

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique  
Université Abderrahmane Mira  
Faculté de la Technologie



Département d'Automatique, Télécommunication et d'Electronique

## Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master

Filière : Automatique

Spécialité : Automatique et système

### Thème

Utilisation de l'intelligence des plantes en commande automatique  
d'un robot mobile

**Préparé par :**

- Mlle. Djerrah Malika
- Mr. Bellouze Zineddine

**Dirigé par :**

Dr. TIGHZERT Lyes

**Examiné par :**

Mr HADDAR Hocine

Mme MEZZAH Samia

Année universitaire : 2022/2023

# Remerciements

*Nous exprimons notre gratitude envers Dieu tout-puissant pour nous avoir accordé la force et la patience nécessaires à l'accomplissement de ce travail.*

*Nous tenons particulièrement à remercier le Dr. Lyes IGHZERT pour l'honneur qu'il nous a fait en acceptant de nous encadrer et pour avoir partagé généreusement ses compétences scientifiques, ses idées et ses conseils précieux tout au long de ce travail.*

*Nous souhaitons également adresser nos remerciements aux membres du jury, M Haddar Hocine et Mme MEZZAH Samia pour avoir consacré leur attention à notre travail et l'avoir évalué.*

# Dédicace

*Je tiens à exprimer ma gratitude envers Dieu le tout-puissant pour m'avoir accordé le courage et la patience nécessaires pour mener à bien ce travail.*

*Je dédie ce travail à mon père et ma mère, à mon frère, à toute ma famille, à tous mes amis qui m'ont soutenu tout au long de ce parcours.*

*Je souhaite également exprimer ma reconnaissance à mon encadreur pour son précieux soutien et ses conseils tout au long de ce travail.*

*Djerrah Malika*

# Dédicace

*Je tiens à exprimer ma gratitude envers Dieu le tout-puissant pour m'avoir accordé le courage et la patience nécessaires pour mener à bien ce travail.*

*Je dédie ce travail à mon père et ma mère, à deux mes frères, à toute ma famille, à tous mes amis qui m'ont soutenu tout au long de ce parcours.*

*Je souhaite également exprimer ma reconnaissance à mon encadreur pour son précieux soutien et ses conseils tout au long de ce travail.*

**Bellouze Zineddine**

## Sommaires

<b>L'intelligence des plantes et l'algorithme APGA.....</b>	<b>3</b>
<b>chapitreI Intelligence des plantes et l'algorithme APGA .....</b>	<b>4</b>
I.1 Introduction.....	4
I.2Optimisation heuristique et méta heuristique.....	4
I.3Les algorithmes inspirés de l'intelligence des plantes .....	5
I.3.1 Algorithme invasive weed optimisation (Iwo).....	6
I.3.2 Artificial Plant Growth Algorithm APGA.....	10
I.4Analyse de performance de l'algorithme APGA .....	11
I.4.1 Analyse de performance.....	11
I.5Conclusion.....	18
<b>chapitreII Modélisation cinématique et dynamique du robot.....</b>	<b>20</b>
II.1 Introduction .....	20
II.2 Les robots mobiles.....	20
II.2.1 Définition .....	20
II.2.2 Classification des robots mobiles.....	20
II.2.3 L'architecture logicielle des robots mobiles .....	21
II.2.4 Les organes de sécurité .....	21
II.2.5 Traitement des informations et gestion des tâches.....	21
II.2.6 La navigation.....	22
II.2.7 La modélisation dans l'environnement.....	22
II.3 Les robots mobiles à roues .....	23
II.3.1 Types des robots mobiles à roues .....	23
II.4 Modélisation d'un robot mobile de type unicycle .....	24
II.4.1 Contraintes cinématiques du robot à entraînement différentiel .....	27
II.4.2 Modélisation cinématique .....	28
II.4.3 Modélisation dynamique.....	29
Conclusion.....	35
<b>chapitreIII Commande et planification de trajectoire .....</b>	<b>37</b>
III.1 Introduction.....	37

III.2	La commande PID .....	37
III.2.1	Description d'un contrôleur PID.....	37
III.2.2	L'équation d'un PID .....	37
III.3	L'implémentation sur Simulink .....	38
III.3.1	Les paramètres de contrôleur PID utilisées.....	45
III.3.2	Commande en vue de poursuite d'une trajectoire circulaire.....	45
III.4	Planification de trajectoire et évitement d'obstacles par APGA.....	48
III.4.1	Planifications de la trajectoire avec APGA.....	49
III.4.2	Résultats et discussion .....	49
III.4.3	Interprétations des résultats.....	55
III.5	Conclusion .....	56

## Listes des figures

Figure I:1les deux catégories des métaheuristiques.....	5
Figure I:2 : Une plante annuelle d'été.....	7
Figure I:3: Une plant annuelles d'hiver.....	7
Figure I:4: La fonction sphère en 2 D.....	12
Figure I:5: la fonction scheffel en 2 D.....	12
Figure I:6: la fonction Rosenbrock en 2 D.....	13
Figure I:7: la fonction Rastrigin en 2 D.....	14
Figure I:8: application d'APGA sur sphère.....	15
Figure I:9: application d'APGA sur Schwefel.....	15
Figure I:10: application d'APGA sur Rosenbrock.....	16
Figure I:11: application d'APGA sur Rastrigin.....	16
Figure II:1: architecture logiciel d'un robot mobile.....	21
Figure II:2: trajectoire d'un robot.....	22
Figure II:3: robot mobile unicycle.....	23
Figure II:4 : robot tricycle.....	24
Figure II:5: robot voiture.....	24
Figure II:6: robot unicycle.....	25
Figure II:7: Robot mobile de type unicycle dans les repères fixe.....	26
Figure II:8: Robot Plateforme.....	28
Figure III:1: modèle cinématique sous Simulink.....	39
Figure III:2: modèle dynamique sous Simulink.....	40
Figure III:3: modèle de moteur droit sous Simulink.....	41
Figure III:4: modèle de moteur gauche sous Simulink.....	41
Figure III:5: la commande globale de robot.....	44
Figure III:6: la trajectoire du robot obtenu.....	45
Figure III:7: la tension appliquée au deux moteurs.....	46
Figure III:8: Le courant appliqué au deux moteurs.....	46
Figure III:9: Le couple moteur appliqué au deux moteurs.....	47
Figure III:10: Les vitesses des deux roues.....	47
Figure III:11 l'erreur de poursuite.....	48
Figure III:12la fonction fitness.....	49
Figure III:13 la trajectoire planifiée.....	50
Figure III:14 la trajectoire planifié.....	50
Figure III:15 La trajectoire planifiée avec APGA.....	51
Figure III:16les tensions des deux moteurs.....	52
Figure III:17le courant des deux moteurs.....	53
Figure III:18le couple moteur des deux moteurs.....	53
Figure III:19la vitesse des deux roues.....	54
Figure III:20 l'erreur de poursuite.....	54

## Liste des tableaux

Tableau I-1 les paramètres utilisées dans l'algorithme APGA.....	14
--	----

## Liste des symboles

APGA: Artificial Plant Growth Algorithm

IWO: Algorithm invasive weed optimisation

DDMR: Differential Drive Mobile Robot

L : inductance de moteur

V : la vitesse linéaire

W : la vitesse angulaire

D : la dimension de l'espace recherché

A : le point milieu de l'axe des roues.

2R : représente le diamètre de la roue du robot.

2L : représente la largeur du robot.

$\omega_r$  et  $\omega_l$  : représentent respectivement la vitesse de rotation de la roue droite et de la roue gauche.

$\theta$  : est l'angle d'orientation du robot.

$\sigma$  : l'écart type



## **Contexte :**

Les robots mobiles sont des robots autonomes capables de se déplacer dans leur environnement. Ils utilisent des systèmes de locomotion tels que des roues, des chenilles ou des jambes pour se déplacer sur différents types de terrains. Ces robots offrent de nouvelles possibilités dans des domaines tels que l'intelligence artificielle et l'apprentissage automatique. Ils sont utilisés comme assistants pour les personnes et utilisent l'apprentissage automatique pour reconnaître les besoins et les émotions des individus. Parmi les types de robots mobiles, les robots mobiles unicycle sont utilisés dans des applications spécifiques où leur agilité et leur compacité sont avantageuses, comme les entrepôts, les centres de distribution, les usines, les environnements hospitaliers et les environnements de recherche. Ils peuvent effectuer des tâches de navigation, de surveillance, de transport, de collecte de données et de manipulation d'objets. Cependant, le contrôle de ces robots nécessite des méthodes sophistiquées, ce qui constitue un défi.

## **Problématique :**

L'une des problématiques essentielles dans le domaine de la robotique mobile est l'exécution de trajectoires. La planification consiste à trouver le chemin le plus court et le plus sûr pour atteindre un objectif à partir des données disponibles sur l'environnement. La commande d'un système non holonome pour le déplacer d'une configuration de départ à une configuration d'arrivée est complexe. La poursuite de trajectoire implique de guider le robot à travers des points intermédiaires pour atteindre sa destination finale dans un délai prédéfini.

## **Solution proposée :**

Ce mémoire se concentre sur la navigation d'un robot mobile unicycle qui évite les obstacles. L'objectif est d'élaborer une méthode de planification de trajectoire pour ce robot en utilisant l'intelligence des plantes. La méthode proposée utilise l'algorithme APGA (Artificial Plant Growth Algorithm), inspiré de la nature, pour la planification de trajectoire. La commande utilisée pour la poursuite de trajectoire est basée sur le contrôleur PID. L'objectif principal est d'utiliser l'intelligence des plantes pour développer

une méthode de planification de trajectoire pour le robot mobile unicycle. Les objectifs spécifiques comprennent la modélisation complète du robot, l'implémentation d'une loi de commande simple et l'utilisation de l'algorithme APGA pour la planification de trajectoire avec évitement d'obstacles.

L'organisation de ce mémoire se divise en trois chapitres :

### **Chapitre I : L'intelligence des plantes et l'algorithme APGA**

Ce chapitre se concentre sur l'intelligence des plantes et explore son potentiel pour résoudre différents types de problèmes. Il présente l'algorithme APGA (Artificial Plant Growth Algorithm) qui s'inspire de la nature et des mécanismes de croissance des plantes. L'algorithme APGA sera étudié en détail, en mettant en évidence ses principes fondamentaux et son fonctionnement.

### **Chapitre II : Modélisation cinématique et dynamique du robot**

Le deuxième chapitre offre une présentation générale des robots mobiles, en mettant l'accent sur les robots mobiles à deux roues. Une attention particulière est portée à la modélisation mathématique du robot mobile unicycle, qui permet de comprendre son comportement et ses contraintes lors de la planification et de la poursuite de trajectoire.

### **Chapitre III : Commande et planification de trajectoire des robots mobiles**

Le troisième chapitre se concentre sur la commande et la planification de trajectoire des robots mobiles. Ce chapitre explore les principes de ces commandes, leur mise en œuvre et leur utilisation pour la poursuite de trajectoire des robots mobiles unicycle. En résumé, ce travail vise à développer une méthode de planification de trajectoire pour les robots mobiles en utilisant l'intelligence des plantes, avec l'algorithme APGA.

# **Chapitre I**

## **L'intelligence des plantes et l'algorithme APGA**

## **chapitreI Intelligence des plantes et l'algorithme APGA**

### **I.1 Introduction**

Les "méta heuristiques" sont particulièrement utiles pour aborder des problèmes considérés comme difficiles. Ces problèmes sont souvent caractérisés par des données incomplètes, incertaines ou bruitées, ou par une capacité de calcul limitée. Les méta heuristiques ont été appliqués avec succès dans une variété de domaines en raison de leur adaptabilité à toute situation pouvant être formalisée comme un problème d'optimisation.

### **I.2 Optimisation heuristique et méta heuristique**

Les heuristiques sont des méthodes de résolution purement algorithmiques qui permettent d'obtenir des solutions à n'importe quel problème décisionnel rapidement.

Une heuristique est une stratégie de bon sens pour se déplacer intelligemment dans l'espace des solutions, afin d'obtenir une solution approchée, la meilleure possible, dans un délai de temps raisonnable.

Deux types d'heuristiques sont principalement utilisées : les heuristiques de construction (par exemple les méthodes gloutonnes), qui construisent itérativement une solution, et les heuristiques de descente, qui à partir d'une solution donnée cherchent un optimum local.

Des heuristiques plus poussées ont été mises au point et ont données naissance à une nouvelle famille d'algorithmes : les métaheuristiques.

Le but d'une méta-heuristique, est de réussir à trouver un optimum global. Pour cela, l'idée est à la fois de parcourir l'espace de recherche, et d'explorer les zones qui paraissent prometteuses ; mais sans être « piégé » par un optimum local.

Les métaheuristiques sont souvent inspirées de processus naturels et sont de plus en plus hybridées avec d'autres méthodes de recherche opérationnelle.

Les principales métaheuristiques utilisées pour résoudre des problèmes à variables discrètes sont :

-Le recuit simulé, qui explore l'espace de recherche tout en acceptant de dégrader sa solution afin de sortir des optima locaux. Tout au long du processus, le recuit va de moins en moins accepter ces dégradations, ce qui va le faire converger vers un optimum (que l'on espère) global.

-La recherche avec tabous, qui a l'inverse du recuit simulé est déterministe et a une notion de mémoire. Le choix du meilleur voisin d'une solution pousse l'algorithme

trouver les optima locaux ; et comme l'exploration de l'espace de recherche est effectué en limitant le voisinage de la solution en rendant « tabous » certains mouvements, l'algorithme doit théoriquement visiter l'optimum global.

- Les algorithmes évolutionnaires, issus de la théorie de l'évolution de Darwin, qui manipulent plusieurs solutions en même temps, et qui en les combinant forment de nouvelles solutions. Le fait d'avoir une population de solutions facilite l'exploration de l'espace de recherche, et les meilleures solutions seront favorisées pour participer à la création de nouvelles solutions ce qui aura pour effet de favoriser les combinaisons des « bonnes caractéristiques », et donc de trouver un optimum global [1].

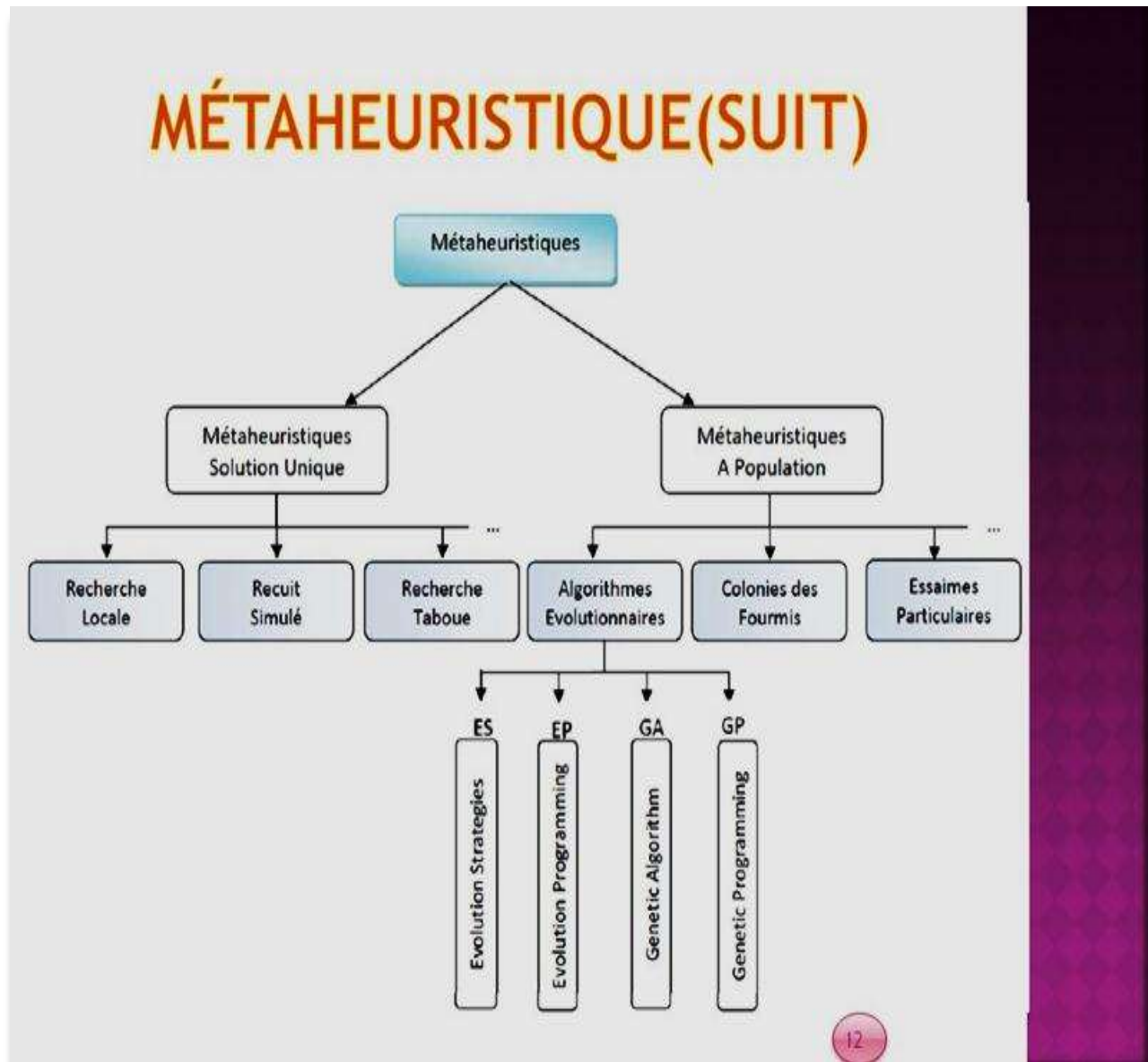


Figure I:1 les deux catégories des métaheuristiques [2]

### I.3 Les algorithmes inspirés de l'intelligence des plantes

Les algorithmes inspirés de l'intelligence des plantes visent à imiter les stratégies adaptatives et les comportements des plantes pour résoudre divers problèmes complexes.

Ces algorithmes tirent parti de principes tels que l'auto-organisation, la coopération et l'adaptation pour créer des systèmes robustes et efficaces.

Au cours des dernières décennies, les chercheurs ont développé de nombreux algorithmes inspirés de la nature afin de tenter de trouver les meilleures solutions pour

Divers problèmes d'optimisation par exemple :

L'optimisation des herbes envahissantes (IWO), inspirée du phénomène de colonisation des herbes envahissantes dans la nature, qui a été proposée par Mehrabian et Lucas (2006), l'optimisation de la croissance des plantes artificielles inspirée dans la nature ceci a inspiré par infeng Liu et ses collègues de l'université de Dalian, en Chine, il a proposé un nouvel algorithme appelé l'algorithme APGA.

### **I.3.1 Algorithme invasive weed optimisation (Iwo)**

#### **I.3.1.1 Définition**

L'algorithme IWO (Invasive Weed Optimization) est un algorithme d'optimisation qui s'inspire du comportement des mauvaises herbes pour résoudre des problèmes d'optimisation. Il a été proposé pour la première fois par Mehrabian et Lucas en 2006.

L'idée de base de l'algorithme IWO est de considérer une population de mauvaises herbes qui se propagent dans un terrain donné. Chaque mauvaise herbe représente une solution candidate à un problème d'optimisation, et la propagation des mauvaises herbes correspond à l'exploration de l'espace de recherche [3].

#### **I.3.1.2 Concepts de base d'Iwo**

L'algorithme IWO est un algorithme qui s'inspire du comportement des mauvaises herbes.

##### **I.3.1.2.1 Définition**

Les mauvaises herbes ont été appelées plantes qui poussent dans de mauvais endroits s'ils sont les plantes qui sont en concurrence avec des plantes que nous voulons. Ils sont en concurrence pour l'eau, la lumière du soleil et des éléments nutritifs dans le sol.

Certains cas, leurs semences contaminent les cultures de semences et réduisent sa valeur.

Certaines mauvaises herbes ont la capacité de modifier la chimie du sol mais de façon subtile avec des effets néfastes sur les espèces de plantes et les animaux qui partagent avec elles le même environnement [4].

#### **I.3.1.2.2 Les catégories des herbes envahissant**

On trouve 3 catégories qui sont :

#### **I.3.1.2.3 Les plantes annuelles**

Contient deux types, les annuelles d'été et les annuelles d'hiver [5].

#### **I.3.1.2.4 Les plantes annuelles d'été**

Les plantes annuelles d'été germent au printemps et, produisent des fleurs et des graines et meurent la même année. Ils ont une propriété de pousser très rapidement et de produire beaucoup de grains.



Figure I:2 : Une plante annuelle d'été [6]



### I.3.1.2.5 Les annuelles d'hiver

Les plantes annuelles d'hiver germent le début novembre et passent l'hiver à l'état de rosettes, puis quand le printemps arrive, elles poussent très rapidement, fleurissent et produisent des graines ensuite, meurent à la fin de la saison.



