Fault Tolerant Control Allocation for Mars Vehicle using Adaptive Control

Monika Marwaha and John Valasek

Texas A&M University , College Station, Texas

AIAA/AAS Astrodynamics Specialist Conference and Exhibit

August 18-21, 2008

Contents

- Problem Statement
- Research Overview
- Structured Adaptive Model Inversion (SAMI)
- Control Allocation
- Numerical Results
- Conclusions
- Future Work
- Challenges and Open Problems

Problem Statement

	Past	Future
Mission	Robotic	Manned
Vehicle	Small	Large
Landing Requirements	< 100 km from target	< 5 km from target
Entry Trajectory	Ballistic	Controlled

Adaptive guidance and control systems are a potential solution

Research Overview

Approach: To design a fault tolerant controller for Mars entry vehicle

Key Issues:

- Nonlinear time varying system
- Guided or Ballistic reference trajectory
- Uncertainties in aerodynamic coefficients, Mars atmospheric density, winds and inertia properties of the vehicle

Research Overview

Solution: To use fault tolerant discrete control allocation coupled with adaptive control

Benefits:

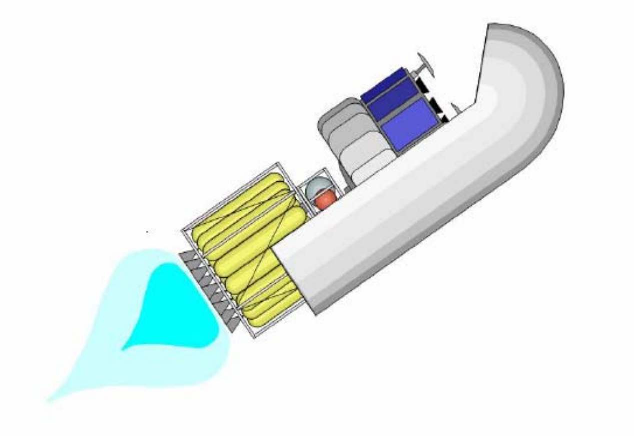
- Track both kinematics and dynamic level states
- Satisfactory tracking performance even in case of jet failures
- Handles uncertainties in aerodynamic coefficients, Mars atmospheric density and vehicle's inertia properties

Mars Ellipsled Video

Vehicle Model

Vehicle Model

- A scaled down model of Mars Ellipsled
- Cylinder with diameter 3.75 m and length 6.323 m



Mars Ellipsled

RCS Model

- Eighteen RCS control jets
- Nine jets on each side in clusters of three
- Each jet influence more than one axis
- Jet on/off time is not considered
- Provide constant thrust throughout flight

Trajectory

- **Vehicle Trajectory**

$$J(\sigma) = 0.25[1 - \sigma^T \sigma] I_{3\times3} + 2[\tilde{\sigma}] + 2\sigma\sigma^T$$

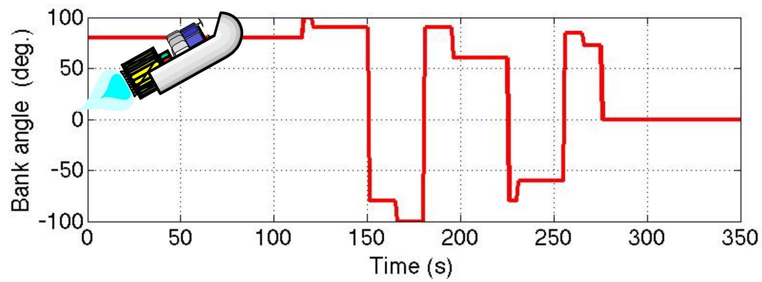
$$I\dot{\omega} + \omega \times I\omega = M_{aerodynamic} + M_{control}$$

where

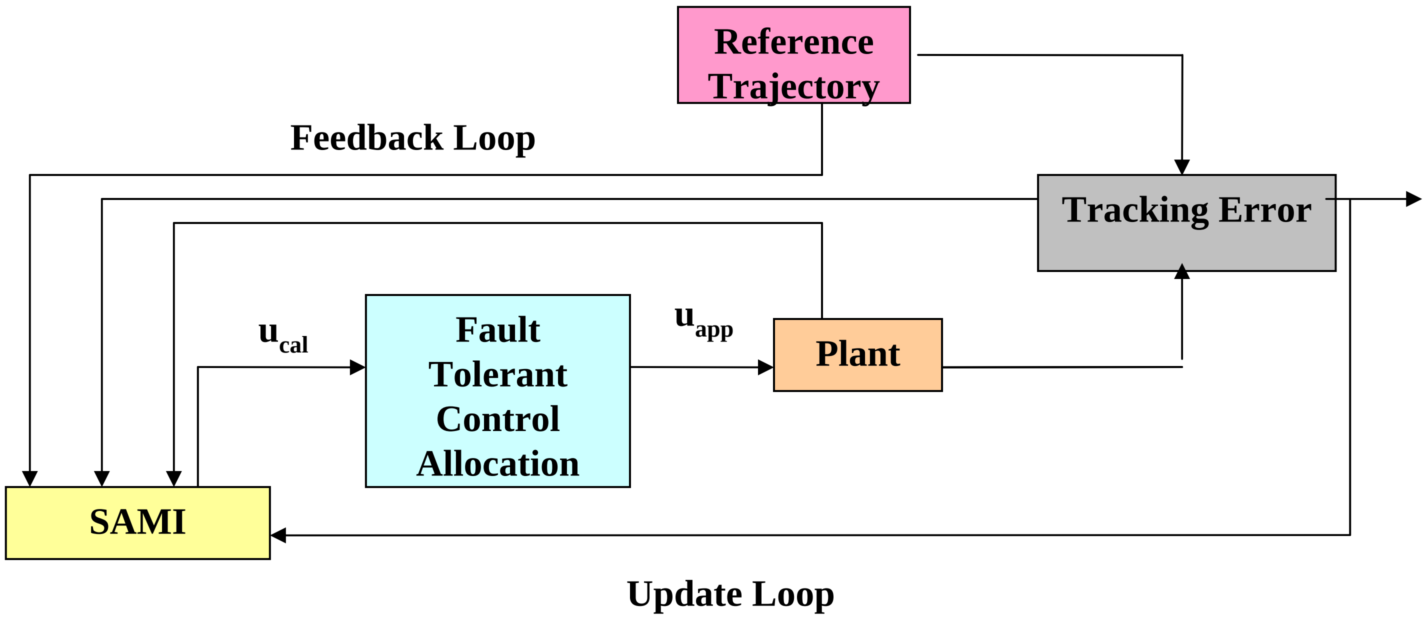
σ Modified Rodrigues Parameters (MRPs)

ω Angular Velocities

- **Reference Trajectory** Discrete bank angle commands.



Flow Diagram of Control Architecture



Structured Model Reference Adaptive Control

Akella, Schaub, Junkins (Texas A&M)

Dynamics

2nd order differential equations

→ Exact kinematic relationship between position and velocity

→ Acceleration level relationships between forces and system parameters

$$F = ma$$

$$\dot{x} = v$$

$$\dot{v} = a = F / m$$

Structured Adaptive Model Inversion

Subbarao, Junkins (Texas A&M)

- Features
  - Dynamic inversion inner-loop, with an MRAC outer-loop to handle system uncertainties.
  - Controls are solved for explicitly:

    System Model $\dot{x} = Af(x) + Bu$

    Reference Trajectory $x_r, \dot{x}_r$

    Control Law $u = B^{-1}(\dot{x}_r - Af(x) - \lambda e)$ so that the error dynamics becomes $\dot{e} = -\lambda e$
  - **Undesirable dynamics are cancelled and replaced with user specified desired dynamics.**
  - **Easily applicable to nonlinear systems.**
  - Error dynamics can be specified.
  - Shown to be very effective for a wide variety of systems.

SAMI

- Acceleration level equation of plant rewritten

$$I_a^*(\sigma)\ddot{\sigma} + C_a^*(\sigma,\dot{\sigma})\dot{\sigma} = P_a^T(\sigma)(u + M_{aero})$$

where

$$P_a(\sigma) = J_a^{-1}(\sigma)$$

$$I_a^*(\sigma) = P_a^T I P_a$$

$$C_a^*(\sigma,\dot{\sigma}) = I_a^* \dot{J}_a P_a + P_a^T [P_a \dot{\sigma}] I P_a$$

$$M_{aero} = Aerodynamic \text{ Moments}$$

Aerodynamic Moments

$$M_{aero} = \frac{1}{2} S_{ref} l_{ref} V^2 \begin{bmatrix} \beta & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \beta \end{bmatrix} \underbrace{\begin{bmatrix} \rho c_{l\beta} \\ \rho c_{m\alpha} \\ \rho c_{n\beta} \end{bmatrix}}_{d^*}$$

d^* → **Unknown Parameters**

$$d^* = D_{est} d$$

$$M_{aero} = \frac{1}{2} S_{ref} l_{ref} V^2 \begin{bmatrix} \beta & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \beta \end{bmatrix} D_{est} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

Minimal Parameterization of the Inertia Matrix

$$\begin{bmatrix} I_{11} & I_{12} & I_{13} \\ I_{12} & I_{22} & I_{23} \\ I_{13} & I_{23} & I_{33} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 & a_2 & a_3 & 0 \\ 0 & a_2 & 0 & a_1 & 0 & a_3 \\ 0 & 0 & a_3 & 0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} I_{11} \\ I_{22} \\ I_{33} \\ I_{12} \\ I_{13} \\ I_{23} \end{bmatrix}$$

Inertia Matrix
9 parameters

Inertia Vector
6 minimal parameters

$$Ia = \Lambda(a)\theta, \quad \forall a \in R^3$$

SAMI

- Equations of motion can be converted to

$$I_a^*(\sigma)\ddot{\sigma} + C_a^*(\sigma,\dot{\sigma})\dot{\sigma} = Y(\sigma,\dot{\sigma},\ddot{\sigma})\boxed{\theta}$$

unknown parameters

- Tracking error

$$\varepsilon = \sigma - \sigma_r$$

- Control Law

design variables

$$u = P_a^{-T}\{Y_a(\sigma,\dot{\sigma},\dot{\sigma}_r,\ddot{\sigma}_r)\theta - \boxed{C_{da}}\dot{\varepsilon} - \boxed{k_{da}}\varepsilon - \boxed{k_i}\int \varepsilon dt\} - M_{aero}$$

