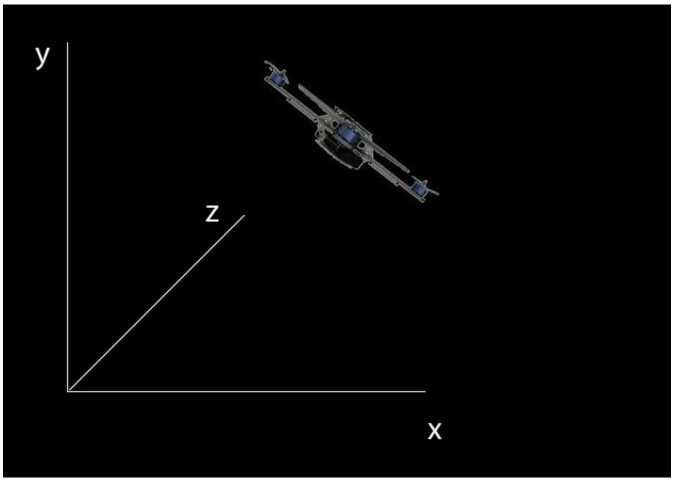
Lecture 2A.1

Transformations

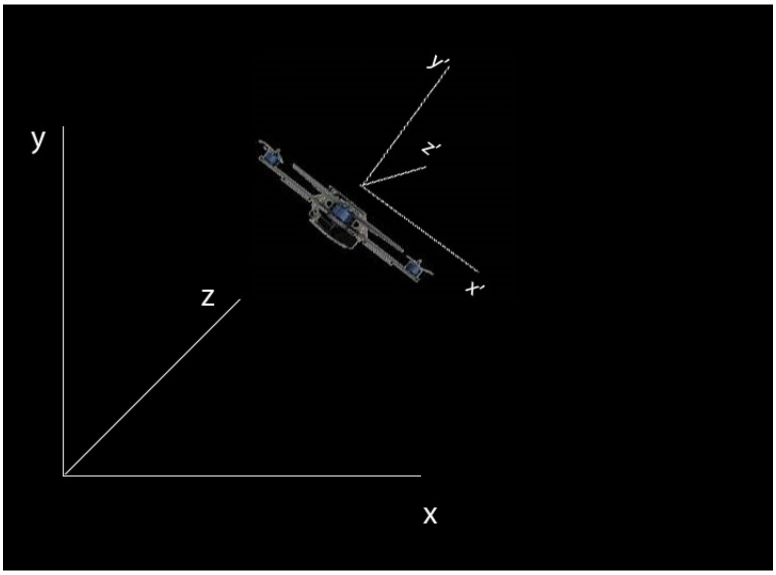
This module will explore the fundamental concepts required to describe three-dimensional motion - both the position and orientation of rigid objects flying through space. We will develop the basic tools to describe three-dimensional displacements through rigid body transformations.

Reference Frames

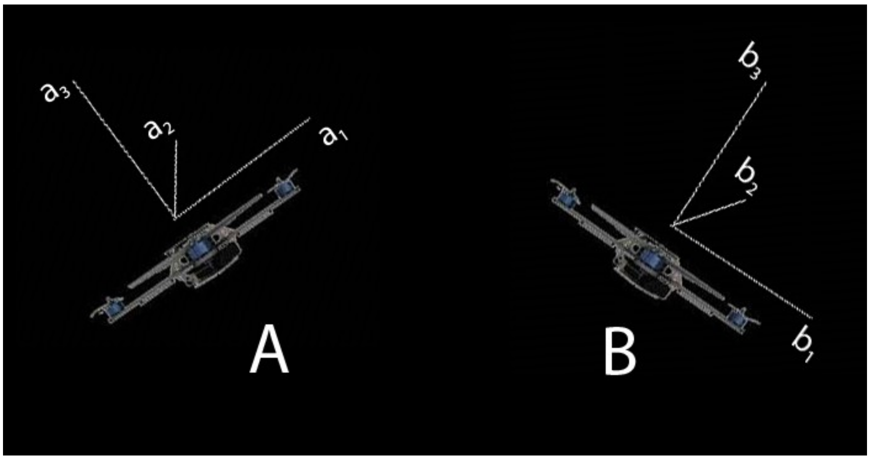
The key concept that underpins these techniques is the **reference frame**, here shown with orthogonal axes x, y, and z. Every position and orientation is described relative to some reference frame:



Each quadrotor can have its own reference frame, distinct from the world reference frame, here shown with the orthogonal axes x', y' and z':



Let's consider two distinct positions and orientations for a quadrotor. We'll call these two positions & orientations A and B, and each will have its own reference frame.



