JC3504 Robot Technology

Lecture 12: Simultaneous Localisation and Mapping

Dr Xiao Li xiao.li@abdn.ac.uk

Dr Junfeng Gao junfeng.gao@abdn.ac.uk

Outline

SLAM Introduction

SLAM Introduction

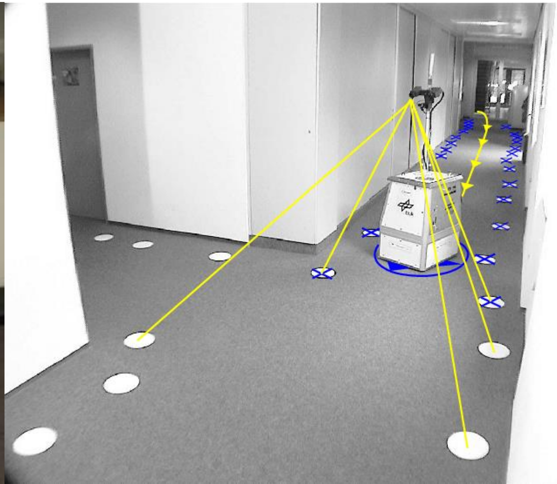
The SLAM (Simultaneous Localisation and Mapping) problem has long been a cornerstone issue in the realm of autonomous mobile robotics. It lays the groundwork for a robot to be transported to an unknown location and to explore the area, constructing metrically accurate maps and pose estimates solely through onboard sensing. This could prove invaluable for any field robot, whether terrestrial, extraterrestrial, under the ocean, or in an unexplored built environment.

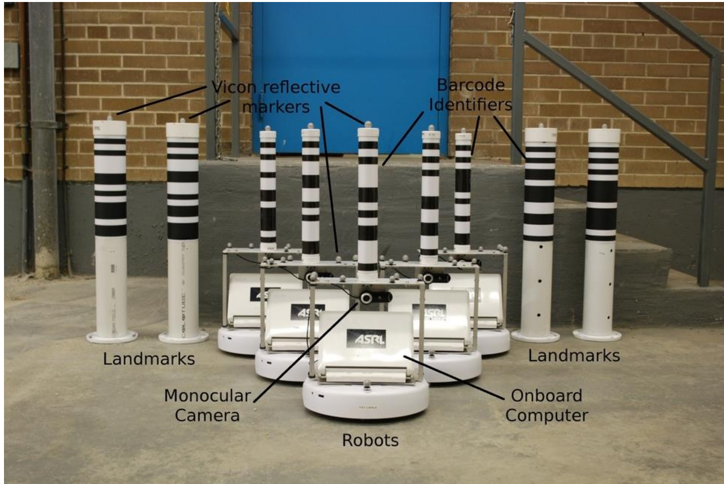
This session will introduce one of the first comprehensive solutions to the problem to build understanding, even though it has since been superseded by versions that are computationally more efficient, boasting a variety of algorithmic speed-ups and accuracy improvements.

Landmark

In SLAM, landmarks are distinct features in the environment that a robot uses as reference points to navigate and map its surroundings more accurately, enabling precise positioning and orientation by providing fixed, recognisable features in the environment.

Landmarks can vary widely, including natural features like trees and rocks, artificial structures like buildings and traffic signs, and specially designed markers or beacons.

