Global Optimization based on Contractor Programming: Application to H_∞ synthesis

Jordan Ninin

with Dominique Monnet and Benoit Clement

LAB-STICC / ENSTA-Bretagne
Brest, France

Prague, June 2015

Introduction

Observations:

- In Automatic, Robotic, Electronic or Mechanic, engineers know very well their problems.
  $\Longrightarrow$ Physical Sense
- In Optimization, the specification of each solver need to classify a model: LP, NLP, MINLP, SDP, DFO,...
  If the model cannot be classify: Modification, Adaptation, Reformulation, ...
  $\Longrightarrow$ Numerical Sense

Introduction

Observations:

- In Automatic, Robotic, Electronic or Mechanic, engineers know very well their problems.
  $\Longrightarrow$ Physical Sense
- In Optimization, the specification of each solver need to classify a model: LP, NLP, MINLP, SDP, DFO,...
  If the model cannot be classify: Modification, Adaptation, Reformulation, ...
  $\Longrightarrow$ Numerical Sense

Physical Solutions $\Longleftrightarrow$ Numerical Solutions

Introduction

Observations:

- In Automatic, Robotic, Electronic or Mechanic, engineers know very well their problems.
  $\Longrightarrow$ Physical Sense
- In Optimization, the specification of each solver need to classify a model: LP, NLP, MINLP, SDP, DFO,...
  If the model cannot be classify: Modification, Adaptation, Reformulation, ...
  $\Longrightarrow$ Numerical Sense

Physical Solutions $\Longleftrightarrow$ Numerical Solutions

$\Longrightarrow$ Goal: to Propose advanced optimization tools to construct the best solver for their own problems.

Contents

Definition: Contractor

Let $\mathbb{K} \subseteq \mathbb{R}^n$ be a "feasible" region.

The operator $\mathcal{C}_{\mathbb{K}} : \mathbb{IR}^n \to \mathbb{IR}^n$ is a contractor for $\mathbb{K}$ if:

$$\forall \mathbf{x} \in \mathbb{IR}^n, \begin{cases} \mathcal{C}_{\mathbb{K}}(\mathbf{x}) \subseteq \mathbf{x}, & \text{(contractance)} \\ \mathcal{C}_{\mathbb{K}}(\mathbf{x}) \cap \mathbb{K} = \mathbf{x} \cap \mathbb{K}. & \text{(completeness)} \end{cases}$$

Example: *Forward-Backward Algorithm*

The operator $\mathcal{C} : \mathbb{IR}^n \to \mathbb{IR}^n$ is a contractor for the equation $f(x) = 0$, if:

$$\forall \mathbf{x} \in \mathbb{IR}^n, \begin{cases} \mathcal{C}(\mathbf{x}) \subseteq \mathbf{x}, \\ x \in \mathbf{x} \text{ and } f(x) = 0 \Rightarrow x \in \mathcal{C}(\mathbf{x}). \end{cases}$$

General Design

$(\tilde{x}, \tilde{f}) =$ **OptimCtc** $([\mathbf{x}], \mathcal{C}_{out}, \mathcal{C}_{in}, f_{cost})$:

- ⋆ Merging of a Branch&Bound Algorithm based on Interval Analysis (spacialB&B) and a Set Inversion Via Interval Analysis (SIVIA).
- ⋆ $\mathcal{C}_{out}, \mathcal{C}_{in}$: contractors designed by the user based on $\mathbb{K}$ and $\overline{\mathbb{K}}$,
- ⋆ $\mathcal{C}_f$: a FwdBwd contractor based on $\{x : f_{cost}(x) \leq \tilde{f}\}$
- ⋆ $\mathcal{B}$: Largest first, smear evaluation, homemade,...

General Design

$(\tilde{x}, \tilde{f}) =$ **OptimCtc** $([\mathbf{x}], \mathcal{C}_{out}, \mathcal{C}_{in}, f_{cost})$:

- ⋆ Merging of a Branch&Bound Algorithm based on Interval Analysis (spacialB&B) and a Set Inversion Via Interval Analysis (SIVIA).
- ⋆ $\mathcal{C}_{out}, \mathcal{C}_{in}$: contractors designed by the user based on $\mathbb{K}$ and $\overline{\mathbb{K}}$,
- ⋆ $\mathcal{C}_f$: a FwdBwd contractor based on $\{x \,:\, f_{cost}(x) \leq \tilde{f}\}$
- ⋆ $\mathcal{B}$: Largest first, smear evaluation, homemade,...

General Design

$(\tilde{x}, \tilde{f})$ = **OptimCtc** ($[\mathbf{x}]$, $\mathcal{C}_{out}$, $\mathcal{C}_{in}$, f_{cost}):

- ⋆ Merging of a Branch&Bound Algorithm based on Interval Analysis (spacialB&B) and a Set Inversion Via Interval Analysis (SIVIA).
- ⋆ $\mathcal{C}_{out}$, $\mathcal{C}_{in}$: contractors designed by the user based on $\mathbb{K}$ and $\overline{\mathbb{K}}$,
- ⋆ $\mathcal{C}_f$: a FwdBwd contractor based on $\{x : f_{cost}(x) \leq \tilde{f}\}$
- ⋆ $\mathcal{B}$: Largest first, smear evaluation, homemade,...

