

UN PROBLEME DYNAMIQUE DE CONTROLE STOCHASTIQUE EN DIMENSION 2 RESOLU PAR UNE METHODE DE PERTURBATIONS REGULIERES

Pandore

INRIA Domaine de Voluceau BP 105 Rocquencourt 78153 Le Chesnay France

March 30, 1989

Abstract

On considère un problème dynamique de contrôle stochastique en dimension 2 dont l'état est modélisé par un processus de diffusion contrôlé défini sur \mathbf{R}^2 . Il s'agit de minimiser l'espérance mathématique d'un critère sur un horizon fini comprenant un coût intégral. Le coût optimal satisfait une équation de Bellman obtenue par la méthode de la programmation dynamique. On ne résoudra pas ici cette équation. On appliquera une méthode de perturbations régulières car le terme de diffusion σ est petit. Ceci permet de construire un feedback affine conduisant à une approximation d'ordre σ^4 de l'optimum.

Contents

1	Notations	1
2	Equation d'évolution du système	1
3	Critère à optimiser	2
4	Conditions d'optimalité	2
5	Etude de l'existence d'une solution de l'équation de Bellman	3
6	Méthode de perturbations régulières	3
6.1	Résolution du problème déterministe	4
6.1.1	Conditions d'optimalité de Pontriaguine	4
6.2	Discrétisation du problème de commande	5
6.2.1	Spécialisation à notre cas particulier	6
6.2.2	Le problème discret	7
6.2.3	Méthode de gradient	7
6.3	Equation de Riccati du problème osculateur	7
6.4	Résultats numériques	8
A	Appendix:Les programmes Fortran générés	10

1 Notations

- Variables d'état : X_1, X_2
- Variables de commande : U_1, U_2
- Coût optimal : V
- Temps : t
- Dimension de l'état : n
- I-ème variable d'état : x_i
- Opérateur dérivée par rapport au temps : ∂_0
- Opérateur dérivée par rapport à x_i : ∂_i

2 Equation d'évolution du système

On considère le processus de diffusion contrôlé défini par l'équation différentielle stochastique:

- $$dX_{1t} = (-U_1 X_1^2 - U_2 X_1 X_2)dt + 0.14dW_{1t} \tag{1}$$

- $$dX_{2t} = (-U_2 - X_2)dt + 0.14dW_{2t} \tag{2}$$

où

- $X_1 \in [-\infty, +\infty]$
- $X_2 \in [-\infty, +\infty]$
- $U_1 \in [-\infty, +\infty]$
- $U_2 \in [-\infty, +\infty]$

W_{i_t} désigne un processus de Wiener, i.e. un processus continu gaussien à accroissements indépendants.

Ce processus de diffusion est bien défini [23]. Il est la limite lorsque le pas en temps h tends vers 0 d'un processus discret markovien X_n^h vérifiant:

- $$E(X_{n+1}^h - X_n^h | \mathcal{F}_n) = h \begin{pmatrix} -U_1 X_1^2 - U_2 X_1 X_2 \\ -U_2 - X_2 \end{pmatrix} + o(h)$$

- $$E((X_{n+1}^h - X_n^h)^{\otimes 2} | \mathcal{F}_n) = 2h \begin{pmatrix} 1.00 \cdot 10^{-2} & 0 \\ 0 & 1.00 \cdot 10^{-2} \end{pmatrix} + o(h)$$

- une condition d'uniforme intégrabilité de l'accroissement $X_{n+1}^h - X_n^h$.
 \mathcal{F}_n représente la σ -algèbre générée par X_0, X_1, \dots, X_n .

3 Critère à optimiser

Il s'agit de minimiser l'espérance mathématique de la fonctionnelle:

$$J(t_0, s) = \int_{t_0}^{t_f} (X_1^2 + U_2^2 + U_1^2)_t dt + 1$$

dans la classe des feedbacks, i.e. des applications $S: [t, X_1, X_2] \mapsto [U_1, U_2]$

4 Conditions d'optimalité

On définit la fonction de Bellman V par:

$$V(t, y_1, y_2) = \min_S (e [J(t, s) | X_{1t} = y_1, X_{2t} = y_2])$$

V satisfait l'équation de la Programmation Dynamique[9,4]

$$\min_{U_1, U_2} (A(U_1, U_2)V + c(U_1, U_2)) + \partial_0 V = 0 \quad (3)$$

avec

$$\begin{aligned} A(U_1, U_2)V &= e_4 + e_3 + e_2 + e_1 \\ e_1 &= 1.00 \cdot 10^{-2} \partial_2^2 V \\ e_2 &= 1.00 \cdot 10^{-2} \partial_1^2 V \\ e_3 &= (-U_1 X_1^2 - U_2 X_1 X_2) \partial_1 V \\ e_4 &= (-U_2 - X_2) \partial_2 V \end{aligned}$$

et

$$c(U_1, U_2) = X_1^2 + U_2^2 + U_1^2 \quad (4)$$

5 Etude de l'existence d'une solution de l'équation de Bellman

D'après P.L.Lions [15], on sait qu'il existe une solution de l'équation de Bellman; la démonstration est basée sur le Principe du Maximum.

6 Méthode de perturbations régulières

On considère un problème de commande sur un horizon fini sans contrainte sur l'état du système ni sur la commande:

$$dx_t = \sigma dW_t + b(x_t, u_t)dt \quad (5)$$

$$V_\sigma(0, x_0) = \min_{u(\cdot)} \left(E \left(\int_0^{t_f} c(x_t, u_t) dt + c_f(x_{t_f}) | x_0 = y \right) \right)$$

b représente le terme de dérive, c le coût instantané, et Φ le coût final.

