

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université de Béjaïa  
Faculté des Sciences Exactes  
Département d'Informatique

## *MEMOIRE DE MASTER RECHERCHE*

En  
Informatique

Option  
*Systemes d'Information Avancés*

## Thème

Prise de décision dans les véhicules autonomes

Présenté par : M. *MECHACHE Abderrahim*

Soutenu le 2 Juillet 2023 devant le jury composé de :

Président	M L. ZERARGA	MCB	Université de Béjaïa
Encadrant	M S. AISSANI	MCA	Université de Béjaïa
Co-Encadrante	Mme T. CHENACHE	Doctorante LMD	Université de Béjaïa
Examinatrice	Mme N. KHOULALENE	MCB	Université de Béjaïa
Examinatrice	Mme K. ZIZI	Doctorante LMD	Université de Béjaïa

Béjaïa, Juillet 2023.

*Au Nom de Dieu Le Plus Puissant ...*

## *Remerciements*

Je tiens à exprimer ma profonde gratitude à mes parents pour leur amour inconditionnel, leur soutien moral et financier tout au long de mes études. Leurs sacrifices et leur confiance en moi ont été un moteur essentiel dans ma réussite académique. Je suis reconnaissant et honoré de les avoir comme parents, et je leur dédie également ce travail. Merci infiniment pour tout ce que vous avez fait pour moi. Je tiens également à exprimer ma profonde gratitude à mes encadrants, AISSANI Sofiane et CHENACHE Tinhinane, pour leur soutien et leur guidance tout au long de ce projet. Leurs conseils avisés, leur patience et leur disponibilité ont été essentiels pour mener à bien cette recherche.

## *Dédicaces*

A mes chers parents, pour tous leurs sacrifices, leur amour,  
leur tendresse, leur soutien et leurs prières tout au long de mes études,  
Aucun mot, aucune dédicace ne pourrait exprimer mon respect, ma considération,  
et mon amour pour les sacrifices qu'ils ont consentis pour mon instruction  
et mon bien-être. Que ce travail soit l'accomplissement de vos vœux tant allégués,  
et le fruit de votre soutien infailible,  
Merci d'être toujours là pour moi.

*M. MECHACHE Abderrahim*

# *Table des matières*

<b>Table des figures</b>	<b>iii</b>
<b>Liste des tableaux</b>	<b>iv</b>
<b>Liste des abréviations</b>	<b>v</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Généralités</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Systèmes de transport intelligents . . . . .	3
1.2.1 Définition . . . . .	3
1.2.2 Evolution . . . . .	3
1.2.3 Avantages, inconvénients et défis à relever . . . . .	4
1.3 Véhicules autonomes . . . . .	5
1.3.1 Définition . . . . .	5
1.3.2 Fonctionnement du véhicule autonome . . . . .	6
1.4 Apprentissage automatique et Véhicules autonomes . . . . .	7
1.4.1 Apprentissage supervisé . . . . .	8
1.4.2 Apprentissage non-supervisé . . . . .	10
1.4.3 Apprentissage par renforcement . . . . .	10
1.5 Conclusion . . . . .	12
<b>2 État de l’art sur la prise de décision dans les véhicules autonomes</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Problématique . . . . .	13
2.3 Etude des travaux existants de prise de décision . . . . .	14
2.4 Etude comparative . . . . .	20
2.4.1 Classification des travaux passés en revus . . . . .	20
2.4.2 Tableau comparatif . . . . .	21
2.4.3 Discussion . . . . .	22
2.5 Conclusion . . . . .	23
<b>3 Proposition d’une approche de prise de décision basée sur l’intégration de l’apprentissage par renforcement avec KNN</b>	<b>24</b>
3.1 Introduction . . . . .	24
3.2 Motivation . . . . .	24

3.3	Notre Proposition . . . . .	25
3.3.1	Acteurs clés de notre approche . . . . .	25
3.3.2	Construction de l'agent . . . . .	28
3.4	Conclusion . . . . .	33
<b>4</b>	<b>Tests et Évaluation</b>	<b>34</b>
4.1	Introduction . . . . .	34
4.2	Environnement de travail . . . . .	34
4.3	Outils et bibliothèques utilisés . . . . .	34
4.3.1	Visual Studio . . . . .	34
4.3.2	Jupyter . . . . .	34
4.3.3	Python . . . . .	35
4.3.4	Numpy . . . . .	35
4.3.5	Pandas . . . . .	35
4.3.6	PyGame . . . . .	35
4.3.7	Scikit-learn . . . . .	35
4.3.8	Matplotlib . . . . .	35
4.4	Création de l'environnement de simulation . . . . .	36
4.5	Résultats . . . . .	36
4.6	Comparaison avec les approches de l'état de l'art . . . . .	40
4.7	Conclusion . . . . .	41
	<b>Conclusion générale et perspectives</b>	<b>42</b>
	<b>Bibliographie</b>	<b>43</b>

## *Table des figures*

1.1	Collecte de données par un véhicule autonome . . . . .	6
1.2	Analyse de données par le véhicule autonome . . . . .	7
1.3	Système de prise de décision dans les véhicules autonomes . . . . .	8
1.4	Interaction de l'agent avec l'environnement . . . . .	11
2.1	Classification des méthodes de prise de décision . . . . .	21
3.1	Distribution des capteurs de véhicule. . . . .	26
3.2	Fonctionnement des capteurs du véhicule. . . . .	26
3.3	Collecte des données . . . . .	29
3.4	données qui représentent les informations de l'état actuel, l'action, la récompense et la Q Value . . . . .	29
3.5	données qui représentent les informations de l'état suivant après l'exécution de l'action . . . . .	30
3.6	Codage de action . . . . .	31
3.7	Processus de prédiction des Q Values. . . . .	32
3.8	Précision de prédiction des Q-Values en fonction de la valeur de K . . . . .	32
3.9	Processus de prise de décision. . . . .	33
4.1	Environnement de simulation créé par PyGame. . . . .	36
4.2	Scénario de simulation. . . . .	38
4.3	Évolution de la vitesse moyenne du l'agent. . . . .	39
4.4	Analyse de la fréquence d'exécution des actions. . . . .	39

## *Liste des tableaux*

2.1	Comparaison des articles. . . . .	22
4.1	Fonction des récompenses. . . . .	37
4.2	Tableau des résultats. . . . .	37
4.3	Tableau comparatif. . . . .	40

## *Liste des abréviations*

<b>CNN</b>	Convolutional Neural Network.
<b>DDPG</b>	Deterministic Policy Gradient
<b>DPG</b>	Deterministic Policy Gradient
<b>DL</b>	Deep Learning.
<b>DMS</b>	Decision Making System.
<b>DBN</b>	Dynamic Bayesian network.
<b>DQN</b>	Deep Q Network.
<b>DRL</b>	Deep Reinforcement Learning.
<b>DT</b>	Decision Trees.
<b>ECU</b>	Electronic Control Unit
<b>GPS</b>	Global Positioning System
<b>IA</b>	Intelligence Artificiel.
<b>KNN</b>	K-Neigherst Neighbors.
<b>MDP</b>	Markov Decision Process.
<b>ML</b>	Machine Learning.
<b>OARL</b>	Observation Adversarial Reinforcement Learning
<b>PPO</b>	Proximal Policy Optimization
<b>Q – Value</b>	Quality Of Value.
<b>RNN</b>	Recurrent Neural Network
<b>RL</b>	Reinforcement Learning.
<b>SAC</b>	Soft Actor-Critic
<b>SAE</b>	Society of Automotive Engineers.
<b>SVM</b>	Support Vector Machine.
<b>STI</b>	Systèmes de transport intelligents.
<b>VA</b>	Véhicule Autonome.
<b>VS</b>	Visual studio.

# *Introduction générale*

De nos jours, le transport est devenu un enjeu majeur pour les villes et les pays du monde entier. Les embouteillages, la pollution et les accidents de la route sont des problèmes récurrents qui ont un impact négatif sur la qualité de vie des citoyens et sur l'environnement. Les véhicules autonomes ont le potentiel de transformer radicalement la manière dont nous voyons le transport. Ils peuvent offrir de nombreux avantages, tels que la réduction des accidents de la route, la diminution des émissions de gaz à effet de serre, l'optimisation des temps de trajet et la libération des conducteurs de la nécessité de se concentrer sur la conduite. En raison de ces avantages, les véhicules autonomes sont de plus en plus considérés comme un élément clé de l'avenir du transport. Ils peuvent être utilisés dans de nombreuses situations, y compris le transport de passagers et de marchandises. Le problème majeur dans les véhicules autonomes est la prise de décision, en effet comment le véhicule doit agir en fonction de son environnement, notre projet de fin de cycle s'inscrit dans cet axe de recherche.

Notre contribution principale consiste à proposer une méthode innovante pour la prise de décision des véhicules autonomes, ces véhicules seront équipés d'un système appelé "système de prise de décision", qui leur permet de choisir la meilleure action à exécuter. Les capteurs du véhicule autonome, fournissent des informations sur l'environnement dans lequel le véhicule se trouve, et le système de prise de décision les utilise pour choisir la meilleure action à prendre, en effet, le véhicule peut décider de ralentir, accélérer, maintenir sa vitesse, changer de voie ou s'arrêter. Une fois que la décision est prise, le système envoie des instructions à des contrôleurs de véhicules, qui s'occupent de contrôler la vitesse, la direction et d'autres aspects du véhicule pour que l'action soit exécutée en toute sécurité.

Nous avons développé une méthode basée sur l'utilisation de l'apprentissage par renforcement avec l'algorithme K le plus-proche voisin (KNN) pour aider les véhicules autonomes à prendre des décisions en toute sécurité. Cette méthode permet au véhicule de prendre des décisions appropriées en se basant sur des situations apprises, lui conférant ainsi une meilleure capacité d'adaptation et de meilleures performances. Ainsi d'intégrer des critères supplémentaires tels que le confort et le respect des règles de conduite, En dotant les véhicules autonomes de mécanismes leur permettant de répondre efficacement aux perturbations de la route, nous cherchons à améliorer leurs capacités de navigation et à garantir des déplacements sûrs et efficaces sans intervention humaine.

Nous avons validé notre approche en créant un environnement de simulation d'une autoroute à trois voies. Nous avons obtenu des résultats prometteurs qui ont démontré une nette amélioration des performances du modèle développé par rapport aux travaux existants. Cette



validation de notre approche pour la prise de décision des véhicules autonomes ouvre la voie à de nouvelles opportunités dans le domaine de la conduite autonome.

Ce travail soumis, accepté et présenté lors du Colloque international sur les Objets et Systèmes Connectés 2023 en Tunisie sous le titre "Enhancing Decision-Making in Autonomous Vehicles : A Study on the Integration of K-Nearest Neighbor Algorithm and Reinforcement Learning" [1]. Cela démontre l'excellence de notre travail et l'importance qu'il revêt pour la communauté scientifique.

Le rapport est structuré en quatre chapitres. Le premier chapitre présente les notions théoriques nécessaires à la compréhension de notre travail. Dans le deuxième chapitre nous effectuons une revue complète de l'état de l'art de la prise de décision dans les véhicules autonomes, en mettant en évidence les principales approches et les avancées récentes. Le troisième chapitre expose en détails notre approche proposée pour améliorer le processus de prise de décision dans les véhicules autonomes. Le quatrième et dernier chapitre est consacré à l'évaluation de notre approche à travers des simulations. Enfin, nous terminons ce mémoire par une conclusion générale, qui conclut le travail et ouvre des perspectives intéressantes qui méritent d'être explorées.

### 1.1 Introduction

Les systèmes de transport intelligents (STI) et les véhicules autonomes sont des sujets de plus en plus importants dans le monde de la technologie et des transports.

Dans ce premier chapitre, nous allons explorer les systèmes de transport intelligents et les véhicules autonomes. Nous allons commencer par définir ces concepts et examiner leur évolution au fil du temps. Nous allons également aborder les avantages, les inconvénients et les défis à relever pour ces technologies. Ensuite, nous allons nous concentrer sur les véhicules autonomes, en définissant leur fonctionnement et en explorant les différents types d'apprentissage automatique utilisé pour leur développement.

### 1.2 Systèmes de transport intelligents

#### 1.2.1 Définition

Le système de transport intelligent (STI) est un système de transport émergent qui existe depuis le milieu des années 60, mais qui a continué à évoluer au fil du temps. Initialement, il prenait la forme de systèmes d'assistance à la conduite et de systèmes de conduite automatisée supervisés par le conducteur (semi-autonomes). Cependant, aujourd'hui, il utilise les technologies avancées que le développement et la recherche nous ont apportées, telles que les capteurs avancés, le traitement de l'information, les systèmes de communication et de contrôle automatique, afin de percevoir et de communiquer des informations importantes et à jour concernant notre environnement que nous, en tant qu'humains, ne pouvons pas voir ou obtenir lorsque nous sommes sur la route. L'objectif est de nous offrir un système de gestion des transports en temps réel, précis et efficace, qui serve de précaution de sécurité pour naviguer et éviter toute collision ou tout trafic susceptible de nous retarder à tout moment [2].

#### 1.2.2 Evolution

Les STI ont connu de nombreuses transformations avant d'atteindre le niveau actuel. Au fil des différentes versions et de leur évolution, ils ont franchi plusieurs niveaux d'autonomie et subi des changements significatifs. Tout a commencé avec une communication à sens unique, où le conducteur recevait une assistance en matière de sécurité de la part de la voiture. Ensuite, la voiture a progressivement acquis la capacité d'effectuer certaines fonctionnalités de manière

indépendante, telles que le système d'assistance au maintien de voie, le freinage d'urgence autonome, Le stationnement automatique, etc. avant de devenir capable de conduire complètement par elle-même. Cela nécessitait une communication bidirectionnelle, rendue possible grâce à des opérations et une gestion interactives du système, ainsi qu'à des réseaux d'information et des dispositifs spécifiques.

Au fur et à mesure de ces évolutions, les STI ont pu offrir des fonctionnalités de plus en plus avancées et autonomes. Ils ont progressivement gagné en intelligence et en capacité à prendre des décisions en fonction de leur environnement. Grâce à la communication bidirectionnelle, les véhicules autonomes peuvent échanger des informations avec leur environnement, tels que d'autres véhicules, les infrastructures routières ou les systèmes de gestion du trafic. Ces échanges d'informations sont essentiels pour assurer une conduite autonome sûre et efficace.

Les STI ont également bénéficié des progrès dans les technologies de l'information et des réseaux. Les dispositifs embarqués dans les véhicules ont été améliorés pour collecter et traiter les données provenant des capteurs et des systèmes de navigation. Les réseaux de communication ont été renforcés pour permettre une transmission rapide et fiable des informations entre les véhicules et les infrastructures.

En résumé, les STI ont connu une évolution remarquable, passant d'une simple assistance à la sécurité du conducteur à des véhicules autonomes capables de conduire de manière indépendante. Tout cela a été rendu possible grâce à l'amélioration des opérations interactives du système, à la gestion des informations et aux réseaux de communication. Ces avancées ont ouvert la voie à une nouvelle ère de mobilité intelligente et autonome [2].

### 1.2.3 Avantages, inconvénients et défis à relever

Les systèmes de transport intelligents ont le potentiel d'améliorer l'efficacité des réseaux de transport [3], Les STI offrent de nombreuses opportunités pour améliorer notre vie quotidienne, les avantages des STIs sont nombreux comme :

#### - Amélioration de la sécurité routière :

Les STI permettent la mise en place de systèmes de surveillance et de détection avancés, tels que la détection des collisions imminentes, la détection de la somnolence du conducteur et l'alerte en cas de dépassement de la limite de vitesse. Ces fonctionnalités contribuent à réduire les accidents et à sauver des vies.

#### - Amélioration de l'efficacité des transports en commun :

Les STI facilitent la coordination et la gestion des systèmes de transport en commun, permettant une meilleure planification des horaires, une réduction des temps d'attente et une amélioration de l'expérience des usagers. Cela encourage l'utilisation des transports en commun, réduisant ainsi le nombre de véhicules individuels sur les routes.

#### - Optimisation de la circulation :

Les STI utilisent des systèmes de gestion du trafic intelligents pour optimiser la fluidité de la circulation, réduire les congestions et minimiser les temps de trajet. Grâce à la collecte de données en temps réel sur le trafic, les STI peuvent proposer des itinéraires alternatifs et des informations en direct aux conducteurs, favorisant ainsi une circulation plus fluide.

Malgré les nombreux avantages des STI, il est important de prendre en compte les défis et les risques qu'ils présentent. Les STI peuvent avoir des impacts négatifs [4], voici quelques exemples sur les difficultés des STI :

### - Coût élevé de déploiement :

La mise en place de systèmes de transport intelligents nécessite des investissements importants en termes d'infrastructures, d'équipements et de technologies. Ces coûts peuvent être un obstacle majeur à la mise en œuvre généralisée des STI, en particulier dans les régions à faibles ressources

### - Protection de la vie privée et sécurité des données :

Les STI collectent et traitent de grandes quantités de données, notamment des informations personnelles sur les usagers de la route. Cela soulève des préoccupations légitimes en matière de protection de la vie privée et de sécurité des données, notamment en ce qui concerne le stockage et la transmission de ces données sensibles.

Relever les défis posés par les STI est essentiel pour maximiser leurs avantages tout en minimisant leurs inconvénients [4], parmi les défis majeurs :

### - Interopérabilité des systèmes :

Les STI nécessitent l'interopérabilité entre différents systèmes et technologies pour permettre une communication et une coordination efficaces. Cela peut être difficile à réaliser en raison des différences de protocoles, de normes et de plateformes entre les différents acteurs impliqués.

