VNU University of Engineering and Technology

ALGORITHMS FOR INTELLIGENT ROBOTS

Lecture 4
Robot Localization with Kalman Filtering

UET-VNU

For numerous tasks a mobile robot needs to know "where it is" either on an ongoing basis or when specific events occur.

vị trí + hướng

- Strong localization – estimating the location of the robot with respect to some global representation of space
- The weak localization problem – knowing if the current location has been visited before.
- The estimation of the location of the robot with respect to the map is known as *localization*, *pose estimation*, or *positioning*.
  - The problem starts with an initial estimate of the robot's location $\boldsymbol{x}$ given by a probability distribution $P(\mathbf{x})$.
  - Sensor based localization is based on the premise that we use sensor data $\mathbf{z}$ in conjunction with a map to produce a refined position estimate $P(\mathbf{x}|\mathbf{z})$.
  - This refined estimate has an increased probability density about the true position of the robot.

- In certain circumstances it may be necessary to infer the robot's position without an a priori estimate of its location (*global localization*).
- A more common version of the localization problem is the need to refine an estimate of the robot's pose continually. This task is known as *pose maintenance* or *local localization.*
- Some methods:
  - Dead reckoning
  - Landmark Measurement
  - Filtering (Kalman filter, particle filter…)

Kalman filtering

- Kalman filtering is an algorithm that uses a series of measurements observed over time, including statistical noise and other inaccuracies, to produce estimates of unknown variables that tend to be more accurate than those based on a single measurement alone.
- If the system and measurement process satisfy certain properties, the Kalman filter provides the provably optimal method for fusing the data.

- State estimation techniques, such as the Kalman filter, typically require:
  - A model of how the system of interest evolves over time. In the context of control theory, the description of the system whose state is of interest is usually referred to as the ***plant model.***
  - A description of permitted control inputs $\mathbf{u}(t)$ and how they affect the system state.
  - A model of how the sensors operate.

Recall: State space models

- A system can be described by a set of parameters or variables that characterize the relevant aspects of its behaviour. The information of interest regarding a system is described by a vector **x**.
- The vector **x** specifies a point in the state space. Some or all of these variables may not be directly measurable. Thus, the state vector must be estimated using some vector of measurements **z**.

- **<u>Plant model</u>**

- The plant model describes how the system state $\boldsymbol{x}_k$ changes as a function of time and control input $\boldsymbol{u}_k$ and noise $\boldsymbol{w}_k$

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1}$$

where *f*(.) is the **state transition function** and $\mathbf{w}_\mathbf{k}$ is a noise function. A common and convenient form for the noise model is zero-mean Gaussian noise with covariance $\boldsymbol{C}_{w_k}$. Of particular interest is a linear plant model of the form:

$$\boxed{x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}}$$

where *A* expresses how the system evolves from one state to another in the absence of inputs and *B* expresses how control inputs modify the state.

- **<u>Measurement model</u>**
  - The measurement model describes how sensor data vary as a function of the system state.

$$\boxed{z_k = h(x_k) + v_k}$$

  - where v_k is a **noise function**, x is the **state of the environment**, and *h*(.) is a function that describes sensor measurements as a function of the system state.
  - Again, a common description of the measurement noise is zero-mean Gaussian noise function with covariance C_{v_k}.
  - A linear measurement model is: $\boxed{z_k = Hx_k + v_k}$

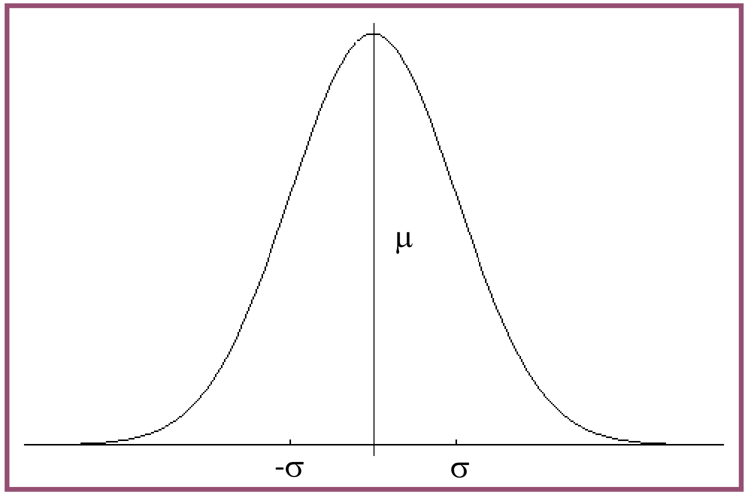
where H is a matrix that expresses how measurements are derived as a linear transformation of the state.

