

TR54
TP Projet :
Intersection Autonome

Professeur encadrant :
LOMBARD Alexandre
ABBAS-TURKI Abdeljalil

Élèves :
ROMET Pierre
CARRION Nicolas
BURGER Valentin
PANASSIM Hessou

Automne 2017

Contents

I	Introduction	2
1	Présentation	3
2	Objectif	4
II	Éléments à notre disposition	5
3	Le circuit	6
4	Politique de négociation	7
5	Stratégie de régulation	8
III	Travail Réalisé	9
6	Intersection	10
6.1	Politique de négociation	10
6.2	Stratégie de régulation	10
7	Communication sans file	11
7.1	Serveur	11
7.2	Client	11
8	Le suivi de ligne	12
IV	Analyse critique	13
9	Difficultés rencontrées	14
10	Solutions apportées	15
V	Conclusion	16

Part I

Introduction

Chapter 1

Présentation

Lors de ce TP Projet, il nous fut proposé d'implémenter des "Véhicules Guidés Autonome" (VGA), afin de mettre en œuvre un contrôle temps réel sans fil, destiné à prévenir le passage d'une intersection selon un protocole prédéfini.

Chapter 2

Objectif

L'objectif du projet est la réalisation d'une intersection autonome en utilisant les robots Léo EV3. Trois robots (ou plus) devront se déplacer sur un circuit en forme de huit et négocier leur droit de passage à l'intersection par communication sans fil afin d'éviter les collisions.

Part II

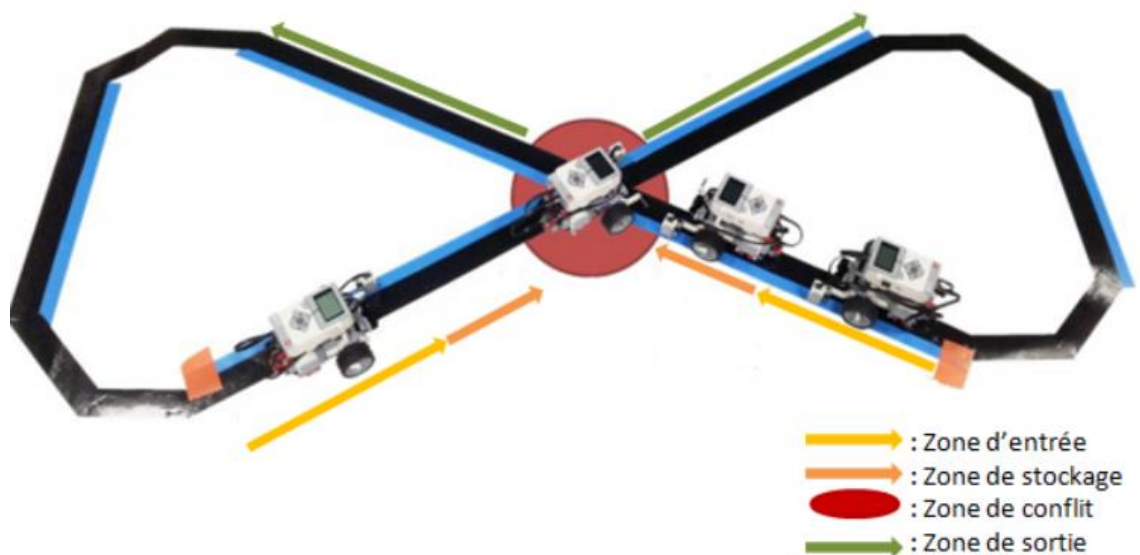
Éléments à notre disposition

Chapter 3

Le circuit

Le circuit mise à notre disposition est découpé en plusieurs zones:

- Zone d'entrée/zone de stockage (approche de l'intersection)
- Zone de conflit (où un seul robot peut être présent)
- Zone de sortie



Chapter 4

Politique de négociation

Afin de mettre en œuvre une régulation au sein de la zone d'intersection, quatre protocoles nous furent proposés afin d'administrer le droit de passage.

- Feux communicants: Programmation horaire en donnant la couleurs de feu à chaque voie (rouge/vert, rouge/rouge, vert/vert, vert/rouge). On peut pousser l'approche pour que le robots soit au courant du temps qui reste pour obtenir le vert et ainsi adapter sa vitesse en fonction.
- La négociation binaire: Les véhicules arrivent et demande l'accès au serveur pour son passage. Puis, si le passage n'est pas occupé, le robot obtient le droit de passage, sinon il est enregistré et il attend l'autorisation, qu'il aura une fois la zone libre. Enfin, lorsque le robot à fini de traverser la zone, il envoie un acquittement, afin de notifier le serveur que la zone de conflit est libre.
- Synchronisation de vitesse: A l'approche de l'intersection (zone d'entrée) un robot envoie une requête de passage au serveur d'intersection. De son côté le serveur construit une séquence de passage ordonnée à partir des requêtes reçues et la retransmet à l'ensemble des robots concernés. La liste contient l'ensemble des robots ayant émis une requête pour franchir l'intersection, leur position et leur vitesse. Tous les robots sont autorisés à franchir l'intersection, mais ils doivent synchroniser leur vitesse avec ceux qui les précèdent dans la séquence. Par exemple, si un robot est en position 3 dans la séquence, il ne pourra traverser qu'après que le robot en position 2 soit sorti de la séquence. Pour ce faire, chaque robot, une fois dans la séquence de passage, doit émettre régulièrement auprès du serveur des messages pour l'informer de sa nouvelle position et sa nouvelle vitesse. Ces informations sont retransmises à tous les robots par la séquence de passage. Tant qu'un robot n'est pas dans la séquence de passage, il doit s'arrêter avant l'intersection.
- Synchronisation par réservation (AIM): Cette méthode fut développé par l'université du Texas, elle consiste à se que le véhicule estime l'horaire d'occupation de la zone de conflit afin de pouvoir demander une réservation. Ainsi, si l'horaire est libre, le serveur l'accepte, sinon il lui refuse. Enfin, en cas de refus, le véhicule devra retarder son horaire.

