



ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

DEPARTMENT OF INDUSTRIAL DESIGN AND PRODUCTION ENGINEERING

Πτυχιακή Εργασία

**Ανάπτυξη εφαρμογών μηχανικής μάθησης σε
προσομοιωμένο ρομποτικό σύστημα μεταβλητής αυτονομίας**

Συγγραφέας

Παρασκευάς Χατζηθάνος (ΑΜ: 44496)

Επιβλέπων: Γρηγόρης Νικολάου

Αθήνα, Σεπτέμβριος 2019

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

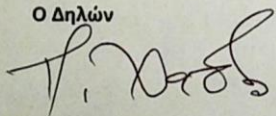
Ο/Η κάτωθι υπογεγραμμένος/η Καθ. Γ. Δ. Κουρής του Ξέρου, φοιτητής του Τμήματος Πολιτικής και Κοινωνικής Επιστήμης και Παιδαγωγικής του Πανεπιστημίου Δυτικής Αττικής, πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε, ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα, σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασή της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση Π.Ε με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού βμήνου από την ημερομηνία ανάθεσής της.

Ο Δηλών



Ημερομηνία

19/09/2019

Ευχαριστίες – Αφιερώσεις

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον Καθηγητή κ. Δρ. Γρηγόριο Νικολάου για την βοήθεια που μου έδωσε και την εμπιστοσύνη που μου έδειξε να πραγματοποιήσω την πτυχιακή μου εργασία μαζί του.

Επίσης θα ήθελα να ευχαριστήσω τον κ. Δρ. Μανώλη Χίου για την πολύτιμη βοήθεια και χρόνο που διέθεσε για την περάτωση της πτυχιακής μου εργασίας. Οι σημαντικές υποδείξεις και συμβουλές τους με κατεύθυναν σ' ένα σωστό τρόπο σκέψης πάνω απ' όλα και μου προσέφεραν σημαντικά εφόδια για την μετέπειτα ακαδημαϊκή μου σταδιοδρομία.

Τέλος, θέλω να εκφράσω ένα τεράστιο ευχαριστώ στην οικογένεια και τους ανθρώπους που στάθηκαν δίπλα μου, για την στήριξη και την εμπιστοσύνη που μου έδειξαν όλα αυτά τα χρόνια των σπουδών μου.

Ημερομηνία Σεπτέμβριος 2019

Συγγραφέας Παρασκευάς Χατζηθάνος

Περίληψη στα ελληνικά

Η πτυχιακή αυτή εργασία πραγματεύεται αλγόριθμους για την πρόβλεψη των σωστών ενεργειών των χειριστών σε προσομοιωμένο ρομπότ. Σήμερα γίνονται πολυάριθμες μελέτες για την πρόβλεψη και αξιολόγηση ενός χειριστή, είτε πρόκειται για ρομπότ είτε για κάποιο όχημα. Το πρώτο βήμα για να επιτευχθεί αυτό είναι να οριστεί τι είναι “σωστή” ή ακίνδυνη οδήγηση.

Σκοπός της εργασίας αυτής είναι η ανάπτυξη μοντέλων που θα προβλέπουν με υψηλή ακρίβεια την “σωστή οδήγηση” ενός ρομποτικού συστήματος. Με τον όρο σωστή οδήγηση εννοείται, η ομαλή εναλλαγή μεταξύ δεξιάς και αριστερής στροφής καθώς και η ομαλή εναλλαγή των γωνιακών ταχυτήτων του ρομποτικού συστήματος. Με την δημιουργία ενός τέτοιου μοντέλου, μπορεί να γίνει σύγκριση της εξόδου των μοντέλων με την εκάστοτε οδήγηση του χειριστή και σαν αποτέλεσμα να αξιολογηθούν οι χειρισμοί σε πραγματικό χρόνο. Συγκεκριμένα πέντε μοντέλα αναπτύχθηκαν. Το πρώτο μοντέλο ονομάζεται Αυτοπαλίνδρομο Μοντέλο (Autoregressive Model), το δεύτερο μοντέλο δημιουργήθηκε με χρονοσειρές Taylor. Τα επόμενα τρία είναι μοντέλα που αναπτύχθηκαν με αλγόριθμους μηχανικής μάθησης οι οποίοι είναι οι εξής, αλγόριθμος γραμμικής κατάβασης δυναμικού (linear regression), αλγόριθμος random forest και τέλος νευρωνικά δίκτυα.

Αρχικά, θα παρουσιαστούν τα πειράματα και πως αυτά υλοποιήθηκαν. Μετά θα γίνει μια θεωρητική ανάλυση των μοντέλων και κλείνοντας, μετά από μια σύντομη περιγραφή της συλλογής και επεξεργασίας των δεδομένων, θα παρουσιαστούν αναλυτικά τα μοντέλα που αναπτύχθηκαν καθώς και τις προβλέψεις που κάνουν.

Λέξεις Κλειδιά: πρόβλεψη, ρομπότ, αλγόριθμους μηχανικής μάθησης, Αυτοπαλίνδρομο μοντέλο, μοντέλο γραμμικής κατάβασης δυναμικού, random forest, νευρωνικά δίκτυα

Περίληψη στα αγγλικά

This thesis deals with algorithms for predicting the correct actions, of operators, on a simulated robot. Numerous studies are currently carried out to predict and evaluate an operator, be it a robot or a vehicle. The first step to doing so is to determine what is "right" or safe driving.

The purpose of this work is to develop models that will accurately predict the "correct driving" of a robotic system. Proper driving means the smooth rotation between the right and left turns as well as the smooth rotation of the angular speeds of the robotic system. By creating such a model, it is possible to compare the output of the models with the individual driving of the user and as a result, evaluate real-time manipulations.

Specifically, five models were developed. The first model is called the Autoregressive Model, the second model was created with Taylor time series. The following three are models developed with machine learning algorithms that are: linear regression, random forest algorithm and neural network.

Initially in this thesis, the experiments will be presented and how they were implemented. Then a theoretical analysis of the models will be carried out and concluding, after a brief description of the data collection and pre-processing, the models developed as well as their predictions will be presented in detail.

Keywords: predict, robot, Autoregressive Model, machine learning algorithms, linear regression, random forest algorithm, neural network

Πίνακας περιεχομένων

ΕΥΧΑΡΙΣΤΙΕΣ – ΑΦΙΕΡΩΣΕΙΣ	III
ΠΕΡΙΛΗΨΗ ΣΤΑ ΕΛΛΗΝΙΚΑ.....	IV
ΠΕΡΙΛΗΨΗ ΣΤΑ ΑΓΓΛΙΚΑ	V
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	VI
ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ	IX
ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ	XII
ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ	1
1.1 ΣΚΟΠΟΣ ΚΑΙ ΣΤΟΧΟΙ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	2
1.2 ΡΟΜΠΟΤΙΚΗ	2
1.3 ΚΑΤΗΓΟΡΙΕΣ ΡΟΜΠΟΤ	3
1.3.1 <i>Κινούμενα Ρομπότ</i>	4
1.3.2 <i>Ρομπότ Σταθερής Βάσης</i>	5
1.3.3 <i>AGVs (automated guided vehicle)</i>	5
1.3.4 <i>UUV και UAV</i>	6
1.4 ΕΡΕΥΝΗΤΙΚΟΙ ΤΟΜΕΙΣ ΣΤΗΝ ΡΟΜΠΟΤΙΚΗ	6
1.4.1 <i>“Έξυπνα” Αμάξια (“Smart” cars)</i>	7
1.4.2 <i>Ρομποτικά συστήματα εξερεύνησης και διάσωσης (SaR)</i>	8
1.5 ΜΕΤΑΒΛΗΤΗ ΑΥΤΟΝΟΜΙΑ.....	9
1.6 ΦΟΡΤΟ ΕΡΓΑΣΙΑΣ ΧΕΙΡΙΣΤΗ	12
ΚΕΦΑΛΑΙΟ 2. ΘΕΩΡΗΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΜΟΝΤΕΛΩΝ ΟΔΗΓΗΣΗΣ ΜΕ ΥΨΗΛΗ ΑΠΟΔΟΣΗ	14
2.1 ΠΡΟΒΛΗΜΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ.....	14
2.2 ΑΠΛΟ ΚΑΙ ΠΟΛΛΑΠΛΟ ΜΟΝΤΕΛΟ ΚΑΤΑΒΑΣΗ ΔΥΝΑΜΙΚΟΥ (LINEAR REGRESSION)	14
2.3 ΑΥΤΟΠΑΛΙΝΔΡΟΜΑ ΜΟΝΤΕΛΑ	16
2.4 ΑΝΑΠΤΥΓΜΑ ΤΕΙΛΟΡ.....	17
2.5 ΜΟΝΤΕΛΟ RANDOM FOREST	17
2.6 ΜΗΧΑΝΕΣ ΔΙΑΝΥΣΜΑΤΩΝ ΥΠΟΣΤΗΡΙΞΗΣ (SUPPORT VECTOR REGRESSION).....	19
2.7 ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	22
2.7.1 <i>Νευρώνες και Αρχιτεκτονική Νευρωνικών Δικτύων</i>	23
2.7.2 <i>Συνάρτηση Κόστους</i>	25

2.7.3	Εκπαίδευση και πρόβλεψη νευρωνικών δικτύων	26
ΚΕΦΑΛΑΙΟ 3. ΡΟΜΠΟΤΙΚΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ (ROS)		28
3.1	ΕΙΣΑΓΩΓΗ ΣΤΟ ROS	28
3.2	ΠΥΡΗΝΑΣ ROS ΚΑΙ ΕΡΓΑΛΕΙΑ ΕΠΙΚΟΙΝΩΝΙΑΣ	29
3.2.1	Θέματα (Topics)	29
3.2.2	Υπηρεσίες (Services)	30
3.2.3	Ενέργειες (Actions)	30
3.3	ΑΝΑΠΤΥΞΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ ROS	31
3.4	ΒΙΒΛΙΟΘΗΚΕΣ ΤΟΥ ROS	32
3.5	Η ΚΟΙΝΟΤΗΤΑ ΤΟΥ ROS	34
3.6	ΠΟΤΕ ΧΡΗΣΙΜΟΠΟΙΕΙΤΑΙ ΤΟ ROS.....	35
ΚΕΦΑΛΑΙΟ 4. ΣΥΛΛΟΓΗ ΚΑΙ ΠΡΟΕΤΟΙΜΑΣΙΑ ΔΕΔΟΜΕΝΩΝ		36
4.1	ΣΥΛΛΟΓΗ ΔΕΔΟΜΕΝΩΝ.....	36
4.2	ΠΡΟΕΡΓΑΣΙΑ ΤΩΝ ΔΕΔΟΜΕΝΩΝ	37
4.2.1	Δειγματοληψία	37
4.2.2	Μετατροπή από <i>rosbag</i> σε μορφή <i>csv</i>	38
4.2.3	Μορφοποίηση	38
4.3	ΠΕΡΙΓΡΑΦΗ ΤΕΛΙΚΗΣ ΜΟΡΦΗΣ ΔΕΔΟΜΕΝΩΝ	39
ΚΕΦΑΛΑΙΟ 5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΟΝΤΕΛΩΝ		40
5.1	ΜΟΝΤΕΛΟ ΚΑΤΑΒΑΣΗΣ ΔΥΝΑΜΙΚΟΥ	41
5.1.1	Γωνιακή ταχύτητα.....	41
5.1.2	Γραμμική ταχύτητα	42
5.1.3	Εντολές χειριστή δεξιά/αριστερά	43
5.1.4	Εντολές χειριστή μπρος/πίσω.....	44
5.2	ΑΥΤΟΠΑΛΙΝΔΡΟΜΑ ΜΟΝΤΕΛΑ	45
5.2.1	Γωνιακή ταχύτητα.....	46
5.2.2	Γραμμική ταχύτητα	47
5.2.3	Εντολές χειριστή δεξιά/αριστερά	49
5.2.4	Εντολές χειριστή μπρος/πίσω.....	50
5.3	ΑΝΑΠΤΥΓΜΑ ΤΕΙΛΟΡ.....	52
5.3.1	Γωνιακή ταχύτητα.....	52
5.3.2	Γραμμική ταχύτητα	53
5.3.3	Εντολές χειριστή δεξιά/αριστερά	54
5.3.4	Εντολές χειριστή μπρος/πίσω.....	55
5.4	RANDOM FOREST.....	56
5.4.1	Γωνιακή ταχύτητα.....	56

5.4.2	Γραμμική ταχύτητα	59
5.4.3	Εντολές χειριστή μπρος/πίσω.....	61
5.4.4	Εντολές χειριστή πάνω/κάτω	63
5.5	ΜΗΧΑΝΕΣ ΔΙΑΝΥΣΜΑΤΩΝ ΥΠΟΣΤΗΡΙΞΗΣ	65
5.5.1	Γωνιακή ταχύτητα.....	65
5.5.2	Γραμμική ταχύτητα	65
5.5.3	Εντολές χειριστή δεξιά/αριστερά.....	66
5.5.4	Εντολές χειριστή πάνω/κάτω	67
5.6	ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ.....	68
5.7	ΚΥΡΙΟΤΕΡΑ ΕΥΡΗΜΑΤΑ/ ΑΠΟΤΕΛΕΣΜΑΤΑ	69
ΚΕΦΑΛΑΙΟ 6. ΣΥΖΗΤΗΣΗ – ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....		71
6.1	ΑΝΑΚΕΦΑΛΑΙΩΣΗ.....	71
6.2	ΣΥΖΗΤΗΣΗ / ΣΥΜΠΕΡΑΣΜΑΤΑ	71
6.3	ΠΡΑΚΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ ΤΗΣ ΕΡΕΥΝΑΣ	72
ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ.....		73

Πίνακας Σχημάτων

Εικόνα 1. Το Tesla Autopilot και το σύστημα αναγνώρισης περιβάλλοντος.....	3
Εικόνα 2. Ρομπότ διάσωσης coyote III.....	4
Εικόνα 3. Γραμμή παραγωγής αποτελούμενη από ρομπότ σταθερής βάσης.....	5
Εικόνα 4. Εξάποδο ρομπότ.....	6
Εικόνα 5. Παράδειγμα ρομπότ εξερεύνησης σε σεισμική καταστροφή στην Ιαπωνία.....	8
Εικόνα 6. Η ταξινόμηση κατά Beer (Beer,2014).....	11
Εικόνα 7. Παράδειγμα γραμμικής κατάβασης δυναμικού.....	15
Εικόνα 8. Παράδειγμα δέντρου αποφάσεων.....	18
Εικόνα 9. Παράδειγμα μοντέλου Random Forest.....	19
Εικόνα 10. Διαφορές σε μεθόδους ταξινόμησης.....	20
Εικόνα 11. Παράδειγμα SVM.....	20
Εικόνα 12. Παράδειγμα SVM με το μέγιστο περιθώριο epsilon.....	21
Εικόνα 13. Αναγωγή SVM σε παραπάνω διάσταση.....	22
Εικόνα 14. Νευρώνας.....	22
Εικόνα 15. Τεχνητός Νευρώνας.....	23
Εικόνα 16. Αρχιτεκτονική δικτύου.....	24
Εικόνα 17. Διαφορές μεταξύ ρηχού και βαθιού νευρωνικού δικτύου.....	25
Εικόνα 18. Παράδειγμα των θεμάτων μιας εφαρμογής στο ROS.....	29
Εικόνα 19. Παράδειγμα υπηρεσίας στο ROS.....	30
Εικόνα 20. Παράδειγμα ενεργειών στο ROS.....	31
Εικόνα 21. Σχήμα επικοινωνίας μεταξύ διαφορετικών κόμβων.....	32
Εικόνα 22. Λεπτομερές σχήμα επικοινωνίας μεταξύ κόμβων.....	33
Εικόνα 23. Πρόβλεψη μοντέλου κατάβασης γωνιακής ταχύτητας στο σετ ελέγχου.....	42
Εικόνα 24. Πρόβλεψη μοντέλου κατάβασης γωνιακής ταχύτητας σετ ελέγχου σε μεγέθυνση.....	42
Εικόνα 25. Πρόβλεψη μοντέλου κατάβασης γραμμικής ταχύτητας στο σετ ελέγχου.....	43
Εικόνα 26. Πρόβλεψη μοντέλου κατάβασης γραμμικής ταχύτητας σετ ελέγχου σε μεγέθυνση.....	43
Εικόνα 27. Πρόβλεψη μοντέλου κατάβασης δεξιάς/αριστερής εντολής στο σετ ελέγχου....	44

Εικόνα 28. Πρόβλεψη μοντέλου κατάβασης δεξιάς/αριστερής εντολής σετ ελέγχου σε μεγέθυνση	44
Εικόνα 29. Πρόβλεψη μοντέλου κατάβασης μπρος/πίσω εντολής στο σετ ελέγχου.....	45
Εικόνα 30. Πρόβλεψη μοντέλου κατάβασης μπρος/πίσω εντολής στο σετ ελέγχου σε μεγέθυνση	45
Εικόνα 31. Μέσο τετραγωνικό σφάλμα και αριθμός καθυστερήσεων μοντέλου γωνιακής ταχύτητας	46
Εικόνα 32. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς μοντέλου γωνιακής ταχύτητας	46
Εικόνα 33. Πρόβλεψη αυτοπαλίνδρομου μοντέλου γωνιακής ταχύτητας.....	47
Εικόνα 34. Μέσο τετραγωνικό σφάλμα και αριθμός καθυστερήσεων γραμμικής ταχύτητας.....	47
Εικόνα 35. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς μοντέλου γραμμικής ταχύτητας	48
Εικόνα 36. Μέσο τετραγωνικό σφάλμα και αριθμός καθυστερήσεων μοντέλου δεξιών/αριστερών εντολών χειριστή.....	49
Εικόνα 37. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς μοντέλου δεξιών/αριστερών εντολών χειριστή.....	49
Εικόνα 38. Πρόβλεψη αυτοπαλίνδρομου μοντέλου δεξιάς/αριστερής εντολής στο σετ ελέγχου.....	50
Εικόνα 39. Μέσο τετραγωνικό σφάλμα και αριθμός καθυστερήσεων μοντέλου μπρος/πίσω εντολών χειριστή.....	50
Εικόνα 40. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς μοντέλου μπρος/πίσω εντολών χειριστή.....	51
Εικόνα 41. Πρόβλεψη αυτοπαλίνδρομου μοντέλου πάνω/κάτω εντολής στο σετ ελέγχου..	51
Εικόνα 42. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ γωνιακής ταχύτητας στο σετ ελέγχου	52
Εικόνα 43. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ γωνιακής ταχύτητας στο σετ ελέγχου σε μεγέθυνση	53
Εικόνα 44. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ γραμμικής ταχύτητας στο σετ ελέγχου.....	53
Εικόνα 45. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ γραμμικής ταχύτητας στο σετ ελέγχου σε μεγέθυνση	54
Εικόνα 46. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ δεξιών/αριστερών εντολών στο σετ ελέγχου.....	54

Εικόνα 47. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ δεξιών/αριστερών εντολών στο σετ ελέγχου σε μεγέθυνση	55
Εικόνα 48. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ πάνω/κάτω εντολών στο σετ ελέγχου	55
Εικόνα 49. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ πάνω/κάτω εντολών στο σετ ελέγχου σε μεγέθυνση	56
Εικόνα 50. Μέσο τετραγωνικό σφάλμα και αριθμός δέντρων μοντέλου	58
Εικόνα 51. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς φόρτο χειριστή	58
Εικόνα 52. Μέσο τετραγωνικό σφάλμα και αριθμός δέντρων μοντέλου γραμμικής ταχύτητας	60
Εικόνα 53. Διαφορά στα σφάλματα μοντέλου γραμμικής ταχύτητας μεταξύ πειραμάτων με φόρτο και χωρίς φόρτο χειριστή.....	60
Εικόνα 54. Μέσο τετραγωνικό σφάλμα και αριθμός δέντρων μοντέλου	62
Εικόνα 55. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς φόρτο χειριστή	62
Εικόνα 56. Μέσο τετραγωνικό σφάλμα και αριθμός δέντρων μοντέλου	64
Εικόνα 57. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς φόρτο χειριστή	64
Εικόνα 58. Πρόβλεψη μοντέλου SVR	65
Εικόνα 59. Πρόβλεψη μοντέλου SVR	66
Εικόνα 60. Πρόβλεψη μοντέλου SVR	67
Εικόνα 61. Πρόβλεψη μοντέλου SVR	68
Εικόνα 62. Πρόβλεψη νευρωνικού δικτύου στο σετ εκπαίδευσης με μικρό σφάλμα	69

Πίνακας Πινάκων

Πίνακας 1. Χρήσιμα θέματα.....	36
Πίνακας 2. Παράδειγμα δεδομένων γωνιακής ταχύτητας	40
Πίνακας 3. Τελικά δεδομένα πειραμάτων	40
Πίνακας 4. Σφάλματα μοντέλου κατάβασης γωνιακής ταχύτητας.....	41
Πίνακας 5. Παράμετροι μοντέλου κατάβασης γωνιακής ταχύτητας.....	42
Πίνακας 6. Σφάλματα μοντέλου κατάβασης γραμμικής ταχύτητας.....	42
Πίνακας 7. Παράμετροι μοντέλου κατάβασης γραμμικής ταχύτητας.....	43
Πίνακας 8. Σφάλματα μοντέλου κατάβασης δεξιάς/αριστερής εντολής	43
Πίνακας 9. Παράμετροι μοντέλου κατάβασης δεξιάς/αριστερής εντολής	44
Πίνακας 10. Σφάλματα μοντέλου κατάβασης μπρος/πίσω εντολής.....	44
Πίνακας 11. Παράμετροι μοντέλου κατάβασης μπρος/πίσω εντολής.....	45
Πίνακας 12. Παράμετροι και σφάλματα αυτοπαλίνδρομου μοντέλου δεύτερης τάξης γωνιακής ταχύτητας.....	46
Πίνακας 13. Παράμετροι και σφάλματα αυτοπαλίνδρομου μοντέλου δεύτερης τάξης γραμμικής ταχύτητας	48
Πίνακας 14. Παράμετροι και σφάλματα αυτοπαλίνδρομου μοντέλου δεύτερης τάξης δεξιών/αριστερών εντολών χειριστή.....	50
Πίνακας 15. Παράμετροι και σφάλματα αυτοπαλίνδρομου μοντέλου δεύτερης τάξης μπρος/πίσω εντολών χειριστή.....	51
Πίνακας 16. Σφάλματα μοντέλου αναπτύγματος Τέιλορ για την γωνιακή ταχύτητα	52
Πίνακας 17. Σφάλματα μοντέλου αναπτύγματος Τέιλορ για την γραμμική ταχύτητα	53
Πίνακας 18. Σφάλματα μοντέλου αναπτύγματος Τέιλορ για τις δεξιά/αριστερά εντολές	54
Πίνακας 19. Σφάλματα μοντέλου αναπτύγματος Τέιλορ για τις πάνω/κάτω εντολές.....	55
Πίνακας 20. Σφάλματα μοντέλου random forest γωνιακής ταχύτητας με βάση τον αριθμό των δέντρων	56
Πίνακας 21. Σφάλματα μοντέλου random forest γραμμικής ταχύτητας με βάση τον αριθμό των δέντρων	59
Πίνακας 22. Σφάλματα μοντέλου random forest με βάση τον αριθμό των δέντρων	61
Πίνακας 23. Σφάλματα μοντέλου random forest με βάση τον αριθμό των δέντρων	63
Πίνακας 24. Μέσα τετραγωνικά σφάλματα και παράμετροι.....	65

Πίνακας 25. Μέσα τετραγωνικά σφάλματα και παράμετροι	65
Πίνακας 26. Μέσα τετραγωνικά σφάλματα και παράμετροι	66
Πίνακας 27. Μέσα τετραγωνικά σφάλματα και παράμετροι	67
Πίνακας 28. Αρχιτεκτονική νευρωνικού δικτύου.....	68

Κεφάλαιο 1. Εισαγωγή

Η Ρομποτική είναι κλάδος της τεχνολογίας που ασχολείται με τη σχεδίαση, την ανάπτυξη και τη μελέτη ρομπότ. Η επιστήμη της Ρομποτικής αποτελεί μια σύνθεση πολλών άλλων επιστημών, κυρίως της πληροφορικής, της ηλεκτρονικής και της μηχανολογίας. Τα ρομπότ είναι αυτόματες μηχανές με προγραμματισμένη συμπεριφορά, η χρήση των οποίων έχει σαν σκοπό την αντικατάσταση ή την βοήθεια του ανθρώπου στην ολοκλήρωση κάποιου έργου, τόσο σε φυσικό επίπεδο όσο και σε επίπεδο λήψης αποφάσεων.

Η ρομποτική επιστήμη έχει κάνει άλματα προόδου και σε συνδυασμό με άλλες επιστήμες έχει υλοποιήσει τεράστια τεχνολογικά επιτεύγματα. Στις μέρες μας διεξάγονται πολυάριθμες μελέτες στην ρομποτική και για συγκεκριμένες περιπτώσεις τα ρομπότ λειτουργούν παράλληλα με ανθρώπινους χειριστές. Για παράδειγμα, τα ρομπότ μπορούν να βοηθήσουν τους ανθρώπους στην οικιακή εξυπηρέτηση, την έρευνα και τη διάσωση, τη χειρουργική επέμβαση και την οδήγηση οχημάτων.

Όντως ερευνητικές δημοσιεύσεις (Manolis Chiou R. S., 2016) δείχνουν ότι συστήματα με μεταβλητή αυτονομία είναι πιο αποδοτικά ως προς τον χειρισμό τους. Σε αυτές τις περιπτώσεις η γνώση της ικανότητας και του φόρτου εργασίας ενός οδηγού είναι ζωτικής σημασίας για το σχεδιασμό τέτοιων συστημάτων. Σε ιδανικές συνθήκες, ο οδηγός και το όχημα πρέπει να γνωρίζουν την κατάσταση του άλλου, έτσι ώστε να επιτυγχάνεται υψηλό επίπεδο συνεργασίας μεταξύ ανθρώπου-μηχανής.

Υπάρχουν αρκετοί τρόποι αξιολόγησης ενός χειριστή ή ενός ρομπότ. Για παράδειγμα ο Boer (Boer, 2000) παρουσιάζει μια μέθοδο που μοντελοποιεί την κίνηση του “τιμονιού” του ρομπότ και χρησιμοποιεί τις προβλέψεις του μοντέλου του ώστε να τις συγκρίνει με την πραγματική οδήγηση ενός χειριστή. Στην παραπάνω έρευνα δεν γίνεται καμία αναφορά στα διαφορετικά μοντέλα που μπορούν να χρησιμοποιηθούν καθώς και στην μοντελοποίηση της γραμμικής ταχύτητας του ρομπότ.

Στην παρούσα πτυχιακή εργασία θα συλλεχθούν δεδομένα πραγματικών πειραμάτων από ρομπότ έρευνας και διάσωσης του πανεπιστημίου του Birmingham. Θα γίνει εκτενής ανάλυση των δεδομένων και αφού γίνει η επιλογή των καταλληλότερων θα δοκιμαστούν διαφορετικές μεθοδολογίες για την μοντελοποίησή τους. Τέλος θα παρουσιαστούν τα αποτελέσματα των μοντέλων, θα συγκριθούν και θα προταθούν τρόποι χρησιμοποίησής τους για περαιτέρω έρευνα.

1.1 Σκοπός και στόχοι της πτυχιακής εργασίας

Ο αρχικός σκοπός της παρούσας πτυχιακής είναι να γίνει ανάλυση της ανάγκης που υπάρχει στις μέρες μας, να αξιολογείται το φόρτο εργασίας καθώς και ο ίδιος ο χειριστής ρομποτικών συστημάτων και αυτόνομων οχημάτων. Έπειτα θα γίνει επεξήγηση μεθόδων μοντελοποίησης των προαναφερθέντων, συγκεκριμένα με αυτοπαλίνδρομο μοντέλο (AR) και μοντέλο αναπτύγματος Taylor (Taylor expansion), καθώς και υλοποίηση αυτών χρησιμοποιώντας κατάλληλα δεδομένα από πειράματα που έχουν διεξαχθεί στο Πανεπιστήμιο του Birmingham. Στόχος της εργασίας είναι η σύγκριση των μοντέλων και η δημιουργία συμπερασμάτων για την αποδοτικότητά τους.

1.2 Ρομποτική

Η ρομποτική είναι ένας διεπιστημονικός κλάδος όπου συνδυάζει επιστήμες όπως την επιστήμη της ηλεκτρονικής, της πληροφορικής και της μηχανικής. Η ρομποτική ασχολείται με το σχεδιασμό, την κατασκευή, τη λειτουργία και τη χρήση ρομπότ. Η λέξη πρωτοεμφανίστηκε σε θεατρικά έργα του 20ου αιώνα και εκτιμάται ότι προέρχεται από την τσεχική λέξη *robota* (ρομπότα) που σημαίνει καταναγκαστική εργασία.

Σύμφωνα με την Ένωση βιομηχανιών ρομποτικής ΗΠΑ, το ρομπότ είναι ένα προγραμματιζόμενο και πολυλειτουργικό μηχάνημα. Σκοπός του είναι κυρίως να μετακινεί υλικά, αντικείμενα, τεμάχια, εργαλεία που είτε είναι δύσκολο να μετακινήσουν άνθρωποι είτε το κάνουν με περίσσια ακρίβεια και ταχύτητα. Κύριο χαρακτηριστικό των ρομπότ είναι ότι είναι επαναπρογραμματιζόμενες μηχανές που μπορούν να πραγματοποιούν μια τεράστια ποικιλία εργασιών.

