3.2 Rigid-Body Transformations

Any of the techniques from Section 3.1 can be used to define both the obstacle region and the robot. Let $\mathcal{O}$ refer to the *obstacle region*, which is a subset of $\mathcal{W}$. Let $\mathcal{A}$ refer to the robot, which is a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$, matching the dimension of $\mathcal{W}$. Although $\mathcal{O}$ remains fixed in the world, $\mathcal{W}$, motion planning problems will require ``moving'' the robot, $\mathcal{A}$.

3.2.1 General Concepts

Before giving specific transformations, it will be helpful to define them in general to avoid confusion in later parts when intuitive notions might fall apart. Suppose that a rigid robot, $\mathcal{A}$, is defined as a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$. A *rigid-body transformation* is a function, $h : \mathcal{A} \rightarrow \mathcal{W}$, that maps every point of $\mathcal{A}$ into $\mathcal{W}$ with two requirements: 1) The distance between any pair of points of $\mathcal{A}$ must be preserved, and 2) the orientation of $\mathcal{A}$ must be preserved (no ``mirror images'').

Using standard function notation, $h(a)$ for some $a \in \mathcal{A}$ refers to the point in $\mathcal{W}$ that is ``occupied'' by a. Let

$$h(\mathcal{A}) = \{h(a) \in \mathcal{W} \mid a \in \mathcal{A}\}, \tag{3.21}$$

which is the image of h and indicates all points in $\mathcal{W}$ occupied by the transformed robot.

Transforming the robot model

Consider transforming a robot model. If $\mathcal{A}$ is expressed by naming specific points in $\mathbb{R}^2$, as in a boundary representation of a polygon, then each point is simply transformed from a to $h(a) \in \mathcal{W}$. In this case, it is straightforward to transform the entire model using h. However, there is a slight complication if the robot model is expressed using primitives, such as

$$H_i = \{a \in \mathbb{R}^2 \mid f_i(a) \leq 0\}. \tag{3.22}$$

This differs slightly from (3.2) because the robot is defined in $\mathbb{R}^2$ (which is not necessarily $\mathcal{W}$), and also a is used to denote a point $(x, y) \in \mathcal{A}$. Under a transformation h, the primitive is transformed as

$$h(H_i) = \{h(a) \in \mathcal{W} \mid f_i(a) \leq 0\}. \tag{3.23}$$

To transform the primitive completely, however, it is better to directly name points in $w \in \mathcal{W}$, as opposed to $h(a) \in \mathcal{W}$. Using the fact that $a = h^{-1}(w)$, this becomes

$$h(H_i) = \{w \in \mathcal{W} \mid f_i(h^{-1}(w)) \leq 0\}, \tag{3.24}$$

in which the inverse of h appears in the right side because the original point $a \in A$ needs to be recovered to evaluate f_i. Therefore, it is important to be careful because either h or h^{-1} may be required to transform the model. This will be observed in more specific contexts in some coming examples.

A parameterized family of transformations

It will become important to study families of transformations, in which some parameters are used to select the particular transformation. Therefore, it makes sense to generalize h to accept two variables: a parameter vector, $q \in \mathbb{R}^n$, along with $a \in \mathcal{A}$. The resulting transformed point a is denoted by $h(q, a)$, and the entire robot is transformed to $h(q, \mathcal{A}) \subset \mathcal{W}$.

The coming material will use the following shorthand notation, which requires the specific h to be inferred from the context. Let $h(q, a)$ be shortened to $a(q)$, and let $h(q, \mathcal{A})$ be shortened to $\mathcal{A}(q)$. This notation makes it appear that by adjusting the parameter q, the robot $\mathcal{A}$ travels around in $\mathcal{W}$ as different transformations are selected from the predetermined family. This is slightly abusive notation, but it is convenient. The expression $\mathcal{A}(q)$ can be considered as a set-valued function that yields the set of points in $\mathcal{W}$ that are occupied by $\mathcal{A}$ when it is transformed by q. Most of the time the notation does not cause trouble, but when it does, it is helpful to remember the definitions from this section, especially when trying to determine whether h or h^{-1} is needed.

