

MA01 - PC5

GELMI DE FREITAS SALVO, Guilherme
guilherme.gelmi@ensta-paris.fr

SANTOS COSTA, Rian Radeck
rian.santos@ensta-paris.fr

1 Cadre du filtre de Kalman

Considérons le système d'équations suivant :

$$\begin{cases} x_{k+1} = F_k x_k + w_k \\ z_k = H_k x_k + v_k \end{cases}$$

avec $x_k \in \mathbb{R}^{n_x}$ le vecteur inconnu d'état du système au pas de temps k , z_k une mesure disponible au pas de temps k , w_k le bruit de modèle (ou d'état) suivant la loi normale $\mathcal{N}(0, W_k)$ avec W_k matrice connue, et v_k le bruit de mesures suivant la loi normale $\mathcal{N}(0, V_k)$ avec V_k matrice connue. Enfin, $F_k \in M_{n_x, n_x}(\mathbb{R})$ et $H_k \in M_{n_z, n_x}(\mathbb{R})$ sont deux matrices connues.

2 Les équations du filtre de Kalman

Le filtre de Kalman est un algorithme répondant à la problématique précédente. Outre l'initialisation, chaque itération se décompose en deux étapes : une étape de prédiction, et une étape de correction.

2.1 Algorithme

2.1.1 Initialisation

On prend une estimation \hat{x}_0 de x_0 arbitraire parmi les valeurs possibles de x_0 , et une matrice de covariance arbitrairement grande (par exemple $P_0 = \alpha I$ avec α arbitrairement grand).

2.1.2 Itération

Pour $k \geq 0$

— Prédiction

$$\begin{cases} \hat{x}_{k+1|k} = F_k \hat{x}_k \\ P_{k+1|k} = F_k P_k F_k^T + W_k \end{cases}$$

— Correction

Calcul du gain de Kalman :

$$K_{k+1} = P_{k+1|k} H_{k+1}^T (H_{k+1} P_{k+1|k} H_{k+1}^T + V_{k+1})^{-1}$$

Mise à jour :

$$\begin{cases} \hat{x}_{k+1} = \hat{x}_{k+1|k} + K_{k+1} (z_{k+1} - H_{k+1} \hat{x}_{k+1|k}) \\ P_{k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k} \end{cases}$$

Il faut percevoir que dans le problème demandé, les matrices F_k , H_k , V_k et W_k sont constantes pour tout k . Alors, on a mis en œuvre l'algorithme en tenant compte des égalités suivantes : $F_k = F$, $H_k = H$, $V_k = V$ et $W_k = W$, $\forall k \in \mathbb{N}$

3 Mise en œuvre

On a mis en œuvre l'algorithme sur python :

```
1 x_hat = []
2 P = []
3
4 # Initialization of x_hat (Predicted x0) and of P (Big enough initial covariance)
5 x_hat.append(np.array([0, 0, 0, 0, 0, 0]).T) # Column vector
6 P.append(1e3 * np.eye(6))
7
8
9 def step(z):
10     """
11     Does a step with a measurement z
12     """
13     def prediction():
14         x_hat_minus = F @ x_hat[-1]
15         P_minus = F @ P[-1] @ F.T + W
16         return x_hat_minus, P_minus
17
18     def correction(z):
19         x_hat_minus, P_minus = prediction()
20         # Kalman gain
21         K = P_minus @ H.T @ np.linalg.inv(H @ P_minus @ H.T + V)
22
23         # Update
24         x_hat_step = x_hat_minus + K @ (z - H @ x_hat_minus)
25         P_step = (np.eye(6) - K @ H) @ P_minus
26
27         x_hat.append(x_hat_step)
28         P.append(P_step)
29
30     correction(z)
31
32 # The measures are read from a text file before in
33 # the code and now we make the steps with them
34 for measure in measures:
35     step(measure)
```

L'implémentation est très simple et suit les itérations présentées dans la section 2.2. Pour mieux comprendre le code, nous pouvons établir les relations suivantes :

