

Lesson 03 Experimental Platform Usage

Dr. Xunhua Dai, Associate Professor, School of Computer Science and Engineering, Central South University, China; Email: dai.xh@csu.edu.cn ; https://faculty.csu.edu.cn/daixunhua

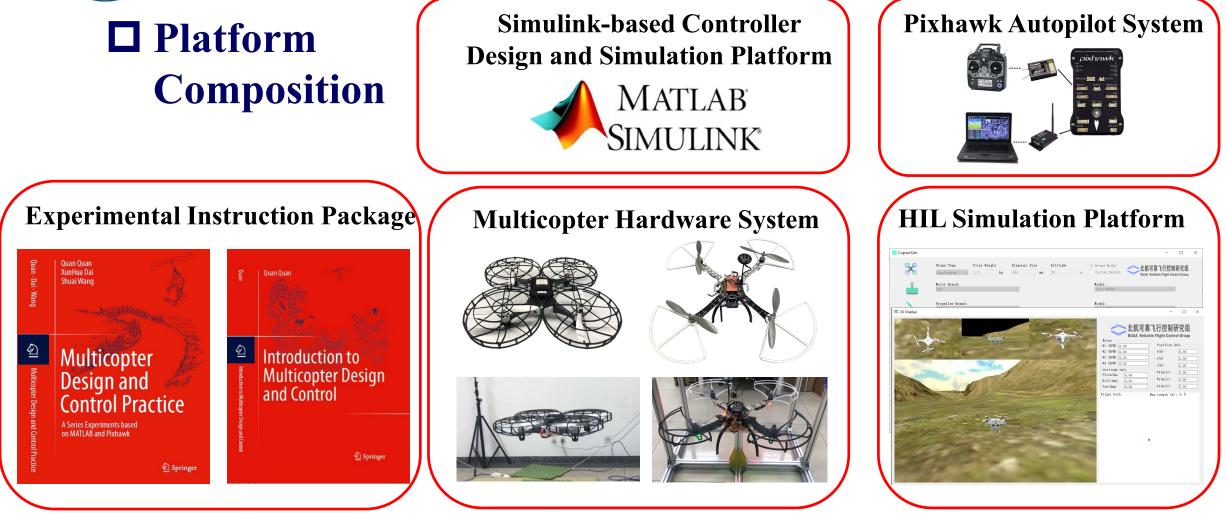




- 2. Controller Design and Simulation Platform
- 3. PSP Toolbox
- 4. Pixhawk Hardware System
- 5. HIL Simulation Platform
- 6. Summary





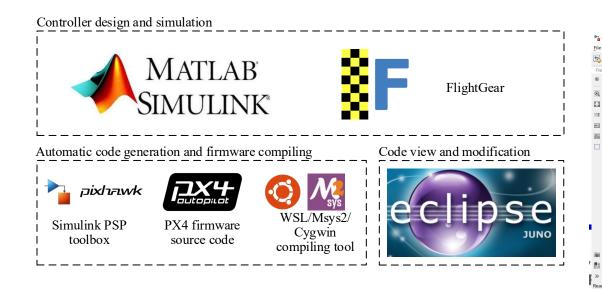


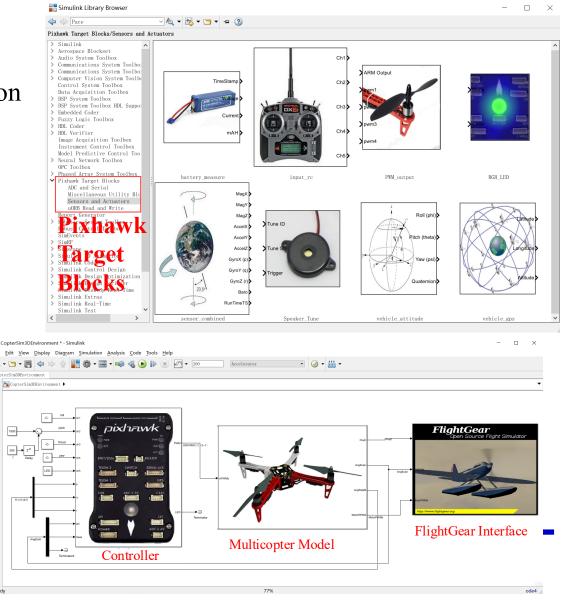




D Platform Composition

- (1) Simulink-based Controller Design and Simulation Platform
- A high-fidelity nonlinear model
- Modular controller design
- A real-time 3D display for the flight status
- Automatic code generation and upload function



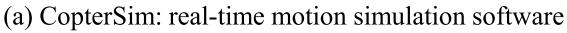




D Platform Composition

(2) HIL Simulation Platform

🍔 CopterSim						- 🗆 ×
Q P C D	Frame Type Quadcopter V	Total Weight 1.5 kg	Diagonal Size			航可靠飞行控制研究组 A Reliable Flight Control Group
_	Motor Brand: DJI		~		Model: 2312 KV960	~
	Propeller Brand: APC		~		Model: 10x4.5MR	~
5	ESC Brand: Hobbywing		~		Model: XRotor 20A	~
	Battery Brand: ACE		V		Model: LiP⊙ 3S-11.1V-25C-5500mÅh	~
Assembled UAV Datab	ase:		Activa	Model Para	ameter Add to Database	Delete from Database
Vehicle ID: UDP Por 1 14560	t: Use DLL Model	: Simulation ∨ PX4≪1.7		Scene Selection: DisplayProgram	Link Vehicle Initia	
Select Pixhawk Com	: Legacy FMU CO	DM6	~	Start Simulation	Stop Simulation	Restart Simulation
				XO	Υ 0	Ζ 0
				Vx 0	Vy 0	Vz 0
				Φ 0	θ	ψ







#1 (RPM)	0.00	Position I	nfo
#2 (RPM)	0.00	x (n)	0.00
#3 (RPM) [0.00	y(n)	0.00
#4 (RPM) [0.00	z(n)	0.00
Attitude : Pitch(de:		$\forall x(n/s):$	0.00
Roll(deg)	0.00	$\forall y(n/s):$	0.00
Yaw(deg)	0.00	$V_{\mathbb{Z}}(n/s)$:	0.00
light Path	1	Map Length (m)	: ± 5

(b) 3DDisplay: 3D visual display software





D Platform Composition

- (3) Pixhawk Autopilot System
- 1) Pixhawk Autopilot
- 2) RC transmitter and receiver
- 3) A microUSB cable
- 4) Ground Control Station (GCS)







D Platform Composition

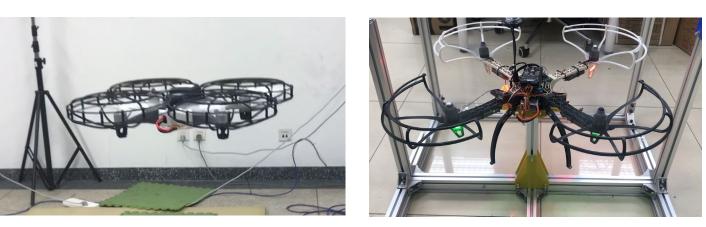
(4) Multicopter Hardware System

1) Airframe system

- 2) Propulion system
- 3) External sensors
- 4) Test stand









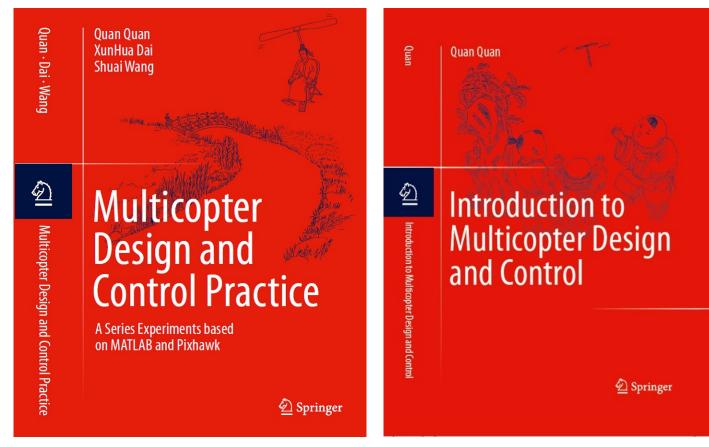


D Platform Composition

(5) Experimental Instruction Package

1) Textbook

- 2) Experiment instructor
- 3) Source code examples
- 4) Video tutorials







D Platform Advantage

- This experimental platform provides interfaces for multicopter controller design in MATLAB/Simulink
- Readers (beginners, students, or engineers) can develop a rapid design and verify the control algorithms by SIL simulation.
- Platform also provides a code generation function to generate Simulink controllers to C/C++ code.
- Platform also provides a HIL simulation platform for preliminary simulation tests on a Pixhawk autopilot system that may help in eliminating potential problems that may exist in flight tests.
- After all the tests are completed, indoor and outdoor flight tests can be carried out by assembling the Pixhawk autopilot onto a real multicopter hardware system. The performance of the designed controllers can be evaluated through experimental tests.





Controller Design and Simulation Platform

This book provides a high-fidelity simulation environment based on Simulink/FlightGear. The main source code file is presented in "e0\1.SoftwareSimExps\CopterSim3DEnvironment.slx".

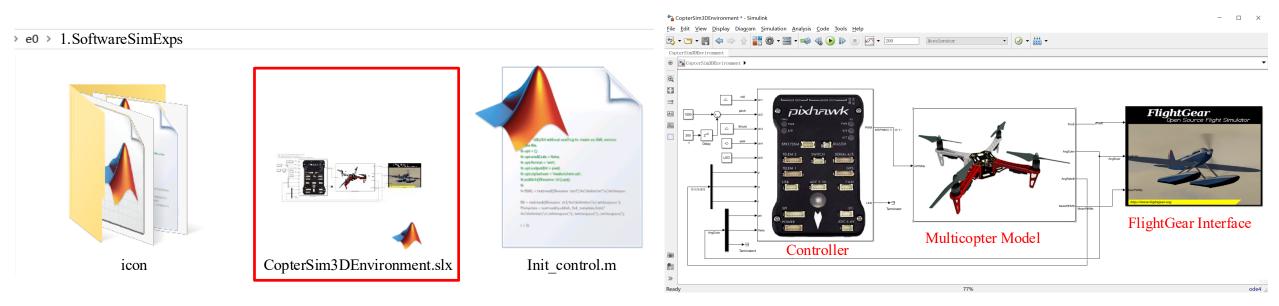


Fig. SoftwareSimExps file diagram

Fig. Simulink SIL simulation example





Controller Design and Simu

Controller

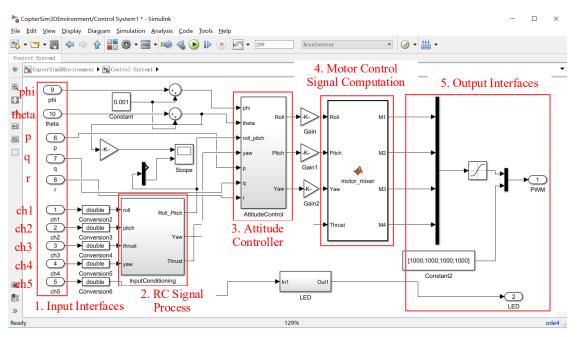


Fig. Internal structure of "Controller" subsystem

As shown in left figure, this example shows a simple attitude controller for pitch and roll angles. The controller receives the control signals from the RC transmitter and controls the multicopter to achieve the desired pitch and roll angles.