Figure I:3: Une plant annuelles d'hiver [6]

### I.3.1.3 Les étapes de formulation de l'algorithme IWO

Pour formuler l'algorithme IWO il existe 6 étapes à suivre :

**Dans l'étape d'initialisation** : une population initiale aléatoire est dispersée sur l'espace de recherche (dimensions N).

**Dans l'étape d'évaluation**, en suivant la valeur de la fonction de performance, chaque plante, à une position donnée dans l'espace, va croître d'une manière différente par rapport aux autres plantes, la façon de croissance d'une plante reflète une évaluation de sa qualité (solution). La plante optimale (vecteur solution) est celle qui réduit au minimum la fonction de coût définie par le concepteur.

**Dans l'étape de reproduction**, Selon la valeur de sa propre fonction de coût, une plante peut produire un certain nombre de graines à partir sa valeur comprise entre 0 ( $s_{min}$ ) ( $min\_seeds$ ) à un nombre maximal ( $s_{max}$ ) ( $max\_seeds$ ). La meilleure plante dans la colonie produira un nombre maximum de graines, alors que la plus mauvaise plante ne peut produire aucune graine.

Entre ces deux valeurs extrêmes, le nombre de graines produites par une plante quelconque est donné par :

$$S_k = \frac{s_{max} - s_{min}}{f_{min} - f_{max}} (f_k - f_{max}) + s_{min} \quad (I.1)$$

Où  $s_k$  est la valeur de reproduction (le nombre de nouveaux individus) pour le même individu, qui a la valeur de coût  $f$ ,  $f_{min}$  et  $f_{max}$  sont respectivement le meilleur et le pire coût de la population.

**Dans l'étape Distribution spatiale**, les graines produites seront réparties autour de leur plante parente, sur des distances aléatoires, en utilisant la distribution uniforme avec une moyenne nulle et un écart-type SD qui diminue itérativement comme le montre l'équation :

$$\delta_{itr} = \left( \frac{itr_{max} - itr}{itr_{max} - 1} \right)^n (\delta_{ini} - \delta_{inal}) + \delta_{inal} \quad (I.2)$$

Avec  $It$  est le nombre d'itérations maximales.

Est l'indice de modulation non linéaire  $\sigma$  initial et  $\sigma$  final sont respectivement l'écart-type initial et final.

**Dans l'étape de la Limitation**, les graines vont se développer et devenir de nouvelles plantes qui seront ajoutées à la colonie, jusqu'à ce qu'un nombre maximum de plantes soit atteint.

**Dans l'étape d'exclusion compétitive**, si le nombre de plantes atteint sa taille maximale, une compétition entre ces individus sera lancée pour garder, seulement, les meilleures plantes dans la colonie.

Après l'atteinte du nombre maximum de plantes dans la colonie, chaque plante à la possibilité de produire un nombre de graines, elles sont évaluées et classées avec leurs parents après avoir trouvé leurs positions dans l'espace de recherche, ensuite on élimine les plantes ayant une fonction de coût inférieur au minimum permis et les meilleures plantes (parents et graines) peuvent survivre, ce mécanisme donne aussi une chance aux plantes, avec une fonction de coût faible, de reproduire dans le cas où leurs graines ont une bonne fonction de coût, dans la colonie [6].

#### **I.3.1.4 Les applications d'Iwo**

Bien que cette méthode soit récente, les domaines d'application des algorithmes (IWO) sont très nombreux. On peut en citer les suivants :

- L'optimisation de fonction.
  
- L'apprentissage des réseaux de neurones tels que MLP (Multi-layer perceptron) [7] .
  
- La conception des réseaux d'antennes et des autres dispositifs micro-ondes [8].
  
- Le marché de l'électricité [9].
  
- Le positionnement optimal des déclencheurs piézoélectriques [10].

possibilité de produire un nombre de graines, elles sont évaluées et classées avec leurs

parents après avoir trouvé leurs positions dans l'espace de recherche, ensuite on élimine les plantes ayant une fonction de coût inférieur au minimum permis et les meilleures plantes (parents et graines) peuvent survivre, ce mécanisme donne aussi une chance aux plantes, avec une fonction de coût faible, de reproduire dans le cas où leurs graines ont une bonne fonction de coût, dans la colonie [6].

#### **I.3.1.5 Avantages de l'algorithme IWO**

Une propriété importante de l'algorithme IWO est qu'elle permet à tous les agents ou plantes (grains) à participer dans le processus de reproduction

- L'algorithme IWO donne une chance aux plantes moins aptes à reproduire et si les grains sont bons, ils peuvent survivre dans la colonie.

- facile à implémenter.

- c'est un algorithme très intelligent.

- efficace pour résoudre des problèmes complexes.

- Le développement des systèmes de commande [11].

Après l'atteinte du nombre maximum de plantes dans la colonie, chaque plante à la

#### **I.3.1.6 Les inconvénients de l'algorithme IWO**

- Sensibilité aux conditions initiales.

- risque de stagnation.

- la complexité informatique [12].

### **I.3.2 Artificial Plant Growth Algorithm APGA**

Dans cette section, nous présentons une nouvelle variante simple d'un algorithme inspiré de la croissance des plantes.

#### **I.3.2.1 Définition**

L'APGA est un algorithme d'optimisation basé sur la croissance des plantes qui utilise des concepts tels que la photosynthèse, la germination et la croissance pour trouver des solutions optimales à des problèmes complexes. Plusieurs auteurs ont proposé des algorithmes inspirés des plantes [29].

#### **I.3.2.2 Les étapes principales de formulation de l'algorithme APGA**

##### **Initialisation :**

- Créer une population initiale de plantes avec des positions aléatoires.
- Évaluer la qualité de chaque plante en utilisant une fonction d'évaluation.
- Définir les paramètres de contrôle tels que la taille de la population et le taux de mutation.
- Répéter jusqu'à atteindre le critère d'arrêt.

**Sélection** : Sélectionner les plantes parentes pour la reproduction en fonction de leur qualité.

**Reproduction** : Créer de nouvelles plantes par les graines des plantes parentes sélectionnées.

**Mutation** : Appliquer des opérations de mutation aux nouvelles plantes pour introduire une exploration du voisinage, la solution est perturbée par une distribution normale.

**Évaluation** : Évaluer la qualité des nouvelles plantes.

**Compétition** : Sélectionner les meilleures plantes parmi la population initiale et les nouvelles plantes générées.

**Remplacement** : Remplacer les plantes élaguées par de nouvelles plantes pour maintenir la taille de la population.

**Mise à jour** : Mettre à jour les paramètres de contrôle si nécessaire (par exemple, taux de mutation adaptatif).

**Fin de la boucle principale** : Rapporter la meilleure solution trouvée.

## **I.4 Analyse de performance de l'algorithme APGA**

### **I.4.1 Analyse de performance**

Dans le domaine de l'optimisation, les fonctions de test sont des fonctions d'évaluation des caractéristiques des algorithmes d'optimisation, telles que taux de convergence, précision, robustesse et performances générales.

Et pour tester l'algorithme on présenter trois fonctions de test.

#### **I.4.1.1 La fonction sphère**

C'est une fonction adaptée à l'optimisation à objectif unique, ce qui signifie qu'elle présente une fonction à objectif unique [14].

Elle est définie par : 
$$f(x) = \sum_{i=1}^p x_i^2 \quad (\text{I.3})$$

L'espace de recherche est : [-100,100]

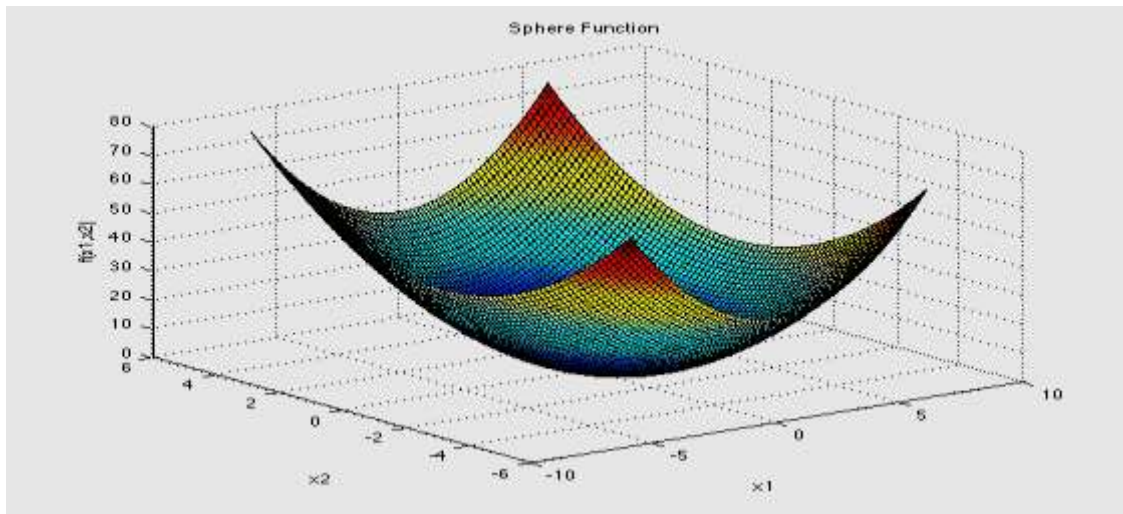


Figure I:4: La fonction sphère en 2 D [14]

#### I.4.1.2 La fonction de Scheffel

La fonction Scheffel modifiée, également connue sous le nom de fonction Schwefel 1.2, est une fonction d'optimisation multimodale et non linéaire souvent utilisée pour évaluer les performances des algorithmes d'optimisation. Elle est définie comme suit [14] :

$$f(x) = \sum_{i=1}^D x_{i=1}^2 + \prod_{i=1}^D |x_i| \quad (I.4)$$

L'espace de recherche :

$$x \in [-10, 10]$$

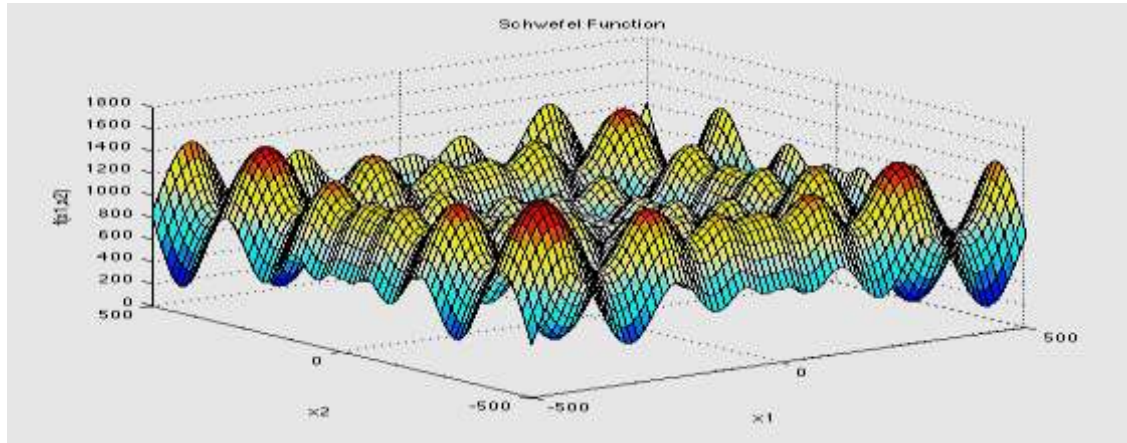


Figure I:5: la fonction scheffel en 2 D

#### I.4.1.3 La fonction de Rosenbrock

La fonction Rosenbrock, également appelée fonction Valley ou Banana, est un problème de test populaire pour les algorithmes d'optimisation basés sur le gradient [14].

##### Domaine d'entrée :

La fonction est généralement évaluée sur l'hypercube  $x_i \in [-5, 10]$ , pour tout  $i = 1, \dots, d$ , bien qu'elle puisse être restreinte à l'hypercube  $x_i \in [-2.048, 2.048]$ , pour tout  $i = 1$

$$\text{Son expression : } f(x) = \sum_{i=1}^{D-1} (x_{i+1} - x_i)^2 - (x_i - 1)^2 \quad (\text{I.5})$$



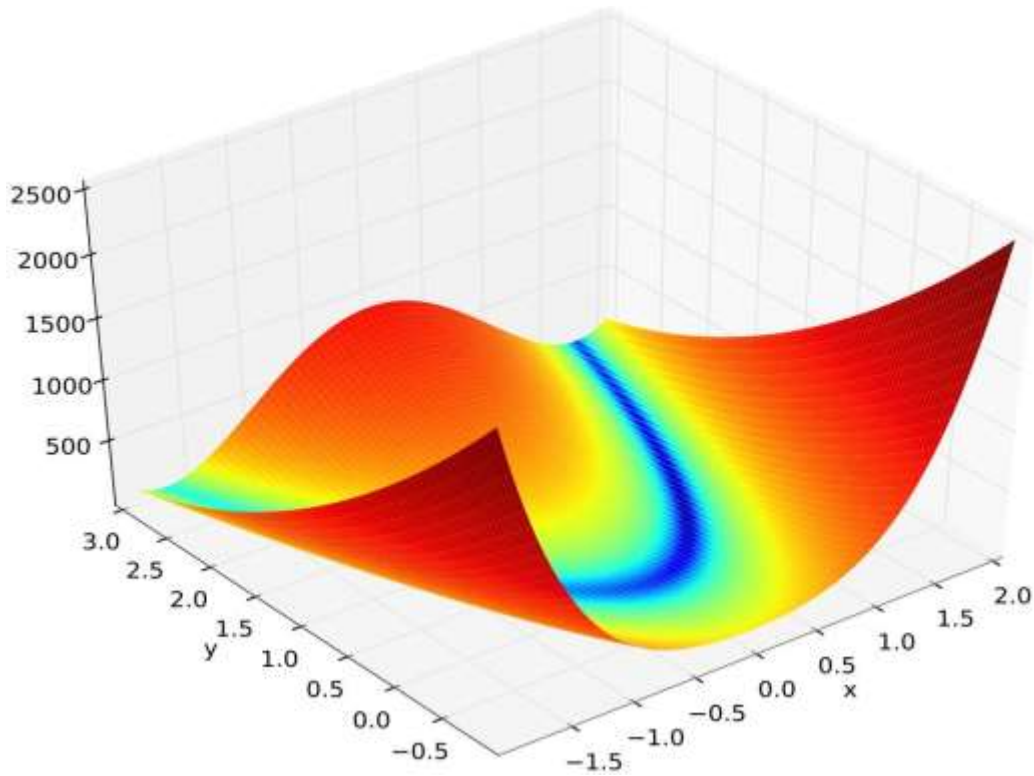


Figure I:6: la fonction Rosenbrock en 2 D

#### I.4.1.4 La fonction de Rastrigin

La fonction de Rastrigin est une fonction mathématique souvent utilisée pour évaluer la performance d'algorithmes d'optimisation. Elle présente des pièges intéressants, sous la forme de ses nombreux minima et maxima locaux. Elle a été proposée, en 1974, par Rastrigin en deux dimensions et a été généralisée par Mühlenbein et al [14].

$$\text{Son expression : } f(x) = a + \sum_{i=1}^D x_i^2 - \text{acos}(2 * \prod_{i=1}^D x_i) \quad (\text{I.6})$$

Son domaine : Ou  $a=10$  et  $x_i \in [-5, 12 ; 5, 12]$  son minimum globale se trouver à l'origine, ou sa valeur est nulle

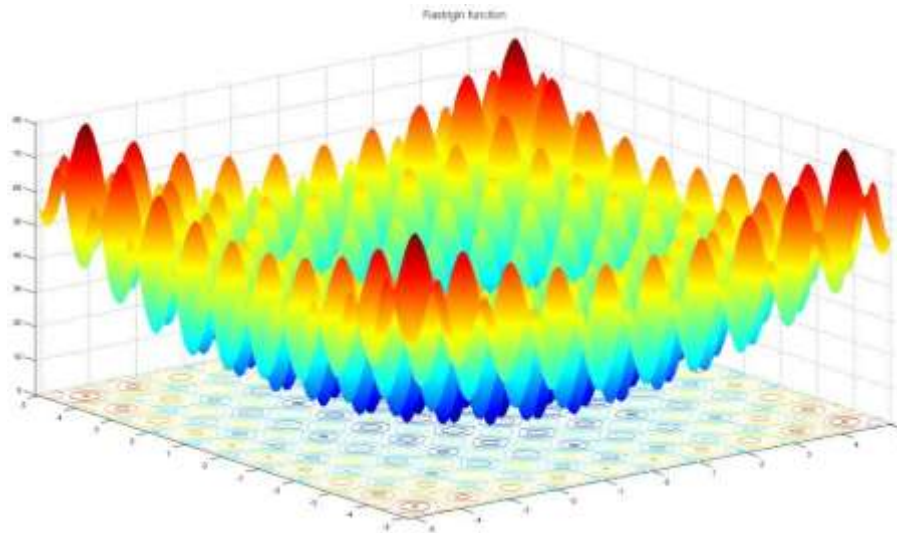


Figure I:7: la fonction Rastrigin en 2 D [14]

#### I.4.1.5 Résultats et discussion :

Le résultat obtenu de la Simulation de l'algorithme APGA avec les 4 fonctions test présentées dans les figures suivantes.

Les paramètres	La description	La valeur
N	Le nombre de solutions dans la population	80
D	La dimension de l'espace de recherché	10
Max	Espace de recherche	100
Min	Espace de recherche	-100

