

COMMANDE OPTIMALE DE SYSTÈMES STOCHASTIQUES

THEOSYS (*)

Résumé. — Nous discutons un ensemble de méthodes conduisant à la résolution exacte ou approchée de problème de contrôle stochastique.

Un système informatique du type système expert, en cours de développement, permettant l'utilisation de ces méthodes est expliqué.

Abstract. — We discuss a set of methods for solving stochastic control problems. These methods give the exact solution or an approximation of it. The state of an expert system, based on them is explained.

INTRODUCTION

Les processus de diffusions sont des objets très utiles pour modéliser les équations différentielles perturbées par un bruit. La commande de tels processus a de nombreuses applications (gestion de stock, maintenance, finance etc...).

Les conditions d'optimalité sont obtenues par la méthode de la programmation dynamique qui conduit à la résolution d'équations aux dérivées partielles non linéaires. La plus simple s'écrit :

$$\begin{cases} \frac{\partial}{\partial t} y + \text{Min}_u \left(\sum_{i=1}^n b_i(x, u) \frac{\partial y}{\partial x_i} + c(x, u) \right) + \Delta y = 0 \\ y(T, x) \text{ donné.} \end{cases}$$

En général, cette équation ne peut pas être résolue analytiquement. On la résout souvent en la discrétisant par des méthodes de type éléments finis ou différences finies. Une autre approche possible qui peut être également vue comme une interprétation stochastique des méthodes précédentes consiste à discrétiser le problème de contrôle stochastique lui-même.

On est conduit alors à résoudre des problèmes de commandes optimales de chaînes de Markov.

Une des principales difficultés pour résoudre le problème discret est le grand nombre d'états de la chaîne de Markov. Ce nombre croît exponentielle-

(*) Cet article est le résultat d'un travail collectif auquel ont participé : A. Bensoussan, F. Delebecque, C. Gomez, M. Goursat, P. V. Kokotovic, J. P. Quadrat, A. Sulem, M. Viot, INRIA, Domaine de Voluceau-Rocquencourt, B.P. 105, 78153 Le Chesnay, France.

ment avec la dimension de l'état du processus de diffusion (ici n). Lorsque n grandit, il devient très rapidement impossible de résoudre le problème par la méthode de la programmation dynamique. Trois approches sont possibles pour éviter cette difficulté :

- optimisation dans une classe de feedbacks découplant le système,
- méthodes de Monte Carlo ou de gradient stochastique,
- méthodes de perturbations.

Une discussion des méthodes d'agrégation est faite. Les méthodes multigrilles dérivant de ces idées d'agrégation, ainsi que des méthodes de perturbation singulières, conduisent à des algorithmes rapides pour résoudre l'équation de la programmation dynamique. La mise en œuvre de l'ensemble de ces approches est en général lourde. En vue d'alléger ce travail, des techniques de manipulation symbolique sont employées pour :

- générer des programmes,
- faire de la démonstration automatique,
- spécifier le problème en langage semi-naturel.

L'état d'avancement de ce travail informatique est indiqué.

Le plan du travail est le suivant :

1. PROBLÈME DE CONTRÔLE STOCHASTIQUE
2. RÉOLUTION DE L'ÉQUATION DE LA PROGRAMMATION DYNAMIQUE
 - 2.1. *Discrétisation*
 - 2.2. *Agrégation*
 - 2.3. *Méthodes multigrilles*
 - 2.4. *Perturbations singulières*
3. OPTIMISATION DANS LA CLASSE DES FEEDBACKS LOCAUX
 - 3.1. *Systèmes ayant la dynamique découplée*
 - 3.2. *Systèmes ayant la propriété de la forme produit*
4. OPTIMISATION DANS UNE CLASSE DE FEEDBACKS DONNÉE A PRIORI
5. MÉTHODES DE PERTURBATION RÉGULIÈRE
6. VERS UN SYSTÈME EXPERT EN CONTRÔLE STOCHASTIQUE
 - 6.1. *Génération de programme*
 - 6.2. *Démonstration automatique*
 - 6.3. *Spécification du problème en langage semi-naturel*

1. LE PROBLÈME DE CONTROLE STOCHASTIQUE

Défini sur l'espace de probabilité (Ω, F, F, P) , on considère le processus de diffusion commandé :

$$dX_t = b(X_t, u(X_t)) dt + \sigma dW_t \tag{1.1}$$

où $X_t \in \mathbb{R}^n$,

W_t est un brownien,

$u : \mathbb{R}^n \rightarrow \mathbb{R}^m$ est une commande en boucle fermée sur l'état,

σ est une matrice (n, n) .

Étant donné :

\mathcal{O} : un ouvert de \mathbb{R}^n de frontière Γ

U : un ensemble fermé de \mathbb{R}^m

τ : un temps d'arrêt défini par : $\tau(\omega) = \{ \text{argmin}_t X_t(\omega) \notin \mathcal{O} \}$

$c : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^+$ un coût instantané

$f : \mathbb{R}^n \rightarrow \mathbb{R}^+$ un coût sur l'état final

$T \in \mathbb{R}^+$: un horizon de gestion

λ : un taux d'actualisation,

nous voulons résoudre l'un des problèmes suivants :

$$y(x, s) = \text{Min}_{u \in U} E \left[\int_s^{T \wedge \tau} c(X_t, u(X_t)) dt + f(X_{T \wedge \tau}) \mid X(s) = x \right], \tag{1.2}$$

$$y(x) = \text{Min}_{u \in U} E \left[\int_0^\tau e^{-\lambda t} c(X_t, u(X_t)) dt + e^{-\lambda \tau} f(X_\tau) \mid X(0) = x \right], \tag{1.3}$$

$$y(x) = \text{Min}_{\emptyset} E \left[\int_0^\tau e^{-\lambda t} c(X_t) dt + e^{-\lambda \tau} f(x_\tau) \mid X(0) = x \right]. \tag{1.4}$$

Nous supposons dans la dernière situation que c est indépendant de u .

Des cas plus généraux peuvent être envisagés tels que la modulation de l'intensité d'un processus de saut, les problèmes de contrôle impulsions etc... Cependant, les exemples discutés ici présentent la plupart des difficultés numériques.

Si l'on note :

$$L(x, u, p, q) = \sum_{i=1}^n b_i(x, u) p_i + \sum_{\substack{i=1, n \\ j=1, n}} a_{ij} q_{ij} + c(x, u) \tag{1.5}$$

$$a = \frac{1}{2} \sigma \sigma^* . \tag{1.6}$$

Les équations de la programmation dynamique des problèmes (1.2), (1.3), (1.4) s'écrivent :

$$\left. \begin{aligned} D_0 y + \operatorname{Min}_{u \in U} L(x, u, Dy, D^2 y) = 0 \quad x \in \mathcal{C} . \\ y(x, t) = f(x, t) \quad x \in \Gamma \quad \text{ou} \quad t = T \end{aligned} \right\} \quad (1.7)$$

$$\left. \begin{aligned} -\lambda y + \operatorname{Min}_{u \in U} L(x, u, Dy, D^2 y) = 0, \quad x \in \mathcal{C} \\ y(x) = f(x) \quad x \in \Gamma \end{aligned} \right\} \quad (1.8)$$

$$\operatorname{Min} [-\lambda y + L(x, Dy, D^2 y), f - y] = 0 \quad x \in \mathbb{R}^n . \quad (1.9)$$

On utilise les notations :

$$\begin{aligned} D_0 &= \frac{\partial}{\partial t} \\ D_i &= \frac{\partial}{\partial x_i} \\ D_{ij} &= \frac{\partial^2}{\partial x_i \partial x_j} \\ D &= [D_1, \dots, D_n] \\ D^2 &= [D_{ij}] . \end{aligned}$$

En général, les équations (1.7), (1.8), (1.9) ne peuvent pas être résolues analytiquement. On les résout de façon approchée en discrétisant le problème.

