Christopher Lum
lum@uw.edu

Lecture 08c
Numerically Linearizing a Dynamic System

The YouTube video entitled ‘Numerically Linearizing a Dynamic System’ that covers this lecture is located at https://youtu.be/1VmeijdM1qs.

Outline

-Implicit Form for Nonlinear ODEs
-Linearization Via Taylor Series
-Calculating Jacobian Matrices
  -Analytical Calculation
  -Numerical Approximation
    -Method 1 : Numerically Calculating Partial Derivatives
    -Method 2 : Matlab/Simulink ‘ Linear Analysis Tool’
    -Method 3 : Matlab/Simulink ‘ linmod’
-Different Linear Models for Different Flight Envelopes

Implicit Form for Nonlinear ODEs

Recall that most dynamic systems can be written in explicit form of

$$\dot{\bar{x}} = \bar{f}(\bar{x}, \bar{u}) \qquad \textbf{(Eq.1)}$$

where $\dot{\bar{x}} \in \mathbb{R}^n$
$\bar{x} \in \mathbb{R}^n$
$\bar{u} \in \mathbb{R}^m$

However, there may be times when it is not possible to explicitly solve for $\dot{\bar{x}}$. In this case, a more generic set of equations for a nonlinear dynamic system is

$$\bar{0} = \bar{F}\left(\dot{\bar{x}}, \bar{x}, \bar{u}\right) \qquad \textbf{\textit{(Eq.2)}}$$

$\bar{F} : \mathbb{R}^{2n+m} \to \mathbb{R}^n$

The benefit of this approach is we can apply it to implicit form scenarios (cases where we cannot solve explicitly for $\dot{\bar{x}}$).

We can define another vector as

$$\bar{\eta} = \begin{pmatrix} \dot{\bar{x}} \\ \bar{x} \\ \bar{u} \end{pmatrix} \in \mathbb{R}^{2n+m} \qquad \textbf{\textit{(Eq.3)}}$$

So we can rewrite Eq.2 as

$$\bar{0} = \bar{F}(\bar{\eta}) \qquad \textbf{\textit{(Eq.4)}}$$

We can view Eq. 4 as simply a function of $2n + m$ variables.

Linearization Via Taylor Series

Recall that we can use a Taylor Series Expansion to write $\bar{F}$.

We first specify a point, $\bar{\eta}$, about which to perform the Taylor Series expansion of $\bar{F}$ about. Note that this can be any point, $\bar{\eta}$, but only those that satisfies Eq.4 are consistent with the dynamics of the system.

Using somewhat abusive notation, we denote the point $\dot{\bar{x}}$, $\bar{x}$, and $\bar{u}$ which satisfies Eq.4 as $\bar{\eta}_o$.

$$\bar{\eta}_o = \begin{pmatrix} \dot{\bar{x}}_o \\ \bar{x}_o \\ \bar{u}_o \end{pmatrix} \qquad \textbf{\textit{(Eq.5)}}$$

One common point to use would be the equilibrium point (this is where $\dot{\bar{x}} = \bar{0}$). However, it is not necessary to pick this point as we can have any value of $\bar{\eta}_o$ as long $\bar{\eta}_o$ satisfies Eq.5.

Therefore, the Taylor Series expansion of the function $F\left(\dot{\bar{x}}, \bar{x}, \bar{u}\right)$ about the point η_o is given by

$$\bar{0} = \bar{F}(\bar{\eta}_o) + \frac{\partial \bar{F}(\bar{\eta})}{\partial \bar{\eta}} \bigg|_{\bar{\eta}=\bar{\eta}_o} (\bar{\eta} - \bar{\eta}_o) + H.O.T.$$

