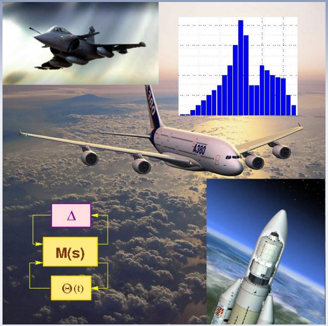


LFT modelling and robustness analysis

Precision versus complexity...

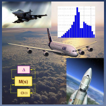
A collective work by (in alphabetical order) :

J-M Biannic, C. Döll, G. Ferreres
F. Lescher and C. Roos

(ONERA/DCSD)

MOSAR meeting – January 26^{th}, 2011.

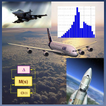
ONERA
THE FRENCH AEROSPACE LAB



How to manage complexity in robustness analysis ?

- In industrial problems, sophisticated models with high level parameters are often encountered (e.g. a rigid / flexible airplane depending on Mach, dynamic pressure and filling degrees of the tanks. . .). This leads to high complexity LFT models, with many highly repeated parameters.
- When analyzing an uncertain closed-loop plant, possibly augmented with multipliers, the order of the representation may be too high for LMI state-space solutions.
- LFT complexity is to be minimized at each step of the modelling phase.
- Keeping a reasonable computational burden despite the unavoidable complexity of the problem:
  - numerous and highly repeated parameters,
  - high order models (because of flexible modes, dynamic multipliers, weighting functions,...)

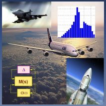
typically requires "KYP Lemma free" and possibly "LMI free" methods



On "KYP Lemma & LMI free" methods

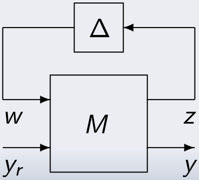
- Whatever the framework (using multipliers such as μ/IQC based techniques or involving Lyapunov functions), a robustness analysis problem often leads to a minimization problem under an **infinite set of LMI constraints**.
- The KYP Lemma is a powerful tool thanks to which the above problem is solved by considering a **single state-space constraint**. However, this relaxation technique introduces numerous scalar variables.
- In this talk, to limit the number of constraints without introducing any slack variable, we focus on a two-step procedure:
  - Optimization on a frequency or parametric grid
  - Validation between grid points

  When possible, LMI techniques are to be avoided in the first step.



Outline

Definition of uncertain or varying parameters



$$\theta_i = \theta_{i,C} + s_{\theta_i}\,\delta\theta_i$$

with

$$\theta_{i,C} = \frac{\theta_{i,max} + \theta_{i,min}}{2}$$

$$s_{\theta_i} = \frac{\theta_{i,max} - \theta_{i,min}}{2}$$

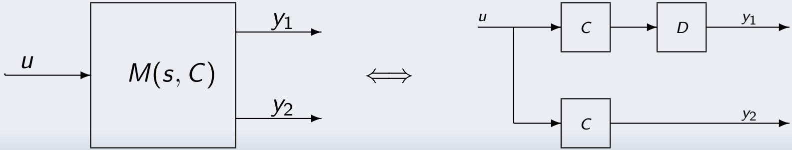
$$\delta\theta_i \in [-1,\ 1]$$

$$\Delta = \mathrm{diag}(\delta\theta_1\, I_{k_1}, \ldots, \delta\theta_n\, I_{k_n}, \ldots \Delta_{NL,l_1}, \ldots, \Delta_{NL,l_o}, \ldots \Delta_{m_1}(s), \ldots, \Delta_{m_p})$$

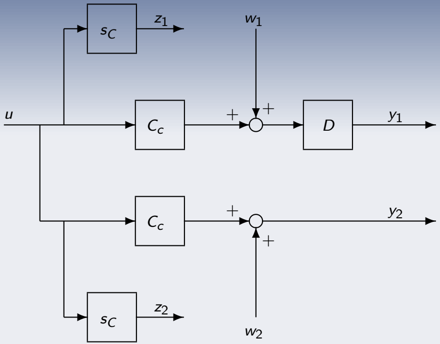
The size of an LFR is the size of the matrix Δ.

Basic transformations for LFR modelling (1/4)

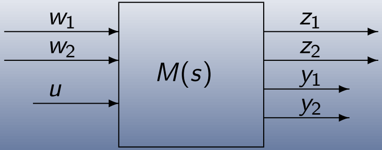
LFR modelling seems to be a straightforward activity.



Basic transformations for LFR modelling (2/4)



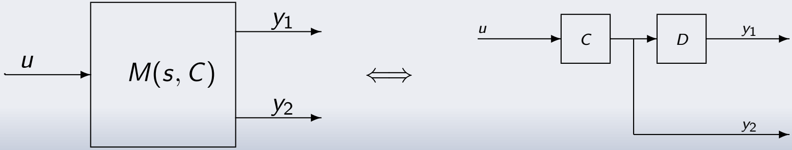
$\Longleftrightarrow$



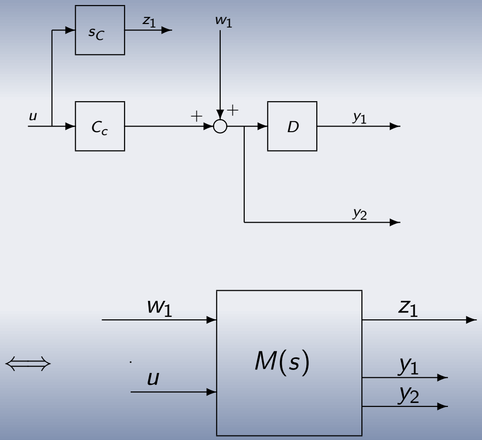
$$\Delta = \begin{bmatrix} \delta_C & 0 \\ 0 & \delta_C \end{bmatrix}$$

Basic transformations for LFR modelling (3/4)

But you can considerably reduce the size by a *good* symbolic pre-processing, here a factorization to the left.

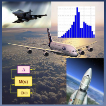


Basic transformations for LFR modelling (4/4)



$$\Delta = \left[\ \delta_C\ \right]$$

This operation is done by the LFR Toolbox functions `sym2lfr` and/or `symtreed`.

