AIC-Automatisation avancée, intelligence artificielle et cognitique

7. Commande multimodale

Jean-Daniel Dessimoz

HAUTE ÉCOLE D'INGÉNIERIE ET DE GESTION DU CANTON DE VAUD
www.heig-vd.ch

Institut d'Automatisation Industrielle
LaRA
Laboratoire de Robotique et Automatisation

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale

AIC-Automatisation avancée, intelligence artificielle et cognitique

Contenu

Contenu des *Exposés et exercices*

Notion de modèle ; métrique pour le traitement d'information et pour la cognitique	4p
Choix d'une structure de commande	2p
Intelligence artificielle et inférences bayésiennes	2p
Commande à logique floue	2p
Commande neuronale	2p
Commande multimodale	2p
Commande à algorithme génétique	2p
Robots mobiles autonomes et humanoïdes	4p
Réserve et contrôle continu (TE, corr.)	6p

Travaux de laboratoire associés

Estimation de grandeurs cognitives (essais en simulation avec programmes d'évitement d'obstacles)	L-AIC-1
Test d'intelligence artificielle selon Turing et utilisation d'Eliza	L-AIC-2
Commande neuronale	L-AIC-3
Commande à logique floue	L-AIC-4
Commande à algorithme génétique	L-AIC-5
Commande multimodale	L-AIC-6
Robot mobile autonome	L-AIC-7
Robot humanoïde NAO	L-AIC-8
Inférences bayésiennes	L-AIC-9
Sur demande, l'étudiant peut échanger l'une des manipulations ci-dessus par un autre sujet (cf. manipulations LaRA)	

AIC-Automatisation avancée, intelligence artificielle et cognitique

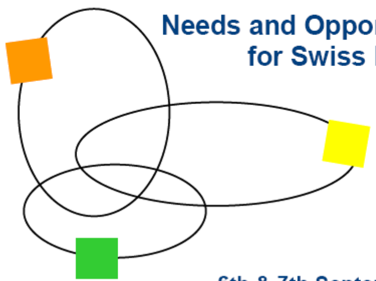
Contenu

- Introduction
- Notion de modèle ; métrique pour le traitement d'information et pour la cognitique
- Choix d'une structure de commande
- Intelligence artificielle et « machine learning »
- Commande à logique floue
- Commande neuronale, yc. « deep learning »
- **Commande multimodale**
- Commande à algorithme génétique
- Robots mobiles autonomes
- Robot humanoïde NAO
- Conclusion

Commande multimodale

IPLnet Integrated Production and Logistics WORKSHOP **2005**

Needs and Opportunities
for Swiss Industry



6th & 7th September 2005

Schloss Böttstein

Schlossweg 20, 5315 Böttstein, Switzerland

Commande multimodale

Largement basé sur l'article suivant:

Novel Modal Control for Strongly Non-stationary Systems

J.-D. Dessimoz[1], M. Dahiya[2], P.-F. Gauthey[1], A. Perrenoud[1]

[1]University of Applied Sciences of Western Switzerland

HES-SO / HEIG-VD / IAI, Yverdon-les-Bains

[2]Indian Institute of Technology, Kanpur, India.

