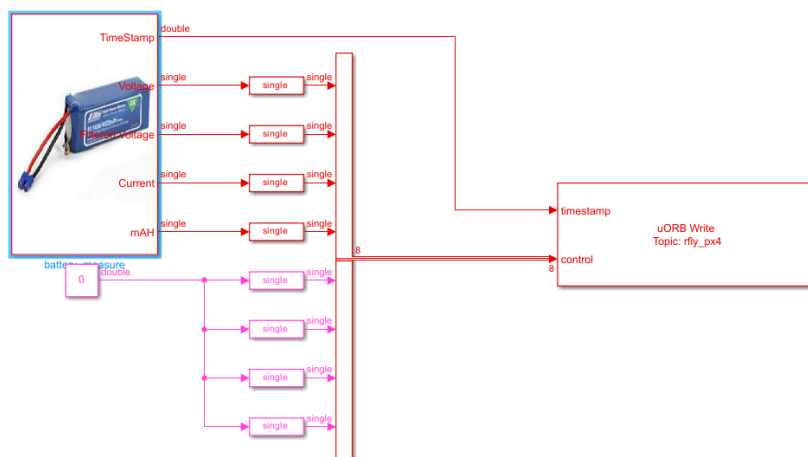
1. Timestamp: (int32)测量的时间戳。
2. Voltage: (double)电池电压，单位为伏特。
3. Filtered Voltage: (single) 过滤电池电压，单位为伏特。
4. Current: (single)电池电流，以安培为单位。
5. mAH: (single)以 mAH 为单位的放电量。

这些数据对于多旋翼飞行器的设计、运行和维护都非常重要，可以更好地管理电池，确保飞行安全和性能稳定。

如下图所示，打开[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\11.StateDataGatAPI](#) 中的 Exp2_BatteryMesDataGet.slx 文件。该系统主要由 uORB Read and Function-Call Trigger 和 uORB Write Advanced 模块构成。

uORB Read and Function-Call Trigger 和 uORB Write Advanced 具体用法和功能请参考：[API.pdf](#)

该模型也用到了 Data Type Conversion（数据类型转换）模块，关于 Data Type Conversion（数据类型转换）模块的详细介绍请参考：[Exp1_VehicleAttDataGet.slx—载具姿态数据获取文件](#)



5.3. Exp3_SensorCombDataGet.slx—传感器融合数据获取文件

多旋翼传感器融合数据指的是通过将多个传感器的数据进行融合，以提高对多旋翼飞行器姿态、位置等信息的准确性和稳定性所得到的数据。

常见的传感器包括：

- 1、加速度计（Accelerometer）：用于测量载具的加速度，从而推断载具的姿态和运动状态。
- 2、陀螺仪（Gyroscope）：用于测量载具的角速度，从而推断载具的旋转状态。
- 3、磁力计（Magnetometer）：用于检测地磁场，从而帮助确定载具的方向。
- 4、气压计（Barometer）：用于测量大气压强，从而推断载具的高度。
- 5、GPS 接收器（GPS Receiver）：用于获取卫星信号，从而确定载具的全球位置。

