

ISAE-SUPAERO - Bureau d'étude

Navigation pour un véhicule terrestre par Filtrage de Kalman

Ph. MOUYON
philippe.mouyon@onera.fr

L'objectif du bureau d'étude est de développer et tester un algorithme de navigation pour un véhicule terrestre de type voiture doté d'un ensemble de capteurs.

1 Modèle de simulation

Les équations de la cinématique à considérer sont celle d'un objet en translation et rotation dans le plan horizontal :

$$\begin{cases} \dot{x} &= V_x \\ \dot{y} &= V_y \\ \dot{\theta} &= \omega \end{cases}$$

où x et y sont les coordonnées du centre de l'essieu arrière (G). V_x et V_y sont les composantes de la vitesse de (G), et ω la vitesse de rotation du véhicule.

Le cap du véhicule est noté θ , et φ est l'angle de braquage des roues. On approxime le comportement à celui d'une bicyclette. On montre alors (par composition des vitesses) que :

$$\begin{cases} V_x &= V_{av} \cos(\theta + \varphi) - L\omega \sin(\theta) \\ V_y &= V_{av} \sin(\theta + \varphi) + L\omega \cos(\theta) \end{cases}$$

où V_{av} est la vitesse de la roue avant et L la distance inter-essieux. On pourra utiliser l'approximation:

$$\begin{cases} V_x &\approx V \cos(\theta) \\ V_y &\approx V \sin(\theta) \end{cases}$$

où V est le module de la vitesse de G, quand l'angle de braquage et la vitesse de rotation sont petits. C'est cette approximation qui est utilisée dans la simulation du véhicule.

Toujours pour des angles de braquage (φ) petits, on a approximativement $\tan \varphi = L/R$ où R le rayon de giration. Or pour un mouvement circulaire uniforme la vitesse V vérifie : $V = R\omega$. On en déduit le modèle dit de *bicyclette* (valable pour les petits angles de braquage) :

$$\omega = \frac{V}{L} \tan \varphi$$

Le véhicule considéré est commandé en vitesse (V) et en angle de braquage (φ). On pourra négliger la dynamique des actionneurs et écrire :

$$\begin{cases} V &= V^* \\ \varphi &= \varphi^* \end{cases}$$

où V^* et φ^* sont les consignes (connues) de vitesse et de braquage. On peut aussi tenir compte de cette dynamique et écrire

$$\begin{cases} \dot{V} &= -\frac{1}{\tau_V} (V - V^*) \\ \dot{\varphi} &= -\frac{1}{\tau_\varphi} (\varphi - \varphi^*) \end{cases}$$

τ_V et τ_φ sont les constantes de temps des actionneurs.

La simulation calcule aussi les vitesses longitudinales de chacune des quatre roues du véhicule. Les expressions de ces vitesses ont été reportées en fin de document.

Les différentes mesures qui sont disponibles pour réaliser la navigation sont :

- Les consignes de conduite du véhicule V^* et φ^* . Elles sont parfaitement connues et échantillonnées à 100 Hz .
- La vitesse de rotation ω . Elle est mesurée par une unité de mesure inertielle (IMU) à la fréquence de 100 Hz . Cette mesure est affectée d'un bruit, d'un biais et d'une légère dérive.
- La position (x,y) du véhicule. Elle est mesurée par un récepteur GPS à la fréquence de 1 Hz . Cette mesure est affectée d'un bruit. Elle présente aussi un biais, mais de manière intermittente.
- Les vitesses de chacune des quatre roues. Elles sont mesurées par des odomètres à la fréquence de 100 Hz . Ces mesures sont quantifiées et bruitées.

L'outil de simulation qui vous est fourni vous permet de faire quelques tests afin évaluer le comportement de ces erreurs :

- macro Matlab *BE_script.m*,
- et schéma Simulink *schemaSIM.slx*.

Les paramètres spécifiant le comportement de la simulation sont accessibles à travers différentes macros :

- *modele_VEHICULE* pour le comportement du véhicule,
- *modele_TRAJECTOIRE* pour la trajectoire suivie,
- *modele_ODOMETRE* pour les odomètres,
- etc...

Vous pouvez développer facilement de nouveaux scénarios de déplacement du véhicule dans la macro *modele_TRAJECTOIRE.m*, ou modifier sa réponse aux consignes dans la macro *modele_VEHICULE.m*.

Vous pouvez aussi modifier les caractéristiques des capteurs (dans les macros *modele_IMU*, etc...) pour simplifier le problème de navigation.

2 Travail demandé

Le travail qui vous est demandé consiste à développer et tester un filtre de Kalman réalisant la navigation, c'est-à-dire permettant d'estimer la position (x,y) et l'orientation (θ) de l'engin. Les estimations doivent être fournies à la fréquence de 100 Hz . Pour cela il vous faut :

- Ecrire un modèle de synthèse.
- Développer le filtre. Les paramètres et les initialisations seront codées dans la macro *model_ESTIMATEUR.m*. L'algorithme sera codé dans la macro *ESTIMATEUR* du schéma Simulink.
- Régler le filtre et analyser ses performances.

