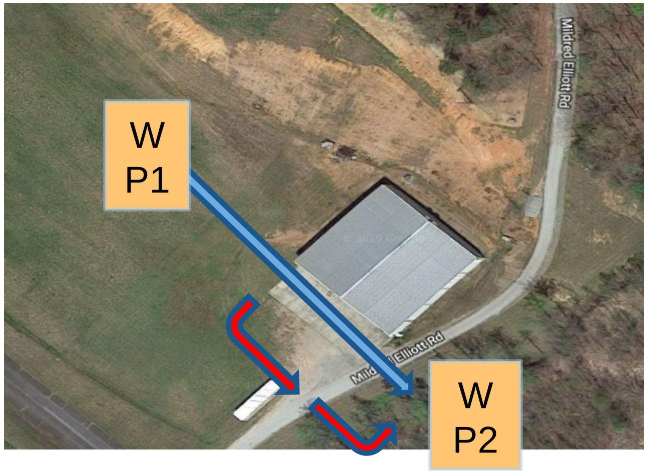
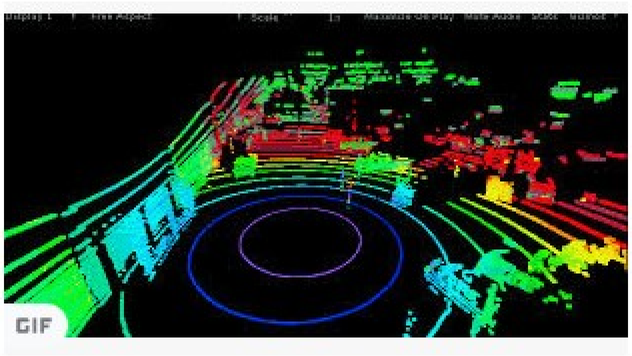
UAV Collision Avoidance by Using a LiDAR

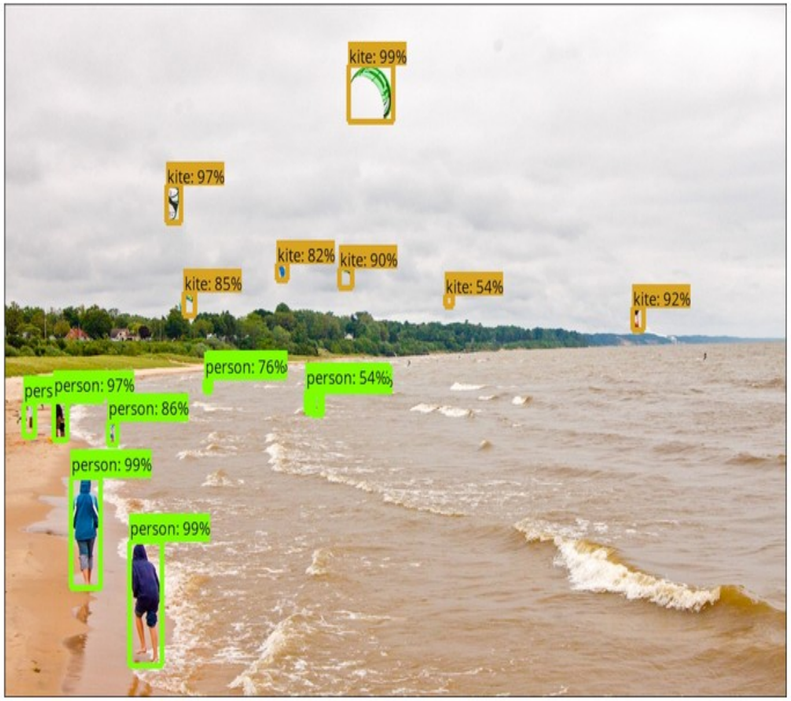
- A Velodyne LiDAR measures range data of up to 100 meters (360 degrees horizontally and 30 degrees vertically).
- Vector field histogram (VFH) generates local path planning by using histogram on grids nearby the UAV.



