eYSIP2017

Navigation In Indoor Environment Using AR Drone 2

Team Members:

Gorantla Balaji
Ridhwan Luthra

Mentors:

Simranjeet
Vamshi

Duration of Internship: 22/05/2017 – 07/07/2017

2017, e-Yantra Publication

Contents

Abstract

Drone applicability is not only limited to the outdoor applications (package delivery, surveillance, target tracking) drones can also be useful in indoor environments for manufacturing/service (e.g. hospitals, greenhouses, production companies and nuclear power plant).

An indoor navigation system is a system that provides navigation within buildings. An indoor navigation system is a system that provides navigation within buildings. The typical GPS that we use cannot provide accurate navigation within a building and for application purpose the hardware parts we use should be of less expensive that is where we tried to avoid expensive sensors like Lidar, 3D cameras, motion capture cameras, etc.,

1.1 Completion status

Navigated the drone from one point to another point in 3D space avoiding obstacles using the given map of the environment in simulation and replicated the navigation in real world scenario successfully.

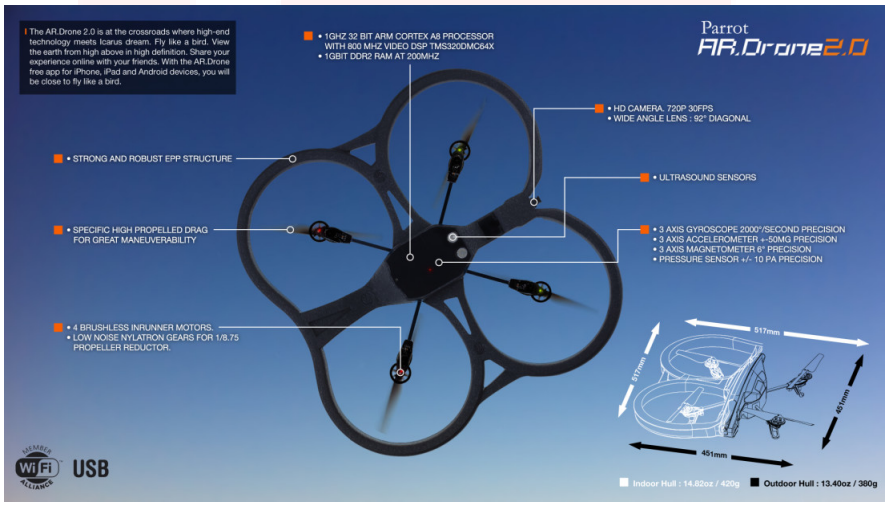
Hardware parts

2.1 List of hardware

- 1. Parrot AR Drone 2.0
- 2. ArUco markers

2.2 Detail of each hardware

- Here is the AR Drone Developer guide for better understanding about AR Drone 2.0.



Specification of AR Drone 2.0

- Here is the ArUco marker guide for the better understanding of creation of markers, marker detection, pose estimation using markers.

Software used

3.1 List of softwares used

We used many packages in this project to navigate drone from one point to another. Namely -

- ardrone_autonomy
- drone_application
- aruco_mapping
- marker_pose_detection
- tum_simulator
- MoveIt!

3.2 Details of software

- ardone_autonomy : This is the driver for AR drone and is necessary to communicate with the drone in simulation and in the physical world.
- drone_application : This is the package we have created for the autonomous navigation of the ardrone 2.0.
- aruco_mapping : This package allows user to create a map of Aruco markers in 2D or 3D space and estimate full 6 DOF pose of the camera. Using Markers to estimate full 6 DOF position only by means of single calibrated camera is well known approach. This package leverages basic aruco_ros functionality and provides tools to create a map of detected markers in 2D/3D space.

- marker_pose_detection : This node publishes a visualization_msgs/Marker on the /Estimated_marker topic. You can Subscribe to this topic to get your location and orientation with respect to the ArUco marker.
- tum_simulator : This package contains the implementation of a gazebo simulator for the Ardrone 2.0.
- MoveIt! : MoveIt! is state of the art software for mobile manipulation, incorporating the latest advances in motion planning, manipulation, 3D perception, kinematics, control and navigation.

3.3 Installation steps

Although ROS has been made to work on a wide variety of systems, we are using Ubuntu 14.04, a popular and relatively user-friendly Linux distribution.

This tutorial assumes that you are familiar with Linux, Python, and ROS. Also that you have already created a workspace(named catkin_ws, if you are using a different name please change this.) and are using catkin built system. This tutorial is built for ubuntu 14.04 and ROS Indigo.
Ubuntu Linux can be downloaded freely from ubuntu.
Go through the instructions to install ROS Indigo

- Individual Package: You can also install a specific ROS package (replace underscores with dashes of the package name):

  sudo apt-get install ros-indigo-PACKAGE

  To find available packages, use:
  apt-cache search ros-indigo

If you want to install the entire functionality, you can go through the complete_install_script

Approach

4.1 Project Flow

Here is a complete flow of our project to make it work successfully.

- Firstly, Installed ROS and necessary pacakages like ardrone_autonomy, tum_simulator for simulation and read the specifications of AR drone and got familierized with AR Drone 2.0.
- Understood the concept of PID Controller and Aruco markers and used them to align the drone to the marker using the camera feed from the quadrotor in simulation and in real.
- Worked on octomap server and generated 3D environment using turtlebot and octomap with a 3D map file.
- Installed Rviz and spawned the quadrotor model, 3D map in Rviz using simulator and also in real.
- Made a literature review about the autonomous navigation.
- Worked on aruco_mapping package to detect the markers for localisation i.e. getting the pose of a drone from the marker.
- Autonomously navigated the drone frome one waypoint to another by using aruco markers and odometry readings.
- Generated a map with the world in accordance to the objects in a room.
- Using aruco markers for localisation, the map of the room for path planning and PID controller navigated the drone autonomously from one waypoint to another avoiding the obstacles in real world.

