



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΕΥΦΥΟΥΣ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ
ΚΥΚΛΟΦΟΡΙΑΚΩΝ ΔΕΔΟΜΕΝΩΝ ΜΕ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΘΕΟΔΩΡΟΣ Ν. ΑΛΕΞΑΚΗΣ
ΝΙΚΟΛΑΟΣ Γ. ΠΕΠΠΕΣ

ΕΠΙΒΛΕΠΩΝ:
Μ. Ε. ΘΕΟΛΟΓΟΥ
ΟΜΟΤΙΜΟΣ ΚΑΘΗΓΗΤΗΣ Ε.Μ.Π.

ΙΟΥΝΙΟΣ 2019

Copyright © Θεόδωρος Ν. Αλεξάκης, Νικόλαος Γ. Πεππές, 2019.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τους συγγραφείς και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η ολοένα και μεγαλύτερη απαίτηση για μετακινήσεις καθώς και η μαζική συσσώρευση πληθυσμού γύρω από τα αστικά κέντρα δημιουργεί την ανάγκη για ανάπτυξη υποδομών και συστημάτων για την ορθή και αποτελεσματική διαχείριση του κυκλοφοριακού φόρτου σε αυτά. Επιπλέον, δεδομένης της ραγδαίας τεχνολογικής εξέλιξης στον τομέα της πληροφορίας και της πληροφορικής, καθίσταται πλέον δυνατό να αναπτυχθούν ευφυή συστήματα διαχείρισης κυκλοφοριακών δεδομένων, τα οποία ξεφεύγουν από τις παραδοσιακές προσεγγίσεις.

Οι σύγχρονοι αυτοκινητόδρομοι φέρουν εγκατεστημένους αισθητήρες σε όλη την έκτασή τους οι οποίοι συλλέγουν δεδομένα όπως η ταχύτητα, η πυκνότητα, η κατεύθυνση, κ.α. Επιπλέον, η θεαματική εξέλιξη των οχημάτων και των υπολογιστικών συστημάτων που έχουν εγκατεστημένα προσφέρουν πολύ μεγάλο πλήθος δεδομένων από πολύ απλά όπως η ταχύτητα κίνησης, η επιτάχυνση, κ.λπ. έως πολύ σύνθετα όπως η κατάσταση του οδηγού και η οδηγική του συμπεριφορά. Ωστόσο, όλα αυτά τα δεδομένα από μόνα τους και χωρίς επεξεργασία δεν είναι ικανά να επιλύσουν τα προβλήματα κυκλοφοριακής συμφόρησης. Ως εκ τούτου, απαιτείται η ανάπτυξη σύνθετων υπολογιστικών μεθόδων και αλγορίθμων οι οποίοι είναι ικανοί να διαχειριστούν όλα αυτά τα δεδομένα με γρήγορο και αξιόπιστο τρόπο. Η μέχρι τώρα αλγοριθμική προσέγγιση αφορούσε σε αλγόριθμους ελέγχου με υψηλή υπολογιστική πολυπλοκότητα και μεγάλες υπολογιστικές απαιτήσεις καθώς για να εξετασθεί αποδοτικά και με ταχύτητα ένα αρκετά μεγάλο πλήθος δεδομένων απαιτείτο τεράστια υπολογιστική ισχύ και επομένως και τεράστιο κόστος.

Η θεωρία για τη μελέτη και τη μοντελοποίηση της κίνησης σε αυτοκινητόδρομους σε συνδυασμό με τη χρήση έξυπνων συστημάτων όπως αισθητήρες, δίκτυα επικοινωνιών κ.α., αποτελούν τα βασικά εργαλεία για την ανάπτυξη ενός σύγχρονου και ευφυούς συστήματος ελέγχου διαχείρισης της κυκλοφορίας. Τα σύγχρονα ευφυή συστήματα διαχείρισης κυκλοφοριακών δεδομένων είναι ένας ενεργός τομέας έρευνας καθώς στην κατεύθυνση αυτή τείνουν να προσαρμοστούν αλγόριθμοι μηχανικής μάθησης και τεχνητής νοημοσύνης.

Σκοπός της παρούσας εργασίας αποτέλεσε η μελέτη, ο σχεδιασμός και η ανάπτυξη αλγορίθμων ελέγχου εισόδου οχημάτων σε αυτοκινητόδρομο με αποδοτικό τρόπο ώστε να αποφεύγονται φαινόμενα συμφόρησης. Η βασική μεθοδολογία που εξετάστηκε είναι αυτή του Ramp Metering η οποία είναι ιδιαίτερα αποδοτική και διαδεδομένη στον συγκεκριμένο τομέα. Έτσι, αξιοποιώντας την υπάρχουσα έρευνα αναπτύχθηκαν κάποιοι βασικοί αλγόριθμοι ελέγχου οι οποίοι αποτέλεσαν τους πυλώνες για την ανάπτυξη ενός σύγχρονου αλγορίθμου μηχανικής μάθησης ο οποίος αξιοποιεί δεδομένα που παρήχθησαν από αυτούς. Η ανάπτυξη όλων των αλγορίθμων έγινε στη γλώσσα προγραμματισμού Python με τη χρήση σύγχρονων βιβλιοθηκών όπως η Pandas, η Tensorflow, η Keras, κ.α.

Λέξεις κλειδιά: Ramp Metering, Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Python, Κυκλοφοριακά Δεδομένα.

Abstract

The ever-increasing demand for transportation as well as the massive accumulation of population in urban centers arise the need for infrastructure and system development in order to manage efficiently the traffic flow. Moreover, given the rapid growth and the evolution of the information and technology area, it is now possible to develop intelligent traffic management systems that go beyond traditional approaches.

Nowadays, the motorways have sensors installed across their range that collect data such as speed, density, direction, and so on. In addition, the rapid evolution of vehicles and computer systems that have been installed on them, offer a very large amount of data from very simple ones such as speed, acceleration, etc. to very complex like the driver's situation and driving behavior. However, these data alone and without any procession, congestion problems cannot be solved. Therefore, the development of complex computational methods and algorithms offers the chance to process these data in a fast and reliable way. The existing algorithmic approach till now was the control model which has high computational complexity and requirements in order to be able to examine efficiently and instantly a large amount of data and therefore demands enormous computational power and cost.

The theory behind the modeling of a motorway's traffic flow combined with the use of intelligent systems such as sensors, communication networks, etc. are the basic tools for developing a modern and intelligent traffic management control system. The intelligent traffic management systems are a dynamic research area, as they tend to adapt machine learning and artificial intelligence algorithms.

The purpose of this thesis is to study, design and develop traffic control algorithms on motorways in an efficient way to avoid congestion. The basic methodology examined, is that of Ramp Metering which is a highly efficient and widespread method. Thus, using existing research, some basic control algorithms have been developed which have been the pillars for developing a modern machine learning algorithm that takes advantage of the generated data. All algorithms have been developed in Python programming language using modern libraries such as Pandas, Tensorflow, Keras, and others.

Key words: Ramp Metering, Artificial Intelligence (AI), Machine Learning, Python, Traffic Data.

Ευχαριστίες

Ευχαριστώ πολύ τον Δρ. Δεμέστιχα Κωνσταντίνο, επιβλέποντα της μεταπτυχιακής εργασίας για την εμπιστοσύνη που μας έδειξε, δίνοντάς μας την ευκαιρία να συνεργασθούμε μαζί του καθώς και για τη βοήθεια και την καθοδήγησή του. Επίσης, θα ήθελα να ευχαριστήσω τον Δρ. Ξυνογαλά Σταύρο για την πολύτιμη βοήθεια και την καθοδήγηση που μας παρείχε καθόλη τη διάρκεια εκπόνησης της παρούσας εργασίας.

Θα ήθελα να εκφράσω τις θερμές ευχαριστίες μου στον φίλο, συνάδελφο και συνεργάτη Πεπέ Νικόλαο για την άποψη συνεργασία που είχαμε κατά τη συγγραφή του παρόντος καθώς και για τη βοήθεια που μου προσέφερε, όποτε αυτό του ζητήθηκε.

Ευχαριστώ ιδιαίτερα τη Σεβαστάκη Μαρία - Αλεξάνδρα, για την επιμέλεια του κειμένου καθώς και τη στήριξή της σε όλη τη διάρκεια των σπουδών μου.

Τέλος, θα ήθελα να εκφράσω τις βαθύτατες ευχαριστίες μου στην οικογένειά μου και στους φίλους μου, οι οποίοι μου προσέφεραν αμέριστη στήριξη σε όλες μου τις προσπάθειες.

Θεόδωρος Ν. Αλεξάκης

Ιούνιος 2019

Η παρούσα μεταπτυχιακή διπλωματική εργασία συντάχθηκε το πρώτο εξάμηνο του 2019 στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, στον Τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής.

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω ιδιαίτερα τον Δρ. Κωνσταντίνο Δεμέστιχα, επιβλέποντα της μεταπτυχιακής εργασίας, υπεύθυνο και μεταδιδακτορικό ερευνητή του εργαστηρίου, για την ανάθεση αυτής κι ενασχόλησης με το συγκεκριμένο θέμα, τη δυνατότητα που μας έδωσε να συνεργαστούμε μαζί του, τη βοήθεια και τη καθοδήγηση του.

Επίσης, θα ήθελα να εκφράσω τις ευχαριστίες μου στον Δρ. Σταύρο Ξυνογαλά, για τη βοήθεια και καθοδήγηση, που μας παρείχε καθόλη τη διάρκεια εκπόνησης της συγκεκριμένης εργασίας, καθώς και την υπομονή που επέδειξε σε διάφορα κρίσιμα σημεία υλοποίησης αυτής.

Θα ήθελα να εκφράσω τις θερμές ευχαριστίες μου στον φίλο, συνάδελφο και συνεργάτη Αλεξάκη Θεόδωρο για την άποψη συνεργασία που είχαμε κατά την υλοποίηση και τη συγγραφή του παρόντος, καθώς και τη συνολική βοήθεια και στήριξη που μου παρείχε καθόλη τη διάρκεια του ΔΠΜΣ Τεχνοοικονομικών Συστημάτων.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, για την ηθική, ψυχική και υλική συμπαράσταση, όλα αυτά τα χρόνια των σπουδών μου και τη βοήθεια περάτωσης αυτών.

Νικόλαος Γ. Πεπέ

Ιούνιος 2019

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΚΟΝΕΣ.....	13
ΔΙΑΓΡΑΜΜΑΤΑ	15
ΚΕΦΑΛΑΙΟ 1. ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΚΥΚΛΟΦΟΡΙΑΚΩΝ ΔΕΔΟΜΕΝΩΝ	19
1.1. Εισαγωγή.....	19
1.2. Έξυπνα Οχήματα (Intelligent Vehicles – IVs).....	20
1.3. Αυτοματοποιημένα Συστήματα Αυτοκινητόδρομων (Automated Highway Systems- AHS).....	21
1.4. Τεχνικές και Μεθοδολογίες Ελέγχου Κυκλοφοριακών Δεδομένων.....	23
1.4.1. Είσοδος μέσω ανισόπεδου κόμβου (Ramp Metering).....	23
1.4.2. Βέλτιστος έλεγχος και μοντέλα πρόβλεψης (Optimal control and model predictive control)	24
1.4.2.1. Μοντέλα πρόβλεψης (Model Predictive Control -MPC)	25
1.4.2.2. Εφαρμογή βέλτιστου ελέγχου και MPC στο Ramp Metering	26
1.4.3. Τεχνητή νοημοσύνη (Artificial Intelligence)	27
1.4.3.1. Εφαρμογή της τεχνητής νοημοσύνης (AI) στο Ramp Metering	29
1.4.4. Στατικός έλεγχος με ανατροφοδότηση (Static feedback control)	29
1.4.4.1. Εφαρμογή στατικού ελέγχου με ανατροφοδότηση στη μέτρηση κεκλιμένου επιπέδου	30
1.4.5. Σύγκριση μεθόδων.....	31
1.4.5.1. Υπολογιστική πολυπλοκότητα	32
1.4.5.2. Συμπερίληψη μελλοντικών δεδομένων/εισόδων	32
1.4.5.3. Βασισμένο σε μοντέλο ή όχι	32
1.4.5.4. Συμπερίληψη αυστηρών περιορισμών	33
1.4.5.5. Επεκτασιμότητα	33
ΚΕΦΑΛΑΙΟ 2. ΠΕΡΙΓΡΑΦΗ, ΟΡΙΣΜΟΣ ΚΑΙ ΑΝΑΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	35
2.1. Γενικά.....	35
2.2. Περιγραφή Λειτουργίας της Ελεγχόμενης Πρόσβασης	36

2.3.	Μέθοδοι Ελεγχόμενης Πρόσβασης	38
2.4.	Ορισμός του Προβλήματος	39
2.4.1.	Πιθανά προβλήματα	40
2.4.2.	Μεταβλητές κατάστασης της μεθόδου	42
2.4.3.	Ενέργειες του συστήματος	44
2.4.4.	Συνάρτηση ανταμοιβής (Reward Function) – Στόχος	45
2.5.	Μαθηματική Μοντελοποίηση Αυτοκινητόδρομου	45
2.5.1.	Μοντέλο κίνησης αυτοκινητόδρομου	45
2.5.2.	Στόχος του μοντέλου Ramp Metering	48
2.6.	Νευρωνικά Δίκτυα και Ramp Metering	48
2.6.1.	Γενικά στοιχεία για τα Νευρωνικά Δίκτυα (Neural Networks – NN)...	49
2.6.1.1.	Βιολογικό μοντέλο και έμπνευση	49
2.6.1.2.	Εκπαίδευση Νευρωνικών Δικτύων	50
2.6.1.3.	Perceptrons	52
2.6.1.4.	Ιδιότητες	53
2.6.1.5.	Εφαρμογές	54
2.6.2.	Reinforcement Learning	56
2.6.3.	Νευρωνικό Δίκτυο εφαρμογής Ramp Metering	57
2.7.	Παραδοχές για την Υλοποίηση του Εξεταζόμενου Προβλήματος	59
ΚΕΦΑΛΑΙΟ 3. ΠΡΟΣΣΕΓΙΣΗ ΚΑΙ ΜΕΘΟΔΟΙ ΥΛΟΠΟΙΗΣΗΣ		63
3.1.	Εισαγωγή και Γενικά Στοιχεία για την Python	63
3.2.	Προγραμματιστική Προσέγγιση	63
3.3.	Πιλοτικό Σενάριο Υλοποίησης	65
3.4.	1 ^ο Σενάριο Υλοποίησης – 1 Λωρίδα & 1 ράμπα	68
3.5.	2 ^ο Σενάριο Υλοποίησης – 1 Λωρίδα & 1 ράμπα	71
3.6.	3 ^ο Σενάριο Υλοποίησης – 3 Λωρίδες & 1 ράμπα	75
3.7.	4 ^ο Σενάριο Υλοποίησης – 3 Λωρίδες & 2 ράμπες	79
3.8.	Αλγόριθμος Νευρωνικού Δικτύου	84

ΚΕΦΑΛΑΙΟ 4. ΑΠΟΤΕΛΕΣΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ	-ΣΧΟΛΙΑΣΜΟΣ	ΜΕΘΟΔΩΝ
89		
4.1. Γενικά		89
4.2. Πύλοτικό Σενάριο Υλοποίησης – Αποτελέσματα		90
4.2.1. Πυκνότητα (Density)		91
4.2.2. Μέση ταχύτητα (Mean speed)		92
4.2.3. Πυκνότητα και μέση ταχύτητα		93
4.3. 1 ^ο Σενάριο Υλοποίησης – 1 Λωρίδα & 1 ράμπα		95
4.3.1. Πυκνότητα (Density)		95
4.3.2. Μέση ταχύτητα (Mean speed)		96
4.3.3. Πυκνότητα και μέση ταχύτητα		97
4.4. 2 ^ο Σενάριο Υλοποίησης – 1 Λωρίδα & 1 ράμπα		99
4.4.1. Πυκνότητα (Density)		99
4.4.2. Μέση ταχύτητα (Mean speed)		100
4.4.3. Πυκνότητα και μέση ταχύτητα		101
4.5. 3 ^ο Σενάριο Υλοποίησης – 3 Λωρίδες & 1 ράμπα		103
4.5.1. Πυκνότητα (Density)		103
4.5.2. Μέση ταχύτητα (Mean speed)		104
4.5.3. Πυκνότητα και μέση ταχύτητα		105
4.6. 4 ^ο Σενάριο Υλοποίησης – 3 Λωρίδες & 2 Ράμπες – Εξαγωγή Δεδομένων Νευρωνικού Δικτύου (Data Mining)		107
4.6.1. Πυκνότητα (Density)		108
4.6.2. Μέση ταχύτητα (Mean speed)		110
4.6.3. Πυκνότητα και μέση ταχύτητα		112
4.7. Νευρωνικό Δίκτυο – Αποτελέσματα		117
4.7.1. Ράμπα 1 – Αποτελέσματα		118
4.7.2. Ράμπα 2 – Αποτελέσματα		122
4.7.3. Γενικά αποτελέσματα		125
ΚΕΦΑΛΑΙΟ 5. ΓΕΝΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΠΕΚΤΑΣΕΙΣ	ΚΑΙ	ΜΕΛΛΟΝΤΙΚΕΣ
127		

5.1. Σύγκριση Αλγορίθμων και Συμπεράσματα	127
5.2. Μελλοντικές Επεκτάσεις	128
5.3. Επίλογος	129
ΒΙΒΛΙΟΓΡΑΦΙΑ	131
ΠΑΡΑΡΤΗΜΑ Α	135
Δομή Python και Παραδείγματα	135
Ονοματοδοσία μεταβλητών	135
Τύποι δεδομένων	135
Ενσωματωμένες δομές δεδομένων	136
Ανάθεση τιμών (assignment)	137
Εσοχές (indentation)	138
Τελεστές	138
Εμβέλεια ονομάτων οντοτήτων (scope)	139
Διαδικαστικός προγραμματισμός	139
Αντικειμενοστραφής προγραμματισμός	140
ΠΑΡΑΡΤΗΜΑ Β	141
Πηγαίος Κώδικας σε Python – Πιλοτικό Σενάριο Υλοποίησης	141
Πηγαίος Κώδικας σε Python – 1 ^ο Σενάριο Υλοποίησης	142
Πηγαίος Κώδικας σε Python – 2 ^ο Σενάριο Υλοποίησης	143
Πηγαίος Κώδικας σε Python – 3 ^ο Σενάριο Υλοποίησης	145
Πηγαίος Κώδικας σε Python – 4 ^ο Σενάριο Υλοποίησης	148

ΕΙΚΟΝΕΣ

Εικόνα 1. Έξυπνο όχημα με αισθητήρες ^[1]	21
Εικόνα 2. Ramp Metering	23
Εικόνα 3. Δομή βέλτιστου ελέγχου	26
Εικόνα 4. Στρατηγική ζήτησης - χωρητικότητας	30
Εικόνα 5. ALINEA.....	31
Εικόνα 6. Ελεγχόμενη πρόσβαση μέσω ράμπας. Πηγή: FHWA 2014.....	36
Εικόνα 7. Σύγκριση ράμπας με και χωρίς ελεγχόμενη πρόσβαση. Πηγή: FHWA 2014	37
Εικόνα 8. Ράμπα εισόδου με ελεγχόμενη πρόσβαση	39
Εικόνα 9. Πρόβλημα 1 - Συμφόρηση στο κύριο ρεύμα	41
Εικόνα 10. Πρόβλημα 2 - Μεγάλη ουρά στη ράμπα.....	41
Εικόνα 11. Μεταβλητές του συστήματος.....	43
Εικόνα 12. Απεικόνιση με δύο ράμπες.....	44
Εικόνα 13. Διάγραμμα συγκέντρωσης-ροής αυτοκινήτων σε αυτοκινητόδρομο ^[30] ..	47
Εικόνα 14. Δομή τυπικού νευρωνικού δικτύου.....	51
Εικόνα 15. Αντιπαραβολή βιολογικού με τεχνητού νευρώνα.....	52
Εικόνα 16. Μοντέλο Thorsten Schmidt-Dumont και Jan H van Vuuren	56
Εικόνα 17. Διάγραμμα ροής πιλοτικού αλγορίθμου	66
Εικόνα 18. Σχηματική απεικόνιση εξεταζόμενης διάταξης πιλοτικής υλοποίησης ...	67
Εικόνα 19. Ενδεικτική απεικόνιση διάταξης 1ου σεναρίου	69
Εικόνα 20. Λογικό διάγραμμα αλγορίθμου 1 ^{ου} σεναρίου υλοποίησης.....	70
Εικόνα 21. Ενδεικτική απεικόνιση διάταξης 2ου σεναρίου	73
Εικόνα 22. Λογικό διάγραμμα 2 ^{ου} σεναρίου υλοποίησης με αυξημένη χωρητικότητα buffer	74
Εικόνα 23. Ενδεικτική απεικόνιση διάταξης 3ου σεναρίου με τρεις (3) λωρίδες	77
Εικόνα 24. Λογικό διάγραμμα αλγορίθμου 3 ^{ου} σεναρίου υλοποίησης με τρεις (3) λωρίδες	78
Εικόνα 25. Λογικό διάγραμμα αλγορίθμου 4 ^{ου} σεναρίου υλοποίησης με τρεις (3) λωρίδες και δύο (2) ράμπες	83
Εικόνα 26. Στιγμιότυπο δεδομένων εκπαίδευσης νευρωνικού δικτύου	85
Εικόνα 27. Στρώματα υλοποίησης νευρωνικού δικτύου	86
Εικόνα 28. Ενδεικτική απεικόνιση διάταξης 4 ^{ου} σεναρίου υλοποίησης με δύο (2) ράμπες και τρεις (3) λωρίδες	89
Εικόνα 29. Στιγμιότυπο από Jupyter Notebook	90
Εικόνα 30. Epochs για τη ράμπα 1	118

Εικόνα 31. Ακρίβεια και απώλεια δεδομένων αποτελεσμάτων για τη ράμπα 1	119
Εικόνα 32. Epochs για τη ράμπα 2	122
Εικόνα 33. Ακρίβεια και απώλεια δεδομένων αποτελεσμάτων για τη ράμπα 2	123

ΔΙΑΓΡΑΜΜΑΤΑ

Διάγραμμα 1. Πυκνότητα πιλοτικού αλγορίθμου	91
Διάγραμμα 2. Μέση ταχύτητα πιλοτικού αλγορίθμου	92
Διάγραμμα 3. Πυκνότητα και μέση ταχύτητα πιλοτικού αλγορίθμου	93
Διάγραμμα 4. Λόγος R πιλοτικού αλγορίθμου.....	93
Διάγραμμα 5. Ιστόγραμμα πυκνοτήτων πιλοτικού αλγορίθμου.....	94
Διάγραμμα 6. Πυκνότητα αλγορίθμου 1 ^{ου} σεναρίου.....	95
Διάγραμμα 7. Μέση ταχύτητα αλγορίθμου 1 ^{ου} σεναρίου.....	96
Διάγραμμα 8. Πυκνότητα και μέση ταχύτητα αλγορίθμου 1 ^{ου} σεναρίου.....	97
Διάγραμμα 9. Λόγος R αλγορίθμου 1 ^{ου} σεναρίου	97
Διάγραμμα 10. Ιστόγραμμα πυκνοτήτων αλγορίθμου 1 ^{ου} σεναρίου	98
Διάγραμμα 11. Πυκνότητα 2 ^{ου} σεναρίου υλοποίησης.....	99
Διάγραμμα 12. Μέση ταχύτητα αλγορίθμου 2 ^{ου} σεναρίου υλοποίησης.....	100
Διάγραμμα 13. Πυκνότητα και μέση ταχύτητα αλγορίθμου 2 ^{ου} σεναρίου.....	101
Διάγραμμα 14. Λόγος R αλγορίθμου 2 ^{ου} σεναρίου	101
Διάγραμμα 15. Ιστόγραμμα πυκνοτήτων αλγορίθμου 2 ^{ου} σεναρίου	102
Διάγραμμα 16. Πυκνότητα 3 ^{ου} σεναρίου υλοποίησης.....	103
Διάγραμμα 17. Μέση ταχύτητα αλγορίθμου 3 ^{ου} σεναρίου υλοποίησης.....	104
Διάγραμμα 18. Πυκνότητα και Μέση ταχύτητα αλγορίθμου 3 ^{ου} σεναρίου	105
Διάγραμμα 19. Λόγος R αλγορίθμου 3 ^{ου} σεναρίου	106
Διάγραμμα 20. Ιστόγραμμα πυκνοτήτων αλγορίθμου 3 ^{ου} σεναρίου	107
Διάγραμμα 21. Πυκνότητα 4 ^{ου} σεναρίου υλοποίησης για T=0 έως T=1500.....	108
Διάγραμμα 22. Πυκνότητα 4 ^{ου} σεναρίου υλοποίησης για T=54000 έως T=55500..	109
Διάγραμμα 23. Πυκνότητα 4 ^{ου} σεναρίου υλοποίησης για T=106500 έως T=108000	109
Διάγραμμα 24. Μέση ταχύτητα αλγορίθμου 4 ^{ου} σεναρίου υλοποίησης για T=0 έως T=1500	110
Διάγραμμα 25. Μέση ταχύτητα αλγορίθμου 4 ^{ου} σεναρίου υλοποίησης για T=54000 έως T=55500	111
Διάγραμμα 26. Μέση ταχύτητα αλγορίθμου 4 ^{ου} σεναρίου υλοποίησης για T=106500 έως T=108000	111
Διάγραμμα 27. Πυκνότητα και μέση ταχύτητα αλγορίθμου 4 ^{ου} σεναρίου για T=0 έως T=1000	112
Διάγραμμα 28. Πυκνότητα και μέση ταχύτητα αλγορίθμου 4 ^{ου} σεναρίου για T=54000 έως T=55000	113

Διάγραμμα 29. Πυκνότητα και μέση ταχύτητα αλγορίθμου 4 ^ο σεναρίου για T=107000 έως T=108000	113
Διάγραμμα 30. Λόγος R αλγορίθμου 4 ^ο σεναρίου για T=0 έως T=1500.....	114
Διάγραμμα 31. Λόγος R αλγορίθμου 4 ^ο σεναρίου για T=54000 έως T=55500.....	114
Διάγραμμα 32. Λόγος R αλγορίθμου 4 ^ο σεναρίου για T=106500 έως T=108000..	115
Διάγραμμα 33. Ιστόγραμμα πυκνοτήτων αλγορίθμου 4 ^ο σεναρίου	116
Διάγραμμα 34. Ιστόγραμμα κατανομής χρόνων κόκκινης ένδειξης σηματοδότη 4ου σεναρίου υλοποίησης	117
Διάγραμμα 35. Διάγραμμα overfit για τη ράμπα 1.....	120
Διάγραμμα 36. Διάγραμμα underfit για τη ράμπα 1.....	120
Διάγραμμα 37. Συσχετιστικά διαγράμματα για τη ράμπα 1.....	121
Διάγραμμα 38. Διάγραμμα overfit για τη ράμπα 2.....	123
Διάγραμμα 39. Διάγραμμα underfit για τη ράμπα 2.....	123
Διάγραμμα 40. Συσχετιστικά διαγράμματα για τη ράμπα 2.....	124
Διάγραμμα 41. Σύσχετιση μεγεθών νευρωνικού δικτύου	125

ΚΕΦΑΛΑΙΟ 1. ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΚΥΚΛΟΦΟΡΙΑΚΩΝ ΔΕΔΟΜΕΝΩΝ

1.1. Εισαγωγή

Τα αστικά συστήματα μεταφορών αποτελούν ακρογωνιαίο λίθο για τις οικονομικές δραστηριότητες όλων των σύγχρονων κοινοτήτων και επομένως όλων των πολιτών που ζουν σε αυτές. Ως συστήματα μεταφοράς συνήθως αναφέρονται οι σιδηρόδρομοι, οι αυτοκινητόδρομοι, τα αεροδρόμια, τα λιμάνια, κ.λπ. Το μεγαλύτερο μέρος των συστημάτων μεταφοράς αποτελούν οι αυτοκινητόδρομοι μέσα σε μία πόλη επηρεάζοντας την ταχύτητα των μεταφορών καθώς και το βιοτικό επίπεδο του μεγαλύτερου μέρους του πληθυσμού άμεσα. Ως εκ τούτου το μεγαλύτερο κομμάτι της έρευνας για τη βελτίωση, μέσω έξυπνων συστημάτων, των συνθηκών μεταφοράς έγκειται στην αξιοποίηση δεδομένων και στην ανάπτυξη συστημάτων που αφορούν τους οδικούς άξονες μιας πόλης.

Η συνεχώς αυξανόμενη ζήτηση για κίνηση στους δρόμους οδήγησε στην ανάπτυξη συστημάτων διαχείρισης κίνησης τα οποία όμως δεν έχουν λύσει επαρκώς το πρόβλημα και συνεχώς εξελίσσονται. Αυτό έχει σαν αποτέλεσμα να αυξάνονται οι χρόνοι μετακίνησης όπως επίσης και τα έξοδα μετακίνησης, γεγονός που οδηγεί σε αυξημένα οικονομικά και κοινωνικά κόστη. Η λύση που προτάθηκε σε προηγούμενες δεκαετίες για την αντιμετώπιση του συγκεκριμένου προβλήματος αφορούσε στην κατασκευή περιφερειακών οδών και στην αύξηση της χωρητικότητας των υπάρχουσών οδικών αρτηριών ωστόσο αυτή η προσέγγιση δεν αντιμετώπισε το πρόβλημα. Με αυτό ως δεδομένο η προσέγγιση που θα ακολουθηθεί στη συγκεκριμένη εργασία είναι αυτή της ανάπτυξης ενός έξυπνου συστήματος διαχείρισης κυκλοφοριακών δεδομένων τα οποία θα συλλέγονται από κάποιο τμήμα κεντρικής οδικής αρτηρίας.

Η διαχείριση και ο έλεγχος της κίνησης συμβάλλουν στη μείωση της κυκλοφοριακής συμφόρησης και κατ' επέκταση στη γενικότερη βελτίωση του οδικού δικτύου. Τα κυκλοφοριακά δεδομένα τα οποία αποτελούν αρωγό στη διαχείριση ενός οδικού δικτύου μπορεί να είναι τα εξής:

- Χωρητικότητα οδικών αρτηριών
- Χρόνοι ταξιδιού
- Ασφάλεια
- Κατανάλωση καυσίμων
- Εκπομπές ρύπων
- Ταχύτητα οχημάτων
- Τύπος οχημάτων

- Κατεύθυνση οχημάτων
- κ.α.

Σήμερα η συλλογή των παραπάνω δεδομένων πραγματοποιείται μέσω περιφερειακών αισθητήρων που είναι εγκατεστημένοι σε κομβικά σημεία στις οδικές αρτηρίες. Τα ήδη εγκατεστημένα συστήματα εκτελούν έλεγχο της κυκλοφορίας μέσω δυναμικών πινακίδων σήμανσης, δυναμικό ορίων ταχύτητας κ.α.

Πλέον των υφιστάμενων συστημάτων οι ραγδαίως αναπτυσσόμενες τεχνολογίες στους τομείς των επικοινωνιών, των αισθητήρων καθώς και των πληροφοριακών δεδομένων συνεπικουρούν στην ανάπτυξη νέων συστημάτων διαχείρισης και ελέγχου της κυκλοφορίας. Έτσι, οδηγούμαστε στην ανάπτυξη έξυπνων συστημάτων μεταφορών (Intelligent Transportation Systems –ITS) και έξυπνων συστημάτων οχημάτων και αυτοκινητόδρομων (Intelligent Vehicle Highway Systems – IVHS). Τα συστήματα ITS και IVHS ενσωματώνουν τόσο τις δομές των υφιστάμενων συστημάτων που βρίσκονται συνήθως εγκατεστημένα στον δρόμο όσο και τα δεδομένα τα οποία διαθέτουν τα οχήματα αυτά καθαυτά. Μέσω των συγκεκριμένων δεδομένων και με τη χρήση υπολογιστικών διαδικασιών επιχειρούν να προβλέψουν και να διαχειριστούν πιο αποτελεσματικά την κυκλοφοριακή συμφόρηση, την αντιμετώπιση εκτάκτων καταστάσεων και τις περιβαλλοντικές επιπτώσεις λόγω των ρύπων οι οποίοι στα αστικά κέντρα επιβαρύνουν το βιοτικό επίπεδο των πολιτών.

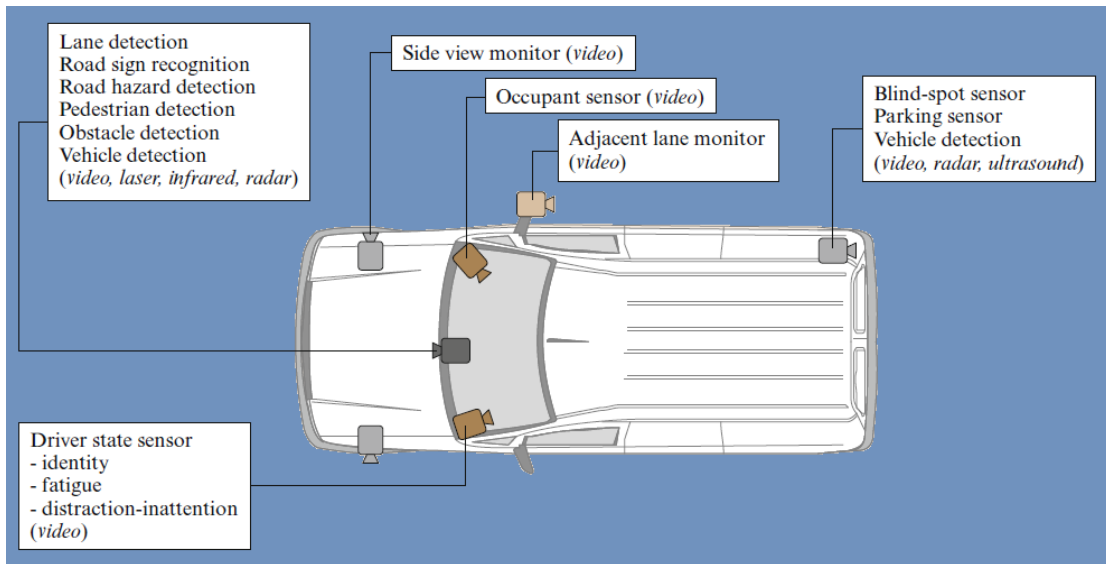
1.2. Έξυπνα Οχήματα (Intelligent Vehicles – IVs)

Τα τελευταία χρόνια όλο και περισσότερο εμφανίζονται οχήματα με μεγάλο αριθμό αισθητήρων και επικοινωνιακών δομών τα οποία παρέχουν πληθώρα πληροφοριών τόσο στον οδηγό αλλά και σε εν δυνάμει διασυνδεδεμένα συστήματα διαχείρισης δεδομένων. Η ανάπτυξη όλο και περισσότερο των παραπάνω συστημάτων οδηγεί στην πλήρως αυτοματοποιημένη οδήγηση κατά την οποία ο οδηγός θα έχει ελάχιστη έως καθόλου εμπλοκή στη διαδικασία.

Ένα έξυπνο όχημα από μόνο του δεν μπορεί να εκτελέσει ορθά σχεδόν καμία διαδικασία εάν δεν έχει αλληλεπίδραση με το περιβάλλον, τον οδηγό αλλά και με συστήματα κυκλοφοριακών δεδομένων ταυτόχρονα. Επομένως κάποιες βασικές απαιτήσεις για την ορθή λειτουργία των έξυπνων οχημάτων είναι οι κάτωθι:

- Η θέση του οχήματος, η οποία θα εντοπίζεται μέσω συστήματος GPS
- Η τρέχουσα κινητική κατάσταση του οχήματος (ταχύτητα, επιτάχυνση, κ.λπ.)
- Το περιβάλλον (δρόμος, οχήματα, κ.λπ.)
- Η κατάσταση του οδηγού, των επιβατών και του φορτίου
- Η επικοινωνία με τα υπόλοιπα συστήματα συλλογής και ανάλυσης κυκλοφοριακών δεδομένων
- Η πρόσβαση σε γεωγραφικά δεδομένα (χάρτες, συντεταγμένες, κ.α.)

Προκειμένου λοιπόν, να χαρακτηριστεί ένα όχημα έξυπνο απαιτούνται πληθώρα αισθητήρων και ένα εξελιγμένο υπολογιστικό σύστημα το οποίο θα συγκεντρώνει όλα τα δεδομένα και θα μπορεί να τα απεικονίσει σε κατανοητή μορφή για τον μέσο οδηγό αλλά ταυτόχρονα και να τα αποστέλλει σε υπερκείμενα συστήματα κυκλοφοριακής διαχείρισης. Παρακάτω φαίνεται η δομή ενός έξυπνου οχήματος για τη διάγνωση του περιβάλλοντα χώρου.



Εικόνα 1. Έξυπνο όχημα με αισθητήρες^[1]

Η κυκλοφορία έξυπνων οχημάτων μπορεί να οδηγήσει βραχυπρόθεσμα και μακροπρόθεσμα σε πολλά οφέλη, οικονομικά, κοινωνικά και περιβαλλοντικά, όπως για παράδειγμα:

- Μείωση χρόνου μετακινήσεων και καυσίμων
- Μείωση εκπεμπόμενων ρύπων
- Μείωση ατυχημάτων
- κ.α.

1.3. Αυτοματοποιημένα Συστήματα Αυτοκινητόδρομων (Automated Highway Systems- AHS)

Η εξέλιξη των ITS και IVHS είναι τα πλήρως αυτοματοποιημένα συστήματα αυτοκινητόδρομων ή AHS. Τα AHS χρησιμοποιούν όλες τις δομές κυκλοφοριακών δεδομένων που βρίσκονται εντός των αυτοκινητοδρόμων καθώς και τα δεδομένα από τα έξυπνα οχήματα. Με όλη αυτή την πληροφορία μπορούν να ελέγξουν πλήρως τη χωρητικότητα του δρόμου και να διαχειριστούν αποδοτικά την κίνηση αλλά και έκτακτες καταστάσεις όπως ατυχήματα, ακινητοποιημένα οχήματα, κ.λπ.

Με βάση τον βαθμό συνεργασίας του αυτοκινητοδρόμου και του οχήματος υπάρχουν διάφοροι τύποι AHS^{[2],[3]}:

- **Αυτόνομα συστήματα οχημάτων:**

Τα οχήματα δεν συνεργάζονται με τον δρόμο και απλά είναι εξοπλισμένα με αισθητήρες και υπολογιστικά συστήματα για τοπικές ενδείξεις.

- **Συστήματα συνεργασίας μεταξύ οχημάτων:**

Τα οχήματα διαθέτουν αισθητήρες και ασύρματες επικοινωνίες προκειμένου να επικοινωνήσουν με γειτονικά οχήματα και να ανταλλάξουν πληροφορίες χωρίς να επικοινωνούν με τον αυτοκινητόδρομο.

- **Συνεργατικά συστήματα οχημάτων και οδικής υποδομής:**

Τα οχήματα επικοινωνούν μεταξύ τους αλλά και με την οδική υποδομή και η λήψη οδηγιών γίνεται από τον συνδυασμό όλων των δεδομένων.

- **Συστήματα διαχείρισης οδικών υποδομών:**

Τα οχήματα αποστέλλουν τις επιθυμητές διαδρομές και ενέργειες (αλλαγή λωρίδας, επιτάχυνση, κ.λπ.) σε ένα κεντρικό σύστημα το οποίο συνεργάζεται με την οδική υποδομή και το οποίο παρέχει οδηγίες για τις ενέργειες και τις κατευθύνσεις που πρέπει να ακολουθήσουν τα οχήματα.

- **Συστήματα πλήρως ελεγχόμενα από το οδικό δίκτυο:**

Όλες οι πληροφορίες από αισθητήρες εντός του δικτύου και από τα οχήματα συλλέγονται και επεξεργάζονται από το οδικό δίκτυο το οποίο έχει τον πλήρη έλεγχο για την κίνηση των οχημάτων χωρίς την παρέμβαση οδηγών αξιοποιώντας έτσι στο βέλτιστο βαθμό τις δυνατότητες της οδικής αρτηρίας.

Κάθε μία από τις παραπάνω κατηγορίες απαιτεί διαφορετικούς υπολογιστικούς πόρους και έχει και διαφορετικό κόστος. Όσο πιο αυτόνομος είναι ο αυτοκινητόδρομος τόσο πιο υψηλό είναι το οικονομικό και χρονικό κόστος υλοποίησης. Επιπλέον, σε τοπικό επίπεδο η διαχείριση μίας τοπικής αρτηρίας είναι εφικτή μέσω υπολογιστικών συστημάτων, ωστόσο όταν μιλάμε για οδικά δίκτυα ολόκληρων χωρών γίνεται εύκολα αντιληπτό ότι απαιτούνται συστήματα υψηλών επεξεργαστικών δυνατοτήτων έτσι ώστε η απόκριση να είναι έγκυρη και έγκαιρη. Γίνεται εύκολα αντιληπτό ότι εάν ο χρόνος απόκρισης τέτοιων συστημάτων δεν είναι σε τάξη δευτερολέπτων αυτό μπορεί να επιφέρει δυσάρεστα αποτελέσματα τόσο για τη διαχείριση του δικτύου όσο και για τους ανθρώπους καθώς μπορεί να οδηγήσει σε ατυχήματα.

1.4. Τεχνικές και Μεθοδολογίες Ελέγχου Κυκλοφοριακών Δεδομένων

Η διαχείριση και ο έλεγχος ενός δικτύου οδικής κυκλοφορίας μπορεί να πραγματοποιηθεί με διάφορες τεχνικές. Μερικές από αυτές που χρησιμοποιούνται στην πράξη είναι οι παρακάτω^{[4],[5],[6]}:

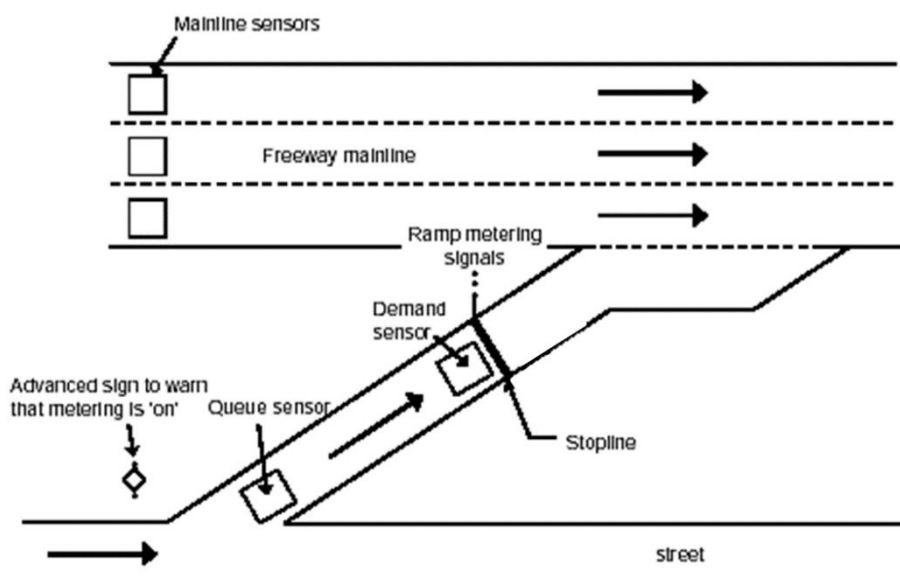
- Βέλτιστος έλεγχος και μοντέλα πρόβλεψης (Optimal control and model predictive control)
- Τεχνητή νοημοσύνη (Artificial Intelligence)
- Στατικός έλεγχος με ανατροφοδότηση (Static feedback control)

Προκειμένου να γίνει πιο σαφής η χρήση των παραπάνω μεθόδων θα εξεταστεί η εφαρμογή τους στην είσοδο μέσω ανισόπεδου κόμβου σε ένα αυτοκινητόδρομο (Ramp Metering).

Οι παραπάνω μέθοδοι μπορούν να χρησιμοποιηθούν και για πληθώρα άλλων εφαρμογών που αφορούν τη διαχείριση της κυκλοφορίας.

1.4.1. Είσοδος μέσω ανισόπεδου κόμβου (Ramp Metering)

Το Ramp Metering είναι μία μέθοδος μέτρησης και ελέγχου της κίνησης η οποία εφαρμόζεται στις ράμπες που βρίσκονται στους αυτοκινητόδρομους. Η μέτρηση των οχημάτων γίνεται μέσω αισθητήρων που βρίσκονται στις εισόδους και εξόδους του αυτοκινητόδρομου. Σκοπός της συγκεκριμένης μεθόδου είναι να ελέγξει την είσοδο οχημάτων στον αυτοκινητόδρομο προκειμένου να αξιοποιήσει με τον καλύτερο δυνατό τρόπο τη χωρητικότητα της οδικής αρτηρίας.^{[7],[8]}



Εικόνα 2. Ramp Metering

Για παράδειγμα, έχουμε έναν αυτοκινητόδρομο με ένα κύριο ρεύμα και μία ράμπα εισόδου όπως φαίνεται στην παραπάνω εικόνα. Η ράμπα έχει μία ζήτηση για κίνηση d_0 που εκφράζεται σε οχήματα ανά ώρα (vh/h), r_0 (vh/h) η κίνηση στη ράμπα που ρυθμίζεται από τις σημάσεις, q_c (vh/h) η χωρητικότητα του αυτοκινητόδρομου και q_{in} (vh/h) η κίνηση του κύριου ρεύματος. Όταν εμφανιστεί κυκλοφοριακή συμφόρηση η μέγιστη ροή εξόδου του κύριου ρεύματος q_{out} (vh/h) είναι μικρότερη από την κανονική ροή του κύριου ρεύματος q_c . Το παραπάνω φαινόμενο ονομάζεται πτώση χωρητικότητας (capacity drop) και στόχος είναι να αποφεύγεται κατά το δυνατόν. Σε περιπτώσεις που δεν εφαρμόζεται το ramp metering η πτώση χωρητικότητας συμβαίνει αρκετά συχνά. Επομένως, με τη χρήση του Ramp Metering και τη συνεχή ανάκτηση και ανάλυση δεδομένων κίνησης το φαινόμενο πτώσης χωρητικότητας αποφεύγεται με αποτέλεσμα η ροή οχημάτων να είναι ομαλή και να μειώνονται οι χρόνοι αναμονής και κυκλοφοριακής συμφόρησης στις κεντρικές αρτηρίες. Από την άλλη υπάρχει πιθανότητα δημιουργίας μεγάλης ουράς οχημάτων πάνω στη ράμπα εισόδου και αντίστοιχα μεγάλες καθυστερήσεις εκεί. Έτσι, γίνεται εύκολα αντιληπτό ότι πρέπει να διατηρηθεί μία ισορροπία μεταξύ της κίνησης στον αυτοκινητόδρομο και στο μήκος της ουράς στην είσοδο.^{[9],[10]}

Οι μέθοδοι ελέγχου που θα αναλυθούν παρακάτω έχουν ως βάση τον διακριτό χρόνο. Αυτό πρακτικά σημαίνει ότι κάθε δείγμα προκύπτει σε χρόνο $t=kT$ όπου T είναι ο χρόνος δειγματοληψίας που θα οριστεί και k ακέραιος αριθμός ο οποίος συμβολίζει το βήμα δειγματοληψίας. Τα δείγματα στη συνέχεια εισάγονται στον ελεγκτή ο οποίος με βάση τις συνθήκες που έχουν οριστεί αποφασίζει εάν θα πρέπει να επιτρέψει την είσοδο οχημάτων στο κύριο ρεύμα και εάν ναι κάθε πότε.

1.4.2. Βέλτιστος έλεγχος και μοντέλα πρόβλεψης (Optimal control and model predictive control)

Ο βέλτιστος έλεγχος καθορίζει μία ακολουθία ενεργειών οι οποίες στην ουσία βελτιστοποιούν την απόδοση μίας διαδικασίας/συνάρτησης λαμβάνοντας υπόψη τις μελλοντικές παραμέτρους και περιορισμούς. Στην περίπτωση διαχείρισης της κυκλοφορίας η ζήτηση για χωρητικότητα αποτελεί τη βασική παράμετρο που λαμβάνεται υπόψη. Γενικότερα τα προβλήματα βελτιστοποίησης διακριτού χρόνου εμπεριέχουν τα κάτωθι:^{[11],[12]}

- Εξισώσεις μοντέλων δυναμικών συστημάτων
- Μία αρχική κατάσταση x_0
- Μία αρχική χρονική στιγμή t_0
- Περιορισμούς
- Μετρήσεις
- Μία μέτρα απόδοσης J

Πιο συγκεκριμένα, έστω ότι έχουμε ένα δυναμικό σύστημα πολλών εισόδων και πολλών εξόδων το οποίο περιγράφεται από την παρακάτω εξίσωση:

$$x(k+1) = f(x(k), u(k), d(k)) \quad (1)$$

Όπου

- $x \in \mathbb{R}^n$
- $u \in \mathbb{R}^m$ ένα διάνυσμα εισόδων m διαστάσεων
- f μία συνεχώς διαφορίσιμη συνάρτηση
- d ο θόρυβος.

Για έναν χρονικό ορίζοντα K , το πρόβλημα βελτιστοποίησης ορίζεται από τα διανύσματα ελέγχου $u(0), u(1), u(2), \dots, u(K-1)$ με τέτοιο τρόπο έτσι ώστε η μήτρα απόδοσης J να έχει τη μικρότερη δυνατή διασπορά σε σχέση με τις αρχικές τιμές λαμβάνοντας υπόψη τους περιορισμούς, τη δυναμική του συστήματος, τις αρχικές συνθήκες, κ.α. Αυτό μαθηματικά εκφράζεται στην εξίσωση που φαίνεται παρακάτω:

$$J = \theta[x(K)] + \sum_{k=0}^{K-1} \varphi[x(k), u(k), d(k)] \quad (2)$$

Με

$$\begin{aligned} x(0) &= x_0 \\ x(k+1) &= f(x(k), u(k), d(k)) && \text{για } k=0, \dots, K-1 \\ u_{\min}(k) &\leq u(k) \leq u_{\max}(k) && \text{για } k=0, \dots, K-1 \\ c(x(k), u(k), k) &\leq 0 && \text{για } k=0, \dots, K-1 \end{aligned}$$

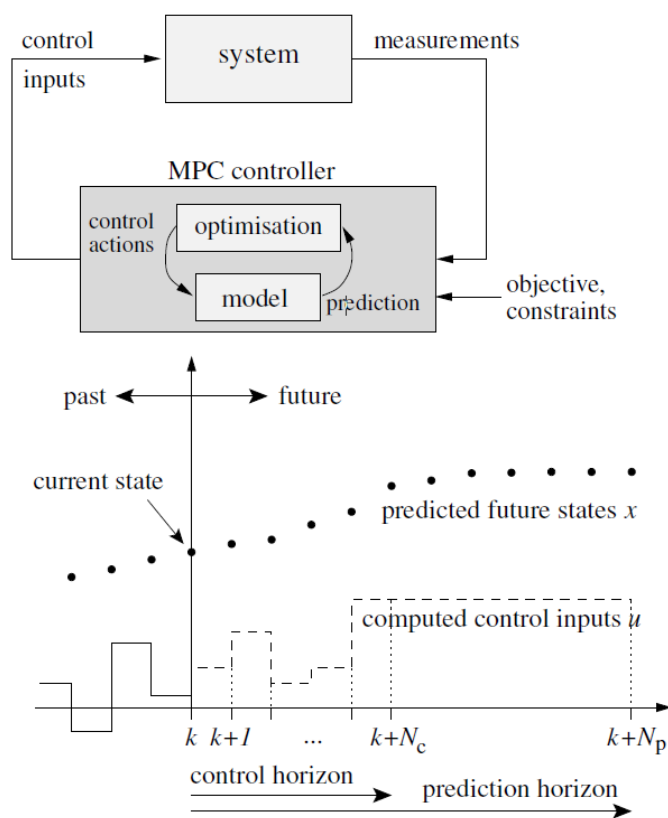
όπου θ η συνάρτηση τερματικού κόστους και φ η Langrangian συνάρτηση. Οι δύο αυτές συναρτήσεις είναι διπλά διαφορίσιμες. Τα u_{\min} και u_{\max} αποτελούν τα όρια των μεταβλητών ελέγχου, το c εκφράζει τους περιορισμούς που υπάρχουν για το διάνυσμα x και d είναι το διάνυσμα θορύβου. Όλα τα παραπάνω είναι γνωστά για διάρκεια $[k, k+KT]$ για $k=0, \dots, K-1$. Έτσι γίνεται αντιληπτό ότι το αποτέλεσμα βασίζεται τόσο στην αρχική κατάσταση όσο και σε όλες τις χρονικές καταστάσεις που είναι διαθέσιμες. Τέτοιου είδους προβλήματα επιλύονται είτε με επίλυση εξισώσεων είτε μέσω δυναμικού προγραμματισμού.

Το βασικό μειονέκτημα των μεθόδων βέλτιστου ελέγχου είναι ότι αποτελούν ένα πρόβλημα ανοικτού βρόχου και έτσι εμπεριέχουν πολλά περιθώρια σφάλματος αλλά και θόρυβο, αλλοιώνοντας έτσι το αποτέλεσμα.

1.4.2.1. Μοντέλα πρόβλεψης (Model Predictive Control -MPC)

Ως εξέλιξη των μοντέλων βέλτιστου ελέγχου αναπτύχθηκαν τα μοντέλα πρόβλεψης (Model Predictive Control -MPC). Τα συγκεκριμένα μοντέλα προέκυψαν από τη βιομηχανία όπου και εδραιώθηκαν καθώς είναι ιδιαίτερα αποτελεσματικά σε πολλές εφαρμογές του συγκεκριμένου κλάδου. Τα MPC χρησιμοποιούν αλγόριθμους ανατροφοδότησης και εκτελούν πολλαπλές φορές τον βέλτιστο έλεγχο με σκοπό την ελαχιστοποίηση του σφάλματος. Η βασική

αρχή λειτουργίας τους είναι ότι μέσω της ανατροφοδότησης ο βέλτιστος έλεγχος μπορεί να εκτελεστεί μέσω προβλέψεων αρκετές φορές. Ωστόσο τα MPC εξακολουθούν να υπόκεινται σε περιορισμούς που συναντώνται στα μοντέλα βέλτιστου ελέγχου. Οι εξωτερικές εισοδοι και οι διαταραχές πρέπει να είναι γνωστές και προσδιορισμένες από την αρχή με όσο το δυνατόν μεγαλύτερη ακρίβεια για όλο τον χρονικό ορίζοντα της πρόβλεψης. Κάτι τέτοιο είναι πάρα πολύ δύσκολο καθώς σε συστήματα με δυναμικό χαρακτήρα τα δεδομένα μπορούν να αλλάξουν ανά πάσα στιγμή καθιστώντας έτσι το μοντέλο πρόβλεψης ανακριβές και δημιουργώντας προβλήματα στην αποτελεσματικότητα διαχείρισης σύνθετων καταστάσεων. Παρακάτω ακολουθεί μία σχηματική απεικόνιση ενός μοντέλου MPC.^{[13],[14]}



Εικόνα 3. Δομή βέλτιστου ελέγχου

1.4.2.2. Εφαρμογή βέλτιστου ελέγχου και MPC στο Ramp Metering

Ένα μακροσκοπικό μοντέλο ελέγχου μπορεί να χρησιμοποιηθεί για να μοντελοποιηθεί ένα σύστημα κυκλοφορίας αυτοκινητοδρόμων. Το μοντέλο αναπαριστά ένα δίκτυο ως κατευθυνόμενο γράφο με συνδέσμους (που αντιστοιχούν σε οδικό δίκτυο με ομοίμορφα χαρακτηριστικά) και κορυφές (που αντιστοιχούν στις ράμπες, διασταυρώσεις κ.λπ.). Στο δίκτυο κάθε σύνδεσμος χωρίζεται σε τμήματα. Το διάνυσμα κατάστασης x για το πρόβλημα ελέγχου συνίσταται σε πυκνότητες $\rho_{m,i}$ και μέσες ταχύτητες $v_{m,i}$ κάθε τμήματος i του εκάστοτε συνδέσμου m και τα μήκη w_o ουράς σε κάθε είσοδο o του δικτύου. Το διάνυσμα διαταραχής (θορύβου) του διανύσματος ελέγχου d περιέχει τις απαιτήσεις d_o στις εισόδους του οδικού

δικτύου. Όλες αυτές οι τιμές λαμβάνονται υπόψη και πρέπει να είναι γνωστές σε ολόκληρη την περίοδο προσομοίωσης. Κατά τον έλεγχο της κυκλοφορίας ο ελεγκτής χρειάζεται να έχει ένα στόχο ή οδηγία, όπως την ελαχιστοποίηση του συνολικού χρόνου που δαπανάται για την κίνηση ενός οχήματος εντός του δικτύου, την ελαχιστοποίηση της συνολικής απόστασης που διανύθηκε, την ελαχιστοποίηση της συνολικής κατανάλωσης καυσίμου, την ελαχιστοποίηση των μηκών της ουράς, τη μεγιστοποίηση της ασφάλειας των οχημάτων στο δίκτυο κ.λπ. ή ένα συνδυασμό αυτών των στόχων.

Χρησιμοποιώντας τον βέλτιστο έλεγχο, ο ελεγκτής υπολογίζει τις βέλτιστες τιμές ελέγχου, με βάση τις απαιτήσεις που έχουν οριστεί. Στην πράξη, αυτές οι απαιτήσεις δεν μπορούν να γίνουν γνωστές εκ των προτέρων. Ωστόσο, συχνά η μελλοντική ζήτηση μπορεί να εκτιμηθεί αρκετά καλά από τις μετρήσεις εκατέρωθεν της ράμπας σε συνδυασμό με ιστορικά δεδομένα.

Τα MPC μοντέλα ακολουθούν την ίδια λογική με τον συμβατικό βέλτιστο έλεγχο απλά μειώνουν τον χρονικό ορίζοντα πρόβλεψης χρησιμοποιώντας ανατροφοδότηση και έτσι το μοντέλο κάνει περισσότερες προβλέψεις σε μικρότερο χρονικό διάστημα δίνοντας καλύτερα αποτελέσματα.^{[15],[16]}

1.4.3. Τεχνητή νοημοσύνη (Artificial Intelligence)

Οι τεχνικές Τεχνητής Νοημοσύνης (AI) στοχεύουν στη διευκόλυνση μίας μηχανής για την επίλυση ενός προβλήματος μέσω νοημοσύνης. Με τον όρο τεχνητή νοημοσύνη, εννοούμε την ικανότητα ενός λογισμικού να αντιλαμβάνεται μια κατάσταση, να αιτιολογεί το πρόβλημα και να ενεργεί. Οι τεχνικές AI χρησιμοποιούνται κυρίως σε συστήματα υποστήριξης αποφάσεων και μπορούν να ταξινομηθούν στις παρακάτω κατηγορίες:^{[17],[18],[19],[20],[21]}

- Αιτιολογία βασισμένη σε συγκεκριμένες περιπτώσεις (Case based reasoning)
- Ασαφής λογική (Fuzzy Logic)
- Συστήματα βασισμένα σε κανόνες (Rule based systems)
- Τεχνητά νευρωνικά δίκτυα (Artificial neural networks)
- Συστήματα πολλαπλών πρακτόρων (Multi-agent systems)
- Γενετικοί αλγόριθμοι (Genetic algorithms)
- Ευφυΐα σμήνους (Swarm intelligence)
- Ενίσχυση της μάθησης (Reinforcement learning)

Η **αιτιολογία βασισμένη σε συγκεκριμένες περιπτώσεις**, όπως υποδηλώνει και το όνομα, λύνει ένα πρόβλημα χρησιμοποιώντας τη γνώση που αποκτήθηκε από προηγούμενες παρόμοιες καταστάσεις. Με αυτόν τον τρόπο, η συγκεκριμένη τεχνική μαθαίνει τον τρόπο επίλυσης ενός νέου προβλήματος, δοκιμάζει την προτεινόμενη λύση χρησιμοποιώντας μεθόδους προσομοίωσης και αποθηκεύει τη νέα λύση σε μια βάση δεδομένων. Ένα βασικό

μειονέκτημα της μεθόδου είναι ότι σε κάποιες περιπτώσεις που δεν έχουν εμφανιστεί στο παρελθόν και δεν υπάρχει γνώση δεν είναι σαφές τι πρέπει να γίνει. Ωστόσο, μπορούν προγραμματιστικά να προστεθούν νέες περιπτώσεις σε πραγματικό χρόνο για την αντιμετώπιση αυτής της αδυναμίας.

Τα συστήματα **ασαφούς λογικής**, όπως και οι άνθρωποι, μπορούν να χειριστούν καταστάσεις στις οποίες οι διαθέσιμες πληροφορίες για το σύστημα είναι ασαφείς. Για την αντιμετώπιση τέτοιων καταστάσεων, ασαφή σύνολα χρησιμοποιούνται για να χαρακτηρίζουν τις μεταβλητές του συστήματος με μη ποσοτικό τρόπο. Τα ασαφή σύνολα μπορούν να προσδιοριστούν με τη χρήση λειτουργιών προσχώρησης σε κατανομή (π.χ. Gaussian, τρίγωνο ή κανονική) λαμβάνοντας μια τιμή μεταξύ 0 και 1 και αυτό δείχνει σε ποιο βαθμό ένα δεδομένο στοιχείο ανήκει σε μία κατηγορία (π.χ. μια ταχύτητα μπορεί να είναι 60% "υψηλή" και 40% "μέση"). Η τιμή αυτή στη συνέχεια μπορεί να χρησιμοποιηθεί για να συνδυάσει διάφορους κανόνες και να οδηγήσει σε συμπεράσματα. Μια από τις κύριες δυσκολίες ενός ασαφούς συστήματος συνήθως είναι η επιλογή των κατάλληλων τιμών συνδρομής για τις μεταβλητές εισόδου και εξόδου. Συστήματα ασαφούς λογικής τις περισσότερες φορές συνδυάζονται και με άλλα συστήματα Τεχνητής Νοημοσύνης.

Τα **συστήματα που βασίζονται σε κανόνες** λύνουν ένα πρόβλημα χρησιμοποιώντας κανόνες "εάν-τότε" (if-then). Αυτοί οι κανόνες προκύπτουν από ειδικές γνώσεις και αποθηκεύονται σε μία βάση συμπερασμάτων. Η μηχανή συμπερασμάτων έχει μια εσωτερική μνήμη που αποθηκεύει κανόνες και πληροφορίες σχετικά με το πρόβλημα, ένα μηχανισμό που αντιστοιχίζει τα διάφορα μοτίβα και ένα μηχανισμό εφαρμογής των κανόνων. Ο μηχανισμός που κάνει την αντιστοίχιση αναζητά όλη την εσωτερική μνήμη προκειμένου να επιλέξει ποιους κανόνες πρέπει να εφαρμοστούν και στη συνέχεια αυτοί εφαρμόζονται από τον αντίστοιχο μηχανισμό. Αυτά τα συστήματα είναι κατάλληλα για την επίλυση προβλημάτων όπου οι εμπειρογνώμονες μπορούν να πάρουν ξεκάθαρες αποφάσεις. Ωστόσο, αυτού του είδους τα συστήματα λειτουργούν μόνο με τους κανόνες που έχουν ήδη δημιουργηθεί και στη βασική τους εφαρμογή δεν περιλαμβάνουν μάθηση.

Τα **τεχνητά νευρωνικά δίκτυα** προσπαθούν να μιμηθούν τον τρόπο με τον οποίο επεξεργάζεται την πληροφορία ο ανθρώπινος εγκέφαλος. Αυτά τα συστήματα είναι χρήσιμα στην επίλυση μη γραμμικών προβλημάτων στα οποία οι κανόνες ή ο αλγόριθμος για την εξεύρεση λύσεων είναι δύσκολο να προσδιοριστούν. Η βασική μονάδα επεξεργασίας σε ένα νευρωνικό δίκτυο ονομάζεται νευρώνας ή κόμβος. Κάθε κόμβος ενεργοποιεί ένα νέο σήμα όταν δέχεται ένα σήμα εισόδου από τους άλλους συνδεδεμένους κόμβους. Αυτοί οι κόμβοι είναι οργανωμένοι σε ένα στρώμα εισόδου, ένα στρώμα εξόδου και ένα πλήθος κρυφών επιπέδων τα οποία είναι διασυνδεδεμένα με συνδέσμους ή συνάψεις, καθένα από τα οποία σχετίζεται με βάρη. Ένα μειονέκτημα είναι ότι τα τεχνητά νευρωνικά δίκτυα είναι μη

ενημερωτικά μοντέλα και δεν παρέχουν εξήγηση για τα αποτελέσματα ή για τυχόν αποτυχία που μπορεί να προκύψει στη διαδικασία.

Στα **συστήματα πολλαπλών πρακτόρων (agents)**, ως agent ορίζεται μία οντότητα που μπορεί να αντιληφθεί το περιβάλλον του μέσω αισθητήρων (sensors) και να διαμορφώσει το περιβάλλον του μέσω ενεργοποιητών (actuators) με τέτοιο τρόπο ώστε να πληρούνται τα κριτήρια απόδοσης. Τα συγκεκριμένα συστήματα αποτελούνται από ένα δίκτυο παραγόντων που αλληλεπιδρούν μεταξύ τους με σκοπό την επίτευξη συγκεκριμένων στόχων. Μια γλώσσα επικοινωνίας υψηλού επιπέδου χρησιμοποιείται από τους agents για σκοπούς επικοινωνίας και διαπραγμάτευσης. Τέτοια συστήματα βρίσκουν εφαρμογή σε πολύ σύνθετες εφαρμογές ωστόσο η δυναμική τους φύση μπορεί να οδηγήσει σε προβλήματα συγκρουόμενων συμφερόντων ή σε κακή διαχείριση πόρων.