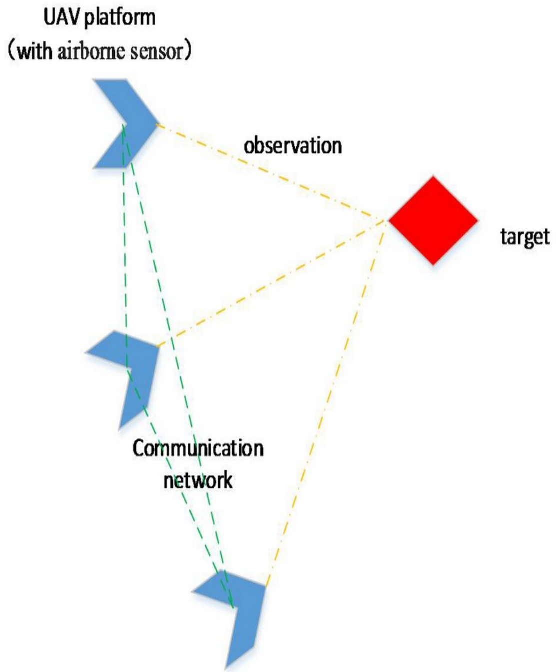


www.unitycoder.com

Cooperative Target Positioning

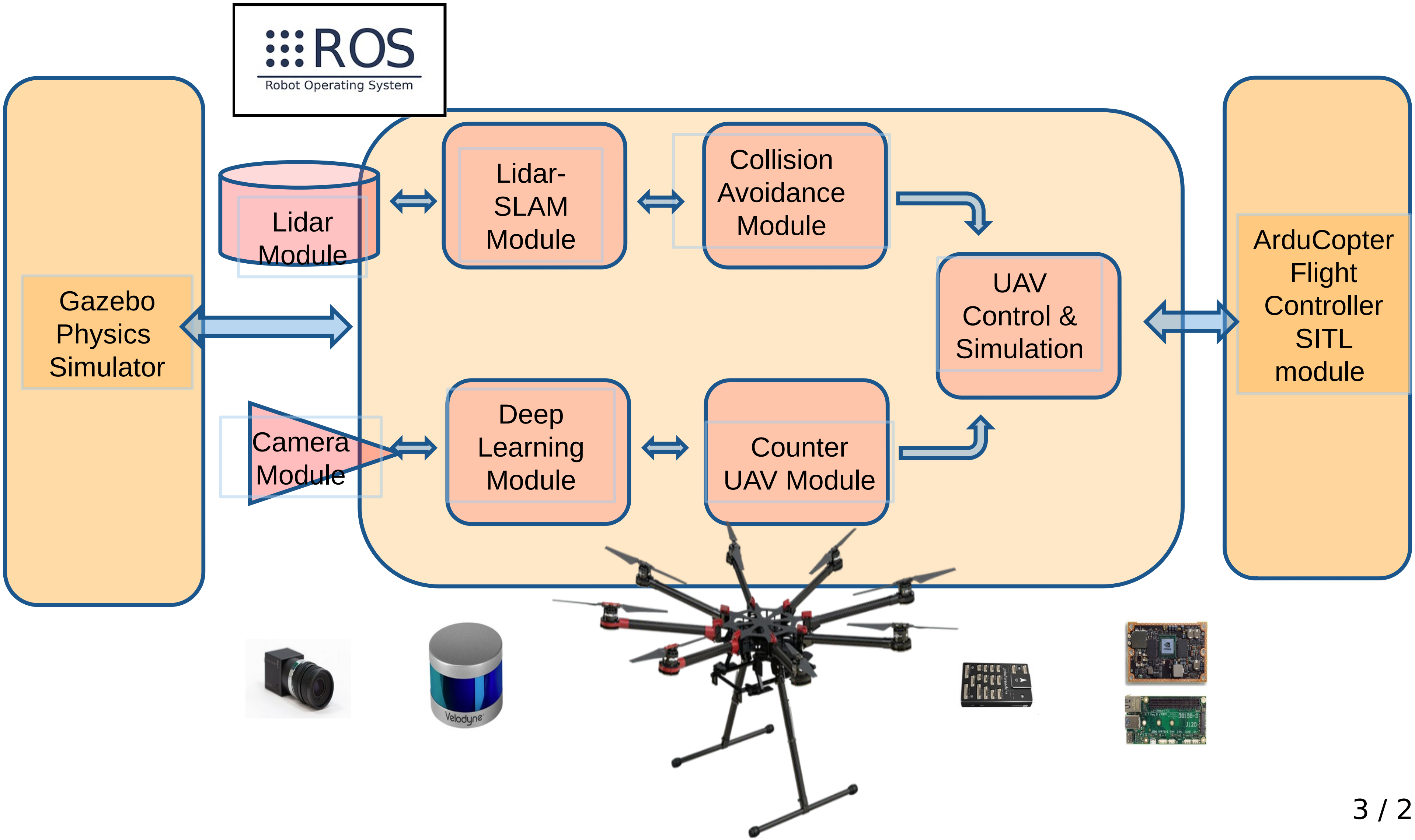


https://github.com/tensorflow/models/tree/master/research/object_detection

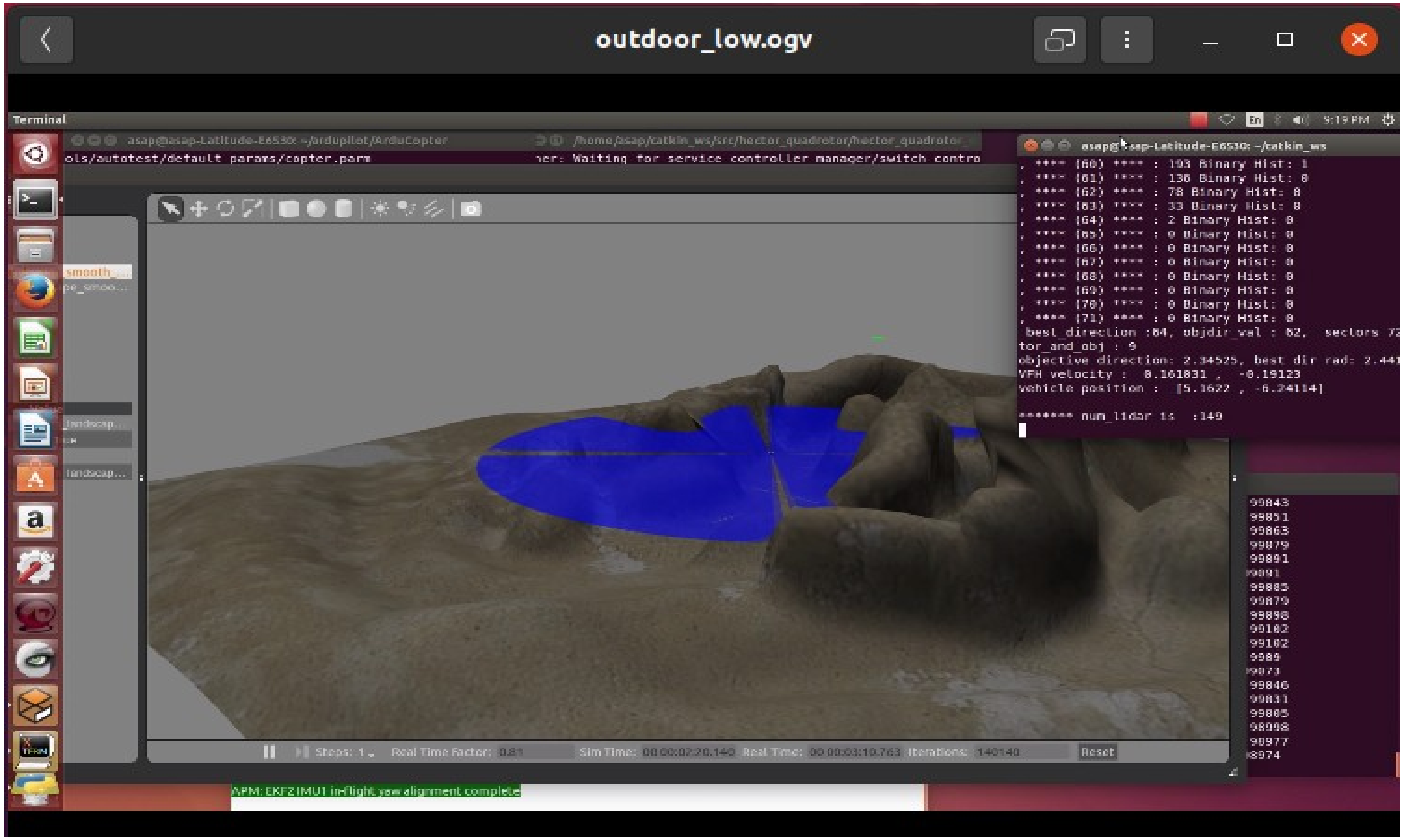


https://www.mdpi.com/2076-3417/8/6/870/htm

UAV System Configuration Example

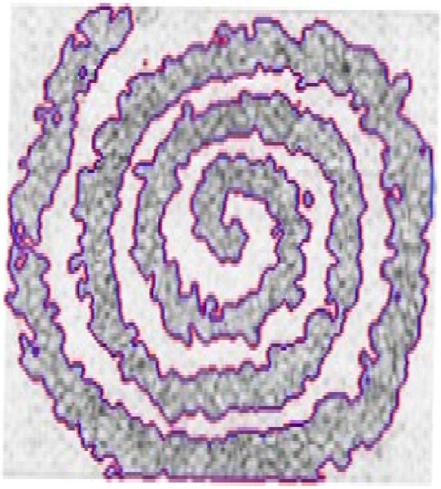


High Fidelity Simulation Enviornment



- https://youtu.be/UDUXXEQqkv8

Visual Tracking and Formation Flight (GA Tech)



- Region-based active contours (RAC) (by Chan-Vese 01)

$$\begin{aligned} E(c_1, c_2, \Psi) &= \mu \int_\Omega \delta(\Psi(x,y)) \|\nabla \Psi(x,y)\| dxdy \\ &+\lambda_1 \int_\Omega |u_0(x,y) - c_1|^2 H(\Psi(x,y)) dxdy \\ &+\lambda_2 \int_\Omega |u_0(x,y) - c_2|^2 (1 - H(\Psi(x,y))) dxdy \end{aligned}$$

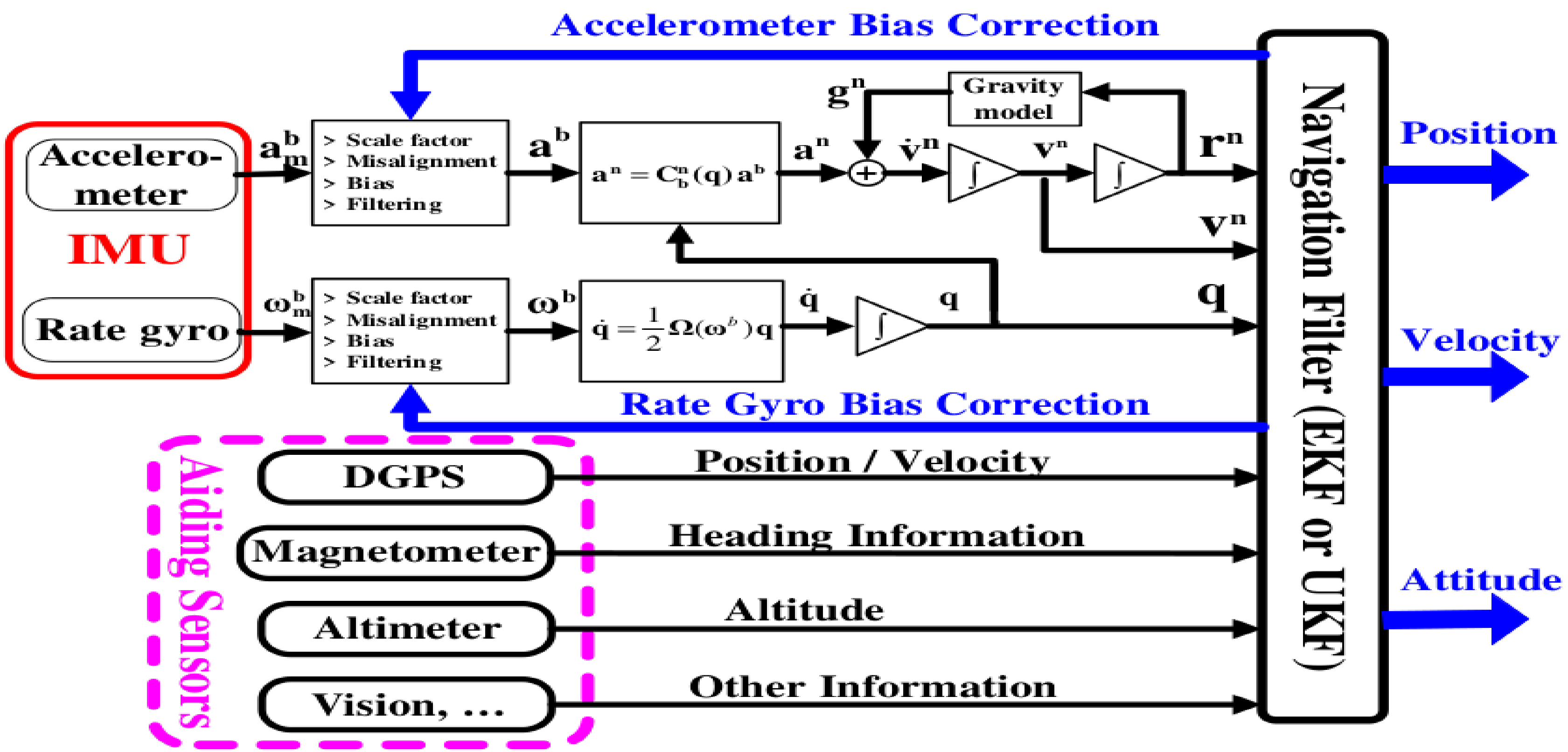
- Fast implementation RAC
- Fast Levelset implementation
- Successful vision-based formation flight in 2007.

