alexandre.lombard@utbm.fr

CONSIGNES

Les robots EV3 utilisés dans ce TP sont fournis déjà montés, il est déconseillé de les démonter ou de les modifier.

Votre code devra être clair, organisé et documenté. Vous respecterez les conventions de programmation de Java et commenterez chacune de vos fonctions en utilisant le standard Javadoc.

PRISE EN MAIN DE LEJOS

LeJOS est une machine virtuelle Java fonctionnant sur les briques LEGO EV3. Elle offre accès à toutes les fonctions d'une JRE embarquée ainsi qu'à une API de programmation dédiée aux briques EV3.

La documentation complète de cette API est disponible à l'adresse suivante : http://www.lejos.org/ev3/docs/

Dans la première partie de ce TP vous apprendrez à utiliser l'API Java pour contrôler les robots.

PILOTAGE DES MOTEURS

Utilisez l'interface RegulatedMotor pour piloter les moteurs. Créez les fonctions suivantes :

- void forward() : le robot se déplace en avant
- void rotate(float angle) : le robot tourne sur lui-même, *angle* est l'angle en radians de la rotation
- void stop() : le robot arrête tout déplacement en cours

UTILISATION DES CAPTEURS

Les robots sont équipés de détecteurs de distance (sonars), de gyroscopes et de capteurs de couleur. Cet exercice montre comment y accéder via l'API Java EV3.

CAPTEUR DE DISTANCE

Créez la fonction float distance() qui retourne la distance renvoyée par le capteur de distance (voir EV3UltrasonicSensor) dans l'API.

Ensuite, appelez cette fonction, affichez le résultat des mesures sur l'écran LCD et enregistrez-les dans un fichier CSV.

Créez la fonction float distance(int n) qui retourne la distance renvoyée par le capteur de distance après application d'un filtre moyenneur de n échantillons. Comparez les résultats avec la méthode distance() pour n = 2, n = 5 et n = 10.

CAPTEUR DE COULEURS

Créez une fonction qui fait avancer votre robot jusqu'à ce que le capteur couleur rencontre un marqueur au sol de couleur définie.

Vous réutiliserez le code produit dans la partie « Pilotage des moteurs » pour gérer le déplacement du robot.

CONTROLE « HAUT-NIVEAU »

Créez une classe Path contenant une liste de points définis par des coordonnées réelles (x, y). Ensuite créez une méthode followPath(Path p) permettant au robot de suivre la liste de points passés en paramètre.

Vous implémenterez deux approches :

alexandre.lombard@utbm.fr

- Dans la première, le robot se déplace en ligne droite entre deux points, s'arrête, tourne sur lui-même puis se dirige sur le point suivant.
- Dans la seconde, le robot ne s'arrête pas sur les points, sa vitesse doit être constante.

ROBOT SUIVEUR

Pour réaliser le robot suiveur vous disposerez de deux robots EV3. Ces deux robots seront placés sur une même ligne, le premier étant le robot « leader » et le second le robot « suiveur ». En utilisant les informations renvoyées par le capteur d'ultra-son le robot « suiveur » essaiera de maintenir une distance de 15 cm avec le robot « leader ».

PROGRAMME DU ROBOT LEADER

Le robot leader avance pendant un temps T_l, puis s'arrête pendant ce même temps. Ceci est répété de manière cyclique. La vitesse du robot doit être fixée à 40% de sa vitesse maximale.

PROGRAMME DU ROBOT SUIVEUR

Pour le robot suiveur, vous appliquerez successivement les quatre politiques suivantes :

- « Tout ou rien » : le robot avance à 50% de sa vitesse maximale s'il ne détecte pas d'obstacle. Si un obstacle est à moins de 15 cm, le robot s'arrête.
- « A un point » : la vitesse du robot est calculée de la manière suivante :

$$\%(t+T_s) = \max\big(\min\big(50, a \times (d(t) - D)\big), 0\big)$$

- « A deux points » : la vitesse du robot est calculée de la manière suivante :

$$\%(t+T_s) = \min\big(\max\big(2.5 \times (d(t) - 20), \min\big(\max(a \times (d(t) - D), 0), \%(t)\big)\big), 50\big)$$

- « Communication sans-fil » (optionnel) : le robot leader doit envoyer sa vitesse actuelle en utilisant le Wifi (un point d'accès Wifi commun, tel qu'un téléphone ou un ordinateur portable sera utilisé). La fréquence d'envoi des messages sera paramétrable. Le robot suiveur doit appliquer la vitesse envoyée par le robot leader. En l'absence de message reçu, la politique « tout ou rien » sera appliquée.

Dans les équations données ci-dessus, %(t) indique la vitesse appliquée le cycle précédent (en pourcentage de la vitesse maximale), T_s est le temps de cycle et $d(t)$ désigne la distance renvoyée par le capteur.

Testez empiriquement les 4 politiques de suivi pour $T_l = 500ms, 3s$ et $8s$ avec différentes valeurs de D, de a et de T_s.

Rédigez un compte-rendu (10 pages maximum) des tests :

- Expliquez le rôle des différents paramètres T_s, D et a
- Enregistrez et affichez la courbe de distance inter-véhiculaire et la courbe de vitesse du véhicule suiveur pour les tests significatifs, justifiant les choix de T_s, de D et de a.
- Donner les observations concernant les courbes choisies et les tests réalisés
- Conclure sur l'intérêt de ces politiques