Παρά την επικρατούσα άποψη ότι η χρήση τους αποσκοπεί στην αντικατάσταση του ανθρώπου, στην πραγματικότητα τα ρομπότ δημιουργούνται με στόχο την βοήθεια του ανθρώπου στην εκτέλεση εργασιών που αφορά τόσο την φυσική εργασία, όσο και στην λήψη αποφάσεων. Ένα παράδειγμα είναι η εφαρμογή της ρομποτικής στην ιατρική. Προφανώς ο γιατρός δεν μπορεί να αντικατασταθεί από το ρομπότ, αλλά με την βοήθεια ενός ρομποτικού συστήματος μπορούν πλέον δύσκολες επεμβάσεις να γίνουν επεμβάσεις ρουτίνας ή να γίνουν επεμβάσεις από τεράστιες αποστάσεις. Ένα άλλο παράδειγμα είναι τα αυτόματα οχήματα. Αν και ακόμα γίνονται μελέτες με επίκεντρο τα αυτόματα οχήματα μερικές εταιρείες έχουν ήδη συμπεριλάβει στα οχήματα τους συστήματα λήψης αποφάσεων. Παράδειγμα η Tesla που δημιούργησε το 2014 το Tesla Autopilot (Εικόνα 1) όπου το έχει ορίσει ως προηγμένο σύστημα υποστήριξης οδηγού. Το σύστημα αυτό έχει την

δυνατότητα να κρατήσει μια σταθερή απόσταση ανάμεσα σε δύο λωρίδες, να προσαρμόσει την ταχύτητα οδήγησης με το προηγμένο προσαρμοστικό σύστημα ελέγχου ταχύτητας, αυτόνομου parking, δυνατότητα αυτόματης αλλαγής λωρίδων και δυνατότητα κλήσης του αυτοκινήτου από και προς γκαράζ ή χώρο στάθμευσης.



Εικόνα 1. Το Tesla Autopilot και το σύστημα αναγνώρισης περιβάλλοντος

1.3 Κατηγορίες ρομπότ

Τα ρομπότ έχουν την δυνατότητα να χωριστούν σε κατηγορίες ανάλογα με το επίπεδο της αυτονομίας τους. Κάποια ρομπότ έχουν άμεση αλληλεπίδραση με τον άνθρωπο. Σε αυτές τις περιπτώσεις πρόκειται κυρίως για τηλεχειριζόμενες μηχανές όπου ο άνθρωπος έχει κατά κόρον τον πλήρη έλεγχο των κινήσεων του ρομπότ. Για παράδειγμα τέτοια ρομπότ είναι τα χειρουργικά ρομπότ που απλά βοηθούν σε επίπεδο καλύτερου ελέγχου κίνησης κατά την διάρκεια της εγχείρησης. Μεσαίας αυτονομίας είναι τα ρομπότ υποβοήθησης του χειριστή.

Τέτοια ρομπότ έχουν τον χειριστή που επιτελεί καθήκοντα μεσαίου έως υψηλού επιπέδου, με το ρομπότ να υπολογίζει αυτόματα πώς να τα επιτύχει την διεργασία πιο αποδοτικά. Για παράδειγμα η προαναφερθείσα Tesla με το Tesla Autopilot (όπου θα αναφερθεί αναλυτικότερα παρακάτω) ή τα ρομποτικά συστήματα έρευνας και διάσωσης (Εικόνα 2). Τέλος υπάρχουν και τα πλήρως αυτόνομα ρομπότ που μπορούν να λειτουργούν χωρίς ανθρώπινη αλληλεπίδραση για παρατεταμένες χρονικές περιόδους. Τέτοια ρομπότ δεν χρειάζεται απαραίτητα να έχουν μεγαλύτερη “νοημοσύνη” ή πιο πολύπλοκες γνωστικές ικανότητες από ρομπότ των προηγούμενων κατηγοριών. Ένα παράδειγμα είναι

τα ρομπότ στις μονάδες συναρμολόγησης που είναι εντελώς αυτόνομα, αλλά λειτουργούν με σταθερό μοτίβο.



Εικόνα 2. Ρομπότ διάσωσης coyote III

Επίσης τα ρομπότ μπορούν να κατηγοριοποιηθούν με βάση την μορφή τους. Υπάρχουν πολλών ειδών ρομπότ τα οποία προσαρμόζονται ανάλογα με τις απαιτήσεις της κάθε εφαρμογής.

1.3.1 Κινούμενα Ρομπότ

Τα κινούμενα ρομπότ έχουν την δυνατότητα να κινούνται στο περιβάλλον και δεν έχουν κάποια σταθερή βάση. Τα κινούμενα ρομπότ μπορούν να είναι "αυτόνομα" (AMR - autonomous mobile robot) που σημαίνει ότι είναι σε θέση να περιηγούνται σε ένα άγνωστο περιβάλλον χωρίς να την ανάγκη καθοδήγησης. Εναλλακτικά, τα κινητά ρομπότ μπορούν να βασίζονται σε συσκευές τηλεχειρισμού ή καθοδήγησης που τους επιτρέπουν να ταξιδεύουν σε μια προκαθορισμένη διαδρομή (AGV - autonomous guided vehicle).

1.3.2 Ρομπότ Σταθερής Βάσης

Αυτά τα ρομπότ αυτά αποτελούνται από διαδοχικούς συνδέσμους που συνδέονται μέσω αρθρώσεων σχηματίζοντας μία κινηματική αλυσίδα. Η αλυσίδα αυτή έχει δύο άκρα. Το ένα είναι η βάση του και το άλλο είναι το άκρο το οποίο έχει συνδεδεμένο κάποιο εργαλείο. Αυτή είναι η δεύτερη πιο δημοφιλής μορφή ρομπότ, μετά τα ανθρωπόμορφα ρομπότ. Ένα παράδειγμα τέτοιου ρομπότ είναι τα βιομηχανικά ρομπότ όπως αυτό που απεικονίζεται στην παρακάτω εικόνα (Εικόνα 3)



Εικόνα 3. Γραμμή παραγωγής αποτελούμενη από ρομπότ σταθερής βάσης

1.3.3 AGVs (automated guided vehicle)

Ένα αυτόματο οδηγούμενο όχημα (AGV) είναι ένα μικρό, φορητό ρομπότ που ακολουθεί γραμμές, καλώδια στο πάτωμα, ραδιοκύματα, μαγνήτες ή λέιζερ για πλοήγηση. Αυτά τα ρομπότ είναι γνωστά και ως ακολουθητές γραμμής (line followers). Αυτά τα ρομπότ χρησιμοποιούνται κυρίως για μεταφορά βαρέων υλικών γύρω από ένα μεγάλο βιομηχανικό κτίριο, όπως ένα εργοστάσιο ή μια αποθήκη.

Βαδίζοντα Ρομπότ

Αυτά τα ρομπότ διαφέρουν σε ένα σημείο κλειδί από τα συμβατικά ρομπότ. Δεν χρησιμοποιούν ρόδες για την κίνηση τους αλλά μηχανικά πόδια. Αυτά τα ρομπότ δημιουργήθηκαν γιατί υπάρχει ανάγκη αποφυγής εμποδίων καθώς και αναρρίχησης σε ανώμαλα εδάφη. Ρομπότ αυτής της κατηγορίας είναι τα ανθρωπόμορφα, τα εξάποδα (Εικόνα 4) και όσα έχουν άκρα αντί για ρόδες.



Εικόνα 4. Εξάποδο ρομπότ

1.3.4 UUV και UAV

Πρόκειται για 2 είδη ρομπότ με πολύ διαφορετικούς σχεδιασμούς αλλά με πανομοιότυπους στόχους. Τα UAV (unmanned aerial vehicle) είναι μη επανδρωμένα εναέρια οχήματα και τα UUV (unmanned underwater vehicle) είναι μη επανδρωμένα υποβρύχια οχήματα. Και τα δύο ειδών ρομπότ λειτουργούν χωρίς μέσα τους να υπάρχει χειριστής, Σκοπός και των δύο είναι η παρακολούθηση και αναγνώριση περιοχών που δεν μπορεί να πάει άνθρωπος. Παράδειγμα τέτοιων ρομπότ είναι τα τετρακόπτερα ή αλλιώς drones.

1.4 Ερευνητικοί τομείς στην ρομποτική

Πολλά πανεπιστήμια ανά τον κόσμο διεξάγουν έρευνες πάνω σε ρομποτικές εφαρμογές. Αν όχι όλα, τα περισσότερα προγράμματα ρομποτικής έρευνας έχουν σαν στόχο την βελτίωση σε έναν ή και παραπάνω τεχνολογικούς τομείς όπως η τεχνητή νοημοσύνη, η ρομποτική όραση και συστήματα ελέγχου μικτής πρωτοβουλίας. Τεχνολογικές εξελίξεις σαν και αυτές θα οδηγήσουν σε βελτιώσεις και καινοτομίες στην

εφαρμογή της ρομποτικής στη βιομηχανία, την ιατρική, το στρατό, την εξερεύνηση του διαστήματος, την υποβρύχια εξερεύνηση και την προσωπική εξυπηρέτηση.

Ένας πολύ σημαντικός τομέας των ερευνών αυτών είναι η αξιολόγηση των χειριστών/οδηγών των ρομπότ. Το πρόβλημα που αποτέλεσε την ανάγκη δημιουργίας αυτού του ερευνητικού ενδιαφέροντος είναι η εύρεση του επιπέδου οδήγησης των χειριστών, την επίπτωση δευτερευόντων διαδικασιών στην οδήγηση του χειριστή και το πως αυτά μεταβάλλεται σε διαφορετικά περιβάλλοντα και καταστάσεις. Παρακάτω θα παρουσιαστούν δύο διαφορετικοί τομείς όπου η αξιολόγηση του χειριστή είναι κρίσιμο ζήτημα.

1.4.1 “Έξυπνα” Αμάξια (“Smart” cars)

Η ανάπτυξη των ημιαυτόνομων οχημάτων αποτέλεσε ένα από τα πρώτα βήματα για την αξιολόγηση των οδηγών. Είναι γνωστό ότι το πνευματικό φόρτο εργασίας παίζει αναπόσπαστο ρόλο στην αλληλεπίδραση του οδηγού με το όχημα και το περιβάλλον, ιδιαίτερα όταν η προσοχή του οδηγού πρέπει να χωριστεί σε πολλαπλές εργασίες. Μια από τις υποθέσεις πολλών ερευνητών, ήταν ότι αν ένα έξυπνο όχημα είναι δυνατόν να εκτιμήσει, σε πραγματικό χρόνο, το μέγεθος της προσοχής που ο οδηγός κατανέμει, τόσο στην οδήγηση όσο και σε άλλες διεργασίες, τότε αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν αναλόγως. Αυτές τις πληροφορίες μπορούν να διαχειριστούν από τα συστήματα ελέγχου και από τον οδηγό μέσω κάποιας διασύνδεσης ανθρώπου-μηχανής. Σαν αποτέλεσμα, οι πληροφορίες που διαβιβάζονται στον οδηγό έχουν μεγαλύτερη ποιότητα και τα επιτηρούμενα αυτόνομα συστήματα προσαρμόζονται, ώστε να λαμβάνουν υπόψη τους αυξημένους χρόνους απόκρισης του οδηγού.

Στις μέρες μας υπάρχουν πραγματικοί περισπασμοί κατά την διάρκεια της οδήγησης, εκ των οποίων οι περισσότεροι υπάρχουν για να βοηθούν τον οδηγό και όχι να τον επιβαρύνουν. Με τη ραγδαία ανάπτυξη συστημάτων πλοήγησης αυτοκινήτων (GPS), των τηλεφώνων και των πλέον ενσωματωμένων συσκευών ευφυούς συστήματος μεταφοράς, η ερευνητική κοινότητα έχει αρχίσει να αναρωτιέται αν είναι ασφαλείς οι χειρισμοί αυτού του είδους εξοπλισμού κατά την διάρκεια οδήγησης οχημάτων. Με σκοπό την εμπορευματοποίηση ή την βελτίωση των παραπάνω εξοπλισμών πολλές εταιρίες διεξάγουν έρευνες που προσδιορίζουν με ακρίβεια το πόσο μεγάλη επίπτωση έχει στον οδηγό η χρήση τέτοιων εξοπλισμών κατά την διάρκεια οδήγησης.

1.4.2 Ρομποτικά συστήματα εξερεύνησης και διάσωσης (SaR)

Σε καταστάσεις έκτακτης ανάγκης, όπως μια μεγάλη σεισμική καταστροφή ή κάποια βλάβη σε έναν πυρηνικό αντιδραστήρα ο άνθρωπος είναι πολύ δύσκολο να μπορέσει να βοηθήσει. Για ακριβώς αυτό τον λόγο, υπάρχουν ρομποτικά συστήματα εξερεύνησης και διάσωσης που μπορούν να χωρέσουν σε μέρη που ο άνθρωπος δεν μπορεί, να λειτουργούν σε περιβάλλοντα που ο άνθρωπος δεν μπορεί, να λειτουργούν συνεχόμενα χωρίς ύπνο, και το πιο σημαντικό είναι ότι μπορούν να αντικατασταθούν (Εικόνα 5).



Εικόνα 5. Παράδειγμα ρομπότ εξερεύνησης σε σεισμική καταστροφή στην Ιαπωνία

Κατά κόρον ο άνθρωπος είναι αυτός που χειρίζεται απευθείας τα ρομπότ που χρησιμοποιούνται για απαιτητικές εργασίες σε απρόβλεπτα περιβάλλοντα, όπως για παράδειγμα κατά τη διάρκεια του παροπλισμού πυρηνικών εγκαταστάσεων. Από την αντίθετη πλευρά τα αυτόνομα ρομπότ χρησιμοποιούνται κυρίως για καθήκοντα και περιβάλλοντα που είναι περιορισμένα, καθώς δεν είναι σε θέση να αντιμετωπίσουν με αποδοτικότητα το απρόβλεπτο περιβάλλον. Επιπρόσθετα, συχνά απαιτείται ένας ανθρώπινος παράγοντας ώστε να λαμβάνει κρίσιμες αποφάσεις κάτω από αβέβαιες ή ελλιπείς πληροφορίες που επέρχονται στο ρομπότ. Αυτά τα ζητήματα ενισχύουν την αρχή της ύπαρξης πάντα ανθρώπου για εφαρμογές κρίσιμης σημασίας για την ασφάλεια τόσο του ρομποτικού συστήματος, όσο και του ανθρώπου.

Ωστόσο, πρόσφατες έρευνες έχουν κάνει τα αυτόνομα ρομπότ πιο ικανά να αντιμετωπίσουν με αξιοπιστία τέτοιες απρόβλεπτες καταστάσεις και εργασίες. Κατά την διάσωση επιζώντων, μετά την τρομοκρατική επίθεση του Παγκόσμιου Κέντρου Εμπορίου της 9ης Σεπτεμβρίου στην Αμερική, πολλά ρομπότ τέθηκαν σε λειτουργία για να βοηθήσουν το υπάρχον δυναμικό διάσωσης. Τα ρομπότ είχαν σαν στόχο τη επιθεώρηση περιοχών κάτω από συντρίμμια και την εξερεύνηση περιορισμένων χώρων. Οι χειριστές ρομπότ λειτουργούσαν υπό συνθήκες καταπόνησης και επιδείνωσης της απόδοσης τους που οδήγησαν σε λάθη στη λειτουργία των ρομπότ.

Αυτή είναι μια από τις αιτίες που ερευνητές υποστηρίζουν ότι πρέπει να δοθεί προτεραιότητα στην ανάπτυξη ευφυών ρομπότ που μπορούν να βοηθήσουν ενεργά τον χειριστή. Όπως και στην περίπτωση των έξυπνων οχημάτων έτσι και στα ρομπότ διάσωσης, αν το ρομπότ έχει την δυνατότητα να εκτιμήσει την ικανότητα του χειριστή, η οποία όπως είπαμε είναι μεταβαλλόμενη λόγω κόπωσης ή απόσπασης προσοχής από δευτερεύουσες εργασίες, τότε αυτή η πληροφορία μπορεί να χρησιμοποιηθεί με διάφορους τρόπους. Για παράδειγμα, μπορεί ο χειριστής να ενημερώνεται για την απόδοση του μέσω μιας διεπαφής και να πράττει αναλόγως. Επίσης, το ρομπότ θα μπορούσε να λάβει αποφάσεις για την αποφυγή κρίσιμων σφαλμάτων, όπως συγκρούσεων, καθώς και καταστάσεων που καταλήγουν στην ολική καταστροφή του ρομπότ.

1.5 Μεταβλητή αυτονομία

Έχει οριστεί (Thomas B Sheridan, 1989) ότι όταν ένας άνθρωπος ενεργεί ως επιβλέπων σε κάποιο σύστημα, τότε αυτό ονομάζεται εποπτικός έλεγχος. Στον πραγματικό κόσμο, είναι μια πολύ δύσκολη δουλειά, να εξαλειφθεί εντελώς η εποπτεία του ανθρώπου. Έχει όμως αρχίσει να γίνεται αντιληπτό ότι δεν χρειάζεται. Πολλοί υποστηρίζουν ότι απαιτείται είναι ένας ανθρώπινος επιβλέπων / χειριστής με λιγότερες ευθύνες, τον οποίον θα τον υποβοηθούν τα ρομπότ που μπορούν να χρησιμοποιήσουν αυξημένη αυτονομία και να αναλάβουν πρωτοβουλίες.

Μια πολύ σημαντική έννοια για αυτό το έργο είναι η έννοια του επιπέδου αυτονομίας (LOA). Το επίπεδο αυτονομίας είναι ο βαθμός, που ένα ρομπότ θα λάβει αποφάσεις και θα ενεργήσει αυτόνομα. Για παράδειγμα τα επίπεδα αυτονομίας κυμαίνονται από πλήρη τηλεχειρισμό ενός ρομπότ από άνθρωπο, έως στο ρομπότ να είναι πλήρως αυτόνομο. Το αρχικό έργο για μια ιεραρχική στρωματοποίηση για τους LOAs αναπτύχθηκε από τους Sheridan και Verplank (Sheridan, 1978). Αυτή η διαστρωμάτωση βασίζεται σε τρία στοιχεία: τη λήψη αποφάσεων όσον αφορά την επιλογή ενεργειών, την εκτέλεση ενεργειών και τις

πληροφορίες που μοιράζεται ο υπολογιστής με τον άνθρωπο. Η διαστρωμάτωση που παρουσιάστηκε από τους Sheridan και Verplank (Sheridan, 1978) είναι η εξής:

1. Ο άνθρωπος κάνει όλη τη δουλειά μέχρι να επιτρέψει στον υπολογιστή να εφαρμόσει τις αποφάσεις του χειριστή.
2. Ο υπολογιστής βοηθάει στον καθορισμό των επιλογών.
3. Ο υπολογιστής βοηθά στον καθορισμό των επιλογών και προτείνει ένα, το οποίο ο άνθρωπος δεν χρειάζεται να ακολουθήσει.
4. Ο υπολογιστής επιλέγει τη δράση και ο άνθρωπος μπορεί ή όχι να το κάνει.
5. Ο υπολογιστής επιλέγει τη δράση και την εφαρμόζει, αν ο άνθρωπος εγκρίνει.
6. Ο υπολογιστής επιλέγει μια ενέργεια, ενημερώνει τον άνθρωπο σε αρκετό χρόνο ώστε ο άνθρωπος να μπορεί να τον σταματήσει.
7. Ο υπολογιστής κάνει όλη τη δουλειά και αναγκαστικά λέει στον άνθρωπο τι έκανε.
8. Ο υπολογιστής κάνει ολόκληρη τη δουλειά και λέει στον άνθρωπο τι έκανε μόνο αν ο άνθρωπος ρωτήσει.
9. Ο υπολογιστής κάνει όλη τη δουλειά και λέει στον άνθρωπο τι έπραξε και ο υπολογιστής αποφασίζει αν θα παρέχει πληροφορίες.
10. Ο υπολογιστής κάνει όλη τη δουλειά, αν αποφασίσει ότι πρέπει να γίνει, και αν αποφασίσει, τότε αποφασίζει εάν θα παρέχει πληροφορίες στον άνθρωπο.

Με αυτή την ταξινόμηση μπορούσαν να γίνουν πολλές έρευνες πάνω σε αυτό το θέμα. Το 1999 ο Endsley και συνεργάτες (Endsley, 1999) εκμοντέρνισε την παραπάνω ταξινόμηση η οποία είναι η εξής:

1. Παρακολούθηση του συστήματος.
2. Παραγωγή επιλογών για την επίτευξη στόχων.
3. Επιλογή/λήψη αποφάσεων.
4. Στρατηγική υλοποίησης των αποφάσεων μέσω μιας διεπαφής ανθρώπου μηχανής.

Το μοναδικό πρόβλημα με αυτές τις ταξινομήσεις είναι ότι μπορούν να χρησιμοποιηθούν για κάθε σύστημα ή τομέα που χρησιμοποιεί κάποια μορφή αυτοματισμού. Ο Beer και οι συνεργάτες του (Beer, 2014) εκσυγχρόνισαν ακόμα πιο πολύ αυτές τις ταξινομήσεις με στόχο να εφαρμόζονται συγκεκριμένα για διεπαφές ανθρώπου μηχανής.

Στην ταξινόμηση που πρότεινε ο Beer (Εικόνα 6) (Beer, 2014) υπάρχουν δύο βασικοί παράγοντες, το έργο και το περιβάλλον. Με βάση αυτή την ταξινόμηση, το σύστημα της πτυχιακής εμπίπτει στην κατηγορία της τηλελειτουργίας και όταν ο άνθρωπος το ελέγχει ή «μοιράζεται τον έλεγχο με ανθρώπινη πρωτοβουλία» όταν το ρομπότ βρίσκεται σε

κατάσταση αυτονομίας. Το σύστημα των πειραμάτων, συγκεκριμένα εμπίπτει στην κατηγορία "τηλεχείριση" και "κοινό έλεγχο με πρωτοβουλία ρομπότ" (Manolis Chiou G. B., 2017).

LORA	Sense	Plan	Act	Description
Manual	H	H	H	The human performs all aspects of the task including sensing the environment, generating plans/options/goals, and implementing processes.
Tele-operation	H/R	H	H/ R	The robot assists the human with action implementation. However, sensing and planning is allocated to the human. For example, a human may teleoperate a robot, but the human may choose to prompt the robot to assist with some aspects of a task (e.g., gripping objects).
Assisted Tele-operation	H/R	H	H/ R	The human assists with all aspects of the task. However, the robot senses the environment and chooses to intervene with task. For example, if the user navigates the robot too close to an obstacle, the robot will automatically steer to avoid collision.
Batch Processing	H/R	H	R	Both the human and robot monitor and sense the environment. The human, however, determines the goals and plans of the task. The robot then implements the task.
Decision Support	H/R	H/R	R	Both the human and robot sense the environment and generate a task plan. However, the human chooses the task plan and commands the robot to implement actions.
Shared Control With Human Initiative	H/R	H/R	R	The robot autonomously senses the environment, develops plans and goals, and implements actions. However, the human monitors the robot's progress and may intervene and influence the robot with new goals and plans if the robot is having difficulty.
Shared Control With Robot Initiative	H/R	H/R	R	The robot performs all aspects of the task (sense, plan, act). If the robot encounters difficulty, it can prompt the human for assistance in setting new goals and plans.
Executive Control	R	H/R	R	The human may give an abstract high-level goal (e.g., navigate in environment to a specified location). The robot autonomously senses environment, sets the plan, and implements action.
Supervisory Control	H/R	R	R	The robot performs all aspects of task, but the human continuously monitors the robot, environment, and task. The human has override capability and may set a new goal and plan. In this case, the autonomy would shift to executive control, shared control, or decision support.
Full Autonomy	R	R	R	The robot performs all aspects of a task autonomously without human intervention with sensing, planning, or implementing action.

Εικόνα 6. Η ταξινόμηση κατά Beer (Beer,2014)

Η αλληλεπίδραση ανθρώπου και ενός τέτοιου συστήματος είναι κατά κόρον ανεξερεύνητη. Μερικές μελέτες έχουν αξιολογήσει τον τρόπο που αλληλοεπιδρούν οι χειριστές με τις διεπαφές χρήστη (Yanco, 2004) (Baker, 2004). Άλλες μελέτες διερεύνησαν την αλληλεπίδραση του ανθρώπου με το ρομπότ για την ανταλλαγή πληροφοριών (Fong, 2003), αλλά δεν διερεύνησαν αρκετές θεματικές, όπως τα θέματα ελέγχου. Αναμφισβήτητα όμως μια πληροφορία που έχουν αρχίσει πολλοί ερευνητές να εξερευνούν είναι το φόρτο εργασίας του χειριστή και πως μπορούν να μεταφέρουν αυτή τη πληροφορία στο ρομπότ.

Για παράδειγμα ο Boer (Boer, 2000) πετυχαίνει να ορίσει το φόρτο εργασίας και να μεταφέρει αυτή την πληροφορία αποτελεσματικά σε ένα σύστημα.

1.6 Φόρτο εργασίας χειριστή

Κατά τον (Eric Farmer & Adam Brownson, 2003) το φόρτο εργασίας δεν είναι κάτι που μπορεί να υπολογιστεί από πριν. Στην ουσία είναι το αποτέλεσμα αλληλεπιδράσεων των απαιτήσεων μιας εργασίας, των συνθηκών/περιβάλλοντος κατά την διάρκεια της και των δεξιοτήτων του χειριστή (Hart, 1988). Ενίοτε ορίζεται από πλευράς παραγόντων όπως οι απαιτήσεις εργασίας ή η προσπάθεια που πρέπει να δαπανηθεί για την εκτέλεση του έργου.

Ο φόρτος εργασίας είναι ίσως και ο πιο σημαντικός λόγος της μείωσης της απόδοσης των χειριστών. Οι χειριστές ενός τέτοιου συστήματος κάνουν συχνά σφάλματα και σαν αποτέλεσμα η αξιολόγηση τους γίνεται πολύ σημαντική. Οι μέθοδοι για την μέτρηση του φόρτου είναι πολυάριθμες. Ουσιαστικά όμως μπορούν να ταξινομηθούν σε τρεις βασικές κατηγορίες (Eric Farmer & Adam Brownson, 2003) μέτρηση φυσιολογικού φόρτου, υποκειμενική μέτρηση φόρτου και μέτρηση επιδόσεων εργασιών.

Ο φυσιολογικός φόρτος είναι πολύ απλή έννοια. Είναι ο φόρτος ο οποίος προκαλεί κάποια φυσική σωματική αντίδραση του χειριστή. Για παράδειγμα μια μέτρηση φυσιολογικού φόρτου είναι η πίεση του ανθρώπου, η αναπνευστική δραστηριότητα, ο καρδιακός ρυθμός, η εγκεφαλική δραστηριότητα, η κίνηση των ματιών και η εφίδρωση των χεριών. Η απλότητα της έννοιας τελειώνει εκεί γιατί αυξάνεται κατά πολύ η πολυπλοκότητα των πειραμάτων καθώς και η ανάλυση των δεδομένων. Συγκεκριμένα τα δεδομένα όχι μόνο είναι δύσκολο να συγκεντρωθούν αλλά αν είναι εφικτό απαιτούν εξειδικευμένο εξοπλισμό. Γι' αυτό τον λόγο πολύ αποφεύγουν να χρησιμοποιούν τέτοια μέτρα για την αναγνώριση-αξιολόγηση του φόρτου.

Η υποκειμενική μέτρηση φόρτου συνήθως είναι ερωτηματολόγια. Μετά από κάθε πείραμα οι χειριστές απαντούν σε ένα ερωτηματολόγιο και καλούνται να αξιολογήσουν το φόρτο εργασίας του εκάστοτε πειράματος. Όπως και το όνομα αυτής της μέτρησης φόρτου έτσι και τα αποτελέσματα είναι εντελώς υποκειμενικά και κατά κύριο λόγο δεν παρέχουν αξιολόγηση φόρτου σε πραγματικό χρόνο.

Με την μέτρηση επιδόσεων είναι δυνατόν να μετρηθεί η επίδοση σε μια εργασία-πείραμα. Είναι γνωστό πως οι άνθρωποι δεν έχουν απεριόριστους γνωστικούς πόρους. Άρα είναι δυνατόν να υποθέσουμε ότι οι επιδόσεις του εκάστοτε χειριστή είναι ανάλογες, ή καλύτερη δυσανάλογες με το φόρτο εργασίας. Όσο μεγαλύτερο το φόρτο τόσο χαμηλότερη

απόδοση. Στην τρέχουσα πτυχιακή παρουσιάζονται τρόποι μοντελοποίησης της οδήγησης του συστήματος μας, με υψηλή επίδοση. Αυτό θα έχει σαν αποτέλεσμα την σύγκριση της εκάστοτε οδήγησης με τα μοντέλα που αναπτύσσονται παρακάτω έτσι ώστε να μπορεί μετέπειτα να χρησιμοποιηθεί το σφάλμα της σύγκρισης. Για παράδειγμα ο Boer (Boer, 2000) χρησιμοποιεί αυτό το σφάλμα για την ανάπτυξη ενός μέτρου αξιολόγησης του φόρτου εργασίας που ονομάζεται εντροπία.

