

# Méthode numérique pour le filtrage des systèmes linéaires à sauts markoviens

Eduardo F. Costa, Benoîte de Saporta

USP-COFECUB UcPh145-13  
Univ. São Paulo  
Inria CQFD, Univ. Bordeaux

# Plan de l'exposé

## Introduction

## Méthode numérique

- Reformulation du problème

- Quantification

- Construction d'une approximation

- Convergence

## Résultats numériques

- Implémentation

- Convergence

- Comparaison de performance

## Conclusion et perspectives

# Système linéaire à sauts markoviens

- ▶  $\theta(t)$  chaîne de Markov à espace d'états fini  $\{1, \dots, N\}$  de matrice génératrice  $\Lambda = (\lambda_{ij})$
- ▶  $v(t)$ ,  $w(t)$  mouvements Browniens standards indépendants et indépendants de  $\theta(t)$

$$\begin{aligned} dx(t) &= A_{\theta(t)}x(t)dt + B_{\theta(t)}u(t)dt + E_{\theta(t)}dw(t) \\ dy(t) &= C_{\theta(t)}x(t)dt + D_{\theta(t)}dv(t) \end{aligned}$$

$\theta(t)$ ,  $y(t)$  observés  $\longrightarrow$  choisir le contrôle  $u(t)$

Filtrer le processus

# Filtre de Kalman

$$\begin{aligned}
 d\hat{x}_K(t) &= A_{\theta(t)}\hat{x}_K(t)dt + K(t)(dy(t) - C_{\theta(t)}\hat{x}_K dt) \\
 \hat{x}_K(0) &= \mathbb{E}[x(0)]
 \end{aligned}$$

- ▶ sans biais
- ▶ minimise la variance de l'erreur

$$\mathbb{E}[\|x(t) - \hat{x}_K(t)\|^2] = \text{tr}(\mathbb{E}[P(t)])$$

$$P(t) = \mathbb{E}[(x(t) - \hat{x}_K(t))(x(t) - \hat{x}_K(t))' \mid \theta(s), 0 \leq s \leq t]$$

$$\text{si } K(t) = P(t)C'_{\theta(t)}(D_{\theta(t)}D'_{\theta(t)})^{-1}$$

## Variance de l'erreur

$$\text{Si } K(t) = P(t)C'_{\theta(t)}(D_{\theta(t)}D'_{\theta(t)})^{-1}$$

### Equation de Riccati

$$\begin{aligned} dP(t) &= A_{\theta(t)}P(t) + P(t)A'_{\theta(t)} + E_{\theta(t)}E'_{\theta(t)} \\ &\quad - P(t)C'_{\theta(t)}(D_{\theta(t)}D'_{\theta(t)})^{-1}C'_{\theta(t)}P(t) \\ P(0) &= \text{Var}[x(0)] \end{aligned}$$

Non calculable on-line : trouver de bonnes méthodes de pré-calcul

# Filtre de Fragoso-Costa

## Contrainte

$$\pi_i(t) = \mathbb{P}(\theta(t) = i) > 0 \text{ pour tous } t \text{ et } i$$

$$d\hat{x}_F(t) = A_{\theta(t)}\hat{x}_F(t)dt + F_{\theta(t)}(t)(dy(t) - C_{\theta(t)}\hat{x}_F(t)dt)$$

$$\hat{x}_F(0) = \mathbb{E}[x(0)]$$

Sans biais et minimise la variance de l'erreur  $\mathbb{E}[\|x(t) - \hat{x}_K(t)\|^2]$  sur la classe des filtres linéaires markoviens si pour tout  $i$

$$F_i(t) = M_i(t)C_i'(D_iD_i'\pi_i(t))^{-1} \text{ avec}$$

## Système pré-calculable d'équations de Riccati couplées

$$dM_i(t) = A_iM_i(t) + M_i(t)A_i' + \sum_{j=1}^N M_j(t)\lambda_{ji} + E_iE_i'\pi_i(t) \\ - M_i(t)C_i'(D_iD_i'\pi_i(t))^{-1}C_i'M_i(t)$$

$$M_i(0) = \text{Var}[x(0)]\pi_i(0)$$

## Exemple

## Paramètres

$$\theta(t) \in \{1, 2\}, \quad \Lambda = \begin{pmatrix} -2 & 2 \\ 0.01 & -0.01 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1750 & 0 & -34.1 \\ 4360.2 & 104.2 & -84.3 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1750 & 0 & -34.1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$E_1 = E_2 = \begin{pmatrix} 1 & 0.2 & -1.9 \\ -0.1 & 1.4 & -0.3 \\ 0.1 & 0.5 & 1 \end{pmatrix}$$

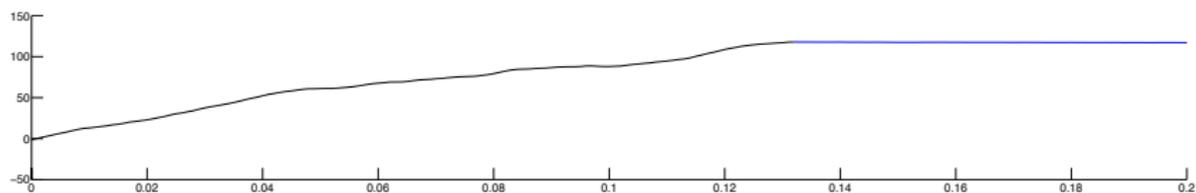
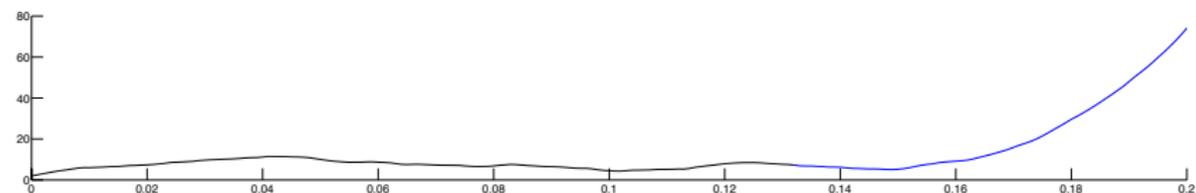
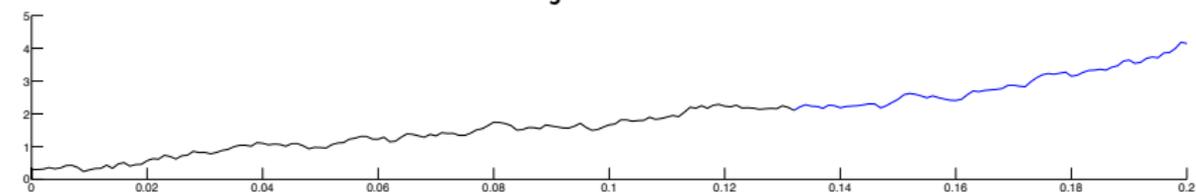
Seules les coordonnées 1 et 3 de  $x(t)$  sont observées

$$C_1 = C_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad D_1 = D_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

