

SORBONNE UNIVERSITÉ FACULTÉ DE SCIENCES
ANNÉE UNIVERSITAIRE 2019-2020 M2
PROJET INTÉGRATIF - MU5SPI10



Projet Intégratif

Rapport Ingénierie Système

Véhicule autonome : le TurtleBot 3 Burger

Encadrants :
Fabien VÉRITÉ
Sylvain ARGENTIERI
Ludovic SAINT-BAUZEL

Table des matières

Introduction	1
1 Architecture opérationnelle	2
1.1 Architecture environnement	2
1.2 Interfaces opérationnelles	3
1.3 Analyse des besoins opérationnels	3
1.4 Analyse et consolidation des contextes opérationnels	4
1.5 Analyse des cas d'utilisation	6
2 Architecture fonctionnelle	7
2.1 Analyse des exigences fonctionnelles	7
2.2 Analyse et architecture fonctionnelle	7
2.3 Architecture fonctionnelle statique	8
2.4 Interfaces fonctionnelles	8
2.5 Architecture fonctionnelle dynamique	10
2.6 Identification des modes de fonctionnement	10
3 Architecture organique	12
3.1 Analyse des exigences organiques	12
3.2 Analyse organique	14
3.3 Architecture organique statique	15
3.4 Architecture organique dynamique	18
3.5 Interfaces organiques	19
4 Éléments d'organisation du projet	21
4.1 Éléments d'organisation globaux	21
4.2 Éléments d'organisation spécifiques à l'équipe <i>Arène</i>	23
4.3 Éléments d'organisation spécifique à l'équipe <i>Robot</i>	25
Conclusion	27

Table des figures

1	Environnement du système.	2
2	Carte des flux MEI du système.	3
3	Cycle de vie du système.	5
4	Cas d'utilisation pour le contexte opérationnel « en service ».	6
5	Scénario opérationnel pour les cas d'utilisation de la figure 4.	6
6	Analyse fonctionnelle du système globale.	7
7	Architecture statique de la fonction du système global.	8
8	Architecture dynamique de la fonction du système globale et ses sous-fonctions.	10
9	Indicateur des modes de fonctionnement.	11
10	Architecture physique statique niveau globale A0.	15
11	Architecture organique statique niveau A1.	16
12	Architecture organique statique niveau A2.	16
13	Scénario organique du scénario opérationnel pour le système « Robot ».	18
14	Scénario organique pour scénario opérationnel pour le système « Arène ».	18
15	Interface organique pour le système « Robot ».	19
16	Interface organique pour le système « Arène ».	20
17	Diagramme de Gantt présentant l'organisation des tâches de l'ensemble du groupe.	21
18	Diagramme de Gantt présentant l'organisation des tâches de l'équipe <i>Arène</i>	23
19	Diagramme de Gantt présentant l'organisation des tâches de l'équipe <i>Robot</i>	25
20	Design de l'arène répondant aux exigences retenu.	27

Liste des tableaux

1	Description de l'environnement du système.	2
2	Analyse des besoins opérationnels pour chaque élément de l'environnement.	4
3	Description des exigences fonctionnelles pour le système global.	7
4	Flux de données entre les sous-fonctions de la fonction du système global.	9
5	Description des modes de fonctionnement.	11
6	Évaluation des risques.	12
7	Tableau des exigences organiques du système global.	13
8	Composants pour le système « Robot ».	14
9	Composants pour le système « Arène ».	15
10	Description des flux physiques.	17
11	Tableau des connexions du robot.	19
12	Tableau des connexions de l'arène.	20

Introduction

Depuis la rentrée 2019, le parcours Ingénierie de Systèmes Intelligents (ISI) fait partie d'une nouvelle mention de Master « Automatique, Robotique ». Dans ce cadre, de nouveaux enseignements autour de la robotique sont proposés en première année, en particulier une unité d'enseignement (UE) de « ROS (Robot Operating System) et robotique expérimentale » au 2nd semestre. Ce projet vise à travailler avec l'équipe enseignante sur la mise en place de la composante expérimentale de l'UE.

L'unité d'enseignement se centre autour d'un robot fabriqué par la société Robotis : le Turtle-Bot 3 Burger. Les travaux pratiques (TP) sont orientés vers les véhicules autonomes. L'UE étant nouvelle, tout le support du cours et des TP est à préparer. Nous intervenons sur la partie expérimentale et en particulier la création d'une arène dans laquelle le robot peut évoluer et tester les compétences des étudiants. L'équipe en charge du développement de ce projet éducatif est composée de 14 étudiants, divisés en deux groupes. Un premier groupe s'occupe de la création et la commandabilité d'une arène physique de type urbain et de sa modélisation virtuelle sur Gazebo, tandis que le second groupe se charge de l'implémentation des fonctionnalités de navigation et de planification du robot au sein de l'environnement.

Le système que nous étudions est composé de deux systèmes : le robot et l'arène. Ce rapport présente le travail de réflexion en amont du projet. Il expose synthétiquement les étapes de conception des systèmes du projet ainsi que leurs interactions avec les utilisateurs et leur environnement. Ce travail est aussi la mise en pratique d'un apprentissage sur l'ingénierie de systèmes. L'objectif premier est d'incorporer des notions d'organisation et de gestion de projet qui sont aujourd'hui indissociables des compétences d'ingénieur et de chercheur.

La suite de ce rapport est divisée en quatre sections. La première se concentre sur l'analyse opérationnelle du système. Cette analyse permet de modéliser les interactions externes du système et répond à la question "Pourquoi?". La seconde section résume l'analyse fonctionnelle du système. Elle vise à identifier les fonctions abstraites du système en répondant à la question "Quoi?". L'analyse fonctionnelle se focalise de manière abstraite sur l'intérieur du système. La troisième partie explique l'analyse organique du système. Elle s'intéresse aux composants concrets du système en répondant à la question "Comment?". L'analyse se fait de manière concrète cette fois sur l'intérieur du système. Enfin, nous présenterons les éléments d'organisation du projet en expliquant la répartition des tâches parmi les membres du groupe.

Dans la suite du rapport, comme nous étudions un système de système, lorsque nous parlerons de système global, nous ferons référence conjointement au système « Robot » et au système « Arène ». Nous utiliserons également pour les figures le code couleur suivant :

- Violet : environnement extérieur ;
- Bleu : acteur extérieur ;
- Rouge : système ;
- Orange : sous-système ;
- Vert : fonction du système ;
- Jaune : sous-fonction du système.

1 Architecture opérationnelle

Cette section présente l'architecture opérationnelle du système. La vision opérationnelle met en avant l'environnement d'un système et les interactions qu'il pourrait avoir. Cette vision permet aussi d'expliciter le contexte de création du projet et les besoins auxquels le système vise à répondre. Nous allons donc considérer les systèmes de notre système global comme des boîtes noires dans cette section.

1.1 Architecture environnement

Avant de rentrer dans les spécificités du système, nous allons d'abord déterminer l'environnement dans lequel il va opérer. La figure 1 montre l'agencement des différents éléments de l'environnement du système. Des informations complémentaires sont données dans le tableau 1.



FIGURE 1 – Environnement du système.

Élément de l'environnement	Description détaillée
Client	Sorbonne Université - UE Master SPI représentée par S.Argentieri, F. Vérité et L. Saint-Bauzel.
Concepteurs	Étudiants M2 spécialités SAR, II, I4 et ISSI.
Utilisateurs	Étudiants suivant l'UE ROS et professeurs.
Environnement	Salle TP fermée et éclairée par une lumière naturelle et des éclairages électriques.
Salle de stockage	<ul style="list-style-type: none"> Pour l'arène : démontée et rangée; Pour le robot : dans des boîtes plastiques (robot, matériel de recharge et pièces en surplus) dans une armoire fermée.

TABLE 1 – Description de l'environnement du système.

1.2 Interfaces opérationnelles

Grâce à l'identification de l'environnement du système, nous pouvons établir une cartographie des échanges (ou flux). Elle donne une vision globale des interactions entre l'environnement et lui-même, l'environnement et le système et enfin entre le système et lui-même. La figure 2 représente la carte des flux MEI (Matières, Énergie, Informations) pour notre système global, celle-ci inclue le robot et l'arène.

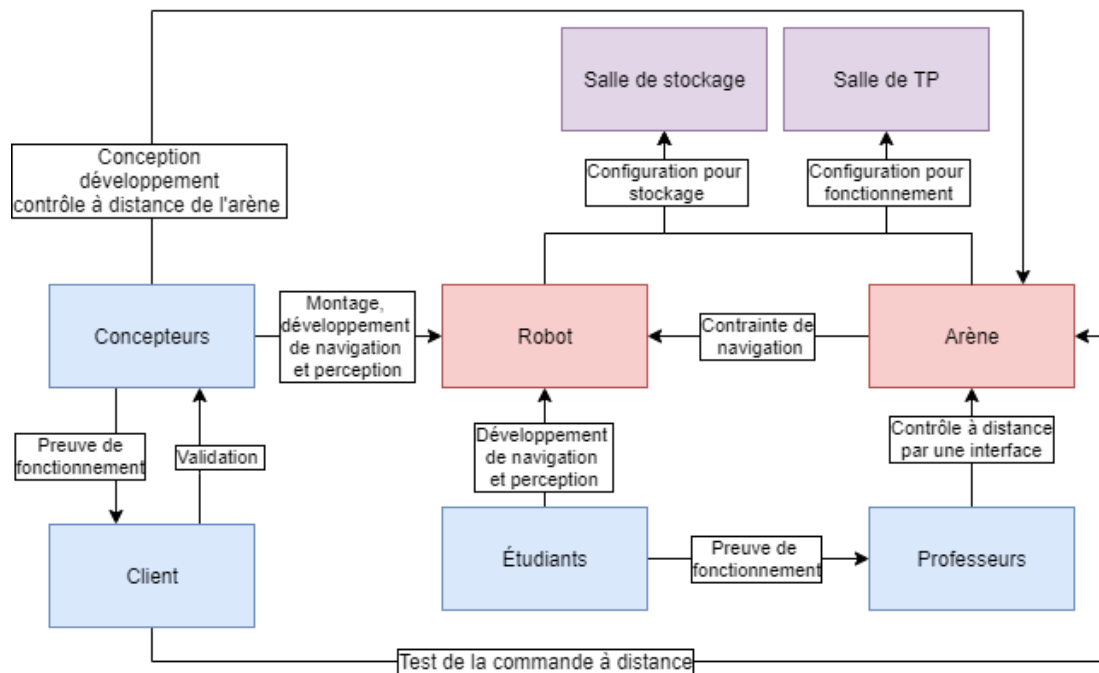


FIGURE 2 – Carte des flux MEI du système.

