Marker-based FastSLAM on the AlphaBot2

Autonomous Systems Project 2023/2024

J. Pessoa
ist198915

J. Gonçalves
ist199995

M. Ribeiro
ist196446

T. Nogueira
ist1100029

Group 27

Instituto Superior Técnico

Presentation Overview

1 Data Association Types

2 Covariation Matrix

3 Alphabot’s Motion Drivers

4 Questions

5 Thank You!

Data Association Types

- Expanded research on the data association topic.
  - Namely, joint compatibility branch and bound (JCBB).

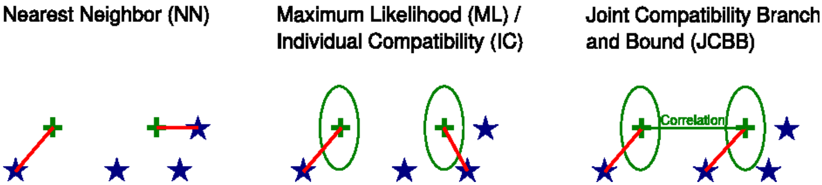

Figure: Data association types

Nearest Neighbor

- Simplest data association method.
- Associates each new observation z_t with the closest feature m_i in the map based on a distance metric, usually Euclidean distance
  $d(z_t, m_i) = \sqrt{(z_{tx} - m_{ix})^2 + (z_{ty} - m_{iy})^2}$
- Complexity: $O(m)$.
- Limitations:
  - Doesn't account for sensor noise.
  - Prone to ambiguity in environments with similar features.

Maximum Likelihood

- Considers the probability of each association hypothesis.
- Calculates a likelihood score $P(z_t|m_i)$ for each possible association between an observation and a landmark.
  $P(z_t|m_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(d(z_t,m_i))^2}{2\sigma^2}\right)$
- Chooses the association with the highest likelihood (most likely to be correct).
- More robust than nearest neighbor due to considering sensor noise.
- Complexity: $O(m^2)$.

Data Association Types

Maximum Likelihood

Measurement	Landmark 1	Landmark 2	New Landmark 1	New Landmark 2	New Landmark 3
1	0.8	0.3	0.1	0.1	0.1
2	0.5	0.7	0.1	0.1	0.1
3	0.01	0.01	0.1	0.1	0.1

Data Association:

- Measurement 1: Best match is Landmark 1 (0.8)
- Measurement 2: Best match is Landmark 2 (0.7)
- Measurement 3: Best match is New Landmark 1 (0.1)

Joint Compatibility Branch and Bound

- Evaluates the overall compatibility score of all possible association combinations for a set of observations
- Employs a branch and bound search algorithm to explore combinations efficiently.
- Basic compatibility score between observation z_i and feature m_j: $C(z_i, m_j) = \exp\left(-\frac{(d(z_i, m_j))^2}{2\sigma^2}\right)$
- Overall compatibility score for an association combination considers individual scores and their spatial relationships.
- Complexity: $O(m^3)$.

Imagine a robot in a room with three landmarks (L_1, L_2, L_3) and takes sensor readings (z_1, z_2, z_3). JCBB considers all combinations of associating these observations to landmarks.

Combination	z_1-Landmark	z_2-Landmark	z_3-Landmark	Compatibility Score (Example Values)
1	L_1	L_2	L_3	0.8 (High)
2	L_1	L_3	L_2	0.5 (Moderate)
3	L_2	L_1	L_3	0.2 (Low)

In this example, Combination 1 might have the highest score because all observations have high compatibility with their assigned landmarks. Finally, JCBB selects Combination 1 as the most likely set of data associations.