Tableau I-1 les paramètres utilisées dans l'algorithme APGA

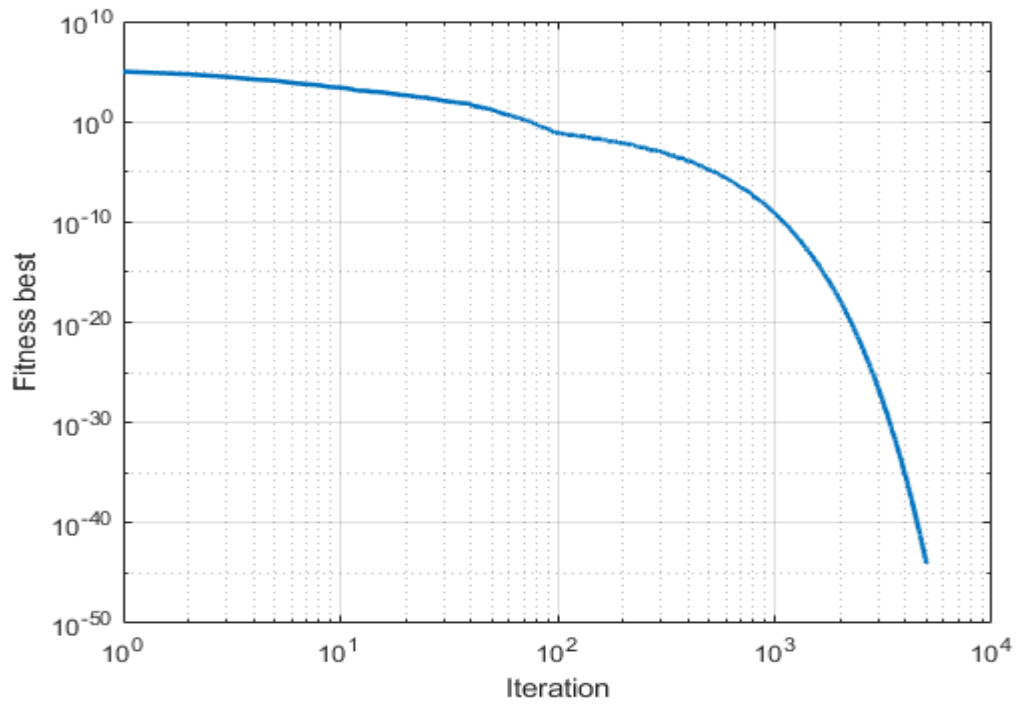


Figure I:8: application d'APGA sur sphère

**D=10**

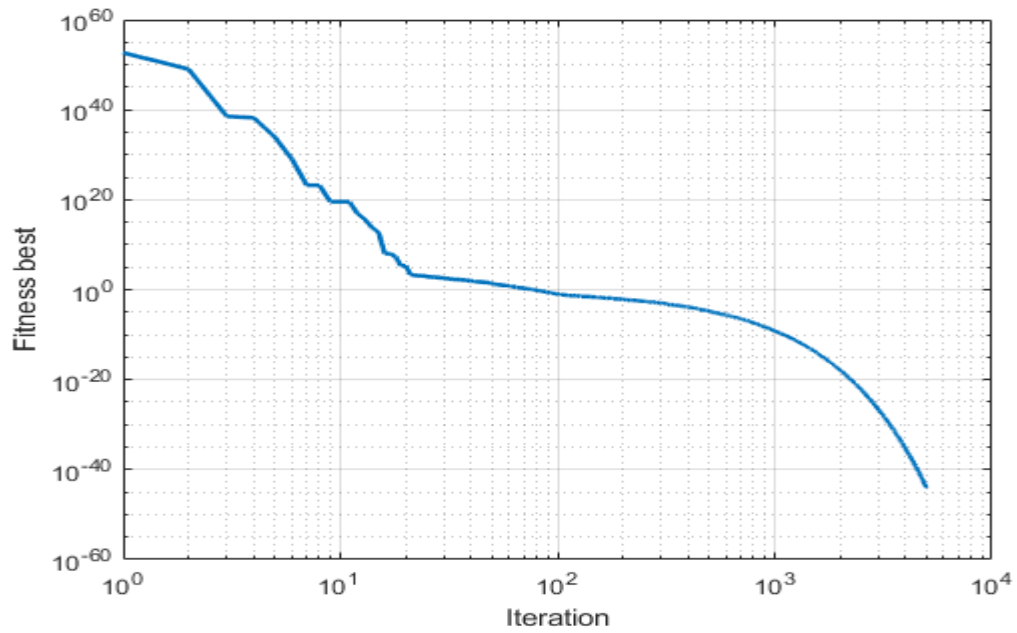


Figure I:9: application d'APGA sur Schwefel

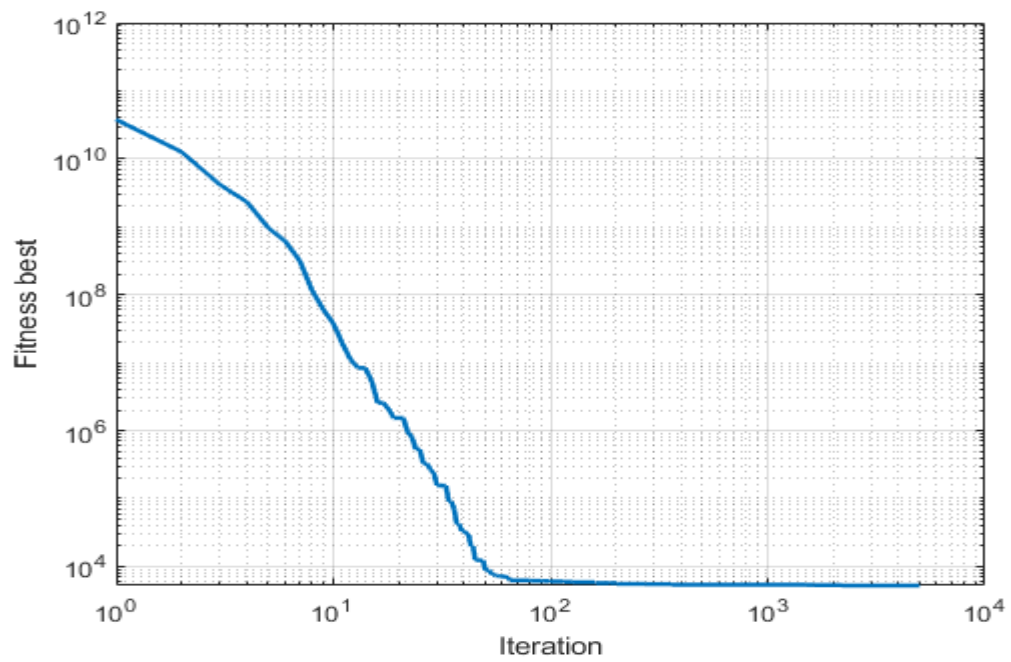


Figure I:10: application d'APGA sur Rosenbrock

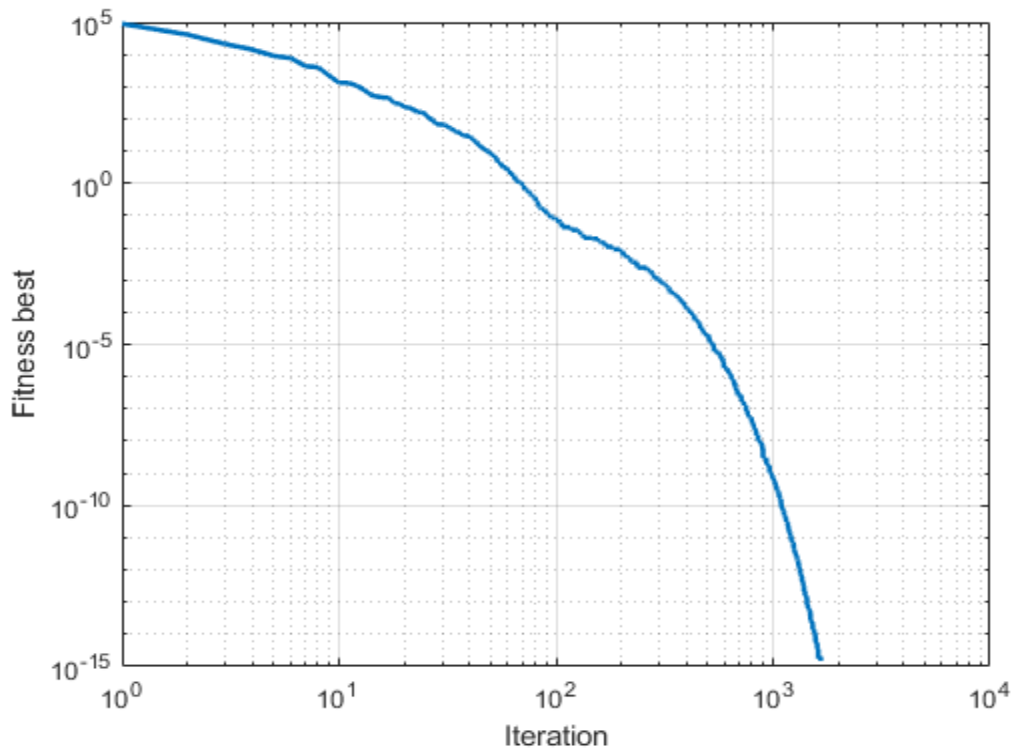


Figure I:11: application d'APGA sur Rastrigin

**I.4.1.6 Interprétations des résultats**

Les équations des fonctions de benchmark sont données dans les équations (I.3, I.4, I.5 et I.6).

-Pour 5000 itérations on n'a obtenu ses figures :

La figure (I.8) présente l'application d'APGA sur la fonction sphère, on remarque que la convergence est bonne.

La figure (I.9) présente l'application d'APGA sur Schwefel.

La figure (I.10) présente l'application d'APGA sur Rosenbrock.

La figure (I.11) présente l'application d'APGA sur Rastrigin.

L'algorithme de croissance artificielle des plantes (APGA) présente différentes caractéristiques en termes de convergence lorsqu'il est appliqué à des fonctions unimodales et multimodales.

Pour les fonctions unimodales, qui présentent un seul minimum global, l'APGA a tendance à converger rapidement vers la solution optimale. L'algorithme peut exploiter efficacement les informations locales et rechercher le voisinage de la meilleure solution, en utilisant les opérations de mutation pour explorer de nouvelles régions de l'espace de recherche. Cela permet à l'APGA de converger rapidement vers le minimum global, en utilisant la compétition entre les plantes pour sélectionner les solutions de haute qualité.

En revanche, pour les fonctions multimodales, qui comportent plusieurs minimums locaux, l'APGA peut rencontrer des défis en termes de convergence. Étant donné que l'algorithme est basé sur des mécanismes de croissance et de compétition entre les plantes, il peut être piégé dans un optimum local si la diversité de la population est insuffisante ou si les opérations de mutation ne sont pas suffisamment exploratoires. L'APGA peut également être sensible aux problèmes de convergence prématurée, où les plantes peuvent converger vers un minimum local avant d'avoir exploré suffisamment l'espace de recherche.

Pour améliorer la convergence de l'APGA sur des fonctions multimodales, différentes techniques peuvent être utilisées, telles que l'adaptation des paramètres de contrôle,

---

l'introduction de mécanismes d'échange d'information entre les plantes, l'utilisation de

stratégies de mutation adaptatives, ou l'ajustement des critères de compétition et d'élagage pour maintenir une diversité suffisante dans la population. Ceci peut faire l'objet d'étude future.

Il est important de noter que la convergence de l'APGA peut varier en fonction de la complexité et des propriétés spécifiques de la fonction d'optimisation, ainsi que des paramètres choisis pour l'algorithme. Des études empiriques et des analyses théoriques

spécifiques aux problèmes d'optimisation particuliers peuvent fournir des informations plus détaillées sur la convergence de l'APGA dans des contextes spécifiques.

## **I.5 Conclusion**

Ce chapitre est consacré à l'intelligence des plantes. Deux algorithmes ont été présentés, IWO et APGA. Ces deux algorithmes sont inspirés du comportement des plantes et de leur capacité de reproduction et d'adaptation. Une analyse de performance de l'algorithme APGA a été présentée sur des fonctions de tests. Dans le chapitre 3, une application de l'algorithme APGA en robotique mobile est présentée. Mais avant d'y parvenir, le chapitre 2 présente la modélisation dynamique du robot utilisé dans cette étude.

# **Chapitre II**

## **Modélisation cinématique et dynamique d'un robot mobile**

## chapitreII Modélisation cinématique et dynamique du robot

### II.1 Introduction

Les robots mobiles présentent un cas particulier en robotique. Leur intérêt réside dans Leur mobilité, destinés à remplir des tâches pénibles (exemple : transport de charges lourdes) Et ils travaillent même en ambiance hostile (nucléaire, marine, spatiale, lutte contre l'incendie, Surveillance...). L'aspect particulier de la mobilité impose une complexité technologique (capteurs, motricité, énergie).

Le contrôle du robot mobile et la commande nécessite la modélisation qui sera présentée dans ce chapitre.

### II.2 Les robots mobiles :

#### II.2.1 Définition

Un robot mobile est un véhicule doté de moyens de locomotion qui lui permettent de se déplacer. Suivant son degré d'autonomie il peut être doté de moyens de perception et de raisonnement [15]

#### II.2.2 Classification des robots mobiles

La classification des robots mobiles se fait suivant plusieurs critères (degré d'autonomie Système de locomotion, énergie utilisée ...).

La classification la plus intéressante, et la plus utilisée est selon leur degré d'autonomie.

- 1-Véhicule télécommandé par un opérateur qui lui impose chaque tâche élémentaire à réaliser.
- 2- Véhicule télécommandé au sens de la tâche à réaliser. Le véhicule contrôle automatiquement ses actions.
- 3- Véhicule semi- autonome réalisant sans l'aide de l'opérateur des tâches prédéfinies.



4- Véhicule autonome qui réalise des tâches semi- définies [15].

### II.2.3 L'architecture logicielle des robots mobiles

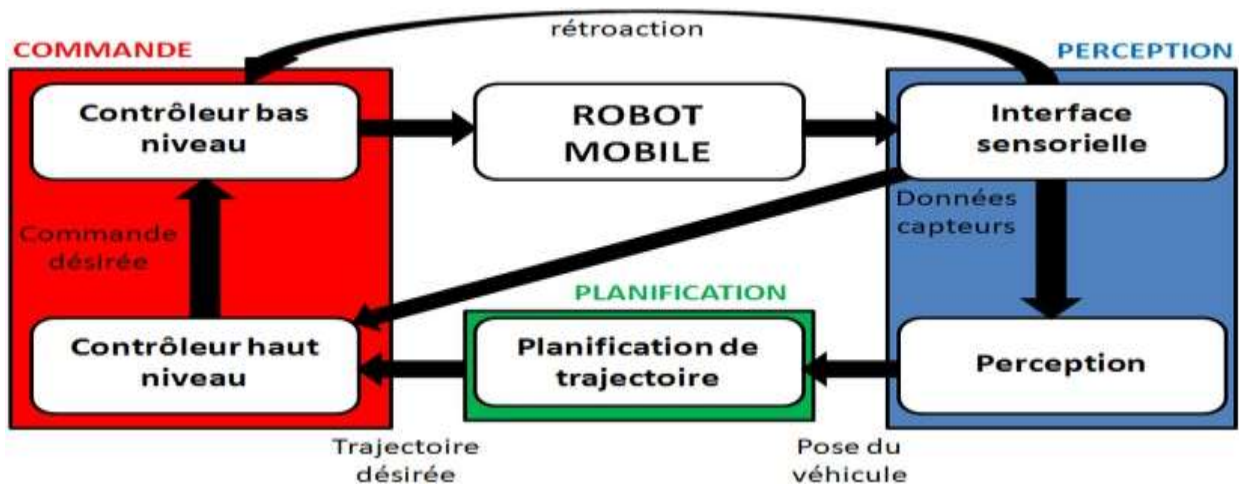


Figure II:1: architecture logiciel d'un robot mobile [17]

### II.2.4 Les organes de sécurité

Un robot, selon la tâche qui lui est confiée, peut être amené à travailler au voisinage du personnel. A ce titre, il est obligatoire qu'il soit doté d'organes garantissant la sécurité. Des capteurs sont disponibles tout autour du mobile afin de détecter un obstacle sur un domaine le plus étendu possible.

Deux types de capteurs sont employés : les capteurs périmétriques assurant la détection avant collision (ultra-son, hyper fréquence, infrarouge...) et les capteurs de Contact détectant une collision ou un choc avec l'environnement (contact électrique sur pare-chocs, résistance variable, fibre optique...) [15].

### II.2.5 Traitement des informations et gestion des tâches

L'ensemble de traitement des informations et gestion des tâches constitue le module information central qui établit les commandes permettant au mobile de réaliser un déplacement et d'activer les divers organes en accord avec l'objectif.

### II.2.6 La navigation

La navigation est une étape très importante en robotique mobile. Elle permet une large autonomie à un robot mobile. Le système de navigation comporte plusieurs modules qui peuvent être traités différemment et parmi lesquels on distingue celui de la localisation et celui de l'évitement d'obstacles. La détection et l'évitement des obstacles sont des étapes fondamentales de l'évolution d'un robot en territoire inconnu. On dispose à cet effet de plusieurs types de capteurs : caméras ; un programme d'analyse des images étant alors nécessaire, capteurs laser, capteurs infrarouge et capteurs à ultrasons [18].

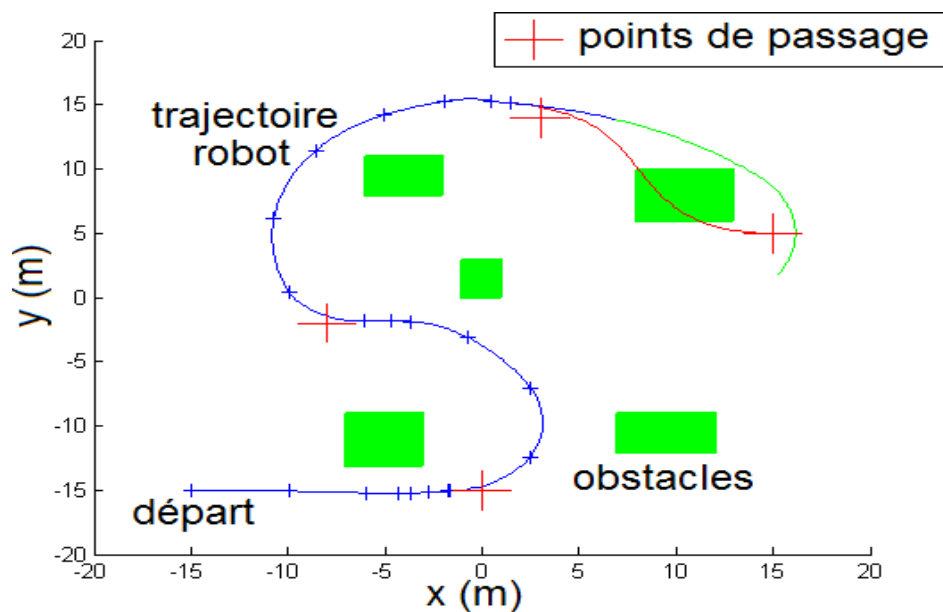


Figure II.2: trajectoire d'un robot [18]

### II.2.7 La modélisation dans l'environnement

La connaissance du milieu dans lequel évolue le robot mobile n'est établie en général qu'après avoir effectué une campagne de mesures appropriées à l'ensemble des éléments constituant l'environnement. Cette procédure est difficile à réaliser, mais elle peut être évitée si le robot construit lui-même son modèle d'environnement de manière dynamique. Par contre, la planification de trajectoire n'est pas utilisable tant que le robot ne dispose pas d'un modèle de l'espace d'évolution ce qui handicape très fortement l'utilisation du robot. A partir de cette base d'informations et d'une loi évaluant les erreurs de représentation, le planificateur peut générer des sous-trajectoires faisables dans certaines parties et par la suite modifier les sous-trajectoires dans d'autres parties à l'aide des informations locales issues des mesures effectuées par les capteurs

d'environnement. Lors de l'exécution d'une trajectoire, le robot acquiert des informations qui vont permettre de reconstituer le plus fidèlement possible le type de l'environnement de manière récursive à l'aide d'un algorithme approprié [19].

### II.3 Les robots mobiles à roues

La mobilité par roues est la structure mécanique la plus communément appliquée. Cette technique assure selon l'agencement et les dimensions des roues un déplacement dans toutes les directions avec une accélération et une vitesse importantes. Le franchissement d'obstacles ou l'escalade de marches d'escaliers est possible [15].

#### II.3.1 Types des robots mobiles à roues

Il existe plusieurs classes de robots à roue, déterminée principalement, par la position et le nombre de roues utilisées. Nous citerons ici les 3 classes principales de robots à roues [20].

##### II.3.1.1 Robot unicycle

Un robot de type uni cycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles (libres) pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices. En effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion[20].

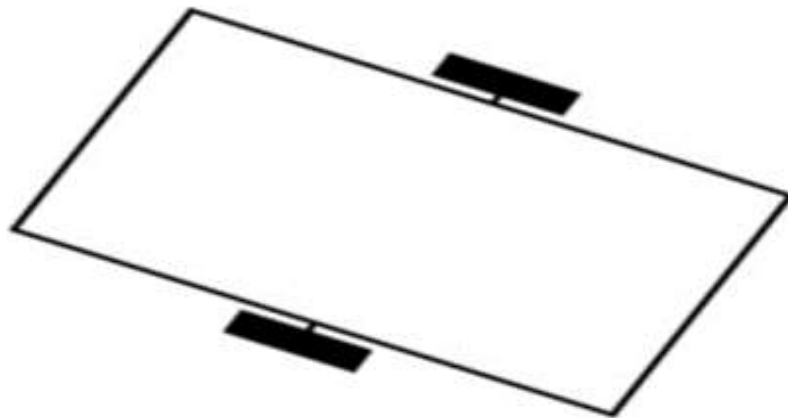


Figure II:3: robot mobile unicycle [20]

### II.3.1.2 Robot tricycle

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal.

Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable. Il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion [20]

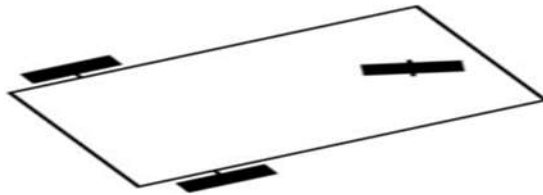


Figure II:4 : robot tricycle [20]

### II.3.1.3 Robot voiture

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées aussi sur un même axe.

Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire. Toutes les autres propriétés du robot voiture sont identiques au robot tricycle, le deuxième pouvant être ramenée au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé.

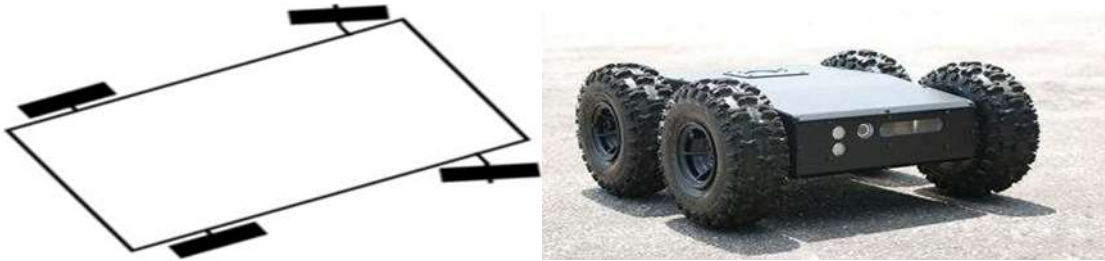


Figure II:5: robot voiture [20]

## II.4 Modélisation d'un robot mobile de type unicycle

Un robot de type uni cycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices [24].

Caractéristiques : Stables, Complexité mécanique faible, non-holonyme.

