La commande prédictive des systèmes non linéaires

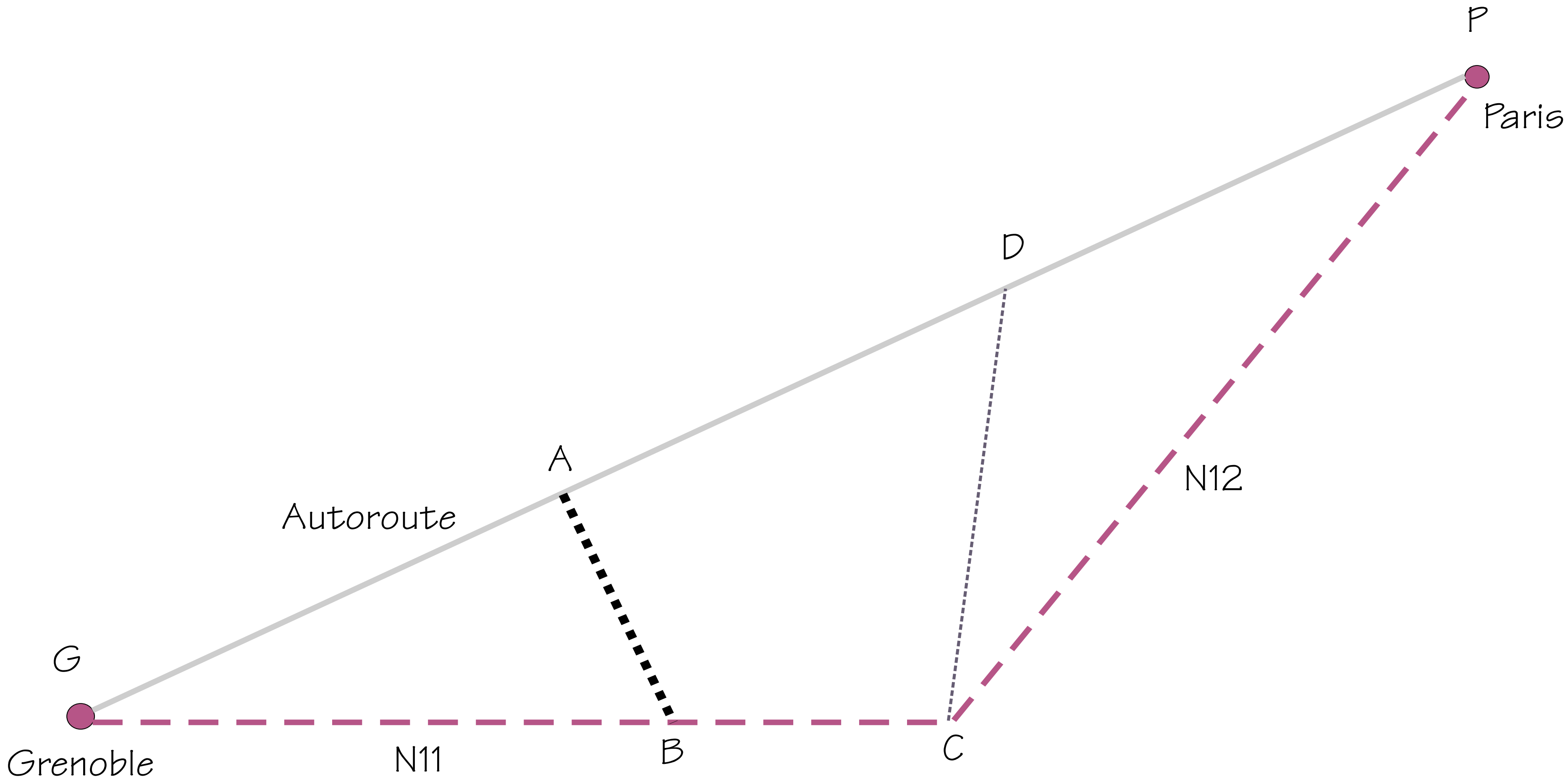
(Premier contact ...)

Mazen ALAMIR

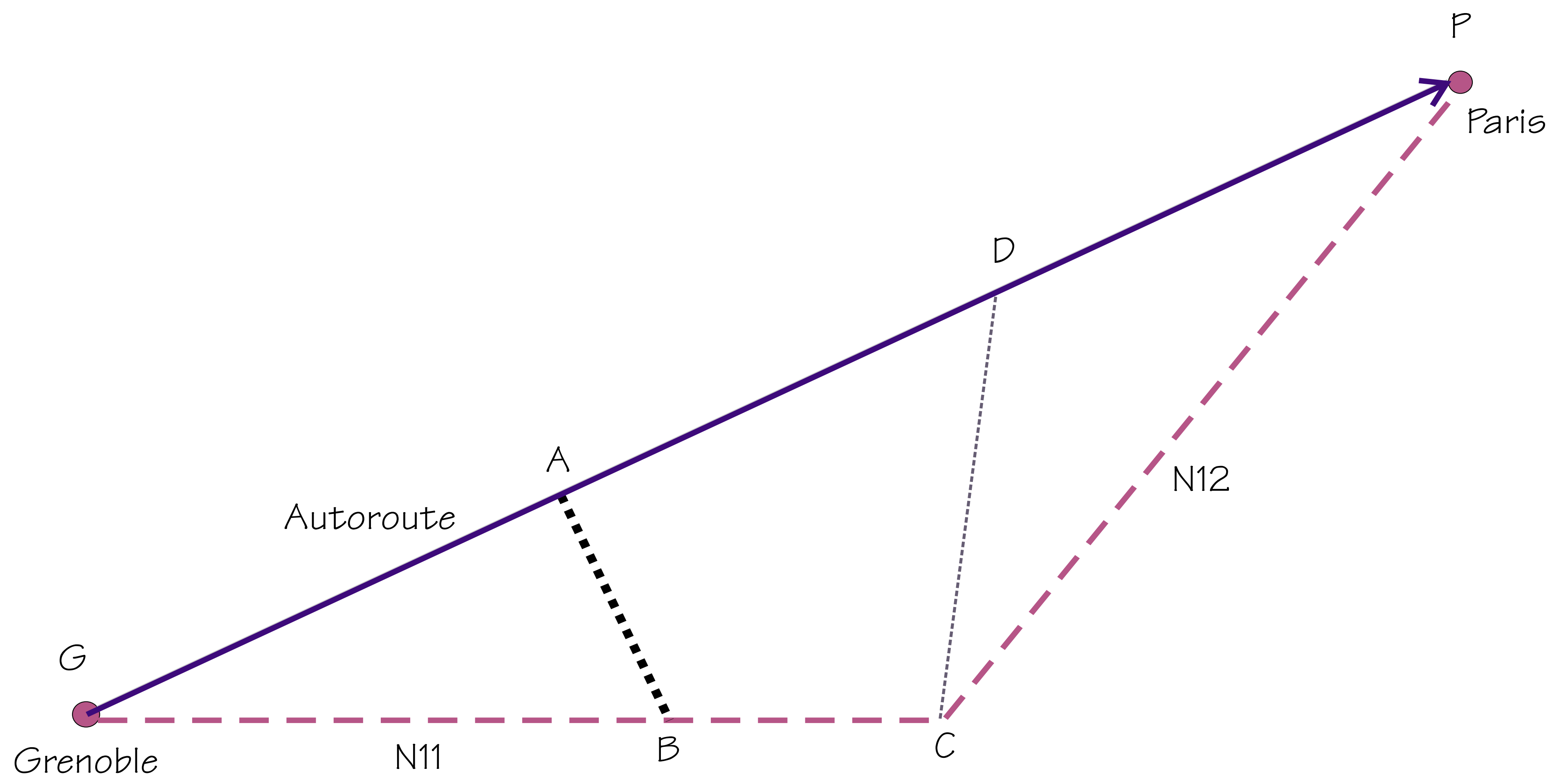
(Chargé de Recherche au CNRS)
Laboratoire d'Automatique de Grenoble

Réunion GT Commande prédictive non linéaire

Principe

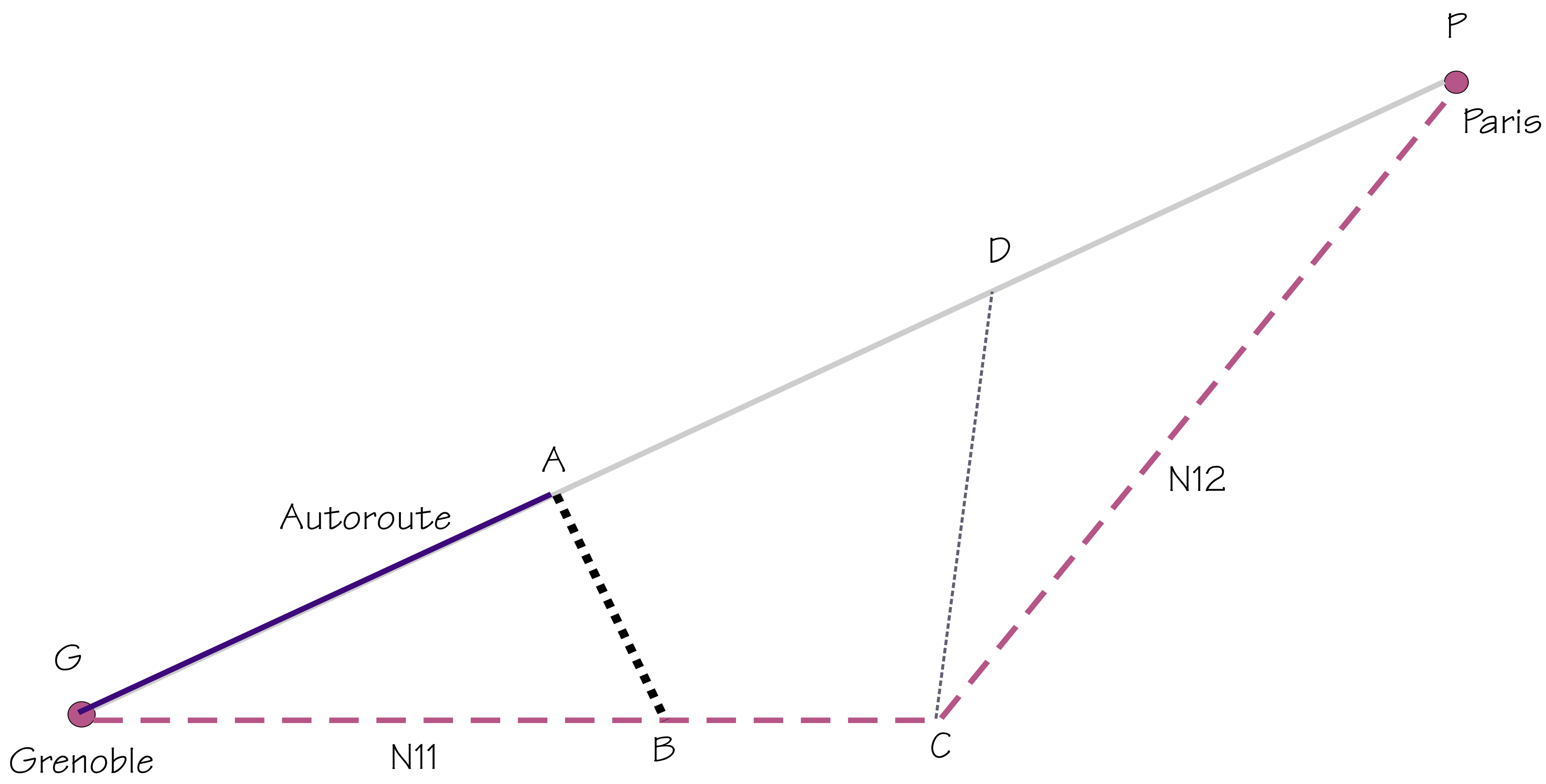


Stratégie(G) : $\longrightarrow$ G-A-D-P



Stratégie(G) : $\longrightarrow$ G-A-D-P

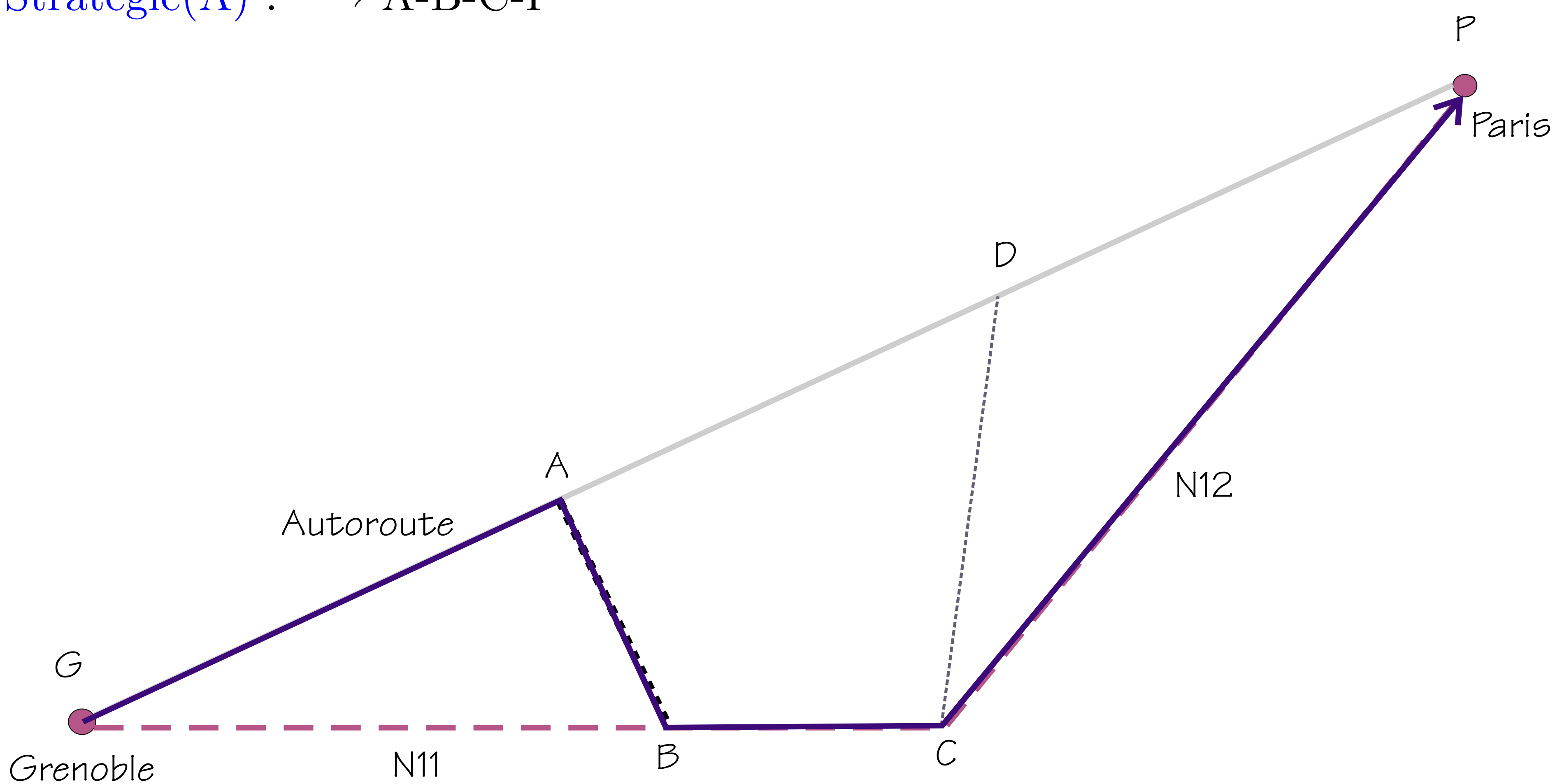
Arrivée à A : Bouchon



Stratégie(G) : $\longrightarrow$ G-A-D-P

Arrivée à A : Bouchon

Stratégie(A) : $\longrightarrow$ A-B-C-P

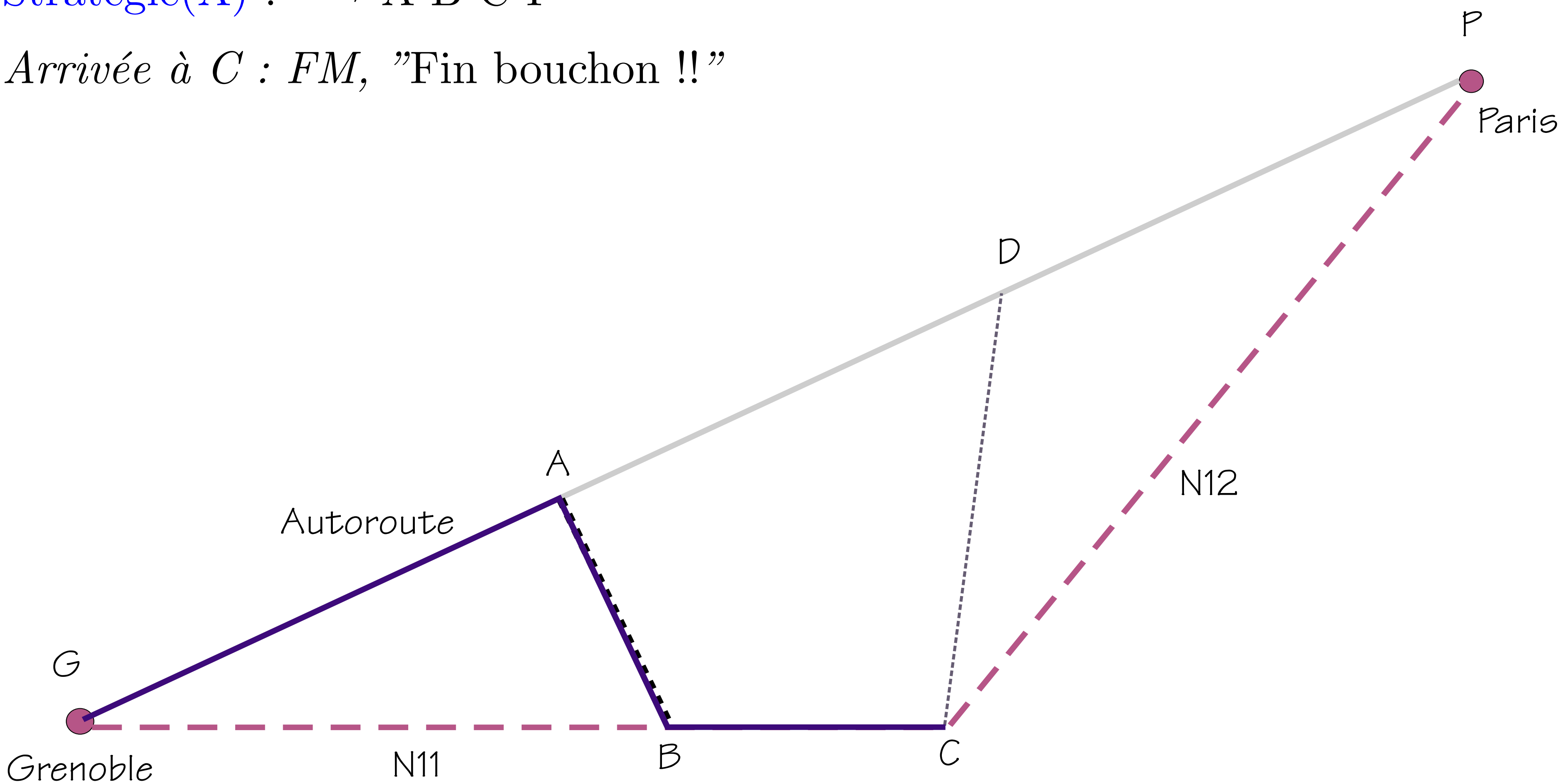


Stratégie(G) : $\longrightarrow$ G-A-D-P

Arrivée à A : Bouchon

Stratégie(A) : $\longrightarrow$ A-B-C-P

Arrivée à C : FM, "Fin bouchon !!"



