

MU4RBI04 Robotique : TP 2– Commandes d'un robot mobile non-holonyme

...

Consignes pour le rapport

- Votre code doit être clair et bien commenté.
- Vous devez fournir un rapport format pdf contenant les réponses aux questions. Le nom de ce fichier sera `tp2.pdf`.
- Les fichiers du code Matlab à rendre devront être nommés `Ex1.m` `Ex2.m` et `Ex3.m`. Les fichiers doivent être mis dans un dossier (zip ou rar) nommé `vousre_nom` puis déposé sur Sakai dans *Devoirs Evalués*.

1 Objectif

L'objectif de ce TP est de simuler le déplacement et les actions associées de robot mobile à roues dans différentes situations opérationnelles, telles que

- le ralliement de points de passage (commande point à point),
- le suivi de trajectoire,
- le suivi de chemin,
- et le suivi d'un robot virtuel ou réel (optionnel).

Pour plus d'informations sur les commandes et les démonstrations de leur stabilité, vous pouvez vous référer à l'article Technique de l'Ingénieur mis à disposition sous Moodle.

Cette partie concerne le module *Exécution de trajectoire* dans un robot mobile qui reçoit du planificateur des trajectoires à suivre ou des points de passage et les traduit par des ordres de déplacement soit au niveau opérationnelle (plateforme) ou articulaire (moteurs des roues).

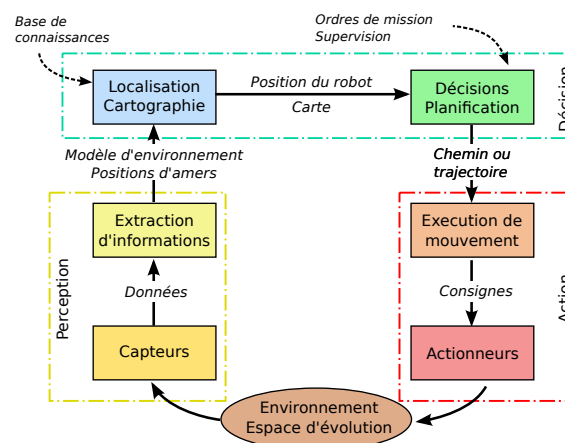


FIGURE 1 – Architecture classique de la boucle perception/décision/action

2 Modèle d'un robot unicycle

Le modèle cinématique direct d'un robot unicycle de largeur $2w$ et de rayon des roues r s'écrit :

$$\begin{cases} u &= \frac{r}{2}(\dot{\varphi}_1 + \dot{\varphi}_2) \\ v &= 0 \\ \omega &= \dot{\theta} = \frac{r}{2w}(\dot{\varphi}_1 - \dot{\varphi}_2) \end{cases} \quad (1)$$

et son inverse sous une forme matricielle :

$$\begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & w \\ 1 & -w \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \quad (2)$$

Le mouvement du robot exprimé en son centre R s'écrit

$$\begin{cases} \dot{x} &= u \cos \theta \\ \dot{y} &= u \sin \theta \\ \dot{\theta} &= \omega \end{cases} \quad (3)$$

Pour un point P attaché au robot distant de l_x du point R , sa vitesse est

$$\begin{cases} \dot{x}_P &= u \cos \theta - l_x \omega \sin \theta \\ \dot{y}_P &= u \sin \theta + l_x \omega \cos \theta \end{cases} \quad (4)$$

Le point P , contrairement au point R , est dit non-contraint, car sa vitesse a deux composantes indépendantes, et entièrement commandables par u, ω .

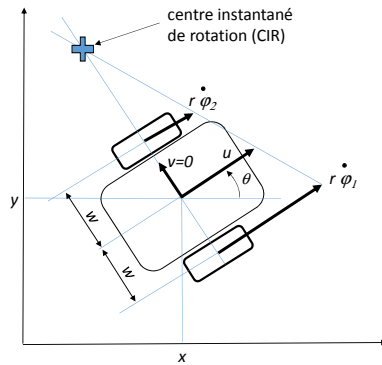


FIGURE 2 – Modèle cinématique d'un unicycle

3 Ralliement de points de passage

Il s'agit d'une approche de commande pratique permettant de réaliser diverses tâches de déplacement en procédant à leur discrétisation pour en déduire une série ordonnée de points de passage. Dans ce cas de figure, le mouvement résultant (réellement réalisé par le robot entre deux points) n'est pas contraint, seul comptant le passage par les différents points.

