MAP434 - Contrôle de modèles dynamiques

Programmation dynamique en temps discret

Alexandre Ern

ENPC - INRIA - X

amphi du mardi 21 mai 2019

Plan de la séance

- Contrôle optimal en temps discret
- Fonction valeur et programmation dynamique
- Application au système LQ
- Optimisation combinatoire
- Poly, chapitre 7

Contrôle optimal en temps discret (1)

- On considère des instants discrets

$$0 = t_0 < t_1 < \dots < t_N = T \quad (N > 0)$$

- Plutôt que de décrire l'état du système et le contrôle par des **fonctions** $x : [0, T] \rightarrow \mathbb{R}^d$ et $u : [0, T] \rightarrow U \subset \mathbb{R}^k$ (sous-ensemble fermé non-vide), on les décrit par des **familles discrètes**

$$(x_m)_{m \in \{0:N\}} \in \mathbb{R}^{d(N+1)}, \quad (u_m)_{m \in \{0:N-1\}} \in U^N$$

- La dynamique discrète d'évolution est

$$x_{m+1} = F_m(x_m, u_m), \quad \forall m \in \{0:N-1\}, \quad x_0 = x$$

où $F_m : \mathbb{R}^d \times U \rightarrow \mathbb{R}^d$, $\forall m \in \{0:N-1\}$, et $x \in \mathbb{R}^d$ est la condition initiale (on la note x car on va la faire varier...)

- étant donnée la CI x , les contrôles $(u_m)_{m \in \{0:N-1\}}$ déterminent par récurrence les états successifs $(x_m)_{m \in \{1:N\}}$

Contrôle optimal en temps discret (2)

- Afin de définir un critère d'optimalité en temps discret, on se donne
 - une famille de fonctions $(G_m)_{m \in \{0:N-1\}}$ avec $G_m : \mathbb{R}^d \times U \rightarrow \mathbb{R}$
 - une fonction $h : \mathbb{R}^d \rightarrow \mathbb{R}$
- Le critère $J_0 : \mathbb{R}^d \times U^N \rightarrow \mathbb{R}$ est

$$J_0(x; u_0, \dots, u_{N-1}) = \sum_{m \in \{0:N-1\}} G_m(x_m, u_m) + h(x_N)$$

- on a explicité la dépendance en la CI $x \in \mathbb{R}^d$
 - l'indice 0 fait référence au fait que la CI est prescrite en t_0
- Le problème de contrôle optimal en temps discret consiste à chercher

$$V_0(x) := \min_{(u_0, \dots, u_{N-1}) \in U^N} J_0(x; u_0, \dots, u_{N-1})$$

Lien avec le contrôle optimal en temps continu

- Pour le problème en temps continu, on a

$$\dot{x}_u(t) = f(t, x_u(t), u(t)), \quad J(u) = \int_0^T g(t, x_u(t), u(t)) dt + h(x_u(T))$$

- Le lien avec l'approche en temps discret se fait en considérant $x_m \approx x_u(t_m)$, $u_m \approx u(t_m)$
- Schéma d'Euler explicite pour la dynamique (pas de temps $\Delta t_m = t_{m+1} - t_m$)
 $x_u(t_{m+1}) = x_u(t_m) + \int_{t_m}^{t_{m+1}} f(t, x_u(t), u(t)) dt = x_u(t_m) + \Delta t_m f(t_m, x_u(t_m), u(t_m)) + o(\Delta t)$

$$F_m(y, v) = y + \Delta t_m f(t_m, y, v), \quad \forall (y, v) \in \mathbb{R}^d \times U$$

- Formule des rectangles pour le critère

$$G_m(y, v) = \Delta t_m g(t_m, y, v), \quad \forall (y, v) \in \mathbb{R}^d \times U$$

et la fonction h est la même que dans le cas continu

Fonction valeur

- La **fonction valeur** pour le problème ci-dessus est $V_0 : \mathbb{R}^d \rightarrow \mathbb{R}$ t.q.