Il y a de nombreuses solutions. On peut utiliser les consignes en V et φ , ou ne pas les utiliser. On peut utiliser la mesure de ω ou non. De même pour les mesures odométriques. Faites des choix pour avoir un modèle de synthèse pas trop complexe, et pour que toutes vos variables d'état soient observables avec les mesures choisies.

La section suivante vous présente quelques propositions de modélisation pour la synthèse du filtre de navigation. Choisissez-en une simple pour commencer. Développer le filtre en faisant attention aux différents pas d'échantillonnage. Paramétrer correctement les matrices Q et R , la condition initiale sur P . Tester sur le scénario proposé, et aussi sur des scénarios simplifiés (pas de biais sur ω_{IMU} , pas de sauts sur le GPS). Passer ensuite à un filtre plus sophistiqué pour obtenir un navigateur qui fonctionne malgré toutes les imperfections des capteurs.

3 Quelques propositions de modèle de synthèse

1. Modèle en $X = [x, y, \theta]$, $U = [V^*, \omega_{IMU}]$, $Z = [x_{GPS}, y_{GPS}]$.

$$\begin{cases} \dot{x} &= V^* \cos \theta + v_x \\ \dot{y} &= V^* \sin \theta + v_y \\ \dot{\theta} &= \omega_{IMU} + v_\theta \\ x_{GPS} &= x + w_x \\ y_{GPS} &= y + w_y \end{cases}$$

où v et w sont les incertitudes assimilées à des bruits de covariances Q et R .

2. Prise en compte du biais gyrométrique.

Il faut rajouter un état b pour représenter ce biais. Les équations en $\dot{\theta}$ et \dot{b} s'écrivent :

$$\begin{cases} \dot{\theta} &= \omega_{IMU} - b + v_\theta \\ \dot{b} &= v_b \end{cases}$$

3. Modèle en $X = [x, y, \theta, V]$, $U = [V^*, \omega_{IMU}]$, $Z = [x_{GPS}, y_{GPS}]$.

$$\begin{cases} \dot{x} &= V \cos \theta \\ \dot{y} &= V \sin \theta \\ \dot{\theta} &= \omega_{IMU} + v_\theta \\ \dot{V} &= -\frac{1}{\tau_V} (V - V^*) \\ x_{GPS} &= x + w_x \\ y_{GPS} &= y + w_y \end{cases}$$

Il y a un état de plus, mais les deux premières équations d'état ne sont plus incertaines.

4. Utiliser les odomètres pour faciliter l'estimation de V .

On vérifie facilement (voir l'annexe) que $(V_{ar. g} + V_{ar. d})/2 = V$ on peut utiliser cette moyenne comme une mesure V_{ODO} de V . Cette mesure de V peut aider au fonctionnement du filtre construit sur le modèle précédent. On ajoute la mesure :

$$V_{ODO} = V + w_V$$

5. Utiliser les odomètres pour estimer ω .

On montre aussi que la demi-différence des vitesses de l'essieu arrière est une mesure de ω : $(V_{ar. d} - V_{ar. g})/2 = H \omega$. Cette mesure ω_{ODO} de ω n'est pas biaisée comme ω_{IMU} et peut donc aider à estimer le biais gyrométrique :

$$\begin{cases} \dot{x} &= V \cos \theta \\ \dot{y} &= V \sin \theta \\ \dot{\theta} &= \omega_{IMU} - b + v_\theta \\ \dot{b} &= v_b \\ x_{GPS} &= x + w_x \\ y_{GPS} &= y + w_y \\ \omega_{ODO} &= \omega_{IMU} - b + w_b \end{cases}$$

6. Vous pouvez aussi proposer et tester d'autres modèles.

4 Annexe :

4.1 Vitesses des roues

Les vitesses longitudinales des roues sont :

$$\begin{aligned} V_{arr.gauche} &= V_x \cos \theta + V_y \sin \theta - \omega H \\ V_{arr.droite} &= V_x \cos \theta + V_y \sin \theta + \omega H \\ V_{av.gauche} &= V_x \cos (\theta + \varphi) + V_y \sin (\theta + \varphi) - \omega (H \cos \varphi - L \sin \varphi) \\ V_{av.droite} &= V_x \cos (\theta + \varphi) + V_y \sin (\theta + \varphi) + \omega (H \cos \varphi + L \sin \varphi) \end{aligned}$$

où H est la demi-longueur d'un essieu. Ces relations s'écrivent encore :

$$\begin{aligned} V_{arr.gauche} &= V - \omega H \\ V_{arr.droite} &= V + \omega H \\ V_{av.gauche} &= (V - \omega H) \cos \varphi + \omega L \sin \varphi \\ V_{av.droite} &= (V + \omega H) \cos \varphi + \omega L \sin \varphi \end{aligned}$$