Illustration: $\mathcal{C}_{in}$, $\mathcal{C}_{out}$

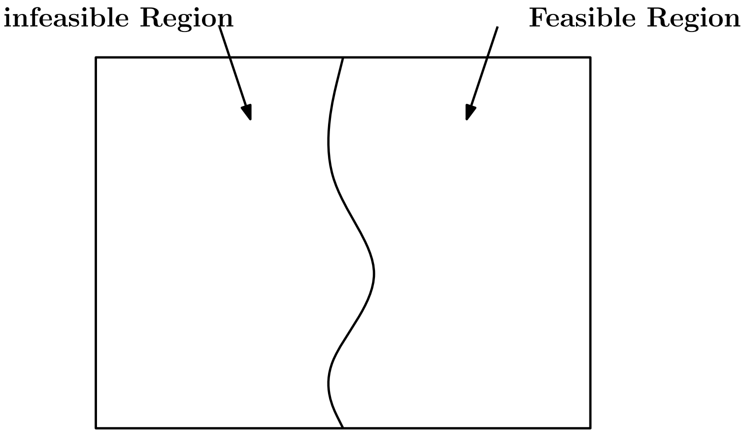


Illustration: $\mathcal{C}_{in}$, $\mathcal{C}_{out}$

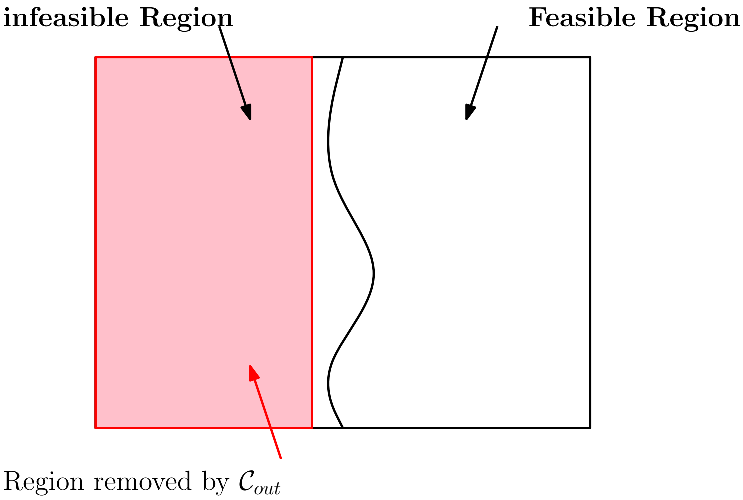
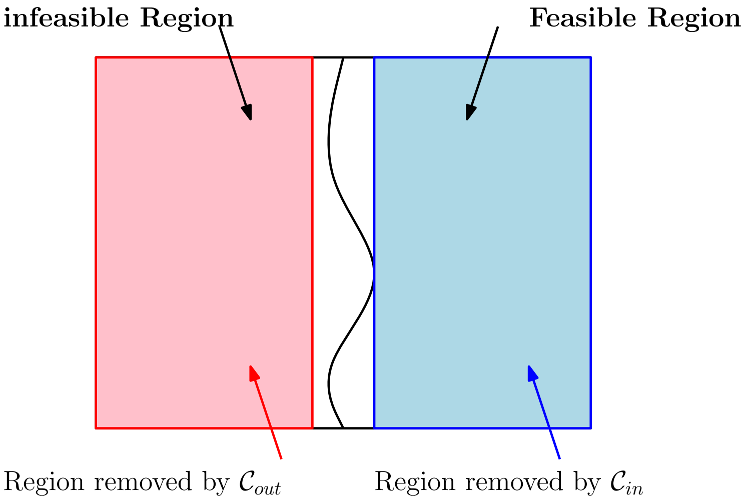


Illustration: $\mathcal{C}_{in}$, $\mathcal{C}_{out}$



The feasability test

Without equation or system,

How to prove that a point is a feasible point?

The feasability test

Without equation or system,

How to prove that a point is a feasible point?

Prove that $x \in \mathbb{K}$ $\not\Leftrightarrow$ Prove that $x \notin \overline{\mathbb{K}}$

The feasability test

Without equation or system,
How to prove that a point is a feasible point?

Prove that $x \in \mathbb{K}$ $\not\Leftrightarrow$ Prove that $x \notin \overline{\mathbb{K}}$

x is contracted by $\mathcal{C}_{in}$ $\Leftrightarrow$ $x \in \mathbb{K}$ $\Leftrightarrow$ $\mathcal{C}_{out}$ proves that x is in $\mathbb{K}$.

$\mathcal{C}_{in}$ will eliminate all the part of a box which **are not** in $\overline{\mathbb{K}}$.
$\mathcal{C}_{out}$ will eliminate all the part of a box which **are not** in $\mathbb{K}$.

Global Optimization based on Contractor

- $\mathcal{L} := \{(\mathbf{x}, \mathit{false})\}$, The boolean indicate if $\mathbf{x}$ is entirely feasible
- Do
  1. Extract from $\mathcal{L}$ a element $(\mathbf{z}, b)$,
  2. Bisect $\mathbf{z}$ following a bisector $\mathcal{B}$: $(\mathbf{z}_1, \mathbf{z}_2)$
  3. for $j = 1$ to 2 :
     - if $b = \mathit{false}$ *(i.e. $\mathbf{x}$ is not completly feasible)* then
       Contract the infeasible region using $\mathcal{C}_{out}$ and $\mathcal{C}_f$,
       Extract $\mathbf{z}_{feas}$ a feasible part of $\mathbf{z}_j$ using $\mathcal{C}_{in}$,
       Insert $(\mathbf{z}_{feas}, \mathit{true})$ in $\mathcal{L}$.
       Insert the rest $(\mathbf{z}_j, \mathit{false})$ in $\mathcal{L}$.
     - else *(i.e. $\mathbf{x}$ is entirely feasible)*
       Contract $\mathbf{z}_j$ using $\mathcal{C}_f$,
       Try to **find** a local optimum without constraint in $[\mathbf{z}_j]$,
       if *succeed* then Update $\tilde{f}$ insert $(\mathbf{z}_j, \mathit{true})$ in $\mathcal{L}$.
- stopping criterion

Global Optimization based on Contractor

- $\mathcal{L} := \{(\mathbf{x}, \mathit{false})\}$, The boolean indicate if $\mathbf{x}$ is entirely feasible
- Do
  1. Extract from $\mathcal{L}$ a element $(\mathbf{z}, b)$,
  2. Bisect $\mathbf{z}$ following a bisector $\mathcal{B}$: $(\mathbf{z}_1, \mathbf{z}_2)$
  3. for $j = 1$ to 2 :
     - if $b = \mathit{false}$ *(i.e. $\mathbf{x}$ is not completly feasible)* then
       Contract the infeasible region using $\mathcal{C}_{out}$ and $\mathcal{C}_f$,
       Extract $\mathbf{z}_{feas}$ a feasible part of $\mathbf{z}_j$ using $\mathcal{C}_{in}$,
       Insert $(\mathbf{z}_{feas}, \mathit{true})$ in $\mathcal{L}$.
       Insert the rest $(\mathbf{z}_j, \mathit{false})$ in $\mathcal{L}$.
     - else *(i.e. $\mathbf{x}$ is entirely feasible)*
       Contract $\mathbf{z}_j$ using $\mathcal{C}_f$,
       Try to **find** a local optimum without constraint in $[\mathbf{z}_j]$,
       if *succeed* then Update $\tilde{f}$ insert $(\mathbf{z}_j, \mathit{true})$ in $\mathcal{L}$.
- stopping criterion