# Exemple

## Trajectoires filtrées

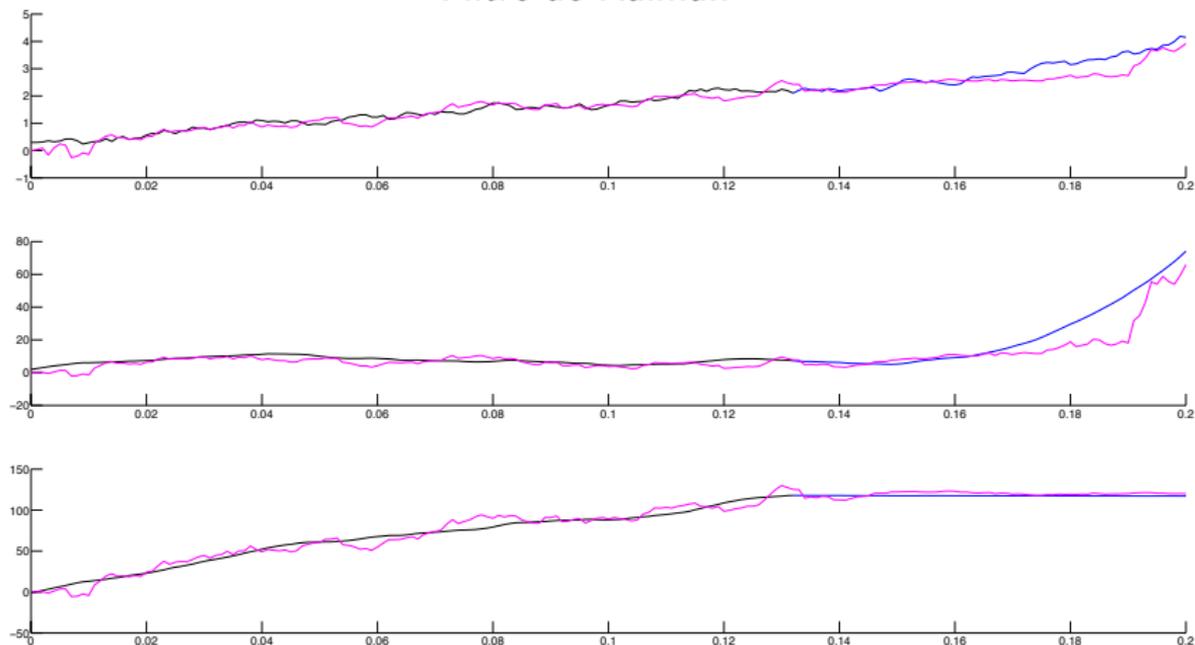
### Trajectoire



# Exemple

## Trajectoires filtrées

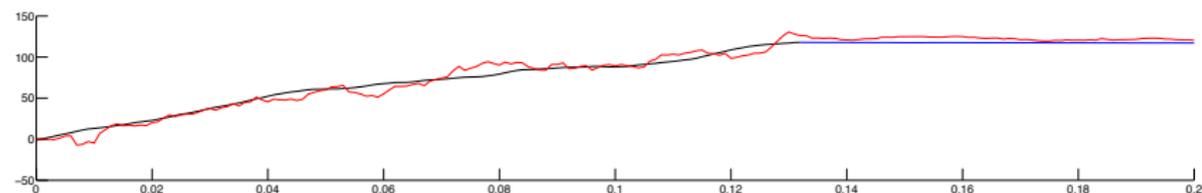
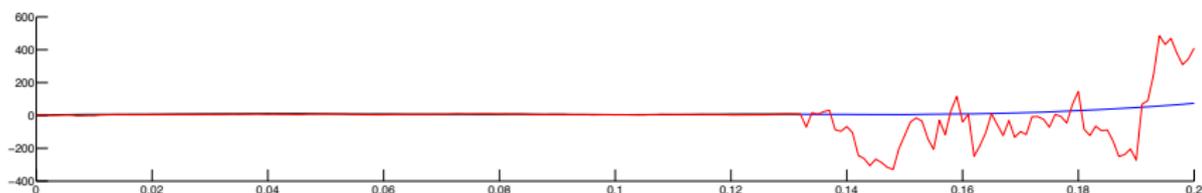
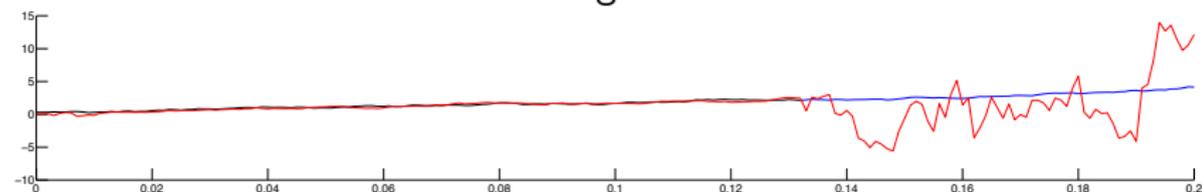
### Filtre de Kalman



# Exemple

## Trajectoires filtrées

### Filtre de Fragoso Costa



# Exemple

Performance ( $10^5$  Monte Carlo)

Filtre de Kalman

$$\mathbb{E} \left[ \int_0^{0.2} \|x(t) - \hat{x}_K(t)\|^2 \right] = 1484$$

Filtre de Fragoso Costa

$$\mathbb{E} \left[ \int_0^{0.2} \|x(t) - \hat{x}_F(t)\|^2 \right] = 3279$$

# Objectif

Proposer une méthode de **pré-calcul** plus efficace

- ▶ pré-calculer des trajectoires **typiques** de  $P(t)$
- ▶ **sélectionner** les trajectoires appropriées en ligne
- ▶ construire une approximation du **filtre de Kalman**

