Introduction to Multi-Agent Programming

6. Cooperative Sensing

Modeling Sensors, Kalman Filter, Markov Localization, Potential Fields

Alexander Kleiner, Bernhard Nebel
Contents

• Introduction
• Modeling and Processing of sensor data
• Kalman Filter
  – Case-Study: Opponent modeling
• Markov Localization as observation filter
  – Case-Study: Cooperative ball recognition
• Potential Fields
  – Case Study: Predicate extraction
• Summary
Introduction

• Cooperative sensing and world modeling follows the goal of deliberate decision making
  – Which is in contrast to reactive acting
• Agents perceive their environment by sensors
  – However, sensing can either be inaccurate or ambiguous
  – Sensing requires the process of world modeling for meaningful and robust decision making
  – Probabilistic models are the first choice for this task
• World models can be used to extract abstract predicates of the world
  – For example, “objectInOpponentGoal(ball)”
Processing Sensor data I
Inertial Measurement Unit (IMU) & Wheel Odometry

- A closed system for detecting orientation and motion of a vehicle or human
- Typically consists of 3 accelerometers, 3 gyroscopes, and 3 magnetometers
- Data rate @100 Hz
- Gyro reliable only within some time period (temperature drift)
- Magnetometer data can locally be wrong (magnetic perturbation)
- Therefore, gyro, accelerometer, and magnetometer data is fused by a Kalman Filter onboard the IMU sensor
- For the estimation of robot poses \((x,y,\theta)\) also wheel odometry, a hardware that counts the number of wheel revolutions per second, is required
Laser Range Finders (LRFs)

- Found on many robots
- Highly accurate, high data rate
- Measures distances and angles to surrounding objects
- Returns distances $d_i$ and angles $\alpha_i$, with $i \in [0...\text{FOV/resolution}]$

<table>
<thead>
<tr>
<th></th>
<th>Sick LMS200</th>
<th>Sick S300</th>
<th>Hokuyo URG-04LX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>4500 g</td>
<td>1200 g</td>
<td>160 g</td>
</tr>
<tr>
<td>Volume</td>
<td>$\approx 20 \text{ cm}^3$</td>
<td>$\approx 15 \text{ cm}^3$</td>
<td>$\approx 5 \text{ cm}^3$</td>
</tr>
<tr>
<td>FOV</td>
<td>180°</td>
<td>270°</td>
<td>240°</td>
</tr>
<tr>
<td>Max. Range</td>
<td>80 m</td>
<td>30 m</td>
<td>4 m</td>
</tr>
<tr>
<td>Max. Ang. Res.</td>
<td>0.25°</td>
<td>0.5°</td>
<td>0.36°</td>
</tr>
<tr>
<td>Accuracy</td>
<td>$\pm 15 \text{ mm}$</td>
<td>$\pm 30 \text{ mm}$</td>
<td>$\pm 10 \text{ mm}$</td>
</tr>
<tr>
<td>Scans per second</td>
<td>30</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Interface</td>
<td>RS-232/RS-422</td>
<td>RS-232/RS-422</td>
<td>RS-232/USB</td>
</tr>
</tbody>
</table>
Processing Sensor data III
Color Cameras

- Sensor that generates **color images**, e.g. with 640x480 pixel resolution @30hz

- Can be used for object detection, e.g. soccer ball
  - Color thresholding, e.g. separation of ball colors from background
  - Determination of relative object location by camera calibration or interpolation
Modeling Sensor noise

- Sensor data is typically noisy, e.g., the distance measurement of a LRF at one meter can be 1m ± 1cm
- Sensor noise is typically modeled by a normal distribution
- Fully described by mean $\mu$ and variance $\sigma^2$

\[
p(x) = \frac{1}{\sqrt{2\pi\sigma^2_x}} e^{-\frac{1}{2} \left( \frac{x - \mu_x}{\sigma_x} \right)^2}
\]

Notation: \(x \sim N(\mu_x, \sigma^2_x)\)

\[
p(x) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma_x)}} e^{-\frac{1}{2} (x - \mu_x)^T \Sigma_x^{-1} (x - \mu_x)}
\]

Notation: \(x \sim N(\mu_x, \Sigma_x)\)
Transformation of density functions I
Linear Transformation

- When processing data from multiple sensors, all observations have to be transformed into a single coordinate system.
- For example, distance and angle measurements of a LRF have to be integrated into a Cartesian coordinate frame.
- Linear transformations can be represented by $F(u) = Au + b$, where $A$ is a $n \times m$ matrix and $b$ a $n$-dimensional vector.

