Optimization based control for Robots
some solutions for the implementation issue

Benoît CLEMENT

June, 30, 2016
Control point of view

Today is dedicated to answer few questions with a control point of view in Robotics.

- Which Robotics?
- How to control a Robot?
- Are the GNC algorithms are a big issue?
Outline

1. **Context**
2. **Control issue**
   - Problem Statement
   - Some answers... with drawbacks
3. **Optimisation based control**
   - Global Optimization
   - General pattern for global optimization
   - Application to $H_\infty$ control
4. **AUV Control Application**
   - CISCREA: description and challenges
   - Robust control
   - Results
5. **Conclusions**
OSM: Teaching and Research Department

Large scope of teaching activities: hydrography, oceanography, embedded electronics, signal processing, information technology, computer science, robotics, etc.

Research topics:
- Hydrography/Oceanography
- Underwater robotics
- Sonar systems
- Data Processing

Application field: Maritime environment, civilian and defense.
Focus on Robotics

- Robotics issues:
  - Guidance, Navigation and Control
  - Group of Robots: interaction management
  - Localisation

- Academic tools:
  - Interval Analysis
  - Data processing
  - Global Optimization
  - Robust Control
Teaching

- Linear Control and Sensors
- Mobile Robotics
- Localisation and Kalman filter
- Prototyping Robots
- Middleware and Compilation
- Simulation and nonlinear control
- Digital conception
- Robust Control
- Vision
- Robotics Architecture
Lab-STICC

- UMR CNRS 6285
- CID : Connaissance Information Decision
- PRASYS : Perception, Robotics, Autonomous Systems
Some Robots

Demonstrations with movies...
What is robust control

**Question**: find a controller that insures performances to the closed loop

- \( P \) et \( K \) : systèmes LTI ou LPV - MIMO
- \( u = \) commandes, \( y = \) mesures
- Transferts \( T_i \) utilisés pour spécifier différents objectifs de performance ou robustesse :

\[
\begin{align*}
\| T_{z_1}/w_1 \| &< \gamma_1 \\
\vdots \\
\| T_{z_p}/w_p \| &< \gamma_p
\end{align*}
\]

\[
K = \begin{pmatrix} A_K & B_K \\ C_K & D_K \end{pmatrix}
\]
Norm interpretations

- The $H_2$-norm:
  - for SISO systems, the induced norm from $l_2$ to $l_\infty$
  - the square root of the average power (is RMS value or power-norm) of the response to a white input signal of unit spectral density or the spectrum/power gain.
  - the square root of the energy contained in the impulse response

- The $H_\infty$-norm:
  - the induced norm from $l_2$ to $l_2$
  - the power/power gain (RMS)
  - the spectrum/spectrum gain
  - an upper bound on the $l_\infty$/power gain, assuming that the input is a persistent sinusoidal signal
  - the peak amplitude of the Bode singular value plot
Control / Objectives / Optimisation Constraints

- **i2p (Impulse to peak)**
  Influence du vent sur l’incidence

- **$H_\infty$**
  Sur la fonction de sensibilité pour les marges de stabilité

- **$H_2$**
  Influence des bruit et Réduction de consommation

- **Filtrage**
  Contrôle des modes avec réglage séparé

---

Control / Objectives / Optimisation Constraints - LMI

Some answers... with drawbacks

- Conservative solution
- Convex optimization
- Full order controller that needs truncature or/and a posteriori stucturation
- Fragility of the solution
Adding a structural constraint

Structure is good for

- gains interpolation for LPV systems
- interpretation for physical behavior
- implementation in embedded system (example of PID next slide)
Focus on implementation

Full order controller:
- \( n = \text{sum of the orders of all of systems} \)
- Ordinary differential equation of order \( n \)
- Implementation in embedded system

PID pseudo code:
```
% Control algorithm - main loop
while (running)
{
    r=adin(ch1) % read setpoint from ch1
    y=adin(ch2) % read process variable from ch2
    P=kp*(r-y) % compute proportional part
    D=ad*D+bd*(y-yold) % compute derivative part
    v=P+I+D % compute temporary output
    u=sat(v,ulow,uhigh) % simulate actuator saturation
    daout(ch1) % set analog output ch1
    I=I+bi*(r-y)+br*(u-v) % update integral
    yold=y % update old process output
    sleep(h) % wait until next update interval
}
```
Is there any alternative to tackle the drawbacks?

