Πολυτεχνείο Κρήτης

Σχολή Μηχανικών Παραγωγής & Διοίκησης

Διπλωματική Εργασία

Ενισχυτική μάθηση στην αυτόνομη οδήγηση, μια συγκριτική αξιολόγηση

Ματσιώρης Δ. Γεώργιος

ΧΑΝΙΑ 2024

Επιτροπή

Ελευθέριος Δοϊτσίδης (Επιβλέπων), Επίκουρος Καθηγητής	Σχολή Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης
Ιωάννης Παπαμιχαήλ, Καθηγητής	Σχολή Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης
Δημήτριος Ιψάκης, Επίκουρος Καθηγητής	Σχολή Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης

Ευχαριστίες

Με το πέρας αυτής της εργασίας και ακολούθως των σπουδών μου στην σχολή Μηχανικών Παραγωγής & Διοίκησης του Πολυτεχνείου Κρήτης θα πρέπει να ευχαριστήσω όσους έχουν συμβάλει σε αυτό.

Ευχαριστώ τον καθηγητή μου κ. Δοϊτσίδη Ελευθέριο για την καθοδήγησή του, καθώς μου έδωσε την δυνατότητα να εργαστώ πάνω στον τομέα των ρομπότ & των αυτόνομων οχημάτων και με βοήθησε σε οποιαδήποτε στιγμή χρειάστηκε.

Στη συνέχεια θα ευχαριστήσω όλους τους καθηγητές και το διδακτικό προσωπικό του Πολυτεχνείου Κρήτης για το έργο τους το οποίο επιδρά καταλυτικά σε κάθε φοιτητή του ιδρύματος.

Με τη μέγιστη σημασία, ευχαριστώ τους γονείς μου των οποίων οι προσπάθειες κατέστησαν δυνατή όλη αυτήν την πορεία.

Και κλείνοντας ευχαριστώ τα αδέρφια μου, τους φίλους μου και τους συγγενείς που συνέβαλαν σε αυτή την διαδρομή και την έκαναν ευχάριστη.

Table of Contents

Πίνακας Εικόνων

Πίνακας Πινάκων

Περίληψη

Στόχος της παρούσας εργασίας είναι να αναπτυχθεί ένα μεθοδολογικό πλαίσιο που θα βοηθά στην ανάπτυξη αυτόνομων πρακτόρων για την καθοδήγηση ηλεκτρικών οχημάτων με δυνατότητες αυτόνομης πλοήγησης. Για την ανάπτυξη του συγκεκριμένου πλαισίου, γίνεται χρήση του μοντέλου ενός ηλεκτρικού αυτοκινήτου πόλης που έχει τη δυνατότητα να φέρει πλήθος διαφορετικών αισθητήρων.

Στα πλαίσια της εργασίας, αρχικά εξελίσσεται ένα προσομοιωμένο μοντέλο που έχει αναπτυχθεί σε παλαιότερη εργασίας. Η λειτουργικότητα του μοντέλου δοκιμάζεται σε ένα προσομοιωμένο περιβάλλον βασισμένο στο λογισμικό CARLA με τη χρήση ενός ειδικά κατασκευασμένου περιβάλλοντος τύπου GYM. Έπειτα αναπτύσσονται αυτόνομοι πράκτορες που έχουν την δυνατότητα να καθοδηγήσουν το μοντέλο του οχήματος, έτσι ώστε να ακολουθήσει ένα προδιαγεγραμμένο μονοπάτι μέσα σε αστικό περιβάλλον. Η διαδικασία βασίζεται στη χρήση του αλγορίθμου A* για την παραγωγή της επιθυμητής τροχιάς και στη συνέχεια τη χρήση των σχετικών δεδομένων για την εκπαίδευση των αυτόνομων πρακτόρων. Για την εκπαίδευση των πρακτόρων χρησιμοποιούνται δύο διαφορετικοί αλγόριθμοι ενισχυτικής μάθησης και πραγματοποιείται συγκριτική μελέτη των αποτελεσμάτων.

Abstract

The objective of this thesis is to develop a methodological framework that facilitates the development of autonomous agents for guiding electric vehicles with autonomous navigation capabilities. To develop this framework, we utilize the model of an urban electric car equipped with a variety of sensors.

Initially, a simulated model from a previous study is enhanced. The functionality of this model is tested in a simulated environment based on the CARLA software using a specially designed GYM environment. Subsequently, autonomous agents are developed to guide the vehicle model along a predetermined path in an urban setting. This process relies on the A* algorithm to generate the desired trajectory, followed by using the corresponding data to train the autonomous agents. Two different reinforcement learning algorithms are employed to train the agents, and a comparative study of the results is conducted.

Εισαγωγή

Η σύγχρονη αυτοκινητοβιομηχανία βρίσκεται υπό συνεχείς μεταμορφώσεις, καθώς εστιάζει στην δημιουργία πλήρως αυτόνομων οχημάτων, εμπλέκοντας διαφορετικούς επιστημονικούς τομείς όπως η μηχανολογία, η πληροφορική και η ηλεκτρολογία. Η εμφάνιση αυτής της τάσης μεταστροφής έγκειται στην ανάγκη για αποτελεσματικότερους τρόπους μεταφοράς, που εστιάζουν στην διευκόλυνση των μετακινήσεων, ενώ ταυτοχρόνως μειώνουν και τις εκπομπές άνθρακα κάτω από το πρίσμα μιας νέας βιώσιμης κοινωνίας.

Όσον αφορά στην αυτονομία των αυτοκινήτων, αυτή κατηγοριοποιείται σε έξι επίπεδα όπως αυτά παρουσιάζονται στο [1], σύμφωνα με τις δυνατότητες του αυτόνομου συστήματος. Τα έξι αυτά επίπεδα ξεκινάν με το επίπεδο 0 στο οποίο δεν υπάρχει καθόλου αυτοματισμός, και καταλήγουν στο 5 όπου το όχημα είναι πλήρως αυτοματοποιημένο. Το επίπεδο 5 ακόμα και με την υιοθέτηση τεχνολογιών αιχμής είναι δύσκολο να επιτευχθεί, καθώς απαιτείται τεράστια υπολογιστική ισχύς για την δημιουργία ενός αυτόνομου λειτουργικού πράκτορα που θα μπορεί να ανταπεξέλθει στα διαφορετικά δυναμικά περιβάλλοντα που υπάρχουν στους δρόμους και στις διαφορετικές προκλήσεις και κινδύνους που συναντώνται.

Επιπρόσθετα, πέραν των φυσικών περιορισμών ανακύπτουν και άλλα ζητήματα όπως η αξιοπιστία και η ασφάλεια του λογισμικού που είναι κρίσιμοι παράγοντες, καθώς εκτός από την αυτόνομη λειτουργία, υπό την ευθύνη τους βρίσκεται και κάθε άλλο υποσύστημα του αυτοκινήτου, όπως π.χ. αυτό της μετάδοσης κίνησης και της διαχείρισης ενέργειας. Ακόμα τίθενται σημαντικά θέματα που σχετίζονται με την ασφάλεια και την ιδιωτικότητα [2]. Επιπρόσθετα, εκτός από το ίδιο το αυτοκίνητο, καθώς η αναφορά γίνεται για μια ολόκληρη κοινωνία αυτοματοποιημένων μεταφορών, είναι αναγκαία η δημιουργία ενός πλήρους οικοσυστήματος έξυπνων μεταφορών και όχι απλά μεμονωμένων οχημάτων [3]. Αυτά τα οικοσυστήματα είναι γνωστά ως Ευφυή Συστήματα Μεταφοράς (Inteligent Trasportation Systems-ITS), μέρος των Συστημάτων των Συστημάτων (Systems of Systems SoS) και εγγενώς η διαχείρισή τους είναι μία πρόκληση [4].

Προφανώς η ανάπτυξη όλου αυτού του συστήματος, αφού έχει γνώμονα ένα πράσινο μέλλον χωρίς αέριους ρύπους, βασίζεται σε μεγάλο βαθμό στα ηλεκτρικά οχήματα. Βασικό ζήτημα όμως για την ανάπτυξη αυτόνομων ηλεκτρικών αυτοκινήτων, είναι η δημιουργία ενός πλαισίου που θα επιτρέπει στους διάφορους ερευνητές να σχεδιάζουν και να δοκιμάζουν οποιαδήποτε εφαρμογές με στόχο την αυτόνομη λειτουργία τους. Μια προσέγγιση αυτού του στόχου που έχει αποκτήσει δυναμική τα τελευταία χρόνια [5], είναι η υιοθέτηση των ψηφιακών διδύμων (Digital Twins-DT). Σύμφωνα με το [6], *«Ψηφιακό δίδυμο είναι η ψηφιακή αναπαράσταση ενός μοναδικού προϊόντος (πραγματική συσκευή, αντικείμενο, μηχανή, υπηρεσία)*

ή ένα μοναδικό σύστημα προϊόντος-υπηρεσίας (ένα σύστημα που αποτελείται από ένα προϊόν και μια σχετική υπηρεσία) που περιλαμβάνει τα επιλεγμένα χαρακτηριστικά, τις ιδιότητες, τις συνθήκες και τις αντιδράσεις της μέσω μοντέλων, πληροφοριών και δεδομένων σε μία μόνο ή ακόμη και σε πολλαπλές φάσεις του κύκλου ζωής». Η εφαρμογή των DT σε αυτόνομα οχήματα είναι προς το παρόν περιορισμένη, αφού μόλις πρόσφατα κατέστη δυνατή λόγω των εξελίξεων σε τομείς όπως το Διαδίκτυο των Πραγμάτων (Internet of Things-IoT), το 5G και άλλα. Στην εργασία [7] παρουσιάζεται μία πλατφόρμα για την έρευνα στους τομείς των αυτόνομων οχημάτων και στα DT. Ωστόσο παραμένει μια εργαστηριακή προσέγγιση σε μοντέλα υπό κλίμακα.

Λαμβάνοντας υπόψιν τις προκλήσεις και τα παραπάνω ανοικτά θέματα, το Εργαστήριο Ευφυών Συστημάτων & Ρομποτικής του Πολυτεχνείου Κρήτης, έχει εξοπλιστεί με ένα διθέσιο αμιγώς ηλεκτρικό αυτοκίνητο πόλης, με στόχο να το μετατρέψει σε πλατφόρμα πειραματισμού στο πεδίο της αυτόνομης οδήγησης, με την ενσωμάτωση διάφορων αισθητήρων και βασισμένο στο υπολογιστικό σύστημα DRIVE AGX της NVIDIA. Η αρχική προσέγγιση για την επίτευξη αυτού του στόχου, είναι η ανάπτυξη ενός προσομοιωμένου μοντέλου του οχήματος χρησιμοποιώντας τον προσομοιωτή CARLA, ένα εργαλείο αιχμής που παρέχει επιλογές προσομοίωσης αυτόνομης οδήγησης σε άκρως ρεαλιστικό περιβάλλον. Η προσέγγιση έχεις ως αρχικό στόχο την εξερεύνηση των δυνατοτήτων των αισθητήρων του CARLA και στη συνέχεια την εισαγωγή του μοντέλου του, τον προσομοιωτή. Απώτερος στόχος είναι η δημιουργία ενός πλήρως λειτουργικού ψηφιακού διδύμου που θα επιτρέπει την αλληλεπίδραση του προσομοιωμένου συστήματος με το πραγματικό.

Στην παρούσα εργασία, μελετάται η λειτουργικότητα της συγκεκριμένης προσέγγισης μέσω της εφαρμογής διαφορετικών αλγορίθμων ενισχυτικής μάθησης για την ανάπτυξη ελεγκτών που θα επιτρέπουν την αυτόνομη λειτουργία του προσομοιωμένου οχήματος. Συγκεκριμένα στόχος είναι το προσομοιωμένο όχημα να μπορεί να ακολουθήσει μια βέλτιστη τροχιά που υπολογίζεται μέσα σε ένα αστικό περιβάλλον με τη χρήση του αλγορίθμου A^*. Το μοντελοποιημένο όχημα και η μέθοδος επίβλεψης της όλης διαδικασίας στηρίζεται στην εργασία [8], όπου και δημιουργήθηκε το μοντέλο του αυτοκινήτου και τα απαραίτητα προγράμματα για την εκπαίδευση και εφαρμογή των αλγορίθμων. Οι αλγόριθμοι που εξετάζονται είναι ο Soft Actor Critic (SAC) και ο Twin Delayed 3 (TD3), με τη βοήθεια του GYM API και της Stable Baselines3.

Η δομή της εργασίας είναι η ακόλουθη. Στο 1ο κεφάλαιο, παρουσιάζονται αναλυτικά οι βασικές έννοιες και το θεωρητικό υπόβαθρο. Στο 2ο κεφάλαιο παρουσιάζεται το περιβάλλον και το πλαίσιο προσομοίωσης, κάθε και τα σχετικά υπολογιστικά εργαλεία που χρησιμοποιήθηκαν αλλά και τα βασικά τεχνικά χαρακτηριστικά του οχήματος. Στο 3ο

κεφάλαιο παρουσιάζονται αναλυτικά τα πειραματικά αποτελέσματα και τέλος στο 4ο κεφάλαιο παρουσιάζονται συμπεράσματα και μελλοντικές επεκτάσεις.

Μέρος της παρούσας εργασίας βρίσκεται στο παρακάτω:

G. Matsioris, A. Theocharous, N. Tsourveloudis and L. Doitsidis, "Towards Developing a Framework for Autonomous Electric Vehicles Using CARLA: A Validation Using the Deep Deterministic Policy Gradient Algorithm," *2024 32nd Mediterranean Conference on Control and Automation (MED)*, Chania - Crete, Greece, 2024, pp. 470-475, doi: 10.1109/MED61351.2024.10566221.

1. Θεωρητικό Υπόβαθρο

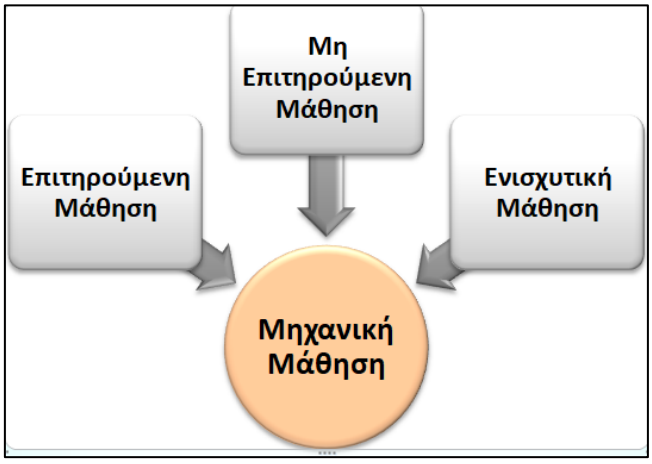
Στο 1ο κεφάλαιο, παρουσιάζονται έννοιες σχετικές με την ενισχυτική μάθηση και περιγράφεται εκτενώς ο τρόπος λειτουργίας της. Γίνεται ανάλυση επιλεγμένων αλγορίθμων που ανήκουν σε αυτή την οικογένεια και χρησιμοποιήθηκαν στα πλαίσια της παρούσας εργασίας. Τέλος παρουσιάζεται μια σύντομη βιβλιογραφική επισκόπηση εφαρμογών της ενισχυτικής μάθησης, στον τομέα της αυτόνομης οδήγησης.

1.1 Τεχνητή Νοημοσύνη & Μηχανική Μάθηση

Η Τεχνητή Νοημοσύνη (ΤΝ), γνωρίζει μεγάλη άνθιση στον χώρο της αυτοκίνησης καθώς η ανάγκη για πιο αποδοτικούς τρόπους μετακίνησης είναι επιτακτική, με την έννοια "πιο αποδοτικός" να αναφέρεται στην επιλογή πορείας, στην αποφυγή εμποδίων, στην κατανάλωση και σε κάθε άλλο πιθανό παράγοντα. Ο ορισμός που μπορεί να δοθεί στην ΤΝ είναι ο εξής: *«Τεχνητή Νοημοσύνη είναι εκείνος ο κλάδος της επιστήμης των υπολογιστών που ασχολείται με το σχεδιασμό ευφυών υπολογιστικών συστημάτων, δηλαδή συστημάτων με χαρακτηριστικά τα οποία σχετίζονται με την ευφυΐα στην ανθρώπινη συμπεριφορά (μάθηση, αιτίαση, επίλυση προβλημάτων, κατανόηση φυσικής γλώσσας, αναγνώριση αντικειμένων κτλ.)»* [9]. Η παρούσα εργασία εστιάζει στην εφαρμογή αυτών των τεχνικών στην αυτόνομη οδήγηση και συγκεκριμένα σε μία κατηγορία του κλάδου της μηχανικής μάθησης καθώς εφαρμόζονται αλγόριθμοι ενισχυτικής μάθησης σε προσομοιωμένο περιβάλλον για την πλοήγηση ενός οχήματος.

Η Μηχανική Μάθηση (Machine Learning) αποτελεί το εργαλείο με το οποίο ένα ευφυές σύστημα μαθαίνει από την εκπαίδευσή του σε συγκεκριμένα δεδομένα, ώστε να αυτοματοποιεί την διαδικασία επίλυσης παρόμοιων προβλημάτων. Αντί λοιπόν να χρειάζεται η εισαγωγή κάθε πιθανού προς λύση προβλήματος από τον χρήστη, ο υπολογιστής μαθαίνει να αναγνωρίζει σχέσεις και μοτίβα. Η εξέλιξη της μηχανικής μάθησης οδήγησε στην δημιουργία πολύπλοκων αλγορίθμων και τεχνικών όπως τα Τεχνητά Νευρωνικά Δίκτυα ή απλώς Νευρωνικά Δίκτυα (Neural Network) [10]. Τα ΝΝ προσπαθούν να προσομοιάσουν σε μία απλοποιημένη μορφή, των ανθρώπινο εγκέφαλο δημιουργώντας ένα δίκτυο "νευρώνων" των οποίων οι είσοδοι είναι η πληροφορία προς επεξεργασία και η έξοδος το αποτέλεσμα του υπολογιστή. Η ενισχυτική μάθηση αποτελεί μέρος των τεχνικών της μηχανικής μάθησης καθώς αποσκοπεί στο να "εκπαιδεύσει" ένα νευρωνικό δίκτυο για κάποια συγκεκριμένη λειτουργία, εν προκειμένω την πλοήγηση του οχήματος. Στην Εικόνα 1 Τύποι Μηχανικής Μάθησης

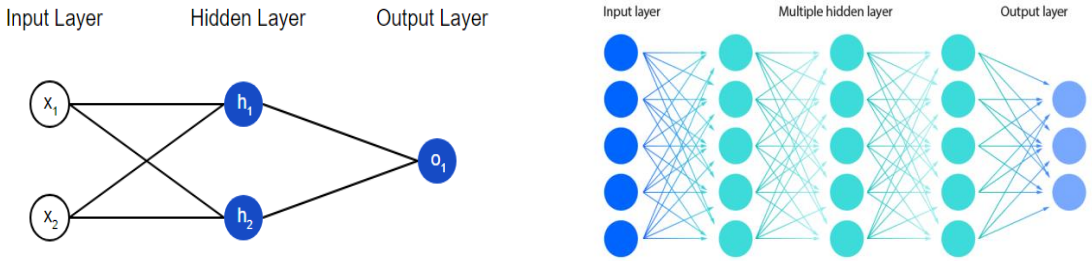
παρουσιάζονται οι τρείς κύριες μέθοδοι μηχανικής μάθησης, που χρησιμοποιούν τα νευρωνικά δίκτυα για την επίλυση προβλημάτων.



Εικόνα 1 Τύποι Μηχανικής Μάθησης

Η πολυεπίπεδη κατασκευή νευρωνικών δικτύων, οδήγησε στη δημιουργία του κλάδου της Βαθιάς Μάθησης (Deep Learning), μέθοδος κατά την οποία η προς επεξεργασία πληροφορία περνά από πολλά επίπεδα νευρώνων με διαφορετικού τύπου αρχιτεκτονική κατά περίπτωση. Στην Εικόνα 2 Στην αριστερή εικόνα φαίνεται ένα απλό νευρωνικό δίκτυο ενώ στην δεξιά ένα ‘βαθύ’ ,

, συγκρίνονται ένα απλό και ένα βαθύ νευρωνικό δίκτυο, όπου και η διαφορά στην πολυπλοκότητα είναι εμφανής. Μέσω της βαθιάς μάθησης, προήλθε αντίστοιχα η βαθιά ενισχυτική μάθηση (Deep Reinforcement Learning – DLR), τεχνική κατά την οποία στο δίκτυο ενισχυτικής μάθησης προστίθενται κι άλλα επίπεδα νευρώνων κάνοντάς το “βαθύ”.



Εικόνα 2 Στην αριστερή εικόνα φαίνεται ένα απλό νευρωνικό δίκτυο ενώ στην δεξιά ένα ‘βαθύ’ [11], [12]

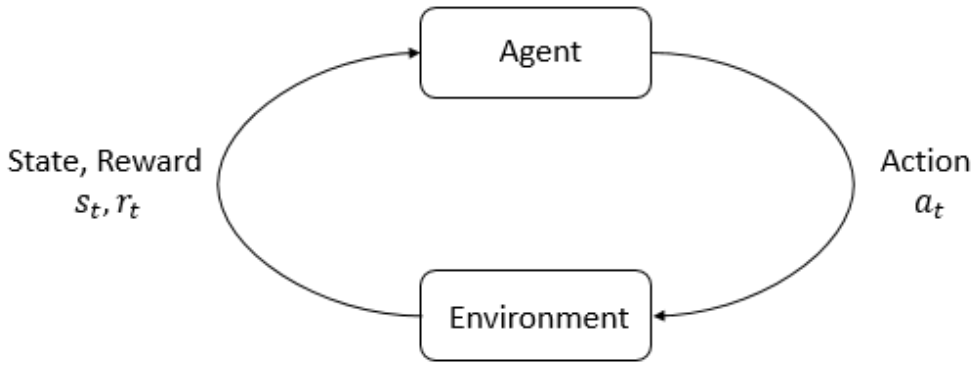
1.2 Ενισχυτική Μάθηση

Η ενισχυτική μάθηση είναι μία τεχνική εκμάθησης κατά την οποία ένα δίκτυο τεχνητής νοημοσύνη εκπαιδεύεται καθώς αλληλεπιδρά με το περιβάλλον του μέσω δοκιμής και σφάλματος και υιοθετεί την βέλτιστη συμπεριφορά βάσει του σήματος ορισμένης συνάρτησης επιβράβευσης που λαμβάνει σε κάθε ενέργεια. Η μέθοδος αυτή μιμείται τον τρόπο που

μαθαίνουν οι άνθρωποι και τα ζώα και παρουσιάζει μεγάλη προσαρμοστικότητα σε ένα ευρύ φάσμα επιστημονικών πεδίων, καθώς έχει επιδείξει την δυνατότητα να επιλύει πολύπλοκα προβλήματα που βασίζονται όχι σε μεμονωμένες αποφάσεις, αλλά σε σειρά ενεργειών [13].

Η μέθοδος δοκιμής και σφάλματος μαζί με την επιβράβευση της αλληλουχίας των πράξεων, είναι τα δύο κυριότερα χαρακτηριστικά της ενισχυτικής μάθησης. Ανάγοντας αυτή την τεχνική στον πραγματικό κόσμο, θα μπορούσε κάποιος να φέρει το παράδειγμα ενός σκακιστή που επιλέγει την κίνησή του αφότου έχει σκεφτεί τις πιθανές απαντήσεις του αντιπάλου αλλά και την επιθυμητή θέση στην οποία θέλει να καταλήξει [14]. Αν και τα τελευταία χρόνια γνωρίζει μεγάλη άνθιση, προτάσεις για την χρήση της υπάρχουν από τις αρχές του 20ου αιώνα με το "Law of Effect" του Thorndike το 1911, την εισαγωγή του όρου "Reinforcement" από τους Pavlov και Anrep το 1927 αλλά και το "Pleasure-Pain System" του Turring το 1948. Σε κάθε περίπτωση, ο κοινός παρονομαστής είναι ότι για κάθε σωστή απόφαση υπάρχει μία επιβράβευση που εν τέλει θα εκπαιδεύσει το ανάλογο σύστημα.

Στην **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**, παρουσιάζεται ένα σ χεδιάγραμμα συστήματος ενισχυτικής μάθησης. Κάθε στάδιο περιγράφεται από την κατάσταση s_t η οποία είναι μεταβλητή που περιγράφει πλήρως το περιβάλλον, την στιγμή t και δεν πρέπει να συγχέεται με την μεταβλητή o που ανήκει στον *χώρο παρατήρησης (observation space)*, ο οποίος περιγράφει μερικώς την κατάσταση s_t. Η μεταβλητή r_t είναι η επιβράβευση την στιγμή t η οποία είναι αποτέλεσμα της πράξης a_t του αλγορίθμου, η οποία ανήκει στον *χώρο δράσεων (action space)*, του οποίου η διαστάσεις είναι ανάλογες των ενεργειών που δύναται να πράξει το σύστημα.



Εικόνα 3 Σχεδιάγραμμα Συστήματος Ενισχυτικής Μάθησης [15]

Όσον αφορά στον τρόπο με τον οποίον ο αλγόριθμος επιλέγει την εκάστοτε ενέργεια αυτός ονομάζεται *πολιτική (policy)*. Είναι η νόρμα την οποία χρησιμοποιεί ο αλγόριθμος για να επιλέξει την ενέργεια a_t και συμβολίζεται ως:

$$a_t = \mu(s_t), \text{για αιτιοκρατική πολιτική} \quad (1.1)$$

$$a_t \sim \pi(\cdot\,|s_t), \text{για στοχαστική πολιτική} \quad (1.2)$$

Η σειρά των ενεργειών με τις αντίστοιχες καταστάσεις δημιουργούν σύνολα τα οποία ονομάζονται *επεισόδια* (episodes ή rollouts) και συμβολίζονται με $\tau = (s_0, a_0, s_1, a_1 \ldots)$. Οι όροι s_0 και a_0 είναι αντίστοιχα η αρχική κατάστασή και η αρχική ενέργεια. Οι μεταβολές από την κατάσταση s_t στην κατάσταση s_{t+1} διέπονται από φυσικούς νόμους και είναι απόρροια της πράξης a_t. Ο ορισμός της κατάστασης s_{t+1} παίρνει λοιπόν την εξής μορφή:

$$s_{t+1} = f(s_t, a_t), \text{για αιτιοκρατική πολιτική} \quad (1.3)$$

$$s_{t+1} \sim P(\cdot\,|s_t, a_t), \text{για στοχαστική πολιτική} \quad (1.4)$$

Η δυναμική μοντελοποίηση των προβλημάτων ενισχυτικής μάθησης γίνεται συνήθως μέσω των αλυσίδων Markov. Οι αλυσίδες Markov, είναι στοχαστικές διαδικασίες κατά τις οποίες η μελλοντική κατάσταση ενός συστήματος βασίζεται στην παρούσα κατάσταση και δυνατές επιλογές μετάβασης από μία κατάσταση στην επόμενη διέπονται από ανάλογες πιθανότητες. Όπως συμπεραίνει κανείς, η σειρά μεταβάσεων από μία κατάσταση στις πιθανές επόμενες σε ένα πρόβλημα ενισχυτικής μάθησης είναι μια αλυσίδα Markov που συνδέει κάθε κατάσταση s_t με τις επόμενες πιθανές.

Η *συνάρτηση επιβράβευσης* και *συνάρτηση αξίας* αν και ακούγονται παρόμοιες, ο ρόλος τους διαφέρει. Όπως αναλύθηκε, ο αλγόριθμος στοχεύει στην μεγιστοποίηση της ανταμοιβής από τις πράξεις του, η ανταμοιβή αυτή λοιπόν δίνεται από την συνάρτηση επιβράβευσης. Όσον αφορά στην συνάρτηση αξίας, είναι η συνάρτηση η οποία δεδομένης μίας κατάστασης s_t ή ενός ζεύγους κατάστασης-πράξης (s_t, a_t) υπολογίζει την αναμενόμενη ανταμοιβή εφόσον ακολουθηθεί ορισμένη πολιτική.

Συνάρτηση Επιβράβευσης

Η συνάρτηση επιβράβευσης εξαρτάται από την παρούσα κατάσταση s_t, την ενέργεια a_t και την επόμενη κατάσταση s_{t+1}, και αναγράφεται ως:

$$r_t = R(s_t, a_t, s_{t+1}), \quad (1.5)$$

χάριν συντομίας όμως συνήθως παραλείπεται η κατάσταση s_{t+1}, κρατώντας μόνο το ζεύγος (s_t, a_t), καταλήγοντας στην:

$$r_t = R(s_t, a_t) \quad (1.6)$$

Εφόσον ο στόχος είναι η μεγιστοποίηση της r_t σε κάθε βήμα και κατ' επέκταση το άθροισμα καθ' όλο το επεισόδιο τ, προκύπτει η εξής έκφραση:

$$R(\tau) = \sum_{t=0}^{T} r_t \quad (1.7)$$

Η παραπάνω εξίσωση είναι για πεπερασμένο διάστημα, χωρίς συντελεστή έκπτωσης. Για άπειρο διάστημα η συνάρτηση έχει σε κάθε στοιχείο του αθροίσματος ένα συντελεστή $\gamma \in (0,1)$, ο οποίος φροντίζει ώστε το άθροισμα να συγκλίνει σε κάποιον πραγματικό αριθμό και να μην απειρίζει.

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t \quad (1.8)$$

<u>Συνάρτηση Αξίας</u>

Η συνάρτηση αξίας, μπορεί να εξαρτάται είτε από την παρούσα κατάσταση s_t, είτε από το ζεύγος (s_t, a_t) και ανά περίπτωση να αλλάζει μορφή βάσει της πολιτικής. Έτσι εντοπίζονται τέσσερα είδη συναρτήσεων αναλόγως την περίπτωση.