### - Acceptation des STI :

Les STI impliquent souvent des changements importants dans les habitudes de déplacement et les comportements des usagers de la route. Il peut être difficile de susciter l'acceptation et l'adoption des STI par les utilisateurs, qui peuvent avoir des réticences liées à la confiance, à la familiarité avec les nouvelles technologies ou à la perception des avantages réels.

## 1.3 Véhicules autonomes

### 1.3.1 Définition

Un véhicule autonome est un véhicule qui peut se déplacer sur la route en toute sécurité sans avoir besoin d'être dirigée par un conducteur. Cela signifie que la voiture est capable de détecter les obstacles et de naviguer sur la route sans l'aide de l'homme [5].

Il y a plusieurs niveaux d'autonomie pour les véhicules, La classification établie par la SAE (Society of Automotive Engineers) [6] divise l'autonomie en cinq niveaux au total.

- **Niveau 0 - Conduite manuelle :** Toutes les tâches sont réalisées par l'Homme.
- **Niveau 1 - Conduite assistée :** Le conducteur reste en charge du véhicule, mais il peut bénéficier de certaines fonctions d'assistance, comme la régulation de vitesse ou l'aide au stationnement.

- **Niveau 2 - Conduite partiellement automatisée** : Le véhicule peut gérer certaines tâches de conduite, comme le maintien de sa trajectoire ou le freinage automatique. Cependant, le conducteur doit toujours être prêt à reprendre le contrôle à tout moment.
- **Niveau 3 - Conduite conditionnellement automatisée** : Le véhicule peut prendre en charge la conduite dans certaines situations, mais le conducteur doit être en mesure de reprendre le contrôle en cas de besoin. À ce niveau, le conducteur peut être autorisé à ne pas surveiller en permanence la route.
- **Niveau 4 - Conduite hautement automatisée** : Le véhicule peut gérer la conduite dans la plupart des situations, mais il peut y avoir des exceptions, comme les conditions météorologiques extrêmes ou les travaux routiers. À ce niveau, le conducteur n'a pas besoin de surveiller la route en permanence.
- **Niveau 5 - Conduite totalement automatisée** : Le véhicule est capable de se conduire seul dans toutes les situations, sans intervention humaine requise. Les passagers peuvent simplement se détendre et laisser le véhicule faire tout le travail.

### 1.3.2 Fonctionnement du véhicule autonome

Les voitures autonomes sont équipées de technologies avancées telles que des capteurs, des logiciels et une intelligence artificielle pour leur permettre de rouler sans intervention humaine. Leur objectif est de fournir une sécurité renforcée et une expérience de conduite plus confortable et pratique pour les passagers. Le déroulement de fonctionnement de ces véhicules suit trois étapes essentielles [7], la collecte des données, l'analyse de données et la prise de décision.

#### 1.3.2.1 La collecte de données

Pour qu'une voiture puisse rouler toute seule, il faut d'abord l'équiper avec plein de détecteurs (caméras, radars, lidars), Tous ces détecteurs sont un peu comme les yeux de la voiture. Ils capturent tout le temps toutes les informations importantes pour conduire, comme la circulation ou les panneaux de signalisation, pour assurer que la voiture peut voir d'une manière efficace son environnement pour conduire en toute sécurité. la figure suivante 1.1 présente les différents capteurs de VA :

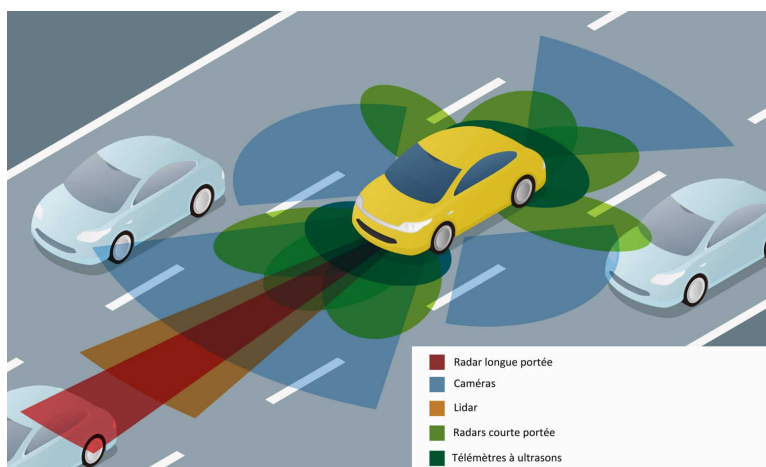


FIGURE 1.1 – Collecte de données par un véhicule autonome [8]

### 1.3.2.2 L'analyse de données

Toutes les informations collectées par les capteurs sont transmises au cerveau de la voiture, qui est un logiciel informatique. Ce logiciel est responsable de l'analyse, de la corrélation et de l'interprétation des données en temps réel, et repose sur une intelligence artificielle. Avant d'analyser l'environnement extérieur, le logiciel a été préalablement entraîné pour reconnaître les différents objets de la route (signaux de la route, piétons, feux de circulation). Il a également appris à mémoriser une variété de scénarios de conduite, tels que l'arrêt soudain d'une voiture. Le cerveau de la voiture est également redondant pour garantir une sécurité maximale, la figure suivante 1.2 montre comment le logiciel informatique analyse les données capturés par le VA :

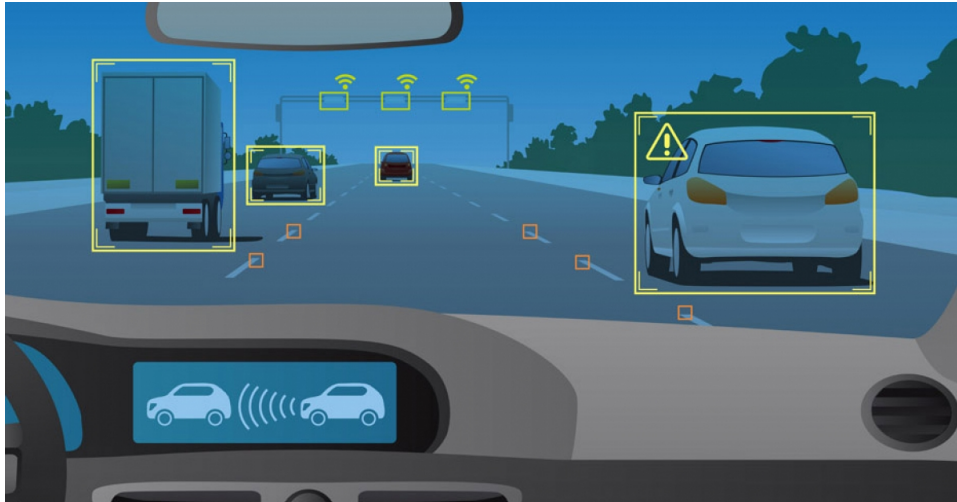


FIGURE 1.2 – Analyse de données par le véhicule autonome [9]

### 1.3.2.3 Prise de décision

Les étapes précédentes permettent de fournir des données exploitables par le véhicule intelligent afin de prendre des décisions.

Une fois la collecte et l'analyse de données terminées, le véhicule doit disposer d'un mécanisme, appelé système de prise de décision DMS lui permettant de prendre une décision, autrement dit d'exécuter une action, cette dernière devrait permettre à l'agent de rouler en toute sécurité dans le trafic réel.

Comme il est illustré dans la figure 1.3, l'action prise aboutit ensuite à une action qui sera exécutée par les contrôleurs du véhicule, ces derniers active ou désactive certaines commandes de deux niveaux de comportements, de **High-Level**, tels que la fusion, le dépassement, le maintien et le changement de voie. et de **Low-Level**, comprenant principalement la longueur, la vitesse, l'accélération et la vitesse angulaire [10].

## 1.4 Apprentissage automatique et Véhicules autonomes

Les voitures autonomes, une technologie essentiellement "intelligente", ne naissent pas intelligentes. Les algorithmes qui contrôlent leurs mouvements apprennent au fur et à mesure que la technologie évolue. Les voitures autonomes représentent un test crucial pour les capacités de l'apprentissage automatique [11].

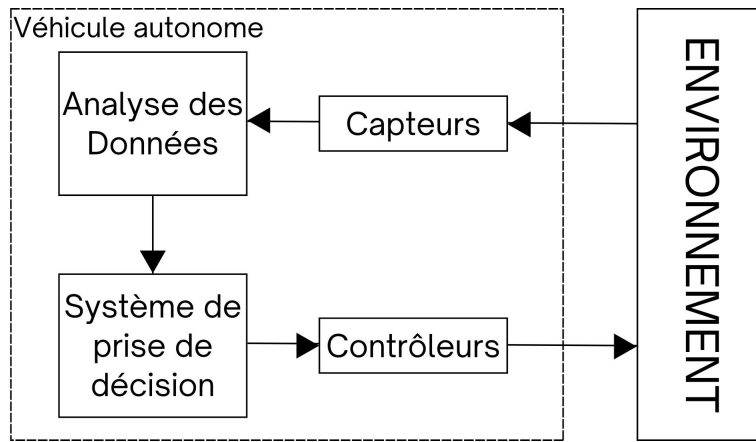


FIGURE 1.3 – Système de prise de décision dans les véhicules autonomes

L'apprentissage automatique (en anglais **Machine Learning**) est l'étude scientifique des algorithmes et des modèles statistiques que les systèmes informatiques utilisent pour effectuer une tâche spécifique sans être explicitement programmés. Les algorithmes d'apprentissage sont utilisés dans de nombreuses applications que nous utilisons au quotidien [12].

Nous présentons les différents types de ML et leurs utilisations dans les Véhicules autonomes :

### 1.4.1 Apprentissage supervisé

En anglais "**Supervised learning**", est une tâche d'apprentissage automatique dans laquelle un ordinateur doit apprendre à associer des entrées à des sorties en se basant sur des exemples de relations d'entrées - sorties. Pour y arriver, il utilise des données d'entraînement qui ont été préalablement étiquetées. Les algorithmes de "supervised learning" ont besoin d'une aide extérieure pour fonctionner. Pour cela, le jeu de données d'entrée est divisé en deux parties : une partie d'entraînement et une partie de test. Le programme cherche donc à trouver une relation (généralement des équations mathématiques) qui relie toute entrée dans la partie d'entraînement avec la sortie associée, une fois cette relation est élaborée il l'applique pour prédire les sorties associées à chaque entrée de la partie de tests [12].

Voici des exemples de modèles utilisés dans ce type

#### 1.4.1.1 Réseaux de neurones :

Un réseau de neurones est une série d'algorithmes qui cherche à reconnaître les relations sous-jacentes dans un ensemble de données en imitant le fonctionnement du cerveau humain. L'apprentissage avec ces algorithmes appelé Apprentissage profond (en anglais **Deep learning DL**), En d'autres termes, les réseaux de neurones font référence à des systèmes de neurones, Ces réseaux peuvent s'adapter aux changements dans les entrées, ce qui leur permet de générer le meilleur résultat possible sans avoir besoin de reconfigurer les critères de sortie [12].

les réseaux de neurones sont utilisés de diverses manières pour améliorer les performances et les capacités des véhicules autonomes :

#### Détection d'objets :

Les Réseaux de neurones Convolutifs (CNN) sont utilisés pour détecter et classifier les objets dans l'environnement du véhicule [13], tels que les piétons, les véhicules, les panneaux

de signalisation, etc.

### **Prédiction de trajectoires :**

Les Réseaux neuronaux Récurrents (RNN) sont employés pour prédire les trajectoires des autres véhicules ou des piétons, permettant au véhicule autonome d'anticiper les mouvements futurs et de prendre des décisions en conséquence [14].

#### **1.4.1.2 K le plus proche voisin (KNN) :**

Le k-plus proches voisins (KNN) est un algorithme de ML, utilisé pour la classification et la régression.

Le fonctionnement du KNN consiste à examiner les k exemples d'apprentissage les plus proches en termes de distance, et attribue la classe majoritaire parmi ces voisins à l'exemple en question s'il s'agit de la classification, ou la moyenne de l'information des voisins dans la régression. La distance peut être calculée de différentes manières [15].

Dans le contexte des véhicules autonomes, le KNN peut être utilisé de différentes manières, notamment :

### **Prédiction de l'action à choisir :**

En analysant les exemples d'apprentissage qui sont similaires à la situation actuelle, le KNN peut aider à prédire l'action appropriée à choisir pour le véhicule autonome. Par exemple, en examinant les cas passés similaires où un véhicule autonome devait décider entre freiner ou accélérer, le KNN peut estimer quelle action serait la plus sûre dans la situation actuelle [16].

### **Reconnaissance de comportements dangereux :**

En comparant les actions des autres usagers de la route avec des exemples d'apprentissage, le KNN peut aider à détecter les comportements dangereux tels que les changements de voie agressifs ou les dépassements imprudents. Ces informations peuvent être utilisées pour adapter le comportement du véhicule autonome et garantir une conduite sûre [17].

#### **1.4.1.3 Support Vector Machines (SVM) :**

(SVM) est un modèle d'apprentissage automatique supervisé avec des algorithmes d'apprentissage associés qui analysent les données utilisées pour la classification et l'analyse de régression. En plus de réaliser une classification linéaire, les SVM peuvent efficacement effectuer une classification non linéaire en utilisant ce qu'on appelle la méthode du noyau, qui mappe implicitement leurs entrées dans des espaces de caractéristiques de grande dimension [12].

Les SVM sont largement utilisés dans les véhicules autonomes pour diverses tâches, notamment :

### **Classification des scènes :**

Les SVM sont utilisés pour classifier les différentes scènes de conduite, par exemple, pour reconnaître les intersections, les routes à sens unique, les ronds-points, etc. [17].



## 1.4.2 Apprentissage non-supervisé

L'apprentissage non supervisé étudie comment les systèmes peuvent apprendre à représenter des schémas d'entrée spécifiques de manière à refléter la structure statistique de l'ensemble des schémas d'entrée. Contrairement à l'apprentissage supervisé ou à l'apprentissage par renforcement, il n'y a pas de sorties cibles explicites ou d'évaluations environnementales associées à chaque entrée. Au lieu de cela, l'apprenant non supervisé utilise des biais préalables pour déterminer quels aspects de la structure de l'entrée doivent être capturés dans la sortie. En d'autres termes, dans ce type d'apprentissage les algorithmes cherchent à identifier des relations entre les entrées pour établir des sous-ensembles des exemples les plus similaires entre eux [18].

### 1.4.2.1 K-means :

K-means est l'un des algorithmes d'apprentissage non supervisé les plus simples qui résolvent le problème bien connu du regroupement. La procédure suit une manière simple et facile de classer un ensemble de données donné en un certain nombre de groupes. L'idée principale est de définir  $k$  centres, un pour chaque groupe. Ces centres doivent être placés de manière astucieuse car différents emplacements entraînent des résultats différents. Donc, le meilleur choix est de les placer aussi loin que possible les uns des autres [11].

Utilisations de l'algorithme K-means dans les véhicules autonomes :

#### Détection de groupes de véhicules :

En analysant les caractéristiques des véhicules détectés, K-means peut être utilisé pour identifier des groupes ou des clusters de véhicules qui se déplacent de manière similaire ou qui partagent des schémas de comportement communs [19].

#### Détection de comportements anormaux :

K-means peut être utilisé aussi pour détecter des comportements anormaux ou des situations dangereuses en comparant le comportement actuel d'un véhicule à des groupes de comportements normaux préalablement appris [19].

## 1.4.3 Apprentissage par renforcement

L'apprentissage par renforcement (en anglais **Reinforcement Learning - RL**) est un algorithme dont l'objectif est d'apprendre des stratégies visant à maximiser les bénéfices en essayant différentes actions. Les actions des agents peuvent être ajustées en fonction des fonctions de récompense, Les données existantes ainsi que les nouvelles données obtenues par l'exploration de l'environnement [10].

L'apprentissage par renforcement est actuellement l'une des méthodes d'apprentissage les plus utilisées pour la prise de décision dans les véhicules autonomes, une approche où les véhicules autonomes apprennent à partir de l'interaction avec l'environnement en utilisant un système de récompense et de punition. Les actions prises par le véhicule sont récompensées ou sanctionnées en fonction de leur impact sur l'objectif souhaité (par exemple, atteindre une destination en toute sécurité va générer des récompenses, faire un accident implique des punitions, etc.). Le véhicule apprend alors à maximiser la récompense cumulée au fil du temps.

Le RL dépend généralement de la décision de Markov Processus (en anglais **Markov decision process**, MDP) pour décrire les états d'interaction de l'agent et environnement. Les

MDP sont utilisés dans de nombreux domaines tels que la robotique, l'économie et le contrôle automatique. il sont utilisés afin de modéliser les résultats de l'exécution des actions sur les états [20].

Dans le RL le MDP est utilisé comme un cadre mathématique qui modélise de manière probabiliste l'interaction entre un agent et l'environnement. L'agent est considéré comme un apprenant ou un décideur qui interagit avec l'environnement dont il dispose d'un modèle qui lui permette de choisir les actions selon une stratégie, et une mémoire qui sauvgarde les expériences de l'agent. À chaque étape de temps, il reçoit une récompense ainsi qu'une représentation de l'état de l'environnement, et exerce une action sur l'environnement qui peut modifier son futur état [21].

