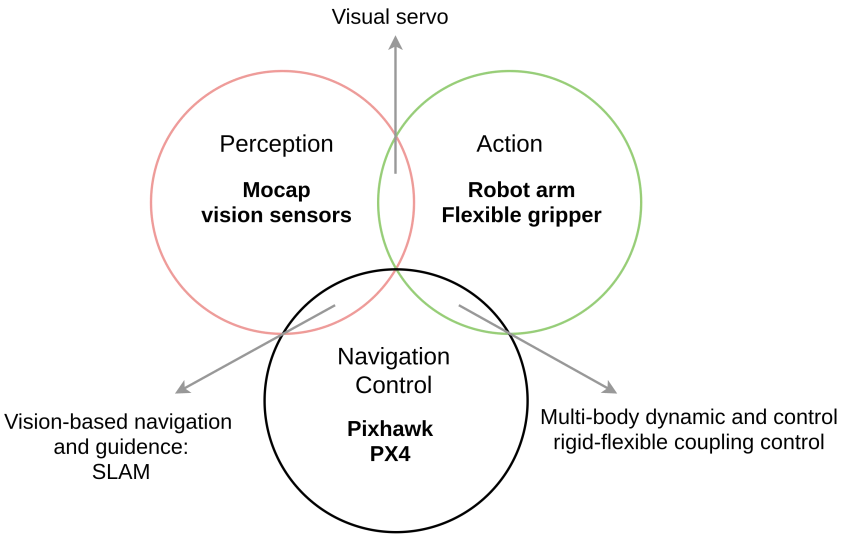
Smart Hawk

Introduction

The Smart Hawk is a aerial robot perception and grab platform. As aerial robot development, the controller and perception is stable, but its function is still limited. Currently, the aerial robot is mostly used in photograph and detection, and agriculture. In order to complete more complex tasks it needs more power actuators such as the gripper and arm.

The platform is for research and application, the following is the system components and and research fields:

- Vision-based navigation and guidance : SLAM
- Visual servo, object detection, object state estimate
- Multi-body dynamic and control, rigid-flexible coupling dynamic and control



1. Software in the loop simulation: CoppeliaSim-px4

1.1. Why CoppeliaSim

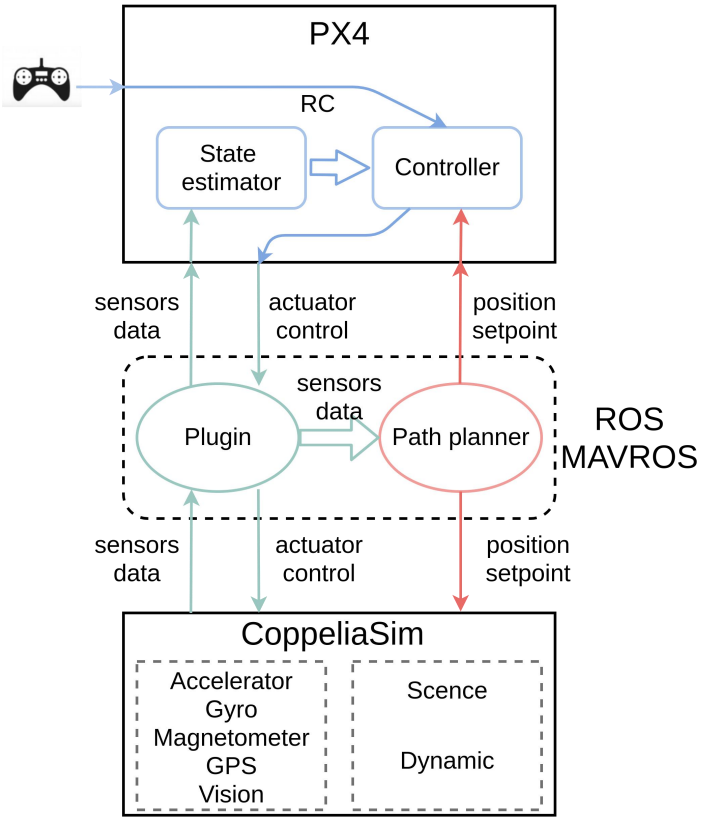
One of the important features of this platform is its arm and gripper, which require a more complex model. At present, Gazebo is a more common simulator, but compare Gazebo to CoppeliaSim, the latter has many advantages and more suitable for our project:

- More user-friendly features for model modification, especially to be able to modify an model from inside the simulator is very useful and practical.
- More user-friendly features for world modelling, that doesn’t require any deep knowledge of XML, and can easily and quickly setup a simulation world.
- ROS plugin can easily connect to the simulator, and there are also many remote control API.
- Less hardware demanding

1.2. Architecture

The simulation includes three parts: CoppeliaSim, ROS, and PX4. ROS as the middle layer between the CoppeliaSim and PX4.