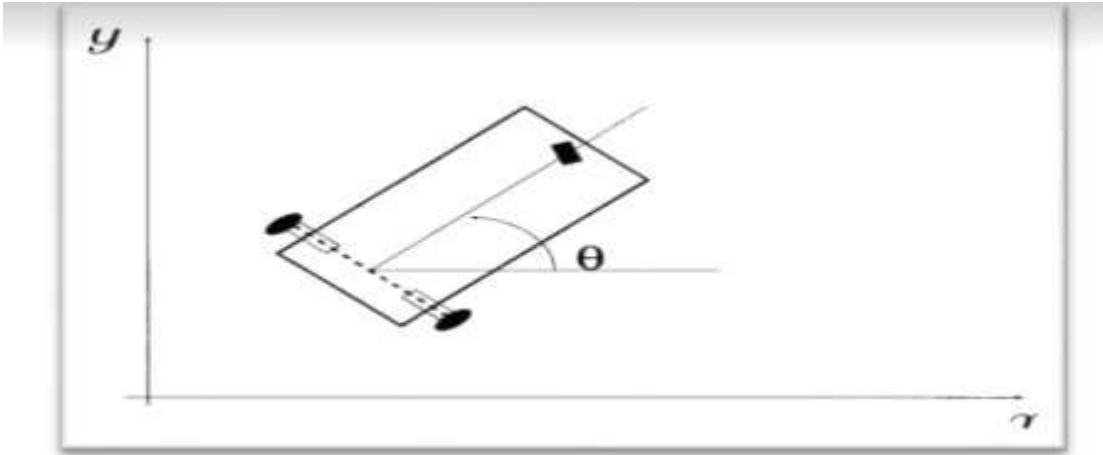


Figure II:6: robot unicycle [19]

### Systèmes de coordonnées

Afin de décrire la position du WMR dans son environnement, deux systèmes de coordonnées différents (frames) doivent être définis.

-Système de coordonnées inertielles : Ce système de coordonnées est un cadre qui est fixé dans l'environnement ou le plan dans lequel le WMR se déplace. De plus, ce référentiel est considéré comme le référentiel et est noté  $X_I, Y_I$ .

-Système de coordonnées du robot : ce système de coordonnées est un cadre attaché au WMR, et donc, se déplaçant avec lui. Ce cadre est noté  $X_r, Y_r$ .

Les deux cadres définis sont représentés sur la figure 19. L'origine du cadre du robot est défini comme étant le point médian A sur l'axe entre les roues. Le centre de masse C du robot est supposé être sur l'axe de symétrie, à une distance d de l'origine A.

Comme le montre la figure (19), la position et l'orientation du robot dans le référentiel

$$\text{inertiel peut être définie comme : } q^I = \begin{bmatrix} x_a \\ y_a \\ \theta \end{bmatrix} \quad (\text{II.1})$$

La question importante qui doit être expliquée à ce stade est la correspondance entre ces

deux trames. La position de tout point sur le robot peut être définie dans le cadre du robot et le cadre inertiel comme suit.

$x^r$                        $x^I$   
 $X^r = [y^r]$ , et  $X^I = [y^I]$  et être les coordonnées du point donné et dans le cadre du robot et le  
 $\theta^r$                        $\theta^I$   
cadre inertiel, respectivement.

Ensuite, les deux coordonnées sont liées par la relation suivante transformation :

$$X^l = (\theta)X^r \quad (\text{II.2})$$

Où  $R(\theta)$  est la matrice de rotation orthogonale

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{II.3})$$

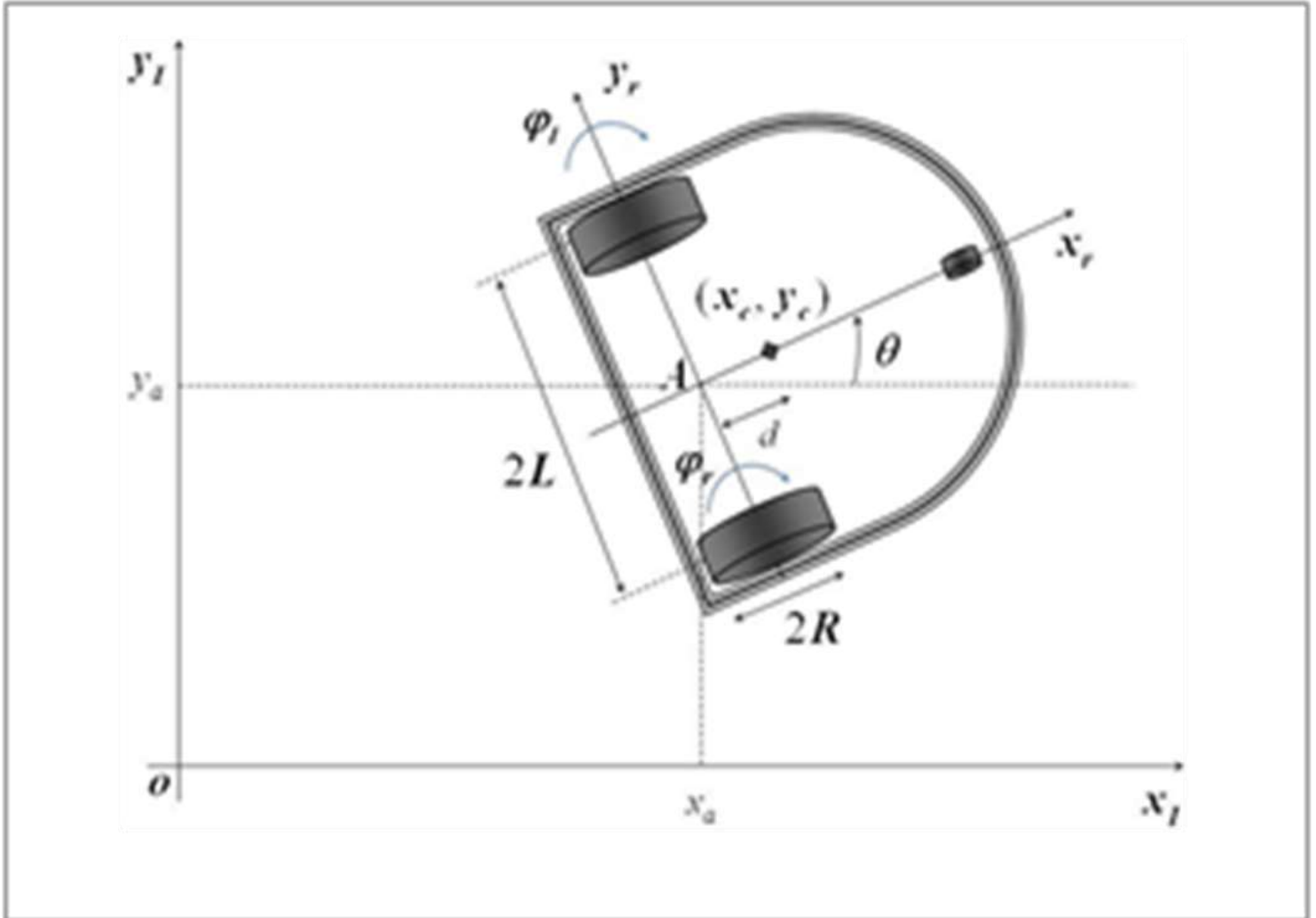


Figure II:7: Robot mobile de type unicycle dans les repères fixe. [24]

-A : le point milieu de l'axe des roues.

-2R : représente le diamètre de la roue du robot.

-2L : représente la largeur du robot.

-  $\dot{\theta}$  et  $\dot{\phi}$  : représentent respectivement la vitesse de rotation de la roue droite et de la roue



gauche.

- $\theta$  : est l'angle d'orientation du robot.

- $d$  : est la distance entre le point A et le point C.

Cette transformation permettra également la gestion du mouvement entre les cadres.

$$\dot{X}^I = (\theta)\dot{X}^r \quad (\text{II.4})$$

On verra dans la section suivante que l'équation (II.4) est très importante dans la dérivation des modèles cinématiques et dynamiques DDMR tels qu'ils décrivent la relation entre les vitesses dans le référentiel inertiel et le châssis robotique.

### I.1.1 Contraintes cinématiques du robot à entraînement différentiel

Le mouvement d'un robot mobile à entraînement différentiel est caractérisé par deux équations de contraintes non holonomes, qui sont obtenues par deux hypothèses principales :

Pas de mouvement de glissement latéral : cette contrainte signifie simplement que le robot ne peut se déplacer que dans un mouvement courbe (avant et arrière) mais pas latéralement. Dans le référentiel du robot, cette condition signifie que la vitesse du point central A est nulle le long de l'axe latéral :

$$\dot{y}_a^r = 0 \quad (\text{II.5})$$

Utilisation de la matrice de rotation orthogonale  $R(\theta)$ , la vitesse dans le référentiel inertiel donne :

$$-\dot{x}_a \sin \theta + \dot{y}_a \cos \theta = 0 \quad (\text{II.6})$$

-Contrainte de roulement pur :

La contrainte de roulement pur représente le fait que chaque roue conserve un point de contact P avec le sol comme le montre la figure II.9. Il n'y a pas de patinage de la roue dans son axe longitudinal ( $x_r$ ) et pas de dérapage dans son axe orthogonal ( $y_r$ ). Les vitesses des points de contact dans le châssis du robot sont liés aux vitesses des roues par :

$$\begin{cases} v_{pR} = R\dot{\phi}_R \\ v_{pL} = R\dot{\phi}_L \end{cases} \quad (\text{II.7})$$

Dans le référentiel inertiel, ces vitesses peuvent être calculées en fonction des vitesses du centre du robot A :

$$\begin{cases} \dot{x}_{pR} = \dot{x}_a + L\dot{\theta} \cos \theta \\ \dot{y}_{pR} = \dot{y}_a + L\dot{\theta} \sin \theta \end{cases} \quad (\text{II.8})$$

$$\begin{cases} \dot{x}_{pL} = \dot{x}_a - L\dot{\theta} \cos \theta \\ \dot{y}_{pL} = \dot{y}_a - L\dot{\theta} \sin \theta \end{cases} \quad (\text{II.9})$$

En utilisant la matrice de rotation  $R(\theta)$ , les équations de contrainte de roulement sont formulées comme suit :

$$\begin{aligned} \dot{x}_{pR} \cos \theta + \dot{y}_{pR} \sin \theta &= R\dot{\phi}_R \\ \dot{x}_{pL} \cos \theta + \dot{y}_{pL} \sin \theta &= R\dot{\phi}_L \end{aligned} \quad (\text{II.10})$$

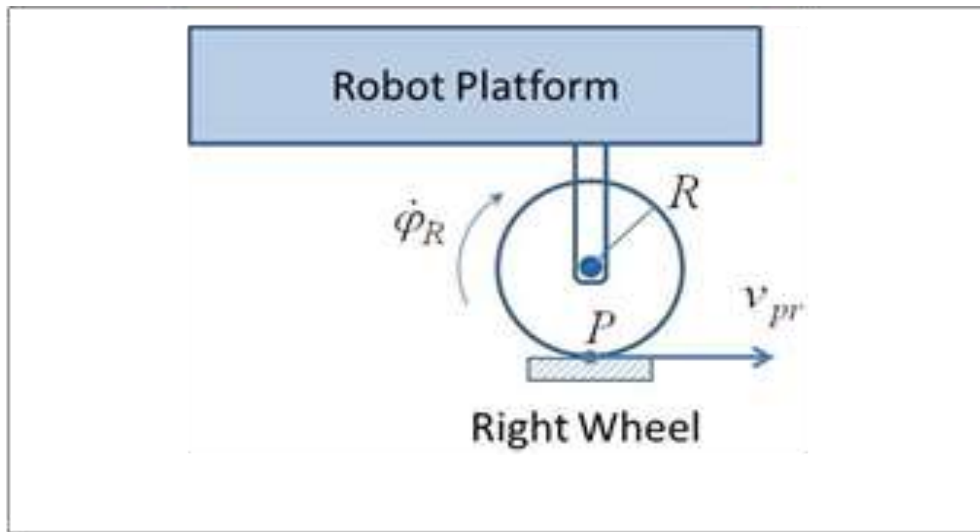


Figure II.8 :Robot Plateforme [24]

En utilisant les vitesses des points de contact de l'équation (x, y) et en substituant dans (x, y), les trois équations de contraintes peuvent s'écrire sous la forme matricielle suivante :

$$\Lambda(q)\dot{q} = 0 \quad (\text{II.11})$$

Où

$$\Lambda(q) = \begin{bmatrix} -\sin \theta & \cos \theta & 0 & 0 & 0 \\ \cos \theta & \sin \theta & L & -R & 0 \\ \cos \theta & \sin \theta & -L & 0 & -R \end{bmatrix} \quad (\text{II.12})$$

Et

$$\dot{q} = [\dot{x}_a \dot{y}_a \dot{\theta} \dot{\phi}_R \dot{\phi}_L]^T \quad (\text{II .13})$$

$$\begin{cases} v_R = R\dot{\phi}_R \\ v_L = R\dot{\phi}_L \end{cases} \quad (\text{II .14})$$

La matrice de contraintes ci-dessus  $\Lambda(q)$  sera utilisée dans la section suivante pour modélisation dynamique du robot.

### I.1.2 Modélisation cinématique

La modélisation cinématique est l'étude du mouvement des systèmes mécanique sans tenir compte des forces qui affectent le mouvement. Pour le robot, l'objectif principal de la modélisation cinématique est de représenter les vitesses du robot en fonction des vitesses des roues motrices ainsi que les paramètres géométrique du robot.

La vitesse linéaire de chaque roue motrice dans le cadre du robot est donc la vitesse linéaire du robot dans le cadre du robot est la moyenne des vitesses linéaire des deux roues

$$v = \frac{v_R + v_L}{2} = R \frac{(\dot{\phi}_R + \dot{\phi}_L)}{2} \quad (\text{II .15})$$

Et sa vitesse angulaire est

$$w = \frac{v_R - v_L}{2L} = R \frac{(\dot{\phi}_R - \dot{\phi}_L)}{2L} \quad (\text{II .16})$$

Les vitesses des DDMR dans le cadre du robot peuvent maintenant être représentées en termes de vitesses du point central A dans le cadre du robot comme suit :

$$\begin{cases} \dot{x}_a^r = R \frac{(\dot{\phi}_R + \dot{\phi}_L)}{2} \\ \dot{y}_a^r = 0 \\ \dot{\theta} = w = R \frac{(\dot{\phi}_R - \dot{\phi}_L)}{2L} \end{cases} \quad (\text{II .17})$$

Ainsi

$$\begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{2L} & \frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \quad (\text{II .18})$$

Les vitesses DDMR peuvent également être obtenues dans le référentiel inertiel comme suit :

$$\dot{q}^I = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} \cos \theta & \frac{R}{2} \cos \theta \\ \frac{R}{2} \sin \theta & \frac{R}{2} \sin \theta \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \quad (\text{II .19})$$

L'équation (II.19) représente le modèle cinématique avant du DDMR. Une autre forme alternative pour le modèle cinématique peut être obtenue en représentant les vitesses DDMR en termes de vitesses linéaires et angulaires de DDMR dans le référentiel robot.

$$\dot{q}^I = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (\text{II .20})$$

### I.1.3 Modélisation dynamique

La dynamique est l'étude du mouvement d'un système mécanique en considérant les différentes forces qui affectent son mouvement contrairement à la cinématique où les forces ne sont pas prises en compte. Le modèle dynamique du DDMR est essentiel pour l'analyse de simulation du mouvement DDMR et pour la conception de divers algorithmes de contrôle de mouvement.

Un DDMR non holonome à  $n$  coordonnées généralisées  $q_1, q_2, \dots, q_n$  et soumis à  $m$  contraintes peut être décrit par les équations de mouvement suivantes :

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - \Lambda^T(q)\lambda \quad (\text{II .21})$$

Où :

$M(q)$  une matrice d'inertie définie positive symétrique  $n \times n$ ,  $V(q, \dot{q})$  est la matrice centripète et de Coriolis,  $F(\dot{q})$  est la matrice de frottement de surface,  $G(q)$  est le vecteur gravitationnel,  $\tau_d$  est le vecteur de perturbations inconnues bornées incluant des dynamiques non modélisées non structurées,  $B(q)$  est la matrice d'entrées,  $\tau$  est le vecteur d'entrées,  $\Lambda^T$  est la matrice associée aux contraintes cinématiques et  $\lambda$  est le vecteur des multiplicateurs de Lagrange [21,22].

#### Approche dynamique de Lagrange

L'approche dynamique de Lagrange est une méthode très puissante pour formuler les équations de mouvement des systèmes mécaniques. Cette méthode, qui a été introduite par Lagrange, est utilisée pour dériver systématiquement les équations du mouvement en considérant les énergies cinétique et potentielle du système donné.

L'équation de Lagrange peut s'écrire sous la forme suivante :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = F - \Lambda^T(q)\lambda \quad (\text{II .22})$$

Où  $L = T - V$  est la fonction Lagrange,  $T$ , est l'énergie cinétique de système,  $V$  est l'énergie potentielle du système,  $q_i$  sont les coordonnées généralisées,  $F$  est le vecteur de force généralisé,  $\Lambda$  est la matrice des

contraintes et  $\lambda$  est le vecteur des multiplicateurs de Lagrange associés aux contraintes.

La première étape de la dérivation du modèle dynamique à l'aide de Lagrange approche consiste à trouver les énergies cinétiques et potentielles qui régissent motion de la DDMR. De plus, comme la DDMR évolue dans le  $X_I, Y_I$  plan, l'énergie potentielle du DDMR est considérée comme nulle.

Pour le DDMR, les coordonnées généralisées sont choisies comme

$$q = [x_a y_a \theta \ \Phi_R \Phi_L]^T \quad (\text{II .23})$$

Les énergies cinétiques du DDMR sont la somme de l'énergie cinétique de la plate-forme du robot sans roues plus les énergies cinétiques des roues et actionneurs.

L'énergie cinétique de la plate-forme du robot est :

$$T_c = \frac{1}{2} m_c v_c^2 + \frac{1}{2} I_c \dot{\theta}^2 \quad (\text{II .24})$$

Alors que l'énergie cinétique des roues droites et gauche est

$$T_{wR} = \frac{1}{2} m_w v_{wR}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_w \dot{\phi}_R^2 \quad (\text{II .25})$$

$$T_{wL} = \frac{1}{2} m_w v_{wL}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_w \dot{\phi}_L^2 \quad (\text{II .26})$$

Où,  $m_c$  est la masse du DDMR sans les roues motrices et les actionneurs (moteurs à courant continu),  $m_w$  est la masse de chaque roue motrice (avec actionneur),  $I_c$  est le moment d'inertie du DDMR par rapport à la verticale passant par le centre de masse,  $I_w$  est le moment d'inertie de chaque roue motrice avec un moteur autour de l'axe de la roue, et  $I_m$  est le moment d'inertie de chaque roue motrice avec un moteur autour du diamètre de la roue.

Toutes les vitesses seront d'abord exprimées en fonction des coordonnées généralisées à l'aide de l'équation générale des vitesses dans le référentiel inertiel.

$$v_i^2 = \dot{x}_i^2 + \dot{y}_i^2 \quad (\text{II .27})$$

Les composants  $X_I$  et  $Y_I$  du centre de masse et des roues peuvent être obtenus en termes des coordonnées généralisées comme suit :

$$\begin{cases} x_c = x_a + d \cos \theta \\ y_c = y_a + d \sin \theta \end{cases} \quad (\text{II .28})$$

$$\begin{cases} x_{wR} = x_a + L \sin \theta \\ x_{wL} = y_a - L \cos \theta \end{cases} \quad 45(\text{II .29})$$

$$\begin{cases} x_{wL} = x_a - L \sin \theta \\ x_{wL} = y_a + L \cos \theta \end{cases} \quad (\text{II .30})$$

En utilisant les équations (II.24)-(II.26) avec les équations (II.27)-(II.30), le total l'énergie cinétique du DDMR est :

$$T = \frac{1}{2}m(\dot{x}_a^2 + \dot{y}_a^2) - m_c d \dot{\theta} (\dot{x}_a \sin \theta - \dot{y}_a \cos \theta) + \frac{1}{2}I_w(\dot{\phi}_R^2 + \dot{\phi}_L^2) + \frac{1}{2}I\dot{\theta}^2 \quad (\text{II.31})$$

Où les nouveaux paramètres suivants sont introduits  $m = m_c + 2m_w$  est la masse totale du robot,  $I = I_c + m_c d^2 + 2m_w L^2 + 2I_m$  et est l'inertie équivalente totale.

