Inertial Odometry using AR Drone's IMU and calculating measurement's covariance

Welcome

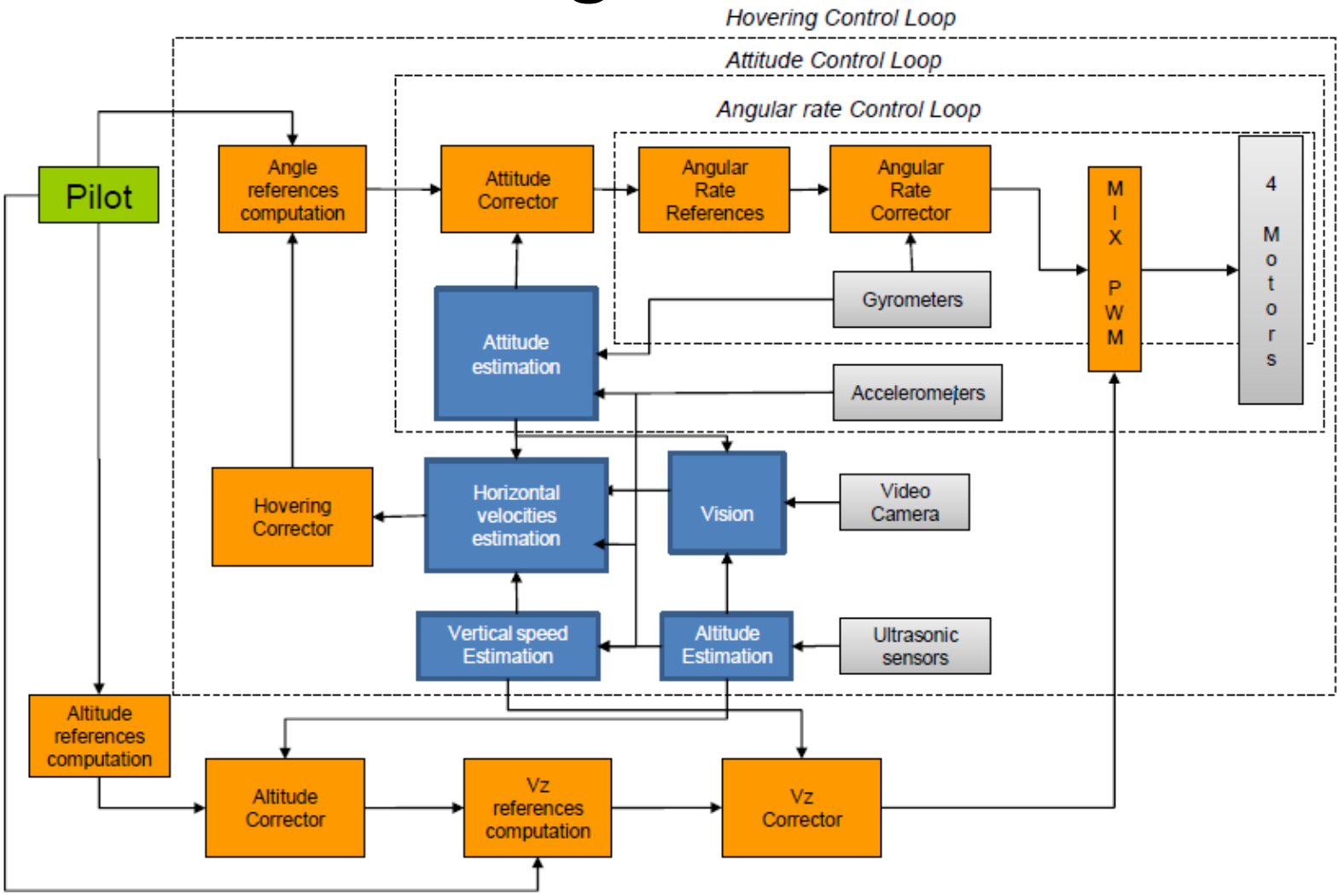
Lab 6

Dr. Ahmad Kamal Nasir

Today's Objectives

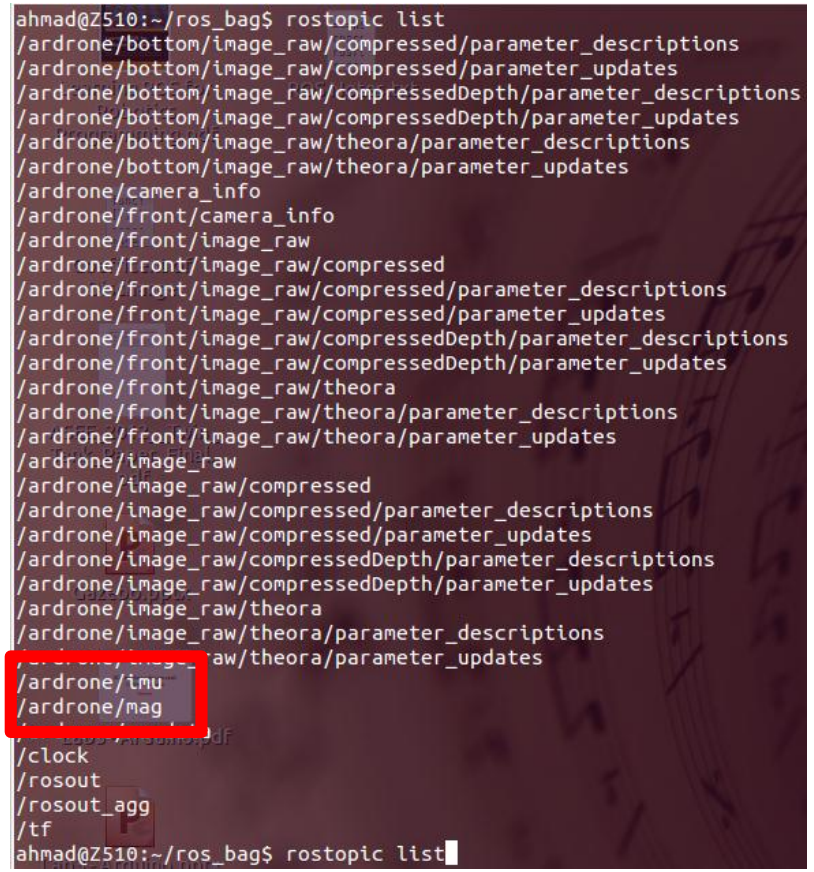
- Introduction to AR-Drone On-board control algorithm
- Orientation/Attitude Estimation using inertial sensors
  - Euler angles from gyroscope
  - Roll, Pitch angles from accelerometer
  - Yaw angle from magnetometer
- Inertial odometry
  - Linear acceleration from accelerometer