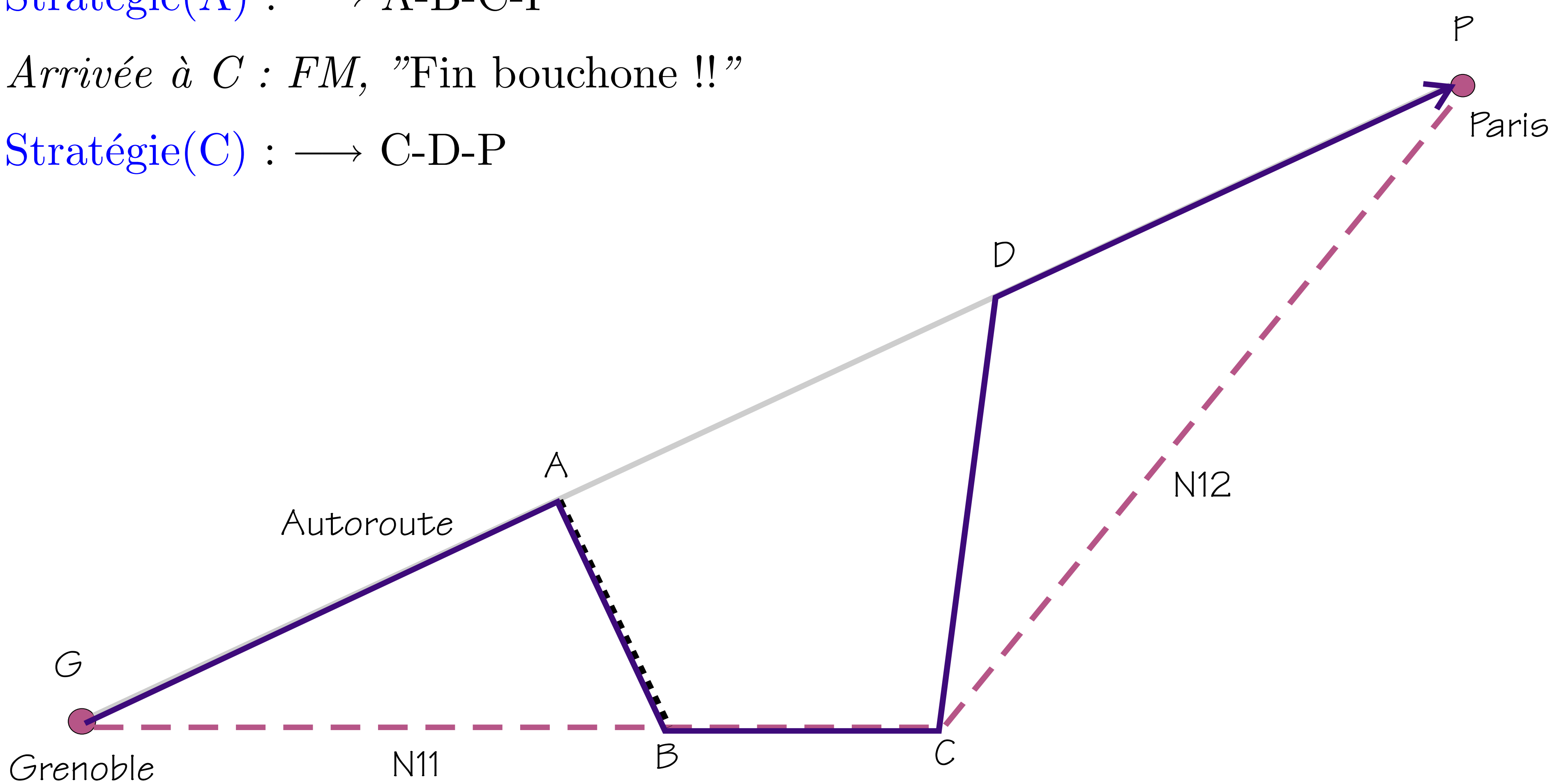
Stratégie(G) : $\longrightarrow$ G-A-D-P

Arrivée à A : Bouchon

Stratégie(A) : $\longrightarrow$ A-B-C-P

Arrivée à C : FM, "Fin bouchone !!"

Stratégie(C) : $\longrightarrow$ C-D-P



Stratégie(G) : $\longrightarrow$ G-A-D-P

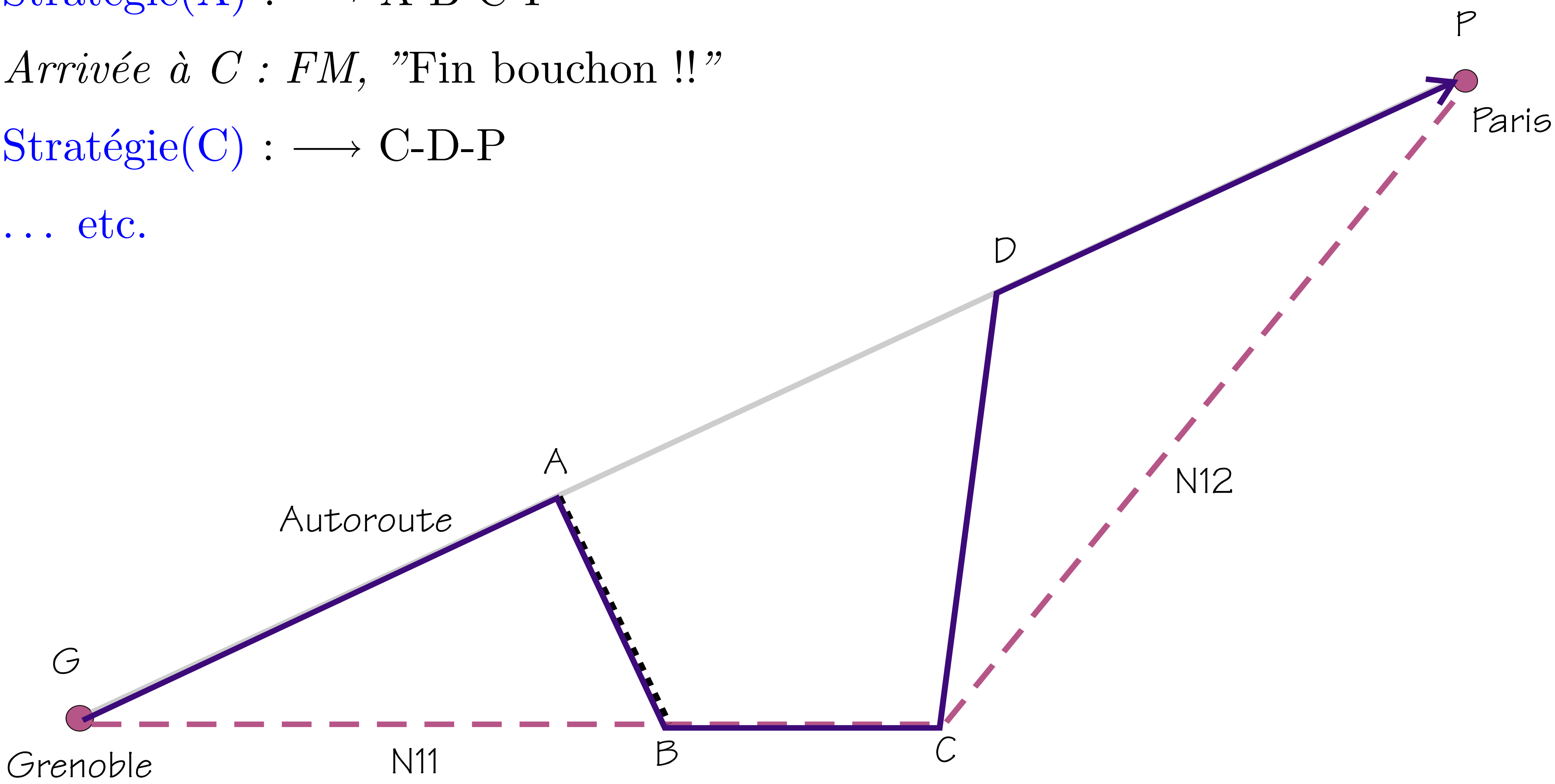
Arrivée à A : Bouchon

Stratégie(A) : $\longrightarrow$ A-B-C-P

Arrivée à C : FM, "Fin bouchon !!"

Stratégie(C) : $\longrightarrow$ C-D-P

... etc.



En automatique (Lee et Markus 1967):

> One technique for obtaining a feedback controller synthesis from the knowledge of open-loop controller is to measure the current control process state and then compute very rapidly for the open-loop control function. The first portion of this function is then used during a short time-interval, after which a new measurement of the process state is made and a new open-loop control function is computed. The procedure is then repeated.

Soit le système dynamique suivant

$$x(k+1) = f(x(k), u(k)) \quad ; \quad x(k) \in \mathbb{R}^n \quad ; \quad u(k) \in \mathbb{R}^m$$

Soit le système dynamique suivant

$$x(k+1) = f(x(k), u(k)) \quad ; \quad x(k) \in \mathbb{R}^n \quad ; \quad u(k) \in \mathbb{R}^m \tag{1}$$

Pour toute séquence de commandes $\tilde{u} := (u_0, u_1, \dots) \in (\mathbb{R}^m)^{\mathbb{N}}$ on note

$$X(k; k_0; x_0; \tilde{u})$$

la solution de (1) partant de l'état initial (k_0, x_0) et subissant la séquence de commande $\tilde{u}$.

Soit le système dynamique suivant

$$x(k+1) = f(x(k), u(k)) \quad ; \quad x(k) \in \mathbb{R}^n \quad ; \quad u(k) \in \mathbb{R}^m$$

Pour toute séquence de commandes $\tilde{u} := (u_0, u_1, \dots) \in (\mathbb{R}^m)^{\mathbb{N}}$ on note

$$X(k; k_0; x_0; \tilde{u})$$

la solution de (1) partant de l'état initial (k_0, x_0) et subissant la séquence de commande $\tilde{u}$.

Pour (k, x) donnés, soit le critère d'optimisation sur la séquence $\tilde{u}$

$$V(x, k, \tilde{u}) := F(x(k+N)) + \sum_{i=k}^{k+N-1} l(x(i), u(i)) \quad ; \quad \tilde{u} \in U \tag{2}$$

où $x(i) = X(i; k; x; \tilde{u})$ et $U \subset (\mathbb{R}^m)^N$

Soit la séquence

$$\hat{u}(x,k) := (\underbrace{\hat{u}_0(x,k)}_{\in\mathbb{R}^m}, \dots, \underbrace{\hat{u}_{N-1}(x,k)}_{\in\mathbb{R}^m}) \in U \subset (\mathbb{R}^m)^N$$

Soit la séquence

$$\hat{u}(x,k) := (\underbrace{\hat{u}_0(x,k)}_{\in\mathbb{R}^m}, \dots, \underbrace{\hat{u}_{N-1}(x,k)}_{\in\mathbb{R}^m}) \in U \subset (\mathbb{R}^m)^N$$

définie par

$$\hat{u}(k,x) := Arg \min_{\tilde{u}\in U} \quad V(x,k,\tilde{u})$$

Soit la séquence

$$\hat{u}(x,k) := (\underbrace{\hat{u}_0(x,k)}_{\in\mathbb{R}^m}, \dots, \underbrace{\hat{u}_{N-1}(x,k)}_{\in\mathbb{R}^m}) \in U \subset (\mathbb{R}^m)^N$$

définie par

$$\hat{u}(k,x) := Arg \min_{\tilde{u}\in U} \quad V(x,k,\tilde{u})$$

La loi de commande (en boucle fermée) à horizon glissant est alors donnée par

$$\boxed{u(k) := \hat{u}_0(k, x(k)) \in \mathbb{R}^m}$$

En résumé, pour définir une commande à horizon glissant, il faut avoir

En résumé, pour définir une commande à horizon glissant, il faut avoir

✓ Un "*simulateur*" du système à commander

En résumé, pour définir une commande à horizon glissant, il faut avoir

- ✓ Un "*simulateur*" du système à commander
- ✓ Un "*critère d'optimisation*" exprimant les objectifs

En résumé, pour définir une commande à horizon glissant, il faut avoir

- ✓ Un "*simulateur*" du système à commander
- ✓ Un "*critère d'optimisation*" exprimant les objectifs
- ✓ Un ensemble de "*contraintes*" sur la commande et/ou sur l'état.

Prenons l'exemple très simple suivant

$$\begin{aligned} x_1(k+1) &= x_1(k) + (1 + x_2^2(k))u(k) \\ x_2(k+1) &= 1.5x_2(k) - x_1(k)e^{u(k)} \end{aligned}$$

l'exemple très simple suivant

$$
\begin{aligned}
x_1(k+1) &= x_1(k) + (1 + x_2^2(k))u(k) \\
x_2(k+1) &= 1.5x_2(k) - x_1(k)e^{u(k)}
\end{aligned}
$$

✓ Instable en boucle ouverte.

l'exemple très simple suivant

$$\begin{aligned} x_1(k+1) &= x_1(k) + (1 + x_2^2(k))u(k) \\ x_2(k+1) &= 1.5x_2(k) - x_1(k)e^{u(k)} \end{aligned}$$

- ✓ Instable en boucle ouverte.
- ✓ le lieu des états d'équilibre est donné par

$$\mathcal{E}_{eq} := \left\{ x_\alpha := \begin{pmatrix} \alpha \\ 2\alpha \end{pmatrix} \quad ; \quad \alpha \in \mathbb{R} \right\}$$

l'exemple très simple suivant

$$\begin{aligned} x_1(k+1) &= x_1(k) + (1 + x_2^2(k))u(k) \\ x_2(k+1) &= 1.5x_2(k) - x_1(k)e^{u(k)} \end{aligned}$$

- ✓ Instable en boucle ouverte.
- ✓ le lieu des états d'équilibre est donné par

$$\mathcal{E}_{eq} := \left\{ x_\alpha := \begin{pmatrix} \alpha \\ 2\alpha \end{pmatrix} \quad ; \quad \alpha \in \mathbb{R} \right\}$$

Il nous reste à définir un critère d'optimisation et un ensemble de contraintes

Définition du critère

Objectif Changement d'état stationnaire

$$\alpha_1 = 0 \quad \longrightarrow \quad \alpha_2 = 1$$

Définition du critère

Objectif Changement d'état stationnaire

$$\alpha_1 = 0 \quad \longrightarrow \quad \alpha_2 = 1$$

Soit alors le critère d'optimisation

$$V(x, k, \tilde{u}) := F(x(k+N)) + \sum_{i=k}^{k+N-1} l(x(i), u(i))$$

Définition du critère

Objectif Changement d'état stationnaire

$$\alpha_1 = 0 \quad \longrightarrow \quad \alpha_2 = 1$$

Soit alors le critère d'optimisation

$$V(x, k, \tilde{u}) := F(x(k+N)) + \sum_{i=k}^{k+N-1} l(x(i), u(i))$$

avec

$$F(.) = 0 \quad ; \quad l(x, u) = \quad \| x - x_{\alpha_2} \|^2_{Q_x} + ru^2$$

Définition des contraintes

Pour obtenir un problème d'optimisation de dimension N_u, on définit

$$U_{N_u} := \left\{ \tilde{u} \in (\mathbb{R}^1)^N \quad | \quad \tilde{u}_j = 0 \quad \forall j \geq N_u - 1 \right\}$$

Définition des contraintes

Pour obtenir un problème d'optimisation de dimension N_u, on définit

$$U_{N_u} := \left\{ \tilde{u} \in (\mathbb{R}^1)^N \quad | \quad \tilde{u}_j = 0 \quad \forall j \geq N_u - 1 \right\}$$

Plus clairement

$$\left\{ \tilde{u} \in U_{N_u} \right\} \quad \Leftrightarrow \left\{ \tilde{u} = (\tilde{u}_0, \ldots, \tilde{u}_{N_u - 1}, 0, \ldots, 0)^T \right\}$$

Définition des contraintes

Pour obtenir un problème d'optimisation de dimension N_u, on définit

$$U_{N_u} := \left\{ \tilde{u} \in (\mathbb{R}^1)^N \quad | \quad \tilde{u}_j = 0 \quad \forall j \geq N_u - 1 \right\}$$

