On LPV Control & Aerospace Applications

Jean-Marc Biannic

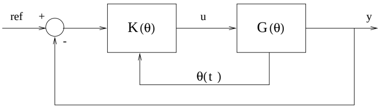
ONERA-DCSD – Toulouse, France
biannic@onera.fr
http://www.onera.fr/staff/jean-marc-biannic/
http://jm.biannic.free.fr/

International Summer School of Automatic
September 2011, Grenoble, France.

ONERA
THE FRENCH AEROSPACE LAB

Introduction

Most aerospace control applications require non stationary controllers since the characteristics of the plant exhibit significant (and possibly fast in some cases) variations over the operating domain.



Today, control engineers still make an intensive use of standard gain-scheduling techniques to solve the problem. Such approaches which consist of an interpolation of local controllers have proved useful in practice but typically suffer from a lack of theoretical guarantees.

In this context, LPV control techniques, which appeared in the early 1990's, initially looked very promising since they apparently offer an elegant path to bridge the gap between theory and practice...

Organisation of the talk

In this talk, we will shortly discuss the following :

- Standard LPV control
  - Polytopic design
  - LFT-based techniques
- Back to gain-scheduling methods
  - Static gains and control structure
  - Stability guarantees ?
- Dynamic inversion
  - Strengths and weaknesses
  - The LPV case

with aerospace illustrations and demo such as :

- missile control
- aircraft control

Outline

1. LPV control
   - LPV systems
   - Polytopic and LFT models
   - Stability & performance
   - Polytopic or LFT design ?
   - Application to missile control
2. Gain scheduling techniques
3. Dynamic inversion
4. Demo

LPV systems

A general definition

LPV systems are defined as follows :

$$\begin{cases} \dot{x} = A(\theta)x + B(\theta)u \\ y = C(\theta)x + D(\theta)u \end{cases}$$

where $\theta = \theta(t)$ is an independent parameter.

Quasi-LPV systems

If there exist matrices F and G such that :

$$F\theta = Gx$$

the above system is "quasi-LPV".

Remark

Most plants in practice are quasi-LPV.

LPV systems

LPV systems and non-linear models

The parameter-varying, nonlinear model :

$$\begin{cases} \dot{x}_1 = \theta x_1 + x_2^2 \\ \dot{x}_2 = x_1 x_2 - x_2 \end{cases}$$

may also be rewritten in a quasi-LPV form :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} \theta & x_2 \\ x_2 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Polytopic and LFT models

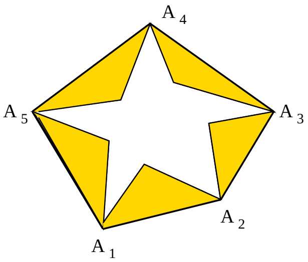
Let us consider :

$$\dot{x} = A(\theta)x \; , \quad \theta = [\theta_1, \ldots, \theta_r]^T \in \mathcal{P} \subset \mathbb{R}$$

Polytopic models

This model is defined as the convex hull of the vertices of $A(\theta)$:

$$\begin{aligned} \mathcal{C}o\{A_1, A_2, \ldots, A_N\} &\triangleq \left\{\sum_{i=1}^{i=N} \alpha_i A_i \;\; \alpha_i \geq 0 \, , \; \sum_{i=1}^{i=N} \alpha_i = 1\right\} \\ &\supset \{A(\theta) \, , \; \theta \in \mathcal{P}\} \end{aligned}$$



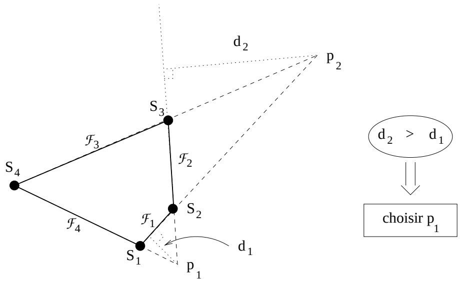
Polytopic and LFT models

"Construction" of a polytopic model

- Grid the operating domain
- Compute $A(\theta)$ for each grid point
- Compute the convex hull

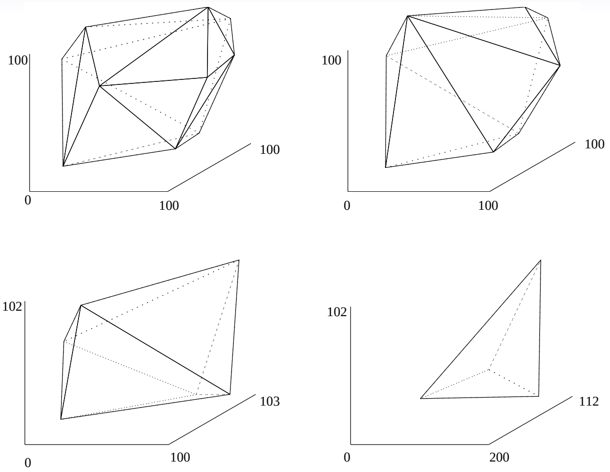
Polytope reduction

The above approach may lead to complex models with numerous vertices, the number of which can be reduced a posteriori...



