Appendices

A. Giavaras

Contents

Appendices

A Coordinate Transformations

In order to track how a rigid body such as a seld-driving car moves, we need to know how we can express its motion using mathematical tools and notation. In this section, we will review how reference frames affect vector coordinates, compare and contrast different rotation representations, and present several reference frames.

A.1 Vectors

Generally, kinematic variables, such as the velocity, are represented in the form of a vector, with both magnitude and direction. In Figure 1, the vector v is presented with a green arrow in a two-dimensional coordinate frame. We have two coordinate frames displayed here. The body frame, defined by axes b_1 and b_2, and the inertial frame defined by axes e_1 and e_2, both in the 2D plane.

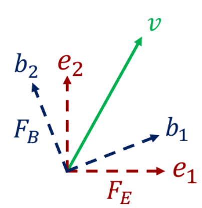


Fig. 1: Schematic of vector rotation in 2D.

By definition a vector is a geometric object that has a magnitude and a direction. Often, we treat the concept of a vector interchangeably with the concept of vector coordinates, or the set of numbers that represent the direction and maginitude of the vector. This, however, is not necessarily correct. If we imagine that the vector is fixed in space, then its coordinates will change depending on the way in which we observe it. More precisely, the same vector quantity will have different coordinates depending on which coordinate frame or reference frame we choose to express it in.

Remark A.1. **Notation**

Let us introduce some notation that will be useful susequently. In frame a, the vector $\mathbf{r}$ has notation $\mathbf{r}_a$. Likewise in frame b, it has the coordinates

$\mathbf{r}_b$. The coordinates of the vector in the two different frames are related through a rotation matrix $\mathbf{C}_{ba}$

$$\mathbf{r}_b = \mathbf{C}_{ba}\mathbf{r}_a \tag{1}$$

The rotation matrix tells us exactly how one frame is rotated with respect to the other.

A.2 2D rotations

Let us first discuss two dimensional rotations. Assume the two coordinate frames; frame e and frame b, which have the same fixed origin. Frame b is rotated by some angle θ relative to frame e. We can then define the rotational matrices C_{eb}, which transforms vectors from the frame b to the frame e and C_{be} which projects the frame e onto frame b using the angle θ as shown.

$$C_{EB} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}, \quad C_{BE} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{2}$$

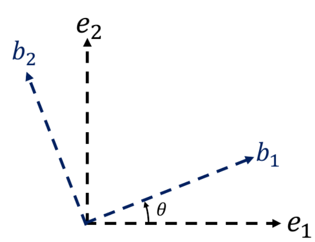


Fig. 2: Rotation angle θ.

Now, let's extend our example to include a translation. Here, we see a two-wheeled robot and we'd like to represent the position of a point P observed by the robot in the robot body frame b, with respect to the inertial frame e. The position of the robot with respect to the inertial frame is (x, y), and the orientation of the robot once again is θ.

The following equations relate the location of point P in the body coordinates P_b and the inertial frame P_e. Note that in general to transform one point from one coordinate to the other coordinate frame, body to inertial and vice versa, requires two terms. The translation of the origin O_{be} and O_{eb} in this case, and the rotation of the axis C_{eb} and C_{be}. Finally, we can summarize the transformation between two coordinate frames using homogeneous coordinates, which lead to a compact matrix multiplication to apply the transformation.

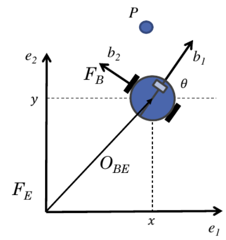


Fig. 3: Robot rotation and translation.

$$P_b = C_{eb}(\theta)P_e + O_{eb} \tag{3}$$

where O_{eb} is the translation term expressed in body frame. Similarly,

$$P_e = C_{be}(\theta)P_b + O_{be} \tag{4}$$

Often, it will also be useful to know how the coordinates of points change as we move from one reference frame to another. For example, we may know the position of a building in some frame, and now we would like to know its position in our current vehicle frame. To compute this, we use vector addition making sure to express all of the coordinates in the same reference frame. We will use superscripts on the coordinates to indicate the start and end point of a 3D vector, again from right to left, and a subscript to indicate the frame in which this is expressed just as before. We can manipulate this expression to solve for the coordinates in the vehicle frame or an appropriate inertial frame, for example, see Figure 4.

