

MAT 4506

Filtrage de Kalman : Application à la poursuite de cibles mobiles

Erwan Le Blévec & Pierre Checchin

10 janvier 2022



Table des matières

1	Introduction	3
2	Poursuite en coordonnées cartésiennes	4
2.1	Pertinence du Filtre de Kalman	4
2.2	Application à la poursuite :	4
2.3	Initialisation du problème	5
2.4	Mise en oeuvre du Filtre	6
2.4.1	Implémentation	6
2.4.2	Premiers résultats	6
3	Introduction d'une nouvelle mesure	9
3.1	Analyse des nouveaux résultats	10
3.1.1	Un peu de théorie.. . . .	11
3.1.2	..Et leurs conséquences	12
3.2	Et dans la pratique ?	13
4	Poursuite en coordonnées polaires	15
4.1	Introduction au Filtre de Kalman étendu	15
4.2	Implémentation du nouveau filtre	15
5	Une extension à la poursuite de plusieurs cibles	16
6	Annexe	16
6.1	Codes	16

1 Introduction

Ce rapport fait suite à la réalisation du TP *Filtage de Kalman - MAT 45 01*, il reprend les trois principales parties le composant. Chacune propose une utilisation du filtre dans un contexte particulier mettant en lumière différents fonctionnements de cet algorithme ainsi que les performances qu'il réalise. Toutes ces applications seront consacrées à la poursuite d'une suite mobile.

Dans un premier temps nous étudierons l'application du Filtre de Kalman dans son cas le plus favorable : celui d'un système d'état linéaire avec des bruits de modèle et de mesure gaussiens. Nous nous éloignerons ensuite de ce cadre pour s'intéresser à celui du Filtre de Kalman Étendu, correspondant au cas où au moins l'une des équations (état ou observation) n'est plus linéaire. Enfin nous développerons rapidement sur les problèmes que soulève une implémentation multi-cibles du filtre.

2 Poursuite en coordonnées cartésiennes

2.1 Pertinence du Filtre de Kalman

Tout d'abord, il convient de formaliser le problème que nous allons considérer, celui de poursuite de cible, et d'expliquer en quoi le filtre est adapté à ce dernier.

Durant tout le rapport nous supposons que notre cible admet, à tout instant n , une trajectoire modélisable par une relation : $X_{n+1} = F * X_n + u_n$

X est ici un vecteur de dimension 4, de la forme $[x, \dot{x}, y, \dot{y}]$ et u une variable aléatoire représentant un bruit qui exprime notre confiance en ce modèle. F est une matrice pouvant dépendre de n , il est important de comprendre que u traduit la confiance que nous avons en la modélisation $X_{n+1} = F * X_n$, issue d'une équation que nous présentons plus bas. Dans le contexte de ce TP par exemple nous verrons que le format de F suppose une vitesse constante durant toute la trajectoire, augmenter ce bruit reviendra notamment à relativiser cette considération.

Et que nous avons accès à des mesures de la position à chaque instant n qui peuvent s'exprimer de la manière suivante : $Y_n = H * X_n + v_n$

Y est un vecteur (sa dimension peut varier suivant les mesures réalisées), v le bruit de la mesure et H une matrice (pouvant dépendre de n). Augmenter v reviendra à diminuer la confiance que nous accorderons aux observations, le filtre sera alors moins guidé par ces dernières et cela entraînera une plus grande sensibilité à de fortes variations de positions.

Le problème peut ainsi être formalisé de la manière suivante :

Connaissant $Y_{1..n+1}$ et les prédictions $X_{1..n}$ estimer X_{n+1} sachant $\begin{cases} X_{n+1} = F_{n+1}X_n + U_{n+1} \\ Y_{n+1} = H_{n+1}X_{n+1} + V_{n+1} \end{cases}$

On remarque $(X_n, Y_n)_n$ forme une chaîne de Markov Cachée

Le filtre de Kalman est un algorithme qui a précisément été conçu pour résoudre ce type d'équations. De plus, dans un contexte où X_0 est un vecteur gaussien et (u_i) et (v_i) sont des bruits centrés gaussiens on a immédiatement que tous les (X_n) le sont. L'étude revient alors à des estimations de propagations d'espérances et de matrices de covariance. On parle dans ce cas de *cadre linéaire gaussien* et les solutions du filtre sont explicites.

2.2 Application à la poursuite :

Une expression intégrale possible de la position est la suivante : $x(t) = x(t') + \int_{t'}^t v(\tau) d\tau$.
En discrétisant $\frac{dx(t)}{dt} = v(t)$ on obtient $x_{n+1} \simeq x_n + v * dt$

Avec une démarche similaire pour les ordonnées et en supposant une vitesse constante nous obtenons :

$$X_{n+1} = \begin{pmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{pmatrix} * X_n \quad \text{Rappel : } X = [x, \dot{x}, y, \dot{y}]^T$$

De même, en supposant que les mesures contiennent la "vraie" position plus un bruit d'erreur :

$$Y_{n+1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} * X_{n+1} + v_{n+1} \quad X \text{ et } Y \text{ n'ont pas toujours la même dimension}$$

En ajoutant dans la première équation un bruit représentant la confiance en ce modèle (vitesse constante) nous obtenons les mêmes expressions que celle énoncées dans la partie précédente. En supposant que le vecteur d'état initial est gaussien et que les bruit de modèle et de mesure sont gaussiens nous pouvons appliquer les résultats du Filtre de Kalman ! (ouf)

On remarque également que H et F ne dépendent pas de n , nous sommes dans le cas simplifié de chaînes homogènes