Defining frames

It was assumed so far that $\mathcal{A}$ is defined in $\mathbb{R}^2$ or $\mathbb{R}^3$, but before it is transformed, it is not considered to be a subset of $\mathcal{W}$. The transformation h *places* the robot in $\mathcal{W}$. In the coming material, it will be convenient to indicate this distinction using coordinate frames. The

origin and coordinate basis vectors of $\mathcal{W}$ will be referred to as the *world frame*.[3.3] Thus, any point $w \in \mathcal{W}$ is expressed in terms of the world frame.

The coordinates used to define $\mathcal{A}$ are initially expressed in the *body frame*, which represents the origin and coordinate basis vectors of $\mathbb{R}^2$ or $\mathbb{R}^3$. In the case of $\mathcal{A} \subset \mathbb{R}^2$, it can be imagined that the body frame is painted on the robot. Transforming the robot is equivalent to converting its model from the body frame to the world frame. This has the effect of *placing*[3.4] $\mathcal{A}$ into $\mathcal{W}$ at some position and orientation. When multiple bodies are covered in Section 3.3, each body will have its own body frame, and transformations require expressing all bodies with respect to the world frame.

Translation

A rigid robot $\mathcal{A} \subset \mathbb{R}^2$ is *translated* by using two parameters, $x_t, y_t \in \mathbb{R}$. Using definitions from Section 3.2.1, $q = (x_t, y_t)$, and h is defined as

$$h(x, y) = (x + x_t, y + y_t). \tag{3.25}$$

A boundary representation of $\mathcal{A}$ can be translated by transforming each vertex in the sequence of polygon vertices using (3.25). Each point, (x_i, y_i), in the sequence is replaced by $(x_i + x_t, y_i + y_t)$.

Now consider a solid representation of $\mathcal{A}$, defined in terms of primitives. Each primitive of the form

$$H_i = \{(x, y) \in \mathbb{R}^2 \mid f(x, y) \leq 0\} \tag{3.26}$$

is transformed to

$$h(H_i) = \{(x, y) \in \mathcal{W} \mid f(x - x_t, y - y_t) \leq 0\}. \tag{3.27}$$

Example 3..2 (Translating a Disc) For example, suppose the robot is a disc of unit radius, centered at the origin. It is modeled by a single primitive,

$$H_i = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 - 1 \leq 0\}. \tag{3.28}$$

Suppose $\mathcal{A} = H_i$ is translated x_t units in the x direction and y_t units in the y direction. The transformed primitive is

$$h(H_i) = \{(x, y) \in \mathcal{W} \mid (x - x_t)^2 + (y - y_t)^2 - 1 \leq 0\}, \tag{3.29}$$

which is the familiar equation for a disc centered at (x_t, y_t). In this example, the inverse, h^{-1} is used, as described in Section 3.2.1. ■

The translated robot is denoted as $\mathcal{A}(x_t, y_t)$. Translation by $(0,0)$ is the *identity transformation*, which results in $\mathcal{A}(0,0) = \mathcal{A}$, if it is assumed that $\mathcal{A} \subset \mathcal{W}$ (recall that $\mathcal{A}$ does not necessarily have to be initially embedded in $\mathcal{W}$). It will be convenient to use the term *degrees of freedom* to refer to the maximum number of independent parameters that are needed to completely characterize the transformation applied to the robot. If the set of allowable values for x_t and y_t forms a two-dimensional subset of $\mathbb{R}^2$, then the degrees of freedom is two.