- make the implementation easy
- keep the constraints formulation for the engineer needs

Yes

- a posteriori structuration with the risk of lack of performance (order reduction, Youla parameter, etc...)
- Choose a structure for the controller (a PID for example)
- Use Optimization... but nonconvex one
Motivation

Observations:

- In Automatic, Robotic, Electronic or Mechanic, engineers know very well their problems.  
  \[\implies\] 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, ...  
  \[\implies\] Numerical Sense

Physical Solutions $\iff$ Numerical Solutions

\[\implies\] Goal: Propose optimization tools to build the best solver for their own problems.
Global Optimization

Definition: Contractor

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

The operator \( C_K : \mathbb{R}^n \rightarrow \mathbb{R}^n \) is a contractor for \( K \) if:

\[
\forall x \in \mathbb{R}^n, \begin{cases} 
C_K(x) \subseteq x, \\
C_K(x) \cap K = x \cap K.
\end{cases}
\]

(contractance)
(completeness)

Example: Forward-Backward Algorithm

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

\[
\forall x \in \mathbb{R}^n, \begin{cases} 
C(x) \subseteq x, \\
\text{and } f(x) = 0 \Rightarrow x \in C(x).
\end{cases}
\]
General Design

\[(\tilde{x}, \tilde{f}) = \text{OptimCtc} \left( [x], C_{out}, C_{in}, f_{cost} \right):\]

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

**infeasible Region**

Region removed by $C_{out}$

Feasible Region
Illustration: $C_{in}, C_{out}$

infeasible Region

Region removed by $C_{out}$

Region removed by $C_{in}$

Feasible Region
The feasibility 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 $x \in K$ \iff Prove that $x \notin \overline{K}$
The feasability test

Without equation or system,

**How to prove that a point is a feasible point?**

Prove that $x \in K$ $\Leftrightarrow$ $x \notin K$

$x$ is contracted by $C_{in}$ $\Leftrightarrow$ $x \in K$ $\Leftrightarrow$ $C_{out}$ proves that $x$ is in $K$.

$C_{in}$ will eliminate all the part of a box which are not in $\overline{K}$.
$C_{out}$ will eliminate all the part of a box which are not in $\overline{K}$. 
Global Optimization based on Contractor

- $\mathcal{L} := \{(x, false)\}$, The boolean indicate if $x$ is entirely feasible
- Do
  1. Extract from $\mathcal{L}$ a element $(z, b)$,
  2. Bisect $z$ following a bisector $B$: $(z_1, z_2)$
  3. for $j = 1$ to $2$ :
      - if $b = false$ (i.e. $x$ is not completly feasible) then
        Contract the infeasible region using $C_{out}$ and $C_f$,
        Extract $z_{feas}$ a feasible part of $z_j$ using $C_{in}$,
        Insert $(z_{feas}, true)$ in $\mathcal{L}$.
        Insert the rest $(z_j, false)$ in $\mathcal{L}$.
      - else (i.e. $x$ is entirely feasible)
        Contract $z_j$ using $C_f$,
        Try to find a local optimum without constraint in $[z_j]$,
        if succeed then Update $\tilde{f}$ insert $(z_j, true)$ in $\mathcal{L}$.
- stopping criterion
Global Optimization based on Contractor

\[ \mathcal{L} := \{(x, \text{false})\} \] The boolean indicate if \( x \) is entirely feasible

Do

1. Extract from \( \mathcal{L} \) a element \((z, b)\),
2. Bisect \( z \) following a bisector \( B: (z_1, z_2) \)
3. for \( j = 1 \) to \( 2 \):
   - if \( b = \text{false} \) (i.e. \( x \) is not completely feasible) then
     - Contract the infeasible region using \( C_{out} \) and \( C_f \),
     - Extract \( z_{feas} \) a feasible part of \( z_j \) using \( C_{in} \),
     - Insert \((z_{feas}, \text{true})\) in \( \mathcal{L} \).
     - Insert the rest \((z_j, \text{false})\) in \( \mathcal{L} \).
   - else (i.e. \( x \) is entirely feasible)
     - Contract \( z_j \) using \( C_f \),
     - Try to find a local optimum without constraint in \([z_j], \)
     - if succeed then Update \( \tilde{f} \) insert \((z_j, \text{true})\) in \( \mathcal{L} \).

- stopping criterion
Global Optimization based on Contractor

- \( \mathcal{L} := \{(x, \text{false})\} \), The boolean indicate if \( x \) is entirely feasible
- Do
  1. Extract from \( \mathcal{L} \) a element \((z, b)\),
  2. Bisect \( z \) following a bisector \( B: (z_1, z_2) \)
  3. for \( j = 1 \) to \( 2 \):
     - if \( b = \text{false} \) (i.e. \( x \) is not completely feasible) then
       Contract the infeasible region using \( C_{\text{out}} \) and \( C_f \),
       Extract \( z_{\text{feas}} \) a feasible part of \( z_j \) using \( C_{\text{in}} \),
       Insert \((z_{\text{feas}}, \text{true})\) in \( \mathcal{L} \).
       Insert the rest \((z_j, \text{false})\) in \( \mathcal{L} \).
     - else (i.e. \( x \) is entirely feasible)
       Contract \( z_j \) using \( C_f \),
       Try to find a local optimum without constraint in \([z_j],\)
       if succeed then Update \( \tilde{f} \) insert \((z_j, \text{true})\) in \( \mathcal{L} \).
- stopping criterion
General pattern for global optimization

