

Axis Transformation

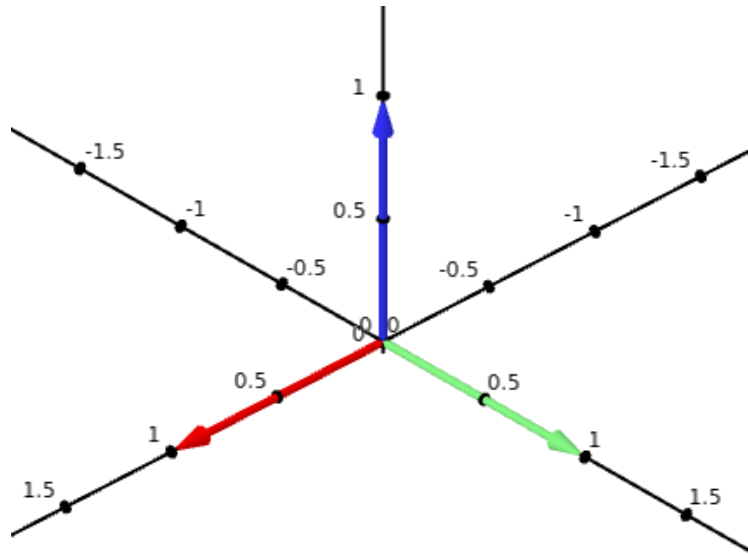
for the NOVA flight computer data generation suite

Paul Conway

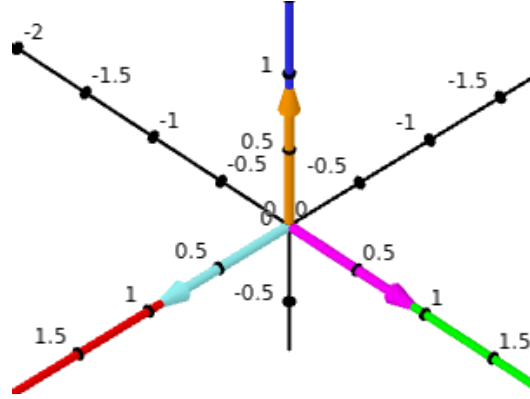
October 26, 2021

Introduction

One of the challenges faced by the NOVA Ground Station is the fact that the flight computer is reporting data from its frame of reference. We will refer to this reference frame as the Vehicle Centered Non-inertial frame, or VCN frame. To make the data more useful, it is important to convert the data from the VCN frame to the frame of reference of the ground station. We will refer to this reference frame as the Ground Station Inertial frame, or GSI frame. Let us examine these two reference frames more closely.

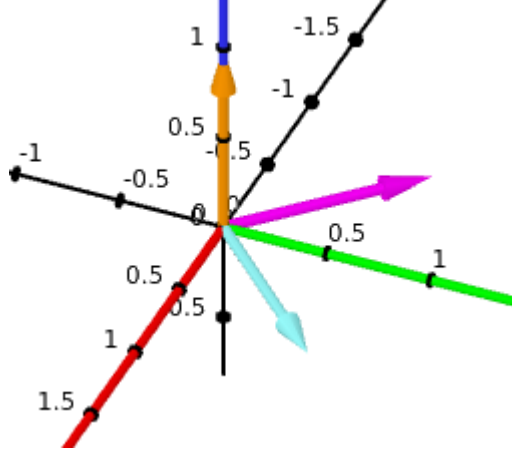


In the image above, you can see three vectors. The red, green, and blue vectors are the \hat{i} , \hat{j} , and \hat{k} unit vectors of the GSI frame. We will color the axes corresponding to each unit vector accordingly. Before the vehicle is launched, it rests on the ground such that the VCN frame aligns with the GSI frame.



In the image above, the \hat{i}_V , \hat{j}_V , and \hat{k}_V vectors have been colored cyan, magenta, and orange respectively. Clearly in this state, transitioning from the VCN to the GSI frame is trivial as there is no change to the vectors. Obviously though, this will change. To understand how to make that transition, we have to understand the changes taking place between the two frames. There are in fact two types of changes that can occur between these reference frames: translation and rotation. However, we are only concerned with rotation. *Translational changes do not affect the transition from one frame to another.* This is important to understand as we will treat both frames as sharing the same origin. In reality of course the vehicle is somewhere other than the ground station.

Rotation Matrices



In the image above, we can see that the VCN frame has been rotated with respect to the GSI frame. Let us define these rotations more rigorously.

Let's begin with what an extrinsic rotation is. An extrinsic rotation is one about a fixed set of axes. In our case, our fixed axes are those defined by GSI frame. The rotation depicted above is an extrinsic rotation. The other type of rotation is an intrinsic rotation, which is one about the rotating axes. If we were to rotate the VCN frame about itself, that would be an intrinsic rotation. Next we will properly define each rotation.

