飞控系统Simulink仿真说明

目录

[飞控系统Simulink仿真说明 1](#_Toc460418870)

[1 Simulink模型构架 2](#_Toc460418871)

[1.1 模块库 3](#_Toc460418872)

[1.1.1 优点 3](#_Toc460418873)

[1.1.2 注意 3](#_Toc460418874)

[1.2 仿真用例 4](#_Toc460418875)

[1.3 日志分析 4](#_Toc460418876)

[1.4 模型管理 5](#_Toc460418877)

[1.5 运行频率管理 5](#_Toc460418878)

[2 Simulink顶层模型 6](#_Toc460418879)

[2.1 数据结构体 7](#_Toc460418880)

[2.2 RemoteController 10](#_Toc460418881)

[2.3 ArduCopter 11](#_Toc460418882)

[RemoteController 11](#_Toc460418883)

[2.4 QuadCopter 11](#_Toc460418884)

[2.5 3DVisualization 12](#_Toc460418885)

[3 QuadCopter 12](#_Toc460418886)

[3.1 动力计算 13](#_Toc460418887)

[3.1.1 电机模型 13](#_Toc460418888)

[3.1.2 合力计算 15](#_Toc460418889)

[3.1.3 合力矩计算 15](#_Toc460418890)

[3.2 动力学模型 16](#_Toc460418891)

[3.2.1 位置计算 17](#_Toc460418892)

[3.2.2 姿态计算 17](#_Toc460418893)

[3.3 传感器建模 19](#_Toc460418894)

[3.3.1 加速度计 19](#_Toc460418895)

[3.3.2 陀螺仪 20](#_Toc460418896)

[3.3.3 磁罗盘 20](#_Toc460418897)

[3.3.4 小结 21](#_Toc460418898)

[4 AHRS 21](#_Toc460418899)

[4.1 模型接口 21](#_Toc460418900)

[4.2 姿态估计算法 22](#_Toc460418901)

[5 ArduCopter 24](#_Toc460418902)

[5.1 动力混合 24](#_Toc460418903)

[5.1.1 动力布局 24](#_Toc460418904)

[5.1.2 电机控制 26](#_Toc460418905)

[5.2 Stabilize模式 26](#_Toc460418906)

[5.3 Althold高度控制 27](#_Toc460418907)

[5.4 Poshold位置控制 27](#_Toc460418908)

[5.4.1 控制框图 27](#_Toc460418909)

[5.4.2 姿态转换 27](#_Toc460418910)

[6 3DVisualization 29](#_Toc460418911)

[模型演示 29](#_Toc460418912)

[附录一 模块库常用模块说明 29](#_Toc460418913)

[Lib\_AP\_Math 30](#_Toc460418914)

[Lib\_AC\_PID 31](#_Toc460418915)

[Lib\_AC\_AttitudeControl 32](#_Toc460418916)

[Lib\_AP\_Motors 33](#_Toc460418917)

[Lib\_Sensors 33](#_Toc460418918)

文档中称现在用的PixHawk3.4版本飞控为“C版飞控”；为了研究方便，Simulink模型中的函数和变量命名尽量和现有飞控一致；变量如无特殊说明，均采用国际单位，正方向与标准NED坐标相同。

# Simulink模型构架

Simulink模型主要作用：算法设计与验证、代码自动生成、3D效果演示。基于Simulink平台，可以将开发分为四个阶段：全数字仿真、基于物理数据仿真、半物理仿真、代码自动生成并进行实际调试。完成Simulink模型的主要工作内容包括飞行器建模、飞控算法设计。

随着以后的完善，Simulink模型会越来越大、越复杂，为了高效开发和应用，有必要对Simulink模型的框架进行思考。

模型全局架构：总体采用Simulink框图结构，复杂逻辑用Stateflow实现，复杂计算用.m脚本（考虑模块复用的继承性，尽量采用Simulink模块的方式），其余用Simulink库文件（可以是自定义的库文件）搭建；飞行器模型采用连续域方式建模，飞控算法部分采用离散域方式设计。

