Machine Learning for Probabilistic Robotics with Webots

Joan Gerard[1]
Promotor: Prof. Gianluca Bontempi [1]

[1]Université Libre de Bruxelles

June 7, 2019

Table of Contents

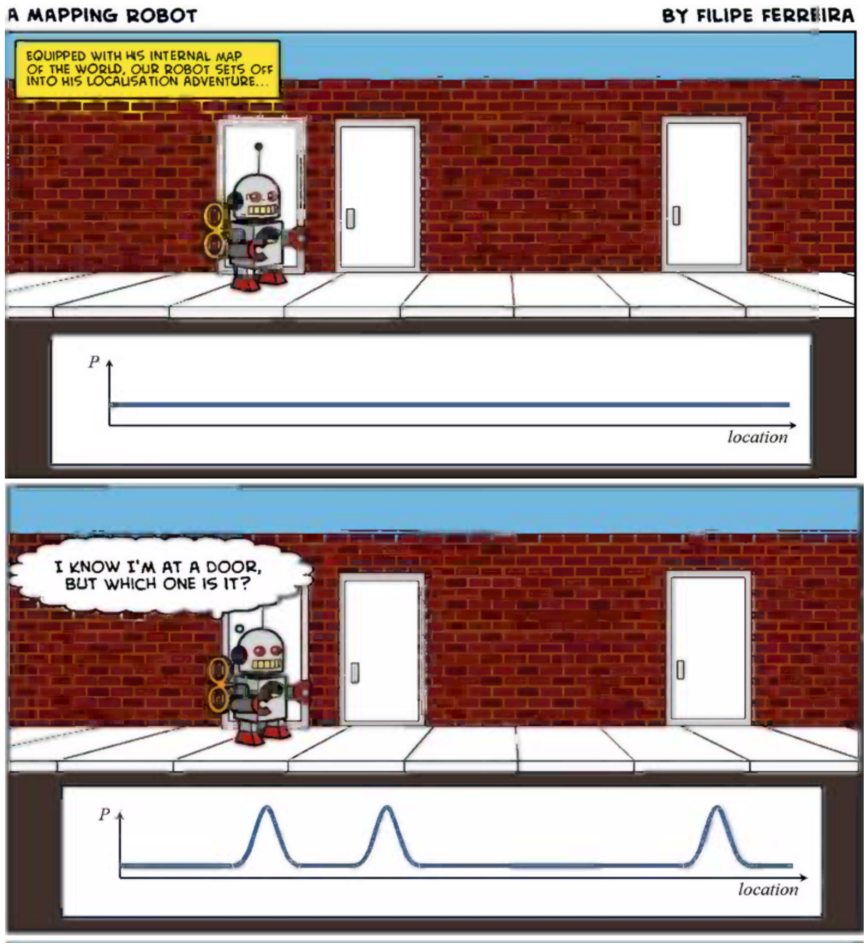
Table of Contents

Where am I?