# Plan de l'exposé

Introduction

Méthode numérique

- Reformulation du problème

- Quantification

- Construction d'une approximation

- Convergence

Résultats numériques

Conclusion et perspectives

# Décomposition de la chaîne de Markov

- ▶  $T_n$  temps de sauts de  $\theta(t)$
- ▶  $S_n = T_n - T_{n-1}$  durées inter-saut
- ▶  $Z_n \in \{1, \dots, N\}$  position après les sauts

$$\theta(t) = \sum_{k=0}^{\infty} Z_k \mathbb{1}_{\{T_k \leq t < T_{k+1}\}} = \sum_{k=0}^{\infty} Z_k \mathbb{1}_{\{0 \leq t - T_k < S_{k+1}\}}.$$

# Décomposition de la solution de l'équation de Riccati

## Reformulation

$$P(t) = \sum_{k=0}^{\infty} P_k(t - T_k) \mathbb{1}_{\{0 \leq t - T_k < S_{k+1}\}}$$

Opérateur de Riccati **ne dépend pas du temps**

$$R(M, i) = A_i M + M A_i' + E_i E_i' - M C_i' (D_i D_i')^{-1} C_i M$$

$$\begin{cases} dP_0(t) = R(P_0(t), Z_0) \\ P_0(0) = \text{Var}[x(0)] \end{cases} \quad \begin{cases} dP_k(t) = R(P_k(t), Z_k) \\ P_k(0) = P_{k-1}(S_k) \end{cases}$$

# Construction de trajectoires typiques

## Principe

Remplacer  $S_k$  par une **approximation discrète** dans les équations

$$\begin{cases} dP_0(t) = R(P_0(t), Z_0) \\ P_0(0) = \text{Var}[x(0)] \end{cases} \quad \begin{cases} dP_k(t) = R(P_k(t), Z_k) \\ P_k(0) = P_{k-1}(S_k) \end{cases}$$

Nombre **fini** de trajectoires pré-calculables **représentatives** de toutes les trajectoires possibles

# Choix de la méthode de discrétisation

[Pagès 98], [Bally, Pagès 03], ...

Quantification d'une variable aléatoire  $X \in L^p(\mathbb{R}^d)$

Approcher  $X$  par  $\hat{X}$  à support fini pour minimiser  $\|X - \hat{X}\|_p$

- ▶ grille finie pondérée  $\Gamma$  avec  $|\Gamma| = K$
- ▶  $\hat{X} = p_\Gamma(X)$  projection au plus proche voisin

Propriétés asymptotiques

Si  $E[|X|^{p+\eta}] < +\infty$  alors

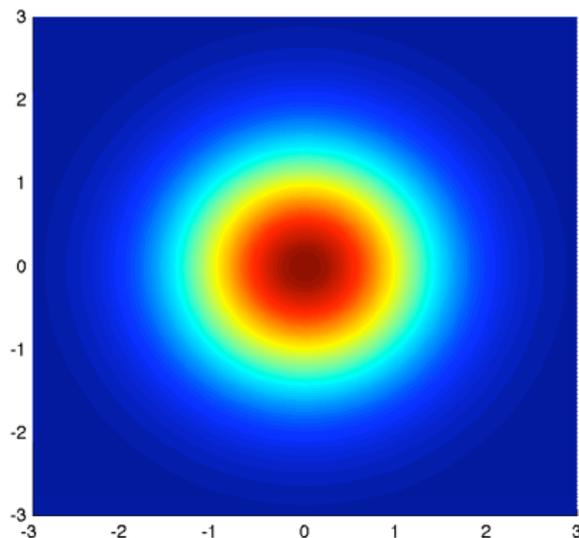
$$\min_{|\Gamma| \leq K} \|X - \hat{X}^\Gamma\|_p \simeq K^{-1/d}$$

# Algorithmes de quantification

Il existe des algorithmes donnant

- ▶ la grille  $\Gamma$
- ▶ la loi de  $\hat{X}$

Exemple:  $\mathcal{N}(0, I_2)$

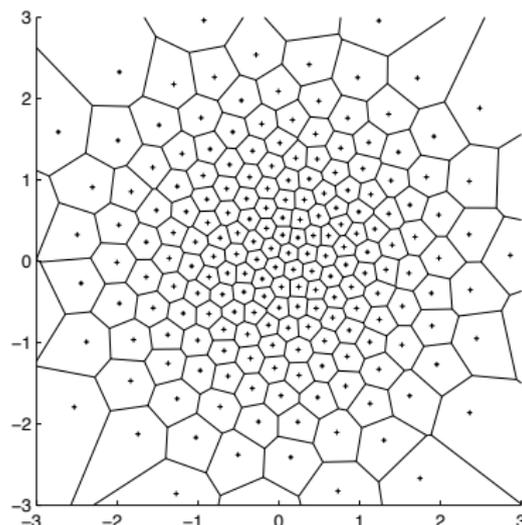


# Algorithmes de quantification

Il existe des algorithmes donnant

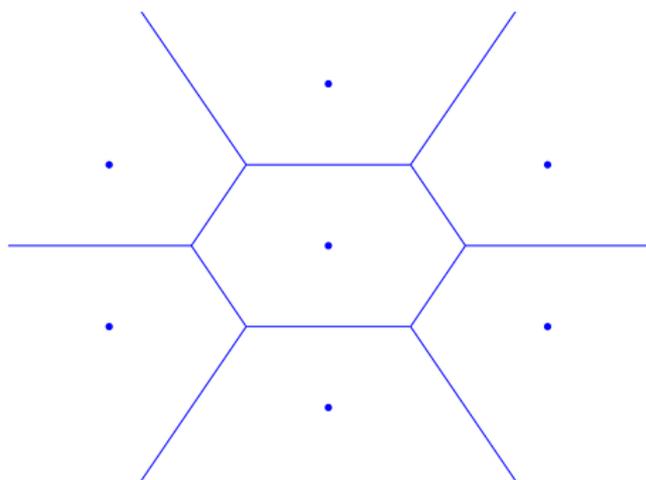
- ▶ la grille  $\Gamma$
- ▶ la loi de  $\hat{X}$