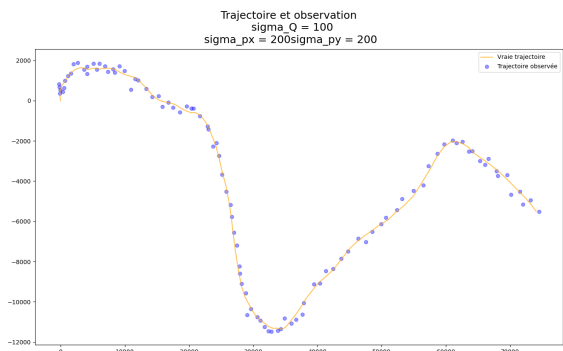
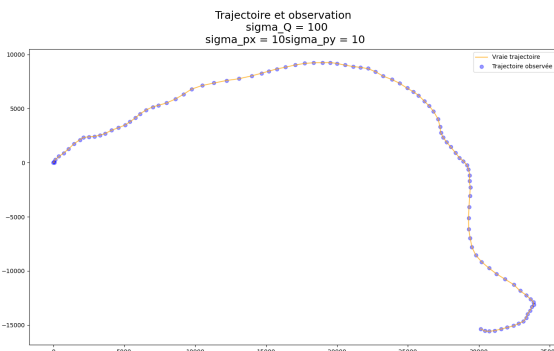
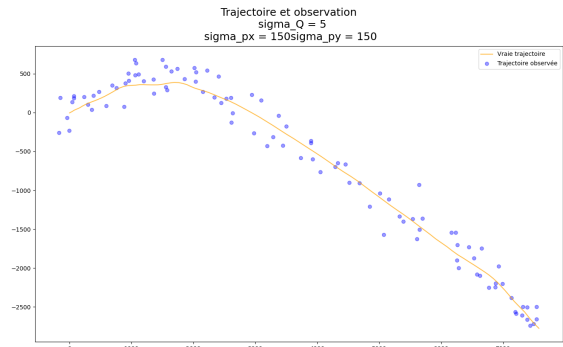
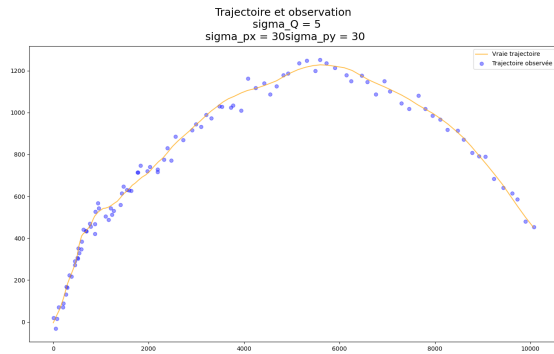
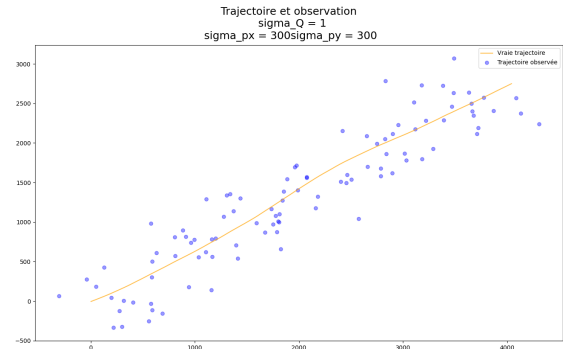
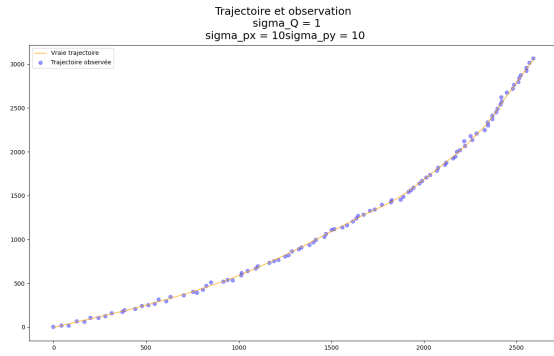
Il pourrait être intéressant de réaliser ce TP en considérant non plus une vitesse mais une accélération constante, et de comparer la finesse des estimations.

2.3 Initialisation du problème

Afin de permettre des analyses et des comparaisons tout au long de ce TP nous générerons à chaque fois une trajectoire "réelle" (celle que nous chercherons à estimer), que nous bruiteons ensuite pour obtenir les observations à filtrer.

La trajectoire réelle sera générée selon l'équation $X_{n+1} = F * X_n + u_n$ où u traduira notre envie de la rendre plus ou moins linéaire (si u augmente la vitesse réelle variera plus fortement et la trajectoire en sera d'autant plus impactée).

Une fois les programmes de création et de bruitage écrits nous obtenons les courbes suivantes :



Lorsque le bruit du modèle est faible (fig (1, 1), (1, 2)) on observe bien une trajectoire relativement proche d'une droite (cohérent vis à vis d'une vitesse constante).

En revanche la trajectoire réelle devient rapidement et fortement non linéaire lorsque σ_Q (u) augmente (2e et 3e ligne). La question sera de savoir comment le filtre réagira face à ces évolutions selon la précision des mesures disponibles.

De manière générale l'importance de la dispersion des observations autour de la vraie courbe corrèle bien avec l'augmentation du bruit de mesure et la complexité de la trajectoire semble bien répondre à une augmentation de σ_Q (forte variation de la vitesse réelle).

Nous pouvons donc estimer que nos programmes fonctionnent. Il est maintenant temps d'implémenter le filtre pour le tester sur ces données.

2.4 Mise en oeuvre du Filtre

2.4.1 Implémentation

Il existe une multitude de manières d'écrire les équations du Filtre de Kalman, j'ai choisi d'utiliser les suivantes :

Prédiction :

$$\bar{X}_{k+1|k} = F * \bar{X}_{k|k}$$

$$P_{k+1|k} = F P_{k|k} F^T + Q$$

Filtrage :

$$K_{k+1} = P_{k+1|k} H^T (H P_{k+1|k} H^T + R)^{-1}$$

$$\bar{X}_{k+1|k+1} = \bar{X}_{k+1|k} + K_{k+1} (Y_{k+1} - H \bar{X}_{k+1|k})$$

$$P_{k+1|k+1} = (I - K_{k+1} H) P_{k+1|k}$$

Avec les conditions initiales : $\begin{cases} \text{Accélération : } a_0 = \mathbf{0} \\ \text{Vitesse : } v_x, v_y = [\mathbf{40}, \mathbf{20}] \\ \text{Position : } [x_0, y_0] = [\mathbf{3}, \mathbf{-4}] \end{cases}$ et des bruits gaussiens centrés de

matrices de covariance notées Q pour le modèle et R pour les mesures (voir les programmes pour les expressions).

Pour l'initialisation de X_o : on choisira une matrice de covariance $P_{o|o}$ "faible" si nous sommes assez certain de la position au départ et inversement $P_{o|o}$ sera d'autant plus "importante" que la position initiale est peu connue.

Il est temps de tester le filtre !

2.4.2 Premiers résultats

Une fois le filtre implémenté nous obtenons les courbes présentées ci-dessous. Nous avons considéré 3 cas possibles pour chaque bruit (avec p_x et p_y regroupés), ce qui nous fait un total de 9 figures.

Le bruit sur la confiance du modèle varie de 1 (grande confiance) à 10 (confiance très faible) avec pour chacune des erreurs d'observations allant de 10 (très faibles) à 300 (très fortes). De cette manière nous pourrions bien observer l'impact séparé et simultané de leurs variations.

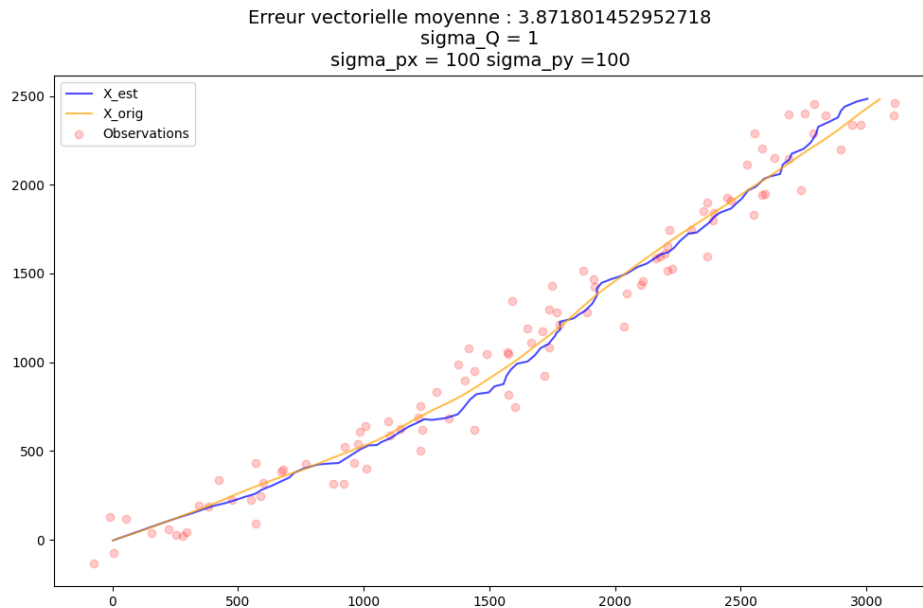


FIGURE 1 – Erreur de modèle faible et de mesure moyenne.

