Méthodes d'optimisation statique et dynamique

Pierre Riedinger

2013-2014

Table des matières

Première partie

Optimisation statique

Introduction

Il est rare qu'une activité humaine ne soit un jour ou l'autre confrontée à un problème de conception, de rationalisation, d'amélioration dont la recherche d'une solution ne conduise à un problème d'optimisation.

Que ce soit en sciences, en mathématiques, en ingénierie, en économie, en finance, dans le commerce, en gestion, les exemples foisonnent. Citons par exemples :

- le calcul d'une structure, d'une forme
- l'allocation de ressources
- la gestion de stock
- l'ordonnancement de taches
- la recherche opérationnelle
- les plans d'investissements
- l'analyse de données
- la modélisation et l'identification paramétrique
- la commande de processus
- etc.

Bien que ces problèmes soient de natures diverses, leur traduction mathématique consiste à :"déterminer un minimum d'une fonction f sur un domaine donné et qui satisfasse un ensemble de contraintes".

Il n'existe cependant pas une méthode générique efficace qui puisse résoudre tous les types de problèmes. En effet, l'efficacité des méthodes et leur applicabilité dépendent notamment :

- de la dérivabilité ou non de la fonction f
- de la possibilité d'un calcul effectif ou non de ces dérivées
- du domaine sur lequel est recherché la solution qui peut être continu ou discret
- du type de contraintes (égalités, inégalités)
- de la classe du problème : linéaire, quadratique, convexe, non linéaire
- de la dimension du domaine (nombres d'inconnues)

Pour chaque cas de figures, des méthodes existent. L'objectif de ce chapitre est d'aider l'utilisateur à choisir un code de calcul adéquat. Pour une meilleure compréhension de ces outils, nous nous efforcerons de présenter les idées à la base de leur fonctionnement.

Il existe deux grandes classes de méthodes :

- les méthodes locales [**?**], [**?**], [**?**], [**?**], [**?**] qui par analogie avec un marcheur en montagne cherchant à rejoindre le fond de la vallée, explorent le terrain localement et s'engagent vers laval. Ces méthodes supposent qu'il est possible d'évaluer la pente et par voie de conséquence que f est différentiable. Ces méthodes ne garantissent pas l'obtention d'un minimum global mais local.
- les méthodes globales cherchent donc un minimum global (sans réelle garantie). Citons la méthode des essaims particulaires qui envoie sur le terrain un grand nombre d'agents qui communiquent et se renseignent sur leur performance respective afin de s'engager dans la meilleure direction, un peu à la manière de chercheurs de champignons [**?**]. Les algorithmes génétiques [**?**] qui, sur un procédé de sélection-mutation, génèrent une "troupe d'élite". Il en existe bien d'autres : Les algorithmes évolutionnaires, la méthode de recherche Tabou, La méthode du recuit simulé, etc. Avec ces méthodes [**?**], la différentiabilité de f n'est pas nécessaire et on évite plus facilement les minima locaux. En revanche, leur réglage est souvent plus problématique car empirique.

Dans ce chapitre, nous n'étudierons que les méthodes locales plus abouties mais l'existence des méthodes globales est à garder à l'esprit. Plus récentes, elles font l'objet d'une activité de recherche importante.

Chapitre 1

Optimisation non contrainte

1.1 Rappels mathématiques

On considère une fonction f de $\mathbb{R}^n \to \mathbb{R}$
f est de classe C^1 voire C^2 (calcul du Hessien)

– On note

$$g(x) = \nabla f(x) = (\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), ..., \frac{\partial f}{\partial x_n}(x))$$

le **gradient** de f au point x et

$$H(x) = \nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & \ddots & & \vdots \\ \vdots & & & \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \cdots & & \frac{\partial^2 f}{\partial x_n^2}(x) \end{pmatrix}$$

le **Hessien** de f évalué au point x.

– **Développement de Taylor à l'ordre 2 (D.L.) :**

$$f(x+h) = f(x) + h^T g(x) + \frac{1}{2} h^T H(x) h + o(h^T h)$$

– **Ligne de niveau**
$L_z = \{x \in \mathbb{R}^n : \ f(x) = z\}$: ensemble des x tels que $f(x) = z$.

– Par définition, **la dérivée de f dans la direction** d est égale à :

$$f'(x;d) = \lim_{\substack{t \to 0 \\ t \in \mathbb{R}}} \frac{f(x+td) - f(x)}{t}$$

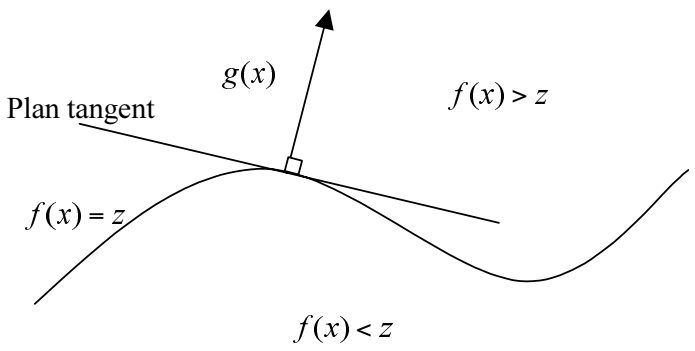


FIGURE 1.1 – Ligne de niveau et gradient

– Lorsque g existe (ce qui sera le cas), on a :

$$f'(x;d) = d^T g(x)$$

Proposition 1 *Si d appartient à la tangente à la courbe de niveau alors*

$$f'(x;d) = 0 \tag{1.1}$$
$$Soit\ d^T g(x) = 0 \tag{1.2}$$

$g(x)$ est donc orthogonal à la courbe de niveau (Figure 1.1).
Soit x_k une suite de points situés sur la ligne de niveau L_z et qui converge vers x.
Posons

$$\alpha_k = \|x_k - x\|$$
$$d_k = \frac{x_k - x}{\|x_k - x\|}; \|d_k\| = 1$$

Alors $x_k - x = \alpha_k d_k$ et $\alpha_k \to 0$ $d_k \to d$ lorsque $k \to +\infty$.
Montrons que $\lim_{k\to\infty} \frac{f(x_k)-f(x)}{\alpha_k} = f'(x;d)$:

$$\begin{aligned}\frac{f(x_k) - f(x)}{\alpha_k} &= \frac{f(x + \alpha_k d_k) - f(x)}{\alpha_k} \\ &= g(x)^T d_k + o(\alpha_k)/\alpha_k \quad \text{par un D.L. à l'ordre 1}\end{aligned}$$

Par passage à la limite, $\lim_{k\to\infty} \frac{f(x_k)-f(x)}{\alpha_k} = f'(x;d)$
Puisque la suite des points x_k appartient à L_z, $f(x_k) - f(x) = 0, \forall k$ d'où le résultat :

$$\lim_{k\to\infty} \frac{f(x_k) - f(x)}{\alpha_k} = f'(x;d) = 0$$

Notation 2 *Par la suite, il arrivera que l'on note l'évaluation d'une fonction f au point x_k par f_k.*

1.2 Formulation du problème

Problème d'optimisation non contraint :

Problème 3 *Trouver le minimum de la fonction f*

$$\min_x f(x), x \in \mathbb{R}^n$$

N.B. : Trouver un maximum de f

$$\max_x f(x) = -\min_x(-f(x)), x \in \mathbb{R}^n$$

1.3 Conditions de minimum local et existence

Théorème 4 *Une condition nécessaire pour que f admette un minimum au point x^*, est que*

$$g(x^*) = 0$$

Proof. D'après le D. L. de f,

$$f(x^* + h) = f(x^*) + h^T g(x^*) + o(\|h\|)$$

Si x^* est un minimum de f, $\forall y \in \mathbb{R}^n$, $\forall t \in \mathbb{R}$ suffisamment petit, on a :

$$f(x^* + ty) - f(x^*) \geq 0$$

Soit, pour $t > 0$,

$$\lim_{t \to 0} \frac{f(x^* + ty) - f(x^*)}{t} \geq 0$$

et pour $t < 0$,

$$\lim_{t \to 0} \frac{f(x^* + ty) - f(x^*)}{t} \leq 0$$

or

$$\lim_{t \to 0} \frac{f(x^* + ty) - f(x^*)}{t} = y^T g(x^*) \underset{\leq 0}{\overset{\geq 0}{,}}$$

$$\text{d'où } y^T g(x^*) = 0, \forall y$$

$$\text{et finalement } g(x^*) = 0$$

■

Remarque 5 ***Attention : le théorème est faux si le domaine de définition de f n'est pas un ouvert (Figure 1.2).***

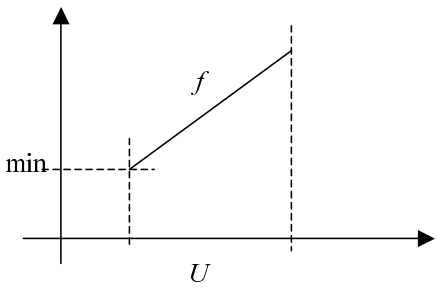


FIGURE 1.2 – Cas d'une fonction affine sur un domaine compact

Remarque 6 ***La réciproque est fausse :*** $g(x) = 0 \not\Rightarrow$ *f atteint un extremum en x*

Théorème 7 *Condition suffisante*

Si la condition nécessaire est satisfaite et si le Hessien $H > 0$ (est défini positif) alors x^* est un minimum

En effet, il suffit de développer à l'ordre 2 la fonction f :

$$f(x^* + ty) = f(x^*) + ty^T g(x^*) + \frac{t^2}{2} y^T H(x^*) y + o(t^2)$$

La condition nécessaire de minimum implique

$$f(x^* + ty) - f(x^*) = \frac{t^2}{2} y^T H(x^*) y + o(t^2)$$

Pour t suffisamment petit, comme $y^T H(x^*) y > 0$,

$$f(x^* + ty) - f(x^*) = \frac{t^2}{2} y^T H(x^*) y + o(t^2) > 0$$

i.e. $f(x^* + ty) > f(x^*)$, $\forall y \in \mathbb{R}^n$, $\forall t \in \mathbb{R}$ suffisamment petit. Donc x^* est un minimum.

Théorème 8 *(Théorème d'existence)* $f : U \subset \mathbb{R}^n$, *fermée* $\to \mathbb{R}$, f *continue*
on suppose que :
ou bien U *est borné*
ou bien $\lim\limits_{\substack{\|x\|\to+\infty \\ x\in U}} f(x) = +\infty$
alors f *admet un minimum sur* U

1.4 Principe des méthodes de descente

Idée : Résoudre une suite de problèmes d'optimisation mono-dimensionnelle

Algorithme général :
Pour $k = 1, 2, ...$
Déterminer une direction de descente d_k depuis la position courante x_k

1. Trouver α_k qui "minimise" $f(x_k + \alpha d_k)$ par rapport à α
2. Poser $x_{k+1} = x_k + \alpha_k d_k$,
3. Retourner en 1 tant que le test d'arrêt n'est pas validé.

Fin Pour

Algorithme 9 ***Test d'arrêt :*** *On choisit un* $\varepsilon > 0$ *et l'algorithme s'arrête lorsque*

$$\|x_{k+1} - x_k\| < \varepsilon$$

ou

$$\|f(x_{k+1}) - f(x_k)\| < \varepsilon$$

ou encore les erreurs relatives

$$\frac{\|x_{k+1} - x_k\|}{\|x_k\|} < \varepsilon$$

ou

$$\frac{\|f(x_{k+1}) - f(x_k)\|}{\|f(x_k)\|} < \varepsilon$$

Remarque 10 ***Attention****, le test ne garantit pas* x_k *proche de* x^* *mais seulement une progression faible de la suite.*

Remarque 11 *La méthode est dite* ***une méthode de descente*** *si la propriété :*

$$\left.\frac{df}{d\alpha}\right|_{\alpha=0} = d_k^T g_k < 0$$
$$= f'(x_k; d_k) < 0$$

est vérifiée à chaque itération où $.g_k = g(x_k)$.
En effet, on a pour $\alpha_k > 0$ *et suffisamment petit,*

$$f(x_{k+1}) < f(x_k)$$

Remarque 12 *Si le point 2 est réalisé à chaque itération, alors la condition nécessaire de minimum donne* $d_k^T g_{k+1} = 0$ *(Figure 1.3).*

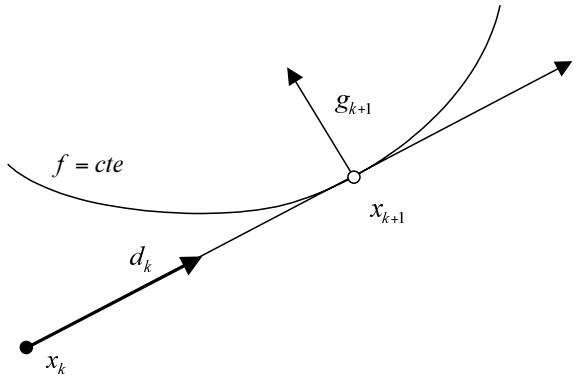
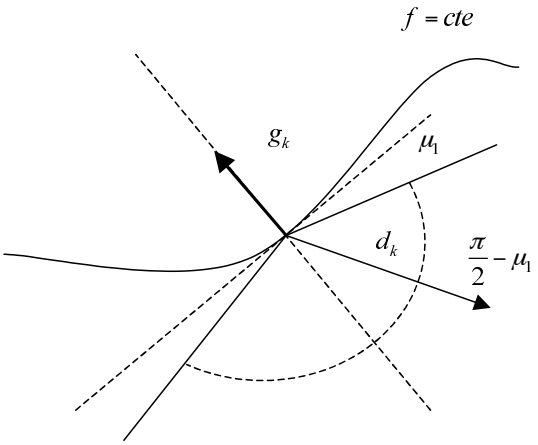


FIGURE 1.3 – $d_k^T g_{k+1} = 0$ au minimum dans la direction d_k

1.5 Critère de convergence

Théorème 13 *Si la méthode de descente respecte à chaque itération, les points (Figure 1.4) :*

i. Direction admissible (direction de descente) :
l'angle $(\widehat{d_k, -g_k}) \leq \frac{\pi}{2} - \mu_1$, $\forall k$ *avec* $\mu_1 > 0$



Direction de descente admissible

ii. Décroissance suffisante de f (Armijo) :
pour $0 < \mu_2 < 1$,

$$f(x_{k+1}) - f(x_k) \leq \mu_2 \alpha_k d_k^T g_k$$

Si on note : $f(\alpha_k) = f(x_k + \alpha_k d_k)$ *cette condition est équivalente à* $f(\alpha_k) - f(0) \leq \mu_2 \alpha_k f'(0)$. *Il faut donc être sous la droite* $f(0) + \mu_2 \alpha_k f'(0)$

iii. Réduction suffisante de la dérivée f′ (Wolfe) : Pour $\mu_2 < \mu_3 < 1$,

$$\left|f'(\alpha_k)\right| \leq \mu_3 \left|f'(0)\right|$$

et si g existe et est uniformément continue sur l'ensemble $\{x \in \mathbb{R}^n : f(x) < f(x_0)\}$
alors ou bien $f_k = f(x_k) \to -\infty$, *ou bien* $g_k = g(x_k) \to 0$.

Commentaires :
- Le point (i.) évite de remonter (mauvaise direction)
- les points (i.) et (ii.) impliquent $d_k^T g_k < 0$ et donc $f(x_{k+1}) < f(x_k)$
- le point (iii.) réduit l'ensemble des points acceptables aux points proches de $\min_\alpha f(\alpha)$

Dans la pratique, on peut choisir par défaut $\mu_2 = 0.001$ et $\mu_3 = 0.9$

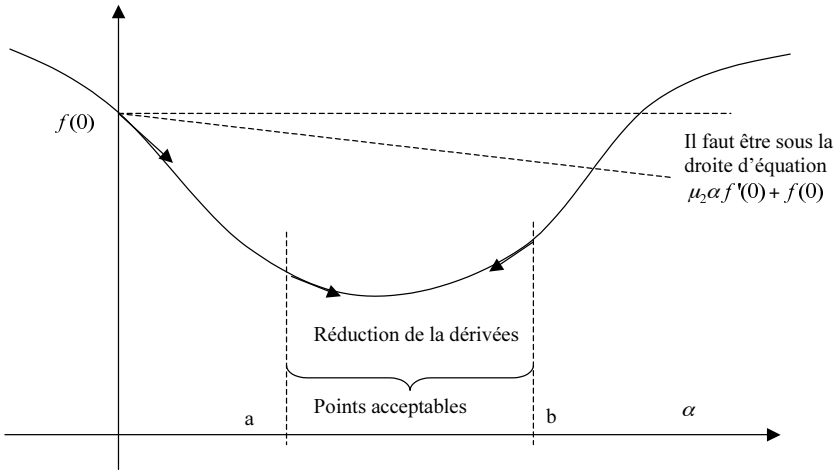


FIGURE 1.4 – Critère de convergence

Maintenant tout l'art est dans la détermination d'algorithmes efficaces pour choisir la direction de descente d_k et le pas de descente α_k.

Supposons que la direction d_k est déjà choisie.

1.6 Procédure de descente dans une direction donnée d_k

De nombreux algorithmes existent et les performances obtenues en dépendent fortement. On obtiendra de meilleurs résultats si on dispose du calcul effectif de gradient g voire de Hessien H. Un algorithme de descente est donc une procédure qui génère une séquence de pas α_i et qui termine lorsque le point obtenu

$$x_i = x_k + \alpha_i d_k$$

est un point acceptable (voir critère de convergence ci dessus).

Généralement, il comporte deux phases :

Phase 1 : Elle consiste à déterminer un intervalle $[a_i, b_i]$ dont on sait qu'il contient un ensemble de points acceptables

Phase 2 : Elle consiste à découper, réduire cet intervalle initial en générant une suite d'intervalles plus petits et dont la longueur tend vers 0. Le pas α_i résultant est le pas recherché.

Voici un algorithme proposé par Fletcher ([**?**]) et permettant d'obtenir ce résultat.

Algorithme :

$\alpha_0 = 0$, $\alpha_1 = 1$
Phase 1 :

Pour $i = 1, 2, ...$ faire
- évaluer $f(\alpha_i)$
- si $f(\alpha_i) > f(0) + \mu_2 \alpha_i f'(0)$ ou $f(\alpha_i) \geq f(\alpha_{i-1})$
  alors $a_i = \alpha_{i-1}$ $b_i = \alpha_i$ fin de phase 1
- évaluer $f'(\alpha_i)$
- si $|f'(\alpha_i)| \leq -\mu_3 f'(0)$ alors fin de phase 1 et 2
- si $f'(\alpha_i) \geq 0$
  alors $a_i = \alpha_i$ $b_i = \alpha_{i-1}$ fin de phase 1
- Choisir $\alpha_{i+1} \in [\alpha_i + (\alpha_i - \alpha_{i-1}), \alpha_i + \tau_1(\alpha_i - \alpha_{i-1})]$

Fin Pour

Phase 2 : Réduction de l'intervalle

Pour $j = i, i+1, ...$ faire
- choisir $\alpha_j \in [a_j + \tau_2(b_j - a_j), b_j - \tau_3(b_j - a_j)]$
- évaluer $f(\alpha_j)$
- si $f(\alpha_j) > f(0) + \mu_2\alpha_j f'(0)$ ou $f(\alpha_j) \geq f(a_j)$
  alors $a_{j+1} = a_j$ $b_{j+1} = \alpha_j$
- sinon évaluer $f'(\alpha_j)$
  - si $|f'(\alpha_j)| \leq -\mu_3 f'(0)$ alors fin de phase 2
  - sinon $a_{j+1} = \alpha_j$
  - si $(b_j - a_j)f'(\alpha_j) \geq 0$ alors $b_{j+1} = a_j$ sinon $b_{j+1} = b_j$

Fin Pour

Dans la pratique, on peut choisir $\tau_1 = 9$ $\tau_2 = 0.1$ $\tau_3 = 0.5$.

L'évaluation de $f'(\alpha)$ est soit obtenue à partir de g soit numériquement à l'aide du taux d'accroissement dans chaque direction de l'espace ce qui nécessite au moins n évaluations de f.

Proposition 14 *La phase 1 fournie un intervalle qui contient un ensemble de points acceptables.*

Proposition 15 *La phase 2 termine avec un point acceptable.*

Remarque importante : Dans la phase 1 et 2, il est recommandé de choisir α à l'aide d'une interpolation polynomiale de f.

Lorsque l'évaluation du gradient est possible (calcul de g effectif i.e. non numérique), on préfère utiliser une interpolation cubique sinon on se contente d'une interpolation quadratique.

Rappel sur l'interpolation :

Sur $[a\ b]$, si on dispose de $f_a = f(a)$, $f_b = f(b)$, $f'_a = f'(a)$ et $f'_b = f'(b)$, il existe un unique polynôme P de degré trois tel que :

$$\begin{array}{ll} P(a) = f_a & P'(a) = f'_a \\ P(b) = f_b & P'(b) = f'_b \end{array}$$

Le minimum de $P(\alpha)$ est alors donné par

$$\alpha = b - (b-a)\frac{f'_b + \beta_2 - \beta_1}{f'_b - f'_a + 2\beta_2}$$

avec

$$\beta_1 = f'_a + f'_b - 3\frac{f_a - f_b}{a - b}$$
$$\text{et } \beta_2 = \sqrt{\beta_1^2 - f'_a f'_b}$$

Pour une interpolation quadratique sans calcul de g, on cherche un polynôme d'ordre 2. Il faut trois points (x_1, x_2, x_3), on peux choisir $x_1 = a$, $x_2 = (a+b)/2$ et $x_3 = b$, l'extremum est donné par :

$$\alpha = \frac{1}{2}\frac{\beta_{23}f_1 + \beta_{31}f_2 + \beta_{12}f_3}{\gamma_{23}f_1 + \gamma_{31}f_2 + \gamma_{12}f_3}$$

avec

$$\beta_{ij} = x_i^2 - x_j^2$$
$$\gamma_{ij} = x_i - x_j$$

Attention un test est à effectuer pour s'assurer que l'extremum est bien un minimum et qu'il appartient bien à l'intervalle $[a\ b]$. Sinon, le minimum est atteint à l'une des extrémités de l'intervalle.

Nous disposons à présent **d'une méthode efficace pour déterminer le pas α_k à chaque itération de l'algorithme général.** Maintenant il nous faut choisir une direction de descente d_k.

1.7 La méthode du gradient

Idée : La direction de descente est celle de plus grande pente ; soit

$$d_k = -g_k.$$

La méthode converge et est donc robuste mais avec en général de piètres performances (phénomène de zig zag).

1.8 La méthode de Newton

Idée : On remplace f par son approximation quadratique au point x_k
Soit

$$q_k(d) = f_k + d^T g_k + \frac{1}{2} d^T H_k d$$

et on minime q_k plutôt que f.

La C.N. donne $H_k d_k = -g_k$
Soit

$$d_k = -H_k^{-1} g_k.$$

q_k a un unique minimum si $H_k > 0$
L'itération est alors obtenue directement par

$$x_{k+1} = x_k + d_k \ (\alpha_k = 1)$$

Théorème 16 *Si f est C^2 et si le Hessien H est défini positif et lipschitz dans un voisinage de la solution x^* alors la méthode converge dans un voisinage de la solution.*
La convergence est quadratique près de la solution i.e.

$$\exists \beta > 0, \lim \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} < \beta$$

Ceci signifie que la méthode améliore à chaque itération de deux chiffres significatifs l'estimation de la solution (prés de la solution).

Inconvénients : il faut évaluer et inverser à chaque itération le Hessien. La méthode peut ne pas converger si H_k n'est pas défini positif.

En résumé : la méthode est coûteuse, peu robuste mais rapide si elle converge.

1.9 Les méthodes Quasi-Newtoniennes

Calculer H numériquement et l'inverser est très coûteux d'où
Idée : Utiliser les valeurs calculées de f_k et g_k pour évaluer H_k^{-1}.
On pose :

$$s_k = x_{k+1} - x_k$$
$$p_k = g_{k+1} - g_k$$

Par un développement limité, on a

$$p_k = H_k s_k + o(s_k)$$

Pour une fonction quadratique, on a même l'égalité : $p_k = H_k s_k$
On pose alors la condition (**Quasi Newton**) suivante, on cherche une matrice noté B_{k+1} telle que

$$s_k = B_{k+1} p_k$$

Autrement dit, l'image par B_{k+1}de l'accroissement de gradient doit correspondre à l'accroissement de la position (comme pour H_k^{-1}).
Il y a plusieurs façons d'obtenir cette condition, la solution n'étant pas unique. Deux méthodes standards s'imposent pour calculer B_{k+1} :

1.9.1 La méthode BFGS (Broyden, Fletcher, Goldfard et Shanno)

$$B_{k+1} = B_k + (1 + \frac{p_k^T B_k p_k}{s_k^T p_k})\frac{s_k s_k^T}{s_k^T p_k} - \frac{s_k p_k^T B_k + B_k p_k s_k^T}{s_k^T p_k}$$
$$B_0 = Id$$

1.9.2 La méthode DFP (Davidson, Fletcher et Powell)

$$B_{k+1} = B_k + \frac{s_k s_k^T}{s_k^T p_k} - \frac{B_k p_k p_k^T B_k}{p_k^T B_k p_k}$$
$$B_0 = Id$$

Inversion : Forme dual complémentaire
Pour inverser les matrices B_{k+1}, on a le résultat :
Pour obtenir $H_{k+1\_DFP} = B_{k+1}^{-1}$: on remplace dans la formule BFGS B_k par H_k et on échange s_k et p_k
Pour obtenir $H_{k+1\_BFGS} = B_{k+1}^{-1}$: on remplace dans la formule DFP B_k par H_k et on échange s_k et p_k

Algorithme :
Une fois le calcul de B_{k+1} effectué :

1. On détermine la nouvelle direction $d_{k+1} = -B_{k+1} g_{k+1}$ (comme pour la méthode de Newton)
2. On utilise la procédure de descente pour obtenir $\alpha_{k+1} = \arg\min f(x_{k+1} + \alpha d_{k+1})$
3. On pose $x_{k+2} = x_{k+1} + \alpha_{k+1} d_{k+1}$ et on itére.

Remarque 17 *Si on ne dispose pas de g_k, on évalue le taux d'accroissement de f dans chaques directions e_i i.e.*

$$\frac{\partial f}{\partial x_i} \approx \frac{\partial f(x + he_i) - f(x)}{h}$$

Proposition 18 *Si la matrice B_k est maintenue définie positive alors d_k est toujours une direction de descente.*

Preuve : Comme $d_k = -B_k g_k$, la condition de descente est vérifiée : $g_k^T d_k = -g_k^T B_k g_k < 0$

Proposition 19 *$B_{k+1} > 0$ si $B_k > 0$ et si $p_k^T s_k > 0$*

Commentaire : La condition $p_k^T s_k > 0$ n'est pas très exigeante car :

$$p_k^T s_k = \alpha_k (d_k^T g_{k+1} - d_k^T g_k)$$

Or comme d_k est une direction de descente, $-d_k^T g_k > 0$ et $\alpha_k > 0$, seul le terme $d_k^T g_{k+1}$ peut être négatif. Mais si on utilise une procédure de descente qui respecte **la condition iii. (Wolfe) du critère de convergence** alors comme

$$\left| d_k^T g_{k+1} \right| < \mu_3 \left| d_k^T g_k \right|, \mu_3 < 1$$

on a la propriété. On voit donc que contrairement à l'algorithme de Newton, on a la garantie de converger vers un point stationnaire. Cette garantie se paye en contrepartie par une vitesse de convergence plus faible.