Exemple:  $\mathcal{N}(0, I_2)$



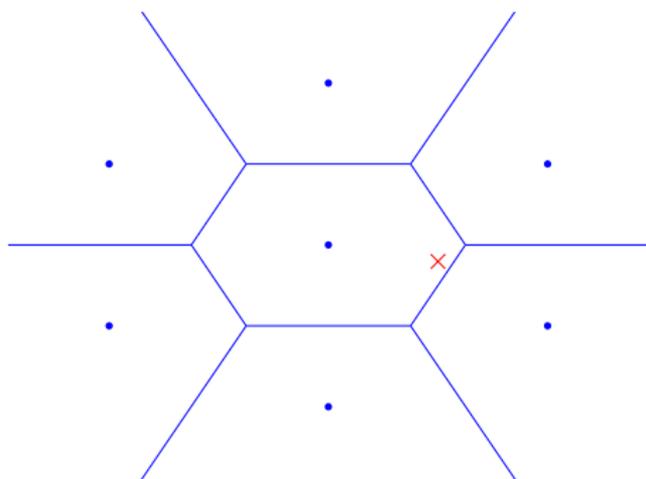
# Mise en oeuvre

Simulateur de trajectoires  $\longrightarrow$  grilles



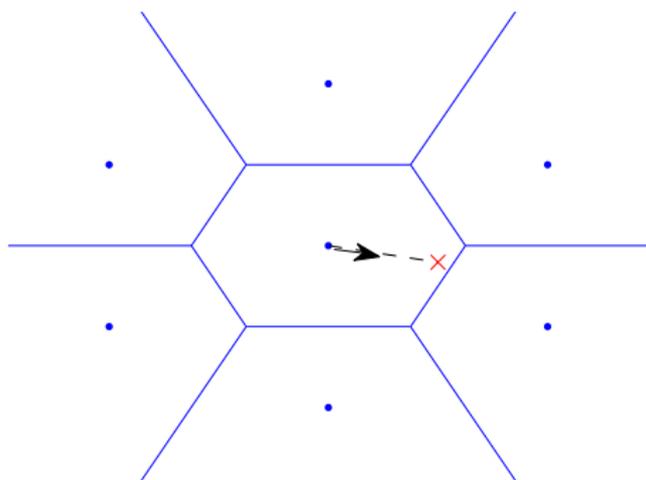
# Mise en oeuvre

Simulateur de trajectoires  $\longrightarrow$  grilles



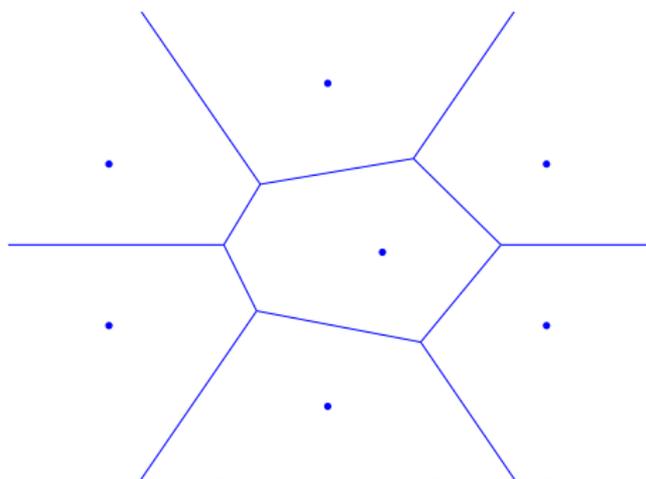
# Mise en oeuvre

Simulateur de trajectoires  $\longrightarrow$  grilles



# Mise en oeuvre

Simulateur de trajectoires  $\longrightarrow$  grilles



# Avantages et inconvénients de la quantification

## Avantages

- ▶ un simulateur de la loi cible suffit
- ▶ construction automatique des grilles
- ▶ vitesse de convergence pour  $\mathbb{E}[|f(X) - f(\hat{X})|]$  si  $f$  lipschitz

## Inconvénients

- ▶ temps de calcul des grilles
- ▶ fléau de la dimension
- ▶ questions ouvertes sur la convergence des algorithmes

# Exemple

Erreur de quantification

$$\theta(t) \in \{1, 2\}, \quad \Lambda = \begin{pmatrix} -2 & 2 \\ 0.01 & -0.01 \end{pmatrix}$$

Calcul de l'erreur  $\mathbb{E}_{\theta(0)}[|S_1 - \widehat{S}_1|^2]^{1/2}$

Nombre de points dans la grille	Erreur pour $\theta_0 = 1$	Erreur pour $\theta_0 = 2$
10	0.0525	10.8224
50	0.0152	2.7726
100	0.0086	1.8666
200	0.0039	1.0417
500	0.0017	0.3829
800	0.0013	0.2222
1000	0.0012	0.2209

# Construction de l'approximation de la variance

## Arbre des trajectoires type

### Trajectoires type

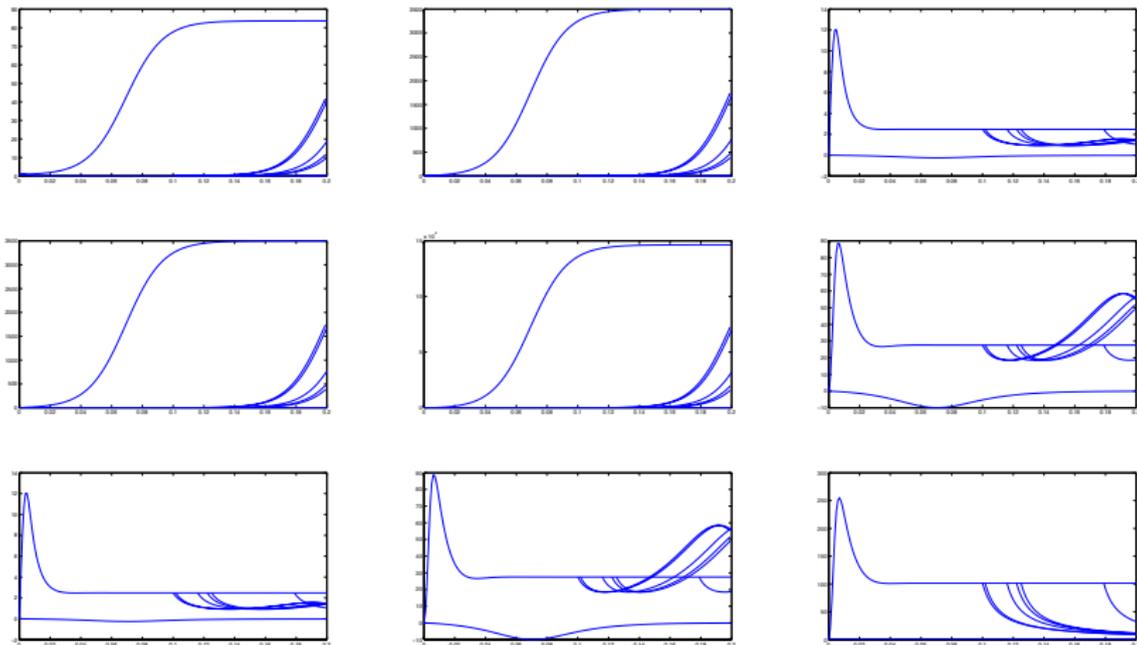
$$\left\{ \begin{array}{l} d\hat{P}_0(t) = R(\hat{P}_0(t), Z_0) \\ \hat{P}_0(0) = \text{Var}[x(0)] \end{array} \right. \quad \left\{ \begin{array}{l} d\hat{P}_k(t) = R(\hat{P}_k(t), Z_k) \\ \hat{P}_k(0) = \hat{P}_{k-1}(\hat{S}_k) \end{array} \right.$$

