

ELE6209A - Systèmes de navigation

Département d'ingénierie électrique
École Polytechnique de Montréal

Été 2022

Instructor: Jérôme Le Ny

Answer the questions on the exam book.
Always clearly and concisely explain the reasoning leading to your answers. Numerical answers without explanations can be counted as wrong, even if the result is correct.
Your answers can be written in English or French.
You do not have to answer the problems in the order provided. However, you must group your answers to the questions of the same problem together.
If relevant, the units in which your numerical answers are expressed must be stated.
Manage your time carefully: the points attributed to each question should give you an indication on how much time to spend on them. Not all questions are of the same difficulty. I recommend you first skim through the exam to identify the problems you are most comfortable with.

Documents allowed: instructor's lecture notes and slides.

A non-programmable calculator is allowed.

Please return the question sheet at the end.

SOLUTION

Question:	1	2	3	4	Total
Points:	3	4	9	4	20
Score:					

- This page intentionally left blank -

X	Y	Distance	Temps
---	---	----------	-------

Table 1: Colonnes du fichier CSV

1. (a) [2 points] Sachant que la position $[n(t_k), e(t_k)]$ et l'angle $\psi(t_k)$ sont connu à l'instant t_k . En se basant sur les équations du chapitre 9.1.2, trouver les équations permettant de calculer $n(t_{k+1})$, $e(t_{k+1})$ et $\psi(t_{k+1})$.
- (b) [1 point] À l'aide du fichier *Q1.ttt* modifier la fonction `updateOdometry(q, ur, ul)`. Celle-ci s'occupe de mettre à jours le déplacement du robot dans la simulation. Trois variables sont en entrée.
 - q : position estimée du robot $[n, e, \psi]$ dans le repère $\{t\}$
 - u_r : Vitesse de rotation de la roue droite
 - u_l : Vitesse de rotation de la roue gauche

Montrer le graphique de l'erreur entre la trajectoire désirée (ligne verte) et la trajectoire obtenue avec l'odométrie (ligne bleue) pour une simulation de 50 secondes.

Solution:

- (a) Vitesses linéaires et angulaires.

$$u = \frac{1}{2}(u_l + u_r)$$

$$w = \frac{1}{L}(u_l - u_r)$$

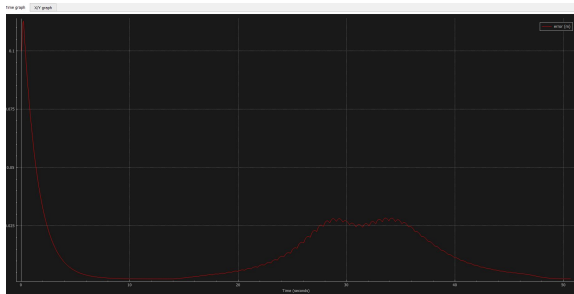
Déplacements linéaires et angulaires.

$$n(t_{k+1}) = n(t_k) + u\Delta t \cos(\psi(t_k))$$

$$e(t_{k+1}) = e(t_k) + u\Delta t \sin(\psi(t_k))$$

$$\psi(t_{k+1}) = \psi(t_k) + w\Delta t$$

- (b) Graphique représentant l'erreur entre la trajectoire désirée et l'odométrie.



2. (a) [3 points] À l'aide du fichier *laser_data.csv* et en se basant sur la théorie du chapitre 5, trouver la position $[x, y]$ de l'objet se déplaçant sur une période de 60 secondes. Le choix du langage de programmation est libre. Utiliser $x_0 = [-1, -1]$ comme premier point de recherche. Le fichier csv est divisé selon les colonnes suivantes:
 - X correspond aux positions x des points repères dans $\{t\}$
 - Y correspond aux positions y des points repères dans $\{t\}$
 - Distance correspond aux distances entre les points repères et l'objet dans $\{t\}$
 - Temps correspond au moment où le point repère est détecté par le laser

Montrer le graphique représentant le déplacement $[x, y]$ de l'objet.

