Chapter 5
Robotic Motion and Odometry

The robotics algorithms in the previous chapters react to data from their sensors by changing the speed and direction of their motion, but the changes were not quantitative. We didn't require the robots to move twice as fast or to turn 90° to the right. Robots in the real world have to move to specific locations and may have engineering constraints on how fast or slow they can move or turn. This chapter presents the mathematics of robotic motion.

Sections 5.1 and 5.2 review the concepts of distance, time, velocity and acceleration that should be familiar from introductory physics. The physics of motion is usually taught using calculus, but a computer cannot deal with continuous functions; instead, discrete approximations must be used as described in Sect. 5.3.

Sections 5.4–5.6 present *odometry*, the fundamental algorithm for computing robotic motion. An approximation of the location of a robot can be obtained by repeatedly computing the distance moved and the change direction from the velocity of the wheels in a short period of time. Unfortunately, odometry is subject to serious errors as shown in Sect. 5.7. It is important to understand that errors in direction are much more significant than errors in distance.

In the simplest implementation, the speed of the wheels of a robot is assumed to be proportional to the power applied by the motors. Section 5.8 shows how the accuracy of odometry can be improved by using *wheel encoders*, which measure the actual number of revolutions of the wheels.

Section 5.9 presents an overview of *inertial navigation*, which is a sophisticated form of odometry based upon measuring linear and angular acceleration and then integrating to obtain velocity and position. The sensors for inertial navigation (accelerometers and gyroscopes) were once very expensive, limiting its application to aircraft and rockets, but new technology called *microelectromechanical systems* has made it possible to build robots with inertial navigation.

M. Ben-Ari and F. Mondada, *Elements of Robotics*,
https://doi.org/10.1007/978-3-319-62533-1_5

Cars cannot move up and down unlike helicopters and submarines which have greater freedom of movement. This is expressed in the concept *degrees of freedom (DOF)* which is the subject Sect. 5.10. Section 5.11 discusses the relation between the DOF and number of *actuators* (motors) in a robotics systems.

The number of DOF of a system does not mean that a system such as a vehicle can move freely in all those directions. A car can move to any point in the plane and orient itself in any direction, but it cannot move sideways, so a difficult maneuver is needed during parallel parking. This is due to the difference between the DOF and the *degrees of mobility (DOM)*, a subject explored in Sect. 5.12, along with the concept of *holonomic motion* that relates DOF and DOM.

5.1 Distance, Velocity and Time

Suppose that a robot moves with a constant *velocity* of 10 cm/s for a period of *time* of 5 s.[1] The *distance* it moves is 50 cm. In general, if a robot moves at a constant velocity v for a period of time t, the distance it moves is $s = vt$. When power is applied to the motors it causes the wheels to rotate, which in turn causes the robot to move at some velocity. However, we cannot specify that a certain power causes a certain velocity:

- No two electrical or mechanical components are ever precisely identical. A motor is composed of magnets and electrical wiring whose interaction causes a mechanical shaft to rotate. Small differences in the properties of the magnet and wire, as well as small differences in the size and weight of the shaft, can cause the shafts of two motors to rotate at slightly different speeds for the same amount of power.
- The environment affects the velocity of a robot. Too little friction (ice) or too much friction (mud) can cause a robot to move slower in comparison with its movement on a dry paved surface.
- External forces can affect the velocity of a robot. It needs more power to sustain a specific velocity when moving uphill and less power when moving downhill, because the force of gravity decreases and increases the velocity. Riding a bicycle at a constant velocity into the wind demands more effort than riding with the wind, and a cross-wind makes the relation between power and velocity even more complicated.

Since $s = vt$ it is sufficient to measure any two of these quantities in order to compute the third. If we measure distance and time, we can compute the velocity as $v = s/t$. Relatively short distances (up to several meters) can be measured accurately (to within 1 cm) using a ruler or a tape measure. The stopwatch application on a smartphone can measure time accurately (hundredths of a second).

[1] Velocity is *speed* in a *direction*. A robot can be moving 10 cm/s forwards or backwards; in both cases, the speed is the same but the velocity is different.

Activity 5.1: Velocity over a fixed distance

- Write a program that sets your robot to a constant forward power setting.
- Mark two lines 1 m apart on the floor. Use a stopwatch to measure the time it takes the robot to move between the lines. Compute the velocity of the robot. Run the program ten times and record the velocities. Do the velocities vary?
- Place the robot on the floor and run it for 5 s. Measure the distance that it moves. Compute the velocity. Run the program ten times and record the velocities. Do the velocities vary?
- Which method gives more precise results?
- Repeat this experiment on different surfaces and discuss the results.

Activity 5.1 shows that for a constant power setting the velocity of a robot can vary significantly. To accurately navigate within an environment, a robot needs to sense objects in its environment, such as walls, marks on the floor and objects.

5.2 Acceleration as Change in Velocity

Activity 5.1 specified constant power settings and thus the velocity of the robot will be (more or less) constant. What happens when the velocity is varied?

Activity 5.2: Change of velocity

- Run the first program from Activity 5.1 varying the distance between the marks: 0.25, 0.5, 1, 1.5, 2 m. For each distance, run the program several times and take the average of the computed velocities. Are the velocities the same for each distance?
- To improve the accuracy of the measurement, place marks on the floor at these distances and use the robot's timer to record the times at which the marks are detected.

In Activity 5.2, you will find that for the longer distances the velocities will be close to each other, but for the shorter distances the velocities will differ considerably. The reason is that the formula $v = s/t$ assumes that the velocity is constant over the entire distance. In reality, a vehicle must *accelerate*—change its velocity—in order to go from standing still to a constant velocity. Similarly, a vehicle must *decelerate* in order to stop.

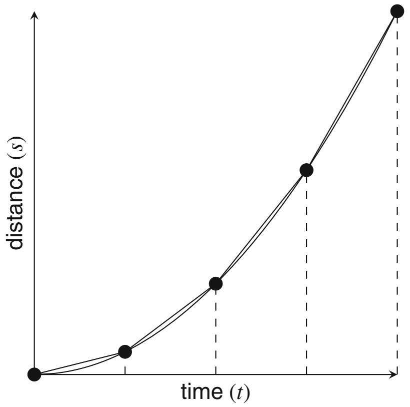
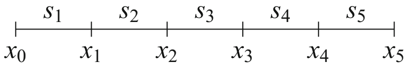


Fig. 5.1 An accelerating robot: distance increase as the square of time

To get a true picture of the motion of a robot, we need to divide its motion into small segments $s_1, s_2, \ldots$:



and measure the distance and time for each segment individually. Then, we can compute the velocities for each segment. In symbols, if we denote the length of the segment s_i by $\Delta s_i = x_{i+1} - x_i$ and the time it takes the robot to cross segment s_i by $\Delta t_i = t_{i+1} - t_i$, then v_i, the velocity in segment s_i is given by:

$$v_i = \frac{\Delta s_i}{\Delta t_i}\,.$$

Figure 5.1 is a graph of distance versus time for an accelerating robot. The time axis has been divided into segments and the slopes $\dfrac{\Delta s_i}{\Delta t_i}$ show the average velocity in each segment which increases with time.

Acceleration is defined as the change in velocity over a period of time:

$$a_i = \frac{\Delta v_i}{\Delta t_i}\,.$$

When the power setting of the robot is set to a fixed value, the force applied to the robot is constant and we expect that the acceleration remains constant, increasing the velocity. However, at a certain point the acceleration is reduced to zero, meaning that the velocity no longer increases, because the power applied to the wheels is just sufficient to overcome the friction of the road and the wind resistance.

Let us see what happens if the power setting is increased with time.

Activity 5.3: Acceleration

- Write a program that causes the robot to accelerate by increasing the power setting periodically. For example, start the robot at power 20 and increase to 40 after 1 s, then to 60 after 2 s, to 80 after 3 s, and finally to 100 after 4 s.
- Place the robot on the track and run the program.
- Record the distances between each change of the power setting. Compute and plot the velocities in each of these segments.