  - Measurements covariance

AR Drone's On-board Control Algorithm



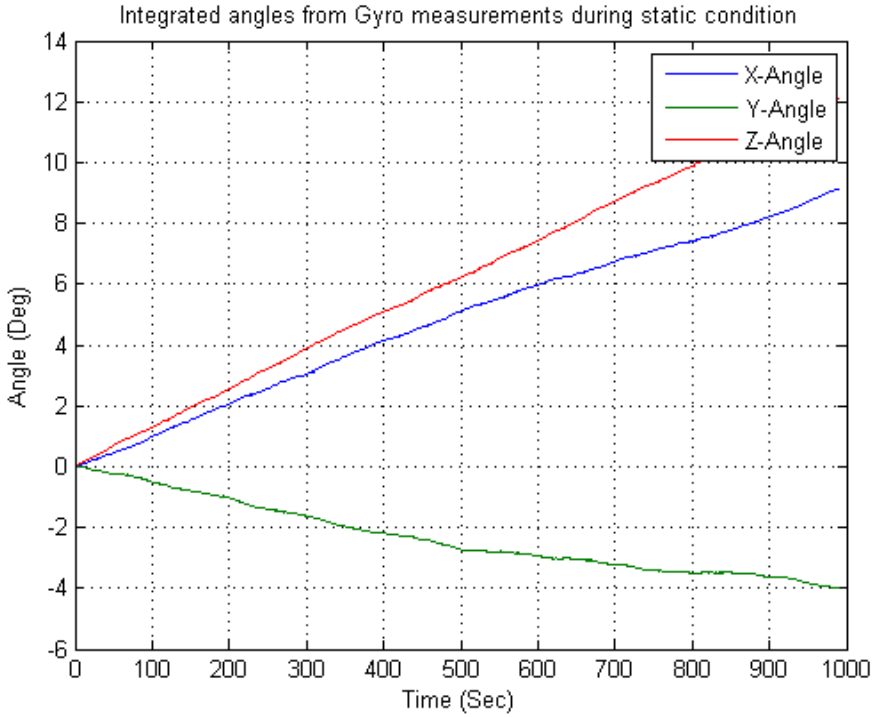
AR-Drone with ROS

- Install ardrone_autonomy packages found at
  - **sudo apt-get install ros-indigo-ardrone_autonomy**
- Use the following command to launch the quadrotor ROS driver, make sure wireless connection between AR-Drone and Computer is already established
  - **rosrun ardrone_autonomy ardrone_driver _realtime_navdata:=False _navdata_demo:=0**