- $\hat{x}_{k+1|k}$ est appelé `x_hat_minus`
- $P_{k+1|k}$ est appelé `P_minus`
- \hat{x}_{k+1} est appelé `x_hat_step`
- P_{k+1} est appelé `P_step`
- `np.eye(n)` crée une matrice identité de taille n par n
- L'opérateur `@` correspond à une multiplication de matrices
- L'index -1 d'une liste (`A[-1]`) accède au dernier élément de cette liste
- L'attribut `.T` est la matrice transposée. (X^T c'est `X.T` dans le code)

4 Premier cas

Dans le premier cas les vecteur des mesures était limité à la position de la cible. Après l'exécution de l'algorithme on a arrivé a ces prédictions¹ :

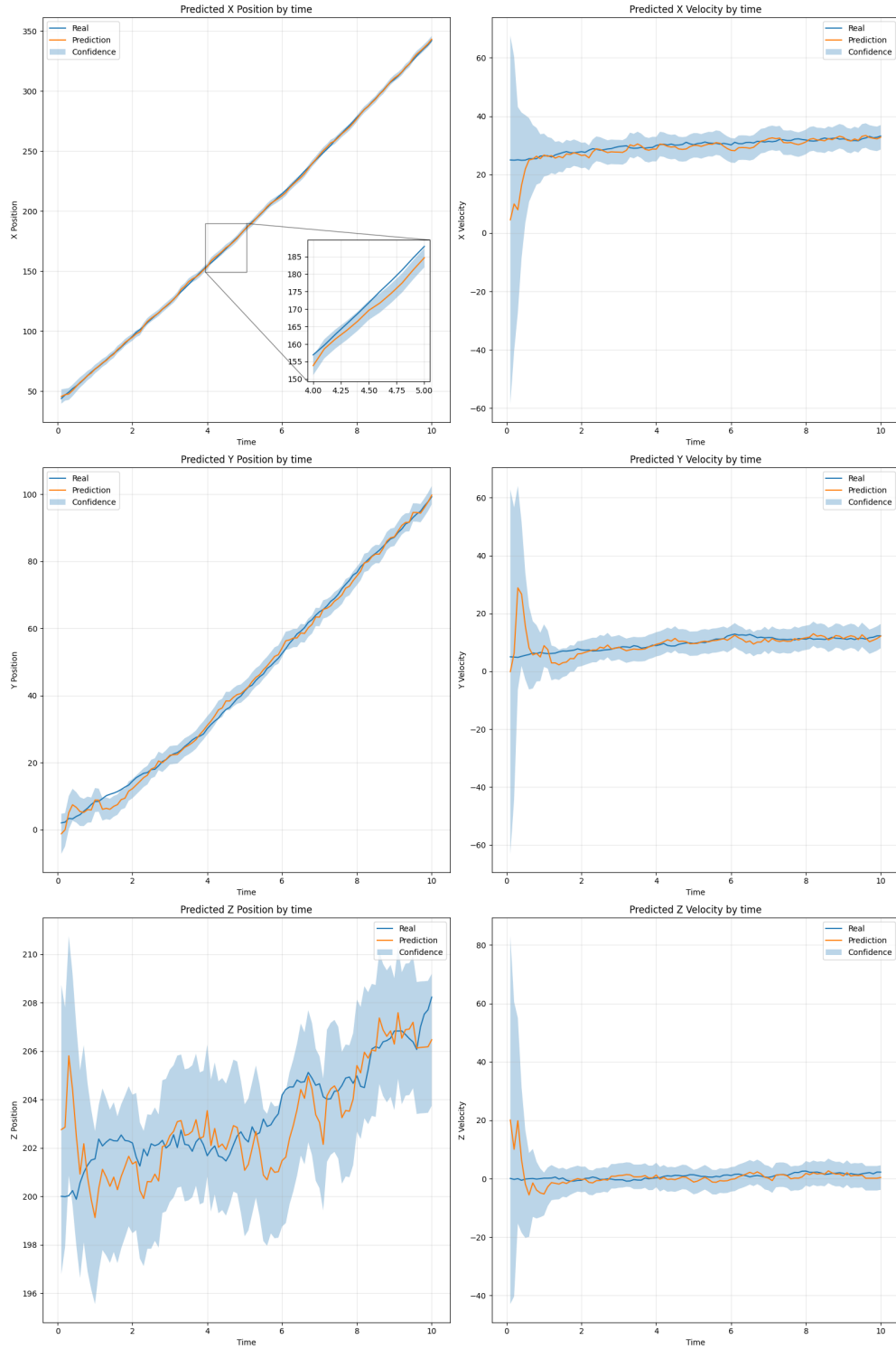


FIGURE 1 – Prédictions de l'algorithme pour les données de "filtreKalman1.txt"

1. La valeur de \hat{x}_0 a été supprimée du graphique, car elle reste toujours égale à 0 avec une variance de 10^3

On peut voir que la trajectoire de la valeur réelle de l'état est toujours (à l'exception de quelques points) située entre l'intervalle de confiance. Donc, la prédiction est suffisamment juste.

Pour vérifier la convergence et la correction de notre algorithme, on peut garantir que la trace de la matrice de covariance est toujours décroissante. Nous confirmons cette hypothèse avec le graphique et le tableau suivants.

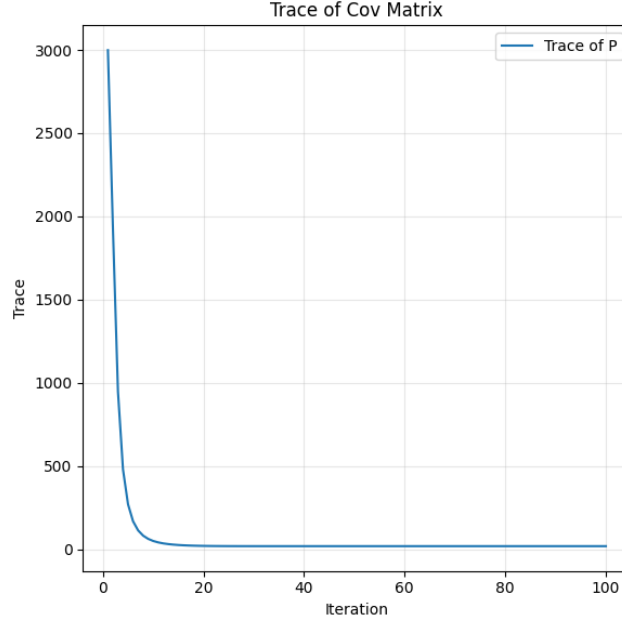


FIGURE 2 – Covariance de l'algorithme pour les données de "filtreKalman1.txt"

1 - 2998.5238	21 - 19.9612	41 - 18.7459	61 - 18.73992466	81 - 18.73984882