1.4.3.1. Εφαρμογή της τεχνητής νοημοσύνης (AI) στο Ramp Metering

Από τις μεθόδους AI που αναφέρθηκαν παραπάνω, η συλλογιστική και η ασαφής λογική είναι αυτές που χρησιμοποιούνται πιο συχνά στη βιβλιογραφία για σκοπούς ελέγχου της κυκλοφορίας. Στην πράξη, συστήματα βασισμένα σε κανόνες χρησιμοποιούνται επίσης συχνά για τη διαχείριση και τον έλεγχο της κυκλοφορίας.

Τα συστήματα που βασίζονται σε περιπτώσεις αναπτύσσονται για ένα ευρύ φάσμα σεναρίων που μπορούν να παρουσιαστούν σε ένα δίκτυο κυκλοφορίας. Τέτοιου είδους συστήματα χρησιμοποιούν ιστορικά και on-line δεδομένα (π.χ., ramp metering, κλεισίματα λωρίδων, κ.λπ.) και προσπαθούν μέσω κατάλληλων προβλέψεων και με βάση τις εισόδους που τους έχουν δοθεί να πραγματοποιήσουν την πιο σωστή πρόβλεψη για τη διαχείριση της κυκλοφορίας.

Στην παρούσα εργασία όπως θα αναφερθεί και σε επόμενο κεφάλαιο θα αναπτυχθούν τόσο αλγόριθμοι συστημάτων που βασίζονται σε κανόνες (if then -else) σε αρχικό στάδιο και σε επέκταση όλων των κοινώς χρησιμοποιούμενων μεθόδων για τη διαχείριση της κυκλοφορίας θα αναπτυχθεί ένα νευρωνικό δίκτυο το οποίο θα ελέγχει δύο (2) ράμπες εισόδου σε ένα τμήμα του αυτοκινητόδρομου.

1.4.4. Στατικός έλεγχος με ανατροφοδότηση (Static feedback control)

Τα δυναμικά συστήματα μπορούν να ελέγχονται με δύο τρόπους: χρησιμοποιώντας έλεγχο ανοιχτού βρόχου (open-loop control) ή έλεγχο κλειστού βρόχου (closed-loop control). Σε ένα σύστημα ανοιχτού βρόχου, η είσοδος δεν εξαρτάται από την έξοδο του συστήματος, ενώ σε ένα σύστημα κλειστού βρόχου, ο έλεγχος είναι συνάρτηση της εξόδου. Τα συστήματα ελέγχου ανατροφοδότησης ή κλειστού βρόχου είναι κατάλληλα για εφαρμογές που περιλαμβάνουν αβεβαιότητα ή σφάλματα στη μοντελοποίηση.

Στις μεθόδους στατικού ελέγχου με ανατροφοδότηση, ο ελεγκτής λαμβάνει μετρήσεις από το σύστημα και καθορίζει τις ενέργειες ελέγχου βασιζόμενος στην τρέχουσα κατάσταση του συστήματος με τέτοιο τρόπο ώστε να βελτιωθεί η απόδοση του. Μερικά παραδείγματα τέτοιων ελεγκτών είναι ελεγκτές ανατροφοδότησης κατάστασης και ελεγκτές PID (Proportional – Integral – Derivative). Ωστόσο, η στρατηγική στατικής ανατροφοδότησης γενικά δεν μπορεί να ανταπεξέλθει σε εξωτερικούς περιορισμούς και αυτό αποτελεί βασικό της μειονέκτημα.^[22]

1.4.4.1. Εφαρμογή στατικού ελέγχου με ανατροφοδότηση στη μέτρηση κεκλιμένου επιπέδου

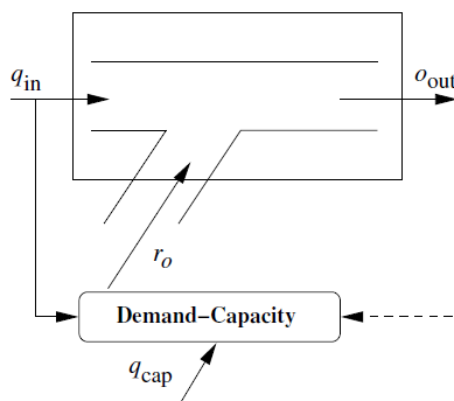
Ένα σύστημα μέτρησης ράμπας που βασίζεται σε στατική ανατροφοδότηση εκτελεί τον έλεγχο χρησιμοποιώντας μετρήσεις κυκλοφορίας σε πραγματικό χρόνο που συλλέγονται από τους μετρητές που βρίσκονται στη ράμπα. Οι δύο βασικές μέθοδοι που χρησιμοποιούν τον στατικό έλεγχο με ανατροφοδότηση είναι:

1. Η στρατηγική ζήτησης – χωρητικότητας
2. ALINEA

Η στρατηγική ζήτησης – χωρητικότητας εκφράζεται με βάση την παρακάτω σχέση:

$$r_o(k) = \begin{cases} q_{cap} - q_{in(k-1)} & \text{έαν } O_{out}(k) \leq O_{cr} \\ q_{min} & \text{διαφορετικά} \end{cases} \quad (3)$$

όπου $r_o(k)$ είναι ο ρυθμός ροής επί της ράμπας στο στάδιο k του δείγματος, $o_{out}(k)$ είναι η χωρητικότητα του αυτοκινητόδρομου, o_{cr} είναι η κρίσιμη πυκνότητα στην οποία η κυκλοφοριακή ροή τείνει να είναι στο μέγιστο της, και q_{min} είναι η προκαθορισμένη ελάχιστη τιμή ροής ράμπας. Αυτή η στρατηγική καθορίζει την τιμή του ρυθμού ροής ως τη διαφορά μεταξύ της μεταγενέστερης δυναμικότητας και της ροής ανάντη, μέχρις ότου να φτάσουμε στην κρίσιμη πυκνότητα. Μόλις φτάσουμε στην κρίσιμη πυκνότητα, ο ρυθμός ροής επί της ράμπας μειώνεται σε μια ελάχιστη τιμή για να αποφευχθεί η εμφάνιση συμφόρησης. Επίσης, αυτή η στρατηγική ζήτησης προσπαθεί να λύσει ή να αποτρέψει τη συμφόρηση μόνο στην περίπτωση που ο αυτοκινητόδρομος λειτουργεί στο μέγιστο.

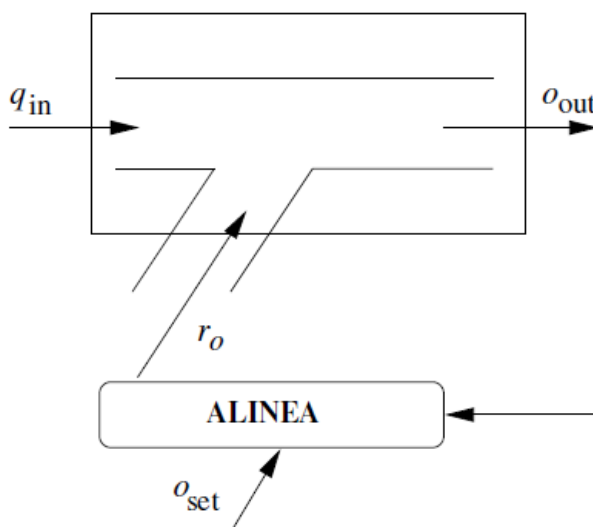


Εικόνα 4. Στρατηγική ζήτησης - χωρητικότητας

Μία ακόμη από τις πιο γνωστές στρατηγικές μέτρησης κεκλιμένων επιπέδων είναι η ALINEA. Η ALINEA είναι ένας αλγόριθμος κλειστού βρόχου ο οποίος καθορίζει τον ρυθμό εισόδου/κίνησης από τη ράμπα κατά τέτοιο τρόπο ώστε η κατάντη ροή να είναι στην επιθυμητή ή προκαθορισμένη τιμή o_{set} :

$$r_o(k) = r_o(k-1) + K_r(o_{set} - o_{out}(k)) \quad (4)$$

όπου $r_o(k)$ είναι ο ρυθμός ροής στη ράμπα στο βήμα k του δείγματος, $o_{out}(k)$ είναι η χωρητικότητα του αυτοκινητόδρομου κατάντη της ράμπας, και K_r είναι η παράμετρος ρύθμισης. Μια από τις δυσκολίες αυτής της μεθόδου είναι ο προσδιορισμός των κατάλληλων τιμών για τις παραμέτρους (set-points).



Εικόνα 5. ALINEA

Παρόλο που οι στρατηγικές ελέγχου στατικής ανατροφοδότησης που αναφέρθηκαν παραπάνω βασίζονται μετρήσεις πραγματικού χρόνου μπορούν να εφαρμοστούν σε τοπικό επίπεδο και με πληροφορίες που συλλέγονται μόνο στο συγκεκριμένο σημείο.^{[23],[24]}

1.4.5. Σύγκριση μεθόδων

Στο παρόν υποκεφάλαιο θα προσεγγίσουμε μία σύγκριση των μεθόδων που αναλύθηκαν παραπάνω η οποία θα εστιάζει στους παρακάτω τομείς:

- Υπολογιστική πολυπλοκότητα
- Συμπερίληψη μελλοντικών δεδομένων/εισόδων
- Βασισμένο σε μοντέλο ή όχι
- Συμπερίληψη αυστηρών περιορισμών
- Επεκτασιμότητα

1.4.5.1. Υπολογιστική πολυπλοκότητα

Παρόλο που οι μέθοδοι σχεδιασμού ελέγχου που συζητήθηκαν παραπάνω παρέχουν μη βέλτιστες επιδόσεις (εκτός από τον βέλτιστο έλεγχο στην περίπτωση χωρίς σφάλματα), η υπολογιστική πολυπλοκότητα αυτών των μεθόδων ποικίλλει ανάλογα με την περιοχή εφαρμογής τους. Οι μέθοδοι στατικού ελέγχου με ανάδραση έχουν χαμηλές υπολογιστικές απαιτήσεις, αλλά στην πράξη εφαρμόζονται κατά κύριο λόγο σε μικρής κλίμακας συστήματα. Οι υπολογιστικές απαιτήσεις για τις ΑΙ τεχνικές είναι σχετικά υψηλότερες αλλά αρκετά μικρότερες σε σχέση με αυτές του βέλτιστου ελέγχου και της MPC ειδικότερα όταν η τελευταία χρησιμοποιεί online βελτιστοποίηση, μεγάλο πλήθος δεδομένων και συνθηκών ελέγχου. Και οι τρεις τεχνικές ελέγχου απαιτούν επίσης υπολογισμούς off-line που αφορούν την αναγνώριση και τη βαθμονόμηση των παραμέτρων του μοντέλου καθώς και τη ρύθμιση παραμέτρων, την εκπαίδευση ή τη δημιουργία κανόνων. Αυτές οι off-line υπολογιστικές απαιτήσεις καθορίζονται κυρίως από το μοντέλο και την πολυπλοκότητά του (τάξη μοντέλου, βαθμό μη γραμμικότητας, μέγεθος δεδομένων, συνθήκες ελέγχου κ.λπ.).

1.4.5.2. Συμπερίληψη μελλοντικών δεδομένων/εισόδων

Οι ελεγκτές στατικού ελέγχου με ανάδραση δεν λαμβάνουν υπόψη τους τις μελλοντικές εξωτερικές εισόδους. Ο βέλτιστος έλεγχος και MPC βασίζονται στην πρόβλεψη και επομένως μπορούν να λάβουν υπόψη τους τη γνώση που παρέχεται από μελλοντικά δεδομένα εισόδου για παράδειγμα, στον έλεγχο της κυκλοφορίας κάποιος θα μπορούσε να ανιχνεύσει κυκλοφοριακές ροές χρησιμοποιώντας ανιχνευτές που βρίσκονται ανάντη και να χρησιμοποιήσει αυτές τις πληροφορίες. Τέλος, οι τεχνικές ΑΙ μπορούν επίσης να χρησιμοποιούν προβλέψεις έτσι ώστε για παράδειγμα σε ένα νευρωνικό δίκτυο να γίνεται καλύτερη εκπαίδευση και το αποτέλεσμα (outputs) του νευρωνικού να είναι ακόμα πιο ακριβές.

1.4.5.3. Βασισμένο σε μοντέλο ή όχι

Όλες οι προσεγγίσεις ελέγχου έχουν ρητά ή σιωπηρά ανάγκη από ένα μοντέλο για τη συλλογή δεδομένων και / ή για σωστή παραμετροποίηση (tuning). Ωστόσο, με τον όρο βασισμένο σε μοντέλο εννοούμε εάν εξετάζεται η μέθοδος ελέγχου ως ένα σαφές μοντέλο προκειμένου να προσδιοριστούν οι βέλτιστες ενέργειες ελέγχου. Παρόλο που οι παράμετροι ενός ελεγκτή στατικής ανάδρασης μπορούν να συντονιστούν χρησιμοποιώντας προσομοιώσεις με βάση ένα μοντέλο, η συγκεκριμένη μέθοδος ελέγχου δεν περιλαμβάνει ρητώς ένα μοντέλο του συστήματος κίνησης εντός της υλοποίησής της. Με παρόμοιο τρόπο οι κανόνες ή οι περιπτώσεις ενός ελεγκτή που βασίζεται σε ΑΙ μπορούν επίσης να προσδιοριστούν χρησιμοποιώντας μοντέλα προσομοίωσης, αλλά τα μοντέλα δεν υπάρχουν ρητά στον ελεγκτή. Από την άλλη πλευρά, οι μέθοδοι βέλτιστου ελέγχου και οι ελεγκτές MPC χρησιμοποιούν

γενικά ένα σαφές μοντέλο του κυκλοφοριακού συστήματος ως πρόβλεψη κατά τον προσδιορισμό των βέλτιστων ενεργειών ελέγχου.

1.4.5.4. Συμπερίληψη αυστηρών περιορισμών

Η εξέταση αυστηρών περιορισμών είναι μια αναπόφευκτη πτυχή που πρέπει να συμπεριληφθεί κατά τον έλεγχο και διαχείριση των ροών κυκλοφορίας. Οι περιορισμοί είναι εκείνοι που πρέπει απαραίτητα να ικανοποιηθούν. Σε γενικές γραμμές, ο έλεγχος στατικής ανάδρασης δεν λαμβάνει υπόψη τους εξωτερικούς περιορισμούς. Αυτή η έλλειψη της συμπερίληψης των περιορισμών αποδεικνύεται ένα από τα σημαντικότερα μειονεκτήματα των μεθόδων στατικής ανάδρασης. Οι περισσότερες μέθοδοι AI επίσης δεν λαμβάνουν ρητά υπόψη αυστηρούς περιορισμούς. Ο βέλτιστος έλεγχος και η MPC από την άλλη χρησιμοποιούν βελτιστοποίηση και ως εκ τούτου μπορούν να συμπεριλαμβάνουν ρητά αυστηρούς περιορισμούς κατά τον καθορισμό των βέλτιστων ενεργειών ελέγχου.

1.4.5.5. Επεκτασιμότητα

Ανάλογα με τη μέθοδο που χρησιμοποιείται, μπορούμε να κατηγοριοποιήσουμε τα συστήματα ελέγχου και διαχείρισης ως εξής:^{[25],[26],[27]}

- Κεντρικά (Centralized)
- Διανεμημένα (Distributed)
- Αποκεντρωμένα (Decentralized)
- Ιεραρχικά (Hierarchical)

Σε ένα κεντρικό σύστημα όλα τα μέτρα ελέγχου διαχειρίζονται από ένα μόνο σημείο, το κέντρο ελέγχου κίνησης. Ωστόσο, στην πράξη μια τέτοια προσέγγιση δεν είναι εφικτή λόγω της έλλειψης επεκτασιμότητας και συνοχής των συστημάτων αυτών, τις απαιτήσεις επικοινωνίας και την υπολογιστική πολυπλοκότητα. Ένας τρόπος αντιμετώπισης αυτών των ζητημάτων είναι να χρησιμοποιηθεί μια κατανεμημένη προσέγγιση στην οποία υπάρχουν πολλοί τοπικοί ελεγκτές και ο καθένας ελέγχει τη δική του περιοχή ή ένα σύνολο μετρήσεων οι οποίες αποστέλλονται σε τοπικούς ελεγκτές που με τη σειρά τους επικοινωνούν μεταξύ τους και συντονίζουν τις ενέργειες τους ως ένα ενιαίο σύνολο. Εάν υπάρχει μόνο επικοινωνία, αλλά δεν υπάρχει συντονισμός, έχουμε μια αποκεντρωμένη δομή ελέγχου, η οποία είναι συνήθως ευκολότερη να εφαρμοστεί έχοντας όμως χειρότερη βελτιστοποίηση. Τα ιεραρχικά πλαίσια ελέγχου συνδυάζουν χαρακτηριστικά του κεντρικού και κατανεμημένου ελέγχου: χρησιμοποιούν ένα ιεραρχικό πλαίσιο με πολλά στρώματα ελέγχου στα οποία οι ελεγκτές χαμηλότερου επιπέδου αναλαμβάνουν μικρές περιοχές επιρροής με ταχεία δυναμική ενώ οι ελεγκτές υψηλότερου επιπέδου αναλαμβάνουν τον συντονισμό αυτών των μικρότερων περιοχών καθώς σε μεγαλύτερη έκταση η δυναμική των συστημάτων είναι μικρότερη.

Σε αυτό το πλαίσιο, οι στατικοί ελεγκτές ανάδρασης εντοπίζονται κυρίως και χρησιμοποιούνται για τον έλεγχο συστημάτων μικρής κλίμακας (με κεντρικό τρόπο) ή ως ελεγκτές χαμηλότερου επιπέδου σύστημα ιεραρχικής δομής. Οι περισσότερες μέθοδοι ελέγχου βασισμένες στο ΑΙ (εκτός από τις προσεγγίσεις πολλαπλών παραγόντων) είναι αποδοτικές σε συστήματα μικρής κλίμακας και μπορούν επίσης να χρησιμοποιηθούν σε ένα ιεραρχικό πλαίσιο ελέγχου. Η MPC μέθοδος (και ο βέλτιστος έλεγχος) καθορίζει τις ενέργειες ελέγχου βάσει της τρέχουσας και της προβλεπόμενης κατάστασης του συστήματος και, ως εκ τούτου, επιτρέπει τον συντονισμό σε επίπεδο συστήματος για τις ενέργειες ελέγχου. Ωστόσο, η MPC μπορεί να χρησιμοποιηθεί σε όλα τα επίπεδα ενός ιεραρχικού πλαισίου ελέγχου, αλλά λόγω της υπολογιστικής της πολυπλοκότητας είναι λιγότερο κατάλληλη για τα χαμηλότερα επίπεδα. Παρακάτω φαίνεται ένας συγκριτικός πίνακας των παραπάνω μεθόδων:

Μέθοδος	Υπολογιστική Πολυπλοκότητα	Περιορισμοί	Μελλοντικά Δεδομένα	Βασισμένο σε μοντέλο	Επεκτασιμότητα
Στατικός έλεγχος με ανάδραση	Χαμηλή	Όχι	Όχι	Όχι ρητά	Τοπικά
Βέλτιστος έλεγχος και MPC	Υψηλή	Ναι	Ναι	Ναι	Συστημικά
ΑΙ	Μέτρια	Όχι	Όχι	Όχι ρητά	Τοπικά

Πίνακας 1. Σύγκριση μεθόδων

ΚΕΦΑΛΑΙΟ 2. ΠΕΡΙΓΡΑΦΗ, ΟΡΙΣΜΟΣ ΚΑΙ ΑΝΑΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

2.1. Γενικά

Η συνεχής αύξηση του πληθυσμού και των οχημάτων όπως προαναφέρθηκε και στο προηγούμενο κεφάλαιο οδηγεί στην αύξηση του κυκλοφοριακού φόρτου στις κεντρικές οδικές αρτηρίες των πόλεων δημιουργώντας έτσι συχνά μεγάλες καθυστερήσεις και μία δυσάρεστη κατάσταση για τους οδηγούς. Οι αυτοκινητόδρομοι στα αστικά κέντρα έχουν σχεδιαστεί προκειμένου να εξυπηρετούν όσο το δυνατόν μεγαλύτερο φόρτο κυκλοφορίας περιφερειακά καθώς και μεταξύ αυτών. Βασικό χαρακτηριστικό και στόχος κατά τον σχεδιασμό όλων των αυτοκινητόδρομων είναι ότι τόσο η είσοδος όσο και η έξοδος από αυτούς πρέπει να είναι κατά το δυνατόν ελεγχόμενη και πραγματοποιείται μέσω ανισόπεδων κόμβων (ramps). Επομένως γίνεται εύκολα αντιληπτό ότι η μελέτη για τη διαχείριση των ανισόπεδων κόμβων αποτελεί μείζονος σημασίας ζήτημα για την ομαλή λειτουργία των αυτοκινητόδρομων και την αποφυγή φαινομένων συμφόρησης.^[28]

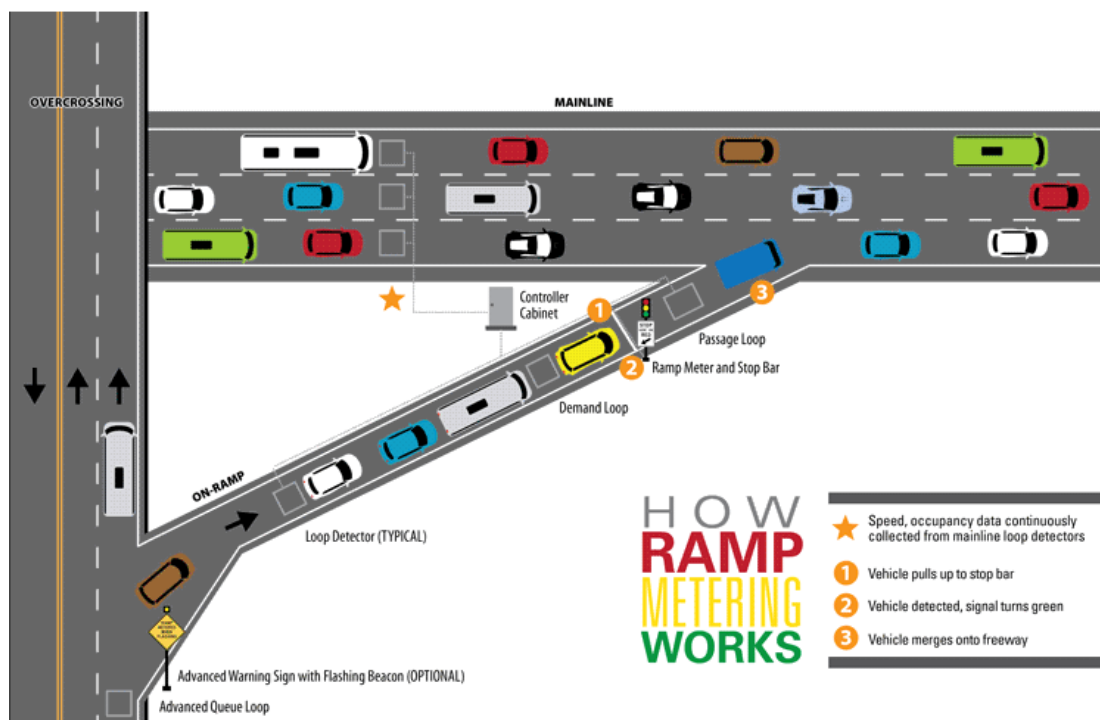
Βασικός στόχος σε κάθε μελέτη για τη διαχείριση της κίνησης είναι η είσοδος και η έξοδος των οχημάτων από τις κύριες οδικές αρτηρίες να γίνεται χωρίς να παρεμποδίζονται τόσο τα οχήματα στο κύριο ρεύμα του αυτοκινητόδρομου όσο και αυτά που εισέρχονται ή εξέρχονται από αυτόν. Συγκεκριμένα, μία ράμπα εισόδου η οποία περιλαμβάνει και την αντίστοιχη βοηθητική λωρίδα επιτρέπει στο εισερχόμενο όχημα να επιταχύνει έτσι ώστε να αναπτύξει την απαραίτητη ταχύτητα και να μην υπάρξει η ανάγκη να ανακοπεί η πορεία ούτε του εισερχόμενου οχήματος αλλά ούτε και του ήδη κινούμενου στον αυτοκινητόδρομο οχήματος.^[28]

Στην παρούσα εργασία θα μελετηθεί η εφαρμογή του Ramp Metering για την είσοδο των αυτοκινήτων στο κύριο ρεύμα του αυτοκινητόδρομου και θα εξεταστούν διάφορα σενάρια λειτουργίας και αποτελεσμάτων. Το ζητούμενο είναι να αναπτυχθούν διάφοροι αλγόριθμοι ελέγχου που αφορούν στο Ramp Metering με τελικό στόχο την κατασκευή ενός νευρωνικού μοντέλου το οποίο θα διαχειρίζεται τα κυκλοφοριακά δεδομένα και θα αποφασίζει για την είσοδο των οχημάτων στο κύριο ρεύμα του αυτοκινητόδρομου. Μετά τη διατύπωση και την υλοποίηση των διαφόρων σεναρίων και αλγορίθμων θα γίνει σύγκριση των αποτελεσμάτων τους κάτω από δεδομένες συνθήκες και θα εξεταστεί η αποτελεσματικότητά τους με κριτήριο τη βελτιστοποίηση της ροής και την αποφυγή φαινομένων κυκλοφοριακής συμφόρησης καθώς και την αποδοτικότητά τους .

2.2. Περιγραφή Λειτουργίας της Ελεγχόμενης Πρόσβασης

Η ελεγχόμενη πρόσβαση στους ανισόπεδους κόμβους (Ramps) αποτελεί ένα βασικό εργαλείο για την επίτευξη της ζητούμενης ροής στους αυτοκινητόδρομους και την αποφυγή κυκλοφοριακής συμφόρησης. Ο βασικός στόχος είναι να μην μειωθεί η στάθμη εξυπηρέτησης ενός κεντρικού οδικού άξονα με δεδομένο πάντα ότι η χωρητικότητα δεν είναι απεριόριστη. Με γνώμονα λοιπόν αυτόν τον στόχο, τοποθετούνται φωτεινοί σηματοδότες στις ράμπες εισόδου, προκειμένου να ελέγχεται ο ρυθμός με τον οποίο εισέρχονται τα οχήματα εντός του κύριου ρεύματος του αυτοκινητόδρομου.^{[28],[35]}

Η ελεγχόμενη πρόσβαση χρησιμοποιείται ευρύτατα στις ράμπες εισόδου έτσι ώστε να επιτυγχάνεται η βέλτιστη δυνατή ρύθμιση της κυκλοφορίας. Πιο συγκεκριμένα, είναι δυνατό να μειωθούν οι χρόνοι αναμονής των οχημάτων για είσοδο στον αυτοκινητόδρομο όσο και ο χρόνος ταξιδιού σε αυτόν. Επιπλέον, μέσω της ελεγχόμενης πρόσβασης αυξάνεται η ασφάλεια κατά την είσοδο στο κυρίως ρεύμα μειώνοντας έτσι τον κίνδυνο και τον αριθμό των ατυχημάτων που προκαλούνται ιδιαίτερα σε καταστάσεις αυξημένης κίνησης. Μέσω κατάλληλων αισθητηρίων ανιχνεύεται η πυκνότητα των οχημάτων στον αυτοκινητόδρομο, μόλις αυτή η πυκνότητα ξεπεράσει κάποιο όριο τότε η ροή εισόδου από τη ράμπα μειώνεται. Εάν τώρα η πυκνότητα των οχημάτων είναι αρκετά υψηλή και υπάρχει πτώση της στάθμης εξυπηρέτησης των οχημάτων που κινούνται εντός του αυτοκινητόδρομου τότε ανακόπτεται τελείως η είσοδος σε αυτόν.

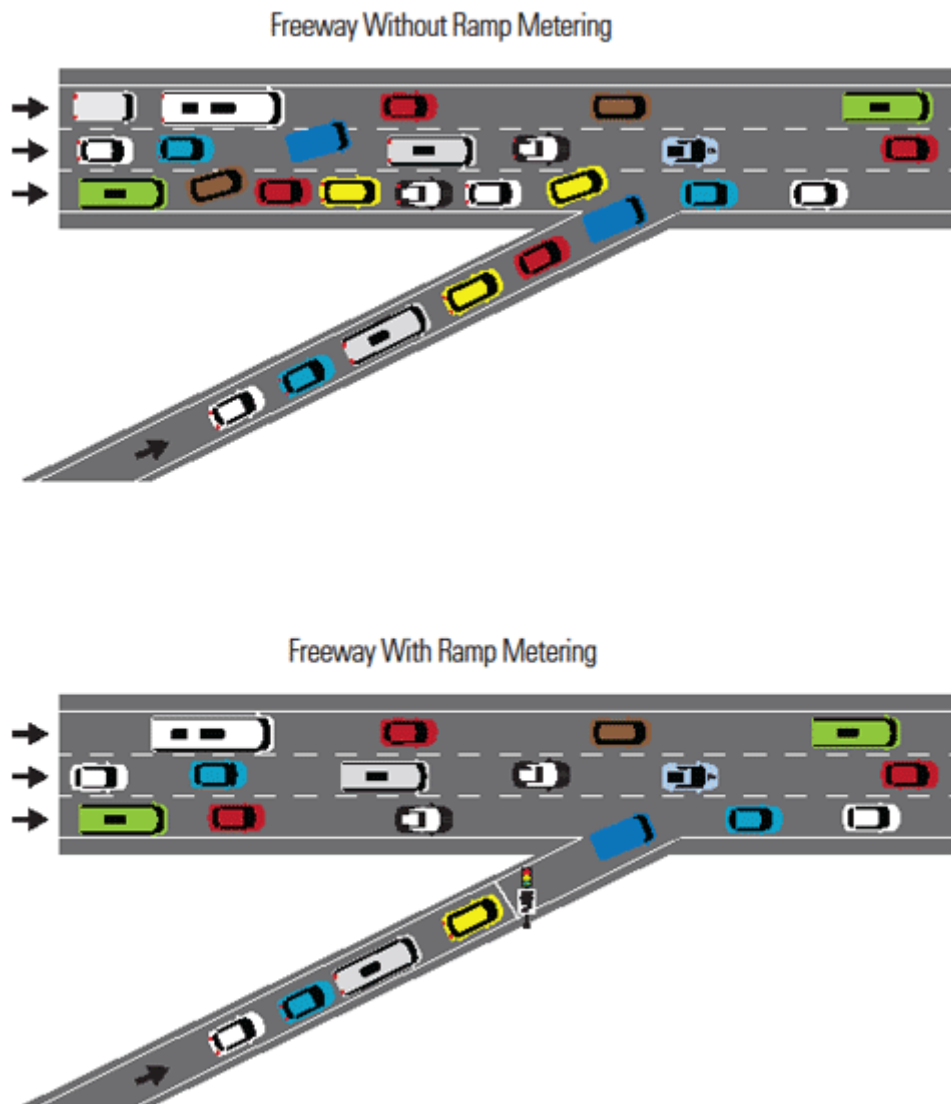


Εικόνα 6. Ελεγχόμενη πρόσβαση μέσω ράμπας. Πηγή: FHWA 2014

Μία ιδέα για την ελεγχόμενη πρόσβαση φαίνεται στην παραπάνω εικόνα. Τα οχήματα εισέρχονται στη ράμπα και μόλις φτάσουν στη γραμμή στάσης σταματούν ή συνεχίζουν την

πορεία τους ανάλογα με την ένδειξη του φωτεινού σηματοδότη που βρίσκεται εκεί. Τόσο στην είσοδο της ράμπας όσο και κατά μήκος αυτής αλλά και στον αυτοκινητόδρομο πριν από αυτή υπάρχουν κατάλληλοι αισθητήρες οι οποίοι συλλέγουν δεδομένα και με βάση αυτά υπολογίζεται η ροή εισόδου στον αυτοκινητόδρομο.^{[28],[35]}

Η ελεγχόμενη πρόσβαση πέρα από τα οφέλη που αφορούν στην αύξηση της μέσης ταχύτητας κίνησης σε ώρες αιχμής καθώς και την ομαλή ροή γενικότερα, προσφέρει μεγαλύτερη ασφάλεια στις μετακινήσεις, εξοικονόμηση καυσίμων και μείωση της ατμοσφαιρικής ρύπανσης.



Εικόνα 7. Σύγκριση ράμπας με και χωρίς ελεγχόμενη πρόσβαση. Πηγή: FHWA 2014

Όπως φαίνεται και στην παραπάνω εικόνα αλλά και με βάση την καθημερινότητα μας γίνεται άμεσα αντιληπτό ότι σε κεντρικές αρτηρίες του οδικού δικτύου που δεν υπάρχει ελεγχόμενη πρόσβαση επικρατεί κυκλοφοριακή συμφόρηση, αυξημένοι ρύποι αλλά ενέχει και αυξημένος κίνδυνος ατυχημάτων.

2.3. Μέθοδοι Ελεγχόμενης Πρόσβασης

Η επιλογή ενός συστήματος ελεγχόμενης πρόσβασης εξαρτάται από τον τελικό στόχο την υπάρχουσα κατάσταση, τους διαθέσιμους χρηματικούς πόρους αλλά και τα λειτουργικά κόστη του συστήματος. Ο παρακάτω πίνακας παρέχει πληροφορίες για τις διάφορες προσεγγίσεις ενός συστήματος ελεγχόμενης πρόσβασης που μπορούν να χρησιμοποιηθούν σε μία εφαρμογή διαχείρισης κίνησης.

	ΠΡΟΚΑΘΟΡΙΣΜΕΝΟΥ ΧΡΟΝΟΥ	ΔΥΝΑΜΙΚΑ
ΤΟΠΙΚΗ ΥΛΟΠΟΙΗΣΗ	<ul style="list-style-type: none"> • Κατάλληλα για προβλήματα με τοπικό χαρακτήρα • Δεν είναι απαραίτητη η μέτρηση της κίνησης • Απαίτηση για περιοδικές αναβαθμίσεις • Ακατάλληλα σε περιβάλλον δυναμικών συνθηκών • Υψηλότερα λειτουργικά κόστη σε σχέση με τα δυναμικά συστήματα 	<ul style="list-style-type: none"> • Κατάλληλα για προβλήματα με τοπικό χαρακτήρα • Απαραίτητη η μέτρηση της κίνησης • Υψηλότερο αρχικό κεφάλαιο και κόστος συντήρησης σε σχέση με τα συστήματα προκαθορισμένου χρόνου • Προσφέρει περισσότερα οφέλη λόγω του δυναμικού χαρακτήρα καθώς αποκρίνεται στις υπάρχουσες συνθήκες
ΣΥΣΤΗΜΙΚΗ ΥΛΟΠΟΙΗΣΗ	<ul style="list-style-type: none"> • Κατάλληλα για προβλήματα μεγάλου εύρους • Δεν είναι απαραίτητη η μέτρηση της κίνησης • Χρησιμοποιούνται σπάνια συγκριτικά με τα δυναμικά συστήματα 	<ul style="list-style-type: none"> • Κατάλληλα για προβλήματα μεγάλου εύρους • Απαραίτητη η μέτρηση της κίνησης • Κατάλληλα για εφαρμογές τύπου διαδρόμου (corridor) • Απαιτούν υψηλή χρηματοδότηση αλλά προσφέρουν τα περισσότερα οφέλη σε σχέση με όλα τα υπόλοιπα συστήματα

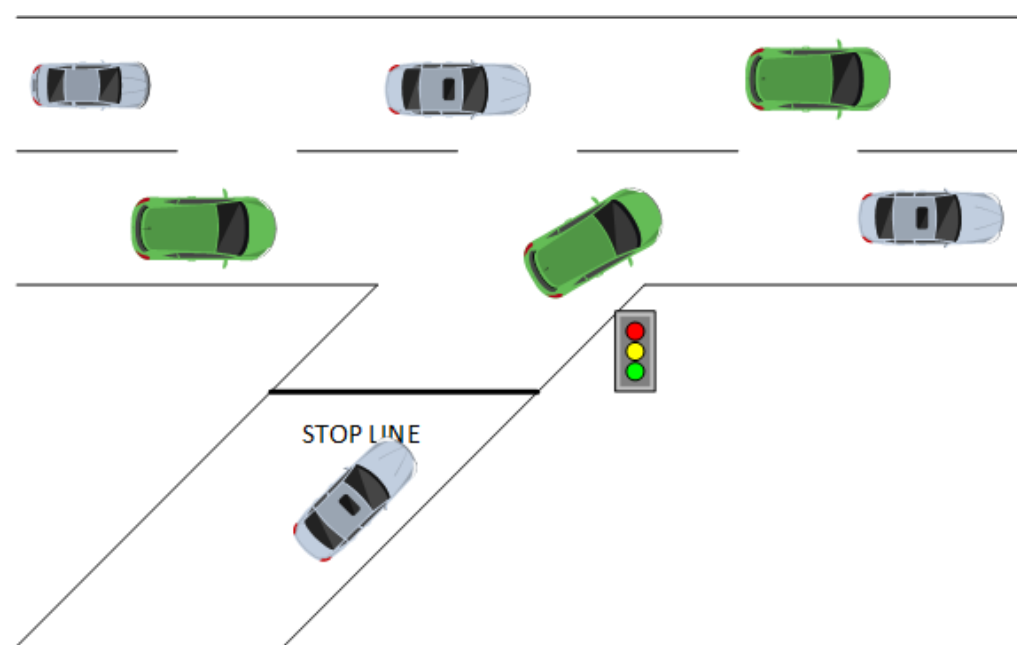
Πίνακας 2. Σύγκριση συστημάτων ελεγχόμενης πρόσβασης

Ένα σύστημα ελεγχόμενης πρόσβασης όπως φαίνεται και από τον παραπάνω πίνακα μπορεί να είναι είτε δυναμικό, είτε προκαθορισμένου χρόνου. Έτσι η σηματοδότηση μπορεί να λειτουργεί με βάση τα κυκλοφοριακά δεδομένα δυναμικά ή με σταθερούς χρόνους. Στα δυναμικά συστήματα η λειτουργία του σηματοδότη καθορίζεται από τα τρέχοντα κυκλοφοριακά δεδομένα στον αυτοκινητόδρομο. Έτσι όταν ο αυτοκινητόδρομος δεν έχει μεγάλη συγκέντρωση οχημάτων ο σηματοδότης έχει πράσινη ένδειξη ενώ σε αντίθετη περίπτωση η ένδειξη είναι κόκκινη. Τα συστήματα προκαθορισμένου χρόνου ρυθμίζουν τη λειτουργία του σηματοδότη με μοναδικό κανόνα τον χρόνο λειτουργίας του έτσι ώστε να μην εισέρχονται στον αυτοκινητόδρομο πολυάριθμες ομάδες οχημάτων. Στην πράξη υπάρχουν και

μεικτά συστήματα τα οποία με βάση τις συνθήκες που επικρατούν στον αυτοκινητόδρομο ρυθμίζουν αντίστοιχα τον χρόνο που ο φωτεινός σηματοδότης επιτρέπει ή απαγορεύει την είσοδο στο κύριο ρεύμα κίνησης. Έτσι εάν η πυκνότητα στον αυτοκινητόδρομο είναι μικρή ο χρόνος που φωτεινός σηματοδότης φέρει κόκκινη ένδειξη απαγορεύοντας την είσοδο είναι μικρός και αντίστοιχα όταν η πυκνότητα είναι μεγάλη ο χρόνος αυτός αυξάνεται.

2.4. Ορισμός του Προβλήματος

Στην παρούσα διπλωματική θα μελετηθεί μεθοδολογία που αφορά το Ramp Metering με βασικό κριτήριο τη διατήρηση ομαλής ροής στον αυτοκινητόδρομο. Στην κατεύθυνση αυτή θα εξεταστεί αρχικά ένα τυχαίο τμήμα ενός αυτοκινητόδρομου το οποίο θα εμπεριέχει μία ράμπα εισόδου με ελεγχόμενη πρόσβαση μέσω φωτεινού σηματοδότη όπως φαίνεται στην παρακάτω σχηματική απεικόνιση.



Εικόνα 8. Ράμπα εισόδου με ελεγχόμενη πρόσβαση

Στο παραπάνω σχήμα φαίνεται μία ράμπα εισόδου με φωτεινό σηματοδότη ο οποίος ανάλογα με την ένδειξή του επιτρέπει ή όχι την πρόσβαση στον αυτοκινητόδρομο. Το ζητούμενο είναι να διατηρείται ομαλή ροή τόσο στο κύριο ρεύμα όσο και στη ράμπα και παράλληλα το μέγεθος της ουράς να διατηρείται σε λογικά επίπεδα επάνω σε αυτό. Η παραπάνω σχηματική απεικόνιση αποτελεί μία αρχική ιδέα για το πρόβλημα και δεν αναπαριστά την πραγματικότητα, ωστόσο αποτελεί μία απεικόνιση πιλότο για τη διατύπωση και την υλοποίηση των αλγορίθμων επίλυσης. Η συγκεκριμένη αναπαράσταση μοντελοποιείται μαθηματικά με τις εξισώσεις που θα παρουσιαστούν σε επόμενο κεφάλαιο.

Μετά την πρώτη υλοποίηση του αρχικού αλγόριθμου η πορεία που θα ακολουθήσει η παρούσα εργασία είναι να εισάγει επιπλέον παραμέτρους, να προστεθούν λωρίδες στον

αυτοκινητόδρομο, να εισαχθούν περισσότερες της μίας ράμπας και να προσαρμοστούν τα δεδομένα όσο πιο κοντά γίνεται στην πραγματικότητα.

Όλα τα σενάρια που θα υλοποιηθούν θα παρουσιαστούν αναλυτικά και με κατάλληλα διαγράμματα ροής έτσι ώστε να γίνει κατανοητή απόλυτα η πορεία που ακολουθήθηκε και το πως κάθε παράμετρος που εντάχθηκε στο πρόβλημα επηρέασε τον υφιστάμενο αλγόριθμο. Κατόπιν της υλοποίησης όλων των πιθανών σεναρίων θα παρουσιαστούν αναλυτικά αποτελέσματα για κάθε μέθοδο υλοποίησης που θα αφορούν:

- Την πυκνότητα κίνησης συνολικά σε όλες τις λωρίδες στο κύριο ρεύμα
- Τη μέση ταχύτητα κίνησης
- Το πλήθος των αυτοκινήτων που εισήλθαν στον αυτοκινητόδρομο από τις ράμπες
- Τον χρόνο εκτέλεσης του αλγορίθμου
- Τις επαναλήψεις που εκτελέστηκαν

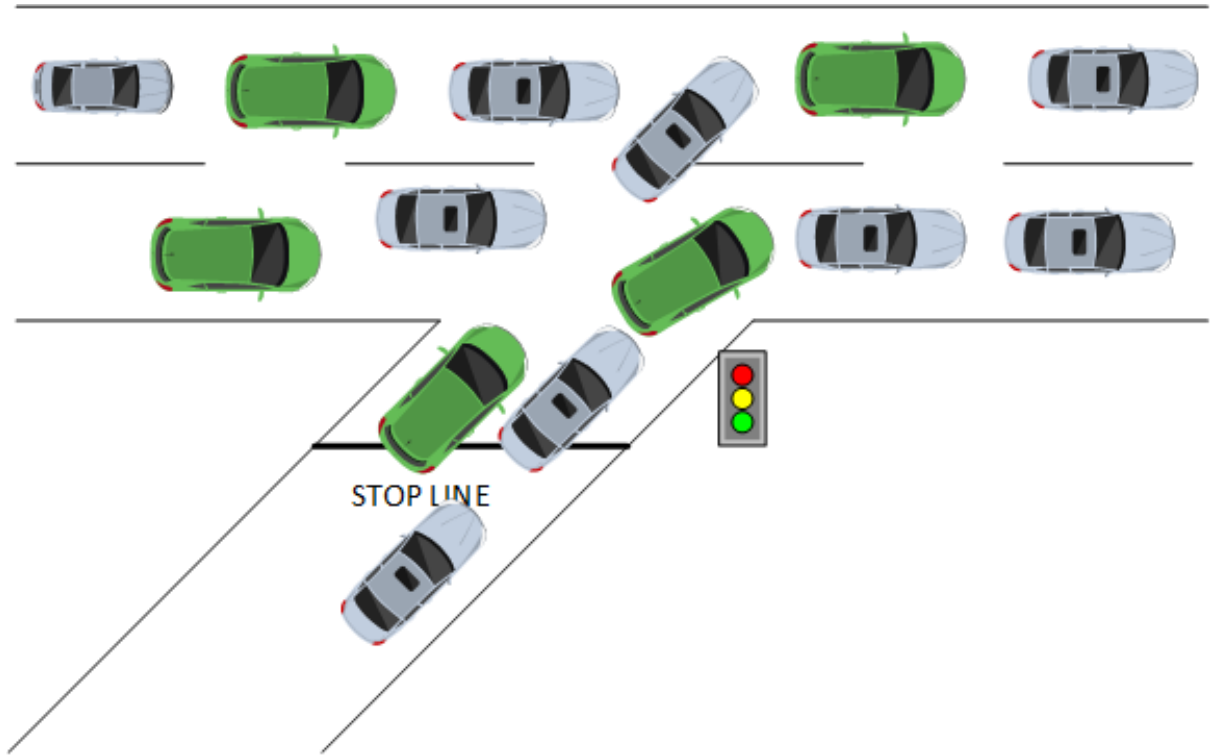
Κάθε υλοποίηση ωστόσο ενέχει τον κίνδυνο να δημιουργηθούν προβλήματα στην κίνηση των οχημάτων τόσο στο κύριο ρεύμα όσο και στη ράμπα τα οποία παρουσιάζονται στις παρακάτω παραγράφους.

2.4.1. Πιθανά προβλήματα

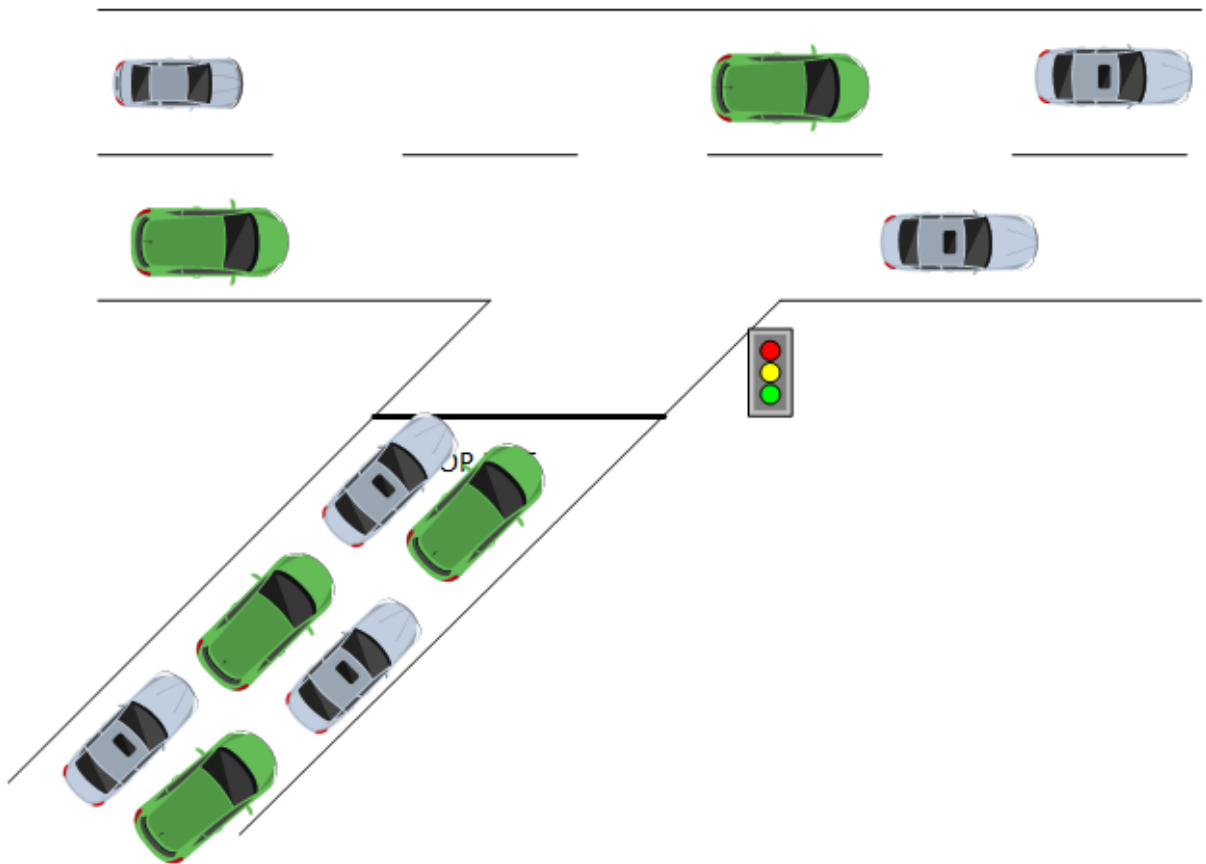
Κατά την υλοποίηση κάθε αλγορίθμου διαχείρισης κίνησης μέσω Ramp Metering σε όλο το εύρος των σεναρίων υπάρχει μεγάλη πιθανότητα να προκύψουν δύο (2) βασικά προβλήματα:

1. Κυκλοφοριακή συμφόρηση στο κύριο ρεύμα του αυτοκινητόδρομου λόγω της αυξημένης ροής εισόδου από τη ράμπα
2. Μεγάλο μέγεθος ουράς επάνω στη ράμπα λόγω της μικρής επιτρεπόμενης διέλευσης οχημάτων προς το κύριο ρεύμα

Στις επόμενες δύο (2) εικόνες φαίνονται σχηματικά και τα δύο (2) προβλήματα. Η εξάλειψή τους σε όλες τις πιθανές συνθήκες είναι σχεδόν αδύνατη ωστόσο στόχος των μεθόδων που θα παρουσιαστούν στην παρούσα εργασία είναι η κατά το δυνατόν σπανιότερη εμφάνισή τους ή η ισορροπία μεταξύ των δύο αυτών ακραίων καταστάσεων. Έτσι η πρόσβαση στον αυτοκινητόδρομο θα πραγματοποιείται με ικανοποιητικό ρυθμό χωρίς να δημιουργείται ουρά επάνω στη ράμπα ενώ στο κύριο ρεύμα η πυκνότητα και κατ' επέκταση η μέση ταχύτητα κίνησης θα διατηρείται στα επιθυμητά όρια εξασφαλίζοντας έτσι την ομαλή ροή και την αποφυγή αύξησης του χρόνου μετακίνησης σε ακραία επίπεδα.



Εικόνα 9. Πρόβλημα 1 - Συμφόρηση στο κύριο ρεύμα



Εικόνα 10. Πρόβλημα 2 - Μεγάλη ουρά στη ράμπα

Προκειμένου να αποφευχθεί ή να περιοριστεί η εμφάνιση των παραπάνω αστοχιών θα εισαχθούν κατάλληλες προγραμματιστικές δικλίδες ασφαλείας οι οποίες θα αλλάζουν τη συμπεριφορά του αλγορίθμου ώστε να περιορίζει τα παραπάνω προβλήματα με κόστος τη βέλτιστη λειτουργία του. Ωστόσο πολλές φορές σε ειδικές καταστάσεις όπως π.χ. ένα ατύχημα, έργα, στον δρόμο, κ.α. είναι σχεδόν ανέφικτο να μην υπάρχει συμφόρηση. Έτσι σε τέτοιες περιπτώσεις προκειμένου να μην υπάρξει ταυτόχρονη εμφάνιση και των δύο προβλημάτων συνήθως ελαχιστοποιείται ή διακόπτεται τελείως η ροή εισόδου από τη ράμπα στο κύριο ρεύμα. Με αυτόν τον τρόπο θα υπάρξει σίγουρα αυξημένη ουρά στη ράμπα αλλά έστω και αργά τα αυτοκίνητα θα κινούνται εντός του αυτοκινητόδρομου και έστω και αργά κάποια θα εισέρχονται σε αυτόν. Εάν όμως επιτρέψουμε στη ράμπα να εισέρχονται αυτοκίνητα συνεχώς τότε σύντομα θα επέλθει πλήρωση του κυρίως ρεύματος με αποτέλεσμα να μην μπορούν να κινηθούν αυτοκίνητα σε αυτό και ταυτόχρονα θα μεγαλώνει η ουρά στη ράμπα με αποτέλεσμα να οδηγηθούμε στη χειρότερη δυνατή κατάσταση. Επισημαίνεται όμως ότι αυτή η κατάσταση αποτελεί μία εξαίρεση η οποία εάν εμφανιστεί αυτό θα γίνει σε εξαιρετικές περιπτώσεις.

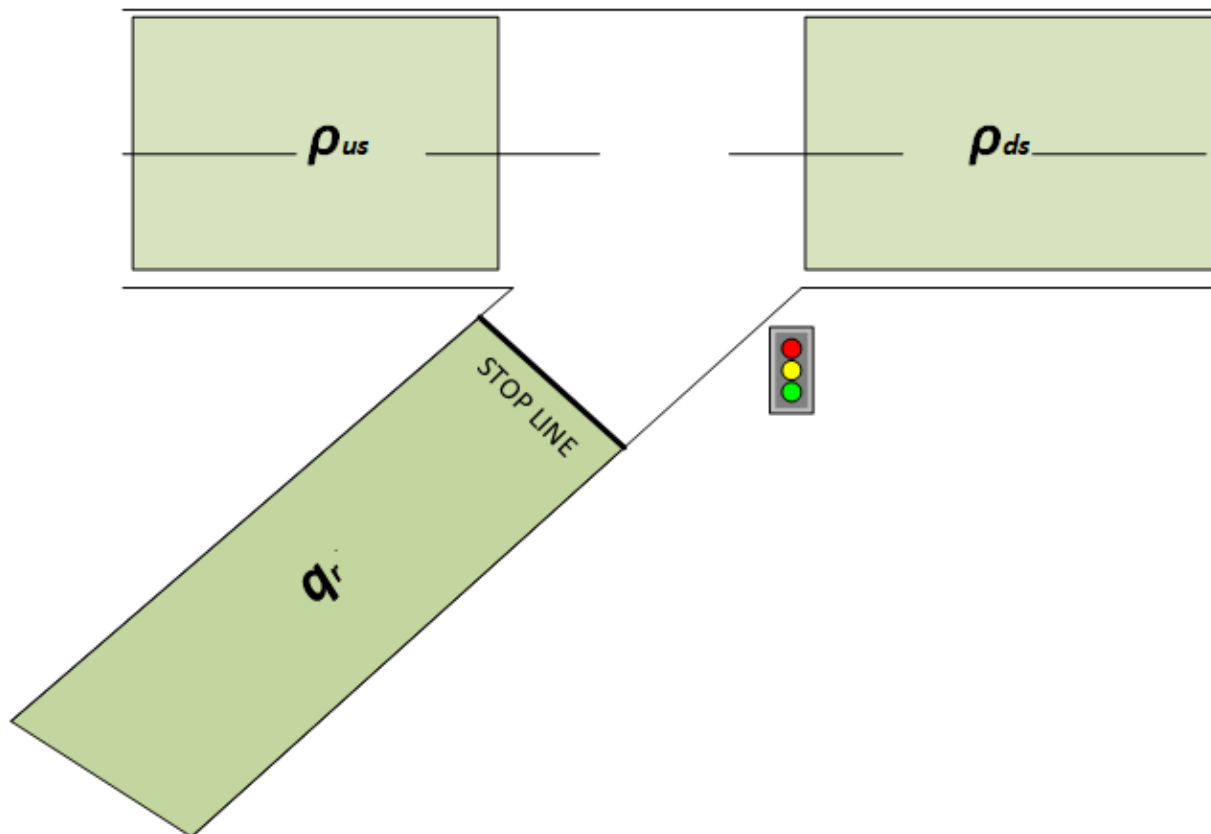
2.4.2. Μεταβλητές κατάστασης της μεθόδου

Η υλοποίηση των αλγορίθμων της παρούσας εργασίας πρέπει να λάβουν υπόψη τους μια σειρά μεταβλητών κατάστασης και παραδοχών, που θα αναφερθούν και σε επόμενα κεφάλαια, έτσι ώστε να γίνει ορθή και λειτουργική μοντελοποίηση του προβλήματος καθώς μέσω αυτών θα εκφραστούν τα επιθυμητά αποτελέσματα αλλά και οι περιορισμοί που επιβάλλονται.

Πιο συγκεκριμένα, οι μεταβλητές κατάστασης που μας ενδιαφέρουν σε ένα τέτοιο πρόβλημα είναι οι:

1. Η πυκνότητα των οχημάτων μετά τη ράμπα εισόδου, η οποία στο εξής θα συμβολίζεται με ρ_{ds}
2. Η πυκνότητα των οχημάτων πριν τη ράμπα εισόδου, η οποία στο εξής θα συμβολίζεται με ρ_{us}
3. Το μήκος της ουράς επάνω στη ράμπα εισόδου, το οποίο στο εξής θα συμβολίζεται με q_r
4. Η μέση ταχύτητα των οχημάτων στον αυτοκινητόδρομο, η οποία θα συμβολίζεται στο εξής με V

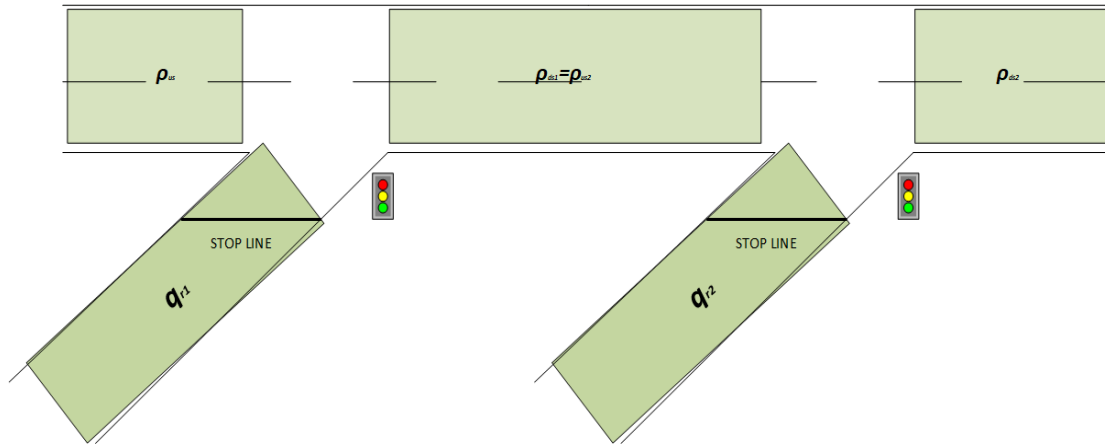
Όλα τα παραπάνω πλην της μέσης ταχύτητας που αφορά όλο το τμήμα του αυτοκινητόδρομου φαίνονται σχηματικά στην παρακάτω εικόνα:



Εικόνα 11. Μεταβλητές του συστήματος

Με βάση την παραπάνω εικόνα καταλαβαίνουμε ότι η πυκνότητα ανάντη και κατόντη της ράμπας πρέπει να διατηρείται σε κάποια επιτρεπτά όρια και είναι αυτές που πληροφορούν το σύστημα για το εάν υπάρχει συμφόρηση τόσο μετά όσο και πριν τη ράμπα. Μετά τη ράμπα ουσιαστικά έχουμε το αποτέλεσμα από την ενέργεια του φωτεινού σηματοδότη, έτσι εάν αυτή η πυκνότητα είναι αυξημένη τότε ο ρυθμός διέλευσης ίσως είναι μεγαλύτερος από αυτόν που πρέπει να είναι. Εάν τώρα και η πυκνότητα ρ_{us} είναι αυξημένη τότε έχουμε εκτεταμένο πρόβλημα καθώς η συμφόρηση έχει επεκταθεί σε όλο το κύριο ρεύμα. Παράλληλα με αυτές τις δύο μεταβλητές λαμβάνεται υπόψη και η παράμετρος q_r καθώς εάν ο χρόνος παραμονής κάθε οχήματος στη ράμπα είναι μεγάλος τότε αυτός ο αριθμός θα αυξάνεται συνεχώς.

Η παραπάνω περιγραφή που αφορά μία ράμπα συστήματος μπορεί να επεκταθεί και για δύο (2) ή περισσότερες ράμπες. Το τελικό ζητούμενο πάντα θα είναι η πυκνότητα του αυτοκινητόδρομου μετά τον έλεγχο εισόδου οχημάτων από τις ράμπες. Έτσι η κάθε ράμπα θα πρέπει να συνεργάζεται και να λαμβάνει υπόψη της τις μεταβολές που ενδεχομένως να προκαλεί στο σύστημα η είσοδος οχημάτων από τις ράμπες που βρίσκονται πριν και μετά από αυτή. Οι μεταβλητές που αναφέρθηκαν παραπάνω ισχύουν και στις περιπτώσεις με περισσότερες ράμπες όμως το τελικό ζητούμενο/στόχος του συστήματος είναι διατήρηση της πυκνότητας μετά και τη τελευταία ράμπα στα επιθυμητά όρια.



Εικόνα 12. Απεικόνιση με δύο ράμπες

Επισημαίνεται ότι η πυκνότητα αποτελεί τη μεταβλητή έλεγχου του συστήματος που θα αναπτυχθεί και αυτή είναι που θα καθορίσει τον έλεγχο που θα πραγματοποιείται. Το μήκος της ουράς επάνω στη ράμπα αποτελεί γενικότερα μία μεταβλητή η οποία χρησιμοποιείται έτσι ώστε να αποφεύγεται συνωστισμός στη ράμπα εφόσον ξεπεράσει κάποιο όριο ωστόσο στην παρούσα εργασία δεν αποτέλεσε περιορισμό για το σύστημα που μελετάται. Επιπλέον, το μελετώμενο σύστημα παρέχει πληροφορία για την πληρότητα της βοηθητικής λωρίδας εισόδου η οποία επηρεάζει τον έλεγχο και άρα την ένδειξη του φωτεινού σηματοδότη.

2.4.3. Ενέργειες του συστήματος

Με βάση τις μεταβλητές κατάστασης που έχουν περιγραφεί παραπάνω το σύστημα αποφασίζει πώς θα ενεργήσει σχετικά με την ένδειξη του φωτεινού σηματοδότη. Αυτή η απόφαση αφορά στον χρόνο που ο φωτεινός σηματοδότης θα επιτρέπει ή θα απαγορεύει τη διέλευση οχημάτων. Πιο συγκεκριμένα μπορεί να οριστεί ένα προκαθορισμένο σύνολο ενεργειών (δηλαδή χρόνων σε δευτερόλεπτα ή λεπτά) κατά το οποίο ο φωτεινός σηματοδότης θα έχει κόκκινη ένδειξη καθώς και η διάρκεια της πράσινης ένδειξης. Έτσι με βάση τις μεταβλητές εισόδου το σύστημα θα διαλέγει τον αποδοτικότερο χρόνο από το προκαθορισμένο αυτό σύνολο χρόνων.

Μία άλλη μέθοδος προσέγγισης αφορά τον δυναμικό προσδιορισμό του παραπάνω χρόνου αφού το σύστημα λάβει υπόψη του όλες τις μεταβλητές εισόδου. Ο κίνδυνος που ενέχει από τη συγκεκριμένη μέθοδο έχει να κάνει με την υστέρηση που παρουσιάζουν τα δεδομένα και το σύστημα συνολικά και έτσι πολλές φορές μπορεί να οδηγήσει σε λάθος εκτίμηση της κατάστασης και άρα σε προβληματική εφαρμογή της ελεγχόμενης πρόσβασης. Ωστόσο εάν το σύστημά μας διαθέτει δεδομένα πραγματικού χρόνου και έχει τη δυνατότητα να προσαρμόζει ταχύτατα τον χρόνο αυτό, τότε η συγκεκριμένη μέθοδος μπορεί να οδηγήσει σε μία πολύ αποδοτική διαχείριση της ελεγχόμενης πρόσβασης σε σχέση με τη μέθοδο του συνόλου προκαθορισμένων χρόνων. Στην παρούσα εργασία θα δημιουργηθεί μεγάλο πλήθος τυχαίων

δεδομένων τα οποία όμως θα είναι κατά το δυνατόν κοντά στην πραγματικότητα έτσι ώστε το σύστημα να εκπαιδευτεί κατάλληλα και να ανταποκρίνεται με ταχύτητα στην προσαρμογή των μεταβλητών ελέγχου. Η ταχύτητα επεξεργασίας και προσαρμογής του συστήματος θα επιτευχθεί με τη χρήση νευρωνικού δικτύου το οποίο αναπτύχθηκε για τις ανάγκες της συγκεκριμένης εργασίας. Το νευρωνικό αυτό δίκτυο θα δεχθεί ένα αρκετά μεγάλο πλήθος δεδομένων τα οποία λόγω της τυχαίας φύσης τους και του αριθμού τους (brute force) θεωρούνται έγκυρα και κατάλληλα για την ορθή εκπαίδευση του νευρωνικού δικτύου και την αποτελεσματικότητά του. Οι ιδιότητες, η δομή και η λειτουργία των νευρωνικών δικτύων περιγράφεται σε επόμενο κεφάλαιο.

2.4.4. Συνάρτηση ανταμοιβής (Reward Function) – Στόχος

Κατά τον σχεδιασμό του συστήματος ο βασικός στόχος είναι η μείωση του χρόνου μετακίνησης για όλους τους χρήστες του αυτοκινητόδρομου. Με βάση τη διεθνή βιβλιογραφία καθώς και τη θεωρία που αφορά την κυκλοφορία οχημάτων ο μέγιστος ρυθμός διέλευσης ο οποίος αφορά τη μέγιστη ροή συναντάται όταν οι τιμές της πυκνότητας είναι κοντά στο κρίσιμο σημείο. Έτσι, γίνεται εύκολα αντιληπτό ότι το σύστημα ελέγχου έχει ως στόχο να διατηρεί τα ρ_{ds} και ρ_{us} κοντά στο κρίσιμο σημείο έτσι ώστε να επιτυγχάνεται η μέγιστη διέλευση οχημάτων. Στον παραπάνω στόχο ωστόσο πρέπει να λαμβάνεται γενικότερα υπόψη και ο περιορισμός που προκύπτει από το μήκος της ουράς στη ράμπα q_r .