Chapter 5

Stratégie de régulation

Concernant la construction de la séquence de passage, devant permettre le franchissement de la zone de conflit, deux stratégies nous furent proposées:

- FIFO (first-in first-out) : Le premier arrivé à l'intersection est le premier à en sortir, dans le cadre de cette politique, un soin particulier devra être apporté à la prévention des situations d'inter-blocage
- Batch : lorsqu'un robot se présente à l'intersection, il est ajouté à la séquence de passage, si et uniquement si, l'une des deux conditions suivantes est remplie :
 - La séquence de passage actuelle est vide
 - Le dernier robot dans la séquence de passage provient de la même voie que le robot actuel
 - Le dernier robot dans la séquence de passage provient de l'autre voie, mais le délai entre l'ajout de ce dernier robot et du robot courant est supérieur à un seuil dT (par exemple 5 secondes)

Part III

Travail Réalisé

Chapter 6

Intersection

6.1 Politique de négociation

La politique de négociation utilisé pour notre projet, afin de gérer le droit de passage des robots au sein de la zone de conflit, est la politique de synchronisation des vitesses. Bien que cette politique ait été imposé par le sujet, elle offre une méthodologie plus complète que les autres politiques présenté plus tôt dans ce chapitre.

En effet, on retrouve la gestion via un serveur centralisé, mais qui offre une mécanique plus sophistiqué, en utilisant plusieurs zones de contrôle. De plus, une interaction inter-robot afin d'adapter la vitesse déplacement entre les robots au sein des zones offre une méthodologie plus élégante.

6.2 Stratégie de régulation

Pour ce projet, deux stratégies de régulation nous furent proposées. Nous avons choisi d'implémenter la politique de "BATCH", étant, après analyse la stratégie la plus performante.

Le principal inconvénient de la stratégie "FIFO", réside dans son utilisation de l'horloge. En effet, dans le cas où deux robots se présentent quasiment au même instant pour franchir la zone d'intersection et que l'un des messages transmis au serveur soit perdu, compromis, ou envoyé avec un retard, alors il y aurait un risque que le deuxième robot s'engage dans la zone d'intersection alors que le premier robot est déjà en train de franchir cette zone.

De plus, cette stratégie ne donne pas de priorité en fonction du nombre de robot sur une section. Ainsi, il peut se produire un cas d'engorgement sur une zone, mais que cette dernière ne puisse pas évacuer ces occupants.

La politique de "BATCH" à l'avantage de pallier à ces deux problèmes, étant certes plus complexe mais couvrant plus précisément les situations de conflits ou d'inter-blocages.

Chapter 7

Communication sans fil

Comme nous l'avons vu dans le chapitre précédant, la communication sans fil est au coeur même de notre projet. En effet, s'est elle qui nous permet de mettre en place notre politique négociation et notre stratégie de régulation. La communication sans fil se base sur une connexion Wifi, client/serveur, avec un échange de données évoluant autour d'une synchronisation centralisée des données.

La communication sans fil est donc divisé en deux parties:

7.1 Partie Client (robot Lejos EV3)

Comme nous l'avons vu précédemment, si nous voulons pouvoir mettre en place une politique négociation et une stratégie de régulation, il va nous falloir échanger des données clefs entre les robots, et le serveur.

Pour cela, nous avons mis en place une communication basé sur le protocole TCP/IP, devant nous permettre de certifier l'envoi des données. Le protocole TCP étant orienté-connexion par le principe de la poignée de main (handshake), il établit au préalable une connexion entre les processus avant de débiter la transmission.

Nous acheminons ainsi du robot au serveur, une structure de données regroupant:

- L'identifiant du robot envoyant les données.
- La vitesse actuelle du robot en déplacement.
- La zone dans laquelle le robot évolue (zone entrée, stockage, conflit et sortie).
- La virage du quel le robot provient, afin de permettre au serveur de déterminer de quel coté le robot se présente, afin de franchir la zone de conflit.

Les données étant transmise sous la forme d'une chaine de caractères afin de pouvoir être envoyé en une seul séquence de contrôle; nous l'avons basé sur le modèle suivant:

	bit de start =	Données	bit de stop
Robot			
Serveur	=	ID + Vitesse + Position + Virage	\n

Ainsi, la même séquence de contrôle pourra être utiliser pour transmettre les informations du serveur au robot.

Il ne lui reste plus qu'à être traité par le serveur, pour utiliser les données.

7.2 Serveur

Chapter 8

Le suivi de ligne

Part IV

Analyse critique

Chapter 9

Difficultés rencontrées

Chapter 10

Solutions apportées

Part V

Conclusion