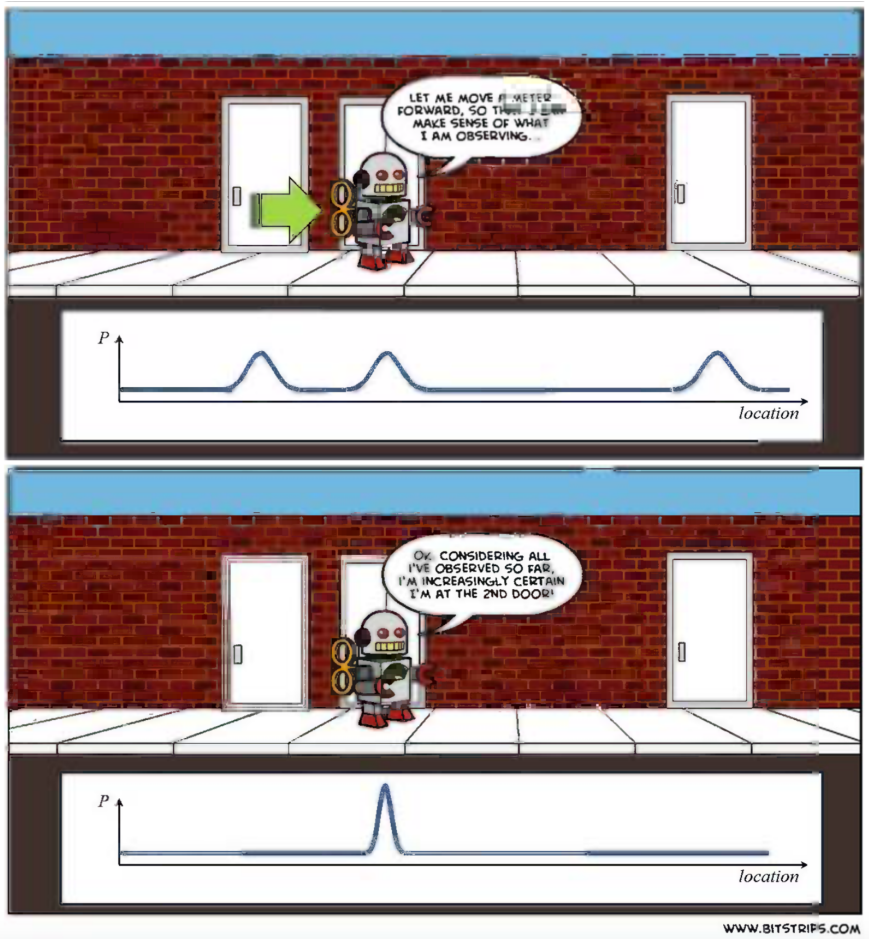
- Robot knows where it is
- Robot is kidnapped
- Robot does not know where it is

Let's figure it out using Bayes Filter!

