Christopher Lum
lum@uw.edu

Lecture 06a
A Nonlinear, 6 DOF Dynamic Model of an Aircraft: The Research Civil Aircraft Model (RCAM)

Lecture is on YouTube

The YouTube video entitled 'A Nonlinear, 6 DOF Dynamic Model of an Aircraft: The Research Civil Aircraft Model (RCAM)' that covers this lecture is located at https://youtu.be/bFFAL9lI2IQ.

Outline

-Introduction to the RCAM Model

-1. Control Limits/Saturation

-2. Intermediate Variables

-3. Nondimensional Aerodynamic Force Coefficients in F_s

-4. Aerodynamic Force in F_b

-5. Nondimensional Aerodynamic Moment Coefficient About Aerodynamic Center in F_b

-6. Aerodynamic Moment About Aerodynamic Center in F_b

-7. Aerodynamic Moment About Center of Gravity in F_b

-8. Propulsion Effects

-9. Gravity Effects

-10. Explicit First Order Form

  -Translations (x_1, x_2, x_3)

  -Rotations (x_4, x_5, x_6)

  -Euler Angles (x_7, x_8, x_9)

-Linear and Nonlinear ODEs

-Matlab Functions

Introduction to the RCAM Model

RCAM stands for "Research Civil Aircraft Model". This is an aircraft model developed by the Group for Aeronautical Research and Technology in Europe (GARTEUR) in the following document

Group for Aeronautical Research and Technology in Europe, “Robust Flight Control Design Challenge Problem Formulation and Manual: the Research Civil Aircraft Model (RCAM)”, GARTEUR/TP-0883. February 17, 1997.

We’ve posted a .pdf file of the original RCAM report. See pg. 6 - 20 (modeling of the plant).

This is a medium sized, two engine transport jet. Some specs

$m = 120,000$ kg (approx 264,000 lbm)

$\bar{c} = 6.6\, m$ (approx 22 ft)

$S = 260\, m^2$ (approx 2800 ft^2)

wingspan approx 40 m (approx 130 ft)

This is very similar to a Boeing 757-200.

State and control vector are given by

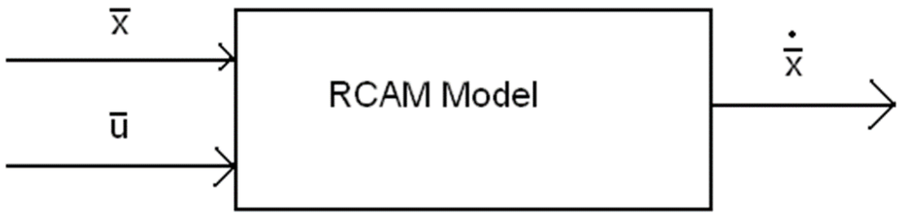
$$\bar{x} = \begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} = \begin{pmatrix} \text{velocity in body } x \text{ axis} \\ \text{velocity in body } y \text{ axis} \\ \text{velocity in body } z \text{ axis} \\ \text{angular rate about body } x \text{ axis} \\ \text{angular rate about body } y \text{ axis} \\ \text{angular rate about body } z \text{ axis} \\ \text{bank Euler angle} \\ \text{pitch Euler angle} \\ \text{yaw Euler angle} \end{pmatrix}$$

$$\bar{u} = \begin{pmatrix} \delta_A \\ \delta_T \\ \delta_R \\ \delta_{\text{th1}} \\ \delta_{\text{th2}} \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} \text{aileron} \\ \text{tail (elevator)} \\ \text{rudder} \\ \text{throttle 1} \\ \text{throttle 2} \end{pmatrix}$$

So at this point, we are in a position to develop the dynamic model of the aircraft. As we showed previously, a state space model of the system with this state and control vector can be given by

$$\dot{\bar{x}} = f(\bar{x}, \bar{u}) = \begin{pmatrix} \frac{1}{m}\bar{F}^b - \bar{\omega}_{b/e}{}^b \times \bar{V}^b \\ I_b{}^{-1}\left(\bar{M}^b - \bar{\omega}_{b/e}{}^b \times I_b\, \bar{\omega}_{b/e}{}^b\right) \\ H(\Phi)\, \bar{\omega}_{b/e}{}^b \end{pmatrix} \qquad \textbf{(Eq.4)}$$

The goal is to now create a function that computes these state derivatives.



With the control inputs defined, we are in a position to generate

$$\dot{\bar{x}} = f(\bar{x}, \bar{u})$$

Notice that this states that the state derivatives are only a function of the state and external inputs (control inputs). This is the same as all state space systems. We cannot have a model like

$$\dot{x}_1 = C_L + \ldots$$

Because C_L is a function of α

$$\dot{x}_1 = C_L(\alpha) + \ldots$$

Let's assume a simple model of linear, $C_L = n\,(\alpha - \alpha_{L=0})$

$$\dot{x}_1 = n\,(\alpha - \alpha_{L=0}) + \ldots$$

This is still not good enough because α is a function of $x_1 = u$, $x_2 = v$, $x_3 = w$. We know that $\alpha = \tan^{-1}(x_3 / x_1)$. We need to distill it down to

$$\dot{x}_1 = n\left(\tan^{-1}(x_3 / x_1) - \alpha_{L=0}\right)$$

As will be seen, this has the potential to generate some very complicated equations.

Clearly, the most difficulty in equations comes from the forces and moments. We will base our model off the Research Civil Aircraft Model (RCAM).

We develop the model in several steps

1. Control Limits/Saturation
2. Intermediate Variables
3. Nondimensional Aerodynamic Force Coefficients in F_s
4. Aerodynamic Force in F_b
5. Nondimensional Aerodynamic Moment Coefficient About Aerodynamic Center in F_b
6. Aerodynamic Moment About Aerodynamic Center in F_b
7. Aerodynamic Moment About Center of Gravity in F_b
8. Propulsion Effects
9. Gravity Effects
10. Explicit First Order Form

1. Control Limits/Saturation

Let's first talk about control inputs, $\overline{u}$ for this model. From page 20 of the RCAM document, we have

Numerical values for rate limits and saturations are given as follows.

- Rate limits for throttle movement are:
  rising slew rate $= 1.6\frac{\pi}{180}$ rad/s, falling slew rate $= -1.6\frac{\pi}{180}$ rad/s,
- throttle limits (saturations) are: $0.5\frac{\pi}{180}\ rad \leq \delta_{TH_i} \leq 10\frac{\pi}{180}\ rad.$

In case of engine failure we can assume that the throttle setting for the failed engine reduces to $\delta_{TH_i} = 0.5\frac{\pi}{180}$ rad with first order system dynamics given by the transfer function $1/(1+3.3s)$.

