KFUPM

COE 584: Robotics

Term 072

Kalman Filter Localization

Ahmad Salam AlRefai, Computer Engineering dept.

Mohammad Shahab, Systems Engineering dept.

19 MAY 2008

Outline

- Introduction
- Robot Localizatoin
- Probablistic Framework
- Kalman Filters
- Kalman Localization
- Landmarks Kalman Localization
- Teams and what they use?

Introduction

- Autonomous Mobile Robots
  - “Rossum’s Universal Robots” play in 1920
- Mobility
- Autonmy
  - Non-autonomous robot
  - Semi-autonoumous robots
  - Fully autonoumous robots
- Applied in many areas like missions to planet, space exploration, (anything difficult for human

Introduction

- Robot Navigation
  - Where am I? robotic localization.
  - Where am going? Goal recognition.
  - How do I get there? Path planning.
- Difficulties
  - Computational power
  - Object and landmark recognition.
  - Obstacle avoidance
  - Multi-sensor Fusion
- Errors and Uncertainty
  - Odometry → wheel slippage
  - Object recognition
  - A notion of uncertainty and belief come from probabilistic point of view.

Robot Localization

- Position tracking: the robot knows the initial location, the goal is to keep track of position while the robot is navigating through the environment.
- Wake up robot (global positioning) robot does not know initial position.
- Kidnapped robot: the robot does exactly know where it is when it is localized but all of a sudden it is transferred or ‘kidnapped to another location.

Robot Localization

- Available Information
  - A-priori Information
    - Maps (geometrical [metric form], topological[features at locations]); SLAM
    - Cause effect relationships (given observations)
  - Navigation information
    - Driving (guidance system)
    - Sensing

Robot Localization

- Relative Position Measurements
  - Odometry (wheel encoder)
  - Inertial Navigation
    - Gyroscopes: measures accelerations in orientation
    - Accelerometers: measures acceleration in x or y axis.
- Absolute Position Measurements
  - Landmark based (active, passive; artificial, natural)
  - Map based: use geometric features(lines → walls, model matching)

Robot Localization

- Multi Sensor Fusion
  - How to combine readings from different sensors.
  - The sensors together can provide more complete picture of a scene at certain time.
  - Can rely on probabilistic approach where notions of uncertainty and confidence are common terminology.

Probabilistic Framework

- Probabilistic Localization
  - Belief, the robot has a belief over where it is: it is the probability density over all locations .

$$Bel(x_k) = P(x_k \mid d_{0 \ldots k})$$

  - Prior versus Posterior
    - $Bel^{-}(x_k)$ the belief after incorporating all information up to step k including latest relative measurement
    - $Bel^{+}(x_k)$ the belief the robot has after it has also included the latest absolute measurement in its belief.

Probabilistic Framework

- Probabilistic Acting and Sensing
  - Acting $P(x_k \mid x_{k-1}, a_{k-1})$
  - Sensing $P(s_k \mid x_k)$
- Localization Formula
  - Initial Belief $Bel^-(x_0)$
  - Updating the Belief

$$Bel^-(x_0) = P\left(x_k \mid z_1, a_1, z_2, a_2, \ldots, z_{k-1}, a_{k-1}\right)$$

$$Bel^+(x_0) = P\left(x_k \mid z_1, a_1, z_2, a_2, \ldots, z_{k-1}, a_{k-1}, z_k\right)$$

Probabilistic Framework

- Incorporating Actions
  - Total Probability Theorem
  - Markov Assumption (future independent of past)

$$Bel^{-}(x_0) = \int P\left(x_k \mid x_{k-1}, a_{k-1}\right) Bel^{+}(x_{k-1}) d_{x-1}$$

  - Bayes Rule
  - Markov Assumption

$$Bel^{+}(x_k) = \frac{P(z_k \mid x_k) Bel^{-}(x_k)}{P(z_k \mid z_1, a_1, \ldots, z_{k-1}, a_{k-1})}$$

$$Bel^{+}(x_k) = \eta_k P(z_k \mid x_k) \int P(x_k \mid x_{x-1}, a_{k-1}) Bel^{+}(x_{k-1}) dx_{k-1}$$

Probabilistic Framework

- Complexity Issues
  - Representation complexity (location space)
  - Modeling Complexity (action and sensor)
- Implementation
  - Discrete Belief
    - Topological Graphs
    - Grid
    - Particle Filters
  - Continous Belief
    - Kalman Filters

Kalman Filters

- Recursive data processing algorithm that estimates the state of a noisy linear dynamic system.
- Richard Kalman discovered the idea in 1960, it is considered one of the greatest discoveries in the history of statistical estimation theory.
- Control and Predictions of dynamic systems are the main areas of Kalman Filter.

