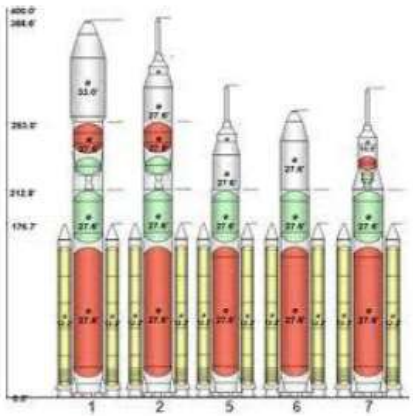
Enhanced Optimal Control For Uncertain Spacecraft Tracking Problem



AIAA Houston Annual Technical Symposium

17 May 2013

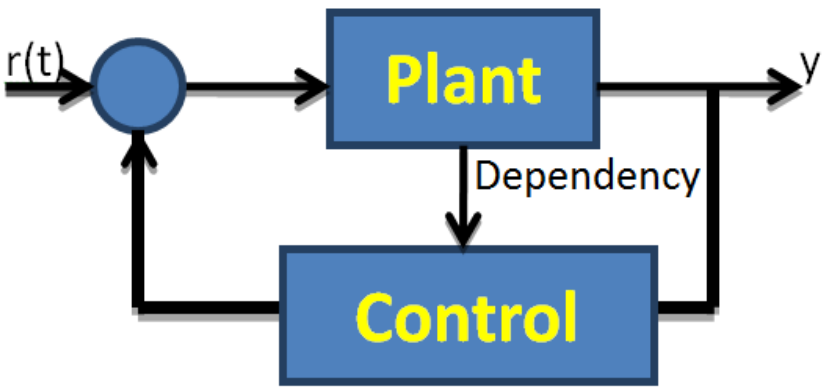
NASA/JSC, Gilruth Center, Houston, Tx.

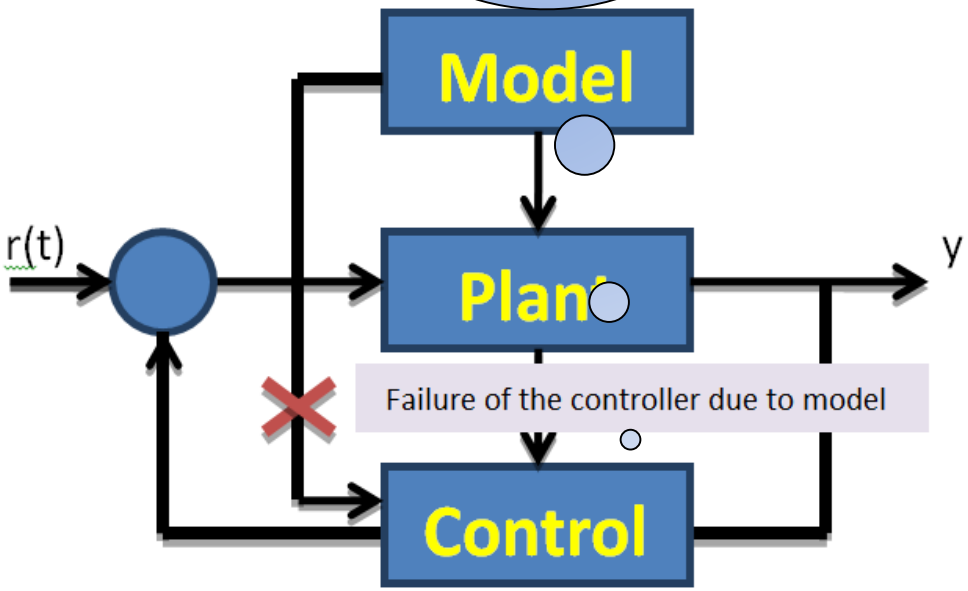
Ahmad Bani-Younes
James Turner

Problem Formulation Gaps

- **Spacecraft Tracking Problem**
  - **Geometric & Dynamic Errors Formu**
  - **Model and parameter variations in t**

In addition to using robust controller designs, can we develop means to adjust gains to compensate for model parameter revisions?





- **Closing the Gaps**
  - **Build Geometric & Dynamic Error Models That do not truncate effects**
  - **Computational Differentiation**

Outline

- **Problem Definition**
  - Sensitivity Calculation: *Scalar problem*
  - Sensitivity Calculation: *Vector problem*
  - Computational Differentiation: *Automates Calculations for High-Order Derivatives*
- **Reference Motion and Error Dynamics**
  - Optimal Reference Motion
  - Motion Error Dynamics
- **Optimal Tracking Control Formulation and Sensitivity Calculations**
  - Open-loop solution
  - Closed-loop solution
- **Future Work and Conclusion**

Sensitivity Calculation

Sensitivity calculations *begin with a differential equation that describes the response of a system to external Loads Two classes of simulation are important*:

- Prediction of the future state motion
- Motion uncertainty predictions due to the variations in the problem initial conditions and model parameter values

There is a possible way to accommodate parametric changes on control gains => Develop implicit function to relate the state feedback gain to the plant parameters over a large family of neighboring designs

Sensitivity Calculations!!

Sensitivity Calculation

Scalar Problem *Optimal Closed Loop Control*

$$\min J = \frac{1}{2} s_f x^2\left(t_f\right) + \frac{1}{2}\int_{t_0}^{t_f} q x^2(\tau) + r u^2(\tau)\, d\tau$$

subject to:

$$\dot{x}(t) = a^0 x(t) + b u(t) \qquad x\left(t_0\right) \text{ given, } t_f \text{ fixed}$$

$\Rightarrow$ optimal gain satisfies nonlinear scalar Ricatti Eq:

$$\dot{s} = -2a^0 s + s^2 \frac{b^2}{r} - q$$

Assume a
Varies by δa
$s(a,t) = ?$

$$s(a,t) = s\left(a^0,t\right) + \left(\frac{\partial s}{\partial a}\right)_{a=a^0} \delta a + \frac{1}{2!}\left(\frac{\partial^2 s}{\partial a^2}\right)_{a=a^0} \left(\delta a\right)^2 + \ldots$$

$$x(a,t) = x\left(a^0,t\right) + \left(\frac{\partial x}{\partial a}\right)_{a=a^0} \delta a + \frac{1}{2!}\left(\frac{\partial^2 x}{\partial a^2}\right)_{a=a^0} \left(\delta a\right)^2 + \ldots$$

Sensitivity Calculation

Vector Problem *Optimal Closed Loop Control*

- The state Eq. $\dot{\mathbf{x}} = A_0\mathbf{x} + B\mathbf{u}$ And the control is $\mathbf{u}(t) = -R^{-1}B^T S(t)\mathbf{x}(t)$
- Riccati Eq.