Global Optimization based on Contractor

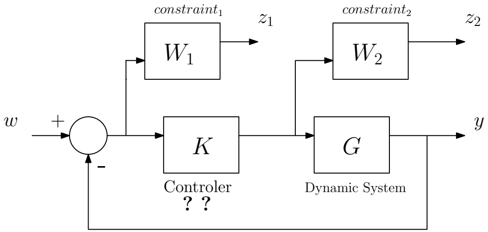
- $\mathcal{L} := \{(\mathbf{x}, \mathit{false})\}$, The boolean indicate if $\mathbf{x}$ is entirely feasible
- Do
  1. Extract from $\mathcal{L}$ a element $(\mathbf{z}, b)$,
  2. Bisect $\mathbf{z}$ following a bisector $\mathcal{B}$: $(\mathbf{z}_1, \mathbf{z}_2)$
  3. for $j = 1$ to 2 :
     - if $b = \mathit{false}$ *(i.e.* $\mathbf{x}$ *is not completly feasible)* then
       Contract the infeasible region using $\mathcal{C}_{out}$ and $\mathcal{C}_f$,
       Extract $\mathbf{z}_{feas}$ a feasible part of $\mathbf{z}_j$ using $\mathcal{C}_{in}$,
       Insert $(\mathbf{z}_{feas}, \mathit{true})$ in $\mathcal{L}$.
       Insert the rest $(\mathbf{z}_j, \mathit{false})$ in $\mathcal{L}$.
     - else *(i.e.* $\mathbf{x}$ *is entirely feasible)*
       Contract $\mathbf{z}_j$ using $\mathcal{C}_f$,
       Try to **find** a local optimum without constraint in $[\mathbf{z}_j]$,
       if *succeed* then Update $\tilde{f}$ insert $(\mathbf{z}_j, \mathit{true})$ in $\mathcal{L}$.
- stopping criterion

Global Optimization based on Contractor

- $\mathcal{L} := \{(\mathbf{x}, \mathit{false})\}$, The boolean indicate if $\mathbf{x}$ is entirely feasible
- Do
  1. Extract from $\mathcal{L}$ a element $(\mathbf{z}, b)$,
  2. Bisect $\mathbf{z}$ following a bisector $\mathcal{B}$: $(\mathbf{z}_1, \mathbf{z}_2)$
  3. for $j = 1$ to 2 :
     - if $b = \mathit{false}$ *(i.e.* $\mathbf{x}$ *is not completly feasible)* then
       Contract the infeasible region using $\mathcal{C}_{out}$ and $\mathcal{C}_f$,
       Extract $\mathbf{z}_{feas}$ a feasible part of $\mathbf{z}_j$ using $\mathcal{C}_{in}$,
       Insert $(\mathbf{z}_{feas}, \mathit{true})$ in $\mathcal{L}$.
       Insert the rest $(\mathbf{z}_j, \mathit{false})$ in $\mathcal{L}$.
     - else *(i.e.* $\mathbf{x}$ *is entirely feasible)*
       Contract $\mathbf{z}_j$ using $\mathcal{C}_f$,
       Try to **find** a local optimum without constraint in $[\mathbf{z}_j]$,
       if *succeed* then Update $\tilde{f}$ insert $(\mathbf{z}_j, \mathit{true})$ in $\mathcal{L}$.
- stopping criterion

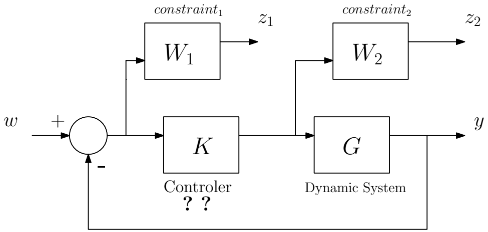
Contents

H_∞ control synthesis under structural constraints



H_∞ control synthesis $\Rightarrow$ Guarantee the robustness and stability

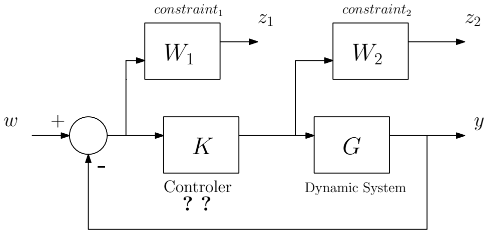
H_∞ control synthesis under structural constraints



H_∞ control synthesis $\Rightarrow$ Guarantee the robustness and stability

$$||P||_\infty = \sup_\omega(\sigma_{\max}(P(j\omega)))$$

H_∞ control synthesis under structural constraints

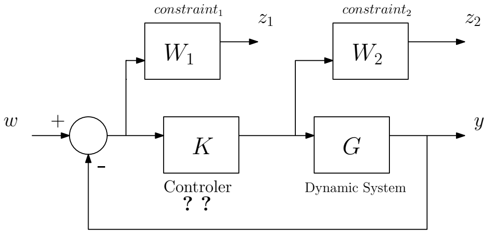


H_∞ control synthesis $\Rightarrow$ Guarantee the robustness and stability

$$||P||_\infty = \sup_\omega(\sigma_{\max}(P(j\omega)))$$

- Classical approach without structural constraint
  $\rightarrow$ LMI system, SDP opimization

H_∞ control synthesis under structural constraints



H_∞ control synthesis $\Rightarrow$ Guarantee the robustness and stability

$$||P||_\infty = \sup_\omega(\sigma_{\max}(P(j\omega)))$$

- Classical approach without structural constraint
  $\rightarrow$ LMI system, SDP opimization
- Classical approach with structural constraint
  $\rightarrow$ Nonsmooth local optimization

Mathematical Modelization

$$\begin{cases} \min\limits_{\mathbf{k},\gamma} & \gamma \\ \forall\omega, & \left\| \dfrac{W_1(j\omega)}{1+G(j\omega)K(j\omega)} \right\|_\infty \leq \gamma, \\ \forall\omega, & \left\| \dfrac{W_2(j\omega)K(j\omega)}{1+G(j\omega)K(j\omega)} \right\|_\infty \leq \gamma, \\ & \text{The closed-loop system must be } \textit{stable}. \end{cases}$$

Mathematical Modelization

$$\begin{cases} \min\limits_{\mathbf{k},\gamma} & \gamma \\ \forall\omega, & \left\| \dfrac{W_1(j\omega)}{1+G(j\omega)K(j\omega)} \right\|_\infty \le \gamma, \\ \forall\omega, & \left\| \dfrac{W_2(j\omega)K(j\omega)}{1+G(j\omega)K(j\omega)} \right\|_\infty \le \gamma, \\ & \text{The closed-loop system must be } \textit{stable}. \end{cases}$$

Stability:

The system is stable iff its poles are strictly negative.

$\Leftrightarrow$

The roots of the denominator of $\frac{1}{1+G(s)K(s)}$ are strictly negative

Mathematical Modelization

$$\begin{cases} \min\limits_{\mathbf{k},\gamma} & \gamma \\ \forall \omega, & \left\| \dfrac{W_1(j\omega)}{1+G(j\omega)K(j\omega)} \right\|_\infty \leq \gamma, \\ \forall \omega, & \left\| \dfrac{W_2(j\omega)K(j\omega)}{1+G(j\omega)K(j\omega)} \right\|_\infty \leq \gamma, \\ & \text{The closed-loop system must be } \textit{stable}. \end{cases}$$

Stability:

The system is stable iff its poles are strictly negative.