En utilisant l'équation (II.22) avec la fonction lagrangienne, L=T les équations de mouvement du DDMR

sont données par :

$$m\ddot{x}_a - m_c d \ddot{\theta} \sin \theta - m_c d \dot{\theta}^2 \cos \theta = C_1 \quad (\text{II .32})$$

$$m\ddot{y}_a - m_c d \ddot{\theta} \cos \theta - m_c d \dot{\theta}^2 \sin \theta = C_2 \quad (\text{II .33})$$

$$m\ddot{\theta} - m_c d \ddot{x}_a \sin \theta - m_c d \ddot{y}_a \cos \theta = C_3 \quad (\text{II .34})$$

$$I_w \ddot{\phi}_R = \tau_R + C_4 \quad (\text{II .35})$$

$$I_w \ddot{\phi}_L = \tau_L + C_5 \quad (\text{II .36})$$

Où  $(C_1, C_2, C_3, C_4, C_5)$ , sont des coefficients liés à la cinématique contraintes, qui peuvent être écrites en termes de multiplicateurs de Lagrange vecteur  $\lambda$  et la matrice de contraintes cinématiques  $\Lambda$ .

$$\Lambda^T(q) = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix} \quad (\text{II .37})$$

Maintenant, les équations de mouvement obtenues (II.32)-(II.36) peuvent être représentées sous la forme

générale donnée par l'équation (II.21) comme

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} = B(q)\tau - \Lambda^T(q)\lambda \quad (\text{II.38})$$

Où

$$B(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{et} \quad \Lambda^T(q)\lambda = \begin{bmatrix} -\sin \theta & \cos \theta & \cos \theta \\ \cos \theta & \sin \theta & \sin \theta \\ 0 & L & -L \\ 0 & -R & 0 \\ 0 & 0 & -R \end{bmatrix} \times \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix}$$

$$M(q) = \begin{bmatrix} m & 0 & -m_c d \sin \theta & 0 & 0 \\ 0 & m & m_c d \cos \theta & 0 & 0 \\ -m_c d \sin \theta & m_c d \sin \theta & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix}$$

$$V(q, \dot{q}) = \begin{bmatrix} 0 & 0 & -m_c d \dot{\theta} \cos \theta & 0 & 0 \\ 0 & 0 & -m_c d \dot{\theta} \sin \theta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Ensuite, le système décrit par l'équation (II.38) est transformé en une forme alternative plus pratique aux fins de contrôle et simulation. L'objectif principal est d'éliminer la contrainte terme  $\Lambda^T(q)\lambda$  dans l'équation (II.38) puisque les multiplicateurs de Lagrange  $\lambda_i$  sont inconnus. Cela se fait d'abord en définissant le vecteur réduit

$$\dot{\eta} = \begin{bmatrix} \dot{\Phi}_R \\ \dot{\Phi}_L \end{bmatrix} \quad (\text{II.39})$$

Ensuite, en exprimant les vitesses de coordonnées généralisées à l'aide du modèle cinématique avant (II.19).

Ensuite nous avons

$$\begin{bmatrix} \dot{y}_a \\ \dot{\theta} \\ \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} = \frac{1}{2} \begin{bmatrix} R \cos \theta & R \cos \theta \\ R \sin \theta & R \sin \theta \\ \frac{R}{L} & -\frac{R}{L} \\ 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \quad (\text{II .40})$$

Cela peut s'écrire sous la forme

$$\dot{q} = S(q)\eta \quad (\text{II .41})$$

On peut vérifier que la matrice de transformation  $S(q)$  est nulle l'espace de la matrice de contraintes  $\Lambda(q)$ . Par conséquent nous avons

$$S^T(q)\Lambda^T(q) = 0 \quad (\text{II .42})$$

Ensuite, en prenant la dérivée temporelle de l'équation (38) donne

$$\ddot{q} = \dot{S}(q)\eta + S(q)\dot{\eta} \quad (\text{II .43})$$

En remplaçant les équations (38) et (40) dans l'équation principale (40), nous obtenons :

$$M(q)[\dot{S}(q)\eta + S(q)\dot{\eta}] + V(q, \dot{q}[S(q)\eta]) = B(q)\tau - \Lambda^T(q)\lambda \quad (\text{II .44})$$

Ensuite, en réarrangeant l'équation et en multipliant les deux côtés par des avances pour

$$S^T(q)M(q)S(q)\dot{\eta} + S^T(q)[M(q)\dot{S}(q) + V(q, \dot{q})S(q)]\eta = S^T(q)B(q)\tau - S^T(q)\Lambda^T(q)\lambda \quad (\text{II.45})$$

Où le dernier terme est identiquement nul. On définit maintenant les nouvelles matrices

$$\bar{M}(q) = S^T(q)M(q)S(q)$$

$$\bar{V}(q) = S^T(q)M(q)\dot{S}(q) + S^T(q)V(q, \dot{q})S(q)$$

$$\bar{B} = S^T(q)B(q)$$

Les équations dynamiques sont réduites à la forme

$$\bar{M}(q)\dot{\eta} + \bar{V}(q, \dot{q})\eta = \bar{B}(q)\tau \quad (\text{II .46})$$

Où

$$\bar{M}(q) = \begin{bmatrix} I_w + \frac{R^2}{4L^2}(mL^2 + I) & \frac{R^2}{4L^2}(mL^2 - I) \\ \frac{R^2}{4L^2}(mL^2 - I) & I_w + \frac{R^2}{4L^2}(mL^2 + I) \end{bmatrix}$$



$$\bar{V}(q, \dot{q}) = \begin{bmatrix} 0 & \frac{R^2}{2L} m_c d\dot{\theta} \\ -\frac{R^2}{2L} m_c d\dot{\theta} & 0 \end{bmatrix}, \quad \bar{B}(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

### I.1.4 Modélisation des actionneurs

Les moteurs à courant continu qui sont généralement utilisés pour entraîner les roues d'un système de robot mobile à entraînement différentiel sont considérés comme le cerveau actionneur. Dans un moteur à courant continu commandé par induit, ce qui est le cas pour notre système DDMR, la tension d'induit  $v_a$  est utilisé comme entrée de commande tout en maintenant constantes les conditions dans le circuit de champ. En particulier, pour un moteur à courant continu à aimant permanent, nous avons les équations suivantes pour le circuit d'induit.

$$\begin{cases} v_a = R_a i_a + L_a \frac{di_a}{dt} + e_a \\ e_a = K_b \omega_m \\ \tau_m = K_t i_a \\ \tau = N \tau_m \end{cases} \quad (\text{II.47})$$

Où,  $i_a$  est le courant d'induit,  $(R_a, L_a)$  est la résistance et l'inductance de l'enroulement d'induit respectivement,  $e_a$  est la force contre-électromotrice,  $\omega_m$  est le rotor vitesse angulaire,  $\tau_m$  est le couple moteur,  $(K_t, K_b)$  sont la constante de couple et constante contre la fem respectivement,  $N$  est le rapport de démultiplication, et est  $\tau$  le couple de sortie appliqué à la roue.

### Conclusion

La modélisation cinématique et dynamique d'un robot mobile à deux roues est présentée dans ce chapitre. Le modèle dynamique obtenu est non linéaire, ceci suggère l'application d'une loi de commande sphéristique pour permettre au robot de se mouvoir, suivre des trajectoires et éviter des obstacles. Ceci est l'objet du chapitre 3.

# Chapitre III

## Commande et planification de trajectoire basée sur l'intelligence des plantes

## chapitreIII Commande et planification de trajectoire

### III.1 Introduction

Dans le chapitre 1 de ce mémoire, nous avons présenté l'algorithme APGA et dans le chapitre 2 nous avons procédé à la modélisation du robot mobile. Dans ce chapitre, nous allons nous focaliser sur la commande du robot en utilisant l'algorithme APGA et la commande PID.

### III.2 La commande PID

#### III.2.1 Description d'un contrôleur PID

Un régulateur PID ou correcteur PID (proportionnel, intégral, dérivé) est une structure de calculs dont, le rôle est d'envoyer un signal de commande au système tout en respectant certaines exigences tant au niveau des limitations du système qu'à des caractéristiques spécifiques de sa réponse et c'est un algorithme de contrôle qui permet d'améliorer les performances d'un asservissement, c'est-à-dire un système ou procédé en boucle fermée [25].

Parmi les avantages de ce régulateur, nous citons [26].

✓ Structure simple.

✓ Bonne performance dans plusieurs processus.

✓ Fiable, même sans un modèle spécifique du système de contrôle.

Il est basé sur trois opérations :

- Coefficients de gain proportionnel : l'erreur est multipliée par un gain  $K_p$  pour améliorer la rapidité du système, plus  $K_p$  est grand plus le temps de réponse diminue, plus dépassement augment et stabilité du système est détériorée.

- Coefficients de gain intégral : l'erreur est intégrée sur un intervalle de temps  $t$ , puis multipliée par un gain  $K_i$  pour éliminer l'erreur résiduelle en régime permanent (erreur statique) et améliore la précision mais, ceci provoque l'augmentation du déphasage ce coefficient est plus grand, le système sera plus lent et oscillatoire.

- Coefficients de gain dérivé : l'erreur est dérivée suivant un temps  $t$ , puis multipliée par un gain  $K_d$  ce qui accélère la réponse du système et améliore la stabilité de la boucle [27].

#### III.2.2 L'équation d'un PID

Les calculs pour délivrer le signal de commande sont basés sur la différence entre sa consigne (position/état souhaité) et sa mesure (position/état actuel du système). Cette différence est appelée : erreur  $e(t)$ .

Ils existent plusieurs architectures pour combiner les trois actions d'un régulateur PID (série, parallèle ou mixte) [28].

Le modèle du PID utilisé dans ce travail est le suivant :

$$command = k_p e + k_i \int e + k_d \frac{e}{dt} \quad (III .1)$$

Le réglage d'un PID consiste à déterminer les coefficients  $K_p$ ,  $K_i$  et  $K_d$ , afin de répondre au cahier des charges préétabli, assurer les performances en termes de stabilité, rapidité et précision.

La correction se fait à base de l'erreur observée qui est la différence entre la consigne (Valeur désirée) et la mesure (valeur réelle).

$$e = consigne - mesure \quad (III .2)$$

### III.3 L'implémentation sur Simulink

Le modèles cinématique dynamique et le modèle des actionneurs obtenu dans le chapitre II sont implémentés ici sur le Simulink.

La figure (III .1) présente le modèle cinématique sur Simulink.

La figure (III. 2) présente le modèle dynamique sur Simulink.

La figure (III .3) présente le modèle de moteur droite sur Simulink.

La figure (III .4) présente le modèle de moteur gauche sur Simulink.

Dans les deux figures (III.3 et III.4) on a utilisé le contrôleur PID.

Le contrôleur PID utilise un retour d'information pour ajuster en temps réel la tension appliquée au moteur, en fonction de l'écart entre la valeur de consigne de la valeur réelle et la valeur mesurée, et pour assurer la précision de la vitesse des deux moteurs.

Les paramètres de robot utilisées : (voir annexe A)

### Kinematic Model

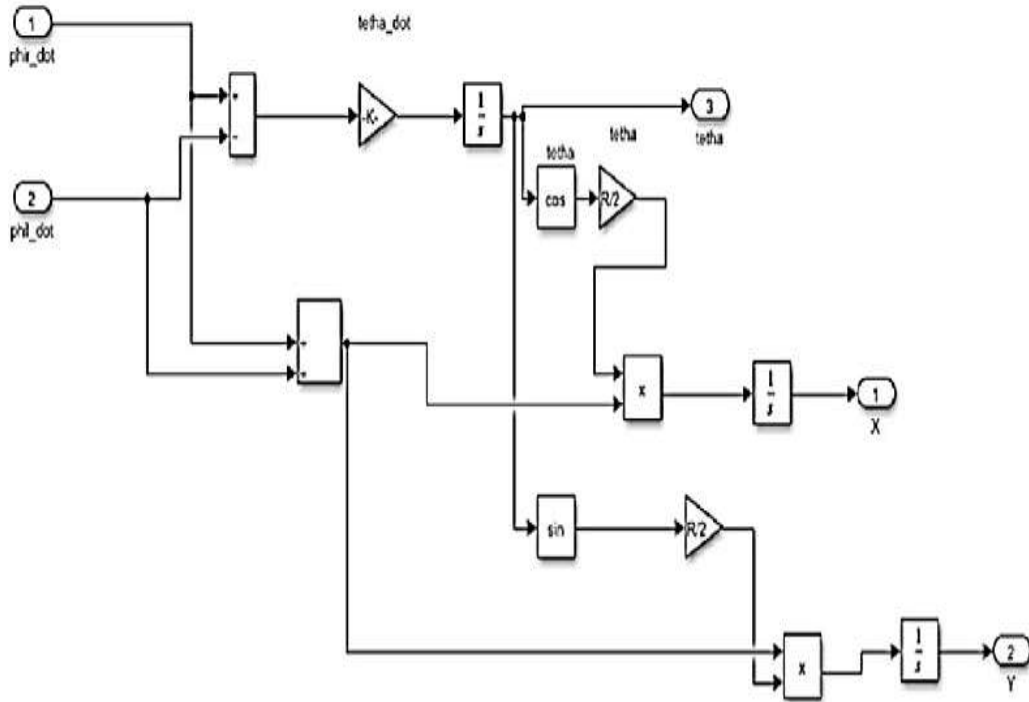


Figure III:1: modèle cinématique sous Simulink

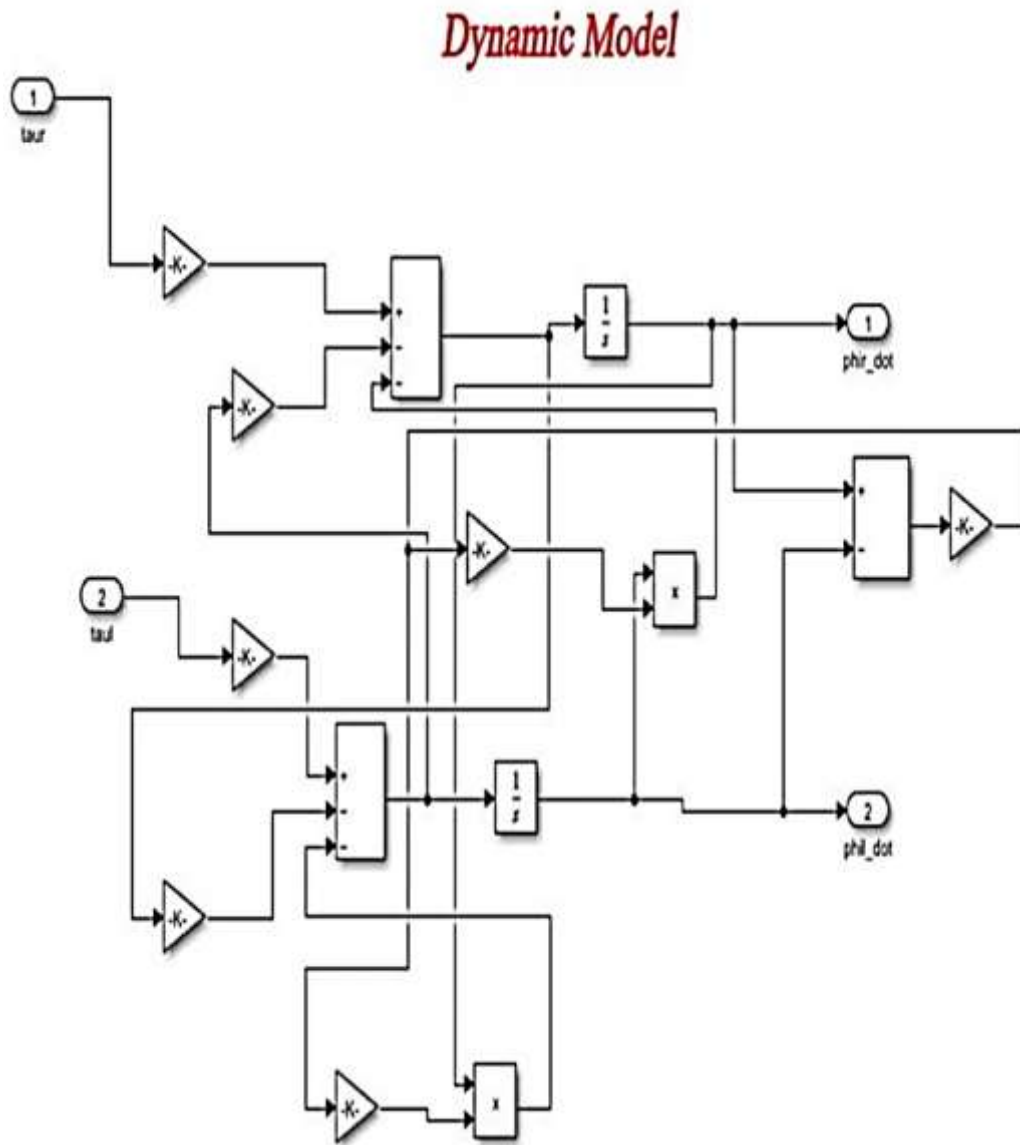


Figure III:2: modèle dynamique sous Simulink

### Right Motor

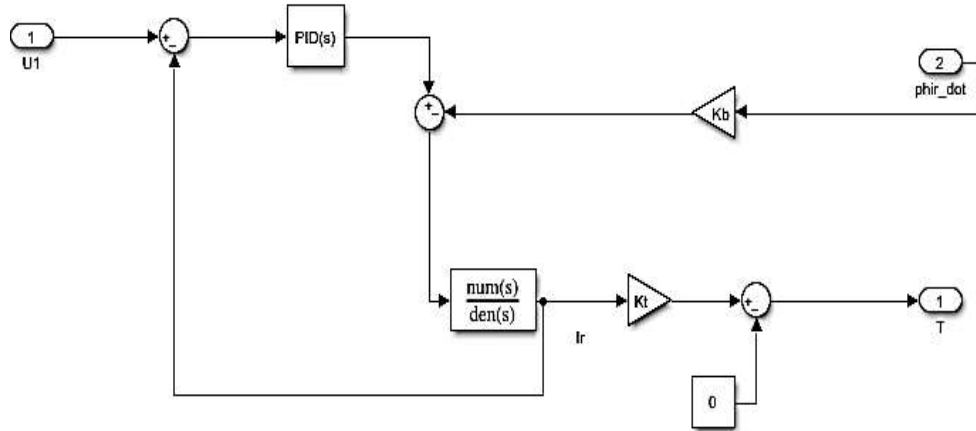


Figure III:3: modèle de moteur droit sous Simulink

### Left Motor

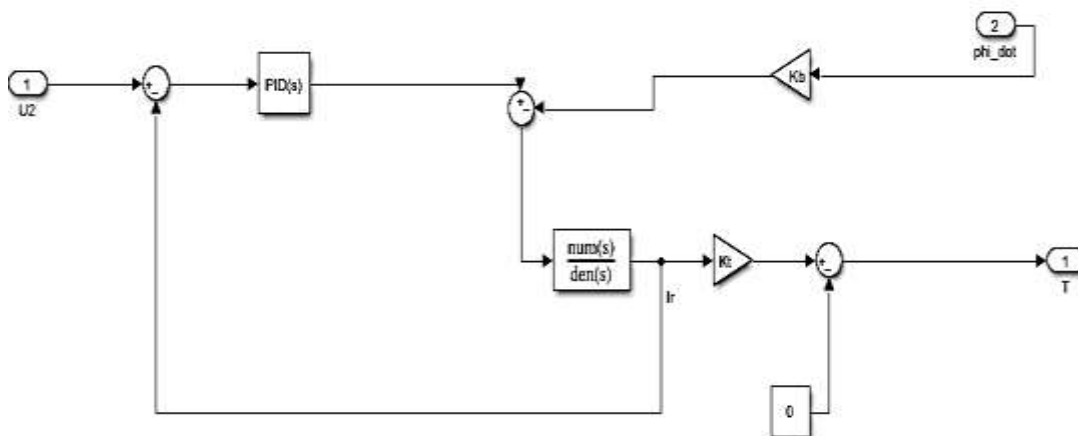


Figure III:4: modèle de moteur gauche sous Simulink

**Commande du mouvement :**

La commande du mouvement se base sur le calcul de la vitesse linéaire  $v$  du robot et de sa vitesse angulaire  $w$  en fonction de sa position actuelle notée  $(x_0, y_0)$  et de la position désirée notée  $(x, y)$ . Cette commande suit les étapes suivantes :