Proposition 20 *La vitesse de convergence de la méthode est superlinéaire i.e.*

$$\exists \beta > 0, \lim \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \beta < 1 \text{ (Convergence linéaire)}$$
$$\text{Si } \beta = 0, \text{la convergence est dite super linéaire}$$

En pratique, les deux méthodes donnent de très bons résultats et sont plus efficaces que la méthode du gradient à pas optimal.

1.10 Les méthodes de type moindres carrés.

Dans un problème de minimisation au sens des moindres carrés la fonction f est une somme de carrés.

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \|F(x)\|^2 = \frac{1}{2} \sum_i F_i^2(x)$$

Ce type de problème est dans la pratique très fréquent et se rencontre lorsque l'on cherche à faire correspondre un modèle avec des données (estimation paramétrique). x représente un ensemble de paramètres dont il faut déterminer le jeu qui permet d'expliquer au mieux les données obtenues par la mesure.

Les problèmes de suivi de trajectoire,

$$\min_{x\in\mathbb{R}^n} \int_{t_1}^{t_2} (y(x,t) - y_{mes}(t))^2 dt$$

en sont une autre illustration. Ici y_{mes} est donné et on cherche à faire coller y à y_{mes} au moyen de x. Si on utilise une formule de quadrature pour estimer l'intégrale, on obtient un problème de minimisation au sens des moindres carrés. Pour ce type de problème, on peut toujours utiliser les méthodes vues précédemment mais on obtiendra de meilleurs résultats en exploitant la structure particulière du problème.

En effet, si on note J la matrice Jacobienne de F, H_i la matrice Hessienne de chaque composante F_i alors

$$\begin{aligned} g(x) &= J(x)F(x) \\ H(x) &= J^T(x)J(x) + Q(x) \\ \text{avec } Q(x) &= \textstyle\sum_i F_i(x)H_i(x) \end{aligned}$$

Idée : La matrice $Q(x) \to 0$ lorsque $\|F(x)\| \to 0$ ($F_i(x) \to 0$ pour x proche de la solution)

1.10.1 La méthode de Gauss Newton

Cette méthode détermine une direction d_k de descente en négligeant le terme $Q(x)$ par la méthode de Newton.

Autrement dit, on fait l'approximation $H \approx J^T J$ et on utilise Newton ce qui donne :

1. On résout
$$J_k^T J_k d_k = -J_k F_k$$
2. On utilise une procédure de descente pour obtenir $\alpha_k = \arg\min f(x_k + \alpha d_k)$

Il arrive que la méthode ne fonctionne pas si le terme $Q(x)$ n'est pas négligeable. On utilisera alors la méthode :

1.10.2 La méthode de Levenberg Marquard

On modifie la méthode de G.N. en ajoutant

$$(J_k^T J_k + \lambda_k Id)d_k = -J_k F_k$$

où le scalaire λ_k permet de contrôler la longueur et la direction de d_k :
- pour $\lambda_k = 0$, on retrouve G.N.
- pour $\lambda_k \to +\infty$, $\|d_k\| \to 0$ et $d_k \approx -\frac{1}{\lambda_k} J_k F_k$, on a donc l'algorithme du gradient et $\|F_{k+1}\| < \|F_k\|$ pour λ_k suffisamment grand. On gagne en robustesse au détriment de la vitesse.

Implémentation : On utilise G.N. avec une procédure polynomiale de descente. L'inversion de $J_k^T J_k$ est obtenue par une factorisation QR.

Pour L.M., il faut pouvoir contrôler efficacement λ_k. Plusieurs solutions et codes de calculs sont proposés à la fois dans la littérature ([**?**]) et sur internet.

1.11 Les méthodes de gradients conjugués

Idée : Utiliser une direction de descente qui combine la direction du gradient $-g$ et une direction non encore explorée (orthogonale).

Pour une fonction quadratique, l'algorithme converge en au plus n coups.

Algorithmes :

$$d_1 = -g_1$$

Pour $k \geq 1$,

$$d_{k+1} = -g_{k+1} + \beta_k d_k$$

(procédé d'orthogonalisation de Schmidt)

avec (**Algorithme de Fletcher-Reeves**) :

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$$

Autre façon (**Algorithme de Polak-Ribière**) :

$$\beta_k = \frac{(g_{k+1} - g_k)^T g_{k+1}}{g_k^T g_k}$$

Les méthodes de gradients conjugués sont souvent employées pour des problèmes de grandes tailles car peu coûteuses.

Chapitre 2

Optimisation contrainte

2.1 Formulation du problème

Trouver le minimum de la fonction f

$$\min_x f(x), x \in \mathbb{R}^n$$

sous les contraintes

$$c_i(x) = 0, i \in E = \{1, ..., m_e\}$$
$$c_i(x) \geq 0, i \in I = \{m_e + 1, ..., m\}$$

où $f : \mathbb{R}^n \to \mathbb{R}$
$c_i : \mathbb{R}^n \to \mathbb{R}$
On supposera que les fonctions c_i et f sont de classes C^2.

Définition 21 ***Points admissibles :*** *On notera D l'ensemble des points x qui satisfont les contraintes c.*

Proposition 22 *Si D est non vide et borné alors il existe un minimum.*

Définition 23 ***Définition d'un déplacement admissible depuis une position*** *x. δ est un déplacement admissible pour un point $x \in D$, si la position résultante*

$$x + \delta \in D$$

Définition 24 ***Définition d'une direction admissible :*** *Soient $x \in D$ un point admissible et x_k une suite de points admissibles qui convergent vers x.*

On considère la suite des déplacements admissibles $\delta_k = x_k - x$, et on note

$$\delta_k = \alpha_k s_k \tag{2.1}$$
$$avec \ \|s_k\| = 1 \tag{2.2}$$

($s_k = \frac{\delta_k}{\|\delta_k\|}$ et $\alpha_k = \|\delta_k\|$). Si $x_k \to x$ alors $\alpha_k \to 0$ et $s_k \to s$. La limite s est appelée une direction admissible au point x.

Définition 25 *On notera $\mathcal{F}(x) = \{$ ensemble des directions admissibles au point $x\}$.*

2.2 Conditions de minimum local

2.2.1 Multiplieurs de Lagrange et conditions d'ordre 1

Cas des contraintes égalités

Lorsque $I = \emptyset$, chaque contrainte égalité définie une hypersurface $S_i = c_i^{-1}(0)$ et le domaine D est donc déterminé par l'intersection de ces hypersurfaces

$$D = \cap_{i \in E} S_i.$$

Notons $a_i(x) = \nabla c_i(x)$ et par $\mathbb{E}$, l'espace engendré par les $a_i(x)$, $i \in E$.

Notons enfin $\Pi(x) = \mathbb{E}^{\perp} = \{z : z \perp a_i(x),\ i \in E\}$. $\Pi(x)$ correspond à l'intersection des hyperplans tangents à chaque surface au point x.

Montrons que les directions admissibles au point $x \in D$ appartiennent à $\Pi(x)$.

Soit $x \in D$. Pour un déplacement admissible δ, i.e. $(x + \delta \in D)$

$$c_i(x + \delta) = c_i(x) + \delta^T a_i + o(\|\delta\|) \tag{2.3}$$

avec $a_i(x) = \nabla c_i(x)$.

Comme le déplacement est admissible $x + \delta \in D$ et $c_i(x + \delta) = 0$.

Par passage à la limite, on obtient

$$s^T a_i(x) = 0, \forall i \in E, \forall s \in \mathcal{F}(x)$$

Autrement dit, les directions admissibles s sont orthogonales à $a_i(x)$, $i \in E$ et donc appartiennent à $\Pi(x)$. Réciproquement on peut montrer que tout vecteur normé $v \in \Pi(x)$ est une direction admissible et conclure que $\mathcal{F}(x) = \Pi(x)$. En effet, par construction, si $y_k = x + \alpha_k v \in \Pi(x)$ avec $\alpha_k \to 0$, on peut considérer la suite $x_k = P(y_k)$ de D obtenue par projection orthogonale sur S (qui existe toujours dans un voisinage de x) et conclure que $v \in \mathcal{F}(x)$.

A présent, supposons que x^* est un minimum et considérons une direction admissible $s \in \Pi(x^*)$. A partir d'une séquence de point de D convergent vers x^* et par un développement limité, il est facile de montrer que la pente de f dans la direction de s ne peut être négative et on doit donc avoir

$$s^T g^* \geq 0$$

Puisque $c_i \in C^1$, $-s \in \mathcal{F}(x^*)$ et il vient $s^T g^* = 0$ (puisque $\pm s^T g^* \geq 0$) pour tout $s \in \Pi(x^*)$.

Donc $g^* \in \Pi(x^*)^{\perp} = (\mathbb{E}^{\perp})^{\perp} = \mathbb{E}$

Et g s'écrit nécessaire comme une combinaison linéaire des $a_i(x^*)$:(Figure 2.1)

$$g^* = \sum_{i \in E} \lambda_i^* a_i^* = A^* \lambda^* \tag{2.4}$$

On définit alors **la fonction Lagrangienne**

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in E} \lambda_i c_i(x)$$

A l'optimum, il vient : $\nabla \mathcal{L}(x^*, \lambda^*) = 0$ i.e.

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = g^* - \sum_{i \in E} \lambda_i^* a_i^* = 0$$
$$\nabla_\lambda \mathcal{L}(x^*, \lambda^*) = c_i^* = 0,\ i \in E$$

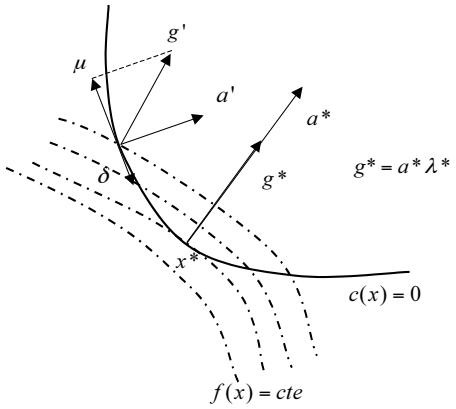


FIGURE 2.1 – $g^* = A^*\lambda^*$

Cas des contraintes inégalités

A la contrainte inégalité $i \in I$, on associe une variable y_i et on remplace la contrainte inégalité $c_i(x) \geq 0$ par une contrainte égalité en écrivant :

$$c_i(x) - y_i^2 = 0$$

On note : $\mathcal{A}^* = E \cup I^*$ où I^* est le sous ensemble de I des contraintes inégalités actives à l'optimum i.e $c_i^* = 0, \forall\ i \in \mathcal{A}^*$.

La fonction Lagrangienne est alors :

$$\mathcal{L}(x, y, \lambda) = f(x) - \sum_{i \in E} \lambda_i g_i(x) - \sum_{i \in I} \lambda_i (c_i(x) - y_i^2)$$

Les conditions du 1er ordre :

$$\nabla_x \mathcal{L} = 0,\ \nabla_y \mathcal{L} = 0$$

conduisent à :

$$g(x^*) + \sum_{i \in E \cup I} \lambda_i a_i(x) = 0$$
$$\lambda_i y_i = 0,\ i \in I$$

d'où l'on tire la condition : $\lambda_i = 0$ ou $y_i = 0$, $i \in I$

- Si $i \notin I^*$, $y_i \neq 0$ et $\lambda_i = 0$, la contrainte n'intervient pas dans la recherche de l'extremum et on a $c_i(x^*) > 0$
- Si $i \in I^*$, $y_i = 0$

En conclusion, on a montré que le gradient de f s'exprime comme une combinaison linéaire des gradients des contraintes actives

$$g^* = \sum_{i \in E \cup I^*} \lambda_i^* a_i^*$$

Par ailleurs, puisque $c_i^* = 0$ et $c_i(x^* + \delta) \geq 0$ pour $i \in I^*$ et δ admissible, les directions admissibles s doivent vérifier

$$s^T a_i^* = 0, \forall i \in E$$
$$s^T a_i^* \geq 0, \forall i \in I^*$$
$$\text{et } s^T g^* \geq 0$$

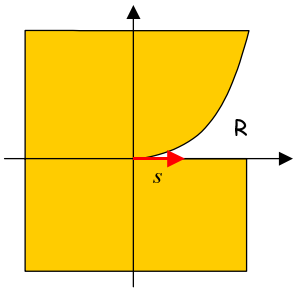


FIGURE 2.2 – Contraintes inégalités : Cas où $-s \notin \mathcal{F}(x)$ et où $-s \in F(x)$

Montrons également que les $\lambda_i^* \geq 0$, $i \in I^*$.
Sinon (cas d'une seule contrainte inégalité) $\exists \lambda < 0$ et on peut toujours choisir

$$s^T a^* = 1$$

d'où

$$s^T g^* = s^T a^* \lambda = \lambda < 0$$

d'où la contradiction.

Définition 26 *On note*

$$F(x^*) = \{s : s \neq 0 \text{ et } s^T a_i^* = 0,\ \forall i \in E,\ s^T a_i^* \geq 0,\ \forall i \in I^*\}$$

On a de manière évidente $F^* \supset \mathcal{F}^*$, réciproquement on a le

Lemma 27 $F^* = \mathcal{F}^*$
si les contraintes sont linéaires ou
si les a_i^* sont linéairement indépendantes au point x^*

La figure (2.2) est un contre exemple avec le cas où $-s \notin \mathcal{F}^*$.
En résumé, on a le théorème :
On définit alors **la fonction Lagrangienne**

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in E \cup I} \lambda_i c_i(x)$$

Théorème 28 *Kuhn-Tucker (KT) Conditions : Sous l'hypothèse de régularité $F^* = \mathcal{F}^*$ et si x^* est un minimiseur local du problème avec contraintes, alors il existe un multiplieur de Lagrange λ^* tel que*

$$\begin{aligned} \nabla_x \mathcal{L}(x^*, \lambda^*) = g^* - \sum_{i \in E \cup I} \lambda_i^* a_i^* = 0 \\ c_i^* = 0, i \in E \\ c_i^* \geq 0, i \in I \\ \lambda_i^* \geq 0, i \in I \\ \lambda_i^* c_i^* = 0, \forall i \end{aligned} \tag{2.5}$$

Remarque : $\lambda_i^* = 0$ si $c_i^* > 0$ (Figure 2.3).

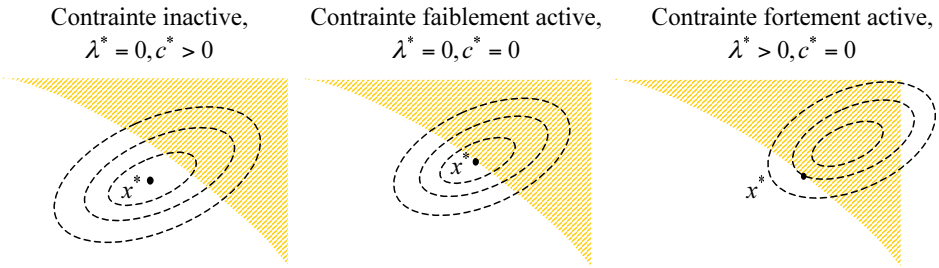


FIGURE 2.3 – Optimum et contrainte inégalité

Remarque concernant la signification de $\lambda_i^* \geq 0$.

Considérons un problème de minimisation d'une fonction f sous une contrainte inégalité : $c(x) \geq 0$.

Remplaçons cette contrainte par $c(x, \varepsilon) = c(x) - \varepsilon \geq 0$ pour un $\varepsilon > 0$. On note par $f^*(\varepsilon)$ la valeur de f à l'optimum en fonction de ε sous la contrainte $c(x, \varepsilon) = 0$.

Il est clair que pour tout $\varepsilon > 0$, $f^*(\varepsilon) \geq f^*(0)$ (restriction de D) et donc que $\left.\frac{df^*(\varepsilon)}{d\varepsilon}\right|_{\varepsilon=0} > 0$

On peut écrire que $f^*(\varepsilon) = \mathcal{L}(x^*(\varepsilon), \lambda^*(\varepsilon), \varepsilon)$ et par dérivation il vient :

$$\frac{df^*(\varepsilon)}{d\varepsilon} = \nabla_x \mathcal{L}(x^*(\varepsilon), \lambda^*(\varepsilon), \varepsilon)\frac{\partial x}{d\varepsilon} + \nabla_\lambda \mathcal{L}(x^*(\varepsilon), \lambda^*(\varepsilon), \varepsilon)\frac{\partial \lambda}{d\varepsilon} + \frac{\partial \mathcal{L}}{\partial \varepsilon}$$

$$\left.\frac{df^*(\varepsilon)}{d\varepsilon}\right|_{\varepsilon=0} = \left.\frac{\partial \mathcal{L}}{\partial \varepsilon}\right|_{\varepsilon=0} = \lambda^* > 0$$

Autrement dit le multiplieur de Lagrange correspond à la dérivé (positive !) $\left.\frac{df^*(\varepsilon)}{d\varepsilon}\right|_{\varepsilon=0}$ de $f^*(\varepsilon)$. Il indique qu'un déplacement vers l'intérieur du domaine admissible D augmente le critère f^*. C'est donc bien une condition nécessaire de minimum.

2.2.2 Conditions d'ordre 2

Cas des contraintes égalités : Si x^* est un minimum local alors pour un déplacement admissible δ, on a

$$\begin{aligned} f(x^* + \delta) &= \mathcal{L}(x^* + \delta, \lambda^*) \\ &= \mathcal{L}(x^*, \lambda^*) + \delta^T \nabla_x \mathcal{L}(x^*, \lambda^*) + \frac{1}{2}\delta^T W^* \delta + o(\delta^T \delta) \\ &= f(x^*) + \frac{1}{2}\delta^T W^* \delta + o(\delta^T \delta) \end{aligned} \tag{2.6}$$

avec $W^* = \nabla_x^2 \mathcal{L}(x^*, \lambda^*) = \nabla_x^2 f(x^*) - \sum_i \lambda_i^* \nabla_x^2 c_i(x^*)$.

En passant à la limite, si $f(x^*)$ est un minimum alors on doit avoir $s^T W^* s \geq 0$ pour toutes les directions admissibles. Ceci détermine une condition nécessaire.

Pour le cas des contraintes inégalités, pour conserver l'égalité (2.6), il faut restreindre les directions admissibles aux directions s telles que $a_i^{*T} s = 0$ si $\lambda_i^* > 0$ sinon (2.6) devient une inégalité.

Dans le cas général :

La condition suffisante vient de :

Théorème 29 ***Condition Suffisante de minimum local***

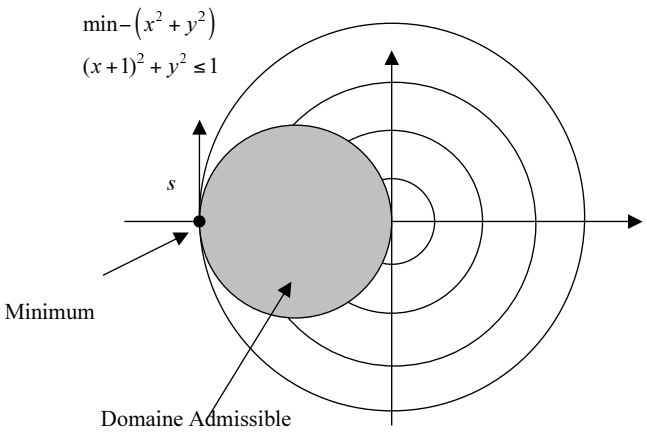


FIGURE 2.4 – Exemple où $s^T \nabla^2 f s < 0$ à l'optimum (courbure négative)

Si (x^*, λ^*) *satisfait les conditions d'ordre 1 et si*

$$s^T W^* s > 0, \ \forall s \in G^* \tag{2.7}$$

où

$$G^* = \{s : s \neq 0, \ a_i^{*T} s = 0, \ i \in \mathcal{A}_+^*, \ a_i^{*T} s \geq 0, \ i \in \mathcal{A}^* \backslash \mathcal{A}_+^*\}$$

et

$$\mathcal{A}_+^* = \{i : i \in E \text{ ou } \lambda_i^* > 0\} \text{ contrainte fortement active}$$

alors x^* *est un minimum local.*

Preuve : Supposons que x n'est pas un minimum local. alors il existe une séquence de D, $x_k \to x^*$ telle que

$$f_k \leq f^*. \tag{2.8}$$

Fixons $\|s_k\| = 1$ comme dans (2.1), par compacité on peut donc en extraire une sous suite qui converge vers s ($\in \mathcal{F}$). D'après (2.8) un développement limité conduit à la condition $s^T g^* \leq 0$. On a alors les deux alternatives suivantes qui ménent à des contradictions :

– Soit $s \notin G$; alors $\exists i : \lambda_i^* > 0$ et $a_i^{*T} s > 0$ or

$$0 \geq s^T g^* = \sum s^T a_i^* \lambda_i^* > 0$$

– Soit $s \in G$; Comme x_k est admissible, $\mathcal{L}(x_k, \lambda^*) \leq f_k$, et depuis (2.6),

$$0 \geq f_k - f^* = \frac{1}{2} \alpha_k^2 s_k^T W^* s_k + o(\alpha_k^2)$$

et la limite est en contradiction avec (2.7).

Attention : La condition porte bien sur le Hessien du Lagrangien W et non sur le Hessien de f comme en témoigne l'exemple de la figure (2.4).

2.3 La programmation linéaire

Problème Linéaire standard :
Trouver min $c^T x$ sous la contrainte $Ax = b$, $x \geq 0$ avec $A_{m \times n}$, $m \leq n$

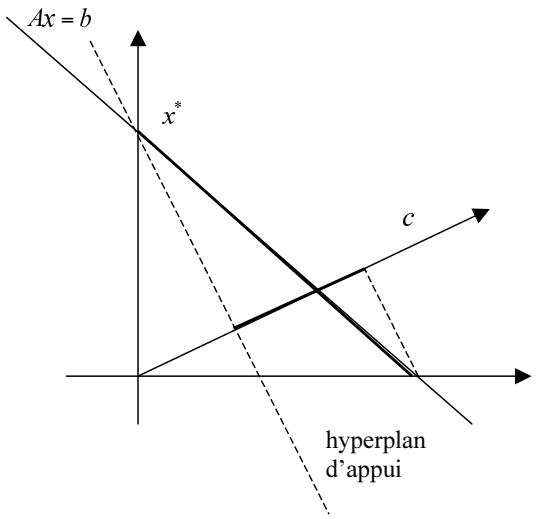


FIGURE 2.5 – Minimum dans $\mathbb{R}^2$, pour une contrainte

Remarque 30 *Tout problème linéaire (f linéaire, c_i linéaires) peut s'écrire sous forme standard*

Remarque 31 *Cas où $x_{\min} \leq x$. La contrainte se transforme en $0 \leq x - x_{\min}$ et on effectue le changement de variable $u = x - x_{\min}$ pour obtenir la forme standard.*

Remarque 32 *Cas où $Ax \leq b$. On considère une variable additionnelle z, on pose $Ax + z = b$, $z \geq 0$ et on retrouve la forme standard.*

Pour un problème de ce type, l'ensemble des points admissibles est un polytope ou simplexe et au moins une solution est située sur l'un des sommets du polytope.

Exemple : Dans R^2, une contrainte définit une droite. La contrainte inégalité $x \geq 0$ réduit cette droite à un segment dont les extrémités se situent sur les axes. La projection orthogonale des contraintes sur la droite αc donne le minimum (Figure 2.5).

Dans R^3, une contrainte définit un plan. La contrainte inégalité $x \geq 0$ réduit ce plan à un triangle. Le minimum obtenu à l'aide de l'hyperplan d'appuie associé à c, se situe alors sur un sommet de D dont deux composantes x_i sont nulles (Figure 2.6). Si on a deux contraintes ; cela définit un segment de droite et le minimum est obtenu à une extrémité soit pour une composante x_i nulle.
[1]

Idée : Exploiter le fait qu'il y a $n - m$ composantes de la solution x^* nulles. Comme il faut éviter de tester tous les sommets possibles car C_n^{n-p} est en général très grand. L'astuce consiste à privilégier des déplacements de sommet en sommet qui entrainent une réduction du coût.

On part d'un vecteur ayant $n - m$ composantes nulles. On écrit alors

$$x^T = [x_B^T; x_N^T]$$

avec x_B : variable "basic" et x_N : variable "nulle".

1. Un hyperplan H de dimension $(n-1)$ est dit supporter un ensemble fermé et convexe $M(\subset \mathbb{R}^n)$ au point $y \in \partial M \cap H$ si M est entièrement situé dans l'un des deux demi-espaces définis par H (∂M désigne la frontière de l'ensemble M). Si un vecteur h est normal intérieur à cet hyperplan d'appui H de M au point y alors $h^T y = \min_{z \in M} h^T z$.

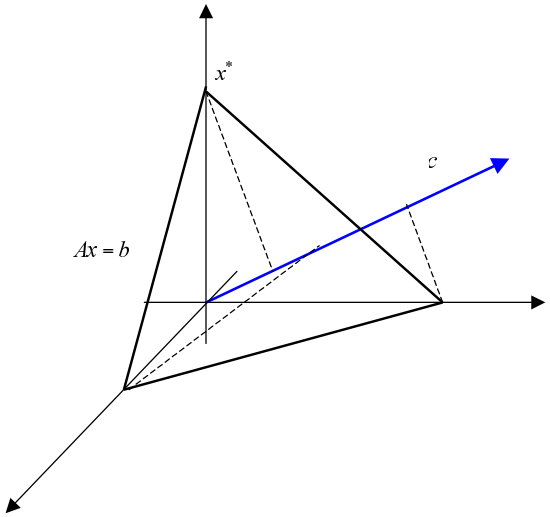


FIGURE 2.6 – Minimum pour une contrainte dans $\mathbb{R}^3$

$$A = [A_B; A_N]$$

où A_B est non singulière si A est de plein rang.

Il vient

$$Ax = A_B x_B + A_N x_N = b$$

Soit $\begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} A_B^{-1} b \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{b} \\ 0 \end{bmatrix}$ et la contrainte $x_i > 0$ donne $\hat{b} > 0$.
La fonction coût est alors réduite à :

$$f = c^T x = c_B^T x_B$$

avec $c^T = [c_B^T; c_N^T]$.

On cherche à présent à évaluer l'influence d'une perturbation de x_N sur f.

Si x_N est non nul, on a

$$x_B = A_B^{-1}(b - A_N x_N) = \hat{b} - A_B^{-1} A_N x_N,$$

d'où

$$\begin{aligned} f(x_N) &= c_B^T(\hat{b} - A_B^{-1} A_N x_N) + c_N^T x_N \\ &= \hat{f} + \hat{c}^T x_N \text{ avec } \hat{c} = c_N - A_N^T A_B^{-T} c_B \end{aligned}$$

Comme les variations des x_N doivent être positives ($x \geq 0$), on voit que x est optimal si

$$\hat{c} \geq 0 \tag{2.9}$$

Sinon on choisit la composante $\hat{c}_q$ la plus négative.
Soit

$$\hat{c}_q = \min_i \hat{c}_i, \hat{c}_i < 0$$

et on fait baisser le critère en augmentant x_q tout en satisfaisant la contrainte $Ax = b$. Mais attention la valeur de x_q est également limitée par la contrainte $x \geq 0$.

Regardons l'effet de x_q sur x_B :

$$\begin{aligned} x_B &= \hat{b} - A_B^{-1} a_q x_q \\ &= \hat{b} + d x_q \end{aligned}$$

avec a_q la q ième colonne de A_N et $d = -A_B^{-1} a_q = -\hat{a}_q$

On observe que si x_q augmente et si $d_i < 0$ alors x_{B_i} diminue.