Inertial Navigation System (INS)

- Integrating acceleration and angular rate from IMU drift away over time from the true trajectory because of various measurement noise sources.
- By combining IMU with aiding sensors can solve the time-degrading accuracy problem of INS.

Integrated INS Navigation

An integrated navigation system (INS) is designed by using EKF with sequential measurements updates to estimate own-ship states.



[1] INS system diagram from Seung-Min Oh' thesis, "Nonlinear Estimation for Vision-Based Air-to-Air Tracking", Georgia Institute of Technology, 2007.

Nonlinear Discrete Process Model

The state space include position(3), velocity(3), quaternion(4), acceleration bias(3), and rate gyro bias(3).

$$\begin{bmatrix} \mathbf{r}^n_{k+1} \\ \mathbf{v}^n_{k+1} \\ \mathbf{q}_{k+1} \\ \mathbf{b}_{a,k+1} \\ \mathbf{b}_{\omega,k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^n_k + \mathbf{v}^n_k \, \Delta t \\ \mathbf{v}^n_k + \left[\, \mathbf{C}^n_b(\mathbf{q}_k) \left(\, \bar{\mathbf{a}}^b_k - \Delta \bar{\mathbf{a}}^b_{imu,k} \, \right) + \mathbf{g}^n \, \right] \Delta t \\ \left[\, \mathbf{I}_{4\times4} + \frac{1}{2}\boldsymbol{\Omega}(\bar{\boldsymbol{\omega}}^b_k) \, \Delta t \, \right] \, \mathbf{q}_k \\ \mathbf{b}_{a,k} \\ \mathbf{b}_{\omega,k} \end{bmatrix} + \begin{bmatrix} \Delta t \, \mathrm{I}_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -\Delta t \, \mathrm{C}^n_b(\mathrm{q}_k) & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} & -\frac{\Delta t}{2}\mathrm{Z}(\mathrm{q}_k) & 0_{4\times3} & 0_{4\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \Delta t \, \mathrm{I}_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \Delta t \, \mathrm{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \mathrm{n}_{r,k} \\ \mathrm{n}_{a,k} \\ \mathrm{n}_{\omega,k} \\ \mathrm{n}_{b_a,k} \\ \mathrm{n}_{b_\omega,k} \end{bmatrix}$$

Since IMU measurement rate is relatively higher than other aiding sensors, they are considered to be a continuous data flow. Moreover, they are treated as the inputs to the process model. [1]

Sequential Measurement Updates (SMU)

- When measurements come from different sensors, the measurement noise at time t(k) are usually uncorrelated. In this case, the measurement covariance matrix R(k) becomes block-diagonal, in which each diagonal block corresponds to each sensor measurement. [1]
- In this framework, multiple aiding sensor measurements with different size and update rates are easily fused with basic high-rate IMU processing.
- The SLAM simulator included IMU(100Hz), GPS(10Hz), vision(10Hz), magnetometer (20Hz), barometer (5Hz), and air-speed sensor (5Hz).

Trajectory Tracking Controller

- Desired acceleration to a waypoint:

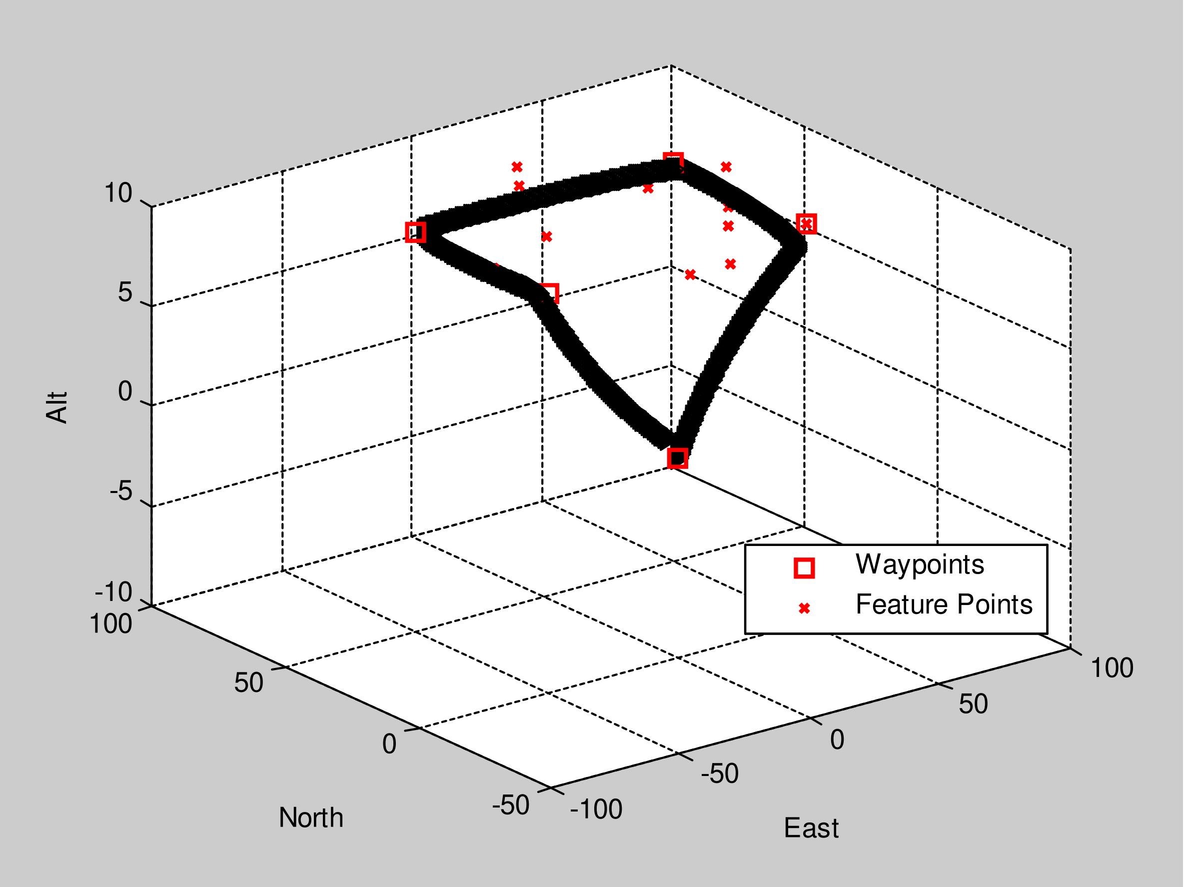
$$\ddot{x}_{\mathrm{des}} = K_x \left(x_{\mathrm{des}} - x \right) + K_{\dot{x}} \left(\dot{x}_{\mathrm{des}} - x \right)$$
$$\ddot{y}_{\mathrm{des}} = K_y \left(y_{\mathrm{des}} - y \right) + K_{\dot{y}} \left(\dot{y}_{\mathrm{des}} - y \right)$$
$$\ddot{z}_{\mathrm{des}} = K_z \left(z_{\mathrm{des}} - z \right) + K_{\dot{z}} \left(\dot{z}_{\mathrm{des}} - z \right)$$

- Desired attitude angles and angular velocities

$$q_{\mathrm{cmd}} = \Theta_{\mathrm{des}} - \Theta$$
$$p_{\mathrm{cmd}} = \Phi_{\mathrm{des}} - \Phi$$
$$r_{\mathrm{cmd}} = \mathrm{AC} \left(\Psi_{\mathrm{des}} - \Psi \right)$$

$$\dot{p} = \frac{1}{\tau_p} \left(p_{\mathrm{cmd}} - p \right)$$
$$\dot{q} = \frac{1}{\tau_q} \left(q_{\mathrm{cmd}} - q \right)$$
$$\dot{r} = \frac{1}{\tau_r} \left(r_{\mathrm{cmd}} - r \right)$$

3D Trajectory for Testing



DGPS Position and Velocity Measurement Model

GPS position and velocity measurements have different latency.

$$\mathbf{y}_k^1 = \mathbf{h}^1(\mathbf{x}_k) + \mathbf{n}_{r,k}^{gps} \quad \Leftrightarrow \quad \mathbf{r}_k^{gps} = \mathbf{r}_{k-L1}^n + \mathbf{C}_b^n(\mathbf{q}_{k-L1})\,\mathbf{r}_{gps}^b + \mathbf{n}_{r,k}^{gps},$$

$$\mathbf{y}_k^2 = \mathbf{h}^2(\mathbf{x}_k) + \mathbf{n}_{v,k}^{gps} \quad \Leftrightarrow \quad \mathbf{v}_k^{gps} = \mathbf{v}_{k-L2}^n + \mathbf{C}_b^n(\mathbf{q}_{k-L2})\,\bar{\boldsymbol{\omega}}_{k-L2}^b \times \mathbf{r}_{gps}^b + \mathbf{n}_{v,k}^{gps},$$

GPS measurement Jacobians.

