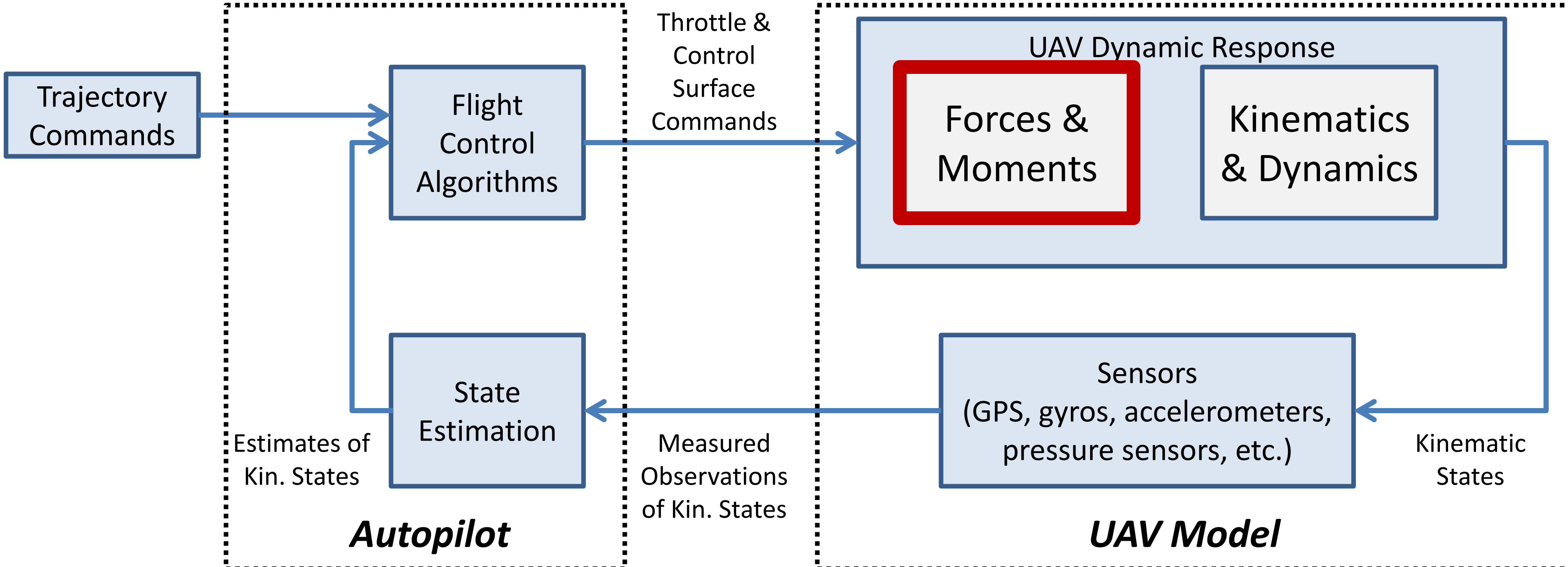
UAV Systems & Control
Lecture 5

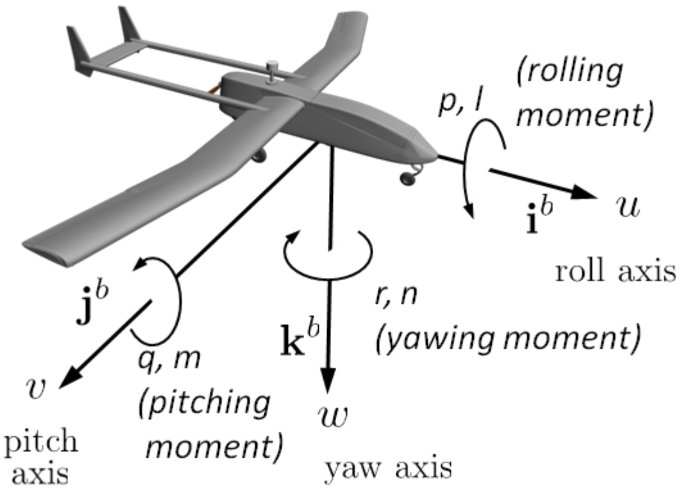
Propeller Forces and Moments
Scalar Equations of Motion
Trim
Linearization
Aircraft Response Modes

UAV System



- In the previous lecture we
  - developed the 12 equations of motion
  - developed models for forces and moments due to gravity and aerodynamics
- In this section we will develop equations for the *forces* and *moments* due to the propeller
- We will also discuss trimming and linearization (steps toward developing an autopilot)

Aircraft Variables



12 State Variables

Name	Description	Metric Units
p_n	Inertial North position of MAV expressed along $\mathbf{i}^i$ in $\mathcal{F}^i$.	m
p_e	Inertial East position of MAV expressed along $\mathbf{j}^i$ in $\mathcal{F}^i$.	m
p_d	Inertial Down position of MAV expressed along $\mathbf{k}^i$ in $\mathcal{F}^i$.	m
u	Ground velocity expressed along $\mathbf{i}^b$ in $\mathcal{F}^b$.	m/s
v	Ground velocity expressed along $\mathbf{j}^b$ in $\mathcal{F}^b$.	m/s
w	Ground velocity expressed along $\mathbf{k}^b$ in $\mathcal{F}^b$.	m/s
ϕ	Roll angle defined with respect to $\mathcal{F}^{v2}$.	rad
θ	Pitch angle defined with respect to $\mathcal{F}^{v1}$.	rad
ψ	Heading (yaw) angle defined with respect to $\mathcal{F}^v$.	rad
p	Body angular (roll) rate expressed along $\mathbf{i}^b$ in $\mathcal{F}^b$.	rad/s
q	Body angular (pitch) rate expressed along $\mathbf{j}^b$ in $\mathcal{F}^b$.	rad/s
r	Body angular (yaw) rate expressed along $\mathbf{k}^b$ in $\mathcal{F}^b$.	rad/s

Other Variables

mass	Vehicle mass, assumed constant	kg
J	3x3 Inertia matrix (Common simplifying assumption: $J_{xy}=J_{yz}=0$)	kg-m^2
f_x	Axial force along x-axis (e.g. majority of thrust and drag components)	N
f_y	Lateral force along y-axis (e.g. sideslip-induced force)	N
f_z	Normal force along z-axis (e.g. majority of lift and gravity compnts.)	N
l	Rolling moment, about x-axis	N-m
m	Pitching moment, about y-axis	N-m
n	Yawing moment, about z-axis	N-m

Vector relationships:

$$\underline{v}_g^b = u_g \hat{i}^b + v_g \hat{j}^b + w_g \hat{k}^b$$

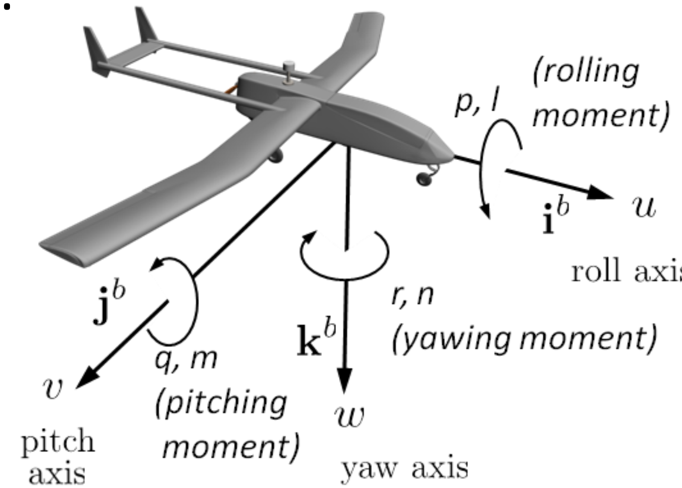
$$\underline{\omega}_{b/i}^b = p\,\hat{i}^b + q\,\hat{j}^b + r\,\hat{k}^b$$

$$\underline{\mathrm{f}}^b = f_x \hat{i}^b + f_y \hat{j}^b + f_z \hat{k}^b$$

$$\underline{\mathrm{m}}^b = l\,\hat{i}^b + m\,\hat{j}^b + n\,\hat{k}^b$$

Equations of Motion

- From Kinematics and Dynamics, the equations of motion are 12 ODEs:



$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} = R_b^{ned} \underline{v}_g^b = \left(R_{ned}^b\right)^T \underline{v}_g^b$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = -\underline{\omega}_{b/i}^b \times \underline{v}_g^b + \frac{1}{mass} \underline{\mathrm{f}}^b$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

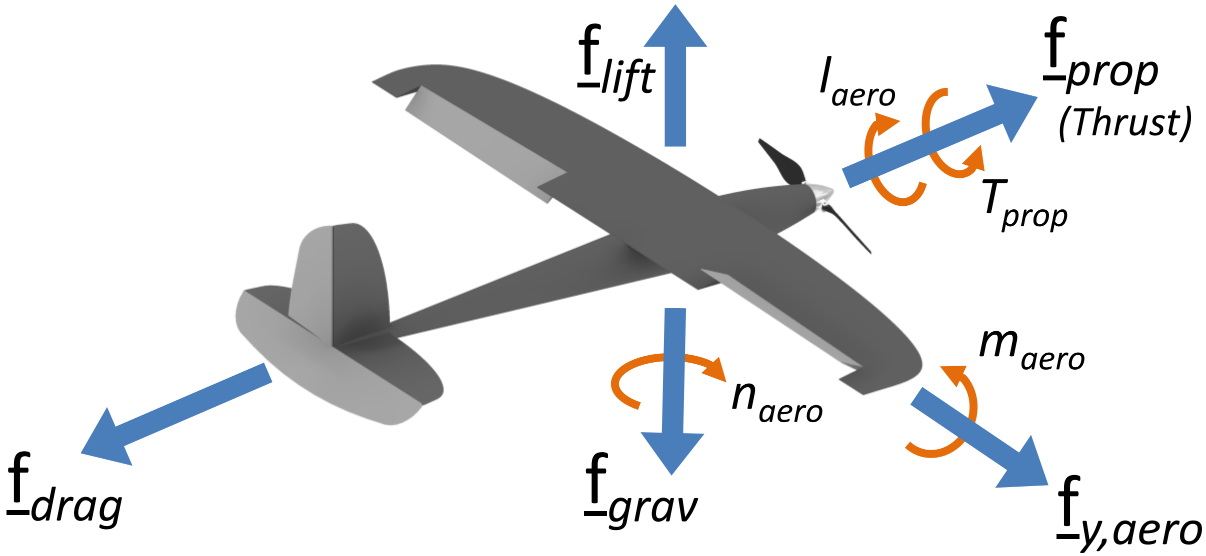
$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathrm{J}^{-1} \left[-\underline{\omega}_{b/i}^b \times \left(\mathrm{J}\, \underline{\omega}_{b/i}^b\right) + \underline{\mathrm{m}}^b \right]$$

The objective of this lecture is to show how to compute the force and moment vectors:

$$\underline{\mathrm{f}}^b = f_x \hat{i}^b + f_y \hat{j}^b + f_z \hat{k}^b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$$

$$\underline{\mathrm{m}}^b = l\, \hat{i}^b + m\, \hat{j}^b + n\, \hat{k}^b = \begin{bmatrix} l \\ m \\ n \end{bmatrix}$$

External Forces and Moments



Note: Modify $\underline{f}_{prop}$ and $\underline{m}_{prop}$ accordingly if propulsion/thrust source is not mounted along x-axis.

Sum of Forces:

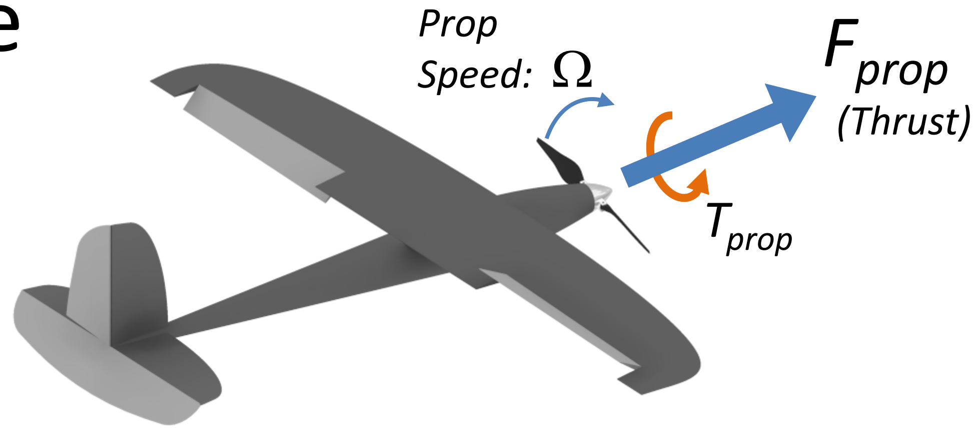
$$\Sigma\underline{f} = \underline{f}_{grav} + \underbrace{\underline{f}_{aero}}_{\underline{f}_{lift} + \underline{f}_{drag} + \underline{f}_{y,aero}} + \underline{f}_{prop}$$

Sum of Moments:

$$\Sigma\underline{m} = \underbrace{\underline{m}_{aero}}_{\begin{bmatrix} l_{aero} \\ m_{aero} \\ n_{aero} \end{bmatrix}^b} + \underbrace{\underline{m}_{prop}}_{\begin{bmatrix} T_{prop} \\ 0 \\ 0 \end{bmatrix}^b}$$

For “trim” flight, forces and moments are balanced.