In reference frame {A}, we choose three linearly-independent basis vectors a_1, a_2, and a_3. While these don't have to be mutually orthogonal, it's convenient to choose them to be mutually orthogonal. The key idea though, is that they must at least be linearly independent. Similarly, in frame {B} we have three linearly-independent vectors, b_1, b_2, and b_3.

Now any vector, v, in three-dimensional space can be written as a linear combination of these independent vectors. In frame {A}, we would write it as a linear combination of vectors a_1, a_2, and a_3:

$$\vec{v} = v_1\vec{a}_1 + v_2\vec{a}_2 + v_3\vec{a}_3$$

In Frame {B}, we would write it in terms of b1, b2, and b3.

$$\vec{v} = v_1\vec{b}_1 + v_2\vec{b}_2 + v_3\vec{b}_3$$

Notation

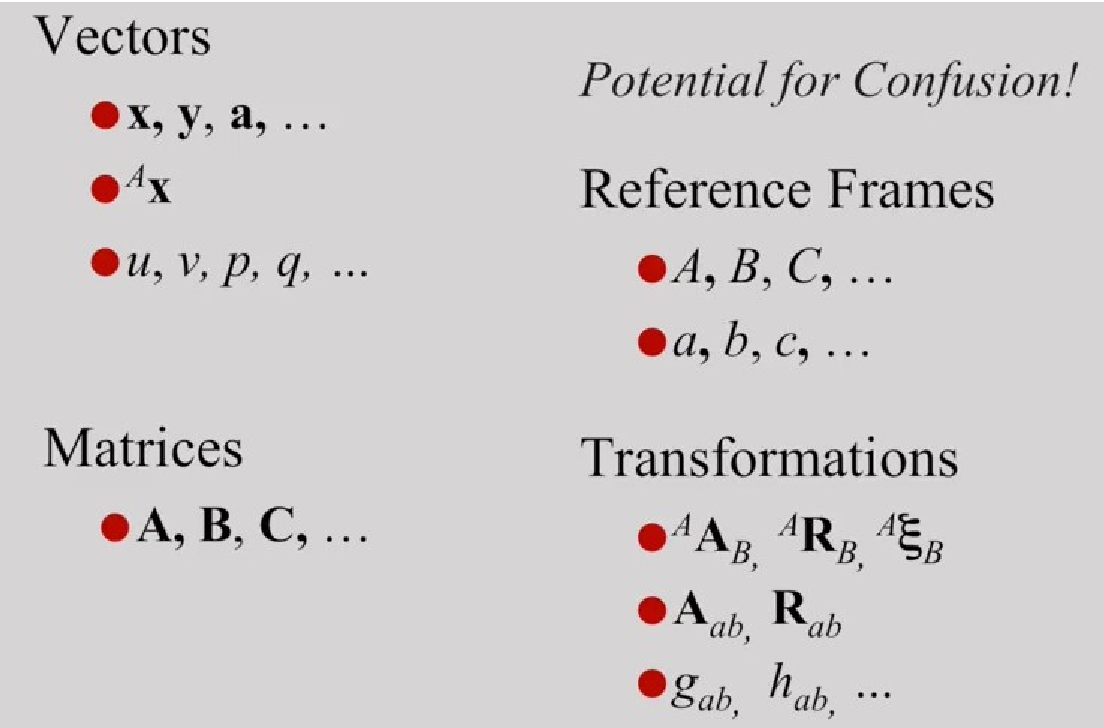
Typically we'll use boldface to denote vectors, and we may use a leading superscript to denote the frame in which we are writing that vector, e.g. ${}^{A}\mathbf{x}$. In some cases, you may also see lower cased italicized letters denoting vectors.