Κεφάλαιο 2. Θεωρητική περιγραφή μοντέλων οδήγησης με υψηλή απόδοση

Στο προηγούμενο κεφάλαιο έγινε μια παρουσίαση της ρομποτικής και των σύγχρονων πεδίων έρευνας της. Επίσης παρουσιάστηκαν οι λόγοι που καθιστούν πολύ σημαντική την ανάγκη που προκύπτει για την αναγνώριση του φόρτου εργασίας. Σε αυτό το μέρος της πτυχιακής θα παρουσιαστεί το θεωρητικό υπόβαθρο των μοντέλων που χρησιμοποιήθηκαν για την μοντελοποίηση και πρόβλεψη οδήγησης με υψηλή απόδοση. Πριν από αυτό όμως, είναι σημαντικό να γίνει μια εισαγωγή στο πρόβλημα το οποίο επιλύουν αυτά τα μοντέλα.

2.1 Πρόβλημα μοντελοποίησης

Σε πάρα πολλές εφαρμογές είναι σύνηθες το πρόβλημα της αναγνώρισης της σχέσης μεταξύ δύο ή και περισσότερων μεταβλητών. Δεν είναι όμως και το μοναδικό πρόβλημα. Στις περισσότερες των περιπτώσεων επιστήμονες καλούνται όχι μόνο να βρουν την συσχέτιση μεταξύ των μεταβλητών αλλά και να την προσδιορίσουν. Και εδώ έρχονται τα μοντέλα. Είναι ο τρόπος με τον οποίο μπορούν να προσδιοριστούν με μεγάλη ακρίβεια οι σχέσεις μεταξύ των μεταβλητών.

Δυστυχώς όμως η εύρεση μιας τέτοιας εξίσωσης είναι πολύ δύσκολη δουλειά. Ένας λόγος που μέχρι και επιστήμονες δυσκολεύονται να δημιουργήσουν ένα τέτοιο μοντέλο είναι γιατί πρέπει να υπάρχει μια γνώση για την φύση της διεργασίας που επρόκειτο να μοντελοποιηθεί. Στις περισσότερες των περιπτώσεων μια απλή ανάλυση των δεδομένων-μεταβλητών αρκεί για να βρεθεί ο κατάλληλος τύπος μοντέλου που μπορεί να προσδιορίσει την σχέση μεταξύ των δεδομένων με επάρκεια.

2.2 Απλό και πολλαπλό μοντέλο κατάβαση δυναμικού (linear regression)

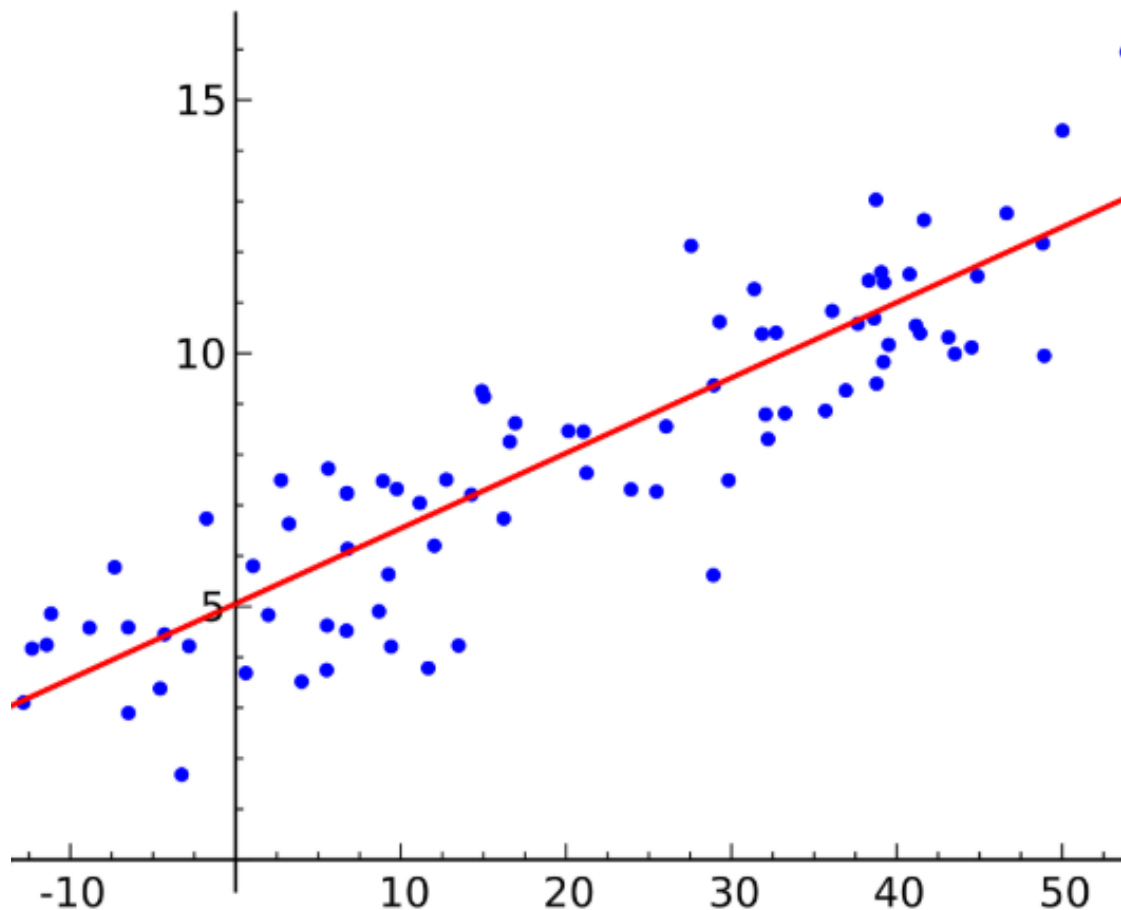
Σύμφωνα με την Γάζη (Γαζή, 2015), απλό γραμμικό μοντέλο κατάβαση δυναμικού έχει την μορφή:

$$Y_i = b_0 + b_1 X_i + e_i$$

Και το πολλαπλό μοντέλο έχει την μορφή:

$$Y_i = b_0 + b_1X_{1i} + b_2X_{2i} + b_3X_{3i} + \dots + e_i$$

Το Y είναι η τιμή που έχει η εξαρτημένη μεταβλητή την χρονική στιγμή i . Τα b_0, b_1 είναι απλοί συντελεστές. Το X είναι η τιμή της ανεξάρτητης μεταβλητής την χρονική στιγμή i και τέλος το e είναι ένα τυχαίο σφάλμα. Είναι εμφανές ότι το μοντέλο ονομάζεται γραμμικό λόγω των συντελεστών-παραμέτρους του οι οποίοι είναι γραμμικοί. Το μοντέλο είναι απλό διότι έχει μια ανεξάρτητη μεταβλητή. Αυτά τα μοντέλα λέγονται και μοντέλα πρώτης τάξης. Η μοναδική διαφορά μεταξύ των απλών/πολλαπλών μοντέλων είναι οι συντελεστές X . Στην περίπτωση του απλού μοντέλου η ανεξάρτητη μεταβλητή είναι μόνο μια ενώ στο πολλαπλό μοντέλο είναι πολλές.



Εικόνα 7. Παράδειγμα γραμμικής κατάβασης δυναμικού

Οι ομοιότητες με τα αυτοπαλίνδρομα μοντέλα είναι πολλές. Για παράδειγμα το πως μπορούν να βρεθούν οι παράμετροι και το πως γίνεται εκτίμηση μετά την εύρεση των κατάλληλων μεταβλητών.

Συγκεκριμένα, ο αλγόριθμος μηχανικής μάθησης του γραμμικού παλίνδρομου μοντέλου είναι απλός, όμως αρχικά πρέπει να αναφερθούν κάποιες βασικές έννοιες και σχέσεις. Η

υπόθεση (hypothesis) είναι η σχέση την οποία χρησιμοποιούμε σαν βασικό χαρακτηριστικό του μοντέλου $h_i = b_0 + b_1X$. Η βασική ιδέα είναι να βρεθούν τα κατάλληλα b_0, b_1 ώστε η υπόθεση (το h) να είναι όσο το δυνατόν πιο κοντά στις πραγματικές τιμές Y . Αυτό μπορεί να μεταφραστεί και σαν ένα πρόβλημα βελτιστοποίησης, ή καλύτερα ελαχιστοποίησης. Ο σκοπός είναι να ελαχιστοποιηθεί η παρακάτω συνάρτηση, που ονομάζεται και συνάρτηση Κόστους. Μπορεί να παρατηρηθεί ότι η συνάρτηση κόστους «κρύβει» μέσα της την μέθοδο ελαχίστων τετραγώνων.

$$J(b_0, b_1) = \frac{1}{2m} \sum_{i=1}^m (hb(X(i)) - Y(i))^2$$

Στην πραγματικότητα όσο η συνάρτηση κόστους μειώνεται τόσο αλλάζουν και οι συντελεστές, άρα και η υπόθεση προσεγγίζει πιο πολύ την πραγματικότητα.

Μέχρι στιγμής είναι ξεκάθαρος ο τρόπος με τον οποίο η συνάρτηση κόστους μειώνετε αλλά δεν έχει αναφερθεί τίποτα ακόμα για το πως αλλάζουν οι συντελεστές. Η βαθμιαία κατάβαση (Gradient Descent) είναι μια μέθοδος ανανέωσης των παραμέτρων. Η βαθμιαία κατάβαση είναι το μέγεθος της αλλαγής των παραμέτρων b και έχει το τύπο:

$$b_j = b_j - a \frac{\partial}{\partial b_j} J(b_j) \quad \text{for } 0 \leq j \leq n, \text{ όπου } n \text{ ο αριθμός των συντελεστών } b$$

Αφού βρεθούν οι συντελεστές, μπορεί να γίνει πρόβλεψη τιμών της μεταβλητής Y χρησιμοποιώντας την μεταβλητή X και τους συντελεστές που αποδίδουν το μικρότερο σφάλμα.

2.3 Αυτοπαλίνδρομα μοντέλα

Μια χρονοσειρά Y ονομάζεται αυτοπαλίνδρομη χρονοσειρά τάξης p και συμβολίζεται ως $AR(p)$. Στην ουσία είναι Μαρκοβιανές διαδικασίες με εξάρτηση υψηλότερης τάξης από μια καθυστέρηση (lag-1) στον ενιαίο χώρο κατάστασης. Ένα αυτοπαλίνδρομο μοντέλο τάξης p έχει την εξής μορφή:

$$Y_t = a_0 + a_1 Y_{t-1} + \dots + a_p Y_{t-p} + e_t$$

Η τάξη p στην ουσία είναι το μέγεθος της υστέρησης ή καθυστέρησης της συνάρτησης. Για παράδειγμα ένα δεύτερης τάξης αυτοπαλίνδρομο μοντέλο έχει την μορφή:

$$Y_t = a_0 + a_1 Y_{t-1} + e_t$$

Στην παραπάνω περίπτωση διαπιστώνουμε το τι σημαίνει αυτοπαλίνδρομο. Ο όρος προέρχεται από το γεγονός ότι η συνάρτηση είναι μοντέλο παλινδρόμησης. Οι μεταβλητές ή παλινδρομητές είναι οι τιμές της μεταβλητής Y με χρονική καθυστέρηση ανάλογη της τάξης. Στην παραπάνω σχέση το e είναι λευκός θόρυβος. Για να βρεθούν οι καλύτεροι

συντελεστές του μοντέλου μια συνηθισμένη μέθοδος που χρησιμοποιείται αρκετά είναι των ελαχίστων τετραγώνων.

$$\sum_{i=1}^n (Y_i - Y_{pred_i})^2$$

Όπου Y_i είναι η πραγματική τιμή της μεταβλητής στον χρόνο $t=i$ και το Y_{pred_i} είναι η εκτιμώμενη τιμή του μοντέλου. Ουσιαστικά είναι μια γραμμική σχέση με p ανεξάρτητες μεταξύ τους μεταβλητές.

$$Y_t = \delta + a_1 Y_{t-1} + \dots + a_p Y_{t-p} + e_t$$

Για την εκτίμηση κάποιας συγκεκριμένης τιμής Y_{t+1} δεν μένει παρά μόνο να χρησιμοποιηθούν οι προηγούμενες τιμές της μεταβλητής Y καθώς και οι σωστές παράμετροι του μοντέλου ώστε να υπολογιστεί. Να σημειωθεί ότι με τον όρο «σωστοί παράμετροι» εννοούνται οι παράμετροι οι οποίοι αποδίδουν το μικρότερο τετραγωνικό σφάλμα.

2.4 Ανάπτυγμα Τέιλορ

Στα μαθηματικά η σειρά Τέιλορ είναι μια αναπαράσταση μια συνάρτησης ως ένα άπειρο άθροισμα πολλών όρων και έχει την μορφή (Dyer, 2002):

$$f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots$$

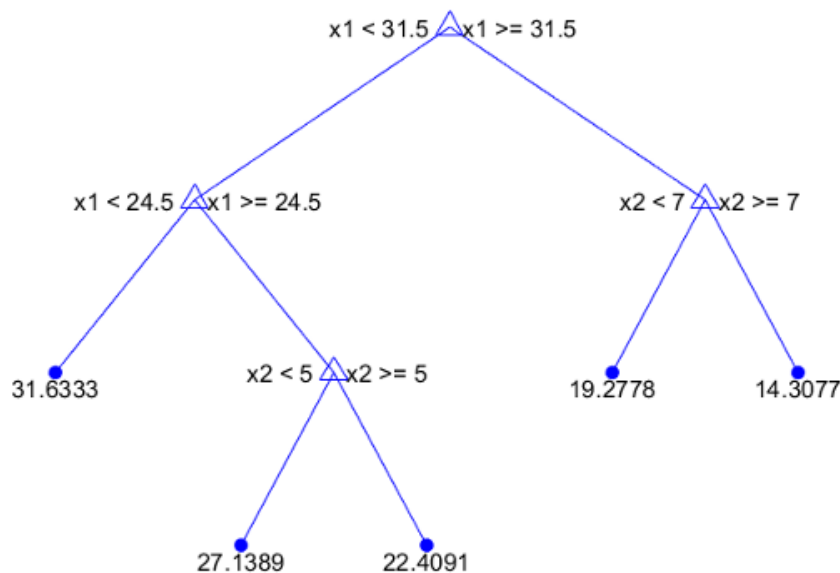
Στην πραγματικότητα μια συνάρτηση (όπως στην περίπτωση μας η συνάρτηση της στροφής προς τα δεξιά του ρομπότ από το -1, μέχρι το +1 που είναι η αριστερή στροφή), μπορεί να προσεγγιστεί χρησιμοποιώντας περιορισμένους όρους της σειράς Τέιλορ. Ο Nakayama και οι συνεργάτες του (Okihiko Nakayama, 1999) μοντελοποίησαν το στρίψιμο του τιμονιού του πειραματικού τους αμαξιού με ανάπτυγμα Τέιλορ της μορφής:

$$\theta_p(n) = \theta(n-1) + (\theta(n-1) - \theta(n-2)) + 1/2 ((\theta(n-1) - \theta(n-2)) - (\theta(n-2) - \theta(n-3)))$$

2.5 Μοντέλο Random Forest

Για να γίνει κατανοητός ο αλγόριθμος random forest πρέπει πρώτα να εξηγηθούν τα δέντρα αποφάσεων. Σύμφωνα με τον Gilles Louppe (Louppe, 2014) ένα δέντρο είναι ένα σύνολο γράφων $G = (V, E)$ στον οποίο οποιοδήποτε δυο κόμβοι συνδέονται με ακριβώς μία διαδρομή. Επίσης αναφέρει ότι ένα ριζωμένο δέντρο έχει, έναν από τους κόμβους, ορίσει ως ρίζα. Επιπλέον, όλα τα δέντρα έχουν την ρίζα στην κορυφή και τα σημεία που βρίσκονται κάτω ονομάζονται φύλλα (Εικόνα 8). Συνεχίζοντας με κάποιους βασικούς

ορισμούς για τα δέντρα αποφάσεων, τα φύλλα ονομάζονται οι τελικοί κόμβοι, δηλαδή οι κόμβοι οι οποίοι δεν χωρίζονται σε άλλους κόμβους. Τέλος ο κόμβος, ο οποίος χωρίζεται σε υπο-κόμβους, ονομάζεται γονικός κόμβος υπο-κόμβων ενώ οι υπο-κόμβοι είναι τα παιδιά του γονικού κόμβου.



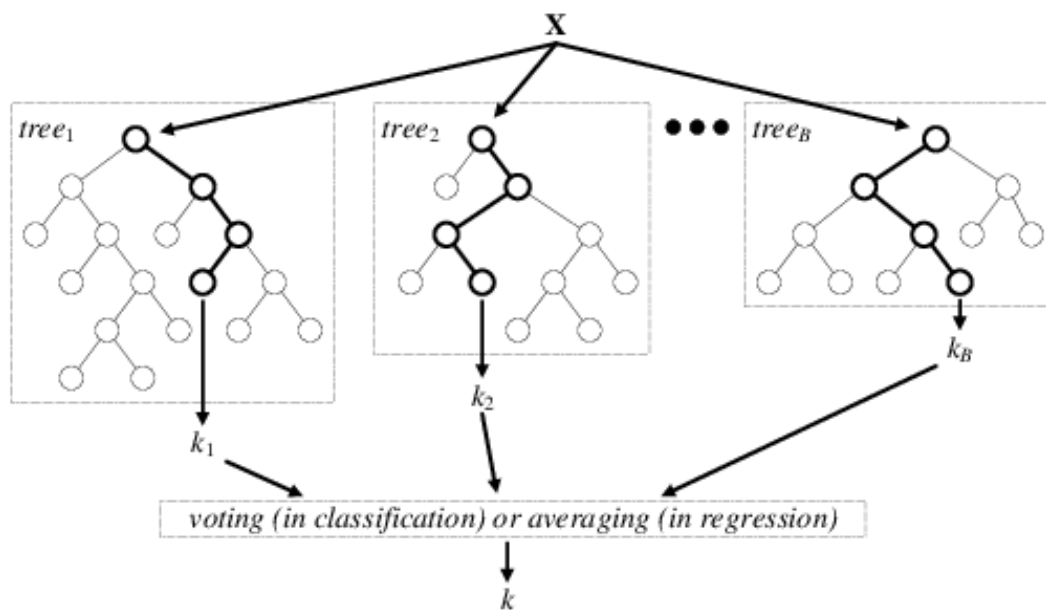
Εικόνα 8. Παράδειγμα δέντρου αποφάσεων

Όταν αναπτύσσονται τα μοντέλα δέντρων αποφάσεων συνήθως ακολουθούνται δύο βήματα, η εγκαθίδρυση και το κλάδεμα (Induction and Pruning). Το βήμα της εγκαθίδρυσης είναι το βασικό «χτίσιμο» του μοντέλου. Αρχικά χωρίζονται τα δεδομένα με τέτοιο τρόπο έτσι ώστε, τα υποσύνολα του χωριζόμενου συνόλου να εμπεριέχουν όσο το δυνατόν περισσότερη πληροφορία από μια κατηγορία δεδομένων. Η επιλογή των δεδομένων που θα χωριστούν καθώς και στο που θα χωριστούν γίνεται μέσω κάποιου αλγορίθμου βελτιστοποίησης όπως για παράδειγμα με την μέθοδο του μέσου τετραγωνικού σφάλματος. Συγκεκριμένα ο τύπος είναι $E = \sum(Yr - Yp)^2$. Όπου Yr είναι οι πραγματικές μετρήσεις και το Yp είναι η προβλεπόμενη τιμή. Αφού αθροίσουμε όλα τα δείγματα των δεδομένων μας μπορούμε να υπολογίσουμε το σφάλμα.

Τα δέντρα αποφάσεων είναι πολύ επιρρεπή στο να προσαρμόζονται πιο πολύ στα δεδομένα που εισάγονται και όχι να γενικεύουν τα αποτελέσματά τους. Για τον λόγο αυτόν βοηθάει το κλάδεμα. Πρακτικά το κλάδεμα είναι η αφαίρεση κόμβων του δέντρου έχουν μικρό αντίκτυπο στην συνάρτηση κόστους.

Ο αλγόριθμος random forest απαρτίζεται από τα δέντρα αποφάσεων. Όπως φαίνεται και στην εικόνα Regression, ο αλγόριθμος απαρτίζεται από διαφορετικά δέντρα

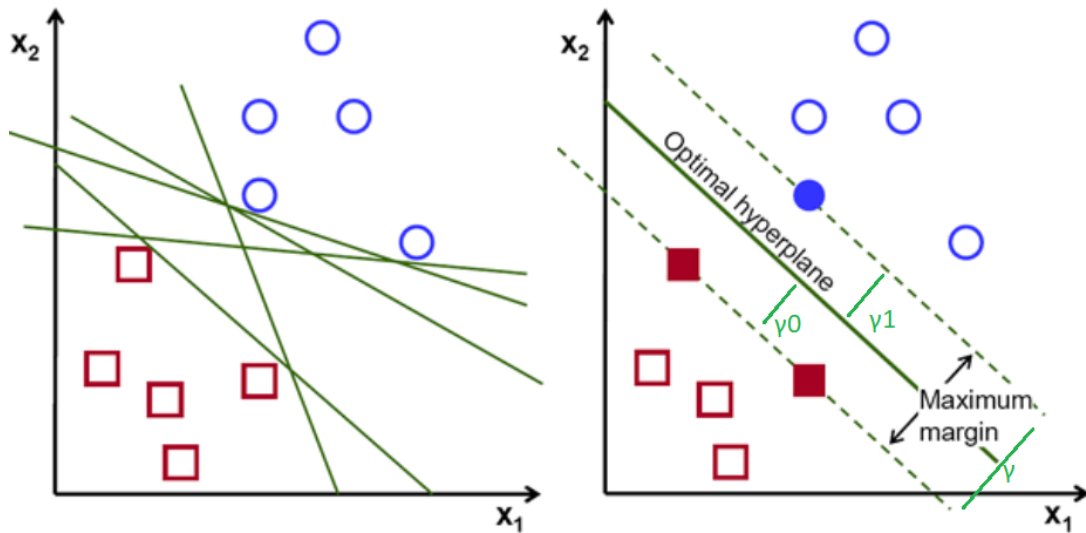
αποφάσεων τα οποία «τρέχουν» παράλληλα χωρίς να υπάρχει κάποιου είδους επικοινωνίας μεταξύ των δέντρων.



Εικόνα 9. Παράδειγμα μοντέλου Random Forest

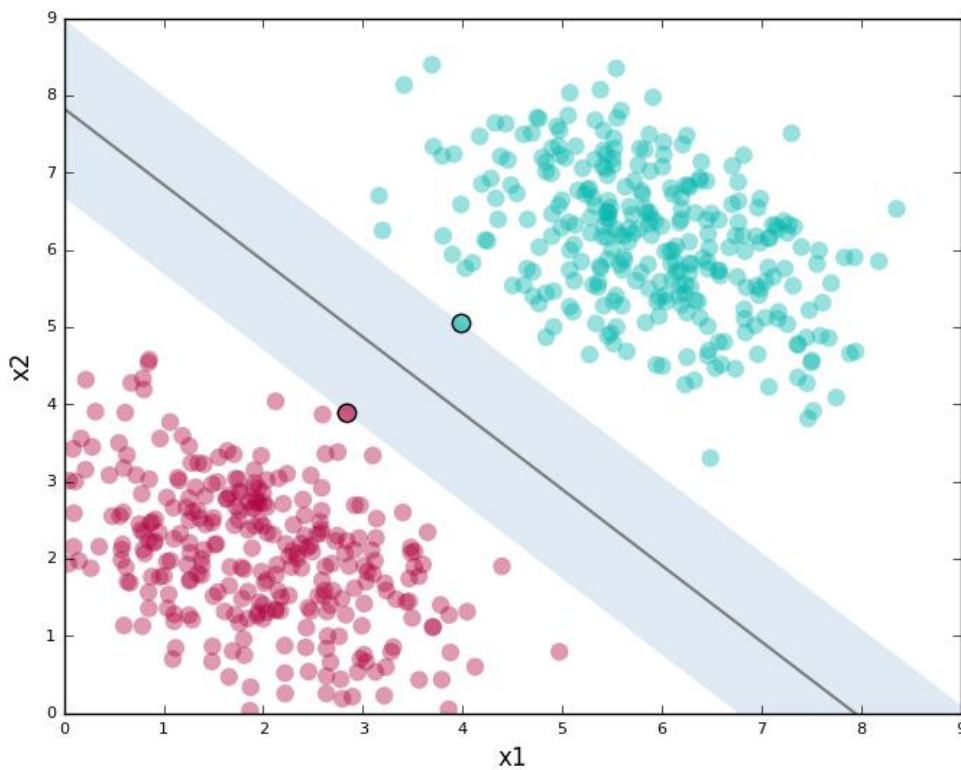
2.6 Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Regression)

Οι μηχανές διανυσμάτων υποστήριξης (SVM) είναι ένας αλγόριθμος μηχανικής μάθησης όπου η βασική ιδέα είναι ο διαχωρισμός των δεδομένων βρίσκοντας την βέλτιστη τομή ή υπερεπίπεδο (hyperplane) μεταξύ των δεδομένων (Scholkopf, 2003) (Εικόνα 10). Είναι λογικό το συμπέρασμα ότι μπορούν να υπάρξουν πολλές ευθείες που μπορούν να διαχωρίσουν δύο γραμμικά διαχωρίσιμες ομάδες δεδομένων (κλάσεις). Με βάση τον Διαμαντάρα (Διαμαντάρας, 2007) για μια διαχωριστική ευθεία, το περιθώριο ταξινόμησης γ είναι το άθροισμα των ελαχίστων αποστάσεων γ_0, γ_1 .



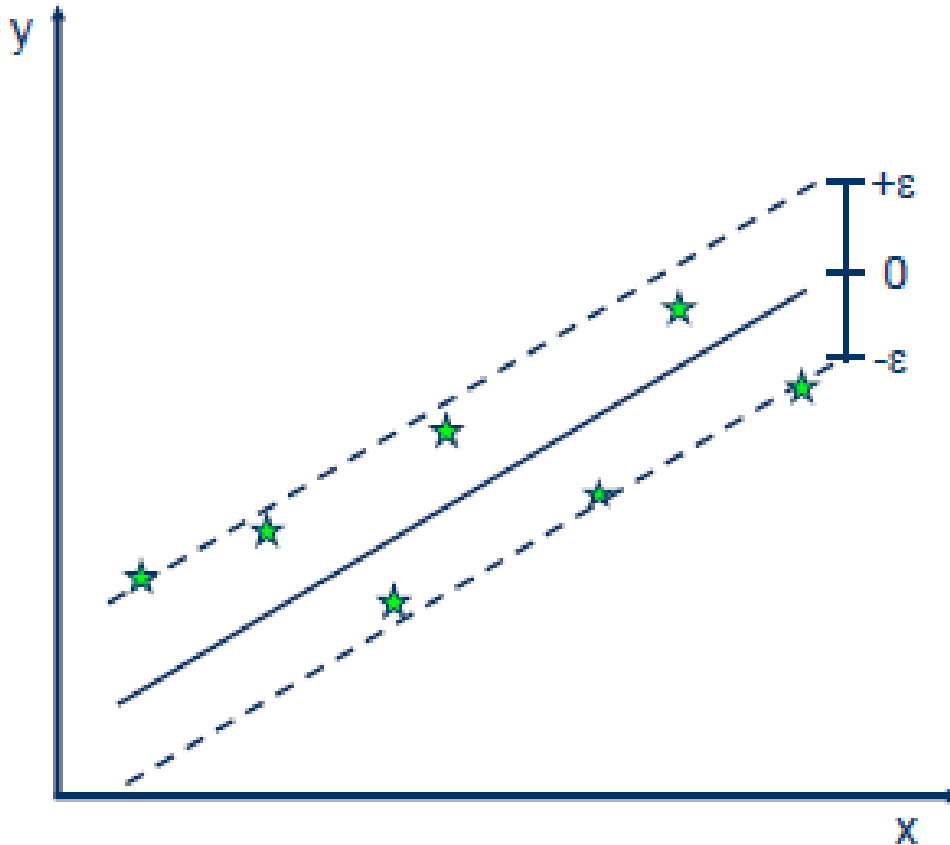
Εικόνα 10. Διαφορές σε μεθόδους ταξινόμησης

Όσα σημεία είναι κοντά στο μέγιστο περιθώριο ονομάζονται διανύσματα υποστήριξης (support vectors) (Εικόνα 10) και είναι πολύ σημαντικά για τον αλγόριθμο γιατί η τοποθεσία τους παρέχει πληροφορία στον αλγόριθμο που χωρίς αυτήν η τομή θα άλλαζε δραστικά. Όταν χρειάζεται να προσεγγίσουμε γραμμικά δεδομένα, ορίζεται ένα περιθώριο ανοχής epsilon (Support Vector Machine - Regression (SVR), n.d.)(Εικόνα 12).