1.3 Analyse des besoins opérationnels

Les clients (professeurs encadrants) ont sollicité la réalisation d'une arène pour utiliser un robot mobile contrôlé avec l'outil informatique ROS Kinetic. Le but est de créer un espace de travail pour les étudiants. L'arène sera utilisée, en combinaison avec les robots, lors des séances de travaux pratiques, par les étudiants afin de tester les connaissances acquises au cours de l'enseignement. Puisque cette matière sera dispensée pendant plusieurs années, le système global pourra être enrichi au fur et à mesure du temps. Avec toutes ces informations, nous sommes en mesure d'identifier des contraintes imposées sur les éléments de l'environnement et du système qui sont résumées dans le tableau 2.

Élément	Service et contraintes	Capacité	Critères
Client	Solutions techniques imposées : <ul style="list-style-type: none"> • TurtleBot 3 Burger, • Codage en python, • Caméra Raspberry PI, • Modèle virtuel de l'arène, • Contrôle de l'arène via ROS. 		Doit être fonctionnel pour fin janvier.
Robot	Pilotage automatique.	<ul style="list-style-type: none"> • Suivre une ligne, • Respecter le code de la route, • Éviter des obstacles. 	Tests sur arène virtuelle et réelle.
Arène	<ul style="list-style-type: none"> • Démontable, • Modulable, • Sécurisée, • Pilotable sous ROS. 	<ul style="list-style-type: none"> • Préserver l'intégrité des robots, • Modifiable par l'équipe pédagogique. 	Tests avec robot virtuel et réel.
Professeurs	Contrôler l'arène.	<ul style="list-style-type: none"> • Contrôle du feu, • du passage à niveau, • de la limitation de vitesse, • du passage piéton. 	Tests sur l'arène virtuelle et réelle.
Étudiants	Développer des fonctions de contrôle.		
Salle de stockage	Espace disponible réduit.	Stocker l'arène après utilisation.	
Salle de TP	Espace disponible réduit.	Contenir l'arène lors de son utilisation.	

TABLE 2 – Analyse des besoins opérationnels pour chaque élément de l'environnement.

1.4 Analyse et consolidation des contextes opérationnels

Nous avons identifié deux contextes opérationnels pour notre système. Le premier est lorsque que le système est « en service » et donc utilisé par les étudiants et les professeurs. Le second contexte opérationnel est lorsque le système est « en stockage ». Grâce à ces deux contextes et les parties précédentes, nous sommes en mesure de faire le cycle de vie du système. Ce cycle est présenté dans la figure 3.

Les cas d'utilisation sont extraits des contextes opérationnels. Comme dit précédemment, nous en avons identifié deux : un lorsque que le système est « en service » et l'autre lorsqu'il est « en stockage ». Le deuxième contexte est moins intéressant à étudier : l'arène sera démontée et les robots rangés dans leur boîte et aucun acteur extérieur interagira avec le système. Nous nous intéressons donc plutôt au premier contexte et son cas d'utilisation.

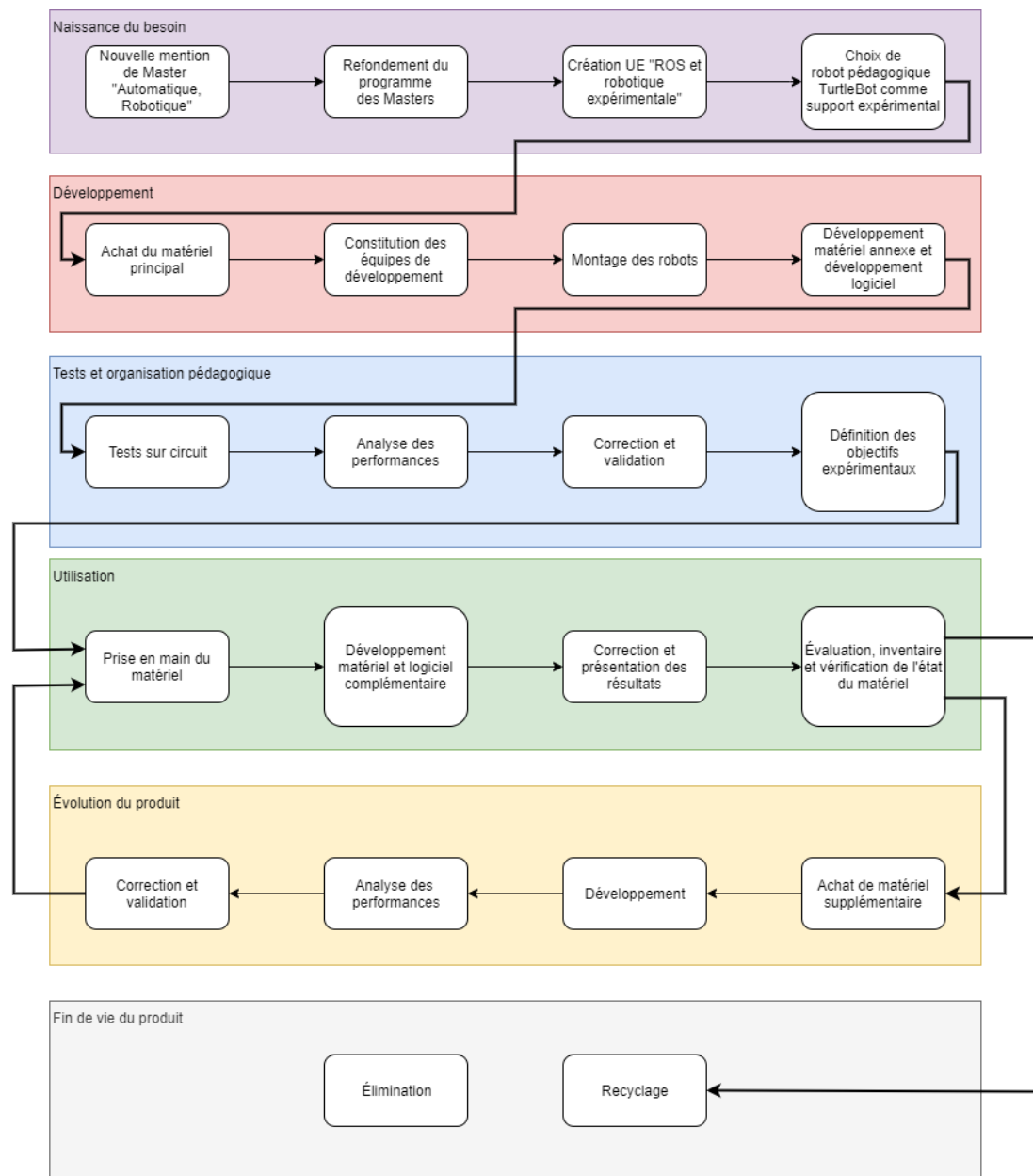


FIGURE 3 – Cycle de vie du système.

Lorsque le système est dans le contexte « en service », il est utilisé à la fois par les étudiants et les professeurs en salle de TP. La figure 4 montre les cas d'utilisation lors du contexte opérationnel « en service ». Le système global doit créer un espace de travail pour les étudiants et les professeurs. Le robot doit répondre aux commandes des étudiants et l'arène doit s'adapter en fonction des commandes des professeurs.

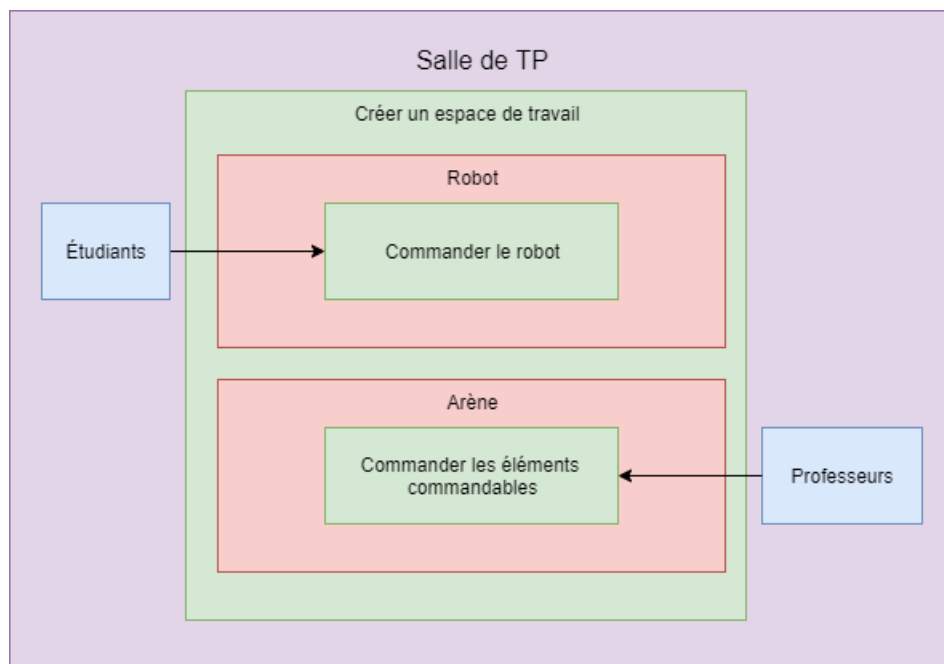


FIGURE 4 – Cas d'utilisation pour le contexte opérationnel « en service ».