*Given* $\mu_u \Sigma_u$  *Wanted*: new mean and variance $\mu_u \Sigma_u$

*Mean $\mu_x$ and covariance $\Sigma_x$ can be computed by:*

\[
\begin{align*}
\mu_x &= E(x) \\
&= E(Au + b) \\
&= AE(u) + b \\
&= A\mu_u + b \\
\Sigma_x &= E((x - E(x))(x - E(x))^T) \\
&= E((Au + b - AE(u) - b)(Au + b - AE(u) - b)^T) \\
&= E((A(u - E(u)))(u - E(u))^T) \\
&= AE((u - E(u))(u - E(u))^T)A^T \\
&= A\Sigma_u A^T
\end{align*}
\]
Linearization is necessary in order to yield a normal distribution—Approximation by Taylor series while skipping higher order terms:

\[ F(u) \approx F(\hat{u}) + \nabla F(\hat{u})(u - \hat{u}) \]

where \( \nabla F(\hat{u}) = \frac{\partial F}{\partial u}(\hat{u}) \) is a \( n \times m \) matrix (also known as Jacobi- Matrix) with partial derivatives of \( F \) at \( \hat{u} \).

Notation according to the linear case:

\[
\begin{align*}
A & = \nabla F(\mu_u) \\
b & = F(\mu_u) - \nabla F(\mu_u)\mu_u
\end{align*}
\]

Mean \( \mu_x \) and covariance \( \Sigma_x \) can then be computed by:

\[
\begin{align*}
\mu_x & = A\mu_u + b \\
& = \nabla F(\mu_u)\mu_u + F(\mu_u) - \nabla F(\mu_u)\mu_u \\
& = F(\mu_u) \\
\Sigma_x & = A\Sigma_u A^T \\
& = \nabla F(\mu_u)\Sigma_u \nabla F(\mu_u)^T
\end{align*}
\]
Transformation of density functions III
Example: LRF Measurement Transformation

- We assume a normal distributed error of distance measurement \( d \) and angle measurement \( \alpha \):

\[
d \sim N(\mu_d, \sigma_d^2) \quad \alpha \sim N(\mu_{\alpha}, \sigma_{\alpha}^2)
\]

- **Transformation function** \( F \):

\[
F(\begin{pmatrix} d \\ \alpha \end{pmatrix}) = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} d \cos \alpha \\ d \sin \alpha \end{pmatrix}
\]

\[
\Sigma_{xy} = \nabla F_{d\alpha} \Sigma_{d\alpha} \nabla F_{d\alpha}^T
\]

\[
\nabla F_{d\alpha} = \begin{pmatrix} \cos \alpha & -d \sin \alpha \\ \sin \alpha & d \cos \alpha \end{pmatrix}
\]

\[
\Sigma_{d\alpha} = \begin{pmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_{\alpha}^2 \end{pmatrix}
\]
Transformation of density functions IV
Example: LRF Measurement Transformation