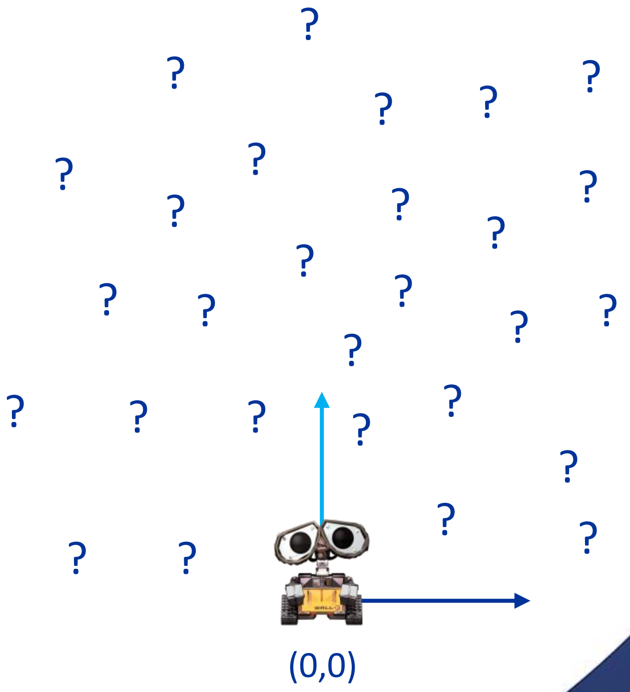




Simplified SLAM Process

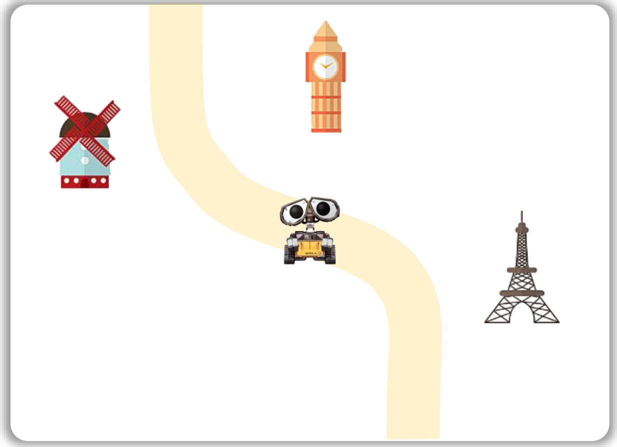
At the outset, the robot is placed in an unknown environment, without a map at its disposal.

For convenience, the robot establishes a coordinate system with itself as the origin.



Simplified SLAM Process

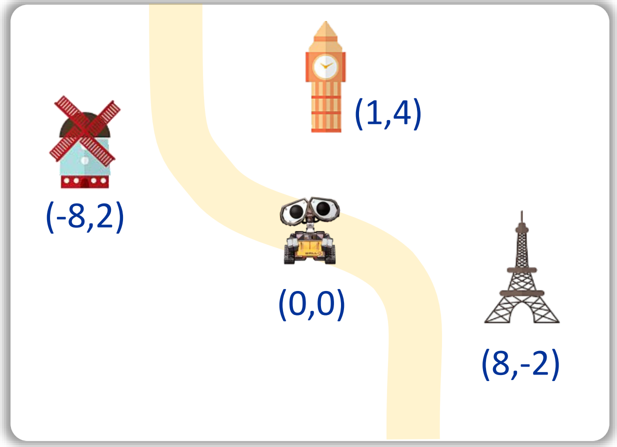
Subsequently, the robot scans the surrounding environment, searching for recognisable landmarks, and creates a snapshot of the surroundings.



UNIVERSITY OF ABERDEEN

Simplified SLAM Process

Since we have already established a coordinate system, we are able to calculate the coordinates of each landmark based on the results from the sensors.



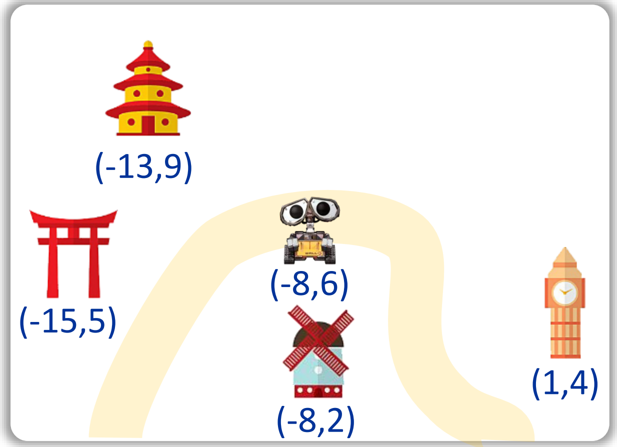
UNIVERSITY OF ABERDEEN

Simplified SLAM Process

Then, the robot moves a certain distance and scans the surrounding environment once again, searching for recognisable landmarks, and creates another snapshot of the surroundings.

