

APP Robots On Antarctica

PHASE 2 : Fonctions avancées de Matlab – projet Kalman

L'objectif de cette seconde phase est de vous perfectionner avec Matlab, notamment avec les commandes avancées (fonctions, scripts de scripts).

Afin de vous préparer à la tâche Cartographeur de votre mission, le responsable scientifique en Antarctique vous envoie en stage au laboratoire IRIMAS à Mulhouse où les équipes de recherche travaillent sur la localisation appliquée aux véhicules autonomes. Par localisation, on définit la position et l'orientation d'un mobile (véhicule ou robot). D'ordinaire, ces données sont fournies par les systèmes de navigation (TomTom, Navigon...) basés sur les données GPS. Malheureusement, les données GPS ne sont pas toujours fiables (à proximité d'immeubles, dans un tunnel, etc...). L'idée est donc de coupler (ou *fusionner*) les données GPS avec les signaux mesurés par les capteurs dits « proprioceptifs ». Cela permet d'avoir une localisation plus robuste aux pertes de signal GPS. Bien sûr, vous n'aurez pas à tout inventer, on vous a envoyé chez des experts de la fusion de données (vos fidèles serveurs Messieurs Lauffenburger et Josso-Laurain).

Durant cette phase 2, vous découvrirez des fonctions avancées de Matlab dans un contexte de recherche, en vous basant sur des données réelles récoltées sur le véhicule instrumenté de l'IRIMAS. Vous prendrez également un peu d'avance sur le programme (filtrage de Kalman) !

A chaque fin de mission, vous validerez vos résultats avec l'enseignant. N'oubliez pas de bien commenter vos scripts et de mettre des légendes aux figures !

Mission 1 : Etude du positionnement par GPS

Votre première mission consiste à étudier le positionnement GPS du véhicule instrumenté de l'IRIMAS qui a parcouru un trajet autour de l'université. A cause de la végétation environnante, vous verrez que le signal GPS n'est pas toujours fiable, d'où la nécessité de fusionner avec les capteurs embarqués. L'ensemble des mesures relevées pendant le parcours est consignée dans le fichier *dataGps.mat* :

Colonne	Intitulé	Données	Unités
1	Tgps	Base de temps du GPS	s
2	Xgps	Abscisse de la position véhicule	m
3	Ygps	Ordonnée de la position véhicule	m
4	Qgps	Note de qualité du GPS	Sans unité
5	DataOk	Indicateur de réception d'une nouvelle donnée GPS	Sans unité

Tableau 1 – Nomenclature du fichier dataGps.mat



Figure 1. Véhicule instrumenté de l'IRIMAS : ARTEMPIS

1. Chargement du jeu de données

Chargez le fichier dataGps.mat

2. Nouvel échantillonnage

Le récepteur GPS fournit ces mesures à une fréquence de $F_{\text{gps}}=10\text{Hz}$ ($T_{\text{gps}}=0.1\text{s}$). Néanmoins, l'enregistrement de l'ensemble des données (GPS et capteurs proprioceptifs) s'effectue à une fréquence de $F_e=50\text{Hz}$ ¹. Aussi, l'indicateur DataOk permet d'identifier la disponibilité d'une nouvelle mesure GPS. Cette donnée est en fait un nombre binaire valant alternativement -1 ou 1. Chaque transition entre ces 2 valeurs (tous les $1/10\text{Hz}=0.1\text{s}$) permet de savoir qu'une nouvelle donnée est transmise par le GPS au système d'acquisition.

Ré-échantillonnez les données GPS à 10Hz et créez les vecteurs Tgps, Xgps, Ygps, Qgps et DataOk en y affectant les valeurs correspondantes

Indice : DataOk

On utilisera exclusivement ces vecteurs dans la suite des traitements.

3. Affichage

Représentez sur la figure précédente la trajectoire non interpolée et annotez (titre, axes, légende...).

La résolution de la trajectoire se définit par la distance entre deux points successifs. Tracez l'évolution de la résolution de la trajectoire en fonction du temps.

Indice : fonctions diff et sum...

Déterminez la distance parcourue par le véhicule durant cet essai.

¹ Donc 5 fois plus rapidement que la réactualisation des mesures GPS. Ceci explique que dans le fichier GPS, on retrouve à chaque fois 5 valeurs identiques. Pour s'en rendre compte, on pourra par exemple tracer sur une figure Tgps en fonction du nombre d'échantillons du fichier.