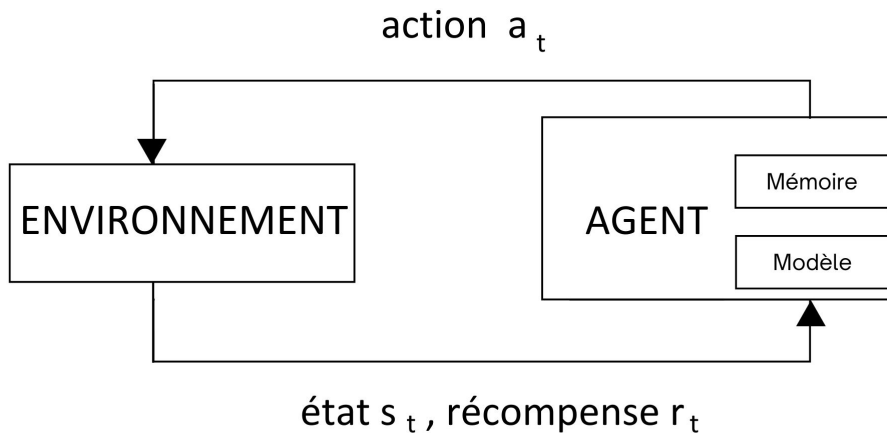


FIGURE 1.4 – Interaction de l'agent avec l'environnement

Voici des exemples de modèles utilisés de type d'apprentissage par renforcement dans les véhicules autonomes.

#### 1.4.3.1 Q-Learning :

Q-Learning est une méthode d'apprentissage par renforcement qui a atteint des performances remarquables dans des domaines tels que les jeux et la robotique. La lettre 'Q' désigne la fonction qui mesure la qualité d'une action exécutée dans un état donné du système définie comme la somme attendue des récompenses futures lorsque l'on entreprend cette action et que l'on suit la politique optimale par la suite [22]. Dans le cadre d'une politique donnée  $\pi$  la valeur réelle d'une action  $a$  dans un état  $s$  est calculé par l'équation 1.1 , où  $R_1, R_2, \dots$  sont les récompenses obtenues à chaque étape du processus d'apprentissage par renforcement. Le facteur d'actualisation  $\gamma \in [0, 1]$  permet d'arbitrer entre l'importance des récompenses immédiates et celle des récompenses ultérieures. La valeur optimale pour une action  $a$  dans un état  $s$  à l'instant  $t$  présenté dans 1.2.

$$Q(s, a) = E[R_1 + \gamma R_2 + \dots | S_0 = s, A_0 = a, \pi] \quad (1.1)$$

$$Q(s_t, a_t) = r + \gamma \max_{a'} Q_\pi(s_{t+1}, a_{t+1}) \quad (1.2)$$

l'objectif de cette algorithmes est d'explorer l'ensemble des actions possibles dans l'environnement (les états) afin de trouver la politique qui va permettre à l'agent de gagner plus de récompenses.

### 1.4.3.2 Deep Q-Networks (DQN) :

le DQN est une méthode d'apprentissage par renforcement qui utilise un réseau neuronal profond pour approximer la fonction de valeur Q. Il apprend à partir d'expériences passées stockées dans sa mémoire appelé "buffer de relecture", et utilise un algorithme d'optimisation pour ajuster les poids du réseau et améliorer les prédictions des valeurs Q [22].

Le DQN utilise un réseau de neurones profond (souvent basé sur des architectures de réseaux de neurones convolutionnels (CNN)), pour apprendre à prédire les valeurs Q. Le réseau prend l'état actuel en entrée et génère les valeurs Q pour toutes les actions possibles dans cet état en sortie. Il est entraîné en utilisant un algorithme d'optimisation, tel que la descente de gradient, pour minimiser la différence entre les valeurs Q prédites et les valeurs Q réelles, qui sont obtenues à partir d'expériences de jeu.

Les méthodes présentés au-dessus sont devenues des méthodes populaire pour améliorer la prise de décision des véhicules autonomes [22]. Grâce à le RL, les véhicules autonomes peuvent apprendre à prendre des décisions intelligentes et adaptatives dans des environnements complexes et dynamiques. Par exemple, en utilisant des algorithmes de RL avancés, un véhicule autonome peut apprendre à choisir la meilleure action à prendre en fonction de son état actuel et des informations sensorielles provenant de son environnement. Cela permet aux véhicules autonomes de naviguer de manière autonome, d'interagir avec les autres usagers de la route et de prendre des décisions éclairées pour optimiser la sécurité et l'efficacité. le RL permet également aux véhicules autonomes de s'adapter à des situations inattendues et d'apprendre de nouvelles stratégies en continu, ce qui est essentiel pour faire face aux défis de la conduite réelle.

En utilisant le RL, les véhicules autonomes peuvent évoluer et s'améliorer au fil du temps, rendant les déplacements plus sûrs et plus agréables pour les passagers et les autres usagers de la route

## 1.5 Conclusion

Les transports intelligents et les véhicules autonomes vont révolutionner nos modes de déplacement dans un futur très proche, les STI et les véhicules autonomes sont des technologies prometteuses qui pourraient révolutionner le monde des transports. Les STI offrent de nombreux avantages, notamment une meilleure gestion du trafic et une réduction de la pollution, tandis que les véhicules autonomes pourraient améliorer la sécurité routière et offrir une plus grande flexibilité dans les déplacements.

L'apprentissage automatique est la clé qui permettra aux véhicules autonomes de devenir une réalité. Grâce aux progrès fulgurants dans ce domaine, les véhicules autonomes deviendront bientôt plus sûrs que les conducteurs humains et transformeront nos habitudes de déplacement. Le prochain chapitre se concentrera sur les méthodes utilisées dans la prise de décision des véhicules autonomes. Nous explorerons les différents algorithmes et approches qui permettent aux véhicules autonomes de prendre des décisions sûrs et efficaces en temps réel.

## Chapitre II

# *État de l'art sur la prise de décision dans les véhicules autonomes*

## 2.1 Introduction

Le domaine de la conduite autonome a connu un développement rapide ces dernières années et la prise de décision dans les systèmes de conduite autonomes est un sujet crucial pour assurer la sécurité et l'efficacité des véhicules autonomes. Dans ce chapitre, nous présentons un état de l'art sur les méthodes existantes pour la prise de décision dans les véhicules autonomes. Dans ce chapitre, nous allons explorer les différentes méthodes existantes pour la prise de décision dans les véhicules autonomes en analysant leurs différences et une comparaison en fonction des critères établis.

## 2.2 Problématique

La prise de décision dans les véhicules autonomes est devenue un sujet d'actualité majeur dans ce domaine, en raison des avancées technologiques et de l'émergence des techniques d'intelligence artificielle (IA). Ces avancées permettent aux véhicules autonomes de disposer de mécanismes puissants pour analyser et traiter les multiples informations provenant de leur environnement, ouvrant ainsi la voie à une conduite plus sécurisée et efficace.

Cependant, les véhicules autonomes doivent également être dotés d'un mécanisme leur permettant de prendre des décisions efficaces en fonction des informations traitées. En d'autres termes, ils doivent pouvoir choisir des actions telles que l'accélération, la rotation, etc., qui leur permettent de circuler en toute sécurité sans intervention humaine, garantissant ainsi une conduite autonome fiable et sécurisée.

Dans cette section, nous examinons plusieurs travaux qui présentent des approches pour la prise de décision dans les véhicules autonomes. Ces recherches visent à développer des mécanismes et des méthodes permettant au véhicule de prendre des décisions efficaces en exploitant les informations disponibles. En d'autres termes, elles cherchent à proposer des méthodes qui aident les véhicules autonomes à sélectionner les actions les plus appropriées en fonction des caractéristiques de leur environnement, favorisant ainsi une conduite plus sûre et plus performante.

## 2.3 Etude des travaux existants de prise de décision

### 1. Highway Traffic Modeling and Decision Making for Autonomous Vehicle Using Reinforcement Learning.

Dans cet article [23] les auteurs ont commencé par la présentation de la modélisation du problème en utilisant un processus de décision markovien stochastique MDP pour modéliser l'interaction entre le véhicule autonome et son environnement, Ces modèles sont utilisés souvent dans le domaine de Robotique, ils permettent de modéliser l'interaction entre l'agent intelligent (et ses actions (les décisions prises)) avec son environnement dans le processus de fonctionnement dont Il reçoit une représentation de l'état de l'environnement et une récompense à chaque pas de temps.

L'environnement du véhicule est représenté par une cellule qui regroupe les voies du trafic. Cette cellule elle-même est composée des petites cellules qui indiquent les emplacements possibles pour les véhicules par rapport au véhicule intelligent qui se trouve dans la cellule centrale.

La problématique étudiée dans cet article est de trouver une stratégie optimale permettant au véhicule d'apprendre à conduire en utilisant une fonction de récompense, les auteurs ont donc utilisé l'apprentissage par renforcement pour optimiser une fonction de récompense qui représente les récompenses obtenues en fonction de l'espace d'état-action :

$$R(s, a) = w^T \phi(s, a)$$

Où  $w$  est le vecteur de poids, et  $\Phi(s, a)$  est le vecteur de caractéristiques avec chaque composante représentant un point de caractéristique unique dans l'espace état-action  $(s, a)$ . Les caractéristiques de  $\Phi(s, a)$  sont :

- Les caractéristiques de l'action (Certains actions sont préférées).
- La position du véhicule autonome par rapport aux limites de la route.
- La stratégie de dépassement.
- Le fait de suivre de près ou non un autre véhicule.
- L'incident de collision.

Les auteurs ont clôturé leur travail avec des tests en modifiant les valeurs des points représentés ci-dessus pour essayer différents stratégie, leurs résultats ont montrés qu'en fonction de l'avancement de résultats l'erreur se minimise.

### 2. Robust Lane Change Decision Making for Autonomous Vehicles : An Observation Adversarial Reinforcement Learning Approach.

Dans cet article [24] les auteurs ont étudié la problématique de changement de la voie dans les véhicules autonomes, ils ont commencé par présenter les méthodes les plus utilisées dans les algorithmes de l'apprentissage par renforcement comme AlphaZeroGo, qui est Une combinaison de l'apprentissage profond et de la recherche arborescente, et Actor critique, où deux réseaux de neurones sont utilisés, un acteur et un critique. dont le réseaux acteur choisit l'action à exécuter, le deuxième réseaux critique cette action (l'accepter ou non).

Selon les auteurs ces méthodes supposent que les observations de l'état du véhicule sont exemptes de perturbations inattendues, ce qui peut ne pas être le cas dans des scénarios du monde réel, et ils ont donc opté pour proposer une approche permettant de gérer les perturbations qui se base sur le fonctionnement suivant :

1. Le trafic est représenté par SUMO génère un état actuel  $s$  et une récompense  $r$  (selon l'état où se trouve le véhicule autonome qui est présenté par la ligne sur laquelle le véhicule se trouve ainsi que la vitesse de chaque véhicule et la distance qui sépare chaque véhicule avec le véhicule autonome) et l'action exécuté.
2. Le « Black-box Attack with Bayesian Optimization » va prédire les perturbations adversaires possibles.
3. Les données générées dans 1 et 2 sont transmises au « Constrained Observation-robust Markov Decision Process » (CORMDP) et « Constrained Observation-robust Actor-critic » (COR-AC).
4. Le COR-AC optimise les perturbations et les transmette au CORMDP qui va choisir une action à exécuter.
5. L'action sera appliquée dans l'environnement simulé par SUMO pour avoir les nouvelles données (état, récompense) dans l'instant  $t+1$  pour faire une autre itération.

En fin les auteurs ont testé leurs approches en la comparant avec d'autres méthodes d'apprentissage comme DQN, PPO, SAC et OARL dans des situations de trafic mixte et aléatoire et les résultats montrent que ce nouveau schéma permet d'avoir une meilleure généralisation et une plus grande résistance aux perturbations.

### **3. Adaptive Game-Theoretic Decision Making for Autonomous Vehicle Control at Roundabouts.**

Dans le cadre de cette étude [25], les auteurs ont exploré le contrôle autonome des véhicules dans des ronds-points en utilisant une approche basée sur la combinaison de la théorie des jeux avec l'apprentissage par renforcement.

À travers leur travail, les auteurs ont cherché à proposer un cadre de prise de décision basé sur la théorie des jeux, où les véhicules sont des joueurs qui doivent prendre des décisions, La théorie des jeux a été utilisée pour déterminer les meilleures actions à prendre en fonction des actions des autres véhicules dans le jeu et pour déterminer la meilleure stratégie pour chaque véhicule à chaque étape du jeu ils ont utilisé la solution Nash de jeu, Ensuite, ils ont combiné cette stratégie avec la politique apprise par l'apprentissage par renforcement pour décider de la meilleure action à prendre dans chaque état en utilisant l'algorithme de Q-learning pour apprendre les politiques de décision optimales pour chaque joueur.

Les auteurs ont effectué des tests sur une simulation de circulation routière à un rond-point dans différentes conditions de trafic pour évaluer les performances de leur algorithme de prise de décision en temps réel. Ils ont comparé leur approche adaptative de la théorie des jeux à une approche de contrôle de véhicule classique basée sur des règles, Les résultats ont montré que l'approche adaptative de la théorie de jeux a réussi à atteindre des temps de trajet plus courts pour les véhicules autonomes que l'approche basée sur des règles. De plus, l'approche de la théorie des jeux a été en mesure de maintenir une distance de sécurité plus importante entre les véhicules et de minimiser le nombre de changements de voie, ce qui contribue à réduire le risque d'accidents.

### **4. A New Approach for Tactical Decision Making in Lane Changing : Sample Efficient Deep Q Learning with a Safety Feedback Reward.**

Dans cet article [26], les auteurs ont proposé d'utiliser la méthode Rainbow DQN pour prendre des décisions tactiques dans des scénarios routiers en tenant compte des contraintes

de sécurité. Ils ont implémenté cet agent pour résoudre le problème de changement de voie.

Rainbow est une méthode de renforcement améliorée de la méthode DQN qui priorise les expériences les plus importantes pour la relecture et met à jour les valeurs Q en conséquence, en utilisant une approche RL de distribution, qui représente la distribution des rendements possibles pour chaque action plutôt qu'une seule valeur.

les auteurs ont proposé ainsi d'utiliser une couche de sécurité qui fournit une rétroaction de récompense à l'agent, Cette couche a été ajoutée à leur réseau de neurones qui prend en compte la distance avec les véhicules adjacents. Si l'agent prend une décision qui met en danger la sécurité, la couche de sécurité retourne une récompense négative au réseau de neurones, qui met à jour ses paramètres pour prendre en compte cette information lors de la prise de décision à l'avenir.

Leur algorithme a atteint la même performance des approches existantes avec moins d'expérience. Ce qui a montré que leur méthode a surpassé ces approches en termes de performance et d'efficacité d'échantillonnage.

### **5. A Decision-Making Strategy for Vehicle Autonomous Braking in Emergency via Deep Reinforcement Learning.**

Cet article [27] propose une approche pour la prise de décision autonome en situation d'urgence dans les scénarios de changement de voie forcé et le freinage, ils ont proposé une approche basée sur l'apprentissage par renforcement en utilisant l'architecture acteur critique qui utilise deux réseaux de neurones : l'acteur qui prend des décisions et le critique qui évalue les actions prises par l'acteur. L'objectif est d'optimiser l'acteur pour maximiser la récompense reçue et le critique pour minimiser l'erreur d'évaluation le véhicule devrait être capable de détecter à l'avance les dangers en prenant en considération le confort des passagers.

Ils ont proposé donc L'algorithme DDPG (Deep Deterministic Policy Gradient) qu'est utilisé pour la prise de décision et le contrôle de freinage autonome des véhicules. Il utilise un réseau de neurones pour apprendre la politique de contrôle de freinage optimale en utilisant des informations sur l'état du véhicule et en explorant l'environnement par des actions de contrôle de freinage générées par le réseau de neurones de l'acteur. L'algorithme utilise une mémoire pour stocker les transitions d'état, d'action et de récompense, qui sont utilisées pour entraîner le réseau de neurones du critique pour prédire la valeur de la récompense à partir de l'état et de l'action. Le réseau de neurones de l'acteur est mis à jour en utilisant le politique gradient de l'acteur pour maximiser la récompense attendue. L'algorithme utilise également des réseaux de neurones cibles pour stabiliser l'apprentissage.

Enfin les auteurs ont mis à l'épreuve leur approche en la comparant avec une approche existante appelée DPG (un algorithme d'apprentissage de politiques déterministes qui a été introduit en 2014), Les auteurs ont observé que les deux approches ont généré des résultats similaires, mais leur approche a été la première à atteindre ces résultats. Cela démontre que leurs agents apprennent plus rapidement, ce qui leur permet de gagner du temps dans la phase d'apprentissage.

### **6. Safe Decision Controller for Autonomous Driving Based on Deep Reinforcement Learning in Nondeterministic Environment.**

Dans cet article [28] les auteurs ont abordé les inconvénients des systèmes d'apprentissage en profondeur par renforcement, selon eux ces systèmes sont basés sur des boîtes noires ne garantissent pas la sécurité du système et l'interopérabilité des paramètres de fonction de

récompense face à des environnements complexes et à l'influence d'incertitudes incontrôlées. Pour cela ils ont proposé une méthode formelle d'apprentissage par renforcement pour garantir la sécurité.