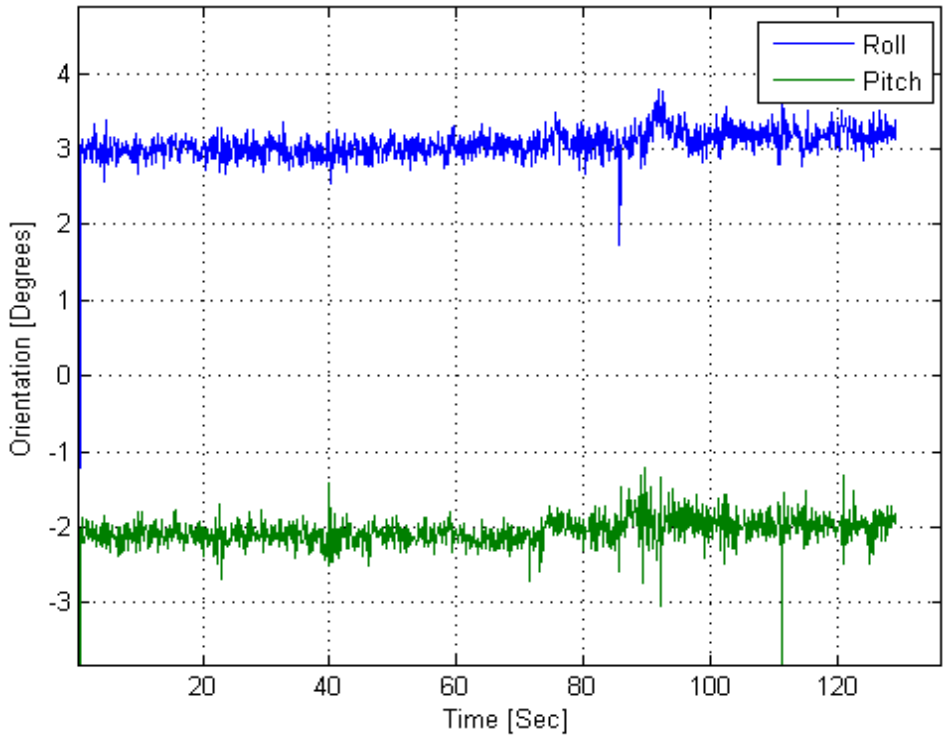
Euler Angles From Gyroscope (Review)

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



Roll, Pitch angles from Accelerometer (Review)

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



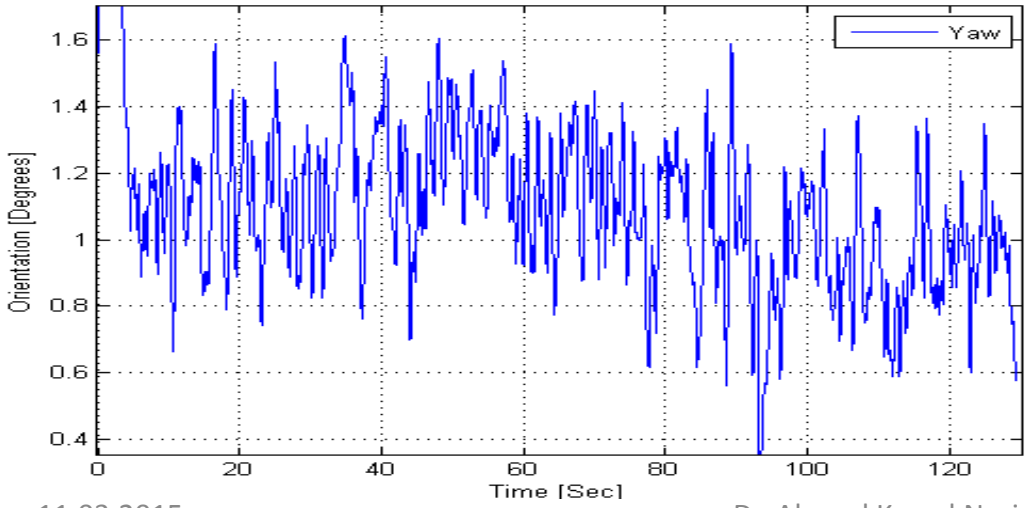
Yaw angle from magnetometer (Review)