La première composante de x_B a atteindre la contrainte $x_i = 0$ est alors obtenue pour

$$x_q = \frac{\hat{b}_p}{-d_p} = \min_{\substack{i \in B \\ d_i < 0}} \frac{\hat{b}_i}{-d_i}$$

Pour cette valeur de x_q, la contrainte $x_{B_p} = 0$ devient active. On recommence donc en permutant les éléments de la colonne p avec la colonne q car x_q n'est plus nul et $x_{B_p} = 0$. Et ainsi de suite jusqu'à l'obtention de (2.9).

Sous la forme d'un tableau cela donne :

c_B^T	c_N^T	0
A_B	A_N	b

Puis par la méthode d'élimination de Gauss

0	$\hat{c}^T$	$-\hat{f}$
Id	$\hat{A}_N = A_B^{-1} A_N$	$\hat{b}$

1. On choisit alors la colonne correspondant au terme $\hat{c}_q$ le plus négatif. Soit a_q la partie de la colonne correspondant à $\hat{A}_N$.
2. Puis on sélectionne la ligne i pour laquelle le rapport $\frac{\hat{b}_i}{a_{qi}}$ est minimum avec $a_{qi} > 0$.
3. On utilise alors l'élément comme nouveau pivot de Gauss
4. On revient au point 1 tant qu'il existe des valeurs de $\hat{c}_q$ négatives.

Exemple 33 *A titre d'exemple, considérons le problème suivant : Trouver*

$$\min x_1 + 2x_2 + 3x_3 + 4x_4$$

sous la contrainte

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 1 \\ x_1 + x_3 - 3x_4 &= 1/2 \\ x &\geq 0 \end{aligned}$$

$k = 0$

	c^T				
f	1	2	3	4	0
	1	1	1	1	1
	1	0	1	−3	1/2

$k = 1$

f	0	0	2	-1	−3/2
x_1	1	0	1	−3	1/2
x_2	0	1	0	4	1/2

$k = 2$

f	0	1/4	2	0	−11/8
x_1	1	3/4	1	0	7/8
x_4	0	1/4	0	1	1/8

d'où la solution $x^* = (7/8, 0, 0, 1/8)$ et $f^* = 11/8$.

Exemple 34 *En utilisant une variable additionnelle $z = (z_1, z_2)$ (voir la remarque en début de section) déterminer*

$$\min x_1 + 2x_2 + 3x_3 + 4x_4$$

sous la contrainte

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &\leq 1 \\ x_1 + x_3 - 3x_4 &\leq 1/2 \\ x &\geq 0 \end{aligned}$$

2.4 La programmation quadratique

Trouver

$$\min q(x) = \frac{1}{2} x^T H x + g^T x$$

sous la contrainte

$$\begin{aligned} a_i^T x &= b_i, i \in E \\ a_i^T x &\geq b_i, i \in I \end{aligned}$$

avec H symétrique
Généralités : Si $H \geq 0$, $\exists$ un minimum global. $(\mathcal{R} \neq \varnothing)$
Si $H > 0$, $\exists!$ un minimum global

2.4.1 Cas où $I = \varnothing$: Les méthodes d'éliminations et Lagrangiennes.

Trouver

$$\min q(x) = \frac{1}{2} x^T H x + g^T x$$

sous la contrainte

$$A^T x = b$$

Hypothèses : $m < n$ et rang $A = m$

Résolution directe (Les méthodes d'éliminations) **Idée :** Transformer par substitution le problème initial en un problème sans contrainte

En posant $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ avec $x_1 \in \mathbb{R}^m$ et $x_2 \in \mathbb{R}^{n-m}$, $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, $g = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$, $H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$, il vient

$$x_1 = A_1^{-T}(b - A_2^T x_2).$$

Puis par substitution dans $q(x)$, on est ramené à résoudre **un problème quadratique** $\min \Psi(x_2)$ **sans contrainte** pour lequel la condition nécessaire $\nabla \Psi(x_2^*) = 0$ permet de déterminer l'unique solution.

Problème : A_1 doit être inversible.

Les méthodes d'éliminations généralisées Posons $Y_{n\times m}$ et $Z_{n\times n-m}$ tels que $[Y : Z]$ est non singulière et

$$A^T Y = Id$$
$$A^T Z = 0$$

Y^T est l'inverse généralisée à gauche de A ($Y^T A = Id$). On voit alors que

$$x = Yb$$

est une solution de $A^T x = b$.

En fait, toutes les solutions s'écrivent :

$$x = Yb + \delta$$

avec $\delta \in Ker(A^T)$ et $\delta = Zy$, $y \in \mathbb{R}^{n-m}$.

En résumé, tout point admissible vérifie

$$x = Yb + Zy.$$

On remplace alors dans $q(x)$, x par $Yb + Zy$ d'où

$$\Psi(y) = \frac{1}{2} y^T Z^T H Z y + (g + HYb)^T Zy + \frac{1}{2}(g + HYb)^T Yb.$$

Si $Z^T HZ > 0$, il existe un unique y^* déterminé par $\nabla\Psi(y^*) = 0$. Soit

$$Z^T HZ y^* = -Z^T(g + HYb)$$

Par ailleurs, en appliquant les conditions KT (2.5), λ^* est obtenu par

$$\lambda^* = Y^T(Hx^* + g).$$

Les choix de Y et Z peuvent être fait de différentes façons :

– Par **une factorisation QR**

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [\begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R$$

avec $Q_{n\times n}$ orthogonal ($Q^{-1} = Q^T$) et R triangulaire supérieure.(Q_1 de dimension $n \times m$, Q_2 de dimension $n \times (n-m)$, R de dimension $m \times m$)

Alors

$$Y = Q_1 R^{-T}$$

et

$$Z = Q_2$$

Remarque : On peut montrer que dans ce cas Yb correspond à la projection orthogonale de l'origine sur la contrainte et Zy au déplacement à faire sur la contrainte pour atteindre le minimum depuis Yb.

Mise en pratique : On résout

$$Yb = Q_1 R^{-T} b$$

en deux étapes. D'abord,

$$R^T u = b$$

puis

$$Yb = Q_1 u.$$

– D'une façon générale, Y et Z peuvent être calculés en choisissant V tel que $[A : V]$ est non singulière ($V_{n\times(n-m)}$) alors $[A : V]^{-1} = \begin{bmatrix} Y^T \\ Z^T \end{bmatrix}$. Dans le premier cas (élimination directe) $V = \begin{bmatrix} 0 \\ Id \end{bmatrix}$ et dans le second (factorisation QR) $V = Q_2$.

Les méthodes Lagrangiennes On utilise la condition nécessaire sur le Lagrangien, soit :

$$L(x, \lambda) = \frac{1}{2}x^T Hx + g^T x - \lambda^T (A^T x - b)$$

et

$$\nabla_x L^* = 0 : Hx^* + g - A\lambda^* = 0$$
$$\nabla_\lambda L^* = 0 : A^T x^* - b = 0$$

Soit encore le système à résoudre $\begin{bmatrix} H & -A \\ -A^T & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = - \begin{bmatrix} g \\ b \end{bmatrix}$

$\begin{bmatrix} H & -A \\ -A^T & 0 \end{bmatrix}$ est symétrique mais pas >0

Si l'inverse existe, on peut écrire :

$$x^* = -Gg + Tb$$
$$\lambda^* = T^T g - Ub$$

avec

$$\begin{bmatrix} H & -A \\ -A^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} G & -T \\ -T^T & U \end{bmatrix}$$

Dans la pratique, la **"Null space méthod"** donne

$$G = Z(Z^T HZ)^{-1} Z^T$$
$$T = Y - GHY$$
$$U = Y^T HGHY - Y^T HY$$

avec Y et Z définis comme pour les méthodes d'éliminations généralisées.

Mise en pratique : On utilise la factorisation de Choleski pour calculer $Z^T HZ = S^T S$ (l'inverse existe si et seulement si $Z^T HZ$ est non singulière). Notons tout de même qu'il existe de nombreuses variantes dans la façon de choisir Z et Y.

2.4.2 Cas général $I \neq \varnothing$: La méthode des contraintes actives

Idée : Résoudre une suite de problèmes quadratiques avec contraintes égalités et utiliser un mécanisme d'ajout-suppression de contraintes pour parvenir à la solution du problème initial.

On note $\mathcal{A}(x) = \{i : c_i(x) = 0\}$ l'ensemble des contraintes actives au point x.

On applique l'algorithme suivant (2.7) :

Pour k=0,1,...

1. A partir du point x_k admissible, on détermine $\mathcal{A}(x_k)$ et on résout le problème initial en ne prenant en compte que les contraintes actives.

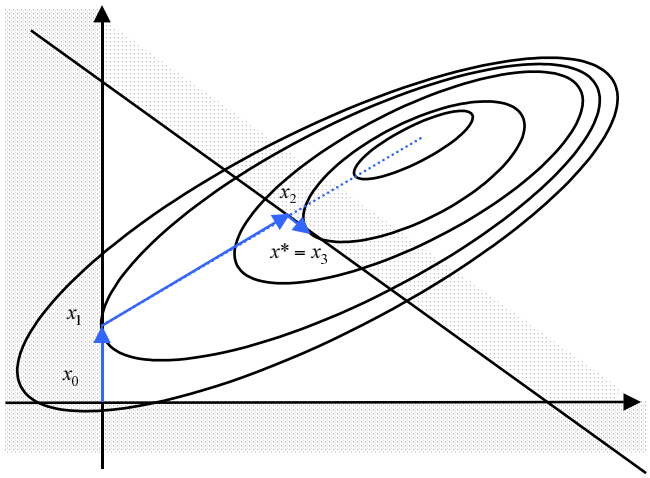


FIGURE 2.7 – La méthode des contraintes actives

Autrement dit, on cherche à résoudre

$$\min_x \frac{1}{2} x^T H x + g^T x \tag{2.10}$$
$$a_i^T x = b_i,\ i \in \mathcal{A}(x_k)$$

si on note $x = x_k + \delta$, ce problème est équivalent à (à faire en exercice) :

$$\min_\delta \frac{1}{2} \delta^T H \delta + g_k^T \delta \tag{2.11}$$
$$a_i^T \delta = 0, i \in \mathcal{A}(x_k)$$

avec $g_k = g + Hx_k$.
On peut alors résoudre ce problème (2.11) par les méthodes sans contraintes inégalités vues ci-dessus. Notons δ_k la solution obtenue.

2. Si δ_k est admissible pour les contraintes $i \notin \mathcal{A}(x_k)$ alors

$$x_{k+1} = x_k + \delta_k$$

3. Sinon il existe un indice $i \notin \mathcal{A}(x_k)$ tel que :

$$a_i^T (x_k + \delta_k) < b_i,\ i \notin \mathcal{A}(x_k)$$

et il faut donc réduire le déplacement δ_k. Comme $a_i^T x_k \geq b_i$ ($x_k \in D$), on a : $a_i^T \delta_k < 0$.

Observer que si $a_i^T (x_k + \delta_k) \geq b_i$ et $a_i^T \delta_k < 0$ alors $1 \leq \frac{b_i - a_i^T x_k}{a_i^T \delta_k}$ et cette inégalité est donc violée pour au moins un indice i.
On choisit alors

$$x_{k+1} = x_k + \alpha_k \delta_k$$

avec

$$\alpha_k = \min(1, \min_{\substack{i \notin \mathcal{A}(x_k) \\ a_i^T \delta_k < 0}} \frac{b_i - a_i^T x_k}{a_i^T \delta_k})$$

4. Si $\alpha_k < 1$, une nouvelle contrainte i devient active, on l'ajoute et on recommence en 1.
5. Si $\delta_k = 0$ et résout (2.11), on calcule le multiplicateur de Lagrange λ_i^k, $i \in \mathcal{A}(x_k)$.

(a) Si $\lambda_i^k \geq 0$ on arrête et $x^* = x_k$ (C.N. sont satisfaites)

(b) Sinon il existe un indice q tel que $\lambda_q^k < 0$ et $\lambda_q^k = \min_{i \in \mathcal{A}(x_k) \cap I} \lambda_i^k$, on enlève la contrainte correspondante de $\mathcal{A}(x_k)$ (on relâche la contrainte égalité puisque l'on peut améliorer le résultat sans elle) et on recommence en 1.

Exemple 35 *Mettre en oeuvre l'algorithme des contraintes actives en partant de* $x_0 = (0.5, 0)$ *pour déterminer*

$$\min(x_1 - 1)^2 + (x_2 - 2)^2$$

sous la contrainte

$$\begin{aligned} x_1 + 2x_2 &\geq 2 \\ x &\geq 0 \end{aligned}$$

Même chose en partant $x_0 = (0, 0.5)$

2.5 La programmation non linéaire

Les méthodes principales sont la pénalisation et **SQP** (Sequential Quadratic Programming).

2.5.1 La méthode de pénalisation

Cas où $I = \varnothing$

Idée : Résoudre une suite de problèmes sans contraintes dont les solutions conduisent à celle du problème originale avec contraintes.

On considère la fonction de pénalisation $\Phi(x, \sigma) = f(x) + \frac{1}{2}\sigma c^T(x)c(x)$ avec σ un nombre positif qui déterminent le poids de la pénalisation.

On cherche alors le minimum suivant x de $\Phi(x, \sigma)$ pour σ fixé. Puis on recommence en faisant tendre σ vers l'infini. La limite des solutions pénalisées fournie la solution de problème original.

Avantage : Facile à implémenter

Inconvénients : Lorsque $\sigma \nearrow$, on obtient un mauvais conditionnement de $\nabla^2\Phi$ ce qui pose des problèmes de convergence. En effet $\nabla^2\Phi_k = W_k + \sigma_k A_k A_k^T$ et comme $A_k^T A_k$ est de rang m, il existe m valeurs propres de $\nabla^2\Phi_k$ qui tendent vers $+\infty$ avec σ_k.

Algorithme :

i. Choisir une séquence $\sigma_k \to +\infty$ par exemple {1,10,100,...}

ii. Pour chaque σ_k, trouver le minimum $x^*(\sigma_k)$

iii. Terminer lorsque $c(x^*(\sigma_k))$ est suffisamment petit

Attention : Dû au mauvais conditionnement, les choix de σ_1 puis de la vitesse d'évolution de σ_k sont délicats et la méthode peut se révéler inefficace.

2.5.2 La méthode SQP (Sequential Quadratic Programming)

C'est le standard pour le cas général.

Idée : Remplacer f non linéaire par l'approximation quadratique au point x_k suivante :

$$q_k(d) = g_k^T d + \frac{1}{2} d^T W_k d$$

où $W_k = \nabla_{xx}^2 L(x_k, \lambda_k))$ et les contraintes par leur linéarisation au point x_k. Soit :

$$\begin{aligned} \min_d q_k(d) &= g_k^T d + \frac{1}{2} d^T W_k d \\ c_i(x_k) + a_i^T(x_k) d &\geq 0, i \in I \\ c_i(x_k) + a_i^T(x_k) d &= 0, i \in E \end{aligned} \tag{2.12}$$

La CN donne (cas $I = \varnothing$)

$$\begin{aligned} W_k d + g_k &= A_k \lambda \\ c_k + A_k^T d &= 0 \end{aligned}$$

La méthode se justifie dans le cas $I = \varnothing$ en considérant l'approximation à l'ordre 1 du gradient de la fonction Lagrangienne :

$$\nabla L(x_k + \delta x, \lambda_k + \delta\lambda) \simeq \nabla L_k + \nabla^2 L_k \begin{bmatrix} \delta x \\ \delta\lambda \end{bmatrix}$$

On impose la condition nécessaire de minimum

$$\nabla L(x_k + \delta x, \lambda_k + \delta\lambda) = 0,$$

et on trouve :

$$\nabla^2 L_k \begin{bmatrix} \delta x \\ \delta\lambda \end{bmatrix} = -\nabla L_k$$

Soit

$$\begin{pmatrix} W_k & -A_k \\ -A_k^T & 0 \end{pmatrix} \begin{bmatrix} \delta x \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} -g_k + A_k\lambda_k \\ c_k \end{bmatrix}.$$

Autrement dit, on applique Newton sur le Lagrangien.
En posant, $\lambda_{k+1} = \lambda_k + \delta\lambda$ et $d_k = \delta x$, on parvient à :

$$\begin{pmatrix} W_k & -A_k \\ -A_k^T & 0 \end{pmatrix} \begin{bmatrix} d_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -g_k \\ c_k \end{bmatrix}$$

c'est la CN que l'on obtient sur le problème posé ci-dessus..

Algorithme

Pour $k = 1, 2, 3, ...$

1. Résoudre le problème quadratique (2.12) pour déterminer $d_k = x_{k+1} - x_k$ et λ_{k+1} (multiplieurs de Lagrange sur les contraintes linéarisées)
2. Poser $x_{k+1} = x_k + \alpha_k d_k$ où α_k est choisi pour obtenir une réduction suffisante sur une fonction de mérite (ou pénalisation) $\Psi(x) = f(x) + \sum_{i \in E} \upsilon_i |c_i(x)| - \sum_{i \in I} \upsilon_i \min(c_i(x), 0)$, $\upsilon_i > 0$.

Convergence : La convergence est assurée localement si (x^*, λ^*) satisfont les conditions de second ordre. Si le point x_0 est suffisamment proche de x^* et si λ_k reste suffisamment proche de λ^* alors la séquence $x_{k+1} = d_k + x_k$ converge vers x^* avec une vitesse d'ordre 2.

Il n'y a pas de garantie dans les autres cas. Le code doit pouvoir modifier le problème (2.12) lorsque q_k n'est pas borné inférieurement sur l'ensemble des points admissibles ou lorsque cet ensemble est vide.

Mise en oeuvre : A chaque itération, au lieu de calculer $\nabla^2_{xx} L(x_k, \lambda_k)$, on préfère utiliser **la méthode BFGS** avec

$$s_k = x_{k+1} - x_k$$
$$y_k = \nabla_x L(x_{k+1}, \lambda_k) - \nabla_x L(x_k, \lambda_k)$$

et (ici $H_k = B_k^{-1}$ voir la section consacrée BFGS-DFP)

$$H_{k+1} = H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

Attention car H_k n'est pas définie positive contrairement au cas sans contrainte. Cependant on peut procéder ainsi :

Quand $y_k^T s_k$ n'est pas suffisamment positif, y_k est réinitialisé à

$$y_k = \theta_k y_k + (1 - \theta_k) H_k s_k,$$

$\theta_k \in [0\ 1]$ et proche de 1 tel que pour $\sigma \in [0\ 1]$

$$y_k^T s_k \geq \sigma s_k^T H_k s_k$$

Régions de confiance (trust région) Idée : l'approximation quadratique de f, notée q_k est valable uniquement dans un voisinage de x_k. Certains algorithmes limitent la recherche sur une région de confiance définie par :

$$\|D_k d\| \leq \Delta_k$$

avec $\Delta_k > 0$. La vitesse de convergence est alors réduite mais la stabilité de ces méthodes augmente.

2.6 Cas des problèmes convexes

Définition 36 *K est convexe si $\forall x_0, x_1 \in K$, $\forall \theta \in [0\ 1]$, $\theta x_0 + (1-\theta) x_1 \in K$*

ou encore pour tout $x_i \in K$, $i = 1, ..., m$ alors tout x s'écrivant $x = \sum_{i=1}^m \theta_i x_i$ avec $\sum_i \theta_i = 1$ et $\theta_i \geq 0$, appartient à K.

Définition 37 *Une fonction est dite convexe sur K si et seulement si $\forall x_0, x_1 \in K$, $f(\theta x_0 + (1-\theta) x_1) \leq \theta f(x_0) + (1-\theta) f(x_1)$.*

Propriétés :

1. $f_1 \geq f_0 + (x_1 - x_0)^T \nabla f_0, \forall x_0, x_1 \in K$ (la tangente est sous la courbe
2. $\nabla^2 f_0 \geq 0, \forall x_0 \in K$ (courbure positive)
3. si f_i, $i = 1, 2, ..., m$ sont convexes sur K et si $\lambda_i \geq 0$ alors $\sum_{i=1}^m \lambda_i f_i$ est convexe.
4. Si $c(x)$ est concave alors l'ensemble $S = \{x : c(x) \geq k\}$ est convexe.

Définition 38 *Un problème d'optimisation est dit convexe lorsque f est convexe sur le domaine des points admissibles. Autrement dit, on cherche à résoudre le problème dit primal :*

$$\textit{Problème primal} : \min f(x)$$
$$\textit{sous la contrainte } x \in K = \{x : c_i(x) \geq 0,\ i = 1, ..., m\}$$

avec f convexe sur K et les $c_i(x)$ $i = 1, ..., m$ concaves.

Proposition 39 *Toute solution d'un problème convexe est une solution globale. L'ensemble des solutions est un ensemble convexe. Si f est strictement convexe, la solution est unique.*

Proposition 40 *Si f et c_i, $i = 1, ..., m \in C^1$ et si les conditions nécessaires de minimum sont satisfaites alors elles sont suffisantes.*

2.6.1 Dualité

Théorème 41 *Si x^* résout le problème primal, si $\mathcal{F}^* \cap \mathcal{D}^* = F^* \cap \mathcal{D}^* = \varnothing$ (??) et si f et c_i, $i = 1, ..., m \in C^1$ alors x^*, λ^* résolvent le problème dual :¨*

$$\textbf{\textit{Problème dual}} : \max_{x,\lambda} L(x, \lambda)$$
$$\textit{sous la contrainte } \nabla_x L(x, \lambda) = 0, \lambda \geq 0$$

De plus , $f^* = L(x^*, \lambda^*)$

Preuve : Pour $\lambda \geq 0$,

$$L(x^*, \lambda^*) = f^* \geq f^* - \sum_{i=1}^{m} \lambda_i c_i^* = L(x^*, \lambda), \text{ puisque } \lambda_i \geq 0, c_i^* \geq 0$$

La convexité de $L = f - \sum_{i=1}^{m} \lambda_i c_i$ (voir propriétés ci-dessus) impliquent

$$\begin{aligned} L(x^*, \lambda^*) &\geq L(x, \lambda) + (x^* - x)^T \nabla_x L(x, \lambda) \\ &\geq L(x, \lambda) \text{ pour } x, \lambda \text{ admissibles} \end{aligned}$$

Exemple 1 : Problème linéaire sous une forme non standard. Le problème primal

$$\begin{aligned} \text{(Primal)} \ &: \min f_0 + c^T x \\ & A^T x \geq b \end{aligned}$$

admet la formulation dual

$$\begin{aligned} \text{(Dual)} \ &: \max f_0 + b^T \lambda \\ & A\lambda = c, \lambda \geq 0 \end{aligned}$$

qui est la formulation d'un problème linéaire standard.

Exemple 2 : Le problème primal

$$\begin{aligned} \text{(Primal)} \ &: \min f_0 + c^T x \\ & A^T x \geq b, x \geq 0 \end{aligned}$$

admet la même formulation duale

$$\begin{aligned} \text{(Dual)} \ &: \max f_0 + b^T \lambda \\ & A\lambda \leq c, \lambda \geq 0 \end{aligned}$$

mais si A a nettement moins de ligne que de colonne alors le problème dual est plus avantageux à résoudre numériquement.

Exemple 3 : Le problème primal

$$\begin{aligned} \text{(Primal)} \ &: \min \frac{1}{2} x^T G x + g^T x \\ & A^T x \geq b \end{aligned}$$

avec $G > 0$, admet la formulation duale

$$\begin{aligned} \text{(Dual)} \ &: \max -\frac{1}{2} \lambda^T (A^T G^{-1} A) \lambda + \lambda^T (b + A^T G^{-1} g) - -\frac{1}{2} g^T G^{-1} g \\ & \lambda \geq 0 \end{aligned}$$

C'est également un problème quadratique mais sa résolution est plus aisée car il ne comporte qu'une contrainte de type borne.

2.6.2 Les algorithmes de points intérieurs

Idée : Eviter comme dans l'algorithme du simplexe de glisser le long des contraintes et donc conserver une certaine distance avec celles-ci afin de pouvoir avancer à plus grands pas [**?**].

Soit un problème linéaire dans sa forme standard :

$$\text{(Primal)}\ \min c^T x : A^T x = b, x \geq 0$$

et son dual :

$$\text{(Dual)}\ \max b^T \lambda : c - A\lambda - s = 0, \lambda \geq 0, s \geq 0$$

Le Lagrangien $L(x, \lambda, s) = c^T x - \lambda^T (A^T x - b) - s^T x$ donne comme C.N. :

$$\begin{aligned} A^T x &= b \\ A\lambda + s &= c \\ s_i x_i &= 0 \\ x &\geq 0 \\ s &\geq 0 \end{aligned}$$

Notons : $S = diag(s)$, $X = diag(x)$, $e^T = [\ 1 \quad 1 \quad \dots \quad 1\ ]$

$$s_i x_i = 0 \Leftrightarrow SXe = 0$$

Il s'agit à présent de créer une suite (x_k, λ_k, s_k) telle que

$$\begin{aligned} x_k > 0, s_k &> 0 \\ \left\| A^T x_k - b \right\| &\to 0, \\ \| A\lambda_k + s_k - c \| &\to 0, \\ s_k^T x_k &\to 0 \end{aligned}$$

On utilise alors une méthode de Newton avec correction : Pour un triplet (x_k, λ_k, s_k) et $\sigma_k \in \left[0 \quad 1\right]$ la direction de recherche est donnée par (w, z, t) satisfaisant le système linéaire :

$$\begin{aligned} A^T w &= b - A^T x_k \\ Az + t &= c - A\lambda_k - s_k \\ S_k w + X_k t &= -S_k X_k e + \sigma_k \mu_k e \\ \mu_k &= x_k^T s_k \end{aligned}$$

Pour $\sigma_k = 0$, on retrouve l'algorithme de Newton. σ_k a pour but d'éloigner de la frontière $s = x = 0$ ce qui permet de prendre des grands pas sur α_k.

Mise à jour :

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k^P w_k \\ \lambda_{k+1} &= \lambda_k + \alpha_k^D z_k \\ s_{k+1} &= s_k + \alpha_k^D t_k \end{aligned}$$

avec α_k^P tel que $x_{k+1} > 0$ et α_k^D tel que $s_{k+1} > 0$.