Μία από τις πιο διαδεδομένες συναρτήσεις ανταμοιβής είναι αυτή της μεθόδου ALINEA η οποία αναφέρθηκε σε προηγούμενο κεφάλαιο, και όπως αυτή περιγράφεται από τους Μ. Παπαγεωργίου, Η. Hadj-Salem, and J-M. Blosseville στο “*ALINEA: A local feedback control law for on-ramp metering*”, Transportation Research Record, vol. 1320, σελ. 58–64, και φαίνεται παρακάτω^[32]:

$$r(t) = -(\rho - \rho_{ds}(t))^2 \quad (5)$$

Όπου ρ η πυκνότητα που ρυθμίζει το σύστημα και ρ_{ds} η πυκνότητα που μετράται μετά τη ράμπα κατά τη διάρκεια του χρόνου t . Η παραπάνω σχέση υψώνεται στο τετράγωνο έτσι ώστε να ενισχύσει την αντίθεση μεγάλων αποκλίσεων από το επιθυμητό αποτέλεσμα (επιθυμητή πυκνότητα).^[23]

2.5. Μαθηματική Μοντελοποίηση Αυτοκινητόδρομου

2.5.1. Μοντέλο κίνησης αυτοκινητόδρομου

Η διατύπωση ενός μαθηματικού μοντέλου για την προσομοίωση της κίνησης στον αυτοκινητόδρομο είναι απαραίτητη για την κατανόηση και υλοποίηση των αλγορίθμων που θα παρουσιαστούν σε επόμενο κεφάλαιο. Αρχικά θεωρούμε ότι έχουμε ένα αυτοκινητόδρομο πολλαπλών λωρίδων l και με μόνο ένα σημείο εισόδου (μία ράμπα). Θεωρώντας ότι τη χρονική

στιγμή k σε ένα συγκεκριμένο τμήμα του αυτοκινητόδρομου έχουμε ροή $q_u(k)$ η οποία εκφράζεται σε οχήματα ανά ώρα ανά λωρίδα ανάντη της ράμπας, $r(k)$ τα οχήματα ανά ώρα που κινούνται στη ράμπα και $q(k)$ τα οχήματα ανά ώρα ανά λωρίδα που κινούνται κατάντη της ράμπας, τότε ο αριθμός N των οχημάτων που κινείται στον αυτοκινητόδρομο τη στιγμή $k+1$ όπως αναφέρουν και οι H. Michael Zhang, Stephen G. Ritchie είναι:^[31]

$$N(k+1) = N(k) + \Delta t(lq_u(k) - lq(k) + r(k)) \quad (6)$$

Έστω ότι κάθε όχημα τη χρονική στιγμή k κινείται με ταχύτητα $U_i(k)$ τότε η μέση ταχύτητα $V(k)$ για το συγκεκριμένο κομμάτι του αυτοκινητόδρομου που εξετάζεται είναι:

$$V(k) = \frac{\sum U_i}{N(k)} \quad (7)$$

Όπου:

- $\sum U_i$ είναι το άθροισμα των ταχυτήτων των οχημάτων τη χρονική στιγμή k
- $N(k)$ ο αριθμός των αυτοκινήτων τη χρονική στιγμή k

Η πυκνότητα που υπάρχει σε ένα τμήμα του αυτοκινητόδρομου τη χρονική στιγμή k ορίζεται ως:

$$d(k) = N(k)/(l\Delta x) \quad (8)$$

Όπου:

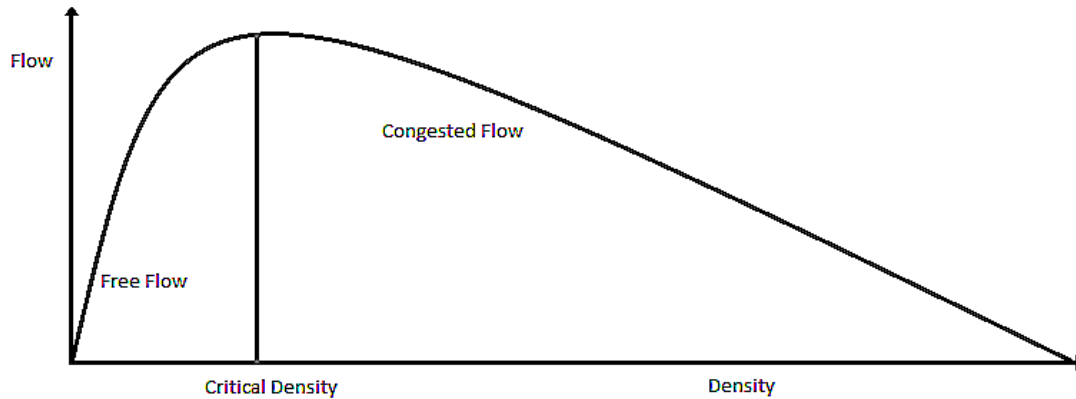
- $d(k)$ είναι η πυκνότητα τη χρονική στιγμή k
- $N(k)$ ο αριθμός των αυτοκινήτων τη χρονική στιγμή k
- l ο αριθμός των λωρίδων του αυτοκινητόδρομου
- Δx το μήκος του τμήματος του αυτοκινητόδρομου που εξετάζεται

Το γινόμενο $l*\Delta x$ ονομάζεται και χωρητικότητα του αυτοκινητόδρομου, μπορούμε να το πάρουμε ως δεδομένο στην υπό μελέτη εφαρμογή και συμβολίζεται με C .

Έτσι από τις παραπάνω σχέσεις για την πυκνότητα και τον αριθμό των αυτοκινήτων έχουμε το εξής:

$$d(k+1) = d(k) + \frac{\Delta t}{\Delta x}(q_u(k) - q(k) + \frac{r(k)}{l}) \quad (9)$$

Με βάση την καθημερινή εμπειρία καθώς και διάφορες μελέτες που έχουν γίνει με κυκλοφοριακά δεδομένα υπάρχει άμεση σχέση μεταξύ της πυκνότητας των αυτοκινήτων και της ροής αυτών μέσα σε ένα αυτοκινητόδρομο. Η σχέση αυτή συνήθως αποδίδεται με το παρακάτω διάγραμμα:



Εικόνα 13. Διάγραμμα συγκέντρωσης-ροής αυτοκινήτων σε αυτοκινητόδρομο^[30]

Από το παραπάνω διάγραμμα γίνονται αντιληπτά τα εξής:

- Η ροή αυξάνεται όσο η πυκνότητα είναι κάτω από την κρίσιμη τιμή (critical density)
- Η ροή αρχίζει και μειώνεται από την κρίσιμη τιμή της πυκνότητας και μετά με αποτέλεσμα όσο συνεχίζει να αυξάνεται η πυκνότητα να αυξάνεται και η κυκλοφοριακή συμφόρηση
- Από ένα σημείο και μετά η πυκνότητα έχει αυξηθεί τόσο πολύ που η ροή είναι πρακτικά μηδέν

Το παραπάνω διάγραμμα μπορεί εκφραστεί και ως μέση ταχύτητα σε σχέση με την πυκνότητα ισχύοντας ακριβώς οι ίδιες παρατηρήσεις με αυτές που αναφέρθηκαν παραπάνω καθώς η μέση ταχύτητα ταξιδιού σε έναν αυτοκινητόδρομο είναι άρρηκτα συνδεδεμένη με την πυκνότητα σε αυτόν. Η αυξημένη πυκνότητα σημαίνει μεγαλύτερο ποσοστό κάλυψης του δρόμου και άρα μικρότερες ταχύτητες κίνησης.

Η εξίσωση (8) καθώς και το διάγραμμα ροής/μέσης ταχύτητας – πυκνότητας αποτελούν την έκφραση του διάσημου μοντέλου LWR (Lighthill, Whitham, Richards) το οποίο αποτελεί μία πολύ καλή προσέγγιση για συστήματα που μελετούν προβλήματα ελέγχου κίνησης.

Στην εξίσωση (9) μπορούμε να εκφράσουμε τη ροή συναρτήσει της πυκνότητας ως $q=f(\rho)$ και να ορίσουμε ως $\alpha=\Delta t/\Delta x$ άρα η (9) μετασχηματίζεται σε:

$$\rho(k+1) = \rho(k) - \alpha f(\rho(k)) + \alpha q_u(k) + \frac{\alpha r(k)}{l} \quad (10)$$

Στην εξίσωση (9) το $\rho(k)$ αποτελεί τη μεταβλητή κατάστασης, το $r(k)$ αποτελεί τη μεταβλητή ελέγχου και το q_u αποτελεί τον θόρυβο για το σύστημα καθώς δεν μπορεί να ελεγχθεί από αυτό.^[31]

Όλες οι παραπάνω εξισώσεις αφορούν στη μοντελοποίηση του αυτοκινητόδρομου και στην περίπτωση μίας (1) ράμπας. Στην παρούσα εργασία θα μελετηθούν περισσότερες της μίας ράμπες και έτσι οι εξισώσεις (6), (9) και (10) μετασχηματίζονται κατ' αντιστοιχία όπως φαίνεται παρακάτω:

$$N(k+1) = N(k) + \Delta t(lq_u(k) - lq(k) + r_1(k) + r_2(k) + \dots + r_n(k)) \quad (11)$$

$$\rho(k+1) = \rho(k) + \frac{\Delta t}{\Delta x}(q_u(k) - q(k) + \frac{r_1(k)+r_2(k)+\dots+r_n(k)}{l}) \quad (12)$$

$$\rho(k+1) = \rho(k) - \alpha f(\rho(k) + \alpha q_u(k) + \frac{\alpha(r_1(k)+r_2(k)+\dots+r_n(k))}{l}) \quad (13)$$

2.5.2. Στόχος του μοντέλου Ramp Metering

Όπως έχει ήδη αναφερθεί σε προηγούμενα κεφάλαια ο στόχος ενός μοντέλου ελέγχου πρόσβασης μέσω μίας ράμπας είναι να αξιοποιείται στον μέγιστο βαθμό ο αυτοκινητόδρομος διατηρώντας την πυκνότητα σταθερή εντός κάποιου προκαθορισμένου εύρους. Η πυκνότητα που πρέπει να ελεγχθεί αφορά το σύνολο του τμήματος του αυτοκινητοδρόμου που εξετάζεται και όχι μεμονωμένα τμήματα αυτού. Έτσι εάν έχουμε περισσότερες της μίας ράμπας τότε θα πρέπει η ολική πυκνότητα όλου του τμήματος του αυτοκινητοδρόμου που καλύπτουν οι ράμπες να εξεταστεί πριν και μετά από αυτές. Με βάση λοιπόν αυτή την προσέγγιση ο τελικός στόχος του μοντέλου είναι μία πυκνότητα η οποία ικανοποιεί το παρακάτω:

$$\rho \in [\rho_c - \varepsilon, \rho_c + \varepsilon] \quad (14)$$

Όπου:

- ρ η πυκνότητα αποτέλεσμα
- ρ_c η κρίσιμη πυκνότητα μετά την οποία πέφτει η ροή
- ε μία παράμετρος που καθορίζει την επιτρεπτή απόκλιση από την κρίσιμη πυκνότητα

Με βάση τα παραπάνω το άρτιο αποτέλεσμα επιτυγχάνεται όταν η πυκνότητα διατηρείται σταθερά στην κρίσιμη τιμή.

2.6. Νευρωνικά Δίκτυα και Ramp Metering

Παραπάνω έγινε μία εισαγωγή και αναλύθηκε το πρόβλημα που πρέπει να επιλυθεί καθώς και παρουσιάστηκε η μαθηματική μοντελοποίηση της κίνησης ενός αυτοκινητόδρομου. Επιπλέον αναφέρθηκαν κάποιες παράμετροι που πρέπει να ληφθούν υπόψη προκειμένου να δημιουργηθεί ένας αλγόριθμος επίλυσης. Για παράδειγμα στον αλγόριθμο ALINEA που έχει ήδη αναφερθεί υπάρχει ένας ρυθμιστικός παράγοντας K_R ο οποίος μπορεί να αντικατασταθεί με ένα νευρωνικό δίκτυο το οποίο με κατάλληλα δεδομένα εισόδου και μετά από εκπαίδευση αυτού μπορεί να παράξει μία ή περισσότερες εξόδους οι οποίες θα αντιστοιχίζονται στην επιθυμητή ενέργεια του συστήματος.

Η επιλογή της ανάπτυξης μεθοδολογίας νευρωνικού δικτύου που αφορά την εφαρμογή Ramp Metering στη συγκεκριμένη εργασία έγινε διότι αποτελεί μία σύγχρονη προσέγγιση η οποία εντάσσεται ακόμα σε τομέα ενεργής έρευνας στην επιστημονική κοινότητα. Επιπλέον το γεγονός ότι εξετάστηκαν σενάρια υλοποίησης με δύο ράμπες αυξάνοντας έτσι την πολυπλοκότητα του προβλήματος και καθιστώντας αρκετά αναποτελεσματική τη χρήση απλών

αλγορίθμων ελέγχου οδήγησαν στη χρήση ενός νευρωνικού δικτύου το οποίο διαχειρίζεται καλύτερα προβλήματα με μεγάλο πλήθος δεδομένων και υψηλή πολυπλοκότητα. Επιπρόσθετα, η δυνατότητα ταχύτητας απόκρισης που έχουν τα συστήματα νευρωνικών δικτύων εφόσον έχουν εκπαιδευτεί σωστά αποτελεί πολύ σημαντική παράμετρο για συστήματα διαχείρισης κίνησης καθώς τα δεδομένα κίνησης παρουσιάζουν πολύ μεγάλες μεταβολές σε μικρά χρονικά διαστήματα.

Μία αρχική προσέγγιση και οδηγός για την επιλογή της συγκεκριμένης μεθοδολογίας αποτέλεσε η μελέτη της δημοσίευσης *“Decentralised reinforcement learning for ramp metering and variable speed limits on highways”* των Thorsten Schmidt-Dumont και Jan H van Vuuren. Στη προαναφερθείσα δημοσίευση γίνεται μία ανάλυση μεταξύ άλλων της μεθόδου του Ramp Metering και την υλοποίηση αυτής με εφαρμογή ενισχυμένης μάθησης (Reinforcement Learning). Σε συνέχεια λοιπόν της ιδέας που περιγράφεται στο συγκεκριμένο paper και δεδομένου ότι το Reinforcement Learning αποτελεί μία κατηγορία μηχανικής μάθησης όπως και τα νευρωνικά δίκτυα αναπτύχθηκε μεθοδολογία και αλγόριθμος για εφαρμογή Ramp Metering με χρήση νευρωνικού δικτύου.

Η δυσκολία σε μία τέτοια υλοποίηση ωστόσο έγκειται στην ανεύρεση μεγάλου πλήθους δεδομένων τα οποία διαδραματίζουν τον σημαντικότερο ρόλο στην εκπαίδευση αυτού και άρα στη λειτουργία του νευρωνικού. Παρακάτω γίνεται μία σύντομη περιγραφή των νευρωνικών δικτύων και πώς αυτά μπορούν να ενταχθούν στη μεθοδολογία ελέγχου πρόσβασης μέσω ανισόπεδου κόμβου σε αυτοκινητόδρομο.

2.6.1. Γενικά στοιχεία για τα Νευρωνικά Δίκτυα (Neural Networks – NN)

2.6.1.1. Βιολογικό μοντέλο και έμπνευση

Ο ανθρώπινος εγκέφαλος αποτελείται κατά κύριο λόγο από ένα ευρύ φάσμα νευρώνων, οι οποίοι είναι μαζικά διασυνδεδεμένοι μεταξύ τους με χιλιάδες διασυνδέσεις ανά νευρώνα. Κάθε νευρώνας είναι ένα εξειδικευμένο κύτταρο το οποίο έχει τη δυνατότητα μετάδοσης ενός ηλεκτροχημικού σήματος. Ο νευρώνας έχει μια διακλαδωτική διάρθρωση εισροών, τους δενδρίτες, ένα κυτταρικό σώμα και μια διακλαδωτική δομή εκροών, τον άξονα. Οι άξονες ενός κυττάρου συνδέονται με τους δενδρίτες ενός άλλου, μέσω μιας σύναψης. Όταν, λοιπόν, ένας άξονας ενεργοποιηθεί, πυροδοτεί ένα ηλεκτροχημικό σήμα κατά μήκος του άξονα. Ένας νευρώνας εκτελεί αυτή τη διαδικασία μόνο όταν το συνολικό σήμα το οποίο λήφθηκε από τους δενδρίτες, υπερβεί ένα συγκεκριμένο επίπεδο, δηλαδή, το κατώφλι ενεργοποίησης (firing threshold).

Με βάση τα παραπάνω και σχεδόν σε πλήρη αναλογία δημιουργήθηκαν τα τεχνητά νευρωνικά δίκτυα τα οποία έχουν ως στόχο την προσομοίωση της λειτουργίας του εγκεφάλου για την επίλυση σύνθετων προβλημάτων αυξημένης πολυπλοκότητας. Η βασική διαφορά των

βιολογικών με τα τεχνητά νευρωνικά δίκτυα είναι ότι τα τεχνητά πρέπει να εκπαιδευτούν από το μηδέν έτσι ώστε να μπορέσουν να λάβουν μία απόφαση.^[34]

2.6.1.2. Εκπαίδευση Νευρωνικών Δικτύων

Μία από τις πιο βασικές ιδιότητες των Νευρωνικών Δικτύων είναι η ικανότητά τους για εκπαίδευση. Η εκπαίδευση αυτή επιτυγχάνεται μέσω της ανταλλαγής τιμών και βαρών, που αποσκοπεί στη βαθμιαία σύλληψη της πληροφορίας η οποία στη συνέχεια θα είναι διαθέσιμη προς ανάκτηση. Υπάρχουν, βέβαια, πολλοί αλγόριθμοι που η εφαρμογή τους έχει στόχο την προσαρμογή των τιμών των βαρών ενός Τεχνητού Νευρωνικού Δικτύου. Ο ορισμός και η ανάπτυξη αυτών των αλγορίθμων μάθησης είναι το βασικό στοιχείο έρευνας των τεχνητών νευρωνικών δικτύων. Η μάθηση μπορεί να είναι είτε με επίβλεψη είτε χωρίς επίβλεψη.^[34]

Μάθηση με επίβλεψη: Η μάθηση αυτή είναι μια διαδικασία η οποία συνδυάζει έναν εξωτερικό εκπαιδευτή και τη συνολική ή γενικευμένη πληροφορία. Κάποιες από τις μεθόδους οι οποίες συγκαταλέγονται σε αυτή την κατηγορία είναι η μάθηση με διόρθωση σφάλματος, η στοχαστική μάθηση, κ.λπ.. Παραδείγματα τα οποία αντιπροσωπεύουν τη μάθηση με επίβλεψη συμπεριλαμβάνουν αποφάσεις για το πότε θα πρέπει να σταματήσει η διαδικασία εκπαίδευσης, αποφάσεις αναφορικά με τη συχνότητα παρουσίας μίας τιμής στο δίκτυο, τα πρότυπα εκπαίδευσης και η παρουσίαση προόδου του δικτύου. Η μάθηση με επίβλεψη χωρίζεται σε δύο ακόμα κατηγορίες:

- στη δομική (structural) και
- στην προσωρινή (temporal) εκμάθηση.

Οι αλγόριθμοι οι οποίοι βρίσκονται στην πρώτη κατηγορία, χρησιμοποιούνται για την εύρεση της βέλτιστης σχέσης μεταξύ εισόδων και εξόδων για κάθε ξεχωριστό ζευγάρι προτύπων. Παραδείγματα της δομικής εκμάθησης αποτελούν η αναγνώριση και η κατηγοριοποίηση προτύπων, ενώ παραδείγματα της προσωρινής εκμάθησης η πρόβλεψη και ο έλεγχος.^[34]

Μάθηση χωρίς επίβλεψη: Οι αλγόριθμοι της εν λόγω μάθησης αναφέρονται ως αυτό-οργανώμενοι (self-organized) και είναι διαδικασίες οι οποίες δεν απαιτούν να είναι παρών ένας «εξωτερικός» δάσκαλος ή επιβλέπων. Βασίζονται, μάλιστα, μόνο σε τοπική πληροφορία καθ' όλη τη διάρκεια της εκπαίδευσης του Τεχνητού Νευρωνικού Δικτύου. Οι συγκεκριμένοι αλγόριθμοι οργανώνουν τα δεδομένα και ανακαλύπτουν τις σημαντικές συλλογικές τους ιδιότητες. Για παράδειγμα, αλγόριθμοι εκπαίδευσης χωρίς επίβλεψη είναι ο αλγόριθμος Hebbian, ο διαφορικός αλγόριθμος Hebbian και ο Min-Max αλγόριθμος.^[34]

Κατά κύριο λόγο οι περισσότερες διαδικασίες εκπαίδευσης είναι off-line. Όταν χρησιμοποιείται όλο το δείγμα προτύπων για την τροποποίηση των τιμών των βαρών, πριν την τελική χρήση του δικτύου ως εφαρμογή, τότε ονομάζεται off-line εκπαίδευση. Οι αλγόριθμοι εκπαίδευσης off-line έχουν την απαίτηση να βρίσκονται στην εκπαίδευση του δικτύου παρόντα όλα τα πρότυπα και συσχετίσεις. Το γεγονός αυτό αποκλείει την πιθανότητα εισαγωγής νέων

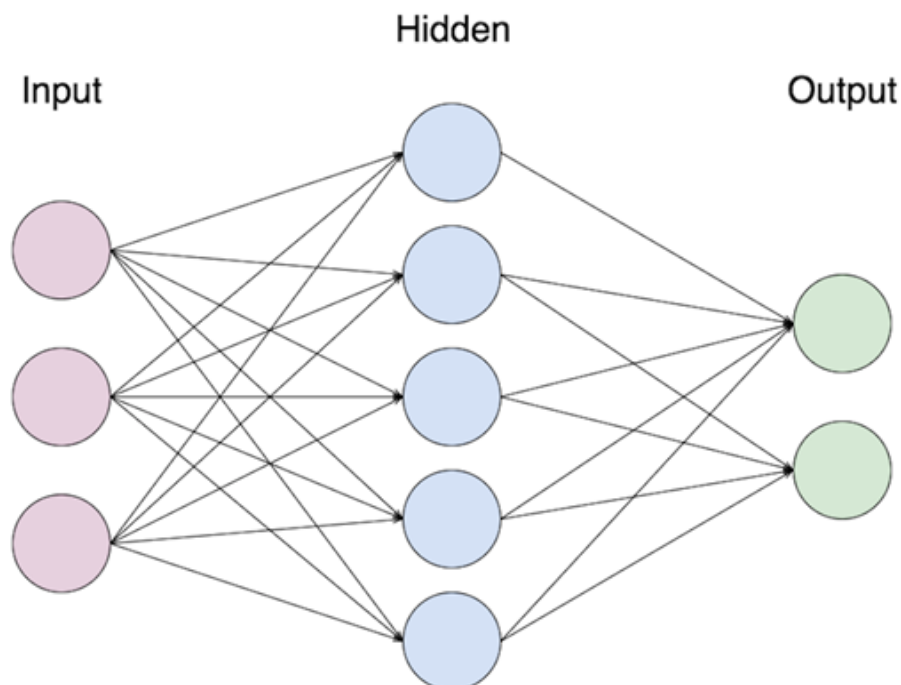
πληροφοριών μέσω νέων προτύπων. Βέβαια, υπάρχουν και Τεχνητά Νευρωνικά Δίκτυα τα οποία δεν αποκλείουν την εισαγωγή νέας πληροφορίας μετά την τελική τους μοντελοποίηση. Αν παρουσιαστεί ανάγκη εισαγωγής νέου προτύπου στο δίκτυο, μπορεί να γίνει απευθείας χωρίς τον κίνδυνο να χαθεί κανένα μέρος της αρχικής πληροφορίας. Το πλεονέκτημα των δικτύων που χρησιμοποιούν off-line διαδικασίες εκπαίδευσης επικεντρώνεται κυρίως στη δυνατότητα να δίνουν καλύτερες λύσεις σε δύσκολα προβλήματα.

Στην πράξη τις περισσότερες φορές υπάρχει μάθηση με επίβλεψη καθώς έχοντας ήδη τη γνώση για τη λειτουργία ενός συστήματος τροποποιούμε τις αποφάσεις του νευρωνικού δικτύου έτσι ώστε η έξοδος να είναι πιο κοντά στα πραγματικά δεδομένα.

Τα νευρωνικά δίκτυα συνήθως αποτελούνται από τα εξής

1. Επίπεδο εισόδων
2. Επίπεδο/-α νευρώνων (κρυφό επίπεδο – hidden layer)
3. Επίπεδο εξόδων

Στο επίπεδο εισόδων έχουμε όλα τα δεδομένα εισόδου που χρειάζονται για τον υπολογισμό των εξόδων. Το επίπεδο νευρώνων είναι το επίπεδο στο οποίο εφαρμόζονται οι αλγόριθμοι εκπαίδευσης με βάση τα δεδομένα που διαθέτει ένα νευρωνικό δίκτυο. Κάθε νευρώνας αποτελείται από έναν αθροιστή και μία συνάρτηση ενεργοποίησης ή μεταφοράς (activation/transfer function). Το επίπεδο εξόδων είναι το αποτέλεσμα που προκύπτει με βάση τις εισόδους και τους νευρώνες του δικτύου.^[34]



Εικόνα 14. Δομή τυπικού νευρωνικού δικτύου

Με βάση λοιπόν την παραπάνω εικόνα ο στόχος ενός νευρωνικού δικτύου είναι έχοντας κάποια δεδομένα να εκπαιδευτεί κατάλληλα ώστε να μπορεί να επιλύσει ένα

πρόβλημα. Το ζητούμενο σε κάθε περίπτωση είναι ο αλγόριθμος εκπαίδευσης του εκάστοτε δικτύου. Μία γενική εξίσωση που ισχύει για τα νευρωνικά δίκτυα είναι η παρακάτω:

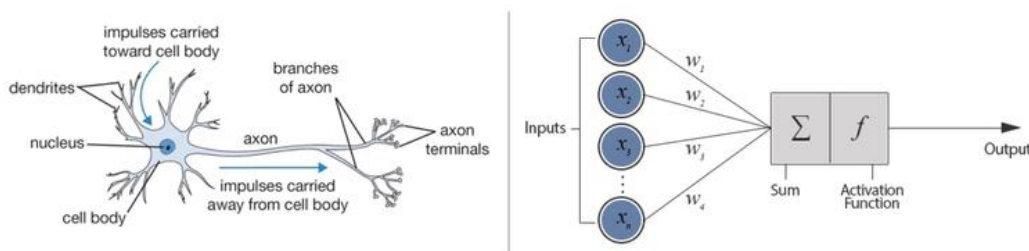
$$y = NN(x, \beta) \quad (15)$$

Όπου:

- $x \in R^n$ και αποτελεί το διάνυσμα εισόδων του νευρωνικού
- $y \in R^m$ και αποτελεί το διάνυσμα εξόδων του νευρωνικού
- $\beta \in R^q$ και αποτελεί το διάνυσμα σύνδεσης των βαρών

2.6.1.3. Perceptrons

Η απλούστερη μορφή νευρωνικού δικτύου είναι το δίκτυο Perceptron, όπως έχει επικρατήσει ο όρος στη διεθνή βιβλιογραφία, που αποτελείται από έναν και μόνο νευρώνα. Όπως ακριβώς τα βιολογικά νευρωνικά δίκτυα τα οποία έχουν δενδρίτες και άξονες έτσι και ο νευρώνας Perceptron έχει δενδρική δομή η οποία αποτελείται από διάφορες εισόδους οι οποίες συνδέονται με μία μοναδική έξοδο. Στην επόμενη εικόνα φαίνεται μία σύγκριση των δύο, βιολογικού νευρωνικού και perceptron:



Εικόνα 15. Αντιπαραβολή βιολογικού με τεχνητού νευρώνα

Με οδηγό την παραπάνω εικόνα έχουμε τα εξής:

- **Κόμβοι εισόδων (input nodes):** Κάθε κόμβος εισόδου μπορεί να έχει μία οποιαδήποτε αριθμητική τιμή σε όλο το εύρος \mathbb{R} των πραγματικών αριθμών.
- **Σύνδεσμοι (connections):** Αποτελούν ουσιαστικά τα βάρη με τα οποία συνδέονται οι εισοδοί. Κάθε βάρος w_i μπορεί να είναι οποιοσδήποτε πραγματικός αριθμός.
- **Άθροισμα κόμβων:** Έπειτα όλοι οι κόμβοι εισόδου πολλαπλασιάζονται με το αντίστοιχο βάρος τους και στη συνέχεια αθροίζονται. Άρα έχουμε:

$$y = f(\sum_{i=1}^n w_i * x_i) \quad (16)$$

- **Συνάρτηση μεταφοράς ή ενεργοποίησης (transfer/activation function):** Αφού σχηματιστεί το παραπάνω άθροισμα τότε δίνεται ως είσοδος σε μία συνάρτηση η οποία ονομάζεται συνάρτηση μεταφοράς ή ενεργοποίησης (transfer or activation function). Η συγκεκριμένη συνάρτηση σε μία απλουστευμένη της μορφή είναι μία ταυτοτική συνάρτηση η οποία ενεργοποιείται μόνο εάν το άθροισμα των εισόδων ξεπερνάει ένα κατώφλι. Για παράδειγμα μπορεί να οριστεί σαν κατώφλι το μηδέν και μόλις το

άθροισμα το ξεπεράσει τότε εκτελεί τον ρόλο της η συνάρτηση. Ωστόσο μπορεί να είναι οποιαδήποτε άλλη συνάρτηση όπως η εκθετική, η σιγμοειδής, λογαριθμική κ.α.

- **Κόμβος εξόδου (output node):** Μόλις η συνάρτηση ενεργοποιηθεί τότε παράγεται ο κόμβος εξόδου ο οποίος όπως γίνεται εύκολα αντιληπτό ότι συνδέεται με τις εισόδους μέσω της συνάρτησης μεταφοράς.
- **Κλίση (bias):** Τέλος, ο κόμβος εξόδου μπορεί να επηρεαστεί και να μετατοπιστεί ανάλογα με τις ανάγκες της εφαρμογής χρησιμοποιώντας την κλίση (bias). Η κλίση μπορεί να θεωρηθεί ως ένας ακόμα κόμβος εισόδου με το δικό του βάρος και ο οποίος είναι πάντα 1 και ουσιαστικά προστίθεται το βάρος στο συνολικό άθροισμα των γινομένων των κόμβων με τα βάρη τους. Εφόσον ληφθεί υπόψη το bias τότε η εξίσωση (16) μετασχηματίζεται όπως φαίνεται παρακάτω:

$$y = f(\sum_{i=1}^n w_i * x_i + b_i) \quad (17)$$

Με βάση λοιπόν τα όσα αναφέρθηκαν οι Perceptrons λειτουργούν σωστά μόνο με αριθμητικά δεδομένα και αυτό τους καθιστά κατάλληλους για κατηγοριοποίηση δεδομένων (classification) καθώς κάθε έξοδος συγκρινόμενη με έναν αριθμό κατώφλι (threshold) μπορεί να καταταχθεί σε μία ομάδα/κλάση.

Έτσι, λοιπόν μπορεί να δημιουργηθεί ένα πολύπλοκο νευρωνικό δίκτυο το οποίο θα αποτελείται από πολλούς perceptrons που η έξοδος κάθε ενδιάμεσου perceptron θα αποτελεί την είσοδο του επόμενου μέχρι να φτάσουμε στον τελευταίο. Η υλοποίηση αυτή βοηθάει στην άρση περιορισμού για γραμμική κατάταξη δεδομένων καθώς πλέον θα μπορούν να χρησιμοποιηθούν πιο σύνθετα κριτήρια κατηγοριοποίησης.

2.6.1.4. Ιδιότητες

Τα τελευταία χρόνια έχει υπάρξει μία έκρηξη ενδιαφέροντος για τα νευρωνικά δίκτυα καθώς εφαρμόζονται με μεγάλη επιτυχία σε ένα ασυνήθιστα μεγάλο φάσμα τομέων της επιστήμης και της τεχνολογίας, όπως τα χρηματοοικονομικά συστήματα, η ιατρική, η επιστήμη του μηχανικού, η γεωλογία, η φυσική, η ρομποτική, η επεξεργασία σήματος κ.α. Στην πραγματικότητα, τα νευρωνικά δίκτυα εισάγονται οπουδήποτε τίθεται θέμα πρόβλεψης, ταξινόμησης ή ελέγχου όπως ακριβώς και στην περίπτωση του Ramp Metering. Η σαρωτική αυτή επιτυχία, μπορεί να αποδοθεί σε δύο βασικά στοιχεία: την ισχύ και την ευχρηστία.^[34]

- **Ισχύς:** Τα νευρωνικά δίκτυα είναι πολύ εξελιγμένες τεχνικές μη γραμμικής μοντελοποίησης, ικανές να μοντελοποιήσουν εξαιρετικά πολύπλοκες λειτουργίες. Η γραμμική μοντελοποίηση υπήρξε ευρέως διαδεδομένη για πολύ καιρό, δεδομένου ότι στα γραμμικά μοντέλα εφαρμόζονται πολύ γνωστές στρατηγικές βελτιστοποίησης. Στις συνήθειες, όμως, περιπτώσεις όπου η γραμμική προσέγγιση δεν ήταν έγκυρη και κατάλληλη, τα μοντέλα αυτά αποτύγχαναν αναλόγως. Τα νευρωνικά δίκτυα βέβαια, αν και επιτρέπουν τη μη γραμμικότητα μέσω χρήσης μη γραμμικών συναρτήσεων

ενεργοποίησης, μεταθέτουν με τη σειρά τους το πρόβλημα στο ζήτημα της διάστασης (του πλήθους των διαφορετικών εισόδων και εξόδων), το οποίο αποτελεί τροχοπέδη στις προσπάθειες μοντελοποίησης μη γραμμικών συναρτήσεων με μεγάλο αριθμό μεταβλητών.^[34]

- **Ευχρηστία:** Τα νευρωνικά δίκτυα εκπαιδεύονται με παραδείγματα. Ο χρήστης συγκεντρώνει αντιπροσωπευτικά δεδομένα και στη συνέχεια, καθώς τα τροφοδοτεί συστηματικά στο δίκτυο μέσω των κατάλληλων αλγορίθμων εκπαίδευσης, το δίκτυο «αντιλαμβάνεται» αυτομάτως τη δομή των δεδομένων και η «γνώση» αυτή εκφράζεται ως κατάλληλες επιλογές συναπτικών βαρών. Επομένως το τελικό αποτέλεσμα της εκπαίδευσης με ένα συγκεκριμένο σύνολο παραδειγμάτων είναι ο προσδιορισμός των κατάλληλων βαρών του δικτύου αυτόματα από τον αλγόριθμο εκπαίδευσης χωρίς ο χρήστης να χρειαστεί κάνει οτιδήποτε πέρα από το να εισάγει τα δεδομένα. Ο χρήστης χρειάζεται να έχει κάποιες ουσιώδεις γνώσεις σχετικά με τον τρόπο επιλογής και προετοιμασίας των δεδομένων, τον τρόπο επιλογής του κατάλληλου νευρωνικού δικτύου και στο πως θα ερμηνευτούν τα αποτελέσματα. Παρά ταύτα, το επίπεδο των γνώσεων του χρήστη που απαιτούνται για μια επιτυχημένη εφαρμογή των νευρωνικών δικτύων, είναι πολύ χαμηλότερο συγκριτικά με κάποια περίπτωση που θα χρησιμοποιούνταν ορισμένες πιο παραδοσιακές, μη γραμμικές στατιστικές μέθοδοι.^[34]

2.6.1.5. Εφαρμογές

Τα νευρωνικά δίκτυα όπως ήδη αναφέρθηκε είναι εφαρμόσιμα σχεδόν σε κάθε κατάσταση στην οποία ισχύει μια σχέση μεταξύ μεταβλητών πρόβλεψης (ανεξάρτητες, εισροές) και προβλεπόμενες μεταβλητές (εξαρτημένες, εκροές), ακόμα και όταν αυτή η σχέση είναι πολύ περίπλοκη για να αποδοθεί με τους συνηθισμένους όρους της «συσχέτισης» ή των «διαφόρων ομάδων». Ενδεικτικά αντιπροσωπευτικά παραδείγματα προβλημάτων στα οποία η ανάλυση των νευρωνικών δικτύων έχει εφαρμοστεί με επιτυχία είναι τα εξής:^[34]

- **Ιατρική διάγνωση:** Ένα ευρύ φάσμα ιατρικά συσχετιζόμενων ενδείξεων, όπως ο συνδυασμός της καρδιακής συχνότητας, τα επίπεδα των διαφόρων ουσιών στο αίμα, ο ρυθμός της αναπνοής μπορούν να παρακολουθηθούν μέσω κατάλληλων νευρωνικών δικτύων. Η εκδήλωση μιας συγκεκριμένης ιατρικής κατάστασης, γίνεται να συσχετιστεί με ένα πολύπλοκο συνδυασμό μεταβολών σε ένα υποσύνολο μεταβλητών που παρακολουθούνται. Τα νευρωνικά δίκτυα έχουν χρησιμοποιηθεί για την αναγνώριση αυτού του προτύπου πρόβλεψης, ώστε να χορηγηθεί η κατάλληλη θεραπεία.
- **Χρηματιστηριακές προβλέψεις:** Οι διακυμάνσεις των τιμών των μετοχών και των χρηματιστηριακών δεικτών είναι ακόμα ένα παράδειγμα ενός πολύπλοκου, πολυδιάστατου, αλλά και σε ορισμένες περιπτώσεις εν μέρει ντετερμινιστικού

φαινομένου. Τα νευρωνικά δίκτυα χρησιμοποιούνται από πολλούς τεχνικούς αναλυτές, ώστε να κάνουν προβλέψεις σχετικά με τις τιμές των μετοχών, βασιζόμενοι σε ένα μεγάλο αριθμό παραγόντων, όπως δηλαδή, τις προηγούμενες επιδόσεις άλλων αποθεμάτων και διαφόρων οικονομικών δεικτών.

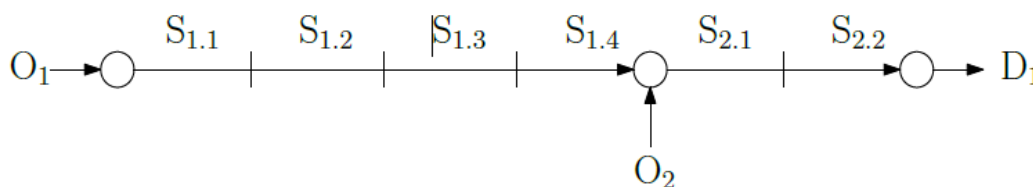
- **Πιστωτική ανάθεση:** Μια ποικιλία από κομμάτια πληροφοριών, τα οποία είναι συνήθως γνωστά για ένα απαιτούμενο δάνειο, για παράδειγμα, η ηλικία του αιτούντος, η εκπαίδευση, το επάγγελμα και πολλά άλλα στοιχεία που μπορεί να είναι διαθέσιμα. Μετά την εκπαίδευση ενός νευρωνικού δικτύου σε ιστορικά δεδομένα η ανάλυση μπορεί να εκτοπίσει τα πιο κατάλληλα και σχετικά χαρακτηριστικά και να τα χρησιμοποιήσει για την ταξινόμηση των αιτούντων ως χαμηλού ή υψηλού κινδύνου.
- **Παρακολούθηση της κατάστασης των μηχανημάτων:** Τα νευρωνικά δίκτυα μπορούν να συμβάλλουν στη μείωση του κόστους με την εξασφάλιση της πρόσθετης εμπειρογνομosύνης για τον προγραμματισμό προληπτικής συντήρησης των μηχανημάτων. Ένα νευρωνικό δίκτυο, λοιπόν, μπορεί να εκπαιδευτεί με τέτοιο τρόπο, ώστε να διακρίνει από τους ήχους τους οποίους παράγει μια μηχανή είτε αν εκτελεί κανονικά τις λειτουργίες της, είτε βρίσκεται στα πρόθυρα εμφάνισης οποιασδήποτε δυσλειτουργίας. Μετά από αυτήν την περίοδο εκπαιδευτικής κατάρτισης, η εμπειρία του ίδιου δικτύου είναι δυνατό να χρησιμοποιηθεί με σκοπό την προειδοποίηση ενός τεχνικού για κάποια επικείμενη βλάβη προτού συμβεί και ενδεχομένως προκαλέσει πολυδάπανες και απρόβλεπτες χρονικές καθυστερήσεις.
- **Συστήματα διαχείρισης κινητήρα:** Τα νευρωνικά δίκτυα έχουν χρησιμοποιηθεί για την ανάλυση των εισροών που δέχονται οι αισθητήρες ενός κινητήρα. Το νευρωνικό δίκτυο ελέγχει μια ποικιλία παραμέτρων με τις οποίες λειτουργεί ο κινητήρας, προκειμένου να επιτευχθεί ένας συγκεκριμένος στόχος. Για παράδειγμα, το δίκτυο αυτό επιχειρεί την ελαχιστοποίηση της κατανάλωσης των καυσίμων.
- **Συστήματα διαχείρισης ζήτησης ενέργειας:** Τα νευρωνικά δίκτυα χρησιμοποιούνται για ανάλυση δεδομένων ζήτησης ενέργειας μέσω καταγραφής των δεδομένων κατανάλωσης και παραγωγής που υπάρχουν στα συστήματα διανομής ενέργειας. Έτσι το νευρωνικό δίκτυο εξετάζει τα δεδομένα αυτά και ρυθμίζει ανάλογα τις μονάδες παραγωγής ενέργειας ανάλογα με την τρέχουσα ζήτηση καθώς και διάφορους άλλους παράγοντες όπως η ημέρα, η ώρα, η εποχή, κ.α. Έτσι μέσω ενός νευρωνικού δικτύου μπορεί να επιτευχθεί αποτελεσματικότερη διαχείριση ενεργειακών δικτύων.

Μετά, λοιπόν, από αυτή τη σύντομη αναφορά σε μερικές από τις εφαρμογές που χρησιμοποιούνται τα νευρωνικά δίκτυα γίνεται άμεσα αντιληπτό ότι η χρήση τους για σύστημα διαχείρισης κίνησης αποτελεί μία συνετή και κατάλληλη επιλογή. Πιο συγκεκριμένα, για την εφαρμογή του Ramp Metering που εξετάζεται στην παρούσα εργασία η χρήση ενός νευρωνικού

δικτύου αποτελεί μία καλή λύση καθώς σε περιπτώσεις που το εξεταζόμενο σύστημα είναι αρκετά περίπλοκο, δηλαδή περιλαμβάνει πολλές λωρίδες, παραπάνω από μία ράμπα, πολλές συνθήκες ελέγχου, κ.α. το πλήθος των δεδομένων είναι τόσο μεγάλο που αλγόριθμοι βέλτιστου ελέγχου και πρόβλεψης απαιτούν μεγάλη υπολογιστική ισχύ και χρόνο για να καταλήξουν σε αποτέλεσμα.

2.6.2. Reinforcement Learning

Όπως ήδη αναφέρθηκε έμπνευση για την ανάπτυξη του νευρωνικού μοντέλου που χρησιμοποιήθηκε στην παρούσα εφαρμογή αποτέλεσε η δημοσίευση “*Decentralised reinforcement learning for ramp metering and variable speed limits on highways*” των Thorsten Schmidt-Dumont και Jan H van Vuuren. Πιο συγκεκριμένα, οι Thorsten Schmidt-Dumont και Jan H van Vuuren προσομοίωσαν ένα κομμάτι αυτοκινητόδρομου χωρισμένο σε τμήματα και με μία ράμπα εισόδου όπως φαίνεται χαρακτηριστικά στην παρακάτω εικόνα:



Εικόνα 16. Μοντέλο Thorsten Schmidt-Dumont και Jan H van Vuuren

Σύμφωνα λοιπόν με την παραπάνω εικόνα υπάρχουν δύο (2) κόμβοι ζήτησης O_1 και O_2 και ένας κόμβος εξόδου ο D_1 . Ο κόμβος O_2 αποτελεί τη ράμπα εισόδου του μελετώμενου συστήματος και ουσιαστικά ο έλεγχος γίνεται πάνω σε αυτή. Μοντελοποιώντας λοιπόν τον αυτοκινητόδρομο όπως φαίνεται παραπάνω και επιλέγοντας ως σημείο ελέγχου έναν φωτεινό σηματοδότη επάνω στη ράμπα στον οποίο καθορίζεται η διάρκεια κατά την οποία αυτός είναι κόκκινος οι Thorsten Schmidt-Dumont και Jan H van Vuuren ανέπτυξαν ένα RL μοντέλο για την επίλυση του προβλήματος. Ως συνάρτηση ανταμοιβής χρησιμοποίησαν τη σχέση (5) που έχει ήδη αναφερθεί ως η συνάρτηση κέρδους ή συνάρτηση στόχος της μεθόδου ALINEA.

Σύμφωνα με το συγκεκριμένο paper σε RL προβλήματα είναι μείζονος σημασία η ισορροπία μεταξύ της εξερεύνησης του διανύσματος των διαθέσιμων ενεργειών (state-action) και της αξιοποίησης των ήδη γνωστών ενεργειών με βάση τα δεδομένα που είχε ως είσοδο το σύστημα. Στην κατεύθυνση αυτή λοιπόν εφαρμόστηκε η παρακάτω σχέση^[29]:

$$\epsilon(s) = \max \left\{ 0,05, \left[\frac{1}{1 + \frac{1}{5N_\alpha(s)} \sum_{i=1}^{\alpha} i(s)} \right] \right\} \quad (18)$$

Όπου:

- $N_\alpha(s)$, ο αριθμός των διαθέσιμων ενεργειών α όταν το σύστημα είναι στην κατάσταση s
- $i(s)$ ο αριθμός των προηγούμενων επισκέψεων στην κατάσταση (s)

Με βάση λοιπόν τη σχέση (18) μία μεταβλητή κατάστασης ϵ ενθαρρύνει την εξερεύνηση νέων καταστάσεων του συστήματος μέσω επαναλαμβανόμενων επισκέψεων στην κατάσταση s έως ότου μειωθεί το ϵ σε τιμή κάτω του 0,05. Η παραπάνω σχέση με ελάχιστες διαφοροποιήσεις μπορεί να εφαρμοστεί για την εφαρμογή ενός kNN (k Nearest Neighbor) αλγορίθμου ο οποίος; μπορεί να χρησιμοποιηθεί για την εφαρμογή του RL μοντέλου του paper. Έτσι, η παραπάνω εξίσωση μετασχηματίζεται σε^[29]:

$$\epsilon(s) = \max \left\{ 0,05, \left[\frac{1}{1 + \frac{1}{11N_{\alpha}(s)} \sum_{i=1}^{\alpha} C^{kNN}(s)} \right] \right\} \quad (19)$$

Όπου:

- $N_{\alpha}(s)$, ο αριθμός των διαθέσιμων ενεργειών α όταν το σύστημα είναι στην κατάσταση s
- $C^{kNN}(s)$, ο αριθμός των προηγούμενων επισκέψεων στην κατάσταση (s), το οποίο δίνεται από τη σχέση:

$$C^{kNN}(s) = \sum_{i=1}^k p_i C(X_i, \alpha) \quad (20)$$

Έτσι, οι σχέσεις (19) και (20) αποτελούν τη μαθηματική διατύπωση του προβλήματος Ramp Metering με βάση το μοντέλο που εξέτασαν και προσομοίωσαν οι Thorsten Schmidt-Dumont και Jan H van Vuuren στη δημοσίευση με τίτλο “*Decentralised reinforcement learning for ramp metering and variable speed limits on highways*”.

2.6.3. Νευρωνικό Δίκτυο εφαρμογής Ramp Metering

Στις προηγούμενες παραγράφους έγινε μία συνοπτική παρουσίαση της βασικής θεωρίας των νευρωνικών δικτύων. Τα σενάρια τα οποία εξετάζονται στην παρούσα εργασία προσεγγίζουν κατά το δυνατό την πραγματικότητα και για τον λόγο αυτό τόσο το νευρωνικό δίκτυο όσο και τα δεδομένα εισόδου και εξόδου του νευρωνικού έγιναν με τέτοιο τρόπο ώστε να είναι εφαρμόσιμα σε πραγματικά σενάρια λειτουργίας.

Σε επέκταση της συνοπτικής παρουσίασης του RL μοντέλου που παρουσιάστηκε στο προηγούμενο υποκεφάλαιο στην παρούσα διπλωματική αναπτύχθηκε ένα νευρωνικό δίκτυο το οποίο εξετάζει ένα τμήμα ενός αυτοκινητόδρομου με τρεις (3) λωρίδες και δύο (2) ράμπες εισόδου. Η προσθήκη τόσο των επιπλέον δύο (2) λωρίδων όσο και της δεύτερης ράμπας αυξάνει σε μεγάλο βαθμό την πολυπλοκότητα καθώς πρέπει να ληφθούν υπόψη αρκετές παράμετροι, ώστε οι δύο (2) ράμπες να μην λειτουργούν αυτόνομα αλλά να συνεργάζονται προκειμένου να αποφευχθούν ανεπιθύμητες ενέργειες και αποτελέσματα στην κυκλοφορία εντός του αυτοκινητόδρομου αλλά και στην είσοδο σε αυτόν από τις ράμπες.

Οι εισοδοί και οι έξοδοι του νευρωνικού δικτύου καθορίζονται πάντα από την εφαρμογή, τον αριθμό των κρυφών επιπέδων, τον αριθμό των νευρώνων σε κάθε κρυφό επίπεδο, τον τύπο του, τις συναρτήσεις μεταφοράς και τον αριθμό των επαναλήψεων εκπαίδευσης. Ο προσδιορισμός όμως όσων στοιχείων αναφέρθηκαν παραπάνω για την

τεκμηρίωση και την ανάπτυξη ενός τέτοιου δικτύου πάντα προκύπτουν μετά από δοκιμές και πειραματισμό. Γενικότερα δεν υπάρχουν κανόνες σχετικά με την επιλογή της τοπολογίας και των παραμέτρων των νευρωνικών δικτύων, έτσι και στην παρούσα εργασία πραγματοποιήθηκε πειραματισμός και ανάλυση της δομής του συγκεκριμένου προβλήματος προκειμένου να επιτευχθεί η επιθυμητή απόδοση.

Κατόπιν λοιπόν έντονου πειραματισμού και υλοποίησης πολλαπλών σεναρίων λειτουργίας μίας (1) ράμπας εισόδου αρχικά και δύο (2) ραμπών στη συνέχεια δημιουργήθηκε τυχαία αλλά βασισμένο πάντα σε όσο το δυνατόν πραγματικές συνθήκες ένα αρκετά μεγάλο πλήθος δεδομένων το οποίο αποτέλεσε το σετ εκπαίδευσης του νευρωνικού δικτύου που αναπτύχθηκε. Πιο συγκεκριμένα αρχικά εκτελέστηκαν πολλαπλές επαναλήψεις διαφορών σεναρίων ελέγχου και από αυτές τις επαναλήψεις προέκυψαν διάφορα δεδομένα τα οποία μετά από μελέτη αποκάλυψαν ποικιλία μοτίβων συσχετισμού και αλληλεξάρτησης των δεδομένων. Τα μοτίβα αυτά εξετάστηκαν ενδελεχώς και οδήγησαν στην επιλογή των δεδομένων εκπαίδευσης.

Τα δεδομένα εκπαίδευσης θα έχουν την παρακάτω μορφή:

T	Section 1	Section 2	Section 3	...	Section n	Density	Mean Speed V	Output time ramp 1	Output time ramp 2
T=1									
T=2									
T=3									
...									
T=n									

Πίνακας 3. Δομή δεδομένων εκπαίδευσης νευρωνικού δικτύου

Πιο συγκεκριμένα:

- η στήλη **T** αφορά τη χρονική στιγμή,
- οι στήλες **Section** αφορούν τα τμήματα του αυτοκινητόδρομου που εξετάζεται και συμπληρώνονται με 0 ή 1 το οποίο υποδηλώνει εάν υπάρχει αυτοκίνητο η όχι στο αντίστοιχο τμήμα του αυτοκινητόδρομου
- η στήλη **Density** περιέχει την τιμή της πυκνότητας σε όλο το μήκος του αυτοκινητόδρομου
- η στήλη **Mean Speed V** έχει τη μέση ταχύτητα των οχημάτων για κάθε χρονική στιγμή στον αυτοκινητόδρομο
- οι στήλες **Output Time Ramp x** αποτελούν τα δεδομένα εξόδου του νευρωνικού δικτύου και έχουν τους χρόνους σε λεπτά τους οποίους ο φωτεινός σηματοδότης είναι κόκκινος

Η βιβλιοθήκη η οποία χρησιμοποιήθηκε για την υλοποίηση του νευρωνικού δικτύου είναι η Keras της γλώσσας Python η οποία αποτελεί μία από τις πιο σύγχρονες βιβλιοθήκες στον συγκεκριμένο τομέα. Η συγκεκριμένη βιβλιοθήκη έχει τη δυνατότητα μέσω κατάλληλων παραμετροποιήσεων οι οποίες προέκυψαν μετά από εκτέλεση εκτενών δοκιμών να παράγει τα κατάλληλα βάρη και συσχετίσεις μεταξύ των δεδομένων και να εκπαιδεύει αποδοτικά το δίκτυο ώστε να επιτυγχάνεται το επιθυμητό αποτέλεσμα. Η Keras πλέον υποστηρίζεται από τη βιβλιοθήκη Tensorflow της Google η οποία αποτελεί ουσιαστικά το backend της. Η Tensorflow και η Keras αποτελούν από τις πιο διαδεδομένες και σύγχρονες βιβλιοθήκες στον τομέα της μηχανικής μάθησης.

Η υλοποίηση του συγκεκριμένου νευρωνικού αποτελεί στην ουσία μία υλοποίηση ενός Perceptron όπως αυτοί έχουν περιγραφεί σε προηγούμενο υποκεφάλαιο. Όσο πιο πολλές ράμπες προσθέτουμε τόσο αυξάνεται η πολυπλοκότητά του δικτύου άρα τόσο πιο πολλοί Perceptrons δημιουργούνται και άρα τόσο πιο επιτακτική γίνεται η ανάγκη για χρήση νευρωνικού δικτύου.

2.7. Παραδοχές για την Υλοποίηση του Εξεταζόμενου Προβλήματος

Στα προηγούμενα υποκεφάλαια αναλύθηκε η μοντελοποίηση του αυτοκινητόδρομου με εξισώσεις που αφορούν τον αριθμό οχημάτων που κινούνται σε αυτόν καθώς και εξισώσεις για την πυκνότητα. Επιπλέον παρουσιάστηκε μία διατύπωση του προβλήματος και των μεταβλητών που το αποτελούν και θα εξεταστούν. Όλες οι εξισώσεις που παρουσιάστηκαν παραπάνω βρίσκουν εφαρμογή στο εξεταζόμενο πρόβλημα και αποτελούν τη μαθηματική μοντελοποίηση του προβλήματός μας.

Χάριν απλότητας θα γίνει μία σύντομη ανασκόπηση και απλοποίηση των εξισώσεων που παρουσιάστηκαν προηγουμένως. Η απλοποίηση των εξισώσεων πραγματοποιήθηκε έτσι ώστε οι γενικευμένες εξισώσεις που ισχύουν για το πρόβλημα να αποκτήσουν μία πιο ρεαλιστική μορφή με βάση τα δεδομένα που εξετάστηκαν κατά την προγραμματιστική υλοποίηση των αλγορίθμων που παρουσιάζονται σε επόμενο κεφάλαιο.

Ως πυκνότητα σε ένα τμήμα ενός αυτοκινητόδρομου ορίζουμε το σύνολο των αυτοκινήτων που βρίσκεται πάνω σε αυτό σε σχέση με τη συνολική του χωρητικότητα. Έτσι έχουμε τα εξής:

- C , χωρητικότητα τμήματος αυτοκινητόδρομου. Η οποία ταυτίζεται με το σύνολο των τομέων (sectors) που έχει ο αυτοκινητόδρομος καθώς κάθε sector μπορεί να έχει το πολύ ένα αυτοκίνητο.

- S , σύνολο αυτοκινήτων επί του αυτοκινητόδρομου στο συγκεκριμένο τμήμα που εξετάζεται. Το άθροισμα υπολογίζεται από το σύνολο των 1 που υπάρχουν στους τομείς του αυτοκινητόδρομου
- ρ , πυκνότητα τμήματος αυτοκινητόδρομου

Επομένως, με βάση την εξίσωση (12) και θεωρώντας τη χωρητικότητα του αυτοκινητόδρομου δεδομένη και σταθερή ως πυκνότητα ορίζουμε:

$$\rho = \frac{S}{c} * 100\% \quad (21)$$

Επομένως το μέγεθος που επηρεάζει την πυκνότητα στο εξεταζόμενο σύστημα είναι το πλήθος των αυτοκινήτων που κινούνται επί του κύριου ρεύματος του αυτοκινητόδρομου. Το συγκεκριμένο αυτό μέγεθος μεταβάλλεται όταν νέα αυτοκίνητα εισέρχονται στον αυτοκινητόδρομο είτε από τις ράμπες εισόδου είτε εμφανίζονται στην αρχή του εξεταζόμενου τμήματος καθώς και όταν αυτοκίνητα εξέρχονται στο τέλος του εξεταζόμενου τμήματος του αυτοκινητόδρομου.

Το μέγεθος της πυκνότητας είναι ένα από τα πιο κρίσιμα μεγέθη σε τέτοιες εφαρμογές καθώς αποτελεί αν όχι τη βασικότερη, μία από τις βασικότερες παραμέτρους για την υλοποίηση αλγορίθμων που αφορούν το Ramp Metering. Γενικότερα και βάσει της διεθνούς βιβλιογραφίας, η διατήρηση του παραπάνω μεγέθους κοντά σε ένα επιθυμητό νούμερο το οποίο καθορίζεται από το μέγεθος και τη χωρητικότητα του αυτοκινητόδρομου αποτελεί στόχο των συγκεκριμένων αλγορίθμων. Έτσι, ο βασικός στόχος είναι η σταθεροποίηση του παραπάνω μεγέθους σε ένα επιθυμητό εύρος τιμών το οποίο κυμαίνεται πλησίον της οριακής τιμής πυκνότητας προκειμένου να αξιοποιούνται στο μέγιστο οι δυνατότητες του κύριου ρεύματος του αυτοκινητόδρομου.

Η συνολική πυκνότητα όμως δεν αποτελεί τον μοναδικό παράγοντα από τον οποίο εξαρτάται ένα τέτοιο πρόβλημα ειδικά στις περιπτώσεις που περισσότερες της μίας ράμπας εξετάζονται στο ίδιο σύστημα. Τότε η πυκνότητα δεν εξετάζεται σαν ένα ενιαίο νούμερο αλλά χωρίζεται σε τμήματα καθώς η επίλυση του προβλήματος πρέπει να εξετάσει συνθήκες που αφορούν τόσο πριν όσο και μετά την εκάστοτε ράμπα αυτή την τιμή καθώς και την πυκνότητα μετά από όλες τις ράμπες η οποία ουσιαστικά αποτελεί και το μέγεθος το οποίο θέτει τον στόχο.

Ένα ακόμα κρίσιμο μέγεθος για την ομαλή ροή των αυτοκινήτων εντός του αυτοκινητόδρομου είναι η μέση ταχύτητα κίνησης στο κύριο ρεύμα. Ως μέση ταχύτητα κίνησης ορίζεται το άθροισμα των ταχυτήτων των αυτοκινήτων στο κύριο ρεύμα προς το σύνολό τους. Έτσι σε συμφωνία με την εξίσωση (7) έχουμε:

$$V = \frac{u_i}{K} \quad (22)$$

Όπου:

- V , η υπολογιζόμενη μέση ταχύτητα των αυτοκινήτων

- u_i , η ταχύτητα του εκάστοτε αυτοκινήτου i εντός του κυρίου ρεύματος στον αυτοκινητόδρομο
- K , το σύνολο των αυτοκινήτων που κινείται επί του αυτοκινητόδρομου

Η μέση ταχύτητα V είναι το μέγεθος που εξετάζει πόσο γρήγορα κινείται το ρεύμα εντός του αυτοκινητόδρομου και άρα πόσο πιο σύντομα διανύουν μία συγκεκριμένη απόσταση τα αυτοκίνητα. Η διατήρηση του συγκεκριμένου μεγέθους σε υψηλά επίπεδα αποτελεί στόχο των αλγορίθμων Ramp Metering. Η μέση ταχύτητα ουσιαστικά υποδεικνύει κατά πόσο υπάρχει αυξημένη ροή στον αυτοκινητόδρομο ή όχι. Έτσι όταν η μέση ταχύτητα είναι αρκετά υψηλή τότε υπάρχει περιθώριο για περαιτέρω αύξηση της πυκνότητας. Ωστόσο όπως και η ροή η οποία εξετάστηκε παραπάνω εάν η πυκνότητα ξεπεράσει το κρίσιμο σημείο τότε η μέση ταχύτητα αρχίζει με τη σειρά της και αυτή να μειώνεται.

Βέβαια η μέση ταχύτητα κίνησης ή η πυκνότητα του αυτοκινητόδρομου από μόνα τους δίνουν αβέβαια αποτελέσματα για την αποδοτικότητα και την ορθή λειτουργία της εκάστοτε μεθόδου. Με δεδομένο αυτό και θέλοντας να μετριάσουμε τους κινδύνους που ενέχουν σε μία μονόπλευρη βελτιστοποίηση ενός μεγέθους και μόνο, κρίθηκε απαραίτητο να εξεταστεί ένας συνδυασμός των δύο παραπάνω παραμέτρων με τη μορφή ενός μαθηματικού λόγου. Ο λόγος που συσχετίζει τα δύο μεγέθη είναι ο παρακάτω:

$$R = \frac{\rho}{V} \quad (23)$$

Ο παραπάνω λόγος είναι αδιάστατος και εκφράζει τη συσχέτιση μεταξύ μέσης ταχύτητας και πυκνότητας. Πιο συγκεκριμένα:

- Όσο πιο μεγάλη είναι η πυκνότητα με σταθερή ταχύτητα τόσο πιο μεγάλος είναι ο λόγος
- Όσο πιο μεγάλη είναι η ταχύτητα με σταθερή πυκνότητα τόσο πιο μικρός είναι ο λόγος
- Όταν η ταχύτητα αυξάνεται με μεγαλύτερο ρυθμό από την πυκνότητα τότε ο λόγος μειώνεται διαφορετικά αυξάνεται
- Όταν η πυκνότητα αυξάνεται και παράλληλα η ταχύτητα μειώνεται τότε ο λόγος αυξάνεται και αντίστροφα

Ο παραπάνω λόγος σε συνδυασμό με τα μεγέθη της μέσης ταχύτητας και της πυκνότητας θα αποτελέσουν τα κριτήρια σύγκρισης και συμπεριφοράς του κάθε αλγορίθμου που θα αναπτυχθεί και με βάση τα αποτελέσματα θα αξιολογηθεί η αποδοτικότητά του υπό διάφορα σενάρια και καταστάσεις λειτουργίας.

Επιπλέον, το παραπάνω σύνολο δεδομένων (μέση ταχύτητα, πυκνότητα, λόγος R) θα αποτελεί μέτρο σύγκρισης μεταξύ των αλγορίθμων καθώς κάτω από ίδιες συνθήκες θα γίνει μία αντιπαραβολή και τεκμηρίωση των αποτελεσμάτων σε επόμενο κεφάλαιο. Ωστόσο πρέπει να επισημανθεί ότι πέρα από τη συμπεριφορά του συγκεκριμένου λόγου για κάθε αλγόριθμο θα εξεταστεί και η αποδοτικότητά του ως προς τον χρόνο εκτέλεσης και προσαρμογής στις

διάφορες καταστάσεις. Κρίνεται απαραίτητο να καθορισθεί μία συνθήκη τερματισμού του κάθε αλγορίθμου έτσι ώστε να μπορούν να πραγματοποιηθούν οι συγκρίσεις μεταξύ των αλγορίθμων αλλά και να παραχθούν τα εκάστοτε αποτελέσματα. Η συνθήκη τερματισμού μπορεί να είναι:

- Προκαθορισμένος αριθμός επαναλήψεων
- Προκαθορισμένη διάρκεια εκτέλεσης του αλγορίθμου

Όσον αφορά την υλοποίηση του νευρωνικού δικτύου κατόπιν της εκπαίδευσης του με τα δεδομένα που αναλύθηκαν σε προηγούμενο υποκεφάλαιο τότε το νευρωνικό θα παράγει τις εξόδους του με βάση την επιθυμητή πυκνότητα. Έτσι στο νευρωνικό θα δίνεται η επιθυμητή πυκνότητα και με βάση αυτή και τις τρέχουσες συνθήκες θα προσαρμόζει τους χρόνους για τις δύο (2) ράμπες έτσι ώστε ο αυτοκινητόδρομος να λειτουργεί πάντα κοντά σε αυτό το ποσοστό πυκνότητας έτσι ώστε να αξιοποιούνται στον μέγιστο βαθμό οι δυνατότητες του εξεταζόμενου συστήματος.