5.3 From Segments to Continuous Motion

As the size of the segments becomes smaller, we obtain the instantaneous velocity of the robot at a single point in time, expressed as a derivative:

$$v(t) = \frac{ds(t)}{dt}\,.$$

Similarly, the instantaneous acceleration of the robot is defined as:

$$a(t) = \frac{dv(t)}{dt}\,.$$

For constant acceleration the velocity can be obtained by integrating the derivative:

$$v(t) = \int a\,dt = a\int dt = at\,,$$

and then the distance can be obtained by integrating again:

$$s(t) = \int v(t)dt = \int a\,t\,dt = \frac{at^2}{2}\,.$$

Example An average car accelerates from 0 to 100 km/h in about 10 s. First, we convert units from km/h to m/s:

$$v_{max} = 100\,\text{km/h} = \frac{100 \cdot 1000}{60 \cdot 60}\,\text{m/s} = 27.8\,\text{m/s}\,.$$

Assuming constant acceleration, $v_{max} = 27.8 = at = 10a$, so the acceleration is 2.78 m/s^2 (read, 2.78 meters per second per second, that is, every second the speed increases by 2.78 meters per second). The distance the car moves in 10 s is:

$$s(10) = \frac{at^2}{2} = \frac{2.78 \cdot 10^2}{2} = 139\,\text{m}\,.$$

Activity 5.4: Computing distance when accelerating

- For various vehicles (racing cars, motorcycles) look up the time required to accelerate from 0 to 100 km/h. Compute the distance moved.
- Assume that the acceleration of a vehicle increases linearly, that is, $a = kt$ for a constant k. What are $v(t)$ and $s(t)$?
- For several values of k and t, compute the final velocities and distances.

Activity 5.5: Measuring motion at constant acceleration

- Write a program that applies the maximum power setting to a robot.
- Place the robot on a surface and run the program.
- When the robot seems to have reached full speed record the time from the start of the run.
- Compare the measured distance to $s = at^2/2$ (Fig. 5.2b).
- Run again and measure the distances at fixed intervals of time. Compute the speeds from the distances divided by the time and compare to $v = at$ (Fig. 5.2a).
- In some robots you can set a target speed and read the actual speed. If your robot can do this, compare the measured speeds with the computed speeds.

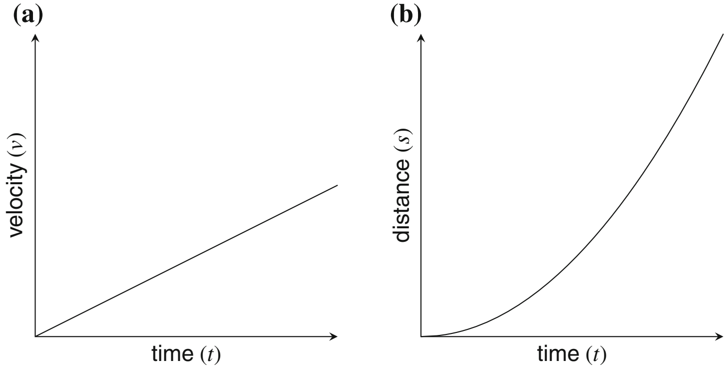


Fig. 5.2 **a** Velocity for constant acceleration. **b** Distance for constant acceleration

5.4 Navigation by Odometry

Suppose that you are in a car and your navigation system issues the following instruction: “In 700 m turn right.” Now your task is very simple: Make observations of your car’s odometer which measures how far you have traveled. When its value approaches 700 m beyond its initial reading, look for a street on the right. An odometer in a car measures speed and time, and multiplies the two values to compute the distance traveled.

Odometry—the measurement of distance—is a fundamental method used by robots for navigation. Measuring time is easy using the internal clock of the embedded computer. Measuring speed is more difficult: in some educational robots wheel encoders are used to count the rotations of the wheels (Sect. 5.8), while in others speed is estimated from properties of the motors. From the distance moved $s = vt$, the new position of the robot can be computed. In one dimension, the computation is trivial, but it becomes a bit more complex when the motion involves turns. This section presents the computation of distance by odometry, first for a robot moving linearly and then for a robot making a turn.

Section 5.7 shows how errors in heading are more serious than errors in distance.

A disadvantage of odometry (with or without wheel encoders) is that the measurements are indirect, relating the power of the motors or the motion of the wheels to changes in the robot’s position. This can be error-prone since the relation between motor speed and wheel rotation can be very nonlinear and vary with time. Furthermore, wheels can slip and skid so there may be errors in relating the motion of the wheels to the motion of the robot. Improved estimates of position can be obtained by using an inertial navigation system, which directly measures acceleration and angular velocity that can be used to determine the robot’s position (Sect. 5.9).

Odometry is a form of *localization*: the robot must determine its position in the environment. In odometry we determine position by measuring the change from the robot’s known initial position, while localization (Chap. 8) refers to the determination of the position of a robot relative to the known positions of other objects such as landmarks or beacons.

5.5 Linear Odometry

Before studying the mathematics of odometry you should try the following Activity:

Activity 5.6: Distance from speed and time

- Run the robot at a constant power setting for a specific period of time and measure the distance moved.

- Repeat the measurement several times. Is the distance constant? If not, how much does it vary as a percentage of the distance?
- Repeat the measurement several times for different power settings. Is the distance measured linear in the power setting? Does the *variation* in the distance measurement on multiple runs depend on the power setting?
- Repeat the measurement for a fixed power setting but for different periods of time and analyze the results.

When a relation between motor power and velocity v has been determined, the robot can compute the distance moved by $s = vt$. If it starts at position $(0, 0)$ and moves straight along the x-axis, then after t seconds its new position is $(vt, 0)$.

This activity should demonstrate that it is possible to measure distance by odometry with reasonable precision and accuracy. A self-driving car can use odometry to determine its position so that it doesn't have to analyze its sensor data continuously to check if the required street has been reached. Given the uncertainties of motion and of the road, the car should not depend only on odometry to decide when to turn, but the error will not be large and the sensor data can be analyzed to detect the turn when odometry indicates that the car is in the vicinity of the intersection.

Activity 5.6 asked you to measure the distance moved in one dimension. Three items of information need to be computed if the motion is in two dimensions: the robot's *position* (x, y) relative to a fixed origin and its *heading* θ, the direction in which the robot is pointing (Fig. 5.3). The triple (x, y, θ) is called the *pose* of the robot. If the robot starts at the origin $(0, 0)$ and moves in a straight line at angle θ with velocity v for time t, the distance moved is $s = vt$. Its new position (x, y) is:

$$
\begin{aligned}
x &= vt \cos\theta \\
y &= vt \sin\theta \,.
\end{aligned}
$$

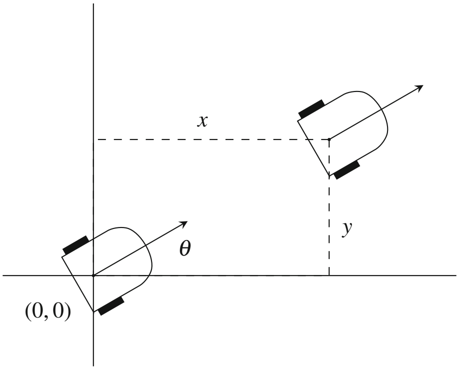


Fig. 5.3 Position and heading

5.6 Odometry with Turns

Suppose that the robot turns slightly left because the right wheel moves faster than the left wheel (Fig. 5.4). In the figure, the robot is facing towards the top of the page; the blue dot is the left wheel, the red dot is the right wheel, and the black dot is the center of the robot which is halfway between the wheels. The *baseline* b is the distance between the wheels, and d_l, d_r, d_c represent the distances moved by the two wheels and the center when the robot turns. We want to compute the new position and heading of the robot.

We can measure d_l and d_r, the distances moved by the two wheels using the method described in Activity 5.6: relating motor power to rotational speed and then multiplying by time. Alternatively, we can use the number of rotations counted by the wheel encoders. If the radius of a wheel is R and the rotational speeds of the left and right wheels are ω_l, ω_r revolutions per second, respectively, then after t seconds the wheel has moved:

$$d_i = 2\pi R\omega_i t, \quad i = l, r\,. \tag{5.1}$$

The task is to determine the new pose of the robot after the wheels have moved these distances.

Figure 5.4 shows the robot initially at pose (x, y, ϕ), where the robot is facing north ($\phi = \pi/2$). After turning θ radians, what is the new pose (x', y', ϕ')? Clearly, the heading of the robot is now $\phi' = \phi + \theta$, but we also have to compute x', y'.

