

الجمهورية العربية السورية
المعهد العالي للعلوم التطبيقية والتكنولوجيا

قسم المعلومات

العام الدراسي 2022/2023

المعهد العالي للعلوم التطبيقية والتكنولوجيا

مشروع تخرج

أعد لنيل درجة الإجازة في هندسة المعلومات

القيادة الذاتية لعربة في بيئة افتراضية باستخدام الذكاء

الصنعي

تقديم

أفين حسين

إشراف

د. عمر حمدون

د. عماد عيسى

م. أنس شيخاني

2023/8/24

كلمة شكر

كل الشكر لمن كان عونهم ومتابعتهم هي السبب والدافع في إنجاز هذا المشروع وأخص الدكتور عماد عيسى على متابعته الدائمة وحرصه على إنجاز العمل والدكتور عمر حمدون على تقديمه كافة أشكال الدعم والمساعدة والمهندس أنس شيخاني على إشرافه الدائم ودعمه المتفاني. وأخص أيضاً المهندس طارق ساعاتي على تقديم كافة أشكال النصح والإرشاد.

أفين حسين

الخلاصة

تعد مسألة القيادة الذاتية من أكثر المسائل البرمجية أهمية وتعقيداً. وتشكل المحاكاة وسيلة واعدة في تطوير حلول القيادة الذاتية بأمان وفاعلية. ويهدف هذا المشروع لبناء بيئة افتراضية لمحاكاة عربة ذاتية القيادة. يعرض هذا المشروع مسألة القيادة الذاتية بثلاث شقوق، الأول هو استطلاع البيئة عن طريق الحساسات والمعطيات الصورية لكشف محددات الطرق والأغراض بالاستعانة بخوارزميات الرؤية الحاسوبية. والثاني هو التحكم بالعربة بهدف البقاء على الطريق باستعمال خوارزميات التعلم. وأما الثالث فهو بناء بيئة محاكاة تسمح بتقييم النماذج المدربة وتدريب نماذج جديدة. وتضمنت الدراسة المرجعية أحدث الخوارزميات حتى تاريخ تنجيز هذا المشروع. إذ تم اعتماد خوارزمية CLRNet لكشف المحددات وخوارزمية YOLOv8 لكشف الأغراض لتفوقهما في العديد من المعايير. وأما بالنسبة للتحكم بالعربة فتم اعتماد خوارزمية التعلم المعزز العميق (Deep Q-Network) لجودة نتائجها وملائمتها للبيئة المنجزة. وأما المحاكاة فتم إنجازها ضمن محرك يونيتي Unity.

Abstract

Self-driving cars technology is one of the most important and complex fields of technology, and simulation is a promising way to develop safe and effective self-driving solutions. This project aims to build a virtual environment for simulating a self-driving car. The project presents the issue of self-driving cars in three aspects: first, surveying the environment through sensors and visual data to detect road markers and objects using computer vision algorithms. Second, controlling the car to stay on the road using machine learning algorithms. Third, building a simulation environment that allows for evaluating trained models and training new models. The study includes the latest algorithms up to the date of this project's implementation. CLRNet algorithm was adopted for lane detection, YOLOv8 algorithm for object detection, as they excel in many criteria. As for controlling the car, Deep Q-Network reinforcement learning algorithm was adopted for its quality results and suitability for the implemented environment. The simulation was implemented within the Unity engine.

فهرس المحتويات

i	الخلاصة
ii	فهرس المحتويات
x	مقدمة عامة
1	توصيف المشروع
1	1.1 مقدمة
1	1.2 إشكالية البحث المطروحة
2	1.3 الأهداف
2	1.4 المتطلبات الوظيفية
2	1.5 المتطلبات غير الوظيفية
3	1.6 المخطط الزمني للمشروع
4	دراسة مرجعية
4	2.1 نظم القيادة الآلية
	2.1.1 اصطلاحات شائعة بخصوص القيادة الآلية (Common Terms Related to Autonomous Vehicles)
4	
5	2.1.2 مستويات القيادة الآلية (Levels of autonomy)
6	2.1.3 الحساسات المستخدمة في القيادة الآلية (Sensors used in autonomous vehicles)
6	
	2.2 خوارزميات الرؤية الحاسوبية في القيادة الآلية (Computer Vision for Autonomous vehicles)
8	
8	2.2.1 خوارزميات اكتشاف محددات الطريق (Lane detection algorithms)
9	2.2.1.1 قاعدة البيانات CULane (CuLane Dataset)
10	2.2.1.2 قاعدة البيانات TuSimple (TuSimple Dataset)
10	2.2.1.3 الطرق المعتمدة على التقطيع (Segmentation-based methods)
11	2.2.1.4 الطرق المعتمدة على مرتكز (Anchor-based methods)
11	2.2.1.5 الطرق المعتمدة على معاملات (Parameter-based methods)

2.2.1.6	خوارزمية شبكة تحسين متعددة الطبقات (Cross Layer Refinement Network)	12
2.2.1.7	معايير التقييم المتبعة والمقارنة (Evaluation metrics and comparason)	13
2.2.2	خوارزميات YOLO لاكتشاف الأغراض (YOLO Object detection algorithms)	14
2.3	خوارزميات التعلم المعزز وتطبيقاتها في القيادة الآلية	16
2.3.1	البنية العامة لخوارزميات التعلم المعزز – مسائل اتخاذ القرار المرحلية بنموذج ماركوف Markov	16
2.3.2	شبكات الجودة (Q-learning Networks)	18
2.3.3	شبكات الجودة العميقة (Deep Q Networks)	19
2.3.4	شبكات الجودة المزدوجة (Double Q-learning)	19
2.3.5	شبكات DQN المزدوجة (Double DQN)	20
2.3.6	مقارنة بين شبكات DQN و شبكات DQN المزدوجة (Double DQN)	20
2.3.7	شبكات الجودة العميقة المعتمدة على بيانات إيضاحية (Deep Q-learning from Demonstration)	21
2.4	محركات الألعاب كبيئة للمحاكاة	23
2.4.1	مهيئات استخدام الذكاء الاصطناعي ضمن محركات الألعاب (Artificial Intelligence Foundations)	23
2.4.1.1	استخدامات تعلم الآلة ضمن المحاكاة (ML uses in simulation)	23
2.4.1.2	خوارزمية السرعة التشاركية لتجنب العوائق (Reciprocal Velocity Obstacles ROV)	24
2.4.1.3	خوارزمية السرعة التشاركية الأمثلية لتجنب العوائق (Optimal Reciprocal Collision Avoidance ORCA)	25
2.4.1.4	مشكلة تجنب العوائق في الزوايا (Cornering)	27
2.4.2	محاكاة فيزياء الجسم الصلب والتصادم ضمن محرك الألعاب (Collision and Rigid Body Dynamics)	27
2.4.2.1	واحدات القياس ضمن محرك الفيزياء (Units)	28

2.4.2.2 الفصل بين القيم الخطية والقيم الزاوية (Separability of Linear and Angular	28
..... Dynamics)	
2.4.2.3 الحركة الخطية (Linear Dynamics)	28
2.4.2.4 محاكاة حركة التوجيه (Steering)	30
بيئة العمل والأدوات المستخدمة.....	33
3.1 محرك الألعاب يونيتي Unity	33
3.1.1 منحنيات التحريك (Animation Curves)	33
3.1.2 تجميع الكائنات (Object pooling)	34
3.1.3 الملاحة (Navigation)	34
3.1.4 آلية عملية نظام الملاحة (Inner Workings of the Navigation System)	35
3.2 مكتبة وكلاء تعلم الآلة ML-Agents	36
3.2.1 مكونات المكتبة (Key Components)	37
3.2.2 التعلم المعزز العميق باستخدام ML-Agents (Deep Reinforcement Learning with	
..... ML-Agents)	38
3.2.3 التعلم بالتقليد باستخدام ML-Agents (Imitation Learning with ML-Agents)	39
3.2.4 واجهة بايثون المنخفضة المستوى للتخاطب (Python Low Level API / PLL-API)	40
التطبيق العملي	43
4.1 مقدمة	43
4.2 إنشاء بيئة المحاكاة	43
4.2.1 إنشاء محاكاة العربة	43
4.2.1.1 محاكاة المنظومة الميكانيكية	44
4.2.1.2 محاكاة الحساسات المرتبطة بالعربة	47
4.2.1.3 هيكلية الصفوف البرمجية المرتبطة بالعربة	48
4.2.2 إنشاء محاكاة المدينة	50
4.2.3 إنشاء محاكاة النظام المروري	50
4.2.3.1 هيكلية الصفوف المرتبطة بنظام نقاط العلام	53

4.3	ربط بيئة المحاكاة مع بيئة التدريب من خلال ML-Agents	54
4.3.1	إنشاء البيئة الافتراضية	54
4.3.2	الحساسات المستخدمة (Sensors)	55
4.3.3	توطين بيئة التعلم ضمن المشهد في يونيتي	56
4.3.3.1	منظومة نقاط المراقبة Checkpoints وكشف الاصطدام	56
4.3.3.2	الصفوف المرتبطة بنقاط المراقبة وكشف التصادم	57
4.3.3.3	تهيئة وكيل التعلم	58
4.3.3.4	الصف CarAgent	58
4.4	توطين خوارزميات الرؤية الحاسوبية ضمن بيئة التدريب	59
5.4	التحكم بالعربة باستخدام خوارزميات التعلم المعزز العميق	60
4.5.1	تدريب خوارزمية DQN على دخل من حساس Lidar	62
4.5.2	تدريب خوارزمية DQN على دخل حساس Lidar ومحددات الطرق وفق خوارزمية CLRNet	62
4.5.3	التعلم من بيانات إيضاحية (Demonstration Learning)	64
4.6	إنشاء واجهات المستخدم	65
4.6.1	إنشاء قناة ربط جانبية لنقل معلومات العرض Custom Side Channel	65
4.6.2	استقبال معلومات العرض ضمن يونيتي	66
4.6.3	الصفوف المرتبطة بعرض المعلومات	67
النتائج العملية		69
5.1	أداة Tensorboard	69
5.2	نتائج تطبيق خوارزمية DQN	70
5.2.1	الحاجة للبيانات الإيضاحية	70
5.2.2	مشكلة النسيان (Catastrophic forgetting)	70
5.2.3	النتائج من أجل التدريب على دخل من حساس Lidar	71
5.2.4	النتائج من أجل التدريب على دخل من حساس Lidar ومحددات الطرق وفق CLRNet	73
a	الخاتمة	
b	المراجع	

قائمة الأشكال والرسوم التوضيحية

- الشكل 1 مستويات القيادة الآلية 6
- الشكل 2 الحساسات المختلفة على عربة ذاتية القيادة ومواقعها. 7
- الشكل 3 صور لتحديات مختلفة تواجه اكتشاف محددات الطرق 9
- الشكل 4 حالات التحديات في بيانات CULane 10
- الشكل 5 بنية شبكة CLRNNet 12
- الشكل 6 مقارنة نتائج CLRNNet مع نماذج كشف المحددات 14
- الشكل 7 البنية العامة لخوارزميات MDP 17
- الشكل 8 مقارنة بين شبكة DQN و Double DQN 21
- الشكل 9 التآرجح الناتج عن استقلال الوكلاء عند محاولتهما لتجنب بعضهما 25
- الشكل 10 حل خوارزمية ROV للاصطدام المتوقع في خطوة واحدة بتغيير مسار كل من الوكيلين 25
- الشكل 11 حالة 3 وكلاء في ORCA 26
- الشكل 12 الزوايا في تجنب العوائق 27
- الشكل 13 مركز ثقل الجسم الصلب ضمن جملة إحداثية مرتبطة بالمشهد 29
- الشكل 14 ارتباط أغراض المحاكاة بمكونات الجسم الصلب 30
- الشكل 15 التوجيه القابل للالتفاف 30
- الشكل 16 التوجيه البسيط 31
- الشكل 17 توجيه آكرمان 31
- الشكل 18 منحنيات التحريك 33
- الشكل 19 شبكة ملاحظة بعناصرها المختلفة 35
- الشكل 20 بنية مكونات ML-Agents 38
- الشكل 21 استخدام بيانات إيضاحية في التعلم المعزز 40
- الشكل 22 القوى المؤثرة في العجلة 44
- الشكل 23 قوة التعليق المطبقة على العجلات 45

الشكل 24	منحني التحريك المستخدم في حساب الانزلاق بدلالة السرعة	46
الشكل 25	منحني التحريك المستخدم في حساب الاحتكاك بدلالة السرعة	46
الشكل 26	منحني التحريك المستخدم في حساب معامل التسارع بدلالة السرعة على السرعة العظمى ...	47
الشكل 27	حساس Lidar الافتراضي المرتبط بالعربة	48
الشكل 28	منظور كاميرا العربة وكاميرا المراقبة	48
الشكل 29	مخطط الصفوف الخاص بالصف CarController والمكونات المرتبطة به	49
الشكل 30	مخطط الصفوف الخاص بالصفوف المرتبطة بالعجلة	49
الشكل 31	المدينة ضمن مشهد يونيتي	50
الشكل 32	شبكة الملاحظة ضمن المدينة	51
الشكل 33	واجهة التحكم بنقاط العلام	51
الشكل 34	نقاط العلام ضمن المدينة	52
الشكل 35	المشاة والعربات في النظام المروري	52
الشكل 36	الصفوف المرتبطة بنقاط العلام	53
الشكل 37	الصفوف المرتبطة بتوليد الازدحام	54
الشكل 38	المسافات المقاسة بأشعة حساس Lidar بدلالة الشعاع	55
الشكل 39	البيانات الصورية من يونيتي والمرسلة إلى بايثون	56
الشكل 40	توضع نقاط المراقبة على الخريطة في المدينة	57
الشكل 41	الصفوف المرتبطة بنقاط المراقبة وكشف التصادم	57
الشكل 42	بنية الصف CarAgent	58
الشكل 43	كشف المحددات باستخدام CLNet على البيانات الصورية المستقبلية من يونيتي	60
الشكل 44	كشف الأغراض باستخدام خوارزمية YOLOv8 على البيانات الصورية المستقبلية من يونيتي	60
الشكل 45	بنية DQN المقترحة على دخل من حساس Lidar	63
الشكل 46	بنية DQN المقترحة على دخل من حساس Lidar ومحددات الطرق وفق خوارزمية CLNet	64
الشكل 47	صفوف القنوات الجانبية المخصصة ضمن بايثون	65
الشكل 48	عرض خرج خوارزمية CLNet ضمن يونيتي	66
الشكل 49	عرض خرج خوارزمية YOLOv8 ضمن يونيتي	66
الشكل 50	الصفوف المتعلقة بإستقبال معلومات العرض	67

الشكل 51	بنية الصف Line	68
الشكل 52	صفوف إظهار السرعة في الواجهة	68
الشكل 53	استعراض ومراقبة النتائج أثناء التدريب عبر Tensorboard	69
الشكل 54	مشكلة النسيان	70
الشكل 55	البيئة المصغرة المستخدمة في التدريب	71
الشكل 56	الربح بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar	71
الشكل 57	دالة الخسارة بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar	72
الشكل 58	سرعة العربة بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar	72
الشكل 59	سرعة الانعطاف بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar	72
الشكل 60	الربح بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar ومحددات الطرق وفق	
73	CLRNet	
الشكل 61	دالة الخسارة بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar ومحددات الطرق وفق	
73	CLRNet	
الشكل 62	السرعة بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar ومحددات الطرق وفق	
74	خوارزمية CLRNet	
الشكل 63	سرعة الانعطاف بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar ومحددات الطرق	
74	وفق خوارزمية CLRNet	

قائمة الجداول

الجدول 1	المخطط الزمني للمشروع.....	3
الجدول 2	بارامترات التدريب لدالة الربح على دخل من حساس Lidar.....	62
الجدول 3	بارامترات التدريب لدالة الربح على دخل من حساس Lidar ومحددات الطرق وفق خوارزمية	
الجدول 4	المعاملات الفائقة المستخدمة في شبكة DQN المقترحة.....	64

مقدمة عامة

تلعب القيادة الآلية دوراً أساسياً في حل المشكلات المرورية كالازدحام وحوادث السير ومن المتوقع أن تغير أنظمة التنقل التي نعرفها بشكل جذري. حيث يعوّل على القيادة الآلية في أن تسهل عملية التنقل لمختلف شرائح المجتمع من كبار السن وذوي الاحتياجات الخاصة وغيرهم مما جعلها محط اهتمام بحثي وصناعي كبير. وتحمل العربات الذاتية القيادة عادةً مجموعة من الحساسات لجمع البيانات المكانية الخاصة بالملاحة إضافة لكاميرات تفيد في تحصيل معلومات أكثر دقة عن الأغراض في البيئة المحيطة. وساهم التطور الذي شهده علم الرؤية الحاسوبية والذكاء الصناعي في توفر خيارات ممتازة يمكن من خلالها معالجة البيانات الصورية والاستفادة منها في القيادة الآلية.

إلا أن التجارب على العربات وتدريب النماذج لتحقيق درجة عالية من الأتمتة في القيادة عملية مكلفة زمنياً ومادياً وخطرة على مجريها في حالة عدم إجراء الاختبارات الكافية للتأكد من موثوقية الحلول المطروحة لهذه المسألة.

وتقدم بيئات المحاكاة الافتراضية حلاً للتساؤلات المطروحة بشأن موثوقية القيادة الآلية يمكن المطورين من تدريب واختبار خوارزمياتهم بشكل أمثل. وتفيد أيضاً في إعادة استخدام خوارزميات التعلم العميق المطورة في نظم المحاكاة على مسألة القيادة الآلية ومن أشهر أمثلتها خوارزميات التعلم المعزز العميق بشبكات الجودة والتي تهتم باتخاذ القرارات الأمثلية بناء على حالة البيئة المحيطة والتي شكلت نخضة علمية سابقاً عند تفوقها على الخبراء البشر في حل العديد من المسائل. وبناءً على ذلك وجدنا من المفيد تطوير بيئة افتراضية تستعمل خوارزميات الرؤية الحاسوبية والتعلم العميق في محاكاة القيادة الذاتية. ولتنفيذ المطلوب تم الانطلاق من إجراء دراسة مرجعية حول القيادة الآلية وخوارزميات الرؤية الحاسوبية التي يمكن الاستفادة منها كما تمت دراسة خوارزميات التعلم بهدف اختيار حلول برمجية ملائمة للمسألة، تم التعرف على بيئة تطوير المحاكاة ومكاتب التواصل المتاحة بين خوارزميات التعلم وخوارزميات المحاكاة وصولاً للجزء العملي الذي يشرح خطوات إنجاز العربة وربطها مع بيئة التعلم وتدريب النماذج وأخيراً تمت مناقشة النتائج وطرح الآفاق المستقبلية للمشروع.

الفصل الأول

توصيف المشروع

يتضمن هذا الفصل لمحة عامة عن المشروع وما يحتويه من متطلبات وظيفية وغير وظيفية.

1.1 مقدمة

وفقاً لمنظمة الصحة العالمية (WHO) فإن 1.3 مليون شخصاً يموتون سنوياً في حوادث السير وهي السبب الأساسي لموت الأطفال والشباب من عمر 5-29. إن الأسباب الرئيسية لحوادث السير هي: القيادة المسرعة، القيادة تحت تأثير الكحول، القيادة المتهورة والطرق الغير آمنة. تدعم السيارات ذاتية القيادة مهاماً كثيرة كمرقبة البيئة الخارجية وتحديد أقصر الطرق والتحكم والملاحة دون الحاجة للتدخل البشري مما يقلل احتمالية حدوث الأخطاء بالإضافة لتقديم العديد من الفوائد مثل تقليل مخاطر حوادث السير وتقليل استهلاك الوقود والازدحام المروري وتساعد في توفير الوقت والمساحة وتسهيل تنقل كبيرى السن وذوي الاحتياجات الخاصة [1].

1.2 إشكالية البحث المطروحة

في ظل الاهتمام المتزايد بالسيارات ذاتية القيادة التي ستشكل مستقبل العربات، يمكن الاستفادة من برمجيات المحاكاة في إيجاد واختبار الخوارزميات والبرمجيات التي تسمح لتلك السيارات بالعمل بشكل أمثل، سهل الاستخدام وعالي الأمان دون الحاجة للتكاليف المادية والبشرية الناتجة عن تجارب القيادة على السيارات الحقيقية.

سنعمل في إطار هذا المشروع لإنشاء بيئة افتراضية تحوي على عربة ذاتية القيادة بالاعتماد على كاميرا وحساسات مختلفة باستخدام خوارزميات الرؤية الحاسوبية والذكاء الصناعي.

1.3 الأهداف

يهدف هذا المشروع إلى:

- التعرف على بيئة العمل المستخدمة Unity والمكاتب المتوفرة ضمنها للربط مع خوارزميات تعلم الآلة.
- إنشاء بيئة محاكاة تتسم بالواقعية وتتضمن العربة والمدينة والنظام المروري من سيارات ومشاة.
- ربط بيئة المحاكاة مع بيئة التعلم لتدريب النماذج.
- توطين خوارزميات الرؤية الحاسوبية على البيانات الصورية من المحاكاة ضمن بيئة التعلم.
- تدريب نماذج القيادة الذاتية بالاعتماد على خوارزميات التعلم.

1.4 المتطلبات الوظيفية

يجب أن يحقق المشروع المتطلبات التالية:

- بناء البيئة الافتراضية التي سيتم فيها تسيير العربة مع الشوارع، إشارات المرور، تقاطعات، ممرات مشاة ومشاة وسيارات أخرى.
- بناء محاكاة واقعية لسيارة مزودة بالحساسات المناسبة وبكاميرا افتراضية تلتقط صور الطريق.
- استخدام خوارزميات الرؤية الحاسوبية لكشف الأغراض ومحددات الطرق ضمن المشهد.
- تدريب نماذج باستخدام خوارزميات التعلم قادرة على الاستفادة من دخل الحساسات وخوارزميات الرؤية الحاسوبية في تحريك وقيادة العربة.

1.5 المتطلبات غير الوظيفية

- إنشاء واجهة مستخدم سهلة الاستخدام.
- الواقعية في محاكاة بيئة التدريب.
- قابلية التوسع كالقدرة على تدريب نماذج جديدة واختبارها باستخدام البرمجية المنجزة.

1.6 المخطط الزمني للمشروع

تم تقسيم العمل على الخطة الزمنية التالية:

الجدول 1 المخطط الزمني للمشروع

المهمة	التاريخ
الدراسة النظرية وتجهيز بيئة العمل والأدوات البرمجية والتآلف معها.	[20/2/2023 – 7/3/2023]
بناء مجسم مدينة افتراضية أولية مزودة بشوارع، إشارات مرور، تقاطعات، ساحات، ممرات مشاة ومشاة.	[8/3/2023 – 14/3/2023]
بناء مجسم سيارة وتنفيذ محاكاة أولية تتضمن ميكانيك وحركة السيارة ضمن المدينة من دخل المستخدم بشكل أولي.	[15/3/2023 – 21/3/2023]
تزويد العربة بكاميرا افتراضية تلتقط صور الطريق لمعالجتها واستخلاص المعلومات المفيدة منها في القيادة.	[22/3/2023 – 23/3/2023]
إنشاء محاكاة ضمن بيئة Unity لحساسات ملائمة مثل Lidar 3D.	[24/3/2023 – 1/4/2023]
توطين خوارزميات الرؤية الحاسوبية لكشف العربات، المارة، إشارات المرور والتقاطعات.	[2/4/2023 – 2/5/2023]
تحضير دخل منظومة العمل من حصيلة خرج الحساسات وصور الكاميرا الملتقطة.	[9/5/2023 – 11/5/2023]
كتابة الجزء الخاص بالدراسة النظرية وتجهيز الدخل ضمن التقرير.	[12/5/2023 – 27/5/2023]
تنفيذ محاكاة تتضمن قيادة السيارة ضمن المدينة من ناتج خوارزميات التعلم.	[28/5/2023 – 15/7/2023]
كتابة القسم المتعلق بتحريك العربة ضمن البيئة الافتراضية من خرج خوارزميات التعلم ضمن التقرير.	[16/7/2023 – 20/7/2023]
إنشاء واجهات المستخدم.	[21/7/2023 – 10/8/2023]

الفصل الثاني

دراسة مرجعية

يتضمن هذا الفصل تعريفاً بالمفاهيم النظرية التي تم استخدامها في إنجاز المشروع.

2.1 نظم القيادة الآلية

لقد كانت دراسة السيارات ذاتية القيادة محط اهتمام الأبحاث بهدف زيادة وثوقية وأمان التنقل. وكانت حصيلة هذه الدراسات إنتاج العديد من أجهزة مساعدة السائق التجارية كمساعد الرحلات البحرية (Adaptive Cruise Control) وجهاز المكابح الإضطرارية الذاتية (Autonomous Emergency Braking) وأجهزة البقاء ضمن حدود الطريق (Lane Keeping Systems). إلا أنه وبالرغم من وجود العديد من الأجهزة المتطورة لمساعدة السائق إلا أن مسألة القيادة الذاتية تعد أكثر تعقيداً لاتساع فضاء القرار المرتبط بها [2].

سيتم في الفقرات التالية تلخيص أهم المفاهيم المتعلقة بالقيادة الذاتية والتي سيتم التطرق لها في هذا المشروع ابتداءً بتعريف المصطلحات والاختصارات الخاصة بالقيادة الآلية ووصولاً إلى التعريف بالحساسات المستخدمة مرجعياً في هذه المسألة.

2.1.1 اصطلاحات شائعة بخصوص القيادة الآلية (Common Terms Related to)

Autonomous Vehicles)

إن القيادة الديناميكية أو ما يدعى (Dynamic Driving Task DDT) هي مجموعة العمليات التي يتطلبها التحكم بعربة ذاتية القيادة، بما يتضمن التحديات البسيطة ولا يتضمن الخطوات التفصيلية لحلّها كتحديد الطريق الذي ستسير عليه العربة والتخطيط للرحلة. وفي هذا الفرع من الدراسات يتم تحديد التحكم العرضاني (steering) والطولاني (acceleration) للعربة ومراقبة البيئة المحيطة ووضع خطط المناورة.

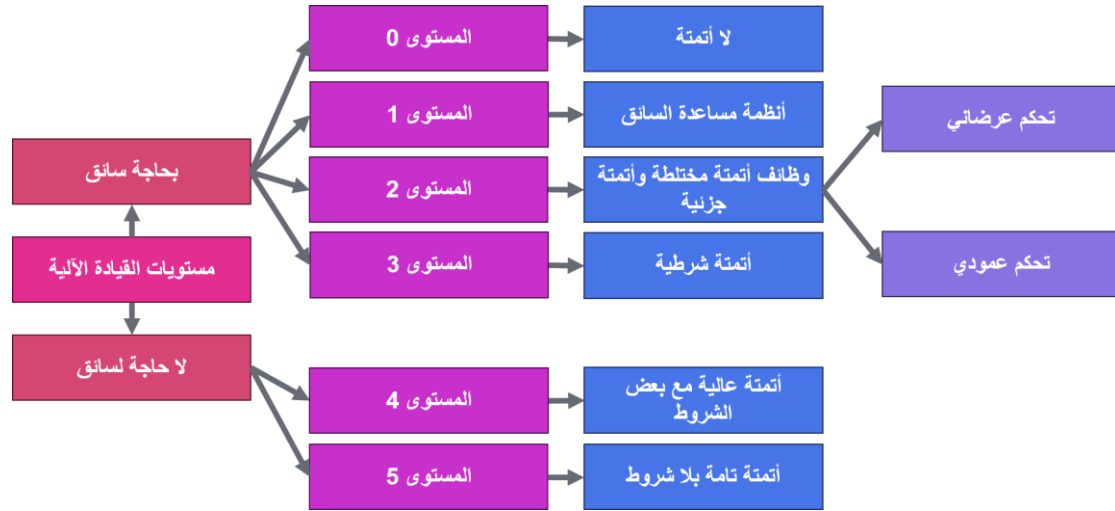
أما مصطلح منظومة القيادة الآلية (Automated Driving System ADS) فتم إطلاقه على البرمجيات والعتاد المستخدم في إجراء القيادة الديناميكية بشكل مستمر سواء أكانت مرتبطة بنوع مجال تصميم تشغيلي (Operational Design Domain ODD) محدد أم لم تكن. وأما مجال التصميم التشغيلي ODD فهو مصطلح يعبر عن ظروف الأنظمة المؤتمتة ضمن بيئة معينة في منطقة جغرافية ما وبوجود ازدحام أو عدم وجوده وسواء أكانت القيادة ضمن بيئة مدنية أم ضمن طرق وعرة وتقوم هذه الظروف بتحديد الشروط التي ينبغي على النظام العمل ضمنها.

