AERSP 565 Course Project: Sensor Fusion for Estimation of the Position and Velocity of a Differential Drive Mobile Robot

Kasra Sinaei*

Electrical Engineering, Pennsylvania State University

Accurate state estimation is crucial for the autonomous operation of any land vehicle to such a degree that a poor state estimator will impede the performance of navigation controllers. In this study, the dynamics of a differential drive ground mobile were simulated to explore some nonlinear frameworks for estimating its state. As the nonlinearity of the second-order dynamics for this system is considerably high, the discrete-time Extended Kalman Filter (EKF) and hybrid Unscented Kalman Filter (UKF) were chosen for the estimation algorithm and were implemented and tested on simulated data, which includes IMU, GPS, Gyroscope, and Encoder sensor readings. Sensor fusion boosts the robustness of the estimation framework in case some sensor data suddenly becomes unavailable. The Unscented Kalman Filter is one of the most employed techniques for nonlinear state estimation that perfectly accounts for system nonlinearity in its predictions and estimations. The results of this project discuss and compare the accuracy of EKF and UKF for the introduced system.

I. Nomenclature

m	=	mass of the robot (kg)
I	=	robot's inertial moment about z-axis ($kgm^2 \frac{kg}{m^2}$)
$\tau_{r,l}$	=	torque applied to the right, left wheel (Nm)
x, y	=	robot's Cartesian position (m)
θ	=	robot's heading (rad)
$\dot{x}$, $\dot{y}$	=	robot's Cartesian position ($\frac{m}{s}$)
$\dot{\theta}$	=	rate of robot's heading change ($\frac{rad}{s}$)
V	=	robot's forward velocity ($\frac{m}{s}$)
ω	=	robot's angular velocity ($\frac{rad}{s}$)

II. Motivation

Navigation and mapping using the mobile robots is a major challenge that effects the whole autonomous drive problem [1]. Applications of such a systems include a wide range of vehicles such as mars/moon rovers, service robots and autonomous cars [2]. Controls of a moving vehicle depends on how accurate one can provide the estimation of the robot's state including it's position and velocity [3]. Measurement data from IMU (Inertial measurement unit) and GPS (Global Positioning System) could be utilized for a fair estimation in most general cases and robots. To better take into account for non-holonomicity of the robot odometry data could be used as well [4]. In this project a non-linear estimation based on appropriate Kalman Filter algorithm will be exploited to fuse the measurement data of a variety of sensors to improve the estimation of a mobile robot's position and velocity.

III. Project Description

In this project state estimation of a differential drive mobile robot will be studied using fusion of IMU, GPS and odometry data by exploiting Kalman Filter based framework. Generalization of the Kalman Filter [5] that suites this problems where the dynamics of system are non-linear, and the process and measurement noise could be correlated or colored will be examined.

*PhD Student, Electrical Engineering, kasra@psu.edu

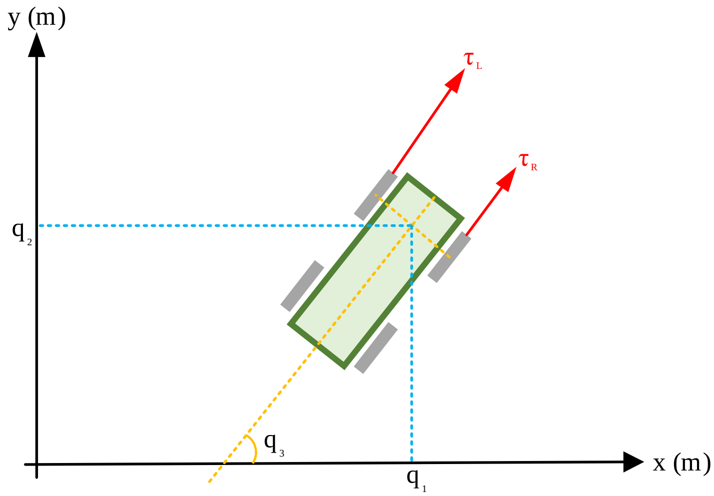

Fig. 1 Schematic of the 2D differential drive robot

This project will be implemented on a simulated differential drive robot using it's dynamic equations of motion and the measurement data will be contaminated with arbitrary fabricated noise to challenge the robustness of estimation framework. The first step of the project will be simulating the non-linear dynamic model of the robot and applying some generic input to the system [6], then using Extended Kalman Filter and Unscented Kalman Filter states of the robot will be estimated and compared to the ground truth data to compute estimation error in presence of noise. Overview of the project phases and objectives are listed below.