2. RÉOLUTION DE L'ÉQUATION DE LA PROGRAMMATION DYNAMIQUE

2.1. Discrétisation de l'équation de la programmation dynamique

Deux approches sont possibles :

a) Discrétisation de l'équation de la programmation dynamique par la méthode des éléments finis Quadrat [15], Goursat-Quadrat [9], ou par des méthodes de différences finies.

b) Discrétisation du problème de contrôle stochastique. On se ramène alors à un problème de contrôle de chaîne de Markov (Kushner [10], Quadrat [14]).

Après discrétisation, on a à résoudre l'un des problèmes suivants :

$$y^n = \operatorname{Min}_{u \in U} [M(u) y^{n+1} + c(u)], \quad y^N = f \quad (2.1)$$

$$\text{Min}_{u \in U} [A_\lambda(u) y^{n-1} + c(u)] + y^n = 0, \quad y^N = f \quad (2.2)$$

$$\text{Min}_{u \in U} [A_\lambda(u) y + c(u)] = 0 \quad (2.3)$$

où $M(u)$ désigne une matrice de transition d'une chaîne de Markov, $A_\lambda(u) = M(u) - (1 + \lambda)I$ avec I la matrice identité.

En général, les matrices A ou M sont de dimension grande et ce nombre croît de façon exponentielle avec la dimension du vecteur d'état X_t .

Dans le paragraphe suivant, nous nous intéressons à des méthodes rapides de résolution de (2.2), (2.3).

2.2. Agrégation exacte (Delebecque-Kokotovic-Quadrat) [20]

Étant donné une matrice $A(n, n)$; l'agrégation consiste à bloc triangulariser la matrice A et à conserver un espace invariant du système. Pour cela, on peut introduire deux matrices : une matrice $C(N, n)$ et une matrice $B(n, N)$ qui correspondent dans la théorie des systèmes linéaires aux matrices d'observation et de contrôle.

On dira alors, suivant Aoki, que le système est *agrégable* pour la matrice d'observation C s'il existe une matrice \bar{A}_C telle que :

$$CA = \bar{A}_C C \quad (2.4)$$

ce qui signifie que l'espace engendré par les vecteurs lignes de la matrice C forme un espace invariant à gauche de la matrice A . On a la notion duale.

On dira, suivant Kokotovic, que le système est *cohérent* pour la matrice de commande B , s'il existe une matrice \bar{A}_B telle que

$$AB = B\bar{A}_B \quad (2.5)$$

ce qui signifie que l'espace engendré par les colonnes de B forment un espace invariant à droite de A .

Les systèmes agrégés sont alors définis par des matrices \bar{A}_C et \bar{A}_B .

Considérons le système dynamique :

$$\dot{x} = Ax \quad x(0) \text{ donné} \quad (2.6)$$

Si A est C agrégable, alors le vecteur $y = Cx$ satisfait l'équation

$$\dot{y} = \bar{A}_C y, \quad y(0) = Cx(0) \quad (2.7)$$

Si A est B cohérent, alors x est de la forme Bv ou v vérifie :

$$\dot{v} = \bar{A}_B v \quad (2.8)$$

à condition que

$$x(0) = Bv(0).$$

Ces notions se particularisent aux chaînes de Markov en spécifiant les matrices B et C .

On se donne une chaîne de Markov de matrice de transition A et une matrice U associée à une partition $\mathcal{U} = \{E_1, E_2, \dots, E_n\}$ de l'espace des états.

La colonne $U_{.i}$ représente le vecteur caractéristique de l'ensemble E_i , c'est-à-dire

$$U_{ji} = \begin{cases} 1 & \text{si } j \in E_i \\ 0 & \text{si } j \notin E_i. \end{cases}$$

La condition d'agrégabilité écrite pour la matrice transposée de A , noté A' , s'écrit

$$AU = U\bar{A} \quad (2.9)$$

correspond exactement à la condition d'agrégabilité de Kemeny-Snell [21]. Elle permet de construire une chaîne de Markov agrégée, dont les états sont les ensembles E_i et la matrice de transition est \bar{A} .

La notion duale de cohérence nécessite l'introduction de la mesure invariante de la chaîne de Markov, c'est-à-dire q définie par :

$$\left. \begin{aligned} qA &= q \\ q\mathbb{1} &= 1 \end{aligned} \right\} \quad (2.10)$$

où $\mathbb{1}$ désigne le vecteur colonne dont toutes les composantes valent 1.

On utilise alors la matrice $q^{\mathcal{U}}$ qui est associée à la probabilité d'être dans un état i connaissant la partition \mathcal{U} pour la mesure invariante. Plus précisément :

$$q_{ij}^{\mathcal{U}} = \begin{cases} \frac{q_{ij}}{\sum_{j \in E_i} q_{ij}} & j \in E_i \\ 0 & \text{ailleurs.} \end{cases} \quad (2.11)$$

On montre que la condition de cohérence pour $B = q^{\mathcal{U}}$ de la chaîne de Markov est une condition suffisante pour que le filtre, basé sur l'observation de l'ensemble E_i auquel appartient l'état de la chaîne de Markov, ait un nombre fini d'états. Cette situation est l'analogie de la recherche de statistiques exhaustives de dimension finie dans le cas du filtre à états continus.

Les liens entre ces notions d'agrégabilité et de cohérence avec la notion, très utilisée dans la théorie des files d'attente, d'équilibre partiel a été faite dans Delebecque-Kokotovic-Quadrat [20].

Un résultat d'agrégation-désagrégation permettant de résoudre le problème complet et non pas seulement le problème agrégé est démontré dans Quadrat [22]. Il améliore un résultat précédent de Komiakhov [23].

Ces idées d'agrégation réalisent, à notre avis, toute leur puissance dans les algorithmes de type multigrille pour résoudre les équations aux dérivées partielles.

2.3. Méthodes multigrilles (Brandt [24], Nicolaïdes [25])

L'idée des méthodes multigrilles est de construire une cascade de problèmes de plus en plus agrégés de façon à résoudre efficacement le discrétisé d'une équation aux dérivées partielles.