ΚΕΦΑΛΑΙΟ 3. ΠΡΟΣΣΕΓΙΣΗ ΚΑΙ ΜΕΘΟΔΟΙ **ΥΛΟΠΟΙΗΣΗΣ**

3.1. Εισαγωγή και Γενικά Στοιχεία για την Python

Όλες οι προσεγγίσεις και τα διάφορα σενάρια αναπτύχθηκαν στη γλώσσα Python. Η Python είναι διερμηνευόμενη (interpreted), γενικού σκοπού (general-purpose) και υψηλού επιπέδου, γλώσσα προγραμματισμού. Ανήκει στις γλώσσες προστακτικού προγραμματισμού (Imperative programming) και υποστηρίζει τόσο τον διαδικαστικό (procedural programming) όσο και τον αντικειμενοστραφή (object-oriented programming) προγραμματισμό. Επιπλέον, είναι μία δυναμική γλώσσα προγραμματισμού (dynamically typed). Δημιουργήθηκε από τον Ολλανδό Γκίντο βαν Ρόσσομ (Guido van Rossum) το 1990 και κυκλοφόρησε για πρώτη φορά το 1991.

Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα από ότι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες καθώς και για την ταχύτητα εκμάθησής της.

Οι διερμηνευτές της Python είναι διαθέσιμοι για εγκατάσταση σε πολλά λειτουργικά συστήματα, επιτρέποντάς της την εκτέλεση κώδικα σε ευρεία γκάμα συστημάτων. Χρησιμοποιώντας εργαλεία τρίτων, όπως το Py2exe ή το Pyinstaller, ο κώδικας της Python μπορεί να πακεταριστεί σε αυτόνομα εκτελέσιμα προγράμματα για μερικά από τα πιο δημοφιλή λειτουργικά συστήματα, επιτρέποντας τη διανομή του βασισμένου σε Python λογισμικού για χρήση σε αυτά τα περιβάλλοντα χωρίς να απαιτείται εγκατάσταση του διερμηνευτή της Python.

Η Python αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL. Το όνομα της γλώσσας προέρχεται από την ομάδα των Άγγλων κωμικών Μόντυ Πάιθον και δεν έχει καμιά σχέση με το φίδι πύθωνα, παρότι το λογότυπό της παραπέμπει σε κάτι τέτοιο.^[33]

Στο Παράρτημα Α υπάρχουν περισσότερες πληροφορίες σχετικά με τη δομή και τη συγγραφή προγραμμάτων σε γλώσσα Python.

3.2. Προγραμματιστική Προσέγγιση

Η υλοποίηση όλων των σεναρίων έγινε με τη χρήση της γλώσσας προγραμματισμού Python, για την οποία έγινε μία σύντομη περιγραφή στο προηγούμενο Υποκεφάλαιο. Η λογική

που ακολουθήθηκε είναι αυτή του αντικειμενοστραφούς προγραμματισμού έτσι ώστε να είναι εύκολο να αλλαχθούν παράμετροι και σενάρια λειτουργίας.

Πιο συγκεκριμένα, δημιουργήθηκαν τρεις (3) βασικές κλάσεις:

- Κλάση **Highway**, εκτελεί όλες τις απαραίτητες συναρτήσεις (functions) που αφορούν το κύριο ρεύμα κίνησης του αυτοκινητόδρομου
- Κλάση **Ramp**, εκτελεί όλες τις απαραίτητες συναρτήσεις (functions) που αφορούν τη ράμπα ή τις ράμπες εισόδου στον αυτοκινητόδρομο
- Κλάση **Aux_Jane**, εκτελεί όλες τις απαραίτητες συναρτήσεις (functions) που αφορούν τη βοηθητική λωρίδα

Οι παραπάνω κλάσεις αναπτύχθηκαν με τέτοιο τρόπο ώστε με ευκολία να μπορούν να τρέξουν όλα τα απαιτούμενα σενάρια αλλά και να προσαρμοστούν κατάλληλα για την υλοποίηση νέων και πιο απαιτητικών. Οι κλάσεις αποτέλεσαν το βασικότερο κομμάτι της υλοποίησης του κώδικα και το πιο «ζωντανό» καθώς σε κάθε υλοποίηση έπρεπε να ενημερώνονται με τρόπο τέτοιο ώστε να είναι πλήρως συμβατές τόσο με τους υφιστάμενους αλγορίθμους όσο και με τους νέους.

Παράλληλα με τις κλάσεις εξετάστηκαν διάφορα σενάρια ελέγχου και υλοποίησης του προβλήματος Ramp Metering, τα βασικότερα των οποίων εξετάζονται στα επόμενα υποκεφάλαια. Τα σενάρια που εξετάστηκαν προέκυψαν μετά από έρευνα και μελέτη της διεθνούς βιβλιογραφίας καθώς και από εμπειρική γνώση έτσι ώστε να είναι όσο το δυνατόν πιο κοντά στην πραγματικότητα.

Επισημαίνεται ότι όλα τα σενάρια που υλοποιήθηκαν παρήγαγαν ρεαλιστικά δεδομένα για τη συμπεριφορά ενός αυτοκινητόδρομου τα οποία ήταν τόσο μεγάλα σε πλήθος που πρακτικά η αξιοποίησή τους από κλασσικούς αλγορίθμους ελέγχου απαιτεί τεράστια υπολογιστική ισχύ. Η παραγωγή τόσο μεγάλου πλήθους τυχαίων δεδομένων (brute force) εξασφαλίζει την αντικειμενικότητά τους και τα καθιστά κατάλληλα για χρήση στην εκπαίδευση ενός νευρωνικού δικτύου. Αξίζει να σημειωθεί ότι η παραγωγή τέτοιου πλήθους δεδομένων είναι μία αρκετά απαιτητική διαδικασία καθώς πρέπει να εξασφαλίζεται η τυχειότητα και η ορθότητά τους έτσι ώστε να μην υπάρξουν αστοχίες που θα οδηγήσουν σε κακή εκπαίδευση και άρα και εσφαλμένη λειτουργία του νευρωνικού. Επομένως, μετά την παραγωγή κάθε συνόλου δεδομένων πραγματοποιήθηκε έλεγχος για το κατά πόσο αυτά είναι ωφέλιμα και ουσιαστικά για το σύστημα που εξετάζεται.

Στα επόμενα υποκεφάλαια γίνεται μία παρουσίαση των βασικότερων σεναρίων που υλοποιήθηκαν και καταγράφονται όλες οι παραδοχές και τα βήματα που ακολουθήθηκαν ενώ παρουσιάζονται σχηματικά και λογικά διαγράμματα για την καλύτερη κατανόηση αυτών. Στο τέλος του παρόντος κεφαλαίου γίνεται εκτενής αναφορά στην υλοποίηση του νευρωνικού δικτύου. Στο Παράρτημα Β υπάρχουν κομμάτια πηγαίου κώδικα για κάθε σενάριο υλοποίησης.

3.3. Πιλοτικό Σενάριο Υλοποίησης

Στην παρούσα παράγραφο θα γίνει ανάλυση μίας πρώτης και πιλοτικής προσέγγισης του εξεταζόμενου προβλήματος. Η αρχική αυτή ανάλυση είναι αρκετά απλοποιημένη ωστόσο αποτελεί τον αλγόριθμο πιλότο και τον πυλώνα ανάπτυξης όλων των σεναρίων και υλοποιήσεων που θα εξεταστούν στις επόμενες παραγράφους. Η αρχική μοντελοποίηση που έγινε, εξέτασε το πρόβλημα ως ένα απλό σύστημα αυτομάτου ελέγχου το οποίο έχει τις εξής παραδοχές:

- Ο αυτοκινητόδρομος αποτελείται από μία (1) και μόνο λωρίδα
- Η ράμπα εισόδου αποτελείται από μία (1) και μόνο λωρίδα
- Ο αυτοκινητόδρομος χωρίζεται σε δέκα (10) τμήματα τα οποία διαθέτουν αισθητήρες οι οποίοι υποδηλώνουν την ύπαρξη (1) ή την απουσία (0) οχήματος στον εκάστοτε τομέα
- Κάθε αυτοκίνητο εισέρχεται από τη ράμπα στη βοηθητική λωρίδα (buffer) και από εκεί στον αυτοκινητόδρομο
- Ένα αυτοκίνητο εντός του αυτοκινητόδρομου μετακινείται από το τμήμα x_k στο τμήμα x_{k+1} από τη χρονική στιγμή t στη χρονική στιγμή $t+1$. Δηλαδή έστω ότι έχουμε την ακολουθία:

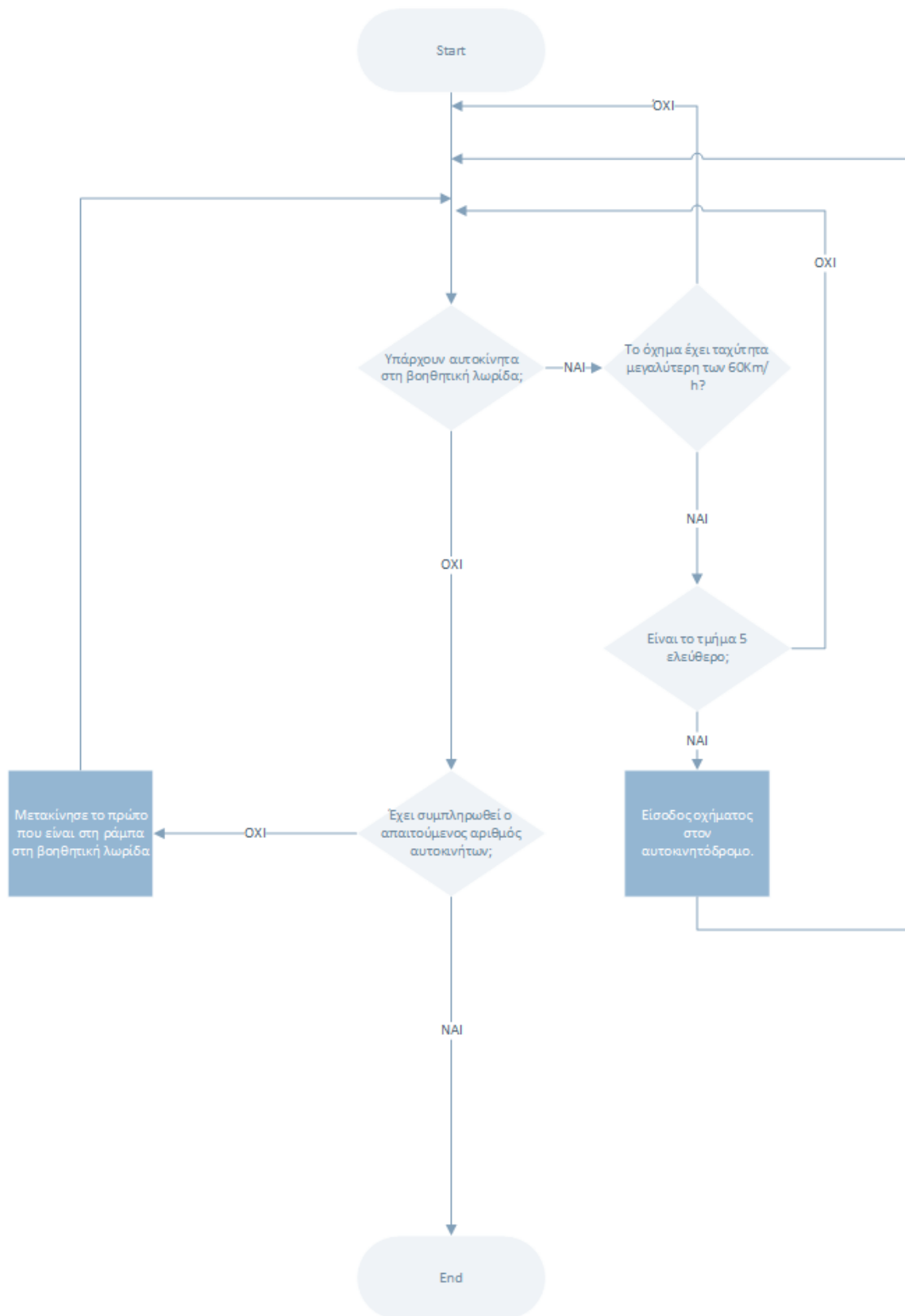
[0 1 1 0 1 0 0 0 1 1] τη

χρονική στιγμή t . Τότε τη χρονική στιγμή $t+1$ και με δεδομένο ότι κάποιο αυτοκίνητο δεν εισέρχεται μέσω της ράμπας στον αυτοκινητόδρομο αυτό θα γίνει:

[0 0 1 1 0 1 0 0 0 1].

- Το τμήμα που καθορίζει για το εάν θα μπει ένα αυτοκίνητο στον αυτοκινητόδρομο ή όχι εάν αυτό είναι κατειλημμένο είναι το τμήμα πέντε (5)
- Εάν το τμήμα πέντε (5) είναι ελεύθερο και το αυτοκίνητο στη βοηθητική λωρίδα έχει ταχύτητα άνω ενός ορίου (π.χ.60 km/h) τότε εισέρχεται στο κύριο ρεύμα διαφορετικά παραμένει στη βοηθητική λωρίδα
- Το σύνολο των αυτοκινήτων είναι που εξετάζεται είναι στατικό (για παράδειγμα χίλια (1000)) στην αρχική υλοποίηση
- Κάθε χρονική στιγμή δημιουργείται τυχαία ένας αριθμός 0 ή 1 και ο αρχικός πίνακας κάνει ολίσθηση προς τα δεξιά

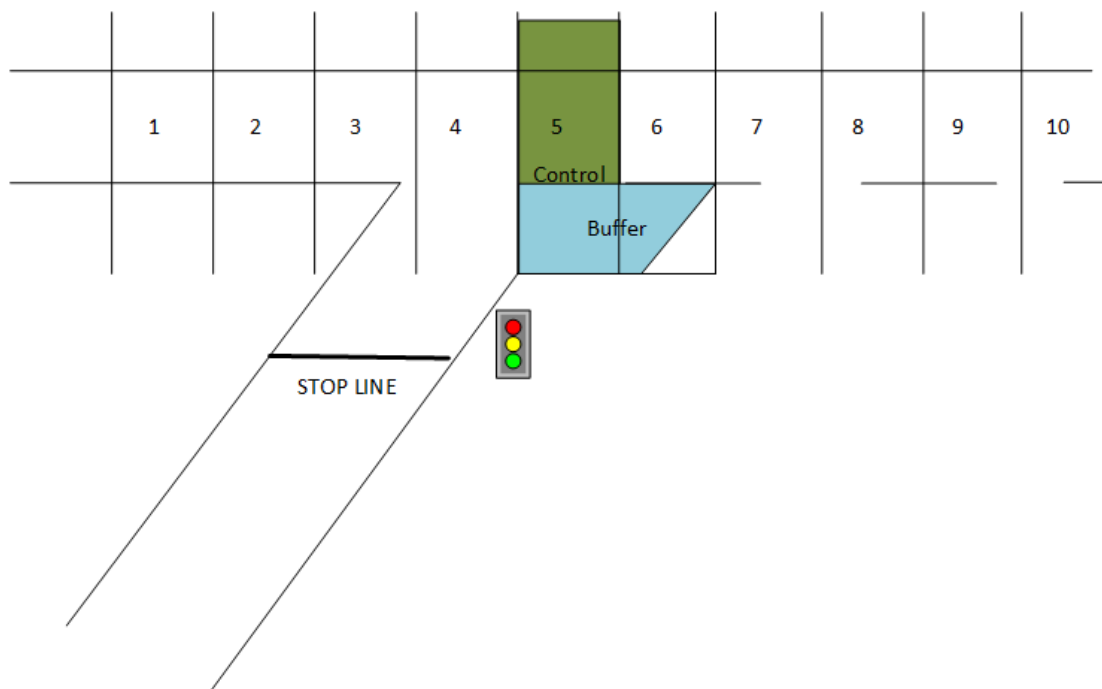
Με βάση λοιπόν όλες τις παραπάνω παραδοχές ο αλγόριθμος επιτρέπει ή όχι την είσοδο ενός αυτοκινήτου φαίνεται στο παρακάτω λογικό διάγραμμα:



Εικόνα 17. Διάγραμμα ροής πιλοτικού αλγορίθμου

Επισημαίνεται ότι στον πίνακα στον οποίο έχουν καταχωρηθεί τα αυτοκίνητα υπάρχει και η πληροφορία για το εάν ένα αυτοκίνητο βρίσκεται στη ράμπα, στη βοηθητική λωρίδα ή στο κεντρικό ρεύμα. Με βάση λοιπόν αυτή την πληροφορία υπολογίζονται τα αθροίσματα για το σύνολο των αυτοκινήτων σε κάθε ένα από τα τμήματα του αυτοκινητόδρομου. Εφόσον υπάρχουν οχήματα στη βοηθητική λωρίδα και αυτά έχουν ταχύτητα άνω των 60χμ για παράδειγμα τότε αυτό που είναι πρώτο εισέρχεται στο κύριο ρεύμα εάν ο τομέας πέντε (5) έχει τιμή μηδέν. Μόλις από τη βοηθητική λωρίδα φύγει ένα αυτοκίνητο τότε ελέγχεται η ράμπα και εάν κάποιος βρίσκεται εντός της ράμπας τότε εισέρχεται στη βοηθητική λωρίδα μέχρι να μπει στο κύριο ρεύμα. Ο αλγόριθμος εκτελείται μέχρις ότου συμπληρωθεί ένας συγκεκριμένος αριθμός που εισήλθαν στον αυτοκινητόδρομο από τη ράμπα για παράδειγμα χίλια (1000) αυτοκίνητα.

Ο παραπάνω αλγόριθμος είναι αρκετά απλοϊκός ωστόσο μπορεί εύκολα να δώσει μία εικόνα ενός συστήματος ελέγχου εισόδου σε ένα αυτοκινητόδρομο. Η λογική που ακολουθεί έχει αρκετές παραδοχές και απαιτεί μεγάλη στατικότητα στα δεδομένα και στη συμπεριφορά του συστήματος ωστόσο με μερικές μικρές τροποποιήσεις θα μπορούσε να εκτελεί τον έλεγχο για συστήματα εισόδου μικρής κλίμακας. Οι υπολογιστικές του απαιτήσεις είναι μικρές και αυτό τον καθιστά κατάλληλο, με κατάλληλες τροποποιήσεις, για συστήματα στα οποία υπάρχουν ελεγκτές με μικρές υπολογιστικές δυνατότητες και η συμπεριφορά του δικτύου είναι ντετερμινιστική.



Εικόνα 18. Σχηματική απεικόνιση εξεταζόμενης διάταξης πιλοτικής υλοποίησης

3.4. 1^ο Σενάριο Υλοποίησης – 1 Λωρίδα & 1 ράμπα

Σε επέκταση του αλγορίθμου πιλότου και με δεδομένο ότι δεν υπάρχει δυνατότητα για έλεγχο της ταχύτητας των οχημάτων επί της βοηθητικής λωρίδας παρακάτω παρουσιάζεται μία γενικευμένη προσέγγιση του προβλήματος. Πιο συγκεκριμένα εξετάζεται η περίπτωση που υπάρχει μόνο μία (1) λωρίδα στον αυτοκινητόδρομο και θέλουμε να ελέγξουμε τον φωτεινό σηματοδότη σε μία (1) ράμπα. Άρα έχουμε στην προκειμένη περίπτωση έχουμε τις παρακάτω παραδοχές:

- Ο αυτοκινητόδρομος αποτελείται από μία (1) και μόνο λωρίδα
- Η ράμπα εισόδου αποτελείται από μία (1) και μόνο λωρίδα
- Ο αυτοκινητόδρομος χωρίζεται σε δέκα (10) τμήματα τα οποία διαθέτουν αισθητήρες οι οποίοι υποδηλώνουν την ύπαρξη (1) ή την απουσία (0) οχήματος στον εκάστοτε τομέα
- Κάθε αυτοκίνητο εισέρχεται από τη ράμπα στη βοηθητική λωρίδα (buffer) και από εκεί στον αυτοκινητόδρομο
- Ένα αυτοκίνητο εντός του αυτοκινητόδρομου μετακινείται από το τμήμα x_k στο τμήμα x_{k+1} από τη χρονική στιγμή t στη χρονική στιγμή $t+1$. Δηλαδή έστω ότι έχουμε την ακολουθία:

[0 1 1 0 1 0 0 0 1 1] τη χρονική στιγμή t . Τότε τη χρονική στιγμή $t+1$ και με δεδομένο ότι κάποιο αυτοκίνητο δεν εισέρχεται μέσω της ράμπας στον αυτοκινητόδρομο αυτό θα γίνει:

[0 0 1 1 0 1 0 0 0 1].

- Τα τμήματα που καθορίζουν για το εάν θα μπει ένα αυτοκίνητο στον αυτοκινητόδρομο ή όχι εάν αυτά είναι κατειλημμένα είναι τα τμήματα τρία (3) και τέσσερα (4)
- Το σύνολο των αυτοκινήτων που εξετάζεται είναι πεπερασμένο και ο αλγόριθμος τερματίζει όταν συμπληρωθεί ένας συγκεκριμένος αριθμός αυτοκινήτων που εισήλθαν στον αυτοκινητόδρομο μέσω της ράμπας

Ο αλγόριθμος εξετάζει την κατάσταση της βοηθητικής λωρίδας και της ράμπας. Έτσι εάν η βοηθητική λωρίδα είναι κατειλημμένη, και υπάρχει αυτοκίνητο στη ράμπα τότε ο φωτεινός σηματοδότης είναι κόκκινος, διαφορετικά εφόσον ο buffer δεν είναι κατειλημμένος ο φωτεινός σηματοδότης είναι πράσινος. Προκειμένου ένα αυτοκίνητο που βρίσκεται στον buffer να μπορεί να εισέλθει στο κύριο ρεύμα του αυτοκινητόδρομου πρέπει είτε τα τμήματα τρία (3) και τέσσερα (4) να είναι ελεύθερα είτε το τμήμα τέσσερα (4) να είναι ελεύθερο και στο τμήμα τρία (3) το όχημα να κινείται με ταχύτητα κάτω από ένα όριο (π.χ. 70 km/h).

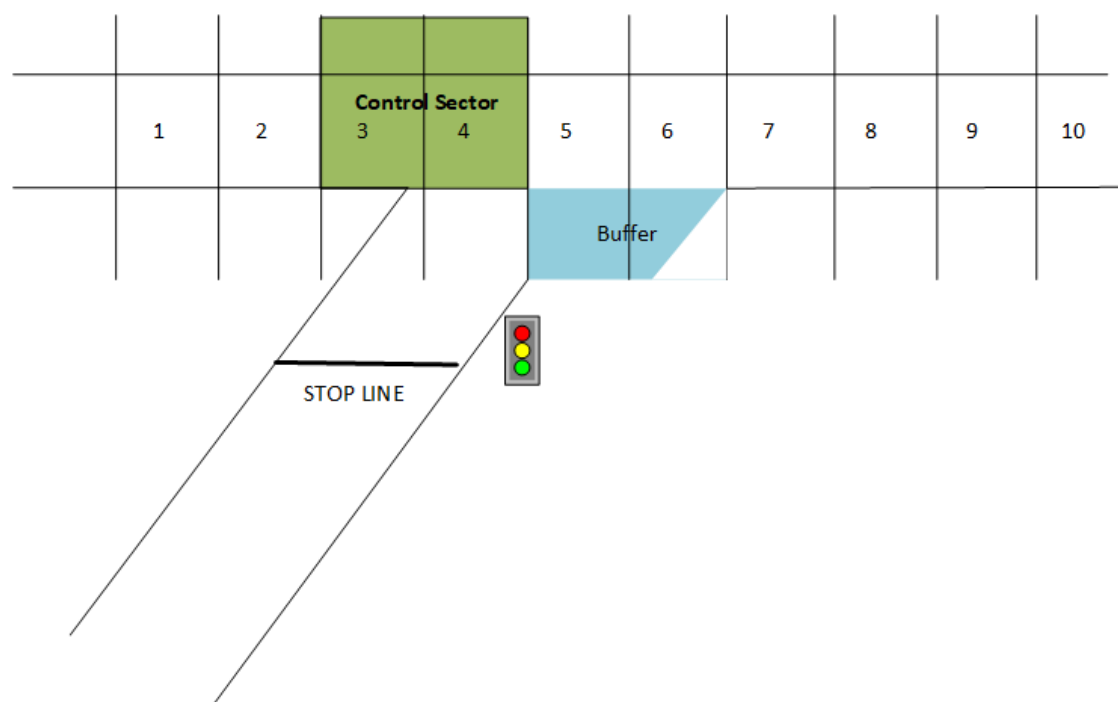
Ο συγκεκριμένος αλγόριθμος αποτελεί μία αποδοτική λύση για μικρά συστήματα που δεν έχουν μεγάλη πολυπλοκότητα όπως στην προκειμένη περίπτωση που έχουμε μόνο μία λωρίδα κυκλοφορίας, μία βοηθητική λωρίδα και μία ράμπα. Επιπλέον, χρειάζεται απλά ένας

αισθητήρας που δηλώνει την ύπαρξη ή όχι αυτοκινήτου στη βοηθητική λωρίδα και ένας μετρητής ταχύτητας επί του αυτοκινητόδρομου για να εξετάζει την ταχύτητα των οχημάτων στο τμήμα τρία (3). Η συγκεκριμένη μέθοδος θα μπορούσε χωρίς ιδιαίτερες απαιτήσεις να εφαρμοστεί σε μία πιλοτική εφαρμογή σε κάποια ράμπα εισόδου. Η απλότητά της επιτρέπει να μπορεί να εκτελεστεί και από πολύ απλό εξοπλισμό όπως έναν Προγραμματιζόμενο Λογικό Ελεγκτή (PLC).

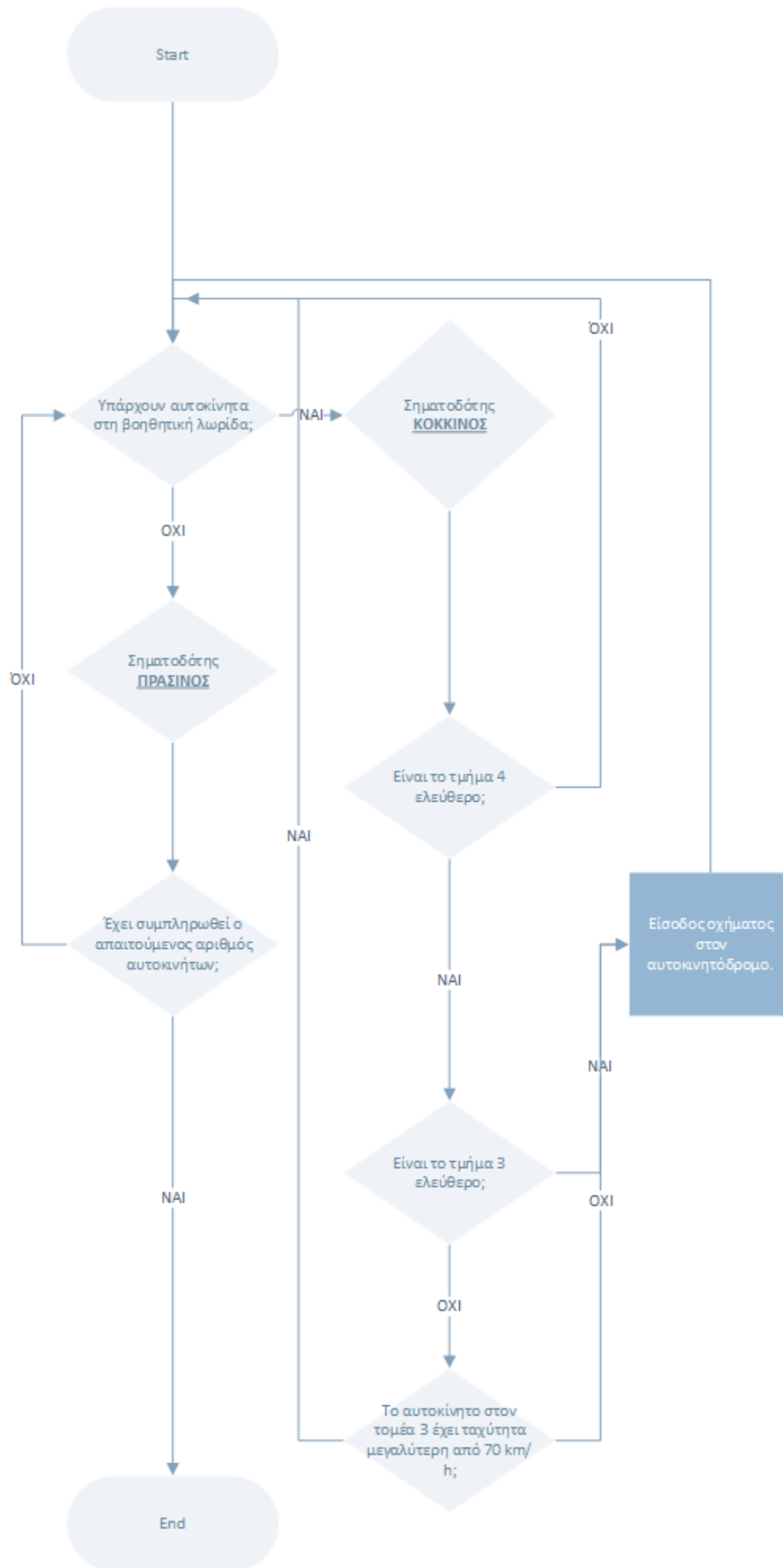
Τα στατιστικά που μπορεί να μας παράσχει η συγκεκριμένη μέθοδος είναι πόσα αυτοκίνητα πέρασαν από τη ράμπα συνολικά, πόσες επαναλήψεις εκτελέστηκαν, ποια είναι η μέση ταχύτητα κάθε στιγμή στο κύριο ρεύμα, ποια είναι η πυκνότητα, ο λόγος R όπως αυτός ορίστηκε σε παραπάνω κεφάλαιο καθώς και διάφορα άλλα στατιστικά στοιχεία που μπορούν να βοηθήσουν στη μελέτη του συστήματος.

Στο επόμενο υποκεφάλαιο θα μελετηθεί η υλοποίηση του ίδιου αλγορίθμου ελέγχου με τη διαφορά ότι η χωρητικότητα της βοηθητικής λωρίδας είναι πέντε (5) αυτοκίνητα, γεγονός που αυξάνει την πολυπλοκότητα των συνθηκών ελέγχου αλλά και των απαιτήσεων του συστήματος.

Παρακάτω παρουσιάζεται μία σχηματική απεικόνιση και ένα λογικό διάγραμμα για τον αλγόριθμο που αναπτύχθηκε στη συγκεκριμένη παράγραφο για την καλύτερη κατανόησή του.



Εικόνα 19. Ενδεικτική απεικόνιση διάταξης 1ου σεναρίου



Εικόνα 20. Λογικό διάγραμμα αλγορίθμου 1^{ου} σεναρίου υλοποίησης

3.5. 2^ο Σενάριο Υλοποίησης – 1 Λωρίδα & 1 ράμπα

Παρακάτω παρουσιάζεται μία ακόμα προσέγγιση του προβλήματος μίας (1) ράμπας και μίας λωρίδας (1). Άρα έχουμε τα εξής:

- Ο αυτοκινητόδρομος αποτελείται από μία (1) και μόνο λωρίδα
- Η ράμπα εισόδου αποτελείται από μία (1) και μόνο λωρίδα
- Η βοηθητική λωρίδα έχει χωρητικότητα μεγαλύτερη του ενός αυτοκινήτου. Το πλήθος των αυτοκινήτων που μπορούν να βρίσκονται εντός της βοηθητικής λωρίδας έχει οριστεί σε πέντε (5) αυτοκίνητα ωστόσο αυτό μπορεί εύκολα να αλλάχθει
- Ο αυτοκινητόδρομος χωρίζεται σε δέκα (10) τμήματα τα οποία διαθέτουν αισθητήρες οι οποίοι υποδηλώνουν την ύπαρξη (1) ή την απουσία (0) οχήματος στον εκάστοτε τομέα
- Κάθε αυτοκίνητο εισέρχεται από τη ράμπα στη βοηθητική λωρίδα (buffer) και από εκεί στον αυτοκινητόδρομο
- Ένα αυτοκίνητο εντός του αυτοκινητόδρομου μετακινείται από το τμήμα x_k στο τμήμα x_{k+1} από τη χρονική στιγμή t στη χρονική στιγμή $t+1$. Δηλαδή έστω ότι έχουμε την ακολουθία:

[0 1 1 0 1 0 0 0 1 1] τη

χρονική στιγμή t . Τότε τη χρονική στιγμή $t+1$ και με δεδομένο ότι κάποιο αυτοκίνητο δεν εισέρχεται μέσω της ράμπας στον αυτοκινητόδρομο αυτό θα γίνει:

[0 0 1 1 0 1 0 0 0 1].

- Τα τμήματα που καθορίζουν για το εάν θα μπει ένα αυτοκίνητο στον αυτοκινητόδρομο ή όχι εάν είναι κατειλημμένα είναι τα τμήματα 3 και 4
- Το σύνολο των αυτοκινήτων που εξετάζεται είναι πεπερασμένο και ο αλγόριθμος τερματίζει όταν εισέλθουν για παράδειγμα χίλια (1000) αυτοκίνητα από τη ράμπα

Ο αλγόριθμος ομοίως με αυτόν που παρουσιάστηκε στο προηγούμενο υποκεφάλαιο εξετάζει την κατάσταση της βοηθητικής λωρίδας, της ράμπας και του κυρίου ρεύματος του αυτοκινητόδρομου. Έτσι εάν η βοηθητική λωρίδα δεν είναι πλήρης (π.χ. πέντε αυτοκίνητα), και υπάρχει αυτοκίνητο στη ράμπα τότε ο φωτεινός σηματοδότης είναι πράσινος, διαφορετικά εφόσον ο buffer είναι γεμάτος ο φωτεινός σηματοδότης είναι κόκκινος. Προκειμένου ένα αυτοκίνητο που βρίσκεται στον buffer να μπορεί να εισέλθει στο κύριο ρεύμα του αυτοκινητόδρομου πρέπει είτε τα τμήματα 3 και 4 να είναι ελεύθερα είτε το τμήμα 4 να είναι ελεύθερο και στο τμήμα 3 το όχημα να κινείται με ταχύτητα κάτω από ένα όριο (π.χ. 70 km/h).

Ο συγκεκριμένος αλγόριθμος αποτελεί, ομοίως με τον προηγούμενο, μία αποδοτική λύση για μικρά συστήματα που δεν έχουν μεγάλη πολυπλοκότητα όπως στην προκειμένη περίπτωση που έχουμε μόνο μία λωρίδα κυκλοφορίας, μία βοηθητική λωρίδα και μία ράμπα. Η διαφοροποίηση με την αύξηση της βοηθητικής λωρίδας σε πέντε (5) αυτοκίνητα έγινε για

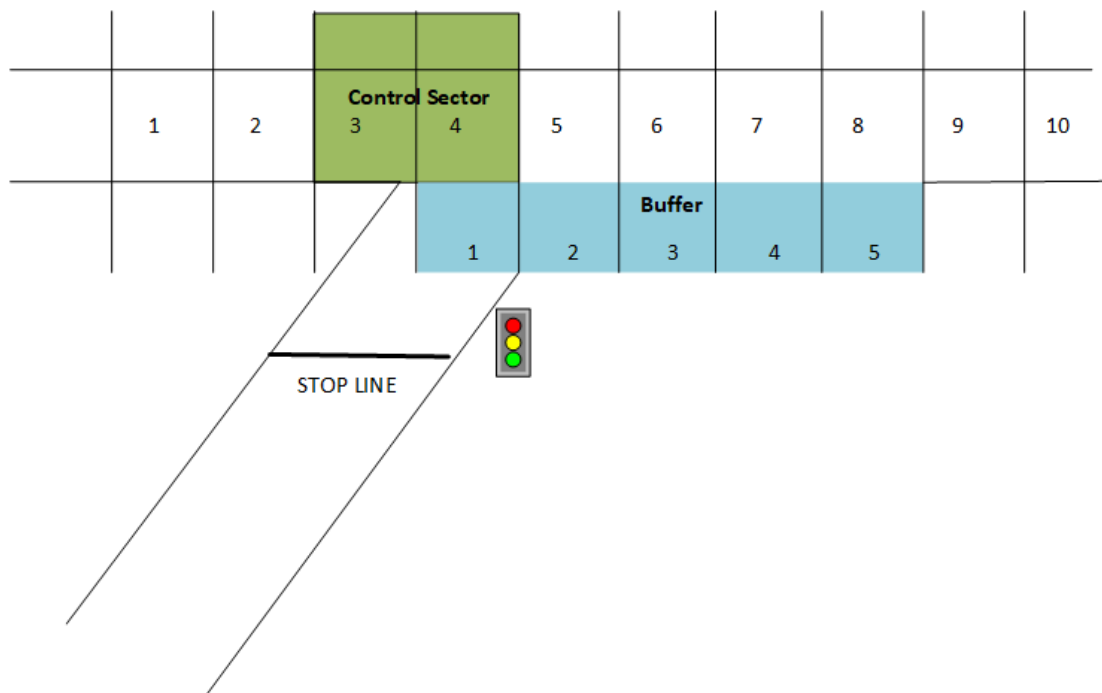
να είναι το μοντέλο πιο κοντά σε πραγματικά δεδομένα και να εξεταστεί κατά πόσο αυτό επηρεάζει την εκτέλεση του αλγορίθμου.

Επιπλέον, όπως έχει ήδη αναφερθεί χρειάζεται απλά ένας αισθητήρας που δηλώνει εάν είναι πλήρης η βοηθητική λωρίδα και ένας μετρητής ταχύτητας επί του αυτοκινητόδρομου για να εξετάζει την ταχύτητα των οχημάτων στο τμήμα 3. Εναλλακτικά θα μπορούσαν επί της βοηθητικής λωρίδας να εγκατασταθούν τόσοι αισθητήρες όσοι και τα τμήματα που υπάρχουν σε αυτή προκειμένου να είναι πιο ακριβής η πληροφορία που παράσχεται. Η συγκεκριμένη μέθοδος θα μπορούσε χωρίς ιδιαίτερες απαιτήσεις να εφαρμοστεί σε μία πιλοτική εφαρμογή σε κάποια ράμπα εισόδου.

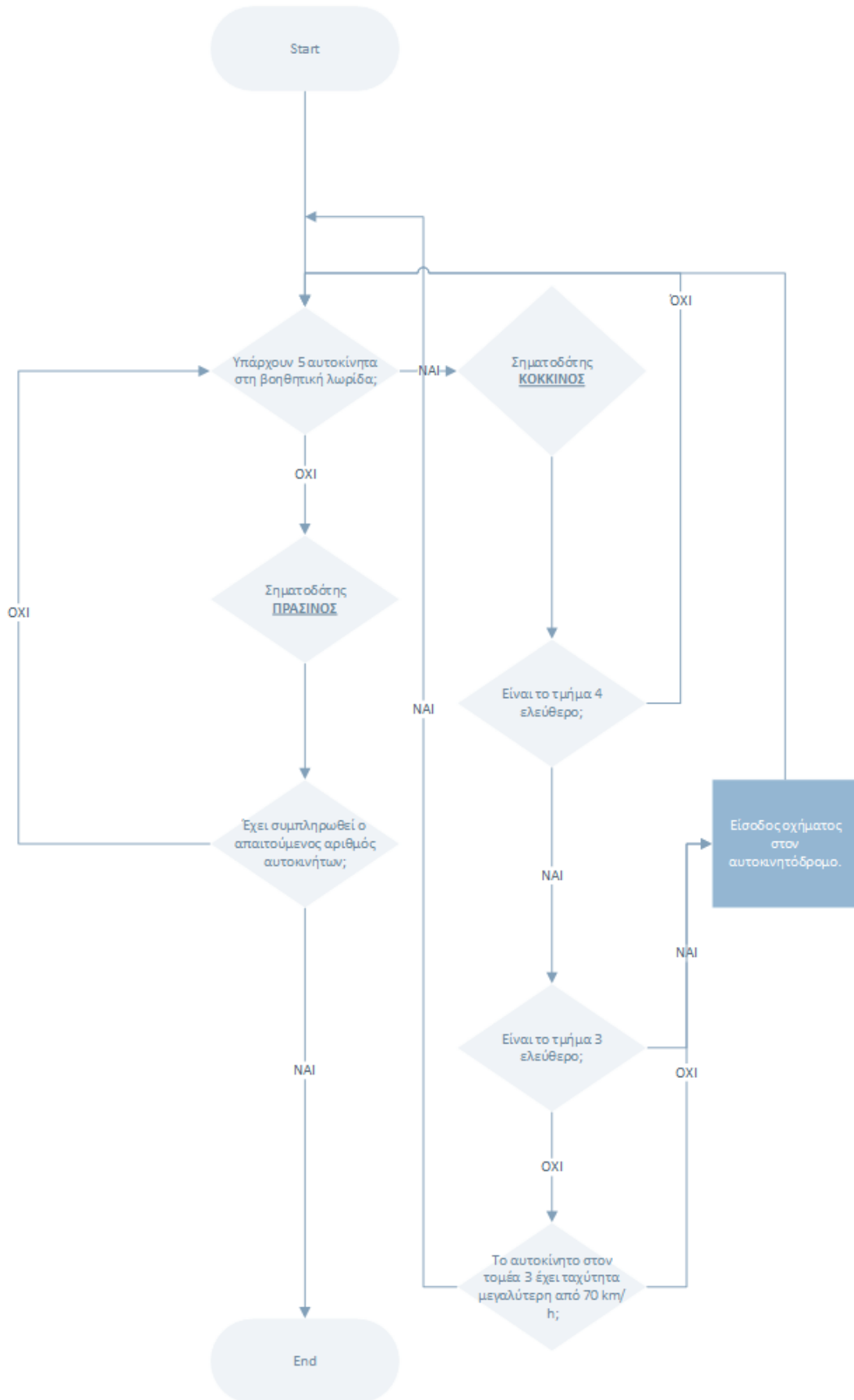
Τα στατιστικά που μπορεί να μας παράσχει η συγκεκριμένη μέθοδος είναι πόσα αυτοκίνητα πέρασαν από τη ράμπα συνολικά, πόσες επαναλήψεις εκτελέστηκαν, ποια είναι η μέση ταχύτητα κάθε στιγμή στο κύριο ρεύμα, ποια είναι η πυκνότητα, ο λόγος R όπως αυτός ορίστηκε σε παραπάνω κεφάλαιο καθώς και διάφορα άλλα στατιστικά στοιχεία που μπορούν να βοηθήσουν στη μελέτη του συστήματος όπως για παράδειγμα ο χρόνος εκτέλεσης του αλγορίθμου.

Στην επόμενη προσέγγιση θα επεκταθεί ο συγκεκριμένος αλγόριθμος. Πιο συγκεκριμένα στο επόμενο υποκεφάλαιο θα μελετηθεί η υλοποίηση του ελέγχου σε αυτοκινητόδρομο τριών (3) λωρίδων στον οποίο τα αυτοκίνητα θα μπορούν να αλλάζουν θέση όχι μόνο προς τα εμπρός αλλά και από τη μία λωρίδα στην άλλη επηρεάζοντας έτσι τον έλεγχο του σηματοδότη.

Παρακάτω παρουσιάζεται μία σχηματική απεικόνιση και ένα λογικό διάγραμμα για τον αλγόριθμο που αναπτύχθηκε στη συγκεκριμένη παράγραφο για την καλύτερη κατανόησή του.



Εικόνα 21. Ενδεικτική απεικόνιση διάταξης 2ου σεναρίου



Εικόνα 22. Λογικό διάγραμμα 2^{ον} σεναρίου υλοποίησης με αυξημένη χωρητικότητα buffer

3.6. 3^ο Σενάριο Υλοποίησης – 3 Λωρίδες & 1 ράμπα

Παρακάτω παρουσιάζεται μία ακόμα προσέγγιση του προβλήματος με μία (1) ράμπα και τρεις (3) λωρίδες. Άρα έχουμε τα εξής:

- Ο αυτοκινητόδρομος αποτελείται από τρεις (3) λωρίδες
- Η ράμπα εισόδου αποτελείται από μία (1) λωρίδα
- Η βοηθητική λωρίδα έχει χωρητικότητα μεγαλύτερη του ενός αυτοκινήτου. Το πλήθος των αυτοκινήτων που μπορούν να βρίσκονται εντός της βοηθητικής λωρίδας έχει οριστεί σε πέντε (5) αυτοκίνητα ωστόσο αυτό μπορεί εύκολα να αλλαχθεί
- Ο αυτοκινητόδρομος χωρίζεται σε τριάντα (30) τμήματα, δέκα (10) ανά λωρίδα τα οποία διαθέτουν αισθητήρες οι οποίοι υποδηλώνουν την ύπαρξη (1) ή την απουσία (0) οχήματος στον εκάστοτε τομέα
- Κάθε αυτοκίνητο εισέρχεται από τη ράμπα στη βοηθητική λωρίδα (buffer), και από εκεί στον αυτοκινητόδρομο
- Ένα αυτοκίνητο εντός του αυτοκινητόδρομου μετακινείται μόνο προς τα εμπρός ή προς τα εμπρός και προς τα αριστερά ταυτόχρονα από τη μία χρονική στιγμή στην επόμενη. Ένα στιγμιότυπο της κατάστασης των τριών (3) λωρίδων θα είναι για παράδειγμα το παρακάτω:

0	1	1	1	0	0	1	1	0	1
1	1	1	0	0	1	0	1	0	0
1	1	1	0	0	1	1	0	1	1

Η αρίθμηση των θέσεων έχει ως εξής:

- Δεξιά λωρίδα θέσεις 1 έως 10
- Μεσαία λωρίδα θέσεις 11 έως 20
- Αριστερή λωρίδα θέσεις 21 έως 30
- Τα τμήματα που καθορίζουν για το εάν θα μπει ένα αυτοκίνητο στον αυτοκινητόδρομο ή όχι εάν είναι κατειλημμένα είναι τα τμήματα 3 και 4
- Ο αλγόριθμος μπορεί να λειτουργεί συνεχώς εφόσον το επιτρέπει το υλικό μας ή μπορεί να σταματήσει μετά από ένα συγκεκριμένο χρονικό διάστημα. Εναλλακτικά, του σεναρίου με τον πεπερασμένο χρόνο εκτέλεσης μπορεί να εκτελεστεί πεπερασμένος αριθμός επαναλήψεων πριν τον τερματισμό του αλγορίθμου.

Ο αλγόριθμος ομοίως με όσα παρουσιάστηκαν στα προηγούμενα υποκεφάλαια εξετάζει την κατάσταση της βοηθητικής λωρίδας, της ράμπας και του κυρίου ρεύματος του αυτοκινητόδρομου. Λόγω της αύξησης των λωρίδων ο συγκεκριμένος αλγόριθμος παρουσιάζει αυξημένη πολυπλοκότητα η οποία έγκειται στην προσομοίωση της κίνησης σε τρεις (3) λωρίδες. Έτσι εάν η βοηθητική λωρίδα δεν είναι πλήρης (π.χ. πέντε αυτοκίνητα), και υπάρχει αυτοκίνητο στη ράμπα τότε ο φωτεινός σηματοδότης είναι πράσινος, διαφορετικά εφόσον ο

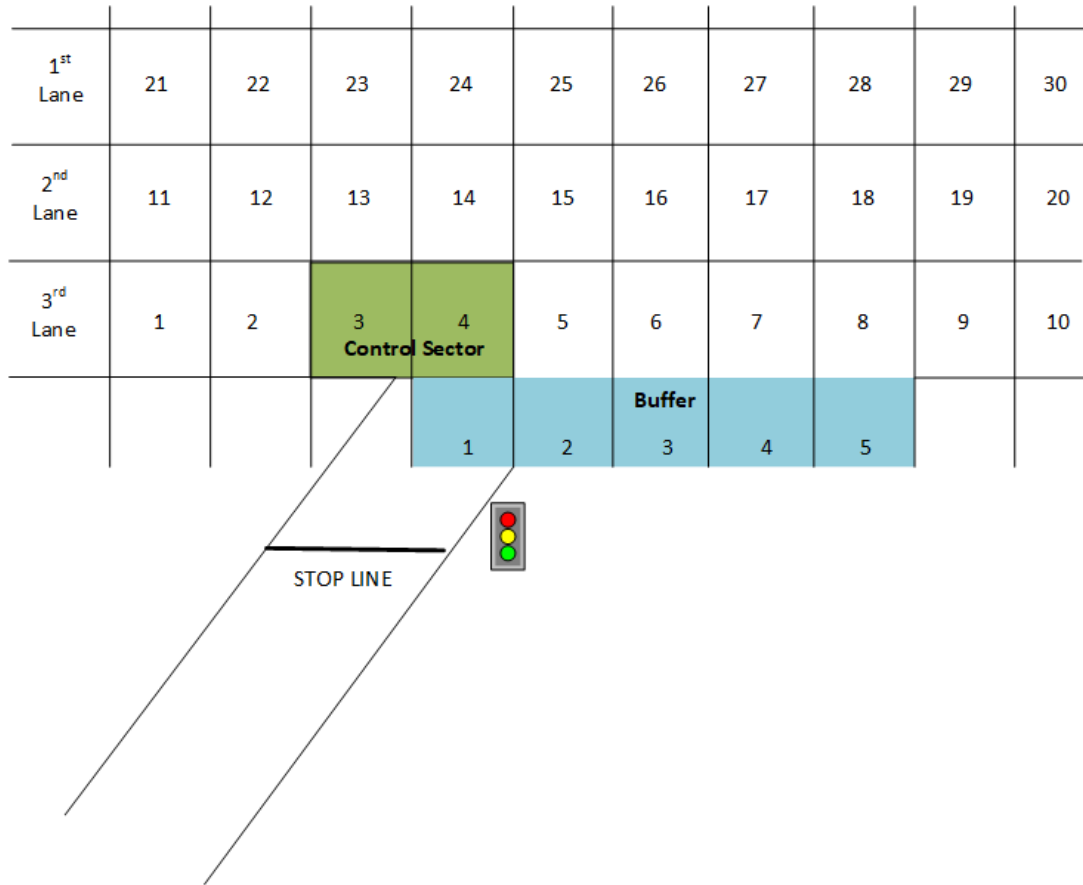
buffer είναι γεμάτος ο φωτεινός σηματοδότης είναι κόκκινος. Προκειμένου ένα αυτοκίνητο που βρίσκεται στον buffer να μπορεί να εισέλθει στο κύριο ρεύμα του αυτοκινητόδρομου πρέπει είτε τα τμήματα 3 και 4 να είναι ελεύθερα είτε το τμήμα 4 να είναι ελεύθερο και στο τμήμα 3 το όχημα να κινείται με ταχύτητα κάτω από ένα όριο (π.χ. 70 km/h).

Ο συγκεκριμένος αλγόριθμος αποτελεί μία αποδοτική λύση για ένα τμήμα αυτοκινητόδρομου με τρεις (3) λωρίδες, μέγεθος το οποίο συναντάται πολύ συχνά και στην πράξη. Η βοηθητική λωρίδα έχει χωρητικότητα πέντε (5) αυτοκίνητα και η συνθήκη ελέγχου έχει να κάνει τόσο με την κίνηση στο κυρίως ρεύμα όσο και με το εάν ο buffer είναι γεμάτος.

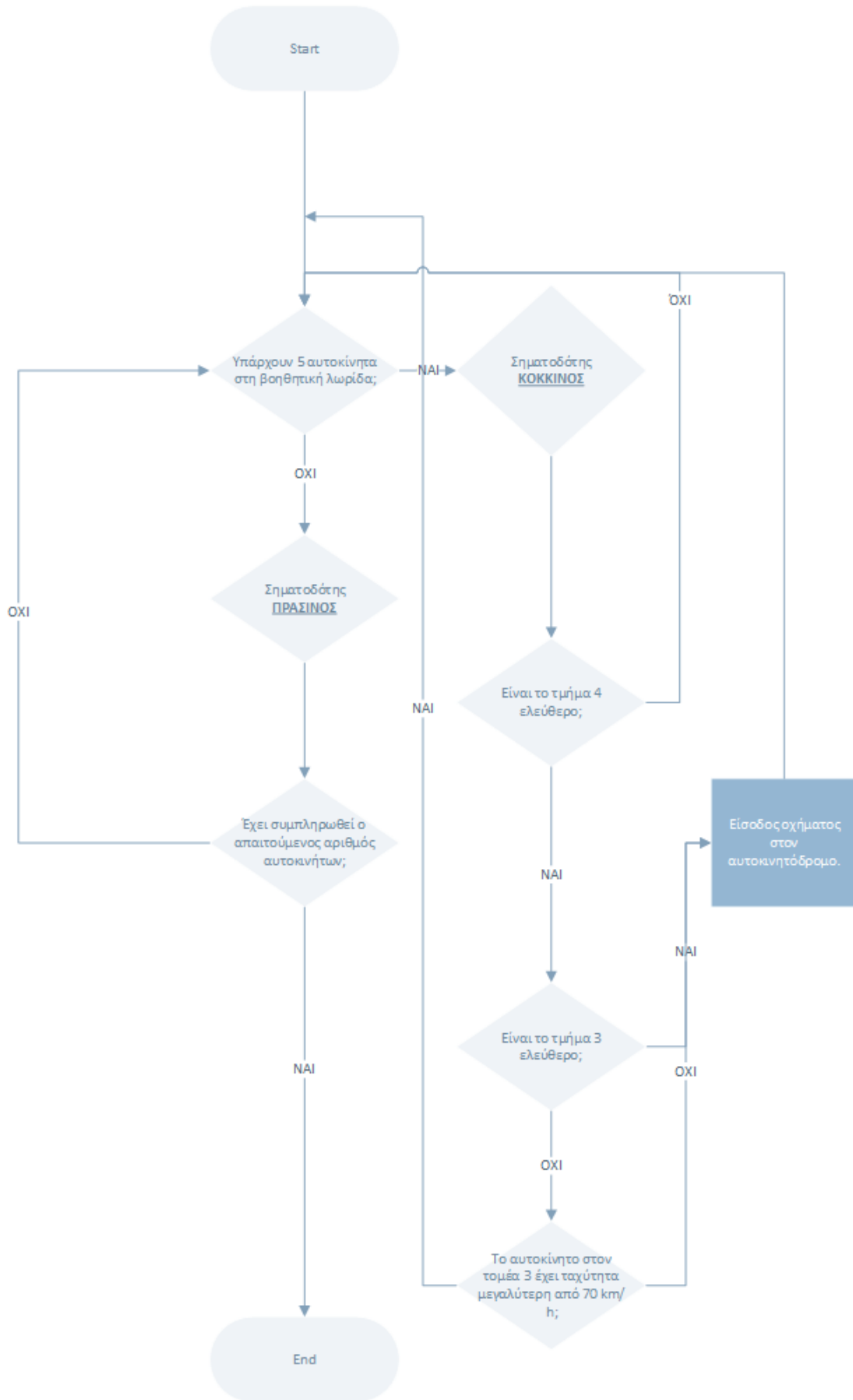
Τα στατιστικά που μπορεί να μας παράσχει η συγκεκριμένη μέθοδος είναι πόσα αυτοκίνητα πέρασαν από τη ράμπα συνολικά, πόσες επαναλήψεις εκτελέστηκαν, ποια είναι η μέση ταχύτητα κάθε στιγμή στο κύριο ρεύμα, ποια είναι η πυκνότητα, ο λόγος R όπως αυτός ορίστηκε σε παραπάνω κεφάλαιο, ο χρόνος εκτέλεσης του αλγορίθμου καθώς και διάφορα άλλα στατιστικά στοιχεία που μπορούν να βοηθήσουν στη μελέτη του συστήματος. Λόγω της ύπαρξης των τριών (3) λωρίδων τα στατιστικά παρουσιάζουν ιδιαίτερη αύξηση στην πολυπλοκότητα εξαγωγής τους. Πλέον όλα τα στατιστικά συμπεριλαμβάνουν όλες τις λωρίδες και όχι μόνο τη δεξιά που συμμετέχει στον έλεγχο της εισόδου από τη ράμπα.

Στην επόμενη προσέγγιση θα γίνει επέκταση του συγκεκριμένου σεναρίου στο οποίο θα υπάρχουν τρεις (3) λωρίδες και δύο (2) ράμπες. Το συγκεκριμένο σύστημα έχει ιδιαίτερα αυξημένη πολυπλοκότητα καθώς εμπλέκεται μεγάλο πλήθος ελέγχων και συνθηκών που πρέπει να εξεταστούν και να ληφθούν υπόψη.

Παρακάτω παρουσιάζεται μία ενδεικτική απεικόνιση και ένα λογικό διάγραμμα για τον αλγόριθμο που αναπτύχθηκε στη συγκεκριμένη παράγραφο για την καλύτερη κατανόηση του.



Εικόνα 23. Ενδεικτική απεικόνιση διάταξης 3ου σεναρίου με τρεις (3) λωρίδες



Εικόνα 24. Λογικό διάγραμμα αλγορίθμου 3^{ου} σεναρίου υλοποίησης με τρεις (3) λωρίδες

3.7. 4^ο Σενάριο Υλοποίησης – 3 Λωρίδες & 2 ράμπες

Παρακάτω παρουσιάζεται η τελική προσέγγιση του προβλήματος πριν την υλοποίηση του νευρωνικού με δύο (2) ράμπες και τρεις (3) λωρίδες με τις εξής παραδοχές:

- Ο αυτοκινητόδρομος αποτελείται από τρεις (3) λωρίδες
- Υπάρχουν δύο (2) ράμπες εισόδου
- Οι ράμπες εισόδου αποτελούνται από μία (1) λωρίδα
- Οι δύο (2) βοηθητικές λωρίδες έχουν χωρητικότητα μεγαλύτερη του ενός αυτοκινήτου. Το πλήθος των αυτοκινήτων που μπορούν να βρίσκονται εντός κάθε βοηθητικής λωρίδας έχει οριστεί σε πέντε (5) αυτοκίνητα ωστόσο αυτό μπορεί εύκολα να αλλαχθεί ανάλογα με τις ανάγκες του συστήματος
- Ο αυτοκινητόδρομος χωρίζεται σε σαράντα πέντε (45) τμήματα, δεκαπέντε (15) ανά λωρίδα τα οποία διαθέτουν αισθητήρες οι οποίοι υποδηλώνουν την ύπαρξη (1) ή την απουσία (0) οχήματος στον εκάστοτε τομέα. Ο αριθμός των τμημάτων του αυτοκινητόδρομου μπορεί να αλλάξει ανάλογα με τις ανάγκες του συστήματος.
- Κάθε αυτοκίνητο εισέρχεται από την εκάστοτε ράμπα στην αντίστοιχη βοηθητική λωρίδα (buffer) και από εκεί στον αυτοκινητόδρομο
- Ένα αυτοκίνητο εντός του αυτοκινητόδρομου μετακινείται μόνο προς τα εμπρός και προς τα αριστερά από τη μία χρονική στιγμή στην επόμενη. Ένα στιγμιότυπο της κατάστασης των τριών (3) λωρίδων θα είναι για παράδειγμα το παρακάτω:

0	1	1	1	0	0	1	1	0	1	1	1	0	0	0
1	1	1	0	0	1	0	1	0	0	0	0	0	0	1
1	1	1	0	0	1	1	0	1	1	0	1	1	0	0

Η αρίθμηση των θέσεων έχει ως εξής:

- Δεξιά λωρίδα θέσεις 1 έως 15
- Μεσαία λωρίδα θέσεις 16 έως 30
- Αριστερή λωρίδα θέσεις 31 έως 45
- Τα τμήματα που καθορίζουν για το εάν θα μπει ένα αυτοκίνητο στον αυτοκινητόδρομο ή όχι εάν είναι κατειλημμένα είναι τα τμήματα 4 και 10, τα οποία βρίσκονται πριν από κάθε ράμπα αντίστοιχα
- Κάθε χρονική στιγμή μπορεί να εισέρχεται ή όχι ένα αυτοκίνητο σε κάθε λωρίδα ή σε κάθε ράμπα με τυχαίο τρόπο
- Ο αλγόριθμος μπορεί να λειτουργεί συνεχώς εφόσον το επιτρέπει το υλικό μας ή μπορεί να σταματήσει μετά από ένα συγκεκριμένο χρονικό διάστημα. Εναλλακτικά, του σεναρίου με τον πεπερασμένο χρόνο εκτέλεσης μπορεί να εκτελεστεί πεπερασμένος αριθμός επαναλήψεων πριν τον τερματισμό του αλγορίθμου.

Ο εν λόγω αλγόριθμος εξετάζει την κατάσταση της βοηθητικής λωρίδας, της ράμπας και του κυρίου ρεύματος του αυτοκινητόδρομου. Λόγω της αύξησης των λωρίδων καθώς και η εισαγωγή της επιπλέον ράμπας ο συγκεκριμένος αλγόριθμος παρουσιάζει αρκετά αυξημένη πολυπλοκότητα και αποτελεί μία επέκταση των όσων έχουν μελετηθεί μέχρι τώρα τόσο στη συγκεκριμένη εργασία όσο και γενικότερα στην υπάρχουσα βιβλιογραφία. Οι ανάγκες ελέγχου πλέον πολλαπλασιάζονται καθώς οι δύο (2) ράμπες πέρα από τον έλεγχο που εκτελείται σε καθεμία ξεχωριστά πρέπει να λειτουργούν και σαν ένα ενιαίο σύστημα καθώς οι ενέργειες κάθε ράμπας επηρεάζουν τον αυτοκινητόδρομο σαν σύνολο.

Πιο συγκεκριμένα κάθε ράμπα εξετάζει την πυκνότητα στο σύνολο του αυτοκινητόδρομου που βρίσκεται πριν από αυτή και ανάλογα ρυθμίζει τη διάρκεια την οποία ο φωτεινός σηματοδότης είναι κόκκινος. Επιπλέον, η πρώτη ράμπα εξετάζει το σύνολο της πυκνότητας και μετά από αυτή καθώς εάν εισέρχονται συνεχώς αυτοκίνητα από τη ράμπα 1 τότε η ράμπα 2 θα μένει κλειστή με αποτέλεσμα να εμφανίζεται κυκλοφοριακή συμφόρηση σε αυτή. Έτσι για κάθε ράμπα ορίζονται τρία διαστήματα πυκνότητας για το εξεταζόμενο τμήμα του αυτοκινητόδρομου τα οποία αντιστοιχίζονται το καθένα σε μία χρονική διάρκεια, για παράδειγμα:

- Πυκνότητα 0%-30% διάρκεια κόκκινης ένδειξης σηματοδότη ένα (1) λεπτό
- Πυκνότητα 30%-60% διάρκεια κόκκινης ένδειξης σηματοδότη δύο (2) λεπτά
- Πυκνότητα πάνω από 60% διάρκεια κόκκινης ένδειξης σηματοδότη τέσσερα (4) λεπτά

Το τμήμα του οποίου η πυκνότητα επηρεάζει την πρώτη ράμπα είναι από τον τομέα 0 έως τον τομέα 4 κάθε λωρίδας και για τη δεύτερη ράμπα είναι από τον τομέα 6 έως τον τομέα 10. Εάν η ολική πυκνότητα του αυτοκινητόδρομου ξεπερνάει το 70% τότε οι δύο ράμπες παραμένουν κλειστές έως ότου μειωθεί η πυκνότητα κάτω από αυτό το όριο. Η πράσινη ένδειξη του φωτεινού σηματοδότη θα διαρκεί συγκεκριμένη διάρκεια κατά την οποία θα μπορεί να περάσει συγκεκριμένος αριθμός αυτοκινήτων, για παράδειγμα πέντε (5) αυτοκίνητα τη φορά.

Ο έλεγχος εισόδου από τη βοηθητική λωρίδα στο κεντρικό ρεύμα ακολουθεί την ίδια λογική που παρουσιάστηκε και στα προηγούμενα υποκεφάλαια. Εάν ο εκάστοτε buffer είναι πλήρης ο φωτεινός σηματοδότης παραμένει κόκκινος ανεξάρτητα από το εάν έχει παρέλθει ο χρόνος που έχει οριστεί με βάση την πυκνότητα. Ένα αυτοκίνητο μπορεί να εισέλθει στο κύριο ρεύμα του αυτοκινητόδρομου από τη βοηθητική λωρίδα της ράμπας 1 όταν το τμήμα 4 της δεξιάς λωρίδας είναι ελεύθερο και αντίστοιχα από τη ράμπα 2 όταν το τμήμα 10 της δεξιάς λωρίδας είναι ελεύθερο.