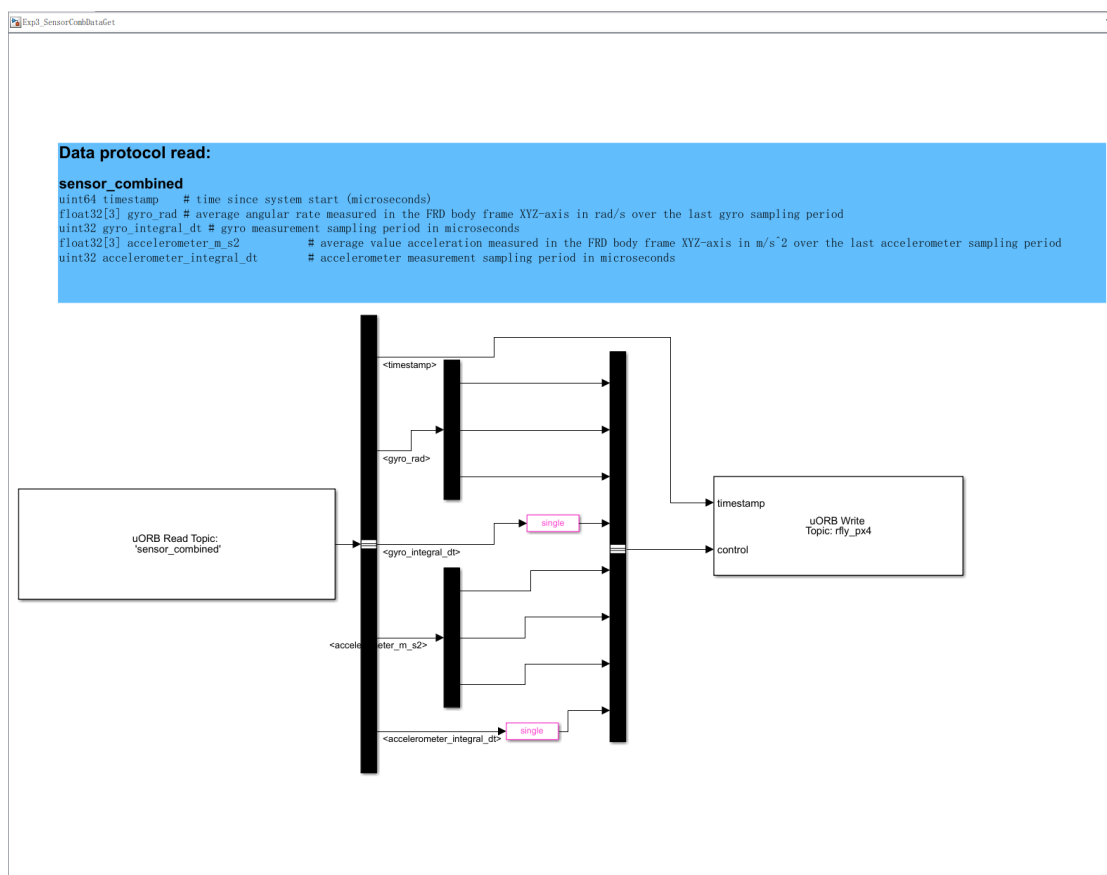
传感器融合算法将从这些传感器中获取的数据进行处理和整合，以提供更加准确和可靠的姿态、位置和运动信息。传感器融合可以采用各种算法，包括卡尔曼滤波器、扩展卡尔曼滤波器（EKF）、粒子滤波器等。这些算法能够有效地处理传感器数据中的噪声、漂移等问题，从而提高飞行器的定位和导航性能。因此，多旋翼传感器融合数据是经过传感器数据融合算法处理后得到的、更加准确和可靠的飞行器姿态、位置和运动信息。

如下图所示，打开[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\11.StateDataGatAPI](#) 中的 Exp3_SensorCombDataGet.slx 文件。该系统主要由 uORB Read and Function-Call Trigger 和 uORB Write Advanced 模块构成。

uORB Read and Function-Call Trigger 和 uORB Write Advanced 具体用法和功能请参考：

[API.pdf](#)

该模型也用到了 Data Type Conversion（数据类型转换）模块，关于 Data Type Conversion（数据类型转换）模块的详细介绍请参考：[Exp1_VehicleAttDataGet.slx—载具姿态数据获取文件](#)



5.4. Exp4_VehicleGPSDataGet.slx—GPS 数据获取文件

多旋翼 GPS 数据通常包括以下信息：

- 1、位置信息：经度和纬度，表示飞行器当前所处的地理位置。
- 2、海拔高度：飞行器相对于海平面的高度。
- 3、速度信息：飞行器的地速和对地速度。
- 4、航向：飞行器当前的航向角，即其朝向的方向。
- 5、卫星信息：可见卫星数、卫星信号强度等信息，用于判断 GPS 定位的质量和稳定性。

这些 GPS 数据对于多旋翼飞行器的导航、定位和航迹规划至关重要。通过处理 GPS 数据，飞行控制系统可以实现自主飞行、航线跟踪和自动返航等功能。

如下图所示，打开[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\11.StateDataGatAPI](#)中的 `Exp4_VehicleGPSDataGet.slx` 文件。该系统主要由 uORB Read and Function-Call Trigger 和 uORB Write Advanced 模块构成。

uORB Read and Function-Call Trigger 和 uORB Write Advanced 具体用法和功能请参考：

[API.pdf](#)