Plus précisément, soit à résoudre le problème :

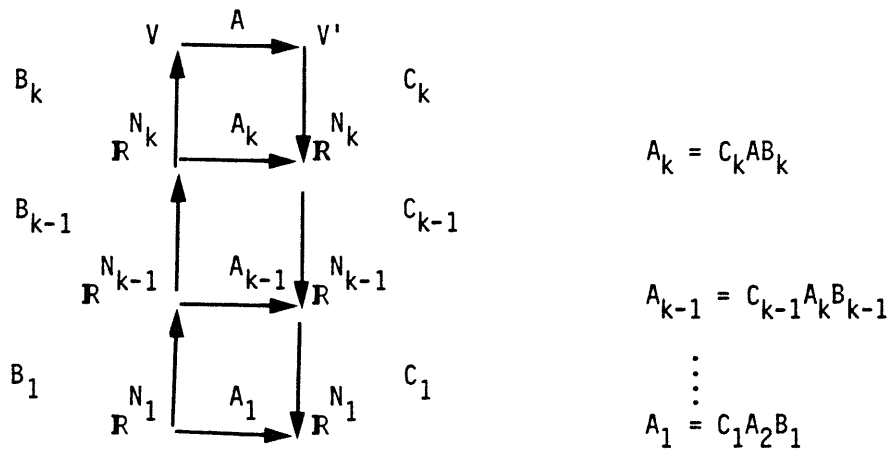
$$Ax + u = 0 \text{ avec } A : V \rightarrow V'. \tag{2.12}$$

V désigne un espace de Hilbert et V' son transposé après identification d'un espace pivot avec son dual.

A est un opérateur linéaire, borné, coercif sur V pour fixer les idées disons $A = \Delta - \lambda, V = H^1$ ou

$$H^1(o) = \{ f \mid f \in L^2(o), Df \in L^2(o) \}.$$

On construit alors la cascade de problème de plus en plus agrégé



$$N_1 < N_2 < \dots < N_k$$

Nous supposons dans un premier temps que les propriétés d'agrégation et de cohérence exacte sont réalisées pour toute cascade et que les opérateurs de projection $P_i = B_i C_i$ conservent les espaces invariants associés aux valeurs propres les plus lentes.

Pour résoudre

$$A_k X_k + U_k = 0 \quad (2.13)$$

avec

$$U_k = C_k u \quad (2.14)$$

qui est le discrétisé du problème initial, on peut utiliser une méthode itérative qui consiste essentiellement à intégrer l'équation différentielle

$$\dot{X}_k = A_k X_k + U_k \quad (2.15)$$

après avoir remarqué que les valeurs propres de A_k sont toutes strictement négatives.

Malheureusement le spectre de la matrice A est très étalé, ce qui conduit à un mauvais conditionnement de la matrice A donc à un grand nombre d'étapes d'intégration de (2.15). Ce conditionnement dépend du pas de discrétisation de l'équation aux dérivées partielles. Les méthodes multigrilles permettent de construire un problème équivalent à (2.15), à savoir :

$$\dot{Z}_k = A_k [I + \alpha_{k-1} P_{k-1} + \alpha_{k-2} P_{k-2} + \dots + \alpha_1 P_1] Z_k + U_k \quad (2.16)$$

où les $\alpha_{k-1}, \alpha_{k-2}, \dots, \alpha_1$ sont choisis de façon à ce que le conditionnement de la matrice

$$E_k = A_k [I + \alpha_{k-1} P_{k-1} + \dots + \alpha_1 P_1] \quad (2.17)$$

soit le meilleur possible.

On remarque que lorsqu'on a calculé le point d'équilibre de (2.16) Z_α il est facile de retrouver le point d'équilibre de (2.15) X_α . On a la relation :

$$(I + \alpha_{k-1} P_{k-1} + \dots + \alpha_1 P_1) Z_\alpha = -A_k^{-1} U_k = X_\alpha \quad (2.18)$$

La raison pour laquelle le conditionnement de (2.17) est meilleur que celui de A_k se comprend facilement. Si l'on ordonne par ordre croissant les valeurs propres de $A_k : \lambda_1, \dots, \lambda_k$, les valeurs propres de E_k deviennent :

$$\begin{aligned} \lambda_i \left(1 + \sum_{i=1}^{k-1} \alpha_i \right) & \quad i = 1, N_1 \\ \lambda_i \left(1 + \sum_{i=2}^{k-1} \alpha_i \right) & \quad i = N_1 + 1, \dots, N_2 \\ \vdots & \\ \lambda_i & \quad i = N_{k-1} + 1, \dots, N_k. \end{aligned}$$

Il est alors possible de choisir les coefficients de façon à rendre le conditionnement de la matrice indépendant du pas de discrétisation.

En pratique, on n'est pas capable de calculer exactement les projecteurs P_{k-1}, \dots, P_1 cependant, on arrive à le faire de façon approchée. Il est possible de montrer que malgré l'approximation des P_k , on obtient un conditionnement indépendant des pas de discrétisation (par exemple Hackbush [26]).

On a montré alors qu'en utilisant ces techniques on dispose d'algorithmes dont la complexité est en $N_k \log N_k$ pour résoudre des problèmes du type (2.2) ou (2.3) Horkay [27]. Rappelons que N_k est le nombre de points de discrétisation du problème qui, lui bien sûr, croît de façon exponentielle avec la dimension du vecteur d'état X_t du problème de contrôle.

2.4. Perturbation singulière de chaîne de Markov

Une autre approche pour simplifier les problèmes est l'utilisation de technique de perturbation singulière. On est cette fois dans la situation du problème (2.2) où le générateur $A_\lambda(u)$ dépend aussi de ϵ .

On réécrit alors (2.2) :

$$\text{Min}_u [B(u) + \epsilon A(u) - \mu \epsilon^k] V^\epsilon + C(u, \epsilon) = 0 \tag{2.19}$$

où $B(u) + \epsilon A(u)$ est un générateur de chaîne de Markov, c'est-à-dire que la somme en ligne est nulle et que les termes hors diagonaux sont positifs.

La situation intéressante est celle où $B(u)$ est bloc diagonal, et correspond par exemple à la discrétisation de problème dans lequel la dynamique peut s'écrire :

$$\begin{aligned} dX_t &= b(X_t, Y_t, u_t) dt + \sigma dW_t \\ dY_t &= \frac{1}{\epsilon} b(X_t, Y_t, u_t) dt + \frac{1}{\sqrt{\epsilon}} \sigma dW_t \\ \text{Min } E \int_0^\tau e^{-\lambda t} \epsilon C(X_t, Y_t, u_t, \epsilon) dt. \end{aligned}$$

Dans cet exemple, $k = 1$ le système a deux échelles de temps, celle associée à la dynamique de Y , qualifiée de rapide, celle associée à X , qualifiée de lente.

On montre le résultat suivant :

THÉORÈME : *Le coût optimal V^ε est analytique en ε . Le développement correspondant*

$$V_x = \sum_{i=0}^{+\infty} \varepsilon^i V_x^i$$

est l'unique solution qui existe de :

$$\text{Min}_{S \in \mathcal{P}_0} \phi^0 \equiv B^S V^0 = 0$$

$$\text{Min}_{S \in \mathcal{P}_i} \phi^i \equiv B^S V^i + A^S V^{i-1} = 0 \quad i = 1, \dots, k-1$$

$$\text{Min}_{S \in \mathcal{P}_i} \phi^i \equiv B^S V^i + A^S V^{i-1} - \mu V^{i-k} + C_{i-n}^S = 0 \quad i = k, k+1, \dots, k+p-1$$

$$\text{Min}_{S \in \mathcal{P}_i} \phi^i \equiv B^S V^i + A^S V^{i-1} - \mu V^{i-k} = 0 \quad i = k+p, k+p+1, \eta$$

$$\phi^i \equiv B^S V^i + A^S V^{i-1} - \mu V^{i-k} = 0 \quad i = \eta+1, \dots, \infty$$

où η est un nombre dépendant seulement de n, p et k , S désigne une politique. Les \mathcal{P}_i sont définis par :

$$\mathcal{P}_0 = \mathcal{P}$$

$$\mathcal{P}_i = \underset{S \in \mathcal{P}_{i-1}}{\text{Argmin}} \phi_{i-1} \quad i = 1, \dots, \eta$$

où : $\{ S : E \rightarrow U \}$ avec E l'espace des états et U un ensemble fini de valeurs possibles pour le contrôle.

