AUV Localization Using Dead Reckoning Techniques with IMU Sensor

EEL5934 Aerial & Marine Robotics – Course Project

Authors:
Brandon Davis & Ryan Gochar

Overview

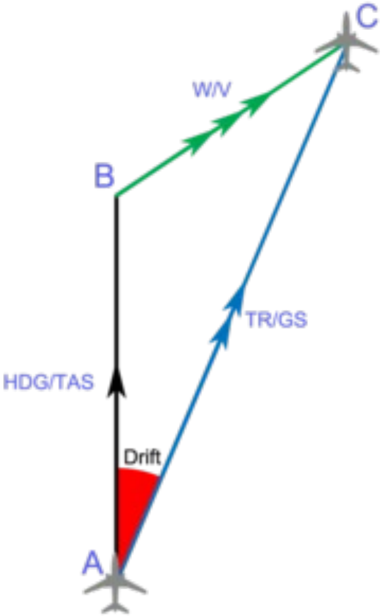
- The purpose of this project is to implement and evaluate a dead reckoning localization algorithm using an inertial measurement unit (IMU) sensor on an autonomous vehicle. The implementation will be applicable for an autonomous underwater vehicle (AUV).

Background

- Most robust and widely used localization system is Global Navigation Satellite System (GNSS).
- Certain environments deny the use of GNSS: underground, indoors, underwater.
- The need for autonomous vehicles to operate in these GNSS denied environments produce the need for alternative localization methods.

Dead Reckoning Overview

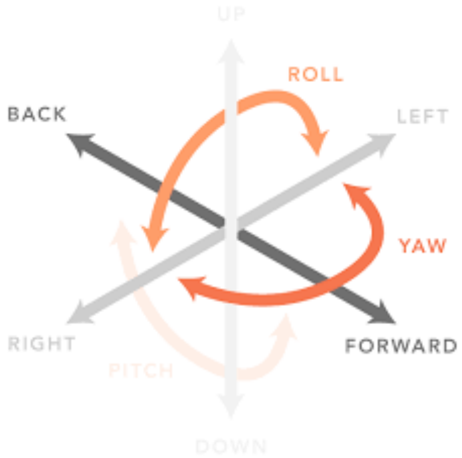
- Method of determining current position based on previous positions and estimates of speed, heading, and elapsed time.

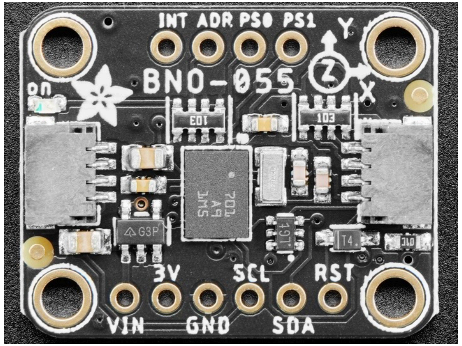


Source: https://en.wikipedia.org/wiki/Dead_reckoning

Internal Measurement Unit (IMU)

- For this project, the Bosch BNO055 9 degrees-of-freedom (DOF) MEMS IMU will be used. This IMU is a combination of these sensors:
  - 3-axis Accelerometer – measures linear velocity along X,Y,Z axis.
  - 3-axis Gyroscope – measures angular velocity around X,Y,Z axis.
  - 3-axis Magnetometer – measures magnetic field along X,Y,Z axis.



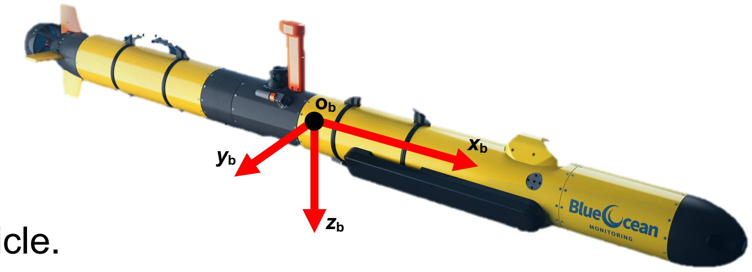


MEMS – Microelectromechanical

Objectives

- Implement a localization scheme using only data from an IMU and measure performance compared to ground truth data (absolute position).
- Implement a Kalman filter based localization scheme using high-rate attitude sensors and low-rate aiding sensors[2].
- Low rate aiding sensors provide a ground truth data measurement for error correction of system. These sensors can be:
    - GNSS (For surface vessels)
    - DVL (Velocity for AUV)
    - Acoustic localization (AUV)
- Verify performance of Kalman Filter dead reckoning using simulated sensor data.
- Verify dead reckoning algorithm is setup to run real-time.