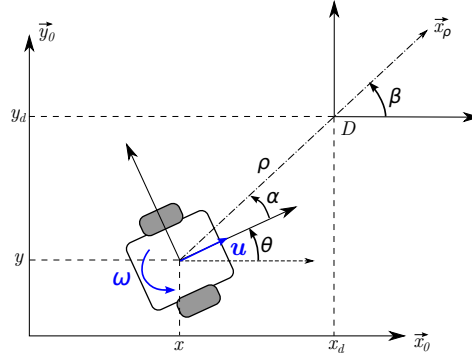


FIGURE 3 – Ralliement de points de passage

Le point de passage D est défini dans le même repère de référence par $\mathbf{p}_d = (x_d, y_d)$. La position relative du robot par rapport à ce point peut s'exprimer en coordonnées polaires (ρ, α) . Introduisons également β l'angle entre l'axe \mathbf{x}_ρ joignant la position du robot au point de passage et l'axe \mathbf{x} du repère de référence. Il s'agit d'une variable permettant de contrôler l'angle d'orientation du robot à son arrivée au point de passage :

$$\begin{cases} \rho &= \sqrt{\Delta x^2 + \Delta y^2} \\ \beta &= \text{atan2}(\Delta y, \Delta x) \\ \alpha &= \beta - \theta \end{cases} \quad (5)$$

où $\Delta x = x_d - x$ et $\Delta y = y_d - y$ représentent l'écart de position.

Considérons la loi de commande linéaire de la forme :

$$\begin{cases} u &= k_\rho \rho \\ \omega &= k_\alpha \alpha + k_\beta \tilde{\beta} \end{cases} \quad (6)$$

où $\tilde{\beta} = \beta_d - \beta$ désigne l'écart de position angulaire par rapport à une consigne d'orientation β_d au point D .

Il est possible de démontrer la stabilité asymptotique autour de ce point d'équilibre si :

$$k_\rho > 0, \quad k_\beta < 0, \quad k_\alpha > k_\rho$$

D'un point de vue pratique, le point à rallier D est considéré comme atteint lorsque la distance $|\rho|$ est inférieure à un certain seuil ρ_s , permettant alors de poursuivre le guidage vers le point de passage suivant.

4 Suivi de trajectoire

Pour ce type de commande, la trajectoire de référence est donnée dans le repère \mathcal{R}_0 , sous la forme $(x_r(t), y_r(t))$ (figure 4). Comme indiqué précédemment, cette trajectoire est supposée

être continûment différentiable. Soit P un point de référence sur le véhicule tel que $(l_x, l_y = 0)$ représente ses coordonnées dans le repère mobile attaché au robot et (x_p, y_p) ses coordonnées dans le repère fixe de référence. Ce point représente le point de contrôle du robot qui doit suivre la trajectoire de référence. L'objectif de la commande est alors de stabiliser autour de zéro l'erreur de suivi $\mathbf{e} = (x_r - x_p, y_r - y_p)$.

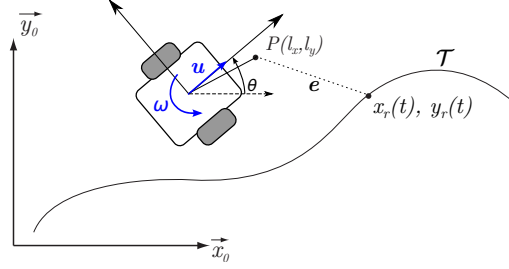


FIGURE 4 – Stabilisation de trajectoire

En effectuant le changement de variable suivant :

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & -l_x \sin \theta \\ \sin \theta & l_x \cos \theta \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \quad (7)$$

il est possible de trouver une loi de commande linéaire en ces nouvelles variables :

$$\begin{cases} v_1 &= \dot{x}_r + k_x(x_r - x_p) \\ v_2 &= \dot{y}_r + k_y(y_r - y_p) \end{cases} \quad (8)$$

qui donne une équation de la dynamique de l'erreur :

$$\dot{\mathbf{e}} = \begin{bmatrix} -k_x & 0 \\ 0 & -k_y \end{bmatrix} \mathbf{e}$$

La commande est stable si $k_x > 0$ et $k_y > 0$. De plus, elle est définie par la transformation (7) qui doit être inversible. Une condition d'existence est donc que $l_x \neq 0$.