$$\dot{S} = -A^T S - SA + SBR^{-1}B^T S - Q$$

$$\dot{s}_{ij} = -a_{j_1 i} s_{j_1 j} - s_{ij_1} a_{j_1 j} + s_{ij_1} b_{j_1 j_2} r^{-1}_{j_2 j_3} b_{j_4 j_3} s_{j_4 j} - q_{ij}$$

$$a_{ij} = a^0_{ij} + \delta a_{ij}$$

Assume a Varies

- Computational Differentiation

$$\dot{S} := \left(\dot{S},\quad \nabla\dot{S},\quad \nabla^2\dot{S},\quad \nabla^3\dot{S},\quad \nabla^4\dot{S}\right)$$

$$\frac{\partial \dot{s}_{ij}}{\partial a_{j_5 j_6}} = -\delta_{ij_6} s_{j_5 j} - a_{j_1 i}\frac{\partial s_{j_1 j}}{\partial a_{j_5 j_6}} - \frac{\partial s_{ij_1}}{\partial a_{j_5 j_6}} a_{j_1 j} - \delta_{jj_6} s_{ij_5} + \frac{\partial s_{ij_1}}{\partial a_{j_5 j_6}} b_{j_1 j_2} r^{-1}_{j_2 j_3} b_{j_4 j_3} s_{j_4 j} + s_{ij_1} b_{j_1 j_2} r^{-1}_{j_2 j_3} b_{j_4 j_3}\frac{\partial s_{j_4 j}}{\partial a_{j_5 j_6}}$$

$$\boldsymbol{u}(t) = -R^{-1}B^T\left\{S + \sum_{n=1}^{4}\frac{1}{n!}\nabla^n S\cdot\delta p^n\right\}\boldsymbol{x}(t)$$

δp^n Denotes an n-th order tensor product for the parameter variations

Computational Differentiation

- **Computational Differentiation (CD)**
  - **First Developed in the 1960's for 1st Order applications**

- **Critical Need for Nth Order Partial Derivative Capabilities**
  - Turner's **Object-Oriented Coordinate Embedding Algorithm (OCEA)** provides

$$f := \left(f, \nabla f, \nabla^2 f, \nabla^3 f, \nabla^4 f \right)$$

✓ Arbitrarily complex partial derivative models Supported

✓ **OCEA** uses the programmer's math model as a template:

✓ Fortran Complier Uses Language Extensions for deriving, coding, and generating the simulation and sensitivity models

✓ All Results are Exact: No symbolic or finite difference tools used .

✓ Analyst Freed from Derivation and Coding for Complex Partials

Optimal Reference Motion

Minimize $$J = \frac{1}{2}\int_{t_0}^{t_f} \left\{ \boldsymbol{\omega}^T Q_\omega \boldsymbol{\omega} + \boldsymbol{\zeta}^T Q_\zeta \boldsymbol{\zeta} + \boldsymbol{u}^T R \boldsymbol{u} \right\} dt$$

Subject to

Angular Velocity Dynamics: $$I\dot{\boldsymbol{\omega}} = -[\boldsymbol{\omega}\times]I\boldsymbol{\omega} + \boldsymbol{u}$$

Attitude Kinematics: $$\dot{\boldsymbol{\zeta}} = f(\boldsymbol{\zeta})\boldsymbol{\omega}$$

Invoking the standard Pontryagin necessary condition for optimality

$$\dot{\boldsymbol{\lambda}}_\omega = -Q_\omega \boldsymbol{\omega} - \boldsymbol{f}(\boldsymbol{\zeta})^T \boldsymbol{\lambda}_\zeta - \left(I[\boldsymbol{\omega}\times] - [(I\boldsymbol{\omega})\times] \right) I^{-1} \boldsymbol{\lambda}_\omega$$

$$\dot{\boldsymbol{\lambda}}_\zeta = -Q_\zeta \boldsymbol{\zeta} - \frac{\partial}{\partial \boldsymbol{\zeta}} \left[\boldsymbol{f}(\boldsymbol{\zeta})\boldsymbol{\omega} \right]^T \boldsymbol{\lambda}_\zeta$$

$$\boldsymbol{u} = -R^{-1} I^{-1} \boldsymbol{\lambda}_\omega = -(IR)^{-1} \boldsymbol{\lambda}_\omega$$