Control Law

- For unknown parameters

$$u = P_a^{-T}\{Y_a(\sigma,\dot{\sigma},\dot{\sigma}_r,\ddot{\sigma}_r)\hat{\theta} - C_{da}\varepsilon - k_{da}\varepsilon - k_i\int \varepsilon dt\} - \underbrace{\frac{1}{2}S_{ref}l_{ref}V^2\begin{bmatrix}\beta & 0 & 0\\ 0 & \alpha & 0\\ 0 & 0 & \beta\end{bmatrix}D_{est}}_{L}\underbrace{\begin{bmatrix}\hat{d}_1\\ \hat{d}_2\\ \hat{d}_3\end{bmatrix}}_{\hat{d}}$$

Let
$$\tilde{\theta} = \theta - \hat{\theta}$$
$$\tilde{d} = d - \hat{d}$$

Error dynamics

$$I_a\ddot{\varepsilon} + (C_{da}\dot{\varepsilon} + k_d\varepsilon + k_i\int \varepsilon dt = \Psi\tilde{\Phi}$$

$$\Psi = [Y_a \quad -P^T L]$$

$$\tilde{\Phi} = [\tilde{\theta} \quad \tilde{d}]$$

Adaptive Laws

Let

$$y = \left[\int \varepsilon dt \;\; \varepsilon \;\; \dot{\varepsilon} \right]$$

Lyapunov Candidate

$$V = \frac{1}{2} y^T P y + \frac{1}{2} \tilde{\Phi}^T P \tilde{\Phi}$$

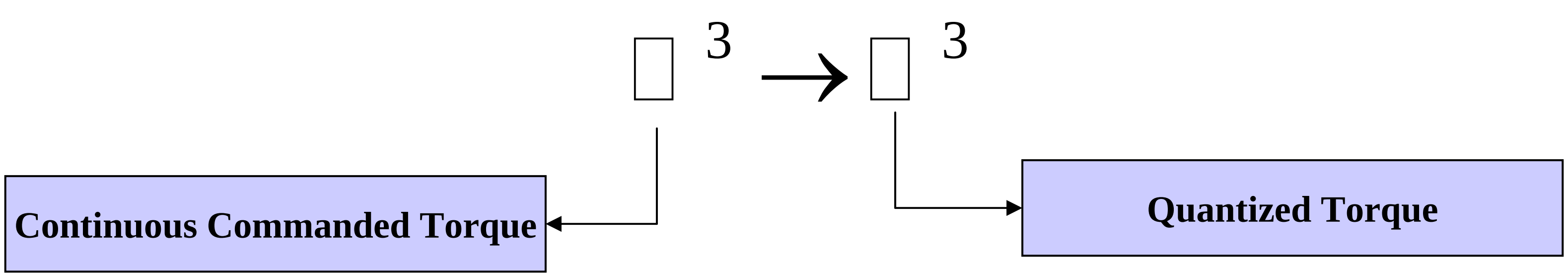
- Parameter update

$$\dot{\hat{\Phi}} = \Gamma \Psi^T P y$$

- Stability analysis shows that $\sigma \to \sigma_r$ and $\omega \to \omega_r$ as $t \to \infty$

Control Allocation

- Why Control Allocation?
  - More jets than desired moments
  - To provide consistent and unique solution to the problem
- Implemented using Mixed Integer Linear Programming (MILP)
- Fault Tolerant Algorithm
- RCS control allocator acts as quantization element



RCS Control Allocation

Doman, Gamble and Ngo

An RCS control allocation formulation:

$$\min_{u} \sum_{i=1}^{3} \left| \tau_{i_{des}} - \sum_{k=1}^{p} T_{i,k}\, u_k \right| + \sum_{k=1}^{p} w_k u_k$$

Torque provided by reaction jets

- Desired Torque to follow reference trajectory
- Calculated using Adaptive Control

On/off Value i.e. 0 or 1

Stability Analysis

For nonlinear system

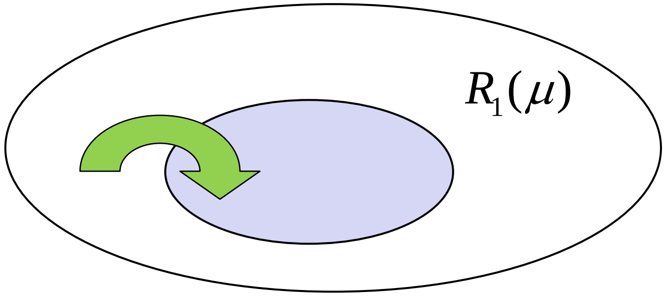
$$\dot{x} = f(x) + G(x)u$$

calculated control $u = k(x)$

applied control $u = q_{\mu}(k(x))$

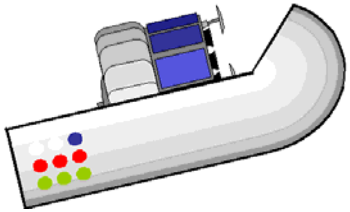
Quantization variable (fixed)

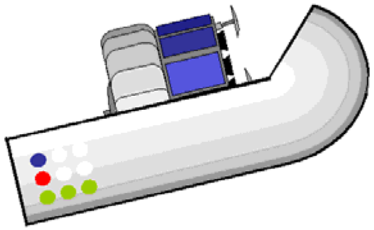
Liberzon has proved that solutions starting in $R_1(\mu)$ enters $R_2(\mu)$ in *finite* time



Controller Evaluation

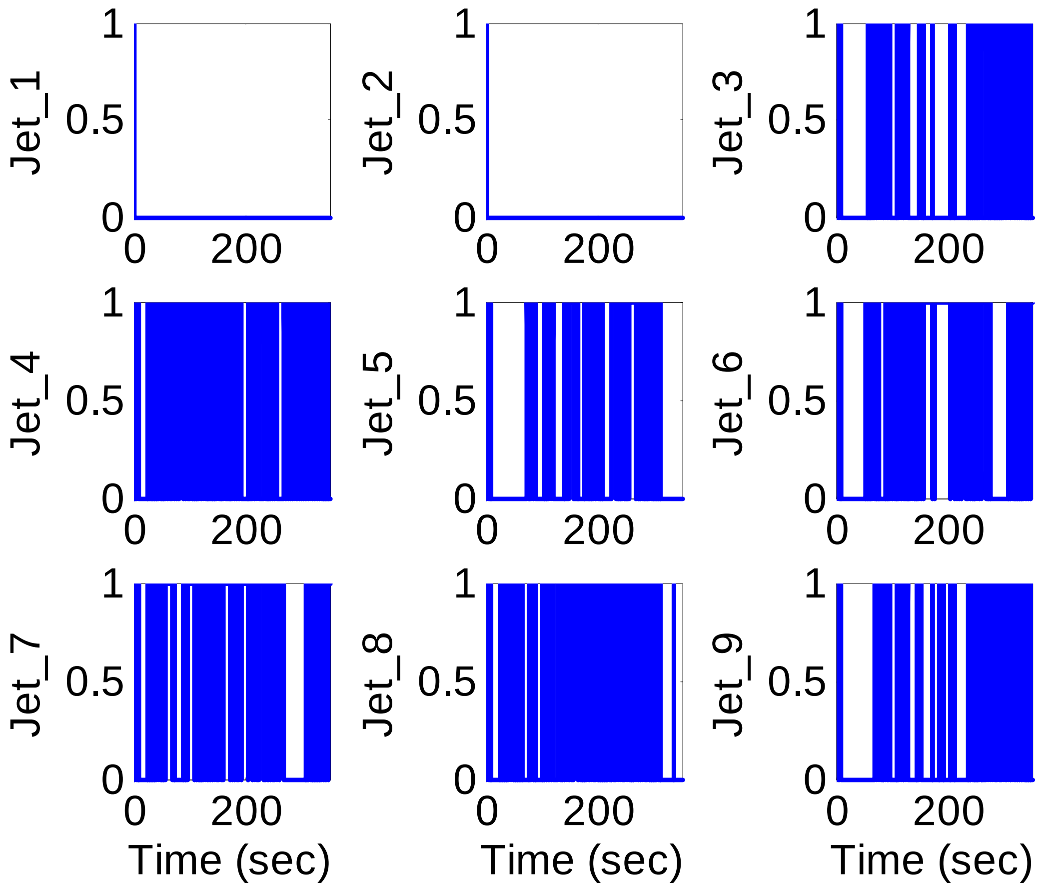
- Uncertainties in moment of mass , inertia and aerodynamic coefficients
- Initial condition errors
- Test cases shown
  - Case 1: Jets 1,2,17,18 are inoperable i.e. always off
  - Case 2: Jets 2,3,4,5,8,9,18 are always on
  - Case 3: Torque producing capability decreased