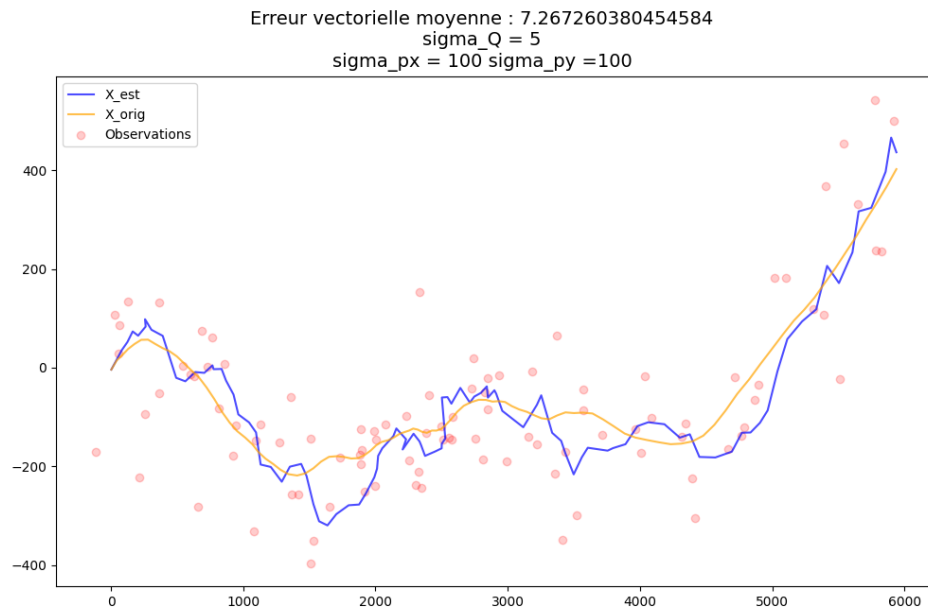


FIGURE 2 – Erreur de modèle moyenne et de mesure moyenne.

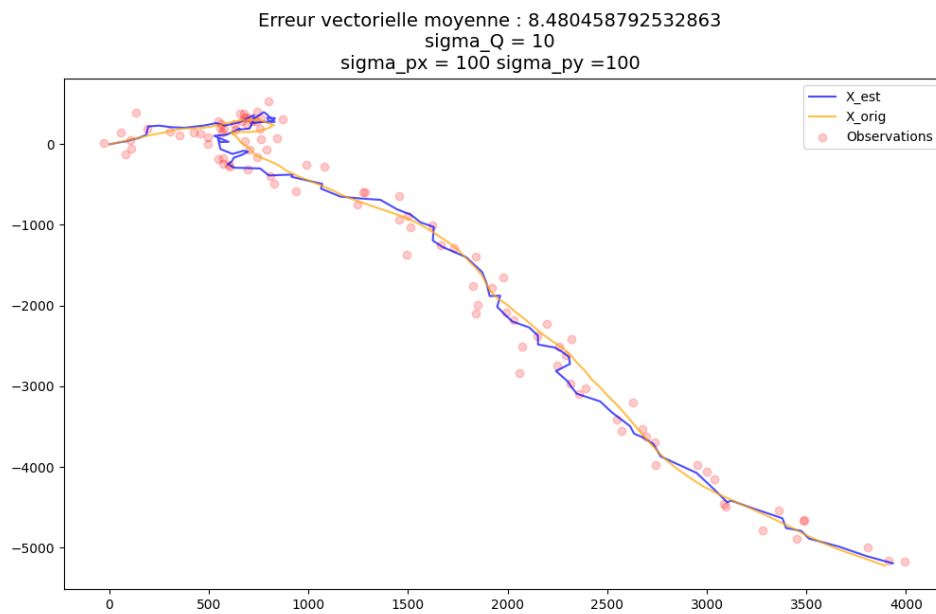
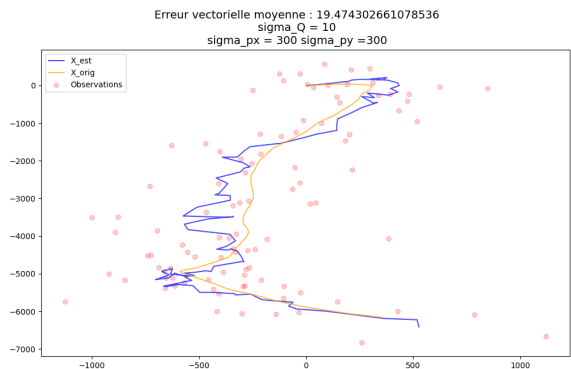
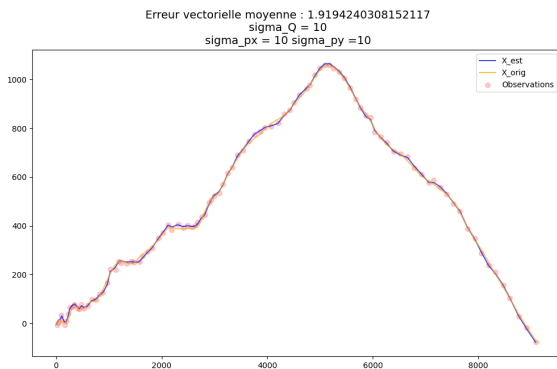
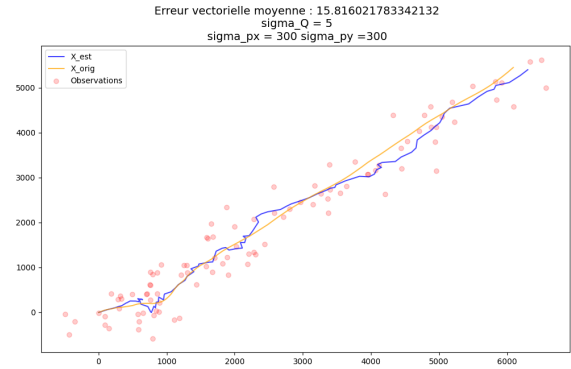
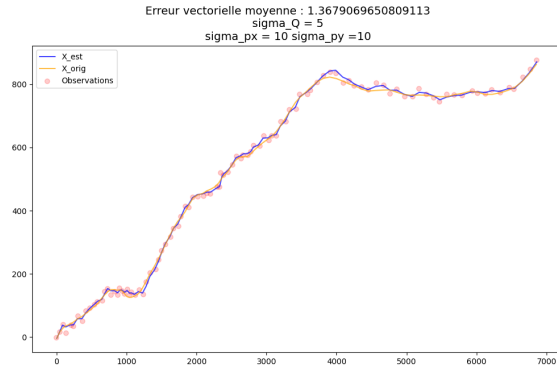
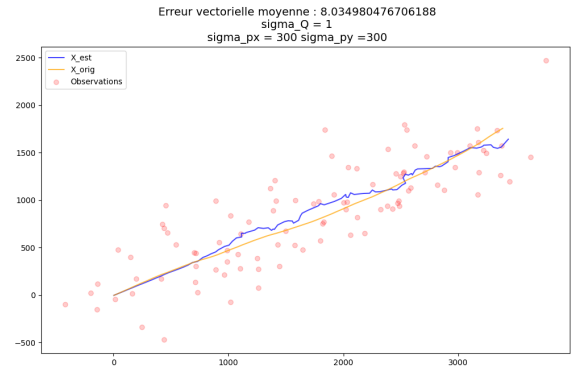
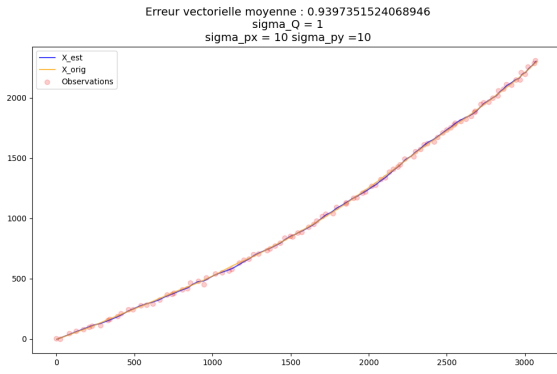