2.7 Cas discret : La programmation en nombre entier.

De nombreux problèmes n'acceptent de solutions qu'en nombres entiers. Par exemple le célèbre problème du sac à dos : On considère un sac d'une capacité P que l'on cherche à remplir avec m produits de façon à maximiser l'utilité du chargement. Soit p_i le poids de chaque objet et u_i son utilité, il nous faut donc résoudre $\max_x \sum_{i=1}^{m} u_i x_i$ sous la contrainte $\sum_{i=1}^{m} p_i x_i \leq P$. Soit la formulation générale :

$$(P_I) \text{ Trouver } : \min_x f(x)$$
$$\text{sous la contrainte } x_i \in \mathbb{Z}$$

Idée : On résout (P_I) sur $\mathbb{R}$. Soit $\hat{x}$ le minimum obtenu
- ou bien $\hat{x} \in \mathbb{Z}^n$ et on a fini
- ou bien $\hat{x} \notin \mathbb{Z}^n \Rightarrow$ il existe une composante $\hat{x}_i \notin \mathbb{Z}$. On crée alors deux sous problèmes (branches).

$$(P^-) : \min_x f(x) \qquad (P^+) : \min_x f(x)$$
$$x \in \mathbb{R}, x_i \leq E(\hat{x}_i) \qquad x \in \mathbb{R}, x_i \geq E(\hat{x}_i) + 1$$

  Notons que si x est admissible pour (P^-), il ne l'est pas pour (P^+) et réciproquement.
- En itérant, on parvient à un arbre binaire qui se termine soit par une solution partielle (locale) admissible $\square$, soit par un problème sans point admissible $\bullet$.
- Il n'est pas nécessaire de développer tout l'arbre et de résoudre tous les sous problèmes d'optimisation. En effet, on peut remarquer que :

1. la valeur de f pour un problème enfant ne peut être que supérieure à la valeur de f pour le problème parent (on réduit le domaine de recherche).
2. si on trouve une solution locale avec un coût de f_i alors toute branche partant d'un noeud j tel que $f_j > f_i$ ne peut que donner un résultat plus mauvais. Donc on n'explore pas ces branches d'où l'algorithme suivant (figure 2.8) :

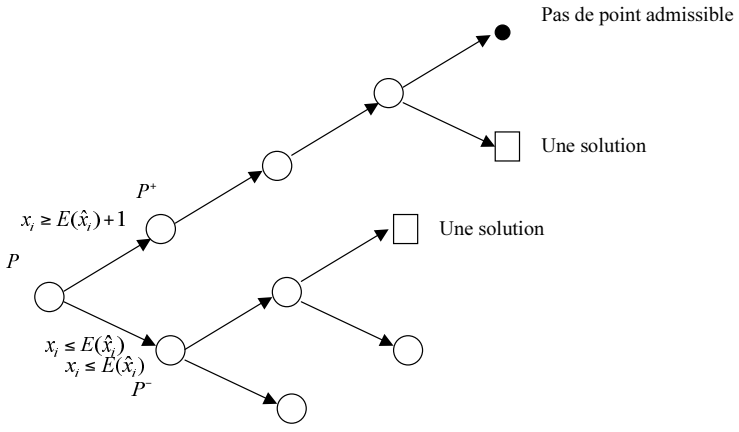


FIGURE 2.8 – Arbre binaire des sous problèmes

Algorithme : "Branch and Bound"

On crée une pile FIFO (First In ,First Out) de sous problèmes avec une borne inférieure L sur leur coût associé.

Pour les noeuds $\square$ qui ont été exploré, on note la meilleure solution $\hat{x}$ et $\hat{f}$ ($\hat{f}$ est initialisée à $+\infty$) et L à $-\infty$).

1. Si la pile est vide, on arrête avec $(x^*, f^*) = (\hat{x}, \hat{f})$. Sinon on prend le problème en haut de la pile
2. Si $L \geq \hat{f}$, on rejette le problème, retour en 1.
3. Sinon on essaye de résoudre le problème
4. Si il n'y a pas de solution admissible, on rejette et on retourne en 1.
5. Sinon soit x', f' la solution trouvée.
6. Si $f' \geq \hat{f}$, on rejette le problème et retour en 1.
7. Sinon on sélectionne un indice i tel que $E({x_i}') < {x_i}'$ et on crée deux sous problèmes (branches) , on les place en haut de la pile avec pour borne inférieure $L = f'$ puis retour en 1.

Deuxième partie

Optimisation dynamique

L'objectif de cette seconde partie est de présenter les méthodes de base de la commande optimale ainsi que quelques résultats couramment utilisés en automatique.

Etant donné un système dynamique, le problème posé est la recherche d'une commande qui transfère le système d'un état initial à un état final en minimisant un critère donné.

Deux grandes classes de méthodes sont traitées sur les applications discrètes et continues :

- Les méthodes de type **variationnel**. Ces méthodes reposent sur l'idée simple suivante : *" Si une commande $\hat{u}$ conduisant à la valeur $\hat{J}$ est optimale, alors toute commande $u \neq \hat{u}$ donnera un résultat moins bon "*. C'est à dire une valeur du critère J supérieure à $\hat{J}$ si l'objectif est de minimiser le critère. Si on peut évaluer l'effet d'une variation de la commande sur le critère : une commande $u = \hat{u} + \delta u$ conduit à un critère $J = \hat{J} + \delta J$, alors la condition $\delta J \geq 0 \ \forall \delta u$ permettra de caractériser $\hat{u}$. L'effet de δu sur J se fait par développement des solutions de l'équation d'évolution et du critère autour de la commande $\hat{u}$, candidate à être la commande optimale. On n'obtiendra donc en général que des conditions locales.
  On étudiera pour le cas discret les applications de la **programmation non linéaire** et le **principe du maximum (ou minimum) de Pontryagin** pour le cas continu.
- Les méthodes issues du principe de **programmation dynamique** que l'on peut énoncer grossièrement de la manière suivante :

  "à partir d'un point d'une trajectoire optimale, il faut minimiser ce qu'il reste du critère pour terminer la trajectoire".

  Nous appliquerons au cas discret et au cas continu (équations d'Hamilton-Jacobi-Bellman).

Enfin ces méthodes seront utilisées pour arriver à l'un des outils de base de l'automatique contemporaine : l'optimisation d'un critère quadratique, pour les systèmes linéaires (problème LQ).

Ce support de cours doit permettre à l'enseignant de ne pas détailler l'ensemble des calculs qui sont souvent assez lourds. C'est la raison pour laquelle nous les avons mis dans ce texte. Les démonstrations ne prétendent pas à la rigueur mathématique : lorsque l'on dérive une fonction c'est qu'on a supposé qu'elle était dérivable ! On indique cependant quelques points délicats qui nécessitent une technique mathématique hors de propos ici.

Chapitre 3

Application de la PNL à la commande

Lorsque les problèmes de commande peuvent se résoudre par la minimisation d'une fonction de plusieurs variables, la PNL peut s'appliquer directement.
Nous traiterons deux cas : l'optimisation paramétrique et la commande d'un système à temps discret.

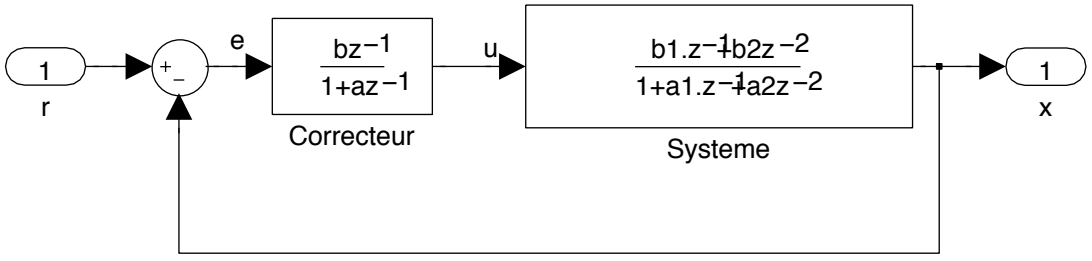
3.1 Optimisation paramétrique

L'optimisation paramétrique consiste à trouver les valeurs d'un ensemble de paramètres de manière à optimiser un critère. Elle est largement utilisée dans les problèmes d'identification : trouver les paramètres d'un modèle qui rendent compte au mieux du comportement d'un système. En commande, le problème est de trouver les meilleurs paramètre d'un correcteur. Pour l'appliquer, il faut disposer du modèle du système à piloter, de la structure du correcteur que l'on va utiliser et du critère à optimiser.

Les avantages de cette méthode sont importants : on peut traiter des problèmes linéaires ou non linéaires, on peut aussi optimiser un correcteur pour un type d'entrée (on n'est pas limité à l'échelon ou la sinusoïde). En revanche il faut avoir déterminé le type de correcteur à utiliser (les cours d'automatique de base peuvent alors être très utiles) et l'optimisation demande, sauf cas très simple, le recours à des algorithmes d'optimisation. Ces algorithmes nécessitent en général un calcul ou une évaluation du gradient du critère à optimiser. Ce gradient ne peut en général pas être calculé de manière directe et nécessite la résolution des équations de sensibilité. C'est ce problème générique que nous allons montrer sur un exemple.

La figure suivante illustre le problème et présente l'exemple sur lequel nous bâtirons l'exposé :
Le problème est alors posé de la façon suivante :

– étant donné le procédé et une structure de correcteur,



– étant donné une entrée choisie à l'avance,
– étant donné un critère (fonction de coût), à minimiser,
quels sont les coefficients du correcteur qui vont minimiser le critère pour l'entrée donnée ?

3.1.1 Modèles mathématiques

Le problème est mis sous la forme mathématique suivante :
– équation du système :

$$x_{n+1} + a_1 x_n + a_2 x_{n-1} = b_1 u_{n-1} + b_2 u_{n-2} \tag{3.1}$$

– équation du correcteur :

$$u_{n+1} + a u_n = b e_n \tag{3.2}$$

– l'erreur est :

$$e_n = r_n - x_n \tag{3.3}$$

– on optimise un critère de type somme :

$$C = \sum_{n=1}^{N} c(e_n, u_n) \tag{3.4}$$

où c est le coût élémentaire qui pénalise l'erreur et l'énergie fournie au système. N est l'horizon sur lequel on fait porter le coût ; il peut être infini.
On choisira un critère quadratique pour des raisons de simplicité

$$c(e_n, u_n) = e_n^2 + \lambda u_n^2 \tag{3.5}$$

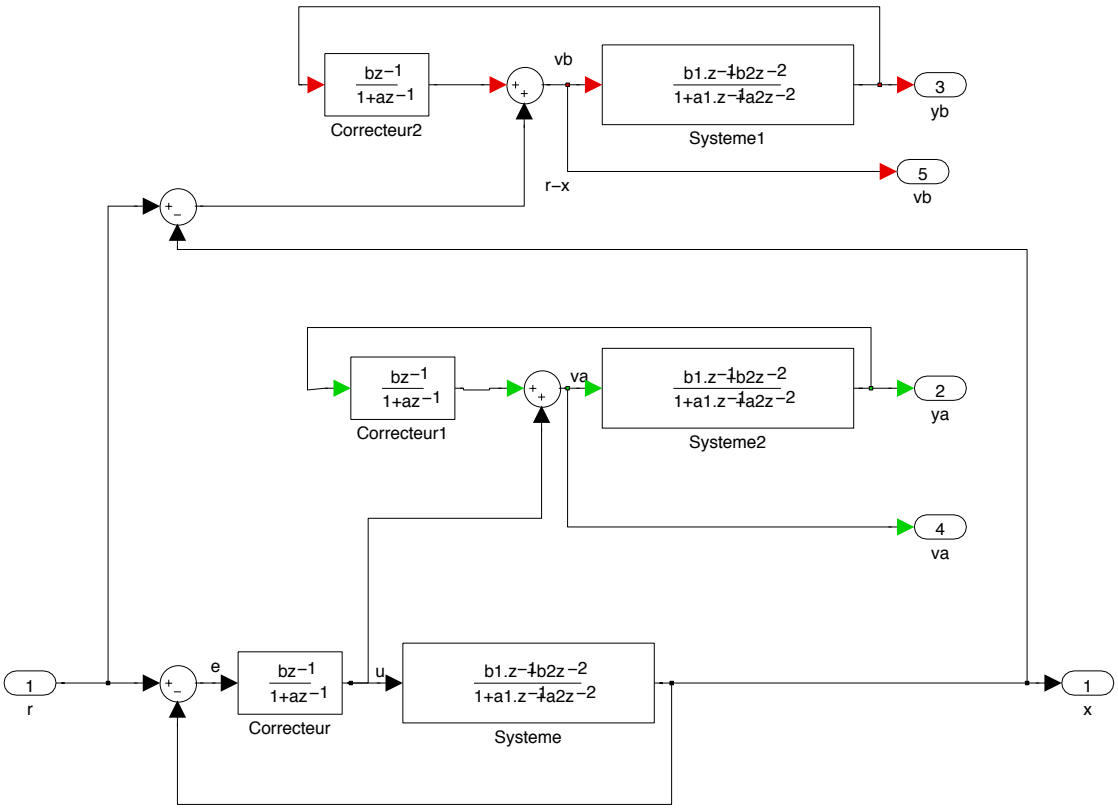
qui minimisera à la fois l'erreur et la commande. Le choix de λ imposera la dynamique du système (plus ou moins rapide, plus ou moins amorti). En pratique, on fait des essais successifs.
L'entrée r_n est donnée, elle devra être représentative des conditions réelles de fonctionnement du système, on prend souvent un échelon, sauf si une entrée spécifique est nécessaire, profil de température par exemple.

3.1.2 Minimisation du critère

Le correcteur est caractérisé par son vecteur de paramètres : $p = [a, b]$.
Les coefficients du correcteur seront optimaux lorsque le critère sera minimum.
On minimisera le critère à l'aide d'un algorithme de Programmation Non Linéaire qui nécessitera le calcul du gradient C_p du critère par rapport au vecteur paramètres.
Il faut donc calculer le gradient du critère. Là est la principale difficulté de la méthode, du moins dans le cas général.
Dans le cas que nous considérons ici (systèmes linéaires mono-entrée/mono-sortie représentés par des relations de récurrence), ce problème prend une forme relativement simple.
Reprenons notre exemple (eq3.4). Le gradient est :

$$C_p = \sum_{i=1}^{N} \left(\frac{\partial c(e_n, u_n)}{\partial e_n} \frac{\partial e_n}{\partial p} + \frac{\partial c(e_n, u_n)}{\partial u_n} \frac{\partial u_n}{\partial p} \right) = \sum_{i=1}^{N} \left(2 e_n \frac{\partial e_n}{\partial p} + 2 u_n \frac{\partial u_n}{\partial p} \right) \tag{3.6}$$

Les vecteurs $\frac{\partial e_n}{\partial p}$ et $\frac{\partial u_n}{\partial p}$ sont appelés : coefficients de sensibilité de e et de u par rapport aux coefficients du correcteur. Il faut les calculer.



3.1.3 Calcul des coefficients de sensibilité

Pour calculer les coefficients de sensibilité, nous sommes obligés de résoudre les équations de sensibilité qui se déduisent des équations (3.1.1) par différenciation par rapport à p.

Posons : $y_a = \frac{\partial x}{\partial a}; y_b = \frac{\partial x}{\partial b}; v_a = \frac{\partial u}{\partial a}; v_b = \frac{\partial u}{\partial b}$

soit :

$$y_{an+1} + a_1 y_{an} + a_2 y_{an-1} = b_1 v_{an-1} + b_2 v_{an-2} \quad (3.7)$$

$$v_{an+1} + a v_{an} + u_n = -b y_{an} \quad (3.8)$$

et :

$$y_{bn+1} + a_1 y_{bn} + a_2 y_{bn-1} = b_1 v_{bn-1} + b_2 v_{bn-2} \quad (3.9)$$

$$v_{bn+1} + a v_{bn} = r_n - x_n - b y_{bn} \quad (3.10)$$

Les conditions initiales sont toutes nulles.

On constate que les équations (3.7) et (3.9) sont identiques à celles du procédé (3.1). Leur résolution ne présente donc aucune difficulté, puisque c'est le même programme qui peut être utilisé.

Les équations (3.8) et (3.10) sont des variantes de (3.2) avec des entrées supplémentaires que l'on détermine par (3.1) et (3.2) : donc toutes ces équations, du procédé comme des fonctions de sensibilité, doivent être résolues en parallèle. **Exercice :** Ecrire la procédure de résolution complète en partant de conditions initiales nulles, et $r_n = 1$ pour tout n.

$(a, b) \mapsto (x, y, u, v) \mapsto (c, c_p)$

3.2 Commande en temps discret

Les conditions d'optimalité de la PNL s'appliquent directement à la résolution des problèmes de commande optimale en temps discret.

3.2.1 Enoncé du problème

On considère un système à temps discret dont l'évolution est représentée par une équation d'état :

$$x_{k+1} = f(x_k, u_k),\ x_k \in \mathbb{R}^n,\ u_k \in \mathbb{R}^m,\ k \in \mathbb{N} \tag{3.11}$$

Il n'y a pas de contraintes sur u et x. Le critère à minimiser est de la forme :

$$J = \sum_{k=0}^{N-1} c_k(x_k, u_k) \tag{3.12}$$

Nous supposerons que l'état initial x_0 est donné et que l'état final x_N doit se trouver sur une variété $\mathcal{V}_f$:

$$x_N \in \mathcal{V}_f = \{x \in \mathbb{R}^n : \varphi(x) = 0\} \tag{3.13}$$

avec $x \mapsto \varphi(x) = [\varphi_1(x), \varphi_2(x), \cdots \varphi_p(x)]^T$ et $\varphi_i : \mathbb{R}^n \to \mathbb{R}$ suffisamment différentiable. L'ensemble des équations (3.2.1) définit le problème.

3.2.2 Solution

Pour résoudre ce problème, l'idée est de considérer l'optimisation de J par rapport à l'ensemble des variables x et u. Les équations d'évolution (3.11) et les conditions sur les états initial et final (eq 3.13) constituent les contraintes entre ces variables. On applique alors les résultats de la PNL avec contraintes égalité.

Soit $\mathcal{L}(x, u, \lambda)$ le Lagrangien :

$$\mathcal{L}(x, u, \lambda) = \sum_{k=0}^{N-1} c_k(x_k, u_k) + \sum_{k=0}^{N-1} \lambda_{k+1}^T (x_{k+1} - f(x_k, u_k)) + \mu^T \varphi(x_N) \tag{3.14}$$

La stationnarité de $\mathcal{L}$ par rapport à toutes les variables donne le groupe d'équations suivant :

$$\frac{\partial \mathcal{L}}{\partial u_k} = 0 = \frac{\partial c_k(x_k, u_k)}{\partial u_k} - f_{u_k}^T(x_k, u_k)\lambda_{k+1},\ k = 0, \cdots, N-1 \tag{3.15}$$

$$\frac{\partial \mathcal{L}}{\partial x_k} = 0 = \frac{\partial c_k(x_k, u_k)}{\partial x_k} - f_{x_k}^T(x_k, u_k)\lambda_{k+1} + \lambda_k,\ k = 0, \cdots, N-1 \tag{3.16}$$

$$\frac{\partial \mathcal{L}}{\partial x_N} = 0 = \lambda_N + \varphi_x^T(x_N)\mu \tag{3.17}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_{k+1}} = 0 = x_{k+1} - f(x_k, u_k),\ k = 0, \cdots, N-1 \tag{3.18}$$

Pour résoudre ces équations, il faut pouvoir résoudre l'équation(3.15) en fonction de u_k : soit $u_k = \phi_k(x_k, \lambda_{k+1})$, expression que l'on reporte dans (3.16) et (3.18). On obtient ainsi un système autonome d'équations de récurrence en x et λ de dimension $2n$.

Les conditions aux limites sont données :

- en 0 par la donnée de x_0,
- en N par l'équation (3.17) qui indique que λ_N est combinaison linéaire des gradients des contraintes en x_N ou, si l'on préfère que λ_N est orthogonal à la variété $\mathcal{V}_f$ en x_N(condition de transversalité).

Remarque 42 *Remarquons que l'on aurait pu prendre comme hypothèse que $x_0 \in \mathcal{V}_0$ pour obtenir λ_0 orthogonal à la variété $\mathcal{V}_0$ en x_0.*

Lorsque l'état est fixé, il n'y a pas de condition sur λ_N; lorsque l'état final est libre $\lambda_N = 0$.

Ces conditions de transversalité nous indiquent que nous disposerons toujours de $2n$ conditions aux limites, mais malheureusement elles se répartiront en : n conditions en 0 et n conditions en N.

3.3 Exemple : Problème LQ en temps discret

Un cas particulier très important en pratique est le cas où le système est linéaire et le critère est quadratique : problème LQ

3.3.1 Position du problème :

Soit le système linéaire :

$$x_{k+1} = Ax_k + Bu_k,\ x \in \mathbb{R}^n,\ u \in \mathbb{R}^p \tag{3.19}$$

et le critère quadratique à minimiser :

$$J = \frac{1}{2}\sum_{k=0}^{N-1}(x_k^T Q x_k + u_k^T R u_k) + \frac{1}{2}x_N^T P_N x_N \tag{3.20}$$

$$Q = Q^T \geq 0,\ R = R^T > 0, P_N = P_N^T \geq 0 \tag{3.21}$$

$$x_0 \text{ donné},\ x_N \text{ libre} \tag{3.22}$$

Nous remarquons que la matrice Q peut toujours s'écrire $Q = C^T C$; en effet si Q est symétrique, elle est diagonalisable par une matrice orthogonale $Q = VDV^T$ où $D = diag(\lambda_1, \lambda_2, \cdots, \lambda_r, 0, \cdots, 0)$ 0 et

$$C = diag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \cdots, \sqrt{\lambda_r}, 0, \cdots, 0)V^T$$

convient.

Cette forme est souvent naturelle car on fait en général intervenir les sorties dans le critère : soit $y^T y$ avec $y = Cx$.

Le problème est donc de trouver la suite de commandes qui minimisent le critère J avec les conditions (3.21) et les conditions aux limites (3.22). Nous supposerons qu'il n'y a pas de contraintes sur u et x.

3.3.2 Résolution par la programmation non-linéaire :

Le problème se résout suivant la procédure expliquée dans la section 3.2.

Equations générales :

Nous avons à trouver le minimum d'une fonction de N variables : $u_0, u_1, ..., u_{N-1}$ en présence des contraintes "égalité" formées par l'équation d'évolution (3.19).

Le minimum sera obtenu en utilisant le Lagrangien $\mathcal{L}$ à l'aide des multiplicateurs de Lagrange.

$$\mathcal{L} = J + \sum_{k=0}^{N-1} \lambda_{k+1}^T(-x_{k+1} + Ax_k + Bu_k) \tag{3.23}$$

En écrivant la stationnarité de $\mathcal{L}$ par rapport aux variables u, x et λ, nous obtenons trois groupes d'équations :

$$\frac{\partial \mathcal{L}}{\partial u_k} = 0 = Ru_k + B^T\lambda_{k+1},\ k = 0, N-1 \text{ équation de commande} \tag{3.24}$$
$$\frac{\partial \mathcal{L}}{\partial x_k} = 0 = Qx_k - \lambda_k + A^T\lambda_{k+1},\ k = 0, N-1 \text{ équation adjointe} \tag{3.25}$$
$$\frac{\partial \mathcal{L}}{\partial x_N} = 0 = P_N x_N - \lambda_N \text{ condition finale} \tag{3.26}$$
$$\frac{\partial \mathcal{L}}{\partial \lambda_{k+1}} = 0 = -x_{k+1} + Ax_k + Bu_k,\ k = 0, N-1 \text{ équation d'évolution} \tag{3.27}$$

De (3.24), on tire la commande optimale :

$$\hat{u}_k = -R^{-1}B^T\lambda_{k+1} \tag{3.28}$$

en remplaçant dans l'équation d'évolution, on trouve un système autonome de $2n$ équations de récurrence :

$$Qx_k - \lambda_k + A^T\lambda_{k+1} = 0 \tag{3.29}$$
$$-x_{k+1} + Ax_k - BR^{-1}B^T\lambda_{k+1} = 0 \tag{3.30}$$

ou :

$$\lambda_k = Qx_k + A^T\lambda_{k+1} \tag{3.31}$$
$$x_{k+1} = Ax_k - BR^{-1}B^T\lambda_{k+1} \tag{3.32}$$

Pour le résoudre, il faut connaître $2n$ conditions aux limites. Nous connaissons x_0 soit n conditions. On a n relations à l'extrémité liant λ_N et x_N (3.26).

En revanche, si x_N avait été fixé, il aurait fallu tenir compte des contraintes induites sur les dernières commandes pour atteindre la valeur fixée. Ce résultat n'est pas fortuit. La programmation non linéaire est fondée sur une méthode variationnelle et lorsqu'on étudie les variations possibles de u autour de $\hat{u}$ il faut respecter les contraintes. Lorsque x_e (valeur de l'état à l'extrémité) est fixée, les commandes u doivent respecter la contrainte. Lorsque x_e appartient à une variété de dimension r, u doit conduire à des perturbations Δx_e qui sont sur le plan tangent à la variété en x_e ce qui nous donne $N - r$ conditions sur λ_{N+1}.

Valeurs propres de la relation de récurrence :

Si on écrit la transformée en z de (3.31-3.32) on obtient :

$$(z^{-1}\mathbb{I} - A^T)z\lambda + Qx = 0 \tag{3.33}$$
$$(z\mathbb{I} - \mathbb{A})x + BR^{-1}B^T z\lambda = 0 \tag{3.34}$$

Soit :

$$\begin{bmatrix} z\mathbb{I}-A & BR^{-1}B^T \\ -Q & z^{-1}\mathbb{I}-A^T \end{bmatrix} \begin{bmatrix} x \\ z\lambda \end{bmatrix} = M(z) \begin{bmatrix} x \\ z\lambda \end{bmatrix} = 0 \tag{3.35}$$

Les valeurs propres de l'opérateur de récurrence sont les valeurs de M pour lesquelles le système (3.35) admet une solution non triviale, donc pour lesquelles $\det M(z) = 0$. Or $\det M(z) = \det M^T(z)$ en raison de la symétrie des matrices.

On peut aussi intervertir les blocs, ce qui revient à un changement d'ordre des composantes soit donc :

$$\begin{aligned}
\det M(z) &= \det M^T(z) && (3.36)\\
&= \det \begin{bmatrix} z\mathbb{I}-A^T & -Q \\ BR^{-1}B^T & z^{-1}\mathbb{I}-A \end{bmatrix} && (3.37)\\
&= \det \begin{bmatrix} z^{-1}\mathbb{I}-A & BR^{-1}B^T \\ -Q & z\mathbb{I}-A^T \end{bmatrix} && (3.38)\\
&= \det M(z^{-1}) = 0 && (3.39)
\end{aligned}$$

On constate que $\det M(z) = \det(M(z^{-1})$ donc si z est valeur propre $\frac{1}{z}$ est aussi valeur propre. Il y a autant de valeurs propres stables que de valeurs propres instables.

Solution : équation de Riccati :

La résolution progressive des équations (3.31-3.32) montre que λ peut s'écrire sous la forme :

$$\lambda_k = P_k x_k,\ k = 0, \cdots, N \text{ d'après (3.26)} \tag{3.40}$$

Cherchons alors directement une solution de cette forme.
De (3.28), (3.40) et (3.19), on tire :

$$\hat{u}_k = -R^{-1}B^T\lambda_{k+1} = -R^{-1}B^T P_{k+1}x_{k+1} = -R^{-1}B^T P_{k+1}(Ax_k + B\hat{u}_k) \tag{3.41}$$

soit :

$$\hat{u}_k = -[R + B^T P_{k+1}B]^{-1}P_{k+1}Ax_k = -K_k x_k \tag{3.42}$$

La commande optimale est un retour d'état.
En remplaçant λ_k par sa forme (3.40) et $\hat{u}_k$ par sa forme (3.42) dans l'équation (3.31), on trouve :

$$P_k x_k = A^T P_{k+1}Ax_k - A^T P_{k+1}B[R + B^T P_{k+1}B]^{-1}P_{k+1}Ax_k + Qx_k \tag{3.43}$$

La validité de cette équation quel que soit x_k est assurée si :

$$\begin{aligned}
P_k &= A^T P_{k+1}A - A^T P_{k+1}B[R + B^T P_{k+1}B]^{-1}P_{k+1}A + Q && (3.44)\\
\text{ou } P_k &= A^T P_{k+1}D_k + Q \text{ avec } D_k = A - BK_k && (3.45)\\
\text{ou encore } P_k &= D_k^T P_{k+1}D_k + Q + K_k^T RK_k && (3.46)
\end{aligned}$$

L'équation (3.44) se résout pas à pas à partir de la condition finale P_N, on en tire la commande optimale d'après (3.42).

