

16.61 Spring 2006  
Lecture 7 Lecture Notes  
ver 1.0

Charles P. Coleman

March 2, 2006

## Review

### Coriolis' Transport Theorem

$$\left\{ \frac{dr}{dt} \right\}_A = \left\{ \frac{dr}{dt} \right\}_B + S(\omega)r$$

### Inertial Acceleration

$$\left\{ \frac{d^2r}{dt^2} \right\}_A = \left\{ \frac{d^2r}{dt^2} \right\}_B + S(\dot{\omega})r + 2S(\omega) \left\{ \frac{dr}{dt} \right\}_B + S(\omega)S(\omega)r$$

### Basic Transformation Matrices

$$R_{\Phi,x} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\Phi & s\Phi \\ 0 & -s\Phi & c\Phi \end{pmatrix} \quad R_{\Theta,y} = \begin{pmatrix} c\Theta & 0 & -s\Theta \\ 0 & 1 & 0 \\ s\Theta & 0 & c\Theta \end{pmatrix} \quad R_{\Psi,z} = \begin{pmatrix} c\Psi & s\Phi & 0 \\ -s\Psi & c\Phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Or their transposes! Know which way you are going in frames! Above they are taking vectors from the reference frame to the rotated frame. In Spong & Vidyassagar and Homework 2, the rotation matrices are taking vectors from the rotated frame to the base frame.

## Rotation Matrix Properties

$$\begin{aligned}R^{-1} &= R^T \\RR^T &= I \\RR^T &= I \\ \det R &= +1 \\ \dot{R} &= S(\omega)R \\ R(t) &= e^{S(\omega t)}R(0)\end{aligned}$$

$$S + S^T = 0$$

$$S = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$$

## Homogenous Transformations

$$\begin{aligned}H_{x,a} &= \begin{pmatrix} & a \\ I & 0 \\ & 0 \\ 0 & 1 \end{pmatrix} & H_{y,b} &= \begin{pmatrix} & 0 \\ I & b \\ & 0 \\ 0 & 1 \end{pmatrix} & H_{z,c} &= \begin{pmatrix} & 0 \\ I & 0 \\ & c \\ 0 & 1 \end{pmatrix} \\ \\ H_{R_x,\theta} &= \begin{pmatrix} & 0 \\ R_x(\theta) & 0 \\ & 0 \\ 0 & 1 \end{pmatrix} & H_{R_y,\theta} &= \begin{pmatrix} & 0 \\ R_y(\theta) & 0 \\ & 0 \\ 0 & 1 \end{pmatrix} & H_{R_z,\theta} &= \begin{pmatrix} & 0 \\ R_z(\theta) & 0 \\ & 0 \\ 0 & 1 \end{pmatrix}\end{aligned}$$

And how to compose them and calculate velocities and accelerations with them

## Addition of Angular Velocities

$$\omega_0^n = \omega_0^1 + R_0^1 \omega_1^2 + \cdots + R_0^{n-1} \omega_{n-1}^n$$

## Newton's Equation

$$F = ma$$

## Everything Euler! Angles, Equations, Rates, and Parameters

### Euler's Equation

Just as we have found the linear acceleration of point masses or points in bodies, and used Newton's equations to find the translational dynamics ( $F = ma$ ), we might want to find angular acceleration of rigid bodies, and then calculate the object's rotational dynamics. To solve this problem we need Euler's equation, which says that the rate of change of the angular momentum of a rigid body is equal to the torques applied to the body. This statement is valid only in an inertial frame.

In the body frame  $B$  we can write

$$\frac{d}{dt} \{H_B\}_A = T_B$$

where  $H_B$  is the angular momentum vector of the rigid vehicle, and  $T_B$  is the net torque acting about the cg of the body (aircraft). The subscript  $A$  represents differentiation with respect to (w.r.t.) the inertial frame  $A$ . The body is rotating with angular velocity  $\omega^{B/A} = \omega_B$  with respect to the inertial frame  $A$ .