• Assume \( d=3000\text{mm}, \alpha=30^\circ, \sigma_d=100\text{mm}, \sigma_\alpha=5.7^\circ \)

\[
\begin{pmatrix}
\mu_x \\
\mu_y
\end{pmatrix} = \begin{pmatrix}
d \cos \alpha \\
d \sin \alpha
\end{pmatrix} = \begin{pmatrix}
2598 \\
1500
\end{pmatrix}
\]

\[
\nabla F_{d\alpha} = \begin{pmatrix}
\cos \alpha & -d \sin \alpha \\
\sin \alpha & d \cos \alpha
\end{pmatrix} = \begin{pmatrix}
0.866 & -1500 \\
0.500 & 2598
\end{pmatrix}
\]

\[
\Sigma_{d\alpha} = \begin{pmatrix}
\sigma_d^2 & 0 \\
0 & \sigma_\alpha^2
\end{pmatrix} = \begin{pmatrix}
10000 & 0 \\
0 & 0.01
\end{pmatrix}
\]

\[
\Sigma_{xy} = \nabla F_{d\alpha} \Sigma_{d\alpha} \nabla F_{d\alpha}^T = \begin{pmatrix}
30000 & -34640 \\
-34640 & 70000
\end{pmatrix}
\]
Kalman Filter I
Introduction

• An optimal recursive data processing algorithm
  – optimal since it processes all data regardless of precision
  – recursive since the filter computes the next estimate based on the last estimate and the latest measurement

• Fusion of two independent measurements of the same concept

• Each measurement has a confidence expressed by the variance of the Gaussian

• Example: two people on a boat estimate their 1D location. The 2nd person ($z_2$) is more skilled than the 1st one ($z_1$)

Images courtesy of Maybeck
Kalman Filter II
Update Formula

one-dimensional:

\[ l_1 \sim N(\mu_1, \sigma_1^2) \]
\[ l_2 \sim N(\mu_2, \sigma_2^2) \]

\[ l = \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} \left( \frac{1}{\sigma_1^2} l_1 + \frac{1}{\sigma_2^2} l_2 \right) \]
\[ \frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \]

n-dimensional:

\[ m_1 \sim N(\mu_1, \Sigma_1) \]
\[ m_2 \sim N(\mu_2, \Sigma_2) \]

\[ m = \left( \Sigma_1^{-1} + \Sigma_2^{-1} \right)^{-1} \left( \Sigma_1^{-1} m_1 + \Sigma_2^{-1} m_2 \right) \]
\[ \Sigma = \left( \Sigma_1^{-1} + \Sigma_2^{-1} \right)^{-1} \]
1st step: players estimate their own position and orientation on the field by matching scans with the field model.

2nd step: Extraction of other players by discarding scan points belonging to field walls and clustering the remaining ones. For each cluster the center of gravity is assumed to correspond to the center of another robot.
Case-Study: Cooperative opponent sensing on a soccer field II

3rd step: Communication of position, heading and velocity of each detected object and own pose to a central multi-sensor integration module

4th step: Assignment of team player IDs to objects → Detection of opponents (red)
Integration of multiple measurements I

State representation:

Each observation is modeled by a random variable: \( \mathbf{x}_s = (x_s, y_s, \theta_s, v_s, \omega_s)^T \)

With mean \( \hat{x}_s \) and covariance \( \Sigma_s \), where \((x_s, y_s)\) is the position, \( \theta_s \) the orientation, and \(v_s, \omega_s\) are the translational and rotational velocities* of the object.

Modeling of the covariance: \( \Sigma_s = \text{diag}(\sigma_{x_s}^2, \sigma_{y_s}^2, \sigma_{\theta_s}^2, \sigma_{v_s}^2, \sigma_{\omega_s}^2) \)

where \( \sigma_{x_s}, \sigma_{y_s}, \sigma_{\theta_s}, \sigma_{v_s}, \sigma_{\omega_s} \) are constant standard deviations determined experimentally

State projection:

\[
\begin{align*}
\hat{\mathbf{x}}_r & \leftarrow F_s(\hat{\mathbf{x}}_r, t) = \\
&= \begin{pmatrix}
\hat{x}_r + \cos(\hat{\theta}_r) \hat{v}_r t \\
\hat{y}_r + \sin(\hat{\theta}_r) \hat{v}_r t \\
\hat{\theta}_r + \hat{\omega}_r t \\
\hat{v}_r \\
\hat{\omega}_r 
\end{pmatrix} \\
\Sigma_r & \leftarrow \nabla F_s \Sigma_r \nabla F_s^T + \Sigma_a(t)
\end{align*}
\]