Methodology



- Consider these reference frames:
  - Body frame – origin at CO $\{o_b\}$ of the vehicle.
  - IMU frame – origin of IMU measurement frame.
  - GNSS frame – origin of GNSS sensor.
  - NED - *North-East-Down:*
    - x_n points toward true North.
    - y_n points towards East.
    - z_n points downward normal to Earth's surface.

- Assumptions:
  - For simulated data, we assume for a *strapdown system* that the IMU and GPS frame are aligned with the BODY frame and share an origin. Therefore, all IMU measurements are in the BODY frame.

Methodology (cont.)

- Prior to using the IMU data for any calculations, the static IMU data needs to be evaluated and filtered for noise[1,2].
  - Filtering of IMU data is commonly done using a Kalman filter[1,2]. For this project either a Kalman filter is used. Tuning of the filter will be required and differ from sensor to sensor based on quality of data.
- In addition to filtering, any static transforms from the IMU coordinate frame convention to the body frame convention needs to be calculated.

Methodology (cont.)

- From the IMU data, the *attitude vector* $\Theta_b = [\phi, \theta, \psi]^T$ is calculated[2] where ϕ is the roll, θ is the pitch, and ψ is the yaw.
- The roll and pitch angles are needed to project the 3-axis magnetometer data onto the horizontal plane using the *rotation matrix* $R_{y,\theta}R_{x,\phi}$ [2]
- Then the magnetic compass heading can be calculated using the horizontal components[2]:

$$\psi_m = -atan2(h_y, h_x)$$

- The velocity in 3-axis is calculated by multiplying acceleration measurements by the elapsed time from previous measurement and adding to a previously known velocity value[1].

Strapdown Navigation Equations

$$\dot{p}^{n}_{nmI} = v^{n}_{nmI}$$

Position vector

$$\dot{v}^{n}_{nmI} = a^{n}_{nmI}$$

Velocity vector

$$\dot{\theta}_{nb} = T^{n}_{b}(t)\omega^{b}_{nb}$$

Euler angles

Methodology – Kalman Filter

- How it works?
- Efficient which makes it good for processing real time data
- Inputs: Acceleration data, Magnetometer data (transformations)
- Implementation: pykalman KalmanFilter
- Define state matrices –
  - F = Transition matrices
  - H = Observation matrices
  - Q = Transition covariance
  - R = Observation covariance
  - X0 = Initial state mean
  - P0 = Initial state covariance
- Output: Filtered Acceleration Data

Methodology – Open-loop IMU Solution

- Direct Filter –
  - Accelerometer and angular rate measurements are *inputs* to *strapdown navigation equations*.
  - Position and velocity are states in the estimator.
- Double integration of Acceleration
  - Velocity
  - Position
- Rotation transformations
- Output: Plot positions against true GPS position

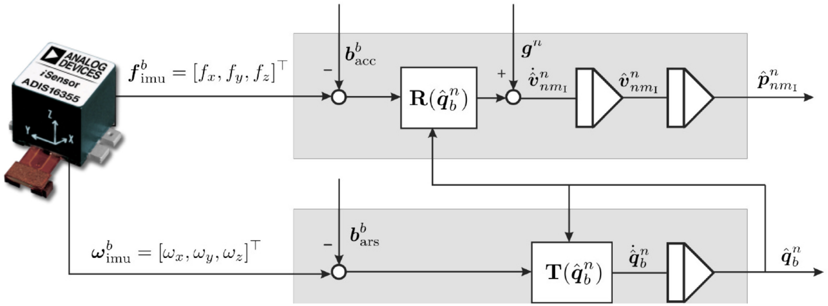


Figure XX. Principle integration of IMU data.[1]

$$\mathbf{p}[\mathbf{k}] = \mathbf{p}[\mathbf{k}-\mathbf{1}] + \mathbf{v} * dt$$

$$\mathbf{v}[\mathbf{k}] = \mathbf{v}[\mathbf{k}-\mathbf{1}] + \mathbf{a} * dt$$

Methodology (cont.)