Εικόνα 11. Παράδειγμα SVM

Σε αυτό που αποσκοπεί αυτή η μέθοδος είναι να βρεθεί η κατάλληλη τιμή του epsilon, από την αρχική τομή, έτσι ώστε τα σημεία δεδομένων που βρίσκονται πλησιέστερα στη τομή και τα διανύσματα υποστήριξης να βρίσκονται εντός αυτής της οριακής γραμμής (boundaries).



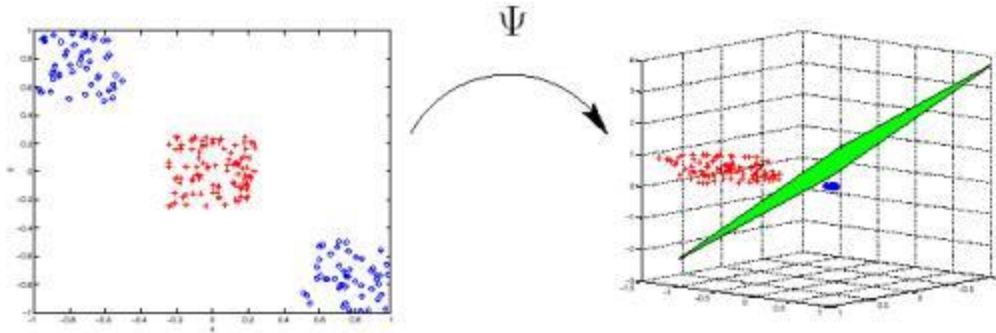
Εικόνα 12. Παράδειγμα SVM με το μέγιστο περιθώριο epsilon

Για να βρεθεί το μοντέλο που προσεγγίζει μια καλύτερα μια ομάδα δεδομένων το πρόβλημα για ακόμα μια φορά ανάγεται σε πρόβλημα ελαχιστοποίησης. Έστω ότι η τομή έχει την μορφή $y = ax + b$. Ελαχιστοποιώντας την συνάρτηση κόστους, με μόνους περιορισμούς το $y - wx - b \leq \epsilon$ και $wx + b - y \leq \epsilon$.

Συνάρτηση κόστους :

$$J = C \sum_{i=1}^m [y(i) (-\log h_w(x(i))) + (1 - y(i)) (-\log (1 - h_w(x(i))))] + \frac{1}{2} \sum_{j=1}^n w_j^2$$

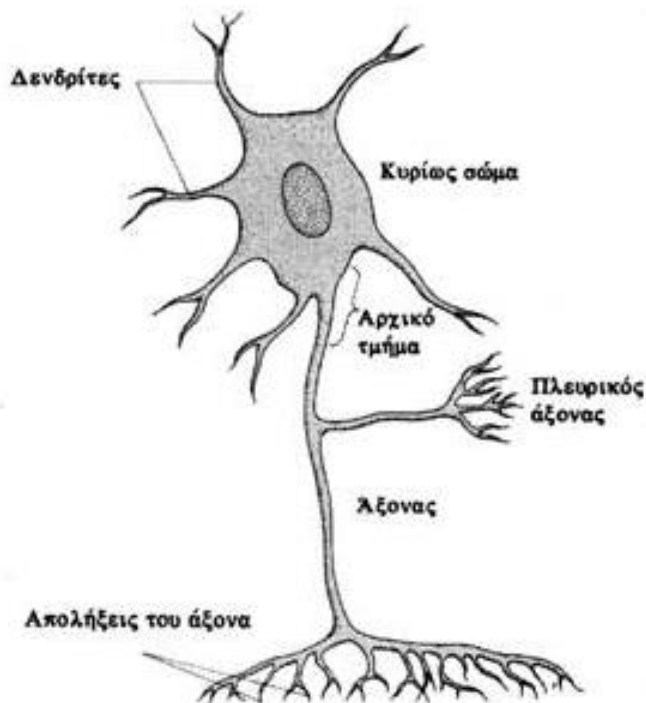
Σε περίπτωση που το υπερεπίπεδο δεν μπορεί να προσεγγιστεί με ευθεία γραμμή τότε με ειδικές μετατροπές τα δεδομένα που τα ανάγει σε χώρο μεγαλύτερων διαστάσεων ώστε να είναι δυνατή η εκτέλεση του γραμμικού διαχωρισμού (Εικόνα 12).



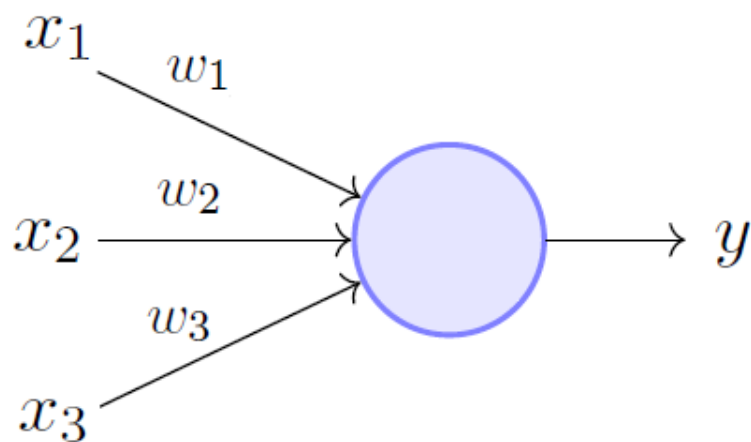
Εικόνα 13. Αναγωγή SVM σε παραπάνω διάσταση

2.7 Νευρωνικά Δίκτυα

Τα νευρωνικά δίκτυα κατά τον Martin (Anthony, 1997) είναι μοντέλα τα οποία είναι βασισμένα στο πως εκτιμάται ότι δουλεύει ο εγκέφαλος. Στην πραγματικότητα χρησιμοποιούν μοντέλα νευρώνων στην απλούστερη μορφή τους ώστε να μπορούν να διατηρήσουν σε κάποιον βαθμό τα πραγματικά χαρακτηριστικά των νευρώνων (Εικόνα 13). Οι νευρώνες αποτελούνται από τρία βασικά τμήματα, τους δενδρίτες, τον άξονα και τις απολήξεις του άξονα ή αλλιώς συνάψεις. Κατά αυτό το μοντέλο έχουν δημιουργηθεί και οι τεχνητοί νευρώνες (Εικόνα 14).



Εικόνα 14. Νευρώνας



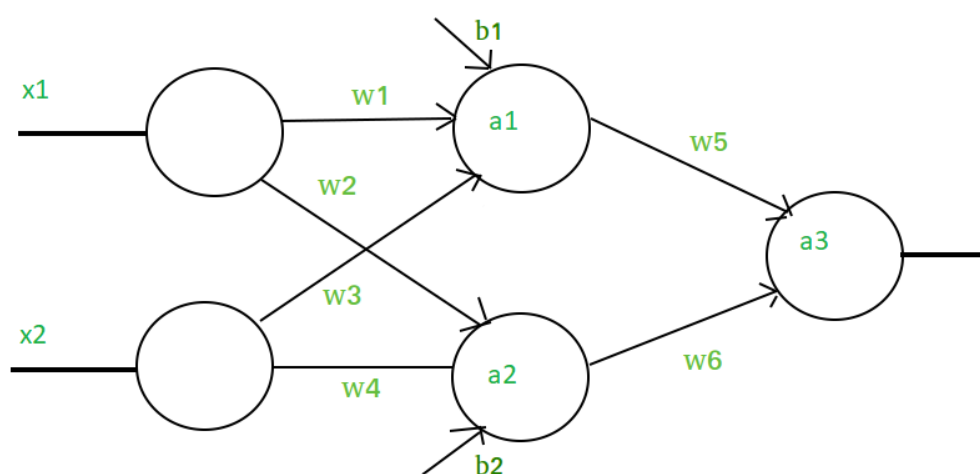
Εικόνα 15. Τεχνητός Νευρώνας

Τα τεχνητά νευρωνικά δίκτυα δημιουργήθηκαν λόγο κάποιας ανάγκης. Τα παραπάνω μοντέλα που παρουσιάστηκαν είχαν ένα κοινό χαρακτηριστικό. Αν δεν είναι εκ φύσεως γραμμικά τα μοντέλα τότε με κάποιες μικρές αλγοριθμικές/μαθηματικές μετατροπές τα μοντέλα γίνονται γραμμικά. Η πραγματικότητα όμως διαφέρει. Στην φύση ελάχιστα φαινόμενα παρουσιάζουν γραμμικές συμπεριφορές. Αυτή ακριβώς είναι η ανάγκη που κάλυψαν επιστήμονες οι οποίοι δημιούργησαν αυτό το είδος μοντέλων. Στην πραγματικότητα με τις μη γραμμικές υποθέσεις, και κατά συνέπεια στα μη γραμμικά μοντέλα είναι πολύ δύσκολη η εξερεύνηση του χώρου των δεδομένων. Ειδικά όταν εμφανίζετε ένα πρόβλημα με πολλά δεδομένα και πολλές μεταβλητές, τότε η πολυπλοκότητα του μοντέλου αυξάνεται δραματικά. Παρακάτω θα γίνει μια απλή περιγραφή των δικτύων Perceptron πολλών στρωμάτων.

2.7.1 Νευρώνες και Αρχιτεκτονική Νευρωνικών Δικτύων

Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από αρκετά μέρη, αλλά με πρωτεύον δομικό στοιχείο τους τεχνητούς νευρώνες. Οι νευρώνες σε ένα δίκτυο έχουν τρεις τύπους, τους νευρώνες εισόδου, εξόδου και υπολογισμού. Οι νευρώνες εισόδου σε αρκετά βασικά μοντέλα δεν κάνουν κάποιον υπολογισμό, απλά δέχονται την πληροφορία και την μεταβιβάζουν σε άλλους νευρώνες συνήθως στους υπολογιστικούς, ανάλογα την μορφή ή αλλιώς, αρχιτεκτονική του δικτύου. Οι υπολογιστικοί νευρώνες λαμβάνουν εισόδους, πολλαπλασιάζόμενες με κάποιο συνοπτικό βάρος, από άλλους νευρώνες και εισάγει το άθροισμα τους σε μια συνάρτηση ενεργοποίησης. Η έξοδος της συνάρτησης είναι ουσιαστικά και η έξοδος του νευρώνα. Τέλος οι νευρώνες εξόδου μετά από τα αθροίσματα που πολλαπλασιάζονται με τα κατάλληλα βάρη τα διοχετεύει στο περιβάλλον αφού περάσουν πρώτα και αυτά από κάποια συνάρτηση.

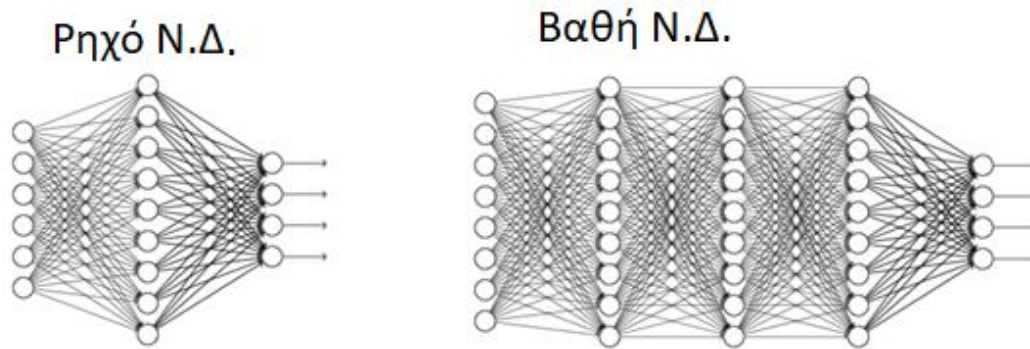
Στις παραπάνω παραγράφους αναφέρθηκε κάτι για αρχιτεκτονική. Η αρχιτεκτονική ενός νευρωνικού δικτύου είναι το πως είναι τοποθετημένοι και συνδεδεμένοι οι νευρώνες του δικτύου. Η αρχιτεκτονική ενός νευρωνικού έχει 3 επίπεδα. Το επίπεδο εισόδου, το κρυφό επίπεδο και το επίπεδο εξόδου. Στην εικόνα 15 το επίπεδο εισόδου (αριστερά) με τους νευρώνες οι οποίοι δέχονται πληροφορία από το περιβάλλον και παρέχουν αυτή τη πληροφορία στο κρυφό επίπεδο (ενδιάμεσο), οι οποίοι με την σειρά τους μεταδίδουν την επεξεργασμένη πλέον πληροφορία στο επίπεδο εξόδου που αποτελείται σε αυτή την αρχιτεκτονική από ένα μόνο νευρώνα (δεξιά). Στην εικόνα 15 υπάρχουν και κάποιες μεταβλητές b_1, b_2 . Αυτές οι μεταβλητές ονομάζονται προκαταλήψεις (bias) οι οποίες υπάρχουν για να προσεγγίζουν καλύτερα τα μοντέλα σε διαφορετικές διεργασίες.



Εικόνα 16. Αρχιτεκτονική δικτύου

Τα νευρωνικά δίκτυα χρησιμοποιούνται για να προσεγγίσουν μοντέλα υψηλών διαστάσεων αλλά ταυτόχρονα μπορούν να χρησιμοποιηθούν και σαν απλά μοντέλα κατάβασης δυναμικού. Τα νευρωνικά δίκτυα τα οποία έχουν πολλά κρυφά επίπεδα ονομάζονται βαθιά νευρωνικά δίκτυα. Όπως γίνεται κατανοητό από το όνομα αυτά τα δίκτυα έχουν πολλά κρυφά επίπεδα και συνήθως αυτά τα επίπεδα αποτελούνται από μικρό αριθμό νευρώνων. Από την αντίθετη πλευρά υπάρχουν τα ρηχά νευρωνικά δίκτυα τα οποία αποτελούνται από ένα μόνο κρυφό επίπεδο αλλά ο αριθμός των νευρώνων είναι κατά πολύ μεγαλύτερος από ότι θα ήταν σε ένα μέσο κρυφό επίπεδο σε ένα βαθύ νευρωνικό δίκτυο. Έχουν γίνει πολλές έρευνες στο πιο είδος νευρωνικού δικτύου είναι το καλύτερο. Σύμφωνα με τον (Schindler, 2017) τα ρηχά νευρωνικά έδειξαν καλύτερα αποτελέσματα με μικρό μέγεθος δεδομένων εκπαίδευσης, πράγμα που είναι λογικό εφόσον ο συνολικός αριθμός νευρώνων, άρα και παραμέτρων, είναι μικρότερος. Από την αντίθετη πλευρά τα βαθιά νευρωνικά δίκτυα είχαν καλύτερη απόδοση με μεγάλο αριθμό δεδομένων εκπαίδευσης

αλλά πέρα από αυτό τα αποτελέσματα τους δεν μπορούσαν να δώσουν μια σίγουρη απάντηση για το πιο νευρωνικό δίκτυο είναι το καταλληλότερο.



Εικόνα 17. Διαφορές μεταξύ ρηχού και βαθιού νευρωνικού δικτύου

Όπως αναφέρει και ο Διαμαντάρας (Διαμαντάρας, 2007) με την σιγμοειδή συνάρτηση ενεργοποίησης μπορούν να ξεπεραστούν βασικά προβλήματα. Συγκεκριμένα αναφέρει ότι η σιγμοειδής συνάρτηση βοηθάει κυρίως γιατί πολλές μέθοδοι βελτιστοποίησης χρησιμοποιούν παραγώγους. Η συνάρτηση που χρησιμοποιείται είναι η ακόλουθη:

$$h(x) = \frac{1}{(1 + e^{-wx})}$$

$$w = \begin{bmatrix} w1 \\ \dots \\ wn \end{bmatrix}, x = \begin{bmatrix} x1 \\ \dots \\ xk \end{bmatrix}, \text{ όπου } n \text{ οι αριθμοί των επιπέδων και}$$

k οι αριθμοί των εισόδων

Για παράδειγμα στην εικόνα 15 στο κρυφό επίπεδο (ενδιάμεσο) στον πρώτο νευρώνα η έξοδος του θα είναι :

$$a_1 = h(w_1x_1 + w_3x_2 + b_1)$$

Αφού υπολογιστούν όλοι οι νευρώνες του κρυφού στρώματος τότε το τελευταίο στρώμα, δηλαδή ο νευρώνας εξόδου θα έχει την μορφή:

$$a_3 = h(w_5a_1 + w_6a_2) \text{ και } h_w = a_3$$

Αυτή η διαδικασία λέγεται ανάκληση δικτύου ή εμπρόσθιο πέρασμα (feed forward propagation) και είναι το πρώτο βήμα για την εκπαίδευση ενός δικτύου.

2.7.2 Συνάρτηση Κόστους

Η συνάρτηση κόστους των νευρωνικών δικτύων είναι η παρακάτω:

$$J(w) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^k [y_k(i) \log((h_w(x(i)))k) + (1 - y_k(i)) \log(1 - (h_w(x(i)))k)]$$

$$+ \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{sL} \sum_{j=1}^{sL+1} (w_j, i(L)^2)$$

- L : Ο συνολικός αριθμός των επιπέδων του νευρωνικού
- sL : Ο αριθμός των νευρώνων χωρίς τις προκαταλήψεις (bias)
- m : Ο αριθμός των δεδομένων που εισάγονται στο νευρωνικό
- k : Ο αριθμός των νευρώνων εξόδου

Στο πρώτο σκέλος της εξίσωσης πριν τις αγκύλες υπάρχει ένα άθροισμα. Σκοπός του είναι να εισάγει και τους νευρώνες εξόδου στην συνάρτηση κόστους. Επίσης το άθροισμα των βαρών πρέπει να συμπεριλάβει τον αριθμό των νευρώνων στο τρέχον επίπεδο μαζί με την προκατάληψη ($sL+1$) και τους νευρώνες του επόμενου επιπέδου χωρίς την προκατάληψη (sL).

2.7.3 Εκπαίδευση και πρόβλεψη νευρωνικών δικτύων

Τα νευρωνικά δίκτυα έχουν την ικανότητα την μάθησης. Στην πράξη αυτό σημαίνει ότι με την χρήση της συνάρτησης κόστους βρίσκονται τα κατάλληλα βάρη (w) έτσι ώστε η έξοδος να προσεγγίζει το φαινόμενο γίνεται η προσπάθεια να μοντελοποιηθεί. Η εκπαίδευση του γίνεται μέσω δύο αλγορίθμων που ονομάζονται εμπρόσθιο πέρασμα (feedforward propagation) και οπίσθιο πέρασμα (Back propagation). Η συνάρτηση αυτή χρησιμοποιείται για να ελαχιστοποιεί την συνάρτηση κόστους. Εν αντίθεση με το εμπρόσθιο πέρασμα που ξεκινάει από τις εισόδους του δικτύου (κεφ. 2.7.1), το οπίσθιο πέρασμα ξεκινάει από την έξοδο. Συγκρίνει την πραγματική/αναμενόμενη έξοδο με την προβλεπόμενη έξοδο του νευρωνικού, και υπολογίζει τα σφάλματα κάθε νευρώνα. Κατ' αυτόν το τρόπο μπορούν τα συνοπτικά βάρη να αλλάζουν κάθε φορά που συγκρίνετε η έξοδος του νευρωνικού με την πραγματική έξοδο. Παρακάτω παρουσιάζεται ο αλγόριθμος του οπίσθιου περάσματος και για ευκολία θα χρησιμοποιηθούν οι ίδιες μεταβλητές από τα παραπάνω κεφάλαια:

1. Δεδομένα εκπαίδευσης $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$
2. $\Delta_{i,j}^{(l)} = 0$
3. Για i από 1 μέχρι m
4. $a^{(1)} = x^{(i)}$

5. Κάνε εμπρόσθιο πέρασμα και υπολόγισε όλα τα $a^{(l)}$ για $l = 0, \dots, L$
6. Υπολόγισε $\delta^{(L)} = a^{(L)} - y^{(i)}$ (Οπίσθιο πέρασμα)
7. Υπολόγισε $\delta^{(L-1)}, \delta^{(L-2)}, \dots, \delta^{(2)}$ με τον τύπο

$$\delta^{(l)} = (w^{(l+1)T} \cdot \delta^{(l+1)} \cdot a^{(l)} \cdot (1 - a^{(l)}))$$

Τα δ είναι το σφάλμα του νευρώνα j στο επίπεδο l του δικτύου

8. Ενημέρωσε τα βάρη με τους τύπους:

$$\Delta_{ij}^{(l)} = \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$$

$$\frac{\partial}{\partial w_{ij}^{(l)}} J(w) = D_{ij}^{(l)}$$

$$D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)} \text{ αν } j = 0 \text{ και } D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)} + \lambda w_{ij}^{(l)} \text{ αν } j \neq 0$$

Αφού αυτός ο αλγόριθμος έχει γίνει πολλές φορές και οι παράμετροι έχουν πάρει τις πιο κατάλληλες τιμές τότε το νευρωνικό δίκτυο μπορεί να κάνει προβλέψεις. Η μέθοδος πρόβλεψης είναι στην ουσία το εμπρόσθιο πέρασμα μόνο που τώρα δεν θα υπάρχει η ανάγκη του οπίσθιου περάσματος διότι δεν θα είναι αναγκαία η αλλαγή των παραμέτρων.

Κεφάλαιο 3. Ρομποτικό λειτουργικό σύστημα (ROS)

3.1 Εισαγωγή στο ROS

Για αυτή την πτυχιακή καθώς και για τα πειράματα που χρησιμοποιήθηκαν, η εργασία έγινε με την βοήθεια του ρομποτικού λειτουργικού συστήματος που από εδώ και πέρα θα αναφέρεται ως ROS. Παρακάτω θα αναφερθούν κάποια βασικά χαρακτηριστικά του ROS όπως έχουν οριστεί επισήμως (Ros Wiki, 2018) καθώς και από πληροφορίες από την ιστοσελίδα (What is ROS?, n.d.).

Το ROS, παρά το όνομά του, δεν είναι λειτουργικό σύστημα. Ούτε είναι πραγματικά ένα λειτουργικό πλαίσιο (framework). Το ROS μοιάζει περισσότερο ένα μεσαίο λογισμικό, κάτι σαν ένα "framework" χαμηλού επιπέδου που βασίζεται σε ένα υπάρχον λειτουργικό σύστημα. Το υπάρχον λειτουργικό σύστημα είναι περιβάλλον με βάση τα Linux και συνήθως επιλέγεται η διανομή των Ubuntu. Ο στόχος ROS είναι να παρέχει ένα πρότυπο για την ανάπτυξη λογισμικού ρομποτικής, που μπορεί να χρησιμοποιηθεί σε οποιοδήποτε ρομπότ. Το ROS αποτελείται κυρίως από 2 πράγματα, τον πυρήνα που παρέχει άφθονα εργαλεία επικοινωνίας και ένα σύνολο από βιβλιοθήκες. Ουσιαστικά είναι ένα λογισμικό "μεσάζοντας", που είναι υπεύθυνο για το χειρισμό της επικοινωνίας μεταξύ προγραμμάτων σε ένα κατανεμημένο σύστημα.

Στο ROS μπορεί να γίνει ανάπτυξη λογισμικού με δύο τρόπους. Αρχικά υπάρχει η δυνατότητα ανάπτυξης μιας βάσης κώδικα με όλα όσα συνθέτουν και εκτελούνται σε ένα μπλοκ κώδικα. Επίσης υπάρχει η δυνατότητα της ανάπτυξης υπο-προγραμμάτων, ένα για κάθε εργασία της αναπτυχθείσας εφαρμογής. Οι περισσότεροι χρήστες του ROS στις μέρες μας πιστεύουν δικαίως ότι η δεύτερη επιλογή είναι η καλύτερη για την ανάπτυξη ενός λογισμικού ρομποτικής. Αυτό συμβαίνει γιατί πρέπει ο κάθε προγραμματιστής να είναι σε θέση να αναπτύξει ένα μέρος της εφαρμογής του και να το εκτελέσει χωρίς την πλήρη εφαρμογή. Αυτό βοηθάει ιδιαίτερα και στην αποφυγή ή εύρεση λαθών στον κώδικα. Για παράδειγμα αν σε μια εταιρία γίνει η χρήση του ROS για την ανάπτυξη ενός ρομποτικού άκρου με σκοπό την υποβοήθηση γιατρών σε ιατρικές επεμβάσεις θα ήταν καταστροφικό να είναι ολόκληρος ο κώδικας της εφαρμογής σε μόνο ένα αρχείο. Αυτό θα καθιστούσε πρακτικά αδύνατη την αναγνώριση του κώδικα από τρίτους πράγμα που δεν γίνεται να επιτραπεί για εφαρμογές που δεν μπορούν να αναπτυχθούν μόνο από ένα άτομο. Επίσης

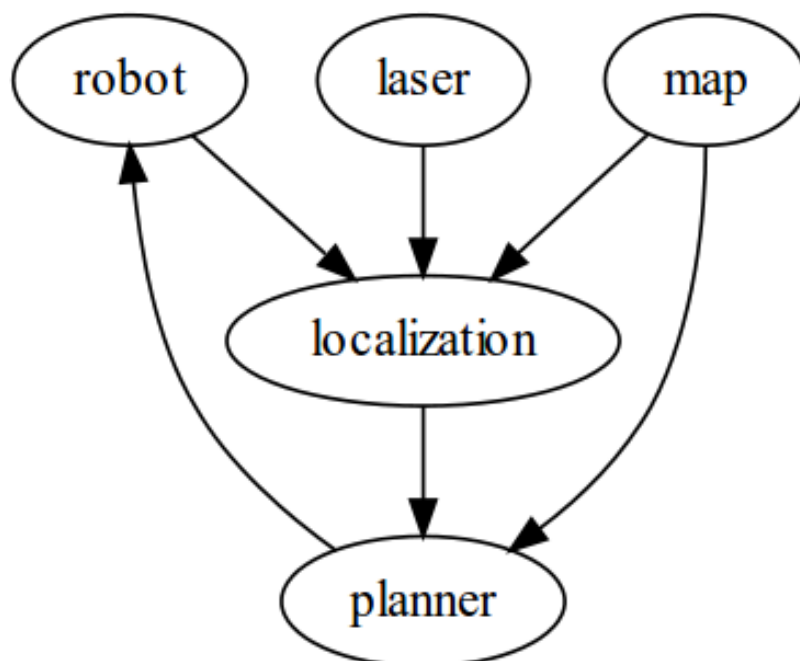
κάθε σφάλμα ή αβλεψία στον κώδικα θα είναι δύσκολο να βρεθεί εάν δεν έχει τμηματοποιηθεί ο κώδικας.

3.2 Πυρήνας ROS και εργαλεία επικοινωνίας

Το ROS παρέχει στους προγραμματιστές την δυνατότητα να διαχωρίζουν εύκολα τη βάση κώδικα σε πακέτα που περιέχουν μικρά προγράμματα, που ονομάζονται κόμβοι (nodes). Αυτά τα προγράμματα μπορούν να επικοινωνήσουν μεταξύ τους με 3 βασικά εργαλεία επικοινωνίας (O'Kane, 2014).

3.2.1 Θέματα (Topics)

Τα θέματα είναι ονομαζόμενοι δίαυλοι πάνω από τα οποία οι κόμβοι ανταλλάσσουν μηνύματα. Τα θέματα έχουν ανώνυμη σημασιολογία δημοσίευσης/εγγραφής (publish/subscribe), η οποία αποσυνδέει την παραγωγή πληροφοριών από την κατανάλωσή της. Πολλές φορές είναι πιο εύκολο να οριστούν τα θέματα σαν νέφη δεδομένων. Σε γενικές γραμμές, οι κόμβοι δεν γνωρίζουν με ποιον επικοινωνούν. Αντ' αυτού, οι κόμβοι που ενδιαφέρονται για δεδομένα μπορούν να εγγραφούν στο σχετικό θέμα.