1) Simulate dynamics of the robot
2) Apply generic input to the simulated robot
3) Specify measurement model for IMU, GPS and Odometry sensors
4) Implement Extended Kalman filter
5) Examine and compare accuracy of the estimation with and without sensor fusion
6) Implement Unscented Kalman filter and compare its result with Extended Kalman filter

IV. Process and Measurement Models

First take a look at simplified schematic of a two wheel differential drive vehicle. It consists of two separately driven wheels that share a same rotation axis [2]. For simplicity we ignore slipping and assume that there is no level change in the ground, so the robot will remain in a same plane and problem reduces to 2 dimensional translation and one rotation (heading change). The system and it's parameters have been illustrated in figure 1.

A. Process Model

For simulation we need to obtain an accurate representation of the robot's dynamic equation. The classic Newton Euler equations of motion that govern the kinetics of a 2 wheel differential mobile robot could be written as:

$$m\ddot{x} = (\tau_r + \tau_l) r \cos\theta \tag{1}$$

$$m\ddot{y} = (\tau_r + \tau_l) r \sin\theta \tag{2}$$

$$I\ddot{\theta} = (\tau_r - \tau_l)(\frac{wr}{2}) \tag{3}$$

where w is the distance between the two driving wheels and r is the wheel's radius. Refer to figure 1, which illustrates schematic of the mobile robot in a 2D plane with 3 states $\begin{bmatrix} x & y & \theta \end{bmatrix}$, from now on we refer to these configuration as $\begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}$. To obtain non-linear state space equations, we add velocities to the state vector such that $\mathbf{x} = \begin{bmatrix} q_1 & \dot{q}_1 & q_2 & \dot{q}_2 & q_3 & \dot{q}_3 \end{bmatrix}$, and the input vector consist of torque applied to the right and left wheel $\mathbf{u} = \begin{bmatrix} \tau_r & \tau_l \end{bmatrix}^T$; then from dynamic equations of motion presented in equation 3one can obtain the following non-linear system for the mobile robot:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ (\tau_r + \tau_l)(\frac{r}{m}) \cos\theta \\ \dot{y} \\ (\tau_r + \tau_l)(\frac{r}{m}) \sin\theta \\ \dot{\theta} \\ (\tau_r - \tau_l)\frac{wr}{2I} \end{bmatrix} \tag{4}$$

B. Measurement model

For each sensor, we have a separate measurement unit as each represents a different physical property as a result an appropriate measurement model is needed for each [6]. These models will be used in estimation algorithm to compute error covariance and then those values will be exploited to update estimation.

- IMU:

$$y_{IMU} = \begin{bmatrix} \ddot{x}_{IMU_k} \\ \ddot{y}_{IMU_k} \end{bmatrix} \rightarrow h_{IMU} = \begin{bmatrix} 0 & \cos\theta & 0 & \sin\theta & 0 & 0 & 0 \\ 0 & -\sin\theta & 0 & \cos\theta & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

- Encoder:

$$y_{Encoder} = \begin{bmatrix} V_{right} \\ V_{rleft} \end{bmatrix} = \begin{bmatrix} \sqrt{\dot{x}^2 + \dot{y}^2} + \frac{w}{2}\dot{\theta} \\ \sqrt{\dot{x}^2 + \dot{y}^2} - \frac{w}{2}\dot{\theta} \end{bmatrix} \rightarrow h_{Encoder} = \begin{bmatrix} 0 & \frac{\dot{x}}{\sqrt{\dot{x}^2+\dot{y}^2}} & 0 & \frac{\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}} & 0 & \frac{w}{2} & 0 \\ 0 & \frac{\dot{x}}{\sqrt{\dot{x}^2+\dot{y}^2}} & 0 & \frac{\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}} & 0 & \frac{w}{2} & 0 \end{bmatrix} \tag{6}$$