- Rate limits for aileron deflection are: $-25\frac{\pi}{180} \leq \dot{\delta}_A \leq 25\frac{\pi}{180}$ rad/s;
  saturations of aileron deflection are: $-25\frac{\pi}{180} \leq \delta_A \leq 25\frac{\pi}{180}$ rad,
- rate limits for tailplane deflection are: $-15\frac{\pi}{180} \leq \dot{\delta}_T \leq 15\frac{\pi}{180}$ rad/s;
  saturations of tailplane deflection are: $-25\frac{\pi}{180} \leq \delta_T \leq 10\frac{\pi}{180}$ rad,
- rate limits for rudder deflection are: $-25\frac{\pi}{180} \leq \dot{\delta}_R \leq 25\frac{\pi}{180}$ rad/s;
  saturations of rudder deflection are: $-30\frac{\pi}{180} \leq \delta_R \leq 30\frac{\pi}{180}$ rad.

For our model, we only focus on the saturation limits for these signals and ignore the rate limits. Effectively this means we are modeling an actuator that can move infinitely quickly but has absolute limits.

$$\overline{u} = \begin{pmatrix} \delta_A \\ \delta_S \\ \delta_R \\ \delta_{\text{th1}} \\ \delta_{\text{th2}} \end{pmatrix} = \begin{pmatrix} \text{aileron} \\ \text{horizontal stabilizer} \\ \text{rudder} \\ \text{throttle 1} \\ \text{throttle 2} \end{pmatrix} \qquad \begin{pmatrix} [-25, 25] \\ [-25, 10] \\ [-30, 30] \\ [0.5, 10] \\ [0.5, 10] \end{pmatrix} \frac{\pi}{180}$$

2. Intermediate Variables

We can calculate some intermediate variables which are functions of the states.

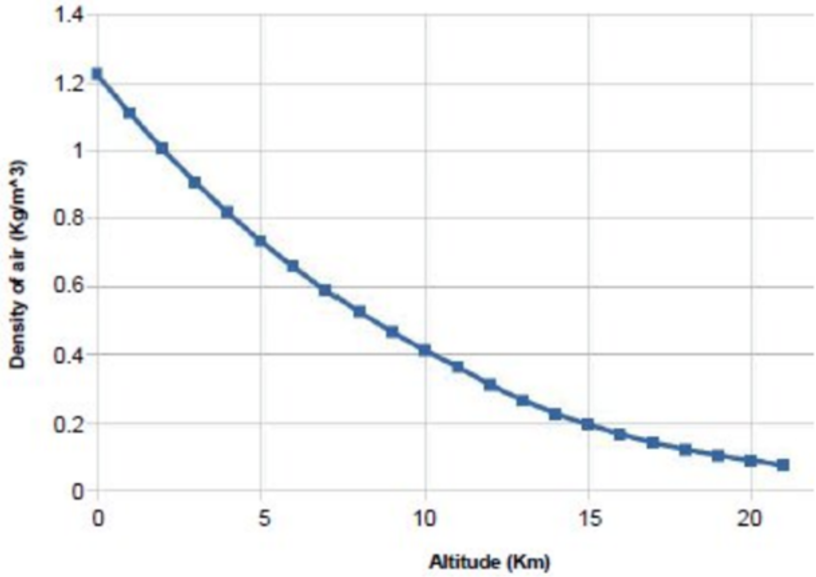
$$V_A = \left(x_1^{\,2} + x_2^{\,2} + x_3^{\,2}\right)^{1/2} \qquad \textbf{(Eq.2.16)}$$

$$\alpha = \tan^{-1}(x_3 / x_1) \qquad (\text{or atan2}(x_3,\, x_1)) \qquad \textbf{(Eq.2.17)}$$

$$\beta = \sin^{-1}(x_2 / V_a) \qquad \textbf{(Eq.2.18)}$$

$$Q = \tfrac{1}{2}\,\rho\, V_A^{\,2} \qquad (\text{dynamic pressure}) \qquad \textbf{(Eq.8)}$$

Note that we assume that ρ is a constant in our version of the RCAM model. A higher fidelity model would include the change in air density as shown below



We can also define $\overline{\omega}_{b/e}{}^{b}$ and $\overline{V}^{b}$, these were defined in Eq.4 and Eq.5 which are repeated here for convenience.

$$\overline{\omega}_{b/e}{}^{b} = \begin{pmatrix} p & q & r \end{pmatrix}^{T} = \begin{pmatrix} x_4 & x_5 & x_6 \end{pmatrix}^{T} \quad \textbf{(Eq.4)}$$

$$\overline{V}^{b} = \begin{pmatrix} u & v & w \end{pmatrix}^{T} = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}^{T} \quad \textbf{(Eq.5)}$$

Note that we use the notation Q to denote dynamic pressure and q to denote pitch rate.

3. Nondimensional Aerodynamic Force Coefficients in F_s

All of the following force coefficients are acting at the aerodynamic center and are expressed in F_s (see Section 2.3.4 of the RCAM document). However note that this appears to only be true for the force coefficients.

2.3.4 Aerodynamic equations

The equations defining aerodynamic forces and moments are determined by means of aerodynamic coefficients. Depending on the method of modelling these coefficients may be defined in different reference frames; e.g. F_W, F_S, or F_B. The reference frame for aerodynamic forces ~~and moments~~ that is used in RCAM is the stability axis frame F_S.

We now that the aerodynamic coefficients are also a function of the states. For small β values, we assume that the wing/body lift is only a function of α. In reality, this is not true and there is a significant dependence on β at higher yaw angles. However, for a first order approximation, we can neglect this. Note that later we will show that even in the wind axis, the lift coefficient does not have a β dependence in the RCAM model.