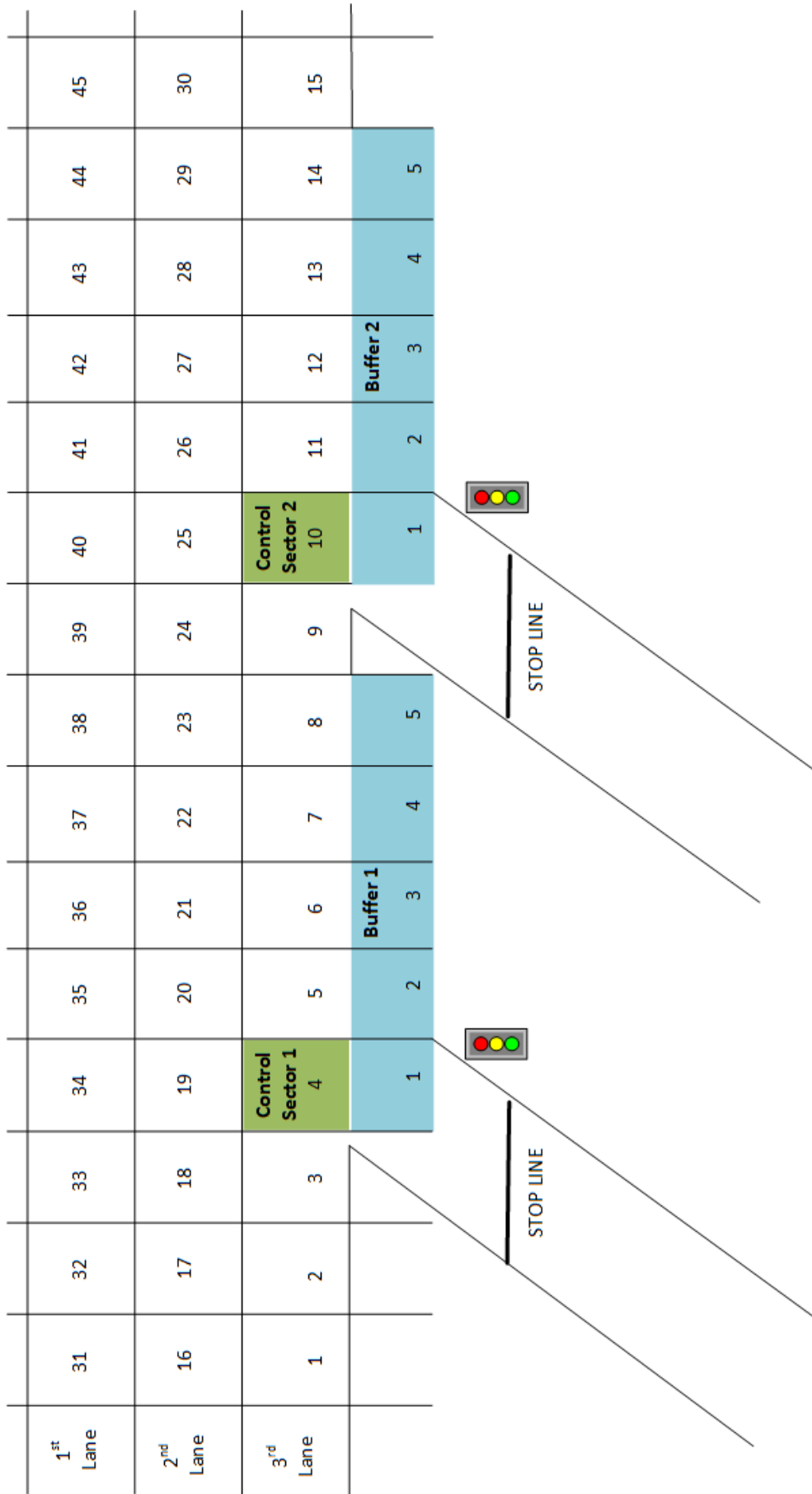
Ο συγκεκριμένος αλγόριθμος αποτελεί μία προσέγγιση η οποία βασίζεται σε ρεαλιστικές συνθήκες με ιδιαίτερα αυξημένη πολυπλοκότητα και πολύ υψηλές υπολογιστικές απαιτήσεις. Επισημαίνεται, ότι η αποτύπωση του προβλήματος με δύο (2) ράμπες και

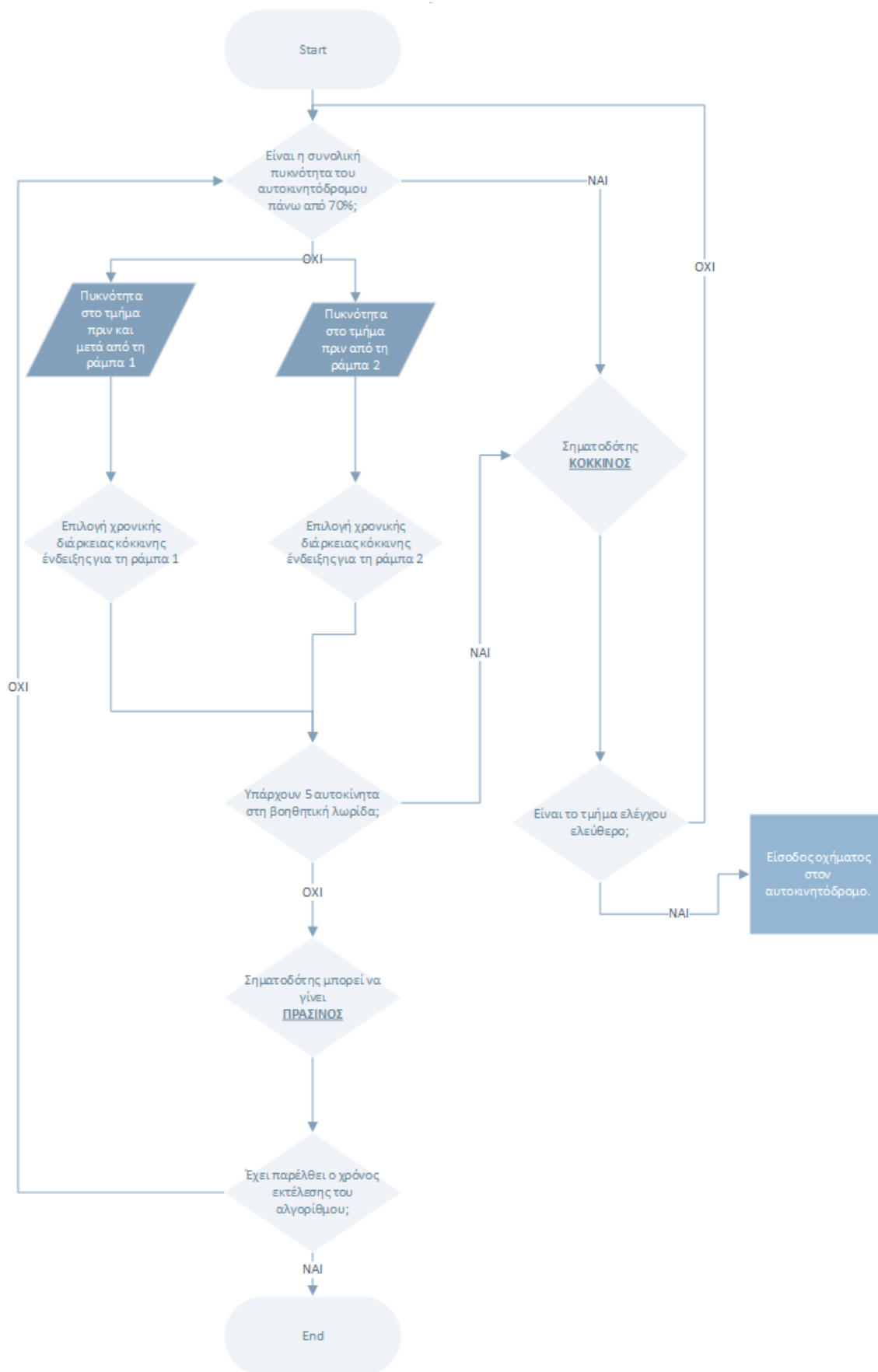
πολλαπλές λωρίδες είναι κάτι το οποίο συναντάται ελάχιστα στη διεθνή βιβλιογραφία ή απλά αναφέρεται σαν επέκταση του προβλήματος του Ramp Metering.

Τα στατιστικά που μπορεί να μας παράσχει η συγκεκριμένη μέθοδος είναι πόσα αυτοκίνητα πέρασαν από τη ράμπα συνολικά, πόσο χρόνο έτρεξε ο αλγόριθμος, ποια είναι η μέση ταχύτητα κάθε στιγμή σε όλες τις λωρίδες στο κύριο ρεύμα, ποια είναι η πυκνότητα στο σύνολο του αυτοκινητόδρομου και ανά λωρίδα, ο λόγος R όπως αυτός ορίστηκε σε προηγούμενο κεφάλαιο καθώς και διάφορα άλλα στατιστικά στοιχεία που μπορούν να βοηθήσουν στη μελέτη του συστήματος. Λόγω της ύπαρξης των τριών (3) λωρίδων καθώς και της δεύτερης ράμπας τα στατιστικά παρουσιάζουν ιδιαίτερη αύξηση στην πολυπλοκότητα εξαγωγής τους. Πλέον όλα τα στατιστικά συμπεριλαμβάνουν όλες τις λωρίδες και όχι μόνο τη δεξιά που συμμετέχει στον έλεγχο της εισόδου από τη ράμπα. Λόγω της αυξημένης πολυπλοκότητας του αλγορίθμου ως συνθήκη περάτωσης τέθηκε ο χρόνος δηλαδή μετά το πέρασ πέντε (5) λεπτών.

Στην επόμενη προσέγγιση η οποία αποτελεί το τελευταίο στάδιο το οποίο εξετάζεται στην παρούσα εργασία θα αναλυθεί η μεθοδολογία και η υλοποίηση του νευρωνικού δικτύου που αποτέλεσε από την αρχή τον βασικό στόχο της παρούσας εργασίας. Η αναγκαιότητα ανάπτυξης ενός νευρωνικού δικτύου προκύπτει εύκολα εάν εξετάσει κανείς και τρέξει τον παραπάνω αλγόριθμο σε έναν Η/Υ. Πρόκειται για έναν αλγόριθμο ο οποίος παράγει με μεγάλη ταχύτητα τεράστιο πλήθος δεδομένων το οποίο επιβαρύνει σε μεγάλο βαθμό τον χρόνο εκτέλεσης κάθε πράξης ελέγχου και άρα μειώνει τη γενικότερη απόδοση του συστήματος. Το συγκεκριμένο μοντέλο που εξετάστηκε είναι ιδιαίτερα αποδοτικό σε συστήματα που περιγραφήκαν σε προηγούμενα κεφάλαια και τα οποία έχουν απλές απαιτήσεις. Από τη στιγμή όμως που το σύστημα επεκτείνεται με πολλαπλές λωρίδες και ράμπες τότε τέτοιου είδους αλγόριθμοι δεν μπορούν να προσαρμοστούν με ταχύτητα στις συνεχείς και ταχείες αλλαγές και έτσι καθίσταται επιτακτική η ανάγκη ανάπτυξης ενός νευρωνικού δικτύου το οποίο θα εκπαιδευθεί κατάλληλα και θα μπορεί να ανταποκριθεί με ταχύτητα στις συνεχείς μεταβολές των συνθηκών.

Παρακάτω παρουσιάζεται μία ενδεικτική απεικόνιση και ένα λογικό διάγραμμα για τον αλγόριθμο που αναπτύχθηκε στη συγκεκριμένη παράγραφο για την καλύτερη κατανόηση του.





Εικόνα 25. Λογικό διάγραμμα αλγορίθμου 4^{ου} σεναρίου υλοποίησης με τρεις (3) λωρίδες και δύο (2) ράμπες

Το παραπάνω λογικό διάγραμμα φαίνεται εν γένει πιο απλό από τα προηγούμενα που παρουσιάστηκαν, κάτι τέτοιο όμως δεν ισχύει στην πράξη. Το παραπάνω διάγραμμα απεικονίζει μία γενικευμένη εικόνα του ελέγχου που απαιτείται. Επιπλέον, η ανάλυση σε λογικό διάγραμμα με λεπτομερή τρόπο του αλγορίθμου θα δημιουργούσε ένα δυσνόητο αποτέλεσμα το οποίο δεν εξυπηρετεί τον σκοπό του και θα συγχίσει τον αναγνώστη.

3.8. Αλγόριθμος Νευρωνικού Δικτύου

Έπειτα από την ολοκλήρωση όλων των σεναρίων ελέγχου των προηγούμενων παραγράφων και έχοντας δημιουργήσει τα απαραίτητα δεδομένα εκπαίδευσης και επαλήθευσης για ένα νευρωνικό δίκτυο έγινε η ανάπτυξη αυτού με τη χρήση της βιβλιοθήκης Keras. Το νευρωνικό αυτό δίκτυο εξετάζει ένα σύστημα όπως αυτό που περιγράφηκε στην προηγούμενη παράγραφο με τρεις (3) λωρίδες και δύο (2) ράμπες.

Στόχος αυτού του αλγορίθμου μηχανικής μάθησης είναι να υπολογίζει τον χρόνο για κάθε ράμπα κατά τον οποίο ο φωτεινός σηματοδότης θα είναι κόκκινος με δεδομένες τις συνθήκες που επικρατούν στον αυτοκινητόδρομο. Ο υπολογισμός αυτός γίνεται ακαριαία αφού το δίκτυο έχει εκπαιδευτεί και έπειτα, καθώς έχοντας εκτελέσει την εκπαίδευση με πληθώρα δεδομένων μπορεί να προσαρμόσει αποδοτικά οποιοσδήποτε συνθήκες και να τις αντιστοιχίσει στους κατάλληλους χρόνους.

Με βάση λοιπόν και τη βιβλιογραφία, η βιβλιοθήκη Keras που χρησιμοποιήθηκε έχει τα εξής χαρακτηριστικά:

- Είναι ένα API (Application Programming Interface) υψηλού επιπέδου νευρωνικού δικτύου το οποίο είναι γραμμένο σε Python.
- Μπορεί να τρέξει πάνω από την Tensorflow, το CNTK ή το Theano.
- Μπορεί να χρησιμοποιηθεί ως βιβλιοθήκη βαθιάς μάθησης (Deep Learning).
- Είναι γρήγορη κι εύκολη.
- Λειτουργεί απρόσκοπτα σε CPU και GPU.

Για τις ανάγκες της προσομοίωσης δημιουργήθηκε ένα νευρωνικό δίκτυο για την ταξινόμηση ενός μοντέλου που αποτελείται από πολλαπλές κλάσεις (multiclass classification). Το σύνολο δεδομένων τύπου csv αποτελείται από 118015 παρατηρήσεις, με 48 μεταβλητές εισόδου (input variables) και δύο (2) μεταβλητές εξόδου (output variables) . Αναλυτικότερα, οι μεταβλητές που χρησιμοποιούνται:

- Η ύπαρξη αυτοκινήτου (1) ή όχι (0) σε κάθε ένα τμήμα του αυτοκινητόδρομου. Έχει θεωρηθεί ότι η μοντελοποίηση του αυτοκινητόδρομου αποτελείται από 45 τμήματα (sectors), όσοι και οι αισθητήρες, που τοποθετούνται.
- Η μέση πυκνότητα (Average Density) των τριών (3) λωρίδων του αυτοκινητόδρομου.

- Η μέση ταχύτητα (Average Speed) των τριών (3) λωρίδων του αυτοκινητόδρομου.
- Ο μέσος λόγος πυκνότητας προς ταχύτητας (Average Ratio) των τριών (3) λωρίδων του αυτοκινητόδρομου.
- Δύο (2) μεταβλητές εξόδου, οι οποίες χρησιμοποιούνται στην υλοποίηση διαφορετικών μοντέλων πρόβλεψης, για τους χρόνους που μένει η ράμπα 1 (r1_t1) και ράμπα 2 κλειστή (r2_t2). Κάθε μια από αυτές τις στήλες αποτελείται από 4 τιμές κατηγοριοποίησης (categorical values), και πιο συγκεκριμένα τις τιμές 0,1,2,4, που αναφέρονται στο χρόνο που μένει η (εκάστοτε) ράμπα κλειστή για το συγκεκριμένο χρονικό διάστημα.

Αρχικά έγινε εισαγωγή των απαραίτητων βιβλιοθηκών keras, pandas και numpy, μαζί με τη βιβλιοθήκη οπτικοποίησης δεδομένων matplotlib, καθώς και των λοιπών βιβλιοθηκών για τη δημιουργία του μοντέλου πρόβλεψης.

Στη συνέχεια πραγματοποιήθηκε ανάγνωση του αρχείου με το σύνολο των δεδομένων (dataset) τύπου csv, και φόρτωση αυτών που χρησιμοποιήθηκαν για τη μοντελοποίηση σε ένα νέο σύνολο δεδομένων τύπου dataframe. Σκοπός αυτού, ήταν η καλύτερη κατανόηση των στηλών του dataset και του τύπου των δεδομένων που περιέχονται σε αυτές.

	1	2	3	4	5	6	7	8	9	10	...	39	40	41	42	43	44	45	Av_Cap	Av_Rat	Av_Sp
0	1	1	0	1	0	0	1	0	1	1	...	1	1	0	0	0	0	1	44	0.42	105.67
1	1	1	1	0	1	0	0	1	0	1	...	0	1	1	1	0	0	0	48	0.47	101.67
2	0	1	1	1	0	1	0	0	1	0	...	1	0	1	1	1	0	0	51	0.51	100.67
3	1	0	1	1	1	0	1	0	0	1	...	0	1	0	1	1	1	0	55	0.55	99.33
4	1	1	0	1	1	1	0	1	0	0	...	1	0	1	1	1	1	1	60	0.63	95.33

Εικόνα 26. Στιγμιότυπο δεδομένων εκπαίδευσης νευρωνικού δικτύου

Όπως γίνεται άμεσα αντιληπτό, όλες οι μεταβλητές εισόδου είναι αριθμητικές και συνεχείς, ενώ δεν έχουν κανένα στοιχείο τύπου κατηγοριοποίησης (categorical type variable). Καθώς δεν υπάρχει καμία μεταβλητή κατηγοριοποίησης δεν χρειάζεται κατ' αντιστοιχία μετατροπή δεδομένων αυτού του τύπου των μεταβλητών.

Δεδομένου ότι τα χαρακτηριστικά εισόδου του δικτύου βρισκόταν σε διαφορετικές κλίμακες, έπρεπε να γίνει τυποποίηση στην είσοδο του μοντέλου σε συγκεκριμένη κλίμακα.

Κατόπιν της ολοκλήρωσης της προ-επεξεργασίας των δεδομένων ακολούθησε η κατασκευή του νευρωνικού δικτύου. Χρησιμοποιήθηκε η βιβλιοθήκη keras προκειμένου να δημιουργηθούν τα απαιτούμενα επίπεδα νευρωνικού δικτύου. Υπάρχουν δύο (2) κύριοι τύποι μοντέλων που είναι διαθέσιμοι στη βιβλιοθήκη Keras, οι Sequential και Model. Στο εν λόγω νευρωνικό δίκτυο έγινε χρήση του Sequential μοντέλου για την υλοποίηση του νευρωνικού δικτύου.

Επιπλέον, χρησιμοποιήθηκε η βιβλιοθήκη dense για τη δημιουργία των απαιτούμενων στρωμάτων του δικτύου. Το μοντέλο που υλοποιήθηκε αποτελείται από 48 χαρακτηριστικά εισόδου και μία μεταβλητή στόχο. Επιπλέον διαθέτει δύο κρυμμένα στρώματα, ένα το οποίο διαθέτει πενήντα (50) κόμβους κι ένα στρώμα εξόδου που δίνει τέσσερις (4) μεταβλητές εξόδου, όσες και οι ζητούμενες τιμές πρόβλεψης, για τις οποίες οι ράμπες μένουν κλειστές (0,1,2,4).

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 50)	2450
dense_2 (Dense)	(None, 4)	204

=====
Total params: 2,654
Trainable params: 2,654
Non-trainable params: 0

Εικόνα 27. Στρώματα υλοποίησης νευρωνικού δικτύου

Κατά την παραμετροποίηση του μοντέλου, έγινε χρήση τυχαίας αρχικής κατανομής βαρών σε κάθε κόμβο, οι οποίες βελτιστοποιήθηκαν κατά την εκπαίδευση του μοντέλου. Η συνάρτηση Relu παρήγαγε τις συνθήκες τη λειτουργία ενεργοποίησης των κρυφών επιπέδων, ενώ παράλληλα έγινε και χρήση της συνάρτησης softmax, καθώς πρόκειται για πρόβλημα ταξινόμησης με πολλαπλές τιμές εξόδου.

Για τη βελτιστοποίηση του νευρωνικού δικτύου, χρησιμοποιήθηκε ο Adam, ο οποίος αντιπροσωπεύει την εκτίμηση της προσαρμοζόμενης ροπής. Ο Adam είναι ένας συνδυασμός του RMSProp και Momentum. Το Momentum λαμβάνει υπόψη τις διαβαθμίσεις του παρελθόντος για να εξομαλύνει την κλίση. Επιπλέον, χρησιμοποιήθηκε η ακρίβεια (accuracy) για τη μέτρηση της απόδοσης του μοντέλου όπως παρουσιάζεται σε επόμενο κεφάλαιο αναλυτικά με τα αποτελέσματα του νευρωνικού δικτύου.

Προκειμένου να επιβεβαιώσουμε την ορθότητα της απόδοσης του αλγορίθμου με την πάροδο του χρόνου καθώς εκπαιδεύεται με τη χρήση των δεδομένων, έγινε σχεδιασμός τόσο της ικανότητας στα δεδομένα εκπαίδευσης όσο και της ικανότητας σε ένα σύνολο δεδομένων που αποκλείστηκαν από τη διαδικασία κατάρτισης.

Με την πάροδο του χρόνου, καθώς μαθαίνει ο αλγόριθμος, το σφάλμα για το μοντέλο στα δεδομένα εκπαίδευσης μειώνεται και το σφάλμα στο σύνολο δεδομένων δοκιμής. Εάν η εκπαίδευση πραγματοποιούνταν για μεγαλύτερο χρονικό διάστημα, οι επιδόσεις στο σύνολο δεδομένων εκπαίδευσης ενδέχεται να μειώνονταν καθώς το μοντέλο θα ήταν υπερ-εκπαιδευμένο και η μάθηση των άσχετων λεπτομερειών και θορύβου στο σύνολο των δεδομένων εκπαίδευσης θα επηρέαζε αρνητικά την απόδοσή του. Ταυτόχρονα, το σφάλμα για το σετ δοκιμών θα αυξανόταν και πάλι, καθώς η ικανότητα του μοντέλου να γενικεύει τις

συνεχείς μειώσεις που λαμβάνουν χώρα σε κρίσιμες τιμές αυτού θα ελαττωνόταν. Κομβικό σημείο είναι αυτό ακριβώς πριν αρχίσει να αυξάνεται το σφάλμα στο σύνολο δεδομένων δοκιμής όπου το μοντέλο έχει καλές ικανότητες τόσο στο σύνολο δεδομένων κατάρτισης όσο και στο σύνολο δεδομένων μη δοκιμασμένων δεδομένων, σε προηγούμενο στάδιο.

Επομένως, με βάση όλα τα παραπάνω γίνεται άμεσα αντιληπτό ότι η ορθή υλοποίηση και η εκπαίδευση ενός νευρωνικού δικτύου εξαρτάται σε μεγάλο βαθμό από την ποιότητα των δεδομένων που του παρέχονται αλλά και από πληθώρα δοκιμών που πρέπει να εκτελεστούν προκειμένου να μην γίνει λανθασμένη εκπαίδευση είτε λόγω ελλিপών είτε λόγω υπερβολικών αναγνώσεων των δεδομένων. Επομένως, ο προσδιορισμός του κρίσιμου αριθμού αναγνώσεων αυτών που θα επιτύχουν την καλύτερη δυνατή απόδοση του νευρωνικού δικτύου αποτελεί μία διαδικασία με αυξημένη δυσκολία και χρονικό κόστος.

Στο επόμενο κεφάλαιο παρουσιάζονται αναλυτικά αποτελέσματα για την απόδοση του δικτύου που υλοποιήθηκε.

ΚΕΦΑΛΑΙΟ 4. ΑΠΟΤΕΛΕΣΜΑΤΑ -ΣΧΟΛΙΑΣΜΟΣ

ΜΕΘΟΔΩΝ ΥΛΟΠΟΙΗΣΗΣ

4.1. Γενικά

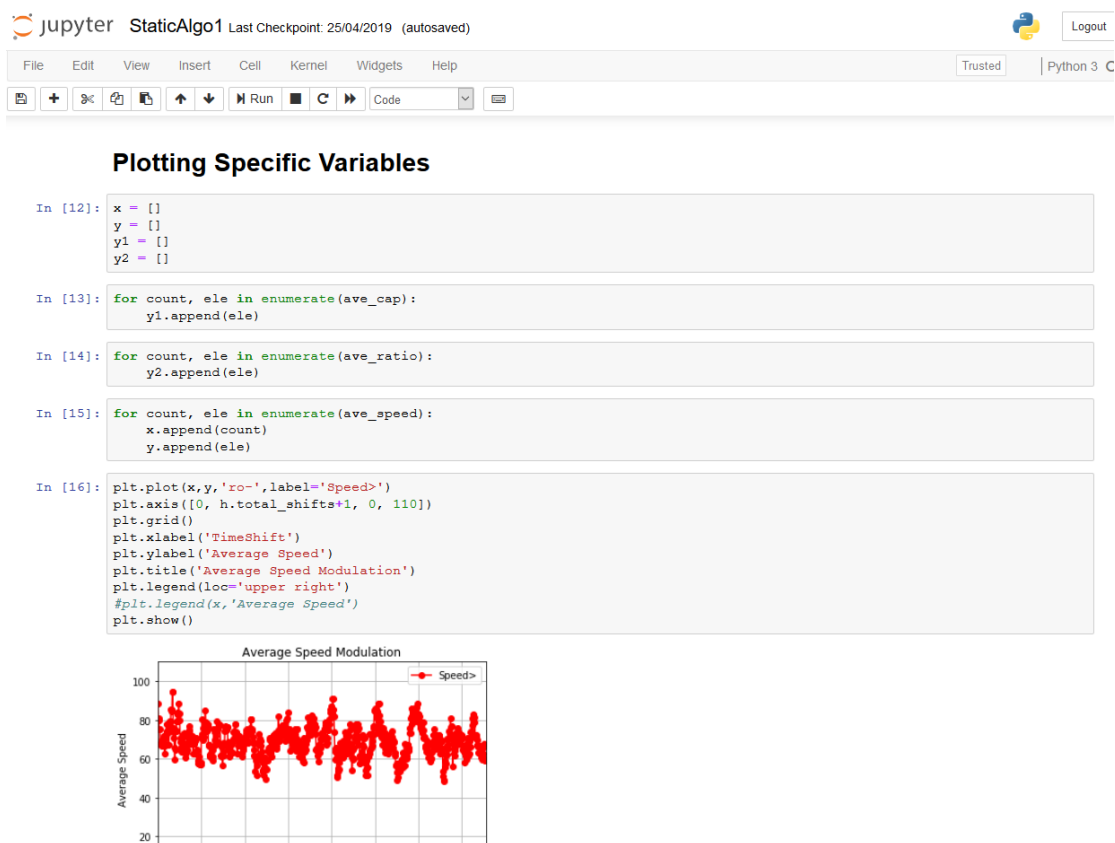
Στο παρόν κεφάλαιο θα παρουσιαστούν αναλυτικά αποτελέσματα με τη μορφή αριθμητικών δεδομένων και διαγραμμάτων για κάθε σενάριο – αλγόριθμο που υλοποιήθηκε και παρουσιάστηκε στο προηγούμενο κεφάλαιο. Η δομή του συγκεκριμένου κεφαλαίου θα ακολουθήσει τη δομή του προηγούμενου κεφαλαίου ανά αλγόριθμο - υλοποίηση έτσι ώστε να υπάρχει άμεση σύνδεση των δύο κεφαλαίων.

Η παραγωγή των αριθμητικών δεδομένων και των αποτελεσμάτων πραγματοποιήθηκε σε κάθε αλγόριθμο ξεχωριστά μέσω συγκεκριμένων εντολών και η εξαγωγή τους έγινε σε αναγνώσιμα αρχεία csv έτσι ώστε να μπορούν να μελετηθούν και να αναλυθούν ακόμα και με απλές εφαρμογές γραφείου. Επιπλέον, των αποτελεσμάτων που εξήχθησαν κάθε αλγόριθμος έχει τη δυνατότητα παράξει «ζωντανά» αποτελέσματα κατά την εκτέλεσή του. Το χαρακτηριστικό αυτό αποτελεί μία πολύ χρήσιμη δυνατότητα της πλατφόρμας Jupyter που χρησιμοποιήθηκε στην παρούσα εργασία για την εκτέλεση των αλγορίθμων και την αποσφαλμάτωση αυτών.

Το Jupyter Notebook είναι μια εφαρμογή web ανοιχτού κώδικα που επιτρέπει τη δημιουργία και των διαμοιρασμό εγγράφων που περιέχουν ζωντανό κώδικα, εξισώσεις, οπτικοποιήσεις και κείμενο. Μερικές από τις δυνατότητες που διαθέτει περιλαμβάνουν:

- τον καθορισμό και τον μετασχηματισμό δεδομένων
- την αριθμητική προσομοίωση
- τη στατιστική μοντελοποίηση
- την οπτικοποίηση δεδομένων
- την εκμάθηση μηχανών
- κ.α.

Το project Jupyter γενικά είναι μια μη κερδοσκοπική πλατφόρμα ανοιχτού κώδικα, η οποία «γεννήθηκε» από το πρόγραμμα IPython το 2014. Η δημιουργία του έχει ως στόχο να υποστηρίξει τη διαδραστική επιστήμη δεδομένων και την επιστημονική πληροφορική σε όλες τις γλώσσες προγραμματισμού.



Εικόνα 29. Στιγμιότυπο από Jupyter Notebook

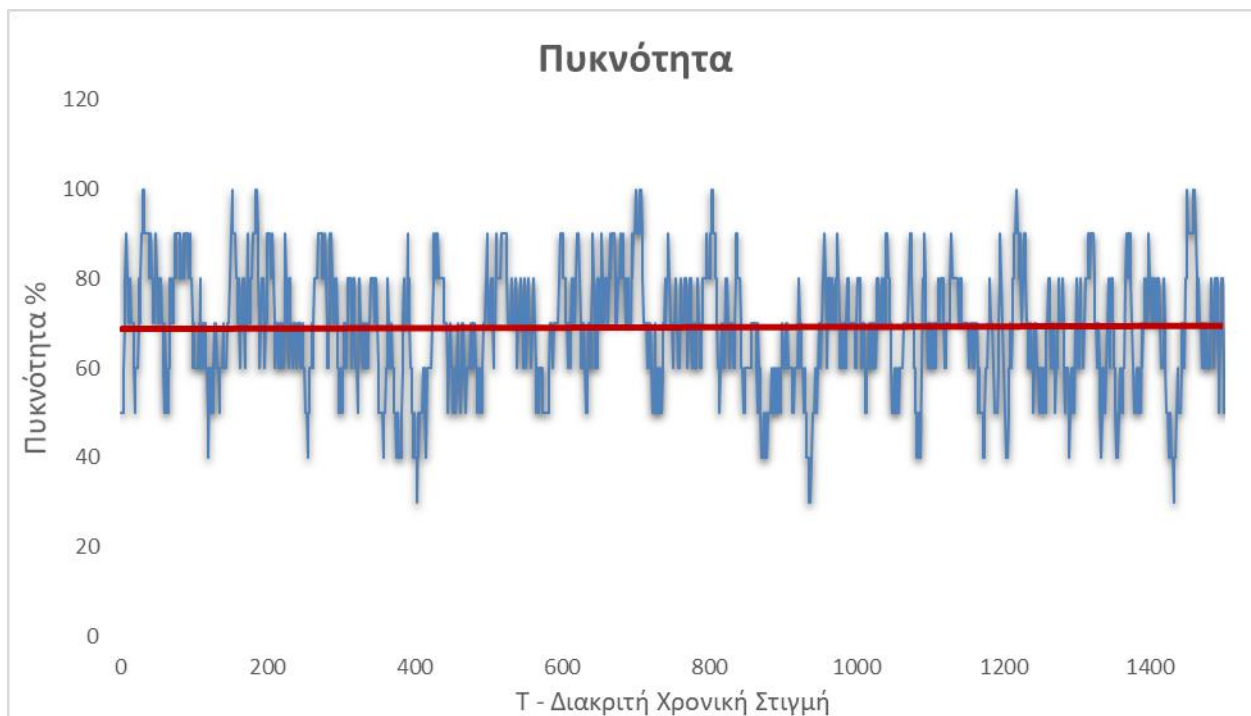
Τα αποτελέσματα που παρουσιάζονται στο παρόν κεφάλαιο παρήχθησαν από το Jupyter αρχικά και στη συνέχεια πραγματοποιήθηκε η ανάλυση και η γραφική απεικόνισή τους με χρήση άλλων εξωτερικών εργαλείων συμβατών με αρχεία csv.

4.2. Πιλοτικό Σενάριο Υλοποίησης – Αποτελέσματα

Η πιλοτική προσέγγιση ή ο αλγόριθμος πιλότος για την παρούσα εργασία έχει περιγραφεί αναλυτικά στο Κεφάλαιο 3 στην αντίστοιχη παράγραφο. Ο συγκεκριμένος αλγόριθμος εκτελέστηκε μέχρις ότου να περάσουν **1000** αυτοκίνητα συνολικά από τη ράμπα εισόδου. Για να περάσουν 1000 αυτοκίνητα από τη ράμπα ο αλγόριθμος εκτέλεσε **3057** επαναλήψεις μέχρι να επιτευχθεί η συνθήκη τερματισμού. Λόγω της απλότητας του συγκεκριμένου αλγορίθμου ο χρόνος εκτέλεσης ήταν αρκετά μικρός, δηλαδή **2,13** δευτερόλεπτα.

Παρακάτω παρουσιάζονται κάποια βασικά διαγράμματα και αριθμητικά αποτελέσματα για την πυκνότητα, τη μέση ταχύτητα και τον λόγο R όπως αυτά έχουν τεκμηριωθεί μαθηματικά στο Κεφάλαιο 2.

4.2.1. Πυκνότητα (Density)

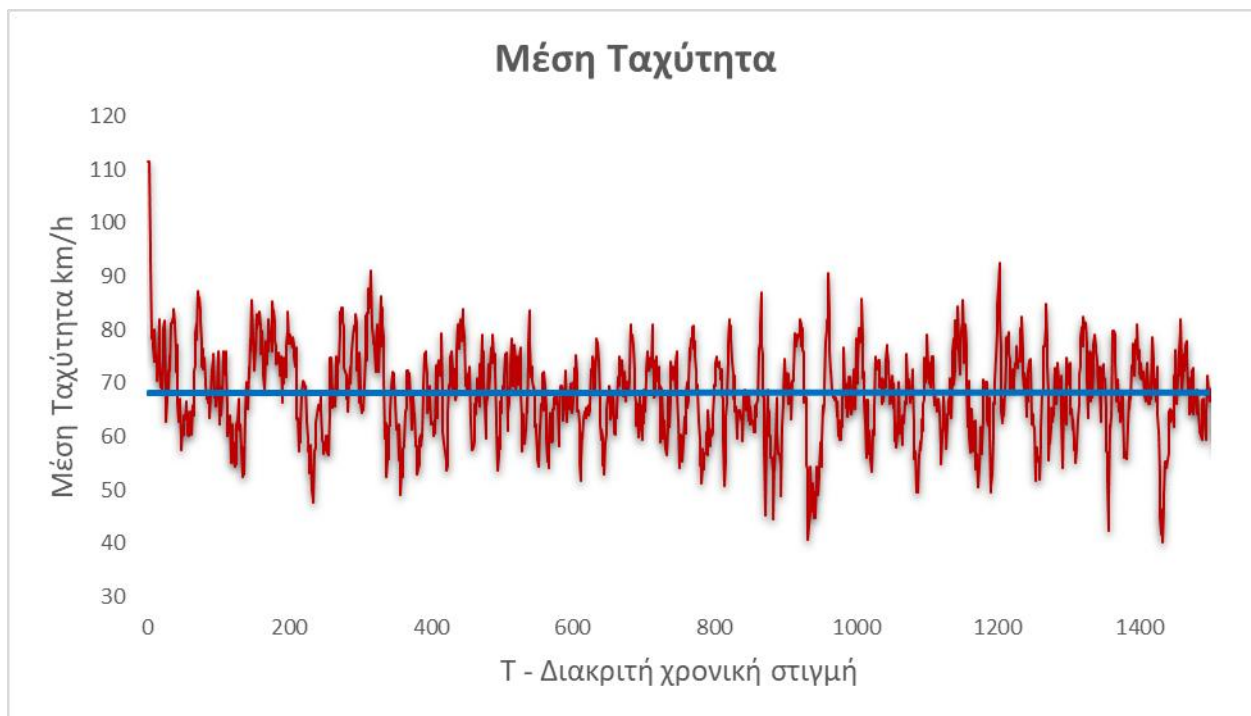


Διάγραμμα 1. Πυκνότητα πιλοτικού αλγορίθμου

Στο παραπάνω διάγραμμα απεικονίζεται η πυκνότητα που υπάρχει στο κύριο ρεύμα του αυτοκινητόδρομου για τις πρώτες 1500 χρονικές στιγμές (επαναλήψεις). Παρατηρούμε ότι η πυκνότητα παρουσιάζει διακυμάνσεις καθ' όλη τη διάρκεια εκτέλεσης του αλγορίθμου και μία σταθερότητα στην τάση όπως φαίνεται από την κόκκινη γραμμή τάσης (trendline). Οι έντονες αυτές διακυμάνσεις είναι απόλυτα λογικές καθώς ο αλγόριθμος προσπαθεί να συγκρατεί την πυκνότητα του κύριου ρεύματος σε κάποιο εύρος γύρω από το 70%.

Με βάση το παραπάνω σύνολο μετρήσεων η μέση υπολογιζόμενη πυκνότητα είναι **69,51%** ποσοστό αρκετά ικανοποιητικό αναλογιζόμενη την απλότητα της συγκεκριμένης μεθόδου.

4.2.2. Μέση ταχύτητα (Mean speed)

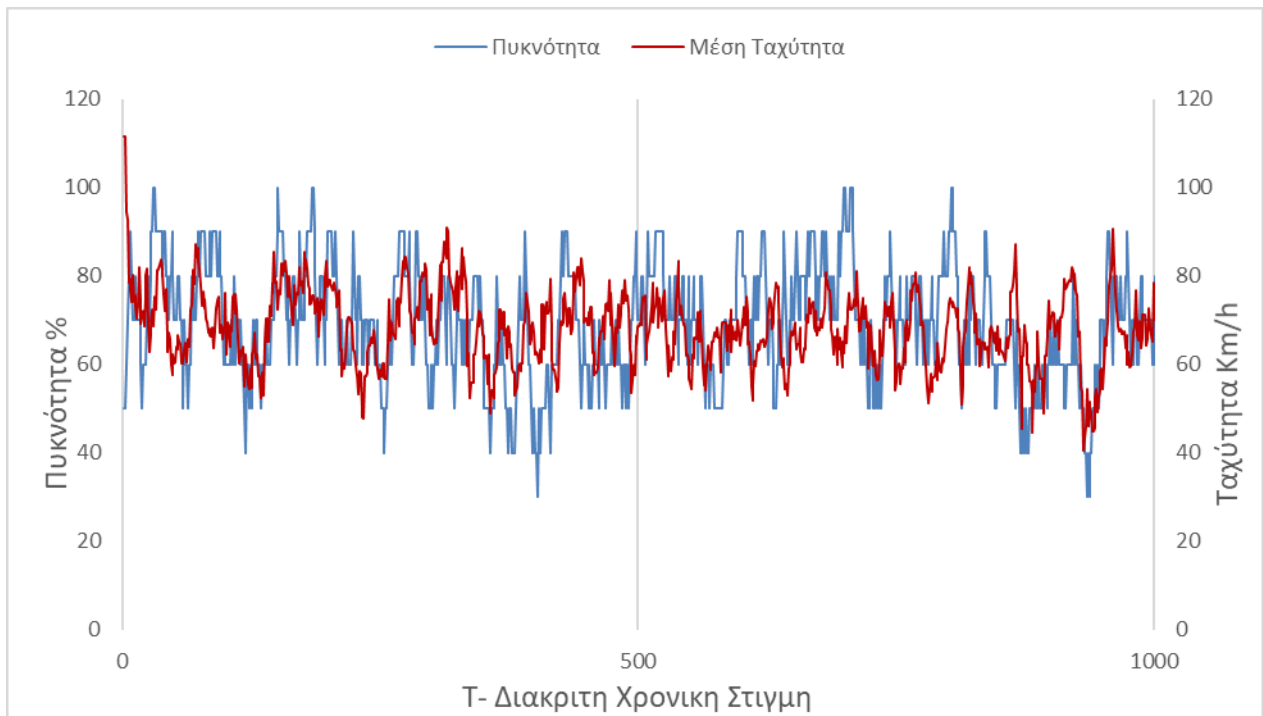


Διάγραμμα 2. Μέση ταχύτητα πιλοτικού αλγορίθμου

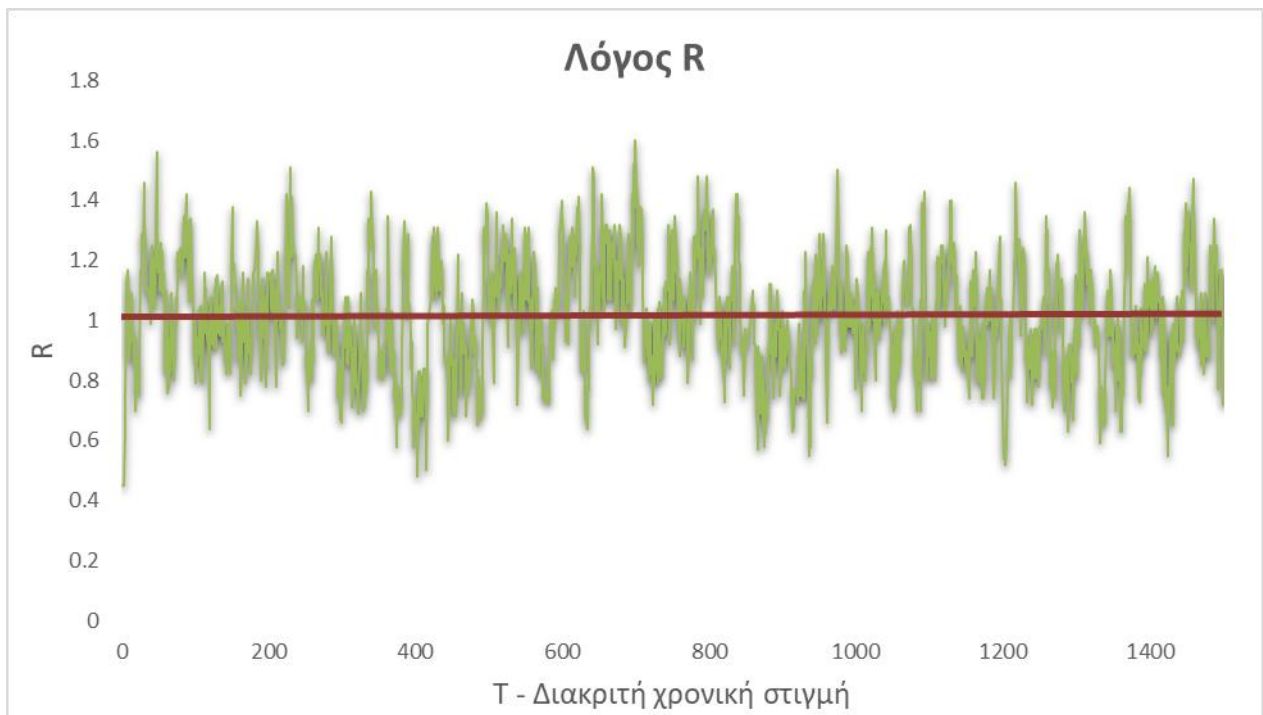
Ομοίως με την πυκνότητα στο παραπάνω διάγραμμα απεικονίζεται η μέση ταχύτητα των οχημάτων που κινούνται στο κύριο ρεύμα του αυτοκινητόδρομου για τις πρώτες 1500 χρονικές στιγμές (επαναλήψεις). Παρατηρούμε ότι η μέση ταχύτητα όπως και η πυκνότητα παρουσιάζει διακυμάνσεις καθ' όλη τη διάρκεια εκτέλεσης του αλγορίθμου. Ωστόσο έχει μία σταθερότητα στη διάθεσή της όπως φαίνεται από τη μπλε γραμμή τάσης (trendline). Οι έντονες ωστόσο αυτές διακυμάνσεις οφείλονται στις συνεχείς αυξομειώσεις της πυκνότητας που φαίνεται παραπάνω.

Με βάση το παραπάνω σύνολο μετρήσεων ο μέσος όρος των μέσων ταχυτήτων που έχουν υπολογιστεί για κάθε χρονική στιγμή είναι **68,33 km/h**. Το εύρος ταχυτήτων που έχει οριστεί για κάθε όχημα που κινείται εντός του αυτοκινητόδρομου είναι 70 έως 130 km/h.

4.2.3. Πυκνότητα και μέση ταχύτητα



Διάγραμμα 3. Πυκνότητα και μέση ταχύτητα πιλοτικού αλγορίθμου



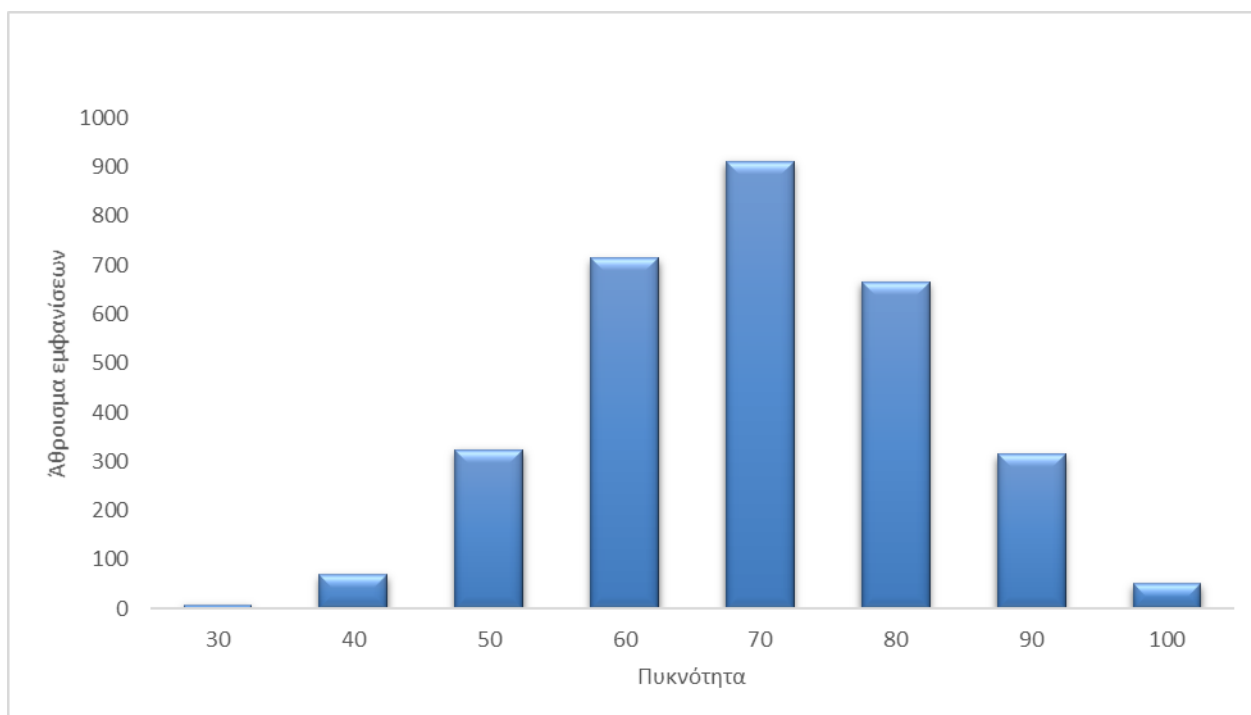
Διάγραμμα 4. Λόγος R πιλοτικού αλγορίθμου

Παραπάνω παρουσιάζονται το συνδυαστικό διάγραμμα μέσης ταχύτητας και πυκνότητας καθώς και το διάγραμμα για τον λόγο R όπως αυτός έχει οριστεί στο Κεφάλαιο 2, δηλαδή πυκνότητα προς ταχύτητα. Παρατηρούμε ότι το διάγραμμα της μέσης ταχύτητας ακολουθεί αντίστροφα με μία μικρή υστέρηση αυτό της πυκνότητας. Αυτό είναι αναμενόμενο

καθώς όσο αυξάνεται η πυκνότητα τόσο αρχίζει και μειώνεται η μέση ταχύτητα. Επιπλέον σε σημεία που πυκνότητα αυξάνεται σε μεγάλο βαθμό η μέση ταχύτητα μειώνεται δραστικά. Παρατηρούμε ότι η μέγιστη μέση ταχύτητα είναι **111,14 km/h** και η ελάχιστη ταχύτητα **33 km/h**. Όπως γίνεται αντιληπτό υπάρχει μία μεγάλη διαφορά ανάμεσα σε αυτά τα δύο (2) νούμερα γεγονός απόλυτα λογικό καθώς ο αλγόριθμος είναι αρκετά απλοϊκός και εκτελεί απότομες μεταβολές.

Το διάγραμμα που αφορά τον λόγο R υποδεικνύει σταθερότητα στην τάση του όπως προκύπτει και από το trendline χρώματος καφέ. Η σταθερότητα αυτή οφείλεται στη σταθερότητα της τάσης της πυκνότητας (αριθμητής) και στη σταθερότητα της τάσης της μέσης ταχύτητας (παρονομαστής). Επομένως παρατηρούμε ότι με τον λόγο R έχουμε μία επαλήθευση του συνόλου των πληροφοριών που παρέχει κάθε αλγόριθμος και αποτελεί ένα χρήσιμο μέγεθος για τη μελέτη των αποτελεσμάτων καθώς προσδίδει μία μορφή επιβεβαίωσης (evaluation) και κατανόησης της συσχέτισης των δύο αυτών κρίσιμων μεγεθών.

Τέλος, το παρακάτω ιστόγραμμα παρουσιάζεται το άθροισμα των εμφανίσεων κάθε τιμής πυκνότητας από το σύνολο (30,40,50,60,70,80,90,100).



Διάγραμμα 5. Ιστόγραμμα πυκνοτήτων πιλοτικού αλγορίθμου

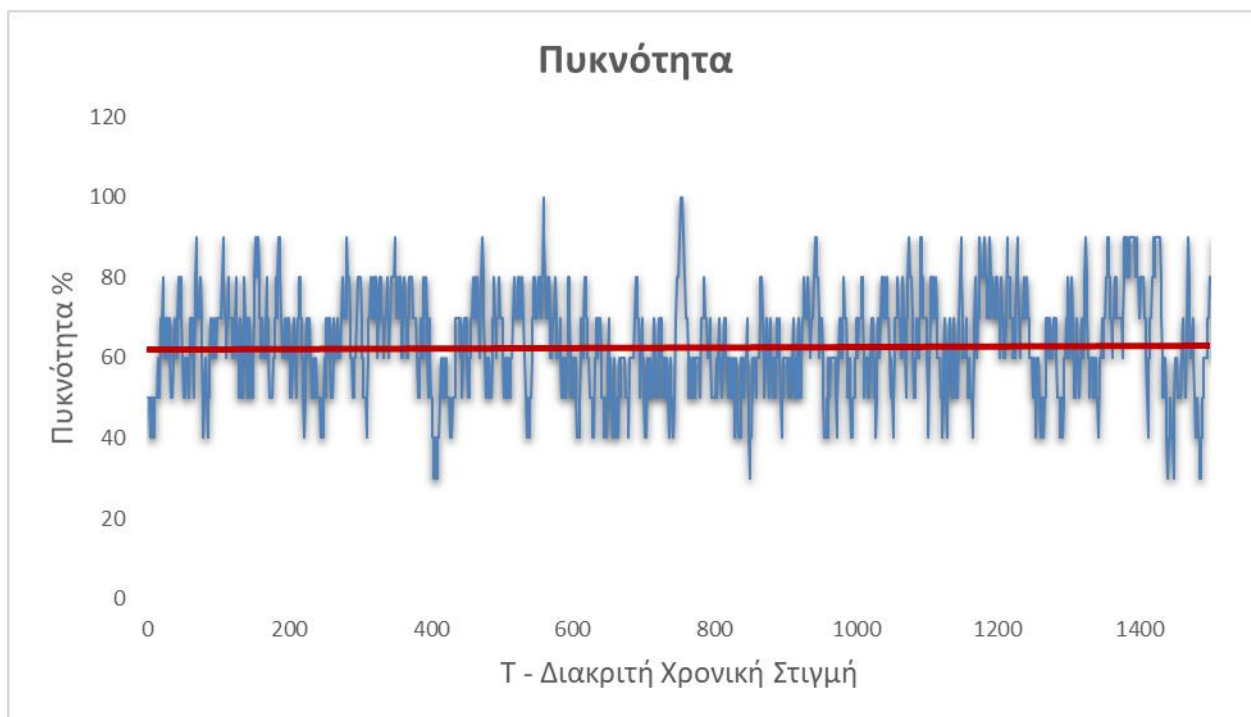
Στο παραπάνω ιστόγραμμα παρατηρούμε ότι ο αλγόριθμος αν και αρκετά απλοϊκός διατηρεί στο μεγαλύτερο διάστημά του την πυκνότητα στο διάστημα 60-80% αποτέλεσμα αρκετά ικανοποιητικό καθώς βρίσκεται γύρω από την οριακή τιμή του 70%.

4.3. 1^ο Σενάριο Υλοποίησης – 1 Λωρίδα & 1 ράμπα

Το 1^ο σενάριο υλοποίησης με μία (1) ράμπα και μία (1) λωρίδα για την παρούσα εργασία έχει περιγραφεί αναλυτικά στο Κεφάλαιο 3 στην αντίστοιχη παράγραφο. Ο συγκεκριμένος αλγόριθμος, ομοίως με τον πιλοτικό, εκτελέστηκε μέχρις ότου να περάσουν **1000** αυτοκίνητα συνολικά από τη ράμπα εισόδου. Για να περάσουν 1000 αυτοκίνητα από τη ράμπα ο αλγόριθμος εκτέλεσε **4383** επαναλήψεις μέχρι να επιτευχθεί η συνθήκη τερματισμού. Παρατηρούμε ότι σε σχέση με τον προηγούμενο αλγόριθμο ο αριθμός των επαναλήψεων αυξήθηκε σε μεγάλο βαθμό και αντίστοιχα αυξήθηκε και ο χρόνος περάτωσης του αλγόριθμου σε **4,08** δευτερόλεπτα..

Παρακάτω ομοίως με πριν παρουσιάζονται κάποια βασικά διαγράμματα και αριθμητικά αποτελέσματα για την πυκνότητα, τη μέση ταχύτητα και τον λόγο R όπως αυτά έχουν τεκμηριωθεί μαθηματικά στο Κεφάλαιο 2.

4.3.1. Πυκνότητα (Density)

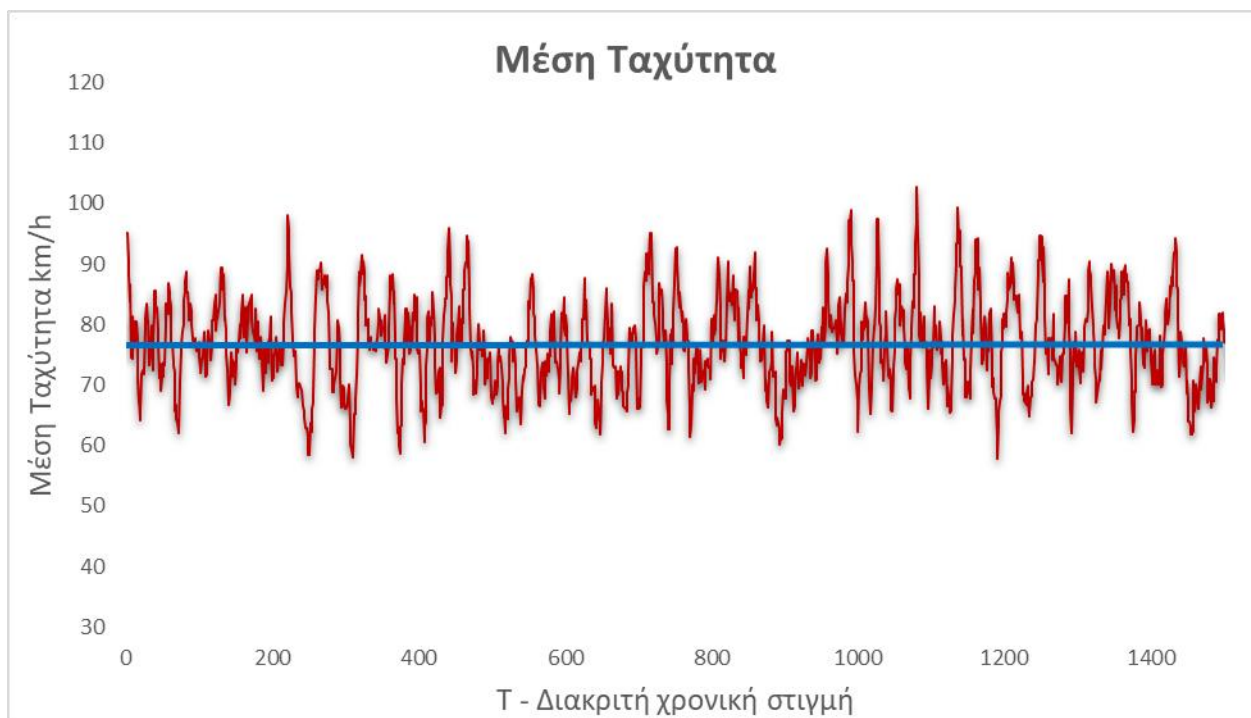


Διάγραμμα 6. Πυκνότητα αλγόριθμου 1^ο σεναρίου

Στο παραπάνω διάγραμμα απεικονίζεται η πυκνότητα που υπάρχει στο κύριο ρεύμα του αυτοκινητόδρομου για τις πρώτες 1500 χρονικές στιγμές (επαναλήψεις). Παρατηρούμε ότι η πυκνότητα παρουσιάζει μικρότερες διακυμάνσεις σε σχέση με τον προηγούμενο αλγόριθμο και μία σταθερότητα ως προς τη διάθεσή της όπως φαίνεται από την κόκκινη γραμμή τάσης (trendline). Αυτό οφείλεται στις αυξημένες συνθήκες ελέγχου που ενσωματώθηκαν στον συγκεκριμένο αλγόριθμο σε σχέση με πριν. Το γεγονός αυτό έχει ως αποτέλεσμα με την πάροδο του χρόνου ο αλγόριθμος να διατηρεί εντός φυσιολογικών πλαισίων την πυκνότητα γύρω από ένα σταθερό μέσο όρο.

Με βάση το παραπάνω σύνολο μετρήσεων η μέση υπολογιζόμενη πυκνότητα είναι **63,45%** ποσοστό χαμηλότερο σε σχέση με πριν γεγονός που οφείλεται στις αυστηρότερες συνθήκες ελέγχου που εκτελεί ο συγκεκριμένος αλγόριθμος.

4.3.2. Μέση ταχύτητα (Mean speed)

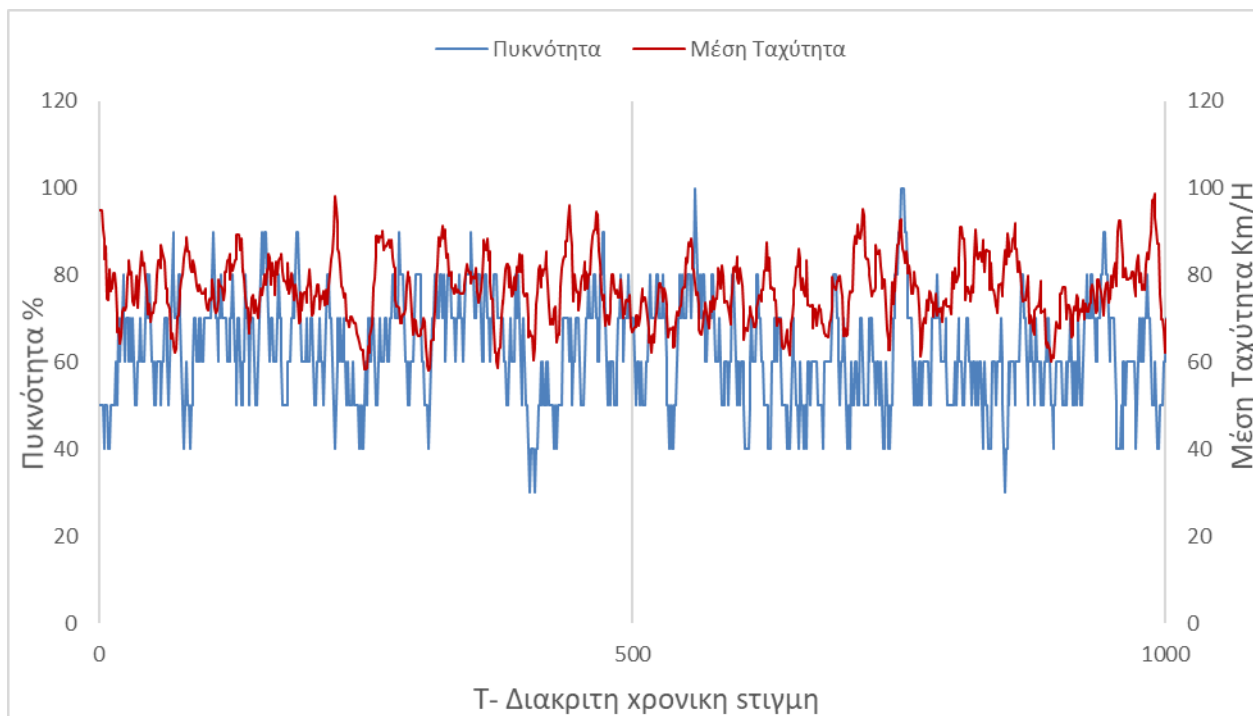


Διάγραμμα 7. Μέση ταχύτητα αλγορίθμου 1^ο σεναρίου

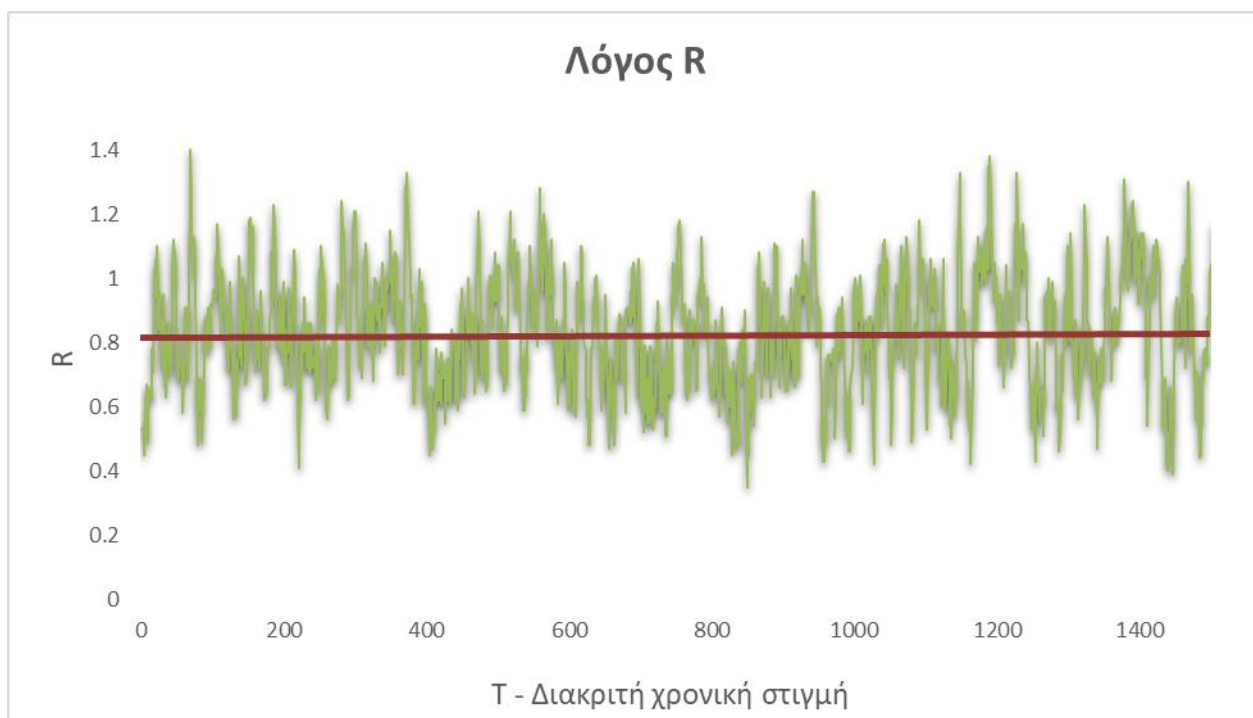
Ομοίως με την πυκνότητα στο παραπάνω διάγραμμα απεικονίζεται η μέση ταχύτητα των οχημάτων που κινούνται στο κύριο ρεύμα του αυτοκινητόδρομου. Παρατηρούμε ότι η μέση ταχύτητα όπως και η πυκνότητα παρουσιάζει διακυμάνσεις καθ' όλη τη διάρκεια εκτέλεσης του αλγορίθμου. Ωστόσο όπως και στον προηγούμενο αλγόριθμο έχει μία σταθερότητα στη διάθεσή της όπως φαίνεται από μπλε γραμμή τάσης (trendline).

Με βάση το παραπάνω σύνολο μετρήσεων ο μέσος όρος που προκύπτει από τις μέσες ταχύτητες ανά χρονική στιγμή είναι **76,71 km/h**. Παρατηρούμε ότι με την αύξηση της αυστηρότητας των συνθηκών ελέγχου η μέση πυκνότητα μειώθηκε αρκετά και παράλληλα ο μέσος όρος των μέσων ταχυτήτων αυξήθηκε σχεδόν 10 km/h. Επομένως, γίνεται αντιληπτό ότι ο συγκεκριμένος αλγόριθμος είναι καλύτερος σε σχέση με πριν καθώς επιτρέπει στα οχήματα να κινούνται με μεγαλύτερες ταχύτητες διατηρώντας την πυκνότητα σε χαμηλότερα επίπεδα. Το εύρος ταχυτήτων που έχει οριστεί για κάθε όχημα που κινείται εντός του αυτοκινητόδρομου είναι 70 έως 130 km/h.

