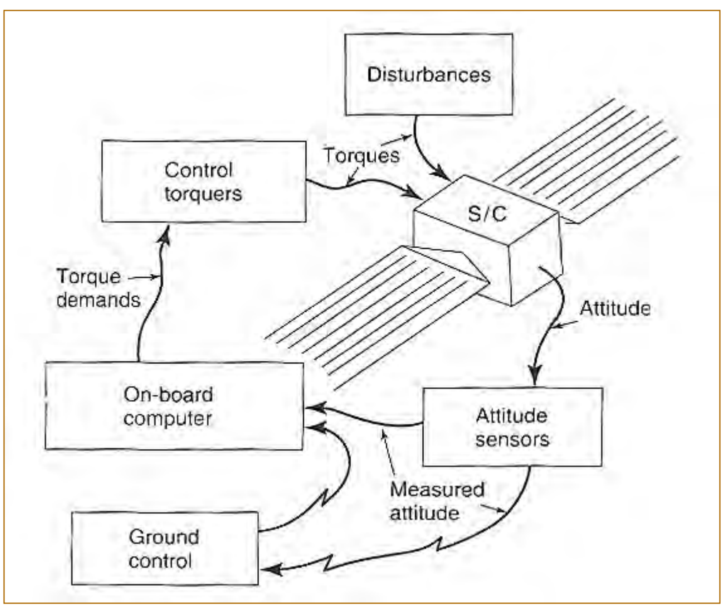
Spacecraft Attitude Control

Space System Design, MAE 342, Princeton University
Robert Stengel

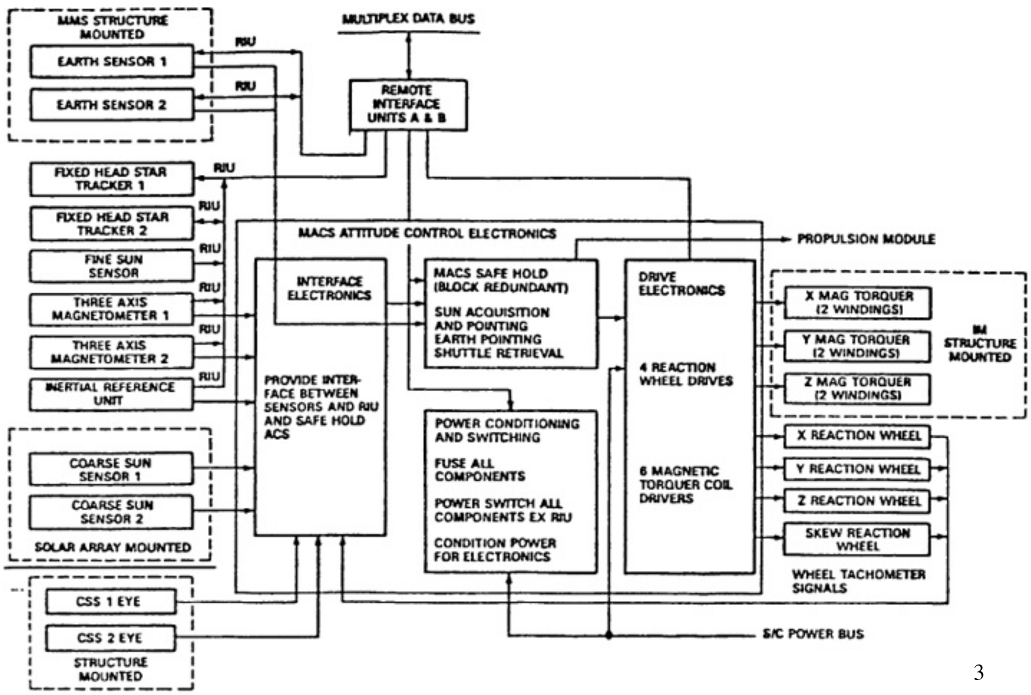
- More on Rotation Matrices
  - Direction cosine matrix
  - Quaternions
- Yo-yo De-Spin
- Continuously Variable Torque Controllers
- On/Off-Torque Controllers

Attitude Control System



Fortescue

UARS Attitude Control System



Spacecraft Attitude Control Inputs

- **On-Board Sensors**
  - Inertial Measurements
    - Accelerometers
    - Angle sensors
    - Angular-rate sensors
  - Optical Sensors
    - Star sensors
    - Sun sensors
    - Horizon sensors
- **Off-Board Observations**
  - Ground-Based Tracking
    - Radar
    - Navigation beacons (VOR/DME, LORAN, …)
  - Spaced-Based Tracking
    - GPS, GLONASS, …

Potential Accuracies of External Attitude Measurements

Reference object	Potential accuracy
Stars	1 arc second
Sun	1 arc minute
Earth (horizon)	6 arc minutes
RF beacon	1 arc minute
Magnetometer	30 arc minutes
Narstar Global Positioning System (GPS)	6 arc minutes

Note: This table gives only a guideline. The GPS estimate depends upon the 'baseline' used (see text).

Fortescue

Spacecraft Attitude Control Outputs

- **Continuous Control Torques**
  - **Control Moment/Reaction Wheel Gyros**
  - **Magnetic Torquers**
  - **Solar Panels**
- **Pulsed Control Torques**
  - **Reaction Control Thrusters (RCS)**
- **One-Shot Devices**
  - **RCS Spin-up**
  - **Yo-Yo De-Spin**

Spacecraft Attitude Disturbances

- **External Torques**
  - Solar radiation pressure
  - Gravity gradient
  - Magnetic fields
  - Aerodynamics
  - Can be put to good use if related to attitude control objectives
- **Vehicle-Based Torques**
  - Mass movement
  - Elasticity
  - Out-gassing

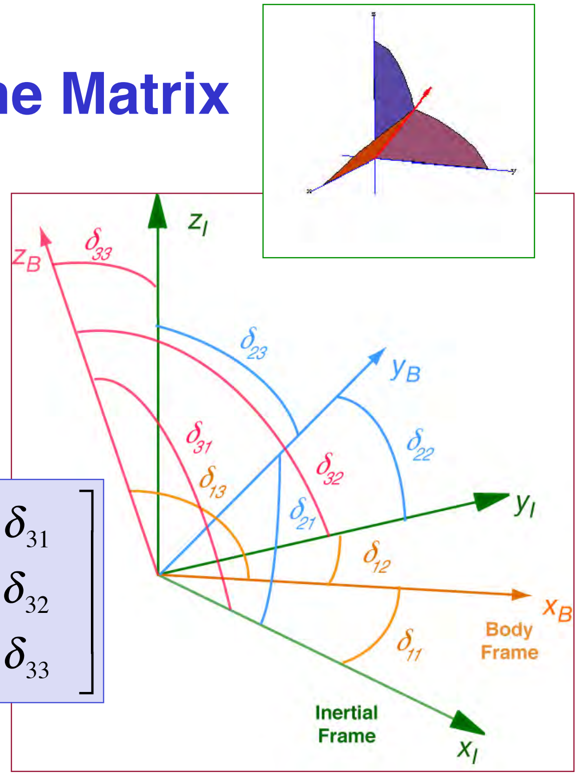
More on Rotation Matrices and Quaternions