On applique une méthode de perturbations car le terme de diffusion σ est petit et la hessienne de l'hamiltonien par rapport à la commande n'est pas génériquement singulière. On utilise un résultat

de Bensoussan qui permet de construire un feedback affine approchant le feedback optimum avec une erreur d'ordre σ^4 . Pour cela on doit résoudre le problème de contrôle optimal déterministe correspondant :

$$\begin{aligned} \dot{x}_t &= b(x_t, u_t) \\ V(0, y) &= \min_{u(\cdot)} \left(\int_0^{t_f} c(x_t, u_t) dt + c_f(x_{t_f}) | x_0 = y \right) \end{aligned} \quad (6)$$

et le problème linéaire quadratique osculateur qui conduit à l'équation de Riccati:

$$\begin{aligned} \dot{R} - RSR + RA + Q + A^t R &= 0 \\ R(t_f) &= 0 \end{aligned} \quad (7)$$

avec

$$\begin{aligned} A &= b_x - b_u H_{uu}^{-1} H_{ux}^t \\ S &= b_u H_{uu}^{-1} b_u^t \\ Q &= H_{xx} - H_{ux} H_{uu}^{-1} H_{ux}^t \end{aligned}$$

A, S, Q sont évalués le long de la trajectoire optimale déterministe, H est l'hamiltonien du problème déterministe défini par

$$H(x, u, p) = b(x, u)p + c(x, u) \quad (8)$$

L'indice x désigne la dérivation par rapport à x . On pose

$$K(t) = -H_{uu}^{-1} (H_{ux}^t + b_u^t R) (t)$$

où R est solution de l'équation de Riccati (7).

On note u_d la commande optimale déterministe et x_d la trajectoire optimale correspondante. On montre alors en utilisant les résultats de Cruz [5] et Fleming [8] ou Bensoussan [3] le:

Theorème 1 *L'application du feedback affine*

$$u_f(t, x(t)) = K(t)(x(t) - x_d(t)) + u_d(t)$$

au système bruité (5) conduit à une perte d'optimalité d'ordre σ^4 .

6.1 Résolution du problème déterministe

Utilisons le principe de Lagrange [1] pour obtenir les conditions d'optimalité de Pontriaguine. Notons \mathcal{L} le Lagrangien défini par

$$\mathcal{L}(x(\cdot), u(\cdot), p) = \int_0^{t_f} p(b(x_t, u_t) - \dot{x}_t) dt + \int_0^{t_f} c(x_t, u_t) dt + c_f(x_{t_f})$$

p désigne la variable adjointe associée à x .

On a

$$\mathcal{L}(x(\cdot), u(\cdot), p) = \int_0^{t_f} H(x, u, p) - p\dot{x}_t dt + c_f(x_{t_f})$$

où H désigne l'hamiltonien du problème déterministe défini en (8).

6.1.1 Conditions d'optimalité de Pontriaguine

On les obtient en annulant le gradient du Lagrangien respectivement par rapport à l'état x , la variable adjointe p , et la commande u :

- Système primal

$$\frac{\partial \mathcal{L}}{\partial p} = 0$$

∂ désigne la différentielle au sens de Gâteaux.

On retrouve ainsi la contrainte

$$\dot{x} = H_p(x, u_d, p)$$

$$x_0 = y$$

- Système dual

$$\frac{\partial \mathcal{L}}{\partial x} = 0$$

On intègre par parties le terme $\int_0^{t_f} p \dot{x}_t dt$

$$\int_0^{t_f} p \dot{x}_t dt = - \int_0^{t_f} \dot{p} x_t dt + p(t_f)x(t_f) - p(t_0)x(t_0)$$

On obtient ainsi l'équation de l'état adjoint

$$\dot{p} = -H_x(x, u_d, p)$$

$$p(t_f) = \partial_x c_f(t_f)$$

- Conditions d'optimalité de l'hamiltonien

$$H_u(x, u_d, p) = 0$$

6.2 Discrétisation du problème de commande

Soit n la dimension de l'état. On considère le coût intégral comme une variable d'état. On introduit donc une variable d'état supplémentaire x_{n+1} solution de l'équation différentielle

$$\dot{x}_{n+1}(t) = c(x_t, u_t)$$

$$x_{n+1}(0) = 0$$

Le coût à optimiser se réduit à un coût sur l'état final

$$J(u(\cdot)) = \Phi(x_{t_f}) \equiv c_f(x(t_f)) + x_{n+1}(t_f)$$

On approxime l'équation d'état à l'ordre 1 :

$$x_{t+h} = x_t + hb$$

où b désigne la nouvelle dynamique :

$$b = (b_1, \dots, b_n, c)$$

h désigne le pas en temps.

Soit p_{t+h} la variable adjointe associée à x_{t+h} .

Le Lagrangien discrétisé s'écrit:

$$\mathcal{L}_h = \Phi(x_{t_f}) - \sum_{t=0}^{t_f-1} p_{t+h}(x_{t+h} - x_t - hb)$$