根据Simulink模型的作用和全局框架，对Simulink整体模型构架有如下要求：

1. 方便算法设计与功能扩充
2. 方便共同开发与管理
3. 容易理解阅读与传承

根据以上需求，可以采用以下方案构建Simulink模型。

## 模块库

以模块库的思想进行建模，既应用两次以上的模块考虑将其放入对应模块库。

### 优点

1. 方便别人应用与共同开发

第一次应用，运行文件夹中slblocks.m即可加载相应库文件，以后应用不必再次加载。

将库文件存放在SVN上，经常跟新库文件夹即可，重新启动Simulink即可自动加载库文件。

1. 方便修改、更新

模块库内的模块修改/更新后，打开.slx模型后软件会自动更新模型内的相应模块。这样在某个模块被多次调用的情况下可以实现一次性修改/更新。

1. 增加易读性

针对模块库内文件，可以在模块属性中添加注释说明，甚至可以有自己模块的帮助文档。

模块库的创建和使用方法可以参见：

http://blog.csdn.net/dfd1r/article/details/6278114

### 注意

1. 应用两次以上的模块考虑建立库，否则不需要，反而麻烦。
2. 做到算法的完全把握，可以自己建立库文件；如果simulink自带模块库完全符合要求，也可以用，但是要彻底清楚模块的计算方法，否则有些情况仿真出问题很难调试。不过要彻底弄清楚模块的计算方法和自己搭建库文件的工作量差不多。
3. 模型中应用库文件模块的地方不能直接对模块进行修改，要右键->Library link->Go to Library，进入库文件后解锁才能修改。
4. 库文件的层次很重要。暂时采用两层库文件结构，既一个库文件里面包换若干子库，各个子库子不再包含库文件。
5. 加入的模块尽量保证matalb可以自动生成代码

## 仿真用例

模型仿真需要控制端输入，比如遥控器输入的期望姿态角、期望高度等；模型运行需要加载很多参数，比如控制模型中的PID参数、AHRS中的滤波参数。可以将这两类数据统称为仿真用例。仿真过程中会经常修改控制参数，或者需要采用批量仿真测试算法（或自动化测试），所以仿真用例的设计要方便阅读、增删、修改。

现有模型采用如下方式：

1. 通过.m文件对仿真参数用例中的数据进行定义和初始化

参数加载可以采用.mat 或.m文件，为方便开发和阅读，选用.m。模型中自定义数据结构体以.mat的形式进行加载。

（2）将仿真用例数据生成.mat文件，模型调用.mat文件中的数据作为输入进行仿真（检验可以，但是考虑到临时修改方便，暂时模型没有用这种方式；需做自动化测试的时候，可以采用.mat文件的形式）

参数分为两类：

1. 常量。比如Pi，radian和degree的比例系数等，这类参数是永远不会改变的，只是在不同情况下我们选择不同的精确位数。这类可以直接放到相应模块中。
2. 控制常量。PID参数等。这类最好在模块接口中有所体现。

## 日志分析

如同飞机上的log管理与分析一样，仿真产生的日志同样需要合理的管理和分析。

常用log，比如飞机的姿态、位置信息，可以采用如下方式：

Log记录：Simulink中将需要分析的数据导入工作空间或生成.mat文件。

Log分析：采用.m文件的形式进行处理分析。

如果有必要的话，Log管理思想可以和pickhawk一致，每个模块有相应的log记录和log分析程序，所有log记录和log分析程序统一管理。

不常用的log可以通过加scope模块简单实现，方便添加删除。

## 模型管理

Simulink模型采用SVN进行代码管理，目前matlab2014b以上版本支持SVN管理。考虑到SVN对Simulink模型的管理效果可能暂时不如它对c文件的管理效果好，可对Simulink模型的管理采用如下方式：

1. 总模型分为几个模块进行开发，不同人开发不同的模块
2. 每个模块库有对应的固定的人进行开发和完善

Simulink模型如果无法实现交叉开发，这就要求开发人员开发前做好分工交流，明确各自开发模块的对外接口，然后每个人开发完善自己的模块。

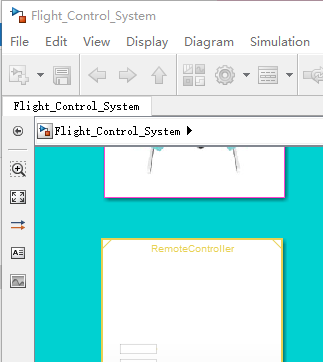
随着Matlab的完善，利用SVN对Simulink模型管理将会越来越方便。

## 运行频率管理

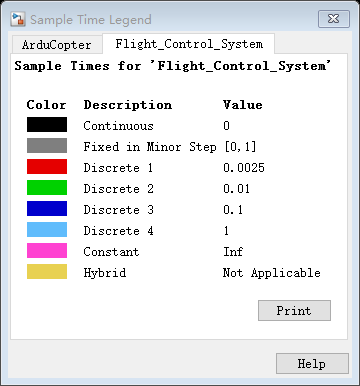
ArduCopter模型中有不同频率的算法模块，这就要求Simulink模型中对不同频率的算法做合适的处理。应用频率转换模块，作为不同频率信号间的转换模块，这样既可以保证算法的频率，又可以保证实时代码生成不会因为频率设置出问题。使用频率转换模块时，最好以最小频率作为中间转换，防止仿真出错（可以参见ArduCopter/loop/scheduler.run/frequence\_change模块）。

