

ECOLE NATIONALE SUPÉRIEURE D'ELECTRONIQUE,  
INFORMATIQUE, TÉLÉCOMMUNICATIONS, MATHÉMATIQUE ET  
MÉCANIQUE DE BORDEAUX



---

## Filtrage Optimal : Filtrage et lissage de Kalman

---

Elèves :

Pierre MINIER

Alexandre VALADE

Encadrant :

Eric GRIVEL

12 janvier 2023

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Simulation de trajectoires et filtrage de Kalman</b>	<b>1</b>
2.1	Modèle de trajectoires de Singer . . . . .	1
2.1.1	Mise en équation et implémentation . . . . .	1
2.1.2	Discussion sur le rôle du paramètre alpha . . . . .	2
2.2	Filtrage de Kalman . . . . .	2
2.3	Lissage du filtrage . . . . .	3
2.3.1	Motivations . . . . .	3
2.3.2	Algorithme . . . . .	3
2.3.3	Résultats . . . . .	4
2.4	Sensibilité du filtrage de Kalman aux paramètres . . . . .	4
2.4.1	Sensibilité de la covariance du bruit de mesure . . . . .	4
2.4.2	Sensibilité de la covariance du bruit de modèle . . . . .	4
2.4.3	Les deux cas simultanément . . . . .	4
<b>3</b>	<b>Organisation du TP</b>	<b>5</b>
<b>4</b>	<b>Conclusion</b>	<b>5</b>
<b>A</b>	<b>Figures</b>	<b>I</b>
<b>B</b>	<b>Code</b>	<b>V</b>

## 1 Introduction

Le filtre de Kalman est une technique de traitement du signal largement utilisée pour l'estimation de l'état d'un système dynamique soumis à des incertitudes. Il repose sur un modèle linéaire et gaussien de l'évolution du système, ainsi que sur des mesures observées de ce dernier. Grâce à sa capacité à intégrer de manière optimale ces deux sources d'information, le filtre de Kalman est capable de fournir des estimations de l'état du système avec une grande précision, même en présence de bruits de mesure ou d'incertitudes sur les paramètres du modèle. Sa flexibilité et sa robustesse en font un outil incontournable dans de nombreux domaines, allant du suivi de cibles à la reconnaissance de la parole.

Dans ce TP, nous utilisons le filtre de Kalman pour estimer une trajectoire synthétique de type Singer qui est soumise à du bruit. Nous expliquons d'abord comment ces trajectoires sont générées. Puis nous leur appliquons un filtrage et lissage de Kalman. Les résultats obtenus sont comparés avec les trajectoires réelles non bruitées afin d'évaluer les performances des traitements mis en place. Enfin, nous étudions la robustesse de ces traitements, en nous intéressant à l'impact de l'estimation du bruit de mesure, du bruit de modèle et de la vitesse de décroissance de la corrélation de l'accélération. Avant de conclure, nous présentons un bilan de l'organisation du TP au sein du groupe.

## 2 Simulation de trajectoires et filtrage de Kalman

### 2.1 Modèle de trajectoires de Singer

#### 2.1.1 Mise en équation et implémentation

Il est essentiel de disposer de signaux synthétiques et de connaître leurs propriétés afin de tester et mettre à l'épreuve le filtre de Kalman. Dans le cadre de ce TP, des trajectoires de type Singer sont utilisées. Ce modèle génère des trajectoires avec des coordonnées en  $x$  et en  $y$  décorrélées, permettant de simuler des mouvements complexes et manoeuvrants, contrairement aux mouvements rectilignes uniformes (MRU) ou uniformément accélérés (MUA) qui donnent lieu à des lignes droites.

Ce type de trajectoire se génère itérativement avec la mise à jour pas-à-pas du vecteur d'état. Les composantes représentent la position, la vitesse et l'accélération selon un axe, la mise à jour de ce vecteur est décrit par (1).

$$\mathbf{X}_k = \Phi_{k-1,k} \mathbf{X}_{k-1} + \mathbf{G}_k \mathbf{u}_k \quad (1)$$

où  $\Phi_{k-1,k}$  est la matrice de passage de l'état  $k-1$  à l'état  $k$ ,  $\mathbf{G}_k$  est un vecteur de gain pour l'état  $k$  et  $\mathbf{u}_k$  est le bruit de modèle pour l'état  $k$ . Lors des simulations,  $\mathbf{G}_k$  est assimilé à la matrice identité.

La matrice permettant le passage de l'état  $k-1$  à l'état  $k$  est définie par (2).

$$\Phi_{k-1,k} = \begin{pmatrix} 1 & T & \frac{1}{\alpha^2} (-1 + \alpha T + e^{-\alpha T}) \\ 0 & 1 & \frac{1}{\alpha} (1 - e^{-\alpha T}) \\ 0 & 0 & e^{-\alpha T} \end{pmatrix} \quad (2)$$

Le bruit de modèle  $\mathbf{u}_k$  est supposé gaussien centré et de matrice de covariance  $\mathbf{Q}$  définie par (3).

$$\mathbf{Q} = 2\alpha\sigma_m^2 \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \quad (3)$$

$$\text{avec} \quad \begin{cases} q_{11} = \frac{1}{2\alpha^5} \left( 2\alpha T - 2\alpha^2 T^2 + 2\alpha^3 \frac{T^3}{3} - 4\alpha T e^{-\alpha T} - e^{-2\alpha T} + 1 \right) \\ q_{12} = \frac{1}{2\alpha^4} (\alpha^2 T^2 + 1 + e^{-2\alpha T} + e^{-\alpha T} (-2 + 2\alpha T) - 2\alpha T) \\ q_{13} = \frac{1}{2\alpha^3} (1 - 2\alpha T e^{-\alpha T} - e^{-2\alpha T}) \end{cases}$$

$$\text{et } \begin{cases} q_{22} = \frac{1}{2\alpha^3} (2\alpha T - 3 + 4e^{-\alpha T} - e^{-2\alpha T}) \\ q_{23} = \frac{1}{2\alpha^2} (1 - e^{-\alpha T})^2 \\ q_{33} = -\frac{1}{2\alpha} (e^{-2\alpha T} - 1) \end{cases}$$

