

FONCTIONS DISPONIBLES

Lors de ce TER, vous avez la possibilité de commander les différents composants de la simulation mise à votre disposition.

A partir de la classe Robots :

1) Gestion des robots

Les 4 robots industriels présents dans cette simulation sont des robots 6 axes de type KUKA LBR IIWA 14 R820. Les fonctions listées ci-dessous vous permettront de contrôler chacun de ces robots. Les arguments de ces fonctions pourront être choisis à l'aide du schéma récapitulatif *Cellule flexible : schéma récapitulatif des arguments*.

→ **EnvoyerPosition(numRobot, numPosition) :**

4 positions ont été prédéfinies pour chacun des robots : les deux zones de chargement/déchargement et les deux postes de travail. Pour y accéder, l'argument *numRobot* permet de choisir le robot que vous souhaitez contrôler et l'argument *numPosition* identifie la position dans laquelle le robot choisi sera envoyé.

Chacun des arguments est un entier compris 1 et 4 (cf *schéma récapitulatif*).

→ **EnvoyerAngles(numRobot, angle1, angle2, angle3, angle4, angle5, angle6, angle7) :**

Si vous souhaitez envoyer un robot dans une autre position que celles prédéfinies, l'argument *numRobot* vous permet de sélectionner ce robot. Les arguments *angle1* à *angle6* sont les angles en degrés des différents axes du robot et l'argument *angle7* permet de faire pivoter la pince.

Par exemple, la position prédéfinie n°1 d'un robot est obtenue à l'aide des angles :

angle1 = 128 ; *angle2* = 90 ; *angle3* = 90 ; *angle4* = 80 ; *angle5* = 90 ; *angle6* = -90 ; *angle7* = -40.

→ **DescendreBras(numRobot) et MonterBras(numRobot) :**

Ces fonctions vont vous permettre de descendre ou de monter le bras du robot *numRobot* suivant l'axe vertical. Pour utiliser ces fonctions, veillez à être dans une des positions prédéfinies du robot.

→ **OuvrirPince(numRobot) et FermerPince(numRobot) :**

Ces fonctions permettent d'ouvrir ou de fermer la pince du robot défini par *numRobot*.

→ **ControlerRobot(numRobot, numPosition, etatBras, etatPince) :**

Cette fonction vous permettra d'envoyer le robot *numRobot* dans une des positions prédéfinies, tout en plaçant le bras dans l'état souhaité du bras ainsi que de la pince. L'argument *numPosition* vous permet de choisir la position prédéfinie. Le bras sera placé dans l'état bas si l'argument *etatBras* vaut -1 et dans l'état haut si l'argument a pour valeur 1. L'argument *etatPince* permet de définir l'état souhaité de la pince, en le fixant à la valeur -1 la pince sera ouverte alors que la valeur 1 fera ouvrir la pince. Les actions se dérouleront de manière successive.

2) Retour des robots

Les fonctions listées ci-dessus permettent de contrôler les différents mouvements possibles des robots. Les fonctions présentées dans la suite de ce document permettent d'avoir un retour de ces robots et de savoir si les mouvements ont bien été réalisés :

→ **RobotInitialise(numRobot)** :

Cette fonction retourne 1 si le robot *numRobot* a été correctement initialisé. Le robot *numRobot* ne pourra pas être contrôlé s'il n'est pas initialisé.

→ **RobotEnPosition(numRobot)** :

Cette fonction retourne 1 lorsque le mouvement précédemment commandé au robot *numRobot* est terminé.

→ **BrasEnPosition(numRobot)** :

Cette fonction retourne 1 si le bras du robot *numRobot* est en position haut, -1 s'il est descendu.

