Sensor Fusion

Manfred Huber

With material from Onur Daskiran, Kamesh Subarrao, and Yan Wan

Fall 2019

Uncertainty

- Sensors are used to provide information about the state of the system
  - Encoders and other internal sensors allow for odometry (calculation of UxV movement)
  - GPS and other external sensors allow to estimate (relative) position based on knowledge of the environment
- Sensor readings contain noise
  - Sensor-based calculations of the state are not accurate

Uncertainty

- Uncertainty in sensors has consequences
  - The effect of errors in sensors that measure change in state add up over time
    - These sensor errors result in drift
    - UxVs using just odometry will eventually get lost
  - The effect of uncertainty in sensors that measure state do not increase
    - These sensor errors produce “jitter”
    - Multiple readings from the same or different sensors in the same state tend to not be consistent
  - To use sensors we need to integrate sensor information

Fusion

- Fusion is the process of combining information from multiple sources or readings
  - Information fusion - Merging of information from heterogeneous sources with differing conceptual, contextual and typographical representations. Involves the combination of information into a new set of information towards reducing uncertainty.
  - Data fusion - The process of integration of multiple data and knowledge representing the same real-world object into a consistent, accurate, and useful representation
  - Sensor fusion – Combination of measurements from several sensors (Multi-Sensor Data Fusion)

Fusion

- Information fusion and data fusion are often employed as synonyms
  - data fusion is used for raw data (obtained directly from the sensors)
  - information fusion is employed to define already processed data.
  - information fusion implies a higher semantic level than data fusion.
- Other terms associated with data fusion include decision fusion, data combination, data aggregation, multisensor data fusion, and sensor fusion.

Fusion (from Durrant-Whyte)

- Different types of data can be combined
  - Complementary: Information provided by the input sources represents different parts of the scene and could thus be used to obtain more complete global information. For example, in the case of visual sensor networks, the information on the same target provided by two cameras with different fields of view is considered complementary;
  - Redundant: When two or more input sources provide information about the same target and could thus be fused to increment the confidence. For example, the data coming from overlapped areas in visual sensor networks are considered redundant;
  - Cooperative: The provided information is combined into new information that is typically more complex than the original information. For example, multi-modal (audio and video) data fusion is considered cooperative.

Sensor Fusion and Navigation

- One of the most prominent applications of Sensor Fusion has been in Navigation.
  - Navigation System – A filter that integrates information from encoders, GPS, gyroscopes, accelerometers, magnetometer measurements. Find where you are w.r.t some reference.
  - A challenging application because of the complexity and numerical burden (process data)
  - On a vehicle platform, there are sensors that relate its orientation to the world or to the platform's frame.

Sensor Fusion

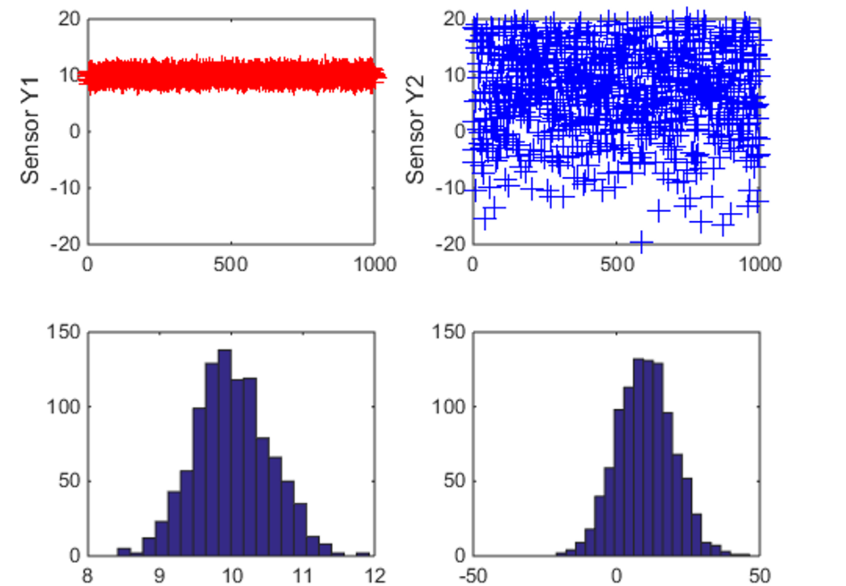
- How do we combine sensor data
  - If the data is orthogonal (i.e. it measures completely independent aspects) fusion is simple
    - Just need to combine the sensor readings into one sensor vector
  - Redundant data involves multiple measurements of the same properties
    - Need to find some "middle"
  - Most sensors in UxVs provide complementary (but not orthogonal) data
    - GPS and IMU measure position and velocity which are complementary