MATLAB ne permet de générer que des bruits blancs gaussiens dont la matrice de covariance est la matrice identité. Pour donner la bonne covariance aux vecteurs gaussiens générés, il est nécessaire de passer par le facteur de Cholevsky de  $\mathbf{Q}$ . La fonction CHOL de MATLAB retourne la matrice  $\mathbf{A}$  telle que :

$$\mathbf{Q} = \mathbf{A}^T \mathbf{A} \quad (4)$$

Mais attention, car la matrice  $\tilde{\mathbf{A}}$  recherchée est telle que :

$$\begin{aligned} \mathbb{E}[\mathbf{u}_k \mathbf{u}_k^*] &= \tilde{\mathbf{A}} \mathbb{E}[\mathbf{w}_k \mathbf{w}_k^*] \tilde{\mathbf{A}}^T \\ &= \tilde{\mathbf{A}} \tilde{\mathbf{A}}^T \\ &= \mathbf{Q} \end{aligned} \quad (5)$$

avec  $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{I})$ .

Les équations (4) et (5) indiquent qu'il faut utiliser la transposée de la matrice  $\mathbf{A}$  calculée par CHOL afin de générer le bon bruit de modèle.

### 2.1.2 Discussion sur le rôle du paramètre $\alpha$

Pour une valeur de  $\alpha$  donné, des trajectoires de Singer peuvent être simulées. Des exemples sont donnés en annexe sur les figures 1a, 1b et 1c.

Sur ces figures, plus  $\alpha$  est grand, et plus les trajectoires semblent rester proches de l'origine. Cette observation est confirmée par la figure 2. Elle donne la distance moyenne parcourue par un ensemble trajectoire, et la distance la plus éloigné de l'origine par cet ensemble, pour différentes valeurs de  $\alpha$ . On remarque d'ailleurs que ces deux courbes se rapprochent à mesure que  $\alpha$  croît. Les paramètres pris pour ces courbes sont  $N = 500$ ,  $T = 1s$ . La variance du bruit de modèle vaut  $\sigma_x^2 = 1.2$  pour les coordonnées selon l'axe des  $x$  et  $\sigma_y^2 = 0.9$  selon l'axe des  $y$ .

L'effet du paramètre  $\alpha$  s'observe aussi en examinant de l'évolution de l'accélération dans le temps. La comparaison des figures 3a et 3b montre qu'augmenter  $\alpha$  augmente la fréquence à laquelle l'accélération varie. Cette observation est conforme au fait que  $\alpha$  contrôle la vitesse de décroissance de l'autocorrélation de l'accélération. En effet, sur la figure 4a on remarque que l'autocorrélation de l'accélération diminue lentement lorsque  $\alpha$  est faible, tandis que sur la figure 4b une forte valeur de  $\alpha$  produit un pic en 0, et presque aucune corrélation autour.

## 2.2 Filtrage de Kalman

Le filtrage de Kalman est ici mis en place pour estimer une trajectoire de type Singer soumise à un bruit. Pour ce faire, le modèle d'observation (6) est utilisé.

$$\mathbf{Y}_k = \mathbf{H} \mathbf{X}_k + \mathbf{V}_k \quad (6)$$

avec  $\mathbf{Y}$  est le vecteur des observations,  $\mathbf{X}$  le vecteur d'état,  $\mathbf{H}$  la matrice d'observation et  $\mathbf{V}$  un bruit de mesure supposé gaussien.

L'objectif est l'estimation du vecteur d'état  $\hat{\mathbf{X}}_{k|k}$ . Le processus est itératif, c'est-pourquoi l'estimation à l'instant  $k$  requiert les valeurs calculées à l'instant  $k - 1$ . Le filtrage se décompose donc en une étape de prédiction, et en une étape de remise à jour.

### Etape de prédiction

Tout d'abord, le vecteur d'état *a priori* est estimé grâce à la matrice de passage  $\Phi_{k-1,k}$  et à l'estimation du vecteur d'état à l'instant précédent  $\hat{\mathbf{X}}_{k-1|k-1}$ . L'équation (7) est la prédiction de ce vecteur d'état à l'instant  $k$ .

$$\hat{\mathbf{X}}_{k|k-1} = \Phi_{k-1,k} \hat{\mathbf{X}}_{k-1|k-1} \quad (7)$$

La matrice de covariance de l'erreur *a priori*  $\mathbf{P}_{k|k-1}$  est donnée par la formule (8).

$$\begin{aligned} \mathbf{P}_{k|k-1} &= \Phi_{k-1,k} \mathbf{P}_{k-1|k-1} \Phi_{k-1,k}^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \\ &= \Phi_{k-1,k} \mathbf{P}_{k-1|k-1} \Phi_{k-1,k}^T + \mathbf{Q}_k \end{aligned} \quad (8)$$