4.3.3. Πυκνότητα και μέση ταχύτητα



Διάγραμμα 8. Πυκνότητα και μέση ταχύτητα αλγόριθμου 1^{ου} σεναρίου



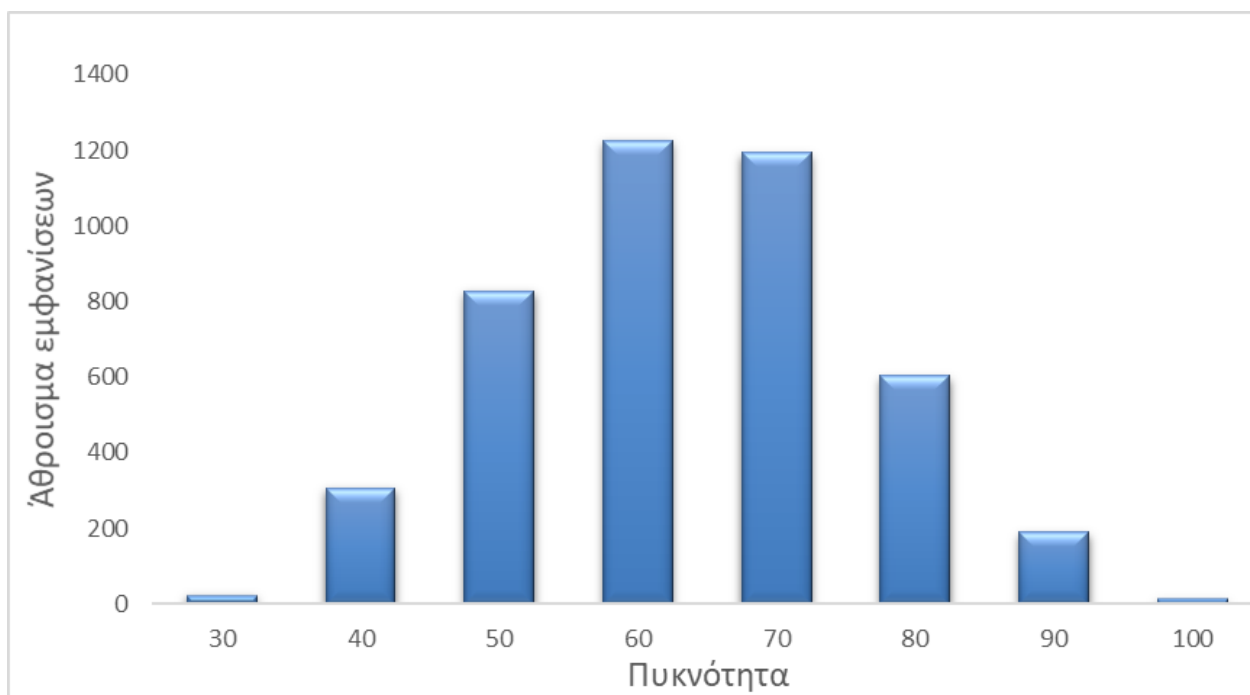
Διάγραμμα 9. Λόγος R αλγόριθμου 1^{ου} σεναρίου

Παραπάνω παρουσιάζονται το συνδυαστικό διάγραμμα μέσης ταχύτητας και πυκνότητας καθώς και το διάγραμμα για τον λόγο R όπως αυτός έχει οριστεί στο Κεφάλαιο 2, δηλαδή πυκνότητα προς ταχύτητα. Παρατηρούμε ότι σε αυτόν τον αλγόριθμο το διάγραμμα της μέσης ταχύτητας συμπεριφέρεται αντίστροφα από αυτό της πυκνότητας, όπως είναι

φυσιολογικό με μία μικρή υστέρηση πάντα η οποία οφείλεται στις ταχείες μεταβολές της πυκνότητας. Δηλαδή σε σημεία που πυκνότητα αυξάνεται σε μεγάλο βαθμό η μέση ταχύτητα μειώνεται δραστικά και αντίστροφα. Παρατηρούμε ότι η μέγιστη μέση ταχύτητα είναι **102,6 km/h** και η ελάχιστη ταχύτητα **57,88 km/h**. Η διαφορά της μέγιστης μέσης ταχύτητας από την ελάχιστη έχει μειωθεί δραστικά σε αυτό το σενάριο ελέγχου κάτι το οποίο υποδηλώνει τον καλύτερο έλεγχο που πραγματοποιείται μέσω του συγκεκριμένου αλγόριθμου και άρα την καλύτερη απόδοση.

Το διάγραμμα που αφορά τον λόγο R όπως και πριν παρουσιάζει σταθερότητα στην τάση του όπως προκύπτει και από το trendline χρώματος καφέ. Η σταθερότητα αυτή οφείλεται στη σταθερότητα της τάσης της πυκνότητας (αριθμητής) και στη σταθερότητα της τάσης της μέσης ταχύτητας (παρονομαστής). Επομένως ομοίως με πριν παρατηρούμε ότι με τον λόγο R έχουμε μία επαλήθευση του συνόλου των πληροφοριών που παρέχει ο αλγόριθμος επιβεβαιώνοντας τη συσχέτιση των δύο αυτών κρίσιμων μεγεθών.

Τέλος, το παρακάτω ιστόγραμμα παρουσιάζεται το άθροισμα των εμφανίσεων κάθε τιμής πυκνότητας από το σύνολο (30,40,50,60,70,80,90,100).



Διάγραμμα 10. Ιστόγραμμα πυκνοτήτων αλγόριθμου 1^ο σεναρίου

Στο παραπάνω ιστόγραμμα παρατηρούμε ότι και ο αλγόριθμος ομοίως με πριν διατηρεί στο μεγαλύτερο διάστημά του την πυκνότητα στο διάστημα 50-80% και ειδικότερα μεταξύ του 60-70% αποτέλεσμα αρκετά ικανοποιητικό καθώς βρίσκεται γύρω από την οριακή τιμή του 70%.

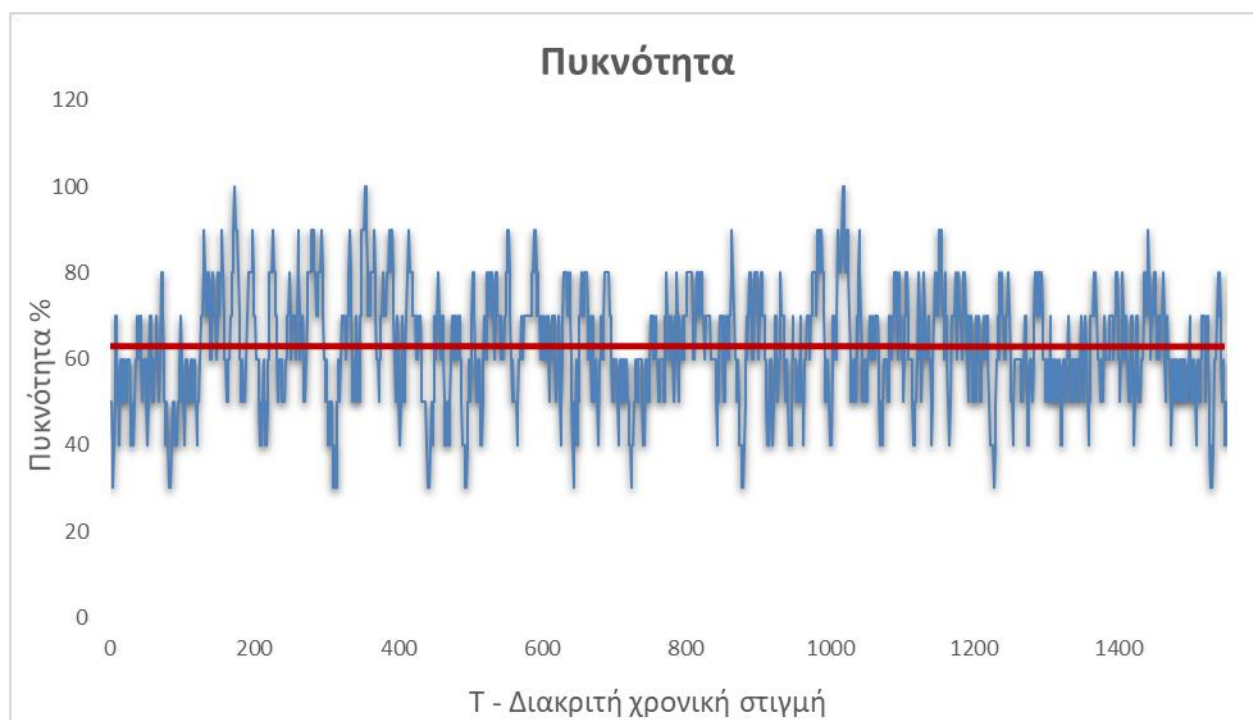
Παρατηρούμε ότι ο συγκεκριμένος αλγόριθμος είχε μικρότερες διακυμάνσεις σε σχέση με τον αρχικό ενώ διατηρεί τη μέση πυκνότητα σε ελαφρώς χαμηλότερο ποσοστό. Αυτή η μικρή ωστόσο μεταβολή στην πυκνότητα έχει σημαντικό αντίκτυπο στην ταχύτητα καθώς βλέπουμε ότι αυξάνεται κατά περίπου 10 km/h.

4.4. 2^ο Σενάριο Υλοποίησης – 1 Λωρίδα & 1 ράμπα

Το 2^ο σενάριο υλοποίησης με μία (1) ράμπα και μία (1) λωρίδα είναι αρκετά όμοιο με το 1^ο καθώς η μοναδική διαφορά είναι ότι πλέον η βοηθητική λωρίδα εισόδου έχει μεγαλύτερη χωρητικότητα. Το συγκεκριμένο σενάριο έχει περιγραφεί αναλυτικά στο Κεφάλαιο 3 στην αντίστοιχη παράγραφο. Ο συγκεκριμένος αλγόριθμος, ομοίως με τους δύο προηγούμενους, εκτελέστηκε μέχρις ότου να περάσουν **1000** αυτοκίνητα συνολικά από τη ράμπα εισόδου. Για να περάσουν 1000 αυτοκίνητα από τη ράμπα ο αλγόριθμος εκτέλεσε **4436** επαναλήψεις μέχρι να επιτευχθεί η συνθήκη τερματισμού. Παρατηρούμε μία πολύ μικρή αύξηση σε σχέση με τον προηγούμενο αλγόριθμο στον αριθμό των επαναλήψεων και αντίστοιχα και στον χρόνο εκτέλεσης **4,13** δευτερόλεπτα.

Παρακάτω παρουσιάζονται κάποια βασικά διαγράμματα και αριθμητικά αποτελέσματα για την πυκνότητα, τη μέση ταχύτητα και τον λόγο R όπως αυτά έχουν τεκμηριωθεί μαθηματικά στο Κεφάλαιο 2.

4.4.1. Πυκνότητα (Density)

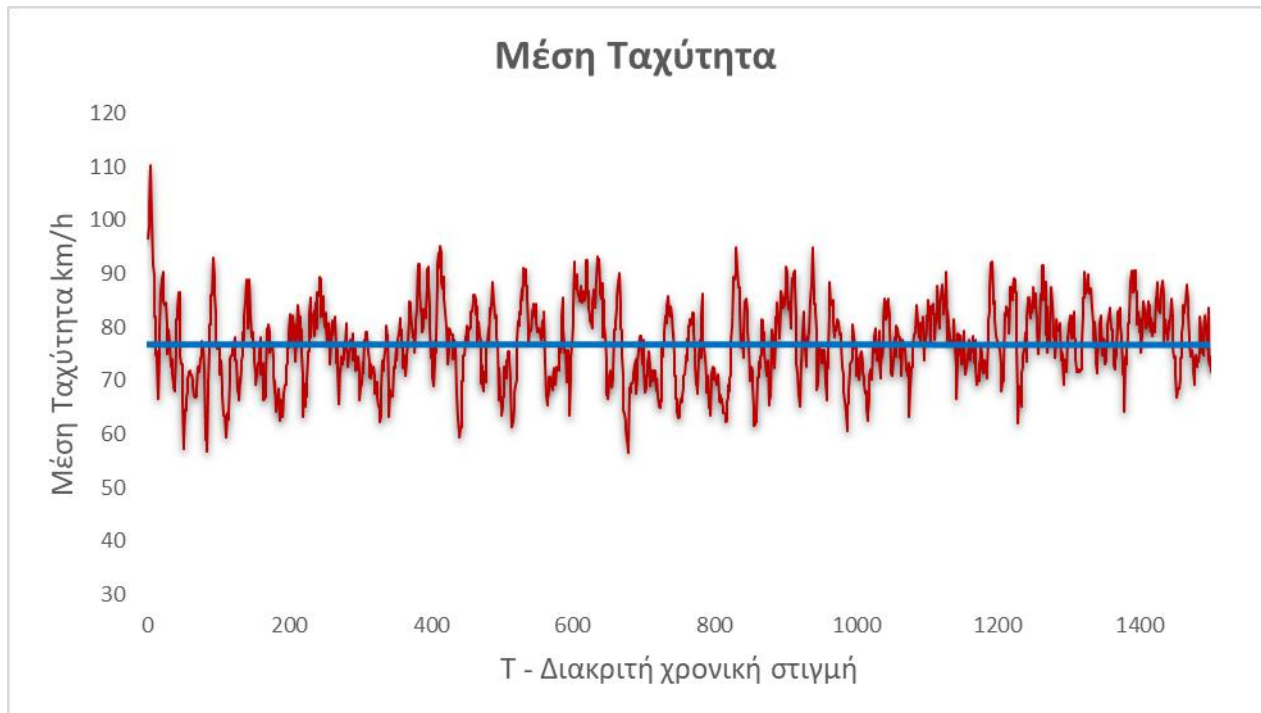


Διάγραμμα 11. Πυκνότητα 2^{ου} σεναρίου υλοποίησης

Στο παραπάνω διάγραμμα απεικονίζεται η πυκνότητα που υπάρχει στο κύριο ρεύμα του αυτοκινητόδρομου για τις πρώτες 1500 επαναλήψεις. Παρατηρούμε ότι η πυκνότητα παρουσιάζει παρόμοιες διακυμάνσεις σε σχέση με τον προηγούμενο αλγόριθμο και μία σταθερότητα ως προς τη διάθεσή της όπως φαίνεται από την κόκκινη γραμμή τάσης (trendline). Αυτό οφείλεται στο γεγονός ότι οι συνθήκες ελέγχου που ενσωματώθηκαν στον συγκεκριμένο αλγόριθμο είναι πανομοιότυπες με τον προηγούμενο. Έτσι τα αποτελέσματα όπως είναι και αναμενόμενο είναι αρκετά κοντά ανάμεσα στο 1^ο και στο 2^ο σενάριο υλοποίησης.

Με βάση το παραπάνω σύνολο μετρήσεων η μέση υπολογιζόμενη πυκνότητα είναι **63,02%** ποσοστό ελαφρώς χαμηλότερο σε σχέση με πριν γεγονός που οφείλεται στην αυξημένη χωρητικότητα της βοηθητικής λωρίδας. Ωστόσο, επειδή η διαφορά είναι αρκετά μικρή κάτω από το 1% θεωρούμε ότι είναι αμελητέα.

4.4.2. Μέση ταχύτητα (Mean speed)

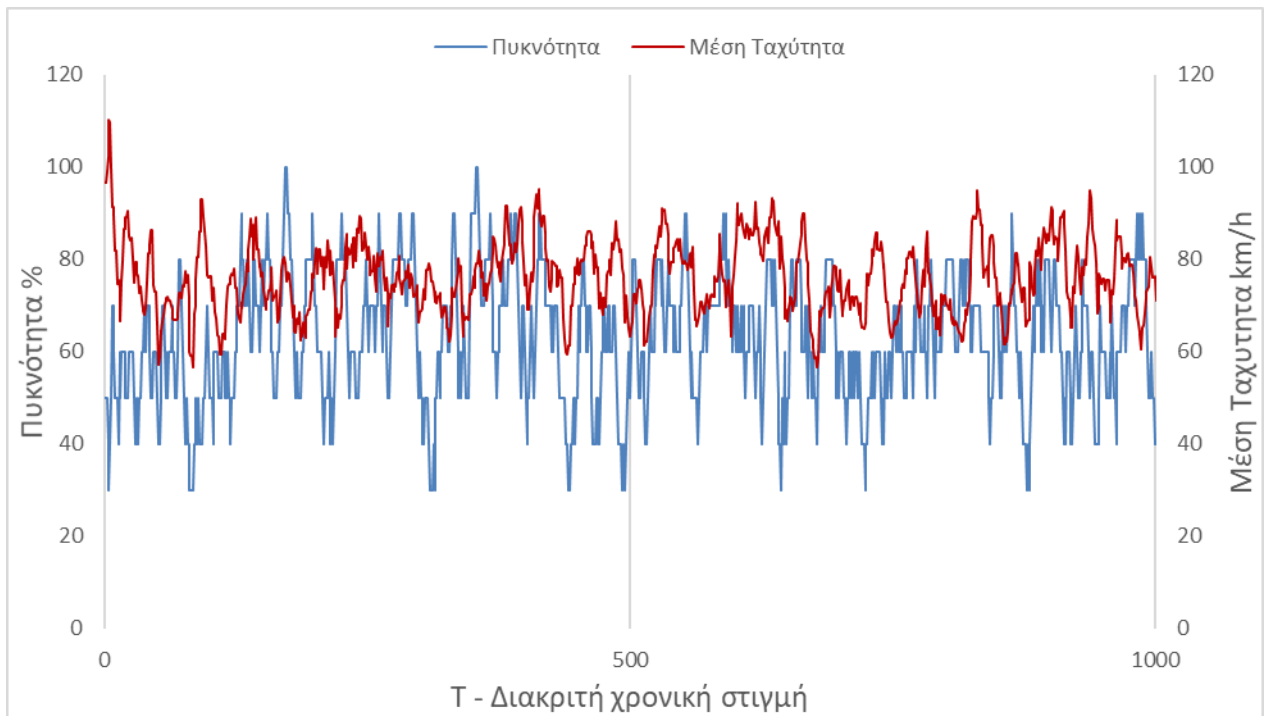


Διάγραμμα 12. Μέση ταχύτητα αλγόριθμου 2^{ου} σεναρίου υλοποίησης

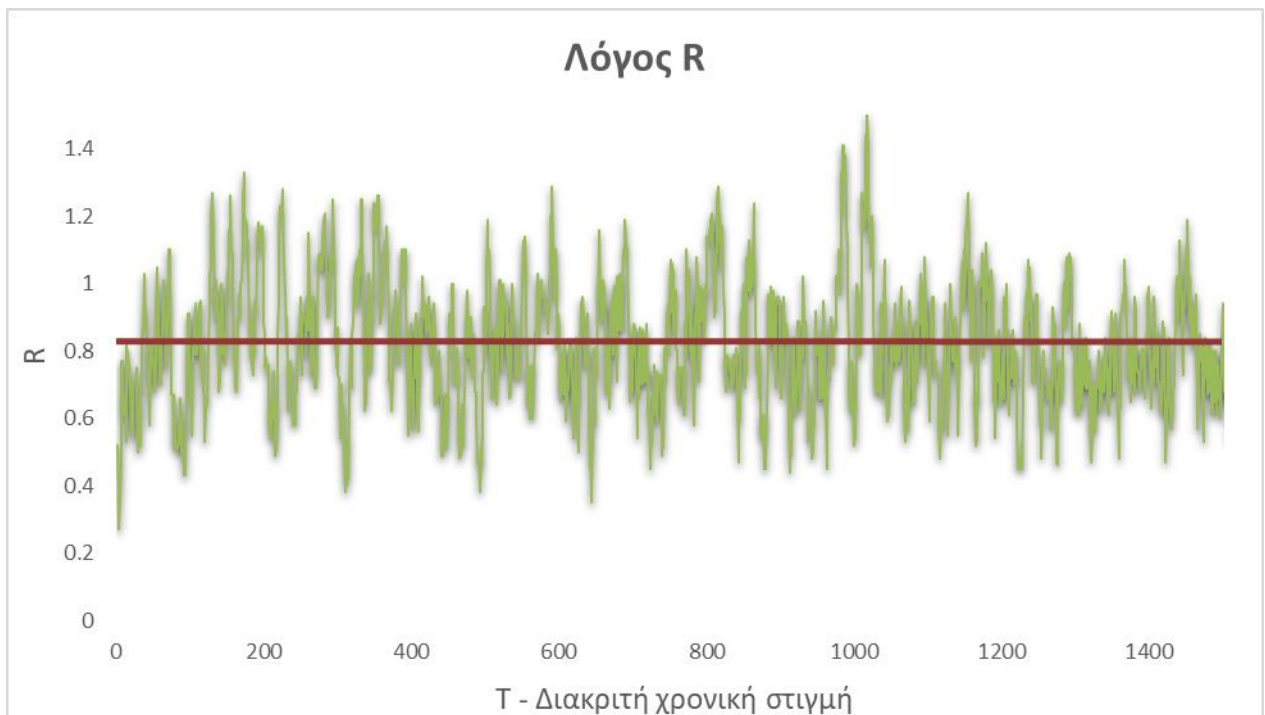
Ομοίως και με τον προηγούμενο αλγόριθμο στο παραπάνω διάγραμμα απεικονίζεται η μέση ταχύτητα των οχημάτων που κινούνται στο κύριο ρεύμα του αυτοκινητόδρομου για το 2^ο σενάριο υλοποίησης. Παρατηρούμε ότι η μέση ταχύτητα όπως και η πυκνότητα παρουσιάζει διακυμάνσεις καθ' όλη τη διάρκεια εκτέλεσης του αλγόριθμου. Ωστόσο όπως και στους δύο προηγούμενους αλγόριθμους έχει μία σταθερότητα στη διάθεσή της όπως φαίνεται από μπλε γραμμή τάσης (trendline). Το αποτέλεσμα αυτό είναι αναμενόμενο λόγω της ομοιότητας των δύο υλοποιήσεων.

Με βάση το παραπάνω σύνολο μετρήσεων η μέση υπολογιζόμενη ταχύτητα που προκύπτει από τις μέσες ταχύτητες ανά χρονική στιγμή είναι **76,73 km/h**. Παρατηρούμε λοιπόν ότι η ο μέσος όρος των μέσων ταχυτήτων αυξήθηκε ελάχιστα. Η συγκεκριμένη αύξηση θεωρείται αμελητέα καθώς οι δύο αλγόριθμοι παρουσιάζουν μεγάλη ομοιότητα. Ωστόσο, ακόμα και αυτή η μικρή αύξηση καθιστά τον συγκεκριμένο αλγόριθμο αποδοτικότερο σε σχέση με τους δύο προηγούμενους. Το εύρος ταχυτήτων που έχει οριστεί για κάθε όχημα που κινείται εντός του αυτοκινητόδρομου είναι 70 έως 130 km/h.

4.4.3. Πυκνότητα και μέση ταχύτητα



Διάγραμμα 13. Πυκνότητα και μέση ταχύτητα αλγόριθμου 2^{ου} σεναρίου



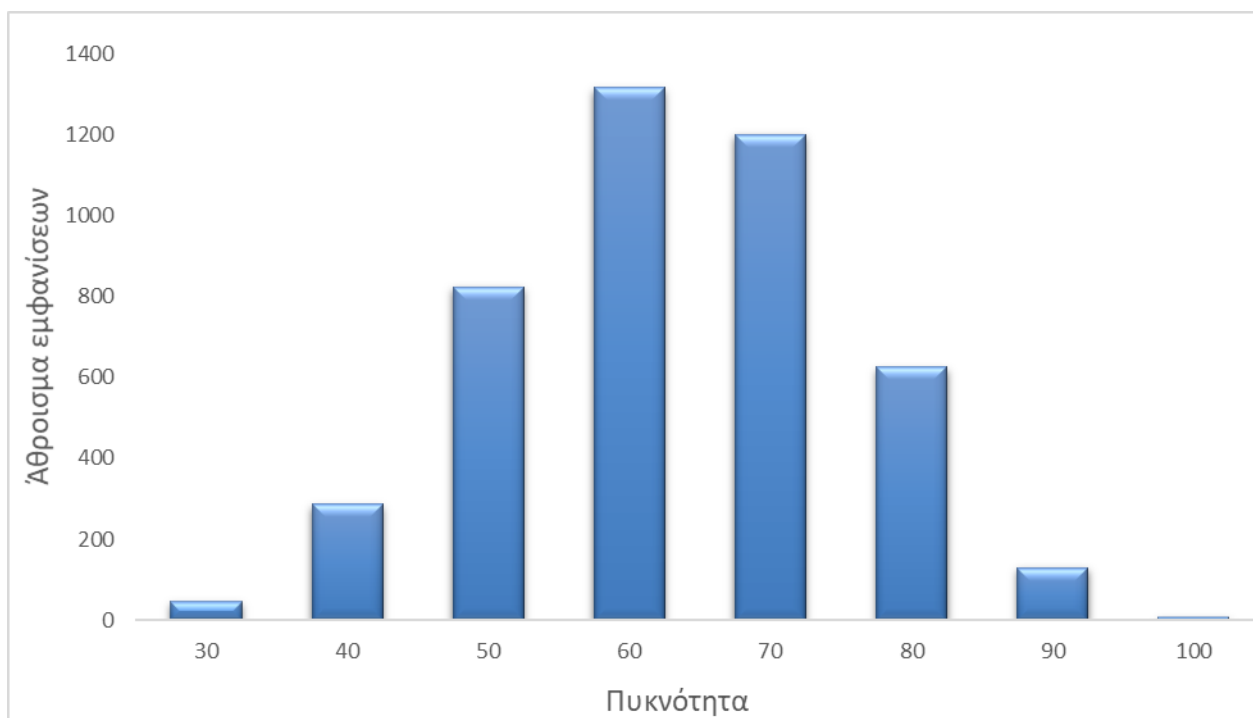
Διάγραμμα 14. Λόγος R αλγόριθμου 2^{ου} σεναρίου

Όπως και προηγουμένως παραπάνω παρατίθενται το συνδυαστικό διάγραμμα μέσης ταχύτητας και πυκνότητας καθώς και το διάγραμμα για τον λόγο R όπως αυτός έχει οριστεί στο Κεφάλαιο 2, δηλαδή πυκνότητα προς ταχύτητα. Παρατηρούμε ότι ομοίως με το 1^ο σενάριο υλοποίησης το διάγραμμα της μέσης ταχύτητας συμπεριφέρεται αντίστροφα από αυτό της

πυκνότητας, όπως είναι φυσιολογικό με μία μικρή υστέρηση πάντα η οποία οφείλεται στις ταχείες μεταβολές της πυκνότητας και της ταχύτητας. Έτσι όταν η πυκνότητα αυξάνεται η μέση ταχύτητα αντίστοιχα μειώνεται. Παρατηρούμε ότι η μέγιστη μέση ταχύτητα είναι **110,33 km/h** και η ελάχιστη ταχύτητα **56,50 km/h**. Η διαφορά της μέγιστης μέσης ταχύτητας από την ελάχιστη αυξήθηκε σε σχέση με το προηγούμενο σενάριο υλοποίησης, ωστόσο αυτό που δημιούργησε τη συγκεκριμένη διαφορά είναι η αύξηση της μέγιστης ταχύτητας ενώ η ελάχιστη διατηρήθηκε στα ίδια επίπεδα με πριν. Αυτό μας δείχνει μία ελαφρώς καλύτερη απόδοση του αλγορίθμου αυτού με την αύξηση της χωρητικότητας της βοηθητικής λωρίδας.

Το διάγραμμα που αφορά τον λόγο R όπως και πριν παρουσιάζει σταθερότητα στην τάση του όπως προκύπτει και από το trendline χρώματος καφέ. Η σταθερότητα αυτή οφείλεται στη σταθερότητα της τάσης της πυκνότητας (αριθμητής) και στη σταθερότητα της τάσης της μέσης ταχύτητας (παρονομαστής). Επιπλέον, ο λόγος R παρουσιάζει κάποιες πολύ υψηλές κορυφές και βυθίσεις οι οποίες αντιστοιχούν κατ' αναλογία σε αυξήσεις της πυκνότητας και μειώσεις της ταχύτητας και τούμπαλιν. Επομένως, μέσω του λόγου R μπορούμε να έχουμε μία πολύ γρήγορη εποπτεία για την αποτελεσματικότητα του αλγορίθμου και να εντοπίσουμε τα σημεία στα οποία υπάρχει αυξημένη κίνηση και συμφόρηση άμεσα.

Τέλος, το παρακάτω ιστόγραμμα παρουσιάζεται το άθροισμα των εμφανίσεων κάθε τιμής πυκνότητας από το σύνολο (30,40,50,60,70,80,90,100).



Διάγραμμα 15. Ιστόγραμμα πυκνοτήτων αλγορίθμου 2^ο σεναρίου

Στο παραπάνω ιστόγραμμα παρατηρούμε ότι και ο αλγόριθμος ομοίως με πριν διατηρεί στο μεγαλύτερο διάστημά του την πυκνότητα στο διάστημα 50-80% με τη μεγαλύτερη

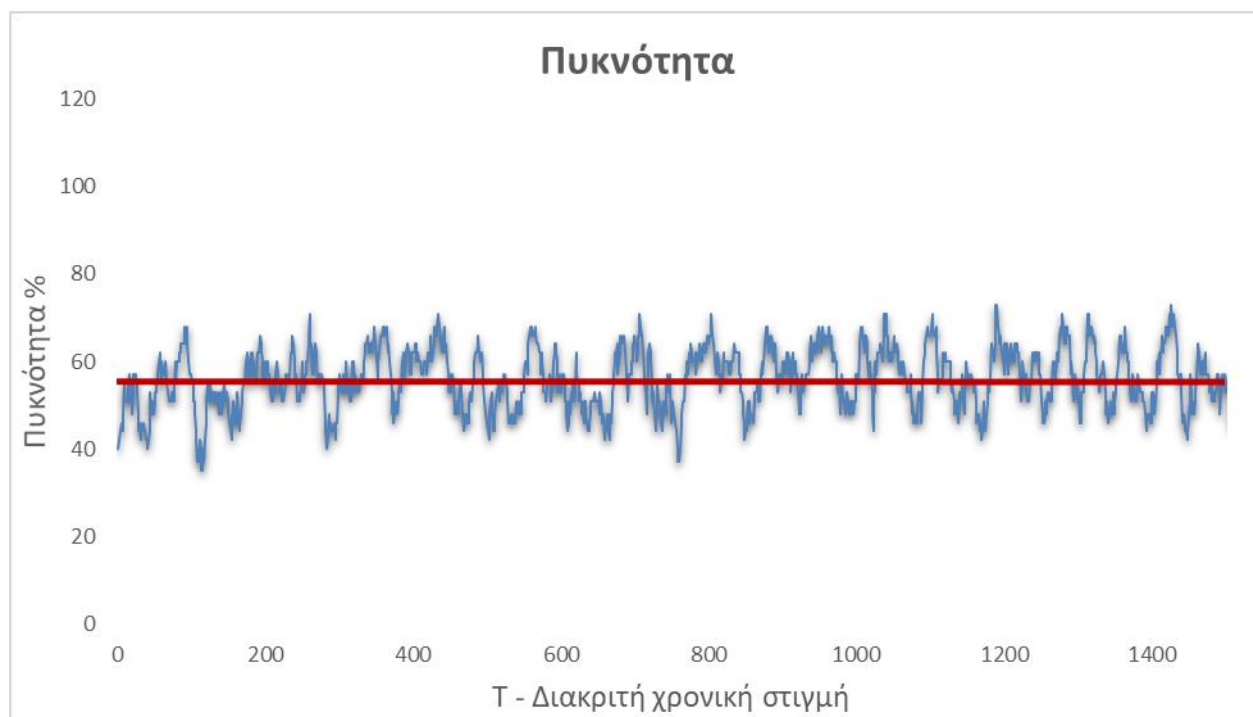
συγκέντρωση να συναντάται στο 60% και αμέσως μετά στο 70%. Το αποτέλεσμα αυτό είναι αρκετά ικανοποιητικό καθώς βρίσκεται γύρω από την οριακή τιμή του 70%.

4.5. 3^ο Σενάριο Υλοποίησης – 3 Λωρίδες & 1 ράμπα

Το 3^ο σενάριο υλοποίησης με μία (1) ράμπα και τρεις (3) λωρίδες, γεγονός που αύξησε την πολυπλοκότητα εκτέλεσης και υλοποίησης του συγκεκριμένου αλγορίθμου. Το συγκεκριμένο σενάριο έχει περιγραφεί αναλυτικά στο Κεφάλαιο 3 στην αντίστοιχη παράγραφο. Ο συγκεκριμένος αλγόριθμος, ομοίως με τους προηγούμενους, εκτελέστηκε μέχρις ότου να περάσουν **1000** αυτοκίνητα συνολικά από τη ράμπα εισόδου. Για να περάσουν 1000 αυτοκίνητα από τη ράμπα ο αλγόριθμος εκτέλεσε **4467** επαναλήψεις μέχρι να επιτευχθεί η συνθήκη τερματισμού. Παρατηρούμε μία πολύ μικρή αύξηση σε σχέση με τον προηγούμενο αλγόριθμο στον αριθμό των επαναλήψεων ωστόσο παρατηρείται μία πολύ μεγάλη αύξηση στον χρόνο εκτέλεσης **12,59** δευτερόλεπτα. Η αύξηση στον χρόνο εκτέλεσης προκύπτει από την ιδιαίτερα αυξημένη πολυπλοκότητα για την προσομοίωση της κίνησης σε τρεις (3) λωρίδες.

Παρακάτω παρουσιάζονται κάποια βασικά διαγράμματα και αριθμητικά αποτελέσματα για την πυκνότητα, τη μέση ταχύτητα και τον λόγο R όπως αυτά έχουν τεκμηριωθεί μαθηματικά στο Κεφάλαιο 2.

4.5.1. Πυκνότητα (Density)



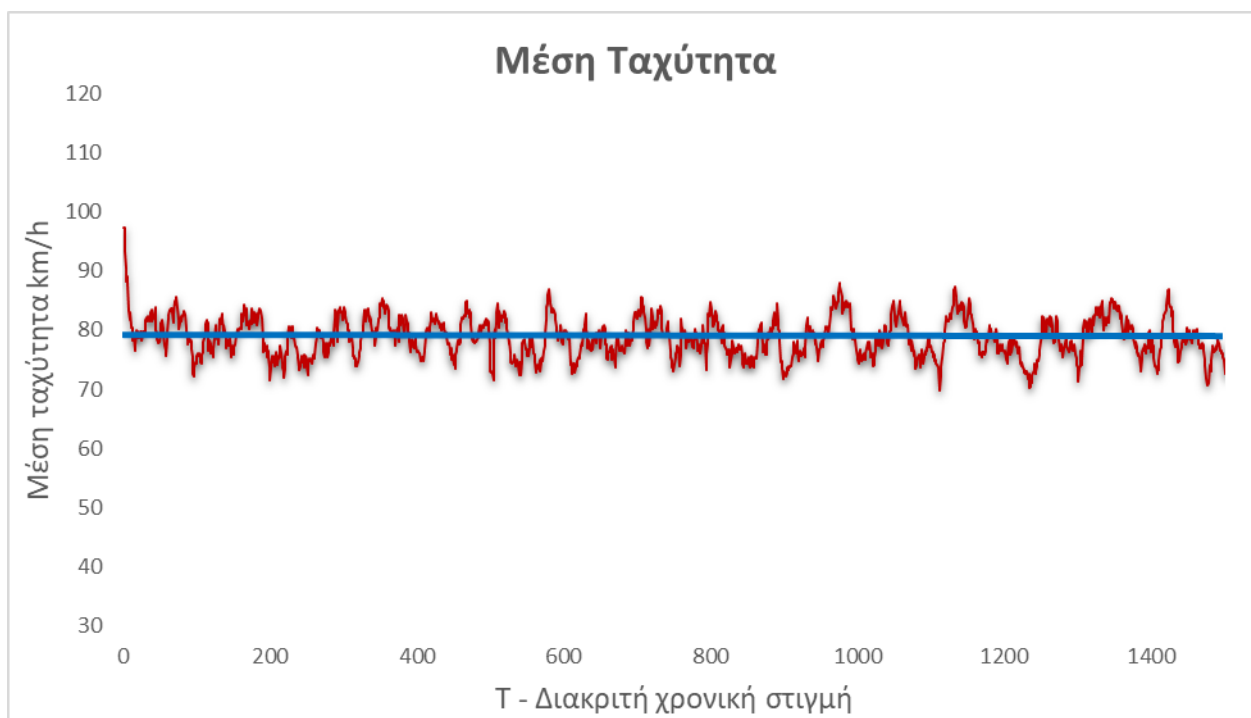
Διάγραμμα 16. Πυκνότητα 3^{ου} σεναρίου υλοποίησης

Στο παραπάνω διάγραμμα απεικονίζεται η πυκνότητα που υπάρχει στο κύριο ρεύμα του αυτοκινητόδρομου για τις πρώτες 1500 χρονικές στιγμές. Παρατηρούμε ότι η πυκνότητα

παρουσιάζει διακυμάνσεις αλλά ταυτόχρονα και μία σταθερότητα ως προς τη διάθεσή της όπως φαίνεται από την κόκκινη γραμμή τάσης (trendline). Σε σχέση με τους προηγούμενους αλγόριθμους και λόγω της αύξησης των λωρίδων παρατηρούμε ότι η πυκνότητα κινείται γύρω από χαμηλότερα ποσοστά. Έτσι τα αποτελέσματα στον συγκεκριμένο αλγόριθμο έχουν διαφοροποιηθεί αρκετά σε σχέση με τους προηγούμενους και αυτό οφείλεται στη διαφορετικότητα της συγκεκριμένης υλοποίησης και στην πιο ρεαλιστική μοντελοποίηση του δρόμου.

Με βάση το παραπάνω σύνολο μετρήσεων η μέση υπολογιζόμενη πυκνότητα είναι **55,37%** το χαμηλότερο ποσοστό σε σχέση με τους αλγόριθμους που έχουν εξεταστεί μέχρι στιγμής. Αυτό οφείλεται στην αύξηση του αριθμού των λωρίδων και άρα στην αυξημένη πολυπλοκότητα της προσομοίωσης της κίνησης των οχημάτων στον αυτοκινητόδρομο.

4.5.2. Μέση ταχύτητα (Mean speed)



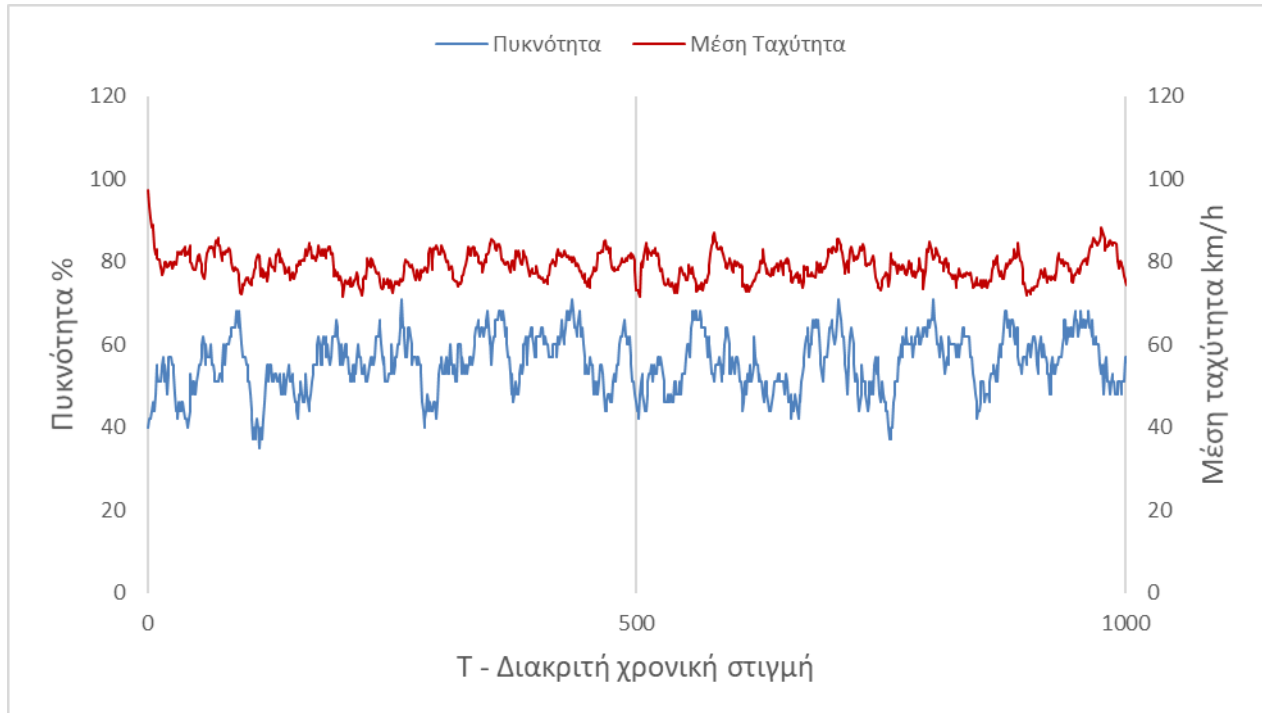
Διάγραμμα 17. Μέση ταχύτητα αλγόριθμου 3^{ου} σεναρίου υλοποίησης

Ομοίως με τα προηγούμενα σενάρια στο παραπάνω διάγραμμα απεικονίζεται η μέση ταχύτητα των οχημάτων που κινούνται στο κύριο ρεύμα του αυτοκινητόδρομου για το 3^ο σενάριο υλοποίησης με τις τρεις λωρίδες κίνησης. Παρατηρούμε ότι η μέση ταχύτητα όπως και η πυκνότητα παρουσιάζει διακυμάνσεις καθ' όλη τη διάρκεια εκτέλεσης του αλγόριθμου. Ωστόσο όπως και σε όλους τους προηγούμενους αλγόριθμους έχει μία σταθερότητα στη διάθεσή της όπως φαίνεται από μπλε γραμμή τάσης (trendline).

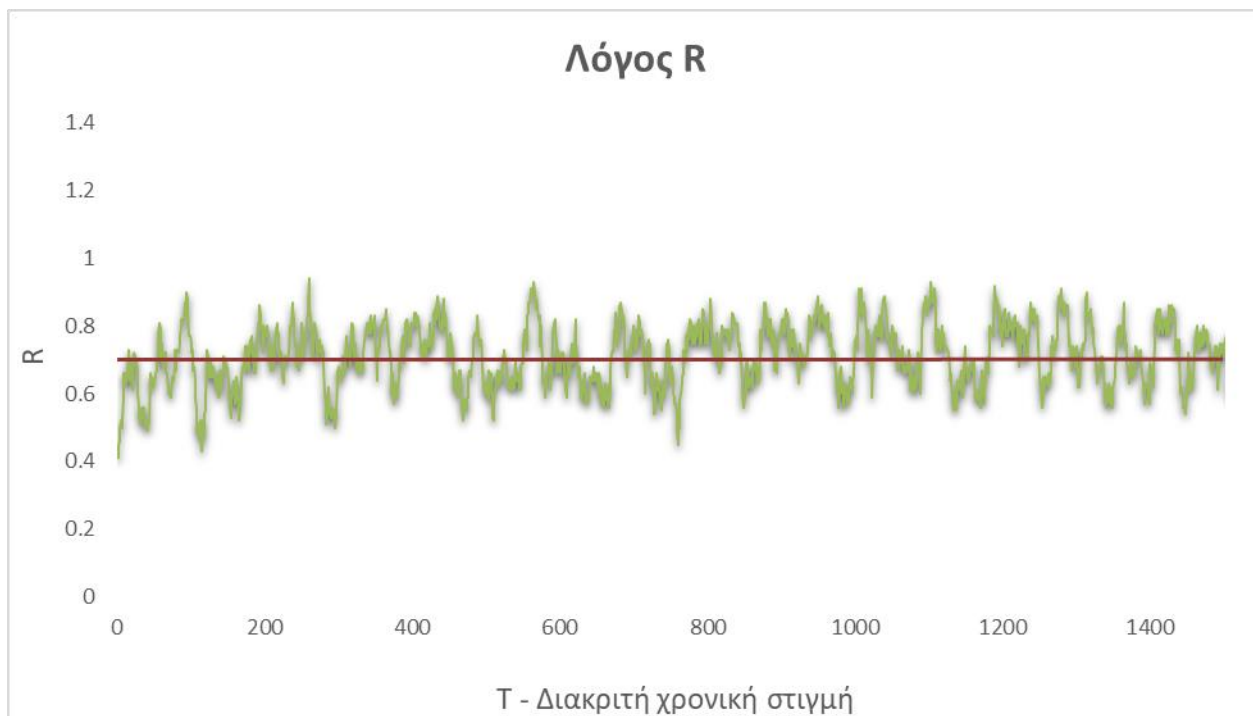
Με βάση το παραπάνω σύνολο μετρήσεων η μέση υπολογιζόμενη ταχύτητα που προκύπτει από τις μέσες ταχύτητες ανά χρονική στιγμή είναι **78,85 km/h**. Παρατηρούμε λοιπόν

ότι ο μέσος όρος των μέσων ταχυτήτων αυξήθηκε αισθητά. Η συγκεκριμένη αύξηση προκύπτει από την αισθητή μείωση των επιπέδων της πυκνότητας που παρουσιάζει το συγκεκριμένο σενάριο υλοποίησης. Το εύρος ταχυτήτων που έχει οριστεί για κάθε όχημα που κινείται εντός του αυτοκινητόδρομου είναι 70 έως 130 km/h.

4.5.3. Πυκνότητα και μέση ταχύτητα



Διάγραμμα 18. Πυκνότητα και Μέση ταχύτητα αλγόριθμου 3^ο σεναρίου

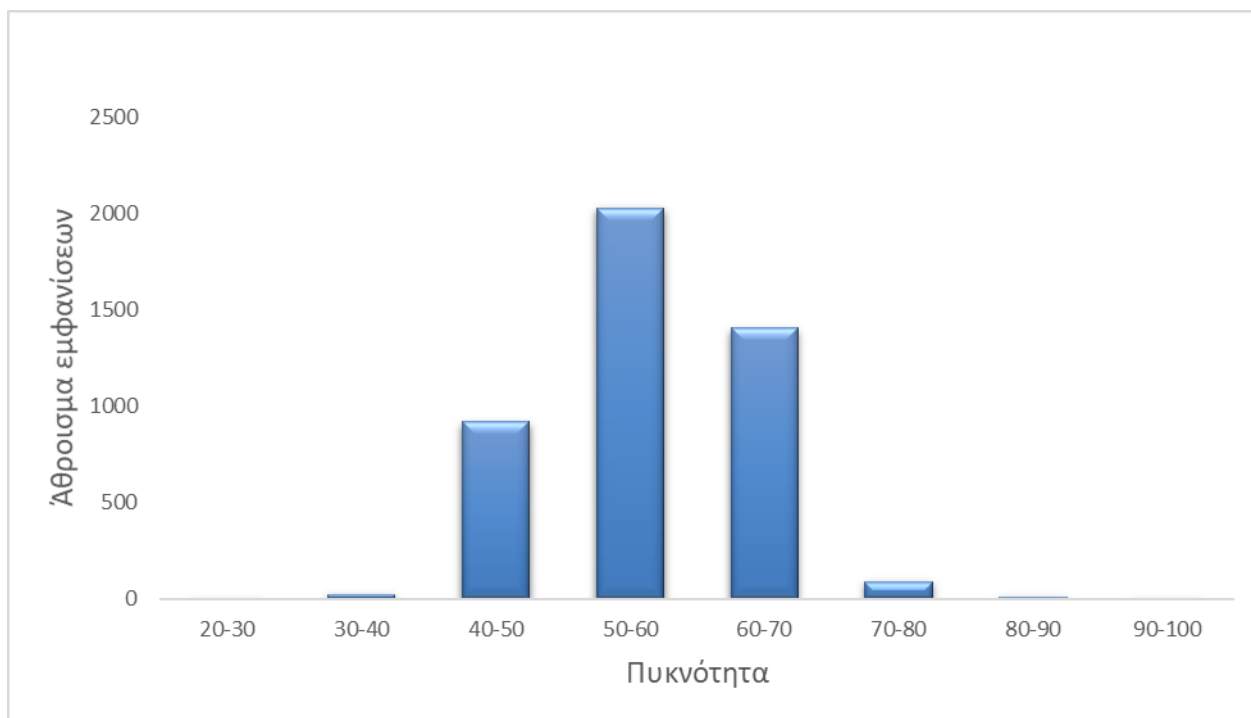


Διάγραμμα 19. Λόγος R αλγόριθμου 3^{ου} σεναρίου

Όπως και προηγουμένως παραπάνω παρατίθενται το συνδυαστικό διάγραμμα μέσης ταχύτητας και πυκνότητας καθώς και το διάγραμμα για τον λόγο R όπως αυτός έχει οριστεί στο Κεφάλαιο 2, δηλαδή πυκνότητα προς ταχύτητα. Παρατηρούμε ότι ομοίως με τις προηγούμενες υλοποιήσεις το διάγραμμα της μέσης ταχύτητας συμπεριφέρεται αντίστροφα από αυτό της πυκνότητας, όπως είναι φυσιολογικό με μία μικρή υστέρηση πάντα η οποία οφείλεται στις ταχείες μεταβολές της πυκνότητας και της ταχύτητας. Έτσι όταν η πυκνότητα αυξάνεται η μέση ταχύτητα αντίστοιχα μειώνεται. Επιπλέον, στον συγκεκριμένο αλγόριθμο παρατηρείται ότι η μέση ταχύτητα κινείται σε υψηλότερα επίπεδα ενώ αντίστοιχα η πυκνότητα σε χαμηλότερα σε σχέση με όλα τα προηγούμενα σενάρια. Έτσι η μέγιστη μέση ταχύτητα είναι **97,33 km/h** και η ελάχιστη ταχύτητα **67,22 km/h**. Η διαφορά της μέγιστης μέσης ταχύτητας από την ελάχιστη έχει περιοριστεί σε σχέση με τα υπόλοιπα σενάρια υλοποίησης και η μέση ταχύτητα διατηρείται σε υψηλότερα επίπεδα όπως εκφράστηκε και από τον μέσο όρο προηγουμένως.

Το διάγραμμα που αφορά τον λόγο R όπως και πριν παρουσιάζει σταθερότητα στην τάση του όπως προκύπτει και από το trendline χρώματος καφέ. Η σταθερότητα αυτή οφείλεται στη σταθερότητα της τάσης της πυκνότητας (αριθμητής) και στη σταθερότητα της τάσης της μέσης ταχύτητας (παρονομαστής) όπως και σε όλα τα υπόλοιπα σενάρια. Επιπλέον, ο λόγος R έχει μικρότερες διακυμάνσεις και βρίσκεται σε χαμηλότερα επίπεδα σε σχέση με τα αποτελέσματα που έχουν εξεταστεί μέχρι τώρα που είναι φυσιολογικό καθώς έχουμε μικρότερα επίπεδα πυκνότητας και μεγαλύτερες μέσες ταχύτητες.

Τέλος, το παρακάτω ιστόγραμμα παρουσιάζει το άθροισμα των εμφανίσεων της πυκνότητας για τα διαστήματα 30%-40%, 40%-50%, 50%-60%, 60%-70%, 70%-80%, 80%-90% και 90%-100%.



Διάγραμμα 20. Ιστόγραμμα πυκνοτήτων αλγορίθμου 3^ο σεναρίου

Στο παραπάνω ιστόγραμμα παρατηρούμε ότι και ο αλγόριθμος διατηρεί στο μεγαλύτερο διάστημά του την πυκνότητα στο διάστημα 40-70% με τη μεγαλύτερη συγκέντρωση να συναντάται ανάμεσα στο 50% με 60%. Το αποτέλεσμα αυτό είναι αρκετά ικανοποιητικό καθώς βρίσκεται γύρω από την οριακή τιμή του 70%. Στον συγκεκριμένο αλγόριθμο έχουμε διαστήματα για την πυκνότητα και όχι μεμονωμένες τιμές καθώς οι τρεις λωρίδες επιτρέπουν τον πιο ακριβή υπολογισμό αυτής.

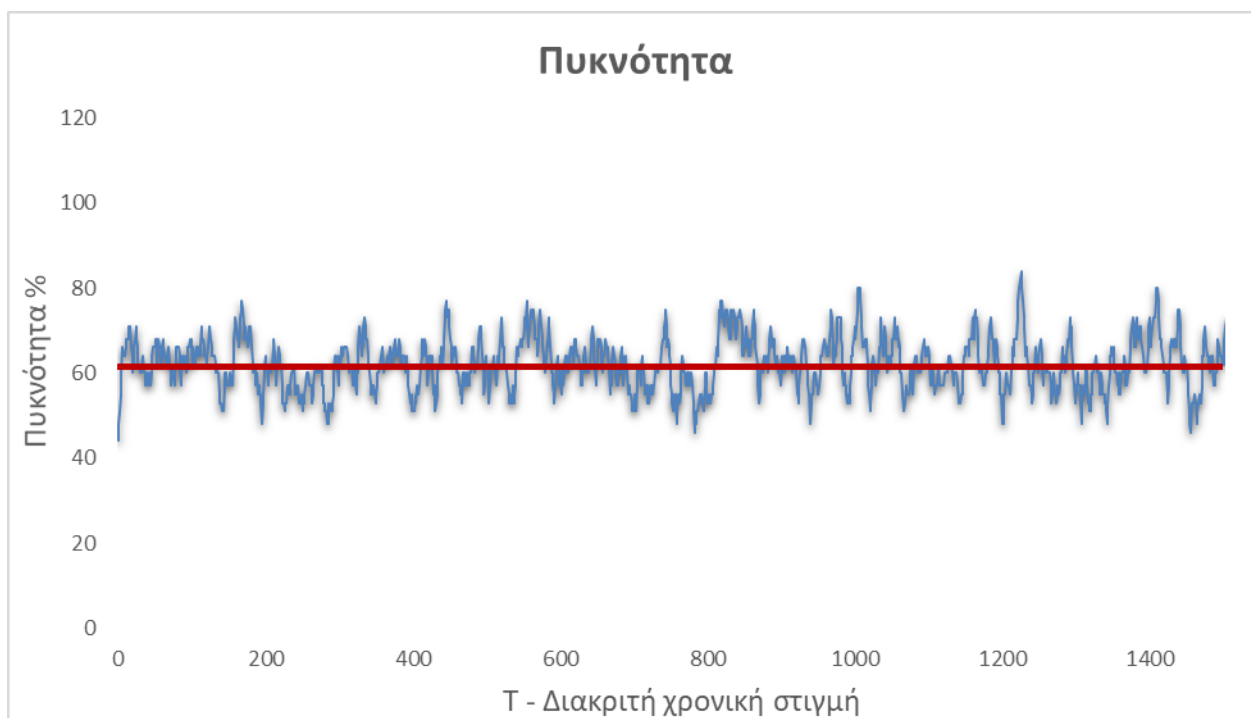
4.6. 4^ο Σενάριο Υλοποίησης – 3 Λωρίδες & 2 Ράμπες – Εξαγωγή Δεδομένων Νευρωνικού Δικτύου (Data Mining)

Το 4^ο σενάριο υλοποίησης με δύο (2) ράμπες και τρεις (3) λωρίδες υλοποιεί ένα αυξημένης πολυπλοκότητας σενάριο υλοποίησης το οποίο αποτέλεσε την πηγή των δεδομένων εκπαίδευσης του νευρωνικού δικτύου που αναπτύχθηκε. Το συγκεκριμένο σενάριο, όπως και όλα τα προηγούμενα, έχει περιγραφεί αναλυτικά στο Κεφάλαιο 3 στην αντίστοιχη παράγραφο. Ο συγκεκριμένος αλγόριθμος, ως συνθήκη τερματισμού είχε χρονική διάρκεια. Η επιλογή αυτή διαφοροποιείται σε σχέση με τους προηγούμενους αλγορίθμους και ο λόγος είναι ότι η ιδιαίτερα αυξημένη πολυπλοκότητά του. Έτσι τα δεδομένα προέκυψαν μετά από εκτέλεση **πέντε (5) λεπτών**. Χαρακτηριστικό που εκφράζει το πόσο απαιτητικός είναι ο συγκεκριμένος

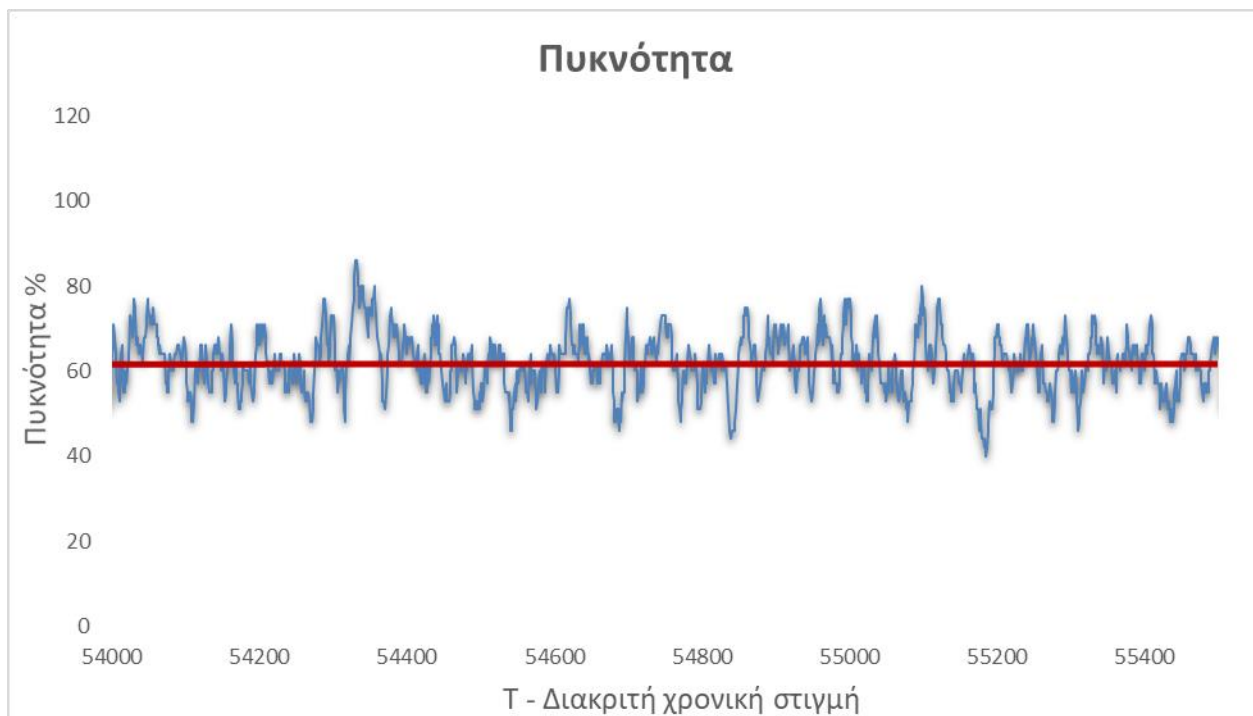
αλγόριθμος είναι ότι σε πέντε λεπτά εκτελέστηκαν **118015** επαναλήψεις και ο αριθμός των αυτοκινήτων που πέρασαν από τη ράμπα 1 είναι πέντε (5) ενώ από τη ράμπα 2 είναι τέσσερα (4). Έτσι σε πέντε (5) λεπτά εκτέλεσης πέρασαν συνολικά εννέα (9) αυτοκίνητα ενώ αντίστοιχα στους προηγούμενους αλγορίθμους σε δευτερόλεπτα μπορούσαν να περάσουν 1000. Το συγκεκριμένο αποτέλεσμα είναι απόλυτα λογικό καθώς πλέον εξετάζεται ένα μοντέλο πολύ κοντά στην πραγματικότητα και αναλογιζόμενοι και την καθημερινή μας εμπειρία σε 5 λεπτά είναι απόλυτα φυσιολογικό να έχουν περάσει ένα φωτεινό σηματοδότη ο οποίος επιτρέπει την είσοδο στο κύριο ρεύμα ενός αυτοκινητόδρομου τόσα αυτοκίνητα.

Παρακάτω παρουσιάζονται κάποια βασικά διαγράμματα και αριθμητικά αποτελέσματα για την πυκνότητα, τη μέση ταχύτητα και τον λόγο R όπως αυτά έχουν τεκμηριωθεί μαθηματικά στο Κεφάλαιο 2. Τα διαγράμματα τα οποία θα παρουσιαστούν παρακάτω θα είναι χωρισμένα ανά κατηγορία και σε κάθε μία από αυτές θα υπάρχουν τρία (3) στιγμιότυπα των 1000 ή 1500 επαναλήψεων έτσι ώστε η πληροφορία να είναι κατανοητή καθώς σε τόσο μεγάλο αριθμό επαναλήψεων είναι αδύνατο να αποτυπωθεί ένα κατανοητό διάγραμμα που θα περιλαμβάνει όλες τις επαναλήψεις – χρονικές στιγμές στον οριζόντιο άξονα.