Kalman Filters

- KF is a state estimator that works on a prediction-correction basis.
  - State Estimator
    - estimate true state of some system
    - Use sensor readings (observations with noise)
  - Beliefs
  - Prediction correction

Kalman Filters

- Assumptions
  - Linear Dynamic System
    - System Model
    - Measurement Model
    - Markov Process

$$x_k = Ax_{k-1} + w_{k-1}$$

$$z_k = Hx_k + v_k$$

  - Noise Characteristics
    - Independence
    - White noise
      - knowing amount of noise at the current time does not help in predicting what the amount of noise will be at any other time.
    - Zero Mean $w_k \succ N(0, Q_k)$
    - Gaussian $v_k \succ N(0, R_k)$

Kalman Filters

Equations

$$Bel^{-}(x_k) = N(\hat{x}^{-}, P_k^{-}) = N(A\hat{x}^{-}_{k-1}, AP^{+}_{k-1}A^{T} + Q_{k-1})$$

$$Bel^{+}(x_k) = N(\hat{x}^{+}_k, P_k^{k}) = N(\hat{x}^{-}_k + K_k(z_k - H\hat{x}^{-}_k), (I - K_k H)P_k^{-})$$

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

Kalman Filters

- Algorithm
  - Initilization

$$N(\hat{x}_0^-, P_0^-).$$

  - Prediction Equations
  - Correction Equations

$$\hat{x}_k^- = A\hat{x}_{k-1}^+$$
$$P_k^- = AP_{k-1}^+A^T + Q_{k-1}.$$

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

$$K_k = P_k^- H^T \left(H P_k^- H^T + R_k\right)^{-1}.$$

Kalman Filters

- Prediction Equations
  - Uncertainty increase
  - Add system noise
- Correction Equations
  - Measurement Prediction
  - Innovation
  - Kalman Gain
  - Posterior Uncertainty

Extended Kalman Filters

- Kalman → Linear
- Non linear systems → Extended
- By linearization process
- Dynamic

Kalman Filter and Particle Filter

- Recall

$$Bel(x(k)) = \eta P(z(k) \mid x(k)) \sum P(x(k) \mid x(k-1), a(k-1)) Bel(x(k-1))$$

  - With

$$Bel(x) = \{x_i, w_i\}$$

  - Discrete Distribution (*m* particles)
  - Motion Model
  - Measurement Model

Kalman Filter and Particle Filter

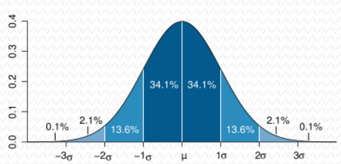
- However, **Kalman Filter (linear)**

$$x(k)^{+} = x(k)^{-} + K(k)\cdot[z(k) - Hx(k)^{-}]$$

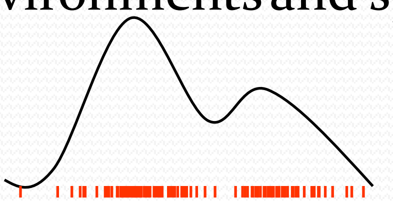
  - Kalman Gain $K(k) = P^{-}(k)H^{T}(HP^{-}(k)H^{T} + R(k))^{-1}$
  - Updated Covariance
  - Motion Model
  - Measurement Model

Kalman Filter and Particle Filter

- **KF:**
  1. Gaussian Assumption (i.i.d.)
  2. Zero Mean with associated Covariance
  3. So, Minimum Computation
  4. Linear Models (Motion & Measurement)
  5. EKF use linearization
  6. Works well in many many applications