The angular momentum of a mass element about the origin of the body frame  $B$ , is the moment of the linear momentum

$$\delta H_B = r_B \times (\omega_B \times r_B) \delta m$$

### Calculation of Body Angular Momentum $H_B$

An element of mass,  $\delta m$ , with position  $r_B$ , in a rotating body, has instantaneous velocity  $(\omega_B \times r_B)$ , and infinitesimal linear momentum

$$\delta(\text{momentum}) = (\omega_B \times r_B) \delta m$$

Let  $\omega_B$  and  $r_B$  be as follows

$$\begin{aligned}\omega_B &= Pb_1 + Qb_2 + Rb_3 \\ r_B &= xb_1 + yb_2 + zb_3\end{aligned}$$

We use the vector triple product to evaluate the expression for  $\delta H_B$

$$\begin{aligned}u \times (v \times w) &= v(w \cdot u) - w(u \cdot v) \\ r_B \times (\omega_B \times r_B) &= \omega_B(r_B \cdot r_B) - r_B(r_B \cdot \omega_B) \\ r_B \times (\omega_B \times r_B) &= \begin{pmatrix} P \\ Q \\ R \end{pmatrix} (x^2 + y^2 + z^2) - \begin{pmatrix} x \\ y \\ z \end{pmatrix} (Px + Qy + Rz)\end{aligned}$$

Now we can calculate  $\delta H_B$

$$\begin{aligned}\delta H_B &= [r_B \times (\omega_B \times r_B)]\delta m \\ \delta H_B &= \left[ \begin{pmatrix} P \\ Q \\ R \end{pmatrix} (x^2 + y^2 + z^2) - \begin{pmatrix} x \\ y \\ z \end{pmatrix} (Px + Qy + Rz) \right] \delta m\end{aligned}$$

Rearranging the terms and integrating over the domain of the body we calculate the angular momentum of the body to be

$$H_B = \begin{pmatrix} P \int (y^2 + z^2) dm & -Q \int xy dm & -R \int xz dm \\ -P \int yx dm & +Q \int (x^2 + z^2) dm & -R \int yz dm \\ -P \int zx dm & -Q \int zy dm & +R \int (x^2 + y^2) dm \end{pmatrix}$$

We use a little matrix algebra to rearrange this

$$H_B = \begin{pmatrix} \int (y^2 + z^2) dm & - \int xy dm & - \int xz dm \\ - \int yx dm & \int (x^2 + z^2) dm & - \int yz dm \\ - \int zx dm & - \int zy dm & \int (x^2 + y^2) dm \end{pmatrix} \begin{pmatrix} P \\ Q \\ R \end{pmatrix}$$

The integrals above are defined to be the moments and cross-product moments of inertia, for example

$$J_{xx} = \int (y^2 + z^2) dm \quad J_{xy} = J_{yx} = \int xy dm$$

We can now write the angular momentum of the body as

$$H_B = \begin{pmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{pmatrix} \begin{pmatrix} P \\ Q \\ R \end{pmatrix}$$

or

$$H_B = J\omega_B$$

## Coriolis' Theorem Applied to Euler's Equation

We can now return to the calculation of Euler's equation with the assistance of Coriolis' theorem.

$$\begin{aligned} \frac{d}{dt} \{H_B\}_A &= T_B \\ \dot{H}_B + \omega_B \times H_B &= T_B \end{aligned}$$

If  $J$  is constant this becomes

$$\begin{aligned} J\dot{\omega}_B &= -\omega_B \times J\omega_B + T_B \\ \dot{\omega}_B &= -J^{-1}(\omega_B \times J\omega_B) + J^{-1}T_B \end{aligned}$$

## Essence of Simulating Aircraft Rotational Dynamics

This is great! We now know how the time rate of change of the angular velocity  $\dot{\omega}_B$ , is affected by the inertial properties of the body  $J$  and any applied torques  $T_B$ . So hopefully we should be able to figure out how the orientation (an element  $R$  in  $SO(3)$ ) of the body (aircraft) evolves in time.

Remember

$$\dot{R} = S(\omega_B)R$$

We will first integrate the expression for  $\dot{\omega}_B$

$$\dot{\omega}_B = -J^{-1}(\omega_B \times J\omega_B) + J^{-1}T_B$$

to find  $\omega_B$  given the inertial properties of the body (aircraft) and applied torques (propulsion or control surfaces).