Plus clairement

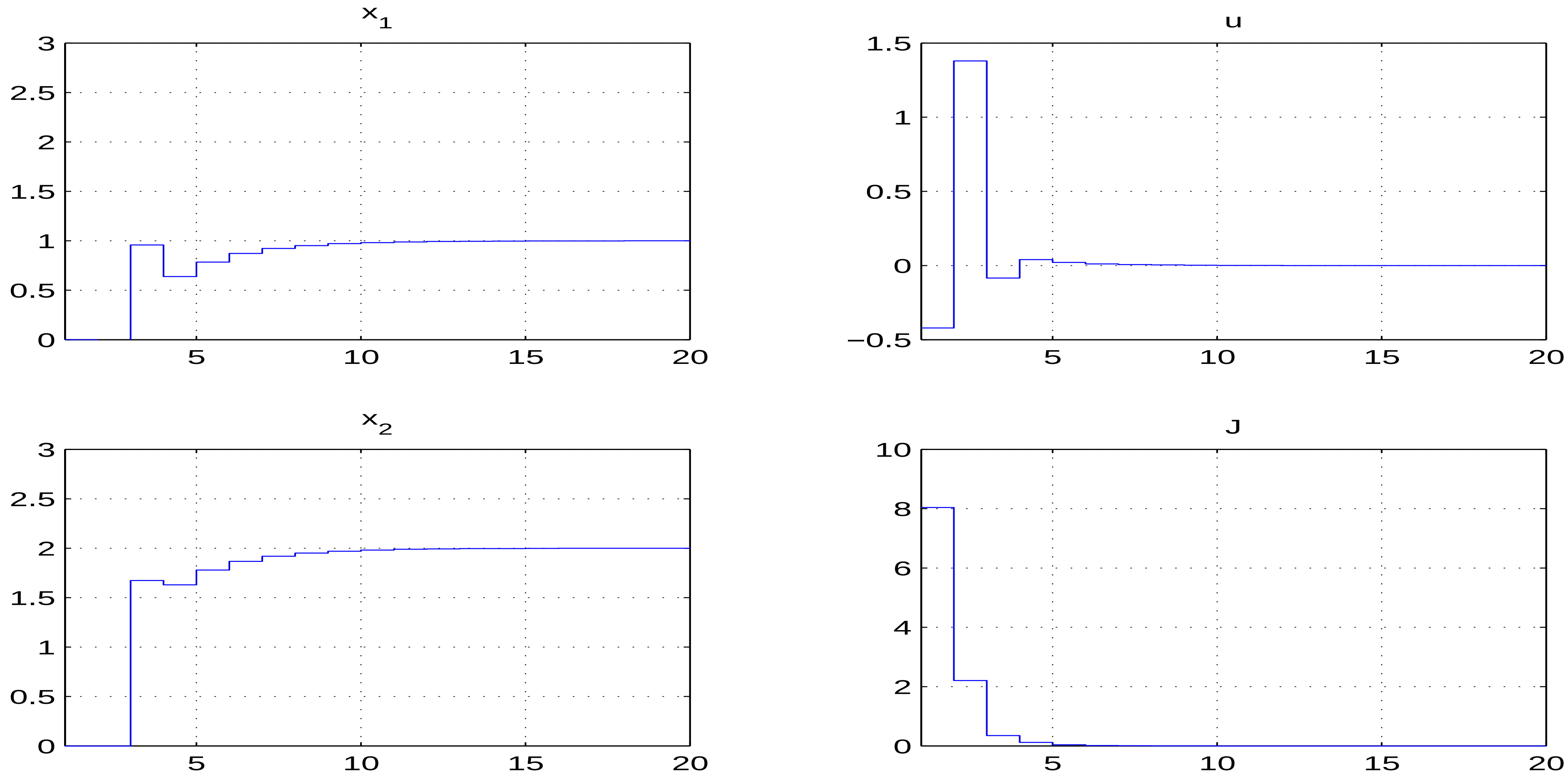
$$\left\{ \tilde{u} \in U_{N_u} \right\} \quad \Leftrightarrow \left\{ \tilde{u} = (\tilde{u}_0, \dots, \tilde{u}_{N_u - 1}, 0, \dots, 0)^T \right\}$$

Il en résulte que

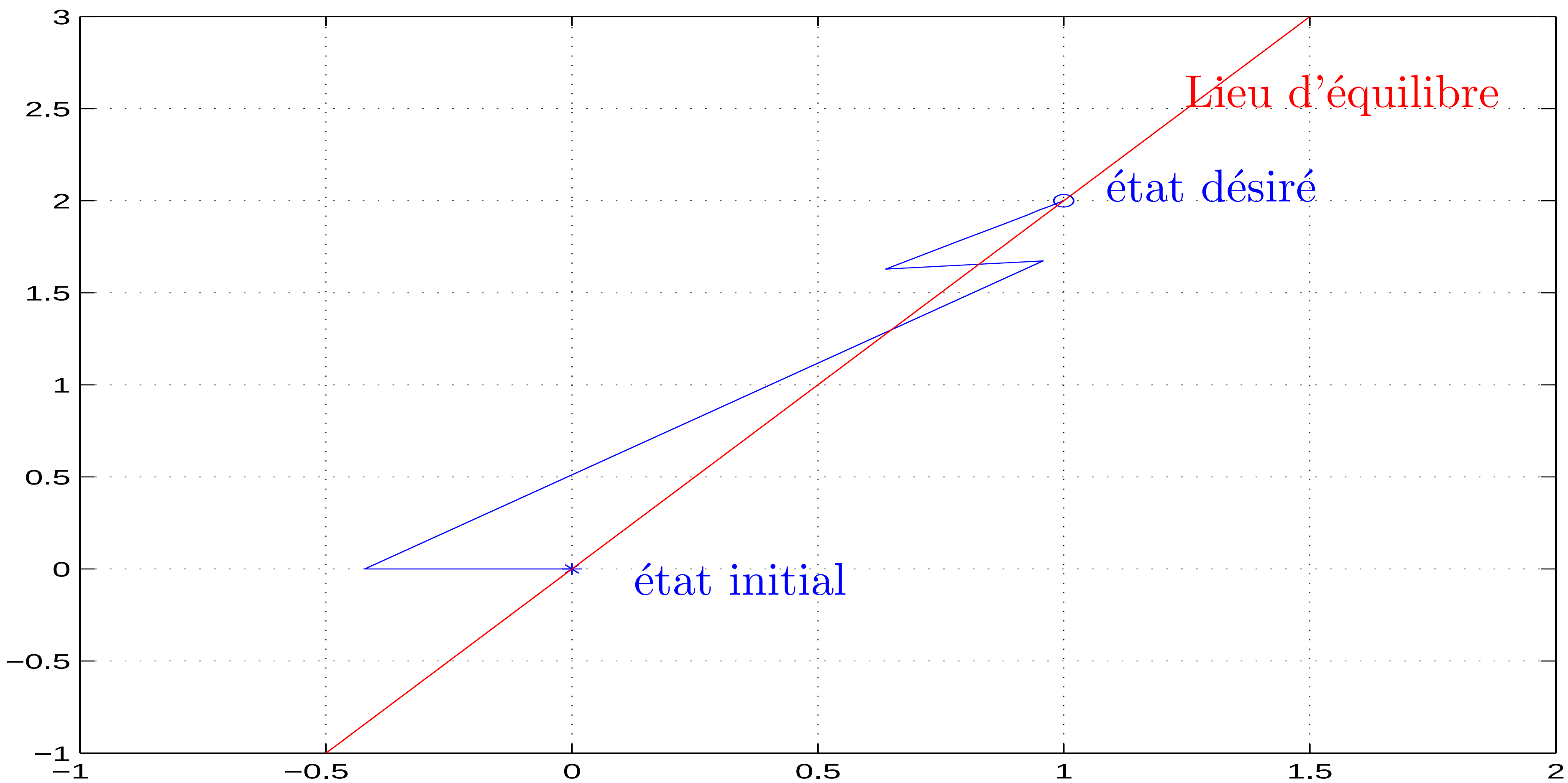
$$U_{N_u} \sim (\mathbb{R}^1)^{N_u}$$

$$\boxed{\text{La variable de décision est de dimension } N_u \text{ mais la pondération porte bien sur un horizon de longueur } N.}$$

Test 1: $N = 2,\ N_u = 2,\ r = 1,\ q_x = 1$



TEST 1: $N = 2,\ N_u = 2,\ r = 1,\ q_x = 1$

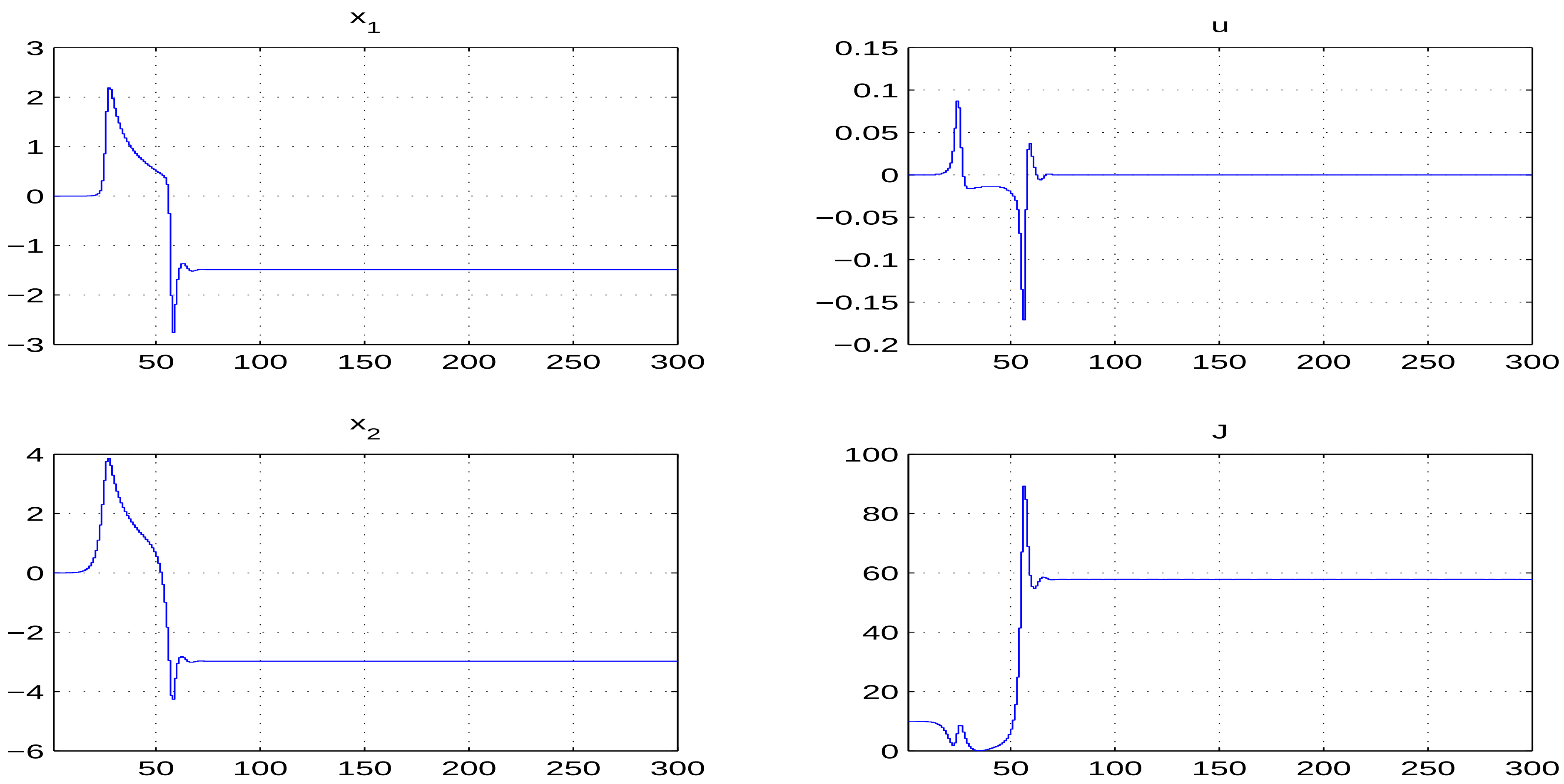


Supposons que l'on ait estimé que le pic de commande est très important et que l'on veuille que u ne dépasse pas 0.1.

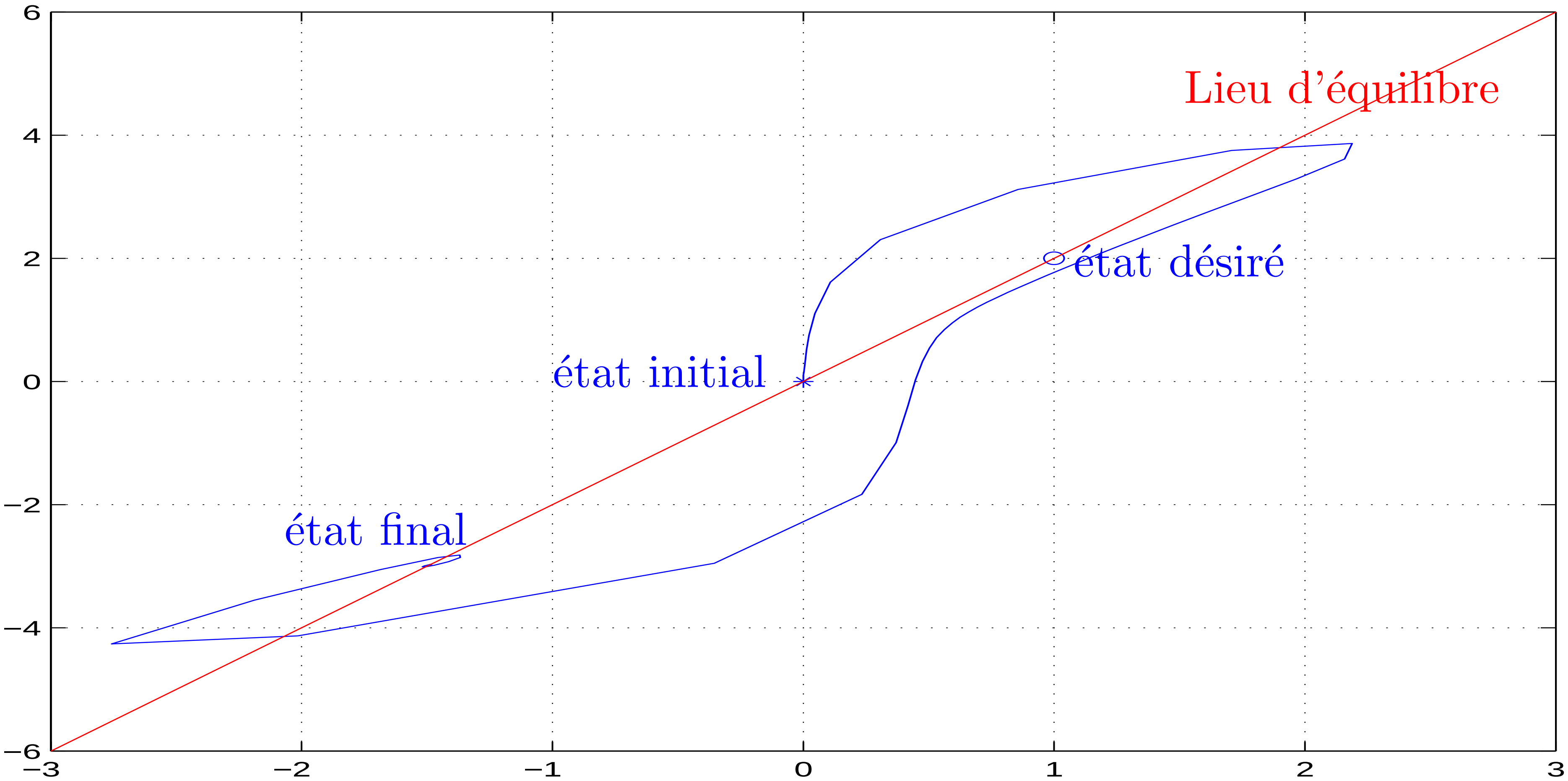
Supposons que pour ce faire, on augmente r en prenant

$$r = 160$$

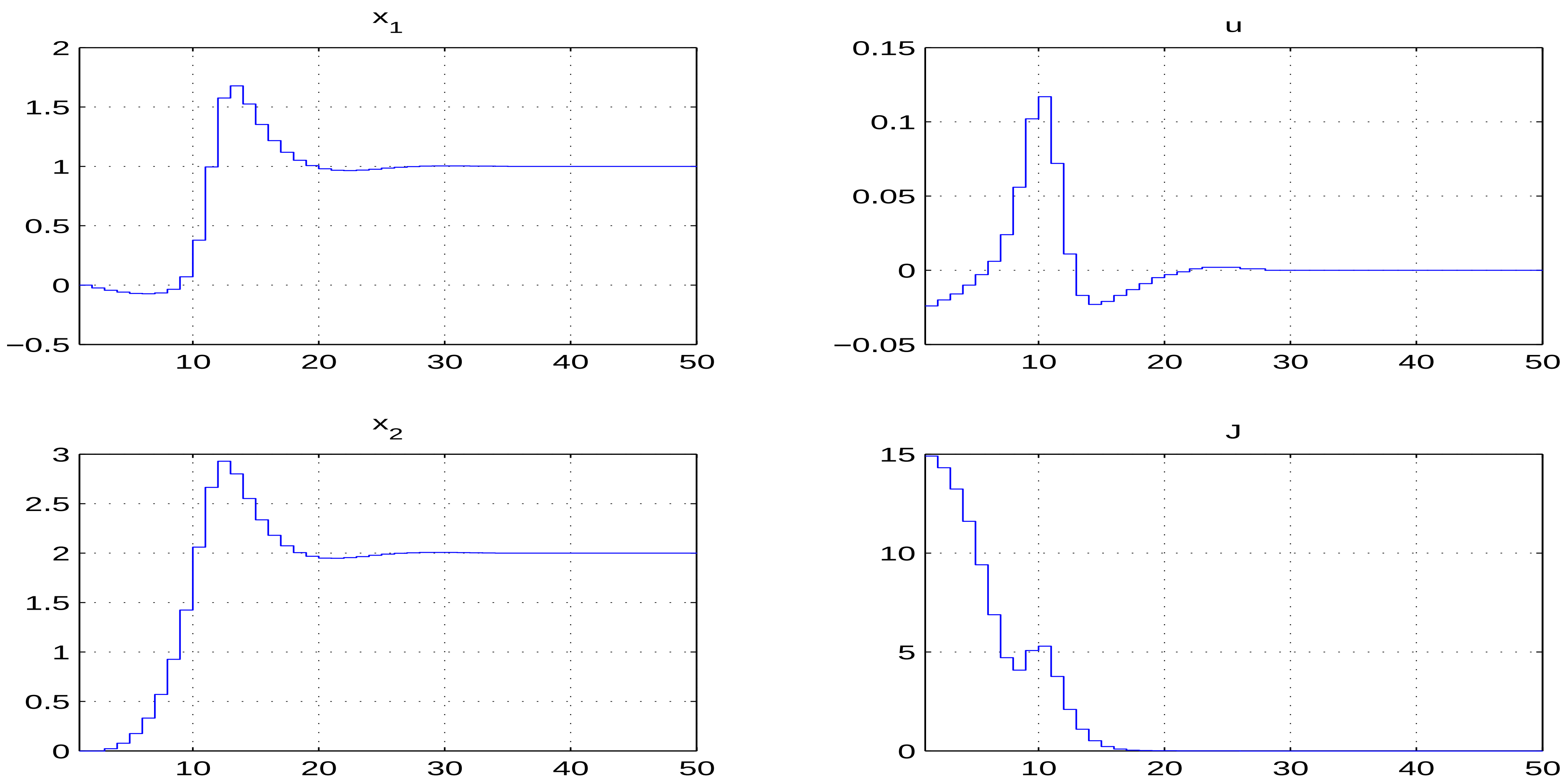
Test 2: $N = 2,\ N_u = 2,\ r = 160,\ q_x = 1$