仿真时可以设置为用颜色区分频率，这样有频率不对的模块可以很明显的发现，防止运行频率不符合要求导致bug。设置方法：

运行模型后，点击图中红框内标签，选择“Colors”，



显示如下：



模型中，算法采用离散域建模，被控对象采用连续域建模，具体仿真设置可以参考相应模型中的Simulation->Model Configuration Parameters窗口的相应设置。

# Simulink顶层模型

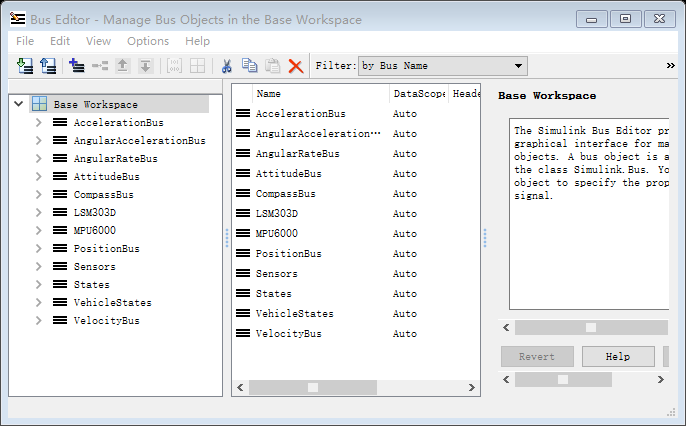
先说明模型中的数据结构体，然后对顶层模型的输入输出接口进行说明。

Flight\_Control\_System.slx为Simulink总体模型，总体模型分为四个顶层模型：RemoteController、ArduCopter、QuadCopter、3DVisualization，这四个模块以reference model的形式嵌入到总体模型中。由于AHRS的特殊性，AHRS模块以reference model的形式嵌入到ArduCopter模型中，方便独立开发。每个reference model模型都会有独立的.slx文件，如ArduCopter.slx、QuadCopter.slx。

## 数据结构体

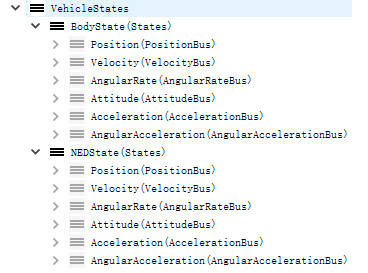
方便建模起见，进行数据类型自定义。模型中自定义的数据类型存储在BusParameters.mat文件中。

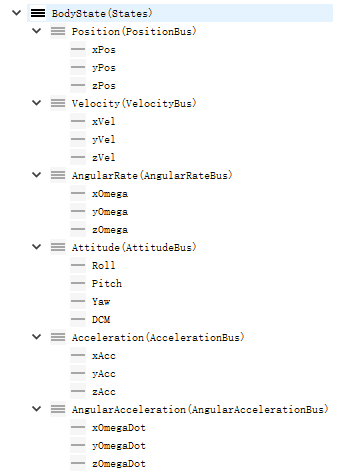
查看方法：双击BusParameters.mat，加载总线，进入模型，选择Edit菜单下的Bus Editor选型，如下图：

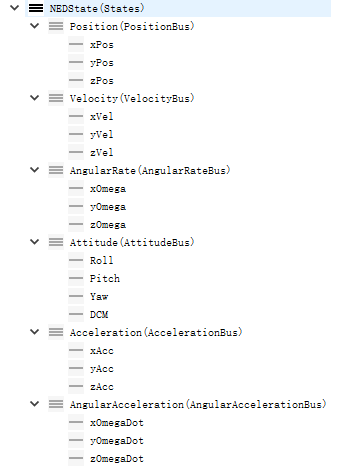


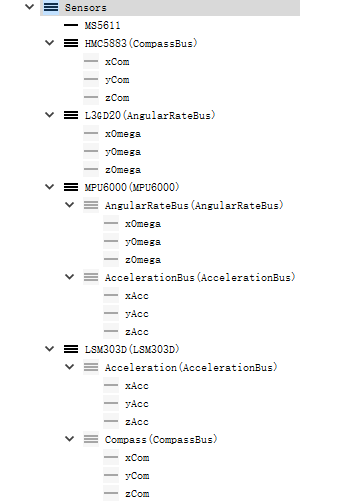
BusParameters可以是.mat或.m文件，两种类型文件可以相互转换（通过Explore完成）

模型中用到的数据类型构架如下：









## RemoteController

RemoteController.slx是遥控器模型，模拟遥控器输出从而控制飞机。遥控器各个通道输出消息定义暂时如下表:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 通道  序号 | 消息内容 | | | 备注 |
| Stabilize | Alt\_hold | Pos\_hold |
| Ch1 | Roll angle | Roll angle | Velocity x | - |
| Ch2 | Pitch angle | Pitch angle/rad | Velocity y | - |
| Ch3 | Yaw rate | Yaw rate | Yaw rate | - |
| Ch4 | Throttle | Climb rate | Climb rate | - |
| Ch5 | Mode（1/2/3 对应stibilize/alt\_hold/pos\_hold） | | | - |
| Ch6 | Simple mode（ 1/2 对应 有头/无头） | | | - |
| Ch7 | - | - | - | - |
| Ch8 | -- | - | - | - |