Αρχικά είναι οι συναρτήσεις *on-policy Value* $V^{\pi}(s)$, όταν δηλαδή επιλέγεται να ξεκινήσει ο αλγόριθμος από την κατάσταση s_0 έχοντας μία πολιτική π η οποία διατηρείται για πάντα ξεκινώντας από την πρώτης πράξη a_0.

$$V^{\pi}(s) = E\,[R(\tau)|s_0 = s] \quad (1.9)$$

Ομοίως λειτουργούν και οι συναρτήσεις on-policy Action-Value $Q^{\pi}(s,a)$ που διατηρούν την επιλεγμένη πολιτική π αλλά ξεκινάνε από το ζεύγος (s_0, a_0), όπου η πρώτη πράξη a_0 έχει επιλεχθεί αυθαίρετα και όχι βάσει πολιτικής.

$$Q^{\pi}(s,a) = E\,[R(\tau)|s_0 = s, a_0 = a] \quad (1.10)$$

Το επόμενο είδος συναρτήσεων αξίας είναι οι συναρτήσεις Optimal Value $V^{*}(s)$, εδώ ο αλγόριθμος πάλι εκκινεί από αρχική κατάσταση s_0 αλλά η πολιτική αυτή τη φορά δεν μένει σταθερή, υπολογίζεται η ανταμοιβή εφόσον σε κάθε κατάσταση s_t επιλεγεί η βέλτιστη ενέργεια a_t, αυτή δηλαδή που θα δώσει την μέγιστη ανταμοιβή από της παρούσες επιλογές.

$$V^{*}(s) = \max_{\pi} E\,[R(\tau)|s_0 = s] \quad (1.11)$$

Οι εξισώσεις Optimal Action-Value $Q^{*}(s,a)$, σε αντιστοιχία με τις προηγούμενες ακολουθούν πάντοτε την βέλτιστη πολιτική αλλά αντί να αρχίζουν με την κατάσταση s_0, ξεκινούν με το ζεύγος (s_0, a_0), όπου a_0 μία αυθαίρετη πρώτη ενέργεια.

$$Q^{*}(s,a) = \max_{\pi} E\,[R(\tau)|s_0 = s, a_0 = a] \quad (1.12)$$

Η συνάρτηση αξίας είναι ο μηχανισμός πρόβλεψης της επιβράβευσης καθώς βρίσκει την αναμενόμενη τιμή των ενεργειών. Είναι σαφώς πιο δύσκολο να υπολογιστεί και απαιτείται περισσότερη υπολογιστική ισχύς καθώς εν αντιθέσει με την επιβράβευση που υπολογίζεται απευθείας μετά την ενέργεια, η συνάρτηση αξίας πρέπει να επαναϋπολογίζει σε κάθε περίπτωση το αποτέλεσμά της. Θα μπορούσε να θεωρηθεί ως το σημαντικότερο στοιχείο ενός

μοντέλου ενισχυτικής μάθησης καθότι με την αποτελεσματική πρόβλεψη της ανταμοιβής επιτυγχάνονται τα καλύτερα αποτελέσματα [13], [14]. Θα πρέπει να σημειωθεί ότι από τις εξισώσεις 1.10 & 1.12 έχει προκύψει ένας ολόκληρος κλάδος της ενισχυτικής μάθησης, οι Q-Learning αλγόριθμοι.

Οι συναρτήσεις αξίας που περιεγραφήκαν παραπάνω ανήκουν σε μία ευρύτερη οικογένεια εξισώσεων, τις εξισώσεις Bellman. Κατ' αυτές τις εξισώσεις η αξία κάθε κατάστασης s_t είναι το άθροισμα της παρούσας αμοιβής αλλά και της επόμενης στην κατάσταση s_{t+1}. Έτσι κάθε μία από τις παραπάνω εξισώσεις τροποποιείται ως εξής:

$$V^{\pi}(s) = E\left[r(s_t, a_t) + \gamma V^{\pi}(s_{t+1})\right] \tag{1.13}$$

$$Q^{\pi}(s, a) = E\left[r(s_t, a_t) + \gamma Q^{\pi}(s_{t+1}, a_{t+1})\right] \tag{1.14}$$

$$V^{*}(s) = \max_{a} E\left[r(s_t, a_t) + \gamma V^{*}(s_{t+1})\right] \tag{1.15}$$

$$Q^{*}(s, a) = E\left[r(s_t, a_t) + \gamma \max_{a_{t+1}} Q^{*}(s_{t+1}, a_{t+1})\right] \tag{1.16}$$

Για να υπολογιστεί η βέλτιστη πολιτική και κατά συνέπεια το βέλτιστο αποτέλεσμα, θα πρέπει να γίνει λόγος για την πιθανότητα που έχει να εμφανιστεί το επεισόδιο τ με δεδομένη στοχαστική πολιτική π, και αρχική κατανομή ρ_0.

$$P(\tau|\pi) = \rho_0(s_0) \prod_{t=0}^{T-1} P(s_{t+1}|s_t, a_t)\pi(a_t|s_t) \tag{1.17}$$

Το αναμενόμενο αποτέλεσμα $J(\pi)$ βάσει της εξίσωσης 1.17 δίνεται από τον ακόλουθο τύπο.

$$J(\pi) = \int_{\tau} P(\tau|\pi)R(\tau) = E[R(\tau)] \tag{1.18}$$

Έχοντας λοιπόν ορίσει την σημασία της βέλτιστης πολιτικής και το αναμενόμενο αποτέλεσμα βάσει πιθανότητας του επεισοδίου τ, η βέλτιστη πολιτική σε οποιαδήποτε περίπτωση μαθηματικά εκφράζεται ως:

$$\pi^{*} = \arg\min_{\pi} J(\pi) \tag{1.19}$$

Σύμφωνα με τον μαθηματικό ορισμό της βέλτιστης πολιτικής μπορεί να οριστεί και η βέλτιστη πράξη a^{*}. Στην κατάσταση s, εφόσον χρησιμοποιείται η συνάρτηση αξίας 1.12, θα επιλεγεί η ενέργεια που μεγιστοποιεί την ανταμοιβή. Ενδέχεται να υπάρχουν πάνω από μία τέτοιες ενέργειες, οπότε και η επιλογή είναι τυχαία.

$$a^{*}(s) = \arg\max_{a} Q^{*}(s, a) \tag{1.20}$$

Από τη στιγμή που ορίζεται μία ενέργεια ως βέλτιστη, αυτομάτως δημιουργείται και το ερώτημα "*Πόσο καλύτερη είναι από τις άλλες επιλογές;*". Για την απάντηση στο ερώτημα αυτό υπάρχει λοιπόν η *συνάρτηση πλεονεκτήματος (Advantage Function)*, η οποία δίνει την διαφορά της αναμενόμενης επιβράβευσης σε σχέση με μία αυθαίρετη επιλογή ενέργειας.

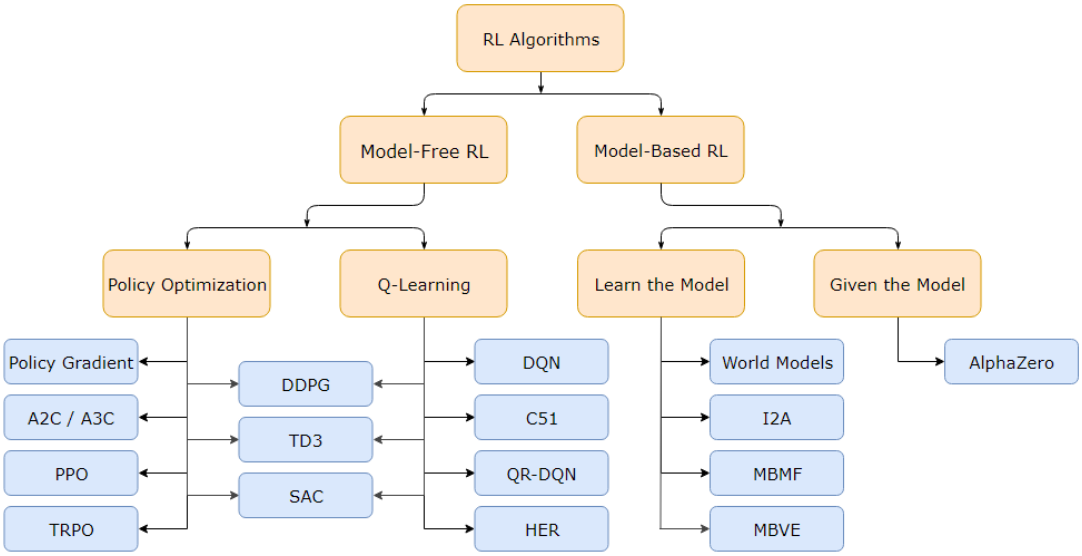
$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s) \tag{21}$$

Συνοψίζοντας τα δομικά μέρη ενός μοντέλου ενισχυτικής μάθησης από τους παραπάνω ορισμούς προκύπτουν τα ακόλουθα:

- S, το διάνυσμα των καταστάσεων
- A, το διάνυσμα των πράξεων
- $R \in \mathbb{R}$, είναι η ανταμοιβή που προκύπτει από τη συνάρτηση επιβράβευσης 1.5
- $P(s_{t+1}|s_t, a_t)$, η συνάρτηση πιθανότητας για την μετάβαση στην κατάσταση s_{t+1} από την κατάσταση s_t με επιλογή a_t [15].

1.3 Ταξινόμηση Αλγορίθμων Ενισχυτικής Μάθησης (Reinforcement Learning)

Η ποικιλία των αλγορίθμων ενισχυτικής μάθησης, δύναται να κατηγοριοποιηθεί σύμφωνα με το εάν βάσει μοντέλου του περιβάλλοντος ή όχι. Σύφμωνα με αυτή την κατηγοροιοποίηση στη συνέχεια θα γίνει ανάλυση σύμφωνα με τις κύριες πολιτικές κάθε μίας από αυτές τις κατηγορίες. Ένας ενδεικτικός τρόπος ταξινόμησης όπως έχει προταθεί στο [15], παρουσιάζεται στην Εικόνα 4.

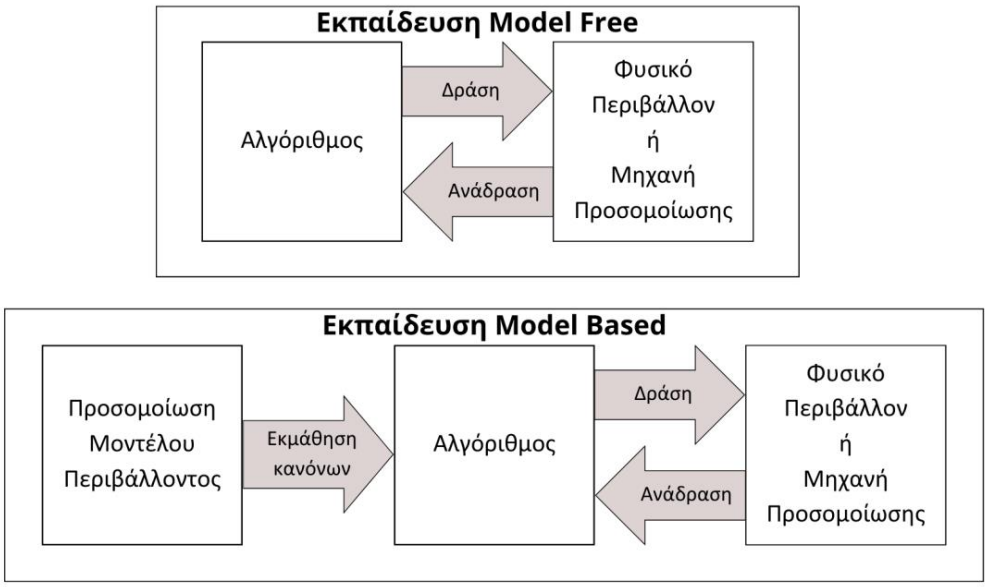


Εικόνα 4 Ταξινόμηση Ενισχυτικής Μάθησης [15]

1.3.1 Εκπαίδευση βασισμένη σε μοντέλο (Model Based)

Αρχικά θα πρέπει να οριστεί το τι είναι το μοντέλο του περιβάλλοντος, ώστε στη συνέχεια να περιγραφεί ο τρόπος με τον οποίο ένα μοντέλο ενισχυτικής μάθησης μαθαίνει από αυτό. Πρακτικά αναφερόμαστε στην μαθηματική περιγραφή του περιβάλλοντος στο οποίο δρα το

εκπαιδευόμενο σύστημα. Οι δοκιμές πιθανών αποφάσεων σε αυτό, δίνουν τα κατάλληλα δεδομένα για την πρόβλεψη της επόμενης ενέργειας και των αναμενόμενων ανταμοιβών. Είναι το μέσο για την δοκιμή ενεργειών ώστε να προβλεφθεί το αποτέλεσμα των συναρτήσεων επιβράβευσης και ανταμοιβής [16], [15]. Ωστόσο, δεν θα πρέπει να συγχέεται με την προσομοίωση του φυσικού συστήματος, το οποίο ελέγχεται από το μοντέλο ενισχυτικής μάθησης και προσομοιώνει την πραγματική εφαρμογή του αλγορίθμου. Τα σχεδιαγράμματα στην **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**, αναπαριστούν έναν α λγόριθμο Model Based και έναν Model Free κάνοντας εμφανή τη διαφορά τους.



Εικόνα 5 Στο πάνω σχέδιο εμφανίζεται το σχεδιάγραμμα ενός *Model Free* αλγορίθμου ενός στο κάτω ενός *Model Based*

Με τον όρο προσομοίωση, στους αλγορίθμους που βασίζονται σε μοντέλα, εννοούμε μία μαθηματική διαδικασία που διενεργεί ο αλγόριθμος προτού προβεί σε ενέργειες που βαθμολογούνται από τη συνάρτηση επιβράβευσης. Αυτή η διαδικασία προσπαθεί να προσομοιώσει την αντίδραση του περιβάλλοντος στις επικείμενες ενέργειες του αλγορίθμου. Εφόσον λοιπόν μέσω αυτού γίνουν γνωστές οι μεταβλητές (s, a, s'), το μοντέλο είναι έτοιμο να προβεί σε αποφάσεις-πράξεις, οι οποίες αλληλεπιδρούν με το φυσικό περιβάλλον (ή περιβάλλον προσομοίωσης ενός φυσικού συστήματος) και βαθμολογούνται. Το πλεονέκτημα αυτής της κατηγορίας, είναι ότι μπορεί να αντιπαραβάλει τις καταστάσεις που συναντά σε σχέση με αντίστοιχες της προσομοίωσης και να βρει την βέλτιστη λύση. Ωστόσο η απόδοση αυτών των αλγορίθμων σε δυναμικό περιβάλλον δεν είναι υψηλή, λόγω της δυσκολίας στην πρόβλεψη καταστάσεων κατά την εκμάθηση του μοντέλου. Οι δύο κύριες υποκατηγορίες αλγορίθμων βασιζόμενων σε μοντέλα είναι αυτές στις οποίες μαθαίνεται το μοντέλο *(learn the model)* και σε αυτές που δίνεται *(given the model)*.

Στους αλγορίθμους που δίνεται το μοντέλο, παρέχονται ουσιαστικά οι κανόνες που διέπουν το περιβάλλον και τη λειτουργία του αλγορίθμου. Η μέθοδος αυτή είναι αρκετά αποτελεσματική για περιβάλλοντα ή παιχνίδια των οποίων οι κανόνες μπορούν να προσδιοριστούν με σαφήνεια και δεν είναι ευμετάβλητα. Χαρακτηριστικό παράδειγμα είναι ο *AlphaZero* αλγόριθμος ο οποίος επιλύει παρτίδες σκάκι πετυχαίνοντας υπεράνθρωπες αποδόσεις [17].

Όσον αφορά στους αλγορίθμους που μαθαίνουν το μοντέλο, είναι αυτοί οι οποίοι εξερευνούν το περιβάλλον και τους κανόνες που το διέπουν εκτελώντας "πειράματα" στην διαδικασία πριν την εφαρμογή τον αποφάσεων σε φυσικό μοντέλο. Ο αλγόριθμος συλλέγει αυτά τα δεδομένα και στη συνέχεια δρα όπως στην προηγούμενη κατηγορία, μαθαίνοντας από αυτά τους κανόνες [16].

1.3.2 Εκπαίδευση δίχως την ύπαρξη μοντέλου (Model Free)

Οι αλγόριθμοι αυτής της κατηγορίας, δεν χρειάζονται κάποια προσομοιωμένη μορφή του περιβάλλοντος για να αλληλεπιδράσουν, αντιθέτως εφαρμόζονται απευθείας και βελτιώνονται από την αλληλεπίδραση υπό πραγματικές συνθήκες. Είναι κατάλληλοι για ταχέως μεταβαλλόμενα περιβάλλοντα, αλλά όπως είναι αναμενόμενο η εκπαίδευσή τους παρουσιάζει σημαντικές δυσκολίες. Χωρίς κάποια μηχανή προσομοίωσης του πραγματικού κόσμου (επεξηγήθηκε η διαφορά από τον μηχανισμό προσομοίωσης του αλγορίθμου) η εφαρμογή αλγορίθμων άνευ μοντέλου είναι υπολογιστικά ακριβή και δύσκολη. Για αυτό τον λόγο, τα τελευταία χρόνια έχουν γίνει άλματα στην έρευνα πάνω σε αυτόν τον τομέα, καθώς έχουν δημιουργηθεί αρκετές εφαρμογές επαυξημένης πραγματικότητας. Οι κατηγορίες που θα αναλυθούν, είναι αυτή των εξισώσεων τύπου Q-Learning ή εξισώσεων που δρουν βάσει μέγιστης αξίας, αυτών που δρουν βάσει βέλτιστης πολιτικής (Policy Optimization) και των αλγορίθμων που συνδυάζουν τα δύο παραπάνω δημιουργώντας τους αλγορίθμους Actor-Critic.

Οι Q-Learning αλγόριθμοι αποσκοπούν στη μεγιστοποίηση της αξίας που προκύπτει από τις συναρτήσεις αξίας και κατά κύριο λόγο χρησιμοποιούν τη συνάρτηση $Q^{\pi}(s,a)$, (1.10). Παρουσιάζουν ικανοποιητικές επιδόσεις ως προς τη διακύμανση της μέγιστης ανταμοιβής που στοχεύουν καθώς δεν παγιδεύονται σε τοπικά μέγιστα, ωστόσο δεν είναι κατάλληλη μέθοδος για προβλήματα πράξεων συνεχούς χώρου, όπως για παράδειγμα την πλοήγησης εναέριων μέσων.

Οι αλγόριθμοι βελτιστοποίησης πολιτικής, είναι αυτοί που σε κάθε βήμα τους αλλάζουν την πολιτική που έχουν και επιλέγουν αυτή που θα δώσει τη μέγιστη επιβράβευση. Οι αλγόριθμοι αυτοί παρουσιάζουν καλύτερη συμπεριφορά από τους Q-Learning σε περιπτώσεις συνεχούς ή πολυδιάστατου χώρου ενεργειών. Αυτό έγκειται στο γεγονός ότι δεν απαιτείται εξαιρετικά

αναλυτική παραμετροποίηση της πολιτικής καθότι μπορούν και τη μεταβάλλουν αναλόγως των αναγκών της εκάστοτε κατάστασης του περιβάλλοντος.

Το τρίτος είδος αυτής της κατηγορίας είναι η μέθοδος Actor-Critic, που συνδυάζει τα πλεονεκτήματα των δύο προηγουμένων, σε ένα πλαίσιο το οποίο θα μπορούσε να εκφραστεί ως "δράση και αξιολόγηση". Χρησιμοποιεί μεθόδους βασισμένους στη μέγιστη αξία ώστε να μάθει μία συνάρτηση τύπου Q-Learning, ενώ ταυτόχρονα βασιζόμενη στην βέλτιστη πολιτική, μεγιστοποιεί και την αντίστοιχη συνάρτηση. Όσον αφορά στον όρο "δράση – αξιολόγηση", έγκειται στο γεγονός ότι αυτού του είδους οι αλγόριθμοι έχουν δύο κύρια δομικά χαρακτηριστικά. Πρώτον είναι ο *ενεργών (actor)* ο οποίος βασιζόμενος στην παρούσα κατάσταση επιλέγει μία πράξη και ο *κριτής (critic)* ο οποίος την αξιολογεί και βάσει αυτής της ανάδρασης μαθαίνει ο πρώτος. Αυτά τα δύο στοιχεία μπορούν να είναι είτε συναρτήσεις, είτε και ολόκληρα νευρωνικά δίκτυα. Κύρια παραδείγματα τέτοιων αλγορίθμων είναι οι *DDPG*, *TD*3 και *SAC*, οι οποίοι και χρησιμοποιούνται στα πλαίσια αυτής της εργασίας και στη συνεχεία αναλύονται διεξοδικά.

1.3.3 Off-Policy & On-Policy

Προτού όμως γίνει η αναλυτική περιγραφή των αλγορίθμων, θα πρέπει να διευκρινιστεί ένας επιπλέον όρος. Πρόκειται για τον χαρακτηρισμό *on-policy* ή *off-policy* για κάποιον αλγόριθμο. Στην πρώτη κατηγορία ανήκουν αυτοί που εξερευνούν ακολουθώντας *ορισμένη πολιτική* την οποία και βελτιώνουν, αξιολογώντας την μόνο βάσει των δεδομένων που αυτή συγκέντρωσε. Αντιθέτως στη δεύτερη κατηγορία ανήκουν οι αλγόριθμοι οι οποίοι εξερευνούν *πιθανές ενέργειες όχι μόνο στα πλαίσια ορισμένη πολιτικής αλλά ακολουθώντας και άλλες τεχνικές όπως τυχαίες πράξεις*. Επιπλέον η αξιολόγηση της κάθε πολιτικής, γίνεται βάσει του συνόλου των συλλεγμένων δεδομένων.

1.3.4 Υπερπαράμετροι (Hyperparameters)

Με τον όρο υπερπαράμετροι (hyperparameters), αναφερόμαστε σε ένα σύνολο παραμέτρων ¨εργαλείων» που επηρεάζουν την λειτουργία του αλγορίθμου. Σε αναλογία, όπως σε μία μηχανή εσωτερικής καύσης μπορεί κάποιος να ρυθμίσει την ροή του καυσίμου στον κινητήρα και πληθώρα παρόμοιων παραμέτρων που επιδρούν στην λειτουργία και την απόδοση, έτσι μπορεί να ρυθμιστεί και η συμπεριφορά κάθε αλγορίθμου μέσω των υπερπαραμέτρων. Η επίδρασή τους στην εκπαίδευση και στη λειτουργία των μοντέλων ενισχυτικής μάθησης είναι βαρύνουσας σημασίας και η μελέτη τους αποτελεί ένα διευρυμένο πεδίο έρευνας.

Η επιλογή του βέλτιστου συνδυασμού παραμέτρων παρουσιάζει πολλές δυσκολίες και συνήθως προτείνεται να βρίσκεται η καλύτερη δυνατή λύση μέσω δοκιμών. Ωστόσο υπάρχουν έρευνες που εστιάζουν σε μεθόδους βελτιστοποίησης των υπερπαραμέτρων με την χρήση

διαφόρων μεθόδων όπως παρουσιάζονται στις [18], [19], [20]. Επίσης ένας ακόμα παράγοντας που δυσκολεύει την εύρεση σταθερών κανόνων, είναι το γεγονός ότι δεν υπάρχουν οι ίδιες υπερπαράμετροι σε κάθε αλγόριθμο. Στην παρούσα εργασία θα μας απασχολήσουν τρεις συγκεκριμένες παράμετροι, οι οποίες -ενδεχόμενός- παρουσιάζουν μεγαλύτερη επίδραση. Στη συνέχεια παρουσιάζονται και περιγράφεται συνοπτικά ο ρόλος τους.

Ο *Ρυθμός Εκμάθησης (learning rate)*, ελέγχει το κατά πόσο μεταβάλλονται οι παράγοντες του δικτύου σε κάθε βήμα της εκπαίδευσης. Κυμαίνεται από 0 έως 1 ανάγοντας σε ποσοστό κάθε φορά την μεταβολή. Υψηλή τιμή οδηγεί σε γρήγορο αποτέλεσμα αλλά πιθανότατα ασταθές, ενώ με χαμηλή τιμή η σύγκλιση είναι πιο αργή αλλά αποφέρει πιο αξιόπιστη λύση.

Ο *Συντελεστής Έκπτωσης γ (discount factor)*, λαμβάνει τιμές από 0 έως 1 και υποδεικνύει το βάρος που έχουν οι μελλοντικές επιβραβεύσεις σε σχέση με τις άμεσες. Τιμές κοντά στο 1 κάνουν τον αλγόριθμο να "σκέφτεται" μακροπρόθεσμα, ενώ τιμές κοντά στο 0 δίνουν βαρύτητα σε βραχυπρόθεσμες ανταμοιβές.

Το *Μέγεθος δεδομένων για εκμάθηση (replay buffer size) & το Μέγεθος δείγματος (batch size)*, αφορά τα δεδομένα που ο αλγόριθμος αποθηκεύει σε κάθε βήμα και επεισόδιο σε δεσμευμένο χώρο μνήμης και από εκεί λαμβάνεται ένα δείγμα κάθε φορά το οποίο και μελετά. Μεγάλο πλήθος αποθηκευμένων δεδομένων οδηγεί σε μεγαλύτερη εμπειρία του αλγορίθμου λόγω ποικιλίας. Ωστόσο πρέπει να βρίσκεται πάντοτε η χρυσή τομή καθώς όσο μεγαλώνει το δείγμα τόσος περισσότερος χώρος μνήμης αποθηκεύεται και τόσο περισσότερη υπολογιστική ισχύς χρειάζεται.

1.4 Deep Deterministic Policy Gradient (DDPG)

Ο Deep Deterministic Policy Gradient [21] έχει βασιστεί στους Deterministic Policy Gradient αλγορίθμους [22] και είναι ένας *αιτιοκρατικός off-policy αλγόριθμος actor-critic*, που μαθαίνει ταυτόχρονα μία Q συνάρτηση και μία πολιτική, χρησιμοποιώντας την συνάρτηση 1.16, επιλέγοντας πάντα την ενέργεια που μεγιστοποιεί άμεσα το αποτέλεσμα (1.20).

Ο DDPG βελτιστοποιεί συνεχείς δράσεις, υπολογίζοντας όπως αναφέρθηκε την μέγιστη αμοιβή κάθε φορά. Με μια πρώτη ματιά θα μπορούσε κανείς να πει ότι το υπολογιστικό κόστος είναι απαγορευτικό, καθώς σε συνεχή χώρο οι πιθανές πράξεις είναι πολλές. Ωστόσο αντί να υπολογίζει συνεχώς την συνάρτηση $\max_a Q^*(s,a)$, θεωρείται η μεγιστοποίηση της μεταβλητής a ως το αποτέλεσμα της πολιτική $\mu_\theta(s)$ και ότι η Q είναι παραγωγίσιμη ως προς α, η συνάρτηση που πρέπει να υπολογιστεί είναι η $Q(s,\mu(s))$.

Όσον αφορά στην πολιτική $\mu_\theta(s)$ του DDPG, αυτή είναι αιτιοκρατική και στόχο έχει την μεγιστοποίηση κάθε φορά της $Q_\varphi(s,a)$. Η συνάρτηση της πολιτικής λοιπόν προς μεγιστοποίηση είναι η:

$$\max_{\theta} E[\, Q_{\varphi}(s, \mu_{\theta}(s))] \tag{1.22}$$

Θεωρώντας ότι με την συνάρτηση 1.16 εκκινεί η διαδικασία μάθησης μέσω ενός νευρωνικού δικτύου $Q_{\varphi}(s,a)$ με παραμέτρους στόχους φ, παράγεται ένα σύνολο $D(s,a,r,s',d)$ που είναι το δείγμα δεδομένων για την εκμάθηση, με d μία μεταβλητή boolean που υποδεικνύει αν η κατάσταση s' είναι τερματική. Με τα παραπάνω δεδομένα μπορεί να οριστεί η συνάρτηση σφάλματος ελαχίστων τετραγώνων Bellman $L(\varphi, D)$, που δείχνει πόσο κοντά έχει φτάσει το δίκτυο στο βέλτιστο. Στόχος κατά την εκπαίδευση είναι η ελαχιστοποίηση της παρακάτω συνάρτησης.

$$L(\varphi, D) = E[(Q_{\varphi}(s,a) - (r + \gamma(1-d)\max_{a'} Q_{\varphi}(s',a')))^2] \tag{1.23}$$

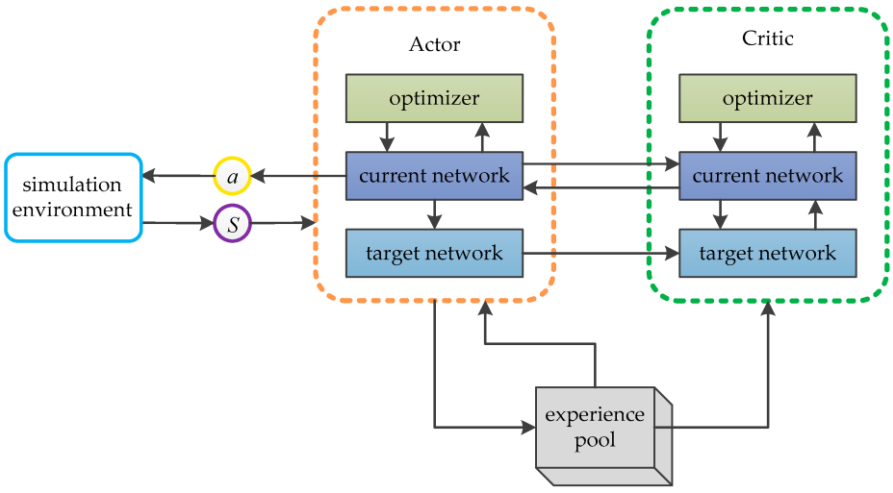
Ο όρος $(r + \gamma(1-d)\max_{a'} Q_{\varphi}(s',a'))$, συμβολίζεται ως φ_{targ} και ονομάζεται *δίκτυο στόχος (target network)* και αυτό γιατί είναι η ποσότητα που προσπαθεί να προσεγγίσει ο αλγόριθμος για την ελαχιστοποίηση της $L(\varphi, D)$. Το φ_{targ} ενημερώνεται σε κάθε ενημέρωση του κυρίως νευρωνικού δικτύου, βάσει της παρακάτω έκφρασης και συντελεί στην απλοποίηση των υπολογισμών για την εύρεση της βέλτιστης ενέργειας.

$$\varphi_{targ} \leftarrow \rho\varphi_{targ} + (1-\rho)\varphi \tag{1.24}$$

Όπου ρ, η υπερπαράμετρος polyak που λαμβάνει τιμές από 0 έως 1 και ορίζει το κατά πόσο μεταβάλλεται σε βήμα το φ_{targ} και φ οι παράμετροι-στόχοι. Συνδυάζοντας όλα τα παραπάνω και την πολιτική-στόχο ως $\mu_{\theta_{targ}}$, η συνάρτηση 1.23 μπορεί να εκφραστεί ως:

$$L(\varphi, D) = E\left[\left(Q_{\varphi}(s,a) - \left(r + \gamma(1-d)Q_{\varphi_{targ}}(s', \mu_{\theta_{targ}}(s'))\right)\right)^2\right] \tag{1.25}$$

Ο τρόπος λειτουργία τους αλγορίθμου DDPG, παρουσιάζεται στην Εικόνα 6, ενώ στον Πίνακα 1, ο ψευδοκώδικας λειτουργίας του.