Polytopic and LFT models

Illustration of polytope reduction

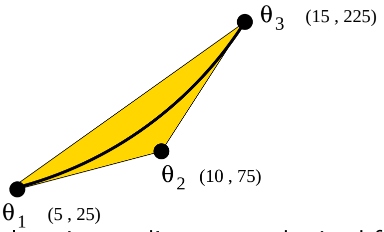


Polytopic and LFT models

From physical parameters to polytopic coordinates

Consider :

$$\dot{x} = A(v, v^2)x = (M_0 + vM_1 + v^2 M_2)x \ , \quad v \in [5, 15]$$



$$\forall v \in [5, 15],$$
$$(v \, , \; v^2) \in \mathcal{C}o\,\{\theta_1 \, , \, \theta_2 \, , \, \theta_3\}$$

The polytopic coordinates are obtained from the linear program :

$$\left\{ \begin{array}{rcrcrcl} 5\alpha_1 & + & 10\alpha_2 & + & 15\alpha_3 & = & v \\ 25\alpha_1 & + & 75\alpha_2 & + & 100\alpha_3 & = & v^2 \\ \alpha_1 & + & \alpha_2 & + & \alpha_3 & = & 1 \end{array} \right.$$

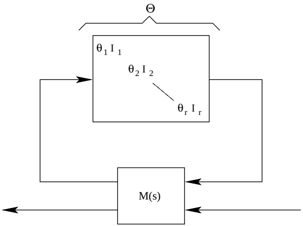
Polytopic and LFT models

LFT Model

This model gives an exact representation of LPV systems whose state-space entries rationally depends on the parameters :

$$A(\theta) = \begin{pmatrix} a_{11}(\theta) & \dots & a_{1n}(\theta) \\ \dots & \dots & \dots \\ a_{n1}(\theta) & \dots & a_{nn}(\theta) \end{pmatrix}$$

where $a_{ij}(\theta)$ is a rational function of θ.



Polytopic and LFT models

Example of LFT modeling

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \left(1 - \frac{\theta^2}{2}\right) x_1 + \theta x_2 + u \\ y = x_1 \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_1 + \theta \left(x_2 - \frac{\theta}{2} x_1\right) + u \\ y = x_1 \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_1 + w_1 + u \\ z_1 = x_2 - w_2 \\ z_2 = \frac{1}{2} x_1 \end{cases} \qquad \begin{matrix} w_1 = \theta z_1 \\ w_2 = \theta z_2 \end{matrix}$$

Polytopic and LFT models

LFT properties

- Sums, products and compositions of LFTs remain LFTs. They can be treated as objects and viewed as a rather straightforward generalization of LTI systems.
- The LFR Toolbox is essentially based on this notion of extended object (generalizing the *ss* class).

LFT reduction

For a given LPV plant, the computation of a minimal size LFT object is a very difficult, still unsolved in the general case, problem. Then, reduction techniques need to be applied a posteriori. Two kinds of approach exist :

- numerical reduction
  - step 1 : bloc reduction via genaralized Kalman approach
  - step 2 : global reduction via structured Grammians.
- symbolic reduction (tree decompositions in graphs)

Stability of LPV systems

Introduction

The LPV system defined by :

$$\dot{x} = A(\theta)x\,,\ \theta \in \mathcal{P}$$

is stable on $\mathcal{P}$ if there exists $X(\theta) > 0$, such that :

$$\forall \theta \in \mathcal{P}\,,\quad A(\theta)^T X(\theta) + X(\theta)A(\theta) + \dot{\theta}\frac{\partial X}{\partial \theta} < 0$$

This inequality ensures that $V(x,\theta) = x^T X(\theta)x$ is a decreasing function along the plant trajectories. If X is chosen independently of θ, the system is quadratically stable. We have :

$$\forall \theta \in \mathcal{P}\,,\quad A(\theta)^T X + XA(\theta) < 0$$

Stability of LPV systems

Quadratic stability of a polytopic model

$$\forall \theta \in \mathcal{P}\,, \quad A(\theta)^T X + XA(\theta) < 0\,, \quad X > 0$$

$$\Updownarrow$$

$$\forall \alpha_i \geq 0, \ \sum_{i=1}^{i=N} \alpha_i = 1, \quad \sum_{i=1}^{i=N} \alpha_i (A_i^T X + XA_i) < 0\,, \quad X > 0$$

$$\Updownarrow$$

$$\forall i = 1 \ldots N\,, \ (A_i^T X + XA_i) < 0\,, \quad X > 0$$

We get a finite number of LMIs...

Performance of LPV systems

The performance of LPV systems can be quantified via the $\mathcal{L}_2$ norm :
Definition : The $\mathcal{L}_2$ norm of $H(.)$ is given by :

$$\gamma = \sup_{u \neq 0} \frac{\|H(u)\|_2}{\|u\|_2}$$

Property : The $\mathcal{L}_2$ norm of an LPV plant is γ-bounded if there exists $X(\theta) > 0$ on $\mathcal{P}$ such that : $\forall \theta \in \mathcal{P}$, $\forall \dot{\theta} \in \mathcal{V}$:

$$\begin{pmatrix} A(\theta)^T X(\theta) + X(\theta) A(\theta) + \dot{\theta} \frac{\partial X(\theta)}{\partial \theta} & X(\theta) B(\theta) & C(\theta)^T \\ B(\theta)^T X(\theta) & -\gamma I & D(\theta)^T \\ C(\theta) & D(\theta) & -\gamma I \end{pmatrix} < 0$$

Performance of LPV systems

Remark

With a quadratic Lyapunov function, the inequality reduces to :

$$\forall\theta \in \mathcal{P}\,, \begin{pmatrix} A(\theta)^T X + XA(\theta) & XB(\theta) & C(\theta)^T \\ B(\theta)^T X & -\gamma I & D(\theta)^T \\ C(\theta) & D(\theta) & -\gamma I \end{pmatrix} < 0$$

which is a straightforward generalization of the $\mathcal{H}_\infty$ norm of LTI models.