Similarly for reference frames, we use italicized letters but these are usually upper case, e.g. *A*, *B*, *C*… However, depending which texts or papers you read, you might also see italicized lowercase letters denoting reference frames.

There's a lot of potential for confusion. And you will often have to go back and 'recalibrate' yourself with new notation when you read something new.

Matrices are denoted by uppercase, boldface letters.

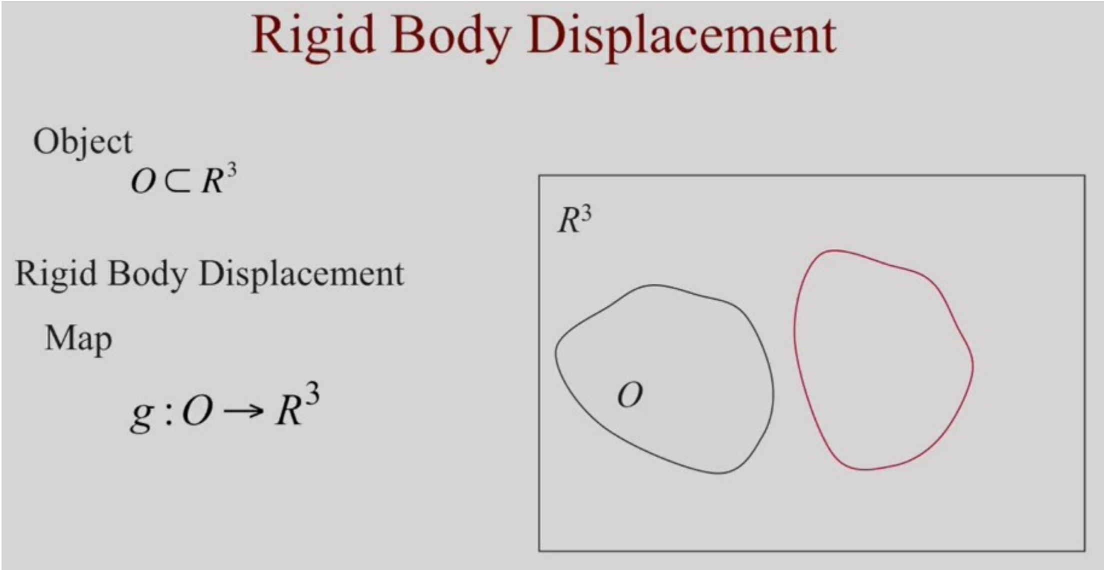
We will also talk about ‘transformations’. This is the notation we use to describe how vectors in one frame can be written in another frame. For example, if we have an uppercase letter A with a leading superscript A and a trailing superscript B, ${}^{A}A_{B}$, that usually indicates that we have an object that transforms vectors in frame {B} into vectors in frame {A}. A lowercase italicised letter with two trailing subscripts, a & b, g_{ab} denotes a transformation from frame {B} into frame {A}.