Direction Cosine Matrix

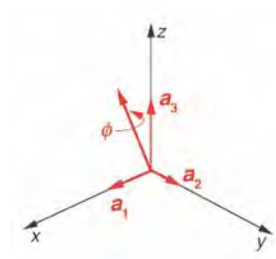
- Cosines of angles between each ***I*** axis and each ***B*** axis
- Projections of vector components in one frame on the other

$$\mathbf{H}_I^B = \begin{bmatrix} \cos\delta_{11} & \cos\delta_{21} & \cos\delta_{31} \\ \cos\delta_{12} & \cos\delta_{22} & \cos\delta_{32} \\ \cos\delta_{13} & \cos\delta_{23} & \cos\delta_{33} \end{bmatrix}$$

$$\mathbf{r}_B = \mathbf{H}_I^B \mathbf{r}_I$$



Euler's Formula



- Rotation from one axis system, ***I***, to another, ***B***, represented by
  - Orientation of axis vector about which the rotation occurs (3 parameters of a unit vector, a_1, a_2, and a_3)
  - Magnitude of the rotation angle, Φ, rad

$$\mathbf{r}_B = \mathbf{H}_I^B \mathbf{r}_I$$
$$= \cos\phi\,\mathbf{r}_I + (1-\cos\phi)\left(\mathbf{a}^T\mathbf{r}_I\right)\mathbf{a} - \sin\phi\left(\mathbf{a}\times\mathbf{r}_I\right)$$

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$$

$$\left(\mathbf{a}^T\mathbf{r}_I\right)\mathbf{a} = \left(\mathbf{a}\mathbf{a}^T\right)\mathbf{r}_I$$

$$\mathbf{H}_I^B = \cos\phi + (1-\cos\phi)\,\mathbf{a}\mathbf{a}^T - \sin\phi\,\tilde{\mathbf{a}}$$

Pisacane, 2005

Quaternion Derived from Euler Rotation Angle and Orientation

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{a}_\phi \\ q_4 \end{bmatrix} = \begin{bmatrix} \sin(\phi/2)\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \\ \cos(\phi/2) \end{bmatrix}$$

- Quaternion vector
  - 4 parameters based on Euler's formula
- Is not singular at $\theta = \pm 90°$
- 4-parameter representation of 3 parameters; hence, it requires a constraint

$$\boxed{\begin{aligned}\mathbf{q}^T\mathbf{q} &= q_1{}^2 + q_2{}^2 + q_3{}^2 + q_4{}^2 \\ &= \sin^2(\phi/2) + \cos^2(\phi/2) = 1\end{aligned}}$$

Rotation Matrix Expressed with Quaternion

From Euler's formula

$$\mathbf{H}_I^B = \left[q_4{}^2 + \left(\mathbf{q}^T\mathbf{q}\right)\right]\mathbf{I}_3 + 2\mathbf{q}\mathbf{q}^T - 2q_4\tilde{\mathbf{q}}$$

$$\mathbf{H}_I^B = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 + -q_2^2 + q_3^2 + q_4^2 \end{bmatrix}$$

Pisacane, 2005

Quaternion Expressed from Elements of Rotation Matrix

$$q_4 = \frac{1}{2}\sqrt{1 + h_{11} + h_{22} + h_{33}}$$

Assuming that $q_4 \neq 0$

$$\mathbf{a}_\phi = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{4q_4}\begin{bmatrix} (h_{23} - h_{32}) \\ (h_{31} - h_{13}) \\ (h_{12} - h_{21}) \end{bmatrix}$$

Pisacane, 2005

Successive Rotations Expressed by Products of Quaternions and Rotation Matrices

Rotation from Frame A to Frame C through Intermediate Frame B

$\mathbf{q}_A^B$: Rotation from A to B

$\mathbf{q}_B^C$: Rotation from B to C

$\mathbf{q}_A^C$: Rotation from A to C

Matrix Multiplication Rule

$$\mathbf{H}_A^C\left(\mathbf{q}_A^C\right) = \mathbf{H}_B^C\left(\mathbf{q}_B^C\right)\mathbf{H}_A^B\left(\mathbf{q}_A^B\right)$$

Quaternion Multiplication Rule

$$\mathbf{q}_A^C = \begin{bmatrix} \mathbf{a}_\phi \\ q_4 \end{bmatrix}_A^C = \mathbf{q}_B^C\mathbf{q}_A^B \triangleq \begin{bmatrix} (q_4)_B^C\mathbf{q}_A^B + (q_4)_A^B\mathbf{q}_B^C - \tilde{\mathbf{q}}_B^C\mathbf{q}_A^B \\ (q_4)_B^C(q_4)_A^B - \left(\mathbf{q}_B^C\right)^T\mathbf{q}_A^B \end{bmatrix}$$

Pisacane, 2005

Quaternion Vector Kinematics

ODE is linear in both q and $\boldsymbol{\omega}_B$

$$\dot{\mathbf{q}} = \frac{d}{dt}\begin{bmatrix} \mathbf{a}_\phi \\ q_4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_4\boldsymbol{\omega}_B - \tilde{\boldsymbol{\omega}}_B\mathbf{a}_\phi \\ \text{---------------} \\ -\boldsymbol{\omega}_B{}^T\mathbf{a}_\phi \end{bmatrix}$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}_B \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

Pisacane, 2005

Propagate Quaternion Vector

$$\frac{d\mathbf{q}(t)}{dt} = \begin{bmatrix} \dot{q}_1(t) \\ \dot{q}_2(t) \\ \dot{q}_3(t) \\ \dot{q}_4(t) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & \omega_z(t) & -\omega_y(t) & \omega_x(t) \\ -\omega_z(t) & 0 & \omega_x(t) & \omega_y(t) \\ \omega_y(t) & -\omega_x(t) & 0 & \omega_z(t) \\ -\omega_x(t) & -\omega_y(t) & -\omega_z(t) & 0 \end{bmatrix}_B \begin{bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \end{bmatrix}$$

Digital integration to compute q(t_k)

$$\mathbf{q}_{\text{int}}(t_k) = \mathbf{q}(t_{k-1}) + \int_{t_{k-1}}^{t_k} \frac{d\mathbf{q}(\tau)}{dt}\, d\tau$$

Normalize q(t_k) to enforce constraint

$$\mathbf{q}(t_k) = \mathbf{q}_{\text{int}}(t_k) \Big/ \sqrt{\mathbf{q}_{\text{int}}{}^T(t_k)\,\mathbf{q}_{\text{int}}(t_k)}$$

Quaternion Interface with Euler Angles

- Quaternion and its kinematics unaffected by Euler angle convention
- Definition of $\mathbf{H}_I^B$ makes the connection
- Specify Euler angle convention (e.g., 1-2-3 or 3-1-3) ; for (1-2-3),

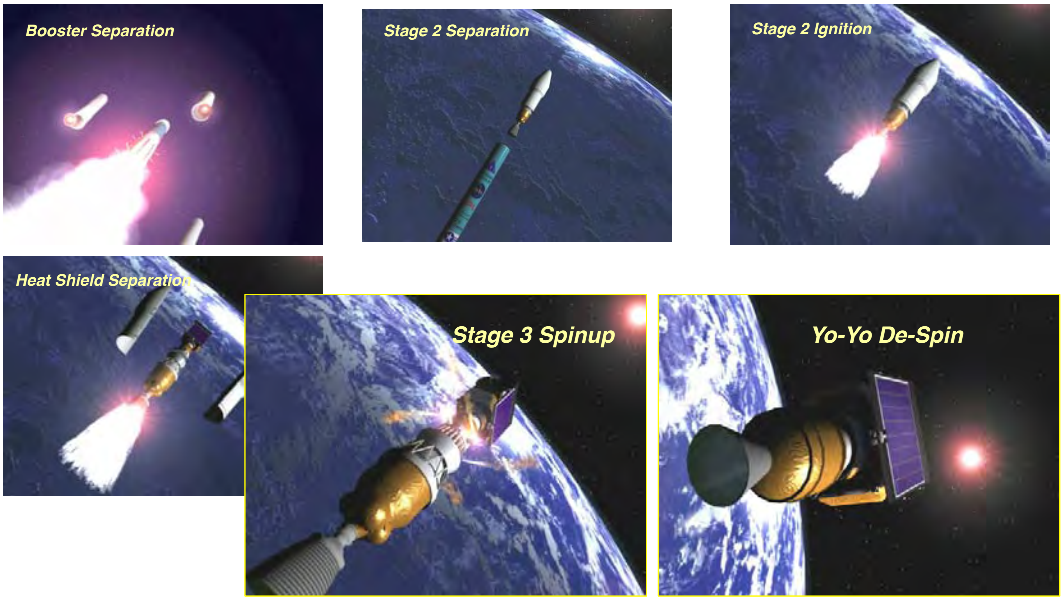
$$\mathbf{H}_I^B = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}_I^B$$

$$= \left[\begin{array}{c|c|c} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \hline -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \hline \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{array}\right]$$

