Example Unscented Kalman Filter Implementation

Peter Kohler

April 27, 2023

Estimation Scenario:
This project's estimation scenario is an attempt to estimate the two-dimensional position (x and y) and the orientation angle of a differential drive two-wheeled robot. The state dynamics were provided in the project description of the default system. In this simulation the input to the state dynamics functions, the wheel velocities, have an added gaussian noise that is zero mean and has a variance of 0.01. The output "measurement" involves the use of an aruco marker. In the 3D simulator, the turtlebot3 robot is equipped with an intel realsense camera. This simulated camera is defined in the simulation's parameters to have noise that is gaussian (zero mean, 0.007 standard deviation) added to each pixel's color channel values (a value between 0 and 1). The digital images of that camera are being fed into an open source computer vision library (OpenCV) that is detecting the aruco marker and calculating the 3D coordinate transform from the camera's lens to the aruco marker. Figure 1 shows a visual representation of the detected aruco marker using the OpenCV library.

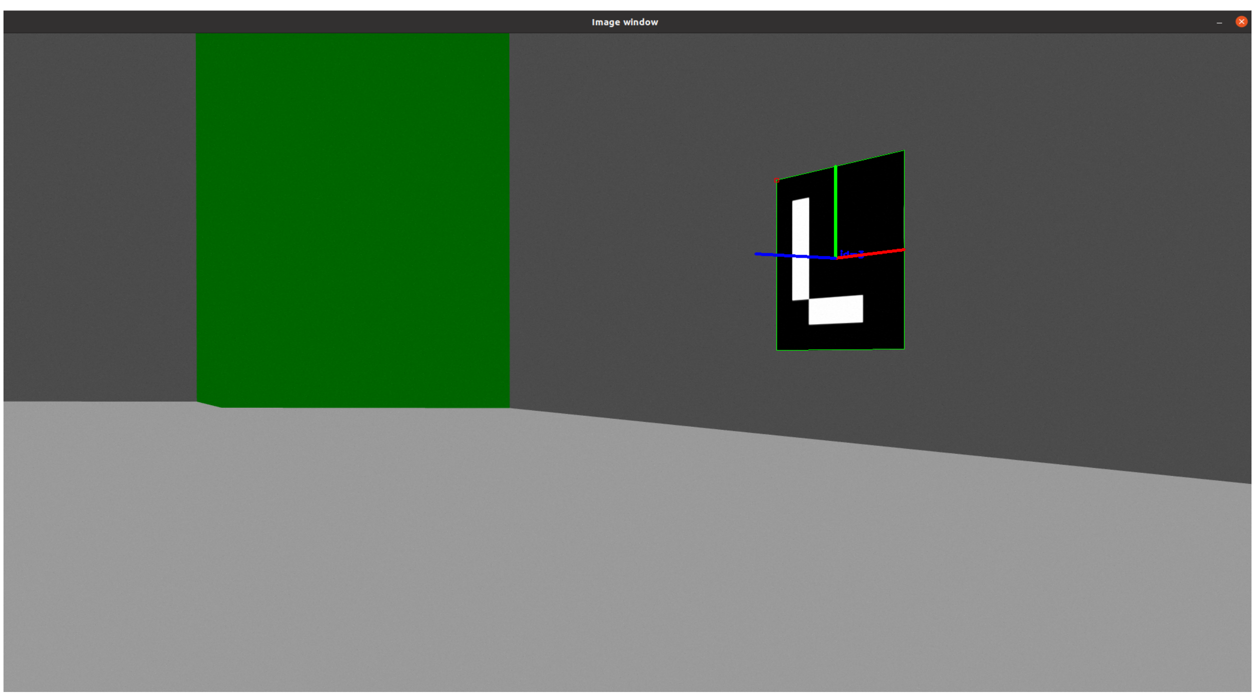

Figure 1: OpenCV Aruco Marker Detection

Figure 2 shows an overhead view of the robot in the simulation space and the white angled lines coming out from the robot indicate the camera's field of view.

