Christopher Lum
lum@uw.edu

Lecture 10a
Navigation Equations

Lecture is on YouTube

The YouTube video entitled ‘The Navigation Equations: Computing Position North, East, and Down’ that covers this lecture is located at https://youtu.be/XQZV-YZ7asE.

Outline

Navigation Equations

Recall that the first 3 states of our flat earth, RCAM model

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \overline{V}^b$$

Navigation equations relate the velocity in the body frame to the velocity in the north east down frame.

From our discussion on Euler angles, we know that we can rotate from a vector expressed in the vehicle carried NED frame (F_v) to a vector expressed in the body frame (F_b) using a series of 3 rotations

$$C_{1/v}(\psi) = \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$C_{2/1}(\theta) = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix}$$

$$C_{b/2}(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix}$$

Together, these formed the DCM

$$C_{b/v}(\phi, \theta, \psi) = C_{b/2}(\phi)\, C_{2/1}(\theta)\, C_{1/v}(\psi)$$

```
          ⎛ Cos[ψ]   Sin[ψ]  0 ⎞
C1v[ψ_] = ⎜ -Sin[ψ]  Cos[ψ]  0 ⎟;
          ⎝   0        0     1 ⎠

          ⎛Cos[θ]  0  -Sin[θ]⎞
C21[θ_] = ⎜  0     1     0   ⎟;
          ⎝Sin[θ]  0  Cos[θ] ⎠

          ⎛1     0        0   ⎞
Cb2[ϕ_] = ⎜0  Cos[ϕ]   Sin[ϕ] ⎟;
          ⎝0  -Sin[ϕ]  Cos[ϕ] ⎠

Cbv[ϕ_, θ_, ψ_] = Cb2[ϕ].C21[θ].C1v[ψ];
```

Taking the transpose of this yields the reverse transformation

$$\overline{V}^{v} = C_{v/b}(\phi, \theta, \phi)\, \overline{V}^{b}$$

```
Cvb[ϕ_, θ_, ψ_] = Cbv[ϕ, θ, ψ] // Transpose;
Cvb[ϕ, θ, ψ] // MatrixForm
```

```
⎛Cos[θ] Cos[ψ]  Cos[ψ] Sin[θ] Sin[ϕ] - Cos[ϕ] Sin[ψ]   Cos[ϕ] Cos[ψ] Sin[θ] + Sin[ϕ] Sin[ψ] ⎞
⎜Cos[θ] Sin[ψ]  Cos[ϕ] Cos[ψ] + Sin[θ] Sin[ϕ] Sin[ψ]  -Cos[ψ] Sin[ϕ] + Cos[ϕ] Sin[θ] Sin[ψ] ⎟
⎝   -Sin[θ]                Cos[θ] Sin[ϕ]                          Cos[θ] Cos[ϕ]            ⎠
```

```
                       ⎛u⎞
Vv = Cvb[ϕ, θ, ψ].     ⎜v⎟;
                       ⎝w⎠
Vv // MatrixForm
```

```
⎛ u Cos[θ] Cos[ψ] + v (Cos[ψ] Sin[θ] Sin[ϕ] - Cos[ϕ] Sin[ψ]) + w (Cos[ϕ] Cos[ψ] Sin[θ] + Sin[ϕ] S
⎜u Cos[θ] Sin[ψ] + w (-Cos[ψ] Sin[ϕ] + Cos[ϕ] Sin[θ] Sin[ψ]) + v (Cos[ϕ] Cos[ψ] + Sin[θ] Sin[ϕ] S
⎝                     w Cos[θ] Cos[ϕ] - u Sin[θ] + v Cos[θ] Sin[ϕ]
```

This is what are listed as the 'Navigation Equations' on pg. 110 of textbook (except instead of position down, they use altitude)

This simply rotates the inertial velocity vector from being expressed in the body frame (F_b) to being expressed in the vehicle carried NED frame (F_v)

$$\overline{V}^v = \begin{pmatrix} \dot{P}_N \\ \dot{P}_E \\ \dot{P}_D \end{pmatrix}$$

