Projet Long N7 session 2020

Documentation

TER atelier flexible

Team SALLAG
Promo 2020

Tuteur : Mme U. Ngueveu Sandra
et M Briand Cyril

27 Janvier 2020 — 6 Mars 2020

Table des matières

1 Introduction

Ce document a pour but de détailler quelques aspects techniques afin que les personnes qui manipuleront les codes puissent s'y retrouver le plus facilement possible.

2 Lancement

Petit guide pour lancer le projet sur votre machine : (Ubuntu 16, ROS Kinetic)
Il est toujours bon de faire :

sudo apt-get update

Installer la librairie modbus avec :

sudo apt-get install libmodbus-dev

Cloner la branche master de ce repository : (si git n'est pas installé, vous pouvez toujours télécharger le projet en zip et en extraire le code)

git clone https ://github.com/PL2020/PL2020

Git ne fonctionne pas sur votre PC, il suffit de télécharger tout le dossier en fichier zip en utilisant l'url précèdent.

Compiler les packages ros en se plaçant dans PL2020/ros_ws et en effectuant :

catkin_make

Sourcez les fichiers compilés (depuis le dossier ros_ws) :

source devel/setup.bash

Exécuter le launch.sh à la racine du projet avec en argument le nombre de robot souhaité (2 ou 4) sachant que par défaut la simulation avec 4 robots se lance :

4 Robots : ./launch.sh ⇔ ./launch.sh 4
2 Robots : ./launch.sh 2

En résumé (pour copier/coller dans un terminal) :

```
git clone https ://github.com/PL2020/PL2020
cd PL2020
cd ros_ws
catkin_make
source devel/setup.bash
cd ..
./launch.sh
```

3 Le shuttle manager

Le shuttle manager est un programme qui permet de suivre les navettes et ainsi savoir quelle navette se trouve dans les zones de chargement et déchargement du robot pour pouvoir colorer ou décolorer les bons cubes présents sur les navettes.

Dans un premier temps, nous avons découpé notre maquette en 34 tronçons. On utilise les capteurs ainsi que les aiguillages pour connaître la position des navettes, c'est-à-dire sur quel tronçon elles se trouvent et dans quel ordre.

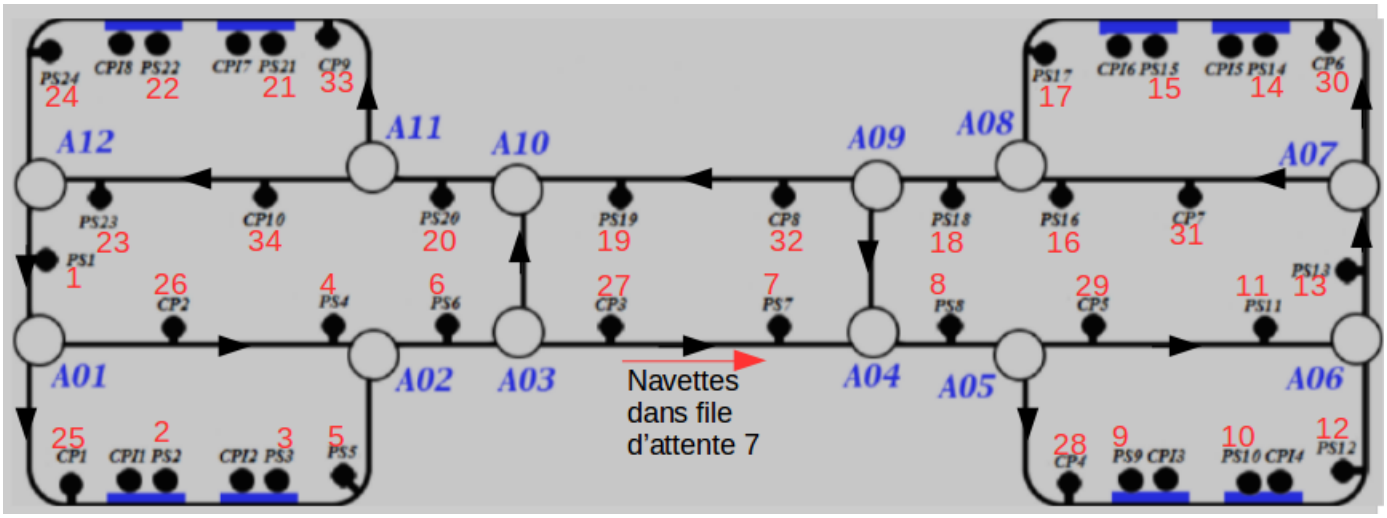

FIGURE 1 – Indices des tronçons du shuttle manager

Si au cours d'une simulation, un produit apparaît sur une mauvaise navette, il y a de forte chance que cela provienne du shuttleManager qui a perdu le fil d'une navette suite à une mauvaise manipulation dans le réseau de Petri. En effet, si une navette quitte un capteur PS précédant un aiguillage (front descendant) et qu'à ce moment là l'aiguillage n'est pas en butée droite ou gauche, on ne sait pas où va aller la navette. On considère donc que la navette est perdue : elle disparaît du shuttle manager, ce qui fausse tout le reste de la simulation. Il faut donc faire attention lors de la réalisation du réseau de Petri, à engager une navette sur un aiguillage que lorsque celui-ci est en butée.

Pour afficher les tronçons et ainsi voir le déplacement des navettes, il faut aller dans :

/nom_dossier/PL2020/ros_ws/src/shuttles/src/main_ShuttleManager.cpp

et décommenter toutes lignes contenant "debug_display". Ensuite pour qu'une fenêtre lui soit dédiée, on modifie le fichier :

/nom_dossier/PL2020/ros_ws/src/launcher/launch/launch_beta.launch

en modifiant la ligne 17 par :

<node name="shuttleManager" pkg="shuttles" type="main_ShuttleManager"
output="screen" launch-prefix="xterm -e"/>

Ainsi, à la prochaine compilation et exécution, une nouvelle fenêtre affiche les différents tronçons et les navettes qui y sont présentes.

Figure 2 – Affichage des différentes tronçons avec les navettes présentes

4 Augmenter le nombre de cube

Pour l'instant, les empilements sont constitués de 6 cubes (1 pour le type de produit et 5 pour les tâches). Si besoin, il est possible d'augmenter le nombre de cube sur les empilements. Voici les différentes étapes à suivre afin d'y parvenir :

- Ajouter le nombre de cube voulu sur tous les empilements des navettes et des postes dans Coppelia. Copier/coller un cube, le nommer pour une table "ProduitN_TX" avec N le numéro du cube et X l'id de la table, ou "ProduitN_Y" avec Y correspondant à la bonne navette (A,B,C,...), le positionner correctement et penser à le mettre dans la bonne hiérarchie ;
- Attribuer des noms signaux de couleur et récupérer leurs handles dans les scripts des navettes et des tables : "Table#X#X" et "ShuttleA#X"
- Changer la valeur de la constante NB_CUBE dans le fichier *commande_locale/src/vrepController.h*
- Changer la valeur de la variable MAX_TACHES (NB_CUBE - 1) dans le checker.

[2]

5 Code couleur sur Coppelia

Sur Coppelia, les différentes couleurs utilisées pour colorer les cubes sont définies dans les scripts des objets de la scène. Le code couleur utilisé est le suivant :

- Les codes couleur des produits sont : 14, 24, 34, 44, 54, 64 ;
- Les codes couleur des tâches finies sont : 13, 23, 33, 43, 53, 63, 73, 83 ;
- Les codes couleur des tâches en cours sont : 12, 22, 32, 42, 52, 62, 72, 82.