$$V_0(x) = \inf_{(u_0, \dots, u_{N-1}) \in U^N} J_0(x; u_0, \dots, u_{N-1})$$

- On va plonger ce problème dans une **famille** de problèmes de contrôle optimal en temps discret **paramétrée par** $n \in \{0:N-1\}$ et $x \in \mathbb{R}^d$
 - la dynamique démarre à l'instant t_n avec l'état x

$$x_{m+1} = F_m(x_m, u_m), \quad m \in \{n:N-1\}, \quad x_n = x$$

- le critère est $J_n : \mathbb{R}^d \times U^{N-n} \rightarrow \mathbb{R}$ t.q.

$$J_n(x; u_n, \dots, u_{N-1}) = \sum_{m \in \{n:N-1\}} G_m(x_m, u_m) + h(x_N)$$

- la fonction valeur est $V_n : \mathbb{R}^d \rightarrow \mathbb{R}$ t.q.

$$V_n(x) = \inf_{(u_n, \dots, u_{N-1}) \in U^{N-n}} J_n(x; u_n, \dots, u_{N-1})$$

- Enfin, on définit $V_N : \mathbb{R}^d \rightarrow \mathbb{R}$ t.q. $V_N(x) = h(x)$

Programmation dynamique

- Le plongement opéré ci-dessus nous conduit à chercher la famille de fonctions valeur $(V_n)_{n \in \{0:N\}}$ sachant que
 - $V_n : \mathbb{R}^d \rightarrow \mathbb{R}, \forall n \in \{0:N\}$
 - $V_N(x) = h(x)$ est connue explicitement
- L'idée est qu'il est possible d'obtenir ces fonctions par la résolution d'une équation, dite de **programmation dynamique** (en temps discret)
- **(Prop. 7.3)** Les fonctions valeur satisfont l'équation fonctionnelle

$$V_n(x) = \inf_{v \in U} \{G_n(x, v) + V_{n+1}(F_n(x, v))\}, \quad \forall x \in \mathbb{R}^d, \forall n \in \{0:N-1\}$$

⇒ **récurrence rétrograde en n** à partir de la condition finale $V_N(x) = h(x)$

- problèmes de minimisation bien posés si, p. ex., U est compact, F_n, G_n continues sur $\mathbb{R}^d \times U$ et h continue sur $\mathbb{R}^d$; V est alors continue sur $\mathbb{R}^d$
- on obtient les **contrôles optimaux** $\tilde{u}_{N-1}(x), \dots, \tilde{u}_0(x)$ comme des **feedbacks**

$$\tilde{u}_n(x) \in \arg \min_{v \in U} \{G_n(x, v) + V_{n+1}(F_n(x, v))\}$$

- enfin, on obtient la trajectoire optimale et les contrôles optimaux

$$\bar{u}_0 = \tilde{u}_0(x_0), \bar{x}_1 = F_0(x_0, \bar{u}_0), \bar{u}_1 = \tilde{u}_1(\bar{x}_1), \bar{x}_2 = F_1(\bar{x}_1, \bar{u}_1), \bar{u}_2 = \tilde{u}_2(\bar{x}_2), \dots$$

Preuve

- Pour $n = N - 1$, comme $x_{N-1} = x$, $x_N = F_{N-1}(x, u_{N-1})$ et $V_N = h$, il vient

$$V_{N-1}(x) = \inf_{u_{N-1} \in U} \left\{ G_{N-1}(x, u_{N-1}) + h(x_N) \right\} = \inf_{v \in U} \left\{ G_{N-1}(x, v) + V_N(F_{N-1}(x, v)) \right\}$$