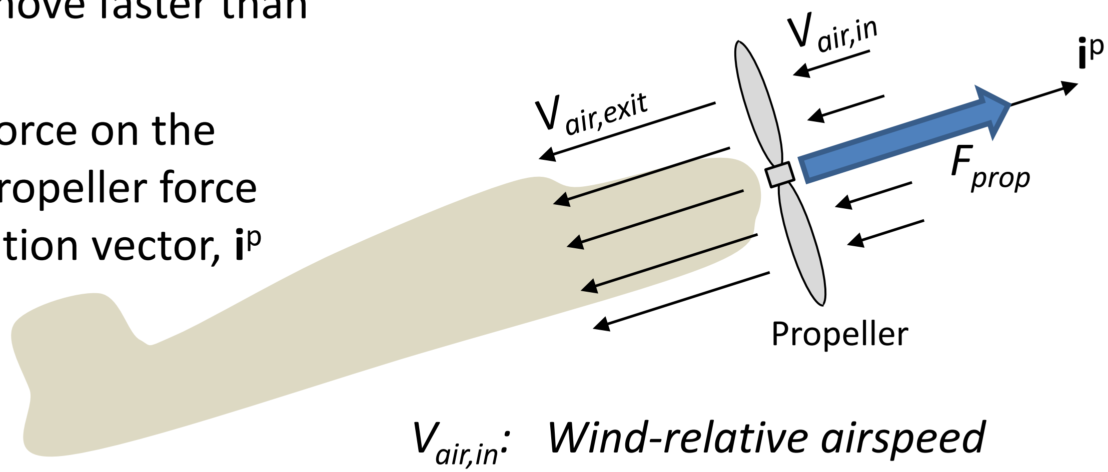
Propeller Force and Moment



- For a propeller-powered UAV, propeller provides thrust force
  - Propeller rotation speed is proportional to throttle setting, $0 < \delta_t < 1$
  - Faster propeller rotation causes an increased thrust force
  - Rotation of propeller causes a counter-torque (moment) on airframe
- Book computes propeller thrust force incorrectly (Sec. 4.3.1)
  - Derived method has negative thrust at zero throttle: $F_{prop} = \frac{1}{2}\rho C_{prop} S_{prop}\left(\left(k_{motor}\delta_t\right)^2 - V_a^2\right)$
  - Book website provides the corrected method described on following slides
    - $\Rightarrow F_{prop} = \rho C_{prop} S_{prop}\left(V_a + \delta_t\left(k_{motor} - V_a\right)\right)\left(\delta_t\left(k_{motor} - V_a\right)\right)$
    - See: http://uavbook.byu.edu/doku.php?id=shared:supplemental_material
    - Also: P. Fitzpatrick, "Calculation of thrust in a ducted fan assembly for hovercraft," tech. rep., Hovercraft Club of Great Britain, 2003.

Force (Thrust) due to Propeller

- Propellers transfer energy to air molecules, causing the discharged air molecules to move faster than the intake air molecules
- Air molecules impart a reaction force on the propeller blades, yielding a net propeller force (F_{prop}) along the propeller orientation vector, $\mathbf{i}^p$



$$F_{prop} = \rho \cdot Q_d \cdot \left(V_{air,exit} - V_{air,in}\right)$$

Q_d: Quantity (volume/s) of air expelled, m³/s

ρ: Air density, kg/m³

$V_{air,in}$: Wind-relative airspeed of input air, m/s

$V_{air,exit}$: Wind-relative airspeed of expelled air, m/s

- Quantity of air being discharged (m³/s) by propeller is given by:

$$Q_d = C_{prop} \cdot S_{prop} \cdot V_{air,exit}$$

C_{prop}: Unit-less propeller efficiency coefficient

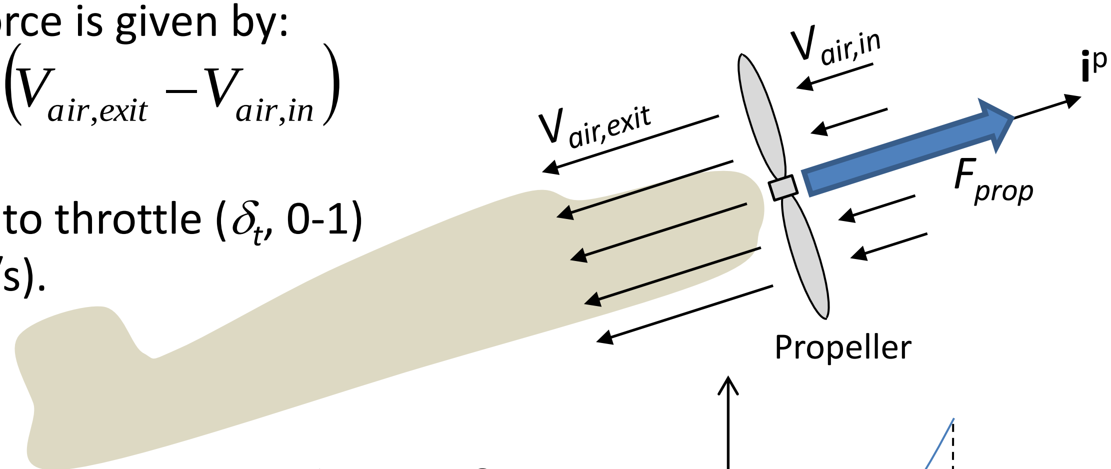
S_{prop}: Area swept out by propeller, m²

$$F_{prop} = \rho \cdot C_{prop} S_{prop} V_{air,exit} \cdot \left(V_{air,exit} - V_{air,in}\right)$$

Force (Thrust) due to Propeller

- From previous, net propeller force is given by:

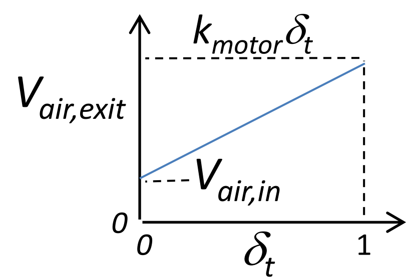
$$F_{prop} = \rho C_{prop} S_{prop} V_{air,exit} \left(V_{air,exit} - V_{air,in} \right)$$

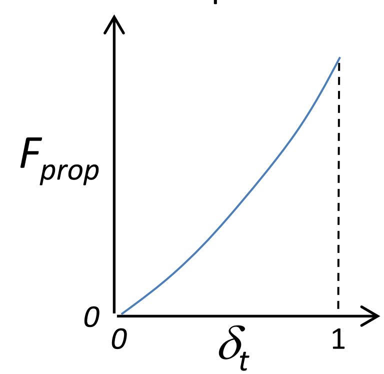


- Exit airspeed is directly related to throttle (δ_t, 0-1) via a motor constant (k_{motor}, m/s).

  At $\delta_t = 0$, $V_{air,exit} = V_{air,in}$
  At $\delta_t = 1$, $V_{air,exit} \approx k_{motor}\delta_t$

  Note: k_{motor} *can be found by measuring exit air speed at full throttle*





- Interpolating:

$$V_{air,exit} \approx V_{air,in} + \delta_t \left(k_{motor} - V_{air,in} \right)$$

$$\Rightarrow \quad F_{prop} = \rho C_{prop} S_{prop} \left(V_{air,in} + \delta_t \left(k_{motor} - V_{air,in} \right) \right) \left(\delta_t \left(k_{motor} - V_{air,in} \right) \right)$$

C_{prop}, S_{prop}, and k_{motor} define propeller thrust performance

- Input airflow ($V_{air,in}$) is the portion of the aircraft's wind-relative velocity vector ($\underline{v}_a$) that is along the propeller orientation unit vector ($\mathbf{i}^p$): $V_{air,in} = \underline{v}_a \cdot \mathrm{i}^p$
  - For a fixed-wing aircraft, $V_{air,in} \approx V_a$ (V_a: airspeed, m/s)

Moment due to Propeller

- A spinning propeller causes a counter-torque (moment) on the airframe

$$T_{prop} = -k_{Tp}\Omega^2$$

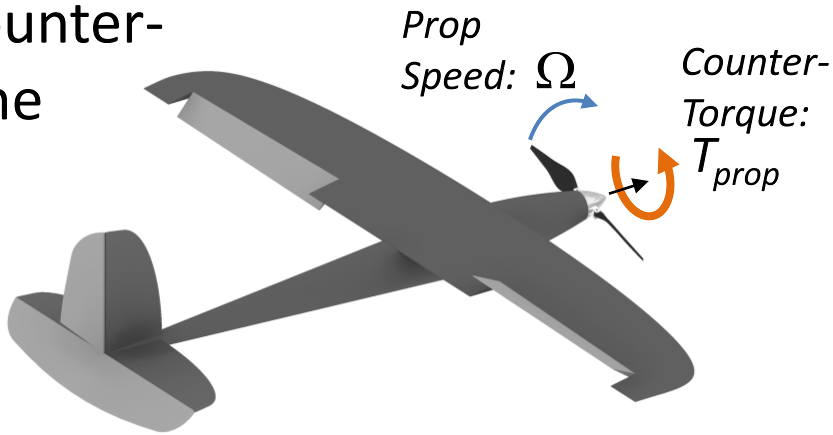
- Propeller speed is proportional to throttle

$$\Omega = k_{\Omega}\delta_t$$

- Thus, propeller counter-torque is:

$$T_{prop} = -k_{Tp}\left(k_{\Omega}\delta_t\right)^2$$

k_{Tp} and k_{Ω} can be determined via experiment



T_{prop}: *Propeller counter-torque, N-m*
Ω: *Propeller rotation speed, rad/s*

k_{Tp} : *Propeller torque constant, kg-m²*
k_{Ω} : *Propeller speed constant, rad/s*

δ_t: *Throttle level (0-1)*

Total Fixed Wing Forces & Moments

$$\underline{\mathrm{f}}^b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \underbrace{(mass)\mathrm{g}\begin{bmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{bmatrix}}_{\underline{\mathrm{f}}^b_{grav}} \underbrace{+\tfrac{1}{2}\rho V_a^2 S\begin{bmatrix} -C_D(\ldots)\cos(\alpha)+C_L(\ldots)\sin(\alpha) \\ C_Y(\ldots) \\ -C_D(\ldots)\sin(\alpha)-C_L(\ldots)\cos(\alpha) \end{bmatrix}}_{\underline{\mathrm{f}}^b_{aero}} + \underbrace{\begin{bmatrix} F_{prop} \\ 0 \\ 0 \end{bmatrix}}_{\underline{\mathrm{f}}^b_{prop}}$$

$$\underline{\mathrm{m}}^b = \begin{bmatrix} l \\ m \\ n \end{bmatrix} = \underbrace{\tfrac{1}{2}\rho V_a^2 S\begin{bmatrix} bC_l(\ldots) \\ cC_m(\ldots) \\ bC_n(\ldots) \end{bmatrix}}_{\underline{\mathrm{m}}^b_{aero}} + \underbrace{\begin{bmatrix} T_{prop} \\ 0 \\ 0 \end{bmatrix}}_{\underline{\mathrm{m}}^b_{prop}}$$

$$F_{prop} = \rho C_{prop} S_{prop}\left(V_a + \delta_t\left(k_{motor} - V_a\right)\right)\left(\delta_t\left(k_{motor} - V_a\right)\right)$$

$$T_{prop} = -k_{Tp}\left(k_\Omega \delta_t\right)^2$$

$$C_L(\ldots) \approx C_{L_0} + C_{L\alpha}\alpha + C_{Lq}\left(\tfrac{c}{2V_a}q\right) + C_{L_{\delta e}}\delta_e$$

$$C_D(\ldots) \approx C_{D_0} + \left|C_{D\alpha}\alpha\right| + \left|C_{Dq}\left(\tfrac{c}{2V_a}q\right)\right| + \left|C_{D_{\delta e}}\delta_e\right|$$

$$C_Y(\ldots) \approx C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\left(\tfrac{b}{2V_a}p\right) + C_{Y_r}\left(\tfrac{b}{2V_a}r\right) + C_{Y_{\delta a}}\delta_a + C_{Y_{\delta r}}\delta_r$$

$$C_l(\ldots) \approx C_{l_0} + C_{l_\beta}\beta + C_{l_p}\left(\tfrac{b}{2V_a}p\right) + C_{l_r}\left(\tfrac{b}{2V_a}r\right) + C_{l_{\delta a}}\delta_a + C_{l_{\delta r}}\delta_r$$

$$C_m(\ldots) \approx C_{m_0} + C_{m\alpha}\alpha + C_{mq}\left(\tfrac{c}{2V_a}q\right) + C_{m_{\delta e}}\delta_e$$

$$C_n(\ldots) \approx C_{n_0} + C_{n_\beta}\beta + C_{n_p}\left(\tfrac{b}{2V_a}p\right) + C_{n_r}\left(\tfrac{b}{2V_a}r\right) + C_{n_{\delta a}}\delta_a + C_{n_{\delta r}}\delta_r$$

Aero Vars:
- α : AoA, ° or rads
- β : Sideslip, ° or rads
- V_a: Airspeed, m/s

Controls:
- δ_e : Elevator, ° or rads
- δ_a : Aileron , ° or rads
- δ_r : Rudder , ° or rads
- δ_t : Throttle , (0-1)

Propeller Params:
- C_{prop}: Efficiency Coef.
- S_{prop}: *Prop. area*, m^2
- k_{motor}: *Motor constant, m/s*
- k_Ω: *Speed constant, m/s*
- k_{Tp}: *Torque constant, kg-m^2*

Body Rates:
- p : about x, °/s or rads/s
- q : about y, °/s or rads/s
- r : about z, °/s or rads/s

Environment:
- g: Gravity, 9.81 m/s^2
- ρ : Air Density, kg/m^3

Aero Params:
- S: Wing area, m^2
- b: Wing span, m
- c: Wing mean chord, m

Equations of Motion

- From Kinematics and Dynamics, the equations of motion are 12 ODEs:

Compact Form

$$\begin{bmatrix}\dot{p}_n\\ \dot{p}_e\\ \dot{p}_d\end{bmatrix} = R_b^{ned}\,\underline{v}_g^b \quad = \quad \left(R_{ned}^b\right)^T \underline{v}_g^b$$

$$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{bmatrix} = -\underline{\omega}_{b/i}^b \times \underline{v}_g^b + \frac{1}{mass}\,\underline{\mathrm{f}}^b$$

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix}$$

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \mathrm{J}^{-1}\left[-\underline{\omega}_{b/i}^b \times \left(\mathrm{J}\,\underline{\omega}_{b/i}^b\right) + \underline{\mathrm{m}}^b\right]$$

Expanded Form

$$\begin{bmatrix}\dot{p}_n\\ \dot{p}_e\\ \dot{p}_d\end{bmatrix} = \begin{bmatrix}\cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi-\cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi\\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi+\cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi\\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta\end{bmatrix}\begin{bmatrix}u\\ v\\ w\end{bmatrix}$$

$$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{bmatrix} = \begin{bmatrix}rv - qw\\ pw - ru\\ qu - pv\end{bmatrix} + \frac{1}{mass}\begin{bmatrix}f_x\\ f_y\\ f_z\end{bmatrix}$$

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix}$$

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}\frac{J_{xz}(J_x-J_y+J_z)}{\Gamma}pq - \frac{J_z(J_z-J_y)+J_{xz}^2}{\Gamma}qr\\ \frac{J_z-J_x}{J_y}pr - \frac{J_{xz}}{J_y}(p^2-r^2)\\ \frac{(J_x-J_y)J_x+J_{xz}^2}{\Gamma}pq - \frac{J_{xz}(J_x-J_y+J_z)}{\Gamma}qr\end{bmatrix} + \begin{bmatrix}\frac{J_z}{\Gamma}l + \frac{J_{xz}}{\Gamma}n\\ \frac{1}{J_y}m\\ \frac{J_x}{\Gamma}n + \frac{J_{xz}}{\Gamma}l\end{bmatrix}$$

$$\text{Where: } \Gamma = J_x J_z - J_{xz}^2$$

Uses $J_{xy}=J_{yz}=0$ Symmetry Assumption

The 12 equations of motion will be used to propagate vehicle states forward in time, given current states and external forces and moments.