- The above constraint characterizes the quadratic $\mathcal{H}_\infty$ performance,
- No bound on the rate-of-variations is taken into account here.

Performance of LPV systems

Quadratic $\mathcal{H}_\infty$ performance of a polytopic model

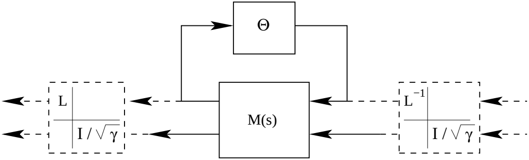
As for stability, checking the performance of a polytopic model reduces to test on the vertices of the polytope :

$$\forall i = 1, \ldots N \,, \; \begin{pmatrix} A_i^T X + X A_i & X B_i & C_i^T \\ B_i^T X & -\gamma I & D_i^T \\ C_i & D_i & -\gamma I \end{pmatrix} < 0$$

Performance of LPV systems

Performance of an LFT model

An extension of the small-gain theorem is used :



The $\mathcal{L}_2$ norm of $\mathcal{F}_u(M(s), \Theta)$ is γ-bounded if there exists an invertible matrix L commuting with Θ and such that :

$$\left\| \begin{pmatrix} L & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{pmatrix} G(s) \begin{pmatrix} L^{-1} & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{pmatrix} \right\|_\infty < 1$$

LPV synthesis

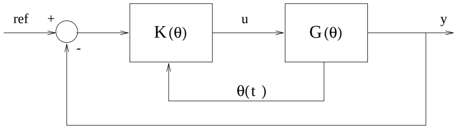
Introduction

We search for a controller, the structure of which copies that of the plant.

Plant : $G(\theta) \begin{cases} \dot{x} = A(\theta)x + B(\theta)u \\ y = C(\theta)x + D(\theta)u \end{cases}$

Controller : $K(\theta) \begin{cases} \dot{x}_K = A_K(\theta)x_K + B_K(\theta)y \\ u = C_K(\theta)x_K + D_K(\theta)y \end{cases}$

Illustration

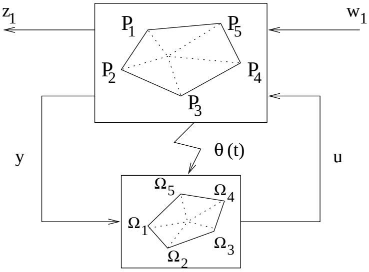


LPV synthesis

Fundamental result

Whatever the chosen model (polytopic or LFT), unlike robust control, the computation of an LPV controller boils down to a convex problem.

Polytopic case



$$P(\theta) = \sum_{i=1}^{i=N} \alpha_i(\theta)\, P_i$$

$$\Omega(\theta) = \sum_{i=1}^{i=N} \alpha_i(\theta)\, \Omega_i$$

$$\theta = \sum_{i=1}^{i=N} \alpha_i(\theta)\, \theta_i\,, \quad \alpha_i(\theta) \geq 0\,, \quad \sum_{i=1}^{i=N} \alpha_i(\theta) = 1$$

LPV synthesis

Polytopic case

Vertices of the design model : $P_i = \begin{pmatrix} A_i & B_{1i} & B_{2i} \\ C_{1i} & D_{11i} & D_{12} \\ C_2 & D_{21} & 0 \end{pmatrix}$

There exists a polytopic controller such that the quadratic $\mathcal{H}_\infty$ performance of the closed-loop is ensured iff there exist R and S such that :

$$\mathcal{N}_R^T \begin{pmatrix} A_iR + RA_i^T & B_{1i} & RC_{1i}^T \\ B_{1i}^T & -\gamma I & D_{11i}^T \\ C_{1i}R & D_{11i} & -\gamma I \end{pmatrix} \mathcal{N}_R < 0 \,, \quad i = 1, \ldots, N$$