We will then use  $\omega_B$  and integrate the strap down equations

$$\dot{R} = S(\omega_B)R$$

to find the orientation of the body  $R(t)$  as a function of time.

We essentially have the following system of (nonlinear) equations

$$\begin{pmatrix} \dot{\omega}_B \\ \dot{R} \end{pmatrix} = \begin{pmatrix} -J^{-1}S(\omega_B)J\omega_B + J^{-1}T_B \\ S(\omega_B)R \end{pmatrix}$$

That's it! That's the core of simulating the rotational dynamics of an aircraft. Now  $J$  may not be constant, but that can be finessed. The next essential question is calculating  $T_B$  based on the aerodynamic characteristics of an aircraft and its control surfaces, but we delay calculating models for  $T_B$  until the last third of the class.

## Parameterization of $R$

One big computational issue (in practice and simulation) is the parameterization of  $R$ . There are at least three cases that are of interest in aerospace simulations and applications. Each has cost and benefits that have to be weighed when choosing an approach to simulation and/or application.

### Case 1: You use Direction Cosines

In strap down inertial reference systems, the body rates  $\omega_B$ , are measured by rate gyros that are “strapped down” to the vehicle.<sup>1</sup> This is great! Plug  $\omega_B$  into Euler's equation, and plug in  $\omega_B$  and the current  $R$  matrix into the strap down equations. Then numerically integrate to find the new values of  $\omega_B$  and  $R$ . Note there are nine equations to integrate to find  $R(t)$ , the body orientation.

$$\begin{pmatrix} \dot{\omega}_B \\ \dot{R} \end{pmatrix} = \begin{pmatrix} -J^{-1}S(\omega_B)J\omega_B + J^{-1}T_B \\ S(\omega_B)R \end{pmatrix}$$

Gosh! Do we *really* need to integrate nine differential equations to find  $R(t)$ , the orientation as a function of time?

### Case 2: You use (standard) Euler Angles roll-pitch-yaw ( $\Phi, \Theta, \Psi$ )

Let's work on integrating fewer differential equations to find  $R(t)$ , the attitude, by choosing a different parameterization for  $SO(3)$ .

---

<sup>1</sup>Wie (1998), p 327.

Let the aircraft-body coordinate (ABC) frame be aligned  $x$ -forward,  $y$ -starboard, and  $z$ -downward  $\{b_1, b_2, b_3\}$ . Use a Cardan Euler angle representation to relate vectors in the north-east-down (NED) on the earth's surface to the ABC frame in the aircraft.

$$p_{ABC} = R p_{NED}$$

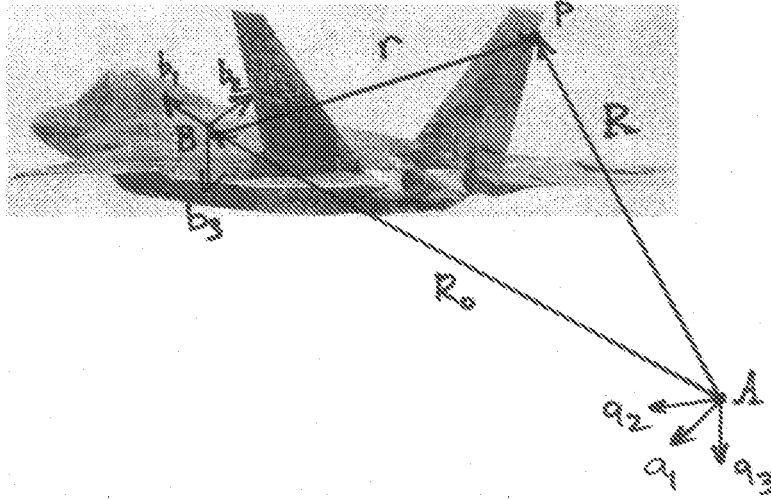
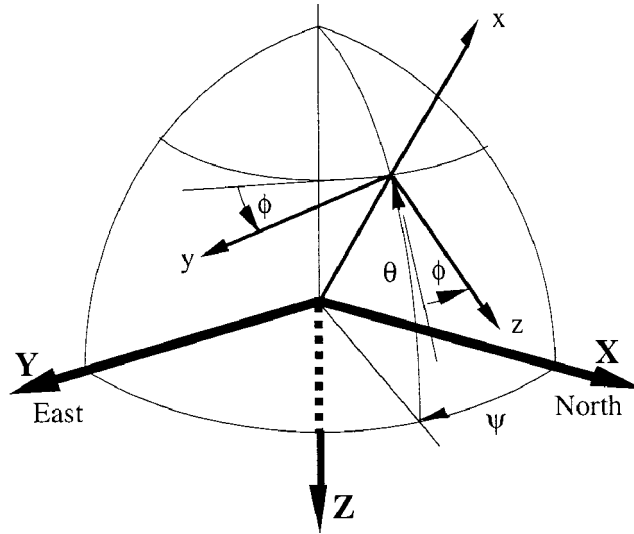