- (b) [1 point] Implémenter l'algorithme développé en (a). Utiliser le fichier de démarrage *Q2.ttt*. La fonction *findPosition* doit être modifiée. Celle-ci utilise les données provenant du laser. Elle s'occupe du *positionfixing*. Trois variables sont en entrée.

- LMpositions : tableau contenant la position de tous les objets détectés par le laser
- LMdistance : tableau contenant la distance avec tous les objets détectés par le laser
- q : position estimée du robot dans le repère {t}

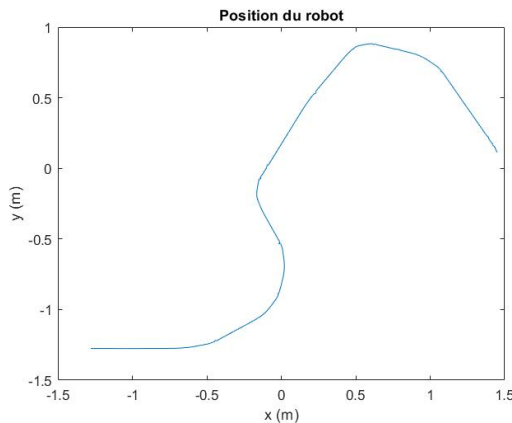
La notion de matrice pseudo-inversible devra être utilisée. Montrer les deux graphiques suivant pour une simulation de 60 secondes:

- Graphique représentant le déplacement $[x, y]$ du robot
- Graphique d'erreurs entre la position calculée et réelle du robot

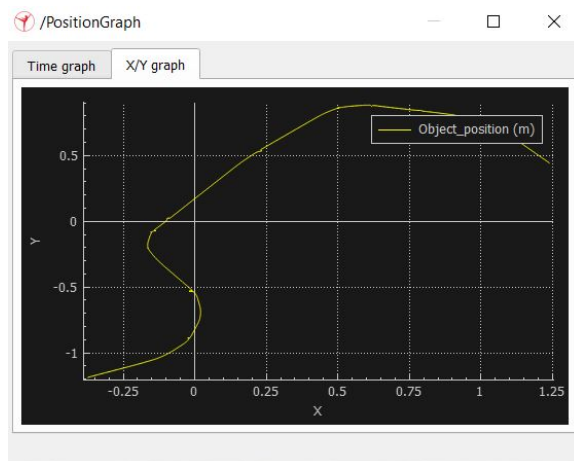
De plus, donner l'erreur moyenne entre la position calculée et réelle pour la même période.

Solution:

- (a) Graphique représentant la position $[x, y]$ de l'objet.



- (b) Graphique représentant la position $[x, y]$ du robot.



L'erreur moyenne est de 0.00083 pour 60 une période de secondes.

Graphique représentant l'erreur entre la position calculée et réelle du robot.



3. Réaliser et comparer 2 filtres de kalman étendu (EKF). Le filtre est utilisé afin de corriger la position estimée avec l'odométrie en utilisant les données lasers.

Les équations d'odométries sont utilisées pour obtenir les équations de navigation:

$$\dot{n} = \frac{1}{2} \cos \psi (R\dot{\theta}_l + w_l + R\dot{\theta}_r + w_r) \quad (1)$$

$$\dot{e} = \frac{1}{2} \sin \psi (R\dot{\theta}_l + w_l + R\dot{\theta}_r + w_r) \quad (2)$$

$$\dot{\psi} = \frac{1}{L} (R\dot{\theta}_r + w_r - R\dot{\theta}_l - w_l) \quad (3)$$

Ici $\dot{\theta}$ correspond à la vitesse de rotation de chaque roue obtenue par odométrie. $w \sim \mathcal{N}(0, 1 \times 10^{-2})$, $w_r \perp w_l$.