Simplified SLAM Process

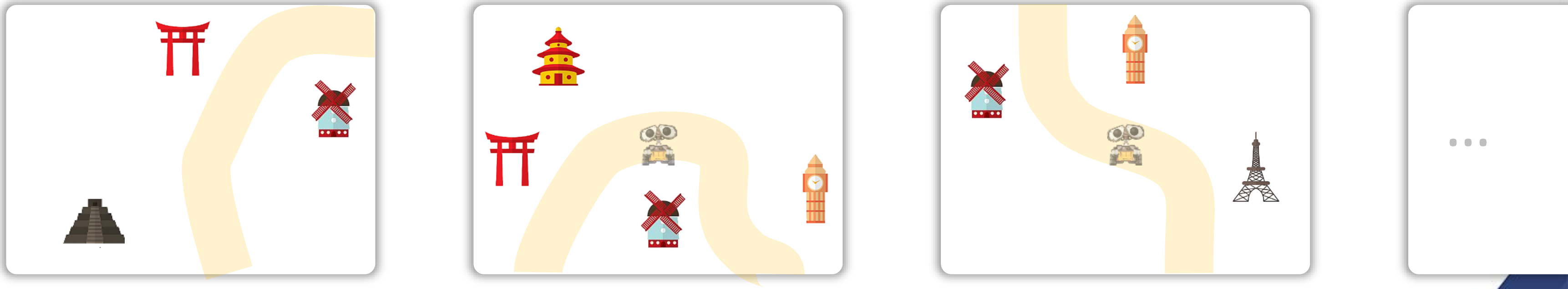
We can refer to the coordinates of known landmarks, and with the robot's displacement sensors, calculate the coordinates of new landmarks.



UNIVERSITY OF ABERDEEN

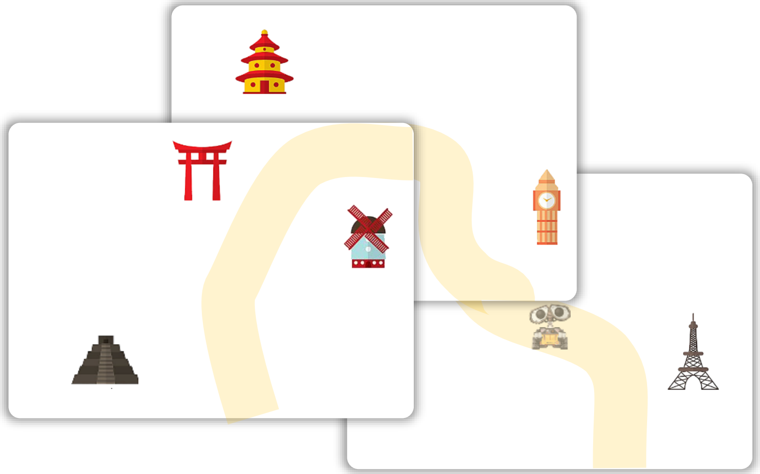
Simplified SLAM Process

As the robot continues to move, it is able to take multiple such photographs. Based on the robot's movement records and the landmarks, it can stitch these photographs together to form a map of the area it is in.



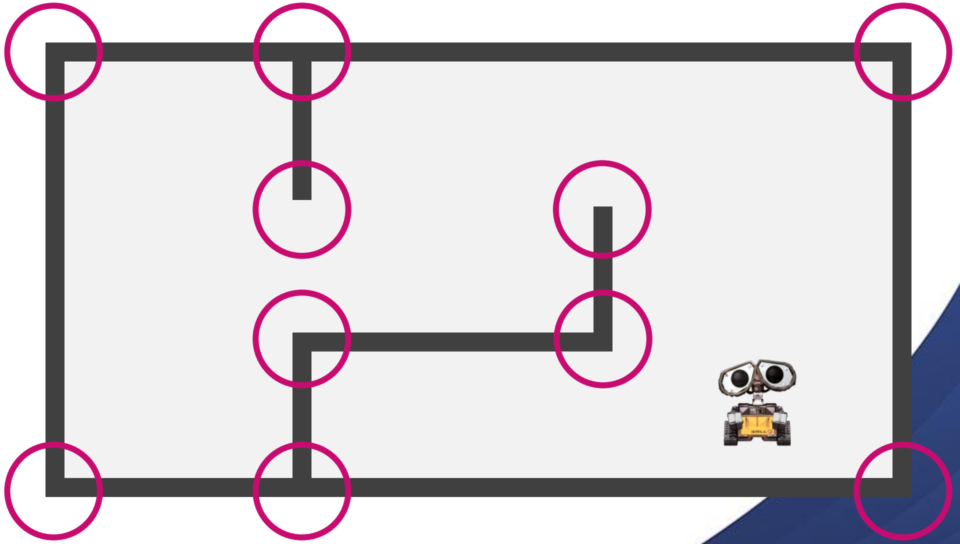
Simplified SLAM Process

As the robot continues to move, it is able to take multiple such photographs. Based on the robot's movement records and the landmarks, it can stitch these photographs together to form a map of the area it is in.