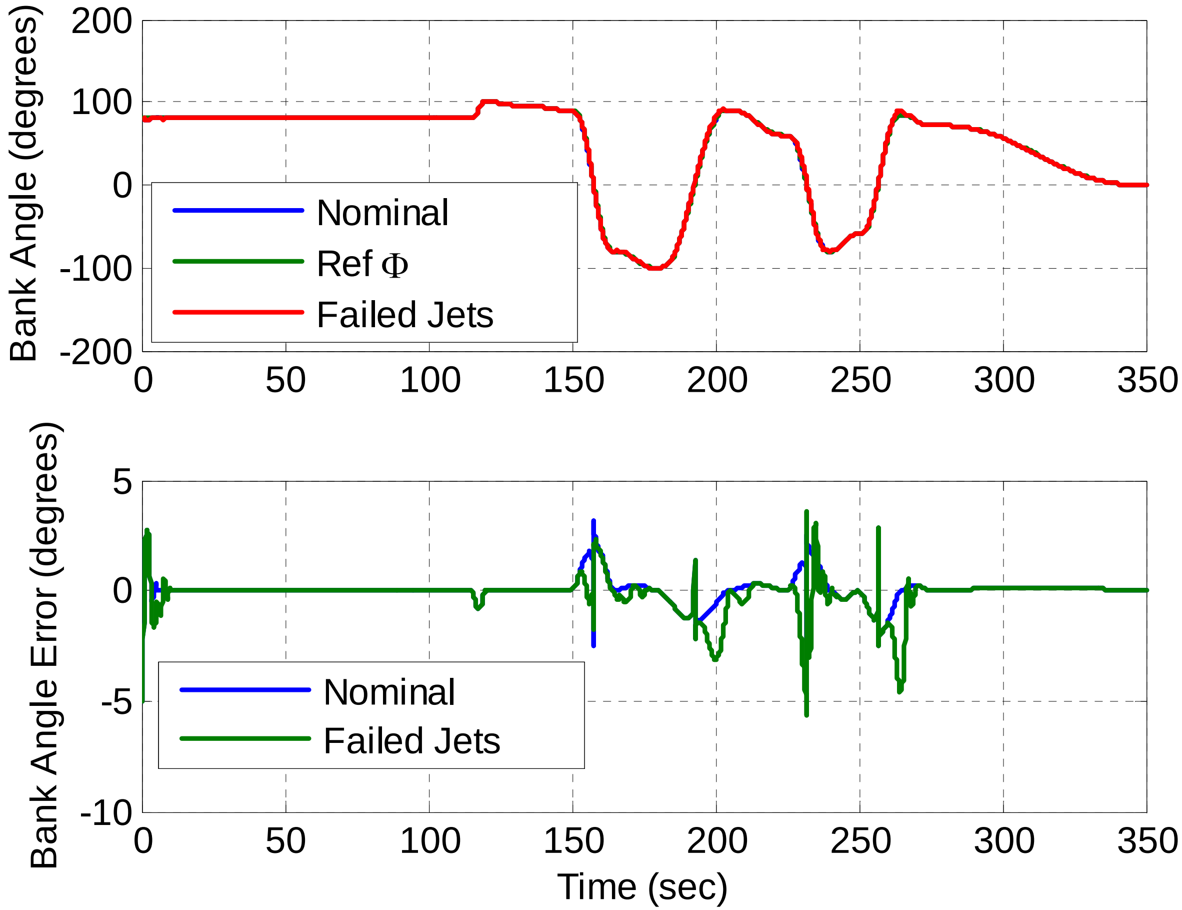
Results

Case 1 : RCS Jets



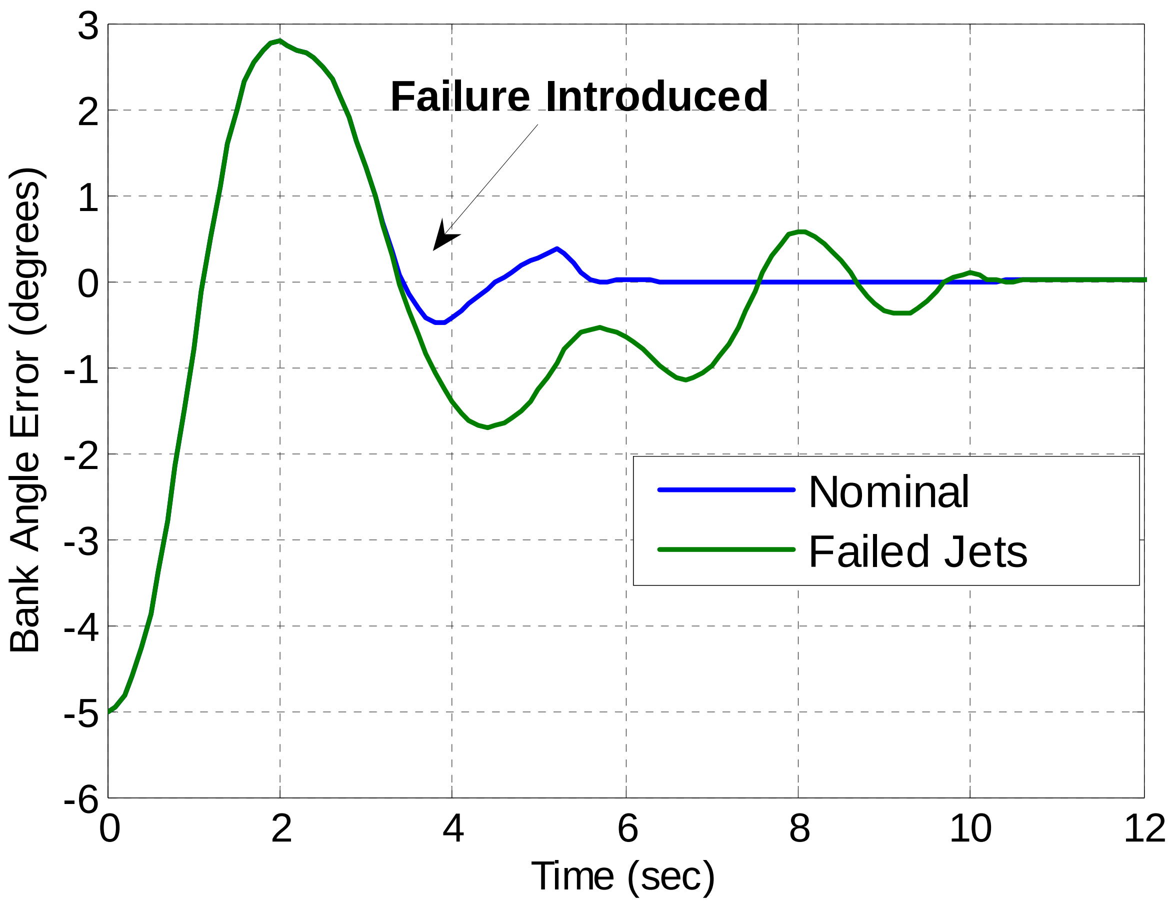
Case 1 : Bank Angle Profile

(4 jets off)



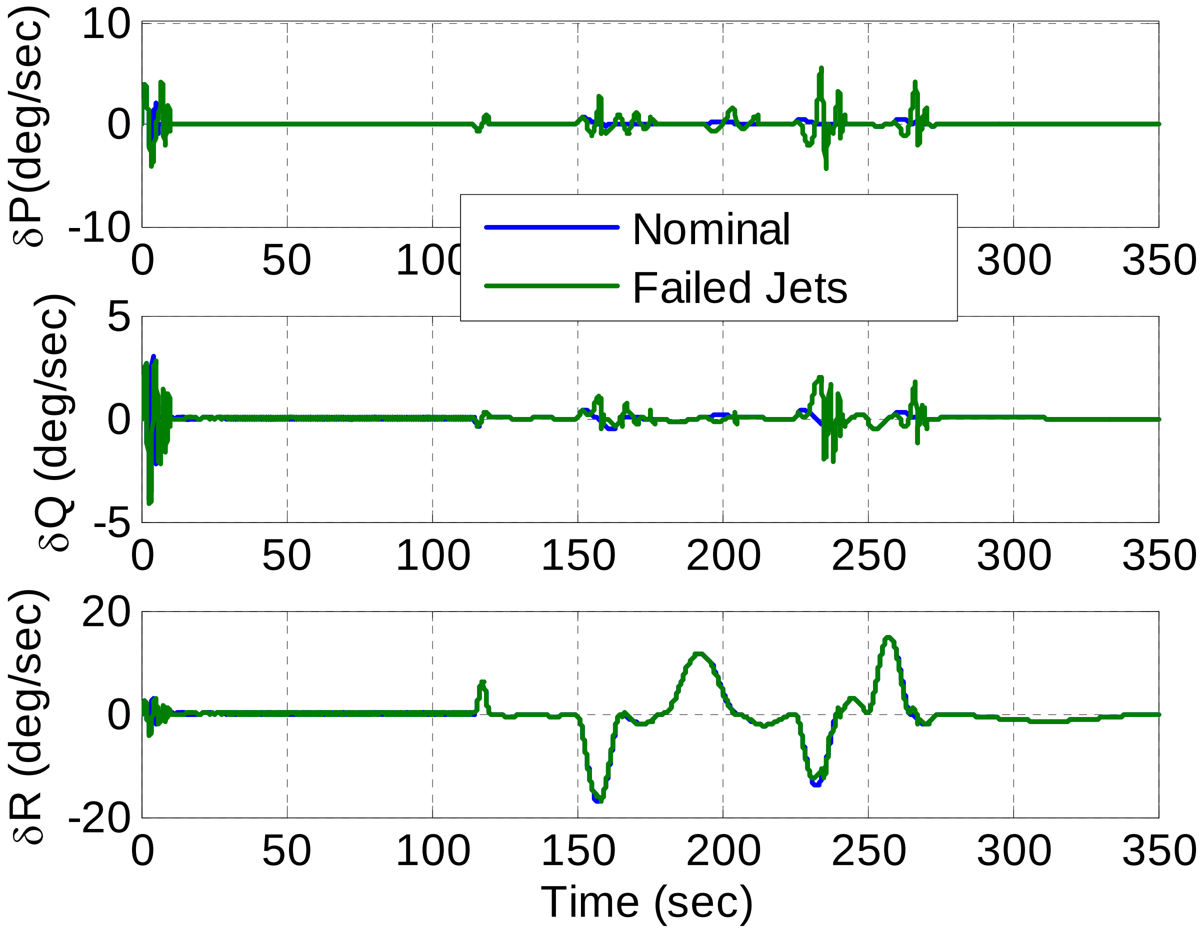
Case 3 : Bank Angle Response to Failure Introduction

(4 jets off)

