

Simultaneous localization and mapping (SLAM) is a concept where, e.g. a robot, determine its location and makes a map of its surroundings. SLAM is useful for when it is hard to manually generate a map of the environment, or when GPS isn't good enough, GPS only offers accuracy within meters. A good application for SLAM is indoor robots.

Since the robot isn't given a map or information about the surroundings, the robot must rely on its sensors to map the area. A similar situation is when you find yourself kidnapped in complete darkness and must rely on your hands to figure out how the environment looks like. You cannot automatically interpret our surroundings and must make measurements, e.g. with our hands, and extract information from that to gain an understanding about the environment you are in. The main difference would be that robots that uses SLAM can be maneuvered by someone having knowledge about the environment. Therefore, the robot doesn't have to worry about crashing and can just move however it is commanded to. The command, e.g. travel forward, are given to the robot and with its sensors the robot calculates its new position, which can be done by relating the turn of a wheel to distance. However, despite giving the robot the command, the robot might not do exactly that. This is due to imperfect sensors and specially to slipping, e.g. the wheel turn but the robot doesn't move. This leads to uncertainty in its position which has to be considered.

SLAM uses features to build a map while being uncertain of its position. Features are special objects that can easily be distinguished. There are different types of features, e.g. QR-codes, or generate features from the images of the surroundings, then a feature might be a specific corner of a desk or a part of a tile on the floor. By using features, the robot has a better understanding where it is which in turn improves the mapping. Since the robot is uncertain in its position when first seeing the feature and due to uncertainty in the measurement, the feature is initialized with some uncertainty. As you can imagine it is much to keep track on, for a robot moving on a flat surface it has three values (x, y and direction) and for each new feature an additional two values (x and y) is added and all these values have their corresponding uncertainties. By moving around the robot sees the features repeatedly and can correct its position and hopefully, in the end, be quite certain. The more certain, the more precise the map will be.

The map is created by knowing its position and measure distances to objects, this can be done in many ways, one is to use Light Detection and Ranging (LIDAR) or compute the distances from images.

The first step of SLAM would be to initialize the map and uncertainties, since the map can be placed anywhere it is a good idea to place it such that the robot is in the origin with zero uncertainties. Next the robot moves around and when a feature is first detected the robot keeps track of its position along with the uncertainty. Later when this feature is detected again, it can be used to correct the position of the robot and hopefully lower its uncertainty. Imagine you walking around in the forest and you believe there will be a road straight ahead in about 500 meters but after 500 meters there is no road, but you find yourself on a hill with a view which shows the road further down the hill. You then correct yourself in your head and keep on walking. This is much like the robot does in SLAM, where the road being a feature, but instead of just updating its position the robot must update the map as well.