

Notes de cours : Commande Optimale de systèmes en temps discret

Eric Parent, Gabriel Lang, Michel Cohen de Lara

November 28, 2010

1 Préface: Pourquoi la commande optimale?

Modéliser un système, c'est d'abord mettre en évidence l'importance des variables d'état pour analyser la structure même de sa dynamique (stabilité, observabilité, contrôlabilité) en réglant la matrice de gain du feedback. Le rôle de l'ingénieur est alors d'étudier la réponse du système ouvert puis d'agir sur celui-ci (de façon plus ou moins heureuse ...) par une boucle de retour d'état: il déplace les points d'équilibre, change leur nature, etc. L'utilisation efficace du placement direct des pôles ne s'acquiert qu'avec la pratique et la connaissance approfondie des mécanismes du système que l'on veut asservir. Placer les pôles trop loin de l'origine peut engendrer une réponse dynamique extrêmement rapide pour laquelle la valeur des commandes à appliquer peut dépasser la puissance des actionneurs que l'on possède. Même si on disposait de cette puissance, l'utilisation d'une matrice de gain importante s'accompagne souvent de bruits qui, eux aussi, sont du même coup amplifiés. Du point de vue des applications pratiques, on a vu que des théorèmes d'approximations nous garantissaient la pertinence d'un travail sur le système linéarisé autour d'un équilibre à imposer (...du moins dans un certain voisinage que l'on espère pas trop restreint!). En conséquence, quoique l'on ne dispose en général que d'actionneurs de puissance limitée, quand les commandes saturent leur gamme de variation admissible, la linéarisation du comportement du système que l'on avait effectuée devient alors inappropriée! L'utilisation d'un critère à optimiser, qui s'introduit d'ailleurs naturellement par des considérations énergétiques, permet en général d'éviter ces situations plus facilement qu'un "réglage à la main" du placement des pôles.

Une seconde justification pour l'emploi de ces techniques issues de la théorie de la commande optimale est que le système à contrôler peut ne pas être commandable au sens défini dans les chapitres précédents. Dans ce cas, bien sûr sans exiger l'impossible pour la trajectoire du système, la théorie développée ci-après fournira une commande qui agira sur le système autant que faire se peut : si la partie non commandable de l'état est stable, le système dans son ensemble se comportera de façon acceptable.

Nous adopterons toujours le point de vue, déjà largement abordé dans ce cours, d'un modèle dynamique décrit par son équation d'état :

- ou en temps discret par l'équation aux différences finies :

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), u(t), t), \quad t \in \mathcal{Z} \quad (1)$$

- en temps continu par l'équation différentielle :

$$\dot{\mathbf{x}} = f(\mathbf{x}, u, t), \quad t \in \mathcal{R} \quad (2)$$

Dans ces équations, x est le vecteur d'état à l'instant t décrivant les grandeurs caractéristiques du modèle qui évoluent dynamiquement par l'action du vecteur des commandes u selon l'équation de transition décrite par la fonction f . De plus, on se donne généralement l'état initial $x_0 = x(0)$ pour $t = 0$.

L'objet essentiel de la théorie de la commande optimale est de trouver une commande u de manière à:

- atteindre à un instant final T un état final $x(T)$ appartenant à une cible fixée C . L'instant final T peut ne pas être fixé a priori ou être rejeté à l'infini.
- minimiser un critère J dépendant de la commande $u(\cdot)$ et donc de la trajectoire $x(\cdot)$ suivie et de l'état final atteint. Ce critère J est additif par rapport à la trajectoire et se met sous la forme :

– temps discret :

$$J(u(\cdot)) = \sum_{t=0}^{T-1} L(\mathbf{x}(t), u(t), t) + V_F(\mathbf{x}(T), T) \quad (3)$$

– temps continu :

$$J(u(\cdot)) = \int_0^T L(\mathbf{x}(t), u(t), t) dt + V_F(\mathbf{x}(T), T) \quad (4)$$

On appelle dans ces équations, V_F le coût final, L le coût instantané.

Signalons le cas particulier où l'instant final n'est pas fixé et où L vaut 1, cas relatif au problème de commande en temps minimum.

Signalons la place particulière au critère quadratique :

$$L(\mathbf{x}(t), u(t), t) = (u(t) - u_c(t))' R(t) (u(t) - u_c(t)) \\ + (\mathbf{x}(t) - \mathbf{x}_c(t))' Q(t) (\mathbf{x}(t) - \mathbf{x}_c(t))$$

où R et Q sont des matrices de pondération définies positives. De plus, $x_c(t)$ et $u_c(t)$, fonctions du temps prédéfinies, peuvent s'interpréter comme des valeurs de "consigne" de la trajectoire et de la commande dont on désire s'écarter le moins possible.

Deux approches sont possibles pour trouver cette commande optimale pour les problèmes (3) + (1) ou (4) + (2).

La première voie est de rechercher une loi de commande en feedback $u = u(x, t)$, la seule effectivement utilisable pour le temps réel quand le système est soumis à des perturbations et pour les questions de stabilité et de commandabilité déjà évoquées dans les chapitres précédents. Elle sera développée dans la première partie de ce chapitre : la programmation dynamique en temps discret permet d'illustrer simplement le principe de Bellman.

La seconde approche considère ce problème (4) + (2) en tant que problème d'optimisation sur l'ensemble des fonctions continues par morceaux $t \mapsto u(t)$. Elle consiste à généraliser la notion de dérivée (Principe du minimum de Pontryagin) du critère le long de trajectoires suivies par le système. Cette méthode variationnelle fournit une loi de commande sous la forme d'une boucle ouverte: $u = u(t)$ ce qui n'offre a priori aucun intérêt opérationnel pour la gestion d'un système en temps réel. Néanmoins, nous la présentons dans la dernière partie de ce cours à cause de ses liens étroits avec la première approche, et parce ce calcul de variations appartient à la culture de l'ingénieur d'aujourd'hui. Cet outil mathématique a donné lieu à de nombreuses applications, notamment en mécanique. Par souci de commodité et de simplification, nous ne développons pas ces deux approches en temps continu, mais à temps discret.