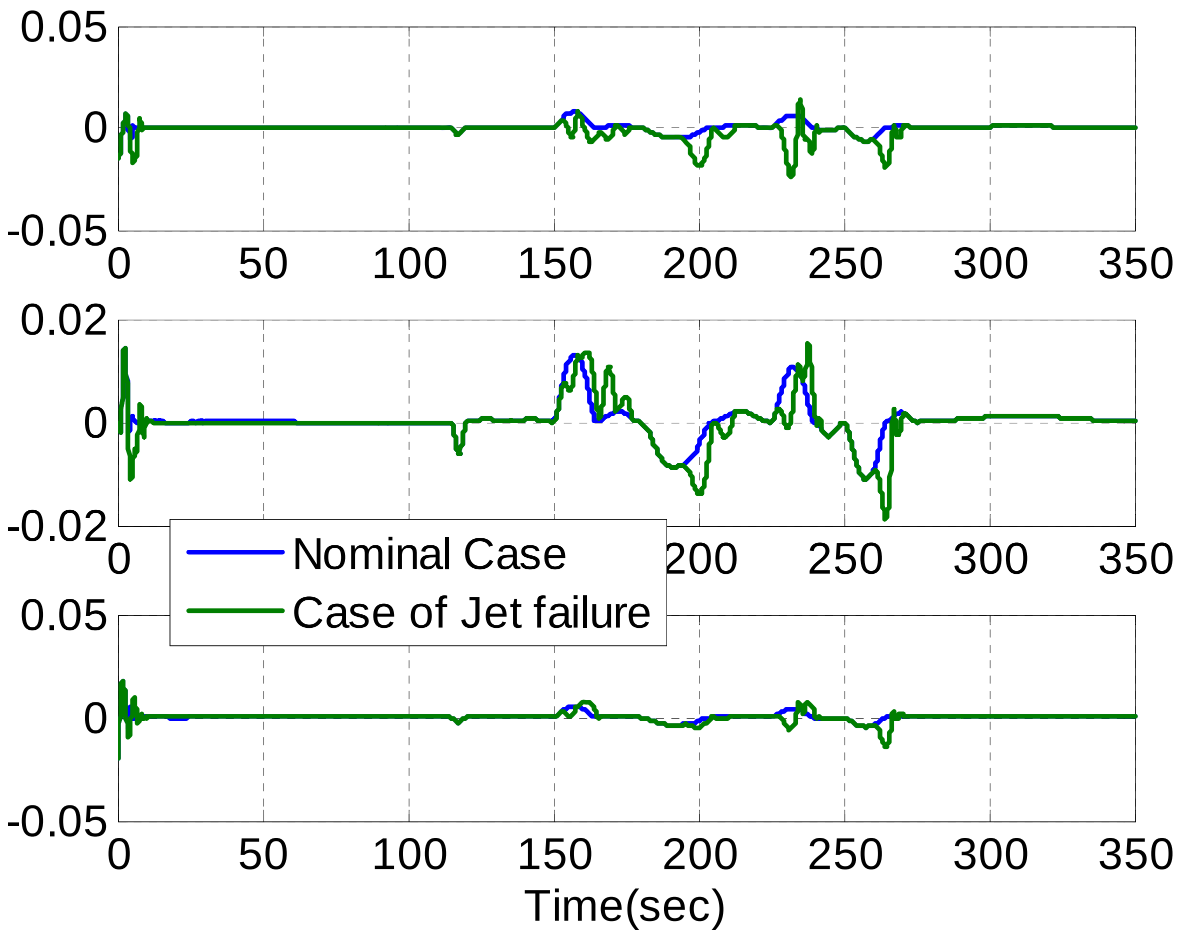


Case 1 : Angular Velocities

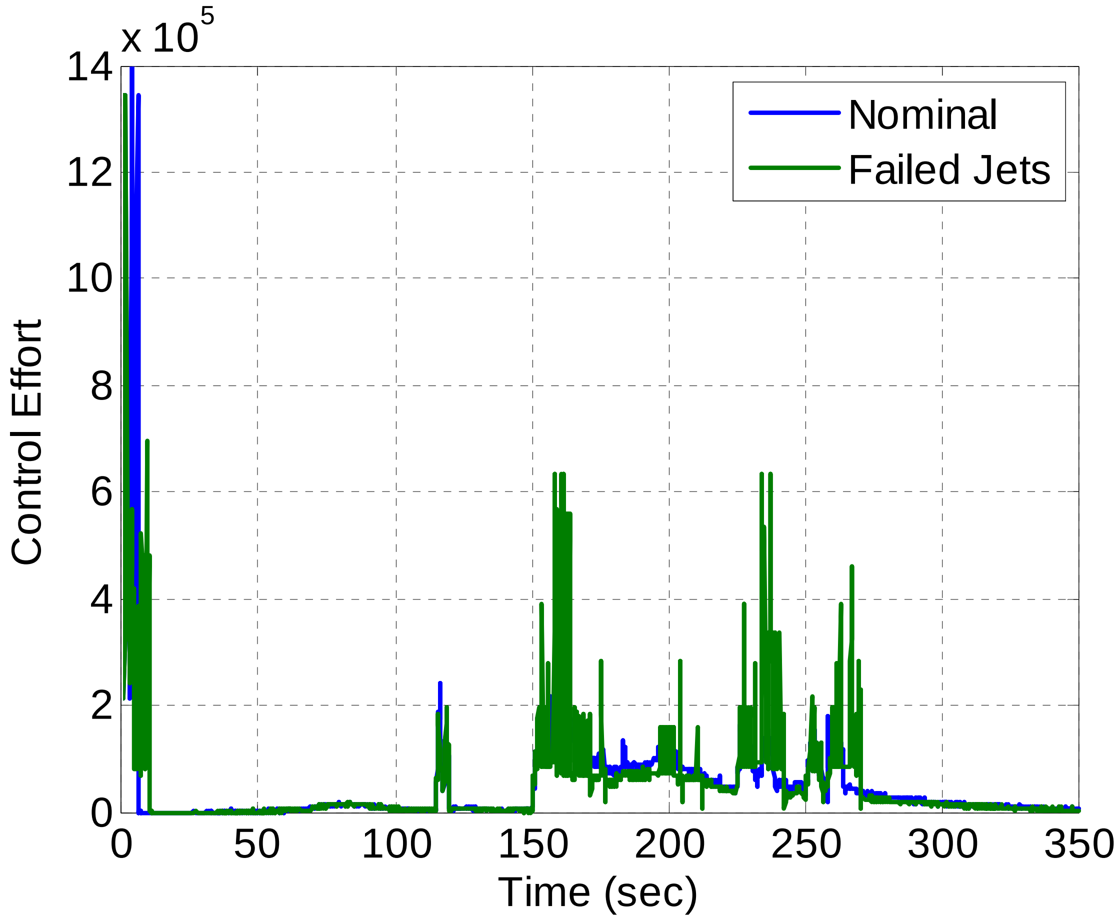
(4 jets off)



Case 1 : MRPs
(4 jets off)

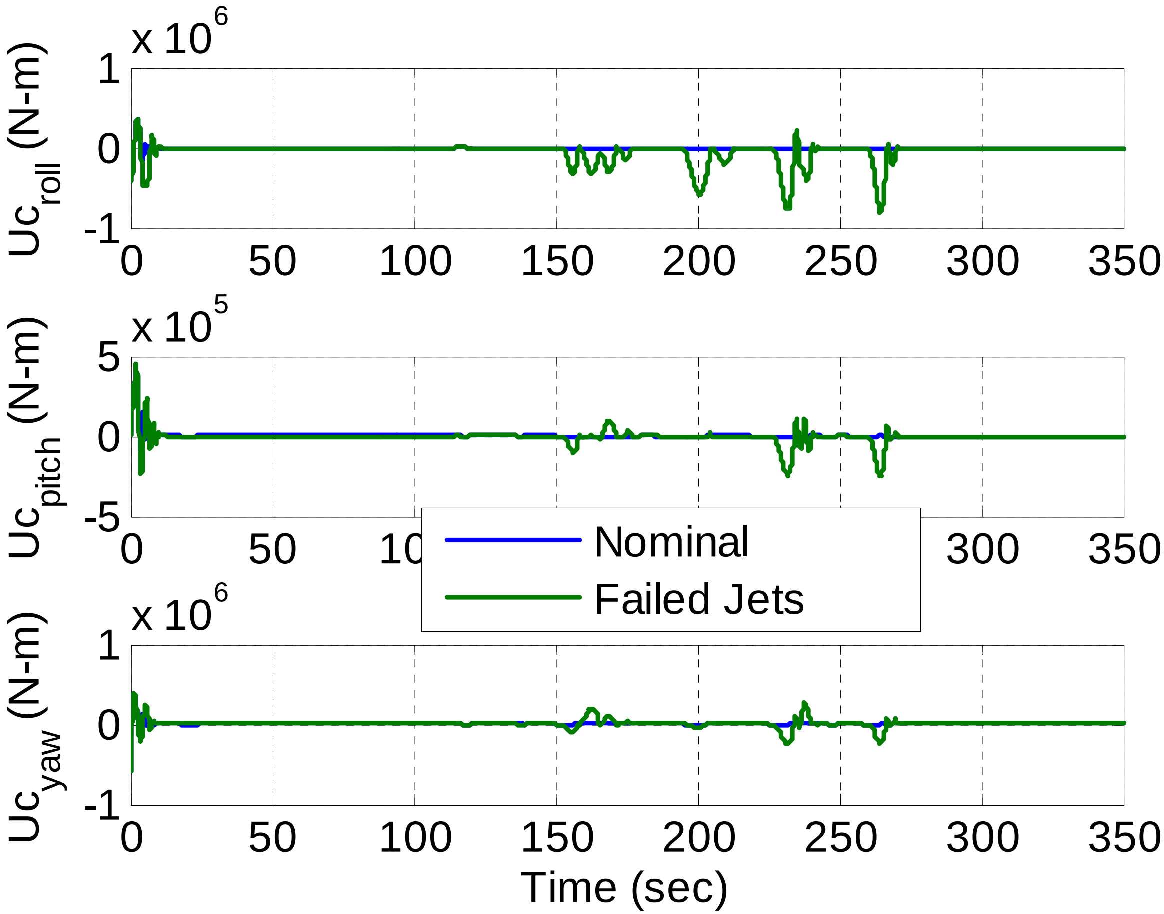


Case 1 : Control Effort
(4 jets off)