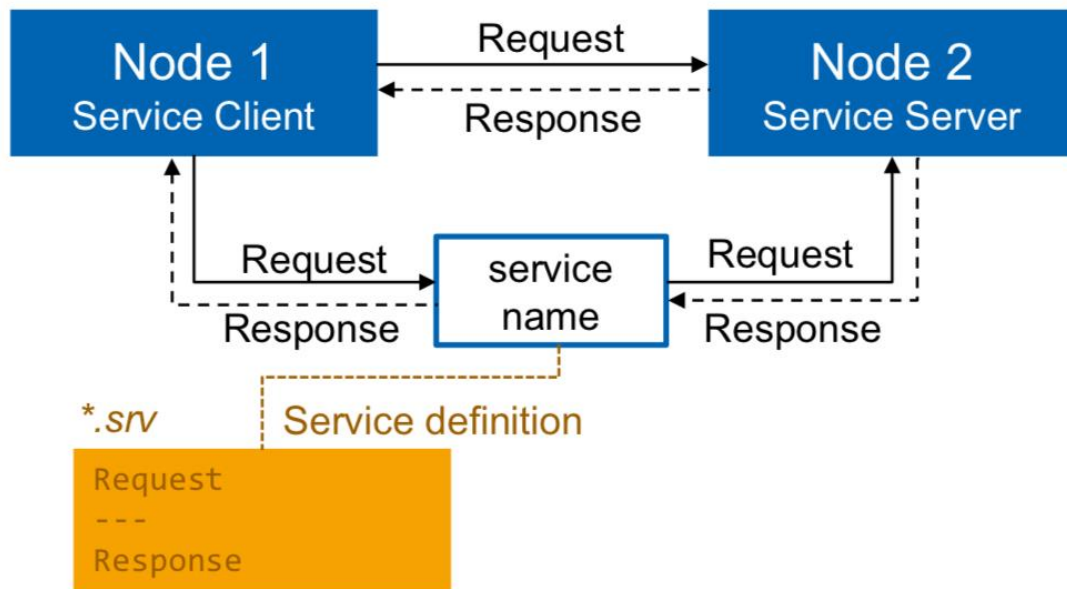


Εικόνα 18. Παράδειγμα των θεμάτων μιας εφαρμογής στο ROS

Οι κόμβοι που παράγουν δεδομένα δημοσιεύονται στο θέμα στο οποίο έχουν προγραμματιστεί να δημοσιεύουν. Μπορούν να υπάρχουν πολλοί εκδότες και συνδρομητές σε ένα θέμα (publishers/subscribers). Ένα παράδειγμα είναι η παρακολούθηση απόστασης ενός οχήματος με το περιβάλλον του. Ο κόμβος παρακολούθησης αυτού του οχήματος θα στείλει μια ροή δεδομένων με τη απόσταση. Τώρα, οποιοσδήποτε άλλος κόμβος μπορεί να εγγραφεί σε αυτό το θέμα και να συλλέξει τα δεδομένα της απόστασης. Όπως για παράδειγμα ο κόμβος των κινητήρων, ή ένας υποθετικός κόμβος ο οποίος αναγράφει σε μια συγκεκριμένη διεπαφή ανθρώπου μηχανής προειδοποιητικά μηνύματα.

3.2.2 Υπηρεσίες (Services)

Οι υπηρεσίες επιτρέπουν την δημιουργία μιας απλής σύγχρονης επικοινωνίας πελάτη / διακομιστή μεταξύ κόμβων. Δηλαδή ορίζεται ένας κόμβος σαν πελάτης και ένας άλλος σαν διακομιστής και αυτή η επικοινωνία παραμένει άθικτη κατά την διάρκεια της λειτουργίας. Πολύ χρήσιμο για αλλαγές ρυθμίσεων σε ένα ρομπότ ή την ζήτηση μιας συγκεκριμένης ενέργειας, για παράδειγμα να ενεργοποιηθεί μια συγκεκριμένη λειτουργία, να ζητηθούν συγκεκριμένα δεδομένα κ.λπ.



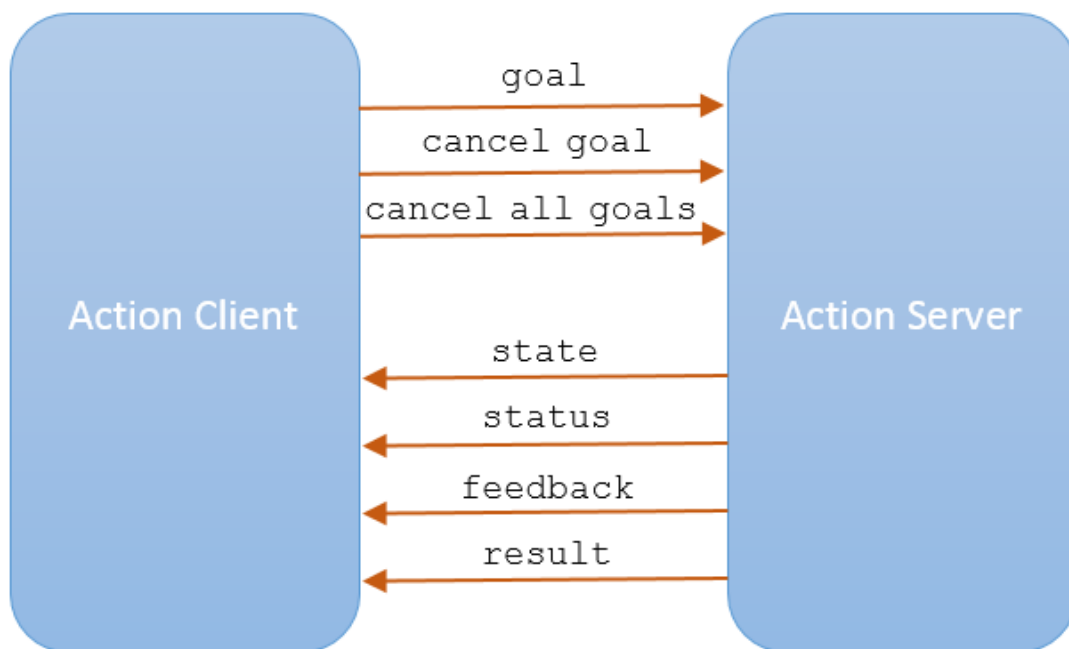
Εικόνα 19. Παράδειγμα υπηρεσίας στο ROS

3.2.3 Ενέργειες (Actions)

Οι ενέργειες βασίζονται στην πραγματικότητα σε θέματα. Υπάρχουν για να παρέχουν μια ασύγχρονη αρχιτεκτονική πελάτη / διακομιστή, όπου ο πελάτης μπορεί να στείλει ένα αίτημα που διαρκούν πολύ (π.χ. ζητώντας να μετακινηθεί ένα ρομπότ σε μια

νέα θέση). Ο πελάτης μπορεί να παρακολουθεί ασύγχρονα την κατάσταση του διακομιστή και να ακυρώσει την αίτηση ανά πάσα στιγμή.

Για να χρησιμοποιηθούν αυτά τα εργαλεία επικοινωνίας, απλά θα πρέπει να κληθούν οι κατάλληλες βιβλιοθήκες στον κώδικα και να οριστούν συγκεκριμένα μηνύματα (ros messages). Στο περιβάλλον του ROS υπάρχει η δυνατότητα να δημιουργηθούν ορισμένες γενικές ρυθμίσεις, που ονομάζονται παράμετροι (ros parameters), έτσι ώστε πολλοί κόμβοι να έχουν πρόσβαση στο ίδιο σύνολο ρυθμίσεων από οπουδήποτε στην εφαρμογή ρομποτικής. Τέλος η πλήρης εφαρμογή θα μπορεί να “τρέξει” με ένα μόνο αρχείο XML, που ονομάζεται αρχείο έναρξης (ros launch file). Το ROS περιλαμβάνει επίσης ένα πλήρες σύστημα καταγραφής, το οποίο είναι αρκετά εύκολο στη χρήση.



Εικόνα 20. Παράδειγμα ενεργειών στο ROS

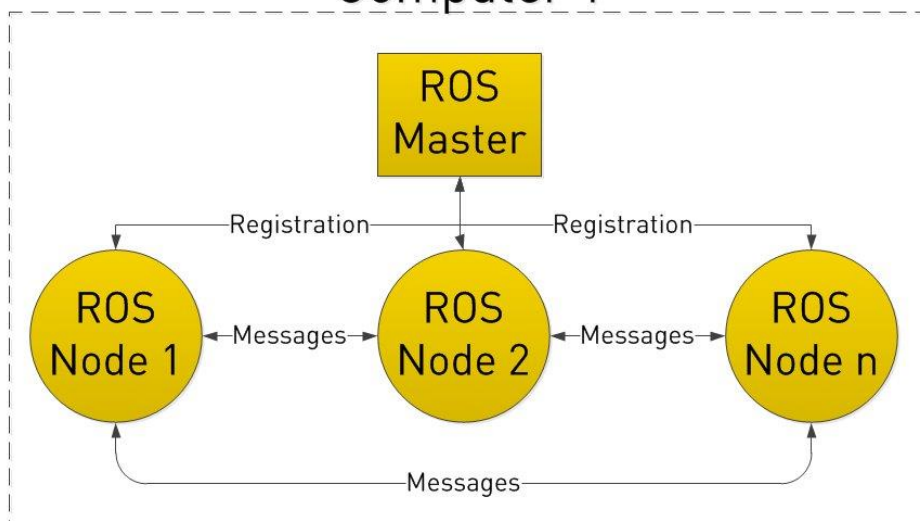
3.3 Ανάπτυξη προγραμμάτων στο περιβάλλον του ROS

Στο ROS το λογισμικό αναπτύσσεται κυρίως σε 2 γλώσσες: C ++ και Python. Αυτές είναι συχνά οι πιο προτιμώμενες και χρησιμοποιούμενες γλώσσες κατά την ανάπτυξη ρομποτικών εφαρμογών. Με την κλήση της βιβλιοθήκης roscpp μπορεί να αναπτυχθεί λογισμικό σε C ++ και καλώντας τη βιβλιοθήκη rospy μπορεί να γραφτεί κώδικας στη γλώσσα της Python. Υπάρχουν επίσης ορισμένες βιβλιοθήκες που μπορούν να χρησιμοποιηθούν σαν μια γέφυρα με άλλες γλώσσες, όπως η rosjava για την Java, και roslibjs ή rosnodajs για την JavaScript.

Ένα ακόμα υπέρ του ROS είναι ότι τα υποπρογράμματα που αναπτύσσονται (κόμβοι) μπορούν να γραφτούν σε οποιαδήποτε γλώσσα. Κάθε εφαρμογή μπορεί έτσι να έχει έναν κόμβο γραμμένο σε Python που επικοινωνεί με έναν κόμβο γραμμένο σε C ++. Αυτό είναι δυνατό επειδή το επίπεδο επικοινωνίας είναι κάτω από το "επίπεδο γλώσσας". Αυτό είναι δυνατόν εξαιτίας του ROS και γιατί χρησιμοποιεί τυπικές υποδοχές TCP / IP για επικοινωνία μεταξύ κόμβων.

ROS

Computer 1



Εικόνα 21. Σχήμα επικοινωνίας μεταξύ διαφορετικών κόμβων

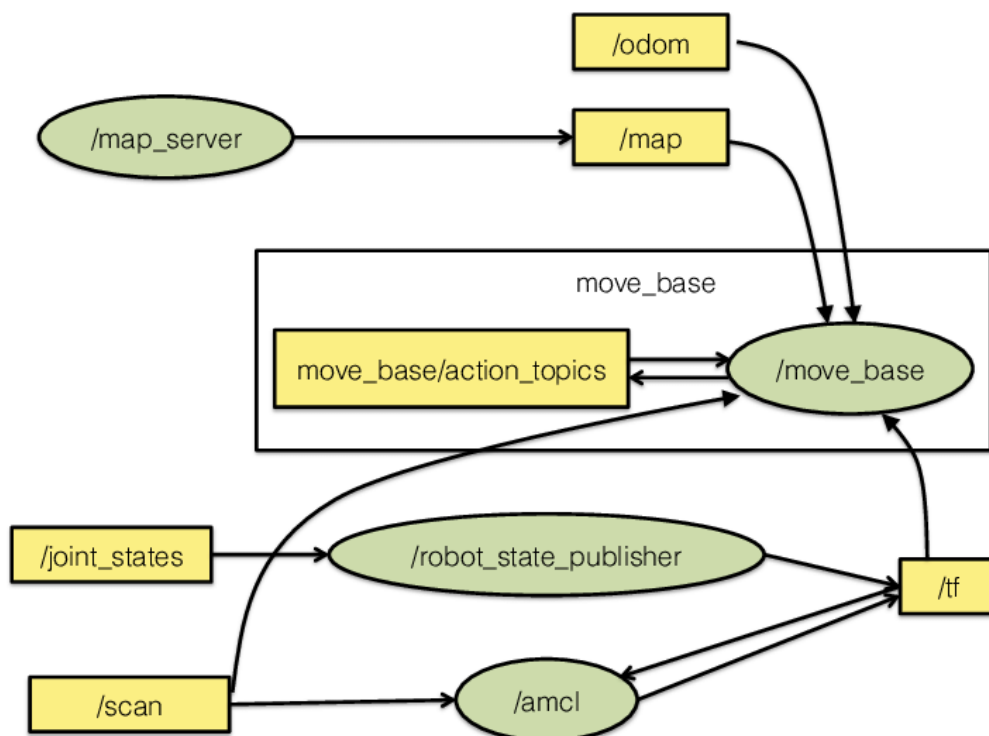
3.4 Βιβλιοθήκες του ROS

Το ROS έχει πολλά περισσότερα πράγματα να προσφέρει από το είναι απλά ένα σύνολο βασικών λειτουργιών και εργαλείων επικοινωνίας. Στην πραγματικότητα, μπορεί να χρησιμοποιηθεί ένα τεράστιο ποσοστό από τις υπάρχουσες βιβλιοθήκες ROS απευθείας. Για παράδειγμα μπορεί να χρησιμοποιηθεί η ανάπτυξη λογισμικού για ένα απλό κινούμενο ρομποτικό σύστημα. Κατά την σχεδίαση ενός τέτοιου συστήματος, μερικά λογισμικά που πρέπει να αναπτυχθούν είναι τα εξής (What is ROS?, n.d.):

- Οδηγός για τους κινητήρες
- Βρόχος ελέγχου PID
- Σχεδιασμός κινήσεων
- Πρόγραμμα οδήγησης κάμερας

Η λίστα μπορεί να γίνει αρκετά μεγαλύτερη αν υπάρξει η ανάγκη να προστεθούν περισσότερες λειτουργίες. Είναι σημαντικό να σημειωθεί ότι κάθε στοιχείο στη λίστα θα απαιτήσει τεράστιο χρονικό διάστημα για να αναπτυχθεί και ακόμα μεγαλύτερο διάστημα για να δοκιμαστεί.

Είναι σαφώς λογικό επακόλουθο το συμπέρασμα ότι θα ήταν αρκετά απαιτητική η ανάπτυξη ενός αλγόριθμου προγραμματισμού κίνησης σε ένα γεμάτο περιβάλλον με κινούμενα εμπόδια. Το πρόβλημα επιλύεται κατά έναν μεγάλο βαθμό με το ROS. Για τα περισσότερα από τα συνηθισμένα προβλήματα στη ρομποτική, μπορεί να βρεθεί ένα πακέτο ROS ανοιχτού κώδικα το οποίο μπορεί να το επιλύσει. Αυτό είναι ένα από τα μεγαλύτερα προτερήματα του λογισμικού ανοιχτού κώδικα (open source). Όταν αναπτυχθεί ένα πακέτο, μπορεί να δημοσιευτεί και να μοιραστεί με την κοινότητα. Για παράδειγμα στην Εικόνα 22 ο κόμβος του Motion Planning είναι συμπεριλαμβανόμενη μέσα στις βιβλιοθήκες του ROS και δεν χρειάζεται να αναπτυχθεί από το μηδέν κάθε φορά που πρέπει να χρησιμοποιηθεί. Για αυτό τον λόγο, άλλοι άνθρωποι μπορούν να το χρησιμοποιήσουν και να συνεχίσουν να το βελτιώνουν.



Εικόνα 22. Λεπτομερές σχήμα επικοινωνίας μεταξύ κόμβων

Επιπλέον, τα εργαλεία προσομοίωσης που παρέχονται από το ROS είναι επίσης εξαιρετικά χρήσιμα. Είναι σημαντικό χάσιμο χρόνου, και σε πολλές περιπτώσεις καταστροφικό

σφάλμα, κατά την ανάπτυξη ενός ρομποτικού συστήματος να πρέπει πάντα να εγκατασταθεί και να δοκιμαστεί το φυσικό ρομπότ για να γίνει ένα πείραμα. Τα εργαλεία προσομοίωσης που παρέχει το ROS, επιτρέπουν τους προγραμματιστές να αναπτύξουν την πλήρη εφαρμογή τους (πάνω από το επίπεδο του προγράμματος οδήγησης υλικού) απευθείας στον υπολογιστή.

3.5 Η κοινότητα του ROS

Το ROS δεν είναι μόνο ένα σύνολο εργαλείων που προαναφέρθηκε. Είναι επίσης μια τεράστια και αναπτυσσόμενη κοινότητα (What is ROS?, n.d.). Και αυτό το σημείο είναι μεγάλης σημασίας. Όταν χρησιμοποιείται μια βιβλιοθήκη, πλαίσιο, ενδιάμεσο λογισμικό ή οτιδήποτε σχετίζεται με το λογισμικό υπάρχουν μεγάλες πιθανότητες να έχει ήδη αναπτυχθεί από κάποιον στην κοινότητα του ROS. Το ROS αναπτύσσεται σήμερα ενεργά από πολλούς ανθρώπους σε όλο τον κόσμο. Επίσης, πολλές εταιρείες χρηματοδοτούν κάποια ανάπτυξη ανοιχτού κώδικα που σχετίζεται με το ROS.

Υπάρχει μια παγκόσμια επιτροπή ROS, Open Robotics, που διατηρεί το όραμα και αποφασίζει για την πορεία που θα ακολουθηθεί για την περαιτέρω ανάπτυξη. Συχνά παρέχουν κάποιες πληροφορίες για το τι συμβαίνει και λαμβάνουν πληροφορίες από την παγκόσμια κοινότητα για να πάρουν ιδέες και να δουν τι είναι σημαντικό για τους χρήστες ROS. Η κοινότητα ROS είναι επίσης πολύ ενεργή στο διαδίκτυο:

- ROS Wiki. Αυτό είναι όπου θα βρείτε τα περισσότερα από τα σεμινάρια, τις εξηγήσεις ιδεών και τους οδηγούς για διαφορετικά πακέτα. Είναι μια πολύ ωραία ιστοσελίδα, αλλά μερικές φορές δεν είναι πλήρης και δεν υπάρχουν κάποιες οδηγίες, οπότε θα πρέπει να βρείτε περισσότερες πληροφορίες αλλού.
- ROS answers. Ένας δικτυακός τόπος E & A, ώστε να μπορείτε να ζητήσετε όλες τις τεχνικά προσανατολισμένες ερωτήσεις σας. Υπάρχει ήδη ένα καλό ποσό απαντήσεων σε ερωτήσεις που μπορείτε να ελέγξετε. Συχνά, αν δεν μπορείτε να βρείτε κάτι στο Wiki, πιθανόν να το βρείτε εδώ.
- ROS Discourse. Αυτό είναι ένα φόρουμ για να μιλήσουμε για μελλοντικές εξελίξεις, έργα που χρησιμοποιούν ROS, και οτιδήποτε σχετίζεται με ROS. Αυτή είναι μια εξαιρετική πηγή για να ελέγχετε από καιρό σε καιρό, έτσι είστε σίγουροι ότι δεν θα χάσετε τίποτα.
- GitHub. Τα περισσότερα από τα πακέτα ROS είναι διαθέσιμα στο GitHub. Εδώ θα μπορείτε να περιηγηθείτε στον κώδικα και να κάνετε κάποιες συμβουλές.

3.6 Πότε χρησιμοποιείται το ROS

Το ROS είναι ένα εργαλείο το οποίο χρησιμοποιείται κυρίως για πολύπλοκες εφαρμογές. Για παράδειγμα σε εφαρμογές όπως ενός ρομπότ 2 τροχών που ακολουθεί κατά μήκος μια γραμμή στο έδαφος μπορεί να υλοποιηθεί με εργαλεία όπως το Arduino. Το ROS χρησιμοποιείται πολύ στις μέρες μας. Είναι ένα εργαλείο που χρησιμοποιούν τόσο ερευνητές όσο και εταιρείες. Ο λόγος είναι η ευκολία στην διαχείριση μιας πολύπλοκης εφαρμογής, όπως ένα ρομπότ με πολλούς διαφορετικούς αισθητήρες και ενεργοποιητές. Το ROS βοηθάει στην δημιουργία ενός κατακευματισμένου συστήματος για όλα αυτά τα στοιχεία. Επίσης με το ROS μπορεί να εξοικονομηθεί χρόνος σε μεγάλο βαθμό. Συνήθως, πολλοί ερευνητές χρειάζονται να εστιάσουν σε ένα πολύ ιδιαίτερο σημείο. Όπως έχει προαναφερθεί με το ROS, τα εργαλεία επικοινωνίας και οι βιβλιοθήκες έχουν ήδη αναπτυχθεί, σαν αποτέλεσμα να εξοικονομείται χρόνος, που θα εστιαζόταν στην ανάπτυξη τους.

Κεφάλαιο 4. Συλλογή και προετοιμασία δεδομένων

Για την δημιουργία ενός αλγόριθμου μηχανικής μάθησης είναι πολύ σημαντικό να παρθούν κάποια μέτρα και να εκτελεστούν κάποιες ενέργειες. Αυτό είναι αναγκαίο γιατί η προετοιμασία των δεδομένων έχει σαν αποτέλεσμα την μακροπρόθεσμη εξοικονόμηση χρόνου και ακόμα πιο σημαντικό είναι ότι η σωστή μορφή των δεδομένων θα βοηθήσει τον αλγόριθμο να μπορεί να «καταλάβει» και να επεξεργαστεί πιο αποδοτικά τα δεδομένα.

4.1 Συλλογή δεδομένων

Τα δεδομένα που χρησιμοποιήθηκαν προέρχονται από ερευνητικά προγράμματα από το πανεπιστήμιο του Birmingham στην Αγγλία. Συγκεκριμένα τα δεδομένα ήταν σε μορφή αρχείου .bag (rosbag). Ένα αρχείο rosbag είναι μια μορφή αρχείου στο ROS για την αποθήκευση δεδομένων μηνυμάτων ROS. Τα rosbags - που έχουν ονομαστεί λόγω της επέκτασης .bag - έχουν σημαντικό ρόλο στο ROS και έχουν γραφτεί διάφορα εργαλεία που επιτρέπουν την αποθήκευση, την επεξεργασία, την ανάλυση και την απεικόνιση των δεδομένων που εμπεριέχουν (Ros Wiki, 2018).

Καταρχάς τα δεδομένα ήταν χωρισμένα σε πειράματα. Υπήρχαν πειράματα βέλτιστης οδήγησης από έμπειρους χειριστές τα οποία ονομάζονται `entropy_0X_noworkload.bag` και όπου X είναι ο αριθμός του χειριστή. Να σημειωθεί ότι υπήρχαν πειράματα με τέσσερεις χειριστές. Επίσης υπήρχαν πειράματα από τους ίδιους χειριστές (με όνομα `entropy_0X_workload.bag` και όπου X είναι ο αριθμός του χειριστή) αλλά η διαφορά ήταν ότι σε αυτά τα πειράματα, οι χειριστές είχαν χαμηλότερη απόδοση λόγω μιας δευτερεύουσας διεργασίας. Σαν πρώτη εργασία έπρεπε να γίνει η επιλογή των δεδομένων. Με συνεχή επικοινωνία με τον υπεύθυνο των πειραμάτων έγινε μια ανάλυση των θεμάτων η οποία παρουσιάζεται παρακάτω.

Πίνακας 1. Χρήσιμα θέματα

Όνομα θέματος του	Περιγραφή
ros	

experiment_started	Αναγράφει την χρονική στιγμή που ξεκίνησε το πείραμα. ¹
cmd_vel	Αναγράφει, ανάλογα με την συχνότητα μεταφοράς δεδομένων του θέματος, την γραμμική και γωνιακή ταχύτητα του ρομπότ
joy	Αναγράφει, ανάλογα με την συχνότητα μεταφοράς δεδομένων του θέματος, όλα τα κουμπιά του τηλεχειριστήριου καθώς και τις τιμές των αναλογικών μοχλών

4.2 Προεργασία των δεδομένων

4.2.1 Δειγματοληψία

Σε διάφορες εφαρμογές είναι σύνηθες τα δεδομένα να είναι πολλά ή να είναι ασύγχρονα. Η ποσότητα των δεδομένων σε πολλές εφαρμογές είναι το σημείο κλειδί για την ανάπτυξη ενός πολύ καλού μοντέλου. Αρκετές φορές όμως τα πολλά δεδομένα έχουν ως αποτέλεσμα πολύ μεγαλύτερους χρόνους λειτουργίας για αλγόριθμους και μεγαλύτερες απαιτήσεις υπολογιστικής και μνήμης (Brownlee, n.d.). Στην περίπτωση των πειραμάτων που χρησιμοποιήθηκαν υπήρχε άλλο πρόβλημα.

Αφού επιλέχθηκαν ποια θέματα θα χρησιμοποιηθούν έγινε μια παρατήρηση. Τα θέματα των rosbags είχαν διαφορετική συχνότητα δειγματοληψίας με τα υπόλοιπα θέματα. Στην συγκεκριμένη εφαρμογή είναι αναγκαίο όλα τα δεδομένα να είναι συγχρονισμένα, γι' αυτό το λόγο ήταν αναγκαία η ανάπτυξη ενός κόμβου στο ros (roscnode) ο οποίος θα συγχρόνιζε όλα τα δεδομένα του rosbag. Ο αλγόριθμος αυτός αποτελούταν από 4 τμήματα. Στο πρώτο τμήμα αρχικοποιήθηκαν τα καινούργια θέματα τα οποία είχαν ίδιο όνομα με τα θέματα που έπρεπε να συγχρονιστούν αλλά με την διαφορά ότι το όνομα του καινούργιου θέματος τελείωνε με κατάληξη `_new`. Στο επόμενο τμήμα έπρεπε να οριστούν τα καινούργια πλέον μηνύματα που θα δεχόντουσαν τα νέα θέματα. Έπειτα αφού τα καινούργια θέματα είχαν πλέον οριστεί και εγγραφεί στα παλιά (subscribe), το καθένα θέμα καλούταν να δημοσιεύσει (publish) τα δεδομένα του με δειγματοληψία η οποία ορίστηκε στα 20hz δηλαδή 50 msec.

Μετάπειτα βρέθηκε, κάθε 150msec, η μέση τιμή των δεδομένων. Αυτό το διάστημα δειγματοληψίας αποτελεί καλή προσέγγιση της χρονικής διάρκειας παρακολούθησης του

¹ Δεδομένα πριν την εκκίνηση των πειραμάτων δεν είναι κατάλληλα για την δημιουργία αλγορίθμων μηχανικής μάθησης. Αυτός είναι ο λόγος για τον οποίο είναι αναγκαίο αυτό το θέμα

χρόνου χειρισμού του χειριστή και αντιστοιχεί στη χαμηλότερη συχνότητα δειγματοληψίας που μπορεί δικαιολογημένα να χρησιμοποιηθεί για να αντιπροσωπεύει την ανταπόκριση ελέγχου του χειριστή σε ένα ευρύ φάσμα χειρωνακτικών εντολών παρακολούθησης (Sheridan T. B., 1989). Μετά την χρήση του καινούργιου κόμβου αποθηκεύτηκαν τα αρχεία `rosbag` με τα επιθυμητά θέματα και με σωστή δειγματοληψία.

4.2.2 Μετατροπή από `rosbag` σε μορφή `csv`

Για την μετατροπή από αρχείο `rosbag` σε `csv` χρησιμοποιήθηκε ένας κόμβος του `ros` που ονομάζεται `rosbag_to_csv` (AtsushiSakai, n.d.). Με την εκκίνηση αυτού του κόμβου υπάρχει η δυνατότητα επιλογής των θεμάτων που πρέπει να μετατραπούν. Στην συνέχεια αποθηκεύονται σε μορφή `csv`.

4.2.3 Μορφοποίηση

Τα δεδομένα από τα καινούργια πλέον `rosbags` δεν είχαν την κατάλληλη μορφή. Τα αρχεία `csv` εμπεριέχαν ταχύτητες για όλους τους άξονες, πράγμα που δεν χρειαζόταν να υπάρχει. Τα δεδομένα του θέματος `cmd_vel` που έπρεπε να αποθηκευτούν ήταν μόνο η γωνιακή ταχύτητα στο άξονα ζ, δηλαδή τον κατακόρυφο άξονα και η γραμμική ταχύτητα στον άξονα x^2 . Η μορφοποίηση του θέματος ήταν σχετικά εύκολη. Απλά διαγράφηκαν όλα τα δεδομένα εκτός από τις δύο ταχύτητες.

Το θέμα `joy` χρειαζόταν πιο σκληρή εργασία. Το θέμα είχε σε μια στήλη όλα τα κουμπιά χωρισμένα με κόμμα και σε διαφορετική στήλη όλες τις τιμές των αναλογικών μοχλών επίσης χωρισμένες με κόμμα. Η λογική της μορφοποίησης αυτής ήταν να γίνει μια προσπέλαση στην λίστα των κουμπιών και στην λίστα των μοχλών και να επιλεγθούν μόνο τα κουμπιά τα οποία χρησιμοποιήθηκαν στην οδήγηση του ρομπότ. Έχοντας πλέον χωρίσει τις τιμές των κουμπιών και τις τιμές των μοχλών δεν έμενε παρά να αποθηκευτούν σε διαφορετικό αρχείο οι τιμές του μοχλού για την αριστερή/δεξιά κίνηση και για την εμπρός/πίσω κίνηση.

Και στις τέσσερις³ καινούργιες κατηγορίες δεδομένων έπρεπε να αφαιρεθούν όσα δεδομένα υπήρχαν πριν την στιγμή που το θέμα `experiment_started` πήρε αληθή τιμή. Η προεργασία που έγινε ήταν απλά να διαγραφούν όσα δεδομένα ήταν πριν την στιγμή αυτή.

² Το ρομπότ μπορούσε να κινηθεί μπρος, πίσω και να στρίψει στον άξονα του. Εξού και η επιλογή των ταχυτήτων.

³ Γραμμική ταχύτητα, γωνιακή ταχύτητα, αριστερή/δεξιά κίνηση μοχλών και μπρος/πίσω κίνηση μοχλών.