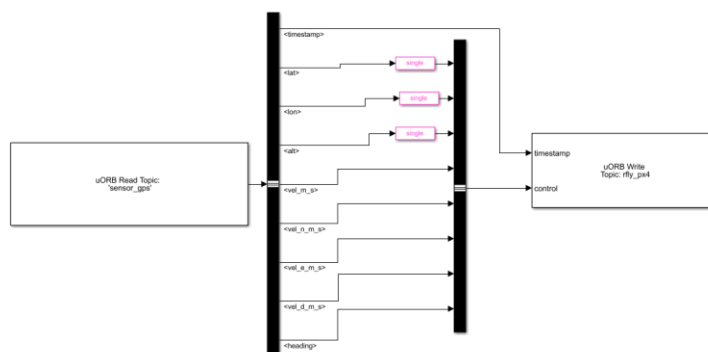
该模型也用到了 Data Type Conversion（数据类型转换）模块，关于 Data Type Conversion（数据类型转换）模块的详细介绍请参考：[Exp1_VehicleAttDataGet.slx—载具姿态数据获取文件](#)

Data protocol read:

```

sensor_gps
uint64 timestamp # time since system start (microseconds)
int32 lat # Latitude in 1E-7 degrees
int32 lon # Longitude in 1E-7 degrees
int32 alt # Altitude in 1E-3 meters above MSL, (millimetres)
float32 vel_n_m_s # GPS ground speed, (metres/sec)
float32 vel_e_m_s # GPS East velocity, (metres/sec)
float32 vel_d_m_s # GPS Down velocity, (metres/sec)
float32 heading # heading angle of XYZ body frame rel to NED, Set to
NaN if not available and updated (used for dual antenna GPS), (rad, [-PI,
PI])

```



100

5.5. Exp5_IMUDataGet.slx—IMU 获取文件

多旋翼 IMU（惯性测量单元）数据通常包括以下信息：

1、加速度数据：沿载具的三个轴（通常是 X、Y 和 Z 轴）的加速度测量值。这些值通常以重力加速度（g）为单位，并在静止状态下应接近于 1g。加速度数据可用于估计载具的加速度、姿态和运动状态。

2、角速度数据：沿载具的三个轴（通常是 X、Y 和 Z 轴）的角速度测量值。这些值通常以角度每秒（degrees per second, dps）或弧度每秒（radians per second, rad/s）为单位。角速度数据可用于估计载具的旋转速率和姿态变化。

3、磁场数据：通常包括沿载具的三个轴（通常是 X、Y 和 Z 轴）的磁场测量值。这些值通常以高斯（Gauss）为单位。磁场数据可用于辅助定位和导航，尤其在结合地磁传感器时。

4、温度数据：IMU传感器的工作温度，通常以摄氏度（℃）或华氏度（℉）为单位。温度数据可以用于校准传感器，并可能影响传感器的性能。

5、传感器状态信息：包括传感器的校准状态、运行状态、健康状态等信息。

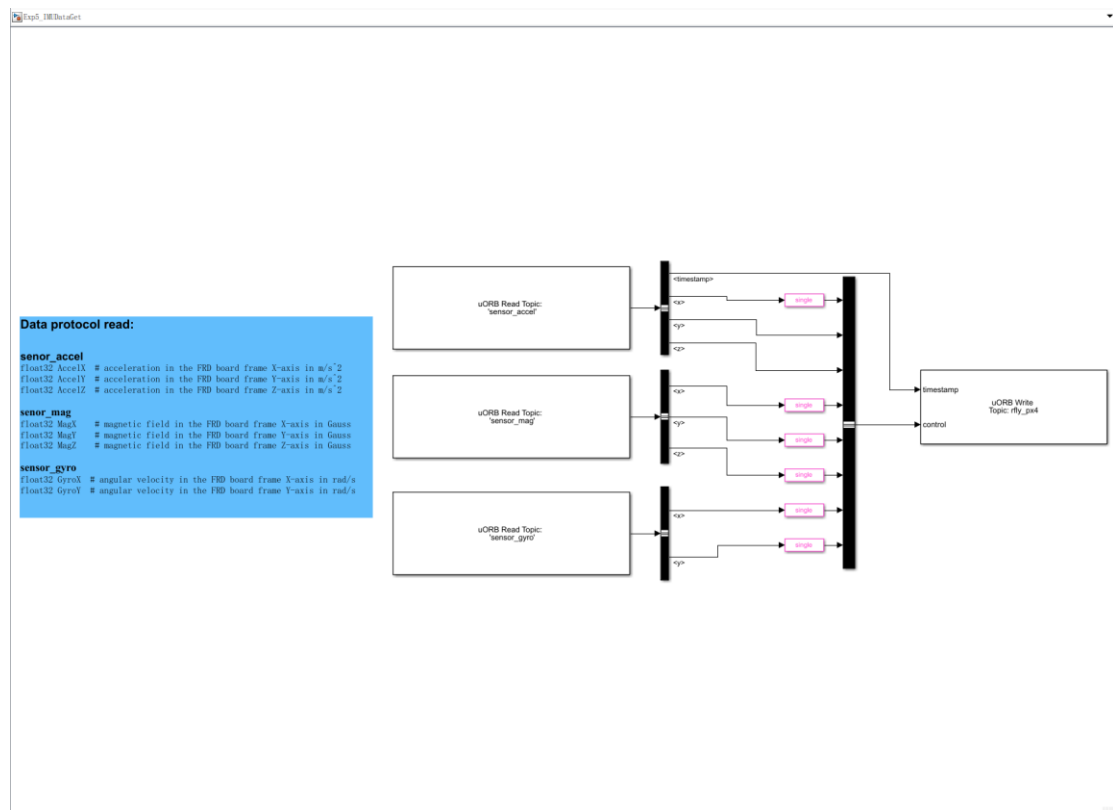
这些 IMU 数据对于多旋翼飞行器的姿态估计、导航和控制至关重要。通过处理 IMU 数据，飞行控制系统可以实现对飞行器的准确控制和稳定飞行。

