State Estimation of Robot Rotation

Dina Ashraf

Abstract

This report discusses the development of sensor fusion technique for Turtlebot 3 localization in a simulation environment. Fusing the readings of robot odometery and IMU acceleration takes place to determine the exact orientation of the robot. The application of Kalman filtering for error elimination will also be discussed in the following sections.

1. Introduction

State estimation has been established as a problem in the world of autonomy. The basic step to develop an autonomous robot is to start by thinking how it perceives the world and how exactly it will know its precise position in the world. Multiple sensors are used to have the ability to perceive and interact with the surrounding environment, there is a problem here as the sensor's readings have always been known to be subjected to several types of errors caused by physical and environmental disturbances. Kalman filtering is applied to reduce the error propagation process and ensure high accuracy estimated output. If the robot can perceive the world from multiple sensor readings, the question is which sensor readings are more reliable? Here is when sensor fusion method was proposed, where we can fuse all sensors readings to come up with an estimated robot state.

2. Software

As mentioned previously the state estimation module was developed for Turtlebot 3 Burger using the gazebo physics simulator for testing and simulation. The Robot Operating System (ROS) is used to combine all the developed modules and codes with the robot and sensor models. Python programming language was used in all the developed codes and graph plotting.

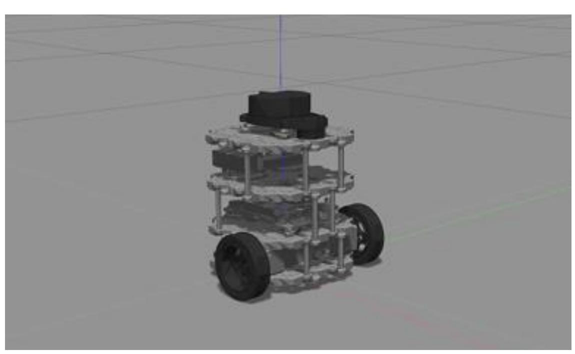

Figure 1: Turtlebot 3

3. State Estimator

In order to estimate the robot's orientation, the odometer sensor readings were fused with the IMU readings using linear Kalman filter. The filter will use the rotational angle readings from the odometer as the predicted angle, as previously mentioned the sensors readings are noisy so the Kalman filter will eliminate the error and combine the readings with IMU measurements for robot's orientation. Odometry readings are streamed on the /odom topic receiving the 2D pose of the robot so we can access its rotational angle values while IMU readings are on /imu topic contains the rotation of the robot and the angular velocity.

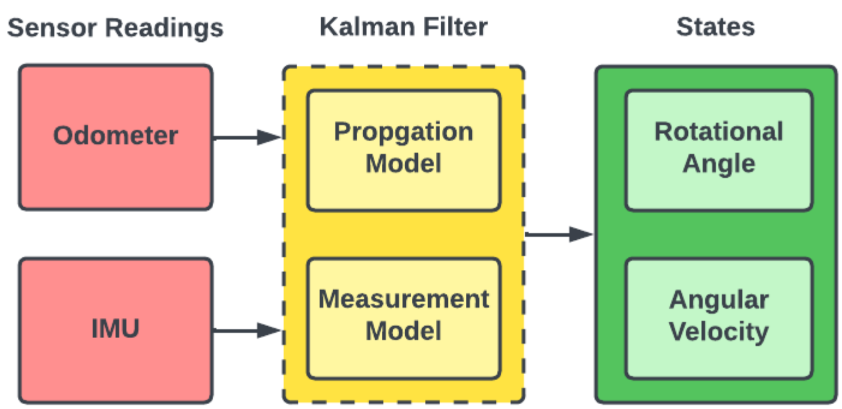

Figure 2: State Estimator Architecture

States and propagation model:

$$\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}^{k+1} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}^{k} + \omega$$

Measurement model:

$$\gamma = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \nu$$

Used variables:	
θ	Rotational angle
$\dot{\theta}$	Angular velocity
γ	Yaw
ω	Process noise
ν	Measurement noise

4. **Results**

The following graph shows the result of the state estimator published on /filtered heading topic plotted against the noisy sensor data published on /noisy_state topic.

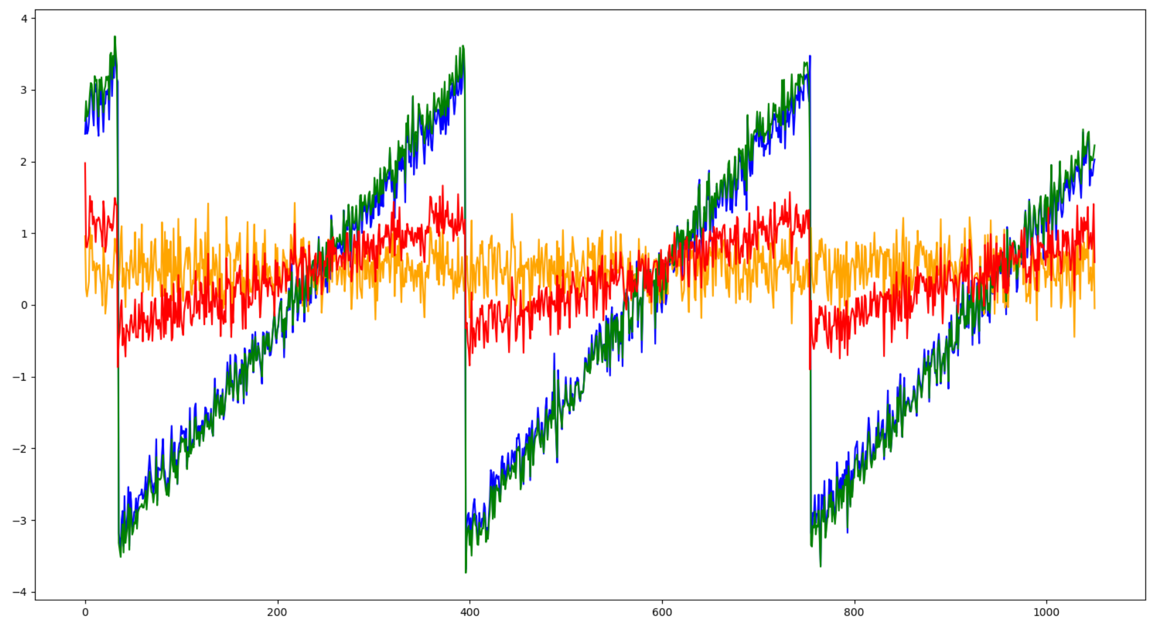

Figure 3: Result Graph

Blue	noisy_state.data[0] (noisy yaw)
Green	filtered_heading.data[0] (filtered yaw)
Orange	noisy_state.data[1] (noisy angular)
Red	filtered_heading.data[1] (filtered angular)

5. **References**

http://wiki.ros.org/Documentation

https://github.com/ROBOTIS-GIT/turtlebot3

https://www.mathworks.com/videos/series/understanding-kalman-filters.html

STATE ESTIMATION FOR ROBOTICS book by Timothy D. Barfoot