En linéarisant ($\Delta x = x - \hat{x}$) les équations ci-dessus, on obtient les matrices A et B:

$$A = \begin{bmatrix} 0 & 0 & -\frac{R}{2} \sin \hat{\psi} (\hat{\theta}_l + \hat{\theta}_r) \\ 0 & 0 & \frac{R}{2} \cos \hat{\psi} (\hat{\theta}_l + \hat{\theta}_r) \\ 0 & 0 & 0 \end{bmatrix} \quad (4)$$

$$B = \begin{bmatrix} \frac{1}{2} \cos \hat{\psi} & \frac{1}{2} \cos \hat{\psi} \\ \frac{1}{2} \sin \hat{\psi} & \frac{1}{2} \sin \hat{\psi} \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \quad (5)$$

$$R = 0.0975$$

$$L = 0.331$$

- (a) [1 point] Selon le modèle linéaire suivant :

$$\Delta \dot{x} = A(t)\Delta x + B(t)w(t) \quad (6)$$

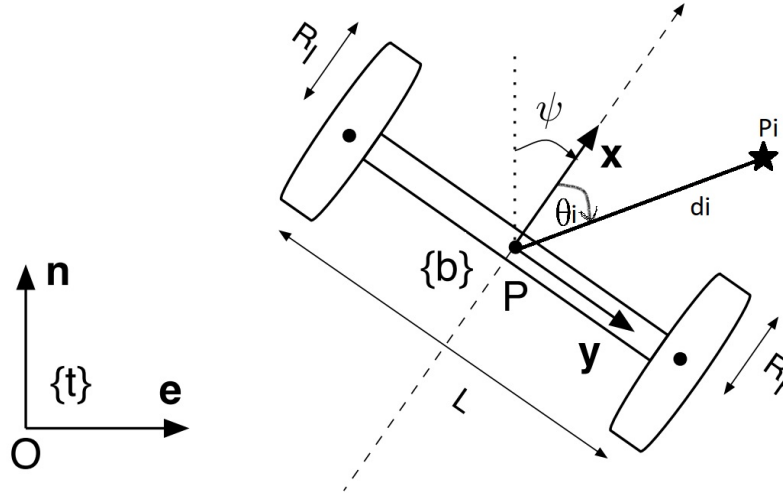
$$\Delta y_k = C\Delta x_k + D\nu(t) \quad (7)$$

Trouver les matrices C et D sachant que les mesures (y_k) correspondent à la position calculée avec un laser. On suppose ici que le laser donne directement la position du robot dans le repère $\{t\}$.

$$\tilde{n} = n + \nu_n$$

$$\tilde{e} = e + \nu_e$$

- (b) [2 points] Selon le même modèle linéaire que la question (a), trouver les matrices C et D sachant que les mesures (y_k) correspondent à l'angle θ_i et la distance d_i dans le repère $\{b\}$. Voir schéma suivant :



Ici,

$$\begin{aligned}\tilde{\theta}_i &= \theta_i \\ \tilde{d}_i &= d_i + \nu_i\end{aligned}$$

- (c) [4 points] Réaliser, dans le langage de votre choix, deux EKF utilisant les matrices trouvées en (a) et (b). Utiliser les deux fichiers de départ. Le fichier *Laser1* contient les documents suivants:

- *estimated_odometry_position.csv*
- *laser_data.csv*
- *odometry_data.csv*
- *true_pose.csv*

laser_data.csv contient la position x et y du robot exprimé dans le repère $\{t\}$ à des instants précis. Le laser utilisé est celui de la question (a). Le bruit sur les mesures laser est $\nu \sim \mathcal{N}(0, 5 \times 10^{-2})$. $\nu_n \perp \nu_e$.

odometry_data.csv contient les valeurs u_l et u_r . Ici $u_i = R\dot{\theta}_i + w_i$, $i = l, r$

estimated_odometry_position.csv contient les mesures x, y et ψ calculé à l'aide des équations trouvées au problème 1.

true_pose.csv contient la position (x, y) ainsi que l'angle ψ réelle du robot.

