

1. 实验名称及目的

1.1. 实验名称

LQR 姿态控制器设计实验

1.2. 实验目的

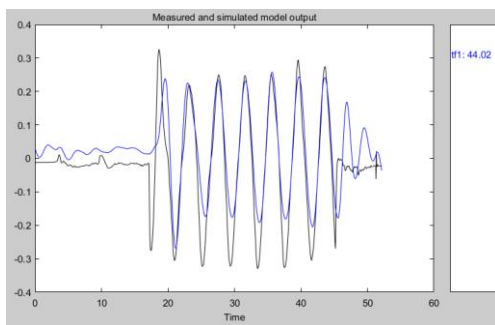
本实验将对四旋翼（以 FSJ150 飞机为例）的姿态作为控制目标，LQR 控制器设计实验包含有 HITL 及 QGC 调参。

1.3. 关键知识点

前置例程

实验开始前应首先完成四旋翼系统的辨识，获取对应通道（例如这里是滚转和俯仰通道）的传递函数及状态空间模型，具体实现参见例程 [..\..\4.RflySimModel\2.AdvExps\e13_CopIdModel\2.HILFreqId\Readme.pdf](#)。本实验基于如下辨识结果

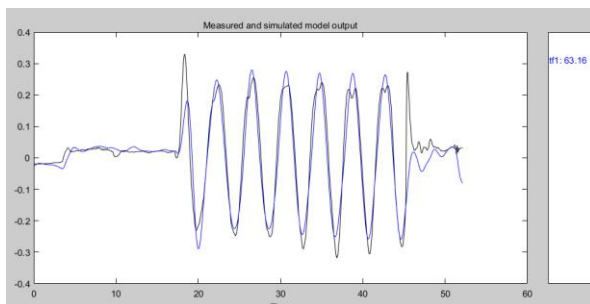
Roll:



$$s^2 + 0.589 s + 5.803$$

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -5.8 & -0.59 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 19.27 \end{bmatrix} M$$

Pitch:



$$s^2 + 1.365 s + 6.813$$

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1.36 & -6.81 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 20.54 \end{bmatrix} M$$

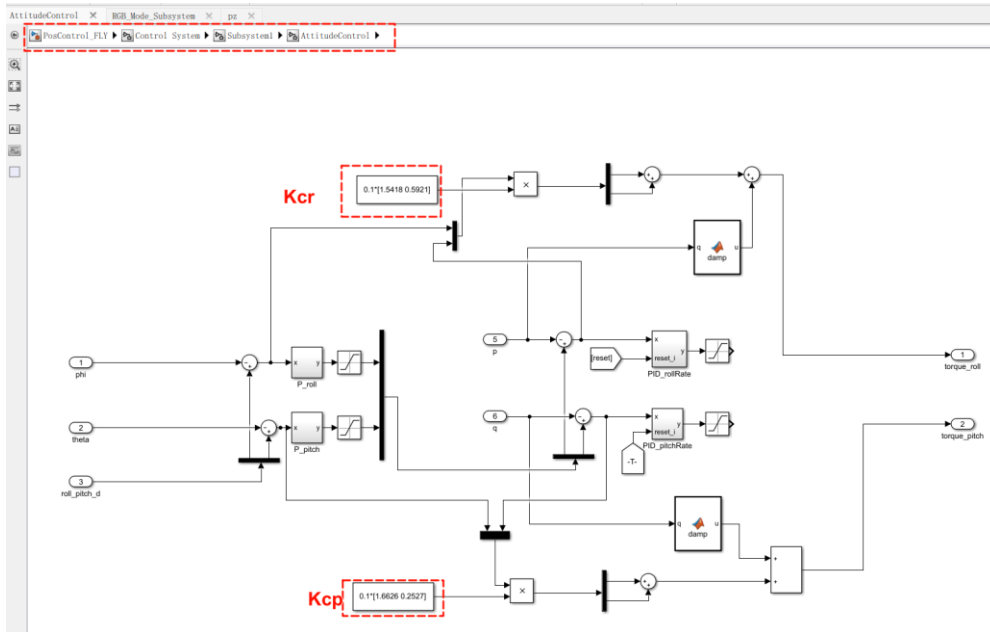
关键知识点 1: 搭建 LQR 姿态控制器

增益计算

计算增益的具体可见程序 [CodeGen HIL\LQR GainCom.m](#), 该程序运行之后计算出: 滚转通道 $K_{cr} = 1.5418 \quad 0.5921$; 俯仰通道 $K_{cp} = 1.6626 \quad 0.2527$