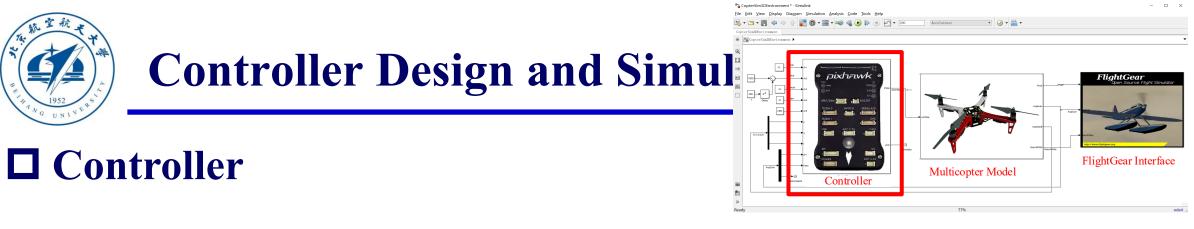
nixhaw

Multicopter Mode

FlightGear Interface

- 1. The "Input Interfaces" module receives the RC transmitter signals and the multicopter state estimation signals
- 2. The "RC Signal Process" module maps the five-channel signals of the RC transmitter to the desired roll and pitch angle values.
- 3. The "Attitude Controller" module computes the desired force and torque values to control the multicopter to the desired attitude.





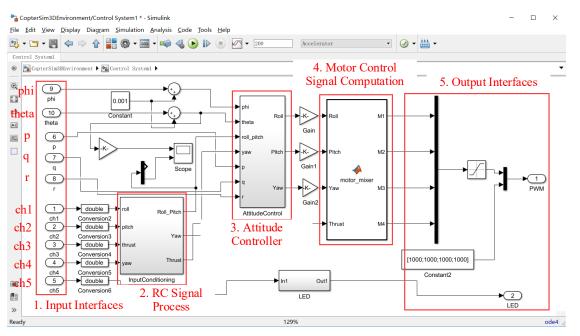


Fig. Internal structure of "Controller" subsystem

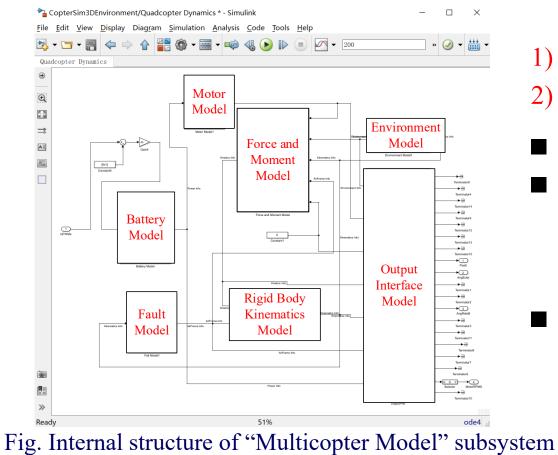
- 4. The "Motor Control Signal Computation" module maps the force and torque values to the control signals (ranging from 1000 to 2000) for the four motors
- 5. The "Output Interfaces" module fills the remaining 4-dimensional control signals and generates an 8-dimensional PWM signal (there are eight PWM output ports on Pixhawk) ranging from 1000 to 2000.





Controller Design and Simul

D Multicopter Model



- 1) Inputs: motor PWM controls
 2) Outputs: flight state and sensor signals
 - "Motor Model" module: it simulates the motor dynamics.

nixhavv

Controlle

Multicopter Model

FlightGear

FlightGear Interface

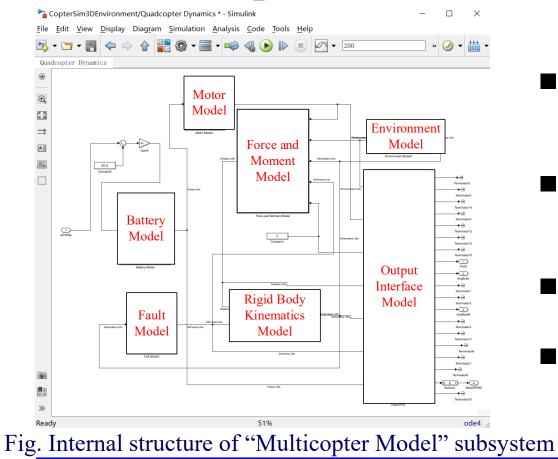
- "Force and Moment Model" module: it simulates all external forces and moments acting on the body, such as the propeller thrust, fuselage aerodynamics, gravity, and ground supporting force.
- "Rigid Body Kinematics Model" module: it calculates the vehicle kinematics of the multicopter, such as speed, position, and attitude.

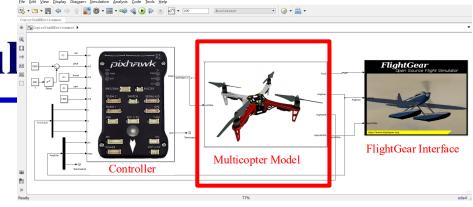




Controller Design and Simul

Multicopter Model





- "Environmental Model" module: it calculates the environmental data, such as gravitational acceleration, air density, wind disturbances, and geomagnetic field
- "Fault Model" module: it is mainly used to inject model uncertainties (related to mass and moment of inertia) as well as faults.
- "Battery Model" module: it simulates the discharge process of the battery.
- "Output Interface Model" module: it packs the output signals in the desired format.





Controller Design and Simul

☐ FlightGear Interface

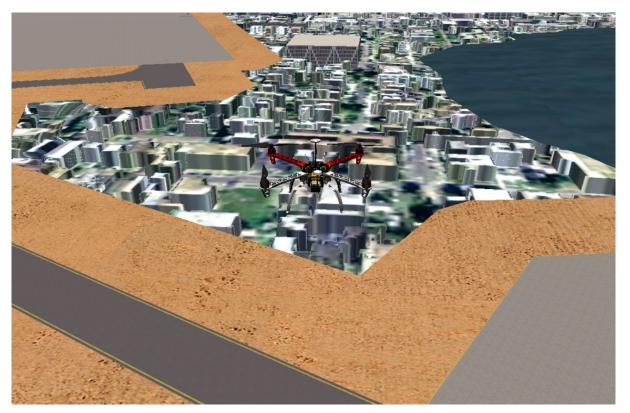
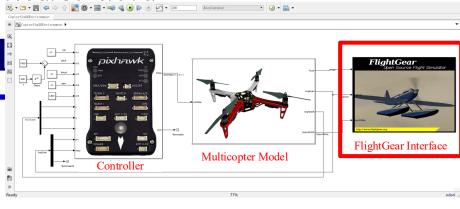


Fig. A quadcopter in FlightGear



1) Function

"FlightGear Interface" subsystem has three input ports representing the multicopter position, Euler angles, and motor PWM signals, respectively. This subsystem sends multicopter flight state information to FlightGear to observe the flight attitude and trajectory of the multicopter in a 3D scene.



NopterSim3DEnvironment * - Simulink

Edit View Display Diagram Simulation Analysis Code Tools Help



Controller Design and S - 🚬 🗕 🦳

□ FlightGear Interface

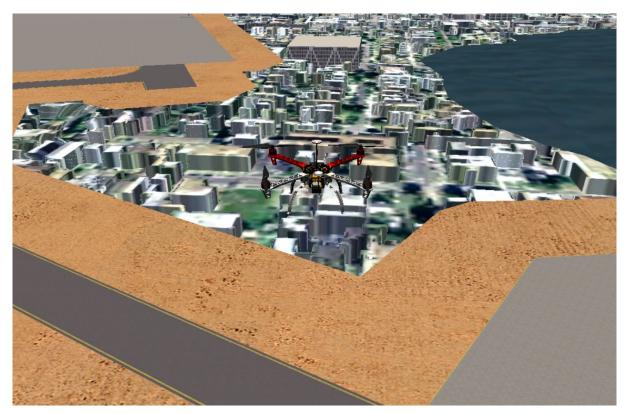
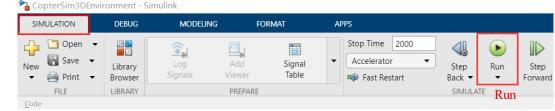


Fig. A quadcopter in FlightGear

(a) Simulink "Run" button on MATLAB 2017b-2019a



(b) Simulink "Run" button on MATLAB 2019b and above

Note: if FlightGear cannot display properly, you 2) Effect can open 3DDisplay to observe the flight effect

1. Double-click the FlightGear-F450 shortcut on the desktop to open FlightGear;

2. Click the "Run" button

on the Simulink

111 190

toolbar to run the "CopterSim3DEnvironment.slx" file;

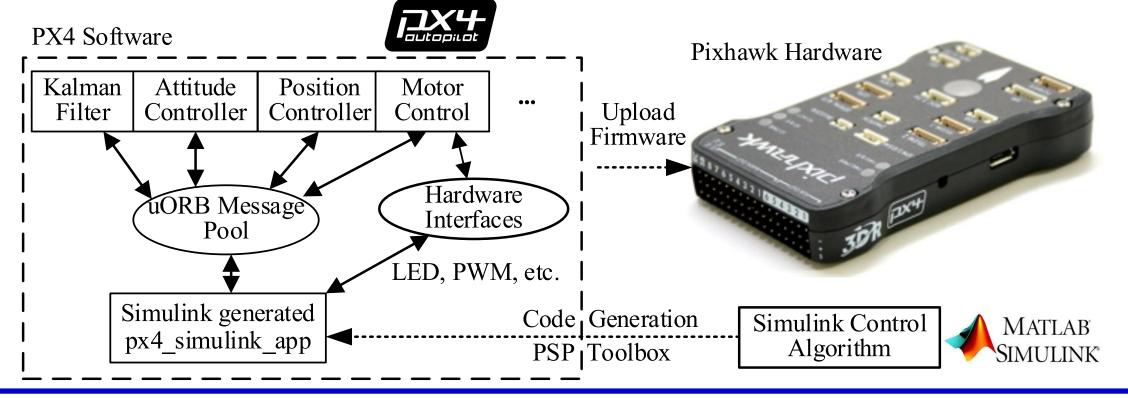
3. Then, the multicopter takes off vertically from the ground. You can slide the slider modules in Simulink to simulate control the roll, pitch, yaw, throttle channels of the controller with an RC system.