Optimal Reference Motion

$Q_\omega = Q_\zeta = R = I_{3\times3}$

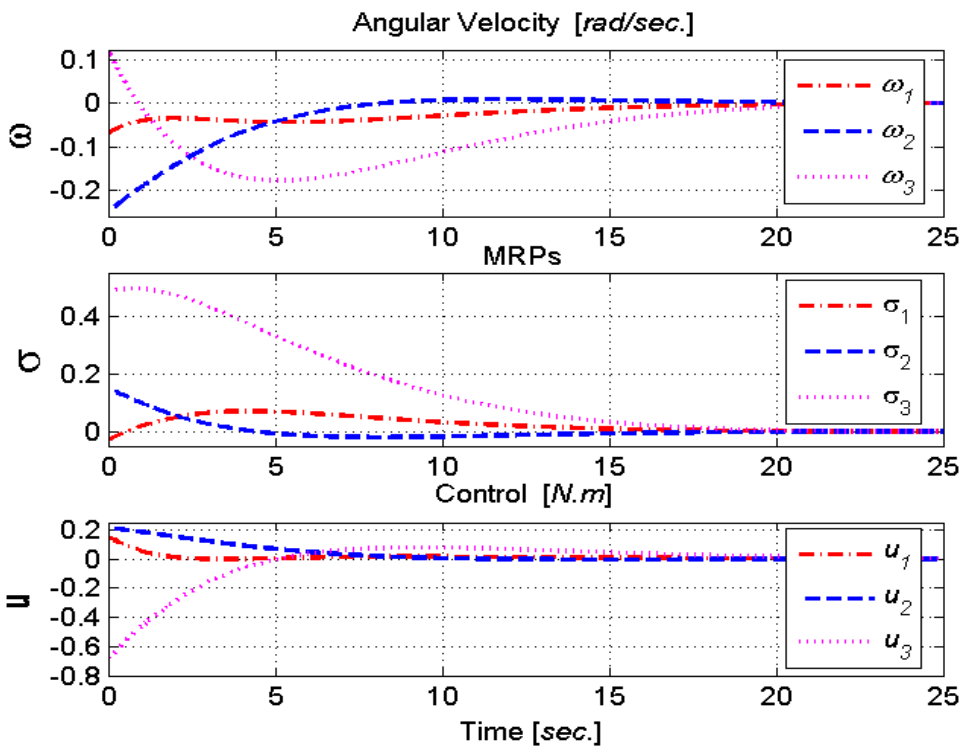
Open-Loop Optimal Control Problem

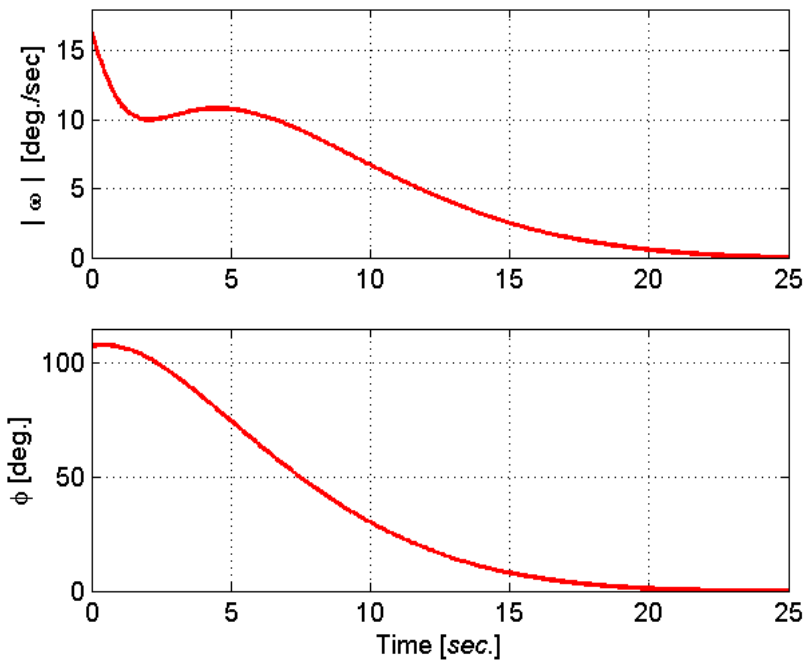
$\boldsymbol{\omega}(0) = [-0.07, -0.25, 0.118]$ rad/sec

$\boldsymbol{\zeta}(0) = [-0.03, 0.15, 0.482]$

$I = diag([3.64, 3.40, 4.41])\ \ Kg.m^2$

$Q_\omega = Q_\zeta = R = I_{3\times3}$ (Identity)





Angular Velocity Error Dynamics (Tracking Control Formulation)

The desired motion is defined in terms of the open-loop reference angular velocity

$$I_r \dot{\boldsymbol{\omega}}_r = -[\boldsymbol{\omega}_r \times] I_r \boldsymbol{\omega}_r + \tau$$

$$\delta\boldsymbol{\omega} = \boldsymbol{\omega} - \boldsymbol{\omega}_r$$

Angular Velocity Error Vector

$$\boxed{\delta\dot{\boldsymbol{\omega}} = -I^{-1}\left\{[\boldsymbol{\omega}_r \times] I - [(I\boldsymbol{\omega}_r)\times]\right\}\delta\boldsymbol{\omega} - I^{-1}[\delta\boldsymbol{\omega}\times] I \delta\boldsymbol{\omega} + I^{-1}\boldsymbol{u} - \dot{\boldsymbol{\omega}}_r - I^{-1}[\boldsymbol{\omega}_r \times] I \boldsymbol{\omega}_r}$$

Full Nonlinear Angular Velocity Error Dynamics Rate

For An Exact Kinematic Model This Equation Predicts Arbitrarily Large Motion Angular Velocity Error Vector Estimates (no linearization)

Motion Error Kinematics

Quaternion Error

$$\delta \boldsymbol{q} = \boldsymbol{q} \otimes \boldsymbol{q}_r^{-1} \Rightarrow \delta \dot{\boldsymbol{q}} = \dot{\boldsymbol{q}} \otimes \boldsymbol{q}_r^{-1} + \boldsymbol{q} \otimes \dot{\boldsymbol{q}}_r^{-1}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{Bmatrix} \boldsymbol{\omega} \\ 0 \end{Bmatrix} \otimes \boldsymbol{q} = \begin{bmatrix} -[\boldsymbol{\omega}\times] & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega} & 0 \end{bmatrix} \boldsymbol{q} = \frac{1}{2}\Omega(\boldsymbol{\omega})\boldsymbol{q}$$

$$\dot{\boldsymbol{q}}_r^{-1} = -\frac{1}{2}\boldsymbol{q}_r^{-1} \otimes \begin{Bmatrix} \boldsymbol{\omega}_r \\ 0 \end{Bmatrix} = -\frac{1}{2}\Gamma(\boldsymbol{\omega}_r)\boldsymbol{q}_r^{-1}$$

$$\boxed{\delta \dot{\boldsymbol{q}} = \frac{1}{2}\left[\Omega(\boldsymbol{\omega}) - \Gamma(\boldsymbol{\omega}_r)\right]\delta \boldsymbol{q}} \Rightarrow \boxed{\delta \dot{\boldsymbol{q}} = \frac{1}{2}\left[\Omega(\delta\boldsymbol{\omega} + \boldsymbol{\omega}_r) - \Gamma(\boldsymbol{\omega}_r)\right]\delta \boldsymbol{q} = \frac{1}{2}\left[\Omega(\delta\boldsymbol{\omega}) + \Gamma(\boldsymbol{\omega}_r)\right]\delta \boldsymbol{q}}$$

Key Step

$$\boxed{\delta \dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} -\left([\delta\boldsymbol{\omega}\times] + 2[\boldsymbol{\omega}_r\times]\right) & \delta\boldsymbol{\omega} \\ \delta\boldsymbol{\omega}^T & 0 \end{bmatrix}\begin{Bmatrix} \delta \boldsymbol{q}_v \\ \delta q_4 \end{Bmatrix}}$$

Rate Valid for Large Changes

General Procedure to Build Large Motion Error Models

Build on Nonlinear Plant & Quaternion Error Models

$$\delta \boldsymbol{q} = \boldsymbol{q} \otimes \hat{\boldsymbol{q}}^{-1} \qquad \delta \dot{\boldsymbol{q}} = \boldsymbol{f}\left(\delta \boldsymbol{q}, \delta \boldsymbol{\omega}\right)$$