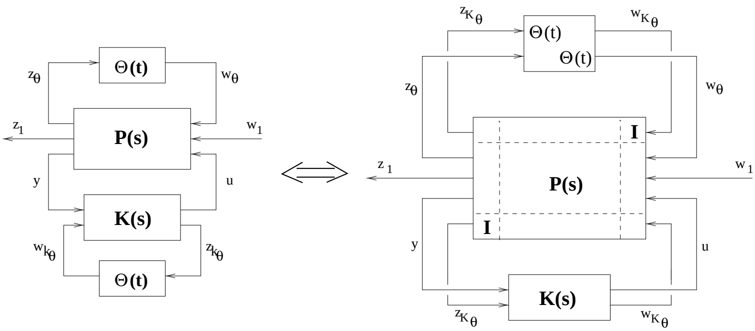
$$\mathcal{N}_S^T \begin{pmatrix} A_i^TS + SA_i & SB_{1i} & C_{1i}^T \\ B_{1i}^TS & -\gamma I & D_{11i}^T \\ C_{1i} & D_{11i} & -\gamma I \end{pmatrix} \mathcal{N}_S < 0 \,, \quad i = 1, \ldots, N$$

with :

$$\mathcal{N}_R = Ker \begin{pmatrix} B_2^T & D_{12}^T \end{pmatrix} \,, \quad \mathcal{N}_S = Ker \begin{pmatrix} C_2 & D_{21} \end{pmatrix}$$

LPV synthesis

LFT case



The repeated Θ block commutes with a class of non necessarily diagonal matrices. This fact implies convexity in the LFT case.

LPV synthesis

Methodology-oriented Extensions

- Performance criteria,
- Bounds on the rate-of-variations of parameters
- Uncertainties

Technical extensions

- Change of variables,
- LMIs rewritting via
  - introduction of new variables,
  - Relaxation techniques,
  - Linearization techniques

LPV synthesis

Bound on the rate-of-variations A rigorous approach consists of using PDLF. For example :

$$X(\theta) = X_0 + \theta_1 X_1 + \ldots + \theta_r X_r$$

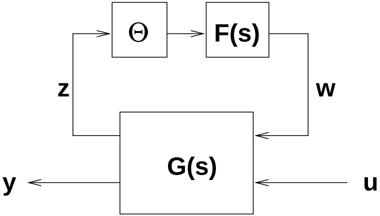
but his also introduces numerous variables...
In practice, one prefers an approximated approach which consists of modifying the design model.

LPV synthesis

Bound on the rate-of-variations

The parameter is filtered via a low-pass function :

$$F(s) = diag(f_1(s), \ldots, f_r(s)) \;, \quad f_i(s) = \frac{1}{1 + \tau_i s}$$



The modified model converges towards the initial one when :

$$\begin{cases} \tau_i \to 0 \\ \dot{\Theta}(t) \to 0 \end{cases}$$

LPV synthesis

Bound on the rate-of-variations

DESIGN PROCEDURE

1. Determination of an LFT-based design model,
2. Model transformation via parametric filtering,
3. Conversion to a polytopic model,
4. Polytopic design,
5. Controller reconstruction,
6. Stability and performance analysis on the initial model,
   - Successful analysis –> End,
   - Failure –> Reduce τ_i and back to step 2.

Application to missile control

Missile model

$$\left\{ \begin{array}{l} \begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} Z_\alpha & 1 \\ M_\alpha & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} Z_\delta \\ M_\delta \end{pmatrix} \delta \\ \eta = N_\alpha \alpha + N_\delta \delta \end{array} \right.$$

with

$$\begin{aligned} M_\alpha &= K_q M^2 (a_m \alpha^2 + b_m |\alpha| + c_m) = M_{\alpha_0} + \theta M_{\alpha_1} \\ Z_\alpha &= K_\alpha M (a_n \alpha^2 + b_n |\alpha| + c_n) \approx a_Z M_\alpha + b_Z \\ N_\alpha &= K_z M^2 (a_n \alpha^2 + b_n |\alpha| + c_n) \approx a_N M_\alpha + b_N \\ Z_\delta &= K_\alpha M \cos\alpha \, d_n \\ M_\delta &= K_q M^2 d_m \\ N_\delta &= K_z M^2 d_m \end{aligned}$$

Application to missile control

Missile model

$$\begin{cases} \begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = (A_0 + \theta A_1) \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} Z_\delta \\ M_\delta \end{pmatrix} \delta \\ \eta = (C_0 + \theta C_1) \begin{pmatrix} \alpha \\ q \end{pmatrix} + N_\delta \delta \end{cases}$$

with :

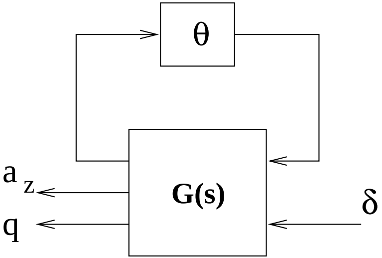
$$A_0 = \begin{pmatrix} a_Z M_{\alpha_0} + b_Z & 1 \\ M_{\alpha_0} & 0 \end{pmatrix} \quad A_1 = \begin{pmatrix} a_Z M_{\alpha_1} & 0 \\ M_{\alpha_1} & 0 \end{pmatrix}$$

$$C_0 = \begin{pmatrix} a_N M_{\alpha_0} + b_N & 0 \end{pmatrix} \quad C_1 = \begin{pmatrix} a_N M_{\alpha_1} & 0 \end{pmatrix}$$

Application to missile control

Missile model

We finally obtain the following LFT model :