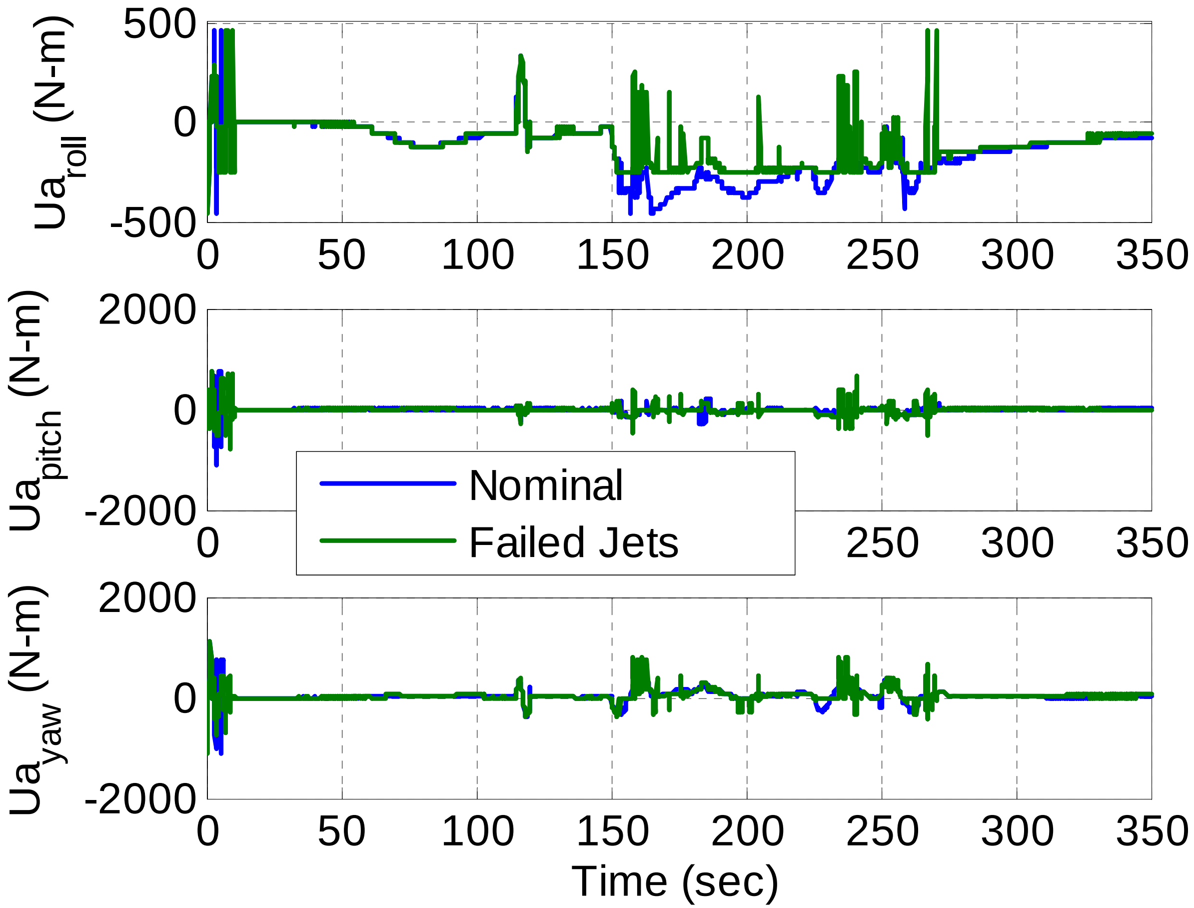


Case 1 : Commanded Moments

(4 jets off)

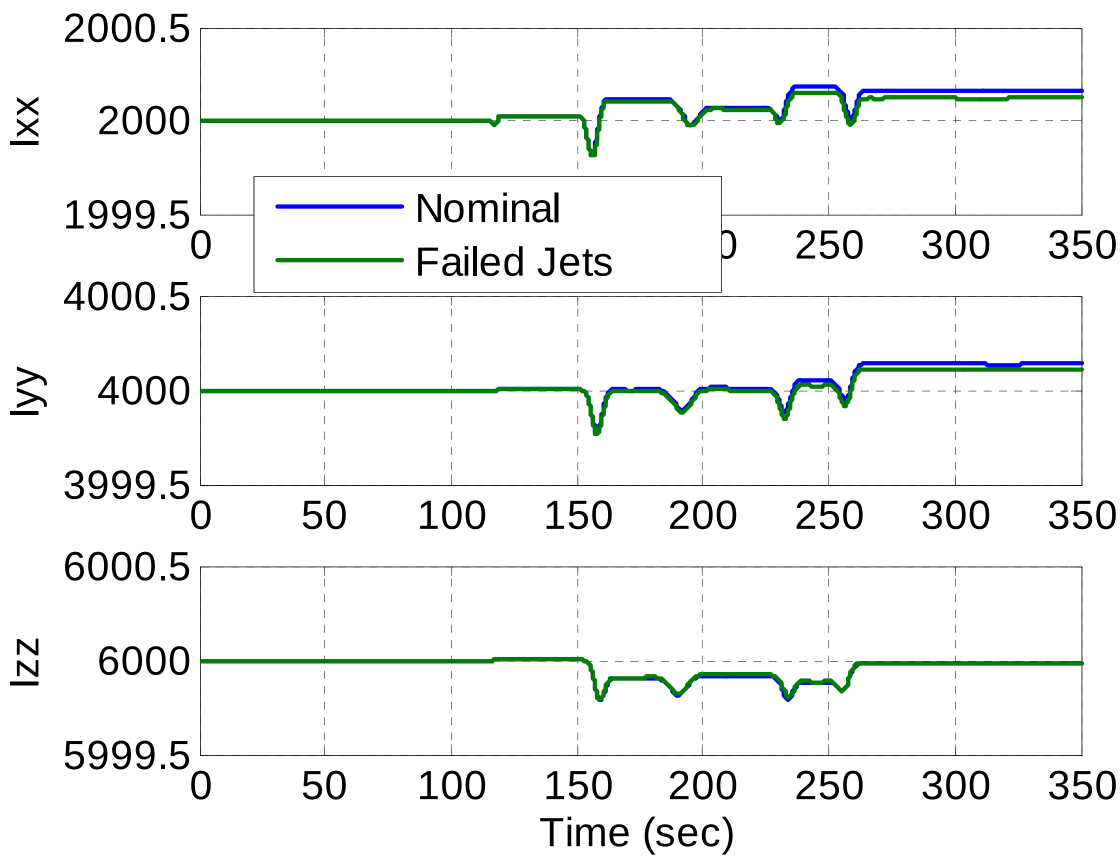


Case 1 : Applied Moments
(4 jets off)

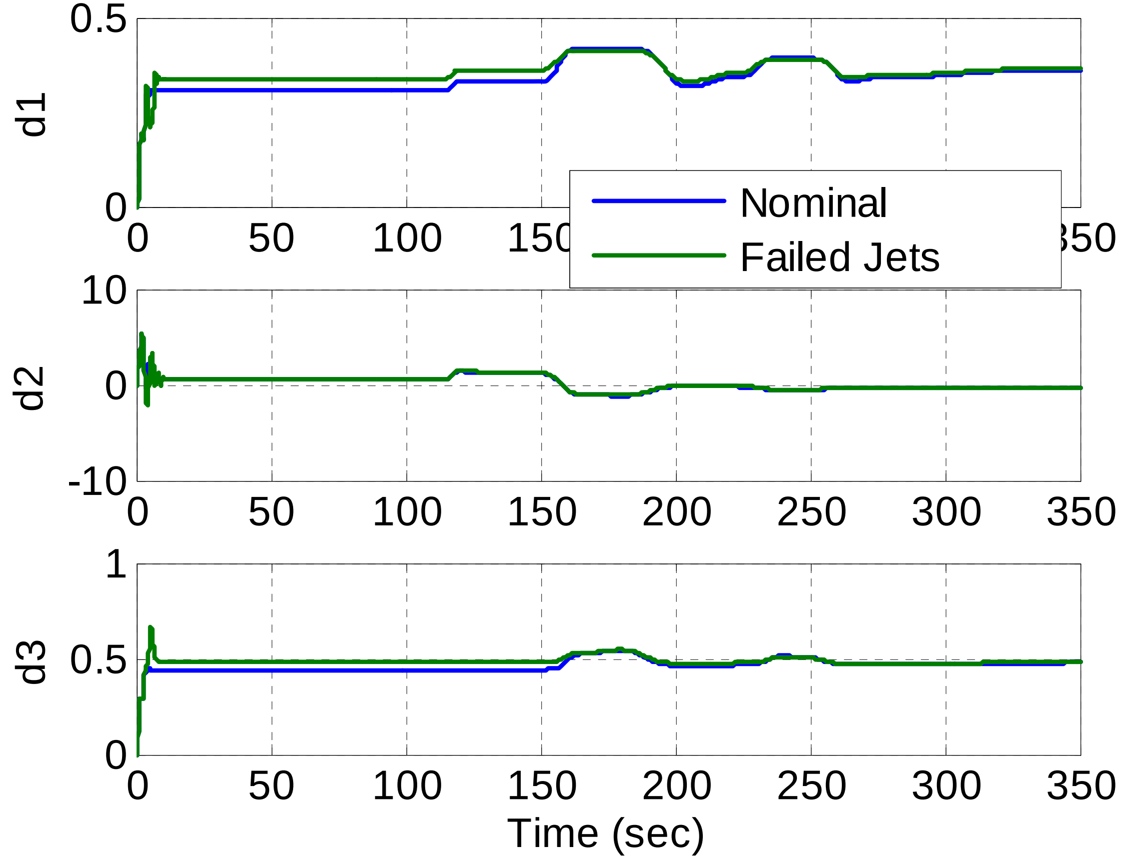


Case 1 : Adaptive Parameters

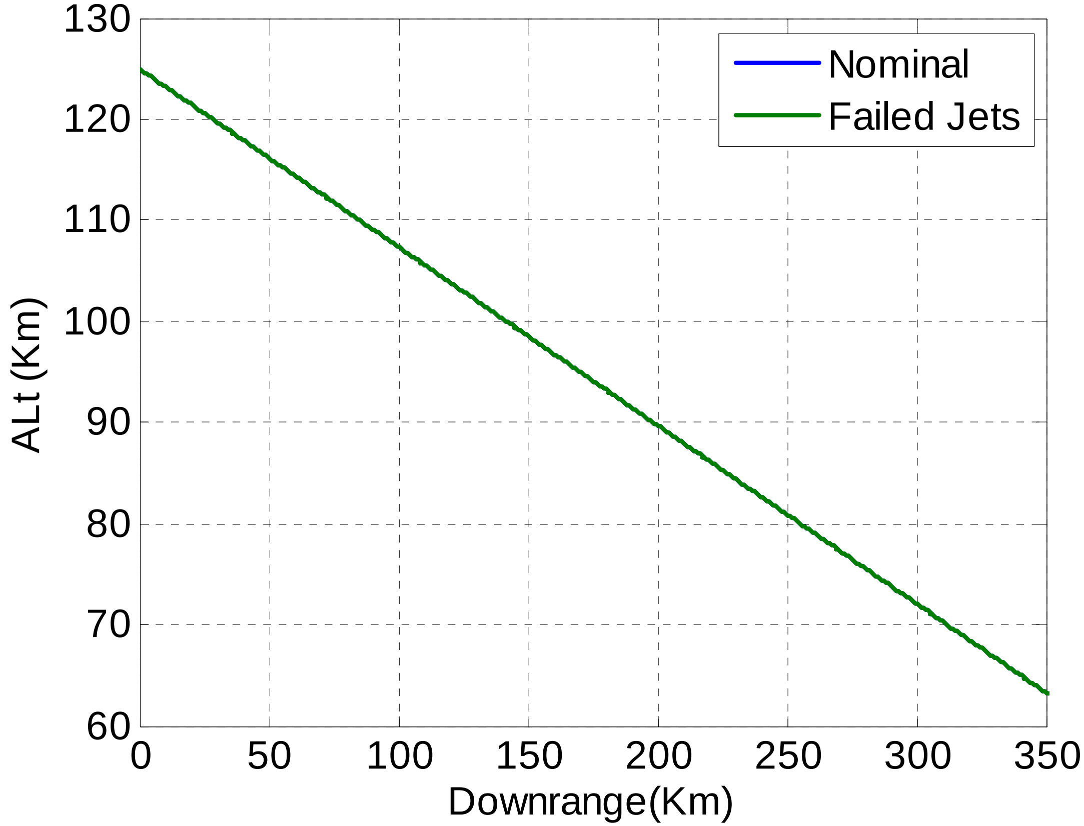
(4 jets off)