La première étape consiste à obtenir un environnement de simulation en modélisant avec précision l'environnement réel du système de conduite autonome, à partir duquel des facteurs environnementaux non déterministes sont obtenus (la météo, les comportements des autres conducteurs, les obstacles imprévus, etc.), et ces facteurs seront analysés pour classifier les risques. Une machine appelée machine de récompense est utilisée pour guider la fonction de récompense dans RL. Le moniteur est utilisé pour définir la barrière de surveillance, qui contrôle les actions de sortie. L'action qui satisfait les propriétés de sécurité est sélectionnée dans l'ensemble d'actions afin de garantir la décision. La quatrième étape est l'exécution de routine de l'apprentissage par renforcement où l'agent reçoit des données de l'environnement de simulation et il exécute les actions.

Enfin, ils ont utilisé le simulateur projet de voiture autonome Udacity pour vérifier l'efficacité de la méthode dans la conduite intelligente en utilisant des scénarios de dépassement et de changement de voie, et leur approche a été comparée avec l'algorithme DQN. Les deux algorithmes ont presque obtenus les mêmes résultats en ce qui concerne le changement fréquent de voie et le niveau de vitesse, mais quant à la sécurité, l'approche proposée dans cet article a démontré un succès plus élevée avec un taux de sécurité égale à 0.9 face à un taux de 0.2 pour l'algorithme de DQN.

### **7. Decision Making for Automated Driving at Unsignalized Intersection.**

Dans ce travail [16] Les auteurs ont traité la prise de décision dans les intersections non signalés (non-guidés) qui représente l'un des problèmes les plus difficiles dans les véhicules autonomes. Les auteurs ont commencé ce travail par modéliser le trafic avec le réseau bayésien dynamique DBN (DYNAMIC BAYESIEN NETWORK) en considérant le véhicule comme un système Constitué de 3 modules : module de suivi de la vitesse, module de contrôle de la direction, et un module de prise de décision tenant compte des intentions qu'est modélisé comme un processus de décision markovien partiellement observable (POMDP), qui est considéré comme l'un des modèles les plus utilisés pour les problèmes de prise de décision séquentielle dans des environnements incertains.

L'intention de conduire de chaque participant au trafic est déduite par le modèle DBN proposé, plusieurs indices ont été utilisés par les auteurs pour déduire les intentions de conduite des autres conducteurs par le modèle DBN, tels que l'état physique d'un véhicule (position, vitesse, accélération, etc.) et les informations sur le réseau routier (géométrie et topologie de l'intersection, règles de circulation, etc.), avec l'intention prédite des autres participants de trafic le système de prise de décision prendra une décision qui permettra au véhicule d'avoir la récompense élevé et d'éviter les collisions.

Un Test a été fait dans un environnement routier réel avec une voiture équipée d'unités de calcul et de divers capteurs (LiDAR, capteurs de vision, GPS) et l'algorithme proposé a été appliqué avec succès à différentes formes d'intersections.

### **8. Modeling Tactical Lane-change Behavior for Automated Vehicles : A Supervised Machine Learning Approach.**

Dans cet article [29] les auteurs ont étudié l'application des méthodes d'apprentissage supervisé dans le changement de voies des véhicules autonomes, afin d'imiter la conduite humaine



les auteurs ont opté pour prédire le changement de voie d'un véhicule dans un trafic réel.

Ils ont utilisé des données fournies par le simulateur NGSIM (Next Generation Simulator), avec un dataSet de 45 min de conduite, qui sont devisés en 3 parties de 15 minutes dont chaque partie correspond à une période de la journée (horaire de travail, etc.), afin d'entraîner les modèles sur plusieurs situations, Les données disponibles sur le stimulateur étaient des données qui correspond à un chaque véhicule indépendamment (position, vitesse, largeur, hauteur, etc.), Les auteurs donc ont opté pour la création des nouvelles données pour renforcer leurs dataSet à partir des données disponibles, comme exemple la différence entre la vitesse de véhicule connecté à l'instant  $t - 1$  et  $t$  :  $Vego(t - 1) - Vego(t) \dots (a)$ , et d même pour les véhicules qui se trouve devant le véhicule intelligent :  $Vp(t - 1) - Vp(t) \dots (b)$ , et derrière le véhicule autonome :  $Vf(t - 1) - Vf(t) \dots (c)$ .

Ensuite, les auteurs ont calculé l'importance de chaque variable (les informations fournies par le simulateur et les informations créés par les auteurs) avec la feature selction pour la prédiction de changement de voie. Les données (b) et (c) avaient le facteur d'impact le plus grand sur la prédiction du changement de voie. Enfin, les modèles ont été entraînés sur l'ensemble d'entraînement et de tests en utilisant la validation croisée, une fois sur la totalité des informations et une autre fois sur les informations ayant un impact important dans la prédiction. Les résultats ont démontré que les modèles d'arbres de décisions sont les plus puissants avec une précisons près de 100% et les autres modèles ont aussi eu des bons résultats comme le KNN et le SVM, ainsi Ils ont obtenu des résultats similaires qu'ils utilisaient toutes les données ou uniquement les plus pertinentes, ce qui montre l'efficacité de leur sélection de données.

### **9. A Novel Lane Change Decision-Making Model of Autonomous Vehicle Based on Support Vector Machine.**

Dans cet article [30], une approche de prise de décision pour le changement de voie a été élaboré et testé, les auteurs ont commencé leur travail par définir et analyser le changement de voie dans les voitures autonomes et la difficulté de cette tâche, les auteurs ont considéré que pour qu'une voiture décide de changer la voie, elle doit faire une analyse profonde en vérifiant trois aspects qui sont le bénéfice, la sécurité et la tolérance.

le bénéfice est L'objectif de changement de voie, qui est souvent l'amélioration de vitesse ou d'avoir un plus grand espace libre devant soi, pour la sécurité c'est quand le véhicule change de voie afin d'éviter les collisions, et dernièrement la tolérance de conduite afin de garantir une conduite confortable en évitant le changement de voie fréquent, et pour arriver à réaliser ces aspects ils ont proposé d'utiliser l'algorithme de d'apprentissage automatique de type supervisé qui est l'algorithme de SVM en la combinant avec l'optimisation bayésien. Pour construire leur dataset les auteurs ont utilisé Next Génération simulation (NGSIM) qui permet de générer des données des véhicules en temps réel. Les données ont été représentées sous la forme suivante :

$$\begin{cases} D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\} \\ x_i = [v_{income}, G_{TP} - G_P, G_{TR}, v_E - v_{TR}, G_P - v_E \cdot t_h] \\ y_i = \{-1, 1\} \end{cases}$$

Où  $D$  représente l'ensemble des véhicules,  $X_i$  représente les informations du véhicule  $i$  ( et  $Y_i$  représente le changement de voie du véhicule  $i$  à un moment donné. où  $G_{TP}$  représente la distance entre le véhicule intelligent et le véhicule précédent dans la voie cible, tandis que  $G_{TR}$  correspond à la distance entre le véhicule intelligent et le véhicule arrière dans la voie cible. Enfin,  $G_P$  indique la distance entre le véhicule intelligent et le véhicule précédent dans la voie d'origine.

L'algorithme SVM a été utilisé pour trouver des hyperplans dans l'espace d'échantillon sur l'ensemble d'apprentissage permettant de prédire les changements de voie des véhicules en temps réel, et cette algorithme est étroitement lié à la valeur de deux paramètres essentiels qui sont  $C$  et  $\sigma$ , il est donc nécessaire d'optimiser ces deux paramètres. L'algorithme d'optimisation bayésienne (BOA) tente de minimiser une fonction objective scalaire  $f(x)$  pour  $x$  dans un domaine borné afin de trouver les meilleures valeurs de  $C$  et  $\sigma$ .

Leur approche a été testée dans un trafic réel et comparée à d'autres algorithmes qui utilisent l'algorithme SVM sans l'optimisation bayésien, les résultats ont démontré que leur approche a surmonté les autres approches dans la moyenne de précision.

### **10. Machine learning method to ensure robust decision-making of AVs.**

Dans cet article [31] les auteurs ont commencé par présenter différents risques concernant la problématique de changement de voie. Ils ont commencé par la création d'une fonction mathématique pour évaluer le risque de collision, cette fonction prend en considération les vitesses relatives entre le véhicule connecté et les véhicules avant et latéraux, les distances de collision entre le véhicule connecté et les véhicules avant et latéraux ainsi que les temps de collision entre ces véhicules.

Ils ont proposé une approche d'apprentissage automatique en utilisant l'algorithme AdaBoost, qui convient à la classification binaire dans laquelle un classificateur fort est obtenu sur la base d'une minimisation itérative des critères de perte exponentielle et d'une combinaison linéaire de classificateurs faibles. Ainsi que cette approche repose sur l'utilisation de SCANeR-studio, un outil qui permet de tester et d'analyser les systèmes de conduite autonome dans un environnement virtuel. Ce simulateur inclut des modèles de caméras et de radars pour la perception, des modèles de véhicules connectés et d'autres véhicules en tant que perturbations, ainsi qu'un modèle pour simuler le comportement humain du conducteur. Il prend également en compte plusieurs paramètres environnementaux tels que les conditions météorologiques, l'éclairage, les infrastructures et les autres usagers de la route.

Les auteurs ont clôturé leur travail en faisant de tests, Les résultats montrent que l'apprentissage automatique basé sur la simulation est un moyen efficace de gérer les différents scénarios auxquels est confronté le véhicule autonome et que cette approche peut améliorer considérablement la prise de décision.

### **11. Interactive Decision Making for Autonomous Vehicles in Dense Traffic.**

Dans cet article [32], les auteurs ont étudié comment un agent autonome peut interagir en toute sécurité avec d'autres participants à la circulation. Leur contribution vise à démontrer comment la prise de décision avec l'utilisation d'un arbre de jeu, peut être réalisée par un véhicule autonome en utilisant des approximations et des raisonnements pour rendre la recherche de l'arbre réalisable.

Dans cette étude, les auteurs ont utilisé la notation de variable  $V_{time}^{agent,action}$  pour calculer les probabilités. Dans le jeu, chaque agent  $i$ , à un instant  $t$  et dans un état  $s$  donné, choisit une action  $a$  en fonction de sa politique de conduite. Tous les agents passent ensuite à un nouvel état  $s_{t+1}$  et reçoivent une récompense  $r$ . La fonction de transition  $P : S \times A \times S \rightarrow [0, 1]$  décrit comment les états  $S$  évoluent lorsque les agents exécutent des actions  $A$ . L'objectif du véhicule est donc de maximiser ces récompenses lors d'une manœuvre de changement de voie en utilisant un arbre de jeu. Les nœuds de l'arbre représentent les états, chaque branche représente un ensemble d'actions possibles, et chaque nœud est associé à une récompense.

Résoudre le problème revient donc à trouver la séquence d'actions optimale avec le meilleur coût.

Après avoir effectué une recherche préliminaire dans l'arbre, les participants interactifs à la circulation sont identifiés, et leurs intentions seront prédites, et le véhicule connecté évalue les propres intentions, puis agit, observe et met à jour les modèles de probabilité.

Enfin les auteurs ont fait des tests où différentes profondeurs de l'arbre de décision ont été évaluées et comparées pour analyser leur influence sur les performances du système, et leur approche démontre une méthode efficace de fusion interactive dans la simulation.

## **12. An Improved Safety Algorithm for Artificial Intelligence Enabled Processors in self driving cars.**

Dans cet article [33] un algorithme basé sur des règles a été proposé pour permettre d'augmenter la sécurité de trafic, en utilisant l'intelligence artificielle dans la phase de l'analyse de l'environnement afin de permettre une analyse plus performante et plus rapide qui facilite la prise de décision. L'algorithme proposé est appelé CNO-LQP a pour but de trouver la meilleure décision à prendre en fonction des règles définies. Le capteur de distance intégré dans les voitures nous permet d'identifier la distance entre les véhicules qui partagent la route avec les voitures autonomes. La méthode proposée intègre l'optimisation non linéaire conjuguée et la propagation linéaire quadratique gaussienne (CNO-LQP) afin d'améliorer la sécurité routière en fournissant une optimisation de trajectoire parfaite et une planification de mouvement et de contrôle du véhicule adéquates. Une fois la trajectoire est planifiée l'algorithme proposé génère des commandes permettant de suivre cette trajectoire choisie et de les envoyer à l'unité de contrôle électronique (ECU) pour les exécuter. Les règles essentielles mentionnées dans l'article sont :

- Si l'obstacle détecté est un piéton, le système envoie un ordre de freinage.
- Si l'obstacle est un autre véhicule, le CNO-LQP génère l'ensemble des commandes permettant de changer la voie et éviter la collision.
- Si un feu de circulation est détectée, l'algorithme choisi un comportement selon la couleur du feu.

Leur modèle a été testé dans le temps réel, et les résultats montrent que la sécurité s'est améliorée en ce qui concerne la détection des obstacles, le contrôle de véhicule et le changement de voie.

## **2.4 Etude comparative**

### **2.4.1 Classification des travaux passés en revue**

Les solutions proposés peuvent être classées selon deux classes, la première classe est celle des méthodes dotés de l'IA, nous parlons donc de l'apprentissage automatique, le type le plus utilisé dans ce type est l'apprentissage par renforcement, d'autres méthodes dans le même type aussi souvent utilisées sont les méthodes l'apprentissage supervisé

la seconde classe utilisée est la classe des méthodes classiques basés sur des algorithmes basiques en utilisant des règles prédéfinis. La figure 2.1 présente la classification des articles étudiés dans cette section.

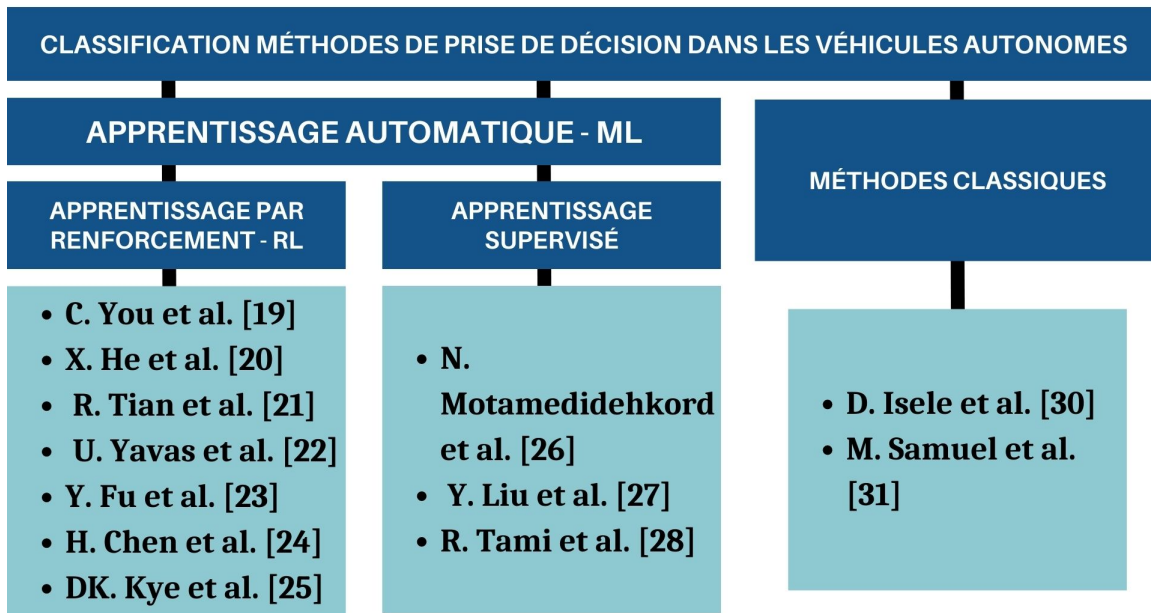


FIGURE 2.1 – Classification des méthodes de prise de décision

### 2.4.2 Tableau comparatif

Afin de comparer les travaux nous nous sommes focalisés sur les critères suivants :

1. **Type de décision à prendre** : Selon les actions prises en compte dans les travaux, le véhicule autonome prend des décisions latérales et/ou longitudinales. Le type de décision choisi est généralement lié au scénario étudié dans l'article. Si les décisions prises concernent des actions latérales, le scénario étudié est souvent le changement de voie. Pour les décisions longitudinales, il s'agit du contrôle de la vitesse. Si tous ces types de décisions sont pris, l'article aborde alors la circulation des véhicules de manière générale.
2. **Environnement** : L'un des principaux enjeux des voitures autonomes réside dans la complexité des environnements. Souvent, les études se focalisent sur des voies à sens unique, mais d'autres travaux portent sur des environnements spécifiques tels que les virages, les intersections, les ronds-points, etc.
3. **Type de l'approche utilisé** : Plusieurs types de solutions ont été proposés, notamment des solutions basées sur l'intelligence artificielle, qui sont divisées en deux parties : l'apprentissage par renforcement et l'apprentissage supervisé. D'autres méthodes reposent sur des algorithmes classiques et/ou des probabilités.
4. **Algorithme utilisé** : Selon l'approche utilisée, chaque article propose un algorithme permettant de prendre des décisions sûres. Ces articles proposent soit de nouvelles dérivations d'algorithmes existants, soit une nouvelle utilisation d'un algorithme existant.
5. **Systèmes de prédiction** : Dans les environnements complexes où le comportement des autres véhicules est inconnu pour le véhicule autonome, ce dernier a besoin d'un mécanisme de prédiction du comportement des autres véhicules et des perturbations de l'environnement. Certains chercheurs intègrent ce mécanisme dans l'algorithme proposé pour la prise de décisions, tandis que d'autres utilisent ou créent un système distinct dédié à la prédiction.

6. **Coût** : Les décisions prises par le véhicule autonome doivent être rapides, dans des délais très courts (en millisecondes). Pour effectuer ces tâches dans un laps de temps suffisamment court, il est nécessaire d'utiliser des technologies coûteuses en termes de temps et de complexité de calcul.