4. Qualité de réception GPS

Q_{gps} est un indicateur de la qualité de réception du GPS. Cette note varie entre 0 (aucune réception) et 18 (réception maximale) : $Q_{gps} \in [0; 18]$

Représenter sur la figure du tracé du circuit, en vert les points de la trajectoire correspondant à une note de qualité de 18, en rouge ceux dont la note de qualité est inférieure à 18. Annoter la figure comme il se doit (titre, axes, légende...).

5. Validation

Validez votre script avec l'enseignant

Mission 2 : Etude du positionnement par données proprioceptives

La localisation à l'estime est une technique qui permet de déterminer la pose (position et cap (x_n, y_n, ψ_n)) d'un véhicule par rapport à un repère lié à sa pose initiale (x_0, y_0, ψ_0) . Le principe s'appuie sur la sommation de vecteurs de déplacements élémentaires. Ces vecteurs de déplacements sont obtenus par intégration des mesures des capteurs. L'inconvénient majeur de cette approche est que la détermination des coordonnées du véhicule est altérée par les erreurs et les dérives des capteurs. Si aucun recalage n'est effectué, la précision du positionnement se dégrade rapidement. Cette solution ne peut donc être utilisée qu'à court terme, en complément d'une méthode de localisation plus robuste telle celle issue d'un GPS.

A partir des mesures disponibles sur le véhicule à un instant n (vitesses longitudinale (V_L), transversale (V_T) et la vitesse de lacet ($\dot{\psi}$)²) (cf. Figure), la pose du véhicule à l'instant $n+1$ s'obtient par :

$$\begin{cases} x_{n+1} = x_n + T_{ech} \cdot (V_{Ln} \cdot \cos \psi_n - V_{Tn} \cdot \sin \psi_n) \\ y_{n+1} = y_n + T_{ech} \cdot (V_{Ln} \cdot \sin \psi_n + V_{Tn} \cdot \cos \psi_n) \\ \psi_{n+1} = \psi_n + T_{ech} \cdot \dot{\psi}_n \end{cases}$$

où T_{ech} est la période d'échantillonnage de l'acquisition, soit 0.02s ($T_{ech}=1/50\text{Hz}$).

Le fichier dataCapt.mat contient les mesures des capteurs proprioceptifs correspondantes au fichier GPS précédemment traité :

Colonne	Intitulé	Données	Unités
1	Tcapt	Base de temps des mesures capteurs	s
2	Vt	Vitesse transversale du véhicule	m/s
3	VL	Vitesse longitudinale du véhicule	m/s
4	Psip	Vitesse de lacet du véhicule	rad/s

Tableau 2 – Nomenclature du fichier dataCapt.mat

² La vitesse de lacet représente la vitesse de rotation du véhicule autour de son axe vertical.

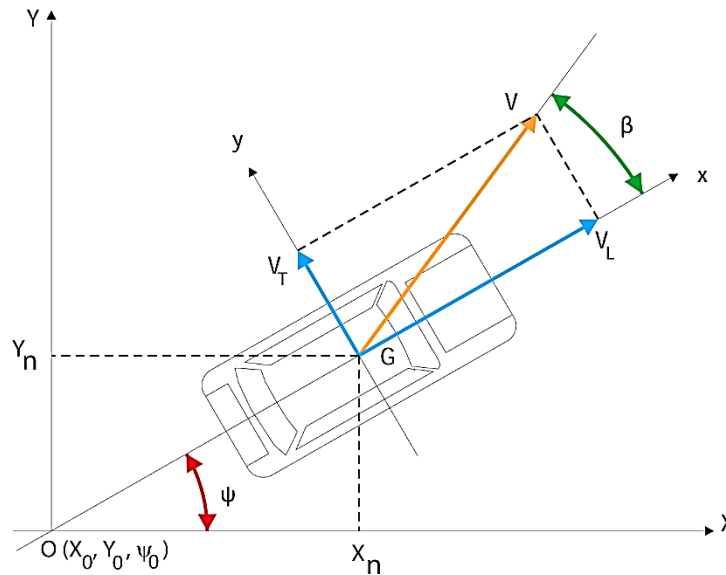


Figure 2 – Modèle d'évolution du véhicule

1. Extraction des données

Chargez le fichier dataCapt.mat.

Créez les vecteurs « colonne » Tcapt, Vt, Vl et Psip avec les valeurs précédemment chargées.

On utilisera exclusivement ces vecteurs dans la suite des traitements.

2. Biais de mesures

Le gyromètre, mesurant la vitesse de lacet $\dot{\psi}$ (variable Psip), est victime d'un biais. A partir des mesures de ce capteur sur les 10 premières secondes de l'acquisition, déterminez la valeur moyenne de cet écart (le véhicule se déplaçant en ligne droite, sa rotation autour de son axe vertical est considérée comme nulle) et compensez-le.