4.3 Περιγραφή τελικής μορφής δεδομένων

Μετά από την προ-επεξεργασία των δεδομένων όλα τα δεδομένα κατατάχθηκαν σε 2 κατηγορίες. Την κατηγορία χωρίς και την κατηγορία με φόρτο εργασίας. Κάθε κατηγορία αποτελείται από τέσσερα πειράματα από τέσσερις διαφορετικούς έμπειρους οδηγούς. Κάθε κατηγορία από αυτές έχει 4 τύπους δεδομένων. Τη γραμμική ταχύτητα, τη γωνιακή ταχύτητα, την αριστερή/δεξιά κίνηση μοχλών και την εμπρός/πίσω κίνηση μοχλών. Κάθε κατηγορία μοντελοποιήθηκε με τα 6 μοντέλα. Και στις δύο κατηγορίες⁴ τα πειράματα των τεσσάρων οδηγών ενώθηκαν μεταξύ τους. Οπότε κάθε τύπου δεδομένων πλέον περιλαμβάνει και τα τέσσερα πειράματα αντί να είναι ξεχωριστά.

Ένας τρόπος για την βελτίωση μιας μηχανής μάθησης είναι να εκπαιδύσουμε πολλά μοντέλα με διαφορετικές αρχιτεκτονικές/συναρτήσεις και να εκτιμήσουμε την ακρίβεια των μοντέλων με γνώμονα αυτό το μοντέλο που γενικεύει καλύτερα (Διαμαντάρας, 2007) με την μέθοδο cross-validation μπορεί να επιτευχθεί αυτό. Έστω ότι υπάρχει σετ δεδομένων μεγέθους n . Για να μπορέσουμε να εκτιμήσουμε την ικανότητα γενίκευσης του μοντέλου είναι δυνατή η εκπαίδευση του μοντέλου με το 90% του σετ και το υπόλοιπο 10% το οποίο δεν έχει χρησιμοποιηθεί για την εκπαίδευση, θα χρησιμοποιηθεί για την ελεγχθεί το πόσο καλά γενικεύει το δίκτυο. Έχοντας λάβει τα παραπάνω υπόψιν το σετ χωρίστηκε σε σετ εκπαίδευσης και ελέγχου⁵.

⁴ Κατηγορία χωρίς και την κατηγορία με φόρτο εργασίας.

⁵ Το 80% του αρχικού σετ έγινε το σετ εκπαίδευσης και το υπόλοιπο έγινε σετ ελέγχου

Κεφάλαιο 5. Αποτελέσματα μοντέλων

Παρακάτω θα παρουσιαστούν σε κάθε κεφάλαιο ξεχωριστά τα διαφορετικά μοντέλα και τα αποτελέσματά τους. Στα μοντέλα (εκτός από το αυτοπαλίνδρομο και το μοντέλο αναπτύγματος Τέιλορ) χρειάστηκε να δημιουργηθούν καινούργια δεδομένα που θα έχουν τον ρόλο των εισόδων του αλγόριθμου. Παρακάτω παρουσιάζονται σαν παράδειγμα τα δεδομένα της γωνιακής ταχύτητας για τις τρεις πρώτες χρονικές στιγμές:

Πίνακας 2. Παράδειγμα δεδομένων γωνιακής ταχύτητας

.angular.z	.angular.z_1lag	.angular.z_2lag
-0.06563	NAN	NAN
-0.06565	-0.06563	NAN
-0.06567	-0.06565	-0.06563

Η έξοδος των μοντέλων αυτών θα ήταν η προβλεπόμενη εντολή ή κίνηση σε κάθε χρονική στιγμή. Συγκεκριμένα στο παράδειγμα του πίνακα 2 σαν είσοδο στους περισσότερους αλγόριθμους θα είναι οι στήλες .angular.z_1lag και .angular.z_2lag. Όπως φαίνεται και στον πίνακα η πρώτη στήλη είναι στην ουσία η πρώτη αλλά καθυστερημένη κατά μία χρονική στιγμή και η δεύτερη στήλη είναι καθυστερημένη κατά δύο.

Τα δεδομένα που χρησιμοποιήθηκαν για τα μοντέλα είναι οι εντολές κίνησης⁶ του χειριστή από το χειριστήριο. Επίσης χρησιμοποιήθηκαν και οι ταχύτητες του ρομπότ⁷.

Πίνακας 3. Τελικά δεδομένα πειραμάτων

Δεδομένα με φόρτο εργασίας	Δεδομένα χωρίς φόρτο εργασίας	Περιγραφή
cmd_features_angular_wl	cmd_features_angular_no_wl	Γωνιακή ταχύτητα χωρίς/με φόρτο
cmd_features_linear_wl	cmd_features_linear_no_wl	Γραμμική ταχύτητα χωρίς/με φόρτο
joy_features_lr_wl	joy_features_lr_no_wl	Εντολές χειριστή (αριστερά δεξιά) χωρίς/με φόρτο

⁶ Εντολή κίνησης πάνω/κάτω και η εντολή δεξιά/αριστερή στροφή.

⁷ Η γραμμική και γωνιακή ταχύτητα.

joy_features_up_wl	joy_features_up_no_wl	Εντολές χειριστή (αριστερά δεξιά) χωρίς/με φόρτο
--------------------	-----------------------	--

Για να μπορεί να επιβεβαιωθεί ότι το μοντέλο μπορεί να πετύχει την καλύτερη δυνατή γενίκευση τα δεδομένα χωρίστηκαν σε ένα σύνολο εκπαίδευσης και σε ένα σύνολο ελέγχου όπως προαναφέρθηκε και στο κεφάλαιο 4.3

Η βασική υπόθεση αυτής της εργασίας είναι ότι δημιουργώντας ένα μοντέλο το οποίο προσεγγίζει τις πιο αποδοτικές μανούβρες και τους πιο αποδοτικούς χειρισμούς, τότε θα μπορεί να χρησιμοποιηθεί και για την αξιολόγηση των χειριστών. Ουσιαστικά όταν κάποιος οδηγός δεν οδηγεί αποδοτικά, το τετραγωνικό σφάλμα που θα υπάρξει μεταξύ πραγματικής και προβλεπόμενης οδήγησης μπορεί να χρησιμοποιηθεί για την αξιολόγηση του χειριστή.

Σε όλα τα παρακάτω γραφήματα εμφανίζεται με *μπλε γραμμή η πραγματική έξοδος* και με *κόκκινη γραμμή οι προβλέψεις*.

5.1 Μοντέλο κατάβασης δυναμικού

Το μοντέλο κατάβασης έχει μια από τις καλύτερες αποδόσεις. Αφού τα δεδομένα χωρίστηκαν σε ένα σύνολο εκπαίδευσης και σε ένα σύνολο ελέγχου, έγινε η κλήση της συνάρτησης `LinearRegression()` και της `regressor.fit`, οι οποίες καλούνται να αρχικοποιήσουν το μοντέλο και μετά να το εκπαιδεύσουν. Μετά καλώντας την συνάρτηση `regressor.predict()` μπορεί να γίνουν προβλέψεις με τις πλέον γνωστές παραμέτρους του μοντέλου.

5.1.1 Γωνιακή ταχύτητα

Τα αποτελέσματα του μοντέλου εμφανίζονται παρακάτω.

Πίνακας 4. Σφάλματα μοντέλου κατάβασης γωνιακής ταχύτητας

MSE ⁸ no_workload ⁹	0.0037787190278056014
RMSE ¹⁰ no_workload	0.06147128620588316
MSE workload ¹¹	0.02602395516059369

⁸ MSE = Μέσο τετραγωνικό σφάλμα

⁹ No workload = Χωρίς φόρτο εργασίας

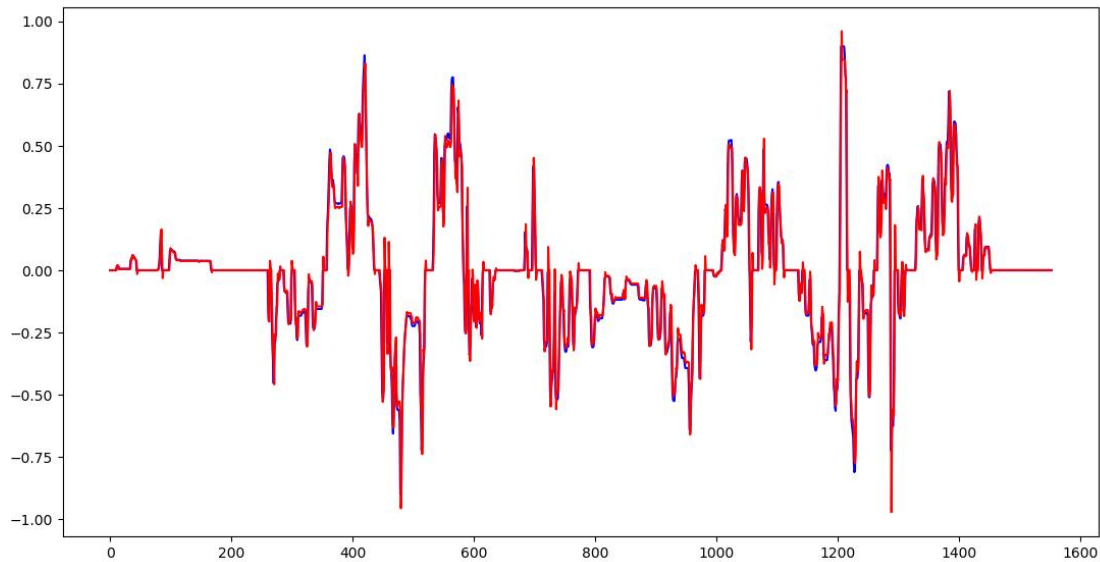
¹⁰ RMSE = Ρίζα μέσου τετραγωνικού σφάλματος

¹¹ Workload = Με φόρτο εργασίας

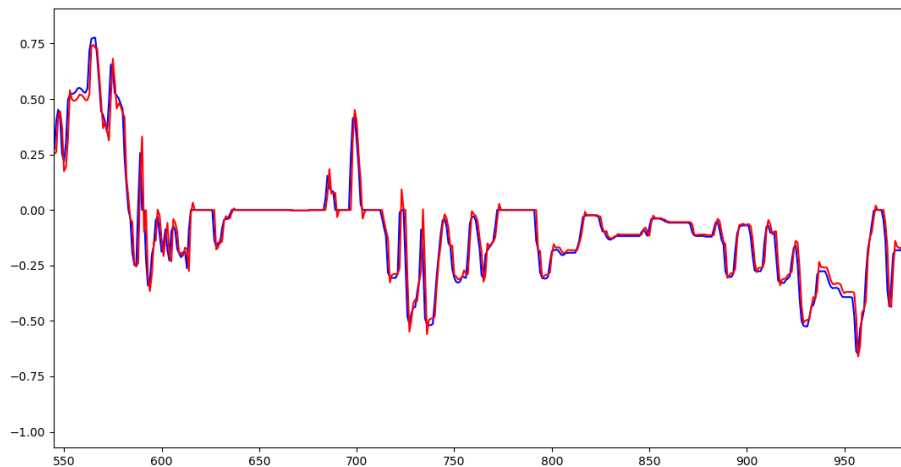
RMSE workload	0.1613194196635783
---------------	--------------------

Πίνακας 5. Παράμετροι μοντέλου κατάβασης γωνιακής ταχύτητας

Παράμετροι μοντέλου	Τιμές παραμέτρων
.angular.z_1lag	1.320557
.angular.z_2lag	-0.380591



Εικόνα 23. Πρόβλεψη μοντέλου κατάβασης γωνιακής ταχύτητας στο σετ ελέγχου



Εικόνα 24. Πρόβλεψη μοντέλου κατάβασης γωνιακής ταχύτητας σετ ελέγχου σε μεγέθυνση

5.1.2 Γραμμική ταχύτητα

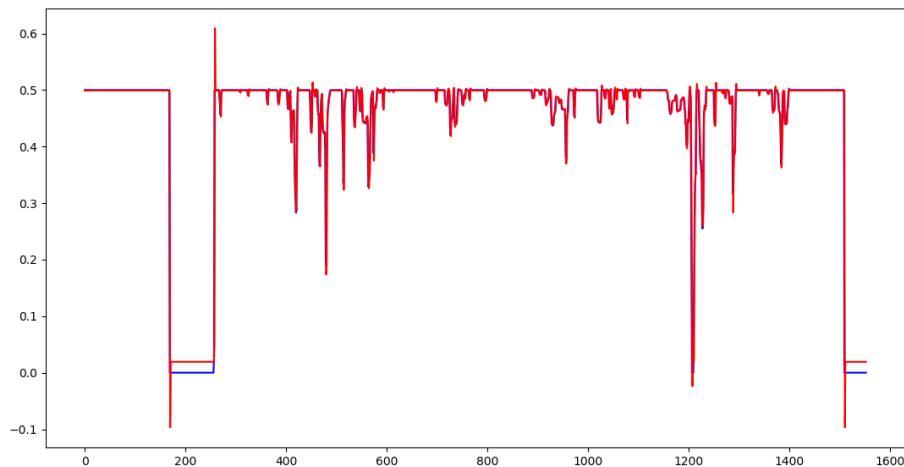
Πίνακας 6. Σφάλματα μοντέλου κατάβασης γραμμικής ταχύτητας

MSE no_workload	0.0007385969863143243
RMSE no_workload	0.02717714087821462

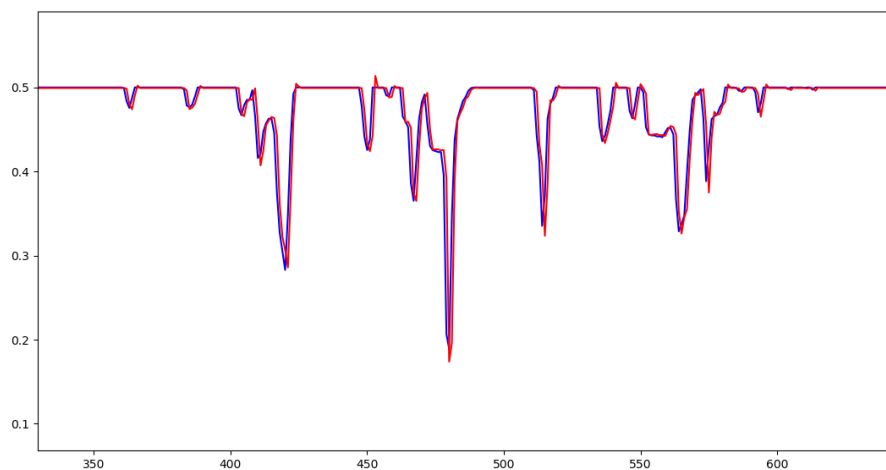
MSE workload	0.007131817923951304
RMSE workload	0.08445009131997018

Πίνακας 7. Παράμετροι μοντέλου κατάβασης γραμμικής ταχύτητας

Παράμετροι μοντέλου	Τιμές παραμέτρων
.linear.x_1lag	1.191529
.linear.x_2lag	-0.231721



Εικόνα 25. Πρόβλεψη μοντέλου κατάβασης γραμμικής ταχύτητας στο σετ ελέγχου



Εικόνα 26. Πρόβλεψη μοντέλου κατάβασης γραμμικής ταχύτητας σετ ελέγχου σε μεγέθυνση

5.1.3 Εντολές χειριστή δεξιά/αριστερά

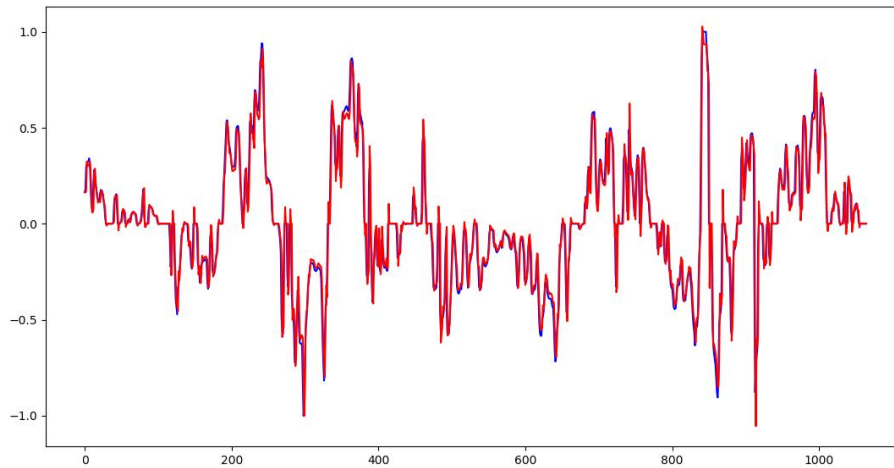
Πίνακας 8. Σφάλματα μοντέλου κατάβασης δεξιάς/αριστερής εντολής

MSE no_workload	0.00720852579672657
RMSE no_workload	0.08490303761778238
MSE workload	0.03341551477684252

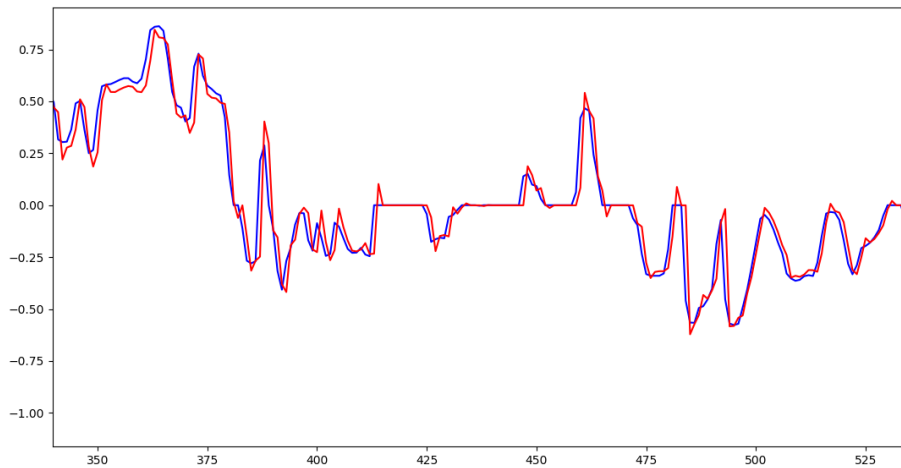
RMSE workload	0.1827991104377768
---------------	--------------------

Πίνακας 9. Παράμετροι μοντέλου κατάβασης δεξιάς/αριστερής εντολής

Παράμετροι μοντέλου	Τιμές παραμέτρων
.joy_left_right_1lag	1.351176
.joy_left_right_2lag	-0.417865



Εικόνα 27. Πρόβλεψη μοντέλου κατάβασης δεξιάς/αριστερής εντολής στο σετ ελέγχου



Εικόνα 28. Πρόβλεψη μοντέλου κατάβασης δεξιάς/αριστερής εντολής σετ ελέγχου σε μεγέθυνση

5.1.4 Εντολές χειριστή μπρος/πίσω

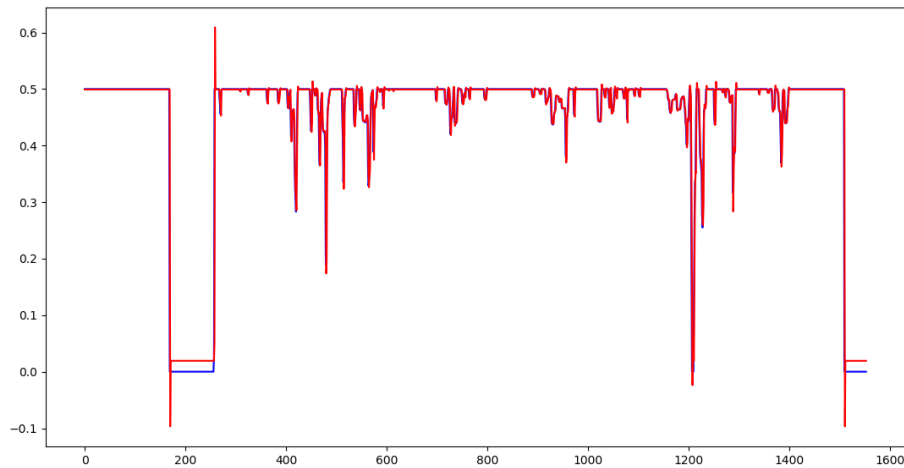
Πίνακας 10. Σφάλματα μοντέλου κατάβασης μπρος/πίσω εντολής

MSE no_workload	0.0007385969863143243
RMSE no_workload	0.02717714087821462

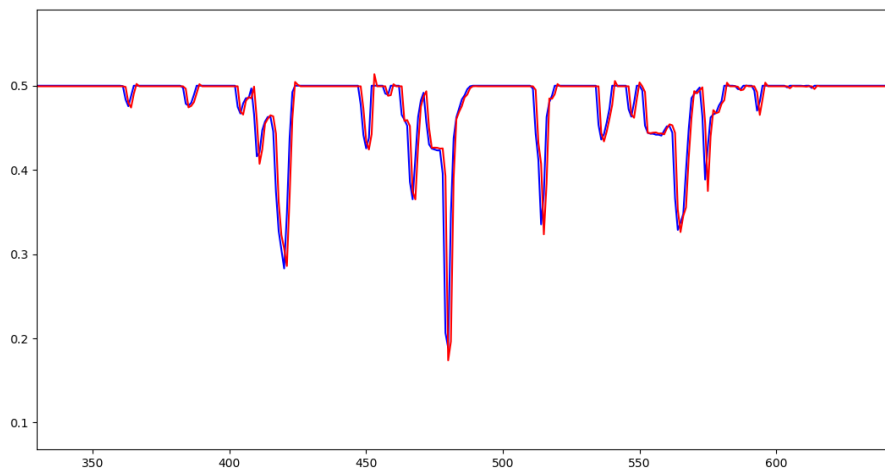
MSE workload	0.007131817923951304
RMSE workload	0.08445009131997018

Πίνακας 11. Παράμετροι μοντέλου κατάβασης μπρος/πίσω εντολής

Παράμετροι μοντέλου	Τιμές παραμέτρων
.joy_up_down_1lag	1.191529
.joy_up_down_2lag	-0.231721



Εικόνα 29. Πρόβλεψη μοντέλου κατάβασης μπρος/πίσω εντολής στο σετ ελέγχου

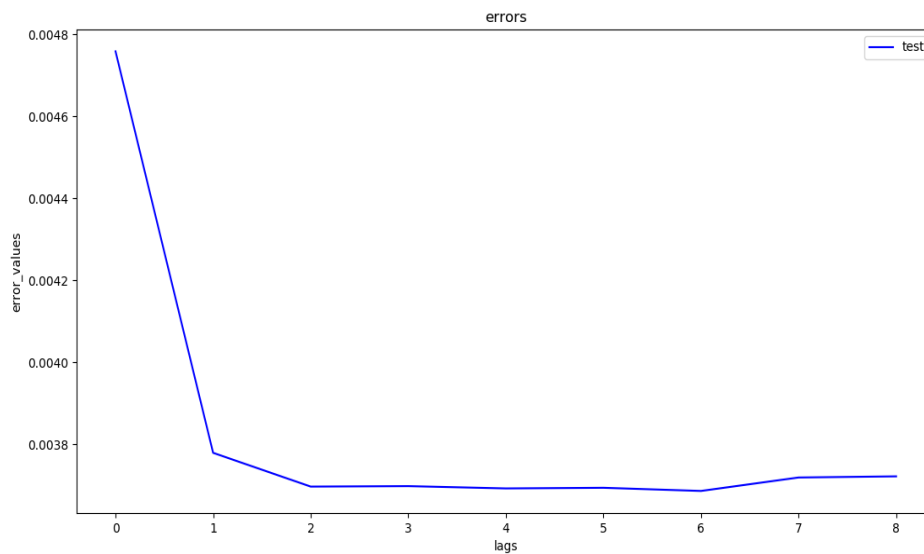


Εικόνα 30. Πρόβλεψη μοντέλου κατάβασης μπρος/πίσω εντολής στο σετ ελέγχου σε μεγέθυνση

5.2 Αυτοπαλίνδρομα μοντέλα

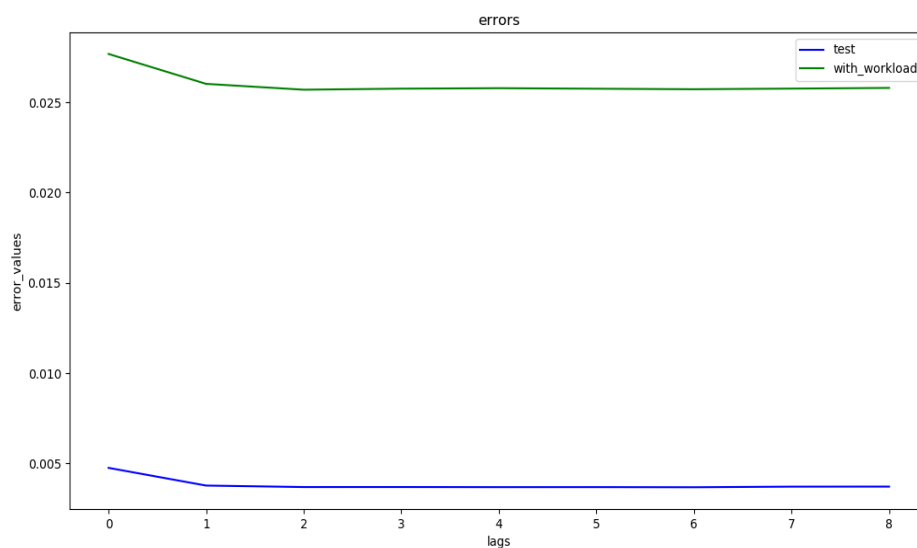
Στα αυτοπαλίνδρομα μοντέλα ακολουθήθηκε διαφορετική τακτική. Η εντολή η οποία εκπαιδεύει το μοντέλο είναι η `model.fit()`. Με την `model.fit` υπάρχει η δυνατότητα επιλογής της τάξης του μοντέλου. Με έναν απλό αλγόριθμο που αυξάνει συνεχώς την τάξη βρέθηκαν τα παρακάτω αποτελέσματα.

5.2.1 Γωνιακή ταχύτητα



Εικόνα 31. Μέσο τετραγωνικό σφάλμα και αριθμός καθυστερήσεων μοντέλου γωνιακής ταχύτητας

Από το γράφημα ότι το μικρότερο σφάλμα βρίσκεται όταν υπάρχουν δύο καθυστερήσεις.