The lift coefficient of the wing/body in F_s is a piecewise function (for α < 19 deg) and is given by

$$C_{L_{wb}} = n(\alpha - \alpha_{L=0}) \qquad \text{if } \alpha \le 14.5 \tfrac{\pi}{180} \text{ rad} \qquad \textbf{(Eq.2.22)}$$
$$= a_3\,\alpha^3 + a_2\,\alpha^2 + a_1\,\alpha + a_0 \qquad \text{if } \alpha > 14.5 \tfrac{\pi}{180}$$

where $\alpha_{L=0} = -11.5 \frac{\pi}{180}$ rad (α at zero lift)

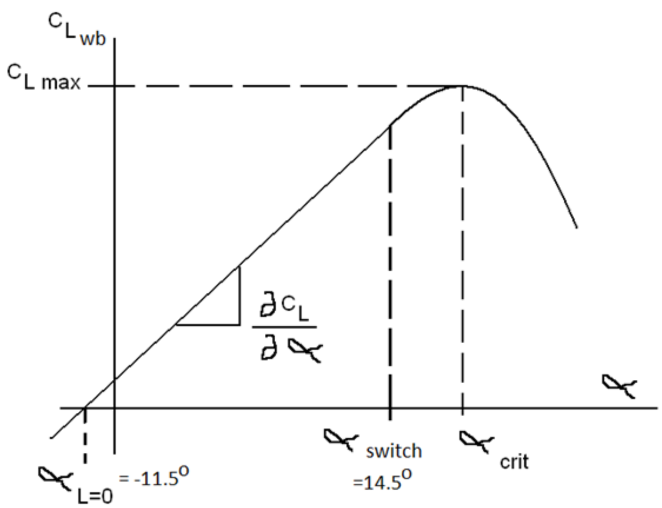
$n = 5.5$

$a_3 = -768.5$

$a_2 = 609.2$

$a_1 = -155.2$

$a_0 = 15.2$



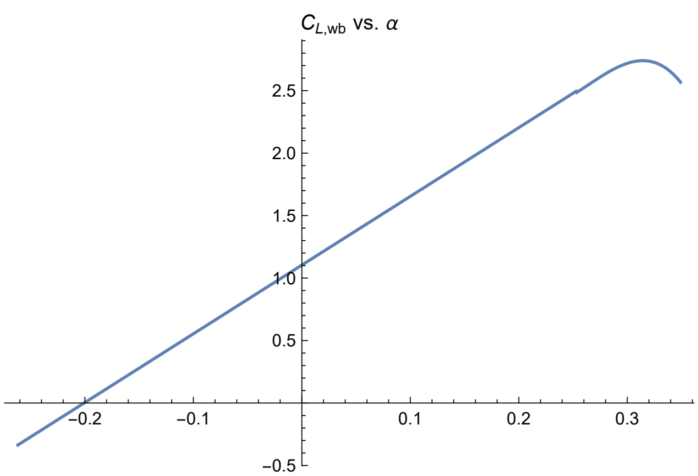
Note that this does not model extreme negative values of angle of attack (this does not include reverse stall conditions)

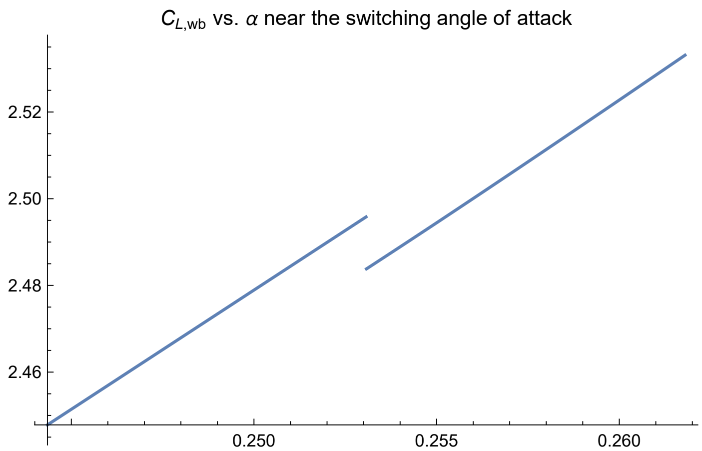
We can examine this function closely

```
(*Define the function*)
αswitch = 14.5 π/180;
αL0 = -11.5 π/180;
n = 5.5;
a3 = -768.5;
a2 = 609.2;
a1 = -155.2;
a0 = 15.2;
CLwb[α_] := If[α ≤ αswitch,
  n (α - αL0),
  a3 α^3 + a2 α^2 + a1 α + a0]

(*Plot this over a range of α values*)
Plot[CLwb[α], {α, -15 π/180, 20 π/180}, PlotRange → All, PlotLabel → "C_{L,wb} vs. α"]

(*Plot near the switching angle of attack*)
Plot[CLwb[α], {α, αswitch - 0.5 π/180, αswitch + 0.5 π/180},
 PlotRange → All, PlotLabel → "C_{L,wb} vs. α near the switching angle of attack"]
```





As we can see, this model for the lift curve is discontinuous near the switching angle of attack of $\alpha = 14.5^o$. This has some potential downstream ramifications. The easiest thing to do at this point would be to modify the lift curve to make the non-linear part exactly match the linear part. This can be achieved by modifying the a_0 parameter

$$C_{L_{\text{wb}}} = n(\alpha - \alpha_{L=0}) \qquad \text{if } \alpha \leq 14.5 \frac{\pi}{180} \text{ rad} \qquad \textbf{\textit{(Eq.2.22)}}$$
$$= a_3 \alpha^3 + a_2 \alpha^2 + a_1 \alpha + a_0 \qquad \text{if } \alpha > 14.5 \frac{\pi}{180}$$

```
temp = Solve[n (αswitch - αL0) == a3 αswitch^3 + a2 αswitch^2 + a1 αswitch + a0new, a0new];
a0new = a0new /. temp[[1]]
15.212
```

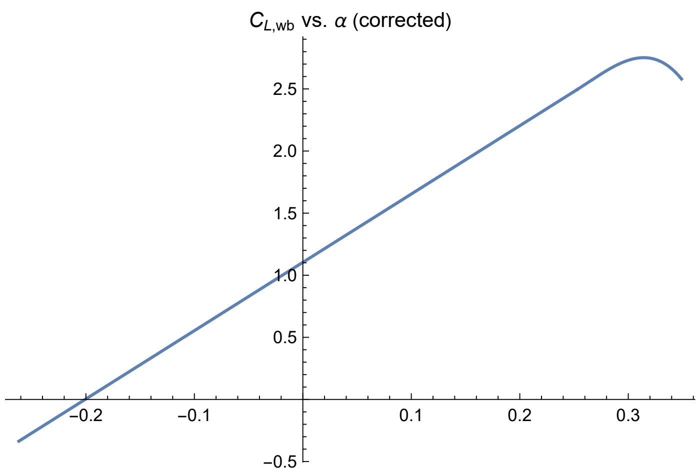
So we see that we should modify the RCAM document and instead of $a_0 = 15.2$, we should use