Calcul de la différence entre la nouvelle position et l'état actuelle du robot

$$dx = x - x_0 \quad (\text{III .4})$$

$$dy = y - y_0 \quad (\text{III .5})$$

Ensuite le calcul de la distance entre les deux points

$$d = \sqrt{dx^2 + dy^2} \quad (\text{III .6})$$

A la fin le calcul de vitesse linéaire et angulaire

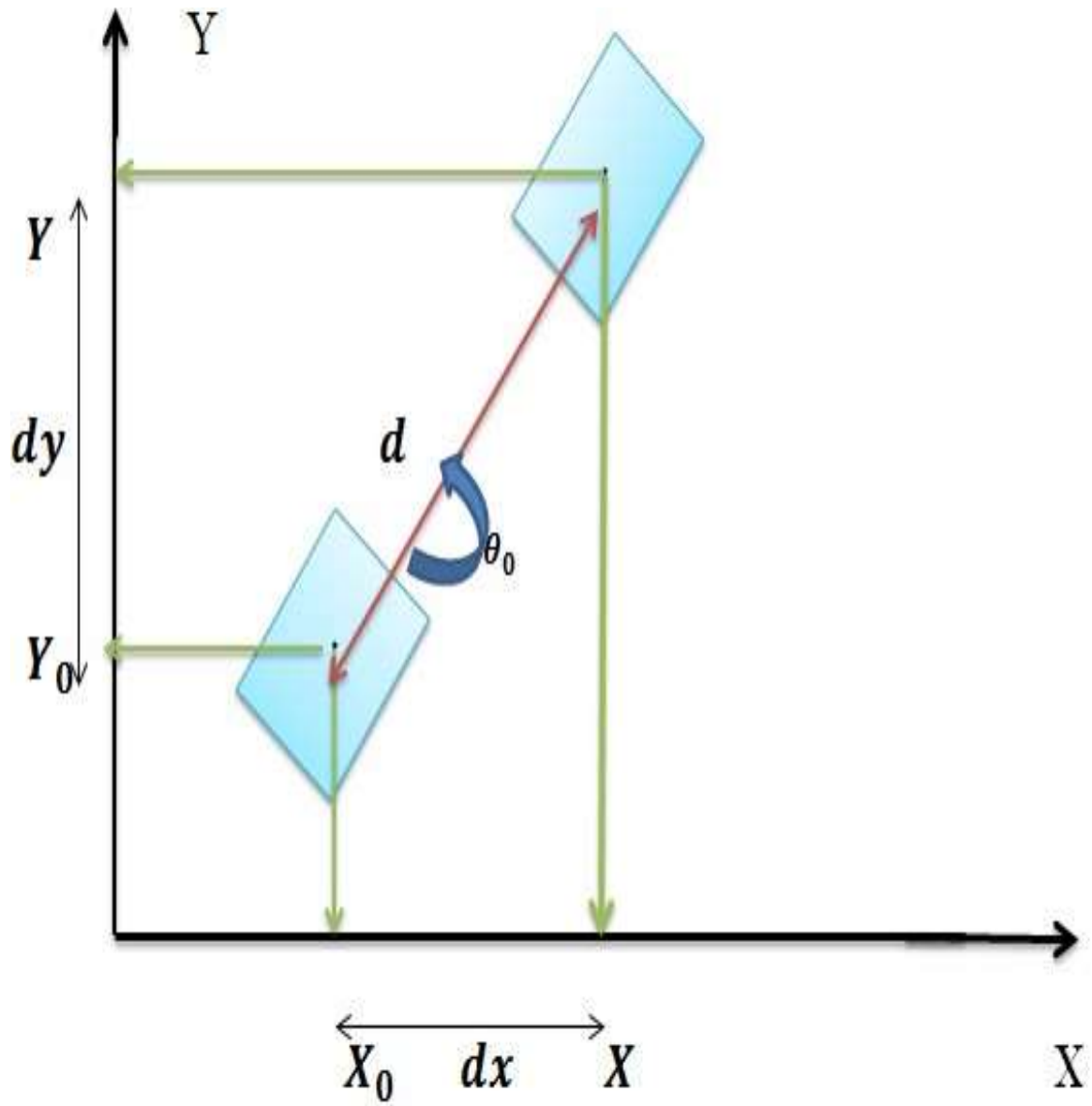
$$v = 0.5 * d \quad (\text{III .7})$$

$$w = 2 * \text{angle} \quad (\text{III .8})$$

Angle : la différence entre le nouvel angle d'orientation du robot et l'angle d'orientation actuel.

Comme le schéma si dessus





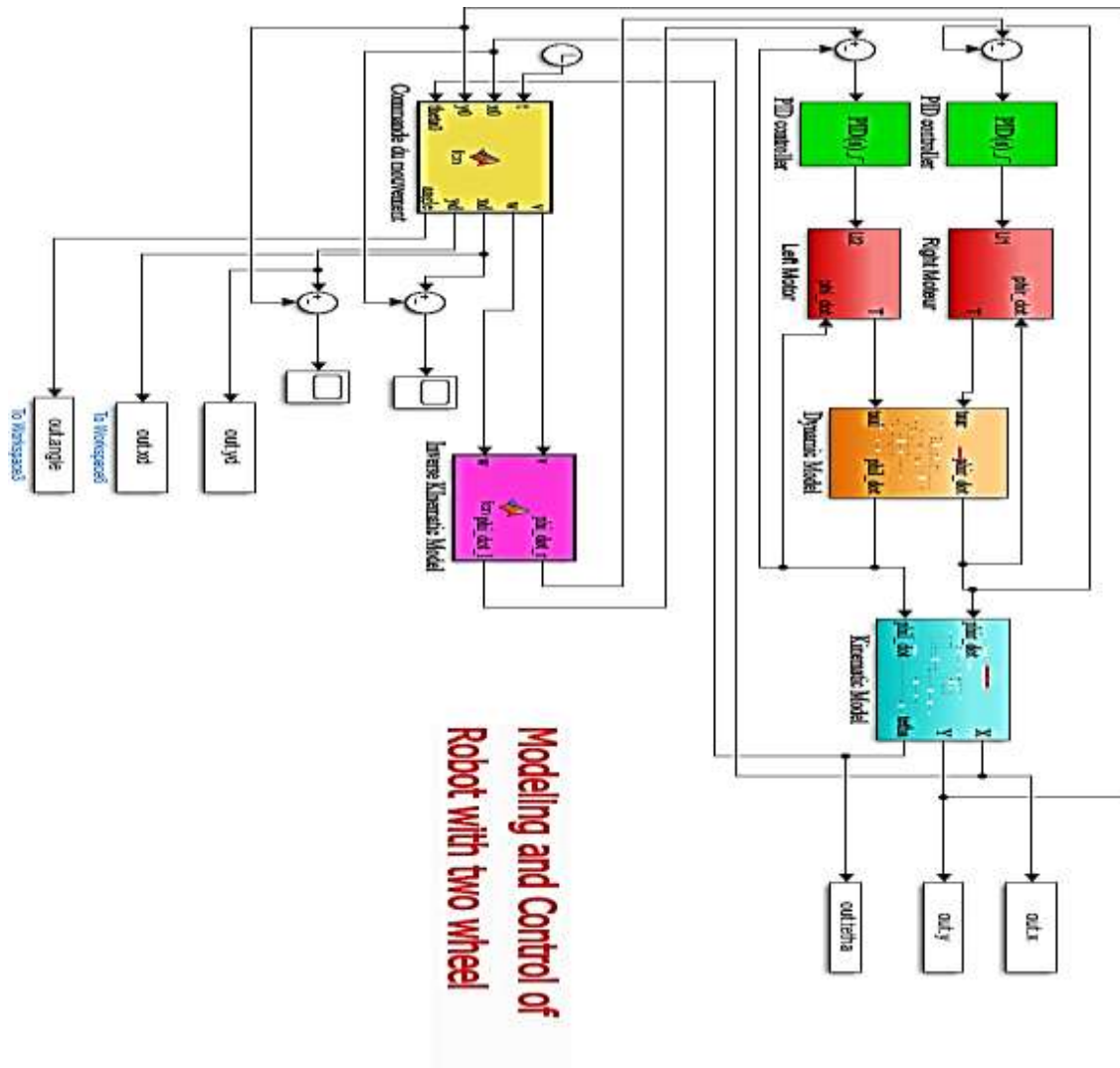


Figure III:5: la commande globale de robot

Le schéma global de la commande présente un bloc de commande de mouvement qui calcule la vitesse linéaire et angulaire, qui est utilisée par la suite par le bloc de modèle cinématique inverse pour calculer la vitesse angle de la roue droite et gauche, et le PID assure la poursuite de la trajectoire du robot quand n'a choisi, les deux moteurs vas mettre en marche le robot ainsi le modèle dynamique vas contrôler le mouvement et le modèle cinématique donne l'état actuel du robot mobile.

### III.3.1 Les paramètres de contrôleur PID utilisés

Les paramètres utilisés pour le contrôleur sont réglés à partir de plusieurs expériences jusqu'à trouver les valeurs exactes dont on a besoin pour la poursuite de la trajectoire du robot.

### III.3.2 Commande en vue de poursuite d'une trajectoire circulaire

#### III.3.2.1 Planification de trajectoire avec PID

Nous testons dans cette section la poursuite d'une trajectoire circulaire en utilisant le contrôleur PID (voir figure (III .5)). La position initiale du robot ( $x_0=0, y_0=0$ ). La durée de la trajectoire est fixée à 300 seconds.

L'objectif de cette commande est d'analyser la poursuite de la trajectoire par le PID.

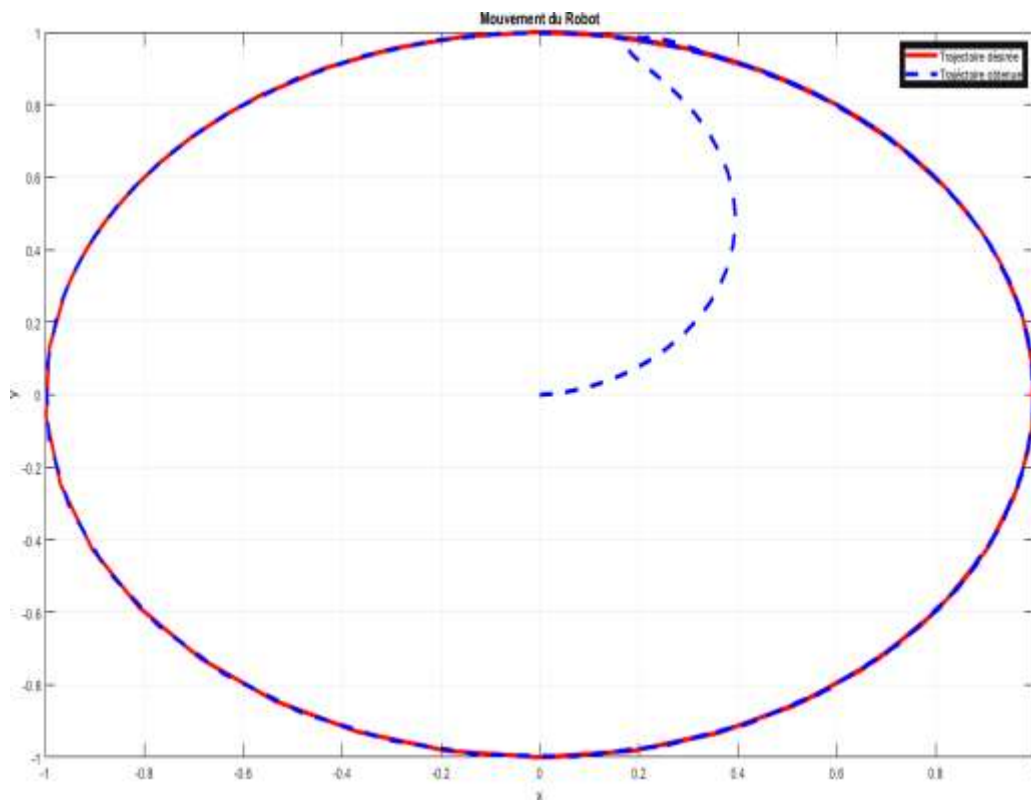


Figure III:6: la trajectoire du robot obtenu.

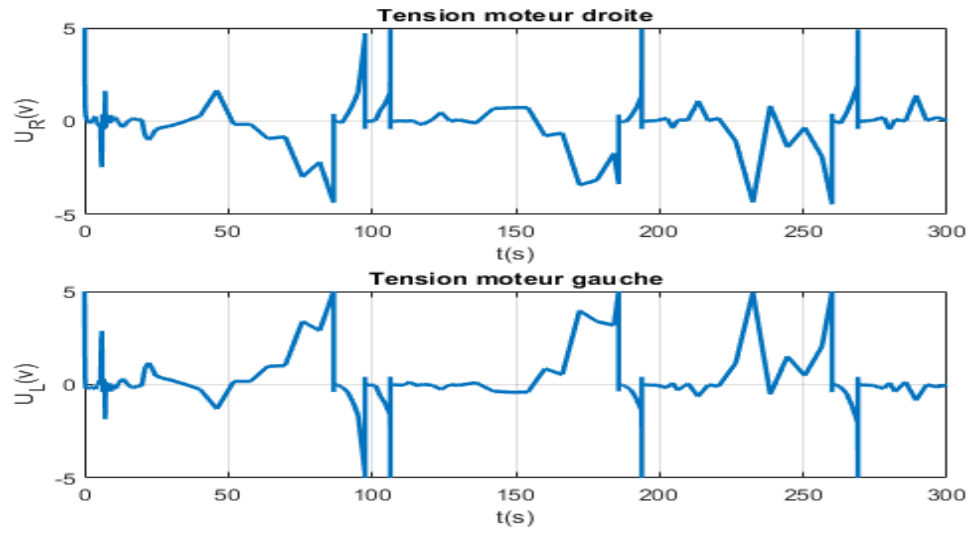


Figure III:7: la tension générée au deux moteurs.

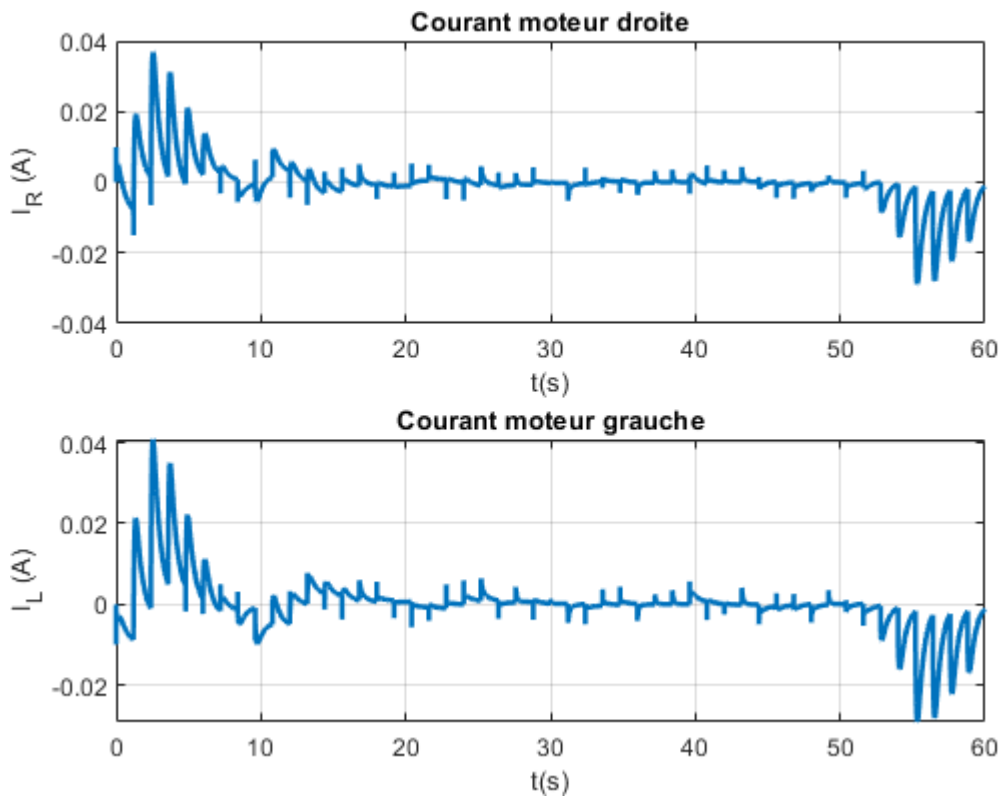


Figure III:8: Le courant des deux moteurs.

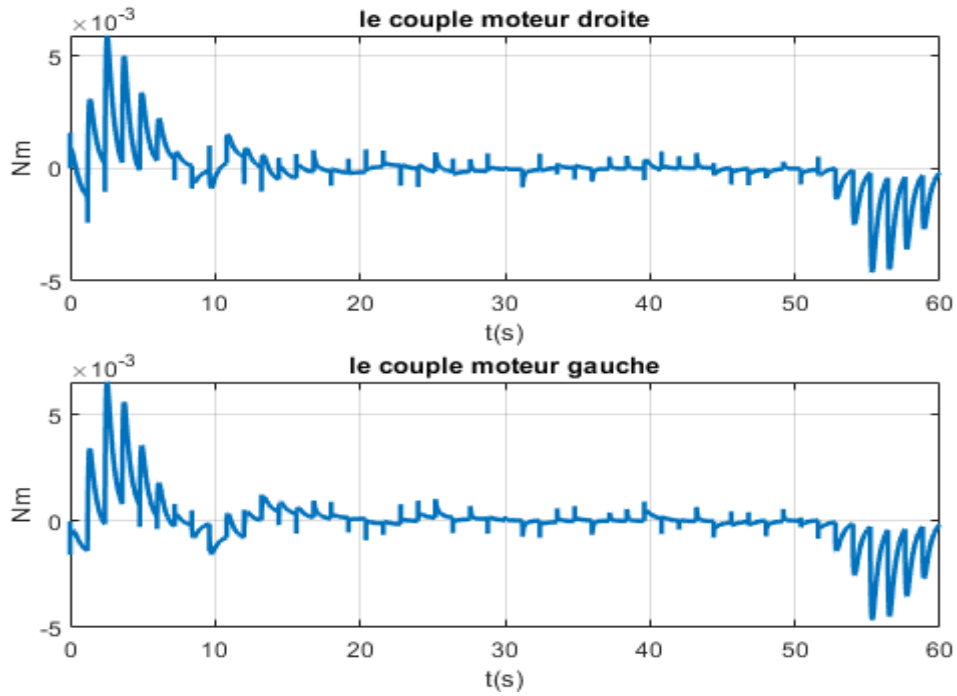


Figure III:9: Le couple des deux moteurs

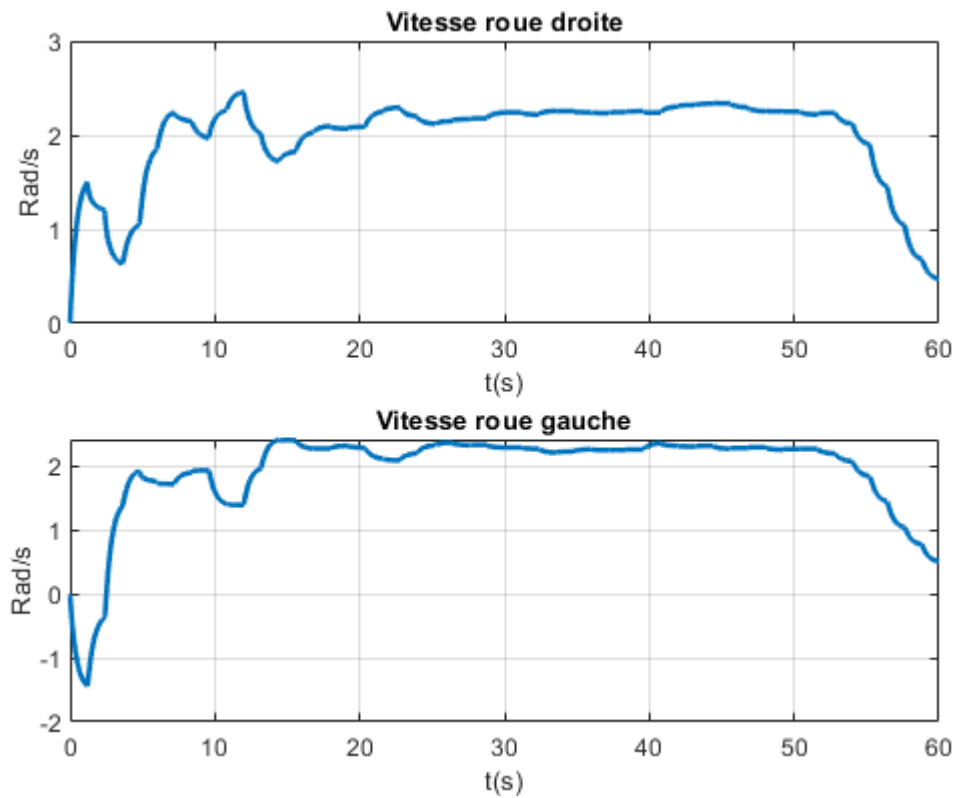


Figure III:10: Les vitesses des deux roues.