CH1_ROLL :Value	CH2_PITCH :Value	CH3_THROTTLR :Value	CH4_YAW :Value	Ch5_MODE :Value
100 1250 1400 1550 1700 190(1100 1250 1400 1550 1700 1900	1100 1250 1400 1550 1700 1900	1100 1250 1400 1550 1700 190(100 1250 1400 1550 1700



The figure below shows the relationship among the PSP toolbox, the PX4 software, and the Pixhawk hardware. Pixhawk Pilot Support Package (PSP) Toolbox s a Simulink toolbox officially released by Mathworks.

This toolbox can use the Embedded Coder in Simulink to automatically compile and deploy Simulink model autopilot algorithms to Pixhawk hardware systems.







D Main Functions

1. The toolbox can simulate and test different multicopter models and flight control algorithms in Simulink and then automatically deploy the algorithms to the Pixhawk autopilot.

2. The toolbox provides many practical examples, including LED control, RC data process, and attitude controller.

- 3. The toolbox provides many interface modules to access the Pixhawk hardware and software components.
- 4. It automatically records flight data from sensors, actuators, and controllers deployed by themselves.

5. It can subscribe and publish uORB topic messages. All messages in the PX4 autopilot software are temporarily stored in a uORB message pool. The subscription function can read topics of interest from the message pool, and the publishing function can publish specific topics to the message pool for other modules.





C Relationship with Pixhawk system

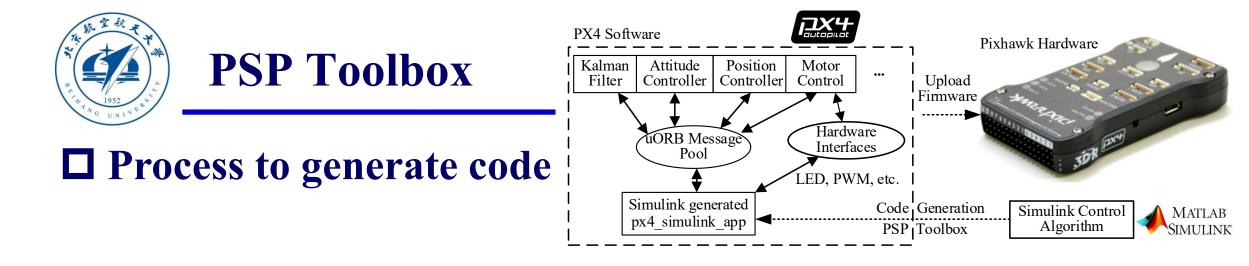
1. The structure of a Pixhawk autopilot system includes two parts: the Pixhawk hardware (similar to the computer hardware) and the PX4 software (similar to operating system and applications on a computer).

2. The PX4 software system can be further divided into several small modules, which run independently in parallel multi-thread. Each module exchanges data with other modules through uORB messages.

3. After the algorithm code is generated by the PSP toolbox, it is embedded into the PX4 software system. This will not affect the operation of the native control modules in the PX4 software. Instead, a new independent module named "px4_simulink_app" will be created to run in parallel with other modules.

5. The native modules in the PX4 software may assess the same hardware outputs as the generated "px4_simulink_app" module, which may cause read and write conflicts. Therefore, in the one-key installation script, the hardware output accessing codes of the PX4 native modules have to be blocked. This will ensure that only the "px4_simulink_app" module can send motor control signals.





(a) The PSP toolbox generates the C/C++ code from the control algorithm designed in Simulink.

(b) The obtained algorithm code is imported into the PX4 source code to generate a "px4_simulink_app" independent of other modules.

(c) The PSP toolbox calls the compiling toolchain (Win10WSL, Msys2, or Cygwin) to compile all the code into a ".px4" PX4 firmware file (similar to a software installation package).

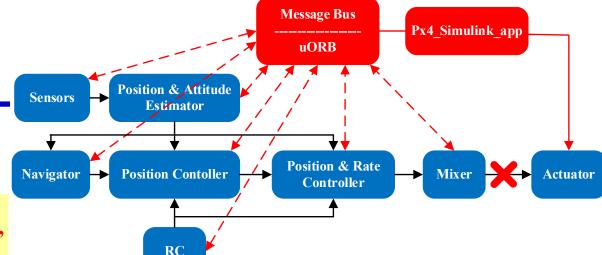
(d) Upload the obtained firmware file to the Pixhawk hardware; then, the Pixhawk autopilot can execute the PX4 software with the generated algorithm code.





Advanced extension features

Note: Another feasible way is to block the module in the startup script "Firmware\ROMFS\px4fmu_common\init.d\rcS"



The generated Simulink code can also be used to replace some of the native modules (sensors, filters, attitude controllers, etc.) of the PX4 software. However, the PX4 software code needs to be manually modified to block the output interface of the corresponding native module. For example, if readers want to use Simulink to replace the attitude filter module (input sensor data, output filtered attitude data) of the PX4 software, they should manually block uORB publishing code of the "vehicle_attitude" msg in the "Position & Attitude Estimator" module. The detailed steps are described next.

(a) Open the "Firmware\src\modules\ekf2\ekf2_main.cpp" file (corresponding to the code of the extended Kalman filter module).

(b) Search the text "_att_pub", and comment out the lines containing "publish" and "att", and replace them with "UNUSED(att);". The UNUSED is used to suppress warning of "variable not used" of compiler.

(c) Create a Simulink controller to publish "vehicle_attitude" message with "uORB Write" PSP module.



D Simulink Pixhawk Target Blocks

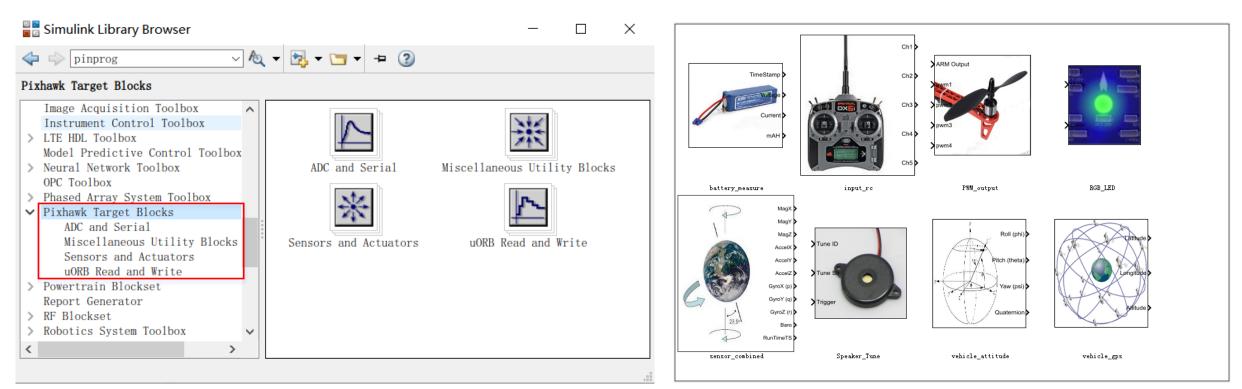


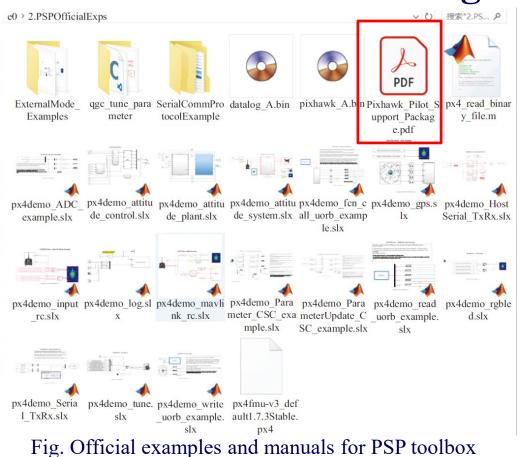
Fig. Simulink Pixhawk Target Blocks of PSP toolbox

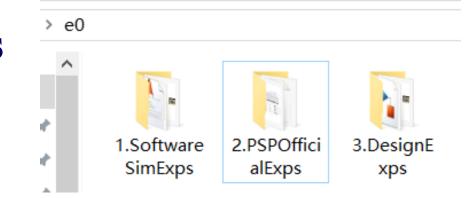
Fig. interfaces of Sensors and Actuators





Gimulink Pixhawk Target Blocks





the PSP toolbox also provides many examples (see folder "e0\2.PSPOfficialExps") with an official manual (see document

"e0\2.PSPOfficialExps\Pixhawk_Pilot_Support_Packag e.pdf" for details) for readers to be quickly familiar with functions and usage methods of PSP toolbox.





Modules in PSP Toolbox

PWM output module : it is used to send PWM signals to PX4IO ports to control the motor. The PWM update frequency and the number of output channels can be configured in the setting box.

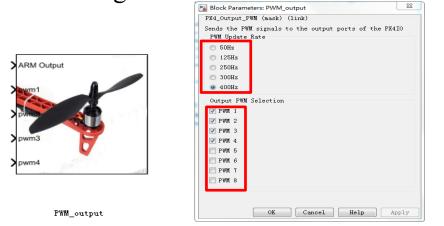


Fig. PWM output module and its parameter

RC input module: It is convenient to select RC channels and other information to be used by Simulink. The definition and application of each option can be viewed by clicking the "help" button of the box or by consulting the official PDF document. The PSP toolbox also provides an example (see file "e0\2.PSPOfficialExps\px4demo_input_rc.slx") to show how to use this module.

		📓 Block Parameters: input_rc 📃
	Ch1	PX4_Input_RC (mask) (link)
		Receiver Inputs from the Pixhawk hardware
		Sample Time
1	Ch2	1/250
di la		Channel Selection
	01-2	🕼 Channel 1 🕼 Channel 2 📝 Channel 3 📝 Channel 4
	Ch3 >	🔽 Channel 5 🔲 Channel 6 📄 Channel 7 📄 Channel 8
		Optional Outputs
Tint. T.	Ch4	Channel Count 🔲 RSSI
		🗖 RC Failsafe 📃 RC Lost Connection
	Ch5 🕨	RC Input Source
		OK Cancel Help Appl
input_rc		

Fig. RC input module and its parameter setting





Modules in PSP Toolbox

Buzzer module: it is used when the buzzer is required to make a warning sound. There is an example (see file "PX4PSP\examples\px4demo_tune.slx") for detailed information.

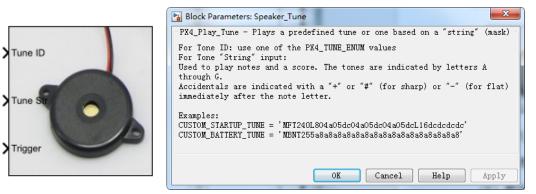


Fig. Buzzer module and its parameter setting box

RGB_LED module: this module can control the blink mode and color of the LED on Pixhawk. The module receives two inputs, namely "Mode" and "Color" representing the mode and color of the LED. The PSP toolbox provides an example (see file "e0\2.PSPOfficialExps\px4demo_rgbled.slx") to study this module.