- We want to compare how introducing an additional sensor measurement will affect the localization solution.
- The process is repeated but instead of using the calculated velocity from IMU data, measured velocity data is used.
- For the purposes of this project, the measured velocity data will come from the GPS data. In a real GNSS denied scenario an alternative sensor type might be used to measure velocity and will depend on the environment/application.

Methodology – Indirect Feedback Kalman Filter

- Indirect Filter –
  - Accelerometer and angular rate measurements are *inputs* to *strapdown navigation equations*.
  - Error estimates are used to update the INS estimates directly to avoid drift.[1]

True state: $\chi[k] = \chi_{\mathbf{ins}}[k] + \delta\chi[k]$

Error state: $\delta\chi[k]$

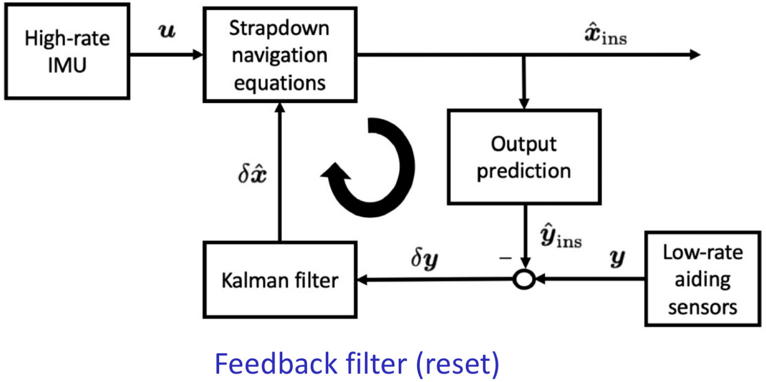
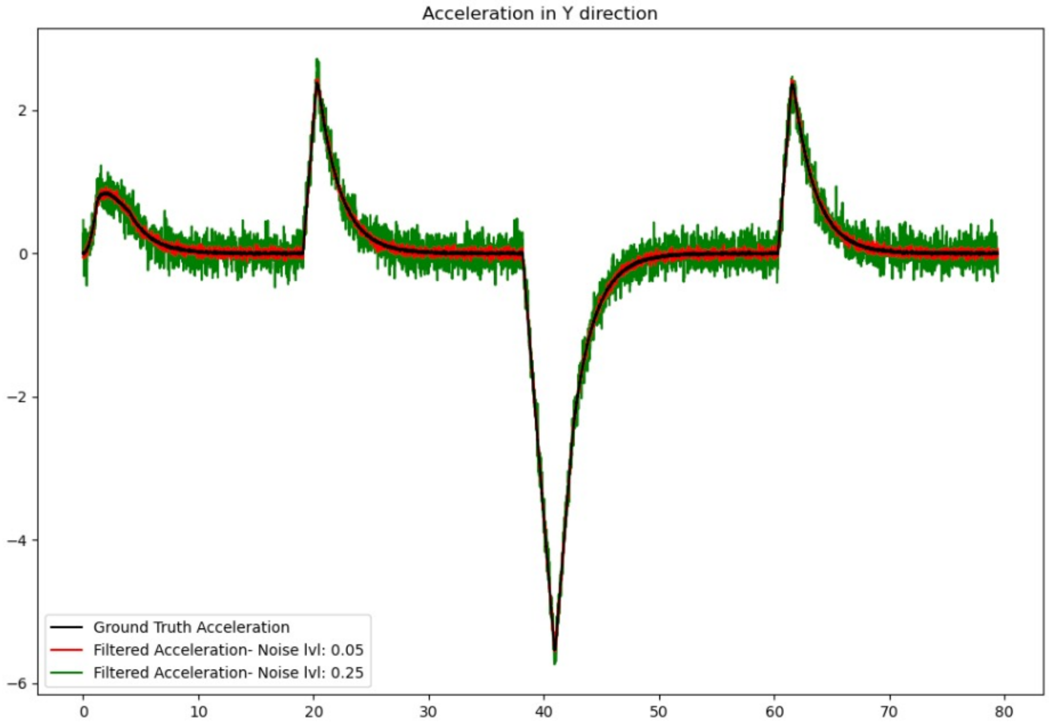
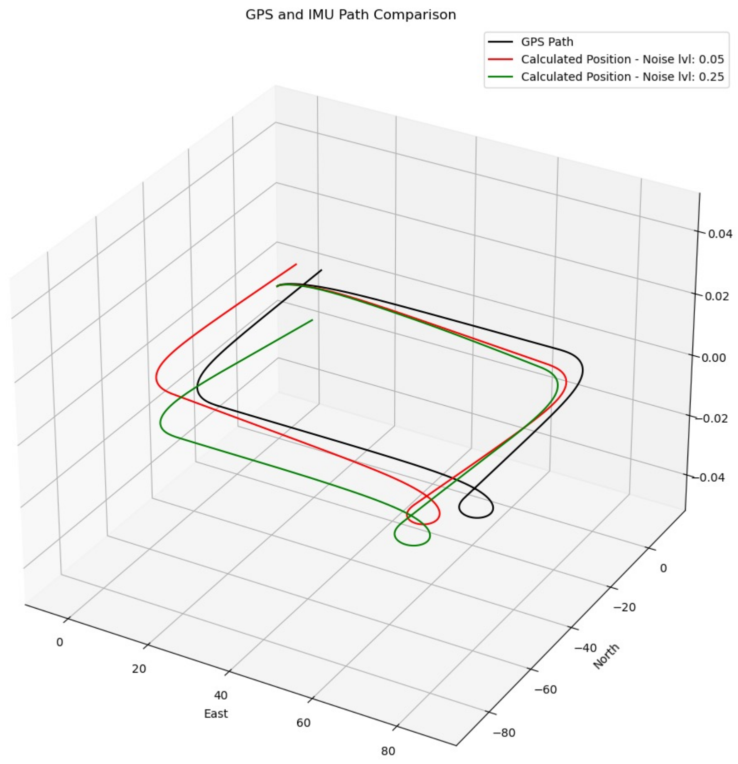