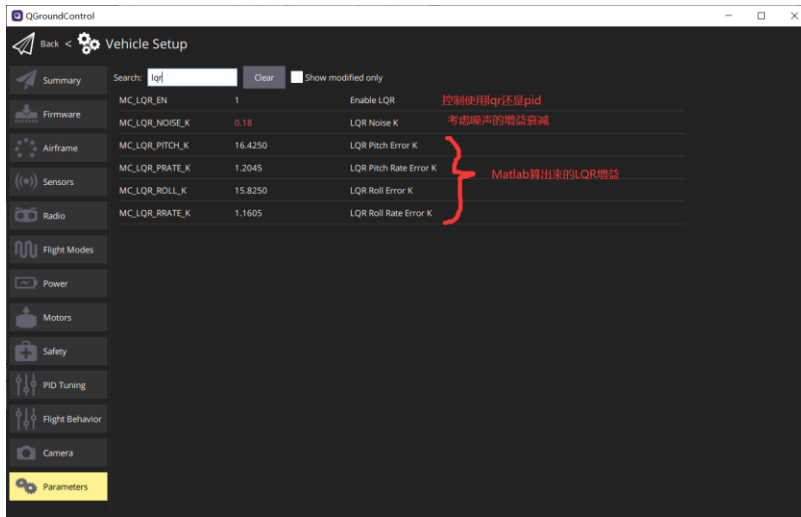
修改控制器

将计算得出的 K_{cr} 、 K_{cp} 分别输入: [CodeGen HIL\PosControl FLY.slx](#) 程序中的滚转和俯仰通道中, 如下图所示。其中, 乘以 0.1 系数是为了考虑到动力学模型中的噪声。



关键知识点 2: 实飞实验 PX4 代码修改

为了便于适用不同机型, 并支持用户自己辨识模型, 将 `lqr` 的参数通过 QGC 暴露给用户。在 QGC 中搜索 `lqr` 即可找到 `lqr` 相关的参数。需要修改对应通道的 PX4 源码, 并在 PX4 中添加自定义参数



LQR 核心代码修改

分别对俯仰和滚转通道应用 lqr，其核心是套用 $u=Kx$ 。角度误差使用的是期望角速率除以角度环的 P 参数 `_mc_roll_p` 和 `_mc_pitch_p`，这样就复用了 PX4 角度环的容错逻辑，以**最少的代码实现可靠的运算**。即 `err_roll` 和 `err_pitch` 的计算部分复用了 PX4 姿态环计算的结果，相当于将姿态环的结果作了逆向运算。**【如果直接读姿态 topic 算姿态误差，起飞时偶现不稳定的概率较大】**

套用 $u=Kx$ 公式时，`u_roll` 和 `u_pitch` 都乘以了考虑噪声影响的增益 `_lqr_noise_k`，对于卓翼和 pixhawk 飞机 `_lqr_noise_k` 取 0.2 附近的值能得到较好的效果。而滚转通道状态反馈系数 `_lqr_roll_k` 和 `_lqr_rollrate_k`，俯仰通道 `_lqr_pitch_k` 和 `_lqr_pitch_rate_k`，由 **MATLAB 计算得到**。最后，为了提高抵抗状态噪声的能力，对角加速度也添加了状态阻尼 **【这是当系统不容易稳定时，给系统增稳的常用方法，`_damping_roll` 和 `_damping_pitch` 是来自 PX4 原生参数】**。

```

Vector3f MulticopterRateControl::lqr(const Vector3f &rate, const Vector3f &rate_sp, const Vector3f &angular_accel){
    float u_roll;
    float u_pitch;

    // This is more stable than directly calculate attitude error
    float err_roll = rate_sp(0) / _mc_roll_p;
    float err_pitch = rate_sp(1) / _mc_pitch_p;

    Vector3f rate_error = rate_sp - rate;

    // LQR Error feedback
    u_roll = _lqr_noise_k * (err_roll * _lqr_roll_k + rate_error(0) * _lqr_rollrate_k);
    u_pitch = _lqr_noise_k * (err_pitch * _lqr_pitch_k + rate_error(1) * _lqr_pitchrate_k);

    // angular rate acc Damping, make system more stable
    u_roll -= _damping_roll * angular_accel(0);
    u_pitch -= _damping_pitch * angular_accel(1);

    return Vector3f(u_roll, u_pitch, 0.0f);
}