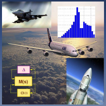


Problem statement

In the analytical model description, the following elements appear very frequently:

1. Look-up tables:
    - Model coefficients often depend on system parameters, for example $C_{yr}(M, \alpha)$, $V_{tas}(V_{cas}, M)$.
    - Controller gains are scheduled with respect to some measurements, for example $K_1(x_{cg})$ or $K_2(V_{cas})$.
2. Functions: exponential, trigonometrical, irrational, piece-wise (non-)linear

Or, the system is described by a family of linearized models ($A(\Delta_i)$, $B(\Delta_i)$, $C(\Delta_i)$, $D(\Delta_i)$) where Δ_i describe the trim conditions. The consistency of the state vectors must first be ensured (modal truncation, reordering) in order to ensure smooth trajectories of the eigenvalues. The continuum of the frequency responses is improved by biconvex optimization.



Rational interpolation & LFR modelling (1/3)

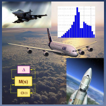
In order to come up with low-order LFRs, the non-rational data must be replaced by rational or polynomial expressions of minimum order and/or with a minimum number of monomials before being transformed into LFRs.

Polynomial interpolations are performed:

$$z(\overline{\delta}) = \sum_{k=1}^{n_p} \gamma_k p_k(\overline{\delta})$$

where $(p_k)_{k\in[1,n_p]}$ is a set of multivariate monomials and $(\gamma_k)_{k\in[1,n_p]}$ are parameters to be determined.
Usually, this problem is solved by minimizing the quadratic error (Least Square):

$$J(\Gamma) = (Z - P\Gamma)^T (Z - P\Gamma)$$

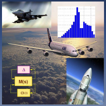


Rational interpolation & LFR modelling (2/3)

If **orthogonal modelling functions** such that $P_i P_j = 0 \; \forall i \neq j$ are used, then the minimum value J_{opt} of $J(\Gamma)$ is given by:

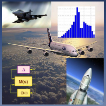
$$J_{opt} = Z^T Z - \sum_{k=1}^{n_p} \frac{(P_k^T Z)^2}{P_k^T P_k}$$

The reduction in the Least Squares criterion $J(\Gamma)$ resulting from the inclusion of the term $\gamma_k p_k(\bar{\delta})$ does not depend on $p_j(\bar{\delta})$ whatever $j \neq k$. This allows to **evaluate each monomial in terms of its ability to reduce $J(\Gamma)$**, regardless of which other monomials are selected.

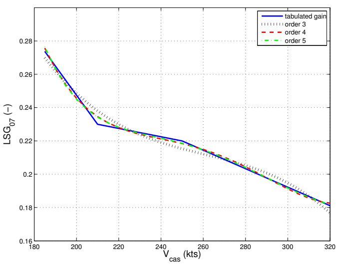


Rational approximation & LFR modelling (3/3)

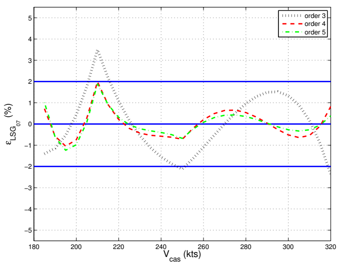
- **data2sym.m** uses a classical Least Square approach.
- **data2poly.m** exploits the Orthogonal Least Square of the previous slide to reduce the LFR complexity.
- In order to reduce even more the LFR complexity, rational approximation has very recently been dealt with using either Levenberg-Marquardt algorithms or quadratic programming on the one hand and Radial Basis Function (RBF) neural networks or Particle Swarm Optimization (PSO) on the other hand. The function **data2rat.m** will soon be added to the LFR toolbox.



Examples for polynomial approximation (1/2)



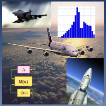
(a) Approximation



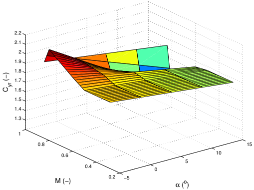
(b) Approximation error

A 4^{th}-order polynomial is needed in order to satisfy $\epsilon_{max} \leq 2\%$ on the whole V_{cas}-range [185, 320] kts :

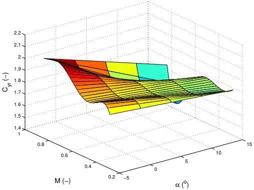
$$\widehat{K} = c_0 + V_{cas}\ \{c_1 + V_{cas}\ [c_2 + V_{cas}\ (c_3 + c_4\ V_{cas})]\}$$



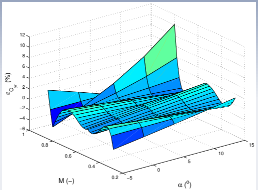
Examples for polynomial approximation (2/2)



(c) Initial data



(d) Approximation

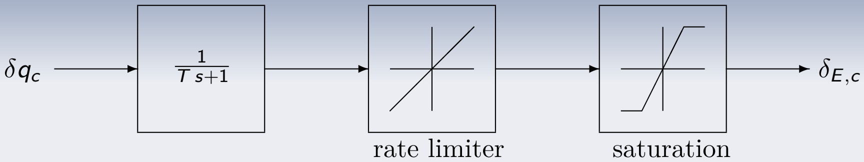


(e) Approximation error

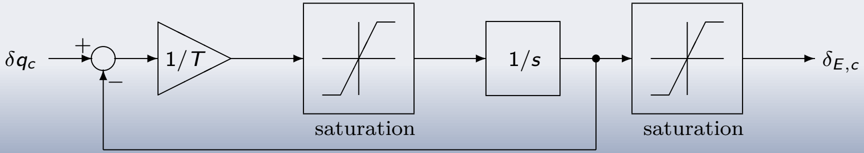
$$\begin{aligned}\hat{C}_{yr} \quad = \quad & c_0 + c_1\, M + c_2\, \alpha + c_3\, M^2 + c_4\, M\alpha + \\ & + c_5\, M^3 + c_6\, M^2\alpha + c_7\, M^4 + c_8\, M^3\alpha + \\ & + c_9\, M^5 + c_{10}\, M^4\alpha\end{aligned}$$

for a chosen maximum error $\epsilon_{max} = 10\%$.

