Chapter 8
Localization

Navigation by odometry (Sect. 5.4) is prone to errors and can only give an estimate of the real pose of the robot. Moreover, the further the robot moves, the larger the error in the estimation of the pose, especially in the heading. Odometry in a robot can be compared to walking with closed eyes, counting steps until we reach our destination. As with odometry, the further we walk, the more uncertain we are about our location. Even when counting steps, we should open our eyes from time to time to reduce the uncertainty of our location. For a robot, what does it mean to "count steps" and "open our eyes from time to time"? It means that for moving short distances odometry is good enough, but when moving longer distances the robot must determine its position relative to an external reference called a landmark. This process is called *localization*.

Section 8.1 starts with an activity that you can use to familiarize yourself with the relation between odometry and localization. Section 8.2 presents classical trigonometric techniques used by surveyors to determine the position of a location on earth by measuring angles and distances to positions whose location is known. Section 8.3 briefly surveys global positioning systems (GPS) which are now widely used for localization. There are environments where GPS is not effective: in buildings and when very precise positions are needed. Robots in these environments use probabilistic localization techniques that are described in Sects. 8.4–8.5.

8.1 Landmarks

Landmarks, such as lines on the ground or doors in a corridor can be detected and identified by the robot and used for localization. The following activity—which uses neither a computer nor a robot—will help you understand the importance of identifying landmarks to correct errors in odometry.

© The Author(s) 2018

M. Ben-Ari and F. Mondada, *Elements of Robotics*,
https://doi.org/10.1007/978-3-319-62533-1_8

Activity 8.1: Play the landmark game

- Choose a path in your home that requires you to pass several doors, for example, from the bed in your room through a hallway to a sofa in the living room and then back.
- Close your eyes and walk along the path applying the following rules:
  - You get 30 points at the beginning of the activity.
  - From time to time you can open your eyes for one second; this action costs 1 point.
  - If you touch a wall it costs 10 points.
- How many points do you have when you complete the path?
- Is it better to open the eyes frequently or to walk the path without ever opening your eyes?

8.2 Determining Position from Objects Whose Position Is Known

In this section we describe two methods which a robot can use to determine its position by measuring angles and distances to an object whose position is known. The first method assumes that the robot can measure the distance to the object and its *azimuth*, the angle of the object relative to north. The second method measures the angles to the object from two different positions. Both methods use trigonometry to compute the coordinates of the robot relative to the object. If the absolute coordinates (x_0, y_0) of the object are known, the absolute coordinates of the robot can then be easily computed.

8.2.1 Determining Position from an Angle and a Distance

Figure 8.1 shows the geometry of a robot relative to an object. In the diagram the object is denoted by the large dot placed at the origin (x_0, y_0) of a coordinate system. The azimuth of the robot θ is the angle between north and the forward direction of the robot; it can be measured by a compass. A laser scanner is used to measure the distance s to the object and the angle ϕ between the forward direction of the robot and the object. The relative coordinates Δx and Δy can be determined by simple trigonometry:

$$\Delta x = s \sin(\theta - \phi), \quad \Delta y = s \cos(\theta - \phi)\,.$$

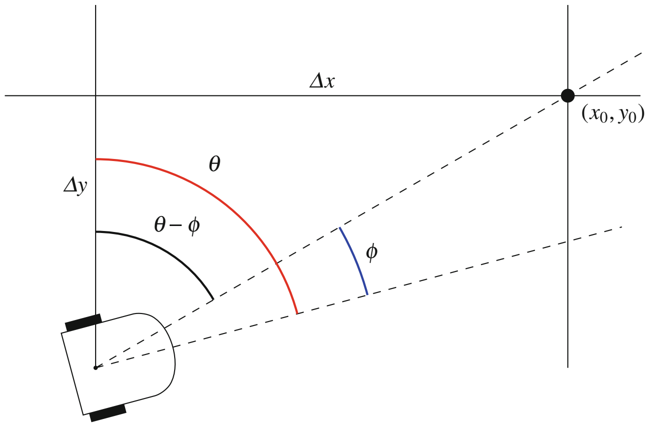


Fig. 8.1 Determining position from an angle and a distance

From the known absolute coordinates of the object (x_0, y_0), the coordinates of the robot can be determined.

Activity 8.2: Determining position from an angle and a distance

- Implement the algorithm.
- To measure the azimuth place the robot lined up with a edge of the table and call it north. Measure the angle to the object using several horizontal sensors or using one sensor and rotating either the sensor or the robot.
- The distance and the angle are measured from the position where the sensor is mounted, which is not necessarily the center of the robot. You may have to make corrections for this.