On définit l'hamiltonien discrétisé par:

$$\mathcal{H}_h(x, u, p) = px + hpb \tag{9}$$

Les conditions d'optimalité de Pontriaguine s'écrivent

- Système primal

$$\frac{\partial \mathcal{L}_h}{\partial p_{t+h}} = 0$$

que l'on peut écrire

$$x_{t+h} = \frac{\partial \mathcal{H}_h}{\partial p}(x_t, u_t, p_t)$$

soit

$$x_{t+h} = x_t + hb$$

$$x_0 = y$$

- Système dual

$$\frac{\partial \mathcal{L}_h}{\partial x_t} = 0$$

que l'on peut écrire

$$p_t = \frac{\partial \mathcal{H}_h}{\partial x}(x_t, u_t, p_{t+h})$$

$$p_{t_f} = \Phi_x(x_{t_f})$$

soit

$$p_t = hp_{t+h}b_x + p_{t+h}$$

- Optimalité de l'hamiltonien

$$\frac{\partial \mathcal{H}_h}{\partial u}(x_t, u_t, p_t) = 0$$

6.2.1 Spécialisation à notre cas particulier

Dans notre cas, l'hamiltonien H du problème continu vaut :

$$H(X_1, X_2, U_1, U_2, P_1, P_2) = P_1(-U_1X_1^2 - U_2X_1X_2) + P_2(-U_2 - X_2) + X_1^2 + U_2^2 + U_1^2$$

Les conditions d'optimalité de Pontriaguine du problème continu deviennent :

- Système primal

$$\dot{X}_1 = -U_1X_1^2 - U_2X_1X_2$$

$$\dot{X}_2 = -U_2 - X_2$$

avec les conditions initiales:

$$X_1(0) = 1$$

$$X_2(0) = 1$$

- Système dual

$$\dot{P}_1 = -2X_1 - P_1(-2U_1X_1 - U_2X_2)$$

$$\dot{P}_2 = P_1U_2X_1 + P_2$$

avec les conditions finales sur les variables duales:

$$P_1(t_f) = 0$$

$$P_2(t_f) = 0$$

- Optimalité de l'hamiltonien

$$0 = 2U_1 - P_1X_1^2$$

$$0 = -P_1X_1X_2 + 2U_2 - P_2$$

6.2.2 Le problème discret

L'hamiltonien du problème discret est ici

$$H_d = h \left(P_1 \left(-U_1X_1^2 - U_2X_1X_2 \right) + P_2(-U_2 - X_2) + X_1^2 + U_2^2 + U_1^2 \right) + P_2X_2 + P_1X_1$$

6.2.3 Méthode de gradient

La commande optimale se calcule en utilisant un algorithme de gradient à pas ρ adaptatif :

$$u_{n+1} = u_n - \rho \frac{dH}{du_n}$$

soit

$$U_{1n+1} = U_{1n} - \rho \left(2U_1 - P_1X_1^2 \right)$$

$$U_{2n+1} = U_{2n} - \rho \left(-P_1X_1X_2 + 2U_2 - P_2 \right)$$

L'adaptation se fait au moyen d'un algorithme de type gradient stochastique cherchant à maximiser la descente de chaque itération.

6.3 Equation de Riccati du problème osculateur

Les coefficients de l'équation de Riccati (7) valent ici:

$$A = \begin{pmatrix} -U_2 X_2 - P_1 X_1^3 - 2U_1 X_1 & -U_2 X_1 - P_1 X_1^2 X_2 \\ 0 & -1 - \frac{P_1 X_1}{2} \end{pmatrix}$$

$$S = \begin{pmatrix} \frac{X_1^2 X_2^2}{2} + \frac{X_1^4}{2} & \frac{X_1 X_2}{2} \\ \frac{X_1 X_2}{2} & 1/2 \end{pmatrix}$$

$$Q = \begin{pmatrix} -2P_1^2 X_1^2 - 2P_1 U_1 + 2 & -P_1 U_2 - P_1^2 X_1 X_2 \\ -P_1 U_2 - P_1^2 X_1 X_2 & -\frac{P_1^2 X_1^2}{2} - \frac{P_1^2 X_2^2}{2} \end{pmatrix}$$

et

$$K(t) = -e_6 e_7$$

$$e_6 = \begin{pmatrix} 0.50 & 0 \\ 0 & 0.50 \end{pmatrix}$$

$$e_7 = \begin{pmatrix} -X_1^2 & 0 \\ -X_1 X_2 & -1 \end{pmatrix} R(t) + \begin{pmatrix} -2P_1 X_1 & -P_1 X_2 \\ 0 & -P_1 X_1 \end{pmatrix}$$

Pour l'intégration numérique de l'équation de Riccati, on utilise un schéma explicite de type Euler. On a choisi ici d'écrire l'équation de Riccati correspondant au problème continu, puis de la résoudre par discrétisation.

Un autre choix possible eût été de discrétiser le problème (5), puis d'appliquer la méthode des perturbations sur le problème discret.

6.4 Résultats numériques

On représente sur les courbes suivantes la trajectoire et la commande optimales déterministes et une approximation de la trajectoire et de la commande optimales bruitées obtenue par simulation en appliquant le feedback affine sur le système bruité.