- ▶ en nombre **fini**
- ▶ croissance **exponentielle** avec le nombre de sauts

# Exemple

## Arbre des trajectoires type

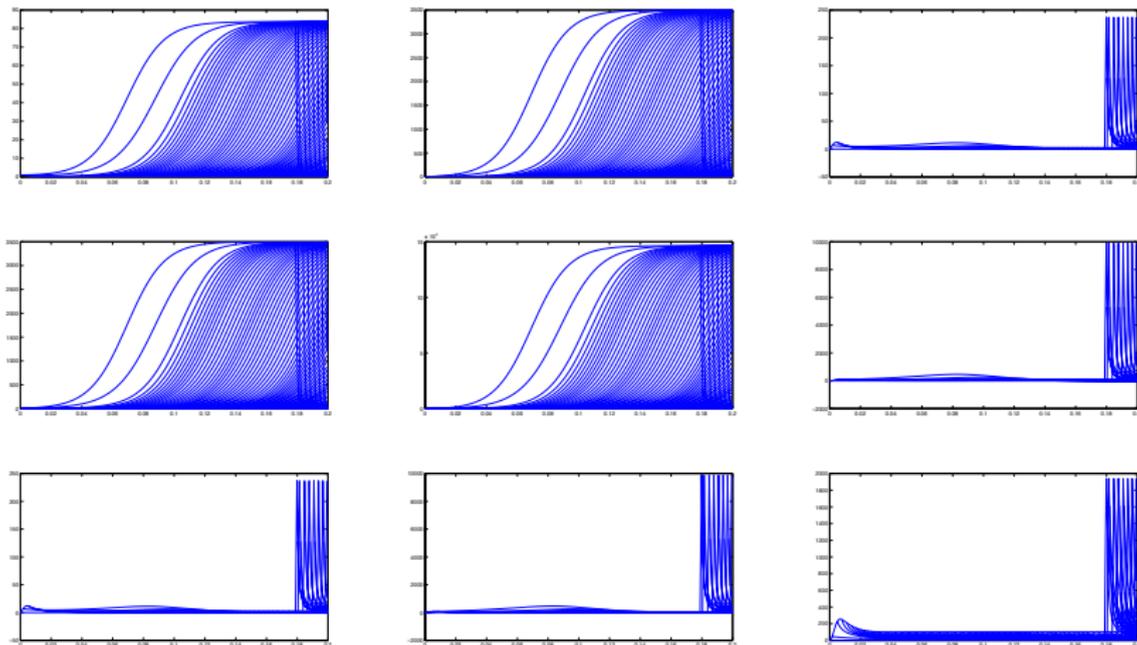
Arbre des trajectoires pré-calculées, 10 points



## Exemple

## Arbre des trajectoires type

Arbre des trajectoires pré-calculées, 200 points



# Construction de l'approximation de la variance

## Reconstruction en ligne

### Reconstruction en ligne

- ▶ pas de temps en ligne  $\delta t$
- ▶ prise en compte des sauts avec **décalage**

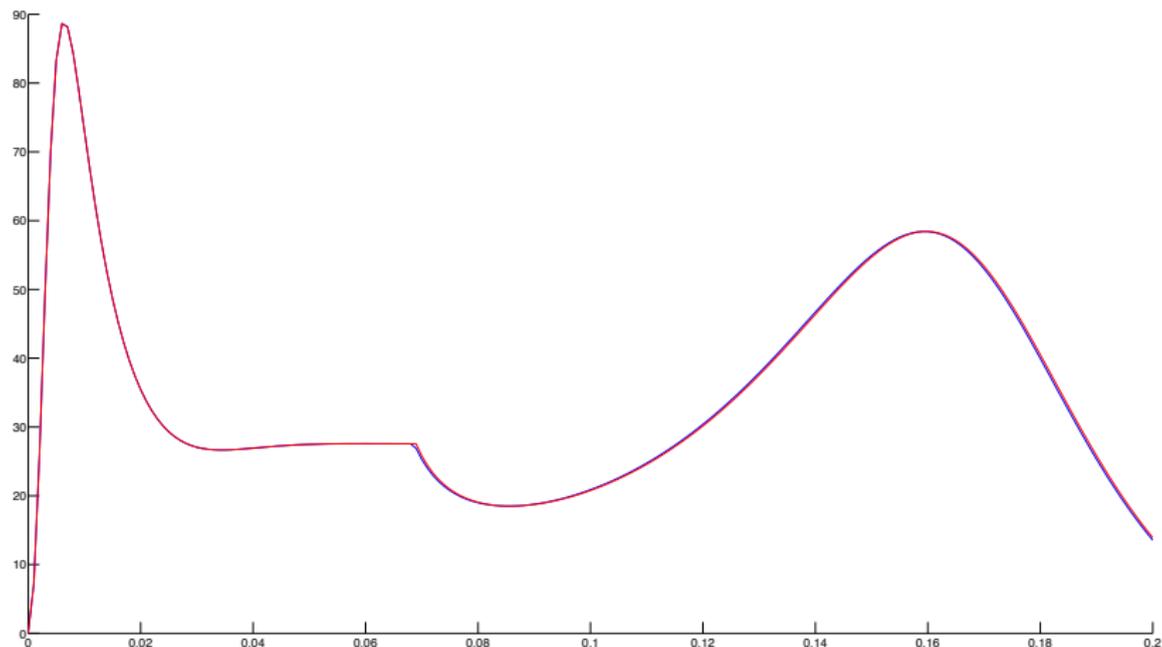
$$\tilde{T}_0 = 0, \quad \tilde{T}_k = \inf\{q; T_k < q\delta t\}\delta t, \quad \tilde{S}_k = \tilde{T}_k - \tilde{T}_{k-1}$$

$$\tilde{P}(t) = \sum_{k=0}^{\infty} \hat{P}_k(t - \tilde{T}_k) \mathbb{1}_{\{0 \leq t - \tilde{T}_k < \tilde{S}_{k+1}\}}$$