5 Suivi de chemin

Considérant un chemin de référence noté $\mathcal{C}^* = (x_r(s), y_r(s))$ et le point de contrôle P fixe dans le repère du robot, il est possible d'exprimer les coordonnées de ce point de contrôle projeté sur la courbe \mathcal{C}^* . Soit S , la projection orthogonale de P sur la courbe, définie comme le point le plus proche appartenant à \mathcal{C}^* . On note \mathcal{R}_s , le repère de Frenet associé à ce point tel que \mathbf{y}_s soit normal à la courbe (figure 5). La configuration relative de robot par rapport à la courbe \mathcal{C}^* peut alors être définie par les variables suivantes :

- s l'abscisse curviligne du point S
- ξ la distance du point P à la courbe \mathcal{C}^*
- θ_e l'orientation relative du robot par rapport à \mathcal{R}_s

La commande de suivi de chemin consiste donc à stabiliser à zéro le paramètre de distance ξ . La variable permettant d'agir sur cette distance est la vitesse de rotation ω , en supposant que la vitesse d'avancement est fixée par ailleurs à une valeur non nulle et positive.

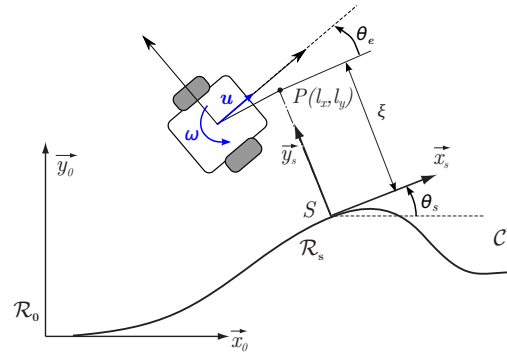


FIGURE 5 – Suivi de chemin

La condition $l_x \neq 0$ est nécessaire pour garantir la commandabilité de ξ par la vitesse de rotation ω . Une commande stabilisante peut alors être obtenue en considérant la loi de retour non-linéaire suivante :

$$\omega = -\frac{u}{l_x \cos \theta_e} \sin \theta_e - u k \xi \quad (9)$$

avec $k > 0$. En supposant $k > 0$ et $l_x u > 0$, on montre que $|\xi|$ est strictement décroissante et que ξ tend donc vers 0 si et seulement si $\theta_e \in]-\frac{\pi}{2}, \frac{\pi}{2}[$.

6 Travail demandé

Deux programmes matlab sont donnés. Le programme `Ex1.m` simule le premier scénario de ralliement de points de passage. Le second programme `trace-robot.m` affiche le robot.

1. Ralliement de points :

- Comparer les comportements observés en fonction des gains de la commande.
- Ajouter le calcul des vitesses des roues et les tracer sur exemple.
- Créer une série de points de passage dans lesquels des angles d'orientation désirés sont associées. Donner un ou deux exemples de parcours.
- Peut-on discrétiser un chemin par un ensemble de points de passage ? Si oui sous quelles conditions ?

2. Suivi de trajectoire :

- Choisir une trajectoire de $(x_r(t), y_r(t))$, définie par deux fonctions continues et différentiables de classe C_1 . Par exemple si vous voulez suivre l'axe (O, \mathbf{x}) à vitesse constante $u=1$ m/s, on aura $x_r(t) = 1 * t$ et $y_r(t) = 0$.
- Définir une position et orientation initiales du robot de telle sorte que l'erreur initiale ne soit pas nulle.
- Contrôleur : calculer l'erreur en position, puis les commandes v_1, v_2 , converties par la suite en u, ω .
- Robot : calculer le mouvement du robot.
- Afficher le robot et la trajectoire de référence (x_r, y_r) .
- Donner deux exemples avec des conditions initiales différentes en position et orientation.
- Comparer les comportements observés en fonction des gains de la commande.

