COMMANDE RÉFÉRENCÉE VISION D'UN ROBOT MOBILE EN ENVIRONNEMENT ENCOMBRÉ

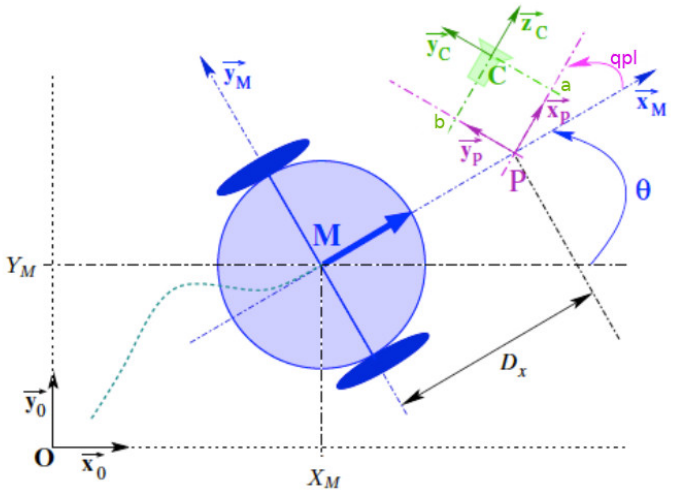
1 Introduction

Le but de cette séance de TP est de réaliser une tâche de navigation référencée vision permettant à un robot mobile de se positionner face à un motif visuel prédéfini malgré la présence d'obstacles imprévus. Pour cela, on utilisera l'asservissement visuel 2D classique et le formalisme des tâches redondantes.

1.1 Modélisation du système robotique

Le robot considéré est un robot différentiel représenté sur la figure 1(a). Il est équipé d'une caméra montée sur une platine commandable en lacet ainsi que d'un laser permettant de détecter les obstacles à proximité.

(a) Le robot Robotino.



(b) Visualisation des différents repères.

FIGURE 1 – Robot et repères considérés.

La figure 1(b) représente une vue de dessus du robot. On introduit les repères suivants :

- $\mathcal{R}(O,\ \vec{x},\ \vec{y},\ \vec{z})$ est le repère lié à la scène,
- $\mathcal{R}_M(M,\ \vec{x}_M,\ \vec{y}_M,\ \vec{z}_M)$ est le repère lié à la base mobile (M étant son point de référence),
- $\mathcal{R}_P(P,\ \vec{x}_P,\ \vec{y}_P,\ \vec{z}_P)$ est le repère lié à la platine (P représentant son centre de rotation),
- $\mathcal{R}_C(C,\ \vec{x}_C,\ \vec{y}_C,\ \vec{z}_C)$ est le repère lié à la caméra (C désignant son centre optique).

On note de plus θ et q_{pl} les angles définissant respectivement les orientations du robot par rapport à la scène et de la platine par rapport au robot. D_x désigne la longueur de l'entraxe, a et b correspondant aux coordonnées de C dans $\mathcal{R}_P$. Enfin, le robot étant commandable en vitesse, son vecteur de commande peut s'écrire $\dot{q} = [v\ \omega\ \omega_{pl}]^T$ où v et ω représentent respectivement les vitesses linéaire et angulaire de la base mobile, ω_{pl} désignant la vitesse angulaire de la platine.

Le mouvement de la caméra est défini par ses vitesses linéaire et angulaire $V_{C/\mathcal{R}}$ et $\Omega_{\mathcal{R}_C/\mathcal{R}}$ par rapport au repère scène $\mathcal{R}$. Or, pour le robot considéré, la caméra ne peut que se déplacer linéairement selon les axes $\vec{y}_C$ et $\vec{z}_C$, et tourner autour de $\vec{x}_C$. On montre alors que les vitesses 'réalisables' peuvent être regroupées dans un vecteur T_{red} qui s'exprime de la manière suivante en fonction de $\dot{q}$:

$$T_{\text{red}} = \begin{pmatrix} V_{y_c} \\ V_{z_c} \\ \Omega_{x_c} \end{pmatrix} = \begin{pmatrix} -\sin q_{pl} & a + D_x \cos q_{pl} & a \\ \cos q_{pl} & -b + D_x \sin q_{pl} & -b \\ 0 & -1 & -1 \end{pmatrix} \dot{q} = J_{\text{red}}\, \dot{q} \tag{1}$$

où J_{red} représente la matrice jacobienne de l'ensemble robot & caméra et définit son MCD. On précise que son déterminant est non nul et égal à D_x.