Ceci permet de faire passer la complexité de l'algorithme de Howard de n^3 à n^2 dans la situation la plus favorable : B bloc diagonal constitué d'environ \sqrt{n} blocs de taille \sqrt{n} . n désigne ici le nombre d'états de la chaîne de Markov. Le cas linéaire pas de minimisation a été étudié dans Delebecque [28], les applications aux cas non linéaires par Quadrat [29].

3. OPTIMISATION DANS LA CLASSE DES FEEDBACKS LOCAUX

En général, ce problème est plus difficile à résoudre que de calculer le feedback global Levine [1]-Quadrat [16]. Cependant, dans le cas particulier suivant, cette approche est intéressante pratiquement.

3.1. Système ayant la dynamique découplée

Nous notons $I = \{ 1, 2, \dots, n \}$ l'ensemble des sous-systèmes. Pour simplifier les notations, nous supposons que la dimension de l'état de chaque sous-système est 1.

Nous étudions le cas particulier où b_i est une fonction de x_i et u_i seulement $\forall i \in I$:

$$b_i : \mathbb{R}^+ \times \mathbb{R} \times V \rightarrow \mathbb{R} \quad . \quad (3.1)$$

$t \quad x_i \quad u_i \quad b_i(t, x_i, u_i)$

Les bruits sont découplés, c'est-à-dire que σ est diagonale. L'ensemble des contrôles admissibles $U = \prod_{i \in I} U_i, U_i \subset \mathbb{R}$. Nous appelons R une stratégie locale, c'est-à-dire :

$$R = (R_1, \dots, R_n)$$

$$R_i : \mathbb{R}^+ \times \mathbb{R} \rightarrow U_i \quad . \quad (3.2)$$

$t \quad x_i \quad u = R_i(t, x_i)$

Sous ces hypothèses, la densité de probabilité de présence en un point (x_1, \dots, x_n) , à l'instant t de X_t noté $p^R(t, x)$ vérifie :

$$p^R(t, x) = \prod_{i=1}^n p^{R_i}(t, x_i) \quad (3.3)$$

et $p_i = p^{R_i}$ satisfait l'équation :

$$L_{i,R_i}^* p_i = 0 \quad p_i(0, x_i) = \mu_i \quad \text{donné si} \quad \mu = \prod_{i=1}^n \mu_i$$

* désigne la transposition

$$L_i = D_0 + b_i \circ R_i D_i + a_{ii} D_i \quad (3.4)$$

où $b \circ R$ note $b(t, x, R(t, x))$.

Notons également

$$C_i^R \circ R_i : \mathbb{R}^+ \times \mathbb{R} \rightarrow \mathbb{R}^+ \quad (3.5)$$

$t \quad x_i \quad \int c \circ R(t, x) \prod_{j \neq i} p_j(t, x_j) dx_j$

c'est-à-dire la moyenne conditionnelle du coût instantané connaissant l'état du sous-système i et les stratégies des autres sous-systèmes.

Les conditions d'optimalité joueur par joueur sont données dans le :

THÉORÈME : *Une condition suffisante pour une stratégie locale R d'être optimale joueur par joueur est qu'elle satisfasse :*

$$\underset{R_i}{\text{Min}} [L_{i,R_i} y_i + C_i^R \circ R_i] = 0. \quad (3.6)$$

Le coût optimal vaut alors :

$$\mu_1(y_1) = \dots = \mu_n(y_n) = \int y_i(0, x_i) \mu_i(dx_i). \quad (3.7)$$

L'algorithme suivant permet de calculer cette stratégie.

Étant donné $\varepsilon, v \in \mathbb{R}^+$:

Étape 1 :

- choisir $i \in I$
- résoudre (3.6)
- si $\mu_i(y_i) \leq v - \varepsilon$ alors

$$v_i = \mu(y_i)$$

$$R_i = \underset{R_i}{\text{Argmin}} [L_{i,R_i} y_i + C_i^R \circ R_i]$$

sinon choisir un autre $i \in I$ jusqu'à ce que

$$\mu_i(y_i) \geq v - \varepsilon \quad \forall i \in I.$$

Étape 2 :

- quand l'étape 1 est terminée

$$\varepsilon := \varepsilon/2$$

- retourner à l'étape 1.

Par cet algorithme, on obtient une suite décroissante $v^{(n)}$ qui converge vers le coût optimal souhaité.

Une preuve de convergence de cet algorithme dans le cas discret est donné dans Quadrat-Viot [7].

La résolution du problème se ramène donc à la résolution d'un système couplé d'équations aux dérivées partielles non linéaires sur \mathbb{R} .

3.2. Système ayant la propriété de la forme produit

L'hypothèse sur le découplage de la dynamique du système est très restrictive. Nous présentons une classe de système à dynamique couplée pour laquelle la méthode précédente s'applique.

Supposons que l'on s'intéresse au problème stationnaire suivant :

$$\text{Min}_S \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T c \circ S(X_t) dt$$

pour un système dont le générateur a les propriétés suivantes :

— étant donné B un générateur de chaîne de Markov sur I et des fonctions

$$\begin{aligned} u_i &: \mathbb{R} \rightarrow \mathbb{R}, \\ &\quad x_i \quad u_i(x_i) \\ u &= (u_1, u_2, \dots, u_n) \\ b &= Bu \\ A &= \frac{1}{2} \sigma \sigma^* \end{aligned}$$

tels qu'il existe Λ une matrice diagonale vérifiant $\Lambda B^* + B\Lambda + 2A = 0$, alors on a le :

THÉORÈME : *La mesure invariante p de la diffusion ($b = Bu, a = A$) a la propriété de la forme produit, c'est-à-dire :*

$$\begin{aligned} p(x) &= k \prod_{i=1}^n p_i(x_i), \quad i \in I \\ p_i(x_i) &= \exp - \frac{1}{\lambda_{ii}} \int_0^{x_i} u_i(s) ds \end{aligned}$$

où k désigne une constante de normalisation.

Avec ce théorème, dans le cas du contrôle ergodique, on est dans la même situation que pour les systèmes à dynamique découplée. Les systèmes vérifiant les hypothèses sont en général les systèmes limites du réseau de file d'attente du type Jackson ou BCMP.

4. OPTIMISATION DANS UNE CLASSE PARAMÉTRÉE DE FEEDBACKS PAR LES MÉTHODES DE TYPE MONTE CARLO

Dans le paragraphe précédent, on se proposait de calculer le feedback local optimal. Cependant, quelquefois l'information locale n'est pas bonne, de plus dans certains cas, on peut avoir une idée a priori sur la forme du feedback

optimal. Par exemple, dans le cas linéaire, on sait que le feedback est linéaire par rapport à l'état. On peut vouloir utiliser cette information de structure. Un moyen pour faire cela est de paramétriser le feedback et d'optimiser les paramètres par une méthode de Monte Carlo. Plus précisément, étant donné le problème de contrôle stochastique :

$$dX_t = b(t, X_t, u_t) dt + dW_t \quad X_t \in \mathbb{R}^n, \quad u_t \in \mathbb{R}^m \quad (4.1)$$

$$\text{Min } E \int_0^T C(t, X_t, u_t) dt .$$

On fait le changement de feedback suivant :

$$u(t) = S(t, X_t, v_t) \quad v_t \in \mathbb{R}^p \quad (4.2)$$

où

$$S : \mathbb{R}^+ \times \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^m$$

est donné.