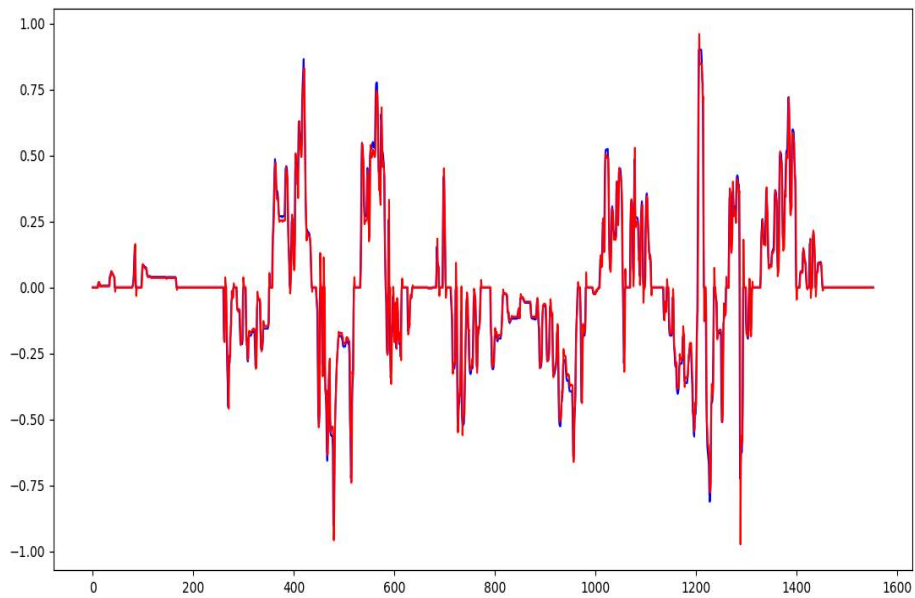
Εικόνα 32. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς μοντέλου γωνιακής ταχύτητας

Πίνακας 12. Παράμετροι και σφάλματα αυτοπαλίνδρομου μοντέλου δεύτερης τάξης γωνιακής ταχύτητας

Lag: 2 Coefficients: [1.32057531 -0.38057089]

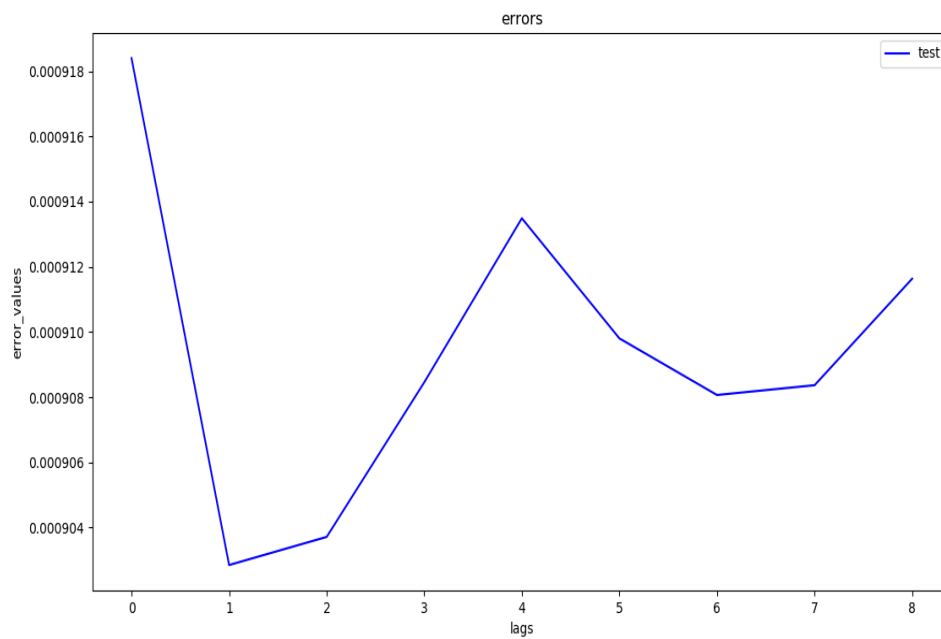
MSE: 0.003, RMSE: 0.054

MSE workload: 0.026, RMSE workload: 0.161

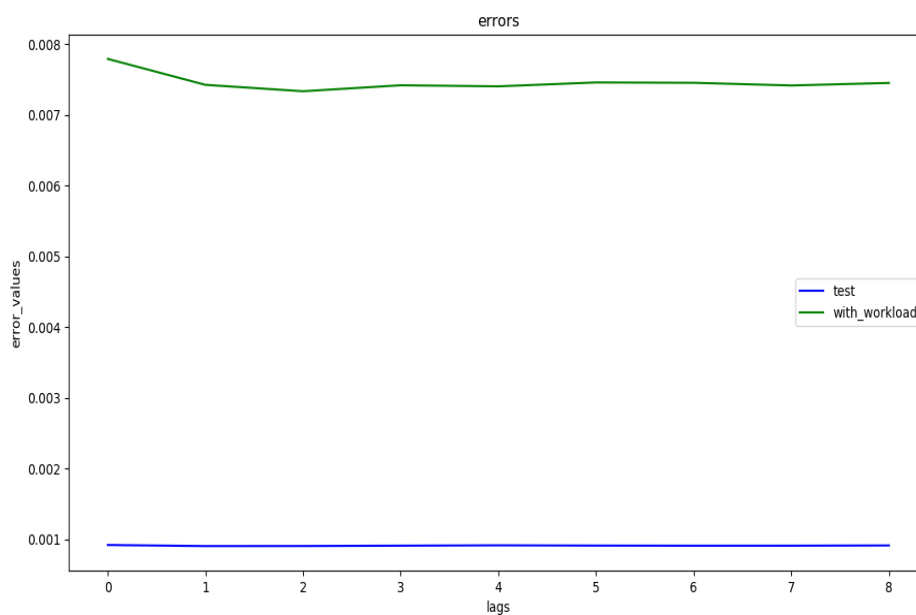


Εικόνα 33. Πρόβλεψη αυτοπαλίνδρομου μοντέλου γωνιακής ταχύτητας

5.2.2 Γραμμική ταχύτητα



Εικόνα 34. Μέσο τετραγωνικό σφάλμα και αριθμός καθυστερήσεων γραμμικής ταχύτητας

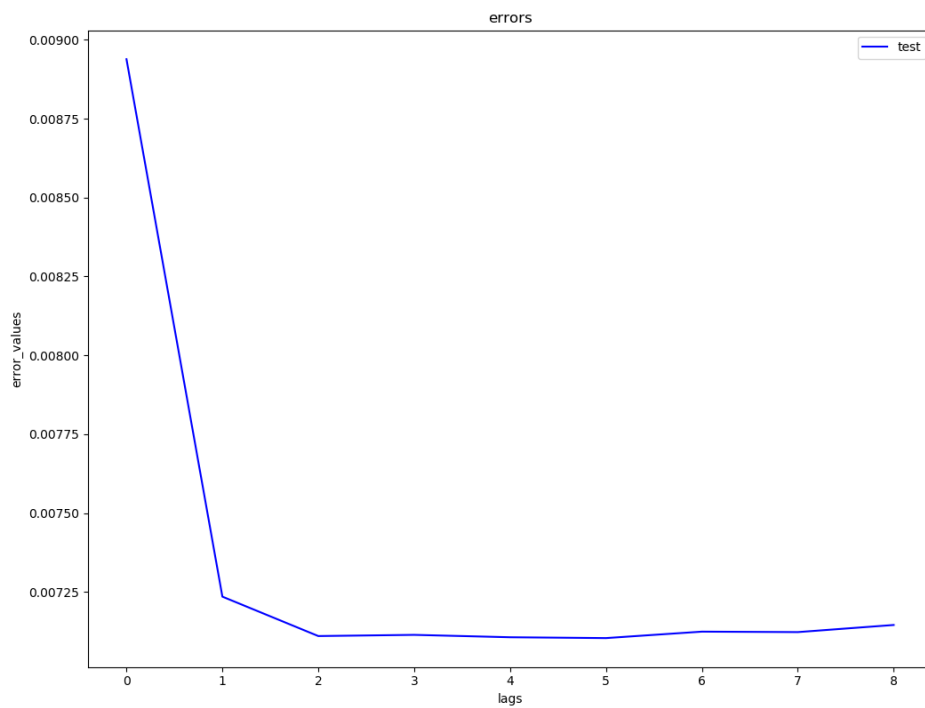


Εικόνα 35. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς μοντέλου γραμμικής ταχύτητας

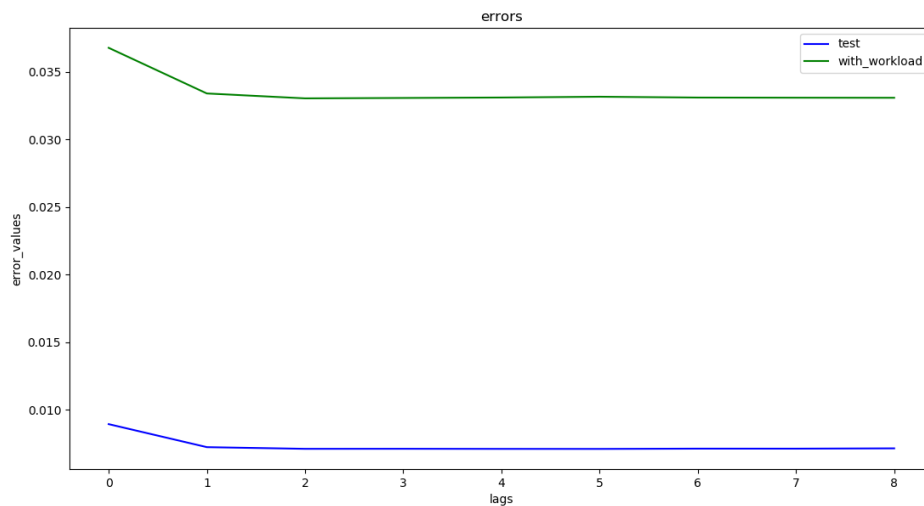
Πίνακας 13. Παράμετροι και σφάλματα αυτοπαλίνδρομου μοντέλου δεύτερης τάξης γραμμικής ταχύτητας

Lag: 2, Coefficients: [1.21108141 -0.21216899] MSE: 0.001, RMSE: 0.030 MSE workload: 0.007, RMSE workload: 0.086

5.2.3 Εντολές χειριστή δεξιά/αριστερά



Εικόνα 36. Μέσο τετραγωνικό σφάλμα και αριθμός καθυστερήσεων μοντέλου δεξιών/αριστερών εντολών χειριστή



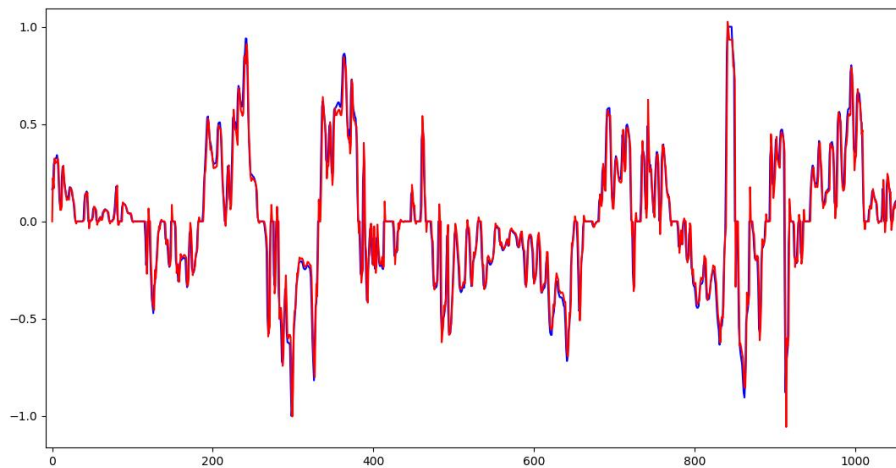
Εικόνα 37. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς μοντέλο δεξιών/αριστερών εντολών χειριστή

Πίνακας 14. Παράμετροι και σφάλματα αυτοπαλίνδρομου μοντέλου δεύτερης τάξης δεξιών/αριστερών εντολών χειριστή

Lag: 2, Coefficients: [1.35120564 -0.41783192]

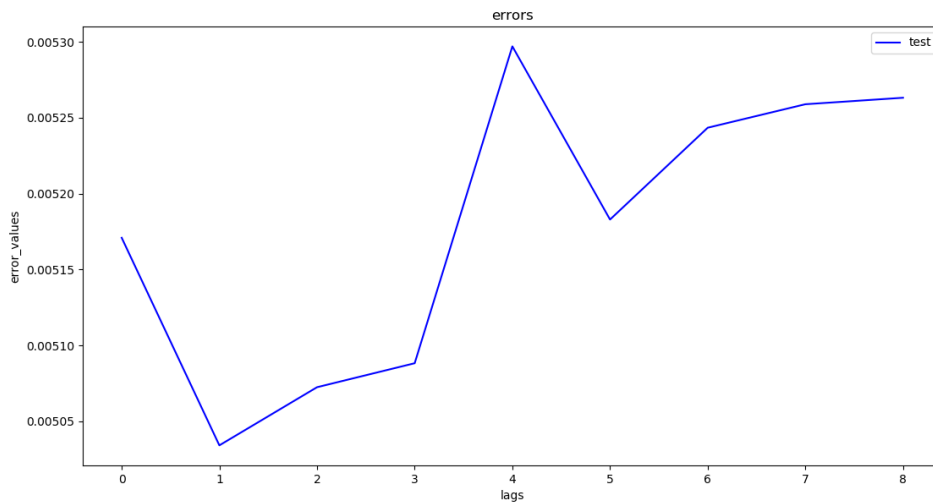
MSE: 0.007, RMSE: 0.085

MSE: 0.033, RMSE: 0.183

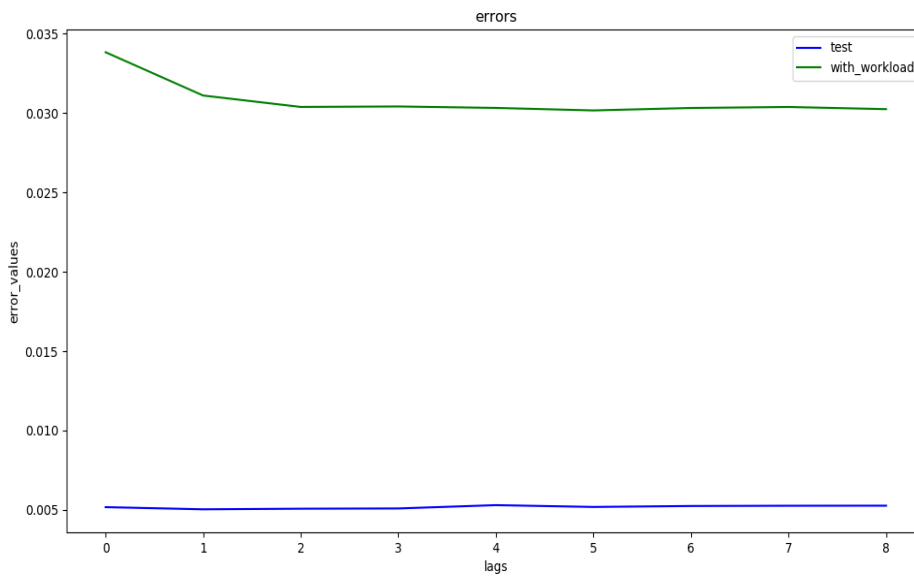


Εικόνα 38. Πρόβλεψη αυτοπαλίνδρομου μοντέλου δεξιάς/αριστερής εντολής στο σετ ελέγχου

5.2.4 Εντολές χειριστή μπρος/πίσω



Εικόνα 39. Μέσο τετραγωνικό σφάλμα και αριθμός καθυστερήσεων μοντέλου μπρος/πίσω εντολών χειριστή



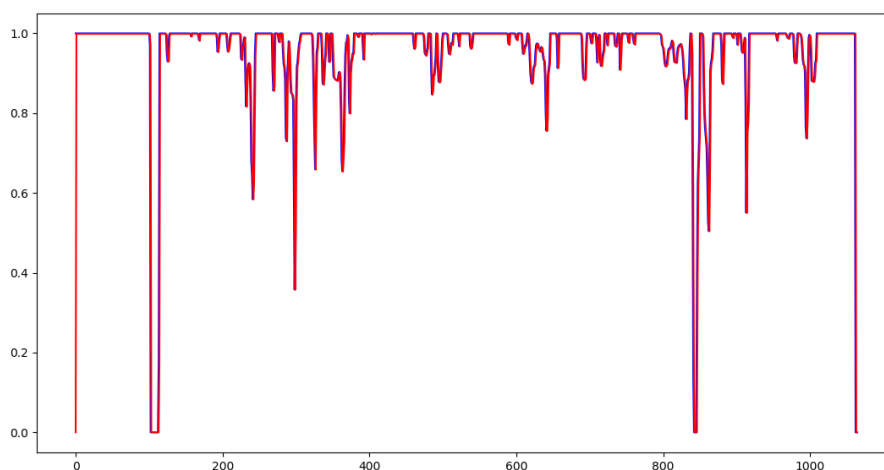
Εικόνα 40. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς μοντέλου μπρος/πίσω εντολών χειριστή

Πίνακας 15. Παράμετροι και σφάλματα αυτοπαλίνδρομου μοντέλου δεύτερης τάξης μπρος/πίσω εντολών χειριστή

Lag: 2, Coefficients: [1.24199898 -0.24360049]

MSE: 0.005, RMSE: 0.071

MSE: 0.031, RMSE: 0.176



Εικόνα 41. Πρόβλεψη αυτοπαλίνδρομου μοντέλου πάνω/κάτω εντολής στο σετ ελέγχου

5.3 Ανάπτυγμα Τέιλορ

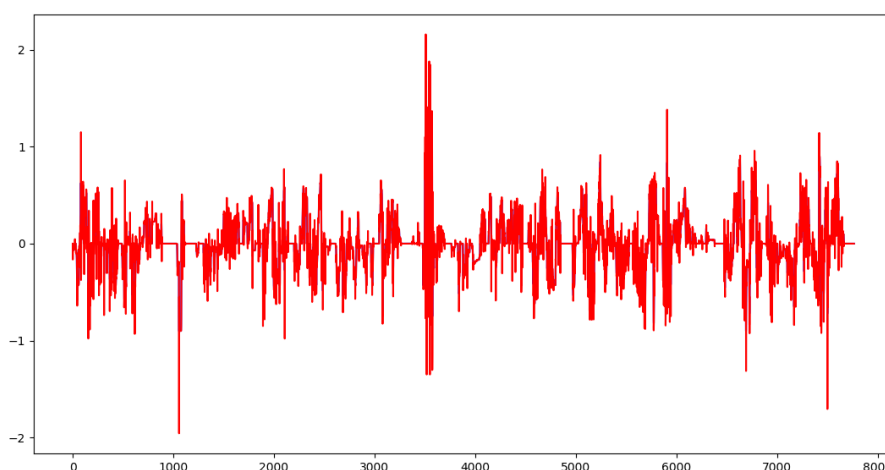
Το μοντέλο αυτό, αν και απλό στην υλοποίηση υστερεί σε ακρίβεια. Η προβλεπόμενη έξοδος είναι πολύ θορυβώδης. Είναι λογικό αν συμπεριλάβει κανείς την ικανότητα των άλλων αλγορίθμων να προσαρμόζουν τις παραμέτρους τους ανάλογα με τις τιμές που χρησιμοποιούνται για την εκπαίδευση του μοντέλου. Γενικά το ανάπτυγμα Τέιλορ και συγκεκριμένα αυτό το οποίο προτείνει ο Νακαγιαμά (Okihiko Nakayama, 1999) δεν εμπεριέχει καμία τέτοια παράμετρο.

5.3.1 Γωνιακή ταχύτητα

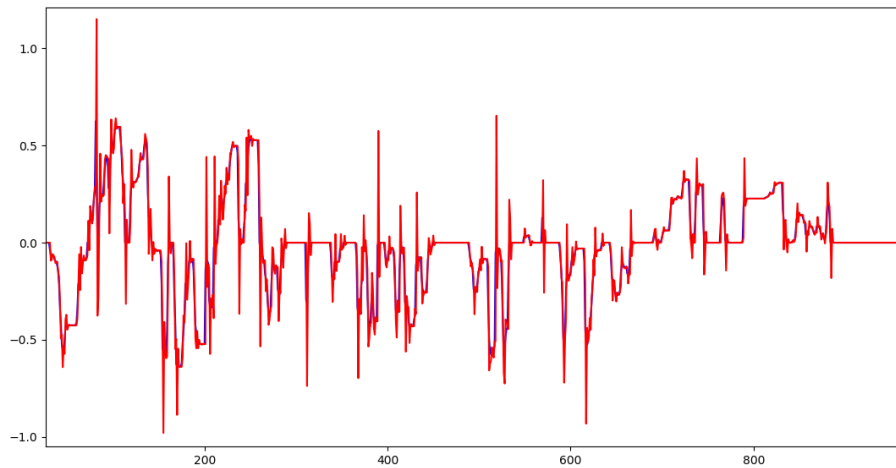
Πίνακας 16. Σφάλματα μοντέλου αναπτύγματος Τέιλορ για την γωνιακή ταχύτητα

MSE test: 0.009, RMSE test: 0.093

MSE workload: 0.068, RMSE workload: 0.260



Εικόνα 42. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ γωνιακής ταχύτητας στο σετ ελέγχου



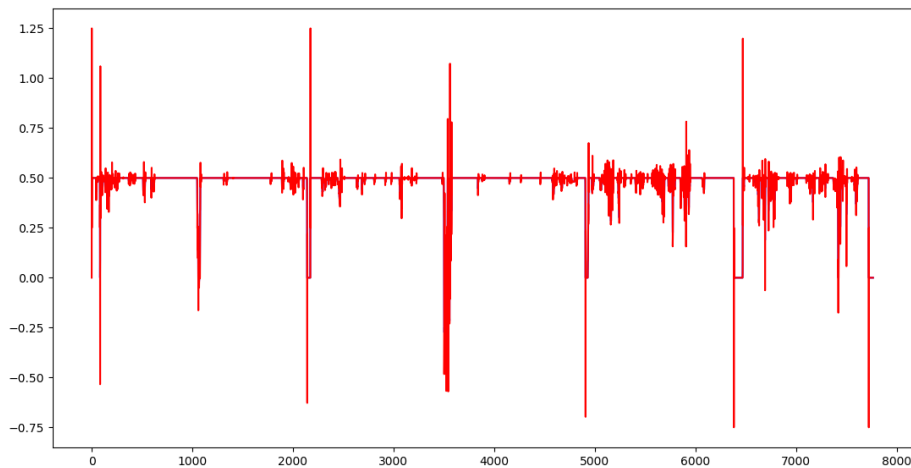
Εικόνα 43. Πρόβλεψη μοντέλου αναπτύγματος Τέιλο γωνιακής ταχύτητας στο σετ ελέγχου σε μεγέθυνση

5.3.2 Γραμμική ταχύτητα

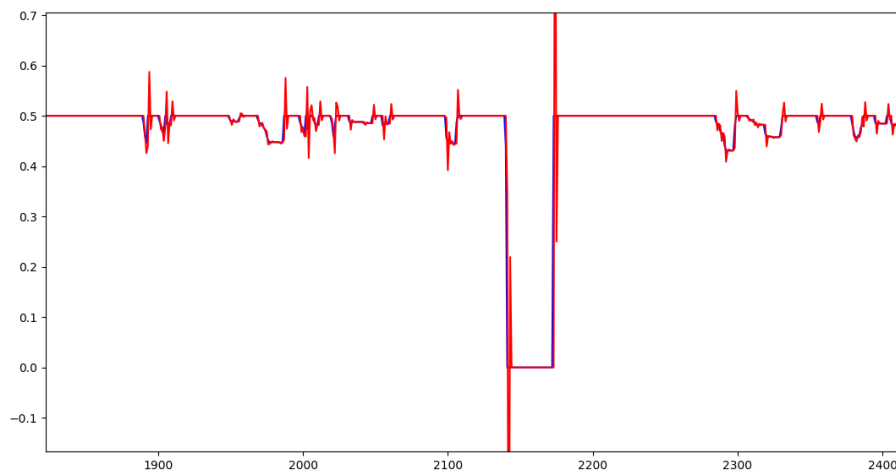
Πίνακας 17. Σφάλματα μοντέλου αναπτύγματος Τέιλο για την γραμμική ταχύτητα

MSE test: 0.001, RMSE test: 0.037

MSE workload: 0.019, RMSE workload: 0.138



Εικόνα 44. Πρόβλεψη μοντέλου αναπτύγματος Τέιλο γραμμικής ταχύτητας στο σετ ελέγχου



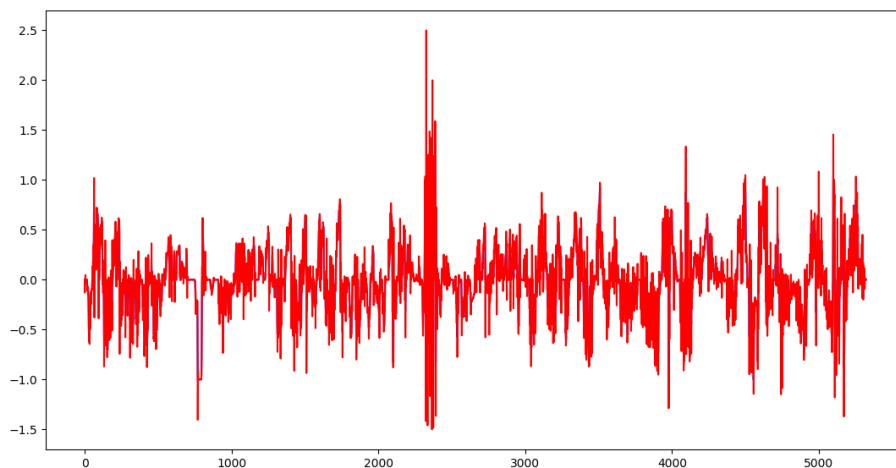
Εικόνα 45. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ γραμμικής ταχύτητας στο σετ ελέγχου σε μεγέθυνση

5.3.3 Εντολές χειριστή δεξιά/αριστερά

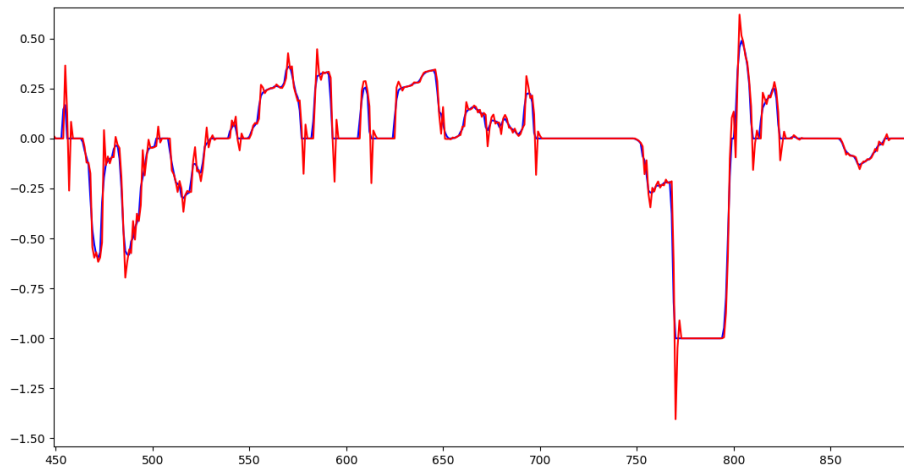
Πίνακας 18. Σφάλματα μοντέλου αναπτύγματος Τέιλορ για τις δεξιά/αριστερά εντολές

MSE test: 0.014, RMSE test: 0.120

MSE workload: 0.083, RMSE workload: 0.288



Εικόνα 46. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ δεξιών/αριστερών εντολών στο σετ ελέγχου



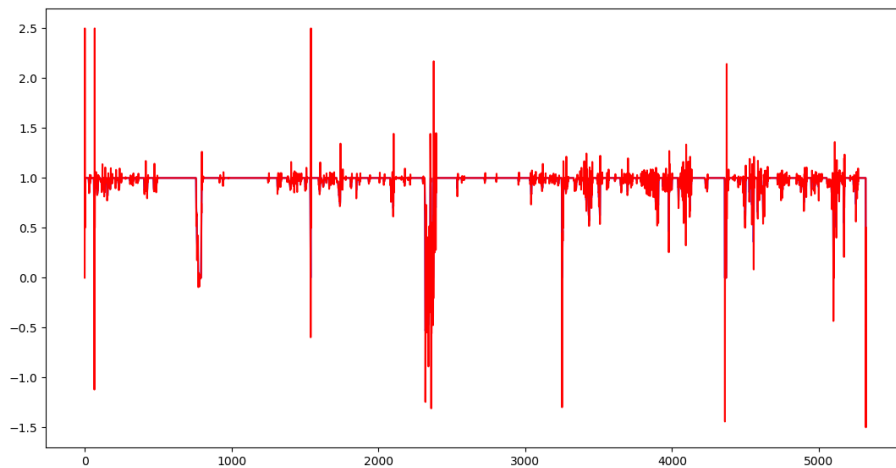
Εικόνα 47. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ δεξιών/αριστερών εντολών στο σετ ελέγχου σε μεγέθυνση

5.3.4 Εντολές χειριστή μπρος/πίσω

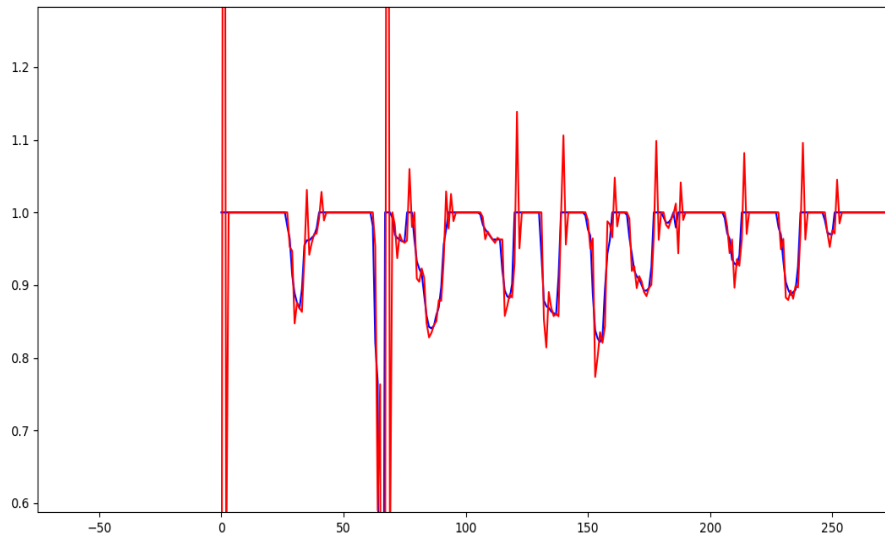
Πίνακας 19. Σφάλματα μοντέλου αναπτύγματος Τέιλορ για τις πάνω/κάτω εντολές

MSE test: 0.007, RMSE test: 0.086

RMSE workload: 0.073, RMSE workload: 0.270



Εικόνα 48. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ πάνω/κάτω εντολών στο σετ ελέγχου



Εικόνα 49. Πρόβλεψη μοντέλου αναπτύγματος Τέιλορ πάνω/κάτω εντολών στο σετ ελέγχου σε μεγέθυνση

5.4 Random Forest

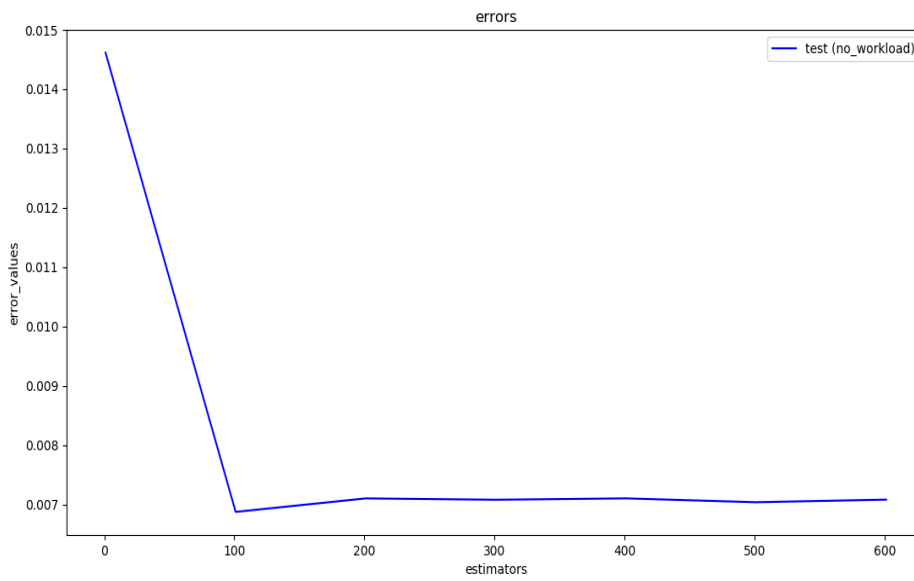
Όπως και στο αυτοπαλίνδρομο μοντέλο, έτσι και για το μοντέλο random forest δημιουργήθηκε ένας αλγόριθμος ο οποίος δημιουργεί το μοντέλο με $K=\{1,101,201,\dots\}$ δέντρα αποφάσεων, ώστε να βρεθούν οι αριθμοί των δέντρων αποφάσεων που έχουν το μικρότερο μέσο σφάλμα. Παρακάτω παρουσιάζονται πίνακες με τον αριθμό των δέντρων καθώς και τα σφάλματα σε σετ εκπαίδευσης χωρίς φόρτο (no_workload) και μετά σε σετ με φόρτο εργασίας (workload). Για κάποια μοντέλα δεν χρειάστηκε να ανέβει υπερβολικά ο αριθμός των δέντρων διότι δεν προσέφερε κάτι παραπάνω. Δηλαδή το σφάλμα δεν κατέβαινε άλλο.

