DRDO DRGE'S VISION BASED OBSTACLE AVOIDANCE DRONE

DOCUMENTATION REPORT

Installation Steps:

The following steps assume that the host has Ubuntu 18.04.5 LTS (Bionic Beaver) installed and is either present as a standalone Operating System(OS) or is dual-booted with another OS. If not, download the Ubuntu ISO file: https://releases.ubuntu.com/18.04/ and install the OS. Note, ROS and Gazebo might not work properly if Ubuntu is run on a Virtual Machine or via WSL.

The Installation Guide contains two parts. The first section consists of installing the prerequisite software (ROS+Ardupilot+Gazebo & Plugins) following the steps given in the initial Installation document provided. In case this installation has already been done, it is still preferable to repeat these steps to avoid any unforeseen errors.

All the commands have to be executed serially. Each set of consecutive commands have been separated accordingly.

Each command is preceded with its description(underlined text) for understanding and must be ignored during execution.

Prerequisite Software:

In a new terminal, run the following commands -

ROS Melodic

You can find these installation instructions here.

Setup your sources.list

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

Update packages and install ROS

```
sudo apt update
```

```
sudo apt install ros-melodic-desktop-full
```

Setup the environment

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
```

source ~/.bashrc

Dependencies

sudo apt install python-rosdep python-rosinstall
python-rosinstall-generator python-wstool build-essential

Rosdep

sudo apt install python-rosdep

sudo rosdep init

rosdep update

Ardupilot

Installing Ardupilot and MAVProxy
Clone ArduPilot In home directory:

cd ~

sudo apt install git

git clone https://github.com/ArduPilot/ardupilot.git

cd ardupilot

git checkout Copter-3.6

git submodule update --init --recursive

Install dependencies:

sudo apt install python-matplotlib python-serial python-wxgtk3.0 python-wxtools python-lxml python-scipy python-opencv ccache gawk python-pip python-pexpect

Use pip (Python package installer) to install mavproxy:

sudo pip install future pymavlink MAVProxy

Open ~/.bashrc for editing:

gedit ~/.bashrc

Add these lines to end of ~/.bashrc (the file open in the text editor):

export PATH=$PATH:$HOME/ardupilot/Tools/autotest

export PATH=/usr/lib/ccache:$PATH

Save and close the text editor

Reload ~/.bashrc :

. ~/.bashrc

Run SITL (Software In The Loop) once to set params:

cd ~/ardupilot/ArduCopter

sim_vehicle.py -w

Gazebo and Plugins

Gazebo
Setup your computer to accept software from http://packages.osrfoundation.org:

sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable `lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-stable.list'

Setup keys:

wget https://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -

Reload software list:

sudo apt update

Install Gazebo:

sudo apt install gazebo9 libgazebo9-dev

Install Gazebo plugin for APM (ArduPilotMaster):

cd ~

git clone https://github.com/khancyr/ardupilot_gazebo.git

cd ardupilot_gazebo

git checkout dev

Build and install plugin

mkdir build

cd build

cmake ..

make -j4

sudo make install

echo 'source /usr/share/gazebo/setup.sh' >> ~/.bashrc

Set paths for models:

echo 'export
GAZEBO_MODEL_PATH=~/ardupilot_gazebo/models' >>
~/.bashrc

. ~/.bashrc

Run Simulator

In one Terminal (Terminal 1), run Gazebo:

gazebo --verbose
~/ardupilot_gazebo/worlds/iris_arducopter_runway.world

In another Terminal (Terminal 2), run SITL:

cd ~/ardupilot/ArduCopter/

sim_vehicle.py -v ArduCopter -f gazebo-iris --console

Close all terminals

Additional Software:

Dronekit:

sudo apt-get install python-pip python-dev

pip install dronekit

pip install dronekit-sitl

OpenCV:

pip install opencv-python

pip install opencv-contrib-python

pip install vector

Creating a workspace:

mkdir -p ~/catkin_ws/src

cd ~/catkin_ws/

catkin_make

source devel/setup.bash

Check the correct path:

echo $ROS_PACKAGE_PATH
cd ~/catkin_ws/src

catkin_create_pkg interiit21 std_msgs rospy roscpp

cd ~/catkin_ws

catkin_make

. ~/catkin_ws/devel/setup.bash

Copy all the contents from the interiit21 folder(all the scripts,files,folders) in the extracted zip file & paste it inside the src folder of interiit21 package. They should pe present in the path catkin_ws/src/interiit21/src.

cd ~/catkin_ws/

catkin_make

MAVROS:

sudo apt-get install ros-melodic-mavros
ros-melodic-mavros-extras

cd ~/catkin_ws

wget
https://raw.githubusercontent.com/mavlink/mavros/master/mavros/scripts/install_geographiclib_datasets.sh

chmod a+x install_geographiclib_datasets.sh

./install_geographiclib_datasets.sh

Install rqt:

sudo apt-get install ros-kinetic-rqt ros-kinetic-rqt-common-plugins
ros-kinetic-rqt-robot-plugins

Gazebo ROS:

sudo apt-get install git

cd ~/catkin_ws/src

git clone https://github.com/ros-simulation/gazebo_ros_pkgs.git -b melodic-devel

rosdep update
rosdep check --from-paths . --ignore-src --rosdistro melodic

cd ~/catkin_ws/

catkin_make

source /opt/ros/melodic/setup.bash

source ~/catkin_ws/devel/setup.bash

roscore &
rosrun gazebo_ros gazebo

Open a new terminal:

gedit ~/.bashrc

Add this line to the end of .bashrc file:

export GAZEBO_MODEL_PATH=~/catkin_ws/src/interiit21/src/models

The following 7 lines must be present at the end of the .bashrc file. In case any line is missing, add them at the end of the file.