Figure 2. Indirect feedback filter for INS.[1]

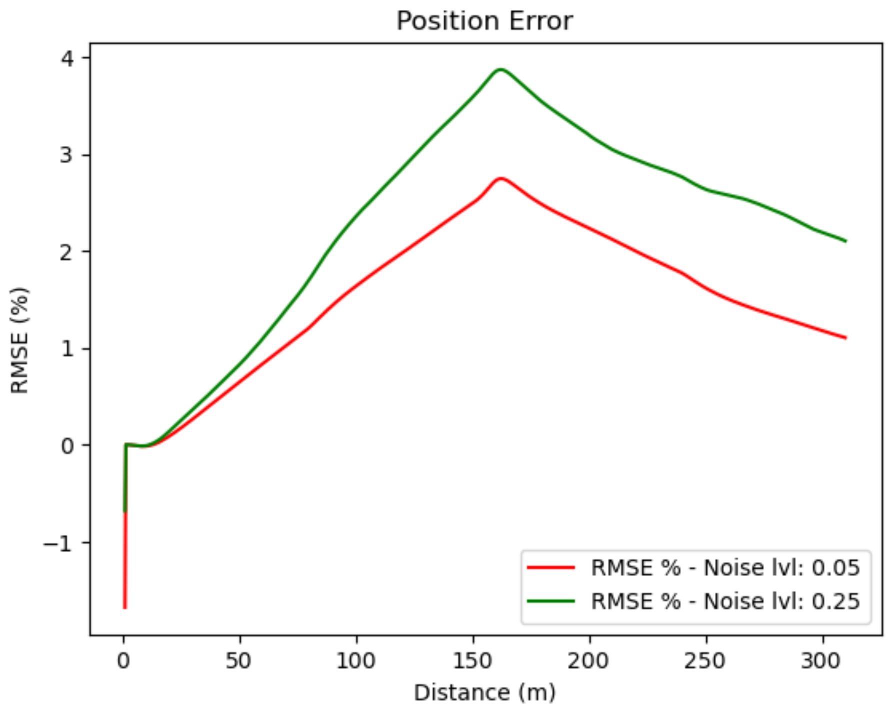
Results – Filtering Acceleration



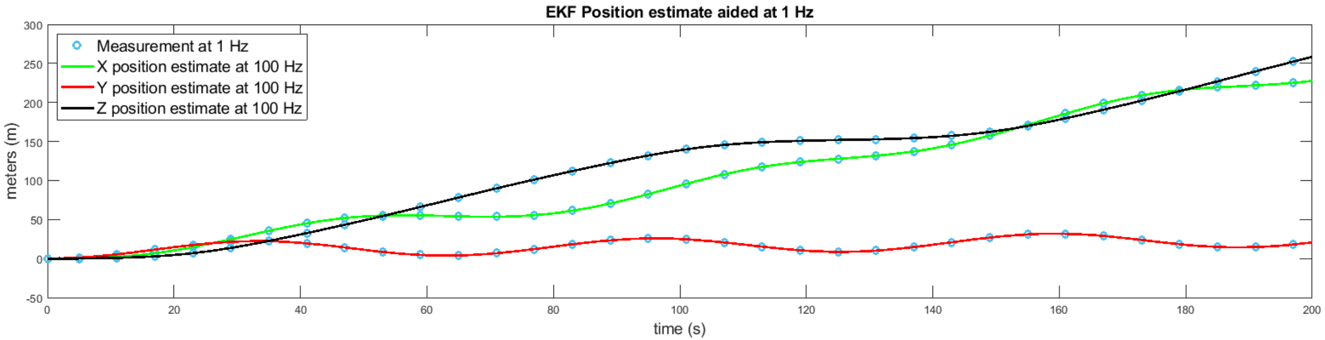
Results – Open-loop IMU Solution