Institut d' Automatisation Industrielle

Commande multimodale

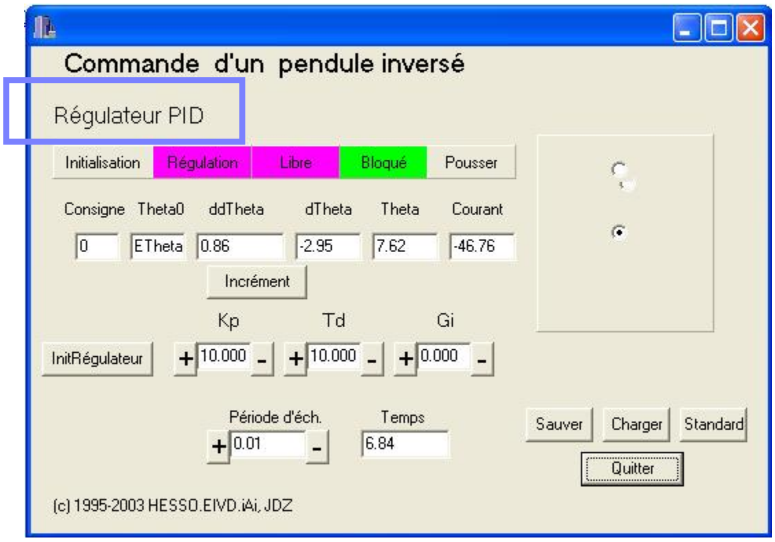
Content

Commande multimodale

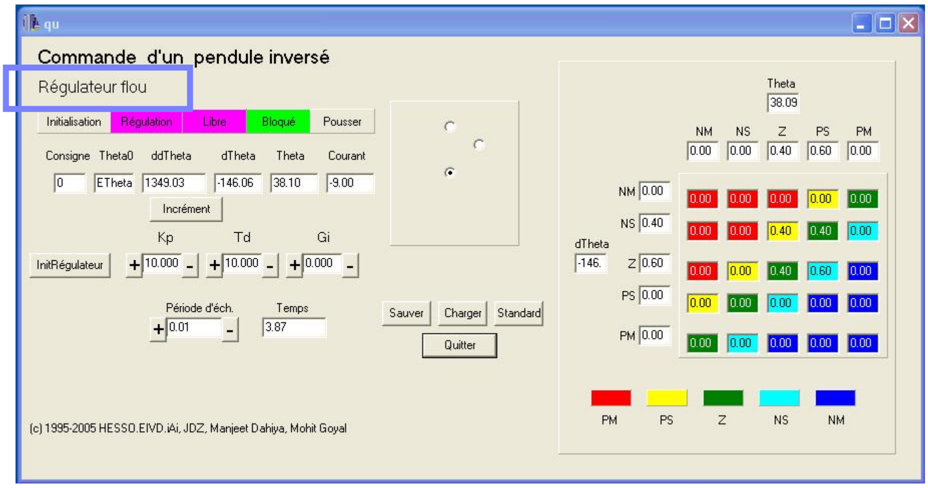
1. Introduction (1 of 2)

- **Classical control theories have been developed for many years.**
  - **Work well when systems are stationary and linear**
- **How to tackle non-stationary, non linear systems?**
- **Core idea:**
  - **divide the state space in smaller domains (local modes, as required by applications) where controllers can be individually, specifically designed by classical approach**
  - **Perform a feature-space based, dynamically changing, weighted average of all controller outputs**

