Etude des trajectoires d'un robot mobile

Profs Q. Louveaux, O. Brüls et F. Nguyen

Introduction aux méthodes numériques et projet
Bachelier en sciences de l'ingénieur, Bloc 1
Université de Liège
Année académique 2018-2019

Introduction

A l'heure actuelle, la robotique connait une évolution particulièrement rapide tant sur le plan des avancées technologiques (nouveaux capteurs et actionneurs, intelligence artificielle, nouveaux matériaux, digitalisation), que sur le plan des applications qui voient le jour dans des secteurs aussi diversifiés que la production industrielle, l'exploration spatiale, les services aux particuliers, les véhicules autonomes, la chirurgie ou encore les prothèses. Un robot est une machine capable d'exécuter des mouvements et des actions de manière autonome à partir d'instructions de haut niveau. Une question essentielle pour les ingénieurs est donc d'assurer une planification des trajectoires et un contrôle efficace des mouvements pour réaliser la tâche demandée. Dans ce projet, vous allez mettre en œuvre des méthodes numériques pour définir les trajectoires d'un robot mobile et analyser ses mouvements sous l'action du système de contrôle.

Description du modèle

On considère un robot illustré à la Figure 1, de géométrie circulaire, évoluant dans le plan horizontal xy et dont l'orientation est définie par le vecteur unitaire $\overrightarrow{t}$. Le robot est actionné par deux roues motrices de rayon r, séparées d'une distance $2\,e$ et disposées symétriquement par rapport au centre. Les roues motrices roulent sans glisser sur le sol et le centre de masse coïncide avec le centre du cercle.

Le mouvement du robot est décrit par les variables suivantes :

— les coordonnées x, y du centre du robot,

— l'angle θ entre le vecteur $\overrightarrow{t}$ et l'axe x,

— les angles de rotation de la roue droite ψ_R et de la roue gauche ψ_L, mesurés positivement autour de l'axe $\overrightarrow{n}$ perpendiculaire à $\overrightarrow{t}$,

— la vitesse linéaire du robot v mesurée dans la direction du vecteur $\overrightarrow{t}$,

— la vitesse angulaire du robot $\omega = d\theta/dt$.

Considérant l'orientation du vecteur $\overrightarrow{t}$, les vitesses selon les axes x, y sont données par

$$\frac{dx}{dt} = v\cos\theta \tag{1a}$$

$$\frac{dy}{dt} = v\sin\theta \tag{1b}$$

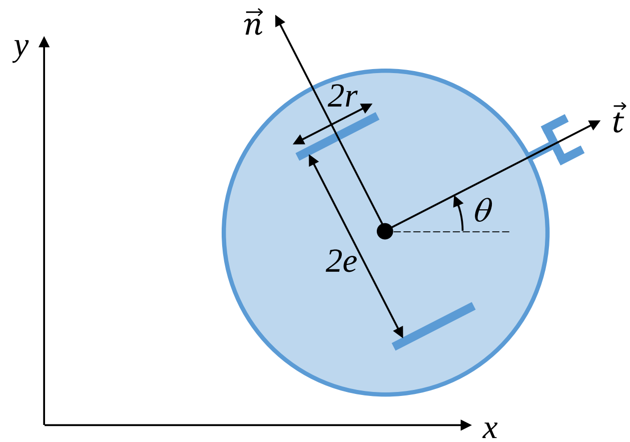

Figure 1 – Robot mobile évoluant dans le plan xy.

Les conditions de roulement sans glissement s'expriment par

$$v = \frac{r}{2}\left(\frac{d\psi_R}{dt} + \frac{d\psi_L}{dt}\right) \tag{2a}$$

$$\omega = \frac{r}{2\,e}\left(\frac{d\psi_R}{dt} - \frac{d\psi_L}{dt}\right) \tag{2b}$$

et peuvent se réécrire comme suit

$$\frac{d\psi_R}{dt} = \frac{1}{r}\,(v + e\,\omega) \tag{3a}$$

$$\frac{d\psi_L}{dt} = \frac{1}{r}\,(v - e\,\omega) \tag{3b}$$

Ensuite, les équations du mouvement sont obtenues à partir de la deuxième loi de Newton et du théorème du moment cinétique. En négligeant l'inertie des roues, on obtient :

$$m\frac{dv}{dt} = f_R + f_L \tag{4a}$$

$$J\,\frac{d\omega}{dt} = e\,(f_R - f_L) \tag{4b}$$

avec

- la masse totale du robot m,
- l'inertie du robot J,
- les forces motrices induites par les roues droite et gauche f_R et f_L.

On propose un modèle des forces motrices tenant compte de la contribution des couples moteurs et des phénomènes de frottement

$$f_R = \frac{\tau_R}{r} - c\,\frac{d\psi_R}{dt} - d\,\tanh\left(\frac{1}{\eta}\frac{d\psi_R}{dt}\right) \tag{5a}$$

$$f_L = \frac{\tau_L}{r} - c\,\frac{d\psi_L}{dt} - d\,\tanh\left(\frac{1}{\eta}\frac{d\psi_L}{dt}\right) \tag{5b}$$

avec

Table 1 – Valeur des paramètres.

Paramètre	Valeur	Unités
e	0.15	m
r	0.1	m
m	10	kg
J	1	kg·m^2
c	0.01	N·s
d	0.8	N
η	0.1	s^{-1}

— les couples moteurs τ_R et τ_L,

— les paramètres du modèle de frottement c (coefficient de frottement visqueux), d (coefficient de frottement sec) et η (paramètre de régularisation du modèle de frottement sec).

En résumé, le système est décrit par un ensemble de sept équations différentielles ordinaires :

$$
\begin{aligned}
d\psi_R/dt &= \frac{1}{r}(v + e\,\omega) && \text{(6a)}\\
d\psi_L/dt &= \frac{1}{r}(v - e\,\omega) && \text{(6b)}\\
dv/dt &= \frac{1}{m}(f_R + f_L) && \text{(6c)}\\
d\omega/dt &= \frac{e}{J}(f_R - f_L) && \text{(6d)}\\
d\theta/dt &= \omega && \text{(6e)}\\
dx/dt &= v\cos\theta && \text{(6f)}\\
dy/dt &= v\sin\theta && \text{(6g)}
\end{aligned}
$$

pour les sept inconnues $\psi_R(t)$, $\psi_L(t)$, $v(t)$, $\omega(t)$, $\theta(t)$, $x(t)$, $y(t)$. Les valeurs des paramètres du modèle sont données dans la Table 1.

Question 1 : Méthodes de la sécante et de la bissection

A la fin du projet, nous aurons besoin de résoudre une équation non-linéaire. La première question du projet consiste à implémenter deux méthodes de recherche de racine qui ont été vues au cours, à savoir la méthode de la bissection et la méthode de la sécante. Ces implémentations doivent être le plus générique possible et donc pouvoir fonctionner pour tout type de fonction et avec le moins d'hypothèses de départ possibles. On demande d'implémenter les deux fonctions

```
function x = secante(@f,x0,x1)
```

et

```
function x = bissection(@f,x0, x1)
```

Table 2 – Points de passage.

i	t_i (s)	x_i^d (m)	y_i^d (m)
1	0	0	0
2	1	1	-1
3	2	5	1
4	3	6	5
5	4	8	4

qui permettent de rechercher la racine d'une fonction MATLAB `f(x)` à partir de deux valeurs initiales `x0` et `x1` selon la méthode de la sécante et de la bissection.

Dans votre implémentation, prenez soin (i) de vérifier les hypothèses de chacune de ces méthodes et de prévoir un comportement adéquat du code dans le cas où ces hypothèses ne sont pas rencontrées, de manière à fournir une solution dans un maximum de cas, (ii) de veiller à réduire autant que possible le nombre d'appels de la fonction `f(x)` à l'intérieur des fonctions `secante` et `bissection`, et ce pour des raisons d'efficacité.

Ces deux fonctions seront testées lors du premier milestone.

Question 2 : Modélisation et simulation

Nous allons maintenant modéliser le robot en considérant que les couples moteurs sont constants dans le temps et fixés à $\tau_R = 1.1$ N·m et $\tau_L = 1$ N·m.

1. Implémenter une fonction `oderobotQ2.m` qui prend en entrée la valeur temps et un vecteur à 7 composantes (ψ_R, ψ_L, v, ω, θ, x et y) et qui renvoie en sortie la dérivée temporelle de ce vecteur en utilisant le système d'équations différentielles (6).
2. Implémenter la méthode d'Euler explicite pour résoudre le système d'équations différentielles sur l'intervalle de temps $[0, t_f]$ avec $t_f = 20$ s et les conditions initiales $\psi_R(0) = 0$, $\psi_L(0) = 0$, $v(0) = 0$, $\omega(0) = 0$, $\theta(0) = 0$, $x(0) = 0$, $y(0) = 0$. Représenter graphiquement la trajectoire du robot dans le plan xy et l'évolution de la vitesse linéaire v en fonction du temps.
3. Utiliser la fonction `ode45` (ou un autre solveur de MATLAB) pour résoudre le système d'équations différentielles dans les mêmes conditions qu'à la question précédente.
4. Analyser la solution obtenue sur un intervalle de temps élargi en redéfinissant la valeur $t_f = 10^4$ s. Discuter le choix du pas de temps de la méthode d'Euler explicite et le choix des tolérances du solveur `ode45`.

Question 3 : Planification de la trajectoire

Dans cette question, nous cherchons à planifier une séquence de mouvements que le robot devra exécuter. Plus précisément, nous allons planifier une trajectoire désirée $x^d(t)$ et $y^d(t)$ reliant plusieurs points de passage pré-définis dans le plan xy. Ensuite, nous évaluerons les mouvements à imposer au niveaux des roues $\psi_R^d(t)$ et $\psi_L^d(t)$ pour parcourir cette trajectoire.

Considérons un ensemble de N points de passage définis par les valeurs t_i, x_i^d, y_i^d avec $i = 1, \ldots, N$ tels que $t_1 = 0$ et $t_N = t_f$ (voir Table 2). La première partie de la question consiste à définir une trajectoire $x^d(t)$ et $y^d(t)$ et sa dérivée par des fonctions polynomiales par morceau.

1. Définir les fonctions $x^d(t)$ et $y^d(t)$ par interpolation polynomiale entre les points de passage. On demande en outre que ces fonctions vérifient des conditions de vitesse nulle en $t = 0$ et $t = t_f$.
2. Calculer les dérivées temporelles dx^d/dt et dy^d/dt. Ces dérivées seront également représentées par des fonctions polynomiales.
   Indice : soit $x(t)$ un polynôme d'ordre 3 représenté dans MATLAB par le vecteur `coefs = [a b c d]` et sa dérivée $dx(t)/dt$ représentée par le vecteur `dcoefs = [0 e f g]` ; le calcul de la dérivée conduit à une relation `dcoefs = coefs * A` où `A` est une matrice constante.

La deuxième partie de la question vise à calculer le mouvement des roues $\psi_R^d(t)$ et $\psi_L^d(t)$. Pour ce faire, nous allons calculer numériquement les valeurs de $\psi_R^d(t_i^*)$ et $\psi_L^d(t_i^*)$ sur une grille temporelle $t_1^*, \ldots, t_n^*$ et construire ensuite les fonctions $\psi_R^d(t)$ et $\psi_L^d(t)$ par interpolation polynomiale. L'intégration des équations (6a) et (6b) sur l'intervalle de temps $[t_{i-1}^*, t_i^*]$ donne

$$\psi_R^d(t_i^*) = \psi_R^d(t_{i-1}^*) + \frac{1}{r}\int_{t_{i-1}^*}^{t_i^*} v^d(t)\,dt + \frac{e}{r}\left(\theta^d(t_i^*) - \theta^d(t_{i-1}^*)\right) \tag{7a}$$

$$\psi_L^d(t_i^*) = \psi_L^d(t_{i-1}^*) + \frac{1}{r}\int_{t_{i-1}^*}^{t_i^*} v^d(t)\,dt - \frac{e}{r}\left(\theta^d(t_i^*) - \theta^d(t_{i-1}^*)\right) \tag{7b}$$

L'angle θ^d et la vitesse linéaire v^d peuvent être déterminés à partir de dx^d/dt et dy^d/dt en résolvant les équations (6f) et (6g). Si $dx^d/dt \neq 0$ ou $dy^d/dt \neq 0$, la solution est donnée par la fonction à deux arguments `atan2` disponible dans MATLAB :

$$\theta^d = \mathrm{atan2}\left(\frac{dy^d}{dt}, \frac{dx^d}{dt}\right) \tag{8a}$$

$$v^d = \frac{dx^d}{dt}\cos\theta^d + \frac{dy^d}{dt}\sin\theta^d \tag{8b}$$

Si $dx^d/dt = 0$ et $dy^d/dt = 0$, alors $v^d = 0$ et θ^d est indéterminé. Concrètement, on demande de déterminer $\psi_R^d(t)$ et $\psi_L^d(t)$ en suivant les étapes suivantes.