Figure 1: Aircraft Simulation Frames

Starting from the reference NED frame:

1. Rotate about the  $z$ -axis ( $b_3$ ), nose right (positive “yaw”  $\Psi$ , positive heading).
2. Rotate about the new  $y$ -axis ( $b'_2$ ), nose up (positive “pitch”  $\Theta$ , positive elevation).
3. Rotate about the new  $x$ -axis ( $b''_1$ ), right wing down (positive “roll”  $\Phi$ , positive bank).

$$R = R_{\Phi,x} R_{\Theta,y} R_{\Psi,z}$$



$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\Phi & s\Phi \\ 0 & -s\Phi & c\Phi \end{pmatrix} \begin{pmatrix} c\Theta & 0 & -s\Theta \\ 0 & 1 & 0 \\ s\Theta & 0 & c\Theta \end{pmatrix} \begin{pmatrix} c\Psi & s\Phi & 0 \\ -s\Psi & c\Phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

We'd like to know the Euler angle rates  $(\dot{\Phi}, \dot{\Theta}, \dot{\Psi})$  as a function of  $\omega_B$  (we'll get  $\omega_B$  from integrating Euler's equation!) so we can integrate these three differential equations and have the attitude,  $R(\Phi, \Theta, \Psi)$ , as function of time.

We know

$$\dot{R} = S(\omega_B)R$$

We could multiply through and then differentiate to relate the Euler angle rates  $(\dot{\Phi}, \dot{\Theta}, \dot{\Psi})$ , and the angular velocity,  $\omega_B$ , but the following relation is algebraically simpler. We know

$$R = R_0^1 R_1^2 R_2^3$$

then

$$\omega_0^3 = \omega_0^1 + R_0^1 \omega_1^2 + R_0^2 \omega_2^3$$

So the angular velocity of the body is related to the Euler angle rates by

$$\omega_B = \begin{pmatrix} \dot{\Phi} \\ 0 \\ 0 \end{pmatrix} + R_{\Phi,x} \begin{pmatrix} 0 \\ \dot{\Theta} \\ 0 \end{pmatrix} + R_{\Phi,x} R_{\Theta,y} \begin{pmatrix} 0 \\ 0 \\ \dot{\Psi} \end{pmatrix}$$

Multiplying through we find

$$\omega_B = \begin{pmatrix} 1 & 0 & -s\Theta \\ 0 & c\Phi & s\Phi c\Theta \\ 0 & -s\Phi & c\Phi c\Theta \end{pmatrix} \begin{pmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{pmatrix}$$

This matrix equation needs to be inverted

$$\begin{pmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{pmatrix} = \frac{1}{c\Theta} \begin{pmatrix} c\Theta & s\Phi s\Theta & c\Phi s\Theta \\ 0 & c\Phi c\Theta & -s\Phi c\Theta \\ 0 & s\Phi & c\Phi \end{pmatrix} \omega_B = A\omega_B$$