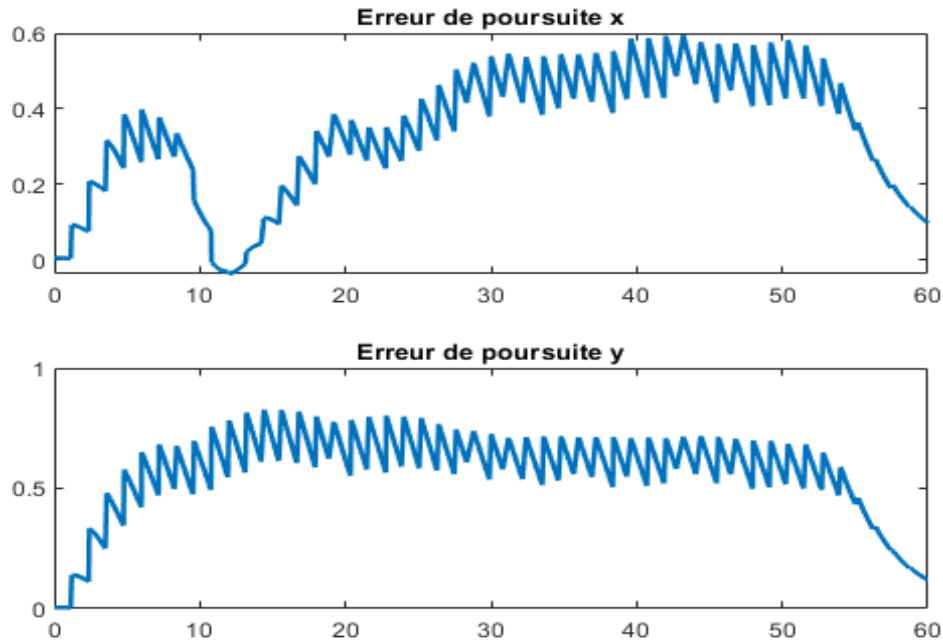


Figure III:11 l'erreur de poursuite

### III.3.2.2 Interprétations des résultats obtenues

La figure (III .6) présente la trajectoire obtenue et la trajectoire désirée, on remarque que le robot arrive à suivre la trajectoire. La figure (III .7) présente la tension appliquée aux deux moteurs et les valeurs de tension trouvée sont réalisables (Acceptables).

La figure (III .8) présente les courants appliqués aux deux moteurs. La figure (III .9) présente les couples moteurs générés. La figure (III .10) présente les vitesses des deux roues. La figure (III .11) montre l'erreur de poursuite on remarque que l'erreur est faible.

### III.4 Planification de trajectoire et évitement d'obstacles par APGA

Dans cette partie de la commande, nous utilisons l'algorithme méta heuristique APGA pour contrôler notre robot et éviter les obstacles. Pour ce faire, nous définissons une fonction fitness qui évalue la qualité de chaque point de l'espace du robot.

$$S = \sum_{j=1}^N 5000 * \exp^{-5\|obj(j)-(x-1)^2\|} \quad (III.9)$$

$$fitness = 100 * \|(dist - x)^2\| + S \quad (III.10)$$

N : la dimension de vecteur des obstacles.

Obs : les obstacles.

Dist : la destination.

X : le point qui prévoie

l'APGA.J : indice de l'obstacle.

La courbe de la fonction fitness, pour 5 obstacles, est donnée dans la figure 32

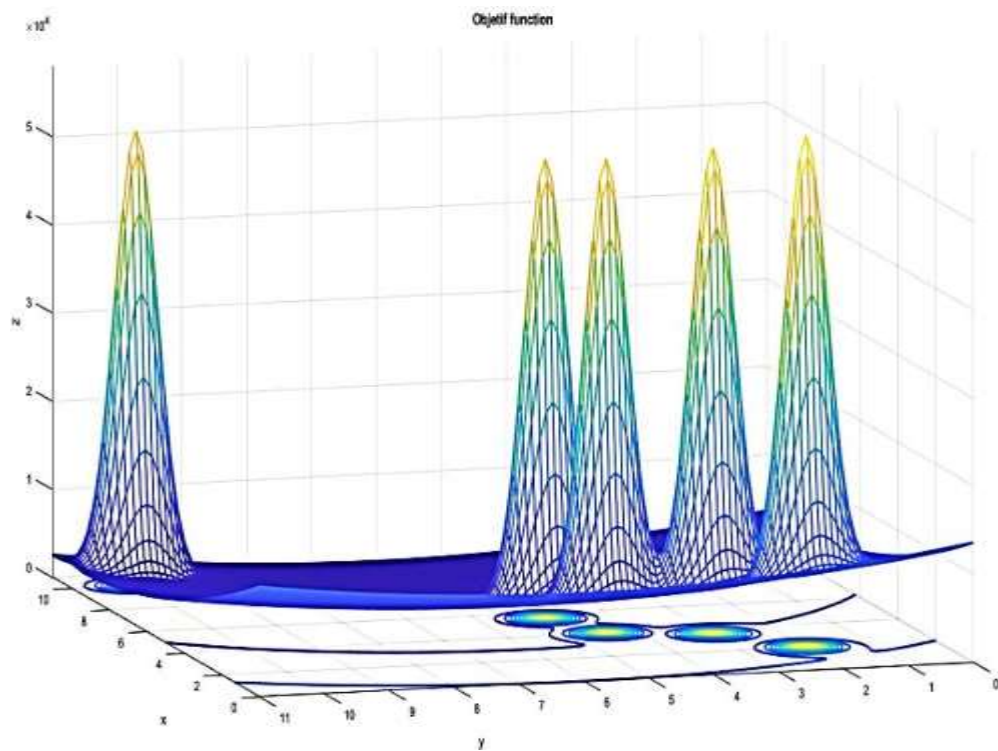


Figure III:12la fonction fitness

#### III.4.1 Planifications de la trajectoire avec APGA

L'algorithme APGA présenté précédemment est utilisé ici pour rechercher un chemin permettant au robot de se déplacer d'un point initial à un point final tout en évitant les collisions avec des obstacles. La position initiale du robot ( $x_0=0$ ,  $y_0=0$ ) et la position de destination ( $x=6$ ,  $y=10$ ). Plusieurs obstacles sont présents sur le chemin du robot.

#### III.4.2 Résultats et discussion

Les figures suivantes illustrent les résultats obtenus lors de la planification d'une trajectoire en utilisant l'algorithme précédent (APGA). Ces résultats ont été obtenus en variant les ensembles de paramètres appliqués à l'algorithme.

##### Expérience 1 :

La taille de la population est  $N=30$ . A chaque itération, l'APGA dicte au robot le point suivant qu'il doit atteindre. Ce point est le meilleur individu dans la population.

La figure suivante représente la première expérience de la trajectoire prévue à l'aide de l'algorithme APGA et sa poursuite par le robot.



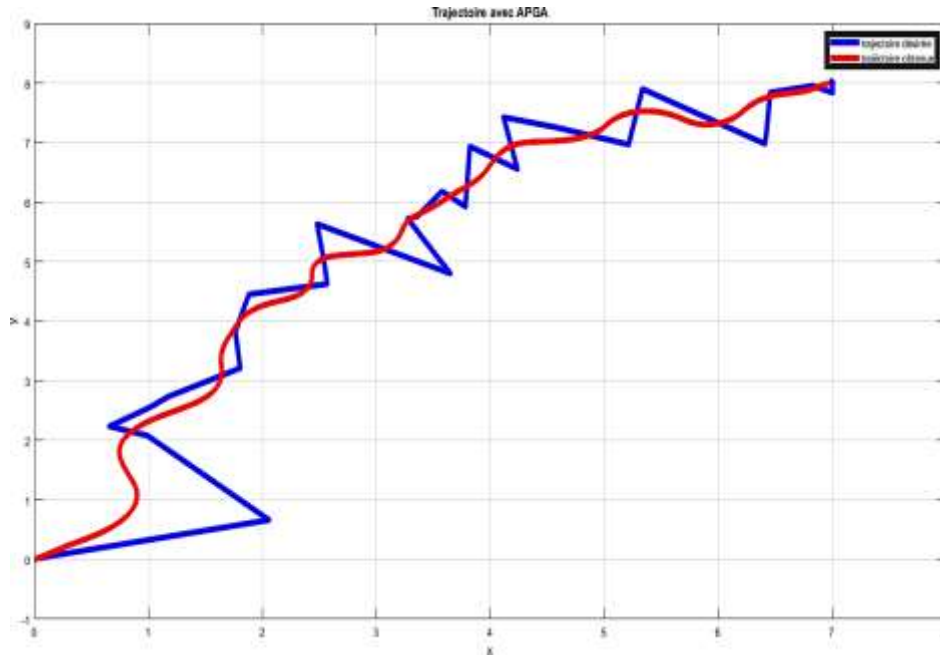


Figure III:13 la trajectoire planifiée

On constate un mouvement pseudo aléatoire et une trajectoire non lisse. Ce qui n'est pas souhaitable.

Expérience 2 :

La taille de la population est  $N=60$ . A chaque itération, l'APGA dicte au robot le point suivant qu'il doit atteindre. Ce point est la moyenne de la population. On constate que la trajectoire est beaucoup plus lisse que le cas précédent.

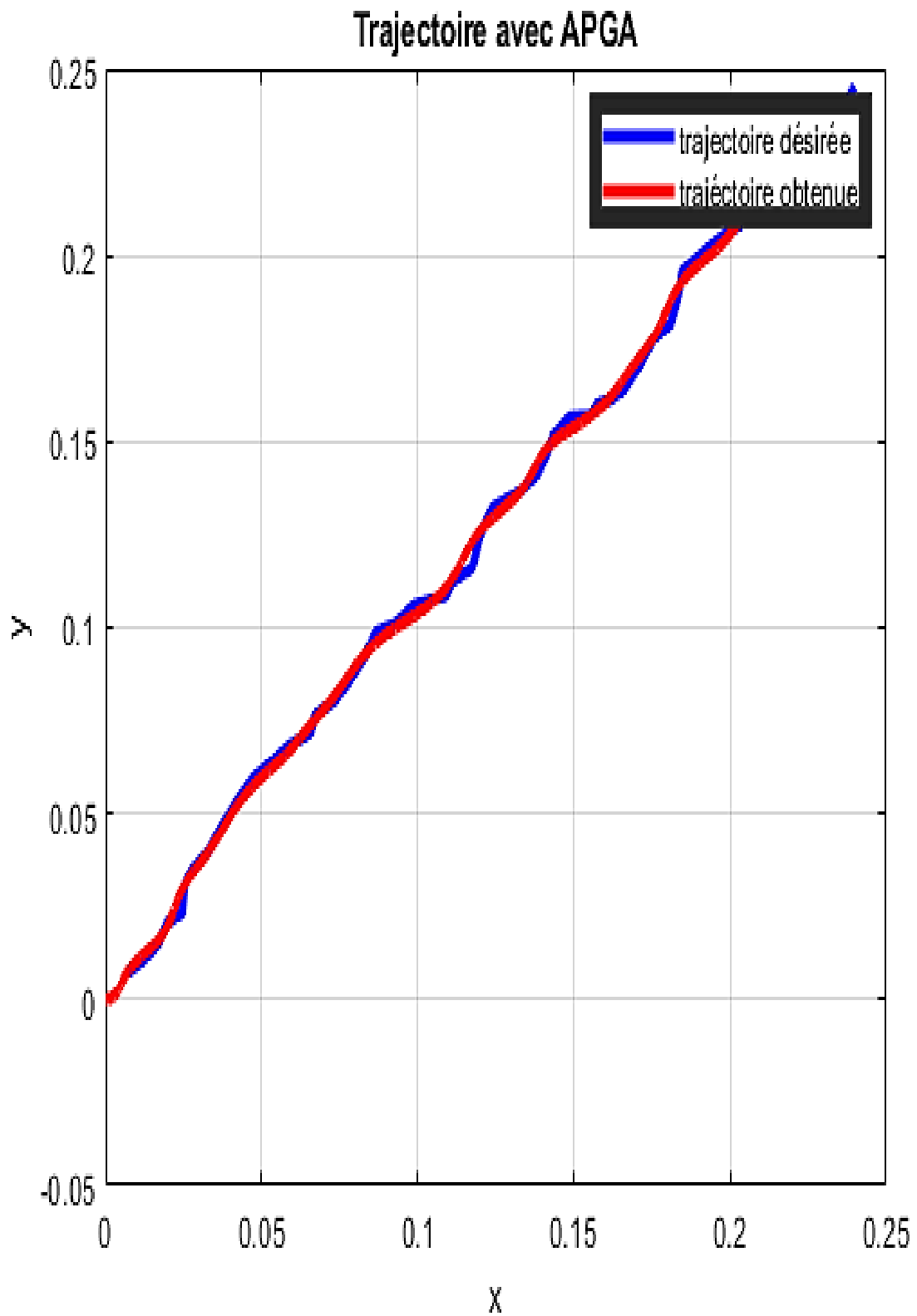


Figure III:14 la trajectoire planifié

Expérience 3 :

La taille de la population est  $N=6000$ . A chaque itération, l'APGA dicte au robot le point suivant qu'il doit atteindre. Ce point est la moyenne de la population

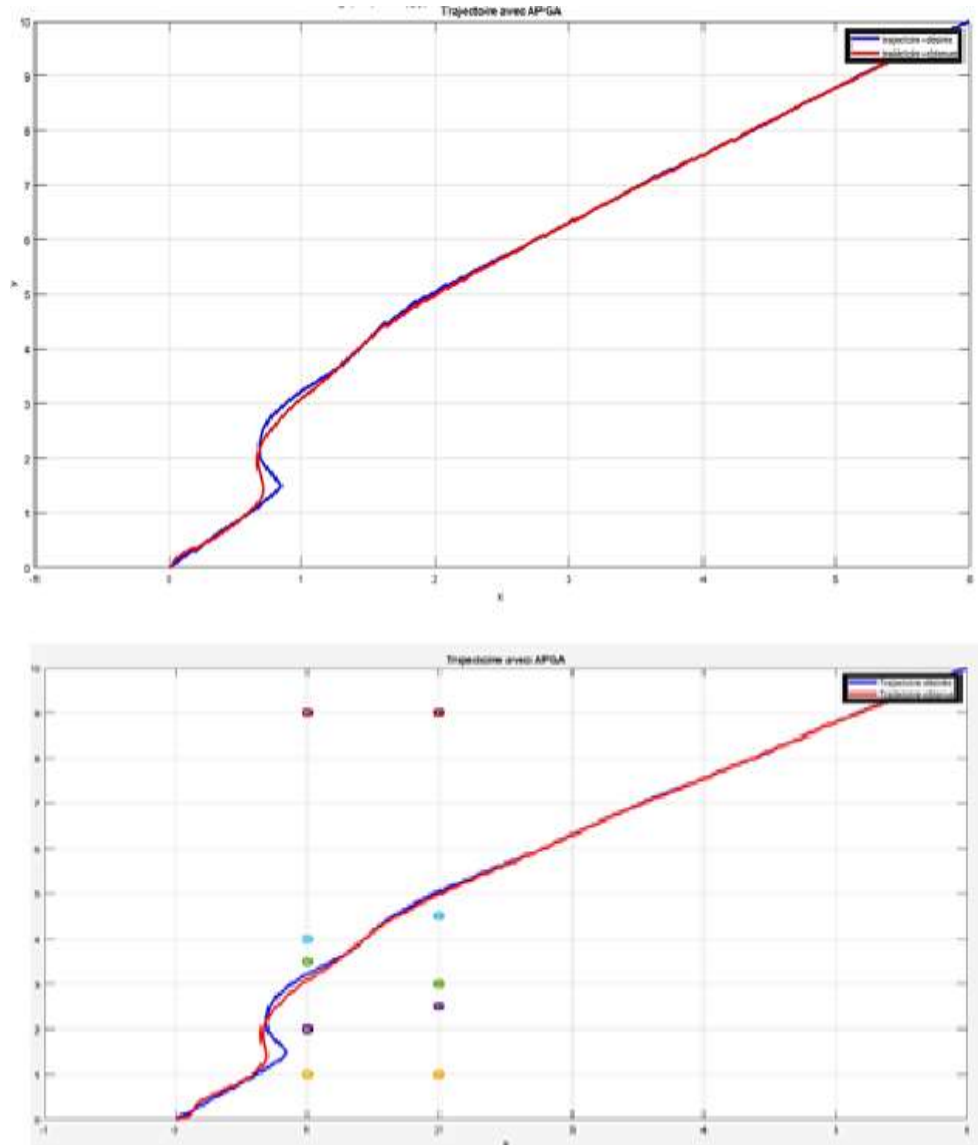


Figure III:15 La trajectoire planifiée avec APGA

On constate que pour ces derniers cas la trajectoire est lisse et l'algorithme APGA permet d'éviter les obstacles présents dans l'environnement du robot.

Dans la suite de ce travail, nous allons présenter la tension, le courant et les vitesses des deux roues pour mieux approfondir et valider le comportement d'APGA du robot.

Pour mieux approfondir et valider le comportement d'APGA et de la commande utilisée, on présente les résultats des courants, tensions, couple moteurs etc.

