Quantitative Local Analysis of Nonlinear Systems using Sum-of-Squares Decompositions'

Andrew Packard, Peter Seiler, Ufuk Topcu, and Gary J. Balas

MUSYN Inc.
P.O. Box 13377
Minneapolis, MN 55414-5377
balas@musyn.com

30 September-1 October 2009

Worst-Case Performance

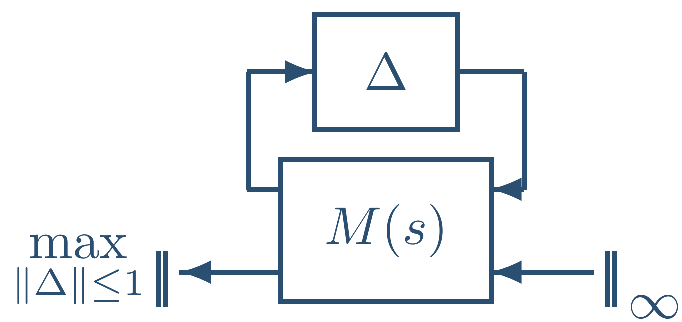
Outline

- Worst-Case Performance for LFTs
- Divide and Conquer, based on upper and lower bounds
- Lower Bound
- Upper Bound
- Correlating two uncertain parameters
- Application to NASA X-38 Crew Return Vehicle

Worst-Case Performance Setup

Systems under consideration:

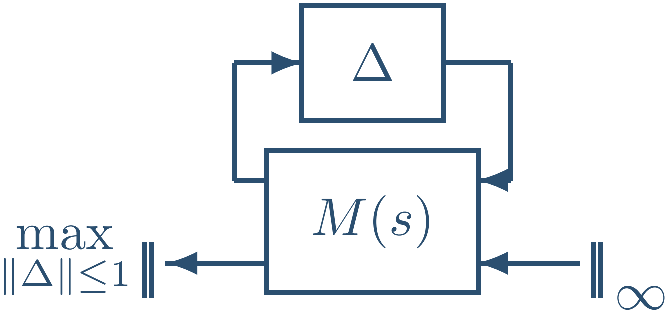
- Linear systems with
  - parametric uncertainty, and/or
  - unmodelled dynamics
- Performance objective involves keeping specific transfer functions "small"



Uncertain relationship between d and e is

$$
\begin{aligned}
e &= \left[M_{22} + M_{21}\Delta\left(I - M_{11}\Delta\right)M_{12}\right]d \\
&=: F_u\left(M,\Delta\right)d \\
&=: T_{d\to e}(\Delta)d
\end{aligned}
$$

Worst-Case Performance Setup (cont'd)



In this diagram, the known elements are separated from unknown elements in this feedback connection.

- Known contains: nominal plant model, controller, manner in which uncertainty enters, disturbances, errors
- Unknown contains: uncertainty in parameters of differential equations, unmodeled dynamics

Why Not Robust Stability?

- Before stability is lost, performance degrades unacceptably.
- Worst-Case Performance, over parameter uncertainties and unmodelled dynamics, is easy to motivate, and uses same mathematical tools as Robust Stability calculations.
- For problems with only real parametric uncertainty modeled, Robust Stability quantities can be discontinuous (in data and frequency)

Justification

Question: is the quantity

$$\max_{\text{allowable } \Delta} \max_{\omega} |T_{d \to e}(\Delta, \omega)|$$

a good measure of "worst-case behavior?"

- Honeywell SRC applied this type of analysis to Shuttle in 1984+.
- Frequency domain criterion – connection to time domain is less precise than usually desired.
- T must be chosen carefully to reflect variables of interest.
- One strategy: assess and normalize the nominal level of performance being achieved by current controller

Performance Objective: Weighted Norm

One method to pick T:

- Start with nominal model, and candidate controller
- Plot closed-loop frequency response from commands (r) and gusts (g) to tracking error e

$$e = G^{\text{nom}} \begin{bmatrix} r \\ g \end{bmatrix}$$

- Find simple weighting functions W_1, W_2 such that

$$|W_1(j\omega)G_1^{\text{nom}}(j\omega)| \approx 1, \qquad |W_2(j\omega)G_2^{\text{nom}}(j\omega)| \approx 1$$

  for all frequencies
- Hence, the nominal model with controller achieves weighted closed-loop performance

$$\max_{\omega} \left\| \begin{matrix} W_1(j\omega)G_1^{\text{nom}}(j\omega) \\ W_2(j\omega)G_2^{\text{nom}}(j\omega) \end{matrix} \right\| \approx 1.4$$

Performance Objective: Weighted Norm (cont'd)

- For perturbed system, use this objective, and see how "bad" it can be made, relative to 1.4 (nominal performance)

$$T(\Delta) := \begin{bmatrix} W_1 G_1(\Delta) \\ W_2 G_2(\Delta) \end{bmatrix}$$

- Find

$$\max_{\Delta \text{ allowable}} \|T(\Delta)\|_\infty$$

Uncertainty Model

Treat two types of model uncertainty:

1. Uncertain real-valued parameters in differential equation model
2. Unmodeled dynamics, with frequency dependent bounds

Normalizing (absorbing offsets and weights into "known" part of system) yields uncertain matrices of the form

$$\Delta = \operatorname{diag}\left[\delta_1 I_{k_1}, \cdots, \delta_n I_{k_n}, \hat{\Delta}_1(s), \cdots, \hat{\Delta}_f(s)\right]$$

each real and transfer function parameter assumed to satisfy

$$|\delta_i| \leq 1, \quad \max_\omega \left|\hat{\Delta}_i(j\omega)\right| \leq 1$$

Easy Fact: Given any complex number γ, with $|\gamma| \leq 1$, and any frequency $\omega_0 > 0$, there is a stable transfer function $\hat{\Delta}(s)$ satisfying

$$\max_\omega \left|\hat{\Delta}(j\omega)\right| = |\gamma|, \qquad \hat{\Delta}(j\omega_0) = \gamma$$

Implication: Ultimately treat the Worst-Case-Performance

Reduction to Constant-Matrix

For the problem, there is a "worst" frequency. At that frequency, the mathematical problem is a single matrix problem. **How?** Allowable Δ satisfy

$$\Delta = \operatorname{diag}\left[\delta_1 I_{k_1}, \cdots, \delta_n I_{k_n}, \hat{\Delta}_1(s), \cdots, \hat{\Delta}_f(s)\right]$$

with

$$|\delta_i| \leq 1, \quad \max_{\omega} \left|\hat{\Delta}_i(j\omega)\right| \leq 1$$

Original problem is

$$\max_{\Delta(s) \text{ allowable}} \; \max_{\omega} \left| F_u\left(\hat{M}(j\omega), \Delta(j\omega)\right)\right|$$

Reduction to Constant-Matrix (cont'd)

Interchange the "max"

$$\max_{\omega} \max_{\Delta(s)\ \text{allowable}} \left| F_u\left(\hat{M}(j\omega), \Delta(j\omega)\right)\right|$$

But at any fixed frequency, the transfer function entries of $\Delta(j\omega)$ can be any complex number with magnitude ≤ 1. So, at each frequency, view Δ as a constant matrix (real and complex entries)

$$\max_{\omega} \underbrace{\max_{\Delta} \left| F_u\left(\hat{M}(j\omega), \Delta\right)\right|}_{\text{constant matrix problem}}$$

Grid frequency range, based on domain-specific expertise, with finite frequencies, $\omega_1, \omega_2, \ldots, \omega_N$, and solve only there

$$\max_{1\leq i\leq N} \max_{\Delta} \left| F_u\left(\hat{M}(j\omega_i), \Delta\right)\right|$$

Worst-Case Performance: Constant Matrix

Focus on constant matrix problem (and solve at many frequencies).

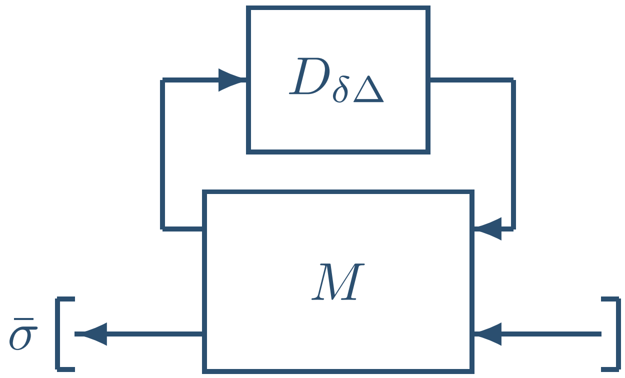
- $M \in \mathbf{C}^{(k+n_e)\times(k+n_d)}$, complex since this will typically be response of $\hat{M}$ at a certain frequency.
- Integers $k_1, \ldots, k_n, k_{n+1}, \ldots, k_{n+f}$, with $k := k_1 + \cdots + k_{n+f}$.
- n uncertain real parameters, $\delta_1, \ldots, \delta_n$, each varies independently in range, $a_i \leq \delta_i \leq b_i$.
- f uncertain matrices, $\Delta_1 \in \mathbf{C}^{k_{n+1}\times k_{n+1}}, \ldots, \Delta_f \in \mathbf{C}^{k_{n+f}\times k_{n+f}}$.
- Associated with the indices k_i, $\mathcal{D}$ denotes the operation which takes $\delta := (\delta_1, \ldots, \delta_n)$ and $\Delta := (\Delta_1, \ldots, \Delta_f)$ into the $k \times k$ block-diagonal matrix.

$$\mathcal{D}_{\delta\Delta} := \operatorname{diag}\left[\delta_1 I_{k_1}, \ldots, \delta_n I_{k_n}, \Delta_1, \ldots, \Delta_f\right]$$

Worst-Case Performance: Constant Matrix (cont'd)

Problem: Given M and the intervals $[a_i \; b_i]$, estimate lower and upper bounds for

$$\max_{\substack{a_i \le \delta_i \le b_i \\ \bar{\sigma}(\Delta_i) \le 1}} \bar{\sigma}\left[M_{22} + M_{21}\mathcal{D}_{\delta\Delta} \left(I - M_{11}\mathcal{D}_{\delta\Delta} \right)^{-1} M_{12} \right]$$



Divide-and-Conquer: Real Parameters

Given M, a and b. Need easily computable bounds L and U

- Lower bound, $L(a,b,M)$, satisfying

$$L(a,b,M) \leq \max_{\substack{a_i \leq \delta_i \leq b_i \\ \bar{\sigma}(\Delta_i) \leq 1}} \bar{\sigma}\left[M_{22} + M_{21}\mathcal{D}_{\delta\Delta}\left(I - M_{11}\mathcal{D}_{\delta\Delta}\right)^{-1} M_{12} \right]$$

- Upper bound, $U(a,b,M)$, satisfying

$$\max_{\substack{a_i \leq \delta_i \leq b_i \\ \bar{\sigma}(\Delta_i) \leq 1}} \bar{\sigma}\left[M_{22} + M_{21}\mathcal{D}_{\delta\Delta}\left(I - M_{11}\mathcal{D}_{\delta\Delta}\right)^{-1} M_{12} \right] \leq U(a,b,M)$$

Bounds L and U presented today have property that for $a = b$, L is reasonably close to U. For $a < b$, gap increases, and a "Divide-and-Conquer" reduces the gap.

Divide-and-Conquer: Real Parameters (cont'd)

Steps of *Divide & Conquer*

1. Initialize list of cubes to the initial cube,

$$[a_1, b_1] \times [a_2, b_2] \times \cdots [a_n, b_n]$$

2. Call upper and lower bounds computations on the initial cube.
3. Find cube in ACTIVE list with largest upper bound.
4. Split cube along longest edge into two cubes, compute bounds on both of these new cubes, and replace.
5. Make any current cube whose upper bound is lower than another cube's lower bound INACTIVE. Go to 3.

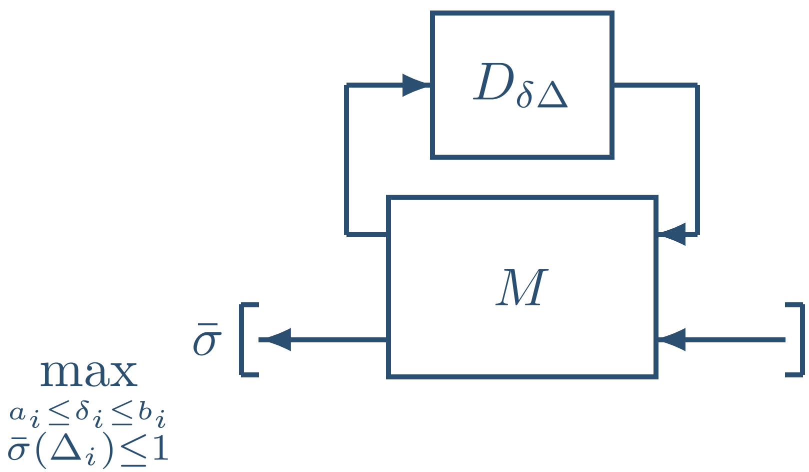
We do not divide on complex $\{\Delta_i\}$. Simply accept gap in L and U that exists even when $a = b$.

Lower Bounds: Iterations

Problem: Given M and the intervals $[a_i \; b_i]$, estimate lower and upper bounds for

$$\max_{\substack{a_i \le \delta_i \le b_i \\ \bar{\sigma}(\Delta_i) \le 1}} \bar{\sigma} \left[M_{22} + M_{21} \mathcal{D}_{\delta\Delta} \left(I - M_{11} \mathcal{D}_{\delta\Delta} \right)^{-1} M_{12} \right]$$

drawn as



Recall,

$$\mathcal{D}_{\delta\Delta} := \operatorname{diag} \left[\delta_1 I_{k_1}, \ldots, \delta_n I_{k_n}, \Delta_1, \ldots, \Delta_f \right]$$

Lower Bounds: Iterations (cont'd)

Strategy – coordinatewise across δ and Δ, but with different approaches:

1. Hold complex uncertainties (Δ_j) fixed, maximize over real parameters using coordinate-wise ascent.

2. Holding real parameters (δ_i) fixed, maximize over complex uncertainties using power method.

3. Iterate.

The individual steps are as follows...

Lower Bounds: Real Parameters

Given M, intervals $[a_1, b_1], \ldots, [a_n, b_n]$, estimate lower bound for

$$\max_{a_i \leq \delta_i \leq b_i} \bar{\sigma} \left[M_{22} + M_{21} \mathcal{D}_\delta \left(I - M_{11} \mathcal{D}_\delta \right)^{-1} M_{12} \right]$$

Evaluating *anywhere* in the cube gives a lower bound on the maximum – try to improve this

- Start at center of $[a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n]$
- Iterate as follows:
  - Holding $\delta_2, \ldots, \delta_n$ fixed at their "current" values, adjust δ_1 in $[a_1, b_1]$ to maximize $\bar{\sigma}$.
  - Holding $\delta_1, \delta_2, \ldots, \delta_{i-1}, \delta_{i+1}, \ldots, \delta_n$ fixed at their "current" values, adjust δ_i in $[a_i, b_i]$ to maximize $\bar{\sigma}$.
  - and so on, cycling back and forth through the δ's

Issues:

- Order of cycling (in general) affects final convergence.
- Initial starting point need not be the center, and (in general) affects final convergence.
- No guarantee that iteration converges to maximum, but *it always does at least as good as taking the value at the center*.
- For a fixed M, as the width of the intervals go to zero, this "bounding" technique gets the right answer.
- How do we maximize over each individual δ_i?

Single Real Parameter: Maximization

With all but one of the δ's held fixed, the problem appears as (with different M, depending on values of parameters being held fixed)

Problem: Given $M \in \mathbf{C}^{(k+n_e)\times(k+n_d)}$. Solve

$$\max_{-1\leq\delta\leq 1} \bar{\sigma}\left[M_{22} + M_{21}\delta\left(I - \delta M_{11}\right)^{-1} M_{12}\right]$$

Mimicking Hamiltonian methods for state-space $\mathcal{H}_\infty$ norm ...

Lemma: Take $\gamma > \|M_{22}\|$. If there is a $\delta_0 \in [-1, 1]$ such that $F_u\left(M, \delta_0 I_k\right)$ has a singular value equal to γ, then the matrix H_γ

$$\begin{bmatrix} M_{11} & M_{12}M_{12}^* \\ 0 & M_{11}^* \end{bmatrix} + \begin{bmatrix} M_{12}M_{22}^* \\ M_{21}^* \end{bmatrix} \left(\gamma^2 I - M_{22}M_{22}^*\right)^{-1} \begin{bmatrix} M_{21} & M_{22}M_{12}^* \end{bmatrix}$$

has a real eigenvalue λ satisfying $|\lambda| \geq 1$ (specifically, $\delta_0 = \frac{1}{\lambda}$).

Single Real Parameter: Maximization (cont'd)

How:

$$F_u\left(M, \delta_0 I_k\right) \text{ has a singular value equal to } \gamma$$

$$\Updownarrow$$

$$\gamma^2 I - F_u(M, \delta_0 I_k)\left[F_u(M, \delta_0 I_k)\right]^* \text{ is singular}$$

$$\Updownarrow$$

$$K_\gamma(\delta) := \left[\gamma^2 I - F_u(M, \delta I_k)\left[F_u(M, \delta I_k)\right]^*\right]^{-1} \text{ has a pole at } \delta = \delta_0$$

Poles of $K_\gamma(\delta)$ are (subset of) reciprocals of eigenvalues of H_γ.

Remarks:

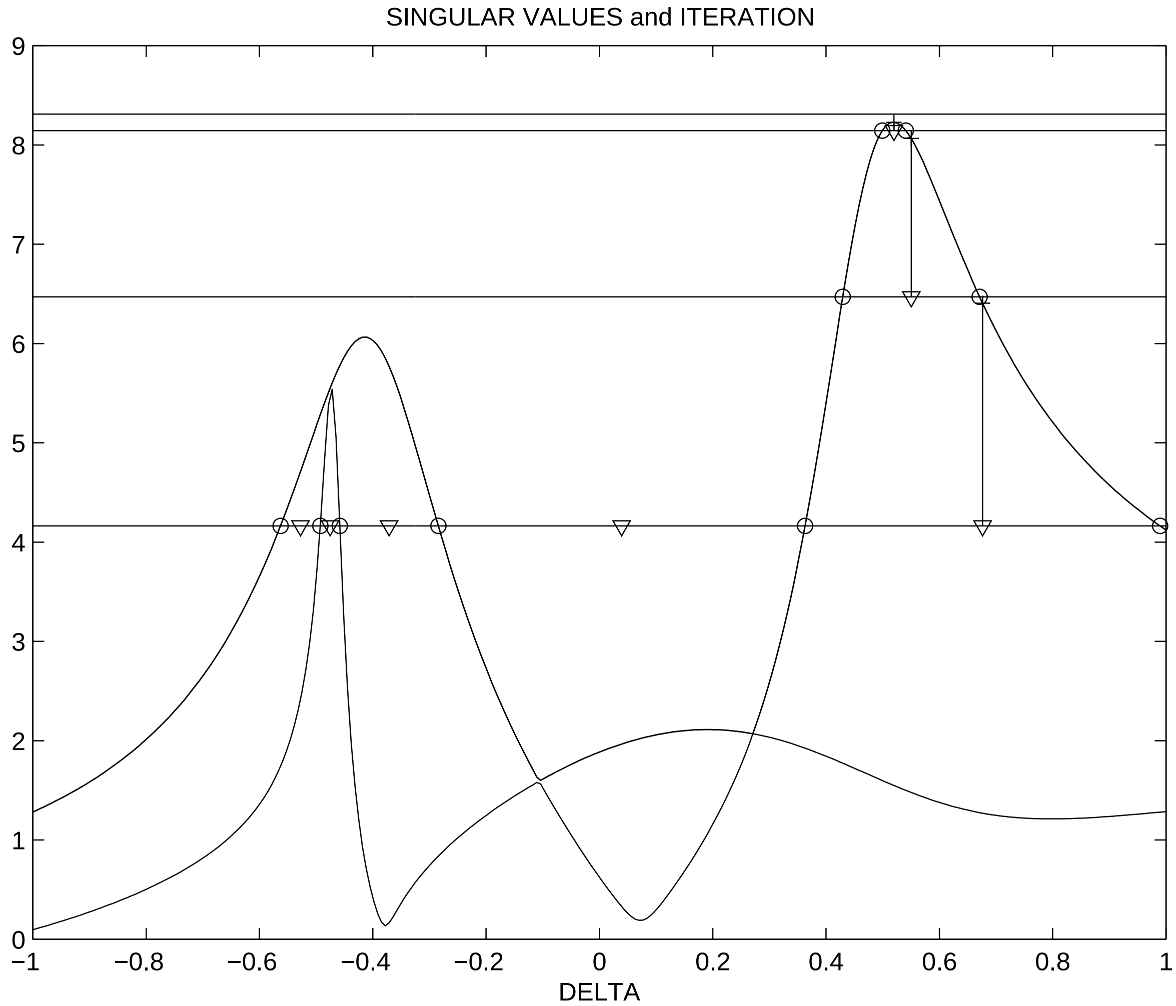
- The real eigenvalues of $2k \times 2k$ complex matrix H_γ give limited information about the sublevel sets of $f(\delta) := \bar{\sigma}\left[F_u(M, \delta I_k)\right]$.
- Iterative algorithm to bound maximum by repeatedly computing the eigenvalues of H_γ for increasing γ.
- Controllability/Observability assumptions on (M_{11}, M_{12}) and (M_{11}, M_{21}) render the theorem necessary and sufficient.