Fusing Redundant Data

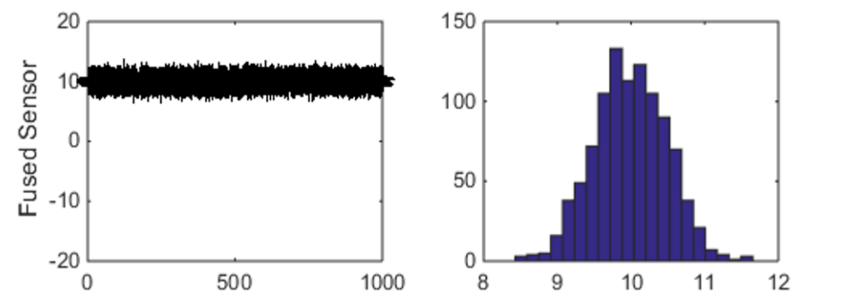
- When combining multiple observations of the same state, we are interested in obtaining the "best" estimate
  - The most common definition of "best" is the one having the lowest expected squared error
    - If the two readings are equally reliable this is the mean of the readings (the average)
  - But what if the two readings are not equally reliable ?
    - They could be from different sensors
    - They could have been taken with different noise

Fusing Redundant Data

- Two sensors y_1 and y_2 to estimate x
  - $y_1 = x + n_1$
  - $y_2 = x + n_2$
- How to estimate x ?
  - Intuitive solutions
    - Delete one measurement?
    - Take the average?
  - Neither is the "best"
    - Need more of the better and less of the worse



Small variance $\sigma_1{}^2$ Large variance $\sigma_2{}^2$



Least Squares Estimator

- Two sensors y_1 and y_2 to estimate x
  - $y_1 = x + n_1$
  - $y_2 = x + n_2$

- *Find* $\hat{x}$ such that

$$J = \frac{1}{2}(\boldsymbol{y} - \hat{x})^T \begin{bmatrix} {\sigma_1}^{-2} & 0 \\ 0 & {\sigma_2}^{-2} \end{bmatrix} (\boldsymbol{y} - \hat{\boldsymbol{x}})$$ is minimized

  - $\hat{x} = \frac{{\sigma_2}^2}{{\sigma_1}^2 + {\sigma_2}^2} y_1 + \frac{{\sigma_1}^2}{{\sigma_1}^2 + {\sigma_2}^2} y_2$

Least Squares Estimator

- More generally:
  - Sensor readings

    $$\boldsymbol{y} = H\boldsymbol{x} + \boldsymbol{n}$$

  - Define H as the covariance matrix
    - Models uncertainty of the sensors and the correlations between the errors of the sensors
  - Minimize $J = \frac{1}{2}(\boldsymbol{y} - H\widehat{\boldsymbol{x}})^T R^{-1} (\boldsymbol{y} - H\widehat{\boldsymbol{x}})$
  - Take derivative of J with respect to $\widehat{\boldsymbol{x}}$ gives
    - $\widehat{\boldsymbol{x}} = (H^T R^{-1} H)^{-1} H^T R^{-1} \boldsymbol{y}$

Fusing Cooperative Data

- In UxVs sensors usually provide different information
  - GPS and laser provide position-related data
  - Encoders and Accelerometers provide velocity-related data
  - Lidar provides relative information data
- How to compute a best estimate out of those
  - Need to model the effect of one type of information on the other
    - Velocity influences position
    - Provides the potential to address “drift” and “jitter”