- Pour $n < N - 1$, on a

$$\begin{aligned} V_n(x) &= \inf_{(u_n, \dots, u_{N-1}) \in U^{N-n}} J_n(x; u_n, \dots, u_{N-1}) \\ &= \inf_{u_n \in U} \inf_{(u_{n+1}, \dots, u_{N-1}) \in U^{N-n-1}} \left\{ \sum_{m \in \{n: N-1\}} G_m(x_m, u_m) + h(x_N) \right\} \\ &= \inf_{u_n \in U} \left\{ G_n(x, u_n) + \inf_{(u_{n+1}, \dots, u_{N-1}) \in U^{N-n-1}} \left\{ \sum_{m \in \{n+1: N-1\}} G_m(x_m, u_m) + h(x_N) \right\} \right\} \\ &= \inf_{u_n \in U} \left\{ G_n(x, u_n) + V_{n+1}(x_{n+1}) \right\} = \inf_{u \in U} \left\{ G_n(x, u) + V_{n+1}(F_n(x, u)) \right\} \end{aligned}$$

- L'argument essentiel est que le critère est **additif** le long des trajectoires

Application au système LQ en temps discret (1)

- La dynamique discrète s'écrit, avec $\mathcal{A} \in \mathbb{R}^{d \times d}$ et $\mathcal{B} \in \mathbb{R}^{d \times k}$,

$$x_{m+1} = \mathcal{A}x_m + \mathcal{B}u_m, \quad m \in \{0:N-1\}, \quad x_0 = x \in \mathbb{R}^d$$

i.e., $F(y, v) = \mathcal{A}y + \mathcal{B}v$

- Le critère à minimiser s'écrit, avec $\mathcal{R} \in \mathbb{R}^{k \times k}$ symétrique > 0 et $\mathcal{Q}, \mathcal{D} \in \mathbb{R}^{d \times d}$ symétriques ≥ 0 ,

$$J_0(x; u_0, \dots, u_{N-1}) = \sum_{m \in \{0:N-1\}} \left\{ \frac{1}{2} u_m^\dagger \mathcal{R} u_m + \frac{1}{2} x_m^\dagger \mathcal{Q} x_m \right\} + \frac{1}{2} x_N^\dagger \mathcal{D} x_N$$

i.e., $G(y, v) = \frac{1}{2} v^\dagger \mathcal{R} v + \frac{1}{2} y^\dagger \mathcal{Q} y$ et $h(y) = \frac{1}{2} y^\dagger \mathcal{D} y$
(pour simplifier, on considère un état cible nul)

- Il n'y a pas de contraintes sur le contrôle : $U = \mathbb{R}^k$

Application au système LQ en temps discret (2)

- L'équation de programmation dynamique est

$$V_n(x) = \inf_{u \in \mathbb{R}^k} \left\{ \frac{1}{2} u^\top \mathcal{R} u + \frac{1}{2} x^\top \mathcal{Q} x + V_{n+1}(\mathcal{A}x + \mathcal{B}u) \right\}$$

avec la condition finale $V_N(x) = \frac{1}{2} x^\top D x$

- Par récurrence rétrograde, on montre que la fonction valeur est de la forme $V_n = \frac{1}{2} x^\top P_n x$ avec P_n **symétrique semi-définie positive** et

$$P_N = D, \quad P_n = \mathcal{A}^\top P_{n+1} \mathcal{A} - \mathcal{E}_{n+1}^\top (\mathcal{F}_{n+1})^{-1} \mathcal{E}_{n+1} + \mathcal{Q}, \quad \forall n \in \{0:N-1\}$$

avec $\mathcal{E}_{n+1} = \mathcal{B}^\top P_{n+1} \mathcal{A}$ et $\mathcal{F}_{n+1} = \mathcal{R} + \mathcal{B}^\top P_{n+1} \mathcal{B}$