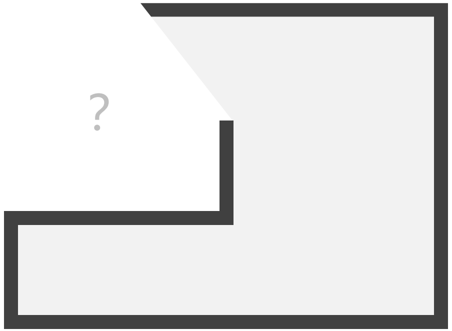
Simplified SLAM Process

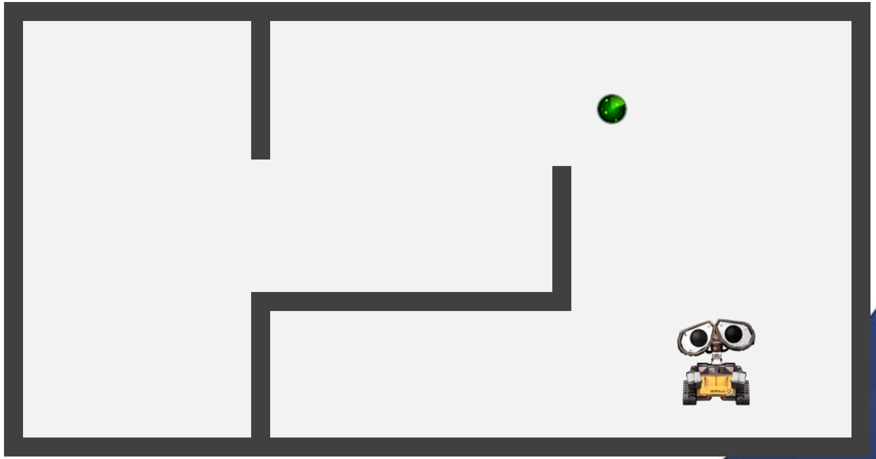
In the absence of landmarks, such as when using LIDAR, edges or corners can be used as references for merging maps.



Simplified SLAM Process

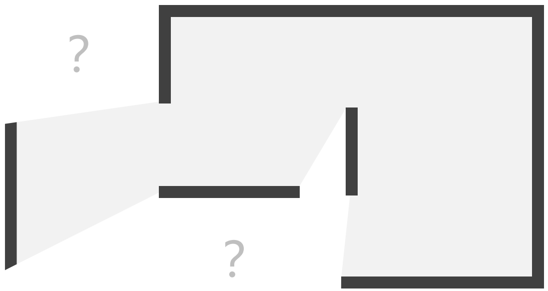
When the robot initially scans its surroundings, only a partial map is obtained due to obstructions by walls.