1.5 Analyse des cas d'utilisation

Nous avons identifié précédemment, pour le contexte opérationnel « en service », les différents cas d'utilisation. Ici, nous analysons ces cas d'utilisation en créant un scénario opérationnel. La figure 5 montre un échange de flux MEI séquentiel entre les systèmes et l'environnement.

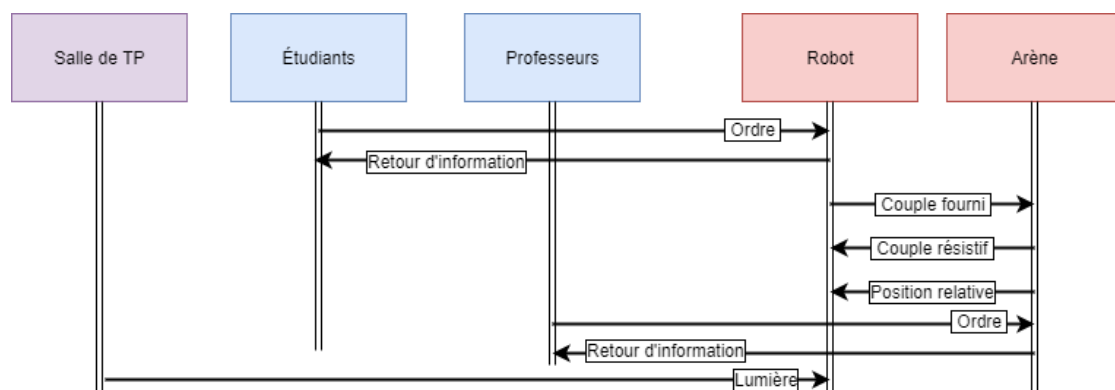


FIGURE 5 – Scénario opérationnel pour les cas d'utilisation de la figure 4.

2 Architecture fonctionnelle

Cette section présente l'architecture fonctionnelle du système, permettant de regarder plus en détail notre système. Nous cherchons dans cette section à identifier les fonctions du système global. Il s'agit de détailler les éléments qui composent le système global et faire abstraction des détails des deux sous-systèmes. En considérant son fonctionnement abstrait, nous pouvons déterminer les fonctionnalités qui lui sont nécessaires pour répondre aux besoins du projet.

2.1 Analyse des exigences fonctionnelles

Nous cherchons ici à traduire les besoins établis par le client en exigences fonctionnelles. Pour répondre aux besoins des professeurs encadrants, le système doit répondre à une seule exigence, présentée dans le tableau 3. Nous nous plaçons dans le scénario opérationnel de la figure 5 et ceci pour toute la suite du rapport.

Nom de la fonction	Critères de performance	Contexte
Créer un espace de travail pour tester les étudiants.	Avoir plusieurs niveaux de complexité.	Le système doit pouvoir s'adapter au niveau des étudiants en proposant des niveaux de difficultés différents.

TABLE 3 – Description des exigences fonctionnelles pour le système global.

2.2 Analyse et architecture fonctionnelle

Après avoir identifié les exigences fonctionnelles du système global, nous pouvons extraire du cas d'utilisation et du scénario opérationnel les fonctions et leurs sous-fonctions. Une fonction principale répond à l'unique exigence fonctionnelle. Elle est présentée dans la figure 6.

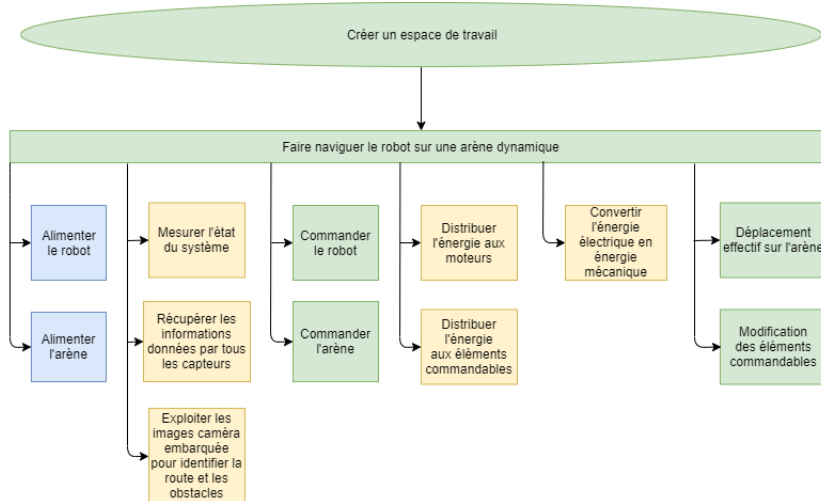


FIGURE 6 – Analyse fonctionnelle du système globale.

2.3 Architecture fonctionnelle statique

L'architecture fonctionnelle statique d'un système consiste à déterminer les interactions entre toutes les fonctions qui font que ce système remplisse bien sa mission. Ces fonctions sont de plusieurs natures, par exemple, il peut s'agir de transformations d'énergie (pour les moteurs), de traitement d'informations récupérées par des capteurs, etc. C'est la succession de toutes ces missions qui permet d'avoir un système fonctionnel. Bien évidemment, afin que le comportement de ce dernier soit celui attendu, l'ordre dans lequel réaliser ces missions est essentiel. La figure 7 représente le schéma de l'architecture fonctionnelle statique du système global.

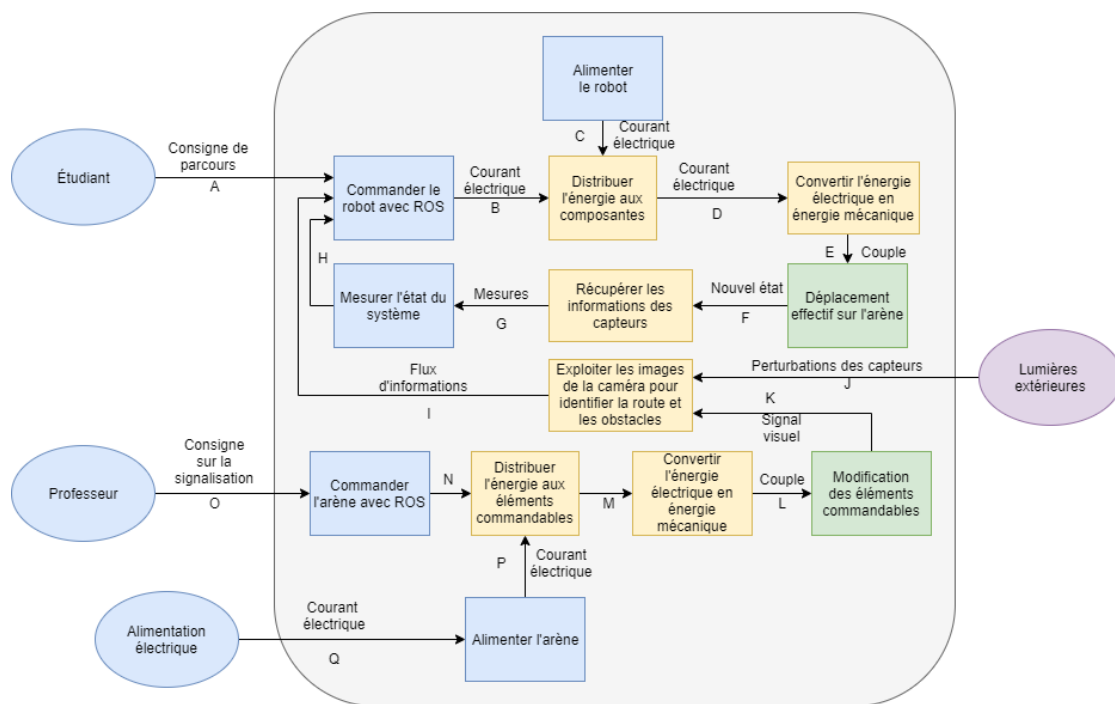


FIGURE 7 – Architecture statique de la fonction du système global.

2.4 Interfaces fonctionnelles

Pour consolider les interfaces fonctionnelles, les flux de données entre les sous-fonctions, qui ont été déterminées dans l'architecture statique du système, sont décrits dans le tableau 4. Les flux internes représentent les flux de données entre les sous-fonctions de la fonction principale tandis que les flux externes représentent les interactions avec l'environnement extérieur. Les flux non-recherchés impliquent un risque sur le système et peuvent être néfastes pour celui-ci.