The length of an arc of angle θ radians is given by its fraction of the circumference of the circle: $2\pi r\,(\theta/2\pi) = \theta r$. For small angles, the distances d_l, d_c, d_r are approximately equal to the length of the corresponding arcs, so we have:

$$\theta = d_l/r_l = d_c/r_c = d_r/r_r\,, \tag{5.2}$$

where r_l, r_r, r_c are the distances from P, the origin of the turn.

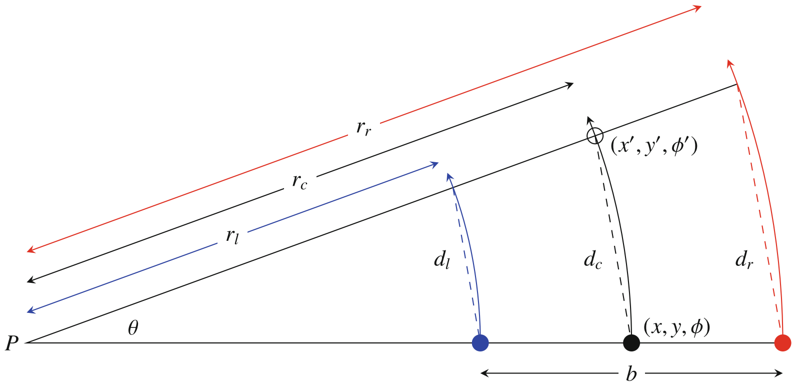


Fig. 5.4 Geometry of a left turn by a robot with two wheels

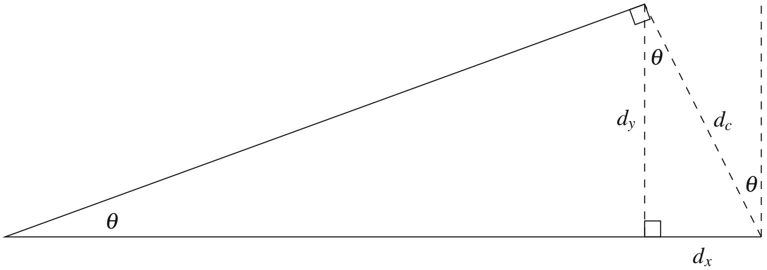


Fig. 5.5 Change in heading

The distances d_l and d_r are obtained from the rotations of the wheels (Eq. 5.1) and the baseline b is a fixed physical measurement of the robot. From Eq. 5.2, the angle θ can be computed:

$$\begin{aligned}\theta r_r &= d_r\\ \theta r_l &= d_l\\ \theta r_r - \theta r_l &= d_r - d_l\\ \theta &= (d_r - d_l)/(r_r - r_l)\\ \theta &= (d_r - d_l)/b\,.\end{aligned}$$

The center is halfway between the wheels $r_c = (r_l + r_r)/2$, so again by Eq. 5.2:

$$\begin{aligned}d_c &= \theta r_c\\ &= \theta\left(\frac{r_l + r_r}{2}\right)\\ &= \frac{\theta}{2}\left(\frac{d_l}{\theta} + \frac{d_r}{\theta}\right)\\ &= \frac{d_l + d_r}{2}\,.\end{aligned}$$

If the distance moved is small, the line labeled d_c is approximately perpendicular to the radius through the final position of the robot. By similar triangles, we see that θ is the change in the heading of the robot (Fig. 5.5). By trigonometry[2]:

$$\begin{aligned}dx &= -d_c \sin\theta\\ dy &= d_c \cos\theta\,,\end{aligned}$$

[2] You were probably expecting cos for dx and sin for dy. That would be the case if the robot were facing along the x axis. However, the initial pose is $\phi = \pi/2$ and we have $\sin(\theta + \pi/2) = \cos\theta$ and $\cos(\theta + \pi/2) = -\sin\theta$.

so the pose of the robot after the turn is:

$$(x', y', \phi') = (-d_c \sin\theta, d_c \cos\theta, \phi + \theta)\,.$$

The formulas show how to compute the changes dx, dy and θ when the robot moves a short distance. To compute odometry over longer distances, this computation must be done frequently. There are two reasons why the intervals between the computations must be short: (a) the assumption of constant speed holds only for short distances, and (b) the trigonometric calculation is simplified by assuming that the distance moved is short.

Activity 5.7: Odometry in two dimensions

- Write a program that causes the robot to make a gentle left turn for a specific period of time.
- Compute the pose $(-d_c \sin\theta, d_c \cos\theta, \theta)$ and compare the result with the values measured using a ruler and a protractor. Run the program several times and see if the measurements are consistent.
- Run the program for different periods of time. How does this affect the accuracy and precision of the odometry computation?

5.7 Errors in Odometry

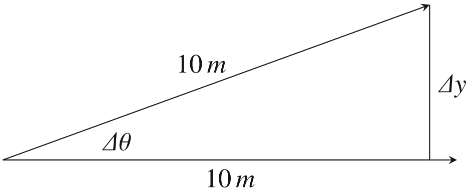
We have already noted that odometry is not accurate because inconsistent measurements and irregularities in the surface can cause errors. In this section we show that even small changes in the direction of the robot's movement can cause errors that are much larger than those caused by changes in its linear motion.

To simplify the presentation, let us assume that a robot is to move 10 m from the origin of a coordinate system along the x-axis and then check its surroundings for a specific object. What is the effect of an error of *up to p%*? If the error is in the measurement of x, the distance moved, then Δx, the error in x is:

$$\Delta x \leq \pm 10 \cdot \frac{p}{100} = \pm \frac{p}{10}\ \mathrm{m}\,,$$

where the value is negative or positive because the robot could move up to $p\%$ before or after the intended distance.

Suppose now that there is an error $p\%$ in the *heading* of the robot, and, for simplicity, assume that there is no error in the distance moved. The geometry is:



The robot intended to move 10 m along the x-axis, but instead it moved slightly to the left at an angle of $\Delta\theta$. Let us compute the left-right deviation Δy. By trigonometry, $\Delta y = 10 \sin \Delta\theta$. An error of $p\%$ in heading is:

$$\Delta\theta = 360 \cdot \frac{p}{100} = (3.6p)^\circ ,$$

so the left-right deviation is:

$$\Delta y \leq \pm 10 \sin(3.6p) .$$

The following tables compare the difference between a linear error of $p\%$ (left) and an error in heading of $p\%$ (right):

$p\%$	Δx (m)
1	0.1
2	0.2
5	0.5
10	1.00

$p\%$	$\Delta\theta$ (°)	$\sin \Delta\theta$	Δy (m)
1	3.6	0.063	0.63
2	7.2	0.125	1.25
5	18.0	0.309	3.09
10	36.0	0.588	5.88

For a very small error like 2%, the distance error after moving 10 m is just 0.2 m, which should put the robot in the vicinity of the object it is searching for, but a heading error of the same percentage places the robot 1.25 m away from the object. For a more significant error like 5% or 10%, the distance error (50 or 100 cm) is still possibly manageable, but the heading error places the robot 3.09 or 5.88 m away, which is not even in the vicinity of the object.

The accumulation of odometry errors as the distance moved gets longer is displayed in Fig. 5.6. The initial position of the robot is denoted by the dot at the origin.

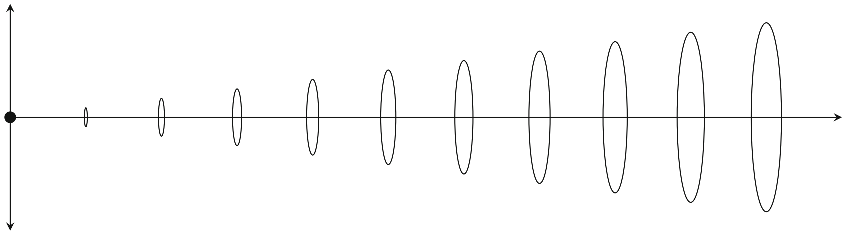


Fig. 5.6 Odometry errors

Assuming an error of at most ±4% in both the linear direction and the heading, the possible positions of the robot after moving $d = 1, 2, \ldots, 10$ m are displayed as ellipses. The minor radii of the error ellipses result from the linear errors:

$$0.04s = 0.04, 0.08, \ldots, 0.4 \;\text{ m},$$

while the major radii of the error ellipses result from the angular errors:

$$d \sin(0.04 \cdot 360^\circ) = d \sin 14.4^\circ \approx 0.25, 0.50, \ldots 2.5 \;\text{ m}.$$

Clearly, the angular errors are much more significant than the linear errors.