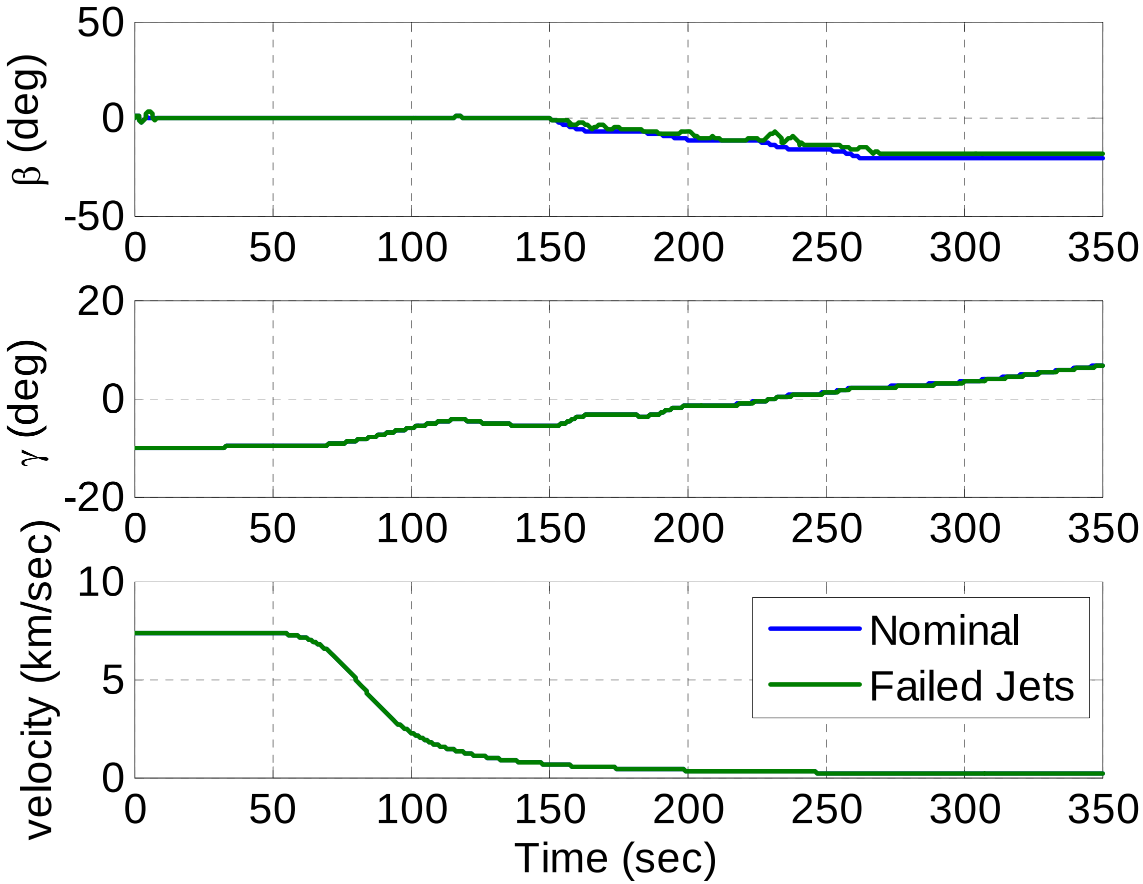
Case 1 : Adaptive Parameters
(4 jets off)



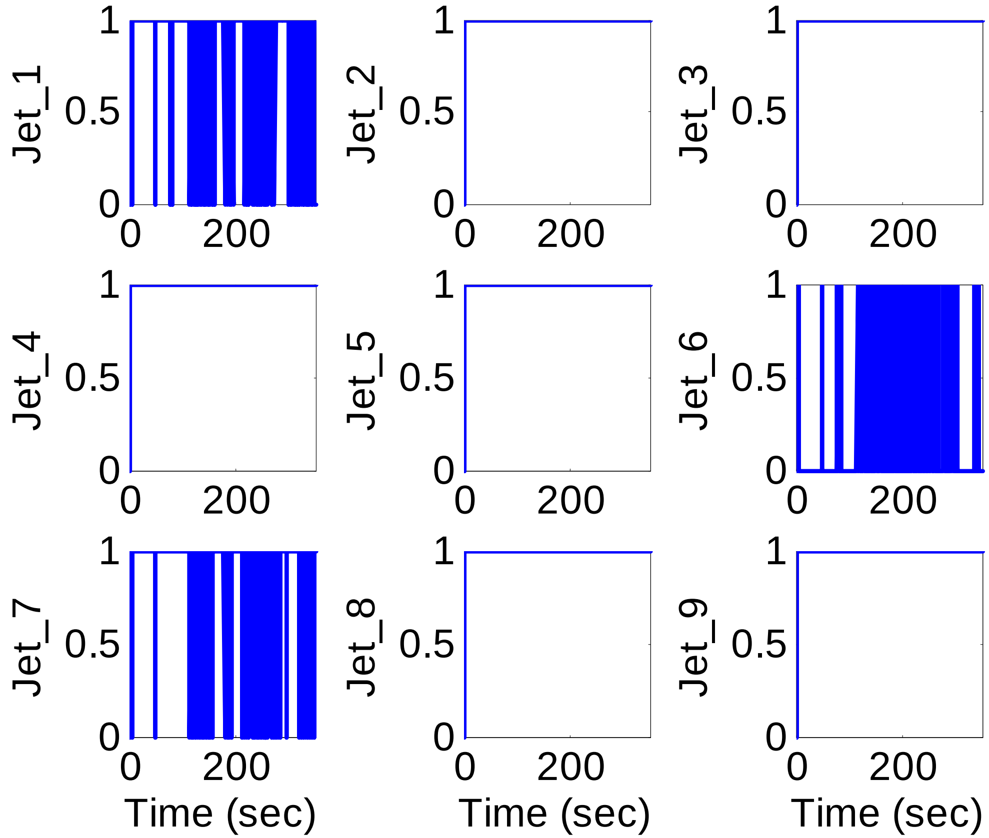
Case 1 : Altitude v/s Downrange
(4 jets off)



Case 1 : Translational States
(4 jets off)