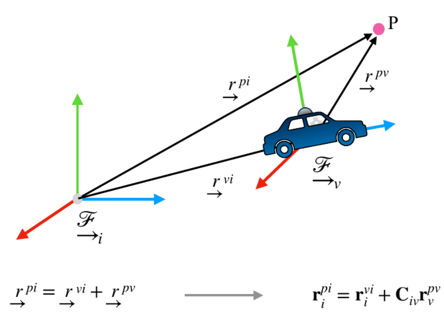


Fig. 4: Schematic of coordinate frames.

A.3 Direction cosine matrix

A critical component of tracking reference frames is tracking their orientation or rotation with respect to some base reference frame. Rotations are particularly tricky mathematical objects and they can be the source of major bugs if not dealt with carefully and diligently. There are many different ways to represent rotations. The most common is to use a three by three rotation matrix as we've done before. This matrix defines the relationship between the basis vectors of two reference frames in terms of dot products. For this reason, it's often called the **direction cosine matrix**.

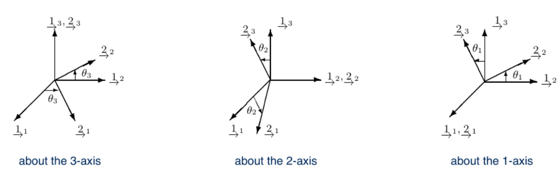
> *Remark* A.2. **Inverse of $\mathbf{C}_{ba}$**
>
> An important property to remember is that the inverse of a rotation matrix is just its transpose.

A.4 Euler angles

Another way of representing a rotation is using three numbers called Euler angles. These angles represent an arbitrary rotation as the composition of three separate rotations about different principal axes as

$$\mathbf{C}(\theta_3, \theta_2, \theta_1) = \mathbf{C}_3(\theta_3)\mathbf{C}_3(\theta_2)\mathbf{C}_1(\theta_1) \tag{5}$$

Figure 5 shows schematically the individual rotations.



$$\mathbf{C}_3 = \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \quad \mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix}$$

Fig. 5: Euler angles.

Euler angles are attractive in part because they are a parsimonious representation requiring only three parameters instead of nine for a full rotation matrix. Unfortunately, Euler angle representations are subject to what are called singularities. Singularities complicate state estimation because they represent particular rotations from which to Euler angles are indistinguishable. Neither quaternions, see section A.5 nor rotation matrices, section A.3, suffer from this problem at the expense of using more parameters.

A.5 Unit quaternion

A second way to represent rotations is to use something called unit quaternions. Quaternions are an interesting mathematical topic in their own right. However, it is sufficient to note that a unit quaternion can be represented as a four-dimensional vector of unit length that parameterizes a rotation about an axis defined by the vector $\mathbf{u}$, and an angle ϕ about that vector, see Figure 6.

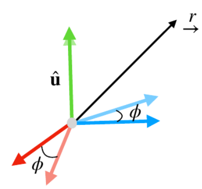


Fig. 6: The vector $\mathbf{u}$, and the angle ϕ.

Thus, we have the follwing equations

$$\mathbf{q} = \begin{bmatrix} q_w \\ \mathbf{q}_\nu \end{bmatrix} = \begin{bmatrix} \cos(\frac{\phi}{2}) \\ \hat{\mathbf{u}} \sin(\frac{\phi}{2}) \end{bmatrix}, \quad ||\mathbf{q}|| = 1 \tag{6}$$

We can convert a quaternion to a rotation matrix by using the followin algebraic expression.

$$\mathbf{r}_b = \mathbf{C}(\mathbf{q}_{ba})\mathbf{r}_a \tag{7}$$

where

$$\mathbf{C}(\mathbf{q}) = (q_w^2 - \mathbf{q}_\nu^T \mathbf{q}_\nu)\mathbf{I} + 2\mathbf{q}_\nu \mathbf{q}_\nu^T + 2q_w[\mathbf{q}_\nu]_\times \tag{8}$$

with

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{9}$$

Quaternions have the advantage that they do not suffer from singularities and they only need four parameters instead of nine in order to represent 3D rotations.