注：为与C版飞控单位一致,表格中速度的单位为cm/s，throttle~[0, 1]。

## ArduCopter

ArduCopter.slx 是飞控算法模块。模块输入输出如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量 | 类型 | 备注 |
| 输入 | Rc\_in | - | RemoteController 输出数据 |
| StateTest | VehicleStates（自定义总线） | 参考数据结构体说明，仅用于AHRS调试对比。自动代码生成的时候要去掉这个输入。 |
| Sensors | Sensors（自定义总线） | 参考数据结构体说明 |
| 输出 | Motor\_out[4] | [4x1]double | 发送给每个电机的PWM控制信号 |

ArduCopter模型中暂时不考虑信号灯模块的控制信号输出。

## QuadCopter

QuadCopter.slx 是四轴飞行器模型。模型输入输出如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量 | 类型 | 备注 |
| 输入 | Motor\_out[4] | [4x1]double | 发送给每个电机的PWM控制信号 |
| 输出 | State | VehicleStates（自定义总线） | 参考数据结构体说明。 |
| Sensors | Sensors（自定义总线） | 参考数据结构体说明 |

## 3DVisualization

3DVisualization.slx是四轴飞行器3D飞行模拟模块，模块输入输出如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量 | 类型 | 备注 |
| 输入 | State | VehicleStates（自定义总线） | 参考数据结构体说明。 |
| 输出 | - | - | 3D动画演示 |

# QuadCopter

本章说明四轴飞行器物理模型建模方法。建模分为三个方面：动力计算、动力学模型、传感器模型。

QuadCopter模型输入输出表格如下：

|  |  |  |
| --- | --- | --- |
| 类型 | 变量 | 说明 |
| 输入 | motor\_out | 输入给四个电机的PWM值 |
| 输出  Sensors | L3GD20 | 陀螺仪(只有速率) |
| MPU-6000 | 集成了3轴MEMS陀螺仪，3轴MEMS加速度计 |
| LSM303D | LSM303D加速度传感器电磁罗盘复合模块 |
| MS5611-01BA | 气压计（输出高度） |
| hmc5883 | 单独的磁罗盘 |

## 动力计算

本小节说明电机产生的力、力矩的计算方法。首先建立电机螺旋桨模型（简称电机模型），然后计算螺旋桨作用于飞机的合力、合力矩。

对应于QuadCopter中的Forces and Moments模块，输入输出如下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类型 | 变量 | 维度 | 意义 | 范围 |
| 输入 | MotorCommands | [4, 1] | 动力混合输出的motor\_out | [1100, 1900] |
| VehicleState | VehicleState Bus | 参见数据结构 | - |
| 输出 | Forces | [3, 1] | 作用于飞机的合力，以大地坐标为参考 | - |
| Moments | [3, 1] | 作用于飞机的合力矩，以大地坐标为参考 | - |

### 电机模型

电机模型输入PWM控制信号，输出电机带动螺旋桨产生的升力和力矩。

1. 动态模型

电机动态模型用于反应电机的响应时间。

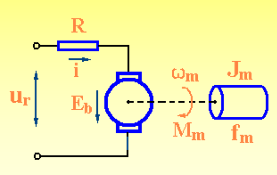
电机输入PWM、输出转速，此时电机模型可以近视为一阶惯性环节M（S），时间常数为T。 电机响应时间设置为t=70ms，则T=t/3。

电机的单位增益传递函数模型如下：



电机模型论证

电机原理图如下：



模型输入驱动电压，输出转动角速度：

电枢回路：  — 基尔霍夫



电枢反电势：  — 楞次定律



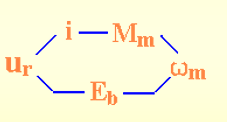
电磁力矩：  — 安培定律



力矩平衡：  — 牛顿定律



变量关系如下图：

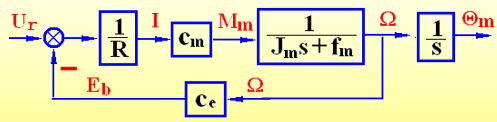


五个变量四个方程，故可以到处输入输出关系：



其中：





直流电动机结构图

1. 力模型

电机的力模型用于稳态情况下计算电机输出升力与PWM信号间的关系。

中间参数x定义：

x=(motor\_output-1000)/10,

因为motor\_output~[1100, 1900]

所以x~[10, 90]