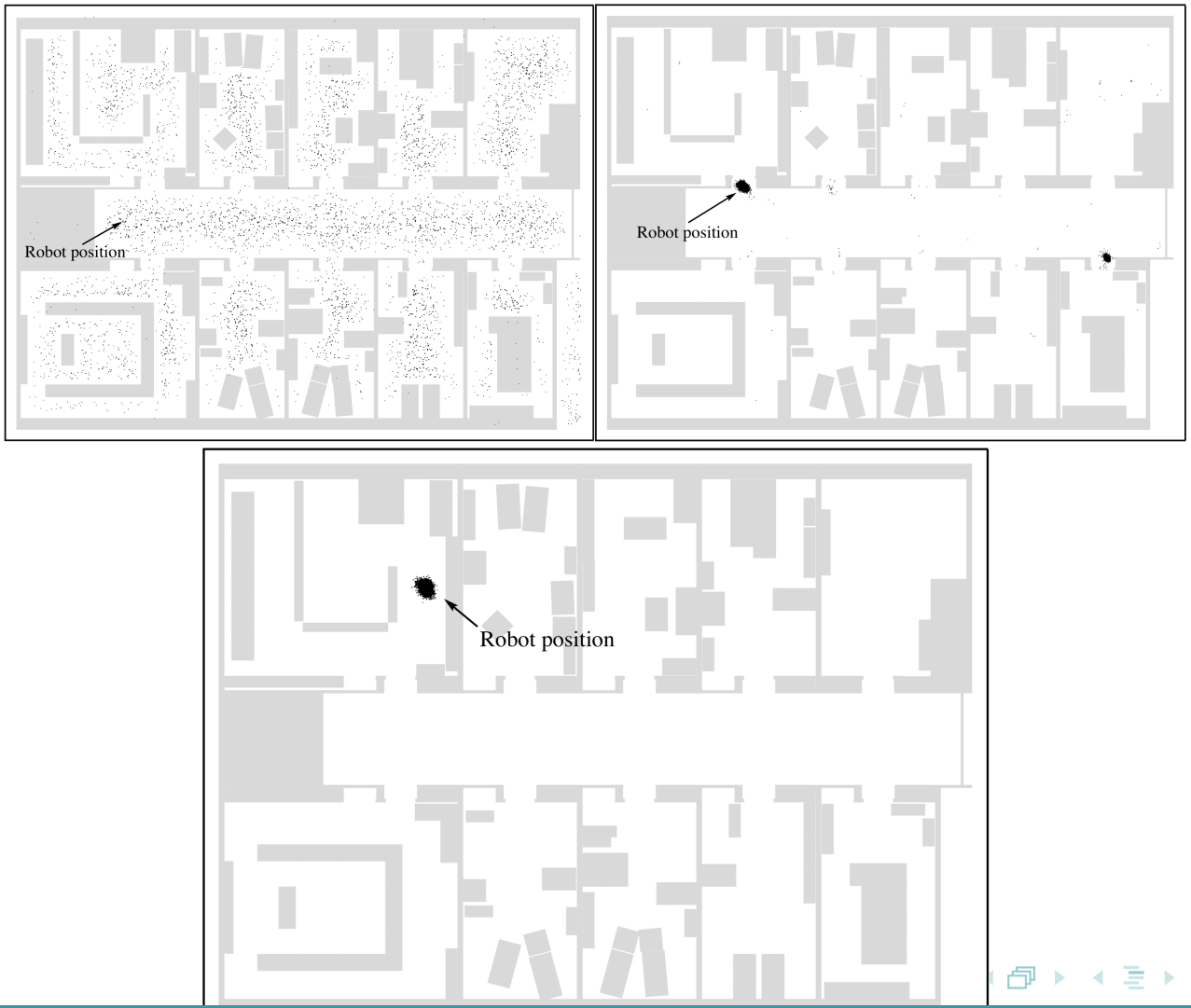
- Prediction



- Error Correction



What about Particles Filter?



Particles Filter Algorithm

- Train/test a prediction model: robot state $\rightarrow$ distance sensor data
- Compare it with the true sensor data and determine a weight for each particle
- More close to the real sensor data more weight will have

---

input : particles, controlAction, sensorData

output: nextParticles

1 *nextParticles* $\leftarrow \emptyset$

2 **foreach** *particle* $\in$ *particles* **do**

3 | particle.state $\leftarrow$ predictState(particle.state, controlAction)

4 | particle.weight $\leftarrow$ calculateWeight(particle.state, sensorData)

5 **end**

6 **for** $m = 1$ *to* $m = |partices|$ **do**

7 | *newParticle* $\leftarrow$ draw i from *particles* with probability $\propto$ $particles[i].weight$

8 | nextParticles.add(newParticle)

9 **end**

Table of Contents

What tool can we use for simulating that problem?

Webots!

- Open Source
- Python, C++, Java, etc.
- 44 robot models
- Robot model creation/customization
- Robust documentation