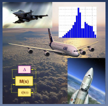
LFRs for rate limiters and position saturations (1/2)



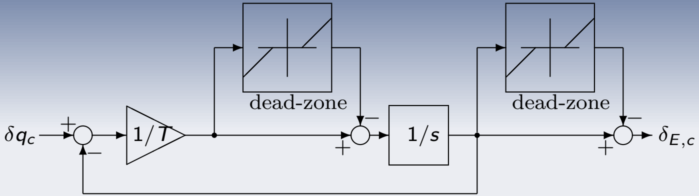
(f) Initial implementation



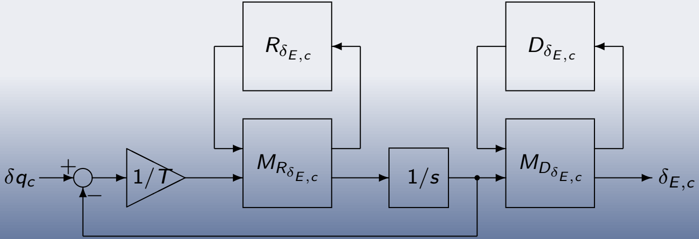
(g) Intermediate implementation of the rate limiter as a saturation



LFRs for rate limiters and position saturations (2/2)



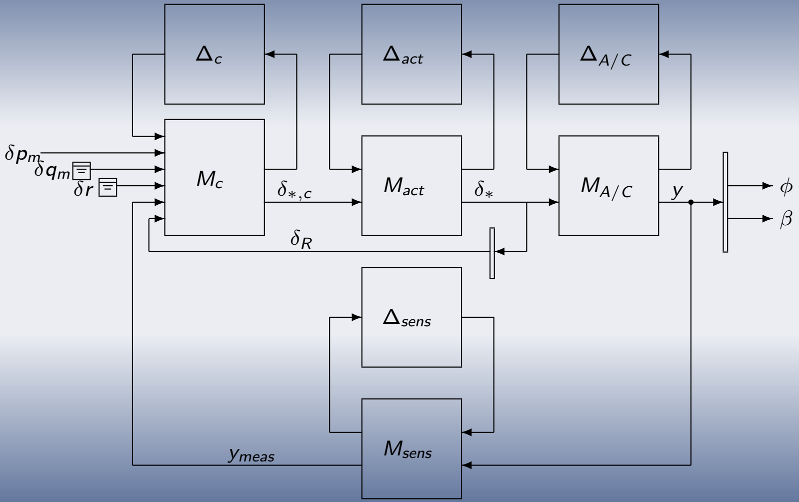
(h) Intermediate implementation of both saturations as dead-zones

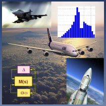


(i) LFR implementation

$$\Delta = \mathrm{DZ}(z)$$
$$M = \left(\begin{array}{c|c} 0 & 1 \\ \hline -1 & 1 \end{array}\right)$$

A closed-loop system (1/2)





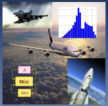
A closed-loop system (2/2)

`slk2lfr.m` opens the loops before and after the Δ_i introducing the artificial inputs w_i and outputs z_i, reorders them in a block-diagonal form Δ_{cl}, and finally linearizes the system in order to obtain the state space representation:

$$\begin{aligned} \dot{x} &= A\,x + \underbrace{\begin{bmatrix} B_1 & B_2 \end{bmatrix}}_{B} \begin{pmatrix} w \\ u \end{pmatrix} \\ \begin{pmatrix} z \\ y \end{pmatrix} &= \underbrace{\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}}_{C} x + \underbrace{\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}}_{D} \begin{pmatrix} w \\ u \end{pmatrix} \end{aligned}$$

of the nominal system and repartitions $(A,\ B,\ C,\ D)$ such that

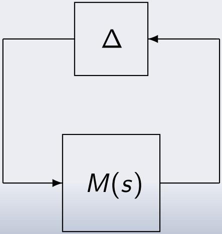
$$\begin{aligned} \begin{pmatrix} \dot{x} \\ z \end{pmatrix} &= \underbrace{\begin{bmatrix} A & B_1 \\ C_1 & D_{11} \end{bmatrix}}_{M_{11}} \begin{pmatrix} x \\ w \end{pmatrix} + \underbrace{\begin{bmatrix} B_2 \\ D_{12} \end{bmatrix}}_{M_{12}} u \\ y &= \underbrace{\begin{bmatrix} C_2 & D_{21} \end{bmatrix}}_{M_{21}} \begin{pmatrix} x \\ w \end{pmatrix} + \underbrace{D_{22}}_{M_{22}} u \end{aligned}$$



Problem statement

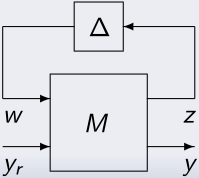
Let $M(s)$ be a stable LTI plant and Δ a time-invariant uncertainty matrix with a given structure $\boldsymbol{\Delta}$. Let $B(\boldsymbol{\Delta}) = \{\Delta \in \boldsymbol{\Delta} \ : \ \overline{\sigma}(\Delta) < 1\}$.

Problem 1: robust stability



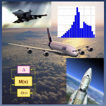
Compute the maximum value k_{max} s.t. the interconnection is stable $\forall \Delta \in k_{max} B(\boldsymbol{\Delta})$

Problem 2: robust H_∞ performance



If robust stability is ensured, compute

$$\gamma_{max} = \max_{\Delta \in B(\boldsymbol{\Delta})} \|\mathcal{F}_u(M(s), \Delta)\|_\infty$$



Brief introduction to μ-analysis

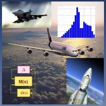
Structured singular value μ

The s.s.v. $\mu(M(j\omega))$ is the inverse of the size $\overline{\sigma}(\Delta)$ of the smallest perturbation $\Delta \in \mathbf{\Delta}$ satisfying $det(I - \Delta M(j\omega)) = 0$. The robustness margin k_{max} is thus obtained as:

$$k_{max} = \frac{1}{\max\limits_{\omega \in \mathbb{R}_+} \mu(M(j\omega))}$$

In the general case, the exact computation of $\mu(M(j\omega))$ is NP hard. A classical strategy consists of:

- computing an **upper bound** μ_{UB} using polynomial-time algorithms to obtain a guaranteed value of the robustness margin,
- computing a **lower bound** μ_{LB} to evaluate conservatism.