Le fichier *Laser2* contient les documents suivants:

- *estimated_odometry_position.csv*
- *laser_data.csv*
- *lookupTable.csv*
- *odometry_data.csv*
- *true_pose.csv*

Les documents, *estimated_odometry_position.csv*, *odometry_data.csv* et *true_pose.csv* sont identiques à ceux du fichier *Laser1*.

laser_data.csv contient la distance d_i et l'angle θ_i dans le repère $\{t\}$ à des instants précis. Le laser utilisé est celui de la question (b). Le bruit sur les mesures laser d_i est $\nu \sim \mathcal{N}(0, 5 \times 10^{-2})$. Le document contient également l'identifiant du point P_i détecté.

lookupTable.csv permet de trouver la position x et y du point P_i^t à l'aide d'un identifiant.

Initialement, le robot se trouve à $\begin{bmatrix} n \\ e \\ \psi \end{bmatrix} = \begin{bmatrix} -1.628 \\ 0 \\ 0 \end{bmatrix}$.

Fournir les erreurs moyennes entre l'état $[n, e]$ calculée et réelle à l'aide des EKF. Comparer les résultats des deux EKF et discuter.

- (d) [2 points] Réaliser le filtre de votre choix sur Coppeliasim. Utiliser le fichier de départ adéquat (*Q3_laser1.ttt* ou *Q3_laser2.ttt*). Modifier la fonction `kalmanFilter`. Trois variables sont en entrée.
- `q` : position estimée du robot dans le repère $\{t\}$
 - `odometryMeasurements` : données provenant des encodeurs des roues droites et gauches
 - `positionFixMeasurements` : position du robot provenant du laser

Montrer l'erreur moyenne ainsi que le graphique de l'erreur entre la position calculée (corrigée à l'aide de l'EKF) du robot vs la position réelle du robot pour une simulation de 90 secondes.

Solution:

- (a) Le laser mesure la position $[n, e]$ du robot

$$\tilde{n} = n + \nu_n \quad (8)$$

$$\tilde{e} = e + \nu_e \quad (9)$$

Ici $\Delta x = x - \hat{x}$:

$$\Delta y_k = \begin{bmatrix} \tilde{n}_k - n_k \\ \tilde{e}_k - e_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta n_k \\ \Delta e_k \\ \Delta \psi_k \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu_{nk} \\ \nu_{ek} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (10)$$

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (11)$$

- (b) Le laser mesure la distance θ_i et la distance d_i du point P_i dans le repère $\{b\}$. Solution pour $y_k = P_i^t$:

$$\tilde{\theta}_i = \theta_i$$

$$\tilde{d}_i = d_i + \nu_i$$

$$\tilde{P}_i^b = \begin{bmatrix} \tilde{d}_i \cos(\tilde{\theta}_i) \\ \tilde{d}_i \sin(\tilde{\theta}_i) \end{bmatrix} \quad (12)$$

$$y_k = \hat{P}_i^t = \hat{P}_b^t + \hat{R}_b^t \tilde{P}_i^b = \begin{bmatrix} \hat{n}_b + \cos(\hat{\psi}) \tilde{d}_i \cos(\tilde{\theta}_i) - \sin(\hat{\psi}) \tilde{d}_i \sin(\tilde{\theta}_i) \\ \hat{e}_b + \sin(\hat{\psi}) \tilde{d}_i \cos(\tilde{\theta}_i) + \cos(\hat{\psi}) \tilde{d}_i \sin(\tilde{\theta}_i) \end{bmatrix} \quad (13)$$

$$\Delta y_k = \begin{bmatrix} 1 & 0 & -\sin(\hat{\psi})\tilde{d}_i\cos(\tilde{\theta}_i) - \cos(\hat{\psi})\tilde{d}_i\sin(\tilde{\theta}_i) \\ 0 & 1 & \cos(\hat{\psi})\tilde{d}_i\cos(\tilde{\theta}_i) - \sin(\hat{\psi})\tilde{d}_i\sin(\tilde{\theta}_i) \end{bmatrix} \begin{bmatrix} \Delta n_k \\ \Delta e_k \\ \Delta \psi_k \end{bmatrix} + \begin{bmatrix} \cos(\hat{\psi})\cos(\tilde{\theta}_i) - \sin(\hat{\psi})\sin(\tilde{\theta}_i) \\ \sin(\hat{\psi})\cos(\tilde{\theta}_i) + \cos(\hat{\psi})\sin(\tilde{\theta}_i) \end{bmatrix} \nu$$