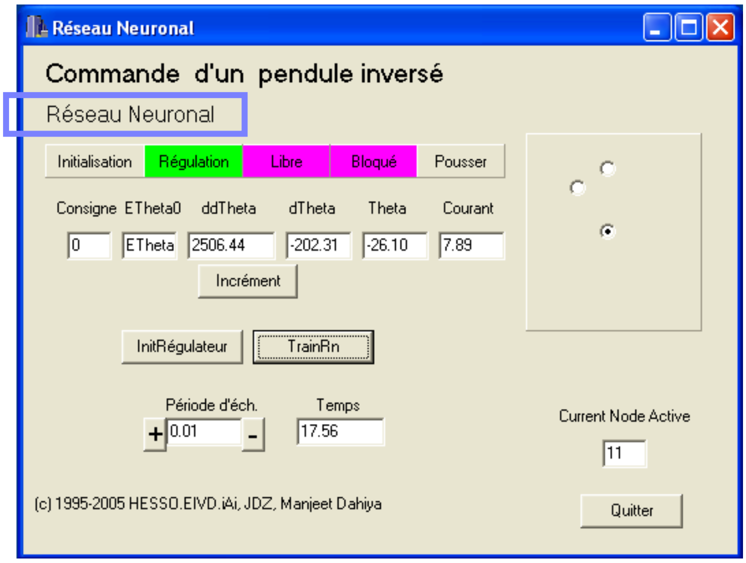
…PID Controller…



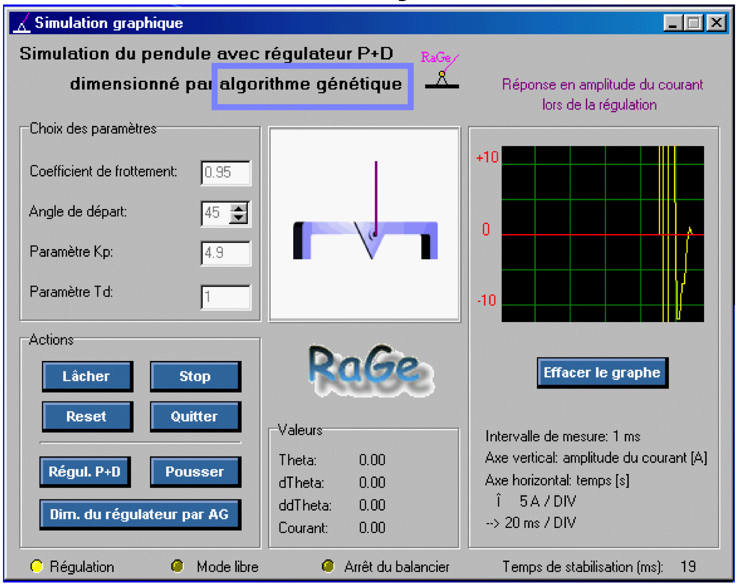
...Fuzzy Controller...



...Neural Controller...



...Evolutionary Controller...



Commande multimodale

1. Introduction (2 of 2)

- **Particular motivation behind this study**
  - **Printing machines at high speed (20 m/s) and high resolution (<0.02mm)**
  - **Robotized handling systems with large changes in inertia and high requirements in terms of dynamics and precision**

Commande multimodale

2. Multimodal Controller (1 of 2)

- **Principle, at design time**
  - **Divide the state space into domains sufficiently homogeneous, along one or more discriminant dimensions (features)**
  - **Design classic controllers in each domain**
    - **e.g. PID, Ziegler-Nichols method**
  - **Define weighting functions, typically distance dependant in feature space (e.g. w=1/(1+d))**

Commande multimodale

2. Multimodal Controller (2 of 2)

- **Procedure, in operation**
  - **"Simultaneously" evaluate the output commands of individual controllers in each domain**
  - **Evaluate instantaneous, local weight values, for all controllers**
  - **Average the weighted contributions from all controllers**

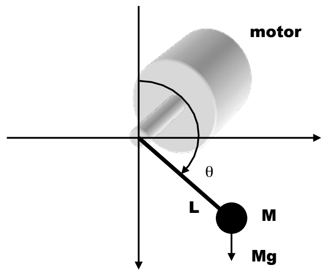
Mathematically:

$$U_m(t) = \frac{\sum_{n=1}^{N} w_n(t) \bullet U_n(t)}{\sum_{n=1}^{N} w_n(t)} = \frac{\sum_{n=1}^{N} w_n(v(t)) \bullet U_n(t)}{\sum_{n=1}^{N} w_n(v(t))}$$

Commande multimodale

3. Example of MMC (1 of 3)

Control of an inverted pendulum



$$J\ddot{\theta} = C_m + MgL\sin\theta$$

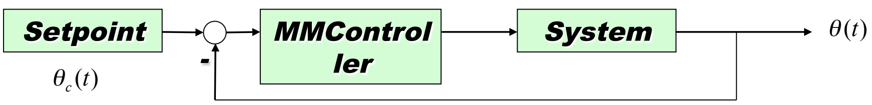
$$C_m = kI_m$$ ***Driving torque***

$$J = ML^2$$ ***Inertia of load***

Commande multimodale

3. Example of MMC (2 of 3)

Main components of the specific MMC



Error signal: $\varepsilon(t) = \theta_c(t) - \theta(t)$

-180°..180° : domain broken into 5 regions

Output of one elementary controller (e.g.PID with a priori term):

$$I_n(t) = K_{pn} \cdot \varepsilon(t) + G_{in}\int \varepsilon(t) + T_{dn}\frac{d\varepsilon(t)}{dt} + A_n$$

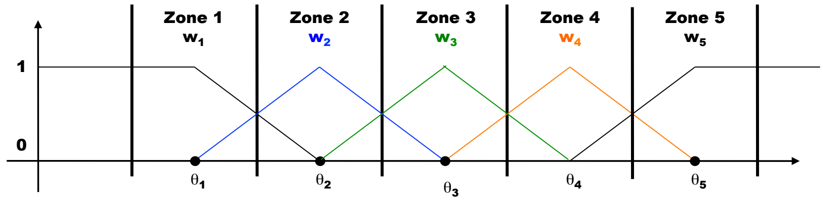
Weighted average of all controller outputs:

$$I_m(t) = \sum_{n=1}^{N} w_n(\theta) \bullet I_n(t) \Big/ \sum_{n=1}^{N} w_n(\theta)$$