$\Leftrightarrow$

The roots of the denominator of $\frac{1}{1+G(s)K(s)}$ are strictly negative

$\Longrightarrow$ Routh-Hurwitz stability criterion

Routh-Hurwitz stability criterion

$$P(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0$$

$v_{1,1} = a_n$	$v_{1,2} = a_{n-2}$	$v_{1,3} = a_{n-4}$	$v_{1,4} = a_{n-6}$
$v_{2,1} = a_{n-1}$	$v_{2,2} = a_{n-3}$	$v_{2,3} = a_{n-5}$	$v_{2,4} = a_{n-7}$
$v_{3,1} = \frac{-1}{v_{2,1}} \begin{vmatrix} v_{1,1} & v_{1,2} \\ v_{2,1} & v_{2,2} \end{vmatrix}$	$v_{3,2} = \frac{-1}{v_{2,1}} \begin{vmatrix} v_{1,1} & v_{1,3} \\ v_{2,1} & v_{2,3} \end{vmatrix}$	$v_{3,3} = \frac{-1}{v_{2,1}} \begin{vmatrix} v_{1,1} & v_{1,4} \\ v_{2,1} & v_{2,4} \end{vmatrix}$	$\dots$
$v_{4,1} = \frac{-1}{v_{3,1}} \begin{vmatrix} v_{2,1} & v_{2,2} \\ v_{3,1} & v_{3,2} \end{vmatrix}$	$v_{4,2} = \frac{-1}{v_{3,1}} \begin{vmatrix} v_{2,1} & v_{2,3} \\ v_{3,1} & v_{3,3} \end{vmatrix}$	$\dots$	$\dots$
$v_{5,1} = \frac{-1}{v_{4,1}} \begin{vmatrix} v_{3,1} & v_{3,2} \\ v_{4,1} & v_{4,2} \end{vmatrix}$	$\dots$	$\dots$	$\dots$
$\vdots$	$\ddots$	$\ddots$	$\ddots$

Routh-Hurwitz stability criterion

$$P(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0$$

$v_{1,1} = a_n$	$v_{1,2} = a_{n-2}$	$v_{1,3} = a_{n-4}$	$v_{1,4} = a_{n-6}$
$v_{2,1} = a_{n-1}$	$v_{2,2} = a_{n-3}$	$v_{2,3} = a_{n-5}$	$v_{2,4} = a_{n-7}$
$v_{3,1} = \frac{-1}{v_{2,1}} \begin{vmatrix} v_{1,1} & v_{1,2} \\ v_{2,1} & v_{2,2} \end{vmatrix}$	$v_{3,2} = \frac{-1}{v_{2,1}} \begin{vmatrix} v_{1,1} & v_{1,3} \\ v_{2,1} & v_{2,3} \end{vmatrix}$	$v_{3,3} = \frac{-1}{v_{2,1}} \begin{vmatrix} v_{1,1} & v_{1,4} \\ v_{2,1} & v_{2,4} \end{vmatrix}$	$\ldots$
$v_{4,1} = \frac{-1}{v_{3,1}} \begin{vmatrix} v_{2,1} & v_{2,2} \\ v_{3,1} & v_{3,2} \end{vmatrix}$	$v_{4,2} = \frac{-1}{v_{3,1}} \begin{vmatrix} v_{2,1} & v_{2,3} \\ v_{3,1} & v_{3,3} \end{vmatrix}$	$\ldots$	$\ldots$
$v_{5,1} = \frac{-1}{v_{4,1}} \begin{vmatrix} v_{3,1} & v_{3,2} \\ v_{4,1} & v_{4,2} \end{vmatrix}$	$\ldots$	$\ldots$	$\ldots$
$\vdots$	$\ddots$	$\ddots$	$\ddots$

If all the value of the **first column** are positive, all roots of P are negative.

Definition of the feasible set

$$\mathbb{K}^1_\omega = \left\{ (k,\gamma,\omega) \ : \ \left\| \frac{W_1(i\omega)}{1+G(i\omega)K(i\omega)} \right\|_\infty \leq \gamma \right\},$$

$$\mathbb{K}^2_\omega = \left\{ (k,\gamma,\omega) \ : \ \left\| \frac{W_2(i\omega)K(i\omega)}{1+G(i\omega)K(i\omega)} \right\|_\infty \leq \gamma \right\},$$

Definition of the feasible set

$$\mathbb{K}^1_\omega = \left\{(k,\gamma,\omega) \ : \ \left\|\frac{W_1(i\omega)}{1+G(i\omega)K(i\omega)}\right\|_\infty \leq \gamma\right\},$$

$$\mathbb{K}^2_\omega = \left\{(k,\gamma,\omega) \ : \ \left\|\frac{W_2(i\omega)K(i\omega)}{1+G(i\omega)K(i\omega)}\right\|_\infty \leq \gamma\right\},$$

$$\mathbb{K}^4 = \bigcap_{\omega\in[10^{-2},10^2]} \mathbb{K}^1_\omega \cap \mathbb{K}^2_\omega.$$

Definition of the feasible set

$$\mathbb{K}^1_\omega = \left\{(k,\gamma,\omega) \ : \ \left\|\frac{W_1(i\omega)}{1+G(i\omega)K(i\omega)}\right\|_\infty \leq \gamma\right\},$$

$$\mathbb{K}^2_\omega = \left\{(k,\gamma,\omega) \ : \ \left\|\frac{W_2(i\omega)K(i\omega)}{1+G(i\omega)K(i\omega)}\right\|_\infty \leq \gamma\right\},$$

$$\mathbb{K}^4 = \bigcap_{\omega\in[10^{-2},10^2]} \mathbb{K}^1_\omega \cap \mathbb{K}^2_\omega.$$

The Routh's condition / stability of the closed-loop system:

$$\mathbb{K}^{Routh} = \{(k,\gamma) \ : \ \begin{cases} a_n(k,\gamma) > 0, \\ a_{n-1}(k,\gamma) > 0, \\ v_{2,1}(k,\gamma) > 0, \\ \\ \dots \end{cases} \}.$$

Definition of the feasible set

$$\mathbb{K}^1_\omega = \left\{(k,\gamma,\omega) \; : \; \left\| \frac{W_1(i\omega)}{1+G(i\omega)K(i\omega)} \right\|_\infty \leq \gamma \right\},$$

$$\mathbb{K}^2_\omega = \left\{(k,\gamma,\omega) \; : \; \left\| \frac{W_2(i\omega)K(i\omega)}{1+G(i\omega)K(i\omega)} \right\|_\infty \leq \gamma \right\},$$

$$\mathbb{K}^4 = \bigcap_{\omega\in[10^{-2},10^2]} \mathbb{K}^1_\omega \cap \mathbb{K}^2_\omega.$$

The Routh's condition / stability of the closed-loop system:

$$\mathbb{K}^{Routh} = \left\{(k,\gamma) \; : \; \begin{cases} a_n(k,\gamma) > 0, \\ a_{n-1}(k,\gamma) > 0, \\ v_{2,1}(k,\gamma) > 0, \\ \\ \dots \end{cases} \right\}.$$

The feasible set of our problem is $\mathbb{K} = \mathbb{K}^4 \cap \mathbb{K}^{Routh}$.