- pour trouver le contrôle optimal $\tilde{u}_n$ associé à la fonction valeur V_n , il suffit de considérer $u \mapsto u^\top \mathcal{R} u + (\mathcal{A}x + \mathcal{B}u)^\top P_{n+1} (\mathcal{A}x + \mathcal{B}u)$
- en utilisant l'hypothèse de récurrence sur la symétrie de P_{n+1} et en réarrangeant, on considère $u \mapsto u^\top \mathcal{F}_{n+1} u + 2u^\top \mathcal{E}_{n+1} x$
- $\mathcal{F}_{n+1}$ est > 0 car $\mathcal{R} > 0$ et $\mathcal{B}^\top P_{n+1} \mathcal{B} \geq 0$
- par minimisation quadratique **sur** $\mathbb{R}^k$, il vient $\tilde{u}_n(x) = -(\mathcal{F}_{n+1})^{-1} \mathcal{E}_{n+1} x$
- en reportant dans la définition de V_n , on obtient l'expression de P_n
- cette expression montre que P_n est **symétrique**; de plus $V_n(x) \geq 0$ (car $J_n(x; u_n, \dots, u_N) \geq 0$); d'où $P_n \geq 0$

Liens avec le système LQ en temps continu

- Le système LQ en temps continu est régi par la dynamique

$$\dot{x}_u(t) = Ax_u(t) + Bu(t), \quad \forall t \in [0, T], \quad x_u(0) = x$$

et le critère à minimiser est, en prenant un état cible nul,

$$J(u) = \int_0^T \left\{ \frac{1}{2} u(t)^\dagger Ru(t) + \frac{1}{2} x_u(t)^\dagger Qx_u(t) \right\} dt + \frac{1}{2} x_u(T)^\dagger Dx_u(T)$$

- On considère les instants discrets $0 = t_0 < t_1 < \dots < t_N = T$
- Le lien se fait en considérant $x_m \approx x_u(t_m)$, $u_m \approx u(t_m)$
 - on suppose pour simplifier que le pas de temps Δt est constant
 - schéma d'Euler explicite pour la dynamique : $A = I + \Delta tA$, $B = \Delta tB$
 - formule des rectangles pour le critère : $R = \Delta tR$, $Q = \Delta tQ$

Retour vers l'équation de Riccati

- Soit $P : [0, T] \rightarrow \mathbb{R}^{d \times d}$ de classe C^1 t.q. $P(t_n) = P_n, \forall n \in \{0:N\}$,
On choisit Δt constant pour simplifier
- En $t_N = T$, on a $P(T) = D$
- En effectuant des développements de Taylor en temps, il vient
 - $\mathcal{A}^\dagger P_{n+1} \mathcal{A} = P(t_n) + \Delta t (\mathcal{A}^\dagger \dot{P}(t_n) + P(t_n) \mathcal{A} + \dot{P}(t_n)) + o(\Delta t)$
 - $\mathcal{E}_{n+1} = \mathcal{B}^\dagger P_{n+1} \mathcal{A} = \Delta t \mathcal{B}^\dagger P(t_n) + o(\Delta t)$
 - $\mathcal{F}_{n+1} = \mathcal{R} + \mathcal{B}^\dagger P_{n+1} \mathcal{B} = \Delta t R + o(\Delta t)$
- Comme $P_n = \mathcal{A}^\dagger P_{n+1} \mathcal{A} - \mathcal{E}_{n+1}^\dagger (\mathcal{F}_{n+1})^{-1} \mathcal{E}_{n+1} + Q$, il vient en simplifiant par $P(t_n)$ et en divisant par Δt ,

$$\dot{P}(t_n) = -\mathcal{A}^\dagger P(t_n) - P(t_n) \mathcal{A} + P(t_n) \mathcal{B} R^{-1} \mathcal{B}^\dagger P(t_n) - Q + o(1)$$

qui est, à $o(1)$ près, l'équation de Riccati pour le problème en temps continu écrite à l'instant t_n

Retour vers le Hamiltonien (1)

- Pour le système LQ en temps continu, le Hamiltonien est

$$H(x, p, u) = p^\dagger (Ax + Bu) + \frac{1}{2} u^\dagger Ru + \frac{1}{2} x^\dagger Qx$$

- Le Hamiltonien minimisé est

$$H_b(x, p) = \min_{u \in \mathbb{R}^k} H(x, p, u)$$

Le Hamiltonien minimisé a bien un sens : on résout un problème de minimisation d'une fonctionnelle fortement convexe sur $\mathbb{R}^k$ ($R > 0$)