Software and Code

5.1 Repository

Here is our Github link for the repository of code

5.2 Scripts

Brief explanation of various parts of code :

- ardrone_teleop_key.py :
  Handle Control of drone using keyboard.

- pose.py :
  Class for pose storage.

- localisation.py :
  Handle usage of EKF.

- kalman_filter.py :
  Implimenation of Extended Kalman Filter.

- move_to_waypoint.py :
  Get drone to follow generated Trajectory.

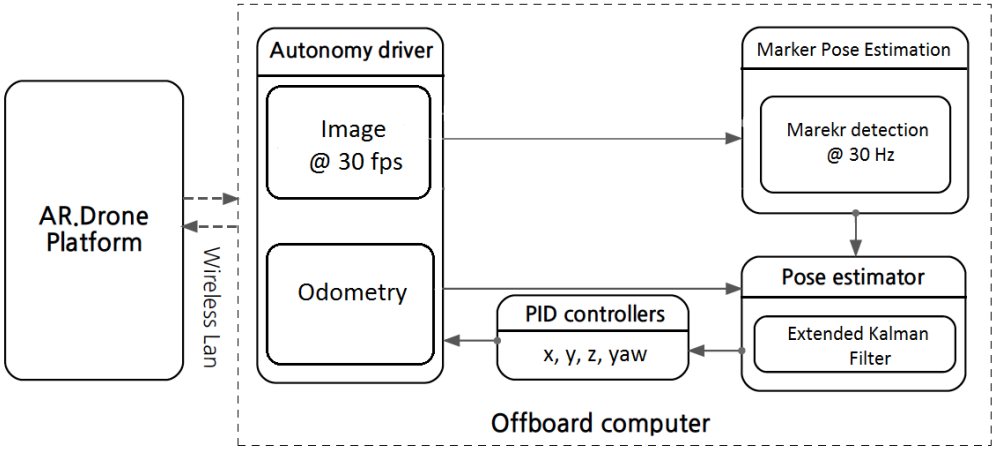
- follow_trajectory.py :
  Extract, generate and send trajectory for excecution.

- pid.py :
  Generic PID implimentation.

- transform_handler.py :
  Generate transform between drone's odom and aruco's world.

Use and Demo

6.1 How it works



6.2 Instruction for demonstrations

- Here is a demonstartion viedo of Emulating AR drone in Rviz and follow instructions to emulate the AR Drone in Rviz
- Here is a demonstartion viedo to Align AR Drone 2.0 to ArUco marker and follow instructions to align the drone with aruco marker.
- Here is a demonstartion viedo of whycon marker follower ardrone in real world and follow instructions to align the drone with whycon marker.
- Here is a demonstartion viedo of real world navigation of ardrone using moveit waypoints and follow the instructions provided in video to move the drone using waypoints generated by moveit!.

Future Work

Our future works to improve our project are:

- Using sensor fusion to get reliable location.
- Applying more robust control systems to handle the dynamics of the quadcopter.
- Generate minimum snap trajectories for the quadrotor.
- Handle dynamic changes in the environment.

Bug report and Challenges

Failures and challenges faced during project:

- moveit
  - moveit is not meant for drones so adapting moveit to work with drones was a challenge.
  - We spent a week to get the entire functionality of moveit adapted for drones.
  - Finally, we used the motion planning from moveit and built our own controller to move the drone.
- Aruco mapping
  - Detection of markers in the aruco mapping package is not reliable when the drone is moving rapidly.
  - Applying PID became and issue as PID needs a derivative and in order for us to find the derivative the function must be continuous. We do not get a continuous function from aruco mapping and hence the edge cases needed to be handled.
- Localisation
  - Using dead reckoning is not an option as the odometry is not at all reliable.
  - The best solution is to apply EKF to fuse the data coming from various sensors like the aruco markers and odometry.
  - This functionality is not fully tested.

Bibliography

[1] Boaz Ben Moshe, Nir Shvalb, Junathan Baadani, Itay Nagar and Harel Levy: *Indoor Positioning and Navigation for micro UAV Drones*, 2012.

[2] Nick Dijikshoorn: *Simultaneous Localization and maping with AR Drone*, July 14,2012.

[3] Anirudha Majumdar and Russ Tedrake: *Funnel Libraries for Real-Time Robust Feedback Motion Planning*, May 2, 2017

[4] Robin Deits and Russ Tedrake: *Efficient Mixed-Integer Planning for UAVs in Cluttered Environments*,

[5] Sergio Garca, M. Elena Lpez, Rafael Barea, Luis M. Bergasa, Alejandro Gmez and Eduardo J. Molinos: *Indoor SLAM for Micro Aerial Vehicles Controlusing Monocular Camera and Sensor Fusion*, 2016

[6] Toma s Krajn k, Mat as Nitsche, Jan Faigl, Petr Vanek, Martin Saska, Libor Preu cil ,Tom Duckett, Marta Mejail: *A Practical Multirobot Localization System*, 6 August, 2013

[7] Benoit Landry: *Planning and Control for Quadrotor Flight through Cluttered Environments*, June, 2015

[8] Daniel Warren Mellinger: *Trajectory Generation and Control for Quadrotors*, 1 January, 2012