Exact Error Models

Build New Variable Mapping Eqns

$$\delta \boldsymbol{\xi} = \boldsymbol{g}\left(\delta \boldsymbol{q}\right)$$

Exact Error Mapping

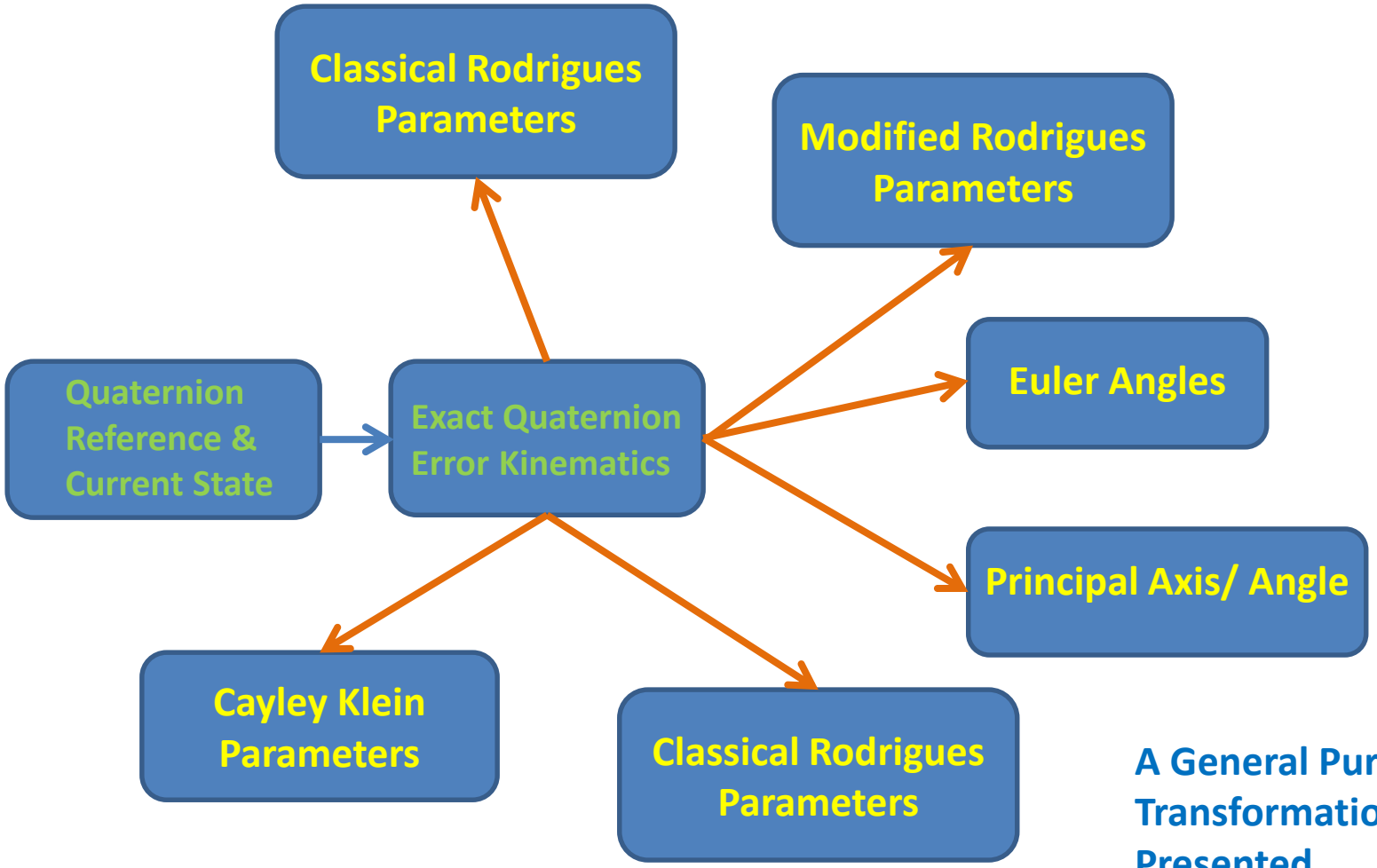
$$\delta \dot{\boldsymbol{\xi}} = \boldsymbol{h}\left(\delta \boldsymbol{q}, \delta \dot{\boldsymbol{q}}\right)$$

$$\boxed{\delta \dot{\boldsymbol{\xi}} = \boldsymbol{\phi}\left(\delta \boldsymbol{\xi}, \delta \boldsymbol{\omega}\right)}$$

Exact *Kinematics* Error Models

KEY INSIGHT: Exploit Exact Quaternion Error Model in Transformation of New Variable Sets. Leads to Models without Truncation Error for Large & Rapid Motions

Nonlinear Model Development



A General Purpose Transformation is Presented

Motion Error Kinematics

Table 1: Attitude and Attitude Error Kinematics.