\( \Sigma_a(t) = \text{diag}(\sigma_{x_a}^2 t, \sigma_{y_a}^2 t, \sigma_{\theta_a}^2 t, \sigma_{v_a}^2 t, \sigma_{\omega_a}^2 t) \)

*Note velocities are determined by differencing the last 10 pose estimates
Integration of multiple measurements II

State update:

(a) Observation of a new object:
\[ \hat{X}_r = \hat{X}_s, \quad \sum_r = \sum_s \]

(b) Observation of a known object:
\[ \hat{x}_r \leftarrow (\sum_r^{-1} + \sum_s^{-1})^{-1}(\sum_r^{-1}\hat{x}_r + \sum_s^{-1}\hat{x}_s) \]
\[ \sum_r \leftarrow (\sum_r^{-1} + \sum_s^{-1})^{-1} \]

Data association problem, i.e. how to associate observations to known objects?

Greedy method:
Search the global world model for the track whose predicted mean is closest to the observation. Assign observation if distance is beyond a certain threshold. → Can be sub-optimal!

Better approach: geometric assignment
Go over all possible sets of assignment pairs \((s_i, r_i)\)
Find assignment that minimizes \(\sum_{i=1}^{n} \text{dist}(s_i, r_i)^2\)
Single Object Tracking from Noisy Data
Example: Ball Tracking

- For example, **global ball position estimation**: stereo vision with robot groups
- Detection of the ball by **vision**, e.g. detecting the ball by color
  - Estimation of the **angle** is quite accurate, however, **distance** is not
  - Kalman Filter **integration** yields an error ellipse with respect to these confidences
  - Fusion of two estimates respects error ellipse: effect of “**triangulation**”
- **Prediction step** (predict next location where ball will be observed):
  - Project ball position into the future using a constant negative ball acceleration (due to friction)
  - Consider a certain projection error
- **Update step** (when new observation is made):
  - Integrate new measurement (using a weighted average on the error)
    - distance error grows with distance
    - angular error is small and constant
Single Object Tracking from Noisy Data
Example: Ball Tracking

Effect of triangulation

Kalman filtering compared to simple averaging: highly confident estimates are more strongly weighted

Kalman filtering

Simple averaging
The Importance of Global Ball Estimation

Minho (Portugal) shoots at our goal from the other side of the field. Our goalie gets this information early on from his team mates and can easily defend.
Single Object Tracking from Noisy Data
Problem of false positives (ghost balls)

Player 2 is hallucinating
Markov Localization as Observation Filter
Introduction

• The Kalman-Filter expects that measurements originate from the same objects
  – However, color thresholding on a soccer field might confuse for example “red t-shirts” with the ball
  – Consequently, Kalman filtering yields poor results

• Markov localization: Simultaneous tracking of multiple hypotheses

• Idea: To filter-out false positives with a probability grid
Probabilistic Localization

Courtesy of Wolfram Burgard
Simple Example of State Estimation

- Suppose a robot obtains measurement $z$
- What is $P(\text{open}|z)$?
Causal vs. Diagnostic Reasoning

- $P(open|z)$ is diagnostic.
- $P(z|open)$ is causal.
- Often causal knowledge is easier to obtain.
- Bayes rule allows us to use causal knowledge:

$$P(open|z) = \frac{P(z|open)P(open)}{P(z)}$$
Example

- $P(z|\text{open}) = 0.6$  \quad $P(z|\neg\text{open}) = 0.3$
- $P(\text{open}) = P(\neg\text{open}) = 0.5$

\[
P(\text{open} | z) = \frac{P(z | \text{open})P(\text{open})}{P(z | \text{open})p(\text{open}) + P(z | \neg\text{open})p(\neg\text{open})}
\]

\[
P(\text{open} | z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67
\]

Marginalization: compute marginal probability $p(z)$

$z$ raises the probability that the door is open.