```

将 lqr 在角速度环中调用。使用一个标志 `_lqr_en` 控制是否启用 lqr。改算法可以和 PX4 原生的 pid 动态切换。当 `_lqr_en` 使能时，lqr 计算的结果将覆盖 PX4 角速度环输出的结果。

```
src > modules > mc_rate_control > MulticopterRateControl.cpp > Run()
132 MulticopterRateControl::Run()
266     actuators.control[actuator_controls_s::INDEX_YAW] = PX4_ISFINITE(att_control(2)) ? att_control(2) : 0.0f;
267     actuators.control[actuator_controls_s::INDEX_THROTTLE] = PX4_ISFINITE(_thrust_sp) ? _thrust_sp : 0.0f;
268     actuators.control[actuator_controls_s::INDEX_LANDING_GEAR] = _landing_gear;
269     actuators.timestamp_sample = angular_velocity.timestamp_sample;
270
271     // call lqr will override pid, this can be dynamically enabled or disabled
272     if (_lqr_en){
273         Vector3f torque = lqr(rates, _rates_sp, angular_accel);
274         actuators.control[actuator_controls_s::INDEX_ROLL] = PX4_ISFINITE(torque(0)) ? torque(0) : 0.0f;
275         actuators.control[actuator_controls_s::INDEX_PITCH] = PX4_ISFINITE(torque(1)) ? torque(1) : 0.0f;
276     }
277
```

在 PX4 中添加自定义参数

常用 PARAM_DEFINE_FLOAT 和 PARAM_DEFINE_INT32 分别定义浮点型参数和整型参数。除了 MC_LQR_EN 为整型，其它参数都为浮点型。LQR 的相关参数都定义在 mc_rate_control_params.c 中。值得特别说明的是，自定义的参数除了宏函数意外的内容，注释也会被显示在 QGC 中，因此注释也应按照样例规范编写。

```
C mc_rate_control_params.c 9+, M X
src > modules > mc_rate_control > C mc_rate_control_params.c > ...
386 PARAM_DEFINE_FLOAT(MC_ACRO_SUPEXPOY, 0.7f);
387
388 /**
389  * LQR Roll Error K.
390  *
391  * After attitude identification, adjust MC_LQR_ROLL_K for your plane.
392  * Can be calculated by MATLAB, as Kc = place(Ac, Bc, P).
393  *
394  * 0 No function
395  * 30.0 very strong vibration
396  *
397  * @min 0
398  * @max 30.0
399  * @decimal 4
400  * @group Multicopter Rate Control
401  */
402 PARAM_DEFINE_FLOAT(MC_LQR_ROLL_K, 1.621f);
403
404 /**
405  * LQR Roll Rate Error K.
406  *
407  * After attitude identification, adjust MC_LQR_RRATE_K for your plane.
408  * Can be calculated by MATLAB, as Kc = place(Ac, Bc, P).
409  */
```