$$C = \begin{bmatrix} 1 & 0 & -\sin(\hat{\psi})\tilde{d}_i\cos(\tilde{\theta}_i) - \cos(\hat{\psi})\tilde{d}_i\sin(\tilde{\theta}_i) \\ 0 & 1 & \cos(\hat{\psi})\tilde{d}_i\cos(\tilde{\theta}_i) - \sin(\hat{\psi})\tilde{d}_i\sin(\tilde{\theta}_i) \end{bmatrix} \quad (14)$$

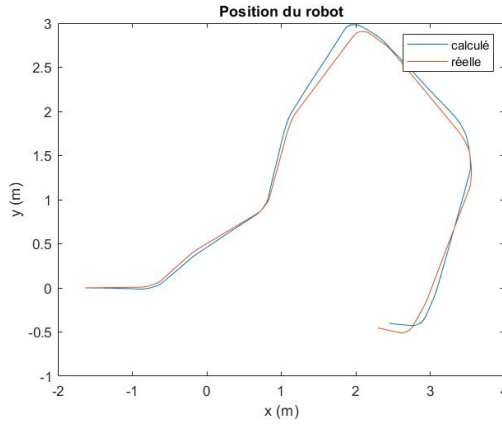
$$D = \begin{bmatrix} \cos(\hat{\psi})\cos(\tilde{\theta}_i) - \sin(\hat{\psi})\sin(\tilde{\theta}_i) \\ \sin(\hat{\psi})\cos(\tilde{\theta}_i) + \cos(\hat{\psi})\sin(\tilde{\theta}_i) \end{bmatrix} \quad (15)$$

(c) Pour l'EKF (*Laser1*), les erreurs moyennes sont :

$$n = -0.0106$$

$$e = 0.0602$$

Graphique représentant l'erreur entre la position calculée et réelle du robot:

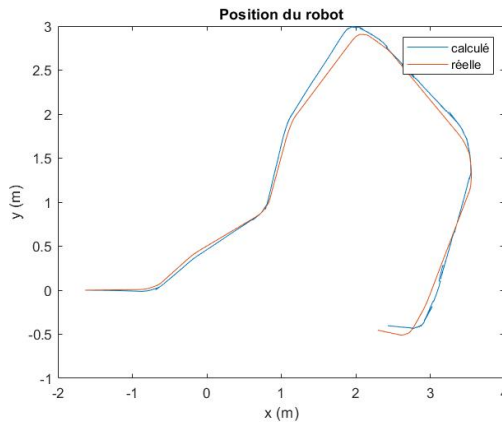


Pour l'EKF (*Laser2*), les erreurs moyennes sont :

$$n = -0.0050$$

$$e = 0.0539$$

Graphique représentant l'erreur entre la position calculée et réelle du robot:



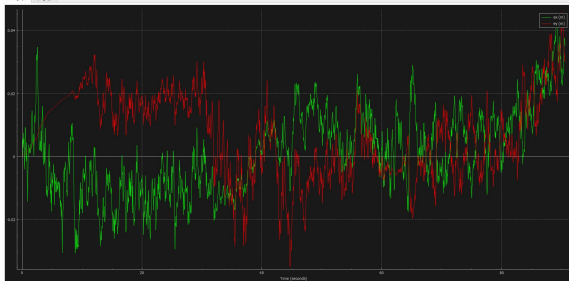
On remarque que les mesures de l'EKF (*Laser1*) ne modifient pas l'angle ψ . Pour cette raison les erreurs de n et e de l'EKF (*Laser2*) sont inférieures. Globalement, l'EKF (*Laser2*) semble meilleurs que l'EKF (*Laser1*).

