Attitude Parametrizations and Kinematics for Use in Rigid Body Dynamics

Kevin Tracy[1-2 *]

[1]Lockheed Martin Space Systems, Sunnyvale, CA
[2]Stanford University Department of Mechanical Engineering, Stanford, CA

The orientation between two unitary orthogonal vector bases can be described using a Direction Cosine Matrix (DCM). If one unitary orthogonal vector basis is fixed to a rigid body, and the other is fixed to a reference frame, this DCM can be considered the attitude of the rigid body. This DCM can be parameterized using one of multiple rotation conventions (Euler angles, axis-angle, quaternion, etc). It is important for the attitude parameterization to be unambiguous in use, with special emphasis on the kinematics of this parameterization. This document will examine two sets of attitude conventions, one used by Thomas Kane, and the other used by James Wertz.

Attitude | Quaternion | Spacecraft | Guidance Navigation and Control | Direction Cosine Matrix | Rigid Body Dynamics | Euler Symmetric Parameters | Rotation Matrix | Axis-Angle | Rotation Vector

Correspondence: ***ktracy@stanford.edu***

Relevant Mathematical Conventions

Cross Product Matrix. The cross product between two vectors can be expressed as matrix multiplication by converting the left vector into a skew-symmetric matrix. This is beneficial because matrix multiplication is easier to manipulate than the cross product. This operation turns a vector into a skew-symmetric matrix. A three dimensional vector $\vec{a}$, with the cross product matrix operator applied, is as follows:

$$[\vec{a}]_\chi = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_\chi = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad \textbf{(1)}$$

This provides for:

$$\vec{a} \times \vec{b} = [\vec{a}]_\chi \vec{b} \quad \textbf{(2)}$$

The order of the cross product can be switched similar to standard cross product convention, and the cross product matrix operator holds.

$$\vec{a} \times \vec{b} = -\vec{b} \times \vec{a} = -[\vec{b}]_\chi \vec{a} \quad \textbf{(3)}$$

The transpose of this matrix is equal to its negative:

$$\left[[\vec{a}]_\chi\right]^T = -[\vec{a}]_\chi \quad \textbf{(4)}$$

The inverse operator describes the same operation, but reversed. The inverse cross product matrix operator converts a skew symmetric matrix into a vector.

$$\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}_{\chi^{-1}} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \vec{a} \quad \textbf{(5)}$$

Matrix Exponential. the matrix exponential will be used in this paper, as well as the matrix logarithm. Both of these apply to exclusively square matrices. The matrix exponential is defined as:

$$\mathrm{e}^{\mathbf{A}} = \sum_{n=0}^{\infty} \frac{\mathbf{A}^n}{n!} \quad \textbf{(6)}$$

It is important to know that when using MATLAB, exp(A) is simply taking the exponential of each element of the matrix. The correct function to use in MATLAB for matrix exponential is expm(A). In Python, it is scipy.linalg.expm(A). The inverse of the matrix exponential is the matrix logarithm, this is defined as:

$$\log(\mathbf{B}) = \sum_{n=1}^{\infty} (-1)^n \frac{(\mathbf{B} - \mathbf{I}_3)^n}{n} \quad \textbf{(7)}$$

Where $\mathbf{I}_3$ is a 3x3 identity matrix. Similarly to the matrix exponential, the MATLAB function log(B) is the element-wise logarithm, and not the matrix logarithm. The correct MATLAB function for the matrix logarithm is logm(B). In Python it is scipy.linalg.logm(B). It is important to note that sometimes the matrix logarithm as written above doesn't converge easily, and more sophisticated numerical methods must be used. Both the MATLAB and Python functions described take advantage of said numerical methods to ensure convergence.

Vector Notation. Vector notation varies based on the textbook/paper referenced. For this paper, the following vector notation will be used:

$${}^{\mathrm{A}}\vec{v}\,{}^{\mathrm{B}}_{\mathrm{C}} \quad \textbf{(8)}$$

- A – Reference frame $\vec{v}$ is with respect to
- B – Point of interest
- C – Basis the vector is expressed in components of

Where the above signifies B's velocity in A, as expressed in the C basis. Below is an example of an angular velocity vector dressed in the described notation:

${}^{\mathrm{A}}\vec{\omega}\,{}^{\mathrm{B}}_{\mathrm{C}}$ – Angular velocity of B in A, expressed in C

The additions to this convention are with position and moment vectors, as well as inertia dyadics. For these, there are two variables in the upper right hand superscript. The following is a position vector from B to A, expressed in basis C:

$$\vec{r}_C^{\,A/B} \quad \textbf{(9)}$$

A – Point of interest

B – Origin of position vector

C – Basis the vector is expressed in components of

And lastly, any vector with the horizontal arrow on top of it, like $\vec{v}$, is a vector of non-unitary norm. Any vector with a hat on top of it, like $\hat{v}$, is a vector of unitary norm.

Reference Frames. In this document, it will be assumed that each reference frame has one unitary orthogonal vector basis attached to it. This will allow the paper to use "frame" and "basis" interchangeably. Examples of possible reference frames are Earth Centered Inertial (ECI), Earth Centered Earth Fixed (ECEF), and the body fixed axes on a spacecraft. A Newtonian reference frame is one that is not rotating or accelerating, and $F = ma$ applies. A vector expressed in basis C is as follows:

$$\vec{v}_C = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_C = v_1\hat{c}_x + v_2\hat{c}_y + v_3\hat{c}_z \tag{10}$$

Unit vectors $\hat{c}_x$, $\hat{c}_y$, $\hat{c}_z$, are the vector components of the right handed orthogonal vector basis C. Scalars v_1, v_2, v_3 are the measure numbers that describe vector $\vec{v}$ in basis C.

Rigid Body Dynamics

A rigid body spinning will have an angular velocity vector. This angular velocity vector should be that of the body with respect to the inertial frame, and will be expressed in the body frame (${}^N\vec{\omega}_B^B$). This is because gyros are mounted on the spacecraft, and this is the vector that the sensor produces natively. This is also the angular velocity that is most conveniently used in Euler's 3D rigid body dynamics equation, and the gyrostat equation.

$${}^N\dot{\vec{\omega}}_B^B = \mathbf{J}^{-1}\big[\vec{M} - \vec{\omega} \times \mathbf{J}\vec{\omega}\big] \tag{11}$$

$\mathbf{J} = \mathbf{J}_B^{B/B_{cg}}$	Inertia dyadic
$\vec{M} = {}^N\vec{M}_B^{B/B_{cg}}$	External moment
$\vec{\omega} = {}^N\vec{\omega}_B^B$	Angular Velocity

This is usually the only angular velocity vector of interest when simulating rigid body dynamics. Care will be taken in this document to preserve this form of the angular velocity vector. Where translational velocity can be directly integrated to position, angular velocity can not be directly integrated to orientation without a little more work.