Single Real Parameter: Algorithm

Algorithm: choose relative stopping tolerance $\epsilon > 0$.

1. Set $\bar{\gamma} := \max \{\bar{\sigma}[F_u(M,-I)], \bar{\sigma}[F_u(M,0)], \bar{\sigma}[F_u(M,I)]\}$. Let $\bar{p}$ be the maximizer from $\{-1,0,1\}$.
2. Define $\gamma := (1+\epsilon)\bar{\gamma}$. Form H_γ and compute eigenvalues.
3. If there are no real eigenvalues with magnitude ≥ 1, STOP. Bounds are $\bar{\gamma} \leq \max_{-1\leq\delta\leq 1} \bar{\sigma}(\cdot) < (1+\epsilon)\bar{\gamma}$, with lower bound achieved by $\delta := \bar{p}$.
4. If there are any real eigenvalues, with magnitude $= 1$, denote their reciprocals as $\{r_i\}_{i=1}^N$.
5. Let $\{p_i\}_{i=1}^{N-1}$ denote the midpoints, $p_i := \frac{1}{2}(r_i + r_{i+1})$.
6. Redefine $\bar{p}$ to be the maximizer below

$$\bar{\gamma} := \max_{1\leq i\leq N-1} \bar{\sigma}\left[M_{22} + M_{21}p_i\left(I - p_i M_{11}\right)^{-1} M_{12}\right]$$



SINGULAR VALUES and ITERATION
9
8
7
6
5
4
3
2
1
0
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
DELTA

One Parameter Maximization: Scalar LFT

With all but one of the δ's held fixed, the problem appears as (with different M, depending on values of parameters being held fixed)

Problem: Given $M \in \mathbf{C}^{(k+1)\times(k+1)}$, and $a < b$, find

$$\max_{a \le \delta \le b} \left| F_u\left(M, \delta I_k\right) \right|$$

Dependence is rational, namely

$$F_u\left(M, \delta I_k\right) = m_{22} + m_{21}\delta\left(I - M_{11}\delta\right)^{-1} m_{12} = \frac{n(\delta)}{d(\delta)}$$

where

- n and d are k'th order polynomials
- coefficients of n and d are complex (since M is)
- n and d are easily computed from M

Write $n(\delta) = f(\delta) + jg(\delta)$ where f and g are real and imaginary parts. Similar for $d(\delta) = h(\delta) + jq(\delta)$. Note that

$$\left|\frac{n(\delta)}{d(\delta)}\right|^2 = \frac{f^2 + g^2}{h^2 + q^2}$$

is differentiable, and has critical points (slope equal 0) at same locations as $|F_u(M, \delta I_k)|$.

Task: Find $\delta \in [a, b]$ where either

$$\frac{d}{d\delta}\left(\left|\frac{n(\delta)}{d(\delta)}\right|^2\right) = 0, \quad \text{or} \quad d(\delta) = 0$$

These are precisely the roots of the polynomial

$$c := [ff' + gg']\left(h^2 + q^2\right) - [hh' + qq']\left(f^2 + g^2\right) = 0.$$

which is order $4k - 2$.

One Parameter Maximization: Scalar LFT (cont'd)

Procedure:

1. Compute n and d from M.
2. Form $c(\delta)$, and compute roots.
3. Evaluate $F_u(M, \delta I_k)$ at a, b and all real roots in interval (a, b).
4. Maximum value of $|F_u(M, \delta I_k)|$ occurs at one of these.

For **just** complex uncertainties,

$$\boldsymbol{\Delta} := \left\{ \operatorname{diag}\left[\Delta_1, \ldots, \Delta_f\right] : \Delta_i \in \mathbf{C}^{m_i \times m_i} \right\}$$

a power-method works well for worst-case gain. Assume $(I - M_{11}\Delta)$ is nonsingular for all $\Delta \in \mathbf{B}_{\boldsymbol{\Delta}}$. If

$$\max_{\Delta \in \mathbf{B}_{\boldsymbol{\Delta}}} \bar{\sigma} \left[M_{22} + M_{21}\Delta \left[I - M_{11}\Delta\right)^{-1} M_{12} \right] =: \gamma^2$$

(+ technical conditions) then there exist unit-vectors a, b, z and w

$$\begin{aligned}
a &= M_\gamma b \\
z_1 &= \frac{\|w_1\|}{\|a_1\|} a_1, \; \ldots \; z_f = \frac{\|w_f\|}{\|a_f\|} a_f, \quad z_{f+1} = \frac{\|w_{f+1}\|}{\|a_{f+1}\|} a_{f+1} \\
w &= M_\gamma^* z \\
b_1 &= \frac{\|a_1\|}{\|w_1\|} w_1, \; \ldots \; b_f = \frac{\|a_f\|}{\|w_f\|} w_f, \quad b_{f+1} = \frac{\|a_{f+1}\|}{\|w_{f+1}\|} w_{f+1}
\end{aligned}$$

where

$$M_\gamma := \left[\begin{array}{ccc|c} H_{1,1} & \cdots & H_{1,f} & \frac{1}{\gamma}H_{1,f+1} \\ \vdots & \ddots & \vdots & \vdots \\ H_{f,1} & \cdots & H_{f,f} & \frac{1}{\gamma}H_{f,f+1} \\ \hline \frac{1}{\gamma}H_{f+1,1} & \cdots & \frac{1}{\gamma}H_{f+1,f} & \frac{1}{\gamma^2}H_{f+1,f+1} \end{array}\right] =: \left[\begin{array}{c|c} M_{11} & \frac{1}{\gamma}M_{12} \\ \hline \frac{1}{\gamma}M_{11} & \frac{1}{\gamma^2}M_{22} \end{array}\right]$$

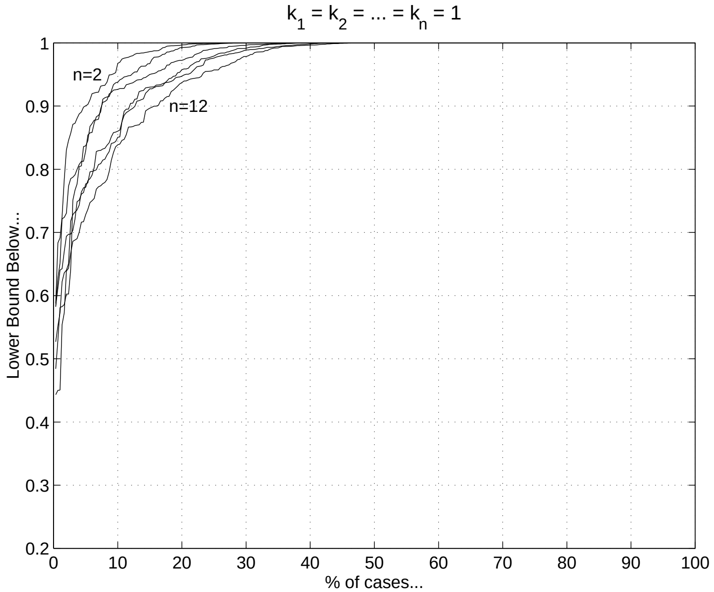
Try to find solutions by iterating, in the order written. Two facts about solutions (existence and what they mean)

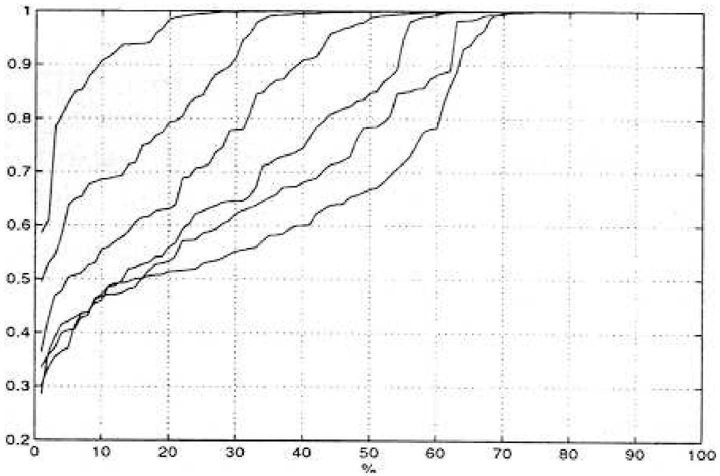
- Any solution (a, b, w, z, γ) yields a $\Delta \in \mathbf{B}_{\boldsymbol{\Delta}}$ achieving a gain of (at least) γ
- The maximum achievable gain is always a solution

Power Method: Comments

A few observations at this point

- Based on comparisons, the coordinate-wise approach to worst-case gain with real parameter uncertainties is favorable to the mixed (real/complex) power algorithm.





- Complex uncertainties can also be individually maximized over. Interestingly, this approach appears to be inferior to existing complex power methods.

- “Coordinate-wise” across the group of real parameters and the group of complex uncertanties is adequate, but could be improved...

Upper Bound: Unit Cube

Suppose real parameters only (for notation): associated with dimensions $k_1, k_2, \ldots, k_n$, define sets of matrices

$$\mathbf{D} := \left\{ \text{diag}\left[D_1, D_2, \ldots, D_n\right] : 0 < D_i = D_i^* \in \mathbf{C}^{k_i \times k_i} \right\}$$

and

$$\mathbf{G} := \left\{ \text{diag}\left[G_1, G_2, \ldots, G_n\right] : G_i = -G_i^* \in \mathbf{C}^{k_i \times k_i} \right\}$$

Applications of the $\mathcal{S}$-procedure (separating hyperplane) yields:

Theorem: If there is an $D \in \mathbf{D}, G \in \mathbf{G}$ and $\beta > 0$ such that

$$A(M, X, G, \beta^2) := \begin{bmatrix} D & 0 \\ 0 & \beta^2 I \end{bmatrix} - M^* \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} M + M^* \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} M \geq 0$$

then

$$\max_{-1 \leq \delta_i \leq 1} \bar{\sigma}\left[F_u\left(M, \mathcal{D}_\delta\right)\right] \leq \beta$$

Upper Bound: Unit Cube (cont'd)

Get best bound via minimization of $\gamma := \beta^2$, subject to the constraints

$$D \in \mathbf{D},\ G \in \mathbf{G},\ \gamma > 0,\ A(M, X, G, \gamma) \geq 0$$

Useful properties

- Linear objective, over the variable (D, G, γ)
- Convex constraints

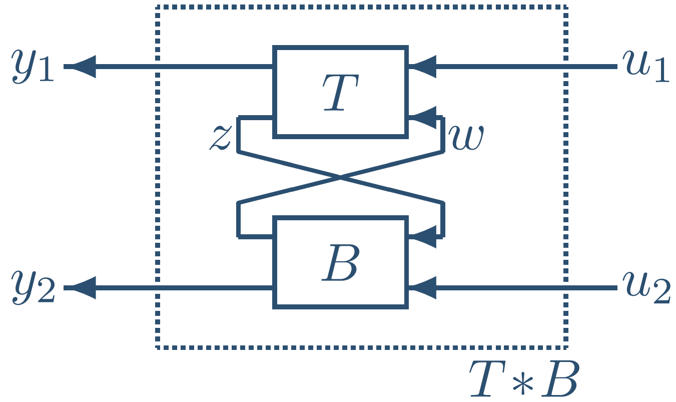
Question: When repeating the computation on a subdivided cube, can anything be re-used from the original cube's computation?

Redheffer Star Products: Definition

T and B are compatiably partitioned matrices, with $T_{22}B_{11}$ well-defined, and square. Consider the constraints

$$\begin{bmatrix} y_1 \\ z \end{bmatrix} = T \begin{bmatrix} u_1 \\ w \end{bmatrix}, \qquad \begin{bmatrix} w \\ y_2 \end{bmatrix} = B \begin{bmatrix} z \\ u_2 \end{bmatrix}$$

drawn as



Fact: For each u_1, u_2, there exist unique vectors z, w, y_1 and y_2 solving the constraints if and only if $\det(I - T_{22}B_{11}) \neq 0$.

In that case, the "star product $(T*B)$ is well-posed," and $T * B$ is defined as the 2×2 block matrix relating the u_i to the y_i.

Redheffer Star Products: A Small Gain Condition

Lemma: Suppose $T \in \mathbf{C}^{(n_1+n_1)\times(n_1+n_1)}$ and $M \in \mathbf{C}^{(n_1+n_2)\times(n_1+n_2)}$ are compatibly partitioned matrices, with $I - T_{22}M_{11}$ invertible. Assume T_{21} is invertible, and $g \in \mathbf{C}$, with $\mathrm{Re}(g) = 0$. If

$$M^*M + g\,(M - M^*) < I \quad \text{and} \quad T^*T + g\,(T - T^*) \leq I,$$

then

$$(T \star M)^* \,(T \star M) + g\left[(T \star M) - (T \star M)^*\right] < I$$

For $g = 0$, this is just: If

$$M^*M < I \text{ and } T^*T \leq I, \quad \text{equivalently } \bar{\sigma}(M) < 1 \text{ and } \sigma(T) \leq 1,$$

then

$$(T \star M)^* \,(T \star M) < I \qquad \text{equivalently } \bar{\sigma}\,(T \star M) < 1$$

which is an easy case.

Statement: Proof

Again: T_{21} is invertible, $T * M$ well-posed, g purely imaginary,

$$M^* M + g\,(M - M^*) < I \quad \text{and} \quad T^* T + g\,(T - T^*) \leq I.$$

Then

$$(T * M)^* \,(T * M) + g\left[(T * M) - (T * M)^*\right] < I$$

Proof: By assumption, $T * M$ is well-posed. Let $u_i \in \mathbf{C}^{n_i}$ be arbitrary, not both 0. Let y_i and z and w be the unique solutions to the defining star-product equations

$$\begin{bmatrix} y_1 \\ z \end{bmatrix} = T \begin{bmatrix} u_1 \\ w \end{bmatrix}, \qquad \begin{bmatrix} w \\ y_2 \end{bmatrix} = M \begin{bmatrix} z \\ u_2 \end{bmatrix}$$

Statement: Proof (cont'd)

Since T_{21} is invertible, it follows that u_2 and z are not both zero. The two hypothesis each combine with the star-product constraints to respectively give

$$\begin{array}{rcl} w^*w + y_2^*y_2 + g\left[(z^*w + u_2^*y_2) - (w^*z + y_2^*u_2)\right] & < & z^*z + u_2^*u_2 \\ y_1^*y_1 + z^*z + g\left[(u_1^*y_1 + w^*z) - (y_1^*u_1 + z^*w)\right] & \leq & u_1^*u_1 + w^*w \end{array}$$

Adding these, and cancelling leaves $y^*y + g\left(u^*y - y^*u\right) < u^*u$, which, since u was arbitrary, and $y = (T * M)u$, implies the desired conclusion. $\sharp$

Remarks: Suppose T satisfies $\|T\| \leq 1$, and $T = T^*$. It is easy to verify that for all imaginary g, T satisfies the hypothesis. Moreover, if T_{21} is not invertible, then both $<$ are changed to $\leq$

With Scalings

Suppose T and M given, with $I - T_{22}M_{11}$ invertible. T_{21} is invertible. Quantities $\beta > 0$, $D = D^* > 0$ and $G = -G^*$ are also given.
If

$$M^* \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} M + \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} M - M^* \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} < \begin{bmatrix} D & 0 \\ 0 & \beta^2 I \end{bmatrix}$$

and

$$T^* \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix} T + \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} T - T^* \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} \leq \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix}$$

then

$$(T*M)^* \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} (T*M) + \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} (T*M) - (T*M)^* \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} < \begin{bmatrix} D & 0 \\ 0 & \beta^2 I \end{bmatrix}$$

Proof: Extension of the previous result.

With Scalings (cont'd)

Remark: Suppose that T satisfies $\|T\| \leq 1$, $T = T^*$, and

$$GT_{ij} = T_{ij}G, \qquad D^{1/2}T_{ij} = T_{ij}D^{1/2}$$

for $1 \leq i \leq 2$, $1 \leq j \leq 2$. Then T satisfies

$$T^* \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix} T + \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} T - T^* \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} \leq \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix}$$

(the main hypothesis).

If T_{21} is not invertible, then both $<$ are changed to $\leq$

Star Products: Associative

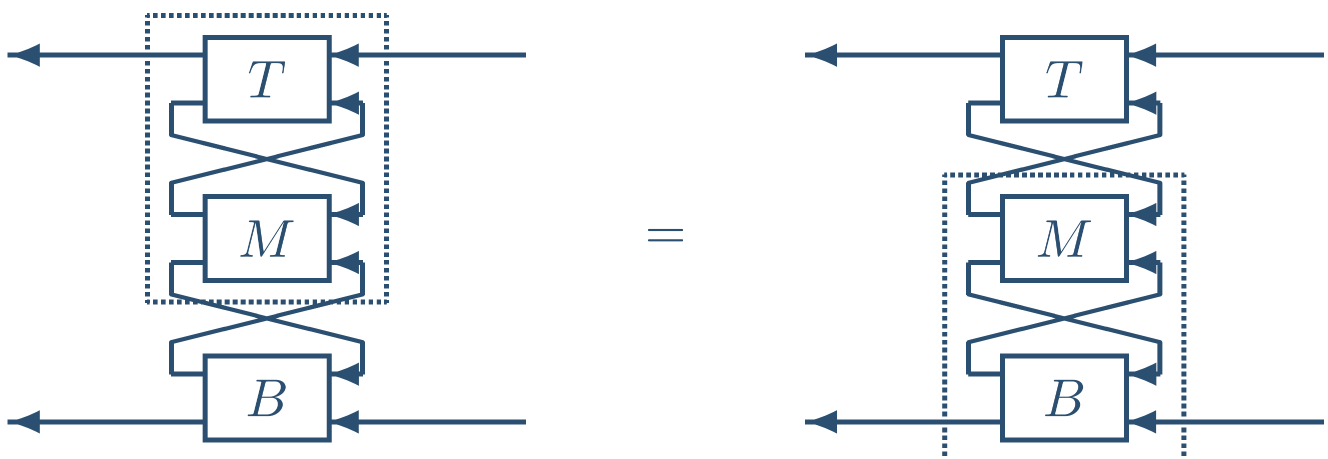
Well-posed star products are associative.