Chapitre 4

Programmation dynamique

Nous exposerons ici le principe de Bellman (1920-1984) appliqué à des systèmes dynamiques

Enoncé par Bellman dans les années 1950 sous forme assez informelle, ce principe débouche sur une gamme de stratégies de recherche d'optimum appelée "programmation dynamique". Largement utilisée dans les recherche de chemin dans des graphes, dans les algorithmes de tri ou de classification, nous l'exposerons de manière formelle dans le cas des systèmes dynamiques avec un critère d'optimisation de type additif. Nous donnerons quelques exemples d'application directe et quelques problèmes classiques qui s'y ramènent.

4.1 Quelques exemples introductifs

4.1.1 Le principe d'optimalité

"Un chemin optimal a la propriété que quelles que soient les conditions initiales et les commandes appliquées (choix effectués) sur une période initiale, la commande à appliquer sur la période restante doit être optimale pour le problème restant avec comme conditions initiales, l'état résultant de l'application des premières commandes".

4.1.2 Deux exemples d'application du principe d'optimalité

Recherche du chemin minimum dans un graphe

Dans le graphe suivant (fig4.1), on cherche le chemin de longueur minimum pour rejoindre l'ensemble de départ d={A,B,C} à l'ensemble d'arrivée a={F,G,H}

L'application du principe d'optimalité nous conduit au raisonnement suivant :

- Si le chemin optimal passe par D, alors pour rejoindre {a} il faudra à partir de D prendre le chemin le plus court soit DF de longueur minimale $\hat{D}$=2.
- Si le chemin optimal passe par E, alors pour rejoindre {a} il faudra à partir de E prendre le chemin le plus court soit EH de longueur minimale $\hat{E}$=1
- Si le chemin optimal part de A, il passera par D et donc aura comme longueur $\hat{A}$=AD+$\hat{D}$=4
- Si le chemin optimal part de B, il passera par D ou E et le meilleur sera donné par min(BD+$\hat{D}$,BE+$\hat{E}$) et donc aura comme longueur $\hat{B}$=3
- Si le chemin optimal part de C, il passera par E et donc aura comme longueur $\hat{C}$=CE+E=5
- Le chemin optimal sera obtenu en prenant min($\hat{A}$,$\hat{B}$,$\hat{C}$) soit BEH=3

Remarque 43 *On peut noter que la recherche est exhaustive : tous les chemins ont été envisagés. L"avantage de cet algorithme par rapport à un algorithme qui recenserait d'abord tous les chemins pour choisir ensuite le plus petit est le nombre de comparaisons effectuées.*

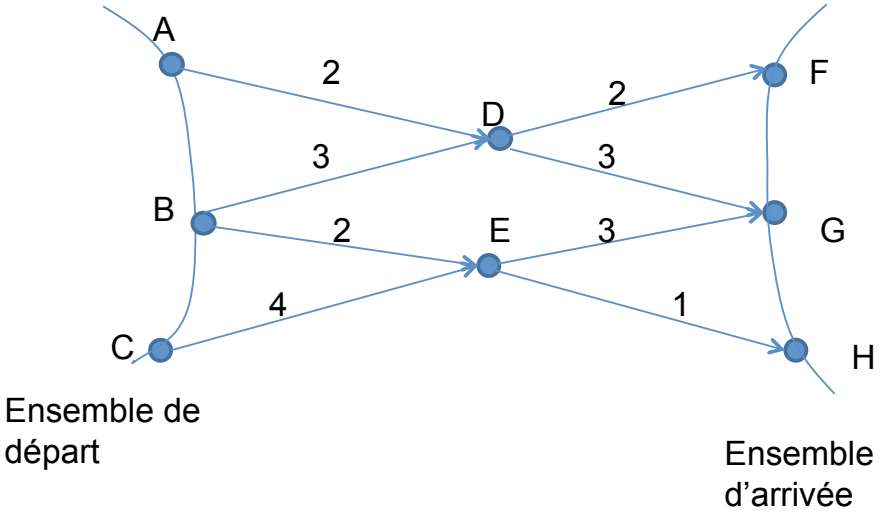


FIGURE 4.1 – Recherche dans un graphe

Commande optimale d'un système à temps discret

Soit le système d'équation

$$x_{k+1} = ax_k + u_n \tag{4.1}$$

On désire aller de l'état initial x_0 à l'état final $x_N = 0$ en minimisant le critère

$$J = \sum_{n=0}^{N-1} \frac{1}{2} u_n^2 \tag{4.2}$$

En appliquant le principe d'optimalité, on déduit :

– Si à l'instant i l'état obtenu par les premières commandes est x_i, alors à partir de cet état il faut chercher la suite de commandes qui minimise

$$J_i = \sum_{n=i}^{N-1} \frac{1}{2} u_n^2 \tag{4.3}$$

et qui transfère le système à $x_N = 0$. Soit $\hat{J}(x_i)$ cette valeur optimale.

– En appliquant le même raisonnement pour résoudre ce nouveau problème et en prenant comme point intermédiaire l'instant $i+1$, on déduit :

$$\hat{J}(x_i) = \min_{u_i}(\frac{1}{2}u_i^2 + \hat{J}(x_{i+1})) \tag{4.4}$$

$$x_{i+1} = ax_i + u_i \tag{4.5}$$

– On voit qu'en appliquant cette récurrence depuis la fin, on pourra obtenir $\hat{J}(x_0)$ et la suite de commandes optimales à condition d'avoir mémorisé toutes les commandes optimales pour tous les états intermédiaires.

Remarque 44 *Si la recherche du coût optimal ne demande que la mémorisation du coût optimal de l'étape précédente dans la recherche, en revanche elle nécessite cette mémorisation pour tout état susceptible d'être atteint à cette étape. L'application de la stratégie optimale en boucle fermée nécessite elle la mémorisation de toutes les commandes optimales à partir de tous les états accessibles. On imagine que cette mémorisation peut être très coûteuse en place mémoire.*

4.2 L'équation d'optimalité

Si on cherche les points communs entre ces deux exemples, on peut conclure que dans les deux cas :

- Le critère est la somme des critères partiels, ce qui a permis de conclure qu'il fallait minimiser le reste pour minimiser le tout,
- pour résoudre le problème à partir du point intermédiaire sur la trajectoire il n'était pas nécessaire de savoir comment ce point avait été atteint, autrement dit, ce point représente un "état", on a donc affaire à un système dynamique.

Avec ces deux propriétés : critère additif (somme des coûts partiels) et système dynamique, le principe d'optimalité se démontre rigoureusement. C'est dans ce cadre que nous donnerons les résultats.

Soit un système dynamique caractérisé par son espace d'état X, l'espace de temps T sur lequel il évolue (ensemble ordonné), son ensemble de fonctions de commande Ω (vérifie l'axiome de concaténation), sa fonction de transition d'état φ :

$$\varphi = X \times T \times T \times \Omega \to X \tag{4.6}$$

$$x(t_2) = \varphi(x_1, t_1, t_2, u) \tag{4.7}$$

qui donne la valeur de l'état à l'instant t_2 lorsque l'état vaut x_1 à l'instant t_1 et que la commande u est appliqué e sur l'intervalle (t_1, t_2), et un critère additif des intervalles de temps.

Lorsque l'état x du système évolue de l'instant t_1 à l'instant t_2 à partir de la condition initiale x_1 sous la commande u, le et l'évolution de l'état x, le coût associé est noté $c(x, t_1, t_2, u)$.

Comme la trajectoire est unique à partir de l'état initial x_1, on peut aussi l'écrire :

$$c(x, t_1, t_2, u) = c(x_1, t_1, t_2, u) \tag{4.8}$$

Ce coût est additif au sens suivant pour tout instant intermédiaire t_i, $t_1 \leq t_i \leq t_2$:

$$c(x_1, t_1, t_2, u) = c(x_1, t_1, t_i, u) + c(x(t_i), t_i, t_2, u) \tag{4.9}$$

$$\text{avec } x(t_i) = \varphi(x_1, t_1, t_i, u) \tag{4.10}$$

4.2.1 Enoncé du problème

Problème 45 *On considère un système dynamique et un coût additif,*
soit :

- $T_1 \subset T$ *un ensemble de temps initiaux,*
- $T_2 \subset T$ *un ensemble de temps finals,*
- $X_1 \subset X$ *un ensemble d'états initiaux,*
- $X_2 \subset X$ *un ensemble d'états finals,*

Déterminer $t_1 \in T_1$, $t_2 \in T_2$, $x_1 \in X_1$, $x_2 \in X_2$, $u \in \Omega$ *tel que* $x_2 = \varphi(x_1, t_1, t_2, u)$ *de manière à ce que* $c(x_1, t_1, t_2, u)$ *soit minimum. La commande sera dite optimale.*

4.2.2 Principle de Bellman

Avec ces conditions, le principe de Bellman se démontre et se traduit par deux théorèmes.

Théorème 46 *Si une commande est optimale entre* t_i *et* t_k*, et si* x^* *est la trajectoire optimale associée alors, pour tout instant intermédiaire* t_j*, cette commande est aussi la commande optimale*

- *entre* t_i *et* t_j *pour le transfert de* $x(t_i)$ *vers* $x(t_j) = x^*(t_j)$

– *entre t_j et t_k pour le transfert $x(t_j) = x^*(t_j)$ vers $x(t_k)$.*

Autrement dit : Si u^* est la commande optimale sur $[t_i, t_k]$ et x^* la trajectoire correspondante, alors les restrictions de u^*, $u^*_{[t_i,t_j]}$ et $u^*_{[t_j,t_k]}$ aux intervalles $[t_i, t_j]$ et $[t_j, t_k]$, sont les commandes optimales pour les deux problèmes :

1. Problème n°1 : partant des conditions initiales, atteindre $x^*(t_j)$ en minimisant le critère $c(x_i, t_i, t_j, u)$,
2. Problème n°2 : partant de $x^*(t_j)$, atteindre les conditions finales en minimisant le critère $c(x^*(t_j), t_j, t_k, u)$.

De ce théorème, en envisageant toutes les valeurs possibles pour l'état à l'instant t_j et en cherchant le minimum, on en déduit le deuxième théorème.

Théorème 47 (équation d'optimalité) *Etant donnés trois instants $t_i \leq t_j \leq t_k$, et les conditions aux limites (x_i, t_i) et (x_k, t_k), on a l'équation suivante :*

$$\min_u c(x_i, t_i, t_k, u) = \min_{u_{[t_i,t_j]}} [c(x_i, t_i, t_j, u_{[t_i,t_j]}) + \min_{u_{[t_j,t_k]}} c(x_j, t_j, t_k, u_{[t_j,t_k]})] \quad (4.11)$$

$$avec\ x_j = \varphi(x_i, t_i, t_j, u_{[t_i,t_j]}) \quad (4.12)$$

$$x_k = \varphi(x_j, t_j, t_k, u_{[t_j,t_k]}) \quad (4.13)$$

Autrement dit : Pour minimiser le coût global de t_i à t_k , il faut minimiser ce que coûte le transfert de t_i à t_j plus le coût minimal pour aller de t_j à t_k.

Proof. (théorème 46)

Soit u^* est la commande optimale sur $[t_i, t_k]$, et $x^*(t) = \varphi(x_i, t_i, t, u^*(\cdot))$, la trajectoire optimale.

Soit

– $\bar{u}_{ij}$ une commande qui transfère le système de (t_i, x_i) à $(t_j, x^*(t_j) = x^*_j)$ en minimisant $c(x_i, t_i, t_j, u)$

– $\bar{u}_{jk}$ une commande qui transfère le système de (t_j, x^*_j) à $(t_k, x^*(t_k) = x^*_k)$ en minimisant $c(x^*_j, t_j, t_k, u)$.

La commande $\bar{u}$, concaténation de $\bar{u}_{ij}$ et $\bar{u}_{jk}$ transfère le système de (t_i, x_i) à $(t_k,\ x^*_k)$ en passant par x^*_j.

Par additivité des coûts puis par définition de $\bar{u}_{ij}$ et $\bar{u}_{jk}$, on a :

$$c(x_i, t_i, t_k, \bar{u}) = c(x_i, t_i, t_j, \bar{u}_i) + c(x^*_j, t_j, t_k, \bar{u}_j) \quad (4.14)$$

$$\leq c(x_i, t_i, t_j, u^*|_{[t_i,t_j]}) + c(x^*_j, t_j, t_k, u^*|_{[t_j,t_k]}) = c(x_i, t_i, t_k, u^*|_{[t_i,t_k]}) \quad (4.15)$$

l'inégalité stricte ne peut donc pas avoir lieu. ■

Proof. (théorème 47)

On considère l'ensemble V des commandes $v(x_j)$ pour aller de (t_i, x_i) à (t_k, x_k) en passant par (t_j, x_j) avec $x_j \in \{$ états possibles en $t_j\}$, et telles que v_{ij} et v_{jk}, restrictions de v aux intervalles $[t_i, t_j]$ et $[t_j, t_k]$ soient optimales pour chaque sous problème.

Si x^* est la trajectoire optimale, $x^*_j \in \{$ états possibles en $t_j\}$, et $u^* \in V(x^*_j)$ d'après le théorème (46).

Par définition de $v(x_j)$:

$$c(x_i, t_i, t_k, v(x_j)) = \min_{v_{ij} \,:\, x(t_j)=x_j} c(x_i, t_i, t_j, v_{ij}) + \min_{v_{jk} \,:x(t_k)=x_k} c(x_j, t_j, t_k, v_{jk}) \tag{4.16}$$

$$c(x_i, t_i, t_k, u^*) = \min_{x_j} c(x_i, t_i, t_k, v(x_j)) \tag{4.17}$$

$$= \min_{x_j} \left(\min_{v_{ij} \,:\, x(t_j)=x_j} c(x_i, t_i, t_j, v_{ij}) + \min_{v_{jk} \,:\, x(t_k)=x_k} c(x_j, t_j, t_k, v_{jk}) \right) \tag{4.18}$$

$$= \min_{x_j} \left(\min_{v_{ij}} c(x_i, t_i, t_j, v_{ij}) + \min_{v_{jk} \,:\, x(t_k)=x_k} c(x(t_j), t_j, t_k, v_{jk}) \right) \tag{4.19}$$

$$= \min_{v_{ij}} \left(c(x_i, t_i, t_j, v_{ij}) + \min_{v_{jk} \,:\, x(t_k)=x_k} c(x(t_j), t_j, t_k, v_{jk}) \right) \tag{4.20}$$

■

4.2.3 Mise en oeuvre

Pour résoudre un problème en utilisant la programmation dynamique, il faut :

1. Mettre le problème dans la formulation "système dynamique, critère additif". Il faut donc :
2. Identifier les espaces de commande, d'état,
3. Trouver l'équation d'évolution,
4. Exprimer le critère.
5. Imaginer la suite des t_i de manière à ce que l'équation d'optimalité (eq 47) puisse se résoudre simplement.
6. Initialiser la récurrence en démarrant en t_2.
7. Terminer lorsque $t_i = t_1$.

4.3 Exemples

4.3.1 Systèmes à temps discret

On considère un système à temps discret dont l'évolution est représentée par une équation d'état :

$$x_{k+1} = f(x_k, u_k),\ x \in \mathbb{R}^n,\ u \in \mathbb{R}^m,\ k \in \mathbb{N} \tag{4.21}$$

Il n'y a pas de contraintes sur u et x. Le critère à minimiser est de la forme :

$$J = \sum_{k=0}^{N-1} c_k(x_k, u_k) + \varphi(x_N) \tag{4.22}$$

Nous supposerons que l'état final x_N est libre, l'état initial x_0 est donné.

Les hypothèses pour appliquer la programmation dynamique (problème 45) sont trivialement vérifiées.

Notons :

$$\hat{J}(x, n, N) \text{ ou } \hat{J}_n(x) \ = \min_{u_n, u_{n+1}, \ldots, u_{N-1}} \sum_{k=n}^{N-1} c_k(x_k, u_k) + \varphi(x_N); x_n = x \tag{4.23}$$

Nous appliquerons l'équation d'optimalité (47) sur les instants $n, n+1$ et N.

$$\hat{J}(x,n,N) \; ou \; \hat{J}_n(x) = \min_{u_n}(c_n(x,u_n) + \hat{J}(f(x,u_n),n+1,N) \tag{4.24}$$

$$ou \; \hat{J}_n(x) = \min_{u_n}(c_n(x,u_n) + \hat{J}_{n+1}(f(x,u_n)) \tag{4.25}$$

La résolution de l'équation (4.24) donne la commande optimale en boucle fermée $\hat{u}_{n,N}(x)$. Comme la solution optimale du problème est obtenue pour $n = 0$, soit

$$\hat{J} = \hat{J}(x_0, 0, N) = \hat{J}_0(x_0), \tag{4.26}$$

l'équation 4.24 est résolue en sens rétrograde de $n = N - 1$ jusque $n = 0$ avec la condition initiale :

$$\hat{J}(x,N,N) \; ou \; \hat{J}_N(x) = \varphi(x) \tag{4.27}$$

On dispose donc d'un algorithme permettant de calculer la solution optimale.

4.3.2 Chemin dans un graphe

Lorsque l'espace d'état contient un nombre fini d'éléments, la dynamique peut être représentée par un graphe. Chaque sommet i est un état, les arcs donnent la fonction de transition d'état. Les arcs sont valués par le coût de leur franchissement : c_{ij} est le coût pour franchir l'arc (i, j). La commande est le chemin choisi pour aller du sommet origine à l'extrémité. Pour introduire le coût terminal, il suffit d'introduire un état fictif t et le coût c_{it} lorsque i est l'état d'arrivée. On appellera S_k l'ensemble des états accessibles à l'étape k.

On supposera qu'il n'existe pas de cycle à coût négatif, le nombre maximum de sommets par lesquels passe le chemin minimum est constitué au plus de N sommets. En posant $c_{ii} = 0$, on pourra toujours considérer qu'il y a N étapes.

L'algorithme de recherche du chemin minimum est alors :

Pour k allant de $N - 1$ à 0, faire :

$$\hat{J}(i,k) = \min_{j \in S_{k+1}} (c_{ij} + \hat{J}(j,k+1)), \; \forall i \in S_k \tag{4.28}$$

$$\hat{J}(i,N) = c_{it}, \; \forall i \in S_N \text{ (Initialisation)} \tag{4.29}$$

$\hat{J}(i, 0)$ est le chemin minimum pour aller de i à l'espace d'arrivée.

Exemple 48 *Déterminer le plus court chemin joignant A à K de la figure 4.2.*

Exemple 49 *Déterminer le plus court chemin joignant x_1 à x_6 de la figure 4.3.*

4.4 Cas stochastique

La programmation dynamique peut aussi se formaliser dans le cas stochastique.

4.4.1 Enoncé du problème

Soit le système

$$x_{k+1} = f(x_k, u_k, w_k); x \in \mathbb{R}^n, \; u \in \mathbb{R}^m, \; k \in \mathbb{N} \tag{4.30}$$

Les w_k sont des variables aléatoires indépendantes (leur loi de probabilité peut dépendre de x_k et u_k).

$$P(w_k \mid x_k, u_k) \tag{4.31}$$

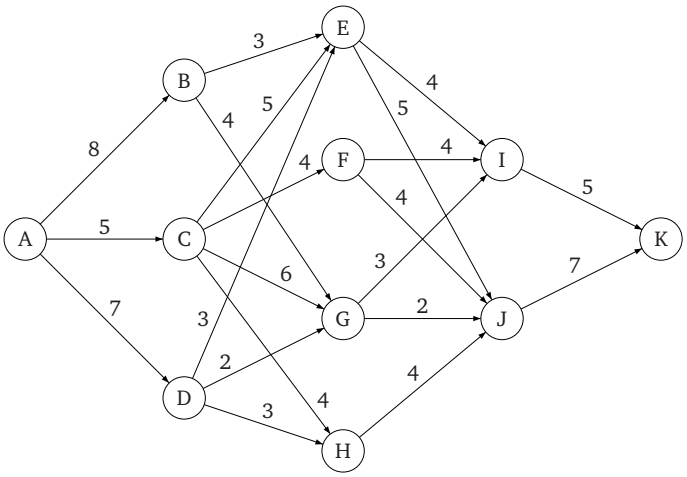


FIGURE 4.2 – Graphe 1

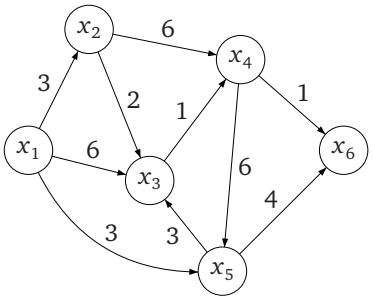


FIGURE 4.3 – Graphe 2

Les commandes sont contraintes $u_k \in U_k$.
Le coût est additif :

$$J = \mathop{\mathbb{E}}_{w_0,w_1,..,w_{N-1}} \{\sum_{k=0}^{N-1} c_k(x_k, u_k, w_k) + \varphi(x_N)\} \tag{4.32}$$

On fera l'hypothèse que la loi de probabilité sur les coûts induite par les v.a. w_k admet une moyenne finie quelle que soit la loi de commande $u_k = u_k(x_k)$.
Le problème consiste à trouver la loi de commande optimale u^* qui minimisera J soit $J^*(x_0)$.

4.4.2 Algorithme

$$J_N^*(x) = \varphi(x) \tag{4.33}$$
$$J_k^*(x) = \min_{u_k} \mathop{\mathbb{E}}_{w_k} (c_k(x_k, u_k, w_k) + J_{k+1}^*(f(x_k, u_k, w_k)) \tag{4.34}$$
$$J^*(x_0) = J_0^*(x_0) \tag{4.35}$$

4.5 Retour sur la commande LQ en temps discret

4.5.1 Problème LQ sur horizon fini

On reprend le problème LQ en temps discret posé à la section 3.3.1. Nous allons maintenant utiliser la méthode générale exposée en 4.3.1
Posons

$$J_p = \frac{1}{2} \sum_{k=p}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + \frac{1}{2} x_N^T P_N x_N. \tag{4.36}$$

Le problème est de minimiser J_0 avec une dynamique déterminée par :

$$x_{k+1} = Ax_k + Bu_k,\ k = 0, 1, 2, \cdots$$

Posons

$$\hat{J}_p(x) = \min_{u_p,u_{p+1},...,u_{N-1}} J_p \tag{4.37}$$
$$x \text{ est l'état à l'instant } p \tag{4.38}$$

L'algorithme de programmation dynamique est appliqué :
Initialisation :

$$\hat{J}_N(x) = \frac{1}{2} x_N^T P_N x_N \tag{4.39}$$

Pour $k = N-1, \cdots, 0$ **:**

$$\hat{J}_k(x) = \min_{u_k} \{\frac{1}{2}(x_k^T Q x_k + u_k^T R u_k) + \hat{J}_{k+1}(Ax_k + Bu_k)\} \tag{4.40}$$
$$\hat{J} = \hat{J}_0(x_0) \tag{4.41}$$

Effectuons le calcul de $\hat{J}_{N-1}(x)$ en appliquant l'équation (4.40) et en calculant explicitement le minimum, ce qui est possible dans le cas sans contrainte où le minimum est obtenu en annulant le gradient. On obtient une équation linéaire en u puisqu'on dérive une forme quadratique. Les résultats sont les suivants :

$$\frac{\partial J_{N-1}(x_{N-1}, u_{N-1})}{\partial u_{N-1}} = 0 \tag{4.42}$$

$$\text{soit } Ru_{N-1} + B^T P_N(Ax_{N-1} + Bu_{N-1}) = 0 \tag{4.43}$$

$$R + B^T P_N B > 0 \; car \; R > 0 \tag{4.44}$$

d'où le résultat :

$$\hat{u}_{N-1} = -[R + B^T P_N B]^{-1} B^T P_N A x_{N-1} = -K_{N-1} x_{N-1} \tag{4.45}$$

$$\hat{J}_{N-1}(x) = \frac{1}{2} x^T P_{N-1} x \tag{4.46}$$

$$P_{N-1} = A^T P_N A + Q - A^T P_N B[R + B^T P_N B]^{-1} B^T P_N A \tag{4.47}$$

Nous supposons x_N libre, si x_N est fixé, les dernières commandes sont imposées et leur nombre dépend de l'indice de commandabilité du système.

Prenons comme hypothèse de récurrence :

$$\hat{J}_k(x) = \frac{1}{2} x^T P_k x \tag{4.48}$$

En appliquant la même démarche que précédemment, nous obtiendrons de manière évidente les mêmes résultats en substituant k à N dans les formules (4.5.1), soit :

$$\hat{u}_k = -[R + B^T P_{k+1} B]^{-1} B^T P_{k+1} A x_k = -K_k x_k \tag{4.49}$$

$$\hat{J}_k(x) = \frac{1}{2} x^T P_k x \tag{4.50}$$

$$P_k = A^T P_{k+1} A + Q - A^T P_{k+1} B[R + B^T P_{k+1} B]^{-1} B^T P_{k+1} A \tag{4.51}$$

L'identité entre les équations ((3.44) et (4.51) montre que les deux approches donnent, heureusement, le même résultat.

Remarque 50 *L'équation (4.51) donne la suite des P_k à partir de P_N dans le sens rétrograde. Ceci est logique : ce n'est pas l'indice k (combien de commandes ont été appliquées depuis le début) qui a une importance, mais $N-k$, c'est à dire combien de commandes il reste à effectuer. Il est alors préférable d'utiliser la notation rétrograde : $l = N-k$. L'équation (4.51) s'écrit alors :*

Pour $l = 1, \cdots, N$ **:**

$$S_l = A^T S_{l-1} A + Q - A^T S_{l-1} B[R + B^T S_{l-1} B]^{-1} B^T S_{l-1} A \tag{4.52}$$

$$S_0 = P_N \tag{4.53}$$

$$\hat{u}_l = -[R + B^T S_{l-1} B]^{-1} B^T S_{l-1} A x \tag{4.54}$$

$$D_l = A - B[R + B^T S_{l-1} B]^{-1} B^T S_{l-1} A \tag{4.55}$$

Le critère minimisé est :

$$J_N = \frac{1}{2} x_N^T S_0 x_N + \frac{1}{2} \sum_0^{N-1} (x_l^T Q x_l + u_l^T R u_l) \tag{4.56}$$

la valeur du critère optimal est :

$$J_N(x_0, S_0) = \frac{1}{2} x_0^T S_N(S_0) x_0 \tag{4.57}$$

Remarque 51 *Nous avons une structure de retour d'état (non constant), donc une boucle fermée, mais qui nécessite la connaissance de l'état. La matrice d'état, à paramètres variables est $D_k = A - BK_k$*

4.5.2 Comportement asymptotique :

Théorème

Quand les matrices A, B, Q et R sont constantes et quand N tend vers l'infini, S_l tend vers une solution stationnaire qui satisfait l'équation algébrique de Riccati (4.58).

$$S = A^T SA + Q - A^T SB[R + B^T SB]^{-1} B^T SA \tag{4.58}$$

Théorème 52 *Soit le système (A, B, C), et le critère $J = \frac{1}{2}\sum_0^\infty (x_k^T Q x_k + u_k^T R u_k)$ avec les matrices $Q = C^T C \geq 0$ et $R = R^T > 0$.*

Si (A, B) est commandable et (A, C) est observable alors :

1. *il existe une matrice $S = S^T > 0$ telle que $\lim_{l\to\infty} S_l = S, \forall S_0 \geq 0$,*
2. *S est l'unique solution > 0 de l'équation algébrique de Riccati (4.58),*
3. *la commande optimale est un retour d'état de gain K,*

$$\hat{u}_k = -Kx_k \tag{4.59}$$

où $K = [R + B^T SB]^{-1} B^T SA$

4. *le système optimal vérifie l'équation*

$$x_{k+1} = Dx_k \tag{4.60}$$

avec $D = A - BK$

5. *ce système est stable,*
6. *les valeurs propres du système optimal sont les n racines stables (module < 1) de (3.39).*

Démonstration

Convergence de (4.51) vers une solution stationnaire

Prenons $S_0 = 0$.