$$= \bar{F}\left(\dot{\bar{x}}_o, \bar{x}_o, \bar{u}_o\right) + \frac{\partial \bar{F}\left(\dot{\bar{x}}, \bar{x}, \bar{u}\right)}{\partial \dot{\bar{x}}} \bigg|_{\bar{\eta}=\bar{\eta}_o} \left(\dot{\bar{x}} - \dot{\bar{x}}_o\right) + \frac{\partial \bar{F}\left(\dot{\bar{x}}, \bar{x}, \bar{u}\right)}{\partial \bar{x}} \bigg|_{\bar{\eta}=\bar{\eta}_o} (\bar{x} - \bar{x}_o) + \frac{\partial \bar{F}\left(\dot{\bar{x}}, \bar{x}, \bar{u}\right)}{\partial \bar{u}} \bigg|_{\bar{\eta}=\bar{\eta}_o} (\bar{u} - \bar{u}_o) + H.O.T.$$

We know that since $\dot{\bar{x}}_o$, $\bar{x}_o$, and $\bar{u}_o$ satisfy Eq.1.5, then $F\left(\dot{\bar{x}}_o, \bar{x}_o, \bar{u}_o\right) = 0$

$$\bar{0} = \frac{\partial \bar{F}\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{\bar{x}}} \Big|_{\bar{\eta}=\bar{\eta}_o} \left(\dot{\bar{x}} - \dot{\bar{x}}_o\right) + \frac{\partial \bar{F}\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \bar{x}} \Big|_{\bar{\eta}=\bar{\eta}_o} \left(\bar{x} - \bar{x}_o\right) + \frac{\partial \bar{F}\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \bar{u}} \Big|_{\bar{\eta}=\bar{\eta}_o} \left(\bar{u} - \bar{u}_o\right) + H.O.T.$$

We can define the partials of the functions as matrices as shown below

$$\frac{\partial \bar{F}\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{\bar{x}}} \Big|_{\eta=\eta_o} = \begin{pmatrix} \frac{\partial F_1\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{x}_1} & \frac{\partial F_1\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{x}_2} & \cdots & \frac{\partial F_1\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{x}_n} \\ \frac{\partial F_2\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{x}_1} & \frac{\partial F_2\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{x}_2} & \cdots & \frac{\partial F_2\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{x}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{x}_1} & \frac{\partial F_n\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{x}_2} & \cdots & \frac{\partial F_n\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \dot{x}_n} \end{pmatrix} \Big|_{\bar{\eta}=\bar{\eta}_o} = E \qquad \textbf{(Eq.6)}$$

$$\frac{\partial \bar{F}\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \bar{x}} \Big|_{\eta=\eta_o} = \begin{pmatrix} \frac{\partial F_1\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial x_1} & \frac{\partial F_1\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial x_2} & \cdots & \frac{\partial F_1\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial x_n} \\ \frac{\partial F_2\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial x_1} & \frac{\partial F_2\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial x_2} & \cdots & \frac{\partial F_2\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial x_1} & \frac{\partial F_n\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial x_2} & \cdots & \frac{\partial F_n\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial x_n} \end{pmatrix} \Big|_{\bar{\eta}=\bar{\eta}_o} = A' \qquad \textbf{(Eq.7)}$$

$$\frac{\partial \bar{F}\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial \bar{u}} \Big|_{\eta=\eta_o} = \begin{pmatrix} \frac{\partial F_1\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial u_1} & \frac{\partial F_1\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial u_2} & \cdots & \frac{\partial F_1\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial u_m} \\ \frac{\partial F_2\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial u_1} & \frac{\partial F_2\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial u_2} & \cdots & \frac{\partial F_2\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial u_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial u_1} & \frac{\partial F_n\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial u_2} & \cdots & \frac{\partial F_n\left(\dot{\bar{x}},\bar{x},\bar{u}\right)}{\partial u_m} \end{pmatrix} \Big|_{\bar{\eta}=\bar{\eta}_o} = B' \qquad \textbf{(Eq.8)}$$

