
TP - FILTRE DE KALMAN

Nossa Iyamu
Telecom SudParis
Institut Polytechnique de Paris
Evry
Nossa.iyamu@telecom-sudparis.eu

ABSTRACT

L'étude vise à comprendre le filtre de Kalman à travers son utilisation. Nous nous positionnons dans le contexte où l'on observe une cible et on veut déterminer sa trajectoire réelle.

Keywords Filtre · Kalman

Introduction

Dans la première partie nous voulons filtrer l'observation d'une trajectoire disposant de ses coordonnées cartésiennes. Dans la seconde partie nous voulons filtrer l'observation d'une trajectoire disposant de ses coordonnées polaires. Dans la dernière partie nous discutons de la possibilité de poursuivre plusieurs cibles.

1 Partie 1: Filtrage cartésien

1.1 Génération d'une trajectoire

Nous générons artificiellement une trajectoire aléatoire: à partir d'une position initiale, nous calculons par récurrence les positions suivante comme combinaison linéaire de la position précédente sommée d'un bruit gaussien U_k , soit:

$$X_{k+1} = F \cdot X_k + U_k$$

X_k représentant le vecteur des positions et vitesse, de la forme: $X_k = [p_x, \dot{p}_x, p_y, \dot{p}_y]_{t=k}^T$

1.2 Modélisation des observations

Ensuite nous voulons modéliser l'observation que nous aurions eu dans le cas réel, c'est à dire l'observation d'une trajectoire bruitée. Pour cela, on crée un vecteur qui est la trajectoire calculée précédemment supplée d'un bruit gaussien V_k :

$$Y_{k+1} = H \cdot X_{k+1} + V_k$$

La matrice H nous sert à relever les coordonnées p_x et p_y seulement.

1.3 Prédiction de la position réelle

Finalement, nous voulons prédire la position réelle de la cible en filtrant la trajectoire bruitée. Nous utilisons pour cela, le filtre de Kalman.

1.3.1 Calcul des relations

Pour utiliser le filtre de Kalman, il nous faut trouver les relations de passage:

$$\hat{x}_{k|k}, P_{k|k} \rightarrow \hat{x}_{k+1|k+1}, P_{k+1|k+1}$$

avec

$$\hat{x}_{k|k} = m_{k|k} = E[X_k | y_{0:k}], \quad P_{k|k} = Covar[X_k | y_{0:k}]$$

Nous utilisons pour cela la loi de Bayes ainsi qu'une formule de convolution de loi normales:

$$P(X_{t+1} | y_{0:t+1}) = \frac{P(y_{t+1} | y_{0:t}, X_{t+1}) P(X_{t+1} | y_{0:t})}{\int_0^\infty \text{numérateur} dX_{t+1}}$$

$$\int_0^\infty N(y, Ax + b, L) N(x, \mu, G) dx = N(y, a\mu + b, L + AGA^T)$$

Nous obtenons finalement que

$$X_{t+1} | y_{0:t+1} \sim \frac{N(y_{t+1}, HX_{t+1}, R) N(X_{t+1}, m_{t+1|t}, P_{t+1|t})}{\int_0^\infty (\text{numérateur}) dX_{t+1}}$$

Enfin, une dernière loi de convolution nous permet d'avoir la forme qu'on veut du numérateur:

$$N(y_{t+1}, HX_{t+1}, R) N(X_{t+1}, m_{t+1|t}, P_{t+1|t}) = q(y_{t+1}) N(X_{t+1}, m_{t+1|t+1}, P_{t+1|t+1})$$

avec

1.

$$m_{t+1|t+1} = m_{t+1|t} + k(y_{t+1} - Hm_{t+1|t})$$

2.

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

3.

$$P_{t+1|t+1} = (I_4 - kH) P_{t+1|t}$$

1.3.2 Utilisation des relations

Après implémentation des ces équations dans le notebook, nous obtenons les figures suivantes (voir annexe)

On observe donc que le filtre génère à partir d'une trajectoire bruitée une trajectoire linéaire qui de plus est proche de la trajectoire non bruitée d'origine. En calculant l'erreur quadratique moyenne on trouve une erreur de 7. Cette erreur dépend des bruits de mesure σ_{px}, σ_{py} .

En effet pour une même trajectoire, lorsque l'on augmente ces valeurs, l'erreur quadratique moyenne augmente: L'erreur varie de 7 à 87 lorsque l'on multiplie ces valeurs par 10.