Le tableau 4.3 illustre une étude comparative des travaux passés en revues.

Article	Critère 1	Critère 2	Critère 3	Critère 4	Critère 5	Critère 6
Article [23]	Les deux types	autoroute à plusieurs voies (Virtuel)	Par renforcement	Q-Learning	inclut	Élevé
Article [24]	Les deux types	autoroute à plusieurs voies (Virtuel)	Par renforcement	OARL	Optimisation Bayésien	Très élevé
Article [25]	Les deux types	Rond point (Virtuel)	Par renforcement	Q-Learning	Théorie des jeux	Élevé
Article [26]	Latérales	autoroute à plusieurs voies (Virtuel)	Par renforcement	Rainbow DQN	inclut	Élevé
Article [27]	Longitudinales	3 voies à sens unique (Virtuel)	Par renforcement	Actor Critic	inclut	Très élevé
Article [28]	Les deux types	3 voies à sens unique (Virtuel)	Par renforcement	DRL	inclut	Élevé
Article [16]	Les deux types	Circulation réel	Par renforcement	Q-Learning	DBN	Élevé
Article [29]	Latérales	autoroute à plusieurs voies (réel)	Supervisé	DT, KNN, SVM ...	inclut	Moyen
Article [30]	Latérales	Circulation réel	Supervisé	SVM	inclut	Moyen
Article [31]	Les deux types	Deux voies à sens unique (Virtuel)	Supervisé	AdaBoost	inclut	Moyen
Article [32]	Les deux types	Circulation réel	Classique	Règles Prédéfinies	inclut	Faible
Article [33]	Latérales	Voie à sens unique (Virtuel)	Classique	Arbre de jeu	inclut	Faible

TABLE 2.1 – Comparaison des articles.

### 2.4.3 Discussion

En comparant les travaux nous trouvons que certains sont focalisés sur tous les types de décisions possibles (longitudinale et latérale), mais d'autres [26],[27],[29],[30],[31] étudient certains décisions, quant à l'environnement, la plupart des travaux proposent des méthodes dans un environnement de plusieurs voies et plusieurs voitures comme [23] [26], certains proposent

des méthodes dans un environnements précis (rond-point [24], intersection [25]), et afin de tester et valider leurs approches ils utilisent des environnements de simulations virtuels.

Le type le plus utilisé c'est l'apprentissage par renforcement, dans le RL l'étude se base sur ce qu'on appelle des relations Etat-actions, premièrement, il faut définir les actions possibles pour l'agent (accélérer, décélérer, tournage) qui est le véhicule autonome, et les caractéristiques de son environnement (route, voies, véhicules, obstacles, conditions météo [28], etc.) à chaque instant qui est l'état, puis définir une fonction de récompense qui sert à donner des punitions si une action exécutée génère un état critique (accident, collision, violation des règles, etc.) et une récompense positive dans le cas où l'agent arrive à son but en toute sécurité ou quand il fait de bons comportements, l'agent donc va exécuter ses tâches en suivant une stratégie, et à chaque exécution il sauvegarde l'ensemble des actions exécutées, les récompenses et l'état de l'environnement pour construire son propre data Set, au fur et à mesure des scénarios d'entraînement l'agent arrive à choisir des actions qui vont lui permettre de se circuler en toute sécurité, ces méthodes sont efficaces et adaptées à toutes les situations complexes, mais elles sont coûteuses à cause de l'utilisation des réseaux de neurones complexes.

d'autres méthodes dans le même type sont les méthodes l'apprentissage supervisé, où il faut commencer par la construction d'un data Set des scénarios déjà exécutés en sauvegardant des données de l'environnement (position, vitesse, actions, distances, etc.) à chaque instant, pour entraîner le système de prise de décision, l'agent va comparer la situation dans laquelle il se trouve avec l'ensemble des situations existantes dans son DATASET afin de trouver la plus similaire pour choisir l'action la plus adéquate. Ces méthodes sont également efficaces et leur avantage réside dans leur coût, qui est inférieur à celui du premier type abordé. mais l'inconvénient est quand le véhicule se trouve dans une situation non traitée dans son environnement.

Dans le type des méthodes classiques, la prise de décision est basée sur des algorithmes [32] et/ou des probabilités [33], dans ce type le système de prise de décision suit des règles précises pour choisir des actions qui sont souvent des combinaisons (contraintes, action), lorsque un sous-ensemble de contraintes est vérifiés ou le seuil d'une probabilité est dépassé ; l'action associé sera sélectionné. Ce dernier type est aussi efficace mais pour les situations simples.

Un autre point très essentiel, et le système de prédiction, ce système sert à fournir les données nécessaires au fonctionnement de système de prise de décision. Avec l'avancement des méthodes et techniques utilisés dans ce domaine, les auteurs cherchent à proposer des systèmes de prise de décision qui lui-même capable de prédire comme [23] et [28], et d'autres préfèrent d'utilisent d'autres systèmes (ou mécanismes) pour résoudre ce problème de prédiction en créant de méthodes par eux-mêmes [16] ou en faisant appel à des méthodes mathématiques [28] ou à la théorie de jeux [25]

## 2.5 Conclusion

Ce deuxième chapitre nous a permis de voir les différentes méthodes et techniques utilisées dans la prise de décision. Après avoir comparé les méthodes de différentes classes, nous avons remarqué que chaque classe et type de méthode présente des avantages et des inconvénients.

Dans le chapitre suivant, nous présenterons une nouvelle approche qui s'appuie sur l'intégration d'une méthode étudiée dans le type d'apprentissage supervisé avec le principe des méthodes de type d'apprentissage par renforcement afin de bénéficier des avantages de chaque type.

## Chapitre III

# *Proposition d'une approche de prise de décision basée sur l'intégration de l'apprentissage par renforcement avec KNN*

### 3.1 Introduction

Dans ce chapitre, nous présentons notre proposition pour la résolution du problème de la prise de décision dans les véhicules autonomes. Nous commençons par une brève présentation de la motivation qui nous a poussés à travailler sur ce sujet. Ensuite, nous présentons notre approche en détaillant les étapes de sa mise en œuvre. Nous terminons ce chapitre par la présentation des différents outils que nous avons utilisés, ainsi que l'évaluation de notre approche, à travers des tests et une discussion des résultats obtenus.

### 3.2 Motivation

Comme abordé dans le chapitre précédent, y'a des approches qui se limitent soit aux manœuvres latérales [26], soit aux manœuvres longitudinales [27], ce qui peut limiter leur robustesse dans des environnements complexes. Notre proposition vise à surmonter ces limites en utilisant l'apprentissage par renforcement pour permettre au véhicule autonome de prendre des décisions sur les deux axes de la conduite, assurant ainsi sa robustesse. Ainsi que comme chaque approche étudie des environnements de conduite précis (comme les ronds-points [25], intersection [30]), nous avons opté pour un environnement de plusieurs voies à sens unique. Nous prévoyons dans cette proposition également d'adapter cette méthode pour permettre au véhicule d'apprendre des critères supplémentaires tels que le confort de conduite, le respect des règles et la sécurité. En outre, étant donné que les perturbations sont fréquentes sur la route, le véhicule autonome doit disposer d'un mécanisme lui permettant de réagir face à ces perturbations [24]. Notre approche vise à doter le véhicule autonome d'un mécanisme de réaction en cas de perturbation.

Dans notre approche, nous utiliserons l'algorithme de KNN (K le-plus proches voisins) [15] pour prendre des décisions basées sur les connaissances acquises. L'avantage de KNN réside dans sa capacité à effectuer des prédictions en se basant sur la similarité des échantillons d'entraînement, ce qui est particulièrement utile dans un problème dédié aux distances. En utilisant les données collectées lors des interactions avec l'environnement, nous pourrions estimer les meilleures actions à prendre en fonction de la similarité entre les nouvelles situations rencontrées et les situations déjà apprises. Cela permettra au véhicule autonome de réagir de

manière appropriée.

### 3.3 Notre Proposition

Dans cette section, nous expliquons la mise en œuvre de notre proposition. Nous commençons par définir les différents acteurs impliqués dans cette approche. Ensuite, nous présentons les trois étapes de mise en œuvre de l'approche : la première consiste à construire la mémoire de l'agent, la deuxième à élaborer le modèle de l'agent, et la dernière à la prise de décision.

#### 3.3.1 Acteurs clés de notre approche

Dans cette section nous présentons les différents acteurs utilisés dans notre approche :

##### 3.3.1.1 Agent

L'apprentissage par renforcement est un processus dans lequel un agent interagit dans un environnement en prenant des décisions, également appelées actions. Dans notre étude, nous considérons un véhicule autonome comme l'agent qui doit être capable de prendre de nombreuses actions pour pouvoir réagir efficacement à différentes situations. D'autres études ont pris en compte des actions telles que changer de voie vers la gauche ou à la droite [34], [35], l'accélération et la décélération [36], du véhicule. Dans notre étude, nous nous concentrons sur les décisions à deux niveaux (ajoute longitudinal et latérale) et définissons l'agent comme un véhicule intelligent avec ces actions :

- Action 0 – Pas de changement.
- Action 1 – Augmenter la vitesse d'un niveau.
- Action 2 – Diminuer la vitesse d'un niveau.
- Action 3 – Changer de voie vers la gauche.
- Action 4 – Changer de voie vers la droite.

Notez que l'objectif de « l'action 0 » est de permettre à l'agent de ne rien faire s'il juge que la meilleure chose à faire dans l'état dans laquelle il se trouve est de ne pas faire des changements.

##### 3.3.1.2 État

L'état est l'ensemble des données récolté à chaque instant à partir de l'environnement par le véhicule autonome, certaines approches [23], [24] se focalisent sur la position des véhicules adversaires ainsi que leur vitesse, [30] ont proposé de récolter les distances qui sépare le véhicule avec les obstacles, dans notre approche nous essayons de définir un état permettant au véhicule de gérer et d'analyser l'environnement d'une manière efficace, nous avons proposé donc d'utiliser les données suivantes :

#### a) La position de véhicule par rapport à la route :

La connaissance de la position du véhicule ainsi que de la voie sur laquelle il se trouve est essentielle pour garantir la sécurité du véhicule. En effet, ces éléments permettent d'éviter toute collision potentielle.



**b) La vitesse actuelle, la meilleure vitesse du véhicule, l'intervalle de vitesse sur la voie :**

Grâce à ces informations, l'agent pourra ajuster sa vitesse en fonction de la plage de vitesses autorisées sur la voie, en visant sa vitesse optimale

**c) Distance avec les obstacles :**

La sécurité routière est un enjeu crucial dans la conduite autonome, car l'agent doit absolument éviter les collisions avec les autres obstacles sur la route. Pour atteindre cet objectif, il est nécessaire non seulement de connaître leur position, mais aussi de prédire leur mouvement. Ainsi, nous suggérons de récupérer les distances avec les obstacles en utilisant la méthode illustré dans la figure suivante :

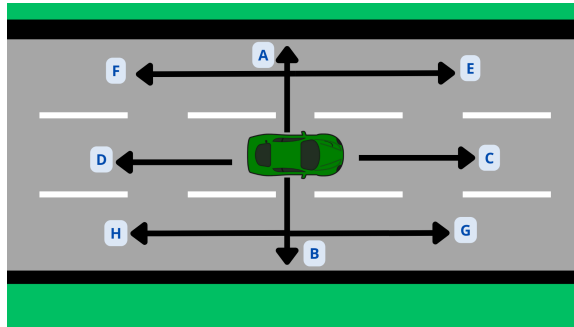


FIGURE 3.1 – Distribution des capteurs de véhicule.

Comme illustré dans 3.1, afin de garantir que le véhicule autonome disposera d'un maximum d'informations sur la route. Les points A et B représentent respectivement la distance latérale entre le véhicule et les véhicules qui se trouvent dans le même axe latérale. Les points C et D indiquent la distance entre le véhicule intelligent et les autres véhicules les plus proches se trouvant sur la même voie, où C représente la distance avec le véhicule devant et D représente la distance avec celui derrière. En ce qui concerne les points E, F (respectivement G, H), ils représentent les distances longitudinales avec les véhicules situés à gauche (respectivement à droite) du véhicule autonome. Cette distribution des points vise à fournir à l'agent un accès optimal aux informations cruciales pour sa prise de décision.

Dans la figure suivante nous expliquons le fonctionnement et l'utilisé de cette distribution.

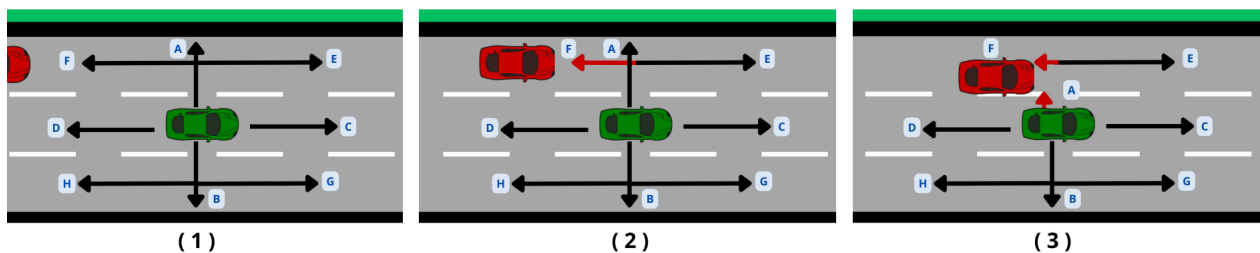


FIGURE 3.2 – Fonctionnement des capteurs du véhicule.

Comme illustré dans la figure 3.2.1, toute partie de la surface autour de véhicule est capté par l'un de ses capteurs, ce qui donne au véhicule un accès aux informations dont nous jugeons

suffisants pour capter toute perturbation possible, Dans Le premier cas 3.2 (1) l'agent ne détecte aucun obstacle, tandis que dans le deuxième cas 3.2 (2), l'obstacle entre dans le champ de capteurs du véhicule et il est capté par le capteur F, la distance récupéré par ce dernier informera l'agent qu'il y'a un véhicule qui se trouve dans la voie à gauche de véhicule qui se rapproche de lui, à partir de là, la valeur récupérée par le capteur F jouera un rôle très important dans la détection d'une perturbation, En Effet, Si la valeur est très petite cela veut dire que le véhicule est très proche de l'agent, en parallèle, si la distance récupéré par A est fixe cela veut dire que le véhicule adverse maintient sa position latérale, mais comme il est illustré dans 3.2 (3) tout changement dans la valeur récupéré par le capteur A dont cette valeur est plus petite veut dire que il y'a une perturbation et le véhicule adverse se rapproche directement pour générer une collision

### 3.3.1.3 Fonction de récompense

La fonction de récompense est cruciale pour l'apprentissage de l'agent en lui attribuant des récompenses ou des punitions selon la qualité de l'action exécutée sur l'état actuel. Certains chercheurs ont proposé de la décomposer en parties qui garantissent la satisfaction de critères de conduite [30], d'autres [28] notamment ont incluant des contraintes pour garantir le respect des règles de la route. Notre approche consiste à diviser la fonction de récompense en parties, chaque critère de conduite que nous souhaitons maîtriser est représenté par une partie distincte de contraintes. Chaque partie a pour objectif d'intégrer spécifiquement l'un de critères de la conduite dans la fonction de récompense. Dans la suite, nous décrivons comment notre fonction de récompense est divisée pour prendre en compte ces différents critères :

$$f(s, a, s') = \left\{ \begin{array}{ll} \text{collision} & \text{(a)} \\ \text{maintenir la direction et la vitesse} & \text{(b)} \\ \text{modification de la ligne et/ou de la vitesse} & \text{(c)} \\ \text{conduire avec la meilleure vitesse} & \text{(d)} \\ \text{positionnement au milieu de la voie} & \text{(e)} \\ \text{ne pas conduire à la meilleure vitesse} & \text{(f)} \\ \text{ne pas se tenir au milieu de la route} & \text{(g)} \\ \text{atteint son objectif} & \text{(h)} \\ \text{mouvement illégal} & \text{(i)} \end{array} \right.$$

#### Partie 01 - La sécurité :

Les contraintes constituant cette partie sont dédiées à la sécurité du véhicule, afin de garantir la sécurité l'agent doit en capable d'éviter toute collision, nous prenons des mesures pour cela en infligeant une pénalité sévère en cas de collision (a).

#### Partie 02 - Conduite confortable :

Dans le but d'améliorer le confort de conduite, il est important que le véhicule n'effectue pas des actions qui peuvent perturber le passager telles que les changements fréquents de voie ou de vitesse. Pour résoudre ce problème, nous avons décidé de donner des petites récompenses à chaque fois l'agent maintient sa position et sa vitesse (b), ou de donner des petites pénalités à

chaque fois que le véhicule effectue un changement de voie ou de vitesse inutile ( $c$ ), de manière à ce que l'agent n'effectue ces changements que s'il y a une bonne raison. D'autre part, nous accordons des récompenses à l'agent lorsque certaines conditions optimales sont remplies. Par exemple, lorsque la conduite est réalisée de manière optimale avec la meilleure vitesse possible ( $d$ ), lorsque l'agent parvient à se positionner au centre de la voie ( $e$ ), ainsi que nous visons à donner des petites punitions lorsque ses contraintes ne sont pas respectées ( $f$  et  $g$ ) et lorsque le véhicule parvient à sa destination ( $h$ ), une récompense lui est attribuée. Ces réussites sont encouragées en offrant des récompenses appropriées, ce qui incite l'agent à se comporter de manière souhaitée pour maximiser sa performance dans ces aspects spécifiques.