Suggestion: To propagate states, use the compact form. Expanded form will be used for insight in developing autopilot.

Fixed Wing EoMs as 12 Scalar Functions

(We'll use these to develop an autopilot)

Derivatives of Positions

$$\dot{p}_n = (\cos\theta\cos\psi)u + (\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi)v + (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)w$$

$$\dot{p}_e = (\cos\theta\sin\psi)u + (\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi)v + (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)w$$

$$\dot{h} = -\dot{p}_d = (\sin\theta)u - (\sin\phi\cos\theta)v - (\cos\phi\cos\theta)w$$

$\longleftarrow$ *Altitude Rate*

Derivatives of Velocities

$$\dot{u} = rv - qw - g\sin\theta + \frac{\rho C_{prop} S_{prop}}{mass}\left\{V_a + \delta_t\left(k_{motor} - V_a\right)\right\}\left\{\delta_t\left(k_{motor} - V_a\right)\right\}$$
$$+ \frac{\rho V_a^2 S}{2(mass)}\left\{-\left(C_{Do} + C_{D\alpha}\alpha + \frac{c}{2V_a}C_{Dq}q + C_{D\delta e}\delta_e\right)\cos\alpha + \left(C_{Lo} + C_{L\alpha}\alpha + \frac{c}{2V_a}C_{Lq}q + C_{L\delta e}\delta_e\right)\sin\alpha\right\}$$

$$\dot{v} = pw - ru + g\cos\theta\sin\phi$$
$$+ \frac{\rho V_a^2 S}{2(mass)}\left\{C_{Yo} + C_{Y\beta}\beta + \frac{b}{2V_a}C_{Yp}p + \frac{b}{2V_a}C_{Yr}r + C_{Y\delta a}\delta_a + C_{Y\delta r}\delta_r\right\}$$

$$\dot{w} = qu - pv + g\cos\theta\cos\phi$$
$$+ \frac{\rho V_a^2 S}{2(mass)}\left\{-\left(C_{Do} + C_{D\alpha}\alpha + \frac{c}{2V_a}C_{Dq}q + C_{D\delta e}\delta_e\right)\sin\alpha - \left(C_{Lo} + C_{L\alpha}\alpha + \frac{c}{2V_a}C_{Lq}q + C_{L\delta e}\delta_e\right)\cos\alpha\right\}$$

Derivatives of Orientations

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta$$

$$\dot{\theta} = q\cos\phi - r\sin\phi$$

$$\dot{\psi} = q\sin\phi\sec\theta + r\cos\phi\sec\theta$$

Derivatives of Body Rates

$$\dot{p} = \frac{J_{xz}\left(J_x - J_y + J_z\right)}{\Gamma}pq - \frac{J_z\left(J_z - J_y\right) + J_{xz}^2}{\Gamma}qr - \frac{J_z}{\Gamma}k_{Tp}k_\Omega^2\delta_t^2$$
$$+ \frac{\rho V_a^2 Sb}{2}\left\{\frac{J_z C_{lo} + J_{xz}C_{no}}{\Gamma} + \frac{J_z C_{l\beta} + J_{xz}C_{n\beta}}{\Gamma}\beta + \left(\frac{b}{2V_a}\right)\frac{J_z C_{lp} + J_{xz}C_{np}}{\Gamma}p + \left(\frac{b}{2V_a}\right)\frac{J_z C_{lr} + J_{xz}C_{nr}}{\Gamma}r + \frac{J_z C_{l\delta a} + J_{xz}C_{n\delta a}}{\Gamma}\delta_a + \frac{J_z C_{l\delta r} + J_{xz}C_{n\delta r}}{\Gamma}\delta_r\right\}$$

$$\dot{q} = \frac{J_z - J_x}{J_y}pr - \frac{J_{xz}}{J_y}\left(p^2 - r^2\right)$$
$$+ \frac{\rho V_a^2 Sc}{2}\frac{1}{J_y}\left\{C_{mo} + C_{m\alpha}\alpha + \frac{c}{2V_a}C_{mq}q + C_{m\delta e}\delta_e\right\}$$

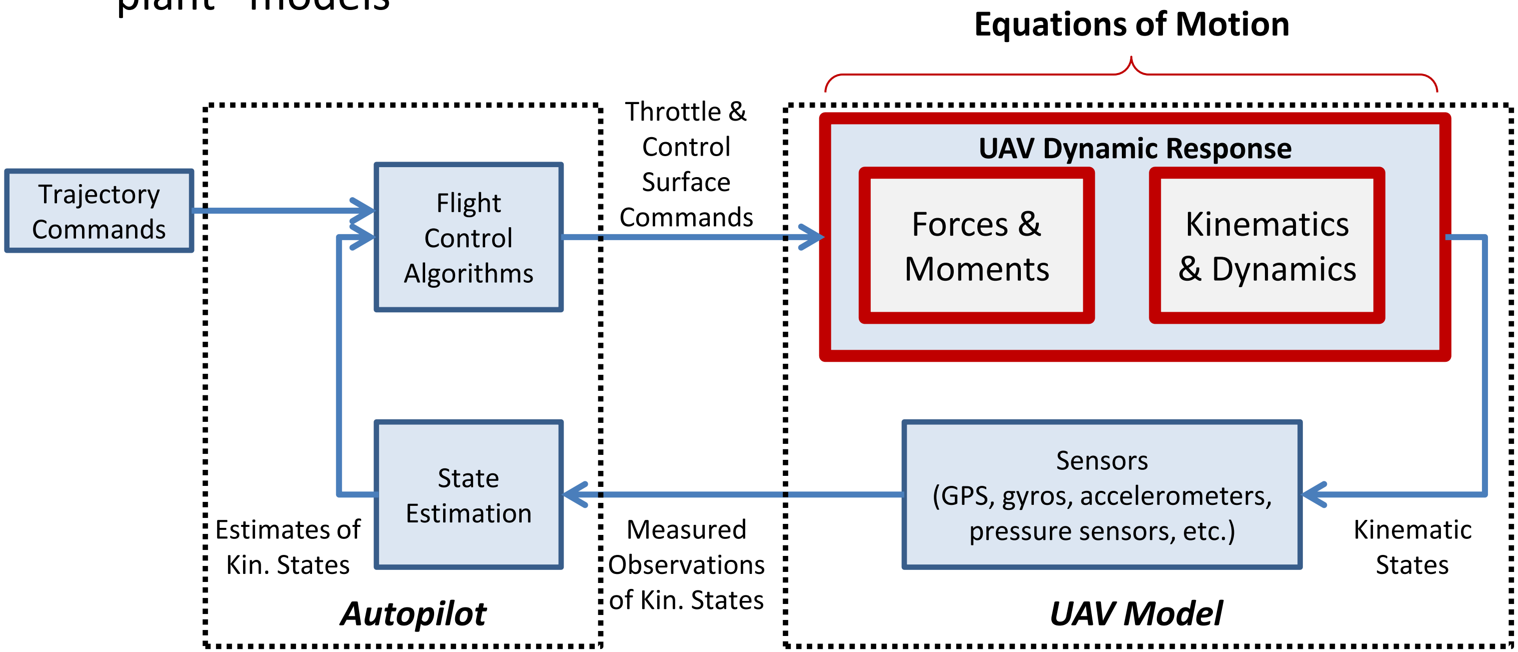
$$\dot{r} = \frac{\left(J_x - J_y\right)J_x + J_{xz}^2}{\Gamma}pq - \frac{J_{xz}\left(J_x - J_y + J_z\right)}{\Gamma}qr - \frac{J_{xz}}{\Gamma}k_{Tp}k_\Omega^2\delta_t^2$$
$$+ \frac{\rho V_a^2 Sb}{2}\left\{\frac{J_x C_{no} + J_{xz}C_{lo}}{\Gamma} + \frac{J_x C_{n\beta} + J_{xz}C_{l\beta}}{\Gamma}\beta + \left(\frac{b}{2V_a}\right)\frac{J_x C_{np} + J_{xz}C_{lp}}{\Gamma}p + \left(\frac{b}{2V_a}\right)\frac{J_x C_{nr} + J_{xz}C_{lr}}{\Gamma}r + \frac{J_x C_{n\delta a} + J_{xz}C_{l\delta a}}{\Gamma}\delta_a + \frac{J_x C_{n\delta r} + J_{xz}C_{l\delta r}}{\Gamma}\delta_r\right\}$$

Assumptions/Simplifications we've made:

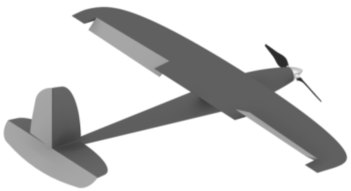
- NED (attached to ground) is an inertial reference frame
- Presuming a flat earth
- Vehicle has constant mass and inertia
- Vehicle is a rigid body
- Vehicle has symmetry about x-z plane
- Small angle-of-attack for linear and attached flow
- Aero moments provided about c.g.
- Provided linearized aero model is valid for reasonable deviations from nominal
- Forces & moments derived for single axial propeller
- Ignoring higher order propeller effects:
  - Rotational drag, blade flapping, motor response characteristics, aerodynamic & propeller interactions, etc.

Equations of Motion

- At this point, we've fully developed the non-linear fixed-wing UAV "plant" Equations of Motion
  - EoMs dictate aircraft dynamic response based on control signals and current states
  - EoMs are essential for simulating vehicle motion, but are too complicated (non-linear and coupled) to use directly in an autopilot controller design
  - In order to design a simple feedback control autopilot, we need simpler "plant" models

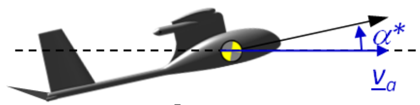


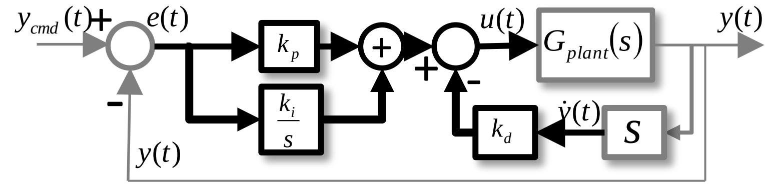
Path to Autopilot Development



Develop the non-linear model of motion

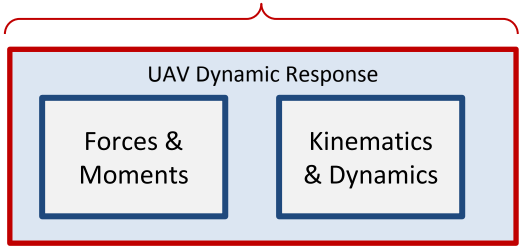
Find “trim” point that balances forces and moments at a nominal speed

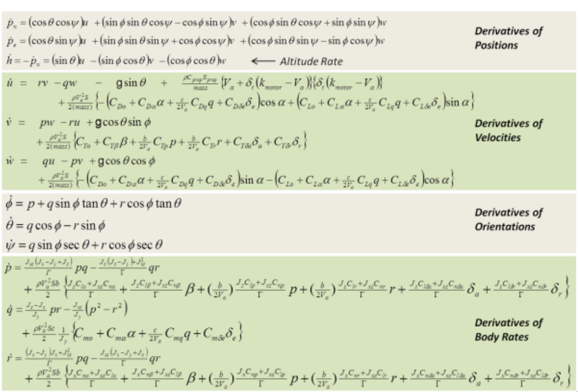




Determine controller structure, e.g.:
- Classical: PID or PI w/ rate feedback, etc.
- Modern: LQR, Full-State FB, H-∞, etc.

Equations of Motion





12 EoMs

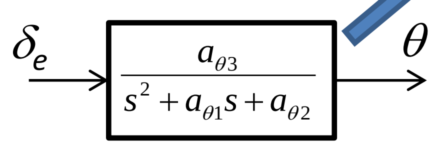
“Linearize” EoMs about trim point to make de-coupled linear response models

Two options:

a) Numerically compute linearized state-space model (Representative, but high order)

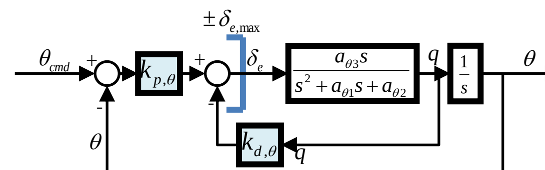
$$\dot{\bar{\mathbf{x}}} = A\bar{\mathbf{x}} + B\bar{\mathbf{u}}$$

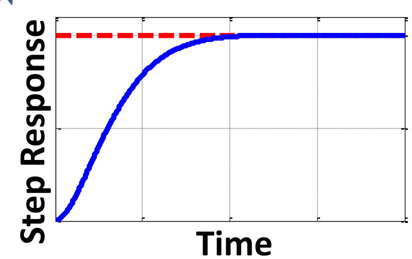
b) Analytically derive simpler 1st & 2nd order linear transfer function response models



Our method

Select controller-specific gains to meet desired performance around “linearized” operating point, e.g.