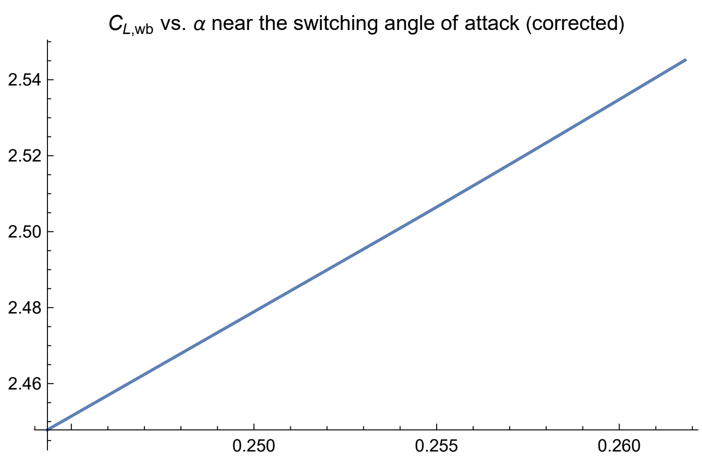
$$\alpha_0 = 15.212$$

```
CLwbNew[α_] := If[α ≤ αswitch,
  n (α - αL0),
  a3 α^3 + a2 α^2 + a1 α + a0new]

(*Plot this over a range of α values*)
Plot[CLwbNew[α], {α, -15 π/180, 20 π/180},
 PlotRange → All, PlotLabel → "C_{L,wb} vs. α (corrected)"]

(*Plot near the switching angle of attack*)
Plot[CLwbNew[α], {α, αswitch - 0.5 π/180, αswitch + 0.5 π/180}, PlotRange → All,
 PlotLabel → "C_{L,wb} vs. α near the switching angle of attack (corrected)"]
```

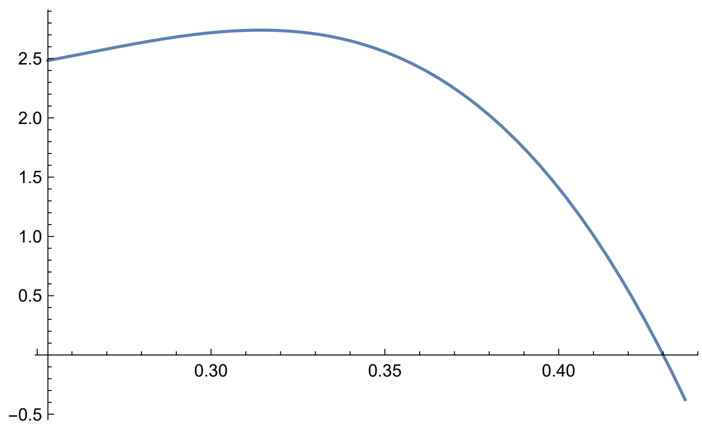




We can calculate the maximum C_L and α_{crit}

```
Plot[a3 α^3 + a2 α^2 + a1 α + a0, {α, 14.5 π/180, 25 π/180}]
temp = Solve[D[a3 α^3 + a2 α^2 + a1 α + a0, α] == 0, α];
αcrit1 = α /. temp[[1]];
αcrit2 = α /. temp[[2]];

αcrit1 180/π
αcrit2 180/π
```




```
12.2725

18.0069
```

So we see the maximum C_L is given as

```
CLwbNew[αcrit2]
2.75183
```

For future analysis, it may be useful to consider the linear portion of the C_L plot

```
CLwbLinear = n (α - αL0);
```

We know that the tail may also contribute a significant amount of lift. The lift from the tail is a function of several variables.

$$\alpha_t = \alpha - \varepsilon + u_2 + 1.3\, x_5\, l_t / V_A \qquad \textbf{(Eq.2.26)}$$

where $\varepsilon = \frac{\partial \varepsilon}{\partial \alpha}(\alpha - \alpha_{L=0})$ **(Eq.2.27)**

V_A = airspeed

```
lt = 24.8;
depsda = 0.25;
ε = depsda (α - αL0);

αt = α - ε + u2 + 1.3 x5 lt / VA
```

$$u2 + \frac{32.24\,x5}{VA} + \alpha - 0.25 \times (0.200713 + \alpha)$$

Since the horizontal tail is fully positionable (u_2), we see that this directly affects the angle of attack of the tail. Furthermore, note that there are some rate effects on the tail surface angle of attack ($x_5 = q$) appears.

So the final lift of the tail is given by

$$C_{L_t} = 3.1 \frac{S_t}{S} \alpha_t \qquad \textbf{(Eq.2.25)}$$

```
St = 64;
S = 260;
CLt = 3.1 St/S αt
```

$$0.763077 \left(u2 + \frac{32.24\,x5}{VA} + \alpha - 0.25 \times (0.200713 + \alpha)\right)$$

Total Lift, Drag, and Sideforce Coefficient in F_s

All of the following aerodynamic coefficients are presented in the stability axis (F_s) (see Section 2.3.4 of the RCAM document). This is somewhat strange since most wind tunnel data is obtained in the wind axis. However, the data could simply have been rotated before being presented to us. We will see that this has some benefits in modeling drag but can also obfuscate the dependence of the forces on the aerodynamic angles (particularly drag).

The total lift coefficient in F_s is given by

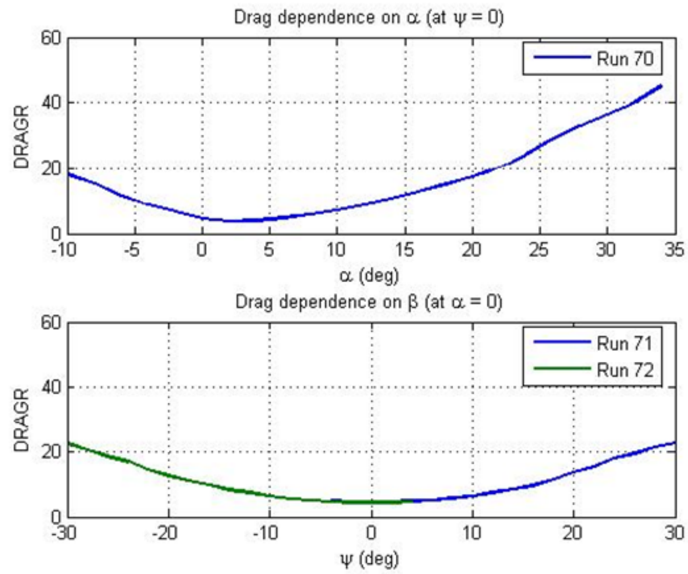
$$C_L = C_{L_{\text{wb}}} + C_{L_t} \qquad \textbf{(Eq.2.21)}$$

```
CL = CLwbLinear + CLt
```

$$5.5 \times (0.200713 + \alpha) + 0.763077 \left(u2 + \frac{32.24\,x5}{VA} + \alpha - 0.25 \times (0.200713 + \alpha)\right)$$

Note that in this case, the modelers were careful to normalize both $C_{L_{\text{wb}}}$ and C_{L_t} with the same area so we can directly add coefficients (note the S_t/S term in the C_{L_t} expression). In general, it is safer to add actual forces and moments (see YouTube video on ‘Manipulating Aerodynamic Coefficients’).