2 La Programmation Dynamique en Temps Discret

2.1 Exemple Illustratif

La programmation dynamique est plus une approche conceptuelle des problèmes d'optimisation séquentielle qu'une technique calculatoire d'optimisation comme la programmation linéaire ou le gradient réduit généralisé. Cette approche consiste à convertir un problème d'optimisation de plusieurs étapes (ou couvrant plusieurs périodes de temps) en une série de problèmes d'une étape (ou période) sans changer la nature du problème.

Considérons le problème du voyageur qui doit se rendre de la ville G à la ville A en cinq étapes en effectuant un chemin de bénéfice maximal.

A chaque ville d'étape de la figure 1 le voyageur reporte en ce point le chemin qui lui reste à parcourir et marque d'un double trait le tronçon le plus intéressant.

Suivons son raisonnement sur les figures 2 et suivantes:

Soit $V(x, i)$, le bénéfice maximal entre la ville x située au début de l'étape i et la ville A; $L(x, u, i)$, le gain immédiat entre les villes x et u à l'étape i .

On voit donc que le raisonnement séquentiel de notre voyageur peut être formalisé sous la forme :

$$V(x, i) = \max_u (L(x, u, i) + V(u, i + 1))$$

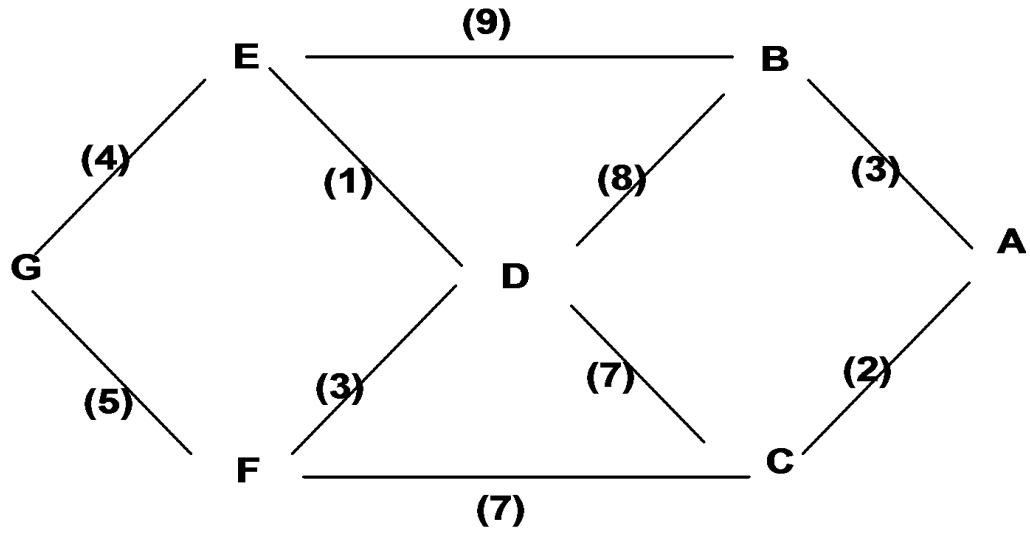


Figure 1: Problème du voyageur de commerce

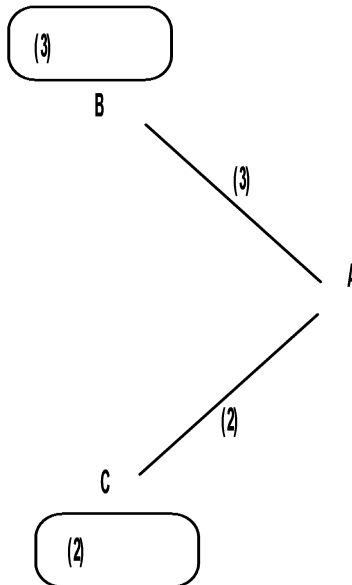


Figure 2: Problème du voyageur de commerce, étape 1

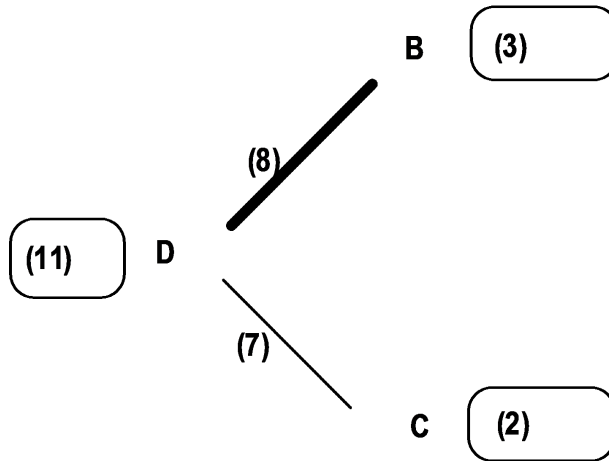


Figure 3: Problème du voyageur de commerce, étape 2

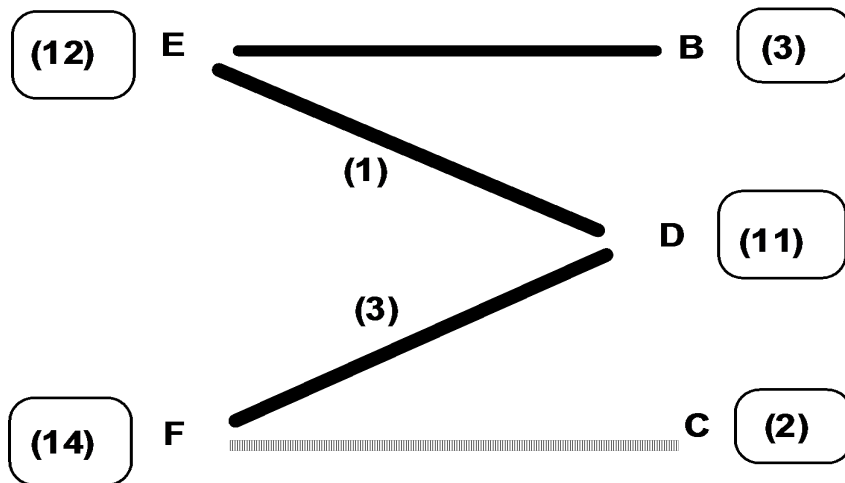


Figure 4: Problème du voyageur de commerce, étape 3

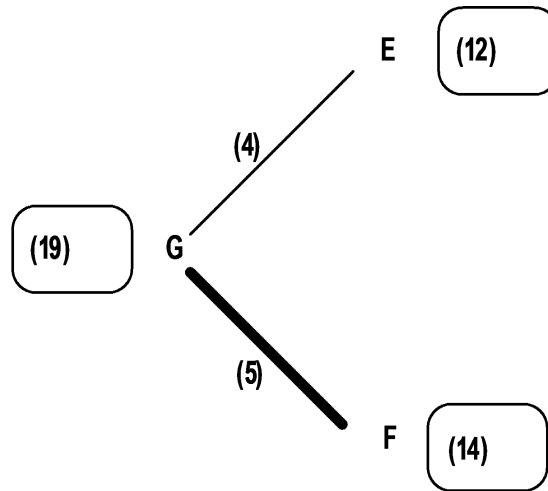


Figure 5: Problème du voyageur de commerce, étape 4

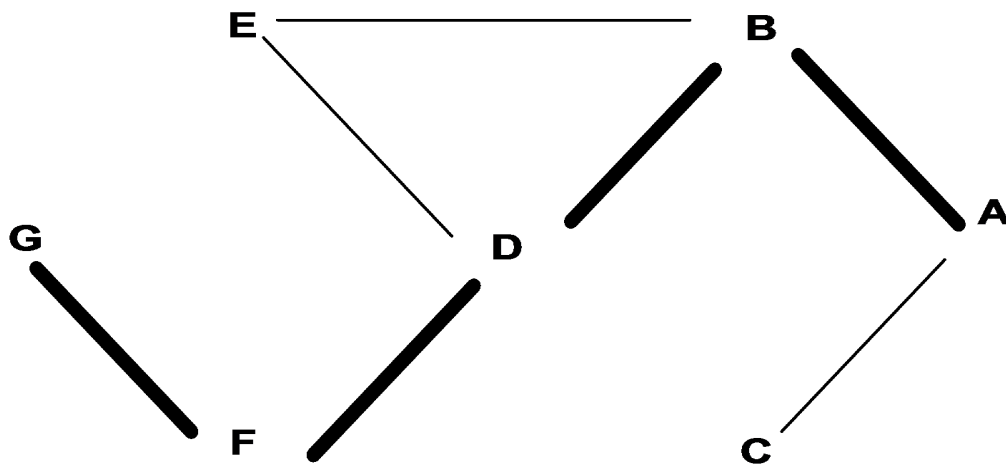


Figure 6: Problème du voyageur de commerce, décisions optimales

et que cette procédure fournit une solution en boucle fermée $u = u(x, i)$, c'est à dire que notre voyageur ne se décide qu'en fonction de la ville x où il se trouve à l'étape i , en oubliant le détail de son itinéraire passé et sans chercher à distinguer son itinéraire futur à long terme.

2.2 Principe de Bellman à temps discret

Definition 1 *Principe d'optimalité de BELLMAN:*

Une suite de commandes optimales dans un processus d'optimisation dynamique est telle que, quels que soient l'état et l'instant considérés sur une trajectoire optimale, les commandes ultérieures constituent, pour le problème ayant cet état et cet instant comme phase initiale, une suite de commandes optimales.