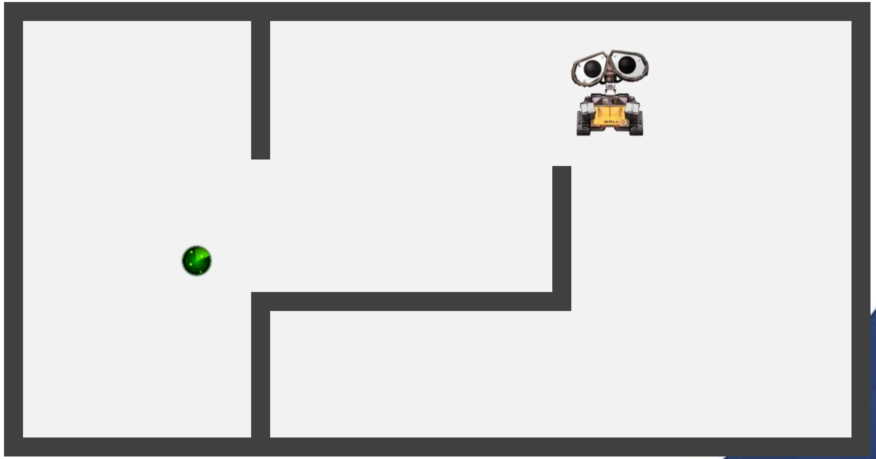




Simplified SLAM Process

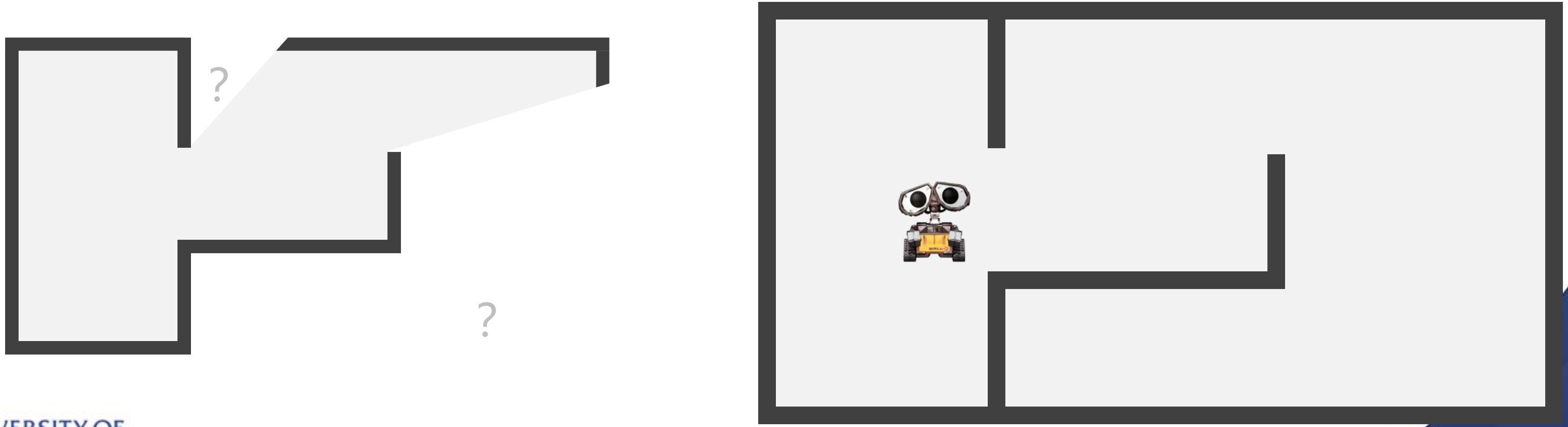
When the robot initially scans its surroundings, only a partial map is obtained due to obstructions by walls.





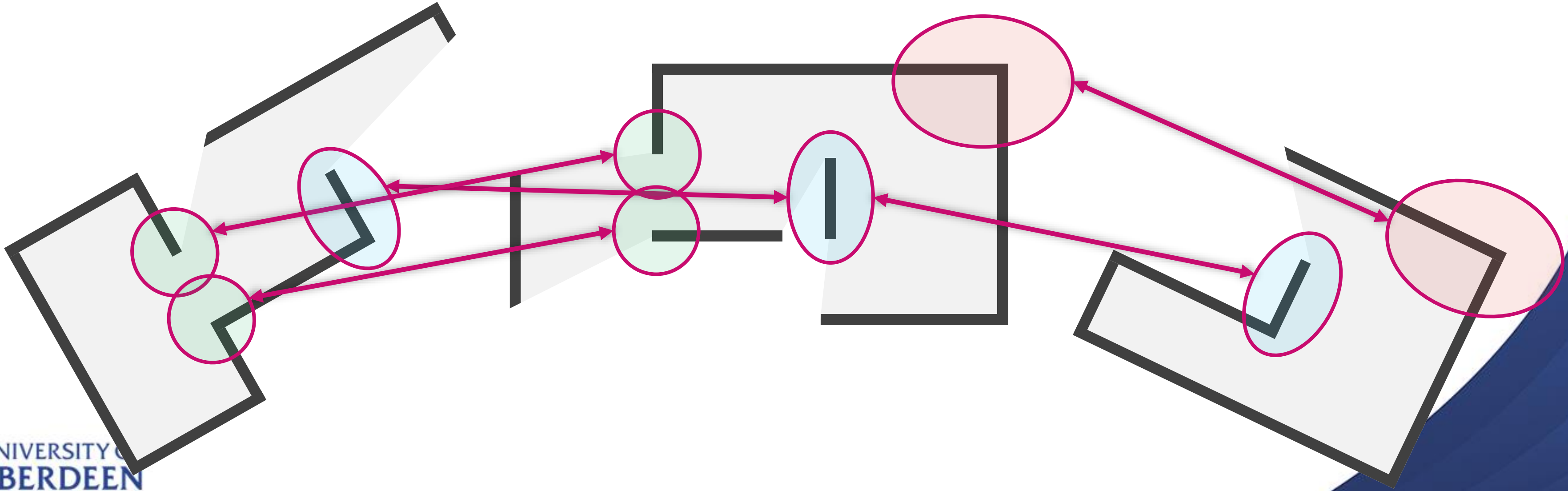
Simplified SLAM Process

When the robot initially scans its surroundings, only a partial map is obtained due to obstructions by walls.