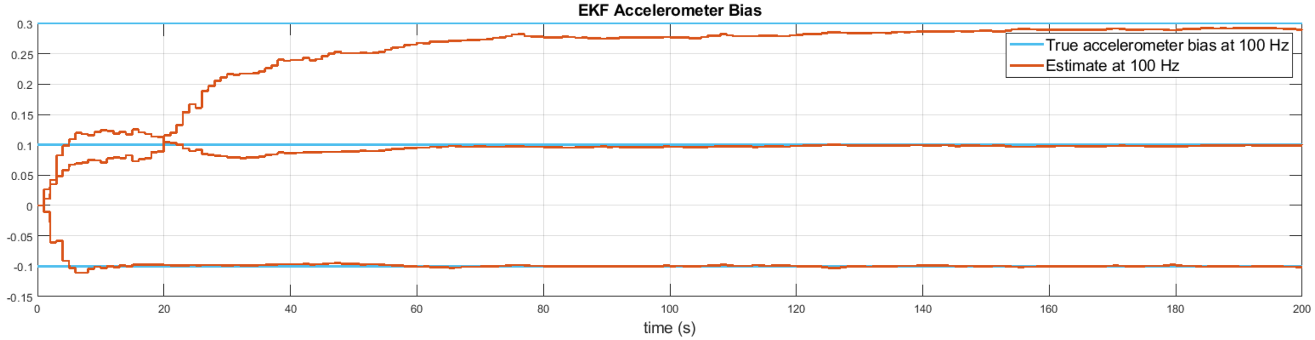


Results – Open-loop IMU Solution

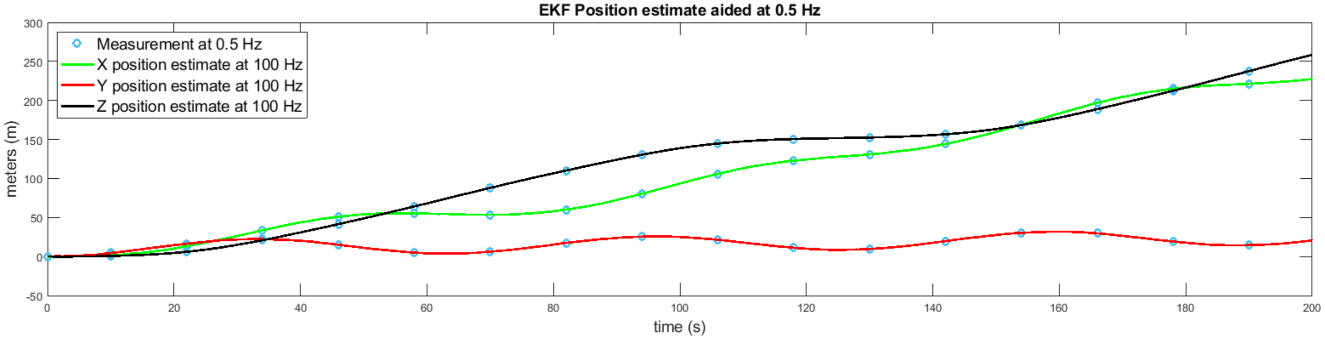


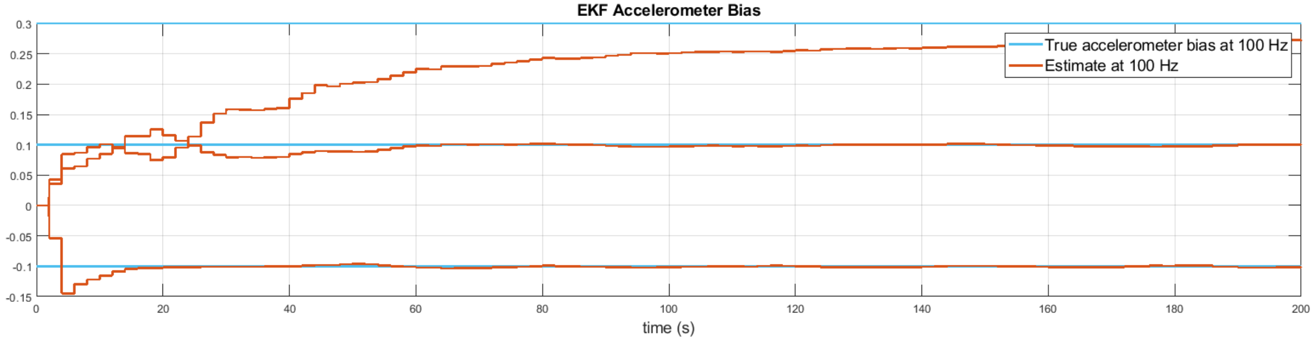
Results – Indirect Feedback Kalman Filter