Ce principe exprime le fait qu'avec les mêmes conditions initiales données, toute "sous-politique" d'une "politique" optimale est elle-même optimale (sinon en raisonnant par l'absurde, on trouverait une solution meilleure que la politique optimale).

Pour formaliser ce raisonnement, posons, pour l'équation 1, avec les conditions aux limites à $t = 0, x(0) = x_0$ et à $t = T, x = x(T)$:

$$\inf J(u(.)) = \sum_{t=0}^{T-1} L(\mathbf{x}(t), u(t), t) dt + V_F(\mathbf{x}(T), T)$$

la fonction $V(x_1, t_1) = \inf J(u(.))$ dite fonction de Bellman pour le problème, avec $t_1 > 0$, à $t = 0, x(0) = x_0$ et à $t = T, x = x(T)$:

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), u(t), t) \quad (5)$$

$$\inf J(u(.)) = \sum_{t=t_1}^{T-1} L(\mathbf{x}(t), u(t), t) dt + V_F(\mathbf{x}(T), T) \quad (6)$$

Notons qu'ont été élucidées les questions de commandabilité, c'est à dire qu'avec ces conditions initiales, on suppose que l'on puisse atteindre le point $(T, x(T))$.

On peut alors écrire :

$$V(\mathbf{x}(t), t) = \min_{u(t), u(t+1), \dots, u(T-1)} \sum_{\tau=t}^{T-1} L(\mathbf{x}(\tau), u(\tau), \tau) + V_F(\mathbf{x}(T), T)$$

$$V(\mathbf{x}(t), t) = \min_{u(t)} \min_{u(t+1), \dots, u(T-1)} L(\mathbf{x}(t), u(t), t) + \sum_{\tau=t+1}^{T-1} L(\mathbf{x}(\tau), u(\tau), \tau) + V_F(\mathbf{x}(T), T)$$

Comme $L(x(t), u(t), t)$ ne dépend pas des commandes ultérieures, il vient:

$$V(\mathbf{x}(t), t) = \min_{u(t)} L(\mathbf{x}(t), u(t), t) + V(\mathbf{x}(t+1), t+1)$$

et, avec ces notations, en faisant apparaître explicitement la dépendance de $x(t+1)$ à $x(t), u(t)$ par l'équation d'état récurrente de premier ordre, il vient alors:

$$V(\mathbf{x}(t), t) = \min_{u(t)} (L(\mathbf{x}(t), u(t), t) + V(f(\mathbf{x}(t), u(t), t), t+1))$$

Comme on sait qu'à T on a $V_F(x(T), T)$ pour coût aux limites, la résolution séquentielle proposée par la programmation dynamique consiste à :

- intégrer l'équation de Bellman à partir de ses conditions aux limites;
- trouver la commande en boucle fermée et l'utiliser pour calculer la fonction de Bellman au pas de temps où l'on se trouve;
- ce dernier calcul donne une nouvelle condition aux limites, ce qui permet d'itérer le processus de calcul.