2 - 1925.0552	22 - 19.6336	42 - 18.7454	62 - 18.73990187	82 - 18.73984868
3 - 946.8684	23 - 19.3877	43 - 18.7448	63 - 18.73988540	83 - 18.73984855
4 - 479.7416	24 - 19.2047	44 - 18.7442	64 - 18.73987371	84 - 18.73984844
5 - 270.2130	25 - 19.0698	45 - 18.7437	65 - 18.73986556	85 - 18.73984834
6 - 168.1772	26 - 18.9714	46 - 18.7431	66 - 18.73985999	86 - 18.73984825
7 - 113.6223	27 - 18.9007	47 - 18.7426	67 - 18.73985627	87 - 18.73984817
8 - 82.0443	28 - 18.8505	48 - 18.7422	68 - 18.73985384	88 - 18.73984811
9 - 62.5629	29 - 18.8155	49 - 18.7418	69 - 18.73985229	89 - 18.73984806
10 - 49.9198	30 - 18.7916	50 - 18.7414	70 - 18.73985131	90 - 18.73984801
11 - 41.3776	31 - 18.7755	51 - 18.7411	71 - 18.73985070	91 - 18.73984798
12 - 35.4187	32 - 18.7650	52 - 18.7409	72 - 18.73985032	92 - 18.73984795
13 - 31.1555	33 - 18.7582	53 - 18.7407	73 - 18.73985006	93 - 18.73984793
14 - 28.0446	34 - 18.7539	54 - 18.7405	74 - 18.73984987	94 - 18.73984791
15 - 25.7397	35 - 18.7513	55 - 18.7403	75 - 18.73984971	95 - 18.73984789
16 - 24.0129	36 - 18.7496	56 - 18.7402	76 - 18.73984956	96 - 18.73984788
17 - 22.7089	37 - 18.7485	57 - 18.7401	77 - 18.73984941	97 - 18.73984788
18 - 21.7194	38 - 18.7477	58 - 18.7401	78 - 18.73984926	98 - 18.73984787
19 - 20.9670	39 - 18.7471	59 - 18.7400	79 - 18.73984911	99 - 18.73984787
20 - 20.3950	40 - 18.7465	60 - 18.7400	80 - 18.73984896	100 - 18.73984786