Direction Cosine Matrix

A direction cosine matrix, or DCM, defines orientation between two unitary orthogonal bases. It is used to transform a vector expressed in one basis, to the same vector expressed in another basis. The vector itself is not being rotated, a DCM simply expresses the vector in a different basis. If there are two bases, N and B, a DCM relating the two will look like the following:

$${}^N\mathbf{R}^B = \begin{bmatrix} \hat{n}_x\cdot\hat{b}_x & \hat{n}_x\cdot\hat{b}_y & \hat{n}_x\cdot\hat{b}_z \\ \hat{n}_y\cdot\hat{b}_x & \hat{n}_y\cdot\hat{b}_y & \hat{n}_y\cdot\hat{b}_z \\ \hat{n}_z\cdot\hat{b}_x & \hat{n}_z\cdot\hat{b}_y & \hat{n}_z\cdot\hat{b}_z \end{bmatrix} \tag{12}$$

The DCM gets its name from the fact that the dot product between two unit vectors is simply the cosine of the angle between them. Because of this, the DCM is a matrix that stores the cosines between all combinations of the three unit vectors on each basis. The way to convert a vector expressed in the basis B ($\vec{v}_B$) to the same vector expressed in basis N ($\vec{v}_N$) would be as follows:

$$\vec{v}_N = \left[{}^N\mathbf{R}^B\right]\vec{v}_B \tag{13}$$

The vector (it's magnitude and direction), are not changing. The only thing a DCM changes about a vector is the basis it is expressed in. This is a critical concept to understand, because DCM's when used in the manner above, do not rotate the vector. Another equivalent, albeit more computationally expensive, expression for expressing a vector in a different basis is below.

$$\vec{v}_N = \Big[\left[{}^N\mathbf{R}^B\right]\left[\vec{v}_B\right]_\chi\left[{}^N\mathbf{R}^B\right]^T\Big]_{\chi^{-1}} \tag{14}$$

Eq. (14) is *unnecessary* when using DCM's, but gives good insight into creating an equivalent operation using a quaternion.

DCM's are orthogonal, which means that the transpose is equal to its inverse. This is convenient for inverting DCM's.

$$\left[{}^B\mathbf{R}^N\right] = \left[{}^N\mathbf{R}^B\right]^{-1} = \left[{}^N\mathbf{R}^B\right]^T \tag{15}$$

This inverted DCM can now be used to take the vector expressed in basis N, and express it in basis B.

$$\vec{v}_B = \left[{}^B\mathbf{R}^N\right]\vec{v}_N \tag{16}$$

DCM's can be multiplied together in the following manner:

$${}^A\mathbf{R}^D = {}^A\mathbf{R}^B\,{}^B\mathbf{R}^C\,{}^C\mathbf{R}^D \tag{17}$$

Linking up the bases and multiplying DCM's will be considered "chaining" in this document.

DCM Kinematics. The kinematic equation for the orientation DCM given an angular velocity between the two reference frames is as follows:

$${}^N\dot{\mathbf{R}}^B = {}^N\mathbf{R}^B\left[{}^N\omega_B^B\right]_\chi \tag{18}$$

If the orientation is parameterized using ${}^B\mathbf{R}^N$, the expression is simply transposed resulting in:

$${}^B\dot{\mathbf{R}}^N = -\left[{}^N\omega_B^B\right]_\chi\,{}^B\mathbf{R}^N \tag{19}$$

The proof for this equation is presented two different ways in the proofs section of this document. There have been no assumptions about bases N and B. One can be inertial, one can be body fixed, they could both be fixed to rigid bodies, these are all possibilities that are governed by the above equations. The only thing that dictates the kinematics of the DCM is the angular velocity between the two reference frames.

Re-orthogonalizing a DCM. While propagating the kinematics of the DCM, the DCM will begin to drift away from orthogonality. Re-orthogonalizing a matrix is computationally expensive, but can be done. The re-orthogonalization requires a Singular Value Decomposition (SVD) function. This SVD function takes an input matrix $\mathbf{R}^*$, and returns a $\mathbf{U}$, Σ, and $\mathbf{V}$, such that $\mathbf{R}^* = \mathbf{U}\Sigma\mathbf{V}^T$.

- Input: $\mathbf{R}^*$ (not orthogonal)
- $[\mathbf{U}, \Sigma, \mathbf{V}] = SVD(\mathbf{R}^*)$
- output: $\mathbf{R} = \mathbf{U}\mathbf{V}^T$ (orthogonalized)

DCM as a Rotation

Another way to describe the orientation between two bases, is with a rotation going from one to the other. This is the cause for a fork between the Kane/Levinson and Wertz/Markley conventions. Kane/Levinson parameterize ${}^N\mathbf{R}^B$ as a rotation from N to B, whereas Wertz/Markley parameterize it as a rotation from B to N. This is more than a notation difference, and changes everything from this point forward in this document. The following sections will be in the Kane/Levinson convention, and will say Kane/Levinson in the title of each section. After a formula sheet condensing all of the important functions for the Kane/Levinson convention, the Wertz/-Markley convention will be examined.

Axis-Angle (Kane/Levinson)

Euler's theorem states that the rotation between any two bases can always be described as a simple rotation about an axis. This means that another way to parameterize orientation is with this axis of rotation, and angle to rotate about. This axis-angle vector, $\vec{\phi}$, is the unit vector axis of rotation, $\hat{r}$, multiplied by the angle rotated, θ.

$$\vec{\phi} = \hat{r}\theta \tag{20}$$

$$\hat{r} - \text{Axis of rotation (unit norm)}$$
$$\theta - \text{rotation angle (radians)} \in [-\pi, \pi]$$

Given an axis-angle vector $\vec{\phi}$, the corresponding $\hat{r}$ and θ can be determined.

$$\hat{r} = \frac{\vec{\phi}}{||\vec{\phi}||} \tag{21}$$

$$\theta = ||\vec{\phi}|| \tag{22}$$

The DCM can be reconstructed from this axis-angle vector with any of the following three expressions:

$$ {}^N\vec{\phi}^B = \hat{r}\theta \qquad s = \sin(\theta)$$
$$\hat{r} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}^T \qquad c = \cos(\theta)$$

$${}^N\mathbf{R}^B = \begin{bmatrix} c + (1-c)r_1^2 & (1-c)r_2r_1 - sr_3 & (1-c)r_3r_1 + sr_2 \\ (1-c)r_1r_2 + sr_3 & c + (1-c)r_2^2 & (1-c)r_3r_2 - sr_1 \\ (1-c)r_1r_3 - sr_2 & (1-c)r_2r_3 + sr_1 & c + (1-c)r_3^2 \end{bmatrix}$$

$${}^N\mathbf{R}^B = \mathbf{I}_3 + \sin(\theta)[\hat{r}]_\chi + (1 - \cos(\theta))[\hat{r}]_\chi[\hat{r}]_\chi \tag{23}$$

$${}^N\mathbf{R}^B = \mathrm{e}^{\left[{}^N\vec{\phi}^B\right]_\chi} \tag{24}$$

If θ is very small, the following approximation can be used:

$${}^N\mathbf{R}^B \approx \mathbf{I}_3 + \left[{}^N\vec{\phi}^B\right]_\chi \tag{25}$$

Eq. (24) can be inverted and the axis-angle vector can be created from the DCM:

$${}^N\vec{\phi}^B = \left[log({}^N\mathbf{R}^B)\right]_{\chi^{-1}} \tag{26}$$

$\hat{r}$ is fixed in both reference frames, this means it is the same expressed in both the N and B frames, like how a door hinge is fixed in both the door frame and the door. This is because the resulting DCM ${}^N\mathbf{R}^B$ has one real eigenvalue ($\lambda = 1$), and the corresponding eigenvector is this unit vector $\hat{r}$. For a matrix ($\mathbf{A}$), the eigenvalue (λ_i), and the corresponding eigenvector($\vec{\text{v}}_i$), satisfy the following equation:

$$\mathbf{A}\vec{\text{v}}_i = \lambda_i\vec{\text{v}}_i$$

This means in the case of a DCM where $\lambda_i = 1$ and $\hat{r} = \vec{\text{v}}_i$:

$${}^N\mathbf{R}^B\hat{r} = 1 \cdot \hat{r}$$

This means that $\hat{r}$ is unchanged by the DCM, and fixed in both reference frames.

