Distributed Intelligent Systems – W4

An Introduction to Localization Methods for Mobile Robots
Outline

• Positioning systems
  – Indoor
  – Outdoor

• Robot localization using proprioceptive sensors without uncertainties
  – Kinematic models and odometry

• Robot localization using proprioceptive sensors with uncertainties
  – Error sources
  – Methods for handling uncertainties

• Robot localization using proprioceptive and exteroceptive sensors
  – Odometry-based and feature-based methods
Robot Localization

- Key task for:
  - Path planning
  - Mapping
  - Referencing
  - Coordination

- Type of localization
  - Absolute coordinates
  - Local coordinates
  - Topological information

N 46° 31’ 13”
E 6 ° 34’ 04”
Positioning Systems
Classification axes

- Indoor vs. outdoor techniques
- Absolute vs. relative positioning systems
- Line-of-sight vs. non-line-of-sight
- Underlying physical principle and channel
- Positioning available on-board vs. off-board
- Scalability in terms of number of nodes
Selected Indoor Positioning Systems

- Overhead cameras and Motion Capture Systems (MCSs)
- Impulse Radio Ultra Wide Band (IR-UWB)
- Infrared (IR) + RF technology
Overhead (Multi-)Camera Systems

- Tracking objects with one (or more) overhead cameras
- Absolute positions, available outside the robot/sensor
- Active, passive, or no markers
- Open-source software available (e.g., SwisTrack, developed at DISAL)
- Major issues: light, calibration
- Essentially 2D

Performance 1 camera system

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>~ 1 cm (2D)</td>
</tr>
<tr>
<td>Update rate</td>
<td>~ 20 Hz</td>
</tr>
<tr>
<td># agents</td>
<td>~ 100</td>
</tr>
<tr>
<td>Area</td>
<td>~ 10 m²</td>
</tr>
</tbody>
</table>
6D Multi-Camera System

- Called also Motion Capture System (MCS)
- 10-50 cameras
- mm accuracy
- 100-500 Hz update
- 2 ms latency
- 6D pose estimation
- 4-5 passive markers per object to be tracked needed
- 200 m^3 arena

ETHZ coordinated ball: throwing
http://www.youtube.com/watch?v=hyGJBV1xnJl
UPenn GRASP: lab
http://www.youtube.com/watch?v=qgqip_0Vjec
IR-UWB System - Principles

• Impulse Radio Ultra-Wide Band
• Based on time-of-flight (TDOA, Time Difference of Arrival)
• UWB tags (emitters, a few cm, low-power) and multiple synchronized receivers
• Emitters can be unsynchronized but then dealing with interferences not trivial (e.g., Ubisense system synchronized)
• Absolute positions available on the receiving system
• Positioning information can be fed back to robots using a standard narrow-band channel
• 6 - 8 GHz central frequency
• Very large bandwidth (>0.5GHz) → high material penetrability
• Fine time resolution → high theoretical ranging accuracy (order of cm)
**IR-UWB System – Performances**

- **Accuracy**: 15 cm (3D)
- **Update rate**: 34 Hz / tag
- **# agents**: ~ 10000
- **Area**: ~ 1000 m²

Ex. State-of-art system (e.g., Ubisense 7000 Series, Compact Tag)

- Degraded accuracy performance if
  - Inter-emitter interferences
  - Non-Line-of-Sight (NLOS) bias
  - Multi-path
Infrared + Radio Technology

- Principle:
  - Belt of IR emitters (LED) and receivers (photodiode)
  - IR LED used as antennas; modulated light (carrier 10.7 MHz), RF chip behind
  - Range: measurement of the Received Signal Strength Intensity (RSSI)
  - Bearing: signal correlation over multiple receivers
  - Measure range & bearing can be coupled with standard RF channel (e.g. 802.11) for heading assessment
  - Can also be used for 20 kbit/s IR com channel
  - Robot ID communicated with the IR channel (ad hoc protocol)

[Pugh et al., *IEEE Trans. on Mechatronics*, 2009]
Infrared + Radio Technology

Performance summary:

- Range: 3.5 m
- Update frequency 25 Hz with 10 neighboring robots (or 250 Hz with 2)
- Accuracy range: <7% (MAX), generally decrease 1/d
- Accuracy bearing: < 9º (RMS)
- LOS method
- Possible extension in 3D, larger range (but more power) and better bearing accuracy with more photodiodes (e.g. Bergbreiter, PhD UCB 2008, dedicated ASIC, up to 15 m, 256 photodiodes, single emitter with conic lense)
Selected Outdoor Positioning Techniques

- GPS
- Differential GPS (dGPS)
Global Positioning System

Note: the first and still most prominent example of GNSS systems (Global Navigation Satellite Systems)
Global Positioning System

- Initially 24 satellites (including three spares), 32 as of December 2012, orbiting the earth every 12 hours at a height of 20,190 km.
- Satellites synchronize their transmission (location + time stamp) so that signals are broadcasted at the same time (ground stations updating + atomic clocks on satellites)
- Location of any GPS receiver is determined through a time of flight measurement (ns accuracy!)
- Real time update of the exact location of the satellites:
  - monitoring the satellites from a number of widely distributed ground stations
  - a master station analyses all the measurements and transmits the actual position to each of the satellites
- Exact measurement of the time of flight
  - the receiver correlates a pseudocode with the same code coming from the satellite
  - the delay time for best correlation represents the time of flight.
  - quartz clock on the GPS receivers are not very precise
  - the range measurement with (at least) four satellites allows to identify the three values (x, y, z) for the position and the clock correction ΔT
- Recent commercial GPS receiver devices allows position accuracies down to a few meters with best satellite visibility conditions.
- 200-300 ms latency, so max 5 Hz GPS updates
dGPS

Position accuracy: typically from a few to a few tens of cm
Robot Localization using On-Board Sensors
Sensors for Localization

- **Proprioceptive sensors:**
  - Epuck:
    - 3D accelerometer
    - Motor step counter
  - Others:
    - Wheel encoder
    - Odometer
    - IMU (inertial measurement unit)

- **Exteroceptive sensors:**
  - Epuck:
    - IR range proximity sensor
    - Camera
  - Others:
    - Laser range finder
    - Ultrasonic range finder
Odometry Definition & Idea

“Using proprioceptive sensory data influenced by the movement of actuators to estimate change in pose over time”

• Start: initial position

• Actuators:
  – Legs
  – Wheels
  – Propeller

• Sensors (proprioceptive):
  – Wheel encoders (DC motors), step counters (stepper motors)
  – Inertial measurement units, accelerometers
  – Nervous systems, neural chains

• Idea: navigating a room with the light turned off
Example from Week 1

- Example: Cataglyphis desert ant
- Excellent study by Prof. R. Wehner (University of Zuerich, Emeritus)
- Individual foraging strategy
- Underlying mechanisms
  - Internal compass (polarization of sun light)
  - Dead-reckoning (path integration on neural chains for leg control; note: in robotics typically using also heading sensors)
  - Local search (around 1-2 m from the nest)
- Extremely accurate navigation: averaged error of a few tens of cm over 500 m path!
More examples

- **Human in the dark**
  - Very **bad** odometry sensors
  - $d_{\text{Odometry}} = O(1/m)$

- **(Nuclear) Submarine**
  - Very **good** odometry sensors
  - $d_{\text{Odometry}} = O(1/10^3 \text{ km})$

- **Navigation system in tunnel uses dead reckoning based on**
  - Last velocity as measured by GPS
  - Car’s odometer, compass

*Picture: Courtesy of US Navy*

*Picture: Courtesy of NavNGo*
Odometry using Wheel Encoders
Optical Encoders

- Measure displacement (or speed) of the wheels
- Principle: mechanical light chopper consisting of photo-barriers (pair of light emitter and optical receiver) + pattern on a disc anchored to the motor shaft
- Quadrature encoder: 90° placement of 2 complete photo-barriers, 4x increase resolution + direction of movement
- Integrate wheel movements to get an estimate of the position -> odometry
- Typical resolutions: 64 - 2048 increments per revolution.