Tracez sur une même figure le signal biaisé et le signal compensé.

Calculez et tracez l'évolution du cap ψ_n pour les deux cas de figure (données biaisée et non biaisée).

3. Localisation proprioceptive du véhicule

Déterminez la localisation du véhicule (x_n, y_n, ψ_n) à partir des mesures des capteurs embarqués dans le cas de la mesure de lacet non biaisée. On posera comme valeurs initiales :

$$\begin{cases} x_0 = 0 \text{ m} \\ y_0 = 0 \text{ m} \\ \psi_0 = -2.18 \text{ rad} \end{cases}$$

Tracer sur une même figure la trajectoire ainsi obtenue.

Tracer sur une même figure, la trajectoire GPS et la trajectoire « non biaisée » issue du calcul précédent.

4. Validation

Validez votre script avec l'enseignant.

Mission 3 : Fusion GPS – capteurs proprioceptifs

On propose de fusionner la position obtenue « non biaisée » dans la mission 2 et celle obtenue grâce au GPS par l'intermédiaire d'un filtre de Kalman étendu³. Le principe de cette démarche est décrit par la Figure . Le filtre s'appuie sur l'ensemble des données (GPS et capteurs proprioceptifs) pour déterminer la localisation finale (X_{EST} , Y_{EST}) du véhicule.

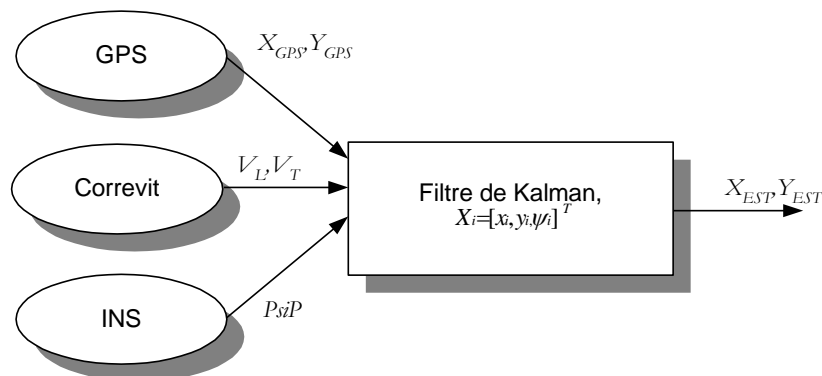


Figure 3 – Architecture de l'algorithme de Fusion

Un filtre de Kalman est un estimateur statistique récursif, articulé autour de 2 étapes (cf. Figure 1) :

- Une étape de prédiction : à partir des mesures proprioceptives (V_T , V_L , ψ) et des équations du modèle d'évolution, on prédit les nouveaux états du système considéré ainsi que l'incertitude sur cette prédiction,
- Une étape de correction : lorsqu'une nouvelle observation (position GPS) est disponible, la prédiction précédemment effectuée est corrigée.

Cette technique permet ainsi de compenser les dérives de la localisation à l'estime à chaque fois (toutes les 0.1s) qu'une mesure GPS valide (note de qualité égale à 18) est disponible. Les étapes qui vont suivre vont vous permettre d'implémenter ce filtre sous Matlab.

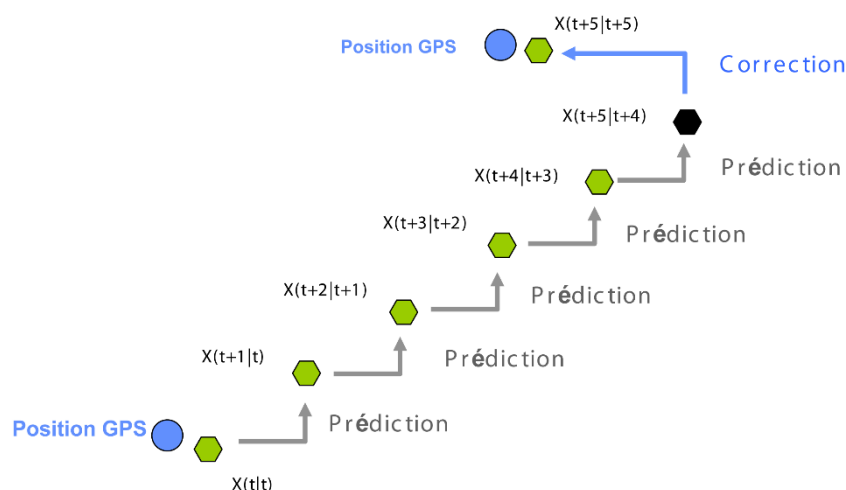


Figure 1 – Principe de prédiction-correction

On note $X_i = [x_i, y_i, \psi_i]^T$ l'état caractérisant la localisation du véhicule.