- Dans le même esprit que ci-dessus, cherchons (formellement) une équation pour la fonction valeur du problème LQ en temps continu
- Soit $V : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}$ une fonction de classe C^1 en (t, x) t.q.

$$V(t_n, x) = V_n(x), \quad \forall n \in \{0:N\}, \forall x \in \mathbb{R}^d$$

Retour vers le Hamiltonien (2)

- On rappelle que $V_n(x) = \inf_{u \in \mathbb{R}^k} \left\{ \frac{1}{2} u^\dagger \mathcal{R} u + \frac{1}{2} x^\dagger \mathcal{Q} x + V_{n+1}(\mathcal{A}x + \mathcal{B}u) \right\}$
 - on a également $\mathcal{A} = I + \Delta t A$, $\mathcal{B} = \Delta t B$, $\mathcal{R} = \Delta t R$, $\mathcal{Q} = \Delta t Q$

- Un développement de Taylor montre que

$$\begin{aligned} V_{n+1}(\mathcal{A}x + \mathcal{B}u) - V_n(x) &= V(t_n + \Delta t, x + \Delta t(Ax + Bu)) - V(t_n, x) \\ &= \Delta t \frac{\partial V}{\partial t}(t_n, x) + \Delta t \frac{\partial V}{\partial x}(t_n, x)^\dagger (Ax + Bu) + o(\Delta t) \end{aligned}$$

- En réarrangeant, en divisant par Δt et en introduisant le Hamiltonien minimisé, il vient

$$\frac{\partial V}{\partial t}(t_n, x) + H_b(x, \frac{\partial V}{\partial x}(t_n, x)) = o(1)$$

- Nous verrons la prochaine fois que la fonction valeur $V : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}$ du problème LQ en temps continu satisfait l'**équation de Hamilton–Jacobi–Bellman (HJB)**

$$\frac{\partial V}{\partial t}(t, x) + H_b(x, \frac{\partial V}{\partial x}(t, x)) = 0, \quad \forall t \in [0, T], \forall x \in \mathbb{R}^d$$

avec la condition en temps final $V(T, x) = \frac{1}{2} x^\dagger D x$

Optimisation combinatoire

- La programmation dynamique a été introduite dans les années 50 par R. Bellman (1920-84)
- Son champ d'applications est **bien plus vaste** que les problèmes de contrôle optimal en temps discret
 - problème du plus court chemin, affectation de ressources (et plus généralement, la **théorie des graphes** et la **recherche opérationnelle**)
 - alignement de séquences en **bio-informatique**
 - détection de ruptures en **analyse statistique** (estimer les instants où un signal présente des changements dans la distribution)
- Heuristiquement, le principe d'optimalité de Bellman est le suivant
 - toute solution optimale résulte de sous-problèmes résolus localement de façon optimale
 - p.ex., lorsqu'on parcourt une trajectoire optimale, à tout instant un contrôle optimal pour le problème restant est celui associé à la trajectoire optimale
 - on obtient ainsi une solution optimale en combinant des solutions optimales d'une série de sous-problèmes

Exemple : problème du sac à dos (1)

- On considère un sac à dos de capacité Q et N objets, énumérés de 1 à N , de valeur v_n et encombrement e_n , $\forall n \in \{1:N\}$
- On pose le problème de remplir le sac à dos en maximisant sa valeur $\sum_{n \in \{1:N\}} u_n v_n$ sous la contrainte $\sum_{n \in \{1:N\}} u_n e_n \leq Q$
 - $u_n = 1$ si l'objet est sélectionné, $u_n = 0$ sinon
 - le problème de maximisation est donc

$$\max_{\substack{(u_1, \dots, u_N) \in \{0,1\}^N \\ \sum_{n \in \{1:N\}} u_n e_n \leq Q}} \left\{ \sum_{n \in \{1:N\}} u_n v_n \right\}$$

