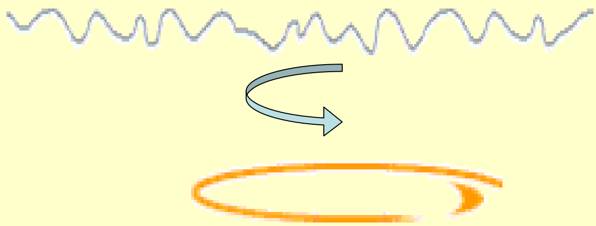
Mathématique et Automatique :

de la boucle ouverte à la boucle fermée



Maïtine bergounioux
Laboratoire MAPMO - UMR 6628
Université d'Orléans

Maitine.Bergounioux@labomath.univ-orleans.fr

Plan

1. Un peu de sémantique …

2. Dimension finie/ dimension infinie

3. Un peu d'optimisation en dimension finie

4. La commande optimale

5. Commande prédictive

1. Un peu de sémantique …

Contrôler: Vérifier (voir *examiner, inspecter, vérifier*) …
Maîtriser, dominer .

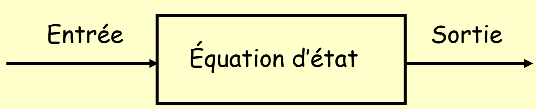
Commander : Exercer son autorité (voir *contraindre, obliger*) …
Prescrire de manière autoritaire …
Techn. Faire fonctionner :
ce mécanisme commande l'ouverture des portes

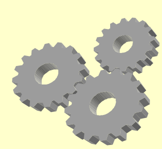
Contrôler ~ passif

Commander ~ actif

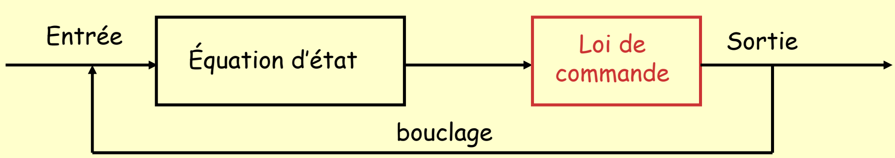
Boucle ouverte/ boucle fermée

Mathématicien → boucle ouverte





Automaticien → boucle fermée



2. Dimension finie / infinie

Système : **A : X** (Entrée) ⟶ **Y** (Sortie)

Outils de l'automatique : filtres, fonction de transfert

Contrôlabilité → Exacte
Contrôlabilité → Approchée

Et si on ne peut pas atteindre un objectif donné ?

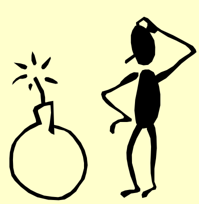
On fait « au mieux » ⟶ **Commande optimale**

3. Un peu d'optimisation en dimension finie

Dimension Infinie

Problème continu ⟶ Discrétisation (EF, DF, POD,…) ⟶ **Dimension Finie** **Problème discret**

Problème continu ⟶ **Algorithme continu** ⟶ **Solution continue**



Problème discret ⟶ **Algorithme discret** ⟶ **Solution discrète**

Solution continue ⟶ Discrétisation (EF, DF, POD,…) ⟶ **Solution discrète**

$$(P) \quad \begin{cases} \min \quad J(x) \\ g(x) \leq 0, \\ h(x) \leq 0, \\ x \in R^n \end{cases}$$

$$C = \{ x \in R^n \mid g(x) \leq 0, h(x) = 0 \}$$

x^* solution locale (resp. globale) de (P) si

$$\forall x \in B(x^*) \cap C \quad (resp. \forall x \in C) \quad J(x^*) \leq J(x)$$

Conditions de Kuhn-Karush-Tucker (KKT)