- GPS:

$$y_{GPS} = \begin{bmatrix} x_{GPS} \\ x_{GPS} \end{bmatrix} \rightarrow h_{GPS} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

- Gyro:

$$y_{Gyro} = \begin{bmatrix} b \\ \dot{\theta} \end{bmatrix} \rightarrow h_{Gyro} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{8}$$

Note that the Gyroscopic sensors have some bias in practice, to enhance the estimation and refining the gyro readings we try to estimate the gyro sensor bias as well. To do so, we augment the sensor bias in the process model states. Notice the number of columns in the measurement model (h_{sensor}) which is a 7. The process model with augmented gyroscope bias is:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ (\tau_r + \tau_l)(\frac{r}{m}) \cos\theta \\ \dot{y} \\ (\tau_r + \tau_l)(\frac{r}{m}) \sin\theta \\ \dot{\theta} \\ (\tau_r - \tau_l)\frac{wr}{2I} + b \\ 0 \end{bmatrix} \tag{9}$$

compare the recent equation with equation 4 to distinguish between the actual system model and the one used for estimating sensor bias which includes gyro bias. The state vector for augmented system is $\begin{bmatrix} x & \dot{x} & y & \dot{y} & \theta & \dot{\theta} & b \end{bmatrix}^T$.

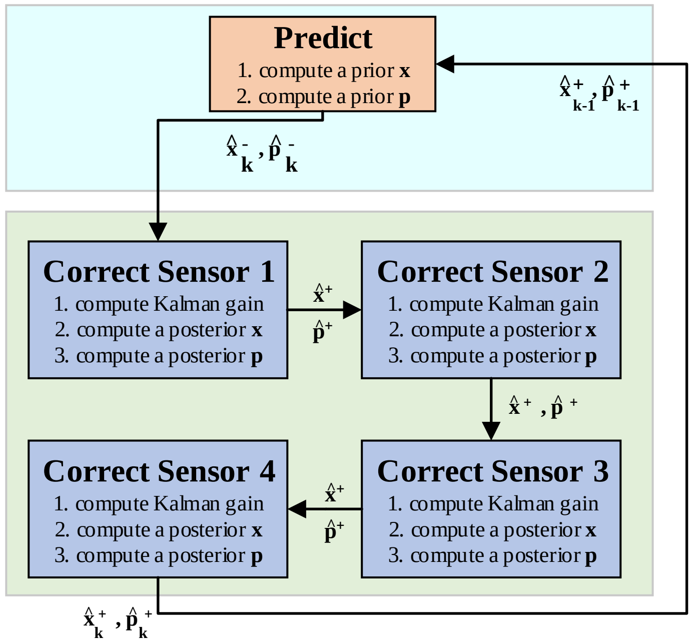

Fig. 2 Schematic block diagram of sensor fusion using Kalman Filter

V. State Estimation and Sensor Fusion

A. Sensor Fusion

In this section, the procedure of using Non-Linear Kalman Filter algorithms to exploit measurements of multiple sensors simultaneously will be discussed. This process is known as sensor fusion and is commonly used in a wide variety of systems. [7] Herein, measurements of IMU, Gyroscope and encoder odometry data are available at the rate of 100Hz and the GPS data is available at 20Hz rate. Using any Kalman Filter framework we can combine all these sensors to enhance estimation of the robot's state.

B. Discrete Extended Kalman Filter

Discrete time linearized Kalman filter works almost the same as regular Kalman filter. The only difference is that instead of a linear model, using partial derivatives of $f(\mathbf{x}, \mathbf{u})$ and $H(\mathbf{x}, \mathbf{u})$ along the trajectory of the robot a linear approximation of the system dynamics and it's measurement models will be used in prediction and update stage [5]. Equations 12-16 are used for one time step update of the EKF.

