

# Procédures de Tests - Système d'exploration multi-robot

Ce document détaille les procédures de test pour chaque fonctionnalité du système d'exploration multi-robot, conformément au requis R.Q.2. Il est divisé en trois sections principales : (1) exécution des tests automatisés, (2) une présentation des tests unitaires, et (3) les procédures de test manuelles pour les fonctionnalités difficiles à tester automatiquement.

## 1. Exécution des Tests Automatisés

### Tests côté Client (Angular)

Pour exécuter les tests côté client :

```
cd client
./run-tests.sh --no-watch
```

Pour exécuter un test spécifique :

```
cd client
./run-tests.sh --include="src/app/services/mission.service.spec.ts"
```

### Tests côté Serveur (NestJS)

Pour exécuter les tests côté serveur :

```
cd server
npx jest
```

Pour exécuter un test spécifique :

```
cd server
npx jest src/mission/mission.service.spec.ts
```

## 2. Tests Unitaires

### Composants Client Testés

- **App Component** : Test de base du composant racine
- **Control Panel Component** : Tests des fonctions d'envoi de commandes aux robots
- **Dashboard Component** : Tests d'affichage des informations principales
- **Robot Status Component** : Tests d'affichage de l'état des robots
- **Sidebar Component** : Tests de la barre latérale de navigation

### Services Client Testés

- **Mission Service** : Tests CRUD pour les missions et historique
- **Robot Service** : Tests des commandes envoyées aux robots

- **WebSocket Service** : Tests de communication en temps réel

### Composants Client Récemment Testés

Les composants suivants ont été testés et disposent maintenant de tests unitaires :

- **Configuration Component** : Tests des fonctionnalités de configuration des modes de roue et autres paramètres
- **Confirmation Dialog Component** : Tests des interactions de confirmation (boutons Oui/Non)
- **Connected Clients Component** : Tests d’affichage des clients connectés
- **Help Dialog Component** : Tests de la boîte de dialogue d’aide
- **Map Component** : Tests de base du composant de carte
- **Mission History Component** : Tests d’affichage et d’interaction avec l’historique des missions
- **Mission Logs Component** : Tests d’affichage et de formatage des logs de mission
- **Navbar Component** : Tests du composant de navigation

### Composants Serveur Testés

- **App Controller** : Tests de base de l’API
- **Robot Controller** : Tests des endpoints de contrôle des robots
- **Robot Gateway** : Tests des connexions WebSocket
- **Mission Service** : Tests de gestion des missions
- **ROS Service** : Tests d’intégration avec ROS

### Tests Problématiques

Certains tests échouent en raison de dépendances externes ou spécifiques à l’environnement :

1. **ros.service.spec.ts** : Nécessite un environnement ROS fonctionnel. Pour tester cette partie, il est recommandé d’utiliser les procédures de test manuelles décrites dans la section 3.
2. **robot.controller.spec.ts** : Échoue avec l’erreur “librcl.so: cannot open shared object file: No such file or directory” car il dépend de la bibliothèque ROS2. Cette erreur est particulièrement fréquente dans les environnements WSL ou conteneurisés. Utiliser les procédures manuelles pour cette partie.
3. **Tests avec dépendances ROS2** : Tout test qui dépend du module `rclnodejs` peut échouer sur des environnements qui n’ont pas ROS2 correctement installé et configuré. Ces fonctionnalités doivent être testées manuellement selon les procédures de la section 3.

### 3. Procédures de Test Manuelles

Cette section décrit les procédures de test pour chaque requis fonctionnel, particulièrement pour les composants où les tests unitaires sont difficiles ou impossibles à mettre en œuvre.

#### R.F.1 - Identification de Robot

**Prérequis :** - Le système est démarré (station au sol et robots) - L'interface web est accessible

**Procédure :** 1. Accéder à l'interface web 2. Observer les robots connectés dans la liste des robots disponibles 3. Cliquer sur le bouton "Identifier" à côté d'un robot spécifique

**Résultat attendu :** - Le robot ciblé s'identifie physiquement (clignote une LED, émet un son) - Le feedback est visible dans l'interface utilisateur

#### R.F.2 - Commandes de Mission

**Prérequis :** - Le système est démarré (station au sol et robots) - Les robots sont connectés et visibles dans l'interface

**Procédure :** 1. Accéder à l'interface web 2. Cliquer sur le bouton "Lancer la mission" 3. Observer le comportement des robots 4. Cliquer sur le bouton "Terminer la mission"

**Résultat attendu :** - Les robots démarrent leur mission après "Lancer la mission" - Les robots s'arrêtent immédiatement après "Terminer la mission" - L'état des robots est mis à jour dans l'interface

#### R.F.3 - Affichage de l'État des Robots

**Prérequis :** - Le système est démarré - Au moins un robot est connecté

**Procédure :** 1. Accéder à l'interface web 2. Observer la section d'état des robots 3. Lancer une mission et observer les changements d'état 4. Terminer la mission et observer les changements d'état

**Résultat attendu :** - L'état de chaque robot (attente, en mission, etc.) est affiché clairement - Les mises à jour se font au moins à une fréquence de 1 Hz - Les transitions entre les états sont visibles lors des changements de mission

#### R.F.4 - Exploration Autonome

**Prérequis :** - Le système est démarré - Les robots sont en état d'attente

**Procédure :** 1. Lancer une mission via l'interface 2. Observer le comportement des robots pendant 5 minutes minimum

**Résultat attendu :** - Les robots se déplacent de manière autonome dans l'environnement - Les mouvements suivent l'algorithme d'exploration implémenté - Aucune intervention humaine n'est nécessaire pendant l'exploration