稳态情况下，电机输出力F与x成二次函数关系：



暂时设置参数如下：

a=3\*1.0909e-04

b=3\*0.0076

c=3\*0.0906

电机模型系数motorCoeffs=[a b c]。

动态情况下：



由于螺旋桨力F都指向Z轴负方向，故前面是负号。

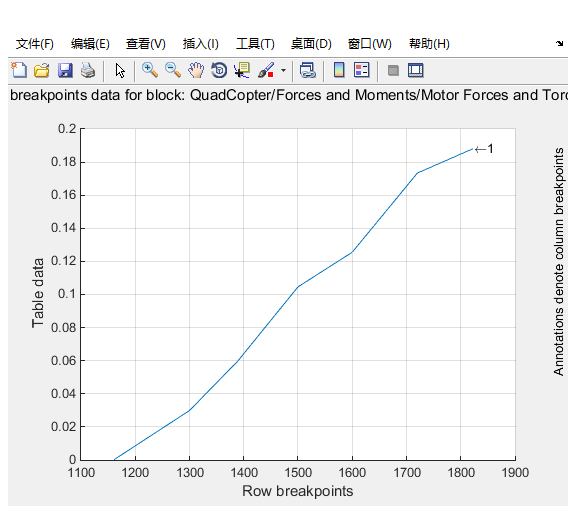
1. 力矩模型

电机力矩模型输入PWM（Motor\_out）控制信号x，输出电机力矩M。

稳态情况下，利用查表方式建模，x即为Mortor\_out：

M=Torque(x)

Torque表格咱取如下设置：



表中，横坐标代表x，纵坐标代表M。

动态情况下：



力矩指向z轴正方向为正。

### 合力计算

本部分计算作用于飞机上的螺旋桨升力与重力的合力（机体坐标系）。

1. 重力计算

机体坐标系下飞机所受重力：

Gravityforce=DCM\*g

其中：

G=[0 0 9.8]`

DCM是由飞机动力学模型输出的旋转矩阵，在VehicleState Bus中的Attitude Bus中。

其它细节参见模型。

1. 合力计算

F=F\_Motor1+ F\_Motor2+ F\_Motor3+ F\_Motor4+ Gravityforce

### 合力矩计算

本部分计算飞机受到的合力矩。

X轴方向合力矩Mx：

Mx=Qlx\*Mxfactor\*MotorForces

轴方向合力矩My：

My=Qly\*Myfactor\*MotorForces

Z轴方向合力矩Mz：

Mz=0-(M1+M2+M3+M4)

其中Qlx为电机相对于x轴的力臂，Qly同理，模型中按对角电机距离为34cm进行计算。Mxfactor为每个电机对x轴力矩的影响因子，Myfactor同理。

Mxfactor=[1 -1 -1 1]（注意力与力矩方向，力的正方向是指向地的）

Mz的计算方程是根据z轴方向动量矩守恒（力矩和为0）来计算的。

合力矩M：



Mxfactor、Myfactor等影响因子可以根据动力布局矩阵MotorLayout自动计算（暂时设置为如上恒定值），方便以后动力布局改变的情况下进行仿真。

## 动力学模型（这两个公式自己多推）

本小节说明已知机体受到的的合力F和合力矩M，求机体运动状态的方法。

机体坐标系的坐标轴都是刚体在原点处的惯性主轴。

应用6自由度的动力学方程进行建模。

飞行器动力学模型输入输出如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类型 | 序号 | 符号 | 维度 | 名称 |
| 输入 | 1 |  | [3, 1] | 飞机受力F在机体坐标系下的表示 |
| 2 |  | [3, 1] | 飞机受到的力矩在机体坐标系下的表示 |
| 输出 | 1 |  | [3, 1] | 飞机速度在NED标系下的表示 |
| 2 |  | [3, 1] | 飞机位置在NED标系下的表示 |
| 3 |  | [3, 1] | 欧拉角，分别代表roll pitch yaw |
| 4 |  | [3, 3] | 从大地到机体坐标系的旋转矩阵 |
| 5 |  | [3, 1] | 飞机运动速度在机体坐标系下的表示 |
| 6 |  | [3, 1] | 飞机转动速度在机体坐标系下的表示 |
| 7 |  | [3, 1] | 对时间的导数 |
| 8 |  | [3, 1] | 飞机加速度在机体坐标系下的表示 |

### 位置计算

预备:公式推导参见理论力学P137.

