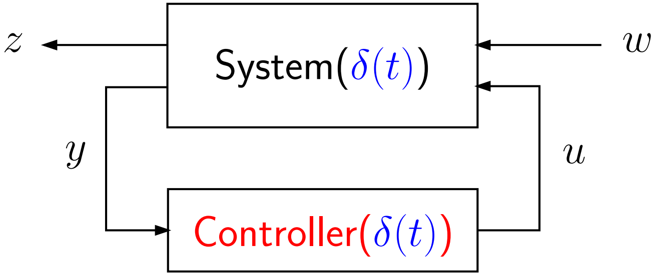
Linear Parametrically-Varying Controller Synthesis

- Direct approach: Parameter dependent Lyapunov functions
- The Polytopic Approach
- Multiplier approach for parameter-independent Lyapunov functions
- An illustrative missile example

Carsten Scherer TUDelft Siep Weiland TU/e technische universiteit eindhoven

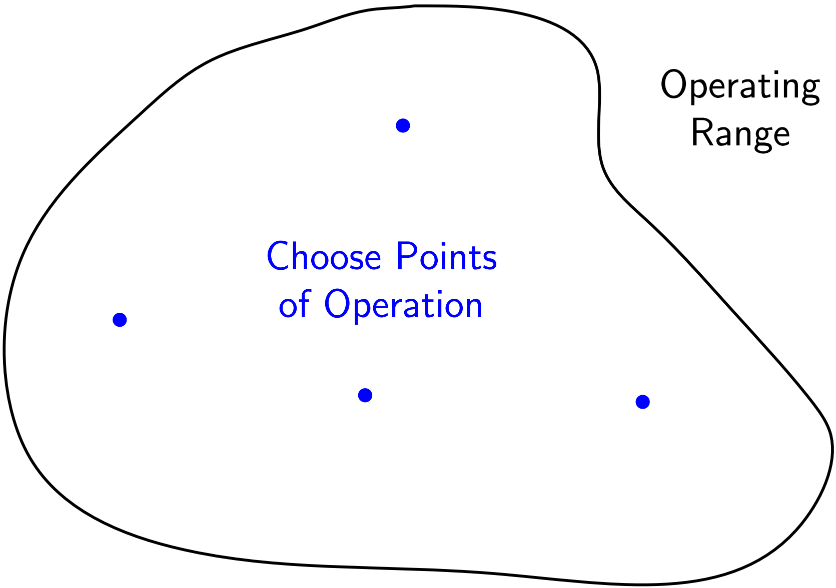
Gain-Scheduling Control for LPV Systems



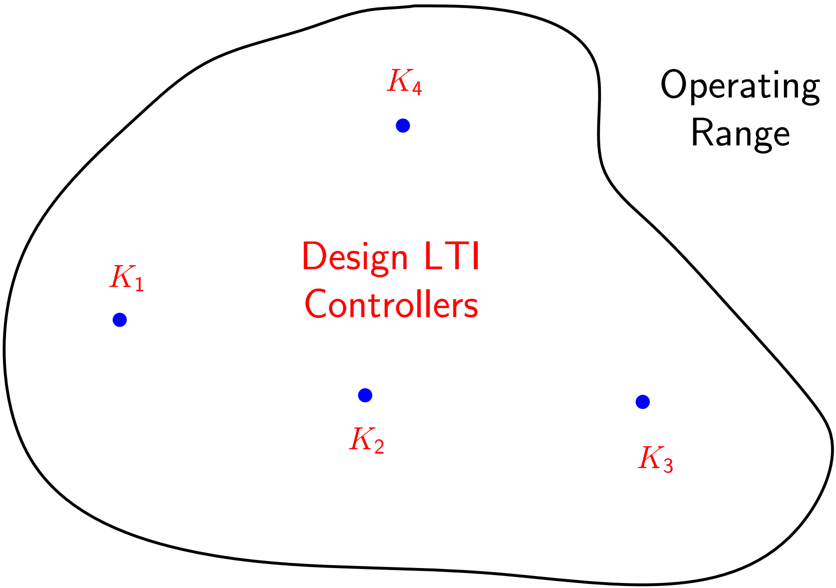
Given a parameter-dependent system, design a controller that stabilizes and achieves optimal performance, with the extra advantage (in contrast to a robust controller) that it can take on-line measurements of the parameters as information into account.