Computation of a μ upper bound

Computing a guaranteed robustness margin involves the computation of a μ upper bound for each frequency $\rightarrow$ infinite-dimensional problem.

Characterization of a mixed-μ upper bound

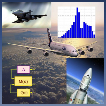
Let β be a positive scalar. If there exist matrices $D \in \mathcal{D}$ and $G \in \mathcal{G}$ s.t.:

$$\overline{\sigma}\left((I+G^2)^{-1/4}\left(\frac{DM(j\omega)D^{-1}}{\beta} - jG\right)(I+G^2)^{-1/4}\right) \leq 1$$

where $\mathcal{D} = \{D \in \mathbb{C}^{m\times m} : det(D) \neq 0 \text{ and } \forall\Delta \in \boldsymbol{\Delta}, D\Delta = \Delta D\}$ and $\mathcal{G} = \{G \in \mathbb{C}^{m\times m} : \forall\Delta \in \boldsymbol{\Delta}, G\Delta = \Delta^* G\}$, then $\mu(M(j\omega)) \leq \beta$.

Two classical strategies:

- using a frequency grid $\rightarrow$ not reliable, especially in case of flexible systems (over-evaluation of the robustness margin)
- considering frequency as a repeated parametric uncertainty $\rightarrow$ not applicable for high-order systems (computational burden)



Key idea of the method

A μ upper bound β_i and matrices D_i, G_i are computed for a frequency ω_i.

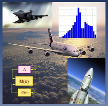
$\beta_i \leftarrow (1+\epsilon)\beta_i$ is then **slightly increased** to enforce a strict inequality:

$$\overline{\sigma}\left((I+G^2)^{-1/4}\left(\frac{DM(\omega_i)D^{-1}}{\beta_i} - jG\right)(I+G^2)^{-1/4}\right) < 1$$

The **key step** is to compute the largest frequency interval $I(\omega_i) \ni \omega_i$ s.t.:

$$\forall \omega \in I(\omega_i), \overline{\sigma}\left((I+G^2)^{-1/4}\left(\frac{DM(\omega)D^{-1}}{\beta_i} - jG\right)(I+G^2)^{-1/4}\right) \leq 1$$

- β_i is thus a guaranteed μ upper bound on the whole frequency interval $I(\omega_i)$, and not only for a single frequency ω_i.
- The determination of the above frequency segment can be achieved by computing the eigenvalues of a Hamiltonian-like matrix.

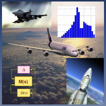


Algorithmic issues

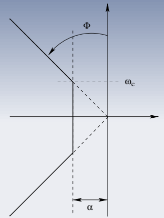
The resulting algorithm consists of an initialization phase followed by a repeated treatment on a list of intervals.

1. **Initialization phase:**
   - (a) Set $\beta_{max} = 0$.
   - (b) Initialize the list of frequency intervals $\mathcal{I}$ to be investigated, for example $\mathcal{I} = \{\Omega_1\} = \{[\omega_{min}, \omega_{max}]\}$.
2. **While $\mathcal{I} \neq \varnothing$, repeat:**
   - (a) Select an interval $\Omega_i \in \mathcal{I}$ and choose a pulsation $\omega_i \in \Omega_i$.
   - (b) Apply the aforementioned procedure.
   - (c) Set $\beta_{max} \leftarrow \max(\beta_i, \beta_{max})$.
   - (d) Update $\mathcal{I}$ by eliminating $I(\omega_i)$ from the list: $\mathcal{I} \leftarrow \mathcal{I} \setminus I(\omega_i)$

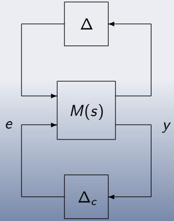
β_{max} is progressively increased while the frequencies in $\mathcal{I}$ are eliminated. At the end, $k_{UB} = 1/\beta_{max}$ is a **guaranteed** robustness margin on $[\omega_{min}, \omega_{max}]$.



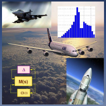
Extensions to performance analysis



- **Modal performance**. It suffices to compute a μ upper bound on the borderline of a truncated sector.



- **H_∞ performance**. These statements are equivalent:
  1. $\gamma_{max} = \max\limits_{\Delta \in B(\mathbf{\Delta})} \|\mathcal{F}_u(M(s), \Delta)\|_\infty \leq \gamma$,
  2. the size $\overline{\sigma}(\Delta_c)$ of the smallest perturbation $\Delta_c = \mathbb{C}^{p \times p}$ s.t. $det(I - M(j\omega)diag(\Delta, \Delta_c)) = 0$ for some $\Delta \in B(\mathbf{\Delta})$ and some $\omega \in \mathbb{R}_+$ is larger than $1/\gamma$,
  3. $\mu\left(diag(I, I/\sqrt{\gamma})M(j\omega)diag(I, I/\sqrt{\gamma})\right) \leq 1 \ \forall \omega \in \mathbb{R}_+$.

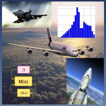


Extensions to performance analysis

- H_∞ **performance** (cont'd). For a given ω_i, the smallest value of γ s.t. $\mu_{UB}\left(diag(I, I/\sqrt{\gamma})M(j\omega_i)diag(I, I/\sqrt{\gamma})\right) < 1$ can be computed:
  - either directly using an LMI characterization of μ_{UB},
  - or iteratively (dichotomy or fixed-point) using the $\overline{\sigma}$ one.

The aforementioned algorithm can thus be applied to compute an upper bound γ_{LB} on the robust H_∞ performance γ_{max}.

- Note that the proposed algorithm can be further extended to **general skew-μ problems** where Δ_c is structured.



Computation of a μ lower bound

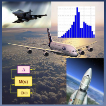
Main features of existing methods (power algorithm, gain-based algorithm):

+ constructive heuristics which compute worst-case uncertainties,
− frequency is fixed $\Rightarrow$ worst-cases can be missed even with a fine grid,
+ good results in the complex and mixed cases (fast and accurate),
− convergence problems in the purely real case (lower bound equal to 0).

Key idea of the proposed method: to obtain in polynomial time a tight μ lower bound over the whole frequency range rather than at a fixed frequency.