Since error is unavoidable, periodically the pose of the robot as computed by odometry must be compared with an absolute position; this becomes the new initial position for further computation. Methods for determining the absolute position of the robot are presented in Chap. 8.

Activity 5.8: Odometry errors

- Write a program to cause the robot to move in a straight line for 2 m. Make sure that the surface is smooth so that it doesn't turn off course and calibrate the motor settings so that the robot moves as straight as possible.
- Vary the motor power of both wheels together so that the robot runs somewhat slower or somewhat faster than before. Plot its position at fixed intervals and see if the error remains linear over the course.
- Vary the motor power of one wheel so that the robot turns slightly to one side. Plot its position at fixed intervals and see if the errors are proportional to the sine of the difference between the original heading and the new heading.

Activity 5.9: Combined effect of odometry errors

- Write a program that causes the robot to move in a straight line for 2 m and then turn 360°. What is the error in the robot's position?
- Write a program that causes the robot to turn 360° and then move in a straight line for 2 m. What is the error in the robot's position? Is there a difference between this error and the error of the previous experiment?
- Write a program that causes the robot to move in a straight line for 2 m, turn 180° and then move in a straight line for 2 m. How far is it from its starting position?

Activity 5.10: Correcting odometry errors

- Modify the program that you wrote for Activity 5.8 to introduce *jitter*, random variation in the power supplied to the motor. Check that the distance that the robot moves in a fixed time is not constant, but changes slightly from run to run.
- Mark a goal line on the floor and compute the time it should take the robot to reach the goal.
- When the robot has moved for that period of time, see if it can find the goal by moving forwards and backwards in small steps until it detects the goal.

5.8 Wheel Encoders

Odometry in a wheeled vehicle like a car can be improved by measuring the rotation of the wheels instead of mapping motor power into velocity. The circumference of a wheel is $2\pi r$, where r is the radius of the wheel in cm, so if n rotations are counted, we know that the robot has moved $2\pi nr$ cm. Wheel encoders can be built that measure fractions of a revolution. If a signal is generated 8 times per revolution, the distance moved is $2\pi nr/8$ cm, which n is now the number of signals counted by the computer.

There are many different ways of implementing wheel encoders. A popular design is to use a light source such as a light-emitting diode (LED), a light sensor and an encoding disk that is attached to the axis of the wheel (Fig. 5.7a). The disk is perforated with holes (Fig. 5.7b) so that whenever the hole is opposite the light source, the sensor generates a signal.

The support for wheel encoders in educational robots varies:

- If a robot does not have wheel encoders it must be calibrated;
- The robot may have wheel encoders that are used internally;
- Some robots like the LEGO® Mindstorms enable the user to read the encoders.

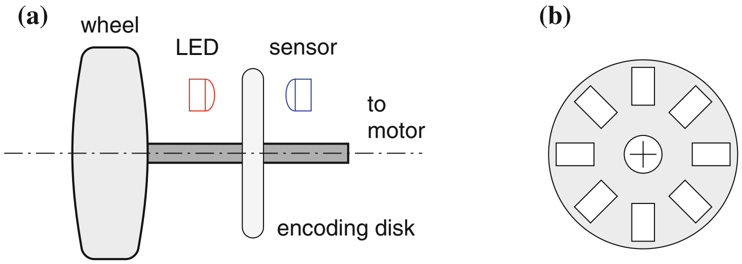


Fig. 5.7 **a** Optical wheel encoder. **b** Encoding disk

The following activity proposes an experiment to measure the distance moved by counting revolutions of a wheel. It can be carried out even if your robot does not have wheel encoders or they are not accessible.

Activity 5.11: Wheel encoding

- Make a mark at the top of a wheel of the robot using chalk or by attaching a narrow piece of colored tape. Write a program that causes the robot to move straight ahead for a fixed period of time. Run the program and take a video of the side of the robot using the camera on your smartphone.
- View the video and determine the number of revolutions by counting the number of times the mark is at the top of the wheel.
- Measure the radius of the wheel and compute the distance moved. How close is the result to the actual distance measured on the floor?
- Repeat the measurement using $n = 2$ and then $n = 4$ equally spaced marks on the wheel. Determine the number of revolutions by counting the number of times that a mark is at the top of the wheel and divide by n. Compute the distance.

5.9 Inertial Navigation Systems

An *inertial navigation system (INS)* directly measures linear acceleration and angular velocity and uses them to calculate the pose of a vehicle. The term *inertial measurement unit (IMU)* is also used, but we prefer the term INS which refers to the entire system. Integrating acceleration from the initial pose to the current time τ gives the current velocity:

$$v = \int_0^\tau a(t)\, dt\,.$$

Similarly, integrating angular velocity gives the change in heading:

$$\theta = \int_0^\tau \omega(t)\, dt\,.$$

In an INS, we are not given continuous functions to integrate; instead, the acceleration and angular velocity are sampled and summation replaces integration:

$$v_n = \sum_{i=0}^{n} a_n \Delta t, \quad \theta_n = \sum_{i=0}^{n} \omega_n \Delta t\,.$$

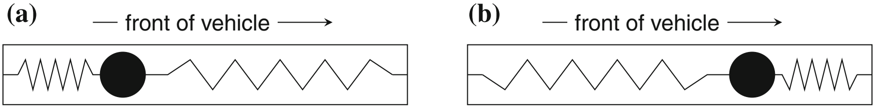


Fig. 5.8 **a** Forward acceleration. **b** Deceleration (braking)

INSs are subject to errors caused by inaccuracies in the measurement itself as well as by variations caused by environmental factors such as temperature, and by wear and tear of the unit. Inertial measurement is often combined with GPS (Sect. 8.3) to update the position with an absolute location.

INSs for robots are constructed with *microelectromechanical systems (MEMS)*, which use integrated circuit manufacturing techniques that combine mechanical elements with electronics that interface with the robot's computer.

5.9.1 Accelerometers

If you have ever flown on an airplane you have experienced a force pushing you back into your seat as a result of the rapid acceleration of the airplane upon takeoff. Upon landing you are pushed away from your seat. You can also experience this in a car that accelerates rapidly or makes an emergency stop. Acceleration is related to force by Newton's second law $F = ma$, where m is the mass. By measuring the force on an object, we measure the acceleration.

Figures 5.8a, b show how an accelerometer can be built from an object (called a *mass*) connected to a spring. The greater the acceleration, the greater the force exerted by the mass upon the spring, which in turn causes the spring to be compressed. The direction that the mass moves gives the sign of the acceleration: forwards or backwards. The magnitude of the force is measured indirectly by measuring the distance that the mass moves. You can see that the diagrams correspond to our experience: when a car accelerates, you are pushed back into the seat, but when it decelerates (brakes) you continue forward.

5.9.2 Gyroscopes

A *gyroscope* ("gyro") uses the principle of Coriolis force to measure angular velocity. This concept is explained in textbooks on physics and we will not go into it here. There are many types of gyros:

- Classical gyros have spinning mechanical disks which are mounted on gimbals so that the axis of rotation remains fixed in space. These gyros are extremely accurate but are very heavy and consume a lot of power. They are found on high-value vehicles such as aircraft and rockets.
- Ring laser gyros (RLG) have (almost) no moving parts and are preferred over mechanical gyros for most applications. They are based on sending two laser beams in opposite directions around a circular or triangular path. If the gyro is rotating, the path followed by one laser beam will be longer than the path followed by the other beam. The difference is proportional to the angular velocity and can be measured and transferred to the navigation computer.
- Coriolis vibratory gyroscopes (CVG) manufactured using MEMS techniques are found in smartphones and robots. They are inexpensive and extremely robust, although their accuracy is not as good are the gyros previously discussed. We now give an overview of how they work.

Figure 5.9 shows a CVG called a *tuning fork gyroscope*. Two square masses are attached by flexible beams to anchors that are mounted on the base of the component. Drivers force the masses to vibrate left and right. If the component rotates, the masses move upwards and downwards a distance proportional to the angular velocity. The masses and the electrodes form the plates of capacitors whose capacitance increases or decreases as the plates come together or move apart.