On approxime la loi du bruit μ par :

$$\mu = \frac{1}{r} \sum_{j=1}^r \delta_{\omega_j}(\omega) \quad (4.3)$$

où les ω_j sont des trajectoires du bruit générées suivant la loi du bruit. On est alors ramené à résoudre :

$$dX_t^j = b(t, X_t^j, S(t, X_t^j, v_t)) dt + dW_t^j \quad (4.4)$$

$$\text{Min}_{v_t} \frac{1}{r} \sum_{j=1}^r \int_0^T c(t, X_t^j, S(t, X_t^j, v_t)) dt .$$

Ce problème d'optimisation est déterministe. Les méthodes classiques de gradient peuvent alors être utilisées.

En pratique, on peut raisonner sur le problème discrétisé en temps pour éviter les difficultés liées à l'existence trajectorielle d'une solution de (4.4).

L'idée du gradient stochastique, par exemple Polyak [3], Kushner-Clark [1], consiste à résoudre (4.4) récursivement par rapport à r .

Le problème (4.1), (4.2) peut se réécrire :

$$\text{Min}_{v \in V} EJ(v) \quad (4.5)$$

avec

$$J(v) = \int_0^T c(t, X_t, S(t, X_t, v_t)) dt$$

dans un cas où on est capable de calculer le gradient de $J(v)$ explicitement pour une trajectoire donnée.

L'algorithme du gradient stochastique s'écrit :

$$v_{r+1} = P(v_r - \rho_r DJ(v_r, w_r)) \quad \rho_r \in \mathbb{R}^+, \quad \forall r \in \mathbb{N} \quad (4.6)$$

$$\sum_{r \in \mathbb{N}} \rho_r = \infty, \quad \sum_{r \in \mathbb{N}} \rho_r^2 < \infty$$

où D désigne ici $\partial/\partial v$ et w_r , une génération des paramètres aléatoires, P_V la projection sur V supposée convexe.

Dans le cas convexe, on peut montrer la convergence globale et presque sûre de (4.6) vers la solution (4.5).

Plus précisément, on peut montrer par exemple :

THÉORÈME : Sous les hypothèses :

1) $v \rightarrow J(\omega, v)$ convexe $\forall \omega$

2) $\int J(\omega, v) \mu(d\omega) < \infty$

3) $\sup_{|v \in V|} |DJ(v, \omega)| \leq q$

4) $EJ(v) - \xi \geq kl^2(v)$.

5) V un ensemble borné,

on a :

$$\lim_{n \rightarrow \infty} E(l^2(v_n)) = 0$$

où

— μ désigne la loi du bruit

— ξ le coût optimal

— $l(v)$ la distance de v à l'ensemble des solutions optimales

— q et $k \in \mathbb{R}^+$.

De plus, si :

$$\rho_r = \frac{1}{kr + \frac{q^2}{y_0 k}}$$

on a :

$$E(l^2(v_r)) \leq \frac{1}{\frac{k^2}{q^2} r + \frac{1}{y_0}}$$

où

$$y_0 = E(l^2(v_0)).$$

La preuve de ce théorème peut être trouvée dans Dodu-Goursat-Hertz-Quadrat-Viot [6].

En utilisant des minoration du type Cramer-Rao, on peut montrer de plus le :

THÉORÈME (Dodu-Goursat-Hertz-Quadrat-Viot [6]) : *Sous l'hypothèse :*

- 1) *bruit fini*
- 2) $v \rightarrow J(\omega, v)$ *est deux fois différentiable et uniformément convexe, alors :*

$$E(\hat{v} - v_\mu)^{\otimes 2} \geq \frac{1}{r} H_\mu^{-1} Q_\mu H_\mu^{-1}$$

avec

$$\begin{aligned} H_\mu &= D^2 E_\mu J(v_\mu) \\ Q_\mu &= E_\mu[(DJ(v_\mu))^{\otimes 2}] \end{aligned}$$

pour $\forall \hat{v}$ une statistique non biaisée de v_μ (l'optimum) définie sur $(\Omega, \mu)^{\otimes r}$.

On remarque que $1/k$ est une estimation de H_μ^{-1} et q^2 une estimation de Q_μ et que donc dans un certain sens, l'algorithme de gradient stochastique est le meilleur possible dans la classe d'algorithmes ayant la même information, c'est-à-dire une statistique de taille r de la loi du bruit.

Il faut remarquer que les hypothèses données ici ne sont pas optimales mais sont un cadre dans lequel le théorème se démontre facilement.

Pour appliquer ce type de méthode, il faut paramétriser le feedback. Pour faire cela, nous proposons de nous donner une classe a priori de fonction de Bellman, de minimiser l'halmiltonien dans lequel on a remplacé la fonction de Bellman par un élément de la classe. On optimise ensuite l'élément de la classe. En procédant ainsi, on obtient un contrôle dont le coût est meilleur d'un ordre de grandeur que la meilleure approximation de la fonction de Bellman dans la classe donnée a priori. Ce résultat peut se montrer en remarquant que l'on a fait ainsi une itération de la méthode de Newton pour résoudre l'équation de Bellman.

5. MÉTHODES DE PERTURBATIONS RÉGULIÈRES

Dans le cas où l'intensité du bruit est faible ε , il est possible de construire un feedback affine conduisant à une erreur en ε^4 par rapport à l'optimum. Le coût de calcul du feedback est la résolution d'un problème déterministe et d'un problème linéaire quadratique, donc bien moindre que celui de la résolution de l'équation de Bellman.

Plus précisément, on se donne le problème de contrôle stochastique :

$$dX_t = f(X_t, u_t) dt + \varepsilon dW_t$$

$$y^\varepsilon(0, x) = \text{Min } E \left(\int_0^T c(X_t, u_t) dt \mid X(0) = x \right) \quad (5.1)$$

où $\varepsilon \in \mathbb{R}^+$ et petit.

On utilise la notation :

$$H(x, u, p) = pf(x, u) + c(x, u). \quad (5.2)$$

On fait l'hypothèse

$$u \rightarrow f(x, u) \text{ linéaire}$$

$$(v, c_{uu} v) \geq k |v|^2 \quad k \in \mathbb{R}^+ \quad k > 0$$

où $c_{uu} = D_{uu} c$.

On considère le problème déterministe correspondant $\varepsilon = 0$ dans (5.1). On note $x_0(t)$ et $u_0(t)$ la trajectoire optimale et le contrôle optimal correspondants.

Le problème linéaire quadratique osculateur au problème déterministe conduit à la résolution de l'équation de Riccati :

$$P' + PA + A^* P - PSP + Q = 0 \quad P(T) = 0 \quad (5.3)$$

avec

$$A = f_x - f_u H_{uu}^{-1} H_{ux}^* \quad (5.4)$$

$$S = f_u H_{uu}^{-1} f_u^* \quad (5.5)$$

$$Q = H_{xx} - H_{ux} H_{uu}^{-1} H_{ux}^* \quad (5.6)$$

où A, S, Q sont évaluées le long de la trajectoire optimale déterministe.