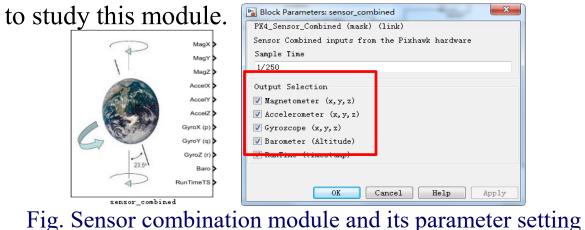
Fig. LED light module and its parameter setting box





D Modules in PSP Toolbox

Sensor combination module: This module can access the sensor data available in the Pixhawk autopilot, which can then be used for controller design in Simulink. Available sensor data include magnetometers, accelerometers, gyroscopes, barometers, and timestamps. The PSP toolbox also provides an example (see file "e0\2.PSPOfficialExps\px4demo_attitude_system.slx")



Attitude data module: it provides an interface to access the attitude estimate (Euler angles and quaternion). The PSP toolbox also provides an example (see file "e0\2.PSPOfficialExps\ px4demo_attitude_system.slx") to study this module.

, ,	Block Parameters: vehicle_attitude				
1	PX4_Vehicle_Attitude (mask) (link)				
Roll (phi)	Vehicle Attitude measurements from the Pixhawk hardware				
	Sample Time				
Pitch (theta)	1/250				
	Output Signal Selection				
Yaw (psi)	🔽 Roll				
1 1	V Pitch				
	🔽 Yaw				
Quaternion	📝 Quaternion (NED)				
,					
vehicle_attitude	OK Cancel Help Apply				

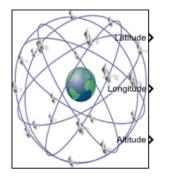
Fig. Attitude data module and its parameter setting





D Modules in PSP Toolbox

GPS data module: it can be used to access the Pixhawk GPS data, which are achieved by subscribing to the uORB topic "vehicle_gps". Therefore, in practical operation, it is necessary to ensure that the GPS module is inserted into the Pixhawk hardware and then works.



Block Parameters: vehicle_gps
PX4_Vehicle_GPS (mask) (link)
Return Values from the GPS topic running on the Pixhawk.
Sample Time
1/250
Output Selection
🕅 Position Timestamp
📝 Altitude
🔲 GPS Timestamp
Telocity
🔄 Fix Type
Number of Satellites
OK Cancel Help Apply

vehicle_gps

Fig. GPS data module and its parameter setting

The PSP toolbox also provides an example (see file "e0\2.PSPOfficialExps\px4demo_gps.slx") to

study this module.

Battery data module: it can be used to obtain the real-time status of the battery. It is implemented by subscribing to the uORB topic "battery_status". Therefore, in practical operation, it is necessary to ensure that the power module is inserted into the Pixhawk hardware and then works correctly.

	Block Parameters: battery_measure		
TimeSterre	PX4_Battery_Measurement		
TimeStamp >	Returns battery measurements from Pixhawk hardware		
	Sample Time		
Voltage	[1/250]		
Current	Parameters		
	Voltage		
mAH 🕨	🔲 Filtered Voltage		
	🕼 Current		
	🕼 mAH		
	📝 Time Stamp		
battery_measure	OK Cancel Help Apply		

Fig. Battery data module and its parameter setting





In fact, almost all modules in PSP toolbox can be released by uORB module. We can use uORB module to access and publish all data in Pixhawk/PX4, which can help us achieve more complex functions. Please see https://dev.px4.io/master/en/middleware/uorb.html

Modules in PSP Toolbox

uORB modules. These modules, presented in left figure, are used to read or write uORB messages from the PX4 autopilot software. All the uORB message topics supported by the PX4 autopilot are listed in the directory "Firmware\msg" of the software package installation directory (the default directory is "C:\PX4PSP"). Double-click the "uORB Write" module in the lower-left figure, then the obtained parameter setting box of the "uORB Write" module is presented in lower-right figure where the uORB topic name and the message variables to be sent can be configured.

Eunction Trigger			Block Parameters: uORB Write × S-Function (mask) (link) ×	
Function Trigger Topic ID: UNDEFINED	VORB Read - Function Trigger: 'differential_pressure'		u0RB Write Output Block Publishes to a user provided named tropic structure.	Fig. "uORB
uORB Read Async	uORB Read Function-Call Trigger	1	u0RB Topic 'vehicle_gps_position' i Open .msg file Open .msg folder Number of Outputs 3 i iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	Write"
וat'	> timestamp	read &	uORB Parameter Names and Data Type	1
uORB Write 'lon' Topic: 'vehicle_gps_position'	uORB Write Topic: battery_status	write	✓ Input Struct param : 'lat' : int32 >> Number of Elements 1 :	module
> 'timestamp_time'	>voltage_v		☑ Input Struct param :	parameter
uORB Write	uORB Write Advanced	modules	'lon' iii int32 >> Number of Elements 1	
> timestamp			☑ Input Struct param :	setting box
uORB Write_dai Topic: battery_status			'timestamp_time' : double >> Number of Elements 1 :	
>voltage_v		北於丁선	□ Input Struct param : □ Input Struct param :	
uORB Write Advanced_dai		北航可會	Sample Time (-1 inherited) -1	
1		BUAA Reli		-



📔 Block Parameters: uORB Write		×
-S-Function (mask) (link)		
uORB Write Output Block		
Publishes to a user provided named tro	opic structure.	
uORB Topic 'vehicle_gps_position'	Open .msg file	Open .msg folder

D Modules in PSP Toolbox

PSP Toolbox

Clicking the "Open .msg file" button in upper-right figure yields the content of the select ".msg" file (see the lower-left figure), and clicking the "Open .msg folder" button yields the list of all supported uORB messages (See the lower-right figure).

Z D:\PX4PSP\Firmware\msg\vehicle_gps_position.msg

	编辑器 视图 视图
1	# GPS position in WGS84 coordinates.
2	# the auto-generated field 'timestamp' is for the position & velocity (microseconds)
3	int32 lat
4	int32 1on # Longitude in 1E-7 degrees
5	int32 alt
6	int32 alt_ellipsoid
7	
8	float32 s_variance_m_s
9	float32 c_variance_rad # GPS course accuracy estimate, (radians)
10	uint8 fix_type # 0-1: no fix, 2: 2D fix, 3: 3D fix, 4: RTCM code differential, 5: Rea
11	
12	float32 eph # GPS horizontal position accuracy (metres)
13	float32 epv # GPS vertical position accuracy (metres)

Fig. uORB message file

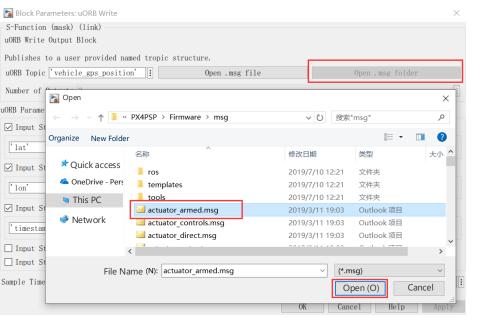


Fig. Pop-up window of "Open .msg folder" button





D Modules in PSP Toolbox

All modules (PWM output, RGB LED, etc.) mentioned in this section are implemented at the underlying code by reading and writing uORB messages. Theoretically, by using the "uORB Read and Write" modules, all messages and intermediate variables used in the PX4 autopilot can be accessed by Simulink. This simplifies the implementation of more advanced functions for controller design. The PSP toolbox also provides two examples (see file "e0\2.PSPOfficialExps\px4demo fcn call uorb e xample.slx", and file " $e0\2.PSPOfficialExps$ px4demo write uorb example.slx") to study this module.

There are two modules for "uORB Write" operation to choose. The suffix "_dai" allows sending extended messages (such as actuator_controls_0) based on a uORB message base class (such as actuator_controls).

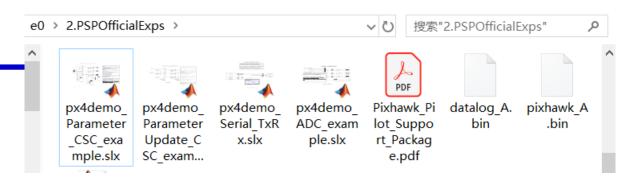
承 uORB Wi	rite Block				_	
uORB M	lessage:	actuator_controls	Apply	Select .msg file	Open .msg file	
		Struct Field stamp stamp_sample rol	1 dc	ataType Enable puble ~ puble ~ puble ~ puble ~		
(a) uORB Write Advanced module. uORB MsgID is the name of ".msg" file.						
		actuator_controls actuator_controls_0	Apply	Select .msg file	Open .msg file	
		Struct Field stamp stamp_sample rol	1 c	DataType Enable louble ~		

(b) uORB Write Advanced_dai module. uORB MsgID can be set different from the name of the ".msg" file.

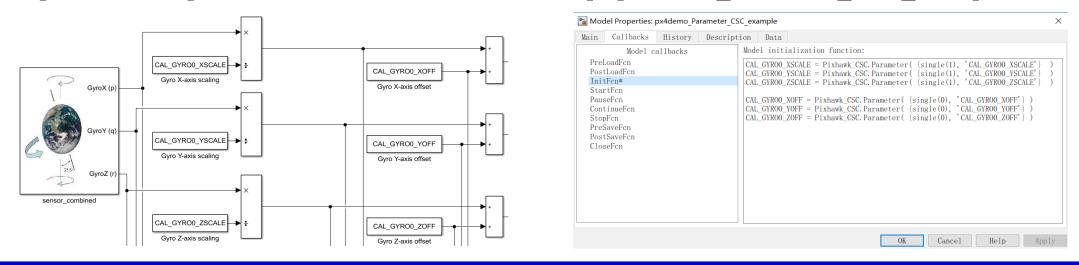
BUAA Reliable Flight Control Group



D Modules in PSP Toolbox



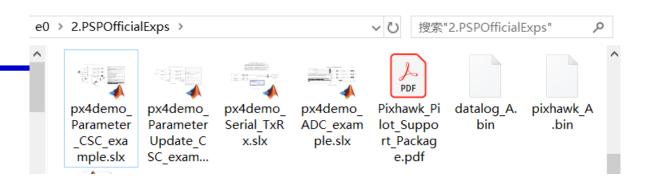
For the sake of convenience for controller parameter tuning in flight tests, the PSP Toolbox also provides interfaces to access the PX4 internal parameters. In this way, the parameters of the controller generated by Simulink can be tuned online in the GCS software, instead of modifying the controller parameters in Simulink, generating code, and uploading the firmware file again. An example of how to access the PX4 internal parameters is presented in file "e0\2.PSPOfficialExps\px4demo_Parameter_CSC_example.slx".