$$\hat{r}_B = {}^N\mathbf{R}^B\ \hat{r}_B \tag{27}$$

$$\hat{r}_B = {}^B\mathbf{R}^N\ \hat{r}_B \tag{28}$$

$$\hat{r}_B = \hat{r}_N \tag{29}$$

Quaternion (Kane/Levinson)

Quaternions are numbers with 1 real part, and 3 imaginary parts.

$$\mathbf{q} = q_1 i + q_2 j + q_3 k + q_4 \tag{30}$$

Attitude can be parameterized using quaternions with unit norm. This quaternion can be expressed as a 4x1 vector with a vector part (coefficients for the imaginary terms), followed by a scalar part (real term).

$$\mathbf{q} = \begin{bmatrix} \vec{q}_{1:3} \\ q_4 \end{bmatrix} = \begin{bmatrix} \text{vector part} \\ \text{scalar part} \end{bmatrix} = \begin{bmatrix} \vec{\text{v}} \\ \text{s} \end{bmatrix} \tag{31}$$

The ordering of the vector and scalar parts change between resources, but scalar last has been chosen for this document. This is just a bookkeeping decision, and doesn't fundamentally change the orientation being described, or any of the kinematics.

The quaternion is constructed using the axis-angle (rotation vector).

$$\mathbf{q} = \begin{bmatrix} \hat{r}\,\sin(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) \end{bmatrix} \tag{32}$$

This quaternion will always have unit norm if constructed from an axis-angle vector.

$$||\mathbf{q}|| = 1 \tag{33}$$

This equation can be rearranged to solve for the axis and angle from a quaternion, but it is not the most numerically accurate method. A more robust method of determining the axis and angle from a quaternion is as follows:

$$\hat{r} = \frac{\vec{q}_{1:3}}{||\vec{q}_{1:3}||} \tag{34}$$

$$\theta = 2\,\mathrm{atan2}(||\vec{q}_{1:3}||, q_4) \tag{35}$$

If $\theta > \pi$, change such that $\theta^{'} = \theta - 2\pi$. The identity quaternion corresponds to a axis-angle rotation with $\theta = 0$

$$\mathbf{q}_{\text{Identity}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \vec{0} \\ 1 \end{bmatrix} \tag{36}$$

A quaternion can be converted into a DCM using any of the following 3 expressions:

$${}^{N}\mathbf{q}^{B} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^{T}$$

$$\vec{q}_{1:3} = \vec{\mathrm{v}}$$
$$q_4 = \mathrm{s}$$

$${}^{N}\mathbf{R}^{B} = \begin{bmatrix} 2q_1^2+2q_4^2-1 & 2(q_1q_2-q_3q_4) & 2(q_1q_3+q_2q_4) \\ 2(q_1q_2+q_3q_4) & 2q_2^2+2q_4^2-1 & 2(q_2q_3-q_1q_4) \\ 2(q_1q_3-q_2q_4) & 2(q_2q_3+q_1q_4) & 2q_3^2+2q_4^2-1 \end{bmatrix} \tag{37}$$

$${}^{N}\mathbf{R}^{B} = (\mathrm{s}^2 - ||\vec{\mathrm{v}}||^2)\mathbf{I}_3 + 2\mathrm{s}[\vec{\mathrm{v}}]_{\chi} + 2\vec{\mathrm{v}}\vec{\mathrm{v}}^{T} \tag{38}$$

$${}^{N}\mathbf{R}^{B} = \mathbf{I}_3 + 2\big[\vec{\mathrm{v}}\big]_{\chi}\big[\mathrm{s}\,\mathbf{I}_3 + [\vec{\mathrm{v}}]_{\chi}\big] \tag{39}$$

Eq. (39) is difficult to invert, with some computational shortcomings, so the following method is used for converting from a DCM to a quaternion:

```
function [q] = q_from_DCM(R)
% Scalar Last, Kane/Levinson Convention
T = R(1,1) + R(2,2) + R(3,3)
if T> R(1,1) && T> R(2,2) && T>R(3,3)
    q4 = .5*sqrt(1+T);
    r  = .25/q4;
    q1 = (R(3,2) - R(2,3))*r
    q2 = (R(1,3) - R(3,1))*r
    q3 = (R(2,1) - R(1,2))*r
elseif R(1,1)>R(2,2) && R(1,1)>R(3,3)
    q1 = .5*sqrt(1-T + 2*R(1,1))
    r  = .25/q1
    q4 = (R(3,2) - R(2,3))*r
    q2 = (R(1,2) + R(2,1))*r
    q3 = (R(1,3) + R(3,1))*r
elseif R(2,2)>R(3,3)
    q2 = .5*sqrt(1-T + 2*R(2,2))
    r  = .25/q2
    q4 = (R(1,3) - R(3,1))*r
    q1 = (R(1,2) + R(2,1))*r
    q3 = (R(2,3) + R(3,2))*r
else
    q3 = .5*sqrt(1-T + 2*R(3,3))
    r  = .25/q3
    q4 = (R(2,1) - R(1,2))*r
    q1 = (R(1,3) + R(3,1))*r
    q2 = (R(2,3) + R(3,2))*r
end
q = [q1;q2;q3;q4]
end
```

Quaternions can be multiplied using the Hamilton Product:

$$\mathbf{q}_1 = \begin{bmatrix} \vec{\mathrm{v}}_1 \\ \mathrm{s}_1 \end{bmatrix} \qquad \mathbf{q}_2 = \begin{bmatrix} \vec{\mathrm{v}}_2 \\ \mathrm{s}_2 \end{bmatrix}$$

$$\mathbf{q}_1 \odot \mathbf{q}_2 = \begin{bmatrix} \mathrm{s}_1\vec{\mathrm{v}}_2 + \mathrm{s}_2\vec{\mathrm{v}}_1 + \vec{\mathrm{v}}_1 \times \vec{\mathrm{v}}_2 \\ \mathrm{s}_1\mathrm{s}_2 - \vec{\mathrm{v}}_1 \cdot \vec{\mathrm{v}}_2 \end{bmatrix} \tag{40}$$

This can be written two equivalent ways, separating out the first or second quaternion on the right:

$$\mathbf{q}_1 \odot \mathbf{q}_2 = \left[\begin{array}{c:c} \mathrm{s}_2\mathbf{I}_3 - [\vec{\mathrm{v}}_2]_{\chi} & \vec{\mathrm{v}}_2 \\ \hdashline -\vec{\mathrm{v}}_2^{T} & \mathrm{s}_2 \end{array}\right] \left[\begin{array}{c} \vec{\mathrm{v}}_1 \\ \hdashline \mathrm{s}_1 \end{array}\right] \tag{41}$$

$$\mathbf{q}_1 \odot \mathbf{q}_2 = \left[\begin{array}{c:c} \mathrm{s}_1\mathbf{I}_3 + [\vec{\mathrm{v}}_1]_{\chi} & \vec{\mathrm{v}}_1 \\ \hdashline -\vec{\mathrm{v}}_1^{T} & \mathrm{s}_1 \end{array}\right] \left[\begin{array}{c} \vec{\mathrm{v}}_2 \\ \hdashline \mathrm{s}_2 \end{array}\right] \tag{42}$$

If multiplication is done using this method, quaternions have all the same chaining properties that rotation matrices have.