Plugin: the plugin depend on MAVROS package, to transform messages between CoppeliaSim and PX4, plugin access to CoppeliaSim by ROS interface, and communicate with PX4 by MAVLINK.



1.3. Setup simulation

1.3.1. Tools install

- Install CoppeliaSim EDU
- Install ROS and MAVROS
- Build PX4 firmware

1.3.2. *Start simulation*

- *Start PX4:* `make px4_sitl_default non_iris`

Run the command the PX4 start and you can see:” Waiting for simulator to accept connection on TCP port 4560” in the terminal.

- *Launch MAVROS and connect to PX4:*

  `roslaunch mavros px4.launch fcu_url:="tcp-l://:4560"`

The MAVROS communicate with PX4 by port 4560, and after run this command you would see: “ Simulator connected on TCP port 4560.”, the MAVROS has connected to the PX4.

- *Run plugin node.*

After run the plugin, the GCS can receive message from PX4, and report.

- *Start CoppeliaSim scene.*

If there are message:”GPS checks passed” and “ EKF aligned”, the simulation is successfully setup.

1.4. Communication and messages

Some items about communication need to pay attention. There are four messages are used: `HIL_SENSORS`, `HIL_STATE_QUATERNION`, `HIL_GPS`, `HIL_ACTUATOR_CONTROLS`. In MAVROS the messages are published to `/mavros/hil/imu_ned`, `/mavros/hil/state`, `/mavros/hil/gps`, `/mavros/hil/actuator_controls`.

1.5. Frame and units

The simulation has three parts, and the messages in different part have different frame and units, it could cause errors sometimes. At first, let’s declare some frames and its abbreviation.

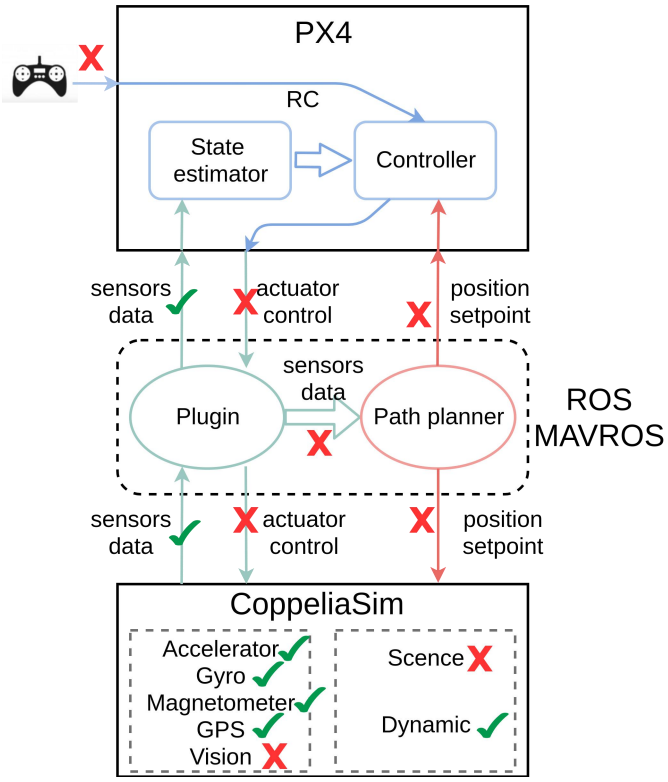
Frame	XYZ	Diagram
NED	X=North, Y = East, Z = Down	
ENU	X=East, Y = North, Z = Up	
FRD	X=Forward, Y = Right, Z = Down	
FLU	X=Forward, Y = Left, Z = Up	

Environment	Frames	Units			
		Acc	GPS	Gyro	Mag
CoppeliaSim					
Plugin					
PX4					

1.6. Schedule

Tasks	Status
Quadcopter model, Sensor model, Frame transform	**Done**
Communication between ROS and, CoppeliaSim	**Done**
Send sensors data to PX4 using MAVROS	**Done**
Input RC signal and receive control signal from PX4 to test the attitude controller	**To do**

Mission mode test	**To do**
Write simulation start script	**To do**
World environment setup	**To do**



1.7. Common issues

In this part, we list some common issues during the tool install and simulation, and its solutions.

#1