$\Rightarrow$ first search a perturbation Δ which brings one pole of the system near a chosen frequency point on the imaginary axis (good initial guess),

$\Rightarrow$ then consider Δ as a fictitious feedback gain allowing to **move this pole freely through the imaginary axis** to obtain a destabilizing perturbation.



Algorithmic issues

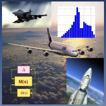
A 2-step procedure is performed at each point ω_i of a **rough frequency grid**:

1. apply the power algorithm to **a regularized μ problem** obtained by adding a small amount ϵ of complex uncertainties Δ_C to the real uncertainties Δ_R:

$$M_{reg}(j\omega_i) = \begin{bmatrix} M(j\omega_i) & \sqrt{\epsilon}M(j\omega_i) \\ \sqrt{\epsilon}M(j\omega_i) & \epsilon M(j\omega_i) \end{bmatrix} \quad \text{and} \quad \Delta_{reg} = diag(\Delta_R, \Delta_C)$$

2. extract the real part Δ_R^* of the resulting worst-case, then compute a matrix $\tilde{\Delta}_R$ which moves one pole of the closed-loop dynamics $A + B(\Delta_R^* + \tilde{\Delta}_R)C$ through the imaginary axis while minimizing $\overline{\sigma}\left(\Delta_R^* + \tilde{\Delta}_R\right)$.

- Step 2 can be recast as a **linear programming problem**.
- The imaginary axis can be crossed at a point $j\tilde{\omega}_i \neq j\omega_i$. Note that $\tilde{\omega}_i$ usually corresponds to a peak value on the μ plot $\Rightarrow$ **tight lower bound**.
- Another algorithm has been developed to handle **mixed uncertainties**.



Extensions to performance analysis

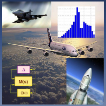
- **Modal performance**. A model perturbation is computed which brings a pole of $A + B(\Delta_R^* + \tilde{\Delta}_R)C$ on the borderline of a truncated sector.

- **H_∞ performance**. In the spirit of the μ lower bound algorithm, a two-step procedure is performed at each point ω_i of a rough frequency grid:

  1. investigate the unit ball $B(\boldsymbol{\Delta})$ by iteratively:
     - **computing the gradient of $\overline{\sigma}(\mathcal{F}_u(M(j\omega_i), \Delta))$**
     - **performing a line search to maximize this quantity** (which boils down to computing the eigenvalues of a Hamiltonian-like matrix)

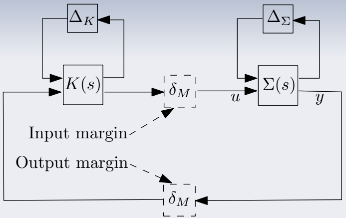
     until the problem is roughly solved at ω_i.

  2. using the value of Δ computed at step 1 as an initialization, repeatedly solve **a quadratic programming problem**, which locally maximizes $\overline{\sigma}(\mathcal{F}_u(M(j\omega), \Delta))$ with respect to both Δ and ω, until convergence.



Extension to unstructured margins

Gain, phase, modulus and time-delay margins for an uncertain system.



⇒ Input or output margins

⇒ SISO or MIMO margins

- SISO: $\delta_M = diag(I, \delta_{M,i}, I)$
- MIMO: $\delta_M = diag(\delta_{M,1}, \ldots, \delta_{M,p})$

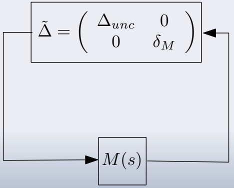
The nature of the uncertainties $\delta_{M,i}$ depends on the considered margin:

- **Gain margin**: $\delta_{M,i} = 1 + \delta_i$, $\delta_i \in \mathbb{R}$
- **Modulus margin**: $\delta_{M,i} = 1 + \delta_i$, $\delta_i \in \mathbb{C}$
- **Phase margin**: $\delta_{M,i} = e^{j\phi_i}$, $\phi_i \in \mathbb{R}$
- **Time-delay margin**: $\delta_{M,i} = e^{-\tau_i s}$, $\tau_i \in \mathbb{R}_+$

Extension to unstructured margins

Non-rational elements must be replaced by rational functions to get an LFR:

- **phase margin**: $e^{j\phi_i}$ replaced by $\frac{1-j\delta_i}{1+j\delta_i}$, $\delta_i \in \mathbb{R}$ (bilinear transformation)
- **time-delay margin**: $e^{-\tau_i s}$ replaced by a static complex function $f(\delta_i)$, $\delta_i \in \mathbb{R}$: for a given margin $\overline{\tau}_i$, the variation range of δ_i depends on ω.

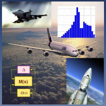


$\Delta_{unc} = diag(\Delta_K, \Delta_\Sigma)$

The computation of unstructured margins is transformed into a **skew-μ analysis problem**:

Compute the maximum value of $\overline{\sigma}(\delta_M)$ such that the LFR $\mathcal{F}_u(M(s), \tilde{\Delta})$ is robustly stable for all Δ_{unc} such that $\overline{\sigma}(\Delta_{unc}) \leq 1$.

Worst-cases and guaranteed margins are obtained with the previous algorithms.



Towards a reduced conservatism

Definition of conservatism η

Relative gap $\eta = \dfrac{x_{UB} - x_{LB}}{x_{LB}}$ between lower and upper bounds on $x = \mu$ or γ.

η can be very high, notably in presence of highly repeated real uncertainties.

Illustration: Closed-loop longitudinal flexible aircraft. 32 states, 2 real uncertainties CT and $OT \in [-1\ 1]$ (filling levels of central and outer tanks).

$$\Delta = \begin{pmatrix} CT\ I_{28} & 0 \\ 0 & OT\ I_{16} \end{pmatrix}$$

Analysis of the **robust $\mathcal{H}_\infty$ performance** between vertical wind velocity w_z and load factor N_z.

$\gamma_{UB} = 5132$, $\gamma_{LB} = 165 \Rightarrow$ conservatism $\eta > 3000\%$!!!

Towards a reduced conservatism

Idea

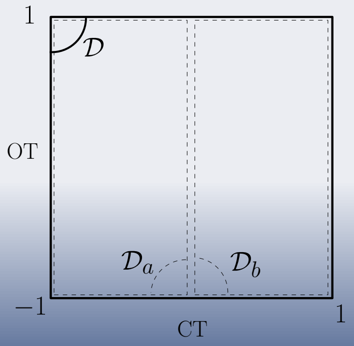
Partition the uncertainties domain $\mathcal{D}$ and perform the analysis on each subdomain.