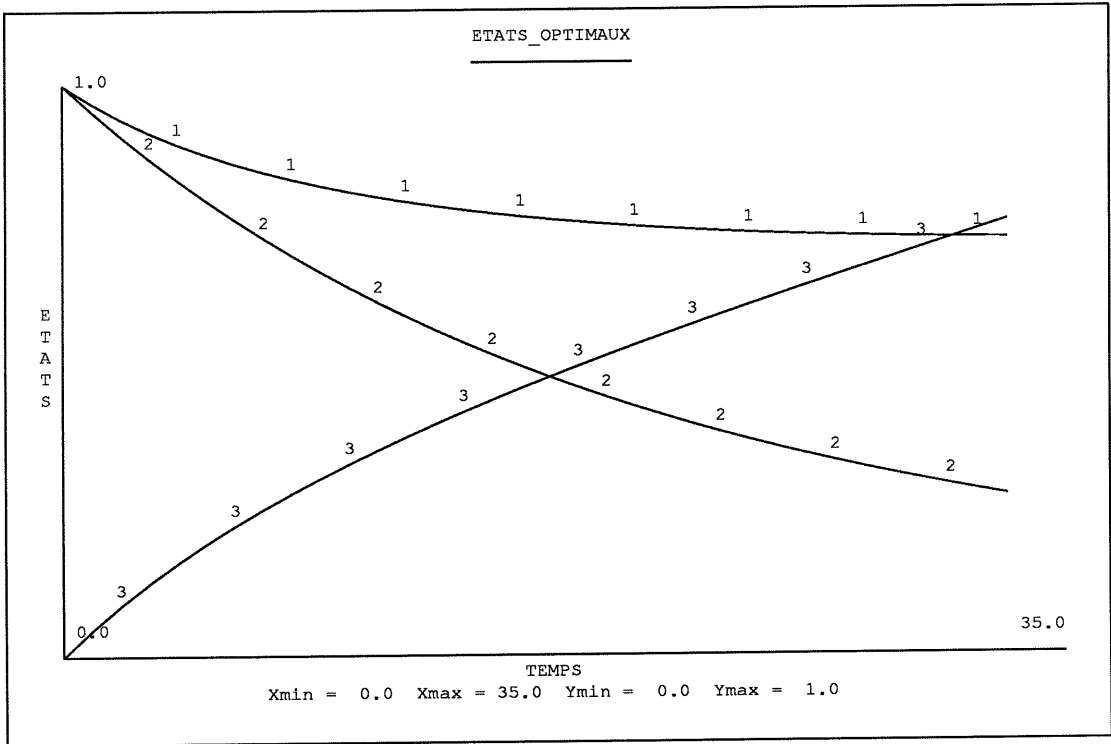


Figure 1: Trajectoire optimale déterministe. La courbe annotée par 3 représente le coût.

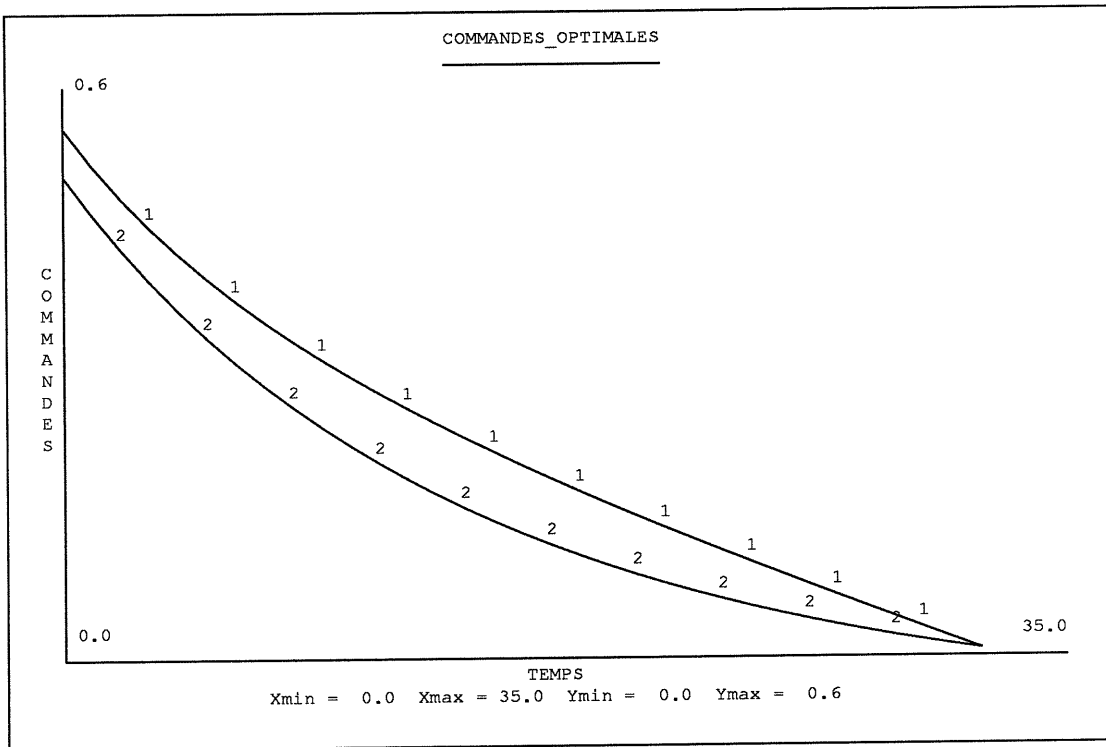


Figure 2: Commandes optimales du système déterministe.