Si x^* solution de (P) alors

$$\exists \lambda^* = (\lambda_1, \ldots, \lambda_p) \in R^p \text{ et } \mu^* = (\mu_1, \ldots, \mu_q) \in R^q$$

$$\forall j \in \{1, \ldots, q\} \quad \mu_j \geq 0$$

$$h(x^*) = 0, g(x^*) \leq 0, \quad \text{(réalisabilité)}$$

$$\forall j \in \{1, \ldots, q\} \quad \mu_j g_j(x^*) = 0, \quad \text{(complémentarité)}$$

$$\nabla L(x^*, \lambda^*, \mu^*) = 0, \text{où } L(x, \lambda, \mu) = J(x) + \sum_{i=1}^{p} \lambda_i h_i(x) + \sum_{j=1}^{q} \mu_j g_j(x)$$

$L(x, \lambda, \mu)$ est le **lagrangien** du problème

4. La commande optimale

4.1 principe

On se donne :

- Un espace de contrôles (ou commandes) : U
- Un système dynamique décrit par son **équation d'état** :

$$x = F\,(u,t)$$

état

Espace d'état

- Des contraintes sur le contrôle : $u \in U_{ad} \subseteq U$
- Eventuellement des contraintes sur l'état : $x \in K \subseteq X$
- Une fonctionnelle coût (ou objectif) : J

L'équation d'état peut être

- Une équation aux différences (espace d'état= espace de suites)
- Une ou des équations différentielles (espace d'état = espace de fonctions à valeurs dans R^n: dimension finie) linéaires ou pas
- Une ou des équations aux dérivées partielles (espace d'état = espace de fonctions à valeurs dans un autre espace de fonctions :dimension infinie) linéaires ou pas

Problème de contrôle optimal = minimisation de J sous contraintes

$$(P)\quad \begin{cases} \min \mathrm{J}\,(x,u) & \\ x = \mathrm{F}\,(u,t) & \longleftarrow \text{Equation d'état} \\ u \in U_{ad} & \longleftarrow \text{Contraintes sur le contrôle} \\ x \in K & \longleftarrow \text{Contraintes sur l'état} \end{cases}$$

4. La commande optimale

4.2 Exemple 1

Equation d'état = Equations différentielles

$$x : [0,+\infty) \to R^n$$

$$\begin{cases} \dfrac{dx}{dt}(t) = f(t,x(t),u(t)), \quad t \in \left]0,+\infty\right[, \\ x(0) = x_o \quad \text{fixé dans } R^n \end{cases}$$

Coût d'un problème

à **horizon fini** : T

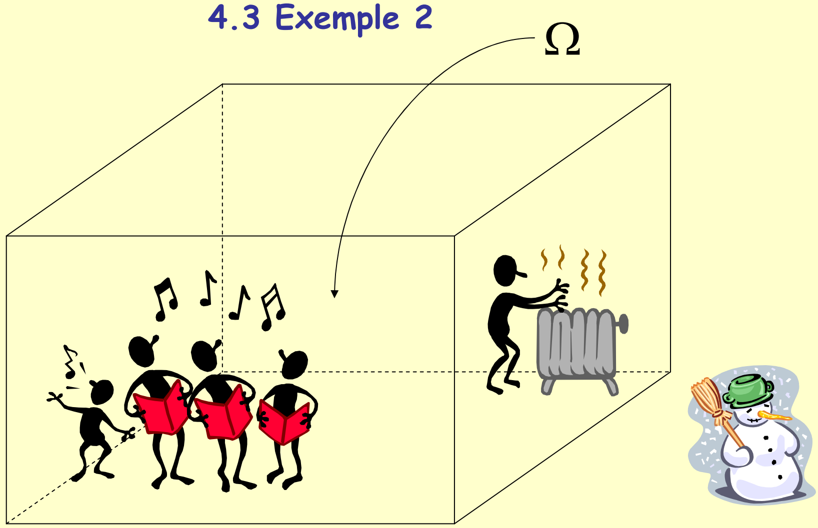
$$J(x,u) = \int_0^T \Phi(x(t),u(t)dt$$

de **temps optimal**

$$J(x,u) = \int_0^t ds$$

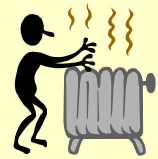
4. La commande optimale

4.3 Exemple 2



Problème de contrôle optimal

$$(P)\quad \begin{cases} \min J(\vartheta,u) = \displaystyle\int_{\Omega} \left|\vartheta(x)-\vartheta_o\right|^2 dx + \frac{\alpha}{2}\int_{\partial\Omega} \left|u(s)\right|^2 ds \\ -\Delta\vartheta = f \ \text{ dans } \ \Omega, \dfrac{\partial\vartheta}{\partial n} = u \text{ sur } \partial\Omega \\ \left|u(s)\right| \le a \ \text{ p.p. } s \end{cases}$$