8.2.2 Determining Position by Triangulation

Triangulation is used for determining coordinates when it is difficult or impossible to measure distances. It was widely used in surveying before lasers were available, because it was impossible to measure long distances accurately. The principle of triangulation is that from two angles of a triangle and the length of the included side, you can compute the lengths of the other sides. Once these are known, the relative position of a distant object can be computed.

Figure 8.2 shows the robot measuring the angles α and β to the object from two positions separated by a distance c. If the two positions are close, the distance can be measured using a tape measure. Alternatively, the distance can be measured by

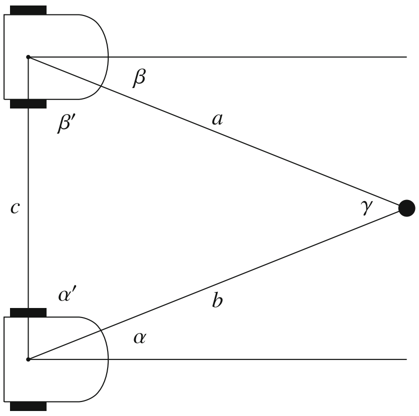


Fig. 8.2 Triangulation

odometry as the robot moves from one position to another, though this may be less accurate. In surveying, if the coordinates of the two positions are known, the distance between them can be computed and used to determine the coordinates of the object.

The lengths a and b are computed using the *law of sines*:

$$\frac{a}{\sin\alpha'} = \frac{b}{\sin\beta'} = \frac{c}{\sin\gamma}\,,$$

where $\alpha' = 90^\circ - \alpha,\ \beta' = 90^\circ - \beta$ are the interior angles of the triangle. To use the law, we need c, which has been measured, and γ, which is:

$$\gamma = 180^\circ - \alpha' - \beta' = 180^\circ - (90^\circ - \alpha) - (90^\circ - \beta) = \alpha + \beta\,.$$

From the law of sines:

$$b = \frac{c\sin\beta'}{\sin\gamma} = \frac{c\sin(90^\circ - \beta)}{\sin(\alpha+\beta)} = \frac{c\cos\beta}{\sin(\alpha+\beta)}\,.$$

A similar computation gives a.

Activity 8.3: Determining position by triangulation

- Implement triangulation.
- Initially, perform a measurement of the angle at one position and then pick the robot up and move it to a new position for the second measurement, carefully measuring the distance c.
- Alternatively, cause the robot move by itself from the first position to the second, calculating c by odometry.

8.3 Global Positioning System

In recent years, the determination of location has been made easier and more accurate with the introduction of the *Global Positioning System (GPS)*.[1] GPS navigation is based upon orbiting satellites. Each satellite knows its precise position in space and its local time. The position is sent to the satellite by ground stations and the time is measured by a highly accurate atomic clock on the satellite.

A GPS receiver must be able to receive data from four satellites. For this reason, a large number of satellites (24–32) is needed so that there is always a line-of-sight between any location and at least four satellites. From the time signals sent by a satellite, the distances from the satellites to the receiver can be computed by multiplying the times of travel by the speed of light. These distances and the known locations of the satellites enable the computation of the three-dimensional position of the receiver: latitude, longitude and elevation.

The advantage of GPS navigation is that it is accurate and available anywhere with no additional equipment beyond an electronic component that is so small and inexpensive that it is found in every smartphone. There are two problems with GPS navigation:

- The position error is roughly 10 m. While this is sufficient to navigate your car to choose the correct the road at an intersection, it is not sufficient to perform tasks that need higher accuracy, for example, parking your car.
- GPS signals are not strong enough for indoor navigation and are subject to interference in dense urban environments.

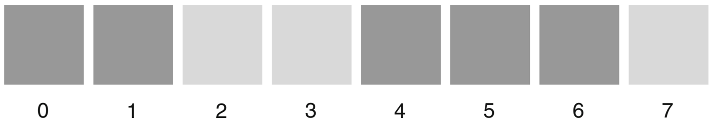
Relativity and GPS

You have certainly heard of Albert Einstein's theory of relativity and probably considered it to be an esoteric theory of interest only to physicists. However, Einstein's theories are used in the GPS computations! According to the special theory of relativity, the clocks on the satellites run *slower* than they do on the Earth (by 7.2 microseconds per day), because the satellites are moving fast relative to the Earth. According to the general theory of relativity, the clocks run *faster* (by 45.9 microseconds per day), because the force of the Earth's gravity is smaller at the distant satellite than it is for us on the surface. The two effects do not cancel out and a correction factor is used when broadcasting the time signals.