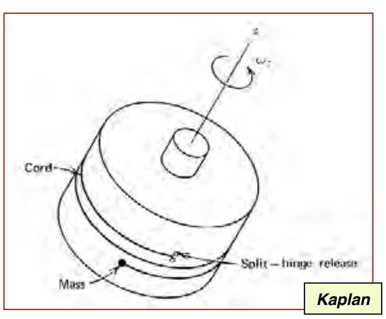
- Apply equations on earlier slide to find $\mathbf{q}(0)$
- Perform trigonometric inversions as indicated to generate [$\Phi(t_k)$, $\theta(t_k)$,$\Psi(t_k)$] from $\mathbf{q}(t_k)$

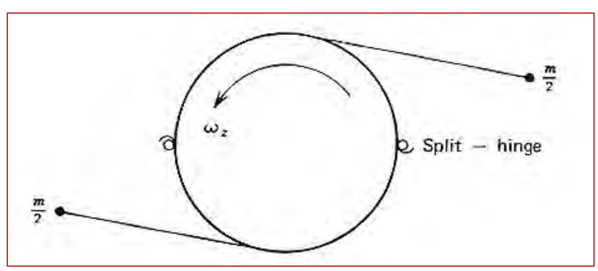
Yo-Yo De-Spin

Mars Odyssey Launch Phases

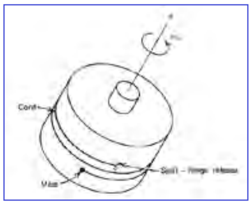


Yo-Yo De-spin

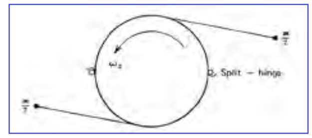




- Satellite is initially spinning at ω_z rad/s
- Angular momentum and rotational energy of satellite plus expendable masses are conserved
- Masses are released, moment of inertia increases, and angular velocity of satellite decreases
- With proper cord length (independent of initial spin rate), satellite is de-spun to zero angular velocity



Yo-Yo De-spin



Angular momentum

$$h_z = I_{zz}\omega_z + mR^2\left[\omega_z + \phi^2\left(\omega_z + \dot{\phi}\right)\right]$$

Rotational energy

$$T = \frac{1}{2}I_{zz}\omega_z^{\,2} + \frac{1}{2}mR^2\left[\omega_z^{\,2} + \phi^2\left(\omega_z + \dot{\phi}\right)^2\right]$$

R = spacecraft radius

l = tether length

$$c = \frac{mR^2 + I_{zz}}{mR^2}$$

m = mass of 2 deployable objects

I_{zz} = satellite moment of inertia

ϕ = angle between split hinge and tangent point

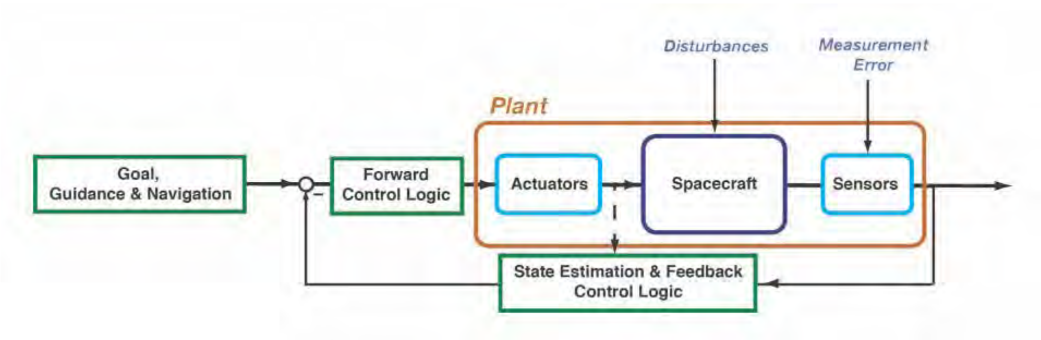
Simultaneous solution for final angular rate

$$\omega_{final} = \omega_{initial}\left(\frac{cR^2 - l^2}{cR^2 + l^2}\right) = 0 \quad if \quad l = R\sqrt{c}$$

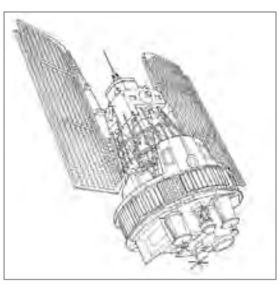
Spaceloft 7 Sounding Rocket De-Spin
https://www.youtube.com/watch?v=5ZqbjQ9ASc8

Continuously Variable Torque Controllers

Overview of Control



Single- or multi-axis interpretation



Single-Axis "Classical" Control of Non-Spinning Spacecraft

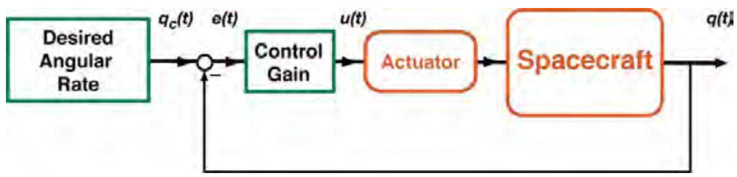
Pitching motion (about the *y* axis) is to be controlled

$$\boxed{\begin{bmatrix} \dot{p}(t) \\ \dot{q}(t) \\ \dot{r}(t) \end{bmatrix} = \begin{bmatrix} M_x(t)/I_{xx} \\ M_y(t)/I_{yy} \\ M_z(t)/I_{zz} \end{bmatrix} - \begin{bmatrix} \left(I_{zz}-I_{yy}\right)q(t)r(t)/I_{xx} \\ \left(I_{xx}-I_{zz}\right)p(t)r(t)/I_{yy} \\ \left(I_{yy}-I_{xx}\right)p(t)q(t)/I_{zz} \end{bmatrix}}$$

- **For motion about the *y* axis only, this reduces to** $\boxed{\dot{q}(t) = M_y(t)/I_{yy}}$
- **Pitch angle equation** $\boxed{\dot{\theta}(t) = q(t)}$

Single-Axis Angular Rate Control of Non-Spinning Spacecraft

- **Small angle and angular rate perturbations**
- **Linear actuator, e.g.,**
  - **Momentum wheel**
- **Linear measurement, e.g.,**
  - **Angular rate gyro**



Simplified Control Law *(C = Control Gain)*

$$\boxed{\begin{aligned} e(t) &= q_c(t) - q(t) \\ u(t) &= C\, e(t) \end{aligned}}$$

Angular Rate Control

$$q(t) = \frac{g_A}{I_{yy}} \int_0^t u(t)\,dt = \frac{C g_A}{I_{yy}} \int_0^t e(t)\,dt = \frac{C g_A}{I_{yy}} \int_0^t \left[q_c - q(t) \right] dt$$

- I_{yy}: moment of inertia
- $q(t)$: angular rate
- $q_c(t)$: desired angular rate
- g_A: actuator gain
- $g_A u(t)$: control torque