Example application: **Gain-scheduling**

Classical Gain-Scheduling

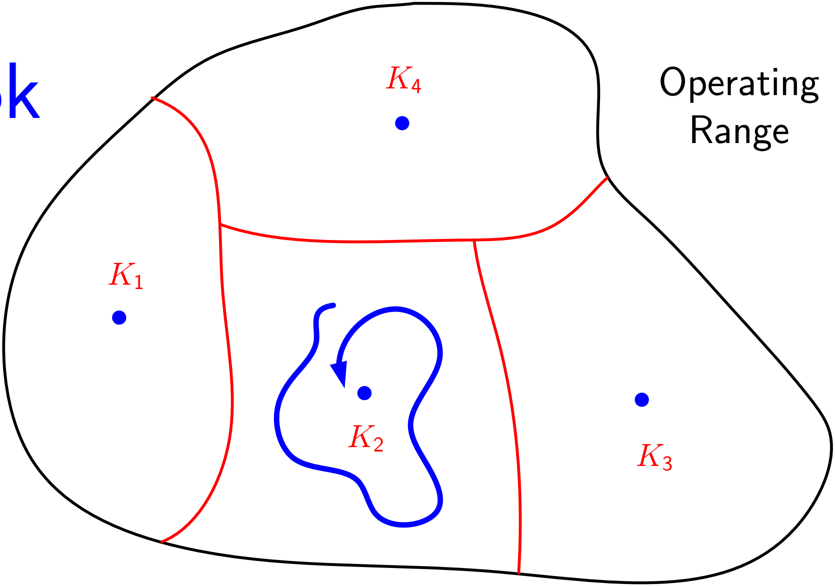


Classical Gain-Scheduling

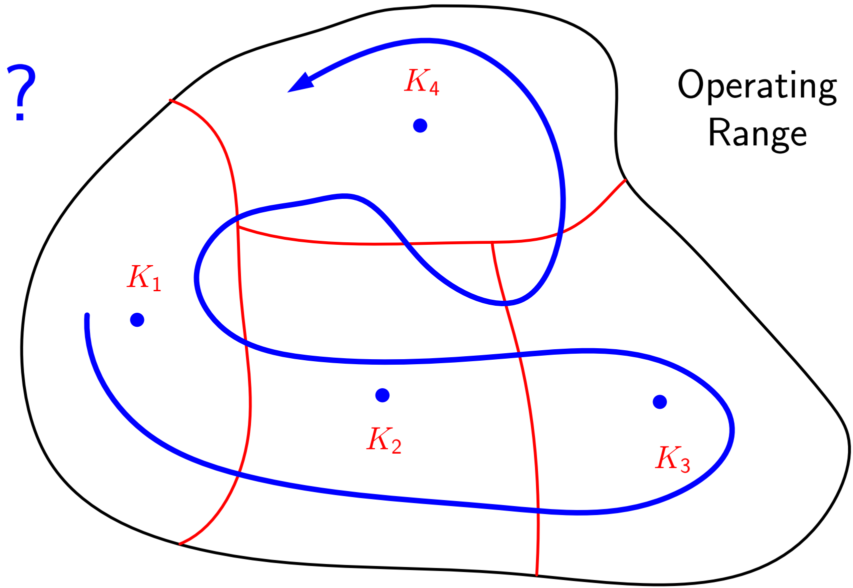


Classical Gain-Scheduling

ok



Classical Gain-Scheduling



Carsten Scherer TU Delft
Siep Weiland TU/e technische universiteit eindhoven

System Descriptions and Problem Formulation

Open-loop system:

$$\begin{pmatrix} \dot{x} \\ \hline z \\ y \end{pmatrix} = \left(\begin{array}{c|cc} A(\delta(t)) & B_1(\delta(t)) & B(\delta(t)) \\ \hline C_1(\delta(t)) & D_1(\delta(t)) & E(\delta(t)) \\ C(\delta(t)) & F(\delta(t)) & 0 \end{array}\right) \begin{pmatrix} x \\ \hline w \\ u \end{pmatrix}$$

Controller:

$$\begin{pmatrix} \dot{x}_c \\ \hline u \end{pmatrix} = \left(\begin{array}{c|c} A_c(\delta(t)) & B_c(\delta(t)) \\ \hline C_c(\delta(t)) & D_c(\delta(t)) \end{array}\right) \begin{pmatrix} x_c \\ \hline y \end{pmatrix}$$

Controlled System:

$$\begin{pmatrix} \dot{\xi} \\ \hline z \end{pmatrix} = \left(\begin{array}{c|c} \mathcal{A}(\delta(t)) & \mathcal{B}(\delta(t)) \\ \hline \mathcal{C}(\delta(t)) & \mathcal{D}(\delta(t)) \end{array}\right) \begin{pmatrix} \xi \\ \hline w \end{pmatrix}.$$

Design controller to achieve robust stability and robust performance.

Convenient Abbreviating Notation

Recall that the formulation of analysis conditions involved to assign to a continuously differentiable matrix-valued function $X(\delta)$ defined on $\boldsymbol{\delta}$ the new continuous function

$$\partial X(\delta, v) := \sum_{k=1}^{p} \partial_k X(\delta) v_k \ \text{ defined on } \ (\delta, v) \in \boldsymbol{\delta} \times \boldsymbol{v}.$$

Although this is an 'unusual' differential operator we employ the symbol ∂ for this mapping from $C^1(\boldsymbol{\delta})$ into $C(\boldsymbol{\delta} \times \boldsymbol{v})$.