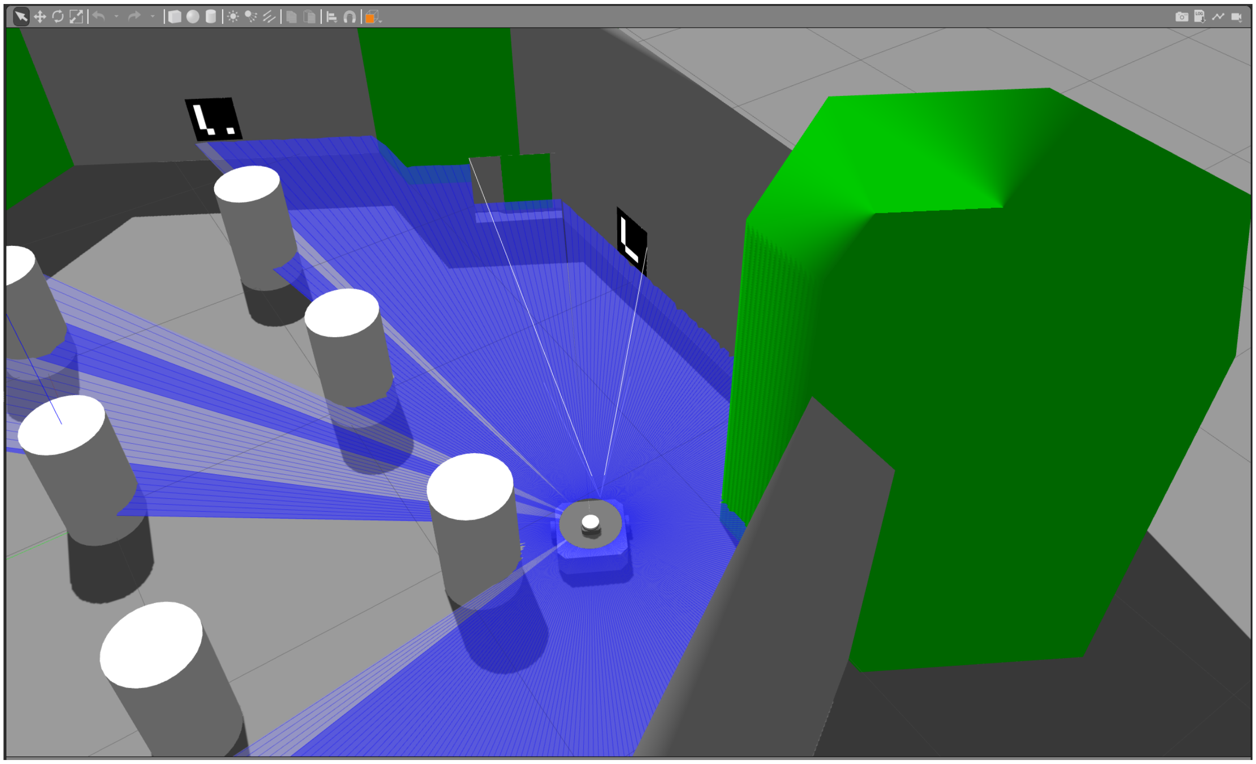

Figure 2: Bird's Eye View of the Simulation

In addition to the 3D coordinate transform, the OpenCV library provides the ID number of the detected aruco marker. Each of the placed aruco markers in the simulation space are unique, there are no duplicates. Since the location of the aruco markers are defined in the simulation world's configuration file, the exact coordinate transform from the world frame origin to the aruco marker is known. Due to the nature of homogenous transforms in that you can chain them together, the 4x4 homogenous transform from the world frame origin to the aruco marker can be multiplied by the homogenous transform of the aruco marker to the camera lens and in turn the transform of lens to the base foot print of the robot. This means that upon detection of an aruco marker in the robot's camera frame it is possible to calculate the location of the robot in the world frame coordinate system. The known X and Y coordinates of the detected aruco marker and the calculated X and Y coordinates of the robot serve as the basis of my observation measurement. My observation is the two argument arc-tangent, commonly referred to as "atan2" using the robot and the aruco marker's coordinates. Equation 1 shows the equation for this observation.

$$\sigma = atan2((Y_R - Y_A), (X_R - X_A)) \tag{1}$$

Where (X_R, Y_R) are the X and Y coordinates of the robot, and (X_A, Y_A) are the X and Y coordinates of the aruco marker. Figure 3 shows a visual representation of the observation.