### Etape de remise à jour

La remise à jour utilise l'observation à l'instant  $k$ , notée  $\mathbf{Y}_k$ , pour affiner la prédiction *a priori*. Cette étape commence par l'estimation du gain de Kalman  $\mathbf{K}_k$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (9)$$

où  $\mathbf{R}_k$  désigne la matrice de covariance des observations à l'instant  $k$ . L'équation (9) ajuste la valeur du gain  $\mathbf{K}_k$  pour « minimiser l'erreur quadratique moyenne commise sur l'estimation du vecteur d'état. » Ce gain est ensuite utilisé lors de la mise à jour de l'estimation du vecteur d'état *a posteriori* (10).

$$\hat{\mathbf{X}}_{k|k} = \hat{\mathbf{X}}_{k|k-1} + \mathbf{K}_k (\mathbf{Y}_k - \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1}) \quad (10)$$

Cette nouvelle estimation du vecteur d'état permet de mettre à jour l'estimation de la matrice d'autocorrélation de l'erreur *a posteriori* (8), pour l'itération suivante.

$$\mathbf{P}_{k|k} = (\mathbf{I}_k - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (11)$$

Les figures correspondant à la trajectoire synthétisée 5a, à sa mesure bruitée 5b et au filtrage de Kalman associé 6a sont consultables en annexe. La trajectoire estimée semble assez fidèle à la trajectoire réelle, mais un phénomène en dents de scie apparaît à cause du bruit de mesure. Ce phénomène s'estompe vers la fin de la trajectoire, à mesure que le filtre « accroche » la trajectoire.

## 2.3 Lissage du filtrage

### 2.3.1 Motivations

Les trajectoires obtenues après filtrage manquent encore de résolution : elles oscillent autour de la trajectoire réelle. Un lissage est alors appliqué. Contrairement au filtrage, cette procédure tire profit de l'ensemble des échantillons car il est effectué après l'acquisition de toutes les données. Cet algorithme est rétrograde. C'est-à-dire qu'il est initialisé avec la dernière valeur du filtrage, la plus précise, pour remonter vers les premières valeurs, moins précises.

### 2.3.2 Algorithme

Trois équations décrivent cette étape. La matrice  $\mathbf{A}$  définie en (12) permet de factoriser les calculs et de rendre plus lisible les deux autres équations de mise à jour. Elle dépend de l'apostériori mis-à-jour de l'instant suivant. L'équation (13) met à jour les informations portées par l'apostériori  $\mathbf{P}(k|k)$  avec les instants suivant  $k$ .

$$\mathbf{A}(k) = \mathbf{P}(k|k) \Phi^T \mathbf{P}(k+1|k)^{-1} \quad (12)$$

$$\mathbf{P}(k|N) = \mathbf{P}(k|k) + \mathbf{A}(k)(\mathbf{P}(k+1|N) - \mathbf{P}(k+1|k)) \mathbf{A}(k)^T \quad (13)$$

L'équation (14) rectifie la position  $\hat{\mathbf{X}}_{k|k}$  à partir de l'ensemble des informations déjà corrigées.

$$\hat{\mathbf{X}}_{k|N} = \hat{\mathbf{X}}_{k|k} + \mathbf{A}(k)(\hat{\mathbf{X}}_{k+1|N} - \Phi \hat{\mathbf{X}}_{k|k}) \quad (14)$$

### 2.3.3 Résultats

Un exemple de trajectoire lissée est fournis sur les figures 6a et 6b. La figure 6b est le lissage appliqué à la figure 6a.

Pour évaluer l'amélioration apportée par le lissage, on calcule la différence entre la trajectoire réelle non bruitée et les trajectoires filtrée et lissée en utilisant la norme 2. Afin de ne pas être dépendant d'une seule réalisation, ces calculs sont répétés 1000 fois. On constate alors une diminution de l'erreur de 33,2 % après lissage.

## 2.4 Sensibilité du filtrage de Kalman aux paramètres

Les paramètres générant les trajectoires de Singer sont parfaitement connus. Il est donc possible de tester la robustesse du filtrage de Kalman en dégradant la qualité des informations qui lui sont donnés, et de comparer ce résultat à celui du filtrage de Kalman ayant reçus les véritables paramètres.

La même trajectoire de Singer sera utilisée pour toutes ces simulations.

### 2.4.1 Sensibilité de la covariance du bruit de mesure

Dans cette section, la matrice de covariance du bruit de mesure est erronée. L'erreur est générée en modifiant la valeur originale de la variance  $\sigma^2$  du bruit additif. La Figure 7 considère des valeurs allant de 20% à 180% de la vraie valeur  $\sigma^2$ . La ligne verticale orange délimite les régimes de sous et sur-estimation de  $\sigma^2$ .

Dans les métriques étudiées (position, vitesse, accélération), le même comportement s'observe. Le régime en sous-estimation est convexe et devient rapidement importante. Alors que le régime en sur-estimation est concave et augmente moins rapidement. On peut en conclure qu'il vaut mieux sur-estimer que sous-estimer la variance du bruit de mesure.

### 2.4.2 Sensibilité de la covariance du bruit de modèle

Dans cette section, la matrice de covariance du bruit de modèle est erronée. Le biais est généré via la valeur du paramètre  $\alpha$  qui varie de 20% à 180% de la valeur ayant servie à la génération de la trajectoire étudiée. Le résultat de la simulation est donné sur la Figure 8, et la ligne verticale orange représente la transition entre les deux régimes.

Les mêmes observations présentées à la section précédente peuvent être dressées. On note tout de même que le régime de sur-estimation est davantage concave : limitant ainsi les erreurs commises.

### 2.4.3 Les deux cas simultanément

Dans cette section, les matrices de covariance de bruit de mesure et de modèle sont erronées simultanément. Chacune de ces matrices sont altérées de la même manière que présenté dans les deux sections précédents, avec des valeurs allant de 20% à 180% d'un paramètre les définissant ( $\sigma^2$  et  $\alpha$ ) lors de la simulation de la trajectoire.