为了在代码中使用这些参数，在需要使用这些参数的类中定义变量。

```
src > modules > mc_rate_control > MulticopterRateControl.hpp > MulticopterRateControl > _rate_control
65 class MulticopterRateControl : public ModuleBase<MulticopterRateControl>, public ModulePa
171     (ParamFloat<px4::params::MC_ACRO_EXPO>) _param_mc_acro_expo,
172     (ParamFloat<px4::params::MC_ACRO_EXPO_Y>) _param_mc_acro_expo_y,
173     (ParamFloat<px4::params::MC_ACRO_SUPEXPO>) _param_mc_acro_supexpo,
174     (ParamFloat<px4::params::MC_ACRO_SUPEXPOY>) _param_mc_acro_supexpo_y,
175
176     (ParamBool<px4::params::MC_BAT_SCALE_EN>) _param_mc_bat_scale_en,
177     (ParamBool<px4::params::MC_LQR_EN>) _param_mc_lqr_en,
178
179     (ParamInt<px4::params::CBRK_RATE_CTRL>) _param_cbrk_rate_ctrl,
180     (ParamFloat<px4::params::MC_LQR_ROLL_K>) _param_mc_lqr_roll_k,
181     (ParamFloat<px4::params::MC_LQR_RRATE_K>) _param_mc_lqr_rollrate_k,
182     (ParamFloat<px4::params::MC_LQR_PITCH_K>) _param_mc_lqr_pitch_k,
183     (ParamFloat<px4::params::MC_LQR_PRATE_K>) _param_mc_lqr_pitchrate_k,
184     (ParamFloat<px4::params::MC_LQR_NOISE_K>) _param_mc_lqr_noise_k,
185     (ParamFloat<px4::params::MC_ROLL_P>) _param_mc_roll_p,
186     (ParamFloat<px4::params::MC_PITCH_P>) _param_mc_pitch_p
187 )
```

支持动态修改的参数，还需在 `parameters_updated()` 函数中修改。

```

MulticopterRateControl::parameters_updated()
{
    // rate control parameters
    // The controller gain K is used to convert the parallel (P +
    // to the ideal (K * [1 + 1/sTi + sTd]) form
    const Vector3f rate_k = Vector3f(_param_mc_rollrate_k.get(),

    _lqr_en = _param_mc_lqr_en.get();
    _lqr_roll_k = _param_mc_lqr_roll_k.get();
    _lqr_rollrate_k = _param_mc_lqr_rollrate_k.get();
    _lqr_pitch_k = _param_mc_lqr_pitch_k.get();
    _lqr_pitchrate_k = _param_mc_lqr_pitchrate_k.get();
    _damping_roll = _param_mc_rollrate_d.get();
    _damping_pitch = _param_mc_pitchrate_d.get();
    _mc_roll_p = _param_mc_roll_p.get();
    _mc_pitch_p = _param_mc_pitch_p.get();
    _lqr_noise_k = _param_mc_lqr_noise_k.get();
}

```

2. 实验效果

实现四旋翼姿态控制。

3. 文件目录

文件夹/文件名称		说明
CodeFly_support y aw	mc_rate_control_params.c	
	MulticopterRateControl.cpp	
	MulticopterRateControl.hpp	
CodeGen_HIL	icon	图片等其他文件。
	HIL_FLY_CFO.slx	LQR 控制器 HITL 仿真和实飞模型文件
	Init_control.m	控制器参数文件
	LQR_GainCom.m	计算 LQR 系数

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 飞控 ^②	1
3	MATLAB 2022b 及以上	遥控器 ^③	1
4		飞思 X450 飞机 ^②	1
5		遥控器接收器	1
6		数据线、杜邦线等	若干

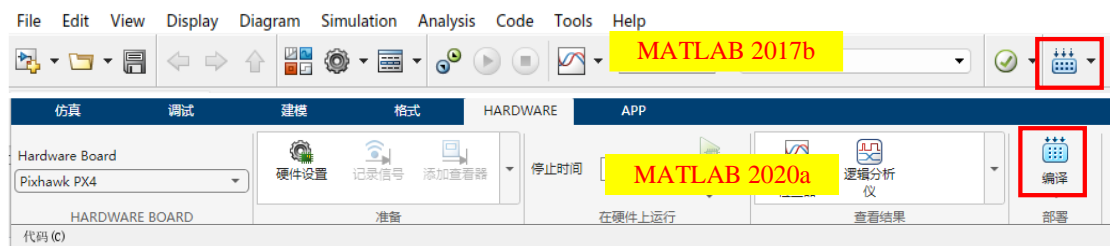
7		SD卡及读卡器	1
---	--	---------	---

- 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_fmuv6x_default, 固件版本为: 1.12.3。其他配套飞控请见: <http://doc.rflysim.com/hardware.html>
- ③ : 本实验演示所使用的遥控器为: 天地飞 WFLY-ET10、配套接收器为: WFLY-RF20 9S。遥控器相关配置见: <https://rflysim.com/doc/zh/B/3.1ET10.html>