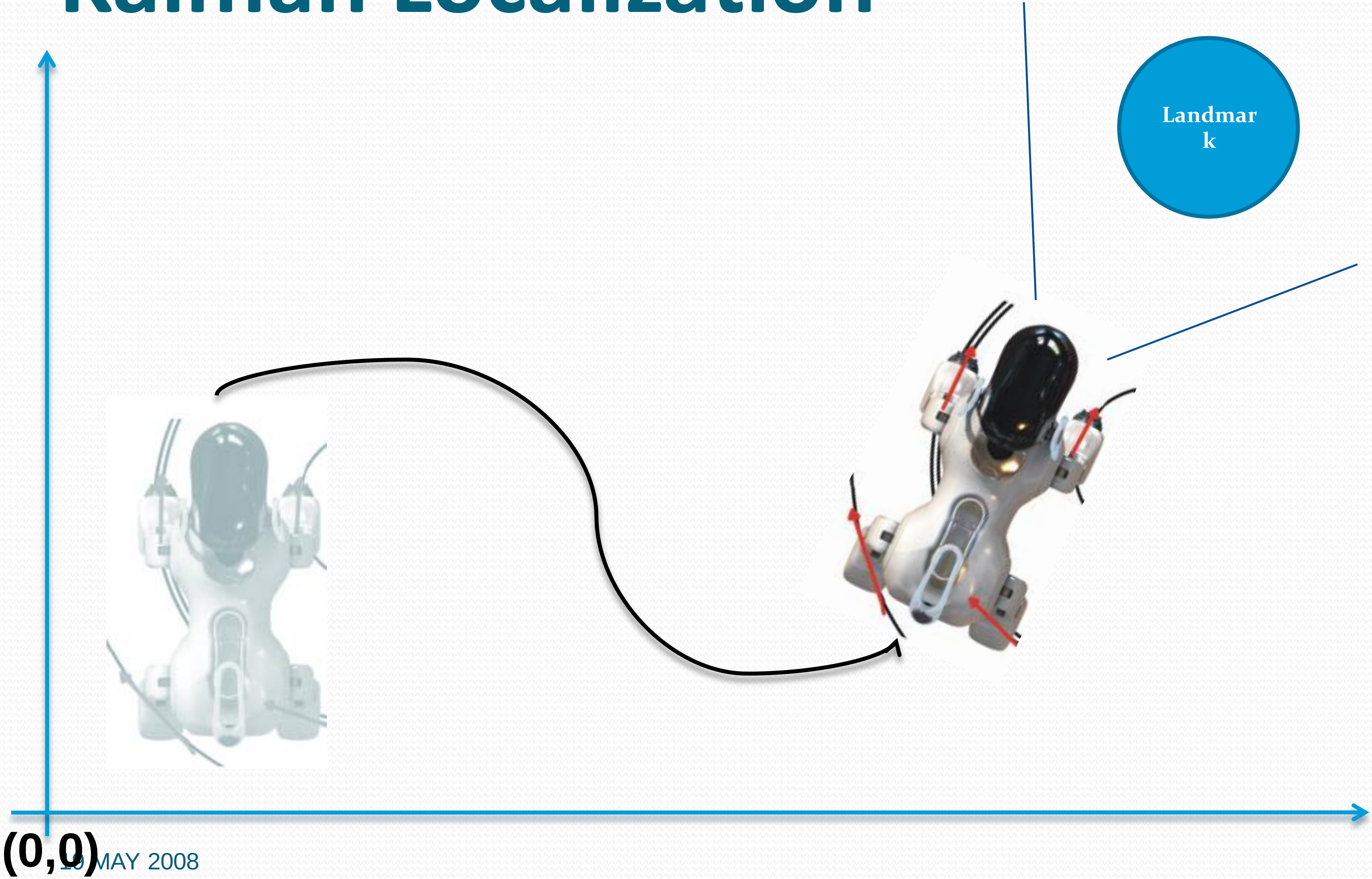
- **PF:**
  1. Act on *m* particles
  2. Distribution defined by the particles
  3. So, High Computation
  4. Non-Gaussian Distribution
  5. Nonlinear Model (Motion or Measurement)
  6. Good for complex environments and systems