Pour donner un sens à (5.3)-(5.6), on fait l'hypothèse que $H_{uu} > 0$ et que $H_{xx} - H_{ux} H_{uu}^{-1} H_{ux}^* > 0$.

On construit alors le feedback affine :

$$u_f(t, x(t)) = u_0(t) + K(t) [x(t) - x_0(t)] \quad (5.7)$$

où

$$K(t) = H_{uu}^{-1} (H_{ux}^* + f_u^* P)(t). \quad (5.8)$$

On montre alors en utilisant les résultats de Cruz [5] et Fleming [8] ou les nouveaux résultats de Bensoussan [2] le :

THÉORÈME : *L'application du feedback affine (5.7) au système (5.1) conduit à une perte d'optimalité d'ordre ε^4 .*

Ce résultat justifie une pratique courante.

6. VERS UN SYSTÈME EXPERT EN CONTROLE STOCHASTIQUE

L'objet de ce paragraphe est de présenter un ensemble d'outils informatiques permettant d'automatiser l'ensemble des tâches pour réaliser une application de contrôle optimal :

- écriture automatique de programme
- démonstration automatique
- spécification de problème
- écriture de rapport.

Les outils présentés sont écrits en LISP-MACSYMA-PROLOG et peuvent tous être utilisés de façon interactive depuis MACSYMA et sont en cours de développement. Les références [33] à [44] entre autres ont servi comme outil ou base de réflexion aux résultats de ce chapitre.

6.1. Génération automatique de programme (Quadrat)

On a construit un ensemble de générateurs de programme permettant d'automatiser l'écriture des programmes fortran correspondant aux différentes méthodes indiquées ci-dessus. Ces générateurs sont en cours de test.

Dans ce qui suit, on montre un exemple de génération pour la méthode de programmation dynamique en indiquant :

- l'équation de la programmation dynamique à résoudre
- le programme d'appel au générateur
- le programme fortran généré.

$$\left[\begin{array}{l} -V + \min_{u_1, u_2} \left[u_1 \frac{\partial V}{\partial x_1} + u_2 \frac{\partial V}{\partial x_2} + u_1^2 + u_2^2 \right. \\ \left. + \left(\left(\frac{1}{2} - x_1 \right) x_2 (1 - x_2) \right)^2 + \left(\left(\frac{1}{2} - x_2 \right) x_1 (1 - x_1) \right)^2 \right. \\ \left. + 2x_2(1 - x_2) + 2x_1(1 - x_1) + x_1(1 - x_1)x_2(1 - x_2) \right] + \frac{\partial^2 V}{\partial x_1^2} + \frac{\partial^2 V}{\partial x_2^2} = 0 \\ \text{sur } x \in [0, 1] \times [0, 1] \\ V = 0 \text{ sur } \begin{cases} x_1 = 0, & x_2 = 0 \\ x_1 = 1, & x_2 = 1 \end{cases} \end{array} \right.$$

Equation de la programmation dynamique à résoudre

Pour résoudre ce problème, on écrira le programme macsyma suivant :

```
appel( ):=
fff:x1*(1-x1)*x2(1-x2),
fsc:((0.5-x1)*x2*(1-x2))**2+((0.5-x2)*x1*(1-x1))**2+2*x2*(1-x2)+2*x1*(1-x1),
condi:<<dirich.x1*x2>>, <dirich.x1*x2>>,
condim:<<dirich.(1-x1)*(1-x2)>>, <dirich.(1-x1)*(1-x2)>>,
type:<2.stat.ecriture.pasmoy.condlim.condi.condim.elp.1.difu.derive
plus.dif.<1.1>, belm.2.newto.an.u1*p1+u2*p2+u1**2+u2**2+fsc+fff.param.<>>,
hjb(type,prodyn,belman));
```

Le programme fortran généré sera alors :

```
subroutine prodyn(n1,n2,epsimp,impmax,v,ro,u,eps,nmax)
dimension v(n1,n2),u(2,n1,n2)
c
c Resolution de 1 equation de Bellman dans le cas ou:
c L etats-temps est: x1 x2
c La dynamique du systeme est decrite par l operateur
c
c
c plus(q2+q1. Minu((0.5-x1)^2 (1-x2)^2 x2^2+(1-x1) x1 (1-x2) x2
c
c
c +2 (1-x2) x2+(1-x1)^2 x1^2(0.5-x2)^2+2 (1-x1) x1+u2^2+p2 u2+u1^2
c
c +p1 u1))
c ou v designe le cout optimal
c ou pi designe sa derivee premiere par rapport a xi
c ou qi designe sa derivee seconde par rapport a xi
c Le probleme est statique
c Les conditions aux limites sont :
c x2=0 v=x1 x2
c x2=1 v=(1-x1) (1-x2)
c x1=0 v=x1 x2
c x1=1 v=(1-x1) (1-x2)
c Les nombres de points de discretisation sont: n1 n2
c x2=1 correspond a i2=n2
c x2=0 correspond a i2=1
c x1=1 correspond a i1=n1
c x1=0 correspond a i1=1
c Le taux d actualisation vaut: 1
c impmax designe le nbre maxi d iterations du systeme implicite
c epsimp designe l erreur de convergence du systeme implicite
c ro designe le pas de la resolution du systeme implicite
c par une methode iterative
c
c Minimisation par la methode de Newton de l'Hamiltonien
c L inversion de la Hessienne est faite formellement
c nmax designe le nombre maxi d iteration de la methode de Newton
c eps designe l erreur de convergence de la methode de Newton
c
h2=float(1)/(n2-1)
h1=float(1)/(n1-1)
u2=u(2,1,1)
u1=u(1,1,1)
hih2=h2**2
hih1=h1**2
h22=2*h2
h21=2*h1
```

```

nm2 = n2 - 1
nm1 = n1 - 1
do 119 i2 = 1, n2, 1
do 119 i1 = 1, n1, 1
v(i1,i2) = 0.0
119 continue
imiter = 1
113 continue
erimp = 0
do 111 i1 = 1, n1, 1
x1 = h1*(i1 - 1)
v(i1,n2) = 0
v(i1,1) = 0
111 continue
do 109 i2 = 2, nm2, 1
x2 = h2*(i2 - 1)
v(n1,i2) = 0
v(1,i2) = 0
110 continue
do 109 i1 = 2, nm1, 1
x1 = h1*(i1 - 1)
q2 = (v(i1,i2 + 1) - 2*v(i1,i2) + v(i1,i2 - 1)) / h1h2
q1 = (v(i1 + 1,i2) - 2*v(i1,i2) + v(i1 - 1,i2)) / h1h1
p2 = (v(i1,i2 + 1) - v(i1,i2 - 1)) / h22
p1 = (v(i1 + 1,i2) - v(i1 - 1,i2)) / h21
niter = 0
w0 = -1.0e + 20
101 continue
niter = niter + 1
if(niter - nmax) 102, 102, 103
103 continue
write(8,901)i1,i2
901 format('newton n a pas converge', 2 i3)
goto 104
102 continue
u1 = -p1/2.0
u2 = -p2/2.0
ww = (0.5 - x1)**2*(1 - x2)**2*x2**2 + (1 - x1)*x1*(1 - x2)*x2 + 2*(1 - x2)*x2 + (
1 1 - x1)**2*x1**2*(0.5 - x2)**2 + 2*(1 - x1)*x1 + u2**2 + p2**2 + u1**2 + p1**2
er = abs(ww - w0)
if(er - eps) 104, 104, 105
105 continue
w0 = ww
goto 101
104 continue

u(1,i1,i2) = u1
u(2,i1,i2) = u2
w0 = ww
w1 = q2 + q1
w0 = w1 + w0
w0 = w0 - v(i1,i2)
vnew = ro*w0 + v(i1,i2)
v(i1,i2) = vnew
erimp = abs(w0) + erimp
109 continue
imiter = imiter + 1
if(imiter - impmax) 116, 115, 115
116 continue
if(epsimp - erimp) 113, 112, 112