TABLE 1 – Valeurs du trace de P pour itération (Premier cas)

On observe ainsi que $Trace(P_{k+1}) < Trace(P_k)$, ce qui indique une amélioration à chaque itération. L'algorithme converge rapidement vers une valeur de covariance de 18,74. Le valeur 0 est toujours 6000 et a été supprimé.

5 Deuxième cas

Dans le deuxième cas on dispose des mesures de position et de vitesse de l'objet. Comme $F_k \in M_{6,6}(\mathbb{R})$, $H_k \in M_{6,6}(\mathbb{R})$ et $V_{k+1} \in M_{6,6}(\mathbb{R})$, les équation du filtre reste la même parce que la dimension des matrices est toujours correcte. Donc, l'algorithme reste le même avec une seule modification de la lecture des données. Les prédictions² réalisées sont :

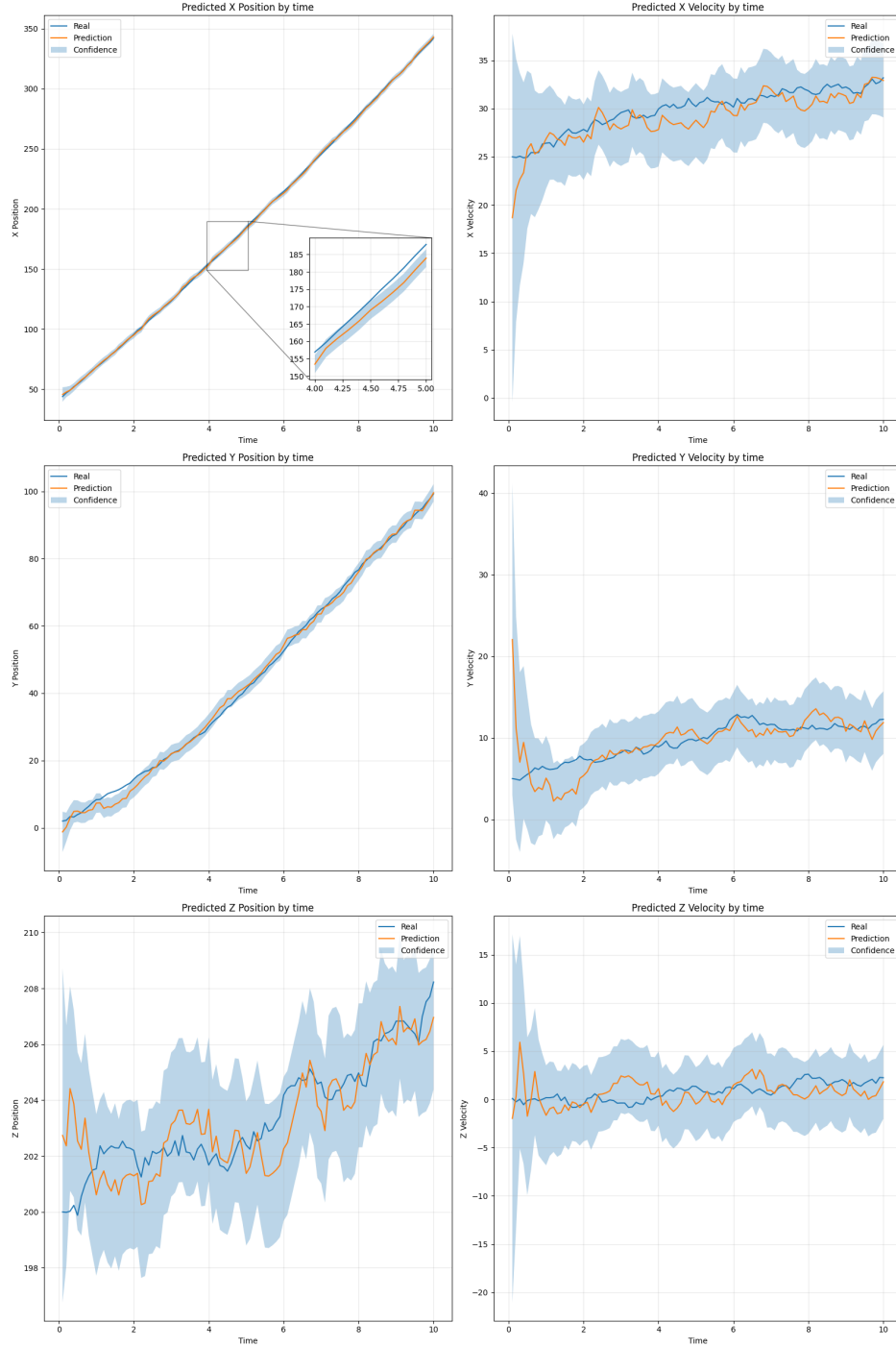


