Khepera IV
Manuel d'utilisation

BASCOUR Gwenael – CAUSSE Jade – DA COSTA Yacine
NAYET Morgan – PARES Lucien

Table des matières

1 Généralité : Acceleration

Nombre de nos programmes utilisent la fonction d'acceleration du robot Khepera : Pour gérer l'accélération du robot, on utilise la fonction :

```
kh4_SetSpeedProfile(accinc, accdiv, minspacc, minspdec, maxsp, dsPic );
```

- accinc : correspond au nombre de pulsations/sec ajoutées à chaque accélération
- accdiv : la vitesse augmente après chaque X+1 pulsations de l'odomètre
- minspacc : correspond au nombre de pulsations/sec minimales ajoutées à chaque accélération
- minspdec : paramètre en lien avec les accélérations linéaires (non utilisé)
- maxsp : valeur maximale de la vitesse possible (1200 pulsations/sec)

2 Généralité : Rotation autour de son axe

Nombre de nos programmes utilisent une méthode de rotation : Le programme récupère les positions actuelles de chacune des 2 roues (unité de mesure utilisée en "pulse" pour le nombre de pulsation affectuées) Le robot va positionner ses roues sur le nombre de pulsation souhaité.

Le nombre de pulsation dont doit varier chacune des 2 roue est identique pour celles-ci et est déterminée par l'équation :

$$\begin{aligned}\text{puls\_cible} &= \text{rapport\_degre\_sur\_360} * \text{périmètre\_cercle\_diametre\_roues} * \text{conversion\_puls\_to\_mm} \\ &= (\text{degre}/360) * (2 * \pi * \text{espacement\_roues}) * \text{revolution\_roue\_pulse}/(\text{diametre\_roue} * \pi) \\ &= (\text{degre}/360) * (2 * \text{espacement\_roues}) * \text{revolution\_roue\_pulse}/\text{diametre\_roue} \\ &= (\text{degre}/360) * (2 * 52,7) * 19456/42\end{aligned}$$

Les nouvelles positions de chacune des deux roues en pulse sont déterminées par les équations :

$$\text{nouvelle\_pos\_roue\_gauche} = \text{ancienne\_pos\_roue\_gauche} + \text{puls\_cible}$$

$$\text{nouvelle\_pos\_roue\_droite} = \text{ancienne\_pos\_roue\_droite} - \text{puls\_cible}$$

3 Acceleration vers un mur

3.1 But du comportement

Le but de ce comportement est de faire accelerer (voir Généralité : Acceleration) le robot khepera vers un mur et de le faire s'arrêter à une distance fournie par l'utilisateur de ce dernier.

3.2 Capteurs & actionneurs utilisés

Ce comportement utiise les capteurs a ultrasons afin de connaitre la distance qui le sépare d'un mur, et utilise les roues afin d'accelerer vers celui-ci.

3.3 Fonctionnement du comportement

Dans un premier temps, le robot va mesurer la distance qui le sépare d'un mur ou d'un quelconque obstacle, puis il va utiliser celle-ci afin de parcourir cette distance en accelerant.

3.4 Calculs utilisés

Pour ce comportement on retrouve un simple calcul, la transformation de la distance mesurée en nombre de pulsations necessaires pour parcourir la distance

$$\text{nbPulsations} = \text{distanceMesuree}/\text{KH4\_PULSE\_TO\_MM}$$

KH4_PULSE_TO_MM étant une constante de la librairie Khepera

4 Arrêt à 50 cm

4.1 But du comportement

L'objectif de ce comportement est de permettre au robot d'avancer jusqu'à se retrouver à 50cm d'un obstacle et de s'arrêter à cette distance de ce dernier.

4.2 Capteurs & actionneurs utilisés

Ce comportement utilise les capteurs à ultrasons pour mesurer la distance qui le sépare d'un obstacle et les roues afin de gérer le déplacement du robot.

4.3 Fonctionnement du comportement

Le robot utilise dans un premier temps ses capteurs à ultrasons afin d'évaluer la distance qui le sépare de l'obstacle repéré. Il soustrait ensuite les 50cm à la distance obtenue et parcours la nouvelle distance calculée pour s'arrêter à 50cm de l'obstacle.

4.4 Calculs utilisés

Seules des convertions d'unité de distance sont utilisées pour ce programme. Conversion de millimètres en centimètres et inversement.