```

```

115 continue
    write(8,907)
907 format(' schema implicite n a pas converge')
112 continue
    do 117 i1=1, n1, 1
        do 117 i2=1, n2, 1
            write(8,900)i1,i2,v(i1,i2)
900 format(' v<', (i3, ', '), i3, '>>:', e14.7, ', ')
            write(8,902)i1,i2,u(1,i1,i2),u(2,i1,i2)
902 format(' u<', (i3, ', '), i3, '>:', (e14.7, ', '), e14.7, '>:')
117 continue
    return
end
    
```

6.2. Démonstration automatique (Gomez)

Afin d'automatiser autant qu'il est possible tout le processus de la modélisation à l'édition du rapport, on a développé une partie inférence qui actuellement démontre des résultats théoriques sur les équations aux dérivées partielles et qui servira par la suite également au choix de la méthode de résolution.

On utilise les notations :

$[w, m, p]$ pour l'espace $w^{m,p} = \{ f \in L^p, \dots, D^m f \in L^p \}$

$[l, p]$ pour l'espace $L^p = \left\{ f \int |f(\omega)|^p d\omega < \infty \right\}$

$[r]$ pour \mathbb{R} .

ω pour x

Ω pour O .

Nous donnons ici un exemple de fonctionnement du programme actuel. Le problème est :

$$\begin{cases} - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(a_i \frac{\partial u}{\partial x_i} \right) + u |u|^{p-2} + u = f & \text{dans } \Omega \\ u = 0 & \text{sur } \Gamma \end{cases}$$

% int représente l'intégrale $\int_{\Omega} \cdot d\omega$.

Le temps total CPU d'exécution est environ 12 s.

En conclusion, l'espace V cherché est :

$$W_0^{1,2q/q-1}(\Omega) \cap L^p(\Omega).$$

Voici la conversation avec MACSYMA telle qu'elle apparaît sur la console :

```

(c1) cogito( );
n est la dimension de l'espace
un element de R**n est note (x[1],...,x[i],...,x[n])
    
```

equation a resoudre dans omega sous la forme "A(u)=f":

$$u + \text{abs}(u)^{(p-2)} * u - \text{sum}(\text{diff}(a[i] * \text{diff}(u, x[i]), x[i]), i, 1, n) = f$$

conditions sur la frontiere gamma sous la forme "u=g" ou "diff(u,x[i,j])=g"

ou "nil" s'il n'y en a pas :

$$u=0;$$

le probleme a resoudre est :

$$- \sum_{i=1}^n \frac{d}{dx_i} \left(a_i \frac{du}{dx_i} \right) + u \text{abs}(u)^{p-2} + u = f \text{ dans omega}$$

avec : u=0 sur gamma

est-ce-correct ?

oui;

nature de u :

solution;

fonction : u

nature : solution

espace : inconnu

proprietes : nil

est-ce-correct ?

oui;

nature de p :

constante;

proprietes de p :

p >= 2;

fonction : p

nature : constante

espace : [r]

proprietes : p >= 2

est-ce-correct ?

oui;

nature de a :

coefficients;

espace de a :

[1,q];

proprietes de q :

nil;

proprietes de a :

a > 0;

fonction : a

nature : coefficient

espace : [1, q]

proprietes : a > 0

est-ce-correct ?

oui;

proprietes de f :

nil;

fonction : f

nature : smembre

espace : inconnu

proprietes : nil

est-ce-correct ?

oui;

recherche de la formulation variationnelle

$$\sum_{i=1}^n \int \left(a_i \frac{du}{dx_i} \frac{dv}{dx_i} \right) + \int (u \text{ abs}(u)^{p-2} v) + \int (uv) = \int (fv)$$

recherche de l'espace dans lequel on va chercher la solution

$$\text{on va chercher la solution dans l'espace } \left[w0, 1, \frac{2q}{q-1} \right] \text{ inter } [1, p]$$

Time = 12174 msec.

(d1) done

6.3. Entrée des données en langage semi-naturel (A. Sulem)

Les premiers essais d'adaptation d'un robot logiciel écrit par Queindec [32] pour alimenter une base de données contenant la description du problème et les actions désirées par l'utilisateur sur le problème a été entreprise.

Au problème de contrôle suivant :

Soit $X_t \in \langle 0, 1 \rangle$ un niveau d'eau satisfaisant l'équation différentielle stochastique

$$dX(t) = (a(t) - u(t)) dt + dY(t) - dZ(t) + dW(t)$$

a représente l'apport d'eau et le contrôle u ($0 \leq u \leq 1$) représente le turbiné ; Y et Z sont des processus croissants, Y (resp. Z) est strictement croissant si $x = 0$ (resp. $x = 1$).

On minimise la fonctionnelle

$$E \int_0^T (D - k(X) u)^2 dt$$

où k est un coefficient énergétique d'ordre 1 et D la demande d'électricité en puissance.

Le dialogue correspondant est :

(robot)

hello

we shall try to solve your stochastic control problem

please give the statement of the problem

= > let x be a 1 dimensional state variable denoting a water-level

state : x ; dimension of x : 1 ; physical meaning of x : water-level ;

= > let u be a 1 dimensional control

control : u ; dimension of u : 1 ;

= > x belongs to $\langle 0, 1 \rangle$

state : x ; dimension of x : 1 ; physical meaning of x : water-level ;

domain of x : $\langle 0, 1 \rangle$;

