Attitude with an Attitude
An Informal Intro to Spacecraft Attitude Determination

Brian Leung

University of Michigan, Class of 2022

November 15, 2020

Contents

Chapter 1

Preface

1.1 Prerequisites

You're welcome to dive in and learn as you go, but knowing some **Physics 1/Dynamics** and **Linear Algebra** will help immensely. Honestly, with how applicable it is, you should just learn linear algebra in general. A class at the undergraduate level should more than enough. I personally learned using UT Austin's "Linear Algebra - Foundations to Frontiers" on edX and ran through the gauntlet of math exercises. Otherwise, you can learn using your own methods.

1.2 Message to the Reader

At the time of writing this, I am not a seasoned spacecraft engineer nor a cutting edge researcher (here's hoping). For now, I'm just a green undergrad in Aerospace Engineering, hoping that my understanding will help you out. I hope you learn something new and feel a little more comfortable in this field!

1.3   Acknowledgement

I owe much thanks to CU Boulder's "Spacecraft Dynamics and Control Specialization" on Coursera, and Professor Hanspeter Schaub, the professor of the course. Seriously, it's great. If you've got the time and access, I highly recommend you look through it.

1.4   What This Text Offers

I really want to emphasize that this text is <u>not</u> a university textbook level reference, bursting at the seams with advanced terminology and derivations. This will not delve into dynamic attitude determination or control theory either.

    This text is essentially CU Boulder's Specialization, Part 1, distilled down to the key takeaways. This will give you a quick, informal intro to the fundamentals of spacecraft attitude description and determination, avoiding lengthy proofs and derivation. It offers conceptual questions along the way to test your knowledge. It can act as a springboard into more advanced topics for the aspiring spacecraft engineer.

Chapter 2

What is Attitude Anyways?

2.1 Definition

Hint: It's not a state of mind. A fairly straightforward definition for attitude is:

**ATTITUDE: A body's angular orientation with respect to a reference frame.**

    A body just refers to something like a plane, spacecraft, or satellite. Angular orientation? Pretty self-explanatory. The important part of this definition is the reference frame.

    Let me pose this question to you. If you're floating around in an infinite black void, how do you decide which way is up? or down? Of course, the problem is a bit more complex than that, but it's a question you can't answer without the help of a *reference frame.*

2.1.1 Frames?

When I say reference frame, or frames in general, I mean coordinate frames: XYZ, $\hat{i}\hat{j}\hat{k}$, 3 basis vectors, etc. You can pretty much glue a

coordinate frame onto anything your heart desires. Attaching frames to things helps ground them in a mathematical context, and that helps in more reasons than I can name. For example, it's a lot easier to say things like "I'm about 10 meters south and 1 meter east of you with respect to your NED frame". This is just a rough example, but there are plenty of other good reasons to put frames on bodies.
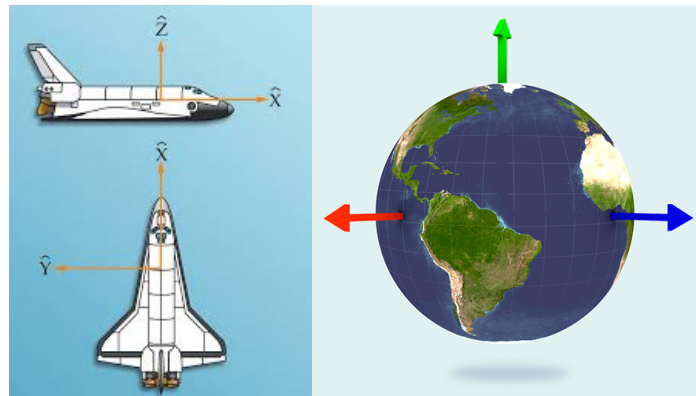
A Frame on the Space Shuttle (L)
ECEF (Earth Centered, Earth Fixed) Frame (R)

BODY: Refers to object of interest (e.g. spacecraft, satellite)
When someone says "Body Frame" that refers to the frame glued to your satellite, space shuttle, UFO, rover. Basically, the thing you're trying to keep track of.