Therefore, we can rewrite the expansion (Eq.A.3) as

$$\bar{0} = E\left(\dot{\bar{x}} - \dot{\bar{x}}_o\right) + A'\left(\bar{x} - \bar{x}_o\right) + B'\left(\bar{u} - \bar{u}_o\right) + H.O.T. \qquad \textbf{(Eq.9)}$$

We can now define the perturbation from the expansion point as

$$\Delta\dot{\bar{x}} = \dot{\bar{x}} - \dot{\bar{x}}_o \qquad \textbf{(Eq.10.1)}$$
$$\Delta\bar{x} = \bar{x} - \bar{x}_o \qquad \textbf{(Eq.10.2)}$$
$$\Delta\bar{u} = \bar{u} - \bar{u}_o \qquad \textbf{(Eq.10.3)}$$

Eq.9 now becomes.

$$\bar{0} = E\,\Delta\dot{\bar{x}} + A'\,\Delta\bar{x} + B'\,\Delta\bar{u} + H.O.T.$$

Finally, dropping the higher order terms *(H.O.T.)* yields the linear approximation of the system and we have

$$\overline{0} \approx E\,\Delta\,\dot{\overline{x}} + A'\,\Delta\,\overline{x} + B'\,\Delta\,\overline{u}$$

$$-E\,\Delta\,\dot{\overline{x}} \approx A'\,\Delta\,\overline{x} + B'\,\Delta\,\overline{u}$$

$$\Delta\,\dot{\overline{x}} \approx A\,\Delta\,\overline{x} + B\,\Delta\,\overline{u} \qquad \textbf{\textit{(Eq.11)}}$$

where $A = -E^{-1} A'$
$B = -E^{-1} B'$

Calculating Jacobian Matrices

Analytical Calculation

If you can analytically calculate the all the partial derivatives in the Jacobian matrices (Eq.6 - Eq.8), you can directly obtain the E, A', and B' matrices

Numerical Approximation

Previously, we talked about how to obtain a trim point about which to linearize. We also investigated how to compute the Jacobian matrices by hand using analytical methods. If the function f is not analytical, it may not be possible to compute the partial derivatives by hand and therefore we need to consider numerical techniques. We will investigate several methods to do this.

1. Numerically calculating partial derivatives
2. The Matlab/Simulink ‘Linear Analysis Tool’
2. The Matlab/Simulink ‘linmod’ function

Method 1 : Numerically Calculating Partial Derivatives

We can now numerically linearize this system. The standard approach to linearize this system is to use the Jacobian method. This involves calculating many partial derivatives and can be done analytically.

However, this can be very difficult if the function to linearize is complicated. Worse, if the function does not have an analytical description (for example if it is a lookup table) we may not be able to calculate analytical derivatives. Therefore, we seek a method to numerically linearize the model. This involves numerically approximating the partial derivatives. We covered this in the lecture entitled ‘Numerically Calculating Partial Derivatives’ at https://youtu.be/G2gxvRjQHxc . In this video, we showed how to compute the gradient of a scalar function with n independent variables.

We can apply this technique to this scenario by realizing that the function $\overline{F}$ is simply n scalar equations, each with $2\,n + m$ independent variables.

$$\overline{F}(\overline{\eta}) = \begin{pmatrix} F_1(\overline{\eta}) \\ F_2(\overline{\eta}) \\ \dots \\ F_n(\overline{\eta}) \end{pmatrix}$$

where $\quad F_i : \mathcal{R}^{2n+m} \to \mathcal{R}^1$

We can extend this idea to multiple dimensions and can therefore numerically approximate elements of the E matrix as follows

$$E(i,\ j) = \frac{\partial F_i\left(\dot{\overline{x}}, \overline{x}, \overline{u}\right)}{\partial \dot{x}_j} \mid_{\eta=\eta_o} \approx \frac{F_i\left(\dot{\overline{x}}_{ij+}, \overline{x}_o, \overline{u}_o\right) - F_i\left(\dot{\overline{x}}_{ij-}, \overline{x}_o, \overline{u}_o\right)}{2\,\Delta\,\dot{x}_{ij}} \qquad (i \in [1,\ n],\ j \in [1,\ n]) \qquad \textbf{\textit{(Eq.B.2)}}$$

where $\quad \dot{\overline{x}}_{ij+} = \begin{pmatrix} \dot{x}_{1_o} & \dot{x}_{2_o} & \dots & \dot{x}_{j_o} + \Delta\,\dot{x}_{ij} & \dots & \dot{x}_{n_o} \end{pmatrix}^T$