Just view it as an abbreviation ...

Translation to Partial Differential Inequality

Find controller and smooth **positive** $\mathcal{X}(.)$ such that for all $\delta \in \boldsymbol{\delta}$, $v \in \boldsymbol{v}$:

$$[*]\left(\begin{array}{ccc|cc} 0 & \frac{1}{2}I & 0 & 0 & 0 \\ \frac{1}{2}I & 0 & I & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & Q_p & S_p \\ 0 & 0 & 0 & S_p^T & R_p \end{array}\right)\left(\begin{array}{cc} \partial\mathcal{X}(\delta, v) & 0 \\ I & 0 \\ \mathcal{X}(\delta)\mathcal{A}(\delta) & \mathcal{X}(\delta)\mathcal{B}(\delta) \\ \hline 0 & I \\ \mathcal{C}(\delta) & \mathcal{D}(\delta) \end{array}\right) \prec 0.$$

Recall that we still have

$$\left(\begin{array}{c|c} \mathcal{A} & \mathcal{B} \\ \hline \mathcal{C} & \mathcal{D} \end{array}\right) = \left(\begin{array}{cc|c} A + BD_cC & BC_c & B_1 + BD_cF \\ B_cC & A_c & B_cF \\ \hline C_1 + ED_cC & EC_c & D_1 + ED_cF \end{array}\right).$$

Design variables are **functions** $\mathcal{X}$, A_c, B_c, C_c, D_c of δ. Linearizing transformation still works with moderate changes only.

Change Controller Parameters

Trafo $\begin{pmatrix} \mathcal{X} & \vline & A_c & B_c & C_c & D_c \end{pmatrix} \to v = \begin{pmatrix} X & Y & \vline & K & L & M & N \end{pmatrix}$:

$$
\begin{array}{rcrcl}
\partial\mathcal{X} & \to & \boldsymbol{Z}(v) & = & \begin{pmatrix} \partial X & 0 \\ 0 & -\partial Y \end{pmatrix} \\
\mathcal{X}\mathcal{A} & \to & \boldsymbol{A}(v) & = & \begin{pmatrix} XA + LC & K \\ A + BNC & AY + BM \end{pmatrix} \\
\mathcal{X}\mathcal{B} & \to & \boldsymbol{B}(v) & = & \begin{pmatrix} XB_1 + LF \\ B_1 + BNF \end{pmatrix} \\
\mathcal{C} & \to & \boldsymbol{C}(v) & = & \begin{pmatrix} C_1 + ENC & CY + EM \end{pmatrix} \\
\mathcal{D} & \to & \boldsymbol{D}(v) & = & \begin{pmatrix} D + ENF \end{pmatrix}
\end{array}
$$

Renders blocks **affine** in new variables v which are **functions** of δ.

Apply Transformation

Finding controller and continuously differentiable $\mathcal{X}(.)$ with

$$[*]\left(\begin{array}{ccc|cc} 0 & \frac{1}{2}I & 0 & 0 & 0 \\ \frac{1}{2}I & 0 & I & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & Q_p & S_p \\ 0 & 0 & 0 & S_p^T & R_p \end{array}\right)\left(\begin{array}{cc} \partial\mathcal{X}(\delta, v) & 0 \\ I & 0 \\ \mathcal{X}(\delta)\mathcal{A}(\delta) & \mathcal{X}(\delta)\mathcal{B}(\delta) \\ \hline 0 & I \\ \mathcal{C}(\delta) & \mathcal{D}(\delta) \end{array}\right) \prec 0 \text{ on } \boldsymbol{\delta}\times\boldsymbol{v}$$

equivalent to finding $v(.)$ with

$$[*]\left(\begin{array}{ccc|cc} 0 & \frac{1}{2}I & 0 & 0 & 0 \\ \frac{1}{2}I & 0 & I & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & Q_p & S_p \\ 0 & 0 & 0 & S_p^T & R_p \end{array}\right)\left(\begin{array}{cc} \boldsymbol{Z}(v(\delta, v)) & 0 \\ I & 0 \\ \boldsymbol{A}(v(\delta)) & \boldsymbol{B}(v(\delta)) \\ \hline 0 & I \\ \boldsymbol{C}(v(\delta)) & \boldsymbol{D}(v(\delta)) \end{array}\right) \prec 0 \text{ on } \boldsymbol{\delta}\times\boldsymbol{v}.$$

If $R_p \succcurlyeq 0$ this is as usual transformed into **linear** inequality in v.