$$M_b = R_x(\phi) \cdot R_y(\theta) \cdot R_z(\psi) \cdot M_i$$

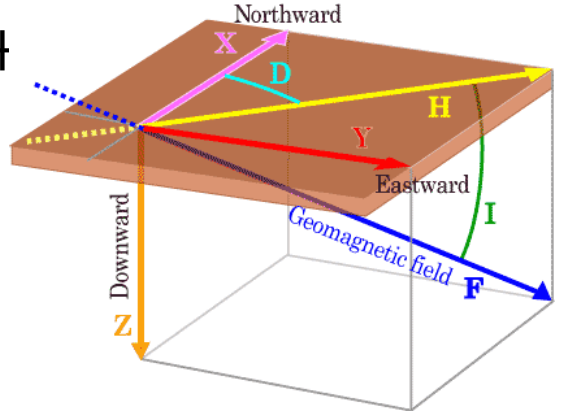
$$\begin{bmatrix} (m_x)\cos(\theta) + (m_y)\sin(\phi)\sin(\theta) + (m_z)\cos(\phi)\sin(\theta) \\ (m_y)\cos(\phi) - (m_z)\sin(\phi) \\ -(m_x)\sin(\theta) + (m_y)\sin(\phi)\cos(\theta) + (m_z)\cos(\phi)\cos(\theta) \end{bmatrix}$$

$$= \begin{bmatrix} B \cdot \cos(\delta) \cdot \cos(\psi) \\ -B \cdot \cos(\delta) \cdot \sin(\psi) \\ B \cdot \sin(\delta) \end{bmatrix}$$

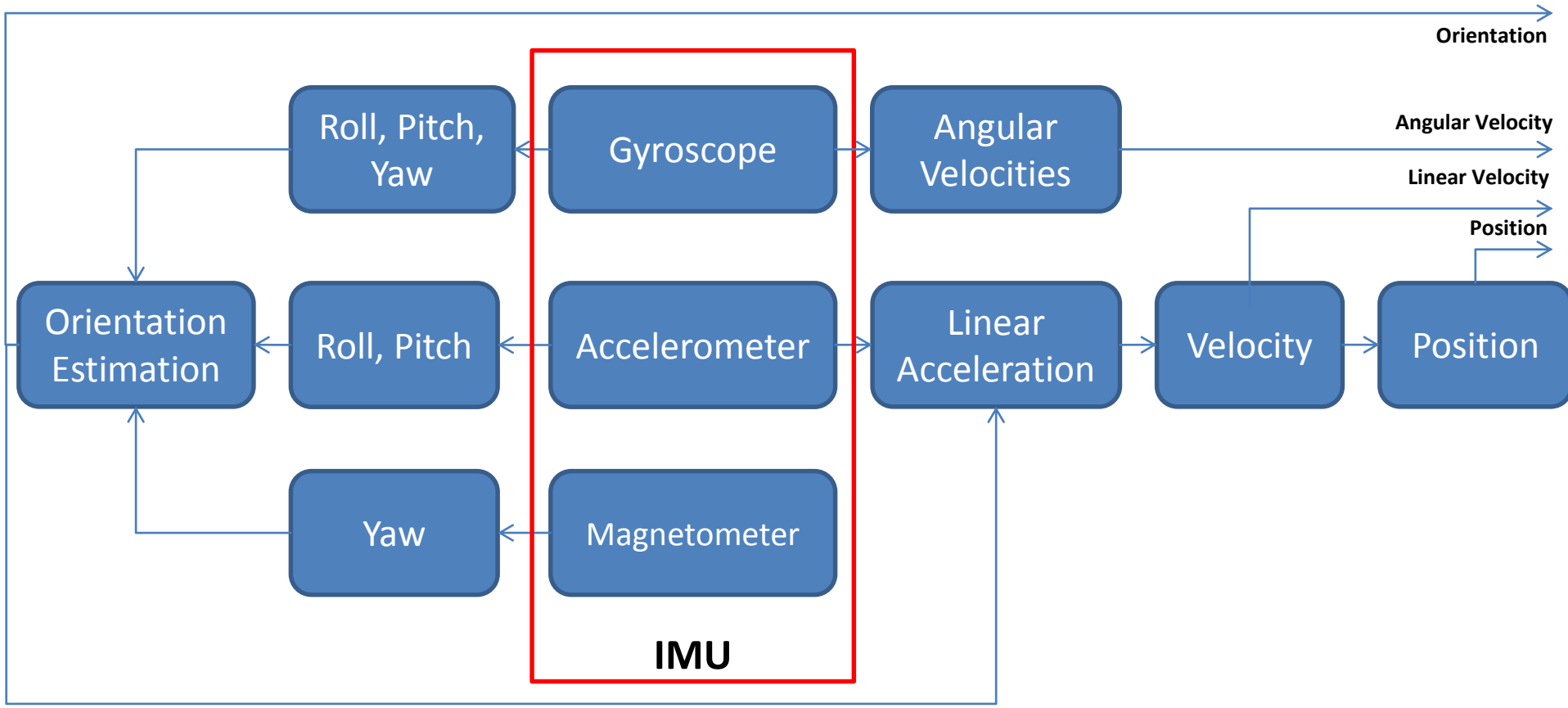
$$y = m_y \cos(\phi) - m_z \sin(\phi)$$



) +



Inertial Odometry



Complementary Filter

$$\theta_t = \alpha \cdot (\theta_{t-1} + \omega_t \cdot \Delta t) + (1 - \alpha) \cdot \theta_a$$