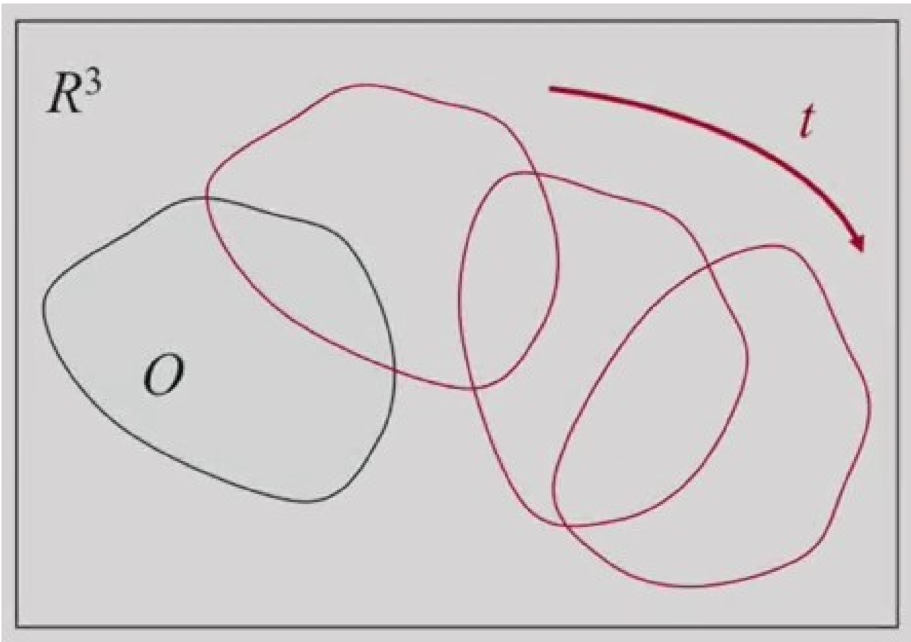
Rigid Body Transformations

A rigid body is an idealised solid body in which deformation is neglected. This means that the distance between any two given points of a rigid body remains constant, regardless of any external forces exerted on it.

Let's talk about rigid body displacements. A rigid body, O, is a collection of points in three dimension. It is a subset of R^3. This object can have multiple positions and orientations. A rigid body displacement is simply a map from this collection of points in the object, to its physical manifestation in R3.



This object may occupy different positions and orientations over time. Accordingly, there will be different rigid body displacements. Each of these is a map of the points in O to their physical manifestations in the real space, R^3.

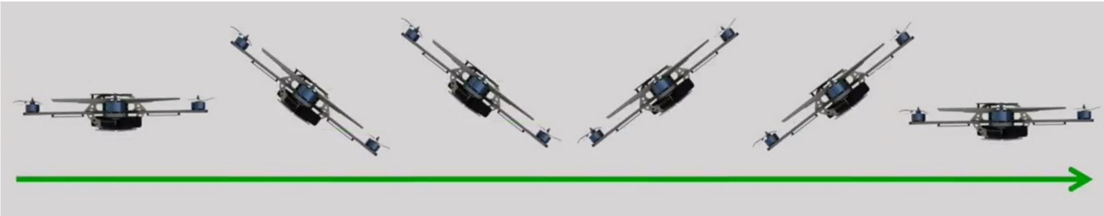


As the body moves through space, that motion is described by a continuous family of maps, so the displacement g is now parameterised by time:

$$g(t) : O \rightarrow R^3$$

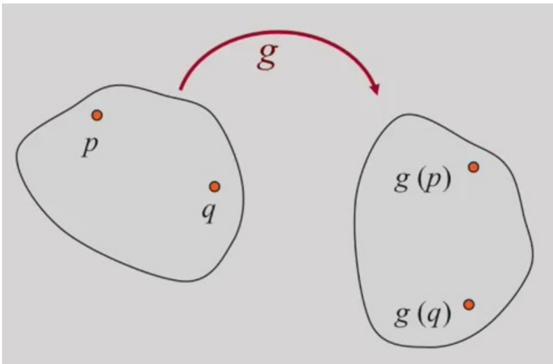
The collection of points in O moves from one position and orientation to another over time, and this is a continuous set of displacements.

This is what we see in a quadrotor. It starts in a horizontal position, moves to another position, accelerating, changing its orientation, changing the direction of the thrust and then reversing the direction of the thrust by pitching back, and then slowing down to the goal position.



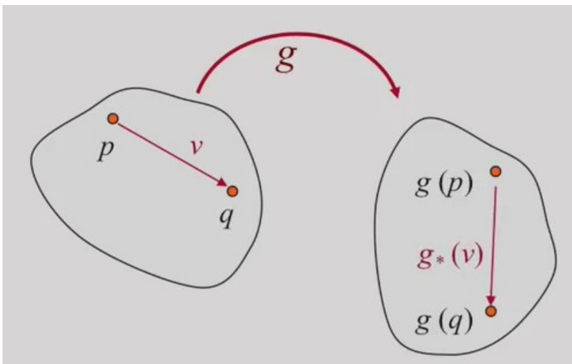
Each of these snapshots is a displacement, and this sequence of displacements represents a continuous family of displacements.

Consider a single point, p, on the rigid body. When the rigid body is displaced, the point, p, gets displaced into a new point, which we will call g(p):



A displacement is essentially a transformation of points. Of course, there are infinite points in a rigid body. If we have a second point, q, the same displacement will take q and move it into a new point, g(q).