INERTIAL: Refers to reference (e.g. Earth)
When someone says inertial frame, they always mean reference. In most basic problems, the Earth and its variety of frames is what you'll refer to as inertial. However, you can also set the inertial frame to something that's not the Earth, like the ISS.

2.1.2   It's All Relative

I must emphasize that *attitude is all relative*. Without knowing and specifying a reference frame, the description of your attitude is basically useless.

2.1.3   Why Worry About Attitude Anyways?

Trying to brush off attitude determination as unimportant is similar to asking "Why do planes have to be headed in the right direction?". Depending on the project/mission, the importance of making sure your object is pointed in the desired direction ranges anywhere from marginal (super basic cubesats) or critical (space telescopes, comms satellites, space shuttle). But 9 times out of 10, attitude is pretty high on the priority list. This topic also has plenty of applications outside of the aero sphere, such as in video games, robotics, and biomechanics.

2.2   4 Basic Truths

1. **You need a minimum of 3 coordinates to describe relative angular displacement in 3D.** There's likely some in-depth proof showing why this is the case, but I'm not qualified enough to comment on it. The best example of a 3 coordinate set is Euler Angles, covered in Ch 4.

2. **A 3 coordinate set will have at least 1 geometric orientation where the coordinates are <u>singular</u>.** When an attitude description becomes **singular**, it's best described as acting ... weird. As in, the math doesn't behave very nicely (divisions by 0, non-uniqueness, etc.). As a real life example, think gimbal lock. The gimbal, when turned a certain way, loses degrees of freedom.

3. **At/near the singularities, the kinematic differential equations are also singular.** When you get close to a singularity in your attitude description, the kinematic differential equations of your attitude description also turn feral. These won't be too heavily covered in this text so don't worry too much about them.

4. **Redundant sets of 4 or more coordinates exist that are universally valid.**

Chapter 3

DCM

3.1 Definition

It doesn't stand for dichloromethane. And please, don't call it a DCM matrix.

DIRECTION COSINE MATRIX: A transformation matrix that converts from one reference frame to another.

We consider the DCM the "mother of all attitude parameterizations", quoting Professor Schaub. When I mean *attitude parameterization*, I mean descriptions your attitude, such as quaternions, Euler angles, etc. Every DCM maps to a parameterization of your choice and every parameterization maps back to a DCM. You could say it's the heart of the parameterization web. With DCMs, we directly interact with the basis vectors of the frames themselves. Before we move any further, let's get some *formalisms* out of the way.

3.1.1 Funky Formalisms

When describing the body frame, I'll use the letter b. Describing the basis vectors that make up the body frame, I'll say $\hat{b}_1, \hat{b}_2, \hat{b}_3$. Likewise, describing the inertial frame basis vectors uses the letter n, like so: $\hat{n}_1, \hat{n}_2, \hat{n}_3$. Constantly writing out $\hat{b}_1, \hat{b}_2, \hat{b}_3$ is gonna get old real fast,

so lets save our hands some work and assemble them into something called a *vectrix*.

$$\{\hat{b}\} = \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix} \quad \{\hat{n}\} = \begin{bmatrix} \hat{n}_1 \\ \hat{n}_2 \\ \hat{n}_3 \end{bmatrix}$$

In addition to saving ink, vectrices are also easier to do linear algebra with, since they're (for all intents and purposes) matrices. Now, we can get into the math.

3.2   The Math: A Tale of 2 Frames

Here's a diagram of the 2 frames. $\hat{b}_1$ forms angles $\alpha_{11}, \alpha_{12}, \alpha_{13}$ with the 3 inertial basis vectors $\hat{n}_1, \hat{n}_2, \hat{n}_3$. This applies for the other body vectors as well, but that's just not shown on the diagram.
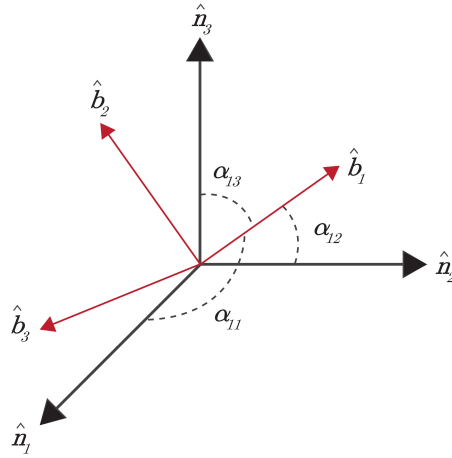
A good way to relate a body frame vectors to an inertial frame vectors is via the following:

$$\hat{b}_1 = \cos(\alpha_{11})\hat{n}_1 + \cos(\alpha_{12})\hat{n}_2 + \cos(\alpha_{13})\hat{n}_3$$
$$\hat{b}_2 = \cos(\alpha_{21})\hat{n}_1 + \cos(\alpha_{22})\hat{n}_2 + \cos(\alpha_{23})\hat{n}_3$$
$$\hat{b}_3 = \cos(\alpha_{31})\hat{n}_1 + \cos(\alpha_{32})\hat{n}_2 + \cos(\alpha_{33})\hat{n}_3$$

8

The cosines find the projection of each $\hat{b}$ against each $\hat{n}$. Being masters of linear algebra, we can immediately organize this whole thing into the following, to convert from inertial to body frame:

$$\{\hat{b}\} = \begin{bmatrix} \cos(\alpha_{11}) & \cos(\alpha_{12}) & \cos(\alpha_{13}) \\ \cos(\alpha_{21}) & \cos(\alpha_{22}) & \cos(\alpha_{23}) \\ \cos(\alpha_{31}) & \cos(\alpha_{32}) & \cos(\alpha_{33}) \end{bmatrix} \{\hat{n}\} = [BN]\{\hat{n}\}$$

And from the definition of a cosine, we can also say:

$$\{\hat{b}\} = \begin{bmatrix} b_1 \cdot n_1 & b_1 \cdot n_2 & b_1 \cdot n_3 \\ b_2 \cdot n_1 & b_2 \cdot n_2 & b_2 \cdot n_3 \\ b_3 \cdot n_1 & b_3 \cdot n_2 & b_3 \cdot n_3 \end{bmatrix} \{\hat{n}\} = [BN]\{\hat{n}\}$$

We'll call this matrix [BN]. The BN matrix we have here is the *Direction Cosine Matrix*. It doesn't take a rocket scientiest to see that the "Direction Cosine" part of the name comes from all these cosines.

3.2.1 Well, Whats The Point?

Glad you asked. Here's a situation: You are on Earth and your friend is on a spaceship in orbit. You just saw a really cool star and want to radio up to your friend where to look, with respect to their frame. You know the DCM that converts from your frame to theirs. How would you achieve this?

If it wasn't obvious, it's via that DCM you have. Lets say that v is the vector to the star. We can describe it in both the inertial frame and body frame, like so:

$$v = v_{b1}\hat{b_1} + v_{b2}\hat{b_2} + v_{b3}\hat{b_3} = [v_b]^T\{\hat{b}\} \ (1)$$

$$v = v_{b1}\hat{n_1} + v_{n2}\hat{n_2} + v_{n3}\hat{n_3} = [v_n]^T\{\hat{n}\} \ (2)$$

The $[v_b]$ and $[v_n]$ just indicate the coefficients that you multiply your basis vectors by. That T in the right side of the equation just indicates a transpose. All in all, real standard coordinate frame stuff.

Well, we have that {b} vectrix in the equation 1 and we know that $\{\hat{b}\} = [BN]\{\hat{n}\}$. So lets get to substituting.

$$[v_b]^T\{\hat{b}\} = [v_n]^T\{\hat{n}\}$$

$$[v_b]^T[BN]\{\hat{n}\} = [v_n]^T\{\hat{n}\}$$

Doing some funky transposing and linear algebra things, we'll discover that:

$$\boldsymbol{v_b = [BN]v_n}$$

and likewise,

$$\boldsymbol{v_n = [BN]^T v_b}$$

As you can see, DCMs not only *help describe how one frame is rotated relative to another*, but it can also help *translate what one vector looks like to an observer in a different frame.*

3.3   Adding and Subtracting DCMs

Despite all we've been doing so far, we are allowed to have more than 2 frames. For example, let's say your UFO is floating outside the ISS. You know the DCM that converts the ISS frame to your UFO frame. You also know the DCM that converts some inertial frame to ISS frame. What if you want to know how to convert from inertial straight to UFO frame?