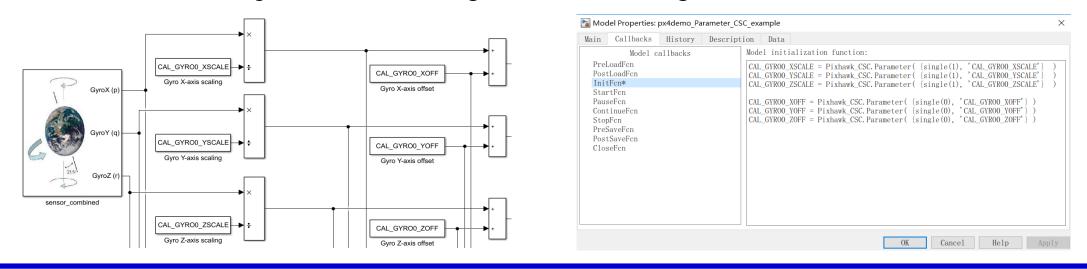




D Modules in PSP Toolbox



PX4 internal parameter access is realized by using the function "Pixhawk_CSC.Parameter({*, *})", which needs to be called in the Simulink initialization function (click "Simple" - "Model Properties" - "Callbacks" - "InitFcn" in the Simulink menu bar). For the examples shown in the lower-left figure, the corresponding parameter initialization script is shown in the figure on the lower right.

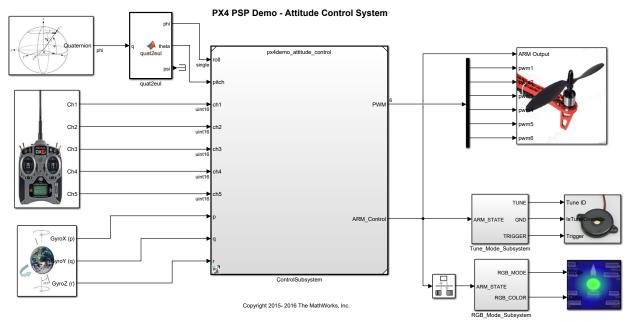






Code-Generation Configuration

1) Preparation of controller for code generation



1. design a controller in Simulink and verify it with SIL simulations

- 2. Copy the verified controller to a Simulink file.
- 3. Connect the input and output ports of the controller subsystem with the input (e.g., combined sensor module and RC input module) and output (e.g., PWM module and uORB modules) interface modules in the PSP library
- 4. An example of the obtained Simulink controller file is presented in the figure on the left. The example file is available in "e0\2.PSPOfficialExps\px4demo_attitude_syst

Fig. Example of Simulink controller connecting with PSP modules em.slx".





Code-Generation Configuration

2) Select target hardware

The new created Simulink file must be configured to support the code generation function of the PSP toolbox. Simulink setting panel can be opened by clicking "setting" button in the Simulink menu bar. Then, go to the "Hardware Implementation" tab and select the "Pixhawk PX4" item in the pull-down menu of the "Hardware board" option.

The CopterSim3DEnvironment - Simulink			
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>D</u> isplay Diag <u>r</u> am <u>Simulation</u> <u>A</u> nalysis <u>C</u> ode <u>T</u> ools	© Configuration Parameters: px4demo_write_uorb_example/Configuration (Active) − □ ×		
🔩 • 🔄 • 🚍 🧇 💠 👌 🕌 🏟 • 🖼 • 🕪 🔩 🕟 🕪	Q Search		
CopterSim3DEnvironment Settings	Solver	Hardware board: Pixhawk PX4	
(a) Simulink "Settings" button on MATLAB 2017b~2019a	Data Import/Export Optimization Diagnostics 	Code Generation None Pixhawk PX4 Pixhawk PX4 Device vendor: A Robot Operating System (ROS)	
SIMULATION DEBUG MODELING FORMAT APPS	Hardware Implementation	► Device details	
Imodeling Imodeling Imodeling Imodeling Imodeling Imodeling Model Compare Model Data Model Schedule Imodeling Model Editor Explorer Editor Imodeling EVALUATE & MANAGE DESIGN Settings SETUP	Model Referencing Simulation Target ▼ Code Generation Report Comments Symbols	Hardware board settings Operating system/scheduler settings Target hardware resources 	
(b) Simulink "Settings" button on MATLAB 2019b and above	Custom Code		
Fig. Simulink "Settings" button for different	Select target hardware in setting panel		

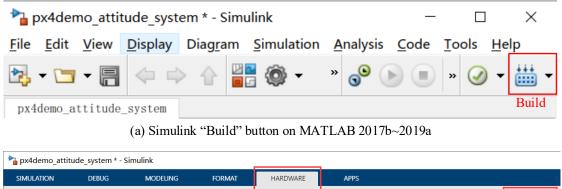


2.Soft and hardware

Code-Generation Configuration

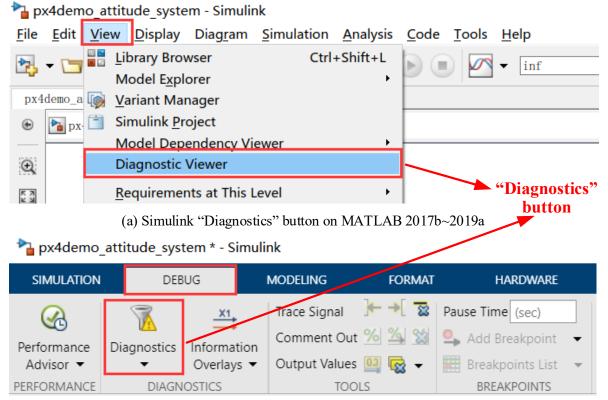
3) Source code compilation and firmware generation

Click the "Build" button. Then, the code generation and compiling process can also be observed by clicking the "Diagnostics" button in Simulink.





(b) Simulink "Build" button on MATLAB 2019b and above



(b) Simulink "Diagnostics" button on MATLAB 2019b and above





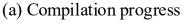
2.Soft and hardware

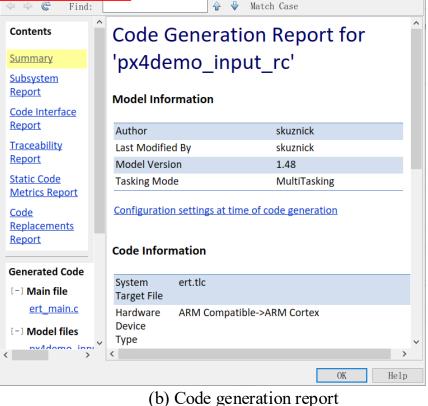
Code-Generation Configuration

4) Compiling process and results

A successful compiling process in the "Diagnostic Viewer" dialog is shown in Fig. (a), where the compiling progress is finished with the following text "Successfully generated all binary outputs". It can also be observed in Fig. (b) that a "Code Generation Report" document will pop up after the compiling process is finished.

ႃ Diagnostic Viewer	_		×	皆 Code Generation	Report
∃ -	0x			🗢 🕆 📽 Find	1:
	-40		_	Contents	Û Co
x4demo_input_rc 💿					
[5/8] Linking C static	library			Summary	'p
<pre>src/modules/px4_simulin</pre>	k_app/li	bmodule	s	Subsystem	
<pre>px4_simulink_app.a [6/8] Linking CXX executable nuttx_px4fmu-v3_default.elf [7/8] Generating px4fmu-v2.bin [8/8] Creating /mnt/c/PX4PSP/Firmware/build/px4fmu- v3_default/px4fmu-v3_default.px4</pre>				<u>Report</u>	Mo
				Code Interface	N/O
				Report	Aut
				Traceability	Las
				Report	Mo
				Static Code	Tas
				Metrics Report	145
"### Finished calling C	MAKE bui	ld proc	ess	Code	Cor
###"				Replacements	
"### Done invoking post			- I	Report	
"### Successfully gener	ated all	. binary			Coc
outputs."			J	Generated Code	_
C:\PX4PSP\examples\px4d	omo innu	t nc en	t n	[-] Main file	Sys
tw>exit /B 0	enio_inpu	ic_i c_ei	-	ert main.c	Tar
### Successful completion	on of bu	ild.			Hai
procedure for model: px				[-] Model files	U TV
				< nv4dama in	> <
Build process completed su	uccessfu	11y			
			4		





 \times





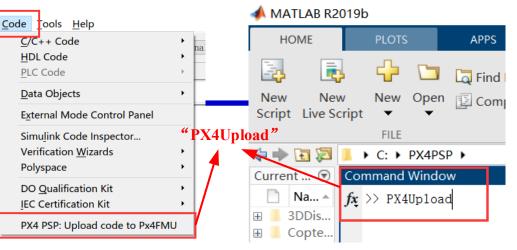
PSP Toolbox

Code-Generation Configuration

5) Upload PX4 firmware to Pixhawk hardware

Loaded firmware for 9,0, size: 879196 bytes, waiting for the bootloader... If the board does not respond within 1-2 seconds, unplug and re-plug the USB connector. PX4 SIMULINK = None attempting reboot on COM3... if the board does not respond, unplug and re-plug the USB connector. attempting reboot on COM3... if the board does not respond, unplug and re-plug the USB connector. attempting reboot on COM3... board does not respond, unplug and re-plug the USB connector. Found board 9,0 bootloader rev 4 on COM3 3d3ef f3073019 d035ab0d 3f60334e 10dda9f8 cdb0cbbd 42cdc6b6 3ba305f7 81532581 84ee3da6 23bc6340 8321be68 edd356c9 1e3b8 5c 5e07decc 9c6be5a2 458a1513 4bbbbc21 eda35ce5 a8b840a5 ef019ca5 c89bb183 bb00f0c0 06db1a26 7375ff57 1ca41d94 24aa662e idtype: =00 vid: 000026ac pid: 00000010 coa: Zu1H//9zzBXIrZQM28WfOdbCDgb5U9Pv8wcwGdA1qw0/YDN0EN2p+M2wy71Czca206MF94FTJYGE7j2mI7xjQIMhvmjt01bJHjuPXF4H3syca+WiRYc VEOu7vCHto1z1qLhApe8BnKXIm7GDuwDwwAbbGiZzdf9XHKQd1CSqZi4= sn: 0038001f3432470d31323533 100.0% Erase 100.0% Program:

Verify : [=====] 100.0% Rebooting.



(a) Simulink "PX4Upload" button for MATLAB 2017b~2019a

Analysis

(b) MATLAB "PX4Upload" command for MATLAB 2019b and above

- Use a USB cable to connect the MicroUSB (on the Pixhawk hardware) with the USB port on the computer.
- 2. For MATLAB 2017b-2019a, click "Code" "PX4 PSP: Upload code to Px4FMU" on the Simulink menu bar, then the firmware will be automatically uploaded to the Pixhawk autopilot; for MATLAB 2019b and above, readers can input the "PX4Upload" in MATLAB.
- 3. Sometimes, the Pixhawk autopilot has to be re-plugged to start the firmware uploading process.





D Hardware Composition and Connection

Hardware components required by this book include an RC transmitter, an RC receiver, a JR signal cable (connecting the Pixhawk autopilot and the RC receiver), a Pixhawk autopilot (Pixhawk 1 is recommended for studying, and higher hardware versions are recommended for outdoor flight tests), and a MicroUSB cable (connecting the computer and the Pixhawk hardware for power supply and data transmission).