8.4 Probabilistic Localization

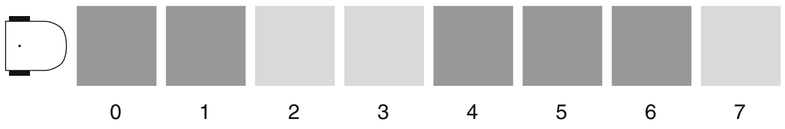
Consider a robot that is navigating within a known environment for which it has a *map*. The following map shows a wall with five doors (dark gray) and three areas where there is no door (light gray):

[1]The generic term is *global navigation satellite system (GNSS)* since GPS refers to the specific system operated by the United States. Similar systems are operated by the European Union (Galileo), Russia (GLONASS) and China (BeiDou), but GPS is frequently used to refer to all such systems and we do so here.

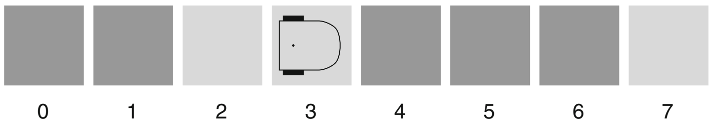


For clarity the doors and walls are drawn as if they are on the floor and the robot moves over them, measuring intensity with ground sensors.

The task of the robot is to enter a specific door, say the one at position 4. But how can the robot know where it is? By odometry, the robot can determine its current position given a known starting position. For example, if the robot is at the left end of the wall:



it knows that it has to move five times the width of each door, while if the robot is at the following position:



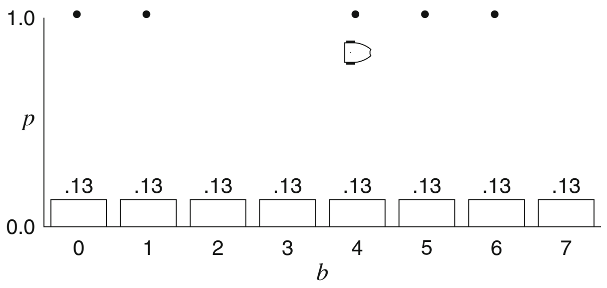
the required door is the next one to the right. Because of errors in odometry, it is quite likely that as time goes by the robot will get lost. In this section, we implement a one-dimensional version of a probabilistic *Markov localization algorithm* that takes into account uncertainty in the sensors and in the robot's motion, and returns the most probable locations of the robot.

Appendix B.1 contains a short tutorial on conditional probability and Bayes rule, including an example of the details of the calculations of uncertainty.

8.4.1 Sensing Increases Certainty

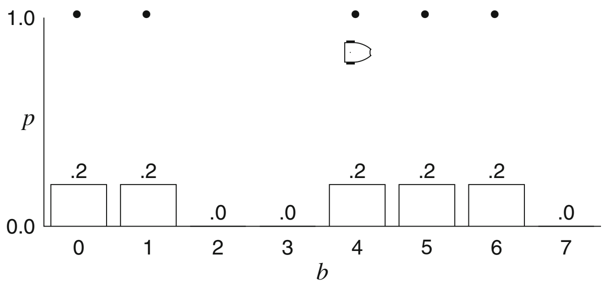
Consider a robot in the above environment of walls and doors that has no information as to its location. The robot assigns a probability to the eight positions where it might be located. Initially, it has no idea where it is, so each position will be assigned the probability $b[i] = 1.0/8 = 0.125 \approx 0.13$, where b is called the *belief array*[2]:

[2]All probabilities will be rounded to two decimal digits for display.

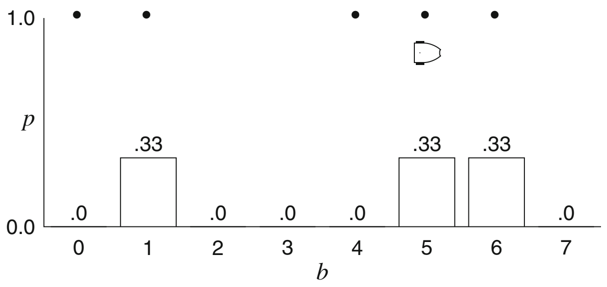


In the plots the dots denote the positions of the doors and a small icon indicates the actual position of the robot which is facing right.

