**Advanced Precision Landing**

|  |  |
| --- | --- |
| **Aruco Arrays** |  |
| Aruco arrays provide a set of arrays, in this case two, one larger, one smaller, to aid in more precise precision landing  The vehicle uses the larger marker for the first part of the descent, and then the smaller for the final part to be more accurate | https://courses.dojofordrones.com/courses/1119413/lectures/24002019 |
| For the purposes of this report, we will use an aruco array featuring markers 129 and 72  129 will be 40x40cm, and will have a maximum range of 10m and a minimum of 2m  Aruco 72 will have a max range of 4m, and a min range of 10cm  Notice the crossover between the two ranges, this helps with accuracy by allowing a smooth transition between detection |  |
| **Editing worlds file and testing performance with larger marker** | <https://courses.dojofordrones.com/courses/1119413/lectures/24002017> |
| Locate the ‘worlds’ folder within the ‘ardupilot\_gazebo’ |  |
| Open ‘aruco\_landing.py’ and locate the code pertaining to loading the aruco marker  Change name to ‘Gazebo/Aruco129’ and ‘size’ to ‘.4 .4’ (40cm x 40cm) |  |
| Launch Gazebo-ROS and connect the SiTL firmware to the iris drone, make sure to use --console to launch the MAVProxy console with it  Once firmware is connected and drone is armable, change into LOITER mode and arm the drone  Use rc command ‘rc 3 1700’ to apply throttle to the vehicle and fly it to a height of 5m (use the console for this)  Once drone reaches a height of 5, use rc command ‘rc 3 1500’ to set the throttle in a stable position to allow the drone to hover in place |  |
| Run ‘track\_aruco.py’ script  After running for a short time, take note of when the ‘percentage of frames found’ info getting printed to the console stabilizes  **NOTE:** Make sure the script is looking for marker number 129 |  |
| Halt the script, fly the vehicle to a height of 10m and run ‘track\_aruco.py’ again  Again take note of the percentage of frames found once it stabilizes |  |
| Halt the script and fly to a height of 12m  Run track\_aruco.py and note the percentage  At this height it should be performing much worse, indicating that 12m is probably out of range for the application with a marker that size |  |
| Put the vehicle into ‘land’ mode with the script still running  Once the vehicle has landed, scroll up on the console and note the height (indicated by the z axis variable) that the script stops recognizing the marker  This is the minimum range for a marker of that size |  |
| **Gazebo World with Aruco Array** |  |
| Open the ‘2\_arucos.world’ file located within the ‘worlds’ folder in the ‘ardupilot\_gazebo’ repo |  |
| Confirm that the visuals labeled ‘qrcode1’ and ‘qrcode2’ correspond to aruco markers 129 and 72 with sizes of 40cm x 40cm and 20cm x 20cm  Note the location offset of marker number 72 in the ‘pose’ field |  |
| Launch gazebo-ros and specify the ‘2\_arucos.world’ file  Roslaunch gazebo-ros iris\_world.launch world\_name:=”2\_arucos.world’ |  |
| Note the aruco array that is loaded with this world file |  |