وأما التعرف على الأغراض والاستجابة للأحداث (Object and Event Detection and Response OEDR) فهي مهمة جزئية من DDT وتتضمن مراقبة البيئة كالكشف وتصنيف الأغراض وتأمين الاستجابة المناسبة للأحداث، تعمل OEDR ضمن ODD وتتيح لنظام ADS أن يميز بين المركبات المختلفة على الطريق والمارة والأغراض وأن يستجيب لأي حدث من شأنه أن يؤثر على أمان القيادة. وأخيراً فإن أنظمة الأمان الفعالة (Active Safety Systems) تقوم بمراقبة الظروف داخل وخارج العربة ذاتية القيادة وتحدد مواضع الخطر المحتملة والموجودة وذلك يشمل الأخطار على المارة والمسافرين والمركبات وتقوم بالتدخل الآلي لتقليل احتمالية حدوث اصطدامات وتقوم بإعلام السائق أو تعديل نظام القيادة أو من خلال التدخل بالأنظمة الجزئية الخاصة بالحركة العمودية (Throttle) والأفقية (Steer) أو بقوى (Suspension) وغيرها [1].

2.1.2 مستويات القيادة الآلية (Levels of autonomy)

عرّف مجتمع مهندسي القيادة الآلية (Society of Automobile Engineers SAE) ستة مستويات من الأتمتة. المستوى صفر للأتمتة هو عدم وجود أي أتمتة. في المستوى صفر من الأتمتة يقوم السائق بعمل كامل عملية DDT بمساعدة بعض أنظمة الأمان الفاعلة. أما في المستوى الأول فيتم الاستفادة من ADS يقوم بتنفيذ بعض التحكم العمودي أو العرضاني لحركة السيارة ولكن لا يتم تنفيذ كامل OEDR. يقوم السائق في هذه الحالة بالتواجد بشكل دائم في العربة ويسمى (DDT fallback) أي أنه المسؤول عن اتخاذ القرارات في حالة فشل DDT في أي من وظائفها. وتقع جميع الوظائف الخاصة بنظم الملاحة الآلية ضمن المستوى الأول من الأتمتة. وأما أنظمة مكابح الطوارئ الآلية فتقع ضمن المستوى الثاني من الأتمتة. حيث يتم ضمن المستوى الثاني من الأتمتة تنفيذ التحكم العمودي والعرضاني من قبل ADS بشكل شبيه بما ذكر في المستوى الأول وأيضاً بدون وجود OEDR كامل وأيضاً بتواجد السائق بشكل دائم للتدخل عند الحاجة. وأما المستوى الثالث فيتم ضمنه تنفيذ DDT بشكل كامل من قبل ADS ومع تواجد السائق حيث يتم إعلامه عندما يوشك النظام على تجاوز حدود ODD وأيضاً عندما يحدث أي خلل في النظام. وأما المستوى الرابع فيتم تنفيذ

كامل DDT ضمن ODD مقيدة من قبل ADS ولا يتطلب أي تدخل بشري حيث يمكن للسائق التدخل في حال الرغبة إلا أن ADS يبقى قادراً على أداء المهمة كاملة دون أي تدخل. وأما في المستوى الخامس والأخير فتتم قيادة ذاتية مؤتمتة بشكل كامل دون أي معوقات ولا يتدخل السائق بأي من العمليات إلا بطلب منه [1]. ويوضح الشكل 1 أدناه مستويات القيادة الذاتية.



الشكل 1 مستويات القيادة الآلية

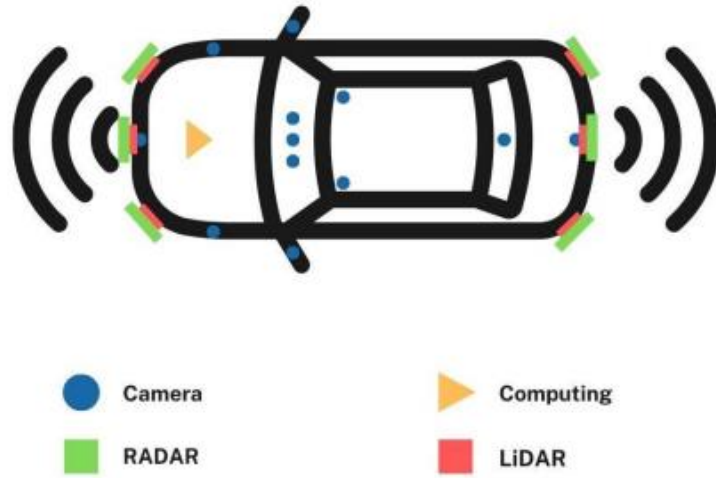
2.1.3 الحساسات المستخدمة في القيادة الآلية (Sensors used in autonomous vehicles)

تعد الكاميرا أهم الحساسات المستخدمة في العربات ذاتية القيادة [1]. حيث تستطيع العربات ذاتية القيادة استطلاع المحيط حولها من كافة الزوايا بواسطة الكاميرات المرتبطة بها. وهذه الحساسات من أفضل الخيارات في التعرف على الأغراض والمحيط. حيث يتم إرسال صور الكاميرا لمنظومة مختصة بخوارزميات ذكاء صناعي تقوم بتحليلها. يوجد أيضاً كاميرات للأشعة تحت الحمراء والتي تستخدم لتحسين الأداء في الظروف التي تصعب فيها الرؤية للكاميرا العادية.

حساسات RADAR المستخدمة أيضاً هي حساسات تستخدم أمواج راديوية لحساب معاملات كالمسافة والسرعة والزوايا. تقوم العربات ذاتية القيادة من خلال هذه الحساسات بإرسال أمواج راديوية وإستقبال إنعكاساتها عن السطوح من خلال مستقبلات الحساس. ومن ميزات هذه الحساسات أداؤها العالي في مختلف ظروف الطقس ولمسافات طويلة ولكنها معرضة للتعرف الخاطئ على الأغراض.

حساسات Lidar هي حساسات تستخدم الضوء (الليزر) كأداة لقياس المسافات وذلك عن طريق قياس الزمن المستغرق لرجوع الضوء للحساس. حيث تقوم هذه الحساسات بإرسال أشعة ليزرية تصطدم بأغراض البيئة

وتنعكس إلى مستقبل صوري. يتم تجميع الأشعة المستقبلية ومقاربتها كسحابة من النقط. ويتم من خلال المستقبلات تجميع الصورة ثلاثية الأبعاد للبيئة. ويعد هذا الحساس أفضل الحساسات وأعلىها سعراً. من الحساسات المستخدمة أيضاً حساس نظام الملاحة العالمي عبر الأقمار الصناعية (Global Navigation Satellite System GNSS) والذي يستقبل إشارات الأقمار الاصطناعية التي توفر معلومات مفيدة عن الموقع الجغرافي وحالة الازدحام والموقع العام. تستخدم حساسات GNSS قواعد المثلثات في تحديد مواقع المستقبلات والمسافات البينية على الأرض. إلا أن الإشارات التي تصل للأقمار الاصطناعية ضعيفة ومعرضة للضجيج وبشكل خاص الناتج عن التشويش. من الحساسات المستخدمة أيضاً حساس Ultrasonic وهو حساس قصير المدى يستخدم في ركن السيارات وهي حساسات رخيصة جداً. لحساسات ألتراسونيك مدى قصير ولكنها ممتازة في السرعات العالية وذلك لاعتمادها أمواج فوق صوتية (تردد عالٍ) في قياس المسافة [3]. أخيراً فإن حساس نظام تحديد المواقع العالمي (Global Positioning System GPS) يستخدم في تأمين معلومات الموقع من الأقمار الاصطناعية، وهي حساسات ذات كلفة منخفضة ومتوفرة للعموم ومفيدة جداً في عمليات تخطيط المسار ويوضح الشكل 2 أدناه على مواقع الحساسات المختلفة على العربة [1].



الشكل 2 الحساسات المختلفة على عربة ذاتية القيادة ومواقعها.

وتبين الدراسات أن حساس Lidar والكاميرا هي أهم الحساسات المعتمدة في القيادة الآلية وذلك سبب اختيارها في هذا المشروع.

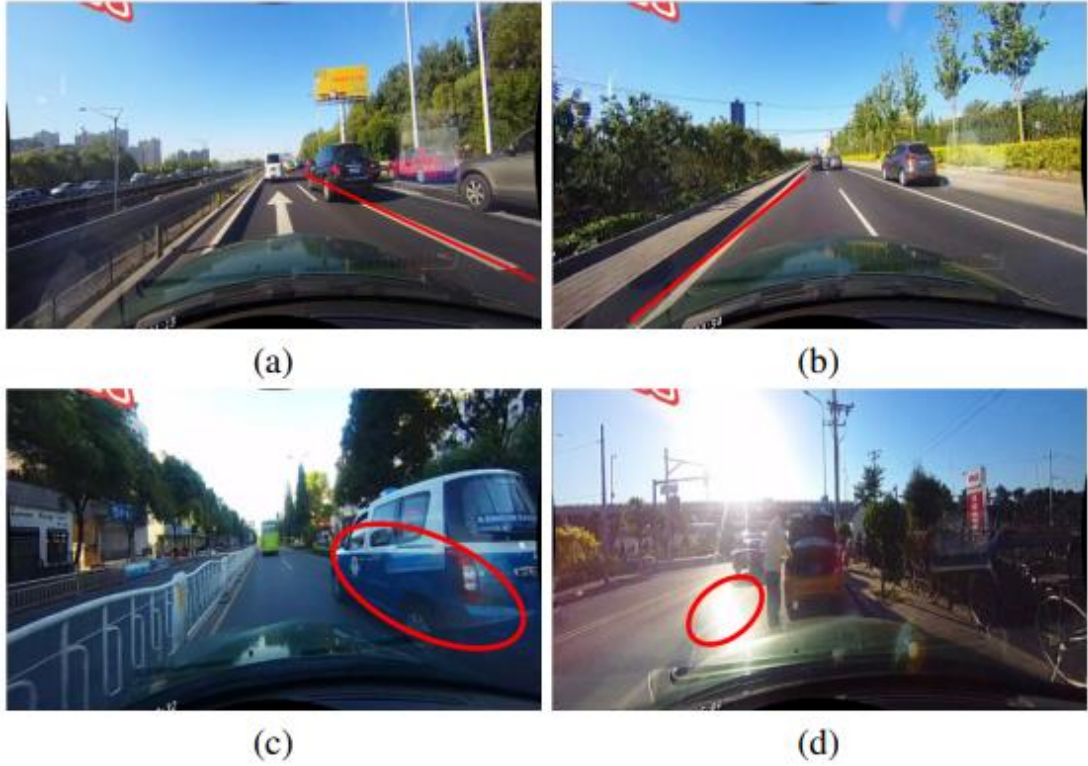
2.2 خوارزميات الرؤية الحاسوبية في القيادة الآلية (Computer Vision for Autonomous vehicles)

تعد الرؤية الحاسوبية من أهم مجالات الذكاء الصناعي. وتستفيد القيادة الآلية من العديد من خوارزميات الرؤية الحاسوبية في أنظمة القيادة الآلية وأنظمة مساعدة السائق (Advanced Driver Assistant Systems ADAS) كخوارزميات التعرف على الأغراض والمشاة وعلامات المرور ومحددات الطريق وسيتم مناقشة بعضها في الفقرات التالية.

2.2.1 خوارزميات اكتشاف محددات الطريق (Lane detection algorithms)

تم تصميم محددات الطريق لمساعدة السائقين أثناء قيادة عرباتهم وخاصة في الظروف الصعبة (ليل، مطر، ضباب)، ويتم الاستفادة منها كمحددات لأنظمة الملاحة الخاصة بالعربات الذكية. حيث أنها تعد من السمات عالية المستوى الدلالي، إلا أنها بحاجة لسمات مفصلة من المستوى الدلالي المنخفض لكي يتم تحديدها بدقة (والسمات ذات المستوى الدلالي المنخفض هي تلك كالحواف والزوايا أما ذات المستوى العالي فمن أمثلتها العناصر والأغراض ضمن المشهد) [4]، وتمكنت العديد من النماذج من الحصول على نتائج مباشرة في اكتشاف محددات الطريق بالاعتماد على شبكة عصبونية تلافيفية (Convolutional Neural Network CNN).

إلا أن العديد من التحديات لاتزال تواجه هذا النوع من النماذج في تحديد المحددات بدقة ولا تزال القدرة على معالجة مختلف مستويات السمات تحدياً تواجهه خوارزمية CNN. ويوضح الشكل 3 أدناه بعض الحالات التي يصعب على CNN حلها وذلك لأنه في الجزء (c) نلاحظ أن محدد الطريق والعلامة المشار إليها يشتركان بالعديد من الخواص إلا أنهما يختلفان في المعنى الدلالي (كلاهما خط أبيض إلا أنهما يحملان معنى مختلف) ومن الصعب التمييز بين السمات عالية المستوى الدلالي. و أما في الجزء (b) الذي يوضح اكتشاف المحددات باستخدام سمات عالية المستوى ولكنه من الواضح أنه وبرغم اكتشاف المحدد إلا أن الموضع الخاص به غير دقيق [4].



الشكل 3 صور لتحديات مختلفة تواجه اكتشاف محددات الطرق

من المشاكل الأخرى التي تواجه اكتشاف المحددات هي حالات عدم وجود دليل بصري على وجود المحدد كما هو موضح في الجزء (d) من الصورة وذلك بسبب وجود إضاءة قوية منعكسة عن الطريق. وتعتمد أحدث النماذج المعتمدة في كشف المحددات على شبكة عصبونية تلافيفية (Convolutional Neural Network CNN) وتصنف ضمن ثلاثة مجموعات سيتم مناقشتها وهي: الطرق المعتمدة على التقطيع (Segmentation-based methods) والطرائق المعتمدة على مرتكز (Anchor-based methods) والطرائق المعتمدة على المعاملات (Parameter-based methods) [4]. وأخيراً سيتم مناقشة خوارزمية شبكة التحسين متعددة الطبقات (Cross Layer Refinement Network CLRNet) والتي تقدم نتائج محسنة عن الطرق السابقة. ولكن بداية سيتم مناقشة أشهر قواعد البيانات المعتمدة في تدريب النماذج على كشف المحددات وهما قاعدتا CULane و TuSimple بالترتيب.

2.2.1.1 قاعدة البيانات CULane (CuLane Dataset)

قاعدة البيانات CULane هي قاعدة بيانات بحثية تحمل بيانات عالية المستوى لأنماط تحديات مختلفة لمسألة كشف المحددات وتستخدم في دراسات الازدحام وكشف محددات الطرق. تم تجميع هذه البيانات عن

طريق كاميرات موضوعة على ستة عربات مقادة من قبل سائقين مختلفين في منطقة بيجين في الصين. وتم تجميع أكثر من 55 ساعة من ملفات الفيديو في 133235 إطاراً. تقسم عناصر البيانات على شكل 88880 مجموعة تدريب و 9675 للتحقق (validation) و 34680 للاختبار (testing). وتقسم مجموعة الاختبار لثمانية أنواع من التحديات الموضحة في الشكل 4 أدناه وهي الإضاءة الخافتة والازدحام وعدم ظهور الخط في الصورة والظلال والأسهم ووجود إضاءة قوية أو انحناء أو في حالات التقاطعات [5].



الشكل 4 حالات التحديات في بيانات CULane

2.2.1.2 قاعدة البيانات TuSimple (TuSimple Dataset)

تتكون قاعدة البيانات TuSimple من 6408 صورة على طريق سريع في الولايات المتحدة ودقة الصور فيها 1280×720 وتتكون من 3626 للتدريب و 358 للتحقق و 2782 للاختبار و تتكون صور الاختبار من صور بظروف جوية متنوعة. وسيتم التحدث في الفقرات التالية عن الخوارزميات التي تعتمد قواعد البيانات هذه في التدريب على اكتشاف محددات الطرق [6].

2.2.1.3 الطرق المعتمدة على التقطيع (Segmentation-based methods)

تقوم الخوارزميات الحديثة بعملية موائمة لكل بكسل على حدى حيث يتم اعتبار مسألة إيجاد المحددات كمسألة تقطيع دلالي (Semantic Segmentation)، ومسائل التقطيع الدلالي هي مسائل تعنى بتصنيف كل بكسل في الصورة على حدى وإعطائه معرف (Label) خاص بدلالته. تقترح خوارزمية الشبكة العصبونية التلافيفية المنظمة (Structured Convolutional Network SCNN) آلية منظومة تراسل لحل مشكلة عدم وجود دليل بصري واضح على وجود محددات الطرق (كحالة إضاءة ساطعة أو خافتة تخفي المحدد)، ومزايا هذه المعالجة عن CNN التقليدية هو أنها لاتعامل الصورة كبنية واحدة بل تعامل كل بكسل على أنه عقدة ضمن بيان ويتم التراسل بين العقد المتجاورة والتي تعبر عن الجوار المحيط بالبكسل وهذا يفيد في إغناء المعلومات المكانية على اعتبار أن البكسلات المجاورة لبكسل يقع على المحدد أكثر احتمالاً بأن تقع على

المحدد. حيث تقدم SCNN تحسناً جيداً على عملية اكتشاف المحددات إلا أنها خوارزمية بطيئة وغير مستحسنة من أجل تطبيقات الزمن الحقيقي [7].

من جهة أخرى، تقترح خوارزمية تجميع تغير السمات المتكررة (Recurrent Feature Shift Aggregator RESA) حلاً لمشكلة التعامل مع تطبيقات الزمن الحقيقي من خلال بنية تجميعية تسمح للشبكة بجمع معلومات الواصفات العامة مع تحسين الأداء [8].

تم استخدام بنية بحث عصبونية (Neural Architecture Search (NAS في خوارزمية CurveLane-NAS لتحسين أداء الخوارزمية في التعامل مع المحددات المنحنية، إلا أن سلبية هذه الخوارزمية تكمن في أنّ عملية NAS مكلفة حسابياً وتتطلب وقتاً طويلاً [9].

وتعد جميع خوارزميات التقطيع مكلفة زمنياً وحسابياً لاعتمادها على طرق تتعامل مع كل بكسل على حدى ولا تنظر لمحددات الطرق كبنية موحدة.

2.2.1.4 الطرق المعتمدة على مرتكز (Anchor-based methods)

تقسم الطرق المعتمدة على مرتكز إلى صنفين هما الطرق المرتكزة على الخطوط والمرتكزة على الصفوف. تعتمد الطرائق المرتكزة على الخطوط خطوطاً محددة مسبقاً كمرجع يتم موائمة النموذج انطلاقاً منه. ومن النماذج المعتمدة على هذه الطريقة Line=CNN و LaneATT [10] والتي تعتمد طريقة مبتكرة لاعتماد المرتكز كمحور اهتمام في الصورة وتجميع البيانات العامة من خلال ذلك (anchor-based attention mechanism) ويتم من خلال ذلك تحقيق نتائج ممتازة في السرعة والأداء.

وتعد خوارزمية CondLaneNet من الطرائق المرتكزة على الصفوف التي تقوم بتعريف اكتشاف محدّدات شرطي يعتمد على مبدأ تحديد نقاط البداية للمحدد ومن ثم القيام بكشف مرتكز على الصفوف [11]. وتعد نقطة الضعف الكبرى لهذا النموذج هي صعوبة تحديد نقاط البداية للمحدد في بعض الحالات المعقدة مما يؤدي لخفض جودة هذه الخوارزمية في بعض الحالات.

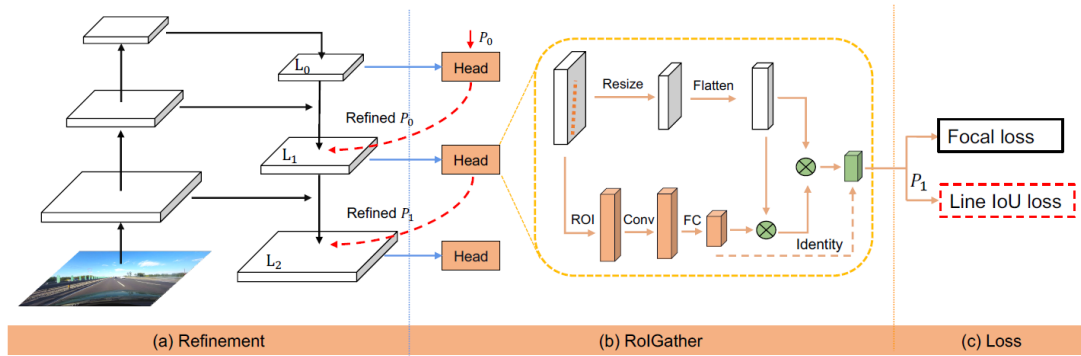
2.2.1.5 الطرق المعتمدة على معاملات (Parameter-based methods)

تقوم هذه الطرق على مبدأ تحديد مجموعة من المعاملات الواصفة للمحدد والقيام بعملية تدريب لتحديد هذه المعاملات ومن أمثلتها خوارزمية (Long-term Semantic Relocalization LSTR) التي تقوم بتوصيف بنية الطريق مع أخذ زاوية الكاميرا بعين الاعتبار وتقوم بتجميع البيانات العامة من خلال محول منفصل [12].

ولكن أحد مشاكل هذه النماذج هو أن وجود خطأ في المعاملات ذات الرتبة الأعلى يؤدي إلى تغيير كبير في شكل المحدد المكتشف وبرغم سرعة هذه النماذج فإنها تواجه صعوبة في تقديم أداء جيد.

2.2.1.6 خوارزمية شبكة تحسين متعددة الطبقات (Cross Layer Refinement Network CLRNet)

تم تطوير خوارزمية CLRNet عام 2022 وتقوم هذه الخوارزمية بالاستفادة بشكل تام من جميع السمات عالية المستوى والمنخفضة. وبشكل أدق فإنه يتم اكتشاف المحددات باستخدام سمات عالية المستوى ومن ثم يتم تحسين النتائج باستخدام سمات دقيقة للحصول على مواقع المحددات بدقة ولهذا ترجع تسمية هذه الخوارزمية بخوارزمية التحسين [4] يتم تحسين النتائج بالتدريج للوصول لدرجة عالية من الدقة. وأما بخصوص مشكلة عدم وجود دليل بصري كحالة الإضاءة الساطعة فيتم حلها باستخدام تقنية تجميع مناطق الاهتمام في الصورة (Region Of Interest Gather- ROIgather) وهي تقنية مقترحة لتجميع البيانات العامة وبناء علاقة بين مناطق الاهتمام ROI الخاصة بالمحدد وبقية المعلومات والتي تأخذ السياق بعين الاعتبار في تحديد موقع المحدد في حال غياب الدليل البصري على وجوده. ويتم اقتراح خاصية أخرى وهي مقدار خسارة Line IoU loss المسؤولة عن تحسين الأداء. وتم اختبار الخوارزمية المقترحة على كل من قواعد بيانات CULane, Tusimple, LLAMAS والحصول على أفضل النتائج من بين جميع الحلول المقترحة على تلك القواعد. ويوضح الشكل 5 أدناه بنية الشبكة المقترحة [4] حيث نلاحظ أن الشبكة تقوم بإنشاء مخططات السمات بالاعتماد على شبكة هرمية للسمات (Feature Pyramid Network FPN) التي تنتج مستويات مختلفة من السمات.



الشكل 5 بنية شبكة CLRNet

ويتم تحسين كل محدّد بالاعتماد على سمات ذات مستوى أعلى ويتم الاعتماد على المزيد من البيانات السياقية ضمن heads لتميز السمات الخاصة بالمحددات مع إضافة قيمة Line IoU loss لتابع الخسارة لزيادة الدقة في

النتائج. وعلى خلاف الطرائق التقليدية في كشف المحددات والتي تعتمد على تمثيل البيانات المكتشفة بصندوق يغطي المحدد فإن هذه الخوارزمية اعتمدت بنية جديدة تدعى Lane Prior يتم من خلالها تمثيل المحدد بنقاط تفصل بينها مسافات ثابتة قدرها $H/(N-1)$ حيث H طول الصورة و N عدد النقاط.

2.2.1.7 معايير التقييم المتبعة والمقارنة (Evaluation metrics and comparason)

تم تقييم هذه الخوارزميات بالاعتماد على مجموعة معايير أولها F1 score المعطى بالعلاقة

$$F1 = \frac{2 \times Percision \times Recall}{Percision + Recall}$$

ومعيار COCO الذي يعتبر معياراً أدق من المعيار السابق لمقدار دقة تحديد المحدد ويعطى بالعلاقة

$$mF1 = (F1@50 + F1@55 + \dots + F1@95)/10$$

حيث $F1@50 + F1@55 + \dots + F1@95$ هي المعايير المستخدمة عند IoU بمقدار 0.5, 0.55, ..., 0.95 بالترتيب.

وتعتمد قاعدة بيانات Tusimple معياراً يعطى بالعلاقة

$$Accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}}$$

حيث C_{clip} هو عدد النقاط الصحيحة المكتشفة من المحدد و S_{clip} هي عدد نقاط المحدد الكلية المفترضة في الصورة. وتعتبر النتيجة صحيحة في حال اكتشاف 85% أو أكثر من النقاط والتي لا تبعد أكثر من 20 بكسل عن النقاط الصحيحة المفروضة. وتم اعتماد معايير أخرى مثل false negative , false positive لتقييم النموذج ويوضح الجدول في الشكل 6 أدناه نتائج الخوارزميات المختلفة بناءً على هذه المعايير [4].

Method	Backbone	mF1	F1@50	F1@75	FPS	GFlops	Normal	Crowded	Dazzle	Shadow	No line	Arrow	Curve	Cross	Night
SCNN [17]	VGG16	38.84	71.60	39.84	7.5	328.4	90.60	69.70	58.50	66.90	43.40	84.10	64.40	1990	66.10
RESA [33]	ResNet34	-	74.50	-	45.5	41.0	91.90	72.40	66.50	72.00	46.30	88.10	68.60	1896	69.80
RESA [33]	ResNet50	47.86	75.30	53.39	35.7	43.0	92.10	73.10	69.20	72.80	47.70	88.30	70.30	1503	69.90
FastDraw [18]	ResNet50	-	-	-	90.3	-	85.90	63.60	57.00	69.90	40.60	79.40	65.20	7013	57.80
E2E [29]	ERFNet	-	74.00	-	-	-	91.00	73.10	64.50	74.10	46.60	85.80	71.90	2022	67.90
UFLD [19]	ResNet18	38.94	68.40	40.01	282	8.4	87.70	66.00	58.40	62.80	40.20	81.00	57.90	1743	62.10
UFLD [19]	ResNet34	-	72.30	-	170	16.9	90.70	70.20	59.50	69.30	44.40	85.70	69.50	2037	66.70
PINet [7]	Hourglass	46.81	74.40	51.33	25	-	90.30	72.30	66.30	68.40	49.80	83.70	65.20	1427	67.70
LaneATT [24]	ResNet18	47.35	75.13	51.29	153	9.3	91.17	72.71	65.82	68.03	49.13	87.82	63.75	1020	68.58
LaneATT [24]	ResNet34	49.57	76.68	54.34	129	18.0	92.14	75.03	66.47	78.15	49.39	88.38	67.72	1330	70.72
LaneATT [24]	ResNet122	51.48	77.02	57.50	20	70.5	91.74	76.16	69.47	76.31	50.46	86.29	64.05	1264	70.81
LaneAF [1]	ERFNet	48.60	75.63	54.53	24	22.2	91.10	73.32	69.71	75.81	50.62	86.86	65.02	1844	70.90
LaneAF [1]	DLA34	50.42	77.41	56.79	20	23.6	91.80	75.61	71.78	79.12	51.38	86.88	72.70	1360	73.03
SGNet [22]	ResNet18	-	76.12	-	117	-	91.42	74.05	66.89	72.17	50.16	87.13	67.02	1164	70.67
SGNet [22]	ResNet34	-	77.27	-	92	-	92.07	75.41	67.75	74.31	50.90	87.97	69.65	1373	72.69
FOLOLane [20]	ERFNet	-	78.80	-	40	-	92.70	77.80	75.20	79.30	52.10	89.00	69.40	1569	74.50
CondLane [12]	ResNet18	51.84	78.14	57.42	173	10.2	92.87	75.79	70.72	80.01	52.39	89.37	72.40	1364	73.23
CondLane [12]	ResNet34	53.11	78.74	59.39	128	19.6	93.38	77.14	71.17	79.93	51.85	89.89	73.88	1387	73.92
CondLane [12]	ResNet101	54.83	79.48	61.23	47	44.8	93.47	77.44	70.93	80.91	54.13	90.16	75.21	1201	74.80
CLRNet (ours)	ResNet18	55.23	79.58	62.21	119/206*	11.9	93.30	78.33	73.71	79.66	53.14	90.25	71.56	1321	75.11
CLRNet (ours)	ResNet34	55.14	79.73	62.11	103/156*	21.5	93.49	78.06	74.57	79.92	54.01	90.59	72.77	1216	75.02
CLRNet (ours)	ResNet101	55.55	80.13	62.96	46/74*	42.9	93.85	78.78	72.49	82.33	54.50	89.79	75.57	1262	75.51
CLRNet (ours)	DLA34	55.64	80.47	62.78	94/151*	18.5	93.73	79.59	75.30	82.51	54.58	90.62	74.13	1155	75.37

الشكل 6 مقارنة نتائج CLRNet مع نماذج كشف المحددات

ويوضح الجدول تفوق خوارزمية CLRNet على جميع الخوارزميات الأخرى في مختلف المعايير وعلى مختلف قواعد البيانات حيث تعد أفضل وأحدث النماذج المتوفرة حتى تاريخ تنجز هذا المشروع (State of the art SOTA) ويعد هذا سبب اختيارها في إجراء كشف المحددات في هذا المشروع.

