Dynamique et commande

Julien Alexandre dit Sandretto

Department U2IS
ENSTA ParisTech
ROB316-2019-2020

Introduction

Cours précédent

Modèle cinématique, repères, stabilisation et suivi, PID, importance lien actionneurs/capteurs

Modèle cinématique suffisant ?

Modèle dynamique

Parfois nécessaire

Besoin de considérer forces, inertie, glissements, frottements

Généralisation

Modèle cinématique peut être vu comme un modèle dynamique simplifié

Modèle dynamique

Généralement issu du principe fondamental de la dynamique, fait apparaître l'accélération (dérivée temporelle de la vitesse) :

$$\sum F = ma$$

Cinématique, dynamique

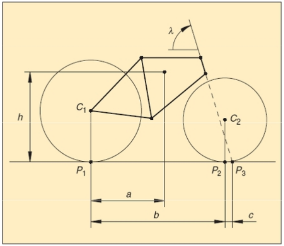
Cinématique : position en fonction de la vitesse et donc issue de la géométrie
Dynamique : vitesse en fonction de l'accélération et donc issue des forces (et moments) en présence

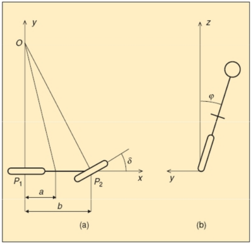
Modèle dynamique

Exemple de la bicyclette extrait intro commande prédictive - Supelec

Seule l'énergie permet l'équilibre (à l'arrêt on tombe)

- **Modèle de la bicyclette :** **Astrom, Klein, Lennartsson, *Control System Magazine*, 2005**





a, h : coordonnées du centre d'inertie

V : vitesse de la roue arrière constante

m : masse totale du système

J_x : moment d'inertie sur x

$D = -J_{xz}$: moment d'inertie sur $x\,z$

φ : angle d'inclinaison (positif à droite)

δ : angle de braquage (positif à gauche)

- **Hypothèses : Linéarisation aux petits angles**
- **Théorème du moment dynamique :**

$$J_x \frac{\mathrm{d}^2\varphi(t)}{\mathrm{d}t^2} = \underbrace{m\,g\,h\,\varphi(t)}_{\substack{\text{couple lié} \\ \text{à la gravité}}} + \underbrace{\frac{DV}{b}\frac{\mathrm{d}\delta(t)}{\mathrm{d}t}}_{\substack{\text{couple lié} \\ \text{aux forces d'inertie} \\ \text{lors du braquage}}} + \underbrace{\frac{mV^2h}{b}\delta(t)}_{\substack{\text{couple lié} \\ \text{aux forces centrifuges}}}$$

Modèle dynamique

Tout comme le modèle cinématique, le modèle dynamique permet d'établir une relation. Ici entre les efforts et les mouvements qui en découlent :

- Problème dynamique direct = calcul des accélérations produites par des efforts donnés, avec une expression matricielle du type $\ddot{q} = F(q, \dot{q}, \tau)$
- Problème inverse = connaissant le mouvement, c'est le calcul des efforts $\tau = G(q, \dot{q}, \ddot{q})$ qui en sont la cause.

Nous nous intéressons au direct (issu du PFD) : cela nous donne une **équation différentielle ordinaire** !

Modèle dynamique

En intégrant le modèle direct $\ddot{q} = F(q, \dot{q}, \tau)$, nous pouvons déterminer le mouvement du système

De nombreuses hypothèses simplificatrices

- corps = solides rigides
- tout ce qui est compliqué est estimé empiriquement
- actionneurs schématisés par des forces ou des couples (fonction des commandes)
- forces de contact suivent la loi approchée de Coulomb

Modèle dynamique

Avantages

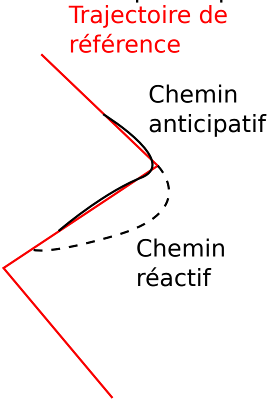
- Prise en compte des forces, de l'inertie, des glissements, etc.
- Permet d'estimer le futur comportement du système par intégration

Inconvénients

- Nécessite plus de connaissances sur le système (matrice d'inertie, coefficient de traînée, de frottement)
- Nonlinéarité
- Nécessite une intégration

Prédiction du futur

En intégrant le modèle dynamique (une EDO), on peut prédire le futur du système en fonction d'une commande.
C'est la philosophie de **l'anticipation** dans le contrôle :



- Prédiction $\Rightarrow$ anticipation
- Prise en compte de contraintes
- conforme aux capacités dynamiques du système (contrairement à la cinématique parfois irréalisable)
- Exploite la dynamique (trajectoire impossible sans, par exemple prise d'élan)

Prédiction des futurs

- Prédire un futur en fonction d'une entrée permet de la valider (respect des contraintes)

Prédire **des** futurs : permet de choisir le "meilleur" futur !

Meilleur implique une notion de coût qui peut faire intervenir :

- commande (consommation de carburant)
- état du système (distance à une consigne, choix parmi plusieurs trajectoires, performances)
- état du système à la fin par rapport à une consigne (coût terminal nul implique le succés de la mission)

Prédiction des futurs

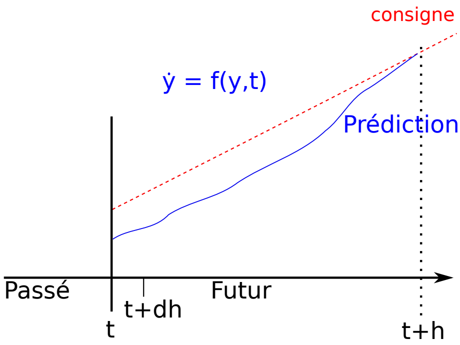
Choisir la commande telle que la prédiction de l'état du système suive une consigne, pour un coût minimal et en respectant des contraintes

$\Rightarrow$ **Commande prédictive**

Commande prédictive

Principe

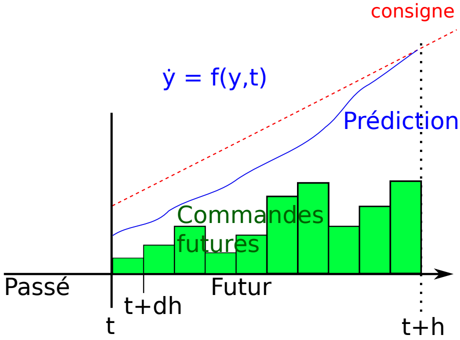
- Modèle numérique pour prédire le futur
- Calcul d'une séquence de commandes en boucle ouverte minimisant un coût et respectant les contraintes sur un horizon fini
- Injection de la première valeur de cette séquence
- Mesure, réitération de la procédure sur un horizon glissant



Commande prédictive

Principe

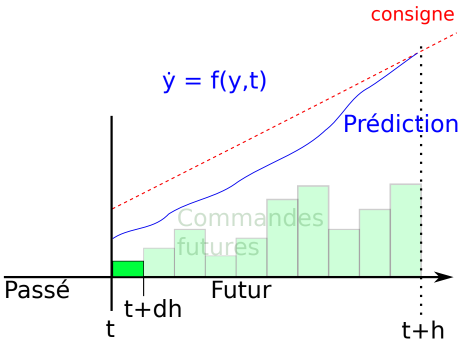
- Modèle numérique pour prédire le futur
- Calcul d'une séquence de commandes en boucle ouverte minimisant un coût et respectant les contraintes sur un horizon fini
- Injection de la première valeur de cette séquence
- Mesure, réitération de la procédure sur un horizon glissant



Commande prédictive

Principe

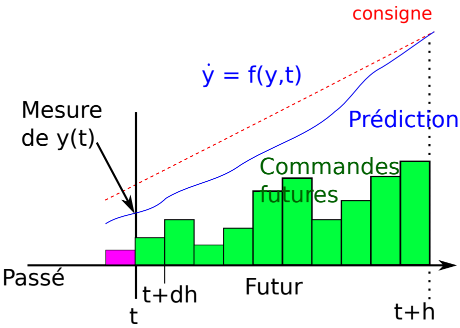
- Modèle numérique pour prédire le futur
- Calcul d'une séquence de commandes en boucle ouverte minimisant un coût et respectant les contraintes sur un horizon fini
- Injection de la première valeur de cette séquence
- Mesure, réitération de la procédure sur un horizon glissant



Commande prédictive

Principe

- Modèle numérique pour prédire le futur
- Calcul d'une séquence de commandes en boucle ouverte minimisant un coût et respectant les contraintes sur un horizon fini
- Injection de la première valeur de cette séquence
- Mesure, réitération de la procédure sur un horizon glissant

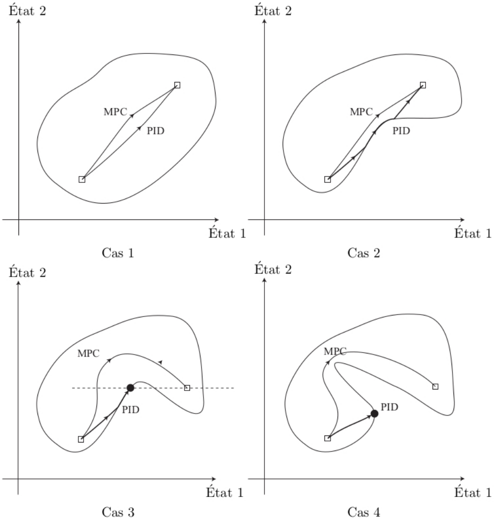


Commande prédictive

Objectifs généraux selon (Qin et Badgwell, 1996) de la Commande Prédictive Basée sur le Modèle :

- éviter d'enfreindre les contraintes d'entrée (saturation actionneurs) et de sortie (sûreté système)
- amener les variables manipulées vers leurs valeurs stationnaires optimales
- amener les variables commandées vers leurs valeurs stationnaires optimales
- éviter les variations excessives des variables manipulées (anticipation)

Différence avec PID



Mise en oeuvre

De nombreuses implémentations différentes, avec différentes fonction de coût, etc.

En général

Système dynamique non linéaire : nonlinear model predictive control (NMPC)

Formalisation générale

Modèle non linéaire

$$\dot{x} = f(x(t), u(t)), \quad x(0) = x_0$$

Soumis aux contraintes d'entrée et d'état :

$$x(t) \in \mathcal{X} \text{ and } u(t) \in \mathcal{U}$$

Commande prédictive non linéaire

Un problème de commande optimale en boucle ouverte à horizon de prédiction fini T_p :

$$\min_{\hat{u}} J(x, \hat{u}, T_p) = \int_t^{t+T_p} (||\hat{x}(\tau; x(t), t)||_Q^2 + ||\hat{u}(\tau)||_R^2) d\tau$$

avec $\hat{x}(t; x, t) = x(t)$ solution à IVP à l'instant t avec condition initiale x à l'instant t

Formalisation générale

Approche Chen et Allgöwer (1998)

Ajout d'une pénalité terminale :

$$\min_{\hat{u}} J(x, \hat{u}, T_p) = ||\hat{x}(t + T_p; x(t), t)||_P^2 + \int_t^{t+T_p} (||\hat{x}(\tau; x(t), t)||_Q^2 + ||\hat{u}(\tau)||_R^2) d\tau$$

Force l'état à 0 à la fin de la prédiction (généralisable après réécriture du modèle)

Résolution

Système	L	NL	NL ou L
Contraintes	L	L	L ou NL
Coût	L	L	L ou NL
Méthode :	MPC	Adaptive MPC ou Gain-scheduled MPC	NMPC

- MPC : problème d'optimisation convexe
- AMPC ou GSMPC : suite de problèmes d'optimisation convexe (linéarisation du système)
- NMPC : optimisation non-convexe...

Algorithme pour le non linéaire : linéarisation

Le système :

$$\dot{x} = f(x, u), \quad x(0) = x_0 \tag{1}$$

doit suivre une référence (une trajectoire) :

$$\dot{x}_r = f(x_r, u_r) \tag{2}$$

développement en série de Taylor à l'ordre 1 de (1) évaluée en (x_r, u_r) :

$$\dot{x} = f(x_r, u_r) + \left.\frac{\partial f(x, u)}{\partial x}\right|_{(x_r, u_r)} (x - x_r) + \left.\frac{\partial f(x, u)}{\partial u}\right|_{(x_r, u_r)} (u - u_r) \tag{3}$$

(3)-(2) donne $\dot{\tilde{\mathbf{x}}} = \mathbf{f_{x,r}}\tilde{\mathbf{x}} + \mathbf{f_{u,r}}\tilde{\mathbf{u}}$ avec $\tilde{x} = x - x_r$ et $\tilde{u} = u - u_r$
équivalent à $\dot{\tilde{x}} = A_r\tilde{x} + B_r\tilde{u}$

Algorithme pour le non linéaire : discrétisation

Schéma d'Euler :

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{x}(k) + (A_r\tilde{x}(k) + B_r\tilde{u}(k))\delta t \\ \Leftrightarrow \tilde{x}(k+1) &= A(k)\tilde{x}(k) + B(k)\tilde{u}(k) \end{aligned}$$

avec $A(k) = I + A_r\delta t$ et $B(k) = B_r\delta t$

Algorithme pour le non linéaire : coût

$$\Phi(\tilde{x}, \tilde{u}) = \sum_{j=1}^{N} \left[\tilde{x}^T(k+j)\ Q\ \tilde{x}(k+j) + \tilde{u}^T(k+j)\ R\ \tilde{u}(k+j) \right] \tag{4}$$

Q et R pondération (matrices positives)

Algorithme pour le linéaire : vectorisation

Besoin d'un seul système pour optimisation avec QP (quadratic programming solver)

linéaire : $A = A(k)$ et $B = B(k)$

$$\begin{array}{rcl}\tilde{x}(k+1) & = & A\tilde{x}(k) + B\tilde{u}(k) \\ \tilde{x}(k+2) & = & A\tilde{x}(k+1) + B\tilde{u}(k+1) \\ \Leftrightarrow \tilde{x}(k+2) & = & A^2\tilde{x}(k) + AB\tilde{u}(k) + B\tilde{u}(k+1) \\ \tilde{x}(k+3) & = & A^3\tilde{x}(k) + A^2B\tilde{u}(k) + AB\tilde{u}(k+1) + B\tilde{u}(k+2)\end{array}$$

Le système à résoudre devient

$$X(k) = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^N \end{pmatrix} \tilde{x}(k) + \begin{pmatrix} B & 0 & \dots & 0 & 0 \\ AB & B & \ddots & \vdots & \vdots \\ \vdots & & & & \\ A^{N-1}B & A^{N-2}B & \dots & AB & B \end{pmatrix} U(k:k+N)$$

Algorithme pour le linéaire : résolution

On a $X(k) = \hat{A}\tilde{x}(k) + \hat{B}U$
Il reste un problème d'optimisation à résoudre...

Approche (très) simplifiée

On veut donc trouver U^* tel que $X(k) = 0$ (erreur à la trajectoire) et donc notre système à résoudre

$$\hat{B}U = -\hat{A}\tilde{x}(k)$$

avec $U_{min} < U^* < U_{max}$ et tel que

$$U^* = argmin \quad U^T RU$$

Une approche aux moindres carrés peut suffire : $U = -\hat{B}^{\#}\hat{A}\tilde{x}(k)$, avec $^{\#}$ la pseudo-inverse

Comment choisir l'horizon ?

Assez compliqué...par essai/erreur, imposé par le système (fréquence appareil de mesure p.ex.), ou la méthode évoluée suivante

Algorithme stabilisant pour le non linéaire

Rappel du système :

$$\dot{x} = f(x, u), \quad x(0) = x_0 \tag{5}$$

fonction de coût

$$J(x, u) = \int_0^{\infty} (||x(t)||_Q^2 + ||u(t)||_R^2) dt \tag{6}$$

trouver u avec pour consigne le zéro.

Algorithme stabilisant pour le non linéaire

Un algorithme plus compliqué [Chen&Allgower 1998] :

1. linéarisation jacobienne (A,B) puis calcul d'un retour d'état linéaire **localement** stabilisant $u = Kx$
2. choisir une constante positive $\alpha < -\lambda_{max}(A_K)$ (avec $A_K = A + BK$) et résoudre l'équation de Lyapunov :

$$(A_K + \alpha I)^T P + P(A_K + \alpha I) = -(Q + K^T RK)$$

   avec P matrice définie positive

3. trouver le plus grand β_1 définissant la région Ω_1 où les contraintes sont satisfaites lorsque $x \in \Omega_1$:

$$\Omega_1 = \{x \in \mathbb{R}^n | x^T Px \leq \beta_1\}, \ \Omega_1 \subset \mathcal{X}, \ \ Kx \in \mathcal{U}$$

Algorithme stabilisant pour le nonlinéaire

4. trouver le plus grand $\beta \in]0, \beta_1]$ définissant une région terminale Ω :

$$\Omega = \{x \in \Omega_1 | x^T P x \leq \beta\}$$

telle que l'état optimal solution du problème suivant soit non positive :

$$\max_x \{x^T P \phi(x) - \alpha x^T P x | x \in \Omega\}, \phi(x) = f(x, Kx) - A_K x$$

5. choisir l'horizon de prédiction T_p tel que $T_p \geq T_c + T_s$ avec T_s le temps maximum pour que le système non commandé atteigne Ω en partant de x_0

Avec cette approche

On obtient une zone dans laquelle notre commande est stabilisante et optimale, lorsque l'on va sortir de cette zone, on fait glisser l'horizon.

Théorique, difficile à mettre en oeuvre

Mais utile pour la vérification, voir TP