

---

# EE565: MOBILE ROBOTICS

## LAB # 5: AR DRONE SETUP WITH ROS AND SENSOR DATA FUSION USING AR DRONE'S ACCELEROMETER AND GYROSCOPE

### IN-LAB WORK

This lab is composed of two parts, Hardware and Simulation. In Hardware experiment students shall acquire IMU sensors data from a real quadrotor, Parrot AR Drone. In the simulation experiment students will work with a gazebo quadrotor model in a virtual environment. **Quadrotors are really dangerous devices; be very careful and don't try to touch rotating propellers! If it fails in the air let it fall!**

### HARDWARE EXPERIMENT:

1. Install ardrone\_autonomy packages found at [{url}](#).
  - **sudo apt-get install ros-indigo-ardrone\_autonomy**
2. Use the following command to launch the quadrotor ROS driver, make sure wireless connection between AR-Drone and Computer is already established
  - **roslaunch ardrone\_autonomy ardrone\_driver \_realtime\_navdata:=False \_navdata\_demo:=0**
3. Use ardrone/navdata topic to acquire sensor information such as orientation, linear and angular velocity
4. To view the live video stream
  - **roslaunch ardrone\_autonomy ardrone\_driver \_realtime\_navdata:=False \_navdata\_demo:=0**
5. Use ardrone/imu topic to acquire raw IMU sensor information, use following command to view a live plot
  - **rqt\_plot /imu/orientation/x:y:z**
6. To take off and land send a std\_msgs/Empty to ardrone/takeoff and ardrone/land respectively
7. To navigate AR-Drone after takeoff send geometry\_msgs/Twist on cmd\_vel topic, value range: -1.0 to 1.0
  - -linear.x: move backward,                   +linear.x: move forward
  - -linear.y: move right,                   +linear.y: move left
  - -linear.z: move down,                   +linear.z: move up
  - -angular.z: turn left,                   +angular.z: turn right
8. Use rosbag to record IMU sensor and magnetometer topic in static condition and **save it for use in lab assignment.**
9. Play the recorded bag and write a node which subscribe to IMU topic and outputs accelerometer and gyroscope measurement.
10. Create a node to subscribe imu topic data, **use the recorded bag file**, and integrate the gyroscope measurements to calculate euler angles.
11. In the above node use accelerometer measurements to calculate roll and pitch angles

### SIMULATION EXPERIMENT:

1. Install gazebo model for quadrotor [{url}](#).
  - **sudo apt-get install ros-indigo- Hector-Quadrotor-demo**
2. Launch gazebo simulation of a quadrotor in a virtual environment and get yourself familiarize with Rviz and gazebo nodes implementing a quadrotor simulation

- **roslaunch hector\_quadrotor\_demo outdoor\_flight\_gazebo.launch**
3. The preferred method of quadrotor navigation is using a joystick, make sure the joystick is connect with the PC
    - **roslaunch hector\_quadrotor\_teleop xbox\_controller.launch**
  4. We can also navigate the simulated quadrotor using the keyboard
    - **rostopic pub ardrone/cmd\_vel geometry\_msgs/Twist -1 -- [0,0,0.1] [0,0,0] && roslaunch teleop\_twist\_keyboard teleop\_twist\_keyboard.py**
  5. ROS has a built-in node for 3D pose estimation using extended Kalman filter called robot\_pose\_ekf [{url}](#) if by default it is not installed use following command to install it
    - **sudo apt-get install ros-indigo-robot-pose-ekf**
 It performs a loosely coupled integration of odometry, IMU and visual odometry data for pose estimation. Download the odometry publisher node (**ardrone\_odometry.cpp**) from LMS required to implement robot-pose-ekf. Modify the launch file as per given instructions.

## LAB ASSIGNMENT

- 1) To get the understanding of Kalman filter we shall implement it in a simple case where the quadrotor is stationary. Suppose we wish to filter accelerometer value which is almost constant except some small random noise. Therefore, the process model is as follows

$$x_t = x_0 + \mathcal{N}(0, \sigma_p^2)$$

Accelerometer measurements are also subjected to random noise, therefore, the measurement model is as follows

$$y_t = x_t + \mathcal{N}(0, \sigma_m^2)$$

Write a simple node which can subscribe to IMU topic and able to separately filter the three accelerometer values using Kalman filter methodology. Publish the estimated state and variance as a custom message consists of two fields. Using rqt\_plot plot the published message.

- a) Record the accelerometer measurements and measure the variance of accelerometer readings.
  - b) Since the measurement variance is fixed, observe the behavior of filter using different process noise variance
  - c) Now observe the estimated state and its variance using different initial values of the state and its variance.
- 2) Calculate Euler angles for an Attitude and Heading Reference System (AHRS) using gyro-rate sensor, accelerometer and magnetometer.

- a) Calculate the Euler angles from gyroscope's body-rate measurements as follows

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_t = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{t-1} + \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \cdot \Delta t$$

- b) The roll and pitch angle from accelerometer can be calculated as follows

$$\begin{bmatrix} \phi \\ \theta \end{bmatrix} = \begin{bmatrix} \text{atan2}(a_y, a_x) \\ -\text{atan2}\left(ax, \sqrt{a_y^2 + a_z^2}\right) \end{bmatrix}$$

- c) The yaw angle from the magnetometer readings can be calculated as follows

$$\begin{aligned} y &= m_y \cos(\phi) - m_z \sin(\phi) \\ x &= m_x \cos(\theta) + m_y \sin(\theta) \sin(\phi) + m_z \sin(\theta) \cos(\phi) \\ \psi &= -\text{atan2}(y, x) \end{aligned}$$