Data Association Types

Comparison of Data Association Types

Method	Complexity	Strengths	Weaknesses
NN	$O(m)$	Simple, efficient	Ignores sensor noise, prone to ambiguity
ML	$O(m^2)$	More robust than NN, considers sensor noise	Doesn't consider spatial relationships
JCBB	$O(m^3)$*	Highest accuracy, leverages spatial relationships	Computationally expensive for large datasets if not developed

*for now

TÉCNICO LISBOA

- JCBB's high complexity ($O(m^n)$) can be prohibitive for real-time applications like FastSLAM.
- **Our choice (for now)**: For FastSLAM, ML offers a good balance between accuracy and efficiency ($O(m^2)$).
- Alternative Approaches: Explore iterations of JCBB **- by performing the algorithm incrementally we can reduce the complexity to O($\mathbf{m^2}$) -** or techniques that leverage specific previously known landmarks to potentially improve data association accuracy within FastSLAM's computational constraints

Covariation Matrix

Covariation Matrix

- We have taken measurements at 4 different distances from the target Aruco markers (60cm, 120cm, 180cm, 240cm) for all the 7 variables $(x, y, z, q_w, q_x, q_y, q_z)$ at study.
- This was performed in order to better understand the variance of these variables.
- For each variable and distance, we had a large number of measurements in different conditions and calculated the variance of that variable at that distance from the markers.
- Along with that, we performed a regression to understand how the variance evolves with distance.

Covariation Matrix
Obtained Graphs and Regressions

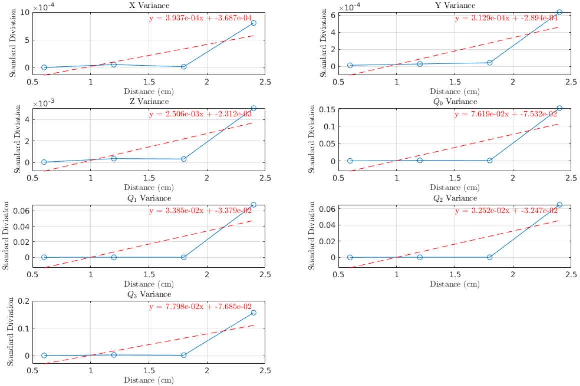

Figure: Calculated standard deviation for various ranges

- Therefore, our dynamic variance is given by:

$$\mathrm{Diag}((a \times range + b)^2)$$

where a and b are distinct for each of the seven variables.

- Without the use of such a technique, low variance values of the measurement, coupled with the high variance associated with the motion model, would result in landmark's position drift
- Here is an example of such behaviour in display:

`https://youtu.be/Hus8uk9nVGs`

Alphabot’s Motion Drivers

Upon trying to retrieve proper rosbags for later processing with our algorithm, we came across several difficult challenges:

- The robot was **unable to move in a straight line** when given only a linear velocity;
- The robot was **unable to spin on itself**, turn left and right;
- The robot's pan and tilt was **jittery and of difficult control**;
- The provided twist message contained velocities that did not correspond to reality due to **wrong processing in the driver's code**;

This made the robot **unusable** and hindered our ability to progress. Solution?

- **Develop a new driver and a proper custom command interface**

The difference is very palpable, as we can see in the following video: `https://youtu.be/F2Mxs-8-AKQ`

Alphabot’s Motion Drivers

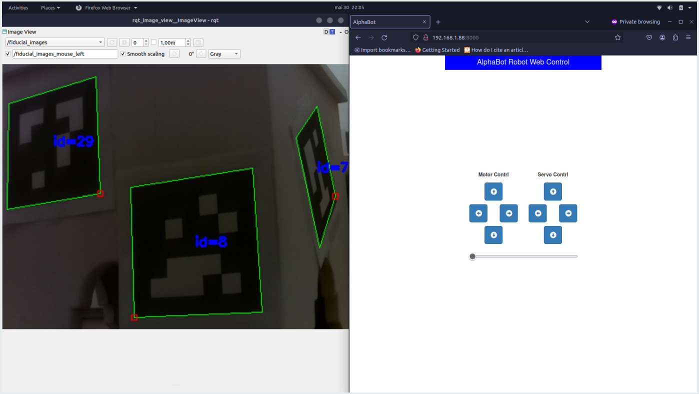

Figure: Developed motion driver’s control interface

Questions

Thank You!