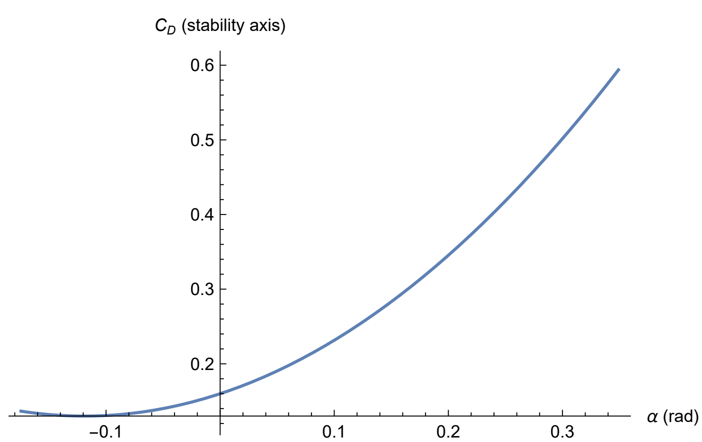
We now turn our attention to the drag coefficient. There are usually significant effects of drag due to angle of attack, α, and side slip angle, β. For example, below are plots of actual wind tunnel data examining the dependence of drag on α and β. Note that the magnitude of drag at non-zero side slip is of the same magnitude as the drag at non-zero angle of attack. Further note that this is raw drag which (if we neglect minor angularity and wall corrections) is expressed in the wind frame, F_w.



In the RCAM model,The total drag coefficient in F_s is given as

$$C_D = 0.13 + 0.07\,(5.5\,\alpha + 0.654)^2 \qquad \textbf{\textit{(Eq.2.29)}}$$

```
CD = 0.13 + 0.07 (5.5 α + 0.654)^2;
Plot[CD, {α, -10 π/180, 20 π/180}, AxesLabel → {"α (rad)", "C_D (stability axis)"}]
```



We can find the minimum drag

```
temp = Solve[D[CD, α] == 0, α];
αMinDrag = α /. temp[[1]];
αMinDrag 180/π
```

-6.81299

Upon initial inspection, one might balk at this model of drag as β does not appear explicitly in the expression. However, note that we expect drag in the wind frame to have a dependence on β. When this drag coefficient is rotated from the stability frame to the wind frame, a dependence on β will be introduced.

The total sideforce coefficient in F_s is given as

$$C_Y = -1.6\,\beta + 0.24\,u_3 \qquad \textbf{\textit{(Eq.2.30)}}$$

```
CY = -1.6 β + 0.24 u3;
```

As expected, the sideforce is affected by the rudder.

Optional: Rotate to Wind Axis

As a sanity check, we can rotate these coefficients to the wind axis

$$\overline{C}_F^{\,s} = \begin{pmatrix} -C_D \\ C_Y \\ -C_L \end{pmatrix}_s$$

```
CFs = (-CD
        CY
       -CL);
CFs // MatrixForm
```

$$\begin{pmatrix} -0.13 - 0.07\,(0.654 + 5.5\,\alpha)^2 \\ 0.24\,\text{u3} - 1.6\,\beta \\ -5.5 \times (0.200713 + \alpha) - 0.763077\left(\text{u2} + \frac{32.24\ \text{x5}}{\text{VA}} + \alpha - 0.25 \times (0.200713 + \alpha)\right) \end{pmatrix}$$

We can rotate this to the wind axis using

$$\overline{C}_F^{\,w} = C_{w/s}(\beta)$$

$$C_{w/s}(\beta) = \begin{pmatrix} \cos(\beta) & \sin(\beta) & 0 \\ -\sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
Cws[β_] = ( Cos[β]  Sin[β]  0
           -Sin[β]  Cos[β]  0
             0        0     1);
CFw = Cws[β].CFs;
CFw // MatrixForm
```

$$\begin{pmatrix} \left(-0.13 - 0.07\,(0.654 + 5.5\,\alpha)^2\right)\text{Cos}[\beta] + (0.24\,\text{u3} - 1.6\,\beta)\,\text{Sin}[\beta] \\ (0.24\,\text{u3} - 1.6\,\beta)\,\text{Cos}[\beta] - \left(-0.13 - 0.07\,(0.654 + 5.5\,\alpha)^2\right)\text{Sin}[\beta] \\ -5.5 \times (0.200713 + \alpha) - 0.763077\left(\text{u2} + \frac{32.24\ \text{x5}}{\text{VA}} + \alpha - 0.25 \times (0.200713 + \alpha)\right) \end{pmatrix}$$

So we see that the drag, sideforce, and lift coefficients in the wind axis are given as

```
Print["C_D^w"]
CDw = -CFw[[1, 1]]
Print[" "]

Print["C_Y^w"]
CYw = CFw[[2, 1]]
Print[" "]

Print["C_L^w"]
CLw = -CFw[[3, 1]]
Print[" "]
```

$C_D{}^w$

$$-\left(-0.13 - 0.07\ (0.654 + 5.5\,\alpha)^2\right) \mathrm{Cos}[\beta] - (0.24\,u3 - 1.6\,\beta)\ \mathrm{Sin}[\beta]$$

$C_Y{}^w$

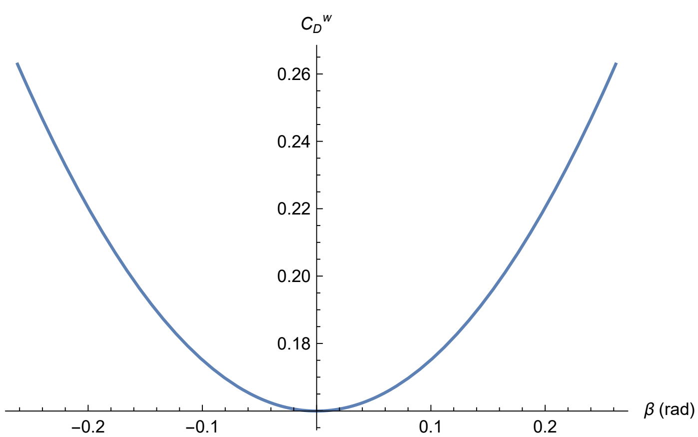
$$(0.24\,u3 - 1.6\,\beta)\ \mathrm{Cos}[\beta] - \left(-0.13 - 0.07\ (0.654 + 5.5\,\alpha)^2\right) \mathrm{Sin}[\beta]$$

$C_L{}^w$

$$5.5 \times (0.200713 + \alpha) + 0.763077 \left(u2 + \frac{32.24\,x5}{VA} + \alpha - 0.25 \times (0.200713 + \alpha)\right)$$

So we see that the drag coefficient in the wind axis indeed has some β dependence.