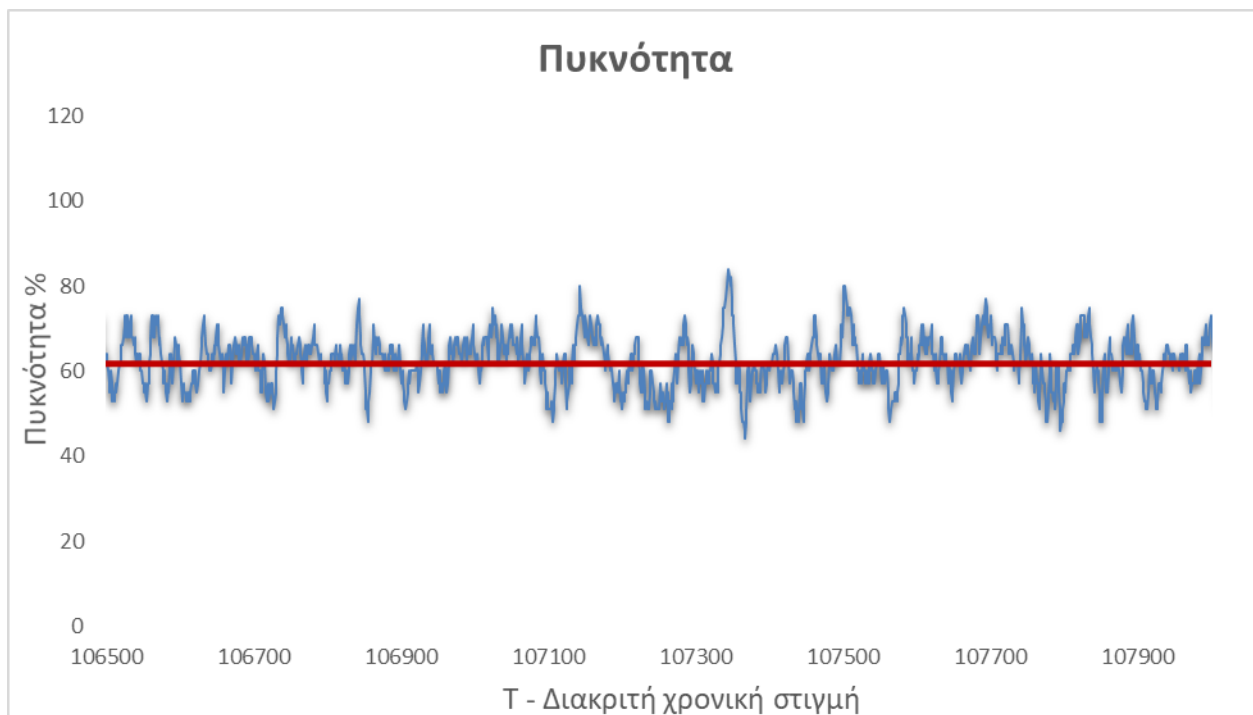
4.6.1. Πυκνότητα (Density)



Διάγραμμα 21. Πυκνότητα 4^{ου} σεναρίου υλοποίησης για $T=0$ έως $T=1500$



Διάγραμμα 22. Πυκνότητα 4^ο σεναρίου υλοποίησης για T=54000 έως T=55500



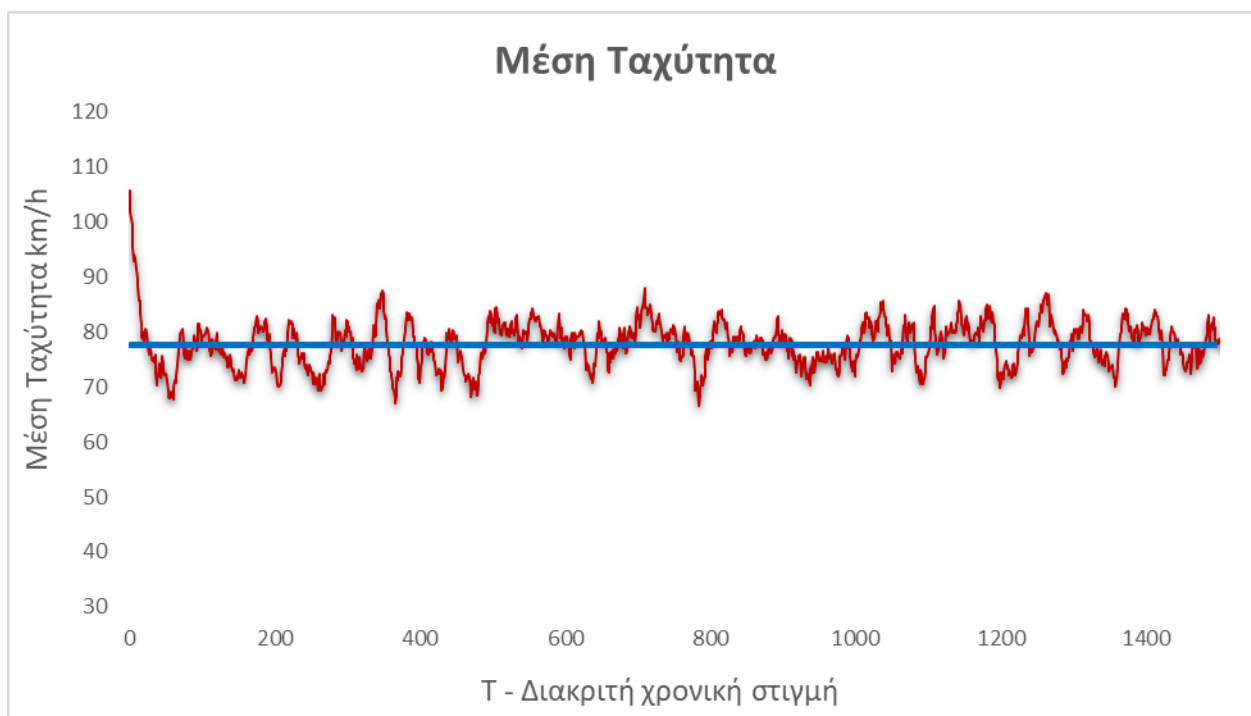
Διάγραμμα 23. Πυκνότητα 4^ο σεναρίου υλοποίησης για T=106500 έως T=108000

Παραπάνω απεικονίζονται τρία (3) στιγμιότυπα της πυκνότητας για 1500 χρονικές στιγμές σε τρία (3) διαφορετικά διαστήματα εκτέλεσης του αλγορίθμου, ένα στην αρχή, ένα στη μέση και ένα στο τέλος. Παρατηρούμε ότι σε όλα τα στιγμιότυπα η πυκνότητα παρουσιάζει διακυμάνσεις αλλά ταυτόχρονα και μία σταθερότητα ως προς τη διάθεσή της όπως φαίνεται από την κόκκινη γραμμή τάσης (trendline). Σε σχέση με τους προηγούμενους αλγορίθμους και λόγω της αύξησης της αυστηρότητας των ελέγχων η πυκνότητα διατηρείται χαμηλότερα

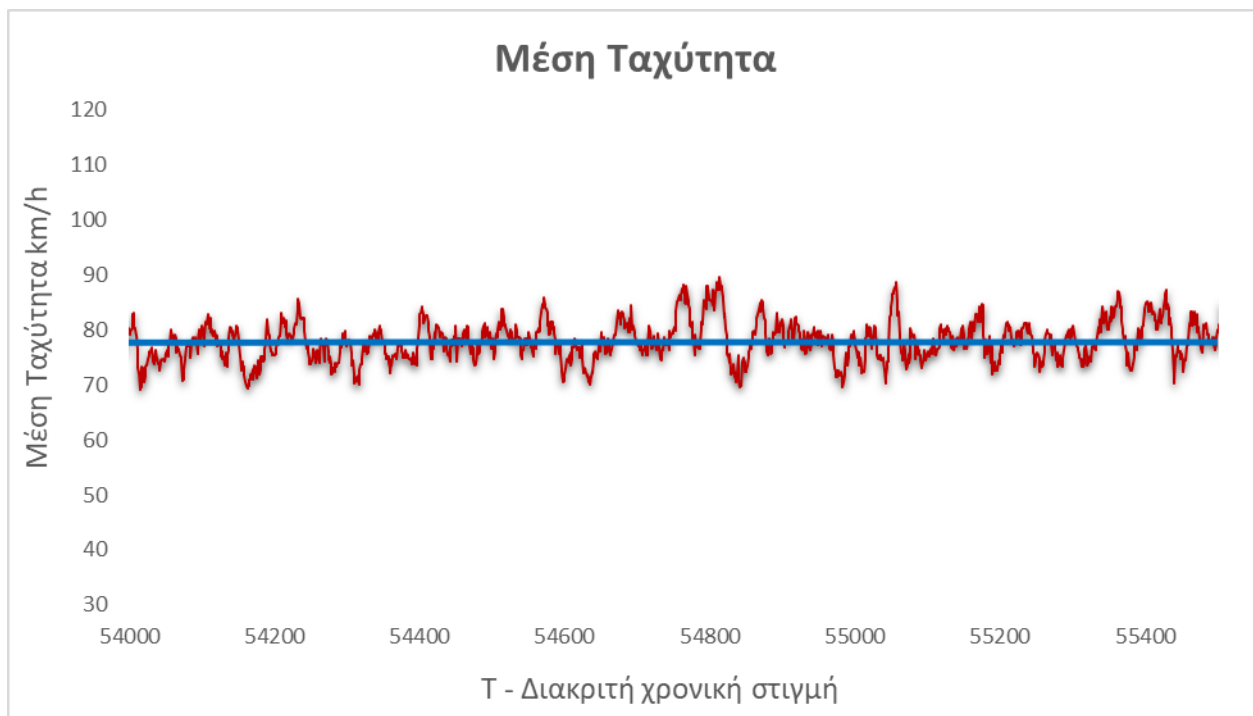
επίπεδα και πέρα κάποιων κορυφών συνήθως κάτω από 80%. Αυτό υποδεικνύει ότι ο αλγόριθμος λειτουργεί σωστά και στον αυτοκινητόδρομο αποφεύγονται προβλήματα συμφόρησης.

Με βάση το σύνολο μετρήσεων από το οποίο προέκυψαν και τα παραπάνω διαγράμματα η μέση υπολογιζόμενη πυκνότητα είναι **61,70%**. Σε σχέση με τον προηγούμενο αλγόριθμο αυτό το νούμερο είναι ελαφρώς μεγαλύτερο γεγονός που οφείλεται στην προσθήκη της δεύτερης ράμπας και άρα στη συνεχή πλήρωση του αυτοκινητοδρόμου σε ποσοστά πυκνότητας έως και 70%. Η αύξηση αυτή δεν είναι κάτι μη αναμενόμενο καθώς όπως έχει ήδη αναφερθεί στα πρώτα δύο (2) κεφάλαια της παρούσας εργασίας στόχος τέτοιων αλγορίθμων και γενικά των μεθόδων Ramp Metering είναι να διατηρούν την πυκνότητα σε ένα συγκεκριμένο εύρος κοντά στην οριακή τιμή που στην προκειμένη περίπτωση είναι το 70%.

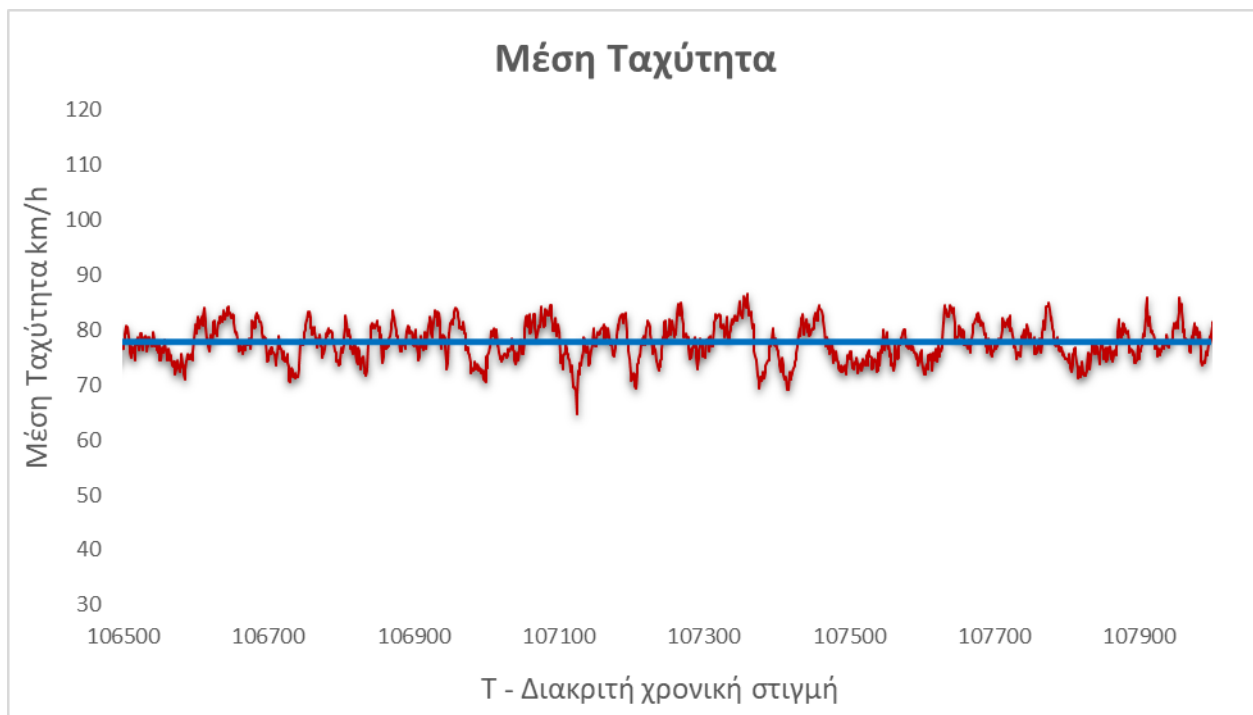
4.6.2. Μέση ταχύτητα (Mean speed)



Διάγραμμα 24. Μέση ταχύτητα αλγορίθμου 4^{ov} σεναρίου υλοποίησης για T=0 έως T=1500



Διάγραμμα 25. Μέση ταχύτητα αλγορίθμου 4^{ου} σεναρίου υλοποίησης για $T=54000$ έως $T=55500$



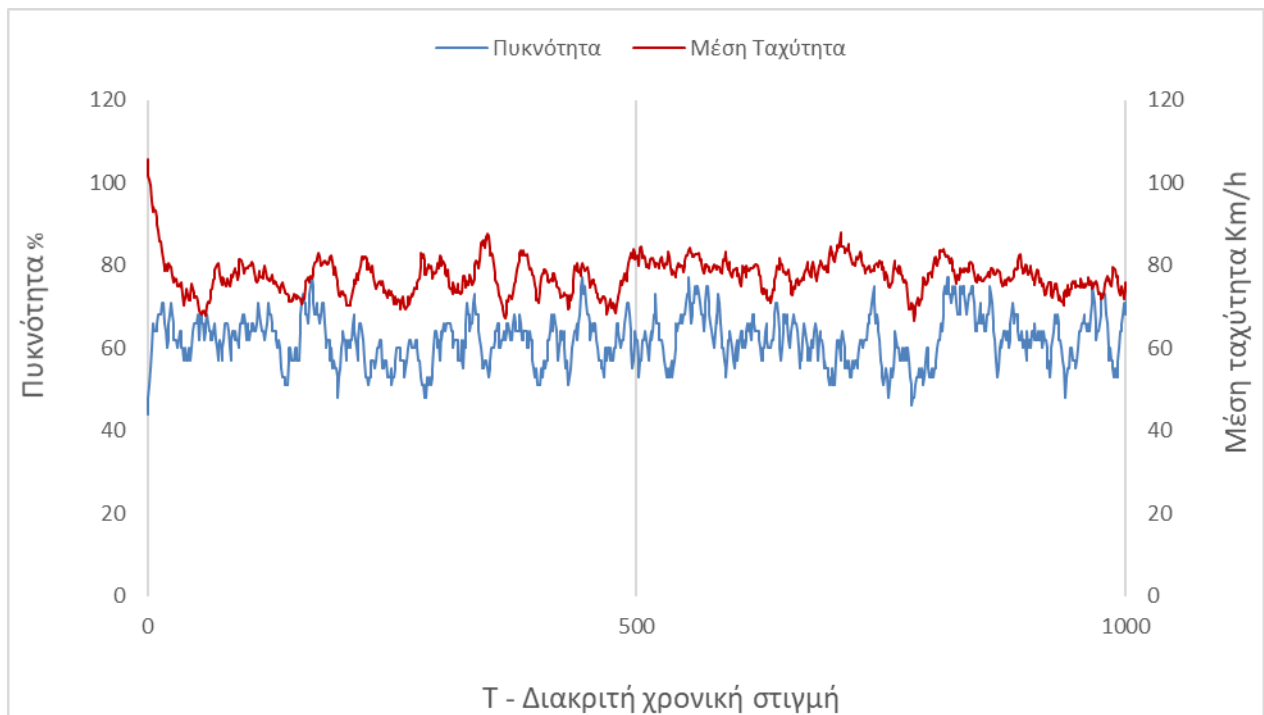
Διάγραμμα 26. Μέση ταχύτητα αλγορίθμου 4^{ου} σεναρίου υλοποίησης για $T=106500$ έως $T=108000$

Ομοίως με την πυκνότητα παραπάνω παρουσιάζονται τρία (3) τυχαία στιγμιότυπα των 1500 χρονικών στιγμών στα οποία απεικονίζεται η μέση ταχύτητα των οχημάτων που κινούνται στο κύριο ρεύμα του αυτοκινητόδρομου. Παρατηρούμε σε όλα τα παραπάνω διαγράμματα ότι η μέση ταχύτητα όπως και η πυκνότητα παρουσιάζει διακυμάνσεις καθ' όλη τη διάρκεια εκτέλεσης του αλγορίθμου οι οποίες όμως κινούνται σε ένα περιορισμένο εύρος πριν ελαχίστων κορυφών και βυθίσεων. Το εύρος αυτό κυμαίνεται μεταξύ 70 km/h και 90 km/h.

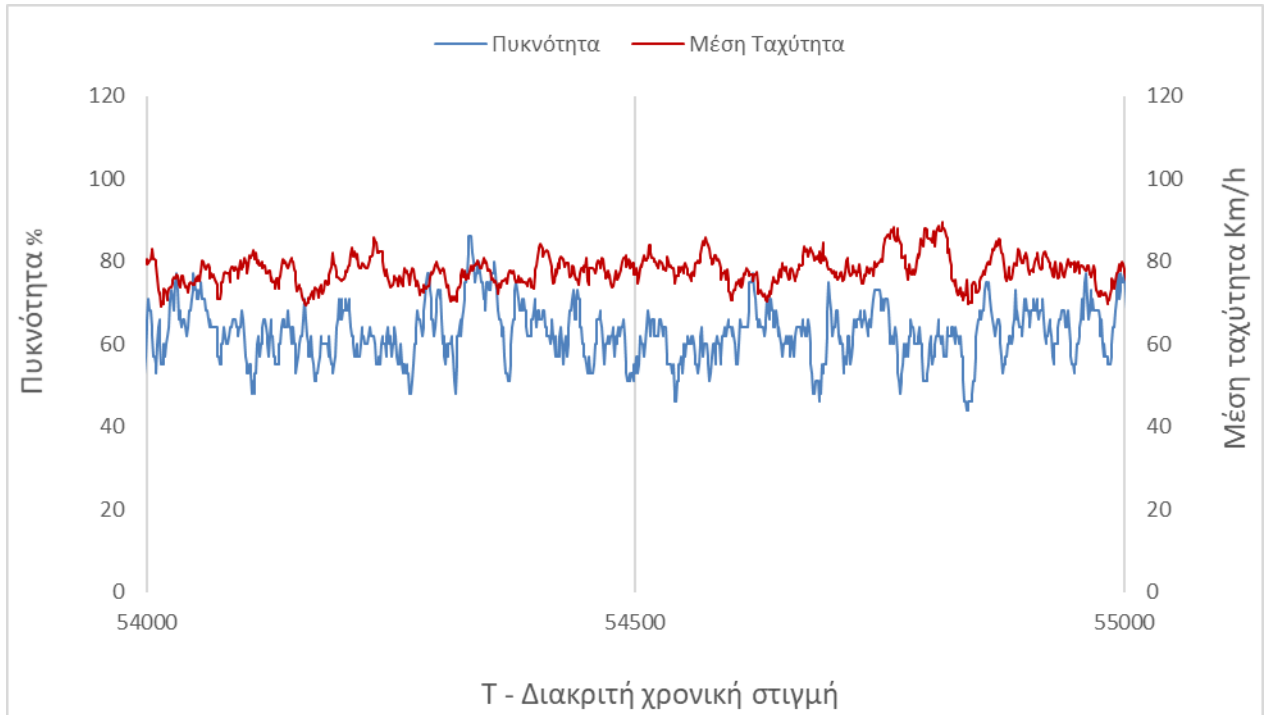
Επιπλέον, ο μέσος όρος των μέσων ταχυτήτων έχει μία σταθερότητα στη διάθεσή του όπως φαίνεται από μπλε γραμμή τάσης (trendline).

Με βάση το σύνολο μετρήσεων από το οποίο προέκυψαν τα παραπάνω διαγράμματα η μέση υπολογιζόμενη ταχύτητα που προκύπτει από τις μέσες ταχύτητες ανά χρονική στιγμή είναι **77,86 km/h**. Παρατηρούμε λοιπόν ότι ο μέσος όρος των μέσων ταχυτήτων διατηρείται σε αρκετά ψηλά επίπεδα γεγονός που εκφράζει ότι παρά την εισαγωγή και δεύτερης ράμπας ο αλγόριθμος λειτουργεί ορθά και υπάρχει συνεχόμενη ροή και με υψηλή ταχύτητα σε όλο το μήκος του τμήματος του αυτοκινητόδρομου που εξετάζεται.

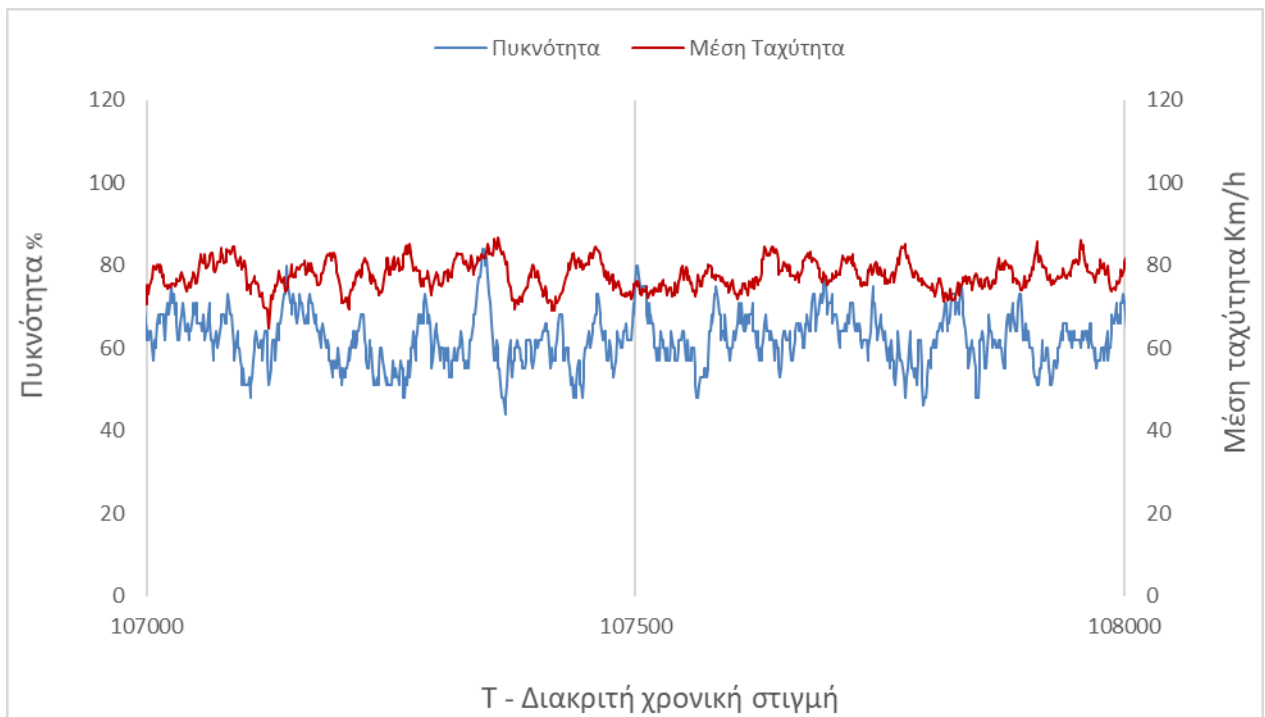
4.6.3. Πυκνότητα και μέση ταχύτητα



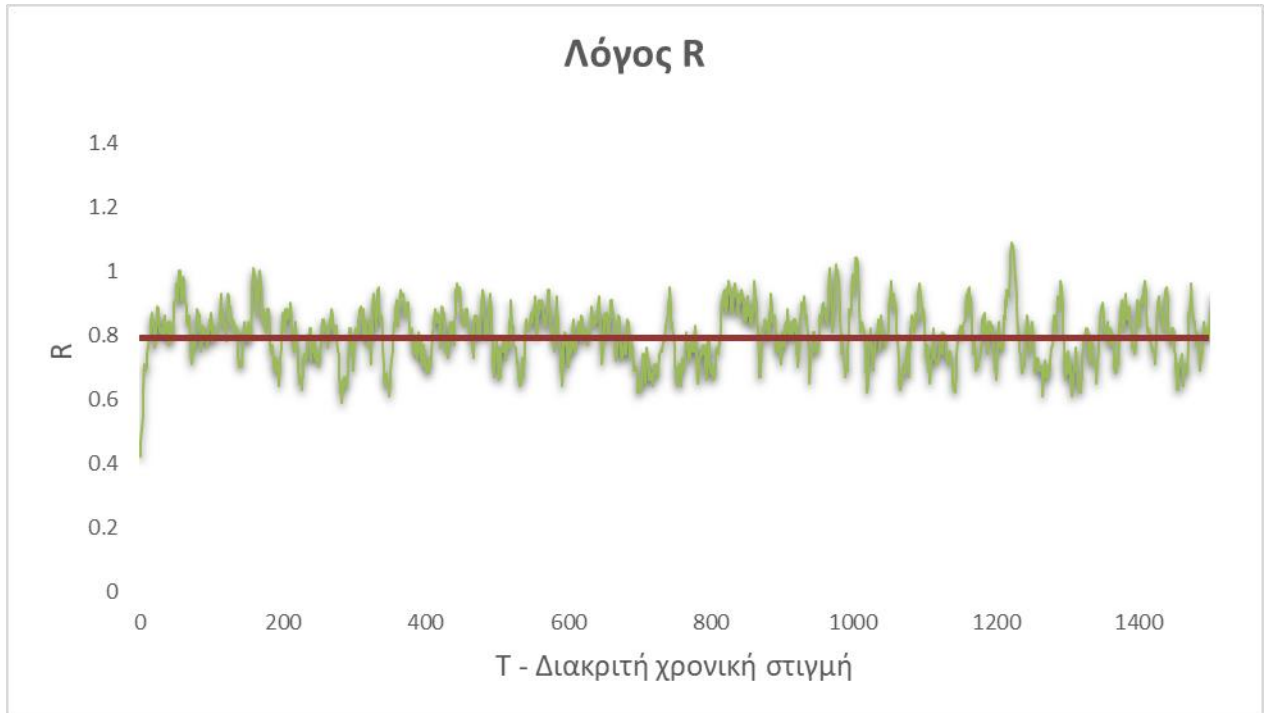
Διάγραμμα 27. Πυκνότητα και μέση ταχύτητα αλγόριθμου 4^ο σεναρίου για T=0 έως T=1000



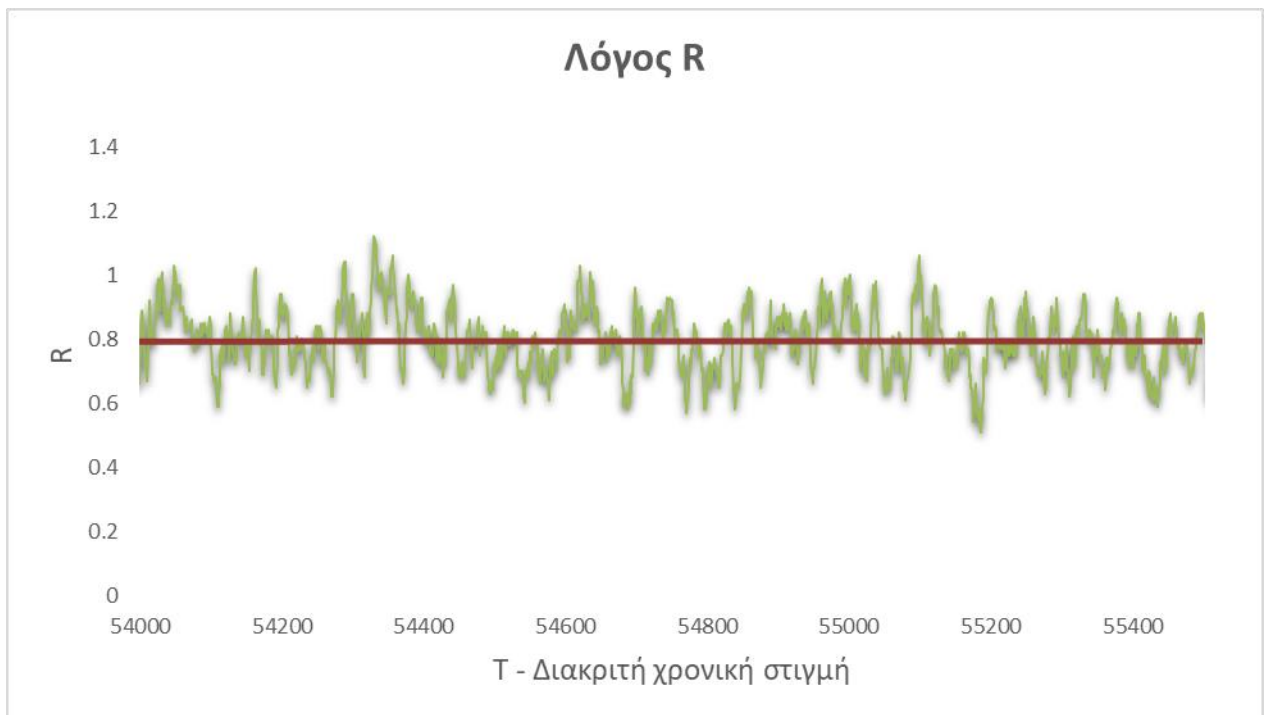
Διάγραμμα 28. Πυκνότητα και μέση ταχύτητα αλγορίθμου 4^ο σεναρίου για T=54000 έως T=55000



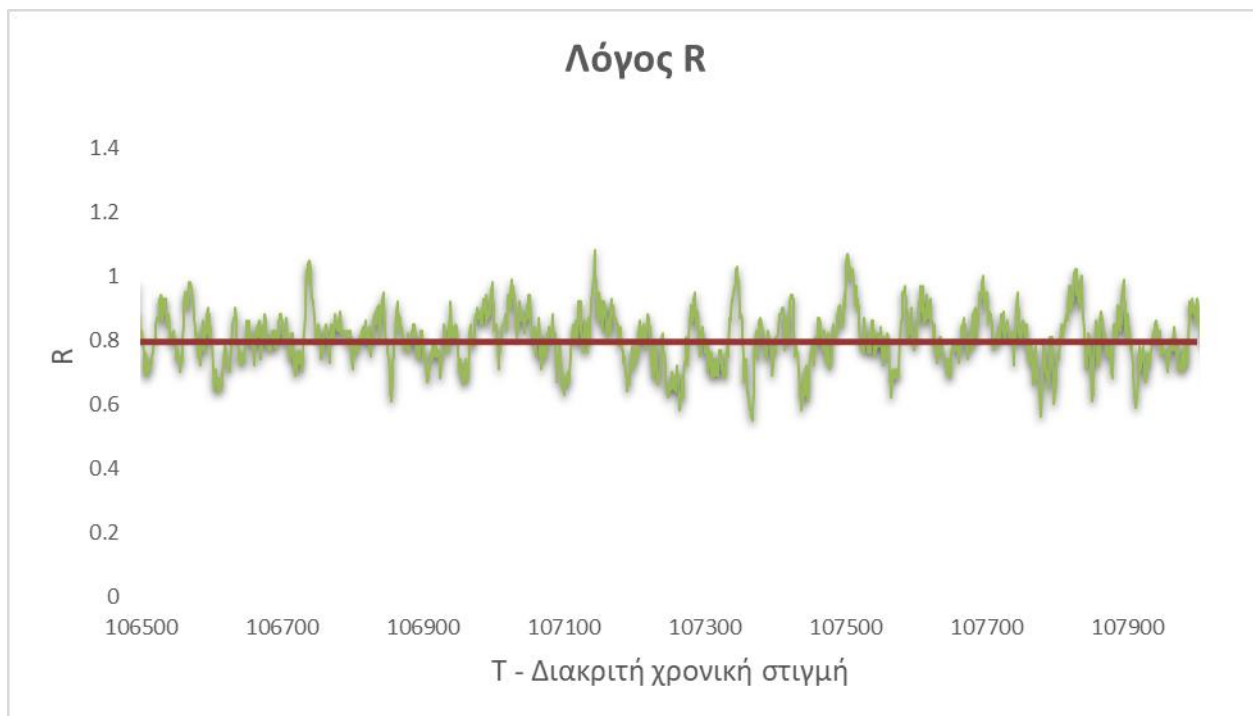
Διάγραμμα 29. Πυκνότητα και μέση ταχύτητα αλγορίθμου 4^ο σεναρίου για T=107000 έως T=108000



Διάγραμμα 30. Λόγος R αλγόριθμου 4^{ου} σεναρίου για $T=0$ έως $T=1500$



Διάγραμμα 31. Λόγος R αλγόριθμου 4^{ου} σεναρίου για $T=54000$ έως $T=55500$

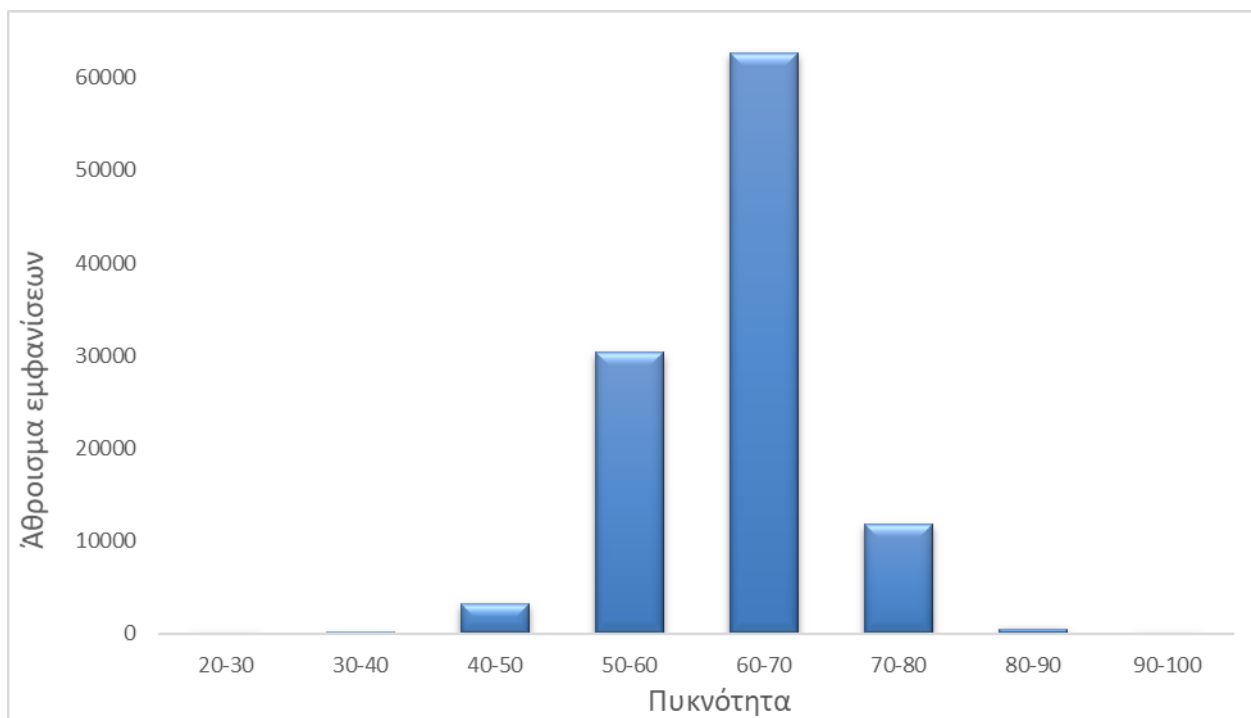


Διάγραμμα 32. Λόγος R αλγόριθμου 4^{ου} σεναρίου για T=106500 έως T=108000

Παραπάνω παρατίθενται τρία (3) στιγμιότυπα εύρους 1000 χρονικών στιγμών για το συνδυαστικό διάγραμμα μέσης ταχύτητας και πυκνότητας καθώς και τρία (3) στιγμιότυπα εύρους 1500 χρονικών στιγμών του διαγράμματος για τον λόγο R όπως αυτός έχει οριστεί στο Κεφάλαιο 2, δηλαδή πυκνότητα προς ταχύτητα. Παρατηρούμε ότι ομοίως με τις προηγούμενες υλοποιήσεις το διάγραμμα της μέσης ταχύτητας συμπεριφέρεται αντίστροφα από αυτό της πυκνότητας, όπως είναι φυσιολογικό με μία μικρή υστέρηση πάντα η οποία οφείλεται στις ταχείες μεταβολές της πυκνότητας και της ταχύτητας. Έτσι όταν η πυκνότητα αυξάνεται η μέση ταχύτητα αντίστοιχα μειώνεται. Στον συγκεκριμένο αλγόριθμο παρατηρείται ότι η μέση ταχύτητα και η πυκνότητα παρουσιάζουν μικρότερες μεταβολές γύρω από ένα συγκεκριμένο εύρος. Η μέγιστη μέση ταχύτητα είναι **105,67 km/h** η οποία όμως συναντάται στις αρχικές χρονικές στιγμές όπως φαίνεται από το αρχικό διάγραμμα. Μετά την εξισορρόπηση του ελέγχου αυτό το νούμερο περιορίζεται περίπου στα 95 km/h. Αντίστοιχα η ελάχιστη ταχύτητα **51,84 km/h**.

Το διάγραμμα που αφορά τον λόγο R όπως και σε όλες τις προηγούμενες περιπτώσεις παρουσιάζει σταθερότητα στην τάση του όπως προκύπτει και από το trendline χρώματος καφέ. Η σταθερότητα αυτή οφείλεται στη σταθερότητα της τάσης της πυκνότητας (αριθμητής) και στη σταθερότητα της τάσης της μέσης ταχύτητας (παρονομαστής) όπως και σε όλα τα υπόλοιπα σεναρία. Όπως είναι φανερό ο λόγος R παρουσιάζει πολύ μεγάλη συσχέτιση με την πυκνότητα και όπως φαίνεται τα διαγράμματα των δύο αυτών τιμών παρουσιάζουν τεράστια ομοιότητα.

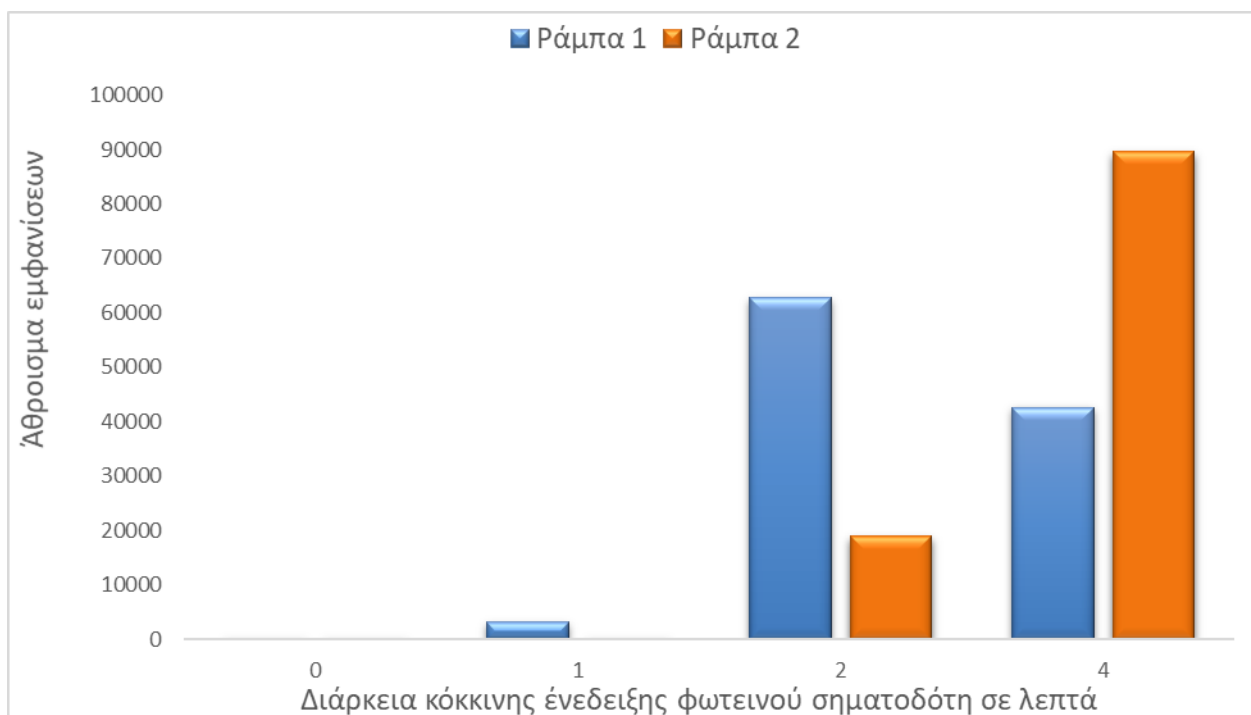
Το παρακάτω ιστόγραμμα παρουσιάζεται το άθροισμα των εμφανίσεων της πυκνότητας για τα διαστήματα 30%-40%, 40%-50%, 50%-60%, 60%-70%, 70%-80%, 80%-90% και 90%-100%.



Διάγραμμα 33. Ιστόγραμμα πυκνοτήτων αλγόριθμου 4^ο σεναρίου

Στο παραπάνω ιστόγραμμα παρατηρούμε ότι και ο αλγόριθμος ομοίως με πριν διατηρεί στο μεγαλύτερο διάστημά του την πυκνότητα στο διάστημα 50-80% με συντριπτική συγκέντρωση ανάμεσα στο 60% με 70%. Το αποτέλεσμα αυτό είναι ιδιαίτερα ικανοποιητικό και σαφώς βελτιωμένο σε σχέση με τον προηγούμενο αλγόριθμο που παρουσιάστηκε, ο οποίος παρουσίαζε τη μεγαλύτερη συγκέντρωση στο 50% με 60%, καθώς βρίσκεται γύρω από την οριακή τιμή του 70% και άρα υποδηλώνει ότι ο έλεγχος που εφαρμόζεται για το Ramp Metering λειτουργεί.

Τέλος, στον συγκεκριμένο αλγόριθμο το ζητούμενο ήταν να καθοριστεί η διάρκεια που ο φωτεινός σηματοδότης έχει κόκκινη ένδειξη από ένα προκαθορισμένο σύνολο {0,1,2,4} σε λεπτά. Η συγκεκριμένη διάρκεια καθορίζεται κάθε χρονική στιγμή από τις παραμέτρους που έχει ο αυτοκινητόδρομος εκείνη τη στιγμή. Οι διάρκειες για τη ράμπα 1 και 2 αποτελούν ουσιαστικά τα outputs του νευρωνικού δικτύου τα αποτελέσματα του οποίου θα παρουσιαστούν στην επόμενη παράγραφο. Παρακάτω φαίνεται ένα ιστόγραμμα στο οποίο παρουσιάζεται ο αριθμός εμφανίσεων για κάθε χρονική διάρκεια της κάθε ράμπας.



Διάγραμμα 34. Ιστόγραμμα κατανομής χρόνων κόκκινης ένδειξης σηματοδότη 4ου σεναρίου υλοποίησης

Όπως φαίνεται στο παραπάνω ιστόγραμμα ο χρόνος των δύο (2) λεπτών κόκκινης ένδειξης του φωτεινού σηματοδότη είναι η πλειοψηφία για την πρώτη ράμπα και ο χρόνος της ένδειξης των τεσσάρων (4) λεπτών για τη δεύτερη ράμπα. Το αποτέλεσμα είναι φυσιολογικό καθώς στην πρώτη ράμπα υπάρχει πηγή αύξησης της πυκνότητας μόνο από το κύριο ρεύμα του αυτοκινητόδρομου που προηγείται της πρώτης ράμπας ενώ στη δεύτερη ράμπα υπάρχουν δύο πηγές εισόδου που είναι το τμήμα του κύριου ρεύματος που προηγείται της ράμπας αυτής και η πρώτη ράμπα.

Συμπερασματικά, λοιπόν, ο συγκεκριμένος αλγόριθμος αποτελεί μία πολύ καλή προσέγγιση της πραγματικότητας με ιδιαίτερα αυξημένη πολυπλοκότητα που πηγάζει τόσο από την προσομοίωση του αυτοκινητόδρομου όσο και από την ύπαρξη δύο ραμπών με αυστηρές συνθήκες ελέγχου.

4.7. Νευρωνικό Δίκτυο – Αποτελέσματα

Το νευρωνικό δίκτυο εκπαιδεύτηκε με ένα υποσύνολο των δεδομένων που παρήχθησαν από το 4^ο σενάριο υλοποίησης που περιγράφηκε παραπάνω. Τα υπόλοιπα δεδομένα που δεν συμμετείχαν στην εκπαίδευση του νευρωνικού δικτύου αποτέλεσαν το σύνολο επαλήθευσης των αποτελεσμάτων του (evaluation set). Έτσι, το νευρωνικό δίκτυο δέχθηκε ως είσοδο για την εκπαίδευσή του ένα σύνολο δεδομένων όπως φαίνεται στον παρακάτω πίνακα της τάξεως των 60000 εγγραφών.

T	Section 1	Section 2	Section 3	...	Section 45	Density	Mean Speed	R	Output time ramp 1	Output time ramp 2
T=1										
T=2										
T=3										
...										
T=n										

Πίνακας 4. Διάταξη δεδομένων εκπαίδευσης νευρωνικού δικτύου

Κατόπιν της εκτέλεσης της εκπαίδευσης του νευρωνικού παρήχθησαν οι προβλέψεις του για τους χρόνους που ο φωτεινός σηματοδότης είναι κόκκινος για τη ράμπα 1 και τη ράμπα 2.

4.7.1. Ράμπα 1 – Αποτελέσματα

Προκειμένου να εξασφαλιστεί η αξιοπιστία των αποτελεσμάτων για τη ράμπα 1, το νευρωνικό δίκτυο εκτέλεσε πενήντα αναγνώσεις (epochs) των δεδομένων εκπαίδευσης όπως φαίνεται παρακάτω:

```
Epoch 36/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0519 - acc: 0.9788 - val_loss: 0.0812 - val_acc: 0.9712
Epoch 37/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0707 - acc: 0.9733 - val_loss: 0.0574 - val_acc: 0.9754
Epoch 38/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0516 - acc: 0.9783 - val_loss: 0.0801 - val_acc: 0.9717
Epoch 39/50
60712/60712 [=====] - 0s 1us/step - loss: 0.0692 - acc: 0.9737 - val_loss: 0.0540 - val_acc: 0.9771
Epoch 40/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0481 - acc: 0.9801 - val_loss: 0.0740 - val_acc: 0.9725
Epoch 41/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0631 - acc: 0.9747 - val_loss: 0.0543 - val_acc: 0.9763
Epoch 42/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0483 - acc: 0.9793 - val_loss: 0.0766 - val_acc: 0.9726
Epoch 43/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0650 - acc: 0.9747 - val_loss: 0.0579 - val_acc: 0.9733
Epoch 44/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0516 - acc: 0.9763 - val_loss: 0.0942 - val_acc: 0.9713
Epoch 45/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0806 - acc: 0.9733 - val_loss: 0.0709 - val_acc: 0.9661
Epoch 46/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0641 - acc: 0.9681 - val_loss: 0.1114 - val_acc: 0.9706
Epoch 47/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0962 - acc: 0.9725 - val_loss: 0.0412 - val_acc: 0.9849
Epoch 48/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0348 - acc: 0.9875 - val_loss: 0.0447 - val_acc: 0.9803
Epoch 49/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0355 - acc: 0.9831 - val_loss: 0.0382 - val_acc: 0.9856
Epoch 50/50
60712/60712 [=====] - 0s 2us/step - loss: 0.0316 - acc: 0.9885 - val_loss: 0.0467 - val_acc: 0.9799
```

Εικόνα 30. Epochs για τη ράμπα 1

Όπως φαίνεται στο παραπάνω σχήμα σε κάθε επόμενο «πέρασμα» η ακρίβεια και το loss γενικά βελτιώνεται. Κάποιες ταλαντώσεις που ενδεχομένως να παρατηρούνται από ένα epoch σε επόμενο είναι φυσιολογικό καθώς το νευρωνικό δίκτυο δεν αποτελεί μία επαναλήπτική μέθοδο που σε κάθε επανάληψη βελτιώνει την ακρίβεια. Επιπλέον, η έννοια του epoch δεν έχει ταυτόσημη σημασία με αυτή της επανάληψης, αρά και η συμπεριφορά του είναι διαφορετική. Η ακρίβεια των αποτελεσμάτων του νευρωνικού βελτιώνεται με βάση την ποιότητα των δεδομένων και την εκπαίδευσή του. Έτσι παραπάνω φαίνεται ότι μετά από 50 epochs το νευρωνικό δίκτυο είχε ακρίβεια της τάξης 97,99% και απώλεια δεδομένων 4,67%.

Έπειτα λοιπόν από την ολοκλήρωση της εκπαίδευσης και την παραγωγή των προβλέψεων για τους χρόνους της ράμπας 1, η σύγκριση των αποτελεσμάτων με το evaluation test έδωσε τα παρακάτω αποτελέσματα:

```
108416/108416 [=====] - 3s 23us/step  
acc: 98.20%  
loss: 3.99%
```

Εικόνα 31. Ακρίβεια και απώλεια δεδομένων αποτελεσμάτων για τη ράμπα 1

Όπως φαίνεται στην παραπάνω εικόνα τα αποτελέσματα για τη ράμπα 1 είναι άκρως θετικά καθώς έχουν ακρίβεια **98,20%** και απώλεια μόλις **3,99%**.

.Στο μοντέλο μας διαπιστώνεται, ότι η απώλεια επικύρωσης και η ακρίβεια επικύρωσης συγχρονίζονται με την ακρίβεια της απώλειας εκπαίδευσης και της εκπαίδευσης.

Ένα ακόμα άκρως σημαντικό αποτέλεσμα το οποίο υποδηλώνει την ορθότητα ή όχι του νευρωνικού δικτύου είναι εάν τα αποτελέσματα των τιμών παρουσιάζουν overfit ή underfit. Η υπερφόρτωση (Overfitting) δίνει μια διαίσθηση ότι του κατά πόσο το δίκτυο έχει απομνημονεύσει τα δεδομένα εκπαίδευσης σε ικανοποιητικό ή πολύ καλό βαθμό, αλλά δεν εγγυάται ότι θα δουλέψει σε αόρατα δεδομένα και γι 'αυτό υπάρχει διαφορά στην ακρίβεια της εκπαίδευσης και της επικύρωσης.

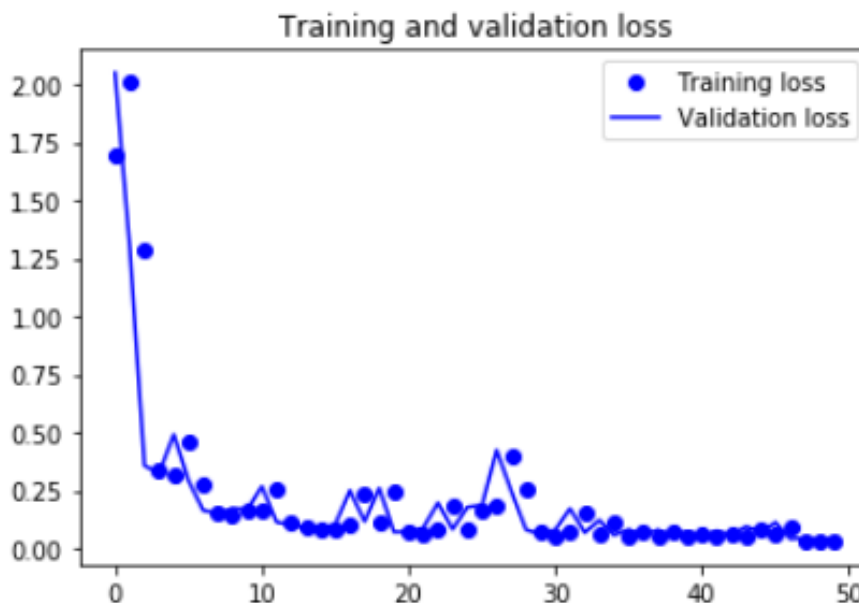
Παρόλο που η γραμμή απώλειας και ακρίβειας επικύρωσης δεν είναι απόλυτα γραμμική όπως φαίνεται παρακάτω δείχνει ότι το μοντέλο δεν έχει υποστεί υπερφόρτωση.

Παρακάτω φαίνονται τα αντίστοιχα διαγράμματα για τους χρόνους της ράμπας 1:



Διάγραμμα 35. Διάγραμμα overfit για τη ράμπα 1

Όπως φαίνεται και προαναφέρθηκε οι χρόνοι για τη ράμπα 1 δεν παρουσιάζουν απόκλιση προς τα πάνω overfit πέρα από κάποιες τιμές στα αρχικά epochs. Η συγκεκριμένη παρατήρηση είναι εξαιρετικά σημαντική καθώς επιβεβαιώνει την αξιοπιστία των αποτελεσμάτων πρόβλεψης.

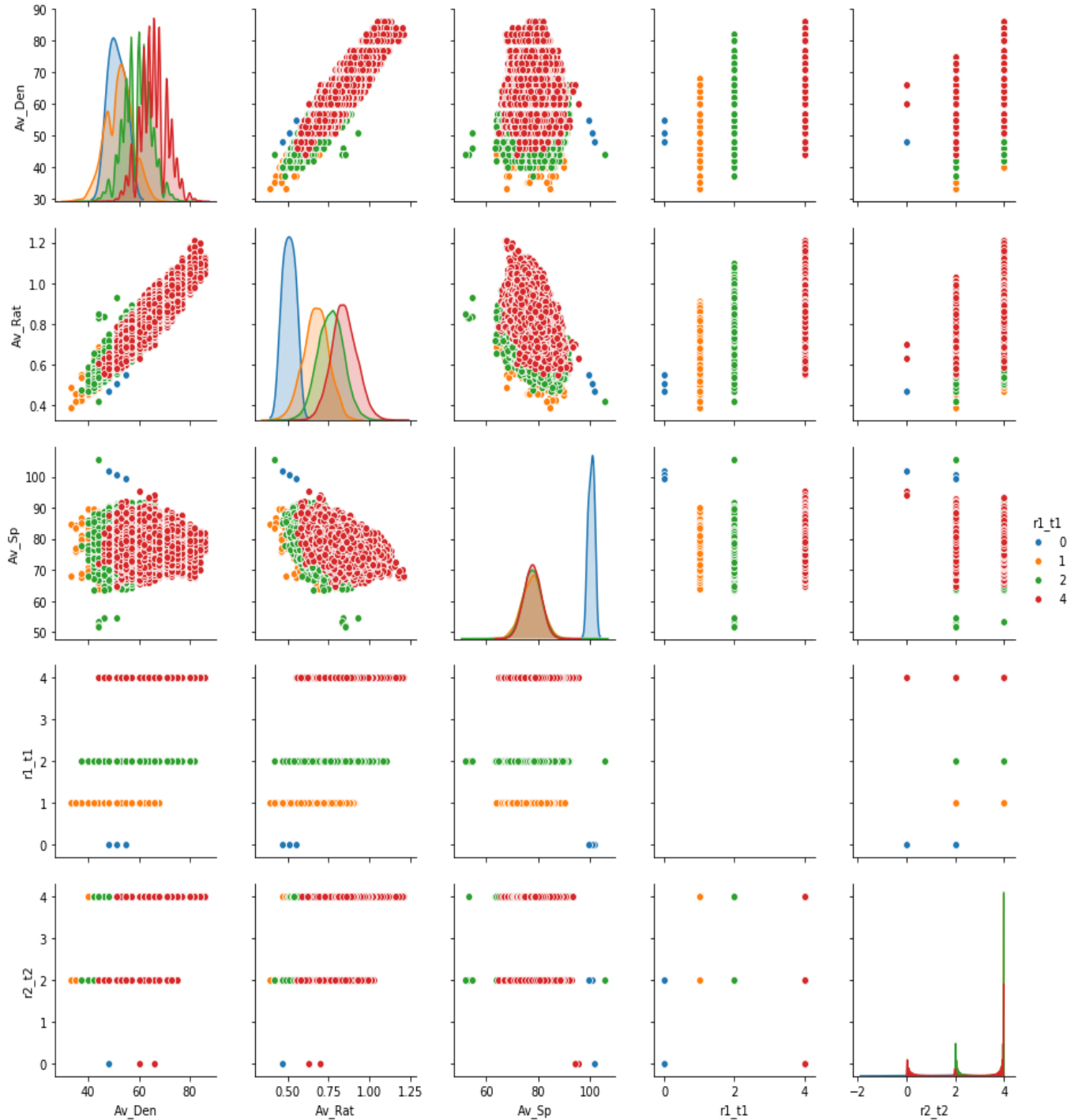


Διάγραμμα 36. Διάγραμμα underfit για τη ράμπα 1

Ομοίως με το overfit δεν παρουσιάζουν απόκλιση προς τα κάτω underfit πέρα από κάποιες τιμές στα αρχικά epochs. Η συγκεκριμένη παρατήρηση είναι εξίσου σημαντική καθώς επιβεβαιώνει την αξιοπιστία των αποτελεσμάτων πρόβλεψης.

Όλα τα παραπάνω αποτελέσματα συνηγορούν στην αξιοπιστία των αποτελεσμάτων του νευρωνικού δικτύου καθώς και τη συνάφεια των δεδομένων εκπαίδευσης που παρασχέθηκαν σε αυτό. Παρακάτω παρουσιάζονται συσχετιστικά διαγράμματα που αφορούν

τα δεδομένα εισόδου του νευρωνικού με το σύνολο των αποτελεσμάτων, δηλαδή των χρόνων κόκκινης ένδειξης του σηματοδότη για τη ράμπα 1. Πιο συγκεκριμένα κάθε μία παράμετρος εισόδου συνδυάζεται με όλες τις υπόλοιπες και εξετάζεται κατά πόσο ο συνδυασμός αυτών ομαδοποιεί και συσχετίζει τις εξόδους του νευρωνικού δικτύου που αναπτύχθηκε.



Διάγραμμα 37. Συσχετιστικά διαγράμματα για τη ράμπα 1

Από τα παραπάνω διαγράμματα γίνεται άμεσα αντιληπτό ότι υπάρχει ευδιάκριτη κατανομή των αποτελεσμάτων. Όπως, φαίνεται και από τον χρωματικό κώδικα που χρησιμοποιήθηκε υπάρχουν διακριτές ομάδες αποτελεσμάτων ανάλογα με τα δεδομένα εισόδου,

πλην ελαχίστων εξαιρέσεων. Ο διαχωρισμός αυτός είναι απόλυτα λογικός και επιθυμητός καθώς υποδηλώνει ότι το δίκτυο έχει εκπαιδευθεί σωστά και είναι σε θέση να παράγει γρήγορα και αξιόπιστα αποτελέσματα.

4.7.2. Ράμπα 2 – Αποτελέσματα

Προκειμένου να εξασφαλιστεί η αξιοπιστία των αποτελεσμάτων για τη ράμπα 2, το νευρωνικό δίκτυο ομοίως με τη ράμπα 1, εκτέλεσε πενήντα αναγνώσεις (epochs) των δεδομένων εκπαίδευσης όπως φαίνεται παρακάτω:

```
Epoch 36/50  
60712/60712 [=====] - 0s 3us/step - loss: 0.0074 - acc: 0.9998 - val_loss: 0.0076 - val_acc:  
0.9999  
Epoch 37/50  
60712/60712 [=====] - 0s 2us/step - loss: 0.0070 - acc: 0.9998 - val_loss: 0.0072 - val_acc:  
0.9999  
Epoch 38/50  
60712/60712 [=====] - 0s 3us/step - loss: 0.0066 - acc: 0.9998 - val_loss: 0.0068 - val_acc:  
0.9999  
Epoch 39/50  
60712/60712 [=====] - 0s 2us/step - loss: 0.0062 - acc: 0.9998 - val_loss: 0.0065 - val_acc:  
0.9999  
Epoch 40/50  
60712/60712 [=====] - 0s 2us/step - loss: 0.0059 - acc: 0.9998 - val_loss: 0.0061 - val_acc:  
0.9999  
Epoch 41/50  
60712/60712 [=====] - 0s 4us/step - loss: 0.0056 - acc: 0.9998 - val_loss: 0.0058 - val_acc:  
0.9999  
Epoch 42/50  
60712/60712 [=====] - 0s 2us/step - loss: 0.0054 - acc: 0.9998 - val_loss: 0.0056 - val_acc:  
0.9999  
Epoch 43/50  
60712/60712 [=====] - 0s 2us/step - loss: 0.0051 - acc: 0.9998 - val_loss: 0.0053 - val_acc:  
0.9999  
Epoch 44/50  
60712/60712 [=====] - 0s 2us/step - loss: 0.0049 - acc: 0.9998 - val_loss: 0.0051 - val_acc:  
0.9999  
Epoch 45/50  
60712/60712 [=====] - 0s 2us/step - loss: 0.0047 - acc: 0.9998 - val_loss: 0.0049 - val_acc:  
0.9999  
Epoch 46/50  
60712/60712 [=====] - 0s 2us/step - loss: 0.0045 - acc: 0.9998 - val_loss: 0.0047 - val_acc:  
0.9999  
Epoch 47/50  
60712/60712 [=====] - 0s 3us/step - loss: 0.0043 - acc: 0.9998 - val_loss: 0.0045 - val_acc:  
0.9999  
Epoch 48/50  
60712/60712 [=====] - 0s 3us/step - loss: 0.0041 - acc: 0.9998 - val_loss: 0.0044 - val_acc:  
0.9999  
Epoch 49/50  
60712/60712 [=====] - 0s 3us/step - loss: 0.0040 - acc: 0.9998 - val_loss: 0.0042 - val_acc:  
0.9999  
Epoch 50/50  
60712/60712 [=====] - 0s 2us/step - loss: 0.0039 - acc: 0.9998 - val_loss: 0.0041 - val_acc:  
0.9999
```

Εικόνα 32. Epochs για τη ράμπα 2

Όπως φαίνεται στο παραπάνω σχήμα σε κάθε επόμενο «πέρασμα» η ακρίβεια και η απώλεια βελτιώνεται. Στη ράμπα 2 συγκριτικά με τη ράμπα 1 παρατηρείται μία σταθερότητα έπειτα από τα αρχικά epochs. Έτσι παραπάνω φαίνεται ότι μετά από 50 epochs το νευρωνικό δίκτυο είχε ακρίβεια της τάξης 99,99% και απώλεια δεδομένων 0,41%. Τα αποτελέσματα για τη ράμπα 2 είναι ακόμα πιο εντυπωσιακά και γίνεται άμεσα αντιληπτό ότι δεν χρειάζεται περαιτέρω εκτέλεση epoch και εμβάθυνση του δικτύου.

Έπειτα λοιπόν από την ολοκλήρωση της εκπαίδευσης και την παραγωγή των προβλέψεων για τους χρόνους της ράμπας 2, η σύγκριση των αποτελεσμάτων με το evaluation test έδωσε τα παρακάτω αποτελέσματα:

108416/108416 [=====] - 4s 33us/step

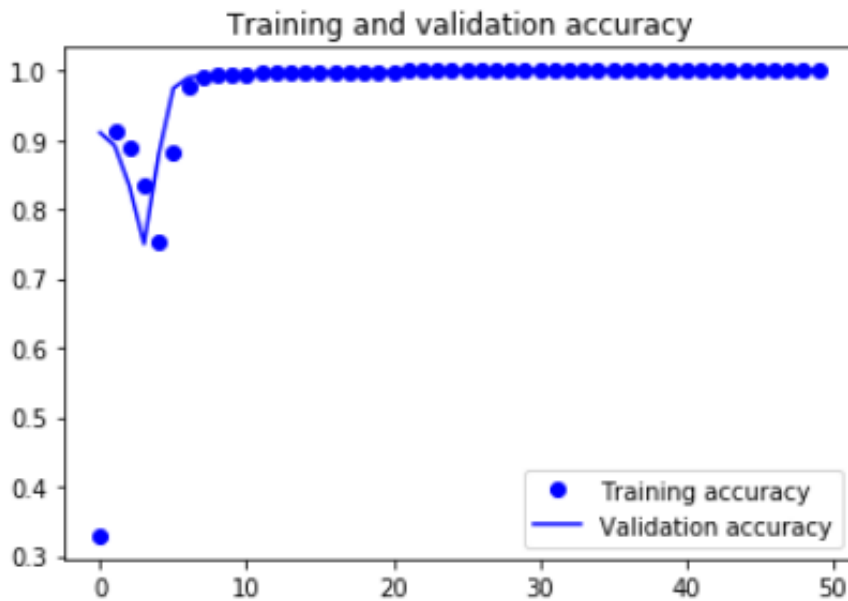
acc: 99.99%

loss: 0.36%

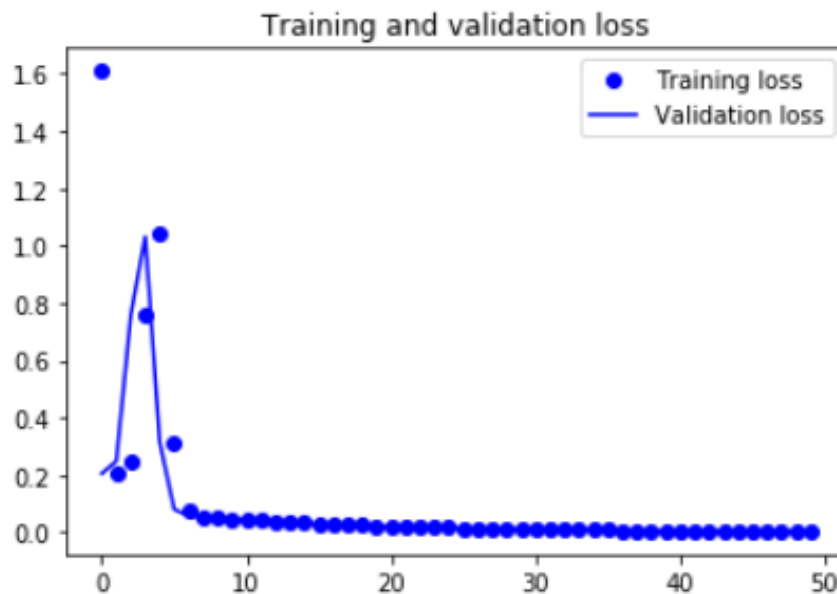
Εικόνα 33. Ακρίβεια και απώλεια δεδομένων αποτελεσμάτων για τη ράμπα 2

Όπως φαίνεται στην παραπάνω εικόνα τα αποτελέσματα για τη ράμπα 2 είναι ιδιαίτερα θετικά καθώς έχουν ακρίβεια **99,99%** και απώλεια μόλις **0,36%**.

Ομοίως με τη ράμπα 1, παρακάτω φαίνονται τα διαγράμματα για το ενδεχόμενο overfit και underfit των χρόνων της ράμπας 2:



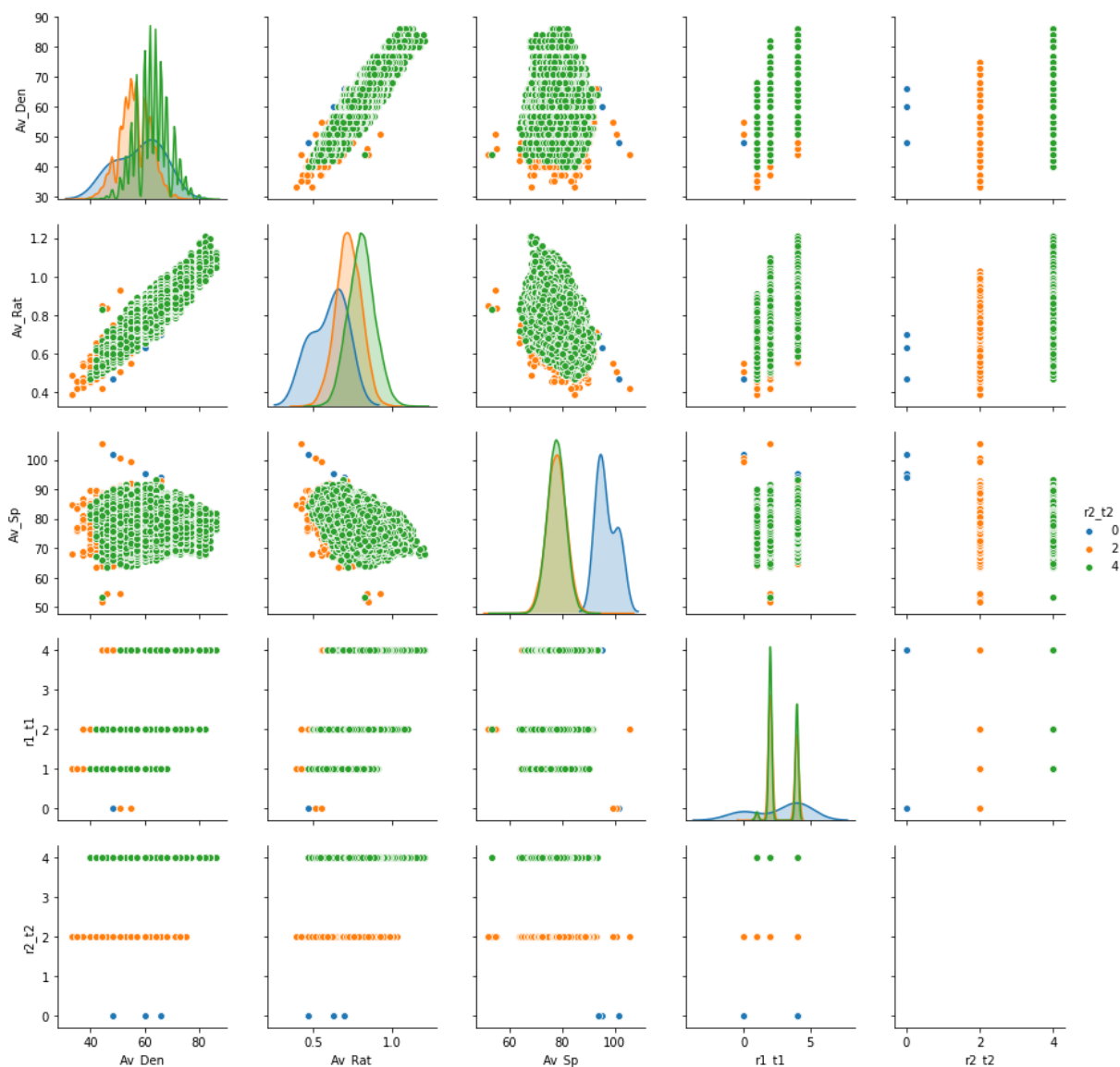
Διάγραμμα 38. Διάγραμμα overfit για τη ράμπα 2



Διάγραμμα 39. Διάγραμμα underfit για τη ράμπα 2

Ομοίως με τη ράμπα 1 το overfit και το underfit δεν παρουσιάζουν απόκλιση πέρα από κάποιες τιμές στα αρχικά epochs. Τα δύο παραπάνω διαγράμματα παρουσιάζουν ακόμα πιο καλά αποτελέσματα σε σχέση με τη ράμπα 1, γεγονός που υποδηλώνει ότι οι προβλέψεις για τη ράμπα 2 είναι ακόμα πιο ακριβείς.