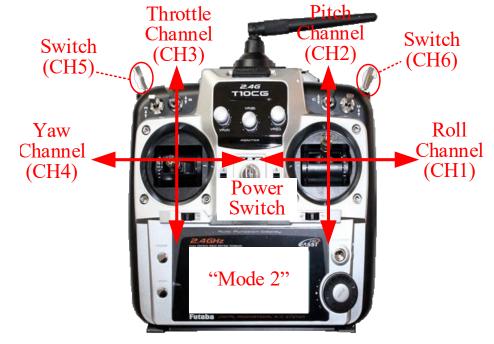






D Basic Operation Method for RC Transmitter

- 1. The RC transmitter used in this book should be set to "Mode 1". The throttle and yaw channels are controlled by the left stick on the RC transmitter, whereas the roll and pitch channels are controlled by the right stick.
- 2. The roll, pitch, throttle, and yaw channels correspond to the CH1 to CH4 of the RC receiver respectively; the upper-left and upper-right three-position switch corresponds to CH5 and CH6 for triggering the autopilot to switch flight modes or enable other functions.



Throttle : control up-down movement

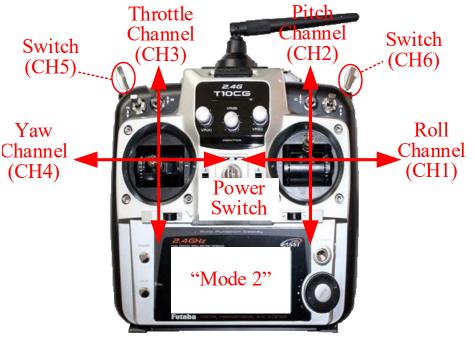
- Pitch : control forward-backward
- Yaw : control vehicle head direction
- Roll : control left-right movement





D Basic Operation Method for RC Transmitter

- 3. The throttle stick (CH3) and the pitch stick (CH2) moves from the bottom position to the top position, and the corresponding PWM value received by Pixhawk changes from 1100 to 1900.
- 4. The roll stick (CH1) and the yaw stick (CH4) move from the left position to the right position, and the corresponding PWM values change from 1100 to 1900.
- 5. The upper-left switch (CH5) and the upper-right switch (CH6) move from the top position (the farthest position from the user), middle position, and bottom position (the closest position from the user), and the corresponding PWM values change from 1100, 1500, to 1900.



Throttle	:	control up-down movement
Pitch	:	control forward-backward

- Yaw : control vehicle head direction
- Roll : control left-right movement





Uploading Firmware through QGC

Open QGroundControl and do the following steps. 1. Click the "Settings" button (the gear icon on the toolbar in the figure) to enter QGC setting page. 2. Click the "Firmware" tab. and then connect the Pixhawk hardware with a USB cable. QGC will automatically detect the Pixhawk autopilot. (Note: the subsequent operations are for Pixhawk 1 hardware, so if you are using other Pixhawks, just click "OK" button to download and use the official firmware, and then you can skip the following steps);

- 3. Click the "Advanced settings" checkbox.
- 4. Click on the "Standard Version (stable)" tab.



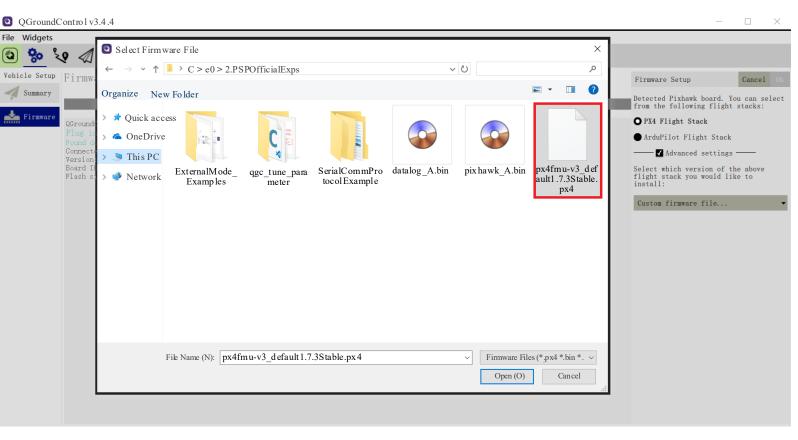
QGroundContr





Uploading Firmware through QGC

- 5. Select the "Custom firmware file .." option in the pop-up menu.
- 6. Click the "OK" button.
- 7. Select file " $\e^{0}2.PSPOfficialExps$ px4fmu-v3 default-1.7.3Stable.px4" in the pop-up file selection window, and click the "Open" button. Then, QGC will upload and burn the firmware into the Pixhawk hardware. (Note: this firmware file is only for Pixhawk 1 with 2M flash or Cube, for other Pixhawk hardware please download desired .px4 firmware at https://github.com/PX4/Firmware /releases/tag/v1.7.0)







D Pixhawk HIL Simulation Mode

1) Open the QGC software, and connect the Pixhawk autopilot with a USB cable. After QGC automatically recognize the Pixhawk autopilot and create a connection for parameter setting and data transmission, the UI of QGC can be seen.

le Widgets						
. وې <mark>څ (د</mark>	🖉 🖪 🔺 🛪	🗴 📥 💷 📋 N/A Manual	Disarmed			
Vehicle Setup	WARNING	: Your vehicle requires set	up prior to flight. Please res	solve the items marked	in red using the menu or	the left.
🚺 Summary	А	irframe 🧧	Radio	•	Flight	Modes
Firmware Airframe Radio	System ID Airframe type Vehicle Firmware Version	1 Simulation (Copter) HIL Quadcopter X 1.6.5dev	Roll Pitch Yaw Throttle Auxl Aux2	1 2 4 3 Disabled Disabled	Mode switch Flight Mode 1 Flight Mode 2 Flight Mode 3 Flight Mode 4 Flight Mode 5 Flight Mode 6	Setup require Unassigne Unassigne Unassigne Unassigne Unassigne
Power		Power 📀	Safety		Cam	era
Safety Tuning Camera	Battery Full Battery Empty Number of Cells	4.05 V 3.40 V 0	Low Battery Failsafe RC Loss Failsafe RC Loss Timeout Data Link Loss Failsafe RTL Climb To RTL, Then	Warning Return mode 0,5 s Disabled 30.0 m Land immediately	Trigger interface Trigger mode	Disabl

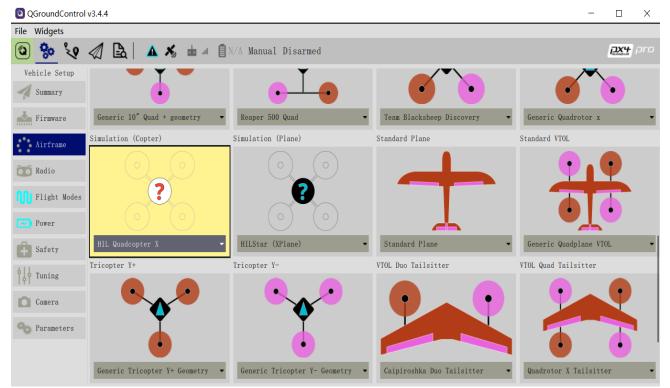




D Pixhawk HIL Simulation Mode

2) Click the "Airframe" tab (the third item) on the QGC setting page to confirm that the "HIL Quadcopter X" airframe mode is selected by default. This setting is critical for subsequent HIL simulation. Otherwise, a manual setup will be required. The method is :

- Select the "HIL Quadcopter X" airframe icon.
- Click the "Apply and Restart" button in the upper right corner of the UI. Then, the autopilot will restart to make the new airframe available.
- Wait for a few seconds; QGC will connect to the autopilot again and will check whether the setting is correctly established.

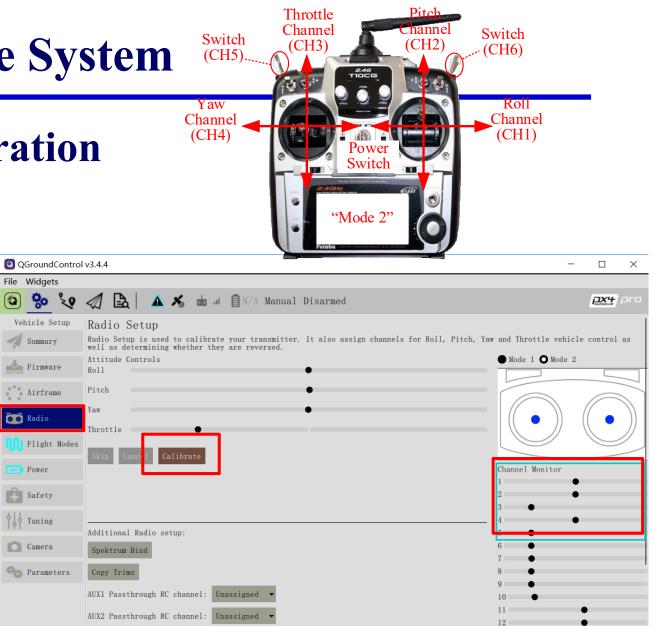






RC Configuration & Calibration

- 1. connect the Pixhawk autopilot and the RC receiver.
- 2. connect the Pixhawk autopilot with the computer.
- 3. Next, turn on the RC transmitter, and open QGC
- 4. click the "Radio" item on the QGC setting page
- Move the sticks of the RC transmitter and observe the trend of channels 1-6 on the Channel Monitor page on QGroundControl
- 6. The first and the fourth sliders in the figure on the right should move from left to right (the PWM values change from 1100 to 1900).



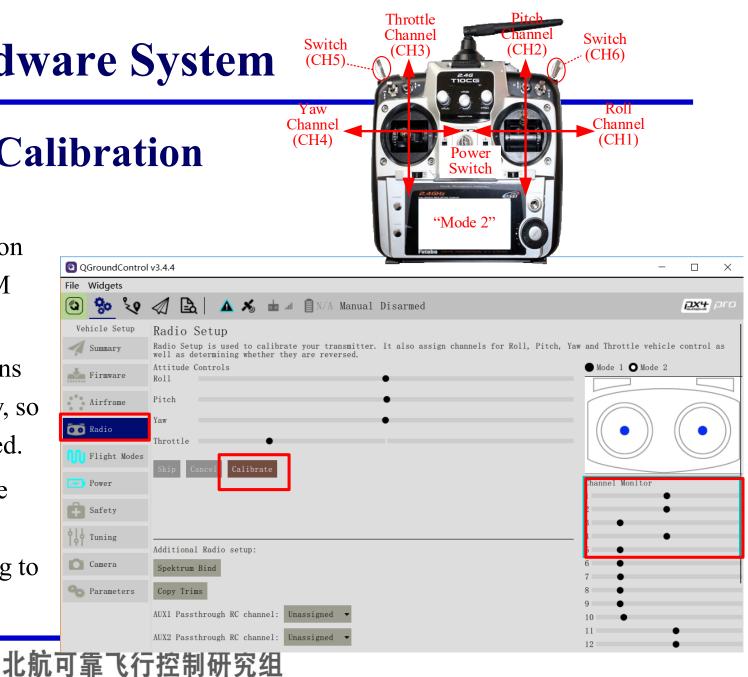




BUAA Reliable Flight Control Group

RC Configuration & Calibration

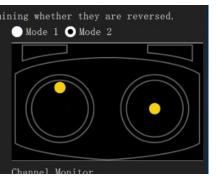
- 7. The third slider on the bottom-right region should move from left to right (the PWM value changes from 1100 to 1900).
- 8. If the above rules are not satisfied, it means that the RC transmitter is not set correctly, so the RC transmitter should be re-configured.
- 9. If the RC is configured correctly, click the "Calibrate" button and complete the RC calibration by moving the sticks according to the instructions on QGC.