FIGURE 3 – Prédictions de l'algorithme pour les données de "filtreKalman2.txt"

2. La valeur de \hat{x}_0 a été supprimée du graphique, car elle reste toujours égale à 0 avec une variance de 10^3

On peut voir que pour les nouvelles mesures la trajectoire de la valeur réelle de l'état est toujours (à l'exception de quelques points) située entre l'intervalle de confiance. Donc, la prédiction est suffisamment juste.

Pour vérifier la convergence et la correction de pour le deuxième cas, on peut garantir que la trace de sa matrice de covariance est toujours décroissante. Nous confirmons cette hypothèse avec le graphique et le tableau suivants.

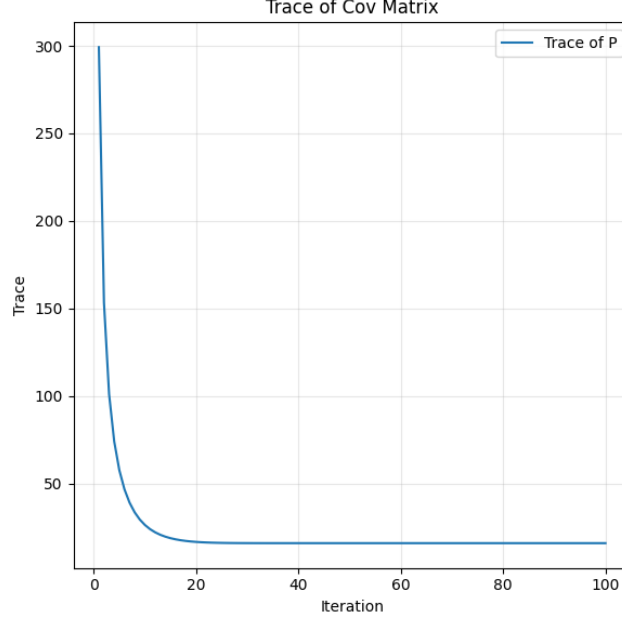


FIGURE 4 – Covariance de l'algorithme pour les données de "filtreKalman2.txt"

