This is a comprehensive overview of the LEAPFROG platform as well as exactly how all of the tools specified in the LEAPFROG Simulation Tools Overview come together to create a comprehensive simulation. This document does not contain instructions on how to install any of the systems of the simulation nor does it contain detailed information regarding what each tool is.

**Hardware Overview**
- Actuator Specifications 2
- Actuator Purpose 2
- Vehicle Systems 3

**Controller Configuration**
- TVC/Gimbal/Engine Attitude Control 4
- ACS 6
  - Notes 6

**Simulation**
- Overview 6
  - Notes 6
- Simulator MAVLink API 7
- PX4 on SITL 7
- Simulator (Gazebo) 7
- QGroundControl 8
- API/Offboard Control 8
LEAPFROG
Hardware Overview

Actuator Specifications

<table>
<thead>
<tr>
<th>Actuators</th>
<th>Saturation Force</th>
<th>Quantity</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gas-powered engine</td>
<td>300 N</td>
<td>1</td>
<td>Center of vehicle</td>
</tr>
<tr>
<td>Roll ACS Thrusters</td>
<td>5 N</td>
<td>2</td>
<td>Edge of vehicle along roll axis (pointed downward)</td>
</tr>
<tr>
<td>Pitch ACS Thrusters</td>
<td>5 N</td>
<td>2</td>
<td>Edge of vehicle along pitch axis (pointed downward)</td>
</tr>
<tr>
<td>Engine roll linear servo</td>
<td>900 N</td>
<td>1</td>
<td>Below bottom platform connected to engine</td>
</tr>
<tr>
<td>Engine pitch linear servo</td>
<td>900 N</td>
<td>1</td>
<td>Below bottom platform connected to engine</td>
</tr>
</tbody>
</table>

Actuator Purpose

<table>
<thead>
<tr>
<th>Actuator</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine</td>
<td>Provide thrust for hovering and translation</td>
</tr>
<tr>
<td>ACS thrusters</td>
<td>Control the attitude of the vehicle</td>
</tr>
<tr>
<td>Linear servos</td>
<td>Controlling the attitude of the engine</td>
</tr>
</tbody>
</table>
Vehicle Systems

Isometric View

Top View

Bottom View

Left View

Right View
Thrusters (2.5N)
\[ t_{p1/2} : \text{pitch thrusters} \]
\[ t_{r1/2} : \text{roll thrusters} \]
\[ t_{y1/2} : \text{yaw thrusters} \]

Engine (300N)

Linear Actuators (servos) (900N)
\[ \text{LA}_S : \text{starboard actuator} \]
\[ \text{LA}_P : \text{port actuator} \]

Controller Configuration

To control the vehicle, the engine angle/attitude is separately controlled from the rest of the vehicle. The engine control is referred to as the thrust vector control (TVC). The control architecture for this uses a modified version of PX4. Separately the attitude control system (ACS) is used to control the attitude of the vehicle (separate from the engine attitude).

TVC/Gimbal/Engine Attitude Control

The controls architecture uses PX4, an open source control software for planes and drones. Specifically, it is based on the quadcopter control architecture:
The difference is that the vehicle doesn’t use the Attitude and Rate Controllers. Instead, this is handled by the linear servos. In the simulation, this is handled by the linear actuator controllers.

1. **Position Controller:**
   a. Input: Desired position for vehicle (position setpoint)
   b. Output: Desired thrust vector (decomposed into thrust magnitude and attitude for engine)

2. **Actuator Controllers (simulation only):**
   a. Input: Desired attitude for engine (attitude setpoint)
   b. Output: Input force for actuators
ACS

The purpose of the ACS controller as currently designed for the LEAPFROG system is to always keep the platform of the vehicle level.

Notes

- The amount of pressure in the gas tanks lowers as the pressure in the tanks decrease which directly affects the performance of the ACS system.

Simulation

The following documentation is adapted from the PX4 simulation documentation with additional information that extends/elaborates on exactly how the simulation works for the LEAPFROG platform.

Overview

The LEAPFROG/PX4 SITL simulation environment can roughly be expressed by the following diagram, which will be elaborated on further below. All parts of the SITL system connect via UDP and can run on either the same computer, another computer on the same network, or even a computer on a different network (given that router settings allow for necessary port access).

Notes

- An important distinction must be made that the software that runs PX4 on SITL is distinctly different from the software that directly controls the model of the vehicle in the Gazebo simulator.
Simulator MAVLink API

The simulator (Gazebo) communicates with the actual PX4 flight stack using the Simulator MAVLink API. This APIs main purpose is to bridge the gap between data and instructions from the controls in PX4 and the Gazebo environment. Specifically, it translates the commands sent from PX4 about how to control the vehicle and what it should be doing so that Gazebo can understand the commands and act accordingly. From Gazebos perspective, sensor and feedback is sent back to PX4, so that the controllers can act accordingly. Additionally, the LEAPFROG Simulation contains a set of custom mavlink messages that were defined and implemented to assist in tuning the controllers that are also sent using this API.

PX4 inputs from simulator
- Sensor and other message
- HIL_SENSOR
- HIL_GPS
- HIL_OPTICAL_FLOW
- HIL_RC_INPUTS_RAW
- HIL_STATE_QUATERNION

Control signals / Telemetry

Flight stack

Simulator

PX4 motor/actuator outputs
- Motor and actuator value messages
- HIL_ACTUATOR_CONTROLS

PX4 on SITL

When PX4 runs in SITL mode, the commands that are originally sent to dedicate hardware interfaces are instead restructured and routed to the desired simulator (Gazebo). The Simulator MAVLink API mentioned above handles all of this communication and should not need to be modified unless more extension of logging information is desired.

Simulator (Gazebo)

The main purpose of Gazebo for the SITL is to use the physics engines to compute the necessary responses that the LEAPFROG vehicle should be making. The LEAPFROG vehicle being used in Gazebo is directly exported from SOLIDWORKS, thus it contains all of the correct
positional and mass properties of the real vehicle. In addition, the Gazebo model of the LEAPFROG vehicle contains the following joints to allow for realistic behaviors.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Joint Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Actuators (servos)</td>
<td>Prismatic</td>
<td>Prismatic joints in Gazebo allow for translation across a single axis, which is the exact behavior of the linear actuators. In addition, the joints used have the exact specifications for saturation force and extension speed as their physical counterparts.</td>
</tr>
<tr>
<td>Platform connection to linear actuators</td>
<td>Ball</td>
<td>The linear actuators are connected to the bottom level of the LEAPFROG vehicle using a ball joint. This is necessary to allow for the full range of movement for the gimballed engine.</td>
</tr>
<tr>
<td>Gimbal Rings</td>
<td>Revolute</td>
<td>Each of the rings of the gimbal used to control the engine are connected via a revolute joint that allows for rotation along a single axis.</td>
</tr>
</tbody>
</table>

**QGroundControl**

QGroundControl connects to the PX4 flight stack using the MAVLink protocol. The setup and connection should be automatic and does not require any additional configuration by default. While it can be used in preliminary testing for telemetry and basic waypoint navigation, it will not be sufficient for the software challenges due to its lack of environment awareness and corresponding navigation methods.

**API/Offboard Control**

The recommended way to achieve offboard control of the vehicle is via MAVLink using MAVROS. Offboard Control is necessary for the guidance and navigation portion of the software challenge and is left for teams to decide what is the optimal algorithms or decisions to make for navigating the vehicle.