La Figure 9 donne en abscisse les variations de  $\sigma^2$ , en ordonnée les variations de  $\alpha$ , et la couleur du pixel associé l'erreur quadratique commise. Deux segments rouges séparent les quatre régimes différents : sur/sous-estimation de  $\sigma^2$  et de  $\alpha$ .

Deux observations sont à faire sur ces images :

1. Concernant la ligne bleu foncée parcourant les images et passant par leur centre. Elle correspond à la zone où l'erreur quadratique est minimale. On en déduit qu'il existe des couples de valeurs  $(\sigma^2, \alpha)$  erronés qui ne produisent pas ou très peu d'erreur par rapport au filtrage de Kalman ayant les bonnes valeurs. Comme cette zone est une courbe continue, à toute valeur erronée de  $\sigma^2$ , il existe une bonne valeur erronée de  $\alpha$  permettant de retrouver les performances fournies par les vrais paramètres de la trajectoire.
2. Concernant les régimes identifiés par les segments oranges. Le régime générant le moins d'erreur est le régime où les deux paramètres  $(\sigma^2, \alpha)$  sont sur-estimés car cette région est en moyenne la plus proche de la teinte bleu foncée.

### 3 Organisation du TP

Tâches	Étudiants	
	Alexandre	Pierre
Trajectoire de Singer	60 %	40 %
Rôle du paramètre $\alpha$	80 %	20 %
Implémentation du filtrage	50 %	50 %
Implémentation du lissage	40 %	60 %
Sensibilité du filtrage	20 %	80 %
Rapport	50 %	50 %

TABLE 1 – Répartition du travail

### 4 Conclusion

En résumé, nous avons utilisé le modèle de trajectoires de Singer pour évaluer l'efficacité du filtrage de Kalman lorsque les données sont bruitées. Ce modèle permet de générer des trajectoires dont les coordonnées dans l'espace sont décorréliées, ce qui permet de simuler des objets manoeuvrants. Ce modèle nous a donc permis de tester l'efficacité du filtrage de Kalman à prédire une trajectoire manoeuvrante. Nous avons constaté que les performances sont améliorées de plus de 30% lorsque le lissage de Kalman, qui prend en compte l'ensemble des informations sur la trajectoire, est utilisé en plus du filtrage de Kalman.

Nous avons également étudié la robustesse du filtrage de Kalman en donnant des informations erronées sur les paramètres au filtre. Nous avons découvert que sur-estimer la variance du bruit de mesure et le coefficient  $\alpha$  est préférable à leur sous-estimation. De plus, pour toute variance erronée, il existe un coefficient  $\alpha$  qui permet de retrouver les performances du filtrage de Kalman avec les bons paramètres.

## A Figures

### Étude des trajectoires de Singer

#### Génération de trajectoires

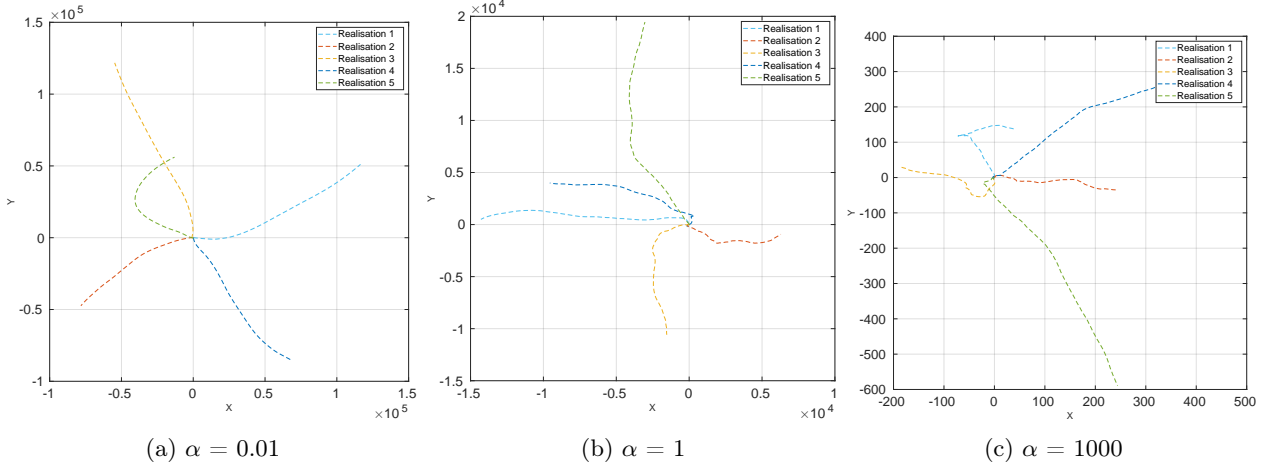


FIGURE 1 – Plusieurs réalisations de trajectoires de Singer pour des valeurs de  $\alpha$  croissantes.