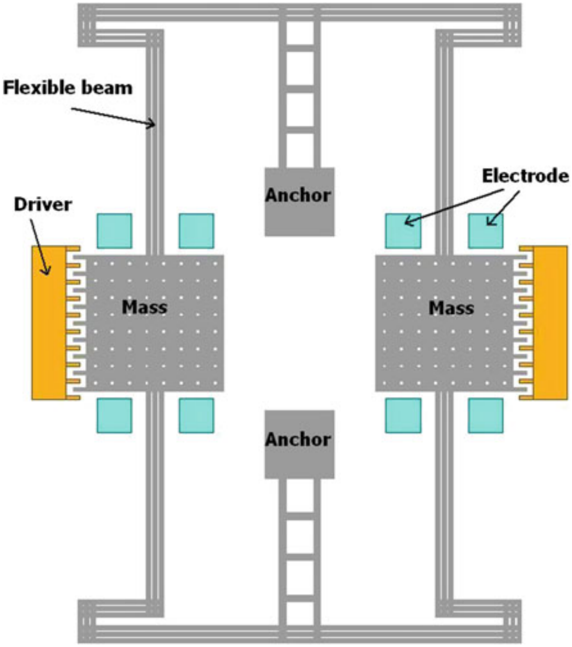


Fig. 5.9 Tuning fork gyroscope (Courtesy of Zhili Hao, Old Dominion University)

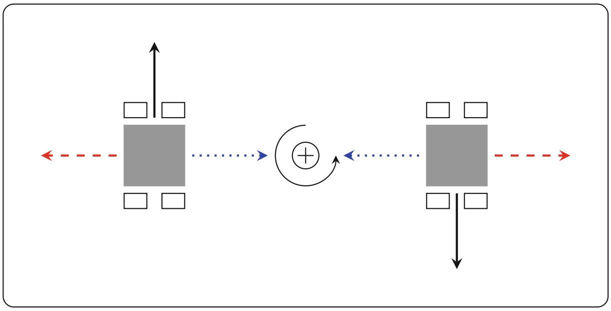


Fig. 5.10 Physics of a tuning fork gyroscope: *red dashed arrows* and *blue dotted arrows* indicate the direction of the vibration; *solid black arrows* indicate the direction of the Coriolis force

The theory of operation of the tuning fork gyroscope is shown in Fig. 5.10. The masses (gray squares) are forced to vibrate at the same frequency like the two prongs of a tuning fork. They vibrate in different directions, that is, they either approach each other (blue dotted arrows) or they move away from each other (dashed red arrows). The component rotates around an axis perpendicular to its center (the circle with a cross denotes the rotational axis which is perpendicular to the plane of the paper). The Coriolis force is a force whose direction is given by the vector cross product of the axis of the rotation and the movement of the mass, and whose magnitude is proportional to the linear velocity of the mass and the angular velocity of the gyro. Since the masses are moving in different directions, the resulting forces will be equal but in opposite directions (solid arrows). The masses approach or recede from the electrodes (small rectangles) and the change in capacitance can be measured by a circuit.

5.9.3 Applications

An inertial navigation system has three accelerometers and three gyroscopes so that the pose of the vehicle can be computed in three dimensions. This is necessary for robotic aircraft and other robotic vehicles. Airbags use an accelerometer that detects the rapid deceleration in the front-back direction that occurs when a car crashes. This causes an explosive expansion of the airbag. One can conceive of more applications for these components in cars. An accelerometer in the up-down direction can detect if the car has fallen into a pothole. A gyroscope measuring rotation around the vertical axis can detect skidding, while the gyroscope measuring rotation around the front-rear axis can detect if the car is rolling over.

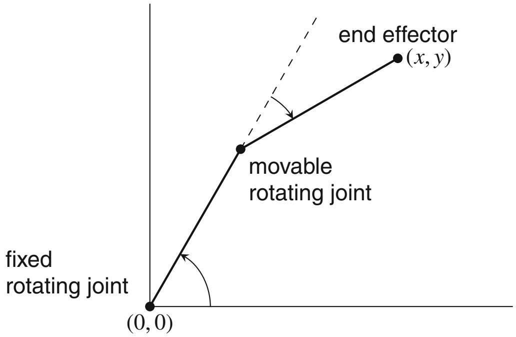


Fig. 5.11 A two-link robotic arm with two DOF

5.10 Degrees of Freedom and Numbers of Actuators

The number of *degrees of freedom (DOF)* of a system is the dimensionality of the coordinates needed to describe a pose of a mobile robot or the pose of the end effector of a robotic manipulator.[3] For example, a helicopter has six DOF because it can move in the three spatial dimensions and can rotate around the three axes. Therefore, a six-dimensional coordinate $(x, y, z, \phi, \psi, \theta)$ is needed to describe its pose.

The terms used to describe rotations

A helicopter can rotate around all three of its axes. The rotations are called: (a) pitch: the nose moves up and down; (b) roll: the body rotates around its lengthwise axis; (c) yaw: the body rotates left and right around the axis of its rotor.

The two-link robotic arm in Fig. 5.11 has only two DOF because its end effector moves in a plane and does not rotate; therefore, it can be described by a two-dimensional coordinate (x, y). By examining Fig. 5.3 again, you can see that a mobile robot moving on a flat surface has three DOF, because its pose is defined by a three-dimensional coordinate (x, y, θ). A train has only one DOF since it is constrained by the tracks to move forwards (or occasionally backwards) along the track. It only takes one coordinate (x), the train's distance from an arbitrary origin of the track, to specify the pose of the train.

We need more information than the degrees of freedom to describe robotic motion. Consider a vehicle like a car, a bicycle or an office chair. Although three coordinates (x, y, θ) are needed to describe its pose, we cannot necessarily move the vehicle directly from one pose to another. An office chair can be moved *directly* to any point of the plane and oriented in any direction. A car or a bicycle at (2, 0, 0°) (pointed along the positive x-axis) cannot be moved directly up the y-axis to position (2, 2, 0°). A more complex maneuver is needed.

[3] This section and the following ones are more advanced and can skipped during your first reading. Furthermore, some of the examples are of robotic manipulators described in Chap. 16.

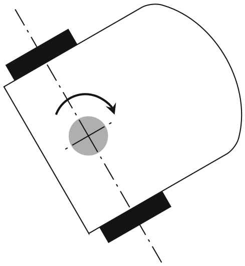


Fig. 5.12 A robot that can only rotate around an axis (*gray dot*)

We need to know the number of its *actuators* (usually motors) and their configuration. A differential drive robot has two actuators, one for each wheel, although the robot itself has three DOF. The motors move the robot along one axis forwards and backwards, but by applying unequal power we can change the heading of the robot. The two-link arm in Fig. 5.11 has two motors, one at each rotating joint, so the number of actuators equals the number of DOF. Finally, a train has only one actuator, the motor that moves it forwards or backwards in its single DOF.

Activity 5.12: Robot that can only rotate

- Figure 5.12 shows a differential drive robot with a fixed rod through its center of rotation. The rod prevents the robot changing its position, allowing it only to rotate around its vertical axis. Characterize this configuration: the number of actuators and the number of DOF.
- What types of tasks could this robot perform? What are the advantages and disadvantages of this configuration?

5.11 The Relative Number of Actuators and DOF

Let us analyze systems where:

- The number of actuators equals the number of DOF;
- The number of actuators is fewer than the number of DOF;
- The number of actuators is greater than the number of DOF.

The Number of Actuators Equals the Number of DOF

A train has one actuator (its engine) that moves the train along its single DOF. The two-line robotic arm in Fig. 5.11 has two actuators and two DOF. A robotic gripper can be built with three motors that rotate the gripper in each of the three orientations (roll, pitch, yaw). The advantage of having an equal number of actuators and DOF is that the system is relatively easy to control: each actuator is individually commanded to move the robot to the desired position in the DOF it controls.

The Number of Actuators is Fewer than the Number of DOF

Mobile robots will usually have fewer actuators than DOF. A robot with differential drive and a car have only two actuators, but they can reach all possible three-dimensional poses in the plane. Having fewer actuators makes the system less expensive, but the problems of planning and controlling motion are much more difficult. Parallel parking a car is notorious for its difficulty: two rotations and a translation are needed to perform a simple lateral move (Fig. 5.21a, b).

An extreme example is a hot-air balloon which has only a single actuator (a heater) that injects more or less hot air into the balloon and thus controls its altitude. However, winds can cause the balloon to move in any of the three spatial directions and even to rotate (at least partially) in three orientations, so the operator of the balloon can never precisely control the balloon. A hot-air balloon therefore differs from an elevator: both have a single actuator, but the elevator is constrained by its shaft to move in only one DOF.