Attitude	Attitude Kinematics	Attitude Error Kinematics
DCM	$\dot{C} = -[\boldsymbol{\omega}\times]C$	$\delta\dot{C} = [\delta\boldsymbol{\omega}\times]\delta C - [\boldsymbol{\omega}_r\times]\delta C + \delta C[\boldsymbol{\omega}_r\times]$
Quaternion	$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} -[\boldsymbol{\omega}\times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}\begin{Bmatrix} \boldsymbol{q}_v \\ q_4 \end{Bmatrix} = \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$	$\delta\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} -([\delta\boldsymbol{\omega}\times] + 2[\boldsymbol{\omega}_r\times]) & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^T & 0 \end{bmatrix}\begin{Bmatrix} \delta\boldsymbol{q}_v \\ \delta q_4 \end{Bmatrix} = \frac{1}{2}\begin{bmatrix} \delta q_4 & -\delta q_3 & \delta q_2 \\ \delta q_3 & \delta q_4 & -\delta q_1 \\ -\delta q_2 & \delta q_1 & \delta q_4 \\ -\delta q_1 & -\delta q_2 & -\delta q_3 \end{bmatrix}\begin{bmatrix} \delta\omega_1 \\ \delta\omega_2 \\ \delta\omega_3 \end{bmatrix} - \begin{Bmatrix} [\boldsymbol{\omega}_r\times]\delta\boldsymbol{q}_v \\ 0 \end{Bmatrix}$
CRPs	$\dot{\boldsymbol{\rho}} = \frac{1}{2}\left[[I_{3\times3}] + [\boldsymbol{\rho}\times] + \boldsymbol{\rho}\boldsymbol{\rho}^T\right]\boldsymbol{\omega}$	$\delta\dot{\boldsymbol{\rho}} = \frac{1}{2}\left[[I_{3\times3}] + [\delta\boldsymbol{\rho}\times] + \delta\boldsymbol{\rho}\delta\boldsymbol{\rho}^T\right]\delta\boldsymbol{\omega} - [\boldsymbol{\omega}_r\times]\delta\boldsymbol{\rho}$
MRPs	$\dot{\boldsymbol{\sigma}} = \frac{1}{4}\left[(1-\boldsymbol{\sigma}^T\boldsymbol{\sigma})[I_{3\times3}] + 2[\boldsymbol{\sigma}\times] + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^T\right]\boldsymbol{\omega}$	$\delta\dot{\boldsymbol{\sigma}} = \frac{1}{4}\left[(1-\delta\boldsymbol{\sigma}^T\delta\boldsymbol{\sigma})[I_{3\times3}] + 2[\delta\boldsymbol{\sigma}\times] + 2\delta\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^T\right]\delta\boldsymbol{\omega} - [\boldsymbol{\omega}_r\times]\delta\boldsymbol{\sigma}$
Euler Angles*	$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}_{ijk} = \mathbf{S}^{-1}(\theta_1, \theta_2, \theta_3)\boldsymbol{\omega}$	$\begin{bmatrix} \delta\dot{\theta}_1 \\ \delta\dot{\theta}_2 \\ \delta\dot{\theta}_3 \end{bmatrix}_{ijk} = \frac{1}{4}(H_{ijk}^T H_{ijk}^T)^{-1} H_{ijk}^T \begin{bmatrix} -([\delta\boldsymbol{\omega}\times] + 2[\boldsymbol{\omega}_r\times]) & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^T & 0 \end{bmatrix}\Theta_{ijk}$
Principal angle/axis	$\dot{\phi} = \boldsymbol{\omega}^T\hat{\boldsymbol{e}}, \quad \dot{\boldsymbol{e}} = \frac{1}{2}\left[[\hat{\boldsymbol{e}}\times] - \cot(\phi/2)[\hat{\boldsymbol{e}}\times][\hat{\boldsymbol{e}}\times]\right]\boldsymbol{\omega}$	$\delta\dot{\phi} = \delta\boldsymbol{\omega}^T\delta\hat{\boldsymbol{e}} = \delta\hat{\boldsymbol{e}}^T\delta\boldsymbol{\omega}$ $\delta\dot{\boldsymbol{e}} = \frac{1}{2}\left[[\delta\hat{\boldsymbol{e}}\times] - \cot(\delta\phi/2)[\delta\hat{\boldsymbol{e}}\times][\delta\hat{\boldsymbol{e}}\times]\right]\delta\boldsymbol{\omega} - [\boldsymbol{\omega}_r\times]\delta\hat{\boldsymbol{e}}$
Cayley-Klein	$\mathrm{col}(\dot{K}) = \frac{1}{2}\Psi_0\begin{bmatrix} -[\boldsymbol{\omega}\times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}\Psi_0^{-1}\mathrm{col}(K),$ where $\Psi_0 = \begin{bmatrix} 0 & 0 & i & 1 \\ i & -1 & 0 & 0 \\ i & 1 & 0 & 0 \\ 0 & 0 & -i & 1 \end{bmatrix}$	$\mathrm{col}(\delta\dot{K}) = \frac{1}{2}\Psi_0\begin{bmatrix} -([\delta\boldsymbol{\omega}\times] + 2[\boldsymbol{\omega}_r\times]) & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^T & 0 \end{bmatrix}\Psi_0^{-1}\mathrm{col}(\delta K)$ where $\Psi_0 = \begin{bmatrix} 0 & 0 & i & 1 \\ i & -1 & 0 & 0 \\ i & 1 & 0 & 0 \\ 0 & 0 & -i & 1 \end{bmatrix}$

*See Ref.[1] for H_{ijk} and Θ_{ijk} definitions, and see Ref.[2] for $\mathbf{S}^{-1}(\delta\theta_1, \delta\theta_2, \delta\theta_3)$ definition.

[1] Bani Younes, A., Mortari, D., Turner, J.D., and Junkins, J.L. "Attitude Error Kinematics," AIAA Journal of Guidance, Control, and Dynamics, accepted for publication.
[2] P.C. Hughes. Spacecraft attitude dynamics. J. Wiley, 1986.

Optimal Tracking Control

Optimal Open-Loop Control Formulation

Minimize

$$J = \frac{1}{2}\Phi(t_f, \delta\boldsymbol{\omega}(t_f), \delta\boldsymbol{\zeta}(t_f)) + \frac{1}{2}\int_{t_0}^{t_f} L(\delta\boldsymbol{\omega}, \delta\boldsymbol{\zeta}, \boldsymbol{u}, t)dt$$

Subject to

$$\dot{\boldsymbol{x}} = [\delta\dot{\boldsymbol{\omega}}^T \quad \delta\dot{\boldsymbol{\zeta}}^T]^T = \boldsymbol{f}(\delta\boldsymbol{\omega}, \delta\boldsymbol{\zeta}, \boldsymbol{u})$$

The Penalties

$$\Phi(t_f, \delta\boldsymbol{\zeta}(t_f), \delta\boldsymbol{\omega}(t_f)) = \delta\boldsymbol{\zeta}_{t_f}^T Q_1 \boldsymbol{\zeta}_{t_f} + \delta\boldsymbol{\omega}_{t_f}^T Q_2 \delta\boldsymbol{\omega}_{t_f}$$

and

$$L(\delta\boldsymbol{\zeta}, \delta\boldsymbol{\omega}, \boldsymbol{u}, t) = \delta\boldsymbol{\zeta}^T Q_3 \boldsymbol{\zeta} + \delta\boldsymbol{\omega}^T Q_4 \delta\boldsymbol{\omega} + \boldsymbol{u}^T R \boldsymbol{u}$$

Tracking Error Dynamics/Kinematics

$$\delta\dot{\boldsymbol{\omega}} = -I^{-1}\{[\boldsymbol{\omega}_r\times]I - [(I\boldsymbol{\omega}_r)\times]\}\delta\boldsymbol{\omega} - I^{-1}[\delta\boldsymbol{\omega}\times]I\delta\boldsymbol{\omega} + I^{-1}\boldsymbol{u} - \dot{\boldsymbol{\omega}}_r - I^{-1}[\boldsymbol{\omega}_r\times]I\boldsymbol{\omega}_r$$

$$\delta\dot{\boldsymbol{\zeta}} = f(\delta\boldsymbol{\zeta})\delta\boldsymbol{\omega} - [\boldsymbol{\omega}_r\times]\delta\boldsymbol{\zeta}$$

Optimal Open-Loop Control Formulation

Co-states

$$\dot{\boldsymbol{\lambda}}_{\delta\omega} = -Q_4\delta\boldsymbol{\omega} - f\left(\delta\boldsymbol{\zeta}\right)^T \boldsymbol{\lambda}_{\delta\zeta} - \left(I[\delta\boldsymbol{\omega}\times] - [(I\delta\boldsymbol{\omega})\times] - \left\{[\boldsymbol{\omega}_r\times]I - [(I\boldsymbol{\omega}_r)\times]\right\}^T\right) I^{-1}\boldsymbol{\lambda}_{\delta\omega}$$

$$\dot{\boldsymbol{\lambda}}_{\delta\zeta} = -Q_3\delta\boldsymbol{\zeta} - \boxed{\frac{\partial}{\partial(\delta\boldsymbol{\zeta})}\left[\boldsymbol{f}\left(\delta\boldsymbol{\zeta}\right)\delta\boldsymbol{\omega}\right]^T}\boldsymbol{\lambda}_{\delta\zeta} - [\boldsymbol{\omega}_r\times]\boldsymbol{\lambda}_{\delta\zeta}$$