$${}^{A}\mathbf{R}^{C} = {}^{A}\mathbf{R}^{B}\,{}^{B}\mathbf{R}^{C} \qquad {}^{A}\mathbf{q}^{C} = {}^{A}\mathbf{q}^{B} \odot {}^{B}\mathbf{q}^{C} \tag{43}$$

A quaternion can be used transform a vector expressed in one reference frame, to another. Similar to how equation Eq. (14) transforms the vector, the following holds for quaternions:

$$\begin{bmatrix} \vec{v}_N \\ 0 \end{bmatrix} = {}^{N}\mathbf{q}^{B} \odot \begin{bmatrix} \vec{v}_B \\ 0 \end{bmatrix} \odot \big[{}^{N}\mathbf{q}^{B}\big]^{\dagger} \tag{44}$$

The inverse, or conjugate, of a quaternion is given by flipping the sign of the vector component.

$$ {}^N\mathbf{q}^B = [{}^B\mathbf{q}^N]^\dagger \tag{45} $$

$$ \mathbf{q} = \begin{bmatrix} \vec{v} \\ s \end{bmatrix} \qquad \mathbf{q}^\dagger = \begin{bmatrix} -\vec{v} \\ s \end{bmatrix} \tag{46} $$

It is also important to realize that the negative of a quaternion describes the same orientation. This is because quaternions double cover SO(3) (group of 3 dimensional orientations) :

$$ [\mathbf{q}] \qquad \text{same orientation as} \qquad [-\mathbf{q}] $$

The positive and negative quaternions describe the same orientation, but one is a rotation with $|\theta| > \pi$ and one is a rotation with $|\theta| < \pi$. This is because a rotation between two frames can have a positive or negative θ, but one will require a larger rotation than the other. It is important in control design work that the quaternion with the positive scalar is chosen. A positive scalar indicates that $|\theta| < \pi$. This is critical because a negative scalar quaternion could mean a 359° rotation, and the positive scalar quaternion a 1° rotation.

Quaternion Kinematics (Kane/Levinson)

Orientation kinematics can be propagated in the DCM, axis-angle vector, or quaternion. The kinematics of the axis-angle vector are computationally inefficient, require trigonometric functions, and have a singularity at $\theta = \pm\pi$, so this method will be avoided. The kinematics of the DCM is expressed earlier in this document, but it drifts out of orthogonality, and it's 9 individual ODE's. Both of these present computational inefficiencies. The kinematics of the quaternion avoid all of these shortcomings: The kinematics are free of trigonometric functions, singularities, and are easily re-normalized at every time step to keep the quaternion of unitary norm. This is why the quaternion is the ideal orientation parameterization for numerical simulation. The kinematics for the quaternion are derived in the proofs section of this document.

$$ {}^N\dot{\mathbf{q}}^B = \frac{1}{2}{}^N\mathbf{q}^B \odot \begin{bmatrix} {}^N\vec{\omega}_B^B \\ 0 \end{bmatrix} \tag{47} $$

This quaternion kinematic ODE shows up in various forms. Below are two more equivalent forms of this equation.

$$ {}^N\mathbf{q}^B = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T $$
$$ {}^N\vec{\omega}_B^B = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T $$

$$ {}^N\dot{\mathbf{q}}^B = \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{48} $$

$$ {}^N\dot{\mathbf{q}}^B = \frac{1}{2}\begin{bmatrix} q_4\mathbf{I}_3 + [\vec{q}_{1:3}]_\times \\ -\vec{q}_{1:3}^T \end{bmatrix} {}^N\vec{\omega}_B^B \tag{49} $$

$$ {}^N\dot{\mathbf{q}}^B = \frac{1}{2}\begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{50} $$

$$ {}^N\dot{\mathbf{q}}^B = \frac{1}{2}\begin{bmatrix} -[{}^N\vec{\omega}_B^B]_\times & {}^N\vec{\omega}_B^B \\ -[{}^N\vec{\omega}_B^B]^T & 0 \end{bmatrix} {}^N\mathbf{q}^B \tag{51} $$

If the attitude is described as ${}^B\mathbf{q}^N$ instead of ${}^N\mathbf{q}^B$, the quaternion kinematics are as such:

$$ {}^B\dot{\mathbf{q}}^N = -\frac{1}{2}\begin{bmatrix} {}^N\vec{\omega}_B^B \\ 0 \end{bmatrix} \odot {}^B\mathbf{q}^N \tag{52} $$

With the corresponding two more equivalent equations:

$$ {}^B\mathbf{q}^N = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T $$
$$ {}^N\vec{\omega}_B^B = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T $$

$$ {}^B\dot{\mathbf{q}}^N = \frac{1}{2}\begin{bmatrix} -q_4 & -q_3 & q_2 \\ q_3 & -q_4 & -q_1 \\ -q_2 & q_1 & -q_4 \\ q_1 & q_2 & q_3 \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{53} $$

$$ {}^B\dot{\mathbf{q}}^N = \frac{1}{2}\begin{bmatrix} -q_4\mathbf{I}_3 + [\vec{q}_{1:3}]_\times \\ \vec{q}_{1:3}^T \end{bmatrix} {}^N\vec{\omega}_B^B \tag{54} $$

$$ {}^B\dot{\mathbf{q}}^N = \frac{1}{2}\begin{bmatrix} 0 & \omega_z & -\omega_y & -\omega_x \\ -\omega_z & 0 & \omega_x & -\omega_y \\ \omega_y & -\omega_x & 0 & -\omega_z \\ \omega_x & \omega_y & \omega_z & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{55} $$

$$ {}^B\dot{\mathbf{q}}^N = \frac{1}{2}\begin{bmatrix} -[{}^N\vec{\omega}_B^B]_\times & -{}^N\vec{\omega}_B^B \\ [{}^N\vec{\omega}_B^B]^T & 0 \end{bmatrix} {}^B\mathbf{q}^N \tag{56} $$

Attitude Conversions (Kane/Levinson)

DCM.

$$\vec{v}_B = {}^B\mathbf{R}^A \vec{v}_A$$

$${}^A\mathbf{R}^D = {}^A\mathbf{R}^B\, {}^B\mathbf{R}^C\, {}^C\mathbf{R}^D$$

$$[{}^B\mathbf{R}^A]^T = [{}^B\mathbf{R}^A]^{-1} = {}^A\mathbf{R}^B$$

Axis-Angle.

$${}^N\vec{\phi}^B = \hat{r}\,\theta$$

$${}^N\mathbf{R}^B = \mathrm{e}^{\left[{}^N\vec{\phi}^B\right]_\chi}$$

$${}^N\vec{\phi}^B = [\log({}^N\mathbf{R}^B)]_{\chi^{-1}}$$

Quaternion.