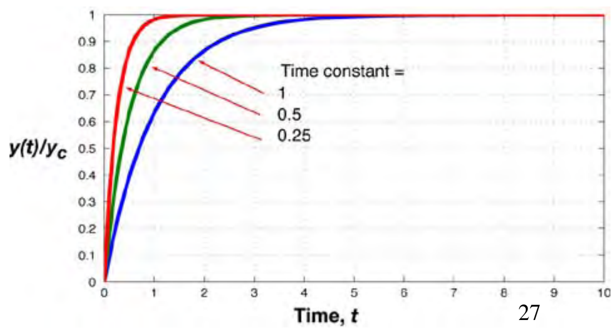
Step Response of Angular Rate Controller

Step input :

$$q_c(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

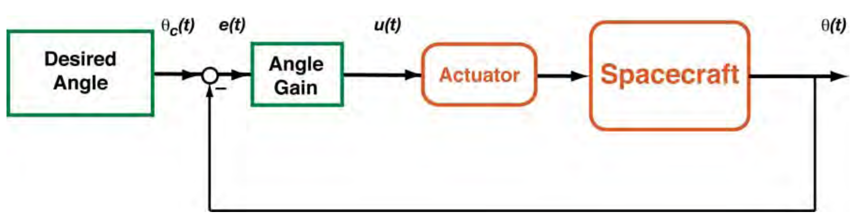
$$q(t) = q_c\left[1 - e^{-\left(\frac{Cg_A}{I_{yy}}\right)t}\right] = q_c\left[1 - e^{\lambda t}\right] = q_c\left[1 - e^{-t/\tau}\right]$$

- where
  - $\lambda = -Cg_A/I_{yy}$ = eigenvalue or root of the system (rad/s)
  - $\tau = I_{yy}/Cg_A$ = time constant of the response (s)



Angle Control of the Spacecraft

- Small angle and angular rate perturbations
- Linear actuator, e.g.,
  - Momentum wheel
- Linear measurement, e.g.,
  - Earth horizon sensor



Angle Control Law *(C = Control Gain)*

$$e(t) = \theta_c(t) - \theta(t)$$
$$u(t) = C\,e(t)$$

Model of Dynamics and Angle Control

- **2^{nd}-order ordinary differential equation**

$$\boxed{\frac{d^2\theta(t)}{dt^2} = \ddot{\theta}(t) = \frac{Cg_A}{I_{yy}}[\theta_c - \theta(t)]}$$

- **Output angle, $\theta(t)$, as a function of time**

$$\boxed{\theta(t) = \frac{g_A}{I_{yy}}\int_0^t\int_0^t u(t)\,dt\,dt = \frac{Cg_A}{I_{yy}}\int_0^t\int_0^t e(t)\,dt\,dt = \frac{Cg_A}{I_{yy}}\int_0^t\int_0^t \left[\theta_c - \theta(t)\right]dt\,dt}$$

Rewrite 2^{nd}-Order Model as Two 1^{st}-Order Equations

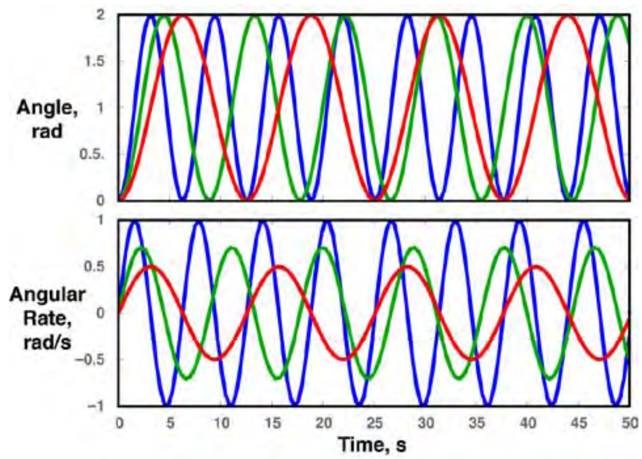
$$\boxed{\begin{aligned} \dot{\theta}(t) &= q(t) \\ \dot{q}(t) &= \frac{Cg_A}{I_{yy}}\left[\theta_c - \theta(t)\right] \end{aligned}}$$

$$\boxed{\begin{bmatrix} \dot{\theta}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \theta(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} 0 \\ g_A / I_{yy} \end{bmatrix} C\left[\theta_c(t) - \theta(t)\right]}$$

$$\boxed{\begin{bmatrix} \dot{\theta}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -Cg_A / I_{yy} & 0 \end{bmatrix}\begin{bmatrix} \theta(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} 0 \\ Cg_A / I_{yy} \end{bmatrix}\theta_c}$$

Simulation of Step Response with Angle Feedback

Objective is to control angle to 1 rad, but solution oscillates about the target

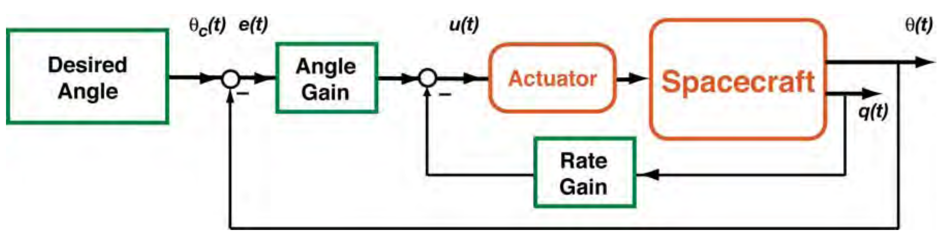


Cg_A/I_{yy} = 1, 0.5, and 0.25

What Went Wrong?

- **No damping!**
- **Solution: Add rate feedback**
- **Control law with rate feedback**

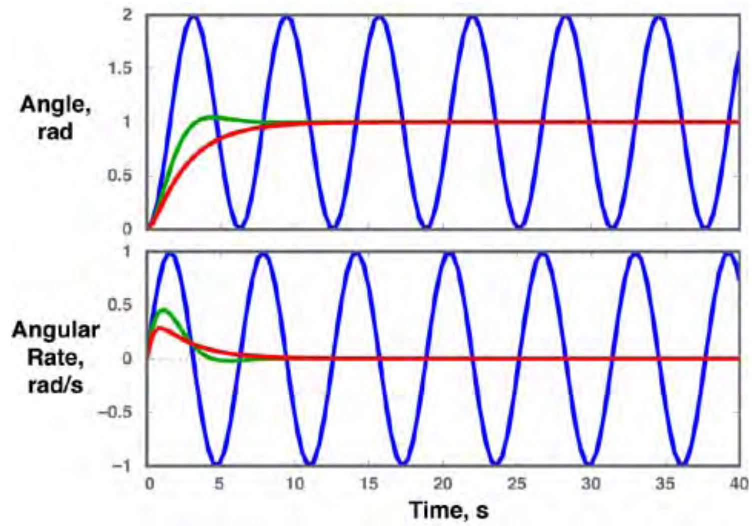
$$u(t) = c_1[\theta_c(t) - \theta(t)] - c_2 q(t)$$



Closed-loop dynamic equation

$$\begin{bmatrix} \dot{\theta}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c_1 g_A / I_{yy} & -c_2 g_A / I_{yy} \end{bmatrix} \begin{bmatrix} \theta(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} 0 \\ c_1 g_A / I_{yy} \end{bmatrix} \theta_c$$

Step Response with Angle and Rate Feedback



$c_1 g_A / I_{yy} = 1$
$c_2 g_A / I_{yy} = 0,\ 1.414,\ 2.828$

2^{nd}-Order Dynamics

Oscillation and damping are induced by linear feedback control

$$\begin{bmatrix} \dot{\theta}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c_1 g_A / I_{yy} & -c_2 g_A / I_{yy} \end{bmatrix} \begin{bmatrix} \theta(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} 0 \\ c_1 g_A / I_{yy} \end{bmatrix} \theta_c$$

$$\begin{bmatrix} \dot{\theta}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n{}^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} \theta(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n{}^2 \end{bmatrix} \theta_c$$