FIGURE 3 – Erreurs de forte et de mesure moyenne.

Et pour les cas restants :



Remarque : Pour calculer "l'erreur" l'énoncé du TP proposait la formule suivante :

$err_quadra = (x_n - \bar{x}_{n|n})^T * (x_n - \bar{x}_{n|n})$ cette formule peut être critiquée d'un point de vu homogénéité : nous additionnons des erreurs sur des vitesses et des erreurs sur des positions. C'est pourquoi il ne faut pas considérer cette erreur comme relevant d'une réalité physique mais d'une purement mathématique : celle de la distance entre deux vecteurs. Ceci explique son nom "erreur vectorielle". Nous ne pourrions donc pas la comparer aux grandeurs atteintes lors du mouvement, qui elles ont une réalité physique.

Il semblerait que pour des erreurs de mesures (p_x, p_y) faibles le filtre reste capable de suivre le mouvement en dépit de variations importantes de σ_Q (la vérification de cette hypothèse est l'objet de la partie suivante).

Cependant nous observons d'ores et déjà les limites du Filtre de Kalman : si dans les cas de grande confiance en le modèle (fig (1,1), 2) l'approximation est bonne malgré des bruits de mesure qui peuvent être conséquents, il ne l'est absolument plus dès que les deux bruits augmentent simultanément. En effet lorsqu'une information est mauvaise le filtre a tendance à s'appuyer sur l'autre pour prédire, mais si les deux sont imprécises les erreurs se cumulent et l'algorithme devient rapidement peu performant (fig (3, 2)).

Par ailleurs nous pourrions être amené à penser que dans un cas très proche du linéaire l'équation d'état à elle seule suffirait pour estimer X. Ce relativisme de l'efficacité du filtre est démenti par les résultats suivants où nous voyons très clairement que même dans un cadre extrêmement proche du linéaire les résultats sont très inférieurs à ceux obtenus par le filtre.

Les figures ci-dessous ont été obtenues à partir de $X_{n+1} = F * X_n + u_n$ uniquement, ce qui correspond au cas où les observations ne sont pas prises en compte. Il faut diviser par 10 les erreurs vectorielles affichées avant de les comparer aux précédentes (petite erreur dans le programme d’affichage).

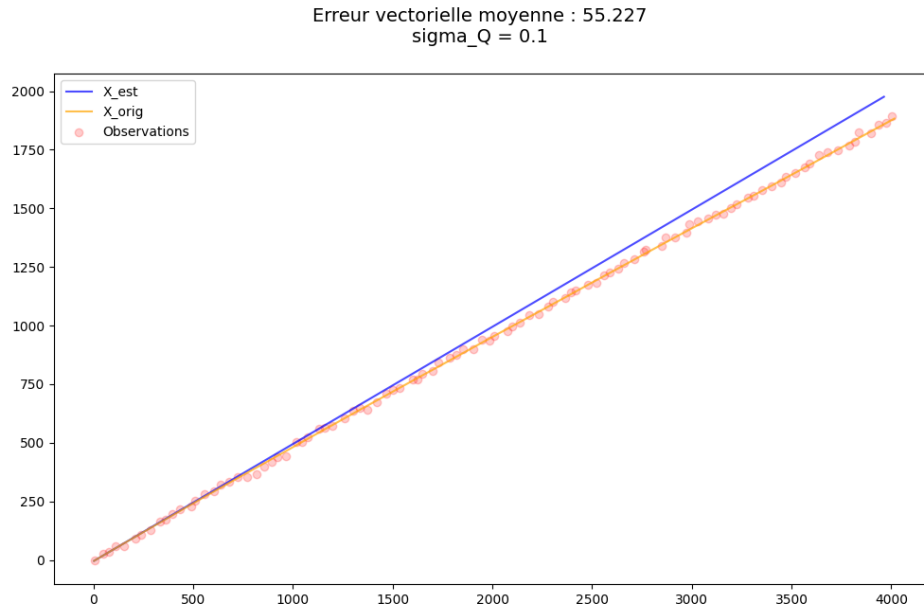
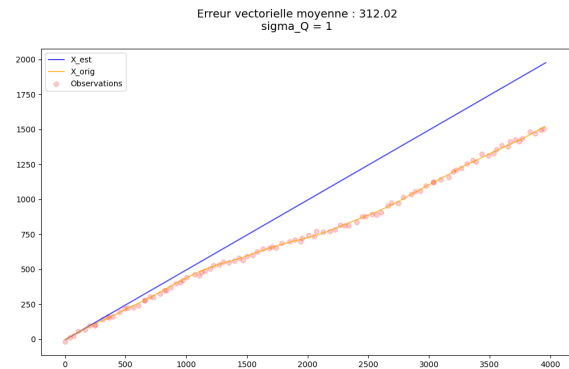
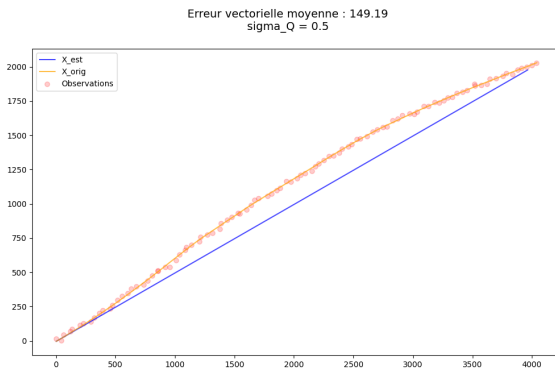


FIGURE 4 – Erreur vectorielle pour des estimations sans filtrage.



Nous remarquons que les erreurs sont supérieures d’un facteur 2 voire plus à celles obtenues par prédiction et filtrage, cela pour une configuration des bruits bien plus avenante. Le filtre Kalman apporte donc une précision supplémentaire très appréciable même dans des cas de bruitages très faibles.

3 Introduction d’une nouvelle mesure

Malgré une première remarque sur l’utilisation de l’erreur vectorielle je pense qu’il peut être intéressant de détailler davantage les conséquences de son utilisation. Le problème principal est, comme il a déjà été énoncé, le manque d’homogénéité qu’elle contient puisque cela rend toute interprétation particulièrement complexe.