Εικόνα 6 Σχηματική αναπαράσταση του DDPG [8]

Πίνακας 1 Ψευδοκώδικας DDPG [15]

1. Εισάγαγε παραμέτρους φ για την Q, θ για την πολιτική και κενό δείγμα δεδομένων D
2. Όρισε παραμέτρους στόχους $\varphi_{targ} \leftarrow \varphi_{targ}$ και $\theta_{targ} \leftarrow \theta_{targ}$
3. **Επανάλαβε**
    a. Παρατήρησε κατάσταση s και επίλεξε ενέργεια $a = clip\big(\mu_\theta(s) + \epsilon, a_{Low}, a_{High}\big),\ \epsilon \sim N$
    b. Έλεγξε ενέργεια α
    c. Έλεγξε επόμενη κατάσταση s', επιβράβευση r και όρισε $d = 1$ αν η s' τερματική, αλλιώς $d = 0$
    d. Αποθήκευσε (s, a, r, s', d) στο D
    e. Αν s' τερματική, ανανέωσε το περιβάλλον
    f. **Αν** είναι ώρα για ενημέρωση του δικτύου **τότε**
        i. **Για** όσες είναι οι ενημερώσεις **κάνε**
            1. Τυχαία επίλεξε δείγμα $B = \{(s, a, r, s', d)\} \in D$
            2. Υπολόγισε τους στόχους $y(r, s', d) = r + \gamma(1 - d)Q_{\varphi_{targ}}(s', \mu_{\theta_{targ}}(s'))$
            3. Ανανέωσε την συνάρτηση Q κατά ένα βήμα χρησιμοποιώντας: $\nabla_\varphi \frac{1}{|B|}\sum_{(s,a,r,s',d)}(Q_\varphi(s, a) - y(r, s', d))^2$
            4. Ανανέωσε την πολιτική κατά ένα βήμα χρησιμοποιώντας: $\nabla_\varphi \frac{1}{|B|}\sum_{s\in B} Q_\varphi(s, \mu_\theta(s))$
            5. Ανανέωσε το δίκτυο-στόχο κατά ένα βήμα χρησιμοποιώντας:

$$\varphi_{targ} \leftarrow \rho\varphi_{targ} + (1 - \rho)\varphi$$

$$\theta_{targ} \leftarrow \rho\theta_{targ} + (1 - \rho)\theta$$

        ii. **Τέλος Για**
    g. **Τέλος Αν**
4. **Έως** σύγκλισης παραμέτρων

1.5 Twin Delayed DDPG (TD3)

Ο Twin Delayed DDPG (TD3) [23], είναι μία παραλλαγή με στόχο την βελτίωση του DDPG ο οποίος σε πολλές περιπτώσεις παρουσιάζει αστάθειες, είτε γιατί παγιδεύεται σε τοπικά μέγιστες λύσεις, είτε γιατί εξειδικεύεται πάνω σε μία ενέργεια (overfitting). Λειτουργεί και αυτός μόνο σε συνεχή χώρο δράσης και ο τρόπος με τον οποίο λειτουργεί είναι παρόμοιος με τον προηγούμενο αλγόριθμο, Η διαφορά, έγκειται στο γεγονός ότι ο TD3 έχει δύο Q συναρτήσεις $Q_{\varphi 1}$ και $Q_{\varphi 2}$ (Twin), απ' όπου και επιλέγεται το μικρότερο από τα δύο αποτελέσματα-στόχος,

$$y(r, s', d) = r + \gamma(1 - d) \min_{i=1,2} Q_{\varphi i,targ}(s', \alpha'(s')). \tag{1.26}$$

Παράλληλα, δεν ανανεώνει την πολιτική του σε κάθε ανανέωση της Q αλλά τοποθετείται μία καθυστέρηση (Delayed), ενώ στην ενέργεια στόχο προσθέτει θόρυβο ώστε να μειώνεται ο κίνδυνος των τοπικών μεγίστων.

$$\alpha'(s') = clip\left(\mu_{\theta_{targ}}(s') + clip(\epsilon, -c, c), a_{Low}, a_{High}\right), \ \epsilon \sim N \tag{1.27}$$

Ωστόσο η εκμάθηση της πολιτικής παραμένει όπως στον DDPG, μεγιστοποιώντας την $Q_{\varphi 1}$ μέσω της συνάρτησης 1.22 και τέλος για τις δύο συναρτήσεις υπολογίζεται το σφάλμα μέσων τετραγώνων Bellman.

$$L(\varphi_i, D) = E\left[\left(Q_{\varphi i}(s, a) - y(r, s', d)\right)^2\right], \ i = 1,2 \tag{1.28}$$

Στον Πίνακα 2 παρουσιάζεται ο ψευδοκώδικας λειτουργίας του αλγορίθμου TD3.

Πίνακας 2 Ψευδοκώδικας TD3 [15]

1. Εισάγαγε παραμέτρους φ_1, φ_2 για την Q, θ για την πολιτική και κενό δείγμα δεδομένων D
2. Όρισε παραμέτρους στόχους $\varphi_{targ,1} \leftarrow \varphi_{targ,1}$, $\varphi_{targ,2} \leftarrow \varphi_{targ,2}$ και $\theta_{targ} \leftarrow \theta_{targ}$
3. **Επανάλαβε**
    a. Παρατήρησε κατάσταση s και επίλεξε ενέργεια $a = clip\big(\mu_\theta(s) + \epsilon, a_{Low}, a_{High}\big),\ \epsilon \sim N$
    b. Έλεγξε ενέργεια α
    c. Έλεγξε επόμενη κατάσταση s', επιβράβευση r και όρισε $d = 1$ αν η s' τερματική, αλλιώς $d = 0$
    d. Αποθήκευσε (s, a, r, s', d) στο D
    e. Αν s' τερματική, ανανέωσε το περιβάλλον
    f. **Αν** είναι ώρα για ενημέρωση του δικτύου **τότε**
        i. **Για** j σε διάστημα ίσο με τις ανανεώσεις **κάνε**
            1. Τυχαία επίλεξε δείγμα $B = \{(s, a, r, s', d)\} \in D$
            2. Υπολόγισε πράξεις στόχους

$$\alpha'(s') = clip\left(\mu_{\theta_{targ}}(s') + clip(\epsilon, -c, c), a_{Low}, a_{High}\right),\ \epsilon \sim N(0, \sigma)$$

            3. Υπολόγισε τους στόχους $y(r, s', d) = r + \gamma(1 - d) \min_{i=1,2} Q_{\varphi i, targ}(s', \alpha'(s'))$
            4. Ανανέωσε την συνάρτηση Q κατά ένα βήμα χρησιμοποιώντας: $\nabla_{\varphi i} \frac{1}{|B|} \sum_{(s,a,r,s',d)} (Q_{\varphi i}(s, a) - y(r, s', d))^2,\ i = 1,2$
            5. **Αν** j mod καθυστέρηση_πολιτικής = 0 **τότε**
                a. Ανανέωσε την πολιτική κατά ένα βήμα χρησιμοποιώντας:
                $\nabla_\varphi \frac{1}{|B|} \sum_{s \in B} Q_{\varphi 1}(s, \mu_\theta(s))$
                Ανανέωσε το δίκτυο-στόχο κατά ένα βήμα χρησιμοποιώντας:
                $\varphi_{targ,i} \leftarrow \rho\varphi_{targ,i} + (1 - \rho)\varphi_{targ,i}, \quad i = 1,2$
                $\theta_{targ,i} \leftarrow \rho\theta_{targ,i} + (1 - \rho)\theta_{targ,i}$
            6. **Τέλος Αν**
        ii. **Τέλος Για**
    g. **Τέλος Αν**
4. **Έως** σύγκλισης παραμέτρων

1.6 Soft Actor Critic (SAC)

Ο Soft Actor Critic (SAC) [24], ανήκει στην ίδια κατηγορία με τους δύο προηγούμενους αλγορίθμους, με την διαφορά όμως ότι είναι στοχαστικός και όχι αιτιοκρατικός όπως οι προηγούμενοι. Λειτουργεί αποκλειστικά σε συνεχείς χώρους και χρησιμοποιεί και αυτός μία διπλή συνάρτηση Q, ωστόσο η σημαντική διαφορά στην εκτέλεσή του είναι η χρήση ενός ρυθμιστή της εντροπίας. Ουσιαστικά πέραν του μεγίστου αποτελέσματος προσπαθεί να ρυθμίσει την πολιτική κατά τέτοιον τρόπο ώστε να εξερευνά σειρές δράσεων με τα μεγαλύτερα δυνατά αποτελέσματα. Η εντροπία είναι χονδρικά το μέτρο τυχαιότητας κάθε επιλογής του αλγορίθμου και μαθηματικά μιλώντας αν η P είναι η συνάρτηση μάζας πιθανότητας της τυχαίας μεταβλητής x τότε δίνεται από τον τύπο:

$$H(P) = E\left[-\log P(\chi)\right] \tag{1.29}$$

Για τους αλγορίθμους που ρυθμίζουν την εντροπία σε κάθε βήμα ανάλογό της επιβραβεύονται και η συνάρτηση της βέλτιστης πολιτικής μετατρέπεται ως ακολούθως:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t, a_t, s_{t+1}) + \alpha H\big(\pi(\cdot\,|s_t)\big)\right)\right] \tag{1.30}$$

Όπου $\alpha > 0$ είναι η υπερπαράμετρος που ορίζει την ισορροπία μεταξύ εντροπίας και αναμενόμενης ανταμοιβής και αναλόγως την εφαρμογή μπορεί να είναι σταθερά ή μεταβλητή. Όπως μεταβλήθηκε η συνάρτηση της βέλτιστης πολιτικής, αντιστοίχως θα μεταβληθούν και οι συναρτήσεις αξίας ώστε να συμπεριλάβουν την επιβράβευση για την εντροπία.

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) + a\sum_{t=0}^{\infty} \alpha H\big(\pi(\cdot\,|s_t)\big)\,|s_0 = s\right] \tag{1.31}$$

$$Q^{\pi}(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) + a\sum_{t=0}^{\infty} \alpha H\big(\pi(\cdot\,|s_t)\big)\,|s_0 = s, \alpha_0 = \alpha\right] \tag{1.32}$$

Οι εξισώσεις (31) και (32) ενώνονται μέσω της επόμενης σχέσης:

$$V^{\pi}(s) = E\left[Q^{\pi}(s,a)\right] + \alpha H\big(\pi(\cdot\,|s_t)\big) \tag{1.33}$$

Έχοντας υπόψη τα παραπάνω και η συνάρτηση Bellman για την Q^{π} μεταβάλλεται.

$$Q^{\pi}(s,a) = E\left[R(s,a,s') + \gamma\left(Q^{\pi}(s',a') + \alpha H\big(\pi(\cdot\,|s_t)\big)\right)\right] = E\left[R(s,a,s') + \gamma V^{\pi}(s')\right] \tag{1.34}$$

Εάν όμως η εντροπία γραφεί βάσει του ορισμού της 1.29, η συνάρτηση 1.34 μπορεί να περιγραφεί και ως:

$$Q^{\pi}(s,a) = E\left[R(s,a,s') + \gamma\big(Q^{\pi}(s',a') - \alpha\log\big(\pi(\alpha'|s_t)\big)\big)\right] \tag{1.35}$$

Και συνεπώς προκύπτει ότι:

$$Q^{\pi}(s,a) \approx r + \gamma\left(Q^{\pi}(s',a') - \alpha \log\left(\pi(\tilde{\alpha}'|s_t)\right)\right),\ \tilde{\alpha}' \sim \pi_{\theta}(\cdot\,|s') \tag{1.36}$$

Η παραπάνω περιγραφή ανήκει στους περισσότερους αλγορίθμους που χρησιμοποιούν την εντροπία για την εκπαίδευσή τους. Όσον αφορά συγκεκριμένα στον SAC, θα πρέπει να σημειωθεί ότι χρησιμοποιείται και εδώ διπλή συνάρτηση Q με $Q_{\varphi 1}$ και $Q_{\varphi 2}$ και ο στόχος υπολογίζεται χρησιμοποιώντας τη συνάρτηση 1.24. Όμως ο παρών αλγόριθμος δεν έχει πολιτική στόχο, αλλά επιλέγει πάντα με την τρέχουσα πολιτική και λόγω του θορύβου που προέρχεται από την στοχαστικότητά του δεν προστίθεται επιπλέον θόρυβος όπως στον TD3. Όσον αφορά στο σφάλμα μέσω τετραγώνων Bellman θα είναι ακριβώς η ίδια συνάρτηση με την 1.28, αλλά σχετικά με τον υπολογισμό των στόχων η συνάρτηση θα μεταβληθεί ως εξής:

$$y(r,s',d) = r + \gamma(1-d)\left(\min_{i=1,2} Q_{\varphi i,targ}\,(s',\tilde{\alpha}') - \alpha \log\left(\pi(\tilde{\alpha}'|s_t)\right)\right),\ \ \tilde{\alpha}' \sim \pi_{\theta}(\cdot\,|s') \tag{1.37}$$

Η πολιτική του αλγορίθμου σε κάθε βήμα θα πρέπει να δρα με στόχο τη μεγιστοποίηση των μελλοντικών αμοιβών και της εντροπίας. Κατ' αυτόν τον τρόπο η συνάρτηση αξίας παίρνει την μορφή που ακολουθεί:

$$V^{\pi}(s) = E\,[Q^{\pi}(s,a)] + \alpha H\left(\pi(\cdot\,|s_t)\right) = E\,[Q^{\pi}(s,a) - \alpha \log \pi(a|s)] \tag{1.38}$$

Για την βελτιστοποίηση της πολιτικής υπάρχει μία δικλείδα όπου στο δείγμα δεδομένων για εκπαίδευση κάποιες πράξεις $\tilde{\alpha}_{\theta}(s,\xi)$, $\xi \sim N(0,1)$, εισάγονται έχοντας υπολογιστεί με μία αιτιοκρατική πολιτική $\mu_{\theta}(s)$ της επιλογής του χρήστη, μεταμορφώνοντας το αναμενόμενο αποτέλεσμα ως εξής:

$$E\,[Q^{\pi_{\theta}}(s,a) - \alpha \log \pi(a|s)] = E\left[Q^{\pi_{\theta}}\left(s,\tilde{\alpha}_{\theta}(s,\xi)\right) - \alpha \log \pi(\tilde{\alpha}_{\theta}(s,\xi)|s)\right] \tag{1.39}$$

Τέλος για να ολοκληρωθεί η μαθηματική περιγραφή πρέπει να εισαχθεί η διπλή συνάρτηση $Q_{\varphi i}$ στην παραπάνω έκφραση, μόνο που αντί να χρησιμοποιείται το αποτέλεσμα της $Q_{\varphi 1}$ για τον υπολογισμό, χρησιμοποιείται το ελάχιστο από τις $Q_{\varphi 1}$ και $Q_{\varphi 2}$.

$$\max_{\theta} E\left[\min_{i=1,2} Q_{\varphi i}\left(s,\tilde{\alpha}_{\theta}(s,\xi)\right) - \alpha \log \pi(\tilde{\alpha}_{\theta}(s,\xi)|s)\right] \tag{1.40}$$

Στον πίνακα 3 παρουσιάζεται ο τρόπος λειτουργίας του αλγορίθμου SAC.

Πίνακας 3 Ψευδοκώδικας SAC [15]

1. Εισάγαγε παραμέτρους φ_1, φ_2 για την Q, θ για την πολιτική και κενό δείγμα δεδομένων D
2. Όρισε παραμέτρους στόχους $\varphi_{targ,1} \leftarrow \varphi_{targ,1}$, $\varphi_{targ,2} \leftarrow \varphi_{targ,2}$ και $\theta_{targ} \leftarrow \theta_{targ}$
3. **Επανάλαβε**
    a. Παρατήρησε κατάσταση s και επίλεξε ενέργεια $a \sim \pi_\theta(\cdot|s)$
    b. Έλεγξε ενέργεια α
    c. Έλεγξε επόμενη κατάσταση s', επιβράβευση r και όρισε $d = 1$ αν η s' τερματική, αλλιώς $d = 0$
    d. Αποθήκευσε (s, a, r, s', d) στο D
    e. Αν s' τερματική, ανανέωσε το περιβάλλον
    f. **Αν** είναι ώρα για ενημέρωση του δικτύου **τότε**
        i. Για j σε διάστημα ίσο με τις ανανεώσεις **κάνε**
            1. Τυχαία επίλεξε δείγμα $B = \{(s, a, r, s', d)\} \in D$
            2. Υπολόγισε τους στόχους

$$y(r, s', d) = r + \gamma(1 - d)\left(\min_{i=1,2} Q_{\varphi i,targ}(s', \tilde{\alpha}') - log\pi_\theta(\tilde{\alpha}'|s')\right),\ \tilde{\alpha}' \sim \pi_\theta(\cdot|s)$$

            3. Ανανέωσε την συνάρτηση Q κατά ένα βήμα χρησιμοποιώντας: $\nabla_{\varphi i} \frac{1}{|B|}\sum_{(s,a,r,s',d)}\left(Q_{\varphi i}(s, a) - y(r, s', d)\right)^2,\ i = 1,2$

            4. Ανανέωσε την πολιτική κατά ένα βήμα χρησιμοποιώντας:

$$\nabla_\varphi \frac{1}{|B|}\sum_{s\in B}\left(\min_{i=1,2} Q_{\varphi 1}(s, \mu_\theta(s)) - a\log\pi_\theta(\tilde{\alpha}_\theta(s)|s)\right), \text{ όπου } \tilde{\alpha}_\theta(s) \sim \pi_\theta(\cdot|s)$$

            Ανανέωσε το δίκτυο-στόχο κατά ένα βήμα χρησιμοποιώντας:

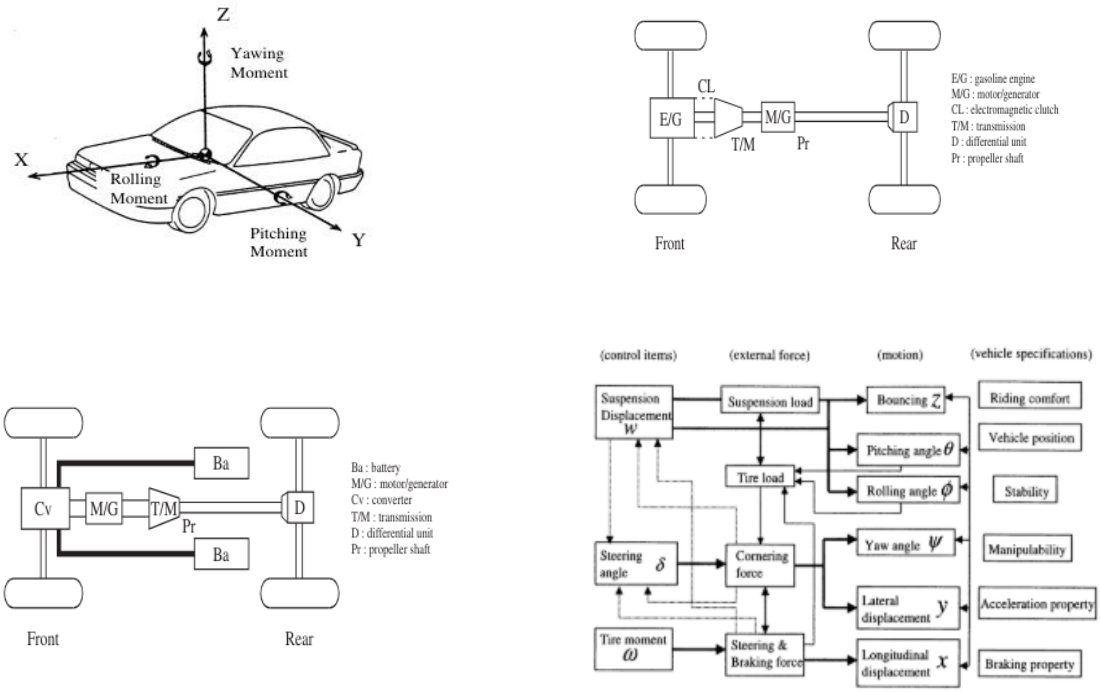
$$\varphi_{targ,i} \leftarrow \rho\varphi_{targ,i} + (1 - \rho)\varphi_{targ,i}, \qquad i = 1,2$$

            5. Τέλος Αν
        ii. Τέλος Για
    g. Τέλος Αν
4. **Έως** σύγκλισης παραμέτρων

1.7 Βιβλιογραφική Ανασκόπηση

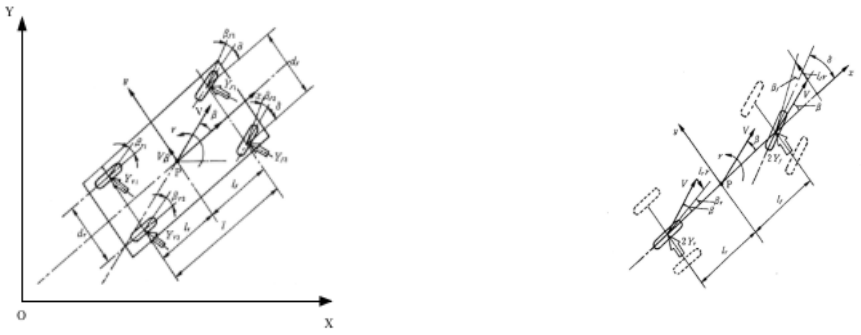
Το παρόν κεφάλαιο έχει στόχο να κάνει μία αναδρομή σε υπάρχουσες εργασίες πάνω στο πεδίο των αυτόνομων ηλεκτρικών αυτοκινήτων, ξεκινώντας από πρότυπες εφαρμογές μαθηματικού ελέγχου και καταλήγοντας σε πλήρεις πλατφόρμες πειραματισμού πάνω στον έλεγχο με ενισχυτική μάθηση.

Στην εργασία [25] προτείνεται ένα μαθηματικό μοντέλο για τη μοντελοποίηση και τον έλεγχο αυτόνομων ηλεκτρικών οχημάτων. Εξετάζεται ένα αμιγώς ηλεκτρικό όχημα και ένα υβριδικό, των οποίων τα σχεδιαγράμματα φαίνονται στην Εικόνα 7.

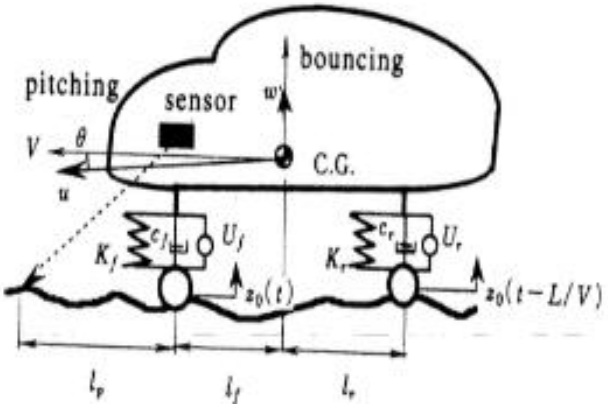


Εικόνα 7 Πάνω Αριστερά: Γωνίες Euler, Πάνω Δεξιά: Σύστημα ισχύος υβριδικού ηλεκτρικού οχήματος, Κάτω Αριστερά: Σύστημα ισχύος αμιγώς ηλεκτρικού οχήματος, Κάτω Δεξιά: Σχεδιάγραμμα συστήματος ελέγχου [25]

Στην εργασία αυτή προτάθηκε ένα σύστημα τετραδιεύθυνσης και ένα απλό σύστημα όπου στρίβουν μόνο οι εμπρόσθιοι τροχοί, ο έλεγχος περιελάβανε και το σύστημα ανάρτησης αλλά και το σύστημα ισχύος.

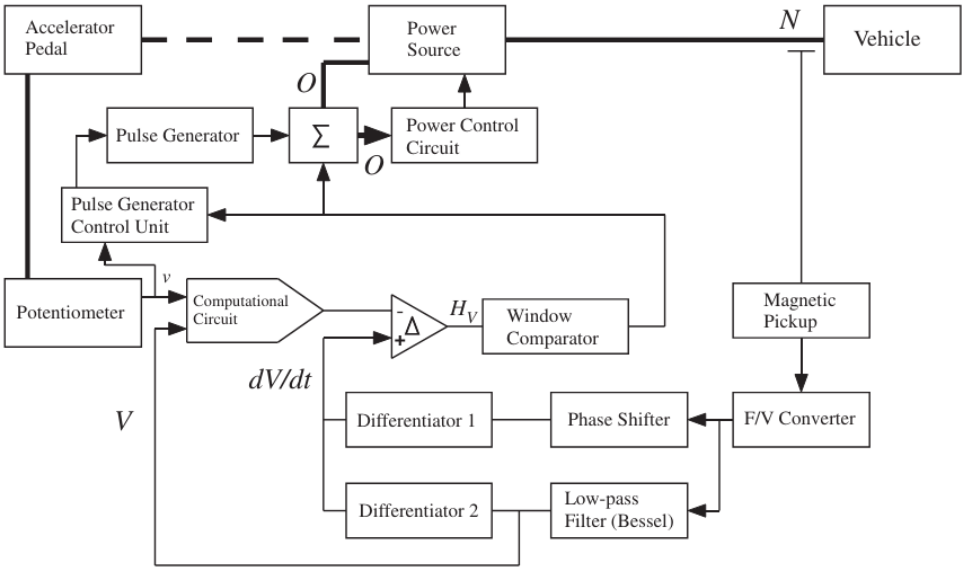


Εικόνα 8 Αριστερά: Σύστημα τετραδιεύθυνσης, Δεξιά: Σύστημα διεύθυνσης δύο τροχών [25]



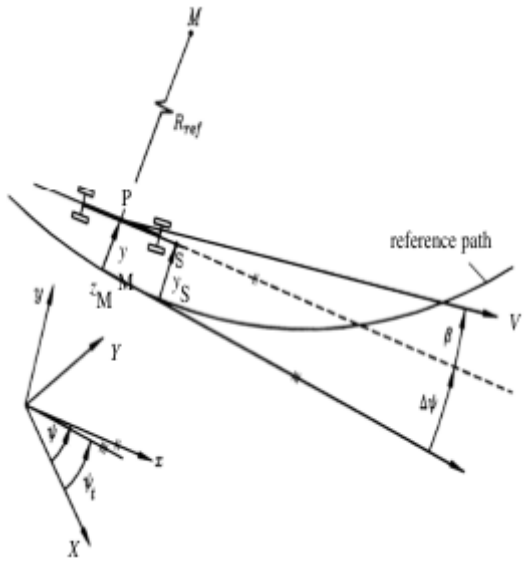
Εικόνα 9 Έλεγχος ανάρτησης [25]

Το εκτεταμένο μοντέλο περιείχε μη γραμμικούς παράγοντες για την διεύθυνση του οχήματος και εισήχθη ένας βέλτιστος ρυθμιστής για αυτό, ώστε το όχημα να μπορεί να κινηθεί κατά μήκος της επιθυμητής πορείας.



Εικόνα 10 Σχεδιάγραμμα ελέγχου οχήματος [25]

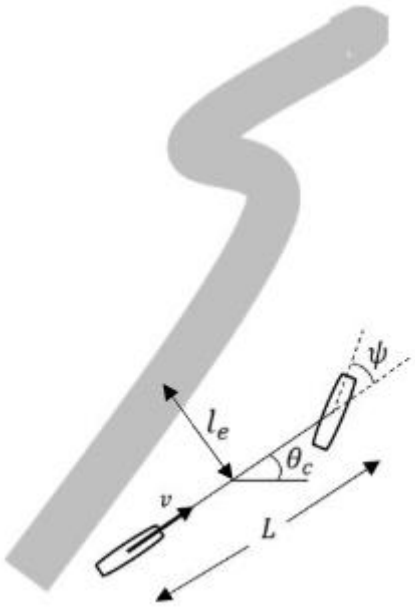
Ο προτεινόμενος ελεγκτής, προορίζεται για αυτόνομα οχήματα στα οποία δεν επιβαίνουν επιβάτες, καθώς δεν φαίνεται να προσφέρει άνεση κατά την μετακίνηση, επειδή υπάρχουν πολλά προβλήματα (ταλαντώσεις, απότομο στρίψιμο κ.λπ.). Ακόμα στην εργασία αυτή προτείνεται μία μέθοδος ελέγχου για τη βελτιστοποίηση των ελευθεριών λειτουργίας μεταξύ δύο μονάδων ισχύος του υβριδικού οχήματος, η οποία είναι μια από τις πιθανές λύσεις για την πρόληψη της περιβαλλοντικών μολύνσεων που προκαλείται από αυτού του είδους τα οχήματα.



Εικόνα 11 Σχέδιο αυτόματου συστήματος διεύθυνσης οχήματος [25]

Στην παρούσα εργασία ωστόσο το ενδιαφέρον εστιάζεται στο σύστημα ελέγχου, που προτάθηκε καθώς το έτος δημοσίευσής της είναι το 2012, και έκτοτε υπήρξαν αρκετές παρόμοιες προτάσεις, που διόρθωσαν τα προβλήματα που προέκυψαν είτε στο απότομο στρίψιμο, είτε στην μη ικανοποιητική απόσβεση των κραδασμών. Ωστόσο η μαθηματική μοντελοποίηση που έγινε αποτελεί ενδιαφέρουσα προσέγγιση και χρήσιμη σε αντίστοιχες εφαρμογές.

Στην δημοσίευση [26], χρησιμοποιείται ένας ελεγκτής για τον έλεγχο της κίνησης του οχήματος, μέσω μιας προκαθορισμένης διαδρομής που εξετάζει διαφορετικά σενάρια οδήγησης παρόμοια με αυτά του πραγματικού κόσμου. Στο πρώτο μέρος της εργασίας, ένας μη γραμμικός ελεγκτής που περιέχει μια σχέση μεταξύ πλευρικών σφαλμάτων, σφάλματος κατεύθυνσης και ταχύτητας οχήματος έχει σχεδιαστεί για να δημιουργήσει μια κατάλληλη γωνία διεύθυνσης για το όχημα ώστε να επιτύχει καλή απόδοση όσον αφορά τη διαδρομή που ακολουθεί με το ελάχιστο σφάλμα απόστασης. Ο προτεινόμενος ελεγκτής επικυρώνεται με προσομοίωση κάτω από διαφορετικές τιμές σταθερών ταχυτήτων.