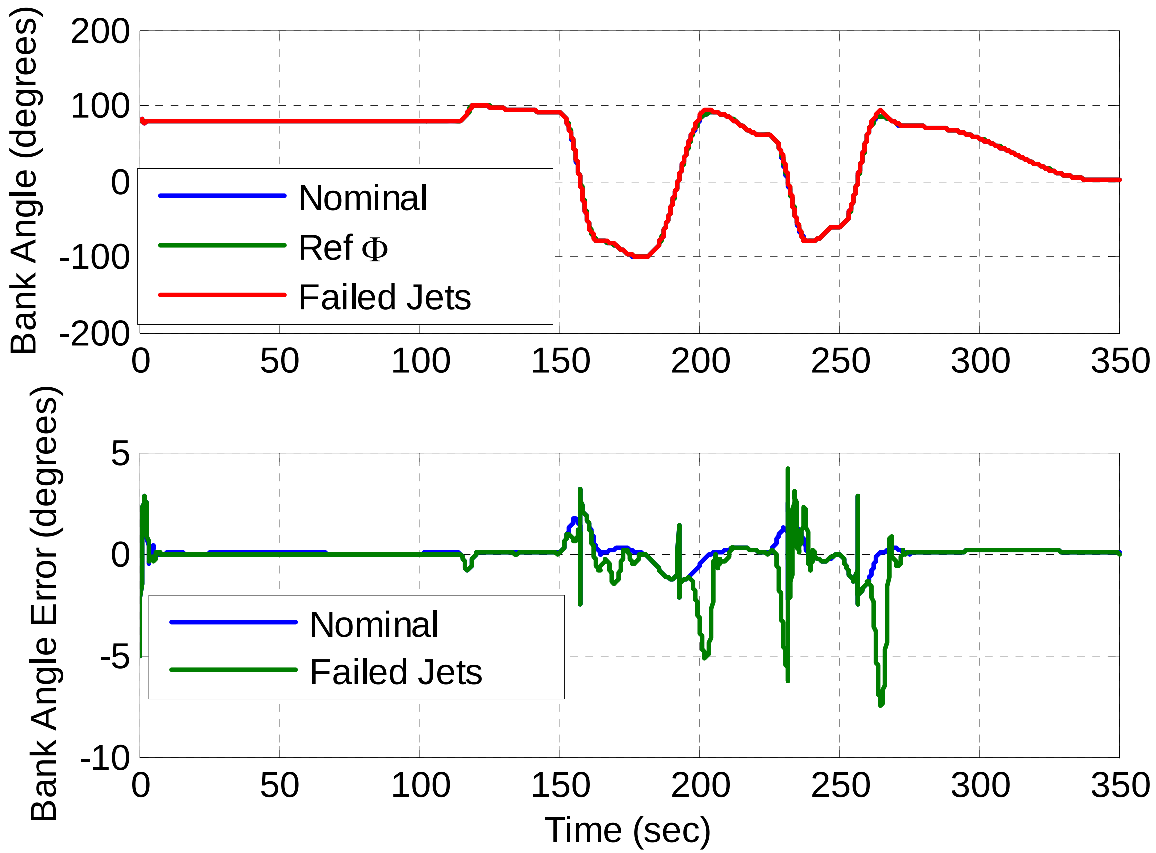


Case 2 : RCS Jets



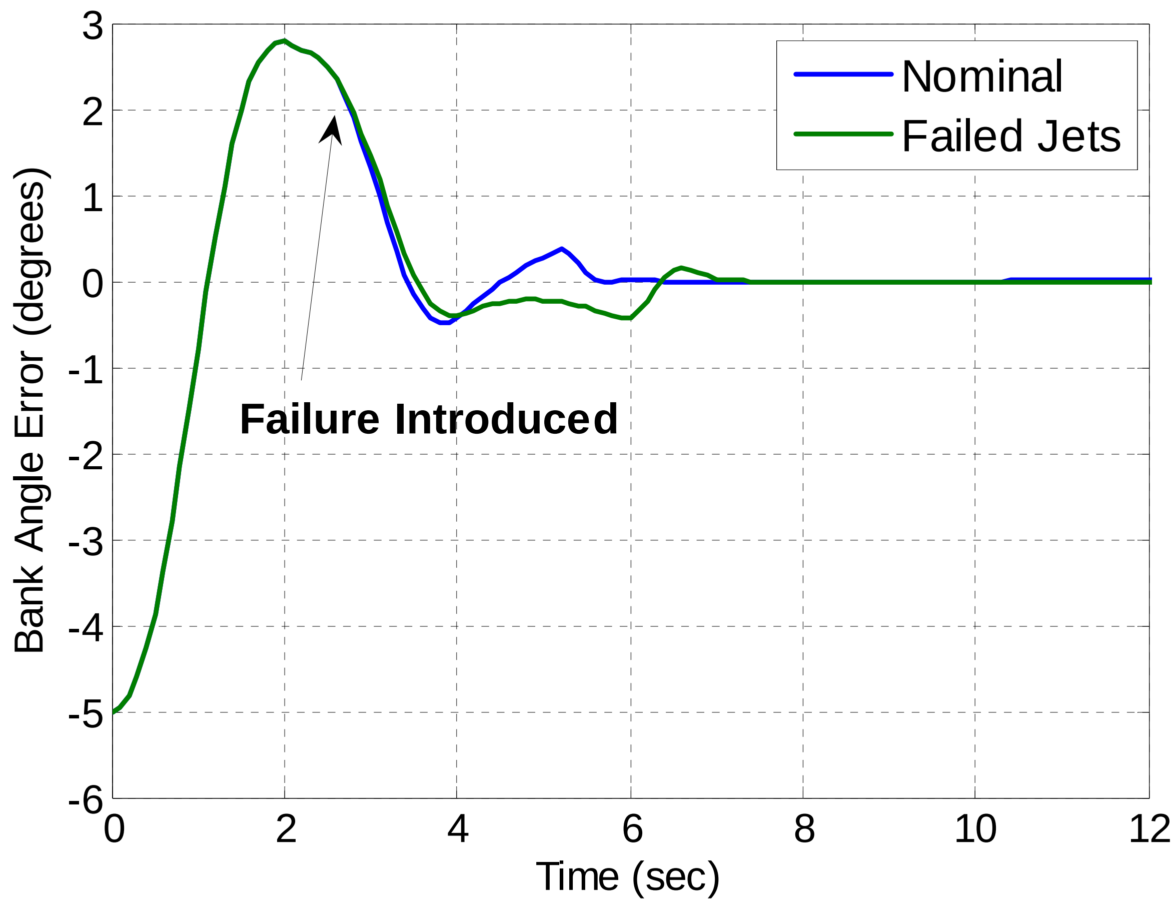
Case 2 : Bank Angle Profile

(7 jets on)



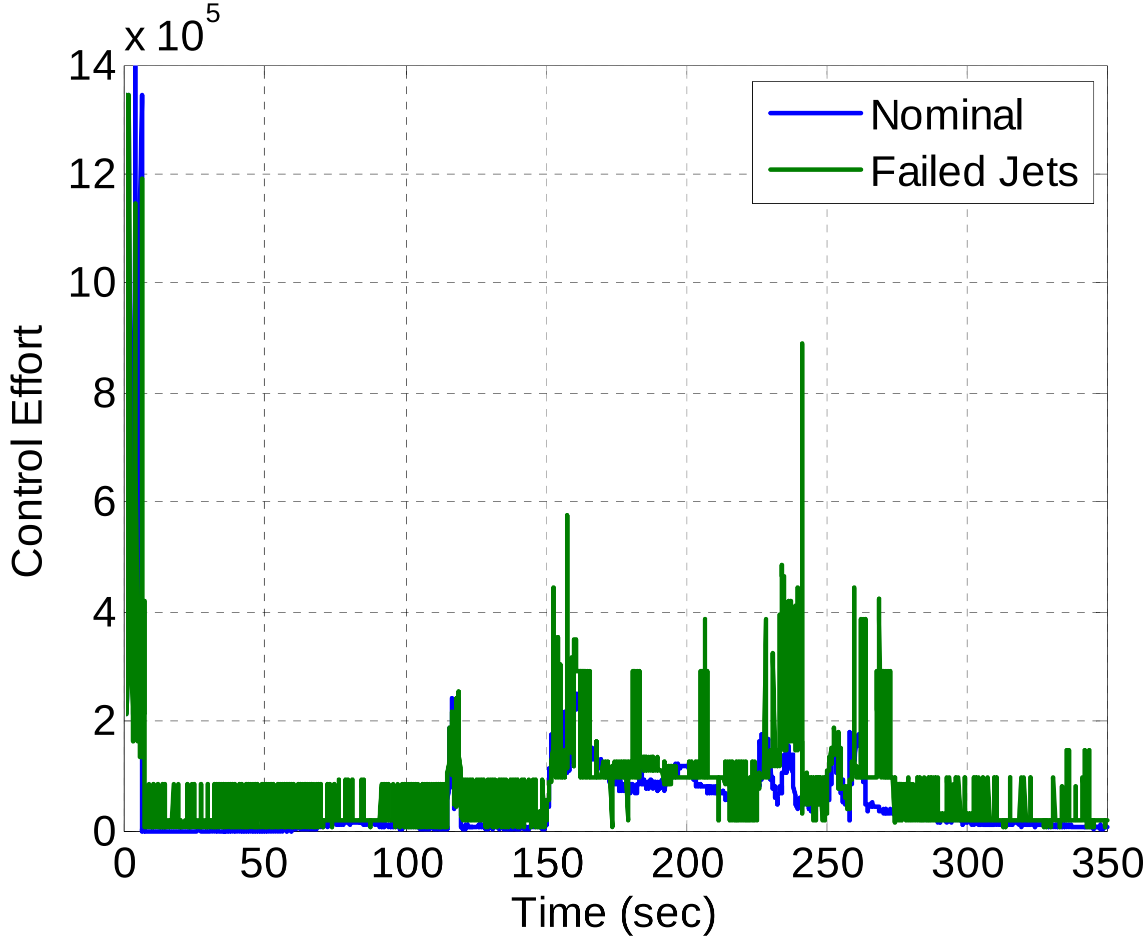
Case 3 : Bank Angle Response to Failure Introduction

(7 jets on)



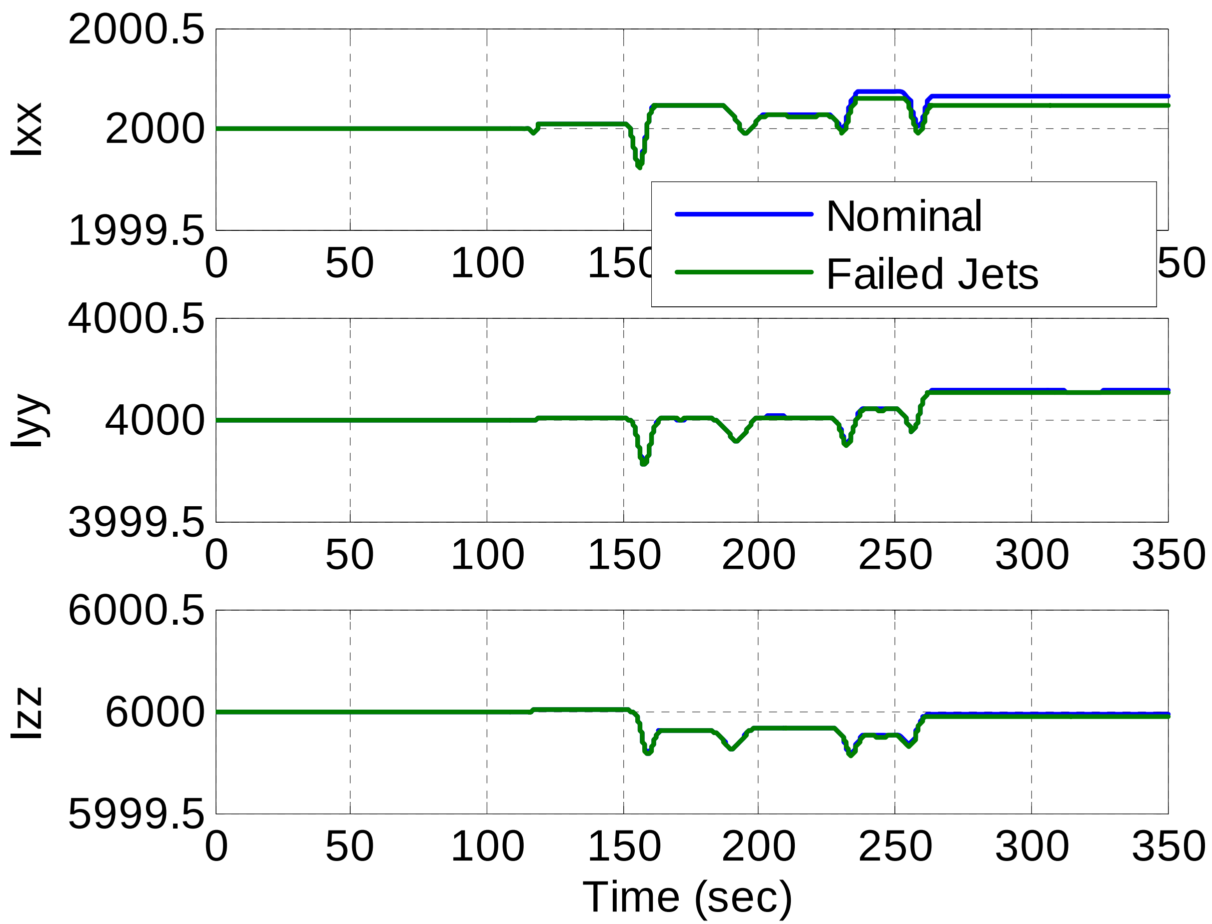
Case 2 : Control Effort

(7 jets on)