For another example of the complex relationship between the DOF and number of actuators, the reader is invited to study flight control in helicopters. Helicopters are highly maneuverable (even more so than airplanes which can't fly backwards), but a pilot controls the helicopter's flight using only three actuators:

- The *cyclic* controls the pitch of the main rotor *shaft* which determines if the helicopter moves forwards, backwards or to either side.
- The *collective* controls the pitch of the *blades* of the main rotor which determines if the helicopter moves up or down.
- The *pedals* control the speed of the tail rotor which determines the direction in which the nose of the helicopter points.

The Number of Actuators is Greater than the Number of DOF

It doesn't seem to be a good idea to have *more* actuators than DOF, but in practice such configurations are often useful. The systems in Fig. 5.13a, b have more actuators than DOF. The robotic manipulator arm in Fig. 5.13a has four links rotating in the plane with actuators (motors) a1, a2, a3, a4 at the joints. We assume that the end effector is fixed to the link from a4 and cannot rotate, so its pose is defined by its position (x, y) and a fixed orientation. Therefore, although the arm has four actuators, it has only two DOF because it can move the end effector only horizontally and vertically.

The mobile robot with an arm (Fig. 5.13b) has three actuators: a motor that moves the robot forwards and backwards, and motors for each of the two rotating joints. However, the system has only two DOF since the pose of its end effector is defined by a two-dimensional (x, y) coordinate. Systems with more actuators than DOF are called *redundant systems*.

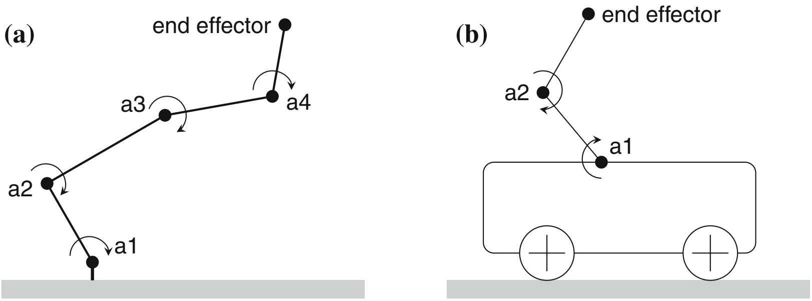


Fig. 5.13 **a** Robot arm: two DOF and four actuators. **b** Mobile robot and arm: two DOF and three actuators

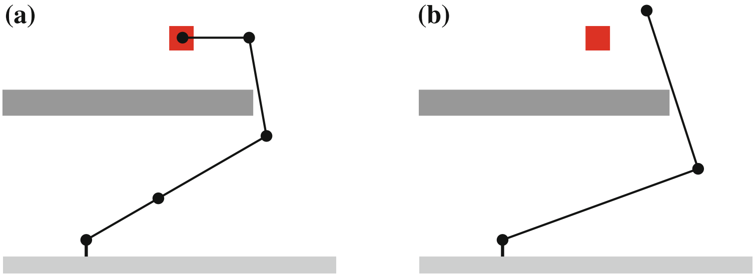


Fig. 5.14 **a** Arm with four actuators can reach a hidden position. **b** Arm with two actuators is blocked by an obstacle

If possible, engineers avoid using more than one actuator acting on the same DOF because it increases the complexity and cost of a system. The inverse kinematics (Sect. 16.2) of a redundant system results in an infinite number of solutions which complicate the operation of the system. For the mobile robot with the arm (Fig. 5.13b), there are an infinite number of positions of the base and arm that bring the end effector to a specific reachable position.

There are situations where a redundant system is required because the task could not be performed with fewer actuators. Figure 5.14a shows how the four-link robotic arm of Fig. 5.13a can move the end effector to a position that is blocked by an obstacle and thus unreachable by a two-link arm (Fig. 5.14b), even though in both configurations the total length of the links is equal.

An important advantage of redundant systems arises from actuators with different characteristics. The mobile robot in Fig. 5.13b can approach the target quickly, although its final position might not be accurate because of errors like uneven terrain. Once the mobile robot stops, the motors in the joints which do not have to deal with the terrain can be precisely positioned. While the positioning is precise, these joints do not have the broad range of the mobile base.

An Example of a System with More Actuators than DOF

Figure 5.15 (top and side views) shows a configuration with *two actuators and one DOF*. The system models a robotic crane that moves a heavy weight to a specific

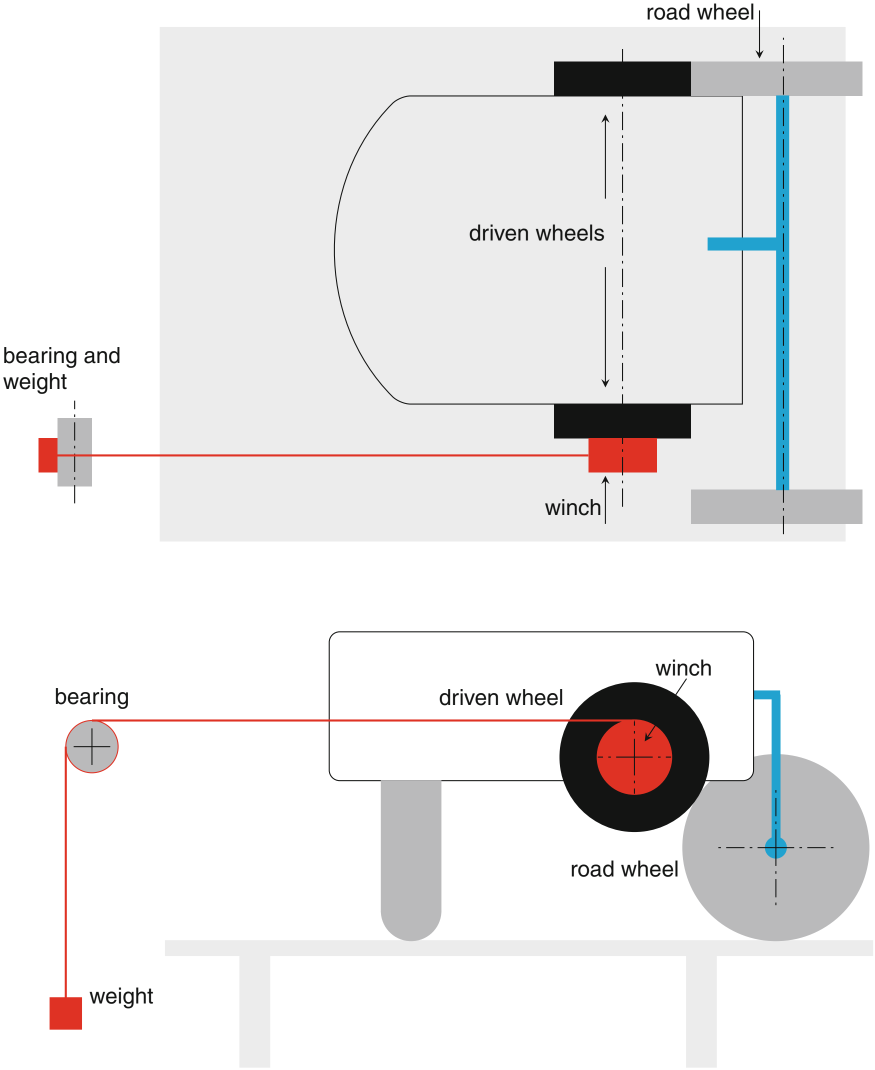


Fig. 5.15 Robotic crane built from a mobile robot and a winch (top view *above*, side view *below*); in the side view the left wheel is not shown

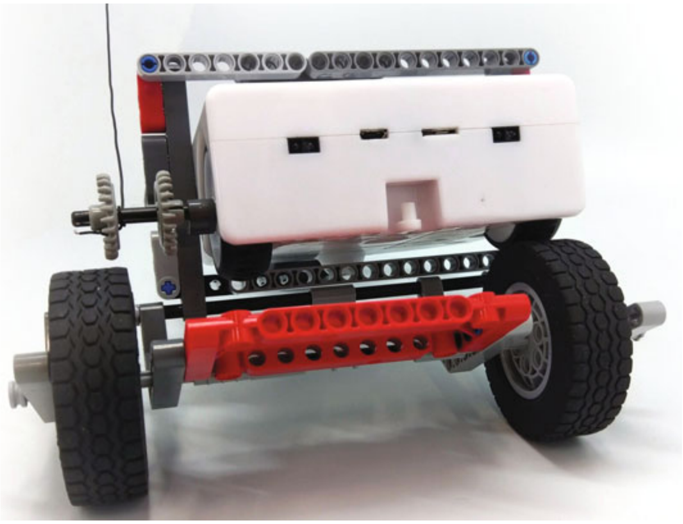


Fig. 5.16 Robotic crane built from a Thymio robot and LEGO® components

vertical position. Figure 5.16 shows a crane built from a Thymio robot and LEGO® components.