Synthesis Inequalities

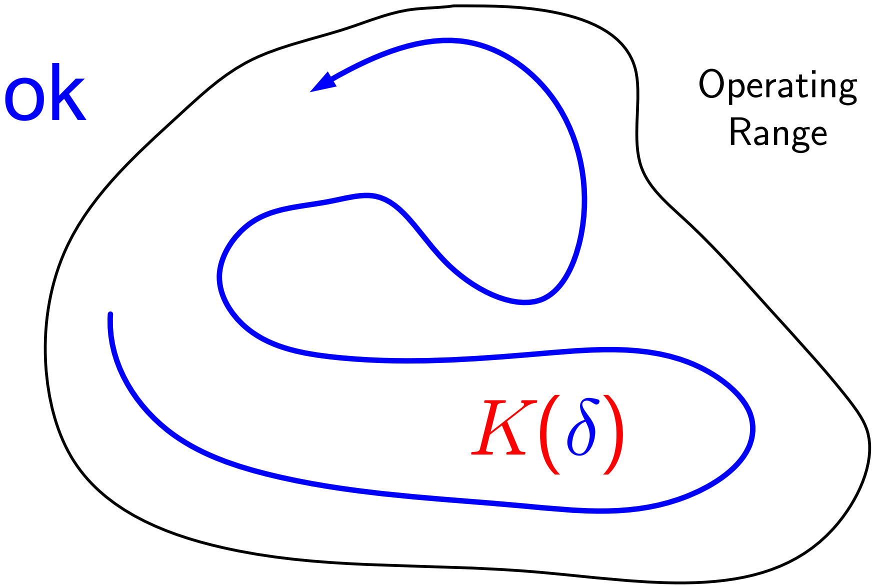
With same techniques as in analysis find function $v(\delta, v)$ such that

$$[*]\left(\begin{array}{ccc|cc} 0 & \frac{1}{2}I & 0 & 0 & 0 \\ \frac{1}{2}I & 0 & I & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & Q_p & S_p \\ 0 & 0 & 0 & S_p^T & R_p \end{array}\right)\left(\begin{array}{cc} \boldsymbol{Z}(v(\delta, v)) & 0 \\ I & 0 \\ \boldsymbol{A}(v(\delta)) & \boldsymbol{B}(v(\delta)) \\ \hline 0 & I \\ \boldsymbol{C}(v(\delta)) & \boldsymbol{D}(v(\delta)) \end{array}\right) \prec 0 \quad \text{on} \quad \boldsymbol{\delta} \times \boldsymbol{v}$$

Follow same procedure as in standard synthesis to construct controller matrices depending on parameters and satisfying analysis inequalities.

Related paper: P. Apkarian, R.J. Adams, Advanced gain-scheduling techniques for uncertain systems, IEEE Cont. Syst. Mag. 6 (1998) 21-32.

Classical Gain-Scheduling



Polytopic Approach

Open-loop system:

$$\begin{pmatrix} \dot{x}(t) \\ \hline z_p(t) \\ y(t) \end{pmatrix} = \left(\begin{array}{c|cc} A(\delta(t)) & B_1(\delta(t)) & B(\delta(t)) \\ \hline C_1(\delta(t)) & D_1(\delta(t)) & E(\delta(t)) \\ C(\delta(t)) & F(\delta(t)) & 0 \end{array}\right) \begin{pmatrix} x(t) \\ \hline w_p(t) \\ u(t) \end{pmatrix}$$

Controller:

$$\begin{pmatrix} \dot{x}_c(t) \\ \hline u(t) \end{pmatrix} = \left(\begin{array}{c|c} A_c(\delta(t)) & B_c(\delta(t)) \\ \hline C_c(\delta(t)) & D_c(\delta(t)) \end{array}\right) \begin{pmatrix} x_c(t) \\ \hline y(t) \end{pmatrix}$$

Controlled System:

$$\begin{pmatrix} \dot{\xi}(t) \\ \hline z_p(t) \end{pmatrix} = \left(\begin{array}{c|c} \mathcal{A}(\delta(t)) & \mathcal{B}(\delta(t)) \\ \hline \mathcal{C}(\delta(t)) & \mathcal{D}(\delta(t)) \end{array}\right) \begin{pmatrix} \xi(t) \\ \hline w_p(t) \end{pmatrix}.$$

Parameter trajectories satisfy $\delta(t) \in \boldsymbol{\delta} = \text{co}\{\delta^1, \ldots, \delta^N\}$.

Analysis

If there exists $\mathcal{X} \succ 0$ such that

$$[*]^T \left(\begin{array}{cc|cc} 0 & \mathcal{X} & 0 & 0 \\ \mathcal{X} & 0 & 0 & 0 \\ \hline 0 & 0 & Q_p & S_p \\ 0 & 0 & S_p^T & R_p \end{array}\right) \left(\begin{array}{cc} I & 0 \\ \mathcal{A}(\delta) & \mathcal{B}(\delta) \\ \hline 0 & I \\ \mathcal{C}(\delta) & \mathcal{D}(\delta) \end{array}\right) \prec 0 \quad \text{for all} \quad \delta \in \boldsymbol{\delta}$$

one has achieved robust quadratic performance for controlled system.