```
Plot[CDw /. {α → 0, u2 → 0, u3 → 0}, {β, -15 π/180, 15 π/180},
 AxesLabel → {"β (rad)", "C_D^w"}]
```



However, the lift coefficient still does not have any β dependence.

4. Aerodynamic Force in F_b

Therefore, the aerodynamic forces expressed in F_s (stability axis) are given by

$$\overline{F}_A{}^s = \begin{pmatrix} -D \\ Y \\ -L \end{pmatrix}_s = \begin{pmatrix} -C_D\,Q\,S \\ C_Y\,Q\,S \\ -C_L\,Q\,S \end{pmatrix} \qquad \textbf{\textit{(Eq.2.31-Eq.2.33)}}$$

We can rotate these forces to F_b using

$$\overline{F}_A{}^b = C_{b/s}\,\overline{F}_A{}^s \qquad \textbf{\textit{(Eq.2.34-Eq.2.36)}}$$

where $$C_{b/s} = \begin{pmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix}$$

5. Nondimensional Aerodynamic Moment Coefficient About Aerodynamic Center in F_b

The aerodynamic moment coefficients about the aerodynamic center in the report are given in F_b.
<u>NOTE THIS IS INCONSISTENT WITH PREVIOUS STATEMENT THAT THESE WERE IN STABILITY AXIS</u>

We can write

$$\overline{C}_{M_{ac}}{}^b = \begin{pmatrix} C_{l_{ac}} \\ C_{m_{ac}} \\ C_{n_{ac}} \end{pmatrix}_b = \overline{\eta} + \frac{\partial C_M}{\partial x}\,\overline{\omega}_{b/e}{}^b + \frac{\partial C_M}{\partial u}\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \qquad \textbf{\textit{(Eq.2.37)}}$$

where $$\overline{\eta} = \begin{pmatrix} -1.4\,\beta \\ -0.59 - 3.1\,\frac{S_t\,l_t}{S\,\overline{c}}\,(\alpha - \varepsilon) \\ \left(1 - \alpha\,\frac{180}{15\,\pi}\right)\beta \end{pmatrix} \qquad \frac{\partial C_M}{\partial x} = \frac{\overline{c}}{V_A}\begin{pmatrix} -11 & 0 & 5 \\ 0 & -4.03\,\frac{S_t\,l_t^2}{S\,\overline{c}^2} & 0 \\ 1.7 & 0 & -11.5 \end{pmatrix}$$

$$\frac{\partial C_M}{\partial u} = \begin{pmatrix} -0.6 & 0 & 0.22 \\ 0 & -3.1\,\frac{S_t\,l_t}{S\,\overline{c}} & 0 \\ 0 & 0 & -0.63 \end{pmatrix}$$

Note that the notation is somewhat abusive in the sense that $\frac{\partial C_M}{\partial x}$ is not really the derivatives of the moments with respect to the states, but rather it is how the moments are influenced by p, q, and r. Similarly, $\frac{\partial C_M}{\partial u}$ is not how the moments are affected by all the controls, but rather it measures how the moments are a function of the first three controls (the effects of the engines will be taken into account later when we deal with the propulsive forces and moments).

6. Aerodynamic Moment About Aerodynamic Center in F_b

We can convert this to an actual vector moment by renormalizing by $q\,S\,\bar{c}$. Notice that all these use a reference length of $\bar{c}$. In other applications, it may be possible to see b_{ref} be used for yaw and roll.

$$\overline{M}_{A_{\text{ac}}}{}^{b} = \overline{C}_{M_{\text{ac}}}{}^{b}\,q\,S\,\bar{c} \qquad \text{(Aerodynamic moments in } F_b \text{ about aerodynamic center)}$$

7. Aerodynamic Moment About Center of Gravity in F_b

We can now transfer these moments to the center of gravity. Note that the RCAM document appears to have a typo in their equation for moment transfer. Recall from a previous homework that the moment transfer equation is given as

$$\overline{M}_{A_{\text{cg}}}{}^{b} = \overline{M}_{A_{\text{ac}}}{}^{b} + \overline{F}_{A}{}^{b} \times \left(\bar{r}_{\text{cg}}{}^{b} - \bar{r}_{\text{ac}}{}^{b}\right)$$

For this aircraft, we will use

$$\bar{r}_{\text{cg}}{}^{b} = \begin{pmatrix} X_{\text{cg}} \\ Y_{\text{cg}} \\ Z_{\text{cg}} \end{pmatrix} = \begin{pmatrix} 0.23\,\bar{c} \\ 0 \\ 0.1\,\bar{c} \end{pmatrix} \qquad \text{(location of CG)}$$

$$\bar{r}_{\text{ac}}{}^{b} = \begin{pmatrix} X_{\text{ac}} \\ Y_{\text{ac}} \\ Z_{\text{ac}} \end{pmatrix} = \begin{pmatrix} 0.12\,\bar{c} \\ 0 \\ 0 \end{pmatrix} \qquad \text{(location of aerodynamic center)}$$

Parameters		Bounds					Nominal
m	MASS	$100\,000\ kg$	$<$	m	$<$	$150\,000\ kg$	$120\,000\ kg$
X_{cg}	XCG	$0.15\ \bar{c}$	$<$	X_{cg}	$<$	$0.31\ \bar{c}$	$0.23\ \bar{c}$
Y_{cg}	YCG	$-0.03\ \bar{c}$	$<$	Y_{cg}	$<$	$0.03\ \bar{c}$	$0.0\ \bar{c}$
Z_{cg}	ZCG	$0.00\ \bar{c}$	$<$	Z_{cg}	$<$	$0.21\ \bar{c}$	$0.10\ \bar{c}$

Table 2.5 Possible parameter choices in RCAM, see also section 3.2.3.

Furthermore, by inspecting Eq.2.38 in the RCAM document, it is clear that the aerodynamic moments that were modeled are actually in the body frame, not stability frame

FROM RCAM DOCUMENT

1. Only forces are rotated from stability to body axis

$$\begin{bmatrix} C_{l_{CG}} \\ C_{m_{CG}} \\ C_{n_{CG}} \end{bmatrix} = \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} + \frac{1}{\bar{c}} \begin{bmatrix} X_{cg} - 0.12\bar{c} \\ -Y_{cg} \\ Z_{cg} \end{bmatrix} \times \left(\overbrace{R_{BS} \cdot \begin{bmatrix} -C_D \\ C_Y \\ -C_L \end{bmatrix}} \right) \qquad (2.38)$$

$$\text{with} \quad R_{BS} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

2. These moments are therefore already in body axis

- Rolling moment in body axes

$$L_A = C_{l_{CG}} \, \frac{1}{2} \, \rho \, V_A{}^2 \, S \, \bar{c} \qquad (2.40)$$

- Pitching moment in body axes

$$M_A = C_{m_{CG}} \, \frac{1}{2} \, \rho \, V_A{}^2 \, S \, \bar{c} \qquad (2.41)$$

- Yawing moment in body axes

$$N_A = C_{n_{CG}} \, \frac{1}{2} \, \rho \, V_A{}^2 \, S \, \bar{c} \qquad (2.42)$$

3. Further validation that these moments are in the body axis

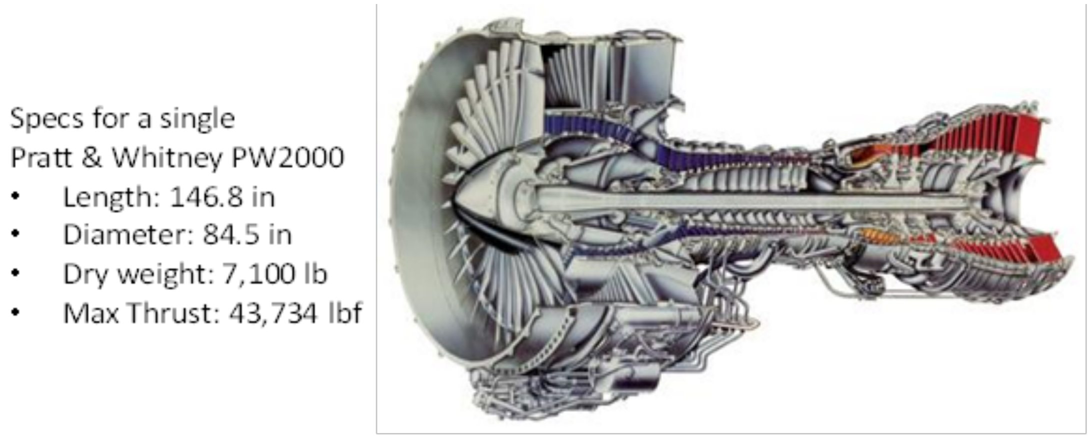
8. Propulsion Effects

Eq.2.43 states that the force from each engine is given by $F_i = \delta_{\text{th}_i} \, m \, g$. Therefore, the thrust of each engine is a force of magnitude

$F_1 = u_4 \, m \, g$ ***(Eq.2.43)***

$F_2 = u_5 \, m \, g$ ***(Eq.2.43)***

If both throttles are at maximum of $10\pi/180$, the aircraft has a thrust to weight ratio of approximately 0.35. Again, for comparison, a single Pratt & Whitney PW2000 engines produces about 42,000 lbf of thrust. The 757-200 has two of these. Combining with the MGTOW of 255,000 lbf yields a thrust to weight of 0.33.



$$\frac{10\,\frac{\pi}{180}\,m\,g * 2}{m\,g} \;//\; N$$

0.349066

$$\frac{2 * 42\,000}{255\,000} \;//\; N$$

0.329412

Assuming that these are aligned perfectly with F_b, the force vector in F_b due to the thrust for a given engine is given by