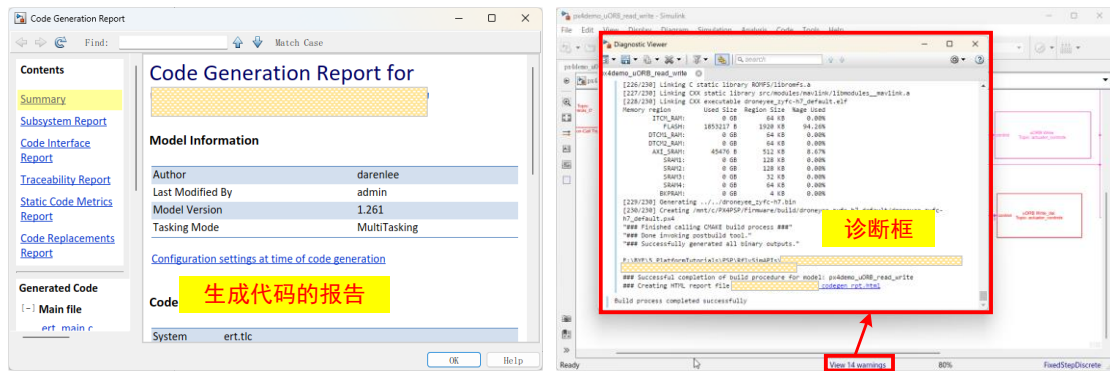
5. 实验步骤

5.1. HITL 实验

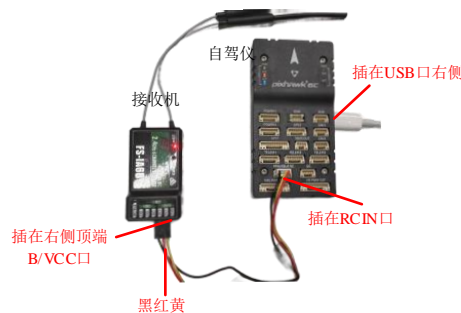
在 MATLAB 中打开 [CodeGen_HIL\Init_control.m](#) 文件并运行, 等待运行完成之后, 打开 [CodeGen_HIL\PosControl_FLY.slx](#) 文件, 点击 Simulink 中的编译按钮。