$\qquad\qquad \dot{\overline{x}}_{ij-} = \begin{pmatrix} \dot{x}_{1_o} & \dot{x}_{2_o} & \dots & \dot{x}_{j_o} - \Delta\,\dot{x}_{ij} & \dots & \dot{x}_{n_o} \end{pmatrix}^T$

Similarly for the A' matrix

$$A'(i,\ j) = \frac{\partial F_i\left(\dot{\overline{x}}, \overline{x}, \overline{u}\right)}{\partial x_j} \mid_{\eta=\eta_o} \approx \frac{F_i\left(\dot{\overline{x}}_o, \overline{x}_{ij+}, \overline{u}_o\right) - F_i\left(\dot{\overline{x}}_o, \overline{x}_{ij-}, \overline{u}_o\right)}{2\,\Delta \mathrm{x}_{ij}} \qquad (i \in [1,\ n],\ j \in [1,\ n]) \qquad \textbf{\textit{(Eq.B.3)}}$$

where $\quad \overline{x}_{ij+} = \begin{pmatrix} x_{1_o} & x_{2_o} & \dots & x_{j_o} + \Delta \mathrm{x}_{ij} & \dots & x_{n_o} \end{pmatrix}^T$

$\qquad\qquad \overline{x}_{ij-} = \begin{pmatrix} x_{1_o} & x_{2_o} & \dots & x_{j_o} - \Delta \mathrm{x}_{ij} & \dots & x_{n_o} \end{pmatrix}^T$

And finally, for the B' matrix

$$B'(i,\ j) = \frac{\partial F_i\left(\dot{\overline{x}}, \overline{x}, \overline{u}\right)}{\partial u_{ij}} \mid_{\eta=\eta_o} \approx \frac{F_i\left(\dot{\overline{x}}_o, \overline{x}_o, \overline{u}_{ij+}\right) - F_i\left(\dot{\overline{x}}_o, \overline{x}_o, \overline{u}_{ij-}\right)}{2\,\Delta \mathrm{u}_{ij}} \qquad (i \in [1,\ n],\ j \in [1,\ m]) \qquad \textbf{\textit{(Eq.B.4)}}$$

where $\quad \overline{u}_{ij+} = \begin{pmatrix} u_{1_o} & u_{2_o} & \dots & u_{j_o} + \Delta \mathrm{u}_{ij} & \dots & u_{m_o} \end{pmatrix}^T$

$\qquad\qquad \overline{u}_{ij-} = \begin{pmatrix} u_{1_o} & u_{2_o} & \dots & u_{j_o} - \Delta \mathrm{u}_{ij} & \dots & u_{m_o} \end{pmatrix}^T$

Notice here that we can apply a different perturbation $\left(\Delta\,\dot{x}_{ij},\ \Delta \mathrm{x}_{ij},\ \Delta \mathrm{u}_{ij}\right)$ to each function. This is because each function may be differently sensitive in different directions.