On note que ces calculs ont été faits en sens rétrograde par rapport au temps. De fait, le lecteur intéressé peut établir des formules de Bellman analogues en inversant le sens du temps, sous la seule réserve que l'équation d'évolution puisse s'inverser aussi sous la forme : $x(t) = h(x(t+1), u(t), t)$

En conclusion nous pouvons exprimer ces résultats et leur caractère de condition nécessaire sous la forme du théorème suivant:

Theorem 2 *Pour le problème (1) + (3), si l'équation fonctionnelle d'Hamilton Jacobi Bellman a une solution avec la condition aux limites $V(x(T), T) = V_F(x(T), T)$ donné, alors le problème a une solution et la loi de commande optimale est donnée par la commande en feedback solution de l'équation fonctionnelle: $u(x, t) = \arg \min L(x, u, t) + V(f(x, u, t), t+1)$*

2.3 Exemple d'allocation optimale de ressources

D'après [5].

Supposons qu'à chaque instant $i = 1, 2, 3, 4$, on dispose des ressources $x(i) \in \mathcal{R}^+$, et que l'on puisse affecter ces ressources au cours de la période $(i, i+1)$ entre deux usages :

- l'usage(a) qui rapporte le gain 2 par unité de ressources affectées, mais qui entraîne une disparition (amortissement) de 20 pour cent des ressources affectées.
- l'usage(b) qui rapporte davantage : gain 3 sur la période par unité affectée, mais pour lequel l'amortissement est plus grand: 50 pour cent.

La décision à prendre à l'instant i est donc la quantité de ressources $u(i) \in [0, x(i)]$ à affecter à l'usage (a), $x(i) - u(i)$ étant affecté à (b).

Les équations régissant ce modèle sont donc, pour l'équation de transition:

$$\mathbf{x}(t+1) = 0.8u(t) + 0.5(\mathbf{x}(t) - u(t)) \quad (7)$$

$$= 0.5(\mathbf{x}(t) + 0.3u(t)) \quad (8)$$

et pour le critère :

$$L(\mathbf{x}, u, t) = -3\mathbf{x} + u$$

Appliquant le principe de Bellman, avec x désignant la ressource disponible à cette étape, on peut donc calculer la perte V optimale à chaque étape:

- étape 4 à 5

$$V(\mathbf{x}, 4) = \min_{u; 0 \leq u \leq \mathbf{x}} (u - 3\mathbf{x}) = -3\mathbf{x} \quad \text{et} \quad \text{donc} \quad u(\mathbf{x}, 4) = 0$$

- étape 3 à 4

$$V(\mathbf{x}, 3) = \min_{u; 0 \leq u \leq \mathbf{x}} (-3\mathbf{x} + u - 3(0.5\mathbf{x} + 0.3u)) \quad \text{et} \quad \text{donc} \quad u(\mathbf{x}, 3) = 0$$

- étape 2 à 3

$$V(\mathbf{x}, 2) = \min_{u; 0 \leq u \leq \mathbf{x}} (-3\mathbf{x} + u - 4.5(0.5\mathbf{x} + 0.3u)) \quad \text{et} \quad \text{donc} \quad u(\mathbf{x}, 2) = \mathbf{x}$$

- étape 1 à 2

$$V(\mathbf{x}, 1) = \min_{u; 0 \leq u \leq \mathbf{x}} (u - 3\mathbf{x} - 5.6(0.5\mathbf{x} + 0.3u)) = 6.48 \mathbf{x} \quad \text{et} \quad \text{donc} \quad u(\mathbf{x}, 1) = \mathbf{x}$$

Le lecteur aura reconnu que ce problème, élégamment résolu par la programmation dynamique, peut se mettre sous forme de programme linéaire à cause des objectifs, des contraintes et de l'équation d'évolution, tous linéaires. Notons au passage que la solution obtenue se comporte avec des solutions en tout ou rien typiques des programmes linéaires : les solutions $u(1), u(2), u(3), u(4)$ sont les valeurs extrémales qui saturent les contraintes, c'est pourquoi, compte tenu des gains marginaux constants, on affecte toutes les ressources soit en totalité sur (a), soit en totalité sur (b).

2.4 De la difficulté de formuler un modèle de programmation dynamique

La partie la plus difficile dans l'utilisation des techniques de Bellman est sans doute de définir correctement le modèle, c'est à dire:

- définir l'état,
- recenser les commandes,
- traduire le comportement dynamique du système par les équations d'état qui décrivent les transitions et les mécanismes d'évolution de l'objet d'étude,
- ne pas oublier d'examiner les contraintes sur le vecteur d'état et le vecteur de commande,
- déterminer un critère d'optimisation sous forme additive,
- choisir le pas de temps.

A cet égard, l'exemple de la colonne à distiller [4, p 76], décrit dans l'exemple représenté par le schéma 7 ci-après, est significatif car la modélisation adoptée ne respecte pas un pas de temps homogène. Une quantité M de matière fermentescible a été initialement placée dans l'alambic. Sous l'action d'une flamme, réglée pour maintenir une pression constante dans la colonne de distillation, le liquide boût, ce qui se traduit par une accélération de la fermentation et une perte de matière par évaporation entraînant des vapeurs chargées d'alcool. Le courant de vapeur traverse une valve qui le sépare en deux parties: un jet retourne à l'alambic, l'autre débouche dans un récipient à l'extérieur qui récupère le produit fini alcoolisé.

Il s'agit d'exécuter la conversion du produit initial en produit alcoolisé d'un titre d'alcool donné en un minimum de temps en jouant sur les réglages de la valve. Pour comprendre son fonctionnement, il est instructif d'examiner ce qui se passe pour les commandes en butée: valve complètement fermée, la solution se concentre en alcool mais aucune production n'est assurée; valve complètement ouverte, la quantité brute évaporée est maximale mais son titre est trop faible en général.

On définira l'étape k lorsque exactement $k - 1$ kg de produits auront été fabriqués.