$$\overline{F}_{E_i}{}^{b} = \begin{pmatrix} F_i \\ 0 \\ 0 \end{pmatrix} \qquad \textbf{\textit{(Eq.9)}}$$

So the total engine force is

$$\overline{F}_{E}{}^{b} = \overline{F}_{E_1}{}^{b} + \overline{F}_{E_2}{}^{b} \qquad \textbf{\textit{(Eq.2.44)}}$$

In addition to the force, the engine thrust is not aligned with the CoG of the vehicle, so it causes a moment about the CoG. The torque due to a single engine in F_b about the CoG is given by

$$\overline{M}_{E_{\mathrm{cgi}}}{}^{b} = \overline{\mu}_i{}^{b} \times \overline{F}_{E_i}{}^{b} \qquad \textbf{\textit{(Eq.2.45)}}$$

where $\quad \mu_i{}^{b} = \begin{pmatrix} X_{\mathrm{cg}} - X_{\mathrm{APT}_i} \\ Y_{\mathrm{APT}_i} - Y_{\mathrm{cg}} \\ Z_{\mathrm{cg}} - Z_{\mathrm{APT}_i} \end{pmatrix}$

Engine Parameters					
X_{APT1}	XAPT1	=	x position of application point of thrust of engine 1 in F_M	0.0	m
Y_{APT1}	YAPT1	=	y position of application point of thrust of engine 1 in F_M	−7.94	m
Z_{APT1}	ZAPT1	=	z position of application point of thrust of engine 1 in F_M	−1.9	m
X_{APT2}	XAPT2	=	x position of application point of thrust of engine 2 in F_M	0.0	m
Y_{APT2}	YAPT2	=	y position of application point of thrust of engine 2 in F_M	7.94	m
Z_{APT2}	ZAPT2	=	z position of application point of thrust of engine 2 in F_M	−1.9	m

Table 2.4 Parameters definitions

Therefore, the total moment due to both engines is simply

$$\overline{M}_{E\,cg}{}^{b} = \overline{M}_{E_{cg1}}{}^{b} + \overline{M}_{E_{cg2}}{}^{b} \qquad \textbf{(Eq.10)}$$

9. Gravity Effects

We know that the gravity force is given by

$$\overline{F}_g = m\,\overline{g}$$

Typically, this is most easily expressed in F_e as

$$\overline{F}_g{}^{e} = m\,\overline{g}^{e} = m\begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \qquad \textbf{(Eq.11)}$$

We can use the Euler angles to rotate the gravity force from F_e to F_b. We have

$$\overline{F}_g{}^{b} = C_{b/2}\,C_{2/1}\,C_{1/e}\,\overline{F}_g{}^{e} \qquad \textbf{(Eq.12)}$$

Therefore, we can write

$$\overline{F}_g{}^{b} = m\begin{pmatrix} -g\sin(x_8) \\ g\cos(x_8)\sin(x_7) \\ g\cos(x_8)\cos(x_7) \end{pmatrix} \qquad \textbf{(Eq.13)}$$

10. Explicit First Order Form

We have now accounted for all the forces and moments. We now form an explicit expression $\dot{\overline{x}}$.

Translations (x_1, x_2, x_3)

We can start with expressions for $\dot{x}_1$, $\dot{x}_2$, and $\dot{x}_3$. Recall that this is given by Eq.2.1 which is repeated here for convenience.

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}_b = \frac{1}{m} \overline{F}^b - \overline{\omega}_{b/e}{}^b \times \overline{V}^b$$

where $\overline{F}^b = \overline{F}_g{}^b + \overline{F}_E{}^b + \overline{F}_A{}^b$

Rotations (x_4, x_5, x_6)

Next, the expression for $\dot{x}_4$, $\dot{x}_5$, and $\dot{x}_6$. Recall that these are given by Eq.6, which is repeated here for convenience.

$$\begin{pmatrix} \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix}_b = I_b{}^{-1}\left(\overline{M}_{cg}{}^b - \overline{\omega}_{b/e}{}^b \times I_b \, \overline{\omega}_{b/e}{}^b\right) \qquad \textbf{\textit{(Eq.6)}}$$

where $\overline{M}_{cg}{}^b = \overline{M}_{A_{cg}}{}^b + \overline{M}_{E_{cg}}{}^b$

Euler Angles (x_7, x_8, x_9)

Finally, the derivatives of the Euler angles are given by Eq.3.7, which is repeated here for convenience.

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}^b \qquad \textbf{\textit{(Eq.3.7)}}$$

We now have the equations of motion expressed in explicit first order form. Namely

$$\dot{\overline{x}} = f_e(\overline{x}, \overline{u})$$

For our system, we can simplify the expression further to a form which is called affine in control. This looks like

$$\dot{\overline{x}} = f(\overline{x}) + g(\overline{x})\,\overline{u}$$

Show how complicated these look like (PowerPoint presentation)

In general, it may not be feasible to express these in explicit form. A more general formulation of the equation of motion are

$$\overline{F}\left(\dot{\overline{x}}, \overline{x}, \overline{u}\right) = \overline{0} \qquad \text{(implicit form)}$$

In our scenario, we can write the explicit form as an implicit form using

$$\overline{F}\left(\dot{\overline{x}}, \overline{x}, \overline{u}\right) = \overline{f}(\overline{x}, \overline{u}) - \dot{\overline{x}} = 0$$

Linear and Nonlinear ODEs

Let us examine the form of linear and nonlinear ordinary differential equations

$$\dot{\overline{x}}(t) = A\,\overline{x}(t) + B\,\overline{u}(t) \qquad \text{(Linear ODE)} \qquad \textbf{\textit{(Eq.1.1)}}$$
$$\overline{y}(t) = C\,\overline{x}(t) + D\,\overline{u}(t)$$

For a nonlinear ODE

$$\dot{\overline{x}}(t) = \overline{f}(\,\overline{x}(t), \overline{u}(t)) \qquad \text{(Nonlinear explicit ODE)} \qquad \textbf{\textit{(Eq.1.2)}}$$
$$\overline{y}(t) = \overline{h}(\overline{x}(t), \overline{u}(t))$$

Note that these are multi-dimensional functions. In other words

$$\overline{f} : \mathbb{R}^{n+m} \to \mathbb{R}^{n}$$
$$\overline{h} : \mathbb{R}^{n+m} \to \mathbb{R}^{p}$$

where n = number of states
m = number of inputs
p = number of outputs

Or more generally

$$0 = \overline{f}\left(\dot{\overline{x}}(t),\ \overline{x}(t), \overline{u}(t)\right) \qquad \text{(Nonlinear implicit ODE)} \qquad \textbf{\textit{(Eq.1.3)}}$$
$$\overline{y}(t) = \overline{h}(\overline{x}(t), \overline{u}(t))$$

In this case, $\overline{f} : \mathbb{R}^{2n+m} \to \mathbb{R}^{n}$

In our case, the RCAM model falls into the non-linear explicit category. We will then use the non-linear model to derive a linear model.

Matlab Functions

With the control inputs defined, we are in a position to generate

$\dot{\bar{x}} = f(\bar{x}, \bar{u}) \quad \Longleftrightarrow$ [xdot] = RCAM_model(x, u);

From a software design standpoint, we would like several features

-allow for multiple number of input arguments
-be able to check that input arguments are reasonable
-compute desired outputs based on inputs

Example code template for a function called rand2 which returns a vector of uniformly distributed samples in the a specified range.