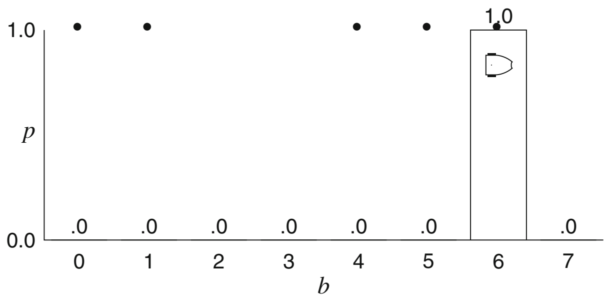
Suppose now that the robot's sensors detect a dark gray area. Its uncertainty is reduced, because it knows that it must be in front of one of the five doors. The belief array shows 0.2 for each of the doors and 0.0 for each of the walls:



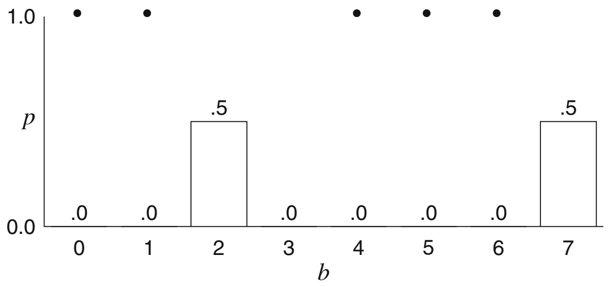
Next the robot moves forwards and again senses a dark gray area. There are now only three possibilities: it was at position 0 and moved to 1, it was at 4 and moved to 5, or it was at 5 and moved to 6. If the robot's initial position were 1 or 6, after moving right it would no longer detect a dark gray area, so it could not have been there. The probability is now 0.33 for each of the three positions 1, 4, 5:



After the next step of the robot, if it again detects a door, it is without doubt at position 6:



If the robot did not detect a door, it is either at position 2 or position 7:



The robot maintains a belief array and integrates new data when it detects the presence or absence of a door. As time goes on, the uncertainty decreases: the robot knows with greater certainty where it is actually located. In this example, eventually the robot knows its position in front of door 6 with complete certainty or it has reduced its uncertainty to one of the two positions 2, 7.

8.4.2 Uncertainty in Sensing

The values returned by the robot's sensors reflect the intensity of the light reflected by the gray colors of the doors and walls. If the difference in the color of a dark gray door and a light gray wall is not very great, the robot may occasionally detect a dark gray door as a light gray wall, or conversely. This can occur due to changes in the ambient lighting or to errors in the sensors themselves. It follows that the robot cannot distinguish between the two with complete certainty.

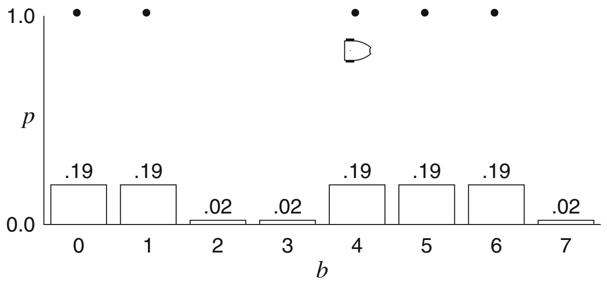
We model this aspect of the world by assigning probabilities to the detection. If the robot senses dark gray, we specify that the probability is 0.9 that it has correctly detected a door and 0.1 that it has mistakenly detected a wall where there was in fact a door. Conversely, if it senses light gray, the probability is 0.9 that it has correctly detected a wall and 0.1 that it has mistakenly detected a door where there was a wall.

Table 8.1 Localization with uncertainty in sensing, sensor = after multiplying by the sensor uncertainty, norm = after normalization, right = after moving right one position