Every pair of points defines a vector. The vector **v** emanates from p and terminates at q. Since g moves p to g(p), & q to g(q), it will also move the vector **v** to a new vector, which we call **g*(v)**.



So the displacement g induces a map on vectors. Remember, the displacement acts on points, but g* acts on vectors.

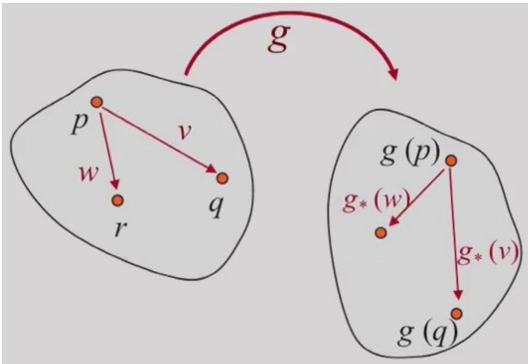
So, what makes the map, g, a *rigid body displacement*?

Two properties must be satisfied. First, the distance between any pair of points remains unchanged in a rigid body displacement. In fact, as explained above, that is the definition of a rigid body. So if we calculate the distance between p and q, and the distance between g(p) and g(q) after the displacement, those two distances must be identical:

$$\|g(p)-g(q)\|=\|p-q\|$$

The second property that rigid body displacements must satisfy has to do with cross-products of vectors that are attached to the rigid body.

So far, we have a single vector emanating from p and terminating in q. Let's choose a third point, r, and generate a second vector going from p to r. What happens to this vector, **w**, when the body gets displaced to its new position and orientation?



g* acts on **w** to generate a new vector, **g*(w)**. Now we can look at the cross-product of **v** and **w** and ask what happens to that cross-product when it goes to **v** cross **w**.

$$g^*(v)\times g^*(w)=g^*(v\times w)$$

If we look at the mapping of **v** into **g*(v)** and the mapping of **w** into **g*(w)**, we can compute the cross-product of **g*(v)** and **g*(w)**. It turns out that the cross product remains the same whether we calculate it before the displacement or after the displacement, provided the displacement is rigid. Thus, cross-products are preserved.

So if we look at the family of maps which preserve distances and cross-products, we essentially get rigid-body displacements.

So, to summarise, the two properties are:

1. Lengths are preserved.
2. Cross-products are preserved.

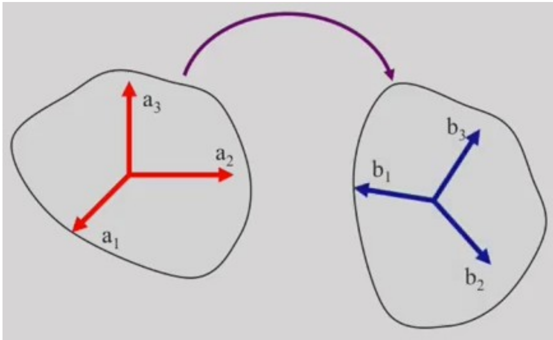
For a rigid body displacement we should be able to prove that, orthogonal vectors map to orthogonal vectors, and further, that g* preserves inner products. That is, if we compute the inner product of two vectors before the displacement, and the inner product of the same two vectors after the displacement, those inner products are the same.

- orthogonal vectors are mapped to orthogonal vectors

- g_* preserves inner products

$$g_\star(v) \cdot g_\star(w) = v \cdot w$$

A set of mutually orthogonal unit vectors attached to the rigid body will remain mutually orthogonal, and will remain unit vectors, after the displacement.



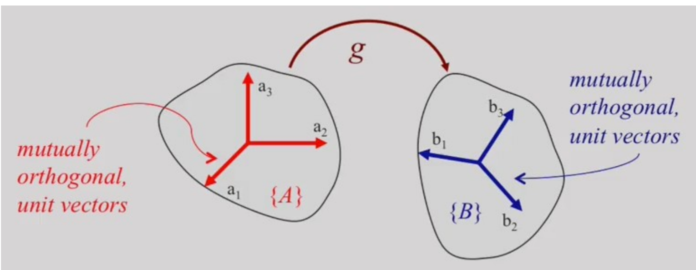
In summary, rigid body transformations, or rigid body displacements, satisfy two important properties. They preserve lengths and they preserve cross-products.