Nom du flux	Descriptions des données	Type de flux
A - Étudiants → Piloter le robot avec ROS	Consigne de l'ordinateur étudiant vers le robot	Externe et recherché
O - Professeur → Piloter l'arène avec ROS	Commandes de l'ordinateur enseignant vers l'arène	Externe et recherché
Q - Alimentation → Alimenter l'arène	Énergie électrique nécessaire pour alimenter l'Arduino et la Raspberry	Externe et recherché
P - Alimenter l'arène → Distribuer de l'énergie aux éléments commandables	Énergie électrique	Interne et recherché
N - Piloter l'arène avec ROS → Distribuer de l'énergie aux éléments commandables	Signaux pour contrôler les éléments commandables	Interne et recherché
M - Distribuer de l'énergie aux éléments commandables → Convertir l'énergie électrique en énergie mécanique	Couple	Interne et recherché
L - Convertir l'énergie électrique en énergie mécanique → Modifier les éléments commandables	Couple	Interne et recherché
K - Modifier les éléments commandables → Exploiter les images de la caméra	Signal visuel	Interne et recherché
J - Lumières extérieures → Exploiter les images de la caméra	Perturbations du capteur	Externe et non-recherché
I - Exploiter les images de la caméra → Piloter le robot	Flux d'informations	Interne et recherché
B - Piloter le robot → Distribuer de l'énergie aux composants	Courant électrique	Interne et recherché
C - Alimenter le robot → Distribuer de l'énergie aux composants	Courant électrique	Interne et recherché
D - Distribuer de l'énergie aux moteurs → Convertir l'énergie électrique en énergie mécanique	Courant électrique	Interne et recherché
E - Convertir l'énergie électrique en énergie mécanique → Déplacer le robot dans l'arène	Couple	Interne et recherché
F - Déplacer le robot sur l'arène → Récupérer les informations des capteurs	Nouvel état du système	Interne et recherché
G - Récupérer les informations des capteurs → Mesurer l'état du système	Données mesurées par chaque capteur	Interne et recherché
H - Mesurer l'état du système → Piloter le robot	Données mesurées pour le système	Interne et recherché

TABLE 4 – Flux de données entre les sous-fonctions de la fonction du système global.

2.5 Architecture fonctionnelle dynamique

Une fois l'ensemble des flux identifiés et cartographiés, il est nécessaire d'apporter une vision dynamique au système en étudiant son comportement. Dans cette situation, les sous-fonctions apparaissent sous la forme de scénarios fonctionnels où les échanges entre les fonctions et avec l'environnement sont entièrement appréciés, comme le montre la figure 8.

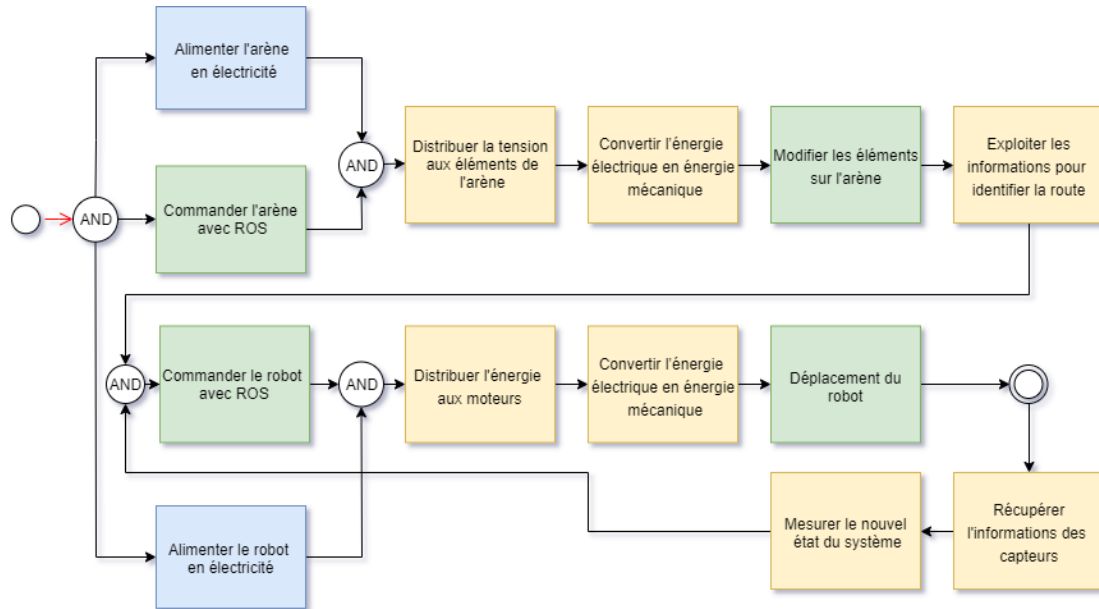


FIGURE 8 – Architecture dynamique de la fonction du système globale et ses sous-fonctions.

2.6 Identification des modes de fonctionnement

L'identification des modes de fonctionnement du système complète l'architecture fonctionnelle en représentant les différents états que peut prendre le système. Selon son état, le système a accès à différentes fonctions.

Les modes de fonctionnement du système couvrent différents contextes opérationnels du système. Le tableau 5 résume les différents modes de fonctionnement en explicitant les fonctions qui sont disponibles et les contextes couverts par ces modes. La figure 9 illustre ces différents modes.

Modes	Fonctions disponibles	Contextes couverts
Arrêt		<ul style="list-style-type: none"> • Demande d'arrêt, • Rechargement des batteries, • Panne du robot ou arène, • Arrêt d'urgence.
Maintenance	<ul style="list-style-type: none"> • Alimentation, • Navigation robot, • Commande arène. 	<ul style="list-style-type: none"> • Configuration de l'arène par le professeur, • Maintenance du robot, de l'arène ou des programmes sur ROS.
Marche autonome	<ul style="list-style-type: none"> • Alimentation, • Navigation robot, • Commande automatique arène. 	<ul style="list-style-type: none"> • Suivi de ligne, • Tests des algorithmes de navigation des étudiants, • Modification des éléments commandables.
Marche téléopérée	<ul style="list-style-type: none"> • Alimentation, • Navigation robot, • Interface commande arène. 	<ul style="list-style-type: none"> • Modification des éléments commandables, • Commande téléopérée du robot.
Défaut ou panne		<ul style="list-style-type: none"> • Coupure d'alimentation, • Panne électronique, • Capteur défectueux, • Panne d'un élément commandable, • Mauvaise configuration de ROS, • Sortie de l'arène.

TABLE 5 – Description des modes de fonctionnement.

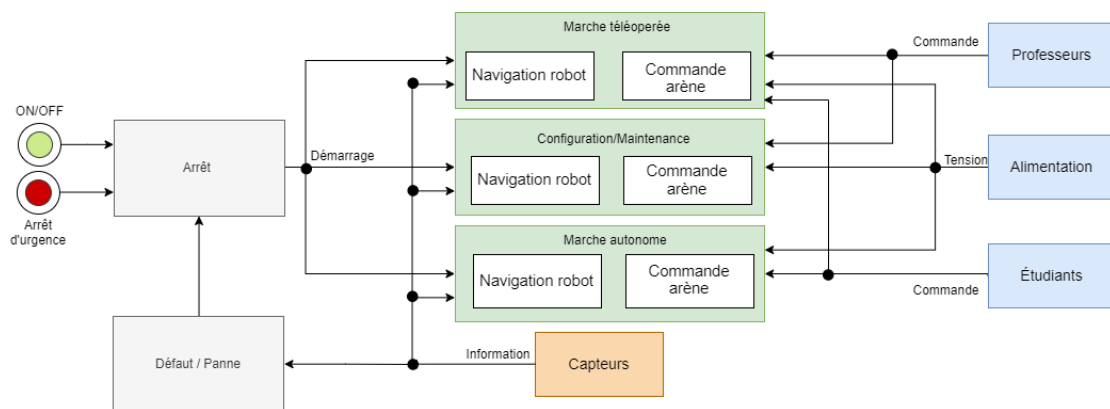


FIGURE 9 – Indicateur des modes de fonctionnement.

3 Architecture organique

Dans cette partie, nous nous concentrons sur la vision organique du système. Cette vision permet de définir les organes du système, c'est à dire ses composants. Elle permet également d'apporter des solutions techniques pour répondre aux besoins.

3.1 Analyse des exigences organiques

Grâce au travail précédent, il est possible d'identifier les risques et les conséquences que ceux-ci pourraient avoir sur le système. Trois risques principaux sont identifiés ainsi que les solutions qui y répondent. Ces informations sont décrites dans le tableau 6.

Risque	Niveau	Conséquences	Solutions
Sortie de route	Important	<ul style="list-style-type: none"> • Collision avec objets (décor ou obstacles), • Sortie d'arène, • Dégât matériel du robot. 	<ul style="list-style-type: none"> • Procédure d'arrêt d'urgence, • Pièces de rechanges (décor, obstacles et robot).
Sortie de l'arène	Important	<ul style="list-style-type: none"> • Dégradation de la caméra, • Dégradation du Lidar, • Dégradation des cartes. 	Bordure de sécurité autour de l'arène.
Collision avec des obstacles	Moyen	<ul style="list-style-type: none"> • Dégât matériel du robot, • Endommagement de l'obstacle. 	<ul style="list-style-type: none"> • Pièces de rechanges (obstacle et robot), • Position et mécanismes des obstacles adaptés aux collisions.

TABLE 6 – Évaluation des risques.

Les exigences organiques se présentent comme les performances à atteindre par le système ou un de ses composants afin de répondre aux besoins et aux contraintes. Le tableau 7 détaille les exigences organiques du système global.

Système	Caractéristique physique	Critères de performance	Contexte	Identifiant
Robot	Caméra	Détection des couleurs et des formes	Le robot doit <ul style="list-style-type: none"> respecter la signalisation, détecter la ligne à suivre. 	A
		Positionnement	La caméra doit être correctement positionnée pour voir la signalisation et la ligne à suivre sur la route.	B
	Lidar	Déterminer la position des obstacles autour du robot	Le robot doit s'arrêter si un objet lui barre la route.	C
	Électronique embarquée	Contrôler les déplacements du robot en fonction des données des capteurs	Le robot doit répondre aux commandes ROS.	D
Arène	Éléments commandables	Accessibles	Les éléments commandables doivent être accessibles facilement en cas de problème avec le robot.	E
	Sécurité de l'arène	Empêcher le robot de tomber	Le robot ne doit pas sortir de l'arène s'il sort de la route.	F
	Élément à suivre	Visible par la caméra du robot	Le robot doit pouvoir distinguer une ligne à suivre.	G
	Composants électroniques	Contrôler l'état des éléments commandables de l'arène	Les composants électroniques alimentés en électricité transmettent les commandes ROS aux éléments commandables.	H

TABLE 7 – Tableau des exigences organiques du système global.