- α represents the rotation about the \hat{i} vector
- β represents the rotation about the \hat{j} vector
- γ represents the rotation about the \hat{k} vector

Together these angles are referred to as Euler angles, and represent the pitch, yaw, and roll of our system respectively. Next we will discuss rotation matrices. We know we can express a vector as a matrix. For now, accept that this vector must be represented as a column vector. For example, a vector $\vec{a} = a_x\hat{i} + a_y\hat{j} + a_z\hat{k}$ can be represented as:

$$\mathbf{A} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}$$

We can of course rotate this vector. Without getting too much into the linear algebra, for now just understand that we can accomplish this rotation by multiplying our vector by a rotation matrix. For principal axes xyz , these matrices are:

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

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

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

In our case, $x = \hat{i}$, $y = \hat{j}$, $z = \hat{k}$. These matrices represent a rotation of angle θ about their respective axis. For example, to rotate our vector \mathbf{A} an angle ϕ about the \hat{j} axis to obtain a new vector \mathbf{B} , we would do the following:

$$\mathbf{B} = R_y(\phi)\mathbf{A} = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}$$

The beauty of this system is that it allows us to easily chain multiple rotations together. For example, if we wanted to obtain matrix \mathbf{B} by rotating \mathbf{A} through angles a , b , and c around the \hat{i} , \hat{j} , and \hat{k} vectors (in that order), we would carry out the following calculation:

$$\mathbf{B} = R_x(a)R_y(b)R_z(c)\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\sin a \\ 0 & \sin a & \cos a \end{bmatrix} \begin{bmatrix} \cos b & 0 & \sin b \\ 0 & 1 & 0 \\ -\sin b & 0 & \cos b \end{bmatrix} \begin{bmatrix} \cos c & -\sin c & 0 \\ \sin c & \cos c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}$$

We could also decide on a different order of rotations. For example, we could rotate \mathbf{A} around the \hat{j} , \hat{k} , then \hat{i} and our calculation would look like:

$$\mathbf{B} = R_y(b)R_z(c)R_x(a)\mathbf{A}$$

We can also just as easily reverse this rotation. If we have \mathbf{B} and want to return to \mathbf{A} from our \hat{j} then \hat{k} then \hat{i} rotation, we would carry out the following calculation:

$$\mathbf{A} = R_x^{-1}(b)R_z^{-1}(c)R_y^{-1}(a)\mathbf{B}$$

You should notice that the order of these applied rotations changed when going in reverse. That is what the next section will discuss.

Dealing with Noncommutivity

I want you to try something. Pick up your phone or another object that has three distinct axes. Hold it out in front of you and rotate it forward, then clockwise. Remember this orientation. Now reorient your phone as it was before you performed those rotations. This time, rotate it clockwise, then forward. You will notice that the orientation is different from the first set of rotations. Why is this? The basic and most concise answer is that *rotations are not commutative*. That is, in general the order in which you perform a set of rotations matters. This fact by the way is why angles can never be expressed as vectors. The commutative property is a fundamental property of elements in a vector space.

This noncommutivity has many implications for us. While we do have a generalized way to rotate a coordinate system with a set of three angles, and reverse those rotations, we have to know the order with which these rotations were applied. We need a way to rotate a coordinate system such that the order of rotations does not matter.

TODO: May Remove This noncommutivity has many implications for us. While we do have a generalized way to rotate a coordinate system with a set of three angles, and reverse those rotations, we have to know the order with which these rotations were applied. In reality, the vehicle is rotating about all axes simultaneously. To better tackle this problem, we should look more closely at the vehicle itself. Let us recap some facts about the vehicle, and introduce some new ones:

- The VCN frame is fixed to the body of the vehicle.
- The \hat{k}_V vector always points in the direction of forward motion. For rockets, this is the direction that the nose points.
- The vehicle is constantly reporting

Glossary of Terms

1. VCN frame

- Vehicle Centered Non-inertial frame.
- This is the frame of reference of the vehicle itself.
- It is expressed with the unit vectors \hat{i}_V , \hat{j}_V , and \hat{k}_V such that $\hat{i}_V \times \hat{j}_V = \hat{k}_V$ completes a right-handed cartesian coordinate system.

2. GSI frame

- Ground Station Inertial frame.
- This is the frame of reference of the ground station.
- It is expressed with the unit vectors \hat{i} , \hat{j} , and \hat{k} such that $\hat{i} \times \hat{j} = \hat{k}$ completes a right-handed cartesian coordinate system.