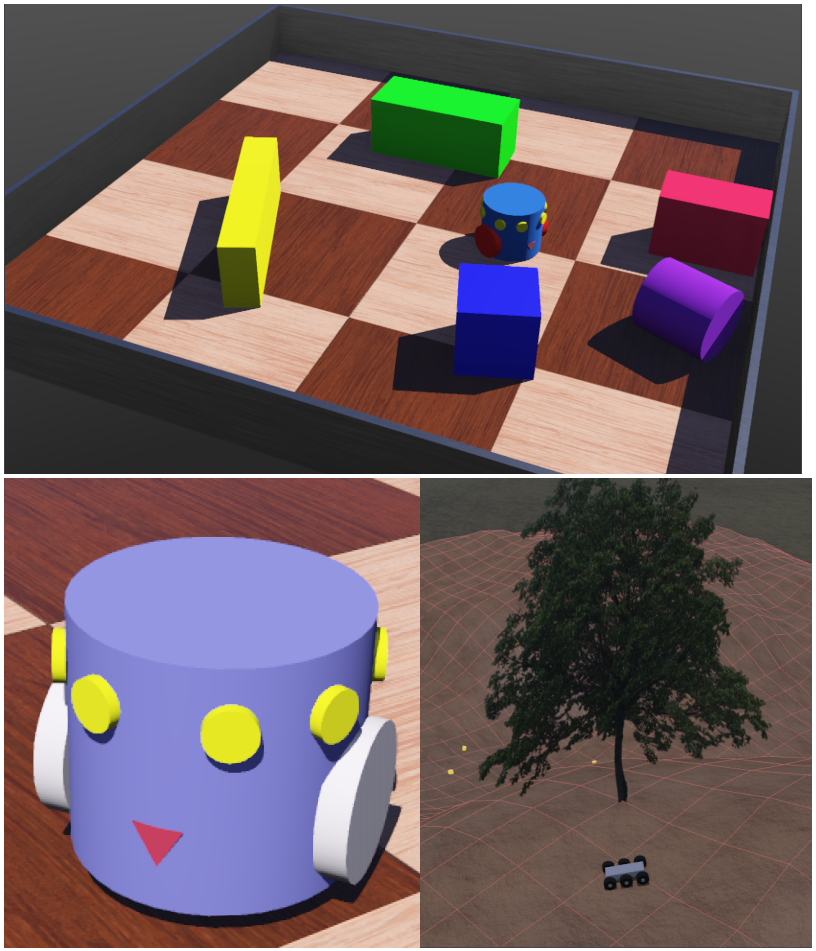


Table of Contents

Main Objective

- The objective will be to exploit the simulation benefits of Webots to introduce Machine Learning techniques together with non-parametric filters (such as Particles Filter) for robot positioning estimation, independently from the kind of robot used for in-door environments.

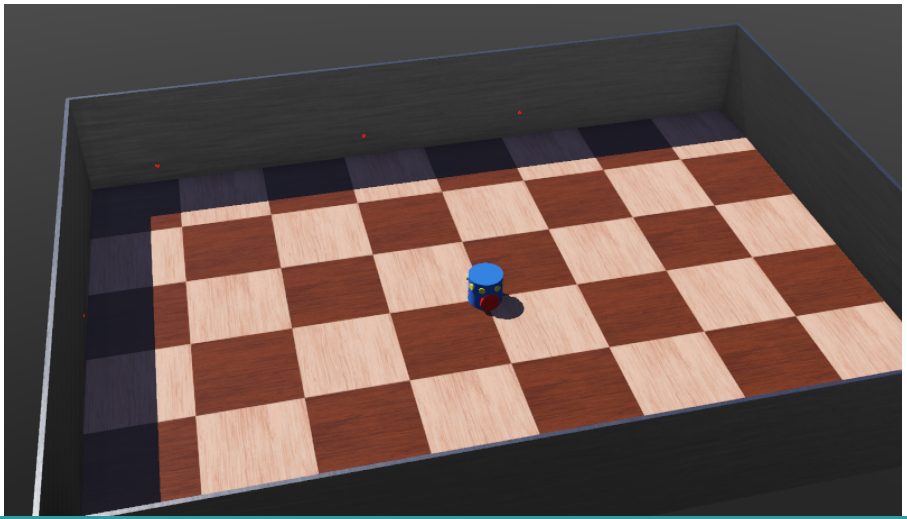
How do we plan to do it?

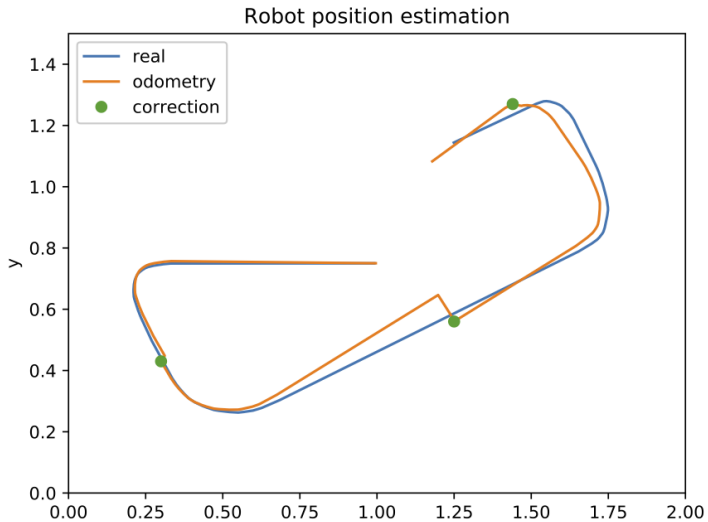
- Using Webots
  - Create a custom robot
  - Randomize control actions while moving avoiding obstacles
  - Use a realistic in-door environment
- Using Machine Learning and Particles Filters
  - Positioning estimation
  - Error correction
- Test the developed technique with other robots

Table of Contents

What have we done so far?

- Robot creation
- Dummy environment creation
- Collect distance sensor data + real position coordinates
- Bayes Filter algorithm implementation
  - *Position prediction*: Odometry techniques
  - *Error correction*: Training a Random Forest model using collected data





Are the results good?

- Yes! but there are improvements to be done:
  - Collect more data (+ Simulations)
  - Randomize robot moves
  - Train/test with other Machine Learning models
  - Improve performance of the error correction algorithm

Any question?