Contractor Modelization: Properties

Let $\mathcal{A}$ a contractor for the equation $f(x) = 0$, and $\mathcal{B}$ a contractor for the equation $g(x) = 0$, then:

Intersection, Composition

$\mathcal{A} \cap \mathcal{B}$ and $\mathcal{A} \circ \mathcal{B}$ are two contractors for the region:

$$\{x \in \mathbb{R}^n \; : \; f(x) = 0 \; AND \; g(x) = 0\}$$

Union

$\mathcal{A} \cup \mathcal{B}$ is a contractor for the region:

$$\{x \in \mathbb{R}^n \; : \; f(x) = 0 \; OR \; g(x) = 0\}$$

Contractor with Quantifiers

Let $\mathcal{C}$ be a contractor for a set $\mathbb{Z} = \mathbb{X} \times \mathbb{Y}$,
$\pi_{\mathbb{X}}$ the projection of $\mathbb{Z}$ over $\mathbb{X}$.

Contractor ForAll / Exists

$$\begin{cases} \mathcal{C}^{\cap\mathbb{Y}}(\mathbf{x}) = \bigcap_{y \in \mathbb{Y}} \pi_{\mathbb{X}}\left(\mathcal{C}(\mathbf{x} \times \{y\})\right), \\ \\ \mathcal{C}^{\cup\mathbb{Y}}(\mathbf{x}) = \bigcup_{y \in \mathbb{Y}} \pi_{\mathbb{X}}\left(\mathcal{C}(\mathbf{x} \times \{y\})\right). \end{cases}$$

Property

$\mathcal{C}^{\cap\mathbb{Y}}$ is a contractor for $\{x : \forall y \in \mathbb{Y}, (x, y) \in \mathbb{Z}\}$
$\mathcal{C}^{\cup\mathbb{Y}}$ is a contractor for $\{x : \exists y \in \mathbb{Y}, (x, y) \in \mathbb{Z}\}$.

Contractor with Quantifiers

Let $\mathcal{C}$ be a contractor for a set $\mathbb{Z} = \mathbb{X} \times \mathbb{Y}$,
$\pi_{\mathbb{X}}$ the projection of $\mathbb{Z}$ over $\mathbb{X}$.

Contractor ForAll / Exists

$$\begin{cases} \mathcal{C}^{\cap \mathbb{Y}}(\mathbf{x}) = \bigcap_{y \in \mathbb{Y}} \pi_{\mathbb{X}} \left(\mathcal{C}(\mathbf{x} \times \{y\}) \right), \\ \\ \mathcal{C}^{\cup \mathbb{Y}}(\mathbf{x}) = \bigcup_{y \in \mathbb{Y}} \pi_{\mathbb{X}} \left(\mathcal{C}(\mathbf{x} \times \{y\}) \right). \end{cases}$$

Property

$\mathcal{C}^{\cap \mathbb{Y}}$ is a contractor for $\{x : \forall y \in \mathbb{Y}, (x, y) \in \mathbb{Z}\}$
$\mathcal{C}^{\cup \mathbb{Y}}$ is a contractor for $\{x : \exists y \in \mathbb{Y}, (x, y) \in \mathbb{Z}\}$.

Contractor with Quantifiers

Let $\mathcal{C}$ be a contractor for a set $\mathbb{Z} = \mathbb{X} \times \mathbb{Y}$,
$\pi_{\mathbb{X}}$ the projection of $\mathbb{Z}$ over $\mathbb{X}$.

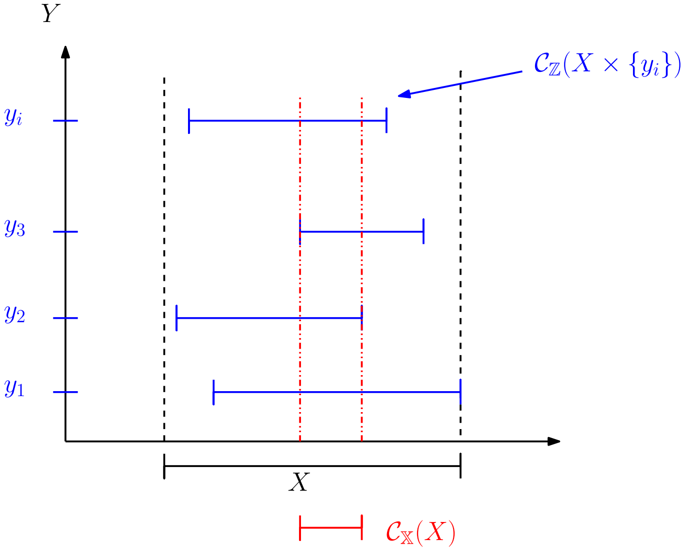
Contractor ForAll / Exists

$$\begin{cases} \mathcal{C}^{\cap\mathbb{Y}}(\mathbf{x}) = \bigcap_{y\in\mathbb{Y}} \pi_{\mathbb{X}}\left(\mathcal{C}(\mathbf{x} \times \{y\})\right), \\ \\ \mathcal{C}^{\cup\mathbb{Y}}(\mathbf{x}) = \bigcup_{y\in\mathbb{Y}} \pi_{\mathbb{X}}\left(\mathcal{C}(\mathbf{x} \times \{y\})\right). \end{cases}$$

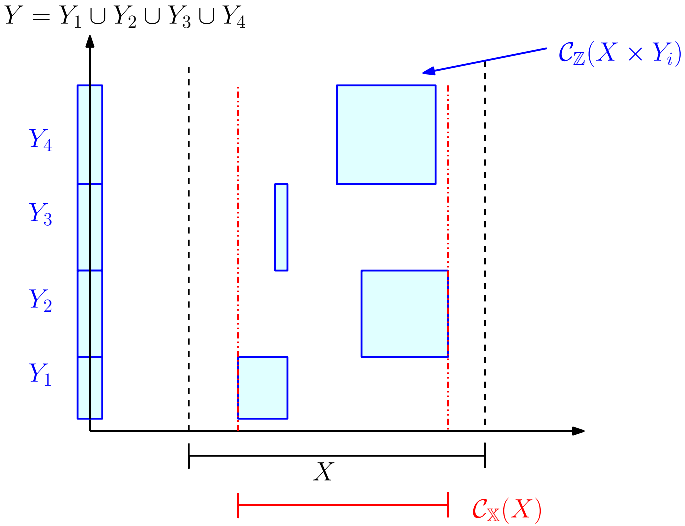
Property

$\mathcal{C}^{\cap\mathbb{Y}}$ is a contractor for $\{x : \forall y \in \mathbb{Y}, (x,y) \in \mathbb{Z}\}$
$\mathcal{C}^{\cup\mathbb{Y}}$ is a contractor for $\{x : \exists y \in \mathbb{Y}, (x,y) \in \mathbb{Z}\}$.