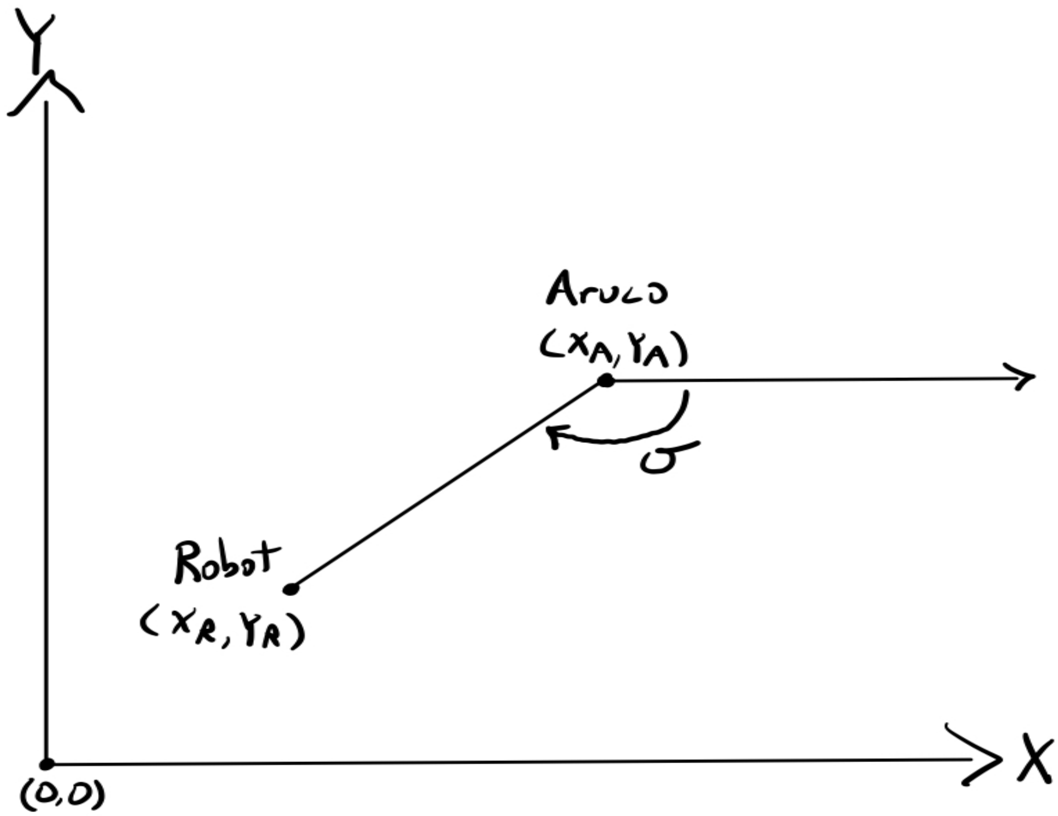

Figure 3: Sigma Observation Depiction

The Results:
The Unscented Kalman Filter was implemented in real time using the Robot Operating System to pull in observed sigma angle calculations alongside the observed location of the aruco marker and robot using the methods described in the previous section. The initial estimate (x_0) x and y values were set to have an error of 10% from the true starting coordinates of the robot. The initial estimate yaw value was set to the correct initial condition because the observation in this implementation does not provide a correction to the yaw angle at all. The output estimated state of both the EKF and UKF were recorded using MATLAB and the results were compared to the ground truth values of the position and pose coming from the simulator. The results of are shown below in figures 4, 5 and 6. The infinite slope line in the yaw angle measurement is due to the fact that the simulator wraps it's yaw angle from $-\pi$ to π. I implemented this in my code to match.