Typically, we are interested in altitude, h, not position down, $\dot{P}_D$, so

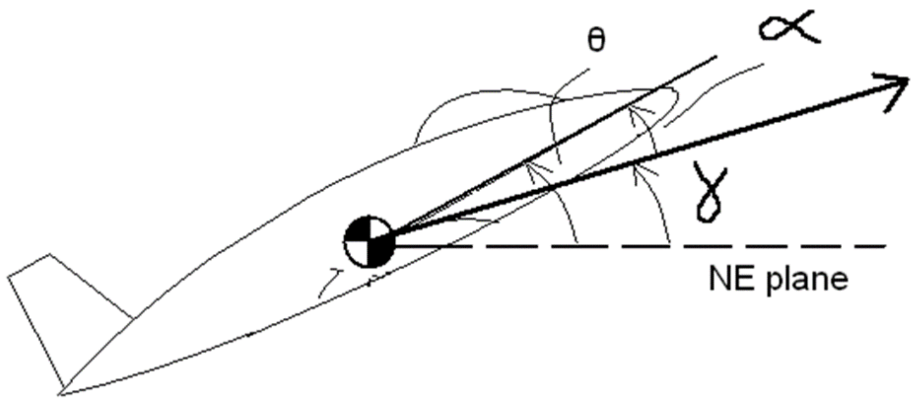
$$\dot{h} = -\dot{P}_D$$

Keeping track of position is therefore a simple matter of integrating $\overline{V}^v$ to obtain $\overline{P}^v$

$$\overline{P}^v(t) = \int_{t_o}^{t} \overline{V}^v(t)\, dt$$

With navigation equation, we can calculate some quantities exactly. For example, recall our previous expression for flight path angle (climb angle)

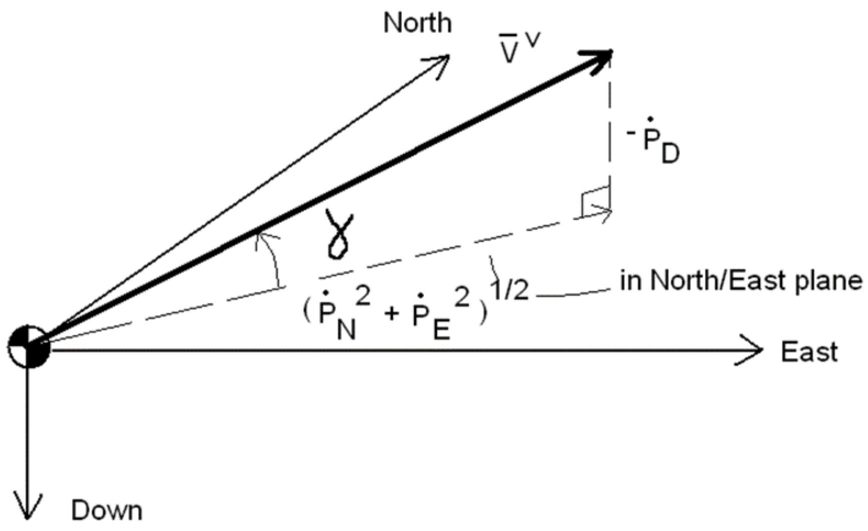
$$\gamma = \theta - \alpha = x_8 - \text{atan}(x_3 / x_1)$$



This is only valid for wings-level flight. A more robust definition of climb angle is

γ = angle that velocity vector $\overline{V}_{\text{CM/e}}$ makes with the NE plane (pg. 101)

We can easily use the navigation equations to get this definition



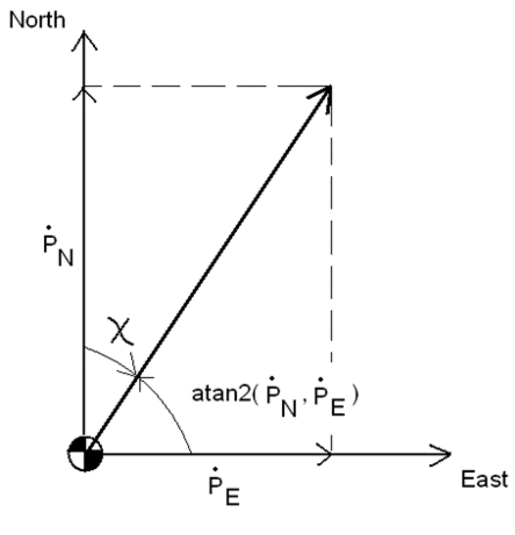
So we can write

$$\gamma = \text{atan2}\left(-\dot{P}_D, \left(\dot{P}_N^{\,2} + \dot{P}_E^{\,2}\right)^{1/2}\right)$$

Similarly, the rate of climb is trivial

$$\text{rate of climb} = \dot{h} = -\dot{P}_D$$

Similarly, the course angle is given by



So we can write

$$\chi = \text{MinimizeAngle}\left(\tfrac{\pi}{2} - \text{atan2}\left(\dot{P}_N, \dot{P}_E\right)\right)$$

where MinimizeAngle reduces an angle to its equivalent in the range of $[0, 2\pi)$