#### **R.F.5 - Évitement d'Obstacles**

**Prérequis :** - Le système est démarré - Une mission d'exploration est en cours - Des obstacles sont présents dans l'environnement

**Procédure :** 1. Observer le comportement des robots face aux obstacles statiques 2. Si possible, placer un obstacle mobile devant un robot en déplacement 3. Observer le comportement lorsque deux robots se croisent

**Résultat attendu :** - Les robots détectent et évitent les obstacles statiques - Les robots détectent et évitent les obstacles mobiles s'ils sont ajoutés - Les robots s'évitent mutuellement lorsqu'ils se croisent

#### **R.F.6 - Retour à la Base**

**Prérequis :** - Le système est démarré - Les robots ont effectué une exploration et sont éloignés de leur position initiale

**Procédure :** 1. Cliquer sur le bouton "Retour à la base" dans l'interface 2. Observer le déplacement des robots

**Résultat attendu :** - Les robots se dirigent vers leur position initiale - Les robots s'arrêtent à moins de 0,3 m de leur position de départ respective - L'état des robots dans l'interface indique qu'ils sont à la base

#### **R.F.7 - Gestion de la Batterie**

**Prérequis :** - Le système est démarré - La possibilité de simuler un niveau de batterie bas est disponible

**Procédure :** 1. Vérifier que le niveau de batterie est affiché dans l'interface 2. Simuler un niveau de batterie de 29% pour un robot (via interface ou simulation) 3. Observer le comportement du robot

**Résultat attendu :** - Le niveau de batterie est correctement affiché pour chaque robot - Lorsque le niveau descend sous 30%, le robot déclenche automatiquement un retour à la base - Une notification est affichée dans l'interface pour indiquer le retour automatique

#### **R.F.8 - Génération de Carte**

**Prérequis :** - Le système est démarré - Une mission d'exploration est en cours ou a été complétée

**Procédure :** 1. Observer la zone de carte dans l'interface utilisateur 2. Vérifier que la carte se génère au fur et à mesure de l'exploration 3. Explorer différentes

zones et observer l'évolution de la carte

**Résultat attendu :** - Une carte est affichée et mise à jour en continu dans l'interface - La carte représente de façon reconnaissable l'environnement exploré - Les objets majeurs (murs, gros obstacles) sont visibles sur la carte

#### **R.F.9 - Affichage de la Position des Robots sur la Carte**

**Prérequis :** - Le système est démarré - Une mission d'exploration est en cours - La carte est en cours de génération

**Procédure :** 1. Observer la position des robots sur la carte générée 2. Suivre visuellement le déplacement d'un robot spécifique 3. Comparer la position affichée avec la position réelle

**Résultat attendu :** - La position de chaque robot est clairement indiquée sur la carte - Les positions sont mises à jour en continu - Les positions affichées correspondent aux positions réelles des robots

#### **R.F.10 - Interface Web Multi-Appareils**

**Prérequis :** - Le système est démarré - Au moins deux appareils différents sont disponibles (PC, tablette, téléphone)

**Procédure :** 1. Accéder à l'interface web depuis un PC 2. Accéder simultanément à l'interface depuis un second appareil 3. Lancer une mission depuis l'appareil désigné comme contrôleur 4. Observer les deux interfaces pendant l'exécution de la mission

**Résultat attendu :** - L'interface s'affiche correctement sur les différents appareils - Les deux appareils peuvent visualiser les données en temps réel - Le contrôle est possible depuis l'appareil désigné - Le nombre de clients connectés est visible dans l'interface

#### **R.F.15 - Modes de Contrôle des Roues**

**Prérequis :** - Le système est démarré - La possibilité de changer de mode de contrôle est disponible

**Procédure :** 1. Accéder à la section de configuration du mode de contrôle 2. Sélectionner un premier mode (ex: Ackerman) 3. Lancer une mission et observer le comportement 4. Terminer la mission 5. Changer pour un second mode (ex: différentiel) 6. Lancer une nouvelle mission et observer les différences

**Résultat attendu :** - L'interface permet de sélectionner différents modes de contrôle - Le comportement des robots varie clairement selon le mode sélectionné - Les transitions entre modes sont fluides et sans erreur

### **R.F.17 - Base de Données des Missions**

**Prérequis :** - Le système est démarré - Plusieurs missions ont été effectuées précédemment

**Procédure :** 1. Accéder à la section d'historique des missions 2. Trier les missions selon différents critères (date, durée, etc.) 3. Sélectionner une mission spécifique et consulter ses détails

**Résultat attendu :** - La liste des missions précédentes est accessible et triable - Les attributs requis sont présents (date, heure, durée, robots, etc.) - Les détails complets d'une mission sont consultables

### **R.F.18 - Sauvegarde et Consultation des Cartes**

**Prérequis :** - Le système est démarré - Des missions avec génération de carte ont été effectuées

**Procédure :** 1. Effectuer une mission avec génération de carte 2. Accéder à l'historique des missions 3. Sélectionner une mission précédente 4. Ouvrir la carte sauvegardée de cette mission

**Résultat attendu :** - Les cartes sont sauvegardées à la fin de chaque mission - L'interface permet d'accéder aux cartes des missions précédentes - Les cartes chargées correspondent bien à l'environnement exploré lors de la mission

### **R.F.19 - Communication P2P entre Robots**

**Prérequis :** - Le système est démarré avec robots physiques - Les robots sont positionnés à des distances différentes de la base

**Procédure :** 1. Activer la fonctionnalité P2P via la commande "P2P" dans l'interface 2. Lancer une mission d'exploration 3. Observer les écrans tactiles des robots

**Résultat attendu :** - Les robots communiquent directement entre eux sans passer par la station - Le robot le plus éloigné affiche une icône spécifique sur son écran - La communication continue même si un robot s'éloigne temporairement de la station