# Exemple

## Reconstruction en ligne

Coordonnée  $P_{23}(t)$  **bleu**: vraie valeur, **rouge**: valeur approchée



# Construction de l'approximation du filtre

Matrice de gain approchée

$$\tilde{K}(t) = \tilde{P}(t)C'_{\theta(t)}(D_{\theta(t)}D'_{\theta(t)})^{-1}$$

Filtre de Kalman approché

$$\begin{aligned} d\tilde{x}_K(t) &= A_{\theta(t)}\tilde{x}_K(t)dt + \tilde{K}(t)(dy(t) - C_{\theta(t)}\tilde{x}_K dt) \\ \tilde{x}_K(0) &= \mathbb{E}[x(0)] \end{aligned}$$

# Régularité des solutions

## Hypothèses

- ▶ Horizon de calcul fini  $T$
- ▶ Nombre  $n$  de sauts avant  $T$  fini (avec grande probabilité)
- ▶  $P(0)$  symétrique définie positive

## Solutions bornées

Il existe  $p$  symétrique définie positive telle que pour  $1 \leq k \leq n$  et  $0 \leq t \leq T$

$$P_k(t) \leq p$$

## Solutions Lipschitz

$\phi_i(q, t)$  solution de  $dM(t) = R(M(t), i)$  avec  $M(0) = q$   
pour tous  $i$ ,  $0 \leq t, t' \leq T$  et  $q, q' \leq p$

$$\|\phi_i(q, t) - \phi_i(q', t')\| \leq \ell \|q - q'\| + \eta |t - t'|.$$

# Convergence

## Vitesse de convergence pour Riccati

Pour  $t \geq 0$

$$\begin{aligned} & \|P(t \wedge T \wedge T_{n+1}) - \tilde{P}(t \wedge T \wedge T_{n+1})\|_2 \\ & \leq \sum_{j=0}^{n-1} \ell^{n-j} \eta \|S_{j+1} - \hat{S}_{j+1}\|_2 \\ & \quad + \eta \delta t + n \|p\| (1 - \exp(-\bar{\lambda} \delta t))^{1/2} \end{aligned}$$

avec  $\bar{\lambda} = \max\{-\lambda_{ii}\}$

## Vitesse de convergence pour le filtre approché

travail en cours

# Plan de l'exposé

Introduction

Méthode numérique

Résultats numériques

- Implémentation

- Convergence

- Comparaison de performance

Conclusion et perspectives

## Construction des grilles de quantification

- ▶ simulateur de loi exponentielle
- ▶ construction d'une grille de quantification du premier temps de saut pour chaque état initial  $\Gamma_i$ ,  $1 \leq i \leq N$
- ▶ enlever les points au-delà de l'horizon de calcul  $T$  et ajouter  $T$

### Nombre effectif de points utilisés

Nombre de points dans la grille initiale	Points $\leq T$ pour $\theta_0 = 1$	Points $\leq T$ pour $\theta_0 = 2$
10	7	1
50	18	1
100	34	2
200	79	2
500	169	2
800	274	2
1000	337	2

# Construction de l'arbre des trajectoires pré-calculées

Calculer récursivement  $\{Z_n, \hat{T}_n, \hat{S}_n, n, \text{compteur}, \hat{P}_{n-1}(\hat{S}_n)\}$  et  $\phi_{Z_n}(\hat{P}_{n-1}(\hat{S}_n), k\delta t)$  pour  $0 \leq k \leq \lceil T/\delta t \rceil$

## Initialisation

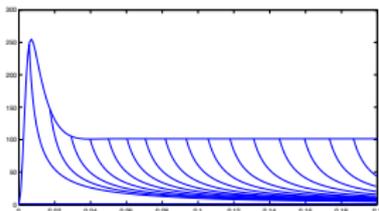
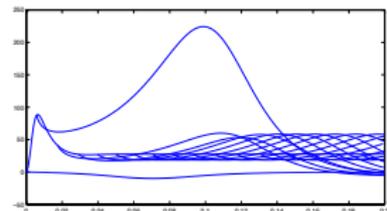
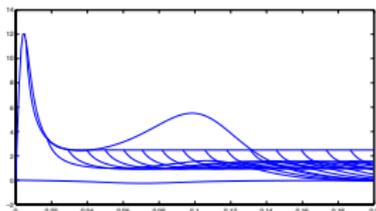
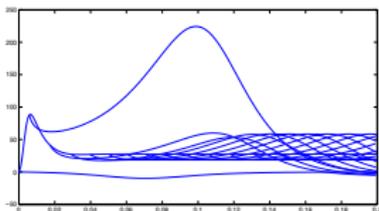
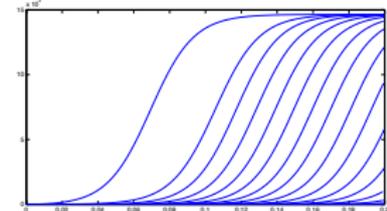
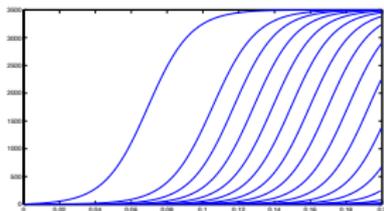
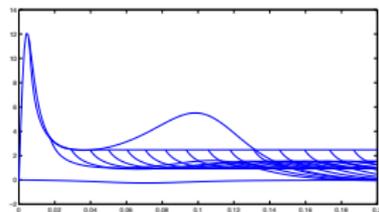
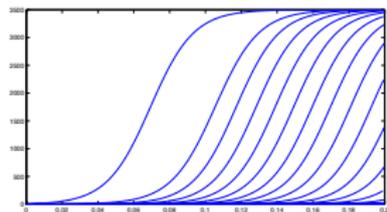
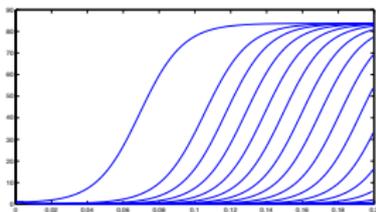
- Pour  $1 \leq i \leq N$ , stocker  $\{i, 0, 0, 0, i, \text{Var}[x(0)]\}$