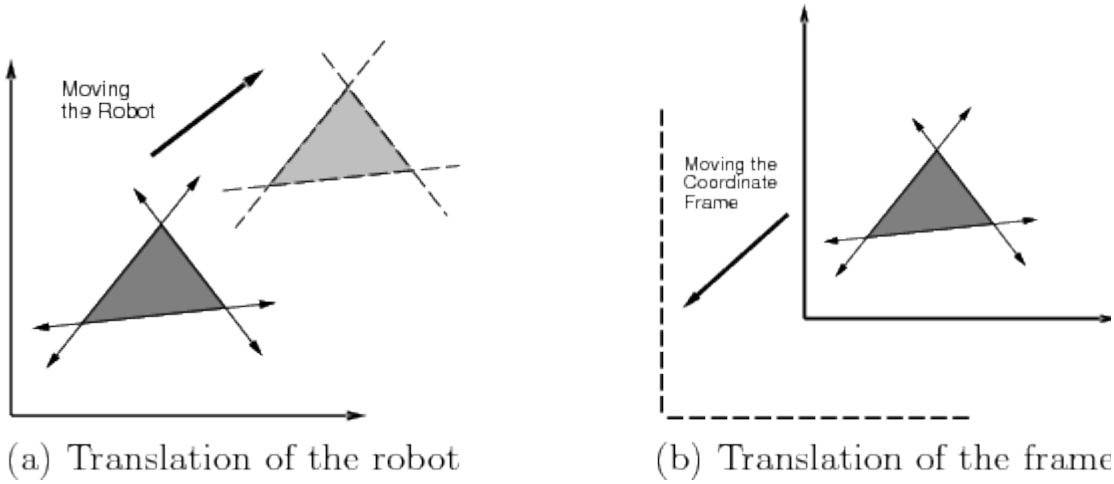

Figure 3.7: Every transformation has two interpretations.

Suppose that $\mathcal{A}$ is defined directly in $\mathcal{W}$ with translation. As shown in Figure 3.7, there are two interpretations of a rigid-body transformation applied to $\mathcal{A}$: 1) The world frame remains fixed and the robot is transformed; 2) the robot remains fixed and the world frame is translated. The first one characterizes the effect of the transformation from a fixed world frame, and the second one indicates how the transformation appears from the robot's perspective. Unless stated otherwise, the first interpretation will be used when we refer to motion planning problems because it often models a robot moving in a physical world. Numerous books cover coordinate transformations under the second interpretation. This has been known to cause confusion because the transformations may sometimes appear ``backward'' from what is desired in motion planning.

Rotation

The robot, $\mathcal{A}$, can be *rotated* counterclockwise by some angle $\theta \in [0, 2\pi)$ by mapping every $(x, y) \in \mathcal{A}$ as

$$(x, y) \mapsto (x \cos\theta - y \sin\theta,\ x \sin\theta + y \cos\theta). \tag{3.30}$$

Using a 2×2 rotation matrix,

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \tag{3.31}$$

the transformation can be written as

$$\begin{pmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{pmatrix} = R(\theta) \begin{pmatrix} x \\ y \end{pmatrix}. \tag{3.32}$$

Using the notation of Section 3.2.1, $R(\theta)$ becomes $h(q)$, for which $q = \theta$. For linear transformations, such as the one defined by (3.32), recall that the column vectors represent the basis vectors of the new coordinate frame. The column vectors of $R(\theta)$ are unit vectors, and their inner product (or dot product) is zero, indicating that they are orthogonal. Suppose that the x and y coordinate axes, which represent the body frame, are ``painted'' on $\mathcal{A}$. The columns of $R(\theta)$ can be derived by considering the resulting directions of the x- and y-axes, respectively, after performing a counterclockwise rotation by the angle θ. This interpretation generalizes nicely for higher dimensional rotation matrices.

Note that the rotation is performed about the origin. Thus, when defining the model of $\mathcal{A}$, the origin should be placed at the intended axis of rotation. Using the semi-algebraic model, the entire robot model can be rotated by transforming each primitive, yielding $\mathcal{A}(\theta)$. The inverse rotation, $R(-\theta)$, must be applied to each primitive.

Combining translation and rotation

Suppose a rotation by θ is performed, followed by a translation by x_t, y_t. This can be used to place the robot in any desired position and orientation.