#### Impact de $\alpha$ sur la trajectoire

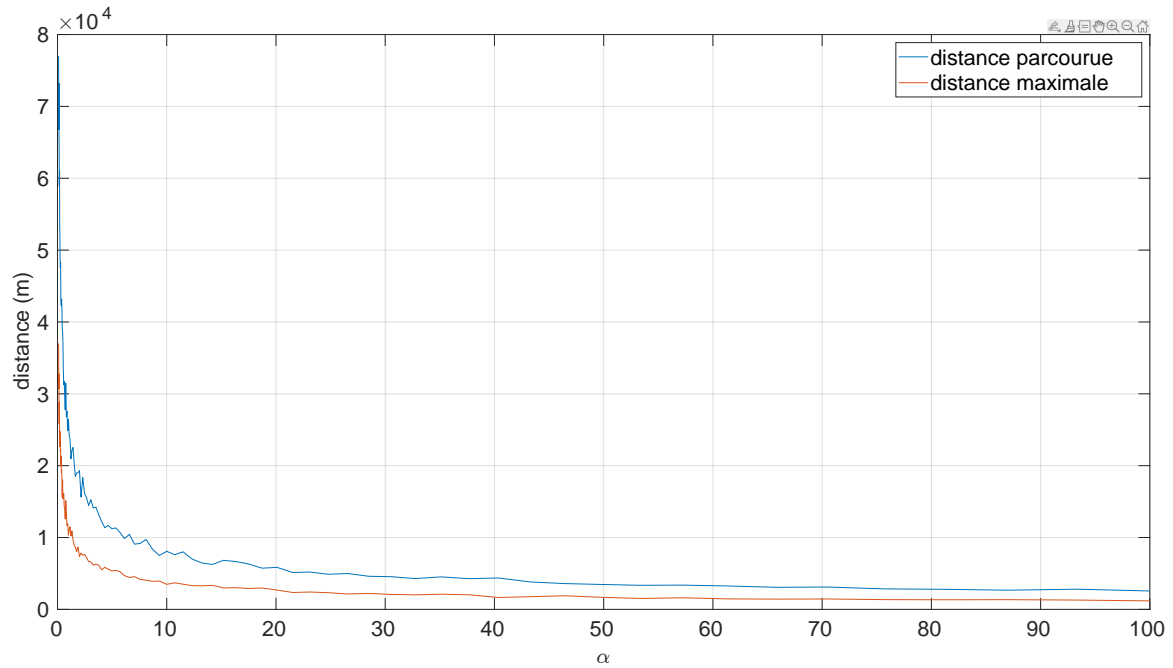
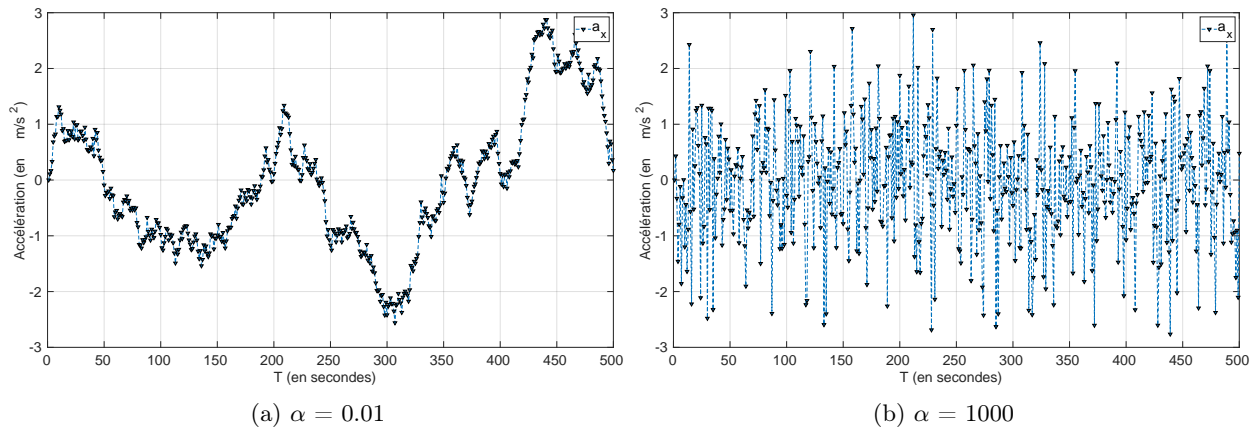
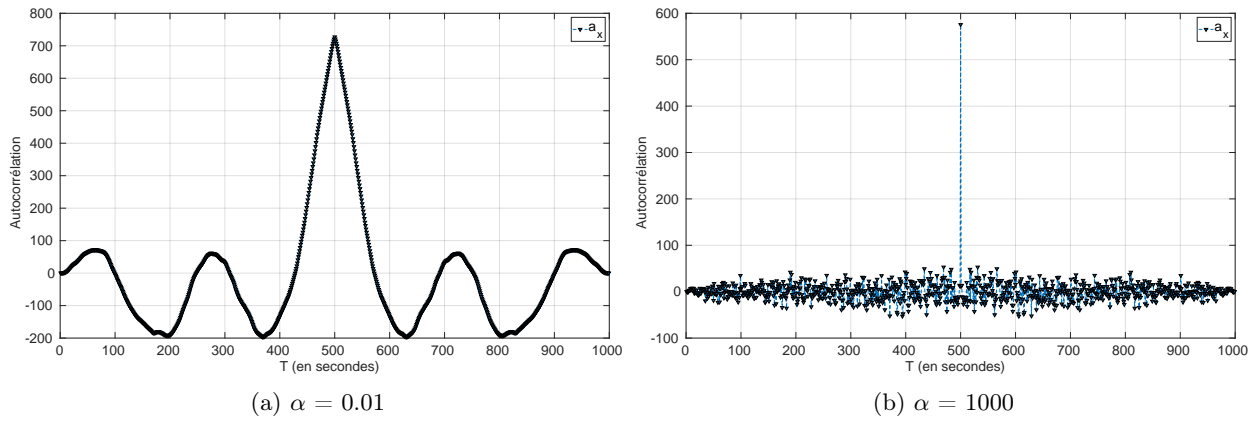


FIGURE 2 – Évolution des distances moyennes parcourues le long de la trajectoire et pour le point le plus éloigné de l'origine, pour des valeurs croissantes de  $\alpha$ .