Bayesian Estimation

- One way to think of the fusion problem is in terms of probabilities
  - Each sensor measurement can indicate a state with a certain probability $P(s \mid o)$
    - Observation probability can be easier measured
      - $P(o \mid s)$ characterizes the precision of the sensor
  - The uncertainty in the relation between states over time can be captured in a transition model
    - $P(s_{t+1} \mid s_t, a_t)$
    - This also captures the relation between velocity and position

Bayesian Estimation

- A Bayesian filter computes the posterior distribution of the state using the observations
  - Discrete case:

$$P(s_t \mid o_t, o_{t-1}, \ldots, o_1) = \frac{P(o_t \mid s_t, o_{t-1}, \ldots, o_1) P(s_t \mid o_{t-1}, \ldots, o_1)}{P(o_t \mid o_{t-1}, \ldots, o_1)}$$

  - Continuous case:

$$p(s_t \mid o_t, o_{t-1}, \ldots, o_1) = \frac{p(o_t \mid s_t, o_{t-1}, \ldots, o_1) p(s_t \mid o_{t-1}, \ldots, o_1)}{p(o_t \mid o_{t-1}, \ldots, o_1)}$$

Bayesian Estimation

- A Bayesian filter computes the posterior distribution of the state using the observations

  - Continuous case:

$$p(s_t \mid o_t, o_{t-1}, \ldots, o_1) = \frac{p(o_t \mid s_t, o_{t-1}, \ldots, o_1)\, p(s_t \mid o_{t-1}, \ldots, o_1)}{p(o_t \mid o_{t-1}, \ldots, o_1)}$$

$$= \frac{p(o_t \mid s_t, o_{t-1}, \ldots, o_1) \int_{s_{t-1}} p(s_t \mid s_{t-1}, o_{t-1}, \ldots, o_1)\, p(s_{t-1} \mid o_{t-1}, \ldots, o_1)\, ds_{t-1}}{\int_{s_{t-1}} p(o_t \mid s_{t-1,} o_{t-1}, \ldots, o_1)\, p(s_{t-1} \mid o_{t-1}, \ldots, o_1) ds_{t-1}}$$

Bayesian Estimation

- If the process is Markov the recursive Bayesian filter can be derived
  - Continuous case:

$$p(s_t \mid o_t, o_{t-1}, \dots, o_1) = \frac{p(o_t \mid s_t, o_{t-1}, \dots, o_1) \int_{s_{t-1}} p(s_t \mid s_{t-1}, o_{t-1}, \dots, o_1)\, p(s_{t-1} \mid o_{t-1}, \dots, o_1)\, ds_{t-1}}{\int_{s_{t-1}} p(o_t \mid s_{t-1,} o_{t-1}, \dots, o_1)\, p(s_{t-1} \mid o_{t-1}, \dots, o_1) ds_{t-1}}$$

$$= \frac{p(o_t \mid s_t) \int_{s_{t-1}} p(s_t \mid s_{t-1})\, p(s_{t-1} \mid o_{t-1}, \dots, o_1)\, ds_{t-1}}{\int_{s_{t-1}} p(o_t \mid s_{t-1})\, p(s_{t-1} \mid o_{t-1}, \dots, o_1) ds_{t-1}} = \alpha\, p(o_t \mid s_t) \int_{s_{t-1}} p(s_t \mid s_{t-1})\, p(s_{t-1} \mid o_{t-1}, \dots, o_1)\, ds_{t-1}$$

Recursive Bayesian Filter

- The recursive Bayesian filter can be broken into two phases
  - Prediction:

$$p(s_t \mid o_{t-1},\ldots,o_1) = \int_{s_{t-1}} p(s_t \mid s_{t-1})\, p(s_{t-1} \mid o_{t-1},\ldots,o_1)\, ds_{t-1}$$