Όλα τα παραπάνω αποτελέσματα συνηγορούν στην αξιοπιστία των αποτελεσμάτων του νευρωνικού δικτύου καθώς και τη συνάφεια των δεδομένων εκπαίδευσης που παρασχέθηκαν σε αυτό όπως συνέβη και με τη ράμπα 1. Παρακάτω όπως και πριν παρουσιάζονται συσχετιστικά διαγράμματα που αφορούν τα δεδομένα εισόδου του νευρωνικού με το σύνολο των αποτελεσμάτων, δηλαδή των χρόνων κόκκινης ένδειξης του σηματοδότη για τη ράμπα 2. Πιο συγκεκριμένα κάθε μία παράμετρος εισόδου συνδυάζεται με όλες τις υπόλοιπες και εξετάζεται κατά πόσο ο συνδυασμός αυτών ομαδοποιεί και συσχετίζει τις εξόδους του νευρωνικού δικτύου που αναπτύχθηκε.



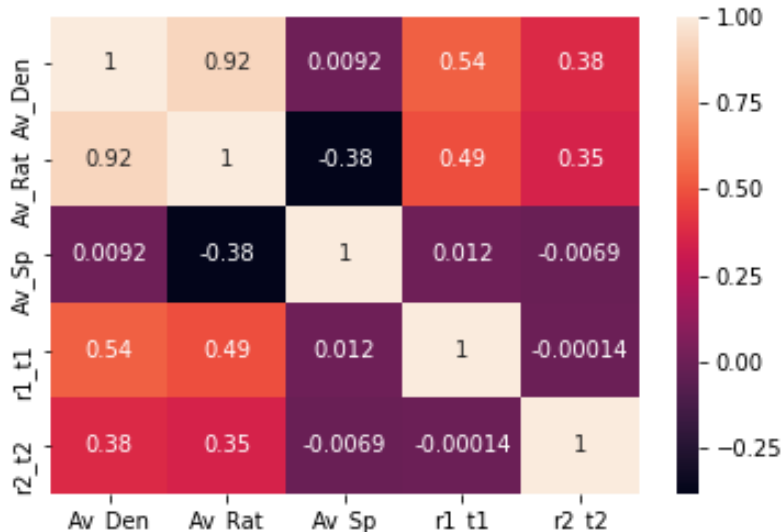
Διάγραμμα 40. Συσχετιστικά διαγράμματα για τη ράμπα 2

Από τα παραπάνω διαγράμματα όπως συνέβη και με τα δεδομένα της πρώτης ράμπας γίνεται άμεσα αντιληπτό ότι υπάρχει ευδιάκριτη κατανομή των αποτελεσμάτων. Όπως, φαίνεται και από τον χρωματικό κώδικα που χρησιμοποιήθηκε υπάρχουν διακριτές ομάδες αποτελεσμάτων ανάλογα με τα δεδομένα εισόδου, πλην ελαχίστων εξαιρέσεων. Ο διαχωρισμός αυτός είναι απόλυτα λογικός και επιθυμητός καθώς υποδηλώνει ότι το δίκτυο έχει εκπαιδευθεί σωστά και είναι σε θέση να παράγει γρήγορα και αξιόπιστα αποτελέσματα χωρίς αστοχίες.

4.7.3. Γενικά αποτελέσματα

Παραπάνω έγινε μία παρουσίαση των αποτελεσμάτων που παράγει το νευρωνικό δίκτυο για κάθε μία ράμπα ξεχωριστά. Από τα αποτελέσματα αυτά γίνεται άμεσα αντιληπτό ότι το νευρωνικό δίκτυο είναι αρκετά αξιόπιστο στους υπολογισμούς τους καθώς η ακρίβεια του κυμαίνεται και στις δύο περιπτώσεις πάνω από 98% και η απώλεια δεδομένων κάτω από 4%. Επιπλέον, έπειτα από την αρχική εκπαίδευση η παραγωγή αποτελεσμάτων γίνεται ταχύτατα μέσα σε λίγα δευτερόλεπτα καθιστώντας έτσι τη χρήση πολύ αποτελεσματική σε πολύπλοκα συστήματα όπως το μελετώμενο τα οποία παρουσιάζουν ταχείες μεταβολές.

Παρακάτω φαίνεται ένα διάγραμμα το οποίο εξετάζει τη συσχέτιση (correlation) μεταξύ των διαφόρων παραμέτρων που έχει το νευρωνικό δίκτυο:



Διάγραμμα 41. Συσχέτιση μεγεθών νευρωνικού δικτύου

Στο παραπάνω διάγραμμα προκύπτει ότι η πυκνότητα έχει τεράστια συσχέτιση με τον λόγο R. Το αποτέλεσμα αυτό επιβεβαιώνει και τα αποτελέσματα όλων των προηγούμενων αλγορίθμων καθώς όπως παρατηρήθηκε το R ακολουθούσε την πυκνότητα. Αντίστοιχα, βλέπουμε ότι το R με τη μέση ταχύτητα έχουν αρνητική συσχέτιση γεγονός απόλυτα φυσιολογικό καθώς αυτά τα δύο μεγέθη είναι αντιστρόφως ανάλογα. Επιπλέον, όπως ήταν αναμενόμενο οι χρόνοι για τη

ράμπα 1 και 2 παρουσιάζουν μεγάλη συσχέτιση με την πυκνότητα καθώς αυτή είναι που τους καθορίζει.

Το πόσο ισχυρή ή όχι είναι η συσχέτιση δύο μεγεθών από το παραπάνω διάγραμμα γίνεται κατανοητό από τον χρωματικό κώδικα και από την τιμή που έχει κάθε κελί. Όσο πιο κοντά στη μονάδα είναι τόσο πιο ισχυρή είναι η σχέση των μεγεθών αυτών. Με βάση λοιπόν αυτό προκύπτουν τα συμπεράσματα που αναφέρθηκαν προηγουμένως.

ΚΕΦΑΛΑΙΟ 5. ΓΕΝΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ **ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ**

5.1. Σύγκριση Αλγορίθμων και Συμπεράσματα

Στο προηγούμενο κεφάλαιο παρουσιάστηκαν αναλυτικά τα αποτελέσματα κάθε αλγορίθμου που αναπτύχθηκε στην παρούσα εργασία και περιγράφηκε στο Κεφάλαιο 3. Η αρχική πιλοτική υλοποίηση είναι αρκετά απλή και αποτέλεσε τον πυλώνα για όλους τους αλγορίθμους από εκεί και πέρα. Κάθε νέος αλγόριθμος είχε αυξημένη πολυπλοκότητα σε σχέση με τους προηγούμενους με αποτέλεσμα ο αλγόριθμος με τρεις (3) λωρίδες και δύο (2) ράμπες, ο οποίος προσομοίωνε ένα άκρως ρεαλιστικό σενάριο λειτουργίας, να απαιτεί υψηλή υπολογιστική ισχύ. Ενδεικτικό της πολυπλοκότητάς του είναι ότι σε πέντε (5) μόλις λεπτά λειτουργίας του παρήχθησαν αποτελέσματα της τάξεως των 150 MB. Έτσι, με βάση αυτό και δεδομένου ότι σκοπός της εργασίας από την αρχή ήταν να δημιουργηθεί ένας αλγόριθμος μηχανικής μάθησης με νευρωνικό δίκτυο ο τελευταίος αλγόριθμος που εξετάστηκε υλοποιεί ένα νευρωνικό δίκτυο το οποίο εκπαιδεύτηκε με το πλήθος των δεδομένων που παρήχθησαν από το σενάριο υλοποίησης των τριών (3) λωρίδων και δύο (2) ραμπών.

Ξεκινώντας από τον πρώτο αλγόριθμο παρατηρείται ότι ο χρόνος εκτέλεσης με τις δεδομένες συνθήκες που περιγράφηκαν είναι πολύ μικρός της τάξεως των δύο (2) δευτερολέπτων και ο αριθμός των επαναλήψεων περί της 3000. Στον επόμενο αλγόριθμο που εξετάστηκε ο οποίος είχε αυξημένες συνθήκες ελέγχου ο χρόνος εκτέλεσης διπλασιάστηκε και ξεπέρασε τα τέσσερα (4) δευτερόλεπτα ενώ αυξήθηκε αισθητά και ο αριθμός των επαναλήψεων και ξεπέρασε κατά πολύ τις 4000. Ο τρίτος αλγόριθμος ο οποίος είχε μεγάλη ομοιότητα με τον δεύτερο στην υλοποίηση παρουσίασε και μεγάλη ομοιότητα στα αποτελέσματα τα οποία βελτιώθηκαν ελαφρώς. Στην τέταρτη υλοποίηση πραγματοποιήθηκε μία ιδιαίτερα σημαντική προσθήκη η οποία ήταν αναγκαία έτσι ώστε το μελετώμενο σύστημα να είναι κατά το δυνατόν πιο κοντά σε πραγματικές συνθήκες. Η προσθήκη αυτή αφορούσε στην ύπαρξη τριών (3) λωρίδων αντί για μία γεγονός που αύξησε ιδιαίτερα την πολυπλοκότητα του αλγορίθμου όπως φαίνεται και από τον χρόνο εκτέλεσης που τριπλασιάστηκε και ξεπέρασε τα δώδεκα (12) δευτερόλεπτα. Το αμέσως επόμενο βήμα ήταν η προσθήκη και δεύτερης ράμπας εισόδου στο μελετώμενο τμήμα του αυτοκινητόδρομου. Ο αλγόριθμος με τρεις (3) λωρίδες και δύο (2) ράμπες είχε υπερβολικά αυξημένη πολυπλοκότητα και για τον λόγο αυτό η συνθήκη τερματισμού ήταν ο χρόνος, δηλαδή μετά το πέρας πέντε (5) λεπτών, και όχι ο αριθμός των αυτοκινήτων που θα εισέλθουν. Ενδεικτικό του συγκεκριμένου αλγορίθμου είναι ότι βελτίωσε αισθητά την απόδοση όσον αφορά τη διατήρηση της πυκνότητας περί το 70% και επιπλέον ότι μέσα σε μόλις πέντε (5) λεπτά παρήγαγε ένα αρχείο αποτελεσμάτων με πάνω από 108000 εγγραφές και μέγεθος περί τα 150 MB.

Έτσι με δεδομένο ότι όσο πιο κοντά σε πραγματικές συνθήκες ήταν το σενάριο υλοποίησης τόσο πιο απαιτητικοί ήταν αλγόριθμοι καθώς απαιτούσαν όλο και περισσότερη ισχύ και χρόνο εκτέλεσης τόσο πιο εμφανής ήταν η ανάγκη για την ανάπτυξη ενός αλγορίθμου μηχανικής μάθησης με χρήση νευρωνικού δικτύου. Το νευρωνικό δίκτυο που αναπτύχθηκε χρησιμοποίησε ως δεδομένα εκπαίδευσης τα δεδομένα που παρήχθησαν από το τελευταίο σενάριο υλοποίησης. Το νευρωνικό δίκτυο εκπαιδεύτηκε έτσι ώστε να παράγει παρά πολύ γρήγορα αποτελέσματα και συγκεκριμένα τους χρόνους κόκκινης ένδειξης του φωτεινού σηματοδότη κάθε ράμπας. Έτσι έπειτα από την αρχική εκπαίδευση παρατηρήθηκε ότι σε ταχύτατο χρόνο το δίκτυο έδινε αποτέλεσμα με ακρίβεια πάνω από 95% και απώλειες δεδομένων κάτω του 5%. Τα αποτελέσματα αυτά έχουν αναφερθεί αναλυτικά στο Κεφάλαιο 4.

Με βάση λοιπόν όλα τα παραπάνω συνεπάγεται το συμπέρασμα ότι είναι απαραίτητη η χρήση αλγορίθμων μηχανικής μάθησης για τέτοιου είδους προβλήματα καθώς πλέον η διαχείριση της κυκλοφορίας οχημάτων εξαρτάται από πληθώρα παραμέτρων και δεδομένων τα οποία οι παραδοσιακοί αλγόριθμοι αδυνατούν να αναλύσουν και να ελέγξουν με ταχύτητα και ακρίβεια. Επιπλέον σε έναν οδικό άξονα τα δεδομένα αλλάζουν με τέτοια ταχύτητα που απαιτείται αντίστοιχη απόκριση και από το υπολογιστικό σύστημα έτσι ώστε να είναι αξιόπιστο το αποτέλεσμα. Ταχύτατες αποκρίσεις σε τόσο δυναμικές συνθήκες μπορούν να παράξουν μόνο αλγόριθμοι τεχνητής νοημοσύνης οι οποίοι είναι εκπαιδευμένοι κατάλληλα στο να λαμβάνουν τέτοιου είδους αποφάσεις.

Τέλος, μέσω της παρούσας εργασίας έγινε μία εκτεταμένη μελέτη και ανάπτυξη αλγορίθμων υλοποίησης του Ramp Metering και παράλληλα παρουσιάστηκαν αναλυτικά αποτελέσματα για κάθε μία περίπτωση ξεχωριστά δίνοντας έτσι μία εικόνα για το πως ένα τέτοιο πρόβλημα μπορεί να εξελιχθεί από ένα σύστημα απλού αυτομάτου ελέγχου σε ένα σύνθετο και πολύπλοκο σύστημα που απαιτεί τεχνητή νοημοσύνη και μηχανική μάθηση.

5.2. Μελλοντικές Επεκτάσεις

Ο αλγόριθμος μηχανικής μάθησης με χρήση νευρωνικού δικτύου που παρουσιάστηκε στο Κεφάλαιο 3 αποτελεί μία σύγχρονη υλοποίηση η οποία πραγματοποιήθηκε με ευρέως χρησιμοποιούμενα εργαλεία όπως η γλώσσα Python και η βιβλιοθήκη Keras. Ένας τέτοιος αλγόριθμος και με δεδομένο την πολύ καλή ποιότητα αποτελεσμάτων που παράγει θα μπορούσε να αποτελέσει πηγή ερευνητικής δραστηριότητα καθώς μπορεί να δεχθεί αρκετές επεκτάσεις. Πιο συγκεκριμένα μερικές από τις προσθήκες/επεκτάσεις που προτείνονται ως αντικείμενο μελλοντικής έρευνας είναι:

- Αύξηση του αριθμού των ραμπών κατά μήκος ενός αυτοκινητόδρομου

- Εφαρμογή σε όλο το εύρος ενός οδικού άξονα και συγκέντρωση δεδομένων από όλα τα σημεία αυτού για ακόμα καλύτερη διαχείριση της κυκλοφορίας από ένα και μόνο κεντρικό σημείο.
- Εκπαίδευση του νευρωνικού δικτύου με πραγματικά δεδομένα από οδικές αρτηρίες
- Αύξηση του βάθους του νευρωνικού δικτύου με περισσότερους ενδιάμεσους κόμβους (hidden layers) για την αύξηση της ακρίβειας των αποτελεσμάτων και τη μείωση της απώλειας δεδομένων
- Επέκταση των συνθηκών ελέγχου για την είσοδο σε αυτοκινητόδρομοι και εκπαίδευση εκ νέου του νευρωνικού δικτύου

5.3. Επίλογος

Στόχος της συγκεκριμένης εργασίας ήταν να μελετηθούν συστήματα διαχείρισης κυκλοφοριακών δεδομένων και να αναπτυχθούν αλγόριθμοι ελέγχου εισόδου με τη μέθοδο Ramp Metering. Με την ολοκλήρωση λοιπόν της παρούσας εργασίας, παραδίδεται ένα σύνολο αλγορίθμων Ramp Metering καθώς και ένας αλγόριθμος μηχανικής μάθησης σε γλώσσα Python. Επιπλέον, ο αλγόριθμος μηχανικής μάθησης που υλοποιήθηκε μπορεί να αποτελέσει εφαλτήριο για μελλοντική έρευνα που αφορά σε μοντέλα διαχείρισης κυκλοφορίας οχημάτων σε αυτοκινητόδρομους με μεθόδους τεχνητής νοημοσύνης και μηχανικής μάθησης.

Κατά την εκπόνηση της παρούσας εργασίας, αποκτήθηκαν γνώσεις και εμπειρία στην ανάπτυξη αλγορίθμων σε γλώσσα Python που αποτελεί μια από τις πιο διαδεδομένες και σύγχρονες γλώσσες προγραμματισμού. Επιπροσθέτως, έγινε μελέτη στο πρόβλημα διαχείρισης κυκλοφοριακών δεδομένων με χρήση τεχνητής νοημοσύνης, το οποίο αποτελεί αντικείμενο ενεργούς έρευνας της επιστημονικής κοινότητας και ένα από τα πιο διαδεδομένα προβλήματα προς επίλυση των σύγχρονων αστικών κέντρων. Καταληκτικά, αξιοσημείωτη είναι η εξέλιξη του συγκεκριμένου κλάδου η οποία απαιτεί συνεχή μελέτη και πειραματισμό με νέες τεχνολογίες έτσι ώστε να είναι σε θέση ένας μηχανικός να συμβαδίσει με τις αυξημένες απαιτήσεις στον συγκεκριμένο τομέα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Alberto Broggi, Alexander Zelinsky, Michel Parent, Charles E. Thorpe, “Intelligent Vehicles” from book “Springer Handbook of Robotics” (σελ.1175-1198)
2. S. Cheon, “An overview of automated highway systems and the social institutional challenges they face”. Technical Report UCTC No. 624, University of California, Berkeley, California, 2002
3. A. Hsu, F. Eskafi, S. Sachs, and P. Varaiya. “Design of platoon maneuver protocols for IVHS”, University of California, Berkeley, California, 1991
4. C. F. Daganzo, “Fundamentals of Transportation and Traffic Operations”, Pergamon Press, 1997
5. P. Kachroo and K. Özbay, “Feedback Control Theory for Dynamic Traffic Assignment. Advances in Industrial Control”, Springer-Verlag, Berlin, 1999
6. M. Papageorgiou, “Applications of Automatic Control Concepts to Traffic Flow Modeling and Control”, Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, Germany, 1983
7. A. Kotsialos and M. Papageorgiou, “Freeway ramp metering: An overview”, In Proceedings of the 3rd Annual IEEE Conference on Intelligent Transportation Systems (ITSC 2000), Dearborn, Michigan, October 2000
8. S. A. Nsour, S. L. Cohen, J. E. Clark, and A. J. Santiago, “Investigation of the impacts of ramp metering on traffic flow with and without diversion”, Transportation Research Record, 1992
9. F. L. Hall and K. Agyemang-Duah, “Freeway capacity drop and the definition of capacity”, Transportation Research Record, 1991
10. A. D. May, “Traffic Flow Fundamentals”, Prentice-Hall, Englewood Cliffs, New Jersey, 1990
11. C. G. Cassandras, D. L. Pepyne, and Y. Wardi, “Optimal control of systems with timedriven and event-driven dynamics”, In Proceedings of the 37th IEEE Conference on Decision and Control, Tampa, Florida, December 1998
12. H. J. Sussmann and J. C. Willems, “300 years of optimal control: From the brachistochrone to the maximum principle”, IEEE Control Systems Magazine, June 1997
13. J. M. Maciejowski, “Predictive Control with Constraints”, Prentice-Hall, Harlow, England, 2002
14. E. F. Camacho and C. Bordons, “Model Predictive Control in the Process Industry”, Springer-Verlag, Berlin, Germany, 1995
15. A. Kotsialos and M. Papageorgiou, “Efficiency and equity properties of freeway network-wide ramp metering with AMOC”, Transportation Research Part C, 12(6), 2004
16. A. Kotsialos, M. Papageorgiou, M. Mangeas, and H. Haj-Salem, “Coordinated and integrated control of motorway networks via non-linear optimal control”, Transportation Research Part C, 2002
17. S. H. Chen, A. J. Jakeman, and J. P. Norton “Artificial intelligence techniques: An introduction to their use for modelling environmental systems”, Mathematics and Computers in Simulation, 78(2):379–400, July 2008
18. G. Weiss, editor, “Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence”, MIT Press, Cambridge, Massachusetts, 1999
19. H. T. Nguyen and E. A. Walker, “A First Course in Fuzzy Logic”, Chapman & Hall Press, Boca Raton, Florida, 2nd edition, 1999

20. D. Hammerstrom “Working with neural networks”, IEEE Spectrum, 30(7):46–53, July 1993
21. X. Yao, “Evolving artificial neural networks”, Proceedings of the IEEE, 87(9):1423–1447, 1999
22. K. J. Åström and B. Wittenmark, “Computer-Controlled Systems — Theory and Applications”, Prentice-Hall, Upper Saddle River, New Jersey, 3rd edition, 1997
23. M. Papageorgiou, H. Hadj-Salem, and F. Middelham, “ALINEA local ramp metering: Summary of field results”, In Proceedings of the 76th Annual Meeting of the Transportation Research Board, Washington, DC, 1997. Paper no. 970032
24. Y. J. Stephanedes, E. Kwon, and K. Chang, “Control emulation method for evaluating and improving traffic-responsive ramp metering strategies”, Transportation Research Record, , 1992
25. K. J. Åstrom, P. Albertos, M. Blanke, A. Isidori, W. Schaufelberger, and R. Sanz, “Control of Complex Systems”, Springer-Verlag, New York, 2000.
26. D. D. Šiljak, “Decentralized Control of Complex Systems”, Academic Press, San Diego, California, 1991
27. A. V. Fursikov, “Optimal Control of Distributed Systems: Theory and Applications”, American Mathematical Society, Boston, Massachusetts, 2000
28. U.S. Department of Transportation, Federal Highway Administration, Office of Operations, “Ramp Metering: A Proven, Cost-Effective Operational Strategy –A Primer”, October 2014
29. Thorsten Schmidt-Dumont, Jan H van Vuuren, “Decentralised reinforcement learning for ramp metering and variable speed limits on highways”, IEEE Transactions On Intelligent Transportation Systems, vol. 14, No. 8, August 2015
30. Champion, Camilla & Champion, Cody, “Agent-Based Implementation of Particle Hopping Traffic Model With Stochastic and Queuing Elements,” 2018
31. H. Michael Zhang, Stephen G. Ritchie, “Freeway ramp metering using artificial neural networks”, Transportation Research Part C: Emerging Technologies, Volume 5, Issue 5, σελ 273-286, 1997
32. M. Papageorgiou, H. Hadj-Salem, and J-M. Blosseville, “ALINEA: A local feedback control law for on-ramp metering”, Transportation Research Record, vol. 1320, σελ 58–64, 1991
33. Python, Ημερομηνία επίσκεψης 29 Απριλίου 2019, <https://el.wikipedia.org/wiki/Python>
34. Νευρωνικό Δίκτυο, Ημερομηνία επίσκεψης 27 Απριλίου 2019, https://el.wikipedia.org/wiki/%CE%9D%CE%B5%CF%85%CF%81%CF%89%CE%BD%CE%B9%CE%BA%CF%8C_%CE%B4%CE%AF%CE%BA%CF%84%CF%85%CE%BF
35. Ι.Μ. Φραντζεσκάκης, Μ.Χ. Πιτσιάβα – Λατινοπούλου, Δ.Α. Τσαμπούλας, “Διαχείριση Κυκλοφορίας” , Αθήνα, 1997

ΠΑΡΑΡΤΗΜΑ Α

Δομή Python και Παραδείγματα

Η γλώσσα χρησιμοποιεί μεταγλωττιστή (compiler) για τη δημιουργία του εκτελέσιμου κώδικα και σχετίζεται με τις γλώσσες προγραμματισμού Tcl, Perl, Scheme, Java και Ruby, καθώς και με την ABC η οποία υπήρξε η αρχική πηγή έμπνευσης για τη δημιουργία της.

Ονοματοδοσία μεταβλητών

Τα ονόματα που λαμβάνουν οι μεταβλητές υπόκεινται στους εξής κανόνες:^[33]

- Είναι λέξεις που αποτελούνται από αλφαριθμητικούς χαρακτήρες (γράμματα και αριθμούς) και την κάτω παύλα (underscore, "_"). Μπορεί να αρχίζει με αλφαβητικό χαρακτήρα ή κάτω παύλα (_).
- Δεν επιτρέπεται να αρχίζει με αριθμό (πχ. var4_1 είναι σωστό, αλλά το 4var_1 είναι λάθος).
- Υπάρχει διάκριση πεζών και κεφαλαίων γραμμάτων (case sensitivity), οπότε και οι μεταβλητές διαφοροποιούνται λόγω πεζών ή κεφαλαίων (πχ. άλλη η μεταβλητή Var και άλλη η var).
- Δεν επιτρέπονται δεσμευμένες λέξεις (λέξεις-κλειδιά, keywords), δηλαδή λέξεις με ιδιαίτερη σημασία (identifiers), όπως print, type, False, float, None, κλπ.

Τύποι δεδομένων

Οι τύποι δεδομένων της Python διακρίνονται στους αρχέγονους (primitive data type), στις πιο σύνθετες δομές δεδομένων (λίστες, πλειάδες, σύνολα, λεξικά) και, λόγω των αντικειμενοστραφών δυνατοτήτων της, στις κλάσεις (classes). Λόγω του ότι η Python είναι δυναμική γλώσσα δεν δηλώνονται οι τύποι των μεταβλητών πριν χρησιμοποιηθούν. Ο τύπος που τους αποδίδεται εξαρτάται από την τιμή που θα πάρουν κατά τη διάρκεια της εκτέλεσης του προγράμματος. Όταν χρησιμοποιηθεί (πχ. σε έκφραση ή συνάρτηση) πρέπει να έχει πάρει τιμή γιατί ο διερμηνευτής (interpreter) δίνει λάθος που το ονομάζει NameError. Γι' αυτόν τον λόγο η Python υποστηρίζει τον ιδιαίτερο τύπο NoneType.^[33]

Οι αρχέγονοι τύποι είναι:^[33]

- int (ακέραιος). Στην Python οι ακέραιοι δεν έχουν περιορισμό στο μέγεθός τους και μπορεί να είσαι όσο μεγάλοι χρειάζεται
- float (κινητής υποδιαστολής)
- str (συμβολοσειρά ή string). Είναι ακολουθία χαρακτήρων που περικλείονται από μονά ή διπλά εισαγωγικά. Σε ειδικές περιπτώσεις μπορεί να γίνει χρήση τριπλών εισαγωγικών (μονών :'" ή διπλών: """).
- bool. Ο λογικός τύπος (Boolean) μεταβλητής με τιμές True ή False.

- **NoneType** . Αυτός ο τύπος μεταβλητής παίρνει την τιμή None με ανάθεση (assignment, = None) και αντιμετωπίζεται, στον σχεδιασμό του προγράμματος, σαν μεταβλητή που δεν έχει τιμή. None παράγουν και οι συναρτήσεις που δεν επιστρέφουν τιμή (δεν έχουν return <έκφραση>), όπως για παράδειγμα η print().

Δεν υπάρχει μεταβλητή τύπου Character (όπως για παράδειγμα στη 7η C/C++). Σε αυτή την περίπτωση χρησιμοποιείται η μεταβλητή String με ένα χαρακτήρα. Η συνάρτηση type() μας επιστρέφει τον τύπο της μεταβλητής. Αν η μεταβλητή δεν έχει πάρει τιμή μας επιστρέφει NoneType.

Ενσωματωμένες δομές δεδομένων

Οι ενσωματωμένες δομές δεδομένων στην Python είναι η Λίστα (List), η Πλειάδα (Tuple), το Σύνολο (Set) και το Λεξικό (Dictionary)^[33].

- **Λίστα (list)**

Η λίστα υλοποιεί τον γνωστό αφηρημένο τύπο δεδομένων. Μοιάζει με τη δομή του πίνακα (array), αλλά σε αντίθεση με αυτόν μπορεί να περιέχει ανομοιογενή αντικείμενα (items) ή αλλιώς στοιχεία (elements), τα οποία μπορεί να είναι οποιαδήποτε δομή δεδομένων (ακόμα και λίστα). Όπως και στους πίνακες τα στοιχεία είναι διατεταγμένα σε συγκεκριμένη σειρά, προσπελάσιμα με δείκτες και επιτρέπεται η επανάληψη τους. Ορίζεται με ένα ζευγάρι τετραγωνισμένων αγκύλων [] και με τα στοιχεία να χωρίζονται μεταξύ τους με κόμμα, πχ. η ακολουθία [<στοιχείο1>, ..., <στοιχείοN>] ορίζει μια λίστα N στοιχείων. Μπορεί να οριστεί μια κενή λίστα μόνο με τις αγκύλες και χωρίς περιεχόμενο. Επειδή η συμβολοσειρά (string) αποτελεί μια διατεταγμένη ακολουθία χαρακτήρων μοιάζει με λίστα και χρησιμοποιεί παρόμοιους τελεστές (operators) και συναρτήσεις (methods). Η μεγάλη διαφορά τους είναι ότι η λίστα είναι δυναμικός ή αλλιώς μεταβλητός (mutable) τύπος δεδομένων, ενώ η συμβολοσειρά στατικός ή αλλιώς αμετάβλητος (immutable) τύπος δεδομένων. Στη λίστα μπορούμε να αλλάξουμε τα στοιχεία που περιέχει. Στη συμβολοσειρά η αλλαγή των χαρακτήρων που περιέχει γίνεται μόνο με αντικατάσταση (overwriting) μέσω ανάθεσης (assignment) από νέα συμβολοσειρά.

- **Πλειάδα (tuple)**

Η πλειάδα υλοποιεί την αντίστοιχη μαθηματική έννοια της πλειάδας. Είναι μια πεπερασμένη ακολουθία στοιχείων διατεταγμένων σε συγκεκριμένη σειρά και μοιάζει πολύ με τη λίστα (list). Ορίζεται με ένα ζευγάρι παρενθέσεων () και με τα στοιχεία να χωρίζονται μεταξύ τους με κόμμα, πχ. η ακολουθία (<στοιχείο1>, ..., <στοιχείοN>) ορίζει μια πλειάδα N στοιχείων. Αν το επιτρέπουν τα συμφραζόμενα μπορούν να παραληφθούν οι παρενθέσεις. Διαφέρει από την δομή της λίστας στο ότι η πλειάδα είναι στατική ή αλλιώς αμετάβλητη (immutable) δομή δεδομένων. Αφού δημιουργηθεί

δεν μπορεί να μεταβληθεί. Η χρήση της πλειάδας είναι προτιμότερη από της λίστας όταν δεν χρειάζεται να αλλάξουμε τα στοιχεία της γιατί θεωρητικά είναι αποτελεσματικότερη σε ταχύτητα στις διάφορες λειτουργίες της. Η πλειάδα που περιέχει ένα στοιχείο ορίζεται πάντα με ένα κόμμα μετά το στοιχείο, πχ. (<στοιχείο>,), διότι οι παρενθέσεις χρησιμοποιούνται και στην ομαδοποίηση των εκφράσεων (expressions) και η γραφή χωρίς κόμμα, πχ. = (<στοιχείο>) λαμβάνεται από τον διερμηνευτή (interpreter) ως ανάθεση (assignment), δηλαδή = <στοιχείο>.

- **Λεξικό (dictionary)**

Το λεξικό (dictionary) είναι μια δομή δεδομένων για την αποθήκευση ζευγαριών τιμών της μορφής κλειδί: τιμή (key: value). Ορίζεται με ένα ζευγάρι αγκύλων {} και με τα ζευγάρια να χωρίζονται μεταξύ τους με κόμμα, πχ. η ακολουθία { κλειδί1 : τιμή1, ..., κλειδίN : τιμήN } ορίζει ένα λεξικό N ζευγαριών. Ανάμεσα στο κλειδί και την τιμή υπάρχει η άνω-κάτω τελεία (:). Τα κλειδιά πρέπει να είναι μοναδικά γιατί χρησιμεύουν στην προσπέλαση της αντίστοιχης τιμής. Τα κλειδιά πρέπει να είναι στατικοί ή αλλιώς αμετάβλητοι (immutable) τύποι δεδομένων, όπως ακέραιοι αριθμοί, συμβολοσειρές, αμετάβλητες πλειάδες (όταν όλα τα στοιχεία της πλειάδας είναι αμετάβλητα).

Ανάθεση τιμών (assignment)

Επειδή η Python είναι δυναμική γλώσσα προγραμματισμού (dynamically typed) διαφέρει από τις άλλες παραδοσιακές γλώσσες προστακτικού προγραμματισμού (Imperative programming) στον τρόπο που γίνεται η ανάθεση τιμών στις μεταβλητές της. Συνήθως οι μεταβλητές δηλώνονται πριν τους ανατεθούν τιμές και μαζί δηλώνεται και ο τύπος τους (integer, floating point, character, κ.λπ.). Στην Python οι μεταβλητές δηλώνονται μαζί με την ανάθεση των τιμών σε αυτές (ακόμα και όταν τους δοθεί η τιμή None) και δεν είναι δεσμευμένες σε κάποιο συγκεκριμένο τύπο δεδομένων. Μία μεταβλητή μπορεί να είναι ακεραία (integer) και η ίδια στη συνέχεια να γίνει συμβολοσειρά (string) αν της ανατεθεί τιμή συμβολοσειράς.

Στην Python, όταν κατά την εκτέλεση ενός προγράμματος, αναθέτουμε τιμή σε μια μεταβλητή x, πχ. x = 2, η τιμή 2 αντιγράφεται σε χώρο μνήμης που δεσμεύεται (memory allocation), μαζί με άλλες ιδιότητες, όπως πχ. ότι ο τύπος της είναι ακέραιος (integer). Στη μεταβλητή x αποδίδεται η διεύθυνση (memory address) αυτής της θέσης στην μνήμη. Η μεταβλητή λειτουργεί σαν δείκτης (pointer). Η ίδια η μεταβλητή δεν έχει κάποια τιμή ή τύπο δεδομένων, αλλά έχει τη θέση της μνήμης που «αντιπροσωπεύει». Σε ένα πρόγραμμα μετά την ανάθεση x = 2 (η μεταβλητή είναι τύπου integer), στη συνέχεια της εκτέλεσης μπορεί να εκτελεστεί η ανάθεση x = "hellow world", οπότε η μεταβλητή «αντιπροσωπεύει» τη συμβολοσειρά "hellow world" (είναι τύπου string). Στη δεύτερη ανάθεση η τιμή 2 δεν «αντιπροσωπεύεται» από κάποια μεταβλητή (δεν χρειάζεται πλέον στο πρόγραμμα) και ο

συλλέκτης απορριμμάτων (garbage collector), της Python, αναλαμβάνει «αυτόματα» να αποδεσμεύσει (memory de-allocation) τον χώρο της μνήμης που αυτή κατέχει.^[33]

Εσοχές (indentation)

Η στοίχιση στα αριστερά ή αλλιώς οι αριστερές εσοχές (indentation) έχουν μεγάλη σημασία γιατί ορίζουν τα block (ομαδοποίηση) του κώδικα. Αυτό σε άλλες γλώσσες, όπως C και Java επιτυγχάνεται με τη χρήση των αγκύλων { }. Στην Python το block κώδικα αρχίζει εκεί που η αριστερή εσοχή αυξάνεται προς τα δεξιά και τελειώνει εκεί που καταργείται η εσοχή. Εσοχή μέσα σε εσοχή σημαίνει block μέσα σε block (nested blocks). Η εσοχή μπορεί να είναι οσοδήποτε μεγάλη ή μικρή, αλλά συνιστάται η χρήση τεσσάρων (4) κενών χαρακτήρων για ευκολότερη αναγνωσιμότητα του κώδικα.^[33]

Για παράδειγμα μια εμφωλιασμένη εντολή if ορίζεται ως εξής:

```
if <λογική έκφραση1> :
    εντολή1
    εντολή2
if <λογική έκφραση2> :
    εντολή3
    εντολή4
    εντολή5
εντολή6
```

Το block της πρώτης if περιέχει τις εντολή1, εντολή2, τη δεύτερη if και τελειώνει με την εντολή5. Η εντολή6 είναι εκτός block (δεν ανήκει στην εξωτερική if). Η εσωτερική if (εμφωλιασμένη) με τη δική της εσοχή ορίζει ένα block μέσα στο block της εξωτερικής if.

Τελεστές

- Τελεστές πράξεων

Υποστηρίζει τους γνωστούς, με τις άλλες γλώσσες προγραμματισμού, τελεστές (operators), όπως τους +, -, *, /, %, που όμως έχουν κάποιες διαφοροποιήσεις ανάλογα με τους τελεστέους (operands), δηλαδή τον τύπο των μεταβλητών, όπου εφαρμόζονται.

Ο τελεστής της πρόσθεσης + όταν οι τελεστέοι είναι Strings εκτελεί την πράξη της συνένωσης (concatenation) και ο πολλαπλασιασμός * την πράξη της αναπαραγωγής (replication) του String όταν αυτό πολλαπλασιάζεται με ακέραιο (integer). Δηλαδή η έκφραση (expression): "abc" + "def" θα δώσει "abcdef" και η "abc" * 3 ή 3 * "abc" θα δώσει "abcabcabc" αντίστοιχα.^[33]

- Τελεστές λογικής

Υποστηρίζει τους λογικούς τελεστές and, or και not για μεταβλητές τύπου Boolean. Τα εμφωλιασμένα (nested) not αποδίδονται ως διπλή άρνηση (άρνηση στην άρνηση). Η σειρά (προτεραιότητα) εκτέλεσης είναι not, and, or.^[32]

Εμβέλεια ονομάτων οντοτήτων (scope)

Στις οντότητες (μεταβλητές, συναρτήσεις, μέθοδοι, κ.λπ.), που δημιουργούνται σε ένα πρόγραμμα, για να μπορούν να χρησιμοποιηθούν (προσπελαστούν), αποδίδονται ονόματα (name binding). Για παράδειγμα στην Python μια οντότητα, όπως είναι μια μεταβλητή, δεν δηλώνεται, αλλά δημιουργείται και ονοματίζεται όταν της γίνει ανάθεση (assignment) κάποιας τιμής. Οντότητες που δημιουργούνται μέσα σε ένα block κώδικα, όπως το σώμα μιας συνάρτησης, μιας εντολής if ή for, κ.λπ., έχουν περιορισμένη εμβέλεια και μπορούν να χρησιμοποιηθούν μόνο μέσα στο block που έχουν ονομαστεί. Ένα τέτοιο όνομα δεν είναι προσπελάσιμο εκτός του block ακόμα και αν έχει την ίδια ονομασία και λέγεται ότι είναι τοπικής εμβέλειας (local scope). Ένα όνομα που δηλώνεται στο κύριο σώμα (block) του προγράμματος μπορεί να προσπελαστούν σε ολόκληρο το πρόγραμμα, ακόμα και στα block που έχουν οριστεί μέσα στο πρόγραμμα και λέγεται ότι είναι καθολικής εμβέλειας (global scope).^[33]

Μέσα σε ένα block ο διερμηνευτής (interpreter) χρησιμοποιεί την τοπική ονομασία μιας οντότητας ακόμα και αν υπάρχει άλλη με την ίδια ονομασία και καθολική εμβέλεια. Συγκεκριμένα ο διερμηνευτής όταν συναντήσει ένα όνομα (πχ. μεταβλητής) μέσα σε ένα block ελέγχει πρώτα αν έχει δημιουργηθεί αυτό το όνομα μέσα στο block (οπότε είναι τοπικής εμβέλειας) και το χρησιμοποιεί, έστω και αν υπάρχει ίδια ονομασία εκτός του block (καθολικής εμβέλειας). Αν δεν το βρει μέσα στο block τότε και μόνο τότε ελέγχει αν υπάρχει τέτοια ονομασία που να έχει δημιουργηθεί στο κύριο μέρος του προγράμματος (είναι καθολικής εμβέλειας) και χρησιμοποιεί αυτήν. Δηλαδή μια καθολικής εμβέλειας μεταβλητή χρησιμοποιείται μέσα σε ένα block, μόνο αν δεν υπάρχει τοπικής εμβέλειας μεταβλητή με το ίδιο όνομα, μέσα στο block.^[33]

Όταν μέσα σε ένα block γίνει η ανάθεση `num = 123`, τότε δημιουργείται (αναγκαστικά) η τοπικής εμβέλειας μεταβλητή `num` ακόμα και αν υπάρχει καθολικής εμβέλειας `num`. Στην περίπτωση που δεν είναι επιθυμητό να δημιουργηθεί η τοπική `num`, αλλά η τιμή 123 να ανατεθεί στην καθολική `num` γίνεται χρήση της δήλωσης `global num`. Δηλώνοντας `global num`, πριν από την `num = 123`, ο διερμηνευτής δεν δημιουργεί τοπική `num`, αλλά δίνει την τιμή 123, στην καθολική που ήδη υπάρχει.^[33]

Διαδικαστικός προγραμματισμός

Για παράδειγμα, ο παρακάτω γεννήτορας (generator) `generate_primes` παράγει πρώτους αριθμούς:^[32]

```
from itertools import count
def generate_primes(stop_at=None):
    primes = []
    for n in count(2):
```

```
if stop_at is not None and n > stop_at:  
    return  
composite = False  
for p in primes:  
    if not n % p:  
        composite = True  
        break  
    elif p**2 > n:  
        break  
if not composite:  
    primes.append(n)  
yield n
```

Και για να καλέσουμε τη συνάρτηση, υπολογίζοντας τους πρώτους αριθμούς από το 1 στο 100:^[33]

```
for i in generate_primes():  
    if i > 100: break  
    print(i)
```

Αντικειμενοστραφής προγραμματισμός

Παράδειγμα αντικειμενοστραφούς προγράμματος γραμμένο σε Python, το οποίο παρουσιάζει μια κλάση Υπάλληλος:^[33]

```
class Employee:  
    def __init__(self, id, name, position):  
        self.id = id;  
        self.name = name;  
        self.position = position;  
    def printEmployee(self):  
        print(self.id, "|", self.name, "|", self.position);  
  
e1 = Employee(1230, "Ιωάννης Παπαδάκης", "Βοηθός Προγραμματιστή");  
e1.printEmployee();
```

ΠΑΡΑΡΤΗΜΑ Β

Πηγαίος Κώδικα σε Python – Πιλοτικό Σενάριο Υλοποίησης

```
##Scenario 1 with 1 highway line with regard to the help line behavior
```

```
while flag:
```

```
    if(flag1==True):
        ave_cap.append(h.one_dcapacity(queue))
        ave_speed.append(h.one_dspped(queue1))
        ave_ratio.append(h.d_sp(queue,queue1))
        print('Starting Timestamp: ', list(queue))
        print('-----')
        flag1=False

    if(ramp.ramp_buf==0):
        flag = False
        print('-----')
        print('The ramp is free')
        print('Total Cars Passed: ', ramp.cars_passed)
        break

    elif(buffer.aux_buff!=0):
        if(queue[4]!=0):
            print('-----')
            print('The car in the help lane cannot change position. New check!')
            print('-----')
            ave_cap.append(h.one_dcapacity(queue))
            ave_speed.append(h.one_dspped(queue1))
            ave_ratio.append(h.d_sp(queue,queue1))
            h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
        elif(queue[4]==0):
            buffer.empty_aux()
            print('-----')
            print('New car entries the highway!')
            print('-----')
            queue[4]=1
            queue1[4]=buffer.aux_in_speed(20,65)
            ave_cap.append(h.one_dcapacity(queue))
            ave_speed.append(h.one_dspped(queue1))
            ave_ratio.append(h.d_sp(queue,queue1))
            h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))

    else:
        ramp.decrease_ramp()
        buffer.fullfil_aux()
```

```
ramp.cars_passed += 1
print('-----')
print('New car in the help lane!')
print('-----')
ave_cap.append(h.one_dcapacity(queue))
ave_speed.append(h.one_dspped(queue1))
ave_ratio.append(h.d_sp(queue,queue1))
h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
```

Πηγαίος Κώδικας σε Python – 1^ο Σενάριο Υλοποίησης

##Scenario 2 with 1 highway line with regard to the highway behavior

while flag:

```
if(flag1==True):
    ave_cap.append(h.one_dcapacity(queue))
    ave_speed.append(h.one_dspped(queue1))
    ave_ratio.append(h.d_sp(queue,queue1))
    print('Starting Timestamp: ', list(queue))
    print('Starting Speed Timestamp: ', list(queue1))
    print('-----')
    flag1=False

if(ramp.ramp_buf==0):
    flag = False
    print('-----')
    print('The ramp is free')
    print('Total Cars Passed: ', ramp.cars_passed)
    break

elif(buffer.aux_buff!=0):
    if(queue[4]!=0):
        print('-----')
        print('The car in the help lane cannot change position. New check!')
        print('-----')
        ave_cap.append(h.one_dcapacity(queue))
        ave_speed.append(h.one_dspped(queue1))
        ave_ratio.append(h.d_sp(queue,queue1))
        h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
    elif(queue[4]==0 and queue[3]==0):
        buffer.empty_aux()
        print('-----')
        print('Car is not in a shor Distance. New car entries the highway!')
        print('-----')
        queue[4]=1
```

```
queue1[4]=buffer.aux_in_speed(55,75)
ave_cap.append(h.one_dcapacity(queue))
ave_speed.append(h.one_dspped(queue1))
ave_ratio.append(h.d_sp(queue,queue1))
h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
elif(queue[4]==0 and queue1[3]<=70):
    buffer.empty_aux()
    print('-----')
    print('New car entries the highway!')
    print('-----')
    queue[4]=1
    queue1[4]=buffer.aux_in_speed(55,75)
    ave_cap.append(h.one_dcapacity(queue))
    ave_speed.append(h.one_dspped(queue1))
    ave_ratio.append(h.d_sp(queue,queue1))
    h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
else:
    ave_cap.append(h.one_dcapacity(queue))
    ave_speed.append(h.one_dspped(queue1))
    ave_ratio.append(h.d_sp(queue,queue1))
    h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
    continue
else:
    ramp.decrease_ramp()
    buffer.fullfil_aux()
    ramp.cars_passed += 1
    print('-----')
    print('New car in the help lane!')
    print('-----')
    ave_cap.append(h.one_dcapacity(queue))
    ave_speed.append(h.one_dspped(queue1))
    ave_ratio.append(h.d_sp(queue,queue1))
    h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
```

Πηγαίος Κώδικας σε Python – 2^ο Σενάριο Υλοποίησης

##Scenario 3 with 1 highway line with regard to the highway & ramp behavior

while flag:

```
if(flag1==True):
    ave_cap.append(h.one_dcapacity(queue))
    ave_speed.append(h.one_dspped(queue1))
    ave_ratio.append(h.d_sp(queue,queue1))
    print('Starting Timestamp: ', list(queue))
```



```
print('Starting Speed Timestamp: ', list(queue1))
print('-----')
flag1=False

while(buffer.aux_buff<5):
    ramp.decrease_ramp()
    buffer.fullfil_aux()
    ramp.cars_passed += 1
    print('-----')
    print('New car in the help lane!')
    print('-----')
    ave_cap.append(h.one_dcapacity(queue))
    ave_speed.append(h.one_dspeed(queue1))
    ave_ratio.append(h.d_sp(queue,queue1))
    h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))

if(ramp.ramp_buf==0):
    flag = False
    print('-----')
    print('The ramp is free')
    print('Total Cars Passed: ', ramp.cars_passed)
    break
elif(buffer.aux_buff!=0):
    if(queue[4]!=0):
        print('-----')
        print("The car in the help lane cannot change position. New check!")
        print('-----')
        ave_cap.append(h.one_dcapacity(queue))
        ave_speed.append(h.one_dspeed(queue1))
        ave_ratio.append(h.d_sp(queue,queue1))
        h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
    elif(queue[4]==0 and queue[3]==0):
        buffer.empty_aux()
        print('-----')
        print('Car is not in a shor Distance. New car entries the highway!')
        print('-----')
        queue[4]=1
        queue1[4]=buffer.aux_in_speed(55,75)
        ave_cap.append(h.one_dcapacity(queue))
        ave_speed.append(h.one_dspeed(queue1))
        ave_ratio.append(h.d_sp(queue,queue1))
        h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
    elif(queue[4]==0 and queue1[3]<=70):
        buffer.empty_aux()
        print('-----')
```

```
print('New car entries the highway!')
print('-----')
queue[4]=1
queue1[4]=buffer.aux_in_speed(55,75)
ave_cap.append(h.one_dcapacity(queue))
ave_speed.append(h.one_dspeed(queue1))
ave_ratio.append(h.d_sp(queue,queue1))
h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
else:
    ave_cap.append(h.one_dcapacity(queue))
    ave_speed.append(h.one_dspeed(queue1))
    ave_ratio.append(h.d_sp(queue,queue1))
    h.time_shift(queue,queue1,random.randint(0,1),random.randint(50,110))
    continue
else:
    continue
```

Πηγαίος Κώδικας σε Python – 3^ο Σενάριο Υλοποίησης

```
## Static Scenario Algorithm Execution on a Multiline Highway
```

```
start = time.time()
```

```
while flag:
```

```
    while(buffer.aux_buff<7):
        ramp.decrease_ramp()
        buffer.fullfil_aux()
        ramp.cars_passed += 1
        print('-----')
        print('New car in the help lane!')
        print('-----')
        h.three_dspeed(speed1,speed2,speed3)
        h.three_dcapacity(line1,line2,line3,mainroad)
        h.three_den_sp()
        ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
        ave_speed.append((h.three_dspeed(speed1,speed2,speed3)))
        ave_ratio.append((h.three_den_sp()))
        ramp1_t1.append(ramp.close_time(h.three_vert_dens(0,line1,line2,line3)))
        #ramp2_t2.append(r2.close_time(h.three_vert_dens(5,line1,line2,line3)))
        print('*****')
        h.three_dtimeshift(line1,line2,line3,mainroad)
        h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
        l1=list(line1)
        l2=list(line2)
        l3=list(line3)
```

```
list1.append(11)
list2.append(12)
list3.append(13)
i += 1

if(ramp.ramp_buf==0):
    flag = False
    print('-----')
    print('The ramp is free')
    print('Total Cars Passed: ', ramp.cars_passed)
    end = time.time()
    break
elif(buffer.aux_buff!=0):
    if(line3[4]!=0):
        print('-----')
        print("The car in the help lane cannot change position. New check!")
        print('-----')
        h.three_dspped(speed1,speed2,speed3)
        h.three_dcapacity(line1,line2,line3,mainroad)
        h.three_den_sp()
        ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
        ave_speed.append((h.three_dspped(speed1,speed2,speed3)))
        ave_ratio.append((h.three_den_sp()))
        ramp1_t1.append(ramp.close_time(h.three_vert_dens(0,line1,line2,line3)))
        #ramp2_t2.append(r2.close_time(h.three_vert_dens(5,line1,line2,line3)))
        print('*****')
        h.three_dtimeshift(line1,line2,line3,mainroad)
        h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
        l1=list(line1)
        l2=list(line2)
        l3=list(line3)
        list1.append(l1)
        list2.append(l2)
        list3.append(l3)
        i += 1
    elif(line3[4]==0 and line3[3]==0):
        buffer.empty_aux()
        print('-----')
        print('Car is not in a shor Distance. New car entries the highway!')
        print('-----')
        line3[4]=1
        speed3[4]=buffer.aux_in_speed(55,75)
        h.three_dspped(speed1,speed2,speed3)
        h.three_dcapacity(line1,line2,line3,mainroad)
        h.three_den_sp()
```

```
ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
ave_speed.append((h.three_dspeed(speed1,speed2,speed3)))
ave_ratio.append((h.three_den_sp()))
ramp1_t1.append(ramp.close_time(h.three_vert_dens(0,line1,line2,line3)))
#ramp2_t2.append(r2.close_time(h.three_vert_dens(5,line1,line2,line3)))
print('*****')
h.three_dtimeshift(line1,line2,line3,mainroad)
h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
l1=list(line1)
l2=list(line2)
l3=list(line3)
list1.append(l1)
list2.append(l2)
list3.append(l3)
i += 1
elif(line3[4]==0 and speed3[3]<=70):
    buffer.empty_aux()
    print('-----')
    print('New car entries the highway!')
    print('-----')
    line3[4]=1
    speed3[4]=buffer.aux_in_speed(55,75)
    h.three_dspeed(speed1,speed2,speed3)
    h.three_dcapacity(line1,line2,line3,mainroad)
    h.three_den_sp()
    ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
    ave_speed.append((h.three_dspeed(speed1,speed2,speed3)))
    ave_ratio.append((h.three_den_sp()))
    ramp1_t1.append(ramp.close_time(h.three_vert_dens(0,line1,line2,line3)))
    #ramp2_t2.append(r2.close_time(h.three_vert_dens(5,line1,line2,line3)))
    print('*****')
    h.three_dtimeshift(line1,line2,line3,mainroad)
    h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
    l1=list(line1)
    l2=list(line2)
    l3=list(line3)
    list1.append(l1)
    list2.append(l2)
    list3.append(l3)
    i += 1
else:
    h.three_dspeed(speed1,speed2,speed3)
    h.three_dcapacity(line1,line2,line3,mainroad)
    h.three_den_sp()
    ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
```

```
ave_speed.append((h.three_ds(speed1,speed2,speed3)))
ave_ratio.append((h.three_den_sp()))
ramp1_t1.append(ramp.close_time(h.three_vert_dens(0,line1,line2,line3)))
#ramp2_t2.append(r2.close_time(h.three_vert_dens(5,line1,line2,line3)))
print('*****')
h.three_dtimeshift(line1,line2,line3,mainroad)
h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
l1=list(line1)
l2=list(line2)
l3=list(line3)
list1.append(l1)
list2.append(l2)
list3.append(l3)
i += 1
continue
else:
continue
```

Πηγαίος Κώδικας σε Python – 4^ο Σενάριο Υλοποίησης

```
## Multi-Ramp Algorithm Scenario Execution on a Multiline Highway
```

```
t_end = time.time() + delay
```

```
while time.time() < t_end:
```

```
if(r1.ramp_buf==0):
```

```
    #flag = False
```

```
    print('-----')
```

```
    print("The ramp is free..Waiting for cars!")
```

```
    print("Total Cars Passed: ', r1.cars_passed)
```

```
    r1.random_fullfil(0,1)
```

```
    #continue
```

```
if(r2.ramp_buf==0):
```

```
    #flag = False
```

```
    print('-----')
```

```
    print("The ramp is free..Waiting for cars!")
```

```
    print("Total Cars Passed: ', r2.cars_passed)
```

```
    r2.random_fullfil(0,1)
```

```
    #continue
```

```
if(total_cap>=70):
```

```
    if(line3[4]!=0 and line3[10]!=0):
```

```
print('-----')
print("The car in the help lane 1 and 2 cannot change position. New check!")
print('-----')
h.three_dspped(speed1,speed2,speed3)
h.three_dcapacity(line1,line2,line3,mainroad)
h.three_den_sp()
ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
ave_speed.append((h.three_dspped(speed1,speed2,speed3)))
ave_ratio.append((h.three_den_sp()))
ramp1_t1.append(r1.close_time(h.three_vert_dens(0,line1,line2,line3)))
ramp2_t2.append(r2.close_time(h.three_vert_dens(6,line1,line2,line3)))
print('*****')
h.three_dtimeshift(line1,line2,line3,mainroad)
h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
l1=list(line1)
l2=list(line2)
l3=list(line3)
list1.append(l1)
list2.append(l2)
list3.append(l3)
i += 1
elif(line3[4]==0 and line3[10]!=0):
    print('-----')
    print("The car in the help lane 2 cannot change position but can in position 1. New check!")
    print('-----')
    buffer1.empty_aux()
    line3[4]=1
    mainroad[2,:]=line3
    speed3[4]=buffer2.aux_in_speed(55,75)
    roadspeed[2,:]=speed3
    h.three_dspped(speed1,speed2,speed3)
    h.three_dcapacity(line1,line2,line3,mainroad)
    h.three_den_sp()
    ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
    ave_speed.append((h.three_dspped(speed1,speed2,speed3)))
    ave_ratio.append((h.three_den_sp()))
    ramp1_t1.append(r1.close_time(h.three_vert_dens(0,line1,line2,line3)))
    ramp2_t2.append(r2.close_time(h.three_vert_dens(6,line1,line2,line3)))
    print('*****')
    h.three_dtimeshift(line1,line2,line3,mainroad)
    h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
    l1=list(line1)
    l2=list(line2)
    l3=list(line3)
    list1.append(l1)
```

```
list2.append(l2)
list3.append(l3)
i += 1
elif(line3[4]!=0 and line3[10]==0):
    print('-----')
    print('The car in the help lane 1 cannot change position but can in position 2. New check!')
    print('-----')
    buffer2.empty_aux()
    line3[10]=1
    mainroad[2,:] = line3
    speed3[10]=buffer2.aux_in_speed(55,75)
    roadspeed[2,:] = speed3
    h.three_dspped(speed1,speed2,speed3)
    h.three_dcapacity(line1,line2,line3,mainroad)
    h.three_den_sp()
    ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
    ave_speed.append((h.three_dspped(speed1,speed2,speed3)))
    ave_ratio.append((h.three_den_sp()))
    ramp1_t1.append(r1.close_time(h.three_vert_dens(0,line1,line2,line3)))
    ramp2_t2.append(r2.close_time(h.three_vert_dens(6,line1,line2,line3)))
    print('*****')
    h.three_dtimeshift(line1,line2,line3,mainroad)
    h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
    l1=list(line1)
    l2=list(line2)
    l3=list(line3)
    list1.append(l1)
    list2.append(l2)
    list3.append(l3)
    i += 1
elif(line3[4]==0 and line3[10]==0):
    print('-----')
    print('New car entries the highway from the two auxiliary buffers!')
    print('-----')
    buffer1.empty_aux()
    buffer2.empty_aux()
    line3[4]=1
    mainroad[2,:] = line3
    speed3[4]=buffer2.aux_in_speed(55,75)
    roadspeed[2,:] = speed3
    line3[10]=1
    mainroad[2,:] = line3
    speed3[10]=buffer2.aux_in_speed(55,75)
    roadspeed[2,:] = speed3
    h.three_dspped(speed1,speed2,speed3)
```

```
h.three_dcapacity(line1,line2,line3,mainroad)
h.three_den_sp()
ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
ave_speed.append((h.three_dspeed(speed1,speed2,speed3)))
ave_ratio.append((h.three_den_sp()))
ramp1_t1.append(r1.close_time(h.three_vert_dens(0,line1,line2,line3)))
ramp2_t2.append(r2.close_time(h.three_vert_dens(6,line1,line2,line3)))
print('*****')
h.three_dtimeshift(line1,line2,line3,mainroad)
h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
l1=list(line1)
l2=list(line2)
l3=list(line3)
list1.append(l1)
list2.append(l2)
list3.append(l3)
i += 1
else:
if(line3[4]!=0 and line3[10]!=0):
print('-----')
print("The car in the help lane 1 and 2 cannot change position. New check!")
print('-----')
h.three_dspeed(speed1,speed2,speed3)
h.three_dcapacity(line1,line2,line3,mainroad)
h.three_den_sp()
ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
ave_speed.append((h.three_dspeed(speed1,speed2,speed3)))
ave_ratio.append((h.three_den_sp()))
ramp1_t1.append(r1.close_time(h.three_vert_dens(0,line1,line2,line3)))
ramp2_t2.append(r2.close_time(h.three_vert_dens(6,line1,line2,line3)))
print('*****')
h.three_dtimeshift(line1,line2,line3,mainroad)
h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
l1=list(line1)
l2=list(line2)
l3=list(line3)
list1.append(l1)
list2.append(l2)
list3.append(l3)
i += 1
elif(line3[4]==0 and line3[10]!=0):
print('-----')
print("The car in the help lane 2 cannot change position but can in position 1. New check!")
buffer1.empty_aux()
print('-----')
```



```

line3[4]=1
mainroad[2,:]=line3
speed3[4]=buffer2.aux_in_speed(55,75)
roadspeed[2,:]=speed3
h.three_dspeed(speed1,speed2,speed3)
h.three_dcapacity(line1,line2,line3,mainroad)
h.three_den_sp()
ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
ave_speed.append((h.three_dspeed(speed1,speed2,speed3)))
ave_ratio.append((h.three_den_sp()))
ramp2_t2.append(r2.close_time(h.three_vert_dens(6,line1,line2,line3)))
if((buffer1.aux_buff<7          and          h.one_vert_dcapacity(4,10,line3)<70)          and
h.three_vert_dens(0,line1,line2,line3)<60):
    r1.ramp_buf -= 1
    buffer1.aux_buff += 1
    r1.cars_passed += 1
    ramp1_t1.append(0)
else:
    ramp1_t1.append(r1.close_time(h.three_vert_dens(0,line1,line2,line3)))
print('*****')
h.three_dtimeshift(line1,line2,line3,mainroad)
h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
l1=list(line1)
l2=list(line2)
l3=list(line3)
list1.append(l1)
list2.append(l2)
list3.append(l3)
i += 1
elif(line3[4]!=0 and line3[10]==0):
    print('-----')
    print('The car in the help lane 1 cannot change position but can in position 2. New check!')
    print('-----')
    buffer2.empty_aux()
    line3[10]=1
    mainroad[2,:]=line3
    speed3[10]=buffer2.aux_in_speed(55,75)
    roadspeed[2,:]=speed3
    h.three_dspeed(speed1,speed2,speed3)
    h.three_dcapacity(line1,line2,line3,mainroad)
    h.three_den_sp()
    ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
    ave_speed.append((h.three_dspeed(speed1,speed2,speed3)))
    ave_ratio.append((h.three_den_sp()))
    ramp1_t1.append(r1.close_time(h.three_vert_dens(0,line1,line2,line3)))

```

```

if(buffer2.aux_buff<7 and h.three_vert_dens(6,line1,line2,line3)<80):
    r2.ramp_buf -= 1
    buffer2.aux_buff += 1
    r2.cars_passed += 1
    ramp2_t2.append(0)
else:
    ramp2_t2.append(r2.close_time(h.three_vert_dens(6,line1,line2,line3)))
print('*****')
h.three_dtimeshift(line1,line2,line3,mainroad)
h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
l1=list(line1)
l2=list(line2)
l3=list(line3)
list1.append(l1)
list2.append(l2)
list3.append(l3)
i += 1
elif(line3[4]==0 and line3[10]==0):
    print('-----')
    print('New car entries the highway from the two auxiliary buffers!')
    print('-----')
    buffer1.empty_aux()
    buffer2.empty_aux()
    line3[4]=1
    mainroad[2,:]=line3
    speed3[4]=buffer2.aux_in_speed(55,75)
    roadspeed[2,:]=speed3
    line3[10]=1
    mainroad[2,:]=line3
    speed3[10]=buffer2.aux_in_speed(55,75)
    roadspeed[2,:]=speed3
    h.three_dspeed(speed1,speed2,speed3)
    h.three_dcapacity(line1,line2,line3,mainroad)
    h.three_den_sp()
    ave_cap.append((h.three_dcapacity(line1,line2,line3,mainroad)))
    ave_speed.append((h.three_dspeed(speed1,speed2,speed3)))
    ave_ratio.append((h.three_den_sp()))
    if((buffer1.aux_buff<7 and h.one_vert_dcapacity(4,10,line3)<70) and
h.three_vert_dens(0,line1,line2,line3)<60):
        r1.ramp_buf -= 1
        buffer1.aux_buff += 1
        r1.cars_passed += 1
        ramp1_t1.append(0)
    else:
        ramp1_t1.append(r1.close_time(h.three_vert_dens(0,line1,line2,line3)))
    
```

```
if(buffer2.aux_buff<7 and h.three_vert_dens(6,line1,line2,line3)<80):
    r2.ramp_buf -= 1
    buffer2.aux_buff += 1
    r2.cars_passed += 1
    ramp2_t2.append(0)
else:
    ramp2_t2.append(r2.close_time(h.three_vert_dens(6,line1,line2,line3)))
print('*****')
h.three_dtimeshift(line1,line2,line3,mainroad)
h.three_speedshift(speed1,speed2,speed3,roadspeed,line1,line2,line3)
l1=list(line1)
l2=list(line2)
l3=list(line3)
list1.append(l1)
list2.append(l2)
list3.append(l3)
i += 1
```