```
function [X] = rand2(varargin)

%RAND2  Generates a uniformly distributed random number in an interval
%
%   [X] = RAND2(LB,UB) Generates a random sample in the interval [LB,UB].
%   This makes use of the built in function RAND
%
%   [X] = RAND2(LB,UB,N) Generates N samples in the interval [LB,UB]
%   returned as a 1xN vector.
%
%
%INPUT:     -LB:    lower bound
%           -UB:    upper bound
%           -N:     number of samples (optional)
%
%OUTPUT:    -X:     vector of samples between lb and ub
%
%Created by Christopher Lum
%lum@uw.edu

%----------------------OBTAIN USER PREFERENCES----------------------------
switch nargin
    case 3
        %User supplies all inputs
        LB = varargin{1};
        UB = varargin{2};
        N  = varargin{3};

    case 2
        %Assume user only wants 1 sample
        LB = varargin{1};
        UB = varargin{2};
        N  = 1;

    otherwise
        error('Invalid number of inputs');
end


%-----------------------CHECKING DATA FORMAT------------------------------
[n_lb,m_lb] = size(LB);
[n_ub,m_ub] = size(UB);
[n_N,m_N] = size(N);


if (n_lb>1) || (m_lb>1)
    error('Lower bound must be a scalar')
end

if (n_ub>1) || (m_ub>1)
    error('Upper bound must be a scalar')
end

if (n_N>1) || (m_N>1)
    error('Number of samples must be a scalar')
end

if (LB>UB)
    error('Upper bound must be greater than or equal to lower bound')
end



%-------------------------BEGIN CALCULATIONS------------------------------
%Generate sample in range [0,1]
y = rand(1,N);

X = y*(UB-LB) + LB;
```

We will examine implementing the RCAM model in Matlab/Simulink in the next lecture.