$${}^N\mathbf{q}^B = \begin{bmatrix} \hat{r}\sin(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) \end{bmatrix} = \begin{bmatrix} \vec{q}_{1:3} \\ q_4 \end{bmatrix}$$

$$\vec{q}_{1:3} = \vec{\mathrm{v}}$$
$$q_4 = \mathrm{s}$$

$${}^N\mathbf{R}^B = \mathbf{I}_3 + 2\left[\vec{\mathrm{v}}\right]_\chi\left[\mathrm{s}\,\mathbf{I}_3 + [\vec{\mathrm{v}}]_\chi\right]$$

$$\mathbf{q}_1 = \begin{bmatrix} \vec{\mathrm{v}}_1 \\ \mathrm{s}_1 \end{bmatrix} \qquad \mathbf{q}_2 = \begin{bmatrix} \vec{\mathrm{v}}_2 \\ \mathrm{s}_2 \end{bmatrix}$$

$$\mathbf{q}_1 \odot \mathbf{q}_2 = \begin{bmatrix} \mathrm{s}_1\vec{\mathrm{v}}_2 + \mathrm{s}_2\vec{\mathrm{v}}_1 + \vec{\mathrm{v}}_1 \times \vec{\mathrm{v}}_2 \\ \mathrm{s}_1\mathrm{s}_2 - \vec{\mathrm{v}}_1 \cdot \vec{\mathrm{v}}_2 \end{bmatrix}$$

$${}^A\mathbf{q}^D = {}^A\mathbf{q}^B \odot {}^B\mathbf{q}^C \odot {}^C\mathbf{q}^D$$

$$\begin{bmatrix} \vec{q}_{1:3} \\ q_4 \end{bmatrix}^\dagger = \begin{bmatrix} -\vec{q}_{1:3} \\ q_4 \end{bmatrix}$$

Attitude Kinematics (Kane/Levinson)

N and B can be any two orthogonal rigid vector bases. It is common for N to be a Newtonian reference frame like ECI, and B to be some body fixed orthogonal rigid vector basis for uses in attitude.

DCM.

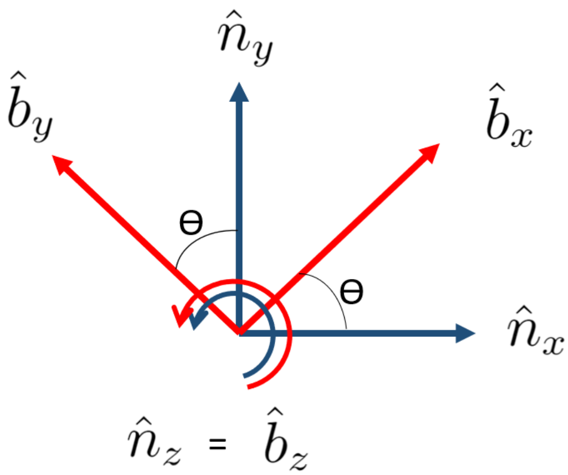
$${}^N\dot{\mathbf{R}}^B = {}^N\mathbf{R}^B\,[{}^N\omega_B^B]_\chi$$
$${}^B\dot{\mathbf{R}}^N = -[{}^N\omega_B^B]_\chi\,{}^B\mathbf{R}^N$$

Quaternion.

$${}^N\dot{\mathbf{q}}^B = \frac{1}{2}{}^N\mathbf{q}^B \odot \begin{bmatrix} {}^N\vec{\omega}_B^B \\ 0 \end{bmatrix}$$
$${}^B\dot{\mathbf{q}}^N = -\frac{1}{2}\begin{bmatrix} {}^N\vec{\omega}_B^B \\ 0 \end{bmatrix} \odot {}^B\mathbf{q}^N$$

Markley/Wertz Conventions

An alternate set of attitude conventions has been made widely popular by James Wertz and Landis Markley with their papers and textbooks. The relationship between the axis-angle and the quaternion is the same for both the Kane/Levinson and Markley/Wertz conventions. The difference comes from the relationship between the DCM and the rotation. For a simple orientation like the one below:



A DCM relating these two bases if $\theta = \frac{\pi}{4}$ would be:

$$^{N}\mathbf{R}^{B} = \begin{bmatrix} \cos(\frac{\pi}{4}) & -\sin(\frac{\pi}{4}) & 0 \\ \sin(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Kane/Levinson convention says this should be parameterized as a rotation from N to B, and the Wertz/Markley convention says this should be parameterized as a rotation from B to N. This results in the following two axis-angle vectors:

$$^{N}\vec{\phi}^{B}_{Kane/Levinson} = \frac{\pi}{4}\hat{n}_z = \frac{\pi}{4}\hat{b}_z$$
$$^{N}\vec{\phi}^{B}_{Wertz/Markley} = -\frac{\pi}{4}\hat{n}_z = -\frac{\pi}{4}\hat{b}_z$$

Because of this, the Kane/Levinson and Wertz/Markley methods are identical for all things DCM related, but when the DCM is parameterized using a rotation, they are the opposites of each other. In order to reconstruct the same DCM, the two conventions have different DCM from axis-angle functions:

$$^{N}\mathbf{R}^{B}(\vec{\phi})_{Kane/Levinson} = \mathrm{e}^{[^{N}\vec{\phi}^{B}]_{\times}}$$

$$^{N}\mathbf{R}^{B}(\vec{\phi})_{Wertz/Markley} = \mathrm{e}^{[-^{N}\vec{\phi}^{B}]_{\times}}$$

This document will examine how this distinction affects axis-angle and quaternion parameterizations in the Wertz/Markley convention. With the Wertz/Markley convention, quaternions can no longer be chained using the Hamilton product:

$$^{A}\mathbf{q}^{D} \neq {}^{A}\mathbf{q}^{B} \odot {}^{B}\mathbf{q}^{C} \odot {}^{C}\mathbf{q}^{D}$$

To multiply quaternions and maintain the multiplicative chaining ability that DCM's display, an additional method of quaternion convention is proposed known as the JPL product. The Hamilton quaternion product is denoted by $\odot$, where as the JPL quaternion product is denoted by $\otimes$.

JPL Product.

$$\mathbf{q}_1 = \begin{bmatrix} \vec{\mathrm{v}}_1 \\ \mathrm{s}_1 \end{bmatrix} \qquad \mathbf{q}_2 = \begin{bmatrix} \vec{\mathrm{v}}_2 \\ \mathrm{s}_2 \end{bmatrix}$$

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} \mathrm{s}_1\vec{\mathrm{v}}_2 + \mathrm{s}_2\vec{\mathrm{v}}_1 - \vec{\mathrm{v}}_1 \times \vec{\mathrm{v}}_2 \\ \mathrm{s}_1\mathrm{s}_2 - \vec{\mathrm{v}}_1 \cdot \vec{\mathrm{v}}_2 \end{bmatrix} \tag{57}$$

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \left[\begin{array}{c|c} \mathrm{s}_2\mathbf{I}_3 + [\vec{\mathrm{v}}_2]_{\times} & \vec{\mathrm{v}}_2 \\ \hline -\vec{\mathrm{v}}_2{}^{T} & \mathrm{s}_2 \end{array}\right] \left[\begin{array}{c} \vec{\mathrm{v}}_1 \\ \hline \mathrm{s}_1 \end{array}\right] \tag{58}$$

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \left[\begin{array}{c|c} \mathrm{s}_1\mathbf{I}_3 - [\vec{\mathrm{v}}_1]_{\times} & \vec{\mathrm{v}}_1 \\ \hline -\vec{\mathrm{v}}_1{}^{T} & \mathrm{s}_1 \end{array}\right] \left[\begin{array}{c} \vec{\mathrm{v}}_2 \\ \hline \mathrm{s}_2 \end{array}\right] \tag{59}$$

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \mathbf{q}_2 \odot \mathbf{q}_1 \tag{60}$$

With JPL quaternion multiplication, quaternions can be chained together in the Markley/Wertz Convention:

$$^{A}\mathbf{q}^{D} = {}^{A}\mathbf{q}^{B} \otimes {}^{B}\mathbf{q}^{C} \otimes {}^{C}\mathbf{q}^{D} \tag{61}$$