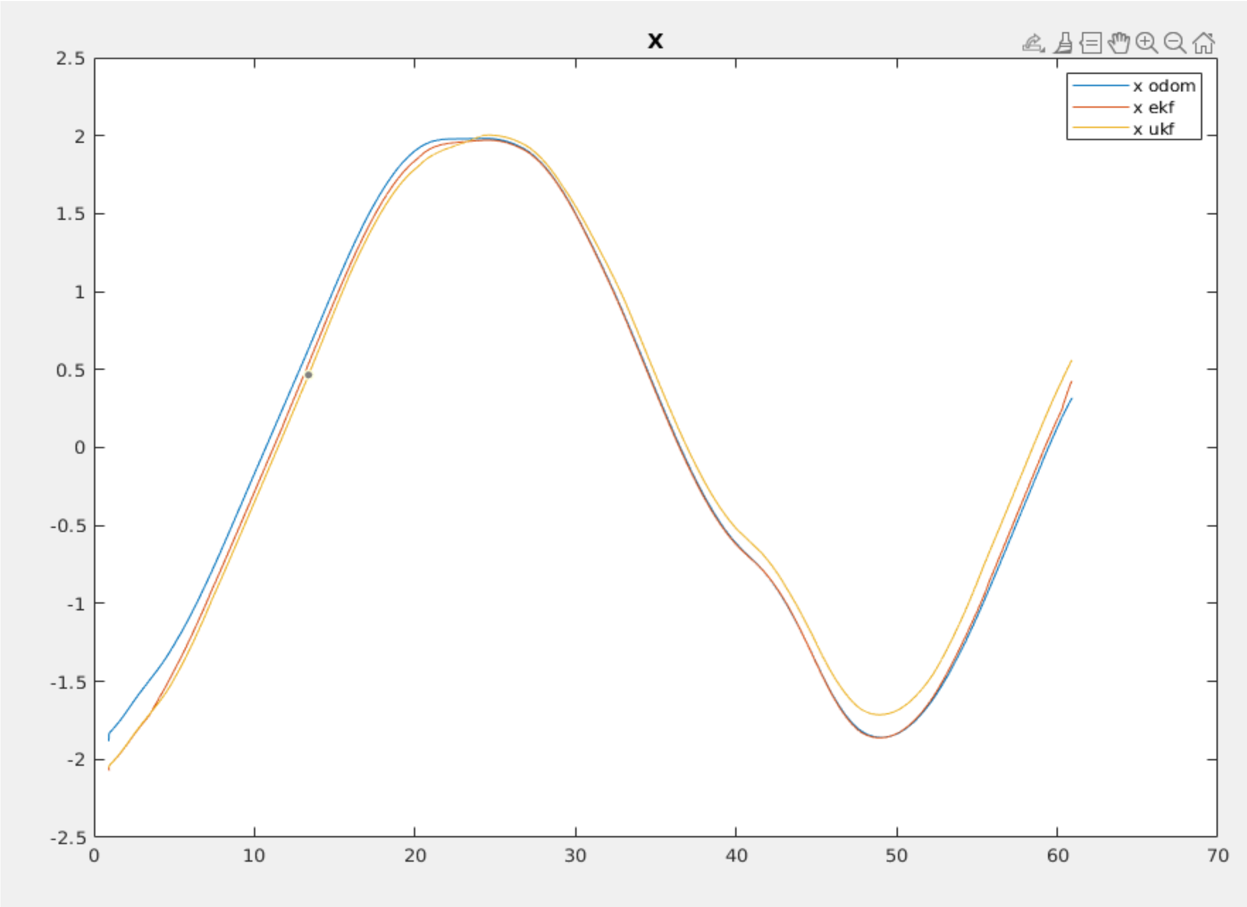

Figure 4: X Coordinate UKF Estimation vs. EKF Estimation vs. Ground Truth

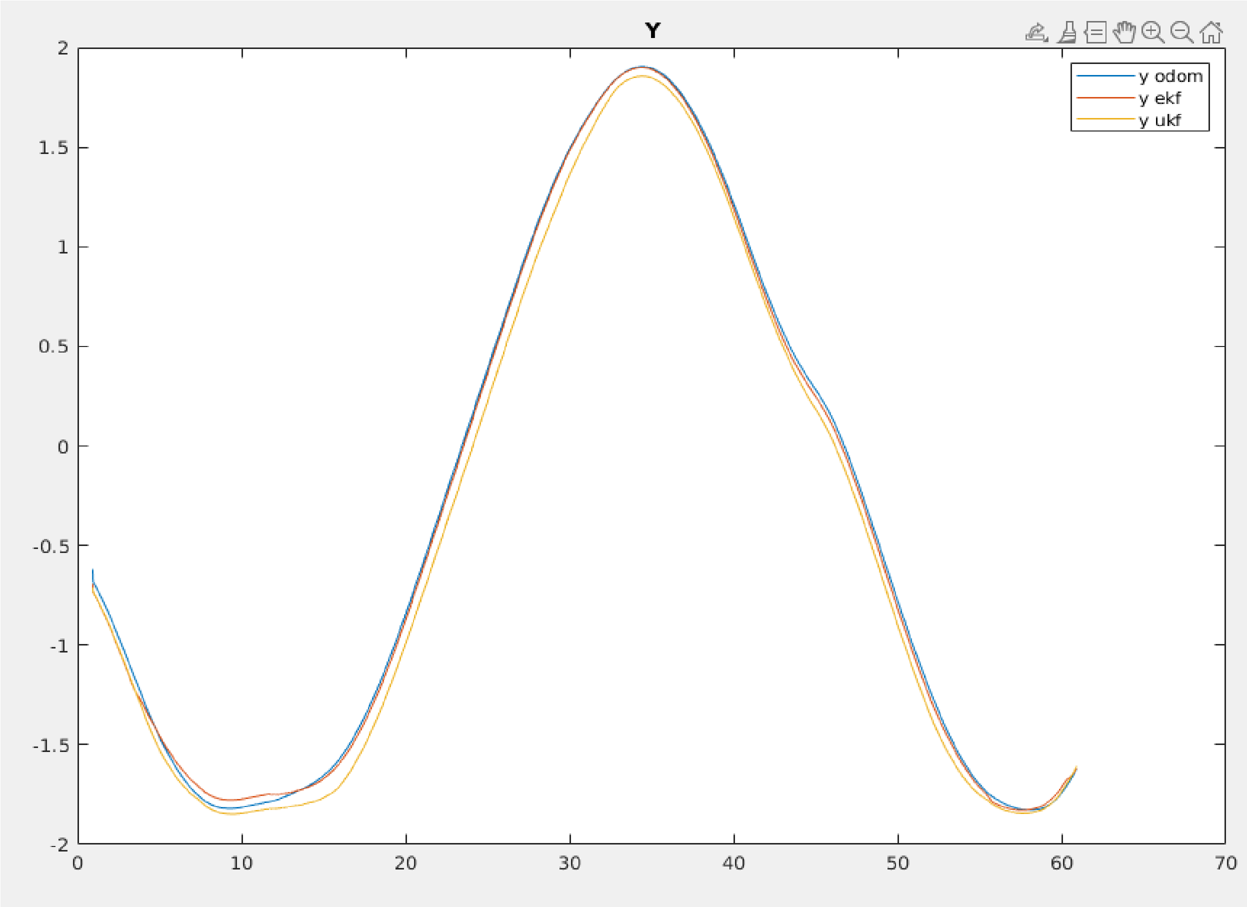