Reduces to LMIs in generators $\delta^1, \ldots, \delta^N$ if

$$\begin{pmatrix} \mathcal{A}(\delta) & \mathcal{B}(\delta) \\ \mathcal{C}(\delta) & \mathcal{D}(\delta) \end{pmatrix} \quad \text{is affine in} \quad \delta.$$

How can we guarantee that? What is the resulting design procedure?

Hypotheses on System Descriptions

Hypothesis on open-loop system:

$$\left(\begin{array}{c|cc} A(\delta) & B_1(\delta) & B \\ \hline C_1(\delta) & D_1(\delta) & E \\ C & F & 0 \end{array}\right) \quad \text{is affine in } \delta.$$

Observe independence of B, E and C, F from δ!Allows to introduce basis matrices Φ of $\ker \begin{pmatrix} B^T & E^T \end{pmatrix}$ and Ψ of $\ker \begin{pmatrix} C & F \end{pmatrix}$ as earlier.

Hypothesis on controller:

$$\begin{pmatrix} A_c(\delta) & B_c(\delta) \\ C_c(\delta) & D_c(\delta) \end{pmatrix} \quad \text{is affine in } \delta.$$

Then $\begin{pmatrix} \mathcal{A}(\delta) & \mathcal{B}(\delta) \\ \mathcal{C}(\delta) & \mathcal{D}(\delta) \end{pmatrix}$ is affine in δ.

From Analysis to Synthesis

Have achieved robust quadratic performance for controlled system if there exists $\mathcal{X} \succ 0$ such that for all $k = 1, \ldots, N$:

$$[*]^T \left(\begin{array}{cc|cc} 0 & \mathcal{X} & 0 & 0 \\ \mathcal{X} & 0 & 0 & 0 \\ \hline 0 & 0 & Q_p & S_p \\ 0 & 0 & S_p^T & R_p \end{array}\right) \left(\begin{array}{cc} I & 0 \\ \mathcal{A}(\delta^k) & \mathcal{B}(\delta^k) \\ \hline 0 & I \\ \mathcal{C}(\delta^k) & \mathcal{D}(\delta^k) \end{array}\right) \prec 0. \qquad \text{(QP)}$$

How to proceed?

- Apply general parameter transformation

  $$(\mathcal{X}, A_c(\delta^k), B_c(\delta^k), C_c(\delta^k), D_c(\delta^k)) \to (X, Y, K_k, L_k, M_k, N_k)$$

  and solve inequalities for $(X, Y, K_k, L_k, M_k, N_k)$, $k = 1, \ldots, N$.

- Eliminate controller parameters. Let us discuss this in detail.

LPV Synthesis Inequalities

After eliminating the controller parameters we arrive at the following synthesis inequalities in X and Y:

$$\begin{pmatrix} Y & I \\ I & X \end{pmatrix} \succ 0$$

$$[*]\left(\begin{array}{cc|cc} 0 & X & 0 & 0 \\ X & 0 & 0 & 0 \\ \hline 0 & 0 & Q_p & S_p \\ 0 & 0 & S_p^T & R_p \end{array}\right)\left(\begin{array}{cc} I & 0 \\ A(\delta^k) & B_1(\delta^k) \\ \hline 0 & I \\ C_1(\delta^k) & D_1(\delta^k) \end{array}\right)\Psi \prec 0, \quad k = 1, \ldots, N$$

$$[*]\left(\begin{array}{cc|cc} 0 & Y & 0 & 0 \\ Y & 0 & 0 & 0 \\ \hline 0 & 0 & \tilde{Q}_p & \tilde{S}_p \\ 0 & 0 & \tilde{S}_p^T & \tilde{R}_p \end{array}\right)\left(\begin{array}{cc} A(\delta^k)^T & C_1(\delta^k)^T \\ -I & 0 \\ \hline B_1(\delta^k)^T & D_1(\delta^k)^T \\ 0 & -I \end{array}\right)\Phi \succ 0, \quad k = 1, \ldots, N.$$

LPV Controller Construction

- Test solvability of synthesis inequalities. If feasible, compute X, Y.
- Construct $\mathcal{X}$ as in the standard synthesis procedure.
- For this storage function matrix $\mathcal{X}$ find controller parameters
$$\begin{pmatrix} A_{c,k} & B_{c,k} \\ C_{c,k} & D_{c,k} \end{pmatrix}$$
such that (QP) holds at each $k = 1, \ldots, N$.
- Let $\delta \in \boldsymbol{\delta}$. If representing as $\delta = \sum_{k=1}^{N} \lambda_k \delta^k$ with $\lambda_k \geq 0$, $\sum_{k=1}^{N} \lambda_k = 1$, the analysis inequalities on slide 12 are satisfied with
$$\begin{pmatrix} A_c & B_c \\ C_c & D_c \end{pmatrix} = \sum_{k=1}^{N} \lambda_k \begin{pmatrix} A_{c,k} & B_{c,k} \\ C_{c,k} & D_{c,k} \end{pmatrix}.$$

Comments

- For controller simulation and implementation one has to proceed as follows: At time t find convex combination coefficients in
$$\delta(t) = \sum_{k=1}^{N} \lambda_k(t)\delta^k \quad \text{and use} \quad \sum_{k=1}^{N} \lambda_k(t) \begin{pmatrix} A_{c,k} & B_{c,k} \\ C_{c,k} & D_{c,k} \end{pmatrix}$$
to define the dynamics of the time-varying controller.