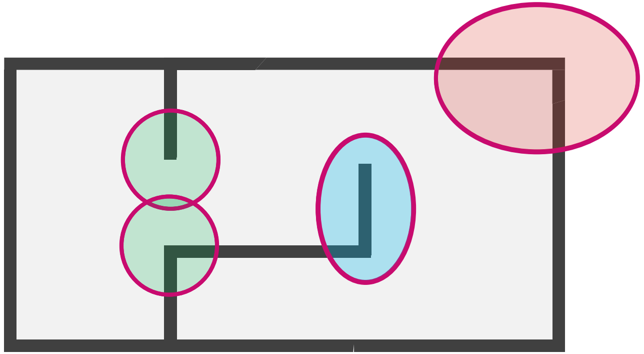
Simplified SLAM Process

Upon acquiring segments of the map, we compare the edges and corners of the map, looking for similar geometric structures, and use them as landmarks to merge different map segments.



Simplified SLAM Process

Upon acquiring segments of the map, we compare the edges and corners of the map, looking for similar geometric structures, and use them as landmarks to merge different map segments.



SLAM Principal Challenges

From the simplified SLAM process, we can summarise its principal challenges.

- **Data Association and Feature Extraction**: One of the paramount challenges in SLAM is accurately identifying and matching features in the environment across different points in time.
- **Sensor Noise and Uncertainty**: All sensors have inherent limitations and are prone to noise and errors.
- **Dynamic and Changing Environments**: Many environments are not static but change over time, with objects moving, appearing, or disappearing.
- **Loop Closure Detection**: Detecting when a device has returned to a previously visited location (loop closure) can significantly improve the accuracy of the map and the location estimate by correcting cumulative errors.

SLAM Algorithms

EKF SLAM

EKF SLAM (Extended Kalman Filter Simultaneous Localisation and Mapping) is a sophisticated technique that utilises the Extended Kalman Filter to concurrently estimate a robot's location and map its environment in real-time, despite the inherent non-linearity of the task.

The key idea in EKF SLAM is to extend the state vector from the robot's position (and potentially pose) to contain the position of all landmarks.

That is, in Kalman Filter: $\hat{x}_k = [x, y, \theta]^T$, where x, y denote the robot position, and θ denotes the robot orientation.

In EKF SLAM,

$$\hat{x}_k = [x, y, \theta, \alpha_1, \beta_1, \alpha_2, \beta_2, \dots]^T$$

where α_i, β_i denote the position of the landmark i.

EKF SLAM

Just like Kalman Filter, EKF SLAM also has two phases: Prediction and Updating.

The prediction phase is the same as of Kalman Filter. Since the landmarks do not move, F, B, and u_k do not involve any operations on α_i and β_i.

The updating phase considers the observations of the landmarks, which is reflected in z_k. So, H_k and K_k also involve the corresponding changes.

Kalman Filter

Prediction

$$\hat{x}_{k-} = F\hat{x}_{k-1} + Bu_k$$
$$P_{k-} = FP_{k-1}F^T + Q_k$$

Updating

$$K_k = P_{k-}\, H_k^T \left(H_k\, P_{k-}\, H_k^T + R_k\right)^{-1}$$
$$\hat{x}_k = \hat{x}_{k-} + K_k(z_k - H_k\, \hat{x}_{k-})$$
$$P_k = (I - K_k H_k) p_{k-}$$

UNIVERSITY OF ABERDEEN

EKF SLAM (Advantages)

The principal advantage of Extended Kalman Filter SLAM (EKF SLAM) lies in its adeptness at providing continuous state estimation and map construction, coupled with its robust handling of sensor noise and uncertainties.

Through the integration of data from various sensors, even when affected by noise, EKF SLAM utilises filtering to mitigate errors, offering relatively precise estimations of both the robot's location and the environmental map.