Par exemple lorsque nous considérons un σ_Q égal à 100 et des bruits de mesure égaux à 1 le modèle se met à ne considérer presque que les observations et est ainsi tout à fait capable de suivre la trajectoire, comme le montre la figure 5. Or nous remarquons que l’erreur vectorielle obtenue est conséquente (du même ordre de grandeur que celle pour un bruit de modèle faible et de mesure fort, à nouveau il faut diviser par 10 l’erreur affichée).

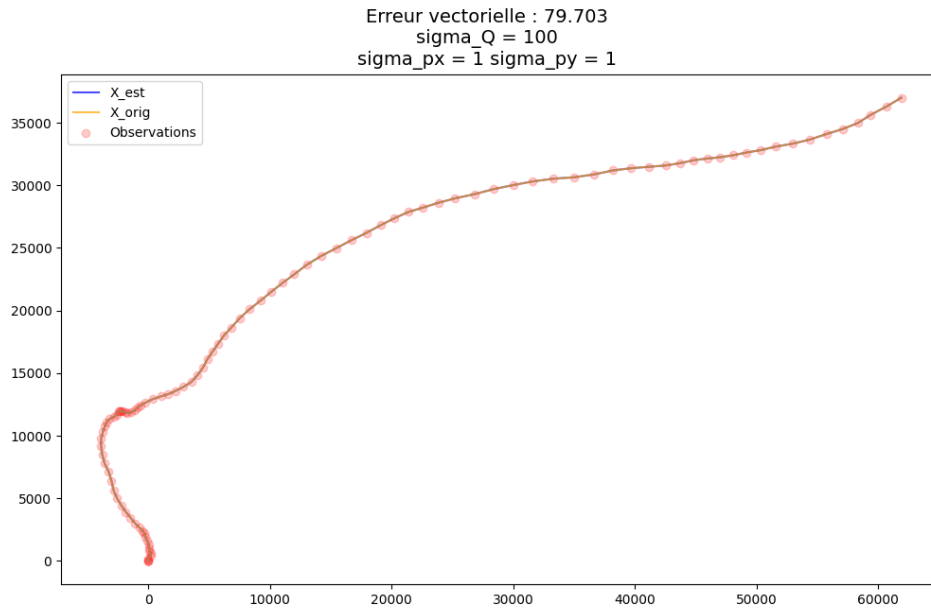


FIGURE 5 – Courbe de suivi pour un bruit de modèle fort de mesure très faible.

Qu'en déduire ? Est-ce l'échelle qui nous trompe ? Les sources des erreurs sont-elles les mêmes que dans le cas précédent ? Est-ce davantage une erreur sur la vitesse ou sur la position ? Toutes ces questions ne peuvent avoir de réponse avec la seule erreur vectorielle.

Dans notre cas l'erreur sur la position est très faible, c'est en réalité l'erreur de vitesse qui fait accroître l'erreur vectorielle, des explications seront présentées dans les parties suivantes.

Pour palier ce problème de dimensionnalité tout en conservant la forme initiale nous pourrions penser à transformer les erreurs en taux avant de les sommer, les grandeurs seraient alors sans dimension et la "physique" respectée.

Il s'agit cependant d'une fausse bonne idée pour la raison suivante :

Si l'on suppose que notre position exacte est $[20, 14]$ et que notre position estimée par le filtre est $[19, 14]$ notre taux d'erreur pour cet instant vaut $1/20 * 100$ soit 5%. En considérant que les grandeurs sont en mètre une erreur d'1 mètre correspondrait à un taux d'erreur de 5%. Si par contre notre position exacte est $[80\,000, 14]$ et celle estimée $[79\,700, 14]$ nous nous sommes trompés de 300 mètres mais notre taux d'erreur n'est que (d'environ) 0.4% !

Dans le cas où la cible est un catamaran de 5' par 3' il est clair que la première erreur est acceptable là où la seconde ne l'est absolument pas, pourtant notre algorithme nous dit le contraire.

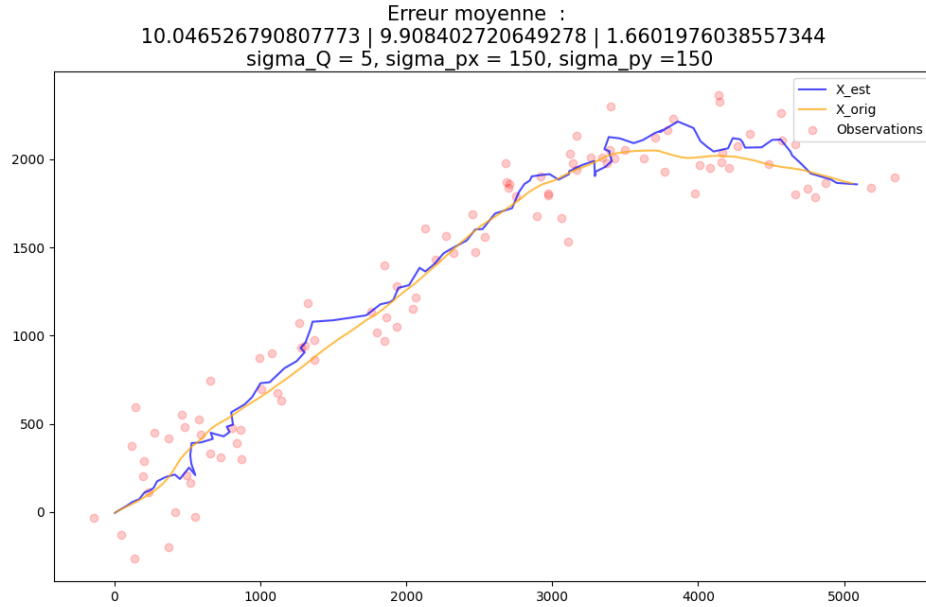
Par ailleurs utiliser une telle mesure reviendrait à considérer que la qualité du filtre dépend de la position où nous plaçons l'origine, ce qui est absurde.

Il n'existe donc pas de mesure de la précision conciliant vitesse et position tout en conservant une interprétabilité physique.

Pour pouvoir interpréter les résultats j'ai choisi de séparer les calculs de l'erreur vectorielle, une première erreur concernera les positions (moyenne de l'erreur sur x et sur y) et une autre les vitesses. Ce choix de mesures apporte davantage d'informations et surtout permet l'interprétabilité, qui est l'objet de la partie suivante.

3.1 Analyse des nouveaux résultats

Pour des raisons de comparabilité nous avons repris des conditions similaires à celles de bruitage de la section précédente. Nous avons juste utilisé un bruit d'observation de 150 au lieu de 100 et supprimé celui de 300 (une erreur de mesure si importante n'a pas vraiment de pertinence, les faiblesses de l'algorithme face à des bruits de 150 sont déjà suffisamment révélatrices).

FIGURE 6 – Prédictions pour $\sigma_Q = 5$ et $\sigma_x = 150$.

Bruits : Q, px, py	Erreur vectorielle 1	Erreur de position 2	Erreur de vitesse 3
1, 10, 10	0.91	0.87	0.26
1, 150, 150	8.21	8.19	0.62
5, 10, 10	1.37	1.03	0.90
5, 150, 150	10.45	9.91	1.66
10, 10, 10	1.81	1.14	1.41
10, 150, 150	10.96	10.56	2.94

TABLE 1 – Erreurs moyennes (sur 100 itérations) du filtre

Nous obtenons les observations suivantes (certaines étaient déjà déductibles avec l'erreur vectorielle, mais pour des raisons d'organisation nous les faisons apparaître ici seulement) :

1. K_n est toujours une matrice de la forme $\begin{pmatrix} a_n & 0 \\ b_n & 0 \\ 0 & a_n \\ 0 & b_n \end{pmatrix}$. Ses termes, (a_n, b_n) semblent évoluer

entre $10e-7$ et 3 et convergent toujours au bout d'une petite dizaine d'itérations. (La suite des $(K_n)_n$ serait donc stationnaire).