OK, so instead of integrating nine strap down equations  $\dot{R} = S(\omega_B)R$  given  $\omega_B$  (as in Case 1), we now only have to integrate three Euler angle equations given  $\omega_B$ .<sup>2</sup>

$$\begin{pmatrix} \dot{\omega}_B \\ \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{pmatrix} = \begin{pmatrix} -J^{-1}S(\omega_B)J\omega_B + J^{-1}T_B \\ A(\Phi, \Theta, \Psi)\omega_B \end{pmatrix}$$

HOWEVER, when the roll angle is  $\pi/2$  the inverse relation,  $A$ , blows up! No large angle maneuvers for this single Euler parameterization of  $SO(3)$ !<sup>3</sup>

Is there a parameterization of  $R$  that has fewer than nine differential equations to integrate, doesn't blow up if we are trying to simulate or track large angle maneuvers, and doesn't involve costly trigonometric functions?

### Case 3: You use Euler Parameters (quaternions), a four parameter representation of $SO(3)$

On a search for "hypercomplex numbers", William Hamilton of Hamilton's principle and Hamilton's equation fame, discovered what he called quaternions which are often (confusingly) known as Euler parameters ( $\epsilon$ ). A quaternion is a 4-tuple  $(q_0, q_1, q_2, q_3)$  sometimes written as

$$q = q_0 + q_1i + q_2j + q_3k$$

---

<sup>2</sup>But these three equations contain trigonometric functions which could be costly to compute! Nothing's for free, baby!

<sup>3</sup>This is your first indication that even though  $SO(3)$  is three-dimensional it needs at least two "charts" (in an atlas) to describe it!

with multiplicative relations

$$\begin{aligned} i^2 = j^2 = k^2 = ijk &= -1 \\ ij = k = -ji \\ jk = i = -kj \\ ki = j = -ik \end{aligned}$$

The complex conjugate of  $q$  is denoted by  $q^*$  and is given by

$$q = q_0 - q_1i - q_2j - q_3k$$

The norm (length) of a quaternion  $|q|$  is given by

$$|q|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2$$

Considering only unit quaternions  $q$

$$|q|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

it can be shown that any rotation matrix  $R$  can be written as

$$R(q) = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}$$

It should be clear that the four parameter unitary quaternions  $q$  and  $-q$  yield the same  $R$  matrix! This is amazing! Each element of  $\text{SO}(3)$  is then covered by two elements of the the group  $SU(2)$ , the special unitary group of order two. Unitary quaternions “represent” elements of the group  $SU(2)$ . The above expression for  $R$  tells us how to get an element of  $\text{SO}(3)$  from an element of  $SU(2)$ . (Aside:  $SU(2)$  is very important in theoretical physics!)

In some sense, the four parameter unitary quaternion ( $SU(2)$ ) representation of  $\text{SO}(3)$  gives us the two charts needed to create a global “atlas” of  $\text{SO}(3)$ , something NO three parameter Euler representation (or any other 3 parameter representation!) can do. This also indicates that we should be able to find four differential relations between the quaternion rates ( $\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4$ ) and the body angular velocity  $\omega_B$ . And since we didn’t see any trigonometric functions in the expression for  $R(q)$ , these differential equations may be linear!

Turns out we can write the unitary quaternion  $q$  as a function of Euler angles. This gives us a start!<sup>4</sup>

$$\begin{aligned} q_0 &= \pm c(\Phi/2)c(\Theta/2)c(\Psi/2) + s(\Phi/2)s(\Theta/2)s(\Psi/2) \\ q_1 &= \pm s(\Phi/2)c(\Theta/2)c(\Psi/2) - c(\Phi/2)s(\Theta/2)s(\Psi/2) \\ q_2 &= \pm c(\Phi/2)s(\Theta/2)c(\Psi/2) + s(\Phi/2)c(\Theta/2)s(\Psi/2) \\ q_3 &= \pm c(\Phi/2)c(\Theta/2)s(\Psi/2) - s(\Phi/2)s(\Theta/2)c(\Psi/2) \end{aligned}$$

Choose plus (+) or minus (-) and stick with it!