  - Measurement:

$$p(s_t \mid o_t, o_{t-1},\ldots,o_1) = \frac{p(o_t \mid s_t)}{p(o_t \mid o_{t-1},\ldots,o_1)}\, p(s_t \mid o_{t-1},\ldots,o_1)$$

Recursive Bayesian Filter

- Benefits of a Bayesian filter
  - Optimal estimates
  - No assumptions about distributions
  - Uniform framework

- Problems of the filter
  - Often computationally intractable
  - Integral might not be analytically solvable

Kalman Filter

- The Kalman filter is a special case of the recursive Bayesian filter for the following assumptions:
  - System and observation model are linear

$$s_t = As_{t-1} + Ba_{t-1} + w_t$$
$$o_t = Hs_t + v_t$$

  - The uncertainty in the system and observation models are Gaussian

$$w_t \sim N(0, Q)$$
$$v_t \sim N(0, R)$$

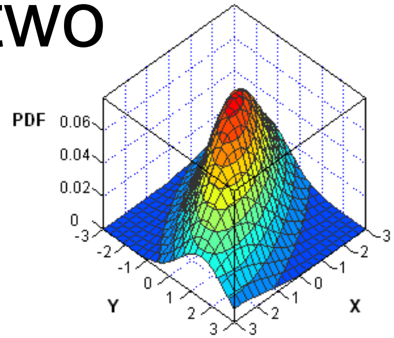
Kalman Filter

- The Kalman filter estimates the posterior distribution in terms of the mean and the Covariance matrix

$$\hat{s}_t = E[s_t]$$

$$P_t = E[(s_t - \hat{s}_t)(s_t - \hat{s}_t)^T]$$

- The posterior distribution is a Gaussian distribution (maintaining the first two moments of the distribution)



Discrete Kalman Filter

- The discrete Kalman filter is a special version of the recursive Bayesian filter
  - Prediction:

$$\hat{s}_t^- = A\hat{s}_{t-1} + Ba_{t-1}$$

$$P_t^- = AP_{t-1}A^T + Q$$

  - Measurement:

$$\hat{s}_t = \hat{s}_t^- + K_t(o_t - H\hat{s}_t^-)$$

$$K_t = P_t^- H^T (HP_t^- H^T + R)^{-1}$$

$$P_t = (I - K_t H)P_t^-$$

Kalman Filter Example

- Ball moving on a track with unknown length and velocity that bounces perfectly at the ends. Observations of the position.
  - What do we need as part of the state ?
    - At least position, x, and velocity, ϑ
    - Could include track length to be more precise
      - Will here model bounce as noise (Violates Gaussian assumption)

$$s_t = \begin{pmatrix} x_t \\ \vartheta_t \end{pmatrix}$$

$$s_t = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} s_{t-1} + w_t$$

$$o_t = s_t + v_t$$

Kalman Filter Example (2)

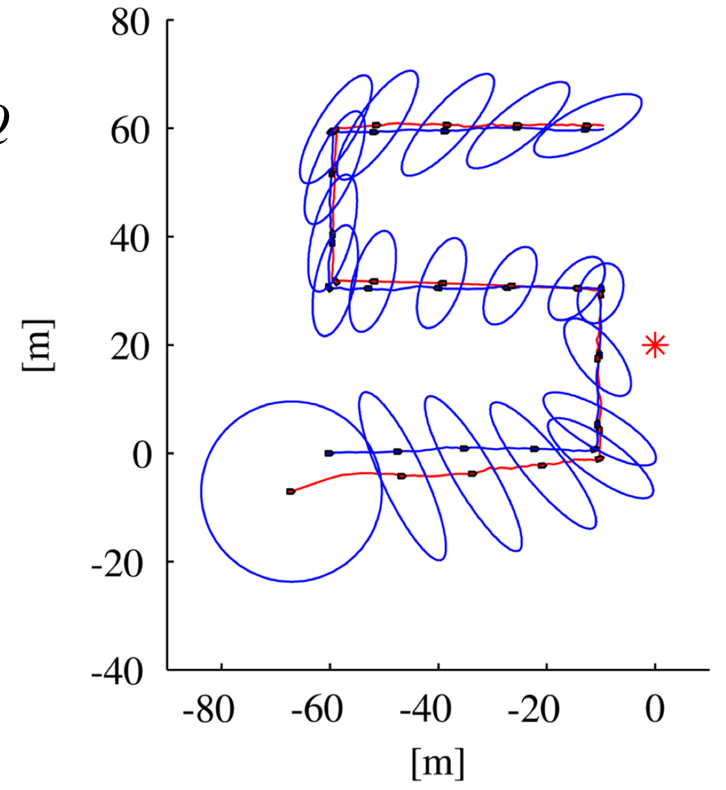
- Example for a simplified 2D holonomic robot using a simple directional Position signal