Gaussians

$$p(x) \sim N(\mu, \sigma^2):$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

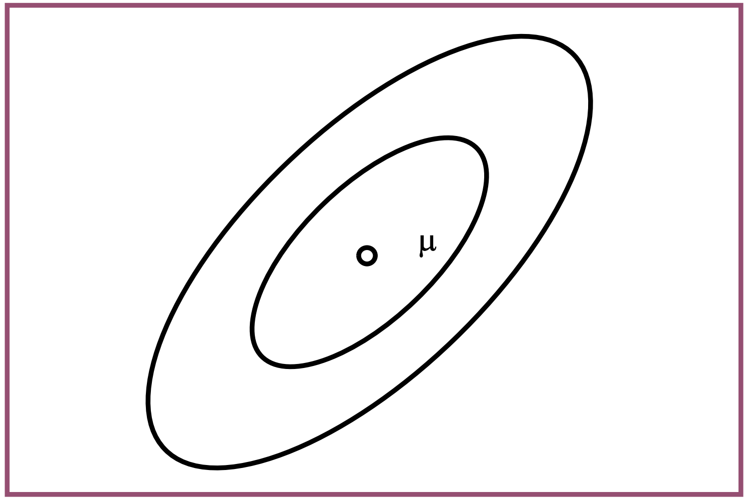
Univariate



$$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}):$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Multivariate



Kalman filtering (for linear systems)

The assumptions made by the Kalman filter are as follows:

- System and measurement noises are Gaussian with zero mean $E[v_i] = 0$ where $E[\ ]$ is the expected value.
- Independent noise: $E[v_i * v_j] = 0$ when $i \neq j$. The system noise variance is given by $E[v_i * v_i] = \sigma_i^2 (k)$.
- A linear model of system evolution over time.
- A linear relationship between the system state (i.e., pose) and the measurements being made.

- If the assumptions do not hold, the Kalman filter can still be used, but the assurances of its optimality will not be valid - sometimes may lead to very poor results.
- Kalman filter is a mechanism for combining information so that reliable information is more heavily weighted. The Kalman gain is introduced to weigh the relative contribution of new measurements to our prior expectations.
- The Kalman gain (K) varies in proportion to the state covariance matrix and inversely as the measurement covariance matrix.
  - Consistent measurement & large system noise → K↑
  - Noisy measurement & small system noise → K↓

The Kalman filter consists of the following stages at each time step (except the initial step).

- **Prediction**.
  - Prediction of how the plant evolves over time
  - Prediction of how uncertainty evolves overtime

- **Update**
  - Calculate the Kalman gain according to uncertainties in system prediction and measurement
  - Compute the difference between the real measurement and the predicted measurement. This difference is called innovation or residual
  - According to the innovation and Kalman gain, the plant states and uncertainty is updated.

Step 1: Prediction.

Using the plant linear model to compute an estimate of the system state at time *k* based on our knowledge of where the robot was at time *k-1,* the input u_{k-1}, and how the system evolves in time:

$$\boxed{\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}}$$

We can also update our certainty of the state as expressed by the state covariance matrix **P**(.) by 'pushing it forward in time' as:

$$\boxed{P_k^- = AP_{k-1}A^T + Q}$$

Q= model noise covariance

So our knowledge about the system's state gradually decays as time passes (if no new information is available).