Natural frequency and damping ratio

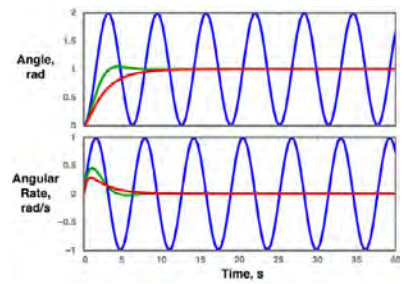
$$\omega_n = \sqrt{c_1 g_A / I_{yy}}$$

$$\zeta = \left(c_2 g_A / I_{yy}\right) / 2\omega_n = c_2 / 2\sqrt{c_1 g_A I_{yy}}$$

Effect of Damping on Eigenvalues, Damping Ratio, and Natural Frequency

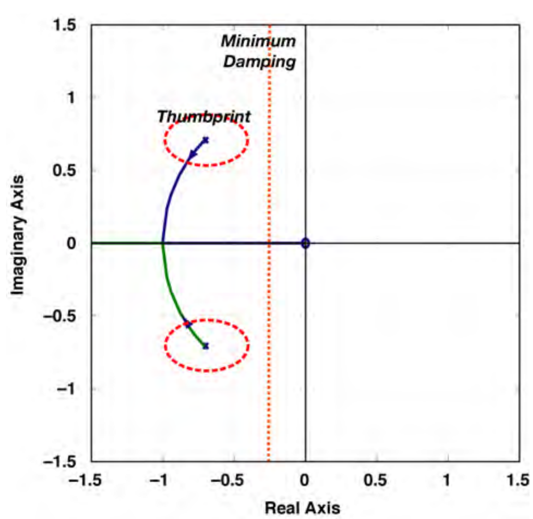
$c_1 g_A / I_{yy} = 1$
$c_2 g_A / I_{yy} = 0, 1.414, 2.828$

Eigenvalues	Damping Ratio, Natural Frequency	
λ_1, λ_2 =	ζ =	ω_n = (rad/s)
0 + 1.0000i 0 - 1.0000i	0	1
-0.7070 + 0.7072i -0.7070 - 0.7072i	0.707	1
-0.4143 -2.4137	*Overdamped*	



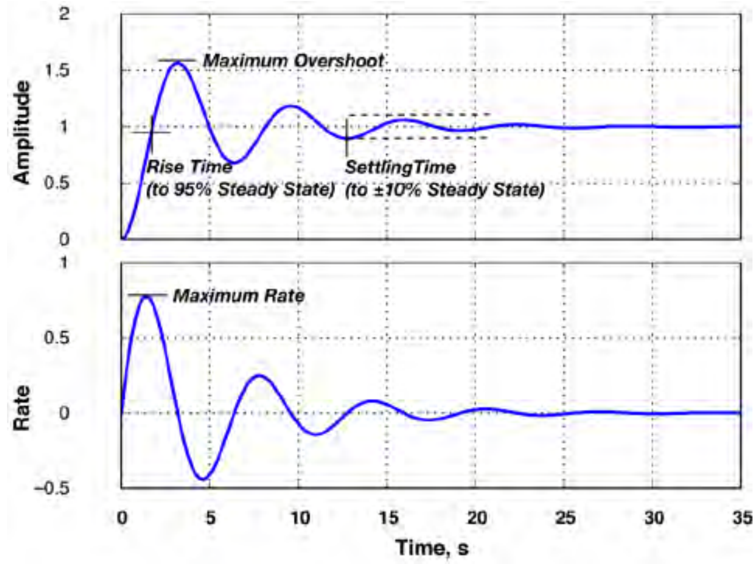
Control System Design to Adjust Roots

Choose control gains to satisfy desirable eigenvalue range



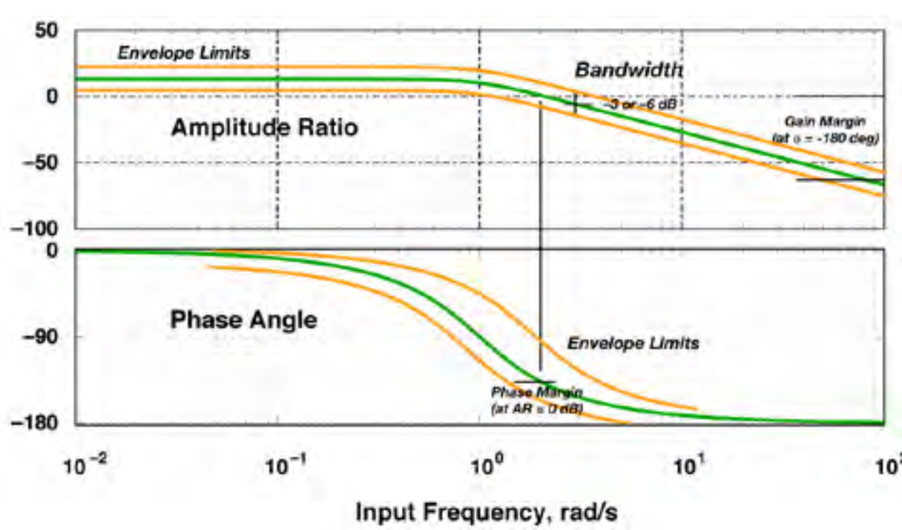
Control System Design to Adjust Transient Response

Choose control gains to satisfy step response criteria



Control System Design to Adjust Frequency Response

Choose control gains to satisfy frequency response criteria



Laplace Transform of the State Vector

Neglecting the initial condition

$$\mathbf{x}(s) = \frac{Adj\left(s\mathbf{I} - \mathbf{F}\right)}{\Delta(s)} \mathbf{G}\,\mathbf{u}(s)$$

Applied to the closed-loop system

$$\begin{bmatrix} \Delta\theta(s) \\ \Delta q(s) \end{bmatrix} = \frac{\begin{bmatrix} c_1 g_A / I_{yy} \\ s c_1 g_A / I_{yy} \end{bmatrix}}{\Delta(s)} \Delta u(s) = \frac{\begin{bmatrix} c_1 g_A / I_{yy} \\ s c_1 g_A / I_{yy} \end{bmatrix} \Delta u(s)}{(s)^2 + \left(c_2 g_A / J\right)(s) + c_1 g_A / J}$$

Frequency Response of the System

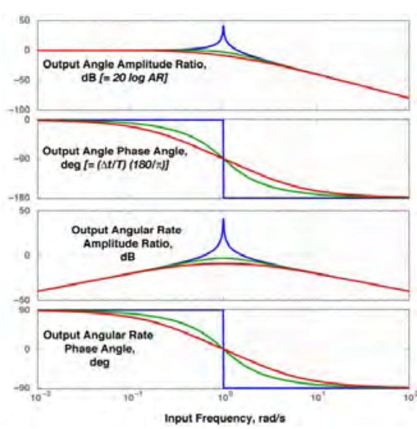
$\sigma = j\omega$

Angle Frequency Response

$$\frac{\Delta\theta(j\omega)}{\Delta u(j\omega)} = \frac{\omega_n^{\,2}}{(j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^{\,2}}$$