$$\begin{pmatrix} x \\ y \end{pmatrix}_t^- = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_{t-1} + \begin{pmatrix} \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix} + Q$$

$$\begin{pmatrix} z_x \\ z_y \end{pmatrix}_t^- = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}_t + R$$

$$Q = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$$

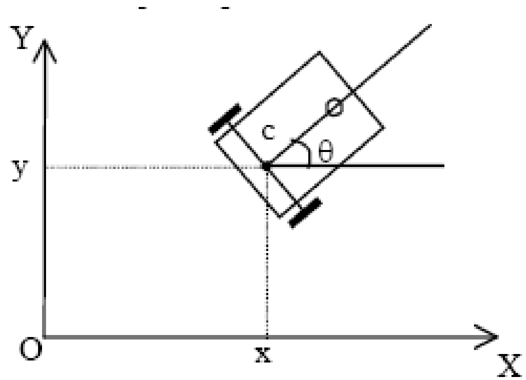
$$R = \begin{pmatrix} \sigma_{z_x}^2 & \sigma_{z_x z_y} \\ \sigma_{z_x z_y} & \sigma_{z_y}^2 \end{pmatrix}$$



Extended Kalman Filter

- In UxS the system model is frequently not linear
  - Orientation introduces nonlinearity
- The Extended Kalman Filter (EKF) relaxes the requirement on linear models
  - Uses the Jacobian matrix as a locally linear approximation of the function.
  - Note: The EKF does not always converge to the correct solution

EKF Equations



- Simplified kinematics for the UGV

$$\dot{\hat{x}} = \frac{r}{2}(\hat{\omega}_L + \hat{\omega}_R)\cos\hat{\theta}$$

$$\dot{\hat{y}} = \frac{r}{2}(\hat{\omega}_L + \hat{\omega}_R)\sin\hat{\theta}$$

$$\dot{\hat{\theta}} = \frac{r}{b}(\hat{\omega}_L - \hat{\omega}_R)$$

$$\dot{x} = V cos(\theta)$$
$$\dot{y} = V sin(\theta)$$
$$\dot{\theta} = \Omega$$

- Sensor observations from GPS and IMU
  - GPS, Accelerometers, Gyroscopes, Magnetometer
  - Could also use Encoders

EKF Equations

- Discretizing the equations:

$$\begin{aligned}\dot{x} &= V\cos(\theta)\\ \dot{y} &= V\sin(\theta)\\ \dot{\theta} &= \Omega\end{aligned} \quad\rightarrow\quad \dot{x} = \frac{x_{k+1} - x_k}{\Delta t} \quad\rightarrow\quad \begin{aligned}x_{k+1} &= x_k + V_k\cos(\theta_k)\Delta t\\ y_{k+1} &= y_k + V_k\sin(\theta_k)\Delta t\\ \theta_{k+1} &= \theta_k + \Omega_k\Delta t\end{aligned}$$