Appelons:

d_n = le pourcentage de la masse évaporée retourné via la valve quand n kg de produits ont été fabriqués,

c_n = nombre de kg de matière non fermentée dans l'alambic à la période n ,

a_n = nombre de kg d'alcool dans l'alambic,

b_n = nombre de kg d'alcool produit,

M = masse initiale.

La conservation de la masse permet d'écrire une première relation d'état:

$$M = a_n + c_n + (n - 1)$$

Si, ayant étudié la physique du phénomène, on note:

$t(a_n, c_n, d_n)$: le temps nécessaire à la fabrication d'un kg de produit connaissant a_n, c_n, d_n ,

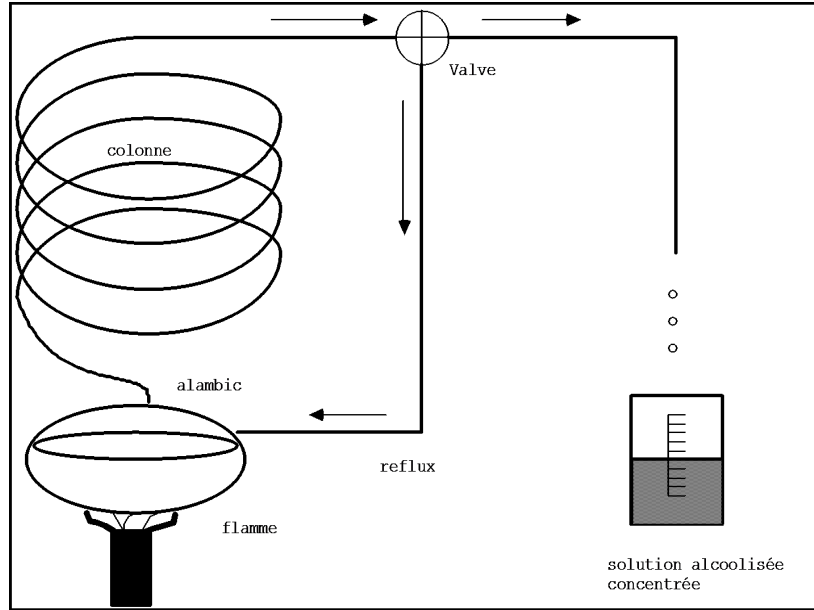


Figure 7: Colonne à distiller

$h(a_n, c_n)$: le titre du n^{eme} kg produit,
 $g(a_n, c_n, d_n)$: la variation de masse d'alcool produit entre la période n et $n + 1$,
 on peut établir les relations de conservation:

$$b_{n+1} = b_n + h(a_n, c_n)$$

tandis que les bilans de fermentation-évaporation imposent que:

$$a_{n+1} = a_n + g(a_n, c_n, d_n) - h(a_n, c_n)$$

$$c_{n+1} = c_n - g(a_n, c_n, d_n) + h(a_n, c_n) - 1$$

L'objectif de production en temps minimum s'écrit:

$$V(a_n, b_n, n) = \min_{d(1), d(2), \dots, d(n)} \sum_{k=1}^{n-1} t(a_k, c_k, d_k)$$

Clairement dans notre modèle, la commande est monodimensionnelle et vaut $u(k) = d_k$.

Le vecteur d'état de dimension 2 seulement vaut, quant à lui, $x_k = (a_k, b_k)$ puisque a_k et c_k sont liés par l'équation de conservation de la masse totale.

On aurait pu aussi bien prendre (c_k, b_k) comme vecteur d'état.

Enfin, du point de vue strictement économique la quantité a_n d'alcool non extrait à l'étape n est sans intérêt et il s'agit de réaliser:

$$W(b_n, n) = \min_{a_n} V(a_n, b_n, n)$$

Pourtant, pour résoudre ce problème par programmation dynamique, il faut bel et bien garder la variable a_n dans le vecteur d'état tout au long des séquences d'optimisation.

2.5 Problèmes numériques de mise en oeuvre

2.5.1 Complexité numérique

Dans notre dernier exemple, la fonction de coût était quelconque et la décision optimale ainsi que la fonction de Bellman, fonctions de l'état, peuvent revêtir n'importe quelle forme. Etant donné la nature digitale des mémoires d'ordinateur, les fonctions quelconques doivent en pratique être discrétisées pour pouvoir être stockées.

Les problèmes engendrés sont de deux types :

- choix des points de discrétisation;
- complexité envahissante des calculs.

Suivant le choix du réseau pour discrétiser l'espace des commandes et l'espace des états, les résultats peuvent notablement changer. La discrétisation de l'espace des commandes (lorsque celle-ci ne consiste pas, comme dans l'exemple précédent, en un choix d'état) peut ne pas être la même que celle de l'espace des états.

Estimons la complexité des calculs si l'on avait discrétisé en M points chacune des variables de l'exemple précédent :

A chacun des n pas de temps, il faut calculer les valeurs pour tous les couples (a, b) soient M^2 calculs par pas de temps. Pour l'optimisation déterminant la commande de la valve on tentera donc M valeurs de d pour chacun des couples (a, b) , c'est dire qu'en tout on effectuera pour l'exemple élémentaire choisi : nM^3 calculs !

C'est le seul véritable inconvénient de la programmation dynamique, mais il est de taille.... C'est cette complexité exponentielle en M qui rend l'usage d'une programmation dynamique classique quasi impossible au delà de deux ou trois dimensions d'états. Pour traiter ces cas, il faut de plus choisir très astucieusement la discrétisation des espaces.

2.5.2 Solutions proposées

Nous n'entrerons pas dans le détail de ces méthodes, mais notons qu'elles consistent toutes à abandonner la recherche d'une commande calculée en feedback pour tous les états possibles du système au profit d'une trajectoire optimale

calculée en boucle ouverte, généralement obtenue par modification d'une trajectoire de référence.

Citons par ordre de complexité:

- la programmation dynamique incrémentale.