<table>
<thead>
<tr>
<th>State</th>
<th>Ch A</th>
<th>Ch B</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₁</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>S₂</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>S₃</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>S₄</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
Pose (Position and Orientation) of a Differential-Drive Robot

From *Introduction to Autonomous Mobile Robots*, Siegwart R. and Nourbakhsh I. R.
Absolute and Relative Motion of a Differential-Drive Robot

Ex. $\theta = \pi/2$

$\dot{\xi}_R = R(\theta) \dot{\xi}_I$

\[
\begin{bmatrix}
\dot{x}_R \\
\dot{y}_R \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 \\
-1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\dot{x}_I \\
\dot{y}_I \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
\dot{y}_I \\
-\dot{x}_I \\
\dot{\theta}
\end{bmatrix}
\]
Forward Kinematic Model

How does the robot move given the wheel speeds and geometry?

- Assumption: no wheel slip (rolling mode only)!
- In miniature robots no major dynamic effects due to low mass-to-power ratio

\[
\begin{align*}
\mathbf{v}(t) &= \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \theta, \dot{\phi}_1, \dot{\phi}_2) \\
\dot{\phi}_i &= \text{wheel i speed}
\end{align*}
\]
Recap ME/PHY Fundamentals

\[ v = \omega r = \dot{\phi} r \]

- \( v \) = tangential speed
- \( \omega \) = rotational speed
- \( r \) = rotation radius
- \( \phi \) = rotation angle
- \( C \) = rotation center
- \( P \) = peripheral point
- \( P' \) = contact point at time \( t \)

Rolling!
Forward Kinematic Model

Linear speed = average wheel speed 1 and 2:

\[ v = \frac{r \dot{\phi}_1}{2} + \frac{r \dot{\phi}_2}{2} \]

Rotational speed = sum of rotation speeds (wheel 1 forward speed -> \( \omega \) anti-clockwise, wheel 2 forward speed -> \( \omega \) clockwise):

\[ \omega = \frac{r \dot{\phi}_1}{2l} + \frac{-r \dot{\phi}_2}{2l} \]

Idea: linear superposition of individual wheel contributions
Forward Kinematic Model

1. \( \dot{x}_R = v = \frac{r \phi_1}{2} + \frac{r \phi_2}{2} \)

2. \( \dot{y}_R = 0 \)

3. \( \dot{\theta} = \omega = \frac{r \phi_1}{2l} + \frac{-r \phi_2}{2l} \)

4. \( \dot{\xi}_I = R^{-1}(\theta) \dot{\xi}_R \)

\[
\dot{\xi}_I = \begin{bmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\frac{r \phi_1}{2} + \frac{r \phi_2}{2} \\
\frac{r \phi_1}{2l} + \frac{-r \phi_2}{2l}
\end{bmatrix}
\]
Odometry

- Given our absolute pose over time, we can calculate the robot pose after some time $t$ through integration.
- Given the kinematic forward model, and assuming no slip on both wheels

\[
\xi_I(T) = \xi_{I0} + \int_{0}^{T} \dot{\xi}_I dt = \xi_{I0} + \int_{0}^{T} R^{-1}(\theta)\dot{\xi}_R dt
\]

- Given an initial pose $\xi_{I0}$, after time $T$, the pose of the vehicle will be $\xi_I(T)$.
- $\xi_I(T)$ computable with wheel speed 1, wheel speed 2, and parameters $r$ and $l$. 


Localization Uncertainties in Odometry
Deterministic Error Sources

- Limited encoder resolution
- Wheel misalignment and small differences in wheel diameter

➤ Can be fixed by calibration
Non-Deterministic Error Sources

• From Week 3: no deterministic prediction possible → we have to describe them **probabilistically**