- Adding linear and angular velocity terms

$$\begin{aligned}x_{k+1} &= x_k + V_k\cos(\theta_k)\Delta t\\ y_{k+1} &= y_k + V_k\sin(\theta_k)\Delta t\\ \theta_{k+1} &= \theta_k + \Omega_k\Delta t\\ V_{k+1} &= \frac{r}{2}\left(\omega_{L,k} + \omega_{R,k}\right)\\ \Omega_{k+1} &= \frac{r}{b}\left(\omega_{L,k} - \omega_{R,k}\right)\end{aligned}$$

$$\mathrm{s_k} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ V_k \\ \Omega_k \end{bmatrix}$$

$$\mathrm{s_{k+1}} = \mathrm{f}(\mathrm{s_k}, a_k)$$

EKF Equations

- We need to calculate the Jacobian for the state transition matrix

$$F_{k+1} = \frac{\partial f}{\partial s}\bigg|_{s_k,u_k} = \begin{bmatrix} 1 & 0 & -V_k \sin(\theta_k)\Delta t & \cos(\theta_k)\Delta t & 0 \\ 0 & 1 & V_k \cos(\theta_k)\Delta t & \sin(\theta_k)\Delta t & 0 \\ 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Measurement matrix is linear assuming sensor unit conversions and frame transformations are performed previously

$$\mathrm{o_k} = h(\mathrm{s_k}) = H_k s_k$$

$$\begin{bmatrix} x_{GPS} \\ y_{GPS} \\ \theta_{Mag} \\ V_{Acc} \\ \Omega_{Gyro} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ V_k \\ \Omega_k \end{bmatrix}$$

EKF Equations

- The filter equations are the same as for the standard Kalman Filter
  - Time Update

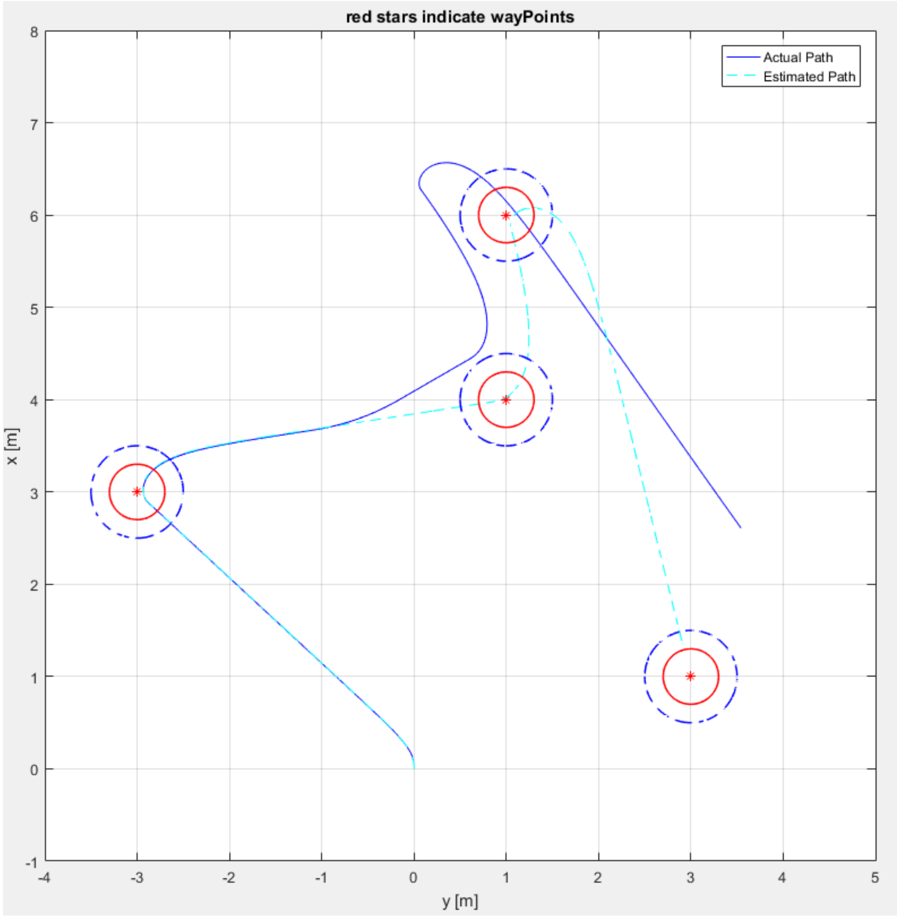
$$s_{k+1}^{-} = f(s_k, a_{k+1})$$
$$P_{k+1}^{-} = F_{k+1} P_k F_{k+1}^T + Q_{k+1}$$