On suppose que l'on connaît une trajectoire pas trop mauvaise de départ. On isole alors une période limitée (par exemple : deux pas de temps) et on cherche si l'on peut améliorer localement les choix des commandes en fixant les valeurs des états en fin et en début de la période limitée choisie. Une fois le problème résolu par une méthode quelconque d'optimisation, on obtient une nouvelle trajectoire de référence, on fait glisser la période de modification et on recommence.

On considère qu'on est à l'optimum lorsqu'aucune trajectoire environnante n'est meilleure, c'est à dire que la trajectoire courante est la solution du problème discrétisé autour de celle-ci.

On a alors un algorithme de complexité nettement inférieure à celui obtenu en programmation dynamique classique.

Par contre, on a introduit une procédure itérative d'approximations successives de la trajectoire optimale. Il peut donc se poser un problème de convergence de la méthode et on peut converger vers une solution non optimale (minimum local).

- la programmation dynamique différentielle.

JACOBSON et MAYNE [8] lui ont consacré un ouvrage entier. La technique consiste à "quadratiser" les fonctions de coût instantané autour d'une trajectoire de référence et à utiliser les propriétés particulières de la commande optimale du système linéaire quadratique déjà étudiées.

- le principe du minimum.

C'est un outil de calcul variationnel général qui fournit une solution en boucle ouverte présentée dans la suite.

2.6 Extension de la programmation dynamique discrète au cas stochastique

Le problème stochastique peut être rapidement présenté dans le contexte du temps discret en introduisant une séquence ε_t ; $t = 0, \dots, T - 1$ de variables aléatoires indépendantes de distribution connue $\mathcal{F}(\varepsilon_t), t = 0, \dots, T - 1$.

Le système est alors décrit par l'équation d'évolution :

$$x(t+1) = f(x(t), u(t), \varepsilon_t, t); \quad t = 0, \dots, T - 1$$

L'état devient une variable aléatoire.

La condition initiale est spécifiée par une valeur initiale de x_o ou une distribution initiale \mathcal{F}_{x_o} pour x_o .

Pour une stratégie u donnée en feedback sur l'état, on montre facilement que la suite des états du système contrôlé est une chaîne markovienne. Comme

on ne connaît plus les trajectoires de l'état, mais seulement leur probabilité d'occurrence, on adopte un critère portant sur l'espérance mathématique:

$$\inf J(u(\cdot)) = \mathbb{E} \left(\sum_{t=0}^{T-1} L(\mathbf{x}(t), u(t), \varepsilon_t, t) + V_F(\mathbf{x}(T), T) \right) \quad (9)$$

En utilisant les propriétés de l'espérance conditionnelle et le fait que le processus markovien des états est sans mémoire, l'additivité du critère et des espérances permet d'utiliser, sous la forme suivante, le principe d'optimisation séquentielle de Bellman:

$$V(\mathbf{x}(t), t) = \min_{u(t)} \mathbb{E}(L(\mathbf{x}(t), u(t), \varepsilon_t, t) + \mathbb{E} \min_{u(t+1), \dots, u(T-1)} \left(\sum_{\tau=t+1}^{T-1} L(\mathbf{x}(\tau), u(\tau), \varepsilon_\tau, \tau) \right. \\ \left. + V_F(\mathbf{x}(T), T) \right)) \quad (10)$$

$$+ V_F(\mathbf{x}(T), T)) \quad (11)$$

$$V(\mathbf{x}(t), t) = \min_{u(t)} \mathbb{E} (L(\mathbf{x}(t), u(t), \varepsilon_t, t) + V(f(\mathbf{x}(t), u(t), \varepsilon_t, t), t + 1)) \quad (12)$$

Le lecteur intéressé par des compléments sur le sujet pourra se reporter aux ouvrages [4], [6] et [9] en ce qui concerne les applications d'ingénierie.

3 Commande en boucle ouverte

3.1 Situation du problème pour le Principe du Minimum

Considérons le problème d'atteindre à partir d'une phase initiale ($x(t=1) = a$) un état final $x(T)$ libre à un instant final T , lui fixé sous la forme

$$x(t+1) = x(t) + f(x(t), u(t), t), \quad t \in \mathbb{N} \quad (13)$$

$$\inf_{u \in \mathcal{U}} J(u(\cdot)) = \sum_{t=1}^T L(x(t), u(t), t) \quad (14)$$

On se limitera dans cette partie à des domaines de commande admissible \mathbb{U}_t ne dépendant pas de l'état, on supposera que l'on n'a pas de contrainte sur le vecteur d'état si ce n'est pour le temps final (à travers un coût final $L(x(T), u(T), T)$), et que $L(x(t), u(t), t)$ et $f(x(t), u(t), t)$ sont des fonctions continuellement dérivables par rapport à leurs arguments x et u .

3.2 Lemme Variationnel

Lemma 3 Soit $x = (x(1), \dots, x(t), \dots, x(T))$ une trajectoire de référence, obtenue avec loi de commande $u = (u(1), (u(2), \dots, u(t), \dots, u(T-1), u(T))$ admissible reliant la phase de départ ($x(1) = a, t = 1$) à ($x(T), T$).

La variation du critère due à une variation admissible $\delta u = (\delta u(1), \delta u(2), \dots, \delta u(t), \dots, \delta u(T-1), \delta u(T))$ de la commande, c'est à dire telle que pour δu suffisamment petit, la

loi de commande $u + \delta u$ soit admissible et transfère la phase initiale $(x(0), 0)$ en l'état à $t = T$, est exprimée par le cumul des variations de l'Hamiltonien selon la commande. L'Hamiltonien est défini par:

$$H(x_t, u_t, \lambda_t, t) = L(x_t, u_t, t) + \lambda_t f(x_t, u_t, t)$$

Et l'on doit avoir, pour tout t , les trois conditions:

- Minimisation de l'effet d'hamiltonien selon u , c'est à dire si u_t n'atteint pas les bornes de son domaine de variation \mathbb{U}_t , :

$$\frac{\partial H(x_t, u_t, \lambda_t, t)}{\partial u_t} = 0;$$

- λ , le vecteur adjoint, est défini par l'équation récurrente :

$$\lambda(t) - \lambda(t-1) = -\frac{\partial H(x_t, u_t, \lambda_t, t)}{\partial x_t}$$

et la condition **finale**:

$$\lambda(T) = 0$$

- x , le vecteur d'état, est défini par l'équation récurrente :

$$x(t) - x(t-1) = \frac{\partial H(x_t, u_t, \lambda_t, t)}{\partial \lambda_t}$$

et la condition **initiale**:

$$x(t=1) = a$$

3.3 Idée de la démonstration du principe du minimum (Pontryaguine)

On introduit les variables duales ou multiplicateurs de Lagrange $(\lambda_t)_{t=1:T}$ associées aux contraintes $x_{t+1} - x(t) - f(x_t, u_t, t) = 0$.