如下图所示，打开[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\11.StateDataGatAPI](#)中的 `Exp5_IMUDataGet.slx` 文件。该系统主要由 uORB Read and Function-Call Trigger 和 uORB Write Advanced 模块构成。

uORB Read and Function-Call Trigger 和 uORB Write Advanced 具体用法和功能请参考：

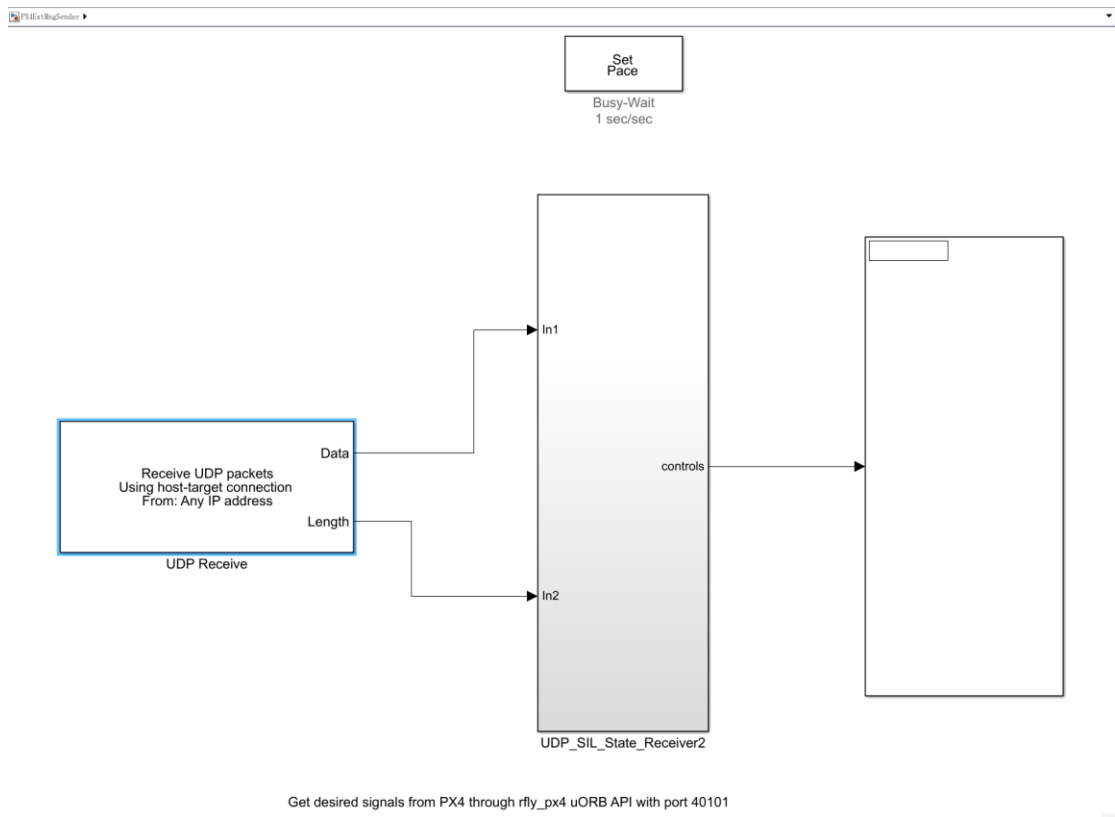
[API.pdf](#)