Suppose that T, M and B are compatibly partitioned matrices, Assume that $T * M$ is well-posed, and that $M * B$ is well-posed.

Then

$$(T * M) * B \text{ is well posed} \quad \Leftrightarrow \quad T * (M * B) \text{ is well posed}$$

Under these conditions,



$(T * M) * B$ $\qquad$ $T * (M * B)$

Upper Bound over Cube: Recenter/Normalize

For a given pair a and b, representing the cube

$$\mathbf{Q}_{[a,b]} := [a_1 \;\; b_1] \times [a_2 \;\; b_2] \times \cdots \times [a_n \;\; b_n]$$

define "center" and "radius" matrices

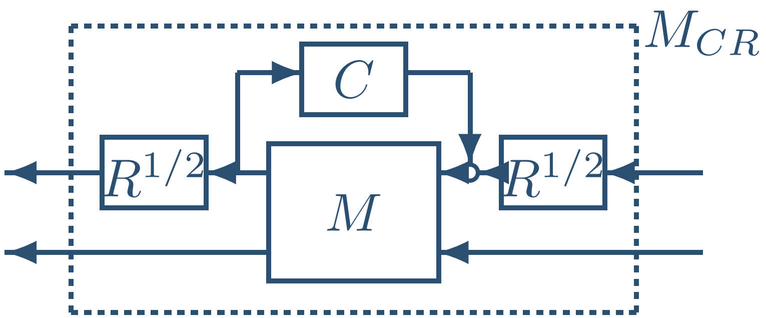
$$C_{[a,b]} := \begin{bmatrix} \frac{b_1+a_1}{2} I_{k_1} & 0 & \cdots & 0 \\ 0 & \frac{b_2+a_2}{2} I_{k_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{b_n+a_n}{2} I_{k_n} \end{bmatrix}$$

and

$$R_{[a,b]} := \begin{bmatrix} \frac{b_1-a_1}{2} I_{k_1} & 0 & \cdots & 0 \\ 0 & \frac{b_2-a_2}{2} I_{k_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{b_n-a_n}{2} I_{k_n} \end{bmatrix}$$

Recenter/Normalize (cont'd)

Define M_{CR} as below



Clearly

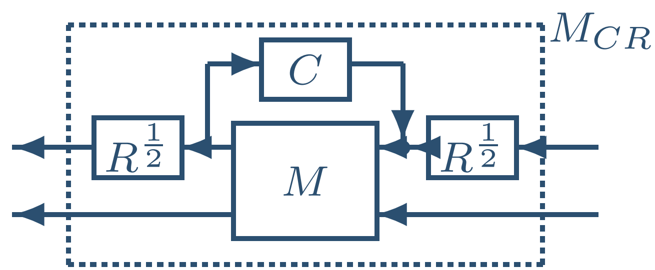
$$\max_{a_i \leq \delta_i \leq b_i} \bar{\sigma}\left[F_u\left(M, \mathcal{D}_\delta\right)\right] = \max_{-1 \leq \xi_i \leq 1} \bar{\sigma}\left[F_u\left(M_{CR}, \mathcal{D}_\xi\right)\right]$$

Recentering/Normalizing: consider the **unit cube** when useful

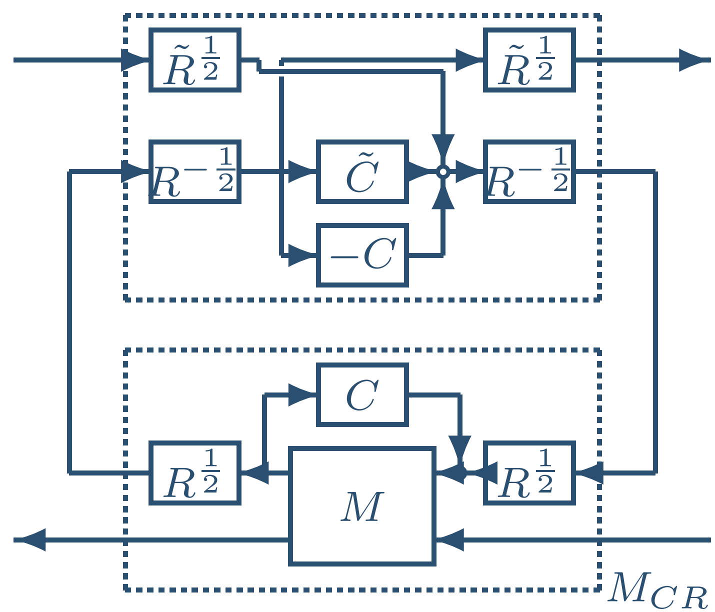
Transform Cube to Cube via Star Product

Suppose two cubes are given, by vectors a, b and $\tilde{a}, \tilde{b}$. Associated with each, define center and radius matrices, $C, \tilde{C}, R$ and $\tilde{R}$. How do we transform from $M_{CR} \rightarrow M_{\tilde{C}\tilde{R}}$?

Start with M_{CR}

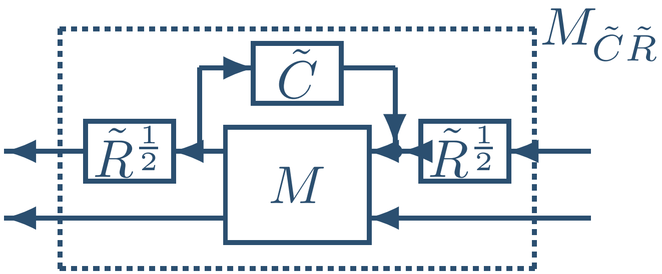


Cancel C and R, replacing with $\tilde{C}$ and $\tilde{R}$



Transform Cube to Cube via Star Product (cont'd)

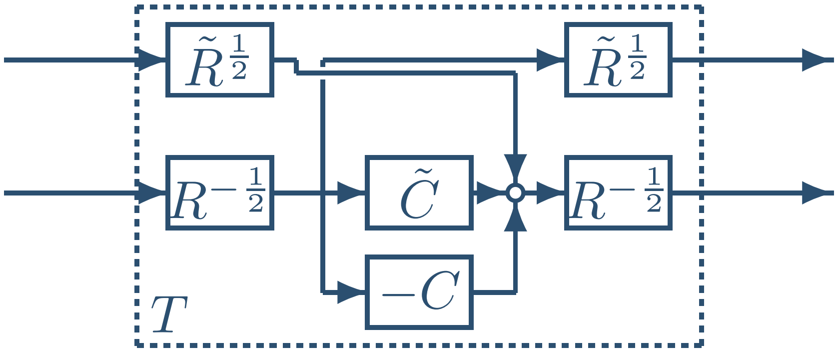
leaving $M_{\tilde{C}\tilde{R}}$



Define matrix T as

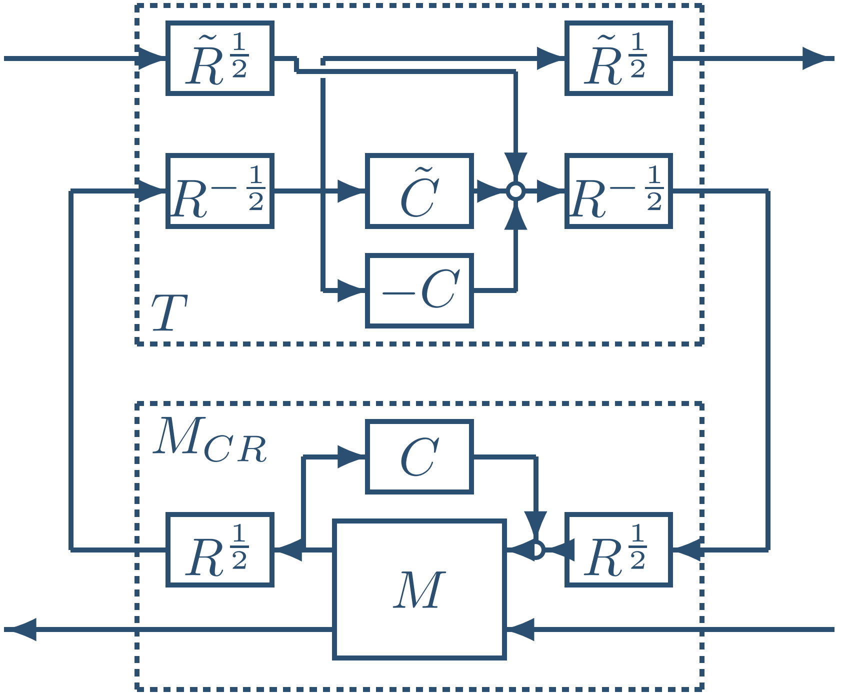
$$T := \begin{bmatrix} 0 & \tilde{R}^{\frac{1}{2}} R^{-\frac{1}{2}} \\ R^{-\frac{1}{2}} \tilde{R}^{\frac{1}{2}} & R^{-\frac{1}{2}} \left(\tilde{C} - C \right) R^{-\frac{1}{2}} \end{bmatrix}$$

Block diagram of T is

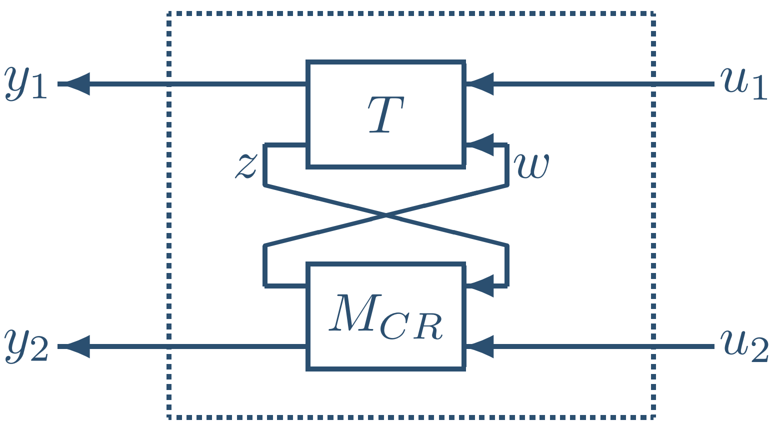


Transform Cube to Cube via Star Product (cont'd)

Through the star-product, T relates M_{CR} to $M_{\tilde{C}\tilde{R}}$



leaving $M_{\tilde{C}\tilde{R}} = T * M_{RC}$



Is $\mathbf{Q}_{[\tilde{a},\tilde{b}]} \subset \mathbf{Q}_{[a,b]}$?

The quantity $\|T\|$ determines whether the cube defined by $\left(\tilde{C}, \tilde{R}\right)$ is contained in the cube defined by (C, R).

The scalar version is:

Lemma: Given $c, \tilde{c} \in \mathbf{R}$, and $r > 0, \tilde{r} \geq 0$. Then

$$c - r \leq \tilde{c} - \tilde{r}, \quad \text{and} \quad \tilde{c} + \tilde{r} \leq c + r$$

if and only if

$$\bar{\sigma} \begin{bmatrix} 0 & \sqrt{\frac{\tilde{r}}{r}} \\ \sqrt{\frac{\tilde{r}}{r}} & \frac{\tilde{c}-c}{r} \end{bmatrix} \leq 1.$$

Is $\mathbf{Q}_{[\tilde{a},\tilde{b}]} \subset \mathbf{Q}_{[a,b]}$? (cont'd)

Hence, $\mathbf{Q}_{[\tilde{a},\tilde{b}]} \subset \mathbf{Q}_{[a,b]}$, if and only if the associated T

$$T = \begin{bmatrix} 0 & \tilde{R}^{\frac{1}{2}} R^{-\frac{1}{2}} \\ R^{-\frac{1}{2}} \tilde{R}^{\frac{1}{2}} & R^{-\frac{1}{2}} \left(\tilde{C} - C \right) R^{-\frac{1}{2}} \end{bmatrix}$$

satisfies $\bar{\sigma}(T) \leq 1$. In any case, $T = T^*$.

Moreover, the structure of $\mathbf{D}$ and $\mathbf{G}$ imply that for $i, j = 1, 2$

$$GT_{ij} = T_{ij}G, \qquad D^{1/2}T_{ij} = T_{ij}D^{1/2}$$

Combining all of these ideas yields the desired result.

Upper-bound Feasability: Sub-divided Cubes

Theorem: Given M and two cubes $\mathbf{Q}_{[a,b]}$ and $\mathbf{Q}_{[\tilde{a},\tilde{b}]}$. Let R, C, $\tilde{R}$ and $\tilde{C}$ be the associated "radius/center" matrices. Assume that $I - M_{11}C$ invertible. If there exist $D \in \mathbf{D}$, $G \in \mathbf{G}$ and $\beta > 0$ such that

$$M_{RC}^* \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} M_{RC} + \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} M_{RC} - M_{RC}^* \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} < \begin{bmatrix} D & 0 \\ 0 & \beta^2 I \end{bmatrix}$$

and $\mathbf{Q}_{[\tilde{a},\tilde{b}]} \subset \mathbf{Q}_{[a,b]}$, then also

$$M_{\tilde{R}\tilde{C}}^* \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} M_{\tilde{R}\tilde{C}} + \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} M_{\tilde{R}\tilde{C}} - M_{\tilde{R}\tilde{C}}^* \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} < \begin{bmatrix} D & 0 \\ 0 & \beta^2 I \end{bmatrix}$$

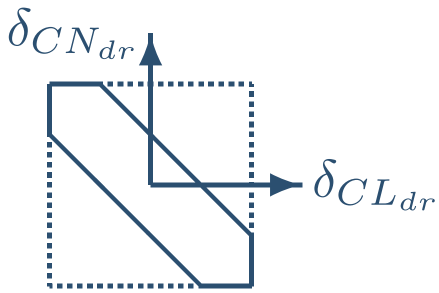
Implication: When subdividing a cube in Divide-and-Conquer scheme,

- the decision variables (D, G, β) obtained in the previous optimization are feasable for subdivided cube.
- hence, the optimization need not first obtain feasability.

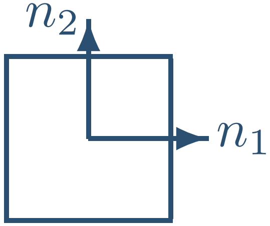
This seems to save about 25-30% in total upper bound computation

Correlated Parameters

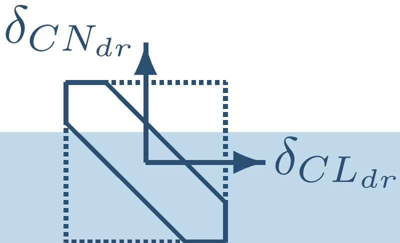
- Bounds are valid for cube aligned with $(\delta_1, \delta_2, ..., \delta_n)$ axis
- In X-38 model, two aerodynamic coefficients, CL_{dr} and CN_{dr} are correlated, and their uncertainties, $\delta_{CL_{dr}}$ and $\delta_{CN_{dr}}$ are correlated, assumed to lie in region below



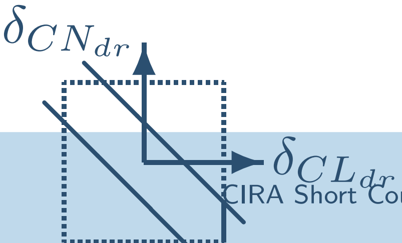
- Create LFT functions that approximately map a unit cube into such a region, for example



into



or



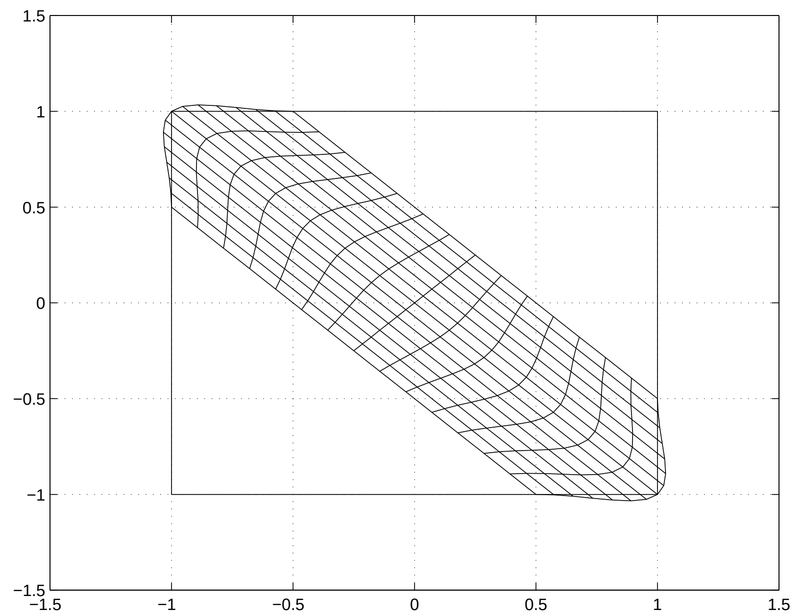
Worst-Case Performance
Setup
Justification
Performance Objective
Uncertainty Model
Real Parameters
Lower Bounds
Algorithm
Scalar LFT
Power Method
Upper Bound
Small Gain
Normalize
Feasability
Correlated Parameters
Curves
wcgain
wcmargin
wcsens
wcgopt
Bibilography
GTM Analysis
GTM Simulation
Uncertainty
Linearization
Flight Data 1
Flight Data 2
Worst-Case Response
Worst-case Simulation

Correlated Parameters (cont'd)

Some formulae which approximately map the cube (n_1, n_2) to (δ_1, δ_2)

$$\begin{aligned}
\delta_{CL_{dr}} &= -\frac{n_1}{\sqrt{2}}\left(\sqrt{2} - 2\beta\frac{n_2^2}{1+n_2^2}\right) + \frac{\beta n_2}{\sqrt{2}} \\
\delta_{CN_{dr}} &= \frac{n_1}{\sqrt{2}}\left(\sqrt{2} - 2\beta\frac{n_2^2}{1+n_2^2}\right) + \frac{\beta n_2}{\sqrt{2}}
\end{aligned}$$

where n_1 and n_2 each independently range from [-1,1]. An example, with $\beta = \frac{1}{2\sqrt{2}}$ is shown below.