### **Partie 03 - Respect des règles :**

Le respect des règles de conduite est un critère crucial pour l'agent dans notre approche. Pour s'assurer que toutes les décisions prises respectent les règles définies (dans le cas d'une autoroute nous considérons comme violation des règles les actions de dépasser à droite ou de dépasser la vitesse maximale autorisée), des punitions sont attribuées à chaque violation de ces règles ( $i$ ).

La division de la fonction de récompense présente un avantage majeur pour l'entraînement, elle permet d'obtenir des informations détaillées sur la performance de l'agent par rapport à chaque critère spécifique. Cela favorise un apprentissage plus ciblé et efficace, ce qui conduit à des performances globales améliorées de l'agent dans la conduite du véhicule. En divisant la fonction de récompense en parties distinctes, nous pouvons mieux évaluer et optimiser les actions de l'agent en fonction de chaque critère individuellement, ce qui contribue à un apprentissage plus précis et équilibré.

Les données collectées incluent *l'état*, *l'action* prise par l'agent et *la récompense* associée. Ces données sont utilisées pour entraîner l'agent à prendre des décisions optimales dans l'environnement.

## **3.3.2 Construction de l'agent**

### **3.3.2.1 Construction de la mémoire de l'agent (Collecte des données)**

Dans cette étape, l'agent interagit de manière aléatoire avec l'environnement, sans aucune stratégie préétablie. L'objectif est d'explorer l'espace des actions et des états pour collecter des données d'apprentissage. Tout d'abord, à chaque itération de l'entraînement, des informations de l'environnement seront captées par les différents acteurs de véhicule après l'exécution d'une action par l'agent, ces informations seront envoyées à l'agent en associant la récompense obtenue. Ensuite l'agent enregistre l'ensemble des données : informations de l'état  $s$ , l'action exécuté  $a$ , la récompense obtenue  $r$  et les informations de l'état suivant  $s+1$  (l'état qui se produit en exécutant l'action  $a$  sur l'état  $s$ ) dans sa mémoire, dans la mémoire de l'agent, à chaque enregistrement une nouvelle information s'ajoute qui est La Q-value, Les Q-values représentent l'estimation de la récompense attendue pour chaque paire d'action-état. L'équation de Bellman [37] est utilisée pour calculer ces Q-values en prenant en compte la récompense immédiate et l'estimation de la récompense future.

$$Q(s, a) = Q(s, a) + \gamma \cdot \max_{a'} Q(s', a') \quad (3.1)$$

Les Q-values sont calculées itérativement à partir des données d'apprentissage et ils se mettent à jour à chaque étape de l'apprentissage, l'agent ensuite refait ces étapes en choisissant toujours l'action aléatoirement jusqu'à arriver à la fin de l'entraînement.

Ces interactions aléatoires permettent à l'agent de découvrir différentes situations et d'acquérir des connaissances sur l'environnement. La mémoire construite dans cette étape sera utilisée dans la prochaine étape comme un jeu de données pour entraîner notre modèle.

La figure 3.3 illustre l'interaction de l'agent avec l'environnement et la construction de sa mémoire.

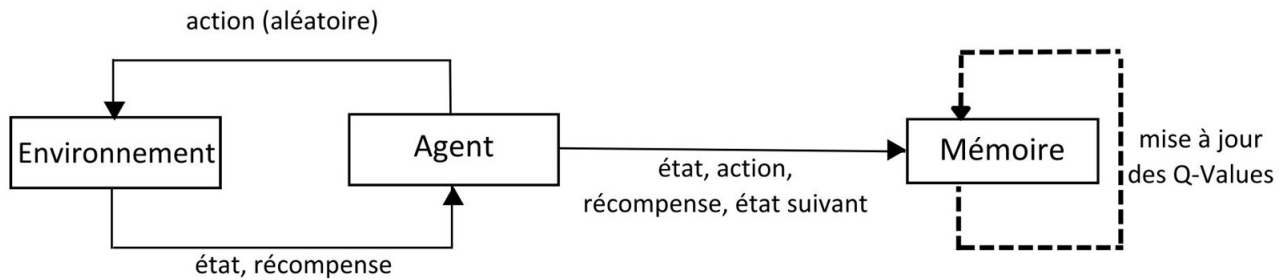


FIGURE 3.3 – Collecte des données

A la fin de cette étape la mémoire de l'agent sera construite, et elle représentera le jeu de données permettant d'entraîner l'agent.

Le jeu données est constitué de :

- Les informations de l'état actuel :
  - $A, B, \dots, H$  : distance récupéré par les capteurs à l'instant  $t$ .
  - $V$  : vitesse de véhicule à l'instant  $t$ .
  - $Y$  : position longitudinale de véhicule
  - $Lane$  : la ligne dans laquelle le véhicule se trouve à l'instant  $t$ .
- L'action et la récompense :
  - $Action$  : l'action exécutée par l'agent à l'instant  $t$ .
  - $Reward$  : la récompense obtenue en exécutant l'action à l'instant  $t$ .
- La Q Value : La Qualité de l'action sur l'état actuel.

	A	B	C	D	E	F	G	H	vitesse	y	lane	action	reward	Q_value
1	10	10	100	100	95	100	100	100	4	42	0	2	-0.5	0.768256
2	10	10	100	100	94	100	100	100	3	42	0	3	-0.6	5.688852
3	10	10	100	100	93	100	100	100	3	41	0	1	-0.6	10.481420
4	10	10	100	100	91	100	100	100	4	41	0	2	-0.6	20.646638
5	10	10	100	100	90	100	100	100	3	41	0	1	-0.6	-1.442391

FIGURE 3.4 – données qui représentent les informations de l'état actuel, l'action, la récompense et la Q Value

- Les informations de l'état suivant :
  - $\underline{A\_new}, \underline{B\_new}, \dots, \underline{H\_new}$  : distance récupéré par les capteurs à l'instant  $t$ .
  - $\underline{V\_new}$  : vitesse de véhicule à l'instant  $t$ .
  - $\underline{Y\_new}$  : position longitudinale de véhicule
  - $\underline{Lane\_new}$  : la ligne dans laquelle le véhicule se trouve à l'instant  $t$ .

	$\underline{A\_new}$	$\underline{B\_new}$	$\underline{C\_new}$	$\underline{D\_new}$	$\underline{E\_new}$	$\underline{F\_new}$	$\underline{G\_new}$	$\underline{H\_new}$	$\underline{vitesse\_new}$	$\underline{y\_new}$	$\underline{lane\_new}$
<b>1</b>	10	10	100	100	94	100	100	100	3	42	0
<b>2</b>	10	10	100	100	93	100	100	100	3	41	0
<b>3</b>	10	10	100	100	91	100	100	100	4	41	0
<b>4</b>	10	10	100	100	90	100	100	100	3	41	0
<b>5</b>	10	10	100	100	88	100	100	100	4	41	0

FIGURE 3.5 – données qui représentent les informations de l'état suivant après l'exécution de l'action

### 3.3.2.2 Élaboration du modèle

Dans cette étape, l'agent utilise les connaissances qu'il a acquises lors de son entraînement avec l'utilisation de l'algorithme de KNN (k-nearest neighbors) Regressor pour prédire la Q-value qu'est considéré comme l'élément clé pour prendre des décisions sûres.

Le KNN joue un rôle crucial dans notre approche, car il nous permet d'estimer les valeurs en se basant sur des données similaires, ce qui est particulièrement important dans le cadre de l'apprentissage par renforcement et spécifiquement dans notre problématique. En effet, Dans notre cas le problème de prédiction des récompenses peut être abordé sous forme d'égalités et d'inégalités. Par exemple, pour éviter les collisions, il suffit d'établir une relation entre la vitesse actuelle du véhicule et les valeurs récupérées par les capteurs. Vu que cette dernière représente une distance dans la réalité, au fur et à mesure de l'entraînement, l'agent parviendra à déterminer une limite au-delà de laquelle tout dépassement provoquera directement une collision. De même, pour la vitesse, grâce au KNN et à la comparaison avec les autres enregistrements, l'agent sera en mesure de déterminer les niveaux de vitesse qui permettent d'obtenir plus de récompenses. Cette capacité n'est pas présente dans les réseaux de neurones, où les prédictions sont basées sur des multiplications visant à trouver des paramètres permettant d'établir des relations linéaires entre les informations de l'agent et la récompense à obtenir.

Tout comme dans l'étape précédente, l'agent reçoit des informations sur l'état à partir de ses capteurs. Il transmet ensuite ces informations au modèle KNN, qui recherche les enregistrements similaires en fonction des informations fournies par l'agent dans 3.4, en tenant compte de l'action.

Dans ce qui suit nous présentons les points essentiels permettant de prédire la Q Value associée à une action en se basant sur cette dernière :

#### a) Codage de l'action :

Comme il est décrit précédemment la recherche se basera sur l'action exécutée, pour cela si nous laissons l'action sur encodé sur une échelle de 1 à 5, l'action aussi sera considéré comme une distance. Pour ceci nous utilisons l'encodage binaire qui permet de distinguer entre les

actions d'une façon efficace. Le codage binaire one-hot est une méthode fréquemment utilisée en apprentissage automatique pour représenter une variable catégorielle ayant n catégories en utilisant n variables binaires distinctes. Cette technique implique de coder la variable ayant n états en n bits, avec un seul bit ayant la valeur 1 et le numéro du bit correspondant au numéro de l'état de la variable [38].

action	Action 0	Action 1	Action 2	Action 3	Action 4
0	0	1	0	0	0
1	2	0	0	1	0
2	3	0	0	0	1
3	1	0	1	0	0
4	2	0	0	1	0

FIGURE 3.6 – Codage de action

**b) Paramètres du modèle :**

Distance : Le choix de la mesure de distance est un paramètre important dans l'algorithme KNN. En effet, cela peut avoir un impact significatif sur la performance de l'algorithme en termes de précision et de temps de calcul. il existe 3 distances, la distance euclidienne qui mesure la distance entre deux points en utilisant le théorème de Pythagore pour trouver la longueur de la ligne droite qui relie les deux points, la distance de Manhattan qui mesure la distance entre deux points en faisant la somme des différences absolues entre leurs coordonnées en x et en y et la distance de Chebyshev, qui mesure la distance entre deux points en prenant la plus grande différence absolue entre leurs coordonnées.

La distance entre deux points, P1(x1, y1) et P2(x2, y2), est calculée en utilisant la formule suivante pour chaque distance [39] :

$$\begin{aligned}
 d(P_1, P_2) &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} && \text{(euclidienne)} \\
 d(P_1, P_2) &= |x_2 - x_1| + |y_2 - y_1| && \text{(Manhattan)} \\
 d(P_1, P_2) &= \max(|x_2 - x_1|, |y_2 - y_1|) && \text{(Chebyshev )}
 \end{aligned}$$

nombre de voisins à considérer - K :

Ce paramètre détermine le nombre de points les plus proches du point à prédire qui seront utilisés pour estimer sa valeur.

Il existe bien d'autres paramètres, comme Le paramètre weights détermine la fonction de pondération à utiliser pour les voisins. et le paramètre algorithme détermine l'algorithme à utiliser pour trouver les k plus proches voisins.

**c) Prédiction des Q-Values :**

Le modèle cherche donc des enregistrements similaires pour chaque combinaison d'informations d'état et d'action en utilisant les paramètres définies au-dessus, autrement dit, le modèle cherche premièrement les exemples similaires aux informations de l'état si l'action exécutée est l'action 0, ensuite il cherchera les exemples similaires si l'action à exécuter est l'action 1, et

ainsi de suite jusqu'à arriver à regrouper 5 sous-ensemble) ce qui donne cinq sous-ensembles d'enregistrements, chacun correspondant à une action spécifique.

Ensuite, le modèle calcule la moyenne des valeurs Q pour chaque sous-ensemble, ce qui représente une estimation de la récompense à gagner pour chaque action. La figure suivante illustre la prédiction des Q-Values associés à chaque action

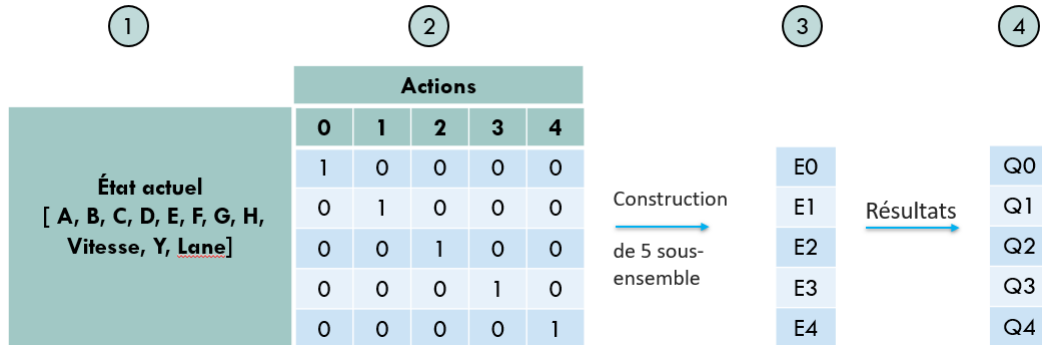


FIGURE 3.7 – Processus de prédiction des Q Values.

A la fin de cette étape, nous aurons créé un modèle capable de prédire les Q values associés à chaque action permettant ainsi à l'agent d'exécuter celle qui correspond à la recomposé la plus maximale.

### Résultats :

Dans ce qui suit nous présentons les résultats obtenus en entraînant notre modèle pour prédire les Q values :

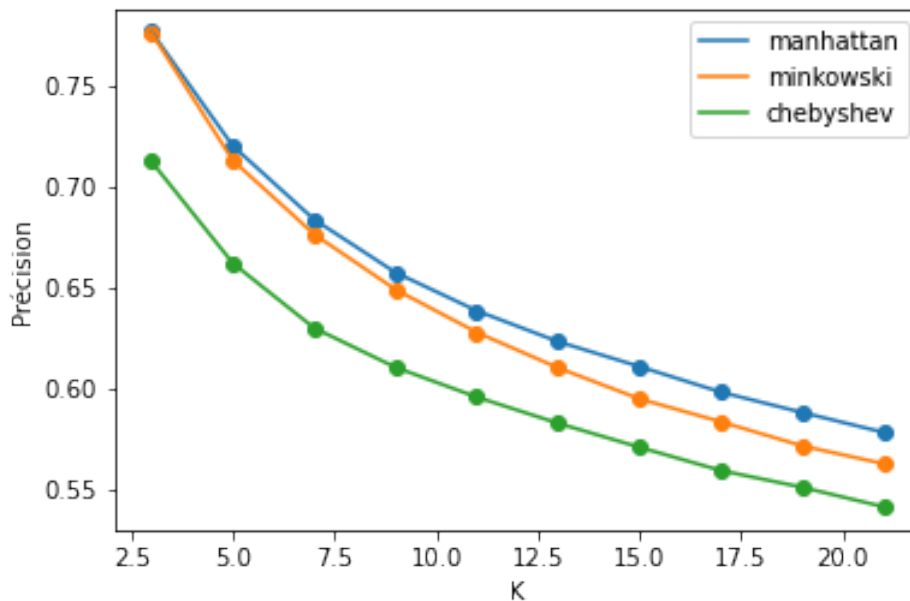


FIGURE 3.8 – Précision de prédiction des Q-Values en fonction de la valeur de K

En analysant ces résultats, nous pouvons observer que la distance *manhattan* semble donner les meilleures performances globales parmi les distances évaluées. Pour  $k=3$ , nous obtenons le meilleur résultat avec une valeur de 0.8 (80 % de précision).

Cependant, pour la distance chebyshev, nous constatons des performances légèrement inférieures avec des scores décroissants à mesure que  $k$  augmente. De même, pour la distance minkowski, nous observons une tendance similaire avec des scores décroissants.

En conclusion, dans cet ensemble de résultats, la distance *manhattan* semble être la plus performante.

### 3.3.2.3 Prise de décision

Ainsi, Les cinq valeurs obtenues dans 3.7 seront ensuite envoyées à l'agent sous forme d'un tableau, où chaque élément contient la valeur  $Q$  associée à son indice dans le tableau. L'agent choisit alors l'indice de l'action ayant la valeur  $Q$  la plus élevée, ce qui correspond à la prédiction de la meilleure récompense potentielle.

L'agent applique ensuite l'action ayant la valeur  $Q$  maximale dans son environnement, avec l'espoir de gagner davantage de récompenses, c'est-à-dire de conduire de manière plus efficace. Ces étapes sont répétées jusqu'à ce que l'agent atteigne la fin de la route.

Toutes les étapes de cette section sont résumées dans la figure suivante :

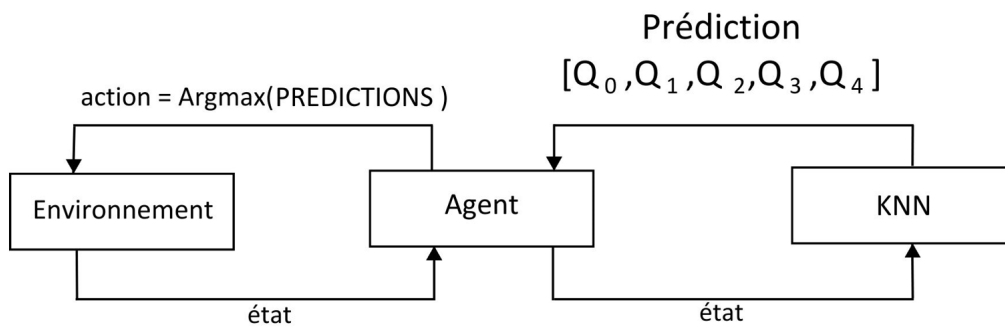


FIGURE 3.9 – Processus de prise de décision.