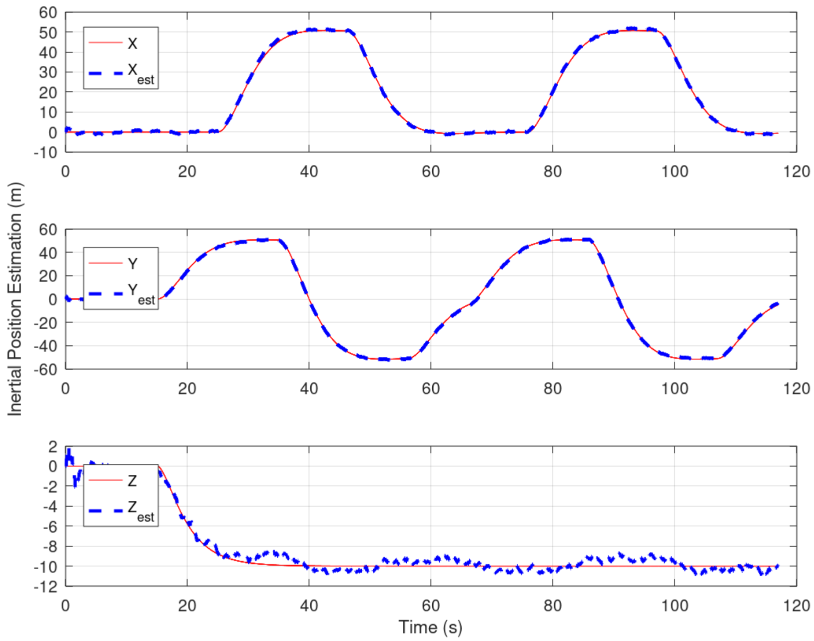
$$\mathbf{H}^1 = \frac{\partial \mathbf{h}^1}{\partial \mathbf{x}} = \left[\left(\frac{\partial \mathbf{h}^1}{\partial \mathbf{r}^n}\right)_{3\times 3} \quad \mathbf{0}_{3\times 3} \quad \left(\frac{\partial \mathbf{h}^1}{\partial \mathbf{q}}\right)_{3\times 4} \quad \mathbf{0}_{3\times 3} \quad \mathbf{0}_{3\times 3} \right],$$

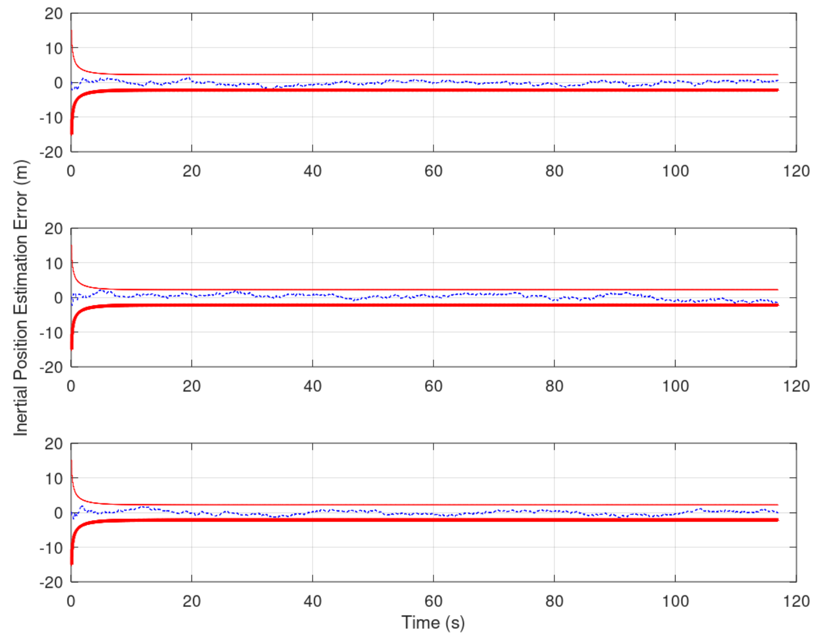
$$\mathbf{H}^2 = \frac{\partial \mathbf{h}^2}{\partial \mathbf{x}} = \left[\mathbf{0}_{3\times 3} \quad \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{v}^n}\right)_{3\times 3} \quad \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{q}}\right)_{3\times 4} \quad \mathbf{0}_{3\times 3} \quad \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{b}_\omega}\right)_{3\times 3} \right],$$

where

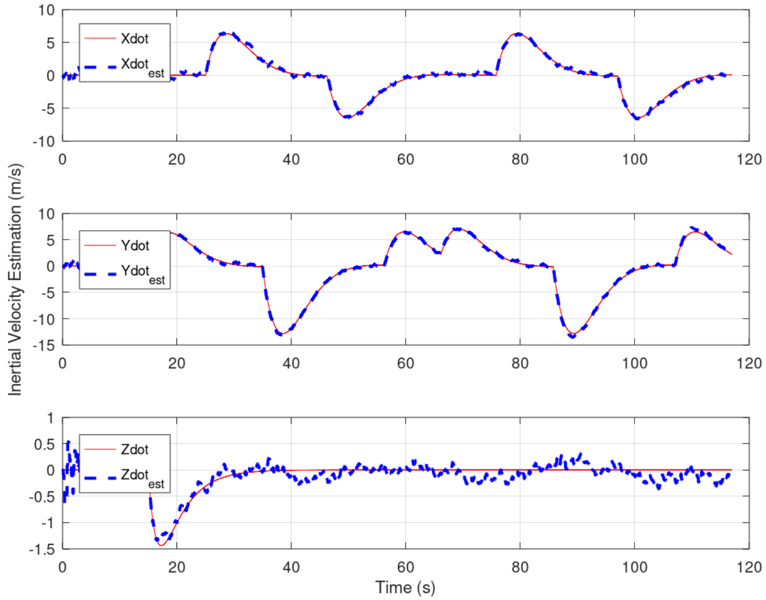
$$\left(\frac{\partial \mathbf{h}^1}{\partial \mathbf{r}^n}\right)_{3\times 3} = \mathbf{I}_{3\times 3}, \quad \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{v}^n}\right)_{3\times 3} = \mathbf{I}_{3\times 3}, \quad \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{q}}\right)_{3\times 4} = \mathbf{0}_{3\times 4}, \quad \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{b}_\omega}\right)_{3\times 3} = \mathbf{0}_{3\times 3},$$