Correlated Parameters (cont'd)

How is this represented as an LFT? Define 5×5 matrices S_l, S_n

$$S_l := \begin{bmatrix} 0 & -\beta\sqrt{2} & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{\beta}{\sqrt{2}} \\ -1 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad S_n := \begin{bmatrix} 0 & -\beta\sqrt{2} & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{\beta}{\sqrt{2}} \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Then for any n_1, n_2, letting N denote

$$N := \begin{bmatrix} n_1 & 0 & 0 & 0 \\ 0 & n_2 & 0 & 0 \\ 0 & 0 & n_2 & 0 \\ 0 & 0 & 0 & n_2 \end{bmatrix}$$

gives

$$F_u(S_l, N) = -\frac{n_1}{\sqrt{2}}\left(\sqrt{2} - 2\beta\frac{n_2^2}{1+n_2^2}\right) + \frac{\beta n_2}{\sqrt{2}}$$

and

$$F_u(S_n, N) = \frac{n_1}{\sqrt{2}}\left(\sqrt{2} - 2\beta\frac{n_2^2}{1+n_2^2}\right) + \frac{\beta n_2}{\sqrt{2}}$$

Correlated Parameters (cont'd)

So: replace every instance of $\delta_{CL_{dr}}$ with $F_u(S_l, N)$, and every instance of $\delta_{CN_{dr}}$ with $F_u(S_n, N)$

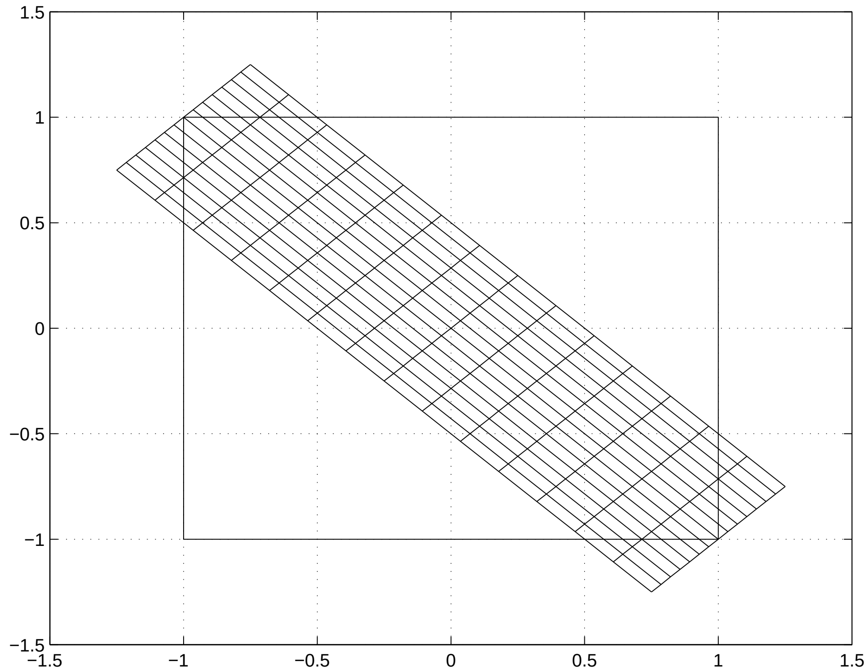
Observation: Nice approximation to region, but LFT representation involves 6 copies of n_2 for every copy of $\delta_{CL_{dr}}$ and $\delta_{CN_{dr}}$ in the original problem.

Correlated Parameters (cont'd)

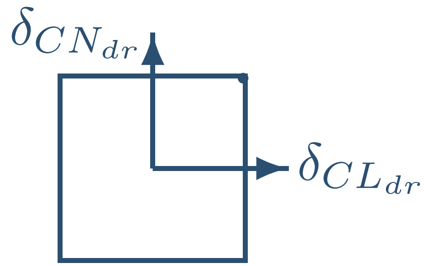
Simpler, though cruder approximation to map the cube (n_1, n_2) to desired region is

$$\begin{aligned} \delta_{CL_{dr}} &= -n_1 - \tfrac{1}{4}n_2 \\ \delta_{CN_{dr}} &= n_1 - \tfrac{1}{4}n_2 \end{aligned}$$

where n_1 and n_2 each independently range from [-1,1]. This is shown below.

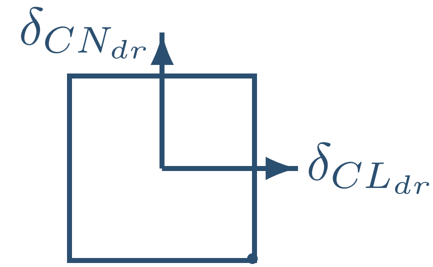


No Correlation Constraint



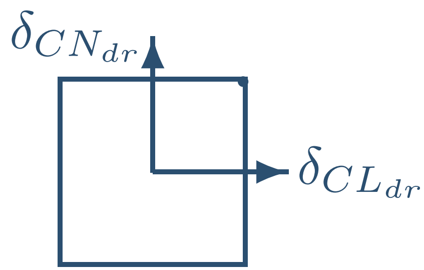
`m44h26`

$2.88 \leq \mathrm{WCP} \leq 2.88$



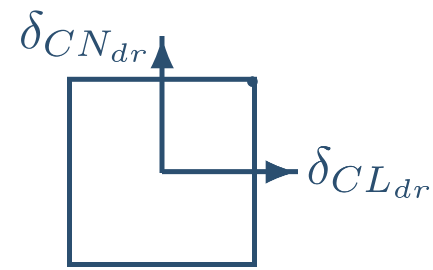
`jscm44h25t10`

$3.84 \leq \mathrm{WCP} \leq 3.86$



`jscm7h42t10`

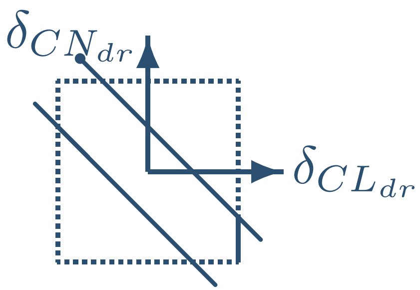
$2.44 \leq \mathrm{WCP} \leq 2.44$



`m7h42t10`

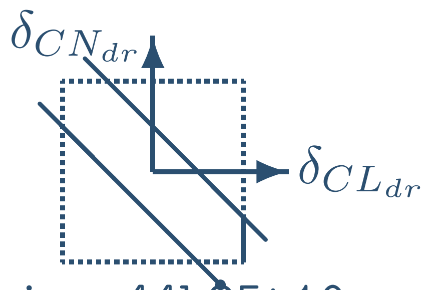
$4.26 \leq \mathrm{WCP} \leq 4.32$

Rotated Linear Fit



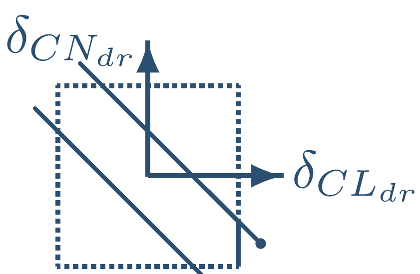
m44h26

$2.06 \leq \text{WCP} \leq 2.09$



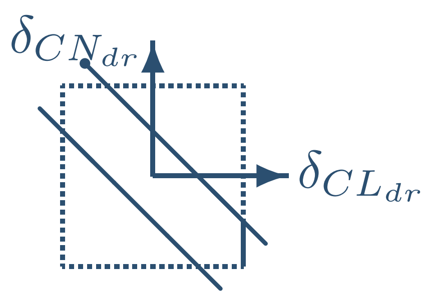
jscm44h25t10

$3.98 \leq \text{WCP} \leq 3.98$



jscm7h42t10

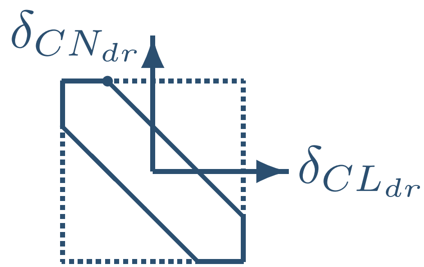
$1.91 \leq \text{WCP} \leq 1.91$



m7h42t10

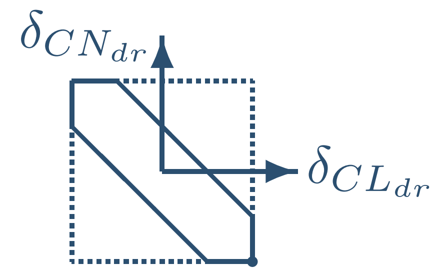
$2.10 \leq \text{WCP} \leq 2.11$

Rational Fit



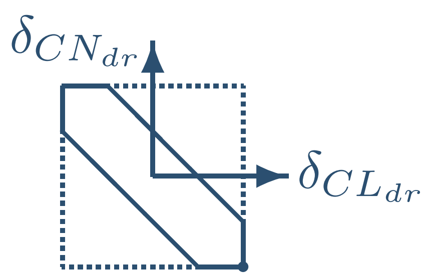
`m44h26`

$$2.03 \leq \text{WCP} \leq 2.04$$



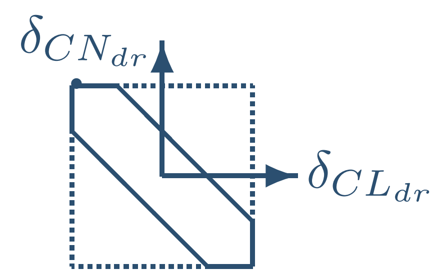
`jscm44h25t10`

$$3.84 \leq \text{WCP} \leq 3.84$$



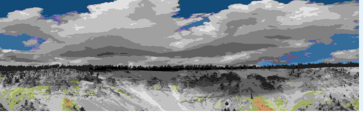
`jscm7h42t10`

$$1.82 \leq \text{WCP} \leq 1.82$$



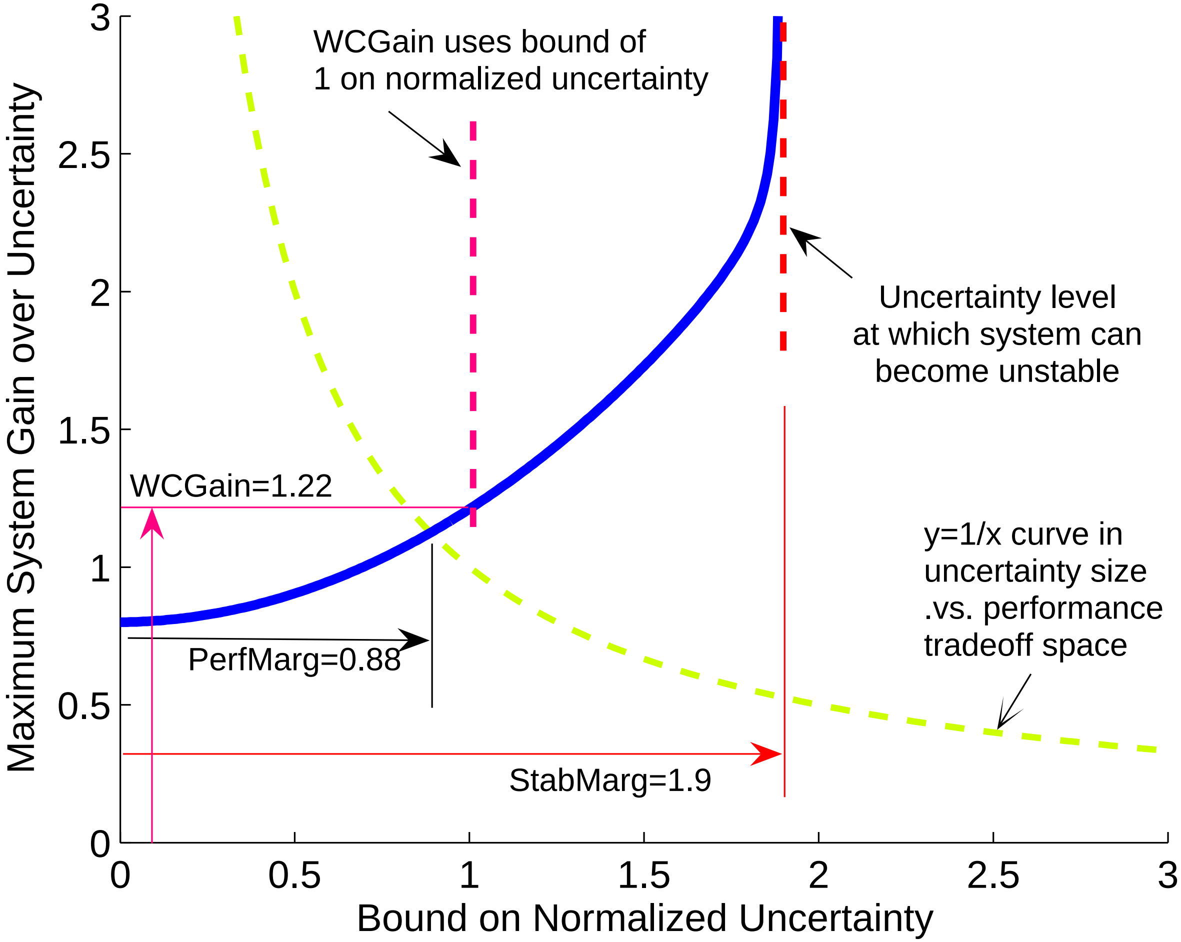
`m7h42t10`

$$1.80 \leq \text{WCP} \leq 1.82$$



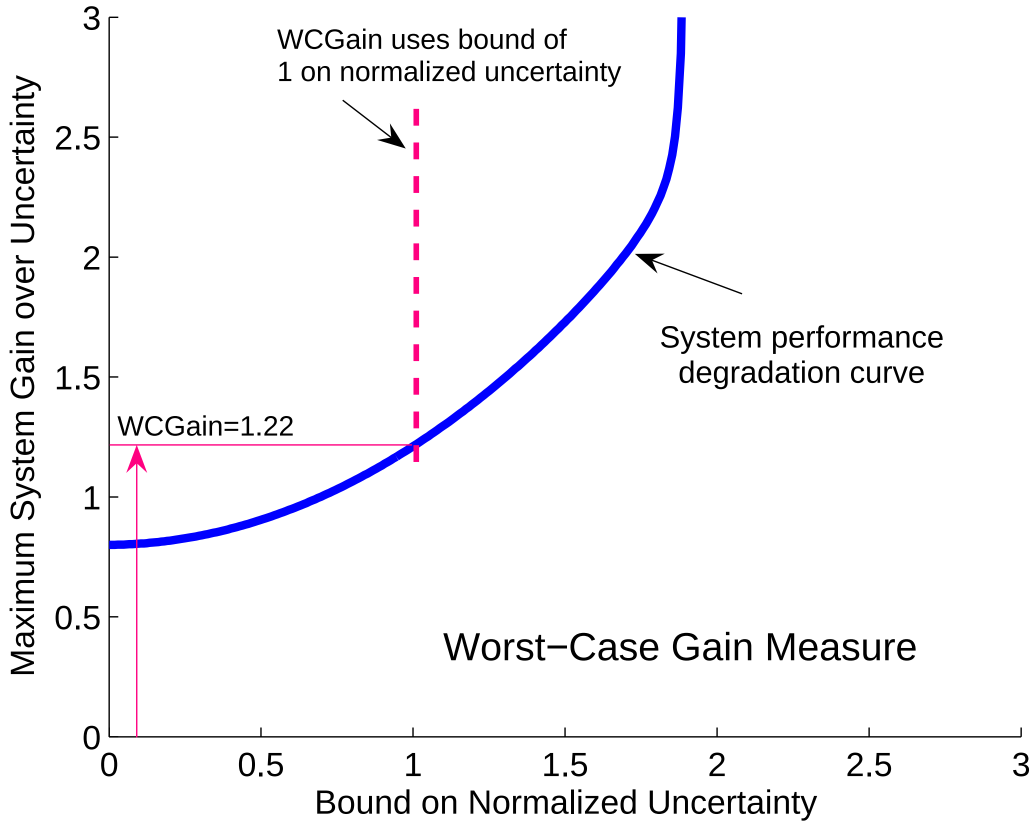
Worst-Case Performance Margin

Generally, "robustness computations" refer to determining specific attributes of the system performance degradation curve. The commands `robuststab`, `robustperf` and `wcgain` all compute single scalar attributes of the system performance degradation curve.



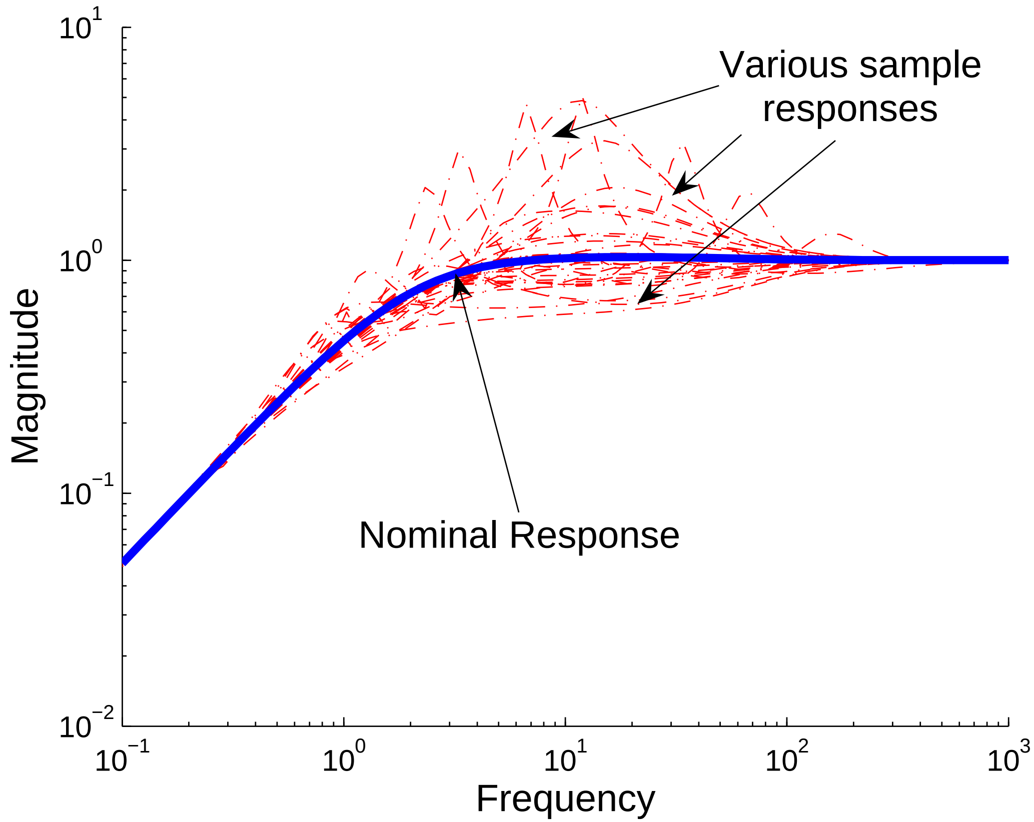
Worst-Case Gain: `wcgain`

The worst-case gain, `wcgain`, measure is the maximum achievable system gain over all uncertain elements whose normalized size is bounded by 1.



Worst-Case Gain: `wcgain` (cont'd)

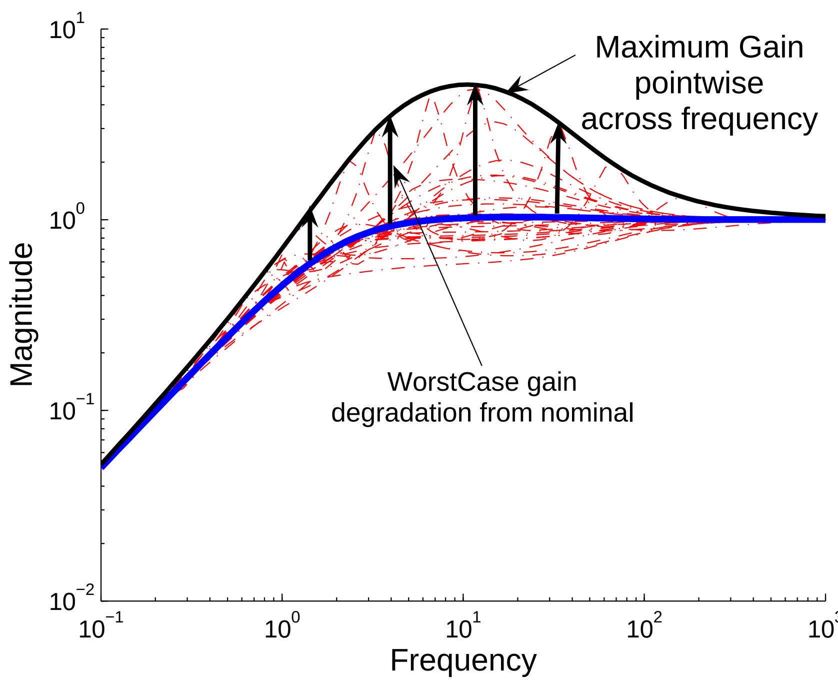
Determining the maximum gain over all allowable values of the uncertain elements is referred to as a *worst-case gain* analysis. "Gain" refers to the frequency response magnitude.