该模型也用到了 Data Type Conversion（数据类型转换）模块，关于 Data Type Conversion（数据类型转换）模块的详细介绍请参考：[Exp1_VehicleAttDataGet.slx—载具姿态数据获取文件](#)



5.6. PX4ExtMsgSender.slx—飞控 uORB 消息监听文件

如下图所示，打开[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\11.StateDataGatAPI](#) 中的 Exp5_IMUDataGet.slx 文件。该模型由 Simulation Pace、UDP Receive、UDP_SIL_State_Receiver 和 Display 模块组成。

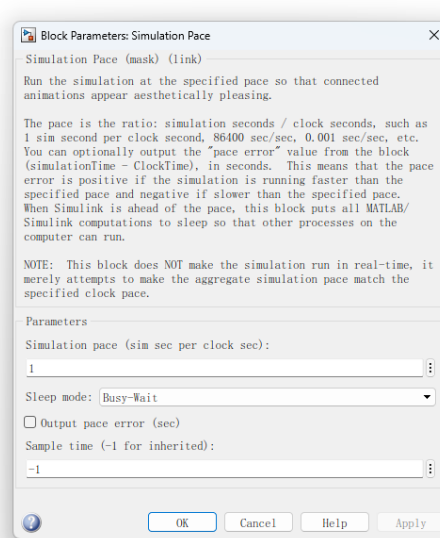
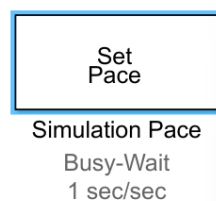


1、Simulation Pace 模块

以指定的速度运行模拟，以便连接的动画看起来美观。

节奏是比率:模拟秒/时钟秒，如 1 sim 秒/时钟秒，86400 秒/秒，0.001 秒/秒等。您可以选择从块(simulationTime - ClockTime)输出“速度错误”值，以秒为单位。这意味着如果模拟比指定的速度快，则速度误差为正，如果比指定的速度慢，则速度误差为负。当 Simulink 领先时，该块将所有 MATLAB/Simulink 计算置于睡眠状态，以便计算机上的其他进程可以运行。

注意:这个块不会使模拟实时运行，它只是试图使聚合模拟速度与指定的时钟速度匹配。



2、UDP Receive 模块

UDP Receive 模块用于接收通过 UDP（User Datagram Protocol）协议发送的数据。UDP 是一种无连接的、不可靠的网络通信协议，常用于在网络上以较低的延迟传输数据。

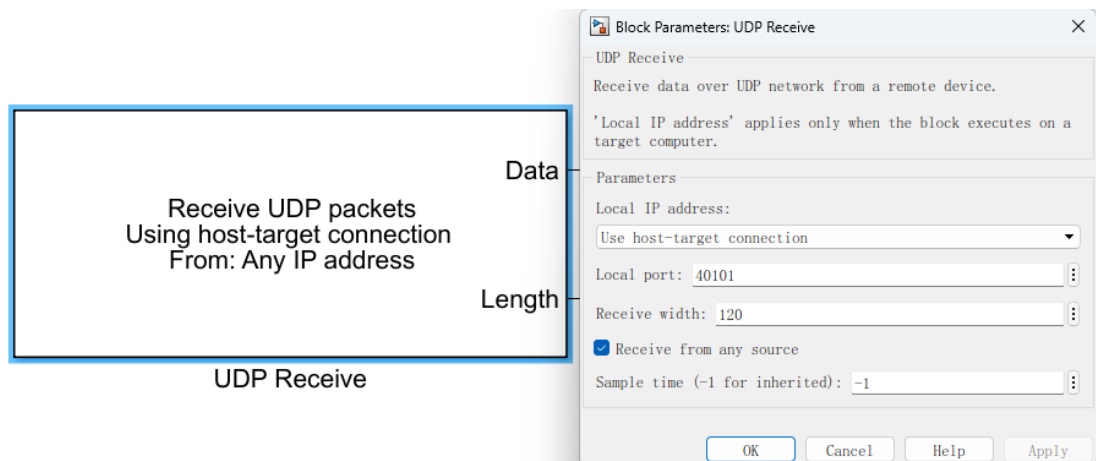
UDP Receive 模块允许 Simulink 模型接收来自网络上特定端口的 UDP 数据包，并将其作为输入提供给模型。这在模拟网络通信或与外部设备进行实时数据交换时非常有用。