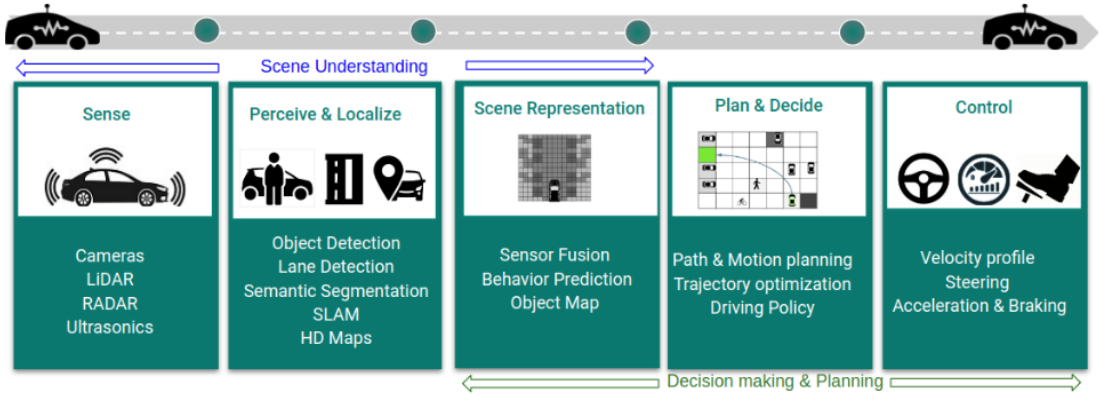
Εικόνα 12 Παράμετροι πλευρικού ελέγχου [26]

Στο δεύτερο μέρος της εργασίας, η στρατηγική γραμμικοποίησης ανάδρασης κατάστασης Εισόδου-Εξόδου, χρησιμοποιείται για τη δημιουργία γραμμικής ταχύτητας που χρησιμοποιείται σε συνδυασμό με την γωνία διεύθυνσης για τον έλεγχο της κίνησης του οχήματος και την ελαχιστοποίηση του πλευρικού σφάλματος. Τα ληφθέντα αποτελέσματα δείχνουν καλή απόδοση με μικρό σφάλμα απόστασης, σε σύγκριση με τις σταθερές ταχύτητες, ενώ η σταθερότητα του συστήματος επαληθεύεται χρησιμοποιώντας την προσέγγιση *Lyapunov*.

Τα προσομοιωμένα αποτελέσματα έδειξαν την καλή απόδοση του σχεδιασμένου ελεγκτή για διαφορετικές σταθερές ταχύτητες. Η στρατηγική γραμμικοποίησης ανάδρασης κατάστασης εισόδου-εξόδου χρησιμοποιείται στο δεύτερο μέρος για τη δημιουργία μιας κατάλληλης γραμμικής ταχύτητας η οποία ενσωματώνεται με τον μη γραμμικό ελεγκτή για να επιτύχει το ελάχιστο σφάλμα απόστασης και να κάνει το όχημα να συγκλίνει περισσότερο προς τη διαδρομή αναφοράς. Τα αποτελέσματα της εργασίας, πέραν του ποσοτικού χαρακτηρισμού τους που είναι αρκετά θετικός μιας και το σφάλμα απόκλισης από την πορεία είναι γενικώς ελάχιστο, το είδος των γραφημάτων παρουσιάζει ενδιαφέρον. Εισάγεται μια πολύ καλή μετρητική διάσταση για την επιτυχία της εφαρμογής, το κατά πόσο ακολουθείται η προδιαγεγραμμένη πορεία και ποιο είναι το σφάλμα επ’ αυτής. Όπως θα δει και ο αναγνώστης στην πορεία αντίστοιχα διαγράμματα παρουσιάζονται και για τα αποτελέσματα της εφαρμογής για την οποία συγγράφεται η εργασία αυτή.

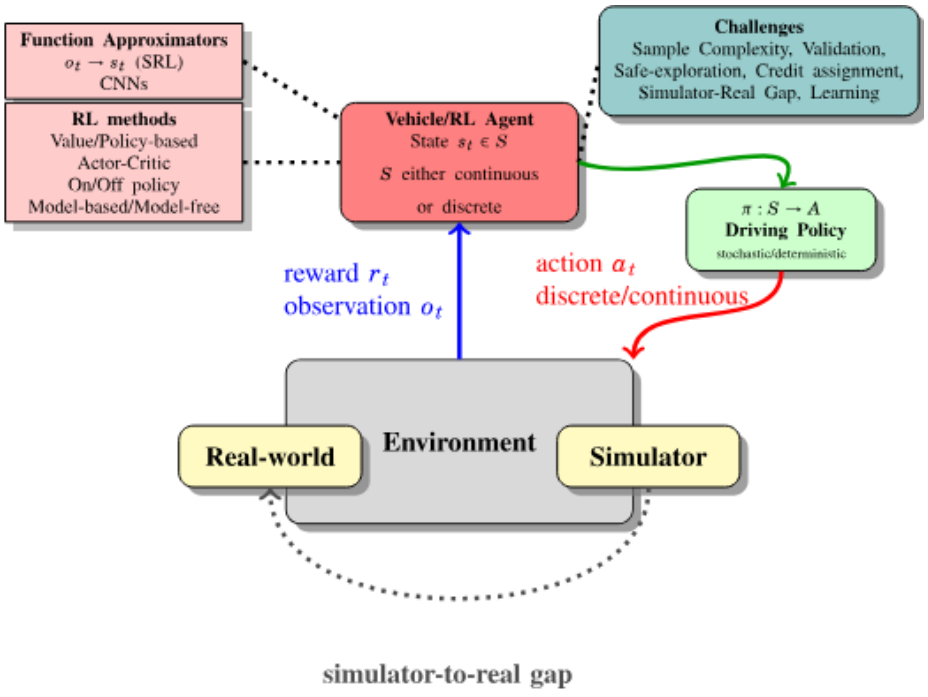
Στη δημοσίευση [26], γίνεται εισαγωγή στην αυτόνομη οδήγηση με ενισχυτική μάθηση, η οποία υπογραμμίζει ότι είναι ένα ισχυρό πλαίσιο ικανό να μάθει περίπλοκες πολιτικές σε περιβάλλοντα πολλών διαστάσεων. Παρέχεται μια ολοκληρωμένη ταξινόμηση των

αυτοματοποιημένων εργασιών οδήγησης όπου έχουν εφαρμοστεί μέθοδοι DRL, δίνοντας έμφαση στην αποτελεσματικότητά τους σε σενάρια που απαιτούν δυναμική λήψη αποφάσεων και αλληλεπίδραση μεταξύ των παραγόντων.



Εικόνα 13 Τυπικά στάδια σε ένα σύγχρονο σύστημα αυτόνομης οδήγησης, που απαριθμεί τις διάφορες εργασίες [27]

Αναφέρονται επίσης βασικές υπολογιστικές προκλήσεις για την ανάπτυξη αυτόνομων πρακτόρων οδήγησης σε πραγματικές συνθήκες, συμπεριλαμβανομένου του ρόλου των προσομοιωτών στην εκπαίδευση, μεθόδων αξιολόγησης, δοκιμών και στρατηγικών για την ενίσχυση της ευρωστίας των λύσεων RL. Παρά τις αξιοσημείωτες εμπορικές επιτυχίες, η ανασκόπηση υπογραμμίζει την έλλειψη εκτεταμένης βιβλιογραφίας και μεγάλης κλίμακας δημοσιευμένων δεδομένων σε αυτόν τον τομέα, σημειώνοντας την ανάγκη για ένα δομημένο πλαίσιο για την οργάνωση των εφαρμογών RL στην αυτόνομη οδήγηση.



Εικόνα 14 Γραφική αναπαράσταση των στοιχείων ενός RL αλγορίθμου με τις προκλήσεις που υπάρχουν κατά την εκπαίδευση [27]

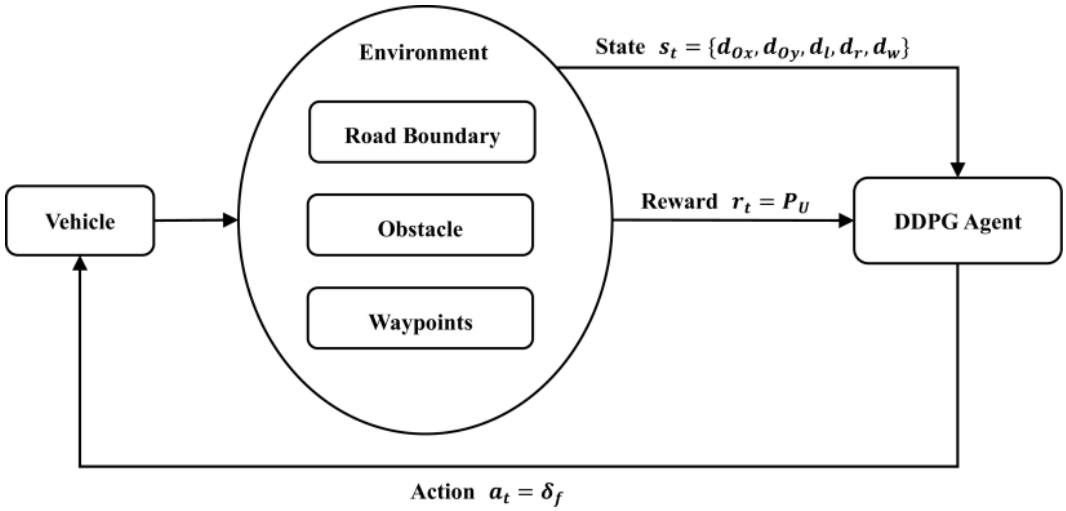
Ενώ τονίζεται η καταλληλόλητα της RL για την λήψη αποφάσεων που είναι εγγενείς στην αυτόνομη οδήγηση, η ανασκόπηση εντοπίζει επίσης σημαντικά εμπόδια. Αυτά περιλαμβάνουν το χάσμα προσομοίωσης-πραγματικότητας, την ανάγκη για αποτελεσματική μάθηση ως προς το δείγμα, τον σχεδιασμό αποτελεσματικών συναρτήσεων επιβράβευσης και την ενσωμάτωση θεμάτων ασφάλειας σε συστήματα λήψης αποφάσεων που βασίζονται σε RL. Μελλοντικές κατευθύνσεις της έρευνας, όπως η ρητή αντιμετώπιση των προκλήσεων συντονισμού πολλών πρακτόρων, περιγράφονται επίσης για την προώθηση της ασφάλειας και της επεκτασιμότητας των τεχνολογιών αυτόνομης οδήγησης. Συμπερασματικά, στην εργασία αυτή προτείνονται τυποποιημένες εφαρμογές και ενδελεχή τεκμηρίωση των αλγορίθμων RL για τη διευκόλυνση των εφαρμογών και της αξιοπιστίας στην έρευνα και τις πρακτικές εφαρμογές.

Η μελέτη [28] διερευνά την αποτελεσματικότητα κάποιων DLR αλγορίθμων, συγκεκριμένα των Double Deep Q-Network (DDQN) και Deep Deterministic Policy Gradient (DDPG), στην πλοήγηση σε δυναμικά και ετερογενή περιβάλλοντα. Οι παραδοσιακές μέθοδοι σχεδιασμού διαδρομής συχνά δεν αποδίδουν τα αναμενόμενα σε σύνθετες ρυθμίσεις, ωθώντας αυτήν την έρευνα να διερευνήσει την προσαρμοστικότητα και την ευρωστία των συγκεκριμένων αλγορίθμων. Διεξήχθησαν εκτενείς προσομοιώσεις για να συγκριθούν οι επιδόσεις των DDQN και DDPG σε διάφορα σενάρια, επισημαίνοντας τα ξεχωριστά πλεονεκτήματά τους.

Η πειραματική αξιολόγηση αποκαλύπτει, ότι καθένας από τους DDQN και DDPG έχει δυνατότητες προσαρμοσμένες σε διαφορετικές περιβαλλοντικές συνθήκες. Η σταθερότητα και η προβλεψιμότητα του DDQN το καθιστούν ιδανικό για στατικά περιβάλλοντα, ενώ η προσαρμοστικότητα του DDPG σε συνεχώς μεταβαλλόμενα σενάρια το καθιστά ανώτερο για δυναμικές ρυθμίσεις. Αυτή η διάκριση στον τρόπο λειτουργία τους τους κατευθύνει και σε διαφορετικές εφαρμογές όπως η έρευνα και διάσωση ή ο βιομηχανικός αυτοματισμός. Για παράδειγμα, ο DDQN θα μπορούσε να είναι πιο αποτελεσματικός σε σταθερές βιομηχανικές ρυθμίσεις, ενώ ο DDPG θα ήταν πιο κατάλληλος για την απρόβλεπτη φύση των επιχειρήσεων έρευνας και διάσωσης. Τα αποτελέσματα της μελέτης ευθυγραμμίζονται με την υπάρχουσα βιβλιογραφία, επιβεβαιώνοντας τη σημασία της επιλογής του σωστού αλγορίθμου με βάση το περιβάλλον εφαρμογής. Ωστόσο, οι προσομοιώσεις μπορεί να μην αποτυπώνουν πλήρως την πολυπλοκότητα των σεναρίων του πραγματικού κόσμου, υποδηλώνοντας την ανάγκη για περαιτέρω έρευνα που περιλαμβάνει δοκιμές στον πραγματικό κόσμο και την εξερεύνηση πρόσθετων αλγορίθμων DRL αλλά και υβριδικών προσεγγίσεων. Αυτά τα ευρήματα ενισχύουν την κατανόησή των δυνατοτήτων της ενισχυτικής μάθησης στην αυτόνομη πλοήγηση και όχι μόνο αλλά προσφέρουν και πρακτική καθοδήγηση για εφαρμογές σε πραγματικά συστήματα.

Στην εργασία [29] παρουσιάζεται η ανάπτυξη ενός συστήματος αντίστοιχο με αυτό που θα περιγραφεί και στην παρούσα εργασία, για την εύρεση της βέλτιστης διαδρομής βάσει του DDPG, ενσωματώνοντας την μέθοδο του πεδίου πιθανοτήτων. Υιοθετείται ένα κινηματικό

μοντέλο αυτοκινήτου, για την περιγραφή των αυτόνομων οχημάτων και στο πεδίο πιθανοτήτων εισάγονται τα στοιχεία του χώρου παρατήρησης, τα εμπόδια, τα όρια του δρόμου και τα σημεία της βέλτιστης διαδρομής ώστε να παραχθεί στη συνέχεια η συνάρτηση ανταμοιβής. Ο χώρος παρατήρησης που κατασκευάστηκε έχει στόχο την ομαλή εντός ορίων του δρόμου πορεία του οχήματος ενώ το χαρακτηριστικό αυτής της προσέγγισης εν αντιθέσει με παλαιότερες είναι ότι ο αλγόριθμος μπορεί να προσαρμοστεί σε ποικίλα περιβάλλοντα κάνοντάς το ιδανικό για αυτόνομα συστήματα (Εικόνα 15). Εν συνεχεία εκτελέστηκαν κάποιες προσομοιώσεις για γενικότερη αξιολόγηση.

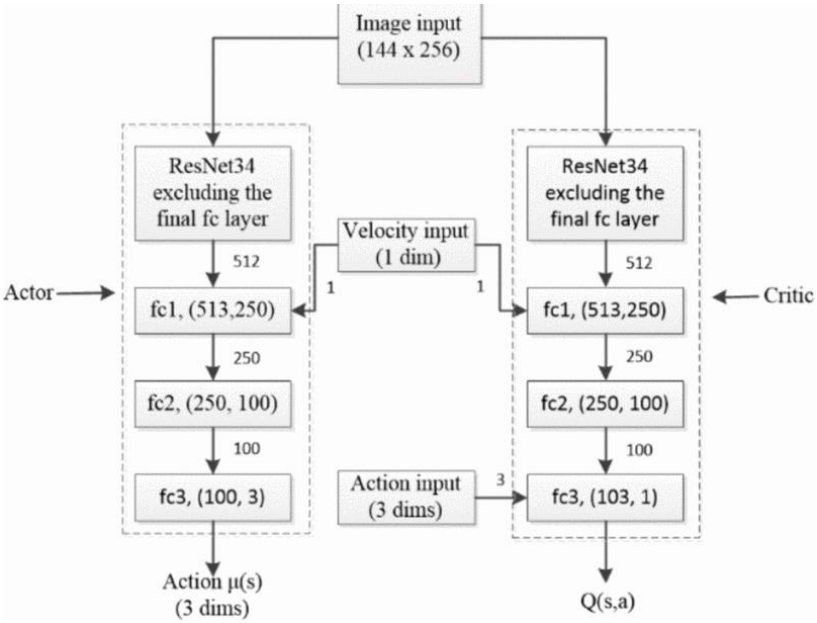


Εικόνα 15 Το πλαίσιο για την εφαρμογή του DDPG [29]

Έχοντας λοιπόν τη συνάρτηση του πεδίου για την πιθανή θέση των εμποδίων, των ορίων του δρόμου και της βέλτιστης διαδρομής που παράχθηκαν με την προτεινόμενη μέθοδο, σχεδιάστηκε η συνάρτηση επιβράβευσης και βάσει αυτών έγινε η εκπαίδευση του αλγορίθμου. Επιπλέον, οι προσομοιώσεις επαληθεύουν ότι ο προτεινόμενος αλγόριθμος μπορεί να αποφύγει αποτελεσματικά τα εμπόδια και να προσαρμοστεί σε διαφορετικά περιβάλλοντα προσαρμόζοντας τα βάρη της συνάρτησης ανταμοιβής, η οποία είναι πιο κατάλληλη για πρακτικές εφαρμογές. Σε σύγκριση με πιο κλασικές μεθόδους ελέγχου, ο DDPG μπορεί να προσαρμόζεται σε διαφορετικές συνθήκες οδήγησης. Ωστόσο, τα σενάρια εκπαίδευσης και δοκιμών στην εργασία αυτή είναι σχετικά ομοιογενή με απλή ρύθμιση εμποδίων, κάτι που απέχει πολύ από τις πραγματικές συνθήκες.

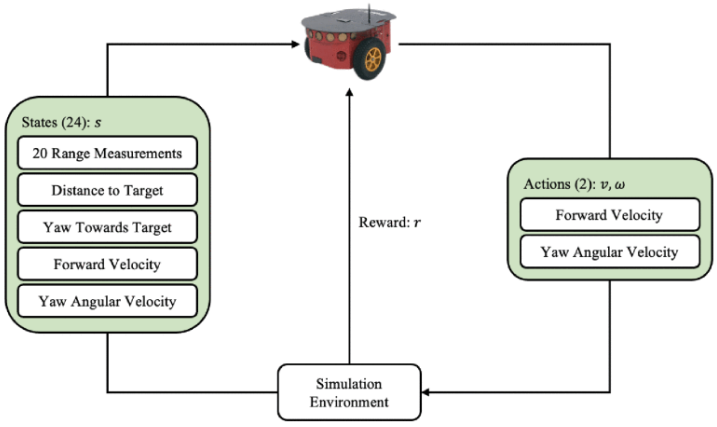
Στην εργασία [30] παρουσιάζεται η εφαρμογή μίας τροποποιημένης έκδοσης του Soft Actor-Critic (SAC) πάνω στην αυτόνομη οδήγηση στο περιβάλλον προσομοίωσης AirSim το οποίο προσφέρει διάφορες δυνατότητες για μεταβολές των συνθηκών ώστε να είναι πιο ρεαλιστικές οι εφαρμογές. Ο αλγόριθμος λαμβάνει την τρέχουσα κατάσταση εικόνας και την ταχύτητα του αυτοκινήτου ως είσοδο και ως έξοδος είναι οι τιμές της επιτάχυνσης, των φρένων και της γωνίας διεύθυνσης, για να διασφαλιστεί η οδήγηση του αυτόνομου οχήματος με τρόπο

παρόμοιο με έναν οδηγό. Πρέπει να σημειωθεί ότι και στην παρούσα εργασία και όπως θα αναλυθεί αναλυτικά σε επόμενο κεφάλαιο, οι μεταβλητές του χώρου δράσης του οχήματος είναι αντίστοιχες με αυτές, δηλαδή η επιτάχυνση (θετική και αρνητική) και η διεύθυνση του οχήματος. Το μοντέλο εκπαιδεύεται αρχικά χρησιμοποιώντας την μέθοδο της μίμησης, παρέχοντας έτσι προεκπαιδευμένα την πολιτική και τα βάρη στον SAC. Για να ενισχυθεί η ευρωστία των εφικτών πολιτικών κατά τη διάρκεια της ενισχυτικής μάθησης, η αρχιτεκτονική ResNet-34 χρησιμοποιείται εντός του πλαισίου του αλγορίθμου (Εικόνα 16).



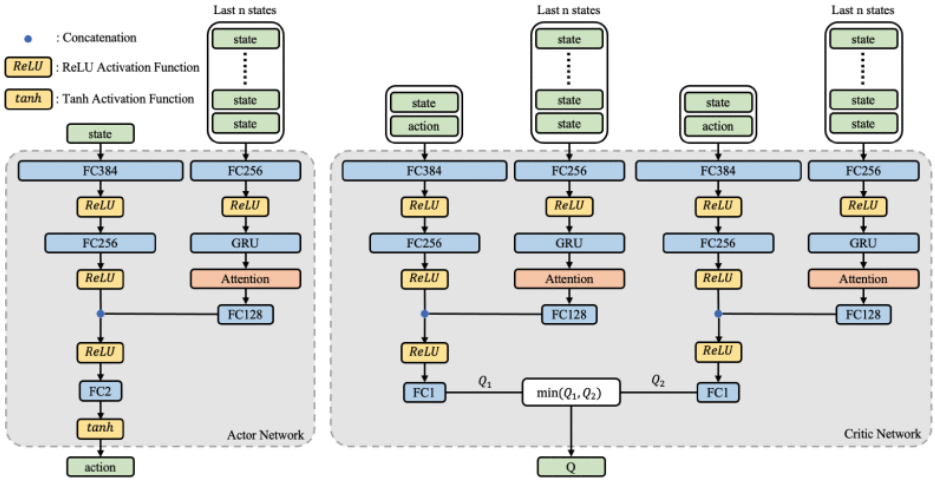
Εικόνα 16 Η δομή του αλγορίθμου [30]

Στη δημοσίευση [31], προτείνεται μία εφαρμογή DRL για πλοήγηση με δεδομένο στόχο που βασίζεται στον TD3, λαμβάνοντας από τον χώρο παρατήρησης τις μετρήσεις του LiDAR, την απόσταση οχήματος και σημείου-στόχο και την απόκλιση από την επιθυμητή κατεύθυνση. Και σε αυτή την εφαρμογή οι μεταβλητές του χώρου δράσης που καλείται να επιλέγει κάθε φορά ο αλγόριθμος είναι και εδώ η επιτάχυνση και η γωνία της πορείας.



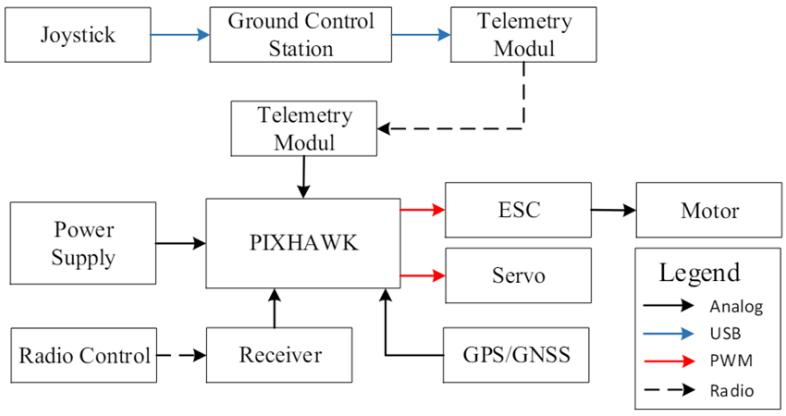
Εικόνα 17 Το μοντέλο RL [31]

Αξιοσημείωτο σε αυτή την εφαρμογή είναι ότι το όχημα μπορεί να αποφεύγει εμπόδια δίχως προηγούμενη γνώση του περιβάλλοντος, παρά μόνο από την εφαρμογή του εκπαιδευμένου δικτύου, το οποίο και εκπαιδεύτηκε σε προσομοιωμένο περιβάλλον. Όπως αναφέρεται, οι δοκιμές του προτεινόμενου συστήματος ήταν άκρως ικανοποιητικές και εν δυνάμει μπορεί να εφαρμοστεί και σε κανονικά οχήματα.



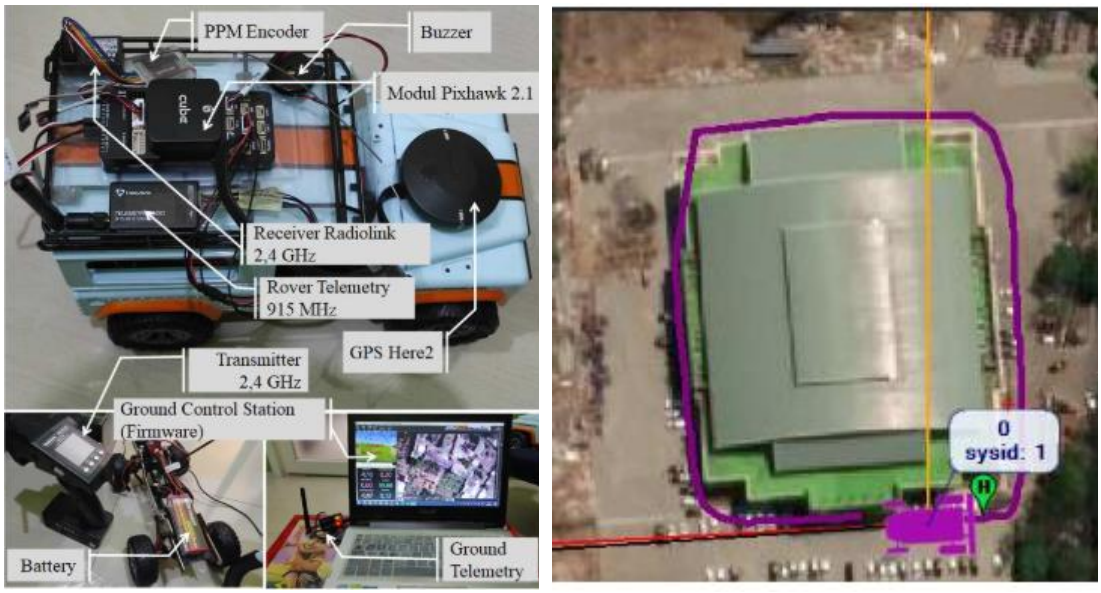
Εικόνα 18 Η προτεινόμενη εφαρμογή του τροποποιημένου TD3 [31]

Η δημοσίευση [32], παρουσιάζει ενδιαφέρον καθώς εγκαθιστά σε ένα αυτόνομο όχημα τους ίδιους αισθητήρες με αυτούς που εφαρμόζονται στην παρούσα εργασία, δηλαδή συστήματα GNSS και IMU. Αναφέρεται στην ανάπτυξη ενός αυτόνομου συστήματος πλοήγησης οχημάτων που αξιοποιεί τεχνικές συγκερασμού δεδομένων από αισθητήρες για τον μετριασμό των σφαλμάτων στην ακρίβεια του GPS. Η στρατηγική ενσωμάτωσης περιλαμβάνει το συνδυασμό αισθητήρα IMU που είναι ενσωματωμένος στον ελεγκτή πτήσης pixhawk 2.1 με τη μονάδα GPS/GNSS Here2, χρησιμοποιώντας μια προσέγγιση που βασίζεται σε φίλτρο Kalman. Αυτή η μέθοδος στοχεύει να ενισχύσει την ικανότητα του συστήματος να προσδιορίζει με ακρίβεια τη θέση και τον προσανατολισμό του οχήματος, κάτι που είναι ζωτικής σημασίας για αξιόπιστη αυτόνομη πλοήγηση.



Εικόνα 19 Διάγραμμα συστήματος πλοήγησης [32]

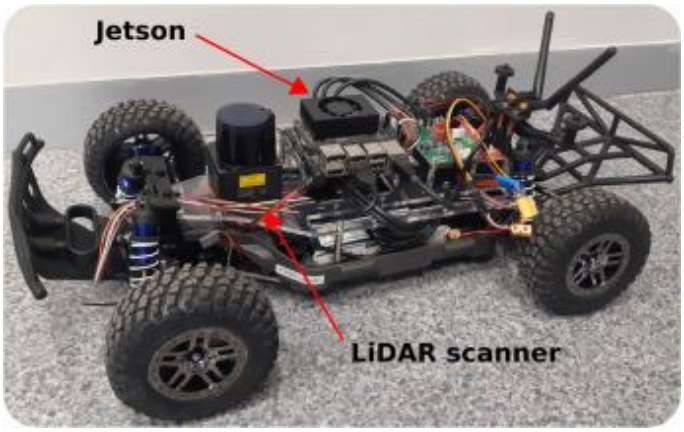
Τα αρχικά αποτελέσματα των δοκιμών έδειξαν υποσχόμενα αποτελέσματα, δείχνοντας την ικανότητα του συστήματος να επιτυγχάνει τις αναμενόμενες γωνίες προσανατολισμού και να σχεδιάζει με ακρίβεια διαδρομές πλοήγησης. Ωστόσο στην επικοινωνία του συστήματος η τεχνολογία του radio control παρουσίασε προβλήματα καθώς διάφορες παρεμβολές δημιούργησαν κωλύματα στην ολοκλήρωση των διαδρομών. Βέβαια το σημείο που χρήζει περισσότερης εμβάθυνσης είναι ο τρόπος με τον οποίον συνδυάζονται τα δεδομένα των αισθητήρων.



Εικόνα 20 Αριστερά: Η πειραματική διάταξη, Δεξιά: Η πορεία με την χρήση GNSS [32]

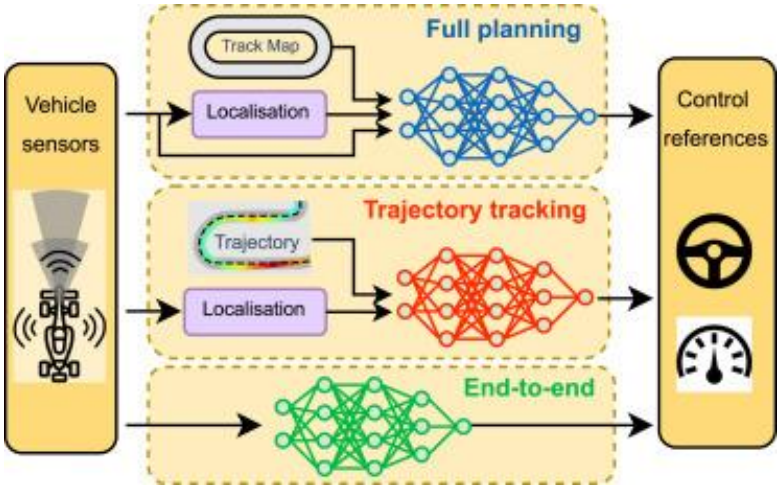
Μεγάλο ενδιαφέρον παρουσιάζει η εφαρμογή [33], όπου ο DDPG εφαρμόζεται για την αυτόνομη πλοήγηση σε ένα ηλεκτρικό όχημα εστιάζοντας ταυτόχρονα και στην κατανάλωση ενέργειας. Εκπαιδεύοντας το μοντέλο μέσω Matlab & Simulink, κατάφεραν να δημιουργήσουν μία αρχιτεκτονική που σε ένα σενάριο οδήγησης σε αυτοκινητόδρομο πέραν της απόδοσης στην πλοήγηση επετεύχθη και η εξοικονόμηση ενέργειας στην μπαταρία του οχήματος. Η σημασία τέτοιων εφαρμογών σε έναν κόσμο που εστιάζει στην βιωσιμότητα είναι μεγάλη, αφού η αυτοματοποίηση των εργασιών πρέπει να γίνεται με γνώμονα την μετάβαση σε τεχνολογίες φιλικές προς το περιβάλλον.