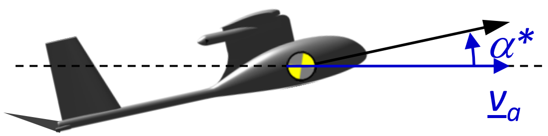




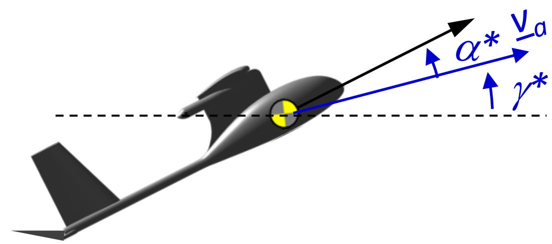
Trim Point

- In general, a trim point is an equilibrium state where forces and moments are balanced to achieve a desired motion
- Examples of trim:

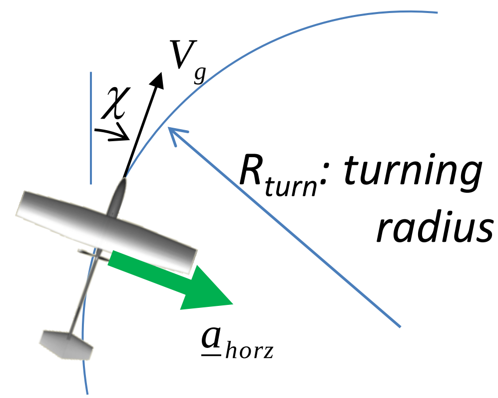
Straight-and-Level Flight



Constant Climb

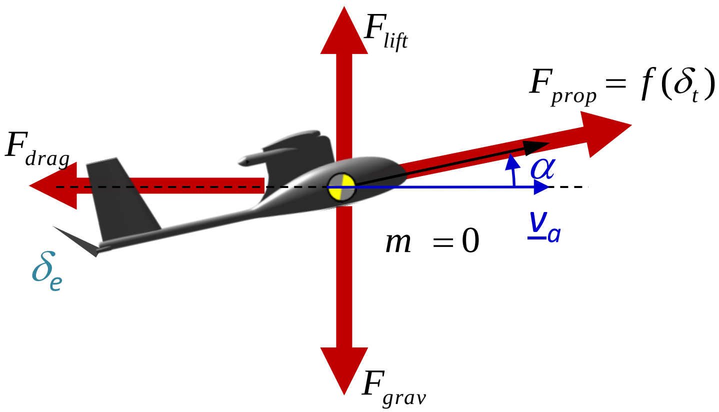


Constant Turn



- We will only use Straight-and-Level flight trim condition
  - We will linearize flight dynamics about this condition
- Numerous methods exist for finding a trim condition
  - Simple method described on next slide (different from book)

Straight-and-Level Trim Point



- Objective: Find the α^*, δ_e^*, and δ_t^* that balance the longitudinal forces and moments to achieve constant straight-and-level flight.
  - We want:

$$\underline{\mathrm{f}}^b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 \\ D.C. \\ 0 \end{bmatrix} \qquad \underline{\mathrm{m}}^b = \begin{bmatrix} l \\ m \\ n \end{bmatrix} = \begin{bmatrix} D.C. \\ 0 \\ D.C. \end{bmatrix}$$

D.C.: Don't Care

- Method: Determine $[\alpha, \delta_e, \delta_t]$ which minimizes a cost function J_{cost}, where:

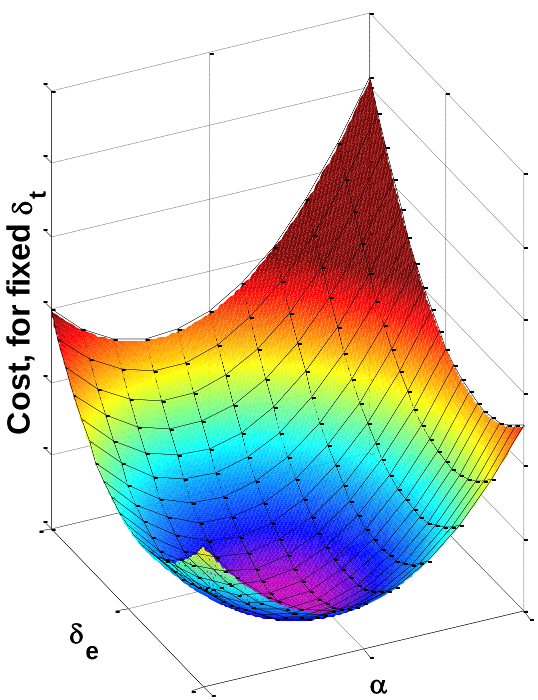
$$J_{cost} = f_x^2 + f_z^2 + m^2$$

Given: $u = V_a \cos\alpha$ (Assumes no wind)

$$w = V_a \sin\alpha$$

$$\theta = \alpha$$

$$q = 0$$



- Matlab provides a minimization routine called fminsearch(). See homework.

Linearization of Equations of Motion

- We need to linearize Equations of Motion to:
  - Gain intuition of aircraft dynamics
  - Develop autopilot control algorithms based on linear feedback theory
    - E.g. PID, Classical 3-loop, LQR, H-infinity, etc.
- Linearization is performed <u>about a trim point</u> and can be achieved:
  - Analytically (i.e. manipulating equations to form linearized derivative equations)
  - Numerically (i.e. computing linearized derivatives via small perturbations)
- We will linearize to form a state-space model representing dynamics of small deviations from trim

$$\dot{\bar{\mathbf{x}}} = A\bar{\mathbf{x}} + B\bar{\mathbf{u}}$$

$\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$: Deviations from trim

Derivatives of Positions

$$\dot{p}_n = (\cos\theta\cos\psi)u + (\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi)v + (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)w$$

$$\dot{p}_e = (\cos\theta\sin\psi)u + (\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi)v + (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)w$$

$$\dot{h} = -\dot{p}_n = (\sin\theta)u - (\sin\phi\cos\theta)v - (\cos\phi\cos\theta)w \quad \longleftarrow \text{Altitude Rate}$$

Derivatives of Velocities

$$\dot{u} = rv - qw - g\sin\theta + \frac{\rho C_{prop} S_{prop}}{mass}\{V_a + \delta_t(k_{motor} - V_a)\}\{\delta_t(k_{motor} - V_a)\} + \frac{\rho V_a^2 S}{2(mass)}\left\{-\left(C_{Do} + C_{D\alpha}\alpha + \frac{c}{2V_a}C_{Dq}q + C_{D\delta e}\delta_e\right)\cos\alpha + \left(C_{Lo} + C_{L\alpha}\alpha + \frac{c}{2V_a}C_{Lq}q + C_{L\delta e}\delta_e\right)\sin\alpha\right\}$$

$$\dot{v} = pw - ru + g\cos\theta\sin\phi + \frac{\rho V_a^2 S}{2(mass)}\left\{C_{Yo} + C_{Y\beta}\beta + \frac{b}{2V_a}C_{Yp}p + \frac{b}{2V_a}C_{Yr}r + C_{Y\delta a}\delta_a + C_{Y\delta r}\delta_r\right\}$$

$$\dot{w} = qu - pv + g\cos\theta\cos\phi + \frac{\rho V_a^2 S}{2(mass)}\left\{-\left(C_{Do} + C_{D\alpha}\alpha + \frac{c}{2V_a}C_{Dq}q + C_{D\delta e}\delta_e\right)\sin\alpha - \left(C_{Lo} + C_{L\alpha}\alpha + \frac{c}{2V_a}C_{Lq}q + C_{L\delta e}\delta_e\right)\cos\alpha\right\}$$

Derivatives of Orientations

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta$$

$$\dot{\theta} = q\cos\phi - r\sin\phi$$

$$\dot{\psi} = q\sin\phi\sec\theta + r\cos\phi\sec\theta$$

Derivatives of Body Rates

$$\dot{p} = \frac{J_{xz}(J_x - J_y + J_z)}{\Gamma}pq - \frac{J_z(J_z - J_y) + J_{xz}^2}{\Gamma}qr + \frac{\rho V_a^2 Sb}{2}\left\{\frac{J_z C_{lo} + J_{xz}C_{no}}{\Gamma} + \frac{J_z C_{l\beta} + J_{xz}C_{n\beta}}{\Gamma}\beta + \left(\frac{b}{2V_a}\right)\frac{J_z C_{lp} + J_{xz}C_{np}}{\Gamma}p + \left(\frac{b}{2V_a}\right)\frac{J_z C_{lr} + J_{xz}C_{nr}}{\Gamma}r + \frac{J_z C_{l\delta a} + J_{xz}C_{n\delta a}}{\Gamma}\delta_a + \frac{J_z C_{l\delta r} + J_{xz}C_{n\delta r}}{\Gamma}\delta_r\right\}$$

$$\dot{q} = \frac{J_z - J_x}{J_y}pr - \frac{J_{xz}}{J_y}(p^2 - r^2) + \frac{\rho V_a^2 Sc}{2}\frac{1}{J_y}\left\{C_{mo} + C_{m\alpha}\alpha + \frac{c}{2V_a}C_{mq}q + C_{m\delta e}\delta_e\right\}$$

$$\dot{r} = \frac{(J_x - J_y)J_x + J_{xz}^2}{\Gamma}pq - \frac{J_{xz}(J_x - J_y + J_z)}{\Gamma}qr + \frac{\rho V_a^2 Sb}{2}\left\{\frac{J_x C_{no} + J_{xz}C_{lo}}{\Gamma} + \frac{J_x C_{n\beta} + J_{xz}C_{l\beta}}{\Gamma}\beta + \left(\frac{b}{2V_a}\right)\frac{J_x C_{np} + J_{xz}C_{lp}}{\Gamma}p + \left(\frac{b}{2V_a}\right)\frac{J_x C_{nr} + J_{xz}C_{lr}}{\Gamma}r + \frac{J_x C_{n\delta a} + J_{xz}C_{l\delta a}}{\Gamma}\delta_a + \frac{J_x C_{n\delta r} + J_{xz}C_{l\delta r}}{\Gamma}\delta_r\right\}$$

$$\begin{bmatrix} \dot{\bar{p}}_n \\ \dot{\bar{p}}_e \\ \dot{\bar{p}}_d \\ \dot{\bar{u}} \\ \dot{\bar{v}} \\ \dot{\bar{w}} \\ \dot{\bar{\phi}} \\ \dot{\bar{\theta}} \\ \dot{\bar{\psi}} \\ \dot{\bar{p}} \\ \dot{\bar{q}} \\ \dot{\bar{r}} \end{bmatrix} = \underbrace{\begin{bmatrix} * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \end{bmatrix}}_{\text{A: 12x12}} \begin{bmatrix} \bar{p}_n \\ \bar{p}_e \\ \bar{p}_d \\ \bar{u} \\ \bar{v} \\ \bar{w} \\ \bar{\phi} \\ \bar{\theta} \\ \bar{\psi} \\ \bar{p} \\ \bar{q} \\ \bar{r} \end{bmatrix} + \underbrace{\begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}}_{\text{B: 12x4}} \begin{bmatrix} \bar{\delta}_e \\ \bar{\delta}_a \\ \bar{\delta}_r \\ \bar{\delta}_t \end{bmatrix}$$

Linear State-space Models

12 EoMs: $\dot{\underline{x}} = f(\underline{x}, \underline{u})$

Trim State: $\underline{x}^*$

We will derive the state space equations about trim.
Recall that the nonlinear equations are given by $\dot{x} = f(x, u)$. The trim condition is $\dot{x}^* = f(x^*, u^*)$. Let $\bar{x} = x - x^*$ be the deviation from trim. Then

$$\begin{aligned}\dot{\bar{x}} &= \boxed{\dot{x}} - \dot{x}^* \\ &= \boxed{f(x, u)} - f(x^*, u^*) \\ &= \boxed{f(x + x^* - x^*, u + u^* - u^*)} - f(x^*, u^*) \\ &= \boxed{f(x^* + \bar{x}, u^* + \bar{u})} - f(x^*, u^*)\end{aligned}$$

Using a Taylor series expansion around trim gives

$$\begin{aligned}\dot{\bar{x}} &= \boxed{f(x^*, u^*) + \frac{\partial f(x^*, u^*)}{\partial x}\bar{x} + \frac{\partial f(x^*, u^*)}{\partial u}\bar{u} + H.O.T} - f(x^*, u^*) \\ &\approx \frac{\partial f(x^*, u^*)}{\partial x}\bar{x} + \frac{\partial f(x^*, u^*)}{\partial u}\bar{u} \\ &\triangleq A\bar{x} + B\bar{u}\end{aligned}$$

Slide courtesy of book website: Beard & McLain, “Small Unmanned Aircraft”, *Princeton University Press*, 2012

Jacobians of Equations of Motion

12 EoMs: $\dot{\underline{x}} = f(\underline{x}, \underline{u})$

$$\dot{\bar{\mathbf{x}}} = A\bar{\mathbf{x}} + B\bar{\mathbf{u}}$$

$\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$: Deviations from trim point

$$A = \frac{\partial f(\mathbf{x}^*, \mathbf{u}^*)}{\partial \mathbf{x}} = \begin{pmatrix} \left[\frac{\partial \dot{\underline{x}}}{\partial p_n}\right] & \left[\frac{\partial \dot{\underline{x}}}{\partial p_e}\right] & \cdots & \left[\frac{\partial \dot{\underline{x}}}{\partial \psi}\right] & \cdots & \left[\frac{\partial \dot{\underline{x}}}{\partial q}\right] & \left[\frac{\partial \dot{\underline{x}}}{\partial r}\right] \end{pmatrix}_{12x12}$$

*Jacobian of f(**x**,**u**) wrt **x**, evaluated at (**x***,**u***)*

$$B = \frac{\partial f(\mathbf{x}^*, \mathbf{u}^*)}{\partial \mathbf{u}} = \begin{pmatrix} \left[\frac{\partial \dot{\underline{x}}}{\partial \delta_e}\right] & \cdots & \left[\frac{\partial \dot{\underline{x}}}{\partial \delta_t}\right] \end{pmatrix}_{12x4}$$

*Jacobian of f(**x**,**u**) wrt **u**, evaluated at (**x***,**u***)*

Jacobians of Equations of Motion

12 EoMs: $\dot{\underline{x}} = f(\underline{x}, \underline{u})$

$$\dot{\bar{\mathbf{x}}} = A\bar{\mathbf{x}} + B\bar{\mathbf{u}}$$

$\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$: Deviations from trim point

