“SLAM for Dummies” for Dummies

- The term SLAM is as stated an acronym for Simultaneous Localization and Mapping.

- SLAM is more a concept that a single algorithm; there are many ways to implement SLAM, including different hardware implementations, different programming languages, and many different algorithms.

- Common hardware used is most SLAM implementations:
    - Range measurement device (LiDAR-Lite in our case)
    - A robot that is capable of accurate odometry and movement
        - Odometry- the use of data from motion sensors to estimate change in position over time

- GOAL of SLAM is to process data collected from odometer and distance sensors and use it to track the position of the robot.
    - The data collected from the distance sensors will be used to create known features of the room, called landmarks
    - This data is passed through some type of filter (example uses an extended Kalman filter), which is responsible for updating where the robot thinks it is based on the landmarks.

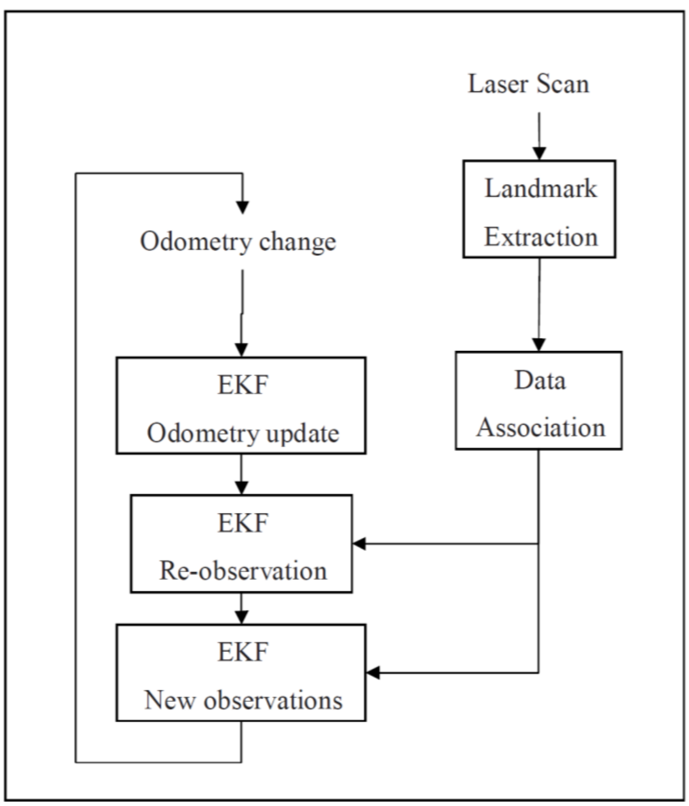

Figure 1 Overview of the SLAM process

- Landmarks are chosen based on the following criteria:
    - Landmarks should be re-observable, meaning that they should be able to be detected from different angle as the robot moves throughout the environment. This usually means landmarks are stationary.
    - Should have enough landmarks so that the robot will never be maneuvering without a visible landmark.
    - Should not have so many landmarks that the robot has difficulty determining which landmarks have been previously seen.
    - Examples include well defined corners, and objects with well-defined edges

- Types of landmark extraction:
    - Spike landmark- range of sensor scan differs by a large amount from one tick to the following tick. Example- legs of a table
    - RANSAC(Random Sampling Consensus)- method that extracts straight lines from the environment. Useful when scanning indoors because it is able to detect straight walls.

- Data association
    - Data association is the process of matching the data from previously observed landmarks to data currently being received from the sensors.
    - This process often times needs to compensate for:
        - Wrongly associate landmarks with previously seen landmarks
        - Observing something as a landmark but failing to ever see it again
        - Re-observing landmarks every time step
    - Solutions to these problems are proposed in full text of SLAM for Dummies – omitted from this document because the algorithms can be quite lengthy

- The Filter
    - Main purpose of filter(EKF in this example) is to estimate the position of the robot from the odometry and sensor data.
    - Overview of the process:
        - Update the current position estimate using the odometry data
        - Update estimated position by re-observing landmarks
        - Add new landmarks to the current position

- Math behind the algorithms
    - X- matrix- contains x-y positions of the robot and the landmarks and the rotational angle theta
    - Covariance matrix
        - Covariance- how strongly correlated two variates are

        - This matrix contains the covariance on the robot position, the covariance on the landmarks, the covariance between robot position and landmarks
    - The Kalman gain- Used to determine how much we ‘trust’ the landmarks and therefore how much we want to use the information we obtain from a landmark
    - If we know LiDAR reading is unreliable at a certain point compared to data from odometer, then the Kalman gain will be low. True for converse also.
    - This matrix contains 2 columns, left contains information about how much should be gained from the data in terms of range, the second contains information about how much should be gained from the data in terms of bearing.
        - Bearing- the direction of a landmark relative to the robot(measured in degrees or milliradians)

- Jacobian of the measurement model
    - Measurement model- shows how to compute an expected range and bearing of the data received from LiDAR(observed landmarks).

$$\begin{bmatrix} \text{range} \\ \text{bearing} \end{bmatrix} = \begin{bmatrix} \sqrt{(\lambda_x - x)^2 + (\lambda_y - y)^2} + v_r \\ \tan^{-1}\left(\dfrac{\lambda_y - y}{\lambda_x - x}\right) - \theta + v_\theta \end{bmatrix}$$

    -
        - Where λ_x is the x position of the landmark and x is the current estimated robot x position, λ_y is y position of the landmark and y is the current estimated robot y position

- More math but it is specific to the EKF and you may not need to understand it, just know how to implement it or any similar filter (such as exponential moving average).