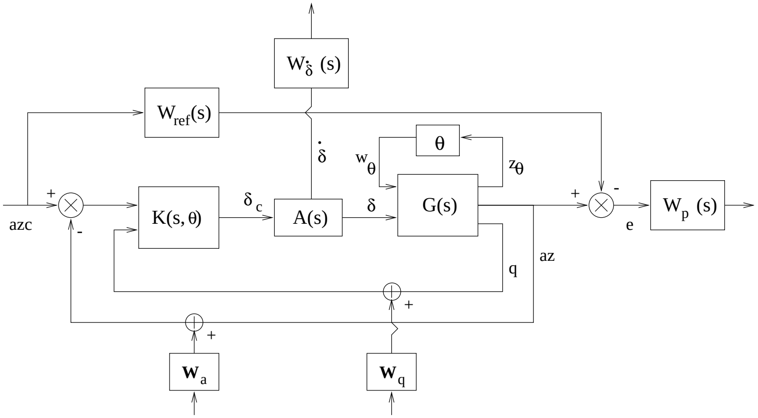


with :

$$G(s) = \left(\begin{array}{cc|cc} a_Z M_{\alpha_0} + b_Z & 1 & a_Z & Z_\delta \\ M_{\alpha_0} & 0 & 1 & M_\delta \\ \hline M_{\alpha_1} & 0 & 0 & 0 \\ a_N M_{\alpha_0} + b_N & 0 & a_N & N_\delta \\ 0 & 1 & 0 & 0 \end{array}\right)$$

Application to missile control

Nominal design model



Application to missile control

Parametric filtering

w_θ is replaced by :

$$\tilde{w}_\theta = \frac{w_\theta}{1+\tau s} = \frac{\theta z_\theta}{1+\tau s}$$

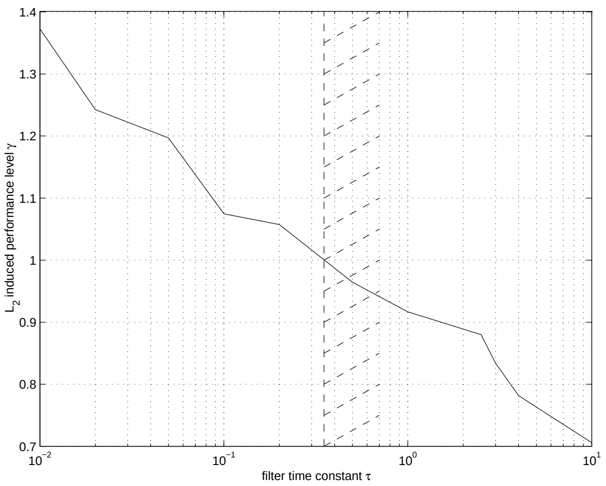
Ideally we would like :

$$\tilde{w}_\theta = \frac{\theta}{1+\tau s} z_\theta$$

To converge to this relation, the filter should be fast with respect to z_θ.
For this application there exists a relationship between z_θ and the normal acceleration for which a first-order reference model with a time constant $\tau_r = 0.35\, s$ is imposed.
Then we choose $\tau \leq 0.35$.

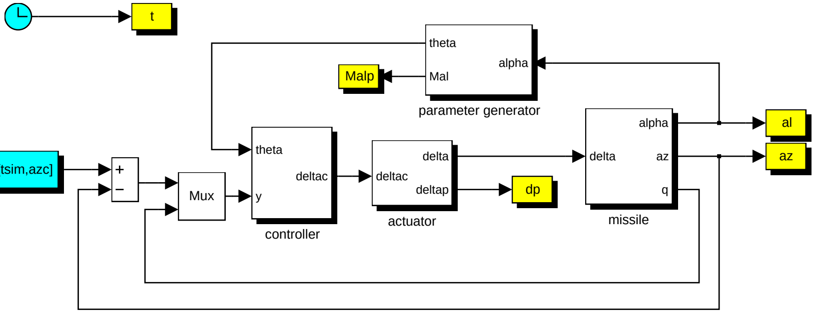
Application to missile control

Performance index as a function of τ



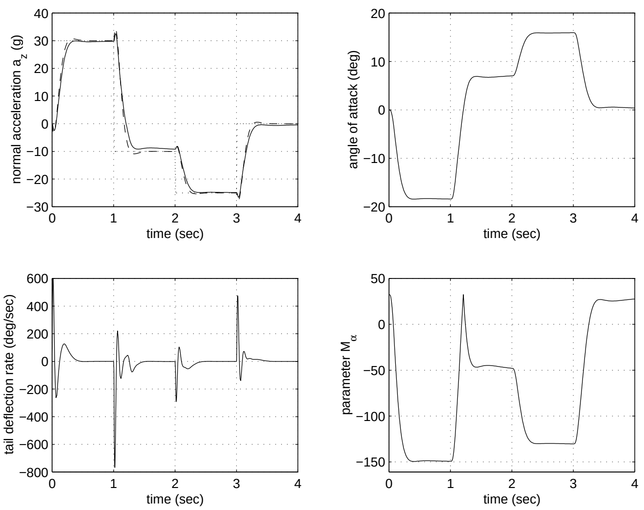
Application to missile control

Simulation model



Application to missile control

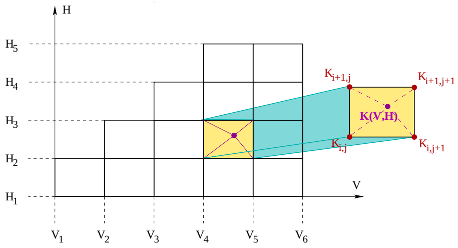
Nonlinear simulation



Outline

General principle

- Define a grid of the operating domain to be covered,
- Trim and linearize the plant about each grid point,
- Perform local design via linear techniques,
- Interpolate local controllers to get the global.