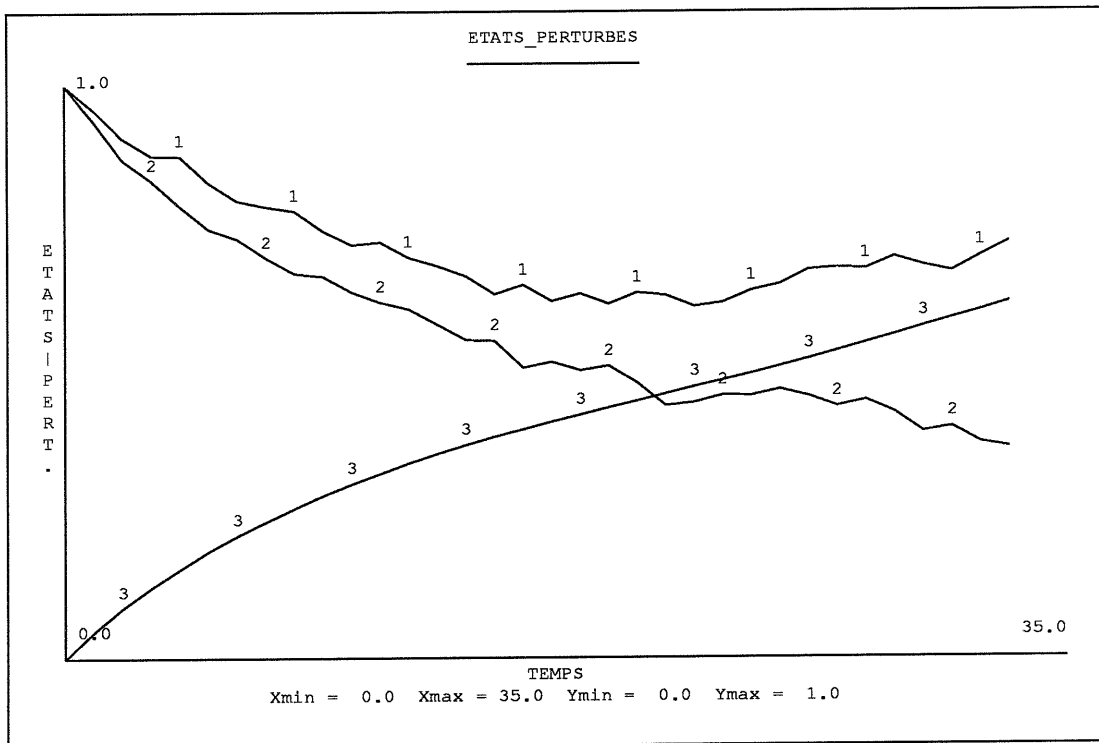


Figure 3: Approximation de la trajectoire optimale bruitée obtenue par simulation en appliquant le feedback affine sur le système bruité.

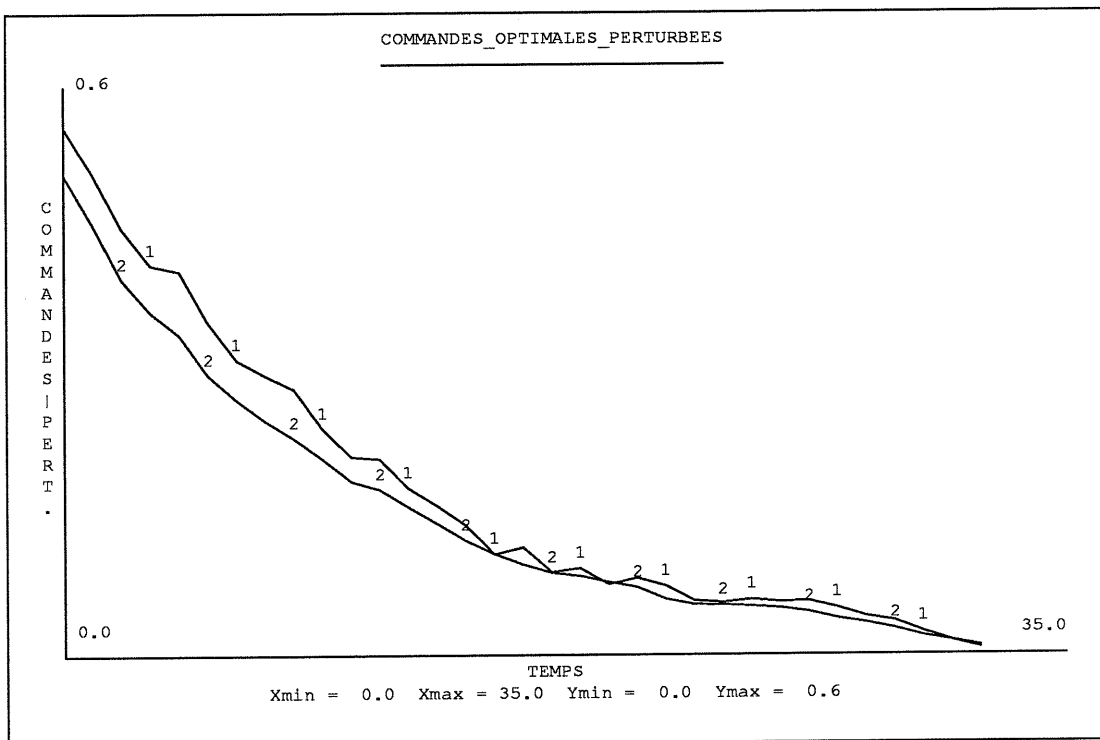


Figure 4: Approximation de la commande optimale obtenue par simulation en appliquant le feedback affine sur le système bruité.