This is why we use two indices to describe the perturbation magnitudes for each function. However in practice, it may be more useful to simply use a constant perturbation for all function and for all variables. In other words

$$\Delta\,\dot{x}_{ij} = \Delta \mathrm{x}_{ij} = \Delta \mathrm{u}_{ij} = \text{small number} \qquad \text{(easier for implementation)}$$

Show how to:

1. Create RCAM_model_implicit.m

2. Skeleton code how to generate ImplicitLinmod.m

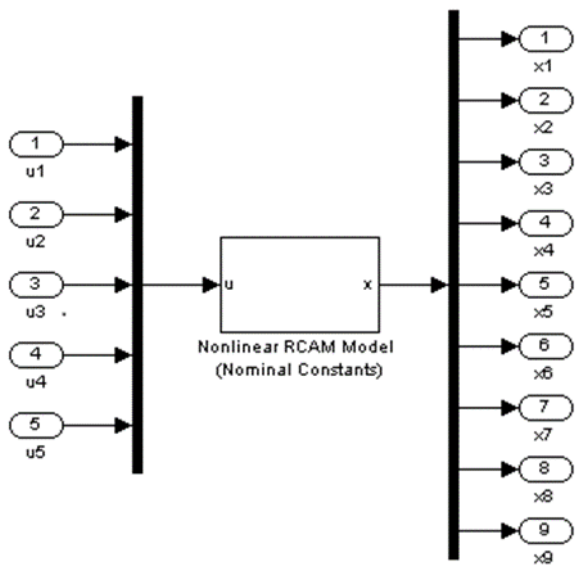
Method 2 : Matlab/Simulink ' Linear Analysis Tool'

See YouTube video ‘Linearizing a Simulink Model Using the Linear Analysis Tool and ‘linmod’’ (https://-youtu.be/M6FQfLmir0I)

Method 3 : Matlab/Simulink ' linmod'

See YouTube video ‘Linearizing a Simulink Model Using the Linear Analysis Tool and ‘linmod’’ (https://-youtu.be/M6FQfLmir0I)

To use the linmod function, we first build a plant model of the RCAM system similar to that shown below



We can then call

[A, B, C, D] = linmod(‘RCAM_model’, Xo, Uo);

To obtain the linearized model about the point $\bar{x}_o$ and $\bar{u}_o$.

Note: You may encounter issues if you specify the file extension of the model in the call to linmod

All methods should produce the similar results.

$$\Delta \dot{\bar{x}} \approx A \Delta \bar{x} + B \Delta \bar{u}$$

where $$A = \frac{\partial \bar{f}(\bar{x}_o, \bar{u}_o)}{\partial \bar{x}} = \frac{\partial \bar{f}(\bar{x}, \bar{u})}{\partial \bar{x}} \Big|_{\substack{\bar{x}=\bar{x}_o \\ \bar{u}=\bar{u}_o}}$$

$$B = \frac{\partial \bar{f}(\bar{x}_o, \bar{u}_o)}{\partial \bar{u}} = \frac{\partial \bar{f}(\bar{x}, \bar{u})}{\partial \bar{u}} \Big|_{\substack{\bar{x}=\bar{x}_o \\ \bar{u}=\bar{u}_o}}$$

Results for the RCAM system are shown below

```
A =

   -0.0354         0    0.0612         0   -1.2298         0         0   -9.8089
         0   -0.1805         0    1.2713         0  -84.9905    9.8089         0
   -0.2204         0   -0.7063         0   82.2155         0         0   -0.1467
         0   -0.0286         0   -1.3460         0    0.5842         0         0
   -0.0010         0   -0.0336         0   -1.1073         0         0         0
         0    0.0077         0    0.0554         0   -0.5533         0         0
         0         0         0    1.0000         0    0.0150         0         0
         0         0         0         0    1.0000         0         0         0
         0         0         0         0         0    1.0001         0         0


B =

         0    0.1094         0    9.8100    9.8100
         0         0    2.3012         0         0
         0   -7.3158         0         0         0
   -0.9486         0    0.3640    0.0407   -0.0407
         0   -2.9193         0    0.3924    0.3924
   -0.0199         0   -0.4081    0.7804   -0.7804
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
```