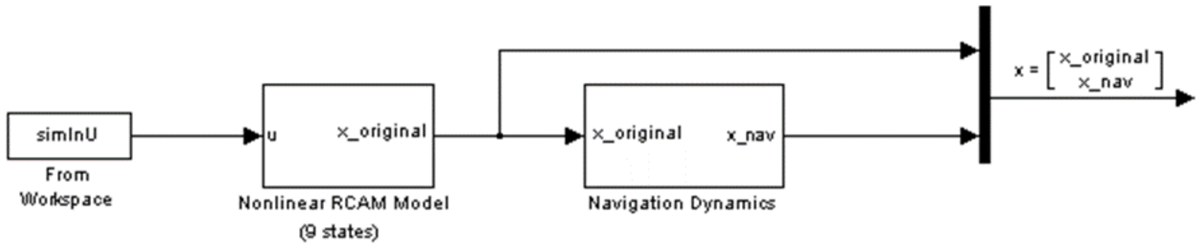
We immediately see that there is a significant difference between ψ (heading angle) and χ (course angle). The course angle is the direction of the center of mass of the vehicle, irrespective of its attitude.

For example, a skilled pilot can yaw the aircraft back and forth (ψ changes significantly) but keep the aircraft traveling in roughly the same direction (χ remains constant).

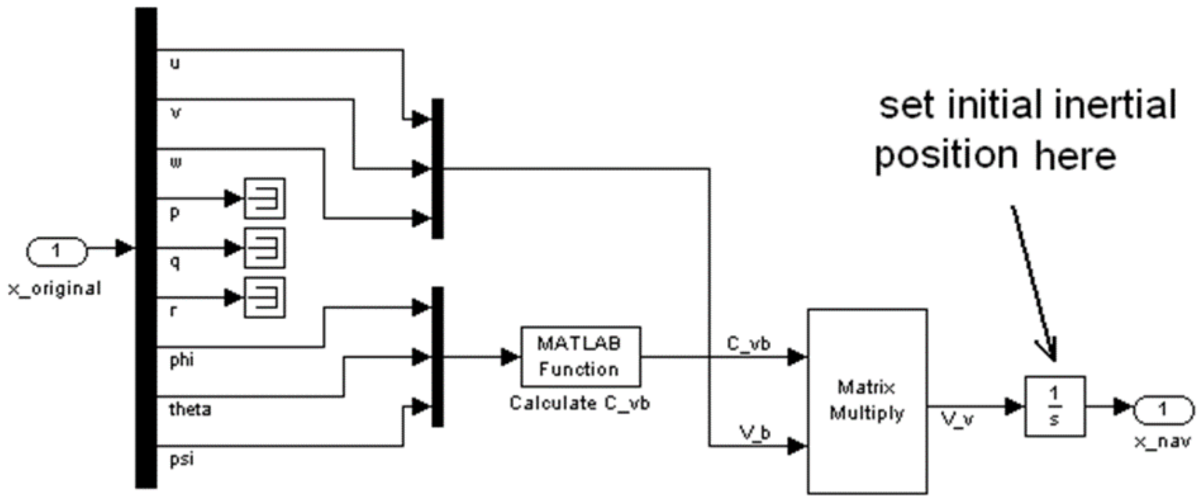
The navigation equations/dynamics can be implemented in several ways

Implementation Option 1: Separate Block

For our RCAM model, the dynamics are were not a function of position, so these can be implemented as a separate block.