2.2.2 خوارزميات YOLO لاكتشاف الأغراض (YOLO Object detection algorithms)

تعد خوارزميات يولو (YOLO - You Only Look Once) من أشهر خوارزميات الكشف عن الأغراض . تم إطلاق يولو في عام 2015 وحظي بشعبية كبيرة بسبب سرعته ودقته. وتم إصدار مجموعة من نماذج يولو أهمها:

- **YOLOv2:** حسّن YOLOv2 النموذج الأصلي عن طريق دمج التطوير المتسلسل (Batch Normalization) وهي تقنية تعتمد في التقليل من الخطأ الناتج عن تغير توزيع الدخل في كل مجموعة مصغرة من الدخل (Mini-batch)، وصناديق الارتكاز (Anchor Boxes) وهي صناديق بأبعاد

محددة ومختلفة تستخدم كنقاط ارتكاز يتم الانطلاق منها للكشف عن الأغراض في الصور وتفيد في تحسين دقة اكتشاف موقع الأغراض، ومجموعات الأبعاد (Dimension Clusters) وهي تقنية تجميع للأغراض ذات الحجم المتقارب وتستخدم في توليد صناديق الارتكاز حيث تسهم في تحسين دقة حجم الغرض المكتشف.

- **YOLOv3**: أطلق YOLOv3 عام 2018، ومن مزاياه استخدام شبكة استخراج مزايا أكثر كفاءة من النموذج السابق، وصناديق ارتكاز متعددة، و تجميع هرمي للسمات (Spatial Pyramid Pooling) والذي يفيد في كشف الأغراض بمختلف أحجامها.
- **YOLOv4**: تم إصدار YOLOv4 في عام 2020، ومن مزاياه منهجية دعم البيانات باستخدام الموزاييك (Mosaic Data Augmentation) وهي طريقة لدعم البيانات بصور إضافية هي عبارة عن دمج موزاييك لأكثر من صورة وذلك يفيد في زيادة دقة الكشف، وتم تعريف دالة خسارة جديدة.
- **YOLOv5**: حسّن YOLOv5 أداء النموذج وأضاف ميزات جديدة مثل تحسين المعاملات (Hyperparameter Optimization)، وتقنيات التتبع (Experiment Tracking)، والتصدير التلقائي للنموذج المدرب إلى صيغ التصدير الشائعة.
- **YOLOv6**: تم إطلاق YOLOv6 كمصدر مفتوح من قبل Meituan في عام 2022 ويتم استخدامه في العديد من روبوتات التوصيل الذاتي للشركة.
- **YOLOv7**: أضاف YOLOv7 مهام إضافية مثل تقدير المواقف (Pose Estimation) على مجموعة بيانات COCO keypoints.
- **YOLOv8**: هو أحدث إصدار من YOLO من Ultralytics. يعد أحدث النماذج المتوفرة إلى وقت تنجيز المشروع (State of the art SOTA)، يطور YOLOv8 على نجاح الإصدارات السابقة، مقدّمًا ميزات وتحسينات جديدة لتحسين الأداء والمرونة والكفاءة. يدعم YOLOv8 مجموعة كاملة من مهام الرؤية الحاسوبية، بما في ذلك الكشف والتجزئة وتقدير المواقف والتتبع والتصنيف. تتيح هذه القابلية للتكيف للمستخدمين الاستفادة من قدرات YOLOv8 عبر تطبيقات ومجالات متنوعة. وبناءً على الدراسات المرجعية تم اختيار نموذج YOLOv8 في كشف الأغراض في هذا المشروع.

2.3 خوارزميات التعلم المعزز وتطبيقاتها في القيادة الآلية

لقد تزايد الاعتماد على خوارزميات التعلم في تطوير نظم المحاكاة المقلدات ومن أكثرها استخداماً في هذا المجال هو التعلم المعزز. يوجد العديد من الأمثلة على خوارزميات التعلم المعزز مثل Q-learning و DQN و DDQN والخوارزميات المطورة منها والتي سيتم شرح الفكرة العامة عنها فيما يلي. تقدم جميع هذه الخوارزميات حلاً لمسألة اختيار أفعال (actions) ملائمة للسياسة العامة (policy) ضمن حالة معينة (state) وذلك بهدف زيادة الربح المحدد للمسألة المطروحة (cumulative reward) والذي قد يكون من أمثلته زيادة عدد نقاط اللاعب في محاكاة ما من خلال اختيار الأفعال الصحيحة في حالاتها المناسبة أو اختيار السرعة وزاوية الدوران لسيارة ذاتية القيادة في مسار ما بهدف قطع أطول مسافة ممكنة.

2.3.1 البنية العامة لخوارزميات التعلم المعزز - مسائل اتخاذ القرار المرحلية بنموذج

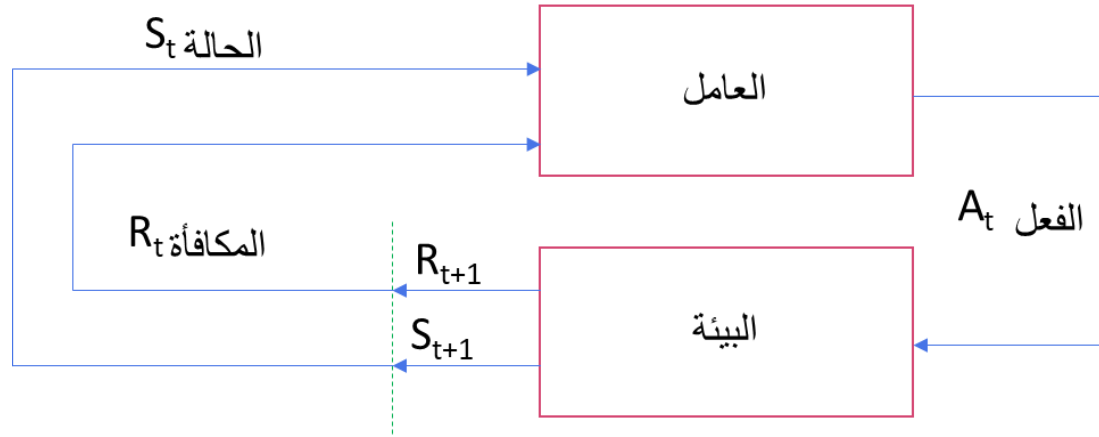
ماركوف Markov Decision Processes (MDPs) - Structuring A

Learning Problem Reinforcement

تتألف مسائل التعلم المعزز عموماً من سلسلة من القرارات المرحلية المرتبطة. ويتم ضمن نموذج ماركوف MDP تعريف مجموعة من المفاهيم المعيارية للتعامل مع هذه المسائل. يعد العامل agent أهم هذه المفاهيم ويعبر عن الكائن الذي يقوم باتخاذ القرارات ضمن البيئة environment التي يوضع فيها [13]. في كل خطوة step يقوم العامل بأخذ تصور عام عن البيئة المحيطة وحالتها state. ويقوم العامل باختيار أفعال actions بناءً على الحالة الحالية للبيئة وتقوم هذه الأفعال بنقل البيئة لحالة جديدة وعندها يتم منح العامل مقداراً من المكافآت reward أو العقوبات بما يتماشى مع أفعاله السابقة. وباختصار فإن المكونات الرئيسية لنموذج MDP هي

- العامل Agent
- البيئة Environment
- الحالة State
- الفعل Action
- المكافأة Reward

ويوضح الشكل 7 أدناه هذه المكونات والعلاقات بينها. وتدعى العملية التي يتم من خلالها اختيار فعل محدد من أجل حالة ما ونقل البيئة لحالة جديدة مع تلقي مكافأة بشكل مستمر Trajectory. حيث يعمل العامل على مضاعفة قيم الربح الناتج عن المكافآت عن طريق اتخاذ قرارات مثلى بخصوص الفعل المناسب لحالة معينة. وهذا يعني أنه لا يتم مضاعفة الربح الحالي فقط من أجل الحالة الحالية بل الربح التراكمي عبر الزمن Cumulative reward [13].



الشكل 7 البنية العامة لخوارزميات MDP

ويتم تعريف الربح التراكمي المخمد G_t الذي تم ذكره سابقاً والذي يحاول العامل جعل قيمته أعظمية خلال فترة التعلم بالعلاقة التالية

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots + \gamma^K R_T$$

حيث T هي الخطوة النهائية (Terminal step). و γ هو معامل التخمين $\in [0,1]$ (والذي وجوده متعلق بوجود عمليات تعلم مستمرة وغير منتهية مما يستدعي تخمين الخطوات اللاحقة بما يتناسب مع بعدها عن الخطوة الحالية).

من المفاهيم التي يجب التطرق إليها أيضاً هو مفهوم السياسة (Policy) ويرمز لها $\pi(a|s)$ وتعبر عن احتمالية اختيار فعل a في حالة s . وهنا يتم دراسة جدوى اختيار العامل لفعل معين بحالة ما عن طريق تابعين يسميان توابع القيمة (Value functions) ومن أهم أشكالها المستخدمة في شبكات الجودة التي سيتم التطرق لها لاحقاً هو تابع القيمة الخاص بالأفعال والذي يعطى بالعلاقة التالية

$$\begin{aligned} q_{\pi}(s, a) &= E_{\pi}[G_t | S_t = s, A_t = a] \\ &= E_{\pi}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a] \end{aligned}$$

ويدعى تابع القيمة الخاص بالأفعال Q-function، وتدعى قيمة هذا التابع من أجل فعل محدد وحالة محددة Q-value وتعبر Q عن Quality ويدل ذلك على جودة اختيار فعل معين في حالة معينة. ويتم اختيار السياسة المتبعة (policy) عن طريق اختيار الفعل الأعلى جودة في كل حالة والذي من أجله قيمة تابع الجودة أعظمية $q_*(s, a)$.

$$q_*(s, a) = \max_{\pi}(q_{\pi}(s, a))$$

يعد هذا المفهوم محورياً في جميع خوارزميات تعلم الجودة (Q-learning) وهي خوارزميات تنتمي لصف خوارزميات الاختلاف الزمني (Temporal difference algorithms (TD)) [13] والتي سيتم الشرح عنها في الفقرة التالية.

2.3.2 شبكات الجودة (Q-learning Networks)

خوارزميات الفارق الزمني TD (Temporal Differencing) هو نوع من الخوارزميات التي تم تطويرها كخوارزمية هجينة تجمع بين أفكار مونتي كارلو والبرمجة الديناميكية. ويقوم مبدؤها على التعلم المعتمد على الملاحظات المستوحاة مباشرة من البيئة المحيطة وليس بناءً على أي نموذج مبني عن تلك البيئة. تعتمد هذه الخوارزميات على مبدأ bootstrapping المستوحى من البرمجة الديناميكية والذي ينص على الاستفادة من القيم المتوقعة لتابع الجودة الخاص بالحالات المستقبلية اللاحقة لدى حساب الجودة من أجل حالة وفعل معينين ولكنه يختلف عن البرمجة الديناميكية في أنه يعتمد الحالة اللاحقة مباشرة للحالة الحالية وليس التوزيع الاحتمالي لجميع الحالات المتوقعة.

تقوم شبكات الجودة بالموازنة بين مرحلتين هما الاستكشاف والاستغلال (Exploration and Exploitation) وهما مفهومان متعلقان بهدف الخوارزمية القائم على تعظيم قيمة الربح التراكمي خلال فترة التعلم حيث يتم في مراحل الاستغلال البحث عن الفعل الذي يوافق القيمة الأعظمية لتابع الجودة ولكن إيجاد هذا الفعل يتطلب القيام بمجموعة من خطوات الاستكشاف الغير جشعة (greedy) والتي توازن بين إيجاد الفعل الأمثل ومن ثم اتخاذه. ويتم تطبيق هذا المفهوم عن طريق تعريف متحول epsilon greedy قيمته تبدأ عند 1 وتتناقص مع التقدم بالخبرة وذلك يسمح باتخاذ قرار بناءً على توليد عدد عشوائي بين 0 و 1 واختيار القيام بالاستكشاف أو الاستغلال بحسب القيمة الكبرى بين العدد المولد و ϵ و يؤدي ذلك وضوحاً إلى أن أغلبية العمليات الاستكشافية ستتم في بداية التدريب ويتم الاستغلال لاحقاً.

إن معظم المسائل ذات الأهمية تتصف بالتعقيد العالي والذي يمنع من حساب درجة الجودة من أجل جميع الأفعال الممكنة في جميع الحالات المتاحة بشكل مستقل. ولذلك يتم الاعتماد على بارامترات محددة لحساب تابع القيمة $q_{\pi}(s, a, \theta_t)$. ويكون عندها المنهج المتبع في شبكات الجودة هو تحديث قيم البارامترات بعد

القيام بفعل A_t ضمن حالة S_t وتلقي مكافأة R_{t+1} مما ينقل البيئة لحالة جديدة S_{t+1} ويتم ذلك التحديث وفق المعادلة

$$\theta_{t+1} = \theta_t + \alpha(Y_t^Q - q_\pi(s, a, \theta_t)) \nabla_{\theta_t} q_\pi(s, a, \theta_t)$$

حيث α هو معدل التعلم و Y_t^Q هو القيمة الهدف التي ترغب الخوارزمية بالوصول إليها والمعرفة وفق المعادلة

$$Y_t^Q = R_{t+1} + \gamma \max_a (q_\pi(s, a, \theta_t))$$

ويمثل هذا التحديث *stochastic gradient decent* الذي يسعى لتقارب قيمة الجودة الحالية $q_\pi(s, a, \theta_t)$ من الجودة العظمى [13] Y_t^Q .

2.3.3 شبكات الجودة العميقة (Deep Q Networks)

إن شبكة الجودة العميقة (DQN) هي شبكة متعددة الطبقات تعطي من أجل حالة s متجهاً من قيم الأفعال $q_\pi(s, \cdot, \theta)$ (actions values)

حيث θ هي بارامترات الشبكة. ومن أجل فضاء حالة من n بعد وفضاء أفعال من m بعد، تكون الشبكة العصبونية تابعاً من \mathbb{R}^n إلى \mathbb{R}^m . لكل شبكة جودة عميقة عاملان مهمان يجب اعتبارهما وهما: استخدام شبكة هدف (Target network) وتكرار التجربة (Experience replay)، إن الشبكة الهدف هي شبكة عصبونية مطابقة للشبكة الأم إلا أنها تأخذ بارامترات θ^- بنسخها من الشبكة الأم كل τ خطوة، بحيث أن $\theta^- = \theta_t$ ، وتبقى ثابتة من أجل الخطوات الأخرى. وإن الهدف المستعمل في شبكة جودة عميقة (DQN) هو

$$Y_t^{DQN} = R_{t+1} + \gamma \max_a (q_\pi(s, a; \theta_t^-))$$

ويتم تكرار التجربة (Experience replay) عن طريق تخزين الانتقالات والاحتفاظ بها ضمن ذاكرة يتم استدعاؤها بشكل منتظم لتحديث الشبكة. وأظهرت شبكات الجودة العميقة تحسناً كبيراً في النتائج بسبب استخدام تكرار التجربة والشبكة الهدف [2].

2.3.4 شبكات الجودة المزدوجة (Double Q-learning)

إن اتخاذ القرار بشأن الفعل الذي يتم اختياره وبشأن مقدار جودته في DQN يتم عن طريق تابع \max يقوم باختيار الفعل ذو الجودة الأعلى. إلا أن أحد مشاكل هذه الشبكات تنتج عن التابع السابق نفسه وهي مشكلة المبالغة في قيم الأفعال وكون قيم جودة الأفعال تميل لأن تكون متفائلة. وأحد الحلول التي تم

تطبيقها لتفادي المشكلة السابقة هو فصل عملية تقييم جودة الأفعال عن عملية اختيارها. وتعد الفكرة السابقة مبرر ظهور خوارزمية الجودة المزدوجة.

يتم في هذه الخوارزمية تعليم تابعين منفصلين للجودة لكل منهما أوزان مستقلة عن الآخر θ' و θ ، عن طريق شبكتين عصيونيتين مستقلتين. ومن أجل كل عملية تحديث للأوزان يتم استخدام إحديهما لاختيار الفعل عن طريق خوارزمية جشعة باستعمال تابع max، ويتم استخدام الأخرى لتحديد جودة الفعل المختار. وتكون المعادلة المعدلة للخوارزمية المذكورة على الشكل التالي

$$Y_t^{DoubleQ} = R_{t+1} + \gamma q_{\pi}(s, \max_a (q_{\pi}(s, a; \theta_t)) \theta'_t)$$

من الجدير ذكره أنه وفقاً للعديد من الأبحاث فإن الأخطاء الناتجة عن تقديرات متفائلة لقيم جودة الأفعال أدت لتعلم سياسات غير أمثلية. هذه الأخطاء قد تنتج بسبب وجود ضجيج ضمن البيئة أو بسبب التقريب في التقديرات الرياضية أو غيرها. وقد وفرت شبكات الجودة المزدوجة حلاً جيداً للمشكلة المطروحة [14].

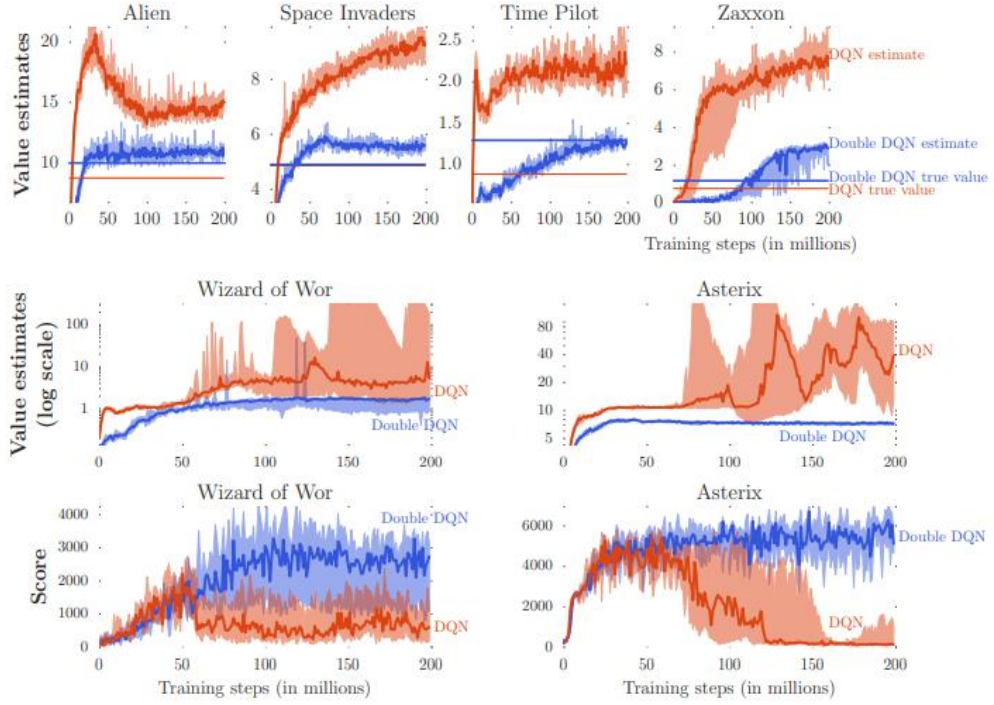
2.3.5 شبكات DQN المزدوجة (Double DQN)

تقوم هذه الخوارزمية على الجمع بين أفكار شبكة الجودة العميقة وشبكة الجودة المزدوجة. بالاستفادة من وجود شبكتين معرفتين ضمن شبكة الجودة العميقة (الشبكة الهدف والشبكة الأم) وبالرغم من أن الشبكتين غير مستقلتين بشكل كامل إلا أنه يمكن استعمالهما في تحقيق الغاية من وجود شبكة مزدوجة. بالتالي يتم انتقاء الأفعال بالاعتماد على الشبكة الأم ويتم اعتماد قيم الجودة في الشبكة الهدف للأفعال المختارة من الشبكة الأم. وتدعى الشبكة الناتجة شبكة DQN المزدوجة. وتصبح معادلة الهدف كالتالي

$$Y_t^{DoubleDQN} = R_{t+1} + \gamma q_{\pi}(s, \max_a (q_{\pi}(s, a; \theta_t)) \theta_t^-) \quad [14]$$

2.3.6 مقارنة بين شبكات DQN و شبكات DQN المزدوجة (Double DQN)

أظهرت النتائج المرجعية المجراة على مجموعة من الألعاب صحة الفرضية القائمة على عدم استقرار نتائج DQN وميلها لإعطاء تقديرات متفائلة لقيم الأفعال. ويوضح الشكل 8 أدناه النتائج المتفائلة لخوارزمية DQN باللون البرتقالي ونتائج Double DQN باللون الأزرق. حيث يتم مقارنة قيم متوسط تقديرات جودة الأفعال على عدد الخطوات خلال تدريب قدره T=125 000 خطوة [14].



الشكل 8 مقارنة بين شبكة DQN و Double DQN

ويوضح الصف الثالث من الشكل السابق مقدار score النموذج المدرب بالاعتماد على كل من الشبكات السابقة ويتضح جلياً تناقص المردود الخاص بالنموذج متناسباً مع تزايد تفاؤل قيم النموذج DQN فيما تكون نتائج الشبكة المزودة مستقرة خلال فترة التدريب [14].

2.3.7 شبكات الجودة العميقة المعتمدة على بيانات إيضاحية (Deep Q-learning)

(from Demonstration)

أظهرت الدراسات أن خوارزميات التعلم المعزز العميق تعطي نتائج جيدة في البيئات البسيطة إلا أنها تتطلب كمّاً هائلاً من بيانات التدريب لإعطاء أداء مقبول وتم اقتراح خوارزميات تتعلم بالاعتماد على مجموعة من البيانات التوضيحية المسجلة لأداء النظام نفسه. حيث قدمت هذه الخوارزميات تسريعاً كبيراً لتقارب شبكات الجودة من تطبيق السياسات المرجوة للعامل.

تقوم الفكرة الأساسية لهذه الخوارزميات على إجراء عملية تدريب مسبق (pretraining) على ذاكرة لتكرار التجربة (Replay memory) مكونة من تجارب مسجلة مسبقاً لمحاكاة خبير بشري أو غير ذلك لأداء المهمة

المطلوبة من الشبكة. حيث يتم خلال فترة التدريب المسبق على شبكة أم وشبكة هدف إضافة مقدار خطأ لتابع الخسارة المتعارف عليه بحيث يعبر الخطأ المضاف عن مقدار الفرق بين الفعل الذي تختاره الشبكة وبين فعل الخبير من أجل الحالة نفسها. حيث يتم تهيئة كل من الشبكة الأم والشبكة الهدف بدايةً بأوزان عشوائية وتتقارب هذه الأوزان من أداء الخبير خلال التدريب. وتعطى علاقة الخطأ المضاف بالعبارة التالية

$$J_E(q) = \max_a (q(s, a) + l(a_E, a)) - q_\pi(s, a_E)$$

حيث يعبر a_E عن فعل الخبير ويعبر المقدار $l(a_E, a)$ عن الفرق بين فعل الخبير والفعل المختار من قبل الشبكة للحالة المدروسة ويساوي هذا المقدار الصفر عند تساويهما ويكون موجباً غير ذلك. ووضوحاً فإن إضافة قيمة هذا الخطأ لعبارة الخسارة يؤدي إلى ترجيح اختيار أفعال الخبير عن غيرها [15].

2.4 محركات الألعاب كبيئة للمحاكاة

إن استخدام أنظمة المحاكاة في تطوير واختبار خوارزميات القيادة الآلية يوفر العديد من المزايا كالتكلفة المنخفضة ومستوى الأمان العالي وسيتم الحديث في الفقرات التالية عن محركات الألعاب كبيئة توفر مزايا عديدة لمحاكاة القيادة الآلية لما توفره من مصداقية عالية وأنظمة مهينة ومعدة للمحاكاة.

2.4.1 مهيئات استخدام الذكاء الاصطناعي ضمن محركات الألعاب (Artificial

Intelligence Foundations)

عادة ما كان الذكاء الاصطناعي الخاص بكل محاكاة ضمن البيئة البرمجية المخصصة لها وليس تحت مظلة محرك الألعاب عموماً، إلا أن الآونة الأخيرة شهدت انبثاق مجموعة أنماط استخدام للذكاء ضمن الألعاب جعلت من مهيئات استخدام الذكاء الصناعي ضمن الألعاب محط اهتمام للمحركات. حيث تم تطوير برمجيات وسيطة من قبل شركة Gameware Navigation تعنى بتوفير البنى التحتية لتطوير ذكاء نظم المحاكاة كمكونات شبكة الملاحة (Navigation Mesh) والتي توفر خوارزميات لإيجاد أقصر طريق (كخوارزمية A*) و خوارزميات تفادي العقبات وخوارزميات لتحديد نقاط الضعف ضمن الألعاب عن طريق نماذج لاعبين آليين [16]. وسيتم التحدث في الفقرات التالية عن بعض استخدامات تعلم الآلة ضمن نظم المحاكاة و الخوارزميات المتبعة في مكونات Navigation Mesh ضمن محركات الألعاب والتي تعنى بتجنب العوائق أثناء الملاحة وهما خوارزميتا ROV و ORCA.

2.4.1.1 استخدامات تعلم الآلة ضمن المحاكاة (ML uses in simulation)

يستخدم تعلم الآلة بكثرة ضمن محركات الألعاب في الكثير من المجالات ولا يقتصر استخدامه على تطوير الذكاء الصناعي الخاص بالمحاكاة. ومن أهم هذه الاستخدامات:

- التخطيط وتوليد المستويات (Map/Level Generation) يستخدم المطورون خوارزميات تعلم الآلة في الإنشاء الآلي لمخططات المحاكاة (كالأقاليم والخرائط) وهذا يوفر القدرة على الاستمرارية في اللعب وتغيير مستمر في المهام. وهو من المجالات المعقدة التطوير.
- توليد النسيج والمظلال (Texture/Shader Generation)

يستخدم تعلم الآلة بكثرة في تقنيات مثل DEEP FAKES التي لاقت رواجاً كبيراً بعد تطور خوارزميات الإنشاء التنافسية (Generative Adversarial Networks GAN).