Courtesy of Wolfram Burgard
Markov Localization as Observation Filter
Prediction & Update I

• **Discretization** of the soccer field into a two-dimensional grid
  – Each cell of the grid reflects the probability \( p(z) \) that the ball is at
    the cell location \( \text{loc}(z) = (x,y) \)

• **Uniform initialization** of the grid before any observation is
  processed, i.e. \( p(z) = 1/\# \text{cells} \)

• **Prediction step:**
  – Simple model of ball motion
    \[
    p(z) \leftarrow \sum p(z | z') p(z')
    \]
    here
    \( p(z | z') \) denotes the probability that the ball moved to cell \( z \) given
    it was at \( z' \).
  – When assuming that all kind of motion directions are equally
    possible, and velocities are **normally distributed** with zero mean
    and covariance \( \sigma_v^2 \), \( p(z | z') \) can be modeled by a time-depended
    Gaussian around \( z' \):
    \[
    p(z | z') \sim N(z', \text{diag}(\sigma_v^2 t, \sigma_v^2 t))
    \]
Markov Localization as Observation Filter
Prediction & Update II

• Update step:
  – Fusion of new ball observation $z_b$ into the grid according to Bayes’ law:
  \[ p(z) \leftarrow \frac{p(z_b | z)p(z)}{\sum_{z'} p(z_b | z')p(z')} \]

  – The sensor model $p(z_b | z)$ determines the likelihood observing $z_b$ given the ball is at position $z$.
    • e.g. less confidence as more far away the ball

• Finally, the Markov grid can be used for outlier rejection
  – Kalman filtering is only applied at the highest peak of the distribution
  – If another peak becomes more likely, the Kalman filter is re-initialized accordingly
Phantom Balls: Development of Probability Distribution I

Consider area with highest peak as possible ball area and use KF there
At RoboCup 2000, 938 out of 118388 (0.8%) ball observations were ignored because of the Markov localization filter.
Demo Webplayer

See www.cs-freiburg.de
Potential Fields
Introduction

- Originally introduced for robot path planning
  - Robot is considered as particle within a force field, the potential field
  - Potential field is generated by overlaying repulsive potentials (e.g. obstacles) and attractive potentials (e.g. goals)
  - The motion of the robot is determined by negating the field’s gradient, leading to the potential minimum
  - Repulsive and attractive potentials are computed separately

- Can also be used for strategic decision making (e.g. CS-Freiburg)
Potential Fields
Grid representation

- **Discretization** of the configuration space into equally sized cells
- **Grid representation** GC is defined for every $q=(x,y)$ as follows:

$$GC = \{ q \in C \mid q = (i\delta_x, j\delta_y), \ i = 1, \ldots, N, \ j = 1, \ldots, M \}$$

where $\delta_x, \delta_y$ are the step sizes in X and Y direction, and $N, M$ are the number of cells along the axes, respectively.
Potential Fields

Potentials

- Potentials are differentiable functions of the type $U : C_{free} \rightarrow \mathbb{R}$, where $C_{free}$ is the set of possible robot configurations.
- Typically, high values indicate obstacles and low values goals.
- Given differentiable potentials, one can compute the force at each configuration $q$ by:

$$\vec{F}(q) = -\vec{\nabla}U(q)$$

- For example, given a 2D work space, force $F(q)$ can be computed from $U(q)$ by:

$$\vec{F}(q) = - \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} U(q) = - \begin{pmatrix} \frac{\partial U(q)}{\partial x} \\ \frac{\partial U(q)}{\partial y} \end{pmatrix}$$
Potential Fields
Attractive Potential

- Influence of potential has to be workspace wide!
- **Linearly decreasing** potential with increasing distance to goal \( q_{ziel} \):
  \[
  U_{att}(q) = \xi \rho(q), \quad \text{with} \quad \rho(q) = \|q - q_{ziel}\| \quad \text{and scaling factor } \xi.
  \]

Computation of force \( F_{att} \):
\[
\vec{F}_{att}(q) = -\xi \vec{\nabla} \rho(q)
= -\xi (q - q_{ziel})/\|q - q_{ziel}\|
\]

- **Singularity** at \( q = q_{ziel} \)!
  - Has to be dealt with separately
Potential Fields
Repulsive Potential

- Influence of potential can be limited in order to simplify computations
- Increasing potential with increasing distance to object:

\[
U_{\text{rep}}(q) = \begin{cases} 
\frac{1}{2} \eta \left( \frac{1}{\rho(q)} - \frac{1}{\rho_0} \right)^2 & \text{falls } \rho(q) \leq \rho_0 \\
0 & \text{falls } \rho(q) > \rho_0 
\end{cases}
\]

Where \( \eta \) is a scaling factor, \( p(q) \) the distance to the obstacle, and \( p_0 \) the maximal influence radius of the potential.