$$A = \frac{\partial f(\mathbf{x}^*, \mathbf{u}^*)}{\partial \mathbf{x}} = \begin{pmatrix}
\frac{\partial \dot{p}_n}{\partial p_n} & \frac{\partial \dot{p}_n}{\partial p_e} & \frac{\partial \dot{p}_n}{\partial p_d} & \frac{\partial \dot{p}_n}{\partial u} & \frac{\partial \dot{p}_n}{\partial v} & \frac{\partial \dot{p}_n}{\partial w} & \frac{\partial \dot{p}_n}{\partial \phi} & \frac{\partial \dot{p}_n}{\partial \theta} & \frac{\partial \dot{p}_n}{\partial \psi} & \frac{\partial \dot{p}_n}{\partial p} & \frac{\partial \dot{p}_n}{\partial q} & \frac{\partial \dot{p}_n}{\partial r} \\
\frac{\partial \dot{p}_e}{\partial p_n} & \frac{\partial \dot{p}_e}{\partial p_e} & \frac{\partial \dot{p}_e}{\partial p_d} & \frac{\partial \dot{p}_e}{\partial u} & \frac{\partial \dot{p}_e}{\partial v} & \frac{\partial \dot{p}_e}{\partial w} & \frac{\partial \dot{p}_e}{\partial \phi} & \frac{\partial \dot{p}_e}{\partial \theta} & \frac{\partial \dot{p}_e}{\partial \psi} & \frac{\partial \dot{p}_e}{\partial p} & \frac{\partial \dot{p}_e}{\partial q} & \frac{\partial \dot{p}_e}{\partial r} \\
\frac{\partial \dot{p}_d}{\partial p_n} & \frac{\partial \dot{p}_d}{\partial p_e} & \frac{\partial \dot{p}_d}{\partial p_d} & \frac{\partial \dot{p}_d}{\partial u} & \frac{\partial \dot{p}_d}{\partial v} & \frac{\partial \dot{p}_d}{\partial w} & \frac{\partial \dot{p}_d}{\partial \phi} & \frac{\partial \dot{p}_d}{\partial \theta} & \frac{\partial \dot{p}_d}{\partial \psi} & \frac{\partial \dot{p}_d}{\partial p} & \frac{\partial \dot{p}_d}{\partial q} & \frac{\partial \dot{p}_d}{\partial r} \\
\frac{\partial \dot{u}}{\partial p_n} & \frac{\partial \dot{u}}{\partial p_e} & \frac{\partial \dot{u}}{\partial p_d} & \frac{\partial \dot{u}}{\partial u} & \frac{\partial \dot{u}}{\partial v} & \frac{\partial \dot{u}}{\partial w} & \frac{\partial \dot{u}}{\partial \phi} & \frac{\partial \dot{u}}{\partial \theta} & \frac{\partial \dot{u}}{\partial \psi} & \frac{\partial \dot{u}}{\partial p} & \frac{\partial \dot{u}}{\partial q} & \frac{\partial \dot{u}}{\partial r} \\
\frac{\partial \dot{v}}{\partial p_n} & \frac{\partial \dot{v}}{\partial p_e} & \frac{\partial \dot{v}}{\partial p_d} & \frac{\partial \dot{v}}{\partial u} & \frac{\partial \dot{v}}{\partial v} & \frac{\partial \dot{v}}{\partial w} & \frac{\partial \dot{v}}{\partial \phi} & \frac{\partial \dot{v}}{\partial \theta} & \frac{\partial \dot{v}}{\partial \psi} & \frac{\partial \dot{v}}{\partial p} & \frac{\partial \dot{v}}{\partial q} & \frac{\partial \dot{v}}{\partial r} \\
\frac{\partial \dot{w}}{\partial p_n} & \frac{\partial \dot{w}}{\partial p_e} & \frac{\partial \dot{w}}{\partial p_d} & \frac{\partial \dot{w}}{\partial u} & \frac{\partial \dot{w}}{\partial v} & \frac{\partial \dot{w}}{\partial w} & \frac{\partial \dot{w}}{\partial \phi} & \frac{\partial \dot{w}}{\partial \theta} & \frac{\partial \dot{w}}{\partial \psi} & \frac{\partial \dot{w}}{\partial p} & \frac{\partial \dot{w}}{\partial q} & \frac{\partial \dot{w}}{\partial r} \\
\frac{\partial \dot{\phi}}{\partial p_n} & \frac{\partial \dot{\phi}}{\partial p_e} & \frac{\partial \dot{\phi}}{\partial p_d} & \frac{\partial \dot{\phi}}{\partial u} & \frac{\partial \dot{\phi}}{\partial v} & \frac{\partial \dot{\phi}}{\partial w} & \frac{\partial \dot{\phi}}{\partial \phi} & \frac{\partial \dot{\phi}}{\partial \theta} & \frac{\partial \dot{\phi}}{\partial \psi} & \frac{\partial \dot{\phi}}{\partial p} & \frac{\partial \dot{\phi}}{\partial q} & \frac{\partial \dot{\phi}}{\partial r} \\
\frac{\partial \dot{\theta}}{\partial p_n} & \frac{\partial \dot{\theta}}{\partial p_e} & \frac{\partial \dot{\theta}}{\partial p_d} & \frac{\partial \dot{\theta}}{\partial u} & \frac{\partial \dot{\theta}}{\partial v} & \frac{\partial \dot{\theta}}{\partial w} & \frac{\partial \dot{\theta}}{\partial \phi} & \frac{\partial \dot{\theta}}{\partial \theta} & \frac{\partial \dot{\theta}}{\partial \psi} & \frac{\partial \dot{\theta}}{\partial p} & \frac{\partial \dot{\theta}}{\partial q} & \frac{\partial \dot{\theta}}{\partial r} \\
\frac{\partial \dot{\psi}}{\partial p_n} & \frac{\partial \dot{\psi}}{\partial p_e} & \frac{\partial \dot{\psi}}{\partial p_d} & \frac{\partial \dot{\psi}}{\partial u} & \frac{\partial \dot{\psi}}{\partial v} & \frac{\partial \dot{\psi}}{\partial w} & \frac{\partial \dot{\psi}}{\partial \phi} & \frac{\partial \dot{\psi}}{\partial \theta} & \frac{\partial \dot{\psi}}{\partial \psi} & \frac{\partial \dot{\psi}}{\partial p} & \frac{\partial \dot{\psi}}{\partial q} & \frac{\partial \dot{\psi}}{\partial r} \\
\frac{\partial \dot{p}}{\partial p_n} & \frac{\partial \dot{p}}{\partial p_e} & \frac{\partial \dot{p}}{\partial p_d} & \frac{\partial \dot{p}}{\partial u} & \frac{\partial \dot{p}}{\partial v} & \frac{\partial \dot{p}}{\partial w} & \frac{\partial \dot{p}}{\partial \phi} & \frac{\partial \dot{p}}{\partial \theta} & \frac{\partial \dot{p}}{\partial \psi} & \frac{\partial \dot{p}}{\partial p} & \frac{\partial \dot{p}}{\partial q} & \frac{\partial \dot{p}}{\partial r} \\
\frac{\partial \dot{q}}{\partial p_n} & \frac{\partial \dot{q}}{\partial p_e} & \frac{\partial \dot{q}}{\partial p_d} & \frac{\partial \dot{q}}{\partial u} & \frac{\partial \dot{q}}{\partial v} & \frac{\partial \dot{q}}{\partial w} & \frac{\partial \dot{q}}{\partial \phi} & \frac{\partial \dot{q}}{\partial \theta} & \frac{\partial \dot{q}}{\partial \psi} & \frac{\partial \dot{q}}{\partial p} & \frac{\partial \dot{q}}{\partial q} & \frac{\partial \dot{q}}{\partial r} \\
\frac{\partial \dot{r}}{\partial p_n} & \frac{\partial \dot{r}}{\partial p_e} & \frac{\partial \dot{r}}{\partial p_d} & \frac{\partial \dot{r}}{\partial u} & \frac{\partial \dot{r}}{\partial v} & \frac{\partial \dot{r}}{\partial w} & \frac{\partial \dot{r}}{\partial \phi} & \frac{\partial \dot{r}}{\partial \theta} & \frac{\partial \dot{r}}{\partial \psi} & \frac{\partial \dot{r}}{\partial p} & \frac{\partial \dot{r}}{\partial q} & \frac{\partial \dot{r}}{\partial r}
\end{pmatrix}_{12x12}$$

*Jacobian of f(**x**,**u**) wrt **x**, evaluated at (**x***,**u***)*

$$B = \frac{\partial f(\mathbf{x}^*, \mathbf{u}^*)}{\partial \mathbf{u}} = \begin{pmatrix}
\frac{\partial \dot{p}_n}{\partial \delta_e} & \frac{\partial \dot{p}_n}{\partial \delta_a} & \frac{\partial \dot{p}_n}{\partial \delta_r} & \frac{\partial \dot{p}_n}{\partial \delta_t} \\
\frac{\partial \dot{p}_e}{\partial \delta_e} & \frac{\partial \dot{p}_e}{\partial \delta_a} & \frac{\partial \dot{p}_e}{\partial \delta_r} & \frac{\partial \dot{p}_e}{\partial \delta_t} \\
\frac{\partial \dot{p}_d}{\partial \delta_e} & \frac{\partial \dot{p}_d}{\partial \delta_a} & \frac{\partial \dot{p}_d}{\partial \delta_r} & \frac{\partial \dot{p}_d}{\partial \delta_t} \\
\frac{\partial \dot{u}}{\partial \delta_e} & \frac{\partial \dot{u}}{\partial \delta_a} & \frac{\partial \dot{u}}{\partial \delta_r} & \frac{\partial \dot{u}}{\partial \delta_t} \\
\frac{\partial \dot{v}}{\partial \delta_e} & \frac{\partial \dot{v}}{\partial \delta_a} & \frac{\partial \dot{v}}{\partial \delta_r} & \frac{\partial \dot{v}}{\partial \delta_t} \\
\frac{\partial \dot{w}}{\partial \delta_e} & \frac{\partial \dot{w}}{\partial \delta_a} & \frac{\partial \dot{w}}{\partial \delta_r} & \frac{\partial \dot{w}}{\partial \delta_t} \\
\frac{\partial \dot{\phi}}{\partial \delta_e} & \frac{\partial \dot{\phi}}{\partial \delta_a} & \frac{\partial \dot{\phi}}{\partial \delta_r} & \frac{\partial \dot{\phi}}{\partial \delta_t} \\
\frac{\partial \dot{\theta}}{\partial \delta_e} & \frac{\partial \dot{\theta}}{\partial \delta_a} & \frac{\partial \dot{\theta}}{\partial \delta_r} & \frac{\partial \dot{\theta}}{\partial \delta_t} \\
\frac{\partial \dot{\psi}}{\partial \delta_e} & \frac{\partial \dot{\psi}}{\partial \delta_a} & \frac{\partial \dot{\psi}}{\partial \delta_r} & \frac{\partial \dot{\psi}}{\partial \delta_t} \\
\frac{\partial \dot{p}}{\partial \delta_e} & \frac{\partial \dot{p}}{\partial \delta_a} & \frac{\partial \dot{p}}{\partial \delta_r} & \frac{\partial \dot{p}}{\partial \delta_t} \\
\frac{\partial \dot{q}}{\partial \delta_e} & \frac{\partial \dot{q}}{\partial \delta_a} & \frac{\partial \dot{q}}{\partial \delta_r} & \frac{\partial \dot{q}}{\partial \delta_t} \\
\frac{\partial \dot{r}}{\partial \delta_e} & \frac{\partial \dot{r}}{\partial \delta_a} & \frac{\partial \dot{r}}{\partial \delta_r} & \frac{\partial \dot{r}}{\partial \delta_t}
\end{pmatrix}_{12x4}$$

*Jacobian of f(**x**,**u**) wrt **u**, evaluated at (**x***,**u***)*

For example

$$w = w^* + \overline{w}$$

$$\dot{\overline{w}} \approx \frac{\partial \dot{w}}{\partial p_n}\overline{p}_n + \frac{\partial \dot{w}}{\partial p_e}\overline{p}_e + \frac{\partial \dot{w}}{\partial p_d}\overline{p}_d + \frac{\partial \dot{w}}{\partial u}\overline{u} + \frac{\partial \dot{w}}{\partial v}\overline{v} + \frac{\partial \dot{w}}{\partial w}\overline{w} + \frac{\partial \dot{w}}{\partial \phi}\overline{\phi} + \frac{\partial \dot{w}}{\partial \theta}\overline{\theta} + \frac{\partial \dot{w}}{\partial \psi}\overline{\psi} + \frac{\partial \dot{w}}{\partial p}\overline{p} + \frac{\partial \dot{w}}{\partial q}\overline{q} + \frac{\partial \dot{w}}{\partial r}\overline{r}$$

$$+ \frac{\partial \dot{w}}{\partial \delta_e}\overline{\delta}_e + \frac{\partial \dot{w}}{\partial \delta_a}\overline{\delta}_a + \frac{\partial \dot{w}}{\partial \delta_r}\overline{\delta}_r + \frac{\partial \dot{w}}{\partial \delta_t}\overline{\delta}_t$$

Numerical Computation of Jacobians

- To numerically compute the Jacobian matrices A and B:
  - Let: x_i be the i^{th} state, e.g. for our model, $w = x_6$
  - Let: $f_i(\underline{x}, \underline{u})$ be the non-linear expression for the derivative of the i^{th} state at condition ($\underline{x}$, $\underline{u}$)
    - e.g $\dot{w} = f_6(\underline{x}, \underline{u})$
  - Then, the i^{th} column of A is:

$$\begin{pmatrix} \dfrac{\partial f_1(\underline{x}^*, \underline{u}^*)}{\partial x_i} \\ \dfrac{\partial f_2(\underline{x}^*, \underline{u}^*)}{\partial x_i} \\ \vdots \\ \dfrac{\partial f_n(\underline{x}^*, \underline{u}^*)}{\partial x_i} \end{pmatrix} \approx \frac{f(\underline{x}^* + \varepsilon \cdot e_i,\ \underline{u}^*) - f(\underline{x}^*, \underline{u}^*)}{\varepsilon}$$

where: ε is a really small value, and $e_i = \begin{bmatrix} 0 & 0 & \cdots 1 \cdots 0 \end{bmatrix}^T$ (i^{th} component is 1)

  - Similarly, the i^{th} column of B is:

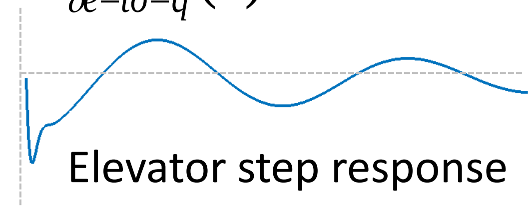
$$\begin{pmatrix} \dfrac{\partial f_1(\underline{x}^*, \underline{u}^*)}{\partial u_i} \\ \dfrac{\partial f_2(\underline{x}^*, \underline{u}^*)}{\partial u_i} \\ \vdots \\ \dfrac{\partial f_n(\underline{x}^*, \underline{u}^*)}{\partial u_i} \end{pmatrix} \approx \frac{f(\underline{x}^*,\ \underline{u}^* + \varepsilon \cdot e_i) - f(\underline{x}^*, \underline{u}^*)}{\varepsilon}$$

Uses of Linear State Space Model

$$\begin{bmatrix}\dot{\bar{p}}_n\\ \dot{\bar{p}}_e\\ \dot{\bar{p}}_d\\ \dot{\bar{u}}\\ \dot{\bar{v}}\\ \dot{\bar{w}}\\ \dot{\bar{\phi}}\\ \dot{\bar{\theta}}\\ \dot{\bar{\psi}}\\ \dot{\bar{p}}\\ \dot{\bar{q}}\\ \dot{\bar{r}}\end{bmatrix} = \underbrace{\begin{bmatrix} * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & *\\ * & * & * & * & * & * & * & * & * & * & * & * \end{bmatrix}}_{\mathbf{A}\ \ 12\mathrm{x}12} \begin{bmatrix}\bar{p}_n\\ \bar{p}_e\\ \bar{p}_d\\ \bar{u}\\ \bar{v}\\ \bar{w}\\ \bar{\phi}\\ \bar{\theta}\\ \bar{\psi}\\ \bar{p}\\ \bar{q}\\ \bar{r}\end{bmatrix} + \underbrace{\begin{bmatrix} * & * & * & *\\ * & * & * & *\\ * & * & * & *\\ * & * & * & *\\ * & * & * & *\\ * & * & * & *\\ * & * & * & *\\ * & * & * & *\\ * & * & * & *\\ * & * & * & *\\ * & * & * & *\\ * & * & * & * \end{bmatrix}}_{\mathbf{B}\ \ 12\mathrm{x}4} \begin{bmatrix}\bar{\delta}_e\\ \bar{\delta}_a\\ \bar{\delta}_r\\ \bar{\delta}_t\end{bmatrix}$$