## Impact sur l'accélération

FIGURE 3 – Évolution de l'accélération du mouvement pour deux valeurs de  $\alpha$ .FIGURE 4 – Evolution de l'autocorrélation de l'accélération pour deux valeurs de  $\alpha$ .

## Trajectoire de singer et mesure bruitée

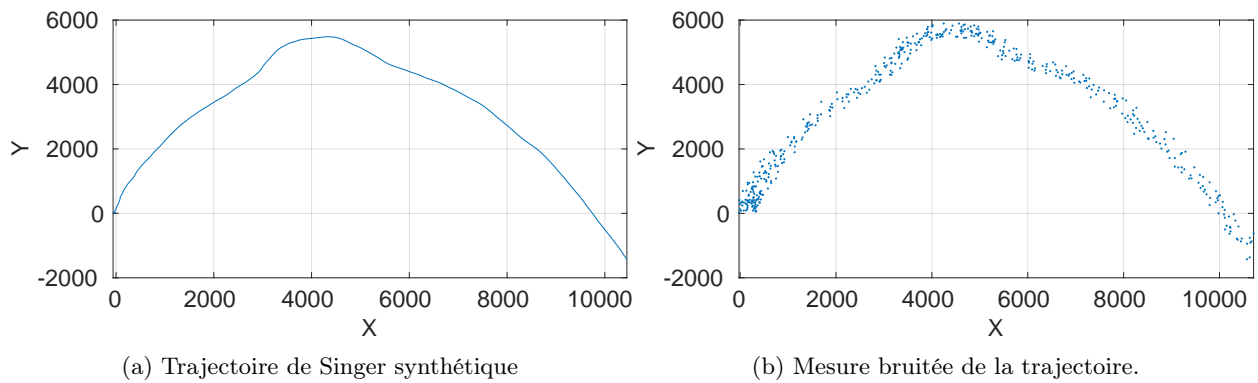


FIGURE 5 – Trajectoire de Singer synthétique et mesures bruitées associées.

## Filtrage et lissage de Singer

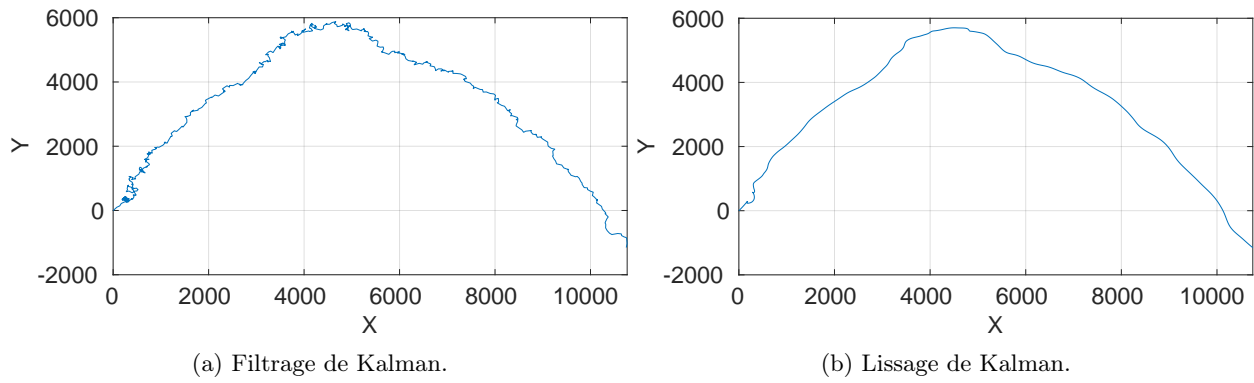


FIGURE 6 – Résultats du filtrage et du lissage de Kalman pour l'estimation de la trajectoire de Singer.

## Sensibilité aux paramètres

### Matrice de covariance du bruit de mesure erronée

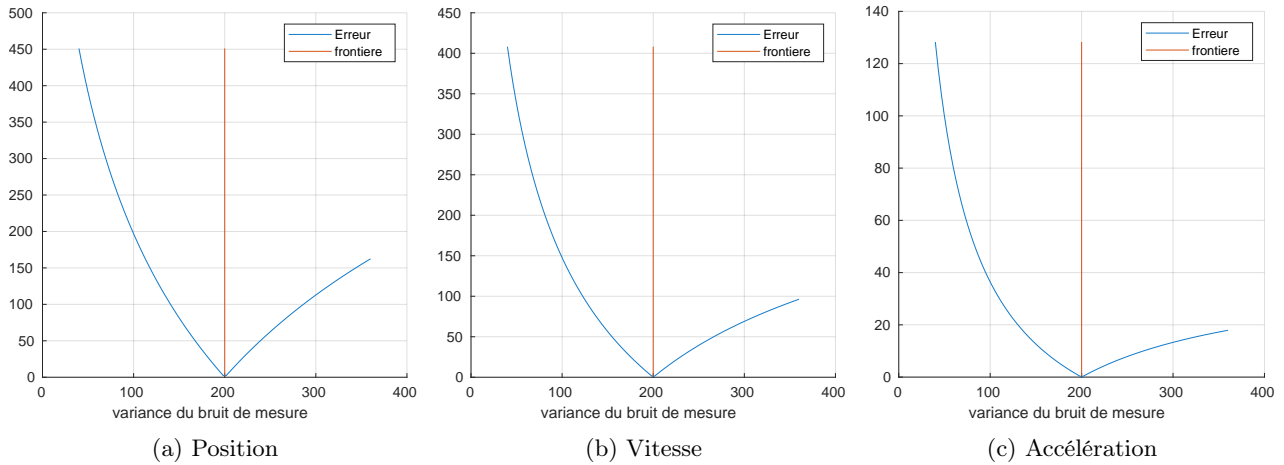


FIGURE 7 – Erreur quadratique par rapport au filtrage de Kalman utilisant la vraie matrice de covariance du bruit de mesure. La ligne orange délimite la sous-estimation de la sur-estimation.