Note: The covariance matrix of vector [x_1, . . . , x_n] is given by:

$$\begin{bmatrix} \sum x_1^2 & \sum x_1 x_2 & \cdots & \sum x_1 x_n \\ \sum x_2 x_1 & \sum x_2^2 & & \sum x_2 x_n \\ & \cdots & & \\ \sum x_n x_1 & \sum x_n x_2 & \cdots & \sum x_n^2 \end{bmatrix}$$

Step 2: **Update**:

The Kalman gain can be expressed as: $$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

The revised state estimate is then given by: $$\hat{x}_k = \hat{x}_k^- + K_k \boxed{(z_k - H\hat{x}_k^-)}$$

Residual/Innovation

And the revised state covariance matrix is given by: $$P_k = (I - K_k H) P_k^-$$

where **I** is the identity matrix.

When this process is used in practice, the system is initialized using the initial estimated state, eg. **P**(0) = **Q**(0).

Kalman Filter Summary

Time Update ("Predict")

(1) Project the state ahead

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

(2) Project the error covariance ahead

$$P_k^- = AP_{k-1}A^T + Q$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

(3) Update the error covariance

$$P_k = (I - K_k H)P_k^-$$

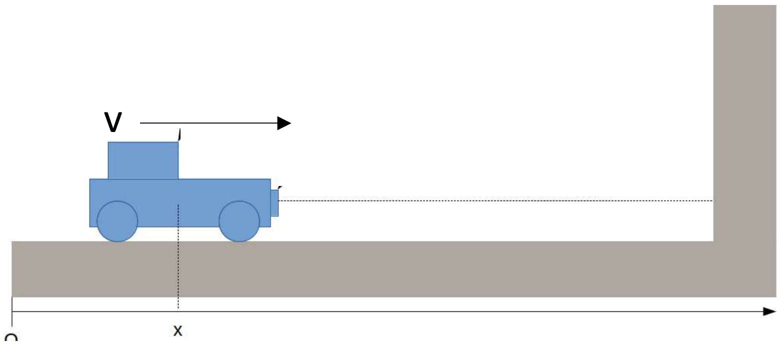
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Kalman filter example

- We'll consider a very simple example for understanding how the filter works.
- Let's consider a robot that moves in a single direction with a constant velocity of v = 1 m/s. Due to the imperfectness of the controller and actuator, the actual velocity of the robot is subjected to noise having the Gaussian distribution with zero mean and the variance of 0.2 m^2/s^2.
- The robot is equipped with a sensor that can measure the position of the robot. However, this sensor is also subjected to noise with zero mean and the variance of 0.1 m^2/s^2. With the sampling rate of 1 Hz, the sensor readings are given in the table below:

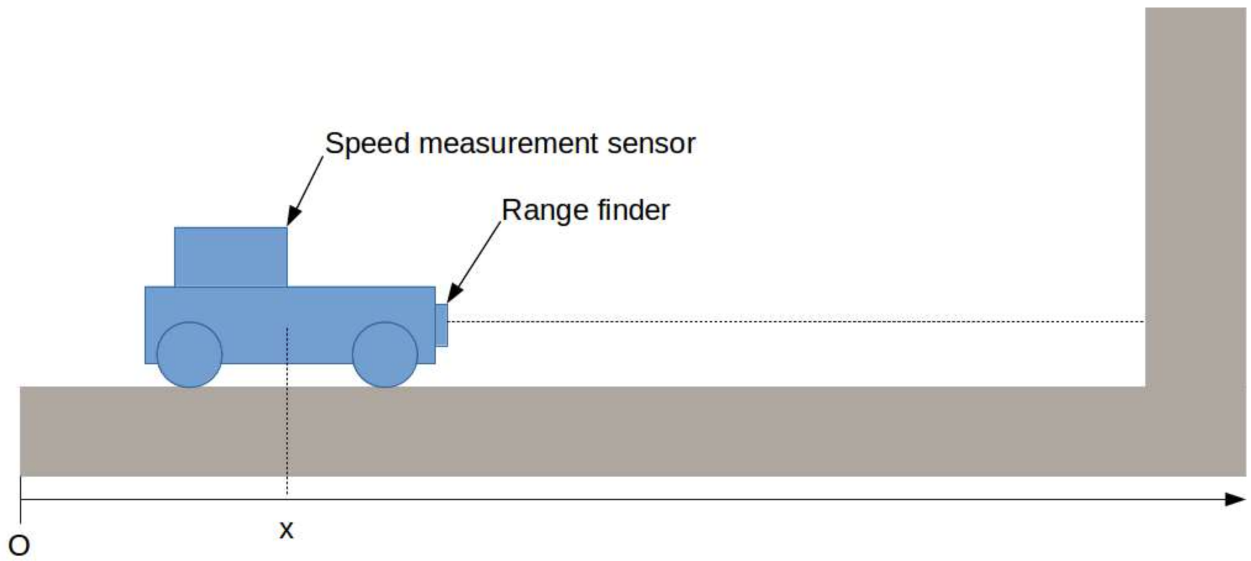
Time (s)	1	2	3	4
Value (m)	1.2	2	3.3	4.1