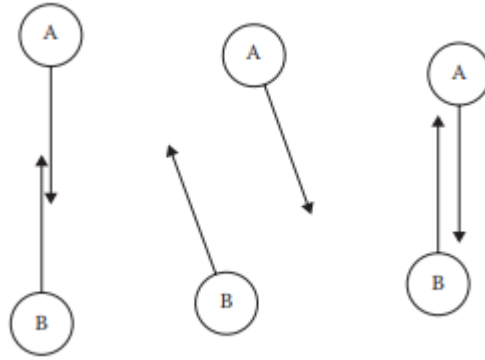
- **توليد النماذج (Model Generation)** يتم استخدام تعلم الآلة أيضاً في مشاريع الإنشاء الآلي للأغراض ثلاثية الأبعاد من خلال القيام بمسح للأغراض صورياً أو وصفها كلامياً ومن أمثلتها تقنيات الواقع الافتراضي والواقع المعزز (Augmented Reality AR/Virtual Reality VR/Meta Reality MR)
- **توليد الأصوات (Audio Generation)** لا يزال هذا المجال قيد التطوير حيث يتم تطوير نماذج تولد موسيقا ومؤثرات صوتية مخصصة لكل محاكاة.
- **اللاعبون الافتراضيون (Artificial Players)** من فوائد هذه النماذج إنشاء لاعبين يلعبون مع اللاعبين الحقيقيين أو استخدام اللاعبين الافتراضيين في تطوير واختبار الألعاب قبل النشر.
- **الشخصيات غير الفعالة و ذكاء الألعاب الصناعي (NPCs or Game AI)** يساهم هذا القسم في إنشاء شخصيات غير لاعبة ضمن المحاكاة تقوم بالتعلم والقيام بأنشطة وأفعال مبتكرة، وفي حين يتم تصميم هذه الأنظمة حالياً بمساعدة خوارزميات الأشجار السلوكية (Behavior Trees) إلا أنه يمكن للنماذج الحديثة توفير تجربة محاكاة أكثر امتعاً للاعبين عن طريق القيام بأفعال جديدة في كل مرة بناء على نتيجة التدريب [17]. ويهتم المشروع بالقسم الأخير من الاستخدامات المذكورة.

2.4.1.2 خوارزمية السرعة التشاركية لتجنب العوائق (Reciprocal Velocity Obstacles ROV)

تستخدم خوارزميات تجنب العوائق بكثرة في محركات الألعاب في بناء ومحاكاة أنظمة الملاحة الآلية كالنظام المروري المنجز في هذا المشروع. وتفيد هذه الخوارزميات في تنظيم حركة مجموعة من الوكلاء ضمن المحاكاة.

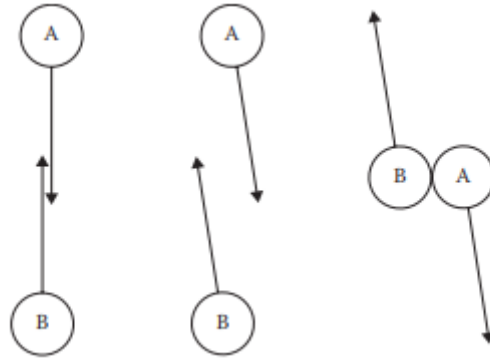
يقوم مبدأ هذه الخوارزمية على أن عند تحديد قيمة السرعة المطلوبة للوصول للهدف ضمن مسار ما بشكل مستقل عن باقي الوكلاء يؤدي لحدوث تصادمات ضمن المشهد. وتنص خوارزميات تجنب العوائق على إيجاد قيمة جديدة للسرعة توازن بين السرعة المطلوبة للوصول للهدف والسرعة التي تقلل من احتمال حدوث تصادمات مع باقي الوكلاء ومع حدود شبكة الملاحة.

وتقدم خوارزمية ROV حلاً لمشكلة اتخاذ القرارات من قبل العملاء بشكل منفصل (Decentralized) ومستقل (Independent) والذي كان سابقاً يؤدي حين محاولة تجنب العوائق لتأرجح الوكلاء ودوران أحدهما حول الآخر كنتيجة عن عدم وجود تواصل بينهما ويوضح الشكل 9 أدناه مثلاً عن هذه الحالة [18].



الشكل 9 التآرجح الناتج عن استقلال الوكلاء عند محاولتهما لتجنب بعضهما

وتعمل هذه الخوارزمية على اتخاذ القرارات للوكيل مع أخذ بعين الاعتبار أن جميع الوكلاء تقوم باتخاذ القرارات في الوقت ذاته وباعتماد على نفس المنهجية. وذلك يقتضي أنه في حالة التقاء الوكلاء فإن كل منهم يقوم بالتحرك بنصف المقدار المطلوب لتجنب الاصطدام وبذلك يتم بالمجمل التحرك بالقيمة المطلوبة للسير السليم. ويوضح الشكل 10 أدناه العملية المذكورة.

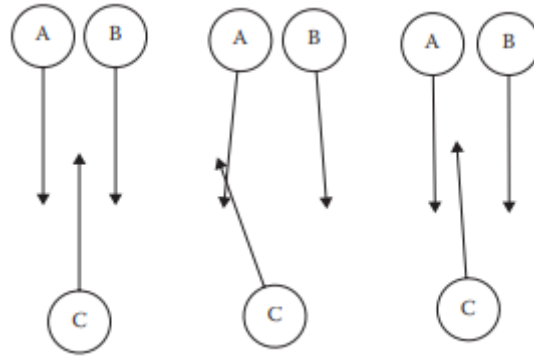


الشكل 10 حل خوارزمية ROV للاصطدام المتوقع في خطوة واحدة بتغيير مسار كل من الوكيلين وبالرغم من أن هذه الخوارزمية لا تعطي حلاً أمثلًا لتجنب العوائق إلا أنها تقدم أداءً جيداً في مختلف الحالات ومنها حالة وجود أكثر من وكيلين وغيرها [18].

2.4.1.3 خوارزمية السرعة التشاركية الأمثلية لتجنب العوائق (Optimal Reciprocal Collision Avoidance ORCA)

عقبت هذه الخوارزمية خوارزمية ROV وأطلق عليها مسمى ROV2. وتعني هذه الخوارزمية بمسألة الأمثلية في إيجاد حل تشاركي بين الوكلاء لتجنب العوائق. وتحافظ هذه الخوارزمية على مبدأ استقلال الوكلاء

المتعارف عليه في VO سابقاً مع إضافة مفهوم جديد وهو المحافظة على التوجه الأولي لتجنب العائق. ويمكن إيضاح هذا المفهوم في حالة وجود 3 عملاء، في خوارزمية ROV كان الحل متأرجحاً وذلك لأنه في حال حاول وكيلان من أصل 3 بتجنب بعضهما فإن جميع الخيارات بالنسبة للوكيل الثالث ستؤدي إلى تصادم مع أحدهما ولكن هذه المشكلة تحل بشكل تلقائي بعد عدة إطارات نتيجة اتخاذ القرار المستمر من قبل العملاء في كل إطار بناء على التغييرات الحاصلة. وفي هذه الحالة بالتحديد تقدم خوارزمية ORCA حلاً أفضل لهذه المشكلة عن طريق الافتراض المبطن بأنه لا داعي لتغيير التوجه الأولي لتجنب وكأن الوكلاء وضعوا صدفة في اتجاهات أولية تضمن تجنب العائق وبالرغم من غرابة هذا الافتراض إلا أنه أثبتت جودة النتائج الناجمة عنه في معظم الحالات لأنه مع تقدم الزمن يؤدي لتزايد المسافة بين الوكلاء مما يسمح بتجنب أفضل. ولكن أحد المشاكل التي تطرحها هذه الخوارزمية هي الإقصاء الكبير للسرعة المحتملة لوكيل معين فمثلاً في حالة 3 وكلاء A, B, C الموضحة في الشكل 11 أدناه [18].

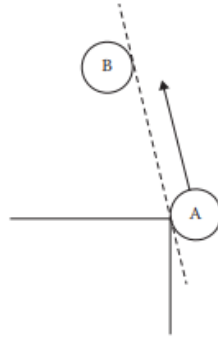


الشكل 11 حالة 3 وكلاء في ORCA

وهنا فإن وجود عائق على يسار A يمنعها من المرور باتجاه اليسار ووجود عائق يمين B يمنعها من الاتجاه لليمين وكلاهما يمنعها من التوجه للأمام أو التوقف أو التراجع. وتحل الخوارزمية هذه المشكلة عن طريق تعريف تابع راحة لجميع القيود على السرعة هو تابع يقوم بإيجاد السرعة الأمثلية التي تزيد من المسافة بين العملاء. وإن هذا التابع هو المسؤول عن حل الحالات المعقدة خلال عدة إطارات وذلك عن طريق اختيار السرعة "الأقل سوءاً" عوضاً عن الأفضل (مثلاً في حالة الصورة فإن خيار C بالتوجه للأمام هو الأقل سوءاً). وإن هذا السلوك يقدم حلاً لا تتجه لاتخاذ خيارات تجنبية قاسية وجذرية مع المحافظة على التوجه الأصلي وذلك يعطي حلاً أمثل وأفضل.

2.4.1.4 مشكلة تجنب العوائق في الزوايا (Cornering)

في خوارزميات VO التقليدية يوجد افتراض أن الوكلاء لديهم اتجاه حركة مفضل للسير باتجاه الهدف وثابت، إلا أن هذا الأمر غير صحيح خصوصاً في الزوايا وذلك بسبب تغير اتجاه الحركة المفضل في كل إطار خلال الدوران حول الزاوية. ويؤدي الافتراض السابق لأداء غريب لخوارزمية RVO وذلك لمحاولتها تقريب الحركة بافتراض اتجاه حركة معين وتغير هذا الاتجاه مع كل إطار. وأما في خوارزمية ORCA فإنه لايسمح بهذا التغير أو التذبذب في الحركة بل يتم الالتزام بتوجه حركة معين، وهذه المنهجية مشاكلها، ومنها عدم قدرة بعض الوكلاء على الحركة أبداً كما هو موضح في الشكل 12 أدناه.



الشكل 12 الزوايا في تجنب العوائق

ويعد هذا السلوك مقبولاً في بعض الحالات كحالة تحرك جميع الوكلاء باتجاه هدف مشترك حيث يمكن لبعضها التحرك وإفساح المجال لغيرها، إلا أنه يؤدي للكثير من الأقفال الميتة (Deadlocks) في حال التقاء العوائق مع الزوايا [18].

2.4.2 محاكاة فيزياء الجسم الصلب والتصادم ضمن محرك الألعاب (Collision and

Rigid Body Dynamics)

تحتوي معظم محركات الألعاب على محرك فيزياء لمحاكاة حركة الأجسام بشكل واقعي نسبياً. وتركز معظم محاكيات الأنظمة الفيزيائية على علم الحركة وهو العلم الذي يدرس القوى وتأثيرها على حركة الأجسام ويخصص بذلك لحركة الجسم الصلب الأهمية الكبرى. ولهذا السبب تقوم محركات الألعاب بتوفير محاكاة للأنظمة الفيزيائية تقوم على افتراضين أساسيين وهما كالتالي

- الميكانيك نيوتوني تقليدي: إذ تتم محاكاة الأجسام بحيث تحقق قوانين نيوتن للحركة وذلك يفترض أن الأجسام كبيرة بما فيه الكفاية لئلا تخضع للقوانين الكوانتية وسرعتها منخفضة بما فيها الكفاية لئلا تخضع لقوانين النسبية (وهذان الأمران محققان بالنسبة للتعامل مع قيادة العربات).
- الأجسام صلبة: جميع الأجسام التي تتم محاكاتها صلبة تماماً ولا يمكن تشويهها. أي أن أشكال الأجسام ثابتة ويتم أخذ هذا المبدأ بالاعتبار أثناء محاكاة الاصطدام. ويتم تبسيط المعادلات الرياضية التي تخضع لها محاكاة الأجسام من خلال هذا الافتراض.
- إن محركات الفيزياء غالباً تسمح بوجود ارتباطات جوهرية بين نظام الفيزياء ونظام التصادم وذلك عن طريق وجود تواجد كشف اصطدام تستدعى عند كل تحديث للجسم الصلب [16].

2.4.2.1 وحدات القياس ضمن محرك الفيزياء (Units)

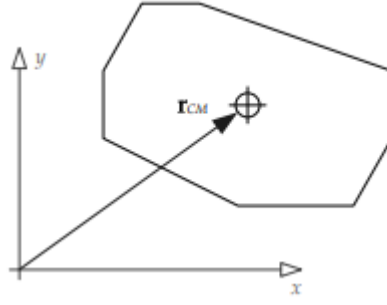
معظم الأجسام الصلبة في المحاكاة تعمل بنظام MKS للوحدات. تقاس في هذا النظام المسافة بالمتر (m) وتقاس الكتلة بالكيلوغرام (kg) و الزمن بالثانية (s). يمكن تغيير هذه الاعتبارات بحيث تكون متسقة ضمن المنظومة. حيث يتطلب ذلك ضرب القيم المعقدة بمعاملات لضمان ذلك الاتساق [16].

2.4.2.2 الفصل بين القيم الخطية والقيم الزاوية (Separability of Linear and Angular Dynamics)

- إن الجسم الصلب الغير مقيد هو جسم يستطيع الحركة بحرية حول ثلاث محاور ونقول عندها أن للجسم 6 درجات حرية (DOF). وتقسّم الحركة لهذه الأجسام لجزئين مستقلين هما
- الحركة الخطية: نتجاهل فيها آثار الدوران وهي حركة تعتبر الجسم نقطة مادية مثالية تكون الكتلة مجمعة فيها ولا يمكنها الدوران.
 - الحركة الزاوية: وهي حركة تصف دوران الجسم حول محور دوران.
- وفيهذا التقسيم في تبسيط محاكاة الأجسام الصلبة [16].

2.4.2.3 الحركة الخطية (Linear Dynamics)

تتم محاكاة الأجسام عن طريق اعتبار موقع الجسم الصلب المرتبط بأغراض المحاكاة شعاعاً يمتد من مركز الاحداثيات المعتبر وحتى مركز ثقل الجسم كما يوضحه الشكل 13 أدناه



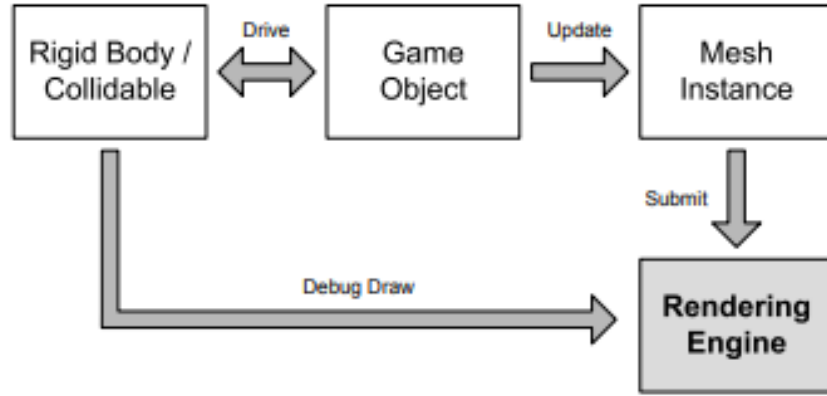
الشكل 13 مركز ثقل الجسم الصلب ضمن جملة إحداثية مرتبطة بالمشهد.

يتم اعتبار السرعة الخطية للجسم (Linear velocity) على أنها المقدار الذي يصف سرعة واتجاه مركز ثقل الجسم مقاسةً بالمتر بالثانية حيث هي المشتق الأول للموضع. وأما تسارع الجسم فهو مشتق السرعة بالنسبة للزمن.

$$v(t) = \frac{dr(t)}{dt} \quad a(t) = \frac{dv(t)}{dt}$$

وأخيراً يتم تطبيق القوى على الجسم، حيث لكل قوة مقدار مطالي (magnitude) واتجاه (direction) وفي حال وجود مجموعة من القوى فإنها تجمع جمعاً شعاعياً. تؤثر القوى على الأجسام الصلبة وفق قانون نيوتن الأول للحركة والذي يعرف بالشكل $\vec{F} = m\vec{a}$.

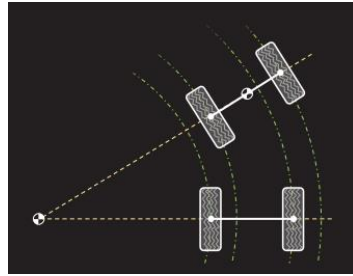
يتم تطبيق القوى ضمن المحاكاة على مكونات الأجسام الصلبة ضمن أغراض المشهد. يمكن تطبيق أي عدد من القوى وذلك ضمن مجال زمني محدد حيث يتم تغيير مقدار واتجاه القوى مع كل إطار. وذلك يتم عبر استدعاء مكاتب نظام الفيزياء (Physics SDKs) من أجل كل إطار. وعادة ما ترتبط مكونات الأجسام الصلبة بأغراض المحاكاة التي تمثل الأجسام المرئية ضمن المشهد، أما مكونات الأجسام الصلبة فهي لاتعدو كونها مكونات برمجية تعبر عن التمثيل الرياضي للمتغيرات المتعلقة بالجسم كالسرعة والكتلة وغيرها. ويوضح الشكل 14 أدناه آلية ارتباطها بأغراض المحاكاة. وقد يرتبط بكل غرض مكون أو أكثر منها [16].



الشكل 14 ارتباط أغراض المحاكاة بمكونات الجسم الصلب.

2.4.2.4 محاكاة حركة التوجيه (Steering)

يتم توجيه العربات من العجلات الأمامية وذلك لأن تدويرها من العجلات الخلفية يؤدي لالتفاف كبير في العربة من أجل تغير صغير في زاوية دوران العجلات. ويوجد أكثر من تمثيل لتوجيه العجلات الأمامية أولها التوجيه القابل للالتفاف (Tunable Steering) الموضح في الشكل 15 أدناه.

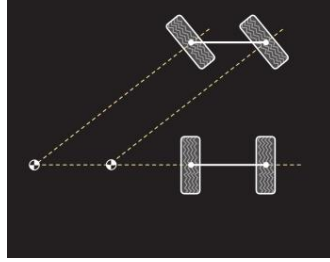


الشكل 15 التوجيه القابل للالتفاف

عند دوران محور العجلات الأمامي تدور العجلات الأمامية والخلفية حول دائرة بمركز مشترك بنفس السرعة الزاوية ويتم حل مشكلة اختلاف البعد عن مركز تلك الدائرة بأن تعطى لكل عجلة سرعة خطية مختلفة وهذه المسألة محلولة ميكانيكياً في نظام التحريك للسيارة.

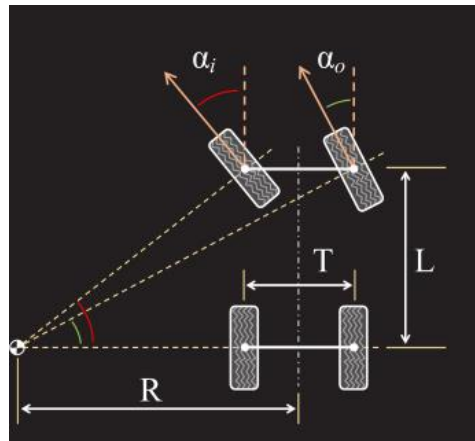
من فوائد هذا التمثيل القضاء على مشكلة الانزلاق الجانبي إلا أن مشكلة هذا التصور لحركة التوجيه هي اضطراب محور العجلة أن يدور بزوايا كبيرة حتى يحصل على الالتفاف المطلوب وهذا يؤدي لتقليل المساحة القابلة للاستخدام من قاعدة العجلات (وهي المسافة بين المحور الأمامي والخلفي للعجلات).

من التمثيلات الأخرى لعملية التوجيه هو التوجيه البسيط (Simple Steering) الموضح في الشكل 16 في هذا النوع من التوجيه يكون لكل عجلة محور دوران خاص بها كما توضح الصورة أعلاه. ويحل هذا التمثيل مشكلة الاستفادة من مساحة قاعدة العجلات إلا أنه يؤدي لمشكلة أخرى وهي الانزلاق المعاكس للتوجيه بشكل أكبر وذلك نتيجة اختلاف نصف قطر الدوران من أجل كل من العجلتين الأماميتين وذلك يؤدي لتعرض كل منهما لمقدار مختلف من الانزلاق [19].



الشكل 16 التوجيه البسيط

الحل الأمثل لهذه المشكلة يعرف بتوجيه آكرمان (Ackerman Steering) الموضح في الشكل 17 المقترح من قبل Rudolph Ackerman وهو توجيه يسمح لكل من العجلات الأمامية الدوران بشكل مستقل عن الأخرى وتطبيق السرعة المناسبة بحيث تلتقي محاور الدوران دوماً بنقطة واحدة مع ممدد محور العجلات الخلفية كما هو موضح بالصورة.



الشكل 17 توجيه آكرمان

توضح الصورة مثلاً عن توجيه آكرمان، حيث تكون زاوية الدوران الخارجية أصغر من الزاوية الداخلية وتمثل كل من القيم

- L هي قاعدة العجلات أي المسافة بين المحور الأمامي والخلفي.
- T هي المسار وهو البعد بين العجلات.
- α_i هي الزاوية الداخلية عن اتجاه الأمام.
- α_o هي الزاوية الخارجية عن اتجاه الأمام.

حيث يمكن بافتراض أن السرعة ثابتة وبغياب القوى الأخرى يمكن تبسيط عبارة الزوايا الداخلية والخارجية
لتحسب من العلاقتين [19]

$$\alpha_i = \arctan\left(\frac{L}{R - \frac{T}{2}}\right)$$

$$\alpha_o = \arctan\left(\frac{L}{R + \frac{T}{2}}\right)$$

وانطلاقاً من الدراسة السابقة تم اختيار تمثيل آكرمان في محاكاة حركة دوران العجلات في المشروع المنجز.

الفصل الثالث

بيئة العمل والأدوات المستخدمة

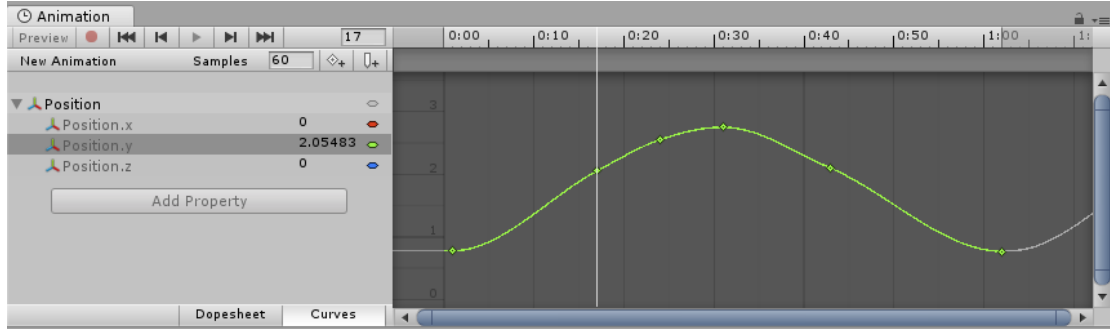
يتضمن هذا الفصل تعريفاً ببيئة العمل والأدوات التي تم استخدامها في إنجاز المشروع.

3.1 محرك الألعاب يونيتي Unity

تقدم هذه الفقرة شرحاً مبسطاً لبعض البرمجيات التي يقدمها محرك الألعاب يونيتي والتي تجعله بيئة مناسبة لتطوير محاكاة لمنظومة القيادة الآلية.

3.1.1 منحنيات التحريك (Animation Curves)

تستخدم منحنيات التحريك في محاكاة حركة الأجسام. وتوفر القدرة على التحكم بكيفية تغير حركة الأجسام عبر الزمن أو بتغير مجموعة أخرى من المتغيرات. ويوضح الشكل أدناه أحد منحنيات التحريك المستخدمة في محاكاة حركة قفز في الهواء للمكعب الموضح في الشكل 18.



الشكل 18 منحنيات التحريك

ولكل منحنى تحريك مجموعة من المفاتيح ونقاط التحكم التي تحدد الأوضاع التي تمر بها الحركة ويتم عرض هذه التفاصيل في محرر المنحنيات (Curve Editor) حيث يسمى الإطار الذي يحوي المفاتيح الخاصة بمنحنى "Keyframe". ولا يقتصر عمل منحنيات التحريك على التحكم بموضع واتجاه الأجسام بل على

التحكم بأي من خصائص أغراض المحاكاة والمتغيرات العامة فيها. حيث يمكن إنشاء تأثيرات وسلوكيات مرئية معقدة عن طريق هذه المكونات. ويمكن إضافة منحنيات تحريك لأي من الأنواع التالية من المتغيرات: Float, Color, Vector2, Vector3, Vector4, Quaternion, Boolean ومن الأمثلة على استخدامات هذه المكونات:

- محاكاة شدة الضوء ودرجة اللون لإعطاء إحاءة بومض الضوء أو نبضات في الإضاءة.
- محاكاة درجة وشدة الصوت لمقطع صوتي لإعطاءه إحاءة واقعية (مثلاً تغيير شدة الصوت أثناء ركض الشخصية)
- محاكاة تغيير المواد أو الأنسجة مثل محاكاة تدفق الماء [20].

3.1.2 تجميع الكائنات (Object pooling)

هو نمط تصميمي يفيد في حالة الحاجة لإنشاء الأغراض وتدميرها بشكل مستمر. وهو نمط تصميمي إنشائي يفيد في تقليل الحاجة لعمليات المعالجة من قبل وحدة المعالجة المركزية CPU. يتم في هذا النمط التصميمي إنشاء جميع الأغراض بشكل مسبق وهذا يلغي الحاجة لإنشاء الأغراض وتدميرها بشكل مستمر. تقوم مجموعات الكائنات (Object pool) بتحسين الأداء بشكل كبير. إن تطبيق هذا النمط أساسي لدى التعامل مع مسائل إنشاء الأغراض بشكل ديناميكي ومن تطبيقاتها أنظمة الازدحام وأنظمة إطلاق النار وغيرها [20].

3.1.3 الملاحة (Navigation)

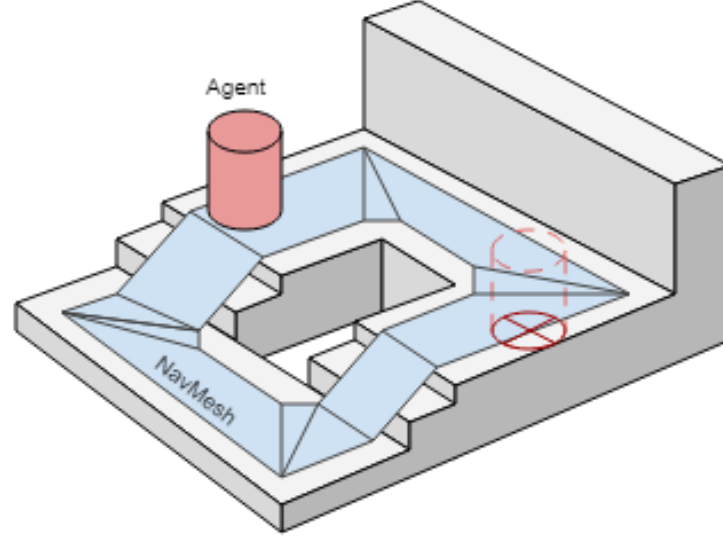
إن أنظمة الملاحة تسمح بإنشاء شخصيات وأغراض تستطيع الملاحة ضمن البيئة. ويتكون نظام الملاحة ضمن يونيتي من الأجزاء التالية:

- **شبكة الملاحة (NavMesh)** هي بنية برمجية تصف المناطق المسموحة للسير ضمن البيئة وتسمح بإيجاد مسار من منطقة قابلة للسير لأخرى. وتسمى عملية إنشاء هذه البنية اعتماداً على هندسة البيئة الخاصة بالمشهد بالخبز (Baking).
- **وكيل الملاحة (NavMesh Agent)** وهو مكون يقوم بإنشاء شخصيات وأغراض تستطيع تجنب بعضها والتحرك باتجاه هدف ما وتجنب العوائق المتحركة.