The distance function should respect the shape of the object, for example:

\[
\rho(q) = \min_{q' \in CB} \| q - q' \|
\]

Computation of force \( F_{\text{rep}} \):

\[
F_{\text{rep}}(q) = \begin{cases} 
\eta \left( \frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(q)} \nabla \rho(q) & \text{falls } \rho(q) \leq \rho_0 \\
0 & \text{falls } \rho(q) > \rho_0 
\end{cases}
\]

Strong rep. potential nearby to the object!
Potential Fields
Computing the potential field

Computation by:

$$U(q) = \sum_{i=1}^{M} U_{att_i}(q_{goal_i}) + \sum_{j=1}^{N} U_{rep_j}(q)$$

However: Reactive path selection can lead into local minima! Better: Finding goal with A* and using Potential Field values as heuristic.
Case-Study: Extracting predicates for playing soccer I

- Predicates are the basis for action selection and strategic decision making
- Can be considered as world model abstractions
- Simple predicates of objects (can be directly computed from positions):
  - \texttt{InOpponentsGoal}(object)
    - Object in opponent goal?
  - \texttt{InOwnGoal}(object)
    - Object in own goal?
  - \texttt{CloseToBorder}(object)
    - The distance to any border is beyond a threshold?
  - \texttt{FrontClear}()
    - Neither another object nor the border is in front?
  - \texttt{InDefense}(object)
    - Object in the last third of the soccer field?
Case-Study: Extracting predicates for playing soccer II

• **Extended predicates:**
  – computed by normalized potential fields: \((f_i: \mathbb{R} \times \mathbb{R} \rightarrow [0..1])\)
  – discretized by grid, e.g., 10x10cm cell size

• **Examples:**
  – \(f_{\text{free}}\): indicates positions under the influence of the opponent
  – \(f_{\text{covered}}\): indicates position covered by teammates
  – \(f_{\text{desired}}\): indicates tactical good positions
Case-Study: Extracting predicates for playing soccer III

- Given the predicates, for each role and action the best next desired position can be computed.

- Combined potential fields:
  - $f_{ballview}$: indicates whether the direct line from the ball to a position is free.
  - Recursive computation:

\[
\begin{align*}
    f_{ballview}(k(z_1)) &= 1 \\
    f_{ballview}(k(z_i)) &= f_{ballview}(k(z_{i-1})) \cdot f_{free}(k(z_i)) \cdot (1 - f_{covered}(k(z_i)))
\end{align*}
\]

Where $z_1, ..., z_n$ are the indices of “lines”, i.e., the cells going from the ball towards the border (star-like).
Summary

• Consistent world models are the key to deliberative acting!

• The Kalman Filter is a tool for accurately estimating object poses
  – However, only single hypotheses can be tracked

• Markov Localization is a tool for robust object tracking by considering multiple hypotheses
  – However, accuracy depends on the chosen discretization

• Best results are yielded by combining both methods

• Potential Fields are an efficient tool for generating predicates from complex representations, simplifying decision making of a mobile agent

• For path planning, care has to be taken on local minima
Literature

• Kalman Filter:
  – Website: http://www.cs.unc.edu/~welch/kalman/

• Markov Localization:

• Cooperative Sensing:

• Potential Fields:
  – C. Reetz, *Aktionsauswahl in dynamischen Umgebungen am Beispiel Roboterfußball*, Diplomarbeit an der Fakultät für Angewandte Wissenschaften, Univ. Freiburg, 1999