1.2 Modélisation de la caméra embarquée

On considère ici le modèle sténopé de la caméra embarquée (cf. figure 2 (à gauche)). Les points de la scène sont donc projetés sur le plan image par une projection perspective (cf. figure 2 (à droite)). Ainsi, si le vecteur $\underline{\mathrm{x}}_p = [x\ y\ z]^T$ représente les coordonnées d'un point de la

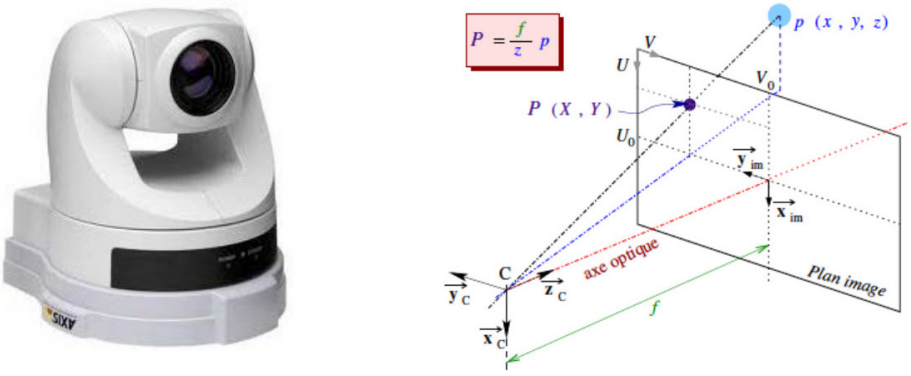


FIGURE 2 – Modèle pinhole de la caméra

scène p exprimées dans le repère caméra R_C, alors sa projection P sur le plan image a pour coordonnées :

$$\underline{\mathrm{X}}_P = \frac{f}{z}\, \underline{\mathrm{x}}_p \tag{2}$$

où f représente la distance focale de la caméra. Dans toute la suite, on supposera que la caméra a une focale unité $f = 1$ (normalisation). Enfin, $\underline{\mathrm{X}}_P$ définit les coordonnées métriques du point projeté P dans l'image.

1.3 Modélisation de l'interaction caméra/environnement

La matrice d'interaction relie le mouvement des indices visuels dans l'image au mouvement de la caméra. En d'autres termes, elle relie la variation du signal capteur aux vitesses de la caméra définies par T_{red}. Dans le cas d'un point P_i et pour le robot considéré, elle s'écrit :

$$L_{\text{red}_i} = \begin{pmatrix} 0 & X_i/z_i & X_iY_i \\ -1/z_i & Y_i/z_i & 1+Y_i^2 \end{pmatrix} \tag{3}$$

où (X_i, Y_i) représente les coordonnées métriques du point projeté dans l'image, z_i ($i = 1$ à 4) désignant la profondeur de ce point. La matrice d'interaction L_{total} associée à N points P_i est donc constituée par la concaténation de l'ensemble des matrices L_{red_i} correspondantes et s'écrit :

$$L_{total} = \begin{pmatrix} L_{\text{red}_1} \\ \vdots \\ L_{\text{red}_N} \end{pmatrix} \tag{4}$$

1.4 Données techniques

Cette partie rassemble les données techniques nécessaires à la simulation :

- Paramètres du robot :

  $D_x = 0.1\text{m}$ $\quad \overrightarrow{PC}^{\mathcal{R}_P} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ $\quad h = 0.4\text{m}$ $\quad r = 0.4\text{m}$

  où h et r désignent respectivement les hauteurs entre la platine P et la plateforme M et entre la plateforme M et le sol.
- Motif de référence : Les valeurs des indices visuels correspondant à une parfaite exécution de la tâche robotique désirée sont les suivantes :

  $s^* = [X_1^* = -0.2,\ Y_1^* = 0.2,\ X_2^* = -0.2,\ Y_2^* = -0.2,\ X_3^* = 0.2,\ Y_3^* = -0.2,\ X_4^* = 0.2,\ Y_4^* = 0.2]^T$
- Cible choisie : La cible de forme carrée comprend quatre points P_i dont les **coordonnées homogènes** dans le repère de la scène R sont les suivantes :

  $$\overrightarrow{OP_1}^{\mathcal{R}} = \begin{pmatrix} 8 \\ 2.2 \\ 1 \\ 1 \end{pmatrix} \quad \overrightarrow{OP_2}^{\mathcal{R}} = \begin{pmatrix} 8 \\ 1.8 \\ 1 \\ 1 \end{pmatrix} \quad \overrightarrow{OP_3}^{\mathcal{R}} = \begin{pmatrix} 8 \\ 1.8 \\ 0.6 \\ 1 \end{pmatrix} \quad \overrightarrow{OP_4}^{\mathcal{R}} = \begin{pmatrix} 8 \\ 2.2 \\ 0.6 \\ 1 \end{pmatrix}$$
- Paramètres complémentaires :

  Période d'échantillonnage $T_e = 10^{-2}$s, Focale $f = 1$, Gain de commande $\lambda = 0.25$, Gain tâche secondaire $\beta = 10$, Distance de sécurité $d_0 = 0.5$ m, Distance de détection $d_{switch} = 1$ m Rayon des obstacles $R = 0.2$ m

1.5 Fonctions fournies

Les fonctions évoluées d'affichage de la trajectoire et des indices visuels **sont fournies**. Tout affichage simple peut être réalisé à l'aide de la commande `plot` (cf. aide en ligne de matlab ou octave).