• Example: accelerometer-based odometry

MEMS-Based accelerometer (e.g., on e-puck)
Odometry in 1D

$X_0 = 0$

$t=0$
Odometry in 1D

\[ X_1 = X_0 + \int_0^1 \int_0^1 a(t) dt \]
Odometry in 1D

\[ X_1 = X_0 + a_1 \Delta t^2 \]
Odometry in 1D

\[ \hat{X}_1 = X_0 + (a_1 + \ddot{a}_1) \Delta t^2 \]

\[ X_1 = X_0 + a_1 \Delta t^2 \]
Odometry in 1D

\[ \hat{X}_2 = \hat{X}_1 + (a_2 + \ddot{a}_2) \Delta t^2 \]

\[ X_2 = X_1 + a_2 \Delta t^2 \]
Odometry in 1D

\[ \hat{X}_3 = \hat{X}_2 + (a_3 + \ddot{a}_3) \Delta t^2 \]

\[ X_3 = X_2 + a_3 \Delta t^2 \]
Odometry in 1D

\[
\hat{X}_4 = \hat{X}_3 + (a_4 + \ddot{a}_4) \Delta t^2 \\
X_4 = X_3 + a_4 \Delta t^2
\]
Odometry in 1D

\[ \dot{X}_5 = \dot{X}_4 + (a_5 + \ddot{a}_5) \Delta t^2 \]

\[ X_5 = X_4 + a_5 \Delta t^2 \]
1D Odometry: Error Modeling

• Error happens!

• Odometry error is cumulative.  
  → grows without bound

• We need to be aware of it.  
  → We need to model odometry error.  
  → We need to model sensor error.

• Acceleration is random variable $A$ drawn from “mean-free” Gaussian (“Normal”) distribution.  
  → Position $X$ is random variable with Gaussian distribution.
1D Odometry with Gaussian Uncertainty

![Diagram of 1D Odometry with Gaussian Uncertainty]
1D Odometry with Gaussian Uncertainty

\[ t = 1 \]

\[ \mu = 1 \]

\[ \sigma = 0.2 \]
1D Odometry with Gaussian Uncertainty

\[ t = 2 \]

\( \mu = 2 \)
\( \sigma = 0.4 \)
1D Odometry with Gaussian Uncertainty

\[ \mu = 3 \]
\[ \sigma = 0.6 \]
1D Odometry with Gaussian Uncertainty

$\text{estimated position}$

$\text{true position}$

$t = 4$

$\mu = 4$

$\sigma = 0.8$

$X[m]$

$0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10$
1D Odometry with Gaussian Uncertainty

\[
\begin{array}{c}
\text{t=5} \\
\mu = 5 \\
\sigma = 1
\end{array}
\]
Non-Deterministic Uncertainties in Odometry based on Wheel Encoders
Nondeterministic Error Sources

- Variation of the contact point of the wheel
- Unequal floor contact (e.g., wheel slip, nonplanar surface)

- Wheels cannot be assumed to roll perfectly
- Measured encoder values do not perfectly reflect the actual motion
- Pose error is cumulative and incrementally increases
- Probabilistic modeling for assessing quantitatively the error
Odometric Error Types

• Range error: sum of the wheel movements

• Turn error: difference of wheel motion

• Drift error: difference between wheel errors lead to heading error
Pose Variation During $\Delta t$

\[
\Delta s = \frac{\Delta s_r + \Delta s_l}{2}
\]
\[
\Delta x = \Delta s \cos(\theta + \frac{\Delta \theta}{2})
\]
\[
\Delta y = \Delta s \sin(\theta + \frac{\Delta \theta}{2})
\]
\[
\Delta \theta = \frac{\Delta s_r - \Delta s_l}{b}
\]

\[
p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}
\]
\[
p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix}
\]

\[
p' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta + \Delta \theta / 2) \\ \Delta s \sin(\theta + \Delta \theta / 2) \\ \Delta \theta \end{bmatrix}
\]

b = 2l = inter-wheel distance
$\Delta s_r$ = traveled distance right wheel
$\Delta s_l$ = traveled distance left wheel
$\Delta \theta$ = orientation change of the vehicle
Noise modeling

Model error in each dimension with a Gaussian $x \rightarrow \bar{x}, \sigma_x; y \rightarrow \bar{y}, \sigma_y; \theta \rightarrow \bar{\theta}, \sigma_\theta$

\[ \Sigma_p = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{x\theta}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{y\theta}^2 \\ \sigma_{x\theta}^2 & \sigma_{y\theta}^2 & \sigma_{\theta\theta}^2 \end{bmatrix} \]

Assumptions:

- Covariance matrix $\Sigma_p$ at the beginning is known
- Errors of the two individual wheels are independent
- Errors are independent of direction of motion
- Errors are proportional to the distance traveled ($k_r$, $k_l$ model parameters)

\[ \sum_\Delta = \text{cov}(\Delta s_r, \Delta s_l) = \begin{bmatrix} k_r |\Delta s_r| & 0 \\ 0 & k_l |\Delta s_l| \end{bmatrix} = \begin{bmatrix} \sigma_{s_r}^2 & 0 \\ 0 & \sigma_{s_l}^2 \end{bmatrix} \]
Actuator Noise → Pose Noise

• How is the actuator noise (2D) propagated to the pose (3D)?

\[ \Sigma_{\Delta} = \begin{bmatrix} \sigma_{s_r}^2 & 0 \\ 0 & \sigma_{s_l}^2 \end{bmatrix} \quad \sigma_{s_r}^2 \rightarrow \sigma_x^2 \quad \sigma_{s_l}^2 \rightarrow \sigma_\theta^2 \]

\[ \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{x\theta}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{y\theta}^2 \\ \sigma_{x\theta}^2 & \sigma_{y\theta}^2 & \sigma_{\theta\theta}^2 \end{bmatrix} = \Sigma_p \]

• 1D to 1D example \( N(\mu_{s_r}, \sigma_{s_r}) \rightarrow N(\mu_x, \sigma_x) \)

• We need to linearize → Taylor Series

\[ x \approx f(s_r) \bigg|_{s_r=\mu_{s_r}} \approx f(s_r) + \frac{1}{1!} \frac{\partial f}{\partial s_r} (s_r - \mu_{s_r}) + \frac{1}{2!} \frac{\partial^2 f}{\partial s_r^2} (s_r - \mu_{s_r})^2 + \ldots \]
Actuator Noise → Pose Noise

\[ x \approx f_1(s_r) \big|_{s_r=\mu_{s_r}} \approx f_1(s_r) + \frac{\partial f_1}{\partial s_r}(s_r - \mu_{s_r}) \]

\[ x \approx f_1(s_i) \big|_{s_i=\mu_{s_i}} \approx f_1(s_i) + \frac{\partial f_1}{\partial s_l}(s_i - \mu_{s_i}) \]

\[ y \approx f_2(s_r) \big|_{s_r=\mu_{s_r}} \approx f_2(s_r) + \frac{\partial f_2}{\partial s_r}(s_r - \mu_{s_r}) \]

\[ \ldots \]

\[ F_{\Delta rl} = \begin{bmatrix} \frac{\partial f_1}{\partial \Delta s_r} & \frac{\partial f_1}{\partial \Delta s_i} \\ \frac{\partial f_2}{\partial \Delta s_r} & \frac{\partial f_2}{\partial \Delta s_i} \\ \frac{\partial f_3}{\partial \Delta s_r} & \frac{\partial f_3}{\partial \Delta s_i} \end{bmatrix} \]

Jacobian

- General error propagation law

\[ \Sigma_{\Delta rl} = F_{\Delta rl} \Sigma_{\Delta} F_{\Delta rl}^T \]
Actuator Noise $\rightarrow$ Pose Noise

How does the pose covariance $\Sigma_p$ evolve over time?

- **Initial covariance of vehicle at $t=0$:**
  \[
  \Sigma_p^{(t=0)} = \begin{bmatrix}
  \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{x\theta}^2 \\
  \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{y\theta}^2 \\
  \sigma_{\theta x}^2 & \sigma_{\theta y}^2 & \sigma_{\theta\theta}^2
  \end{bmatrix} = \begin{bmatrix}
  0 & 0 & 0 \\
  0 & 0 & 0 \\
  0 & 0 & 0
  \end{bmatrix}
  \]

- **Additional noise at each time step $\Delta t$:**
  \[
  \Sigma_{\Delta rl} = F_{\Delta rl} \Sigma_{\Delta} F_{\Delta rl}^T
  \]

- **Covariance at $t=1\Delta t$:**
  \[
  \Sigma_p^{(t=1\Delta t)} = \Sigma_p^{(t=0)} + \Sigma_{\Delta rl} = \Sigma_{\Delta rl}
  \]