A Appendix: Les programmes Fortran générés

```
PROGRAM M_EPS
REAL OMEGA2(34),OMEGA1(34),ETAT3(34),ETAT2(34),ETAT1(34),ETATP3(
1 34),ETATP2(34),ETATP1(34),COM2(34),COM1(34),S11(34),S21(34),S12
2 (34),S22(34),R11(34),R21(34),R12(34),R22(34),COMP2(34),COMP1(34
3 ),COMN2(34),COMN1(34),VAL3(34),VAL2(34),VAL1(34)
EXTERNAL RO,ALEA
LISPFUNCTION FLETAT1 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*2
1 0,INTEGER(1))
INTEGER NETAT1(1)
CHARACTER*20 ETAT1F
LISPFUNCTION FLETATP1 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*
1 20,INTEGER(1))
INTEGER NETATP1(1)
CHARACTER*20 ETATP1F
LISPFUNCTION FLVAL1 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*20
1 ,INTEGER(1))
INTEGER NVAL1(1)
CHARACTER*20 VAL1F
LISPFUNCTION FLETAT2 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*2
1 0,INTEGER(1))
INTEGER NETAT2(1)
CHARACTER*20 ETAT2F
LISPFUNCTION FLETATP2 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*
1 20,INTEGER(1))
INTEGER NETATP2(1)
CHARACTER*20 ETATP2F
LISPFUNCTION FLVAL2 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*20
1 ,INTEGER(1))
INTEGER NVAL2(1)
CHARACTER*20 VAL2F
LISPFUNCTION FLETAT3 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*2
1 0,INTEGER(1))
INTEGER NETAT3(1)
CHARACTER*20 ETAT3F
LISPFUNCTION FLETATP3 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*
1 20,INTEGER(1))
INTEGER NETATP3(1)
CHARACTER*20 ETATP3F
LISPFUNCTION FLVAL3 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*20
1 ,INTEGER(1))
INTEGER NVAL3(1)
CHARACTER*20 VAL3F
LISPFUNCTION FLCOM1 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*20
1 ,INTEGER(1))
INTEGER NCOM1(1)
CHARACTER*20 COM1F
```

```

LISPFUNCTION FLCOMP1 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*2
1 O,INTEGER(1))
  INTEGER NCOMP1(1)
  CHARACTER*20 COMP1F
LISPFUNCTION FLCOM2 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*20
1 ,INTEGER(1))
  INTEGER NCOM2(1)
  CHARACTER*20 COM2F
LISPFUNCTION FLCOMP2 'CLIMAX::FORT-MACS' (REAL(34),CHARACTER*2
1 O,INTEGER(1))
  INTEGER NCOMP2(1)
  CHARACTER*20 COMP2F
  ETAT1(1)=1
  ETAT2(1)=1
  ETAT3(1)=0.0
  NETAT1(1)=34
  ETAT1F= 'ETAT1'
  NETATP1(1)=34
  ETATP1F= 'ETATP1'
  NVAL1(1)=34
  VAL1F= 'VAL1'
  NETAT2(1)=34
  ETAT2F= 'ETAT2'
  NETATP2(1)=34
  ETATP2F= 'ETATP2'
  NVAL2(1)=34
  VAL2F= 'VAL2'
  NETAT3(1)=34
  ETAT3F= 'ETAT3'
  NETATP3(1)=34
  ETATP3F= 'ETATP3'
  NVAL3(1)=34
  VAL3F= 'VAL3'
  NCOM1(1)=34
  COM1F= 'COM1'
  NCOMP1(1)=34
  COMP1F= 'COMP1'
  NCOM2(1)=34
  COM2F= 'COM2'
  NCOMP2(1)=34
  COMP2F= 'COMP2'
  CALL PONTRIAG(ETAT1,ETAT2,ETAT3,VAL1,VAL2,VAL3,COM1,COM2,COM
1 N1,COMN2,0.03,34,COUTN,RO,9.0e-4,IER,100)
  CALL RICATI(ETAT1,ETAT2,VAL1,VAL2,COM1,COM2,S11,S21,S12,S22,
1 R11,R21,R12,R22,0.03,34)
  CALL SIMU(OMEGA1,OMEGA2,ETAT1,ETAT2,ETAT3,ETATP1,ETATP2,ETAT
1 P3,COM1,COM2,COMP1,COMP2,R11,R21,R12,R22,0.03,34,ALEA)
  CALL FLETAT1(ETAT1,ETAT1F,NETAT1)

```


S22(NT)=0
 R11(NT)=0.0
 R12(NT)=0.0
 R21(NT)=0.0
 R22(NT)=0.0

c-faire

```

DO 1000 IO=1,NT1
  XO=HT*(NT-IO)
  X1=ETAT1(NT-IO)
  X2=ETAT2(NT-IO)
  P1=VAL1(NT-IO)
  P2=VAL2(NT-IO)
  U1=COM1(NT-IO)
  U2=COM2(NT-IO)
  Q11=S11(NT-IO+1)
  Q12=S12(NT-IO+1)
  Q21=S21(NT-IO+1)
  Q22=S22(NT-IO+1)
  S11(NT-IO)=(-0.5*HT*Q11**2*X1**2-1.0*HT*P1*Q11*X1-0.
1 5*HT*P1**2)*X2**2+((-0.5*HT*Q11*Q21-0.5*HT*Q11*Q12)*X1-2.0*HT*Q
2 11*U2-0.5*HT*P1*Q21-0.5*HT*P1*Q12)*X2-0.5*HT*Q11**2*X1**4-2.0*H
3 T*P1*Q11*X1**3-2.0*HT*P1**2*X1**2-4.0*HT*Q11*U1*X1-2.0*HT*P1*U1
4 -0.5*HT*Q12*Q21+Q11+2.0*HT
  S12(NT-IO)=(-0.5*HT*Q11*Q12*X1**2-0.5*HT*P1*Q12*X1)*
1 X2**2+(-0.5*HT*P1*Q11*X1**2+(-0.5*HT*Q11*Q22-0.5*HT*Q12**2-0.5*
2 HT*P1**2)*X1-1.0*HT*Q12*U2-0.5*HT*P1*Q22)*X2-0.5*HT*Q11*Q12*X1*
3 *4-1.0*HT*P1*Q12*X1**3+(-1.0*HT*Q11*U2-2.0*HT*Q12*U1-0.5*HT*P1*
4 Q12)*X1-1.0*HT*P1*U2+Q12*(HT*(-0.5*Q22-1.0)+1.0)
  S21(NT-IO)=(-0.5*HT*Q11*Q21*X1**2-0.5*HT*P1*Q21*X1)*
1 X2**2+(-0.5*HT*P1*Q11*X1**2+(-0.5*HT*Q11*Q22-0.5*HT*Q21**2-0.5*
2 HT*P1**2)*X1-1.0*HT*Q21*U2-0.5*HT*P1*Q22)*X2-0.5*HT*Q11*Q21*X1*
3 *4-1.0*HT*P1*Q21*X1**3+(-1.0*HT*Q11*U2-2.0*HT*Q21*U1-0.5*HT*P1*
4 Q21)*X1-1.0*HT*P1*U2+Q21*(HT*(-0.5*Q22-1.0)+1.0)
  S22(NT-IO)=-0.5*HT*Q12*Q21*X1**2*X2**2+((-0.5*HT*P1*
1 Q21-0.5*HT*P1*Q12)*X1**2+(-0.5*HT*Q21*Q22-0.5*HT*Q12*Q22)*X1)*X
2 2-0.5*HT*Q12*Q21*X1**4-0.5*HT*P1**2*X1**2+((-1.0*HT*Q21-1.0*HT*
3 Q12)*U2-1.0*HT*P1*Q22)*X1+HT*(-0.5*Q22**2-2*Q22)+Q22
  R11(NT-IO)=0.5*Q11*X1**2+P1*X1
  R12(NT-IO)=0.5*Q12*X1**2
  R21(NT-IO)=(0.5*Q11*X1+0.5*P1)*X2+0.5*Q21
  R22(NT-IO)=0.5*Q12*X1*X2+0.5*P1*X1+0.5*Q22