## 3.4 Conclusion

Ce chapitre a été consacré à la présentation de notre contribution pour la prise de décision dans les véhicules autonomes, Notre méthode est basée sur l'intégration de l'apprentissage par renforcement avec l'algorithme de k-le plus proche voisin. Nous avons commencé par la présentation des acteurs participants dans l'approche, et nous avons aussi décrit les différentes phases de mise en oeuvre de notre approche.



## *Tests et Évaluation*

### 4.1 Introduction

Dans cette section, nous allons nous concentrer sur la phase de test et de validation de notre algorithme d'apprentissage par renforcement. Cette phase est cruciale car elle nous permettra de mesurer la performance de notre algorithme en fonction des critères mises dans la fonction de récompense et de vérifier s'il peut généraliser efficacement à des situations complexes.

### 4.2 Environnement de travail

Les principales caractéristiques de la machines utilisée pour implémenter et tester notre approche sont :

- Type de la machine : LENOVO
- Processeur : Intel Core i3-1115G4 @ 3.00GHz
- RAM : 8 Go
- Système d'exploitation : Windows 11 (64 bits)

### 4.3 Outils et bibliothèques utilisés

#### 4.3.1 Visual Studio

Visual Studio Code (ou VS Code) est un éditeur de code source gratuit et open source développé par Microsoft. Il est disponible pour Windows, macOS et Linux. VS Code offre une interface utilisateur moderne, une prise en charge de nombreux langages de programmation et des fonctionnalités qui simplifient la tâche aux programmeurs [40]

#### 4.3.2 Jupyter

Jupyter est une application web open source qui permet de créer et de partager des documents interactifs contenant du code en direct, des équations, des visualisations et des textes explicatifs. Jupyter prend en charge de nombreux langages de programmation, notamment Python [41]

### 4.3.3 Python

Python est un langage de programmation de haut niveau, interprété et orienté objet. Il est facile à apprendre et à utiliser, ce qui le rend populaire pour une grande variété de tâches de programmation, notamment la science des données, la création de scripts et la programmation Web [42].

### 4.3.4 Numpy

Numpy est une bibliothèque de calcul numérique en Python. Elle fournit des structures de données pour représenter des tableaux multidimensionnels et des matrices, ainsi que des fonctions mathématiques pour travailler avec ces tableaux. Numpy est largement utilisé en science des données, en apprentissage automatique et en traitement d'images [43].

### 4.3.5 Pandas

Pandas est une bibliothèque de manipulation de données en Python. Elle fournit des structures de données pour représenter des tableaux de données, appelés DataFrames, ainsi que des fonctions pour les manipuler, les nettoyer et les analyser. Pandas est utilisé pour le traitement des données en science des données, en finance et dans d'autres domaines [44].

### 4.3.6 PyGame

PyGame est une bibliothèque de développement de jeux en Python. Elle fournit des outils pour créer des graphismes, gérer les entrées utilisateur et jouer des sons dans des jeux en 2D. PyGame est souvent utilisé pour l'enseignement de la programmation de jeux et pour la création de petits jeux indépendants [45].

### 4.3.7 Scikit-learn

Scikit-learn est une bibliothèque logicielle open-source populaire en machine learning pour le langage de programmation Python. Elle offre une large gamme d'algorithmes d'apprentissage automatique supervisés et non supervisés, ainsi que des outils pour la préparation et la transformation des données, la sélection de modèles, l'évaluation des performances et bien plus encore. Scikit-learn est largement utilisé pour développer des modèles prédictifs, effectuer des tâches de classification, de régression, de regroupement (clustering) et de réduction de dimensionnalité. Il est apprécié pour sa facilité d'utilisation, sa documentation exhaustive et sa communauté active [46].

### 4.3.8 Matplotlib

Matplotlib est une bibliothèque de visualisation de données en Python. Elle fournit des outils pour créer des graphiques en 2D et en 3D à partir de données, ainsi que des fonctions pour personnaliser l'apparence des graphiques. Matplotlib est largement utilisé en science des données, en finance et dans d'autres domaines pour la visualisation de données [47].

## 4.4 Création de l'environnement de simulation

Dans le cadre de notre travail nous avons créé un environnement simulé à l'aide de PyGame. Cet environnement est une route à trois voies sur laquelle notre véhicule autonome commence dans l'une des voies avec une vitesse minimale. Notre véhicule autonome dispose de 5 niveaux de vitesse, dont la 3ème est sa meilleure vitesse.

L'environnement est conçu pour charger les autres véhicules (ou "adversaires") de manière aléatoire dans la route. Selon la voie dans laquelle le véhicule obstacle se trouve, il aura une vitesse moyenne associée à cette voie. Le véhicule autonome doit atteindre la fin de la route en évitant les collisions avec les autres véhicules. En cas de collision, l'environnement est réinitialisé pour permettre au véhicule autonome de recommencer son parcours.

Notre objectif en créant cet environnement est de permettre la visualisation du comportement du véhicule autonome et des autres véhicules dans l'environnement. Cette visualisation nous permettra d'analyser les performances de notre véhicule autonome et de comprendre comment il prend ses décisions en temps réel. En observant le comportement du véhicule autonome, nous pourrions identifier les erreurs et les améliorations possibles pour optimiser son comportement, en plus, il nous permettra de fournir un cadre de test pour évaluer les performances de notre véhicule autonome en termes de prise de décision. En utilisant cet environnement, nous pouvons analyser les décisions prises par notre véhicule autonome en temps réel et évaluer son efficacité en termes de sécurité et de rapidité.



FIGURE 4.1 – Environnement de simulation créé par PyGame.

## 4.5 Résultats

Nous avons essayé notre approche dans 6 tests de 100 scénarios de 500 mètres ce qui est égale à 300km (1 test = 100 scénario, 1 scénario = 500 mètre;  $6 \cdot 100 \cdot 500$ ) de conduite, avec un nombre de voitures adversaires égale à 5 dans chaque scénario, et afin d'évaluer et entraîner notre agent de manière approfondie, nous avons décidé de mettre en œuvre plusieurs paramètres dans la fonction de récompense, chacun représentant une stratégie distincte, La Stratégie 1 vise à appliquer une approche simple en attribuant de petites récompenses en cas de bon comportement et de petites punitions en cas de mauvais comportement. Une récompense élevée est accordée lorsque l'agent atteint son objectif, tandis qu'une punition sévère est infligée en cas de collision, La Stratégie 2 vise à accentuer l'échelle des récompenses et des punitions, en attribuant des valeurs très élevées pour les collisions et l'atteinte de l'objectif par rapport aux autres critères. L'objectif est de mettre en évidence l'importance cruciale de ces deux aspects pour l'agent, La Stratégie 3 se concentre davantage sur les bons comportements en attribuant des récompenses en cas de bon comportement et en ignorant les mauvais comportements. Les paramètres restent les mêmes pour l'atteinte de l'objectif, la violation des règles, les collisions, ce tableau montre les différentes paramètres pour chaque contrainte de la fonction de récompense dans chaque stratégie :

Stratégie	a	b	c	d	e	f	g	h	i
Stratégie 1	-100	+1	-1	+2	+1	-2	-1	+100	-10
Stratégie 2	-1000	+0.1	-0.1	+0.2	+0.1	-0.2	-0.1	+1000	-10
Stratégie 3	-1000	+0.1		+0.2	+0.1			+1000	-10

TABLE 4.1 – Fonction des récompenses.

Ces différentes stratégies nous permettent de tester diverses approches de récompense et de punition, nous permettant ainsi de comprendre comment notre agent réagit à chaque scénario et de déterminer les meilleurs paramètres pour optimiser ses performances.

Dans ce qui suit nous explorons les résultats vis-à-vis des contraintes mises dans les sections précédentes :

#### 4.4.1 La sécurité et le respect des règles :

Dans les scénarios effectués nous calculons le nombre de collision dans chaque test afin de calculer en moyenne la précision de chaque stratégie, le tableau suivant 4.2 représente la précision dans chaque test avec la précision finale de chaque stratégie qui est la moyenne des précisions obtenues

La Stratégie 1 a donné des résultats cohérents avec une précision moyenne de 90,7%. Bien que les performances soient relativement stables, il y a une légère variation d'un test à l'autre, avec une précision oscillant entre 88% et 94%. Cela indique que la stratégie est globalement efficace, mais peut encore être améliorée pour réduire l'écart entre les performances maximales et minimales.

La Stratégie 2 a montré des résultats prometteurs avec une précision moyenne de 94,8%. Les tests ont révélé une augmentation significative de la précision, atteignant même 98% dans certains cas. Cela suggère que cette stratégie, qui met l'accent sur les collisions et l'atteinte de l'objectif, a permis à l'agent d'apprendre de manière plus efficace et de prendre des décisions plus judicieuses.

En revanche, la Stratégie 3 a produit des résultats moins convaincants avec une précision moyenne de 87,2%. Les tests ont montré des performances variables, oscillant entre 81% et 93%. Cette stratégie, axée sur les bons comportements, semble moins efficace dans ce contexte et pourrait nécessiter des ajustements pour améliorer les performances.

Stratégie	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Précision
Stratégie 1	90	93	88	94	89	90	90.7
Stratégie 2	91	93	97	97	93	98	94.8
Stratégie 3	87	81	83	89	93	90	87.2

TABLE 4.2 – Tableau des résultats.

En conclusion, les résultats suggèrent que la Stratégie 2 semble être la plus efficace, offrant la précision la plus élevée dans la plupart des tests, pour cela dans la figure suivante nous faire une discussion sur un comportement de l'agent pour éviter une collision en respectant les règles de conduite en utilisant cette stratégie.

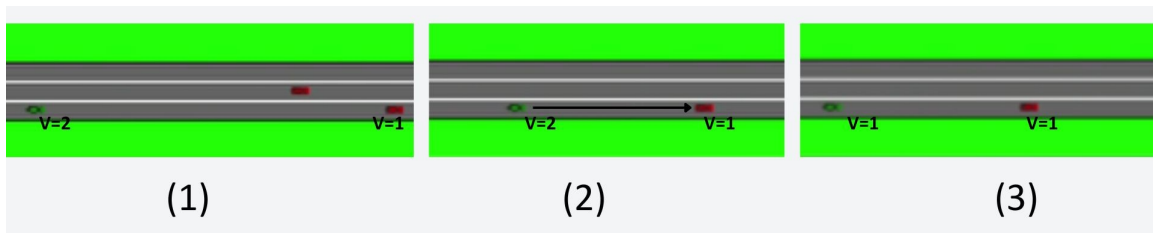


FIGURE 4.2 – Scénario de simulation.

Dans la figure (1) nous remarquons que le véhicule adverse circule devant l'agent avec une vitesse inférieure à celle de l'agent qui se circule avec la vitesse de la voie, l'agent circule avec une vitesse de niveau 2 ( $V=2$ ), et le véhicule adverse circule avec une vitesse de niveau 1 ( $V=1$ ), qui va automatiquement générer une collision en cas de aucun changement de vitesses, dans la figure (2) le véhicule adverse est capturé par le capteur C, les valeurs récupérés par ce dernier se minimisent en fonction de l'avancement de véhicule adverse, l'agent dans la figure (3) afin d'éviter la collision et pour respecter les règles de la conduite il diminue sa vitesse et continue avec la même vitesse jusqu'à arriver au but.

#### 4.4.2 Conduite Confortable :

Pour évaluer le confort de conduite du véhicule, nous avons pris en compte deux aspects : la conduite à la meilleure vitesse possible et la stabilité du niveau de vitesse et de direction du véhicule [28]. Afin de mesurer ces éléments, nous avons analysé la moyenne des vitesses auxquelles le véhicule circule (figure 4.3) ainsi que le nombre d'exécutions de chaque action effectuée (figure 4.4). Ces deux mesures nous permettent d'évaluer la conduite confortable du véhicule et de déterminer si la vitesse est maintenue de manière stable et si les actions exécutées sont cohérentes. Dans l'analyse du graphe des vitesses 4.3, nous observons une tendance générale où la moyenne des vitesses se situe autour de la meilleure vitesse possible, ce qui suggère que le véhicule circule généralement à sa vitesse optimale. Au départ, nous avons constaté une variation entre les différentes stratégies, due à la fluctuation de la moyenne (lorsqu'il y a peu de valeurs, chaque nouvelle mesure affecte considérablement la moyenne), ainsi qu'à la situation spécifique dans laquelle l'agent se trouvait au début des tests. Cependant, à mesure que l'agent progresse dans les tests, nous observons une stabilité de la moyenne et une maîtrise accrue de la vitesse. En comparant les stratégies, nous constatons que la deuxième stratégie se rapproche le plus de la meilleure vitesse. Concernant le deuxième graphe 4.4, pour les première et deuxième stratégies, nous remarquons que l'action la plus fréquemment utilisée est l'action 0 (aucun changement), par rapport aux autres actions. Cela indique que la plupart du temps, l'agent maintient sa vitesse et sa direction pour assurer une conduite confortable. En ce qui concerne la troisième stratégie, le graphe montre qu'en plus de l'action 0, une autre action (action 4) a été exécutée un nombre équivalent de fois. Cela est dû au fait qu'aucune punition n'a été attribuée à ces changements, ce qui peut entraîner une conduite dangereuse et inconfortable. Cela renforce l'hypothèse selon laquelle de petites punitions doivent être attribuées à ces actions, afin que l'agent comprenne qu'elles ne sont pas bénéfiques et qu'il est préférable de maintenir sa position et sa direction.

D'après ces analyses nous constatons que le critère de la conduite confortable a été maîtrisé par notre agent parfaitement dans la première et la deuxième stratégie.

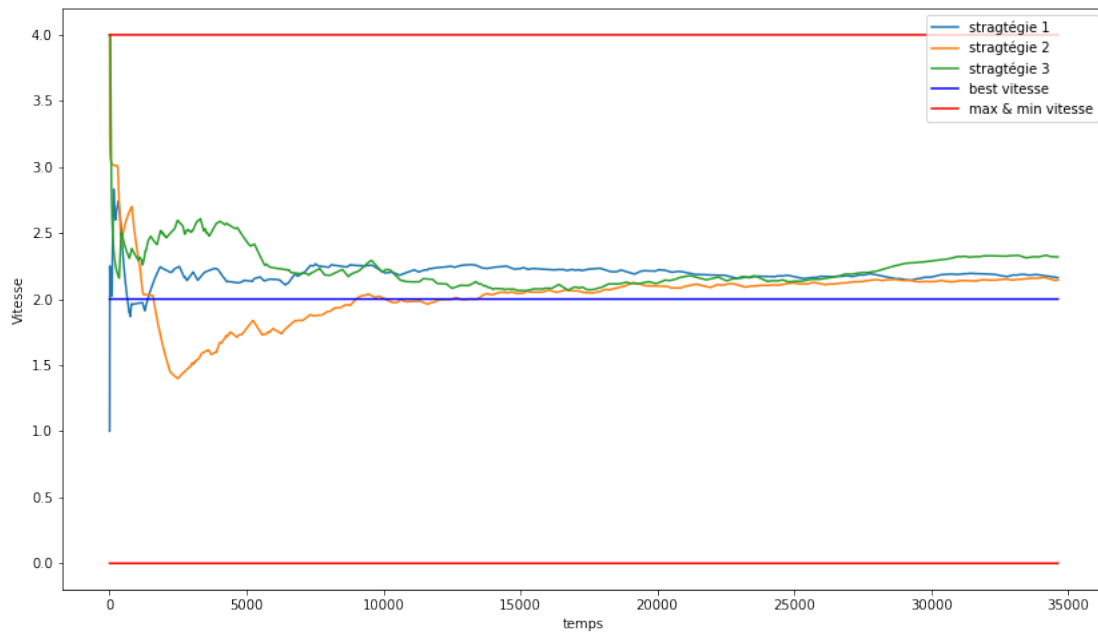


FIGURE 4.3 – Évolution de la vitesse moyenne de l'agent.

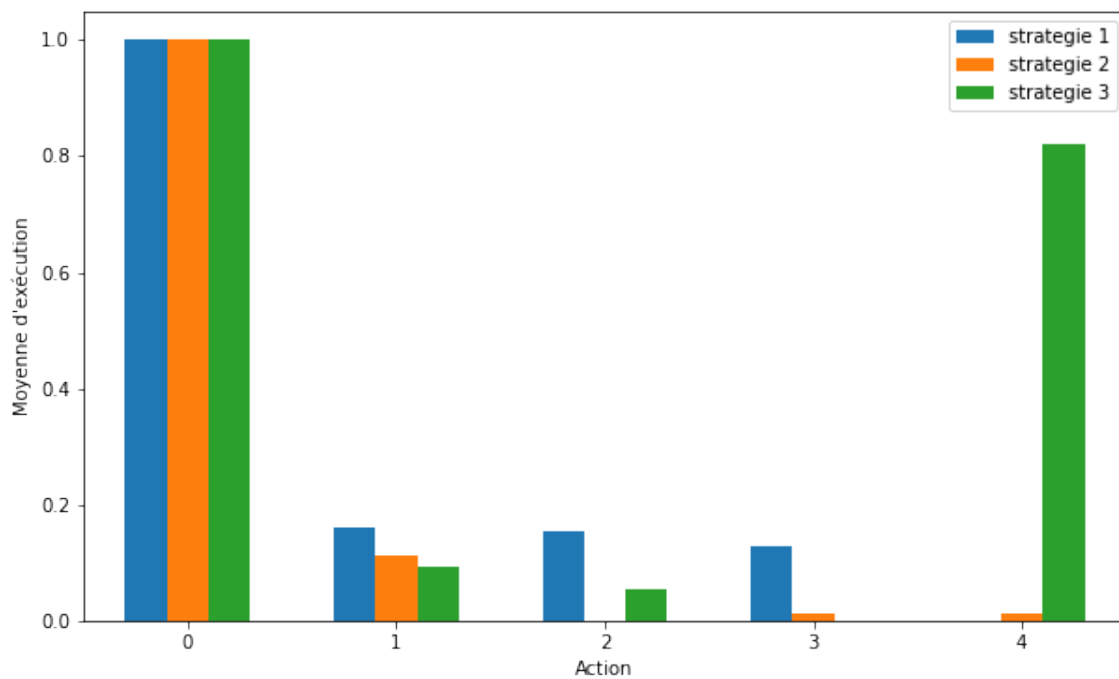


FIGURE 4.4 – Analyse de la fréquence d'exécution des actions.