- **Covariance at $t=2\Delta t$:**
  \[
  \Sigma_p^{(t=2\Delta t)} = F_p \Sigma_p^{(t=1\Delta t)} F_p^T + F_{\Delta rl} \Sigma_{\Delta} F_{\Delta rl}^T
  \]

\[
F_p = \begin{bmatrix}
  \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \theta} \\
  \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \theta} \\
  \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \theta}
  \end{bmatrix}
\]
Actuator Noise → Pose Noise

Algorithm

Precompute:

- Determine actuator noise $\Sigma_\Delta$
- Compute mapping actuator-to-pose noise incremental $F_{\Delta rl}$
- Compute mapping actuator-to-pose noise absolute $F_p$

Initialize:

- Initialize $\Sigma_p^{(t=0)} = [0]$

Iterate:

$$\Sigma_p^{(t=(k+1)\Delta t)} = F_p \Sigma_p^{(t=k\Delta t)} F_p^T + F_{\Delta rl} \Sigma_\Delta F_{\Delta rl}^T$$
Classical 2D Representation

Ellipses: typical $3\sigma$ bounds

Courtesy of R. Siegwart and R. Nourbakhsh
Robot Localization based on Proprioceptive and Exteroceptive Sensors
Features

• Odometry based position error grows without bound.
• Use relative measurement to features (“landmarks”) to reduce position uncertainty

• Feature:
  • Uniquely identifiable
  • Position is known
  • We can obtain relative measurements between robot and feature (usually angle or range).

• Examples:
  • Doors, walls, corners, hand rails
  • Buildings, trees, lanes
  • GPS satellites

Courtesy of Albert Huang
Automatic Feature Extraction

- High level features:
  - Doors, persons

- Simple visual features:
  - Edges (Canny Edge Detector 1983)
  - Corner (Harris Corner Detector 1988)

- Simple geometric features:
  - Lines
  - Corners

- “Binary” feature

Courtesy of R. Siegwart and R. Nourbakhsh
Feature-Based Navigation

\[ t = 5 \]

\[ \mu = 5; \sigma = 1 \]
Feature-Based Navigation

$t=5$

$\mu = 5; \sigma = 1$
Feature-Based Navigation

$t = 5$

$\mu = 5; \sigma = 1$

$r = 3.2 m$
Feature-Based Navigation

\[ t = 5 \]

\[ \mu = 5; \sigma = 1 \]

\[ \mu = 5.8; \sigma = 1.2 \]

\[ r = 3.2m \]
Sensor Fusion

• Given:
  - Position estimate \( X \leftarrow N(\mu=5; \sigma=1) \)
  - Range estimate \( R \leftarrow N(\mu=3.2; \sigma=1.2) \)

What is the best estimate AFTER incorporating \( r \) ?

→ **Kalman Filter**

• Requires:
  - Gaussian noise distribution for all measurements
  - Linear motion and measurement model
  - …

Feature-Based Navigation

$\mu = 5; \sigma = 1$

$\mu = 5.8; \sigma = 1.2$

$r = 3.2m$
Feature-Based Navigation

\[ t=5 \]

\[ \mu = 5.5; \sigma = 0.6 \]

\[ \mu = 5; \sigma = 1 \]

\[ \mu = 5.8; \sigma = 1.2 \]

\[ r = 3.2m \]
Feature-Based Navigation

\[ t = 5 \]

\[ \mu = 5.5; \sigma = 0.6 \]
Feature-Based Navigation

\[ t=6 \]

\[ \mu = 6.5; \sigma = 0.8 \]
Feature-Based Navigation

Belief representation through Gaussian distribution

- **Advantages:**
  - Compact (only mean and variance required)
  - Continuous
  - Powerful tools (Kalman Filter)

- **Disadvantages:**
  - Requires Gaussian noise assumption
  - Uni-modal
  - Cannot represent ignorance ("kidnapped robot problem")
Feature-Based Navigation

![Diagram showing estimated and true positions at t=5]
Feature-Based Navigation

![Graph showing estimated position, true position, and range-based position with time t=5 and range r=3.2m.](image-url)
Feature-Based Navigation