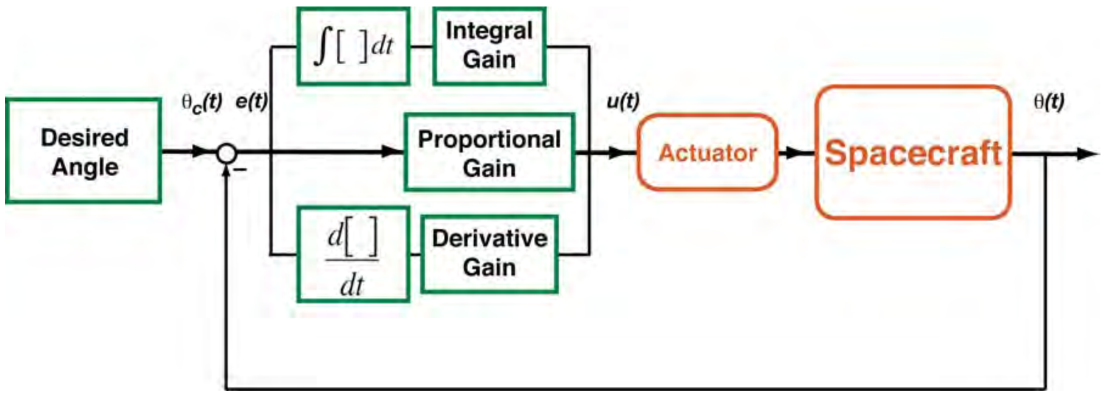
Rate Frequency Response

$$\frac{\Delta q(j\omega)}{\Delta u(j\omega)} = \frac{(j\omega)\,\omega_n^{\,2}}{(j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^{\,2}}$$



- **Bode plot**
  - 20 log(Amplitude Ratio) [dB] vs. log ω
  - Phase angle (deg) vs. log ω

Proportional-Integral-Derivative (PID) Controller



PID Control Law (or compensator):

$$e(t)=\theta_C(t)-\theta(t)$$

$$u(t)=c_I\int e(t)dt+c_P e(t)+c_D\frac{de(t)}{dt}$$

Proportional-Integral-Derivative (PID) Controller

Control Law Transfer Function:

$$\boxed{e(s)=\theta_C(s)-\theta(s)}$$

$$\boxed{u(s)=c_P e(s)+c_I\frac{e(s)}{s}+c_D s e(s)}$$

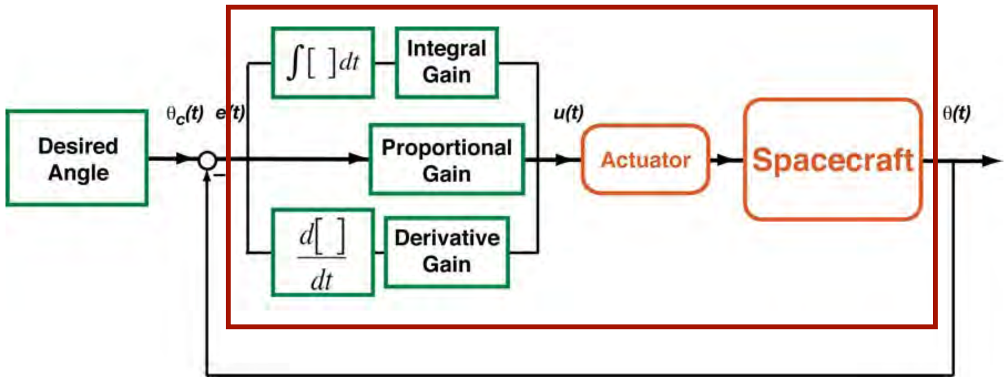
$$\boxed{\frac{u(s)}{e(s)}=\frac{c_I+c_P s+c_D s^2}{s}}$$

Differentiator produces rate term for damping
Integrator compensates for persistent (bias) disturbance

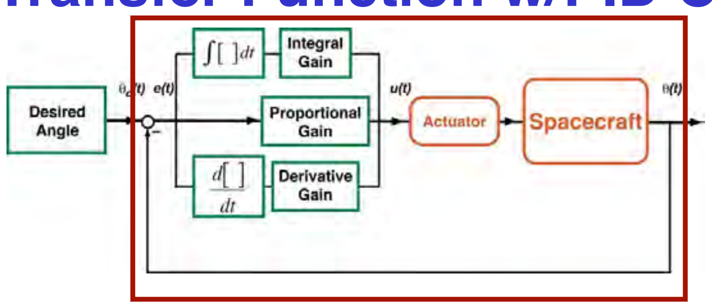
Proportional-Integral-Derivative (PID) Controller

Forward-Loop Angle Transfer Function:

$$\frac{\theta(s)}{e(s)} = \left[\frac{c_I + c_P s + c_D s^2}{s}\right]\left[\frac{g_A}{I_{yy} s^2}\right]$$



Closed-Loop Spacecraft Control Transfer Function w/PID Control



Closed-Loop Angle Transfer Function:

$$\frac{\theta(s)}{\theta_c(s)} = \frac{\dfrac{\theta(s)}{e(s)}}{1+\dfrac{\theta(s)}{e(s)}} = \frac{\left[\dfrac{c_I + c_P s + c_D s^2}{I_{yy} s^3} g_A\right]}{1+\left[\dfrac{c_I + c_P s + c_D s^2}{I_{yy} s^3} g_A\right]}$$

$$= \frac{c_I + c_P s + c_D s^2}{c_I + c_P s + c_D s^2 + g_A / I_{yy} s^3}$$

Closed-Loop Frequency Response w/PID Control

$$\boxed{\frac{\theta(s)}{\theta_c(s)} = \frac{c_I + c_P s + c_D s^2}{c_I + c_P s + c_D s^2 + g_A / I_{yy} s^3}}$$

Let $s = j\omega$. As ω -> 0

$$\boxed{\frac{\theta(j\omega)}{\theta_c(j\omega)} \rightarrow \frac{c_I}{c_I} = 1}$$

Steady-state output = desired steady-state input

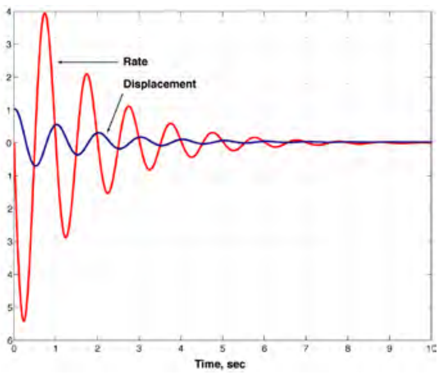
As ω -> ∞

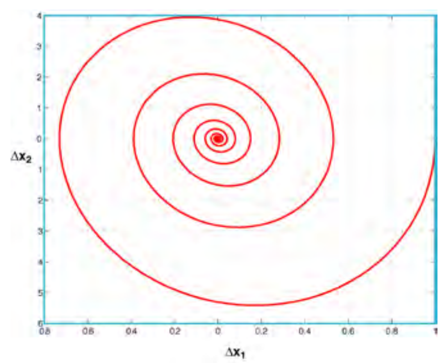
$$\frac{\theta(j\omega)}{\theta_c(j\omega)} \rightarrow \frac{-c_D\omega^2}{-jI_{yy}\omega^3} g_A = \frac{c_D}{jI_{yy}\omega} g_A = -\frac{jc_D}{I_{yy}\omega} g_A$$

$$AR \rightarrow \frac{c_D}{I_{yy}\omega} g_A; \quad \phi \rightarrow -90 \text{ deg}$$