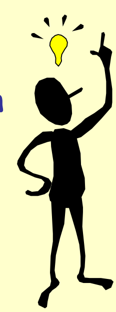
Système d'optimalité

$$(S)\quad \begin{cases} -\Delta\vartheta^* = f \ \text{ dans } \ \Omega, \dfrac{\partial\vartheta^*}{\partial n} = u^* \ \text{ sur } \partial\Omega & \longleftarrow \text{Équation d'état} \\ -\Delta p^* = \vartheta^* - \vartheta_o \ \text{ dans } \ \Omega, -\dfrac{\partial p^*}{\partial n} = 0 \text{ sur } \partial\Omega & \longleftarrow \text{Équation adjointe} \\ u^* = \Pi_{U_{ad}}\left(-\dfrac{p^*}{\alpha}\right) & \longleftarrow \text{Projection} \end{cases}$$

Boucle fermée !!

4. La commande optimale

4.3 Démarche « standard » du mathématicien



- Modéliser et formuler le problème
- Préciser le cadre fonctionnel
- Prouver l'existence d'une solution
- Discuter l'unicité (quitte à rajouter des contraintes)
- « Caractériser » la solution par un système d'optimalité ou un principe de Pontryagin
- Utiliser les conditions d'optimalité pour calculer explicitement ou numériquement la solution

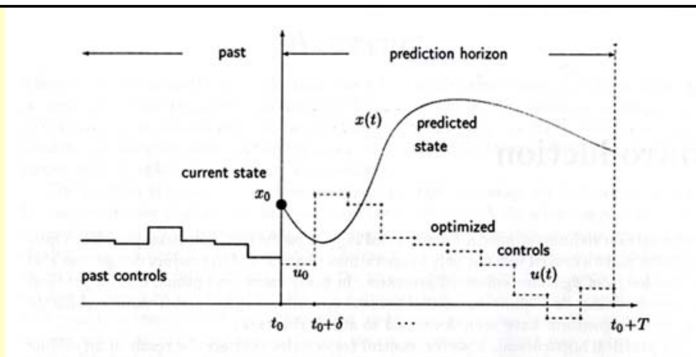
Il n'est donc pas question de loi de commande ni de boucle fermée…

5. La commande prédictive

5.1 Principe

But : faire de la commande **en temps réel…**

on détermine un contrôle au temps t_o en résolvant un problème de contrôle optimal sur une horizon de **prédiction** : $[t_o, t_o + T]$. On n'utilise la commande trouvée que pour un temps voisin de t_o, et à l'instant $[t_o+\delta, t_o + \delta + T]$ on résoud un nouveau problème de contrôle optimal. On a utilisé une fenêtre (ou un horizon) « *glissante* ». On obtient donc une suite de problèmes d'optimisation formulés et résolus en temps réel ce qui autorise la correction rapide des perturbations.…



Algorithme « standard »

1.Formulation du problème d'optimisation avec les données de l'étape k
- Initialisation de la méthode de calcul de la solution
- Itérations
- Arrêt quand un critère d'arrêt est vérifié (ou quand un temps limite est atteint)

2.On donne la première valeur du contrôle à l'usine

3.On passe de k à k+1 et on retourne à 1.