= > u denotes the turbinated-water

control : u ; dimension of u : 1 ; physical meaning of u : turbinated-water ;

```

==> u belongs to <0, 1>
control : u ; dimension of u : 1 ; physical meaning of u : turbiné-water ; domain of u : <0, 1>;
==> let a-u be the drift term
drift : a-u;
==> a denotes a water-input
physical meaning of a : water-input ;
==> z-k is the instantaneous-cost
instantaneous-cost : z-k ;
==> please cancel z-k
instantaneous-cost : z-k ; deleted
==> (z-k*u)**2 is the instantaneous-cost
instantaneous-cost : (z-k*u)**2;
==> z is an electrical-power-demand
electrical-power-demand : z;
==> k denotes an energetic-coefficient
physical meaning of k : energetic-coefficient ;
==> k is of the order 1
physical meaning of k : energetic-coefficient ; order of k : 1;
==> what is x
state : x ; dimension of x : 1 ; physical meaning of x : water-level ;
domain of x : <0, 1>;
==> T is the horizon
horizon : T;
==> what do you know
horizon : T;
electrical-power-demand : z;
instantaneous-cost : (z-k*u)**2;
drift : a-u;
control : u ; dimension of u : 1 ; physical meaning of u : turbiné-water ;
domain of u : <0, 1>;
state : x ; dimension of x : 1 ; physical meaning of x : water-level ;
domain of x : <0, 1>;
==> let l be the diffusion term
diffusion : l ;
==> for x=0 the limit condition is of the neuman type and equal to 0
state : x ; dimension of x : 1 ; physical
meaning of x : water-level ; domain of x : <0, 1>; limit condition for x=0 : 0 ; neuman type.
==> x=1 neuman 0
state : x ; dimension of x : 1 ; physical meaning of x : water-level ;
domain of x : <0, 1>; limit condition for x = 1 : 0 ; neuman type.
limit condition for x = 0 : 0 ; neuman type.
==> what do you know
diffusion : l ;
horizon : T;
electrical-power-demand : z;
instantaneous-cost : (z-k*u)**2;
drift : a-u;
control : u ; dimension of u : 1 ; physical meaning of u : turbiné-water ; domain of u : <0, 1>;
state : x ; dimension of x : 1 ; physical meaning of x : water-level ;
domain of x : <0, 1>; limit condition for x = 1 : 0 ; neuman type.
limit condition for x = 0 : 0 ; neuman type.
==> stop
good bye

```

(les lignes commençant par ==> sont écrites par l'utilisateur, les autres sont répondues par l'ordinateur).

RÉFÉRENCES

- [1] J. P. AUBIN, *Approximation of Elliptic Boundary-value Problems*, Wiley Interscience, New York, 1972.
- [2] A. BENSOUSSAN, *Méthodes de Perturbation en Contrôle Optimal* (à paraître). 1982.
- [3] A. BENSOUSSAN, J. L. LIONS, *Applications des inéquations variationnelles en contrôle stochastique*, Dunod, 1978.
- [4] P. F. CIARLET, *The finite element method for elliptic problems*, North Holland, 1978.
- [5] J. B. CRUZ, *Feedback Systems*, McGraw-Hill.
- [6] J. C. DODU, M. GOURSAT, A. HERTZ, J. P. QUADRAT, M. VIOT, *Méthodes de gradient stochastique pour l'optimisation des investissements dans un réseau électrique*, EDF Bulletin Série C, n° 2, 1981.
- [7] I. EKELAND, R. TEMAM, *Analyse convexe et problèmes variationnels*, Dunod, 1974.
- [8] W. H. FLEMING, *Control for small noise intensities*, SIAM J. Control, vol. 9, n° 3, 1971.
- [9] M. GOURSAT, J. P. QUADRAT, *Analyse numérique d'inéquations quasi variationnelles elliptiques associées à des problèmes de contrôle impulsif*, IRIA Report, 1975.
- [10] H. J. KUSHNER, *Probability methods in stochastic control and for elliptic equations*, Academic Press, 1977.
- [11] H. J. KUSHNER, D. S. CLARK, *Stochastic approximation methods for constrained and unconstrained systems*, Springer Verlag, 1978.
- [12] P. L. LIONS, B. MERCIER, *Approximation numérique des équations de Jacobi-Bellman*, RAIRO, 14, pp. 369-393, 1980.
- [13] B. T. POLYAK, *Subgradient methods : a survey of soviet research in nonsmooth optimization*, C. Lemarechal et R. Mifflin eds. Pergamon Press, 1978.
- [14] J. P. QUADRAT, *Existence de solution et algorithme de résolutions numériques de problèmes stochastiques dégénérés ou non*, SIAM Journal of Control, mars 1980.
- [15] J. P. QUADRAT, *Analyse numérique de l'équation de Bellman stochastique*, IRIA Report, 1975.
- [16] J. P. QUADRAT, *On optimal stochastic control problem of large systems Advances in filtering and optimal stochastic control*, Lecture Notes in Control and Computer Science n° 42, Springer-Verlag, 1982.
- [17] J. P. QUADRAT, M. VIOT, *Product form and optimal local feedback for multi-index Markov chains*, Allerton Conference, 1980.
- [18] F. STROOCK, S. R. S. VARADHAN, *Multidimensional Diffusion Process*, Springer Verlag, 1979.
- [19] M. GOURSAT, J. P. QUADRAT, *Numerical methods in stochastic control*, à paraître dans l'encyclopédie sur la Théorie des systèmes.
- [20] F. DELEBECQUE, P. V. KOKOTOVIC, J. P. QUADRAT, *Agregation and coherency in networks and Markov chain soumis à international journal of control*.
- [21] KEMENY-SNELL, *Finite Markov chains Van Nostrand*, Princeton, 1967.
- [22] J. P. QUADRAT, *On aggregation methods*, IFAC, conférence sur les perturbations singulières, OHRID, Yougoslavie, juillet 1982.
- [23] V. A. KOMIAKHOV, *Generalization of one proof of convergence of a process of iterative aggregation for solving systems of linear equation*, Automatika y Telemekhanika n° 7, pp. 170-173, juillet 1973.

- [24] A. BRANDT, *Multilevel adaptative computations of multigrid iterations*, journal vol. 18, n° 10, octobre 1980.
- [25] R. A. NICOLAIDES, *On the l2 convergence of an algorithm for solving finite element equations*, Math. Comp. 31 (77), pp. 892-906.
- [26] W. HACKBUSH, *Multigrids Methods lecture note in math.*, Springer-Verlag, 1982.
- [27] F. HORKAY, Thèse à paraître.
- [28] F. DELEBECQUE, *A reduction process for perturbed Markov chain*, SIAM, applied math, 1983.
- [29] J. P. QUADRAT, *Optimal control of perturbed Markov Chains. The multitime scale*, Udine, juillet 1982, Springer-Verlag.
- [31] J. LEVINE, *Non classical information and optimality in continuous time*, Dynamic team problems CDC conferences. San Antonio, décembre 1983.
- [32] C. QUEINNEC, *LISP Langage d'un autre type*, Eyrolle, 1983.
- [33] C. LIVERCY, *Théorie des programmes*, Dunod, 1978.
- [34] *Machine Lisp Manual*, MIT Press, 1982.
- [35] *MACSYMA Manual*, MIT Press, 1983.
- [36] P. GLOESS, *Logis User's Manuel*, Université de Compiègne, janvier 1984.
- [37] P. GLOESS, *Understanding Expert Systems*, Université de Compiègne, janvier 1984.
- [38] J. M. HULLOT, *CEYX Manual*, INRIA, 1984.
- [39] A. V. AHO, J. D. ULLMAN, *The theory of parsing translation and compiling*, Prentice-Hall, 1972.
- [40] M. J. GORDON, A. J. MILNER, C. P. WADSWORTH, *Edinburgh LCF. Lecture Notes in Computer Science n° 78*, 1979.
- [41] C. DELOBEL, M. ADIBA, *Bases de données et systèmes relationnels*, Dunod, 1982.
- [42] W. R. SWARTOUT, X. PLAIN, *A System for creating and explaining expert consulting programs*, Artificial Intelligence, 21-1983.
- [43] E. KANT, *On the efficient Synthesis of efficient programs*, Artificial Intelligence, 1983.
- [44] D. HOFSTADTER, *Gödel Escher Bach Vintage Bodes*, New York, 1980.