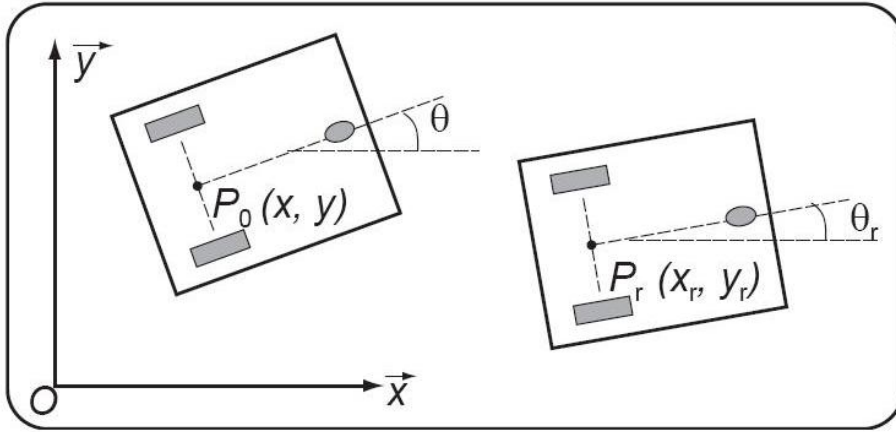
3. Suivi de chemin :

- Définir une droite d'équation cartésienne $ax+by+c=0$ comme chemin de référence, choisir une vitesse de parcours $u=1$.
- Contrôleur :
 - calculer l'erreur latérale signée ξ (distance entre le point et la droite suivant l'axe y de Frenet), et l'erreur angulaire (différence entre l'orientation courante et l'orientation de la droite de référence),
 - calculer la commande ω .
- Robot : calculer le mouvement du robot.
- Afficher le robot et le chemin de référence,
- Donner deux exemples avec des conditions initiales différentes en position et orientation.
- Comparer les comportements observés en fonction des gains de la commande.

7 Bonus : Suivi d'un robot réel ou virtuel

On cherche à contrôler la trajectoire $(x(t), y(t))$ du centre du robot (centre de l'essieu) ainsi que son orientation $\theta(t)$. La trajectoire de référence doit satisfaire une certaine condition de faisabilité (non-holonomie $-\dot{x} \sin \theta + \dot{y} \cos \theta = 0$) et ne peut pas être quelconque. Pour s'en assurer, l'idée ici est de dire que cette trajectoire résulte elle-même du mouvement du mouvement d'un robot de référence virtuel. Soit (x_r, y_r) la trajectoire de robot de référence, donc obéît

$$\begin{cases} \dot{x}_r = u_r \cos \theta_r \\ \dot{y}_r = u_r \sin \theta_r \\ \dot{\theta}_r = \omega_r \end{cases}$$



La loi de commande est définie par

$$\begin{cases} u = (w_1 + u_r) / \cos(\theta - \theta_r) \\ \omega = w_2 \cos^2(\theta - \theta_r) + \omega_r \end{cases} \quad (\theta - \theta_r) \neq \pm\pi/2$$

avec

$$\begin{cases} w_1 = -k_1 |u_r| (z_1 + z_2 z_3) \\ w_2 = -k_2 u_r z_2 - k_3 |u_r| z_3 \end{cases} \quad \begin{matrix} k_1 > 0 \\ k_2, k_3 > 0 \end{matrix}$$

et les 3 paramètres d'erreur exprimés dans le repère du véhicule de référence

$$\begin{cases} z_1 = e_{1,r} = (x - x_r) \cos(\theta_r) + (y - y_r) \sin(\theta_r) \\ z_2 = e_{2,r} = -(x - x_r) \sin(\theta_r) + (y - y_r) \cos(\theta_r) \\ z_3 = \tan e_{3,r} = \tan(\theta - \theta_r) \end{cases}$$

stabilise asymptotiquement l'erreur $e = (x - x_r, y - y_r, \theta - \theta_r)$ à zéro.

Une fois calculées la vitesse longitudinale u et la vitesse angulaire ω , on peut donc déduire les vitesses des roues φ_1 et φ_2 grâce à l'équation (2) .

Cette commande présentée ici peut être utilisée au cas d'une poursuite par le robot d'un autre véhicule réel qui est cinématiquement identique.

1. Expérimenter les conditions suivantes $x(t = 0) = 0, y(t = 0) = 2000, \theta(t = 0) = 0, x_r(t = 0) = 0, y_r(t = 0) = 0, \theta_r(t = 0) = 0$ et $u_r(t) = 100, \omega_r(t) = 0, k_1 = k_2 = 10^{-4}, k_3 = 0.1$.
2. Afficher avec des couleurs différentes le robot ainsi que le véhicule de référence.
3. Comparer les comportements observés en fonction des gains de la commande.
4. Faire un suivi d'un cercle avec et sans décalage initiale.