1 - 299.2513	21 - 16.5201	41 - 15.9867	61 - 15.98453556	81 - 15.98451056
2 - 153.2669	22 - 16.3807	42 - 15.9864	62 - 15.98452863	82 - 15.98451053
3 - 101.0380	23 - 16.2753	43 - 15.9861	63 - 15.98452351	83 - 15.98451051
4 - 74.0667	24 - 16.1963	44 - 15.9859	64 - 15.98451976	84 - 15.98451049
5 - 57.6640	25 - 16.1376	45 - 15.9857	65 - 15.98451705	85 - 15.98451047
6 - 46.7618	26 - 16.0944	46 - 15.9855	66 - 15.98451511	86 - 15.98451046
7 - 39.1198	27 - 16.0629	47 - 15.9854	67 - 15.98451374	87 - 15.98451044
8 - 33.5781	28 - 16.0401	48 - 15.9852	68 - 15.98451279	88 - 15.98451043
9 - 29.4664	29 - 16.0239	49 - 15.9851	69 - 15.98451213	89 - 15.98451042
10 - 26.3664	30 - 16.0125	50 - 15.9850	70 - 15.98451167	90 - 15.98451041
11 - 24.0018	31 - 16.0045	51 - 15.9849	71 - 15.98451136	91 - 15.98451041
12 - 22.1830	32 - 15.9990	52 - 15.9848	72 - 15.98451114	92 - 15.98451040
13 - 20.7758	33 - 15.9952	53 - 15.9847	73 - 15.98451099	93 - 15.98451040
14 - 19.6826	34 - 15.9925	54 - 15.9847	74 - 15.98451089	94 - 15.98451040
15 - 18.8316	35 - 15.9907	55 - 15.9847	75 - 15.98451081	95 - 15.98451039
16 - 18.1688	36 - 15.9895	56 - 15.9846	76 - 15.98451075	96 - 15.98451039
17 - 17.6529	37 - 15.9886	57 - 15.9846	77 - 15.98451070	97 - 15.98451039
18 - 17.2522	38 - 15.9880	58 - 15.9846	78 - 15.98451066	98 - 15.98451039
19 - 16.9421	39 - 15.9874	59 - 15.9846	79 - 15.98451062	99 - 15.98451039
20 - 16.7032	40 - 15.9870	60 - 15.9845	80 - 15.98451059	100 - 15.98451039

TABLE 2 – Valeurs du trace de P pour itération (deuxième cas)

On observe ainsi que $Trace(P_{k+1}) < Trace(P_k)$, ce qui indique une amélioration à chaque itération. L'algorithme converge rapidement vers une valeur de covariance de 15,98. Le valeur 0 est toujours 6000 et a été supprimé.

6 Comparaison

L'algorithme du filtre de Kalman commence par une prédiction de l'état courante a partir de l'état précédent. Dans la phase suivante on utilise les mesures pour corriger l'état prédit dans l'intention de obtenir plus de précision. Dans le premier cas on n'a pas les mesures sur la vitesse de la cible, ainsi la phase de mise à jour modifie la valeur calculée par la prévision en utilisant les mesures de position fois le facteur de la matrice de gain. Dans le deuxième, le mise à jour corrige l'état de vitesse prédit avec les mesures de vitesse et donc la estimation est plus précise. Ainsi, cela justifie le valeur de l'écart-type moins importante.

De plus, on peut voir que les état de vitesse s'approchent des données réelles plus rapidement quand on les corrige avec leurs mesures. Une dernière observation porte sur la trace de la matrice P, qui est plus faible dans le deuxième cas, ce qui indique une plus grande précision lorsqu'il y a plus d'informations disponibles pour la mesure.

7 Troisième cas

Dans le troisième cas on n'a que des mesures de vitesse.