- ce problème pourrait se résoudre en considérant les 2^N possibilités...
- La programmation dynamique permet de résoudre ce problème bien plus efficacement
 - on suppose que Q et les encombrements e_n sont des entiers
 - l'état du sac à dos est décrit par un entier $q \in \{0:Q\}$
 - on considère la fonction valeur $V_n : \{0:Q\} \rightarrow \mathbb{R}$, où $V_n(q)$ est la valeur optimale d'un sac à dos de capacité q pour les objets énumérés de n à N

Problème du sac à dos (2)

- Pour $n = N$ (seul l'objet d'indice N est considéré), on a

$$V_N(q) = \begin{cases} v_N & \text{si } e_N \leq q \\ 0 & \text{sinon} \end{cases} \quad \tilde{u}_N(q) = \begin{cases} 1 & \text{si } e_N \leq q \\ 0 & \text{sinon} \end{cases}$$

Noter que le contrôle optimal est obtenu comme un feedback p.r. à l'état q

- On stocke la fonction valeur et le contrôle optimal dans des tableaux à $(Q + 1)$ lignes et N colonnes
- Exemple numérique : $Q = 6$, $N = 3$, valeurs $(3, 3, 5)$, encombrements $(3, 3, 4)$

$V_n(q)$	1	2	3
6			5
5			5
4			5
3			0
2			0
1			0
0			0

$\tilde{u}_n(q)$	1	2	3
6			1
5			1
4			1
3			0
2			0
1			0
0			0

Problème du sac à dos (3)

- L'équation de programmation dynamique s'écrit

$$V_n(q) = \max_{\substack{u \in \{0,1\} \\ q \geq ue_n}} \left\{ uv_n + V_{n+1}(q - ue_n) \right\}$$

- $\tilde{u}_n(q) = 0$ si $e_n > q$ ou si $e_n \leq q$ et $V_{n+1}(q) > V_{n+1}(q - e_n) + v_n$
 - $\tilde{u}_n(q) = 1$ si $e_n \leq q$ et $V_{n+1}(q) < V_{n+1}(q - e_n) + v_n$
 - $\tilde{u}_n(q) \in \{0, 1\}$ si $e_n \leq q$ et $V_{n+1}(q) = V_{n+1}(q - e_n) + v_n$
- En résumé, on a

$$V_n(q) = \begin{cases} V_{n+1}(q) & q < e_n \\ \max(V_{n+1}(q), v_n + V_{n+1}(q - e_n)) & q \geq e_n \end{cases}$$

et le contrôle optimal est

- $\tilde{u}_n(q) = 0$ dans le premier cas
- $\tilde{u}_n(q) = 0$ dans le deuxième cas si $V_{n+1}(q) < v_n + V_{n+1}(q - e_n)$
- $\tilde{u}_n(q) = 1$ dans le deuxième cas si $V_{n+1}(q) > v_n + V_{n+1}(q - e_n)$
- $\tilde{u}_n(q) \in \{0, 1\}$ dans le deuxième cas si $V_{n+1}(q) = v_n + V_{n+1}(q - e_n)$

Problème du sac à dos (4)

- On remplit les tableaux donnant $V_n(q)$ et $\tilde{u}_n(q)$ **de la droite vers la gauche**

$V_n(q)$	1	2	3
6	6	5	5
5	5	5	5
4	5	5	5
3	3	3	0
2	0	0	0
1	0	0	0
0	0	0	0

$\tilde{u}_n(q)$	1	2	3
6	1	0	1
5	0	0	1
4	0	0	1
3	{0,1}	1	0
2	0	0	0
1	0	0	0
0	0	0	0

- commande optimale $(\tilde{u}_1(6), \tilde{u}_2(3), \tilde{u}_3(0)) = (1, 1, 0)$, valeur optimale de 6
- on a effectué QN comparaisons (comparer à 2^N)
- l'**algorithme glouton** (choisir les objets par valeur décroissante) donne la commande $(0, 0, 1)$ et la valeur de 5 qui est **sous-optimale**