Global Optimization based on Contractor

\[ \mathcal{L} := \{(x, false)\}, \text{ The boolean indicate if } x \text{ is entirely feasible} \]

Do

1. Extract from \( \mathcal{L} \) a element \((z, b)\),
2. Bisect \( z \) following a bisector \( B: (z_1, z_2) \)
3. for \( j = 1 \) to \( 2 \):
   - if \( b = false \) (i.e. \( x \) is not completely feasible) then
     Contract the infeasible region using \( C_{out} \) and \( C_f \),
     Extract \( z_{feas} \) a feasible part of \( z_j \) using \( C_{in} \),
     Insert \((z_{feas}, true)\) in \( \mathcal{L} \).
     Insert the rest \((z_j, false)\) in \( \mathcal{L} \).
   - else (i.e. \( x \) is entirely feasible)
     Contract \( z_j \) using \( C_f \),
     Try to find a local optimum without constraint in \( [z_j] \),
     if succeed then Update \( \tilde{f} \) insert \((z_j, true)\) in \( \mathcal{L} \).

stopping criterion
**$H_\infty$ control synthesis** under structural constraints

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

\[
\|P\|_\infty = \sup_{\omega} (\sigma_{\text{max}}(P(j\omega)))
\]

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

\[
\begin{align*}
\min_{k, \gamma} & \quad \gamma \\
\forall \omega, \quad & \left\| \frac{W_1(j\omega)}{1 + G(j\omega)K(j\omega)} \right\|_{\infty} \leq \gamma, \\
\forall \omega, \quad & \left\| \frac{W_2(j\omega)K(j\omega)}{1 + G(j\omega)K(j\omega)} \right\|_{\infty} \leq \gamma,
\end{align*}
\]

The closed-loop system must be stable.

Stability:

The system is stable iff its poles are strictly negative.

\[\iff \quad \text{The roots of the denominator of } \frac{1}{1 + G(s)K(s)} \text{ are strictly negative} \]

\[\implies \text{Routh-Hurwitz stability criterion}\]
### Routh-Hurwitz stability criterion

The Routh-Hurwitz stability criterion is used to determine the stability of a control system. The characteristic polynomial is given by:

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

The criterion states that if all the values in the first column of the Routh array are positive, all roots of \( P(s) \) are negative.

The Routh array is constructed as follows:

| \( v_{1,1} = a_n \) | \( v_{1,2} = a_{n-2} \) | \( v_{1,3} = a_{n-4} \) | \( v_{1,4} = a_{n-6} \) |
|\( v_{2,1} = a_{n-1} \)| \( v_{2,2} = a_{n-3} \)| \( v_{2,3} = a_{n-5} \)| \( v_{2,4} = a_{n-7} \) |
| \( v_{3,1} \) | \( v_{3,2} \) | \( v_{3,3} \) | \( v_{3,4} \) |
| \( v_{4,1} \) | \( v_{4,2} \) | \( v_{4,3} \) | \( v_{4,4} \) |
| \( v_{5,1} \) | \( v_{5,2} \) | \( v_{5,3} \) | \( v_{5,4} \) |

Continued...
Routh-Hurwitz stability criterion

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

\[
\begin{array}{c|c|c|c|c}
  v_{1,1} &= a_n & v_{1,1} &= a_{n-1} & v_{1,1} &= a_{n-2} & v_{1,1} &= a_{n-4} & v_{1,1} &= a_{n-6} \\
  v_{2,1} &= a_{n-1} & v_{2,1} &= a_{n-1} & v_{2,1} &= a_{n-3} & v_{2,1} &= a_{n-5} & v_{2,1} &= a_{n-7} \\
  v_{3,1} &= -1 & v_{3,1} &= a_{n-3} & v_{3,1} &= a_{n-5} & v_{3,1} &= a_{n-7} & \cdots & \cdots & \cdots \\
  v_{4,1} &= -1 & v_{4,1} &= a_{n-2} & v_{4,1} &= a_{n-4} & v_{4,1} &= a_{n-6} & \cdots & \cdots & \cdots \\
  v_{5,1} &= -1 & v_{5,1} &= a_{n-1} & v_{5,1} &= a_{n-3} & v_{5,1} &= a_{n-5} & v_{5,1} &= a_{n-7} & \cdots & \cdots & \cdots \\
  \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
\end{array}
\]