Coupled Linear Model

$H_{\delta e-to-q}(s)$



A & B matrices can be used to obtain linear responses from inputs to states

Gain intuition of generic aircraft response modes

$$\begin{bmatrix}\dot{\bar{u}}\\ \dot{\bar{w}}\\ \dot{\bar{q}}\\ \dot{\bar{\theta}}\\ \dot{\bar{p}}_d\end{bmatrix} = \underbrace{\begin{bmatrix} * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & * \end{bmatrix}}_{\mathbf{A}_{\mathrm{LON}}\ \ 5\mathrm{x}5} \begin{bmatrix}\bar{u}\\ \bar{w}\\ \bar{q}\\ \bar{\theta}\\ \bar{p}_d\end{bmatrix} + \underbrace{\begin{bmatrix} * & *\\ * & *\\ * & *\\ * & *\\ * & * \end{bmatrix}}_{\mathbf{B}_{\mathrm{LON}}\ \ 5\mathrm{x}2} \begin{bmatrix}\bar{\delta}_e\\ \bar{\delta}_t\end{bmatrix}$$

Decoupled Longitudinal Linear Model

$$\begin{bmatrix}\dot{\bar{v}}\\ \dot{\bar{p}}\\ \dot{\bar{r}}\\ \dot{\bar{\phi}}\\ \dot{\bar{\psi}}\end{bmatrix} = \underbrace{\begin{bmatrix} * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & * \end{bmatrix}}_{\mathbf{A}_{\mathrm{LAT}}\ \ 5\mathrm{x}5} \begin{bmatrix}\bar{v}\\ \bar{p}\\ \bar{r}\\ \bar{\phi}\\ \bar{\psi}\end{bmatrix} + \underbrace{\begin{bmatrix} * & *\\ * & *\\ * & *\\ * & *\\ * & * \end{bmatrix}}_{\mathbf{B}_{\mathrm{LAT}}\ \ 5\mathrm{x}2} \begin{bmatrix}\bar{\delta}_a\\ \bar{\delta}_r\end{bmatrix}$$

Decoupled Lateral Linear Model

Decoupled Linear Models are often the basis for autopilot controllers

Decoupling Linear State Space Models

- Most aircraft are designed to be fairly de-coupled:
  - Longitudinal motion (pitching, climbing, airspeed)
    - 5 states: [u, w, q, θ, p_d] and two controls: [δ_e δ_t] *(elevator and throttle)*
  - Lateral motion (rolling, yawing, turning)
    - 5 states: [v, p, r, ϕ, ψ] and two controls: [δ_a δ_r] *(aileron and rudder)*
- We can isolate the longitudinal and lateral states in A and B
  - Let: $k_{pd}=3$, $k_u=4$, $k_v=5$, $k_w=6$, $k_\phi=7$, $k_\theta=8$, $k_\psi=9$, $k_p=10$, $k_q=11$, $k_r=12$
  - Let: $k_{\delta e}=1$, $k_{\delta a}=2$, $k_{\delta r}=3$, $k_{\delta t}=4$
  - Longitudinal
    - $A_{lon} = A([k_u\ k_w\ k_q\ k_\theta\ k_{pd}], [k_u\ k_w\ k_q\ k_\theta\ k_{pd}])$ [5x5]
    - $B_{lon} = B([k_u\ k_w\ k_q\ k_\theta\ k_{pd}], [k_{\delta e}\ k_{\delta t}])$ [5x2]
  - Lateral
    - $A_{lat} = A([k_v\ k_p\ k_r\ k_\phi\ k_\psi], [k_v\ k_p\ k_r\ k_\phi\ k_\psi])$ [5x5]
    - $B_{lat} = B([k_v\ k_p\ k_r\ k_\phi\ k_\psi], [k_{\delta a}\ k_{\delta r}])$ [5x2]

$$\begin{bmatrix}\dot{\bar u}\\ \dot{\bar w}\\ \dot{\bar q}\\ \dot{\bar \theta}\\ \dot{\bar p}_d\end{bmatrix} = \underbrace{\begin{bmatrix}* & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\end{bmatrix}}_{A_{LON}\ 5\times5}\begin{bmatrix}\bar u\\ \bar w\\ \bar q\\ \bar \theta\\ \bar p_d\end{bmatrix} + \underbrace{\begin{bmatrix}* & *\\ * & *\\ * & *\\ * & *\\ * & *\end{bmatrix}}_{B_{LON}\ 5\times2}\begin{bmatrix}\bar\delta_e\\ \bar\delta_t\end{bmatrix}$$

Decoupled Longitudinal Linear Model

$$\begin{bmatrix}\dot{\bar v}\\ \dot{\bar p}\\ \dot{\bar r}\\ \dot{\bar \phi}\\ \dot{\bar \psi}\end{bmatrix} = \underbrace{\begin{bmatrix}* & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\\ * & * & * & * & *\end{bmatrix}}_{A_{LAT}\ 5\times5}\begin{bmatrix}\bar v\\ \bar p\\ \bar r\\ \bar \phi\\ \bar \psi\end{bmatrix} + \underbrace{\begin{bmatrix}* & *\\ * & *\\ * & *\\ * & *\\ * & *\end{bmatrix}}_{B_{LAT}\ 5\times2}\begin{bmatrix}\bar\delta_a\\ \bar\delta_r\end{bmatrix}$$

Decoupled Lateral Linear Model

Longitudinal State-space Equations

- We will solve for linearized state-space models (A & B) **numerically**
- But, it is informative to derive them *analytically*
  - Very cumbersome!
  - Example derivations for two components of longitudinal model: (actually derived on next slide)

11th equation in 12 EoMs

$$\dot{q} = \frac{J_z - J_x}{J_y} pr - \frac{J_{xz}}{J_y}\left(p^2 - r^2\right) + \frac{\rho V_a^2 S c}{2}\frac{1}{J_y}\left\{C_{mo} + C_{m\alpha}\alpha + \frac{c}{2V_a}C_{mq}q + C_{m\delta e}\delta_e\right\}$$

$$q = q^* + \bar{q}$$

$$\dot{\bar{q}} \approx \frac{\partial \dot{q}}{\partial u}\bar{u} + \frac{\partial \dot{q}}{\partial w}\bar{w} + \frac{\partial \dot{q}}{\partial q}\bar{q} + \frac{\partial \dot{q}}{\partial \theta}\bar{\theta} + \frac{\partial \dot{q}}{\partial p_d}\bar{p}_d + \frac{\partial \dot{q}}{\partial \delta_e}\bar{\delta}_e + \left\{\begin{array}{c}\text{Less dominant}\\ \text{lateral terms, etc.}\end{array}\right\}$$

$$\boxed{\frac{\partial \dot{q}}{\partial \delta_e} = \frac{\rho V_a^{*2} S c C_{m_{\delta_e}}}{2J_y}}$$

$$\boxed{\frac{\partial \dot{q}}{\partial w} = \frac{w^* \rho S c}{J_y}\left[C_{m_0} + C_{m_\alpha}\alpha^* + C_{m_{\delta_e}}\delta_e^*\right] + \frac{\rho S c C_{m_\alpha} u^*}{2J_y} + \frac{\rho S c^2 C_{m_q} q^* w^*}{4J_y V_a^*}}$$

$$\begin{bmatrix}\dot{\bar{u}}\\ \dot{\bar{w}}\\ \dot{\bar{q}}\\ \dot{\bar{\theta}}\\ \dot{\bar{p}}_d\end{bmatrix} = \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{A_{LON}\ 5\text{x}5}\begin{bmatrix}\bar{u}\\ \bar{w}\\ \bar{q}\\ \bar{\theta}\\ \bar{p}_d\end{bmatrix} + \underbrace{\begin{bmatrix} * & * \\ * & * \\ * & * \\ * & * \\ * & * \end{bmatrix}}_{B_{LON}\ 5\text{x}2}\begin{bmatrix}\bar{\delta}_e\\ \bar{\delta}_t\end{bmatrix}$$

Decoupled Longitudinal Linear Model

Note: Star notations (*) represent trimmed state values

Longitudinal State-space Equations

11th equation in 12 EoMs

$$\dot{q} = \frac{J_z - J_x}{J_y} pr - \frac{J_{xz}}{J_y}\left(p^2 - r^2\right) + \frac{\rho V_a^2 Sc}{2}\frac{1}{J_y}\left\{C_{mo} + C_{m\alpha}\alpha + \frac{c}{2V_a}C_{mq}q + C_{m\delta e}\delta_e\right\}$$

```
%% Define symbolic variables
syms u w alpha p q r g rho S m Va c de Jx Jy Jz Jxz Cm_0 Cm_alpha Cm_q Cm_de

%% Define qdot, and replace Va and alpha terms
qdot = (Jz-Jx)/Jy*p*r-Jxz/Jy*(p^2-r^2)+rho*Va^2*S*c/2/Jy*(Cm_0+Cm_alpha*alpha+c/2/Va*Cm_q*q+Cm_de*de);
qdot = subs(qdot, alpha, atan(w,u));
qdot = subs(qdot, Va, sqrt(u^2+w^2))


%% Derive: d(qdot)/d(de), Deriv. of qdot wrt de
d_qdot_dde = diff(simplify(qdot),de)
%  (Cm_de*S*c*rho*(u^2 + w^2))/(2*Jy)
d_qdot_dde = subs(d_qdot_dde, u^2+w^2, Va^2)
%  (Cm_de*S*Va^2*c*rho)/(2*Jy)


%% Derive: d(qdot)/d(w), Deriv. of qdot wrt w
d_qdot_dw = diff(simplify(qdot),w)
%  Long equation consisting of imag(u), real(u), etc.
d_qdot_dw = subs(d_qdot_dw, imag(w), 0);
d_qdot_dw = subs(d_qdot_dw, imag(u), 0);
d_qdot_dw = subs(d_qdot_dw, real(w), w);
d_qdot_dw = subs(d_qdot_dw, real(u), u);
d_qdot_dw = simplify(d_qdot_dw);
d_qdot_dw = subs(d_qdot_dw, (u^2+w^2)^(1/2), Va);
d_qdot_dw = subs(d_qdot_dw, atan2(w,u), alpha);
d_qdot_dw = simplify(d_qdot_dw)
% (S*c*rho*(2*Cm_alpha*Va*u + 4*Cm_0*Va*w
%   + 4*Cm_alpha*Va*alpha*w + 4*Cm_de*Va*de*w + Cm_q*c*q*w))/(4*Jy*Va)
```

Re-arrange & replace state variables with trimmed * notation:

$$\boxed{\frac{\partial \dot{q}}{\partial \delta_e} = \frac{\rho {V_a^*}^2 S c C_{m_{\delta_e}}}{2J_y}}$$

$$\boxed{\frac{\partial \dot{q}}{\partial w} = \frac{w^* \rho S c}{J_y}\left[C_{m_0} + C_{m_\alpha}\alpha^* + C_{m_{\delta_e}}\delta_e^*\right] + \frac{\rho S c C_{m_\alpha} u^*}{2J_y} + \frac{\rho S c^2 C_{m_q} q^* w^*}{4 J_y V_a^*}}$$

Longitudinal State-space Equations

If solved analytically, the de-coupled longitudinal state space model is:

$$\begin{pmatrix} \dot{\bar{u}} \\ \dot{\bar{w}} \\ \dot{\bar{q}} \\ \dot{\bar{\theta}} \\ \dot{\bar{p}}_d \end{pmatrix} = \begin{pmatrix} X_u & X_w & X_q & -g\cos\theta^* & 0 \\ Z_u & Z_w & Z_q & -g\sin\theta^* & 0 \\ M_u & M_w & M_q & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -\sin\theta^* & \cos\theta^* & 0 & -u^*\cos\theta^* - w^*\sin\theta^* & 0 \end{pmatrix} \begin{pmatrix} \bar{u} \\ \bar{w} \\ \bar{q} \\ \bar{\theta} \\ \bar{p}_d \end{pmatrix} + \begin{pmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & 0 \\ M_{\delta_e} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{\delta}_e \\ \bar{\delta}_t \end{pmatrix}$$

Modifications due to new propeller model

Longitudinal	Formula
X_u	$\frac{u^*\rho S}{\mathrm{m}}\left[C_{X_0} + C_{X_\alpha}\alpha^* + C_{X_{\delta_e}}\delta_e^*\right] - \frac{\rho S w^* C_{X_\alpha}}{2\mathrm{m}} + \frac{\rho S c C_{X_q} u^* q^*}{4 m V_a^*} - \frac{\rho S_{\mathrm{prop}} C_{\mathrm{prop}} u^*}{\mathrm{m}} \delta_t^* \left[1 + \left(1 - \frac{k_{motor}}{V_a^*}\right)\left(1 - 2\delta_t^*\right)\right]$
X_w	$-q^* + \frac{w^*\rho S}{\mathrm{m}}\left[C_{X_0} + C_{X_\alpha}\alpha^* + C_{X_{\delta_e}}\delta_e^*\right] + \frac{\rho S c C_{X_q} w^* q^*}{4 m V_a^*} + \frac{\rho S C_{X_\alpha} u^*}{2\mathrm{m}} - \frac{\rho S_{\mathrm{prop}} C_{\mathrm{prop}} w^*}{\mathrm{m}} \delta_t^* \left[1 + \left(1 - \frac{k_{motor}}{V_a^*}\right)\left(1 - 2\delta_t^*\right)\right]$
X_q	$-w^* + \frac{\rho V_a^* S C_{X_q} c}{4\mathrm{m}}$
X_{δ_e}	$\frac{\rho V_a^{*2} S C_{X_{\delta_e}}}{2\mathrm{m}}$
X_{δ_t}	$\frac{\rho S_{prop} C_{prop}}{\mathrm{m}}\left(k_{motor} - V_a^*\right)\left(2\delta_t^* k_{motor} + V_a^* - 2V_a^*\delta_t^*\right)$
Z_u	$q^* + \frac{u^*\rho S}{\mathrm{m}}\left[C_{Z_0} + C_{Z_\alpha}\alpha^* + C_{Z_{\delta_e}}\delta_e^*\right] - \frac{\rho S C_{Z_\alpha} w^*}{2\mathrm{m}} + \frac{u^*\rho S C_{Z_q} c q^*}{4 m V_a^*}$
Z_w	$\frac{w^*\rho S}{\mathrm{m}}\left[C_{Z_0} + C_{Z_\alpha}\alpha^* + C_{Z_{\delta_e}}\delta_e^*\right] + \frac{\rho S C_{Z_\alpha} u^*}{2\mathrm{m}} + \frac{\rho w^* S c C_{Z_q} q^*}{4 m V_a^*}$
Z_q	$u^* + \frac{\rho V_a^* S C_{Z_q} c}{4\mathrm{m}}$
Z_{δ_e}	$\frac{\rho V_a^{*2} S C_{Z_{\delta_e}}}{2\mathrm{m}}$
M_u	$\frac{u^*\rho S c}{J_y}\left[C_{m_0} + C_{m_\alpha}\alpha^* + C_{m_{\delta_e}}\delta_e^*\right] - \frac{\rho S c C_{m_\alpha} w^*}{2 J_y} + \frac{\rho S c^2 C_{m_q} q^* u^*}{4 J_y V_a^*}$
M_w	$\frac{w^*\rho S c}{J_y}\left[C_{m_0} + C_{m_\alpha}\alpha^* + C_{m_{\delta_e}}\delta_e^*\right] + \frac{\rho S c C_{m_\alpha} u^*}{2 J_y} + \frac{\rho S c^2 C_{m_q} q^* w^*}{4 J_y V_a^*}$
M_q	$\frac{\rho V_a^* S c^2 C_{m_q}}{4 J_y}$
M_{δ_e}	$\frac{\rho V_a^{*2} S c C_{m_{\delta_e}}}{2 J_y}$

Derived on previous slide

See Book for derivation and nomenclature. (Note: Book uses *h* instead of p_d.)

