



**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE

Département de génie informatique et génie logiciel

INF3995

Projet de conception d'un système informatique

Procédures de tests

Équipe No **105**

Marc Jodel Dumesle ALCINDOR	2000081
Bailly Jonathan Clovis YARO	2128882
Zakarya KHNISSI	1989641
Samuel KOUAKOU	2068795
Axelle LETIEU	2152361

Décembre 2024

Démarrage du système

Afin de lancer le projet, assurez vous de suivre les étapes suivantes:

Simulation

- Mettre la variable SIMULATION à 1 dans le fichier .env
- À la racine du repo, executer la commande source launch_station.sh
- Une fois la station au sol lancée, sélectionnez le bouton "Simulation".
- Puis dans l'interface qui s'affiche cliquer sur "Start ROS" pour lancer la simulation Gazebo.
- Patientez 1 minute le temps que le système complet soit lancé.

Robots Physiques

Assurez vous que la station au sol et les robots soient sur le même réseau WIFI puis suivre les étapes suivantes:

- À la racine du repo, executer "source launch_robot.sh 1" sur l'un des robots
- Puis, à la racine du repo sur l'autre robot, executer "source launch_robot.sh 2"
- Sur la station au sol, mettre la variable SIMULATION à 0 dans le fichier .env
- À la racine du repo, executer la commande source launch_station.sh

Tests des requis fonctionnels

Note: Avant de procéder avec les tests, il est nécessaire de :

- Cliquez sur le bouton "Get Started" Pour Commencer
- Choisissez un mode (Simulation/ Robot physique/Database/Help)
- Pour simulation : cliquer sur "Start ROS"

R.F.1 : Identifier le robot :

Sur l'interface utilisateur :

- Choisissez un mode (Simulation/ Robot physique)
- Cliquez sur le bouton "Identify Robot"

Test validé si :

Le robot effectue un son.

REMARQUE: Assurez-vous de vérifier le speaker

- Ouvrez les paramètres du robot.
- Sous les réglages audio, vérifiez que le speaker sélectionné est bien "USB..." ou "Digital output audio"
- Assurez-vous que le volume est supérieur à 0.
- Testez le speaker en lançant un son pour confirmer qu'il fonctionne correctement

R.F.2 : Lancer/Arrêter la mission :

Lancer la mission

Sur l'interface utilisateur :

- Cliquez sur le bouton "Start Mission"

Test validé si :

Le robot démarre l'exploration et fait des mouvements aléatoires

Arrêter la mission

Sur l'interface utilisateur :

- Cliquez sur le bouton "Stop Mission"

Test validé si :

Le robot arrête ses mouvements aléatoires

R.F.3 : Afficher le robot et son état :

Interface : Une fois le mode de mission choisi, suivant que la mission est lancée, arrêtée ou en attente, l'état du robot s'affiche en vert.

Test validé si :

Situations	Résultats attendus
Choisissez le mode (Simulation/Physique)	Le robot affiche l'état " <i>Waiting</i> " en attente
Cliquez sur le bouton "Start Mission"	Le robot affiche l'état " <i>Moving</i> " En mouvement
Cliquez sur le bouton "Stop Mission"	Le robot affiche l'état " <i>Stopped</i> " Arrêté

R.F.4 : Exploration autonome

Sur l'interface utilisateur :

- Cliquez sur le bouton "Start Mission"

Test validé si :

Le robot explore l'environnement sans intervention externe

Le robot modifie sa trajectoire lorsqu'un obstacle est détecté.

R.F.5 : Détection des obstacles :

Remarque: Pour tester ce requis, le robot doit être dans une mission

Sur l'interface utilisateur :

- Cliquez sur le bouton "Start Mission"

Test validé si :

Le robot détecte et évite les obstacles sans les toucher.

R.F.6 : Retour à la base :

Remarque: Pour tester ce requis, le robot doit être dans une mission

Sur l'interface utilisateur :

- Cliquez sur le bouton "Start Mission"

Laissez le robot explorer pendant un moment

- Cliquez sur le bouton "Return From Mission".

Test validé si :

Le robot revient à moins de 0,3m de sa position de départ

R.F.7 : Retour automatique à la base (batterie faible) :

Remarque: Pour tester ce requis, le robot doit être dans une mission

Sur l'interface utilisateur :

- Cliquez sur le bouton "Start Mission"

Test validé si :

Une fois la batterie à 30 %, le robot retourne automatiquement à sa base.