## Boucle tant que stock non vide

- lire la première ligne du stock  $\{i, t, s, n, \text{cpt}, M\}$
- mémoriser  $\phi_i(M, k\delta t)$  pour  $0 \leq k \leq \lceil T/\delta t \rceil$
- si  $\Gamma_i(t) = \{s' \in \Gamma_i ; s' + t \leq T\} \neq \emptyset$ , pour tout  $s' \in \Gamma_i(t)$ 
  - calculer  $M' = \phi_i(M, s')$
  - pour  $1 \leq j \leq N, j \neq i$  stocker  $\{j, t + s', s', n + 1, \text{cpt}++, M'\}$
- ôter la première ligne du stock

# Exemple

## Arbre des trajectoires pré-calculées, 50points



## Exemple

## Nombre de branches pré-calculées

Nombre de points dans la grille initiale	Points $\leq T$ pour $\theta_0 = 1$	Points $\leq T$ pour $\theta_0 = 2$	Nombre de branches
10	7	1	10
50	18	1	21
100	34	2	63
200	79	2	120
500	169	2	1 293
800	274	2	10 745
1000	337	2	298 440

# Calcul du filtre en temps réel

## Initialisation

- ▶  $P(0) = \text{Var}[x(0)]$ ,  $K(0) = P(0)C'_{\theta(0)}(D_{\theta(0)}D'_{\theta(0)})^{-1}$
- ▶ lire la première branche pré-calculée correspondant à  $\theta(0)$

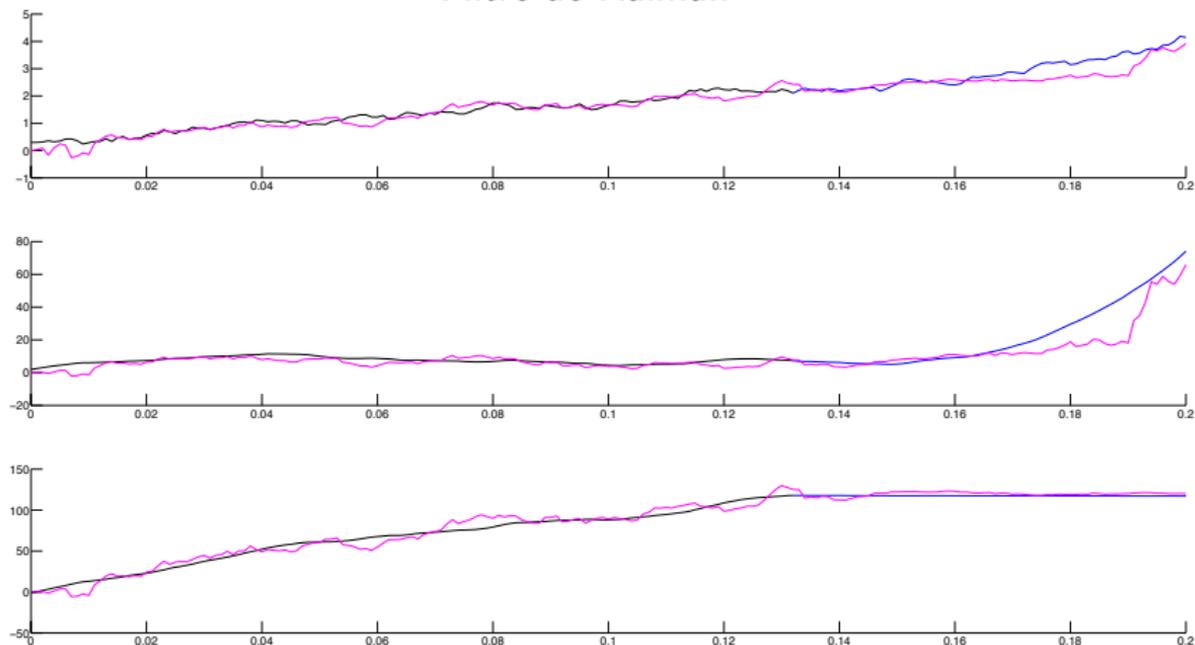
## Boucle pour $1 \leq k \leq [T/\delta t]$

- ▶ calculer  $\tilde{x}(k\delta t)$  avec la valeur pré-calculée de  $P$  et  $K$  en  $(k-1)\delta t$
- ▶ Si pas de saut de  $\theta$  entre  $(i-1)\delta t$  et  $i$ 
  - ▶ lire la valeur pré-calculée de  $P$  en  $k\delta t$ , calculer  $K$  en  $k\delta t$
- ▶ Sinon
  - ▶ chercher la branche pré-calculée correspondant à la nouvelle valeur de  $\theta$  et à la projection dans les grilles de quantification de la durée depuis le dernier saut
  - ▶ prendre la valeur initiale de cette branche pour  $P$  en  $k\delta t$ , calculer  $K$  correspondant en  $k\delta t$