Control

$$\boldsymbol{u} = -R^{-1}I^{-1}\boldsymbol{\lambda}_{\delta\omega} = -(IR)^{-1}\boldsymbol{\lambda}_{\delta\omega}$$

$$\frac{\partial}{\partial(\delta\boldsymbol{\zeta})}\left[\boldsymbol{f}\left(\delta\boldsymbol{\zeta}\right)\delta\boldsymbol{\omega}\right]$$

Quaternion

$$f\left(\delta\boldsymbol{q}\right) = \frac{1}{2}\begin{bmatrix} \delta q_4 & -\delta q_3 & \delta q_2 \\ \delta q_3 & \delta q_4 & -\delta q_1 \\ -\delta q_2 & \delta q_1 & \delta q_4 \\ -\delta q_1 & -\delta q_2 & -\delta q_3 \end{bmatrix} \quad and \quad \frac{\partial}{\partial(\delta\boldsymbol{q})}\left[f\left(\delta\boldsymbol{q}\right)\delta\boldsymbol{\omega}\right] = \frac{1}{2}\begin{bmatrix} -[\delta\boldsymbol{\omega}\times] & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^T & 0 \end{bmatrix}$$

CRPs

$$f\left(\delta\boldsymbol{\rho}\right) = \frac{1}{2}\left[[I_{3\times3}] + [\delta\boldsymbol{\rho}\times] + \delta\boldsymbol{\rho}\delta\boldsymbol{\rho}^T\right] \quad and$$

$$\frac{\partial}{\partial(\delta\boldsymbol{\rho})}\left[f\left(\delta\boldsymbol{\rho}\right)\delta\boldsymbol{\omega}\right] = [I_{3\times3}]\delta\boldsymbol{\rho}^T\delta\boldsymbol{\omega} - \frac{1}{2}[\delta\boldsymbol{\omega}\times]$$

MRPs

$$f\left(\delta\boldsymbol{\sigma}\right) = \frac{1}{4}\left[(1-\delta\boldsymbol{\sigma}^T\delta\boldsymbol{\sigma})[I_{3\times3}] + 2[\delta\boldsymbol{\sigma}\times] + 2\delta\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^T\right] \quad and$$

$$\frac{\partial}{\partial(\delta\boldsymbol{\sigma})}\left[f\left(\delta\boldsymbol{\sigma}\right)\delta\boldsymbol{\omega}\right] = \frac{1}{2}\left[[I_{3\times3}]\delta\boldsymbol{\sigma}^T\delta\boldsymbol{\omega} + \delta\boldsymbol{\sigma}\delta\boldsymbol{\omega}^T - [\delta\boldsymbol{\omega}\times] - \delta\boldsymbol{\omega}\delta\boldsymbol{\sigma}^T\right]$$

Optimal Open-Loop Control Sensitivity Calculations

Optimal Control: $\boldsymbol{u} = -(IR)^{-1}\boldsymbol{\lambda}_{\delta\omega}$

If we assume the system experiences some dynamics errors due to the ***spacecraft moment of inertia****, the resulting perturbation will subsequently influence the control calculation. To handle the gain perturbation induced by the parameter variations we assume the new "perturbed" plant parameter is given by*

$$I = I^* + \delta I$$

Where the nominal value is $I = I^*$

Optimal Open-Loop Control Sensitivity Calculations

Computational Differentiation (OCEA) automates

$$\dot{\boldsymbol{\lambda}}_{\delta\omega} := \left(\dot{\boldsymbol{\lambda}}_{\delta\omega},\quad \nabla\dot{\boldsymbol{\lambda}}_{\delta\omega},\quad \nabla^2\dot{\boldsymbol{\lambda}}_{\delta\omega},\quad \nabla^3\dot{\boldsymbol{\lambda}}_{\delta\omega},\quad \nabla^4\dot{\boldsymbol{\lambda}}_{\delta\omega}\right)$$

$$\dot{\boldsymbol{\lambda}}_{\delta\zeta} := \left(\dot{\boldsymbol{\lambda}}_{\delta\zeta},\quad \nabla\dot{\boldsymbol{\lambda}}_{\delta\zeta},\quad \nabla^2\dot{\boldsymbol{\lambda}}_{\delta\zeta},\quad \nabla^3\dot{\boldsymbol{\lambda}}_{\delta\zeta},\quad \nabla^4\dot{\boldsymbol{\lambda}}_{\delta\zeta}\right)$$

The extrapolated sensitivity feedback gain is calculated from the truncated Taylor Series expansion:

$$\boldsymbol{\lambda}_{\delta\omega} = \boldsymbol{\lambda}^*_{\delta\omega} + \sum_{n=1}^{4}\frac{1}{n!}\nabla^n\boldsymbol{\lambda}_{\delta\omega}.\delta I^n, \quad \text{and} \quad \boldsymbol{\lambda}_{\delta\zeta} = \boldsymbol{\lambda}^*_{\delta\zeta} + \sum_{n=1}^{4}\frac{1}{n!}\nabla^n\boldsymbol{\lambda}_{\delta\zeta}.\delta I^n$$

The implemented feedback control is defined by

$$\boldsymbol{u}(t) = -(IR)^{-1}\left\{\boldsymbol{\lambda}_{\delta\omega} + \sum_{n=1}^{4}\frac{1}{n!}\nabla^n\boldsymbol{\lambda}_{\delta\omega}\cdot\delta I^n\right\}$$

δI^n Denotes an n-th order tensor product for the parameter variations

Optimal Tracking Control

Optimal Closed-Loop Control Formulation

Minimize $$J = \frac{1}{2}\Phi(t_f, x_i(t_f)) + \frac{1}{2}\int_{t_0}^{t_f} \left\{ q_{j_3 j_4} x_{j_3} x_{j_4} + r_{j_5 j_6} u_{j_5} u_{j_6} \right\} dt$$

Subject to $$\dot{x}_i = a_{ij_1} x_{j_1} + c_{ijkl} x_j x_k x_l + t_{imn} x_m x_n + d_i + b_{ij_2} u_{j_2}$$

Co-states $$\dot{\lambda}_i = -q_{ij_4} x_{j_4} - \lambda_{i_1} \left\{ a_{i_1 i} + c_{i_1 jki} x_j x_k + c_{i_1 jil} x_j x_l + c_{i_1 ikl} x_k x_l + t_{i_1 mi} x_m + t_{i_1 in} x_n \right\}$$