Le Lagrangien s'écrit

$$\mathcal{L} = \sum_{t=1}^T L(x_t, u_t, t) - \sum_{t=1}^T \lambda_t (x_{t+1} - x(t) - f(x_t, u_t, t))$$

avec $\lambda_T = 0$ car aucune contrainte ne s'applique en $t = T$ sur l'état x_{T+1} . Les conditions d'optimalité s'écrivent

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \lambda_t} = 0; t = 1, 2, 0, \dots, T-1 \\ \frac{\partial \mathcal{L}}{\partial u_t} = 0; t = 1, 2, 0, \dots, T \\ \frac{\partial \mathcal{L}}{\partial x_t} = 0; t = 2, 0, \dots, T-1 \end{cases}$$

Soit encore

$$\begin{aligned}
x_{t+1} &= x_{(t)} + f(x_t, u_t, t) \\
0 &= \frac{\partial L(x_t, u_t, t)}{\partial u_t} + \lambda_t \times \frac{\partial f(x_t, u_t, t)}{\partial u} \\
0 &= \frac{\partial L(x_t, u_t, t)}{\partial x} + \lambda_t \left(1 + \frac{\partial f(x_t, u_t, t)}{\partial x} \right) - \lambda_{t-1}
\end{aligned}$$

En introduisant: $H_t(x_t, u_t, \lambda_t, t) = L(x_t, u_t, t) + \lambda_t f(x_t, u_t, t)$ l'Hamiltonien, il vient

$$\begin{aligned}
x_{t+1} &= x_{(t)} + f(x_t, u_t, t) \\
0 &= \frac{\partial H(x_t, u_t, \lambda_t, t)}{\partial u_t} \\
\lambda_t - \lambda_{t-1} &= -\frac{\partial H(x_t, u_t, \lambda_t, t)}{\partial x_t} \\
x_1 &= a, x_T = b, \lambda_T = 0
\end{aligned} \tag{15}$$

Dans le cas plus général où u est sujet aux contraintes $u \in U_t$, l'équation 15 est remplacée par

$$u_t = \underset{u \in U_t}{\text{ArgMin}} (H(x_t, u_t, \lambda_t, t))$$

3.4 Exemple d'application: fonction de production du type Ricker

$$\begin{aligned}
x_{t+1} &= F(x_t - u_t) \\
\inf_{u \in \mathcal{U}} J(u(\cdot)) &= \sum_{t=1}^T L(x(t), u(t), t) \\
L(x(t), u(t), t) &= \alpha^{t-1} L(x(t), u(t))
\end{aligned}$$

avec $F(z) = r(1 - \frac{z}{K})z$, $F'(u) = r(1 - 2\frac{z}{K})$, $f(x_t, u_t, t) = F(x_t - u_t) - x_t$

$$\begin{aligned}
0 &= \frac{\partial L(x_t, u_t, t)}{\partial u_t} - \lambda_t F'(x_t - u_t) \\
\lambda_t - \lambda_{t-1} &= -\frac{\partial L(x_t, u_t, t)}{\partial x_t} - \lambda_t F'(x_t - u_t) \\
x_1 &= a, x_T = b, \lambda_T = 0
\end{aligned} \tag{16}$$

D'où

$$\frac{\partial L(x_t, u_t, t)}{\partial u_t} = \lambda_t F'(x_t - u_t)$$

En supposant $L(x, u, t) = \alpha^{t-1}L(x, u)$ et qu'à $T \rightarrow \infty$ il existe une commande stable (u^*, x^*) , l'équilibre est donné par

$$\lambda_t - \lambda_{t-1} = (\alpha^{t-1} - \alpha^{t-2}) \frac{\frac{\partial L(x, u)}{\partial u}}{F'(x - u)}$$

et

$$\begin{aligned} \lambda_t - \lambda_{t-1} &= -\alpha^{t-1} \frac{\partial L(x, u)}{\partial x} - \lambda_t (F'(x_t - u_t) - 1) \\ \lambda_t - \lambda_{t-1} &= -\alpha^{t-1} \frac{\partial L(x, u)}{\partial x} - \alpha^{t-1} \frac{\partial L(x, u)}{\partial u} \left(1 - \frac{1}{F'(x_t - u_t)}\right) \\ \lambda_t - \lambda_{t-1} &= -\alpha^{t-1} \left(\frac{\partial L(x, u)}{\partial x} + \frac{\partial L(x, u)}{\partial u} \left(1 - \frac{1}{F'(x_t - u_t)}\right) \right) \end{aligned}$$

d'où

$$(\alpha^{t-1} - \alpha^{t-2}) \frac{\frac{\partial L(x, u)}{\partial u}}{F'(x - u)} = -\alpha^{t-1} \left(\frac{\partial L(x, u)}{\partial x} + \frac{\partial L(x, u)}{\partial u} \left(1 - \frac{1}{F'(x_t - u_t)}\right) \right)$$

Soit

$$F'(x - u) \frac{\frac{\partial L(x, u)}{\partial x} + \frac{\partial L(x, u)}{\partial u}}{\frac{\partial L(x, u)}{\partial u}} = \alpha^{-1}$$

Cette équation doit être associée à la stabilité asymptotique du système

$$x = F(x - u)$$

À résoudre alors deux équations pour les deux inconnues u et x donnant les points de fonctionnement du système en situation d'exploitation durable, à particulariser selon la forme de $L(x, u)$.

4 Exercices d'application

4.1 Exercices de PD déterministe

d'après [4]

1- Fusée:

À des fins d'exploration des confins de la galaxie, une fusée transportant une masse utile M_0 doit être accélérée jusque 1/10 de la vitesse de la lumière à partir d'une station orbitale (gravité nulle). Il s'agit de construire une fusée à n étages, de masse initiale la plus faible possible.