Contractor CtcForAll: $\mathbb{X} = \{x : \forall y \in \mathbb{Y}, (x, y) \in \mathbb{Z}\}$



Contractor CtcExist: $\mathbb{X} = \{x \ : \ \exists y \in \mathbb{Y}, (x, y) \in \mathbb{Z}\}$



Construction of Contractors $\mathcal{C}_{out}$ of the feasible set $\mathbb{K}$

$\mathcal{C}_{out}$ will eliminate all the part of a box which are not in $\mathbb{K}$.

$$\begin{aligned}
\mathbb{K}^1_\omega &= \left\{(k,\gamma,\omega)\ :\ \left\|\frac{W_1(i\omega)}{1+G(i\omega)K(i\omega)}\right\|_\infty \leq \gamma\right\},\\
\mathbb{K}^2_\omega &= \left\{(k,\gamma,\omega)\ :\ \left\|\frac{W_2(i\omega)K(i\omega)}{1+G(i\omega)K(i\omega)}\right\|_\infty \leq \gamma\right\},
\end{aligned}$$

$$\mathbb{K} = \left(\bigcap_{\omega\in[10^{-2},10^2]} \mathbb{K}^1_\omega \cap \mathbb{K}^2_\omega\right) \cap \mathbb{K}^{Routh}.$$

Construction of Contractors $\mathcal{C}_{out}$ of the feasible set $\mathbb{K}$

$\mathcal{C}_{out}$ will eliminate all the part of a box which are not in $\mathbb{K}$.

$$\mathbb{K} = \left(\bigcap_{\omega \in [10^{-2}, 10^2]} \mathbb{K}^1_\omega \cap \mathbb{K}^2_\omega \right) \cap \mathbb{K}^{Routh}.$$

Construction of Contractors $\mathcal{C}_{out}$ of the feasible set $\mathbb{K}$

$\mathcal{C}_{out}$ will eliminate all the part of a box which are not in $\mathbb{K}$.

$$\mathbb{K} = \left(\bigcap_{\omega \in [10^{-2}, 10^2]} \mathbb{K}^1_\omega \cap \mathbb{K}^2_\omega \right) \cap \mathbb{K}^{Routh}.$$

1. Create the contractor $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_{Routh}$ based on $\mathbb{K}^1_\omega$, $\mathbb{K}^2_\omega$ and $\mathbb{K}^{Routh}$:
   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, . . .

Construction of Contractors $\mathcal{C}_{out}$ of the feasible set $\mathbb{K}$

$\mathcal{C}_{out}$ will eliminate all the part of a box which are not in $\mathbb{K}$.

$$\mathbb{K} = \left(\bigcap_{\omega \in [10^{-2}, 10^2]} \mathbb{K}^1_\omega \cap \mathbb{K}^2_\omega \right) \cap \mathbb{K}^{Routh}.$$

1. Create the contractor $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_{Routh}$ based on $\mathbb{K}^1_\omega$, $\mathbb{K}^2_\omega$ and $\mathbb{K}^{Routh}$:
   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, . . .
2. Inter: $\mathcal{C}_3(\mathbf{k}, \gamma, \omega) = \mathcal{C}_1(\mathbf{k}, \gamma, \omega) \cap \mathcal{C}_2(\mathbf{k}, \gamma, \omega)$.

Construction of Contractors $\mathcal{C}_{out}$ of the feasible set $\mathbb{K}$

$\mathcal{C}_{out}$ will eliminate all the part of a box which are not in $\mathbb{K}$.

$$\mathbb{K} = \left(\bigcap_{\omega \in [10^{-2}, 10^2]} \mathbb{K}^1_\omega \cap \mathbb{K}^2_\omega \right) \cap \mathbb{K}^{Routh}.$$

1. Create the contractor $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_{Routh}$ based on $\mathbb{K}^1_\omega$, $\mathbb{K}^2_\omega$ and $\mathbb{K}^{Routh}$.
   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...
2. Inter: $\mathcal{C}_3(\mathbf{k}, \gamma, \omega) = \mathcal{C}_1(\mathbf{k}, \gamma, \omega) \cap \mathcal{C}_2(\mathbf{k}, \gamma, \omega)$.
3. CtcForAll: $\mathcal{C}^{\cap\omega}(\mathbf{k}, \gamma) = \bigcap_{\omega \in [10^{-2}, 10^2]} \mathcal{C}_3(\mathbf{k}, \gamma, \omega)$.

Construction of Contractors $\mathcal{C}_{out}$ of the feasible set $\mathbb{K}$

$\mathcal{C}_{out}$ will eliminate all the part of a box which are not in $\mathbb{K}$.

$$\mathbb{K} = \left(\bigcap_{\omega \in [10^{-2}, 10^2]} \mathbb{K}^1_\omega \cap \mathbb{K}^2_\omega \right) \cap \mathbb{K}^{Routh}.$$

1. Create the contractor $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_{Routh}$ based on $\mathbb{K}^1_\omega$, $\mathbb{K}^2_\omega$ and $\mathbb{K}^{Routh}$.
   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...
2. Inter: $\mathcal{C}_3(\mathbf{k}, \gamma, \omega) = \mathcal{C}_1(\mathbf{k}, \gamma, \omega) \cap \mathcal{C}_2(\mathbf{k}, \gamma, \omega)$.
3. CtcForAll: $\mathcal{C}^{\cap\omega}(\mathbf{k}, \gamma) = \bigcap_{\omega \in [10^{-2}, 10^2]} \mathcal{C}_3(\mathbf{k}, \gamma, \omega)$.
4. Inter: $\mathcal{C}_{out} = \mathcal{C}^{\cap\omega} \cap \mathcal{C}_{Routh}$.

Construction of Contractors $\mathcal{C}_{in}$ of the unfeasible set $\overline{\mathbb{K}}$

$\mathcal{C}_{in}$ will eliminate all the part of a box which are not in $\overline{\mathbb{K}}$.

$$\overline{\mathbb{K}^1_\omega} = \left\{(k,\gamma,\omega) \ : \ \left\| \frac{W_1(i\omega)}{1+G(i\omega)K(i\omega)} \right\| > \gamma \right\},$$

$$\overline{\mathbb{K}^2_\omega} = \left\{(k,\gamma,\omega) \ : \ \left\| \frac{W_2(i\omega)K(i\omega)}{1+G(i\omega)K(i\omega)} \right\| > \gamma \right\},$$

$$\overline{\mathbb{K}} = \left(\bigcup_{\omega\in[10^{-2},10^2]} \overline{\mathbb{K}^1_\omega} \cup \overline{\mathbb{K}^2_\omega} \right) \cup \overline{\mathbb{K}^{Routh}}.$$

Construction of Contractors $\mathcal{C}_{in}$ of the unfeasible set $\overline{\mathbb{K}}$

$\mathcal{C}_{in}$ will eliminate all the part of a box which are not in $\overline{\mathbb{K}}$.

$$\overline{\mathbb{K}} = \left(\bigcup_{\omega \in [10^{-2}, 10^{2}]} \overline{\mathbb{K}^1_\omega} \cup \overline{\mathbb{K}^2_\omega} \right) \cup \overline{\mathbb{K}^{Routh}}.$$

Construction of Contractors $\mathcal{C}_{in}$ of the unfeasible set $\overline{\mathbb{K}}$

$\mathcal{C}_{in}$ will eliminate all the part of a box which are not in $\overline{\mathbb{K}}$.

$$\overline{\mathbb{K}} = \left(\bigcup_{\omega \in [10^{-2}, 10^{2}]} \overline{\mathbb{K}_\omega^1} \cup \overline{\mathbb{K}_\omega^2} \right) \cup \overline{\mathbb{K}^{Routh}}.$$

1. Create the contractor $\mathcal{C}_{\overline{1}}$, $\mathcal{C}_{\overline{2}}$ and $\mathcal{C}_{\overline{Routh}}$ based on $\overline{\mathbb{K}_\omega^1}$, $\overline{\mathbb{K}_\omega^2}$ and $\overline{\mathbb{K}^{Routh}}$:
   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...

Construction of Contractors $\mathcal{C}_{in}$ of the unfeasible set $\overline{\mathbb{K}}$

$\mathcal{C}_{in}$ will eliminate all the part of a box which are not in $\overline{\mathbb{K}}$.

$$\overline{\mathbb{K}} = \left(\bigcup_{\omega\in[10^{-2},10^2]} \overline{\mathbb{K}^1_\omega} \cup \overline{\mathbb{K}^2_\omega} \right) \cup \overline{\mathbb{K}^{Routh}}.$$

1. Create the contractor $\mathcal{C}_{\overline{1}}$, $\mathcal{C}_{\overline{2}}$ and $\mathcal{C}_{\overline{Routh}}$ based on $\overline{\mathbb{K}^1_\omega}$, $\overline{\mathbb{K}^2_\omega}$ and $\overline{\mathbb{K}^{Routh}}$:
   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...
2. Union: $\mathcal{C}_{\overline{3}}(\mathbf{k},\gamma,\omega) = \mathcal{C}_{\overline{1}}(\mathbf{k},\gamma,\omega) \cup \mathcal{C}_{\overline{2}}(\mathbf{k},\gamma,\omega).$

Construction of Contractors $\mathcal{C}_{in}$ of the unfeasible set $\overline{\mathbb{K}}$

$\mathcal{C}_{in}$ will eliminate all the part of a box which are not in $\overline{\mathbb{K}}$.

$$\overline{\mathbb{K}} = \left(\bigcup_{\omega \in [10^{-2}, 10^2]} \overline{\mathbb{K}^1_\omega} \cup \overline{\mathbb{K}^2_\omega} \right) \cup \overline{\mathbb{K}^{Routh}}.$$

1. Create the contractor $\mathcal{C}_{\overline{1}}$, $\mathcal{C}_{\overline{2}}$ and $\mathcal{C}_{\overline{Routh}}$ based on $\overline{\mathbb{K}^1_\omega}$, $\overline{\mathbb{K}^2_\omega}$ and $\overline{\mathbb{K}^{Routh}}$:
   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...
2. Union: $\mathcal{C}_{\overline{3}}(\mathbf{k}, \gamma, \omega) = \mathcal{C}_{\overline{1}}(\mathbf{k}, \gamma, \omega) \cup \mathcal{C}_{\overline{2}}(\mathbf{k}, \gamma, \omega)$.
3. CtcExist: $\mathcal{C}^{\cup\omega}(\mathbf{k}, \gamma) = \bigcup_{\omega \in [10^{-2}, 10^2]} \mathcal{C}_{\overline{3}}(\mathbf{k}, \gamma, \omega)$.

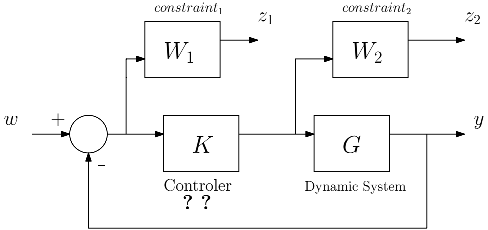
Construction of Contractors $\mathcal{C}_{in}$ of the unfeasible set $\overline{\mathbb{K}}$

$\mathcal{C}_{in}$ will eliminate all the part of a box which are not in $\overline{\mathbb{K}}$.

$$\overline{\mathbb{K}} = \left(\bigcup_{\omega \in [10^{-2}, 10^{2}]} \overline{\mathbb{K}^1_\omega} \cup \overline{\mathbb{K}^2_\omega} \right) \cup \overline{\mathbb{K}^{Routh}}.$$

1. Create the contractor $\mathcal{C}_{\overline{1}}$, $\mathcal{C}_{\overline{2}}$ and $\mathcal{C}_{\overline{Routh}}$ based on $\overline{\mathbb{K}^1_\omega}$, $\overline{\mathbb{K}^2_\omega}$ and $\overline{\mathbb{K}^{Routh}}$:
   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...
2. Union: $\mathcal{C}_{\overline{3}}(\mathbf{k}, \gamma, \omega) = \mathcal{C}_{\overline{1}}(\mathbf{k}, \gamma, \omega) \cup \mathcal{C}_{\overline{2}}(\mathbf{k}, \gamma, \omega)$.
3. CtcExist: $\mathcal{C}^{\cup\omega}(\mathbf{k}, \gamma) = \bigcup_{\omega \in [10^{-2}, 10^{2}]} \mathcal{C}_{\overline{3}}(\mathbf{k}, \gamma, \omega)$.
4. Union: $\mathcal{C}_{in} = \mathcal{C}^{\cup\omega} \cup \mathcal{C}_{\overline{Routh}}$.

First Application with second order dynamic system



The transfer function of the dynamic system:

$$G(s) = \frac{1}{s^2 + 1.4s + 1}, \quad K(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{1+s}.$$

$$W_1(s) = \frac{s+100}{100s+1}, \qquad W_2(s) = \frac{10s+1}{s+10}.$$

Overview of the equation

$$\forall \omega \in [10^{-2}, 10^{2}], \left\| \frac{W_1(j\omega)}{1 + G(j\omega)K(j\omega)} \right\|_\infty \leq \gamma.$$

$$\Longleftrightarrow$$

$$\forall \omega \in [10^{-2}, 10^{2}], \frac{w^2\left(w^2+1.0\right)\left(w^2+10000.0\right)\left(25.0\, w^4-1.0\, w^2+25.0\right)}{(10000.0\, w^2+1.0) f_1(\mathbf{k},\gamma,\omega)} \leq \gamma.$$