# Exemple

## Trajectoires filtrées

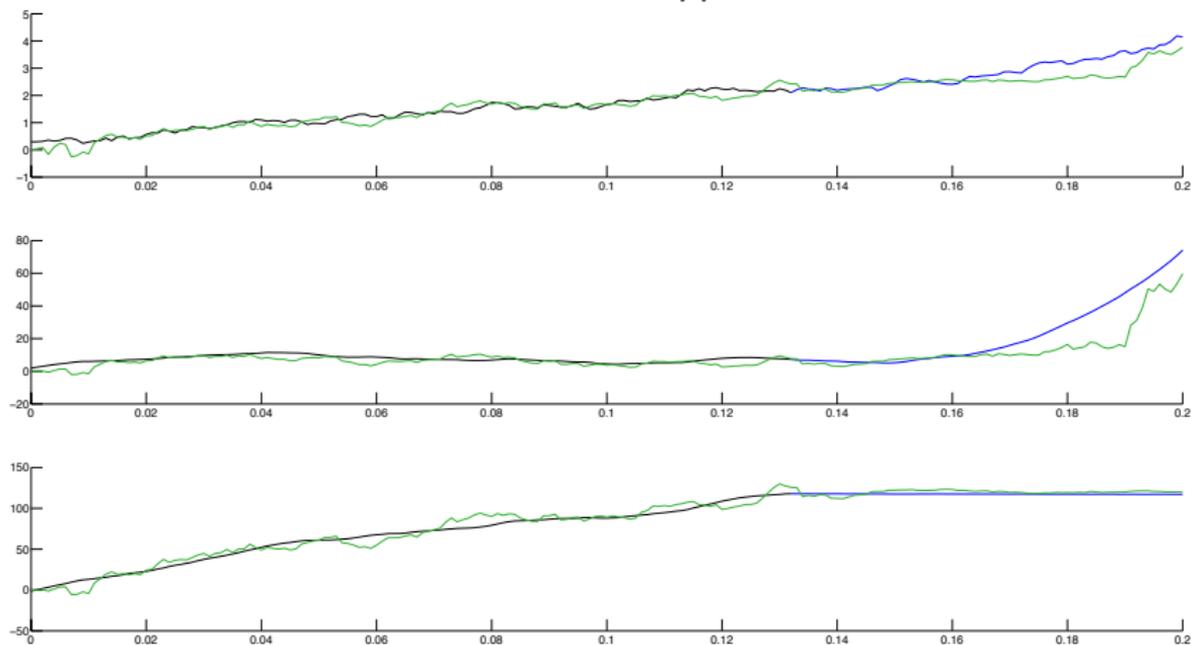
### Filtre de Kalman



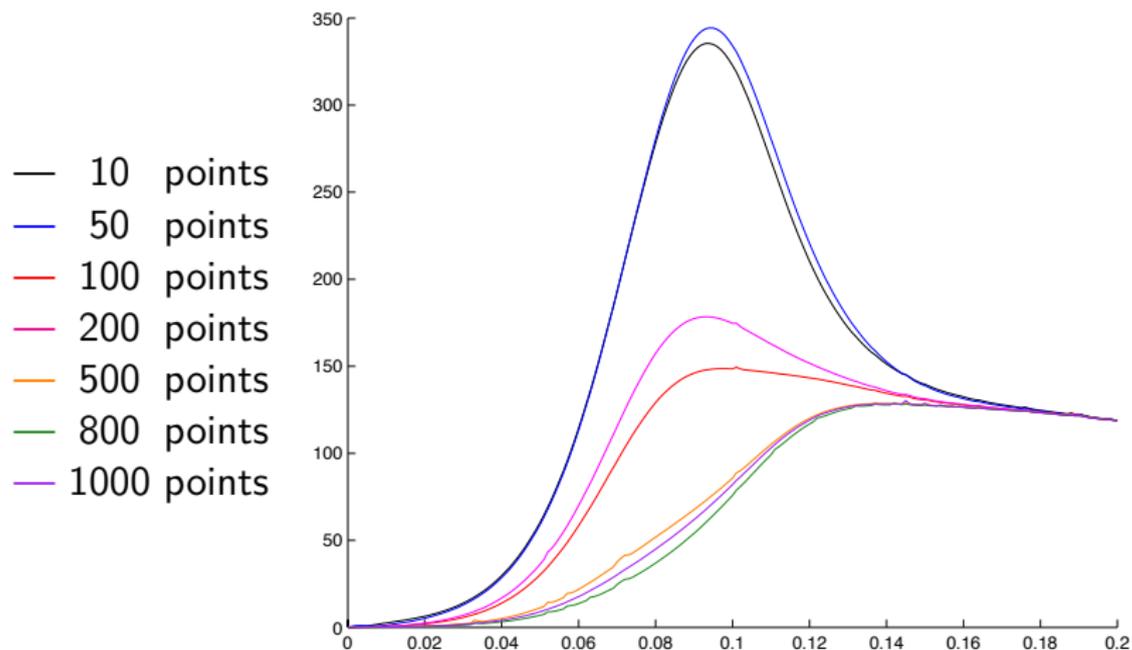
# Exemple

## Trajectoires filtrées

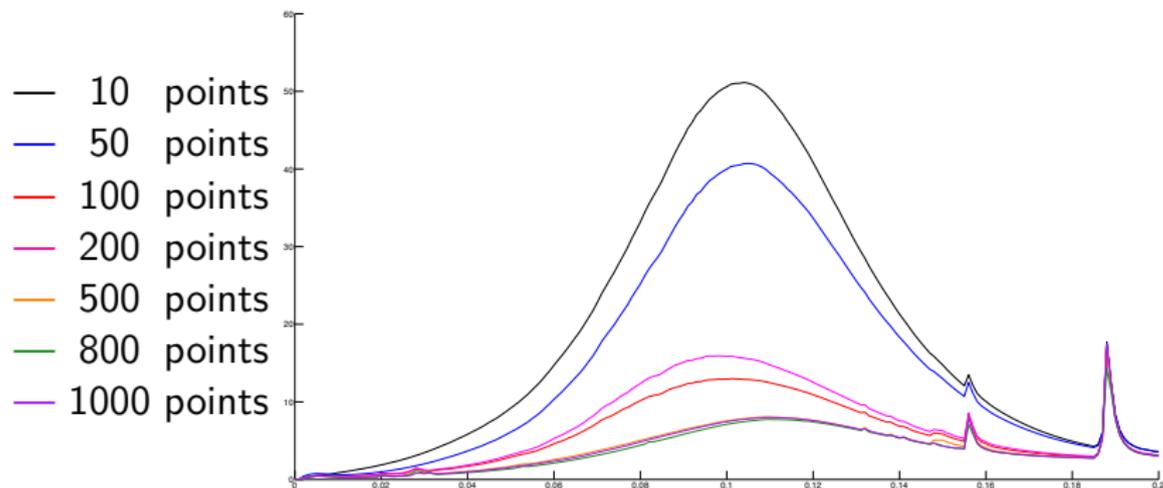
### Filtre de Kalman approché



# Erreur $\mathbb{E}[\|P(t) - \tilde{P}(t)\|]$



Erreur  $\mathbb{E}[\|\hat{x}(t) - \tilde{x}(t)\|^2]^{1/2}$



## Comparaison des erreurs intégrées

Nombre de points	Erreur Kalman	Erreur Kalman approché	Erreur Frago Costa
10	1484	1600	3279
50	1484	1551	3279
100	1484	1487	3279
200	1484	1490	3279
500	1484	1482	3279
800	1484	1481	3279
1000	1484	1481	3279

# Plan de l'exposé

Introduction

Méthode numérique

Résultats numériques

Conclusion et perspectives

# Conclusion et perspectives

## Avantages et inconvénients

- ▶ bonne approximation du filtre de Kalman réalisable en temps réel
- ▶ meilleurs résultats que le filtre de Fragoso-Costa même pour des grilles à peu de points
- ▶ un horizon correspondant à très peu de sauts possibles
- ▶ explosion des pré-calculs avec le nombre de sauts

## Améliorations possibles

- ▶ structures des branches stockées / branches inexistantes
- ▶ horizon mobile
- ▶ quantification de  $P(t)$  sur un horizon plus long

MERCI