Some comments

- The above approach has been successfully applied on aerospace applications for 30 years.
- Standard grid combines slowly varying parameters (mach and altitude) with fixed ones such as the mass, the center-of-gravity location, the configuration...
- In contrast with LPV control, no stability proof can be given a priori between grid points. Critical problems may appear in case of fast varying parameters (induced dynamics),
- The success of gain-scheduling approaches is well-established thanks to :
  - a good behavior observed in practice,
  - a low conservatism when compared to LPV techniques

Recent improvements

Born in the 60's and successfully applied in the aerospace industry for years, gain scheduling techniques offer two significant advantages when compared to LPV methods :

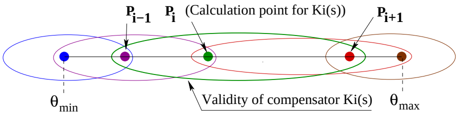
- they are numerically cheap, which makes tuning easier,
- they are not conservative.

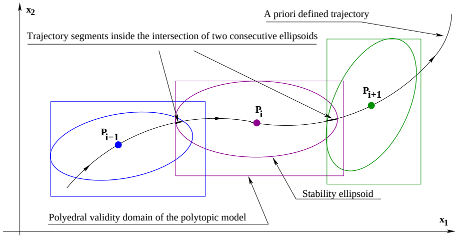
In return, the aforementioned weak points have justified recent work in the domain :

- interpolation of dynamic compensators (limited impact when the controller can be structured)
- interpolation with stability proof,
- gain-scheduling for quasi-LPV plants ("velocity-based linearization").

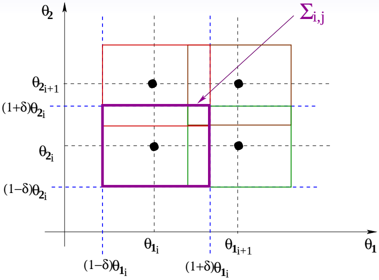
Interpolation with stability proof

Illustrations





Interpolation of locally robust controllers



- J-M. Biannic, C. Roos, A. Knauf. *Design and analysis of fighter aircraft flight control laws.* European Journal of Control, 2006.

Controllers validation

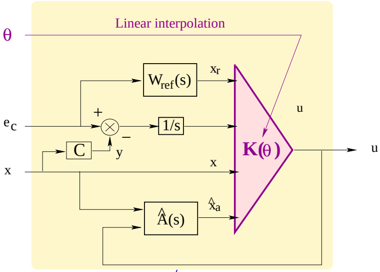
Modified LPV design (based on a polytopic approach)

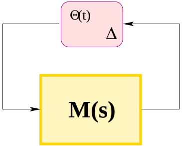
$$\begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \sum_{i=1}^{N} \alpha_i(\theta) \begin{pmatrix} A_{K_i} & B_{K_i} \\ C_{K_i} & D_{K_i} \end{pmatrix} \begin{bmatrix} x_K \\ y \end{bmatrix}$$

with $\alpha_i(\theta) = \mathcal{F}_u(T_i, \theta I_{n_i})$

+ LFT model of the plant possibly including uncertainties

Gain scheduling by interpolation of locally robust controllers





Standard form for ... robustness analysis

Outline

Introduction (1/2)

Dynamic inversion consists of "erasing" the nonlinear elements of a plant such that a robust linear technique can then be applied. For the purpose of illustration consider the nonlinear but scalar model :

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

and define the linearizing input v such that :

$$u = \frac{1}{g(x)}(v - f(x))$$

then :

$$\dot{x} = v$$

There remains to compute $K(s)$ such that x converges to x_c :

$$v = K(s)\begin{bmatrix} x_c \\ x \end{bmatrix}$$

Introduction (2/2)

In most cases, the inversion is not perfect since :

$$u = \text{sat}\left(\frac{1}{\hat{g}(x)}(v - \hat{f}(x))\right)$$

then it follows :

$$\dot{x} = (1 + \delta(t))v + w$$

The linear controller $K(s)$ must be robust against parametric uncertainties and external perturbations. Several techniques can be used. Note that a P.I. structure :

$$v = K_p(x_c - x) + \frac{K_i}{s}(x_c - x)$$

generally offer a good robustness level. Moreover, this type of controller eliminates static parts of external perturbations.