Kalman Localization

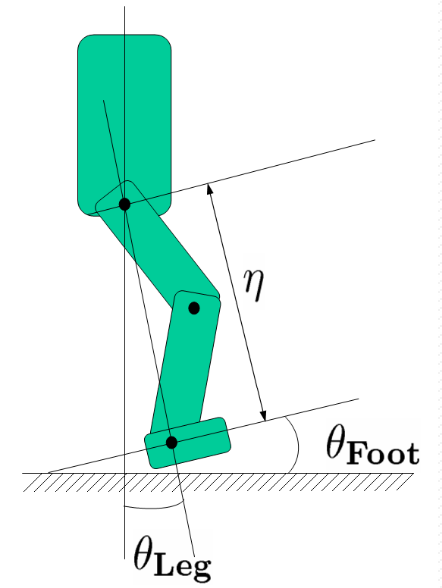
- 3 Problems:
  1. Position Tracking:
     - Given the initial location of the robot, we want the KF to keep track of the position
  2. Kidnapped Robot
     - The robot is fully aware of its location, but all of a sudden is transferred to another location
  3. Global Localization (e.g. RoboCup)
     - the robot does not know its initial position

Kalman Localization



Kalman Localization

- Robot Motion Model (Odometry)



- For a robot,

$$\begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} f_x(x(k-1), u(k-1)) \\ f_y(x(k-1), u(k-1)) \\ f_\theta(x(k-1), u(k-1)) \end{bmatrix}$$

$u(k-1)$
$x(k-1)$ $\mapsto$ Joints (motors) command → **Robot Mechanics** $\{f_x, f_y, f_\theta\}$ → $\begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} + w(k)$

Nonlinear (equations)Model

- *Nonlinearities: e.g. sin(.), cos(.), etc*

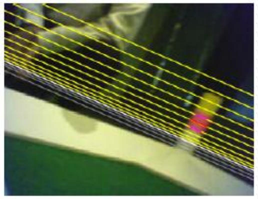
$w(k) \sim N(0, Q(k))$

Kalman Localization

- Robot Measurement (Observation) Model
  - For a **Landmark**

$$z(k) = \begin{bmatrix} z_x(k) \\ z_y(k) \\ z_\theta(k) \end{bmatrix} = \begin{bmatrix} h_x(x(k), l(k)) \\ h_y(x(k), l(k)) \\ h_\theta(x(k), l(k)) \end{bmatrix}$$

  - z_x, z_y, z_θ is location and orientation of a landmark relative to the robot,
  - $l(k)$ is location of landmark in the environment/map with uncertainty



Vision System

$h(.)$

Nonlinear (equations)Model

$$\begin{bmatrix} z_x(k) \\ z_y(k) \\ z_\theta(k) \end{bmatrix} + w(k)$$

$$w(k) \sim N(0, R(k))$$

Kalman Localization

- Recall "**Kalman Equation**":