Βέβαια πέραν των εφαρμογών αυτόνομης οδήγησης για καθημερινή χρήση των οχημάτων δε λείπουν και τα παραδείγματα εφαρμογών της ενισχυτικής μάθησης σε αγωνιστικά οχήματα. Στην δημοσίευση [34] παρουσιάζεται η εφαρμογή των DDPG, TD3 και SAC σε προσομοίωση ενός πραγματικού οχήματος υπό κλίμακα.



Εικόνα 21 Το υπό κλίμακα όχημα [34]

Η εφαρμογή αποτελείται από την εκπαίδευση των μοντέλων των αλγορίθμων σε τρεις διαφορετικές προσεγγίσεις για την αυτόνομη οδήγηση σε διάφορες πίστες αγώνων. Οι προσεγγίσεις είναι αρχικά ο πλήρης σχεδιασμός της πίστας, όπου έχοντας γνώση της διαδρομής αλλά και της θέσης του οχήματος εκτελείται η οδήγηση. Στη συνέχεια εφαρμόζεται η διαδικασία που ακολουθείτε η πορεία όπου δεδομένων σημείων στόχων και της θέσεως του οχήματος αυτό καλείται να ακολουθεί τα σημεία στόχους. Τέλος η end-to-end αρχιτεκτονική, δέχεται ακατέργαστες πληροφορίες από τον δρόμο τον οποίο καλείται να ακολουθήσει το όχημα και επί τόπου δίνει εντολές άνευ προηγούμενου σχεδιασμού.



Εικόνα 22 Προσεγγίσεις αυτόνομης οδήγησης [34]

Όπως αποδεικνύεται από τα αποτελέσματα την καλύτερη απόδοση είχε η πλήρης χρήση δικτύου TN, σε κάθε περίπτωση δοκιμάστηκαν και οι τρεις αλγόριθμοι. Παρόλο που η εφαρμογή ήταν επιτυχής σημειώνεται ότι η μεταφορά από την προσομοίωση στο πραγματικό όχημα παρουσίασε σημαντική απόκλιση, γεγονός που γεννά φυσικά νέες προκλήσεις για την έρευνα.

2. Το Πλαίσιο της Προσομοίωσης

Στο παρόν κεφάλαιο θα περιγραφούν τα δομικά στοιχεία της προσομοίωσης (προσομοιωτής, μοντελοποιημένο όχημα), το περιβάλλον GYM και η βιβλιοθήκη Stable Baselines3 που διευκολύνει την εφαρμογή αλγορίθμων ενισχυτικής μάθησης και η δομή του μοντέλου ενισχυτικής μάθησης που εφαρμόστηκε.

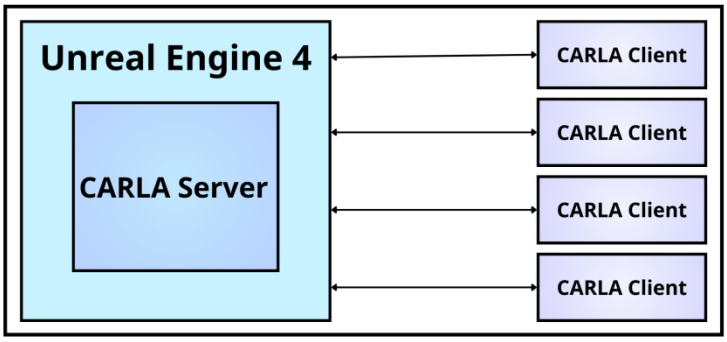
2.1 Περιβάλλον προσομοίωσης CARLA

Ο προσομοιωτής CARLA [35], [36] είναι μια πλατφόρμα προσομοίωσης ανοιχτού κώδικα που αναπτύχθηκε ώστε να λειτουργεί μέσω της Unreal Engine 4 (UE4) με σκοπό να υποστηρίξει την ανάπτυξη και τον έλεγχο συστημάτων αυτόνομης οδήγησης. Όντας λογισμικό ανοιχτού κώδικα υποστηρίζεται από την ομάδα προγραμματιστών που το δημιούργησε αλλά και από ανεξάρτητους χρήστες παρέχοντας σε κάθε μέλος πρόσβαση σε βιβλιοθήκες έτοιμες προς χρήση με αυτοκίνητα, αισθητήρες και χάρτες βασισμένους στο πρωτόκολλο ASAM OpenDRIVE [37], παρέχοντας ταυτόχρονα έτοιμους κώδικες για τον έλεγχο όλων αυτών. Πέραν αυτών των πλεονεκτημάτων η UE4 δίνει την δυνατότητα για τροποποιήσεις κάθε παραμέτρου βάσει των αναγκών του χρήστη. Είναι δυνατή η εισαγωγή του χάρτη της επιλογής του χρήστη, ανεξάρτητου μοντέλου αυτοκινήτου και αισθητήρων καθώς και φυσικών παραμέτρων. Το αποτέλεσμα είναι μία άκρως ρεαλιστική προσομοίωση που διέπεται από φυσικούς νόμους δίνοντας τη δυνατότητα για πειράματα με δεδομένα ικανά να ανταποκριθούν στο φυσικό περιβάλλον. Όλοι αυτοί οι λόγοι το καθιστούν εργαλείο αιχμής και απόδειξη για αυτό είναι ευρεία χρήση του στους κλάδους της έρευνας αλλά και της βιομηχανίας. Στη συνέχεια θα περιγραφεί ο τρόπος λειτουργίας του CARLA αλλά και τα επιμέρους χαρακτηριστικά του.

Εικόνα 23 Το περιβάλλον του CARLA στην UE4

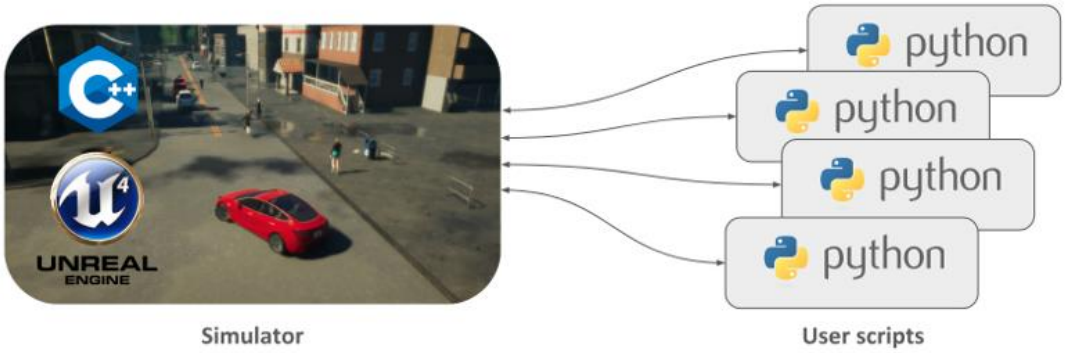
2.1.1 Τρόπος λειτουργίας του περιβάλλοντος CARLA

Η λειτουργία του προσομοιωτή βασίζεται σε αρχιτεκτονική διακομιστή – διακομιζομένου (server – client). Ο διακομιστής είναι υπεύθυνος για όλες τις υπολογιστικές εργασίες που αφορούν στην προσομοίωση των φυσικών παραμέτρων και στην ενημέρωση του περιβάλλοντος κάθε στιγμή. Μέσω αυτών των διεργασιών καθίστανται δυνατές οι αλληλεπιδράσεις μεταξύ των οχημάτων, των πεζών και κάθε άλλου στοιχείου του περιβάλλοντος. Ο πυρήνας όλης αυτής της πολύπλοκης διαδικασίας είναι η UE4 που ως μηχανή προσομοίωσης μπορεί να αναπαριστά ρεαλιστικά κάθε τι που υπολογίζεται από τον διακομιστή. Αρχιτεκτονική αυτή φαίνεται στην Εικόνα 24.



Εικόνα 24 Η αρχιτεκτονική server-client

Η λειτουργία του διακομιζομένου (client) είναι ο έλεγχος κάθε οντότητας εντός της προσομοίωσης. Μέσω αυτού ελέγχονται τα οχήματα, οι αισθητήρες, εξάγονται τα δεδομένα αλλά και δίνονται οι εντολές για κάθε μέσο που ελέγχεται. Ο τρόπος επικοινωνίας server-client είναι το API (Application Programming Interface) του CARLA απ' όπου δίνεται κάθε εντολή και υπάρχει αμφίδρομη μεταφορά δεδομένων. Το API διατίθεται είτε σε Python είτε σε C++, δίνοντας ευελιξία στον χρήστη για την εφαρμογή που χρειάζεται. Ωστόσο δεν υπάρχει περιβάλλον διεπαφής για τον χρήστη και η κάθε εντολή πρέπει να δίνεται μέσω τερματικού στον υπολογιστή, όπως φαίνεται στην Εικόνα 25.



Εικόνα 25 Σχεδιάγραμμα διεπαφής χρήστη και προσομοιωτή [36]

Σε κάθε server μπορούν να συνδεθούν πάνω από ένας client, καθώς μπορεί το πρόγραμμα ελέγχου του αυτοκινήτου να είναι διαφορετικό από αυτό για τον έλεγχο των πεζών ή του καιρού επί παραδείγματι. Ακόμα ένα σημείο που πρέπει να αναφερθεί για τον τρόπο λειτουργίας είναι το σύγχρονο και το ασύγχρονο καθεστώς. Στη σύγχρονη λειτουργία ο server ανανεώνει το περιβάλλον και κάθε παράμετρο εντός του μόνο κατ' εντολή του client ο οποίος και ελέγχει τον ρυθμό ενημέρωσης. Στην ασύγχρονη λειτουργία, η οποία και χρησιμοποιείται σε αυτή την εργασία, ο server εκτελεί κάθε διεργασία της προσομοίωσης αυτόνομα και στον υψηλότερο δυνατό ρυθμό, ενώ ταυτόχρονα επεξεργάζεται και τα αιτήματα του client όσο το δυνατόν ταχύτερα. Η ασύγχρονη λειτουργία προσφέρει προσομοιώσεις πιο κοντά στον πραγματικό κόσμο και με υψηλότερη απόδοση.

Όσον αφορά στην συνδεσιμότητα της CARLA, προσφέρονται διάφορα πρωτόκολλα επικοινωνίας όπως TCP/IP, UDP και WebSocket προσφέροντας ευελιξία στον χρήστη για την επιλογή καναλιού επικοινωνίας εξασφαλίζοντας μάλιστα και την μεταφορά δεδομένων σε πραγματικό χρόνο. Επιπλέον ενσωματώνεται το Robot Operating System (ROS), δίνοντας τη δυνατότητα για ομαλή σύνδεση και μεταξύ άλλων εργαλείων και βιβλιοθηκών ROS. Το τελευταίο αυτό χαρακτηριστικό διευκολύνει ιδιαίτερα τη χρήση του CARLA για την έρευνα και ανάπτυξη πάνω στον τομέα των ρομπότ και των αυτόνομων συστημάτων.

2.1.2 Actors, Blueprints & Χάρτες

Στον προσομοιωτή κάθε δυναμικό στοιχείο όπως τα οχήματα, οι πεζοί και οι αισθητήρες αναφέρεται ως "actor" και όπως μπορεί κάποιος να υποθέσει είναι θεμελιώδη στοιχεία κάθε προσομοίωσης αφού δημιουργούν το περιβάλλον αλληλεπίδρασης. Ως "Blueprints" αναφέρονται οι έτοιμοι "actors" που μπορούν να εισαχθούν από τις βιβλιοθήκες του CARLA.

Actors

Πιο συγκεκριμένα μπορούν να οριστούν ως "actors" όλες οι δομές που αλληλεπιδρούν στην προσομοίωση και συνεισφέρουν στο να γίνει πιο ρεαλιστική. Μπορούν να είναι οχήματα, πεζοί, σηματοδότες και πινακίδες και επιπλέον τα αισθητήρια όργανα. Οι actors ελέγχονται σε κάθε περίπτωση από το API του προσομοιωτή και η λειτουργία τους γίνεται κατά παραγγελία κάθε client. Η συμπεριφορά τους μπορεί να είναι προκαθορισμένη ή να αλλάζει βάσει των αναγκών του χρήστη όταν αυτός δίνει τις κατάλληλες εντολές ακόμα και όταν η προσομοίωση είναι υπό εξέλιξη. Αυτά τα χαρακτηριστικά δίνουν τη δυνατότητα για εφαρμογές με ακριβή δεδομένα που πλησιάζουν σε πραγματικά σενάρια.

Blueprints

Τα blueprints διευκολύνουν τον χρήστη αφού μπορεί να εισάγει έτοιμους actors χωρίς να χρειάζεται να τους δημιουργήσει από το μηδέν. Από τις βιβλιοθήκες του CARLA μπορούν να

χρησιμοποιηθούν απευθείας και δίνεται μάλιστα η δυνατότητα να τροποποιηθούν τα χαρακτηριστικά κάθε αντικειμένου, όπως το χρώμα, οι διαστάσεις ή η συμπεριφορά του. Εκμεταλλευόμενος λοιπόν αυτή τη δυνατότητα ο χρήστης μπορεί να ρυθμίσει το περιβάλλον στα μέτρα του και να χρησιμοποιήσει οχήματα και αισθητήρες όπως χρειάζεται για την κάθε εφαρμογή. Ο Πίνακας 4 που ακολουθεί, εμπεριέχει μία ενδεικτική λίστα από τον κατάλογο των bluprints.

Πίνακας 4 Ενδεικτικά παραδείγματα Blueprints

Blueprints οχημάτων	***Blueprints πεζών***	***Blueprints οδικής σήμανσης***
• Σεντάν • Trucks • Μέσα μαζικής μεταφοράς	• Άνθρωποι (παιδιά, ενήλικες, ηλικιωμένοι) • Αθλούμενοι (πχ ποδηλάτες, δρομείς) • Ζώα	• Ταμπέλες κυκλοφορίας • Φανάρια • Διαβάσεις
Blueprints Περιβάλλοντος	***Blueprints Δυναμικών στοιχείων***	***Ειδικά Blueprints***
• Δέντρα • Φράχτες • Κιγκλιδώματα	• Δυναμικά εμπόδια (πεζοί, οχήματα) • Προσωρινά οδοφράγματα	• Αισθητήρες (κάμερες, LiDARs) • Σήμανση τροχιάς • Καιρικά φαινόμενα

Χάρτες

Οι χάρτες του προσομοιωτή, είναι τα στοιχεία που καθορίζουν το εικονικό περιβάλλον και θέτουν τα όρια και τις διαδρομές εντός της προσομοίωσης. Περιλαμβάνουν αστικά και αγροτικά τοπία με κάθε οδικό στοιχείο όπως δρόμους και διασταυρώσεις αλλά και δομές όπως κτήρια ή φυσικά τοπία. Είναι σχεδιασμένοι ώστε να μιμούνται πραγματικά τοπία με ρεαλιστικές αναπαραστάσεις. Ο χρήστης μπορεί είτε να χρησιμοποιήσει χάρτες από την βιβλιοθήκη του CARLA αλλά μπορεί και να εισαγάγει δικούς του μέσω διάφορων πρωτοκόλλων, ενδεικτικά παραδείγματα από τους διαθέσιμους χάρτες φαίνονται στην Εικόνα 26. Επίσης υπάρχει η δυνατότητα να χρησιμοποιηθεί μόνο η αναπαράσταση χάριν εξοικονόμησης υπολογιστικής ισχύος. Ακόμα δίνεται η δυνατότητα στον χρήστη να εισαγάγει δικό του χάρτη βάσει του προτύπου OpenDRIVE, χρησιμοποιώντας την εφαρμογή RoadRunner της MathWorks.

Εικόνα 26 Αριστερά: πάνω φαίνεται ο χάρτης Town10 με ένα στιγμιότυπο της πόλης με νυχτερινή ρύθμιση από κάτω, Δεξιά: πάνω φαίνεται ο χάρτης Town7 με αντίστοιχο στιγμιότυπο της πόλης κάτω

2.2 Μοντέλο Οχήματος EcoCar

Το όχημα το οποίο επιλέχθηκε ως πλατφόρμα για τους πειραματισμούς είναι το EcoCar City (Εικόνα 27), ένα αμιγώς ηλεκτροκίνητο αυτοκίνητο πόλης, το οποίο φαίνεται στην παρακάτω εικόνα. Το αυτοκίνητο αυτό επιλέχθηκε καθότι ο απλός σχεδιασμός του επιτρέπει πειραματισμούς και στην εφαρμογή των αποτελεσμάτων της προσομοίωσης στο φυσικό όχημα που βρίσκεται στο Εργαστήριο Ρομποτική & Ευφυών Συστημάτων του Πολυτεχνείου Κρήτης. Στον Πίνακας 5 συγκεντρώνονται τα τεχνικά του χαρακτηριστικά.

Εικόνα 27 EcoCar Cty

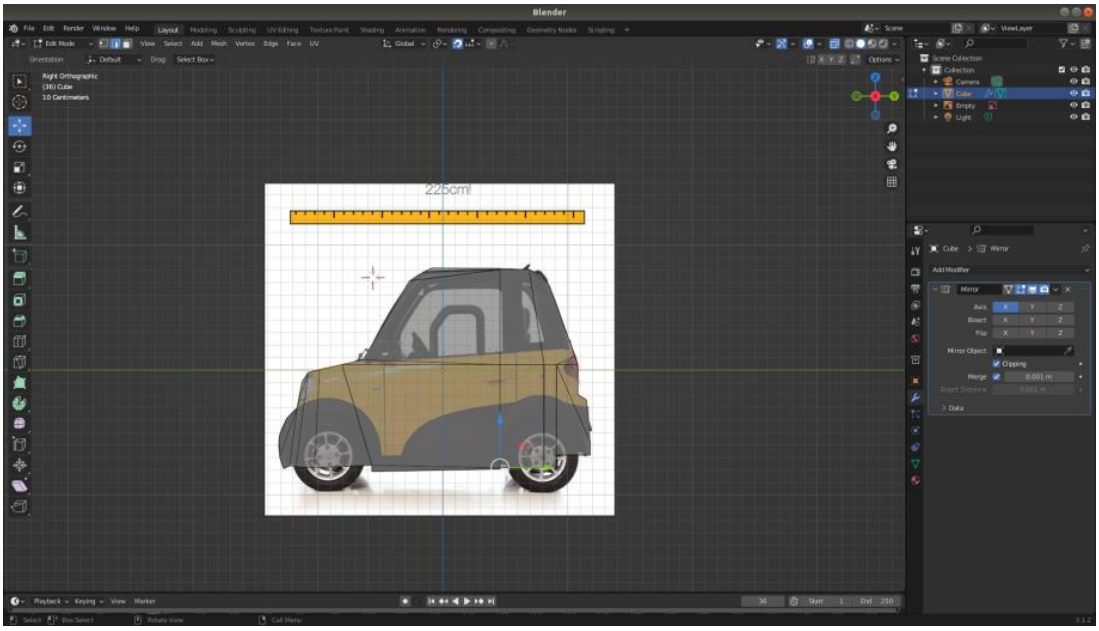
Πίνακας 5 Κύρια Τεχνικά Χαρακτηριστικά EcoCar City

Αμάξωμα:	Εξωτερικές Διαστάσεις: 2245x1290x1570mm Βάρος με Μπαταρία: 600kg
Κινητήρας:	Τύπος: AC ηλεκτρικός κινητήρας Ονομαστική Ισχύς: 7,5kW ~ 10hp Μέγιστη Ταχύτητα: 80km/h
Συσσωρευτής:	Τύπος: Li ion Χωρητικότητα: 7,5kWh

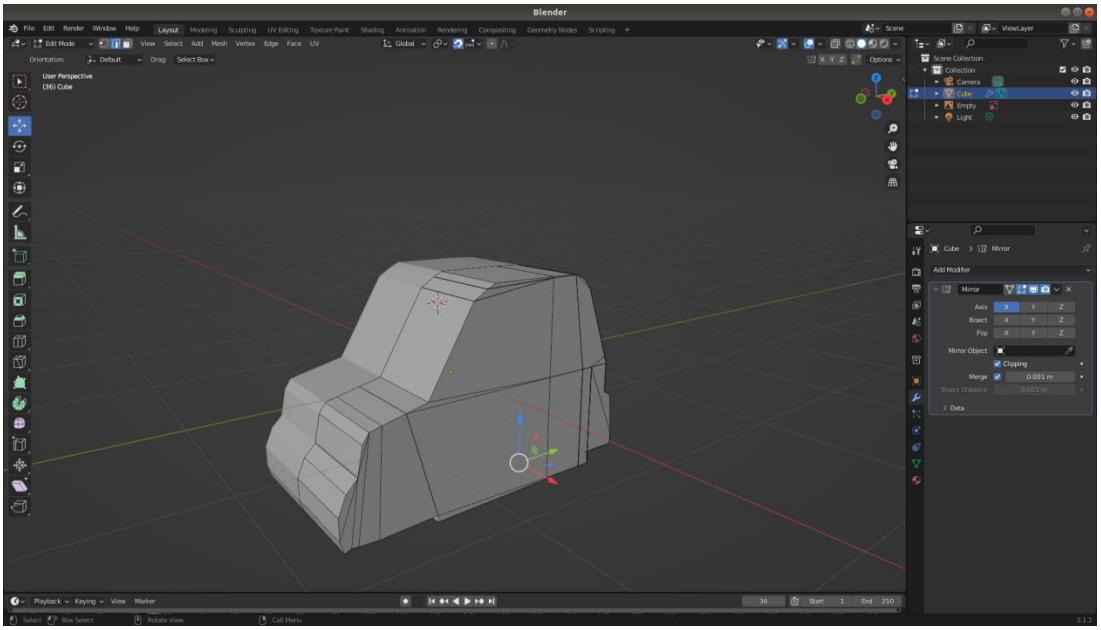
2.2.1 Τρισδιάστατη Μοντελοποίηση Οχήματος

Για τις ανάγκες της προσομοίωσης το όχημα έπρεπε μοντελοποιηθεί σε κατάλληλο πρόγραμμα CAD και να εισαχθεί στο CARLA καθότι δεν υπήρχε έτοιμο αντίστοιχο μοντέλο. Για το σκοπό αυτό χρησιμοποιήθηκε το Blender, λογισμικό τρισδιάστατης μοντελοποίησης ανοιχτού κώδικα που χρησιμοποιείται ευρέως στη βιομηχανία για διάφορες εφαρμογές όπως μοντελοποίηση αντικειμένων, προσομοιώσεις ρευστών, κινούμενα σχέδια και άλλα. Ο πρωταρχικός στόχος του μοντέλου του οχήματος είναι να ταιριάζει με τις διαστάσεις και τη φυσική του φυσικού οχήματος, χωρίς να δίνεται μεγάλη έμφαση στις σχεδιαστικές λεπτομέρειες χάριν οικονομίας χρόνου.

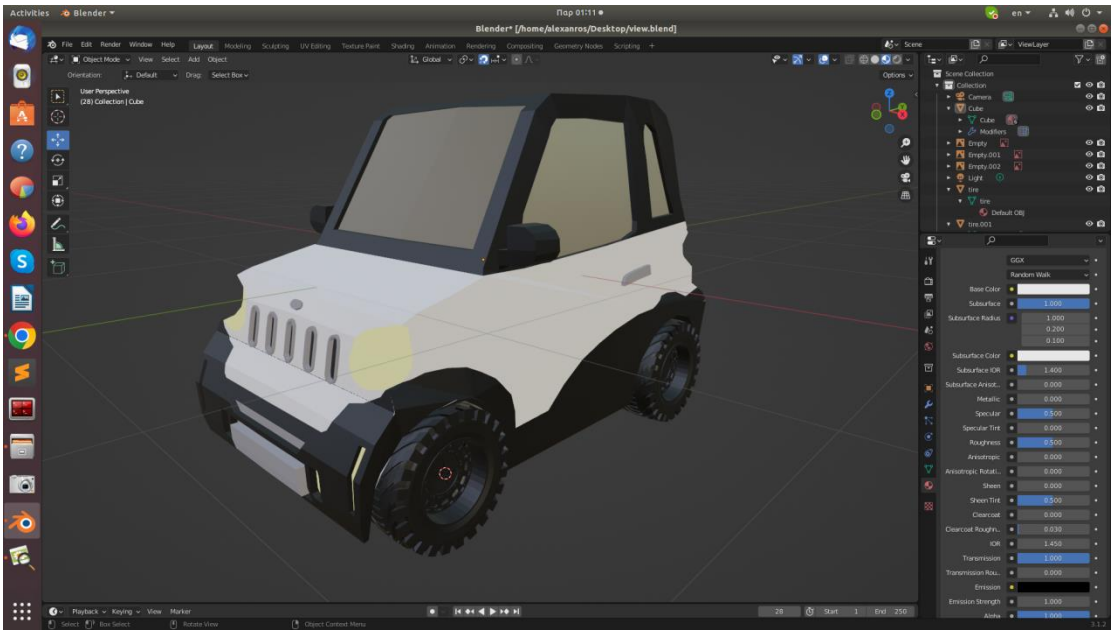
Για την σχεδίαση του αυτοκινήτου στο πρόγραμμα, εισήχθησαν φωτογραφίες από τις πλευρές του οχήματος, όπως στην Εικόνα 28, για την σχεδίαση του περιγράμματος του. Αυτό βοήθησε στην δημιουργία ενός συμπαγούς πρωτοτύπου που στη συνέχεια του προστέθηκαν ορισμένες λεπτομέρειες όπως οι θόλοι των τροχών, τα σχέδια των παραθύρων και χρώματα. Στη συνέχεια οι φωτογραφίες που παρουσιάζονται είναι από αυτή την διαδικασία.



Εικόνα 28 Προσαρμογή διαστάσεων του οχήματος



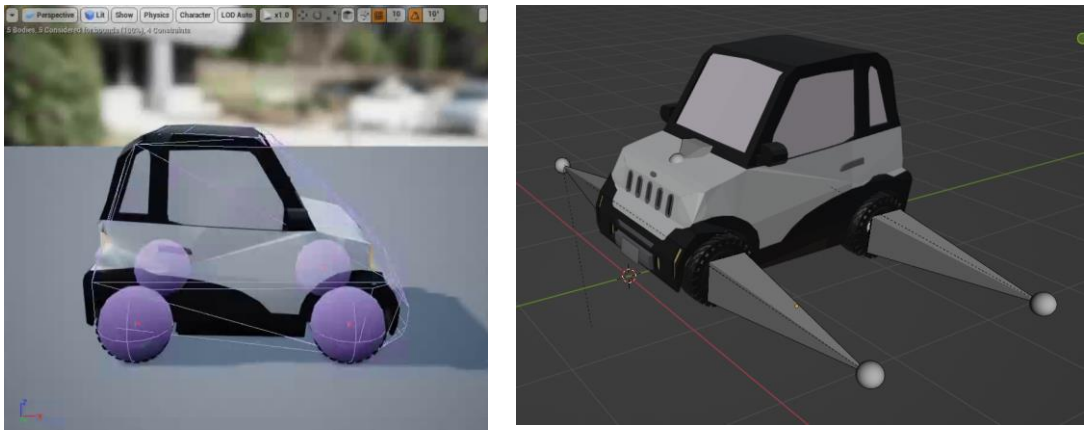
Εικόνα 29 Συμπαγές CAD μοντέλο



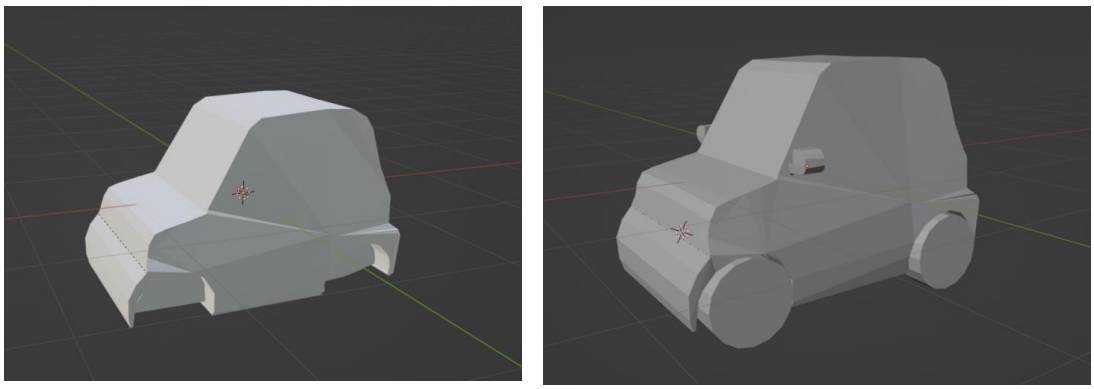
Εικόνα 30 Μοντέλο CAD με λεπτομέρειες

2.2.2 Εισαγωγή Μηχανικών Ιδιοτήτων & Καμπύλη Ροπής

Για να εισαχθεί στον προσομοιωτή το όχημα δεν αρκεί να υπάρχει μόνο σχεδιαστικά το τρισδιάστατο μοντέλο αλλά πρέπει να του δοθούν και φυσικές ιδιότητες και επιπλέον η μηχανική σχέση των τροχών με το όχημα. Η σύνδεση των τροχών απεικονίζεται στην Εικόνα 31. Για το υπόλοιπο αυτοκίνητο απαιτούνται οι δομές των Physical Mesh και Raycast Sensor Mesh, οι οποίες εμφανίζονται στην Εικόνα 32. Φαίνονται πανομοιότυπες καθότι η διαφορά τους είναι ο τύπος αρχείου που εισάγεται στον προσομοιωτή. Το Physical Mesh βοηθά στον υπολογισμό της φυσικής του οχήματος από την UE4, ενώ το Raycast Sensor Mesh καθορίζει το σχήμα που ανιχνεύεται από τους διάφορους αισθητήρες του προγράμματος.