```

1000 CONTINUE

c-fin-de-faire.

RETURN
 END

SUBROUTINE PONTRIAG(X1,X2,X3,P1,P2,P3,U1,U2,UN1,UN2,HT,NT,CNEW,RO,
 1 EPS,IERRO,MAXUNTILO)

```

REAL X1(NT),X2(NT),X3(NT),P1(NT),P2(NT),P3(NT),U1(NT),U2(NT),UN1(N
1 T),UN2(NT)
EXTERNAL RO

c
c
C      Resout le probleme de controle optimal defini par:
C      - l'hamiltonien discret: P3 X3
C              2
C      + HT (P1 (- U2 X1 X2 - U1 X1 ) + P2 (- X2 - U2)
C              2      2      2
C      + P3 (X1 + U2 + U1 )) + P2 X2 + P1 X1
c
C      - le cout sur l'etat final: X3 + 1
C      - les contraintes sur le controle :
C              MINF =< U1 =< INF
C              MINF =< U2 =< INF
c
C      Les parametres du sousprogramme sont:
C      - XI,1=<i=< 3 , les variables d'etat,
C      - UI,1=<i=< 2 , les variables de commande,
C      - PI,1=<i=< 3 , les variables duales,
C      - UNI necessaires a l'utilisation de dimensions variables,
C      - HT le pas en temps,
C      - NT le nombre de pas en temps,
C      - CNEW le nouveau cout dans l'iteration,
C      - EPS le test de convergence de la methode du gradient,
C      - MAXUNTILO le nombre maxi d'iterations,
C      - IERR indicateur de sortie par maxuntil0: ierr=1,
C      - RO(N) la fonction definissant le pas de la methode du
C      gradient comme fonction du numero d'iteration N.
c
      IERRO=0
c-faire
      DO 1000 IO=1,NT
          U1(IO)=1.0
          U2(IO)=1.0
      1000      CONTINUE
c-fin-de-faire.
c-until ABS(COLD - CNEW) . LT . EPS - faire-liste_until
c-initialisation
      NUNTILO=0
      CNEW=1.0e10
      WRITE(8,1008)
      NT1=NT-1
c-debut-d'iteration-d'until
      1001      CONTINUE
          NUNTILO=NUNTILO+1
c-debut-liste_until

```

```

c-faire
      DO 1002 IO=1,NT1
        XO=HT*(IO-1)
        X1(IO+1)=HT*(-U2(IO)*X1(IO)*X2(IO)-U1(IO)*X1(I
1      O)**2)+X1(IO)
        X2(IO+1)=X2(IO)+HT*(-X2(IO)-U2(IO))
        X3(IO+1)=X3(IO)+HT*(X1(IO)**2+U2(IO)**2+U1(IO)
1      **2)
1002      CONTINUE
c-fin-de-faire.
      P1(NT)=0
      P2(NT)=0
      P3(NT)=1

c-faire
      DO 1003 IO=1,NT1
        XO=HT*(NT-IO)
        P1(NT-IO)=HT*(2*X1(NT-IO)*P3(NT-IO+1)+(-U2(NT-
1      IO)*X2(NT-IO)-2*U1(NT-IO)*X1(NT-IO))*P1(NT-IO+1))+P1(NT-IO+1)
        P2(NT-IO)=P2(NT-IO+1)+HT*(-P2(NT-IO+1)-U2(NT-I
1      O)*X1(NT-IO)*P1(NT-IO+1))
        P3(NT-IO)=P3(NT-IO+1)
1003      CONTINUE
c-fin-de-faire.
c-faire
      DO 1004 IO=1,NT1
        XO=HT*(IO-1)
        UN1(IO)=U1(IO)-HT*(2*U1(IO)*P3(IO+1)-X1(IO)*
1      *2*P1(IO+1))*RO(NUNTILO)
        UN2(IO)=U2(IO)-HT*(2*U2(IO)*P3(IO+1)-P2(IO+1
1      )-X1(IO)*X2(IO)*P1(IO+1))*RO(NUNTILO)
        U1(IO)=UN1(IO)
        U2(IO)=UN2(IO)
1004      CONTINUE
c-fin-de-faire.
      COLD=CNEW
      CNEW=X3(NT)+1
      WRITE(8,1009) CNEW

c-fin-liste_until.
c-tests-de-sortie-d'until
      IF(ABS(COLD-CNEW) . LT . EPS)GOTO 1006
      IF(NUNTILO.gt.MAXUNTILO)GOTO 1005
c-reiterer-until
      GOTO 1001
c-sortie-d'until-depassement-du-maximum-d'iterations
1005      CONTINUE
      WRITE(8,1007)
      IERRO=1
1006      CONTINUE