En simulation: La batterie perd 10% de sa capacité chaque 2mns.

Code couleurs de la batterie :

En vert : Niveau plein (100% - 60%)

En jaune : Niveau Moyen (50% - 30%)

En rouge : Niveau faible (30%- 0%)

R.F.8 : Production de la carte :

Sur l'interface utilisateur :

- Cliquez sur le bouton "Start Mission"

Test validé si :

Dès le début de la mission, la carte est affichée dans l'interface utilisateur.

R.F.9 : Position des robots :

Sur l'interface utilisateur :

- Cliquez sur le bouton "Start Mission"

Test validé si :

Dès le début de la mission, un point coloré avec le nom du robot (limo_105_x) est affichée sur la carte (x étant le numéro du robot, entre 1 et 4)

R.F.10 : Interface Web multi-appareils :

Remarque: Pour tester ce requis, vous devez disposer de deux appareils. Assurez-vous d'être connectés sur le même réseau.

Sur l'interface utilisateur :

- Cliquez sur le bouton "Start Mission"

Sur le deuxième appareil, entrez l'url suivant: <http://localhost:4200> et procéder aux mêmes étapes que celles décrites plus haut

Test validé si :

Une fois connecté sur le deuxième appareil, vous devriez avoir accès à la même interface utilisateur.

R.F.11 : Carte 3D en couleur :

Un terminal s'ouvre afin de compiler le projet pour la simulation.

- Cliquez sur le bouton "Switch 3D View" pour voir la carte en 3D.

Sur l'écran interactive dédié à la vue 3D:

- (Clique gauche + trainer) souris pour faire une translation et changer d'angle de vue
- (Clique droit + trainer souris) pour faire une rotation et changer angle de vue
- utiliser la roue de la souris pour zoomer et dézoomer

Test validé si :

La carte s'affiche en 3D.

R.F.13 : Détection d'élévation négative :

Remarque: Ce requis doit être testé sur le robot physique. Placez le robot en hauteur, par exemple sur une table

Placez vous dans le terminal du projet embarqué et sur une autre fenêtre celui du robot:

📁 project_ws

- Lancer le topic orbbec_camera avec ros2 launch orbbec_camera dabai.launch.py sur le terminal du robot avec la caméra installé après l'avoir colcon build et sourcé au préalable
- Lancer le châssis en se plaçant dans le terminal du projet embarqué avec ros2 launch limo_bringup limo_start.launch.py après avoir colcon build et sourcé

Une fois c'est fait et que les topics de la caméra (camera/depth/points) et cmd_vel sont actifs, lancer dans le terminal du projet ros2 run cliff_detector cliff_detection_node.py

Test validé si :

À quelques centimètres du bout de la table, le robot recule et s'arrête. Un message "Vide détecté" est affiché dans le terminal.

R.F.15 : Modes de contrôle des roues (simulation) :

Sur l'interface utilisateur :

- Cliquez sur le bouton bascule pour choisir le mode "Ackerman" ou "Contrôle différentiel"
- Cliquez sur le bouton "Start Mission" (4)

***Pour changer les modes de contrôle des roues une autre fois :**

- Sur le terminal gnome ouvert récemment “Ctrl + C”, pour mettre fin au processus.
- faire étape (1) jusqu'à (4)
- choisir les modes de contrôles de(s) robot(s)

Test validé si :

Le robot modifie son comportement en fonction des types de roues.

R.F.17 : Base de données :

Remarque: Assurez vous de bien démarrer le serveur

Sur l'interface utilisateur :

- Au moment de choisir un mode, cliquez sur “Database”
- Un fois la table des données chargée, cherchez une mission dans la barre de recherche. Le chargement de la table peut prendre 30 à 40s
- Filtrez les colonnes en mode croissant/décroissant en cliquant sur les icônes (▲ / ▼)
- Cliquez sur “Voir plus” pour visualiser la carte associée à la mission le cas échéant et les informations de log.

Test validé si :

Les données sont bien chargées et affichées.

R.F.18 : Visualisation de carte :

Sur l'interface utilisateur :

- Au moment de choisir un mode, cliquez sur “Database”
- Une fois les informations de la base de données chargées, cliquez sur “Voir Plus”, ensuite sur “Carte”

Test validé si :

La carte de la mission est affichée.