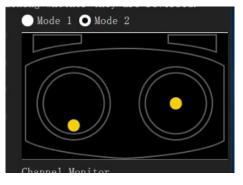


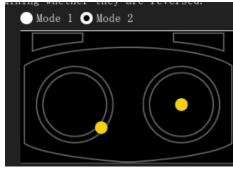


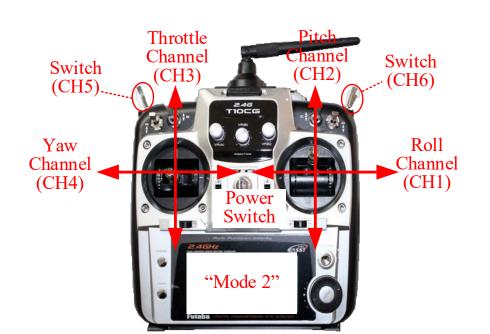
Configuration & Calibration

10. Click QGC's "Calibrate" – "Next" button, then move the RC's sticks according to the figures on the right (just follow the hints on QGC) to finish the RC calibration





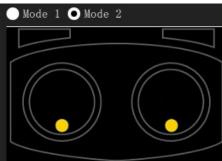




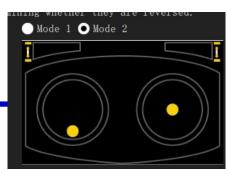














□ Flight Mode Settings

After the RC transmitter is successfully calibrated, enter the "Flight Modes" setting page (see the figure on the right) and select "Mode Channel" as the previously tested CH6 channel. Since the CH6 channel is a three-position switch, the top position (the farthest position from the user), middle position, and bottom position (the closest position from the user) of the switch correspond to "Flight Mode 1, 4, 6" in figure.

QGroundControl	v3.4.4			_		\times
File Widgets		ຟ 🔋 N∕A Manual Disarmed				
Vehicle Setup	Flight Modes Setup	configure the transmitter switche Switch Settings	s associated	with Flight Modes		
Firmware	Mode channel: Channel 6 -	Acro switch channel	-	Arm switch chann		
Airframe	Flight Mode 1 Stabilized - Flight Mode 2 Unassigned -	Landing gear switch channel Loiter switch channel	Unassigned -	Kill switch char Offboard switch	channel	
M Flight Modes	Flight Mode 3 Unassigned •	Position Control switch channel Return switch channel		Rattitude switch Stabilize switch		
Power	Flight Mode 4 Altitude 👻	Channel Monitor				
Safety	Flight Mode 5 Unassigned -					
Camera	Flight Mode 6 Position 💌 Use Multi Channel Mode Select	tion				
Parameters						





□ Flight Mode Settings

- 2. Associate these three modes to "Stabilized" (the stabilized mode, only including attitude control), "Altitude" (the altitude hold mode, including attitude and altitude control), and "Position" (the loiter mode, including attitude and position control).
- In subsequent HIL simulations, you can experience different control effects by switching between different modes.

O QGroundControl	v3.4.4	_		\times
File Widgets				
	🛷 🖹 🗮 🦟 📥 💷 🗍 N/A Manual Disarmed			
Vehicle Setup	Flight Modes Setup Flight Modes Setup is used to configure the transmitter switches associated with Flight Flight Mode Settings Switch Settings	it Modes.		
Firmware	Mode channel: Channel 6 - Acro switch channel Unassigned - Arm switch	ch channe	1	
Airframe	Flight Mode 1 Stabilized - Landing gear switch channel Unassigned - Kill swit	ch chann	el	
💿 Radio	Flight Mode 2 Unassigned - Position Control switch channel Unassigned - Rattitude			
₩ Flight Modes	Flight Mode 3 Unassigned - Return switch channel Unassigned - Stabilize	switch	channel	map
Power	Flight Mode 4 Altitude 👻 Channel Monitor			
Safety	Flight Mode 5 Unassigned -			
tuning	Flight Mode 6 Position -			
Camera	Use Multi Channel Mode Selection			
🇠 Parameters				





The HIL simulation platform includes a Real-time Motion Simulation Software — CopterSim and a 3D Visual Display Software — 3DDisplay.

😹 CopterSim	- 🗆 X	🐺 3D Display	- 🗆 X
	Frame Type Total Weight Diagonal Size Altitude 		北航可靠飞行控制研究组 BUAA Reliable Flight Control Group #1 (RFM) 0.00 #2 (RFM) 0.00
	Propeller Brand: Model: AFC 10x4.5MR		#3 (RPM) 0.00 y(m) 0.00 #4 (RPM) 0.00 z(m) 0.00 Attitude info Vx(m/s): 0.00
٤ 📃 ک	ESC Erand: Model: Hobbywing V Rotor 20A		Roll(deg) 0.00 Vy(n_/s): 0.00 Yaw(deg) 0.00 Vz(n_/s): 0.00
	Battery Brand: Model: ACE LIPo 3S-11.1V-25C-5500mAh		Flight Path Map Length (m): ± 5
Assembled UAV Datab	Activate! Model Parameter Add to Database Delete from Database		
Vehicle ID: UDP Po 1 14560	rt: Use DLL Model: Simulation Mode: 3D Scene Selection: Link Vehicle Initial Pos: Yaw Angle:		
Select Pixhawk Con	m: Legacy FMU COM6		
	X 0 Y 0 Z 0 Vx 0 Vy 0 Vz 0 Φ 0 Θ 0 Ψ 0		





	\bigotimes
\bigcirc	\odot
Copter	Sim.exe

	🎇 CopterSim						- 0	×
	010 0 0	Frame Type Total Quadcopter 1.5	Weight Diagonal kg 450	Size Alt nn 50			亢可靠飞行控制研究组 A Reliable Flight Control Group	
	_	Motor Brand: DJI	\sim			Model: 2312 KV960	~	
a)		Propeller Brand: APC	~			Model: 10x4.5MR	~	
	5	ESC Brand: Hobbywing	~			Model: XRotor 20A	~	
		Battery Brand: ACE	~			Model: LiPo 3S-11.1V-25C-5500mAh	~	
5)	Assembled UAV Databa	ise:	~	Activate!	Model Parame	Add to Database	Delete from Databa	ise
c)	Vehicle ID: UDP Por 1 14560	t: Use DLL Model: ~	Simulation Mode: PX4≪1.7 HIL		e Selection: ayFrogram ~	Link Vehicle Initia		,
d)	Select Pixhawk Com	Legacy FMU COM6		~ S	tart Simulation	Stop Simulation	Restart Simulati	.on
				X O		¥ 0	ZO	
e)				Vx 0		Vy 0	Vz 0	
				φ 0		θ	ψ 0	

Double-click the CopterSim shortcut on the 1) Windows desktop to open the CopterSim software, whose UI is presented in the figure on the left. The default simulation model and parameters are the same as for the Simulink multicopter model used in the SIL simulation system. This is because the CopterSim is developed based on the code generation technique with the Simulink multicopter model. CopterSim needs to run on a x64 Windows computer platform with a serial port and a MicroUSB cable to communicate with the Pixhawk autopilot.





CopterSim

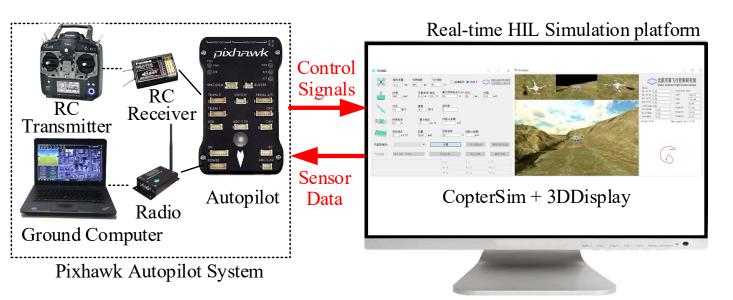
[🍔 CopterSim									-		×
	<mark>ରୁ</mark> ଡ଼ ୯ ତ	Frame Type Quadcopter 🛛 🗸	Total Weight 1.5 kg	Diagonal S 450	.ze nn	Altitude 50	n	 Brand Model Custom Design 		靠飞行控制 able Flight Co		
	_	Motor Brand: DJI		~				Model: 2312 KV960				
a)		Propeller Brand: APC		~				Model: 10x4.5MR				
	5	ESC Brand: Hobbywing		~				Model: XRotor 20A				
		Battery Brand: ACE		~				Model: LiP⊙ 3S-11.1V-2	5C-5500mAh	~		
b)	Assembled UAV Databa	ise:		~	Activa	ite!	Model Par	rameter Add to	Database)elete fro	m Databa:	se
c)	Vehicle ID: UDP Por 1 14560	t: Use DLL Model:	Simulat: ∨ PX4≪1.	ion Mode: 7 HIL		Scene Selec DisplayProg		Link Veh	icle Initial Pos 0 y: 0	s: Yaw An yaw:		
d)	Select Pixhawk Com	: Legacy FMU CO	[6			Start Si	mulation	Stop Si	mulation	Restart	Simulati	on
e)						x 0 Vx 0 Φ 0		Υ 0 ∇y 0 θ 0	Z C Vz V	0		

2) As shown in the figure on the left, the UI of CopterSim is divided into two parts. The upper part, presented in Fig. (a), is the input interface to design a multicopter by selecting popular components on the market. The lower part presented in Figs. (b)-(e) is the interface to connect with the autopilot for HIL simulation. Note that CopterSim enables by default only the basic functions required by this book. Registration is required to use many other practical functions, such as swarm simulation, high-fidelity UE4 scenes, and HIL simulations for other aerial vehicles (e.g., fixed-wing aircraft).





CopterSim



Principle of HIL simulation: CopterSim sends sensor data to the Pixhawk autopilot, and then the autopilot solves the motor PWM control signal and returns it to CopterSim. As a result, the Pixhawk autopilot can perform real-time control on the simulated multicopter in CopterSim, as well as control a real multicopter. Meanwhile, CopterSim will send the attitude and position information of the multicopter to the local network through the UDP protocol, and the 3DDisplay receives the multicopter flight information to complete the corresponding real-time 3D scene display.





HIL Simulation Platform

(b) Assembled UAV Database:

Model Parameter

Activate!