Assume the following equations represent the partial derivatives for process and measurement models:

$$F_{k-1} = \left.\frac{\partial f_{k-1}}{\partial x}\right|_{\hat{x}^+_{k-1}} \quad L_{k-1} = \left.\frac{\partial f_{k-1}}{\partial w}\right|_{\hat{x}^+_{k-1}} \tag{10}$$

$$H_k = \left.\frac{\partial h_k}{\partial x}\right|_{\hat{x}^-_k} \quad M_k = \left.\frac{\partial h_k}{\partial v}\right|_{\hat{x}^-_k} \tag{11}$$

also note that all states of the system are contaminated with noise as well as sensor readings. This means $L = \mathbb{I}^{7\times7}$ and $h_{sensor} = \mathbb{I}^{m\times m}$, where $\mathbb{I}$ represent the identity matrix and m is the dimension of sensor data.

time update (prediction step)

$$P^-_k = F_{k-1}P^+_{k-1}F^T_{k-1} + L_{k-1}Q_{k-1}L^T_{k-1} \tag{12}$$

$$\hat{x}^-_k = f_{k-1}\left(\hat{x}^+_{k-1}, u_{k-1}, 0\right) \tag{13}$$

measurement update (repeat for each sensor)

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + M_k R_k M_k^T \right)^{-1} \tag{14}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \left[y_k - h_k \left(\hat{x}_k^-, 0 \right) \right] \tag{15}$$

$$P_k^+ = \left(I - K_k H_k \right) P_k^- \tag{16}$$

C. Hybrid Unscented Kalman Filter

To improve the estimation of systems with severe nonlinearities, this algorithm uses an unscented transform to estimate the mean and covariance of the random variables under nonlinear transforms. This approach results in a significantly lower linearization error compared to EKF [5]. One step time update for estimation using the UKF is as follows:

- Initialize Filter

$$\hat{x}_0^+ = E\left(x_0\right) \tag{17}$$

$$P_0^+ = E\left[\left(x_0 - \hat{x}_0^+ \right) \left(x_0 - \hat{x}_0^+ \right)^T \right] \tag{18}$$

- Generate adequate sigma point, transform them using the non-linear state propagation function $f(.)$ and predict mean and error covariance.

$$\hat{x}_{k-1}^{(i)} = \hat{x}_{k-1}^+ + \tilde{x}^{(i)} \quad i = 1, \cdots, 2n \tag{19}$$

$$\tilde{x}^{(i)} = \left(\sqrt{nP_{k-1}^+} \right)_2^T \quad i = 1, \cdots, n \tag{20}$$

$$\tilde{x}^{(n+i)} = -\left(\sqrt{nP_{k-1}^+} \right)_i^T \quad i = 1, \cdots, n \tag{21}$$

$$\hat{x}_k^{(i)} = f\left(\hat{x}_{k-1}^{(i)}, u_k, t_k \right) \tag{22}$$

update prior estimation:

$$\hat{x}_k^- = \frac{1}{2n} \sum_{i=1}^{2n} \hat{x}_k^{(i)} \tag{23}$$

$$P_k^- = \frac{1}{2n} \sum_{\imath=1}^{2n} \left(\hat{x}_k^{(i)} - \hat{x}_k^- \right) \left(\hat{x}_k^{(i)} - \hat{x}_k^- \right)^T + Q_{k-1} \tag{24}$$