3.2 Analyse organique

L'analyse organique permet d'identifier les composants des systèmes grâce aux décompositions fonctionnelles. Les composants des systèmes « Robot » et « Arène » sont présentés respectivement dans les tableaux 8 et 9. Elles répondent aux exigences organiques présentées dans le tableau 7. En plus de cela, nous avons pris en compte certains risques qui sont résumés dans le tableau 6. Enfin, pour que le niveau de difficulté puisse être adapté au niveau des étudiants, nous avons créé trois circuits en utilisant trois bandes adhésives de couleurs différentes.

Sous-système	Élément	Type	Réponse
TurtleBot 3 Burger	Lidar	Capteur	Besoin client
	OpenCR	<ul style="list-style-type: none"> • Carte contrôleur des moteurs, • Centrale inertielle. 	
	Raspberry Pi	Carte micro-contrôleur	
	Dynamixel XL430-W250	Servo Moteur	
	Batterie Li-ion	Alimentation	
Système de contrôle	ROS	Système informatique de contrôle programmable	<ul style="list-style-type: none"> • Exigence organique C • Exigence organique D
Caméra	Raspberry Pi v2	Capteur	<ul style="list-style-type: none"> • Besoin client • Exigence organique A
	Support caméra	Pièce imprimée 3D	Exigence organique B

TABLE 8 – Composants pour le système « Robot ».

Sous-système	Élément	Type	Réponse
Système de commande et d’interface	Raspberry Pi	Carte micro-contrôleur	Exigence organique H
	ROS	Système informatique de contrôle, programmable	
Alimentation	Câble d’alimentation carte Raspberry Pi	Câble électrique et transformateur électrique	
Composants électroniques	Carte Arduino	Câble électrique et transformateur électrique	
	Raspberry Pi	Carte micro-contrôleur	
Éléments commandables	Panneaux limitation de vitesse	Afficheurs LED 7 segments	Exigence organique E
	Feu tricolore	LED de couleur	
	Passage à niveau	Moteur à courant continu	
	Passage piéton		
Route	Lignes de circuit	Bandes adhésives de couleur	Exigence organique G

TABLE 9 – Composants pour le système « Arène ».

3.3 Architecture organique statique

Les flux fonctionnels et opérationnels peuvent être ramenés dans le contexte réel en définissant physiquement ces derniers comme le montrent les figures 10, 11 et 12. Ces figures sont complétées par la description des flux donnés dans le tableau 10

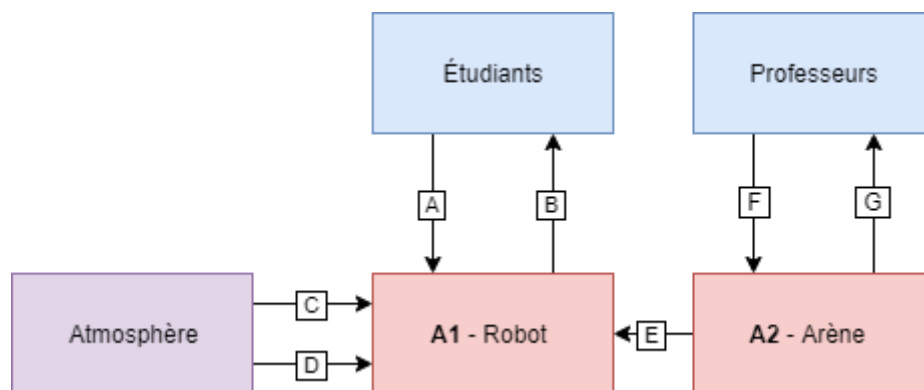


FIGURE 10 – Architecture physique statique niveau globale A0.

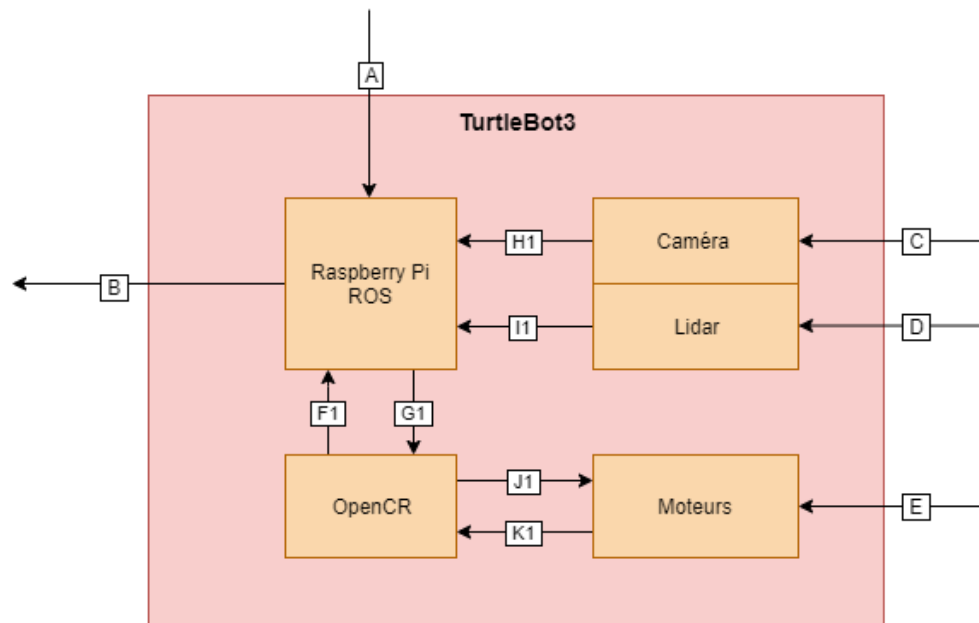


FIGURE 11 – Architecture organique statique niveau A1.

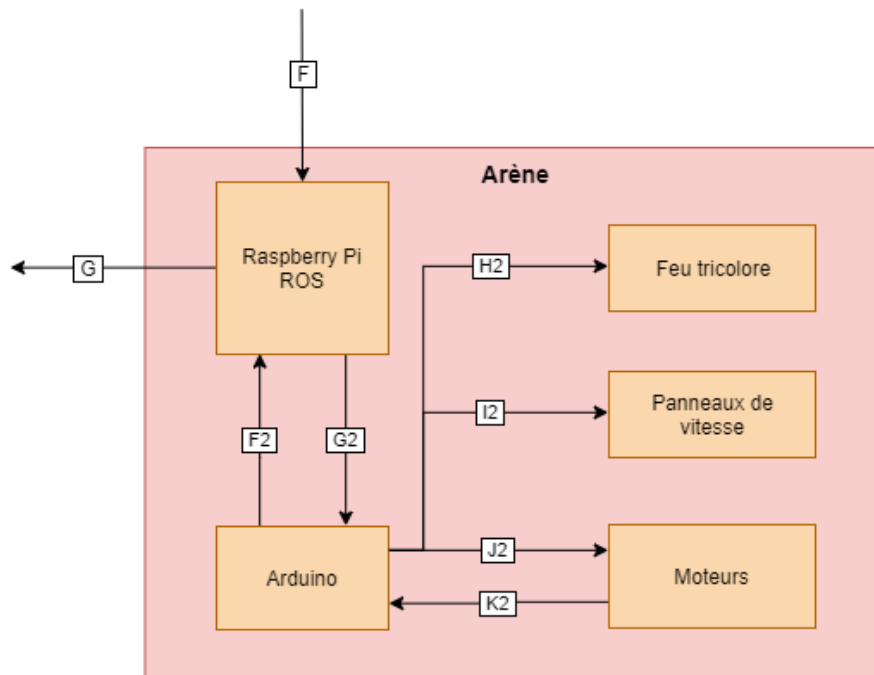


FIGURE 12 – Architecture organique statique niveau A2.

Nom du flux	Type de flux fonctionnel	Type de lien
A - Commandes	Étudiants → Robot	Réseau Wi-Fi
B - Retour information	Robot → Étudiants	Réseau Wi-Fi
C - Lumière	Atmosphère → Robot	Physique
D - Friction	Atmosphère → Robot	Physique
E - Signalisation	Arène → Robot	Physique
F - Commande des éléments commandables de l'arène	Professeurs → Arène	<ul style="list-style-type: none"> • Interface ROS, • Connexion Wi-Fi.
G - Retour d'information	Arène → Professeurs	<ul style="list-style-type: none"> • Interface ROS, • Connexion Wi-Fi.
F1 - Retour d'état moteurs	OpenCR → ROS	Câble
G1 - Commandes moteurs	OpenCR → ROS	Câble
H1 - Image numérique compressé	Caméra → ROS	Câble
I1 - Données de distances par rapport aux obstacles 360°	Lidar → ROS	Câble
J1 - Commande moteurs	OpenCR → Moteurs	Câble
K1 - État moteurs	Moteurs → OpenCR	Câble
F2 - Retour d'état éléments commandables	Arduino → ROS	Câble
G2 - Commandes éléments commandables	ROS → Arduino	Câble
H2 - Signal indiquant état	Arduino → Feu Tricolore	Câble
I2 - Signal indiquant vitesse à afficher	Arduino → Panneaux de vitesse	Câble
J2 - Commande moteurs	Arduino → Moteurs	Câble
K2 - État moteurs	Moteurs → Arduino	Câble

TABLE 10 – Description des flux physiques.

3.4 Architecture organique dynamique

En utilisant les éléments de l'architecture organique statique, il est possible de reprendre le scénario opérationnel (cf figure 5) pour obtenir le comportement organique du système au travers de scénario organique du système et de ses sous-systèmes synthétisés par les figures 13 et 14.