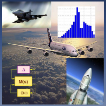
Partition $\mathcal{D}$ into $\mathcal{D}_a$ and $\mathcal{D}_b$ by cutting along the CT axis:

$$\gamma_{UB} = max\left(\gamma_{UB}^{\mathcal{D}_a}, \gamma_{UB}^{\mathcal{D}_b}\right) = 593$$

The conservatism η is now equal to 260% (instead of 3000%).

η strongly reduced by partitioning $\mathcal{D}$.



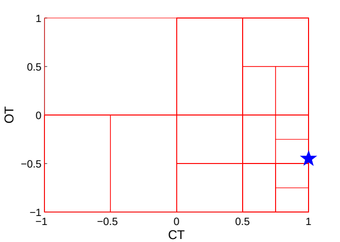


Towards a reduced conservatism

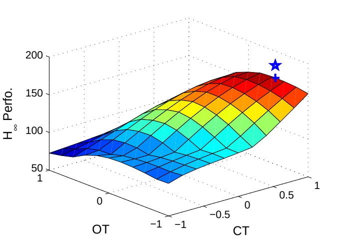
Branch and Bound algorithm

Iterate this partitioning until a *specified* conservatism η_{tol} is reached.

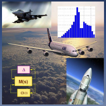
At each step, the domains $\mathcal{D}_i$ for which $\gamma_{UB}^{\mathcal{D}_i} > (1+\eta_{tol})\gamma_{LB}$ are partitioned.



Partition of $\mathcal{D}$ ($\eta_{\mathbf{tol}} = \mathbf{10\%}$)



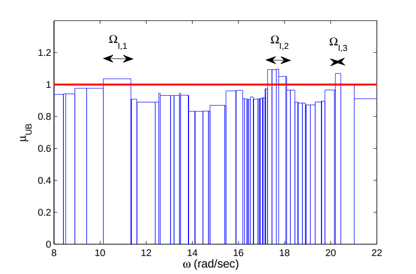
$\mathcal{H}_\infty$ performance on the parameters grid



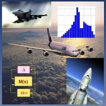
Towards a reduced conservatism

Branch and Bound algorithm: reduction of computational cost

- At step N, for the uncertainty domain $\mathcal{D}_N$, the condition $\eta \leq \eta_{tol}$ can be validated for a part Ω_V of the frequency domain Ω.
- At step $N+1$, the robustness analysis is only performed inside the frequency domain $\Omega_I = \Omega - \Omega_V$ and on a subdomain of $\mathcal{D}_N$.



$\Rightarrow$ This strategy reduces the analysis to the critical frequency intervals.



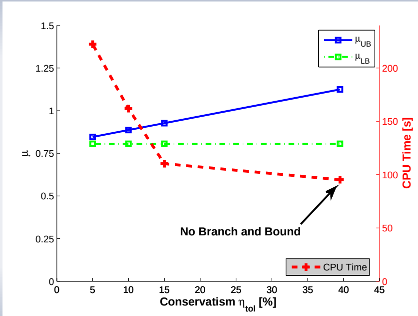
Application: robust stability analysis

Description of the model

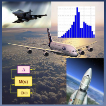
Longitudinal model of a flexible passenger aircraft:

- 22 states
- 4 parameters characterizing the aircraft mass configuration:
  - + *CT* and *OT* filling levels of the central and outer tanks
  - + *PL* embarked payload
  - + X_{CG} gravity center position

$$\Delta = diag(CT\ I_{48}, OT\ I_{28}, PL\ I_{15}, X_{CG}\ I_4)$$

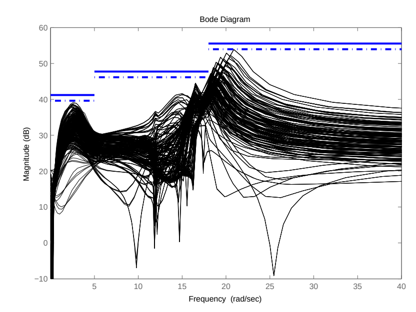


μ bounds and CPU time vs conservatism



Application: robust performance analysis

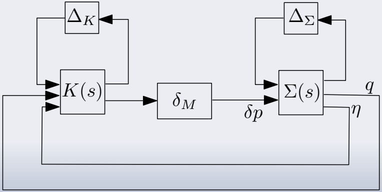
H_∞ performance from vertical wind velocity w_z to vertical load factor N_z.
Analysis performed on 3 frequency bands (looking for secondary peaks).

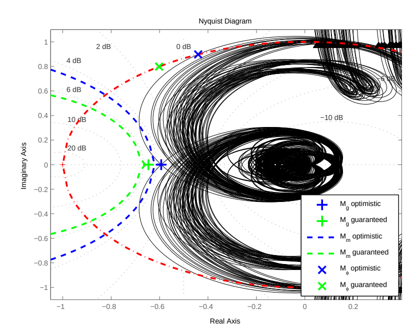


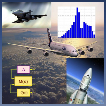
blue $\rightarrow$ robust performance bounds ($\eta_{tol} = 20\%$)
black $\rightarrow$ frequency responses on a tight parametric grid

Application: robust unstructured margins

SISO case - margins at the system input







A "μ-inspired" approach

Let $M(s)$ be a stable LTI plant. Let $\Delta = diag(\Delta_{TI}, \Delta_{TV})$ be composed of **time-invariant and arbitrarily fast time-varying** structured uncertainties.

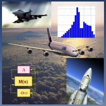
Robust stability problem: compute the maximum value k_{max} s.t. the interconnection $M(s) - \Delta$ is stable $\forall \Delta \in k_{max} B(\boldsymbol{\Delta})$.

Let $\beta > 0$. If there exist matrices $D(\omega) = diag(D_{TI}(\omega), D_{TV}) \in \mathcal{D}$ and $G(\omega) = diag(G_{TI}(\omega), G_{TV}) \in \mathcal{G}$ s.t. $\forall \omega \in \mathbb{R}_+$:

$$M^*(j\omega)D(\omega)M(j\omega) + j(G(\omega)M(j\omega) - M^*(j\omega)G(\omega)) < \beta^2 D(\omega)$$

then $k_{max} \geq 1/\beta$.