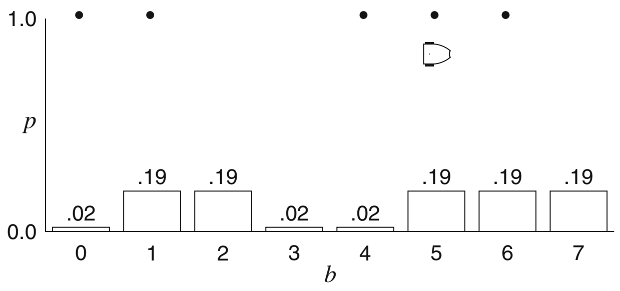
Position door?	0 ●	1 ●	2	3	4 ●	5 ●	6 ●	7
Initial	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
Sensor	0.11	0.11	0.01	0.01	0.11	0.11	0.11	0.01
Norm	0.19	0.19	0.02	0.02	0.19	0.19	0.19	0.02
Right	0.02	0.19	0.19	0.02	0.02	0.19	0.19	0.19
Sensor	0.02	0.17	0.02	0.00	0.02	0.17	0.17	0.02
Norm	0.03	0.29	0.03	0.00	0.03	0.29	0.29	0.03
Right	0.03	0.03	0.29	0.03	0.00	0.03	0.29	0.29
Sensor	0.03	0.03	0.03	0.00	0.00	0.03	0.26	0.03
Norm	0.07	0.07	0.07	0.01	0.01	0.07	0.63	0.07

We continue to display the computations in graphs but you may find it easier to follow them in Table 8.1. Each row represents the belief array of the robot following the action written in the first column.

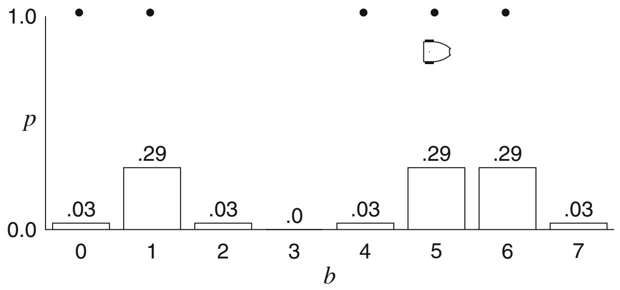
Initially, after sensing dark gray at a position where there is a door, we only know with probability $0.125 \times 0.9 = 0.1125$ that a door has been correctly detected; however, there is still a $0.125 \times 0.1 = 0.0125$ probability that it has mistakenly detected a wall. After normalizing (Appendix B.2), the belief array is:



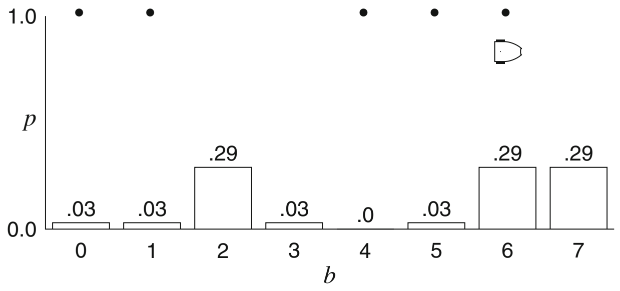
What happens when the robot moves one position to the right? Its belief array must also move one position to the right. For example, the probability 0.19 that the robot was at position 1 now becomes the probability that it is at position 2. Similarly, the probability 0.02 that the robot was at position 3 now becomes the probability that it is at position 4. The probability is now 0 that the robot is at position 0 and the probability b_7 becomes b_8, so the indices become 1–8 instead of 0–7. To simplify the computations and the diagrams in the example, the indices 0–7 are retained and the value of b_8 is stored in b_0 as if the map were cyclic. The belief array after the robot moves right is:



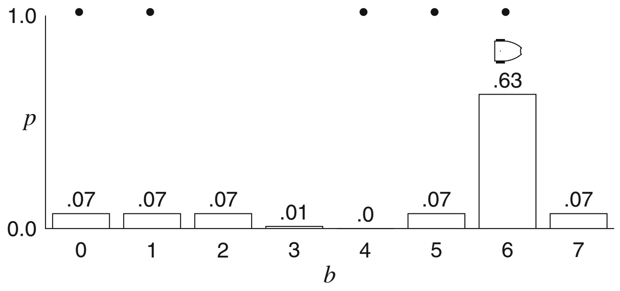
If the robot again senses dark gray, the probability of being at positions 1, 5 or 6 should increase. Computing the probabilities and normalizing gives:



Now the robot moves right again:



and senses a third dark gray area. The belief array becomes:



Not surprisingly, the robot is almost certainly at position 6.

Activity 8.4: Localization with uncertainty in the sensors

- Implement probabilistic localization with uncertainty in the sensor.
- How does the behavior of the algorithm change when the uncertainty is changed?
- Run the algorithm for different starting positions of the robot.