- **الوصلات خارج الشبكة (OffMesh Link)** تسمح هذه الوصلات بتأمين الاتصال بين المناطق التي لا يمكن التعبير عنها كمنطقة قابلة للسير (كالقدرة على القفز فوق عتبة).
- **عقبات شبكة الملاحة (NavMesh Obstacle)** هي المكونات التي تسمح بتوصيف الأغراض المتحركة لوكلاء الملاحة لتمكينهم من تجنبها. ويتم ذلك عن طريق تغيير الشبكة بشكل جذري للتحرك حول العائق أو عبر إيجاد طرق أخرى للوصول للهدف [20]. وسيتم شرح آلية عمل هذه الأجزاء في الفقرات التالية.

3.1.4 آلية عملية نظام الملاحة (Inner Workings of the Navigation System)

يتطلب تحريك الشخصيات أو الوكلاء ضمن بيئة معينة حل مشكلتين: الأولى هي فهم المشهد و إيجاد الهدف ضمنه والثاني هو إيجاد أفضل الطرق للوصول لهذا الهدف. ويوضح الشكل 19 أدناه مثلاً عن وكيل ملاحة ضمن شبكة ملاحة مع توضيح المناطق القابلة للسير ضمنها.



الشكل 19 شبكة ملاحة بعناصرها المختلفة

يتم تمثيل الوكيل ضمن شبكة الملاحة بأسطوانة بنصف قطر ما، ويتم بعدها بناء شبكة الملاحة بالاعتماد على البنية الهندسية للمشاهد عن طريق تحديد المناطق التي يمكن للوكيل الوقوف عليها والحركة ضمنها. ويتم تخزينها

كمجموعة من المضلعات، حيث يتم تخزين حدود هذه المضلعات والمضلعات المجاورة لها ويتم بهذه المنهجية حل المشكلة الأولى.

و أما بالنسبة للمشكلة الثانية المتعلقة بإيجاد أفضل المسارات للوصول للهدف وتجنب العوائق، فيتم ذلك عن طريق توصيف نقطة البداية ونقطة الهدف على الخريطة المنشأة وتحديد أقرب المضلعات لها. ويتم بعدها البحث من نقطة البداية عن طريق زيارة جميع المضلعات المجاورة حتى الوصول للمضلع الهدف ويتم عادة القيام بعملية البحث هذه عن طريق خوارزميات مثل A* المستخدمة في محرك يونيتي. وتستعمل يونيتي خوارزمية ROV¹ لتوقع وتجنب الاصطدامات.

3.2 مكتبة وكلاء تعلم الآلة ML-Agents

مكتبة وكلاء تعلم الآلة ضمن بيئة يونيتي (Unity Machine Learning Agents Toolkit- ML-Agents Toolkit) هي مكتبة مفتوحة المصدر تمكن بيئة المحاكاة أن تكون بيئة لتدريب وكلاء بخوارزميات الذكاء الصناعي. ويمكن تدريب الوكلاء بخوارزميات التعلم المعزز (Reinforcement Learning) والتعلم بالتقليد (Imitation Learning) والتطور العصبي (Neuroevolution) وأيضاً تسمح للباحثين بكتابة خوارزميات تعلم الآلة الخاصة باستخدام واجهة تخاطب برمجي خاصة بلغة بايثون (Python API) ويمكن استخدام هذه المكتبة في تطوير تطبيقات الواقع المعزز والواقع الافتراضي ونظم المحاكاة والأبحاث. ويمكن الاستفادة من تعلم الوكلاء في مختلف التطبيقات وأهمها التحكم بشخصيات المحاكاة الغير فاعلة (Non-Playable Characters NPC) ويمكن الاستفادة منه أيضاً في الاختبارات المؤتمتة للألعاب في النسخ قبل النشر من خلال شخصيات ذكية تفيد في تقييم وتطوير مجال صناعة الألعاب. وتقدم هذه المكتبة خدمات مهمة بدرجة كبيرة للباحثين ومطوري الألعاب على حد سواء. حيث توفر القدرة على تقييم خوارزميات الذكاء الصناعي ضمن بيئة يونيتي الغنية للمحاكاة. حيث وإن كانت التطبيقات الفيزيائية لخوارزميات التعلم المعزز تتطلب مجسماً واقعياً لتنفيذ المهام المطلوبة منها (كالروبوتات والعربات ذاتية القيادة وغيرها) إلا أنه يعد من المكلف وغير المجدي إجراء عمليات التدريب على المجسمات الفيزيائية وهنا يأتي دور محاكاة يونيتي الغنية والواقعية في محاكاة بيئة التدريب لتطوير خوارزميات ونماذج مفيدة وآمنة [21].

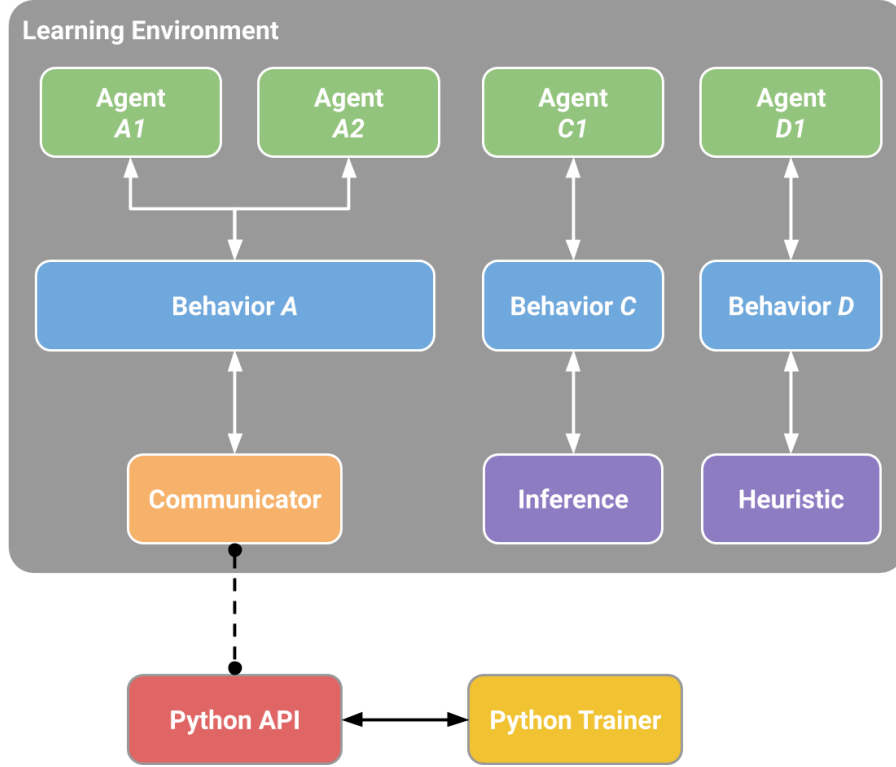
¹الملفلة في (21)

3.2.1 مكونات المكتبة (Key Components)

تتكون ML-Agents من خمسة مستويات رئيسية:

- **بيئة التعلم (Learning Environment):** والتي تتكون من المشهد ضمن بيئة يونيتي وجميع شخصيات المحاكاة. حيث تقوم الوكلاء بالمراقبة واتخاذ الأفعال والتعلم ضمن المشهد في يونيتي. وتوفر المكتبة بنية برمجية تسمى (com.unity.ml-agents package) والتي تمكن المطور من تحويل المشهد في يونيتي إلى بيئة للتدريب بعد تعريف الوكلاء والسلوكيات المرتبطة بهم.
- **واجهة التخابط البرمجية الخاصة بلغة بايثون (Python API):** وتحتوي على واجهة منخفضة المستوى بلغة بايثون (Low-level Python Interface) تسمح بالتخابط مع بيئة التدريب. وهذه الواجهة ليست جزءاً من محرك البحث يونيتي بل مكوناً منفصلاً يتواصل مع يونيتي من خلال أداة تواصل (Communicator). وتتواجد هذه الواجهة ضمن محفظة مخصصة (Python Package) تسمى mlagents-envs والتي تستخدم في عملية التدريب باستخدام الخوارزميات التي توفرها المكتبة كما يمكن استخدامها أيضاً لأغراض أخرى يجعل يونيتي يعمل عمل محرك للمحاكاة فقط والتواصل باستخدامها لكتابة خوارزميات تعلم آلة مخصصة وسيتم التفصيل عن هذه العملية في فقرة واجهة بايثون المنخفضة المستوى للتخابط (Python Low-Level API / PLL-API).
- **المتخابط الخارجي (External Communicator):** يربط بيئة التعلم مع PLL-API ويتواجد ضمن بيئة التعلم.
- **مدرب بايثون (Python):** ويحتوي على جميع خوارزميات تعلم الآلة التي تمكن الوكلاء من التعلم. وتمت كتابتها بلغة بايثون وهي جزء من محفظة بايثون (mlagents Python Package) ولها تعليمية وحيدة وهي mlagents-learn ويمكن من خلالها استخدام الخوارزميات المتوفرة ضمن المكتبة وجميع مدربات بايثون جزء من PLL-API.
- **مغلف جيم (Gym Wrapper):** وهو طريقة شائعة تستخدم من قبل الباحثين في مجال تعلم الآلة عبر مغلف (Wrapper) توفره بيئة OpenAI ويسمى جيم (Gym) وتوفر المكتبة نسخة منه ضمن المحفظة (ml-agents-envs package).
- **مغلف بيتينغ-زو (PettingZoo Wrapper):** وهو واجهة تخابط برمجي بلغة بايثون (Python API) ويستعمل للتخابط مع المحاكاة عديدة الوكلاء (multi-agent Simulation) وهو أيضاً جزء من المحفظة (ml-agents-envs package).

ويوضح الشكل 20 أدناه بنية مكونات المكتبة ML-Agents [21].



الشكل 20 بنية مكونات ML-Agents

3.2.2 التعلم المعزز العميق باستخدام ML-Agents (Deep Reinforcement

Learning with ML-Agents)

توفر ML-Agents خوارزميتين للتعلم المعزز العميق وهما:

- خوارزمية أمثلية السياسة المقاربة (Proximal Policy Optimization PPO)
- خوارزمية تنعيم الفاعل=الناقد (Soft Actor-Critic SAC).

إن خوارزمية PPO هي الخوارزمية الافتراضية ضمن البيئة وهي خوارزمية مستقرة نسبياً. وأما خوارزمية SAC فهي خوارزمية خارجة عن السياسة (Off=Policy) على عكس PPO وهذا يعني أنها قادرة على التعلم من

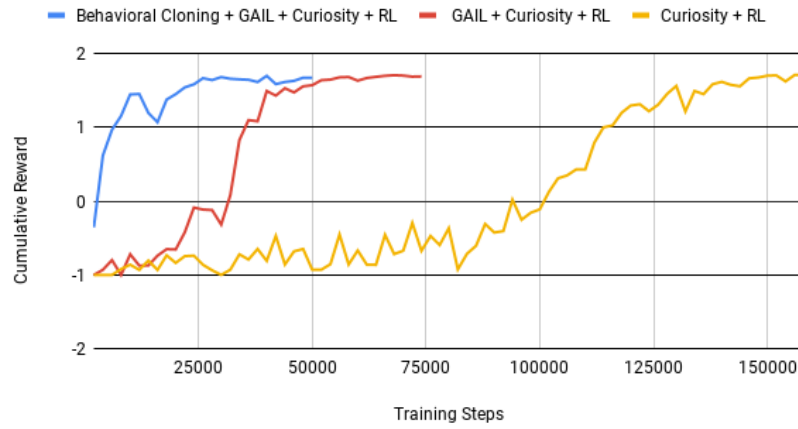
الخبرة في أي لحظة من لحظات التعلم وهذا يجعلها بحاجة لعدد أقل من تجارب التدريب. وهذا يجعلها خياراً أفضل للبيئات البطيئة [21].

3.2.3 التعلم بالتقليد باستخدام ML-Agents (Imitation Learning with ML-Agents)

ML-Agents)

من المنطقي عند تدريب الوكيل على أداء مهمة ما استخدام مجموعة من التجارب التوضيحية لشرح المهمة المطلوبة للوكيل بدلاً من استخدام المكافآت فقط في مساعدته على التعلم وتوفر مكتبة ML-Agents خوارزميتين للتعلم بالتقليد إلى جانب التعلم المعزز وهما خوارزميتا التعلم بالتقليد عن طريق التوليد الخصومي (Generative Adversarial Imitation Learning/ GAIL) والتعلم بالاستنساخ السلوكي (Behavioral Cloning) ويمكن إجراء التعلم بالتقليد بشكل منفصل أو على التوازي مع التعلم المعزز ويفيد عند استخدامه بشكل منفصل في استنساخ سلوك معين وأما عند استخدامه مع التعلم المعزز فيفيد في تقليل زمن التعلم وهذا مفيد بشكل خاص في البيئات الفقيرة بإشارات الربح والمكافأة (Sparse Reward Environment) وهي الحالة التي يصعب على الوكيل اكتشاف السلوك المطلوب منه بالاعتماد على التجارب العشوائية وحدها لقلّة مصادر الحصول على مكافأة (يقوم الوكيل بالكثير من التجارب العشوائية قبل الحصول على مكافأة). ويوضح الشكل أدناه النتائج باستخدام الخوارزميتين المذكورتين مع بيئة مطورة ضمن المكتبة تسمى (Pyramid Environment) خلال 6 مراحل (Episodes) حيث تشير قيمة Curiosity إلى تطبيق لشبكة (Curiosity-driven Exploration by Self-supervised Prediction) والتي تهتم باختصار بتشجيع الوكيل على استكشاف البيئة. ونلاحظ في الشكل 21 أن التعلم باستخدام بيانات إيضاحية قام بتسريع التعلم بأكثر من 4 أضعاف كما نلاحظ في الخط الأزرق [20].

Reinforcement Learning using Demonstrations on Pyramids



الشكل 21 استخدام بيانات إيضاحية في التعلم المعزز

وكخلاصة، فإن مكتبة ML-Agents تقدم 3 طرائق لتدريب الوكلاء من خلال 4 خوارزميات مبنية ضمن المكتبة وهي BC, GAIL وخوارزميات التعلم المعزز العميق PPO و SAC وتتيح استخدامها بشكل مستقل أو على التوازي. وأما في حال الرغبة في استخدام أي خوارزميات أخرى لا تدعمها المكتبة فينتحتم استخدام PLL-API والتي سيتم شرحها في الفقرة التالية.

3.2.4 واجهة بايثون المنخفضة المستوى للتخاطب (Python Low Level API / PLL-API)

PLL-API)

تحتوي محفظة بايثون mlagent python package على مركبتين وهما:

- واجهة تخاطب منخفضة المستوى (low-level API) والتي تسمح بالتخاطب بشكل مباشر مع بيئة يونيتي والتي تنشأ ضمن mlagents_envs.
- ونقطة دخل مباشرة للتدريب من خلال الأمر mlagents-learn والذي يسمح بتدريب بيئة يونيتي باستخدام الخوارزميات التي تدعمها المكتبة للتعلم المعزز والتعلم بالتقليد التي تم التطرق لها سابقاً.

وتستعمل واجهة التخاطب المنخفضة المستوى للتخاطب مباشرة مع بيئة التعلم وبذلك تتيح للمطور حرية بناء وتطوير خوارزميات خاصة وتدريبها.

وتتكون هذه الواجهة من الأقسام الرئيسية التالية:

- بيئة يونيتي (UnityEnvironment) وهي الواجهة الرئيسية بين تطبيق يونيتي وبين الرماز البرمجي الخاص بالتدريب. وتستخدم UnityEnvironment للتحكم بالمحاكاة وتشغيل جلسة التدريب.
 - اسم السلوك (BehaviorName) وهو المعرف الأساسي للسلوك ضمن المحاكاة ويخزن كسلسلة محرفية (String).
 - رقم هوية الوكيل (AgentId) رقم صحيح يخزن بمتحول من نوع int ويحدد هوية الوكيل ضمن المحاكاة.
 - خطوات القرار (DecisionSteps) تحتوي على بيانات الوكيل التابعة للسلوك "Behavior" كقيم الأفعال (Actions) والمراقبات (Observations) والربح (Rewards). ويقوم الوكلاء بعملية طلب القرار من البيئة، حيث يتم تخزين خطوات القرار الخاصة فقط بالوكلاء الذين قاموا بطلب القرار منذ آخر استدعاء للتابع env.step().
 - خطوات الانتهاء (TerminalSteps) تحتوي على بيانات الوكيل التابعة لنفس السلوك ضمن المحاكاة مثل DecisionSteps ولكنها مخصصة للوكلاء الذين انتهت حلقة التدريب الخاصة بهم (Episode).
 - واصفات السلوك (BehaviorSpec) تصف شكل بيانات المراقبة (Observations) ضمن كل من DecisionSteps و TerminalSteps.
- ويجري التواصل بين يونيتي وبايثون من خلال مقبس مفتوح (open socket) من دون تحقق (authentication) بافتراض أن الشبكة التي يتم العمل ضمنها آمنة، حيث يتم التواصل مع بيئة التدريب من خلال إنشاء ملف لبنية المحاكاة (binary build file) وتحميله ضمن بايثون.
- وتتخاطب الواجهة مع بيئة يونيتي من خلال الواجهة البرمجية BaseEnv بالاعتماد على الإجراءات التالية:
- **Reset** وترسل للبيئة إشارة إعادة ضبط للبيئة وتعيد None.
 - **Step** ترسل رسالة للتقدم في البيئة وتعيد القيمة None وليست step متوافقة مع أي من التوابع Update أو FixedUpdate ضمن حلقة المحاكاة ضمن يونيتي بل تستمر المحاكاة ضمن يونيتي دون إشارة step أو reset حتى يحتاج الوكيل لقرار يأخذ قيمته من فعل يتم تحديده في بايثون.
 - **Close** إشارة إغلاق للمحاكاة وإيقاف التواصل.
 - **Behavior Spec** تعيد تخطيطاً (mapping) لقيم BehaviorSpec الخاصة بكل BehaviorName وتحتوي على بيانات أشكال المراقبات وقيم ActionSpec وهي قيم أشكال الأفعال المطلوبة وتكون أحجام هذه الخصائص مثبتة بالنسبة لكل بيئة.

- **Get Steps** تعيد Tuple بقيم DecisionSteps, TerminalSteps وتحوي TerminalSteps أيضاً على سبب انتهاء حلقة التدريب (فيما لو وصل الوكيل للقيمة العظمى لعدد الخطوات ضمن حلقة أم هل تمت مقاطعته).
- **Set Actions** تحدد الأفعال لمجموعة من الوكلاء. ويتم ذلك عن طريق إرسال ActionTuple مكون من مصفوفة ثنائية البعد np.array من النوع dtype=np.int32 للأفعال ذات القيم المتقطعة (discrete) ومن النوع dtype=np.float32 من أجل الأفعال ذات القيم المستمرة (continues).
- **Set Action for Agent** تحدد الأفعال لوكيل محدد [21].

الفصل الرابع

التطبيق العملي

يتضمن هذا الفصل شرحاً لخطوات التنفيذ العملي للمشروع

4.1 مقدمة

يتضمن هذا القسم ثلاثة أجزاء أساسية، الجزء الأول هو بناء بيئة المحاكاة والذي يشتمل على بناء المدينة والعربة وتزويدها بالحساسات والميكانيك والنظام المروري من السيارات والمشاة وعلامات المرور، والجزء الثاني هو ربط بيئة المحاكاة مع بيئة التدريب بالاستعانة بمكتبة ML-Agents، وأما الجزء الثالث فيتكون من تدريب نموذج القيادة الذاتية تدريجياً بدايةً بالتدريب بالاستعانة بحساس ليدار وثم القيام بكشف محددات الطرق وتدريب عملية "Lane keeping" ومن ثم إجراء عملية كشف الأغراض ضمن المشهد وتدريب العربة على استخدام دخل الحساسات السابقة وأخيراً مكاملة الأجزاء التدريبية ضمن المدينة المنشأة في الجزء الأول. وسيتم الحديث عن الأجزاء السابقة بالتفصيل في الفقرات التالية.

4.2 إنشاء بيئة المحاكاة

تم تنجز هذا الجزء من خلال إنشاء العربة ومحاكاة المنظومة الميكانيكية المرتبطة بها وتزويدها بالحساسات اللازمة لعملية التدريب لاحقاً (حساس ليدار وكاميرا)، وتنجز محاكاة مناسبة للمدينة والمنظومة المرورية المرتبطة بها (مشاة وعربات وعلامات مرور) وسيتم تفصيل الخطوات بالترتيب.

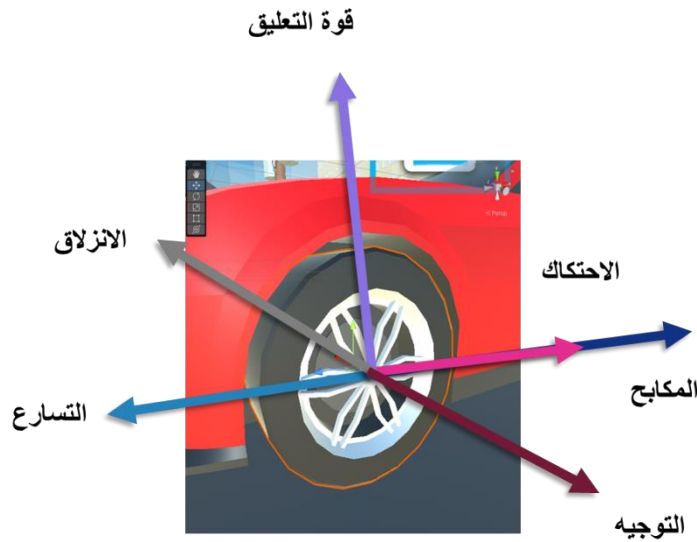
4.2.1 إنشاء محاكاة العربة

تمت محاكاة العربة عن طريق محاكاة المنظومة الميكانيكية المرتبطة بها وتزويدها بالحساسات اللازمة في التدريب وسيتم الحديث عن هذين القسمين بالترتيب.

4.2.1.1 محاكاة المنظومة الميكانيكية

توفر يونيتي بعض الخواص لإنشاء محاكاة لفيزياء العربة عن طريق مكونات الجسم الصلب ضمن محرك الفيزياء، إلا أنه ولغرض زيادة القدرة على التحكم بدرجة الواقعية في المحاكاة قدر الإمكان لزيادة وثوقية نماذج القيادة الذاتية المطورة بالاعتماد على هذه المحاكاة ولجعل المنظومة الفيزيائية قادرة على التوسع خلال تطور المشروع لم تعتمد تلك الخواص بل تم إنشاء نظام فيزياء مخصص (customized) يعتمد على محاكاة القوى المؤثرة في العربة.

حيث تمت محاكاة ستة قوى وهي قوة التعليق (suspension) وقوة الانزلاق (slipping) وقوة الاحتكاك (friction) وقوة التسارع (acceleration) وقوة المكابح (breaks) وقوة التوجيه (steering). يوضح الشكل 22 هذه القوى وسيتم الحديث عن تطبيق كل منها بالترتيب:

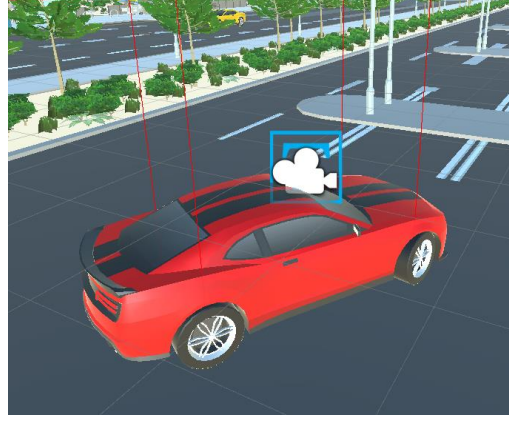


الشكل 22 القوى المؤثرة في العجلة

- **التعليق (Suspension):** هذه القوة مرتبطة بتوتر النابض الذي يربط العربة بالعجلات وهي القوة المسؤولة عن ثبات واستقرار العربة خلال الحركة ويقوم النابض بامتصاص الصدمات عند تعرض العجلة لعائق ما والحفاظ على التوازن لئلا تؤثر الصدمات على استقرار العربة. وتم تطبيقه من خلال حساب قيمة الانزياح عن وضع التوازن (offset) وذلك عن طريق إطلاق شعاع افتراضي من كل عجلة (Raycast) وحساب المسافة عن موضع الاصطدام وتم حساب القوة من المعادلة:

$$\vec{K} = (\vec{offset} \times strength) - (\vec{v} \times damping)$$

حيث **strength** هي مقدار متعلق بصلابة النابض و \vec{v} هي المركبة الواقعة على محور النابض من سرعة العجلة (وتحسب عن طريق حساب جداء سلمي بين اتجاه النابض المفترض وسرعة العجلة) وأخيراً فإن **damping** هو مقدار التخميد الذي يعمل على محاكاة منظومة التخميد (Damper) الموجودة في العربات وهي عبارة عن سائل لزج يعمل على مخامدة اهتزاز النابض عند الاصطدام بالعوائق. ويوضح الشكل 23 قوة التعليق المطبقة على العجلات باللون الأحمر.

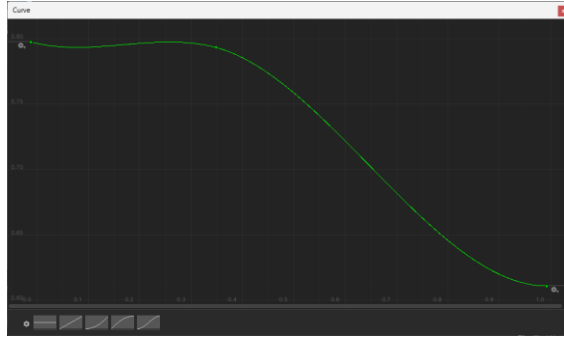


الشكل 23 قوة التعليق المطبقة على العجلات

- **قوة الانزلاق (Slipping):** عند انعطاف العربة باتجاه ما، تخلق مركبة السرعة الموازية لمحور العجلة قوة انزلاق تدفع العربة للانزلاق عن الطريق وتحسب السرعة المطلوب الحصول عليها لضمان عدم انزلاق العجلة بالعلاقة

$$\vec{v} = -1 \times \vec{v}_s \times \text{grip} \times \text{slip}(v)$$

حيث \vec{v}_s هي سرعة الانعطاف و **grip** هو معامل يعبر عن درجة تشبث العجلة بالطريق وزيادته تؤدي لتخفيف الانزلاق بدرجة أكبر إلا أنه وإن ازداد أكثر من اللازم قد يؤدي لانقلاب العربة بسبب التشبث الزائد، وإن كان أقل من اللازم ازدادت قدرة السيارة على الدوران على الطريق (Drifting) ويمكن التحكم بهذا المقدار حيث يتم معايرته لتحقيق التأثير المطلوب. أما $\text{slip}(\vec{v})$ فهو تابع لسرعة الانعطاف يلعب دوراً هاماً في زيادة واقعية المحاكاة عن طريق جعل قيمة التشبث أكبر عندما تبدأ العجلة بالانزلاق ومن ثم تقليلها عندما تزداد سرعة الانزلاق بشكل كبير ولا تستطيع العجلة مقاومته. وتم تطبيق هذه الخاصية بالاستعانة بمنحنيات التحريك لبناء تابع يتصرف بالشكل المذكور وأخذ معامل بتقدير قيمته من أجل كل سرعة ويوضح الشكل 24 أدناه هذا التابع:



الشكل 24 منحنى التحكم المستخدم في حساب الانزلاق بدلالة السرعة

، ويتم عندها تطبيق قوة احتكاك معاكسة لمقدار الانزلاق الناتج وتعطى بالعلاقة التالية:

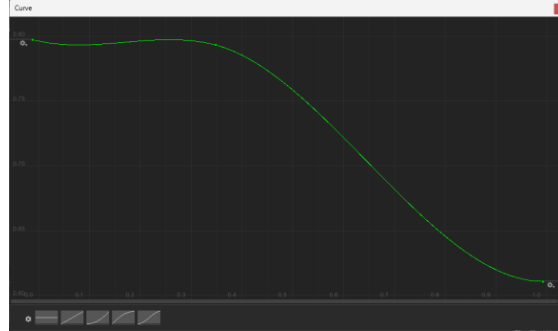
$$\vec{S} = m \times \vec{v}$$

حيث m هي كتلة العجلة.