(d) Pour l'EKF (*Laser1*), les erreurs moyennes pour une période de 120 secondes sont :

$$n = 0.00110$$

$$e = 0.00647$$

Graphique représentant l'évolution de l'erreur dans le temps (90 secondes):

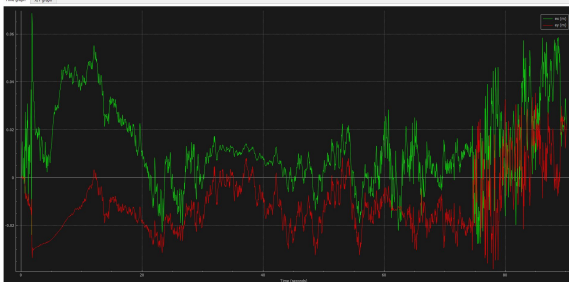


Pour l'EKF (*Laser2*), les erreurs moyennes pour une période de 120 secondes sont :

$$n = 0.01267$$

$$e = -0.01081$$

Graphique représentant l'évolution de l'erreur dans le temps (90 secondes):



4. [4 points] Voici le schéma du système:

Dans le fichier de départ *Q4.ttt* deux fonctions sont à modifier.

- $q = \text{updateOdometry}(q, u_r, u_l)$
- $kResult = \text{kalmanFilter}(q, \text{odometryMeasurements}, \text{positionFixMeasurements})$

La fonction *updateOdometry* s'occupe de mettre à jour le déplacement du robot dans la simulation. Cette fonction représente le module Cinématique du schéma 1. Trois variables sont en entrée.

- q : position estimée du robot dans le repère $\{t\}$
- u_r : Vitesse de rotation de la roue droite
- u_l : Vitesse de rotation de la roue gauche

La fonction *kalmanFilter* se charge de la fusion des données par l'intermédiaire de l'EKF. Cette fonction représente le module EKF du schéma 1. Trois variables sont en entrée.

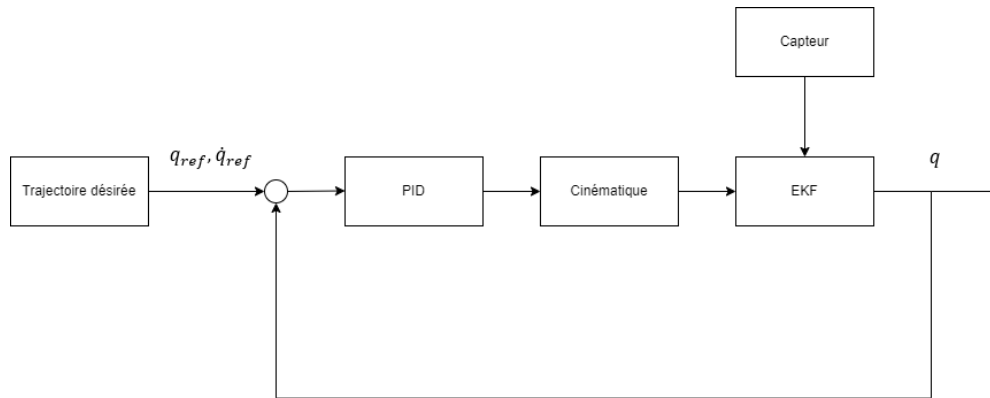


Figure 1: Schéma de controle

- q : position estimée du robot dans le repère $\{t\}$
- `odometryMeasurements` : données provenant des encodeurs des roues droites et gauches
- `positionFixMeasurements` : position du robot provenant du position fixing

Le laser est le même que celui utilisé en

Montrer l'erreur moyenne entre la position calculée (corrigée à l'aide de l'EKF) du robot versus la position réelle du robot pour une simulation de 240 secondes. Démontrer aussi, une capture d'écran du trajet effectué suite à une simulation de 240 secondes.

Solution: L'erreur moyenne est de 0.04214.

Capture d'écran représentant le parcours effectué:

