Automatique Avancée – Commande non-linéaire

P4: Techniques numériques: Introduction à la Commande Prédictive

Pascal Morin

pascal.morin@sorbonne-universite.fr

Sorbonne Université

Plan

Introduction

- Les techniques vues dans les parties P1-P3 sont de nature analytique. Ceci offre de nombreux avantages:
  - Garanties de stabilité, au moins locale.
  - Expressions explicites des lois de commande.
  - Bien adaptées au "temps réel" car ne nécessite pas de calculs lourds
- Elles peuvent cependant être difficiles à mettre en œuvre dans certains cas:
  - Systèmes non-linéaires de grande dimension (pas toujours linéarisables exactement, pas de fonction de Liapounov "simple").
  - Existence de contraintes fortes sur l'état ou la commande, e.g.,

$$\|u\| \leq u_{\max}\ , \quad \|\dot{u}\| \leq \dot{u}_{\max}\ , \quad \|x\| \leq x_{\max}$$

  - Problèmes de stabilisation de trajectoires pour des systèmes non-linéarisables exactement.

Introduction

- Une approche alternative: chercher une résolution numérique du problème.
- Intérêt:
  - On fait reposer la difficulté sur l'algorithmique et sur la puissance de calcul du processeur
  - Qui a beaucoup progressé depuis 20 ans
- Attention: ne croyez pas que cela va résoudre tous les problèmes!
  - Si le problème a été mal analysé et mal posé, il se peut qu'il n'y ait aucune solution.
  - Si la méthode de résolution est mal choisie, on risque de ne trouver aucune solution.
  - Si la mise en œuvre de la solution algorithmique n'est pas efficace, on risque de ne pas être "temps réel".
  - Les contraintes de temps de calcul restent fortes pour des systèmes dynamiques rapides et de grande dimension.

Introduction

Quelques éléments historiques:

- La principale approche de ce type est la Commande Prédictive (MPC pour "Model Predictive Control) aussi appelée "Commande par modèle interne"
- et ses nombreuses variantes: NMPC (Non-linear Model Predictive Control), GMPC (Generalized Model Predictive Control), eMPC (explicit Model Predictive Control), etc.
- La Commande Prédictive apparaît à la fin des années 1970,
- dans le contexte de l'industrie chimique/pétro-chimique: systèmes de grandes dimensions, dynamiques lentes (mieux adaptées à l'approche).
- Ses champs d'application se sont beaucoup élargies depuis, notamment en robotique.

Principe de la commande prédictive

Partons d'un système de commande général avec des contraintes:

$$\begin{cases} \dot{x} = f(x,u) \\ x \in \mathcal{X}\ ,\ \ u \in \mathcal{U} \end{cases}$$

où:

- $\mathcal{X}$ est un domaine connexe de $\mathbb{R}^n$ qui exprime les contraintes sur l'état du système,
- $\mathcal{U}$ un domaine compact de $\mathbb{R}^m$ qui exprime les contraintes sur la commande.

La commande prédictive étant une approche numérique par nature, on commence par exprimer le système en temps discret:

$$\forall k = 1, 2, \cdots \begin{cases} x_{k+1} = f_k(x_k, u_k) \\ x_k \in \mathcal{X}\ ,\ \ u_k \in \mathcal{U} \end{cases}$$

Principe de la commande prédictive

Cette discrétisation peut se faire de différentes façons:

- Schéma d'Euler explicite:

$$f_k(x_k, u_k) = x_k + \delta \frac{\partial f}{\partial x}(x_k, u_k)$$

  avec δ le pas de discrétisation.

- Intégration exacte quand c'est possible (nous verrons des exemples plus tard):

$$f_k(x_k, u_k) = x_k + \int_{k\delta}^{(k+1)\delta} f(x(s), u_k)\, ds$$

- Combinaison des deux.
- Etc.

Principe de la commande prédictive

Définition de l'objectif de commande:

Soit $k_0 \in \mathbb{N}$ et $x_{k_0} := \bar{x}_{k_0} := x(k_0\delta) \in \mathcal{X}$. On suppose qu'à l'instant $t_0 := k_0\delta$, l'objectif de commande peut être posé sous la forme du problème d'optimisation sous contrainte $(\mathcal{P}_{k_0})$ suivant:

$$(\mathcal{P}_{k_0}) \left\{ \begin{array}{l} \min_{\bar{u}(.)} J_{k_0}(\bar{x}(.), \bar{u}(.), N) \\ \\ \text{sous les contraintes:} \\ \forall k = k_0, \cdots, k_0 + (N-1)\,, \\ \qquad \bar{x}_{k+1} := f_k(\bar{x}_k, \bar{u}_k) \in \mathcal{X}\,,\ \ \bar{u}_k \in \mathcal{U} \end{array} \right.$$

où $J_{k_0}(\bar{x}(.), \bar{u}(.), N)$ est une fonction positive dépendant de $\bar{x}_{k_0}, \cdots, \bar{x}_{k_0+N},\ \bar{u}_{k_0}, \cdots, \bar{u}_{k_0+N-1}$, qui ne s'annulle que lorsque l'objectif de commande est réalisé.

Principe de la commande prédictive

Définition de l'objectif de commande:

- Si l'objectif est de stabiliser asymptotiquement un équilibre (x^*, u^*), un exemple de choix de critère J est:

$$J_{k_0}(\bar{x}(.), \bar{u}(.), N) := \sum_{k=k_0}^{k_0+N-1} F(\bar{x}_k, \bar{u}_k) + V(\bar{x}_{k_0+N})$$

avec

$$F(\bar{x}_k, \bar{u}_k) = (\bar{x}_k - x^*)^T Q(\bar{x}_k - x^*) + (\bar{u}_k - u^*)^T R(\bar{u}_k - u^*)$$
$$V(\bar{x}_k) = (\bar{x}_k - x^*)^T P(\bar{x}_k - x^*)$$

- Dans ce cas, le critère n'est autre qu'une version discrétisée et à horizon fini de la commande LQR.
- L'horizon d'optimisation N peut éventuellement être infini, mais la plupart du temps on ne saura trouver une solution au problème d'optimisation que pour N fini.

Principe de la commande prédictive

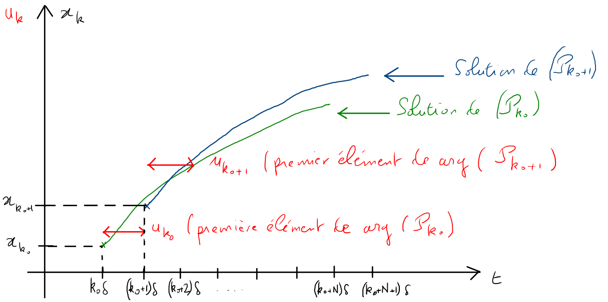
Le principe du MPC, illustré sur la figure ci-dessous, est le suivant:

Répéter pour $k_0 = 0, 1, \cdots$:

_ Résoudre $(\mathcal{P}_{k_0})$

_ Soit $(\bar{u}_{k_0}, \cdots, \bar{u}_{k_0+N-1}) = \arg\min_{\bar{u}(.)} J_{k_0}(\bar{x}(.), \bar{u}(.), N)$

_ Poser $u_{k_0} := \bar{u}_{k_0}$



Principe de la commande prédictive

Justification de l'approche:

- La résolution de $(\mathcal{P}_{k_0})$ à l'instant $t = k_0\delta$ fournit une séquence de commandes $\bar{u}_{k_0}, \cdots, \bar{u}_{k_0+N-1}$ de type "boucle ouverte".
- En effet, cette séquence est calculée seulement à partir de la connaissance de x_{k_0}.
- Si l'on appliquait toute la séquence ainsi déterminée, on aurait un fonctionnement en boucle ouverte sur tout l'intervalle de temps $[k_0\delta, (k_0 + N)\delta]$, et donc une commande peu robuste.
- L'idée fondamentale du MPC est de n'appliquer que le premier élément de cette séquence, $\bar{u}_{k_0}$, puis de réitérer le processus en $t = (k_0 + 1)\delta, (k_0 + 2)\delta, \cdots$.
- Ce qui va permettre d'obtenir un fonctionnement de type "feedback".

Principe de la commande prédictive

Intérêts de l'approche:

- L'approche est "potentiellement" applicable à n'importe quel système.
- Elle permet d'intégrer facilement des contraintes.
- Elle contient un aspect de performance (car basée sur la minimisation d'une fonction de coût).
- Elle contient implicitement un aspect d'anticipation (on optimise sur un horizon de temps "long").
- Elle s'applique indifféremment à un problème de stabilisation de point fixe ou de trajectoire.

Principe de la commande prédictive

Difficultés/dangers de l'approche:

- Le problème de minimisation peut être très compliqué à résoudre, et conduire à des solutions très sous-optimales (minimas locaux).
- Il n'y a pas de preuve générale que l'approche conduise à un système stable en boucle fermée (mais il y a des preuves de stabilité dans des cas particuliers). Le coût terminal V joue un rôle bénéfique par rapport à cela (s'il est bien choisi, V joue un rôle similaire à une fonction de type Lyapunov).
- Il y a un compromis à trouver sur la valeur de N:
  - Une grande valeur de N est bénéfique du point de vue de la stabilité.
  - Mais conduit à un problème d'optimisation beaucoup plus lourd en calculs.

Principe de la commande prédictive

Difficultés/dangers de l'approche:

- De même, il y a un compromis à trouver sur la valeur de δ:
  - Il faut que δ soit petit vis-à-vis de la dynamique du système puisque la commande va être maintenu constante sur un intervalle de temps de longueur δ (risque d'instabilité si période d'échantillonage de la commande trop longue).
  - Il faut que δ soit grand vis-à-vis des temps de calcul car sinon on risque aussi d'engendrer des instabilités (retard d'application de la commande).
- D'où la difficulté à appliquer la méthode à des systèmes de dynamique rapide, et de grande dimension.

Mise en œuvre pour les systèmes linéaires

On considère dans toute cette partie un système linéaire

$$\dot{x} = Ax + Bu$$

où l'on suppose des contraintes affines sur l'état et la commande, i.e.:

$$Gx \leq g \,, \quad Hu \leq h$$

où G et H sont des matrices constantes et g, h des vecteurs constants à coefficients positifs. Par exemple, une contrainte sur la composante x_1 de $x \in \mathbb{R}^2$ du type

$$-a \leq x_1 \leq b$$

avec $a, b > 0$ pourra s'écrire

$$\begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix} x \leq \begin{pmatrix} b \\ a \end{pmatrix}$$

Mise en œuvre pour les systèmes linéaires

Discrétisation de la dynamique: Le discrétisé s'écrit

$$x_{k+1} = \bar{A} x_k + \bar{B} u_k$$

- Schéma d'Euler explicite: On a

$$\bar{A} = I + \delta A \, , \bar{B} = \delta B$$

- Intégration exacte explicite: La formule d'intrégration d'un système linéaire donne:

$$x_{k+1} = e^{A\delta} x_k + e^{A\delta} \int_0^\delta e^{-As} B u_k(s) \, ds$$

En considérant le fait que la commande sera constante sur chaque intervalle de temps $[k\delta, (k+1)\delta)$, on a donc

$$\bar{A} = e^{A\delta} \, , \quad \bar{B} = e^{A\delta} \int_0^\delta e^{-As} B \, ds$$

Mise en œuvre pour les systèmes linéaires

Définition du critère à minimiser:

- Le succès de la méthode MPC pour un système donné repose sur l'existence d'algorithmes permettant de résoudre le problème de minimisation considéré.
- Pour les systèmes linéaires, cela repose sur les outils de l'optimisation convexe.
- Plus précisément, des fonctions de coût quadratiques en l'état et la commande et le fait de considérer des contraintes affines conduisent à un problème d'optimisation quadratique (QP pour "quadratic programming en Anglais)

Mise en œuvre pour les systèmes linéaires

Définition du critère à minimiser:

Definition

On appelle problème d'optimisation quadratique un problème du type

$$\begin{array}{rl} \text{minimiser} & \frac{1}{2}\xi^T P\xi + c^T\xi \\ \text{sous la contrainte} & N\xi \leq d \end{array}$$

où P est une matrice symétrique, c et d des vecteurs constants, et ξ l'inconnue.

Remarques:

- Si P est SDP, le problème d'optimisation quadratique est un problème d'optimisation strictement convexe sur un ensemble convexe. Il admet donc une solution unique.
- Il existe plusieurs méthodes permettant de résoudre ce type de problème de façon efficace (polynomiale en temps).

Mise en œuvre pour les systèmes linéaires

Définition du critère à minimiser:

Un choix de critère classiquement utilisé pour les systèmes linéaires est:

$$J_{k_0}(\bar{x}(.), \bar{u}(.), N) := \Sigma_x + \Sigma_u + \Sigma_{\Delta u}$$

avec

$$\begin{aligned}
\Sigma_x &= \sum_{p=1}^{N} (\bar{x}_{k_0+p} - x^*_{k_0+p})^T Q_{k_0+p} (\bar{x}_{k_0+p} - x^*_{k_0+p}) \\
\Sigma_u &= \sum_{p=1}^{N} (\bar{u}_{k_0+p-1} - u^*_{k_0+p-1})^T R (\bar{u}_{k_0+p-1} - u^*_{k_0+p-1}) \\
\Sigma_{\Delta u} &= \sum_{p=1}^{N-1} (\Delta\bar{u}_{k_0+p} - \Delta u^*_{k_0+p})^T S (\Delta\bar{u}_{k_0+p} - \Delta u^*_{k_0+p})
\end{aligned}$$

Définition du critère à minimiser:

Remarques:

- Les variables x^* et u^* correspondent respectivement à l'état et à la commande de référence. Par exemple, si l'objectif est de stabiliser un point d'équilibre (x_0, u_0), $\forall k,\ x_k^* = x_0$ et $u_k^* = u_0$.
- Ici, pour tout k, et toute variable ξ, $\Delta\xi_k := \xi_k - \xi_{k-1}$.
- Les matrices Q_k, R, et S sont supposées SDP.
- Le terme $\Sigma_{\Delta u}$ a pour effet de pénaliser les fortes variations de commande.
- La plupart du temps, on aura $Q_k = Q\,, \forall k = k_0+1, k_0+N-1$. Par contre, Q_N, associé au coût terminal, peut être une matrice différente jouant un rôle de "type fonction de Liapounov".

Mise en œuvre pour les systèmes linéaires

Ré-écriture comme un problème QP:

- Afin de pouvoir exploiter les outils d'optimisation existants pour les problèmes QP, le problème doit être ré-écrit sous la forme canonique de la définition du Slide 18.
- On commence par vérifier la propriétés suivante:

$$\forall p = 1, 2, \cdots , \quad x_{k_0+p} = \bar{A}^p x_{k_0} + \sum_{i=0}^{p-1} \bar{A}^i \bar{B} u_{k_0+p-i}$$

- Ceci permet de reformuler le critère sous la forme d'un critère QP en la variable $\bar{u}_{k_0,N} = (\bar{u}_{k_0}^T, \cdots , \bar{u}_{k_0+N-1}^T)^T$, et les contraintes également comme des contraintes affines sur cette variable.

Mise en œuvre pour les systèmes linéaires

Ré-écriture comme un problème QP:

Plus précisément, on pourra vérifier les expressions suivantes:

$$\begin{aligned}\Sigma_x &= \sum_{p=1}^{N} z_{k_0,p}^T Q_{k_0+p} z_{k_0,p} + \sum_{p=1}^{N} \bar{u}_{k_0,N}^T M_p^T Q_{k_0+p} M_p \bar{u}_{k_0,N} \\ &\quad + 2\sum_{p=1}^{N} z_{k_0,p}^T Q_{k_0+p} M_p \bar{u}_{k_0,N}\end{aligned}$$

avec

$$M_p = \left(\bar{A}^{p-1}\bar{B}\ ,\ \bar{A}^{p-2}\bar{B}\ ,\ \cdots\ ,\ \bar{B}\ ,\ 0\ ,\ \cdots\ ,\ 0\right)$$

et $z_{k_0,p} := \bar{A}^p x_{k_0} - x^*_{k_0+p}$.

Mise en œuvre pour les systèmes linéaires

Ré-écriture comme un problème QP:

- On remarque que le premier terme de Σ_x n'a aucune influence sur le problème de minimisation. Il peut donc être enlevé et on obtient une expression de la forme d'un problème QP en $\bar{u}_{k_0,N}$.
- On vérifie facilement que le terme Σ_u est de la même forme:

$$\Sigma_u = (\bar{u}^*_{k_0,N})^T \bar{R}\bar{u}^*_{k_0,N} + \bar{u}^T_{k_0,N}\bar{R}\bar{u}_{k_0,N} - 2(\bar{u}^*_{k_0,N})^T \bar{R}\bar{u}_{k_0,N}$$

avec $\bar{u}^*_{k_0,N} = ((\bar{u}^*_{k_0})^T, \cdots, (\bar{u}^*_{k_0+N-1})^T)^T$ et
$\bar{R} = diag(R, \cdots, R)$.

Mise en œuvre pour les systèmes linéaires

Ré-écriture comme un problème QP:

- Finalement, on vérifie que:

$$\Sigma_{\Delta u} = (\bar{u}^*_{k_0,N})^T \bar{S} \bar{u}^*_{k_0,N} + \bar{u}^T_{k_0,N} \bar{S} \bar{u}_{k_0,N} - 2(\bar{u}^*_{k_0,N})^T \bar{S} \bar{u}_{k_0,N}$$

avec

$$\bar{S} = \begin{pmatrix} S & -S & 0 & \cdots & \cdots & 0 \\ -S & 2S & -S & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -S & 2S & -S \\ 0 & \cdots & \cdots & 0 & -S & S \end{pmatrix}$$

Outils numériques

Outils Matlab:

- Résolution des problèmes QP: fonction matlab "quadprog"
- Outils généraux pour le MPC: MPC toolbox

Packages Python:

- Résolution des problèmes QP: packages "cvxopt" (Convex Optimization), "quadprog" (Quadratic Programming), "qpOASES"
- Software general pour le MPC: toolbox "do-mpc"

Extension au cas non-linéaire

L'extension du MPC au cas général pose plusieurs difficultés:

- Contrairement au cas linéaire, on n'aura généralement pas une formule simple exprimant x_{k_0+p} en fonction de x_{k_0} et de $\bar{u}_{k_0,N}$, et de plus cette relation sera rarement linéaire/$\bar{u}_{k_0,N}$.
- On va donc perdre les propriétés fortes de convexité de J,
- et on va perdre avec cela les garanties de solution associées aux problèmes QP.
- La plupart du temps on ne pourra avoir que des convergences locales, sous réserve d'une bonne "initial guess".

Extension au cas non-linéaire

De façon générale, le NMPC (Nonlinear Model Predictive Control) va conduire a un problème d'optimisation du type:

$$\begin{array}{rl} \text{minimiser} & J(\xi, \Theta) \\ \text{sous les contraintes} & \left\{ \begin{array}{l} G(\xi, \Theta) \leq g \\ H(\xi, \Theta) = h \end{array} \right. \end{array}$$

- avec ξ l'inconnue et Θ un vecteur de paramètres (incluant $x_{k_0}, N, x_k^*, u_k^*, \cdots$.
- Les contraintes d'égalité apparaissent si l'on utilise l'approche simultannée, dans laquelle l'état est ajouté comme inconnue et la dynamique est intégrée dans les contraintes, i.e.

$$x_{k+1} - f_k(x_k, u_k) = 0$$

- ce qui nécessite des outils numériques spécifiques pour traiter ces contraintes d'égalité.

Outils numériques

- Une méthode classiquement utilisée est la méthode OQS:Optimisation Quadratique Successive, connue en Anglais sous le sigle SQP: Sequential Quadratic Programming.
- Il s'agit d'une méthode de type Newton avec linéarisation des contraintes.
- Outils Matlab:
  - fonction matlab "fmincon".
  - Outils généraux: Optimization toolbox.
- Packages Python:
  - Package "SciPy", Sous-package "Optimize".
  - Software general pour le MPC: toolbox "do-mpc".

Extension au cas non-linéaire

Pour plus de détails sur le NMPC, on pourra consulter:

- T.A. Johansen, *Introduction to Nonlinear Model Predictive Control and Moving Horizon Estimation.* In Selected Topics on Constrained and Nonlinear Control, 2011.
- R. Findeisen, F. Allgöwer *An introduction to Nonlinear Model Predictive Control.* 21st Benelux Meeting on Systems and Control, 2002.
- D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert. *Constrained model predictive control: Stability and optimality.* Automatica, 2000.