Example (1/4)

Fast longitudinal dynamics is given by :

$$\begin{cases} \dot{\alpha} = q + Z \\ J\dot{q} = W + B\delta_e \end{cases} \quad (1)$$

with :

$$\begin{cases} Z = \frac{g}{V\cos\beta}(\cos\alpha\cos\theta\cos\phi + \sin\alpha\sin\theta + \tilde{a}_z\cos\alpha - \tilde{a}_x\sin\alpha) - p_i\tan\beta \\ W = q_d SL(C_{M_\alpha}(M_a)\alpha + qLC_{M_q}/V) + (J_{zz} - J_{xx})pr + J_{xz}(r^2 - p^2) \\ B = q_d SLC_{M_\delta}(M_a) \end{cases} \quad (2)$$

and

$$\begin{cases} p_i = p\cos\alpha + r\sin\alpha \\ \tilde{a}_x = ga_{x_m} + \lambda g(q^2 + r^2) \\ \tilde{a}_z = ga_{z_m} + \lambda g(\dot{q} - pr) \end{cases} \quad (3)$$

Example (2/4)

Two nested loops

$$\begin{cases} \dot{\alpha} = q + Z \\ J\dot{q} = W + B\delta_e \end{cases} \tag{4}$$

The structure of system (4) can be divided as follows :

$$\begin{cases} \dot{\alpha} = q + Z \\ u = q \\ y = \alpha \end{cases} \quad \& \quad \begin{cases} J\dot{q} = W + B\delta_e \\ u = \delta_e \\ y = q \end{cases} \tag{5}$$

This new structure is interesting since we now have scalar models.

Example (3/4)

Outer loop : angle-of-attack (α) control :

$$\left\{ \begin{array}{l} \dot{\alpha} = q + Z \\ u = q \\ y = \alpha \end{array} \right. \tag{6}$$

If q can be controlled such that :

$$q = -Z + k_p(\alpha_c - \alpha) + k_i \int (\alpha_c - \alpha) \tag{7}$$

alors :

$$\frac{\alpha}{\alpha_c} = \frac{k_p s + k_i}{s^2 + k_p s + k_i} \tag{8}$$

With $k_p = 2\xi_c\omega_c$ et $k_i = \omega_c^2$ and filtering α_c by a first-order feedforward system (to compensate de stable zero $-K_i/k_p$) we obtain the desired response on α.

Example (4/4)

Inner loop : pitch rate (q) control

$$\begin{cases} J\dot{q} = W + B\delta_e \\ u = \delta_e \\ y = q \end{cases} \tag{9}$$

The objective is now to drive q such that the constraint (7) imposed by the outer-loop is satisfied. To achieve this, we invert (9) :

$$\delta_e = B^{-1}\left(-W + \frac{J}{\tau_c}(q_c - q)\right)$$

then :

$$\frac{q}{q_c} = \frac{1}{1 + \tau_q s}$$

where τ_q must remain smaller than ω_c to ensure time scale separation.

Strengths and weaknesses

- ☺ treats a large class of nonlinear plants
- ☺ treats the actual problem
- ☺ automatically adapts to the operating point
- ☺ easily takes varying specifications into account
- ☺ easy to tune
- ☺ *a posteriori* analysis possible in the LPV case

- ☹ unstable zeros
- ☹ saturations
- ☹ inversion errors
- ☹ state feedback $\Rightarrow$ estimation
- ☹ flexible modes
- ☹ analysis is difficult in the general case

LPV case (1/2)

Consider the LPV plant :

$$\left\{ \begin{array}{l} \dot{x} = A(\theta)x + Bu \\ z = Lx \end{array} \right. \tag{10}$$

where only A is assumed to be parameter-varying.

This system may be rewritten :

$$\left\{ \begin{array}{l} \dot{x} = A_0 x + w(x,\theta) + Bu \\ z = Lx \end{array} \right. \tag{11}$$

with :

$$w(x,\theta) = (A(\theta) - A_0)\, x \tag{12}$$

LPV case (2/2)

Using the linearizing control input :

$$u = v - B^{-1} w(x, \theta) \tag{13}$$

the initial LPV plant becomes LTI :

$$\begin{cases} \dot{x} = A_0 x + B v \\ z = L x \end{cases} \tag{14}$$

so that the problem may now be treated by a standard LTI approach :

$$v = H(s) z_c + K(s) x$$

Finally, the LPV control law is :

$$u = H(s) z_c + \left(K(s) - B^{-1}(A(\theta) - A_0)\right) x$$

The closed loop can then be written in an LPV format for which an LFT model is available. Then, several analysis techniques can be used to check stability and performance...

Outline

1. LPV control
2. Gain scheduling techniques
3. Dynamic inversion
4. Demo
   - Model description
   - Specifications
   - Dynamic-inversion based solution
   - MATLAB Demo

Model description (1/3)

We consider a model which represents the behavior of a fast aircraft on a large operating domain. The speed range is $[100\ 300]$ m/s.

$$\left\{\begin{array}{l} \dot{\alpha} = \underbrace{z_\alpha \alpha}_{w_\alpha} + q + w_{pert} \\ \dot{q} = \underbrace{m_\alpha \alpha + m_q q}_{w_q} + m_\delta \delta_m \end{array}\right. \tag{15}$$