Overview of the equation

$$\forall \omega \in [10^{-2}, 10^{2}], \left\| \frac{W_1(j\omega)}{1+G(j\omega)K(j\omega)} \right\|_\infty \leq \gamma.$$

$$\Longleftrightarrow$$

$$\forall \omega \in [10^{-2}, 10^{2}], \frac{w^2\left(w^2+1.0\right)\left(w^2+10000.0\right)\left(25.0\,w^4-1.0\,w^2+25.0\right)}{(10000.0\,w^2+1.0)\,f_1(\mathbf{k},\gamma,\omega)} \leq \gamma.$$

$$f_1(\mathbf{k},\gamma,\omega) = 25.0\mathrm{kd}^2 w^4 + 25.0\mathrm{kp}^2 w^2 + 25.0\mathrm{kp}^2 w^4 - 1.0\mathrm{ki}\left(50.0\mathrm{kd}w^2 + 70.0w^2 + 70.0w^4\right) + \mathrm{ki}^2\left(25.0w^2 + 25.0\right) + 120.0\mathrm{kd}w^4 - 50.0\mathrm{kd}w^6 + 50.0\mathrm{kp}w^2 - 50.0\mathrm{kp}w^6 + 25.0w^2 + 24.0w^4 + 24.0w^6 + 25.0w^8 + 50.0\mathrm{kdkp}w^4$$

Overview of the equation

$$\forall \omega \in [10^{-2}, 10^{2}], \left\| \frac{W_1(j\omega)}{1 + G(j\omega)K(j\omega)} \right\|_\infty \leq \gamma.$$

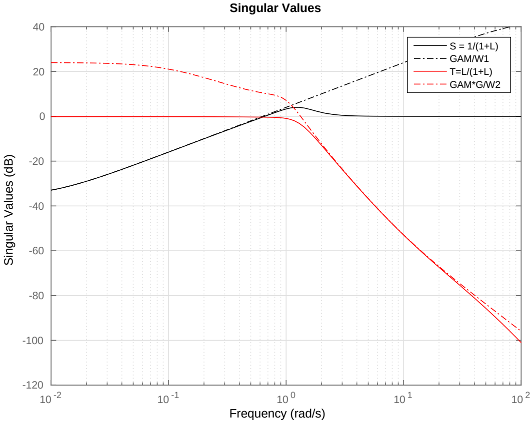
$$\Longleftrightarrow$$

$$\forall \omega \in [10^{-2}, 10^{2}], \frac{w^2\left(w^2+1.0\right)\left(w^2+10000.0\right)\left(25.0\,w^4-1.0\,w^2+25.0\right)}{(10000.0\,w^2+1.0)f_1(\mathbf{k},\gamma,\omega)} \leq \gamma.$$

$$f_1(\mathbf{k},\gamma,\omega) = 25.0\mathrm{kd}^2 w^4 + 25.0\mathrm{kp}^2 w^2 + 25.0\mathrm{kp}^2 w^4 - 1.0\mathrm{ki}\left(50.0\mathrm{kd}w^2 + 70.0w^2 + 70.0w^4\right) + \mathrm{ki}^2\left(25.0w^2 + 25.0\right) + 120.0\mathrm{kd}w^4 - 50.0\mathrm{kd}w^6 + 50.0\mathrm{kp}w^2 - 50.0\mathrm{kp}w^6 + 25.0w^2 + 24.0w^4 + 24.0w^6 + 25.0w^8 + 50.0\mathrm{kdkp}w^4$$

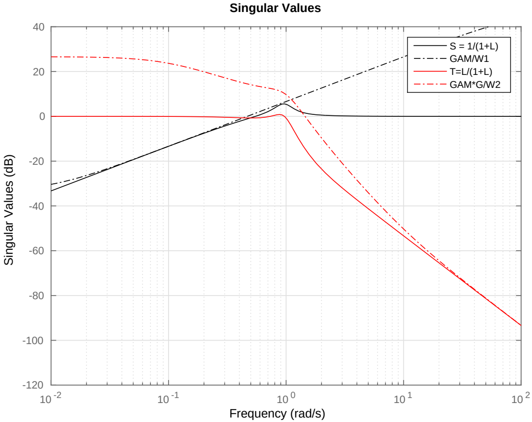
$$\forall u \in [-2, 2], \omega = 10^u.$$

Results with hinfsyn of Matlab



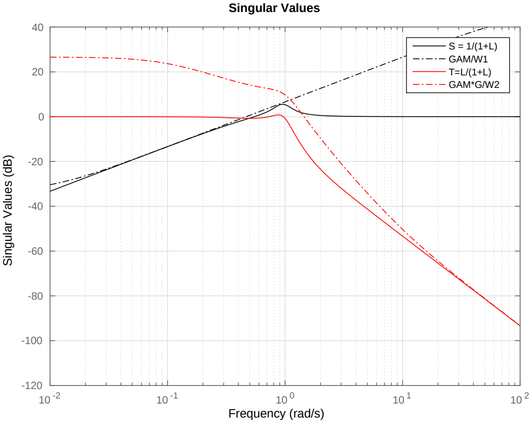
$$\gamma = 1.5887$$

Results with hinfstruct of Matlab



$$\gamma = 2.1414$$

Results with Global Optimization of IBEX

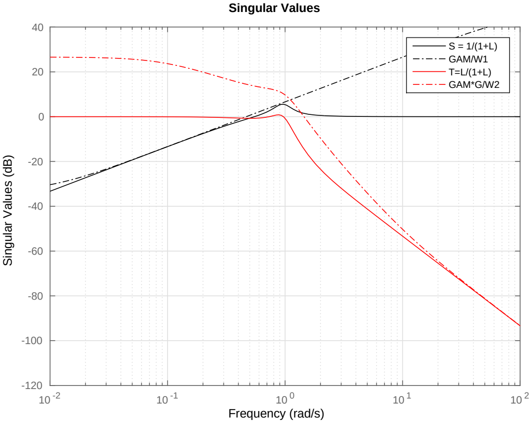


$$\gamma = 2.1414$$

$\Longrightarrow$ same result as with hinfstruct,

but with a global optimality proof!

Results with Global Optimization of IBEX



$$\gamma = 2.1414$$

$\Longrightarrow$ same result as with hinfstruct,

but with a global optimality proof!

Conclusion

Contractor Programming:

- Generate the Modelization and the adapted Solver in the same time,
- Consider heterogeneous constraints without changing the solver,
- Give all the tools to the expert of the application.

Conclusion

IBEX

http://www.ibex-lib.org

- Interval Arithmetic Interface: Filib, Gaol, Profil/BIAS.
- Affine Arithmetic.
- Linear Solver Interface: Soplex, CPLEX, CLP.
- Symbolic and Automatic Differentiation.
- AMPL Interface.
- Reliable computation of Ordinary Differential Equation (DynIBEX)
- CSP solver, Global Optimization solver.
- Available on Linux, MacOSX and Windows.

Fork it on GitHub

http://github.com/ibex-team/ibex-lib