4.2 Modèle bicyclette

Ce modèle relie (très approximativement) le rayon de giration du véhicule R et l'angle de braque des roues φ , connaissant la distance inter-essieux (L) :

$$\tan \varphi \approx \frac{L}{R}$$

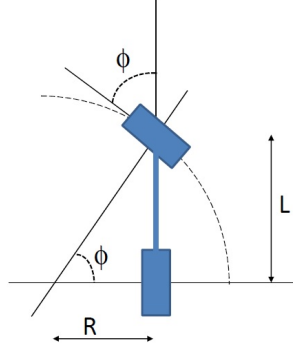


Figure 1: Modèle bicyclette

Comme pour un mouvement de rotation uniforme la cinématique de rotation s'écrit : $V = R\omega$, on en déduit que :

$$\omega \approx \frac{V}{L} \tan \varphi$$

4.3 Erreur GPS

Cette erreur est composée d'un bruit blanc centré (dont la ddp est supposée gaussienne), et de sauts. On peut penser que ces sauts correspondent aux modifications dans la constellation utilisée pour calculer la position. Nous supposons que l'occurrence des sauts est poissonnienne et que leur amplitude est uniforme.

Si T désigne l'intervalle entre deux sauts, T suit la loi exponentielle $f_T(\tau) = \lambda e^{-\lambda\tau}$ où λ est le paramètre de Poisson. On en déduit que : $E(T) = 1/\lambda$ et $F_T(\tau) = 1 - e^{-\lambda\tau}$ (où E désigne l'espérance et F_T la fonction de répartition).

Pour simuler les sauts, on choisira deux paramètres : T_m la durée moyenne entre sauts, et \bar{S} , l'amplitude maximale des sauts.

Quand un saut survient (à une date t), on tire aléatoirement son amplitude S selon la loi uniforme sur $[-\bar{S}, +\bar{S}]$. Pour cela, on prend $u = rand$ (i.e. aléatoire uniforme sur $[0, 1]$), et $s = \bar{s}(2u - 1)$ (i.e. solution de $F_S(s) = u$).

On calcule la date $t + \tau$ du saut suivant en effectuant un tirage aléatoire de la variable aléatoire T selon la loi exponentielle définie ci-dessus avec $\lambda = 1/T_m$. Pour cela, on prend $u = rand$ (i.e. aléatoire uniforme sur $[0, 1]$), et $\tau = -T_m \log(1 - u)$ (i.e. solution de $F_T(\tau) = u$).

4.4 Simulation de variables aléatoires

On effectue des tirages aléatoire x d'une loi X de fonction de répartition $F_X(x)$ en résolvant : $F_X(x) = u$ où u a été tiré selon une loi uniforme sur $[0, 1]$.

Examinons le cas où X a une ddp $f_X(x)$ triangulaire centrée.
La densité est de la forme :

$$f_X(x) = \frac{1}{\overline{X}} \begin{cases} 1 + \frac{x}{\overline{X}} & \text{si } x \in [-\overline{X}, 0] \\ 1 - \frac{x}{\overline{X}} & \text{si } x \in [0, +\overline{X}] \\ 0 & \text{sinon} \end{cases}$$

si $[-\overline{X}, +\overline{X}]$ est le support. Sa fonction de répartition, obtenue en intégrant, est donc :

$$F_X(x) = \frac{1}{2} \begin{cases} (1 + \frac{x}{\overline{X}})^2 & \text{si } x \in [-\overline{X}, 0] \\ 2 - (1 - \frac{x}{\overline{X}})^2 & \text{si } x \in [0, +\overline{X}] \\ 0 & \text{sinon} \end{cases}$$

On prendra donc :

$$x = \overline{X} \begin{cases} \sqrt{2u} - 1 & \text{si } u \in [0, 0.5] \\ 1 - \sqrt{2(1-u)} & \text{si } u \in [0.5, 1] \end{cases}$$

4.5 Simulation d'actionneurs saturés

On suppose que les actionneurs répondent comme des systèmes du second ordre (figure 2). On prend $\omega_0 = 1/\tau$ où τ est le temps de réponse, et $\xi = 0.75$. La représentation d'état choisie fait apparaître en sortie des intégrateurs la grandeur à réguler (V sur le schéma) et sa dérivée.

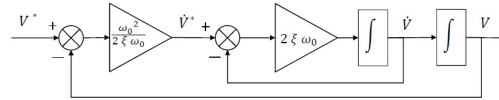


Figure 2: Modèle du second Ordre

Il est alors assez simple d'introduire les contraintes de saturation de ces grandeurs en remplaçant les intégrations par des intégrations saturées (figure 3) à l'aide d'un gain d'anti-windup.

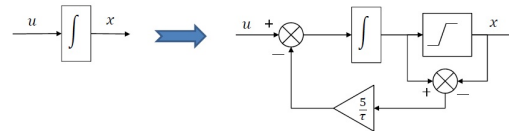


Figure 3: Intégrateur saturé