3. Etablir une fonction `fun_vd` qui reçoit en entrée la valeur temps, les fonctions dx^d/dt et dy^d/dt établies à la question 3.2 et qui calcule en sortie la variable v^d au temps considéré.
4. Définir les instants t_i^* selon une grille régulière sur l'intervalle $[0, t_f]$ avec $n = 101$. Calculer $\psi_R^d(t_i^*)$ et $\psi_L^d(t_i^*)$ pour $i = 2, \ldots, n$ à partir des formules (7) avec les valeurs initiales $\psi_R^d(t_1^*) = \psi_L^d(t_1^*) = 0$. L'intégration de $v^d(t)$ sur chaque sous-intervalle peut être réalisée numériquement, par exemple avec la fonction MATLAB `integral`.
5. Construire les fonctions $\psi_R^d(t)$ et $\psi_L^d(t)$ par interpolation polynomiale.

Question 4 : Suivi et optimisation de la trajectoire

Le premier objectif de la question 4 est de définir les couples moteurs de manière à suivre la trajectoire établie à la question 3, c'est-à-dire de manière à ce que le mouvement réel des roues $\psi_R(t)$ et $\psi_L(t)$ s'approche du mouvement souhaité $\psi_R^d(t)$ et $\psi_L^d(t)$. Nous considérons que les couples moteurs sont définis par :

$$\tau_R(t) = -k \left(\psi_R(t) - \psi_R^d(t)\right) \tag{9a}$$

$$\tau_L(t) = -k \left(\psi_L(t) - \psi_L^d(t)\right) \tag{9b}$$

où k est un paramètre du système de contrôle appelé "gain" et $\psi_R(t) - \psi_R^d(t)$ et $\psi_L(t) - \psi_L^d(t)$ sont les erreurs entre le mouvement réalisé par les roues du robot et le mouvement souhaité. Pour évaluer la précision du mouvement dans le plan xy, nous utilisons l'erreur finale sur la position E définie par

$$E = \sqrt{(x(t_f) - x^d(t_f))^2 + (y(t_f) - y^d(t_f))^2} \tag{10}$$

1. Etablir une nouvelle fonction `oderobotQ4` qui calcule la dérivée du vecteur comprenant les composantes (ψ_R, ψ_L, v, ω, θ, x et y) selon le système d'équations différentielles (6), avec les couples moteurs calculés selon l'équation (9).
2. Calculer le mouvement du robot par résolution de l'équation différentielle, en partant des conditions initiales $\psi_R(0) = 0$, $\psi_L(0) = 0$, $v(0) = 0$, $\omega(0) = 0$, $\theta(0) = \theta_0^d$, $x(0) = x^d(0)$, $y(0) = y^d(0)$ et en choisissant la valeur du gain $k = 100$.
3. Comparer graphiquement la trajectoire réelle et la trajectoire prescrite et déterminer l'erreur finale E.
4. Analyser l'influence du gain k.

La dernière partie de cette question vise à réduire le temps de parcours de la trajectoire, tout en assurant que l'erreur finale ne dépasse pas un certain seuil critique $E \leq E_{\max}$. Pour réduire le temps de parcours, on modifie les instants t_i associés aux points de passage $t_i^r = a\, t_i$, où le paramètre $a \leq 1$ représente le facteur de réduction.

4. Définir une fonction permettant de calculer l'erreur finale E en fonction de la variable a.
5. Déterminer la valeur de a telle que $E(a) = E_{\max}$ avec $E_{\max} = 0.1$ m et $k = 100$. Utiliser à cette fin la fonction `bissection` ou la fonction `secante` de la question 1.
6. Proposer une autre définition pour l'erreur E permettant d'évaluer la précision de la trajectoire sur l'ensemble de l'intervalle de temps $[0, t_f]$. Résoudre les questions 4.4 et 4.5 avec cette nouvelle définition.

Consignes générales

— Le travail comporte un code de calcul MATLAB et un rapport d'une longueur de 10 pages maximum.

— Le code doit être correct et écrit par vous (ce que nous vérifierons à la présentation orale).

— Le code doit être soigné et commenté.
— Le code doit utiliser au maximum les possibilités vectorielles de MATLAB.
— Pour toute fonction, nous sommes susceptibles de vous demander de montrer un *profile* MATLAB et de l'interpréter.

Critères d'évaluation

La note finale n_f sera définie par la moyenne pondérée

$$n_f = 0.2 \times n_m \, + \, 0.35 \times n_r \, + \, 0.45 \times n_o$$

où n_m est la note de l'évaluation continue (milestones), n_r est la note du rapport et du code MATLAB et n_o est la note de la présentation orale.

Cependant, **une note inférieure ou égale à 9/20 pour l'oral** implique que c'est cette note qui est octroyée à l'étudiant.

Évaluation continue (poids $0,2$)

Deux "milestones" permettront de vérifier votre état d'avancement en cours de projet. Lors du milestone 1 organisé le 1er mars, votre groupe devra effectuer une démonstration du programme développé pour la **Question 1**. Lors du milestone 2 organisé le vendredi 22 mars, votre groupe devra effectuer une démonstration du programme développé pour les **Questions 2 et 3**. Une cote sur 10 sera attribuée selon les critères suivants :

— le programme donne une réponse complète à la question : 10/10 ;
— le programme donne une réponse partielle à la question : entre 5 et 9/10 ;
— le programme est bien avancé mais il ne fonctionne pas : entre 3 et 4/10 ;
— le programme n'est pas bien avancé : entre 1 et 2/10.

Une absence non justifiée sera sanctionnée par une note individuelle de 0/10, indépendamment du résultat du groupe.

Rapport et code MATLAB (poids $0,35$)

Un fichier .zip par groupe comprenant un rapport au format PDF et accompagné des fichiers .m de votre programme doit être soumis via la plateforme eCampus au plus tard pour le mercredi 10 avril à 23h59. Le nom du fichier .zip et le nom du fichier .pdf doivent respecter le format suivant : "NumeroGroupe_NomA_NomB_NomC.xxx" (exemple : l'archive "27_Dupond_Beckers_Bastin.zip" doit inclure le fichier "27_Dupond_Beckers_Bastin.pdf").

— La longueur du rapport ne peut dépasser 10 pages et ne doit pas comporter d'introduction.
— Pour chaque question, les résultats obtenus doivent être illustrés.
— La justification des choix numériques est très importante. Pensez à expliquer les choix qui vous ont semblé cruciaux.
— La forme du rapport est prise en compte. Il est recommandé de suivre les règles de bonne pratique pour la réalisation d'un rapport scientifique qui feront l'objet d'une présentation le mercredi 27 mars. Le nombre de pages étant limité, il est inutile de répéter l'énoncé. Allez donc à l'essentiel.
— La qualité du code (efficacité et soin) est également considérée dans l'évaluation.

Présentation orale (poids $0,45$)

La présentation orale est **individuelle**. Vous devez faire une démonstration du programme de votre groupe et répondre à des questions supplémentaires. Les éléments suivants seront pris en considération :

- la maîtrise des éléments de base de MATLAB,
- la maîtrise du programme réalisé par votre groupe,
- les justifications et éclaircissements par rapport aux choix réalisés et aux résultats présentés dans le rapport,
- la maîtrise des notions théoriques vues au cours.

Deuxièmes sessions

Les groupes qui, en première session, ont obtenu pour le rapport et le code une note $n_r \geq 12/20$ sont, s'ils le souhaitent, dispensés de remettre un nouveau code et un nouveau rapport lors de la session de septembre. Dans ce cas, seul l'oral doit être représenté et compte pour 100% de la note finale. Pour les autres groupes, un nouveau code et un nouveau rapport doivent être remis 5 jours avant l'examen oral. La note finale est alors définie par la moyenne pondérée des notes de deuxième session

$$n_f = 0.35 \times n_r \, + \, 0.65 \times n_o$$

où n_r est la note du rapport et du code MATLAB et n_o est la note de la présentation orale.