- chose sigma points based on the prior, transform them,

$$\hat{x}_k^{(i)} = \hat{x}_k^- + \tilde{x}^{(i)} \quad i = 1, \cdots, 2n \tag{25}$$

$$\tilde{x}^{(i)} = \left(\sqrt{nP_k^-} \right)_\imath^T \quad i = 1, \cdots, n \tag{26}$$

$$\tilde{x}^{(n+i)} = -\left(\sqrt{nP_k^-} \right)_i^T \quad i = 1, \cdots, n \tag{27}$$

compute predicted measurements and calculate the error,

$$\hat{y}_k^{(i)} = h\left(\hat{x}_k^{(i)}, t_k \right) \tag{28}$$

where $\hat{y}_k^{(i)} = h\left(\hat{x}_k^{(i)}, t_k \right)$; finally use Kalman Filter to update the posterior estimation.

$$K_k = P_{xy} P_y^{-1} \tag{29}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \left(y_k - \hat{y}_k \right) \tag{30}$$

$$P_k^+ = P_k^- - K_k P_y K_k^T \tag{31}$$

$$\tag{32}$$

where $P_y = \frac{1}{2n} \sum_{\imath=1}^{2n} \left(\hat{y}_k^{(i)} - \hat{y}_k \right) \left(\hat{y}_k^{(i)} - \hat{y}_k \right)^T + R_k$, and $P_{xy} = \frac{1}{2n} \sum_{i=1}^{2n} \left(\hat{x}_k^{(i)} - \hat{x}_k^- \right) \left(\hat{y}_k^{(i)} - \hat{y}_k \right)^T$. For more detailed explanations and procedures refer to chapter 13 of [5].

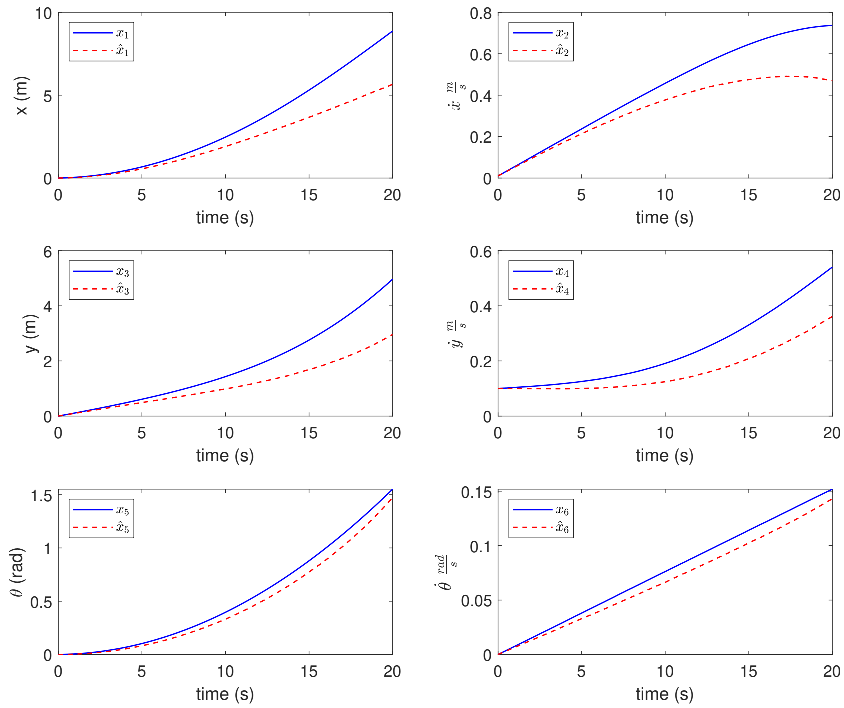

Fig. 3 Estimation of robot using Extended Kalman Filter (EKF) for 20 seconds

VI. Simulation and Results

An arbitrary input has been applied to the simulated robot in MATLAB, and sensor measurements that have been contaminated with zero-mean Gaussian noise, as well as the true state of the robot, have been saved in .mat files to be used as ground truth. The Extended Kalman Filter (EKF) is quite straightforward to implement; refer to 'SensorFusion.mlx' to see the details and executed codes. The Unscented Kalman Filter (UKF) is slightly more difficult to implement, and a MATLAB class has been dedicated to the computations. See the details in 'hybridUKF.m'. The unscented transform has been used twice in this algorithm: once for updating the prior (predict) and once for updating the posterior (update). Refer to 'unscentedTransform.m' for more details.