Donc en résumé les 3 aspects définis dans les sections précédentes ont été maîtrisés parfaitement les 1e et 2e stratégie.

## 4.6 Comparaison avec les approches de l'état de l'art

Pour valider notre approche, nous avons choisi de ne pas seulement visualiser le comportement global du véhicule dans l'environnement de simulation, mais aussi de comparer nos résultats avec d'autres approches, Dans [28], ils ont utilisé plusieurs critères pour valider leurs approches, parmi ces critères nous trouvons deux critères que nous avons aussi étudié. Le premier critère pour mesurer le confort de la conduite est la vitesse moyenne de véhicule (en calculant la vitesse à laquelle l'agent se déplace en moyenne sur une certaine distance), et afin de permettre la comparaison, nous normalisons les données sur une même échelle de 0 à 4. et le deuxième critère est le taux de sécurité (l'inverse du taux collisions). Nous comparons donc notre approche avec cet article, dont les chercheurs ont également comparé leur approche avec deux autres méthodes : l'une basée sur des règles [48], et l'autre utilisant l'algorithme DQN de Tensorflow [49].

Le tableau 4.3 présente les résultats des comparaison effectuées.

Toutes les autres approches ont considéré que la vitesse moyenne du véhicule devait être proche de la vitesse maximale afin d'arriver à destination rapidement et de gagner du temps. Par conséquent, la moyenne de vitesse doit se rapprocher du maximum de l'échelle utilisée, qui est 4. Les résultats obtenus concernant le contrôle de la vitesse ont été prometteurs (3,62 dans [48], 3,69 dans [49] et 3,73 dans [28]). Dans notre approche, nous avons considéré que la vitesse optimale ne tend pas forcément vers la vitesse maximale, notre véhicule n'essaye d'atteindre sa vitesse maximale que quand il veut dépassé un autre véhicule, cela a plusieurs avantages : en l'occurrence la réduction des gaz à effet de serre, l'assurance d'un certain confort aux passagers et une meilleure utilisation des dispositifs du véhicule, ce qui augmentera sa durée de vie. Donc, même dans notre approche, la vitesse moyenne était globalement proche de la vitesse que nous avons considérée comme optimale, qui est de 2, et où nous avons obtenu une moyenne de 2,1.

En termes de sécurité, notre approche a largement surpassé les approches [48] et [49]. Notre approche a également surpassé l'approche de [28] avec un écart significatif (94% contre 90%).

Approche	Niveau de vitesse moyenne normalisé sur une échelle de 0 à 4	Vitesse optimale considéré dans l'approche	différence	Sécurité
Rule-based [48]	3.62	4	<b>-0.38</b>	0.6
DQN-based [49]	3.69	4	<b>-0.31</b>	0.2
Approche de [28]	3.73	4	<b>-0.27</b>	0.9
Notre méthode	2.1	2	<b>+0.1</b>	0.94

TABLE 4.3 – Tableau comparatif.

En conclusion, notre approche d'apprentissage par renforcement pour la prise de décision dans les véhicules autonomes a obtenu de bons résultats, à la fois en termes de vitesse moyenne optimale et en terme de sécurité, par rapport aux autres approches existantes auxquelles nous l'avons comparée. Ceci démontre le potentiel prometteur de notre approche pour la navigation autonome.

### 4.7 Conclusion

Au cours de ce chapitre, nous avons présenté les outils utilisés dans l'implémentation de notre approche. Nous avons également décrit la manière dont nous avons testé l'approche pour évaluer ses performances par rapport aux critères énoncés dans les sections précédentes. Nos résultats démontrent que notre approche décisionnelle pour les véhicules autonomes atteint un taux de réussite très élevé, avec une précision de 94,8%. Cela permet de garantir efficacement le respect des règles et la sécurité. De plus, notre méthode permet au véhicule d'apprendre des critères supplémentaires tels que le confort de conduite et de conduite. Compte tenu des perturbations fréquentes sur la route



## *Conclusion générale et perspectives*

Les véhicules autonomes représentent une avancée technologique majeure dans l'industrie automobile. L'un des défis clés dans le développement de ces véhicules est la capacité à prendre des décisions intelligentes, similaires à celles prises par les conducteurs humains. Les chercheurs travaillent sans relâche pour développer des systèmes capables de reproduire les capacités de raisonnement et de prise de décision des conducteurs humains.

Dans ce mémoire, nous avons proposé une nouvelle approche de prise de décision pour les véhicules autonomes, qui a pour but de développer un système de prise de décision permettant à un véhicule de conduire seul sans intervention humaine, en garantissant sa sécurité, le respect des règles de la route et la conduite confortable. Dans le premier chapitre nous avons présenté les notions basiques permettant de comprendre le sujet en mettant le point sur les véhicules autonomes. Dans le deuxième chapitre nous avons étudié quelques travaux qui traitent ce sujet, en faisant une comparaison entre eux afin de bien comprendre ce sujet. Dans le troisième chapitre, nous avons proposé une nouvelle approche basée sur l'intégration de l'apprentissage par renforcement avec l'algorithme KNN, Dans le dernier chapitre nous avons créé un environnement de simulation afin de tester notre méthode, les résultats ont montré le succès de cette méthode.

Pour les perspectives futures, il est important de continuer à améliorer la performance et la robustesse de cette approche. Nous envisageons d'améliorer la fonction de récompense afin de permettre au véhicule de circuler dans n'importe quel environnement, et de modifier la distribution des capteurs en fonction de l'environnement dans lequel le véhicule évolue, assurant ainsi une couverture optimale pour différents types de routes. Une deuxième perspective serait l'implémentation d'un prototype logiciel à essayer sur des véhicules.

## *Bibliographie*

- [1] A. Mechache, T. Chenache, S. Aissani, and M. Omar. Enhancing decision-making in autonomous vehicles : A study on the integration of k-nearest neighbor algorithm and reinforcement learning. In Colloque international sur les Objets et Systèmes Connectés 202, pages 1–5, 2023.
- [2] KN. Qureshi and AH. Abdullah. A survey on intelligent transportation systems. Middle-East Journal of Scientific Research, 2013.
- [3] P. Zhong-Ren, E. Beimborn, and M. Neluheni. A framework for the evaluation of the benefits of intelligent transportation systems. Wisconsin Department of Transportation, 2000.
- [4] S. Faucheux, C. Hue, and O. Petit. Ntic et environnement : Enjeux, risques et opportunités. Futuribles, 2002.
- [5] C. Badue, R. Guidolini, Raphael V. Carneiro, P. Azevedo, Vinicius B. Cardoso, A. Forechi, L. Jesus, R. Berriel, Thiago M. Paixão, F. Mutz a. Lucas, Thiago O.-Santos, and Alberto F. Self-driving cars : A survey. Expert Systems with Applications, 165 :113816, 2021.
- [6] SAE International. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. Technical Report J3016, SAE International, 2016.
- [7] Centre d'études de l'énergie atomique. Essentiel sur la voiture autonome. <https://www.cea.fr/comprendre/Pages/nouvelles-technologies/essentiel-sur-voiture-autonome.aspx>, 2021. Consulté le 1 juin 2023.
- [8] Opportunités-Technos. Le 6e sens des voitures autonomes passe par la 5g. <https://opportunités-technos.com/le-6e-sens-des-voitures-autonomes-passe-par-la-5g/>, 2023. Consulté le 9 juin 2023.
- [9] Fondation MAIF. Véhicules « autonomes » : en progrès mais pas fiables à 100 %. <https://www.fondation-maif.fr/pageArticle.php?rub=1&id=405>, 2022. Consulté le 11/06/2023.
- [10] Q. Liu, X. Li, S. Yuan, and Z. Li. Decision-making technology for autonomous vehicles : Learning-based methods, applications and future outlook. In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pages 30–37, 2021.
- [11] J. Stilgoe. Machine learning, social learning and the governance of self driving cars. Social studies of science, 2018.
- [12] B. Mahesh. Machine learning algorithms - a review. International Journal of Science and Research, 2020.

- [13] S. Gadri and NEH. Adouane. Efficient traffic signs recognition based on cnn model for self-driving cars. International Conference on Intelligent Computing Optimization, 2022.
- [14] J. Liu, X. Mao, Y. Fang, D. Zhu, and M. Max Q.-H. A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving. In 2021 IEEE International Conference on Robotics and Biomimetics (ROBIO), pages 978–985, 2021.
- [15] S. Uddin, I. Haque, H. Lu, MA. Moni, and E. Gide. Comparative performance analysis of k-nearest neighbour (knn) algorithm and its different variants for disease prediction. Scientific Reports, 12(1) :1–11, 2022.
- [16] DK. Kye, SW. Kim, and SW. Seo. Decision making for automated driving at unsignalized intersection. In 2015 15th International Conference on Control, Automation and Systems (ICCAS), pages 522–525, 2015.
- [17] P. Raktrakulthum and C. Netramai. Vehicle classification in congested traffic based on 3d point cloud using svm and knn. In 2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE), pages 1–6, 2017.
- [18] P. Dayan, M. Sahani, and G. Deback. Unsupervised learning. The MIT Encyclopedia of the Cognitive Sciences., 1999.
- [19] Y. Liu, W. Li, and Y. Li. Network traffic classification using k-means clustering. In Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007), pages 360–365, 2007.
- [20] G. Thomas and R. Sutton. Markov Decision Processes. 2021.
- [21] F. Eckstein. Learning to Fly – Building an Autopilot System based on Neural Networks and Reinforcement Learning. PhD thesis, Hagen Open University, 2020.
- [22] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. AAAI Conference on Artificial Intelligence, 2016.
- [23] C. You, J. Lu, D. Filev, and P. Tsiotras. Highway traffic modeling and decision making for autonomous vehicle using reinforcement learning. In 2018 IEEE Intelligent Vehicles Symposium (IV), pages 1227–1232, 2018.
- [24] X. He, H. Yang, Z. Hu, and C. Lv. Robust lane change decision making for autonomous vehicles : An observation adversarial reinforcement learning approach. IEEE Transactions on Intelligent Vehicles, 8(1) :184–193, 2023.
- [25] R. Tian, S. Li, N. Li, I. Kolmanovsky, A. Girard, and booktitle. Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts. 2018 IEEE Conference on Decision and Control (CDC), pages 321–326, 2018.
- [26] U. Yavas, T. Kumbasar, and N. K. Ure. A new approach for tactical decision making in lane changing : Sample efficient deep q learning with a safety feedback reward. In 2020 IEEE Intelligent Vehicles Symposium (IV), pages 1156–1161, 2020.
- [27] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang. A decision-making strategy for vehicle autonomous braking in emergency via deep reinforcement learning. IEEE Transactions on Vehicular Technology, 69(6) :5876–5888, 2020.
- [28] H. Chen, Y. Zhang, UA. Bhatti, and M. Huang. Safe decision controller for autonomous drivingbased on deep reinforcement learning innondeterministic environment. Sensors, 23(3), 2023.
- [29] N. Motamedidehkordi, S. Amini, S. Hoffmann, F. Busch, and MR. Fitriyanti. Modeling tactical lane-change behavior for automated vehicles : A supervised machine learning

- approach. In 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), pages 268–273, 2017.
- [30] Y. Liu, X. Wang, L. Li, S. Cheng, and Z. Chen. A novel lane change decision-making model of autonomous vehicle based on support vector machine. IEEE Access, 7 :26543–26550, 2019.
- [31] R. Tami, B. Soualmi, A. Doufene, J. Ibanez, and J. Dauwels. Machine learning method to ensure robust decision-making of avs. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pages 1217–1222, 2019.
- [32] D. Isele. Interactive decision making for autonomous vehicles in dense traffic. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pages 3981–3986, 2019.
- [33] M. Samuel. An improved safety algorithm for artificial intelligence enabled processors in self driving cars. Journal of artificial intelligence, 1(02) :95–104, 2019.
- [34] M. Kamrania, A. R. Srinivasanc, S. Chakrabortyc, and AJ. Khattaka. Applying markov decision process to understand driving decisions using basic safety messages data. Transportation Research Part C : Emerging Technologies, 115 :102642, 2020.
- [35] A. Shetty, M. Yu, A. Kurzhanskiy, Offer Grembek, H. Tavafoghi, and P. Varaiya. Safety challenges for autonomous vehicles in the absence of connectivity. Transportation Research Part C : Emerging Technologies, 128 :103133, 2021.
- [36] C. Liu et M. Tomizuka. Enabling safe freeway driving for automated vehicles. In 2016 American Control Conference (ACC), pages 3461–3467, 2016.
- [37] A. Leite, M. Candadai, and E. Izquierdo. Reinforcement learning beyond the bellman equation : Exploring critic objectives using evolution. In Artificial Life Conference Proceedings 32, pages 441–449. MIT Press, 2020.
- [38] T. Njdountche. Électronique numérique 3 : Machines avec un nombre fini d'états, volume 3. ISTE éditions, 2016.
- [39] Latifa Greche, Maha Jazouli, Najia Es-Sbai, Aicha Majda, and Arsalane Zarghili. Comparison between euclidean and manhattan distance measure for facial expressions classification. In 2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), pages 1–4, 2017.
- [40] Visual studio code - code editing. redefined. <https://code.visualstudio.com/>. Consulté le 1/06/2023.
- [41] Project jupyter | home. <https://jupyter.org/>. Consulté le 1/06/2023.
- [42] Python.org. What is python? executive summary. <https://www.python.org/doc/essays/blurb/>. Consulté le 1/06/2023.
- [43] Numpy. <https://numpy.org/>. Consulté le 1/06/2023.
- [44] pandas documentation — pandas 2.0.2 documentation. <https://pandas.pydata.org/docs/>. Consulté le 1/06/2023.
- [45] Pygame front page — pygame v2.5.0 documentation. <https://www.pygame.org/docs/>. Consulté le 1/06/2023.
- [46] scikit-learn : machine learning in python — scikit-learn 1.2.2 documentation. <https://scikit-learn.org/stable/index.html>. Consulté le 1/06/2023.
- [47] Python plotting — matplotlib 3.4.3 documentation. <https://matplotlib.org>. Consulté le 1/06/2023.

- [48] S. Li, K. Shu, C. Chen, and D. Cao. Planning and decision-making for connected autonomous vehicles at road intersections : A review. Chinese Journal of Mechanical Engineering, 34(1) :133–148, 2021.
- [49] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, and G. Ostrovski. Human-level control through deep reinforcement learning. Nature, 518(7540) :529–533, 2015.

## Résumé

Avec les avancées technologiques fulgurantes de ces dernières années, il est fort probable que les véhicules autonomes connaîtront un immense succès dans un futur proche, transformant ainsi la façon dont nous nous déplaçons et interagissons avec nos véhicules. Pour que ces véhicules puissent circuler en toute sécurité, ils doivent être capables de prendre des décisions rapides et efficaces face à des situations complexes. L'objectif principal de ce mémoire est de répondre au problème de prise de décision dans les véhicules autonomes. Pour atteindre cet objectif nous avons proposé une méthode basée sur l'intégration de l'apprentissage par renforcement avec l'algorithme de k le plus proche voisin (KNN). une méthode qui permettra au véhicule d'imiter l'apprentissage humain grâce à l'apprentissage par renforcement et de prendre des décisions sûres grâce à l'algorithme KNN, en utilisant des connaissances apprises lors de son entraînement. Dans le cadre de ce travail nous avons créé un environnement de simulation afin de tester notre méthode et les résultats obtenus ont démontré la faisabilité de notre proposition.

**Mot clés :** Véhicules autonomes, prise de décision, K le plus proche voisin (KNN), apprentissage par renforcement.

## Abstract

With the meteoric technological advances of recent years, it's highly likely that autonomous vehicles will be a huge success in the near future, transforming the way we travel and interact with our vehicles. For these vehicles to travel safely, they need to be able to make quick and effective decisions in complex situations. To achieve this goal, we have proposed a method based on the integration of reinforcement learning with the k-nearest-neighbor (KNN) algorithm, a method that will enable the vehicle to mimic human learning through reinforcement learning and to make safe decisions through the KNN algorithm, using knowledge learned during training. As part of this work, we created a simulation environment to test our method, and the results obtained demonstrated the feasibility of our proposal.

**Key words :** Autonomous vehicles, decision-making, k- nearest neighbor algorithm (KNN), reinforcement learning.