以下是 UDP Receive 模块的用法：

配置接收端口：在使用 UDP Receive 模块之前，您需要指定要接收数据的 UDP 端口号。这通常在模块的参数设置中完成。

接收数据：UDP Receive 模块会在每个仿真步骤中检查指定端口是否有数据到达。如果有数据到达，模块将数据读取并将其提供给 Simulink 模型的输出端口。

处理接收到的数据：一旦数据被接收到，您可以在 Simulink 模型中使用其他模块来处理这些数据，比如数据分析、控制算法等。



3、UDP_SIL_State_Receiver 模块

该模块中有一个 VehicleDataPerse 函数，这个函数的目的是检查输入数据的长度和校验和，并在通过检查时将输入数据的前 48 个字节复制到输出变量 y 中，并将 dataOk 设置为 1 表示数据有效。

函数解析如下：

接受两个输入参数 Data 和 Length，并返回两个输出 y 和 dataOk。

```
function [y,dataOk] = fcn(Data,Length)
```

初始化了一个大小为 48x1 的 uint8 类型的数组 y，并初始化 dataOk 为 0。

```
y=uint8(zeros(48,1));  
dataOk = 0;
```

检查输入参数 Length 是否等于 48，如果不等于，则函数返回空（return;），即不执行后面的代码。

```
if Length ~= 48  
    return;  
end
```

从输入参数 Data 中提取前 4 个字节，并将其视为 int32 类型的 checksum（校验和）。

```
checksum=typecast(Data(1:4), 'int32');
```

如果这个 checksum 不等于预先设定的值 1234567898，则同样返回空。

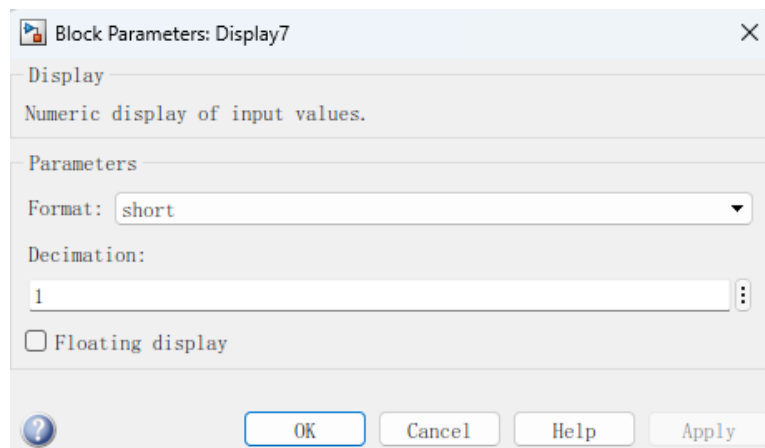
```
if checksum ~= 1234567898
    return;
end
```

如果上述条件都满足，则将 Data 中的前 48 个字节复制到输出变量 y 中，并将 dataOk 设置为 1，表示数据有效

```
y = Data(1:48);
dataOk = 1;
```