- Determination of $\lambda_k(t)$ requires the solution of an LP. In order to enforce uniqueness (e.g. to assure continuity in time) one could enforce in addition that e.g. $\sum_{k=1}^{N} \lambda_k(t)^2$ is minimized (why?).

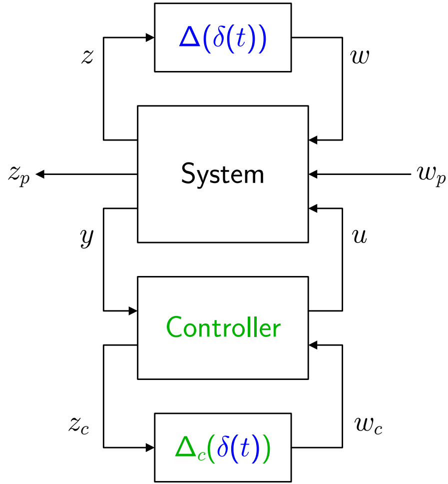
- This (simplest) procedure for designing LPV controllers is implemented in the Robust Control Toolbox in Matlab.

Configuration for Multiplier LPV Synthesis

Design parameter-dependent controller guaranteeing

- exponential stability
- quadratic performance

If some parameters coincide with state-components procedure leads to nonlinear controller!



Again concentrate on **stability** - no performance channel.

Carsten Scherer TU Delft
Siep Weiland TU/e technische universiteit eindhoven

System Descriptions

Uncontrolled LTI part:

$$\begin{aligned} \dot{x} &= Ax + B_1 w + Bu \\ z &= C_1 x + D_1 w + Eu \\ y &= Cx + Fw \end{aligned}$$

w: uncertainty input

z: uncertainty output

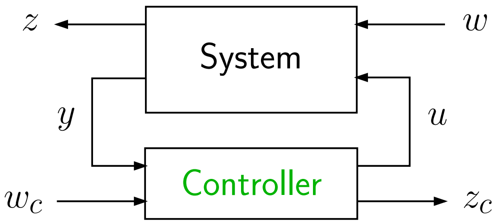
u: control input

y: measured output

Controller LTI part:

$$\dot{x}_c = A_c x_c + B_c \begin{pmatrix} y \\ w_c \end{pmatrix}$$

$$\begin{pmatrix} u \\ z_c \end{pmatrix} = C_c x_c + D_c \begin{pmatrix} y \\ w_c \end{pmatrix}$$



Fundamental Trick to Solve LPV Problem

The interconnection can be seen to result from

$$\begin{pmatrix} \dot{x} \\ \hline z \\ z_c \\ \hline y \\ w_c \end{pmatrix} = \left(\begin{array}{c|cc|cc} A & B_1 & 0 & B_2 & 0 \\ \hline C_1 & D_1 & 0 & D_{12} & 0 \\ 0 & 0 & 0 & 0 & I \\ \hline C_2 & D_{21} & 0 & D_2 & 0 \\ 0 & 0 & I & 0 & 0 \end{array}\right) \begin{pmatrix} x \\ \hline w \\ w_c \\ \hline u \\ z_c \end{pmatrix}$$

controlled with

$$\dot{x}_c = A_c x_c + B_c \begin{pmatrix} y \\ w_c \end{pmatrix}, \quad \begin{pmatrix} u \\ z_c \end{pmatrix} = C_c x_c + D_c \begin{pmatrix} y \\ w_c \end{pmatrix}$$

Can solve the **robust control problem** for this interconnection.

LPV Synthesis: Step I

Synthesis inequalities **without non-convex coupling**: $(k = 1, \ldots, N)$:

$$\begin{pmatrix} Y & I \\ I & X \end{pmatrix} \succ 0, \quad Q \prec 0, \quad \tilde{R} \succ 0$$

$$[*] \left(\begin{array}{cc|cc} 0 & X & 0 & 0 \\ X & 0 & 0 & 0 \\ \hline 0 & 0 & Q & S \\ 0 & 0 & S^T & R \end{array}\right) \left(\begin{array}{cc} I & 0 \\ A & B_1 \\ \hline 0 & I \\ C_1 & D_1 \end{array}\right) \Psi \prec 0, \quad [*] \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} \Delta(\delta^k) \\ I \end{pmatrix} \succ 0$$

$$[*] \left(\begin{array}{cc|cc} 0 & Y & 0 & 0 \\ Y & 0 & 0 & 0 \\ \hline 0 & 0 & \tilde{Q} & \tilde{S} \\ 0 & 0 & \tilde{S}^T & \tilde{R} \end{array}\right) \left(\begin{array}{cc} A^T & C_1^T \\ -I & 0 \\ \hline B_1^T & D_1^T \\ 0 & -I \end{array}\right) \Phi \succ 0, \quad [*] \begin{pmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^T & \tilde{R} \end{pmatrix} \begin{pmatrix} -I \\ \Delta(\delta^k)^T \end{pmatrix} \prec 0$$