Lets represent the UFO frame as U, the ISS frame as I, and the inertial frame to N. We want to find the DCM [UN], but we only know [UI] and [IN]. The solution is super simple:

$$\boldsymbol{[UN] = [UI][IN]}$$

This is what we call *DCM Addition.* But what about *DCM Subtraction*?

Let's say we already know [UN] and we know [IN]. How would we find the UFO frame relative to the ISS frame [UI]? Following our addition we can say:

$$[UI] = [UN][NI]$$

But wait a second, we don't know [NI]? We actually do. Transposing an orthogonal matrix gives the inverse.

$$[IN]^T = [NI]$$

Thus we can say:

$$[\boldsymbol{UI}] = [\boldsymbol{UN}][\boldsymbol{IN}]^{\boldsymbol{T}}$$

Doing this to [UN] subtracts out the part where I is related to N.

3.4 Properties

Well, what makes a DCM a DCM?

1. **DCMs are 3x3**

2. **The magnitude of each row and column is 1.**

   Ensuring the rows and columns are unit length makes the math less chaotic. Let's say that some malevolent DCM doesn't maintain unit length. When you multiply some vector through this DCM, the vector that comes out on the other side isn't the same length as the original. And now, not only do you have to keep track of a ton of numbers, you have to scale that vector's components up or down. Since we want to make our lives as easy as possible, this ain't the move.

3. **Every DCM element is $\leq |1|$**

   A consequence of the above. Not exactly hard to see why.

4. **DCMs are orthogonal**

   Using orthogonal matrices is convention because they're mathematically friendly. The fact that a DCM is orthogonal is the reason we can use the transpose to transform backwards from body to inertial. The inverse of an orthogonal matrix is just its transpose after all.

5. **The determinant of a DCM is** 1 **(Unit, Orthogonal, and Right Handed)**

   This property is essentially a result of the previous properties piled together. A unit orthogonal matrix has a determinant of ± 1. Why ± 1 you ask? Well the sign of the determinant indicates the handedness of the DCM. A +1 indicates a right handed frame and -1 is a left handed frame. A left handed frame technicallyyyyy isn't wrong, but right handed frames are convention. So for the sake of consistency, let's just say DCMs are right handed.

6. **DCMs are non-singular!**

   There's no divisions by zero, no weird functions, and since we're working at the lowest level, not many ambiguities can arise.

3.4.1 Wait a minute...

If DCMs are non-singular, why don't we exclusively use them to describe attitude? While DCMs are great and non-singular, they're not always the easiest/intuitive paramaterization to use.

The sheer amount of elements in the DCM is not very convenient. 9 elements to keep track of attitude? With properties that we constantly have to enforce? Tacking on angular velocity ω and multiple frames, the amount of numbers flying around can quickly get out of hand.

DCMs aren't super easy to visualize. If I hand you a list of 9 numbers, would you be able to immediately tell how the body frame is rotated? Probably not.

However, the DCM is super useful for translating between attitude parameterizations.

3.5 Test Your Knowledge

1. Identify the illegal DCMs:

$$a. \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad b. \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad c. \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.866 & -0.5 \\ 0 & 0.5 & 0.866 \end{bmatrix} \quad d. \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. Given DCMs [KB], [KJ], [NJ], how would you find DCM [BN]?

Chapter 4

Euler Angles

If you've heard of yaw pitch or roll, you've most certainly heard of Euler Angles.

EULER ANGLES: A parameterization that describes attitude in 3 sequential rotations.

The definition is pretty self explanatory. To describe an attitude in Euler Angles, you need to specify your type of euler angle, often referred to as a *set*. Then, you specify the degree value of the angles that belong to this set. Each of the angles describe how far you rotate the frame around a specified axis.