```


c-fin-d'until.

```
      WRITE(8,1010)
1007 FORMAT(' MAXUNTILO' )
1008 FORMAT('COUT_PONTRI:[')
1009 FORMAT(' ',e12.5,',')
1010 FORMAT(' []]$')
      RETURN
      END
```

```
      SUBROUTINE SIMU(OMEGA1,OMEGA2,X1,X2,X3,XP1,XP2,XP3,U1,U2,UP1,UP2,R
1  11,R21,R12,R22,HT,NT,ALEA)
      REAL XP1(NT),XP2(NT),XP3(NT),X1(NT),X2(NT),X3(NT),UP1(NT),UP2(NT),
1  U1(NT),U2(NT),OMEGA1(NT),OMEGA2(NT),R11(NT),R21(NT),R12(NT),R22
2  (NT)
      REAL*8 RAN
      EXTERNAL RO,ALEA
      NT1=NT-1
      RAN=0.7156734
      XP1(1)=X1(1)
      XP2(1)=X2(1)
      XP3(1)=X3(1)
```

c-faire

```
      DO 1000 IO=1,NT1
          XO=HT*(IO-1)
          CALL ALEA(RAN)
          OMEGA1(IO)=3.4641016*(RAN-0.5)
          CALL ALEA(RAN)
          OMEGA2(IO)=3.4641016*(RAN-0.5)
          UP1(IO)=R12(IO)*(XP2(IO)-X2(IO))+R11(IO)*(XP1(IO)-X1
1  (IO))+U1(IO)
          UP2(IO)=R22(IO)*(XP2(IO)-X2(IO))+R21(IO)*(XP1(IO)-X1
1  (IO))+U2(IO)
          XP1(IO+1)=HT*(-UP2(IO)*XP1(IO)*XP2(IO)-UP1(IO)*XP1(I
1  O)**2+0.1*OMEGA1(IO)/HT**0.5)+XP1(IO)
          XP2(IO+1)=XP2(IO)+HT*(-XP2(IO)-UP2(IO)+0.1*OMEGA2(IO
1  )/HT**0.5)
          XP3(IO+1)=XP3(IO)+HT*(XP1(IO)**2+UP2(IO)**2+UP1(IO)*
1  *2)
1000      CONTINUE
c-fin-de-faire.
      RETURN
      END
```

References

[1] ALEXEEV V., TIKHOMIROV V., FOMINE V. (1982). Commande optimale. Edition MIR.

- [2] BELLMAN R. (1971). Introduction to the mathematical theory of control processes. Academic press.
- [3] BENSOUSSAN, A. (1982). Méthodes de Perturbations en Contrôle Optimal.
- [4] BENSOUSSAN A., LIONS J.L. (1978). Applications des inéquations variationnelles en contrôle stochastique. Dunod.
- [5] CRUZ, J.B. Feedback Systems. McGraw-Hill.
- [6] DERMAN C. (1970) Finite state markovian decision processes. Academic Press
- [7] DODU, J.C.-GOURSAT, M.-HERTZ, A.-QUADRAT, J.P.-VIOT, M. (1981). Méthodes de gradient stochastique pour l'optimisation des investissements dans un réseau électrique. EDF Bulletin Serie C, n0 2.
- [8] FLEMING, W.H. (1971). Control for small noise intensities. SIAM J. Control, vol.9, #3.
- [9] FLEMING W.H., RISHEL R. (1975). Deterministic and Stochastic Optimal Control. Springer Verlag, New York.
- [10] KIEFER, E.-WOLFOWITZ, J. (1952). Stochastic estimation of the maximum of a regression function. Ann. Math. Statist., 23, 3.
- [11] KRYLOV N.V. (1980). Controlled diffusion processes. Springer Verlag.
- [12] KUSHNER, H.J.-CLARK, D.S. (1978). Stochastic approximation methods for constrained and unconstrained systems. Springer Verlag.
- [13] LIONS J.L. (1969). Quelques méthodes de résolution des problèmes aux limites non linéaires. Dunod Gauthier-Villars. Paris.
- [14] LIONS J.L. (1966). Contrôle optimal des systèmes gouvernés par des équations aux dérivées partielles. Dunod.
- [15] LIONS P.L. (1982). Generalized Solutions of Hamilton-Jacobi Equations. Research Notes in Mathematics, 69, Pitman.
- [16] POLYAK, B.T. (1976). Convergence and convergence rate of iterative stochastic algorithms. Automatica i Telemekhanika, Vol.12, pp.83-94.
- [17] POLYAK, B.T. (1978). Subgradient methods; a survey of soviet research in nonsmooth optimization. C.Lemarechal and K.Mifflin eds. Pergamon Press.
- [18] POLYAK, B.T.-TSYPKIN, Y.Z. (1973). Pseudogradient adaptation and training algorithms. Automatica i Telemekhanika, Vol.3.
- [19] PONTRYAGIN L.S., BOLTYANSKII V.G.R., GAMKRELIDZE V., MISCHENKO E.F. (1964). The mathematical theory of optimal processes. Pergamon Press. Edition Mir en français 1974.
- [20] QUADRAT J.P. (1975). Analyse numérique de l'équation de Bellman stochastique. IRIA Report.
- [21] ROBBINS, H.-MONRO, H.S. (1951). A stochastic approximation method. Ann. Math. Statist., 22, pp 400-407.

- [22] ROSS S.(1983). Introduction to stochastic dynamic programming. Academic Press.
- [23] STROOCK F., VARADHAN S.R.S.(1979). Multidimensional Diffusion Process. Springer Verlag.