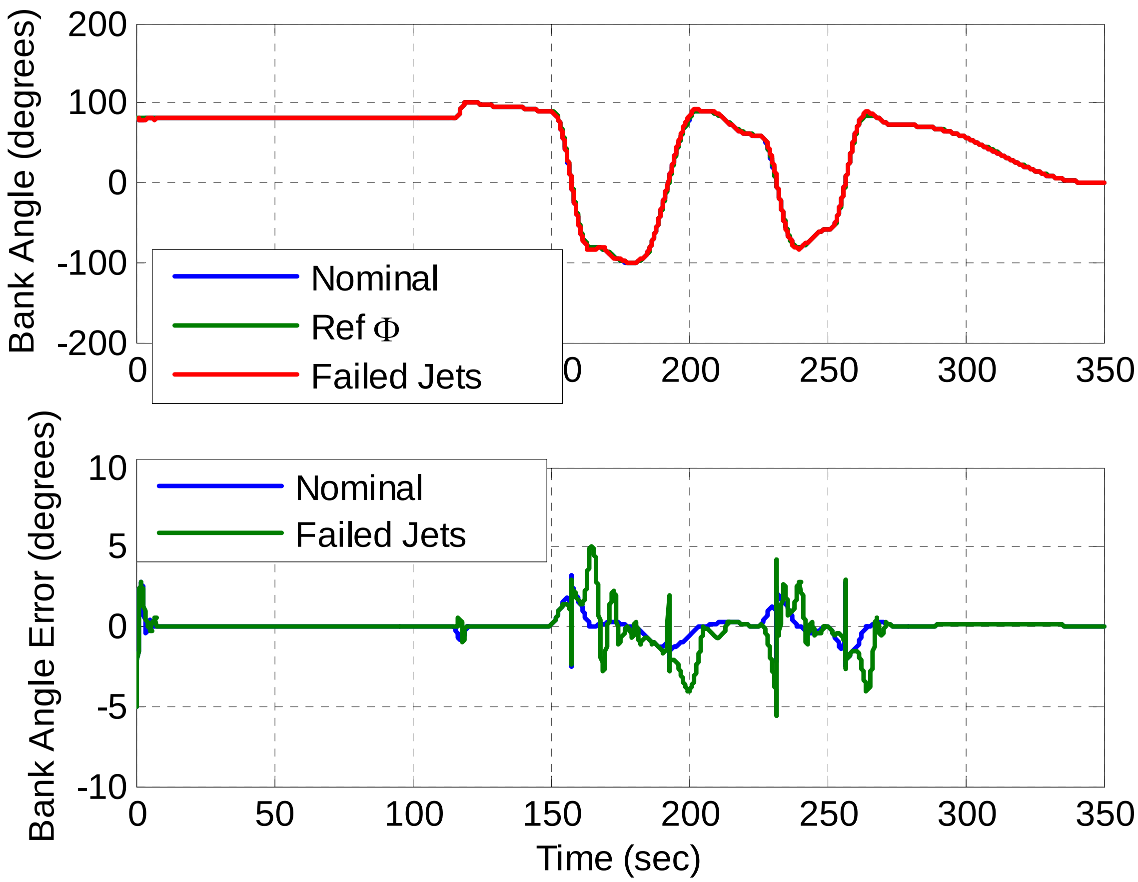


Case 2 : Adaptive Parameters

(7 jets on)

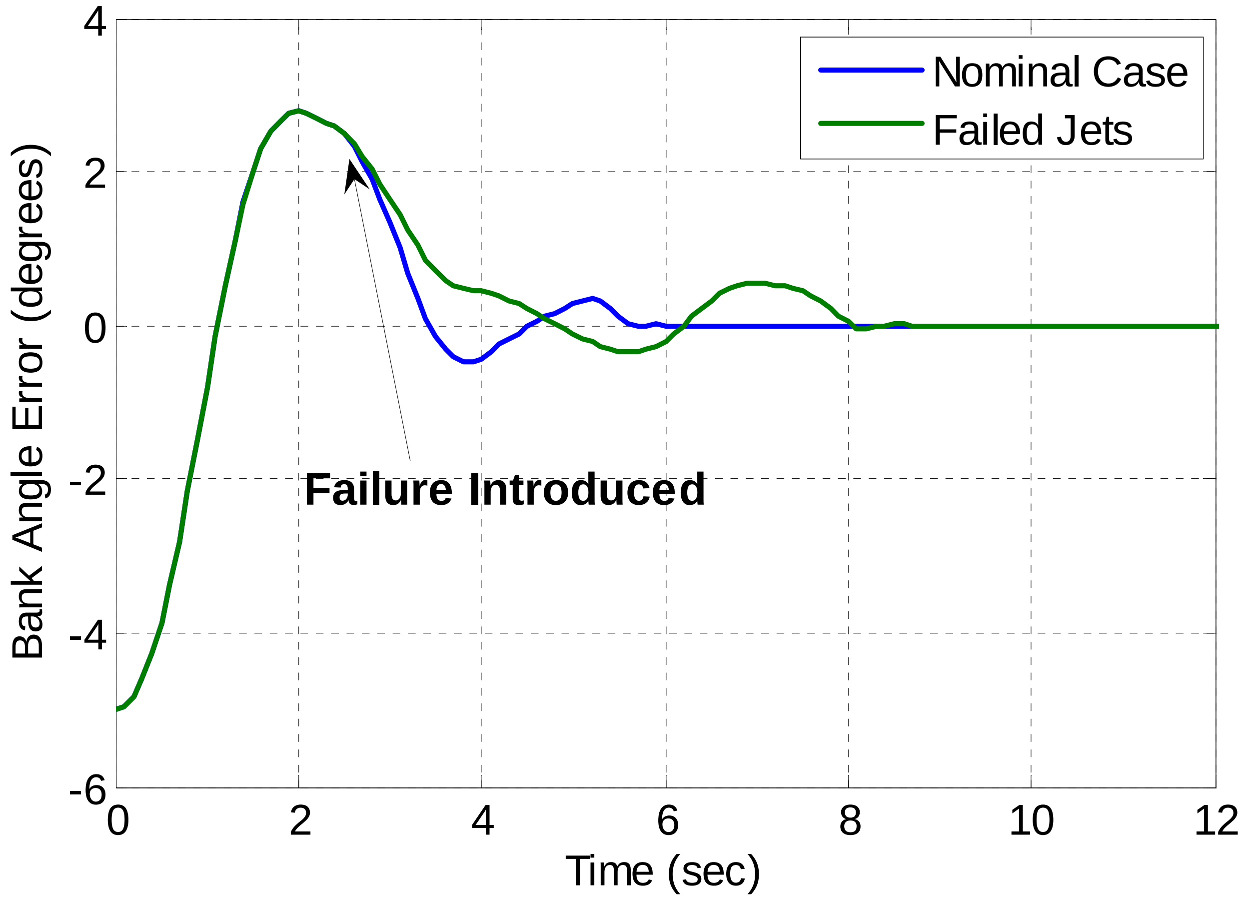


Case 3 : Bank Angle Profile
(Reduced Torque)



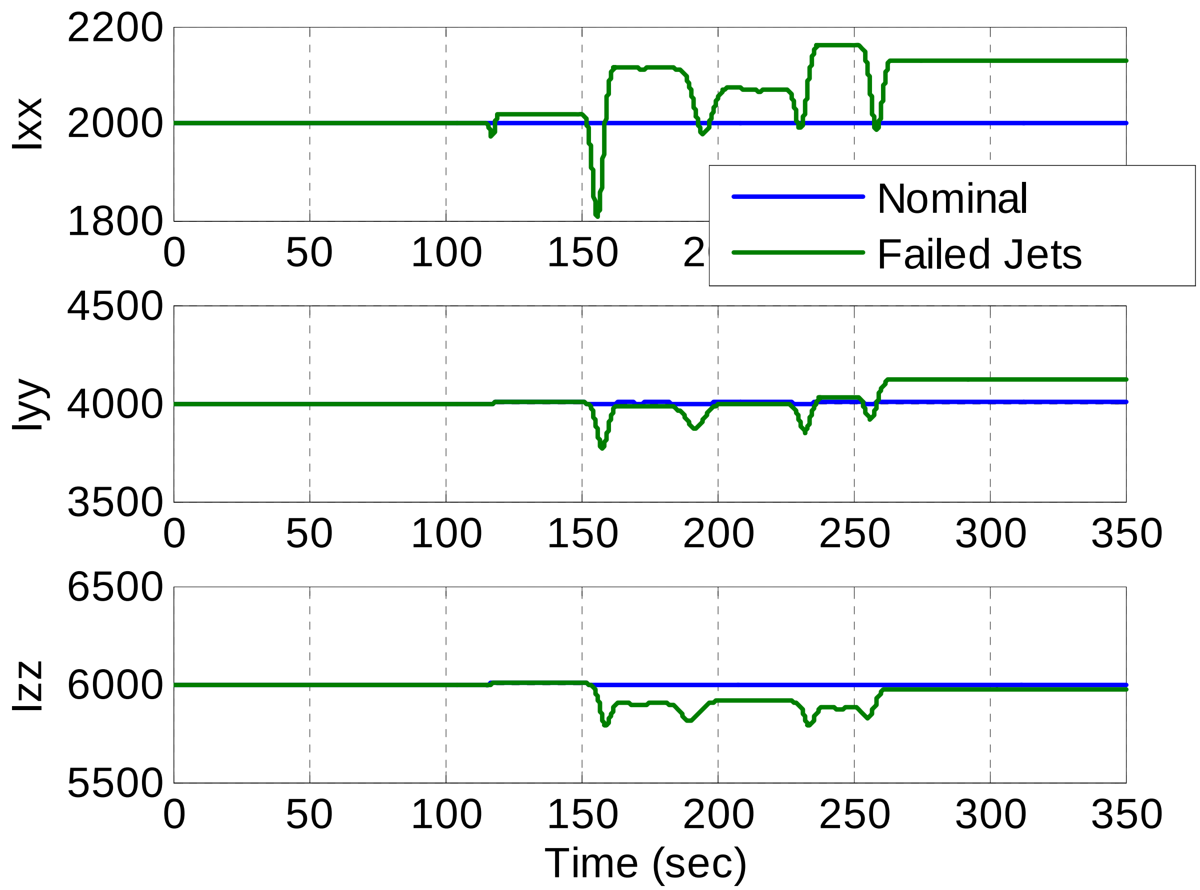
Case 3 : Bank Angle Response to Failure Introduction

(Reduced Torque)



Case 3 : Adaptive Parameters

(Reduced Torque)



Conclusions

- SAMI based update laws perform very well to handle the plant parameter uncertainties
- Controller provides good trajectory tracking performance even in case of failures
- Control algorithm is capable of handling different kinds of failures

Future Work

- Integration of adaptive trajectory generation
- Some algorithm is required to detect failure