2 Partie 2: Filtrage polaire

2.1 Environnement polaire

Nous disposons à présent des coordonnées polaires qui sont fortement liées aux coordonnées cartésiennes:

Coordonnées polaires : (r, θ)

Coordonnées cartésiennes : (x, y)

Formules :

$$x = r \cdot \cos(\theta)$$

$$y = r \cdot \sin(\theta)$$

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \arctan\left(\frac{y}{x}\right)$$

2.2 Transposition du problème

Le vecteur d'observation est donc à présent

$$Y_t = [r(X_t), \theta(X_t),] + V_t$$

soit

$$Y_t = h(X_t) + V_t$$

D'où:

$$Y_t | X_t \sim N(h(X_t), R)$$

En remplaçant alors dans l'équation de la partie 1, le produit HX_t par les coordonnées polaires de la position $h(X_t)$, on remarque que la relation liant X à Y n'est plus linéaire. L'utilisation du filtre de Kalman tel quel utilisé en partie 1 n'est donc plus possible, il est nécessaire de faire une approximation.

2.3 Approximation

Nous allons linéariser la fonction h au point $\hat{x}_{t|t-1}$ car il s'agit de la meilleure approximation de la position à cet instant. La formule de Taylor au premier ordre nous donne:

$$f(x) \approx f(a) + \nabla f(a) \cdot (x - a)$$

On obtient donc

$$Y_t = h(m_{t|t-1}) + \nabla h(m_{t|t-1}) \cdot (X_t - m_{t|t-1}) + V_t$$

D'où:

$$Y_t | X_t \sim N(h(m_{t|t-1}) + \nabla h(m_{t|t-1}) \cdot (X_t - m_{t|t-1}), R)$$

Cependant, en posant $Z_t = Y_t - h(m_{t|t-1}) + \nabla h(m_{t|t-1}) \cdot m_{t|t-1}$, on a:

$$Z_t = \nabla h(m_{t|t-1}) \cdot X_t + V_t$$

$$Z_t | X_t \sim N(\nabla h(m_{t|t-1}) \cdot X_t, R)$$

Nous pouvons donc à présent se ramener au problème de la partie 1, soit:

$$X_{t+1} | Z_{0:t+1} \sim \frac{N(Z_{t+1}, \nabla h(m_{t+1|t})X_t + 1, R)N(X_{t+1}, m_{t+1|t}, P_{t+1|t})}{\int_0^\infty (\text{numérateur}) dX_{t+1}}$$

avec

1.

$$m_{t+1|t+1} = m_{t+1|t} + k(y_{t+1} - \nabla h(m_{t+1|t})m_{t+1|t})$$

2.

$$k = P_{t+1|t} \nabla h(m_{t+1|t})^T (\nabla h(m_{t+1|t}) P_{t+1|t} \nabla h(m_{t+1|t})^T + R)^{-1}$$

3.

$$P_{t+1|t+1} = (I_4 - k \nabla h(m_{t+1|t})) P_{t+1|t}$$

2.4 Filtre de Kalman pour détection polaire

Comme on peut le constater en annexe, le filtrage de Kalman étendu nous permet de surmonter le problème du modèle polaire. Cependant, pour un tel modèle, on fait une approximation supplémentaire, celle de la position de X . Si l'utilisation de cette méthode n'est pas rigoureuse, cela pourrait engendrer des erreurs supplémentaire pour lors du filtrage.

3 Poursuite multi-cible

3.1 Obstacles

Plusieurs obstacles sont à prendre en compte pour ce qui est du filtrage multi-cible, tels que:

- Ambiguïté de l'association des cibles : Dans une scène vidéo avec plusieurs objets en mouvement, il peut être difficile d'associer correctement les mesures aux cibles correspondantes. Cela peut conduire à des erreurs d'association, ce qui compromet la précision de la poursuite.
- Occlusions : Les objets peuvent être partiellement ou complètement occultés par d'autres objets, rendant difficile la mise à jour précise de l'état de la cible. Cela peut entraîner des erreurs dans la prédiction et la correction de la trajectoire.
- Gestion des cibles multiples : La gestion simultanée de plusieurs cibles peut augmenter la complexité du suivi, car les interactions entre les différentes cibles doivent être prises en compte.