The system is build from a mobile robot with differential drive, but the wheels are not directly used to control the motion of the system. Instead, each wheel is an independent actuator. (Recall that the power to each wheel of a differential drive robot can be set independently to any value in a range such as −100 to 100.)

The robot faces left. The right driven wheel in Fig. 5.15 (the black rectangle at the top of the top view and hidden behind the robot in the side view) controls a pair of (gray) road wheels that move the robot rapidly forwards and backwards. In turn, this causes the cable to move the weight rapidly up and down.

The road wheels are mounted on a structure (in blue) that is fixed to the robot body. There are several options for transferring power from the right driven wheel to the road wheels: friction, pulleys and belts, and gears. Each option has its own advantages and disadvantages, and all three are used in cars: the clutch uses friction, belts are used for timing and to run auxiliary components like water pumps, and gears are used in the transmission to control the torque applied to each wheel.

The left driven wheel (the black rectangle at the bottom of the top view and at the front of the side view) controls a winch (red) that rolls or unrolls a cable attached to the weight that moves up or down over a fixed bearing. The winch has a diameter much smaller than the diameter of the driven wheels, so it can move the weight in small increments as the left driven wheel rotates. The design goal is to be able to perform precise positioning of the weight even though the winch moves the cable at a much slower speed than does the robot body.

There are two activities for this section. This activity is for readers who have good construction skills and an appropriate robotics kit. The second activity suggests alternate ways of demonstrating the concept of two actuators in one DOF.

Activity 5.13: Robotic crane

- Construct the robotic crane shown in Fig. 5.15. Explain your choice of mechanism for connecting the driven wheel to the road wheels.
- Write a program that given the current position of the weight and a goal position moves the weight to the goal position. Alternatively, send commands to the motors using a remote control device or a computer connected to the robot.
- Experiment with the relative rotational speeds of the left and right driven wheels that control the road wheels and the winch, respectively. Should you move the two actuators separately or simultaneously?

Activity 5.14: Robotic crane (alternatives)

- Write a program that causes a mobile robot to move forwards and backwards. Place a piece of black tape relatively far from the initial position of the robot. The goal is to cause the robot to stop as near as possible to the start of the tape *without* continuously checking the sensor.
- The program has three modes of operation. (1) The robot moves fast, checking its sensor occasionally, and stopping when it detects the tape. (2) As in (1) but the robot moves slowly, checking its sensor relatively often. (3) As in (1) but when the tape is detected, the robot moves backwards using the speed and sampling period as in (2).
- Run the program and compare the results of the three modes: the time until the robot stops and error between the robot's final position and the start of the tape.
- Alternatively, run the program with the three modes on a computer and experiment with the motion parameters and the sampling periods. You will need to choose a model for the motion: constant velocity, constant acceleration, or (more realistically) acceleration then constant velocity and finally deceleration when the tape is detected.

5.12 Holonomic and Non-holonomic Motion

Section 5.10 presented the concept of degree of freedom (DOF) and the role of the number of actuators. There is another concept that links the DOF and the actuators in the case of mobile robot: the *degree of mobility (DOM)*. The degree of mobility δ_m corresponds to the number of degrees of freedom that can be *directly accessed* by the actuators. A mobile robot in the plane has at most three DOF ((x, y) position and heading), so the maximal degree of mobility of a mobile robot is $\delta_m = 3$.

Let consider the DOM of various vehicles. A train has one DOF because it can only move forwards along the tracks, and it has one actuator, its engine, that directly affects this single degree of freedom. Therefore, a train has a degree of mobility of $\delta_m = 1$, meaning that the single DOF can be directly accessed by the actuator.

A robot with differential drive has three DOF. The two actuators are the two motors which act on the wheels. They can directly access two DOF: (a) if both wheels turn at the same speed, the robot moves forwards or backwards; (b) if the wheels have speeds in opposite directions, the robot rotates in place. Therefore, we can directly access the DOF along the forward axis of translation and the DOF of the heading, but we cannot directly access the DOF of the lateral axis of translation (Fig. 5.17a). A differential drive mobile robot has a degree of mobility $\delta_m = 2 < 3 = \#\text{DOF}$.

A car, like a robot with differential drive, has only two actuators for three DOF: one actuator, the motor, gives direct access to the degree of freedom along the longitudinal axis of the car, enabling it to move forwards and backwards. The other actuator, the steering wheel, does *not* give direct access to any additional DOF, it can only orient the first DOF. The car cannot rotate around the vertical axis and it cannot move laterally (Fig. 5.17b). Therefore, a car has only one degree of mobility, $\delta_m = 1$. Intuitively, you can see the lower degree of mobility of a car compared with a robot with differential drive by noting that the robot can rotate in place while the car cannot.

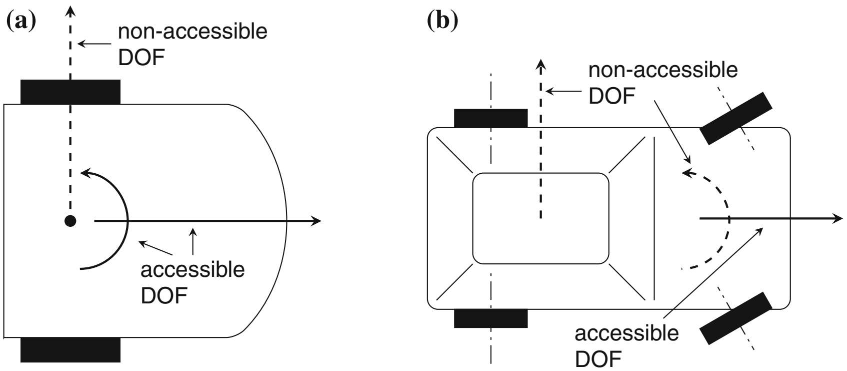


Fig. 5.17 **a** Accessible and non-accessible DOF for a robot with differential drive. **b** Accessible and non-accesible DOF for a robot with Ackermann steering

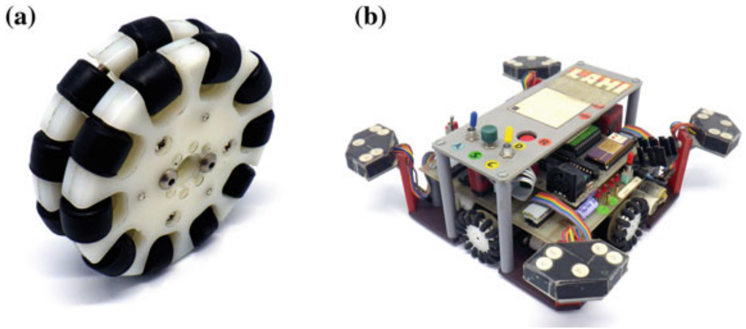


Fig. 5.18 **a** Swedish wheel. **b** Omnidirectional robot (Courtesy LAMI-EPFL)

By itself, a standard wheel has $\delta_m = 2$: it can roll forwards and backwards and it can rotate around the vertical axis that goes through the point of contact of the wheel with the ground. A wheel cannot move sideways, which is actually a good thing because it prevents the vehicle from skidding off the road during a turn. In the car, the degree of mobility is reduced even further to $\delta_m = 1$, because there are two pairs of wheels, one in the front and one in the rear of the car. This configuration makes it impossible for the car to rotate around its vertical axis, even though the individual wheels can do so (usually only the front wheels). The limitation to $\delta_m = 1$ gives stability to the car—it cannot skid laterally and it cannot rotate—making it easy and safe to drive at high speeds. In fact, an accident can occur when rain or snow reduce the friction so that the car can skid or rotate.

An autonomous mobile robot can profit if it has a greater DOM $\delta_m = 3$. To directly access the third DOF, the robot needs to be able to move laterally. One method is to have the robot roll on a ball or a castor wheel like an office chair. Another method is to use *Swedish wheels* (Fig. 5.18a). A Swedish wheel is a standard wheel that has small free wheels along its rim so that it can move laterally, enabling direct access to the third DOF.

