MAP MY WORLD

Mapping your environment is the next logical step after your robot localizes itself. This project is about helping the agent understand its environment and map them so it is aware of its surroundings. This will also address the kidnapped robot problem. Addressing the problem of the robot mapping its environment and localizing itself can be covered in 5 types of algorithms:

Extended Kalman Filter

Sparse Extended Information Filter

Extended Information Filter

FastSLAM

GraphSLAM

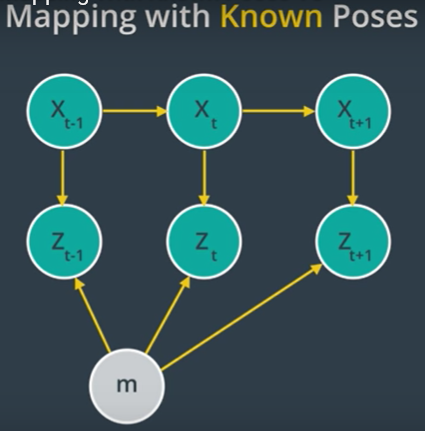
The FastSLAM algorithm uses the particle filter approach with a low dimensional extended Kalman filter to solve the SLAM problem. Real Time Appearance Based Mapping is used to help us with this problem. Mapping the environment poses two different challenges –

Unknown map and poses

Huge hypothesis space

The occupancy grid mapping algorithm helps us in estimating the map with unknown poses. It presents a discrete approximation of the map. But complexities like the map size and the memory size to hold the estimate of maps, noise from actuators and sensors, ambiguity makes map estimation all the more difficult.

Mapping with known poses is a problem that is represented as below

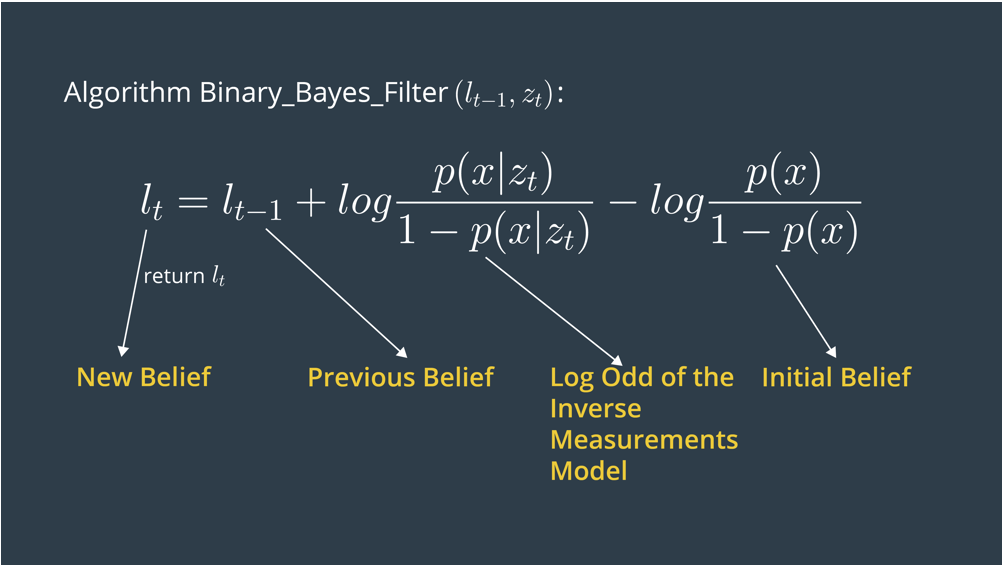


* Xt – poses
* Zt – measurements
* M - map

This will then help derive the map with noisy measurement. But map estimation happens after SLAM, thus empowering it by using the exact robot poses filtered from SLAM.

The occupancy grid divides the space into finite number of grid cells. The sensor in the robot then aids in filling up the grid with 0 or a 1, based on if it finds an obstacle there. The estimate of the map will be 2 to the power of the number of cells.

The Binary Bayes algorithm helps in estimating the occupancy probability of each grid cell by the following formula:



The inverse sensor model also helps in filling up the grid in even the borders of the vision cone of the robot and avoid numerical instabilities. This will help in representing the 3D world as accurately as possible. An optimal representation will require characteristics like probabilistic data representations to accommodate sensor noise and dynamic environments, distinguishable free space vs unoccupied space, efficiency with limited memory. The following maps will help us achieve this:

2.5D Maps – Also known as height maps, that data about the environment as height measured

Elevation Maps – 2D grids of height and certainty of the measurement

Extended Elevation Maps – Stores data as patches of height mean, height variance and depth value.