3.2 Solutions

Des solutions permettent de partiellement résoudre ces obstacles telles que:

- Détection des cibles: Utiliser un algorithme pour identifier chaque objet dans chaque image de la séquence vidéo. Ceci est fait par des technique de détection d'objet tel que YOLO (You Only Look Once) ou SSD (Single Shot Multibox Detector)
- Filtres de Kalman: Initialiser un filtre de Kalman pour chaque cible détectée, dont l'état initial serait basé sur la position détectée de la cible et d'autres informations pertinentes.
- Interactions : Si les cibles interagissent entre elles, il faudrait ajuster les modèles de mouvement et les stratégies d'association pour tenir compte de ces interactions.
- Optimisation des performances : Il est possible d'optimiser les performances du système en ajustant les paramètres des filtres de Kalman, en optimisant les seuils de détection, et en explorant d'autres améliorations possibles.

4 Conclusion

Le Filtre de Kalman est efficace pour approximer une trajectoire dans un domaine cartésien, mais un peu plus complexe dans le domaine polaire. Il devient plus difficile à manipuler à mesure que le problème se complexifie, par exemple lorsque il y a plusieurs cibles dont le comportement n'est pas simple à modéliser. Il existe cependant des méthodes pour résoudre ce type de problème tel que cités plus haut, ou encore l'utilisation du filtre de Kalman étendu utilisé en partie 2, permettant de gérer un système non linéaire. Cependant, il est essentiel de disposer de machines suffisamment puissantes en termes de puissance de calcul. Il reste cependant un très bon outil pour le filtrage de comportement et peut aussi avoir un usage pertinent dans le domaine de la finance, notamment pour la prédiction de la volatilité des marchés financiers.

Annexe

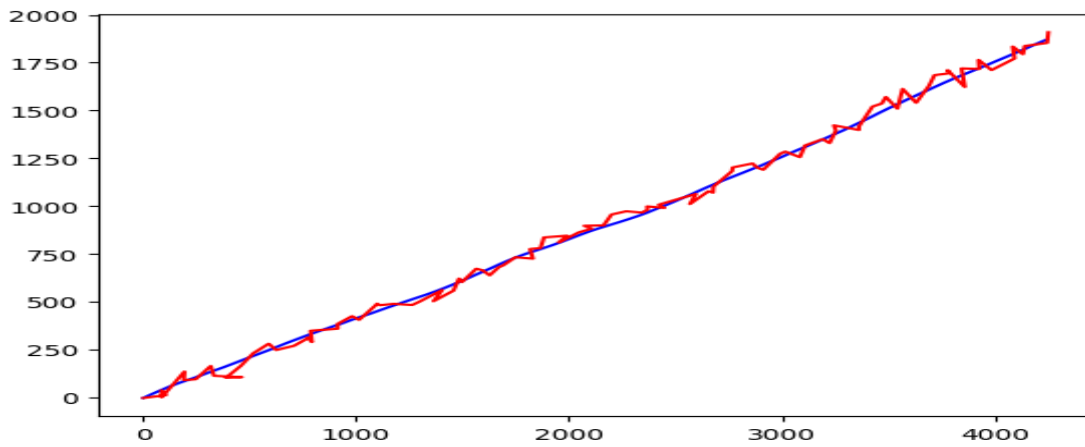


Figure 1: Trajectoire réelle (bleu) et trajectoire réelle bruitée (rouge)



Figure 2: Trajectoire bruitée (rouge) et trajectoire filtrée(vert)

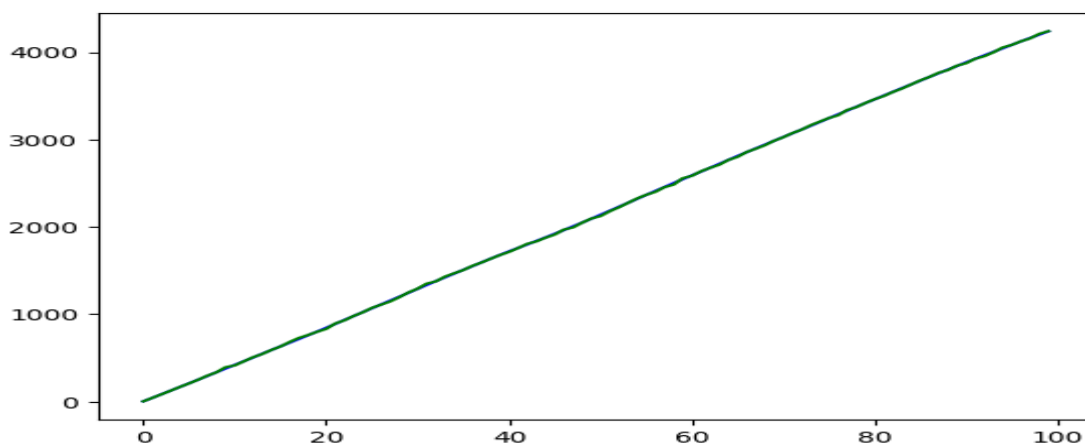


Figure 3: Trajectoire réelle (bleu) et trajectoire filtrée (vert)