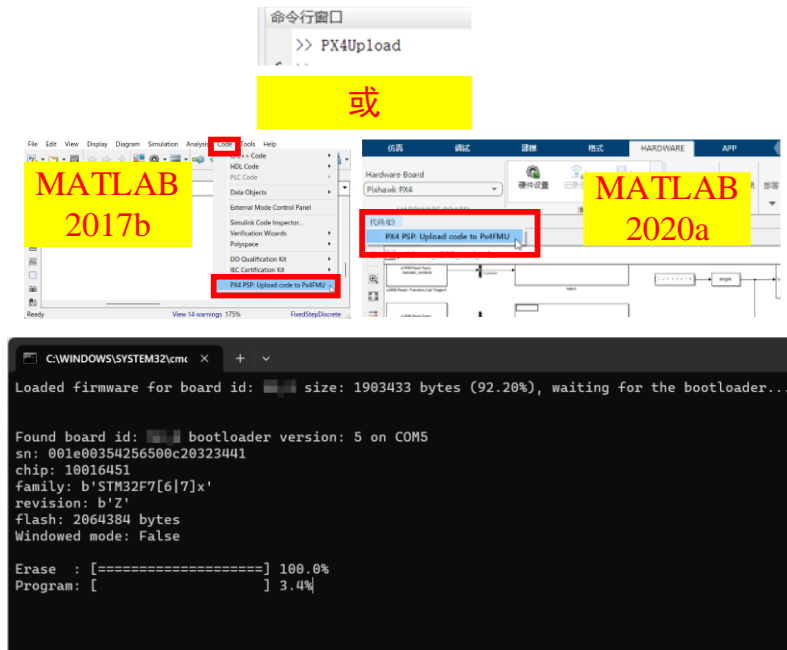


在 Simulink 的下方点击 View diagnostics 指令, 即可弹出诊断对话框, 可查看编译过程。在诊断框中弹出 Build process completed successfully, 即可表示编译成功, 左图为生成的编译报告。

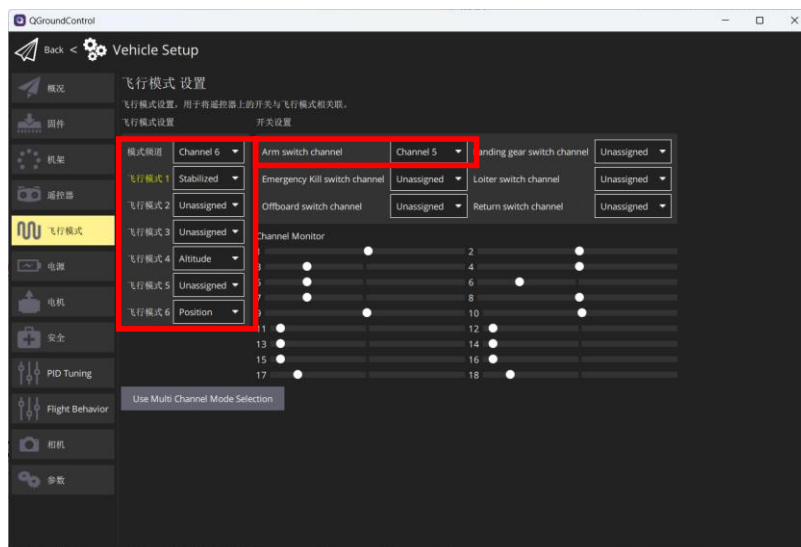
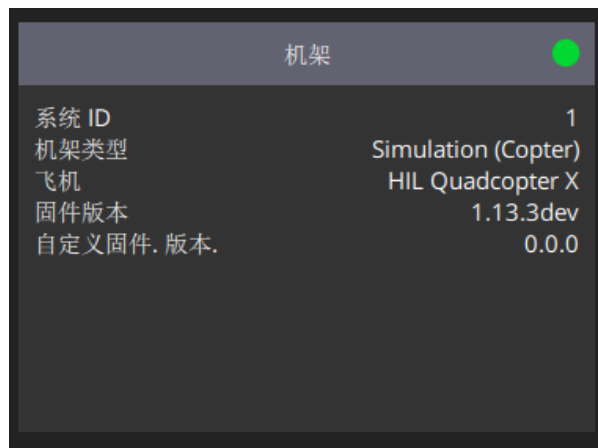


用 USB 数据线链接飞控与电脑。在 MATLAB 命令行窗口输入: PX4Upload 并运行或点击 PX4 PSP: Upload code to Px4FMU, 弹出 CMD 对话框, 显示正在上传固件至飞控中, 等待上传成功。