***Boxed equations are deviations from the Kane/Levinson convention.**

Attitude Conversions (Markley/Wertz)

DCM.

$$\vec{v}_B = {}^B\mathbf{R}^A\vec{v}_A$$

$${}^A\mathbf{R}^D = {}^A\mathbf{R}^{B\,B}\mathbf{R}^{C\,C}\mathbf{R}^D$$

$$[{}^B\mathbf{R}^A]^T = [{}^B\mathbf{R}^A]^{-1} = {}^A\mathbf{R}^B$$

axis-angle.

$${}^N\vec{\phi}^B = \hat{r}\theta$$

$$\boxed{{}^N\mathbf{R}^B = \mathrm{e}^{[-{}^N\vec{\phi}^B]_\chi}}^*$$

$$\boxed{{}^N\vec{\phi}^B = -[\log({}^N\mathbf{R}^B)]_{\chi^{-1}}}^*$$

Quaternion.

$${}^N\mathbf{q}^B = \begin{bmatrix}\hat{r}\sin(\frac{\theta}{2})\\ \cos(\frac{\theta}{2})\end{bmatrix} = \begin{bmatrix}\vec{q}_{1:3}\\ q_4\end{bmatrix}$$

$$\vec{q}_{1:3} = \vec{\mathrm{v}}$$
$$q_4 = \mathrm{s}$$

$$\boxed{{}^N\mathbf{R}^B = \mathbf{I}_3 - 2[\vec{\mathrm{v}}]_\chi[\mathrm{s}\mathbf{I}_3 - [\vec{\mathrm{v}}]_\chi]}^*$$

$$\mathbf{q}_1 = \begin{bmatrix}\vec{\mathrm{v}}_1\\ \mathrm{s}_1\end{bmatrix} \qquad \mathbf{q}_2 = \begin{bmatrix}\vec{\mathrm{v}}_2\\ \mathrm{s}_2\end{bmatrix}$$

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix}\mathrm{s}_1\vec{\mathrm{v}}_2 + \mathrm{s}_2\vec{\mathrm{v}}_1 - \vec{\mathrm{v}}_1 \times \vec{\mathrm{v}}_2\\ \mathrm{s}_1\mathrm{s}_2 - \vec{\mathrm{v}}_1 \cdot \vec{\mathrm{v}}_2\end{bmatrix}$$

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \mathbf{q}_2 \odot \mathbf{q}_1$$

$$\boxed{{}^A\mathbf{q}^D = {}^A\mathbf{q}^B \otimes {}^B\mathbf{q}^C \otimes {}^C\mathbf{q}^D}^*$$

$$\begin{bmatrix}\vec{q}_{1:3}\\ q_4\end{bmatrix}^\dagger = \begin{bmatrix}-\vec{q}_{1:3}\\ q_4\end{bmatrix}$$

Attitude Kinematics (Markley/Wertz)

N and B can be any two orthogonal rigid vector bases. It is common for N to be a Newtonian reference frame like ECI, and B to be some body fixed orthogonal rigid vector basis for uses in attitude.

DCM.

$${}^N\dot{\mathbf{R}}^B = {}^N\mathbf{R}^B\,[{}^N\omega^B_B]_\chi$$
$${}^B\dot{\mathbf{R}}^N = -[{}^N\omega^B_B]_\chi\,{}^B\mathbf{R}^N$$

Quaternion.

$$\boxed{{}^N\dot{\mathbf{q}}^B = -\frac{1}{2}{}^N\mathbf{q}^B \otimes \begin{bmatrix}{}^N\vec{\omega}^B_B\\ 0\end{bmatrix}}^*$$

$$\boxed{{}^B\dot{\mathbf{q}}^N = \frac{1}{2}\begin{bmatrix}{}^N\vec{\omega}^B_B\\ 0\end{bmatrix} \otimes {}^B\mathbf{q}^N}^*$$

```
function [q] = q_from_DCM(R)
% Scalar Last, Wertz/Markley Convention
T = R(1,1) + R(2,2) + R(3,3)
if T> R(1,1) && T> R(2,2) && T>R(3,3)
    q4 = .5*sqrt(1+T);
    r  = .25/q4;
    q1 = (R(3,2) - R(2,3))*r
    q2 = (R(1,3) - R(3,1))*r
    q3 = (R(2,1) - R(1,2))*r
elseif R(1,1)>R(2,2) && R(1,1)>R(3,3)
    q1 = .5*sqrt(1-T + 2*R(1,1))
    r  = .25/q1
    q4 = (R(3,2) - R(2,3))*r
    q2 = (R(1,2) + R(2,1))*r
    q3 = (R(1,3) + R(3,1))*r
elseif R(2,2)>R(3,3)
    q2 = .5*sqrt(1-T + 2*R(2,2))
    r  = .25/q2
    q4 = (R(1,3) - R(3,1))*r
    q1 = (R(1,2) + R(2,1))*r
    q3 = (R(2,3) + R(3,2))*r
else
    q3 = .5*sqrt(1-T + 2*R(3,3))
    r  = .25/q3
    q4 = (R(2,1) - R(1,2))*r
    q1 = (R(1,3) + R(3,1))*r
    q2 = (R(2,3) + R(3,2))*r
end
% invert quaternion for Wertz/Makley
    convention
q = [-q1;-q2;-q3;q4]
end
```

Alternative Quaternion Kinematics (Wertz/Markley). The following equations are equivalent to those listed above, but are in slightly different forms that may be more useful for writing code.

$$
\begin{aligned}
{}^{B}\mathbf{q}^{N} &= \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T \\
{}^{N}\vec{\omega}_B^{B} &= \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T
\end{aligned}
$$

$$
{}^{B}\dot{\mathbf{q}}^{N} = \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{62}
$$

$$
{}^{B}\dot{\mathbf{q}}^{N} = \frac{1}{2}\begin{bmatrix} q_4\mathbf{I}_3 + [\vec{q}_{1:3}]_\chi \\ -\vec{q}_{1:3}^{T} \end{bmatrix} {}^{N}\vec{\omega}_B^{B} \tag{63}
$$

$$
{}^{B}\dot{\mathbf{q}}^{N} = \frac{1}{2}\begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{64}
$$

$$
{}^{B}\dot{\mathbf{q}}^{N} = \frac{1}{2}\begin{bmatrix} -[{}^{N}\vec{\omega}_B^{B}]_\chi & {}^{N}\vec{\omega}_B^{B} \\ -[{}^{N}\vec{\omega}_B^{B}]^T & 0 \end{bmatrix} {}^{B}\mathbf{q}^{N} \tag{65}
$$

If the quaternion is ${}^{N}\mathbf{q}^{B}$, the kinematics are as such:

$$
\begin{aligned}
{}^{N}\mathbf{q}^{B} &= \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T \\
{}^{N}\vec{\omega}_B^{B} &= \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T
\end{aligned}
$$

$$
{}^{N}\dot{\mathbf{q}}^{B} = \frac{1}{2}\begin{bmatrix} -q_4 & -q_3 & q_2 \\ q_3 & -q_4 & -q_1 \\ -q_2 & q_1 & -q_4 \\ q_1 & q_2 & q_3 \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{66}
$$

$$
{}^{N}\dot{\mathbf{q}}^{B} = \frac{1}{2}\begin{bmatrix} -q_4\mathbf{I}_3 + [\vec{q}_{1:3}]_\chi \\ \vec{q}_{1:3}^{T} \end{bmatrix} {}^{N}\vec{\omega}_B^{B} \tag{67}
$$

$$
{}^{N}\dot{\mathbf{q}}^{B} = \frac{1}{2}\begin{bmatrix} 0 & \omega_z & -\omega_y & -\omega_x \\ -\omega_z & 0 & \omega_x & -\omega_y \\ \omega_y & -\omega_x & 0 & -\omega_z \\ \omega_x & \omega_y & \omega_z & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{68}
$$

$$
{}^{N}\dot{\mathbf{q}}^{B} = \frac{1}{2}\begin{bmatrix} -[{}^{N}\vec{\omega}_B^{B}]_\chi & -{}^{N}\vec{\omega}_B^{B} \\ [{}^{N}\vec{\omega}_B^{B}]^T & 0 \end{bmatrix} {}^{N}\mathbf{q}^{B} \tag{69}
$$