- **قوة الاحتكاك (Friction):** هي قوة مرتبطة بشكل مباشر مع قوة التسارع وتم حسابها من العلاقة التالية:

$$\vec{F} = -\overrightarrow{\text{acceleration direction}} \times \text{friction}(v)$$

عن طريق حساب سرعة العربة وتطبيق قوة تعاكس جهة السرعة **acceleration direction** مع الضرب بمعامل احتكاك **friction(v)** يتم حسابه باستخدام تابع للسرعة، إذ تم بناء هذا التابع باستخدام منحنيات التحكم أيضاً بحيث يقوم بمقاومة السرعة إلى حدٍ ما ولكن تأثيره يضعف مع ازدياد السرعة بشكل كبير. ويوضح الشكل 25 منحنى التحكم المذكور.

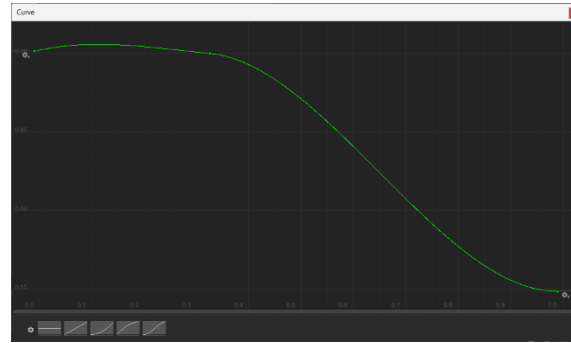


الشكل 25 منحنى التحكم المستخدم في حساب الاحتكاك بدلالة السرعة

- **قوة التسارع (Acceleration):** هي قوة محرك العربة ويتم حسابها من العلاقة التالية:

$$\vec{A} = -\overrightarrow{\text{acceleration direction}} \times \text{accelerate}(v/v_{max}) \times \text{throttle} \times v_{max}$$

عن طريق تابع $accelerate(v/v_{max})$ تم إنشاؤه ليحسب تسارع العربة اعتماداً على سرعتها الحالية v والسرعة العظمى v_{max} بحيث تتسارع العربة حتى تتقارب من السرعة العظمى وعندها تخف قدرة العربة على التسارع وضرب قوة التسارع بمقدار التسارع المطبق على العربة (من قبل المستخدم أو من قبل نظام القيادة الذاتية) وهو $throttle$. وبني التابع باستخدام منحني تحريك يوضحه الشكل 26.



الشكل 26 منحني التحريك المستخدم في حساب معامل التسارع بدلالة السرعة مقسومة على السرعة العظمى

- **قوة المكابح (Breaks):** هي قوة مرتبطة بقوة التسارع بشكل مباشر أيضاً ويتم حسابها بالعلاقة التالية:

$$\vec{B} = -\overrightarrow{acceleration\ direction} \times Breaks \times f_{breaks}$$

حيث f_{breaks} هو معامل يعبر عن مقدار قوة الكبح في العربة و Breaks هو مقدار الكبح المطبق عليها من قبل السائق أو منظومة القيادة الآلية.

- **قوة التوجيه (Steering):** تم اعتماد نموذج توجيه آكرمان² وتطبيق مقدار دوران على العجلات للحصول على قيم للزوايا الداخلية والخارجية توافق القيم المحسوبة في الدراسة النظرية.

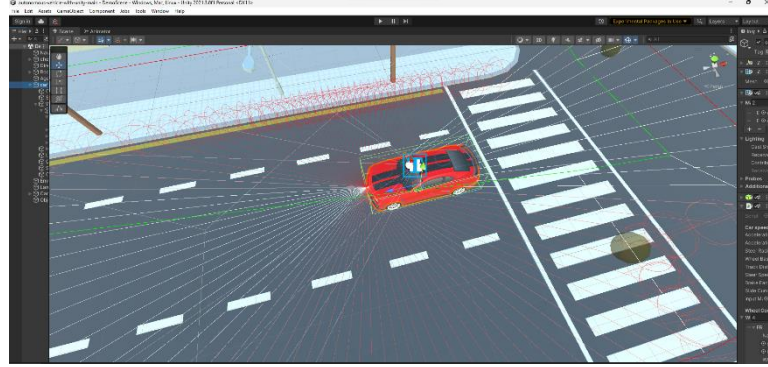
4.2.1.2 محاكاة الحساسات المرتبطة بالعربة

تم تزويد العربة بحساس Lidar افتراضي وكاميرا تلتقط الصور ومثبتة في مقدمة العربة. وهذا الحساس مرتبط بأحد أغراض المشهد ضمن السيارة ويعمل عن طريق إرسال أشعة افتراضية من العربة وبزايا مختلفة على محيط العربة وتم اختيار إرسال 50 شعاعاً للأمام و 50 شعاعاً للخلف والزوايا بين كل شعاعين متتاليين هي θ

$$\theta = \frac{360}{100} = 3.6^\circ$$

² المفصلة في (30)

بالإضافة لشعاع إضافي منطلق من مقدمة العربة يجعل العدد الكلي 101 شعاعاً، وهو عدد كافٍ لاستطلاع البيئة المحيطة. وتم اختيار مسافة 20 متر كالمسافة العظمى للحساس لضمان تتبع جيد للأغراض القريبة. ويوضح الشكل 27 الحساس المذكور.



الشكل 27 حساس Lidar الافتراضي المرتبط بالعربة

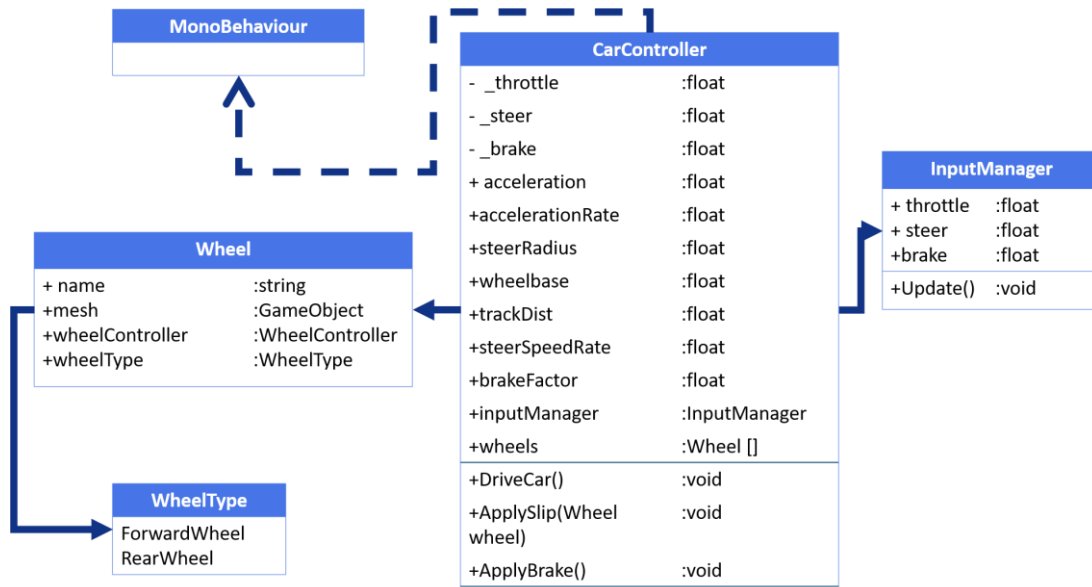
وأما بالنسبة للبيانات التصويرية فتم تجميعها بواسطة كاميرا مثبتة في مقدمة العربة وكاميرا ترافق العربة من الأعلى ويتم الانتقال بين المنظورين بشكل سلس بالاستعانة بمركبات cinemachine كما تم وضع خريطة مصغرة لمراقبة العربة أثناء التدريب من منظور كاميرا المراقبة يوضحها الشكل 28 في الزاوية العليا اليمنى.



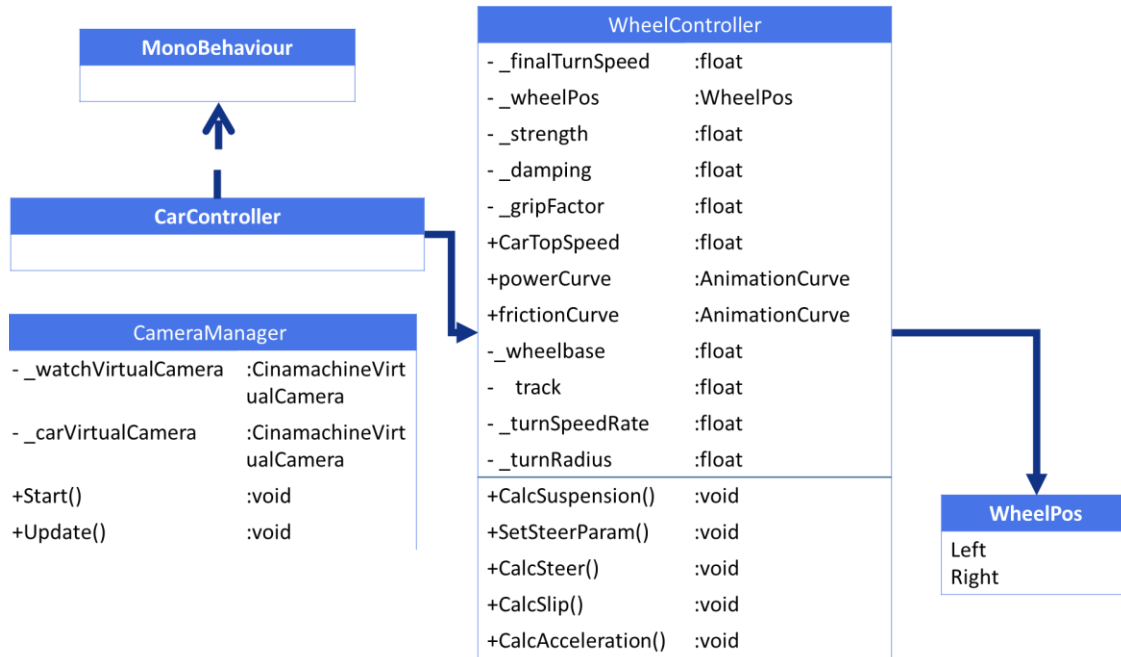
الشكل 28 منظور كاميرا العربة وكاميرا المراقبة

4.2.1.3 هيكلية الصفوف البرمجية المرتبطة بالعربة

يوضح الشكل 29 والشكل 30 أدناه بنية الصفوف البرمجية المرتبطة بالعربة.



الشكل 29 مخطط الصفوف الخاص بالصف CarController والمكونات المرتبطة به



الشكل 30 مخطط الصفوف الخاص بالصفوف المرتبطة بالعجلة

- **الصف CarController:** هو الصف المسؤول عن الربط بين الدخول (من المستخدم أو من القيادة الآلية) وبين العجلات والتحكم بها في الصف WheelController لتأمين المدخلات اللازمة لحساب القوى من خلال التتابع ApplyBrake() و ApplySlip() وقيادة العربة من خلال التابع DriveCar().

- **الصف WheelController:** هو الصف المسؤول عن التحكم بالعجلات وتطبيق القوى عليها من خلال التوابع CalcSteer() و CalcSlip() و CalcSuspension() و CalcAcceleration().
- **الصف InputManager:** هو الصف المسؤول عن تلقي الدخل في حالة التحكم من المستخدم وتوفير التحكم العمودي throttle والعرضاني steer والمكابح brakes.
- **الصف CameraManager:** هو الصف المسؤول عن التحكم بمنظور الكاميرا والتنقل بين منظور العربة ومنظور المراقبة في حال القيادة من دخل المستخدم.

4.2.2 إنشاء محاكاة المدينة

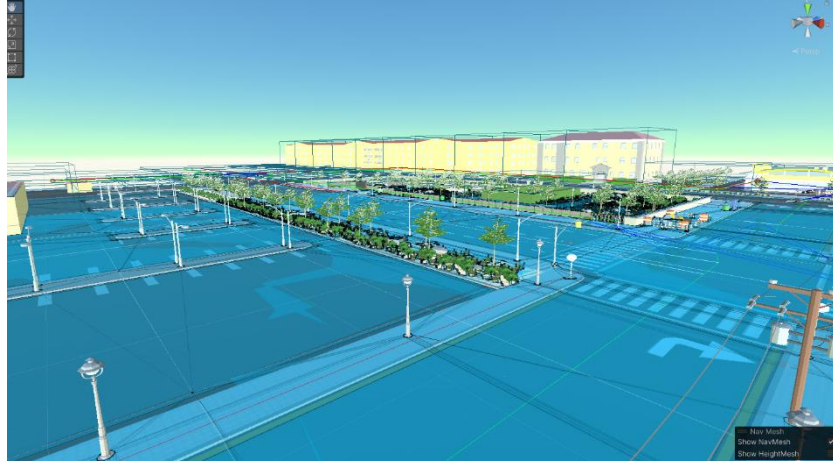
تمت محاكاة المدينة ضمن محرك يونيتي لتشمل مجموعة متنوعة من الطرقات والأرصفة والتقاطعات والأبنية وإشارات المرور ومحددات الطرق. ويوضح الشكل 29 المدينة ضمن المشهد في يونيتي.



الشكل 31 المدينة ضمن مشهد يونيتي

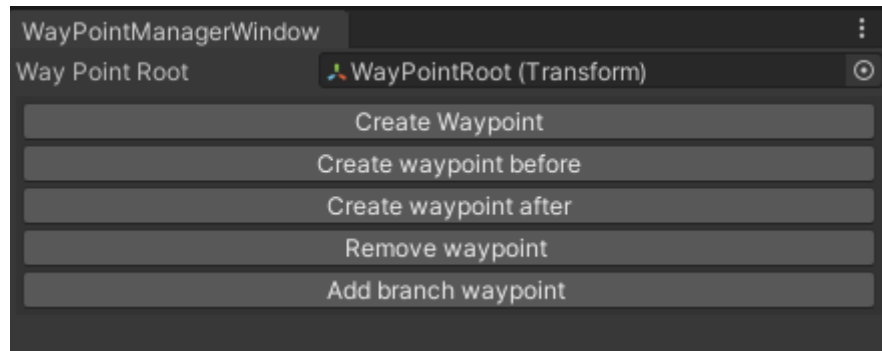
4.2.3 إنشاء محاكاة النظام المروري

يتألف النظام المروري المنشأ من نوعين من الكائنات هي العربات والمارة. وتم إنجازه على شقين، الأول هو تحديد المناطق القابلة للسير والذي تم بالاستعانة بمكونات شبكة الملاحة NavigationMesh ويوضح الشكل 32 الصورة أدناه شبكة الملاحة ضمن المدينة.



الشكل 32 شبكة الملاحة ضمن المدينة

بعد تحديد المناطق القابلة للسير يتوجب تحديد وجهات للعربات والمشاة للسير إليها. وتم ذلك من خلال تنجيز نظام نقاط علام waypoint system وذلك عن طريق إنشاء واجهة برمجية تسمح بإضافة نقاط العلام ضمن الخريطة وحذفها وإظهارها ضمن المشهد، كما تمت إضافة خواص للتفرع branching لزيادة مقدار العشوائية في تحركات الكائنات حيث تسمح لهم باتخاذ أي اتجاه بشكل عشوائي في حالة وجود تقاطعات بين الطرق. ويوضح الشكل 33 الواجهة المنجزة.



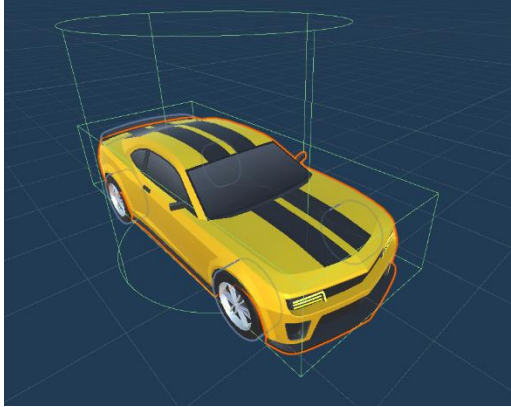
الشكل 33 واجهة التحكم بنقاط العلام

وقمت إضافة نقاط العلام ضمن المدينة باستعمال الأداة المنجزة كما يوضحه الشكل 34.



الشكل 34 نقاط العلام ضمن المدينة

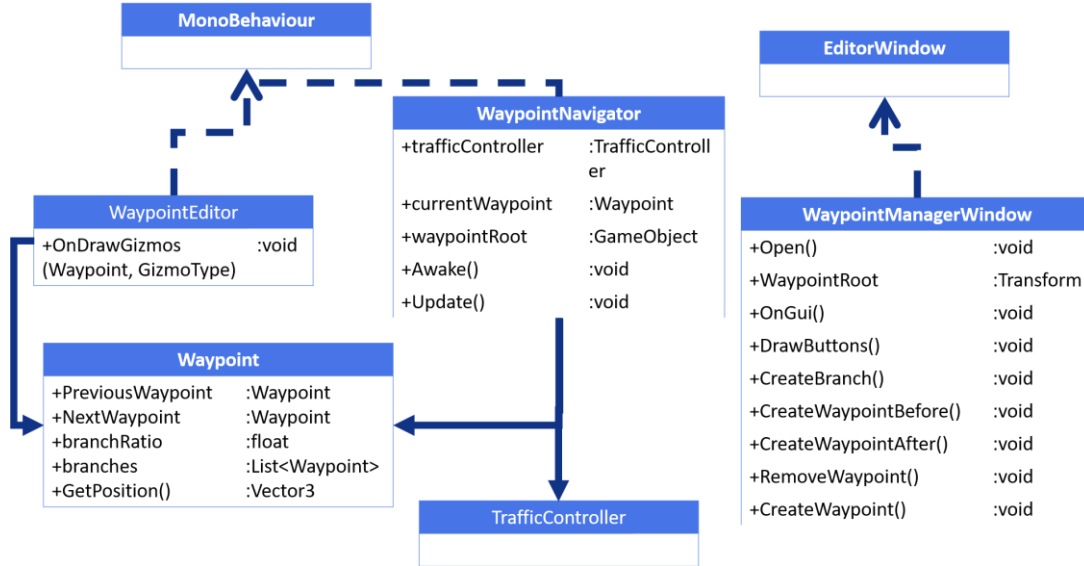
وتم تنجيز منظومة تسمح بتوليد الكائنات المستخدمة في النظام المروري وفق النمط التصميمي object pooling للتقليل من العبء الناتج عن توليد وتدمير الأغراض بشكل مستمر، حيث مُنِحت لكل كائن سرعة وتوجه وخيارات عشوائية للتفرع في المسارات بهدف تحقيق أكبر كم ممكن من الواقعية. وطُبقت كل من منظومة توليد الكائنات وتحديد نقاط العلام وتحديد المسارات على كل من العربات والمشاة. حيث تمت إضافة شخصيات مشاة وتزويدها بتحريك animation للمشاة ويوضح الشكل 35 أدناه شخصيات المشاة والسيارات العابرة.



الشكل 35 المشاة والعربات في النظام المروري

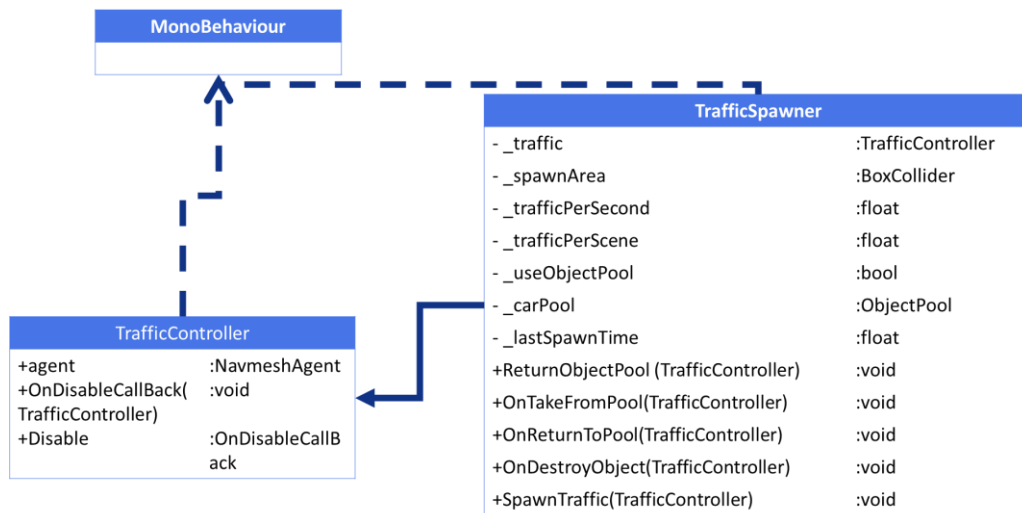
4.2.3.1 هيكلية الصفوف المرتبطة بنظام نقاط العلام

يوضح الشكل 36 هيكلية الصفوف المرتبطة بنقاط العلام.



الشكل 36 الصفوف المرتبطة بنقاط العلام

- **الصف Waypoint:** هو الصف المسؤول عن تعريف نقطة العلام كعقدة ضمن بيان ترتبط بالعقدة السابقة PreviousWaypoint واللاحقة NextWaypoint وبمجموعة الأفرع branches المستخدمة للتفرع في التقاطعات و branchRatio وهي نسبة احتمال اختيار الكائن القيام بالتفرع عند الاختيار العشوائي، وأما التابع GetPosition() فهو تابع يستخدم عند الربط مع نظام توليد الكائنات والسير باتجاه موقع النقطة.
- **الصف WaypointEditor:** هو الصف المسؤول عن رسم نقاط العلام ضمن المشهد على شكل بيان كما هو موضح في الشكل 34 حيث ترسم الوصلات بين العقد والعقد السابقة واللاحقة والمتفرعة ويتم إضاءة العقد الفعالة ورسم عرض كل عقدة على الخريطة. ينجز هذا الصف الواجهة EditorWindow.
- **الصف WaypointManagerWindow:** هو الصف المسؤول عن تعريف واجهة التحكم بنقاط علام ويحوي التوابع المستخدمة في إنشاء الواجهة ضمن Open() و DrawButtons() وفي إنشاء وحذف وتعديل نقاط العلام والأفرع.
- **الصف WaypointNavigator:** هو الصف المسؤول عن الملاحظة بين نقاط العلام عن طريق تحديد وجهة الكائنات اعتماداً على الاتجاه الذي تسير فيه والتفرع في حال وجود أفرع متوفرة. وأما فيما يتعلق بتوليد الكائنات فيوضح الشكل 37 بنية الصفوف المرتبطة به.



الشكل 37 الصفوف المرتبطة بتوليد الازدحام

- **الصف TrafficController:** هو المسؤول عن ربط مكونات توليد الازدحام بوكيل شبكة الملاحظة.
- **الصف TrafficSpawner:** هو الصف المسؤول عن توليد الكائنات ويتيح خيار استعمال ObjectPooling في التوليد عن طريق التتابع الموضحة في الشكل 37، ويتم ضمن التابع SpawnTraffic توليد كائنات ضمن منطقة يتم تحديدها على الخريطة عن طريق المكون _spawnArea وبحسب عدد الكائنات المطلوبة في المشهد ومعدل إنشائها. وبهذا تم إنجاز النظام المروري وبيئة المحاكاة.

4.3 ربط بيئة المحاكاة مع بيئة التدريب من خلال ML-Agents

تم استعمال الإصدار 20 من مكتبة ML-Agents وهو آخر إصدار متوفر حتى تاريخ إنجاز المشروع.

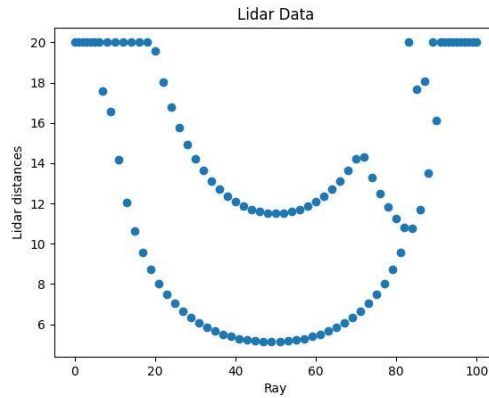
4.3.1 إنشاء البيئة الافتراضية

تم إنشاء بيئة افتراضية ضمن anaconda باسم mlagents-r20-v1 واستعمالها لتنصيب المكاتب اللازمة ضمن خوارزمية التعلم. وتم ربط البيئة الافتراضية مع خوارزمية التعلم ضمن برنامج Jupyter عن طريق إنشاء نواة جديدة (kernel) واستعمال البيئة الافتراضية كمسار تنفيذ رماز بايثون ضمن النواة.

كما تم ذكره سابقاً فإن تنجيز خوارزميات تعلم مخصصة غير متوفر ضمن الخيارات التي تقدمها مكتبة ML-Agents ولذلك تم بناء standalone build من المحاكاة لكل من أنظمة التشغيل Windows, Linux, Mac واستعمال واجهة تخاطب منخفضة المستوى (PLL-API) في فتح قناة تواصل مع المحاكاة. تمكن هذه القناة من إرسال واستقبال الرسائل بين Unity و Anaconda.

4.3.2 الحساسات المستخدمة (Sensors)

تم استطلاع البيئة عن طريق إرسال خرج حساس Lidar الموضح في الشكل 38 وخرج الكاميرا الموضح في الشكل 39.



الشكل 38 المسافات المقاسة بأشعة حساس Lidar بدلالة الشعاع

وباعتبار ترقيم الأشعة يبدأ بالشعاع الواقع في مقدمة العربة ويتناوب على الشعاع الأيمن والأيسر بترتيب مماثل من حيث البعد عن مقدمة العربة (مثلاً 0,1,-1,2,-2) فإن المخطط يدل على وجود جدارين على جانبي العربة وطريق خالٍ من الأمام والخلف حيث نلاحظ على المحور x (محور أشعة الليدار) أن أول وآخر الأشعة لم تصطدم بأي غرض وقاست أعظم مسافة وهي 20 متراً.



الشكل 39 البيانات الصورية من يونيتي والمرسلة إلى بايثون

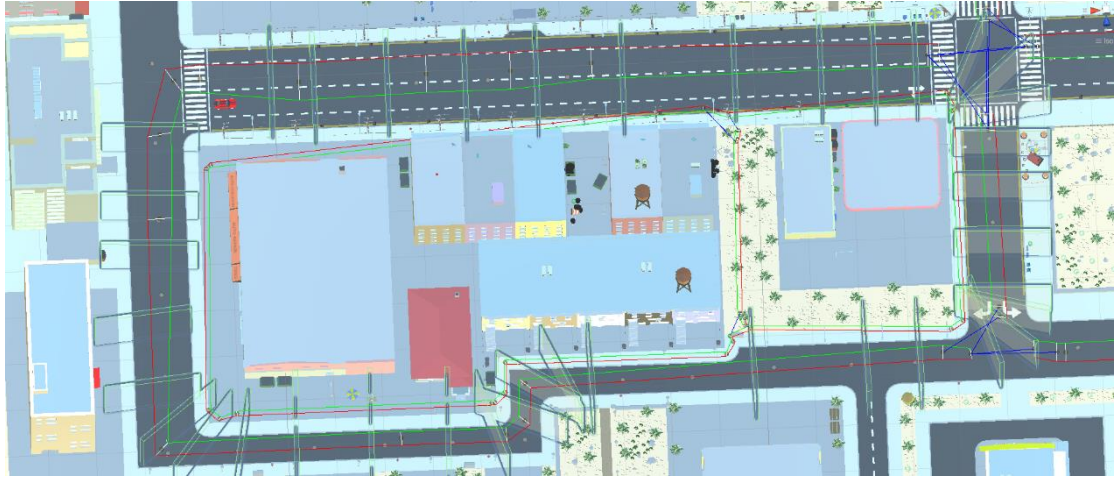
وتم اختيار زاوية الكاميرا على العربة لتتوافق مع الزوايا المستخدمة في قاعدة البيانات CuLane لاستخدامها لاحقاً في معالجة الصور. وبهذا تم ربط الحساسات المناسبة بالعربة ضمن المحاكاة.