This renders EKF SLAM particularly suited to applications requiring accurate localisation and mapping, such as autonomous navigation and robotic exploration.

EKF SLAM (Challenges)

The foremost challenge facing EKF SLAM is its computational complexity, which significantly escalates with the addition of environmental features. In EKF SLAM, the state vector encompasses the robot's position and the positions of all features in the map, with the covariance matrix's dimension being the square of the state vector's dimension.

As the map features increase, the number of states requiring estimation surges, leading to substantial increases in computational and storage demands. This presents a considerable challenge for systems with limited computational resources, especially in large-scale or feature-rich environments.

EKF SLAM (Demo)

< Video: 12 - EKF SLAM from Scratch.mp4>

Graph-based SLAM

Graph-based SLAM, an advanced technique within the Simultaneous Localisation and Mapping (SLAM) sphere, operates by constructing and optimising a graph to estimate a robot's position within its environment alongside mapping said environment.

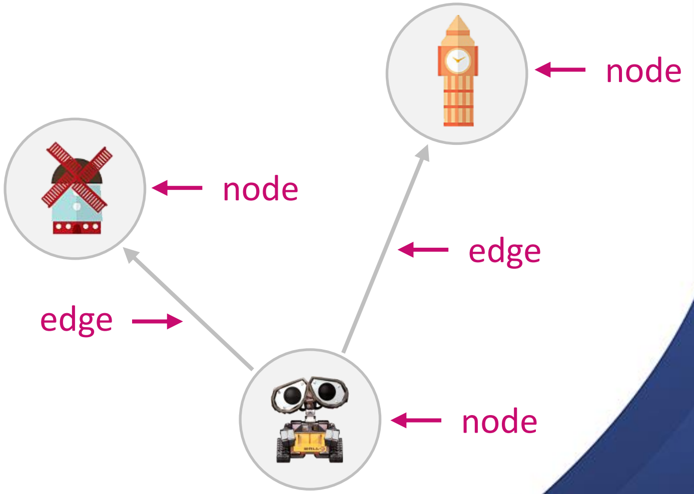
This methodology's crux lies in utilising a graph structure to represent the robot's movements and observations within the environment. In this graph, nodes symbolise the robot's positions (or environmental features), while edges depict the relative relationships between these positions, such as movements or the distances and directions derived from observations.

Graph-based SLAM

Node Creation

A new node is added to the graph whenever the robot reaches a new position or detects a new environmental feature.

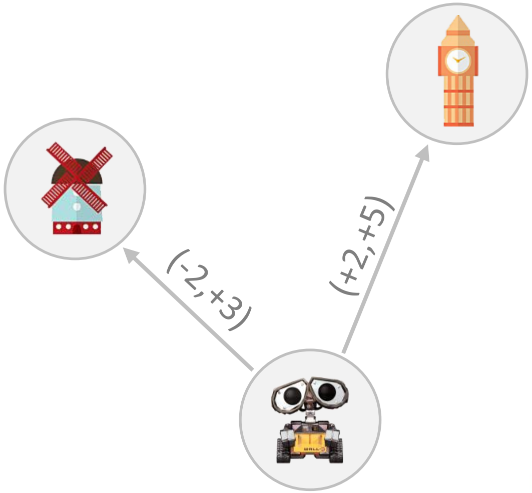
These nodes represent either the robot's positions or specific points within the environment.



Graph-based SLAM

Edge Creation

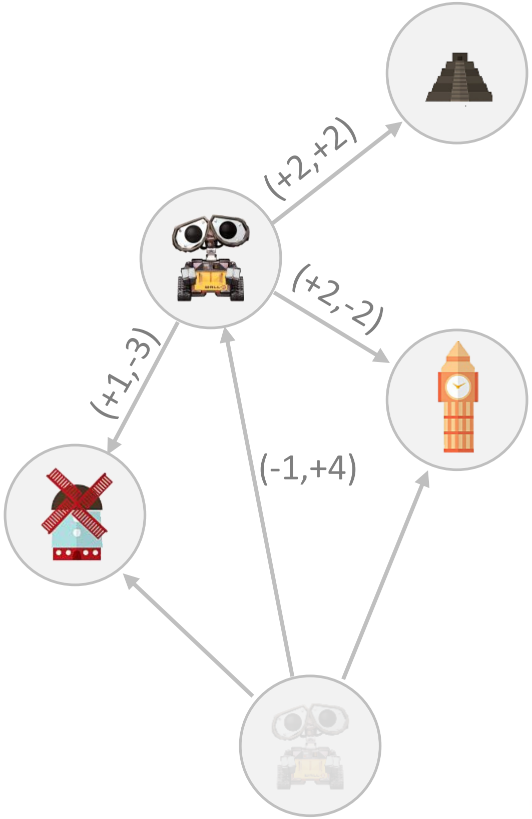
This edge contains relative positional information from one node to the other, derived from the robot's motion sensors (like wheel encoders, Inertial Measurement Units (IMUs)) or environmental observation sensors (such as cameras, LiDAR).