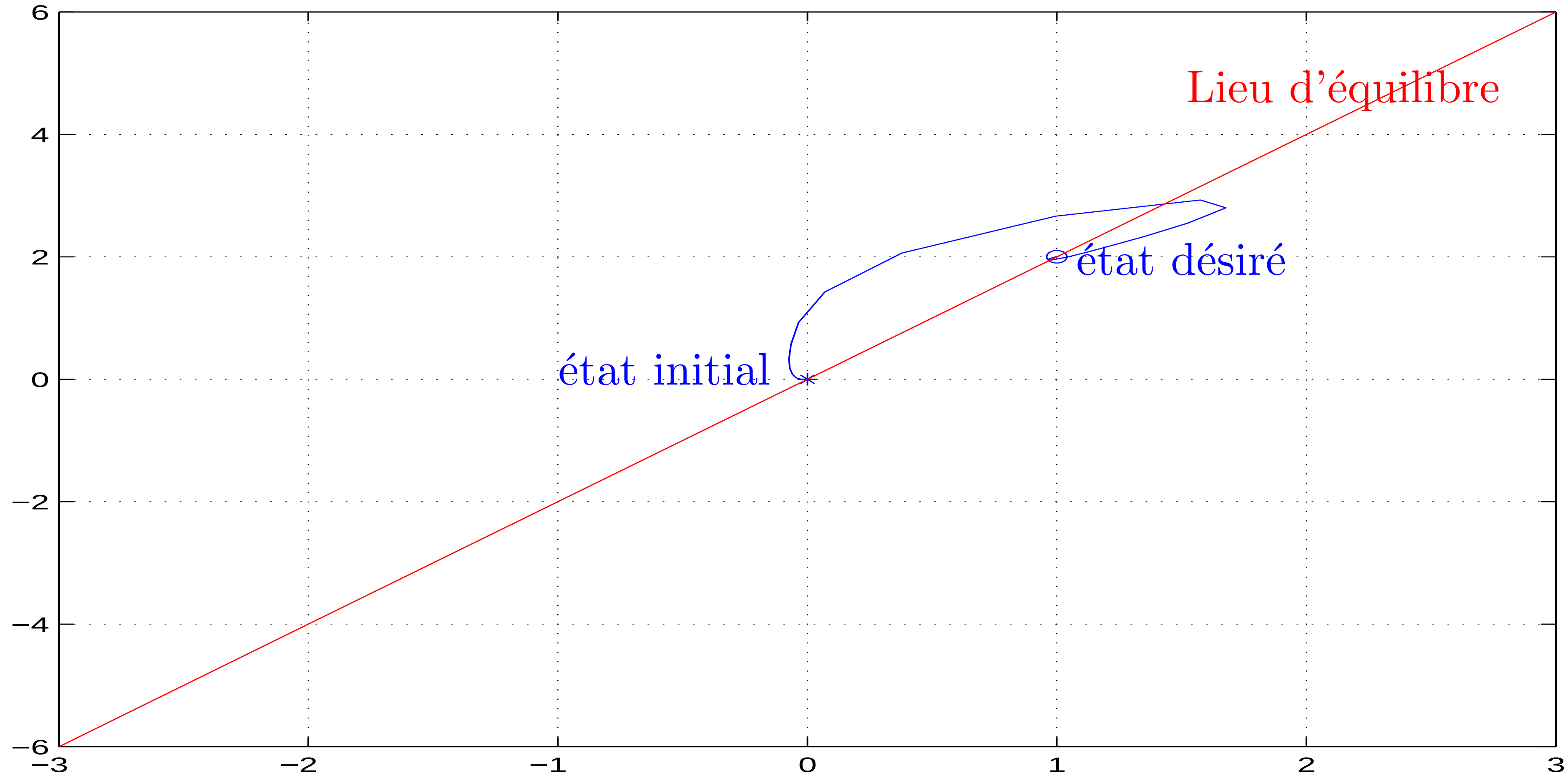
Test 2: $N = 2,\ N_u = 2,\ r = 160,\ q_x = 1$



Test 3: $N = 3,\ N_u = 2,\ r = 160,\ q_x = 1$



Test 3: $N = 3$, $N_u = 2$, $r = 160$, $q_x = 1$



Le passage à de $N = 2$ à $N = 3$ (l'augmentation de la longueur de l'horizon de pondération) a permis d'atteindre l'état désiré ... mais

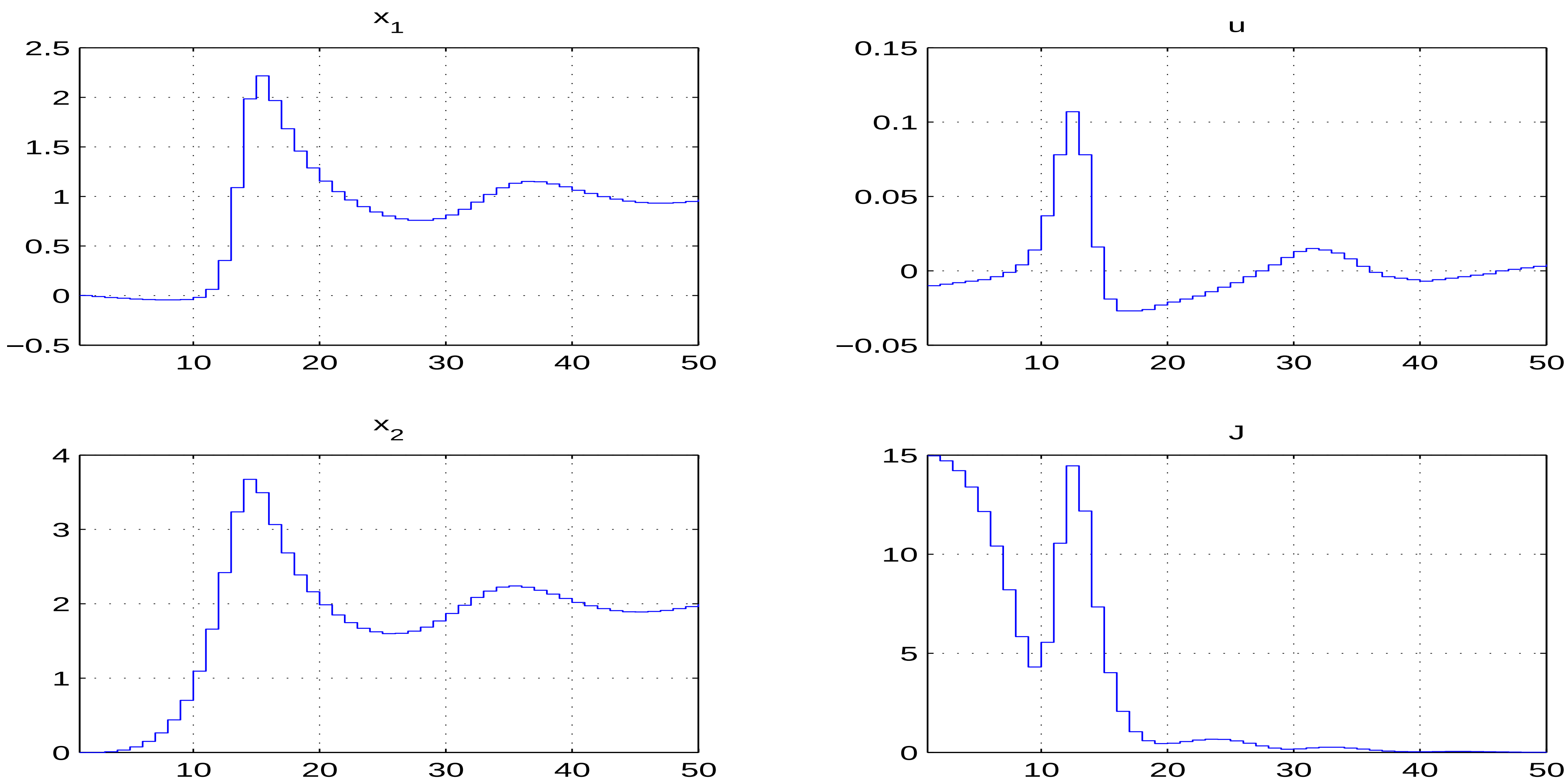
La commande a re-dépassé le seuil de 0.1

Augmentons encore r en prenant

$$r = 400$$

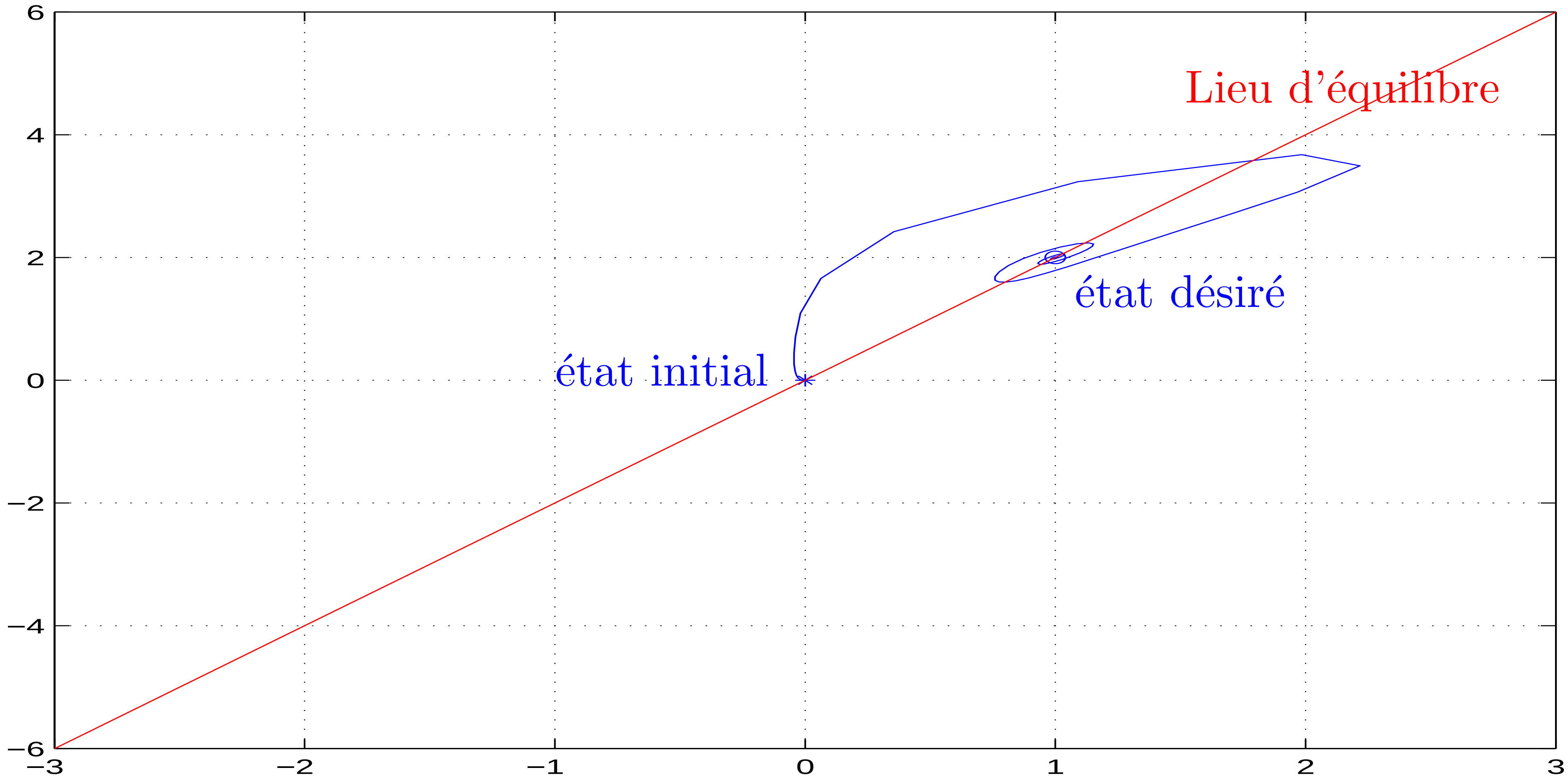
LABORATOIRE D'AUTOMATIQUE DE GRENOBLE

Test 4: $N = 3,\ N_u = 2,\ r = 400,\ q_x = 1$



CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE INP Grenoble UNIVERSITE JOSEPH FOURIER ELESA

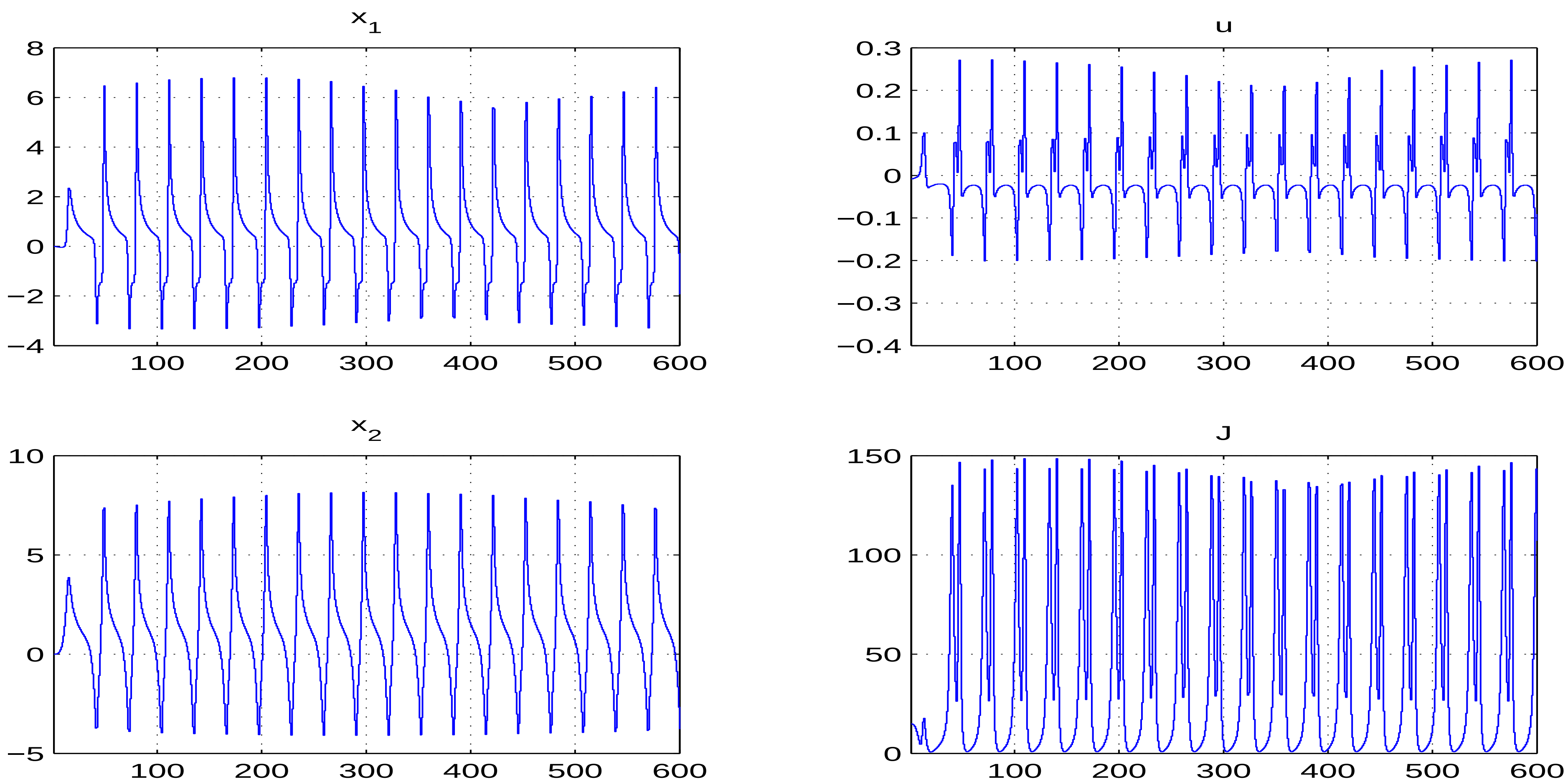
Test 4: $N = 3,\ N_u = 2,\ r = 400,\ q_x = 1$



Cela ne suffit pas, prenons $r = 500$

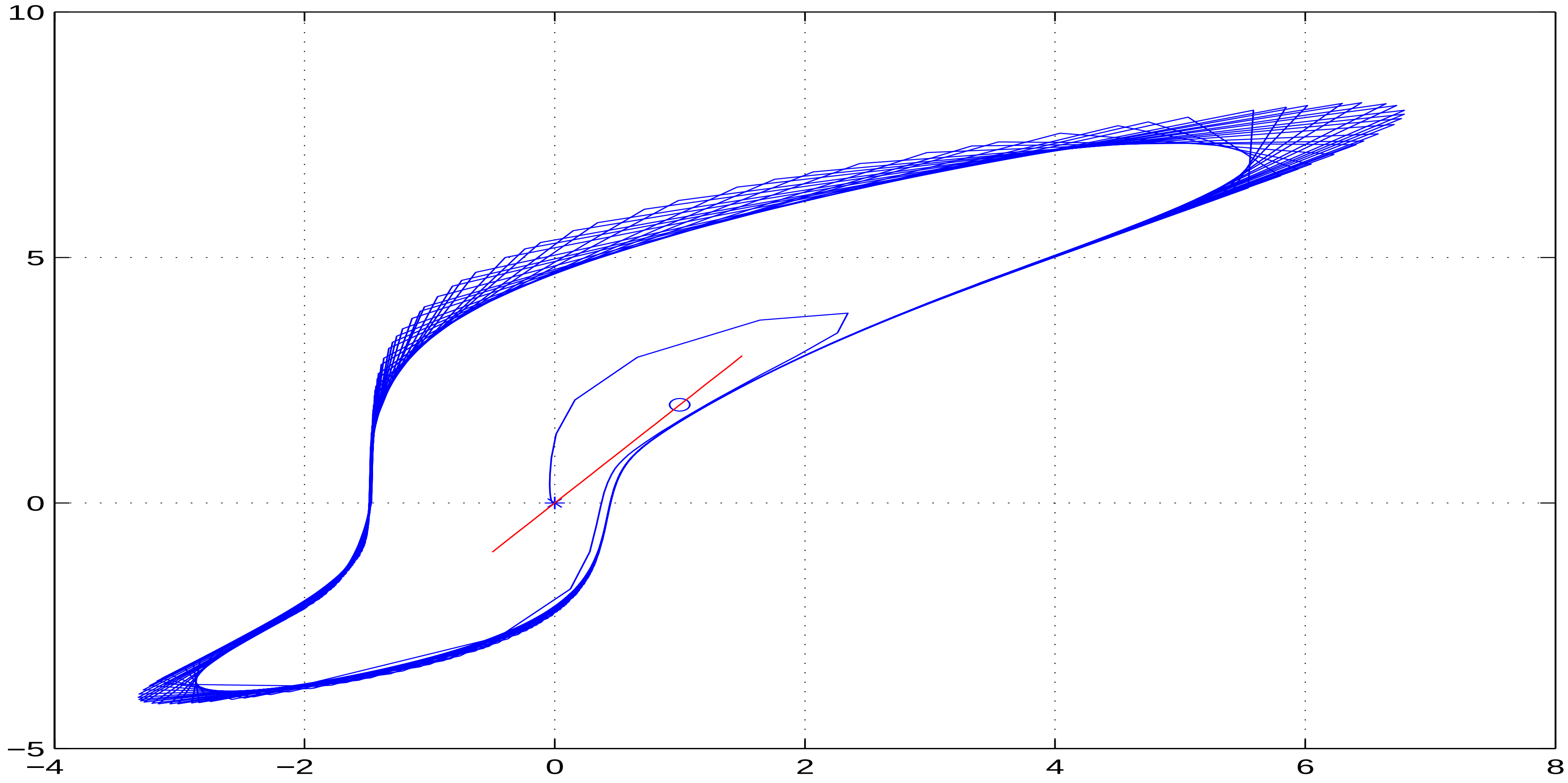
LABORATOIRE D'AUTOMATIQUE DE GRENOBLE

Test 5: $N = 3,\ N_u = 2,\ r = 500,\ q_x = 1$



CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
INP Grenoble
UNIVERSITE JOSEPH FOURIER
ELESA

TEST 5: $N = 3,\ N_u = 2,\ r = 500,\ q_x = 1$



- Apparition d'un cycle limite.
- En plus, la commande dépasse 0.25 !!!!

