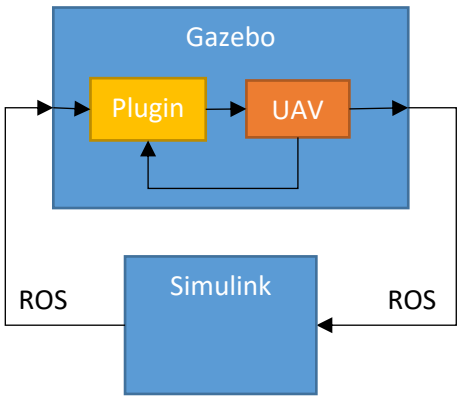
IMAV 2017 Virtual Challenge

1. Usecases

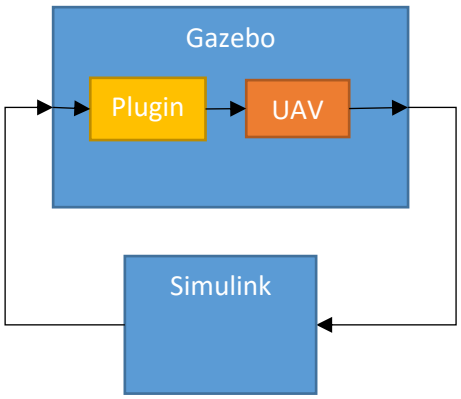
a) Environment to test high-level navigation, guidance and planning algorithms.



Gazebo handles the physics of the rigid body (UAV), while the attached plugin takes care of the stabilization of the system. Therefore, the IMU data is used directly from the simulation in the inner loop (without noise), and the forces and torques are exerted on the body mimicking the effect of the propellers.

The outer loop (involving Simulink) handles the high-level planning based on the image and sensor data acquired by the virtual instruments. This is done through ROS.

b) Environment to test low-level stabilization.



To test the stabilization algorithms the 2 parts (Gazebo, Simulink) should be synchronised, so the system has a deterministic behaviour for robust testing and analysis. Using only ROS would not provide this, but as it was highlighted here it is possible to step through the simulation using an external trigger.

2. Requirements

a) The requirements for this usecase:

- Data from Gazebo:
  - Video
  - LIDAR
  - Ultrasonic
  - Absolute position/orientation (for debugging)
  - ?
- Data to Gazebo:
  - Velocity commands (x, y, z, yaw)
  - Drone respawn command (in case of unrecoverable situations)
  - ?
- Simulink controller:
  - Basic ROS communication
  - Example to highlight built-in features that could be utilized by the teams
  - Example for code gen
  - ?

b) The requirements for this usecase:

- Simulation that represents the dynamics of the real systems
- Ability to apply a set of test conditions
- Ability to use our built-in control tools

3. Comments

a)

The current setup works this way. Similarly to the Mission on Mars Rover project, and the drone demo by Jeihun Lee, the system is stabilized, and only high level commands are used to guide the system. It would not take too much time to finish this, and provide an environment for testing for the virtual challenge.

b)

We have some pretty good examples/webinars promoting MBD with drones (here), which I think would provide a better workflow then binding Simulink and Gazebo together. With code gen the synthesised controller could be used in the Gazebo plugin with the appropriate wrapping and also on the physical systems. This way the teams could pick up some extra tools provided by MathWorks, and also (I think) the overall approach would be more convenient.