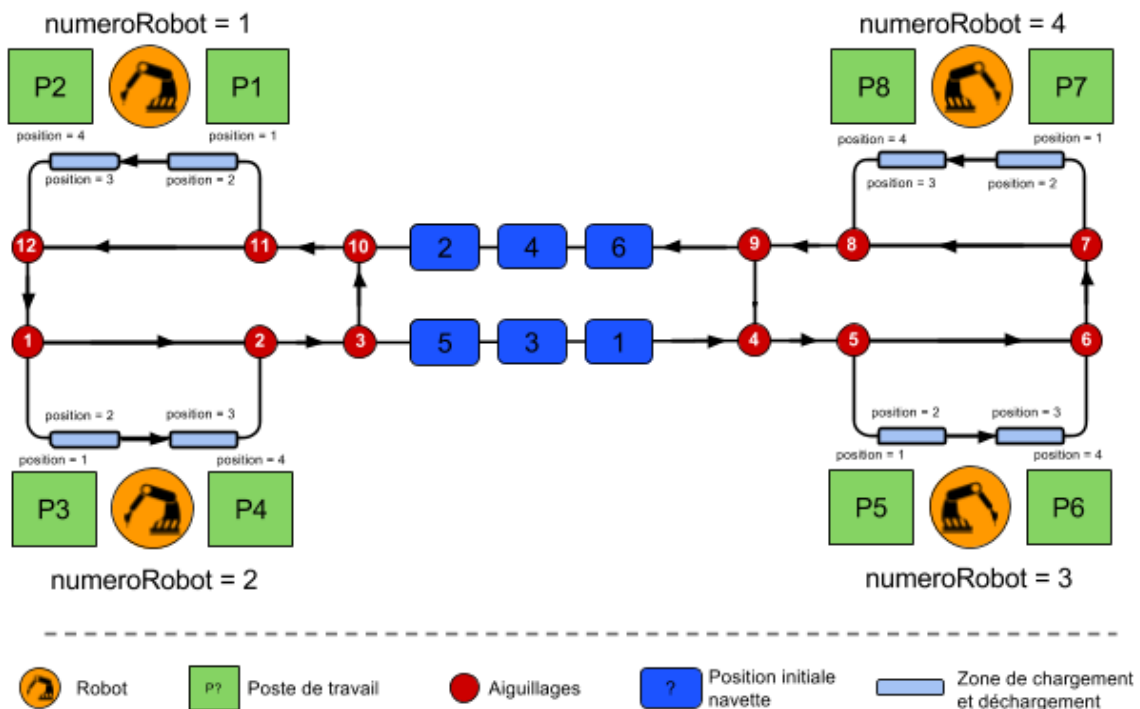
→ **PinceEnPosition(numRobot)** :

Cette fonction retourne 1 si la pince du robot *numRobot* est fermée, -1 si la pince est ouverte.

3) Retour des tâches

→ **TraitementFini(numTache)** :

Cette fonction retourne 1 lorsque le poste *numTache* a fini le traitement de son produit.



Cellule flexible : schéma récapitulatif des arguments

A partir de la classe Commande :

1) Gestion de l'arrivée des navettes

→ **NouvelleNavette()** :

Cette fonction retourne le numéro de la navette qui vient d'être créée par le mode automatique ou de manière manuelle.

→ **ReinitialiserNouvelleNavette()** :

Cette fonction permet de réinitialiser la variable qui permet de gérer l'arrivée des navettes.

2) Gestion de l'arrêt et du départ des navettes aux zones de chargement/déchargement

→ **NavetteStoppeeVide(numPoste)** :

Cette fonction renvoie le numéro de la navette stoppée au niveau de la zone de chargement/déchargement *numPoste* si cette navette est vide, 0 sinon.

→ **NavetteStoppee(numPoste)** :

Cette fonction renvoie le numéro de la navette stoppée au niveau de la zone de chargement/déchargement *numPoste* si cette navette contient un produit, 0 sinon.

→ **NavettePartie(numPoste)** :

Cette fonction doit être appelée lorsqu'une navette part de la zone de chargement/déchargement *numPoste* pour réinitialiser les variables associées à la présence d'une navette.

3) Gestion des destinations et des navettes disponibles

→ **DefinirDestination(numNavette, nDestination)** :

Cette fonction permet de définir la destination suivante *nDestination* de la navette *numNavette*.

→ **NavetteDisponible()** :

Cette fonction retourne le numéro d'une navette vide circulant dans la partie centrale D0 de la cellule.

4) Gestion des produits

→ **ProduitSurNavette(numNavette)** :

Cette fonction renvoie le produit présent sur la navette *numNavette*.

→ **PiecePrise(numPoste)** :

Cette fonction signale à la zone de chargement/déchargement *numPoste* qu'un produit a été récupéré par le robot associé et fait repartir la navette. On rappelle que le bras du robot doit être en position haute avant de redémarrer une navette.

→ **PieceDeposee(numPoste)** :

Cette fonction signale à la zone de chargement/déchargement *numPoste* qu'un produit a été déposé par le robot associé et fait repartir la navette. On rappelle que le bras du robot doit être en position haute avant de redémarrer une navette.