Here's an example if it ain't too clear:
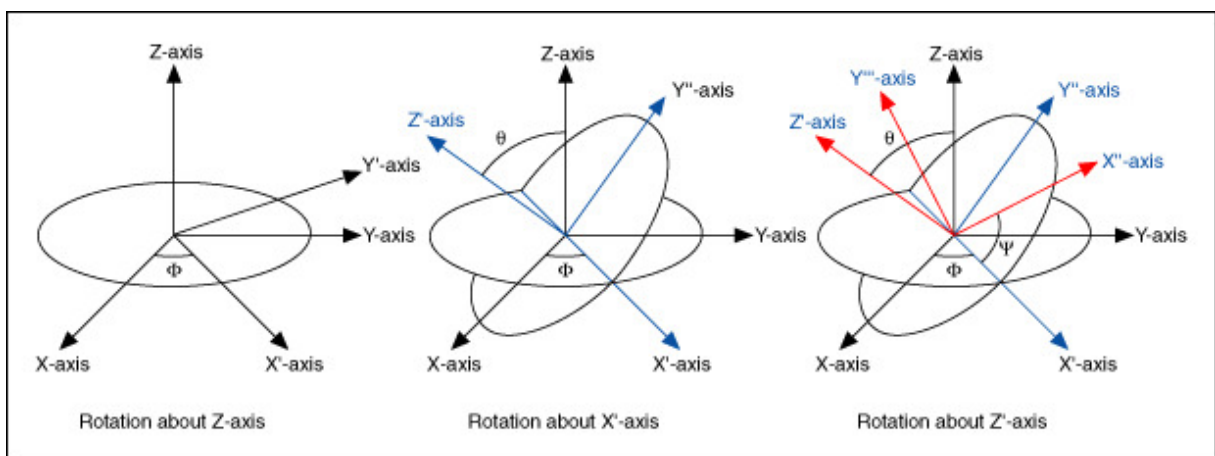
This is what's described as a 3-1-3 Euler Angle, since the rotations operate on the Z, X, and Z body axes. The rotation sequence does the following:

1. Rotate the frame around the Z-axis by angle ϕ. This forms a new frame with X' and Y' axes.

2. Rotate the frame around the X' axes by angle θ. This forms the Z' and Y'' axes.

3. Finally, rotate the frame around the Z' axis by angle ψ. This forms the X'' and Y''' axes, and the rotation is finished.

Of course, this is not the only set out there.

4.1 Types of Euler Angles

Euler angles fall into two categories.

1. awdawd

4.2 Euler Angles and DCMs

Chapter 5

Principal Rotation Vectors

Euler Angles are great, but you *really* don't wanna use them for spacecraft or other things that can tumble a lot.

5.1 Euler's Principal Rotation Theorem

Also known as *Euler's Rotation Theorem*, it states that:

THEOREM: A rigid body can be brought from a start to an end orientation by a single rigid rotation (ϕ rad/deg) around some principal axis ($\hat{e}$). This axis is fixed in both the start and end orientation.

Chapter 6

The Quaternion

Googling "quaternion math" will probably scare you, unless you're a math major. Taken in the context of PRVs however, they're actually pretty straightforward. They're probably one of the most popular non-singular attitude descriptors out there.

Just to get this out of the way, you'll probably hear the term "Euler Parameters" being used (EPs for short) and you might wonder what the difference is. Quaternions are the mathematical construct and Euler Parameters are the *coefficients* of the quaternion.

6.0.1   Well, what is a quaternion?

Quaternions form a number system that extends the complex number system. It was invented by William Rowan Hamilton back in 1843, after having an epiphany on a bridge near his home in Ireland. They're usually represented with 4 terms:

$$a + bi + cj + dk$$

Where a,b,c,d are the real coefficients and i,j,k well... aren't. I'm not gonna go too in the weeds with this since I'm not qualified enough to say more.

Chapter 7

Attitude Determination

Now that you got a good chunk of the preliminaries out of the way, let's get into how it's applied. The whole idea of *Attitude Determination* is broken up into 2 types of problems:

1. **Static Attitude Determination:** You take all of your measurements at the same time, and try to solve for the optimal geometry of the measurements, figuring out your attitude at that instant of time.

2. **Dynamic Attitude Determination:** You take your measurements over time and try to figure out how your body is spinning around. This is a far harder problem and one that this text will not cover.

Chapter 8

TRIAD Method

Chapter 9

Wahba's Problem

Chapter 10

Davenport's Q

Chapter 11
QUEST

Chapter 12

Answer Key

12.1   Chapter 1 DCM

1. a & b

2. $[KB]^T[KJ][NJ]^T$