Figure 3 scrutinizes the performance of discrete time EKF. As shown in the mentioned figure, there is a considerable drift in both position and velocity estimation. These values follow a second order dynamic system that was described as a combination of non-linear terms including trigonometric functions and divisions. However, the gyroscope bias estimation is more accurate compared to other states of the system.

Figure 4 illustrates the gyroscopic sensor bias estimation. Even-though, it was initialized at the value 0, the filter has estimated the true value in the first few iterations.

The estimation precision for the UKF could be examined via the plots of figure 5. As you can see the drift between the true state and estimated states has been diminished significantly compared to EKF. This is the result of using non-linear unscented transform which approximates the mean and covariance more accurately than the conventional linearization used in the EKF.

To better see the performance gap between these two methods robot's trajectory in the 2D plane has been juxtaposed to the estimated trajectories of EKF and UKF in the figure 6. Overall, it can be concluded that the Unscented Kalman Filter (UKF) provides significantly better state estimation compared to the Extended Kalman Filter (EKF) when applied to a nonlinear model. The results of the simulation demonstrate that UKF is able to handle the high nonlinearity of the system dynamics much more effectively than EKF. Therefore, UKF is a more suitable estimation technique for nonlinear systems with severe nonlinearity.

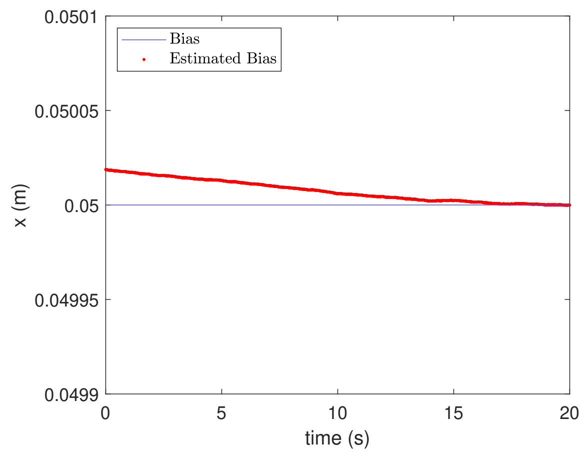

Fig. 4 Gyroscope Bias Estimation

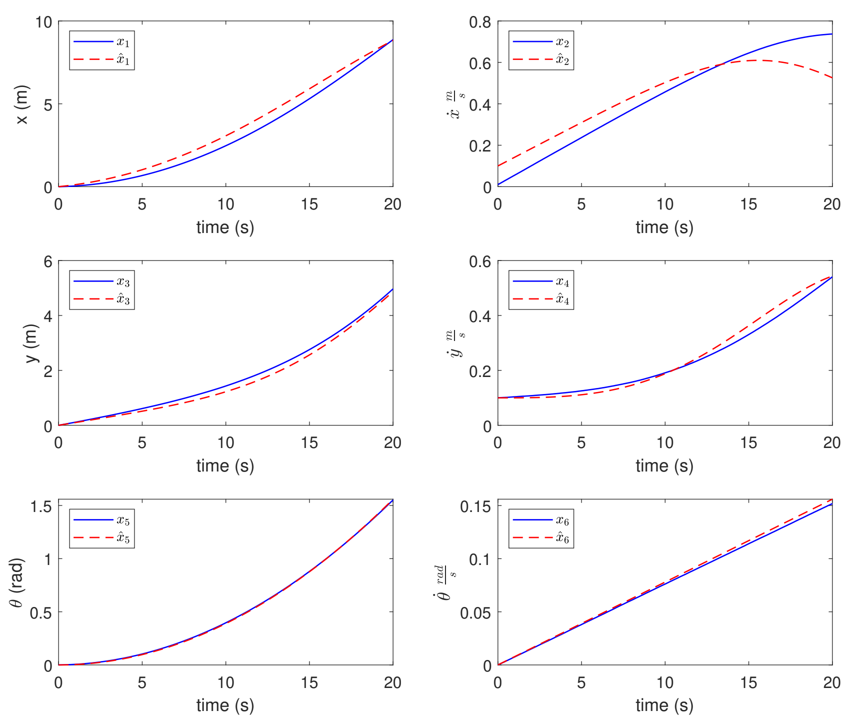

Fig. 5 Estimation of robot using Unscented Kalman Filter (UKF) for 20 seconds

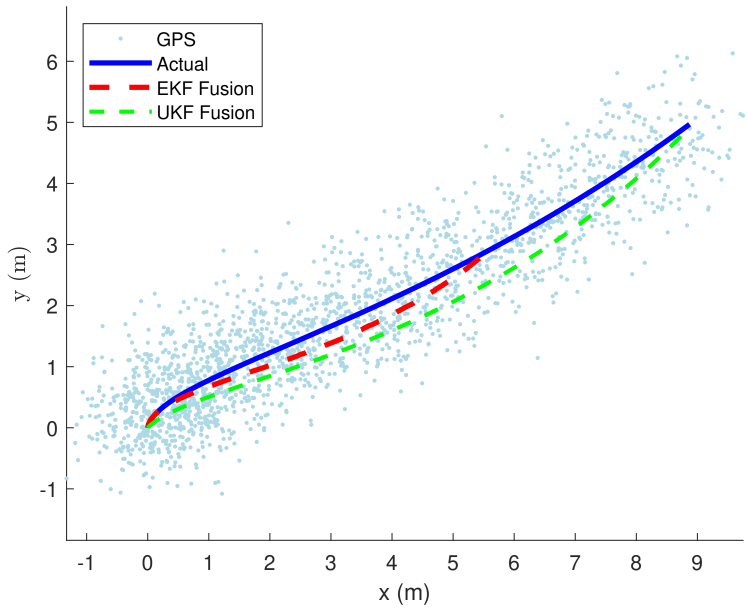