4.3.3 توطين بيئة التعلم ضمن المشهد في يونيتي

تم ذلك عبر خطوتين، الأولى هي إنجاز منظومة نقاط مراقبة لتسجيل تقدم العربة على المسار وإجراء كشف عن الاصطدام للاستفادة منه في تجنب العوائق والثانية هي تنجيز البرمجيات الخاصة بإرسال واستقبال البيانات من وكيل التعلم.

4.3.3.1 منظومة نقاط المراقبة Checkpoints وكشف الاصطدام

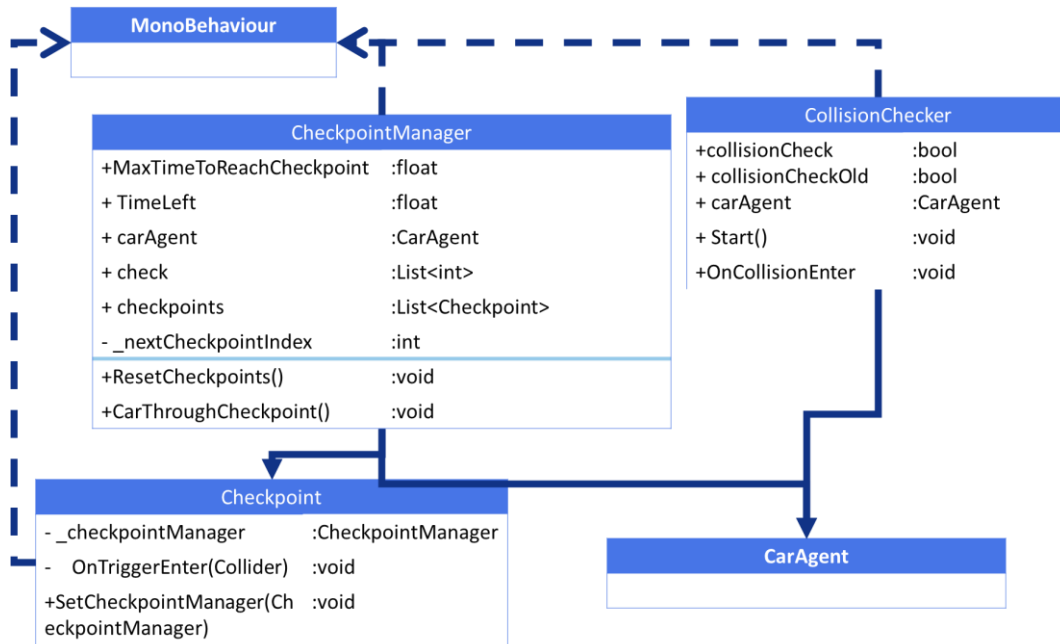
عند تدريب العربة على القيادة ضمن المشهد سيتحتم مكافأتها على الأفعال الجيدة ومعاقبتها على الأفعال السيئة ولتحقيق ذلك برمجياً ضمن يونيتي تم تنجيز منظومة نقاط مراقبة checkpoints يمكن إضافتها ضمن المشهد كما يوضح الشكل بحيث تستطيع اكتشاف العربة عند مرورها من أي من نقاط المراقبة. وأما كشف الاصطدام مع العوائق فتم عن طريق استخدام مصادمات الأجسام المرتبطة بالعربة والبيئة في إشعار حدوث تصادم.



الشكل 40 توضع نقاط المراقبة على الخريطة في المدينة

4.3.3.2 الصفوف المرتبطة بنقاط المراقبة وكشف التصادم

يوضح الشكل 41 الصفوف المرتبطة بنقاط المراقبة وكشف التصادم.



الشكل 41 الصفوف المرتبطة بنقاط المراقبة وكشف التصادم

- **الصف Checkpoint:** هو الصف المسؤول عن تعريف نقطة مراقبة ويرتبط بكل نقطة مراقبة نسخة منه. ويكشف هذا الصف عبور العربة من إحدى النقاط بإرسال إشعار للصف CheckpointManager عند التابع OnTriggerEnter().

- **الصف CheckpointManager:** هو الصف المسؤول عن إدارة نقاط المراقبة وتسجيل تقدم العربة ضمن المسار بالاتجاه الصحيح ضمن التابع CarThroughCheckpoint وهو المسؤول عن إعادة ضبط النقاط عند بداية كل Episode ضمن التابع ResetCheckpoints.
- **الصف CollisionChecker:** هو الصف المسؤول عن إرسال إشعار لوكيل التعلم عند الاصطدام بعائق.

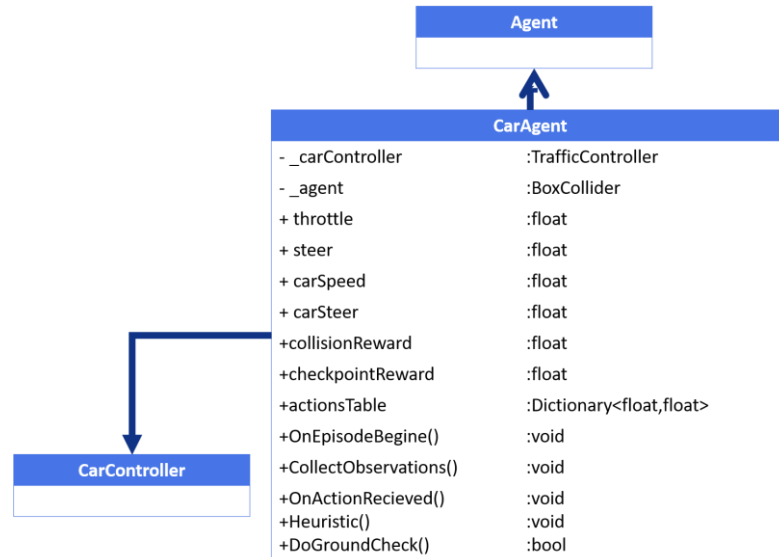
4.3.3.3 تهيئة وكيل التعلم

تمت تهيئة وكيل التعلم عن طريق إضافة مكونين إلى العربة وهما مكونان تتيحهما مكتبة ML-Agents وهما المكون BehaviourParameters و DecisionRequester . وتم تحديد عدد الحساسات المستخدمة عن طريق BehaviourParameters وتحديد عدد الأفعال المطلوبة. وأما DecisionRequester فيستخدم عند استلام الرسائل من بايثون في كل خطوة عن طريق إرسال إشعار جاهزية للاستلام. وتم تحديد فترة الاستلام كل 5 خطوات بناءً على التجريب.

تم أيضاً بناء صف وكيل تعلم خاص بالعربة وتنجيته لاستقبال الأفعال وإرسال المراقبات.

4.3.3.4 الصف CarAgent

يوضح الشكل 42 بنية الصف CarAgent.



الشكل 42 بنية الصف CarAgent

مهمة هذا الصف هي إرسال مراقبات البيئة إلى بايثون عبر التابع CollectObservation واستقبال الأفعال منه في التابع OnActionRecieved حيث تمت مقابلة كل فعل مع قيم للتحكم العمودي والعرضي ضمن الفهرس

actionsTable وتتم إرسال قيم دالة الربح التي يتلقاها الوكيل إضافة إلى أنه تم إرسال قيمتين هما سرعة العربة والسرعة الزاوية في كل خطوة مع المراقبات ليتم الاستفادة منها لاحقاً في تحليل النتائج وتطور سرعة الوكيل مع التعلم.

تم تطبيق التابع Heuristic للاستفادة منه في وضع inference mode من أجل قيادة العربة من قبل المستخدم (دون تدريب). وأخيراً ضمن التابع DoGroundCheck يتم التأكد من استقرار العربة على الطريق للاستفادة من ذلك في دالة الربح لاحقاً. وبهذا يكون قد أتم الربط بين بيئة المحاكاة وبيئة التعلم.

4.4 توطين خوارزميات الرؤية الحاسوبية ضمن بيئة التدريب

تم اعتماد خوارزمية CLNet المفصلة في الدراسة المرجعية لتفوقها الواضح في كشف المحددات حيث تعد أحدث خوارزمية متوفرة حتى تاريخ المشروع وأما بالنسبة لكشف الأغراض في المشهد فتم اعتماد خوارزمية YOLOv8 لتتأجها الممتازة والمتطورة.

أتاح ناشرو الورقة البحثية [4] مجموعة من النماذج المدربة لخوارزمية CLNet على بيانات من القاعدة CuLane والتي تم اعتمادها في كشف المحددات ضمن المشهد. وبسبب حداثة هذه الخوارزمية لا تتوفر تنجييزات رسمية لاستخدام الخوارزمية في كشف المحددات على صورة، ولهذا تم تحويل النموذج المدرب إلى صيغة onnx واستخلاص المعلومات المفيدة منه لتطبيقها على الصور في كل إطار. للتمكن من استعمال الخوارزمية تم استخدام نسخة "cudaToolkit 12.1" ونسخة "torch 2.1.0.dev20230723+cu121" و "torchvision 0.16.0.dev20230723+cu121".

تعيد الخوارزمية سلسلة من المحددات لكل محدد مجموعة من النقاط يتم تحديد إحداثيات كل من فواصل وترتيب بكسلات المحدد. ويوضح الشكل 43 أدناه إظهار خرج CLNet على البيانات الصورية المستقبلية من يونيتي باستعمال opencv.



الشكل 43 كشف المحددات باستخدام **CLRNet** على البيانات الصورية المستقبلية من يونيتي

وتم تطبيق خوارزمية YOLOv8 على البيانات الصورية وخرج تطبيق الخوارزمية هو بنية من إحداثيات بكسلات حدود الغرض مع نوعه ورقم هويته ومعامل التكبير ونسبة وثوقية النتيجة. ويوضح الشكل 44 أدناه مثالاً على إظهار خرج الخوارزمية باستعمال opencv.



الشكل 44 كشف الأغراض باستخدام خوارزمية YOLOv8 على البيانات الصورية المستقبلية من يونيتي

وبهذا تم توطين خوارزميات الرؤية الحاسوبية ضمن محرك يونيتي بنجاح.

4.5 التحكم بالعربة باستخدام خوارزميات التعلم المعزز العميق

تم اقتراح حل يعتمد على تشكيل مسألة اتخاذ القرار المرحلية بنموذج ماركوف (MDP) خاصة بالبيئة المنشأة في هذا المشروع لحل مسألة التحكم عن طريق اتخاذ قرارات مرتبطة بحالة العربة. ومركبات هذه المسألة هي $\langle S, A, T, R, \gamma \rangle$ حيث S هي حالة مجموعة الحالات و A هي مجموعة الأفعال و T هي النموذج الذي

ينقل العربة لحالة جديدة و R هي دالة الربح و γ هي معامل التخميد. وتعرف هذه الخواص على الشكل التالي بالنسبة للعربة:

- **المراقبات (Observations):** يتم تحديد حالة البيئة من خلال المراقبات وستعتمد في هذا المشروع مراقبات حساس Lidar ونتائج تطبيق خوارزميات الرؤية الحاسوبية على البيانات الصورية كدخل.
- **الأفعال (Actions):** اعتمد التحكم العمودي (throttle) والعرضاني (steer) كأفعال التحكم ضمن المسألة، إلا أن التعامل مع مقادير مستمرة (كقيمة التسارع المطبق على العربة) يعد مكلفاً حسابياً وزمنياً ولأغراض هذا المشروع تم اعتماد تقطيع (Discretization) للأفعال على الشكل التالي:
 - تم تقطيع أفعال التحكم العمودي لخمسة قيم تعبر عن سرعة عالية وسرعة متوسطة في الاتجاهين للأمام والخلف وسرعة معدومة للوقوف في المكان (تحاكي هذه القيم دخل المستخدم في القيادة اليدوية وتدخل خوارزميات المحاكاة لتحديد السرعة الفعلية بناءً على القوى المطبقة).
 - تم تقطيع أفعال التحكم العرضاني لثلاث قيم تعبر عن التفاف بزاوية صغيرة ومتوسطة وعدم الالتفاف. وهذا يجعل العدد الكلي 15 فعلاً.
- **دالة الربح (Reward):** إن الغاية من تدريب النموذج هو تعليم العربة القيادة ضمن حدود الطريق وعدم الاصطدام بالعقبات والتحرك بسرعة واستقرار. ولتحقيق هذه الغاية تم اعتماد دالة ربح تعتمد على منظومة نقاط المراقبة checkpoints التي تم وضعها على الخريطة ويتراكم الربح في حال التقدم باتجاه الهدف ضمن r_{ch} ويعاقب وكيل التعلم في حال الاصطدام بالعقبات والبيئة المحيطة عن طريق r_{col} ، ويعاقب أيضاً بشكل صغير في كل خطوة لتحفيزها على الاستمرار وعدم الوقوف في r_p وتعاقب على انفصال العربة عن الطريق لتحفيز الاستقرار في r_g .
- وتم إنشاء دالة ربح r_l ، خاصة بالتعلم مع بيانات صورية تعتمد الدالة المقترحة خرج خوارزمية CLNet في إجراء LaneKeeping يهدف لبقاء العربة ضمن حدود المحدد ولذلك أضيف إلى دالة الربح مقدار خسارة يعبر عن حالة عدم ظهور المحدد في المشهد أو عدم تساوي المقدار الظاهر منه بين كل من المحددين (بما يعبر عن عدم وجود السيارة بشكل متوازن ضمن المحددات).

وبذلك تعطى دالة الربح المقترحة بالشكل التالي:

$$r_{ch} = \begin{cases} r_{checkpoint}, & \text{If vehicle reaches a checkpoint} \\ 0, & \text{Otherwise} \end{cases}$$

$$r_t = \begin{cases} r_{task\ completion}, & \text{If vehicle successfully finishes the task} \\ 0, & \text{Otherwise} \end{cases}$$

$$\begin{aligned}
r_{col} &= \begin{cases} -r_{collision}, & \text{If vehicle hits a wall or obstacle} \\ 0, & \text{Otherwise} \end{cases} \\
r_p &= \begin{cases} -r_{steady\ penalty}, & \text{Steady penalty each step} \end{cases} \\
r_g &= \begin{cases} -r_{groundCheck}, & \text{If vehicle disconnects from the road} \\ 0, & \text{Otherwise} \end{cases} \\
r_l &= \begin{cases} -r_{maxLanes}, & \text{If no lanes detected} \\ -r_{lanes}, & \text{different lane sizes detected} \end{cases}
\end{aligned}$$

وهذا يجعل الربح الكلي:

$$r = r_{ch} + r_t + r_{col} + r_g + r_p + r_l$$

وتم اعتماد خوارزمية DQN بناءً على ملائمتها للمسألة ونتائجها الجيدة والتي تمت مناقشتها في الدراسة المرجعية وتم تدريب نموذج باستخدام مراقبات من دخل حساس Lidar بشكل منفصل ونموذج لحساس Lidar مع نتائج خوارزمية CLNet ستم مناقشة بنية كل من النموذجين.

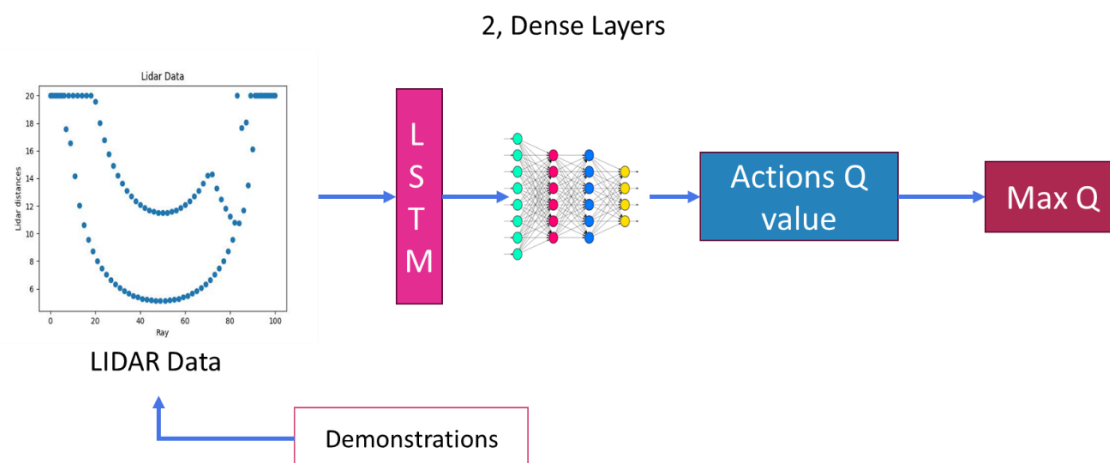
4.5.1 تدريب خوارزمية DQN على دخل من حساس Lidar

تم اعتماد البارامترات التالية في تدريب الخوارزمية من دخل حساس Lidar:

$r_{checkpoint}$	10
$r_{task\ completion}$	100
$r_{collision}$	-10
$r_{steady\ penalty}$	-0.0005
$r_{groundCheck}$	-10
r_{lanes}	0
$r_{maxLanes}$	0

الجدول 2 بارامترات التدريب لدالة الربح على دخل من حساس Lidar

ويوضح الشكل 45 بنية الخوارزمية المقترحة.



الشكل 45 بنية DQN المقترحة على دخل من حساس Lidar

4.5.2 تدريب خوارزمية DQN على دخل حساس Lidar ومحددات الطرق وفق

خوارزمية CLRNet

تم اعتماد البارامترات التالية في تدريب الخوارزمية من دخل حساس Lidar ومحددات الطرق وفق

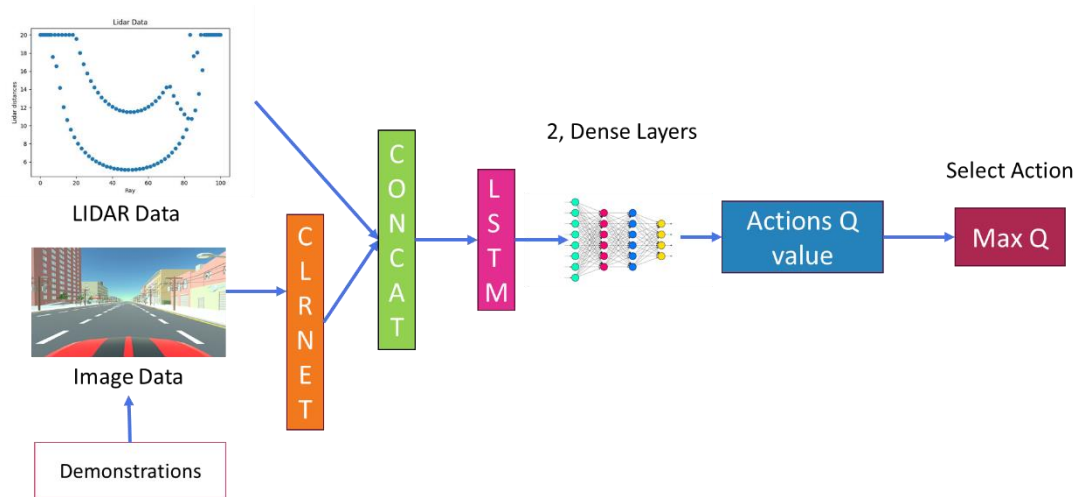
خوارزمية CLRNet

$r_{checkpoint}$	10
$r_{task\ completion}$	100
$r_{collision}$	-10
$r_{steady\ penalty}$	-0.0005
$r_{groundCheck}$	-10
r_{lanes}	$-0.000005 \times (points(lane1) - points(lane2))$
$r_{maxLanes}$	-0.000005×50

الجدول 3 بارامترات التدريب لدالة الربح على دخل من حساس Lidar ومحددات الطرق وفق خوارزمية

CLRNet

حيث $points(lane)$ هي عدد النقاط المكتشفة من كل محدد والفرق بينها يدل على توازن العربة بين المحددات. ويوضح الشكل 46 أدناه بنية الخوارزمية المتبعة.



الشكل 46 بنية DQN المقترحة على دخل من حساس Lidar ومحددات الطرق وفق خوارزمية CLRNNet وتم في كل من الخوارزميتين المنجزتين في (4.5.1) و(4.5.2) اعتماد المعاملات الفائقة (hyperparameters) الموضحة في الجدول 4.

المعاملات الفائقة	تابع التفعيل	نوع الشبكة	البيانات
Time steps=4 #of cell states=256	-	LSTM	حساس Lidar #data =202
# of units=512	ReLu	Fully Connected	
Time steps=4 #of cell states=256	-	LSTM	حساس Lidar مع بيانات CLRNNet #data=207
# of units=512	ReLu	Fully Connected	

الجدول 4 المعاملات الفائقة المستخدمة في شبكة DQN المقترحة

4.5.3 التعلم من بيانات إيضاحية (Demonstration Learning)

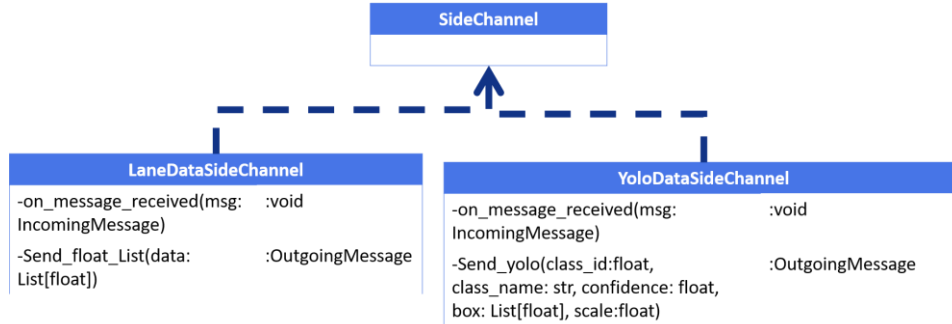
توضح الدراسات المرجعية على الفوائد الكبيرة التي تمت مناقشتها للتعلم من بيانات إيضاحية في تحسين أداء DQN وتسريعه. ولتحقيق ذلك تم تسجيل بيانات إيضاحية عن طريق قيادة العربة من قبل المستخدم بالاستفادة من التابع Heuristic ضمن الصف CarAgent، حيث تم تسجيل الأفعال المختارة من قبل السائق وقيم الربح والمراقبات المعبرة عن حالة البيئة الموافقة لكل فعل يتم اختياره. وتم إضافة هذه التجارب إلى تكرار التجربة (Experience replay) الخاص بخوارزمية DQN ليتم التعلم عليها قبل البدء بعملية استكشاف البيئة.

4.6 إنشاء واجهات المستخدم

يحتاج عرض البيانات ضمن الواجهات قسمين الأول هو إرسال بيانات العرض من بايثون والثاني هو إنشاء أساليب العرض المناسبة في يونيتي. بالنسبة للقسم الأول، نظراً لأن الحجم المتاح لتراسل البيانات ضمن المراقبة والأفعال محدود. تم إنشاء قنوات ربط جانبية custom side channels لإرسال معلومات كل من خرج خوارزميتي CLNet و YOLO. وأما للقسم الثاني، فتم إنشاء الواجهات البرمجية المناسبة لعرضها من جهة يونيتي.

4.6.1 إنشاء قناة ربط جانبية لنقل معلومات العرض Custom Side Channel

توفر ML-Agents بعض قنوات الربط الجاهزة لأنماط بيانات محددة، إلا أن بيانات الخرج المراد عرضها لاتقع ضمن الأنواع المشمولة، ولذلك تم إنشاء قناتي ربط مخصصتين لكل من CLNet و YOLO. بالنسبة للمحددات فتم إرسال 4 نقاط على شكل إستيفاء من كل محدد يتم إكتشافه إضافة لإرسال عدد المحددات الكلي عن طريق قناة سميت LaneDataChannel وتفعيلها عند إنشاء البيئة. وكذلك الأمر بالنسبة إلى YoloDataChannel التي تم عبرها إرسال كل من هوية واسم ودرجة وثوقية وحجم الأغراض المكتشفة إضافة إلى الإحداثيات الأربع لزوايا الصندوق المحيط به. ويوضح الشكل 47 بنية هذه الصفوف ضمن بايثون.



الشكل 47 صفوف القنوات الجانبية المخصصة ضمن بايثون

4.6.2 استقبال معلومات العرض ضمن يونيتي

تم استقبال معلومات العرض ضمن يونيتي باستخدام هوية القنوات المنشأة في بايثون واستعمال هذه البيانات في عرض خطوط المحددات ورسم صناديق حول الأغراض الموضحة ضمن الشكل 48 والشكل 49. ونظراً لعدم توفر نسخة مجانية من OpenCV خاصة بيونيتي تم رسم الخطوط بالاستعانة بمكونات خطوط ثلاثية الأبعاد. ويظهر الشكلان عرض خطوط المحددات وصناديق الأغراض ضمن واجهة يونيتي.



الشكل 48 عرض خرج خوارزمية CLNet ضمن يونيتي

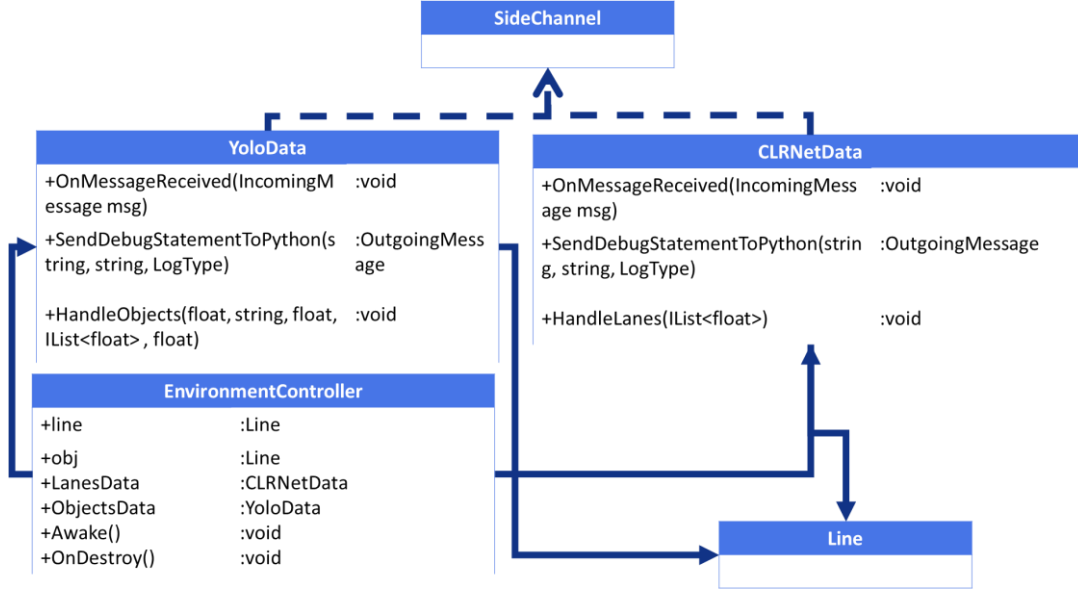


الشكل 49 عرض خرج خوارزمية YOLOv8 ضمن يونيتي

وتم عرض كل من سرعة العربة والسرعة الزاوية للالتفاف كما هو موضح في الزاوية العليا اليسرى.

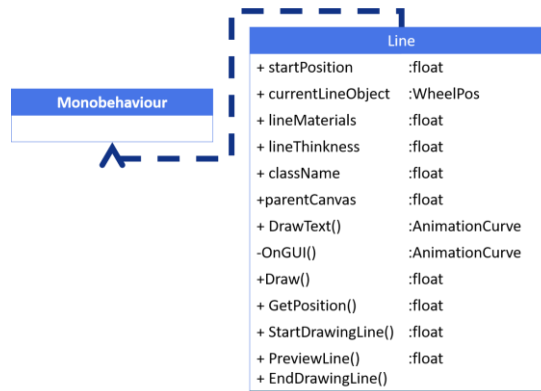
4.6.3 الصفوف المرتبطة بعرض المعلومات

في يونيتي، الصفوف المسؤولة عن عرض المعلومات العرض موضحة في الشكل 50 و 51.