Figure 5: Y Coordinate UKF Estimation vs. EKF Estimation vs. Ground Truth

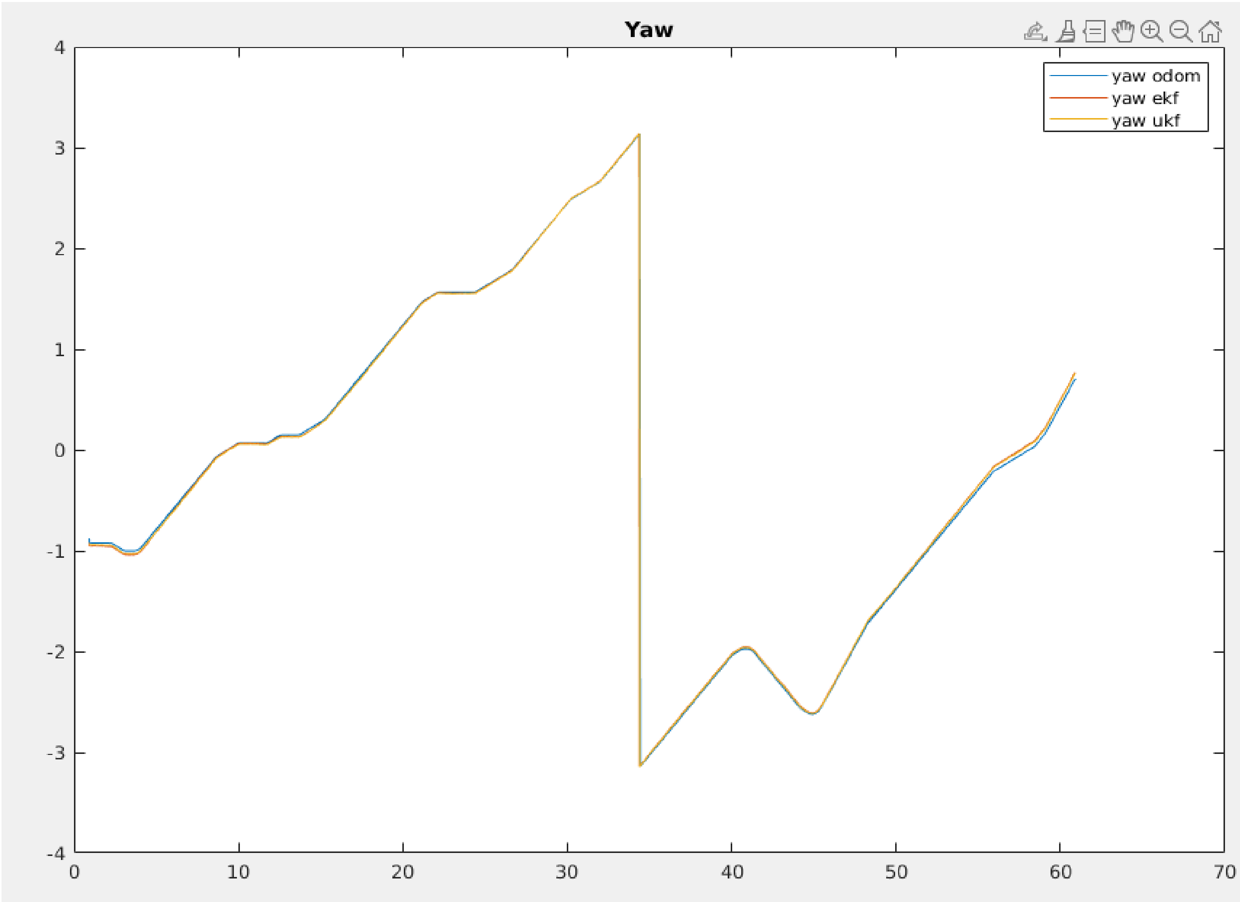

Figure 6: Yaw Coordinate UKF Estimation vs. EKF Estimation vs. Ground Truth

The UKF seems to perform consistently a little worse than the EKF does. I haven't been able to figure out how to remove that persistent error in the UKF's state estimation as of yet.