- Estimate the position of the robot using the Kalman filter given that its initial position is $x_0 = 0$ with the variance $P_0 = 0.1$.

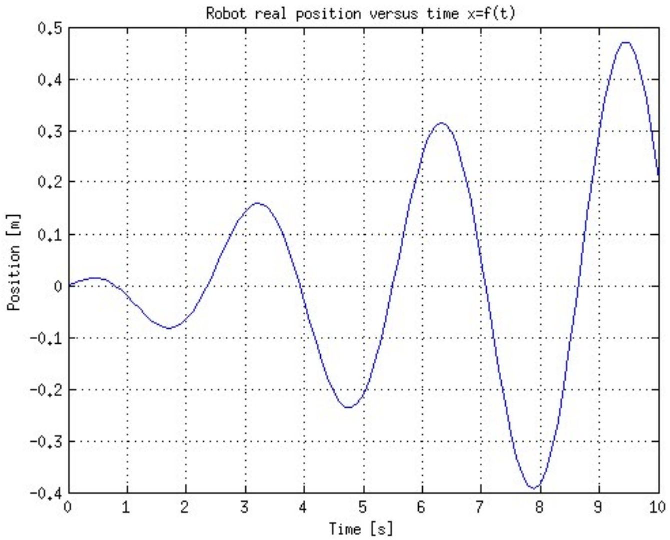


Kalman filter example 2

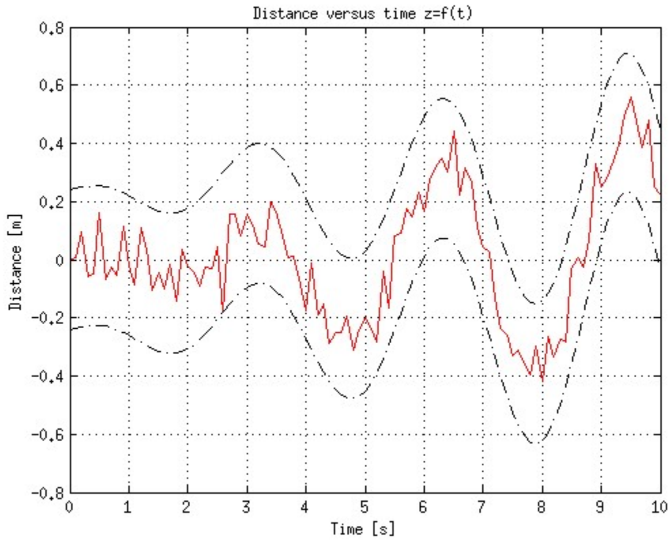
- Again, let's consider a robot that move in a single direction in front of a wall.
- Assume that the robot is equipped with two sensors : a speed measurement sensor and a distance measurement sensor (range finder). We'll consider in the following that both sensors are noisy.

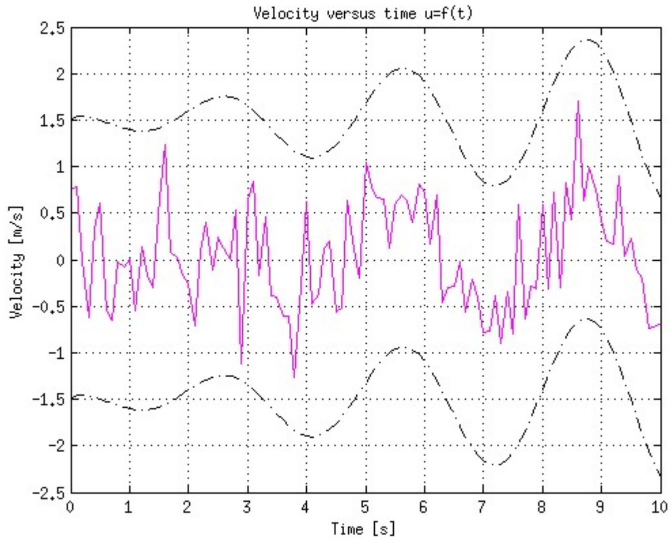


Our goal is to estimate, as accurately as possible, the position x of the robot:

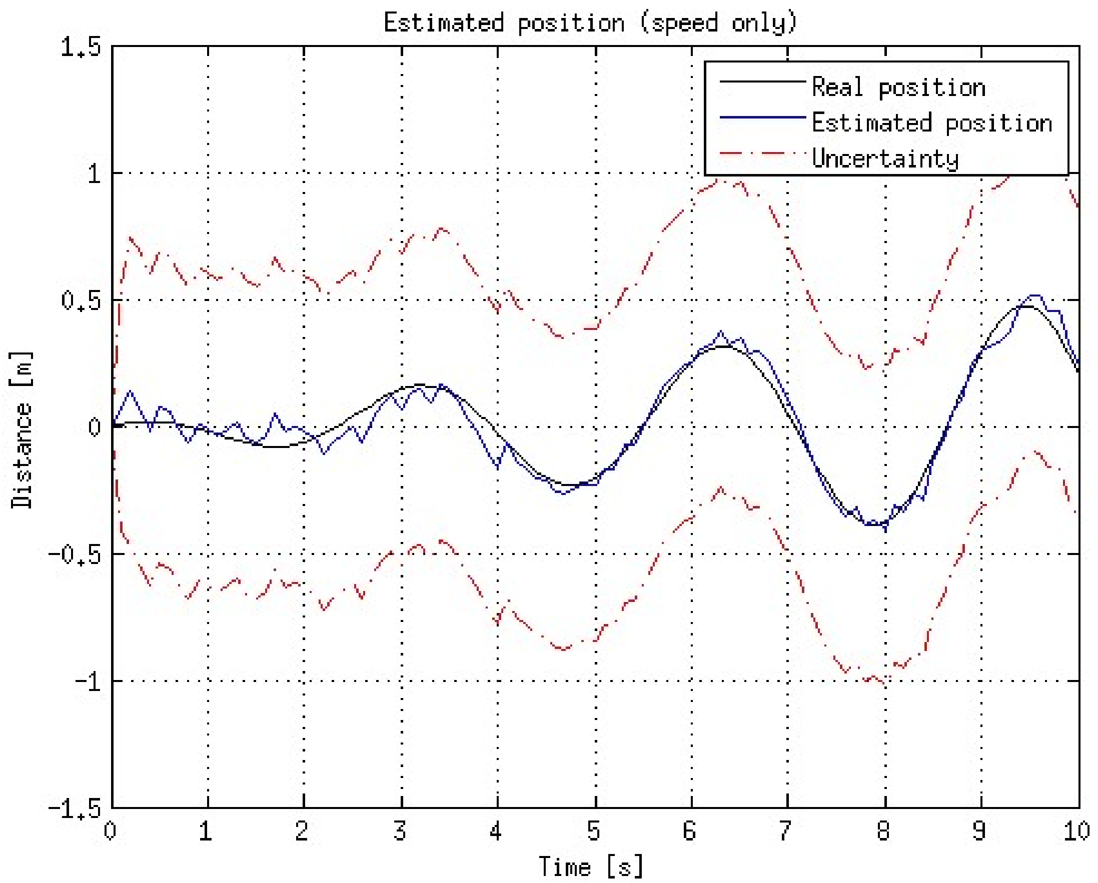


Input of the system are a noisy distance measurement and a noisy velocity measurement:





Result


Estimated position (speed only)
Real position
Estimated position
Uncertainty
Distance [m]
Time [s]
1.5
1
0.5
0
-0.5
-1
-1.5
0
1
2
3
4
5
6
7
8
9
10