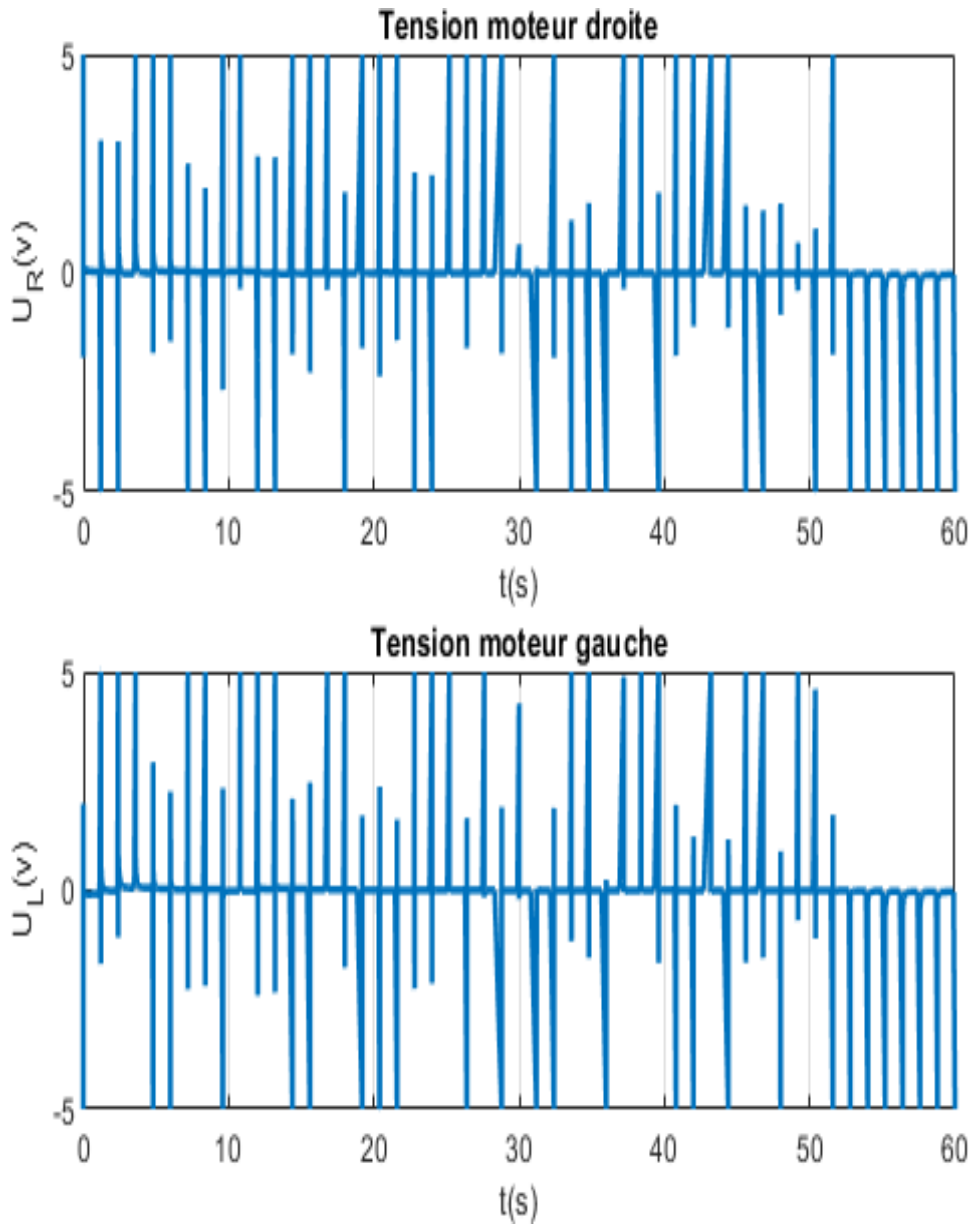


Figure III:16 les tensions des deux moteurs

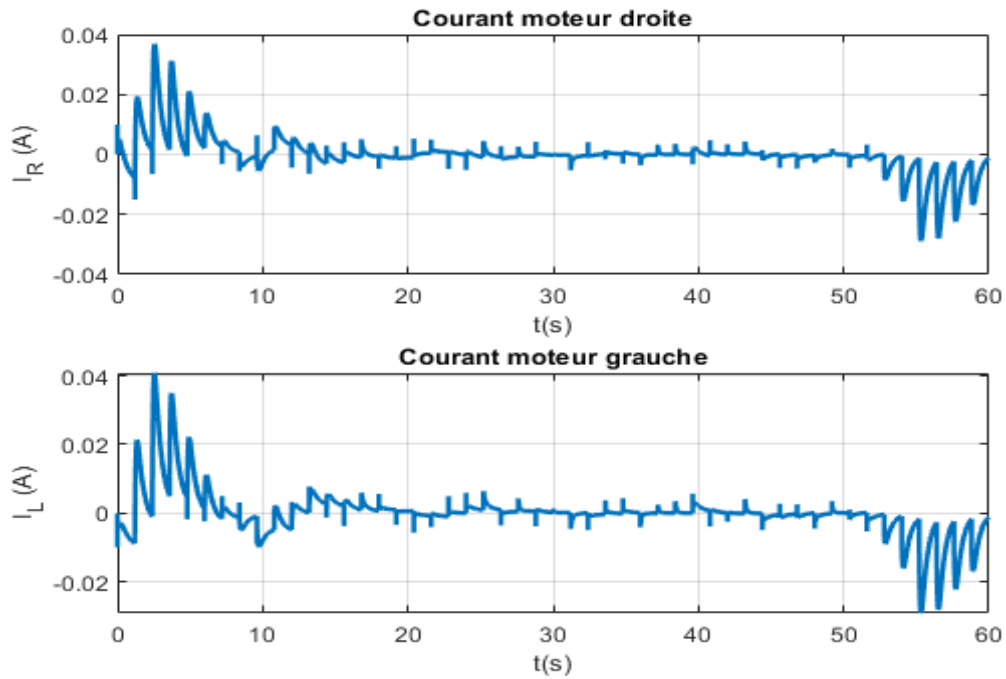


Figure III:17le courant des deux moteurs

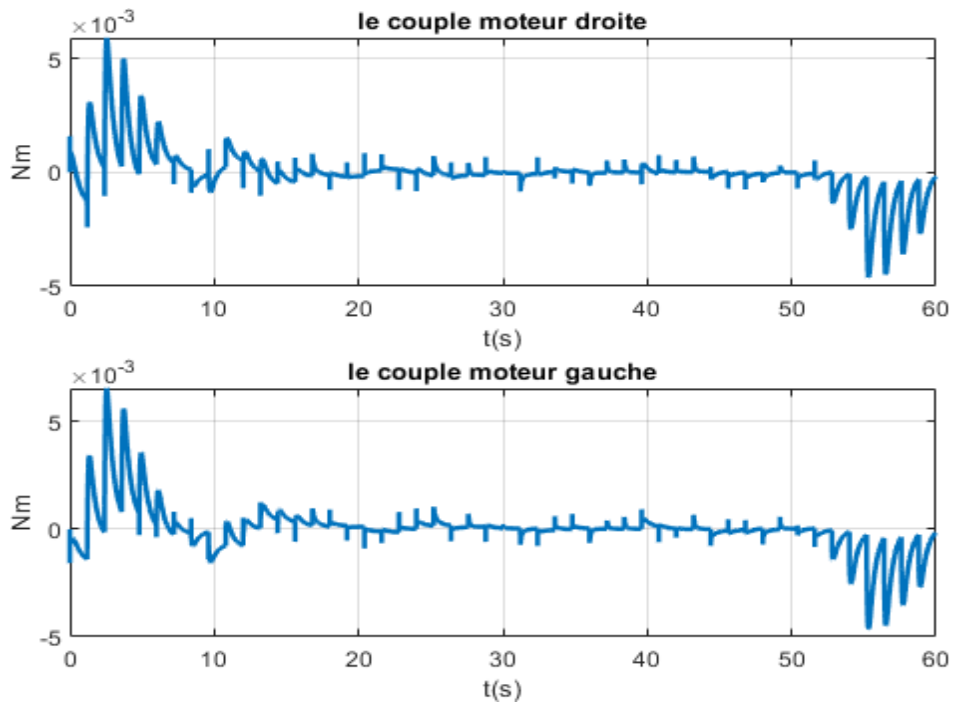


Figure III:18le coupler des deux moteurs

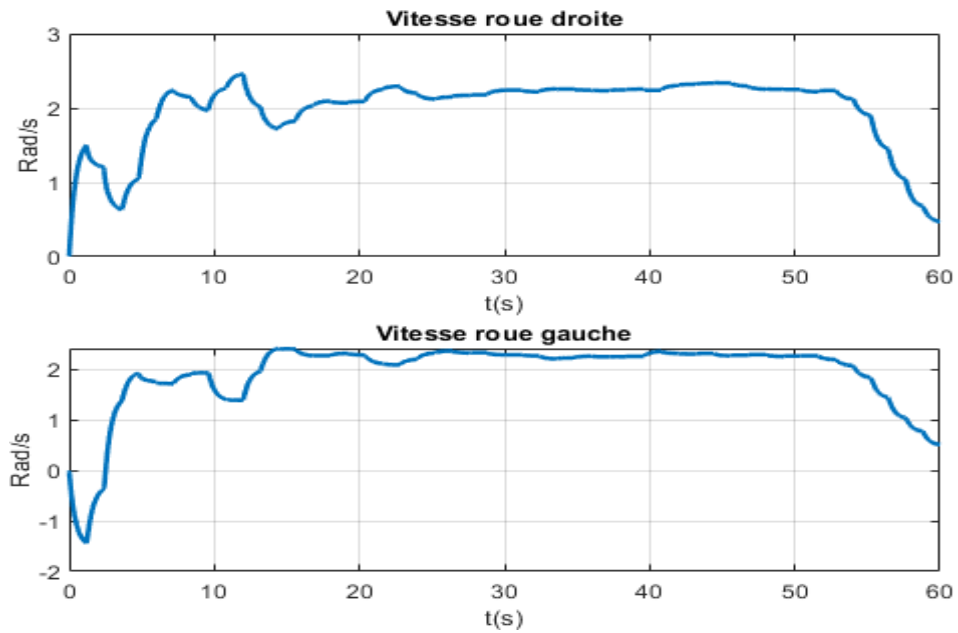


Figure III:19 la vitesse des deux roues

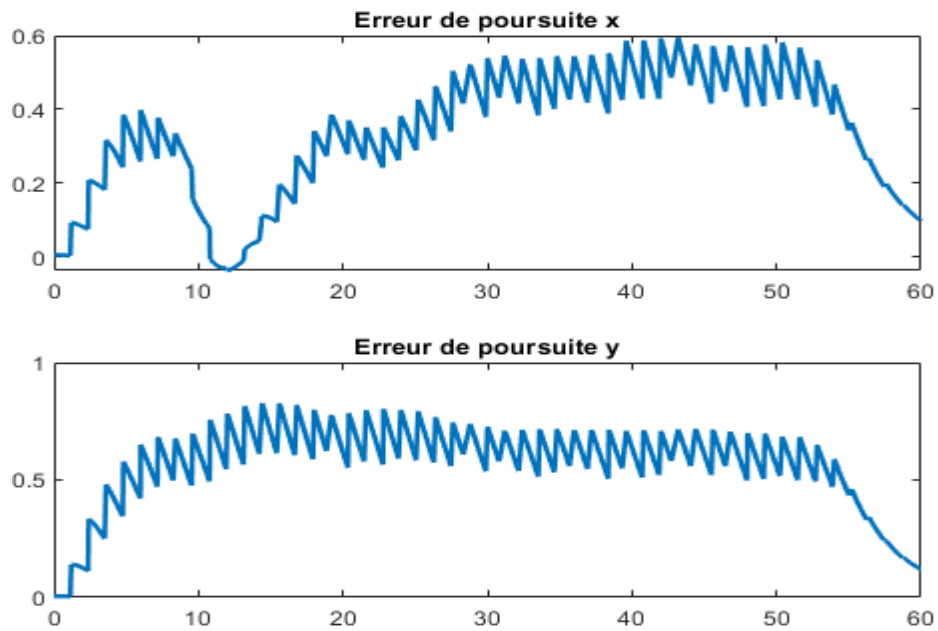


Figure III:20 l'erreur de poursuite

### **III.4.3 Interprétations des résultats**

- Les figures (III .13, III .14 et III .15) montrent la trajectoire planifiée par l'algorithme APGA, allant de la position initiale à la position finale, en vue d'éviter tous les obstacles sur le chemin.

-La figure (III .13) représente de la trajectoire prévue et met en évidence un grand nombre de pics, ce qui rend la trajectoire souhaitée difficile à suivre par le robot. Elle présente un comportement stochastique.

-La figure (III .14) représente la trajectoire obtenue après avoir augmenté la taille de la population et en utilisant la moyenne de la population come point objectif à chaque itération. Cette amélioration a permis d'obtenir une trajectoire plus lisse, ce qui facilite le suivi de la trajectoire par le robot.

-La figure (III .15) illustre les résultats obtenus après avoir augmenté la taille de la population à 600. Cette combinaison a permis d'obtenir la meilleure trajectoire désirée, ce qui permet au robot de réaliser une poursuite parfaite, en suivant la trajectoire de manière précise et sans difficultés. L'évitement d'obstacle est achevé.

-La figure (III .15) présente la trajectoire planifiée par l'APGA au milieu des obstacles. Il est notable que les paramètres utilisés dans cet algorithme ont une influence significative sur le comportement de la trajectoire.

-La figure (III .16) présente la tension appliquée aux deux moteurs. La figure (III .17) présente Le courant appliqué aux deux moteurs. La figure (III .18) présente le couple moteur généré par les deux moteurs .La figure (III .19) présente les vitesses des deux roues. La figure (III .20) montre l'erreur de poursuite en remarque que l'erreur est assez faible.

L'algorithme APGA considère la planification de trajectoire comme un problème d'optimisation. Les obstacles présentent des pics (maximums) sur la fonction fitness, tandis que l'objectif (destination) est un minimum. L'algorithme par de la position initiale en choisissant des

population de grande taille, et on suivant l'évolution de sa moyenne, l'algorithme dessine une trajectoire à travers le temps. Cette dernière est suivie par le robot, étape par étape ; cette méthode est une méthode de planification active, elle s'opère en temps réel, en ligne. Cependant, il faut une analyse plus profonde pour valider toutes ses performances et comprendre tous ses inconvénients. Ceci peut faire l'objet d'études futures.

### **III.5 Conclusion**

Ce chapitre, présente l'application d'APGA pour la planification et la commande de mouvement du robot mobile en temps réelles (enligne). La poursuite de trajectoire est assurée par deux PID qui contrôlent les deux moteurs. L'algorithme APGA considère le problème de planification comme un problème d'optimisation où les obstacles sont des maximums de la fonction objectif et la destination est un minimum. En utilisant une grande population, l'algorithme parvient à trouver une trajectoire lisse. La poursuite de trajectoire est assurée par un PID. L'approche étudiée présente de bons résultats. Cependant, une analyse plus approfondie est nécessaire



## Conclusion générale

Dans ce mémoire, nous avons exposé et mis en œuvre une méthode inspirée du comportement des plantes pour la planification de trajectoires et l'évitement d'obstacles d'un robot mobile. Notre objectif principal était d'exploiter les principes de l'intelligence végétale afin de développer une méthode de planification de trajectoire pour un robot mobile à deux roues. Notre travail s'est articulé autour de trois phases clés.

La première section du mémoire a été consacrée à la présentation de l'approche basée sur l'intelligence des plantes, en particulier l'algorithme IWO (Invasive Weed Optimization). Dans le deuxième chapitre, nous avons introduit les concepts généraux liés aux robots mobiles, puis nous avons procédé à la modélisation cinématique et dynamique du robot, avec une implémentation concrète réalisée dans l'environnement Simulink.

Dans le dernier chapitre de notre travail, nous avons exploré l'application de la méthode APGA pour la planification de trajectoires et le contrôle d'un robot mobile. Les résultats obtenus grâce à cette approche ont été concluants.

Plusieurs perspectives d'avenir se dessinent à partir de nos travaux :

- Mener des expérimentations dans des environnements variés pour évaluer la robustesse de notre méthode.
- Effectuer une analyse approfondie de la consommation énergétique du robot lors de ses déplacements.
- Explorer les possibilités d'optimisation des paramètres de l'algorithme, visant à améliorer les performances de la planification de trajectoires.

- Appliquer notre méthode à des situations pratiques et concrètes, en partenariat avec des entreprises ou des organismes de recherche.

## Références

- [1] En ligne KHELIFI, M., & GUENDOUZ, L. (2020). Un algorithme génétique pour la couverture des cibles dans les réseaux de Capteurs sans fil (These de doctorat, Université Ibn Khaldoun-Tiaret-).
- [2] OUMAIMA, M. M., ENA, M. S. B., RACHIQ, P. M. S., & AMRANI, P. M. K. L'UTILISATION DES PLANTES AROMATIQUES ET MÉDICINALES CONTRE LES ADVENTICES.
- [3] EL HOUDA, C. N. (2021). Utilisation des algorithmes métaheuristiques dans L'optimisation numérique.
- [4] OUMAIMA, M. M., ENA, M. S. B., RACHIQ, P. M. S., & AMRANI, P. M. K. L'UTILISATION DES PLANTES AROMATIQUES ET MÉDICINALES CONTRE LES ADVENTICES.
- [5] McCully, K. V., & Jensen, K. (2004). Guide de lutte intégrée: contre les mauvaises herbes dans les cultures de fraises. Nouveau Brunswick Agriculture, Pêches et Aquaculture.
- [6] Kenane, El Hadi. Contribution à l'optimisation d'un réseau d'antennes intelligentes au système UMTS. Diss. 2018
- [7] Zaharis, ZD, Lazaridis, PI, Cosmas, J., Skeberis, C. et Xenos, TD (2014). Synthèse d'un réseau d'antennes à gain élevé presque optimal avec inclinaison du lobe principal et remplissage nul à l'aide de l'optimisation des mauvaises herbes envahissantes initialisée par Taguchi. Transactions IEEE sur la radiodiffusion , 60 (1), 120-127.
- [8] Bouteraa, M. (2021). TECHNOLOGY FACULTY SCIENTIFIQUE WORKS (Doctoral dissertation, university of M'sila).
- [9] Sahraei-Ardakani, M., Roshanaei, M., Rahimi-Kian, A. et Lucas, C. (2008, décembre). Une étude de la dynamique du marché de l'électricité à l'aide de l'optimisation de la colonisation par les mauvaises herbes envahissantes. En 2008 IEEE Symposium On Computational Intelligence and Games (pp. 276-282). IEEE.
- [10] Rad, HS, & Lucas, C. (2007, septembre). Un système de recommandation basé sur un algorithme d'optimisation des mauvaises herbes envahissantes. En 2007, Congrès IEEE sur le calcul évolutif (pp. 4297-4304). IEEE.
- [11] Mehrabian, AR, & Yousefi-Koma, A. (2007). Positionnement optimal d'actionneurs

piézoélectriques sur une dérive intelligente à l'aide d'algorithmes bio-inspirés. *Science et technologie aérospatiales*, 11 (2-3), 174-182.

[12] Misaghi, M., & Yaghoobi, M. (2019). Algorithme amélioré d'optimisation des mauvaises herbes envahissantes (IWO) basé sur la théorie du chaos pour une conception optimale du contrôleur PID.

*Journal of Computational Design and Engineering*, 6 (3), 284-295.

[13] Sinem Akyol .artificial intelligence review 47 avril2017

[14] Bernard, M. J. Olivia Doppelt-Azeroual. To cite this version: HAL Id: tel.

[15] Dhaouadi, R., & Hatab, A. A. (2013). Dynamic modelling of differential-drive mobile robots using Lagrange and newton-Euler methodologies: A unified framework. *Advances in Robotiques & Automation*, 2(2), 1-7.

- [16] Farkh, R. (2011). *Commande PID des systèmes à retard* (Doctoral dissertation, École Nationale d'Ingénieurs de Tunis).
- [17] Deremetz, M. (2018). *Contribution à la modélisation et à la commande de robots mobiles autonomes et adaptables en milieux naturels* (Doctoral dissertation, Université Clermont Auvergne(2017-2020)).
- [18] Morette, N. (2009). *Contribution à la Navigation de robots mobiles: approche par modèle direct et commande prédictive* (Doctoral dissertation, Université d'Orléans).
- [19] Morette, N. (2009). *Contribution à la Navigation de robots mobiles : approche par modèle direct et commande prédictive* (Doctoral dissertation, Université d'Orléans).
- [20] Lens, S. (2008). *Locomotion d'un robot mobile*.
- [21] Djama, D., & Gasmia, D. (2022). *Réalisation de la partie commande d'un robot mobile minier* (Doctoral dissertation).
- [22] Djama, D., & Gasmia, D. (2022). *Réalisation de la partie commande d'un robot mobile minier* (Doctoral dissertation).
- [23] Chebili, Z. (2018). *Contribution à la Modélisation dynamique et la Commande des robots mobiles non-holonomes à roues*.
- [24] Dhaouadi, R., & Hatab, A. A. (2013). *Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework*. *Advances in Robotics & Automation*, 2(2), 1-7.
- [25] Dhaouadi, R., & Hatab, A. A. (2013). *Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework*. *Advances in Robotics & Automation*, 2(2), 1-7.
- [26] SEDINI, Chahrazed, and Nasre-Eddine CHERIGUI. *Conception et commande d'un quadrotor UAV à base d'Arduino*. Diss.
- [27] Viollet, S., Dupeyroux, J., & Serres, J. (2020). *Le robot fourmi AntBot : Conception et réalisation d'un robot bio-inspiré*. *Techniques & culture*, 73, 128-141.
- [28] Qian, Y. (2013). *Conception et Commande d'un Robot d'Assistance à la Personne* (Doctoral dissertation, Ecole centrale de Lille).

# Annexe (A)

**Table 1 les paramètres du robot**

Les parameters	La description	La valeur
R	La Rayon du roué	0.1 m
L	La largeur du robot	0.2 m
I w	Le moment d'inertie de chaque roue	0.0001 N.m
I	Le moment d'inertie totale	0.001 N.m
mc	La masse totale	1 Kg
l	Inductance de moteur	0.001 H
r	La resistance de moteur	1.9 $\Omega$

**Résumé :** Un robot mobile est un véhicule doté de moyens de locomotion qui lui permettent de se déplacer. Suivant son degré d'autonomie il peut être doté de moyens de perception et de raisonnement.

Ce mémoire aborde l'utilisation de l'intelligence des plantes pour développer une méthode de planification de trajectoire pour un robot mobile unicycle.

Nous avons commencé par présenter deux algorithmes inspirés de la nature, il s'agit de IWO et APGA, afin de d utiliser ce dernier pour étudier la planification de trajectoire en évitant les obstacles,

Ensuite nous avons traité la modélisation qui consiste à rassembler toutes les méthodes du robot afin de pouvoir établir un modèle dynamique en utilisant le formalisme de Lagrange. Puis, nous avons montré une architecture de contrôle basée sur la régulation PID.

Finalement, nous avons présenté la planification avec PID et la planification avec APGA et leurs résultats.

## Abstract:

A mobile robot is a vehicle with means of locomotion that allow it to move.

According to his degree of autonomy he may be endowed with means of perception and reasoning.

This dissertation discusses the use of plant intelligence to develop a trajectory planning method for a unicycle mobile robot.

We started by presenting two algorithms inspired by nature, IWO and APGA, in order to use the latter to study trajectory planning by avoiding obstacles,

We then dealt with the modelling which consists in gathering all the methods of the robot in order to establish a dynamic model using the Lagrange formalism. Then we showed a control architecture based on PID regulation.

Finally, we presented the planning with PID and the planning with APGA and their results.

## ملخص

الروبوت المتنقل هو مركبة ذات وسائل حركة تسمح له بالتحرك. وفقاً لدرجة استقلاليته، يمكن أن يُمنح وسائل الإدراك والمنطق. تتأقش هذه الأطروحة استخدام الذكاء النباتي لتطوير طريقة تخطيط المسار لروبوت متنقل أحادي الدراجة. بدأنا بتقديم خوارزميتين مستوحاة من الطبيعة IWO و APGA، من أجل استخدام الأخيرة لدراسة تخطيط المسار من خلال تجنب العقبات، ثم تعاملنا مع النمذجة التي تتمثل في جمع جميع طرق الروبوت من أجل إنشاء نموذج ديناميكي باستخدام شكلية لاغرانج. ثم أظهرنا بنية التحكم بناءً على تنظيم PID. أخيراً، قدمنا التخطيط مع PID والتخطيط مع APGA ونتائجها.