Co-states (assumed form) $$\lambda_{i_4} = k_{i_4 j_8} x_{j_8} + p_{i_4}$$

The Control Gains:

$$\dot{p}_i = -k_{ij} d_j + k_{ij} b_{jk} r_{lk}^{-1} b_{ml} p_m - p_n a_{ni}, \quad p_i(t_f) = 0$$

$$\dot{k}_{ij} = -k_{kj} a_{ki} + k_{il} b_{lm} r_{nm}^{-1} b_{on} k_{oj} - k_{il} a_{lj} - q_{ij} - p_r t_{rji} - p_r t_{rij}, \quad k_{ij}(t_f) = \mathrm{K}_f$$

Optimal Closed-Loop Control Sensitivity Calculations

Optimal Control $u_l = -r_{ml}^{-1} b_{im} \left\{ k_{ij} x_j + p_i \right\}$

If we assume the system experiences some dynamics errors due to the ***spacecraft moment of inertia****, the resulting perturbation will subsequently influence the control calculation. To handle the gain perturbation induced by the parameter variations we assume the new "perturbed" plant parameter is given by*

$$I = I^* + \delta I$$

Where the nominal value is $I = I^*$

Optimal Closed-Loop Control Sensitivity Calculations

Computational Differentiation (OCEA) automates

$$\dot{k}_{ij} := \left(\dot{k}_{ij}, \quad \nabla \dot{k}_{ij}, \quad \nabla^2 \dot{k}_{ij}, \quad \nabla^3 \dot{k}_{ij}, \quad \nabla^4 \dot{k}_{ij} \right)$$

$$\dot{p}_i := \left(\dot{p}_i, \quad \nabla \dot{p}_i, \quad \nabla^2 \dot{p}_i, \quad \nabla^3 \dot{p}_i, \quad \nabla^4 \dot{p}_i \right)$$

The extrapolated sensitivity feedback gain is calculated from the truncated Taylor Series expansion:

$$K = K^* + \sum_{n=1}^{4} \frac{1}{n!} \nabla^n K . \delta I^n, \quad \text{and} \quad p = p^* + \sum_{n=1}^{4} \frac{1}{n!} \nabla^n p . \delta I^n$$

The implemented feedback control is defined by

$$u_l(t) = -r_{ml}^{-1} b_{im} \left\{ \left\{ k_{ij} + \sum_{n=1}^{4} \frac{1}{n!} \nabla^n k_{ij} \cdot \delta I^n \right\} x_j + p_i + \sum_{n=1}^{4} \frac{1}{n!} \nabla^n p_i \cdot \delta I^n \right\}$$

δI^n Denotes an n-th order tensor product for the parameter variations

Optimal Tracking Control

$Q_\omega = Q_\zeta = R = I_{3\times3}$

Closed-Loop Optimal Control Problem

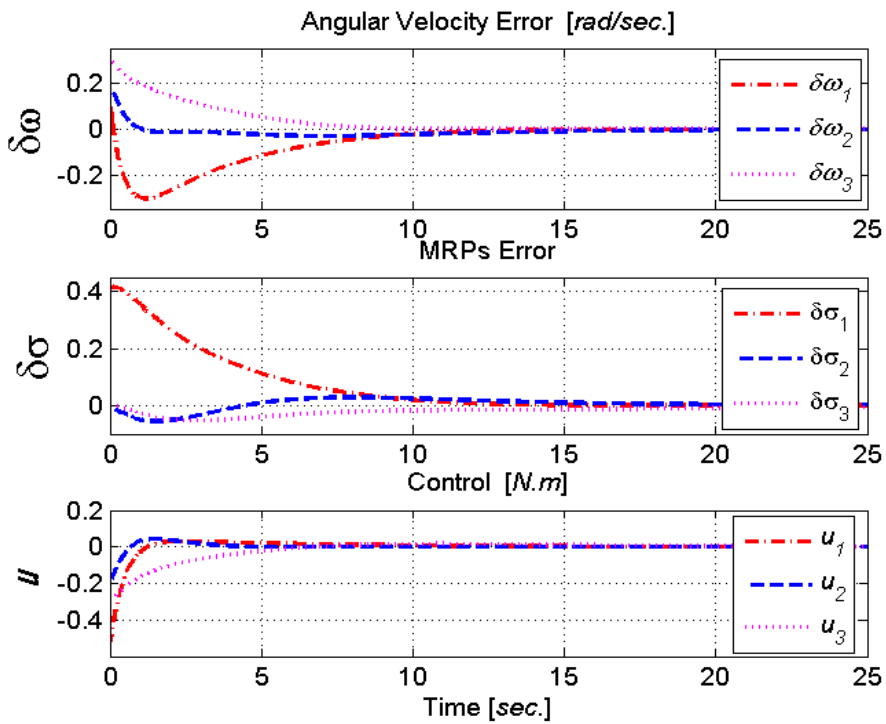
$$\boldsymbol{\omega}(0) = [0.1, 0.2, 0.3]\,\text{rad/sec},\ \boldsymbol{\zeta}(0) = [-2.4142, 0, 0]$$

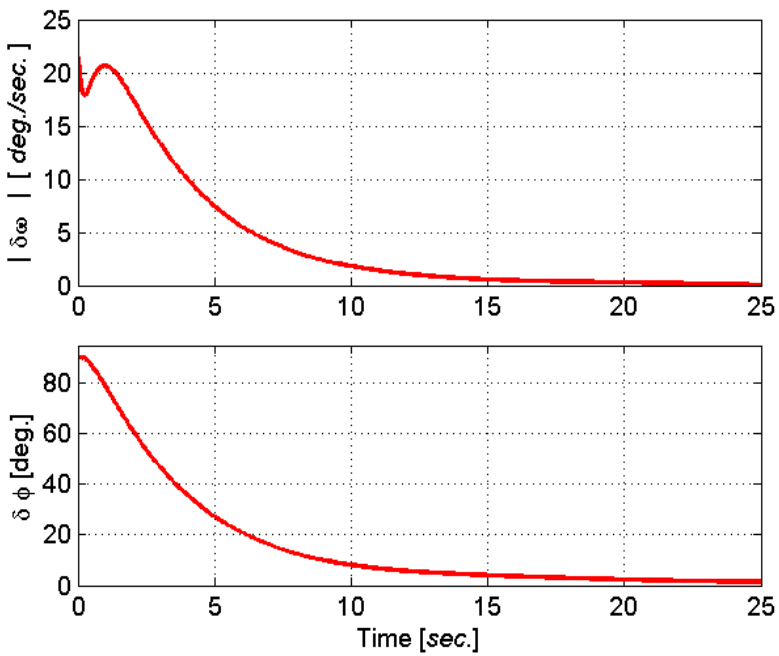
$$I = diag([0.5, 0.6, 1.0]) \quad Kg.m^2$$

$$Q_\omega = Q_\zeta = R = I_{3\times3} \quad \text{(Identity)}$$

$$p_i = 0 \ \text{ and } \ k_{ij} = \delta_{ij}$$

$$\text{where} \begin{cases} \delta_{ij} = 0 & \text{if } i \neq j \\ \delta_{ij} = 1 & \text{if } i = j \end{cases}$$