Note: The terms *rigid-body displacement* and *rigid-body transformation* are frequently used interchangeably. There is an important semantic difference between the two.

- *Transformations* generally refer to relationships between reference frames attached to two different rigid bodies,
- *Displacements* describe relationships between positions and orientations of a frame attached to the rigid-body as it moves around.

We have now seen an abstract description of what a rigid-body displacement is. Let's explore how we can actually do calculations to describe rigid body displacements, and then manipulate these displacements.

We start with the assumption that we have mutually orthogonal unit vectors attached to every rigid body.



If it's a transformation, we're referring to two different rigid bodies. If it's a displacement, it is two distinct positions and orientations of the same rigid body.

We can write the mutually orthogonal unit vectors in one frame as a linear combination of the mutually orthogonal unit vectors in the other frame, thus:

$$\begin{aligned}
\mathbf{b}_1 &= R_{11}\mathbf{a}_1 + R_{12}\mathbf{a}_2 + R_{13}\mathbf{a}_3 \\
\mathbf{b}_2 &= R_{21}\mathbf{a}_1 + R_{22}\mathbf{a}_2 + R_{23}\mathbf{a}_3 \\
\mathbf{b}_3 &= R_{31}\mathbf{a}_1 + R_{32}\mathbf{a}_2 + R_{33}\mathbf{a}_3
\end{aligned}$$

Here we see b_1, b_2, b_3 expressed as linear combinations of a_1, a_2, a_3. The coefficients are given by R_{11}, R_{12}, … and so on. This collection of nine coefficients can be gathered into a matrix, and we call this matrix a ***rotation matrix***.

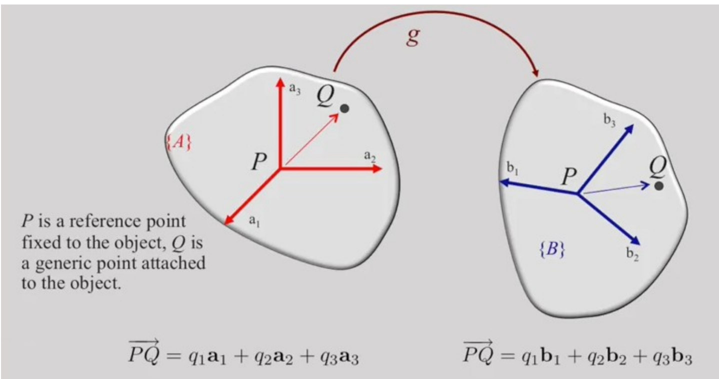
$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

The rotation matrix has several interesting properties.

1. It turns out that this matrix is **orthogonal**, which means if we take this matrix and multiply it by its transpose, we get the identity matrix. If we take the transpose of the matrix and multiply it by the matrix, we also get the identity.
2. The matrix is **special orthogonal**. In other words, the determinant of the matrix is +1.
3. Rotation matrices are **closed under multiplication**. The product of any two rotation matrices is also a rotation matrix.
4. The inverse of a rotation matrix is also a rotation matrix.

These properties will come in useful as we progress.

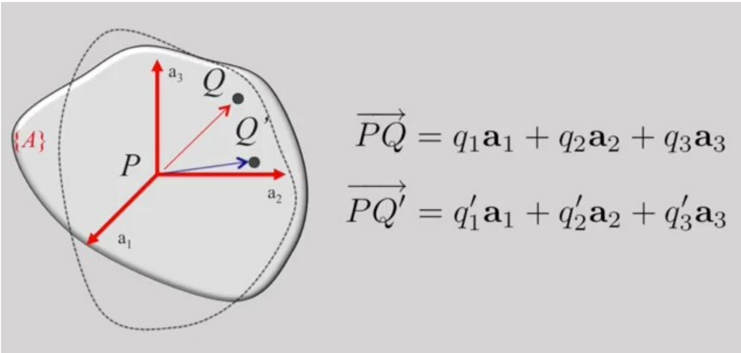
Now let's start looking at the structure of a rotation matrix. We have a rigid body, and the diagram shows two distinct positions and orientations of that body. We also have a point, q, which we will follow as it moves from one position to another:



The position vector **PQ** (where P is the origin) can be written in two different ways, depending on which snapshot we consider. In the first case, we write it as a linear combination of $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$, and in the second case, we've written it as a linear combination of $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$.