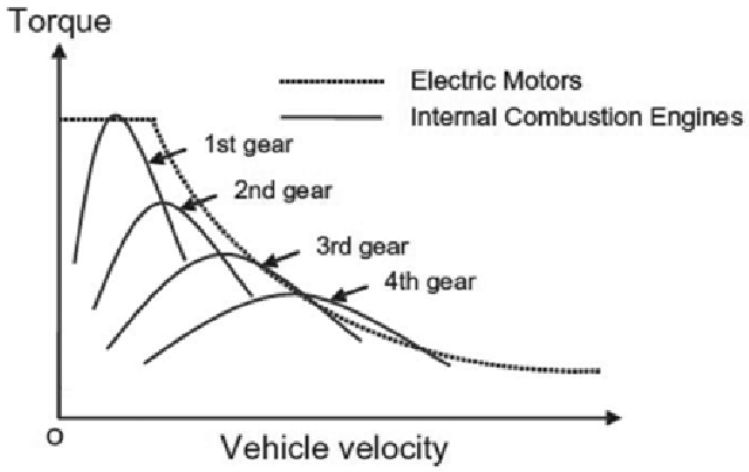
Εικόνα 31 Στάδια εισαγωγής φυσικών ιδιοτήτων στους τροχούς



Εικόνα 32 Αριστερά: Raycast Sensor Mesh, Δεξιά: Physical Mesh

Πέραν των ανωτέρω ιδιοτήτων, για την καλύτερη φυσική ανταπόκριση του οχήματος κατά τις προσομοιώσεις, επιλέχθηκε να συμπεριληφθεί και η καμπύλη ροπής του οχήματος. Στοιχείο που οδηγεί το όχημα σε πιο ρεαλιστική συμπεριφορά κατά την επιτάχυνση και επιβράδυνση του. Η καμπύλη ροπής απεικονίζει τη σχέση μεταξύ ροπής και στροφών ενός κινητήρα. Σε κινητήρες εσωτερικής καύσης, η ροπή αυξάνεται σταδιακά, φτάνοντας στο μέγιστο σε μεσαίο εύρος στροφών. Αντίθετα, στους ηλεκτροκινητήρες, η μέγιστη ροπή είναι διαθέσιμη άμεσα από χαμηλές στροφές και φθίνει έπειτα από τις 2.500, προσφέροντας γρήγορη

επιτάχυνση, όπως φαίνεται και στο διάγραμμα της Εικόνα 33. Τα χαρακτηριστικά του κινητήρα του EcoCar δεν ήταν διαθέσιμα, έτσι βάσει της συνάρτησης $Torque(Nm) = \frac{9.5488 * kW}{rpm}$ (2.1). Τελικώς η καμπύλη που εισάχθηκε φαίνεται στην Εικόνα 34.



Εικόνα 33 Διάγραμμα ροπής κινητήρα εσωτερικής καύσης & ηλεκτροκινητήρα [38]

Εικόνα 34 Καμπύλη ροπής EcoCar

Η διαδικασία ολοκληρώνεται όταν όλα τα απαραίτητα αρχεία εισαχθούν στην Unreal Engine 4 και μπορούν από εκεί να χρησιμοποιηθούν από το CARLA. Το EcoCar είναι πλέον blueprint του προσομοιωτή και μπορεί να χρησιμοποιηθεί σε οποιαδήποτε εφαρμογή θελήσει ο χρήστης.

2.4 GYM Application Programming Interface (API)

Στην ενότητα που ακολουθεί θα περιγραφεί το εργαλείο GYM, το οποίο χρησιμοποιείται για την ενορχήστρωση της εκπαίδευσης του μοντέλου ενισχυτικής μάθησης. Θα αναλυθεί ο τρόπος λειτουργίας του, τα πλεονεκτήματα της χρήσης του καθώς και το περιβάλλον το οποίο δημιουργήθηκε για την παρούσα εργασία.

Όσον αφορά στην ενισχυτική μάθηση το GYM αποδεικνύεται ένα πολύ χρήσιμο εργαλείο καθώς προσφέρει ένα σύνολο με έτοιμα περιβάλλοντα για την εκπαίδευση αλγορίθμων που

καλύπτουν ένα φάσμα από απλά προβλήματα ελέγχου όπως το Atari έως και πολύπλοκες προσομοιώσεις [39].

Αναπτύχθηκε από την OpenAI ως GYM αλλά συνεχίστηκε από μία ομάδα προγραμματιστών ως GYMNASIUM [40], καθώς η πρώτη έπαψε την συντήρηση και ανανέωση του. Είναι μια ανοιχτού κώδικα βιβλιοθήκη της Python, η οποία περιλαμβάνει έτοιμες συναρτήσεις που με την κλήση τους μπορεί το μοντέλο RL να συλλέξει απαραίτητα δεδομένα από τους χώρους παρατήρησης και να τα αξιολογήσει με στόχο την εκτέλεση νέων αποφάσεων σε κάθε βήμα της εκπαίδευσης. Ωστόσο δεν περιορίζεται μόνο στην εκπαίδευση του αλγορίθμου αλλά και αφότου το μοντέλο είναι έτοιμο μπορεί να διαχειρίζεται τα δεδομένα που συλλέγονται και βάσει της εμπειρίας που έχει αποθηκευτεί να επιλέγεται η βέλτιστη δράση. Τα δομικά του μέρη είναι ο χώρος παρατήρησης, ο χώρος δράσης καθώς και ο μηχανισμός ανταμοιβής που δεν είναι τίποτε άλλο από την συνάρτηση επιβράβευσης.

Οι δυνατότητες του GYM το καθιστούν ένα ισχυρό εργαλείο στην έρευνα καθώς απλοποιεί μία πολύ επίπονη κατά τ' άλλα διαδικασία. Έτσι είτε πρόκειται για ακαδημαϊκή ή βιομηχανική χρήση βοηθά στην ταχύτερη ανάπτυξη της ενισχυτικής μάθησης. Ακόμα το γεγονός ότι ο χρήστης μπορεί να δημιουργήσει δικά του περιβάλλοντα δίνει την δυνατότητα για ακόμη μεγαλύτερη εξέλιξη και έρευνα γύρω από νέες τεχνικές σε περιβάλλοντα που απαιτούν έλεγχο όπως η ρομποτική και η αυτόνομη οδήγηση. Με άξονα αυτή την ευέλικτη φύση του GYM και λόγω του γεγονότος ότι δεν υπήρχε έτοιμο περιβάλλον για τα πειράματα του EcoCar στο CARLA δημιουργήθηκε το περιβάλλον CarlaEnv [8]. Σε αυτό συμπεριλαμβάνονται όλες οι λειτουργίες για την επικοινωνία με τον server του προσομοιωτή, την τοποθέτηση του οχήματος και των άλλων στοιχείων της προσομοίωσης, την επικοινωνία με τους αισθητήρες και το όχημα αλλά και την εκτέλεση της εκπαίδευσης του μοντέλου RL. Η όλη διαδικασία μπορεί να χωριστεί σε τρία βήματα.

Αρχικά ενεργοποιείται η κλάση που ορίζει το περιβάλλον και τα στοιχεία του, συμπεριλαμβανομένων όλων των συναρτήσεων για τον έλεγχο του οχήματος και την εκπαίδευση του αλγορίθμου. Ορίζονται οι χώροι παρατήρησης και δράσης και κάθε άλλη παράμετρος των επιπλέον στοιχείων της προσομοίωσης. Στο δεύτερο βήμα δημιουργείται η συνάρτηση που ανανεώνει το περιβάλλον με το πέρας κάθε επεισοδίου εκπαίδευσης. Αυτό συνεπάγεται την επανατοποθέτηση του οχήματος και των αισθητήρων στην προσομοίωση και τον υπολογισμό της βέλτιστης διαδρομής. Σε αυτό το βήμα επιστρέφεται κάθε φορά η συνολική παρατήρηση που λαμβάνει ο αλγόριθμος για τους αναγκαίους υπολογισμούς. Τέλος στο τρίτο στάδιο δημιουργείται η συνάρτηση βήματος μέσω της οποίας εκτελεί τις δράσεις του ο αλγόριθμος, συλλέγονται οι πληροφορίες από τον χώρο παρατήρησης και υπολογίζεται η ανταμοιβή για κάθε ενέργεια.

2.4.1 Εύρεση Βέλτιστης Διαδρομής με τη βοήθεια του A^*

Το μοντέλο RL που αναπτύσσεται σε αυτήν την εργασία καλείται να ελέγξει την πλοήγηση ενός αυτόνομου οχήματος. Βασικό μέρος σε αυτή τη διαδικασία είναι η εύρεση της βέλτιστης διαδρομής μεταξύ αρχικού και τελικού σημείου κατά την οποία θα πλοηγηθεί το όχημα. Για την εύρεση λοιπόν αυτής της διαδρομής επιλέχθηκε ο αλγόριθμος A^* [41]. Πρόκειται για αλγόριθμο που υπολογίζει την διαδρομή με το μικρότερο κόστος μεταξύ δύο σημείων ελέγχοντας κάθε πιθανή λύση.

Η συνάρτηση προς ελαχιστοποίηση είναι η $f(n) = g(n) + h(n)$ (2.2), όπου $g(n)$ είναι το κόστος μετάβασης έως το σημείο που βρίσκεται ο αλγόριθμος κάθε φορά και $h(n)$ είναι το εκτιμώμενο κόστος για την μετάβαση στα υπό εξέταση σημεία. Προφανώς επιλέγεται το ελάχιστο κόστος. Η τιμή της παραμέτρου $h(n)$ δίνεται από μεθόδους εκτίμησης, ώστε να μην χρειαστεί να υπολογιστεί απόλυτα η τιμή χάριν εξοικονόμησης υπολογιστικού κόστους. Η κάθε μία από τις τρεις μεθόδους εφαρμόζεται αναλόγως τη φύση του προβλήματος όπως περιγράφεται στη συνέχεια. Η Εικόνα 35 βοηθά στην καλύτερη κατανόηση των περιπτώσεων χρήσης κάθε μεθόδου.

Αρχικά η απόσταση Μανχάταν, υπολογίζει το άθροισμα των απόλυτων αποστάσεων μεταξύ παρόντος και επόμενου σημείου, ως προς του άξονες x & y όπως φαίνεται στην εξίσωση 2.2. Χρησιμοποιείται όταν η κίνηση επιτρέπεται μόνο σε τέσσερεις διευθύνσεις, δηλαδή μόνο στον οριζόντιο και στον κάθετο άξονα.

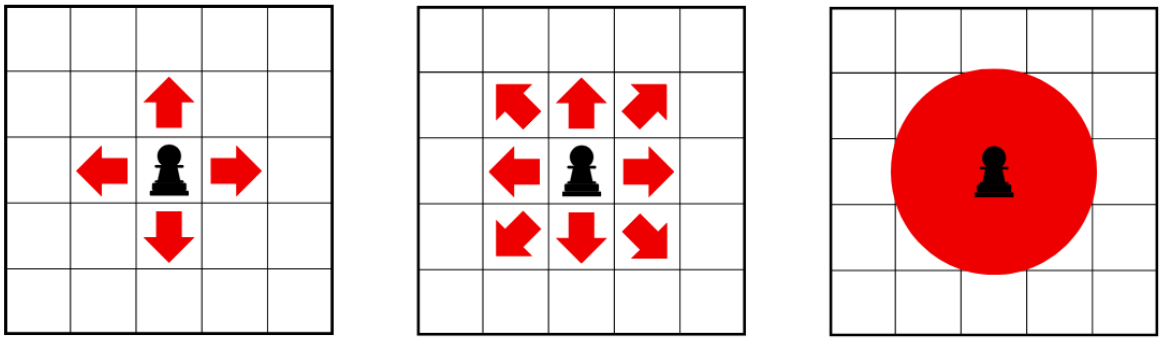
$$h = |x - x_0| + |y - y_0| \tag{2.3}$$

Στη συνέχεια όταν η κίνηση επιτρέπεται σε 8 διευθύνσεις όπως φαίνεται στην Εικόνα 35 στο κέντρο, η εκτίμηση δίνεται από τον τύπο της διαγωνίου απόστασης όπως φαίνεται στη συνέχεια. Οι παράμετροι D_1 που είναι το μήκος κάθε κόμβου (συνήθως 1) και D_2 που είναι η απόσταση μεταξύ δύο διαδοχικών κόμβων που ισούται με $\sqrt{2}$, όσο δηλαδή η διαγώνιως τετραγώνου με πλευρά μήκους 1.

$$h = D_1 * (|x - x_0| + |y - y_0|) + (D_2 - 2 * D_1) * \min(|x - x_0|, |y - y_0|) \tag{2.4}$$

Τέλος αν η κίνηση μπορεί να γίνει σε οποιαδήποτε κατεύθυνση όπως δεξιά στην Εικόνα 35, τότε επιλέγεται η Ευκλείδεια απόσταση η οποία δίνεται από την ακόλουθη συνάρτηση.

$$h = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{2.5}$$

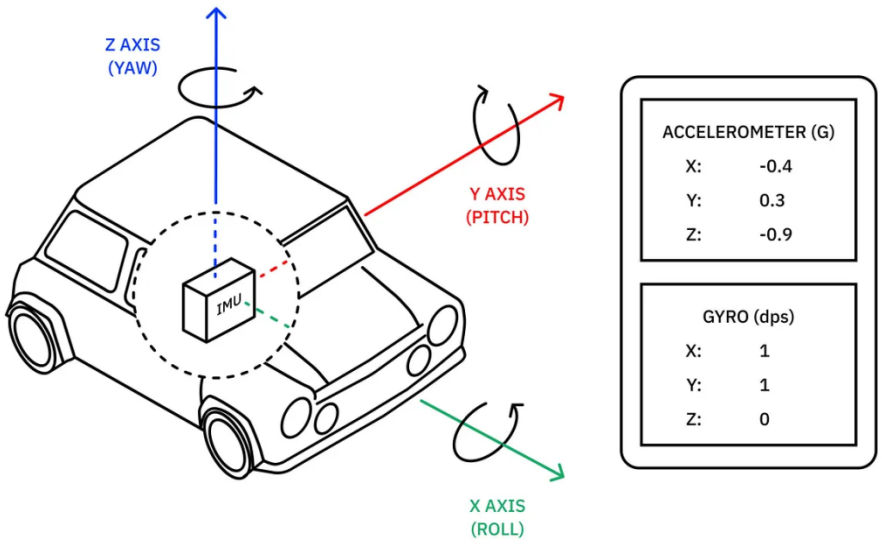


Εικόνα 35 Αριστερά: κίνηση σε 4 διευθύνσεις, Κέντρο: κίνηση σε 8 διευθύνσεις, Δεξιά: κίνηση προς πάσα κατεύθυνση

2.4.2 Εντοπισμός Θέσης & Αισθητήρια Όργανα

Στο σημείο αυτό αφού έχει εξηγηθεί το πώς βρίσκεται η πορεία που πρέπει να ακολουθήσει το όχημα και πώς διαχειρίζονται τα δεδομένα θα πρέπει να γίνει αναφορά στους αισθητήρες θέσης. Το αυτοκίνητο φέρει μία μονάδα GNSS και μία IMU, των οποίων τα δεδομένα αξιοποιούνται για τον εντοπισμό του στίγματος του οχήματος αλλά και τον προσανατολισμό του. Το CARLA δίνει την δυνατότητα χρήσης συστήματος συντεταγμένων βάσει μοντέλου GPS, έτσι κάθε στιγμή η μονάδα GNSS δίνει το γεωγραφικές συντεταγμένες του οχήματος στον χάρτη και μπορεί να εντοπίζεται η θέση του ως προς τα επιθυμητά σημεία.

Η IMU έχοντας ενσωματωμένη πυξίδα βρίσκει την απόκλιση από τον γεωγραφικό βορρά του προσομοιωτή και έτσι γνωρίζοντας την κατεύθυνση στην οποία κινείται το όχημα είναι δυνατός ο υπολογισμός της απόκλισης της πορείας του από την κατεύθυνση που θα το οδηγούσε στο σημείο στόχο (Εικόνα 36).



Εικόνα 36 Αναπαράσταση Λειτουργίας μίας IMU [42]

2.5 Stable Baselines 3

Για την εκπαίδευση των μοντέλων ενισχυτικής μάθησης χρησιμοποιήθηκε η βιβλιοθήκη Python Stable Baselines3 [43], που παρέχει έτοιμες συναρτήσεις για την εφαρμογή των αλγορίθμων. Είναι ανοιχτού κώδικα, δομημένη πάνω στο PyTorch και ουσιαστικά απλοποιεί την δημιουργία των προγραμμάτων για την εκπαίδευση ή την εφαρμογή εκπαιδευμένων μοντέλων. Καλύπτει πλήρως τις ανάγκες που προκύπτουν κατά τη δημιουργία RL μοντέλων καθώς εμπεριέχει παραδείγματα και συναρτήσεις έτοιμα προς χρήση. Είναι σχεδιασμένη ώστε να λειτουργεί σε συνεργασία με το GYM και ως εκ τούτου είναι κατάλληλή για τις εφαρμογές αυτής της εργασίας. Ακόμα η δυνατότητα κλήσης συναρτήσεων συλλογής δεδομένων και απεικόνισής τους σε TensorBoard είναι επιπλέον θετικά χαρακτηριστικά.

Όσον αφορά στην εφαρμογή της βιβλιοθήκης, έγιναν τα εξής βήματα. Αρχικά δημιουργήθηκε το πρόγραμμα *train.py* το οποίο είναι υπεύθυνο για την εκπαίδευση του αλγορίθμου και των παραμέτρων του. Καθένας από τους υπό εξέταση αλγορίθμους κλήθηκε ως module που υπάρχει ήδη στη βιβλιοθήκη και μέσω κατάλληλων συναρτήσεων ορίστηκαν οι διάφορες παράμετροι της εκπαίδευσης. Στο δεύτερο βήμα αφού ολοκληρώθηκε η εκπαίδευση των αλγορίθμων, δημιουργήθηκε το πρόγραμμα *load.py* το οποίο χρησιμοποιείται για να «φορτώνει» το αλγόριθμο στην πλατφόρμα στην οποία θα χρησιμοποιηθεί, στο CarlaEnv.

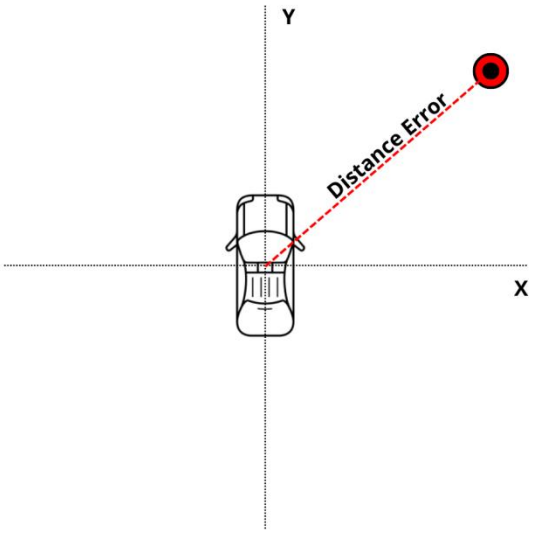
2.6 Δημιουργία Μοντέλου Ενισχυτικής Μάθησης

2.6.1 Χώρος Παρατήρησης

Όπως αναφέρθηκε και στην πρώτη ενότητα ένα από τα δομικά μέρη του μοντέλου RL είναι ο χώρος παρατήρησης, στον οποίο είναι ορισμένες οι μεταβλητές μέσω των οποίων ο αλγόριθμος αξιολογεί τα αποτελέσματα της δράσης του. Στο παρόν μοντέλο υπάρχουν τρεις τέτοιες μεταβλητές, που θα περιγραφούν αναλυτικά στη συνέχεια.

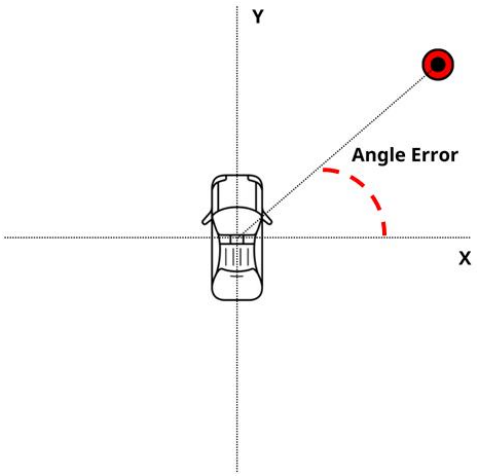
Η πρώτη μεταβλητή είναι η απόσταση του στίγματος του οχήματος κάθε στιγμή από το σημείο της βέλτιστης διαδρομής στο οποίο θα έπρεπε να βρίσκεται και λαμβάνει τιμές από 0 έως 22 μέτρα, διάστημα που είναι ικανοποιητικό καθώς εάν το αυτοκίνητο παρέκλινε τόσο θα ήταν σίγουρα εκτός των επιτρεπτών ορίων (Εικόνα 37). Ο υπολογισμός αυτής της τιμής γίνεται με εφαρμογή του τύπου της Ευκλείδειας απόστασης μεταξύ των δύο συντεταγμένων του οχήματος. Όπως προαναφέρθηκε το CARLA χρησιμοποιεί σύστημα συντεταγμένων GPS, το γεωγραφικό μήκος και πλάτος μετρούνται σε μοίρες και κάθε μία μοίρα αντιστοιχεί κατά προσέγγιση σε 111.000 μέτρα. Έτσι λαμβάνοντας το στίγμα του αυτοκινήτου και βρίσκοντας τη διαφορά με το σημείο σε μοίρες, το αποτέλεσμα κοινωνικοποιείται στην κλίμακα [0-22] και έπειτα ακολουθεί η εξίσωση 2.6 η οποία δίνει ακριβώς την τιμή της μεταβλητής παρατήρησης.

$$D = \sqrt{(διαφορά\ γεωγραφικού\ πλάτους)^2 + (διαφορά\ γεωγραφικού\ μήκους)^2} \quad (2.6)$$



Εικόνα 37 Απόσταση Βέλτιστου Σημείου

Η δεύτερη μεταβλητή είναι η γωνία απόκλισης της κατεύθυνσης του οχήματος ως προς το επόμενο σημείο-στόχο. Η μεταβλητή αυτή μετράται σε μοίρες και το πεδίο τιμών της είναι το [180,-180], με το 0 να σημαίνει ότι το όχημα κινείται ακριβώς προς το σημείο-στόχο. Ο προσανατολισμός του αυτοκινήτου ως προς τον βορρά δίνεται από την IMU σε ακτίνια και μέσω της μετατροπής $μοίρες = \frac{ακτίνια*180}{\pi}$ λαμβάνεται σε μοίρες, ωστόσο θα πρέπει να υπολογιστεί και ως προς το επιθυμητό σημείο. Γνωρίζοντας το σημείου και το στίγμα του οχήματος, δημιουργείται ένα σχετικό σύστημα συντεταγμένων με κέντρο το όχημα. Στη συνέχεια τοποθετείται το σημείο στο σχετικό αυτό σύστημα και με την συνάρτηση $atan2$ υπολογίζεται η απόκλισή του από τον άξονα Y και συνεπώς από την κατεύθυνση του οχήματος (Εικόνα 38).



Εικόνα 38 Απόκλιση Κατεύθυνσης

Η τρίτη και τελευταία μεταβλητή είναι η ταχύτητα του οχήματος. Η μεταβλητή αυτή μετράται ώστε να μπορεί έπειτα να διασφαλιστεί η εντός των μηχανικών ορίων του οχήματος ταχύτητα που είναι τα $80km/h$. Το CARLA διαθέτει την συνάρτηση *get_velocity()*, η οποία λειτουργεί ως ταχύμετρο και δίνει την ταχύτητα του οχήματος ή οποιουδήποτε άλλου *actor* για ορισμένο άξονα κίνησης. Έτσι έχοντας την ταχύτητα στους άξονες X και Y χρησιμοποιείται η εξίσωση 2.7 και υπολογίζεται η ταχύτητα στην κατεύθυνση του EcoCar. Τέλος μετατρέπονται οι μονάδες από m/s σε km/h και έχει ολοκληρωθεί ο χώρος παρατήρησης.

$$V = \sqrt{(ταχύτητα\ X)^2 + (ταχύτητα\ Y)^2} \tag{2.6}$$

2.6.2 Χώρος Δράσης

Ο χώρος δράσης του μοντέλου αποτελείται από τις μεταβλητές εκείνες που μπορούν λαμβάνουν τιμές τέτοιες ώστε να μεταβάλλεται η κατάσταση του περιβάλλοντος. Στην εφαρμογή αυτή χρησιμοποιούνται δύο μεταβλητές δράσης, η επιτάχυνση και η πηδαλιούχηση, δηλαδή το κατά πόσο επιταχύνει ή επιβραδύνει και στρίβει το όχημα.

Η επιτάχυνση λαμβάνει τιμές στο διάστημα [-1,1] όπου -1 σημαίνει 100% φρένο ενώ το 1 σημαίνει πλήρης επιτάχυνση. Εφόσον λάβει την τιμή 0 διατηρεί την ταχύτητα που έχει. Όσον αφορά την πηδαλιούχηση και αυτής της μεταβλητής οι τιμές βρίσκονται στο [-1,1], με την τιμή -1 να σημαίνει ότι στρίβει τελείως αριστερά και με την τιμή 1 δεξιά. Όταν λαμβάνεται η τιμή 0 το όχημα κρατά σταθερή την πορεία του.

2.6.3 Συνάρτηση Επιβράβευσης

Η συνάρτηση επιβράβευσης αποτελεί ένα πολύ βασικό μέρος του μοντέλου, αφού μέσω αυτής υπολογίζεται κάθε φορά το αν η δράση του αλγορίθμου κινούταν προς την επιθυμητή κατεύθυνση ή όχι. Εν προκειμένω δημιουργήθηκαν 6 παράμετροι με το ανάλογο βάρος στη συνάρτηση η κάθε μία αναλόγως την σημασία της.

Αρχικά ο πρώτος παράγοντας είναι η ταχύτητα του οχήματος που δεν πρέπει να υπερβαίνει ούτε τα μηχανικά όρια αλλά ούτε και το όριο ταχύτητος εντός κατοικημένων περιοχών που είναι τα 50 km/h. Εφόσον λοιπόν η ταχύτητα είναι εντός ορίων έχει συντελεστή 0 στη συνάρτηση και για κάθε βήμα που υπερβαίνει το όριο λαμβάνει την τιμή -2. Ονομάζοντας λοιπόν τη μεταβλητή στη συνάρτηση ως $speed_{reward}$, ο κανόνας αυτός περιγράφεται μαθηματικά στην εξίσωση 2.7.

$$speed_{reward} = \begin{cases} -2\ \boldsymbol{εάν}\ ταχύτητα > 50km/h \\ 0\ \boldsymbol{αλλιώς} \end{cases} \tag{2.7}$$

Οι επόμενες δύο μεταβλητές αφορούν τις εναπομείνασες τιμές του χώρου παρατήρησης, την απόσταση και την απόκλιση από το επιθυμητό σημείο. Και οι δύο τιμές επιβραβεύονται

όταν λαμβάνουν την τιμή 0, ειδάλλως αφαιρούνται οι τιμές που προκύπτουν από τις 2.8 και 2.9.

$$distance_{reward} = \begin{cases} e^{\text{τιμή επιτάχυνσης}} \ \boldsymbol{\text{εάν}} \ \text{επιτάχυνση} > 0 \\ -e^{\text{τιμή επιτάχυνσης}} \ \boldsymbol{\text{εάν}} \ \text{επιτάχυνση} < 0 \end{cases} \quad (2.8)$$

$$angle_{reward} = \begin{cases} e^{\text{τιμή επιτάχυνσης}} \ \boldsymbol{\text{εάν}} \ \text{επιτάχυνση} > 0 \\ -e^{\text{τιμή επιτάχυνσης}} \ \boldsymbol{\text{εάν}} \ \text{επιτάχυνση} < 0 \end{cases} \quad (2.9)$$

Η επιτάχυνση είναι ο επόμενος παράγοντας. Με στόχο το όχημα να έχει τη μέγιστη δυνατή επιτάχυνση καθ' όλη την προσομοίωση ο παράγοντας αυτός ρυθμίζεται έτσι ώστε το μοντέλο να επιβραβεύεται όταν η επιτάχυνση είναι θετική και να τιμωρείται όταν είναι αρνητική. Για αυτόν τον λόγο η τιμή της επιτάχυνσης υψώνεται εις το e όπως φαίνεται στη 2.10.

$$acceleration_{reward} = \begin{cases} e^{\text{τιμή επιτάχυνσης}} \ \boldsymbol{\text{εάν}} \ \text{επιτάχυνση} > 0 \\ -e^{\text{τιμή επιτάχυνσης}} \ \boldsymbol{\text{εάν}} \ \text{επιτάχυνση} < 0 \end{cases} \quad (2.10)$$

Όσον αφορά στην πηδαλιούχηση του οχήματος, προτιμάται να μην αλλάζει τακτικά κατεύθυνση ώστε να αποφεύγεται το ζιγκ-ζαγκ. Έτσι όπως παρουσιάζεται και στην 2.11, εφόσον δεν είναι μηδενική η τιμή πάντοτε υπάρχει ποινή εάν στρίβει το όχημα.

$$steering_{reward} = 1 - 2 * |\text{τιμή πηδαλιούχησης}| \quad (2.11)$$

Τέλος προστίθεται ακόμα ένας παράγοντας επιβράβευσης ο $bonus\_reward$ που ενεργοποιείται όποτε το όχημα φθάνει ένα σημείο στόχο. Κατ' αυτόν τον τρόπο ενθαρρύνεται το μοντέλο να προσεγγίζει τη βέλτιστη διαδρομή.

$$bonus_{reward} = \begin{cases} \text{σημεία που προσεγγίστηκαν} \ \boldsymbol{\text{αν}} \ \text{προσεγγίστηκε σημείο} \\ 0 \ \boldsymbol{\text{αλλιώς}} \end{cases} \quad (2.11)$$

Συνδυάζοντας τους παραπάνω παράγοντες και πολλαπλασιάζοντάς τους με τα κατάλληλα βάρη προκύπτει η παρακάτω συνάρτηση επιβράβευσης, η οποία χρησιμοποιείται σε όλη την εργασία.

$$reward = speed_{reward} + distance_{reward} + 2 * angle_{reward} + acceleration_{reward} + 2 * steering_{reward} + 5 * bonus_{reward} \quad (2.12)$$

3. Πειράματα & Πειραματικά Αποτελέσματα

3.1 Σχεδίαση Πειραμάτων

Τα πειράματα που σχεδιάστηκαν, είχαν ως αρχικό στόχο την αξιολόγηση της εκτέλεσης της αυτόνομης πλοήγησης από το όχημα. Ως γνώμονας για αυτή την αξιολόγηση λαμβάνεται η εργασία [8], που εφαρμόζει με επιτυχία τον DDPG, υπό ορισμένες συνθήκες. Εν συνεχεία στην εν λόγω εργασία εφαρμόζονται οι αλγόριθμοι TD3 και SAC σε δύο παραλλαγές ο καθένας ώστε αρχικά να αποδειχθεί ότι επιτυχώς ολοκληρώνουν την διαδρομή και στη συνέχεια με την μεταβολή του ρυθμού μάθησης να συγκριθούν τα αποτελέσματά τους.

Συνολικά αξιολογήθηκαν δύο εκδοχές από τον κάθε αλγόριθμο. Κάθε ένας τους εκπαιδεύτηκε για **50.000 βήματα** αρχικά με **ρυθμό μάθησης 0,001** και στη συνέχεια με **ρυθμό μάθησης 0,0003**. Η επιλογή αυτών των τιμών βασίστηκε στις προκαθορισμένες παραμέτρους της SB3 όπου για τον SAC ορίζει τον ρυθμό μάθησης σε 0,0003 και για τον TD3 σε 0,001. Έτσι θεωρήθηκε ενδιαφέρουσα η σύγκριση των δύο αλγορίθμων έχοντας αμφότεροι τις ίδιες παραμέτρους. Στον παρακάτω πίνακα παρουσιάζονται συνολικά όλες οι υπερπαράμετροι που είχαν τα μοντέλα που δοκιμάστηκαν.

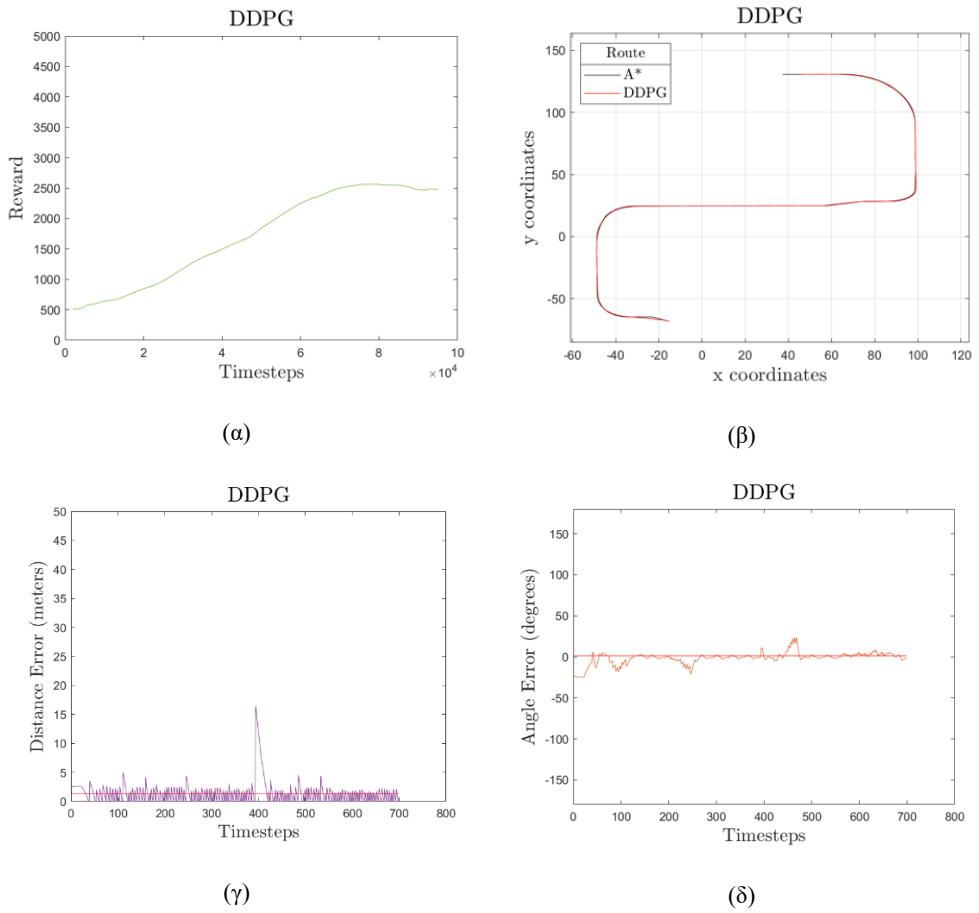
Πίνακας 6 Κατάλογος μοντέλων αλγορίθμων που αξιολογήθηκαν

Υπερπαράμετροι	**SAC**		**TD3**	
	Μοντέλο 1	**Μοντέλο 2**	**Μοντέλο 3**	**Μοντέλο 4**
Ρυθμός Μάθησης	0,001	0,0003	0,001	0,0003
Πολιτική	Mlp	Mlp	Mlp	Mlp
Συντελεστής Polyak	0,005	0,005	0,005	0,005
Συντελεστής γ	0,99	0,99	0,99	0,99
Batch Size	256	256	256	256

Η διαδικασία που ακολουθήθηκε έχει ως εξής. Αρχικά αφού τελείωσαν όλες οι απαραίτητες ρυθμίσεις, κάθε μοντέλο εκπαιδεύτηκε για 50.000 βήματα στην πόλη "Town10'' του CARLA. Μέσω ρυθμίσεων στο το πρόγραμμα *train.py* που δημιουργήθηκε με την χρήση της SB3 ορίστηκε η αποθήκευση ενός μοντέλου του αλγορίθμου ανά 1.000 βήματα. Με το πέρας των 50.000 βημάτων και έχοντας ως δείκτη την μέγιστη τιμή επιβράβευσης που πέτυχε κάθε αποθηκευμένο μοντέλο, επιλέχθηκαν αυτά με το μεγαλύτερο σκορ για να αξιολογηθούν περαιτέρω. Στην επόμενη ενότητα παρουσιάζονται συνοπτικά τα αποτελέσματα της εφαρμογής του DDPG στην εργασία [8], ως σημείο αναφοράς.

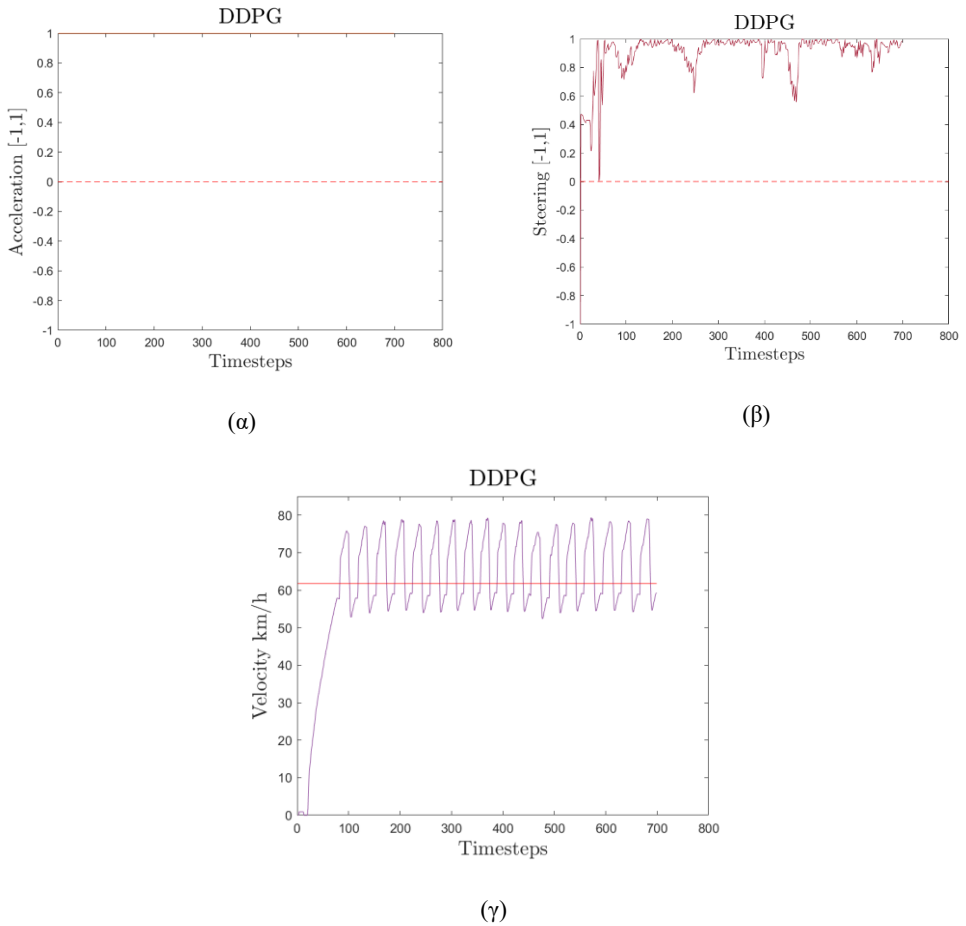
3.2 DDPG

Η εφαρμογή του DDPG αποτελούνταν από την εκπαίδευση του και την εφαρμογή όπως και στην παρούσα εργασία και η επίδοσή του όπως αποδεικνύεται στη συνέχεια είναι ιδιαίτερα καλή. Πρέπει να σημειωθεί ότι η εκπαίδευση του DDPG έγινε για σχεδόν 100.000 βήματα, εν αντιθέσει των μοντέλων που εκπαιδεύτηκαν για 50.000.



Εικόνα 39 Αποτελέσματα εφαρμογής DDPG (α) Εξέλιξη της συνάρτησης επιβράβευσης, (β) τροχιά του οχήματος με εφαρμογή του DDPG, (γ) μεταβολή του σφάλματος απόστασης, (δ) μεταβολή του σφάλματος κατεύθυνσης [8]

Ξεκινώντας με την πορεία της συνάρτησης επιβράβευσης στην Εικόνα 39 (α), βλέπει κανείς ότι έχει ομαλή και συνεχώς αύξουσα πορεία έως το σημείο 75.000 περίπου, προσεγγίζοντας έτσι την βέλτιστη μορφή της καμπύλης εκπαίδευσης. Στη συνέχεια στην Εικόνα 39 (β) οι διαδρομές του A^* και του οχήματος συμπίπτουν, απόδειξη ότι ο αλγόριθμος κατάφερε να προσεγγίσει σε μεγάλο βαθμό την βέλτιστη διαδρομή αφού το μέσο σφάλμα όπως φαίνεται στην Εικόνα 39 (γ) είναι περίπου στο 1,5 μέτρο. Τέλος και το σφάλμα απόκλισης στην Εικόνα 39 (δ) δεν παίρνει μεγάλες τιμές αφού κατά μέσο όρο αποκλίνει περίπου για 1 μοίρα.

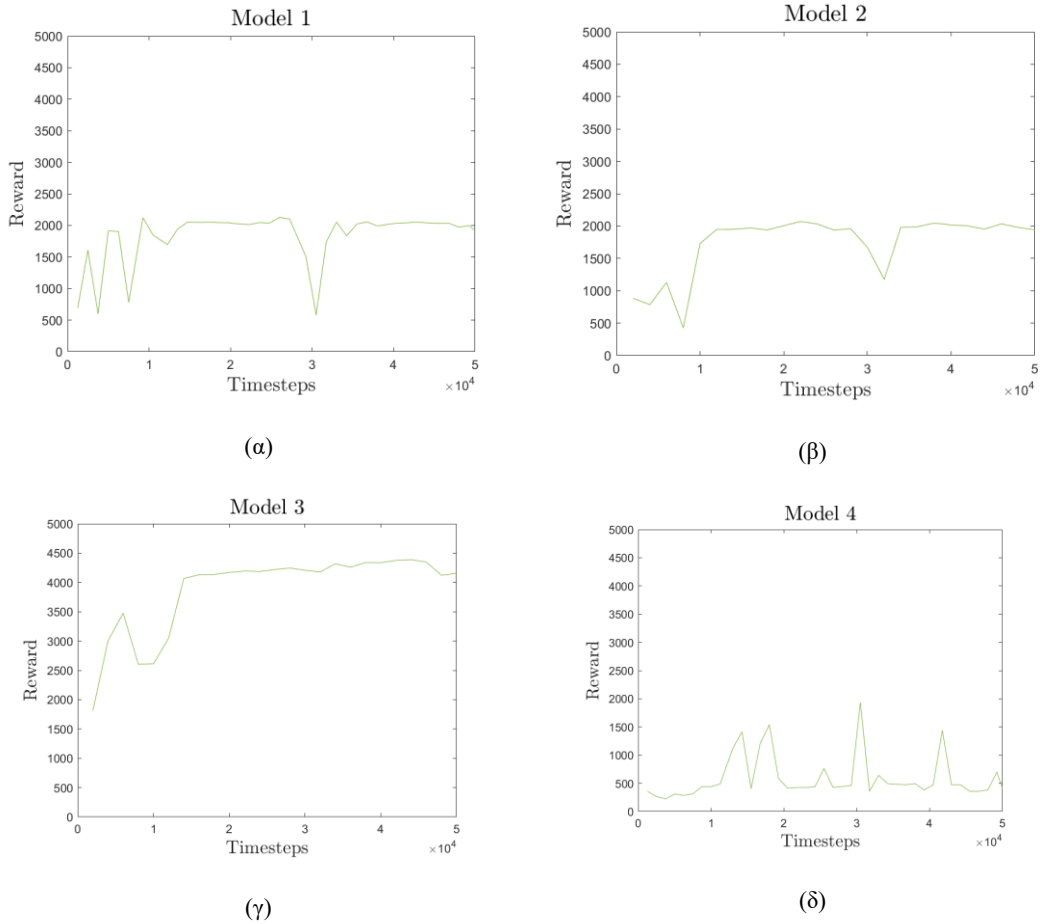


Εικόνα 40 Αποτελέσματα εφαρμογής DDPG (α) Μεταβολή της επιτάχυνσης, (β) μεταβολή της πηδαλιούχησης, (γ) μεταβολή της ταχύτητας [8]

Στη συνεχεία παρουσιάζονται οι τιμές των μεταβλητών δράσης και το προφίλ ταχύτητας. Η τιμή της επιτάχυνσης στην Εικόνα 40 (α) είναι εξ αρχής η μέγιστη δυνατή, εν αντιθέσει το κατά πόσο στρίβει το όχημα αυξομειώνεται όπως και είναι λογικό για να διορθώνει την πορεία κάθε στιγμή, όπως παρουσιάζει το γράφημα στην Εικόνα 40 (β). Το προφίλ της ταχύτητας στην Εικόνα 40 (γ) παρουσιάζει συνεχείς αυξομειώσεις δημιουργώντας μία μορφή ‘πριονιού’, αποτέλεσμα της προσπάθειας για διόρθωση κάθε φορά βάσει των περιορισμών. Όσο για τη μέση ταχύτητα αυτή είναι υψηλή καθώς υπερβαίνει τα $60km/h$. Η γενική εικόνα της επίδοσης του αλγορίθμου είναι ιδιαίτερα καλή εφόσον με μικρές αποκλίσεις και υψηλή ταχύτητα φτάνει στον προορισμό του. Αυτό που θα παρατηρήσει ο αναγνώστης είναι το γεγονός ότι σε αυτή την περίπτωση η πορεία ολοκληρώνεται σε περίπου 700 βήματα ενώ στις επόμενες τα μοντέλα 2, 3 & 4 διαρκούν για 800.

3.3 Πειραματικά Αποτελέσματα

Τα αποτελέσματα που εξάχθηκαν και αξίζουν μελέτης είναι αρχικά η πορεία της επιβράβευσης κάθε μοντέλου όπως παρουσιάζεται στα γραφήματα της Εικόνα 41.



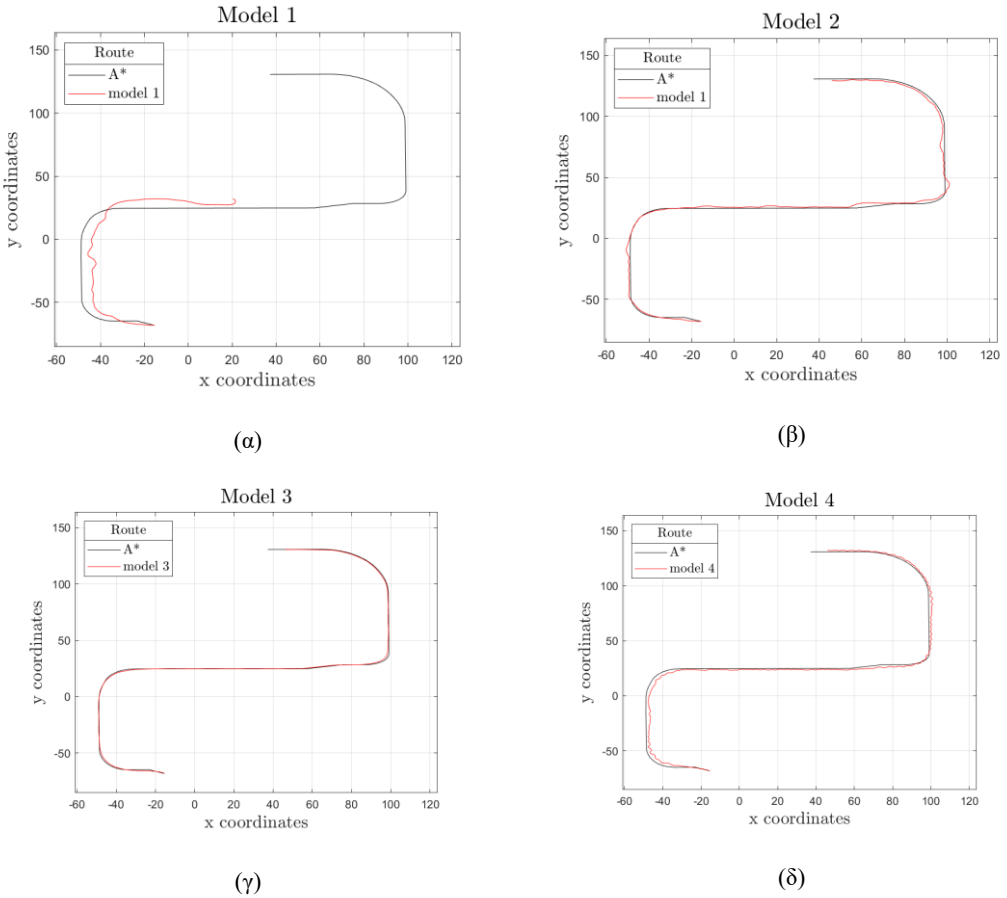
Εικόνα 41 Οι καμπύλες επιβράβευσης των μοντέλων RL (α) SAC lr = 0.001, (β) SAC lr = 0.0003, (γ) TD3 lr = 0.001, (δ) TD3 lr = 0.0003

Τα γραφήματα στην Εικόνα 41 ακολουθούν την πορεία της συνάρτησης επιβράβευσης του κάθε μοντέλου λαμβάνοντας τιμές ανά 1.000 βήματα του αλγορίθμου. Αυτός είναι και ο λόγος που η αρχή της κάθε καμπύλης δεν είναι στο 0 (όπου και η τιμή θα ήταν 0), αλλά εκκινούν από το χιλιοστό βήμα και φθάνουν έως το 50.000 όπου και ολοκληρώνεται η εκπαίδευση. Θα πρέπει να σημειωθεί ότι οι τιμές του άξονα Υ δεν προσαρμόστηκαν στην κάθε καμπύλη, ώστε να είναι εμφανής η σχετική διαφορά της μεταξύ των μοντέλων.

Στα πρώτα δύο μοντέλα, στις Εικόνα 41 (α), (β), όπου εξετάζεται ο αλγόριθμος SAC η καμπύλη φαίνεται να έχει παρόμοια πορεία με πιο βελτιωμένο αποτέλεσμα στο μοντέλο 2. Παρότι λίγο μετά τα 30.000 βήματα αμφότερα υπάρχει μία βύθιση στην τιμή της επιβράβευσης, η γενική τάση της καμπύλης είναι η επιθυμητή καθότι στο τέλος είναι μακράν υψηλότερη της αρχικής, γεγονός που αποδεικνύει ότι βάσει της ορισμένης συνάρτησης ο αλγόριθμος μαθαίνει να δρα καλύτερα κάθε φορά.

Όσον αφορά στα δύο επόμενα μοντέλα του αλγορίθμου TD3 η πρώτη προσέγγιση παρουσιάζει εξαιρετική απόδοση και μάλιστα την υψηλότερη εκ των τεσσάρων με την καμπύλη να είναι σχεδόν η ιδανική εξαιρώντας την πτώση στο 10.000 βήμα. Το μοντέλο 4 ωστόσο δεν παρουσιάζει ιδανική πορεία εκπαίδευσης και λαμβάνοντας υπόψη το γεγονός ότι η τάση της καμπύλης είναι σταθερή, το μοντέλο δεν βελτιώθηκε σε αυτή την πορεία. Βέβαια όπως θα παρουσιαστεί στη συνέχεια το μοντέλο 4 είχε καλύτερη επίδοση στην πλοήγηση έναντι άλλων με καλύτερη πορεία εκπαίδευσης. Η αντίφαση αυτή υποδεικνύει ότι ο αλγόριθμος αυτός μπορεί να αποδώσει βέλτιστα αποτελέσματα στο παρόν πρόβλημα αλλά με διαφορετική ρύθμιση των παραμέτρων από την υπάρχουσα.

Στη συνέχεια παρουσιάζεται ίσως το πιο κρίσιμο κριτήριο απόδοσης του κάθε αλγορίθμου, η πορεία του εν συγκρίσει με τη βέλτιστη διαδρομή του A^*.



Εικόνα 42 Οι διαδρομές των μοντέλων εν συγκρίσει της βέλτιστης για την περίπτωση (α) SAC lr = 0.001, (β) SAC lr = 0.0003, (γ) TD3 lr = 0.001, (δ) TD3 lr = 0.0003

Τα γραφήματα της Εικόνα 42, αναπαριστούν την διαδρομή του κάθε μοντέλου με κόκκινο και το βέλτιστο μονοπάτι του A^* με μαύρο σε καρτεσιανό σύστημα συντεταγμένων. Προφανώς το μοντέλο 3 του TD3 με ρυθμό μάθησης 0,001 είναι το βέλτιστο καθότι η διαδρομή που ακολούθησε δεν παρεκκλίνει καθόλου της πορείας του A^*. Αυτό βέβαια εύκολα θα μπορούσε κανείς να το προβλέψει παρατηρώντας την συνάρτηση επιβράβευσης. Αυτό που δεν θα

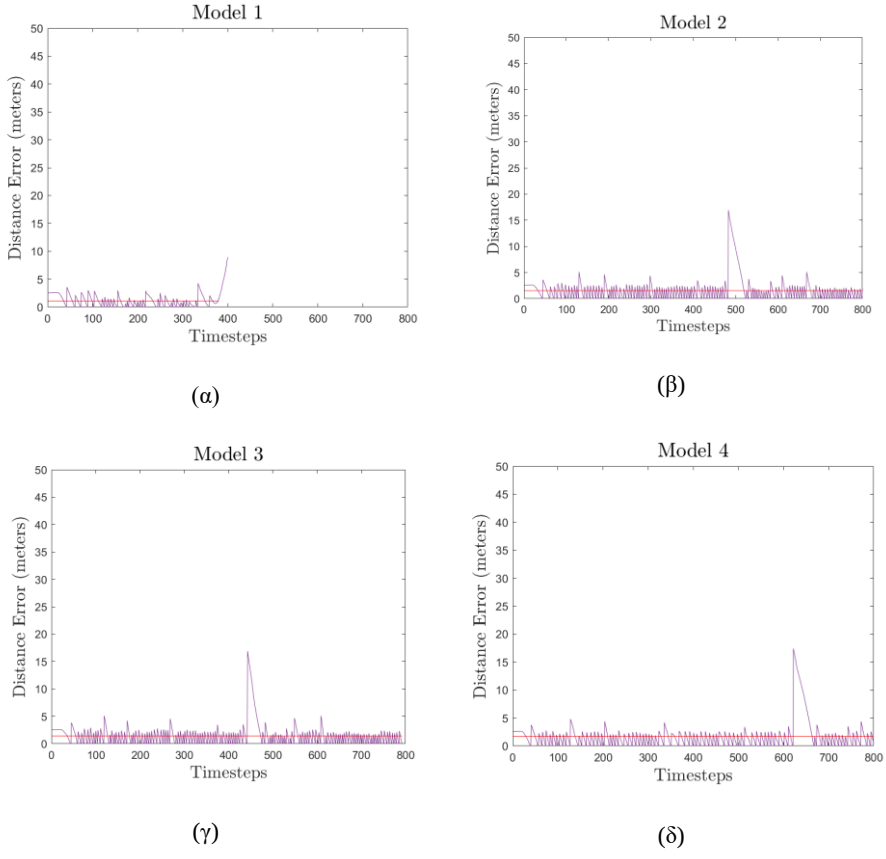
μπορούσε να προβλεφθεί από τα γραφήματα της Εικόνα 41 είναι οι πορείες των τριών άλλων μοντέλων. Τα μοντέλα 2 και 4 έχουν παρόμοιες διαδρομές οι οποίες συγκλίνουν στη βέλτιστη με μικρές παρεκκλίσεις ενίοτε. Παρά τη μεγάλη διαφορά στις καμπύλες επιβραβεύσεως τους το αποτέλεσμα σε μία πιο ολιστική μορφή αξιολόγησης θα ήταν παρόμοιο. Τέλος η πορεία του μοντέλου 1 αποτελεί ισχυρό επιχείρημα στο ότι δεν μπορεί κάποιος να εξάγει σαφή συμπεράσματα κοιτώντας μόνο την πορεία της επιβράβευσης. Κατά την εκπαίδευση ο SAC με ρυθμό μάθησης 0,001 παρουσίασε πολύ καλύτερη συμπεριφορά από τον TD3 με ρυθμό μάθησης 0,0003 αλλά εν ώρα εκτέλεσης ο δεύτερος αποδείχθηκε πολύ αποτελεσματικότερος.

Οι διαφορές που υπάρχουν στα αποτελέσματα αυτά σχετίζονται άμεσα με τη φύση κάθε αλγορίθμου καθότι όπως αναλύθηκε στο πρώτο κεφάλαιο η αρχή λειτουργίας καθενός διαφέρει του άλλου. Υπό διαφορετικές συναρτήσεις επιβράβευσης πιθανότατα τα αποτελέσματα που προέκυψαν να ήταν ακριβώς τα αντίθετα. Ωστόσο δεδομένων των παρόντων ο αλγόριθμος που υπερισχύει είναι ο TD3 με ρυθμό μάθησης 0,001 καθώς η εφαρμογή του εκπαιδευμένου μοντέλου έδωσε ένα όχημα που ακολουθά την διαδρομή που του δίνεται πιστά. Στην Εικόνα 43 παρουσιάζονται διάφορα στιγμιότυπα του EcoCar κατά την διάρκεια της προσομοίωσης στο περιβάλλον του CARLA.

Εικόνα 43 Στιγμιότυπα από τις προσομοιώσεις του EcoCar

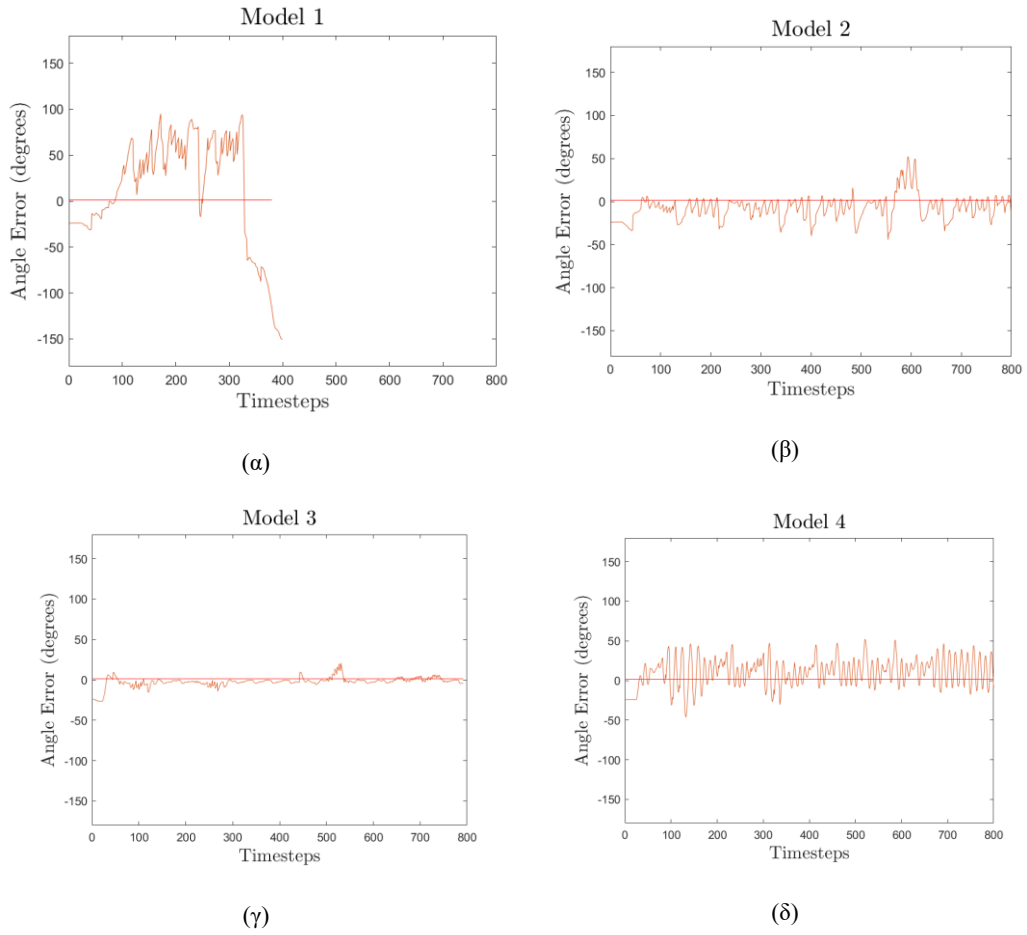
Εικόνα 44 Η βέλτιστη διαδρομή στην πόλη προσομοίωσης

Τα γραφήματα που ακολουθούν στην Εικόνα 45 είναι οι μεταβλητές των χώρων παρατήρησης και δράσης, πρέπει να σημειωθεί ότι όσον αφορά στο μοντέλο 1 οι ενδείξεις του σταματούν στο βήμα 400 έναντι του βήματος 800 το υπολοίπων αλγορίθμων καθότι εκεί είναι το σημείο όπου παύει η πορεία του όπως φαίνεται στην Εικόνα 42.



Εικόνα 45 Σφάλμα απόστασης μεταξύ βέλτιστου σημείου και σημείου οχήματος για την περίπτωση (α) SAC lr = 0.001, (β) SAC lr = 0.0003, (γ) TD3 lr = 0.001, (δ) TD3 lr = 0.0003

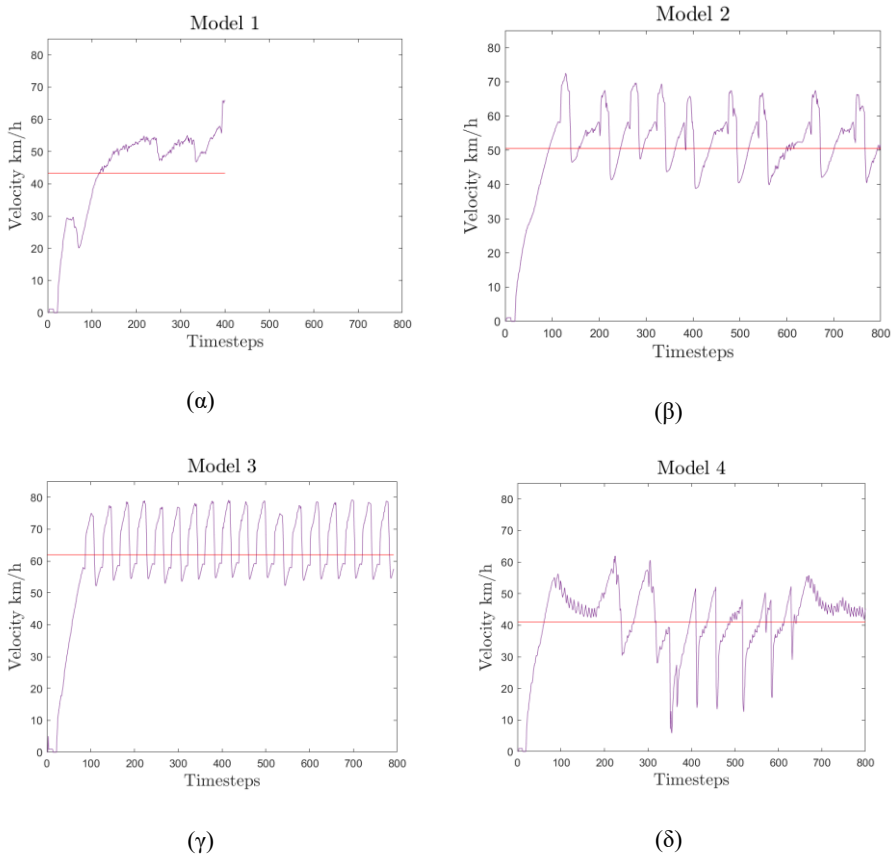
Ξεκινώντας στην Εικόνα 45, παρουσιάζεται η πορεία της απόστασης του αυτοκινήτου σε σχέση με το σημείο όπου έπρεπε να βρίσκεται. Σε όλα τα γραφήματα παρουσιάζεται με κόκκινη συνεχή γραμμή η μέση τιμή αυτού του σφάλματος η οποία δεν υπερβαίνει τα 1,4 μέτρα, σφάλμα ανεκτό για όχημα το οποίο κινείται ακολουθώντας μόνο σημεία στον χάρτη χωρίς παραμέτρους όπως η αναγνώριση της λωρίδας του δρόμου.



Εικόνα 46 Γωνία απόκλισης από το επιθυμητό σημείο (α) SAC lr = 0.001, (β) SAC lr = 0.0003, (γ) TD3 lr = 0.001, (δ) TD3 lr = 0.0003

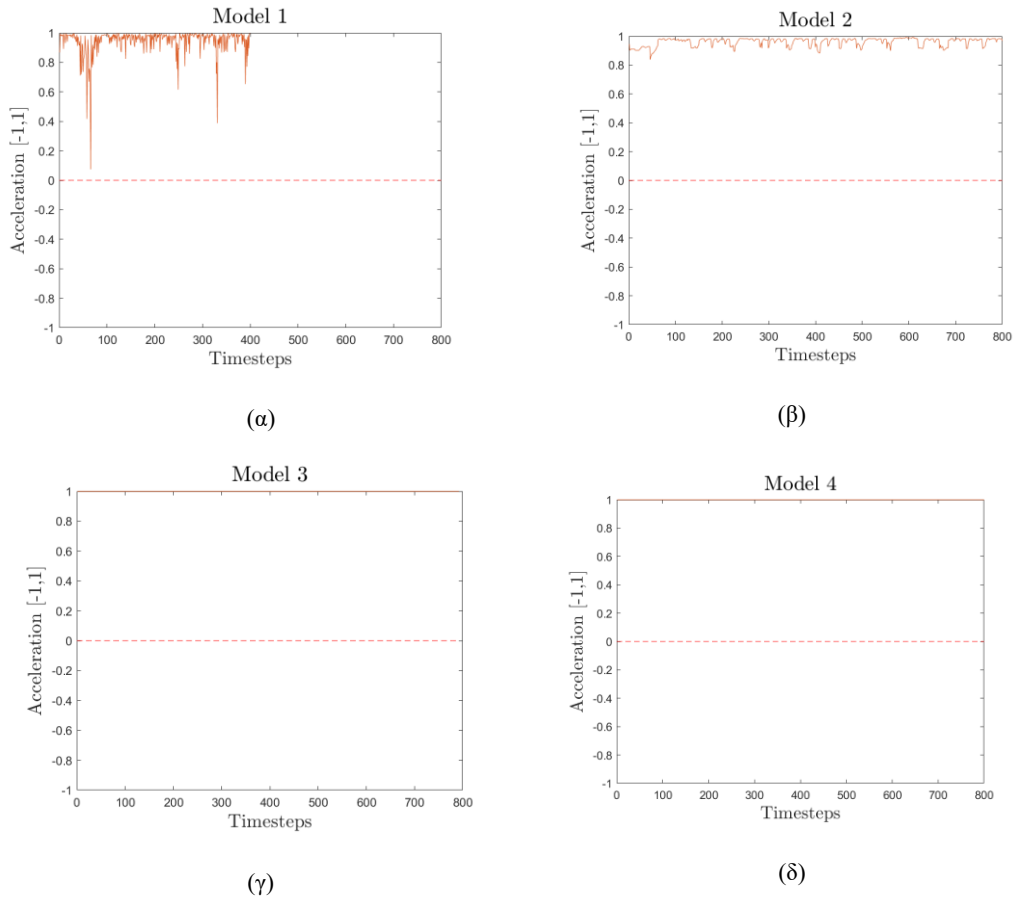
Συνεχίζοντας τα γραφήματα στην Εικόνα 46, παρουσιάζουν την επόμενη μεταβλητή παρατήρησης που είναι η γωνία απόκλισης από το επιθυμητό σημείο, όπως αυτή είχε επεξηγηθεί στο προηγούμενο κεφάλαιο. Το εύρος που μπορεί να πάρει κυμαίνεται στο διάστημα [-180,180] και η βέλτιστη τιμή του είναι το 0. Η κόκκινη συνεχή γραμμή είναι και εδώ η μέση τιμή της απόκλισης η οποία κυμαίνεται περίπου στη 1 μοίρα σε όλα τα μοντέλα. Είναι αξιοσημείωτο ότι και στο σφάλμα απόκλισης και στο σφάλμα απόστασης όλα τα μοντέλα έχουν σχεδόν την ίδια μέση τιμή γεγονός που υποδεικνύει ότι η συνάρτηση επιβράβευσης, της οποίας το αποτέλεσμα καθορίζει κάθε επόμενη κίνηση, είναι αξιόπιστη στην εφαρμογή της. Όσον αφορά στο ποιο μοντέλο έχει τη βέλτιστη απόδοση ως προς το σφάλμα απόκλισης, είναι

αυτό του οποίου οι τιμές κινούνται πιο κοντά στο 0, δηλαδή του μοντέλου 3. Τα μοντέλα 2 και 4 έχουν παρόμοια επίδοση με το ένα να κινείται επί του μηδενός και το άλλο υπό.



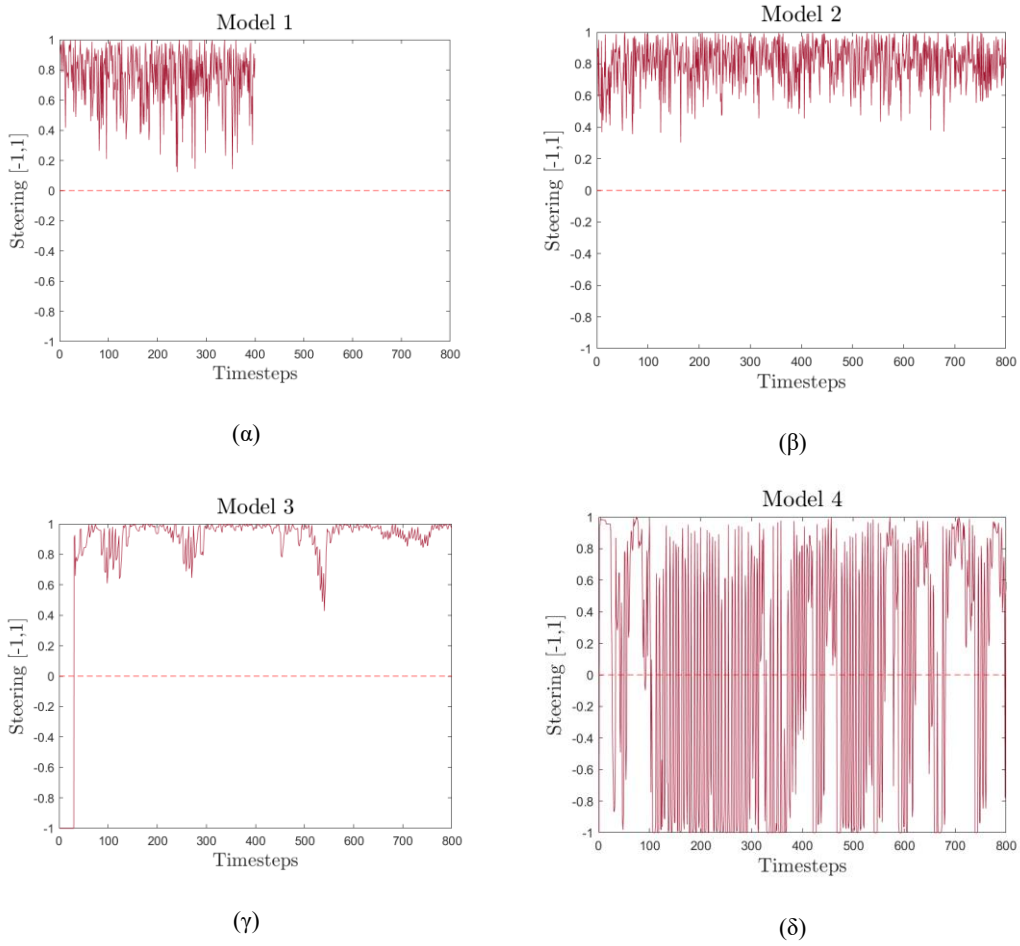
Εικόνα 47 Προφίλ ταχύτητας για την περίπτωση του (α) SAC lr = 0.001, (β) SAC lr = 0.0003, (γ) TD3 lr = 0.001, (δ) TD3 lr = 0.0003

Το προφίλ της ταχύτητας, που είναι και η τρίτη μεταβλητή του χώρου παρατήρησης, δείχνει το πόσο ομαλά και γρήγορα κινήθηκε το όχημα. Το μοντέλο 1 στην Εικόνα 47 (α) έως ότου σταματήσει να κινείται είχε συνεχώς αυξανόμενη ταχύτητα με μέση τιμή περίπου $42km/h$, φαίνεται από τα μοντέλα να είναι αυτό που κινήθηκε με τον ομαλότερο τρόπο καθώς δεν έχει συνεχείς αυξομειώσεις οι οποίες αναγνωρίζονται από το γράφημα με τη μορφή 'πριονιού' όπως στις άλλες τρεις περιπτώσεις. Πιο γρήγορα απ' όλα τα μοντέλα κινήθηκε το τρίτο στην Εικόνα 47 (γ) με μέση ταχύτητα περίπου $61km/h$ ενώ υιοθέτησε μία στρατηγική συνεχούς αυξομείωσης της ταχύτητας ώστε να μένει εντός του ορίου που καθορίστηκε, αντίστοιχα κινήθηκε και το δεύτερο μοντέλο στην Εικόνα 47 (β) χωρίς όμως να έχει το ίδιο υψηλή ταχύτητα. Όσον αφορά στο τελευταίο στην Εικόνα 47 (δ), αυτό είχε την χειρότερη απόδοση βάσει κριτηρίων άνεσης και ταχύτητας καθώς έχει την πιο ανώμαλη καμπύλη ταχύτητας αλλά και την μικρότερη μέση τιμή σε αυτή.



Εικόνα 48 Επιτάχυνση για την περίπτωση του (α) SAC lr = 0.001, (β) SAC lr = 0.0003, (γ) TD3 lr = 0.001, (δ) TD3 lr = 0.0003

Στα γραφήματα της Εικόνα 48 παρουσιάζεται η πορεία της τιμής της επιτάχυνσης σε κάθε περίπτωση. Όπως είχε περιγραφεί η επιτάχυνση και η πηδαλιούχηση λαμβάνουν τιμές στο διάστημα [-1,1] με το 0 να είναι το σημείο όπου δεν μεταβάλλεται η κατάσταση του οχήματος. Σε όλες τις περιπτώσεις φαίνεται ότι η επιτάχυνση έχει μονίμως θετικές τιμές και μάλιστα στα μοντέλα 3 και 4 στις Εικόνα 48 (γ) & (δ), είναι κατά το πλείστον στη μέγιστη τιμή με ανεπαίσθητες αυξομειώσεις, στο μοντέλο 2 στην Εικόνα 48 (β) φαίνεται περισσότερο ότι ανά διαστήματα μειώνεται το πολύ κατά 10%. Εν τέλει το πρώτο μοντέλο στην Εικόνα 48 (α) είναι αυτό που αισθητά μειώνει την επιτάχυνση. Σε αυτό το σημείο θα πρέπει να σημειωθεί ότι η πλήρης επιτάχυνση δεν σημαίνει μόνιμη αύξηση της ταχύτητας. Θα πρέπει ο αναγνώστης να έχει υπόψη ότι εντός του προσομοιωτή ισχύουν οι φυσικοί νόμοι και σαφώς και η απώλεια ενέργειας μέσω της τριβής. Για αυτό μία ελάττωση στην επιτάχυνση μπορεί να μη σημαίνει ότι ενεργοποιούνται τα φρένα αλλά εφόσον η παροχή ισχύος μειώνεται έτσι και μειώνεται η ταχύτητα κάνοντας την επίδραση των απωλειών γίνεται πιο αισθητή.



Εικόνα 49 Πηδαλιούχηση για την περίπτωση του (α) SAC lr = 0.001, (β) SAC lr = 0.0003, (γ) TD3 lr = 0.001, (δ) TD3 lr = 0.0003

Τέλος η δεύτερη μεταβλητή του χώρου δράσης, παρουσιάζεται στην Εικόνα 49. Η τιμή που δίνεται κάθε φορά στο αυτοκίνητο για να στρίψει κυμαίνεται και αυτή στο [-1,1] με το 0 να είναι η ευθεία κατεύθυνση. Όσο μικρότερη η διακύμανση της καμπύλης τόσο πιο ομαλή είναι η πορεία του οχήματος. Είναι εμφανές ότι το μοντέλο 3 στην Εικόνα 49 (γ) είναι αυτό με την πιο ομαλή πορεία, ενώ το μοντέλο 4 στην Εικόνα 49 (δ) είναι αυτό με την πιο ομαλή, παρόλο που και τα δύο χρησιμοποιούν τον ίδιο αλγόριθμο διαφέροντας μόνο στον ρυθμό μάθησης.

Εν κατακλείδι έχοντας λάβει ως δεδομένα κανείς τα παραπάνω αποτελέσματα μπορεί εύκολα να συμπεράνει ότι το μοντέλο του TD3 με ρυθμό μάθησης 0,001 είναι το πιο αποτελεσματικό και αξιόπιστο. Μάλιστα πρέπει να αναφερθεί η ομοιότητα των αποτελεσμάτων του με αυτά της εφαρμογής του DDPG που περιεγράφηκε στην προηγούμενη ενότητα. Τέλος σημειώνεται ότι και τα υπόλοιπα τρία μοντέλα εμπεριέχουν αξιόλογα σημεία τα οποία δεν θα μπορούσαν να εμφανιστούν αν λαμβανόταν ως μέτρο μόνο ένας παράγοντας και όχι ο συνδυασμός όλων των παραπάνω.

4. Συμπεράσματα & Μελλοντικές Επεκτάσεις

Στα πλαίσια της παρούσας εργασίας, εφαρμόστηκαν δύο διαφορετικοί αλγόριθμοι ενισχυτικής μάθησης πάνω στην αυτόνομη πλοήγηση ενός οχήματος. Ο στόχος της εργασίας ήταν η πλοήγησης του EcoCar βάσει της βέλτιστης διαδρομής που εξάχθηκε από την εφαρμογή του αλγορίθμου A^*. Οι προσομοιώσεις της εργασίας αυτής στηρίχθηκαν στην εργασία [8], όπου και δημιουργήθηκε το προσομοιωμένο μοντέλο του αυτοκινήτου, καθώς και τα απαραίτητα προγράμματα του περιβάλλοντος GYM CarlaEnv και των αρχικών προγραμμάτων για την εκπαίδευση και μετέπειτα χρήση των αλγορίθμων. Στα πλαίσια της παρούσας εργασίας πραγματοποιήθηκαν όλες οι αναγκαίες τροποποιήσεις και προσαρμογές που ήταν αναγκαίες για το σύνολο των πειραμάτων και της μελέτης που πραγματοποιήθηκε.

Το όχημα EcoCar που χρησιμοποιήθηκε αποτελεί ένα αμιγώς ηλεκτρικό διθέσιο αυτοκίνητο πόλης, το οποίο βρίσκεται στο Εργαστήριο Ρομποτική & Ευφυών Συστημάτων του Πολυτεχνείου Κρήτης, κατάλληλο για πειραματισμούς λόγω της σχεδίασής του. Οι προσομοιώσεις εκτελέστηκαν στο λογισμικό CARLA, μία υπερρεαλιστική μηχανή προσομοίωσης ανοιχτού κώδικα η οποία δίνει τη δυνατότητα για ανάπτυξη εφαρμογών αυτόνομης οδήγησης αλλά και αξιόπιστων δεδομένων βασισμένη στην Unreal Engine 4. Οι αλγόριθμοι που εξετάστηκαν είναι ο Soft Actor Critic (SAC) και ο Twin Delayed 3 (TD3) σε δύο μοντέλα ο καθένας με διαφορετικό ρυθμό μάθηση. Τα τέσσερα μοντέλα εκπαιδεύτηκαν με τη βοήθεια του GYM API και της Stable Baselines3, ενώ στη συνέχεια αξιολογήθηκαν τα αποτελέσματά τους βάσει κριτηρίων όπως η πορεία εν συγκρίσει με την διαδρομή που υπολόγισε ο A^* και το κατά πόσο ήταν ομαλή. Εν τέλει ο πιο αποτελεσματικός αλγόριθμος αποδείχτηκε ο TD3 με ρυθμό μάθησης 0,001, ο οποίος εκτέλεσε την πλοήγηση σχεδόν με μηδενική απόκλιση.

Η εργασία αποτελεί ένα ακόμη βήμα προς την δημιουργία ενός πλήρους ψηφιακού διδύμου για το φυσικό μοντέλο του EcoCar. Μελλοντικές ενέργειες είναι περαιτέρω εμβάθυνση στην ενισχυτική μάθηση και τους παρεμφερείς τομείς όπως η βελτιστοποίηση των υπερπαραμέτρων σε κάθε περίπτωση. Αυτό μπορεί να περιλαμβάνει την εφαρμογή και άλλων υπαρχόντων αλγορίθμων όπως ο PPO αλλά και ο συνδυασμός δύο διαφορετικών μεταξύ τους. Επίσης όπως αποδείχτηκε μία μικρή μεταβολή σε κάποια παράμετρο αποφέρει τελείως διαφορετικά αποτελέσματα, έτσι θα ήταν ωφέλιμη η έρευνα πάνω σε αλγορίθμους βελτιστοποίησης παραμέτρων με στόχο την εύρεση των βέλτιστων για κάθε εφαρμογή. Επιπλέον είναι επιθυμητή η πλήρης εκμετάλλευση αισθητήρων όπως το LiDAR και η κάμερα ώστε να μπορεί να υιοθετηθεί σύστημα αποφυγής εμποδίων. Ακόμα χρήσιμη θα ήταν και η εφαρμογή του συστήματος ROS για τη διαχείριση των υποσυστημάτων του οχήματος ως ρομπότ, καθώς θα βελτιώσει την επικοινωνία μεταξύ των συστημάτων αλλά θα καταστήσει εύκολη και την

παράλληλη μελέτη διαφόρων τομέων όπως η ρύθμιση των αισθητήρων και ο έλεγχος των ενεργοποιητών. Τέλος μακροπρόθεσμο στόχο αποτελεί η πλήρης ενσωμάτωση όλων αυτών των ψηφιακών μέσων στο φυσικό όχημα και η δημιουργία ενός λειτουργικού ψηφιακού διδύμου.

Βιβλιογραφία

[1] SAE, «Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,» www.sae.org. [Ηλεκτρονικό]. Available: https://www.sae.org/standards/content/j3016_202104/. [Accessed: Jul. 10, 2024].

[2] H. Vdovic, J. Babic και V. Podobnik, «Automotive Software in Connected and Autonomous Electric Vehicles: A Review,» *IEEE Access,* pp. 166365-166379, 2019.

[3] R. Memon, K. Arezoo, L. Ghamari και K. Alipour, «Autonomous Driving Systems: An Overview of Challenges in Safety, Reliability and Privacy,» *2022 15th International Conference on Human System Interaction (HSI),* 2022.

[4] M. A. Assaad, R. Talj και A. Charara, «2018 13th Annual Conference on System of Systems Engineering (SoSE),» *A system of systems framework: Cooperative Maneuvers Manager for Autonomous Vehicles,* 2018.

[5] N. Sarantinoudis, G. Tsinarakis, L. Doitsidis, N. Tsourveloudis και G. Arampatzis, «Bibliometric Analysis on Applications of Digital Twins in Autonomous Vehicles,» *2023 31st Mediterranean Conference on Control and Automation (MED),* pp. 95-100, 2023.

[6] R. Stark and T. Damerau, "Digital Twin," in *Advances in Production Technology*, vol. 1, 2019, pp. [missing]. doi: 10.1007/978-3-642-35950-7_16870-1.

[7] N. Sarantinoudis, G. Tsinarakis, L. Doitsidis, S. Chatzichristofis και G. Arampatzis, «A ROS-Based Autonomous Vehicle Testbed for the Internet of Vehicles,» *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT),* pp. 726-733, 2023.

[8] Α. Θεοχάρους, «*Αυτόνομη Πλοήγηση Ηλεκτροκίνητου Αυτοκινήτου Πόλης,*», Διπλωματική Εργασία, Σχολή Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης, Χανιά, Ελλάδα, 2023.

[9] Α. Γεωργούλη, Τεχνητή νοημοσύνη [Προπτυχιακό Ενχειρίδιο], Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις, 2015.

[10] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Mark.*, vol. 31, no. 3, pp. 685-695, Sep. 2021. doi: 10.1007/s12525-021-00475-2. [Online]. Available: https://doi.org/10.1007/s12525-021-00475-2.

[11] V. Zhou, «Machine Learning for Beginners: An Introduction to Neural Networks,» victorzhou.com. [Ηλεκτρονικό]. Available: https://victorzhou.com/blog/intro-to-neural-networks/. [Accessed: Jul. 10, 2024].

[12] IBM, «What is a Neural Network?,» www.ibm.com. [Ηλεκτρονικό]. Available: https://www.ibm.com/topics/neural-networks. [Accessed: Jul. 10, 2024].

[13] A. K. Shakya, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," *Expert Syst. Appl.*, vol. 231, p. 120495, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423009971. doi: 10.1016/j.eswa.2023.120495.

[14] R. S. Sutton και A. G. Barto, "Reinforcement Learning: An Introduction," 2nd ed. Cambridge, Massachusetts London, England. The MIT Press, 2018.

[15] A. Joshua, «Spinning Up in Deep Reinforcement Learning,» 2018.

[16] H. Dong, Z. Ding, and S. Zhang, *Deep Reinforcement Learning Fundamentals, Research and Applications*, Singapore: Springer, 2020. doi: 10.1007/978-981-15-4095-0.

[17] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan και D. Hassabis, «Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm,» 2017.

[18] M. Kiran και M. Ozyildimir, «HYPERPARAMETER TUNING FOR DEEP REINFORCEMENT LEARNING APPLICATIONS *,» 26, Jan, 2022. [Ηλεκτορνικό]. Availiable: https://doi.org/10.48550/arXiv.2201.11182 [Accessed: Jul. 10, 2024]

[19] F. Felten, D. Gareev, E.-G. Talbi και G. Danoy, «Hyperparameter Optimization for Multi-Objective Reinforcement Learning,» 2023. [Ηλεκτορνικό]. Availiable: https://arxiv.org/abs/2310.16487 [Accessed: Jul. 10, 2024]

[20] T. Eimer, M. Lindauer, and R. Raileanu, "Hyperparameters in Reinforcement Learning and How To Tune Them," arXiv, vol. 2306.01324, 2023. [Online]. Available: https://arxiv.org/abs/2306.01324. [Accessed: Jul. 19, 2024].

[21] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv, vol. 1509.02971, 2019. [Online]. Available: https://arxiv.org/abs/1509.02971. [Accessed: Jul. 19, 2024].

[22] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," in Proc. 31st Int. Conf. Mach. Learn., Beijing, China, Jun. 2014, vol. 32, no. 1, pp. 387-395. [Online]. Available: http://proceedings.mlr.press/v32/silver14.pdf. [Accessed: Jul. 19, 2024].

[23] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," arXiv, vol. 1802.09477, 2018. [Online]. Available: https://arxiv.org/abs/1802.09477. [Accessed: Jul. 19, 2024].

[24] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv, vol. 1509.02971, 2019. [Online]. Available: https://arxiv.org/abs/1509.02971. [Accessed: Jul. 19, 2024].

[25] K. Moriwaki, "Mathematical modeling and control of an autonomous electric vehicle for navigation and guidance," 2012 IEEE International Electric Vehicle Conference, Greenville, SC, USA, 2012, pp. 1-8, doi: 10.1109/IEVC.2012.6183182.».

[26] I. of Electrical και E. Engineers, Proceedings of 2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM) : Algiers, Algeria, October 29-31, 2018.

[27] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Sallab, S. Yogamani και P. Perez, «Deep Reinforcement Learning for Autonomous Driving: A Survey,» *IEEE Transactions on Intelligent Transportation Systems*, τόμ. 23, αρ. 6, pp. 4909-4926, 6 2022.

[28] I. M. Tabakis και M. Dasygenis, «Deep Reinforcement Learning-Based Path Planning for Dynamic and Heterogeneous Environments,» *2024 Panhellenic Conference on Electronics and Telecommunications, PACET 2024 - Proceedings*, 2024.

[29] Y. Li, Y. Chen, T. Li, J. Lao και X. Li, «DDPG-Based Path Planning Approach for Autonomous Driving,» *Proceedings of 2023 IEEE 12th Data Driven Control and Learning Systems Conference, DDCLS 2023*, pp. 1306-1311, 2023.

[30] I. of Electrical και E. Engineers, 2021 21st International Conference on Control, Automation and Systems (ICCAS) : 12-15 Oct. 2021..

[31] J. Jia, X. Xing and D. E. Chang, "GRU-Attention based TD3 Network for Mobile Robot Navigation," 2022 22nd International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, Republic of, 2022, pp. 1642-1647, doi: 10.23919/ICCAS55662.2022.10003950.

[32] M. Khosyi'In, E. N. Budisusila, S. A. Dwi Prasetyowati, B. Y. Suprapto και Z. Nawawi, «Design of Autonomous Vehicle Navigation Using GNSS Based on Pixhawk 2.1,» *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Τόμ. %1 από %22021-October, pp. 175-180, 2021.

[33] G. Basile, S. Leccese, A. Petrillo, R. Rizzo και S. Santini, «Sustainable DDPG-based Path Tracking For Connected Autonomous Electric Vehicles in extra-urban scenarios,» *2023 IEEE IAS Global Conference on Renewable Energy and Hydrogen Technologies (GlobConHT)*, pp. 1-7, 2023.

[34] B. D. Evans, H. W. Jordaan και H. A. Engelbrecht, «Comparing deep reinforcement learning architectures for autonomous racing,» *Machine Learning with Applications*, τόμ. 14, p. 100496, 2023.

[35] [1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," arXiv, vol. 1711.03938, 2017. [Online]. Available: https://arxiv.org/abs/1711.03938. [Accessed: Jul. 19, 2024].

[36] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in Proc. 1st Annu. Conf. Robot Learn., 2017, pp. 1-16.

[37] ASAM, «ASAM OpenDRIVE®,» www.asam.net. [Ηλεκτρονικό]. Available: https://www.asam.net/standards/detail/opendrive/. [Accessed: Jul. 19, 2024].

[38] R. Zhang, K. Li, F. Yu, Z. He και Z. Yu, «Novel electronic braking system design for EVS based on constrained nonlinear hierarchical control,» *International Journal of Automotive Technology*, pp. 707-718, 08 2017.

[39] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," arXiv, vol. 1606.01540, 2016. [Online]. Available: https://arxiv.org/abs/1606.01540. [Accessed: Jul. 19, 2024].

[40] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. de Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis, "Gymnasium," Zenodo, Mar. 2023. [Online]. Available: https://zenodo.org/record/8127025. [Accessed: Jul. 19, 2024]. doi: 10.5281/zenodo.8127026.

[41] P. E. Hart, N. J. Nilsson και B. Raphael, «A Formal Basis for the Heuristic Determination of Minimum Cost Paths,» *IEEE Transactions on Systems Science and Cybernetics*, αρ. 4, pp. 100-107, 1968.

[42] Driverless Staff, «How Can Autonomous Vehicles Navigate When GNSS Isn’t Working? What You Need to Know about Dead Reckoning for Autonomous Vehicle Navigation | Driverless Report,» www.driverlessreport.com. [Ηλεκτρονικό]. Available: https://www.driverlessreport.com/how-can-autonomous-vehicles-navigate-when-gnss-isnt-working-what-you-need-to-know-about-dead-reckoning-for-autonomous-vehicle-navigation. [Accessed: Jul. 19, 2024].

[43] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," J. Mach. Learn. Res., vol. 22, no. 268, pp. 1-8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html. [Accessed: Jul. 19, 2024].