Commande multimodale

3. Example of MMC (3 of 3)

Computation of weights (ex. linear, 1-dimensional)



$$For \quad \theta_n \le \theta \le \theta_{n+1} \quad \begin{cases} w_n = \dfrac{\theta_{n+1} - \theta}{\theta_{n+1} - \theta_n} \\ w_{n+1} = 1 - w_n \end{cases}$$

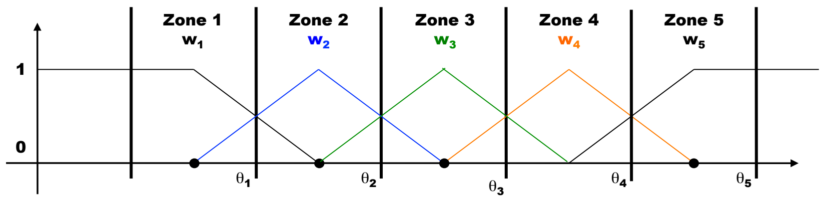
$$For \quad \theta \le \theta_1 \qquad w_1 = 1$$

$$For \quad \theta_N \le \theta \qquad w_N = 1$$

θ_1	*-135°*
θ_2	*-30°*
θ_3	*0°*
θ_4	*30 °*
θ_5	*135°*

Commande multimodale

3b. Remarque

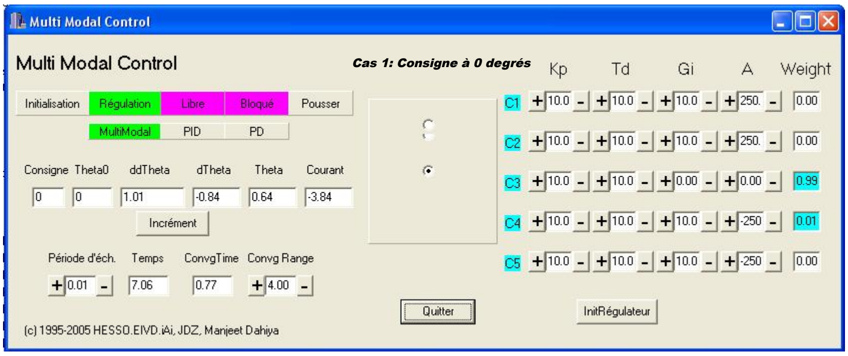


Il n’est pas nécessaire que la somme des poids vale 1 partout immédiatement. Si ce n’est pas le cas, on peut très simplement obtenir un nouveau jeu, normalisé, en divisant les premiers par la somme en chaque point.

Commande multimodale

4. Implementation and tests (1 of 3)

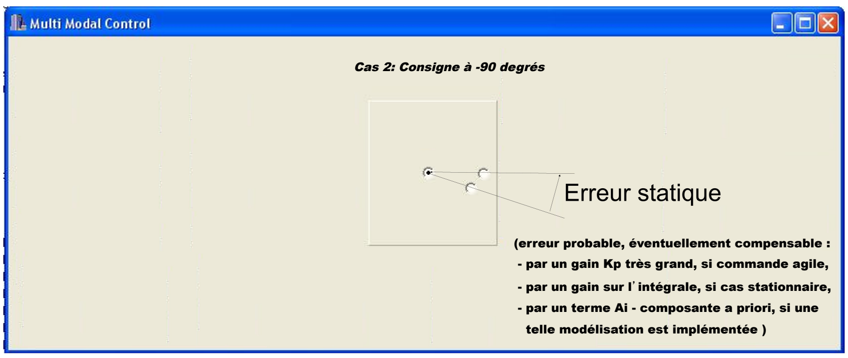
Graphical interface, C++ implementation



Commande multimodale

4. Implementation and tests (1 of 3)

Graphical interface, C++ implementation



4. Implementation and tests (2 of 3)

Benchmark with PD controllers

	Residual error (asymptotic value)	
Set point	***PD***	***Multi - PD***
-90°	*30.4°*	*9.8°*
-30°	*30.2°*	*<2°*
0°	*<2°*	*<2°*
30	*30.8°*	*<2°*
45°	*34.7°*	*<2°*
80°	*32.6°*	*10.3°*
90°	*30.4°*	*9.8°*
135°	*16.7°*	*<2°*