5. La commande prédictive

5.2. Itérations en temps réel

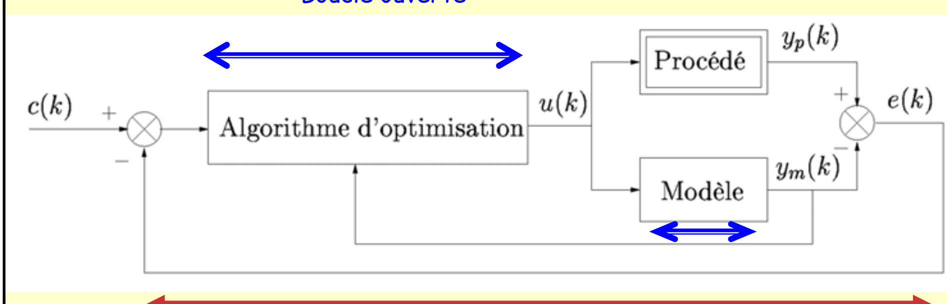
Algorithme

1. Préparation de la kième itération en temps réel autant que possible sans la connaissance des données de l'étape k
2. Quand les données de l'étape k sont disponibles, on modifie le problème, et on fait les calculs qui permettent d'obtenir la première valeur du contrôle rapidement.
3. On donne cette valeur immédiatement à l'usine
4. On finit les calculs de la kième itération.
5. On passe de k à k+1 et on retourne à 1.

5. La commande prédictive

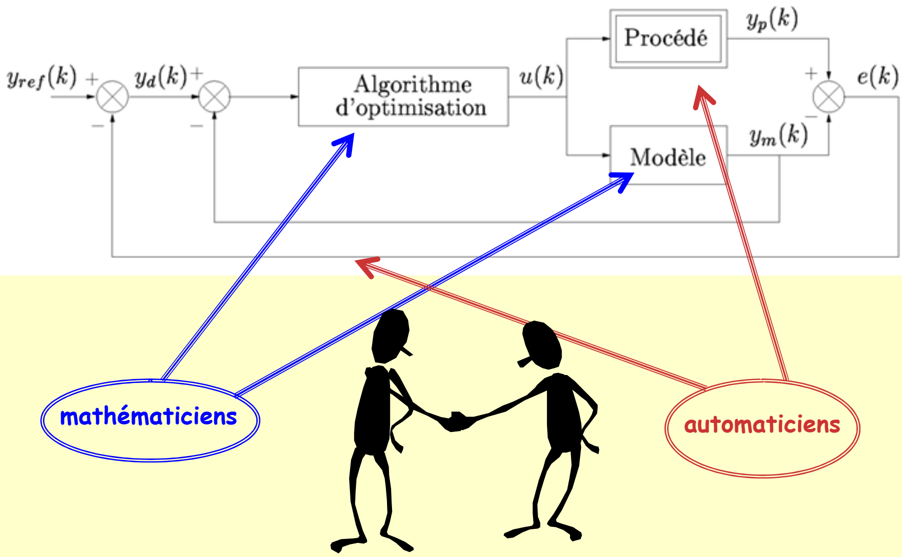
5.3. Commande prédictive par modèle interne

Boucle ouverte



Boucle fermée

Poursuite de trajectoires



Bibliographie

1. **M. Bergounioux**,*Optimisation et contrôle des systèmes linéaires*, Dunod, 2001
2. **M. Bergounioux**, *A quel moment peut-t'on discrétiser en contrôle optimal ?,* Actes CIFA 2000, pp. 366-368, 2000
3. **M. Diehl**, *Real -Time Optimization for Large Scale Nonlinear Processes*, PhD thesis, Ruprecht-Karls-Universität, Heidelberg,2001
4. **P. Dufour**, *Commande prédictive de systèmes non linéaires à paramètres répartis et applications*, Thèse, Université d'Orléans, 2000
5. **C.E. Garcia D.M. Prett et M. Morari**, *Model predictive control: Theory and practice- a survey*, Automatica, pp. 25-335, 1989
6. **J-L. Lions**, *Contrôle optimal des équations aux dérivées partielles*,Dunod, Paris, 1968
7. **M.Morari et E. Zafiriou**, *Robust Control*, Dunod, 1983
8. **P. Rouchon**, *Contrôle de systèmes décrits par des équations aux dérivées partielles : deux exemples types*, Actes CIFA 2000, pp.988-1000, 2000