³ Il ne s'agit pas ici de s'intéresser à la théorie du filtrage de Kalman (étudié en 3^{ème} année !!) mais uniquement de programmer ce procédé avec Matlab. Il ne sera donc pas obligatoirement nécessaire de se confronter à la théorie relative à cette solution.

1. Définition des constantes

Un filtre de Kalman nécessite la configuration de différentes matrices (bruits de mesures, bruits d'états, etc.) permettant de tenir compte des imperfections des capteurs.

Définir la matrice Q des bruits du modèle telle que $Q = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.0001 * \pi/180 \end{bmatrix}$

Définir la matrice des bruits de mesure du GPS telle que $R_{gps} = \begin{bmatrix} 0.015^2 & 0 \\ 0 & 0.015^2 \end{bmatrix}$

Définir la matrice des bruits de mesures des capteurs proprioceptifs telle que

$$R_{pro} = \begin{bmatrix} 8 \cdot 10^{-7} & 0 & 0 \\ 0 & 7.96 \cdot 10^{-7} & 0 \\ 0 & 0 & 4.17 \cdot 10^{-6} \end{bmatrix}$$

Définir la matrice d'observation reliant les mesures GPS au vecteur d'état $X(x_n, y_n, \psi_n)$ telle que

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Définir la matrice d'incertitude (de covariance) de la prédiction telle que

$$P = \begin{bmatrix} R_{gps}(1,1) & 0 & 0 \\ 0 & R_{gps}(2,2) & 0 \\ 0 & 0 & \pi/3 \end{bmatrix}$$

2. Programmation du filtre de Kalman

Comme les mesures des capteurs proprioceptifs sont acquises à une fréquence de 50Hz alors que les mesures du GPS sont réactualisées à une cadence de 10Hz, la phase de correction sera effectuée uniquement lorsqu'une nouvelle mesure du GPS est disponible (toutes les 0.1s), la phase de prédiction étant réalisée à la cadence de 50Hz. Cet algorithme se base sur le fait que le GPS est le capteur le plus précis. Par conséquent, il sert de capteur de référence et est utilisé pour corriger la position obtenue par localisation à l'estime (cf. Figure 1). Lorsqu'aucune mesure GPS valide n'est disponible (note de qualité différente de 18), seules les équations de localisation à l'estime sont utilisées (sans aucune correction). Ce principe est présenté sur le pseudo-code suivant :

Pour n=debut jusqu'à fin de mesures

%Phase de Prédiction du filtre

Calculer A et B

Prédire les états X du modèle d'évolution (cf. Equation 1)

Calculer la matrice de covariance de la prédiction $P = A \cdot P \cdot A^T + B \cdot R_{pro} \cdot B^T + Q$

%Phase de correction

Si nouvelle localisation GPS valide

Calculer le gain de Kalman $K = P \cdot C^T \cdot [C \cdot P \cdot C^T + R_{gps}]^{-1}$

Définir un vecteur des mesures GPS $measure_gps = \begin{bmatrix} X_{gps}(n) \\ Y_{gps}(n) \end{bmatrix}$

Corriger la prédiction des états $X = X + K \cdot (measure_gps - C \cdot X)$

Corriger la matrice de covariance $P = (I - K \cdot C) \cdot P$

Fin si

Fin pour

Avec :

$$A = \begin{bmatrix} 1 & 0 & -T_{ech} \cdot (V_{T_{n-1}} \cdot \cos(\psi_{n-1}) + V_{L_{n-1}} \cdot \sin(\psi_{n-1})) \\ 0 & 1 & T_{ech} \cdot (V_{L_{n-1}} \cdot \cos(\psi_{n-1}) - V_{T_{n-1}} \cdot \sin(\psi_{n-1})) \\ 0 & 0 & 1 \end{bmatrix},$$

$$B = \begin{bmatrix} T_{ech} \cdot \cos(\psi_{n-1}) & -T_{ech} \cdot \sin(\psi_{n-1}) & 0 \\ T_{ech} \cdot \sin(\psi_{n-1}) & T_{ech} \cdot \cos(\psi_{n-1}) & 0 \\ 0 & 0 & T_{ech} \end{bmatrix} \text{ et } I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Traduisez ce pseudo-code en langage Matlab et exécutez votre script afin de déterminer la localisation du véhicule à l'aide de ce filtre.

3. Validation

Validez vos scripts avec l'enseignant.