Note that translations and rotations do not commute! If the operations are applied successively, each $(x, y) \in \mathcal{A}$ is transformed to

$$\begin{pmatrix} x\cos\theta - y\sin\theta + x_t \\ x\sin\theta + y\cos\theta + y_t \end{pmatrix}. \tag{3.33}$$

The following matrix multiplication yields the same result for the first two vector components:

$$\begin{pmatrix} \cos\theta & -\sin\theta & x_t \\ \sin\theta & \cos\theta & y_t \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x\cos\theta - y\sin\theta + x_t \\ x\sin\theta + y\cos\theta + y_t \\ 1 \end{pmatrix}. \tag{3.34}$$

This implies that the 3×3 matrix,

$$T = \begin{pmatrix} \cos\theta & -\sin\theta & x_t \\ \sin\theta & \cos\theta & y_t \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.35}$$

represents a rotation followed by a translation. The matrix T will be referred to as a *homogeneous transformation matrix*. It is important to remember that T represents a rotation *followed by* a translation (not the other way around). Each primitive can be transformed using the inverse of T, resulting in a transformed solid model of the robot. The transformed robot is denoted by $\mathcal{A}(x_t, y_t, \theta)$, and in this case there are three degrees of freedom. The homogeneous transformation matrix is a convenient representation of the combined transformations; therefore, it is frequently used in robotics, mechanics, computer graphics, and elsewhere. It is called homogeneous because over $\mathbb{R}^3$ it is just a linear transformation without any translation. The trick of increasing the dimension by one to absorb the translational part is common in projective geometry [804].

3D translation

The robot, $\mathcal{A}$, is *translated* by some $x_t, y_t, z_t \in \mathbb{R}$ using

$$(x, y, z) \mapsto (x + x_t, y + y_t, z + z_t). \tag{3.36}$$

A primitive of the form

$$H_i = \{(x, y, z) \in \mathcal{W} \mid f_i(x, y, z) \leq 0\} \tag{3.37}$$

is transformed to

$$\{(x, y, z) \in \mathcal{W} \mid f_i(x - x_t, y - y_t, z - z_t) \leq 0\}. \tag{3.38}$$

The translated robot is denoted as $\mathcal{A}(x_t, y_t, z_t)$.

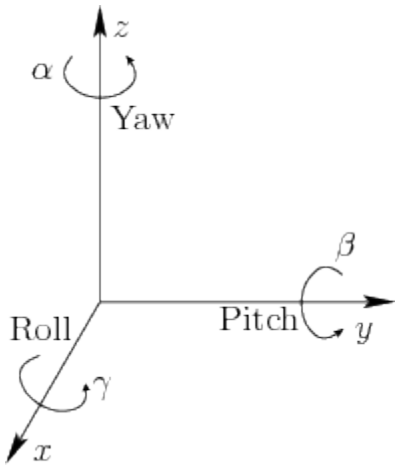

Figure 3.8: Any three-dimensional rotation can be described as a sequence of yaw, pitch, and roll rotations.

Yaw, pitch, and roll rotations

A 3D body can be rotated about three orthogonal axes, as shown in Figure 3.8. Borrowing aviation terminology, these rotations will be referred to as yaw, pitch, and roll:

1. A *yaw* is a counterclockwise rotation of α about the z-axis. The rotation matrix is given by

$$R_z(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.39}$$

2.

3. Note that the upper left entries of $R_z(\alpha)$ form a 2D rotation applied to the x and y coordinates, whereas the z coordinate remains constant.

4. A *pitch* is a counterclockwise rotation of β about the y-axis. The rotation matrix is given by

$$R_y(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix}. \tag{3.40}$$

5.

6. A *roll* is a counterclockwise rotation of γ about the x-axis. The rotation matrix is given by

$$R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix}. \tag{3.41}$$

7.

Each rotation matrix is a simple extension of the 2D rotation matrix, (3.31). For example, the yaw matrix, $R_z(\alpha)$, essentially performs a 2D rotation with respect to the x and y coordinates

while leaving the z coordinate unchanged. Thus, the third row and third column of $R_z(\alpha)$ look like part of the identity matrix, while the upper right portion of $R_z(\alpha)$ looks like the 2D rotation matrix.