5 Labyrinthe

5.1 But du comportement

Le but de ce comportement est de permettre au robot de sortir d'un labyrinthe sans encombre.

5.2 Capteurs & actionneurs utilisés

Ce comportement utilise les capteurs infrarouge afin d'évaluer s'il se trouve proche où non d'un mur. Ceux-ci fonctionnent avec la lumière, de ce fait, la luminosité ambiante ou encore la couleur des murs du labyrinthe peuvent influencer l'évaluation du robot sur la distance qui le sépare d'un mur. Les roues sont aussi utilisées pour gérer les différents déplacement du robot au sein du labyrinthe.

5.3 Fonctionnement du comportement

Lorsque le robot se trouve dans le labyrinthe, il suit grâce à ses capteurs infrarouge le mur se situant sur sa droite et ce durant tout le temps pendant lequel il se trouve à l'intérieur du labyrinthe. Le programme possède 5 comportements différents pour gérer le déplacement du robot dans le labyrinthe :
S'il ne détecte pas d'obstacle sur sa droite : il va alors déduire qu'il doit aller dans cette direction. Pour cela, il va s'avancer de la moitié de son diamètre (70mm avec une marge de 2mm) et ensuite effectuer une rotation à 90° pour prendre le chemin accessible à sa droite.
S'il détecte toujours la présence d'un mur à sa droite, et aucun obstacle devant, alors il va se contenter de continuer d'avancer.
S'il capte le mur à sa droite et un obstacle à l'avant, alors il va vérifier que rien ne se trouve sur sa gauche et effectuer une rotation de 90° pour aller sur le chemin libre à sa gauche.
S'il détecte un obstacle à droite, devant et à gauche, il se trouve dans un cul de sac et fera donc une rotation de 180° afin de faire demi-tour.
S'il capte qu'il s'éloigne trop du mur de droite, il va effectuer une rotation de 5° vers le mur afin de revenir à une distance convenable de celui-ci.
S'il capte qu'il se rapproche trop du mur de droite et qu'il y a un fort risque de collision, il va effectuer une rotation de 5° sur la gauche afin de revenir à une distance convenable de celui-ci.

5.4 Calculs utilisés

Les calculs utilisés dans ce programme sont deux de la rotation pour la bonne orientation du robot. (cf. Généralité : rotation autour de son axe)

6 Parcours distance

6.1 But du comportement

Le but de ce programme est de permettre au robot de parcourir la distance demandée par un utilisateur.

6.2 Capteurs & actionneurs utilisés

Seules les roues et l'odomètre sont utilisés pour permettre le déplacement du robot en ligne droite.

6.3 Fonctionnement du comportement

Dans un premier temps l'utilisateur est amené à saisir une distance en millimètre. Le robot parcours ensuite la distance saisie par l'utilisateur.

6.4 Calculs utilisés

Le calcul utilisé pour que le robot puisse effectuer la bonne distance est le suivant :

$$nouvelle_pos = \text{ancienne\_pos} + (\text{distance\_mm} * \text{revolution\_roue\_pulse}/(\text{diametre\_roue} * \pi)) \quad = \text{ancienne\_pos}$$

7 Cartographie

7.1 But du comportement

Le but de ce comportement est de permettre au robot de cartographier son environnement en affichant la distance qui le sépare des différents obstacles présents tout autour de lui.

7.2 Capteurs & actionneurs utilisés

Les capteurs utilisés sont les capteurs à ultrasons pour évaluer la distance qui le sépare des obstacle ainsi que les roues pour permettre sa rotation.

7.3 Fonctionnement du comportement

Le robot va dans un effectuer sa cartographie en 3 temps. Dans un premier temps, il va évaluer avec les capteurs à ultrasons la distance qui le sépare des objets se situant devant lui et sur les côtés. Il va ensuite stocker les valeurs obtenues en vue de l'affichage ultérieur des données à l'utilisateur. Dans un second temps, il va effectuer une rotation de 180° et réaliser un nouveau scan de son environnement, mais, cette fois ci sur les objets se trouvant initialement derrière lui. Il va également stocker les données obtenues. Et enfin dans un 3ème et dernier temps, il va à nouveau effectuer une rotation de 180° afin de revenir dans son état de départ. C'est à ce moment là qu'il affichera à l'utilisateur la distance en centimètre qui le sépare des divers obstacles qu'il aura pu rencontrer, ainsi que leur position par rapport à son orientation initiale.