2. Les limites des termes de K ($(a_n)_n$ et $(b_n)_n$ donc) ne semblent pas dépendre des conditions initiales (X_0) .
3. σ_Q semble avoir davantage d'impact sur l'erreur de vitesse que sur celle de position (Voir Tableau 1 lignes 1 et 5)

La relation inverse est observée : une augmentation du bruit d'observation semble entraîner une bien plus forte accentuation du côté de l'erreur de position que du côté de celle de vitesse. (Une augmentation d'un facteur 10 contre une d'un facteur 4, ligne 1 et 2)

De manière générale nous remarquons que si notre confiance en la modélisation s'érode les erreurs augmentent mais faiblement. Si par contre le bruit de mesure devient important, les erreurs explosent. Cela traduit une certaine souplesse du filtre de Kalman vis à vis de la qualité de la modélisation mais une grande dépendance en la fiabilité des observations.

3.1.1 Un peu de théorie..

Avant de se lancer dans l'interprétation et/ou la justification de ces résultats, il convient de retourner quelque peu à la théorie du filtre.

Si l'on devait résumer le coeur de son action nous pourrions présenter les choses ainsi :

Supposons que nous sommes à l'instant k et que nous venons de recevoir la mesure y_{k+1} , pour en déduire la position "exacte" le filtre effectue les étapes suivantes :

- **Prédiction** : (nous sommes dans le cas gaussien donc nous ne considérons que les covariances et les espérances) $\bar{x}_{k+1|k} = F * \bar{x}_{k|k}$ (1)
- L'équation d'observation nous donne la mesure que nous sommes supposés attendre : $\bar{y}_{k+1} = H * \bar{x}_{k+1|k}$ (2)
- La différence avec la mesure obtenue nous donne l'erreur de prédiction : $z_{k+1} = y_{k+1} - H * \bar{x}_{k+1|k}$ (3)
- **Filtrage** : nous choisissons de corriger cette erreur linéairement (principe de ce filtre), c'est à dire que nous ajoutons un terme fonction de cette erreur à notre première prédiction : $\bar{x}_{k+1|k+1} = \bar{x}_{k+1|k} + K_{k+1} * z_{k+1}$ (4)
Où K_{k+1} est choisi de sorte à minimiser : $\mathbb{E}[(x_{k+1} - \bar{x}_{k+1|k+1})^T (x_{k+1} - \bar{x}_{k+1|k+1})]$
 K traduit en quelque sorte notre confiance en l'observation

Ce sont ces 4 étapes qui faonnent les valeurs que nous obtenons dans nos résultats. En remplaçant les équations (1), (2) et (3) dans (4) nous obtenons les expression suivantes (par "symétrie" de K je n'exprime que la position suivant x , notée $x_{k|k}^0$ et la vitesse de cette coordonnée, v_k^x). (**T = 1 ici**)

$$x_{k+1|k+1}^0 = a_{k+1}x_{k+1}^0 + (1 - a_{k+1})(x_{k|k}^0 + Tv_k^x) \quad (5)$$

$$v_{k+1}^x = (1 - b_{k+1}T)v_k^x + b_{k+1}(x_{k+1}^0 - x_{k|k}^0) \quad (6)$$

Les termes de K traduisants la confiance que nous avons en l'observation, ils seront d'autant plus faibles (proches de 0) que le bruit sera grand. Ceci est bien représenté dans nos équations : si a et b tendent vers 0 l'observation x_{k+1} n'est plus du tout prise en compte. À l'inverse, si elles sont proches de 1 le modèle fera bien plus confiance aux observations qu'à l'état précédent (termes en $1-a$, $1-b$).

3.1.2 ..Et leurs conséquences

Nous remarquons immédiatement que dans le cas où les prédictions sont parfaites à chaque itérations ($z_{k+1} = 0$) les vitesses sont constantes (4), et puisque nous sommes dans un cas de prédiction parfait, cela implique que *dans la réalité* les vitesses doivent être constantes.

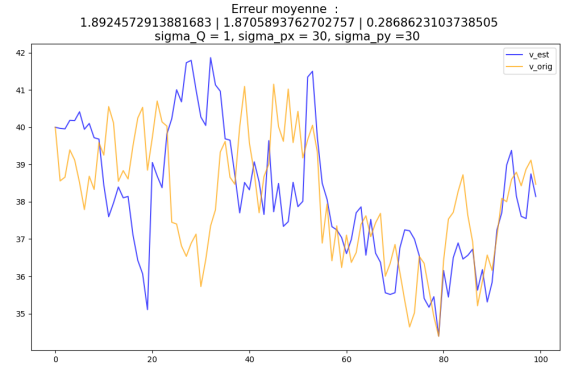
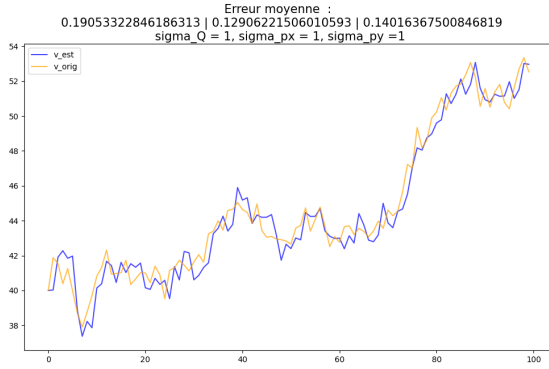
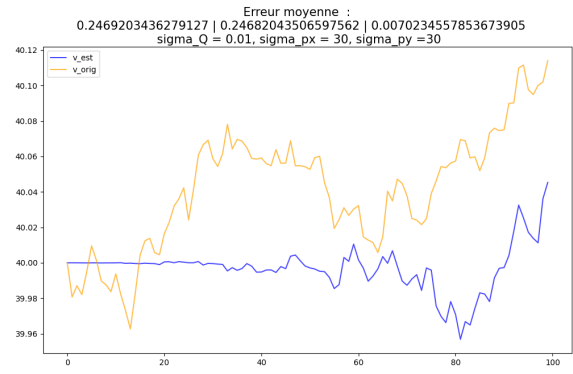
Une condition nécessaire à la nullité de l'erreur sur les vitesses est donc qu'elles soient constantes, nous retrouvons bien les résultats énoncés partie I.

Pour (6) : $v_{k+1}^x = (1 - b_{k+1}T)v_k^x + b_{k+1}(x_{k+1}^0 - x_{k|k}^0)$

La première chose qui ressort de cette équation est le fait que, à l'instar des positions, nous n'avons pas accès à la dérivée pour estimer la valeurs de la nouvelle vitesse. L'ampleur des variations de cette dernière est seulement déductible de la différence entre les positions *mesurée* et *prédite* (second terme de (6)). Ainsi si l'une ou l'autre se révèle fortement biaisée (fort bruitage des mesures, mauvaises prédiction précédente) l'erreur de l'algorithme va rapidement augmenter. De la même manière, si les variations de vitesses sont importantes ($\sigma_Q > 5$) l'algorithme ne pourra que se baser sur l'observation pour les suivre, et dans le cas où le bruit de mesure est important il risque fort d'en être incapable (ce qui a été observé dans le Tableau 1, dernière ligne)

Les figure suivantes montrent bien l'importance qu'a σ_Q sur l'évolution des vitesses - notamment la fameuse "constance" supposée par le modèle - et la difficulté immédiate que présente l'algorithme à les suivre dès que les observations sont moins précises.