4. Implementation and tests (3 of 3)

Benchmark with PID controllers

	Convergence Times in seconds	
Set point	***PID***	***Multi - PID***
-90°	*217*	*7.7*
-30°	*112*	*8.5*
0°	*18*	*1.6*
30	*98*	*6.4*
45°	*133*	*5.2*
80°	*196*	*4.1*
90°	*207*	*3.9*
135°	*182*	*0.1*

Commande multimodale

5. Logiciel (1 de 3)

Extraits de code – Simulation d'un balancier

```
void UpdateBalancier(/*float Courant,
     float &Theta, float &dTheta, float &ddTheta*/)
 {
  ddTheta=360/Lm*g*sin(Theta/180*pi)+300*Courant;
  dTheta=0.97*dTheta+ddTheta*dt;
  Theta=Theta+dTheta*dt;
  t=t+dt;
  return;
  }
```


Commande multimodale

5. Logiciel (2 de 3)

Extraits de code – Régulateur PID avec composante a priori

```
float Controller1() // Calculer Courant
 {
  ConsigneTheta=StrToFloat(Form1->EConsigne->Text);
  Ecart1=ConsigneTheta-Theta;

  Integrale1=Integrale1+Ecart1*dt;
  Courant1=Kp1*Ecart1+Td1*(Ecart1-
              OldEcart1)/dt+Gi1*Integrale1+A1;

  OldEcart1=Ecart1;

  return Courant1;
}
```

Commande multimodale

5. Logiciel (3 de 3)

Extraits de code – Moyenne pondérée des régulateurs

```
{W1=weight1(Theta);
 W2=weight2(Theta);
 W3=weight3(Theta);
 W4=weight4(Theta);
 W5=weight5(Theta);
 weightTemp = W1+W2+W3+W4+W5;

 if (weightTemp==0)
 Courant= 0;

 else
 Courant=
(W1*(Controller1())+
 W2*(Controller2())+
 W3*(Controller3())+
 W4*(Controller4())+
 W5*(Controller5()))/(weightTemp) ;
}
```


Commande multimodale

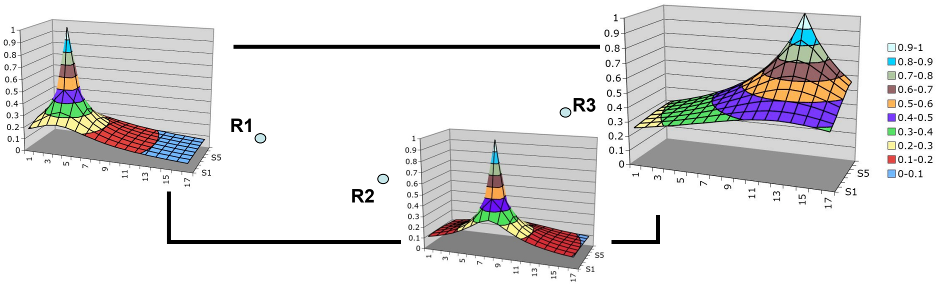
6. Exemple bidimensionnel

- **Amélioration de performances par régulateurs multiples**
- **Extrapolation de chaque régulateur Ri avec une surface de pondération en « cloche » (exemples: gaussiennes, ou ci-dessous fonctions de distances inverses Wi pour 3 régulateurs et un domaine 2-D, càd. à deux paramètres)**
- **Avec normalisation en tous points avec une division par la somme des poids des régulateurs en ces points**

$$w_i = \frac{1}{k_i \cdot (1 + d_i)}$$

$$d_i(x,y) = \sqrt{(x - R_{ix})^2 + (y - R_{iy})^2}$$

$$w_i' = \frac{w_i}{w_1 + w_2 + w_3}$$



7. Conclusion (1 of 2)

- Very well adapted for digital control
- Some light similarities with fuzzy controllers (weighting functions and membership functions, multiple components)
- Much more sophistication however in terms of working points (modes), elementary controller, feature space and weights (re. distances)

7. Conclusion (2 of 2)

- Fit to application specifications
- The test implementation proves that MMC can deliver much better performance figures than (single) classical controllers
- Addresses Swiss needs and opportunities: moderate increase of complexity (e.g. order of 10x computation load) for best possible results (leading quality)