This is an overview of all the tools used in the LEAPFROG simulation. This is not a guide on how to use or install the tools, rather this is a reference for what they are, why they are used for LEAPFROG, and where to find more information about them.

**PX4**
- Changes to PX4 2
- References 2
- Notes 2

**Software in the Loop (SITL)**
- Notes 2

**MAVLink**
- Changes to MAVLink 3
- References 3

**Gazebo**
- References 4

**ROS**
- MAVROS 4
- Notes 4
- References 4

**Extras**
- QGroundControl 5
- PlotJuggler 5
- References 6
PX4

PX4 is an open source flight control software that is often used in drones and other unmanned vehicles. It was chosen for the LEAPFROG platform because it is made for the COTS flight controller, pixhawk.

The official PX4 documentation can be very helpful for debugging setup and getting a better understanding of its capabilities since the LEAPFROG documentation will only be going into how this project specifically utilizes PX4.

Changes to PX4

For the LEAPFROG simulation, we extended the PX4 Firmware to include support for the specifications of the LEAPFROG system. This includes:

- Creating new model with the actuator specifications of the LEAPFROG vehicle
- Adding necessary controllers for ACS and TVC systems to control vehicle while working within the PX4 system
- Removing unnecessary features and models from PX4

References

- LEAPFROG enabled version of PX4:  
  https://github.com/SERC-LEAPFROG/LEAPFROG-Firmware

Notes

[4/9/21] This will not be functional until teams sign up on github and are given access to their own private version. Specifically, the submodule for sitl_gazebo will give a 404 error on github because it is not being made available yet. This will likely be changed in the future to have sitl_gazebo link to an incomplete version instead of giving a 404 error.

Software in the Loop (SITL)

SITL is a mode for PX4, where the flight stack is meant to run completely on a computer. The primary component of the SITL is a modified version of PX4 that is designed to run with only software. The modifications essentially simulate the real vehicles controls, dynamics, and actuations, while still using the majority of the original PX4 software.
Notes

SITL alone isn’t really an individual piece of software, rather it is an idea and set of applications and paradigms to test out the PX4 system without the use of dedicated hardware. However, when SITL is being referenced by LEAPFROG (and PX4) it usually refers to PX4 on SITL mode.

MAVLink

Micro Air Vehicle Link (MAVLink) is protocol for communicating with unmanned vehicles. It is a very lightweight protocol with MAVLink v2 having just 14 bytes of message overhead. Because of the nature of the protocol, it is very well suited for limited communication bandwidth.

MAVLink messages are defined within XML files, which are then used by the MAVLink toolchain to generate MAVLink libraries for the necessary language. This allows for applications to utilize the MAVLink protocol and message set in a variety of complex applications across numerous microcontrollers and operating systems.

Changes to MAVLink

By default, PX4 uses the standard C mavlink library; however, for the LEAPFROG Simulation the original message set was extended to add custom items to aid in the tuning process for the ACS and TVC controllers.

References

- Mavlink with custom LEAPFROG messages: https://github.com/SERC-LEAPFROG/LEAPFROG-Mavlink

Gazebo

Gazebo is an open-source 3D robotics simulator. It boasts high-fidelity, dynamics simulations through the use of different physics engines (ODE, Bullet, Simbody, and DART), realistic 3D rendering and graphics from OGRE, sensors and noise configurations, and most importantly for LEAPFROG, a system for creating custom plugins that directly access the Gazebo API.
The most important feature of Gazebo as it relates to PX4 is the ability to create a custom model that accurately represents the LEAPFROG flight platform. The model being used in Gazebo was directly exported from SOLIDWORKS and contains all of the mass properties of the real vehicle with the necessary joints that dictate the type of actuators for the vehicle. Additionally, with the custom model and joints, custom gazebo plugins must be created to interface with the ACS and TVC systems accordingly.

References

- Gazebo API Documentation: [https://osrf-distributions.s3.amazonaws.com/gazebo/api/11.0.0/index.html](https://osrf-distributions.s3.amazonaws.com/gazebo/api/11.0.0/index.html)

ROS

Robot Operating System, or ROS, is an open-source meta-operating system that provides a large set of tools, controls, messaging, and applications for robotic systems. The LEAPFROG system does not use ROS at its core, rather it uses ROS to extend the capabilities of PX4 with offboard/external applications or programs.

MAVROS

MAVROS is a MAVLink extendable communication node for ROS. It's primary purpose within the LEAPFROG Simulation is to communicate with PX4 over MAVLink for external applications.

Notes

[4/9/21] The PX4 development recently started to recommend migrating to ROS2; however, the LEAPFROG system is using ROS 1 with MAVROS for offboard control. The primary motivation for using the ROS 1 and MAVROS configuration was to minimize the changes to data flow in the PX4-SITL-Gazebo workflow.
References

- http://wiki.ros.org/noetic
- http://wiki.ros.org/mavros

Extras

The following tools are not required, but are highly recommended to aid in the development and debugging process.

QGroundControl

QGroundControl is a Ground Control Station (GCS) for the MAVLink protocol. It provides full flight control and mission planning for MAVLink enabled devices, so a properly configured LEAPFROG system should be able to be directly controlled via QGroundControl. Mission Planner is another popular GCS used for MAVLink devices; however, QGroundControl is recommended for its ease of use and superior cross-platform support.

Example interface for mission control
PlotJuggler

PlotJuggler is a time-series visualization tool with native ROS and PX4 log support. Its main advantages are live PX4 telemetry or ROS message plotting as well as PX4 ulog support for post processing and analysis.

Example plotting interface with custom message transformation being applied on-the-fly
https://github.com/facontidavide/PlotJuggler

References

- http://qgroundcontrol.com/
- https://github.com/facontidavide/PlotJuggler
- http://wiki.ros.org/plotjuggler