Remarque : Attention à l'échelle !



Nous remarquons qu'à σ_Q constant l'erreur sur les vitesses baisse lorsque le bruit d'observation diminue, cela provient du fait que dans ce cas les vitesses peuvent être retrouvée directement et de manière précise à partir des positions (second terme de l'expression, pour $b \simeq 1$, avec $T=1$). Ces figures expliquent également l'origine des erreurs dans le cas de bruits importants (fig 4) : la vitesse est particulièrement mal suivie, ce qui propage des erreurs à chaque itération.

Remarque :

Il pourrait être intéressant de comparer les erreurs en utilisant non plus $x_{k|k}^0$ mais $x_{k+1|k+1}^0$, censée être plus proche de la position exacte, pour estimer v_{k+1}

Pour (5) :
$$x_{k+1|k+1}^0 = a_{k+1}x_{k+1}^0 + (1 - a_{k+1})(x_{k|k}^0 + T v_k^x)$$

Comme énoncé précédemment la pauvreté d'information que subit l'estimation des vitesses n'est pas directement applicable à l'explication des erreurs sur la position. En théorie les prédictions de ces dernières sont capables de suivre de grandes variations puisque la vitesse est présente dans l'équation itérative (si les variations sont grandes alors nécessairement la vitesse est importante).

Pour rappel l'équation à l'origine de l'expression de x est la suivante : $x(t) = x(t') + \int_{t'}^t v(\tau) d\tau$, forme que l'on retrouve bien, accompagné d'un terme supplémentaire (x_{k+1}), dans (5).

Pour les positions les origines des erreurs sont justement ces différences que l'on observe entre les deux expressions. Tout d'abord dans l'algorithme $x(t')$ est remplacé par $x_{k|k}^0$, qui est une estimation et non la valeur exacte de l'ancienne position. Ensuite, l'expression (5) indique que la vitesse est supposée constante entre deux itérations (terme en $T \cdot v$), ce qui n'est pas le cas en réalité. Enfin, certes la présence de la vitesse dans l'équation permet de suivre de grandes variations des positions, mais encore faudrait il que cette vitesse soit proche de la bonne ! Or comme nous l'avons vu l'estimation de la vitesse souffre de quelques lacunes..

Ce sont ces trois principaux points qui je pense sont à l'origine des erreurs sur la position. On remarque d'ailleurs le rôle de balance entre prédiction et mesure que joue K . Si les observations sont bonnes il leur attribuera plus d'importance qu'aux prédictions, et inversement.

3.2 Et dans la pratique ?

L'ensemble des modélisations jusqu'à présent se déroulaient dans un cadre parfait : les observations arrivaient une par une et ne manquaient jamais, permettant un filtrage optimal à chaque

instant. Mais que faire si celles ci viennent à manquer pendant un certain intervalle de temps ? Par exemple si, dans le cas de poursuite d'un avion, un nuage le cache ?

Une variante pour que l'algorithme puisse faire face à ce type de situation est, en cas de manque, de considérer que les prédictions en cas d'absence des observations.

Cette implémentation nous donne les résultats suivants :

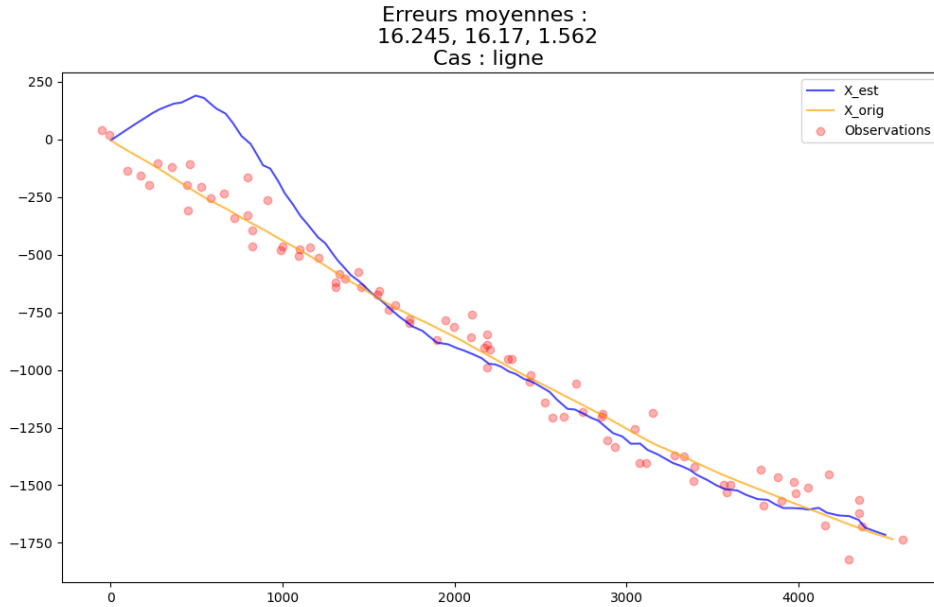


FIGURE 7 – Prédiction avion de ligne.

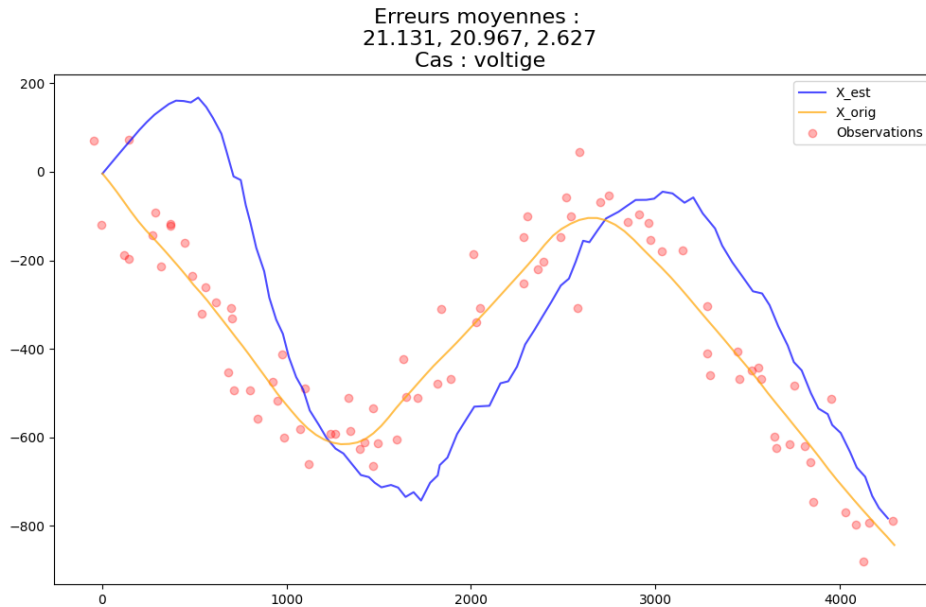


FIGURE 8 – Prédiction avions de voltige.