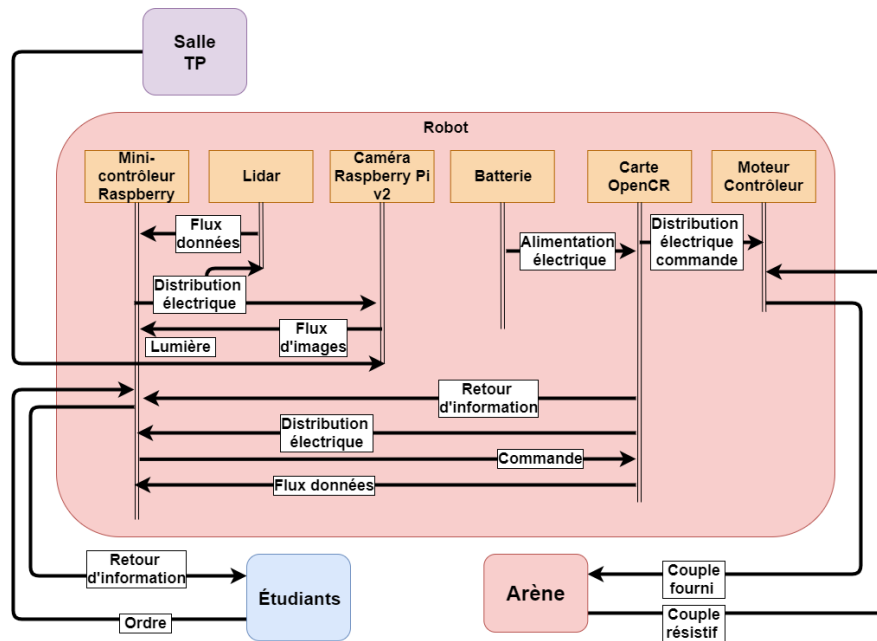


FIGURE 13 – Scénario organique du scénario opérationnel pour le système « Robot ».

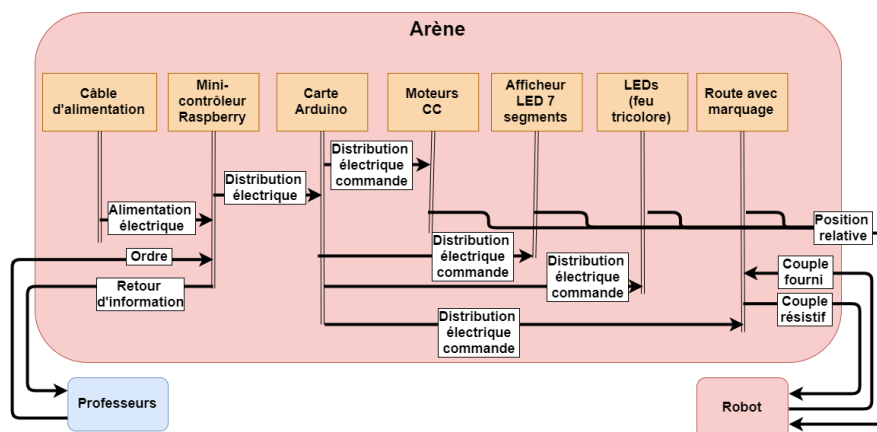


FIGURE 14 – Scénario organique pour scénario opérationnel pour le système « Arène ».

3.5 Interfaces organiques

Les figures 15 et 16 représentent respectivement les interfaces organiques du robot et de l'arène dans le contexte « en service ». Les tableaux 11 et 12 viennent compléter respectivement les interfaces organiques du robot et de l'arène.

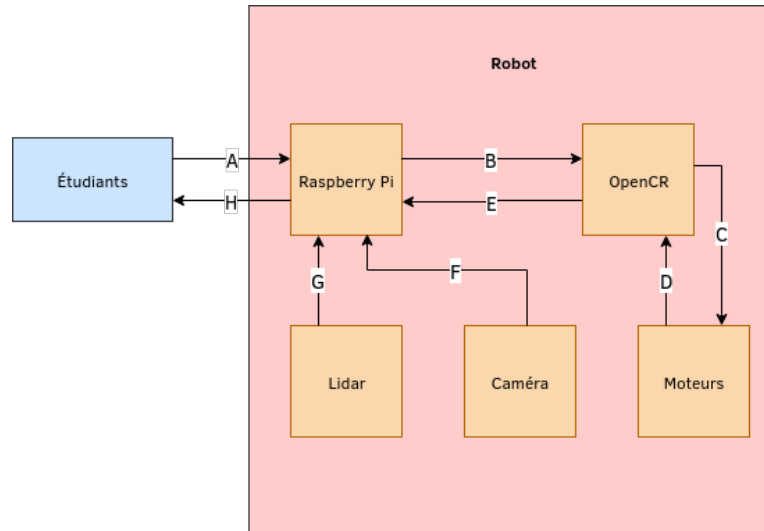


FIGURE 15 – Interface organique pour le système « Robot ».

Nom de l'interface	Type de flux fonctionnel	Type de lien	Type d'interface
A - Étudiants → Raspberry Pi	Commande pour le robot	Réseau Wi-Fi	Externe
B - Raspberry Pi → OpenCR	Commande des moteurs	Câble	Interne
C - OpenCR → Moteurs	Commande des moteurs	Câble	Interne
D - Moteurs → OpenCR	État des moteurs	Câble	Interne
E - OpenCR → Raspberry Pi	Retour d'état des moteurs	Câble	Interne
F - Caméra → Raspberry Pi	Images numériques compressées	Câble	Interne
G - Lidar → Raspberry Pi	Coupe horizontale à 360° de l'environnement	Câble	Interne
H - Raspberry Pi → Étudiants	Retour d'état des capteurs	Réseau Wi-Fi	Externe

TABLE 11 – Tableau des connexions du robot.

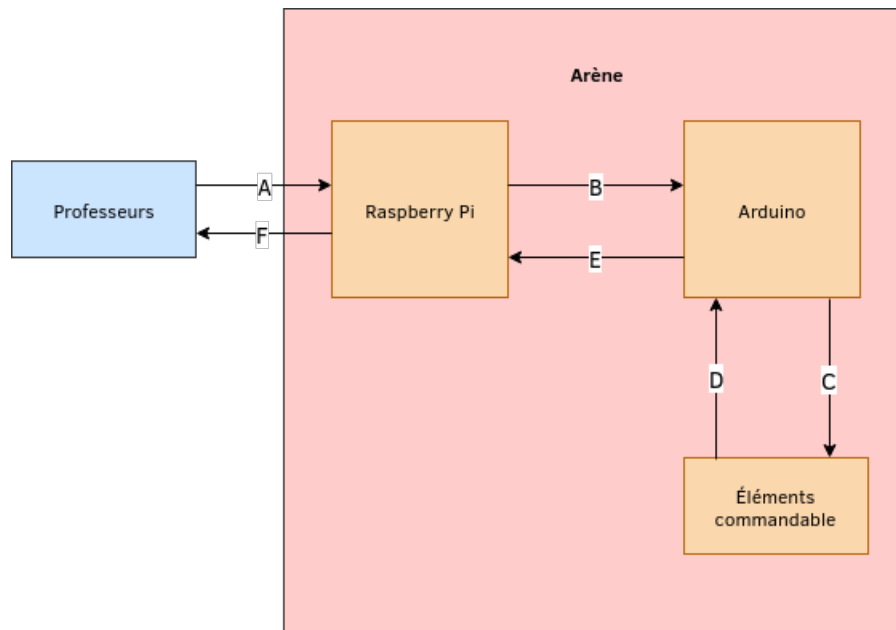


FIGURE 16 – Interface organique pour le système « Arène ».

Nom de l'interface	Type de flux fonctionnel	Type de lien	Type d'interface
A - Professeurs → Raspberry Pi	Commandes ROS	Réseau Wi-Fi	Externe
B - Raspberry Pi → Arduino	Commandes aux éléments commandables	Câble	Interne
C - Arduino → Éléments commandables	Signaux électriques	Câble	Interne
D - Éléments commandables → Arduino	Retour d'état	Câble	Interne
E - Arduino → Raspberry Pi	Retour d'état	Câble	Interne
F - Raspberry Pi → Professeurs	État des éléments commandable	Réseau Wi-Fi	Externe

TABLE 12 – Tableau des connexions de l'arène.

4 Éléments d'organisation du projet

Le système étant composé de deux systèmes (le robot et l'arène), nous avons créé une équipe de sept étudiants pour chaque élément. Deux types de tâches ont été identifiés, les tâches dites globales qui concernent les deux équipes et les tâches dites spécifiques qui ne concernent qu'une équipe.

Pour réaliser le planning prévisionnel, nous avons donc organisé les tâches en trois diagrammes de Gantt, séparant les éléments globaux concernant l'ensemble du groupe, des éléments spécifiques à l'équipe *Arène* ou à l'équipe *Robot*. Pour l'équipe en charge de l'arène, nous avons d'abord listé l'ensemble des éléments nécessaires à la création de l'arène. Ceci inclue les matériaux et la création d'algorithmes d'apprentissage permettant la reconnaissance des éléments de signalisation. Le même travail a été réalisé pour les tâches incombant à l'équipe responsable de la navigation du robot dans l'arène, notamment sur le travail d'algorithmie. Il s'agit, entre autre, le suivi de lignes et de trajectoires ainsi que la détection d'obstacles au moyen de capteurs présents sur le robot. Un dernier élément à prendre en compte est la sophistication du modèle par l'ajout d'une caméra.

La suite présente le détail de toutes les étapes nécessaires à la conduite du projet dans sa globalité.

4.1 Éléments d'organisation globaux

L'organisation du projet est basée sur une cohésion entre travail individuel et travail collectif. En effet, chacun doit réaliser des tâches individuelles, telles que le montage des robots, ainsi que divers apports à des tâches globales, telles que la préparation des documents à rendre en fin de projet. Le diagramme de Gantt en figure 17 présente le détail des tâches ainsi que leur organisation temporelle.