PARTIE 1

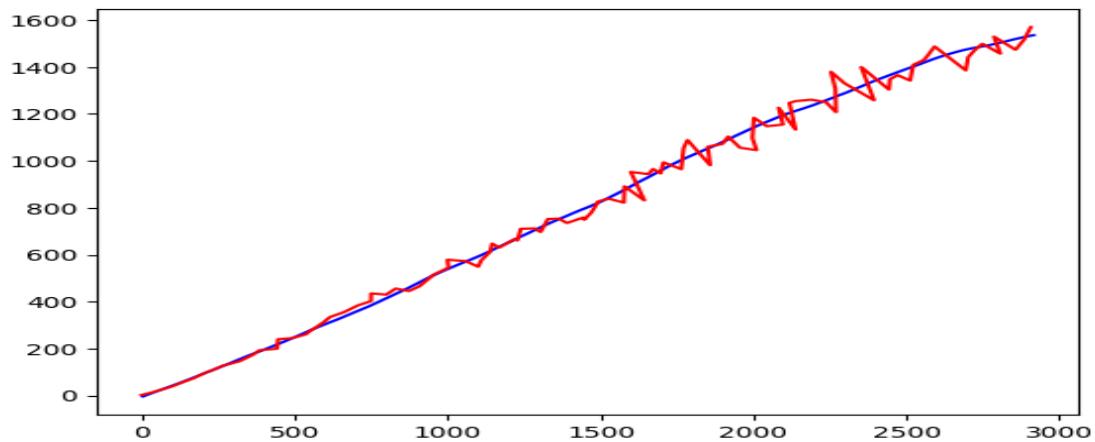


Figure 4: Trajectoire bruitée (rouge) et filtrée (bleu) par filtre de Kalman étendu

```
[48] def filtre_de_kalman_radar(F, Q, R, y_k, x_kalm_prec, P_kalm_prec): #doit etre modifié car on remplace le produit lineaire Hxk par une opération qui n'est plus linéaire en xk
    x = x_kalm_prec
    P = P_kalm_prec
    x_int = np.dot(F, x)
    P_int = Q + np.dot(F, np.dot(P, F.T))
    x_int2 = x_int[0]**2 + x_int[2]**2
    x2 = x[0]**2 + x[2]**2
    r = np.sqrt(x2)
    teta = np.arctan(x[2]/x[0])
    #il m'aurait fallu h(mt+1/t) et grad_h(mt+1/t) mais grace a la traslation Z je n'ai besoin que de h(mt+1/t)
    h = np.array([r, teta])
    grad_h = np.array([(x_int[0]/r, 0, x_int[2], 0), [-1/2, 0, x[0]/x2, 0]])
    K = P_int @ grad_h.T @ np.linalg.inv(grad_h @ P_int @ grad_h.T + R)
    x = x_int + np.dot(K, y_k - np.dot(grad_h, x_int))
    P = np.dot(np.eye(4) - np.dot(K, grad_h), P_int)
    return x, P

#P = P_kalm = I4
x_est = np.zeros((4,T))
P = np.zeros((T,4,4))
x_est[0,0] = x_init
P[0, :, :] = P_kalm
for i in range(1,T):
    x_est[:,i] = filtre_de_kalman_radar(F, Q, R, vecteur_y[:,i], vecteur_x[:,i-1], P[i-1, :, :])[0]
    P[i, :, :] = filtre_de_kalman_radar(F, Q, R, vecteur_y[:,i], vecteur_x[:,i-1], P[i-1, :, :])[1]

plt.plot(x_est[0,:], x_est[2,:], color = 'green')
plt.plot(vecteur_x[0,:], vecteur_x[2,:], color = 'blue')
#plt.plot(vecteur_y[0,i], vecteur_y[1,i], color = 'red')

#print(x_est)
```

Figure 5: Code du filtre de Kalman étendu
PARTIE 2