Each represenation has advantages and disadvantages. A rotation matrix can represent any rotation but requires nine parameters and has six constraints. A unit quaternion can also be used to represent any rotation, but it also has a constraint. To use a unit quaternion to actually rotate a vector, we also require some additional algebra beyond simple matrix multiplication. Finally, Euler angles are unconstrained, intuitive to visualize and use only three parameters, but are subject to singularities. Figure 7 summarizes the three approaches

	Rotation Matrix	Unit quaternion	Euler angles
Expression	$\mathbf{C}$	$\mathbf{q} = \begin{bmatrix} \cos\frac{\phi}{2} \\ \hat{\mathbf{u}}\sin\frac{\phi}{2} \end{bmatrix}$	$\{\theta_3, \theta_2, \theta_1\}$
Parameters	9	4	3
Constraints	$\mathbf{CC}^T = \mathbf{1}$	$\lvert\mathbf{q}\rvert = 1$	*None**
Singularities?	No	No	*Yes!*

Fig. 7: Comparison of rotation representation approaches.

To summarize, vector quantities can be expressed in different reference frames through rotations and translations. Rotations can be parameterized by rotation matrices, quaternions, or Euler angles, each of which has advantages and disadvantages.

B Reference Frames

In this section we will review four commonly used reference frames that are used in order to localize a rigid body.

B.1 Earth-Centered Inertial Frame or ECIF

The first frame is the Earth-Centered Inertial Frame or ECIF. This frame has its origin at the center of the earth, the z axis points true north, and the x and y axis are fixed with respect to the very distant stars. This means that although the earth rotates about the z axis, the x and y axes do not move.

B.2 Earth-Eentered Earth-Fixed Frame, or ECEF

Next, the Earth-Eentered Earth-Fixed Frame, or ECEF is just like ECIF except that its x axis is aligned with the Prime Meridian and spins with the earth. The y axis is determined by the right-hand rule.

The difference between the ECEF and ECIF is that the former is fixed to the earth, while the ECIF, is fixed with respect to the distance stars.

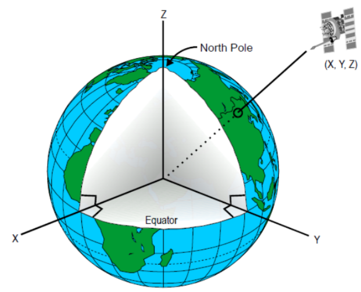


Fig. 8: Schematic of ECIF.

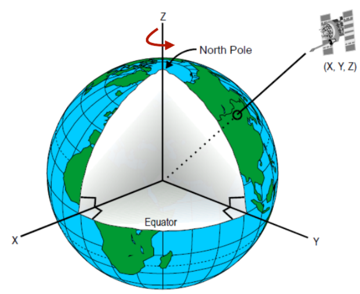


Fig. 9: Schematic of ECEF.

B.3 North East Down or NED

Although ECEF and ECIF are useful when we discuss satellites and inertial sensing onboard aircraft, for practical car applications, we will usually want to use a frame that is fixed with respect to the ground. For this, we will use what we referred to as the navigation frame. A very common navigation frame is one that is attached to some germane starting point and aligned with north, east, and down abbreviated as NED. Figure 10 shows schematically the NED frame

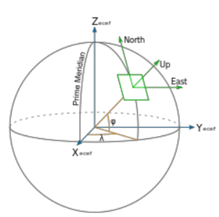


Fig. 10: Schematic of NED.

B.4 Body and sensor frames

Often, we also need to think about a sensor frame that is rigidly attached to a sensor like a LiDAR, a GPS receiver, or an inertial measurement unit. This frame will typically be distinct from the general vehicle frame, which can be placed anywhere on the vehicle that is convenient, at the center of mass for example. For localization, we will often ignore the distinction between the vehicle and sensor frame and assume that if we can track the sensor, we should be able to track any point on the vehicle given proper calibration.

References

[1] Åström K. J., Murray R. M. *Feedback Systems. An Introduction for Scientists and Engineers*

[2] Philip , Florent Altche1, Brigitte dAndrea-Novel, and Arnaud de La Fortelle *The Kinematic Bicycle Model: a Consistent Model for Planning Feasible Trajectories for Autonomous Vehicles?* HAL Id: hal-01520869, https://hal-polytechnique.archives-ouvertes.fr/hal-01520869

[3] Marcos R. O., A. Maximo *Model Predictive Controller for Trajectory Tracking by Differential Drive Robot with Actuation constraints*