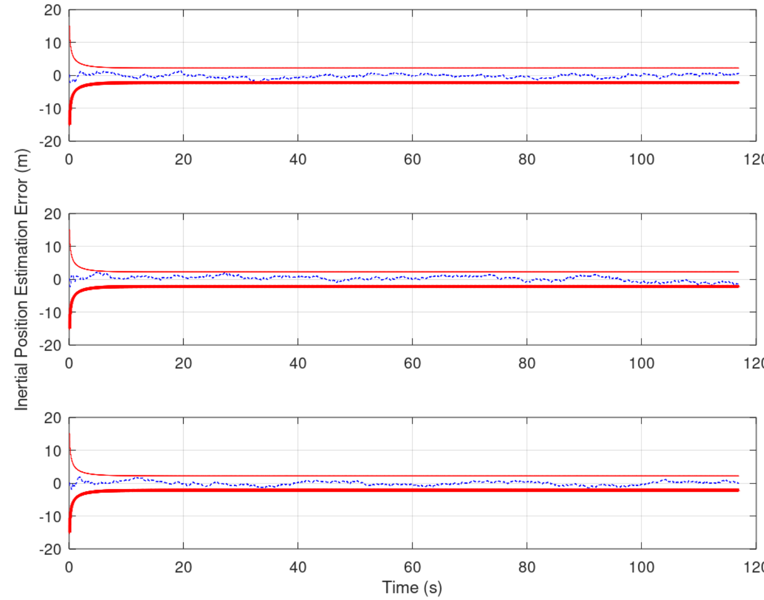
[IMU-GPS]:UAV Position Estimation



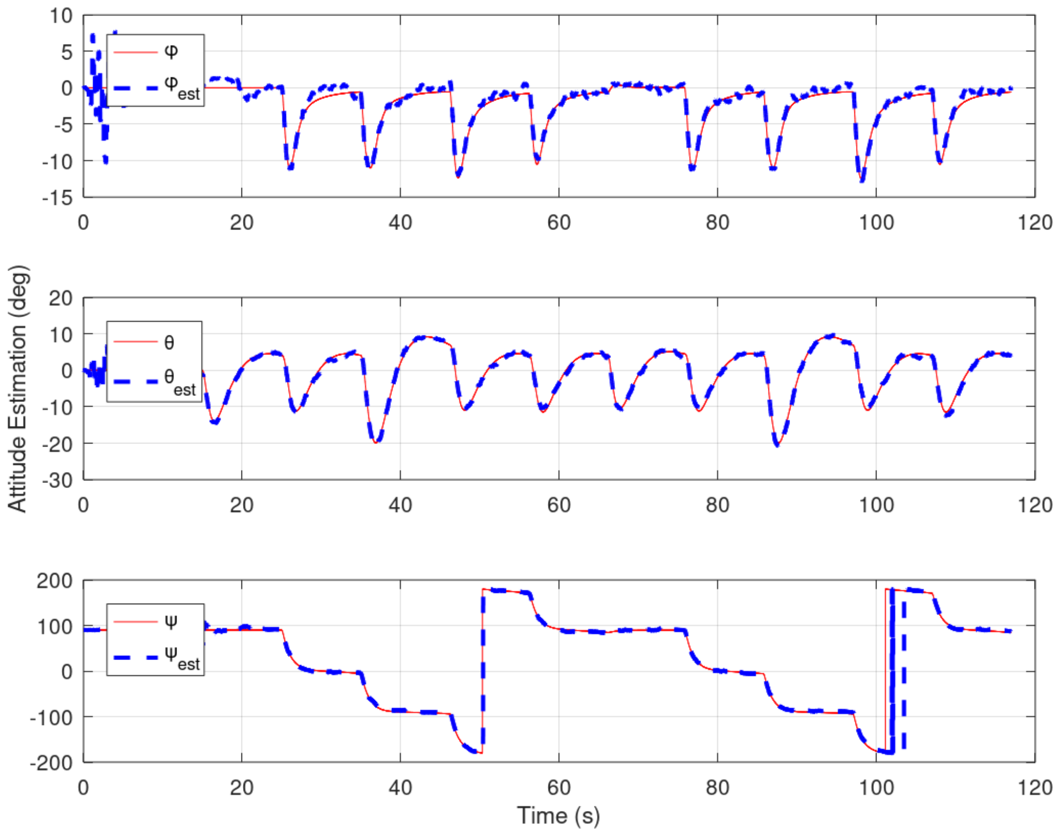


[IMU-GPS]: UAV Velocity Estimation

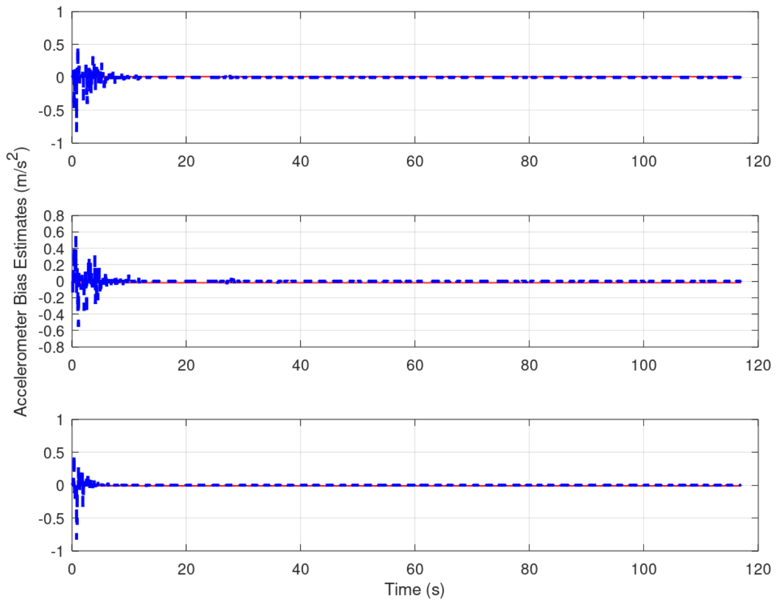


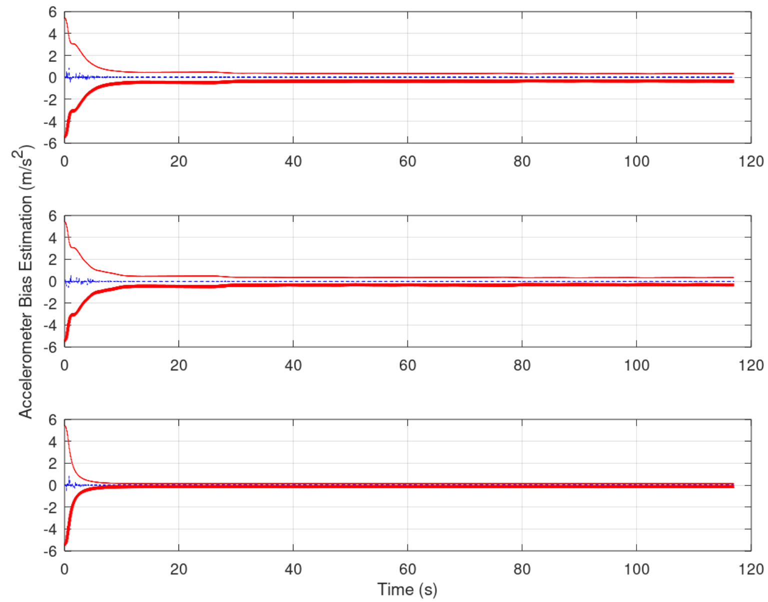


[IMU-GPS]: UAV Attitude Estimation

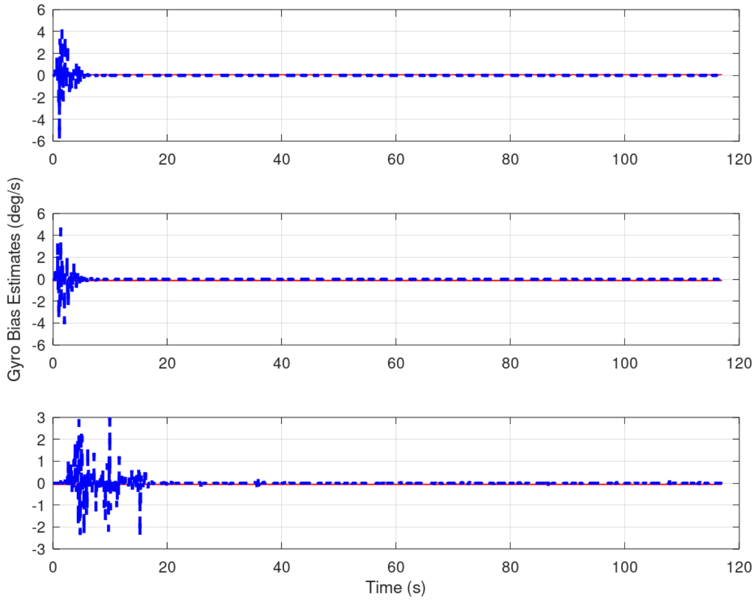


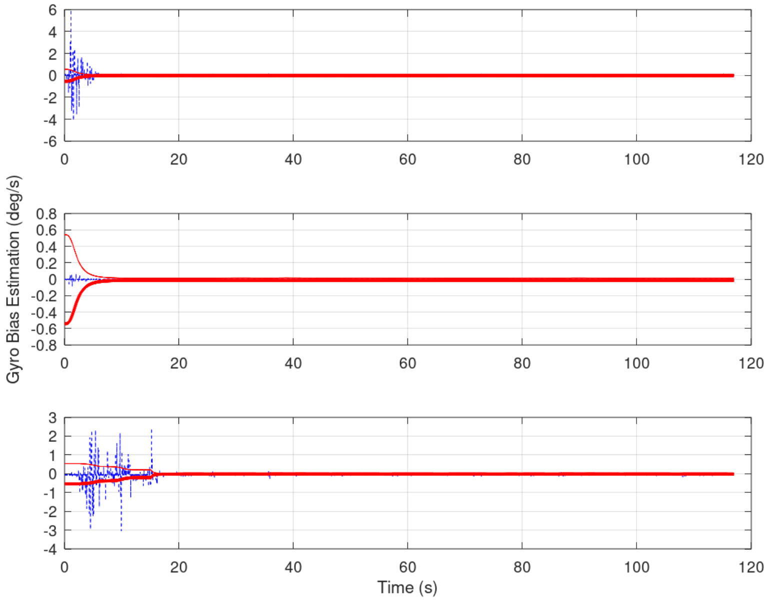
[IMU-GPS]: Acceleration Bias Estimation





[IMU-GPS]: Gyro Bias Estimation



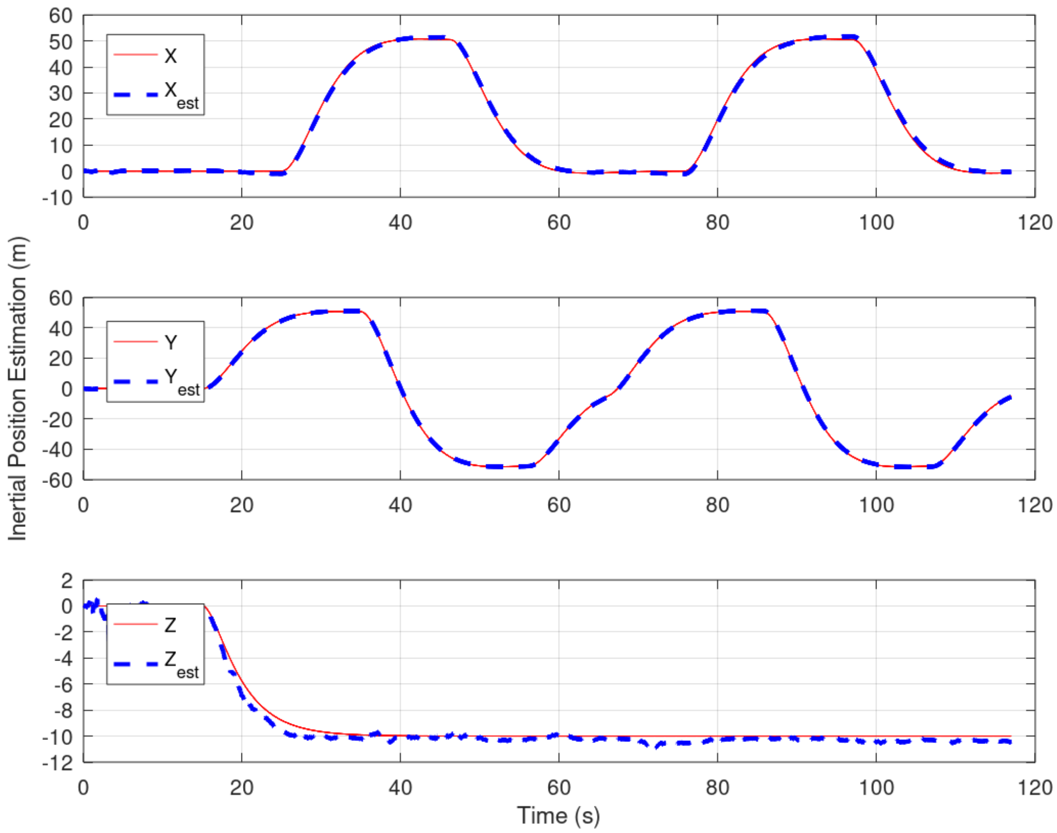


SLAM

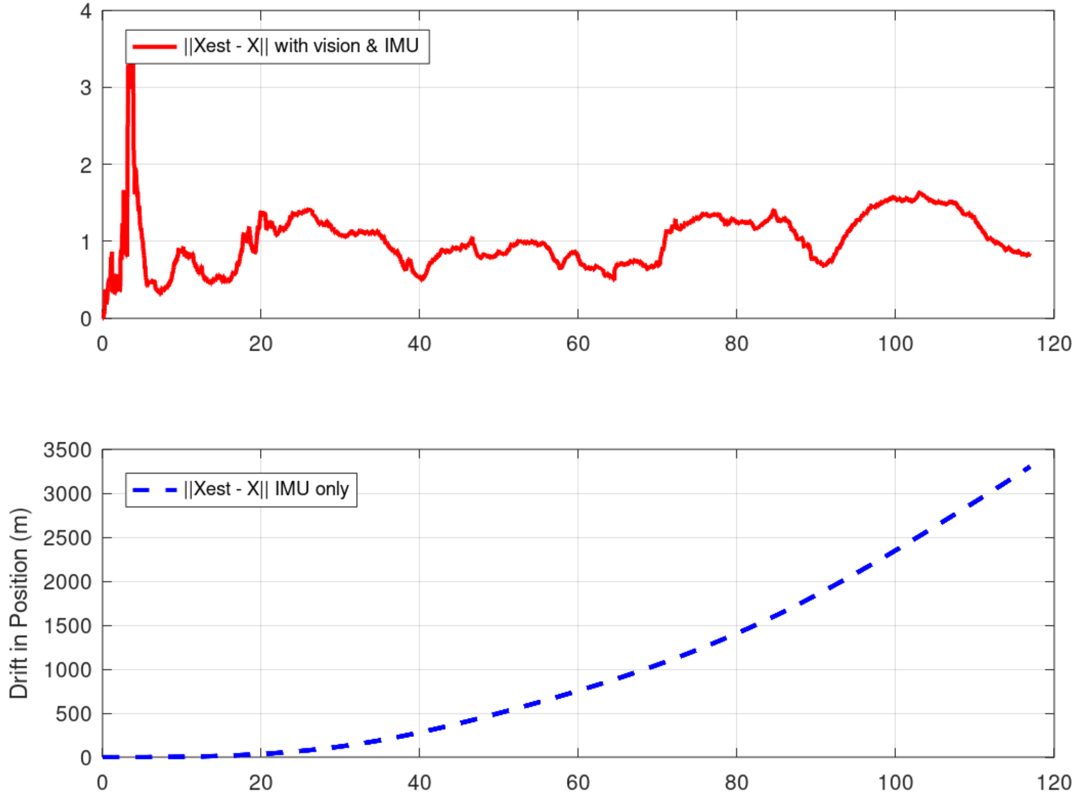
- SLAM: Simultaneous localization and mapping.
- SLAM can be used to GPS denied cases.
- SLAM filter can seamlessly incorporate aiding sensors such as IMU, camera, altimeter, magnetometer, airspeed sensors, etc.
- The same process model is used.
- Measurement model with vision-range sensor:

$$\mathbf{z}_i = \begin{bmatrix} r_i \\ \beta_i \\ \alpha_i \end{bmatrix} = \begin{bmatrix} (x_c^2 + y_c^2 + z_c^2)^{1/2} \\ f \cdot \frac{y_c}{x_c} \\ f \cdot \frac{z_c}{x_c} \end{bmatrix} = h(\mathbf{X}_{rel}^{cam})$$