Lateral State-space Equations

If solved analytically, the de-coupled lateral state space model is:

$$\begin{pmatrix}\dot{\bar{v}}\\ \dot{\bar{p}}\\ \dot{\bar{r}}\\ \dot{\bar{\phi}}\\ \dot{\bar{\psi}}\end{pmatrix} = \begin{pmatrix} Y_v & Y_p & Y_r & g\cos\theta^*\cos\phi^* & 0\\ L_v & L_p & L_r & 0 & 0\\ N_v & N_p & N_r & 0 & 0\\ 0 & 1 & \cos\phi^*\tan\theta^* & q^*\cos\phi^*\tan\theta^* - r^*\sin\phi^*\tan\theta^* & 0\\ 0 & 0 & \cos\phi^*\sec\theta^* & p^*\cos\phi^*\sec\theta^* - r^*\sin\phi^*\sec\theta^* & 0\end{pmatrix}\begin{pmatrix}\bar{v}\\ \bar{p}\\ \bar{r}\\ \bar{\phi}\\ \bar{\psi}\end{pmatrix} + \begin{pmatrix} Y_{\delta_a} & Y_{\delta_r}\\ L_{\delta_a} & L_{\delta_r}\\ N_{\delta_a} & N_{\delta_r}\\ 0 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}\bar{\delta}_a\\ \bar{\delta}_r\end{pmatrix}$$

Lateral	Formula
Y_v	$\frac{\rho S b v^*}{4mV_a^*}\left[C_{Y_p}p^* + C_{Y_r}r^*\right] + \frac{\rho S v^*}{\mathrm{m}}\left[C_{Y_0} + C_{Y_\beta}\beta^* + C_{Y_{\delta_a}}\delta_a^* + C_{Y_{\delta_r}}\delta_r^*\right] + \frac{\rho S C_{Y_\beta}}{2\mathrm{m}}\sqrt{u^{*2}+w^{*2}}$
Y_p	$w^* + \frac{\rho V_a^* S b}{4m}C_{Y_p}$
Y_r	$-u^* + \frac{\rho V_a^* S b}{4m}C_{Y_r}$
Y_{δ_a}	$\frac{\rho V_a^{*2} S}{2\mathrm{m}}C_{Y_{\delta_a}}$
Y_{δ_r}	$\frac{\rho V_a^{*2} S}{2\mathrm{m}}C_{Y_{\delta_r}}$
L_v	$\frac{\rho S b^2 v^*}{4V_a^*}\left[C_{p_p}p^* + C_{p_r}r^*\right] + \rho S b v^*\left[C_{p_0} + C_{p_\beta}\beta^* + C_{p_{\delta_a}}\delta_a^* + C_{p_{\delta_r}}\delta_r^*\right] + \frac{\rho S b C_{p_\beta}}{2}\sqrt{u^{*2}+w^{*2}}$
L_p	$\Gamma_1 q^* + \frac{\rho V_a^* S b^2}{4}C_{p_p}$
L_r	$-\Gamma_2 q^* + \frac{\rho V_a^* S b^2}{4}C_{p_r}$
L_{δ_a}	$\frac{\rho V_a^{*2} S b}{2}C_{p_{\delta_a}}$
L_{δ_r}	$\frac{\rho V_a^{*2} S b}{2}C_{p_{\delta_r}}$
N_v	$\frac{\rho S b^2 v^*}{4V_a^*}\left[C_{r_p}p^* + C_{r_r}r^*\right] + \rho S b v^*\left[C_{r_0} + C_{r_\beta}\beta^* + C_{r_{\delta_a}}\delta_a^* + C_{r_{\delta_r}}\delta_r^*\right] + \frac{\rho S b C_{r_\beta}}{2}\sqrt{u^{*2}+w^{*2}}$
N_p	$\Gamma_7 q^* + \frac{\rho V_a^* S b^2}{4}C_{r_p}$
N_r	$-\Gamma_1 q^* + \frac{\rho V_a^* S b^2}{4}C_{r_r}$
N_{δ_a}	$\frac{\rho V_a^{*2} S b}{2}C_{r_{\delta_a}}$
N_{δ_r}	$\frac{\rho V_a^{*2} S b}{2}C_{r_{\delta_r}}$

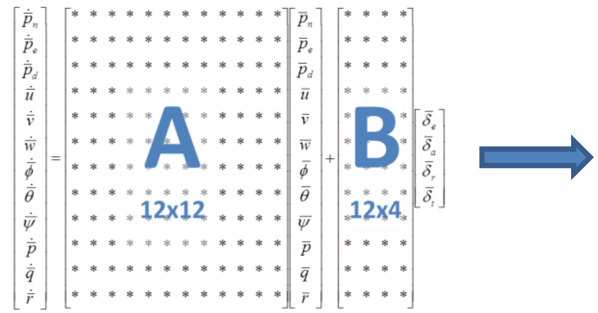
See Book for derivation and nomenclature.

Full Linear Aircraft Response Models

- The state space models provide an efficient means of representing aircraft dynamics (near the trim point)

$$\dot{\bar{\mathbf{x}}} = A\bar{\mathbf{x}} + B\bar{\mathbf{u}}$$

(12 states, 4 inputs)



H(s):
12x4 matrix of Transfer Functions from 4 control deflections to 12 states

- We can gain more insight by converting the state space models into Laplace transfer functions
- Using the full 12-state, 4-input model, a multi-input, multi-state model can be converted to Laplace transfer functions via:

$$H(s) = (sI - A)^{-1} B$$

- The result is a 12x4 matrix of Laplace transfer functions
  - $H_{i,j}(s)$ is the transfer function from the j^{th} input to the i^{th} state
- In Matlab, this is accomplished via:

```
s=tf('s');              % Create a Laplace s variable
H = inv(s*eye(12)-A)*B; % Convert to Laplace transfer functions
H = minreal(H);         % Use min. realization (i.e. cancel identical poles and zeros)
H = zpk(H);             % Convert from polynomial num and den to zero/pole form
H(i,j)                  % Display the TF from the jth input to the ith output
```

Reduced Order Modes

In traditional literature, aerodynamicists have defined several open-loop aircraft dynamics modes. We can use the transfer functions from the state-space models to investigate these modes.

Longitudinal Modes
- Short Period Mode
- Phugoid Mode

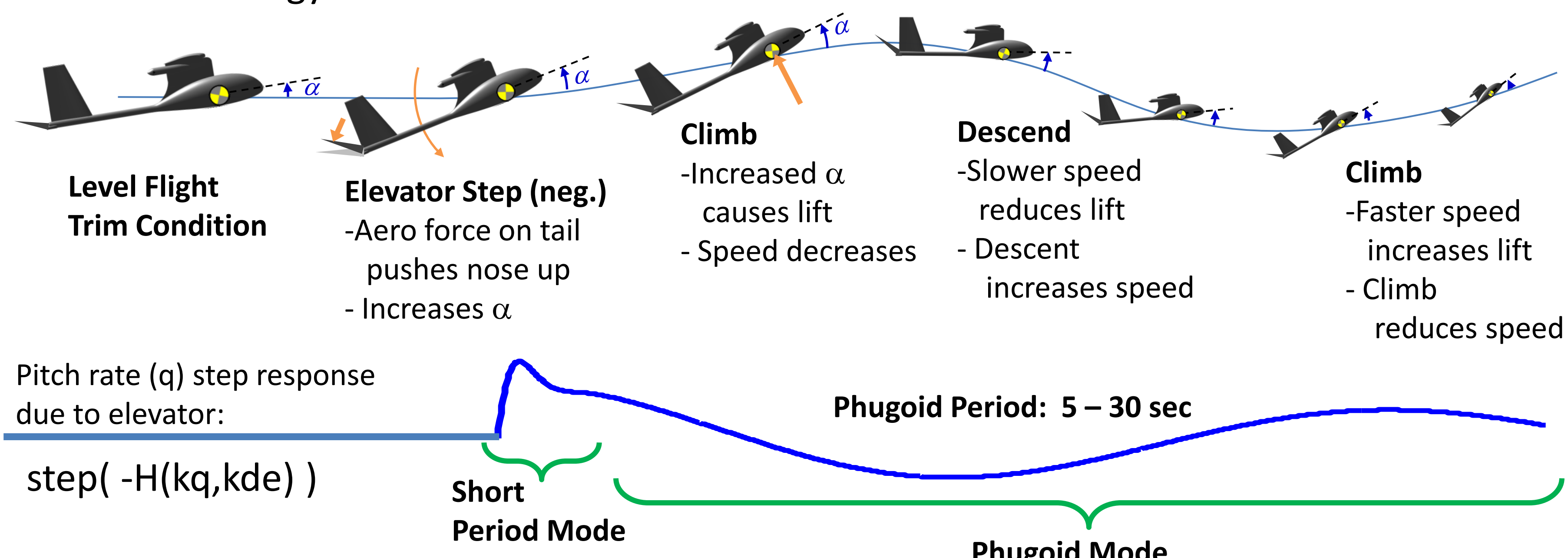
Lateral Modes
- Roll Mode
- Spiral Mode
- Dutch-roll Mode

Aircraft Longitudinal Modes

- Elevator motion induces two separate oscillatory modes in longitudinal states (u, w, q, θ, p_d), e.g.:

$$H(k_q, k_{\delta e}) = \frac{numerator}{\left(s^2 + 2\zeta_{phugoid}\omega_{n,phugoid}s + \omega_{n,phugoid}^2\right)\left(s^2 + 2\zeta_{short}\omega_{n,short}s + \omega_{n,short}^2\right)} \qquad \omega_{n,phugoid} << \omega_{n,short}$$

- <u>**Short Period Mode**</u>: Fast, damped mode ($\omega_{n,short}$, ζ_{short}) causing quick pitch response to an elevator step
- <u>**Phugoid Mode:**</u> Slow, lightly damped mode ($\omega_{n,phugoid}$, $\zeta_{phugoid}$) resulting from speed & altitude energy transfer



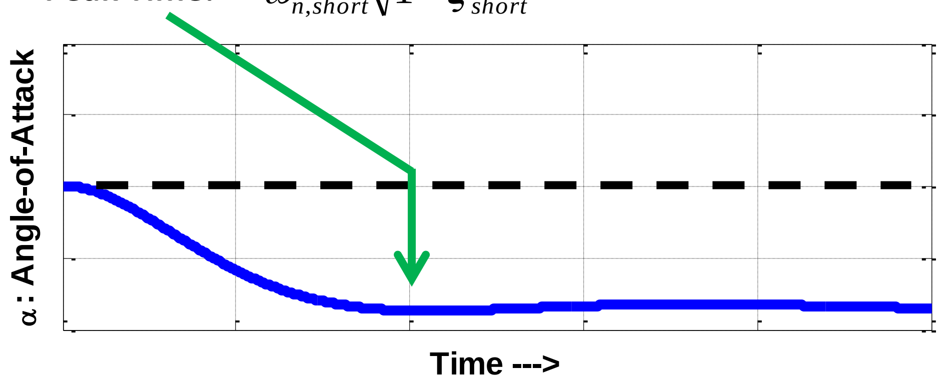
Aircraft Longitudinal Mode: Short Period

- **<u>Short Period Mode</u>**: Fast, damped mode ($\omega_{n,short}$, ζ_{short}) causing quick pitch response to an elevator step

  -Changes Angle-of-Attack quickly

  - Positive elevator causes negative Angle-of-Attack

Short Period Peak Time: $$\frac{\pi}{\omega_{n,short}\sqrt{1-\zeta_{short}^2}} \leq 1\sec$$

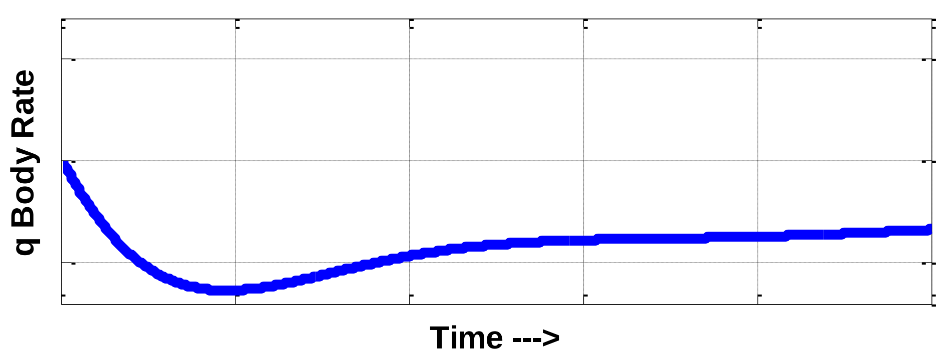


$$w = V_a \sin\alpha\cos\beta \approx V_a \sin\alpha$$

$$\overline{w} = \tfrac{\partial w}{\partial \alpha}\overline{\alpha} \approx V_a^* \cos\alpha^* \overline{\alpha}$$

$$\overline{\alpha} \approx \frac{\overline{w}}{V_a^* \cos\alpha^*}$$

step(H(kw,kde)/P.Va0/cos(P.alpha0))