Mobile robots that can directly access all three DOF ($\delta_m = 3$) are called *omnidirectional robots*. Figure 5.18b shows an omnidirectional robot constructed with four Swedish wheels. The two pairs of wheels on opposite sides of the robot can directly move the robot left, right, forwards and backwards. This configuration is redundant but very easy to control. To avoid redundancy, most omnidirectional robots have three Swedish wheels mounted at an angle of 120° from each other (Fig. 5.19). This configuration has $\delta_m = 3$ but is not easy to control using the familiar x, y coordinates.

The relative values of the DOF and the DOM of a robot define the concept of holonomic motion. A robot has *holonomic motion* if $\delta_m =$ #DOF and it has *non-holonomic motion* $\delta_m <$ #DOF. A holonomic robot like the one in Fig. 5.18b can directly control all its DOF without difficult maneuvers. Figure 5.20 shows how easy it is for the omnidirectional robot with Swedish wheels (Fig. 5.19) to perform parallel parking.

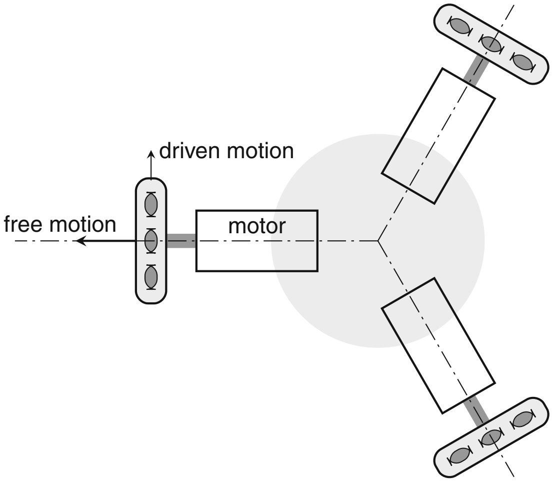


Fig. 5.19 Omnidirectional robot with three Swedish wheels

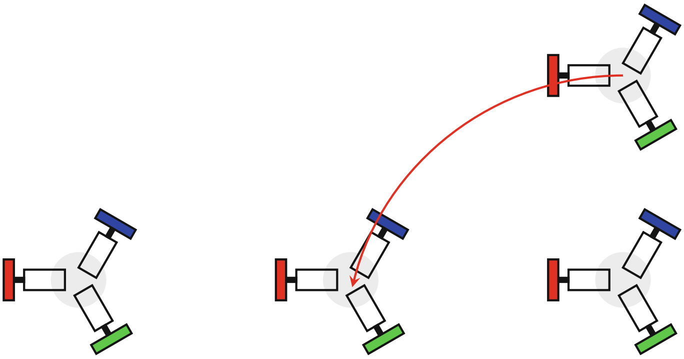


Fig. 5.20 Parallel parking by an omnidirectional robot

A car and a robot with differential drive are non-holonomic because their DOM ($\delta_m = 1$ and 2, respectively) are lower than their DOF which is three. Because of this limited degree of mobility, these vehicles need complex steering maneuvers, for example, to perform parallel parking. There is a significant difference between the two vehicles. The differential drive robot needs three separate movements, but they are very simple (Fig. 5.21a): rotate left, move backwards, rotate right. The car also needs three separate movements, but they are extremely difficult to perform correctly (Fig. 5.21b). You have to estimate where to start the maneuver, how sharp to make each turn and how far to move between turns. The higher DOM of the differential drive robot is advantageous in this situation.

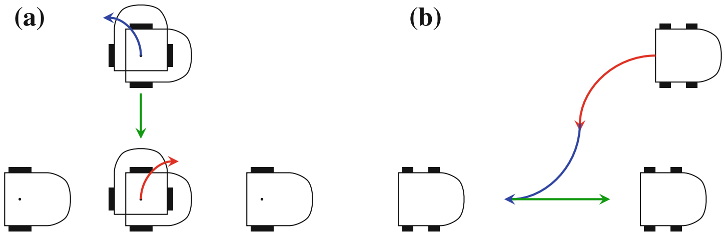


Fig. 5.21 **a** Parallel parking for a non-holonomic differential drive robot. **b** Parallel parking for a non-holonomic car

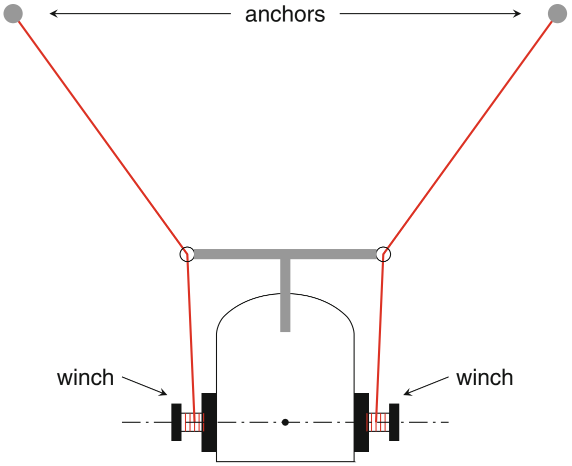


Fig. 5.22 Robot for cleaning a wall

Activity 5.15: Holonomic and non-holonomic motion

- Look again at the mobile robot which is constrained to rotational motion only (Fig. 5.12). What is its degree of mobility δ_m? It is holonomic or not?
- Figure 5.22 shows a robot for cleaning the walls of a building. There are two anchors from which cables descend, passing through eyes fixed to the robot's body and then to winches powered by the robot's wheels. By rolling and unrolling the cables, the robot moves up and down the wall. However, if the two motors do not cause the wheels move precisely at the same rotational velocity, the robot will swing from side to side. How many DOF and how many DOM does this robot have? Is it holonomic or not?

5.13 Summary

A mobile robot like a self-driving car or a Mars explorer will not have landmarks always available for navigation. Odometry is used to bring the robot to the vicinity of its goal without reference to the environment. The robot estimates its speed and rotational velocity from the power applied to its motors. Odometry can be improved by using wheel encoders to measure the number of revolutions of the wheels, rather than inferring the velocity from the motor power. The change in the position of an inexpensive robot moving in a straight line can be computed by multiplying speed by time. If the robot is turning, trigonometric calculations are needed to compute the new position and orientation. Even with wheel encoders, odometry is subject to errors that can be very large if the error is in the heading.

Inertial navigation uses accelerometers and gyroscopes to improve the accuracy of odometry. Integrating acceleration gives velocity and integrating angular velocity gives the heading. Microelectromechanical systems have made inertial navigation inexpensive enough for use in robotics.

The DOF of a system is the number of dimensions in which it can move—up to three dimensions on a surface and up to six dimensions in the air or underwater—but a robot may be constrained to have fewer than the maximum number of DOF. An additional consideration is the number and configuration of the actuators of a robot which define its degree of mobility. If the DOM is equal to the number of DOF, the robot is holonomic and it can move directly from one pose to another, although it may be difficult to control. If the DOM is less than the number of DOF, the robot is non-holonomic; it cannot move directly from one pose to another and will require complex maneuvers to carry out some tasks.

5.14 Further Reading

A detailed mathematical treatment of odometry errors in two dimensions is given in [5, Sect. 5.24]. For an overview of inertial navigation see [3, 4]. Advanced textbooks on robotics present holonomy [1, 2, 5, 6].

References

1. Correll, N.: Introduction to Autonomous Robots. CreateSpace (2014). https://github.com/correll/Introduction-to-Autonomous-Robots/releases/download/v1.9/book.pdf
2. Craig, J.J.: Introduction to Robotics: Mechanics and Control, 3rd edn. Pearson, Boston (2005)
3. King, A.: Inertial navigation—forty years of evolution. GEC Rev. **13**(3), 140–149 (1998). http://www.imar-navigation.de/downloads/papers/inertial_navigation_introduction.pdf
4. Oxford Technical Solutions: What is an (INS) inertial navigation system? http://www.oxts.com/what-is-inertial-navigation-guide/

5. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: Introduction to Autonomous Mobile Robots, 2nd edn. MIT Press, Cambridge (2011)
6. Spong, M.W., Hutchinson, S., Vidyasagar, M.: Robot Modeling and Control. Wiley, New York (2005)