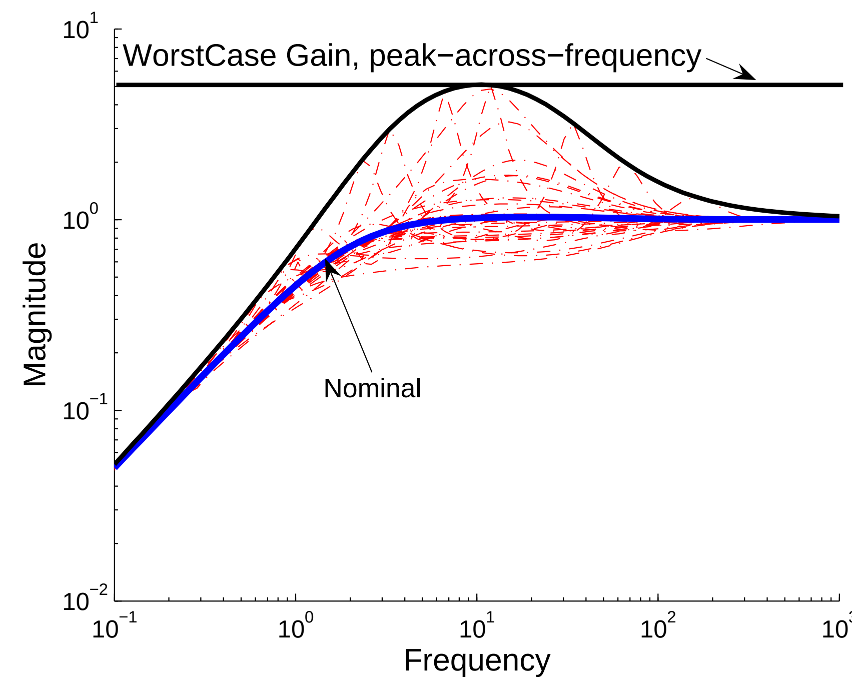
Worst-Case Gain: `wcgain` (cont'd)

`WCGAIN` can perform two types of analysis on uncertain systems:

pointwise-in-frequency worst-gain analysis yields the frequency-dependent curve of maximum gain, corresponds to maximum value at each and every frequency.



peak-over-frequency worst-gain analys (default) computes the largest value o frequency-response magnitude across a frequencies.



Worst-Case Gain: `wcgain` (cont'd)

```
[MAXGAIN,MAXGAINUNC,INFO] = WCGAIN(SYS)
```

- `MAXGAIN` contains
  - `UpperBound` peak (across frequency) upper bound on worst-case gain,
  - `LowerBound` peak lower bound on worst-case gain.
  - `CriticalFrequency` frequency at which the maximum gain occurs.
- `MAXGAINUNC` worst-case uncertainty values (of size `UpperBound`).
- `INFO` Structure with the following fields:

`Sensitivity`: Structure of percentages corresponding to `MAXGAIN` sensitivity to variations in each uncertainty level.

`Frequency`: Frequency vector used in analysis.

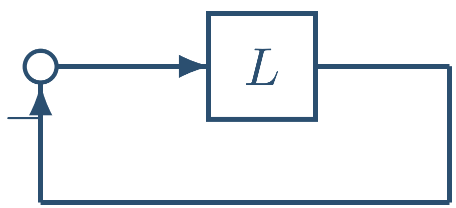
`ArrayIndex`: Array index (for arrays of uncertain systems) where the maximum gain occurs. Value is scalar,

Worst-Case Margin: `wcmargin`

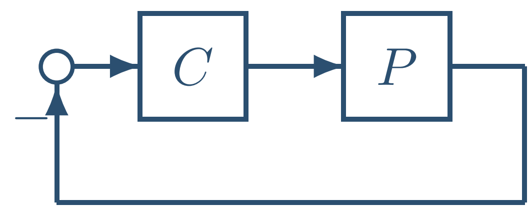
Worst-case margin, `wcmargin`, calculates the largest disk margin such that for values of the uncertainty and all gain and phase variations inside the disk, the closed-loop system is stable.

```
[wcmargi,wcmargo] = wcmargin(L), (wcmargin(P,C))
```

Worst-case input and output loop-at-a-time gain/phase margins of the feedback loop consisting of loop transfer function `L` with negative feedback (`C` in negative feedback with `P`).



1-dof architecture



1-dof architecture

`wcmargi` and `wcmargo` are structures corresponding to the loop-at-a-time worst-case, single-loop gain and phase margin of the channel. `wcmargi` and `wcmargo` contain the fields:

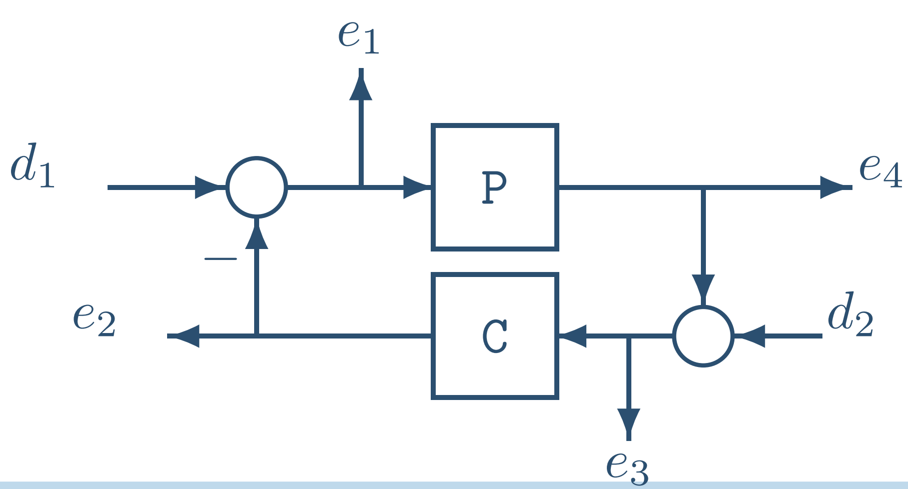
- `GainMargin`, `PhaseMargin`, `Frequency`, `MarginSens`

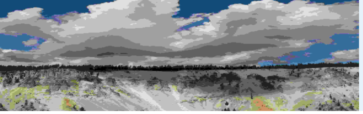
Worst-Case Sensitivity: `wcsens`

Worst-Case Performance
Setup
Justification
Performance Objective
Uncertainty Model
Real Parameters
Lower Bounds
Algorithm
Scalar LFT
Power Method
Upper Bound
Small Gain
Normalize
Feasability
Correlated Parameters
Curves
wcgain
wcmargin
wcsens
wcgopt
Bibilography
GTM Analysis
GTM Simulation
Uncertainty
Linearization
Flight Data 1
Flight Data 2
Worst-Case Response
Worst-case Simulation

`wcsens` calculates the worst-case sensitivity and complementary sensitivity functions of a plant-controller feedback loop.

```
wcst = wcsens(L,type,scaling,opt)
wcst = wcsens(P,C,type,scaling,opt)
```





Worst-Case Sensitivity: `wcsens`

```
wcst = wcsens(P,C,type,scaling,opt)
```

Each sensitivity substructures, `WCST.Si`, `WCST.Ti`, etc, contains five fields:

`MaximumGain`	Lower bound, LowerBound, and upper bound, Up-. perBound, on the sensitivity function, maximum gain at, CriticalFrequency
`BadUncertainValues`	Structure of uncertain elements value which maximize the System gain.
`System`	Sensitivity transfer matrix.
`BadSystem`	Worst-case sensitivity transfer matrix (FRD object).
`Sensitivity`	Each entry indicates local sensitivity of `MaximumGain`.

Worst-Case Gain Option: `wcgopt`

Used by `wcgain`, `wcmargin`, `wcsens`, and `wcnorm`

```
options = wcgopt('name1',value1,'name2',value2,...)
```

Object Property	Description
Sensitivity	Margin sensitivity to individual uncertainties. Default 'on'.
LowerBoundOnly	If LowerBoundOnly is 'on', only lower bound computed. Default 'off'
FreqPtWise	Set to 1 applies stopping criteria at every frequency point. Default = 0.
ArrayDimPtWise	For indices specified for stopping
Default	Default values of all `wcgopt` properties
Meaning	Description of all `wcgopt` properties
VaryUncertainty	% uncertainty variation for sensitivity calculations. Default is 25.
AbsTol	Upper and Lower Absolute Stopping Tolerance. Default = 0.02.
RelTol	Upper/Lower Relative Stopping Tolerance. Default = 0.05.
MGoodThreshold	Multiplicative (UpperBound) Stopping Threshold, Default = 1.04.
AGoodThreshold	Additive (UpperBound) Stopping Threshold, Default = 0.05.
MBadThreshold	Multiplicative (LowerBound) Stopping Threshold, Default = 5.
ABadThreshold	Additive (LowerBound) Stopping Threshold, Default = 20.
NTimes	Number LowerBound Restarts. Default = 2.
MaxCnt	Number of LowerBound cycles. Default = 3.
MaxTime	Maximum computation time (secs). Default = 720.

Bibilography

1. Boyd, Balakrishnan and Kabamba, "A bisection method for computing the $\mathbf{H}_\infty$ norm of a transfer matrix and related problems," *Math Control Signals and Systems*, 2(3):207-219, 1989.
2. Bruinsma and Steinbuch, "A fast algorithm to compute the H_∞ norm of a transfer function matrix," *Systems and Control Letters*, 14, pp. 287-293, 1990.
3. R. de Gaston and M. Safonov, "Exact calculation of the multiloop stability margin," *IEEE Trans. Auto. Control*, vol. 33, pp. 156-171, 1988.
4. J.C. Doyle, "Analysis of feedback systems with structured uncertainties", *IEE Proceedings*, **129**, Part D, pp. 242–250, (1982).
5. M. Fan, A. Tits, and J. Doyle, "Robustness in the presence of joint parametric uncertainty and unmodeled dynamics," *IEEE Trans. Auto. Control*, vol. 36, Jan. 1991, pp. 25-38.
6. S. Glavaski and J. Tierno, "Advances in worst-case $\mathcal{H}_\infty$ performance computation," *Proceedings of the 1998 IEEE International Conference on Control Applications*, Trieste, Italy, 1-4 Sept. 1998, IEEE, 1998. pp. 668–673
7. M. Newlin and P. Young, "Mixed mu problems and branch and bound techniques" *Int. J. Robust Nonl. Control*, **7**, pp. 145–164., (1997).
8. A. Packard and J. Doyle, "The Complex Structured Singular Value," *Automatica*, **29**, pp. 71–109, (1993).

Bibilography (cont'd)

9. R. Redheffer, "On a certain linear fractional transformation," *J. Math. Phys.*, **39**. pp. 269–286, (1960).
10. M.G. Safonov, "Tight bounds on the response of multivariable systems with component uncertainty," *1978 Allerton Conference on Communication, Control and Computing*, pp. 451-460.
11. M.G. Safonov, "Stability margins of diagonally perturbed multivariable feedback systems," *Proceedings of the IEE*, **129**, Part D, pp. 251–256, (1982).
12. A. Sideris and R. Sanchez Peña, "Robustness margin calculations with dynamic and real parametric uncertainty," *IEEE Trans. Auto. Control*, vol. 35, no. 8, pp. 970-974, August 1990.
13. A. Sideris, and R. S. Sánchez Peña, "Fast computation of the multivariable stability margin for real interrelated uncertain parameters," *IEEE Trans. Auto. Control*, vol. 34, no. 12, pp. 1272-1276, December, 1989.
14. J. Tierno and J. Doyle, "Control problems and the polynomial time hierarchy," *Proceedings 34th IEEE Conference on Decision and Control*, New Orleans, 13-15 Dec. 1995, pp. 2927-31.

Bibilography (cont'd)

15. Young, P.M.; Newlin, M.P.; Doyle, J.C. "Computing bounds for the mixed mu problem," *International Journal of Robust and Nonlinear Control*, Oct. 1995, vol.5,

16. Young, P.M.; Doyle, J.C. "Properties of the mixed mu problem and its bounds," *IEEE Transactions on Automatic Control*, Jan. 1996, vol.41, (no.1):155-9.

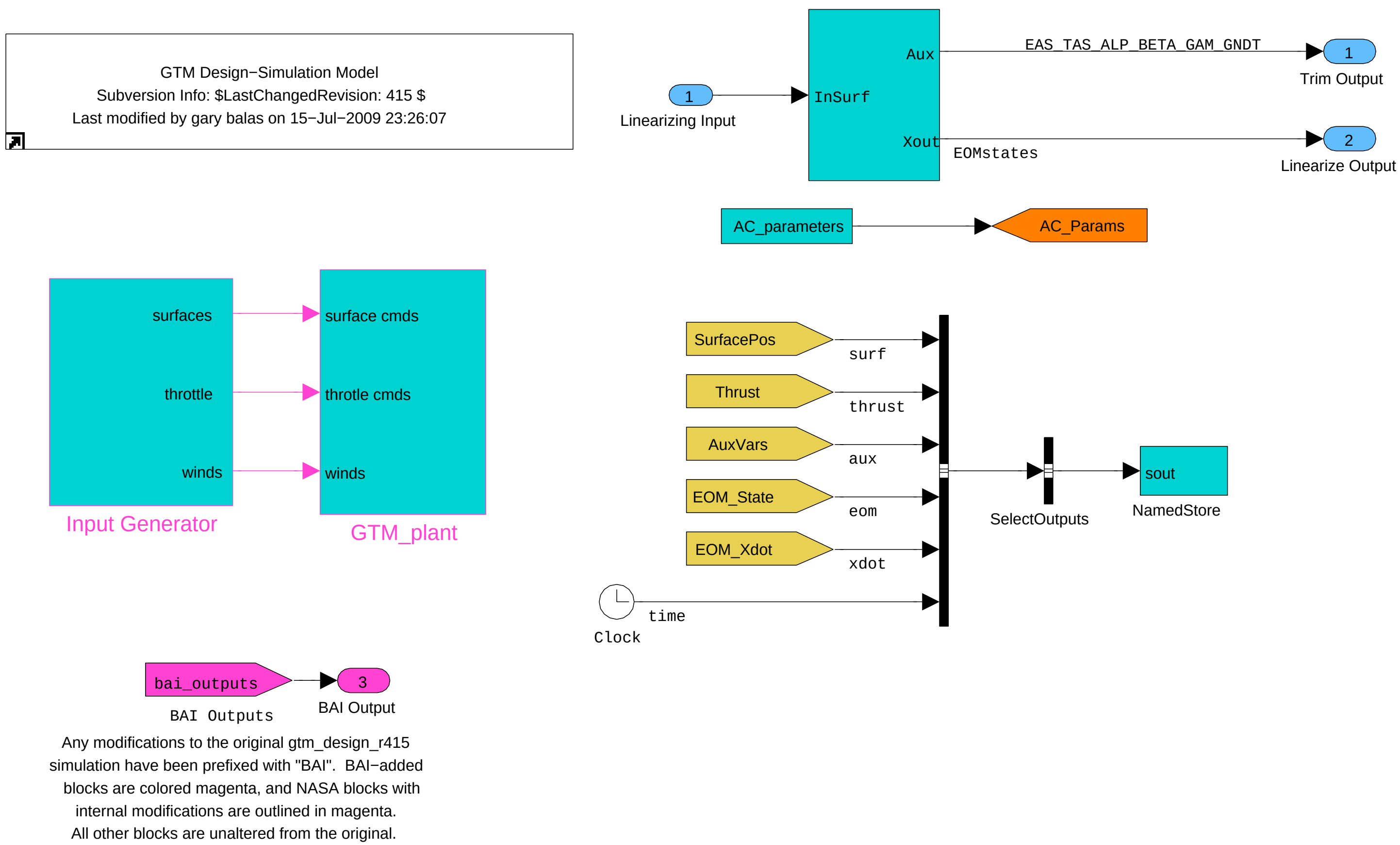
17. Young, P.M.; Doyle, J.C. "A lower bound for the mixed mu problem," *IEEE Transactions on Automatic Control*, Jan. 1997, vol.42, (no.1):123-8.

NASA Generic Transport Model (GTM) Aircraft

MUSYN developed tools and performed analysis as part of Barron Associates Inc., Alec Bateman PI, ROME Phase II SBIR contract.

- 5.5% dynamically-scaled, remotely piloted, twin-turbine swept wing aircraft, NASA Langley Research Center.
- Simulation modified to include 18 parametric aerodynamic coefficients uncertainties, 8 dynamic actuator uncertainties.

- Aerodynamic coefficient and unmodeled actuator dynamic uncertainties in *GTM_plant* block.

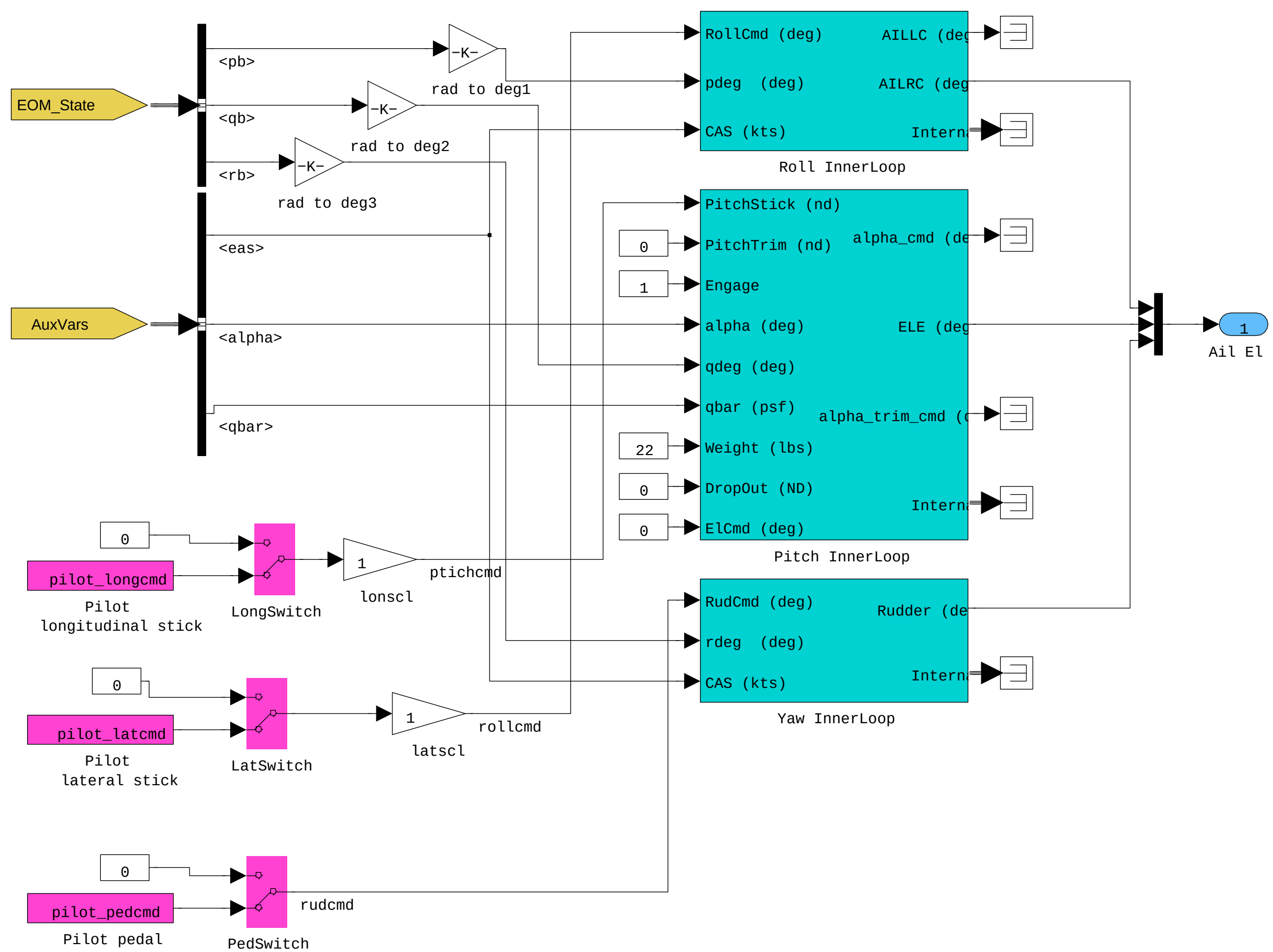


NASA Baseline GTM Controller

Yaw and roll axes controller scheduled with air speed: KCAS 1 at 50 knots, 0 at 100 knots

- 95 knots, 1320 ft altitude: roll and yaw controllers effectively

The cyan blocks were cut and paste from NASA's model (BaselineFCL−S2.mdl)
that Austin Murch emailed March 19, 2009 to Bateman and Lichter.