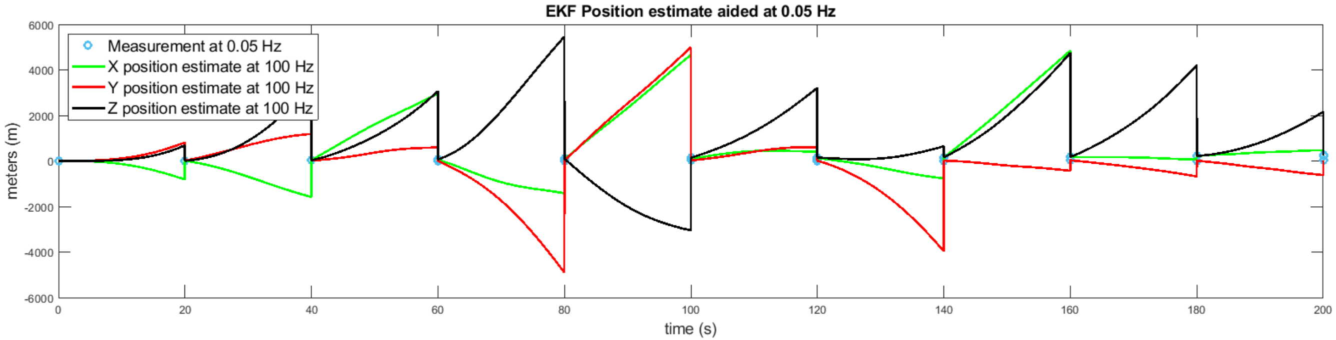


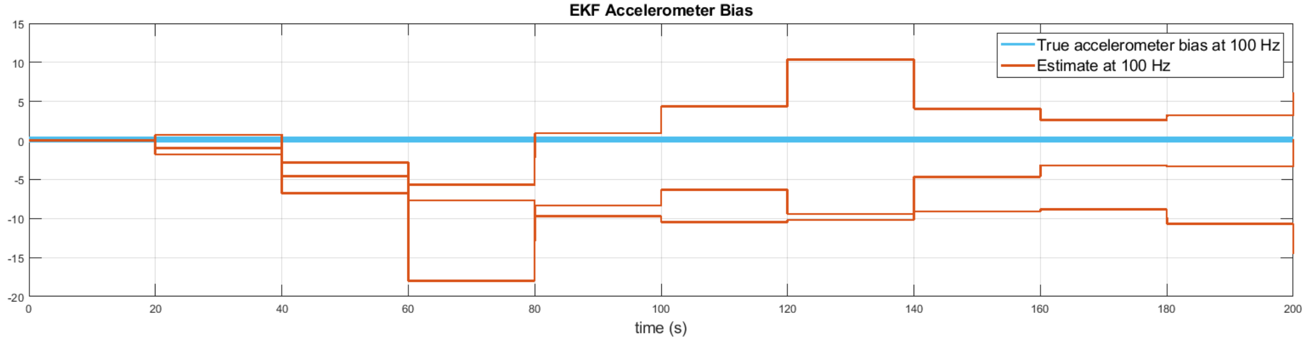
Results – Indirect Feedback Kalman Filter