飞行器的运动可以分解为绕定轴转动、平移运动。设飞行器以角速度绕NED坐标系中某定轴转动，坐标系与飞行器固连，坐标系各个轴矢量为，

根据泊松公式，有：



飞行器平移运动的速度为，则:



其中：



应用泊松公式，有：





上方程是基于动系坐标推到得出。

也可基于非惯性系牛顿定律推导出，略。

飞行器位置动力学方程如下：（推导出矩阵形式）



其中：



其中：

：飞机受力F在机体坐标系下的表示

： 飞机飞行速度在机体坐标系下的表示

： 飞机转动速度在机体坐标系下的表示

### 姿态计算

根据刚体定点运动的欧拉动力学方程（矩阵形式），有：





推导参见P362

其中：



： 飞机受到的力矩在机体坐标系下的表示

： 机体坐标系下，飞机相对于自身中心的惯量矩阵

根据位置动力学方程和转动动力学方程，两个不相关方程，两个未知量，可以求解出：、，从而得到，由下关系可求的可求得：



旋转矩阵:



根据欧拉角定义， 可得：



上式整理得：



故根据上式可以求出欧拉角，根据旋转矩阵定义，可得：



和由如下公式得到：



这样8个输出参数就都得到了。

## 传感器建模

本部分对传感器进行建模，建模暂时考虑稳态下传感器特性。

### 加速度计

输入输出参数如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类型 | 序号 | 符号 | 维度 | 名称 |
| 输入 | 1 |  | [3, 3] | 从大地到机体坐标系的旋转矩阵 |
| 2 | g | [3, 1] | 重力加速度在NED坐标系下的表示 |
| 3 | g0 | [3, 1] | 加速度计的初始值 |
| 4 | N | [3, 3] | 传感器测量噪声协方差矩阵 |
| 5 | B | [3, 1] | 传感器零偏 |
| 6 | T | 1 | 响应时间常数，在内部Transfer Fcn中设置 |
| 输出 | 1 | a | [3, 1] | 加速度计测量值 |

计算方法：



上式中，传感器的初值并不合理。比如建模会使得加速度计的初值是B，这个并不合理，故改成：



根据现阶段采用的加速度计硬件工作原理，加速度计测量的是非重力加速度。大机动状态下，应该根据机体加速度和重力两个作用求解加速度计的输出值，以后完善。

### 陀螺仪

输入输出参数如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类型 | 序号 | 符号 | 维度 | 名称 |
| 输入 | 1 | Omega | [3, 1] | 飞行器旋转速度在机体坐标系下的表示 |
| 3 | N | [3, 3] | 传感器测量噪声协方差矩阵 |
| 4 | B | [3, 1] | 传感器零偏 |
| 5 | T | 1 | 响应时间常数 |
| 输出 | 1 | Gyro | [3, 1] | 陀螺仪测量值 |

计算方法：



### 磁罗盘

输入输出参数如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类型 | 序号 | 符号 | 维度 | 名称 |
| 输入 | 1 |  | [3, 3] | 从大地到机体坐标系的旋转矩阵 |
| 2 |  | [3, 1] | 地磁场强度在NED坐标系下的表示 |
| 3 |  | [3, 1] | 地磁场强度初始值 |
| 4 | N | [3, 3] | 传感器测量噪声协方差矩阵 |
| 5 | B | [3, 1] | 传感器零偏 |
| 6 | T | 1 | 响应时间常数 |
| 输出 | 1 | m | [3, 1] | 磁罗盘测量值 |

计算方法：



### 小结

以后有必要根据相应传感器的工作机理进行建模，这样模型才能更真实的模拟大机动状态下传感器的真实输出值。

# AHRS

AHRS计算量较大、算法比较复杂、可独立成系统，故为方便研究，将AHRS模块以reference model 的形式嵌入到ArduCopter模型中。

## 模型接口

其输入输出如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类型 | | 传感器型号 | 信息 | 备注 |
| 输入  Sensors | | L3GD20 | 陀螺仪(只输出速率) | 数据类型说明参见数据结构体部分。输出部分可根据实际需要进行修改 |
| MPU-6000 | 集成了3轴MEMS陀螺仪，3轴MEMS加速度计 |
| LSM303D | LSM303D加速度传感器电磁罗盘复合模块 |
| MS5611 | 气压计（输出高度） |
| hmc5883 | 单独的磁罗盘 |
| 输出  ahrs  （vehicle state Bus） | BodyState  （机体坐标系下） | Position | 位置 |
| Velocity | 速度 |
| Acceleration | 加速度 |
| Attitude | 姿态 |
| AngularRate | 角速度 |
| AngularAcceleration | 角加速度 |
| NEDState  （NED坐标系下） | Position | 位置 |
| Velocity | 速度 |
| Acceleration | 加速度 |
| Attitude | 姿态 |
| AngularRate | 角速度 |
| AngularAcceleration | 角加速度 |