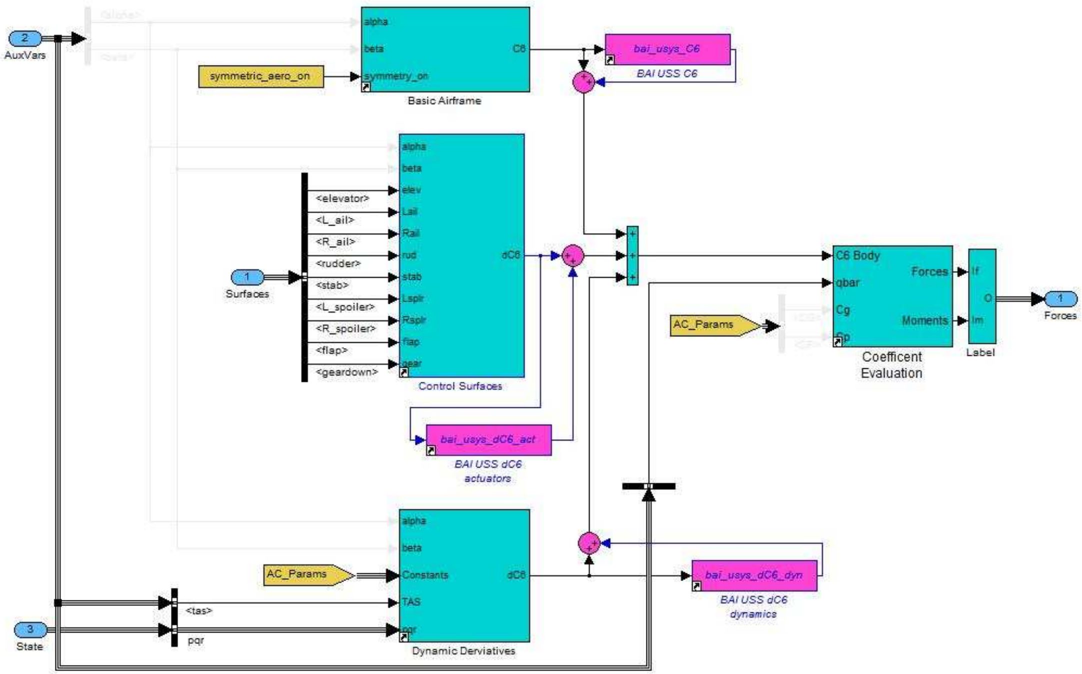


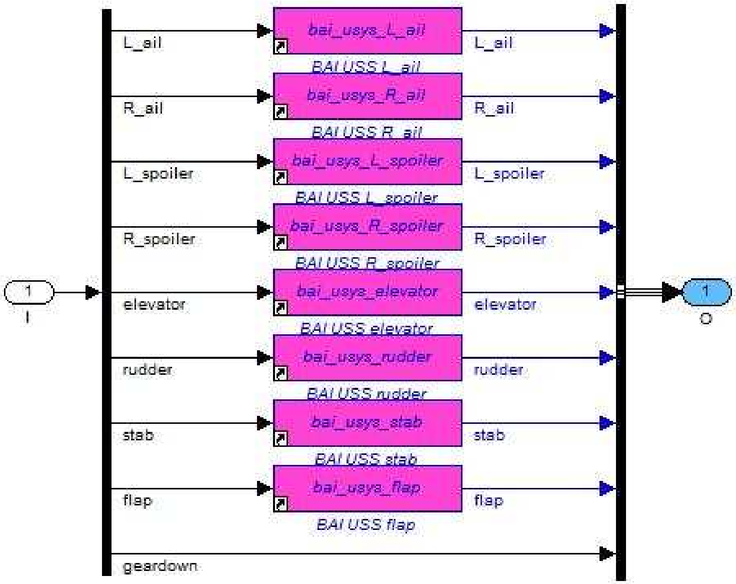
Aerodynamic and Unmodeled Dynamic Uncertainty

Parametric uncertainty in six body-axis aerodynamic coefficients: C_u, C_v, C_w, C_p, C_q, and C_r. Multiplicative input uncertainty is used to model actuator uncertain dynamics.

```
% Body aero coefficients, increments from nominal (+/- 20%)
gainCu = ureal('gain_Cu',0,'PlusMinus',0.2);
gainCv = ureal('gain_Cv',0,'PlusMinus',0.2);
gainCw = ureal('gain_Cw',0,'PlusMinus',0.2);
gainCp = ureal('gain_Cp',0,'PlusMinus',0.2);
gainCq = ureal('gain_Cq',0,'PlusMinus',0.2);
gainCr = ureal('gain_Cr',0,'PlusMinus',0.2);
bai_usys_C6 = diag([gainCu, gainCv, gainCw, gainCp, gainCq, gainCr]);
W_rudder = 1;
bai_usys_rudder = 1+W_rudder*ultidyn('gain_rudder',[1 1],'Bound',0.2);
```

Aerodynamic and Unmodeled Dynamic Uncertainty



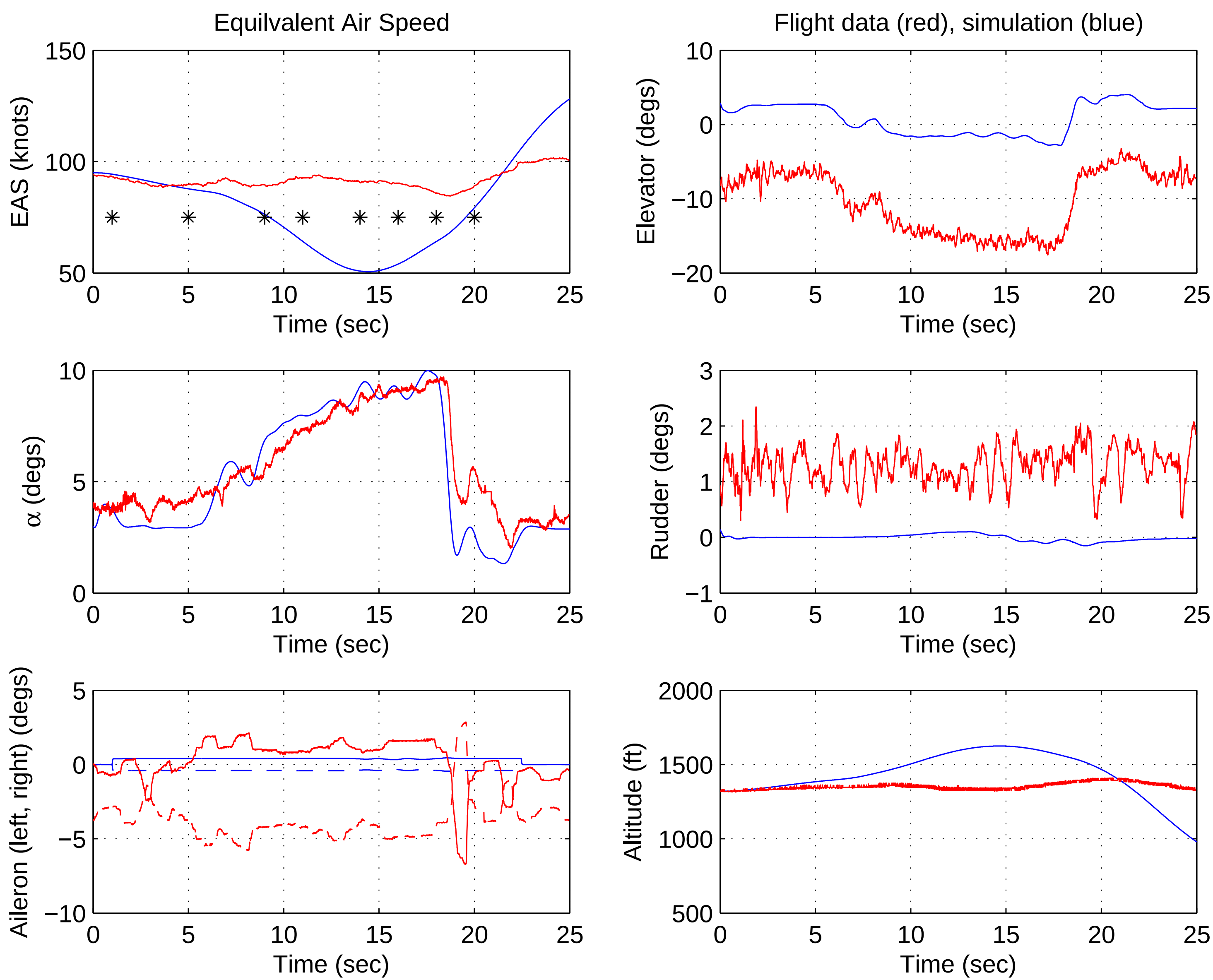


Linearization Along a Trajectory

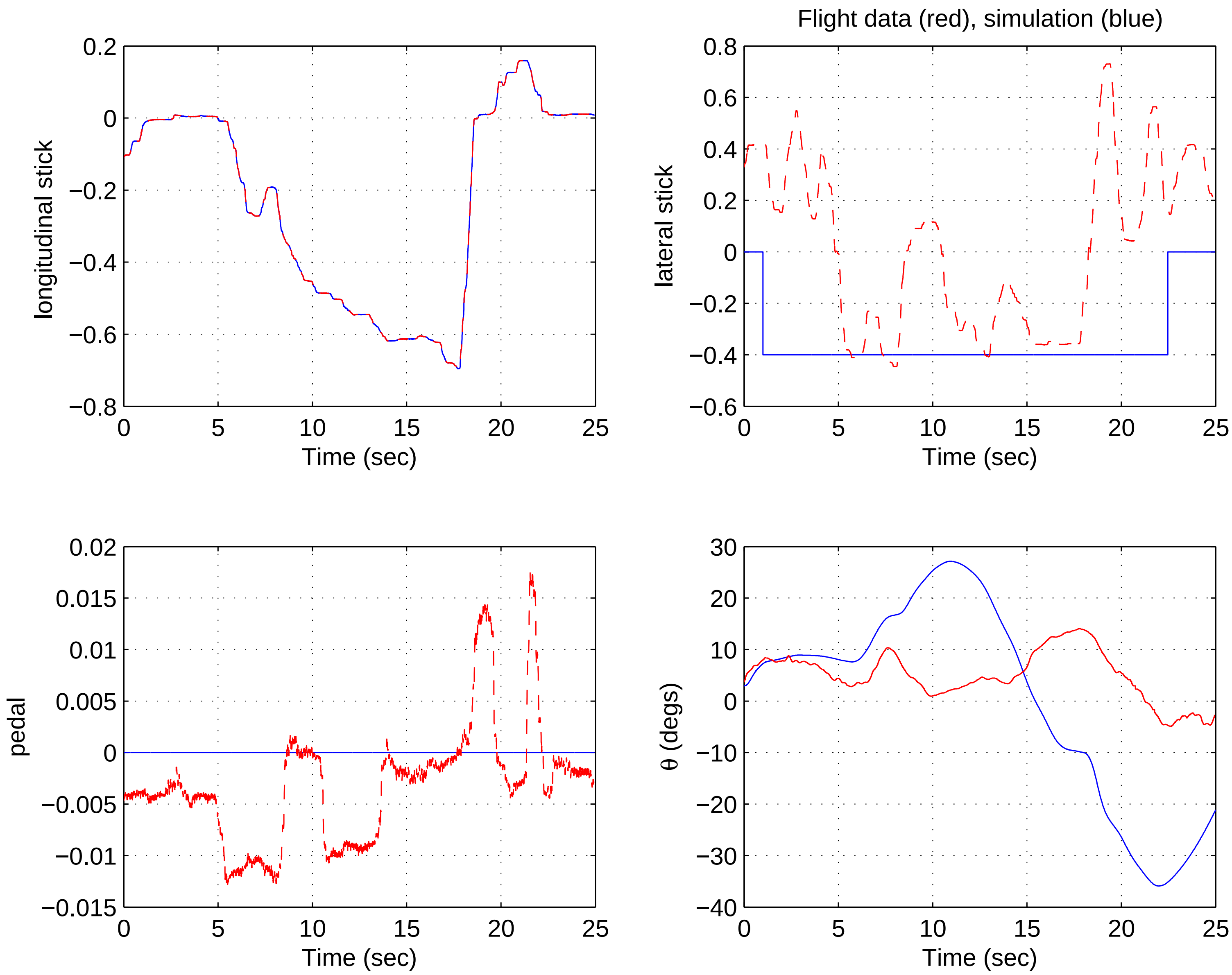
To reduce the time associated with flight test envelope expansion, flying the aircraft along a specific trajectory which encounters a variety of operating conditions can be used to complement standard fixed operating point tests.

- `ulinearize` (R2009b) generates a LTI uncertain system at a given operating point(s) based on `linearize`.
- Generate a family of uncertain linearized models along a trajectory.

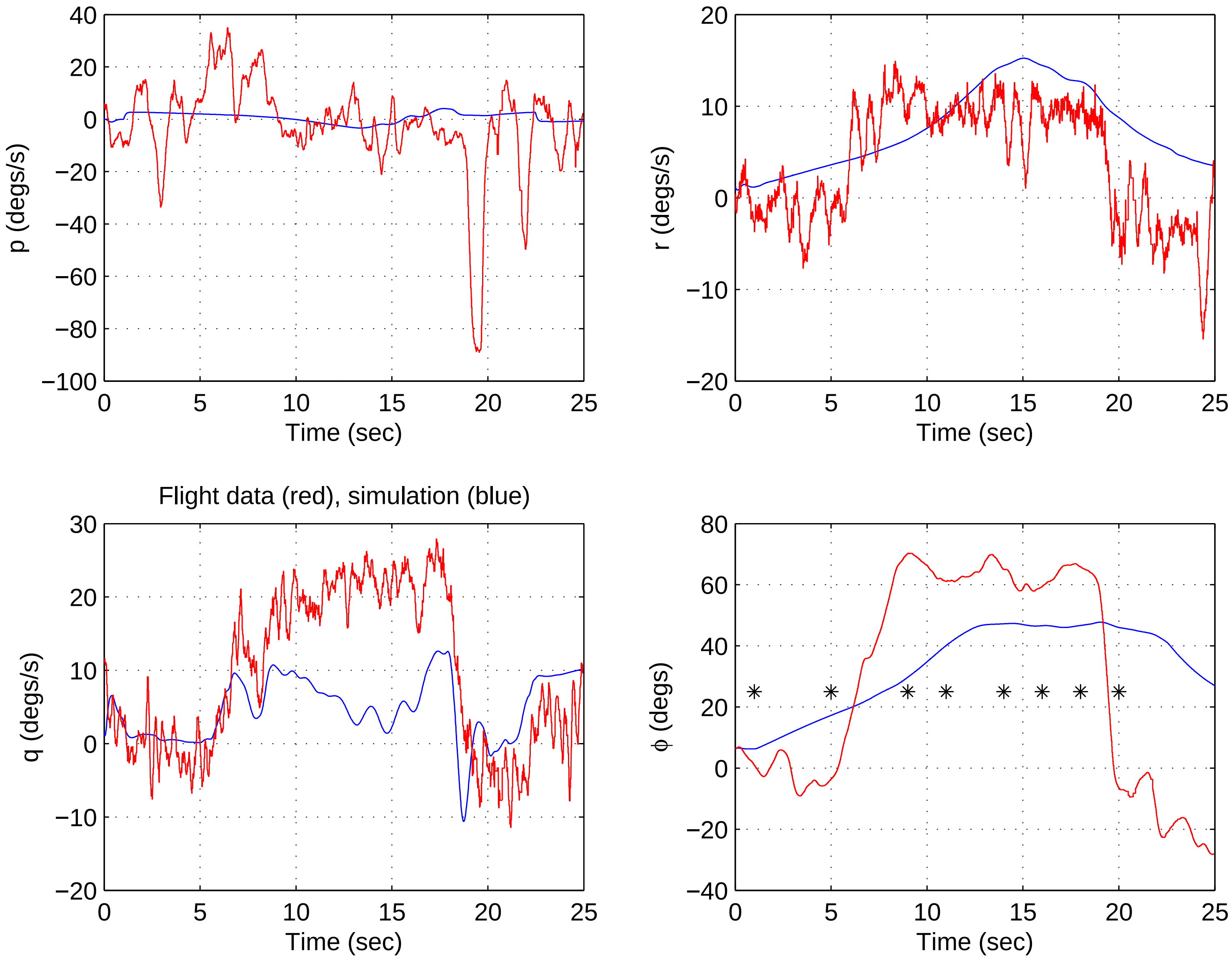
GTM Flight Data and Simulation: 1



GTM Flight Data and Simulation: 2



GTM Flight Data and Simulation: 3



GTM Trajectory: $u = 95ft/s, 1320ft$ altitude

Linear analysis along constant bank angle turn

```
loop_op = operpoint('gtm_design');
ActOut = ['gtm_design/GTM_plant/Actuators/Bus Selector2'];
loopio_gtm(1) = linio(ActOut,1,'outin','on');    % Lail_cmd
loopio_gtm(2) = linio(ActOut,2,'outin','on');    % Rail_cmd
loopio_gtm(3) = linio(ActOut,5,'outin','on');    % elev_cmd
loopio_gtm(4) = linio(ActOut,6,'outin','on');    % rud_cmd
tlin = [1 5 9 11 14 16 18 20];
[ugtmloop,ugtmloopOP] = ...
  ulinearize('gtm_design',tlin,loopio_gtm,loop_op);
```

GTM Trajectory: $u = 95ft/s, 1320ft$ altitude

Classical gain and phase margins, worst-case margins

```
[cm_gtmloop,dm_gtmloop] = loopmargin(-ugtmloop.Nominal);
om = logspace(-1,1.5,40);
opt = wcgopt('ArrayDimPtWise',1,'MaxTime',0,'Sensitivity','off');
ugtmloopg = frd(ugtmloop,om);
for i=1:length(tlin)
  [wcmargi,wcmargo] = wcmargin(-ugtmloopg(:,:,i),opt);
  win{i} = wcmargi;
  wout{i} = wcmargo;
end
```

Excellent margins up to 11 seconds, which corresponds to air speeds between 55 and 65 ft/s, 40 to 50 deg bank angle