On minimise $\frac{1}{2}\sum_0^{N-1} (x_l^T Q x_l + u_l^T R u_l)$, le résultat optimal est $\frac{1}{2} x_0^T S_N(0) x_0$.

Montrons que

$$x_0^T S_N(0) x_0 \leq x_0^T S_{N+1}(0) x_0, \ \forall N, x_0.$$

Soit $\hat{u}_{0,N+1}, \hat{u}_{1,N+1}, \cdots, \hat{u}_{N,N+1}$ la commande optimale en $N+1$ étapes,

$$\begin{aligned}
\frac{1}{2} x_0^T S_{N+1}(0) x_0 =& \hat{J}_{N+1}(x_0, 0) && (4.61)\\
=& \frac{1}{2}\sum_{l=0}^{N} (\hat{x}_{l,N+1}^T Q \hat{x}_{l,N+1} + \hat{u}_{l,N+1}^T R \hat{u}_{l,N+1}) && (4.62)\\
\geq& \frac{1}{2}\sum_{l=0}^{N-1} (\hat{x}_{l,N+1}^T Q \hat{x}_{l,N+1} + \hat{u}_{l,N+1}^T R \hat{u}_{l,N+1}) && (4.63)\\
\geq& \min_{u_0,\ldots,u_{N-1}} \frac{1}{2}\sum_{l=0}^{N-1} (x_l^T Q x_l + u_l^T R u_l) = \hat{J}_N(x_0, 0) && (4.64)\\
=& x_0^T S_N(0) x_0 && (4.65)
\end{aligned}$$

La suite $k \to x_0^T S_k(0) x_0$ est donc non décroissante quel que soit x_0. La quantité est bornée supérieurement par le coût obtenu en appliquant les commandes qui ramènent le système à

zéro (système commandable) et zéro ensuite, donc cette suite converge $\forall x_0$. Ceci entraîne la convergence de tous les termes de la matrice, donc de $S_k(0)$ vers une matrice constante $S(0)$ symétrique non négative.

En passant à la limite dans (4.53) on obtient (4.58).

Stabilité de $D = A - BK$

Montrons que la fonction $x \to V(x) = x^T Sx$ est une fonction de Lyapounov du système.
Calculons $\Delta V_k = V(x_{k+1}) - V(x_k)$.
L'équation (4.58) peut aussi s'écrire :

$$\begin{aligned} S &= D^T SD + Q + K^T RK && (4.66)\\ \text{avec } D &= A - BK && (4.67)\\ \text{et } K &= [R + B^T SB]^{-1} B^T SA && (4.68) \end{aligned}$$

D'où :

$$\Delta V_k = x_{k+1}^T Sx_{k+1} - x_k^T Sx_k = x_k^T (D^T SD - S)x_k = -x_k^T (Q + K^T RK)x_k \leq 0 \quad (4.69)$$

En général on n'a pas $\Delta V_k < 0$, on ne peut donc pas conclure directement.
En itérant l'équation (4.69), on obtient :

$$x_{k+1}^T Sx_{k+1} = x_0^T Sx_0 - \sum_{i=0}^{k} x_i^T (Q + K^T RK)x_i \quad (4.70)$$

Soit

$$x_{k+1}^T Sx_{k+1} + \sum_{i=0}^{k} x_i^T (Q + K^T RK)x_i = x_0^T Sx_0 \quad (4.71)$$

et on sait que $x_{k+1}^T Sx_{k+1} \geq 0$ donc la série $\sum_{i=0}^{k} x_i^T (Q + K^T RK)x_i$ converge et donc

$$x_k^T (Q + K^T RK)x_k \to 0$$

quand $k \to 0$ ce qui implique

$$Cx_k \to 0 \text{ et } Kx_k \to 0 \text{ quand } k \to 0. \quad (4.72)$$

L'hypothèse d'observabilité de (A, C) va nous permettre de conclure que $x_k \to 0$.
En effet :

$$\begin{aligned} x_k &= Idx_k && (4.73)\\ x_{k+1} &= Ax_k - BKx_k && (4.74)\\ x_{k+2} &= A^2 x_k - ABKx_k - BKx_{k+1} && (4.75)\\ &\cdots && (4.76)\\ x_{k+n} &= A^n x_k - \cdot && (4.77) \end{aligned}$$

En multipliant toutes les lignes à gauche par C on déduit :

$$\begin{aligned} &Cx_k = Cx_k && (4.78)\\ &Cx_{k+1} + CBKx_k = CAx_k && (4.79)\\ &Cx_{k+2} + CABKx_k + CBKx_{k+1} = CA^2 x_k && (4.80)\\ &\cdots && (4.81)\\ &Cx_{k+n} + C\cdots = CA^n x_k && (4.82) \end{aligned}$$

Les termes à gauche du signe = tendent vers zéro (4.72), donc $\mathcal{O}(A, C)x_k \to 0$ donc x_k tend vers zéro quand $k \to \infty$.

S défini positif

Il est évident que $S \geq 0$ puisque le critère ne peut pas être négatif.
Supposons qu'il existe $x_0 \neq 0$ tel que $x_0^T S x_0 = 0$ alors d'après la formule (4.71)

$$x_k^T (Q + K^T R K) x_k = 0, \ \forall k$$

donc $Cx_k = 0, \ \forall k$ d'où $x_k = 0$ et $x_0 = 0$ par l'observabilité de (A, C), ce qui contredit l'hypothèse.

Conditions initiales arbitraires $S_0 \neq 0$

Soit $S_k(S_0)$ la solution de (4.53) et $K(S_0)$ le gain optimal correspondant.
On a

$$x_0^T S_k(0) x_0 \leq x_0^T S_k(S_0) x_0, \ \forall x_0.$$

Si on applique la commande $K = K(0)$ au problème avec $S_0 \neq 0$ on doit trouver un résultat moins bon qu'avec la commande optimale $K(S_0)$ de ce problème, soit :

$$x_0^T S_k(S_0) x_0 \leq x_0^T D^{kT} S_0 D^k x_0 + x_0^T (\sum_{i=0}^{k-1} D^{iT} (Q + K^T R K) D^i) x_0 \tag{4.83}$$

$$= x_0^T D^{kT} S_0 D^k x_0 + x_0^T (\sum_{i=0}^{k-1} D^{iT} (S - D^T S D) D^i) x_0 \tag{4.84}$$

d'après (4.66). Dans la somme précédente, les termes se compensent 2 à 2 et il ne reste que

$$x_0^T D^{kT} S_0 D^k x_0 + x_0^T S x_0 - x_0^T D^{kT} S D^k x_0.$$

Comme le système est asymptotiquement stable, $D^k x_0 \to 0$, on a alors l'inégalité :

$$x_0^T S_k(0) x_0 \leq x_0^T S_k(S_0) x_0 \leq x_0^T S x_0, \ \forall k, \ x_0 \tag{4.85}$$

Comme $S_k \to S$, on en déduit le résultat.

Unicité de la solution :

Supposons que $\tilde{S}$ soit une autre solution > 0 de l'équation (4.58), alors $S_k(\tilde{S}) = \tilde{S} \ \forall k$ (toute solution de l'équation algébrique vérifie l'équation de récurrence) et $S_k(\tilde{S}) \to S$.

Pôles en boucle fermée

Nous avons démontré dans la section (3.3.2) que la suite des états et des paramètres de Lagrange vérifient, lorsqu'ils sont optimaux, les équations de récurrence LTI ((3.31-3.32)) dont les valeurs propres sont données par (3.39). D'autre part les états optimaux (puisqu'ils sont obtenus par application de la commande optimale) vérifient l'équation de récurrence (4.60) stable. Les valeurs propres du système optimal sont donc les n racines stables (module < 1) de (3.39).

4.5.3 Système bruité :

L'équation (4.86) représente l'évolution du système :

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{4.86}$$

Hypothèses : Les variables aléatoires w_k sont indépendantes et indépendantes des x et u. L'espérance $\mathbb{E}(w_k) = 0$ et les moments d'ordre deux existent.

Le problème est de minimiser :

$$\mathop{\mathbb{E}}_{w}\{J = \frac{1}{2}\sum_{k=0}^{N-1}(x_k^T Q x_k + u_k^T R u_k) + \frac{1}{2}x_N^T P_N x_N\}$$

On applique l'algorithme de la section 4.4.2.

$$J_N^*(x) = \frac{1}{2}x^T P_N x \tag{4.87}$$

$$J_k^*(x) = \min_{u_k}\mathop{\mathbb{E}}_{w_k}\{\frac{1}{2}(x_k^T Q x_k + u_k^T R u_k) + J_{k+1}^*(Ax + Bu_k + w_k)\} \tag{4.88}$$

$$J^*(x_0) = J_0^*(x_0) \tag{4.89}$$

Le calcul de $J_{N-1}^*(x)$ est le suivant :

$$J_{N-1}^*(x) = \min_{u_{N-1}}\mathop{\mathbb{E}}_{w_k}\{\frac{1}{2}(x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1}) + \frac{1}{2}x_N^T P_N x_N\} \tag{4.90}$$

$$= \min_{u_{N-1}}\mathop{\mathbb{E}}_{w_k}\{\frac{1}{2}(x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1}) + \frac{1}{2}(Ax_{N-1} + Bu_{N-1} + w_{N-1})^T P_N (Ax_{N-1} + Bu_{N-1} + w_{N-1})\} \tag{4.91}$$

en utilisant le fait que $E(w_{N-1}) = 0$, il reste :

$$J_{N-1}^*(x) = \min_{u_{N-1}}(\frac{1}{2}(x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1}) + \frac{1}{2}(Ax_{N-1} + Bu_{N-1})^T P_N (Ax_{N-1} + Bu_{N-1})) + \frac{1}{2}E\{w_{N-1}^T P_N w_{N-1}\} \tag{4.92}$$

On trouve donc le même résultat que dans le cas déterministe pour le calcul de u_{N-1}^*. La seule différence est dans la valeur du critère optimal auquel se rajoute $\frac{1}{2}E\{w_{N-1}P_N w_{N-1}\}$. On en déduit par récurrence que les résultats trouvés dans le cas déterministe sont inchangés. La valeur du critère optimal est maintenant :

$$J_0^*(x_0) = \frac{1}{2}x_0^T P_0 x_0 + \frac{1}{2}\sum_{0}^{N-1}E\{w_k^T P_{k+1} w_k\} \tag{4.93}$$

4.6 Cas continu : équation d'Hamilton-Jacobi-Bellman

4.6.1 Problème

On considère le système :

$$\dot{x} = f(x, u, t) \tag{4.94}$$

$$x(t_0) = x_0 \tag{4.95}$$

trouver $\hat{u}_{[t_0,t_f]}$ qui minimise le critère

$$J(x_0, t_0, t_f, u) = \int_{t_0}^{t_f} L(x, u, t)dt + \phi(x(t_f)) \tag{4.96}$$

Dans ce qui suit, l'ensemble U est un ouvert de $\mathbb{R}^m$, les fonctions f, L et ϕ sont supposées de classe C^1.

On suppose qu'il existe une commande optimale $\hat{u}(.)$ qui réalise le minimum de J pour tout état initial x_0 et tout instant de départ t_0.

4.6.2 Utilisation de la programmation dynamique

Posons : $\hat{J}(x_0, t_0, t_f) = \inf_{u_{[t_0,t_f]}}(J(x_0, t_0, t_f, u))$.

Hypothèse : On supposera le problème sans contrainte et toutes les conditions de continuité et dérivabilité nécessaires satisfaites. En particulier $\hat{J}$ est supposé être C^1.

Notons $\hat{J}(x(t), t, t_f)$ le coût optimal obtenu en partant à l'instant t de la position $x(t)$ en minimisant le critère sur l'intervalle de temps $[t, t_f]$.

$\hat{J}(x(t), t, t_f)$ est appelée *fonction de Bellman* et vérifie l'équation du même nom.

En appliquant la méthode de programmation dynamique, on a :

$$\hat{J}(x_0, t_0, t_f) = \inf_{u_{[t_0,t]}} (\int_{t_0}^{t} L(x, u, \tau)d\tau + \hat{J}(x(t), t, t_f)); \forall t, t_0 \leq t \leq t_f \tag{4.97}$$

où $x(t)$ est l'état du système en t, obtenu à partir de x_0 par la commande $u_{[t_0,t]}$.

Pour trouver une équation fonctionnelle sur $\hat{J}$, on réitère la programmation dynamique sur l'intervalle $[t, t_f]$ en choisissant comme instant intermédiaire $t + \Delta t$.

On obtient alors :

$$\hat{J}(x(t), t, t_f) = \inf_{u_{[t,t+\Delta t]}} (\int_{t}^{t+\Delta t} L(x, u, \tau)d\tau + \hat{J}(x(t + \Delta t), t + \Delta t, t_f)) \tag{4.98}$$

On peut calculer le terme de droite au premier ordre en Δt en évaluant l'intégrale par la formule de la moyenne.

$$\int_{t}^{t+\Delta t} L(x, u, \tau)d\tau = L(x(t), u(t), t)\Delta t + o(\Delta t) \tag{4.99}$$

et développant $\hat{J}(\ x(t + \Delta t), t + \Delta t, t_f)$ au premier ordre le long de la trajectoire optimale :

$$\begin{aligned}\hat{J}(x(t + \Delta t), t + \Delta t, t_f) = \ & \hat{J}(x(t), t, t_f) + \\ & \frac{\partial \hat{J}}{\partial x}^T (x(t), t, t_f) f(x(t), u(t), t)\Delta t + \frac{\partial \hat{J}}{\partial t}(x(t), t, t_f)\Delta t + o(\Delta t)\end{aligned} \tag{4.100}$$

L' équation (4.100) suppose la continuité de la commande u sur l'intervalle $[t, t+\Delta t]$. Nous ne pourrons donc obtenir par cette méthode que des conditions pour qu'une commande u continue soit optimale. Cette hypothèse sera levée dans le théorème de Pontriaguine.

$\hat{J}$ et $\hat{J}_t$ qui apparaissent dans la relation (4.100), ne dépendent pas de u. Ils sortent donc de l'opérateur $\inf_{u_{[t,t+\Delta t]}}$ dans (4.98). $\hat{J}$ se trouve alors des deux côtés du signe égal dans l'équation (4.98) et se simplifie. Dans cette équation Δt se met alors en facteur et se simplifie.

Ces simplification faites, en faisant tendre Δt vers 0, il reste l' équation suivante :

$$\frac{\partial \hat{J}}{\partial t}(x, t, t_f) = -\inf_{u(t)}[L(x, u(t), t) + \frac{\partial \hat{J}}{\partial x}^T (x, t, t_f) f(x, u(t), t)] \tag{4.101}$$

Soit $u^*(\frac{\partial \hat{J}}{\partial x}^T (x, t, t_f), x, t)$ la valeur de u qui minimise l'expression précédente. Alors en remplaçant $u(t)$ par $u^*(\frac{\partial \hat{J}}{\partial x}^T (x, t, t_f), x, t)$, on obtient l'équation aux dérivées partielles sur $\hat{J}$ connue sous le nom "Hamilton-Jacobi-Bellman" (4.102).

$$\begin{aligned}\frac{\partial \hat{J}}{\partial t}(x, t, t_f) &= -[L(x, u^*(x, t), t) + \frac{\partial \hat{J}}{\partial x}^T (x, t, t_f) f(x, u^*(\frac{\partial \hat{J}}{\partial x}^T (x, t, t_f), x, t), t)] && (4.102)\\ avec\ \hat{J}(x, t_f, t_f) &= \phi(x) && (4.103)\end{aligned}$$

La résolution de cette EDP donne $\hat{J}$, $\frac{\partial \hat{J}}{\partial t}, \frac{\partial \hat{J}}{\partial x}$. En remplaçant $\frac{\partial \hat{J}}{\partial x}^T(x,t,t_f)$ par sa valeur dans l'expression de $u^*(\frac{\partial \hat{J}}{\partial x}^T(x,t,t_f),x,t)$, on obtient la commande optimale (en boucle fermée) $\hat{u}(x,t)$. La résolution de l'équation d'évolution (4.6.1) avec cette commande donne la trajectoire optimale.

Théorème :

Soit le système (4.6.1) et le critère (4.96), les fonctions f, L et ϕ sont continûment différentiables.
On pose :

$$H(x,u,\lambda,t) = L(x,u,t) + \lambda^T f(x,u,t)$$

l'Hamiltonien du système, on suppose que H admet un minimum en fonction de u (x, λ, t fixés) pour $u = \bar{u}(x,\lambda,t)$ et que $\bar{u}$ est continûment différentiable.

Si $\hat{J}(x(t),t,t_f)$ est solution de l'équation HJB avec $\hat{J}(x,t_f,t_f) = \phi(x)$, alors $\hat{J}$ est la valeur optimale du critère et la commande optimale est donnée par

$$\hat{u}(t) = \bar{u}(x(t), \frac{\partial \hat{J}(x(t),t,t_f)}{\partial x}, t)$$

Réciproquement si $f, L, \phi, \hat{J}, \hat{u}$, sont continûment différentiables et sont optimaux alors HJB est satisfaite.

4.7 Exemple : Problème LQ en temps continu

4.7.1 Formulation du problème

Soit le système linéaire :

$$\dot{x} = Ax + Bu,\ x \in \mathbb{R}^n,\ u \in \mathbb{R}^m \tag{4.104}$$

On cherche la commande qui minimise le critère quadratique :

$$V(x_0,t_0,t_f,u) = \frac{1}{2}\int_{t_0}^{t_f}[x^T(t)Qx(t) + u^T(t)Ru(t)]dt + \frac{1}{2}x(t_f)^T P_f x(t_f) \tag{4.105}$$

où Q est une matrice définie non négative de rang p, $R = R^T > 0$, $P_f = P_f^T \geq 0$, $x(T)$ est libre, il n'y a pas de contraintes sur u.

On écrira $Q = C^T C$, et on supposera (A,B) commandable et (A,C) observable.

4.7.2 Résolution par les équations HJB :

Notons :

$$V(x,t,t_f,u) = \frac{1}{2}\int_{t}^{t_f}[x^T(\tau)Qx(\tau) + u^\tau(\tau)Ru(\tau)]d\tau + \frac{1}{2}x(t_f)^T P_f x(t_f) \tag{4.106}$$

Comme le problème est autonome (t n'intervient pas explicitement dans la dynamique et dans le critère), la commande optimale

$$\hat{u}(x, \frac{\partial \hat{V}}{\partial x})$$

est donnée par le minimum de H par rapport à u, soit

$$\hat{u} = -R^{-1}B^T\frac{\partial \hat{V}}{\partial x} \tag{4.107}$$

Conjecture 53 *Comme* $\hat{V}(x,t_f,t_f) = \frac{1}{2}x(t_f)^T P_f x(t_f)$, *on peut conjecturer que* $\hat{V}(x,t,t_f)$ *est une forme quadratique de* x *:* $\hat{V}(x,t,t_f) = \frac{1}{2}x^T P(t)x$

$\hat{V}$ vérifie l'équation aux dérivées partielles d'HJB soit :

$$\frac{\partial \hat{V}}{\partial t} = -[\frac{1}{2}(x^T Q x + \hat{u}^T R \hat{u}) + \frac{\partial \hat{V}}{\partial x}^T (Ax + B\hat{u})] \tag{4.108}$$

En utilisant les relations :

$$\hat{u} = -R^{-1}B^T \frac{\partial \hat{V}}{\partial x},$$

$$\frac{\partial \hat{V}}{\partial t} = \frac{1}{2}x^T \dot{P}(t)x,$$

$$\frac{\partial \hat{V}}{\partial x} = Px,$$

et en utilisant le fait que

$$x^T PAx = (x^T PAx)^T = x^T A^T Px,$$

on obtient finalement à l'équation différentielle de Riccati indépendante de x :

$$\dot{P} = -PA - A^T P + PBR^{-1}B^T P - Q \tag{4.109}$$

$$P(t_f) = P_f \tag{4.110}$$

L'intégration de ce système détermine donc la commande optimale à appliquer
Résumons les résultats obtenus.

$$\left| \begin{array}{c} \dot{P} = -PA - A^T P + PR^{-1}B^T P - Q \\ P(t_f) = P_f \\ \hat{u}(t) = -R^{-1}B^T P(t)x(t) = -K(t)x(t) \\ \hat{V}(x,t,t_f) = \frac{1}{2}x^T P(t)x \end{array} \right| \tag{4.111}$$

La valeur du critère est positive quel que soit x, t donc $P > 0$

Chapitre 5

Principe du minimum (Pontriaguine)

Le plus vieux problème d'optimisation dont l'histoire nous a laissé une trace, remonte à l'antiquité. Il s'agissait de déterminer le plus court chemin joignant deux points. Une réponse intuitive souvent confondue avec celle du temps minimal, est tirée de l'expérience quotidienne de nos déplacements. Un bout de ficelle et quelques considérations géométriques confirment que nos déplacements, de préférence en ligne droite, ne sont pas toujours dénués de fondements. Vient ensuite, avec la construction de Carthage en 850 avant J-C, le problème posé par la reine Didon qui demanda pour la construction de sa ville "autant de terre que pouvait en contenir la peau d'un taureau". Pour résoudre ce problème, la Reine Didon découpa la peau en minces lanières qui, mises bout à bout, pouvaient entourer l'espace d'une ville. La solution de ce problème de la recherche de la plus grande aire obtenue à partir d'une courbe fermée était connue des grecs pour être le cercle.

Ce n'est qu'à la fin du $XVII^{ième}$ siècle, avec le Brachystochrone de Jean Bernoulli que le premier problème d'optimisation fait intervenir une dynamique et pose le problème de la recherche d'un chemin optimal. Considérant deux points A et B et un corps M soumis à son poids, Bernoulli s'interroge sur la trajectoire que doit suivre le corps M pour relier ces deux points en un temps minimal. Ce problème attire l'attention et l'enthousiasme de Leibniz, Newton, L'Hôpital, Tschirnhaus ainsi que de Jacques le frère de Jean Bernoulli. Les solutions apportées par ce groupe vont être à l'origine des fondements du calcul des variations développé par la suite par Euler, Lagrange et Legendre et complété plus tard par Weierstrass. Ces travaux marquent le début de la théorie de la commande optimale des systèmes dynamiques telle que nous l'entendons aujourd'hui.

5.1 Formalisation du problème :

La formulation actuelle d'un problème de commande a pour origine les problèmes de pilotage des missiles rencontrés durant les années 40. Ce problème a intéressé un grand nombre de mathématiciens et d'ingénieurs dont les efforts ont donné naissance à une nouvelle discipline scientifique : la théorie de la commande optimale. En toute généralité, le problème de commande peut être énoncé de la manière suivante :

Problème 54 *On entend par commande et on note $u(.)$ tout élément d'une classe de fonctions définies sur un intervalle $[t_0, t_f]$ et prenant leurs valeurs dans un sous-ensemble U de $\mathbb{R}^m$. A toute commande $u(.)$ est associée un arc ou portion de trajectoire $x(.)$ appelé état, solution de l'équation différentielle :*

$$\dot{x}(t) = f(x(t), u(t), t), \qquad x(t_0) = x_0 \tag{5.1}$$

avec $x(t) \in \mathbb{R}^n$. La fonction f est donnée et x_0 peut être librement choisi dans un ensemble C_0.

Un problème de commande optimale consiste alors à déterminer $u(.)$, x_0, t_f, x_f tels que la fonction critère :

$$J = \phi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t)dt \tag{5.2}$$

soit minimisée sous les contraintes

$$x_0 = x(t_0) \in C_0, x_f = x(t_f) \in C_f$$

Le critère J est spécifié par un problème physique ou non, et représente par exemple une dépense d'énergie, d'argent, de temps, etc. L'instant initial t_0 est donné et l'instant final t_f peut être spécifié ou non. Il peut être également envisagé de considérer une dépendance explicite par rapport au temps t du domaine de commande U ainsi qu'une contrainte sur l'état de type :

$$g(t, x(t)) \geq 0 \tag{5.3}$$

La fonction ϕ représente un coût ou critère terminal à payer à l'arrivée. Lorsque le problème contient à la fois un terme intégral et un terme final, on a un problème de Bolza. Lorsque seul le terme intégral apparaît, c'est un problème de Lagrange et dans l'autre cas, il s'agit d'un problème de Mayer.

5.2 Idée de démonstration du théorème de Pontriaguine

Nous donnerons une idée de la démonstration pour un problème plus simple que celui décrit plus haut et nous indiquerons ultérieurement comment déduire les résultats généraux.

On utilise la formulation de Pontriaguine avec les hypothèses simplificatrices suivantes :
- t n'apparaît pas explicitement dans les fonctions L et f,
- il n'y a pas de critère terminal,
- C_0 et C_f sont des sous variétés différentiables de $\mathbb{R}^n$.

Rappel : D est une variété : $D = \cap_i S_i$ où les S_i sont des hypersurfaces $S_i = \{g_i^{-1}(0)\}$, avec les $g_i : \mathbb{R}^n \to \mathbb{R}$ différentiables. Autrement dit, D est défini par les $x \in \mathbb{R}^n$ tels que $g_i(x) = 0$, pour tout i.

On reformule le problème (54) en introduisant la composante supplémentaire x^0 régit par :

$$\frac{dx^0}{dt} = L(x, u) \tag{5.4}$$

On appelle système complété le système de vecteur d'état $z = (x^0, x)$ régi par :

$$\frac{dz}{dt} = g(z, u) = \begin{bmatrix} L(x, u) \\ f(x, u) \end{bmatrix} \tag{5.5}$$

L'objectif est alors de minimiser $x^0(t_f)$.

On peut faire les remarques suivantes :
- x^0 n'apparaît évidemment pas dans les expressions de L et f .
- z évolue de $z(t_0) = \begin{bmatrix} 0 \\ x_0 \end{bmatrix}$ à $z(t_f) = \begin{bmatrix} z_0(t_f) = J \\ x(t_f) = x_f \end{bmatrix}$.
- la commande optimale sera celle qui donnera $x^0(t_f)$ minimum.

Pour exprimer qu'une commande $\hat{u}$ est optimale on s'assurera que toute commande "voisine" donne un moins bon résultat c'est à dire que $x^0(t_f) \geq \hat{x}^0(t_f)$.

5.2.1 Construction des commandes perturbées :

On construit des commandes "voisines" de $\hat{u}$ en remplaçant $\hat{u}(t)$ par une valeur constante ω sur un petit intervalle de temps. Cette valeur respecte les contraintes possibles sur la commande. On prolonge la commande par sa dernière valeur si l'instant final peut varier.

On envisage donc un ensemble de perturbations par rapport à $\hat{u}$ de la manière suivante : Pour $\varepsilon > 0$, $a \geq 0$, $b \geq 0$ et $\omega \in U$

$$u = \hat{u} + \delta u = u(t, a, b, \tau, \omega) = \begin{cases} \omega \ si \ t \in [\tau - \varepsilon b, \tau]; \\ \hat{u}(t) \ ailleurs \\ \hat{u}(t_f) \ si \ t \in [t_f, t_f + \varepsilon a]; \end{cases}$$

Si $a = 0$, on dit que la perturbation est spatiale.