The yaw, pitch, and roll rotations can be used to place a 3D body in any orientation. A single rotation matrix can be formed by multiplying the yaw, pitch, and roll rotation matrices to obtain

$$R(\alpha,\beta,\gamma) = R_z(\alpha)\, R_y(\beta)\, R_x(\gamma) = \begin{pmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\,c \end{pmatrix} \quad (3.42)$$

It is important to note that $R(\alpha,\beta,\gamma)$ performs the roll first, then the pitch, and finally the yaw. If the order of these operations is changed, a different rotation matrix would result. Be careful when interpreting the rotations. Consider the final rotation, a yaw by α. Imagine sitting inside of a robot $\mathcal{A}$ that looks like an aircraft. If $\beta = \gamma = 0$, then the yaw turns the plane in a way that feels like turning a car to the left. However, for arbitrary values of β and γ, the final rotation axis will not be vertically aligned with the aircraft because the aircraft is left in an unusual orientation before α is applied. The yaw rotation occurs about the z-axis of the world frame, not the body frame of $\mathcal{A}$. Each time a new rotation matrix is introduced from the left, it has no concern for original body frame of $\mathcal{A}$. It simply rotates every point in $\mathbb{R}^3$ in terms of the world frame. Note that 3D rotations depend on three parameters, α, β, and γ, whereas 2D rotations depend only on a single parameter, θ. The primitives of the model can be transformed using $R(\alpha,\beta,\gamma)$, resulting in $\mathcal{A}(\alpha,\beta,\gamma)$.

Determining yaw, pitch, and roll from a rotation matrix

It is often convenient to determine the α, β, and γ parameters directly from a given rotation matrix. Suppose an arbitrary rotation matrix

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{3.43}$$

is given. By setting each entry equal to its corresponding entry in (3.42), equations are obtained that must be solved for α, β, and γ. Note that $r_{21}/r_{11} = \tan\alpha$ and $r_{32}/r_{33} = \tan\gamma$. Also, $r_{31} = -\sin\beta$ and $\sqrt{r_{32}^2 + r_{33}^2} = \cos\beta$. Solving for each angle yields

$$\alpha = \tan^{-1}(r_{21}/r_{11}), \tag{3.44}$$

$$\beta = \tan^{-1}\left(-r_{31}/\sqrt{r_{32}^2 + r_{33}^2}\right), \tag{3.45}$$

and

$$\gamma = \tan^{-1}(r_{32}/r_{33}). \tag{3.46}$$

There is a choice of four quadrants for the inverse tangent functions. How can the correct quadrant be determined? Each quadrant should be chosen by using the signs of the numerator and denominator of the argument. The numerator sign selects whether the direction will be above or below the x-axis, and the denominator selects whether the direction will be to the

left or right of the y-axis. This is the same as the 2function in the C programming language, which nicely expands the range of the arctangent to $[0, 2\pi)$. This can be applied to express (3.44), (3.45), and (3.46) as

$$\alpha = 2(r_{21}, r_{11}), \tag{3.47}$$

$$\beta = 2\Big(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}\Big), \tag{3.48}$$

and

$$\gamma = 2(r_{32}, r_{33}). \tag{3.49}$$

Note that this method assumes $r_{11} \neq 0$ and $r_{33} \neq 0$.

The homogeneous transformation matrix for 3D bodies

As in the 2D case, a homogeneous transformation matrix can be defined. For the 3D case, a 4×4 matrix is obtained that performs the rotation given by $R(\alpha, \beta, \gamma)$, followed by a translation given by x_t, y_t, z_t. The result is

$$T = \begin{pmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & x_t \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & y_t \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma & z_t \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.50}$$

Once again, the order of operations is critical. The matrix T in (3.50) represents the following sequence of transformations:

1. Roll by γ
2. Pitch by β
3. Yaw by α
4. Translate by (x_t, y_t, z_t).

The robot primitives can be transformed to yield $\mathcal{A}(x_t, y_t, z_t, \alpha, \beta, \gamma)$. A 3D rigid body that is capable of translation and rotation therefore has six degrees of freedom.