Location Correction

$$x(k)^{+} = x(k)^{-} + K(k) \cdot [z(k) - h(x(k)^{-})]$$

Location Prediction

$$K(k) = P^{-}(k) H^{T} (H P^{-}(k) H^{T} + R(k))^{-1}$$

Uncertainty

- with

Motion Model Prediction

$$x(k)^{-} = f(x^{+}(k-1), u(k-1)) + w(k-1)$$

$$H = \left.\frac{\partial h}{\partial x}\right|_{x^{-}(k)}$$

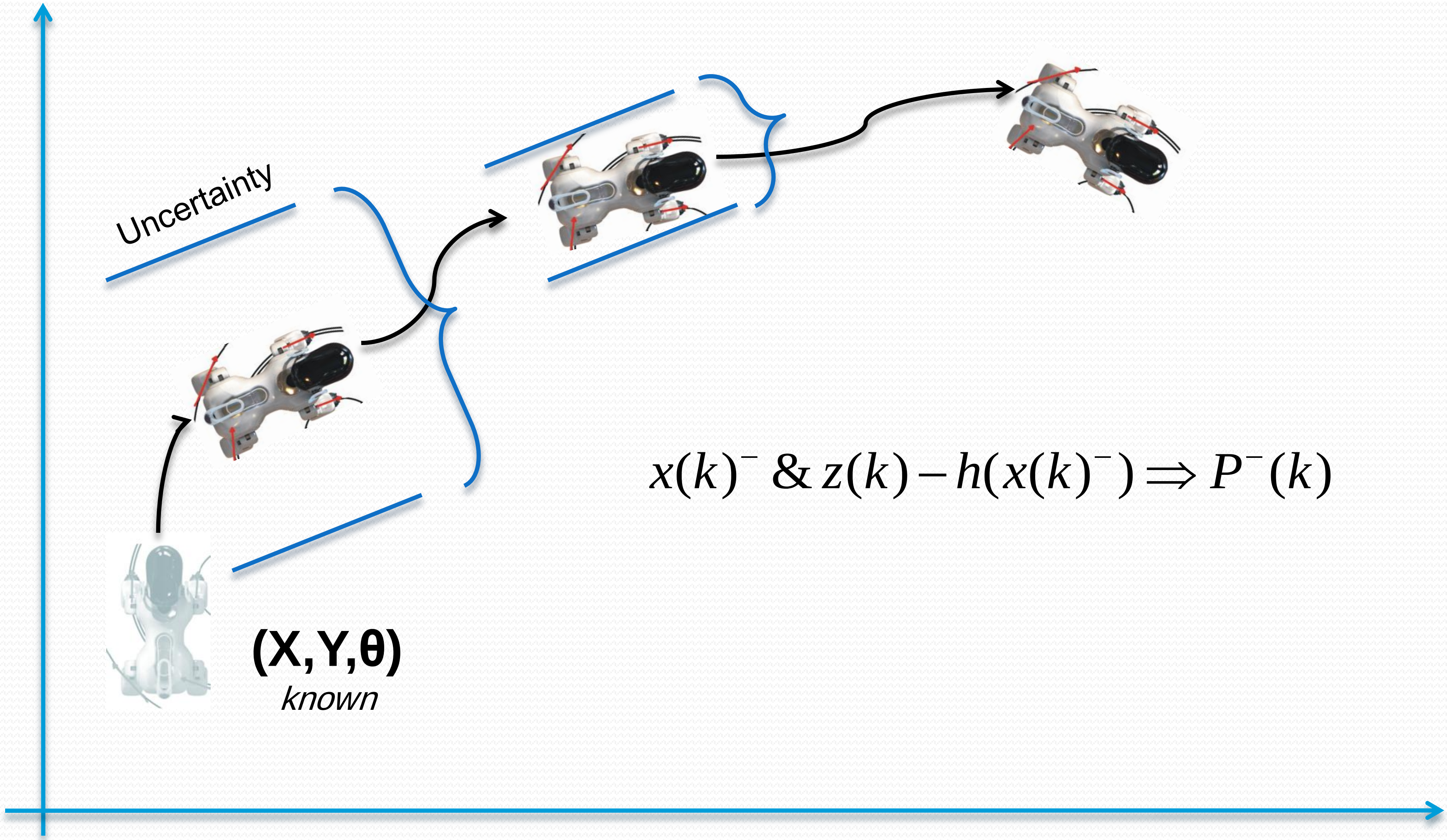
***Example of linearization**: Taylor expansion1st term of Measurement Model*

Kalman Localization

- Position Tracking
  - Initial information are available
  - KF job is to take care of:
    - Motion errors and uncertainties
    - Sensor (Measurement) errors and uncertainties
    - The increase in uncertainty due to prediction and the decrease in uncertainty due to correction keep each other in balance, i.e. convergence