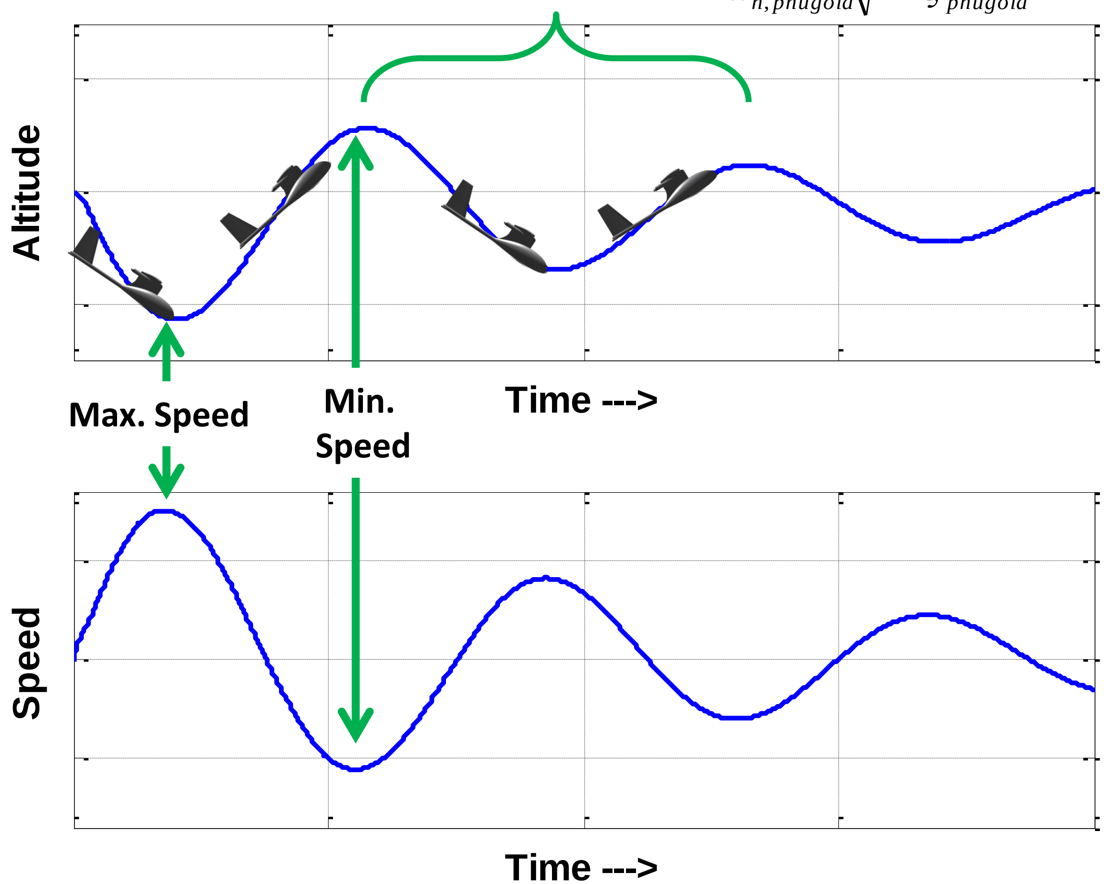


step(H(kq,kde))

Aircraft Longitudinal Mode: Phugoid

- **Phugoid Mode:** Slow, lightly damped mode ($\omega_{n,phugoid}$, $\zeta_{phugoid}$) resulting from speed & altitude energy transfer
  - Named after Greek word "to fly" (actually means "to flee")

Phugoid Period = $\dfrac{2\pi}{\omega_{n,phugoid}\sqrt{1-\zeta^2_{phugoid}}}$ (Proportional to speed, 5-30 sec)



$\bar{h} = -\bar{p}_d$

impulse(-pi/180*H(kpd,kde))

Note:

H(kpd,kde): m/rad

pi/180*H(kpd,kde): m/deg

impulse(pi/180*H(ku,kde))

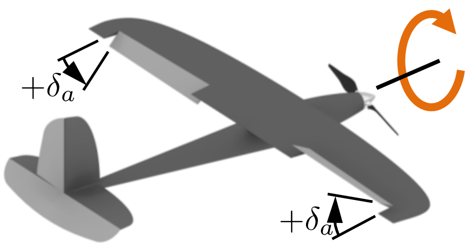
Aircraft Lateral Modes

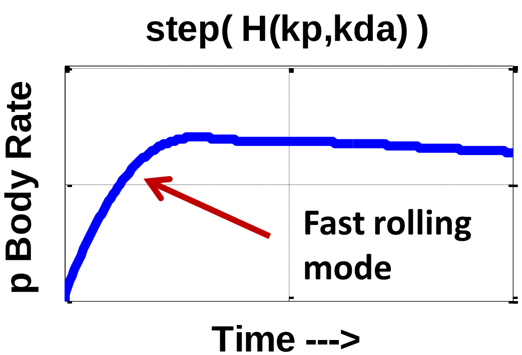
Aileron/rudder motion induces three modes in lateral states (v, p, r, ϕ, ψ)

$$H(k_r, k_{\delta r}) = \frac{numerator}{(s-\lambda_{rolling})(s-\lambda_{spiral})(s^2 + 2\zeta_{dutch}\omega_{n,dutch}s + \omega_{n,dutch}^2)}$$

<u>Rolling Mode</u>: First order, fast stable mode between aileron and roll rate

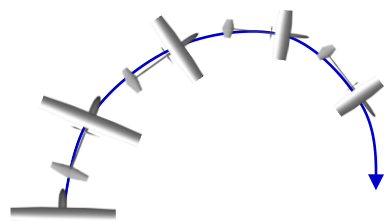
$$\lambda_{rolling} < 0 \text{ (stable)}$$





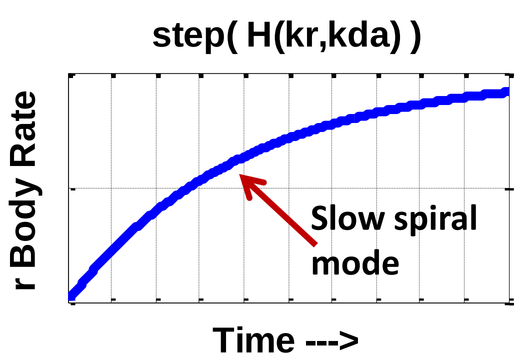
Short Time Scale (~2 sec)

<u>Spiral Mode:</u> First order, *really slow* mode between aileron/rudder and yaw/course. λ_{spiral} is small



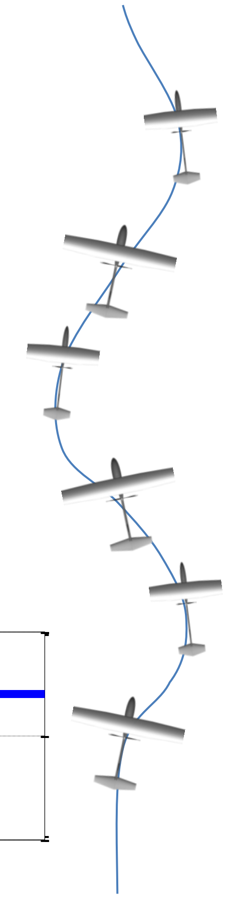
$\lambda_{spiral} < 0:$ Slowly stable

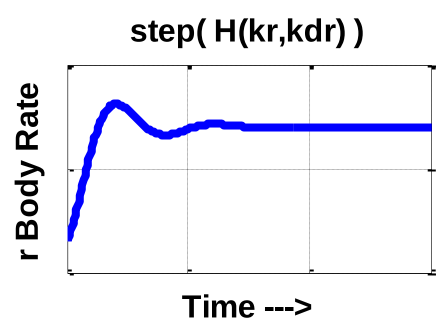
$\lambda_{spiral} > 0:$ Slowly unstable



Very Long Time Scale (~1+ min)

<u>Dutch Roll Mode:</u> Second order mode, coupling between yaw, roll and sideslip. Like a duck wagging its tail.





Fairly Short Time Scale (~3 sec)

Lecture 5 Homework, 1/5

1) Implement propeller forces and moments in UAV 6DOF and combine with others (from previous homework). Print out uavsim_forces_moments.m.

Check: uavsim_forces_moments(ones(20,1),P)' = [2.1301 6.9345 4.4475 0.0422 -0.0678 -0.0718]

2) Develop a longitudinal trim routine, and trim for P.Va0=13 m/s

Modify compute_longitudinal_trim() appropriately. Note, compute_longitudinal_trim() uses a subfunction called cost_function(), which calls uavsim_forces_moments() and returns a cost. The routine uses fminsearch() to find the [α, δ_e, δ_t] which minimizes the cost, J_{cost}. The resulting trimmed states and control surfaces are used as the initial values (e.g. P.w0, P.theta0, P.delta_e0, etc.)

Uncomment "P=compute_longitudinal_trim(P);" in "load_uavsim.m", and re-run "load_uavsim.m" (Leave the line uncommented from now on!)

A) Turn in the resulting longitudinal trim values [α^*, δ_e^*, δ_t^*], in degrees for the 2 angles

B) By default, uavsim_control() outputs [P.delta_e0 P.delta_a0 P.delta_r0 P.delta_t0], which were set in the trim routine. Run uavsim. Is the UAV trimmed? Describe. (Note: For convenience, the propeller torque parameter has been zeroed.)

3) A trim routine can be used to determine idealized capabilities. If throttle must stay between 0 and 1, elevator must stay between -45deg and 45deg, and alpha must stay between -30deg and 30deg, what are the minimum and maximum valid airspeeds (to the nearest 0.1 m/s) of this vehicle (i.e. run trim routine for various P.Va0 values around the nominal of P.Va0=13). Run uavsim at these min and max speeds. Does it fly trimmed? What angle-of-attack and throttle settings are necessary at each of these extreme speeds to maintain lift and counteract drag?

4) After a fresh "load_uavsim", *temporarily* add some propeller counter-torque, P.k_Tp = 5e-6. What are the effects? Determine (via trial-and-error to the nearest 0.1°) an aileron that mostly nullifies the rolling moment. (UAV will still have some sideslip and yaw rate. A combination of aileron and rudder would be needed to nullify both rolling moment and sideslip.)
Return P.k_Tp to zero afterward.

Lecture 5 Homework, 2/5

Re-run load_uavsim!
(Should start from trim)

5) Develop a routine to make the linearized A and B matrices

– Modify linearize_uavsim.m appropriately. Note, linearize_uavsim() has a subfunction:
  xdot = eval_forces_moments_kin_dyn(x,deltas,P)
  This subroutine calls both uavsim_forces_moments() and uavsim_kin_dyn() to generate xdot as a function of x and the control deflections.

A) Run "[A B]=linearize_uavsim(P)" using the trimmed P found in Problem #2 (with P.Va0=13 m/s). Extract and print out the longitudinal and lateral submatrices (A_lon, B_lon, A_lat, B_lat). Do your submatrices have the same form described in notes?

B) Do some spot checks to verify correctness (see slides for reference):

Verity that A_lon(1,4) = $-g \cos \theta^*$
Verify that A_lon(3,3) = M_q
Verify that B_lon(3,1) = M_de

Note: use the zpk transfer functions to determine the poles, natural frequencies and dampings.
DO NOT use the book's analytical methods (Section 5.6).

C) Print out the zpk form of the transfer function from delta_e to q.
What are the phugoid and short period natural frequencies and dampings?

D) Print out the zpk form of the transfer function from delta_r to r.
What are the rolling and spiral eigenvalues (poles)? What are the dutch roll natural frequency and damping? (Remember that stable poles are negative. e.g. for G(s)=3/(s+4), the pole is -4, not +4.)

Lecture 5 Homework, 3/5

Re-run load_uavsim!
(Should start from trim)

6) Compare linear longitudinal response with uavsim

- A) Modify uavsim_control() to perform a 0.001 deg elevator step response:

  ```
  delta_e=P.delta_e0 + .001*pi/180;
  ```

  Run uavsim starting with the P.Va0=13 m/s trim condition for 20 seconds and compare the resulting pitch body rate (q) with the linear result:

  ```
  plot( out.time_s, out.q_dps, 'b', …  % uavsim output
        out.time_s, step(0.001*H(kq,kde), out.time_s),'r:'); % linear step
  ```

  Do they match? Highlight both the short period response and the phugoid response.
- B) Estimate the phugoid period from the plot in 5(A). Does the phugoid period match what you expect?
- C) The described modes can result from any deviation from trim, not just control surface changes. <u>Undo the elevator step</u> in uavsim_control(), and temporarily overwrite the initial pitch to 25 deg: P.theta0 = 25*pi/180.
  Run uavsim and describe the result. Specifically, note the interplay between altitude, airspeed, pitch and flight path angle. What aircraft mode are you witnessing?

Lecture 5 Homework, 4/5

Re-run load_uavsim!
(Should start from trim)

7) Compare linear lateral response with uavsim

- A) Modify uavsim_control() to perform a 0.001 deg aileron step response:

  delta_a=P.delta_a0 + .001*pi/180;

  Run uavsim for 5 seconds (starting with the P.Va0=13 m/s trim) and compare resulting roll body rate (p) with linear result:

```
plot( out.time_s, out.p_dps, 'b', ...  % uavsim output
      out.time_s, step(0.001*H(kp,kda), out.time_s),'r:'); % linear step
```

  Do they match? Highlight the rolling mode response. How quickly does roll rate respond to an aileron step (peak time)?
- B) Similarly, perform a 1 deg aileron step and compare p with linear response. Do they still match for the first 10 seconds? If not, explain why. Run the simulation for a long time. What happens to the UAV?
- C) The dutch roll mode can be difficult to see. <u>Undo the aileron step</u> from above and temporarily add some initial sideslip, via setting P.v0=5 m/s from the command line. Run the simulation and plot the first 2 seconds of roll and sideslip, highlighting the dutch roll oscillation. (No need to compare with linear response)

Lecture 5 Homework, 5/5

Undo changes from previous questions, and re-run load_uavsim!

Recommended Reading: 4.3, 4.5, 5.5, 5,6
Notes:
- In 4.3, propeller model in book is incorrect
- In 5.5 & 5.6, book derives state space matrices and aircraft modes analytically. Useful, but unnecessary. Read lightly.
- Homework uses routines in “Later_Use_Files” directory (or otherwise provided). Copy relevant files to uavsim directory.

8) Just for fun, try manually flying the UAV.

- Copy the following from “Later_Use_Files” (unless otherwise provided by instructor):
  - allow_figure_motion.m (Allows Matlab to recognize button-down mouse movements.)
  - joystick.m (Creates a virtual joystick, via allow_figure_motion.m)
- After re-running load_uavsim, set P.manual_flight_flag to 1 and re-run uavsim. (Doing so will overwrite the outputs of uavsim_control.m)
- Use mouse to directly control elevator (up/down) and aileron (right/left) on virtual joystick.
- Use keys ‘s’ and ‘w’ to change throttle.
- Use keys ‘a’ and ‘d’ to change rudder.
- Can you maintain stable flight? Can you do any acrobatics? How long can you fly before either crashing, violating the angle-of-attack limit, or getting bored.
- If uavsim isn’t able to run in near-real-time, try increasing P.Tlog and/or P.Tvis. If that doesn’t work, don’t bother with this problem.