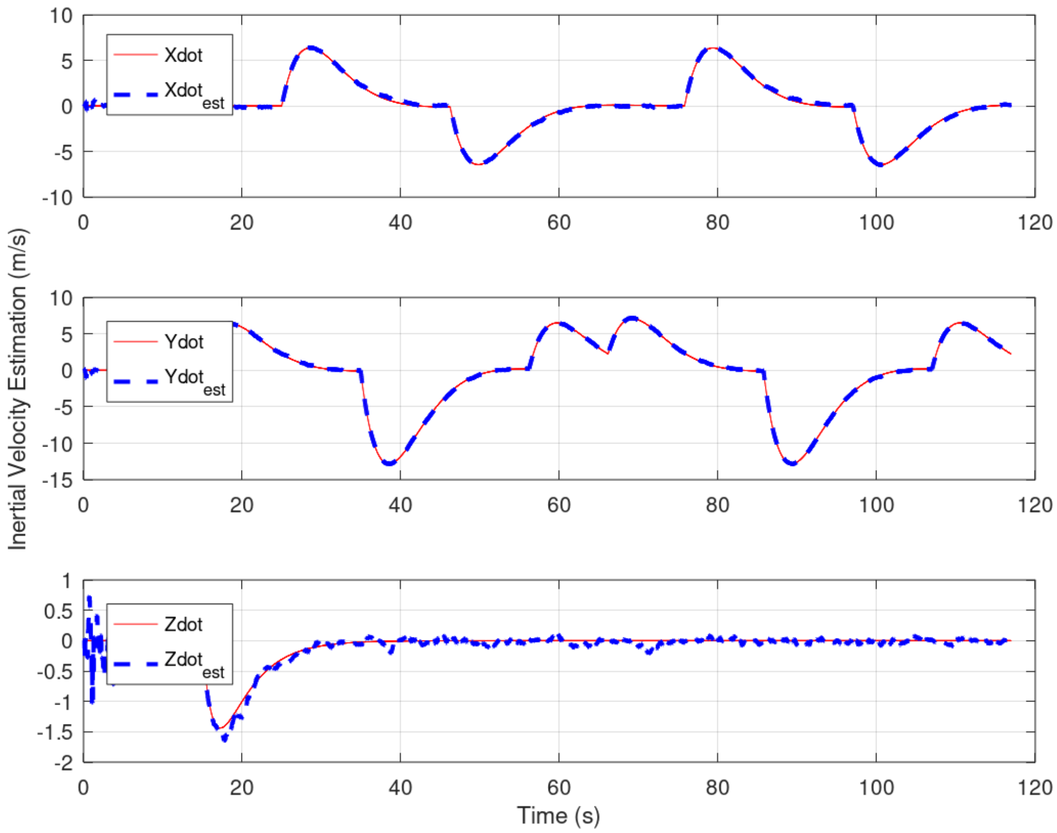
[IMU-CAM-LiDAR]: Position Estimation



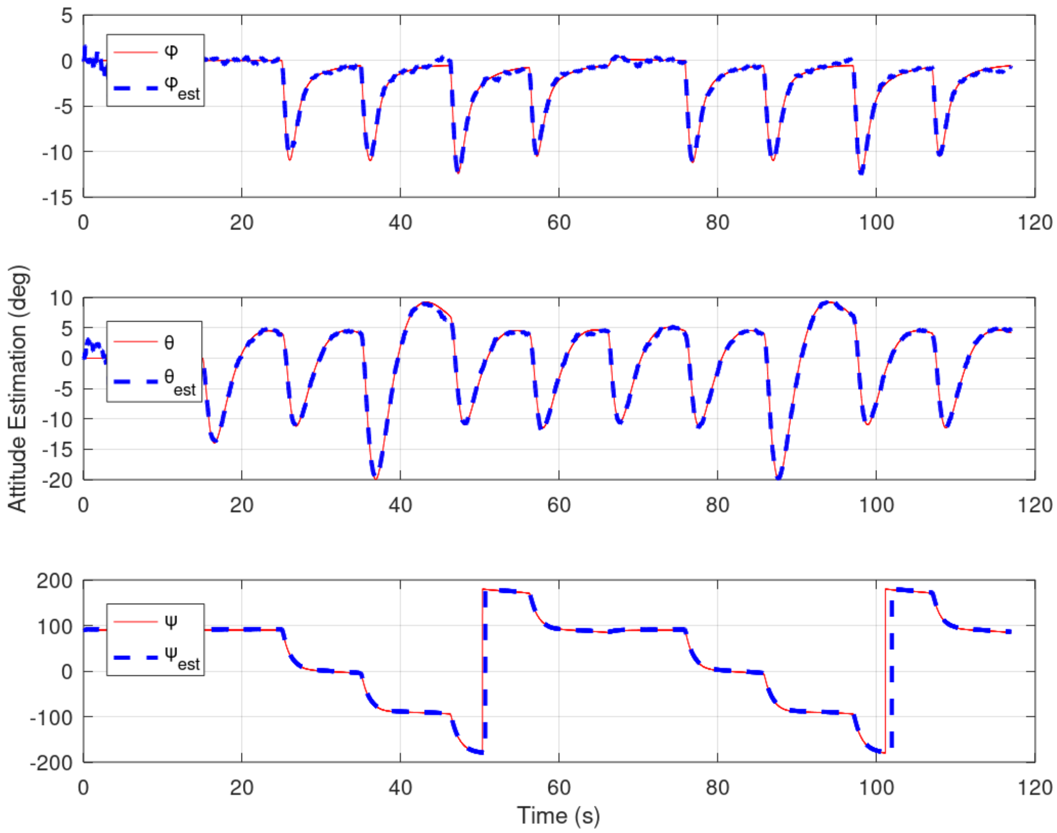
[IMU-CAM-LiDAR]: Position Estimation RMS Error