## 姿态估计算法

为使控制仿真更接近实际，加入简单的姿态估计算法。姿态估计算法采用三阶kalman滤波。

（1）状态方程：



其中：





是系统噪声。

因为：



所以：



观测方程：



其中：



acc和m分别是加速度计测量值和磁力计测量值。

是观测噪声

是观测矩阵，设置为单位阵。

1. 时间更新方程





其中Q是系统噪声协方差矩阵。

1. 测量更新方程







其中R是观测噪声协方差矩阵。

最后根据：



求出旋转角速度估计。

# ArduCopter

现在完成的ArduCopter算法中，支持自稳、定高、位置控制三种模式，其中前两个模式和C版飞控的总设计思路一致，位置控制算法为自己设计的简单算法。三种模式都支持有头、无头模式的转换。

为保证控制算法可实现性，ArduCopter算法中不要含有代数环。

## 动力混合

根据飞行器动力布局，计算动力混合矩阵，从而计算输出到每个电机的PWM控制信号。

### 动力布局

将动力的物理分布抽象为为动力布局矩阵，将每个电机对姿态的影响作用抽象为动力混合矩阵。

在one\_HZ\_loop中会根据飞机的螺旋桨（电机）布局计算相应的动力混合矩阵，模型中电机布局采用Ghost2.0方式，如下表：

表中紫色部分构成动力布局矩阵MotorLayout。

|  |  |  |  |
| --- | --- | --- | --- |
| 电机序号 | 位置 | 安装角（度） | 转动方向 |
| 1 | 右上角 | 45 | 逆时针（-1） |
| 2 | 左下角 | -135 | 逆时针（-1） |
| 3 | 左上角 | -45 | 顺时针（-1） |
| 4 | 右下角 | 135 | 顺时针（-1） |

注：安装角定义为电机所在机臂与Y轴（NED坐标）正方向夹角。

当有需要增加电机或是修改电机布局后，只要所有电机安装在与机体平行的水平平面内、并且升力垂直向上，需要相应修改以上动力布局矩阵MotroLayout。以后若是电机安装不在与机体平行的水平平面内或升力不垂直向上，可定义更多维度的安装角。

动力混合矩阵MotorMatrix定义如下表，程序会根据MotorLayout矩阵计算MotorMatrix，而不需要手动设置。MotorMatrix计算方法同新飞控动力混合部分；不同之处是：C版飞控采用的向量形式进行编程，Simulink模型采用矩阵形式。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MotorMatrix | Throttle\_factor | Roll\_factor | Pitch\_factor | Yaw\_factor |
| 电机1 | MotorMatrix（1,1） | MotorMatrix（1,2） | MotorMatrix（1,3） | MotorMatrix（1,4） |
| 电机2 | MotorMatrix（2,1） | MotorMatrix（2,2） | MotorMatrix（2,3） | MotorMatrix（2,4） |
| 电机3 | MotorMatrix（3,1） | MotorMatrix（3,2） | MotorMatrix（3,3） | MotorMatrix（3,4） |
| 电机4 | MotorMatrix（4,1） | MotorMatrix（4,2） | MotorMatrix（4,3） | MotorMatrix（4,4） |

举例说明：MotorMatrix（1,1）定义为电机1的Throttle\_factor。

动力混合矩阵是分局电机布局自动计算出来的娥

### 电机控制

根据速率控制器输出信号，应用动力混合矩阵，计算输出给每个电机的PWM控制信号。

电机控制对应于Simulink模型中fast\_loop下的motors\_output模块。其输入输出如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 变量 | 意义 | 范围 |
| 输入  Thrust[4] | \_roll\_in | Desired roll control from attitude controllers | [-1, 1] |
| \_pitch\_in | Desired pitch control from attitude controllers | [-1, 1] |
| \_yaw\_in | Desired yaw control from attitude controllers | [-1, 1] |
| \_throttle\_in | last throttle input from set\_throttle caller | [0, 1] |
| 输出  motor\_output[4] | - | final pwm values sent to the motors | [1100,  1900] |
| 中间变量 | \_thrust\_rpyt\_out[4] | throttle outputs to motors, dimention 4 | [0, 1] |

计算方法：

\_thrust\_rpyt\_out =MotorMatrix\*thrust

motor\_output=1100+800\*\_thrust\_rpyt\_out

## Stabilize模式

控制流程和算法总体和C版飞控一致，故不详述。

## Althold高度控制

定高模式下，控制中长度相关单位（位移，速度，加速度）为了与C版飞控一致，统以用cm作为单位。

控制流程和算法总体和C版飞控基本一致，故不详述。

## Poshold位置控制

位置控制没有参考C版飞控进行设计，故说明设计方法。

### 控制框图

位置控制输入期望速度向量V和偏航角，飞机按照期望作动，Poshold控制框图如下：



### 姿态转换

主要对上图中姿态转换部分进行说明。

姿态转换：根据期望加速度，计算得到相应的姿态。

将姿态对加速度的影响作用抽象为加速度分配矩阵。定义加速度分配矩阵AccMatrix如下：

|  |  |  |
| --- | --- | --- |
| AccMatrix | Axfactor（body fixed） | Ayfactor（body fixed） |
| Roll | AccMatrix (1,1) | AccMatrix (1,2) |
| Pitch | AccMatrix (2,1) | AccMatrix (2,2) |