High-frequency response "rolls off" and lags input

State ("Phase")-Plane Plots

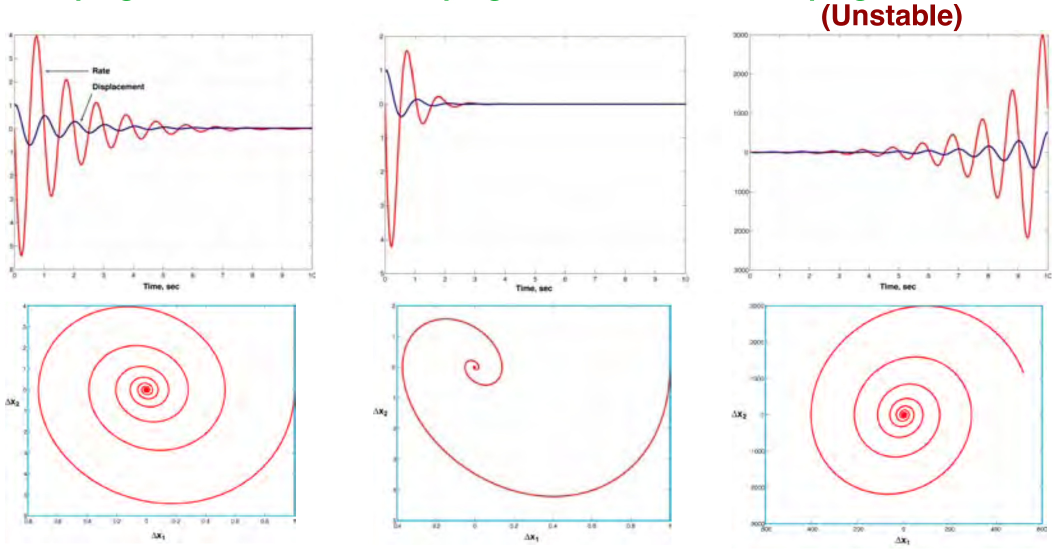




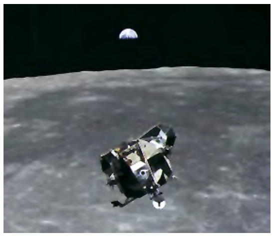
Cross-plot of angle (or displacement) against rate
Time not shown explicitly in phase-plane plot

Effect of Damping Ratio on State-Plane Plots

Damping ratio = 0.1 | **Damping ratio = 0.3** | **Damping ratio = –0.1 (Unstable)**



On/Off-Torque Controllers



Single-Axis State History with Constant Thrust

What if the control torque can only be turned ON or OFF?

$$\begin{bmatrix} \dot{\theta}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} 0 \\ g_A / I_{yy} \end{bmatrix} u(t)$$

$$u(t) = +1, \quad 0, \text{ or } -1$$

What is the time evolution of the state while a thruster is on [$u(t) = 1$]?

$$q(t) = \left(g_A / I_{yy}\right) t + q(0)$$

$$\theta(t) = \left(g_A / I_{yy}\right) t^2 / 2 + q(0)t + \theta(0)$$

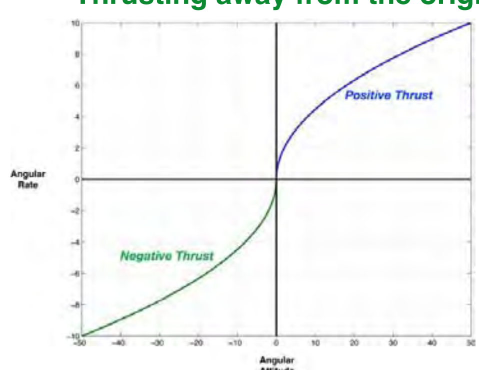
Neglecting initial conditions, what does the phase-plane plot look like?

Constant-Thrust (Acceleration) Trajectories

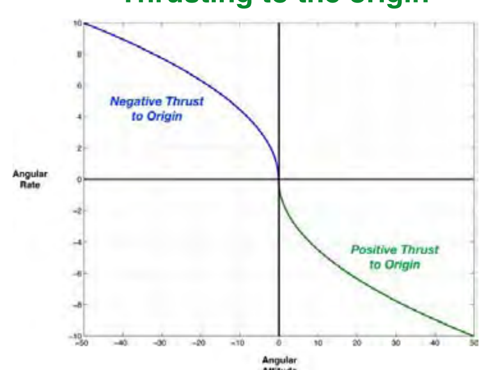
For u = 1, ***Acceleration = g_A/I_{yy}***

For u = –1, ***Acceleration = $-g_A/I_{yy}$***

Thrusting away from the origin

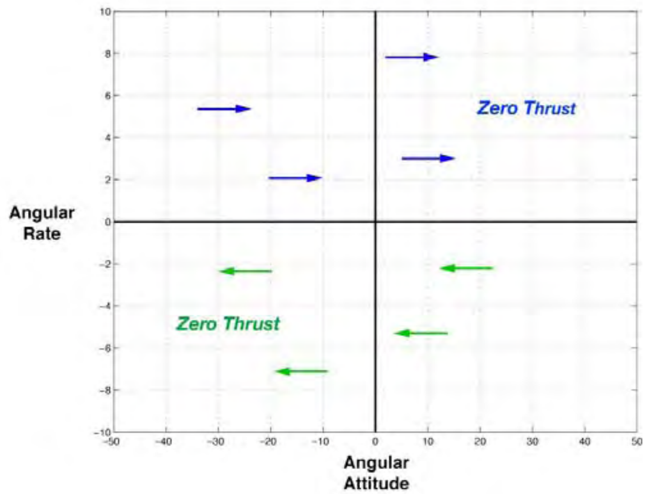


Thrusting to the origin



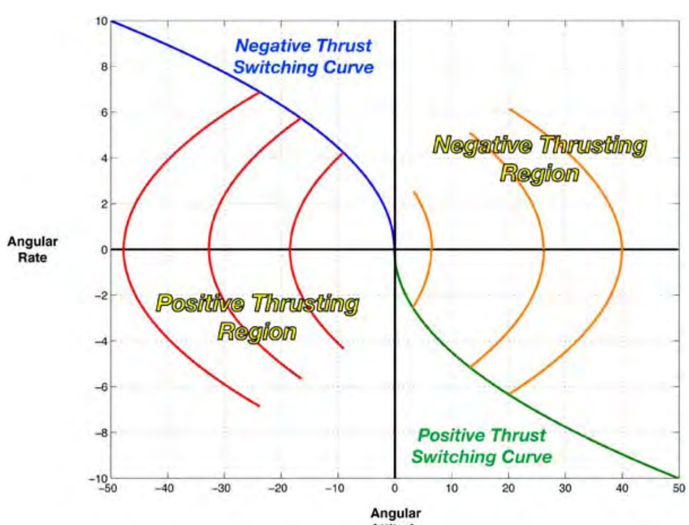
With zero thrust, what does the phase-plane plot look like?

Phase Plane Plot with Zero Thrust



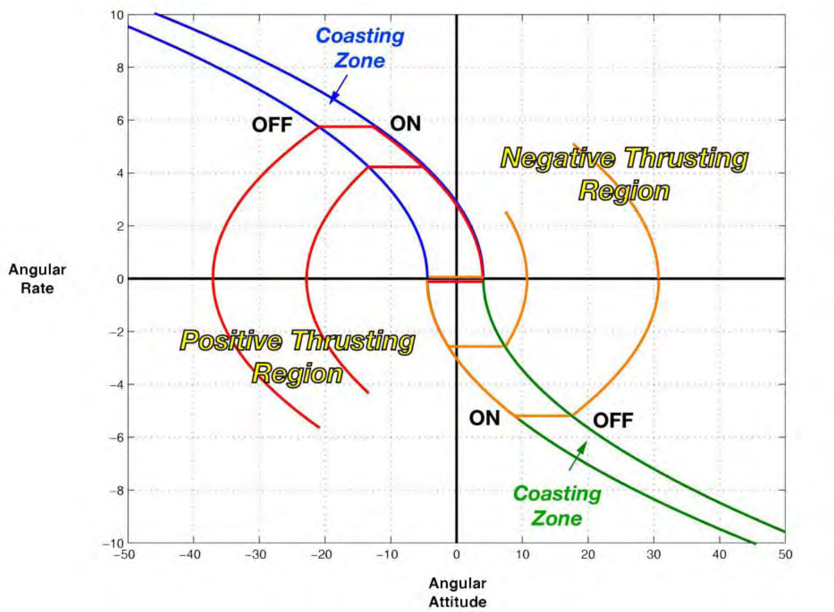
How can you use this information to design an on-off control law?