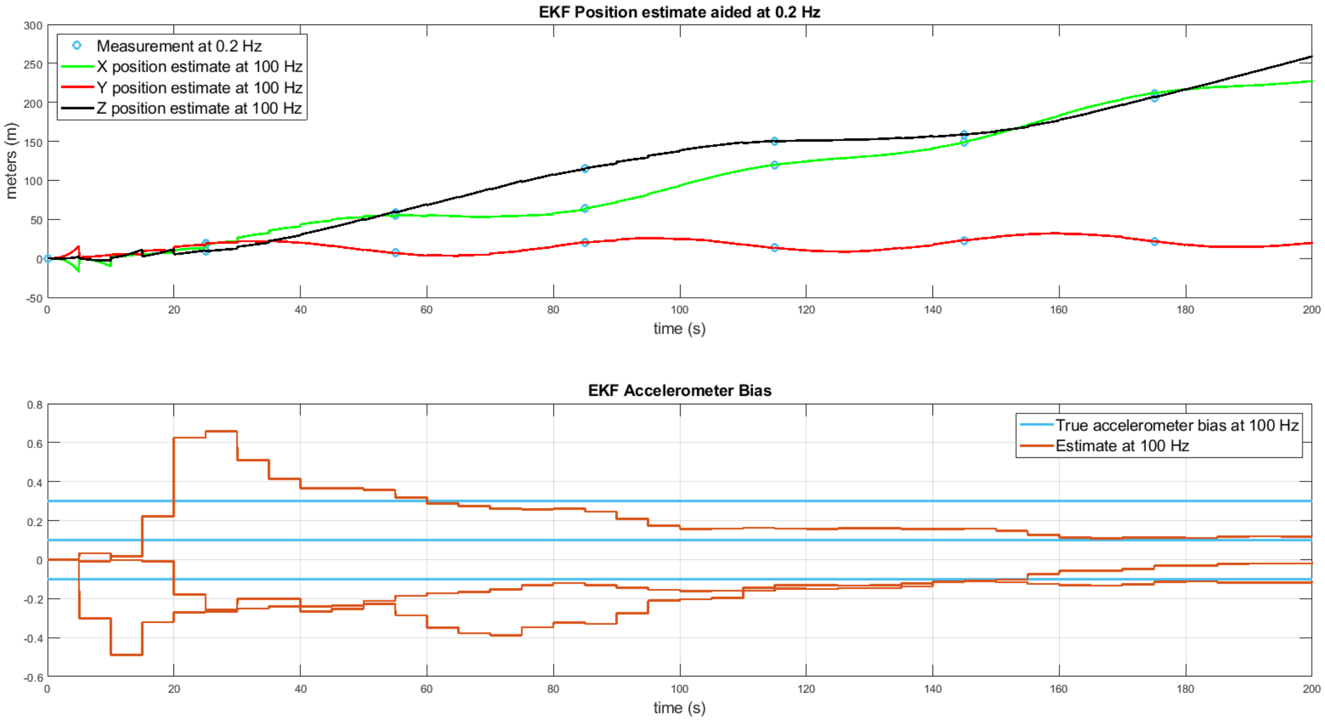


Results – Indirect Feedback Kalman Filter

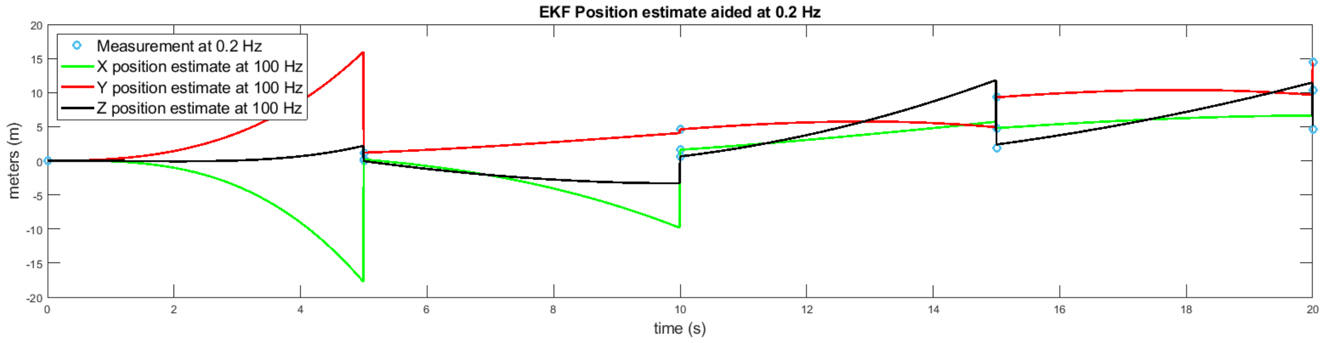


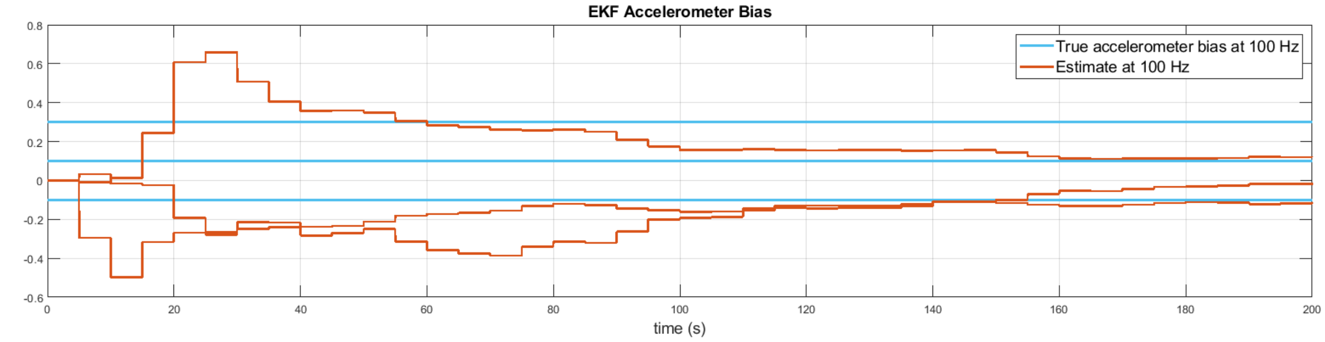


Results – Indirect Feedback Kalman Filter



Results – Indirect Feedback Kalman Filter





References

[1] Toy, A. Durdu and A. Yusefi, "Improved Dead Reckoning Localization using IMU Sensor," 2022 International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 2022, pp. 1-5, doi: 10.1109/ISETC56213.2022.10010239.

[2] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control,* 2^{nd} ed. Hoboken, NJ, USA: Wiley, 2021.

Questions?