Then it turns out that the quaternion rates  $\dot{q} = (\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4)^T$  can be written as a linear function of  $\omega_B$  using a special  $4 \times 4$  skew matrix  $S_q(\omega_B)$ .

$$\begin{aligned} \dot{q} &= -\frac{1}{2}S_q(\omega_B)q \\ \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} &= -\frac{1}{2} \begin{pmatrix} 0 & P & Q & R \\ -P & 0 & -R & Q \\ -Q & R & 0 & -P \\ -R & -Q & P & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} \end{aligned}$$

This is great! Four linear differential equations as opposed to nine linear or 3 nonlinear. No blowup and no trigonometric functions. A quaternion representation lends itself nicely to onboard calculations and computer simulations which is why they are currently so popular in aircraft and spacecraft applications and simulations.

Pulling the story all together we now have the following quaternion story for the rotational dynamics of an aircraft:

$$\begin{pmatrix} \dot{\omega}_B \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} = \begin{pmatrix} -J^{-1}S(\omega_B)J\omega_B + J^{-1}T_B \\ \\ -\frac{1}{2}S_q(\omega_B)q \\ \end{pmatrix}$$

## Summary

For numerically simulating the rotation dynamics of an aircraft you essentially have three basic choices:

---

<sup>4</sup> $q$  can also be written as a function of the direction cosines.

### Direction Cosines

$$\begin{pmatrix} \dot{\omega}_B \\ \dot{R} \end{pmatrix} = \begin{pmatrix} -J^{-1}S(\omega_B)J\omega_B + J^{-1}T_B \\ S(\omega_B)R \end{pmatrix}$$

### Euler Angles

$$\begin{pmatrix} \dot{\omega}_B \\ \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{pmatrix} = \begin{pmatrix} -J^{-1}S(\omega_B)J\omega_B + J^{-1}T_B \\ A(\Phi, \Theta, \Psi)\omega_B \end{pmatrix}$$

### Quaternions/Euler Parameters

$$\begin{pmatrix} \dot{\omega}_B \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} = \begin{pmatrix} -J^{-1}S(\omega_B)J\omega_B + J^{-1}T_B \\ -\frac{1}{2}S_q(\omega_B)q \end{pmatrix}$$

You actually have a few others, but these are the basic three you should *really* be familiar with!

## Pulling it All Together: Aircraft Simulation

We can pull all of what we've been discussing over the last few weeks together into one coherent picture about aircraft simulation.

### Background Theory

In the beginning of the course I mentioned that rigid body motions had a special structure, that they evolved on the Euclidean group  $E(3)$  which was a semidirect product of translations represented by points  $p$  in three-space  $\mathbb{R}^3$  (a group!), and proper rotations in three-space represented by elements  $R$  of the special orthogonal group  $SO(3)$ .

$$E(3) = \mathbb{R}^3 \ltimes SO(3)$$

We then stated that Newton's equation,

$$\frac{d}{dt_A}(mv) = F$$

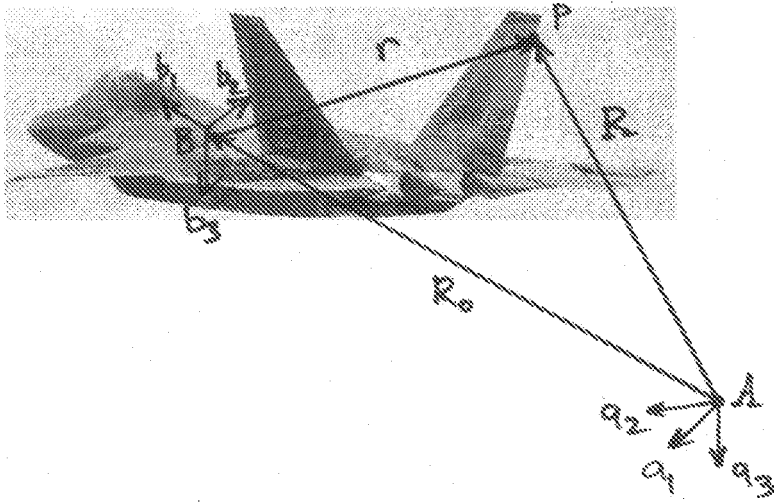


Figure 2: Aircraft Simulation Frames

described the translational dynamics ( $\mathbb{R}^3$ ) of the center of mass of the rigid body where the inertial parameter  $m$  codifies all of the properties of the body that are needed for this calculation, and that Euler's equation,

$$\frac{d}{dt_A} (J\omega) = T$$

described the rotational dynamics (on  $SO(3)$ ) of a rigid body where the inertial "tensor"  $J$  encodes all of the properties that are need for the calculation of rotational dynamics.<sup>5</sup>