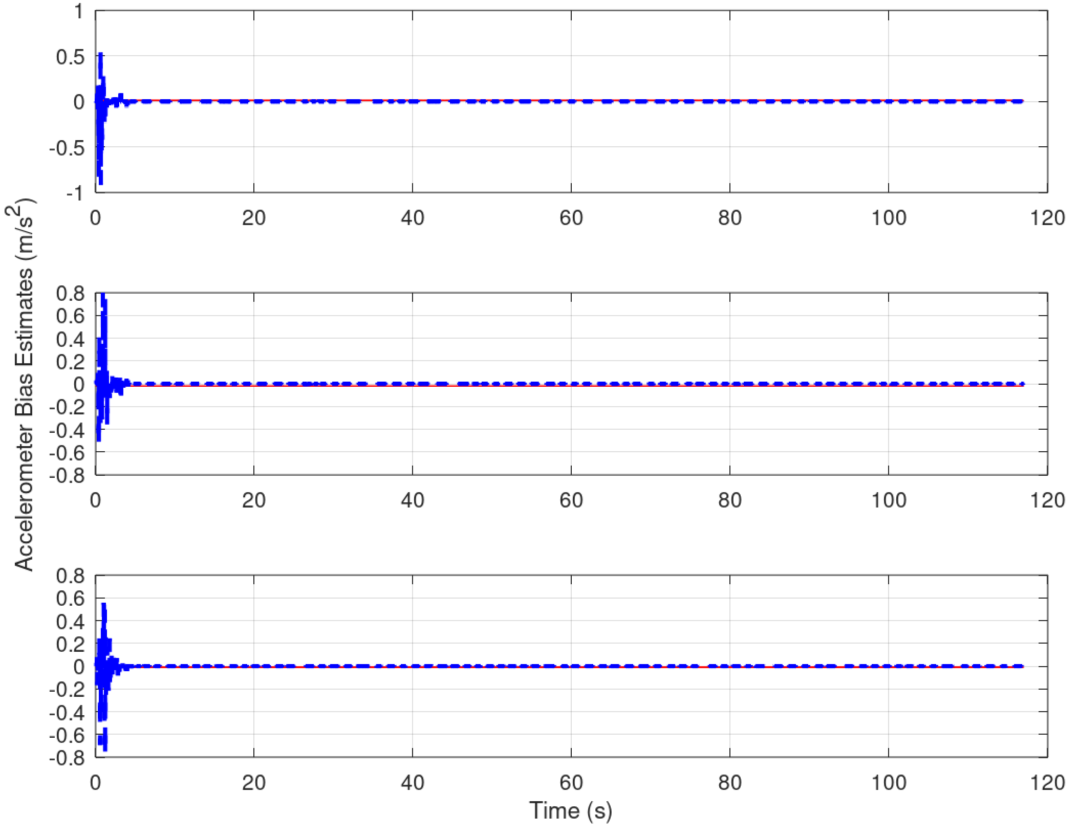
[IMU-CAM-LiDAR]: Velocity Estimation



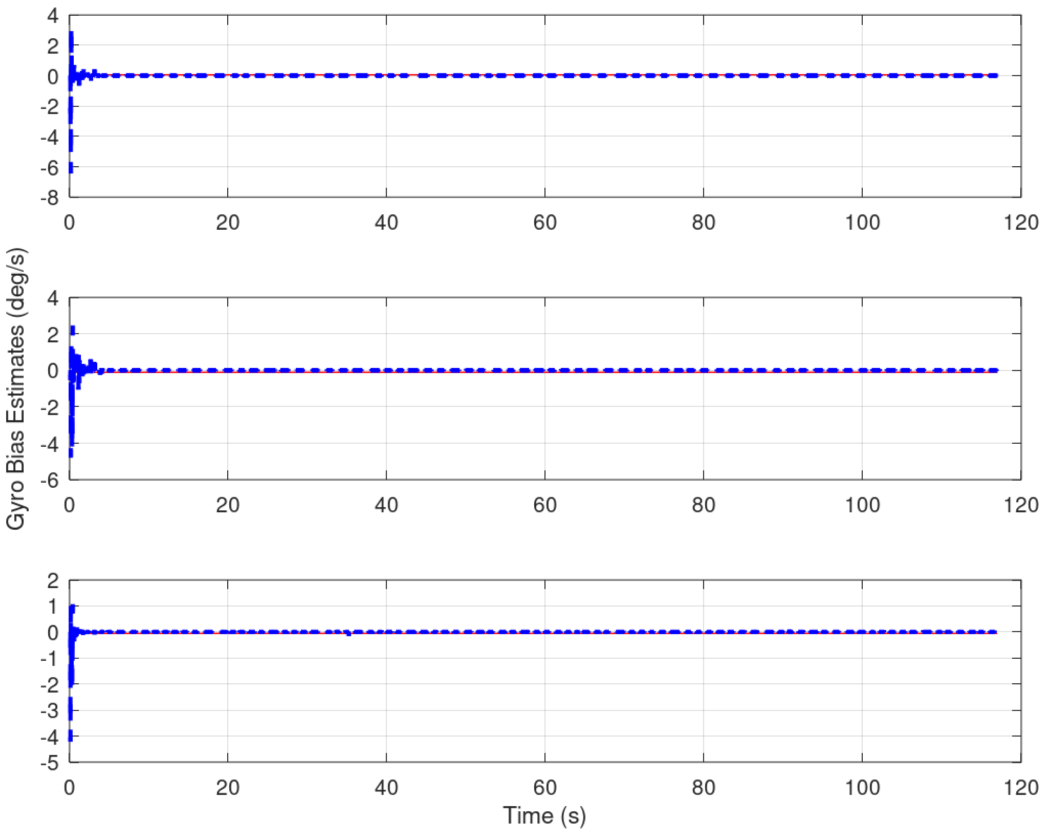
[IMU-CAM-LiDAR]: Attitude Estimation



[IMU-CAM-LiDAR]: IMU Accel. Bias Estimation



[IMU-CAM-LiDAR]: IMU Gyro Bias Estimation



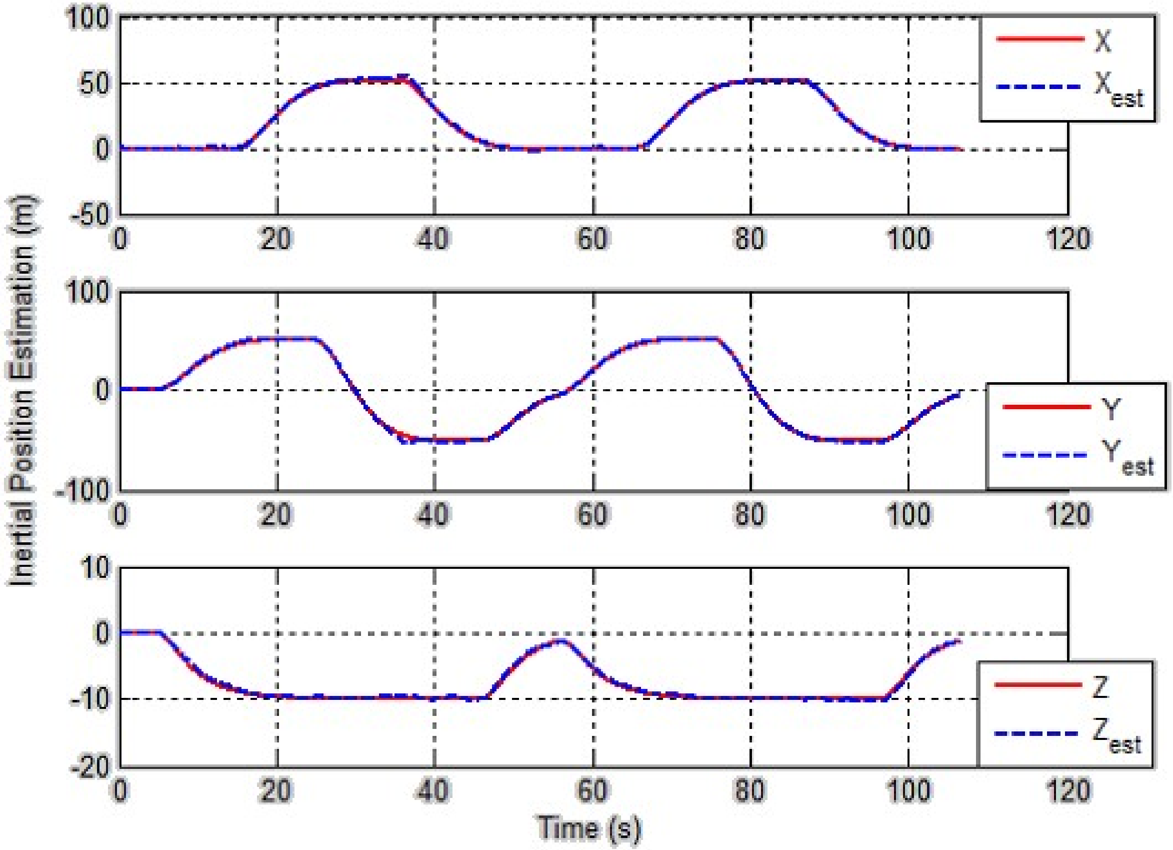
[IMU-CAM]: IDP Based Monocular SLAM

- Without a range sensor, the initialization of new features in the map is delayed until there is enough parallax to estimate the depth of the features.
- Inverse Depth Parametrization (IDP) approach of Montiel, et. al. (2006)
  - Overcomes the initialization problem of monocular SLAM using a modified representation of feature point-Filtering can begin as soon as feature point is detected

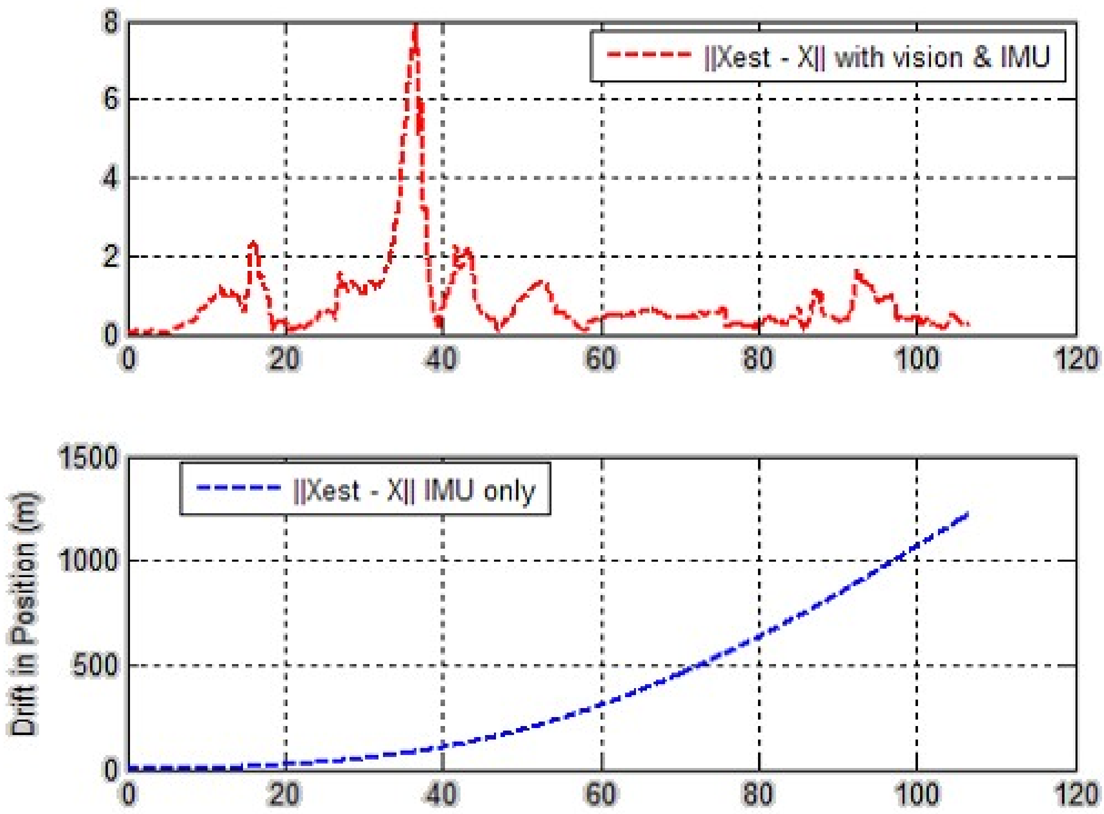
$$\left[x_f, y_f, z_f\right] \rightarrow \left[x_{ci}, y_{ci}, z_{ci}, \frac{1}{r_i}, \beta_i, \gamma_i\right]$$

  - IDP paramterization creates additional computational cost
  - Inertially-aided monocular SLAM reduces the scale bias.

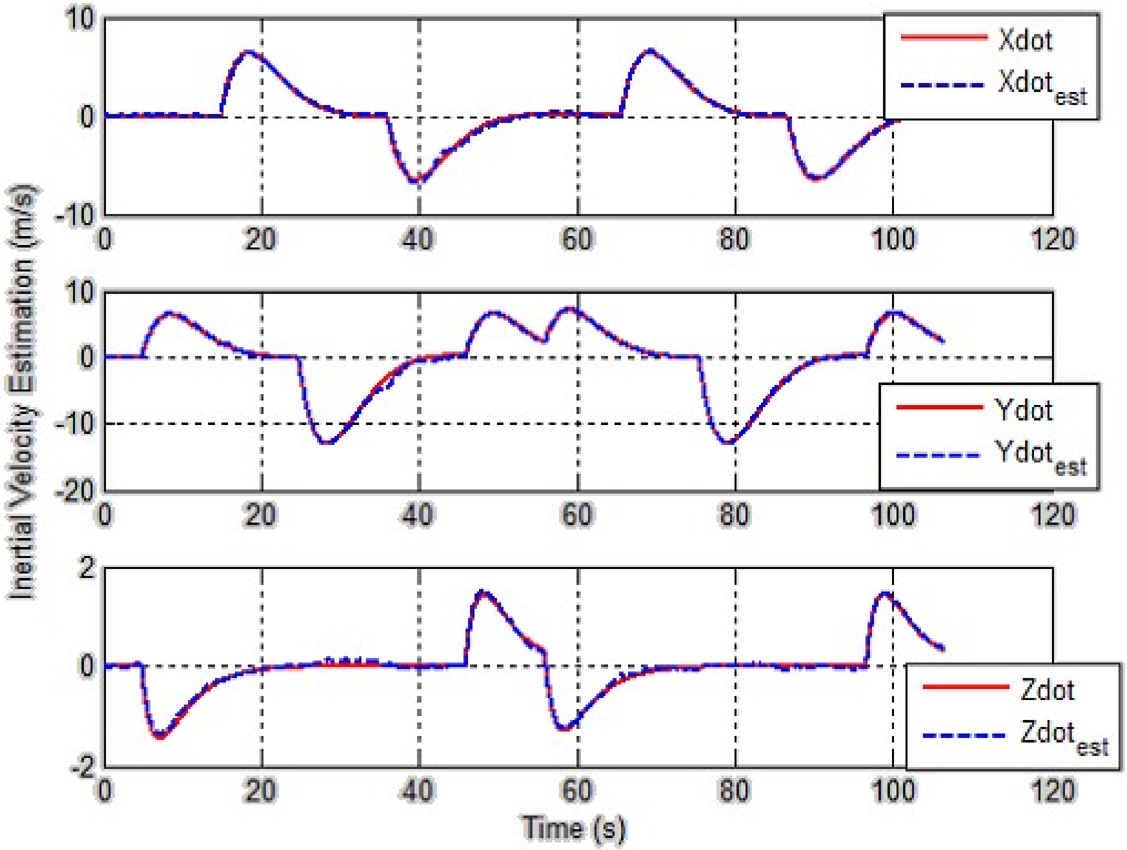
[IMU-CAM]: Position Estimation



[IMU-CAM]:Position Estimation RMS Error



[IMU-CAM]: Velocity Estimation



[IMU-CAM]: Attitude Estimation