Si $b = 0$, on dit que la perturbation est temporelle (la commande dure plus ou moins longtemps).

Il faut s'assurer qu'avec ces éléments, on peut fabriquer un ensemble de perturbations suffisamment "riche" pour qu'en jouant sur les paramètres (a, b, τ, ω), les conditions nécessaires que l'on obtiendra soient réalistes (c'est à dire pas trop loin des conditions suffisantes).

5.2.2 Influence des perturbations de commande :

Nous appellerons z l'état obtenu avec la commande u, et $\hat{z}$ l' état obtenu avec la commande optimale.

- Influence d'une perturbation spatiale.

– pour $t < \tau - \varepsilon b$
$z(t) = \hat{z}(t)$ quand $t \in [t_0, \tau - \varepsilon b]$ puisque $\hat{u}(t)$ est appliqué jusqu'à $\tau - \varepsilon b$

– pour $t = \tau$

$$z(\tau) = \hat{z}(\tau) + \delta z(\tau) \tag{5.6}$$
$$= \hat{z}(\tau - \varepsilon b) + \int_{\tau - \varepsilon b}^{\tau} g(\hat{z}(\sigma) + \delta z(\sigma), \omega) d\sigma \tag{5.7}$$
$$= \hat{z}(\tau - \varepsilon b) + \varepsilon b g(\hat{z}(\tau), \omega) + o(\varepsilon) \tag{5.8}$$

$$\hat{z}(\tau) = \hat{z}(\tau - \varepsilon b) + \int_{\tau - \varepsilon b}^{\tau} g(\hat{z}(\sigma), \hat{u}(\sigma)) d\sigma \tag{5.9}$$
$$= \hat{z}(\tau - \varepsilon b) + \varepsilon b g(\hat{z}(\tau), \hat{u}(\tau)) + o(\varepsilon) \tag{5.10}$$

$$\delta z(\tau) = \varepsilon b [g(\hat{z}(\tau), \omega) - g(\hat{z}(\tau), \hat{u}(\tau))] + o(\varepsilon) \tag{5.11}$$

– pour $t \geq \tau$
On peut donc considérer que l'influence d'une perturbation spatiale se traduit par une déviation par rapport à la trajectoire optimale ; il faut trouver comment cette déviation se transporte à l'extrémité à l'instant t_f. δz est régi par l'équation de variation obtenue en développant au premier ordre :

$$\frac{d}{dt}(\hat{z} + \delta z) = g(\hat{z} + \delta z, \hat{u}) = g(\hat{z}, \hat{u}) + G_z(\hat{z}, \hat{u}) \delta z + o(\delta z)$$

d'où :

$$\frac{d\delta z}{dt} = G_z(\hat{z}(t), \hat{u}(t))\delta z \tag{5.12}$$

avec

$$G_z = \begin{bmatrix} \frac{\partial g_0}{\partial z_0} & \frac{\partial g_0}{\partial z_1} & \cdot & \frac{\partial g_0}{\partial z_n} \\ \frac{\partial g_1}{\partial z_0} & \frac{\partial g_1}{\partial z_1} & \cdot & \frac{\partial g_1}{\partial z_n} \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial g_n}{\partial z_0} & \frac{\partial g_n}{\partial z_1} & \cdot & \frac{\partial g_n}{\partial z_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial L}{\partial z_0} & \frac{\partial L}{\partial z_1} & \cdot & \frac{\partial L}{\partial z_n} \\ \frac{\partial f_1}{\partial z_0} & \frac{\partial f_1}{\partial z_1} & \cdot & \frac{\partial f_1}{\partial z_n} \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial f_n}{\partial z_0} & \frac{\partial f_n}{\partial z_1} & \cdot & \frac{\partial f_n}{\partial z_n} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\partial L}{\partial x_1} & \cdot & \frac{\partial L}{\partial x_n} \\ 0 & \frac{\partial f_1}{\partial x_1} & \cdot & \frac{\partial f_1}{\partial x_n} \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \frac{\partial f_n}{\partial x_1} & \cdot & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{5.13}$$

- Influence d'une perturbation temporelle :
on a de manière évidente

$$\delta z(a, 0, t, \omega) = \varepsilon a g(\hat{z}(t_f), \hat{u}(t_f)) + o(\varepsilon)$$

5.2.3 Transport de la déviation de trajectoire

La commande optimale est caractérisée par le fait que toute autre commande conduit à un critère plus grand, donc que pour toutes les commandes de la famille que nous venons de décrire

$$\delta J = \delta z_0(t_f) > 0.$$

La difficulté provient du fait que la perturbation de commande se traduit par un écart de trajectoire à l'instant τ et que la caractérisation de l'optimalité ne se fait qu'à l'instant t_f. Il faut donc relier $\delta z(\tau)$ de (5.11) à $\delta z_0(t_f)$. On procède de la manière suivante :

- Pour trouver $\delta z_0(t_f)$ à partir de $\delta z(t_f)$ il faut extraire la première composante donc, par exemple faire le produit scalaire de $\delta z(t_f)$ avec le vecteur de base de l'axe z_0.
  Il suffit en réalité de calculer le produit scalaire de $\delta z(t_f)$ avec un vecteur $\tilde{\lambda}_f$ satisfaisant les conditions suivantes :
  $\tilde{\lambda}_f^T = \begin{bmatrix} \lambda_{0f} & \lambda_{1f} & \cdots & \lambda_{nf} \end{bmatrix}$ et tel que :

$$\lambda_{0f} > 0 \text{ et } \lambda_f^T \delta x(t_f) = (\lambda_{1f}, \lambda_{2f}, \cdots, \lambda_{nf})\delta x(t_f) = 0 \tag{5.14}$$

  Alors

$$\tilde{\lambda}_f^T \delta z(t_f) = \lambda_{0f} \delta z_0(t_f).$$

  Autrement dit λ_f est orthogonal aux δx admissibles à l'instant t_f, ou encore λ_f est orthogonal au plan tangent à la variété C_f au point $x(t_f)$. On choisira $\lambda_0 = 1$.
- L'évolution de δz de τ à t_f est solution de l'équation (5.12). Mais plutôt que d'essayer de ré soudre cette équation différentielle linéaire à paramètres variant dans le temps, il est plus astucieux de chercher directement à évaluer l'évolution de la forme linéaire $\tilde{\lambda}^T(t)\delta z(t)$ le long de la trajectoire optimale. Pour ce faire on utilise une propriété classique des équations différentielles linéaires :
  Lemma 55 *Soit un système différentiel linéaire :*

$$\dot{y} = M(t)y \tag{5.15}$$

  on appelle système adjoint le système

$$\dot{\zeta} = -M^T(t)\zeta \tag{5.16}$$

  la forme linéaire $\zeta^T(t)y(t)$ est constante pour tout y et tout ζ solutions de (5.15) et (5.16).
  Pour démontrer ce lemme, il suffit de constater que $\frac{d}{dt}\zeta^T(t)y(t) = 0$

Appliquons le lemme précédent à $z(t)$ et à un vecteur $\tilde{\lambda}(t)$ solution de l'équation adjointe de (5.12) :

$$\frac{d\tilde{\lambda}}{dt} = -G_z^T(\hat{z}(t), \hat{u}(t))\tilde{\lambda} \text{ avec } \tilde{\lambda}(t_f) = \tilde{\lambda}_f \tag{5.17}$$

alors $\tilde{\lambda}^T(\tau)\delta z(\tau) = \tilde{\lambda}^T(t_f)\delta z(t_f) = \tilde{\lambda}_f^T \delta z(t_f) = \delta z_0(t_f) = \delta J$
La condition d'optimalité s'exprime donc à l'instant τ :

$$\tilde{\lambda}^T(\tau)\delta z(\tau) > 0 \tag{5.18}$$

ou, grâce à (5.11)

$$\tilde{\lambda}^T(\tau)\varepsilon b[g(\hat{z}(\tau),\omega) - g(\hat{z}(\tau),\hat{u}(\tau))] > 0 \ \forall \varepsilon, b > 0, \forall \omega \tag{5.19}$$

ou encore

$$\tilde{\lambda}^T(\tau)g(\hat{z}(\tau),\omega) > \tilde{\lambda}^T(\tau)g(\hat{z}(\tau),\hat{u}(\tau))\forall\omega \tag{5.20}$$

On en déduit que la valeur $\hat{u}(\tau)$ est la valeur qui minimise

$$H(\hat{z},\tilde{\lambda},\cdot) = \tilde{\lambda}^T(\tau)g(\hat{z}(\tau),\cdot) = \sum_{i=0}^{n} \tilde{\lambda}_i(\tau)g_i(\hat{z}(\tau),\cdot), \ \forall\tau \in (t_0,t_f)$$

H est appelé hamiltonien du problème de commande.
Explicitons l'équation (5.17) grâce à (eq 5.13) en écrivant $\tilde{\lambda} = [\lambda_0 \ \lambda]^T$

$$\frac{d}{dt}\begin{bmatrix}\lambda_0\\ \lambda_1\\ .\\ \lambda_n\end{bmatrix} = -\begin{bmatrix}0 & 0 & . & 0\\ \frac{\partial L}{\partial x_1} & \frac{\partial f_1}{\partial x_1} & . & \frac{\partial f_1}{\partial x_1}\\ . & . & . & .\\ \frac{\partial L}{\partial x_n} & \frac{\partial f_n}{\partial x_n} & . & \frac{\partial f_n}{\partial x_n}\end{bmatrix}\begin{bmatrix}\lambda_0\\ \lambda_1\\ .\\ \lambda_n\end{bmatrix} \tag{5.21}$$

On en déduit :

$$\frac{d\lambda_0}{dt} = 0 \ ; \lambda_0(t) = \lambda_0(t_f) = \lambda_0 = 1 \tag{5.22}$$

Explicitons maintenant l'hamiltonien H avec les variables x et λ

$$H(x,\lambda,u) = L(x,u) + \sum_{1}^{n} \lambda_i f_i(x,u) \tag{5.23}$$

Les équations (5.1) et (5.21) peuvent se ré écrire sous forme condensée, appelées "équations canoniques de Hamilton " :

$$\dot{x} = \frac{\partial H}{\partial \lambda} \tag{5.24}$$

$$\lambda = -\frac{\partial H}{\partial x} \tag{5.25}$$

Lorsque t_f est libre, on doit aussi avoir un critère supérieur au critère optimal , donc $\delta J = \delta z_0(t_f) = \tilde{\lambda}_f^T \delta z(t_f) = \tilde{\lambda}_f^T a g(\hat{z}(t_f),\hat{u}(t_f)) \geq 0 \ \forall a$, ce qui entraîne $\tilde{\lambda}_f^T g(\hat{z}(t_f),\hat{u}(t_f)) = 0$, soit

$$H(\hat{x}(t_f),\lambda(t_f),\hat{u}(t_f)) = 0 \tag{5.26}$$

Enfin on peut montrer que l'hamiltonien optimal est constant le long de la trajectoire optimale.

5.3 Théorème du minimum, cas autonome :

Nous pouvons maintenant énoncer le théorème du minimum.

Théorème 56 *Si $\hat{u}(.)$ et $\hat{x}(.)$ sont respectivement une commande admissible optimale et la trajectoire correspondantes pour le problème (54), alors il existe une fonction absolument continue $\hat{\lambda}(.)$ telle que ($\hat{x}$,$\hat{\lambda}$, $\hat{u}$) vérifient les équations canoniques de Hamilton (5.24 et 5.25) et les conditions du minimum :*

$$H(\hat{x}, \hat{\lambda}, \hat{u}) = \inf_{u \in U} H(\hat{x}, \hat{\lambda}, u) = M(\hat{x}, \hat{\lambda}) \tag{5.27}$$

aux instants initial et final il existe α_0 et α_f tels que :

$$\lambda(t_0) = \alpha_0^T \frac{\partial C_0}{\partial x} \tag{5.28}$$

$$\lambda(t_f) = \alpha_f^T \frac{\partial C_f}{\partial x} \tag{5.29}$$

La fonction $M(\hat{x}(.), \hat{\lambda}(.))$ est constante le long de la trajectoire.
$M = 0$ si t_f est libre.

Remarque 57 *Les conditions énoncées sont des conditions nécessaires pour qu'une commande et la trajectoire correspondante soient optimales. Elles n'en garantissent pas l'existence.*

Remarque 58 *Les conditions de transversalité (5.28-5.29) associées aux conditions aux extrémités sur x fournissent $2n$ conditions ré parties en t_0 et t_f. Elles traduisent le fait que :*
- *$\lambda(t_0)$ est orthogonale au plan tangent à la variété C_0 au point x_0*
- *$\lambda(t_f)$ est orthogonale au plan tangent à la variété C_f au point x_f*

Remarque 59 *Les conditions de transversalité entraînent :*
- *Si pour un indice i, $x_i(t_f)$ est libre, alors $\frac{\partial C_f}{\partial x_i} = 0$ et $\lambda_i(t_f) = 0$.*
- *Si $x(t_f)$ est libre, c'est à dire C_f est $\mathbb{R}^n$, alors $\lambda(t_f) = 0$.*
- *Si $x(t_f)$ est fixé à un point x_f, $\lambda(t_f)$ est libre.*

5.4 Quelques généralisations

Le théorème a été énoncé dans le cas autonome et sans critère terminal. On peut facilement lever ces restrictions.

5.4.1 Cas non autonome

Le temps intervient de manière explicite dans les équations :
Le système est décrit par :

$$\dot{x}(t) = f(x(t), u(t), t) \tag{5.30}$$

le critère :

$$J = \int_{t_0}^{t_f} L(x(t), u(t), t)dt \tag{5.31}$$

Les conditions finales :

$$x_0 = x(t_0) \in C_0(t_0), x_f = x(t_f) \in C_f(t_f) \tag{5.32}$$

Il suffit que l'une de ces conditions soient réalisées pour que le problème devienne non autonome.

On se ramène au cas autonome en introduisant la variable d'état supplémentaire :

$$x_{n+1} = t,$$

ou encore :

$$\dot{x}_{n+1} = 1,\ x_{n+1}(t_0) = t_0, x_{n+1}(t_f) \text{ est libre} \tag{5.33}$$

Le système ainsi complété de variable d'état (x, x_{n+1}) devient autonome et le théorème précédent s'applique.

Appelons $\mathcal{H}$ l'hamiltonien de ce problème en gardant H pour la définition (eq5.23).

$$\mathcal{H} = H + \lambda_{n+1} \tag{5.34}$$

On peut donc observer que la valeur de u qui minimise $\mathcal{H}$ minimise aussi H.

Comme $\lambda_{n+1}(t) = -\frac{\partial H}{\partial t}$; H n'est plus constant le long de la trajectoire optimale.

Comme $x_{n+1}(tf)$ est libre, la condition de transversalité donne $\lambda_{n+1}(t_f) = 0$.

5.4.2 Introduction d'un critère terminal

Le critère à optimiser est :

$$J = \int_{t_0}^{t_f} L(x(t), u(t))dt + \phi(x(t_f), t_f) \tag{5.35}$$

On se ramène au cas standard en incluant le critère terminal dans l'intégration :

$$\phi(x(t_f), t_f) = \int_{t_0}^{t_f} \frac{d}{dt}\phi(x(t), t)dt \tag{5.36}$$

$$= \int_{t_0}^{t_f} (\phi_t + \phi_x^T \dot{x})dt \tag{5.37}$$

$$= \int_{t_0}^{t_f} (\phi_t + \phi_x^T f(x, u))dt \tag{5.38}$$

Le critère J peut donc s'écrire :

$$J = \int_{t_0}^{t_f} (L(x, u) + \phi_t + \phi_x^T f(x, u))dt \tag{5.39}$$

L'hamiltonien du critère s'écrit :

$$\mathcal{H}(x, \lambda, u) = L(x, u) + \phi_t + \phi_x^T f(x, u) + \lambda^T f(x, u) \tag{5.40}$$

$$= H(x, \tilde{\lambda}, u) + \phi_t \ avec\ \tilde{\lambda} = \lambda + \phi_x \tag{5.41}$$

On conclut immédiatement que la commande optimale qui doit minimiser $\mathcal{H}(x, \lambda, .)$, doit minimiser $H(x, \tilde{\lambda}, .)$ puisque ϕ_x et ϕ_t ne dépendent pas de u.

Calculons

$$\frac{d\tilde{\lambda}}{dt} = \dot{\lambda} + \dot{\phi}_x \tag{5.42}$$

$$= -\frac{\partial \mathcal{H}(x, \lambda, u)}{\partial x} + \dot{\phi}_x \tag{5.43}$$

$$= -\frac{\partial L}{\partial x} - \frac{\partial}{\partial x}[(\phi_x + \lambda)^T f] - \frac{\partial \phi_t}{\partial x} + \dot{\phi}_x \tag{5.44}$$

$$= -\frac{\partial L}{\partial x} - \phi_{xx} f - f_x^T(\phi_x + \lambda) - \phi_{tx} + \phi_{xx} f + \phi_{xt} \tag{5.45}$$

$$= -\frac{\partial L}{\partial x} - f_x^T \tilde{\lambda} = -\frac{\partial H}{\partial x} \tag{5.46}$$

De manière évidente

$$\frac{dx}{dt} = \frac{\partial H}{\partial \tilde{\lambda}}$$

On a donc exactement les mêmes conditions que pour le cas standard. Seules changent les conditions de transversalité :

aux instants initial et final il existe α_0 et α_f tels que :

$$\tilde{\lambda}(t_0) - \phi_x(x_0, t_{0)} = \alpha_0^T \frac{\partial C_0}{\partial x} \tag{5.47}$$

$$\tilde{\lambda}(t_f) - \phi_x(x_f, t_f) = \alpha_f^T \frac{\partial C_f}{\partial x} \tag{5.48}$$

Remarquons que si ϕ dépend de t alors, évidemment, t_f est libre donc $\mathcal{H}(\hat{x}, \lambda, \hat{u}) = 0$. On en déduit que :

$$\phi_t = -M(x, \tilde{\lambda}) \textit{ le long de la trajectoire optimale} \tag{5.49}$$

5.5 Comment appliquer le théorème

Contrairement à ce que l'on pourrait imaginer, pour chercher la commande optimale on ne commence pas par chercher les commandes qui transfèrent le système de l'état initial à l'état final désiré pour ensuite chercher la meilleure. On cherche d'abord à caractériser les commandes optimales et seulement après, avec les commandes satisfaisant aux conditions nécessaires d'optimalité, on cherche à satisfaire les conditions aux limites.

5.5.1 Exemple : Commande en temps minimum

Soit le système inertiel d'équation :

$$\ddot{z} = u \tag{5.50}$$

La commande u est contrainte :

$$|u| \leq 1 \tag{5.51}$$

On veut transférer le système de l'état initial : $z(0) = 10,\ \dot{z}(0) = 0$ à l'état final $z(t_f) = 0,\ \dot{z}(t_f) = 0$ en temps minimum.

La démarche est la suivante :

1. Mettre les équations du système sous la forme d'une équation d'état normalisée ($\dot{x} = f(x, u)$)
   Ici on posera $x_1 = z$, $x_2 = \dot{z}$. On obtient alors :

$$\dot{x}_1 = x_2 \tag{5.52}$$

$$\dot{x}_2 = u \tag{5.53}$$

2. Mettre le critère sous forme intégrale. On cherche une commande en temps minimum, on veut donc minimiser t_f soit le critère :

$$J = \int_0^{t_f} dt \tag{5.54}$$

3. Ecrire l'hamiltonien du problème :

$$H(x, \lambda, u) = 1 + \lambda_1 x_2 + \lambda_2 u \tag{5.55}$$

4. A cette étape, on est ramené au problème classique de trouver le minimum d'une fonction d'un nombre fini de variables (u). On applique donc les méthodes de la partie I.
   Ici la solution est simple : H étant une fonction affine de u ses extrema sont sur la frontière donc
   $$\hat{u} = -signe(\lambda_2).$$

5. On utilise maintenant les équations de Hamilton pour trouver des informations sur le comportement du vecteur adjoint :
   $$\dot{\lambda}_1 = -\frac{\partial H}{\partial x_1} = 0 \tag{5.56}$$
   $$\dot{\lambda}_2 = -\frac{\partial H}{\partial x_2} = -\lambda_1 \tag{5.57}$$
   On en déduit que λ_1 est constant et que $\lambda_2(t) = -\lambda_1 t + \lambda_2(0)$.

6. On en déduit que $\lambda_2(t)$ change au plus une fois de signe sur l'intervalle $(0, t_f)$.
   Les commandes satisfaisant les conditions nécessaires d'optimalité sont donc :
   $$u(t) = -1 \text{ pour } t \in (0, t_c),\ u(t) = 1 \text{ pour } t \in (t_c, t_f) \tag{5.58}$$
   $$u(t) = 1 \text{ pour } t \in (0, t_c),\ u(t) = -1 \text{ pour } t \in (t_c, t_f) \tag{5.59}$$
   $$t_0 \leq t_c \leq t_f \tag{5.60}$$

7. Il faut maintenant chercher dans cet ensemble de commandes celles qui satisfont les conditions aux limites. Comme l'état initial et l'état final sont imposés, il n'y a aucune information sur le vecteur adjoint. Ici le problème est simple à résoudre : en considérant l'évolution de la vitesse et les conditions initiales, on déduit immédiatement que la vitesse doit être < 0, donc la commande optimale est : $\hat{u}(t) = -1$ pour $t \in (0, t_c)$; $\hat{u}(t) = 1$ pour $t \in (t_c, t_f)$. On peut alors résoudre les équations et en déduire t_f.
   $$x_2(t) = -t,\ x_1(t) = -\frac{t^2}{2} + 10 \textit{ pour } t \in (0, t_c),\ x_1 = -\frac{x_2^2}{2} + 10 \tag{5.61}$$
   $$x_2(t_c) = -t_c,\ x_1(t_c) = -\frac{t_c^2}{2} + 10 \tag{5.62}$$
   $$x_2(t) = t - 2t_c,\ x_1(t) = \frac{t^2}{2} - 2t_c t + t_c^2 + 10 \textit{ pour } t \in (t_c, t_f); \tag{5.63}$$
   $$x_2(t_f) = t_f - 2t_c, x_1(t_f) = \frac{t_f^2}{2} - 2t_c t_f + t_c^2 + 10 \tag{5.64}$$
   d'où le résultat :
   $$x_2(t_f) = 0 \Rightarrow t_c = \frac{t_f}{2} \tag{5.65}$$
   $$x_1(t_f) = \frac{t_f^2}{2} - t_f^2 + \frac{t_f^2}{4} + 10 \tag{5.66}$$
   $$x_1(t_f) = 0 \Rightarrow t_f = \sqrt{40}; t_c = \sqrt{10} \tag{5.67}$$
   dans le plan de phase :
   $$\textit{pour } t \in (0, t_c)\ x_1 = -\frac{x_2^2}{2} + 10 \tag{5.68}$$
   $$\textit{pour } t \in (t_c, t_f)\ x_1 = \frac{x_2^2}{2} \tag{5.69}$$

D'une manière générale, on montre que pour toute condition initiale (α, β), la fonction temps minimal est déterminée par :

$$T(\alpha,\beta) = \begin{cases} -\beta + \sqrt{2\beta^2 - 4\alpha} \text{ quand } (\alpha,\beta) \text{ est à gauche de } S \\ +\beta + \sqrt{2\beta^2 + 4\alpha} \text{ quand } (\alpha,\beta) \text{ est à droite de } S \end{cases}$$

où la surface de commutation S est donnée par $y^2 = 2|x|$. On peut alors observer que sur ce simple exemple, la fonction coût optimal n'est pas différentiable en tout point de S et donc que HJB ne peut être utilisé !

Remarque 60 *En règle générale l'étape 4 n'est pas si simple ! On obtient (si on sait trouver le minimum analytiquement) $\hat{u}(x,\lambda)$ qu'il faut reporter dans les expressions de $\dot{x}$ et $\dot{\lambda}$ pour obtenir un système de 2n équations différentielles à ré soudre avec les conditions aux limites.*

5.5.2 Exemple : Commande à énergie minimum

Considérons à présent un système linéaire défini par l'équation d'état :

$$\begin{aligned} \dot{x} &= Ax + Bu \\ x(0) &= x_0 \end{aligned}$$

où les matrices A, B sont respectivement de dimension $n \times n$ et $n \times m$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$.

Nous cherchons maintenant à comparer les états suivant l'énergie qu'il faut pour les atteindre. Pour pouvoir effectuer cette comparaison, encore faut-il préciser la commande. On peut en effet dépenser des énergies très différentes pour atteindre un état selon la manière dont on s'y prend.

Pour pouvoir comparer, nous choisissons de comparer les énergies minimales à dépenser pour atteindre les états en partant de l'origine, soit $x_0 = 0$.

Cherchons à résoudre le problème suivant : trouver la commande u qui permette d'aller de l'origine à x en minimisant l'énergie dépensée soit

$$I(x,t) = \int_0^t \|u\|_2^2 \, ds = \int_0^t u^T u \, ds$$

avec $x(t) = x$.

1. On impose l'état initial et final donc on a aucune information sur λ à ces instants.
2. L'hamiltonien du problème s'écrit :

$$H = \frac{1}{2} u^T u + \lambda^T (Ax + Bu)$$

3. La condition nécessaire de minimum : $\frac{\partial H}{\partial u} = 0$ donne

$$u(s) = -B^T \lambda(s).$$

4. Le système adjoint est déterminé par

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = -A^T \lambda$$

soit $\lambda(s) = \exp(-A^T s)\lambda_0$. Par substitution, on en déduit la valeur de la commande :

$$u(s) = -B^T \exp(-A^T s)\lambda_0)$$

d'où

$$\dot{x}(s) = Ax(s) - BB^T \exp(-A^T s)\lambda_0$$

Par intégration,

$$\begin{aligned} x(t) &= -\int_0^t \exp A(t-s)BB^T \exp(-A^T s)\lambda_0 ds \\ &= \left[\int_0^t \exp A(t-s)BB^T \exp A^T(t-s)ds\right] \exp(-A^T t)\lambda_0 \\ &= W_{\mathcal{C}}(t)\exp(-A^T t)\lambda_0 \\ &= W_{\mathcal{C}}(t)\lambda(t) \end{aligned}$$

On voit que pour atteindre $x(t) = x$, il faut que l'état soit dans l'image de $W_{\mathcal{C}}(t)$. Nous supposerons que le système est commandable, donc que l'image de $W_{\mathcal{C}}(t)$ est $\mathbb{R}^n$, $W_{\mathcal{C}}(t)$ est inversible.
d'où :

$$\lambda(t) = W_{\mathcal{C}}^{-1}(t)x(t)$$

et on en déduit que pour tout s, la commande optimale est déterminée par :

$$\begin{aligned} \hat{u}(s) &= -B^T\lambda(s) \\ &= -B^T \exp(A^T(t-s))\lambda(t) \\ &= -B^T \exp(A^T(t-s))W_{\mathcal{C}}^{-1}(t)x(t) \end{aligned}$$

Calculons alors la valeur du critère :

$$\begin{aligned} 2I &= \int_0^t u^T u ds = x(t)^T W_{\mathcal{C}}^{-1}(t) \int_0^t \exp A(t-s)BB^T \exp A^T(t-s)ds \; W_{\mathcal{C}}^{-1}(t)x(t) \\ &= x^T(t)W_{\mathcal{C}}^{-1}(t)x(t) \end{aligned}$$

On vient de mettre en évidence que **le Grammien de commandabilité au temps t, $W_{\mathcal{C}}(t)$, donne par inversion l'énergie minimale nécessaire pour déplacer le système de l'origine à $x(t)$. Autrement dit,** *"plus c'est commandable, moins c'est cher".*

La section suivante va nous permettre de déterminer la matrice $W_{\mathcal{C}}^{-1}(t)$ dans un contexte plus générale.