OctoMap – Uses probabilistic occupancy estimation to convert and integrate point clouds into 3D occupancy maps

SLAM

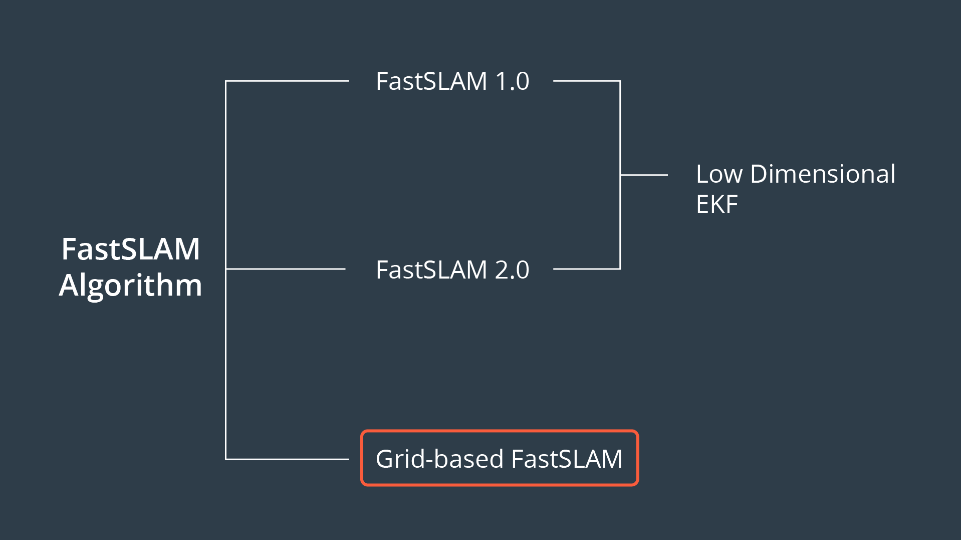
This addresses the issues of calculating the posterior map based on controls and measurements.

Online SLAM – Resolve posterior map considering the previous control and measurements only, not historical data.

Full SLAM – Resolve posterior map over the entire set of values of control and measurements

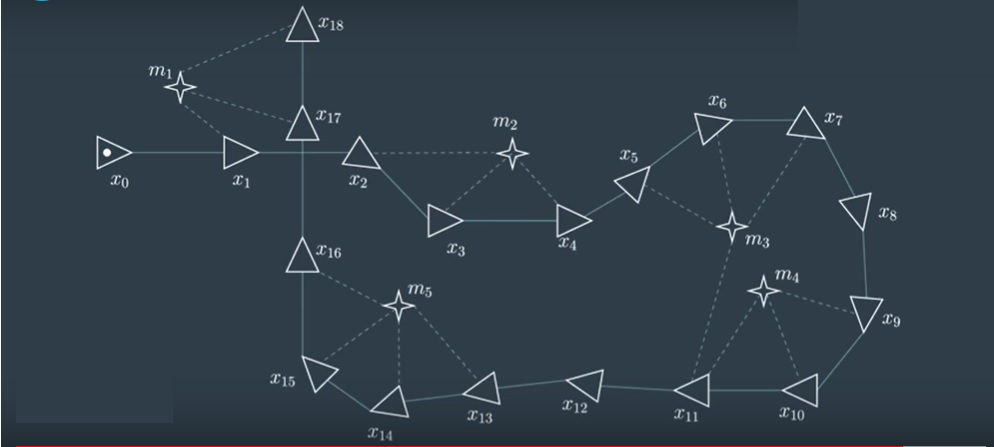
SLAM problems are also both continuous and discrete. The continuous nature comes from the sensors and their measurements that are gathered when the robot moves around The discrete nature helps in identifying features that it has encountered before, hence helping the robot close the loop.

FastSLAM solves the Full SLAM problem by estimating the trajectory using a particle filter approach and estimating the map using a low dimensional Extended Kalman Filter.



Graph SLAM

This uses a feature map, where the robot relates its pose and position to a feature tagged to calculate the posterior map.



The constraints, defined a bond between particles, will be adjusted so the overall error in the constraints is the minimum. Graph SLAM constitutes of a front end and back end part. The front end part will calculate the graph based on the sensor measurements. The back end outputs the most probable and optimal configuration of all the complete graph with all possible constraints.

Real Time Appearance Based Mapping (RTABMap) makes use of a process called loop closure to identify previously visited places. The number of features to be compared for loop closure keeps expanding the more the robot travels. This will increase the complexity. This is overcome by using memory management and a technique called Bag Of Words.