Graph-based SLAM

Edge Creation

As the robot moves from one position to another or observes a feature from a certain position, an edge is established between the respective nodes in the graph.

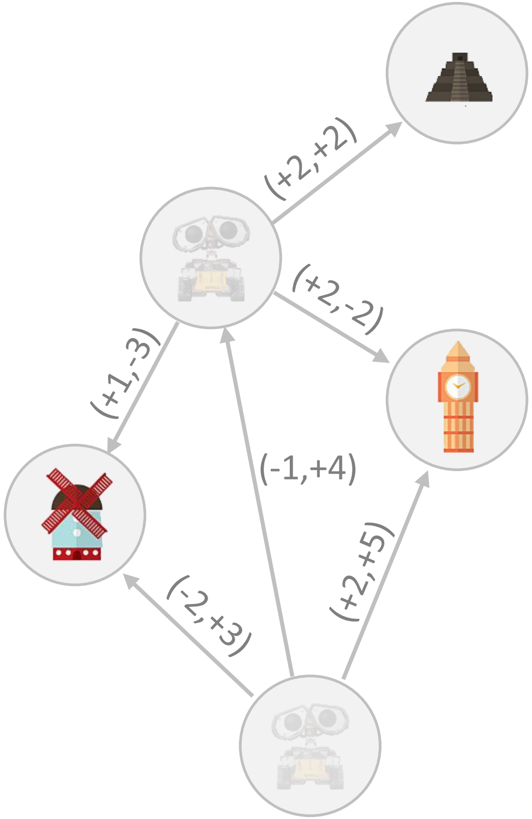


Graph-based SLAM

Graph Optimisation

As new positional and environmental observation information is continually gathered and incorporated into the graph through new nodes and edges, discrepancies may arise due to sensor inaccuracies.

Graph-based SLAM employs graph optimisation algorithms (such as g2o, Ceres, etc.) to adjust the positions of nodes within the graph, thus minimising these discrepancies and optimising the graph's structure. The aim of graph optimisation is to minimise the errors across all edges, typically solved as a non-linear least squares problem.



Graph-based SLAM (Advantages)

- High Accuracy: By globally optimising the graph structure, Graph-based SLAM can effectively reduce cumulative errors, offering high-accuracy localisation and mapping outcomes.
- Strong Adaptability: This method adapts well to SLAM problems in large-scale environments, unaffected by the complexity of the environment.
- Flexibility: Graph-based SLAM is adaptable to various types and configurations of sensors, allowing for the combined use of different sensor data.
- Scalability: The graph structure facilitates localised data association, making it suitable for distributed computation and modular processing in large environments.

Graph-based SLAM (Challenges)

- Computational Resources: Graph optimisation is a computationally intensive process, requiring significant computational resources, especially when dealing with large-scale environments and extensive data.
- Data Association: Correctly associating observational data with nodes in the graph presents a challenge in large-scale or feature-sparse environments.
- Loop Closure Detection: To further enhance the accuracy of SLAM, detecting when the robot returns to a previously visited location (loop closure) is a complex issue in practical applications.

EKF SLAM (Demo)

< Video: 12 - Graph-Based SLAM>

Conclusion

Simultaneous Localization and Mapping (SLAM) is a key capability for mobile robots to operate autonomously in the world.

There exist robust implementations for environments with strong landmarks, that is landmarks that can be reliably localized and identified, that use different forms of optimization to find a collection of poses that are most likely given the available measurements.

SLAM strongly benefits from additional sensors, that can provide additional evidence, in particular beacon-based sensors such as GPS.

How to deal with environments with dynamical objects, that is changing maps, remains an open problem.

Additional Source

<Video: 12 - Understanding SLAM Using Pose Graph Optimization>