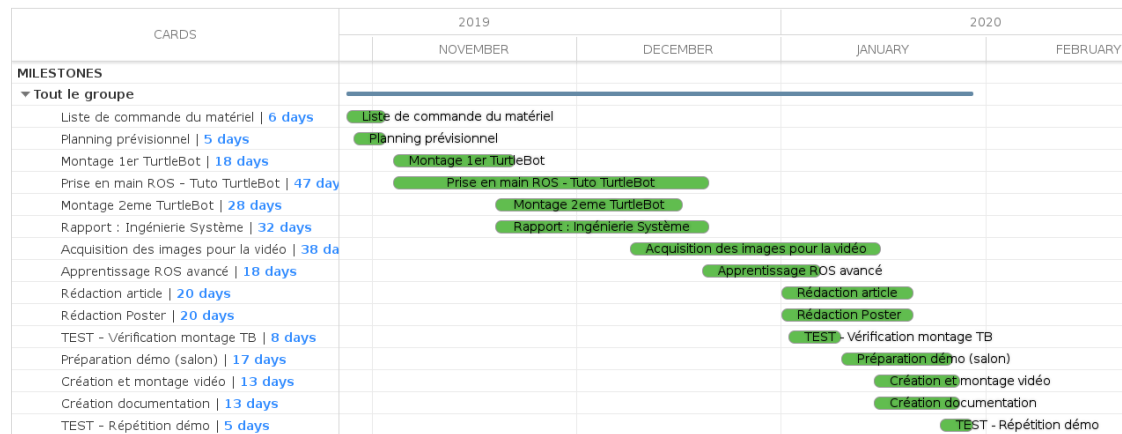


FIGURE 17 – Diagramme de Gantt présentant l'organisation des tâches de l'ensemble du groupe.

Liste matériel Pour pouvoir commander le matériel nécessaire, cette première tâche est indispensable. Elle aboutit à la création d'une liste composée des matériels à commander accompa-

gné de trois devis chiffrés. Ce travail doit être fait en respectant une liste spécifique de fournisseurs, ce qui peut compliquer la recherche de certains éléments.

Planning prévisionnel Réaliser le planning prévisionnel permet de réfléchir en amont à toutes les étapes qui devront être réalisées lors du projet et de poser des dates limites afin de tenir les délais. Il permet aussi d'expliquer et attribuer les tâches à chacun lors des réunions d'équipe. Une attention particulière doit être donnée à adapter le planning à tous les aspects du projet. En effet, la mise en évidence de risques et de détails des architectures physiques et fonctionnelles permettent notamment de ne pas oublier de tâches et de leur attribuer le temps nécessaire.

Montage du 1^{er} TurtleBot Puisqu'il est nécessaire d'avoir l'ensemble des robots montés pour le début des TP dispensés dès février 2020 aux étudiants de M1, il faut que chaque étudiant du l'équipe monte au moins deux robots TurtleBot 3 Burger. Cette première date limite permet d'assurer le montage rapide d'au moins 14 d'entre eux.

Prise en main de ROS Pour que tous puissent participer réellement au projet et acquérir de nouvelles compétences, chacun doit apprendre à prendre en main le logiciel ROS, indispensable dans beaucoup de projets de robotique.

Montage du 2^{ème} TurtleBot Tout comme pour le montage des 14 premiers robots, cette deuxième date limite, permet d'obtenir une réserve de 28 TurtleBots montés. Cela constitue un premier apport de matériel qui sera complété par d'autres robots montés par l'équipe pédagogique.

Rapport ingénierie système Le rapport d'ingénierie système permet de poser les bases et grandes lignes du projet. Ce travail permet de réaliser l'organisation en amont en prenant en compte le système dans les environnements auxquels il sera exposés. Il permet également de prévoir les risques et prévenir les potentiels problèmes liés aux différents systèmes et sous-systèmes.

Acquisition des images pour la vidéo Lors du montage des 2^{ème} TurtleBots, simulations virtuelles, montages de l'arène ou des tests dans l'arène physique, l'acquisition d'images et de vidéos est importante afin de réaliser la vidéo finale du projet.

Apprentissage ROS avancé En plus de la prise en main permettant quelques manipulations sur ROS, il sera indispensable de s'investir pour maîtriser davantage le logiciel. Cela permettra de modéliser et intégrer virtuellement l'arène sous ROS ainsi que de gérer l'ensemble des commandes avancées, réellement adaptées et efficaces pour contrôler le robot et l'arène connectée.

Article de projet Le compte rendu de projet sera rédigé sous la forme d'un article et présentera, le travail préliminaire concernant l'étude mécanique du système ainsi que la création de l'environnement et le développement des différents scénarios suivis par le robot. Les difficultés, et résultats finaux seront également présentés avec des pistes d'évolution pour aller plus loin.

Rédaction du poster de présentation Un poster de présentation sera réalisé afin d'accompagner et compléter la démonstration finale de projet.

TEST - Vérification montage TurtleBots Pour commander les robots à partir des algorithmes implémentés sous ROS, il est nécessaire de les connecter à un même réseau internet. Par ailleurs, plusieurs étapes d'intégration de logiciel sont nécessaires lors du montage des TurtleBots pour permettre la communication entre ROS et les robots, puis sur le robots entre la Raspberry PI et l'OpenCR. L'équipe doit s'assurer de la bonne installation de ces éléments sur chaque robot.

Préparation de la démonstration en salon Pour la présentation publique et finale du projet, il sera question de réaliser une démonstration des capacités du robot afin de montrer le travail accompli. Pour mettre l'accent sur le travail réalisé aussi bien en commande robotique que sur la complexité de développement de l'arène, la mise en place de différents scénarios sera élaborée.

Création et montage vidéo Les images acquises durant tout le projet, permettront la création d'une vidéo de présentation et de démonstration. Plus visuel qu'un rapport, elle permettra de mettre rapidement en valeur le travail réalisé sous une forme accessible.

Création documentation La création de documents permet l'amélioration du projet ainsi qu'une aide à la compréhension des étudiants ou personnes souhaitant réaliser un travail similaire.

TEST - Répétition démo La répétition de la démonstration permet de vérifier le bon fonctionnement de la commande robotique au sein de l'arène connectée pour l'ensemble des scénarios envisagés. Il s'agit de la dernière partie avant la présentation finale du projet.

4.2 Éléments d'organisation spécifiques à l'équipe Arène

L'équipe Arène est chargée de modéliser et créer l'environnement où évoluera le robot. Il s'agira d'une arène de type urbain. La réalisation de l'arène comprend l'assemblage simple des matériaux comme la connexion de tous les éléments électroniques commandables, comprenant notamment des obstacles mobiles. Le but est de contrôler l'arène depuis le logiciel ROS. Le diagramme de Gantt en figure 18 présente le détail des tâches pour l'équipe Arène ainsi que leur organisation temporelle.

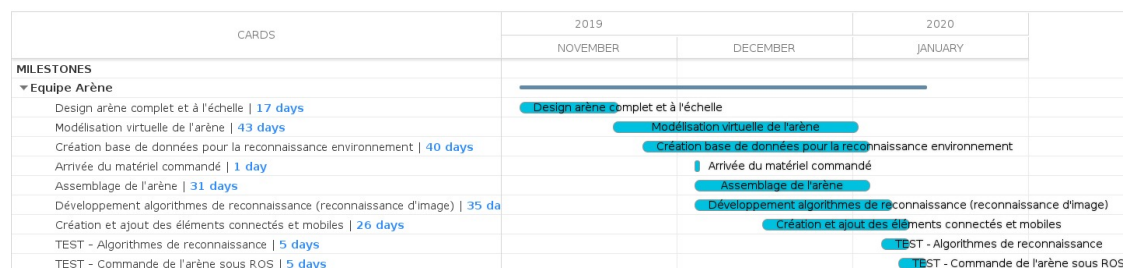


FIGURE 18 – Diagramme de Gantt présentant l'organisation des tâches de l'équipe Arène.

Design de l'arène complet et à l'échelle Au moyen des plans à main levée et aux idées d'enrichissement de l'environnement dans lequel évoluera le robot, il est nécessaire de réaliser un design complet et à l'échelle permettant de prévoir toute difficulté et d'attribuer le matériel

commandé aux différentes étapes de construction.

De cette façon, tous les câblages qui devront être réalisés seront optimisés et anticipés.

Modélisation virtuelle de l'arène En même temps que le design à l'échelle avance, il sera nécessaire de créer son équivalent virtuel en modélisation sous Gazébo, afin de simuler virtuellement dans l'arène. Après intégration de ce dernier dans ROS, il permettra de faire évoluer le robot dans son environnement virtuel pour les tests de commandes préliminaires, avant les tests physiques.

Création la base de données pour reconnaissance de l'environnement Pour que le robot puisse reconnaître les éléments importants de l'environnement dans lequel il évolue, il est nécessaire de récupérer des bases de données rassemblant des éléments d'environnement de type urbain notamment de panneaux de signalisation et de feu tricolore. Une fois l'arène enrichie, il sera possible de réaliser l'acquisition d'images de cette dernière depuis le robot afin d'obtenir des bases de données plus représentatives pour les étapes de reconnaissance et de prise de décisions qui suivront.

Arrivée du matériel commandé Cette date marque le début de nombreuses autres tâches nécessitant d'avoir le matériel commandé. Cela concerne essentiellement le travail de développement et construction de l'arène physique.

Assemblage de l'arène En suivant les recommandations et en faisant attention aux difficultés mises en lumière par le design complet à l'échelle et les autres études préliminaires, il sera essentiel de monter l'arène en commençant par les objets et obstacles non-commandables. Cette étape sera l'occasion d'évaluer les premiers déplacements du robot en suivi de ligne, dans un environnement sans obstacle.