Proofs

Kinematic Transport Theorem. Also known as "The Golden Rule of Vector Differentiation" by Paul Mitiguy from Stanford. This theorem is used to differentiate a vector with respect to a different reference frame.

$$
\frac{{}^{N}d\vec{v}}{dt} = \frac{{}^{B}d\vec{v}}{dt} + {}^{N}\vec{\omega}^{B} \times \vec{v} \tag{70}
$$

This theorem is very useful because a vector expressed in B can not be directly differentiated with respect to N. This theorem lets the vector be differentiated in the reference frame it is expressed in, with the addition of the cross product between the angular velocity of the new reference frame, and the vector. This theorem will be useful in deriving the kinematics of the DCM.

DCM Kinematics Proof. Given below are two methods for deriving the kinematics of a DCM.

Derivation 1. Given a fixed point on a rotating reference frame, the DCM kinematics can be derived. Given a Newtonian (non rotating, non accelerating) reference frame N, a reference frame B is rotating with some angular velocity ${}^{N}\omega^{B}$. If the origins of the N and B reference frames are coincident, the velocity of a point fixed on B is as follows: **Assumptions**

- Origins of B and N are coincident, ($r^{B_0/N_0} = \vec{0}$)
- Point Q is fixed in B, (${}^{B}\vec{v}^{Q} = \vec{0}$)

This means the velocity of point Q in frame N is calculated as follows using the kinematic transport theorem:

$$
\begin{aligned}
{}^{N}\vec{v}_N^{Q} &= \frac{{}^{N}d\vec{r}^{Q/N_0}}{dt} \\
{}^{N}\vec{v}_N^{Q} &= \cancel{\frac{{}^{B}d\vec{r}^{Q/N_0}}{dt}} + {}^{N}\omega^{B} \times \vec{r}^{Q/N_0} \\
{}^{N}\vec{v}_N^{Q} &= {}^{N}\mathbf{R}^{B}\big[[{}^{N}\omega_B^{B}]_\chi\, \vec{r}_B^{Q/N_0}\big]
\end{aligned}
$$

Alternatively, the kinematic transport theorem can be omitted if the vector is transformed to N before being differentiated:

$$
\begin{aligned}
{}^{N}\vec{v}_N^{Q} &= \frac{{}^{N}d[{}^{N}\mathbf{R}^{B}\ \vec{r}_B^{Q/N_0}}{dt}] \\
{}^{N}\vec{v}_N^{Q} &= {}^{N}\dot{\mathbf{R}}^{B}\ \vec{r}_B^{Q/N_0} + \cancel{{}^{N}\mathbf{R}^{B}\ \dot{\vec{r}}_B^{Q/N_0}}
\end{aligned}
$$

By combining both of these results, and removing the $\vec{r}_B^{Q/N_0}$ from both sides, the following expression remains:

$$
{}^{N}\dot{\mathbf{R}}^{B} = {}^{N}\mathbf{R}^{B}\ [{}^{N}\omega_B^{B}]_\chi \tag{71}
$$

This is the kinematic equation for the attitude DCM given an angular velocity taken on the rotating body, expressed in the body frame. If the attitude convention ${}^{B}\mathbf{R}^{N}$ is used, the expression is simply transposed resulting in:

$$
{}^{B}\dot{\mathbf{R}}^{N} = -[{}^{N}\omega_B^{B}]_\chi\ {}^{B}\mathbf{R}^{N} \tag{72}
$$

The change in attitude kinematics that comes with the switched attitude convention is a potential cause for erroneous simulation results.

Derivation 2. If the attitude is slightly perturbed to due to a constant angular velocity ${}^{N}\omega_B^{B}\delta t = \vec{\phi}$ an infinitesimal δt, the new attitude can be expressed as:

$$
\begin{aligned}
{}^{N}\mathbf{R}^{B_{t+\delta t}} &= {}^{N}\mathbf{R}^{B_t\,B_t}\mathbf{R}^{B_{t+\delta t}} \\
{}^{N}\mathbf{R}^{B_{t+\delta t}} &= {}^{N}\mathbf{R}^{\,B_t}\mathrm{e}^{B_t\vec{\phi}^{B_{t+\delta t}}}
\end{aligned}
$$

Given the small angle approximation introduced in equation Eq. (25)

$$
\begin{aligned}
{}^{N}\mathbf{R}^{B_{t+\delta t}} &= {}^{N}\mathbf{R}^{\,B_t}\big[\mathbf{I}_3 + [{}^{B_t}\vec{\phi}^{B_{t+\delta t}}]_\chi\big] \\
{}^{N}\mathbf{R}^{B_{t+\delta t}} &= {}^{N}\mathbf{R}^{\,B_t}\big[\mathbf{I}_3 + [{}^{N}\omega_B^B\delta t]_\chi\big] \\
{}^{N}\mathbf{R}^{B_{t+\delta t}} &= {}^{N}\mathbf{R}^{\,B_t} + {}^{N}\mathbf{R}^{\,B_t}[{}^{N}\omega_B^B\delta t]_\chi \\
{}^{N}\mathbf{R}^{B_{t+\delta t}} &= {}^{N}\mathbf{R}^{\,B_t} + \delta t\,{}^{N}\mathbf{R}^{\,B_t}[{}^{N}\omega_B^B]_\chi
\end{aligned}
$$

Taking a difference of the attitudes before and after the rotation results in the following:

$$
\frac{{}^{N}\mathbf{R}^{B_{t+\delta t}} - {}^{N}\mathbf{R}^{B_t}}{\delta t} = \frac{\delta t\,{}^{N}\mathbf{R}^{\,B_t}[{}^{N}\omega_B^B]_\chi}{\delta t}
$$

The resulting equation for the DCM kinematics matches that of derivation 1 in equation Eq. (71).

$$
{}^{N}\dot{\mathbf{R}}^{B} = {}^{N}\mathbf{R}^{B}\,[{}^{N}\omega^{B}{}_{B}]_\chi
$$

Quaternion Kinematics Proof (Kane/Levinson). Given a very small rotation applied to an initial quaternion:

$$
{}^{N}\mathbf{q}^{B_{t+\delta t}} = {}^{N}\mathbf{q}^{B_t} \odot {}^{B_t}\mathbf{q}^{B_{t+\delta t}}
$$

And assuming a constant angular velocity over this infinitesimal δt in the direction of an axis-angle vector $\vec{\phi} = \hat{r}\theta$:

$$
{}^{N}\mathbf{q}^{B_{t+\delta t}} = {}^{N}\mathbf{q}^{B_t} \odot \begin{bmatrix} \hat{r}\,\sin(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) \end{bmatrix}
$$

The small angle approximation removes the trigonometric functions.