Contrary to the LTI case, it is impossible to independently solve the problem at each frequency (D_{TV} and G_{TV} must be constant $\forall \omega \in \mathbb{R}_+$).



Computing a guaranteed stability margin

First approach: frequency-domain algorithm

1. Define a coarse frequency grid $(\omega_i)_{i\in[1,N]}$ of $[\omega_{min},\omega_{max}]$.
2. **Solve a finite dimensional optimization problem** on the grid, *i.e.* minimize β s.t. $\forall i \in [1, N]$:

$$M^*(j\omega_i)D(\omega_i)M(j\omega_i) + j(G(\omega_i)M(j\omega_i) - M^*(j\omega_i)G(\omega_i)) < \beta^2 D(\omega_i)$$

3. With D_{TV} and G_{TV} being fixed, **slightly increase β and validate the result on the whole frequency range** using the same frequency elimination technique as for the μ upper bound computation.
4. If validation fails, add a worst-case frequency to the grid and go back to step 2. Otherwise, stop.

At the end, $k_{UB} = 1/\beta$ is a **guaranteed** robustness margin on $[\omega_{min},\omega_{max}]$.

Computing a guaranteed stability margin

Second approach: time-domain algorithm

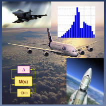
Assumption: D_{TI} and G_{TI} are **constant** on the whole frequency range.

Let $M(s) = C(sI - A)^{-1}B + D_0$. Let $\beta > 0$. If there exist matrices $D = diag(D_{TI}, D_{TV}) \in \mathcal{D}$, $G = diag(G_{TI}, G_{TV}) \in \mathcal{G}$ and $Z = R + jS$, where $R = R^*$ and $S = S^* \geq 0$, s.t.:

$$\begin{bmatrix} A^*Z + Z^*A & Z^*B - jC^*G & C^*D \\ B^*Z + jGC & -\beta^2 D + j(GD_0 - D_0^*G) & D_0^*D \\ DC & DD_0 & -D \end{bmatrix} \leq 0$$

then $k_{max} \geq 1/\beta$.

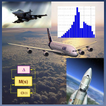
1. Solve the aforementioned LMI.
2. With D_{TV} and G_{TV} being fixed, apply the μ upper bound algorithm to compute frequency-dependent D_{TI} and G_{TI}.



IQC-based analysis

General comments

- "The" generalization of μ analysis to a (much) richer class of problems: analysis of the standard interconnection structure $M(s)$-Δ, where Δ contains neglected dynamics, uncertain/scheduling parameters (LTI or time-varying, with or without a bound on the rate of variation), delays, generic non-linearities inside a sector (with or without a restriction on its slope), specific non-linearities for which a particular IQC description is developed.
- Classical state-space LMI solution with the KYP Lemma. IQC toolbox by Kao, Megretski, Jonsonn, Rantzer.
- Untractable when the order of the augmented closed loop with multipliers is too high. Two solutions have been proposed in the literature.



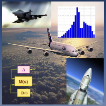
IQC-based analysis

Proposed solutions in the literature

- KYDP: a dedicated solver by Wallin, Hanson and others.
- A frequency domain cutting plane solution (Kao)
  - Use of a cutting plane technique to solve the optimization problem on a frequency grid (convex constraints are iteratively approximated by linear constraints)
  - Validation between the grid points using an Hamiltonian-like solution.

At ONERA...

- A variation is under development: LMI optimization on a frequency grid and validation between the grid points. Optimizing w.r.t. matrix variables can be much more efficient than optimizing w.r.t. scalar variables.
- OK for dealing with the complexity of the state-space representation. But what about the highly repeated parametric uncertainties ?



Time-varying Lyapunov functions

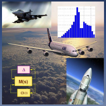
Time-varying Lyapunov functions offer a nice and flexible framework for stability and performance analysis of LPV plants. They will often outperform time-invariant functions by permitting the introduction of bounds on the rate-of-variations of the parameters. But they will also lead to much more complex conditions. We focus here on a possible way to manage with the complexity of such conditions.

Let us consider first a simple LPV closed-loop model which depends on a single parameter δ such that $\delta(t) \in \mathcal{I}$ and $\dot{\delta}(t) \in \mathcal{J}$:

$$\dot{x} = A_c(\delta(t))x \tag{1}$$

with:

$$A_c(\delta(t)) = D + \delta(t)C(I_q - \delta(t)A)^{-1}B \tag{2}$$



Time-varying Lyapunov functions

Stability of (1) for any admissible trajectory of δ is guaranteed whenever there exists a PDLF $V(x,\delta) = x'P(\delta)x$ such that:

$$\forall(\delta,\nu) \in \mathcal{I} \times \mathcal{J}\,,\ \begin{cases} P(\delta) > 0 \\ A_c(\delta)'P(\delta) + P(\delta)A_c(\delta) + \nu\frac{\partial P}{\partial \delta}(\delta) < 0 \end{cases} \tag{3}$$

Focusing on a polynomial dependance

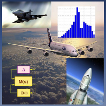
$$P(\delta) = P_0 + \delta P_1 + \ldots + \delta^r P_r \tag{4}$$

we get

$$\forall\delta \in \mathcal{I}\,,\ F(\delta, P) = \text{diag}\left(-P(\delta), \Psi(\delta,\underline{\nu}), \Psi(\delta,\overline{\nu})\right) < 0 \tag{5}$$

with

$$\Psi(\delta,\nu) = \sum_{i=0}^{r} \delta^i (A_c(\delta)'P_i + P_i A_c(\delta)) + \nu \sum_{i=1}^{r} i\delta^{i-1} P_i \tag{6}$$



Time-varying Lyapunov functions

As we did before in the frequency domain, we grid the parametric interval $\mathcal{I}$ so that the infinite set of inequalities in (5) becomes:

$$F(\delta_i, P) < 0\,,\ i = 1, \ldots, N \tag{7}$$

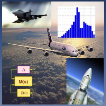
The above conditions are:

- **numerically tractable** (LMIs *w.r.t.* P_0,P_1,...,P_r).
- **non conservative** ((5) $\Rightarrow$ (7))

But, they must be tested *a posteriori* on the **continuum**. Rewriting $F(\delta, P_0, \ldots, P_r)$ as an **LFT** in δ:

$$F(\delta, P_0, \ldots, P_r) = F_{22} + \delta F_{21}(I - \delta F_{11})^{-1} F_{12} \tag{8}$$

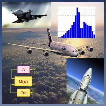
such a test – inspired by the frequency-domain approach – boils down to testing the eigenvalues of $X = F_{11} - F_{12}F_{22}^{-1}F_{21}$.