Développement des algorithmes des reconnaissance Le développement des algorithmes de reconnaissance devront permettre de déterminer le type des obstacles détectés par la caméra du robot.

Ces algorithmes s'appuieront d'abord sur des bases de données existantes de signalisation et seront enrichies par les images récoltées par le robot comme évoqué précédemment.

Création et ajout des éléments électroniques et commandables Sur l'arène simple et assemblée, seront ajoutés les éléments électroniques (Arduino, Raspberry Pi, câbles, etc.) et les éléments commandables (piéton qui traverse, passage à niveau, feu tricolore, limitations de vitesse, etc.).

TEST - Algorithme de reconnaissance Pour approuver les algorithmes de détection d'obstacle, la première étape est l'implémentation des ces obstacles dans la simulation virtuelle. Une fois le comportement approuvé en arène virtuelle, face aux panneaux, feu et autres obstacles, le travail est testé dans l'arène physique.

TEST - Commande de l'arène sous ROS Pour que l'environnement urbain soit réaliste et interactif (piéton et passage à niveau mobile, feu qui change, etc.), il est nécessaire que les éléments commandables puissent l'être depuis la modélisation réalisée dans le module de ROS appelé Gazébo. Cette étape permet la vérification de leur implémentation et leur contrôlabilité depuis le logiciel dédié.

4.3 Éléments d'organisation spécifique à l'équipe *Robot*

L'équipe robot est chargée d'adapter la navigation, et donc le mouvement du robot, à l'arène construite ainsi qu'aux obstacles pouvant se placer sur son chemin. L'ensemble des commandes réalisées doit permettre de s'adapter aux trois scénarios de difficulté différente pour tester les compétences des étudiants de M1 qui suivront l'UE. Le diagramme de Gantt en figure 19 présente le détail des tâches pour l'équipe *Robot* ainsi que leur organisation temporelle.

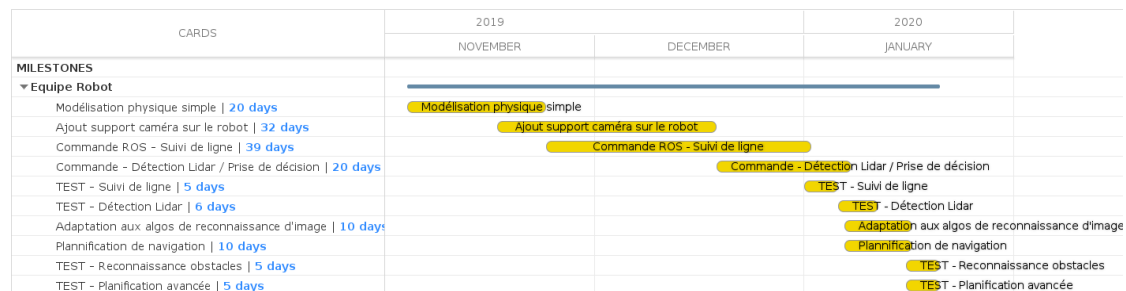


FIGURE 19 – Diagramme de Gantt présentant l'organisation des tâches de l'équipe *Robot*.

Modélisation physique simple Comprendre la théorie, c'est aussi important. Pour ne pas exécuter des programmes rédigés par le constructeur sans les comprendre et avoir une réelle approche scientifique, nous réaliserons une étape préliminaire. L'étude du système physique, plus précisément sa modélisation robotique nous permettra de comprendre le fonctionnement du robot et de ses déplacements ainsi que de ses limites.

Cette mise en équation théorique sera une étape importante pour la mise en place d'une commande efficace et de son optimisation.

Ajout d'un support caméra sur le robot Le robot étant muni d'un Lidar, la détection d'objet est possible mais pas celle de sa nature ou de la couleur d'un feu par exemple. Il est donc essentiel pour de nombreuses raisons, d'ajouter au TurtleBot une caméra. Pour cela il sera nécessaire de concevoir et mettre en place un support pour maintenir ce nouvel élément. L'élément sera réalisé par modélisation 3D et imprimé par impression 3D.

Commande ROS - Suivi de ligne La commande de navigation du robot est synthétisée sous ROS en plusieurs étapes. La première tâche à implémenter est le suivi de lignes. Où qu'il soit dans l'arène, le robot doit toujours se trouver sur une ligne de couleur correspondant au scénario qu'il exécute et ainsi toujours se trouver au centre de la route. Le suivi de ligne sera utilisé dans tous les scénarios, aussi bien un environnement neutre, sans obstacle, qu'en fonctionnalité de base dans un environnement enrichi. Il assurera une navigation sur la « route » dédiée.

Commande - Détection Lidar / Prise de décision Pour améliorer la commande et considérer d'avantage l'environnement, on utilise les propriétés du Lidar, qui permet de détecter des obstacles en utilisant un laser. Cette technologie est utilisable même en milieu non éclairé et peut donc soit compléter ou remplacer la caméra pour certaines tâches ou tronçons de route. En fonction de ce qui est détecté ou non, la commande doit adapter la navigation du robot ou enclencher l'arrêt d'urgence si nécessaire.

TEST - Suivi de ligne Cette étape permettra la validation du modèle de commande réalisé sur l'arène virtuelle et de l'implémenter dans l'environnement physique. Le plus important est de s'assurer que le robot ne sort jamais de la route et que la navigation se fait sans à-coups.

TEST - Détection Lidar Ces tests permettront la validation des algorithmes de détections d'obstacles et leur interprétation entraînant la prise de décision correspondante. Le robot doit notamment s'arrêter lors de la détection d'un obstacle (détectable par Lidar) et savoir suivre la route dans un environnement non éclairé (comme le tunnel), où la caméra ne détecte plus les lignes de couleur pour le suivi.

Adaptation aux algorithmes de reconnaissance d'image L'équipe Arène ayant développé des algorithmes permettant de détecter et identifier les panneaux de signalisation, feu tricolore et obstacles, une commande doit être développée en réponse aux différents éléments détectés. Il s'agit de l'implémentation de comportements en réponse à l'analyse de l'environnement (ralentir, freiner, etc.).

Planification de navigation Une dernière optimisation sera réalisée pour assurer une navigation la plus autonome possible. Il s'agit de donner une destination finale que le robot doit rejoindre de façon optimisée. Cette action sera implémentée dans l'arène commandée incluant tous les éléments et obstacles.

TEST - Reconnaissance obstacle Les tests doivent permettre de vérifier la navigation en sécurité du TurtleBot en prenant tous les obstacles en compte : piéton qui traverse, passage à niveau qui se ferme, feu qui change, etc.

TEST - Planification avancée La dernière vérification permet de valider un certain degré d'autonomie au robot où la seule information donnée par l'utilisateur pour le scénario est la destination. Le robot doit en suite trouver le chemin optimiser pour ce rendre au point imposé tout en s'adaptant à son environnement, autrement dit en évitant les obstacles et en respectant le code de la route qui s'applique dans l'arène.

Conclusion

L'étude du besoin nous a permis d'imaginer une arène qui respecte les différentes exigences du client et celles issues de notre réflexion. La figure 20 montre le design de l'arène réalisé avec le logiciel AutoCAD. Nous avons imaginé trois circuits de niveau différent.

- Le circuit de niveau 1, modélisé par la piste blanche, ne demande qu'un suivi de ligne et l'arrêt à un feu tricolore.
- Le circuit de niveau 2, modélisé par la piste bleue, fait le tour de l'arène et demande, en plus, de respecter les limitations de vitesse, de s'arrêter au passage piéton et au passage à niveau ainsi que de naviguer dans le tunnel en utilisant le Lidar. La zone de parking est disponible mais il n'est pas nécessaire de l'utiliser.
- Le circuit de niveau 3, modélisé par la piste rouge, demande une prise de décision aux intersections et la navigation en zone de travaux.

Plusieurs obstacles de l'arène sont commandables à distance via une interface ROS pour permettre aux encadrants des travaux pratiques de s'assurer du fonctionnement dynamique de l'arène. Les éléments commandables sont le feu tricolore, le passage piéton, le passage à niveau et les panneaux de vitesse.

Pour répondre au problème posé, nous avons également développé notre cahier des charges autour des fonctionnalités nécessaires à l'intégration du robot dans les différents scénarios. L'objectif est notamment d'assurer une navigation fluide, sans oscillations et en toute sécurité. La première étape pour atteindre cet objectif est le développement de l'algorithme de suivi de ligne. Une fois cet élément implémenté et testé, la commande a été complétée afin de permettre la détection des obstacles et signalisation à respecter lors de la navigation. Le travail a donc été d'adapter le comportement du robot à la détection d'informations pertinentes et identifiées comme dangers (passage à niveau, piéton sur la voie, etc.) ou points d'attention (limitation de vitesse, feu, etc.)

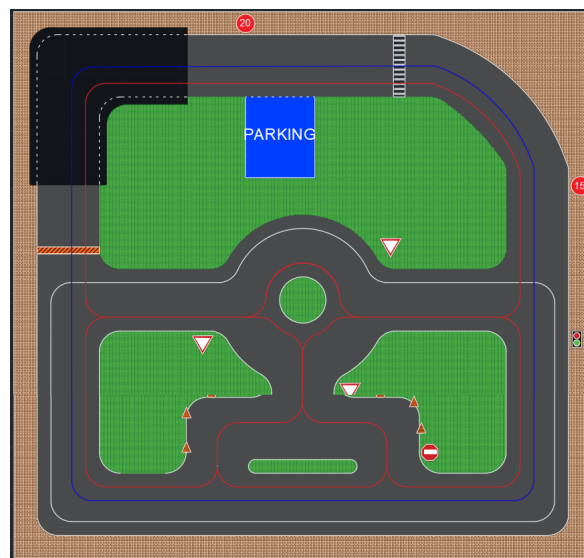


FIGURE 20 – Design de l'arène répondant aux exigences retenu.