In both cases, the derivative has to be taken with respect to an inertial frame  $A$ . In the case that a frame  $B$  is rotating with respect to the inertial frame  $A$  at the angular rate  $\omega^{B/A} = \omega_B$ , then Coriolis' Theorem

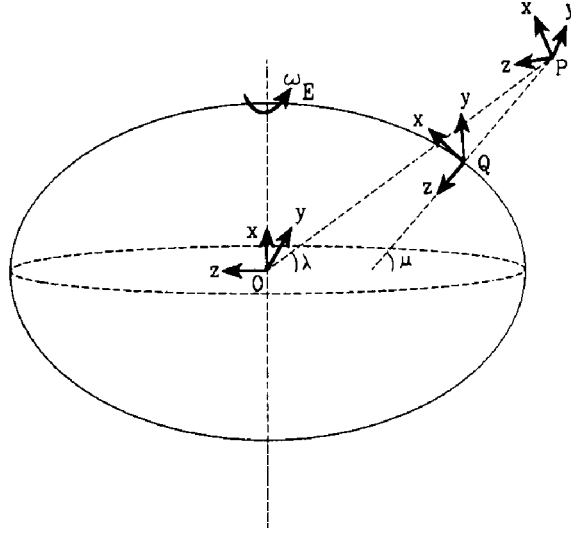
$$\frac{d}{dt_A} r_B = \frac{d}{dt_B} r_B + S(\omega_B)r_B$$

can be applied to calculate the derivative of any vector  $r_B$  in the body frame  $B$  with respect to the inertial frame  $A$ .

---

<sup>5</sup>The fact that the dynamics on  $E(3)$  decouple so nicely is almost magic! But more on that later.

## The Simulator



Suppose that in an Earth-centered inertial (ECI) frame, the center of gravity (cg) of an aircraft is at point  $p$ . Further suppose that that  $R_B^A = R$  is the rotational matrix that takes vectors from the inertial ECI frame (Frame  $A$ ) to the aircraft-body coordinate (ABC) frame (Frame  $B$ ). Let the aircraft cg have velocity  $v_B$  in the ABC frame, let  $\omega_E$  be the angular velocity of the Earth, and let  $\omega_B$  be the angular velocity of the aircraft. Then, we can write the complete dynamics of the aircraft (Translational+Rotational/Newton+Euler) as a set of (nonlinear) differential equations.

$$\begin{pmatrix} \dot{p} \\ \dot{v}_B \\ \dot{\omega}_B \\ \dot{q} \end{pmatrix} = \begin{pmatrix} S(\omega_E) & R^T & 0 & 0 \\ Rg(p) - RS^2(\omega_E) & -[S(\omega_B) + RS(\omega_E)] & 0 & 0 \\ 0 & 0 & -J^{-1}S(\omega_B)J & 0 \\ 0 & 0 & 0 & -\frac{1}{2}S_q(\omega_B) \end{pmatrix} \begin{pmatrix} p \\ v_B \\ \omega_B \\ q \end{pmatrix} + \begin{pmatrix} 0 \\ F_B/m \\ J^{-1}T_B \\ 0 \end{pmatrix}$$

This system of equations results from the straight forward application of Newton's and Euler's equations and Coriolis' Theorem. Its solution is at the core of (round-the-earth) simulations. The system can be modified easily for flat-earth simulations, and quaternions do not have to be used to integrate the kinematic attitude equations. Local wind models can also be added to the simulation.

$F_B$  represents the sum of the aerodynamic and propulsion forces, and  $T_B$  the aerodynamic and propulsion torques. These result from the aircraft model chosen or developed. A simulation package might consist of a core that integrates the above system of equations, and several packages that calculate  $m$ ,  $J$ ,  $F_B$  and  $T_B$  for a given aircraft.