Time-varying Lyapunov functions

A first algorithm

1. Select the order r of the polynomial Lyapunov function,
2. Set $i = 1$ and define an elementary initial grid for the interval $\mathcal{I}$
$$\mathcal{G}_1(\mathcal{I}) = \{\delta_1\}\,, \text{ with } \delta_1 \in \mathcal{I}$$
3. Solve the LMI feasibility problem (7) for $\mathcal{G}_i(\mathcal{I})$,
4. If the problem is infeasible, increase r then go back to step 2 or stop the algorithm (**failure**).
5. From the spectrum of X, compute validity intervals $\{I(\delta_i)\}_{i=1\ldots N}$,
6. If $\mathcal{I} \subset \bigcup_{i=1\ldots N} I(\delta_i)$: stability proved $\rightarrow$ end (**success**).
7. Select new points $\delta_{i1}, \ldots, \delta_{iq} \notin \bigcup_{i=1\ldots N} I(\delta_i)$ and update the grid:
$$\mathcal{G}_i(\mathcal{I}) \;\rightarrow\; \mathcal{G}_{i+1}(\mathcal{I}) = \mathcal{G}_i(\mathcal{I}) \cup \{\delta_{i1}, \ldots, \delta_{iq}\}$$
Then go back to step 3.



Time-varying Lyapunov functions

Extension to several parameters

The interval $\mathcal{I}$ is replaced by a normalized hypercube $\mathcal{B} = [-1\,,\ 1]^q$ and we want to check that:

$$\forall \delta = [\delta^{[1]}, \ldots, \delta^{[q]}] \in \mathcal{B},\ F(\delta, P) < 0 \qquad (9)$$

with:

$$F(\delta, P) = F_{22} + F_{21}\Delta(I - F_{11}\Delta)^{-1}F_{12} \qquad (10)$$

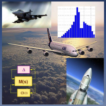
and:

$$\Delta = diag(\delta^{[1]} I_{n_1}, \ldots, \delta^{[q]} I_{n_q}) \qquad (11)$$

Since $F(\mathbf{0}, P) < 0$, conditions (9) are equivalent to :

$$\forall \Delta \in \mathcal{B}_\Delta,\ det(I - X\Delta) \neq 0 \qquad (12)$$

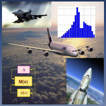
and can then be checked via standard μ tests...



Time-varying Lyapunov functions

Extended algorithm

1. Select the order r of the polynomial Lyapunov function,
2. Set $i = 1$, normalize the parameters and define an elementary initial grid for the unit hypercube $\mathcal{B}$: $\mathcal{G}_1(\mathcal{B}) = \delta_1 \in \mathcal{B}$
3. Solve the LMI feasibility problem (7) for $\mathcal{G}_i(\mathcal{B})$,
4. If the problem is infeasible, increase r then go back to step 2 or **stop**.
5. If $\bar{\mu}_\Delta(X) < 1 \rightarrow$ **end** (**success**)
6. If $\underline{\mu}_\Delta(X) \geq 1 \rightarrow$ update the grid with the calculated worst case δ_i^*: $\mathcal{G}_i(\mathcal{B}) \ \rightarrow \ \mathcal{G}_{i+1}(\mathcal{B}) = \mathcal{G}_i(\mathcal{B}) \cup \{\delta_i^*\}$ and go back to step 3.
7. If $\underline{\mu}_\Delta(X) < 1$: no conclusion can be given $\rightarrow$ split the hypercube into smaller domains and perform μ tests on each sub-domains so as to reduce the gap between upper and lower bounds. If stability cannot still be proved, then increase r and go back to step 2.



Time-varying Lyapunov functions

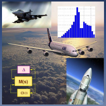
Some comments on complexity

By avoiding the KYP Lemma or its generalizations, the above algorithms offer less conservative and cheaper solutions. However, the proposed methods are not "LMI free" and there are still open issues:

- when the order of the Lyapunov function must be increased, the number of variables grows rapidly and lead to a numerically intractable LMI problem,
- when the unit ball to be cleared must be further gridded, the number of constraints in the LMI problem might become too high...
- the μ tests which are used to clear the unit ball might be conservative and time-consuming.

At ONERA we then focus on possible ways of limiting:

- the number of variables despite the possible use of high-order PDLF,
- the conservativeness of the μ tests



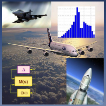
Time-varying Lyapunov functions

Application

Stability analysis of a single-axis satellite AOCS for which a **parameter varying** controller has been designed. The parameter δ is linked to the pointing error so that the controller exhibits a specific behavior according to the pointing mode (rough or fine).

Stability is required $\forall\delta \in [0\,,\,0.994]$ and $\forall\dot{\delta} \in [-0.1\,,\,0.1]$.
The first Algorithm is applied and leads after a few seconds to the following results...

PDLF order	δ	$\dot{\delta}$
0	$[0\,,\,0.47[\,\cup\,[0.47\,,\,0.994]$	0
1	$[0\,,\,0.994]$	$[-0.1\,,\,0.01]$
2	$[0\,,\,0.994]$	$[-0.1\,,\,0.1]$



Conclusions

LFT modelling and robustness analysis have received a growing attention at ONERA/DCSD over the past 10 to 15 years. Several tools, with a high maturity level, are already available:

- SMT : The skew μ Toolbox *(version 3)*,
- LFRT : The LFR Toolbox and its Simulink extension

Both packages can be downloaded from:

http://www.onera.fr/staff-en/jean-marc-biannic/

As is illustrated in this talk, current efforts are devoted to the challenging tradeoff between precision and complexity. Resulting from these efforts, new tools should soon appear in a **unified toolbox**.