Chaque étage comprend 90 pour cent de combustible et 10 pour cent de structure qui est abandonnée avant la mise à feu de l'étage suivant.

L'accélération dans le vide est régie par la loi newtonienne de l'action et la réaction: $M(t) \frac{dV(t)}{dt} = -\frac{dM(t)}{dt}$ où $M(t)$ désigne la masse de la fusée à l'instant

t et $V(t)$ sa vitesse. Des unités ont été choisies de telle sorte que la vitesse de la lumière vaille 1.

QUESTIONS:

Numéroter les étages de 1 à n de telle sorte que 1 soit le dernier étage à être mis à feu. En appelant M_n la masse de la fusée au moment de l'allumage de l'étage n et x_n l'augmentation de vitesse due à l'étage n ,

a) Montrer que $M_n = M_{n-1}g(x_n)$ par intégration de l'équation de Newton avec

$$g(x) = \frac{0.9 \exp x}{(-0.1 + \exp x)}$$

b) Formuler le problème sous forme de programmation dynamique. Bien préciser quelle est la définition adoptée pour l'état, la commande et le critère. En appelant $f(n, c)$ la plus petite masse de la fusée pour accélérer jusqu'à la vitesse c (quelconque) en n étages, écrire l'équation fonctionnelle de Bellman pour $f(n, c)$.

c) Montrer que la solution est obtenue pour $x_i = \frac{c}{n}$, pour tous les i allant de 1 à n .

2- BUS:

Une rue (à sens unique) d'une ville américaine est constituée de m blocs équidistants, un bus traverse la ville en empruntant cette rue d'un bout à l'autre. La municipalité a décidé d'aménager un nombre fixé n d'arrêts de bus. Elle souhaite les placer de telle sorte que la distance totale parcourue à pied par les usagers des transports en commun soit minimale.

On appelle B_j le nombre d'usagers issus du bloc j et C_i le nombre d'usagers se rendant au bloc i par ce bus. On fait l'hypothèse qu'un usager marche jusqu'à l'arrêt de bus le plus proche de son domicile, monte dans le bus et descend à l'arrêt le plus proche de son lieu de travail.

(Bien sûr $\sum_{i=1}^m C_i = \sum_{j=1}^m B_j$)

1) Ecrire une équation fonctionnelle dont la solution alloue l'emplacement des arrêts de bus de telle sorte que la distance totale effectuée à pied par les usagers de bus soit minimale.

2) On a besoin de 3 mn à pied pour marcher le long d'un bloc de maisons, le bus à pleine vitesse couvre 5 blocs par minute mais il lui faut 2 mn pour décélérer jusqu'à un arrêt, charger et décharger les passagers et réaccélérer jusqu'à pleine vitesse.

Poser le problème et écrire une équation fonctionnelle dont la solution donne le nombre d'arrêts de bus qui minimise la somme des temps de déplacement de la population.

3- SYSTÈME QUADRATIQUE LINÉAIRE:

Appliquer le principe de Bellman au cas d'un système quadratique en objectif et linéaire en évolution et montrer qu'en mettant la solution $V(x(t), t)$ sous la forme quadratique $x'(t)P(t)x(t)$, la matrice P vérifie une équation particulière.

4.2 Programmation dynamique stochastique

d'après [10]

1 - QUITTE OU DOUBLE

Monsieur Patrick Dupont doit participer à un jeu radiophonique appelé "quitte ou double".

Ce jeu se déroule de la façon suivante:

N questions de difficultés croissantes sont posées au candidat. Une bonne réponse à la première question vaut $g = 100F$.

Si le candidat répond correctement à la question k ($k = 1 \dots N - 1$), il peut décider:

- soit de continuer à jouer et tenter de doubler ses gains en répondant juste à la question suivante;

- soit d'abandonner le jeu et se retirer avec ce qu'il a gagné.

Si sa réponse à la question k est fautive, il perd tout.

On suppose qu'une écoute assidue de ce jeu, a permis à Monsieur Patrick Dupont d'estimer la probabilité p_k qu'il a de mal répondre à la question k .

1- Donner une règle permettant à Monsieur Patrick Dupont de savoir quand il doit volontairement se retirer du jeu.

(Quel modèle permet de formaliser ce jeu? Ne pas hésiter à tracer un graphe. Quel principe permet de donner la règle demandée?)

2- Application: $N = 4, p_1 = \frac{1}{15}, p_2 = \frac{2}{15}, p_3 = \frac{1}{5}, p_4 = \frac{2}{3}$

3- Le jeu est modifié. Le candidat a droit à une erreur avant d'être éliminé. Une bonne réponse à la question k lui assure un gain $g_k = 2^{k-1}g$.

Peut-on formaliser le jeu par le même modèle qu'en 1.?

Expliquer la réponse.

Quelle est alors la règle rationnelle d'abandon recommandée par la programmation dynamique?

References

- [1] V. Alexéev, E. Galéev, V. Tikhomirov, *Recueil de problèmes d'optimisation*, Editions MIR, 1987.
- [2] E. Beltrami, *Mathematics for dynamic modeling*, Academic Press, 1987.
- [3] P. Bernhard, *Commande optimale, décentralisation et jeux dynamique*, Dunod, 1976.
- [4] E. Denardo, *Dynamic programming: Models and applications*, Prentice Hall, 1982.
- [5] R. Faurre, *Notes d'optimisation*, cours de l'Ecole Polytechnique, 1979.
- [6] P. Faurre, M. Robin, *Eléments d'automatique*, Dunod, 1984.
- [7] B. Friedland, *Control system design*, Mac Graw-Hill, NY, 1986.

- [8] D. Jacobson and D. Mayne, *Differential dynamic programming*, American Elsevier Publishing Company, 1970.
- [9] C.S. Tapiero, *Applied stochastic models and control in management*, North Holland, 1988.
- [10] Roseaux, *Exercices et problèmes résolus de recherche opérationnelle*, tome 1, Masson, Paris, 1983.