$$x(k)^{+} = x(k)^{-} + K(k) \cdot [z(k) - h(x(k)^{-})]$$

Kalman Localization



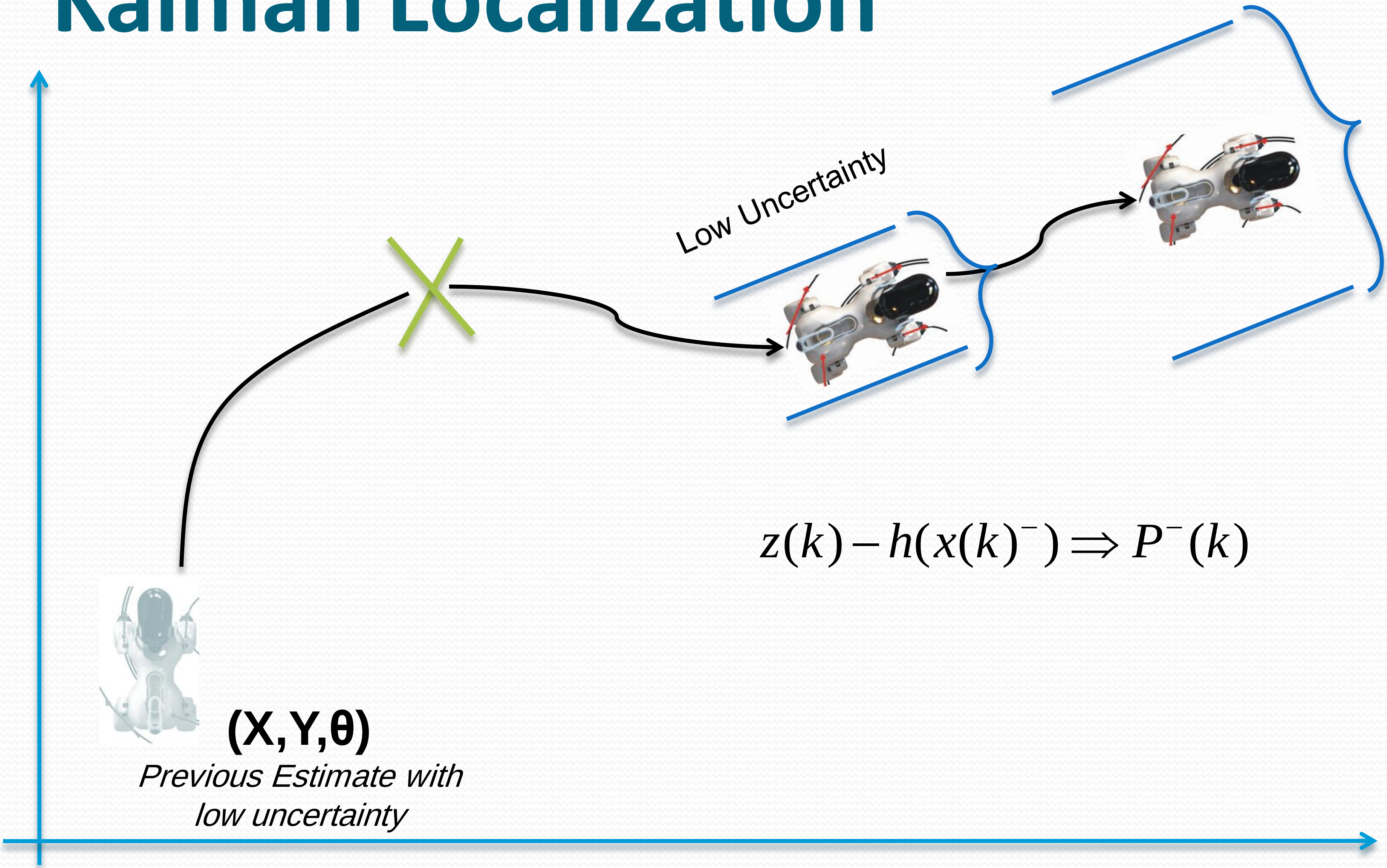
$$x(k)^{-} \,\&\, z(k) - h(x(k)^{-}) \Rightarrow P^{-}(k)$$

Kalman Localization

- Kidnapped Robot
  - If the robot has been kidnapped, it is at different location than where it thinks it is
  - KF do its work:
    - The prediction went wrong
    - When measurement arrive, correction will be proportional to the measurement uncertainty
    - It may take a while before the state estimates have adjusted to the new location
    - If the robot would be able to detect a kidnap, it can take appropriate action it to re-localize itself quicker

$$x(k)^{+} = x(k)^{-} + K(k) \cdot [z(k) - h(x(k)^{-})]$$

Kalman Localization



$$z(k) - h(x(k)^{-}) \Rightarrow P^{-}(k)$$

Global Localization

- Initially: Robot's ***belief*** in the location is **uniform** not Gaussian
- the uncertainty in the prior (initial) state estimate is extremely large
- the **Kalman Gain** becomes the measurement matrix (model):

$$K(k) = H^{-1}$$

- **First** state estimate:

$$x^{+}(k) = H^{-1} z(k)$$

$$x(k)^{+} = x(k)^{-} + K(k) \cdot [z(k) - h(x(k)^{-})]$$

Localization with Landmarks!

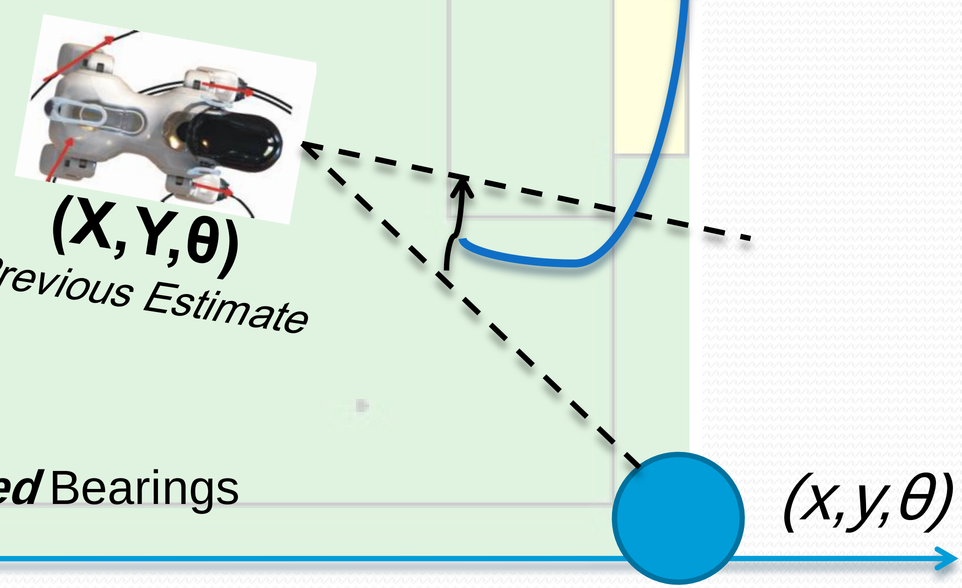
- We have a ***map*** with the location of landmarks in global coordinates
- Measurements are landmarks locations relative to the robot coordinates
- So, 'Correspondence Model' is needed

$$z(k) = \begin{bmatrix} z_x(k) \\ z_y(k) \\ z_\theta(k) \end{bmatrix} = \begin{bmatrix} h_x(x(k), l(k)) \\ h_y(x(k), l(k)) \\ h_\theta(x(k), l(k)) \end{bmatrix}$$

Localization with Landmarks!

- Correspondence Model is like what is done in German Team:
  - Field lines
  - Goals
  - Beacons

$$h_1(x,l) = f(x, y, X, Y, \theta, \cos(.), \sin(.), etc...)$$



$h(x,l)$ ⟶ e.g. ***Expected*** Bearings

RoboCup Teams Localization

Team/Univ	Technique	Remarks
Nubots, University of Newcastle, Australia	EKF	When there is insufficient information available or ambiguous, problems with slow estimate 'drift' with time. Some Adaptive Control is done!
BabyTigers DASH, Osaka City University, Japan	MCL	
Cerberus, Bogazici University, Turkey	S-LOC	Mixture Concepts, explained in 2005 report
Eagle Knights, ITAM, Mexico	Triangulation	+ Correction Algorithms
Team Chaos, University of Murcia, Spain	Fuzzy Logic	*claiming* extended techniques with only natural landmarks
S.P.Q.R., Universit`a di Roma, Italy	MCL + SIR	Two stages: MCL then Sampling/Importance Sampling
UChile1, Universidad de Chile, Chile	MCL + EKF	Tried a faster Adaptive-MCL but accuracy low
sharPKUngfu, Peking University, China	MCL	Collaborative Localization

Questions

- ?