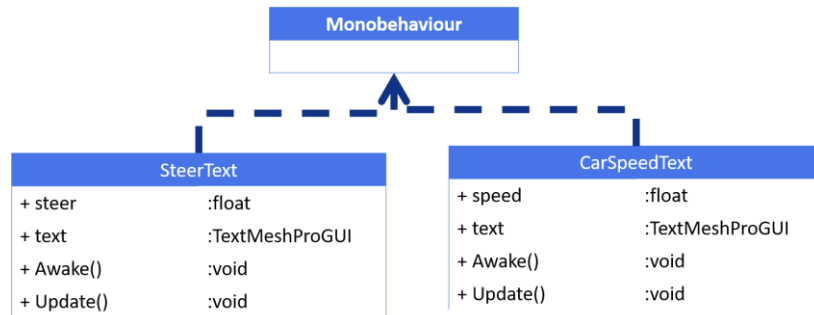
الشكل 50 الصفوف المتعلقة بإستقبال معلومات العرض

- **الصف YoloData**: الصف المسؤول عن إستقبال معلومات خرج خوارزمية Yolo وينجز الواجهة SideChannel. يتم ضمن التابع HandleObjects توجيه البيانات اللازمة للصف Line ليقوم برسم مربع حول الغرض المكتشف وكتابة اسمه ومعلوماته.
- **الصف CLNNetData**: هو الصف المسؤول عن إستقبال معلومات خرج CLNNet وينجز الواجهة SideChannel في يونيتي ويتم ضمن التابع HandleLanes إرسال المعلومات المناسبة للصف Line لرسم المحددات.
- **الصف EnvironmentController**: هو الصف المسؤول عن تفعيل القنوات الجانبية عند بدء المحاكاة وإلغاء تفعيلها عند الانتهاء.



الشكل 51 بنية الصف Line

- الصف Line: يقوم هذا الصف برسم خطوط عن طريق التابع Draw() الذي يأخذ بياناته عن طريق تحويل الإحداثيات من إحداثيات الصورة إلى إحداثيات في الفضاء ثلاثي الأبعاد وإسقاطها على الشاشة مجدداً ضمن التابع GetPosition ومن ثم يتم رسم خطوط بالاستعانة بمكون LineRenderer عن طريق تمرير الإحداثيات الجديدة له. يتم ضمن التابع DrawText كتابة اسم الغرض على مربعه عند الاكتشاف وهو خاص بخوارزمية Yolo.
- وأما الصفوف المتعلقة بكتابة سرعة العجلات والسرعة الزاوية فهي موضحة في الشكل 52.



الشكل 52 صفوف إظهار السرعة في الواجهة

- الصف SteerText: مسؤول عن عرضة سرعة الانعطاف الزاوية بوحدة rpm.
 - الصف CarSpeedText: مسؤول عن عرض السرعة بوحدة kph.
- وبهذا يكون قد تم إنجاز واجهات المستخدم التي تسمح باستعراض عملية التدريب بسهولة ووضوح.

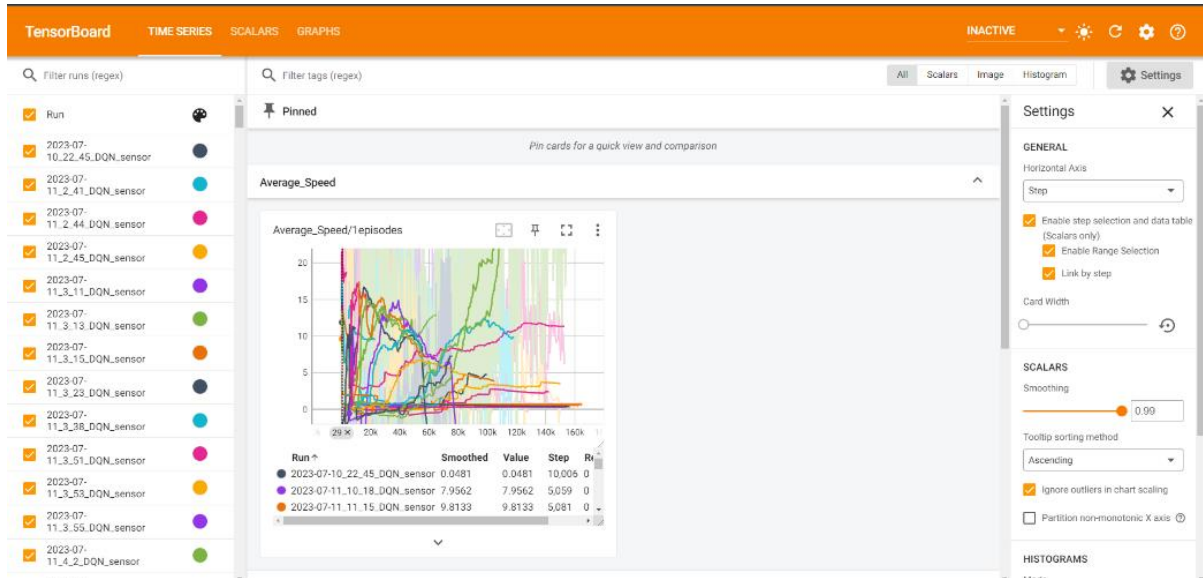
الفصل الخامس

النتائج العملية

يتضمن هذا الفصل النتائج المستخلصة من القسم العملي

5.1 أداة Tensorboard

تم ربط البيئة الافتراضية مع أداة Tensorboard لمراقبة النتائج أثناء التدريب وتحليل النموذج المدرب. وتم إرسال 4 قيم يتم تحديثها في كل خطوة إلى الأداة وهي مقدار الربح (Reward) والخسارة (Cost) والسرعة (Throttle) والانعطاف (Steer). وتم حفظ النموذج المدرب بصيغة checkpoint يمكن حفظها واستكمال التدريب عليها. ويمكن استعراض جميع نتائج التدريب بالانتقال إلى ملف savedNetworks حيث تحفظ كافة النماذج المدربة و الاستعراض عبر تعليمة `tensorboard --logdir = "/"` ويوضح الشكل 53 أدناه النتيجة.



الشكل 53 استعراض ومراقبة النتائج أثناء التدريب عبر Tensorboard

5.2 نتائج تطبيق خوارزمية DQN

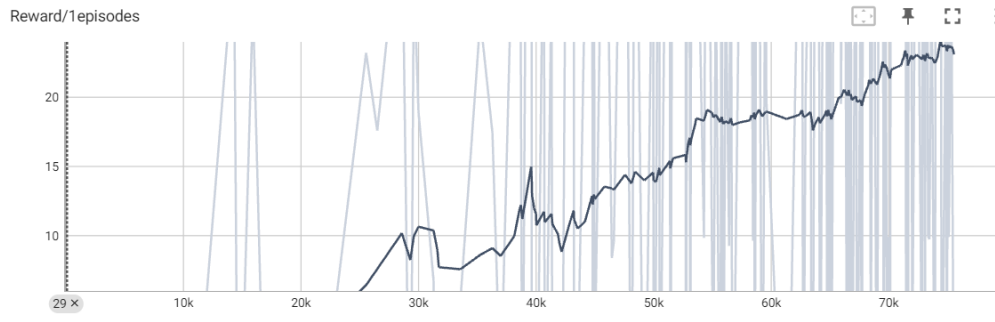
تم تدريب الخوارزمية على 150 ألف خطوة بحجم ذاكرة Replay memory قدره 100 ألف تجربة ومعدل تخميد للاستكشاف قدره $1/100000$. وكانت النتائج على الشكل التالي.

5.2.1 الحاجة للبيانات الإيضاحية

أوضحت التجارب على تدريب النموذج بشكل مباشر على استطلاع البيئة يستغرق وقتاً طويلاً جداً حتى يبدأ بالتعلم بشكل واضح. وتم إعزاء ذلك لأن البيئة فقيرة بمصادر المكافأة (Sparse Reward Environment) والمسألة معقدة لدرجة لا تسمح للوكيل باكتشاف الحل بسرعة ولذلك تم اقتراح اعتماد البيانات الإيضاحية في التدريب.

5.2.2 مشكلة النسيان (Catastrophic forgetting)

تم بداية اعتماد حجم ذاكرة قدره 10000 تجربة إلا أنه وبرغم تدريب العربة على السير بشكل صحيح أثناء التدريب، كانت نتائج مرحلة الاختبار بعيدة عن الأداء الذي قدمته أثناء التدريب. وتم استنتاج نسيان الوكيل لما تعلمه بسبب صغر حجم الذاكرة الذي أدى لأنه بعد تعلمه السياسات الناجحة لم يعد يحصل على أي مراقبات من الحالات العشوائية ضمن البيئة مما أدى لأنه لم يعد يعرف التصرف في الحالات العامة غير المثالية. ولحل هذه المشكلة تمت زيادة حجم الذاكرة. ويوضح الشكل 54 استقرار قيمة الربح عند سياسة غير أمثلية بسبب النسيان.



الشكل 54 مشكلة النسيان

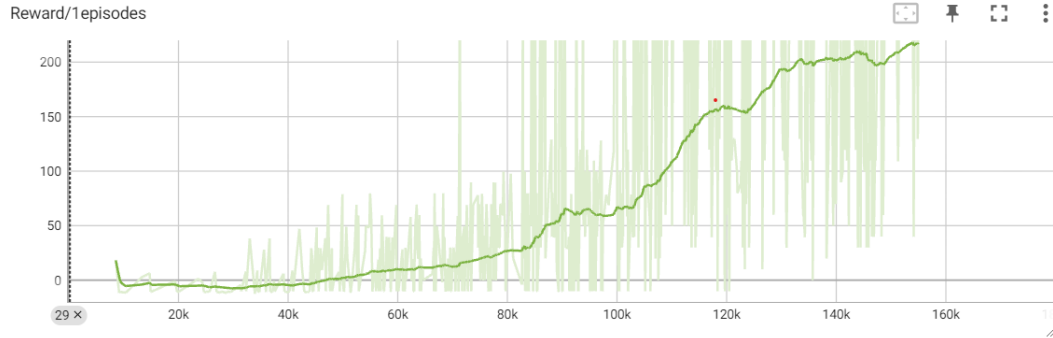
5.2.3 النتائج من أجل التدريب على دخل من حساس Lidar

تم اعتماد التدريب في التعلم بسبب تعقيد المسألة الكبير ويوضح الشكل 55 أدناه البيئة المصغرة التي تم التدريب عليها.



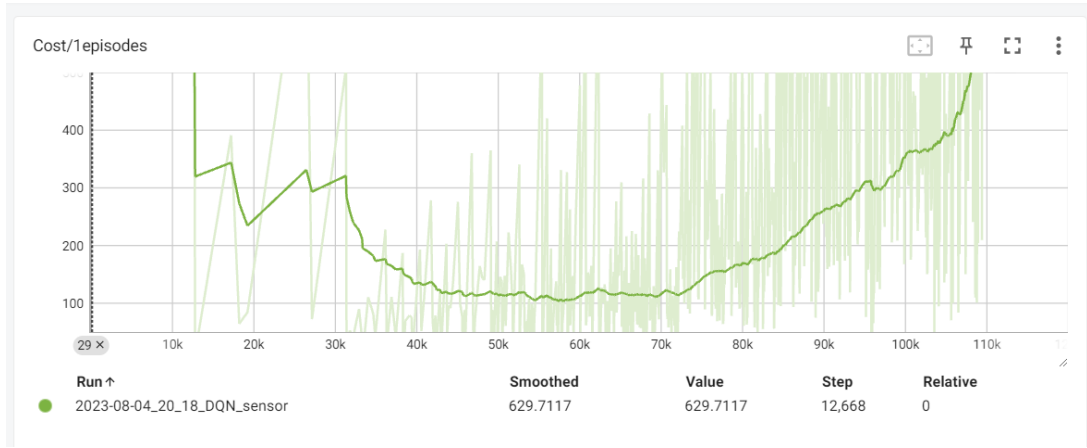
الشكل 55 البيئة المصغرة المستخدمة في التدريب

قامت العربة بإنجاز المهمة عند 120 ألف خطوة وتم الاستمرار بالتدريب حتى 150 ألف. وتوضح الأشكال أدناه كل من دالة الربح في الشكل 56 حيث نلاحظ استقرار الربح عند القيمة العظمى بعد 130 ألف خطوة.



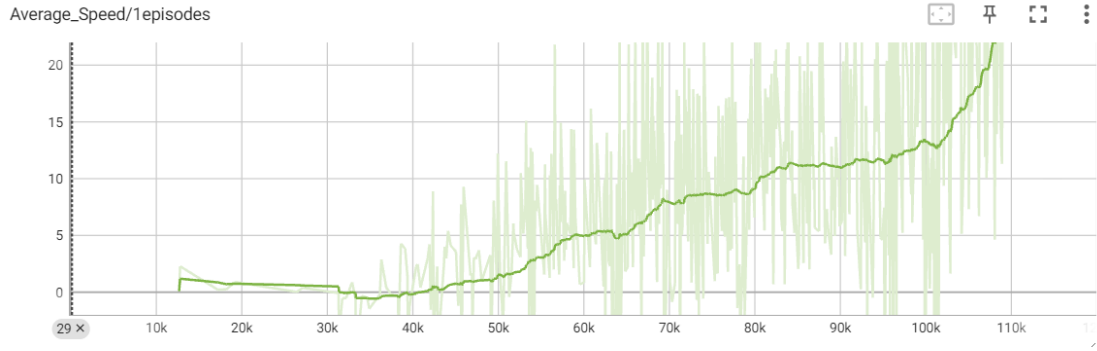
الشكل 56 الربح بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar

وفي الشكل 57 أدناه نلاحظ أن دالة الخسارة كانت تنخفض تدريجياً مع زيادة التعلم إلا أنه ومع زيادة التعلم بشكل كبير فإن دالة الخسارة تزداد ويرجع ذلك لمشاكل في استقرار خوارزمية DQN تتعلق بأنه مع زيادة التعلم يزداد طول الحلقة (لأن السيارة تقطع مسافة أكبر) وبذلك تتراكم الخسارة وأيضاً تصبح دالة الخسارة أكثر قسوة مع زيادة الخبرة لأن الشبكة المستخدمة في تحديد الأفعال الهدف هي نفسها المستخدمة في تحديد قيم الأفعال في كل خطوة.



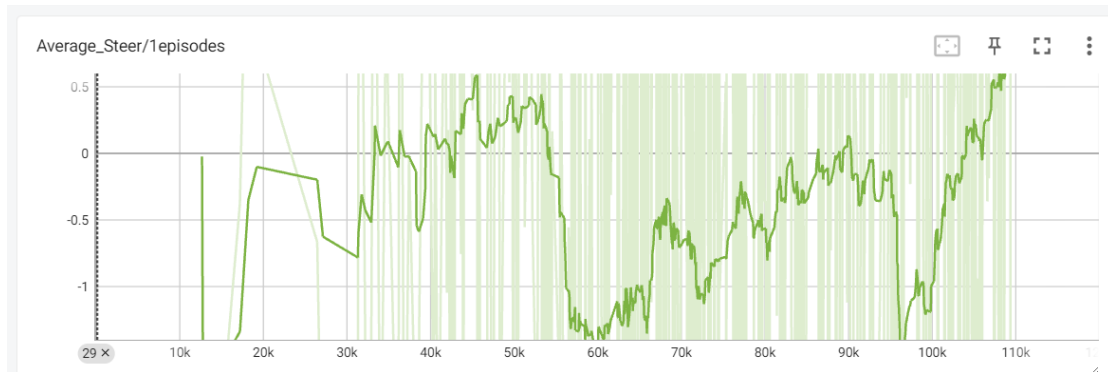
الشكل 57 دالة الخسارة بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar

وأما بالنسبة للسرعة فكما نلاحظ في الشكل 58 ، تزداد السرعة مع التعلم.



الشكل 58 سرعة العربة بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar

وأما سرعة الانعطاف الموضحة في الشكل 59 فتتأرجح حول الصفر مما يوضح تعلم سياسة السير ضمن خط مستقيم.

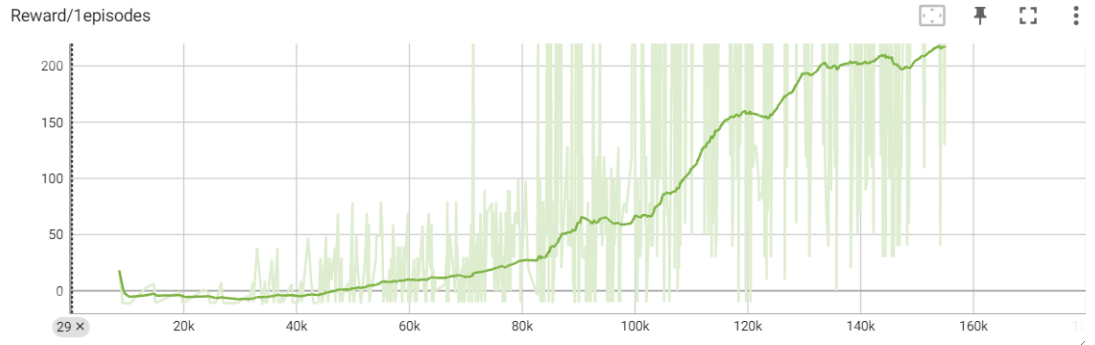


الشكل 59 سرعة الانعطاف بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar

5.2.4 النتائج من أجل التدريب على دخل من حساس Lidar ومحددات الطرق وفق

CLRNet

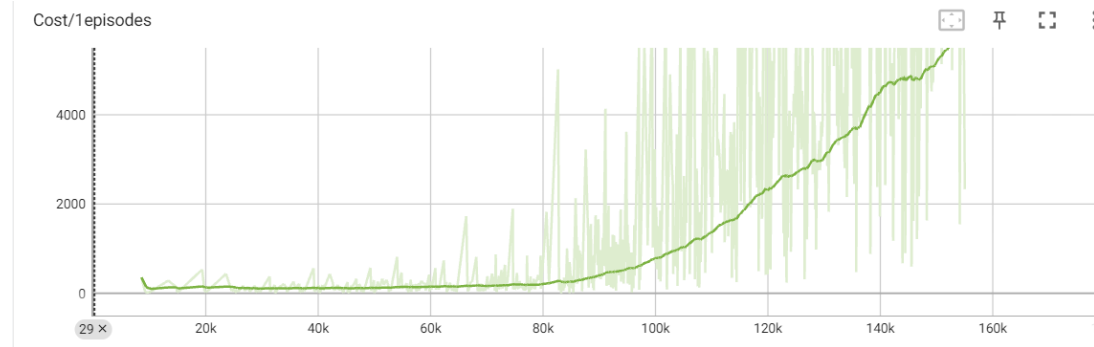
بعد التدريب، قامت العربة بتحقيق المهمة عند 100 ألف خطوة وتم الاستمرار بالتدريب حتى 150 ألف خطوة. ويوضح الشكل 60 من دالة الربح حيث نلاحظ استقرار الربح عند القيمة العظمى بعد 120 ألف خطوة.



الشكل 60 الربح بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar ومحددات الطرق وفق

CLRNet

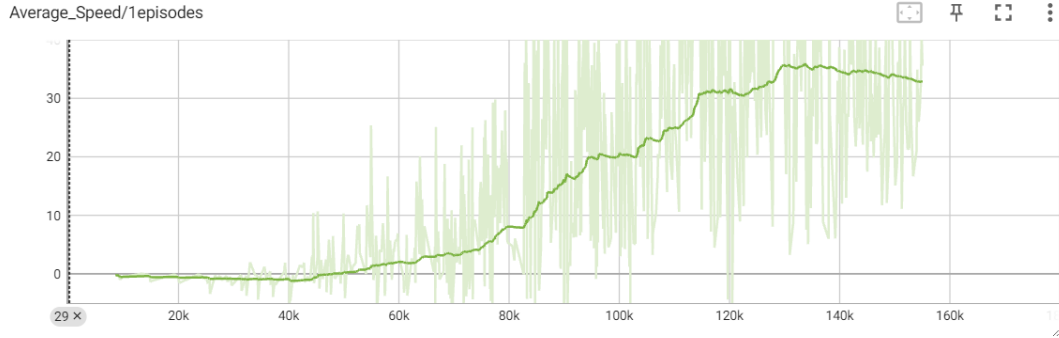
وكانت الخوارزمية أسرع في الوصول للحل مع إضافة البيانات الصورية. وأما بالنسبة للخسارة فهي موضحة في الشكل 61. ونلاحظ استقرار الخسارة في المراحل الأولى من التعلم وتزايدها بعد تعلم حل المسألة للأسباب المذكورة سابقاً والتي تتعلق بطبيعة خوارزمية DQN.



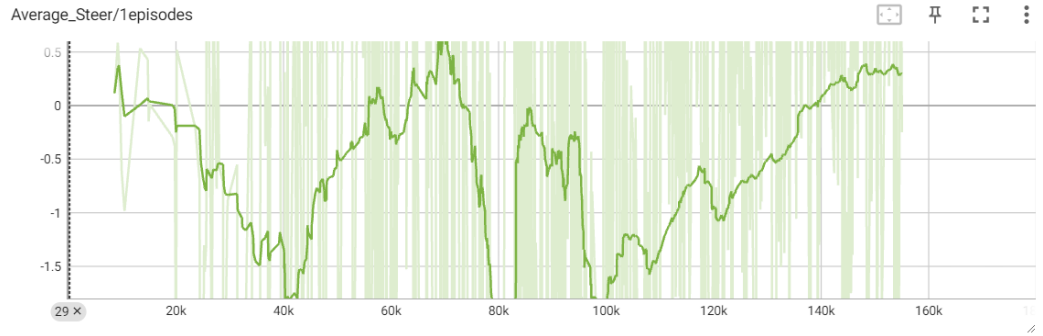
الشكل 61 دالة الخسارة بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar ومحددات الطرق

وفق CLRNet

وأما بالنسبة للسرعة وسرعة الانعطاف فهما موضحان في الشكل 62 و 63 على الترتيب بنتائج مشاهدة للتطبيق على حساس ليدار فقط باختلاف أن السرعة استقرت عند قيمة أقل لدى إجراء تتبع المحددات وهو مبرر لضمان الاستقرار في الطريق.



الشكل 62 السرعة بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar ومحددات الطرق وفق خوارزمية CLRNet



الشكل 63 سرعة الانعطاف بدلالة الخطوة خوارزمية DQN على دخل من حساس Lidar ومحددات الطرق وفق خوارزمية CLRNet

وبهذا يكون قد تم إنجاز تدريب العربة باستخدام خوارزميات التعلم المعزز العميق بالاستفادة من دخل الحساسات المختلفة وخوارزميات الرؤية الحاسوبية ضمن بيئة المحاكاة.

الختامة

تم في هذا المشروع إنشاء محاكي في بيئة افتراضية يمكن استخدامه في تطوير حلول لمسألة القيادة الذاتية بالاعتماد على خوارزميات الرؤية الحاسوبية والتعلم المعزز العميق. وتوظيف إمكانيات محرك يونيتي في تقديم بيئة محاكاة واقعية تغني حلول مسألة القيادة الذاتية وإجراء الربط بينها وبين خوارزميات الرؤية الحاسوبية والتعلم العميق لتوفير القدرة على تطوير النماذج باستخدام المحاكى المنجز. وقد تمكنا من إنشاء محاكاة ضمن بيئة يونيتي تقدم إمكانيات تتمتع بدرجة عالية من الدقة والواقعية، وتم ربط بيئة يونيتي بخوارزميات الرؤية الحاسوبية لكشف المحددات وكشف الأغراض ضمن المشهد وبخوارزميات التعلم المعزز الحديثة والمتطورة مما يفسح المجال لآفاق كبيرة فيما يتعلق بتدريب نماذج القيادة الذاتية ضمن يونيتي. وتم إغناء الحل بواجهات استعراض لخرج تلك الخوارزميات توفر مراقبة وتطوير سهل لاختبار وتدريب النماذج.

وبما أن الصعوبات جزء لا يتجزأ من أي عمل، فقد واجهت هذا المشروع بعض الصعوبات كان أهمها قلة المصادر المتعلقة باستخدام ML-Agents في حالات الاستخدام المتقدمة وعدم وجود تجهيزات مسبقة أو مجانية للتعامل مع خوارزميات الرؤية الحاسوبية ضمن يونيتي وخاصة خلال استخدام خوارزميات حديثة. واستغرق التدريب ساعات طويلة على عتاد الحواسيب الشخصية البسيط. وتعقيد المسألة الكبير المتطلب للتدرج في التدريب، والتي كانت السبب الأساسي في اتخاذ المشروع هذا التوجه في التنجيز العملي. وأما بالنسبة للآفاق، فإن هذا المشروع يمتلك العديد من الإمكانيات وأهمها اختبار خوارزميات أخرى درست ضمن الدراسة المرجعية مثل DDQN والتي قد تقدم حلولاً جديدة لمشاكل استقرار القيادة. ويمكن أيضاً تطوير منظومة تجنب العوائق ضمن المشهد بالاعتماد على خوارزميات تحكم يمكن دمج نتائجها مع النموذج المطور ضمن هذا المشروع.

المراجع

- [1] D. Parekh, N. Poddar, A. Rajpurkar, M. Chahal, N. Kumar, G. Joshi and W. Cho, "A Review on Autonomous Vehicles: Progress, Methods and Challenges," *Electronics*, 2022.
- [2] K. Min, H. Kim and K. Huh, "Deep Distributional Reinforcement Learning Based High Level Policy Determination," *IEEE TRANSACTIONS ON INTELLIGENT VEHICLES*, 2019.
- [3] N. K. Arzoo Miglani, "Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges," *Vehicular Communications*, Vols. ISSN 2214-2096, 2019.
- [4] T. Z. a. Y. H. a. Y. L. a. W. T. a. Z. Y. a. D. C. a. X. He, "CLRNet: Cross Layer Refinement Network for Lane Detection," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 888-897, 2022.
- [5] J. S. P. L. X. W. a. X. T. Xingang Pan, "Spatial As Deep: Spatial CNN for Traffic Scene Understanding," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
- [6] H. Abualsaud, "Reliable Multilane Detection and Classification by Utilizing CNN as a Regression Network," in *Computer Vision – ECCV 2018 Workshops*, Munich, Germany, 2018.
- [7] T. Wang, J. Xiong, X. Xu and Y. Shi, "SCNN: A General Distribution Based Statistical," in *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019.
- [8] T. Zheng, "RESA: Recurrent Feature-Shift Aggregator for Lane Detection," *CoRR*, 2020.
- [9] H. Xu, "CurveLane-NAS: Unifying Lane-Sensitive Architecture Search and Adaptive Point Blending," *CoRR*, vol. abs/2007.12147, 2020.
- [10] L. T. Torres, "Keep your Eyes on the Lane: Real-time Attention-guided Lane Detection," *CoRR*, vol. abs/2010.12035, 2020.
- [11] L. Liu, "CondLaneNet: a Top-to-down Lane Detection Framework Based on Conditional Convolution," *CoRR*, vol. abs/2105.05003, 2021.
- [12] E. Stenborg, "Long-term Visual Localization using Semantically Segmented

- Images," *CoRR*, vol. abs/1801.05269, 2018.
- [13] R. S. S. a. A. G. Barto, Reinforcement learning: an introduction, Second edition, Cambridge, MA: The MIT Press, 2018.
- [14] H. v. H. a. A. G. a. D. Silver, "Deep Reinforcement Learning with Double Q-learning," *CoRR*, 2015.
- [15] T. Hester, "Learning from Demonstrations for Real World Reinforcement Learning," *CoRR*, vol. abs/1704.03732, 2017.
- [16] J. Gregory, "Game Engine Architecture," in *Second Edition*, Boca Raton, FL 33487-2742, CRC Press, 2015, pp. 647-740.
- [17] M. Lanham, Learn Unity ML-Agents--Fundamentals of Unity Machine Learning, B3 2PB, UK.: Packt Publishing, 2018.
- [18] "RVO and ORCA How They Really Work," in *Game AI Pro 3- Collected Wisdom of Game AI Professionals*, New York, A K Peters/CRC Press, 2017, pp. 237-248.
- [19] J.-S. Zhao, "Design of an Ackermann Type Steering Mechanism," *Journal of Mechanical Engineering Science*, 2013.
- [20] "Unity Manual," Unity. Inc, [Online]. Available: <https://docs.unity3d.com/Manual/index.html>. [Accessed 2023].
- [21] "Unity ML-Agents Toolkit," Unity Technologies, [Online]. Available: <https://github.com/Unity-Technologies/ml-agents/tree/develop>. [Accessed 2023].