$$
{}^{N}\mathbf{q}^{B_{t+\delta t}} = {}^{N}\mathbf{q}^{B_t} \odot \begin{bmatrix} \hat{r}\,\frac{\theta}{2} \\ 1 \end{bmatrix}
$$

Given that $\hat{r}\theta = \vec{\phi}$, and that given constant angular velocity, $[{}^{N}\vec{\omega}_B^B]\delta t = {}^{B_t}\vec{\phi}^{B_{t+\delta t}}$, this can be reduced further. For the remainder of the proof, ${}^{N}\vec{\omega}_B^B$ will be abbreviated with $\vec{\omega}$.

$$
\begin{aligned}
{}^{N}\mathbf{q}^{B_{t+\delta t}} &= {}^{N}\mathbf{q}^{B_t} \odot \begin{bmatrix} \frac{{}^{B_t}\vec{\phi}^{B_{t+\delta t}}}{2} \\ 1 \end{bmatrix} \\
{}^{N}\mathbf{q}^{B_{t+\delta t}} &= {}^{N}\mathbf{q}^{B_t} \odot \begin{bmatrix} \frac{\vec{\omega}\,\delta t}{2} \\ 1 \end{bmatrix} \\
{}^{N}\mathbf{q}^{B_{t+\delta t}} &= {}^{N}\mathbf{q}^{B_t} \odot \left[\begin{bmatrix} \vec{0} \\ 1 \end{bmatrix} + \begin{bmatrix} \frac{\vec{\omega}\,\delta t}{2} \\ 0 \end{bmatrix} \right]
\end{aligned}
$$

Given the identity quaternion discussed in equation (36), this can be simplified:

$$
{}^{N}\mathbf{q}^{B_{t+\delta t}} = {}^{N}\mathbf{q}^{B_t} + {}^{N}\mathbf{q}^{B_t} \odot \begin{bmatrix} \frac{\vec{\omega}\,\delta t}{2} \\ 0 \end{bmatrix}
$$

Equation can be differenced to find the derivative of the quaternion:

$$
\begin{aligned}
{}^{N}\dot{\mathbf{q}}^{B(t)} &= \frac{{}^{N}\mathbf{q}^{B_{t+\delta t}} - {}^{N}\mathbf{q}^{B_t}}{\delta t} \\
{}^{N}\dot{\mathbf{q}}^{B(t)} &= \frac{{}^{N}\mathbf{q}^{B_t} \odot \begin{bmatrix} \frac{\vec{\omega}\,\delta t}{2} \\ 0 \end{bmatrix}}{\delta t}
\end{aligned}
$$

Given the nature of quaternion multiplication described in equation (40), in the case where a scalar is zero, constants can be pulled out of the corresponding vector component and placed outside the multiplication operation. This now defines the quaternion kinematic ODE ready for use in simulation.

$$
{}^{N}\dot{\mathbf{q}}^{B} = \frac{1}{2}{}^{N}\mathbf{q}^{B} \odot \begin{bmatrix} {}^{N}\vec{\omega}_B^B \\ 0 \end{bmatrix} \tag{73}
$$

Quaternion to DCM Proof (Kane/Levinson). Looking back at the Eq. (14) to transform a vector using a quaternion:

$$
\begin{bmatrix} \vec{x}_N \\ 0 \end{bmatrix} = {}^{N}\mathbf{q}^{B} \odot \begin{bmatrix} \vec{x}_B \\ 0 \end{bmatrix} \odot [{}^{N}\mathbf{q}^{B}]^\dagger
$$

$$
\begin{bmatrix} \vec{x}_N \\ 0 \end{bmatrix} = \begin{bmatrix} \vec{v} \\ s \end{bmatrix} \odot \begin{bmatrix} \vec{x}_B \\ 0 \end{bmatrix} \odot \begin{bmatrix} -\vec{v} \\ s \end{bmatrix}
$$

$$
\begin{bmatrix} \vec{x}_N \\ 0 \end{bmatrix} = \begin{bmatrix} \vec{v} \\ s \end{bmatrix} \odot \left[\begin{array}{c:c} s\mathbf{I}_3 + [\vec{v}]_\chi & -\vec{v} \\ \hdashline \vec{v}^T & s \end{array}\right] \left[\begin{array}{c} \vec{x}_B \\ \hdashline 0 \end{array}\right]
$$

$$
\begin{bmatrix} \vec{x}_N \\ 0 \end{bmatrix} = \left[\begin{array}{c:c} s\mathbf{I}_3 + [\vec{v}]_\chi & \vec{v} \\ \hdashline -\vec{v}^T & s \end{array}\right] \left[\begin{array}{c:c} s\mathbf{I}_3 + [\vec{v}]_\chi & -\vec{v} \\ \hdashline \vec{v}^T & s \end{array}\right] \left[\begin{array}{c} \vec{x}_B \\ \hdashline 0 \end{array}\right]
$$

$$
\begin{bmatrix} \vec{x}_N \\ 0 \end{bmatrix} = \left[\begin{array}{c:c} (s\mathbf{I}_3 + [\vec{v}]_\chi)^2 + \vec{v}\vec{v}^T & -(s\mathbf{I}_3 + [\vec{v}]_\chi)\vec{v} + s\vec{v} \\ \hdashline -\vec{v}^T(s\mathbf{I}_3 + [\vec{v}]_\chi) + s\vec{v}^T & \vec{v}^T\vec{v} + s^2 \end{array}\right] \left[\begin{array}{c} \vec{x}_B \\ \hdashline 0 \end{array}\right]
$$

This is reduced to the following:

$$
\vec{x}_N = \big[(s\mathbf{I}_3 + [\vec{v}]_\chi)^2 + \vec{v}\vec{v}^T\big]\vec{x}_B \tag{74}
$$

$$
\vec{x}_N = \big[s_3^{\mathbf{I}} + s[\vec{v}]_\chi + s[\vec{v}]_\chi + [\vec{v}]_\chi[\vec{v}]_\chi + \vec{v}\vec{v}^T\big]\vec{x}_B \tag{75}
$$

Given the following two properties:

$$
\vec{v}\vec{v}^T = [\vec{v}]_\chi[\vec{v}]_\chi + (\vec{v}^T\vec{v})\mathbf{I}_3
$$

And

$$
\begin{aligned}
||\mathbf{q}|| &= 1 \\
\vec{v}^T\vec{v} + s^2 &= 1 \\
\vec{v}^T\vec{v} &= 1 - s^2
\end{aligned}
$$

Substituting back into Eq. (75), the result is:

$$
\vec{x}_N = \big[\mathbf{I}_3 + 2\vec{v}_\chi[s\mathbf{I}_3 + \vec{v}_\chi]\big]\vec{x}_B
$$

Resulting in the following DCM function:

$$
{}^{A}\mathbf{R}^{B} = R({}^{A}\mathbf{q}^{B}) = \mathbf{I}_3 + 2\vec{v}_\chi[s\mathbf{I}_3 + \vec{v}_\chi] \tag{76}
$$

Bibliography.

1. Paul Mitiguy *Advanced Dynamics and Motion Simulation.* Motion Genesis, 2018.
2. Thomas Kane, Peter Likins, David Levinson *Spacecraft Dynamics* McGraw-Hill Book Company, 1983
3. James Wertz *Spacecraft Attitude Determination and Control* Kluwer Academic Publishers, 1978
4. F. Landis Markley, John L. Crassidis *Fundamentals of Spacecraft Attitude Determination and Control* Springer, 2014