Fig. 6 Estimated trajectory of the robot using the EKF and UKF compared to the ground truth data

Appendix

A folder including MATLAB codes that contain all sources to regenerate simulation results of this project, has been submitted with this document.

- **EKF**: update_post.m, update_prior.m, encoder_derivative.m, imu_derivative.m, gyro_derivative.m, gps_derivative.m
- **UKF**: systemDynamics.m, observationEquation.m, unscentedTransform.m, hybridUKF.m
- **Simulation and Plots**: SensorFusion.mlx, draw_fig.m, draw_traj.m

References

[1] Von Der Hardt, H.-J., Wolf, D., and Husson, R., "The dead reckoning localization system of the wheeled mobile robot ROMANE," *1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems (Cat. No. 96TH8242)*, IEEE, 1996, pp. 603–610.

[2] Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D., *Introduction to autonomous mobile robots*, MIT press, 2011.

[3] Klancar, G., Zdesar, A., Blazic, S., and Skrjanc, I., *Wheeled mobile robotics: from fundamentals towards autonomous systems*, Butterworth-Heinemann, 2017.

[4] Girbés-Juan, V., Armesto, L., Hernández-Ferrándiz, D., Dols, J. F., and Sala, A., "Asynchronous sensor fusion of GPS, IMU and CAN-based odometry for heavy-duty vehicles," *IEEE Transactions on Vehicular Technology*, Vol. 70, No. 9, 2021, pp. 8617–8626.

[5] Simon, D., *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*, John Wiley & Sons, 2006.

[6] Al Khatib, E. I., Jaradat, M. A., Abdel-Hafez, M., and Roigari, M., "Multiple sensor fusion for mobile robot localization and navigation using the Extended Kalman Filter," *2015 10th international symposium on mechatronics and its applications (ISMA)*, IEEE, 2015, pp. 1–5.

[7] Van Der Merwe, R., Wan, E., and Julier, S., "Sigma-point Kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation," *Aiaa guidance, navigation, and control conference and exhibit*, 2004, p. 5120.