Add to Database Delete from Database

etailed Parameters			- 🗆 X
	Multicopter Mass: m Acceleration of Gravity: g Inertia Matrix: Jxx	1.4 9.8 0.0211	kg m/s^2 kg.m^2
	J=diag(Jxx, Jyy, Jzz): Jyy	0.0219	kg. m ²
	: Jzz	0.0366	kg. m ²
	Distance of Motor to Center: d	0.225	M
	Prop. Thrust Coef. (Tp/0)^2): CT	1.105e-05	N/(rad/s) [^]
otor Num. 4 Arm Num. 4 Config. (x o <mark>ix</mark>	Prop. Moment Coef. (Mp/ω^2) : CM	1.779e-07	N.m/(rad/s
	Throttle σ to Motor Speed ω ss:	1148	rad/s
	$(\omega_{ss=CR*}\sigma+\omega_b)$: ω_b	-141.4	rad/s
overing Time 18 min Motor Power 35.2 W	Motor-Propeller Inertia: Jm	0.0001287	kg.m [°] 2
hrottle Perce60.9 % Battery Voltage 11.8 V	Motor Response Time Constant: Tm		s
otor Current 3.74 A Battery Current 15.5 A	Air-Drag D Coef.(D=Cd*V^2): Cd	0.073	N/(m/s)^2
otor Speed 5235.8 rpm Power Efficiency 76 %	Air-Torque M Coef.(M=Cm*w^2): Cm		N.m/(rad/s)^2

Fig. Model parameter configuration dialog

- 1. Click the "Model Parameter" button in the middle of the CopterSim UI. The model parameter configuration dialog in the figure on the right will pop up; the model parameters stored in the previous simulation will be displayed here.
- 2. The parameter dialog mainly includes two parts: the hover information (hover endurance, throttle, output power, motor speed, etc.) and the basic multicopter parameters (total mass, the moment of inertia, size, thrust coefficient, and drag coefficient).

Fig. 模型配置错误提示框





HIL Simulation Platform

(b) Assembled UAV Database:

E Detailed Parameters			- 🗆 X
	Multicopter Mass: m	1.4	kg
	Acceleration of Gravity: g	9.8	m/s^2
	Inertia Matrix: Jxx	0.0211	kg.m^2
	J=diag(Jxx, Jyy, Jzz): Jyy	0.0219	kg.m^2
y, y,	: Jzz	0.0366	kg.m^2
	Distance of Motor to Center: d	0.225	m
	Prop. Thrust Coef. (Tp/0^2): CT	1.105e-05	N/(rad/s)^
	Prop. Moment Coef. (Mp/ω^2): CM	1.779e-07	N.m/(rad/s
Rotor Num. 4 Arm Num. 4 Config. (x oix	Throttle σ to Motor Speed ω_{ss} :	1148	rad/s
	$(\omega_{ss=CR*\sigma+\omega_b})$: ω_b	-141.4	rad/s
Hovering Time 18 min Motor Power 35.2 W	Motor-Propeller Inertia: Jm	0.0001287	kg.m^2
Throttle Perce60.9 % Battery Voltage 11.8 V	Motor Response Time Constant: Tm	0.05	S
Motor Current 3.74 A Battery Current 15.5 A	Air-Drag D Coef.(D=Cd*V^2): Cd	0.073	N/(m/s)^2
Motor Speed 5235.8 rpm Power Efficiency 76 %	Air-Torque M Coef.(M=Cm*w^2): Cm	0.0055	N.m/(rad/s)^2
Noise Level (0~1): 0	Default Params. Store and Appl	y Params.	Cancel

Fig. Model parameter configuration dialog

3. Clicking the "Restore to Default Params" button will restore the model parameters to the default values; clicking the "Save and Apply Params" button will store the current parameters to the database for subsequent HIL simulations.

Model Parameter

Activate!

Add to Database

Delete from Database

4. A noise level between 0-1 or larger than 1 can also be selected to represent the noise level of actual sensors. This enables the possibility of testing the anti-interference ability of the designed control algorithms.

Fig. 模型配置错误提示框





HIL Simulation Note 1: If you are using Pixhawk to perform HIL simulation, please plug Pixhawk in the computer and wait for about ten seconds until boot process to complete. Then, you can press "Start Simulation" on CopterSim.

Note 2: After the simulation start, you should also wait for about ten seconds until the message box showing "** EKF initialization finished". Then you can use RC to arm and control the drone.

Vehicle ID: UDP Port: 1 14560	Use DLL Model:	Simulation Mode: PX4≪1.7 HIL ✓	3D Scene Selection: 3DDisplayProgram ~	Link	Vehicle Initial Pos x: 0 y: 0	: Yaw Angle: yaw: 0 °
Select Pixhawk Com:	Legacy FMU COM6	\sim	Start Simulation	Ste	op Simulation	Restart Simulation
Connect to SerialPort! SerialPort connection is Multicopter #> COM6 Connect to COM6 successf Enter Stablized Mode! Init MAVLink			x 0 vx 0 φ 0	Υ 0 Vy 0 θ 0	ΖΟ Vz ψ	

Start and Stop Simulation: After the multicopter parameters and the noise level are configured, connect the Pixhawk autopilot with the computer. The serial port of the Pixhawk autopilot will be listed in the "Select Pixhawk Com" drop-down menu. Select the Pixhawk serial port (usually described by the text "FMU"), and click the "Start Simulation" button to start the HIL simulation. clicking the "Stop Simulation" button will stop the HIL simulation, and clicking the "Restart Simulation" will re-initialize the multicopter position and states to their initial values.





D 3DDisplay



3DDisplay.exe

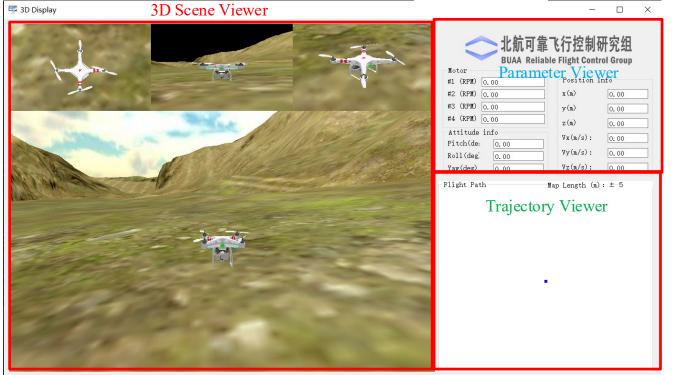


Fig. Main UI of 3DDisplay software



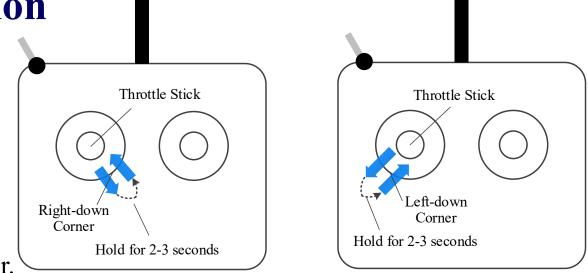
Double-click the 3DDisplay shortcut on the Windows desktop to open the 3DDisplay software. As shown in figure on the left, the "3D Scene Viewer" on the left side of the 3DDisplay UI presents the current flight status of the multicopter in the 3D scene. The basic flight parameters are displayed in the upper right window of the 3DDisplay UI, including motor speed, position, and attitude information. The flight trajectory of the multicopter is displayed on the lower right window of the 3DDisplay UI.



□ Flight Tests with HIL Simulation

In the HIL simulation platform, when controlling a real multicopter, it is convenient to control the simulated multicopter with a real RC transmitter to perform basic actions, such as arming, taking off, manual flight, landing, etc. The detailed steps are described next.

1. Push up the POWER switch to turn on the RC transmitter.



(b) Disarm action

(a) Arm action

2. Correctly connect the computer with the Pixhawk hardware system (including the Pixhawk autopilot and the RC receiver) and start the HIL simulation in CopterSim according to the procedure mentioned above.

3. As shown in Fig. (a), arm the Pixhawk autopilot by moving the left-hand stick on the RC transmitter (CH3) to the lower-right corner for 2-3 seconds.

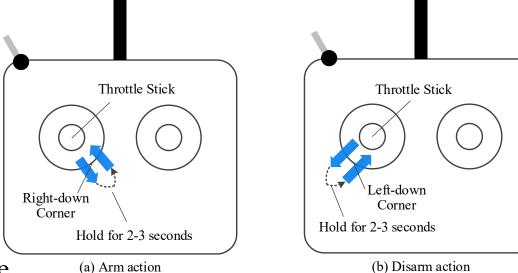




□ Flight Tests with HIL Simulation

4. Pixhawk is successfully armed when its LED turns from slow flashing to always on Higher Pixhawk hardware (e.g., Pixhawk 2/3/4/5) starts to discard LED module, so an external I2C LED module is required to observe the lighting effect, and the CopterSim print message "Detect Px4 Armed" is received from Pixhawk. If arming Pixhawk fails, please disconnect all hardware and software and repeat the above steps.

5. Pull up the left-hand stick on the RC transmitter (CH3) for the multicopter to take off and fly up to a certain altitude. Next, vertically move the left-hand stick to verify the vertical motion control of the multicopter.



6. Horizontally move the left-hand stick on the RC transmitter (CH4) to verify the yaw angle motion control of the multicopter.





□ Flight Tests with HIL Simulation

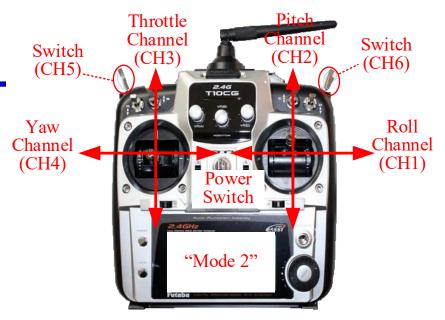
7. Vertically move the right-hand stick on the RC transmitter (CH2) to verify the pitch angle control as well as the forward and backward motion control of the multicopter.

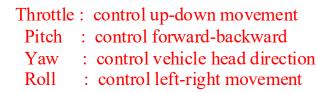
8. Horizontally move the right-hand stick on the RC transmitter (CH1) to verify the roll angle control as well as the left and right motion control of the multicopter.

9. Change the position of the top-right switch on the RC transmitter (CH6) to verify the mode switching control of the multicopter.

10. Pull down the left-hand stick on the RC transmitter (CH3) to land the multicopter to ground.

11. Move the left-hand stick on the RC transmitter (CH3) to the lower-left corner for 2-3 seconds to disarm the Pixhawk.





12. Click the "Stop Simulation" button on the CopterSim UI to stop the HIL simulation. Then, disconnect all software and hardware connections between the computer and Pixhawk.







All course PPTs, videos, and source code will be released on our website
<u>https://rflysim.com/en</u>

For more detailed content, please refer to the textbook: Quan Quan, Xunhua Dai, Shuai Wang. *Multicopter Design and Control Practice*. Springer, 2020 <u>https://www.springer.com/us/book/9789811531378</u>

If you encounter any problems, please post question at Github page <u>https://github.com/RflySim/RflyExpCode/issues</u>

If you are interested in RflySim advanced platform and courses for rapid development and testing of UAV Swarm/Vision/AI algorithms, please visit:

https://rflysim.com/en/4_Pro/Advanced.html





Thank you!