7.1 Analyse de la matrice de gain de Kalman

$$F = \begin{pmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad P = \alpha I_6, \quad W = T \begin{pmatrix} \sigma_p^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_p^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_p^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_v^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_v^2 \end{pmatrix}$$

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Calcul pour k = 0 :

$$P_{k+1|k} = F_k P_k F_k^T + W_k = \alpha \begin{pmatrix} 1+T^2 & 0 & 0 & T & 0 & 0 \\ 0 & 1+T^2 & 0 & 0 & T & 0 \\ 0 & 0 & 1+T^2 & 0 & 0 & T \\ T & 0 & 0 & 1 & 0 & 0 \\ 0 & T & 0 & 0 & 1 & 0 \\ 0 & 0 & T & 0 & 0 & 1 \end{pmatrix} + T \begin{pmatrix} \sigma_p^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_p^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_p^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_v^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_v^2 \end{pmatrix}$$

$$H_{k+1} P_{k+1|k} H_{k+1}^T + V_{k+1} = \begin{pmatrix} \alpha + T\sigma_v^2 + \sigma_z^2 & 0 & 0 \\ 0 & \alpha + T\sigma_v^2 + \sigma_z^2 & 0 \\ 0 & 0 & \alpha + T\sigma_v^2 + \sigma_z^2 \end{pmatrix}$$

$$K_{k+1} = P_{k+1|k} H_{k+1}^T (H_{k+1} P_{k+1|k} H_{k+1}^T + V_{k+1})^{-1} = \begin{pmatrix} \frac{\alpha T}{\alpha + T\sigma_v^2 + \sigma_z^2} & 0 & 0 \\ 0 & \frac{\alpha T}{\alpha + T\sigma_v^2 + \sigma_z^2} & 0 \\ 0 & 0 & \frac{\alpha T}{\alpha + T\sigma_v^2 + \sigma_z^2} \\ \frac{\alpha + T\sigma_v^2}{\alpha + T\sigma_v^2 + \sigma_z^2} & 0 & 0 \\ 0 & \frac{\alpha + T\sigma_v^2}{\alpha + T\sigma_v^2 + \sigma_z^2} & 0 \\ 0 & 0 & \frac{\alpha + T\sigma_v^2}{\alpha + T\sigma_v^2 + \sigma_z^2} \end{pmatrix}$$

Nous pouvons noter que les termes diagonaux nous dit que n'existe pas un mélange des différentes composantes dans la phase de correction de l'état. Donc chaque composant de l'état est corrigée seulement pour leur composant de l'état prévit. Comme nous utilisons les mesures de vitesse pour estimer la position, le bruit, qui est plus important dans les mesures de vitesse, se propage à l'estimation de

la position. Si on fait $K_{k+1}H_{k+1}P_{k+1|k}$, on peut voir que les éléments diagonaux de la matrice de covariance dépendent du bruit σ_v .

Dans les premières itérations du filtre de Kalman, lorsqu'il y a peu d'informations sur les mesures de position, les prédictions ne seront pas corrigées de manière significative. Cependant, étant donné que des données de vitesse sont disponibles, le changement de position sera tout de même raisonnablement bien prédit. Cela signifie que si la position réelle X de la cible augmente, la position X prédite augmentera également, même s'il existe un écart important en termes d'amplitude entre les deux valeurs. Par exemple, si la prédiction initiale de la position X commence à 0 et que la position réelle est de 1000, si la position X augmente de manière linéaire selon un facteur α , la prédiction suivra également une augmentation linéaire avec un facteur proche de α .

Une autre observation importante est que le gain de Kalman est proportionnel à la matrice de covariance prédite et à l'inverse de la somme de la matrice de mesure et de la matrice de covariance prédite. Cela implique un *"trade-off"* entre ces deux facteurs lors de la correction de la prédiction précédente. À un instant donné, les informations ajoutées par la mesure peuvent ne pas être suffisantes pour contrebalancer pleinement l'effet de la variabilité inhérente du système, ce qui entraîne une augmentation de la matrice de covariance a posteriori. Ce concept est encore plus clair dans le cas d'un pas de temps sans observations, où aucune nouvelle information n'est introduite et la matrice de covariance a posteriori augmente inévitablement.