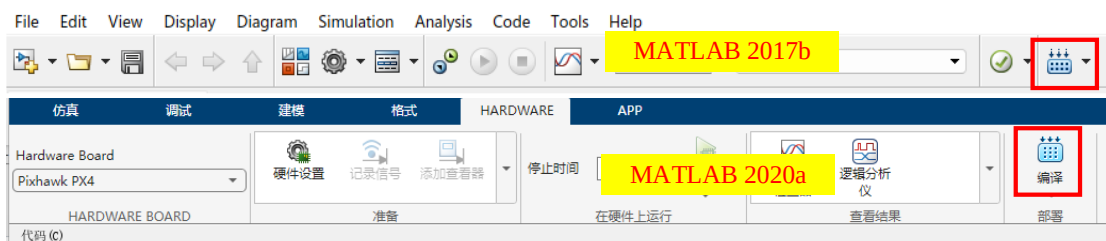
4、Display 模块

Display 块通常用于调试和监视仿真的输出。它可以显示信号的数值、波形图、二维或三维图形等。

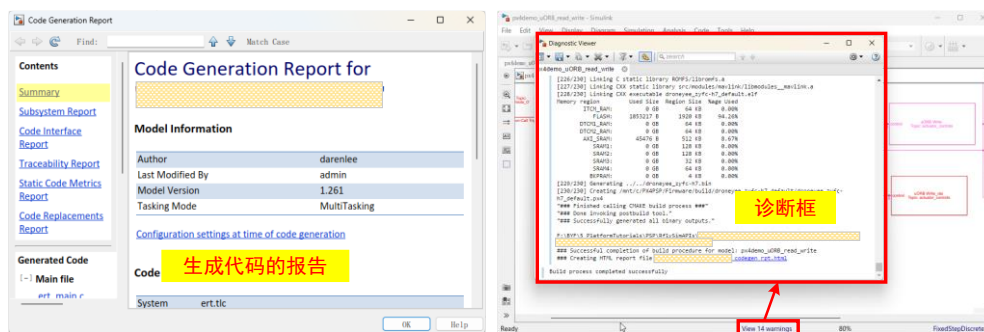


5.7. 飞控状态数据读取

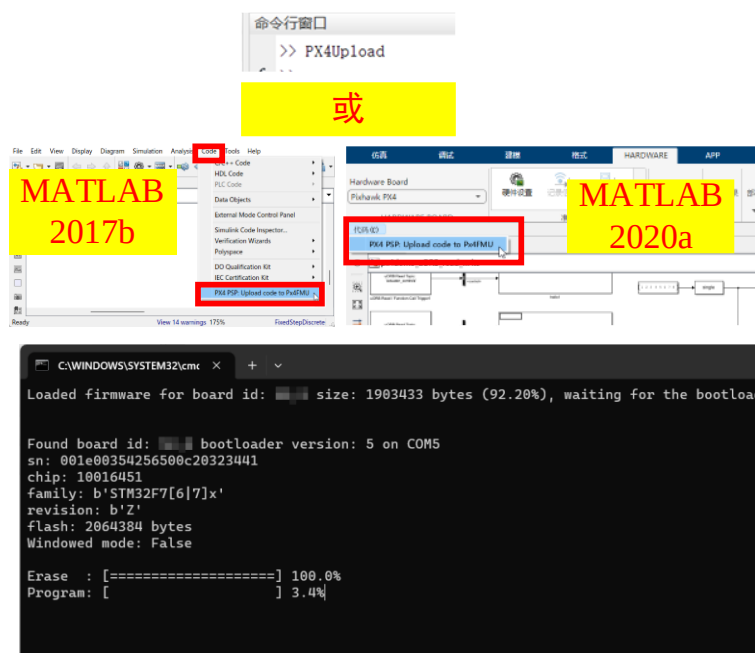
打开 MATLAB 软件，在 MATLAB 中打开 [Exp5_IMUDataGet.slx](#) 文件，在 Simulink 中，点击编译命令。



在 Simulink 的下方点击 View diagnostics 指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出 Build process completed successfully，即可表示编译成功，左图为生成的编译报告。



用 USB 数据线链接飞控与电脑。在 MATLAB 命令行窗口输入：PX4Upload 并运行或点击 PX4 PSP: Upload code to Px4FMU，弹出 CMD 对话框，显示正在上传固件至飞控中，等待上传成功。