Rappelons que jusqu'ici, les contraintes étaient définies par l'ensemble admissible

$$U_{N_u} := \Big\{ \tilde{u} \in (\mathbb{R}^1)^N \quad | \quad \tilde{u}_j = 0 \quad \forall j \geq N_u - 1 \Big\}$$

Rappelons que jusqu'ici, les contraintes étaient définies par l'ensemble admissible

$$U_{N_u} := \left\{ \tilde{u} \in (\mathbb{R}^1)^N \quad | \quad \tilde{u}_j = 0 \quad \forall j \geq N_u - 1 \right\}$$

Soit la nouvelle définition des contraintes,

$$U_{N_u} := \left\{ \tilde{u} \in [-0.1, 0.1]^N \quad | \quad \tilde{u}_j = 0 \quad \forall j \geq N_u - 1 \right\}$$

qui permet d'exprimer explicitement le souhait d'avoir $|u| \leq 0.1$.

Rappelons que jusqu'ici, les contraintes étaient définies par l'ensemble admissible

$$U_{N_u} := \left\{\tilde{u} \in (\mathbb{R}^1)^N \quad | \quad \tilde{u}_j = 0 \quad \forall j \geq N_u - 1\right\}$$

Soit la nouvelle définition des contraintes,

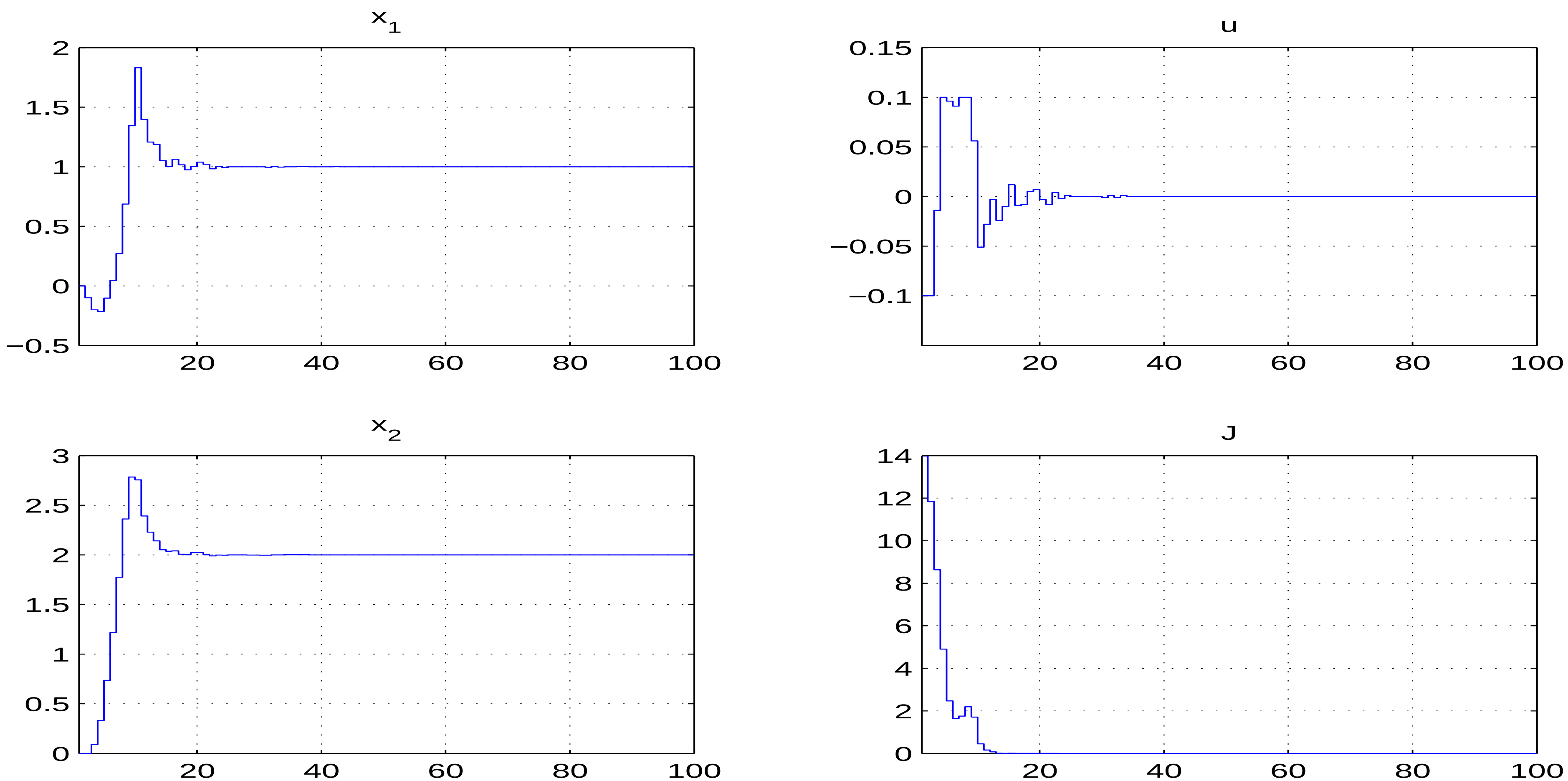
$$U_{N_u} := \left\{\tilde{u} \in [-0.1, 0.1]^N \quad | \quad \tilde{u}_j = 0 \quad \forall j \geq N_u - 1\right\}$$

qui permet d'exprimer explicitement le souhait d'avoir $|u| \leq 0.1$.

Testons cette nouvelle configuration avec

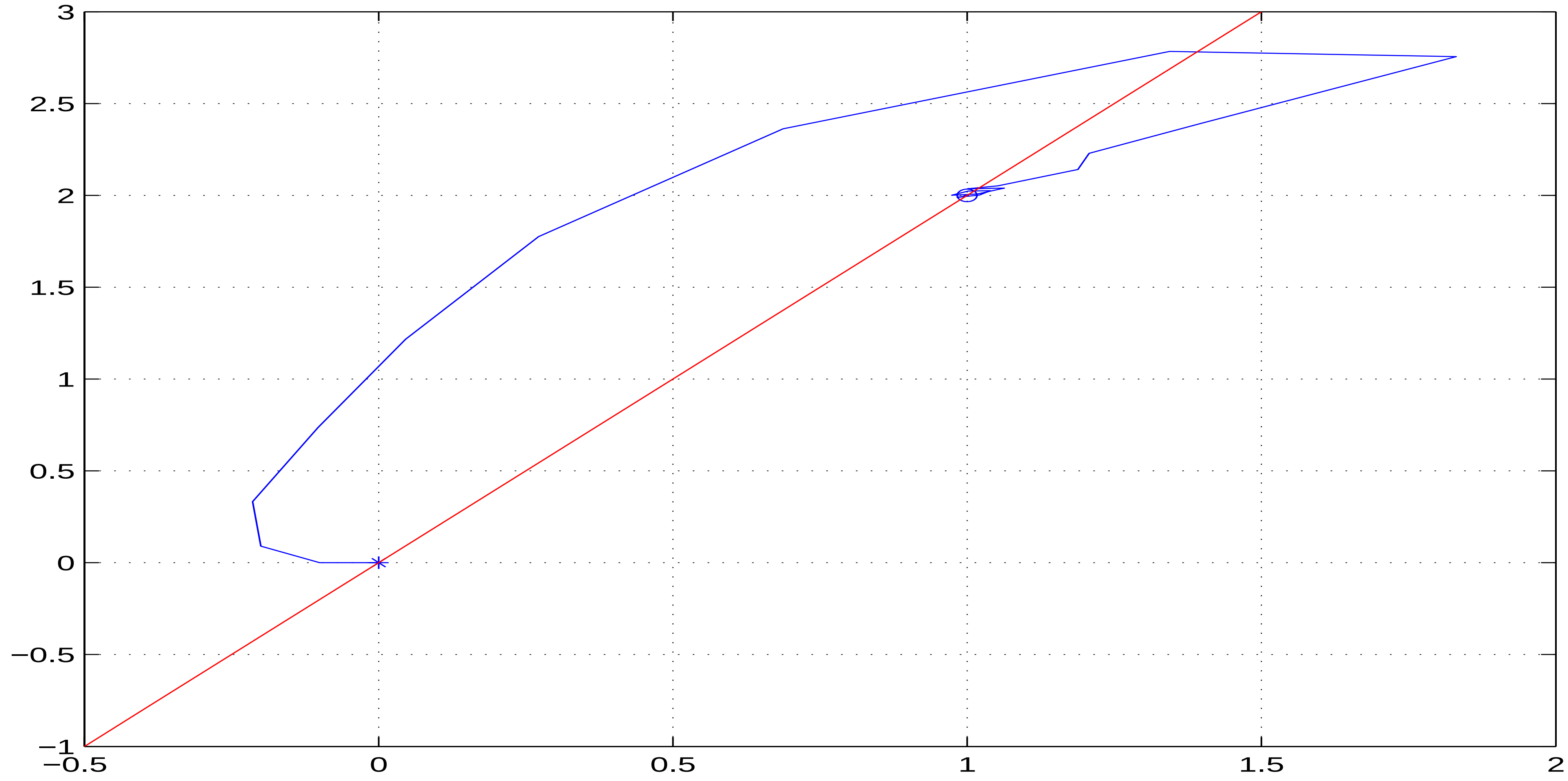
$$r = 1\ ,\ q_x = 1\ ,\ N = 3\ ,\ N_u = 3$$

Test 6: $N = 3$, $N_u = 3$, $r = 1$, $q_x = 1$ et commande contrainte à 0.1



LABORATOIRE D'AUTOMATIQUE DE GRENOBLE

Test 6: $N = 3$, $N_u = 3$, $r = 1$, $q_x = 1$ et commande contrainte à 0.1



1. Cela vaut le coup ... !

1. Cela vaut le coup ... !

   ✓ Généricité

   Nous n'avons jamais examiner la structure du système.

1. Cela vaut le coup ... !

   ✓ Généricité

   Nous n'avons jamais examiner la structure du système.

   ✓ Gestion aisée des contraintes

   Il aura suffi de les ajouter dans la formulation du problème.

1. Cela vaut le coup ... !

   ✓ Généricité

   Nous n'avons jamais examiner la structure du système.

   ✓ Gestion aisée des contraintes

   Il aura suffi de les ajouter dans la formulation du problème.

2. On ne peut pas faire n'importe quoi ... !

   La convergence n'est pas toujours obtenue.

1. Cela vaut le coup ... !

   ✓ Généricité

   Nous n'avons jamais examiner la structure du système.

   ✓ Gestion aisée des contraintes

   Il aura suffi de les ajouter dans la formulation du problème.

2. On ne peut pas faire n'importe quoi ... !

   La convergence n'est pas toujours obtenue.

$\longrightarrow$ Il faut un minimum d'étude théorique de stabilité.

Soit le système dynamique suivant

$$x(k+1) = f(x(k), u(k)) \quad ; \quad x(k) \in \mathbb{R}^n \quad ; \quad u(k) \in \mathbb{R}^m \quad ; \quad f(0,0) = 0$$

Soit le système dynamique suivant

$$x(k+1) = f(x(k), u(k)) \quad ; \quad x(k) \in \mathbb{R}^n \quad ; \quad u(k) \in \mathbb{R}^m \quad ; \quad f(0,0) = 0$$

soit le critère d'optimisation sur la séquence $\tilde{u} \in (\mathbb{R}^m)^N$

$$V(x, k, \tilde{u}) := F(x(k+N)) + \sum_{i=k}^{k+N-1} l(x(i), u(i)) \quad ; \quad \tilde{u} \in U(x, k)$$

avec $l(x, u)$ définie positive dans ces arguments.

Soit le système dynamique suivant

$$x(k+1) = f(x(k), u(k)) \quad ; \quad x(k) \in \mathbb{R}^n \quad ; \quad u(k) \in \mathbb{R}^m \quad ; \quad f(0,0) = 0$$

soit le critère d'optimisation sur la séquence $\tilde{u} \in (\mathbb{R}^m)^N$

$$V(x, k, \tilde{u}) := F(x(k+N)) + \sum_{i=k}^{k+N-1} l(x(i), u(i)) \quad ; \quad \tilde{u} \in U(x, k)$$

avec $l(x, u)$ définie positive dans ces arguments et $U(x, k)$ défini par

$$U(x, k) := \left\{ \tilde{u} \in (\mathbb{R}^m)^N \quad \text{telle que} \quad X(k+N; k; x; \tilde{u}) = 0 \right\}$$

Soit le système dynamique suivant

$$x(k+1) = f(x(k), u(k)) \quad ; \quad x(k) \in \mathbb{R}^n \quad ; \quad u(k) \in \mathbb{R}^m \quad ; \quad f(0,0) = 0$$

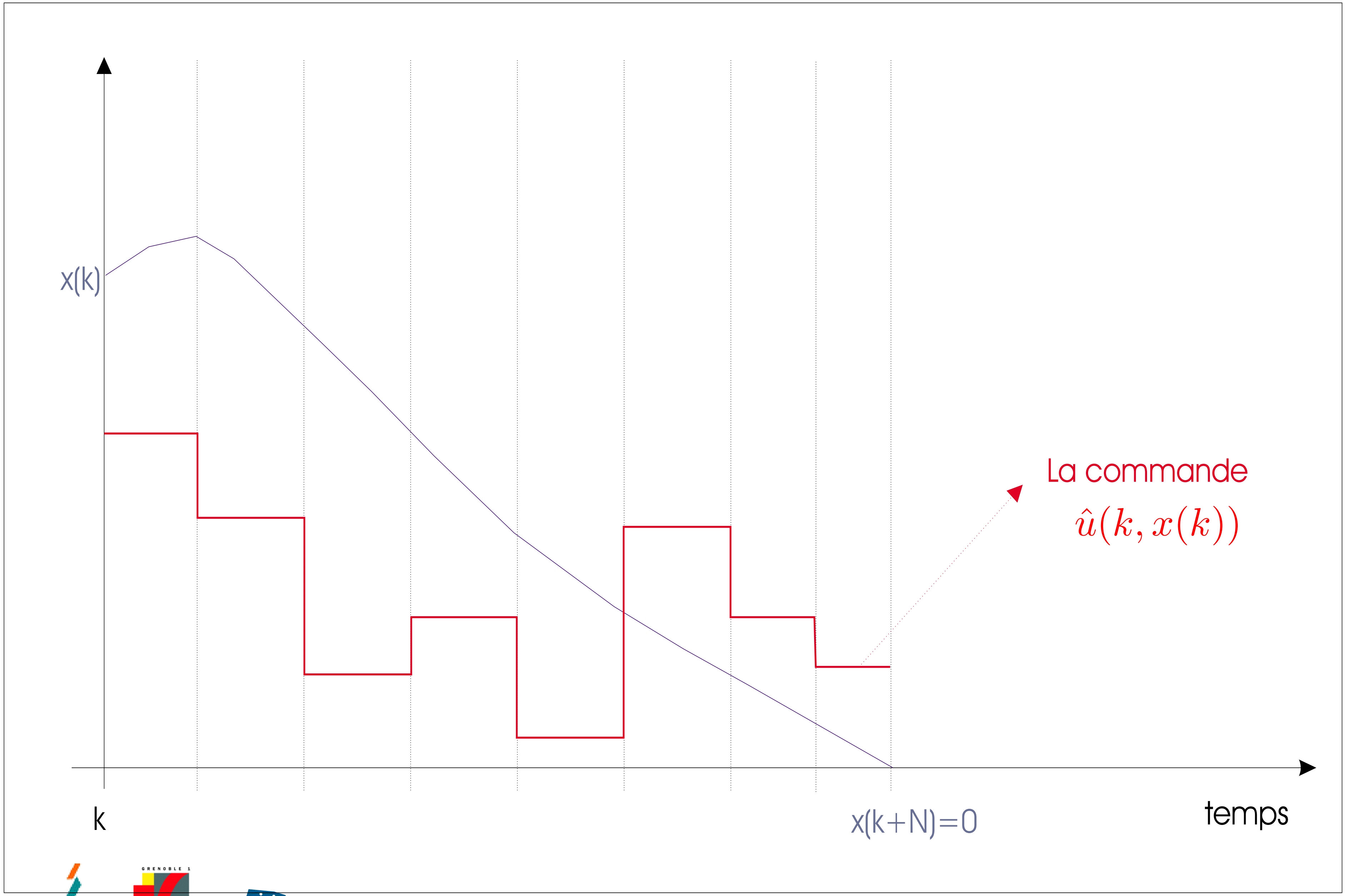
soit le critère d'optimisation sur la séquence $\tilde{u} \in (\mathbb{R}^m)^N$