L'importance supérieure de l'erreur sur la position et la vitesse peut être expliquée avec les expressions obtenues dans la partie précédente : pour une étape où l'observation vient à manquer le filtrage n'a pas lieu, la vitesse est ainsi considérée constante et notre position simplement prédite à partir de l'équation. d'état. Or il est fort probable que 1. la vitesse a changé, ce qui entraîne l'apparition d'une première erreur 2. la prédiction soit mauvaise, (même dans un cadre quasi parfait - $\sigma_Q \simeq 0.01$, $p_{x,y} \simeq 1$ - les résultats de la prédictions pures sont mauvais voir 2.4.2). Le filtre

dépendant fortement des prédictions passées dans ce contexte, ces erreurs se répercuteront sur les suivantes, réduisant leur qualité.

Dans une situation comme celle-ci il faut donc espérer que les variations de la vitesse ne seront pas trop importantes, le cas échéant l'algorithme sera incapable de poursuivre la cible si l'intervalle de temps d'absence est long.

Il pourrait être intéressant de regarder l'impact de ce "retard" de l'observation sur la rapidité de convergence de K .

4 Poursuite en coordonnées polaires

4.1 Introduction au Filtre de Kalman étendu

Certains phénomènes de suivi n'admettent pas une modélisation linéaire de la forme $X_{n+1} = F_{n+1}X_n + u_{n+1}$ et $Y_{n+1} = HX_{n+1} + v_{n+1}$ mais une davantage complexe qui peut être exprimée sous la forme $X_{n+1} = f(X_n)$ et $Y_{n+1} = h(X_{n+1})$ où f est une fonction de \mathbb{R}^d dans \mathbb{R}^d et h une de \mathbb{R}^d dans \mathbb{R}^m .

Or si ces fonctions présentent des non-linéarités les résultats explicites du filtre (exprimés partie 2.4.1) ne sont plus valables. Pour palier ce problème une solution est de changer d'algorithme, d'en choisir un plus contraignant en terme de puissance de calcul mais capable de suivre ce type de phénomène. Nous pourrions par exemple nommer celui du filtrage particulaire.

Une autre est d'essayer de se ramener à une situation linéaire où nous pourrions à nouveau utiliser les résultats du filtre. C'est l'approche que propose de réaliser cette partie.

4.2 Implémentation du nouveau filtre

Pour toute fonction $f \in C^1$ de \mathbb{R}^d dans \mathbb{R}^m on peut écrire : $f(x) \simeq f(x_0) + J(f)(x_0)(x - x_0)$ où J est la jacobienne de f .

Nous retrouvons une expression linéaire dans nos équations ! Les résultats du filtre sont à nouveau utilisables, il ne reste plus qu'à calculer les gradients pour obtenir J .

Dans le cadre de ce TP nous considérerons que seule l'équation sur Y admet une non-linéarité, cela provenant du référentiel de coordonnées de l'observation disponible, qui devient polaire et non plus cartésien.

Nous avons alors : $\forall n \in \mathbb{N}, Y_n = [r_n, \theta_n] = f(X_n) = [\sqrt{p_n^x^2 + p_n^y^2}, \arctan(\frac{p_n^y}{p_n^x})]$

Remarque :

- L'expression de θ est simplifiée, elle n'est valide que dans le cas où $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, cette condition doit être surveillée (il y a peu de chance que cela soit vérifié à tout instant dans un cas réel).
- En réalité il est possible de contourner ce problème sans passer par l'expression au premier ordre de f mais simplement en considérant que $X = [r, \dot{r}, \theta, \dot{\theta}]$ (nous gardons en bonus les mêmes expressions de H et F que celles initiales !) puis en transformant toutes les prédictions polaires en cartésiennes.

Nous obtenons :

$$\bar{H}_k = \begin{pmatrix} \frac{p_n^x}{\sqrt{p_n^x^2 + p_n^y^2}} & 0 & \frac{p_n^y}{\sqrt{p_n^x^2 + p_n^y^2}} & 0 \\ -\frac{p_n^y}{p_n^x^2 + p_n^y^2} & 0 & \frac{p_n^x}{p_n^x^2 + p_n^y^2} & 0 \end{pmatrix}$$

Que nous ré-implémentons dans l'équations (il faut veiller à bien gérer les constantes qui apparaissent). Les résultats sont les suivants :

Bruits : Q, px, py	Erreur vectorielle 1	Erreur de position 2	Erreur de vitesse 3
1, 10, 10	1.36	1.33	0.28
1, 50, 50	15.76	15.75	0.69
1, 120, 120	18.40	18.40	0.52
5, 10, 10	5.67	5.51	1.30
5, 50, 50	17.01	16.88	2.11
5, 120, 120	<i>93.48</i>	<i>93.44</i>	2.88
10, 10, 10	15.62	15.32	3.04
10, 50, 50	<i>21.21</i>	<i>21.01</i>	2.91
10, 120, 120	<i>365.17</i>	<i>364.98</i>	<i>9.57</i>

TABLE 2 – Erreurs moyennes (sur 100 itérations) du filtre, une valeur en italique indique une forte variance

Il apparaît que la variance des erreurs est énorme dans certaines configurations (de l'ordre d'un facteur 5 à 10) dans ces cas les valeurs des erreurs ne sont pas représentatives.

Ce tableau met en lumière le problème majeur du filtre de Kalman étendu : dès que l'erreur de prédiction devient un tout petit peu trop importante elle est fortement accentuée par l'approximation linéaire du premier ordre et se propage alors à tous les termes suivants du fait de la forme intrinsèque du filtre de Kalman.

L'utilisation d'une telle approche n'est donc viable qu'avec des bruit de mesures très faibles, ce qui requiera une grande attention dans le choix des outils de mesure.

5 Une extension à la poursuite de plusieurs cibles

Un des grands défis de la généralisation multi-tâches du filtre à de la poursuite multi-cibles est l'identification.

Même si nous nous trouvons dans un cadre fortement adapté au filtre (linéaire gaussien, bruits de modèle et de mesure faibles) l'algorithme n'assurera de bons résultats que tant que des observations seront disponibles (voir partie 3.2). Si, par exemple, plusieurs cibles se retrouvent cachées au même instant par un même obstacle, comment les identifier une fois de nouveau visible ? Car comme nous l'avons montré dans ce rapport, le filtre se base uniquement sur ses prédictions pour estimer les grandeurs dans ce contexte, ce qui le rend particulièrement sensible aux fortes variations de vitesses qui sont considérées constantes. Si l'une des cibles change sa vitesse de manière conséquente (notamment sa direction) l'algorithme sera incapable de les différencier à la sortie.

Une possibilité pour contourner ce problème est d'augmenter le nombre d'observations disponibles, comme par exemple en ajoutant des caractéristiques permettant l'identification d'une cible. Mais ajouter un nombre important de caractéristiques (il y a des cas où il ne vaut mieux pas confondre deux cibles, être certain de l'identification demandera un grand nombre de données) fait en parallèle augmenter la dimension du système à résoudre. Pour des applications de haute précision et en temps réel cela risque d'être problématique.

6 Annexe

6.1 Codes

L'ensemble des fonctions, modules et figures utilisés pour réaliser ou approfondir ce TP sont disponibles à l'adresse suivante : https://github.com/Erwanlbv/TSP_P1_Kalman.git