```
source /opt/ros/melodic/setup.bash
export PATH=$PATH:$HOME/ardupilot/Tools/autotest
export PATH=/usr/lib/ccache:$PATH
source /usr/share/gazebo/setup.sh
export GAZEBO_MODEL_PATH=~/ardupilot_gazebo/models
source ~/catkin_ws/devel/setup.bash
export GAZEBO_MODEL_PATH=~/catkin_ws/src/interiit21/src/models
```

Testing:

Before proceeding with the actual simulation, make sure the following lines have been added in the beginning of the python scripts.

In the catkin_ws/src/interiit21/src folder, open the 3 python scripts, algo.py; rgb.py & follow.py and make sure the following line: '#! /usr/bin/env python' is present in the beginning. If not, add the same in each file.

algo.py rgb.py sample.txt test.py follower

```
#! /usr/bin/env python
from dronekit import connect, VehicleMode, LocationGlobalRelative
from pymavlink import mavutil
from std_msgs.msg import Float64
import rospy
import time
import argparse
from std_msgs.msg import Int64


right = 0
left = 0
forward = 1
right_wall = 0
left_wall = 0
rotating = 0
start = 0
last_wall = "none"
turn = 0
height_up = 0
height_down = 0
count = 0
turned = 0
flag = 0
parser = argparse.ArgumentParser()
parser.add_argument('--connect', default='127.0.0.1:14550')
args = parser.parse_args()
```

Also, make sure that the folder: catkin_ws/src/interiit21/src contains the ‘test.txt’ file which is the calibration file for the RGB camera.

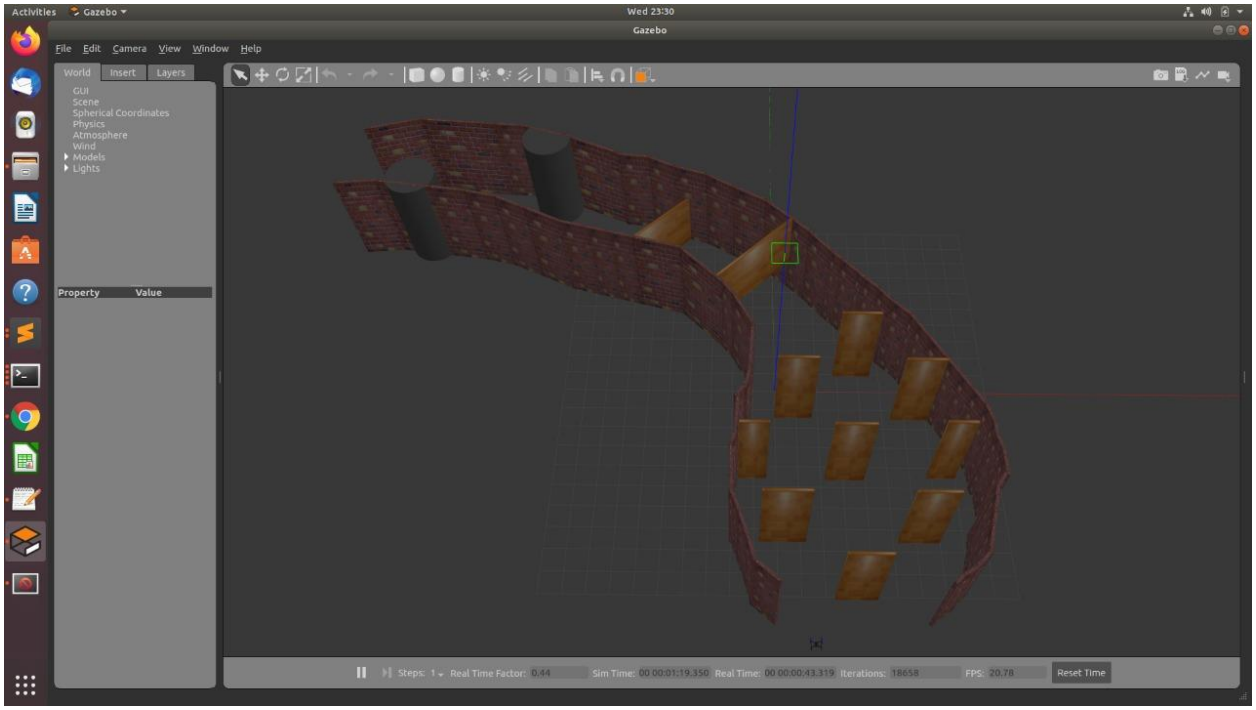
Workspace explained with the changes made in the plugins/parameters:

PointCloudCutoffMax parameter has been changed.

Bash files have been made for directly running a simulation. The evaluators just have to run the bash file corresponding to each world to start the simulation.

For eg: To test our algorithm in the 1st world, open up a new terminal and enter the following command:
sh start_sim1.sh
Similarly run for other 3 worlds.
This should fire up 5 new terminals corresponding to each script. The last terminal shows the status of the mission i.e whether the drone is searching for Aruco Marker or whether it has detected and landed on the same as mentioned in the Problem Statement.
A delay of 20 seconds has been given in launching each script so that proper sync can be maintained between each of them.
In case the bash file doesn't execute properly, execute it again or individually run the 3 scripts after 'roslaunch interiit21 interiit_world1.launch' and after launching the ardupilot.

Algorithm:



Basic Zig-Zag Algorithm is followed. It has a real-time obstacle avoidance feature along with dynamic altitude change in case it detects an obstacle on both sides. The algorithm iteratively runs to cover the entire world coming out on the other side.