5.6 Problème LQ en temps continu

On reprend le problème LQ en temps continu présenté à la section 4.7.1.

5.6.1 Equations générales

On applique le théorème de Pontriaguine en présence d'un critère terminal.

1. Les équations sont déjà sous forme "équation d'état" et le critère sous forme intégrale.
2. L'hamiltonien du problème est :

$$H = \frac{1}{2}(x^T Q x + u^T R u) + \lambda^T(Ax + Bu)$$

3. La commande optimale est obtenue en exprimant la condition nécessaire :

$$\frac{\partial H}{\partial u} = Ru + B^T\lambda = 0,$$

d'où

$$\hat{u} = -R^{-1}B^T\lambda \tag{5.70}$$

4. L'équation adjointe est :

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = -Qx - A^T\lambda \tag{5.71}$$

5. En remplaçant u par sa valeur optimale dans l'équation d'état, on parvient au système différentiel :

$$\begin{aligned} \dot{x} &= Ax - BR^{-1}B^T\lambda & (5.72)\\ \dot{\lambda} &= -Qx - A^T\lambda & (5.73) \end{aligned}$$

ou encore :

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = M \begin{bmatrix} x \\ \lambda \end{bmatrix} \tag{5.74}$$

La matrice M est appelée : "matrice hamiltonienne du système".
En résolvant formellement le système (5) :

$$\begin{bmatrix} x \\ \lambda \end{bmatrix}(t) = \exp(M(t-t_0)) \begin{bmatrix} x \\ \lambda \end{bmatrix}(t_0)$$

avec les conditions aux extrémités : $x(t_0) = x_0$ et $\lambda(t_f) = \frac{\partial(\frac{1}{2}x^T P_f x)}{\partial x}|_{t=t_f} = P_f x(t_f)$ car $x(T)$ est libre, on voit que la solution générale peut se mettre sous la forme :

$$\lambda(t) = P(t)x(t),\ P(t_f) = P_f \tag{5.75}$$

En effet en remplaçant t_0 par t_f, on a :

$$\begin{bmatrix} x \\ \lambda \end{bmatrix}(t) = \exp(M(t-t_f)) \begin{bmatrix} Id \\ P_f \end{bmatrix} x(t_f)$$

d'où l'on déduit

$$\lambda(t) = [0_{n\times n}\ Id_n] \exp(M(t-t_f)) \begin{bmatrix} Id \\ P_f \end{bmatrix} x(t_f)$$

et

$$P(t) = [0_{n\times n}\ Id_n] \exp(M(t-t_f)) \begin{bmatrix} Id \\ P_f \end{bmatrix}$$

5.6.2 Solution : équation de Riccati :

On peut donc chercher la solution directement sous la forme :

$$\lambda = P(t)x(t)$$

Par dérivation :

$$\dot{\lambda} = \dot{P}(t)x(t) + P(t)\dot{x}(t) = \dot{P}x + P(Ax + Bu)$$

$$\dot{P}x + PAx - PR^{-1}B^TPx = -Qx - A^TPx$$

On en déduit que P vérifie l'équation différentielle, dite équation de Riccati

$$\dot{P} = -PA - A^T P + PBR^{-1}B^T P - Q \tag{5.76}$$
$$P(t_f) = P_f \tag{5.77}$$

L'intégration de ce système détermine donc la commande optimale à appliquer via (5.70).

5.6.3 Valeurs propres du système différentiel :

Le système différentiel admet des valeurs propres opposées (si s est valeur propre, $-s$ est valeur propre)

$$\det \begin{bmatrix} sI - A & BR^{-1}B^T \\ Q & sI + A^T \end{bmatrix} = \det \begin{bmatrix} sI - A^T & Q \\ BR^{-1}B^T & sI + A \end{bmatrix} \text{ par transposition,} \tag{5.78}$$
$$= \det \begin{bmatrix} sI + A & BR^{-1}B^T \\ Q & sI - A^T \end{bmatrix} \tag{5.79}$$
$$\text{par changement de la numérotation des composantes,} \tag{5.80}$$
$$= \det \begin{bmatrix} -sI - A & BR^{-1}B^T \\ Q & -sI + A^T \end{bmatrix} \tag{5.81}$$

car

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = \begin{vmatrix} -A & B \\ C & -D \end{vmatrix} \tag{5.82}$$
$$= \begin{vmatrix} I & 0 \\ 0 & -I \end{vmatrix} \begin{vmatrix} A & B \\ C & D \end{vmatrix} \begin{vmatrix} -I & 0 \\ 0 & I \end{vmatrix} \tag{5.83}$$

donc : $\det \begin{bmatrix} sI - A & BR^{-1}B^T \\ Q & sI + A^T \end{bmatrix} = \det \begin{bmatrix} -sI - A & BR^{-1}B^T \\ Q & -sI + A^T \end{bmatrix}$
ce résultat montre qu'il y a autant de valeurs propres stables que de valeurs propres instables.

5.6.4 Comportement asymptotique

Les théorèmes

Théorème 61 *Lorsque (A,B) est commandable et (A,C) observable (ou la partie non observable est asymptotiquement stable),*

1. *quand $t_f \to \infty$ la solution de l'équation de Riccati (ER) avec la condition finale $P(t_f) = P_f$ tend vers une valeur stationnaire P indépendante de P_f,*
2. *P est l'unique solution définie positive de l'équation algébrique de Riccati,*

$$0 = PA + A^T P - PR^{-1}B^T P + Q \tag{5.84}$$

3. *le système ainsi régulé est asymptotiquement stable,*

$$\hat{u}(t) = -R^{-1}B^T P x(t) = -Kx(t) \tag{5.85}$$
$$D = A - BK \text{ est stable} \tag{5.86}$$

4. *les valeurs propres de D sont les valeurs propres à parties réelles négatives de la matrice hamiltonienne.*

Proof. Appelons$P_S(t, t_f)$ la solution de l'équation de Riccati avec $P(t_f) = S$

1. Soit donc $P_0(t,t_f)$ la solution quand $P(t_f)=0$

   Montrons que $P_0(t,t_f)$ a une limite $\bar{P}_0(t)$ quand $t_f \to \infty$

   En effet
   - $\hat{V}(x,t_0,t_f)$ est borné car comme (A,B) est commandable, $\exists u$, tel que le système arrive à zéro en temps fini t_1. Il suffit d'appliquer cette commande sur $[t_0,t_1]$ puis $u=0$ pour $t>t_1$ pour obtenir un critère fini supérieur au critère optimal.
   - $\hat{V}(x,t_0,t_f)$ est une fonction non décroissante de t_f. Evident : si $\hat{V}(x,t_0,t_2)<\hat{V}(x,t_0,t_1)$ pour $t_2>t_1$, il suffit d'appliquer la commande $\hat{u}[t_0,t_2]$ sur l'intervalle $[t_0,t_1]$ pour obtenir un résultat meilleur que $\hat{V}(x,t_0,t_1)$.
   - On en déduit que $\hat{V}(x,t_0,t_f)$ a une limite $\bar{V}(x,t_0)$ quand $t_f \to \infty$.
   - Comme $\hat{V}(x,t_0,t_f)=\frac{1}{2}x^T P_0(t,t_f)x,\ \forall x\in\mathbb{R}^n$, on en déduit que $P_0(t,t_f)\to\bar{P}_0(t)$ quand $t_f\to\infty$. $\bar{P}_0(t)$ est solution de l'ER.

   Montrons que $\bar{P}_0(t)=\bar{P}_S(t)\ \forall S$
   - Comme le système (A,C) est reconstructible, la convergence du critère qui implique $u\to 0$ quand $t\to\infty$ entraîne aussi $x\to 0$ quand $t\to\infty$ (car $Cx\to 0\Rightarrow x\to 0$ si (A,C) est détectable : $\mathcal{O}(A,C)x\to 0\Rightarrow x\to 0$), alors pour u qui minimise le critère complet, on a :

$$\begin{aligned}
\frac{1}{2}x_0^T\bar{P}_S(t)x_0 &= \lim_{t_f\to\infty}\frac{1}{2}x_0^T P_S(t,t_f)x_0 && (5.87)\\
&= \int_t^\infty \frac{1}{2}(x^TQx+u^TRu)dt + \lim_{t_f\to\infty} x^T(t_f)Sx(t_f) && (5.88)\\
&= \int_t^\infty \frac{1}{2}(x^TQx+u^TRu)dt && (5.89)\\
&= \frac{1}{2}x_0^T\bar{P}_0(t)x_0,\ \forall t,x_0 && (5.90)
\end{aligned}$$

   donc $\bar{P}_S(t)=\bar{P}_0(t),\ \forall t,S$

   Montrons que $\bar{P}_0(t)$ est constant : $\bar{P}_0(t)=P_0$:
   - En effet la valeur du critère optimal quand $t_f\to\infty$ ne dépend pas de l'instant initial. A,B,Q,R étant constant, tout est invariant par translation dans le temps :

$$\hat{V}(x,t_0,t_f)=\hat{V}(x,t_0+d,t_f+d).$$

     Par passage à la limite :

$$\begin{aligned}
\lim_{t_f\to\infty}\hat{V}(x,t_0,t_f) &= \bar{V}(x,t_0) && (5.91)\\
&= \lim_{t_f\to\infty}\hat{V}(x,t_0+d,t_f+d) && (5.92)\\
&= \bar{V}(x,t_0+d),\ \forall d && (5.93)
\end{aligned}$$

     donc pour tout d, $\bar{V}(x,t_0)=\bar{V}(x,t_0+d)$ et $\bar{P}_0(t)=P_0,\ \forall t$.

2. **Montrons que P_0 vérifie l'ER et $P_0>0$**

   Puisque $\bar{P}_0(t)$ vérifie l'ER, P_0 aussi. Comme $\dot{P}_0=0$, il vérifie l'équation algébrique de Riccati (EAR)

$$\begin{aligned}
P_0A+A^TP_0+Q-P_0BR^{-1}B^TP_0=0 && (5.94)\\
\text{ou encore } P_0D+D^TP_0+Q+K^TRK=0 && (5.95)
\end{aligned}$$

   avec $D=A-BK$, $K=R^{-1}B^TP_0$ et $\hat{u}=-Kx$.

   Montrons qu'il existe une seule solution >0 de l'EAR (Cas (A,C) observable)

Supposons qu'il existe $P_1 \geq 0$, solution de l'EAR, alors $P_{P_1}(t, t_f)$ vérifie l'ER avec $P_{P_1}(t_f, t_f) = P_1$. Comme la solution de ER est unique et $\dot{P}_{P_1}(t_f, t_f) = 0$, alors pour tout t $P_{P_1}(t, t_f) = P_1 = \bar{P}_{P_1}(t)$. Mais comme $\bar{P}_0(t) = \bar{P}_S(t)$, $\forall S$ alors $P_1 = P_0$.
Supposons qu'il existe $x_0 \neq 0$ tel que $x_0^T P_0 x_0 = 0$. L'intégrale $\int_0^\infty (x^T Q x + u^T R u) dt$ ne peut être nul que si $u(t) \equiv 0$, donc

$$x(t) = C \exp(At) x_0$$

et

$$0 = \int_0^\infty x^T Q x dt = x_0^T (\int_0^\infty \exp(A^T t) C^T C \exp(At) dt) x_0,$$

ce qui est impossible puisque le grammien d'observabilité est de rang n.

3. **Montrons que le système ainsi commandé est asymptotiquement stable.**
Ce résultat se déduit du fait que le critère optimal est une fonction de Lyapounov pour le système commandé.
En effet : $V^*(x) = \frac{1}{2} x^T P_0 x > 0, \ x \neq 0$

$$\frac{dV^*(x)}{dt} = \frac{1}{2}(x^T P_0 \dot{x} + \dot{x}^T P_0) x \tag{5.96}$$

$$= \frac{1}{2}(x^T P_0 D x + x^T D^T P_0 x) \tag{5.97}$$

$$= -\frac{1}{2} x_0^T (Q + K^T R K) x_0 \leq 0 \tag{5.98}$$

On montre qu'il ne peut y avoir de valeur propre nulle. En effet si x_0 est un vecteur propre associé à la valeur propre $\lambda = 0$ alors $x_0^T [Q + K^T R K] x_0 = 0$ et donc $x_0^T Q x_0 = 0$ et $x_0^T K^T R K x_0 = 0$ donc x_0 est dans le noyau de C ce qui est incompatible avec l'observabilité.

4. On sait que les valeurs propres du système font partie des valeurs propres de M qui sont opposées. Comme le système est stable, on peut conclure que les valeurs propres du système régulé sont les valeurs propres à parties réelles négatives de cette matrice.

■

5.6.5 Robustesse

L'égalité caractéristique

En utilisant l'expression du gain de retour (eq 5.85), l'équation de Riccati (eq 5.84) peut se mettre sous la forme (5.99)

$$-PA - A^T P + K^T R K = Q \tag{5.99}$$

En ajoutant $0 = sP - sP$, on obtient :

$$P(sI - A) + (-sI - A^T) P + K^T R K = Q \tag{5.100}$$

Ce qui en multipliant à droite par $(sI - A)^{-1} B$ et à gauche par $B^T (-sI - A^T)^{-1}$ donne :

$$B^T (-sI - A^T)^{-1} P B + B^T P (sI - A)^{-1} B + B^T (-sI - A^T)^{-1} K^T R K (sI - A)^{-1} B = B^T (-sI - A^T)^{-1} Q (sI - A)^{-1} B \tag{5.101}$$

En utilisant : $RK = B^T P$, en ajoutant R de chaque coté et en factorisant, on a l'égalité caractéristique du régulateur LQ :

$$[I + B^T (-sI - A^T)^{-1} K^T] R [I + K(sI - A)^{-1} B] = R + B^T (-sI - A^T)^{-1} Q (sI - A)^{-1} B \tag{5.102}$$

De cette équation, en prenant $s = j\omega$ on tire une inéquation fondamentale dans l'étude de la robustesse puisque $Q \geq 0$:

$$[I + B^T (-j\omega I - A^T)^{-1} K^T] R [I + K(j\omega I - A)^{-1} B] > 0 \tag{5.103}$$

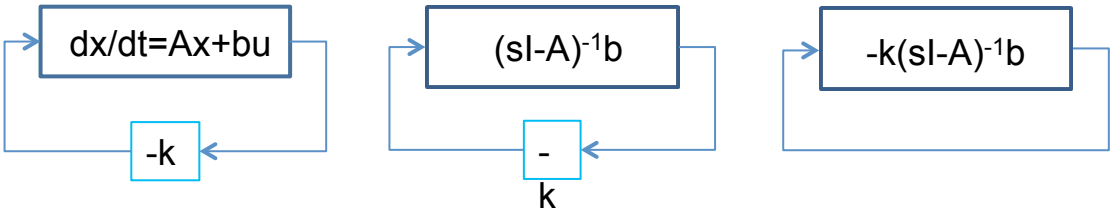


FIGURE 5.1 – Schéma de régulation LQ

Marges de phase, marge de gain

On considère les systèmes mono-entrée ; $u \in \mathbb{R}$
Le critère s'écrit : $J = \int_0^\infty (x^T Q x + u^2) dt$
Le régulateur LQ conduit à un retour d'état $-kx$ selon le schéma (fig5.6.5).
Cette commande est donc équivalente à une fonction de transfert $G(s) = k(sI - A)^{-1}b$ bouclée par un retour unitaire.
L'égalité fondamentale appliquée à ce cas donne :

$$[1 + G(s)][1 + G(-s)] = 1 + b^T(-sI - A^T)^{-1}Q(sI - A)^{-1}b$$

Soit dans le domaine fréquentiel :

$$|1 + G(j\omega)|^2 = 1 + b^T(-j\omega I - A^T)^{-1}Q(j\omega I - A)^{-1}b \geq 1, \ \forall \omega$$

Traduit dans le lieu de Nyquist (fig 5.6.5), cette inégalité montre que la courbe de Nyquist est extérieure au cercle de centre -1 et de rayon 1. Les marges de phase et de gain s'en déduisent :

$$\Delta G = \infty, \ \Delta\phi \geq 60^\circ$$

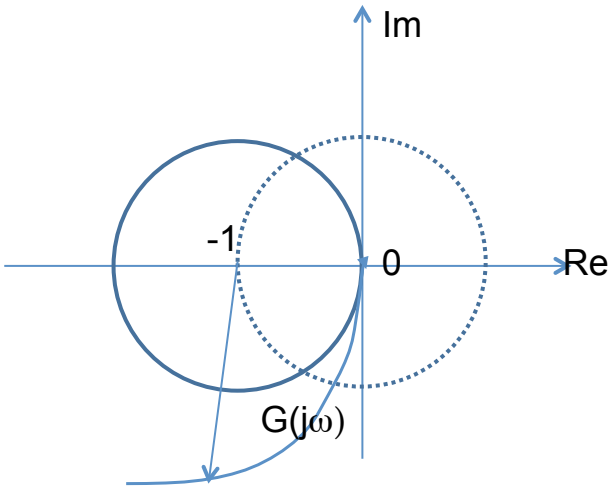


FIGURE 5.2 – Lieu de Nyquist d'une régulation LQ

5.6.6 Degré de stabilité imposé

La construction d'un correcteur optimal nous garantit la stabilité asymptotique. Les valeurs propres en BF sont donc à parties réelles négatives. On pourra imposer une marge de stabilité plus importante (parties réelles inférieures à $-\alpha$) en considérant le correcteur qui minimise le critère :

$$J = \int_0^\infty e^{2\alpha t}(x^T Q x + u^T R u) dt \tag{5.104}$$

En effet, on se ramène au problème classique en posant $\tilde{x} = e^{\alpha t}x$, $\tilde{u} = e^{\alpha t}u$.

On obtient alors le nouveau problème LQ avec :

$$I = \int_0^\infty (\tilde{x}^T Q\tilde{x} + \tilde{u}^T R\tilde{u})dt \tag{5.105}$$

$$\dot{\tilde{x}} = (A + \alpha I)\tilde{x} + B\tilde{u} \tag{5.106}$$

On peut montrer que ce système est commandable. En effet, (A, B) commandable est équivalent à :

Si il existe $V \in \mathbb{R}^n$ tel que $V^T e^{At}B = 0$, $\forall t$ alors $V = 0$ (le noyau de la matrice de commandabilité est réduit à 0).

On a alors aussi : Si $\exists V$ tel que $e^{\alpha t}V^T e^{At}B = V^T e^{((\alpha I+A)t)}B = 0$ $\forall t$ alors $V = 0$

On peut donc en conclure que le régulateur optimal conduit donc à un système asymptotiquement stable. Donc $\tilde{x}(t) \to 0$ quand $t \to \infty$ soit $e^{\alpha t}x(t) \to 0$ quand $t \to \infty$ et les valeurs propres en BF ont une partie réelle inférieure à $-\alpha$.

La commande est

$$\tilde{u} = -R^{-1}B^T P_\alpha \tilde{x}$$

soit

$$u = -R^{-1}B^T P_\alpha x.$$

avec P_α obtenu en remplaçant A par $A + \alpha I$ dans l'ER (eq 5.84).

Dans l'égalité fondamentale (eq 5.102), ceci revient à changer Q en $Q + 2\alpha P_\alpha$.

Dans le cas mono-entrée, on en déduit :

$$|1 + G_\alpha(j\omega)|^2 = 1 + b^T(-j\omega I - A^T)^{-1}(Q + 2\alpha P_\alpha)(j\omega I - A)^{-1}b \geq 1, \ \forall\omega$$

$$|1 + G_\alpha(j\omega)|^2 \geq |1 + G_0(j\omega)|^2$$

car $P_\alpha > 0$; le lieu de Nyquist est encore plus éloigné du point (-1).

Chapitre 6

Résolution numérique

6.1 Méthodes de résolution

Il existe plusieurs façons de résoudre numériquement un problème de commande optimale. La manière la plus directe consiste tout simplement à traiter le problème de commande optimale comme un problème d'optimisation classique. Facile à mettre en oeuvre par sa très grande souplesse (prise en compte des contraintes en l'espace et en temps aisées), cette façon de procéder comporte néanmoins quelques inconvénients : la lourdeur des calculs dû à la dimension du problème, une précision moyenne, la possibilité de terminer sur un minimum local et le renoncement à la synthèse d'un feedback.

Les autres méthodes que l'on peut qualifier d'indirectes, utilisent les deux grands principes énoncés précédemment. Elles sont plus difficiles à mettre en oeuvre, plus sensible mais permettent d'obtenir une plus grande précision voire une commande en boucle fermée.

6.1.1 Méthodes directes

Il s'agit ni plus ni moins que de traduire la formulation du problème de commande optimale sous la forme d'un problème d'optimisation statique puis d'utiliser une méthode adéquate de résolution en fonction de la nature des équations obtenues (voir les chapitres correspondants). Pour un problème en temps continu, on doit appliquer un schéma de discrétisation. On préférera pour des questions de précision et de stabilité un schéma implicite.

Par exemple, si on utilise la formule du trapèze, l'équation d'état sera pour $k = 0, 1, ..., N$:

$$x_{k+1} = x_k + \frac{h}{2}(f_{k+1} + f_k) = 0$$

pour un pas $h = (t_f - t_0)/N$ de discrétisation temporel. En notant
$x_k = x(kh)$,
$u_k = u(kh)$,
$t_k = t_0 + kh$,
$f_k = f(x_k, u_k, t_k)$.
Le critère intégrale devient alors une somme :

$$J = \phi_N + \frac{h}{2}(L_0 + 2\sum_{k=1}^{N-1} L_k + L_N)$$

Il est clair que la dimension du problème est grande par le nombre de variable ainsi que par le nombre de contraintes égalités. Le choix d'une méthode d'optimisation capable de prendre en compte les problèmes de grandes tailles est donc nécessaire (matrices creuses, conditionnement, etc).

Méthode de tirs multiples

On préfère pour une question de convergence, appliquer la variante suivante : on découpe l'intervalle $[t_0, t_f]$ en M sous intervalles $[\tau_i\ \tau_{i+1}]$, $i = 0, ..., M-1$ avec $M-1$ conditions initiales arbitraires z_i, $i = 1, ..., M-1$ (la première est déjà fixée puisqu'il s'agit de x_0).

On écrit les équations d'états discrètisées pour chaque sous intervalle $[\tau_i\ \tau_{i+1}]$ en partant de la condition initiale z_i. On ajoute afin d'assurer le recollement de la solution aux bornes des sous intervalles, les contraintes $x(\tau_i) = z_i$, $i = 1, ..., M-1$. Les bornes des intervalles τ_i ne sont pas nécessairement fixées.

Cette manière de procédé est dites de multi-tirs est s'avère dans la pratique bien plus efficace [**?**] bien qu'elle augmente la dimension du problème.

Méthode pseudospectrale

Ces méthodes reprennent la formulation multi-tirs mais avec une approche optimisation paramétrique : l'état, la commande et l'adjoint sont remplacés par une approximation polynomiale sur $[\tau_i\ \tau_{i+1}]$. Ces méthodes réduisent la taille du problème : les (x_i, u_i, λ_i) sont remplacés par les coefficients des polynômes d'interpolation beaucoup moins nombreux. Une toolbox Matlab GPOPS existe et utilise cette approche [**?**].

6.1.2 Les méthodes indirectes

Par le principe du minimum

On cherche à résoudre le système hamiltonien et satisfaire les conditions de transversalité.

La première des méthodes que l'on puisse utiliser est une méthode de tirs.

1. On choisit un couple initial $p_0 = (x, \lambda)(t_0)$ admissible i.e. vérifiant les conditions initiales et de transversalité.
2. On intègre à l'aide d'un schéma numérique le système hamiltonien avec une commande déterminée à chaque pas par la minimisation de l'hamiltonien.
3. On vérifie si les conditions finales et de transversalité sont satisfaites ou non
4. Si oui on arrête sinon on corrige le tir en modifiant p_0 et on recommence.

Le choix du mécanisme de correction s'effectue en observant que l'ensemble des conditions à satisfaire aux deux bouts de l'équation différentielle peut s'exprimer comme une contrainte égalité, de type

$$G(p_0, p_{t_f}(p_0)) = 0 \tag{6.1}$$

On utilise alors une méthode de recherche d'un zéro. Dans la pratique, la sensibilité $p_{t_f} = (x_{t_f}, \lambda_{t_f})$ par rapport à p_0 est grande et il est préférable d'utiliser **une méthode de tirs multiples** (voir fin de la section précédente).

Remarque 62 *Lorsque le temps t_f n'est pas fixé, le temps d'intégration est également une condition initiale à déterminer au même titre que p_0.*

Lorsque x_0 est fixé et x_{t_f} est libre, une deuxième méthode peut être :

1. On choisit une suite initiale de commandes u^k admissibles
2. On intègre à l'aide d'un schéma numérique les équations d'état
3. On intègre en temps rétrograde le système adjoint en partant de la condition de transversalité ad hoc $\lambda(t_f) = \alpha_0 \frac{\partial \phi}{\partial x}(t_f, x(t_f))$
4. On vérifie si la condition d'optimalité de H est satisfaite ou non
5. Si oui on arrête sinon on corrige la suite des commandes u^k à l'aide d'un algorithme de façon à satisfaire $\frac{\partial H}{\partial u} = 0$ et on recommence.

Par l'équation de HJB

A la condition que la fonction coût optimal $\hat{J}$ soit C^1 (ce qui est rarement le cas, par exemple si le feedback est discontinus), l'équation d'HJB

$$\frac{\partial \hat{J}}{\partial t} = -\inf_u \left(L(x(t), u, t) + \frac{\partial \hat{J}^T}{\partial x} f(x(t), u, t) \right) \text{ p.p.} \tag{6.2}$$

avec la condition aux limites $\hat{J}(x(t_f), t_f, t_f) = \phi(x(t_f), t_f)$ est satisfaite. Comme toute edp non stationnaire elle doit être résolue sur un domaine temporel et spatial.

La plus simple des méthodes est obtenue pour un schéma de discrétisation aux différences finies (Il existe de nombreuses autres méthodes).

Par exemple, en dimension 1, si on discrétise l'espace selon une grille de pas h_x et le temps avec un pas h_t et en notant i l'indice spatial et j l'indice temporel, on obtient pour un schéma d'Euler implicite en temps le schéma : pour $i = 1, ..., N$ et $j = 1, ..., M$

$$\frac{\hat{J}_i^{j+1} - \hat{J}_i^j}{h_t} = -\inf_{u_{j+1}} \left(L_i^{j+1} + \frac{\hat{J}_i^{j+1} - \hat{J}_{i-1}^{j+1}}{h_x} f_i^{j+1} \right)$$

avec $f_i^{j+1} = f(x_i^{j+1}, u^{j+1}, t^{j+1})$ et $L_i^{j+1} = L(x_i^{j+1}, u^{j+1}, t^{j+1})$ et la condition finale $\hat{J}_i^M = \phi(x_i, t_M)$.

Rappelons tout de même que cette méthode lourde est particulièrement avantageuse si la minimisation de l'hamiltonien permet de déterminer $u = u(x, \frac{\partial J}{\partial x}, t)$ auquel cas la résolution de l'équation de HJB fournira une commande en boucle fermée.