## Matrice de covariance du bruit de modèle erronée

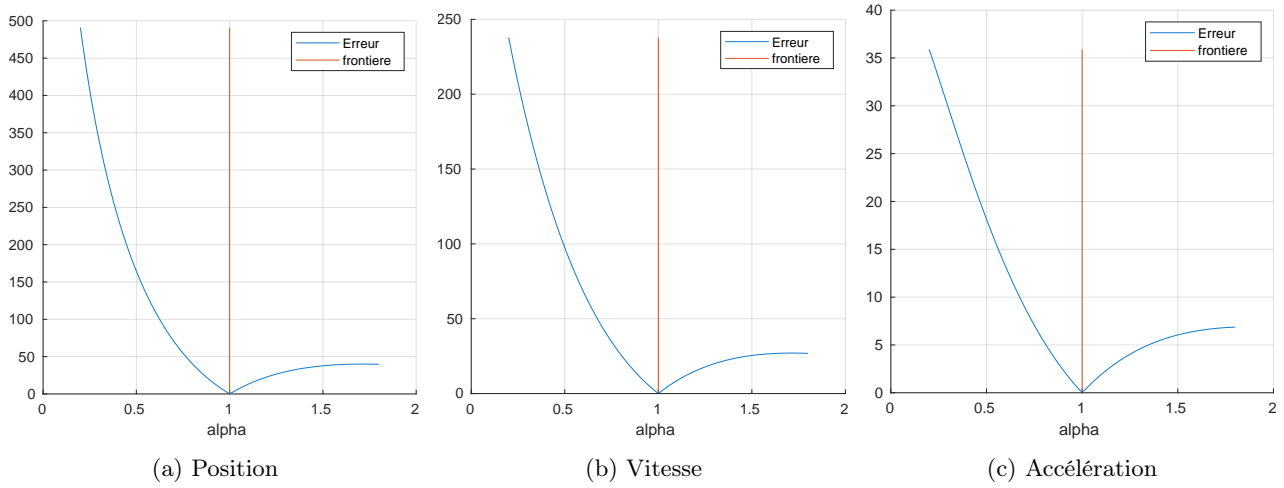


FIGURE 8 – Erreur quadratique par rapport au filtrage de Kalman utilisant la vraie matrice de covariance du bruit de modèle. La ligne orange délimite la sous-estimation de la sur-estimation.

## Matrice de covariance du bruit de modèle erronées

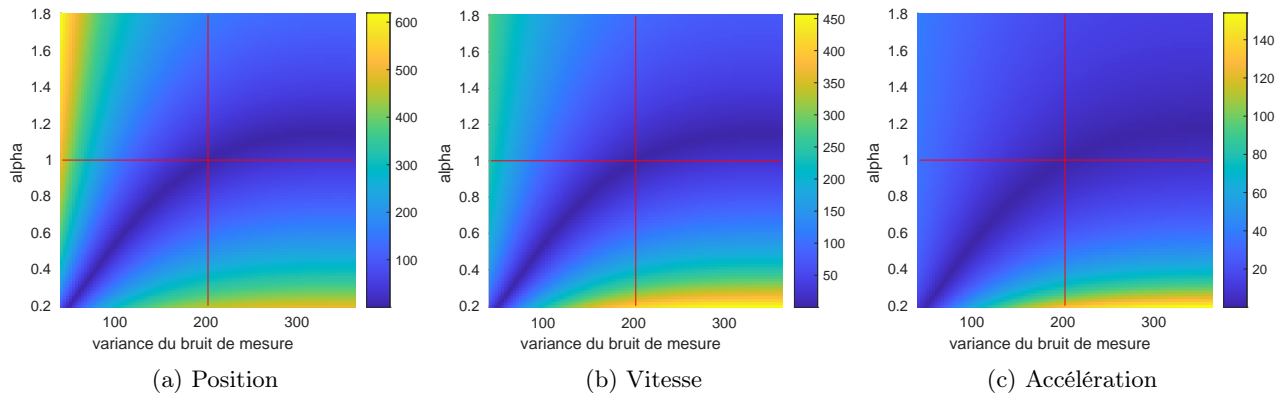


FIGURE 9 – Erreur quadratique par rapport au filtrage de Kalman utilisant les vraies matrices de covariance du bruit de mesure et de modèle. Les segments oranges délimitent les différents régimes (sous et sur estimation des paramètres étudiés)

## B Code

### Simulation d'un mouvement de Singer

```

1 function [X] = sim_singer(N, alpha, T, sigma2, G)
2 %SIM_SINGER simule sur N points l'evolution d'un processus a une dimension
3 %et trois variables d'etat : position, vitesse, acceleration
4 %
5 % Arguments :
6 %   N       : Nombre de points a simuler
7 %   alpha   : vitesse de decroissance de l'autocorrelation de l'acceleration
8 %   T       : Periode d'echantillonnage
9 %   sigma2  : Variance de l'acceleration et du jerk (par default : 0)
10 %   G       : Gain du bruit de modele
11 %
12 % Retour :
13 %   X       : Vecteur d'etat sur 500 points
14
15 nb_var_etat = 3; % position, vitesse et acceleration
16 X = zeros(nb_var_etat, N);
17 phi = get_phi(alpha, T);
18 Q = get_Q(sigma2, alpha, T);
19 A = chol(Q).'; % Q = A*(A^t)
20
21 % mise a jour
22 for i=1:N-1
23     X_next = phi * X(:, i);
24     W       = A * randn(nb_var_etat, 1);
25     X(:, i+1) = X_next + G(i)*W;
26 end