8.5 Uncertainty in Motion

As well as uncertainty in the sensors, robots are subject to uncertainty in their motion. We can ask the robot to move one position to the right, but it might move two positions, or it might move very little and remain in its current position. Let us modify the algorithm to take this uncertainty into account.

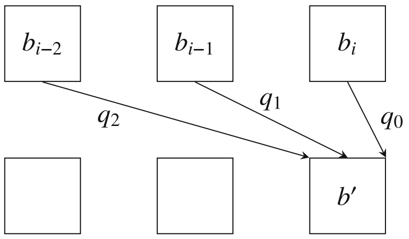
Let b be the belief array. The belief array is updated using the formula:

$$b'_i = p_i \, b_i \,,$$

where b'_i is the new value of b_i and p_i is the probability of detecting a door (in the example, p_i is 0.9 for $i = 0, 1, 4, 5, 6$ and p_i is 0.1 for $i = 2, 3, 7$). If the motion is certain the robot moves one position to the right, but with uncertain motion, the following computation takes into account the probabilities q_j that the robot actually moves $j = 0, 1, 2$ positions:

$$b'_i = p_i \,(b_{i-2}\, q_2 + b_{i-1}\, q_1 + b_i\, q_0)\,,$$

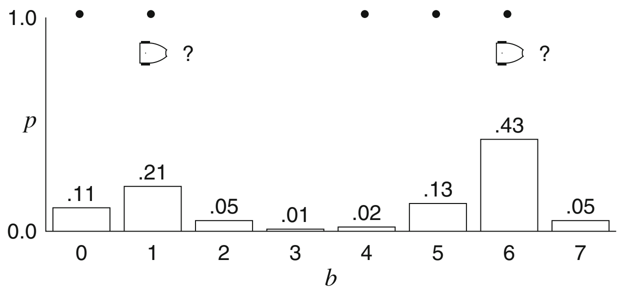
as shown in the following diagram:



It is highly likely that the robot will move correctly, so reasonable values are $q_1 = 0.8$ and $q_0 = q_2 = 0.1$. With these values for the uncertainty of the motion and the previous values for p_i, the calculation of the belief array after three moves is shown in Table 8.2 and its final value is shown in the following diagram:

Table 8.2 Localization with uncertainty in sensing and motion, sensor = after multiplying by the sensor uncertainty, norm = after normalization, right = after moving right one position

Position door?	0 ●	1 ●	2	3	4 ●	5 ●	6 ●	7
Initial	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
Sensor	0.11	0.11	0.01	0.01	0.11	0.11	0.11	0.01
Norm	0.19	0.19	0.02	0.02	0.19	0.19	0.19	0.02
Right	0.05	0.19	0.17	0.04	0.04	0.17	0.19	0.17
Sensor	0.05	0.17	0.02	0.00	0.03	0.15	0.17	0.02
Norm	0.08	0.27	0.03	0.01	0.06	0.25	0.28	0.03
Right	0.06	0.12	0.23	0.05	0.01	0.07	0.23	0.25
Sensor	0.05	0.10	0.02	0.01	0.01	0.06	0.21	0.02
Norm	0.11	0.21	0.05	0.01	0.02	0.13	0.43	0.05



The robot is likely at position 6, but we are less certain because the probability is only 0.43 instead of 0.63. There is a non-negligible probability of 0.21 that the robot is at position 1.

Activity 8.5: Localization with uncertainty in the motion

- Implement probabilistic localization with uncertainty in the computation of the motor power.
- How does the behavior of the algorithm change when the uncertainty is changed?
- Run the algorithm for different starting positions of the robot.

8.6 Summary

Odometry provides an estimation of the position of a robot. A robot can use surveying techniques to compute its position relative to an object of known position. GPS gives excellent data on location, however, it may not be accurate enough and interference with reception from the satellites limits its use in indoor environments. If there are multiple known objects that the robot can sense and if it has a map of its environment, it can use probabilistic localization to estimate its position with high probability, although the probability will be reduced if there is a lot of uncertainty in the sensors or in the motion of the robot.

8.7 Further Reading

Probabilistic methods in robotics are treated in depth in [1]. There is a wealth of information on GPS at http://www.gps.gov. An implementation of probabilistic localization using the Thymio educational robot is described in [2].

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
2. Wang, S., Colas, F., Liu, M., Mondada, F., Magnenat, S.: Localization of inexpensive robots with low-bandwidth sensors. In: Distributed Autonomous Robotic Systems (DARS). IEEE (2016)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.