Switching-Curve Control Law for On-Off Thrusters



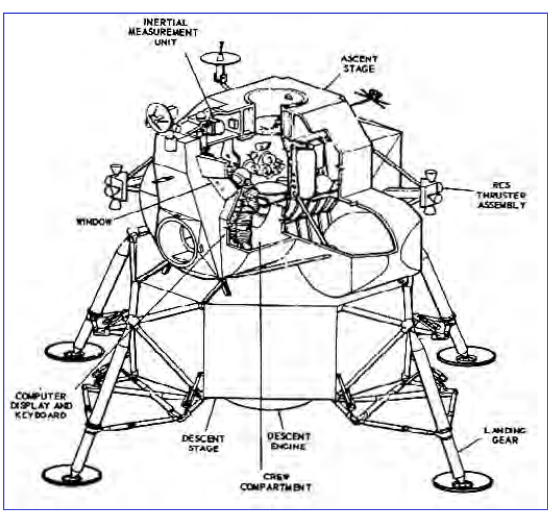
- **Origin (i.e., zero rate and attitude error) can be reached from any point in the state space**
- **Control logic:**
  - Thrust in one direction until switching curve is reached
  - Then reverse thrust
  - Switch thrust off when errors are zero

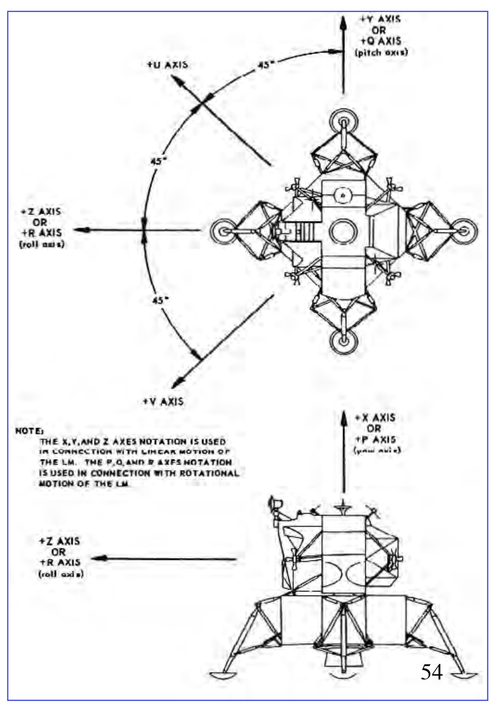
Switching-Curve Control with Coasting Zone



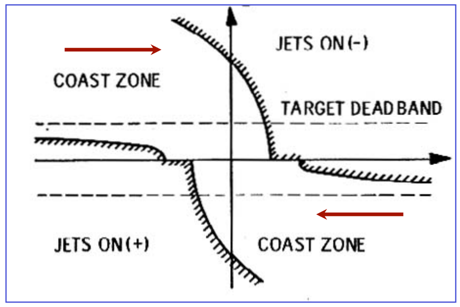
Apollo Lunar Module Control

- **16 reaction control thrusters**
  - **Control about 3 axes**
  - **Redundancy of thrusters**
- **LM Digital Autopilot**



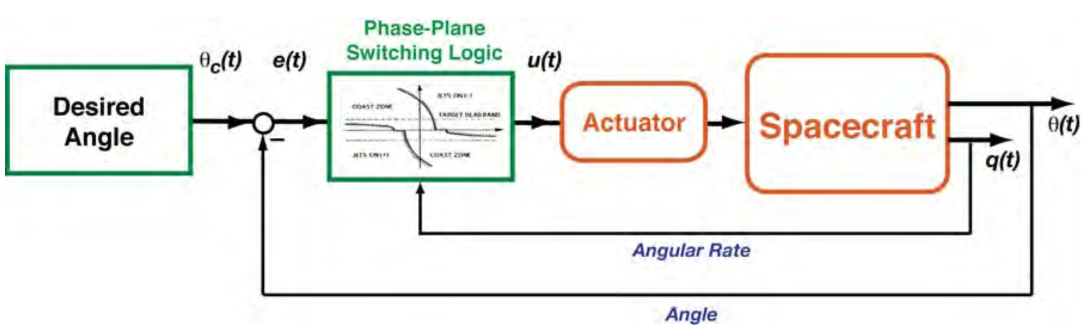


Apollo Lunar Module Phase-Plane Control Logic



- Coast zones conserve RCS propellant by limiting angular rate
- With no coast zone, thrusters would chatter on and off at origin, wasting propellant
- State limit cycles about target attitude
- Switching curve shapes modified to provide robustness against modeling errors
  - RCS thrust level
  - Moment of inertia

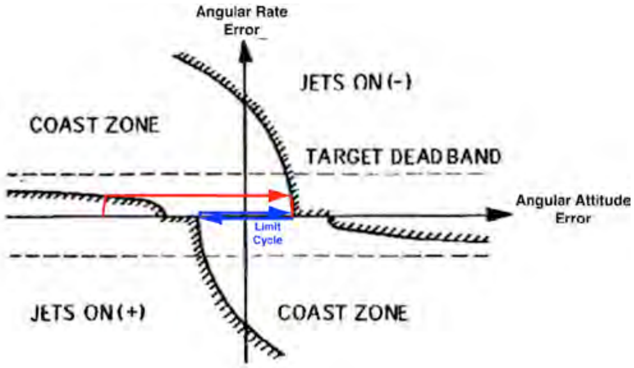
Apollo Lunar Module Phase-Plane Control Law



Switching logic implemented in the Apollo Guidance & Control Computer

More efficient than a linear control law for on-off actuators

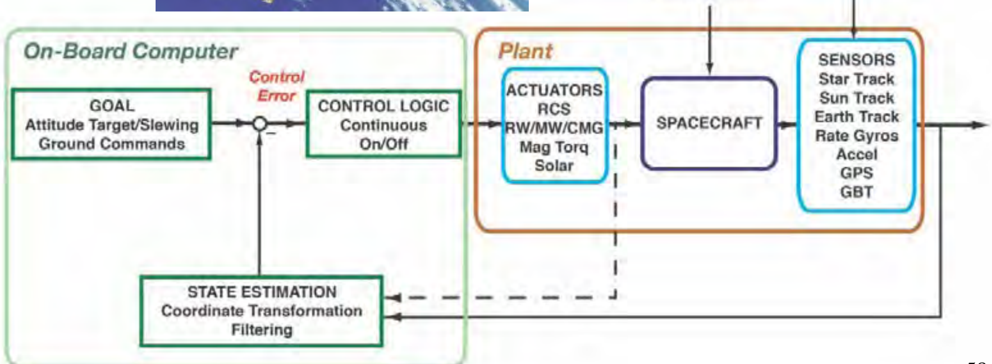
Typical Phase-Plane Trajectory



- With angle error, RCS turned on until reaching OFF switching curve
- Phase point drifts until reaching ON switching curve
- RCS turned off when rate is 0-
- Limit cycle maintained with minimum-impulse RCS firings
  - Amplitude = ±1 deg (coarse), ±0.1 deg (fine)

Multi-Axis Spacecraft Control

Asymmetry Introduces Dynamic Coupling, Complicating Control



Next Time:
Sensors and Actuators

Supplemental Material

GOES Attitude Control Sub-System