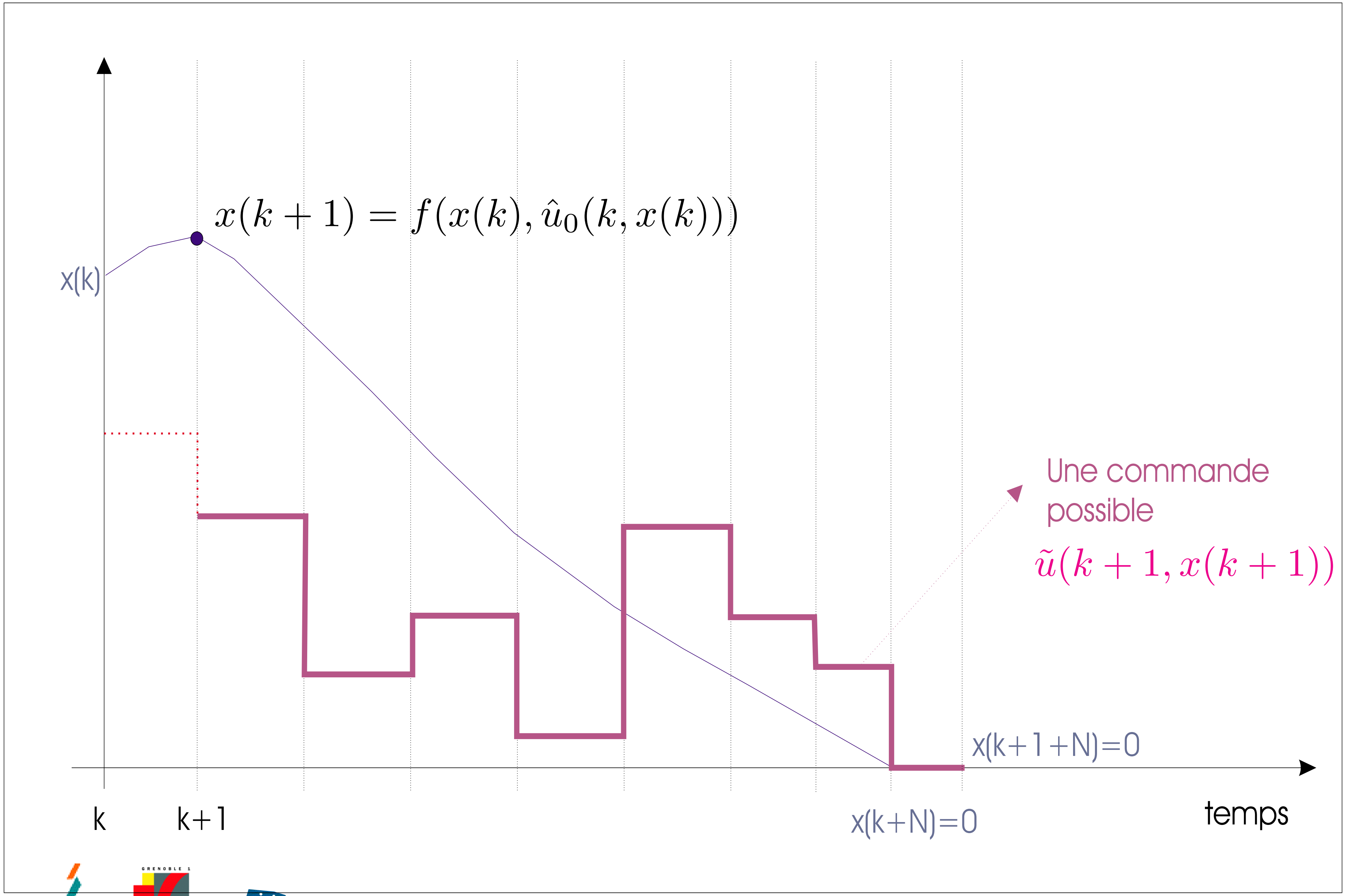
$$V(x,k,\tilde{u}) := F(x(k+N)) + \sum_{i=k}^{k+N-1} l(x(i), u(i)) \quad ; \quad \tilde{u} \in U(x,k)$$

avec $l(x,u)$ définie positive dans ces arguments et $U(x,k)$ défini par

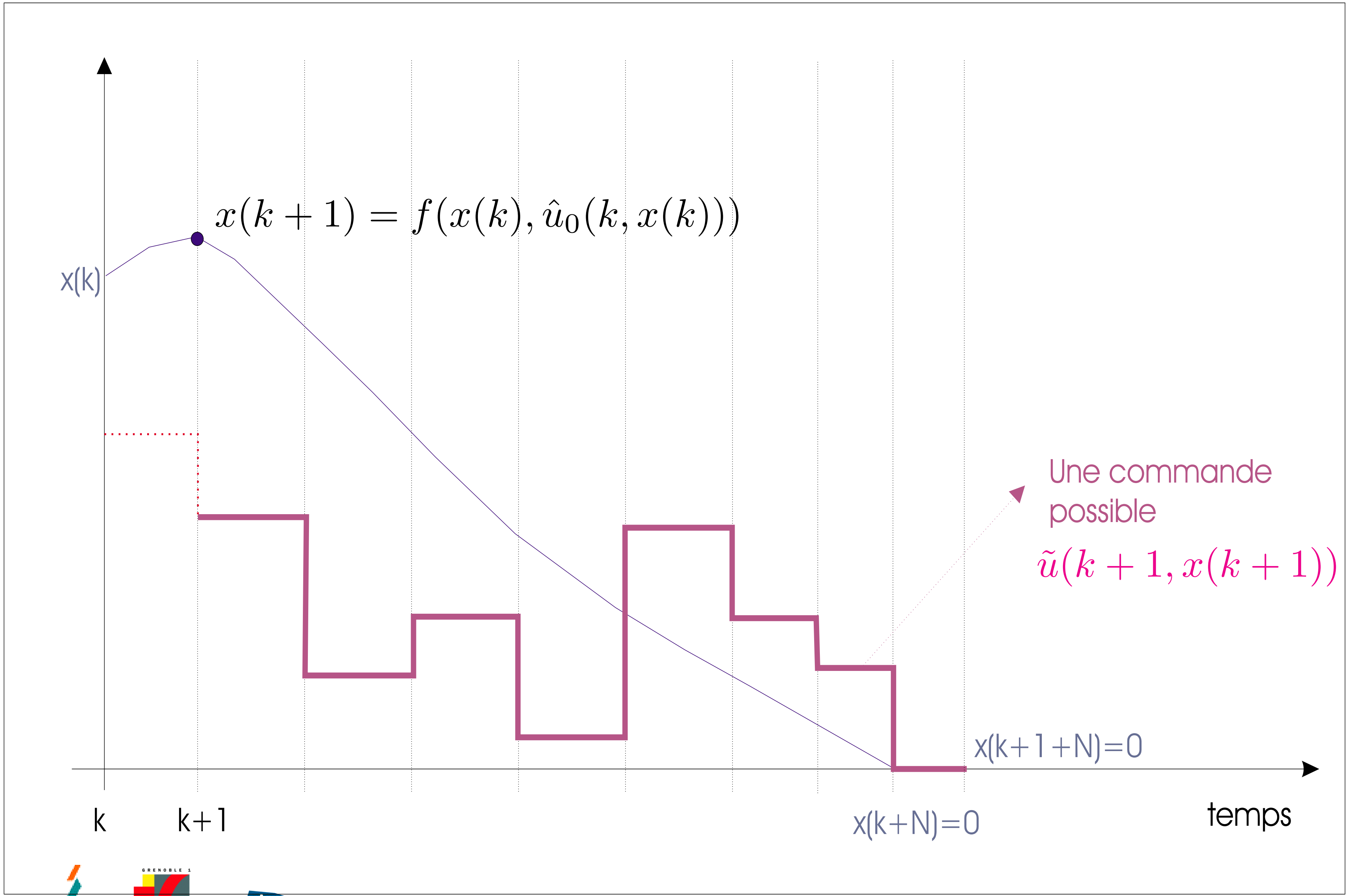
$$U(x,k) := \left\{ \tilde{u} \in (\mathbb{R}^m)^N \quad \text{telle que} \quad X(k+N; k; x; \tilde{u}) = 0 \right\}$$

$\Longrightarrow$ Stabilité asymptotique au sens de Lyapunov de $x = 0$ pour le sysème en boucle fermée.

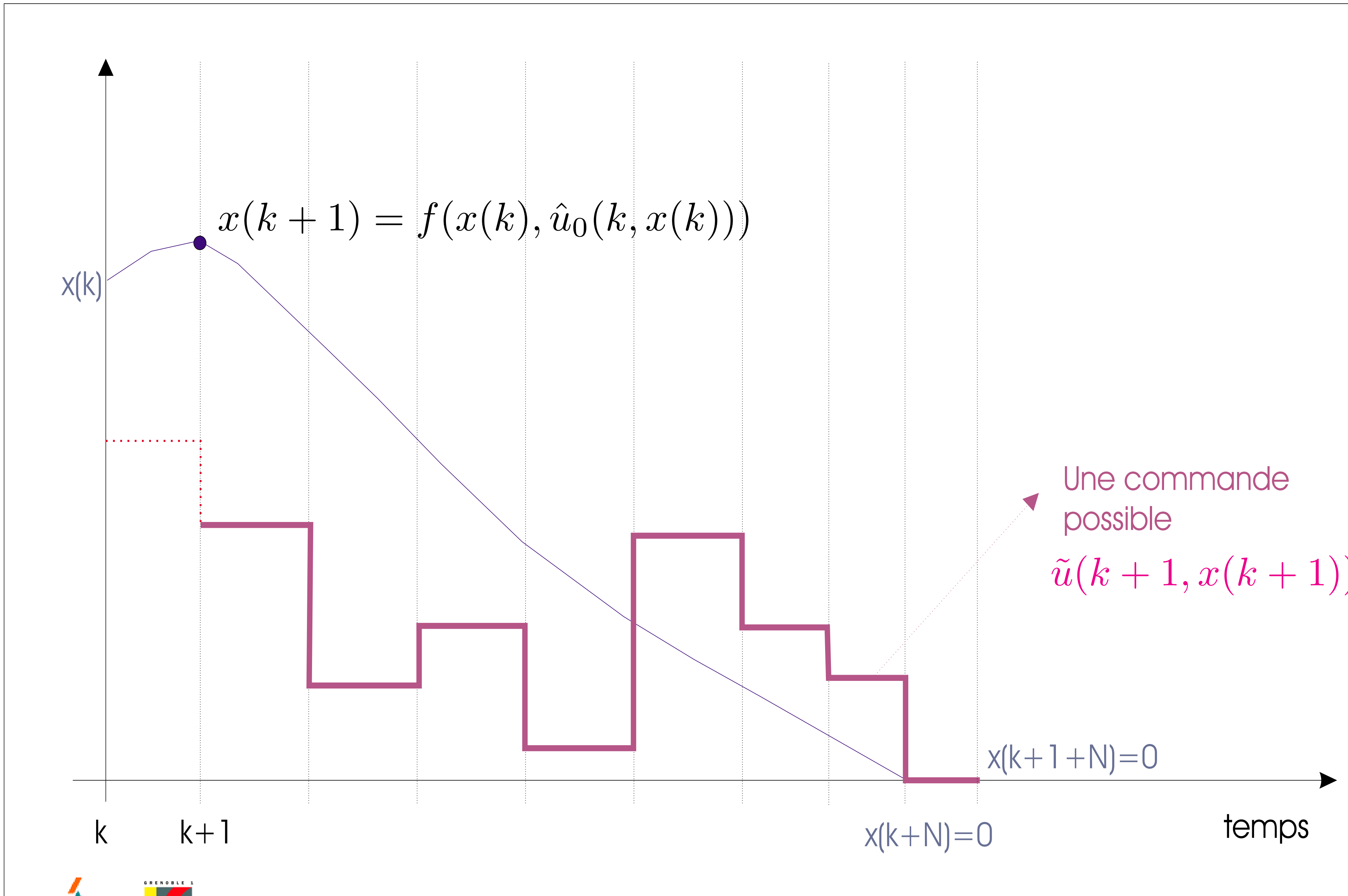

x(k)
La commande
$\hat{u}(k, x(k))$
k
x(k+N)=0
temps


$x(k+1) = f(x(k), \hat{u}_0(k, x(k)))$
x(k)
Une commande possible
$\tilde{u}(k+1, x(k+1))$
x(k+1+N)=0
k
k+1
x(k+N)=0
temps

$$V\Big(x(k+1), \hat{u}(k+1, x(k+1))\Big) \leq V\Big(x(k+1), \tilde{u}(k+1, x(k+1))\Big)$$



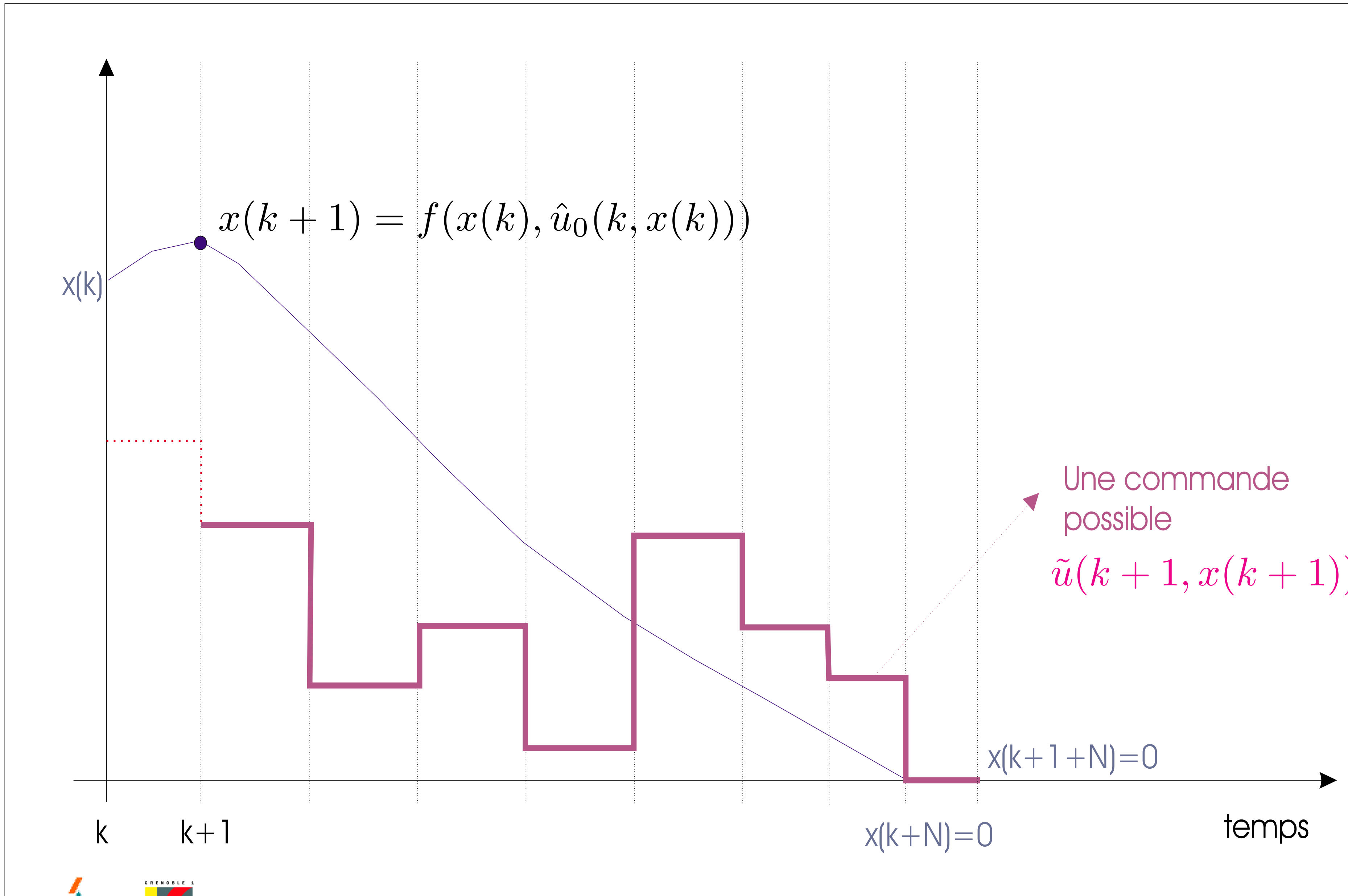
$$
\begin{aligned}
V\Big(x(k+1), \hat{u}(k+1, x(k+1))\Big) &\leq V\Big(x(k+1), \tilde{u}(k+1, x(k+1))\Big) \\
&\leq V\Big(x(k), \hat{u}(k, x(k))\Big) - l(x(k), u(k))
\end{aligned}
$$



$$V\Big(x(k+1),\hat{u}(k+1,x(k+1))\Big) \le V\Big(x(k+1),\tilde{u}(k+1,x(k+1))\Big)$$

$$\le V\Big(x(k),\hat{u}(k,x(k))\Big) - l(x(k),\hat{u}_0(k,x(k)))$$

$$\Delta\hat{V}(k,x(k)) \le -l(x(k),\hat{u}_0(k,x(k)))$$



$$\Delta \hat{V}(k, x(k)) \leq -l(x(k), \hat{u}_0(k, x(k)))$$

Or comme $l(x(k), \hat{u}_0(k, x(k))$ est définie positive dans ses arguments, le principe de LaSalle permet de conclure à la stabilité globale asymptotique

$$\lim_{k \to \infty} (x(k), \hat{u}_0(k, x(k))) = (0, 0)$$

avec $\hat{V}(k, x(k))$ comme fonction de Lyapunov.

Ne pas oublier le rôle fondamental joué par

- La contrainte finale.
- La définie-positivité du critère (surtout par rapport à x).
- Le caractère optimal de la solution $\hat{u}(k, x(k))$.
  (Au moins sous-optimal)

Des résultats semblables sont disponibles pour les systèmes continus.

avec des difficultés techniques liées à l'existence des solutions.

Le principe étant exactement le même.