LPV Synthesis: Step II

Find extension

$$\begin{pmatrix} Q_e & S_e \\ S_e^T & R_e \end{pmatrix} = \left(\begin{array}{cc|cc} Q & * & S & * \\ * & * & * & * \\ \hline S^T & * & R & * \\ * & * & * & * \end{array}\right) \quad \text{with inverse} \quad \left(\begin{array}{cc|cc} \tilde{Q} & * & \tilde{S} & * \\ * & * & * & * \\ \hline \tilde{S}^T & * & \tilde{R} & * \\ * & * & * & * \end{array}\right)$$

such that

$$\begin{pmatrix} Q & * \\ * & * \end{pmatrix} \prec 0 \quad \text{and} \quad \begin{pmatrix} \tilde{R} & * \\ * & * \end{pmatrix} \succ 0.$$

This is **always** possible.

Size of extension determines size of scheduling function $\Delta_c(.)$.

LPV Synthesis: Step III

Find scheduling function $\Delta_c(\delta)$ such that for all $\delta \in \boldsymbol{\delta}$

$$\left(\begin{array}{cc} \Delta(\delta) & 0 \\ 0 & \Delta_c(\delta) \\ \hline I & 0 \\ 0 & I \end{array}\right)^T \left(\begin{array}{cc|cc} Q & * & S & * \\ * & * & * & * \\ \hline S^T & * & R & * \\ * & * & * & * \end{array}\right) \left(\begin{array}{cc} \Delta(\delta) & 0 \\ 0 & \Delta_c(\delta) \\ \hline I & 0 \\ 0 & I \end{array}\right) \succ 0$$

or equivalently

$$\left(\begin{array}{cc} -I & 0 \\ 0 & -I \\ \hline \Delta(\delta)^T & 0 \\ 0 & \Delta_c(\delta)^T \end{array}\right)^T \left(\begin{array}{cc|cc} \tilde{Q} & * & \tilde{S} & * \\ * & * & * & * \\ \hline \tilde{S}^T & * & \tilde{R} & * \\ * & * & * & * \end{array}\right) \left(\begin{array}{cc} -I & 0 \\ 0 & -I \\ \hline \Delta(\delta)^T & 0 \\ 0 & \Delta_c(\delta)^T \end{array}\right) \prec 0.$$

Have explicit formula for $\Delta_c(\delta)$.

LPV Synthesis: Step IV

Design LTI-part of controller

$$\dot{x}_c = A_c x_c + B_c \begin{pmatrix} y \\ w_c \end{pmatrix}, \quad \begin{pmatrix} u \\ z_c \end{pmatrix} = C_c x_c + D_c \begin{pmatrix} y \\ w_c \end{pmatrix}$$

as quadratic performance controller for extended system

$$\left(\begin{array}{c} \dot{x} \\ \hline z \\ z_c \\ \hline y \\ w_c \end{array}\right) = \left(\begin{array}{c|cc|cc} A & B_1 & 0 & B_2 & 0 \\ \hline C_1 & D_1 & 0 & D_{12} & 0 \\ 0 & 0 & 0 & 0 & I \\ \hline C_2 & D_{21} & 0 & D_2 & 0 \\ 0 & 0 & I & 0 & 0 \end{array}\right) \left(\begin{array}{c} x \\ \hline w \\ w_c \\ \hline u \\ z_c \end{array}\right)$$

with performance index $\begin{pmatrix} Q_e & S_e \\ S_e^T & R_e \end{pmatrix}$.

Some Remarks on Sketched Procedure

- Obtained solution of LPV problem with full block scalings.
  Can further enlarge scaling set. Requires behavioral controller [1].

- Comparison to direct implemented version in LMI-toolbox:

  - Allows rational parameter dependence

  - matrices $B(\delta)$, $E(\delta)$, $C(\delta)$, $F(\delta)$ allowed to depend on parameters

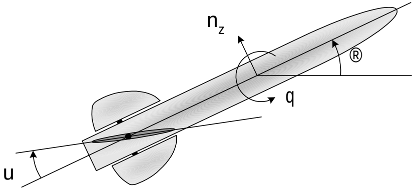
  - Care-free scheduling without determination of $\lambda_k(t)$.

- Exists extension to parameter dependent Lyapunov functions.

---

[1] C.W. Scherer, LPV control with full block multipliers, *Automatica* **37** (2001) 361-375.

