

EXAMEN DE ROBOTIQUE MOBILE

Epreuve avec documents de cours - Durée : 1h

Nom :

Prénom :

1. Soit un système robotique S constitué de deux robots mobiles circulaires (de même rayon r) holonomes, A et B , qui sont reliés en leur centre par une barre qui peut pivoter autour de l'axe vertical des robots (la distance entre les robots est fixe). L'environnement W est un plan qui contient des obstacles polygonaux. Une fonction *MODELISATION* permet de modéliser W sous forme d'une grille 2D (*MNT* binaire ou *bitmap*). La valeur d'une cellule de la grille égale 0 (1) si la cellule est sans (avec) obstacle. La taille d'une cellule est un carré dont le coté mesure $\frac{r}{2}$.

On veut planifier la trajectoire des robots en imposant les positions initiales et finales (positions qui respectent leur liaison).

- (a) Proposer une méthode de planification de trajectoire pour résoudre ce problème sachant que la localisation et la commande des robots se font sans erreur. Décrire les fonctions mises en oeuvre et leurs enchaînements. Justifier vos choix et expliquer le fonctionnement de vos fonctions. Quelle est la dimension de l'espace des configurations ?
 - (b) On suppose maintenant que le robot A est de type voiture avec un rayon de giration borné (B reste holonome). Donner les conséquences de cette contrainte sur votre approche. Expliquer comment vous pouvez résoudre le problème de planification dans ce cas ? Quelle est la dimension de l'espace des configurations ?
2. On désire connaître la position absolue d'un robot mobile Rob à l'aide d'un capteur CL dans le repère global (O, X_0, Y_0) .

Un système de balises $B_j (j = 1, \dots, n)$ couvre l'environnement de travail (la position des balises (x_{b_j}, y_{b_j}) est connue précisément).

Lorsqu'on interroge le capteur CL , on lui donne, en entrée, les numéros des deux balises qu'on cherche à détecter, par exemple $DETECTE(i, j)$. S'il détecte les deux balises, B_i et B_j , il retourne la plus courte distances robot-balise, l'indice de la balise la plus proche et l'angle entre la direction du robot et la direction des balises (voir figure 1) [valeurs retournées = d, k, α]. Sinon il ne retourne rien [valeur retournée = NUL].

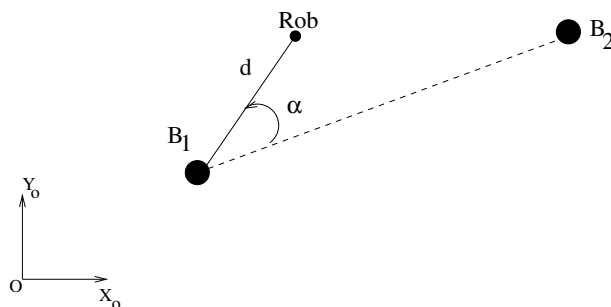


FIGURE 1 –

- Expliquer comment effectuer la localisation avec ce capteur en utilisant le minimum de balise.
- Donner le système d'équations permettant de calculer la position du robot (ne pas résoudre).
- Que se passe-t-il si le capteur retourne la distances avec une erreur ? Donner une interprétation géométrique de cette erreur sur la position du robot.

3. Expliquer le fonctionnement de l'algorithme de planification probabiliste suivant (*PPA*) sachant qu'on connaît :
 - la géométrie des différents corps (polyèdres) du robot ainsi que leurs liaisons,
 - le placement et les caractéristiques géométriques de tous les obstacles (polyèdres).

Algorithm 1 Algorithme PPA(q_{start}, q_{goal}, k) - INPUT : configuration initiale, finale et un entier

```

1: T.init  $\leftarrow q_{start}$  et T.goal  $\leftarrow q_{goal}$  // Initialiser les arbres T.init et T.goal.
2: for  $i \leftarrow 1$  to  $k$  do
3:   Prendre aléatoirement T= T.init ou T.goal avec probabilité  $\frac{1}{2}$ .
4:   repeat
5:     Prendre aléatoirement un noeud  $m$  de T
6:     repeat
7:       Générer aléatoirement un noeud  $p$  dans le voisinage de  $m$ 
8:     until ( $p$  est sans collision)
9:     Ajouter  $q$  dans T
10:  until (un nouveau noeud est créé dans T)
11:   $\tau \leftarrow \text{CONNECT}$ 
12:  If  $\tau \neq \text{NIL}$  return  $\tau$ 
13: end for
14: return Echec

```

Algorithm 2 Algorithme CONNECT

```

1:  $m \leftarrow$  le noeud le plus récent créé
2:  $m' \leftarrow$  noeud le plus proche dans l'arbre ne contenant pas  $m$ 
3: if distance( $m, m'$ )  $\leq \rho$  then
4:   Connecter  $m$  et  $m'$  par une arête
5:    $\tau \leftarrow \text{CHEMIN}(q_{start}, q_{goal})$  // calcule le chemin connectant  $q_{start}$  et  $q_{goal}$ 
6:   Return TEST-PATH( $\tau$ ) //fonction retourne NIL si  $\tau$  en collision, sinon retourne  $\tau$ .
7: end if
8: return NIL

```

Comparer cette approche probabiliste *PPA* avec les méthodes *PRM* et *RRT*.