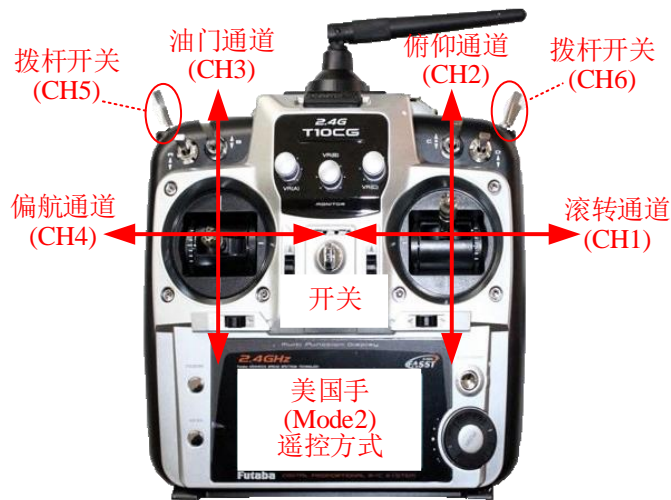
上传成功后，打开 QGroundControl 软件，确认为如下设置：



双击打开 [CodeGen_HIL/HITLRun.bat](#) 文件，在弹出的 CMD 对话框中输入插入的飞控 COM 端口号，即可自动启动 RflySim3D、CopterSim、QGroundControl 软件，等待 CopterSim

的状态框中显示：PX4: GPS 3D fixed & EKF initialization finished。即可在 QGroundControl 中设置飞机起飞等操作。

遥控器的设置如下图，通过控制不同的通道即可在 RflySim3D 中观察到无人机的飞行姿态，完成硬件在环仿真。**注：具体设置请见本平台的[遥控器配置手册](#)。**



通过 CH5 解锁之后，在 RflySim3D 中即可看到飞机正常起飞，通过 Step 4 中 CH1~CH4 调整飞机姿态和高度。

5.2. QGC 调参实验

Step1: 替换 PX4 源码

将 [CodeFly_support_yaw](#) 目录下修改好的源码替换到[安装目录]\Firmware\src\modules\m_c_rate_control 下

Step2: 重新编译固件并烧录

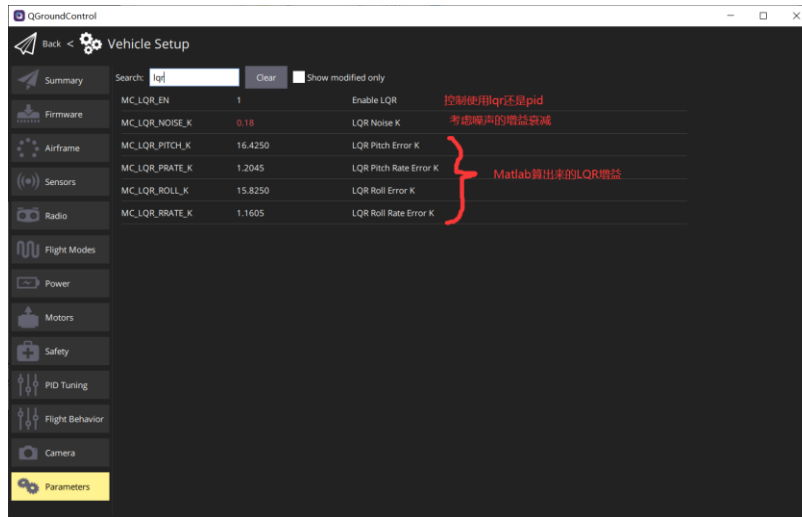
整个删除[安装目录]\Firmware\build 文件夹，在桌面 RflyTools 中打开 WinWSL 终端，输入 `make px4_fm_u-v6c_default` 编译固件

```
root@Rfly: /mnt/c/PX4PSP/Fir  X  +  v
root@Rfly:/mnt/c/PX4PSP/Firmware# make px4_fm_u-v6c_default
```

通过 QGC 烧录编译好的固件，具体参见[..\..\..\2.RflySimUsage\1.BasicExps\10_Firmware-Upload](#)

Step3: QGC 调参

启动 [CodeGen_HIL\HITLRun.bat](#)，在在 QGC 参数页面中搜索 `lqr` 即可找到 `lqr` 相关的参数。在实际中，需要手动微调的参数只有 `MC_LQR_NOISE_K`。



6. 参考资料

- [1]. 全权,杜光勋,赵峙尧,戴训华,任锦瑞,邓恒译.多旋翼飞行器设计与控制[M],电子工业出版社, 2018.
- [2]. 全权,戴训华,王帅.多旋翼飞行器设计与控制实践[M],电子工业出版社, 2020.

7. 常见问题

Q1: ***

A1: ***