7.4 Calculs utilisés

Le calcul utilisé ici est celui permettant les deux rotations de 180° qu'effectue le robot pour réaliser le scan de son environnement. (cf. Généralité : rotation autour de son axe)

8 Parcours Acceleration

8.1 But du comportement

Le but de ce comportement est de parcourir un mètre en accelerant (voir Généralité : Acceleration)

8.2 Capteurs & actionneurs utilisés

Ce comportement utilise les roues afin de se mouvoir.

8.3 Fonctionnement du comportement

A l'appel de la méthode "parcourir", le mode du robot est positionné sur "RegPosition" soit le contrôle des roues et des moteurs par l'odometre et le profil de la vitesse est modifié pour limité les variations de vitesse par le robot.
Le programme récupère les positions actuelles de chacune des 2 roues (unité de mesure utilisée en "pulse" pour pulsation)
Le robot va positionner ses roues sur le nombre de pulsation souhaité, pour celà il avancera tout droit.

8.4 Calculs utilisés

La nouvelle position de chaque roue en pulse est déterminée par l'équation :

$$\begin{aligned} \text{nouvelle\_pos} &= \text{ancienne\_pos} + (\text{distance\_mm} * \text{revolution\_roue\_pulse}/(\text{diametre\_roue} * \pi)) \\ &= \text{ancienne\_pos} + (\text{distance\_mm} * 19456/(42 * \pi)) \end{aligned}$$

9 Parcours pièce

9.1 But du comportement

Le but de ce comportement est de permettre au robot de se déplacement librement dans une pièce en évitant les obstacles qu'il rencontre.

9.2 Capteurs & actionneurs utilisés

Ce comportement utilise les capteurs à infrarouge pour détecter les obstacles à proximité et les roues afin de se déplacer.

9.3 Fonctionnement du comportement

Le robot avance à vitesse constante et en ligne droite jusqu'à ce qu'il détecte un obstacle à proximité. Une fois détecté, il va s'arrêter et effectuer une rotation (cf. Généralité : rotation autout de son axe).
Selon le capteur qui détecte l'obstacle le robot réalisera une rotation différente :

- Capteur avant-gauche : Le robot va réaliser une rotation de 90° dans le sens horaire.
- Capteur avant : Le robot va réaliser une rotation de 135° dans un sens aléatoire.
- Capteur avant-droit : Le robot va réaliser une rotation de 90° dans le sens anti-horaire

9.4 Calculs utilisés

Les calculs utilisés lors de la rotation cf. Généralité : rotation autour de son axe

10 Pivoter autour de son axe

10.1 But du comportement

Le but de ce comportement est de faire en sorte que le robot tourne autour de son axe, d'un angle donné par l'utilisateur

10.2 Capteurs & Actionneurs utilisés

Ce comportement utilise les roues afin de tourner et les odomètres fin de se reperer

10.3 Fonctionnement du comportement

(cf. Généralité : Rotation autour de son axe)

10.4 Calculs utilisés

(cf. Généralité : Rotation autour de son axe)

11 Pivoter autour d'une roue

11.1 But du comportement

Le but de ce comportement est de faire en sorte que le robot tourne autour d'une de ses roues (choisie par l'utilisateur), d'un angle donné par l'utilisateur

11.2 Capteurs & Actionneurs utilisés

Ce comportement utilise les roues afin de tourner et les odomètres fin de se reperer

11.3 Fonctionnement du comportement

A l'appel de la méthode "pivoter", le mode du robot est positionné sur "RegPosition" soit le contrôle des roues et des moteurs par l'odomètre et le profil de la vitesse est modifié pour limité les variations de vitesse par le robot.

Le programme récupère les positions actuelles de chacune des 2 roues (unité de mesure utilisée en "pulse" pour le nombre de pulsation affectuées)
Le robot va positionner ses roues sur le nombre de pulsation souhaité.

11.4 Calculs utilisés

Le nombre de pulsation dont doit varier 1 des 2 roues est déterminée par l'équation :

$puls\_cible = \text{rapport\_degre\_sur\_360} * perimetre\_cercle\_diametre\_roues * conversion\_puls\_to\_mm \quad = (\text{degr}$

Les nouvelles positions de chacune des deux roues en pulse sont déterminées par les équations :

nouvelle_pos_roue_mobile = ancienne_pos_roue_mobile+puls_ciblenouvelle_pos_roue_immobile = ancienne_p