Where

$\theta_t = Current\ estimate$

$\theta_{t-1} = Previous\ estimate$

$\Delta t = Sampling\ interval$

$\theta_a = Accelerometer\ measurements$

$\omega_t = Gyroscope\ measurements$

$\alpha = Weighting\ constant$

Iterative Complementary Filter

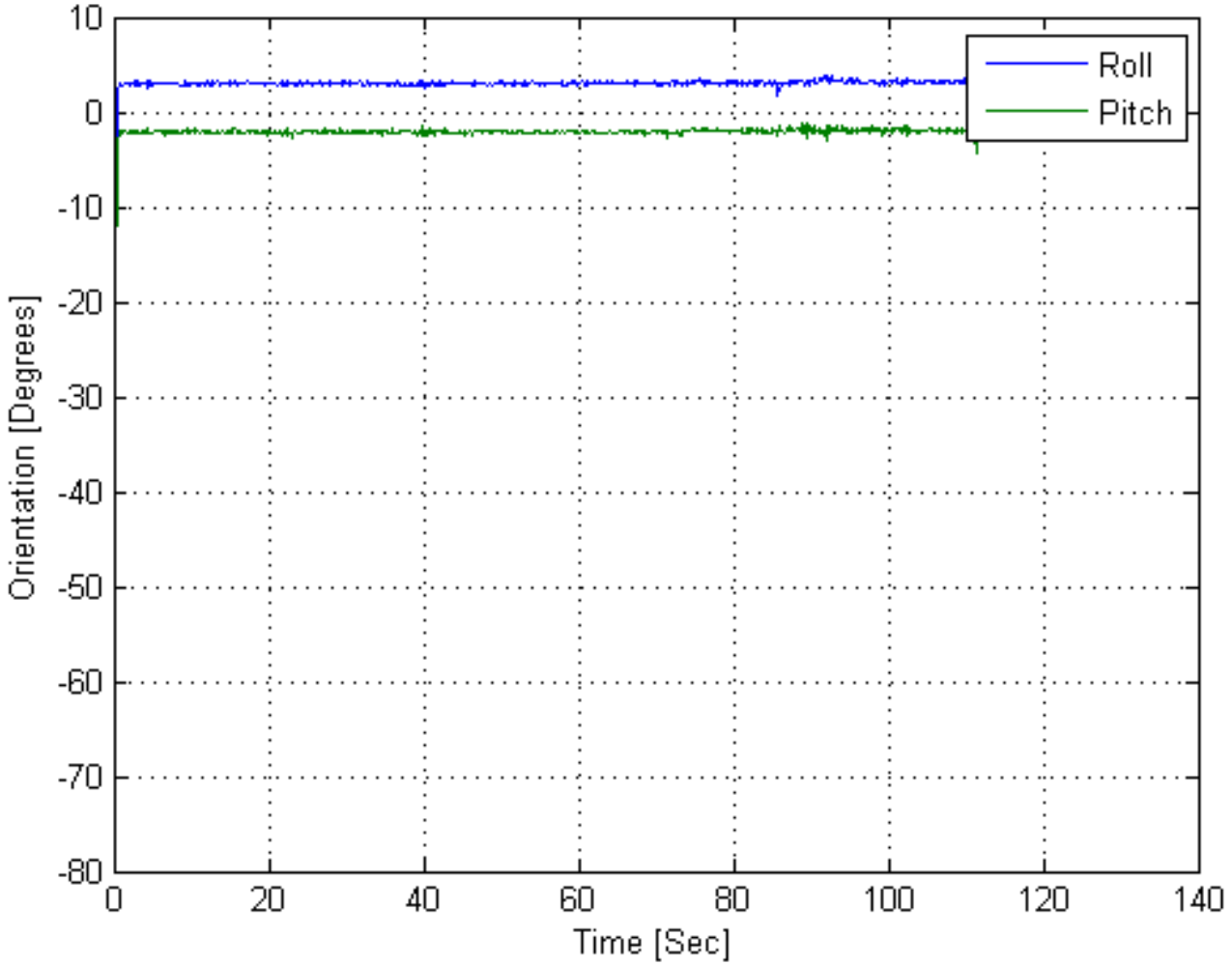
Prediction

$$\theta_t^+ = \theta_{t-1} + \omega_t \cdot \Delta t$$

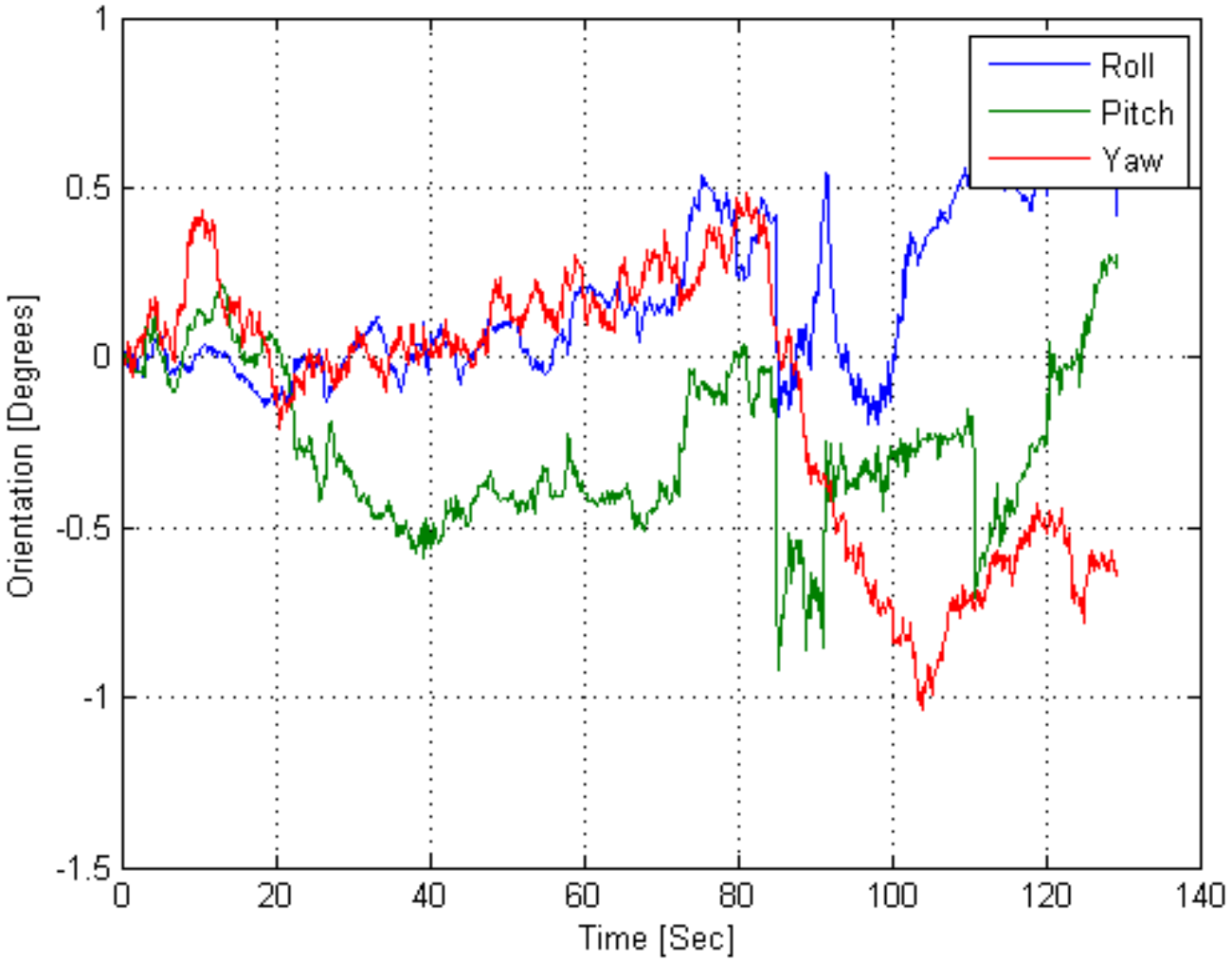
Correction

$$\theta_t = \theta_t^+ + \alpha \cdot (\theta_a - \theta_t^+)$$

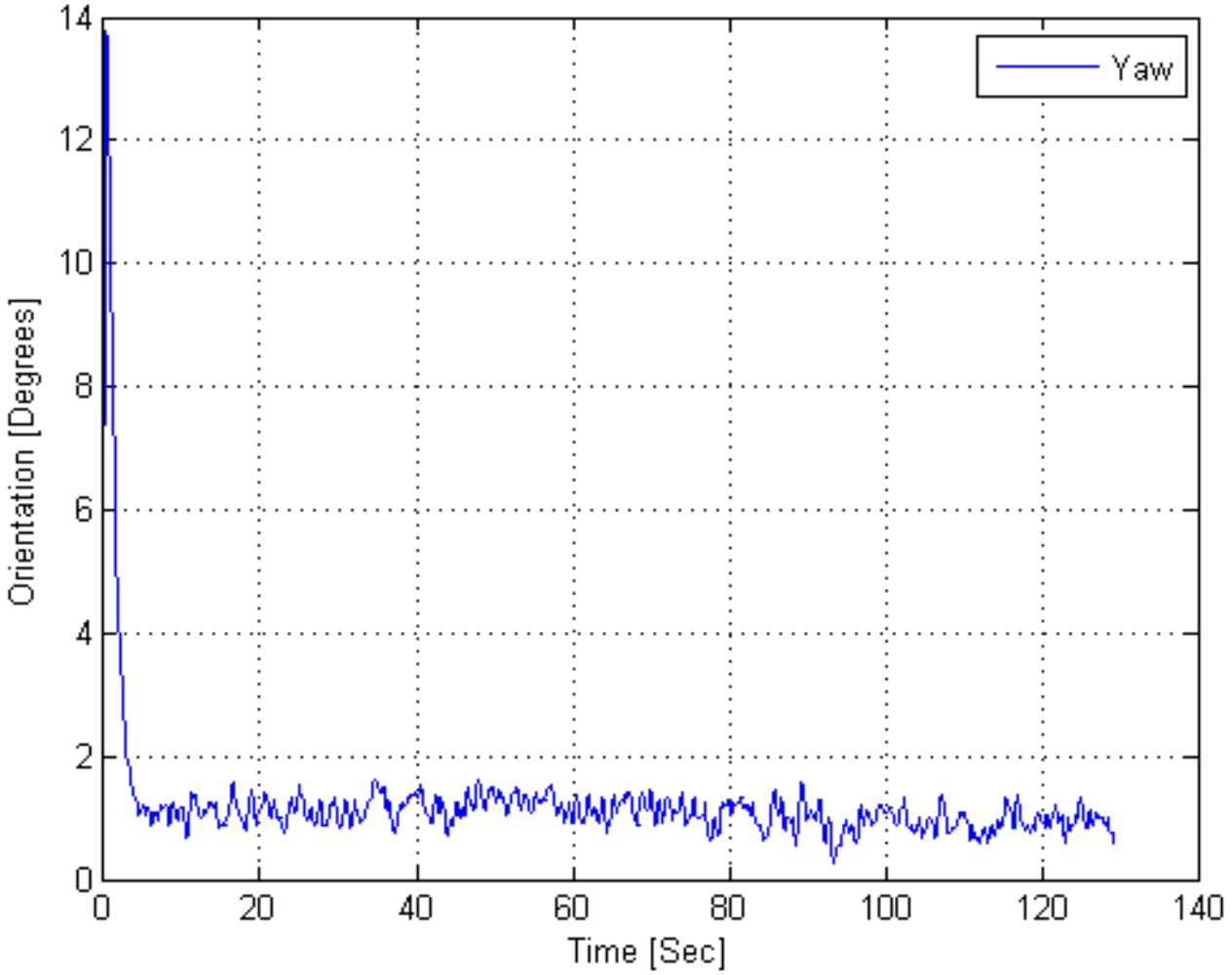
Accelerometer Only



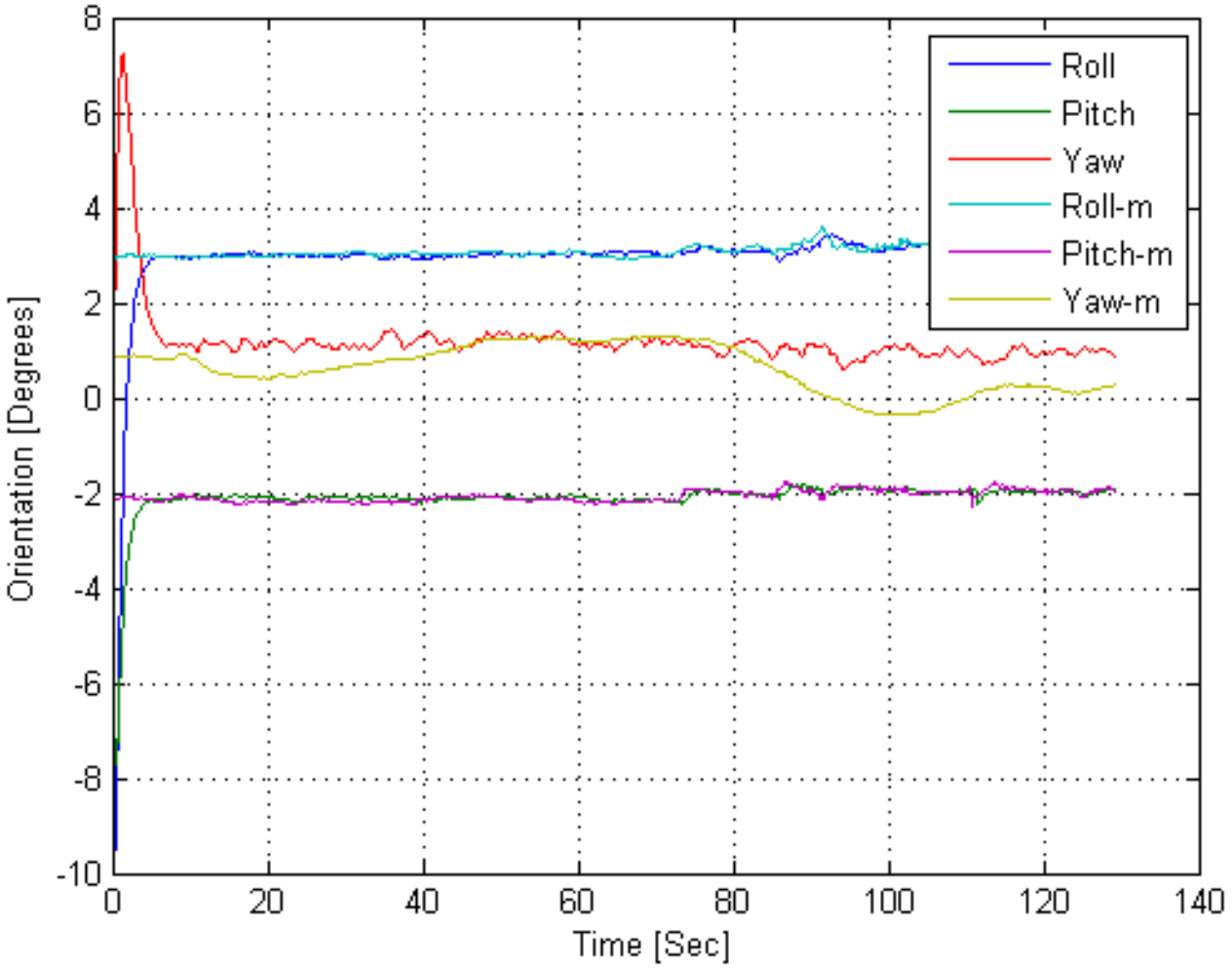
Gyroscope Only



Magnetometer Only



Fused (Complementary Filter)



In-Lab Tasks

- Start the AR Drone, make wireless connection, use ardrone/navdata topic to acquire sensor information.
- Write code for a node that reads the AR Drone navdata. You'll get linear acceleration, gyro-rate and magnetometer readings about x, y and z axes.
- Now place the quadrotor in a static state. Ideally your IMU readings should give constant values.
- Estimate Euler angles (orientation) upon each callback (accelerometer + magnetometer):
  - Using the linear acceleration (accelerometer) readings, find the Roll and Pitch angles (ϕ_a, θ_a). [See Class Lecture 6]
  - Once you have the Roll and Pitch angles, find the Yaw angle (ψ_m) using Magnetometer readings. [Slide 44]
- Use Gyro readings (p, q, r) to get fused Euler angles (ϕ, θ, ψ), upon each callback. Here you'd need values of $(\phi_a, \theta_a, \psi_m)$ upon each iteration. Basically, implement the following equations (E.g. choose $\alpha = 0.8$) : [Slide 33]

$$\begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}_t = \alpha \begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}_{t-1} + (1-\alpha)\begin{bmatrix}\phi_{am}\\ \theta_{am}\\ \psi_{am}\end{bmatrix}_t \quad for \quad 0 < \alpha < 1$$

$$\begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}_t = \begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}_{t-1} + \left(\begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix}\right)_t \Delta t$$

- Δt can be found using timestamps.
- Publish fused Euler angles as an odometry message and visualize in RViz. Assume position to be $(0, 0, 0)$. Verify your implementation by moving and rotating the quadrotor by hand.
- Remember to record a bagfile having IMU data, it will be used in Lab Assignment.

Lab Assignment

- ***INERTIAL ODOMETRY***
- Estimate the position (pose) along the three axes from accelerometer:
  - Convert the accelerometer readings from body frame to inertial frame. Use the Euler angles found in-Lab. [Lecture 6, Slide 22]
  - Implement rectangular integration step to find linear velocities from linear accelerations (in inertial frame). You'd need to find Δt using timestamps.
  - Once you have linear velocities from accelerometer readings, find the position along the three axes by integration again.
- ***MEASUREMENT COVARIANCE***
- Estimate the covariance matrix for estimated Roll, Pitch and Yaw. For this, run your code while keeping the robot in a static state. Record readings for about 60 seconds, and find all entries of the mean vector (3 x 1) and covariance matrix of the Euler angles statistically.
- Take a few different values of α and see which gives most certain estimates.

Questions