![Diagram showing estimated position, true position, and range-based position at t=5 with r = 3.2m]
Feature-Based Navigation

![Graph showing estimated position, true position, and range-based position with t=5 and r=4.2m and r=3.2m]
Feature-Based Navigation
Feature-Based Navigation

![Graph with estimated and true positions at t=5]
Feature-Based Navigation

\[ t=0 \]
Feature-Based Navigation

\[
\forall \text{ particles } s_i : s_i^{t=1} = s_i^{t=0} + a + \tilde{a}(N(\mu = 0, \sigma = 0.2))
\]
Feature-Based Navigation

∀ particles $s_i$: $s_i^{t=2} = s_i^{t=1} + a + \tilde{a}(N(\mu = 0, \sigma = 0.2))$

\[ \begin{align*}
\mu &= 2 \\
\sigma &= 0.4
\end{align*} \]
Feature-Based Navigation

\[ \forall \text{ particles } s_i : s_i^{t=3} = s_i^{t=2} + a + \tilde{a}(N(\mu = 0, \sigma = 0.2)) \]

\[ \mu = 3 \]
\[ \sigma = 0.6 \]

estimated position
true position

t = 3
Feature-Based Navigation

∀ particles $s_i$: $s_i^{t=4} = s_i^{t=3} + a + \tilde{a}(N(\mu = 0, \sigma = 0.2))$

$\mu = 4$

$\sigma = 0.8$

$t = 4$
Feature-Based Navigation

∀ particles $s_i$: $s_i^{t=5} = s_i^{t=4} + a + \dot{a}(N(\mu = 0, \sigma = 0.2))$

$\mu = 5$

$\sigma = 1$

t = 5

estimated position
true position
Feature-Based Navigation

∀ particles $s_i$: $s_{i=5}^t = s_{i=4}^t + a + \tilde{a}(\mathcal{N}(\mu = 0, \sigma = 0.2))$
Feature-Based Navigation

∀ particles $s_i$: $s_i^{t=5} = s_i^{t=4} + a + \tilde{a}(N(\mu = 0, \sigma = 0.2))$

Estimated position
True position
Range-based position

$t=5$

$r = 3.2m$
Feature-Based Navigation

∀ particles $s_i$: $s_i^{t=5} = s_i^{t=4} + a + \tilde{a}(\mathcal{N}(\mu = 0, \sigma = 0.2))$
Feature-Based Navigation

∀ particles $s_i$: $s_i^{t=5} = s_i^{t=4} + a + \bar{a}(\mathcal{N}(\mu = 0, \sigma = 0.2))$

$\text{estimated position}$
$\text{true position}$
$\text{range-based position}$

$t = 5$

$r = 4.2m$
$r = 3.2m$
Feature-Based Navigation

∀ particles $s_i$: $s_i^{t=5} = s_i^{t=4} + a + \bar{a}(N(\mu = 0, \sigma = 0.2))$
Feature-Based Navigation

\[ \forall \text{particles } s_i: s_i^{t=5} = s_i^{t=4} + a + \tilde{a}(N(\mu = 0, \sigma = 0.2)) \]
Feature-Based Navigation

$X[m]$

$t=5$

estimated position
true position
range-based position

$r = 4.2m$

$r = 3.2m$
Feature-Based Navigation

![Graph showing estimated, true, and range-based positions at t=5 with distances r=4.2m and r=3.2m.](image-url)
Feature-Based Navigation

Belief representation through particle distribution

• Advantages:
  • Can model arbitrary beliefs
  • No assumptions on noise characteristic

• Disadvantages:
  • No unique solution
  • Not continuous
  • Computationally expensive
  • Tuning required
Feature-Based Navigation
Feature-Based Navigation
Conclusion
Take Home Messages

• There are several localization techniques for indoor and outdoor systems
• Each of the localization methods/positioning system has advantage and drawbacks
• Odometry is an absolute positioning method using only proprioceptive sensors but affected by a cumulative error
• Localization error can be modeled and estimated
• The error propagation methods/filtering techniques are tools applicable to a large variety of noisy problems, not just localization and navigation
• Feature-based navigation is a way to compensate odometry limitations
Additional Literature – Week 4

Books