暂时根据需要定义2x2的AccMatrix矩阵，以后有必要参考高度控制，整合成3x3的AccMatrix。

AccMatrix可根据MotorLayout自动计算得到，暂时根据目前动力布局计算得：



1. 有头模式

有头模式下，期望速度是以（在四轴物理模型部分中进行了定义说明）坐标系为参考，故现将其转换为NED坐标系（因为反馈信号以NED坐标系为参考）下的期望速度：



其中是矩阵（在四轴物理模型部分中进行了定义说明）的xy平面旋转部分，

故

有头模式下，期望加速度是以坐标系为参考，故：



（2）无头模式

无头模式下，期望加速度是以NED坐标为参考，故：



# 3DVisualization

3维可视化建模。根据旋转矩阵DCM和大地坐标系下的位置P对飞行器动态过程模拟仿真。

Simulink 3D Animation将simulink信号通过三维图形的方式进行展示和互动，可以形象化的看到系统对象在虚拟世界的动态。模型用VRML(Virtual Reality Modeling Language)建立，这是一个标准的3D建模语言；模型通过虚拟现实浏览器（如Matalb的VR Sink）就可以进行动态演示。

总体系统调用3DVisualization进行动态演示，这里不做实现方法的具体说明。

## 模型演示

1. 说明输入期望
2. 看仿真曲线
3. 说明机头方向
4. 看演示

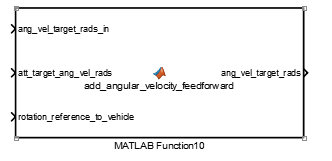
QuadCopter 视角演示（可以放大）

Vista\_vert视角演示 下方是机头

# 附录一 模块库常用模块说明

库文件内模块的命名和变量命名基本和C版飞控一致。

对库文件中的常用模块进行简单说明。很多模块用的次数较少，为了方便管理将其加入到模块库中，比如如下模块：



模块用于将前馈加入到目标角速度中，这个模块只要在姿态控制器中被调用，但是姿态控制器又被各个飞行模式都调用，为了方便跟新管理，将其加入库中，但是在其他模块中不适用这个模块。对于这种模块暂时不做说明。

## Lib\_AP\_Math

|  |  |  |
| --- | --- | --- |
| 序号 | 模块 | 功能 |
| 1 |  | 返回与旋转矩阵m对应的四元数qout |
| 2 |  | 四元素求逆 |
| 3 |  | 将欧拉角转化为四元数 |
| 4 |  | 四元数乘法 |
| 5 |  | 四元数转化为对应的旋转矢量 |
| 6 |  | 将四元素转化为对应的欧拉角 |
| 7 |  | 将角度转化为弧度 |

## Lib\_AC\_PID

|  |  |  |
| --- | --- | --- |
| 序号 | 模块 | 功能 |
| 1 |  | 计算P控制量 |
| 2 |  | 计算I控制量 |
| 3 |  | 计算D控制量 |
| 4 |  | 一阶低筒滤波器 |
| 5 |  | PID控制器 |
| 6 |  | 可分别输出P、I、D控制量的PID控制器。 |

## Lib\_AC\_AttitudeControl

|  |  |  |
| --- | --- | --- |
| 序号 | 模块 | 功能 |
| 1 |  | 平方根控制器 |
| 2 |  | 将欧拉角的导数转化为机体坐标系下的旋转速度。 |

## Lib\_AP\_Motors

|  |  |  |
| --- | --- | --- |
| 序号 | 模块 | 功能 |
| 1 |  | 计算电机对姿态的影响因子 |

## Lib\_Sensors

|  |  |  |
| --- | --- | --- |
| 序号 | 模块 | 功能 |
| 1 |  | 加速度传感器 |
| 2 |  | 陀螺仪传感器 |
| 3 |  | 磁罗盘 |
| 4 |  | 简单磁罗盘传感器 |
| 5 |  | 简单陀螺仪传感器 |
|  |  | 简单加速度传感器 |
| 6 |  | 三维向量加入噪声模块 |

# 常用公式及其证明

欧拉公式

# 模型更新

## Plant-Airframe

Matlab help：

Airframe Trim and Linearize 模型线性化

Flat Earth to LLA位置与经纬高度的转换 （gps）

Drag Calculation 空气阻力

Thrust Dynamics 推力动态建模

Bus setup 简化

## Environment

WGS84 Gravity Model重力模型

大气模型（温度，空气流速，压力，密度）

地磁场模型

## Sensors

GPS

IMU