Sont également fournies les fonctions permettant de simuler la caméra (`visu.m`) et le laser (`distAndAlpha.m`). La simulation du déplacement du robot sous l'action d'une commande est également donnée.

2 Stratégie de commande

La tâche de navigation considérée consiste à positionner la caméra embarquée face à une cible composée de quatre points dans un environnement susceptible d'être encombré d'obstacles imprévus. Afin de réaliser cette tâche, on propose la stratégie de navigation suivante :

- Tant qu'aucun obstacle n'est détecté, le robot est commandé par l'asservissement visuel 2D classique.
- Dès qu'un obstacle est détecté et considéré comme suffisamment proche, le robot est commandé pour garantir la non collision.
- Lorsque l'obstacle ne présente plus de danger, l'asservissement visuel 2D est à nouveau utilisé.

Les obstacles sont détectés à l'aide d'un laser embarqué qui fournit d et α définis comme indiqué sur la figure 3 :

- d est la distance entre le point O_0 et le point le plus proche sur la surface de l'obstacle défini par sa projection orthogonale.
- α est l'angle entre la tangente à l'obstacle et la direction du robot définis respectivement par les axes $\vec{x}_{ob}$ et $\vec{x}_M$.

Dans ce cas, on montre que :

$$\dot{d} = -v\sin(\alpha) - \omega D_x \cos(\alpha) \tag{5}$$

$$\dot{\alpha} \approx \omega - v\cos(\alpha)\frac{1/R}{1 + d/R} \tag{6}$$

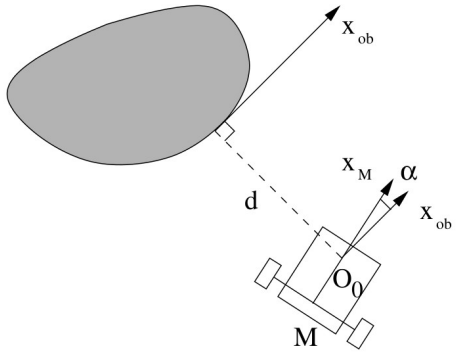


FIGURE 3 – Évitement d'obstacle : distance et orientation relative robot/obstacle.

3 Travail demandé

Partie 1 : Asservissement visuel classique

1. On suppose l'environnement libre et on souhaite réaliser un asservissement visuel 2D classique. Rappeler l'expression de la commande $\dot{q}$ permettant d'annuler exponentiellement l'erreur $s - s^*$. Compléter le programme fourni et vérifier que le robot effectue bien

la tâche souhaitée (cf. TP de l'an dernier).

Partie 2 : Formalisme des tâches redondantes

2. On rajoute maintenant un obstacle et on suppose que la configuration initiale du robot est proche de ce dernier. Il y a donc un risque de collision. Pour cela, on propose de définir une tâche d'évitement garantissant que :
   ① le robot contourne l'obstacle en passant à une distance de sécurité d_0
   ② la cible soit conservée dans l'image tout au long du contournement
   Indiquer les commandes du robot impactées par la contrainte définie en ①. Conclure quant à la redondance de la tâche de contournement.
3. On cherche maintenant à modéliser la tâche d'évitement à l'aide du formalisme des tâches redondantes. Rappeler le principe de ce formalisme.
4. On propose ici que la tâche principale soit définie par le contournement de l'obstacle, la tâche secondaire étant alors donnée par le centrage de la cible dans l'image.
   (a) Justifier ce choix.
   (b) Déterminer l'erreur e_d à annuler pour contourner l'obstacle à une distance d_0. En déduire la jacobienne [1] J_d à partir des informations fournies précédemment. Étudier le rang de J_d. La tâche est-elle redondante ?
   (c) Proposer un critère h permettant de conserver la cible dans l'image pendant le mouvement de contournement. Déterminer son gradient.
   (d) En utilisant le formalisme des tâches redondantes, écrire la tâche globale e modélisant l'évitement et donner l'expression de $\dot{q}$ permettant d'annuler cette erreur.
   (e) Programmer cette loi de commande et valider son fonctionnement.
5. Valider maintenant la tâche complète en proposant des conditions de basculement entre les commandes. Constatez-vous d'éventuels problèmes ?

Partie 3 : Amélioration de la loi de commande

6. On propose maintenant d'ajouter une grandeur à la tâche principale pour aider au contournement de l'obstacle. Définir une nouvelle erreur e_d qui réalise cette modification en vous aidant des conditions de basculement définies précédemment. La tâche secondaire peut rester identique. Programmer cette loi de commande et valider son fonctionnement
7. Selon les résultats et le temps disponible, proposez des améliorations possibles.

1. J_d est telle que $\dot{e}_d = J_d\dot{q}$.