If all the value of the \textbf{first column} are positive, all roots of \( P \) are negative.
Definition of the feasible set

\[ \mathbb{K}_1^\omega = \left\{ (k, \gamma, \omega) : \left\| \frac{W_1(i\omega)}{1+G(i\omega)K(i\omega)} \right\|_\infty \leq \gamma \right\}, \]

\[ \mathbb{K}_2^\omega = \left\{ (k, \gamma, \omega) : \left\| \frac{W_2(i\omega)K(i\omega)}{1+G(i\omega)K(i\omega)} \right\|_\infty \leq \gamma \right\}, \]

\[ \mathbb{K}_4 = \bigcap_{\omega \in [10^{-2}, 10^2]} \mathbb{K}_1^\omega \cap \mathbb{K}_2^\omega. \]

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

\[ \mathbb{K}^{Routh} = \left\{ (k, \gamma) : \begin{align*} a_n(k, \gamma) &> 0, \\ a_{n-1}(k, \gamma) &> 0, \\ v_{2,1}(k, \gamma) &> 0, \\ \ldots \end{align*} \right\}. \]

The feasible set of our problem is \( \mathbb{K} = \mathbb{K}_4 \cap \mathbb{K}^{Routh} \).
Contractor Modeling: Properties

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

**Intersection, Composition**

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

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

**Union**

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

$$\{ x \in \mathbb{R}^n : f(x) = 0 \text{ OR } g(x) = 0 \}$$
Contractor with Quantifiers

Let $C$ be a contractor for a set $Z = X \times Y$, $\pi_X$ the projection of $Z$ over $X$.

**Contractor ForAll / Exists**

\[
\begin{align*}
C \cap Y(x) &= \bigcap_{y \in Y} \pi_X (C(x \times \{y\})) , \\
C \cup Y(x) &= \bigcup_{y \in Y} \pi_X (C(x \times \{y\})).
\end{align*}
\]

**Property**

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

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

\[ C_Z(X \times \{y_i\}) \]

\[ C_X(X) \]

\[ Y \]

\[ X \]

\[ y_1 \]

\[ y_2 \]

\[ y_3 \]
Application to $H_\infty$ control

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

\[ Y = Y_1 \cup Y_2 \cup Y_3 \cup Y_4 \]

\[ X \]

\[ C_Z(X \times Y_i) \]

\[ C_X(X) \]
Construction of Contractors $C_{out}$ of the feasible set $\mathcal{K}$

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

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

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

\[
\mathcal{K} = \left( \bigcap_{\omega \in [10^{-2}, 10^2]} \mathcal{K}^1_\omega \cap \mathcal{K}^2_\omega \right) \cap \mathcal{K}_{Routh}.
\]
Construction of Contractors $C_{out}$ of the feasible set $K$.

$C_{out}$ will eliminate all the part of a box which are not in $K$.

$$K = \left( \bigcap_{\omega \in [10^{-2}, 10^2]} K^1_\omega \cap K^2_\omega \right) \cap K^{Routh}. $$
Construction of Contractors $C_{out}$ of the feasible set $K$

$C_{out}$ will eliminate all the part of a box which are not in $K$.

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

1. Create the contractor $C_1$, $C_2$ and $C_{Routh}$ based on $K_\omega^1$, $K_\omega^2$ and $K_{Routh}$:

   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, $\ldots$
Construction of Contractors $C_{out}$ of the feasible set $K$

$C_{out}$ will eliminate all the part of a box which are not in $K$.

$$K = \left( \bigcap_{\omega \in [10^{-2}, 10^2]} K_1^\omega \cap K_2^\omega \right) \cap K^{Routh}.$$

1. Create the contractor $C_1$, $C_2$ and $C_{Routh}$ based on $K_1^\omega$, $K_2^\omega$ and $K^{Routh}$:

   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ... 

2. Inter: $C_3(k, \gamma, \omega) = C_1(k, \gamma, \omega) \cap C_2(k, \gamma, \omega)$. 
Construction of Contractors $C_{out}$ of the feasible set $K$

$C_{out}$ will eliminate all the part of a box which are not in $K$.

$$K = \left( \bigcap_{\omega \in [10^{-2}, 10^2]} K_1^\omega \cap K_2^\omega \right) \cap K_{Routh}.$$  

1. Create the contractor $C_1$, $C_2$ and $C_{Routh}$ based on $K_1^\omega$, $K_2^\omega$ and $K_{Routh}$:

   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...

2. Inter: $C_3(k, \gamma, \omega) = C_1(k, \gamma, \omega) \cap C_2(k, \gamma, \omega)$.

3. CtcForAll: $C_{\cap \omega}(k, \gamma) = \bigcap_{\omega \in [10^{-2}, 10^2]} C_3(k, \gamma, \omega)$.  

B. Clement

Control for Robots

SHARC 2016

Context

Control issue

Problem Statement

Some answers... with drawbacks

Optimisation based control

Global Optimization

General pattern for global optimization

Application to $H_\infty$ control

AUV Control Application

CISCREA: description and challenges

Robust control

Results

Conclusions
Construction of Contractors $C_{out}$ of the feasible set $K$

$C_{out}$ will eliminate all the part of a box which are not in $K$.

$$K = \left( \bigcap_{\omega \in [10^{-2}, 10^2]} K_1^\omega \cap K_2^\omega \right) \cap K_{Routh}.$$  

1. Create the contractor $C_1$, $C_2$ and $C_{Routh}$ based on $K_1^\omega$, $K_2^\omega$ and $K_{Routh}$:
   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, . . .

2. Inter: $C_3(k, \gamma, \omega) = C_1(k, \gamma, \omega) \cap C_2(k, \gamma, \omega)$.

3. CtcForAll: $C^{\omega}(k, \gamma) = \bigcap_{\omega \in [10^{-2}, 10^2]} C_3(k, \gamma, \omega)$.

4. Inter: $C_{out} = C^{\omega} \cap C_{Routh}$. 
Construction of Contractors $\mathcal{C}_{in}$ of the unfeasible set

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

\[
\overline{\mathcal{K}}_1^1 = \left\{ (k, \gamma, \omega) : \left\| \frac{W_1(i\omega)}{1+G(i\omega)K(i\omega)} \right\| > \gamma \right\},
\]

\[
\overline{\mathcal{K}}_2^2 = \left\{ (k, \gamma, \omega) : \left\| \frac{W_2(i\omega)K(i\omega)}{1+G(i\omega)K(i\omega)} \right\| > \gamma \right\},
\]

\[
\overline{\mathcal{K}} = \bigcup_{\omega \in [10^{-2}, 10^2]} \overline{\mathcal{K}}_1^1 \cup \overline{\mathcal{K}}_2^2 \cup \overline{\mathcal{K}}_{Routh}.
\]
Construction of Contractors $C_{in}$ of the unfeasible set

$C_{in}$ will eliminate all the part of a box which are not in $\overline{K}$.

$$\overline{K} = \left( \bigcup_{\omega \in [10^{-2}, 10^2]} \overline{K}^1_{\omega} \cup \overline{K}^2_{\omega} \right) \cup \overline{K}^{Routh}.$$
Construction of Contractors $C_{in}$ of the unfeasible set

$C_{in}$ will eliminate all the part of a box which are not in $\overline{K}$.

$$\overline{K} = \left( \bigcup_{\omega \in [10^{-2},10^2]} \overline{K_1} \cup \overline{K_2} \right) \cup \overline{K_{Routh}}.$$ 

1. Create the contractor $C_1$, $C_2$ and $C_{Routh}$ based on $\overline{K_1}$, $\overline{K_2}$ and $\overline{K_{Routh}}$:
   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...
Construction of Contractors $C_{in}$ of the unfeasible set

$C_{in}$ will eliminate all the part of a box which are not in $\overline{K}$.

\[
\overline{K} = \left( \bigcup_{\omega \in [10^{-2}, 10^2]} \overline{K}_1^\omega \cup \overline{K}_2^\omega \right) \cup \overline{K}_{Routh}.
\]

1. Create the contractor $C_1$, $C_2$ and $C_{Routh}$ based on $\overline{K}_1^\omega$, $\overline{K}_2^\omega$ and $\overline{K}_{Routh}$:
   - Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...
2. Union: $C_3(k, \gamma, \omega) = C_1(k, \gamma, \omega) \cup C_2(k, \gamma, \omega)$. 

---

Control for Robots

SHARC 2016

Context

Control issue

Problem Statement

Some answers... with drawbacks

Optimisation based control

Global Optimization

General pattern for global optimization

Application to $H_\infty$ control

AUV Control Application

CISCREA: description and challenges

Robust control

Results

Conclusions
Construction of Contractors $C_{in}$ of the unfeasible set

$C_{in}$ will eliminate all the part of a box which are not in $\overline{K}$.

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

1. Create the contractor $C_{1,\omega}^{1}$, $C_{2,\omega}^{2}$ and $C_{Routh}$ based on $\overline{K}_{1,\omega}^{1}$, $\overline{K}_{2,\omega}^{2}$ and $\overline{K}_{Routh}$:

   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...

2. Union: $C_{3,\omega}^{3}(k, \gamma, \omega) = C_{1,\omega}^{1}(k, \gamma, \omega) \cup C_{2,\omega}^{2}(k, \gamma, \omega)$.

3. CtcExist: $C_{\omega}^{\cup}(k, \gamma) = \bigcup_{\omega \in [10^{-2}, 10^2]} C_{3,\omega}^{3}(k, \gamma, \omega)$. 

Construction of Contractors $C_{in}$ of the unfeasible set

$C_{in}$ will eliminate all the part of a box which are not in  $\overline{K}$.

$$\overline{K} = \left( \bigcup_{\omega \in [10^{-2}, 10^2]} \overline{K}_1^\omega \cup \overline{K}_2^\omega \right) \cup \overline{K}_{Routh}.$$ 

1. Create the contractor $C_1$, $C_2$ and $C_{Routh}$ based on $\overline{K}_1^\omega$, $\overline{K}_2^\omega$ and $\overline{K}_{Routh}$:

   Contractor based on inequality system: Forward-Backward algorithm, HC4-revise, PolytopeHull, ...

2. Union: $C_3(k, \gamma, \omega) = C_1(k, \gamma, \omega) \cup C_2(k, \gamma, \omega)$.

3. CtcExist: $C \cup \omega(k, \gamma) = \bigcup_{\omega \in [10^{-2}, 10^2]} C_3(k, \gamma, \omega)$.

4. Union: $C_{in} = C \cup \omega \cup C_{Routh}$. 

\[ \text{B. Clement} \]
First Application with second order dynamic system

The transfer function of the dynamic system:

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

\[ W_1(s) = \frac{s + 100}{100s + 1}, \quad W_2(s) = \frac{10s + 1}{s + 10}. \]
Overview of the equation

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

\[ \iff \]

\[ \forall \omega \in [10^{-2}, 10^2], \frac{w^2(w^2+1.0)(w^2+10000.0)(25.0w^4-1.0w^2+25.0)}{(10000.0w^2+1.0)f_1(k,\gamma,\omega)} \leq \gamma \]
Overview of the equation

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

\[ \iff \]

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

\[ f_1(k, \gamma, \omega) = 25.0 kd^2 w^4 + 25.0 kp^2 w^2 + 25.0 kp^2 w^4 - 1.0 ki (50.0 kd w^2 + 70.0 w^2 + 70.0 w^4) + ki^2 (25.0 w^2 + 25.0) + 120.0 kd w^4 - 50.0 kd w^6 + 50.0 kp w^2 - 50.0 kp w^6 + 25.0 w^2 + 24.0 w^4 + 24.0 w^6 + 25.0 w^8 + 50.0 kd k p w^4 \]
Control for Robots

SHARC 2016

Context

Control issue

Problem Statement

Some answers... with drawbacks

Optimisation based control

Global Optimization

General pattern for global optimization

Application to $H_\infty$ control

AUV Control Application

CISCREA: description and challenges

Robust control

Results

Conclusions

Overview of the equation

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

$$\iff$$

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

$$f_1(k, \gamma, \omega) = 25.0 kd^2 w^4 + 25.0 kp^2 w^2 + 25.0 kp^2 w^4 - 1.0 ki (50.0 kd w^2 + 70.0 w^2 + 70.0 w^4) + ki^2 (25.0 w^2 + 25.0) + 120.0 kd w^4 - 50.0 kd w^6 + 50.0 kp w^2 - 50.0 kp w^6 + 25.0 w^2 + 24.0 w^4 + 24.0 w^6 + 25.0 w^8 + 50.0 kd kp w^4$$

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

The same problem is proposed with 2 existant tools and compared with the new approach.

1. **HINF SYN** of Matlab - full order controller with convex optimization based on LMI ($\gamma = 1.5887$);

2. **HINFSTRUCT** of Matlab - structured controller with local optimization ($\gamma = 2.1414$).

3. **Global Optimization of IBEX** ($\gamma = 2.1414$)
Results with **HINFSYN** of Matlab

\[ \gamma = 1.5887 \]
\[ \gamma = 2.1414 \]
Results with Global Optimization of IBEX

\[ \gamma = 2.1414 \]

\( \rightarrow \) same result as with \textsc{Hinfstruct}, but with a global optimality proof!
Results with Global Optimization of IBEX

\[ \gamma = 2.1414 \]

\[ \longrightarrow \text{same result as with HINFSTRUCT, but with a global optimality proof!} \]
To keep in mind

**Contractor Programming:**

- Generates the Modeling 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.
### AUV CISCREA

<table>
<thead>
<tr>
<th><strong>Size</strong></th>
<th>0.525m (L) 0.406m (W) 0.395m (H)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Weight in air</strong></td>
<td>15.56kg (without payload and floats)</td>
</tr>
<tr>
<td><strong>Degrees of Freedom</strong></td>
<td>Surge, Sway, Heave and Yaw</td>
</tr>
<tr>
<td><strong>Propulsion</strong></td>
<td>2 vertical and 4 horizontal propellers</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td>2 knots (Surge) and 1 knot (Sway, Heave)</td>
</tr>
<tr>
<td><strong>Depth Rating</strong></td>
<td>50m</td>
</tr>
<tr>
<td><strong>On-board Battery</strong></td>
<td>2-4 hours</td>
</tr>
</tbody>
</table>
AUV CISCREA model

Rigid-body dynamic:

\[ M_{RB} \dot{\nu} + C_{RB}(\nu)\nu = \tau_{env} + \tau_{hydro} + \tau_{pro} \]  \hspace{1cm} (1)

Hydrodynamic formulations:

\[ \tau_{hydro} = -M_{A}\dot{\nu} - C_{A}(\nu)\nu - D(|\nu|)\nu - g(\eta) \]  \hspace{1cm} (2)

Damping:

\[ D(|\nu|) = D_{L} + D_{N}|\nu|\nu \]  \hspace{1cm} (3)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_{RB} )</td>
<td>AUV rigid-body mass and inertia matrix</td>
</tr>
<tr>
<td>( M_{A} )</td>
<td>Added mass matrix</td>
</tr>
<tr>
<td>( C_{RB} )</td>
<td>Rigid-body induced coriolis-centripetal matrix</td>
</tr>
<tr>
<td>( C_{A} )</td>
<td>Added mass induced coriolis-centripetal matrix</td>
</tr>
<tr>
<td>( D(</td>
<td>\nu</td>
</tr>
<tr>
<td>( g(\eta) )</td>
<td>Restoring forces and moments vector</td>
</tr>
<tr>
<td>( \tau_{env} )</td>
<td>Environmental disturbances (wind, waves and currents)</td>
</tr>
<tr>
<td>( \tau_{hydro} )</td>
<td>Vector of hydrodynamic forces and moments</td>
</tr>
<tr>
<td>( \tau_{pro} )</td>
<td>Propeller forces and moments vector</td>
</tr>
</tbody>
</table>
We consider that there are no dependencies between the yaw dynamic and dynamics along other axis. Resulting Yaw dynamic:

\[(I_{YRB} + I_{YA})\ddot{x} + D_{YN}|\dot{x}|\dot{x} + D_{YL}\dot{x} = K_t \tau_i\]  \hspace{1cm} (4)

However, $H_\infty$ synthesis requires a linear system. Thus, the CISCREA yaw model could be linearized as:

\[(I_{YRB} + I_{YA})\ddot{x} + (D_{YLA} + \delta)\dot{x} = K_t \tau_i,\]  \hspace{1cm} (5)
Control objectives specifications

We aim to synthesize a controller to meet the following objectives:

1. Small tracking error \( e \).
2. External perturbation rejection.

External perturbation can be modeled as a control disturbance signal \( d \).
\( H_\infty \) formulation

The two objectives can be formulate as \( H_\infty \) constraints:

1. Small tracking error:

   \[
   \frac{|e(i\omega)|}{|r(i\omega)|} \leq |W_e^{-1}(i\omega)| \iff \| T_{r \to e}(i\omega) W_e(i\omega) \|_\infty \leq 1
   \]

2. External perturbation rejection:

   \[
   \frac{|y(i\omega)|}{|d(i\omega)|} \leq |W_y^{-1}(i\omega)| \iff \| T_{d \to y}(i\omega) W_y(i\omega) \|_\infty \leq 1
   \]
Bode diagrams of Weighted functions

Bode Diagrams

- \( G \)
- \( W_e^{-1} \)
- \( W_d^{-1} \)

Magnitude (dB) vs. Frequency (rad/s)

Control for Robots

Optimisation based control

AUV Control Application

Conclusions
Min Max Problem

- The controller $K(k, s)$ depends on free parameters $k$.
- $T_{r\rightarrow e}(k, s) = \frac{1}{1+G(s)K(k, s)}$ depends on $k$
- $T_{d\rightarrow y}(k, s) = \frac{G(s)}{1+G(s)K(k, s)}$ depends on $k$

The constraint satisfaction problem is:

Find $k$, $\max(\|T_{r\rightarrow e}(k, s)W_e(s)\|_\infty, \|T_{d\rightarrow y}(k, s)W_y(s)\|_\infty) \leq 1$

- $\|T(s)\|_\infty = \sup_{\omega} |T(i\omega)|$

The Min Max problem is:

$$\min \sup_k \{\max_{\omega \geq 0} (\|T_{r\rightarrow e}(k, s)W_y(s)\|, \|T_{d\rightarrow y}(k, s)W_y(s)\|)\}$$
Solving the Min Max problem

We solve the Min Max problem with Global optimization based on interval analysis.

- Existing methods are based on local optimization. They only provide an upper bound of the objective function.
- Global optimization provides an enclosure of the objective function. It is possible to prove that the CSP (Constraint Satisfaction Problem) is not feasible.
Uncertainties

The model of the CISCREA carries uncertainties. The controller is synthesized from a nominal model, and robustness to uncertainties must be analyzed.