```

Fonction 1 – sim\_singer.m

### Filtrage de Kalman

```

1 function [X_hat, P] = kalman(Y, phi, H, Q, R, G)
2 %KALMAN applique le filtre de Kalman sur le vecteur d'etat observe Y,
3 %connaissant la matrice de transition phi, la matrice de covariance du
4 %bruit de modele, le filtre H, et la matrice de covariance du bruit des observations R.
5     [nb_var_etat, N] = size(Y); % nombre d'echantillons a traiter
6     X_hat = zeros(nb_var_etat, N); % estimation des vecteurs d'etat
7     X_k_apriori = zeros(nb_var_etat, 1); % avant la connaissance de y_k
8     X_k_aposter = zeros(nb_var_etat, 1); % apres la connaissance de y_k
9     K_k = zeros(nb_var_etat, nb_var_etat); % gain de Kalman
10    P_apriori = zeros(nb_var_etat, nb_var_etat, N); % (3x3xN)
11    P_aposter = zeros(nb_var_etat, nb_var_etat, N); % (3x3xN)
12
13    for k = 2:N
14        X_k_apriori = phi * X_k_aposter;
15        P_apriori(:, :, k) = phi * P_aposter(:, :, k-1) * phi.' + G(k)*Q*G(k).';
16        K_k = P_apriori(:, :, k) * H.' / (H * P_apriori(:, :, k) * H.' + R);
17        X_k_aposter = X_k_apriori + K_k * (Y(:, k) - H*X_k_apriori);
18        P_aposter(:, :, k) = (eye(nb_var_etat) - K_k * H) * P_apriori(:, :, k);
19        X_hat(:, k) = X_k_aposter;
20    end
21 end

```

Fonction 2 – kalman.m

## Lissage de Kalman

```

1 function X_smooth = lissage(X_hat, P, phi)
2 %X_SMOOTH realise un lissage de Kalman sur la trajectoire designee par X_hat
3 %
4 % Arguments :
5 % X_hat      : Les N vecteurs d'etat estimes par filtrage de Kalman selon
6 %              une dimension (de taille 3xN)
7 % P          : Les N matrices d'erreur de prediction a priori et a posteriori
8 %              estimees au cours du filtrage de Kalman.
9 %              Structure :
10 %              P.aposter (3,3,N) — estimations P(k|k)
11 %              P.apriori (3,3,N) — estimations P(k|k-1)
12 % phi        : Matrice de passage d'une etape a la suivante du modele de trajectoire
13 %
14 % Retour :
15 % X_smooth : Vecteurs d'etat lisses de taille (3,N)
16     N = length(X_hat);
17     X_smooth = zeros(size(X_hat));
18     X_smooth(:, N) = X_hat(:, N);
19     P_smooth = P.aposter(:, :, N);
20     for k = N-1:-1:1
21         A = P.aposter(:, :, k) * phi.' / P.apriori(:, :, k+1);
22         P_smooth = P.aposter(:, :, k) + A*(P_smooth - P.apriori(:, :, k+1))*A.';
23         X_smooth(:, k) = X_hat(:, k) + A*(X_smooth(:, k+1) - phi*X_hat(:, k));
24     end
25 end

```

Fonction 3 – lissage.m

## Matrice de covariance du bruit de modèle

```

1 function Q = get_Q(sigma, a, T)
2 %GET_Q genere la matrice de covariance du bruit de modele pour un mouvement de Singer
3 %
4 % Arguments :
5 % sigma : Variance de l'acceleration et du jerk
6 % a      : Parametre alpha, vitesse de decroissance de l'autocorr. de
7 % l'acceleration
8 % T      : Periode d'echantillonnage du mouvement
9 %
10 % Retour :
11 % Q      : Matrice de covariance du bruit de modele
12
13     at = a*T;
14     q11 = 1/(2*a^5) * (2*at - 2*at^2 + 2/3*at^3 - 4*at*exp(-at) - exp(-2*at) + 1);
15     q12 = 1/(2*a^4) * (at^2 + 1 + exp(-2*at) + exp(-at)*(-2 + 2*at) - 2*at);
16     q13 = 1/(2*a^3) * (1 - 2*at*exp(-at) - exp(-2*at));
17     q22 = 1/(2*a^3) * (2*at - 3 + 4*exp(-at) - exp(-2*at));
18     q23 = 1/(2*a^2) * (1 - exp(-at))^2;
19     q33 = 1/(2*a^1) * (1 - exp(-2*at));
20
21     Q = 2*a*sigma * [q11 q12 q13;
22                     q12 q22 q23;
23                     q13 q23 q33];
24 end

```

Fonction 4 – get\_Q.m

## Matrice de covariance du bruit de mesure

```
1 function R = get_R(sigma2, nb_var_etat)
2 %GET_R genere la matrice de covariance du bruit de mesure
3   R = sqrt(sigma2)*eye(nb_var_etat);
4 end
```

Fonction 5 – get\_R.m

## Matrice de passage

```
1 function phi = get_phi(alpha, T)
2 %GET_PHI genere la matrice de passage pour un mouvement de Singer
3   at = alpha * T;
4   phi = [1   T (exp(-at) + at - 1)/alpha^2 ; ...
5          0   1 (1 - exp(-at))/alpha       ; ...
6          0   0 exp(-at)                   ];
7 end
```

Fonction 6 – get\_phi.m