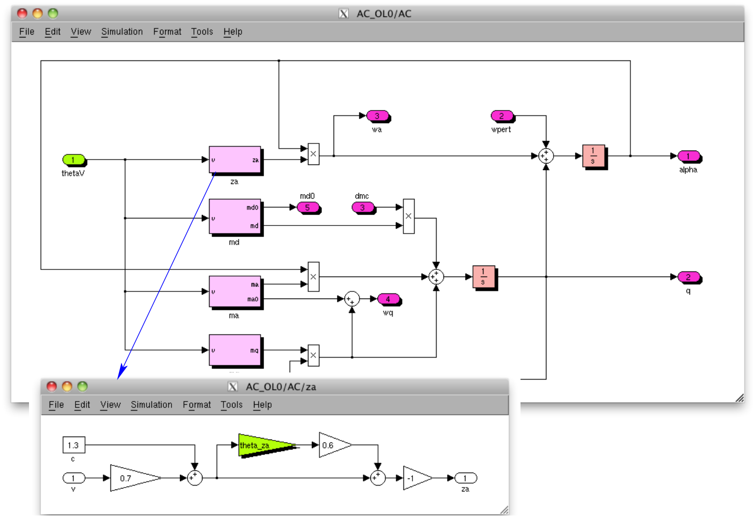
with :

$$\left\{\begin{array}{l} z_\alpha = -(1 + 0.6\theta_{z_\alpha})(1.3 + 0.7\theta_v) \\ m_\alpha = 0.8 + (\theta_{m_\alpha} + 1.5\delta_{m_\alpha})(5 + 4\theta_v) \\ m_q = -(1 + 0.35\theta_{m_q})(0.9 + 0.45\theta_v) \\ m_\delta = -40(1 + 0.25\delta_{m_\delta})(1 + 1.15\theta_v + 0.35\theta_v^2) \end{array}\right. \tag{16}$$

where θ_{z_α}, θ_{m_α}, θ_{m_q} et θ_v denote measurable parametric variations, while δ_{m_α} et δ_{m_δ} correspond to parametrric uncertainties.

Model description (2/3)

A SIMULINK view



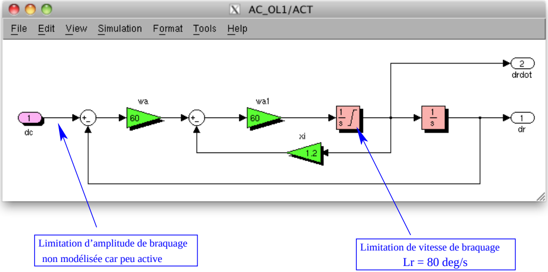
Model description (3/3)

Actuator model

The actuator is described by a second-order dynamic model with pulsation $\omega_a = 60\, rad/s$ and damping $\xi_a = 0.6$. Without saturations we have :

$$\delta_{m_r} = \frac{\omega_a^2}{s^2 + 2\xi_a s + \omega_a^2}\delta_{m_c}$$

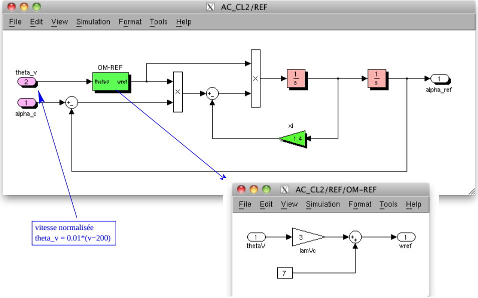
With saturations (rate limitations), we use :



Specifications

- angle-of-attack tracking with a second-order reference model with $\xi_{ref} = 0.7$ and varying pulsation ω_{ref} :
  - $V = 100m/s \Rightarrow \omega_{ref} = 4$
  - $V = 300m/s \Rightarrow \omega_{ref} = 10$
- α max $= 30\ deg$
- Avoid saturations.

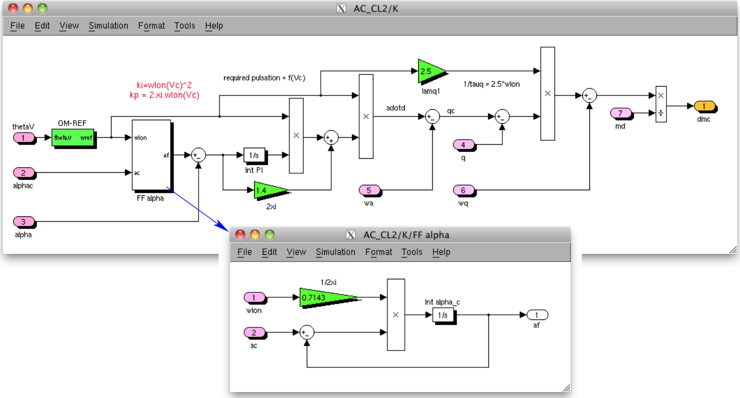
SIMULINK for the reference model



Dynamic-inversion based solution

The proposed solution is based on chapter 3. A SIMULINK implementation is given next.

SIMULINK view of the control law



MATLAB Demo

We conclude the talk by a MATLAB demo to illustrate the following :

- open-loop properties,
- implementation of a parameter-varying controller,
- closed-loop properties without saturarions,
- effects of speed variations,
- effects of saturations.

Conclusion

Despite their theoretical advantages, standard LPV techniques remain rarely used in aerospace applications because :

- they are still very conservative,
- they lead to numerically intractable problems as soon as one tries to reduce conservatism,
- in practice, they are still limited to a maximum of 2 or 3 parameters.

Further research efforts are then clearly required to solve the above issues. It is also believed that forthcoming works on :

- enhanced gain-scheduling techniques
- LPV-based dynamic inversion methods

in combination with advanced robustness analysis methods will offer nice alternatives for aerospace control applications in the future...