We can analytically verify some of the results. For example, if we examine rows 7-9 of the implicit non-linear ODEs, we have

$$\begin{pmatrix} F_7(\dot{\bar{x}}, \bar{x}, \bar{u}) \\ F_8(\dot{\bar{x}}, \bar{x}, \bar{u}) \\ F_9(\dot{\bar{x}}, \bar{x}, \bar{u}) \end{pmatrix} = H(\Phi)\, \bar{\omega}_{b/e}{}^{b} - \begin{pmatrix} \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{pmatrix}$$

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

$$= \begin{pmatrix} 1 & \sin(x_7)\tan(x_8) & \cos(x_7)\tan(x_8) \\ 0 & \cos(x_7) & -\sin(x_7) \\ 0 & \sin(x_7)/\cos(x_8) & \cos(x_7)/\cos(x_8) \end{pmatrix} \begin{pmatrix} x_4 \\ x_5 \\ x_6 \end{pmatrix}^{b} - \begin{pmatrix} \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{pmatrix}$$

So for example, to calculate the $A'(8, 1)$ entry, we compute

$$A'(8, 1) = \frac{\partial F_1(\dot{\bar{x}}_o, \bar{x}_o, \bar{u}_o)}{\partial x_1}$$

$$= \frac{\partial}{\partial x_1}\left[x_5 \cos(x_7) - x_6 \sin(x_7) - \dot{x}_7\right] \quad \text{evaluated at } \dot{\bar{x}} = \dot{\bar{x}}_o,\ \bar{x} = \bar{x}_o \text{ and } \bar{u} = \bar{u}_o$$

$$= 0$$

In fact, we see that the bottom 3 rows of the A' matrix should have a lot of zero elements. In fact, only columns 4, 5, 6, 7, and 8 should have non-zero elements.

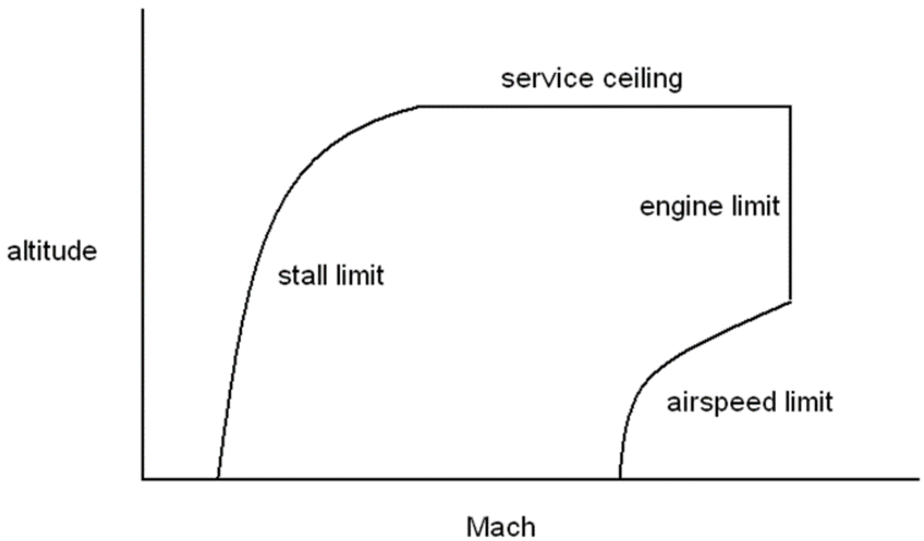
For the B' matrix, all three of the bottom rows should all be zero ($F_7 - F_9$ are not functions of $\overline{u}$)

Different Linear Models for Different Flight Envelopes

The non-linear model that we have developed is valid for almost all flight regimes (there were some restrictions like $\alpha < 14.5$ degress, etc). However, we cannot make much progress with analysis of the non-linear model so we want to linearize.

So using a combination of both the numerical optimization (to find trim points) and the numerical linearization (to generate linear models), we can perform this procedure to obtain different linear models for different flight envelopes.

Can perform gain scheduling based on different operating regimes (textbook pg. 259)



The problem is we need to obtain a lot of different linear models and then schedule the gains according to where in the flight envelope we are operating.