Problématiques potentielles

- ✓ *Preuve de stabilité des formulations de base.*
  (Assouplir les hypothèses)
- ✓ *Résolution numérique d'un PNL donné.*
  (rapidité, fiabilité, catégorie)
- ✓ *Inventer des nouvelles formulations génériques.*
  (Sous optimalité, réduction du coût de calcul)
- ✓ *Exploiter la structure d'un problème concret.*

Exploiter la structure d'un pb concret

✓ Systèmes mécaniques non plats

(Pendule inversé ($x \in \mathbb{R}^4$), Bille sur un rail)

♯ *Problème d'optimisation dans* $\mathbb{R}^2$

♯ *Temps de calcul sur PIII 600 Mhz: qq* 0.01 *secondes*

Eur. J. of Control 99

Exploiter la structure d'un pb concret

- ✓ Systèmes mécaniques non plats
- ✓ Satellite en mode défaillant

  $(x \in \mathbb{R}^6, u \in \mathbb{R}^2$

  Optimisation dans $\mathbb{R}^2$

  Temps de calcul: qq 0.01 *secondes.*

Eur. J. of Control 99

J. of Dyn. Syst. Meas. & Control 2003 , Journal of Optimization in Engineering 2003

Exploiter la structure d'un pb concret

- ✓ Systèmes mécaniques non plats
- ✓ Satellite en mode défaillant
- ✓ Missile en mission d'interception

  ($x \in \mathbb{R}^4$ $u \in \mathbb{R}$)

  Optimisation dans $\mathbb{R}_+$

  (qq 0.0001 sec ou qq 0.01 sec selon algorithme)

Eur. J. of Control 99

J. of Dyn. Syst. Meas. & Control 2003 , Journal of Optimization in Engineering 2003

Control Engineering Practice 2001

Exploiter la structure d'un pb concret

- ✓ Systèmes mécaniques non plats
- ✓ Satellite en mode défaillant
- ✓ Missile en mission d'interception
- ✓ Systèmes non-holonomes chaînés

  $x \in \mathbb{R}^n$, $u \in \mathbb{R}^2$, optimisation quadratique

  énergie minimale

  temps minimal sous saturation

Eur. J. of Control 99

J. of Dyn. Syst. Meas. & Control 2003, , Journal of Optimization in Engineering 2003

Control Engineering Practice 2001

European Journal of Control 96

Journal of Optimisation theory and applications 2003

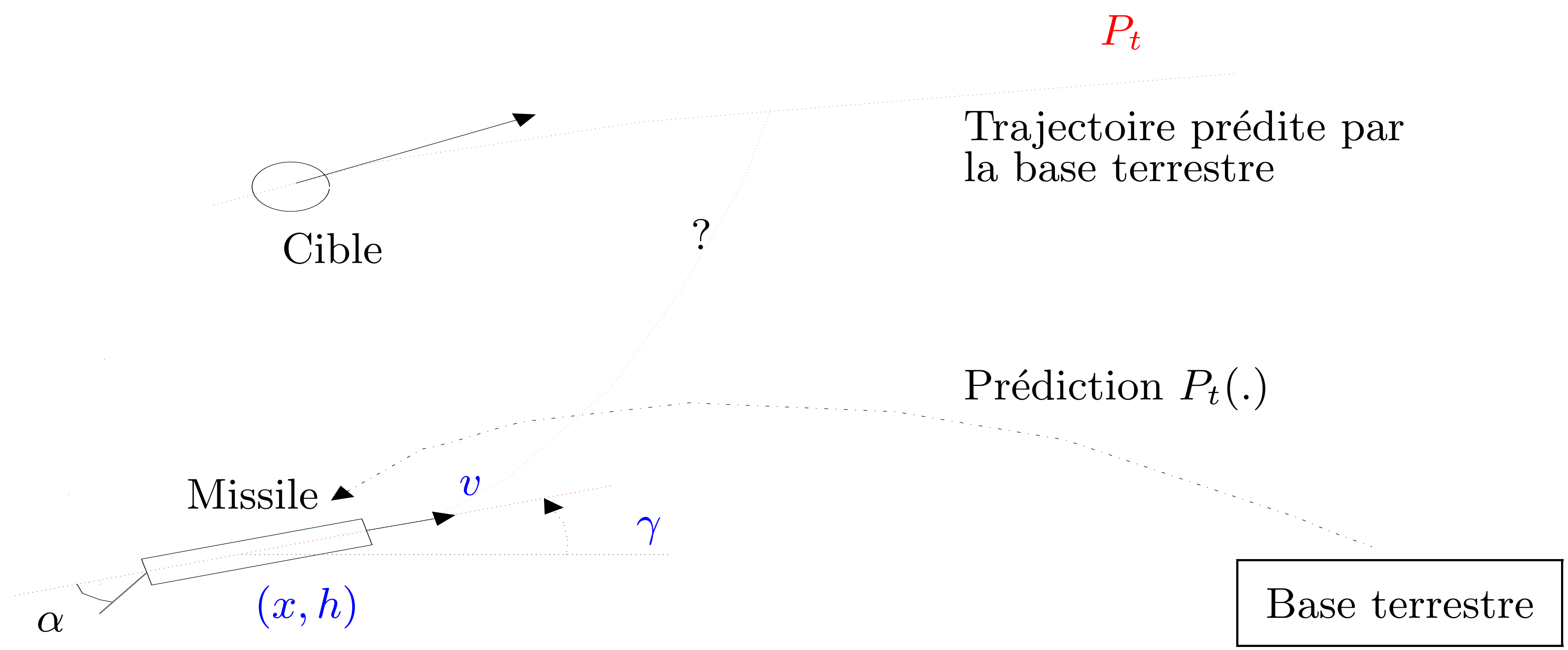
LABORATOIRE D'AUTOMATIQUE DE GRENOBLE

Exploiter la structure d'un pb concret

- ✓ Systèmes mécaniques non plats
- ✓ Satellite en mode défaillant
- ✓ Missile en mission d'interception
- ✓ Systèmes non-holonomes chaînés

Eur. J. of Control 99
J. of Dyn. Syst. Meas. & Control 2003, , Journal of Optimization in Engineering 2003
Control Engineering Practice 2001
European Journal of Control 96
Journal of Optimisation theory and applications 2003

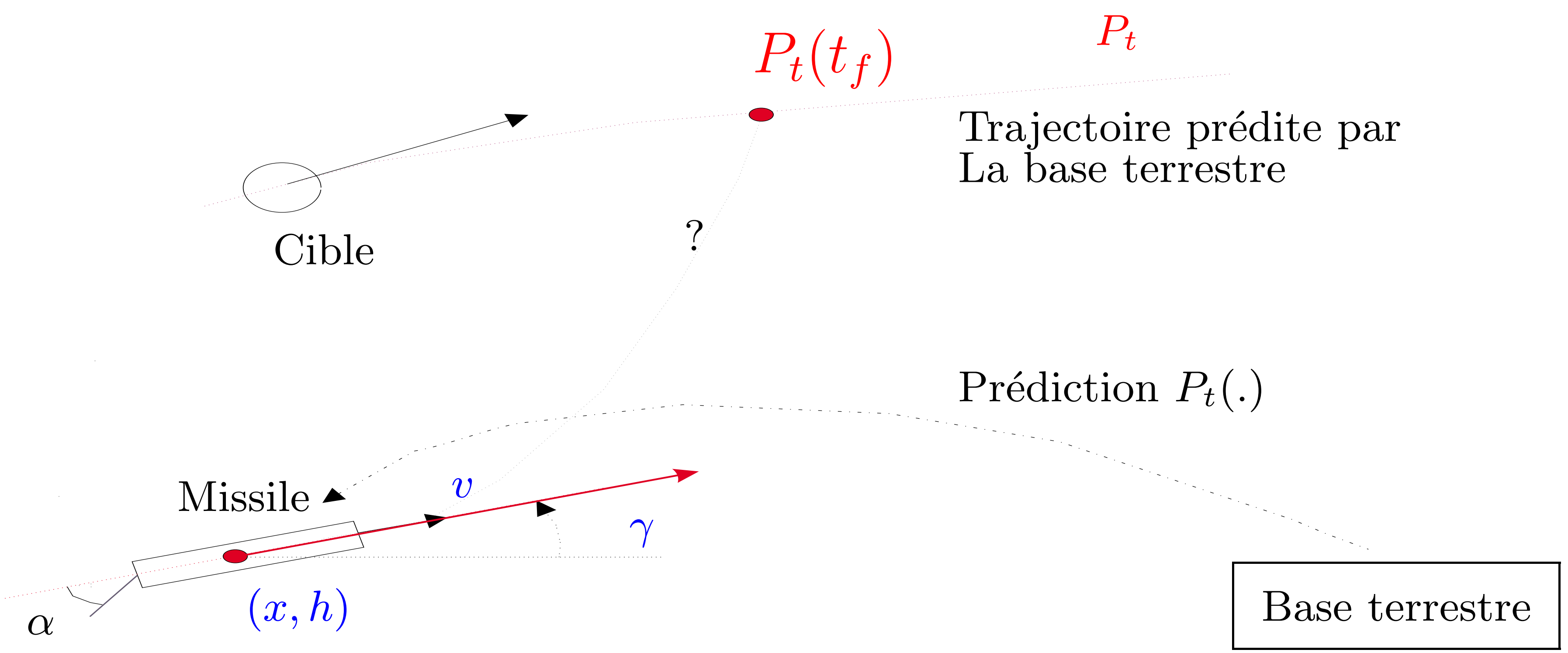
Interception en temps minimal



LABORATOIRE D'AUTOMATIQUE DE GRENOBLE

Interception en temps minimal

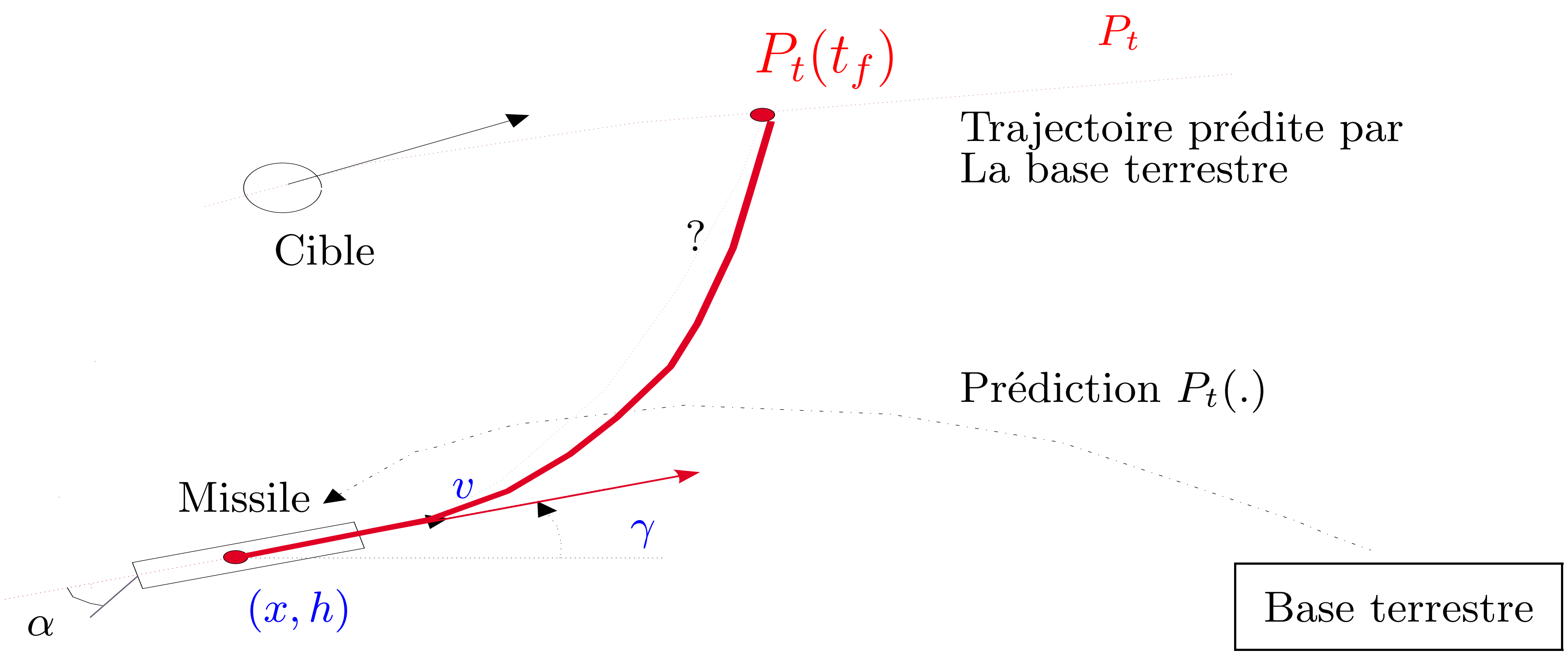
Choix de t_f candidat $\quad \rightarrow$ point d'interception potentiel



Interception en temps minimal

Choix de t_f candidat $\rightarrow$ point d'interception potentiel

Une seule parabole d'interception $y = Q(x)$

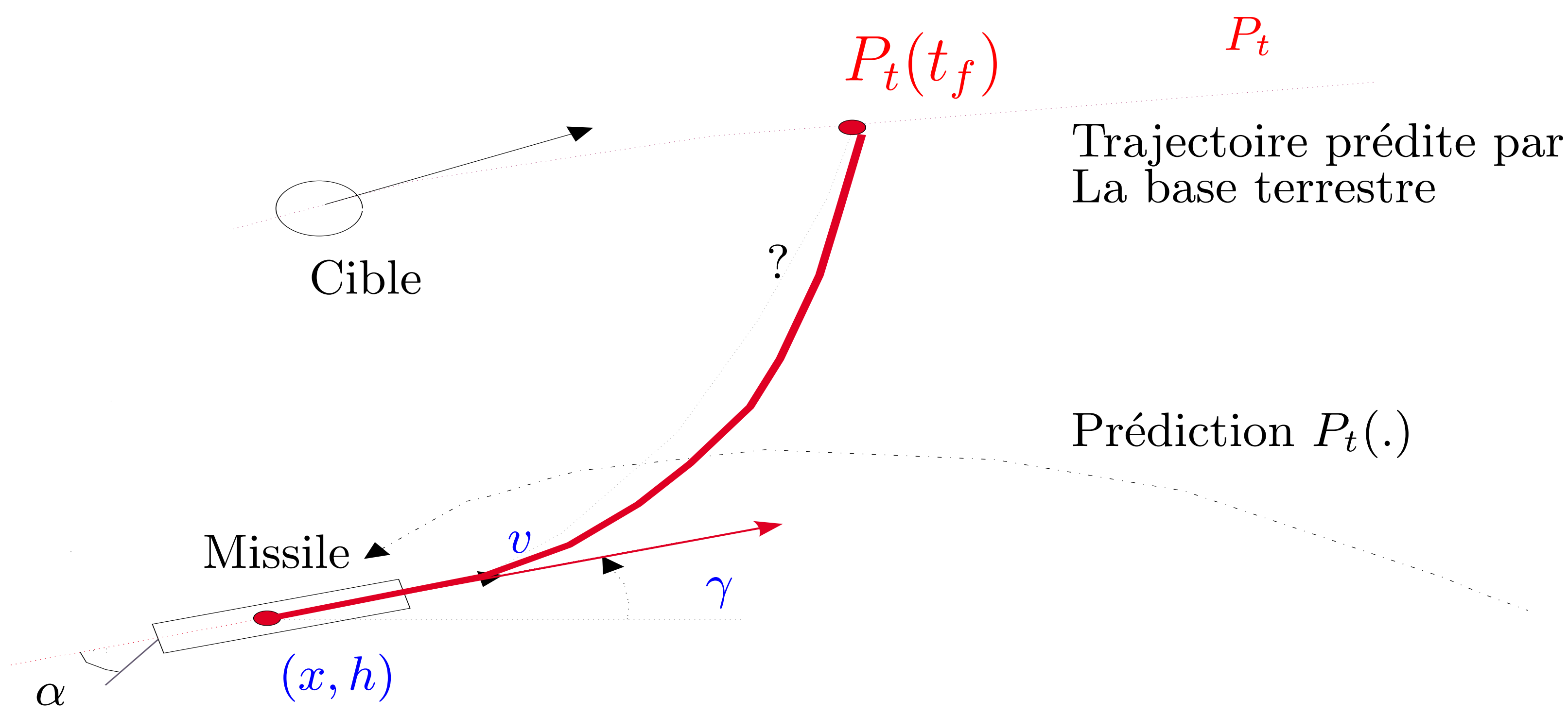


$$\min_{t_f}\left\{ t_f > 0 \mid \int_0^{t_f} v(\tau)|_{y=Q(x)} d\tau = \int_{y=Q(x)} \sqrt{1+(\frac{dy}{dx})^2}dx \right\}$$

Interception en temps minimal

Choix de t_f candidat $\quad\rightarrow$ point d'interception potentiel

Une seule parabole d'interception $y = Q(x)$

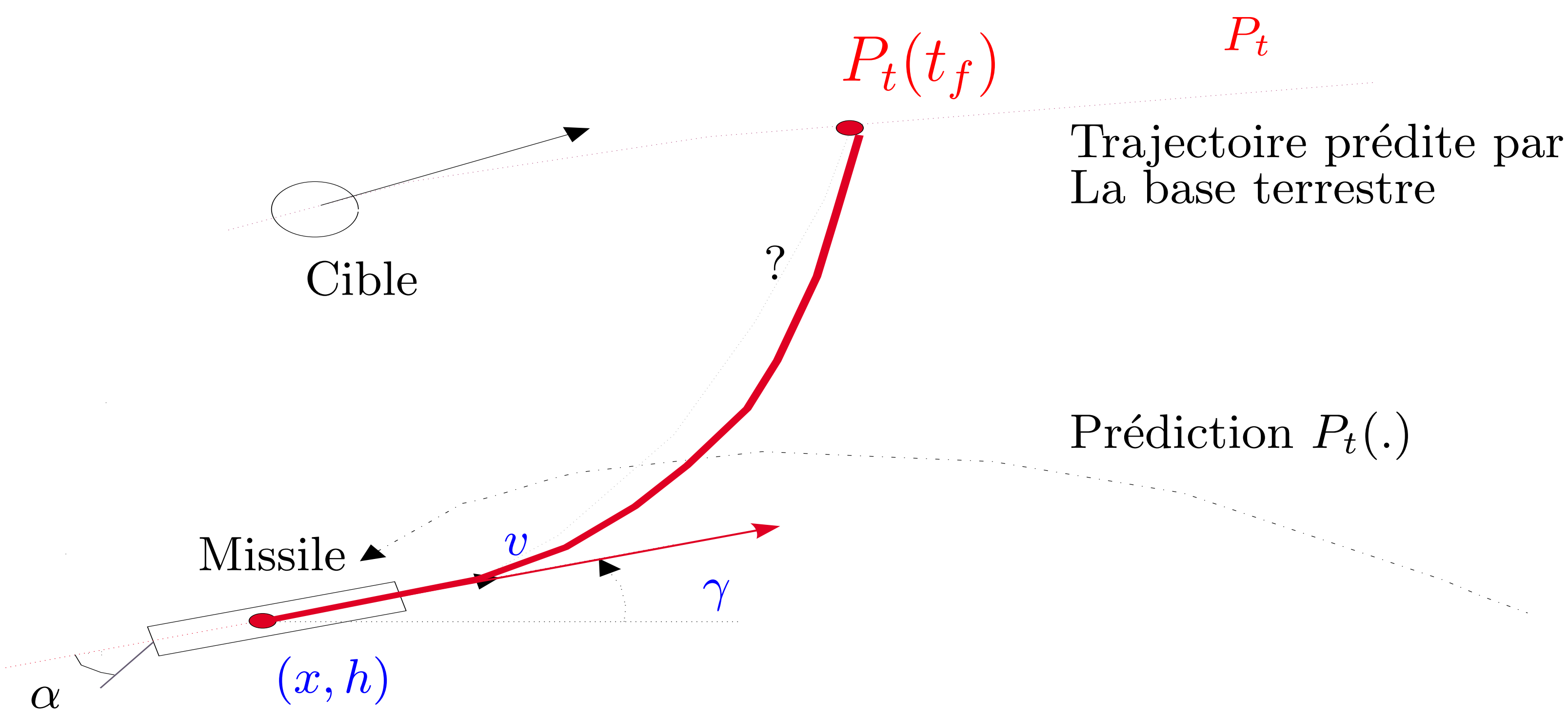


$$\min_{t_f}\left\{t_f > 0 \mid \quad \int_0^{t_f} v(\tau)|_{y=Q(x)} d\tau = \int_{y=Q(x)} \sqrt{1+(\frac{dy}{dx})^2 dx}\right\}$$