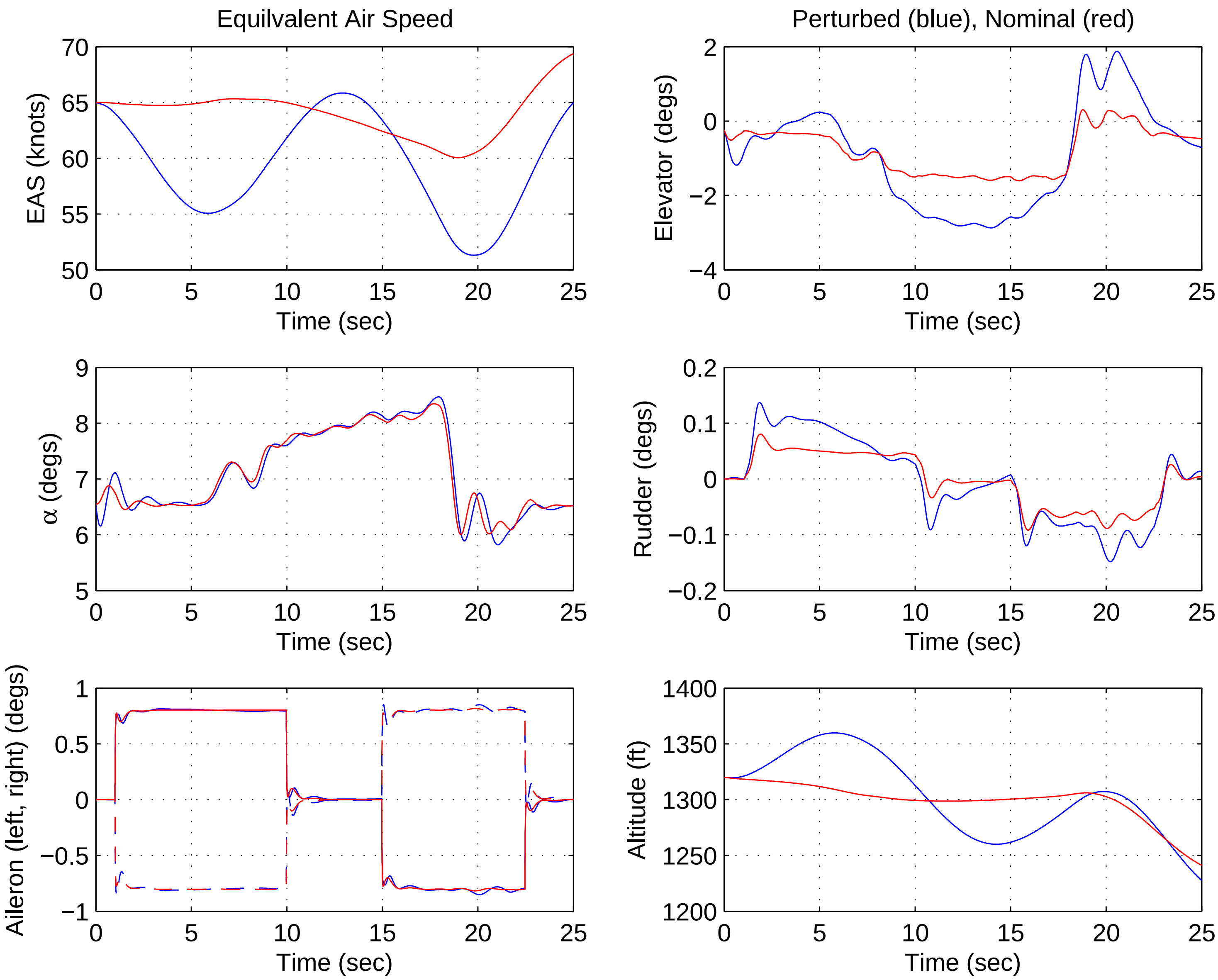
GTM Analysis: $u = 95ft/s, 1320ft$ altitude

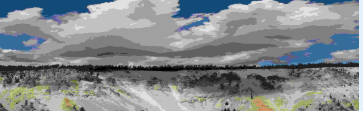
Time = 9 seconds

```
>> dm_gtmloop(1,3)
ans =
     GainMargin: [0.0163 61.5273]
    PhaseMargin: [-88.1377 88.1377]
      Frequency: 17.8939
>> win{3}(1)
ans =
     GainMargin: [0.0332 30.0920]
    PhaseMargin: [-86.1934 86.1934]
      Frequency: 15.1178
          WCUnc: [1x1 struct]
    Sensitivity: []
```

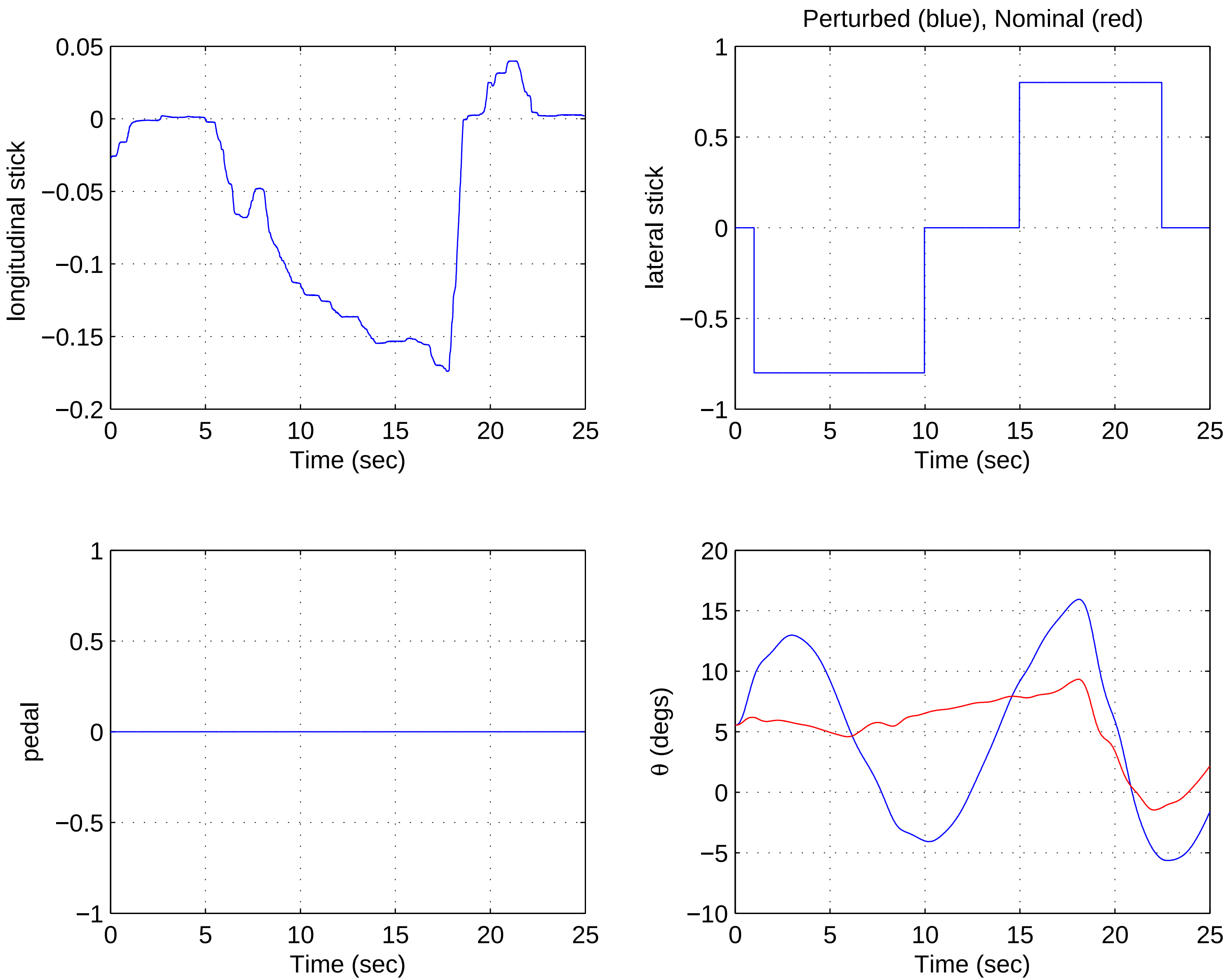
Time = 11 seconds

```
>> dm_gtmloop(1,3)
ans =
     GainMargin: [0.0193 51.7716]
    PhaseMargin: [-87.7869 87.7869]
      Frequency: 16.4869
>> win{3}(1)
ans =
     GainMargin: [1 1]
    PhaseMargin: [0 0]
      Frequency: 0.4375
          WCUnc: [1x1 struct]
    Sensitivity: []
```


Equilvalent Air Speed
EAS (knots)
70
65
60
55
50
0
5
10
15
20
25
Time (sec)
Perturbed (blue), Nominal (red)
Elevator (degs)
2
0
−2
−4
0
5
10
15
20
25
Time (sec)
α (degs)
9
8
7
6
5
0
5
10
15
20
25
Time (sec)
Rudder (degs)
0.2
0.1
0
−0.1
−0.2
0
5
10
15
20
25
Time (sec)
Aileron (left, right) (degs)
1
0.5
0
−0.5
−1
0
5
10
15
20
25
Time (sec)
Altitude (ft)
1400
1350
1300
1250
1200
0
5
10
15
20
25
Time (sec)

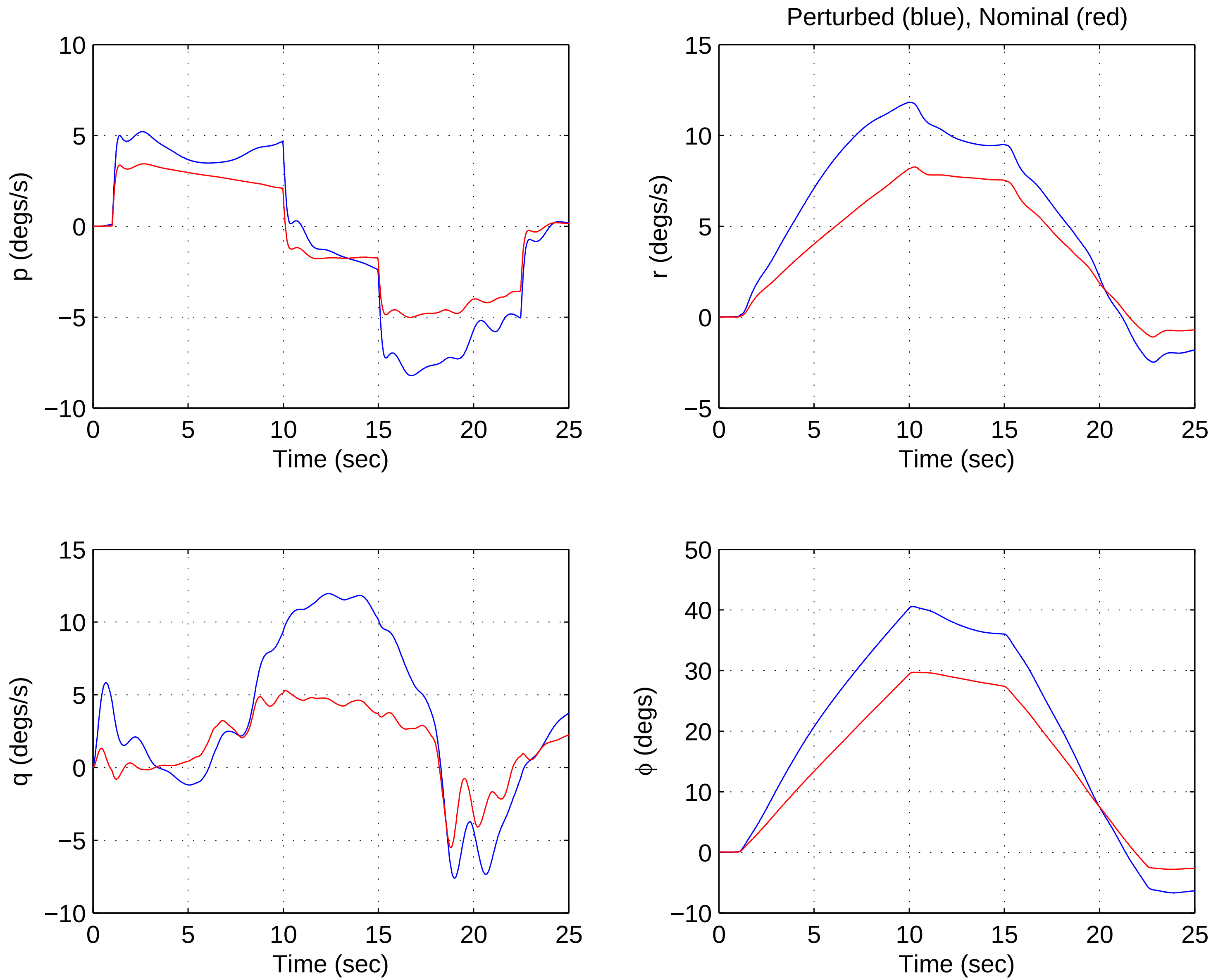


GTM Flight Data and Simulation: 2



GTM Flight Data and Simulation: 3

Poor margins at 11 sec, air speed 60 to 70 ft/s, 25 to 30 deg bank angle.



Worst-case Simulation

- Worst-case simulation is a time-domain robust performance test performed directly on a parameterized nonlinear model.
- `wcsim` performs worst-case simulation on a nonlinear Simulink model that contains uncertain real parameters.
- Advantage: There is great flexibility in choosing time-domain performance metrics.
- Disadvantage: Gradient based optimization can be time-consuming and is not guaranteed to find the globally optimal solution.

Worst-case Simulation Problem

Consider a parameterized set of nonlinear ODEs:

$$\begin{aligned} \dot{x}(t) &= f(x(t), t, p) \\ y(t) &= h(x(t), t, p) \\ x(0) &= x_0 \end{aligned}$$

$p \in P \subseteq \mathbb{R}^{n_p}$ is a constant parameter vector upon which the model depends and P is the set of allowable parameter value. The worst-case simulation problem is:

$$\max_{p \in P} G(y_p)$$

subject to: y_p is the output of the nonlinear system for parameter p

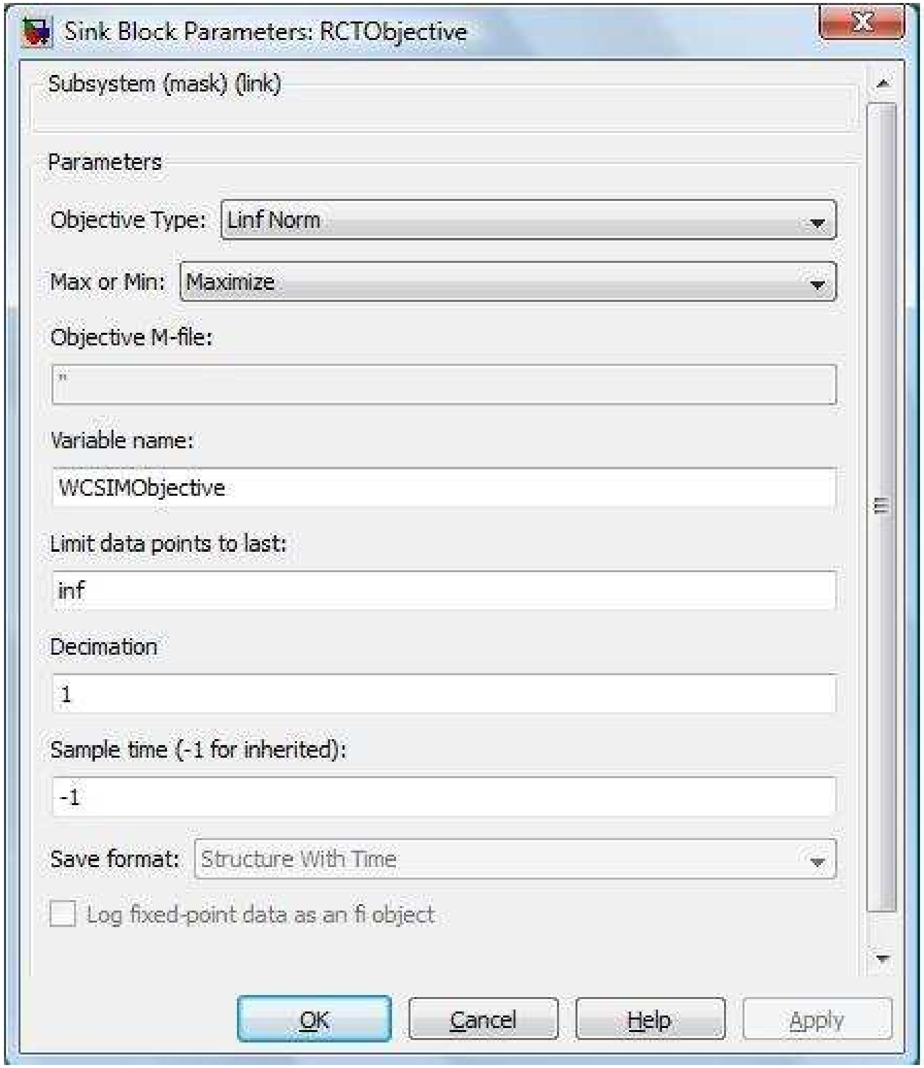
$$l \leq H(y_p) \leq u$$

where G and H are objective and constraint functions. For example, $G(y) := \|y\|_\infty$ is the peak magnitude of y and $G(y) := \left[\int_0^{t_f} y^T(t) y(t) dt\right]^{1/2}$ is the L_2 norm. The following slides describe how this problem is formulated in Simulink.

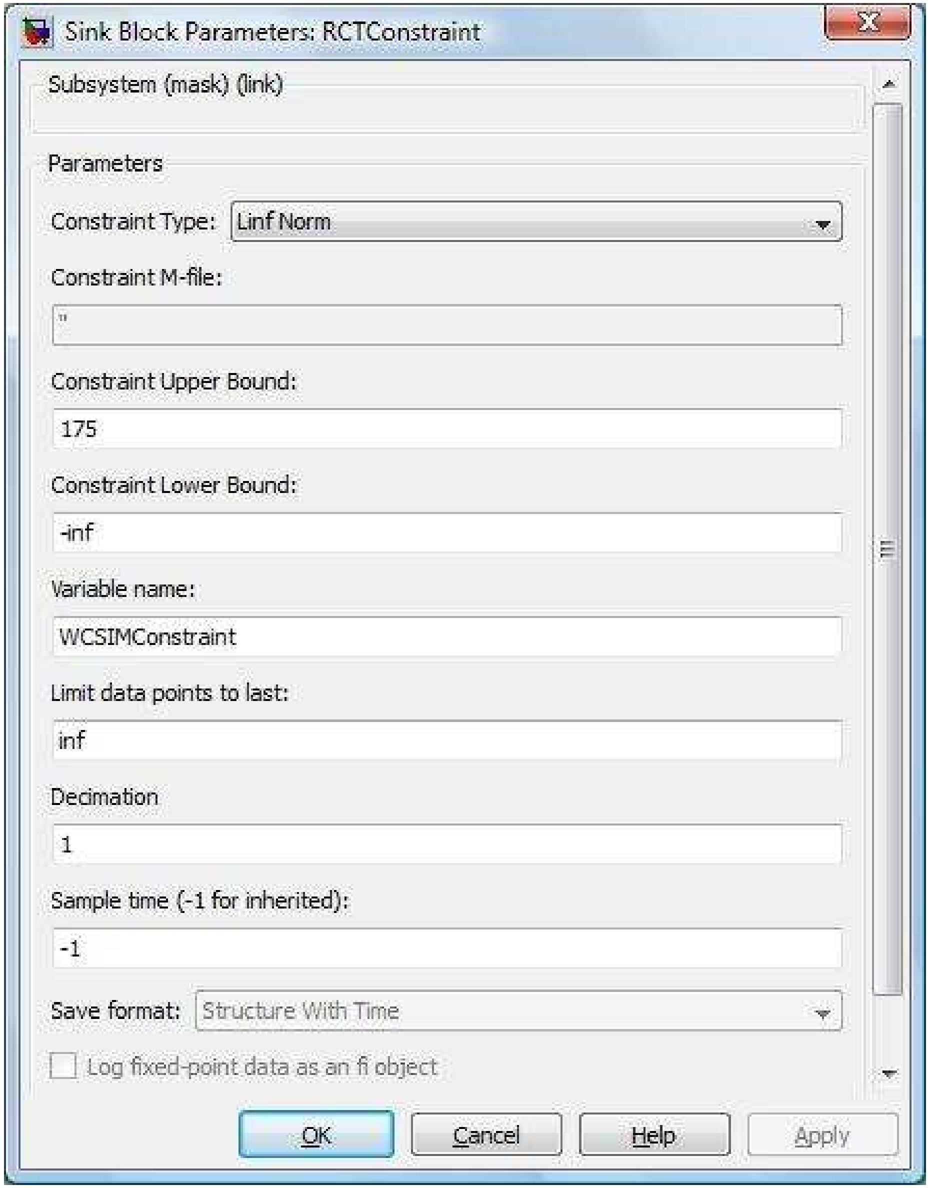
`wcsim`: Parameters

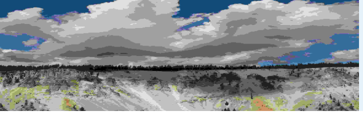
- Robust Control Toolbox (RCT) Uncertain State Space (USS) blocks can be used to create models of parameter-dependent nonlinear systems within Simulink.
- A Simulink model that has USS blocks depending on n_p real parameters is in the form of a parameterized nonlinear ODE.
- The allowable set of parameter vectors is in the form $P := \{p \in \mathbb{R}^{n_p} \; : \; \underline{p}_i \le p \le \bar{p}_i, \; i = 1, \ldots, n_p\}$.