Optimal Tracking Control

$Q_\omega = Q_\zeta = R = I_{3\times3}$

Closed-Loop Optimal Control Problem

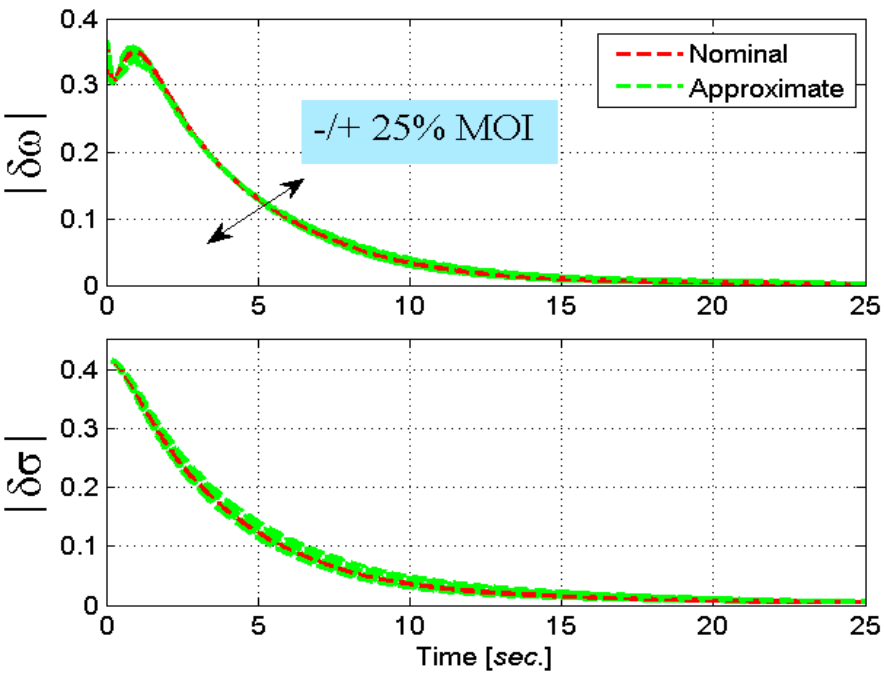
$$\boldsymbol{\omega}(0) = [0.1, 0.2, 0.3]\,\text{rad/sec},\ \boldsymbol{\zeta}(0) = [-2.4142, 0, 0]$$

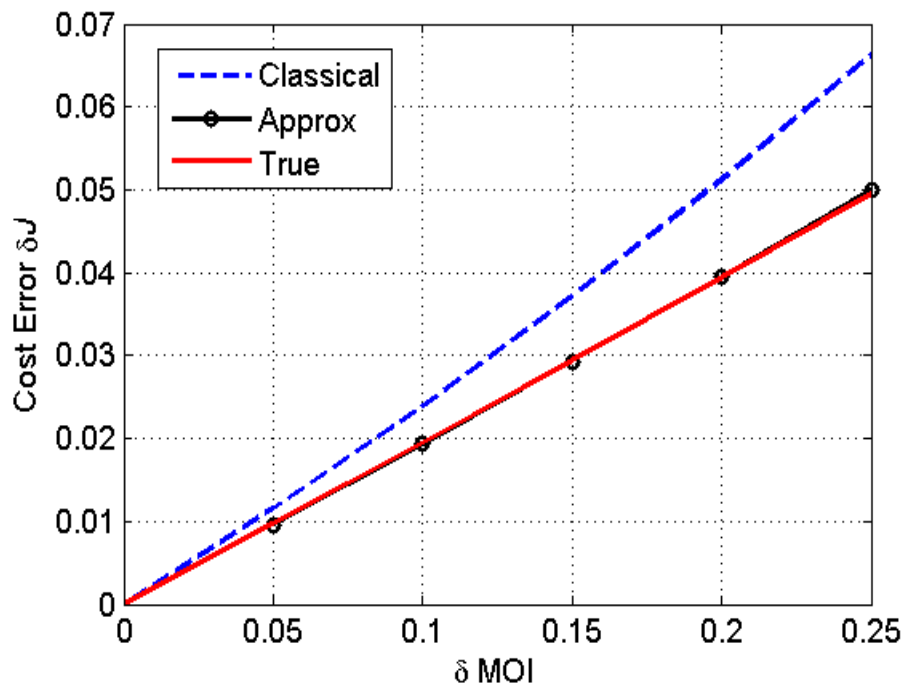
$$I = diag([0.5, 0.6, 1.0]) \quad Kg.m^2$$

$$Q_\omega = Q_\zeta = R = I_{3\times3} \quad \text{(Identity)}$$

$$p_i = 0 \ \text{ and } \ k_{ij} = \delta_{ij}$$

$$\text{where} \begin{cases} \delta_{ij} = 0 & \text{if } i \neq j \\ \delta_{ij} = 1 & \text{if } i = j \end{cases}$$





Conclusion & Future Work

- Optimal spacecraft maneuver strategies are presented for both open- and closed-loop problem formulations

- Several innovations are introduced
  - Coordinate variable transformations are identified that enable error kinematics models to be developed for arbitrarily large rigid body rotational motion
  - Large rotational motion models are presented for several frequently used sets of attitude variables
  - Large angular rate dynamics models are presented for supporting closed-loop tracking problem formulations
  - A generalized Taylor-series based approach is introduced for a Tacking feedback control formulation that accounts for very large changes in the spacecraft mode parameters

- Numerical simulation results are presented that demonstrate that the new control gain sensitivity-based approach can easily accommodate moment of inertia uncertainties as large as **25%** and still achieve the spacecraft maneuver objectives

Conclusion & Future Work

- The generalized sensitivity approach is expected to be broadly useful, because it replaces a need for re-computing the optimal solution for each parameter gain change or introducing a gain scheduling strategy, with a pre-computed gain calculation for handling a large family of model parameter changes

- The OCEA (Object Oriented Coordinate Embedding) computational differentiation toolbox is used for automatically generating the first- through fourth-order partial derivatives required for the generalized control sensitivity differential equation

- Future Extensions
  - Study other parameter changes; such as uncertain initial conditions
  - Apply the same methodology on other optimal control problems (Enhanced Optimal Control Solution)
  - Discrete time problems
  - Terminal Constraint problems

Thanks for listening ...!!

Thank You!

AHMAD BANIYOUNES
olalahmad@gmail.com