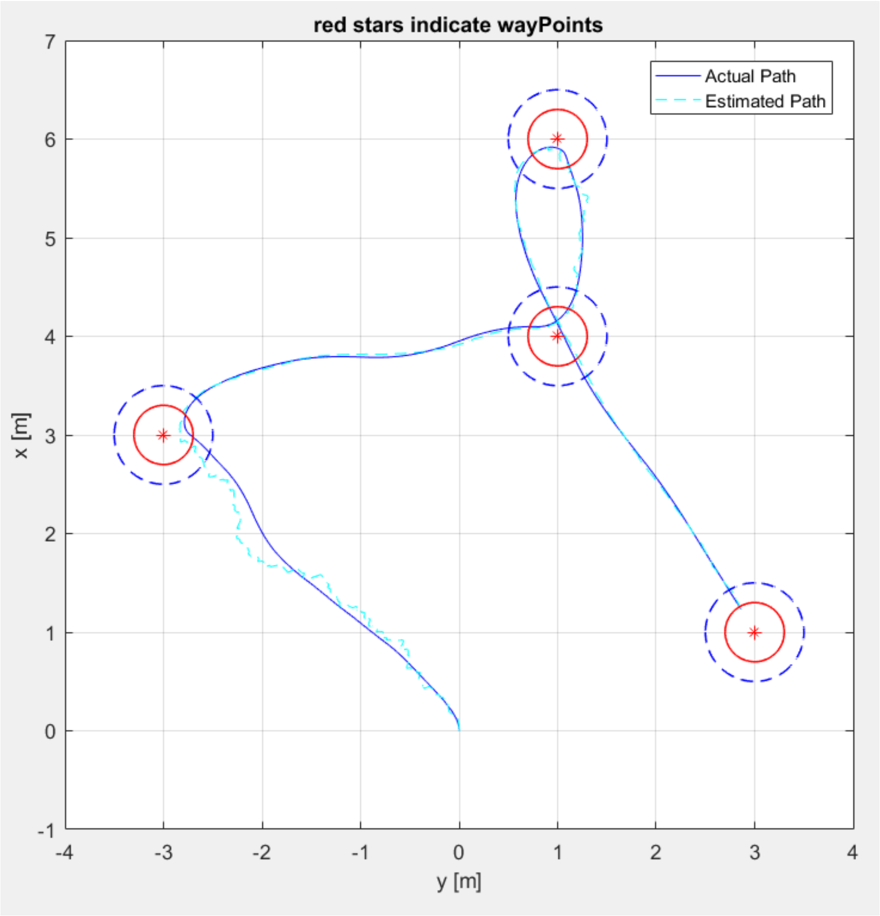
  - Measurement Update

$$\delta_{k+1} = o_{k+1} - h(s_{k+1}^{-})$$
$$K_{k+1} = P_{k+1}^{-} H_{k+1}^T (H_{k+1} P_{k+1}^{-} H_{k+1}^T + R_{k+1})^{-1}$$

$$s_{k+1} = s_{k+1}^{-} + K_{k+1} \delta_{k+1}$$
$$P_{k+1} = (I - K_{k+1} H_{k+1}) P_{k+1}^{-}$$

Simulation Result





Sensor Fusion

- Sensor fusion allows to integrate sensor data from multiple sources with other knowledge about the behavior of the system
  - Bayesian filtering/estimation provides a unified framework for optimal estimation
    - System model captures knowledge about system behavior
    - Observation model captures sensor characteristics
  - If system model is linear and uncertainty is Gaussian, the Kalman Filter provides an efficient solution
  - If the system is non-linear but the uncertainty is Gaussian, the Extended Kalman Filter can be used