Let's focus on the vector, **PQ**.

I have two distinct position vectors, and they are both **PQ**. We can translate the rigid-body so that the two vectors have the same origin. We denote one vector as **PQ** and the second vector as **PQ'**.



Again, we can express the vector as a linear combination of a_1, a_2, and a_3. But now I'm going to get two different sets of coefficients. If I look at **PQ**, it has coefficients q_1, q_2 and q_3. If I look at **PQ'** prime, it has coefficients q_1', q_2', and q_3'.

I can ask how to write q_1', q_2', and q_3' as a function of q_1, q_2, q_3. Is there a matrix that connects them? And I can also ask the reverse question. Is there a matrix that connects q_1, q_2, q_3 to q_1', q_2', and q_3'?

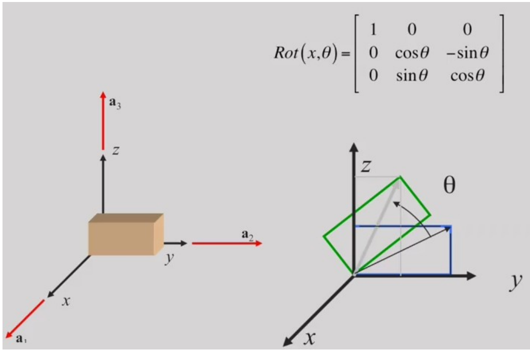
It turns out that there is, and this matrix is exactly the same one that we looked at earlier: the **rotation matrix**:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \underbrace{\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}}_{\textit{rotation matrix}} \begin{bmatrix} q_1' \\ q_2' \\ q_3' \end{bmatrix}$$

$$\begin{bmatrix} q_1' \\ q_2' \\ q_3' \end{bmatrix} = \underbrace{\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}}_{\textit{rotation matrix}} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

If we know how to write the mutually orthogonal unit vectors in one frame as a function of the vectors in the other frame, and I can do this by calculating the rotation matrix, the same matrix tells me how to transform vectors in one frame to another frame.

As an example, let's first look at something very simple. Consider a rectangular prism, with its axes aligned with the x, y, and z axes (or the a_1, a_2, and a_3 unit vectors).

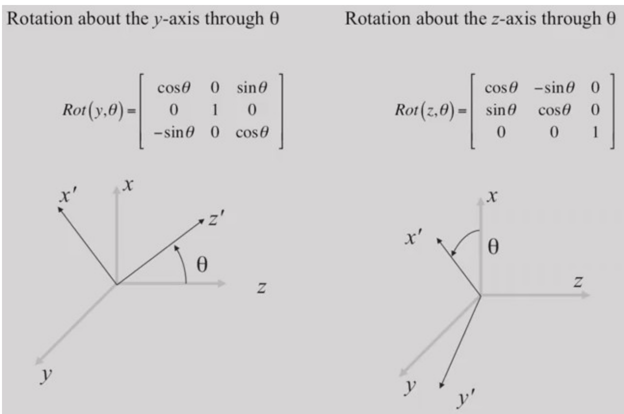


Let's rotate this rigid body about the x axis, through an angle, θ. We only need one parameter to describe this rotation, and that's the angle, θ. We can write this rotation matrix and we find that it has nine numbers:

$$Rot(x,\theta)=\begin{bmatrix}1&0&0\\0&\cos\theta&-\sin\theta\\0&\sin\theta&\cos\theta\end{bmatrix}$$

Four of the numbers are zero, there's an identity at the top-left, and the other numbers are essentially sines and cosines of the angle of rotation, θ. We can verify this through geometry - the calculations are actually quite simple.

Here are two more examples of rotation matrices. The one of the left describes a rotation through an angle, θ, about the y axis. The one on the right describes a rotation through an angle, θ, about the z axis:



Once more, in each case, there are four zeros, a one, and everything else is either the cosine or a sine of the rotation angle, θ.

In general, rotation matrices can look more complicated, but when you have rotations about the x, y, or z axes, they assume these very simple forms.