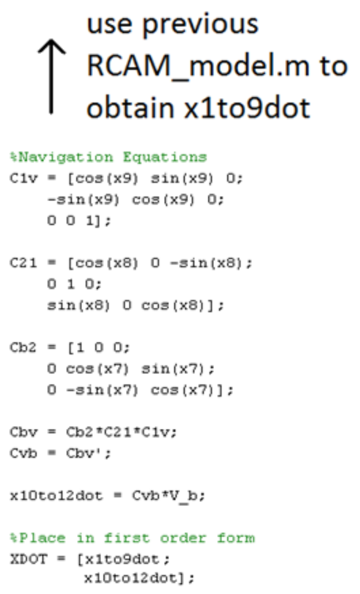


Where the Navigation Dynamics block simply looks like



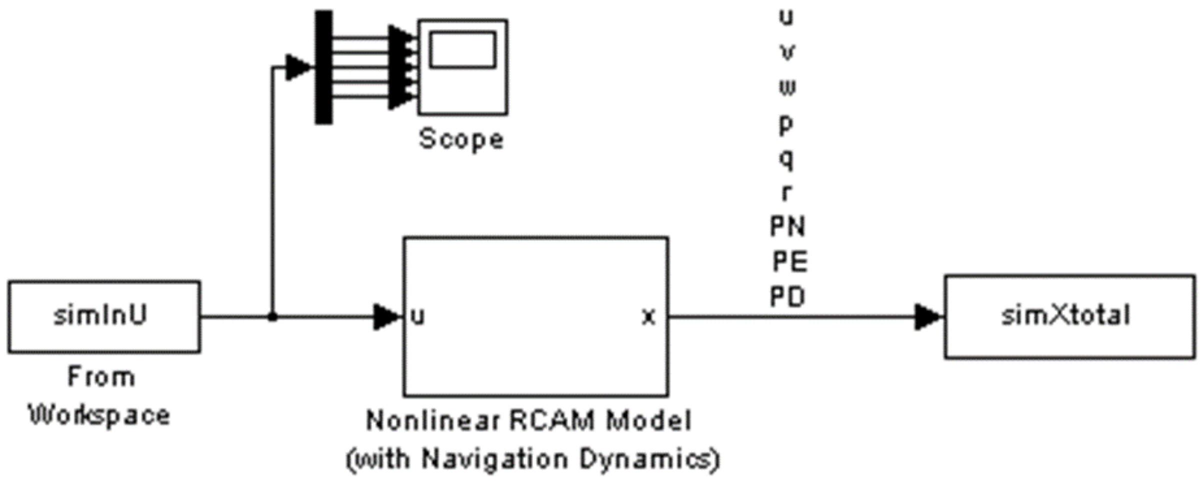
Implementation Option 2: Change RCAM_model.m function

Alternatively, you integrate these directly into your RCAM model by changing the last few lines of the RCAM_model.m function


use previous RCAM_model.m to obtain x1to9dot

```
%Navigation Equations
C1v = [cos(x9) sin(x9) 0;
    -sin(x9) cos(x9) 0;
    0 0 1];

C21 = [cos(x8) 0 -sin(x8);
    0 1 0;
    sin(x8) 0 cos(x8)];

Cb2 = [1 0 0;
    0 cos(x7) sin(x7);
    0 -sin(x7) cos(x7)];

Cbv = Cb2*C21*C1v;
Cvb = Cbv';

x10to12dot = Cvb*V_b;

%Place in first order form
XDOT = [x1to9dot;
        x10to12dot];
```

Now your Simulink model outputs



SHOW THAT AEROSPACE BLOCKSET EOMS HAS THIS EQUATION EMBEDDED

Altitude Models

Talk about how this can feed back into system (air density table lookups, etc.)

Effect on Trim Points

What about trim solutions? Previously, for steady state straight and level flight, we had said that we would like to impose the following constraints

1-9. $\dot{\overline{x}}_{\text{original}} = 0$ (steady state flight)
10. $V_a = 85\,m/s$ (airspeed)
11. $\gamma = 0$ (straight and level flight path)
12. $x_2 = v = 0$ (no sideslip)
13. $x_7 = \phi = 0$ (no bank angle)
14. $x_9 = \psi = 0$ (heading north)

where $\overline{x}_{\text{original}} = (\, u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi\,)^T = (\, x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9\,)^T$

Now our RCAM model has 12 states and 5 controls, so we have 17 degrees of freedom in our decision vector. Therefore we can specify 3 more soft constraints for our numerical optimization. This is simply given as

1-9. $\dot{\overline{x}}_{\text{original}} = 0$ (steady state flight)
10. $V_a = 85\,m/s$ (airspeed)
11. $\gamma = 0$ (straight and level flight path)
12. $x_2 = v = 0$ (no sideslip)
13. $x_7 = \phi = 0$ (no bank angle)
14. $x_9 = \psi = 0$ (heading north)
15. $x_{10} = P_{N_{\text{desired}}}$ (north position)
16. $x_{11} = P_{E_{\text{desired}}}$ (east position)
17. $x_{12} = P_{D_{\text{desired}}}$ (down position)

Cost function can them be simply calculated like we did previously. Note that you can no longer have $\dot{\overline{x}} = 0$ because $\dot{x}_{10}$, $\dot{x}_{11}$, and $\dot{x}_{12}$ cannot all be zero. This is because if they were all zero, then airspeed would be zero (assuming no wind). So in this situation, we are not finding an equilibrium point, but we are finding a trim point.