- An uncertainty is represented by an interval: \( p \) is the vector of uncertainties.
- \( G_\Delta(s, p) \), \( p \in \mathbb{P} \) describe the uncertain system.
- The closed loop system stability and performances are robust if and only if: \( \forall p \in \mathbb{P}, \)
  \[
  \max(|| T_{\Delta r \rightarrow e}(p, s)W_e(s)||_\infty, || T_{\Delta d \rightarrow y}(p, s)W_y(s)||_\infty) \leq 1
  \]
- The robustness condition can be validated with interval analysis in a reliable way.
Controller synthesis

- **PID controller:**
  \[ K(k, s) = k_p + \frac{k_i}{s} + \frac{k_d s}{1 + \tau s} \]
  
  - \( k = (k_p, k_i, k_d, \tau) \)

- **CISCREA model:**
  \[ G(s) = \frac{6.725}{s^2 + 2s} \]

- **Weighting functions:**
  \[ W_e = \frac{0.1s^2 + 0.7109s + 2.527}{s^2 + 0.2248s + 0.02527}, \quad W_y = \frac{0.1s + 0.9935}{s + 0.03142} \]

- \( k \) is searched in \([0, 2]^4\)
Solution to the Min Max problem computed:

\[ k^* = (1.987, 1.731, 0.638, 0.001) \]

\[ \| T_{r \rightarrow e}(k^*, s) \|_\infty = 0.325 \]

\[ \| T_{d \rightarrow y}(k^*, s) \|_\infty = 0.154 \]

\[ \min \sup_{k, \omega \geq 0} \{ \max(|T_{r \rightarrow e}(k, s)W_y(s)|, |T_{d \rightarrow y}(k, s)W_y(s)|) \} \in [0.225, 0.325] \]
Robustness Analysis

Uncertain CISCREA model:

\[ G_\Delta(s, p) = \frac{6.725}{s^2 + ps}, \quad p \in [0, 4] \]

- \[ \| T_{\Delta r \rightarrow e}(k^*, s, p) \|_\infty \leq 0.82 \]
- \[ \| T_{\Delta d \rightarrow y}(k^*, s, p) \|_\infty \leq 0.162 \]
- \[ \| T_{r \rightarrow e}(k^*, s) \|_\infty = 0.325 \]
- \[ \| T_{d \rightarrow y}(k^*, s) \|_\infty = 0.154 \]
Tracking error constraint

- Tracking error constraint graph
  - Magnitude (dB) vs. Pulsation (rad/s)
  - Two curves:
    - $W_e^{-1}$
    - $T_{re}$
Context

Control issue

Optimisation based control

AUV Control Application

Conclusions

Results

Control for Robots

SHARC 2016

Optimisation based control

Global Optimization

General pattern for global optimization

Application to $H_\infty$ control

AUV Control Application

CISCREA: description and challenges

Robust control

Conclusions

B. Clement

Control for Robots

Perturbation rejection

<table>
<thead>
<tr>
<th>Pulsation (rad/s)</th>
<th>Magnitude (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-3}$</td>
<td>-50</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>-40</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>-30</td>
</tr>
<tr>
<td>$10^{0}$</td>
<td>-20</td>
</tr>
<tr>
<td>$10^{1}$</td>
<td>-10</td>
</tr>
<tr>
<td>$10^{2}$</td>
<td>0</td>
</tr>
<tr>
<td>$10^{3}$</td>
<td>10</td>
</tr>
</tbody>
</table>

$W^{-1}_{y}$

$T_{d\rightarrow y}$

Perturbation Rejection

$P_{y}$
Step response without perturbation

![Step response graph](image-url)
Step response with perturbation

![Graph showing step response with perturbation]

**Yaw comparison**

- **Yaw optimized tuning [rad]**
- **Yaw Matlab tuning [rad]**
- **Target Yaw**

**Axes:**
- **Amplitude [rad]**: 0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6
- **Time [s]**: 0, 50, 100, 150, 200, 250, 300
Conclusion for the robot control

- Robust control synthesis method based on global optimization: the optimal PID

- Robustness analysis with respect to uncertainties with experiments on a real underwater robot
Conclusions

- Need: structured control based on end-used demand
- Answers: an original approach based on global optimization (change the hegemony of SPD)
- Perspectives: generalization of the concept for nonlinear control, temporal specifications, etc...
- Others applications:

![Image of a rocket taking off and a submarine in water]

B. Clement
These results are obtained with the collaboration of Jordan Ninin (Associate Professor), Dominique Monnet (PhD Student) and Juan Luis Rosendo (PhD Student)

- N. Imbert et B. Clement. Launcher Attitude control : some answers to the robustness issue. In Proc. of 16th IFAC Symposium on Automatic Control in Aerospace, Saint-Petersburg, 2004