Interception en temps minimal

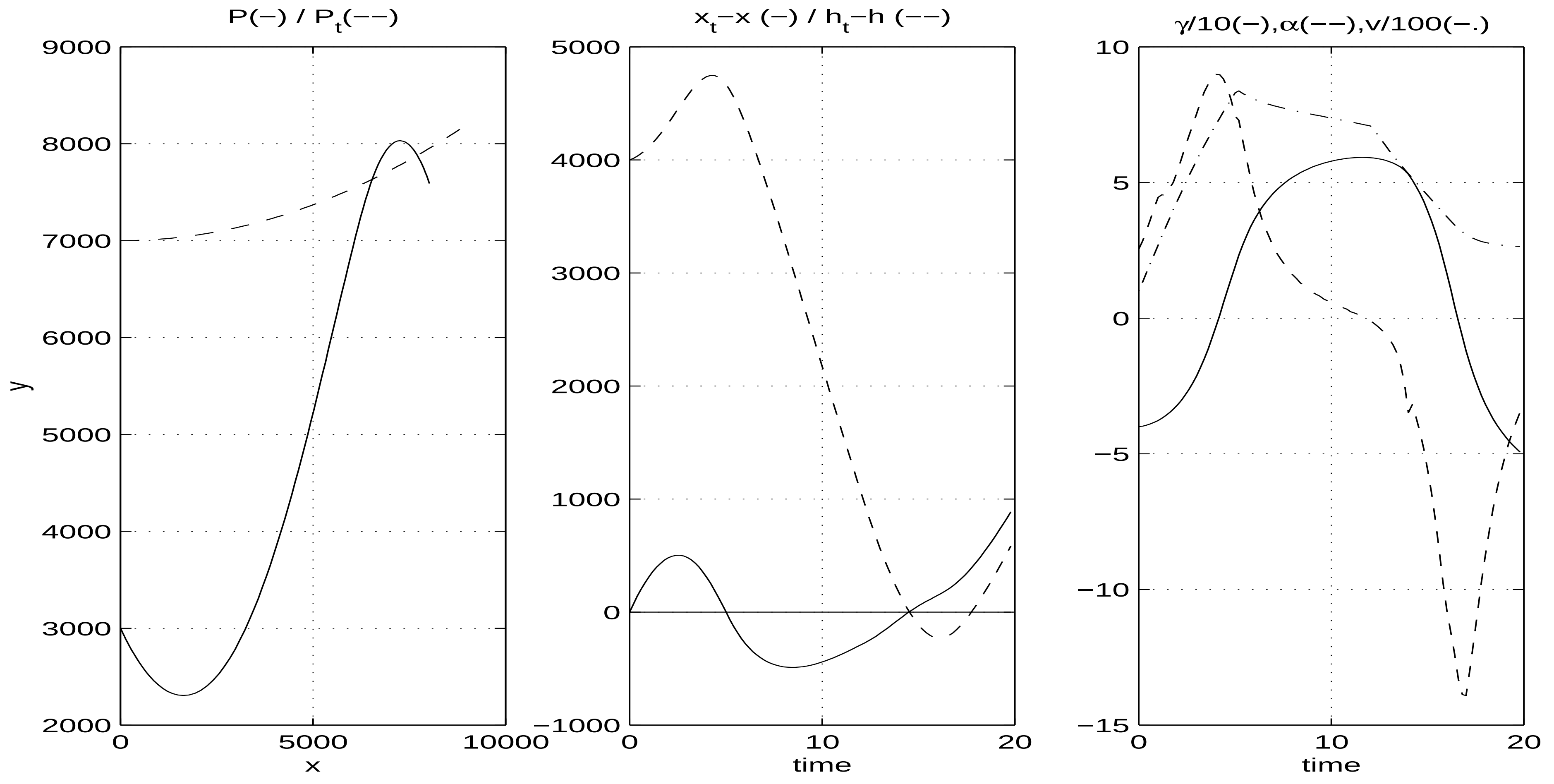
Choix de t_f candidat $\quad \rightarrow$ point d'interception potentiel

Une seule parabole d'interception $y = Q(x)$

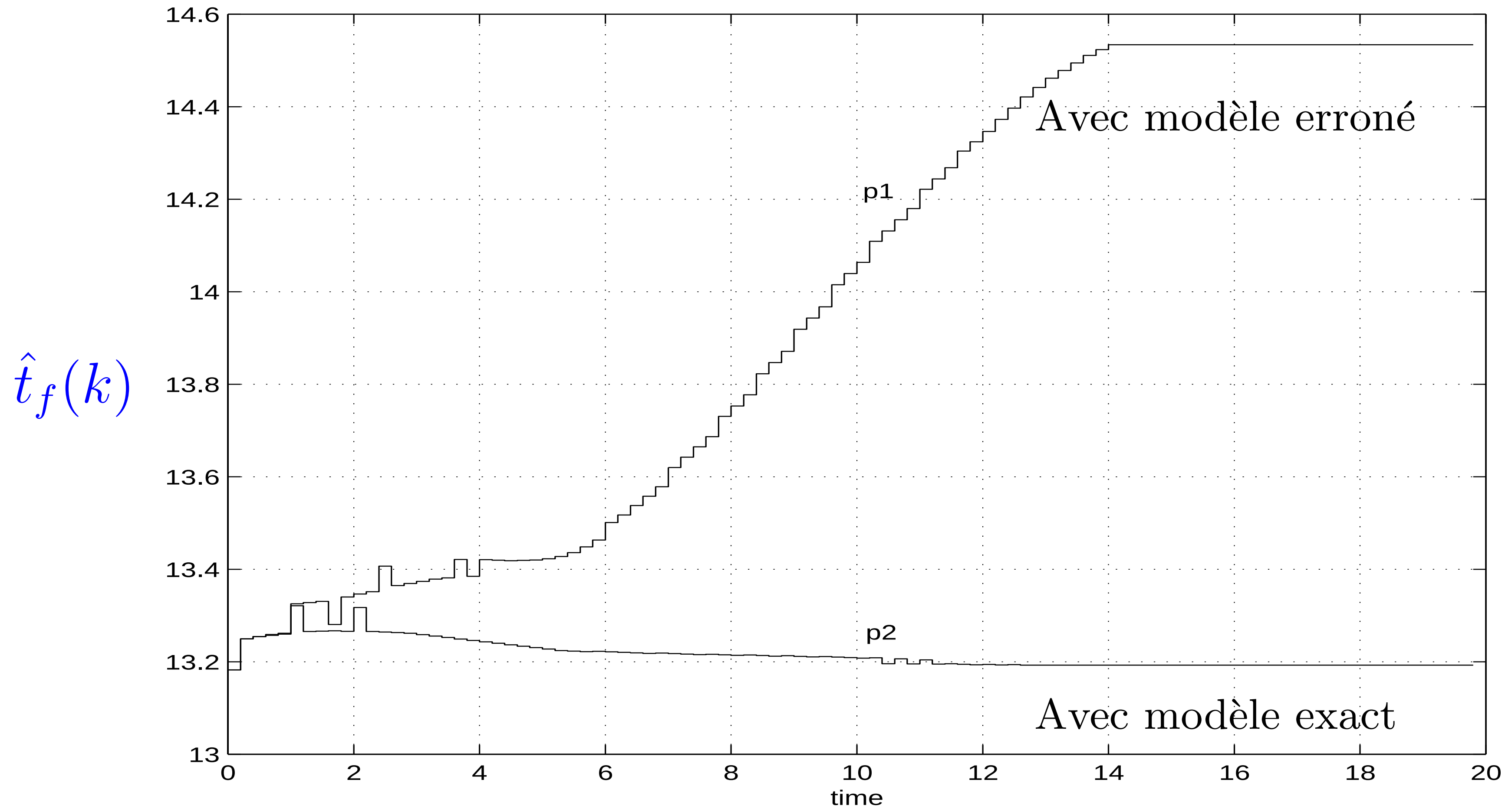


Temps de calul = 0.02
(Pentium III, 600 Mhz, en compilé)

Interception en temps minimal



Interception en temps minimal



Systèmes non-holonomes chaînés

$$
\begin{aligned}
\dot{x}_1 &= u_1 \\
\dot{x}_2 &= u_2 \\
\dot{x}_{i+1} &= x_i u_1 \quad i \in \{2, \dots, n-1\}
\end{aligned}
$$

avec

$$
-u_j^{max} \leq u_j \leq u_j^{max} \quad j \in \{1, 2\}
$$

Systèmes non-holonomes chaînés

$$\begin{aligned}
\dot{x}_1 &= u_1 \\
\dot{x}_2 &= u_2 \\
\dot{x}_{i+1} &= x_i u_1 \quad i \in \{2, \dots, n-1\}
\end{aligned}$$

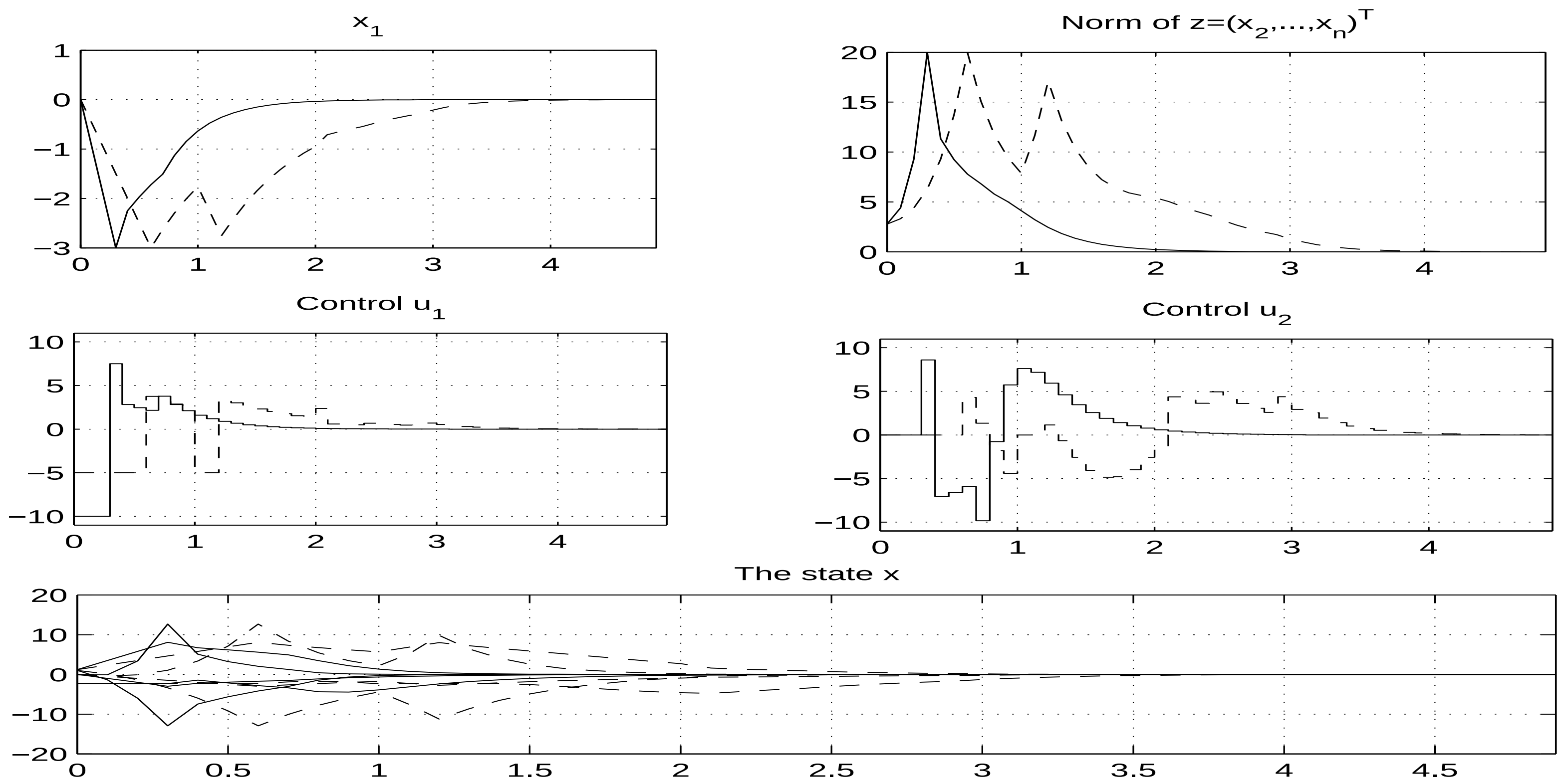
avec

$$-u_j^{max} \leq u_j \leq u_j^{max} \quad j \in \{1, 2\}$$

L'étude du problème de stabilisation en temps minimal **conduit à des arcs singuliers**

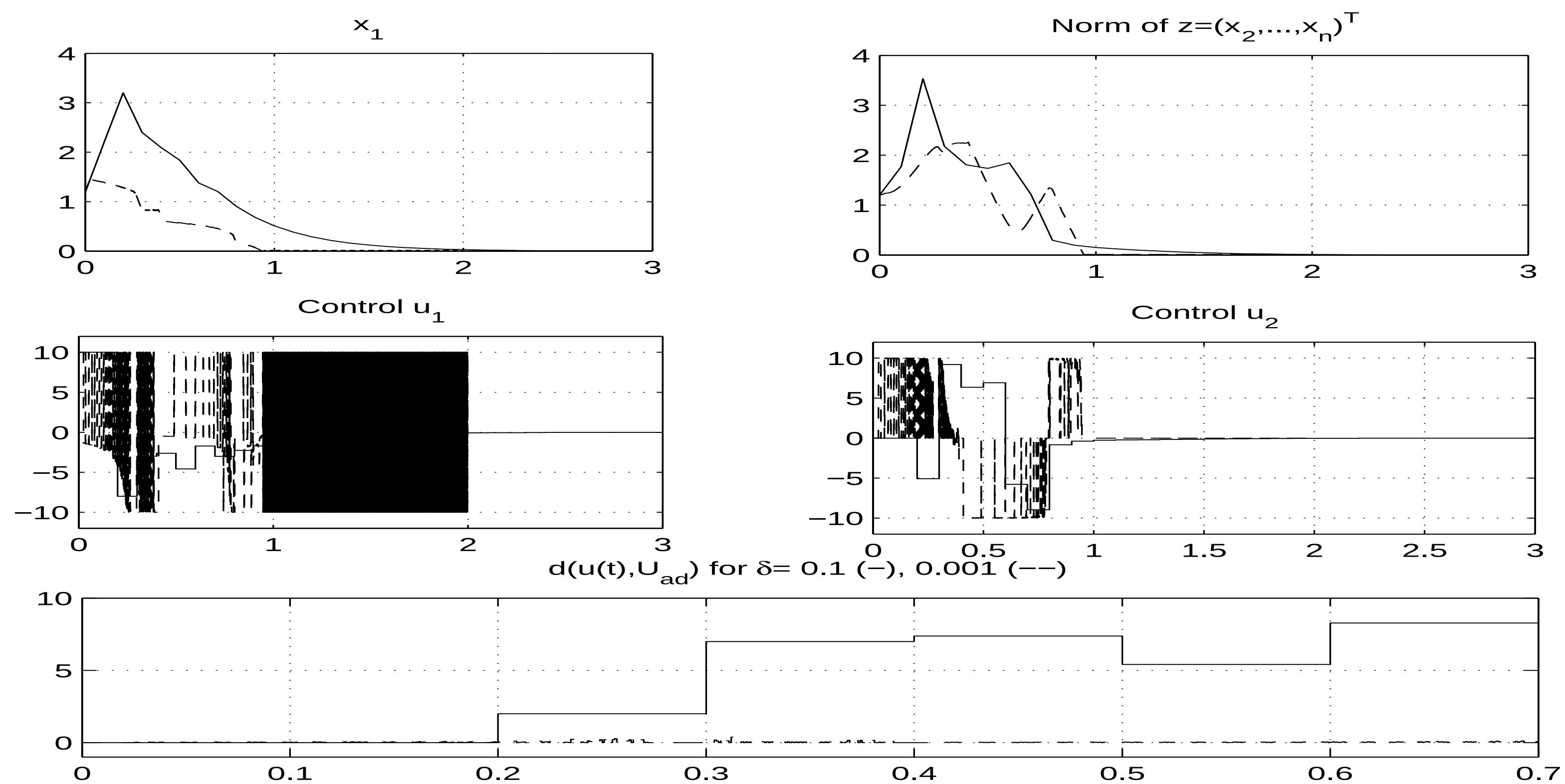
Systèmes non-holonomes chaînés

(- -) Saturation à 5 / (-) Saturation à 10



Systèmes non-holonomes chaînés

Lorsque $\delta \to 0$ la solution tend vers la frontière de $\mathcal{U}_{ad}$



Conclusion

- ✓ Pour les systèmes lents (NPC Bien établi)
  - Pousser à l'extrême la gestion des critère et des contraintes
  - Choisir le bon algorithme.
- ✓ Pour les systèmes rapides (NPC sous exploitée)
  - " *Trouver la faille*"
  - Choisir la bonne paramétrisation
  - Optimisation de dimension réduite
  - Sous optimalité réfléchie.