- The objective function for a worst-case simulation is specified with an RCT `Objective Function` block.
- The `Objective Function` block is similar to a `To Workspace` block with the `Save Format` set to `Structure With Time`.
- The objective function is specified through the block dialog box.
- Simulating the system will create an output variable in the workspace with the all the fields generated by a `To Workspace` block: `time`, `signals`, and `blockName`.
- The output variable will have the additional field `objective`. The Objective function value is stored in `objective.value`.



- The constraints for a worst-case simulation are specified with RCT `Constraint Function` blocks.
- The `Constraint Function` blocks are similar to the `Objective Function` blocks.
- The constraint function is specified through the block dialog box.
- The output variable generated by a `Constraint Function` will have the following fields in addition to the normal `To Workspace` fields: `constraint.value`, `constraint.lowerbound`, and `constraint.upperbound`.





wcsim: Optimization

- wcsim uses fmincon to perform the gradient-based optimization and thus requires the optimization toolbox.
- wcsim returns the worst-case uncertainties in the structure wcuvars.
- The total computation time for wcsim with no constraint blocks will be roughly $(n_p + 1)M\tau$ where τ is the computation time for one simulation and M is the number of iterations.
  - Unconstrained problems with n_p parameters require $n_p + 1$ objective function evaluations per iteration.
  - Constrained problems will require additional evaluations and hence will take more time.
- The convergence of fmincon depends on the starting value of the parameter vector.
  - The initial parameter values can be specified in the Uncertainty Values field of the USS blocks.

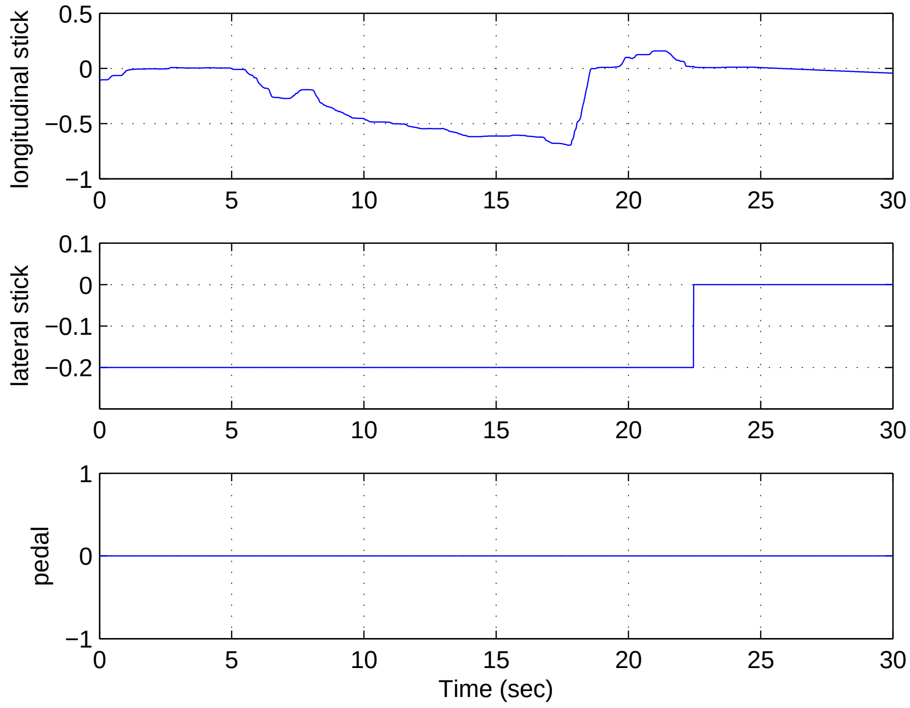
GTM Parameter Uncertainties

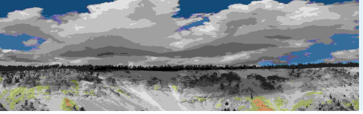
The GTM model was modified to have a total of $n_p = 18$ real uncertainties.

- The forces and moments in the GTM model are calculated using six body-axis aerodynamic coefficients: C_u, C_v, C_w, C_p, C_q, and C_r.
- Each of these six coefficients is a sum of three terms: 1) basic airframe 2)control surface increments, and 3) angular rate increments.
- 20% uncertainty was placed on each of the basic airframe and control surface coefficients.
- 40% uncertainty was placed on each of the angular rate increments.

GTM Unconstrained Worst-Case Simulation

- For each attitude signal we first performed an (unconstrained) worst-case simulation with all combinations of maximizing and minimizing the L_∞ and L_2 norms.
- We set the optimization options to restrict `fmincon` to at most four iterations.
- All parameters were initialized at their nominal values and all dynamic actuator uncertainties were held fixed at their nominal values.
- In all simulations, the baseline GTM controller was used with the pilot inputs shown in the figure.





GTM Unconstrained Worst-Case Simulation Results

Cost Func.	Comp. Time	Step 0	Step 1	Step 2	Step 3
$\max L_\infty(\phi)$	977.4	1.11	1.56	1.98	2.00
$\min L_\infty(\phi)$	888.0	1.11	0.26	0.25	0.24
$\max L_2(\phi)$	935.5	4.29	6.05	6.36	6.36
$\min L_2(\phi)$	867.8	4.29	0.82	0.81	0.81
$\max L_\infty(\theta)$	905.2	0.64	0.65	0.88	0.89
$\min L_\infty(\theta)$	893.9	0.64	0.53	0.50	0.48
$\max L_2(\theta)$	1015.0	1.74	2.37	2.78	2.78
$\min L_2(\theta)$	907.8	1.74	1.60	1.46	1.44
$\max L_\infty(\psi)$	1045.6	7.87	11.13	15.08	15.10
$\min L_\infty(\psi)$	843.8	7.87	2.54	X	X
$\max L_2(\psi)$	1039.2	26.79	50.81	51.02	51.06
$\min L_2(\psi)$	1014.1	26.79	11.90	11.88	11.84

Table 1: Comparison of worst-case attitude and computation times

It takes roughly $\tau = 11.72$sec to perform one simulation of the GTM Simulink model. The total time to run `wcsim` is expected to be $(M+1)(n_p+1)\tau = 891$ sec. This estimate is in fair agreement with the actual computation times provided in the Table.

GTM Unconstrained Worst-Case Simulation Results

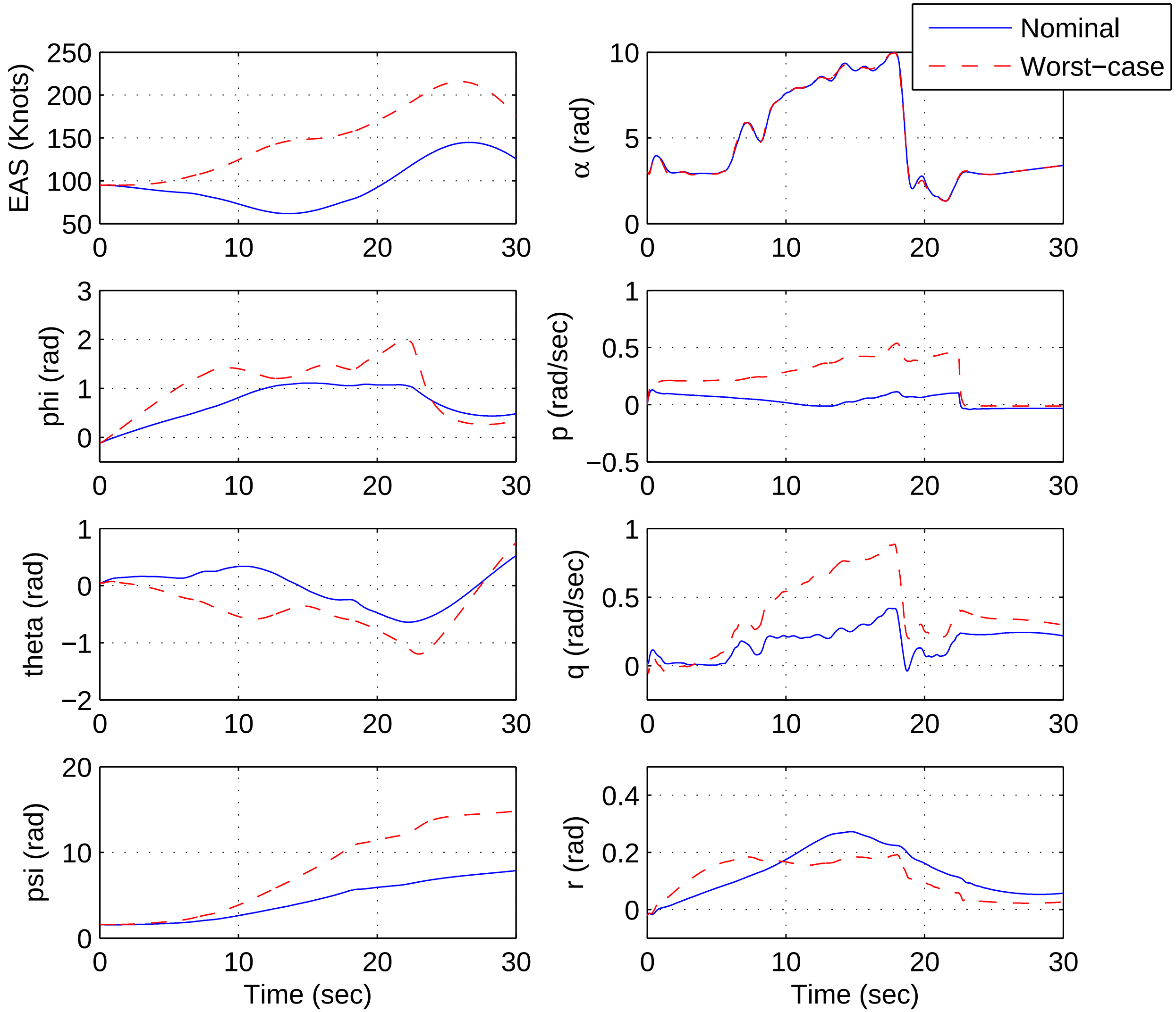


Figure 1: Nominal and wcsim Roll for $\max L_\infty(\phi)$

GTM Unconstrained Worst-Case Simulation Results

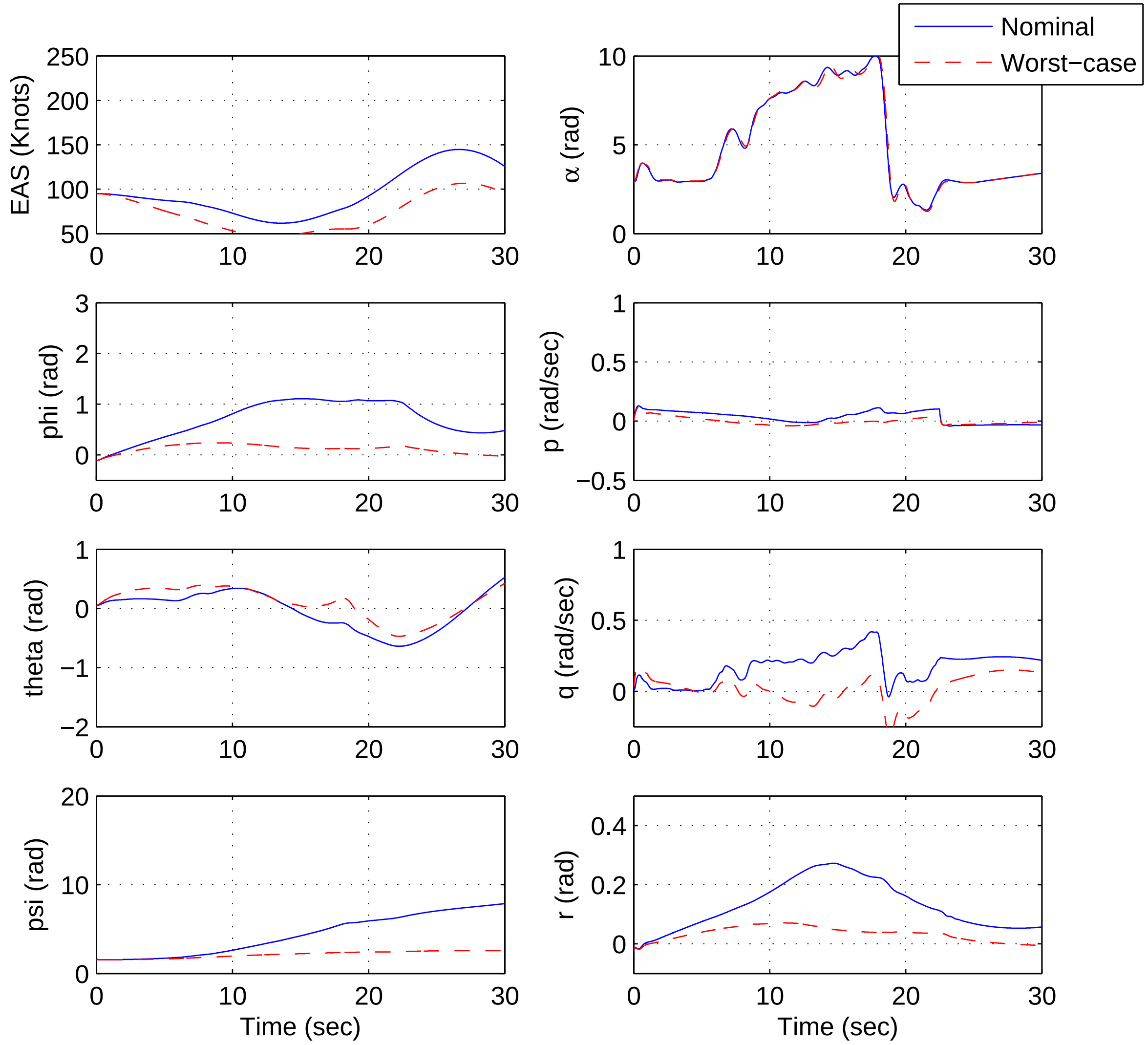


Figure 2: Nominal and wcsim Roll for $\min L_\infty(\phi)$

GTM Constrained Worst-Case Simulation

- One RCT `Constraint Function` block was added to the GTM Simulink model to constrain EAS to be less than 175 knots.
- The objective function is set to maximize the L_∞ norm of ϕ.
- We set the optimization options to restrict `fmincon` to six iterations.
  - Six iterations were needed for convergence in this example.
- After five iterations, the objective function is $\|\phi\|_\infty = 1.66$. For comparison, the unconstrained worst-case simulation achieved an objective function of 2.00.
- `wcsim` returns parameter values which cause the EAS to just touch the constraint.
- The total computation time was 2169.2 sec.

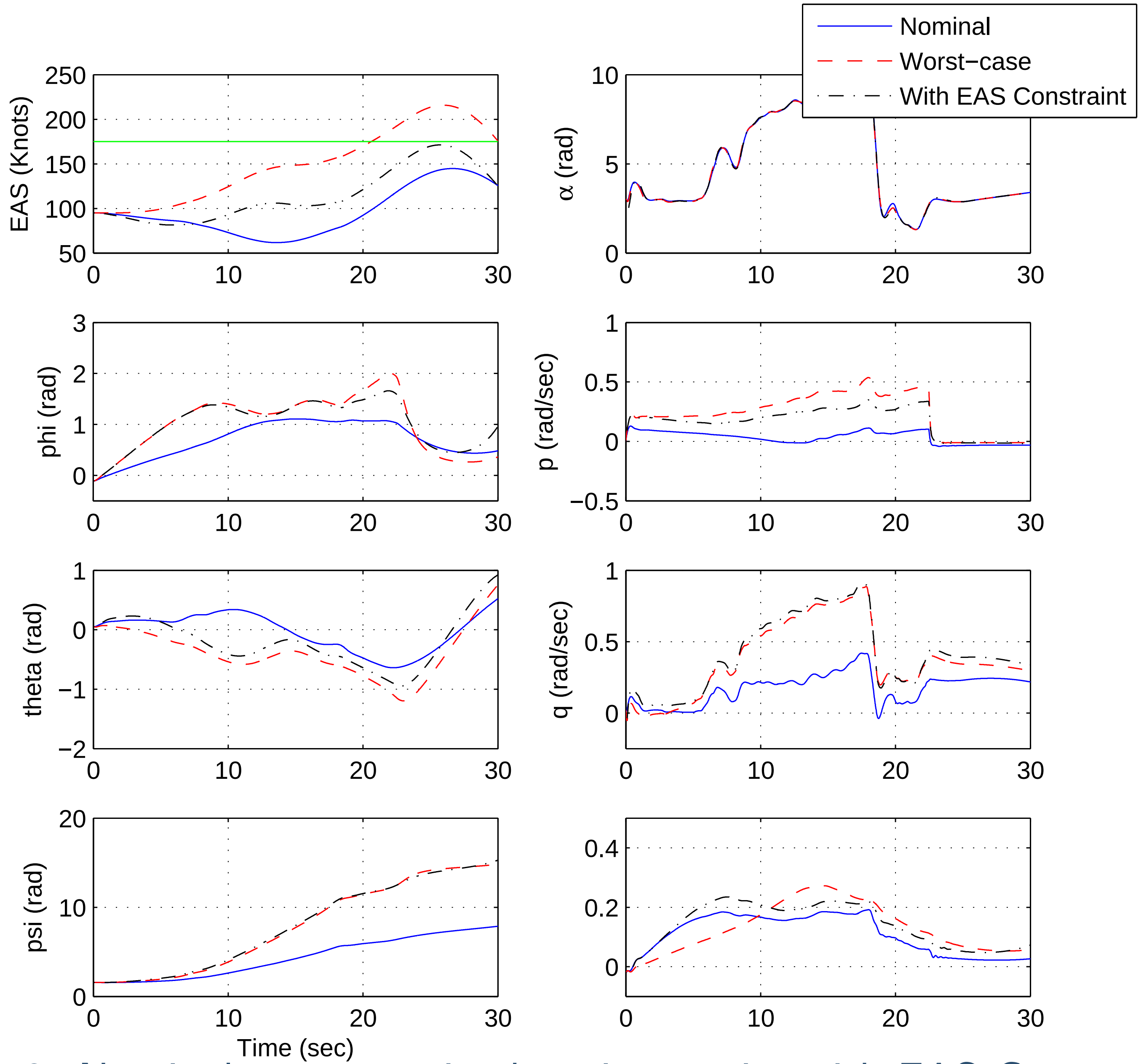


Figure 3: Nominal, unconstrained wcsim, wcsim with EAS Constraint