High-Performance Aircraft System



u: Control input

α: Measurable parameter

n_z: Tracked output

Nonlinear system description with aerodynamic effects:

$$\begin{aligned}
\dot{\alpha} &= KM\left[\left(a_n\alpha^2+b_n\alpha+c_n(2-M/3)\right)\alpha+d_nu\right]+q \\
\dot{q} &= M^2\left[\left(a_m\alpha^2+b_m\alpha-c_m(7-8M/3)\right)\alpha+d_mu\right] \\
n_z &= M^2\left[\left(a_n\alpha^2+b_n\alpha+c_n(2-M/3)\right)\alpha+d_nu\right]
\end{aligned}$$

Carsten Scherer TU Delft Siep Weiland TU/e technische universiteit eindhoven

Main Idea

Rewrite as linear parameter-varying system

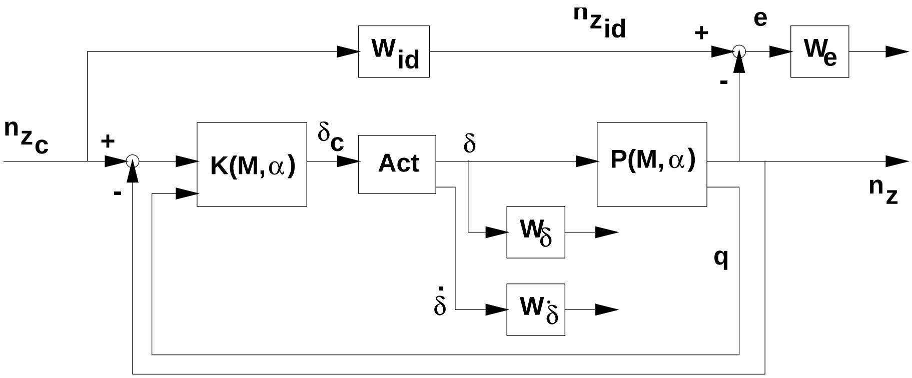
$$\begin{aligned}
\dot{\alpha} &= K\delta_1\left[\left(a_n{\delta_2}^2+b_n\delta_2+c_n(2-\delta_1/3)\right)\alpha+d_nu\right]+q \\
\dot{q} &= {\delta_1}^2\left[\left(a_m{\delta_2}^2+b_m\delta_2-c_m(7-8\delta_1/3)\right)\alpha+d_mu\right] \\
n_z &= {\delta_1}^2\left[\left(a_n{\delta_2}^2+b_n\delta_2+c_n(2-\delta_1/3)\right)\alpha+d_nu\right]
\end{aligned}$$

with bounds $2 \leq \delta_1(t) \leq 4$ and $-20 \leq \delta_2(t) \leq 20$.

Design good controller scheduled with $\delta_1(t)$, $\delta_2(t)$

$\rightarrow$ Is good controller for nonlinear system

Interconnection Structure



Model-Matching

Let controlled system approximately act like ideal model $\mathbf{W_{id}}$.

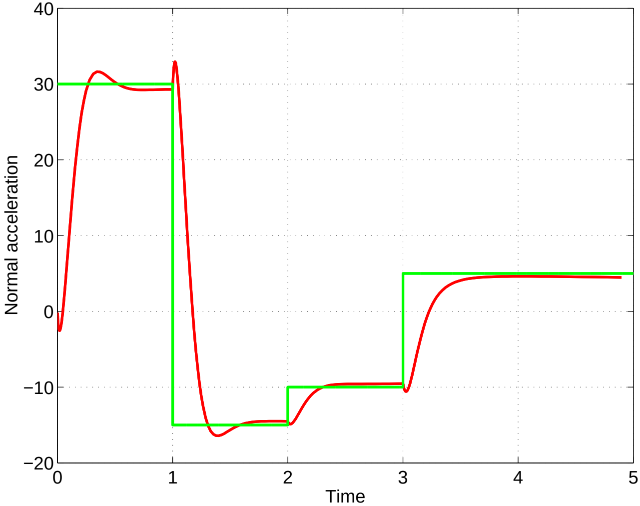
Synthesis with Convex Hull Relaxation

$M(t)$ decreases in 5 seconds from 4 to 2.

Normal acceleration

Reference

Response



Some Book References

[1] S.P. Boyd, G.H. Barratt, Linear Controller Design - Limits of Performance, Prentice-Hall, Englewood Cliffs, New Jersey (1991).

[2] S.P. Boyd, L. El Ghaoui et al., Linear matrix inequalities in system and control theory, Philadelphia, SIAM (1994).

[3] L. El Ghaoui, S.I. Niculescu, Eds., Advances in Linear Matrix Inequality Methods in Control, Philadelphia, SIAM (2000).

[4] A. Ben-Tal, A. Nemirovski, Lectures on Modern Convex Optimization. Philadelphia, SIAM Publications (2001).

[5] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge (2004).

[6] Manuals for "LMI Control Toolbox" and "μ Analysis and Synthesis Toolbox"

[7] C.W. Scherer, S. Weiland, DISC Lecture Notes "LMI's in Control".