打开 QGroundControl 软件，设置机架为 “HIL Quadcopter X” 之后，关闭 QGC。

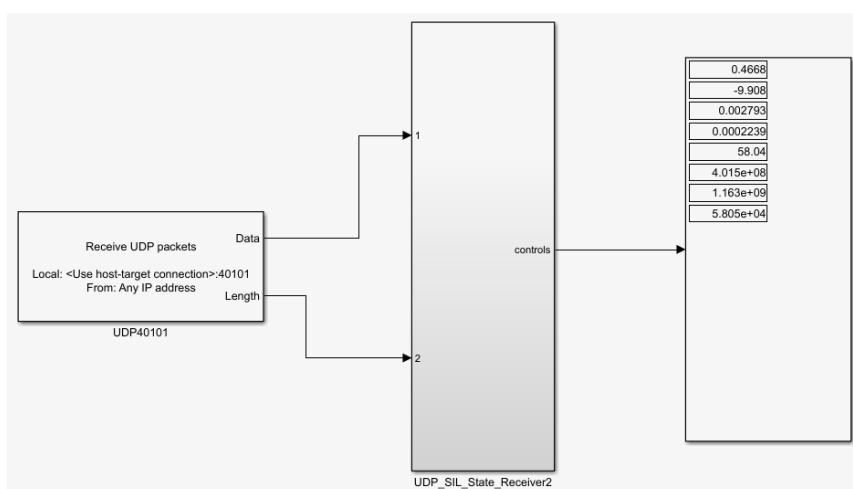


打开 CopterSim 软件，进行如下图所示设置之后，其中飞控选择一栏需要选择本电脑所识别到的飞控 COM 端口号。设置完成之后，点击开始仿真。

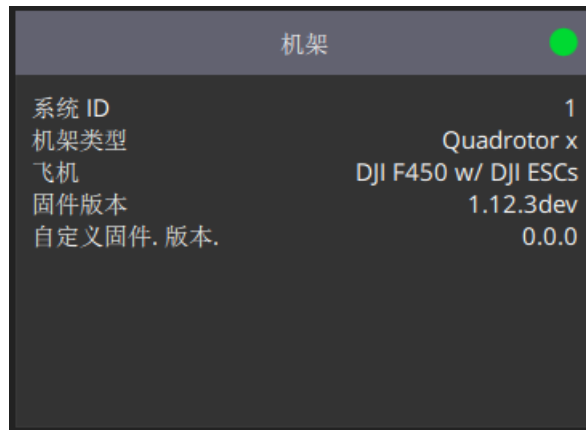
等待 CopterSim 显示初始化完成。

在 CopterSim 运行的运行过程中，在 MATLAB 中打开 PX4ExtMsgSender.slx 监听程序，点击运行即可监听到写入的 uORB 消息，如下图所示：

注：此时获取到的为 CopterSim 软件中生成的传感器数据。若想获取到飞控中传感器数据请见 Step 7~8。



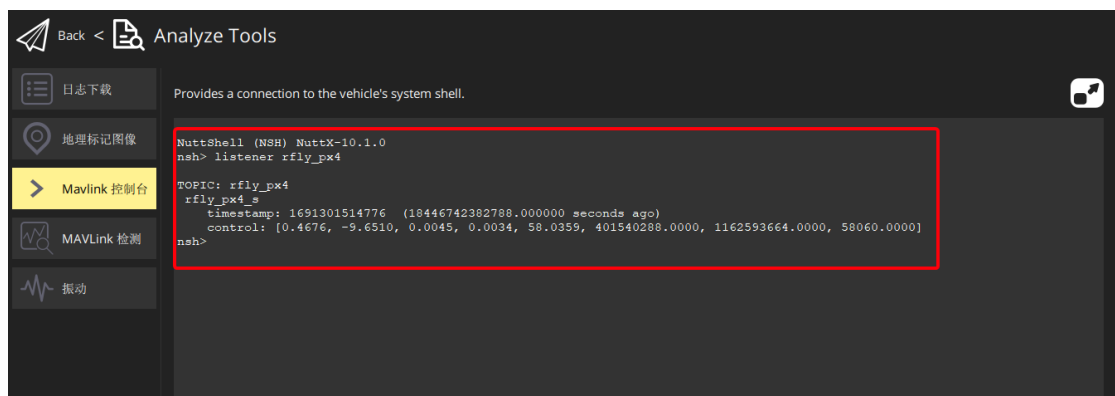
上述 Step4~6 为在硬件在环仿真时，由模型生成的各种状态数据；而在实飞时，需要打开 QGroundControl 软件，切换机架为 “DJI 450 w/DJI ESCs” 如下图所示。



在 QGroundControl 软件初始界面下，点击左上角 Logo 在弹出的对话框中，选中 Analyze Tools，在 Mavlink 控制台中输入：

```
listener rfly_px4
```

即可得出所订阅的 uORB 消息，如下图所示：



6. 参考资料

本实验硬件在环仿真数据流： Simulink 控制器算法生成代码下载到飞控中，然后用 USB 实体信号线替代 Simulink 中的虚拟信号线。CopterSim 将传感器数据（例如，加速度计、气压计、磁力计等）通过 USB 数据线发送给飞控系统；飞控系统内的 PX4 自动驾驶软件将收到传感器数据进行滤波和状态估计，将估计的状态信息通过内部的 uORB 消息总线发送给控制器；控制器再通过 USB 数据线将“rfly_px4”的 uORB 消息回给 CopterSim，CopterSim 通过 UDP 的方式以 40100 系列端口将该数据转发出去，启动 Simulink 的 PX4ExtMsgSender.slx 即可实时监听到所订阅的各种传感器信息，从而形成一个硬件在环仿真闭环。

本实验实飞数据流： 在 Simulink 中搭建各种传感器数据的获取文件，该文件内部均是一 uORB 消息订阅的方式订阅各种传感器的数据；自动代码生成的过程中会在 PX4 软件系统中生成名为 px4_simulink_app 的模块应用，通过烧录到飞控中与其他模块形成并行运行。在 QGC 中通过监听“rfly_px4”的 uORB 消息即可接收所自定义的消息。

本实验所涉及到的各种数据的具体定义请见：【RflySim 平台安装目录】\Firmware\msg/*中。

7. 常见问题

Q1: ***

A1: ***