5.4.1 Γωνιακή ταχύτητα

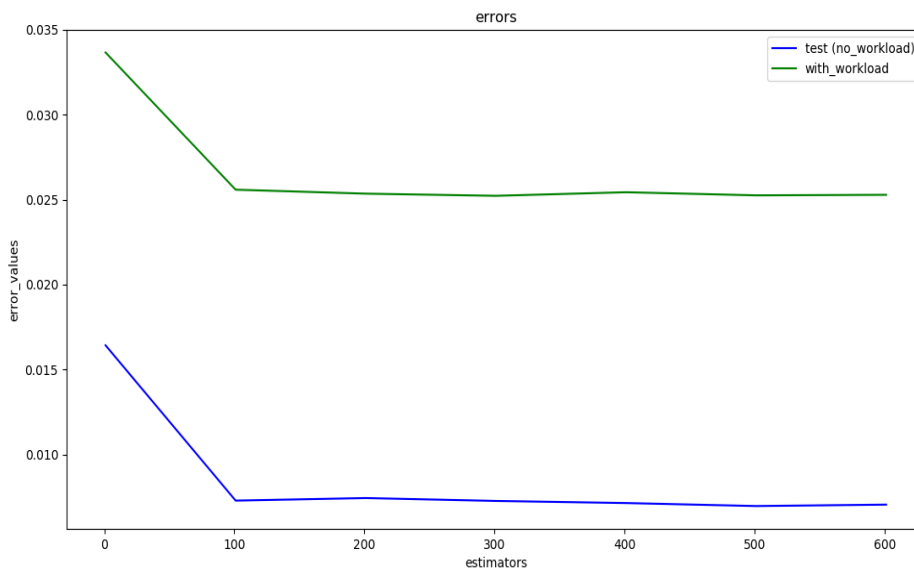
Πίνακας 20. Σφάλματα μοντέλου random forest γωνιακής ταχύτητας με βάση τον αριθμό των δέντρων

MSE no_workload 1 : 0.016426847057449765
RMSE no_workload 1 : 0.1281672620346154
MSE workload 1 : 0.03366528118953824
RMSE workload 1 : 0.18348101043306425
MSE no_workload 101: 0.0072855639784131215
RMSE no_workload 101: 0.08535551521965713

MSE workload 101 : 0.025586738075737324 RMSE workload 101 : 0.15995855111789842
MSE no_workload 201 : 0.007434953702013739 RMSE no_workload 201 : 0.08622617759134252 MSE workload 201 : 0.0253521701400275 RMSE workload 201 : 0.1592236481808764
MSE no_workload 301 : 0.007265957589321827 RMSE no_workload 301 : 0.0852405865144171 MSE workload 301 : 0.025225625687952974 RMSE workload 301 : 0.15882577148546445
MSE no_workload 401 : 0.007141535941887994 RMSE no_workload 401 : 0.08450760878103222 MSE workload 401 : 0.025436432293705458 RMSE workload 401 : 0.15948803181964927
MSE no_workload 501 : 0.006965301789275739 RMSE no_workload 501 : 0.08345838357694055 MSE workload 501 : 0.025253096715138784 RMSE workload 501 : 0.15891222959589607
MSE no_workload 601 : 0.007048955936092949 RMSE no_workload 601 : 0.08395806057843969 MSE workload 601 : 0.025283951760450486 RMSE workload 601 : 0.15900928199463857



Εικόνα 50. Μέσο τετραγωνικό σφάλμα και αριθμός δέντρων μοντέλου



Εικόνα 51. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς φόρτο χειριστή

5.4.2 Γραμμική ταχύτητα

Πίνακας 21. Σφάλματα μοντέλου random forest γραμμικής ταχύτητας με βάση τον αριθμό των δέντρων

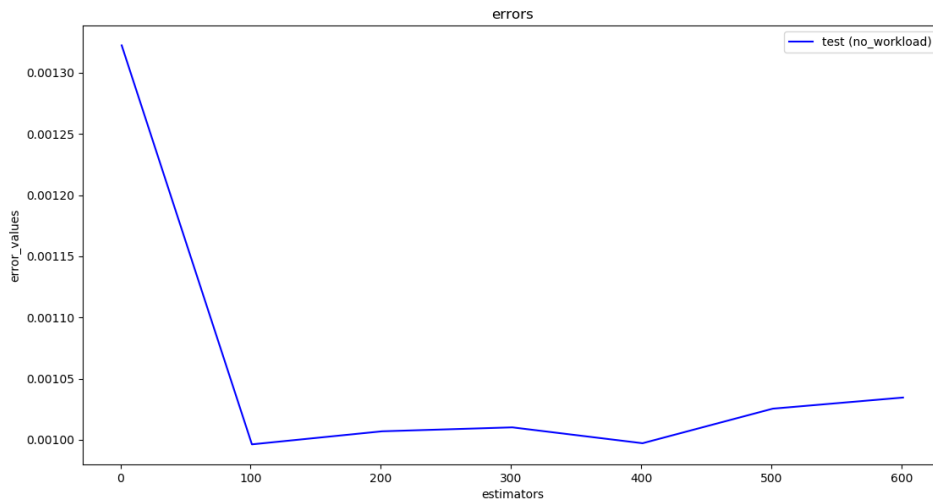
MSE no_workload 1 : 0.0013224204474039819
RMSE no_workload 1 : 0.03636509930419525
MSE workload 1 : 0.012023453632326828
RMSE workload 1 : 0.10965150994093437
MSE no_workload 101 : 0.0009963452370273752
RMSE no_workload 101 : 0.031564936829136456
MSE workload 101 : 0.009295557970557455
RMSE workload 101 : 0.09641347400938033
MSE no_workload 201 : 0.0010070123189338655
RMSE no_workload 201 : 0.0317334574059283
MSE workload 201 : 0.009333354363029577
RMSE workload 201 : 0.09660928714688655
MSE no_workload 301 : 0.0010102249307058892
RMSE no_workload 301 : 0.03178403578380016
MSE workload 301 : 0.009214074427584325
RMSE workload 301 : 0.09598997045308601
MSE no_workload 401 : 0.0009972608532692812
RMSE no_workload 401 : 0.03157943719050865
MSE workload 401 : 0.009209226582609443
RMSE workload 401 : 0.09596471529999681
MSE no_workload 501 : 0.0010255076591085288
RMSE no_workload 501 : 0.03202354850900394
MSE workload 501 : 0.009441233944142638
RMSE workload 501 : 0.09716601228898218

MSE no_workload 601 : 0.0010345945699343595

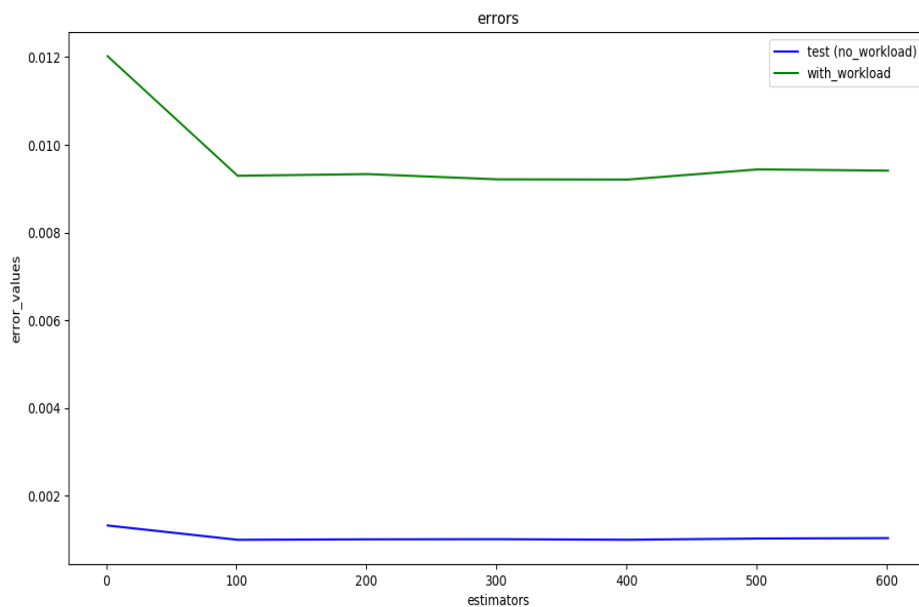
RMSE no_workload 601 : 0.03216511417567734

MSE workload 601 : 0.009412238672295542

RMSE workload 601 : 0.0970166927507609



Εικόνα 52. Μέσο τετραγωνικό σφάλμα και αριθμός δέντρων μοντέλου γραμμικής ταχύτητας



Εικόνα 53. Διαφορά στα σφάλματα μοντέλου γραμμικής ταχύτητας μεταξύ πειραμάτων με φόρτο και χωρίς φόρτο χειριστή

5.4.3 Εντολές χειριστή μπρος/πίσω

Πίνακας 22. Σφάλματα μοντέλου random forest με βάση τον αριθμό των δέντρων

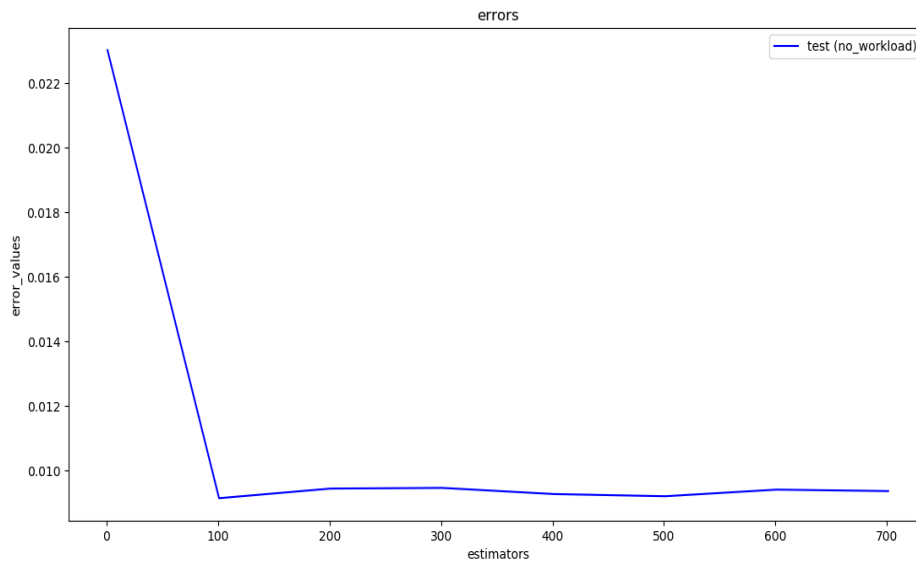
MSE no_workload 1 : 0.023020047465424468 RMSE no_workload 1 : 0.15172358902103678 MSE workload 1 : 0.0491987036853882 RMSE workload 1 : 0.2218078079901341
MSE no_workload 101 : 0.009145677676568561 RMSE no_workload 101 : 0.09563303653324284 MSE workload 101 : 0.034632353121220816 RMSE workload 101 : 0.18609769778592322
MSE no_workload 201 : 0.009443889380063226 RMSE no_workload 201 : 0.09717967575611285 MSE workload 201 : 0.034597833346857454 RMSE workload 201 : 0.186004928286477
MSE no_workload 301 : 0.00946444963717794 RMSE no_workload 301 : 0.09728540300157028 MSE workload 301 : 0.03423410119202296 RMSE workload 301 : 0.18502459618121847
MSE no_workload 501 : 0.009206380664174608 RMSE no_workload 501 : 0.09594988621241095 MSE workload 501 : 0.034304461738623865 RMSE workload 501 : 0.1852146369448804
MSE no_workload 601 : 0.009411173358496996 RMSE no_workload 601 : 0.09701120223199482 MSE workload 601 : 0.034499106287751734 RMSE workload 601 : 0.185739350401986

MSE no_workload 701 : 0.009366183910272198

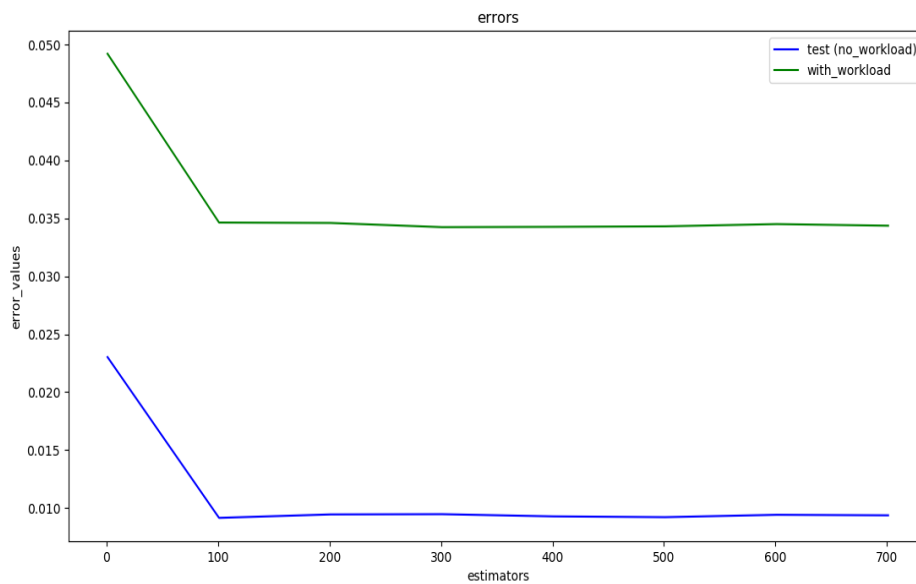
RMSE no_workload 701 : 0.09677904685556786

MSE workload 701 : 0.03435933373508466

RMSE workload 701 : 0.18536270858801307



Εικόνα 54. Μέσο τετραγωνικό σφάλμα και αριθμός δέντρων μοντέλου



Εικόνα 55. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς φόρτο χειριστή

5.4.4 Εντολές χειριστή πάνω/κάτω

Πίνακας 23. Σφάλματα μοντέλου random forest με βάση τον αριθμό των δέντρων

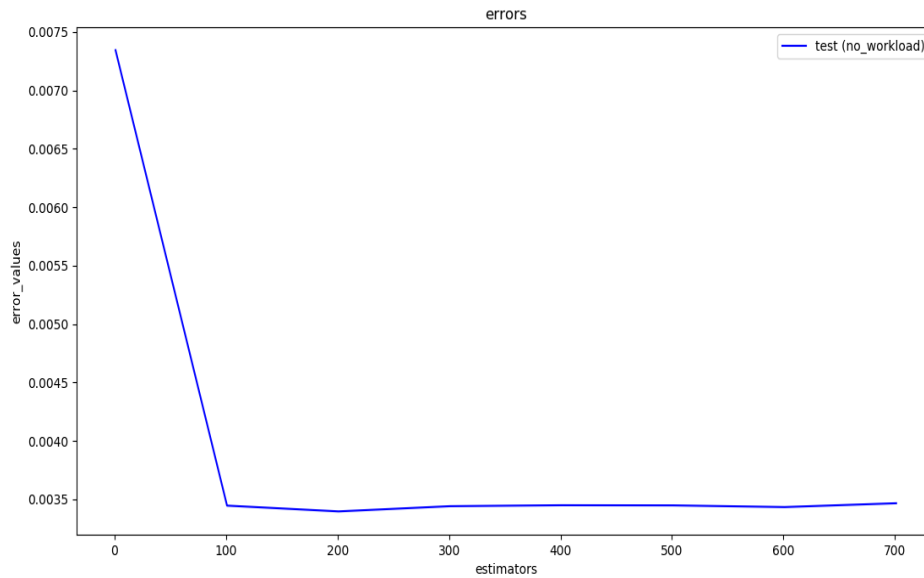
MSE no_workload 1 : 0.007344759646510025
RMSE no_workload 1 : 0.08570157318573578
MSE workload 1 : 0.0547819297587723
RMSE workload 1 : 0.23405539890968613
MSE no_workload 101 : 0.0034455562994395545
RMSE no_workload 101 : 0.05869886114261123
MSE workload 101 : 0.037497602928749436
RMSE workload 101 : 0.19364297800010574
MSE no_workload 201 : 0.0033958829949884512
RMSE no_workload 201 : 0.05827420522828648
MSE workload 201 : 0.037222668151602654
RMSE workload 201 : 0.19293177071597786
MSE no_workload 301 : 0.003440706860890384
RMSE no_workload 301 : 0.05865753882401122
MSE workload 301 : 0.037164684993484125
RMSE workload 301 : 0.19278144359217805
MSE no_workload 401 : 0.0034488095759571714
RMSE no_workload 401 : 0.05872656618564695
MSE workload 401 : 0.037303107735081296
RMSE workload 401 : 0.1931401246118509
MSE no_workload 501 : 0.0034474972382324528
RMSE no_workload 501 : 0.058715391834104735
MSE workload 501 : 0.03694140019027137
RMSE workload 501 : 0.1922014573052748

MSE no_workload 701 : 0.0034659125133683663

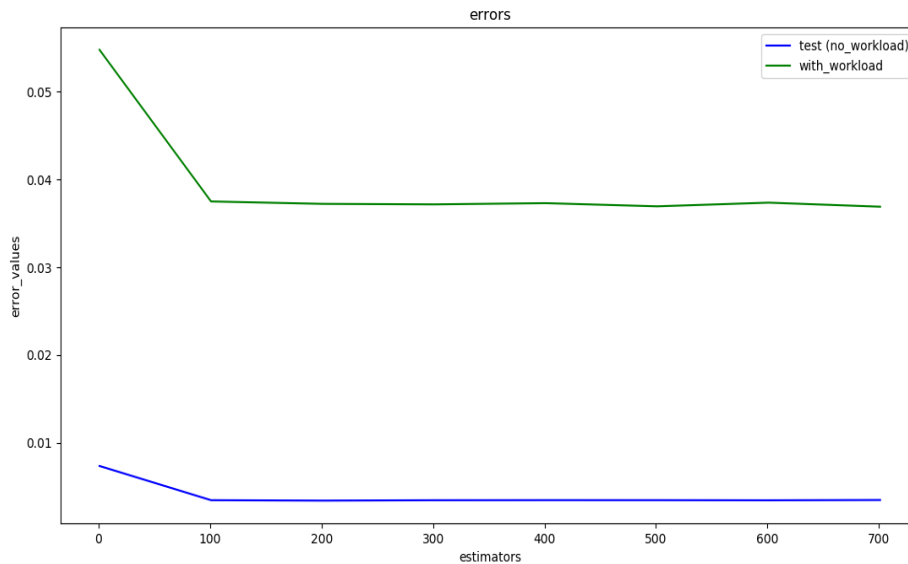
RMSE no_workload 701 : 0.05887200109872575

MSE workload 701 : 0.0368980323363167

RMSE workload 701 : 0.19208860543071446



Εικόνα 56. Μέσο τετραγωνικό σφάλμα και αριθμός δέντρων μοντέλου



Εικόνα 57. Διαφορά στα σφάλματα μεταξύ πειραμάτων με φόρτο και χωρίς φόρτο χειριστή

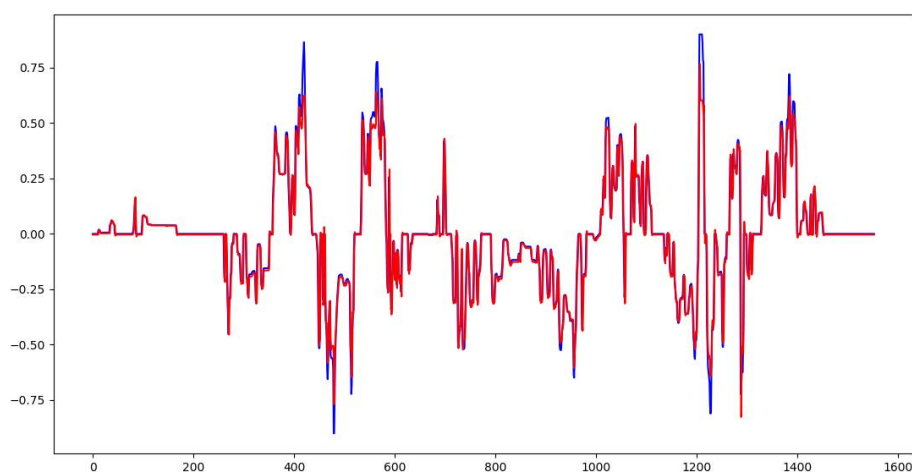
5.5 Μηχανές διανυσμάτων υποστήριξης

Στις μηχανές διανυσμάτων υποστήριξης άρχισαν να εμφανίζονται κάποια προβλήματα. Λόγο σχετικής υψηλής πολυπλοκότητας, στο η συνάρτηση κόστους του μοντέλου ήταν δύσκολο να ελαχιστοποιηθεί, με αποτέλεσμα τα μοντέλα να μην είναι αποδοτικά και να μην προσεγγίζουν την πραγματική έξοδο με ακρίβεια.

5.5.1 Γωνιακή ταχύτητα

Πίνακας 24. Μέσα τετραγωνικά σφάλματα και παράμετροι

```
MSE no_workload 0.008674394039708077, RMSE no_workload 0.09236954029177085
MSE workload 0.0261651497579178, RMSE workload 0.16175645198234845
c=0.1632, gamma=0.354, epsilon=0.1037
```

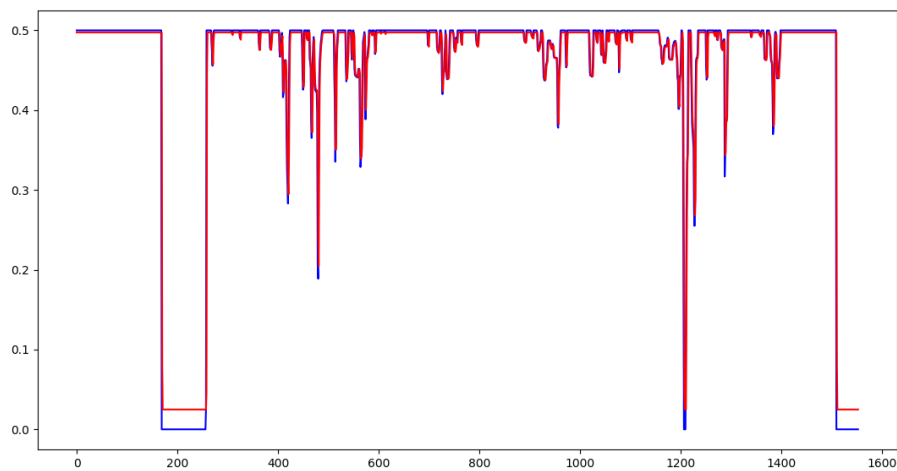


Εικόνα 58. Πρόβλεψη μοντέλου SVR

5.5.2 Γραμμική ταχύτητα

Πίνακας 25. Μέσα τετραγωνικά σφάλματα και παράμετροι

```
MSE no_workload 0.0008285399022633013, RMSE no_workload 0.028784369061407293
MSE workload 0.008017144870054801, RMSE workload 0.08953851054186016
c=0.241, gamma=0.375, epsilon=0.03
```

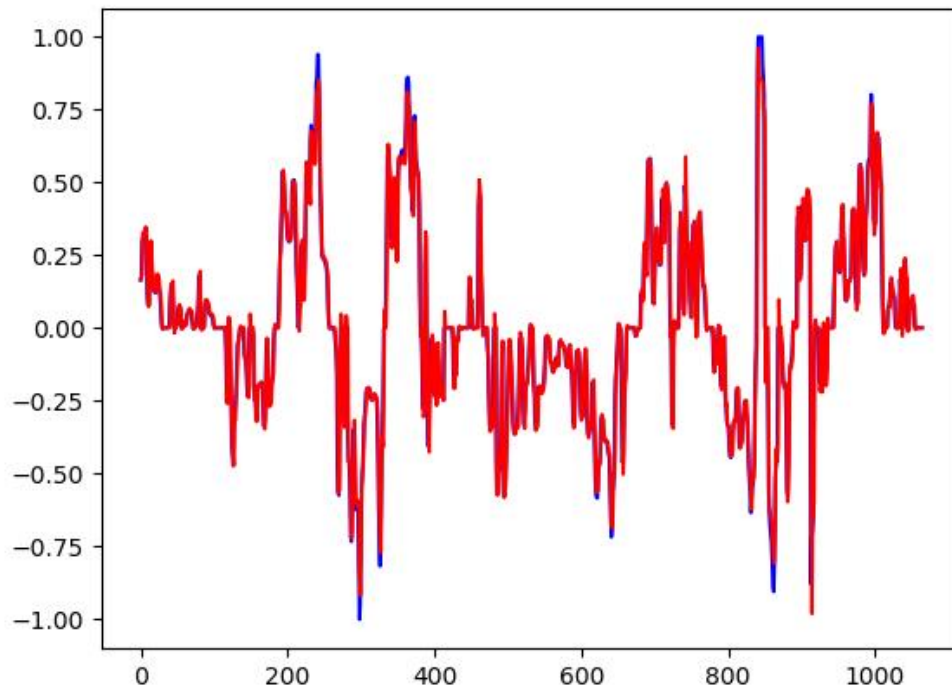


Εικόνα 59. Πρόβλεψη μοντέλου SVR

5.5.3 Εντολές χειριστή δεξιά/αριστερά

Πίνακας 26. Μέσα τετραγωνικά σφάλματα και παράμετροι

MSE no_workload 0.0075858898673123, RMSE no_workload 0.08709701411249
MSE no_workload 0.0321072131977769, RMSE no_workload 0.1791848576129
c=0.12, gamma=0.25, epsilon=0.026

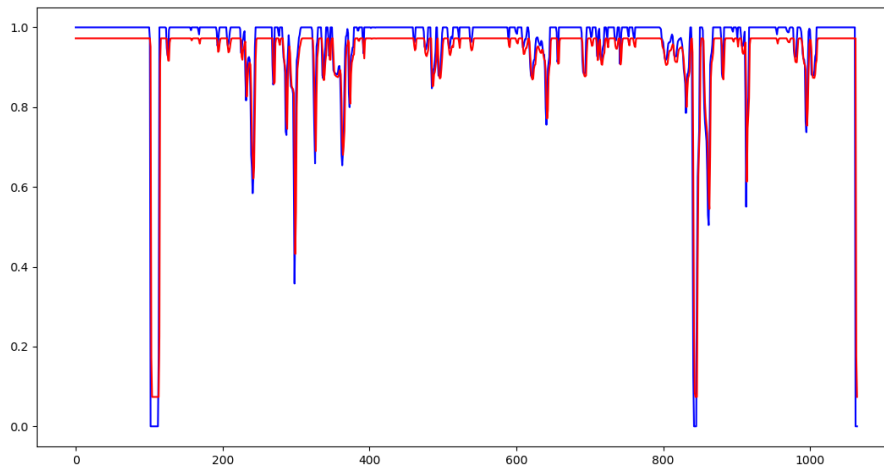


Εικόνα 60. Πρόβλεψη μοντέλου SVR

5.5.4 Εντολές χειριστή πάνω/κάτω

Πίνακας 27. Μέσα τετραγωνικά σφάλματα και παράμετροι

MSE no_workload 0.004765728881131914, RMSE no_workload 0.06903425874978245
MSE no_workload 0.03453265292070079, RMSE no_workload 0.18582963412949183
c=0.1632, gamma=0.3541, epsilon=0.1037



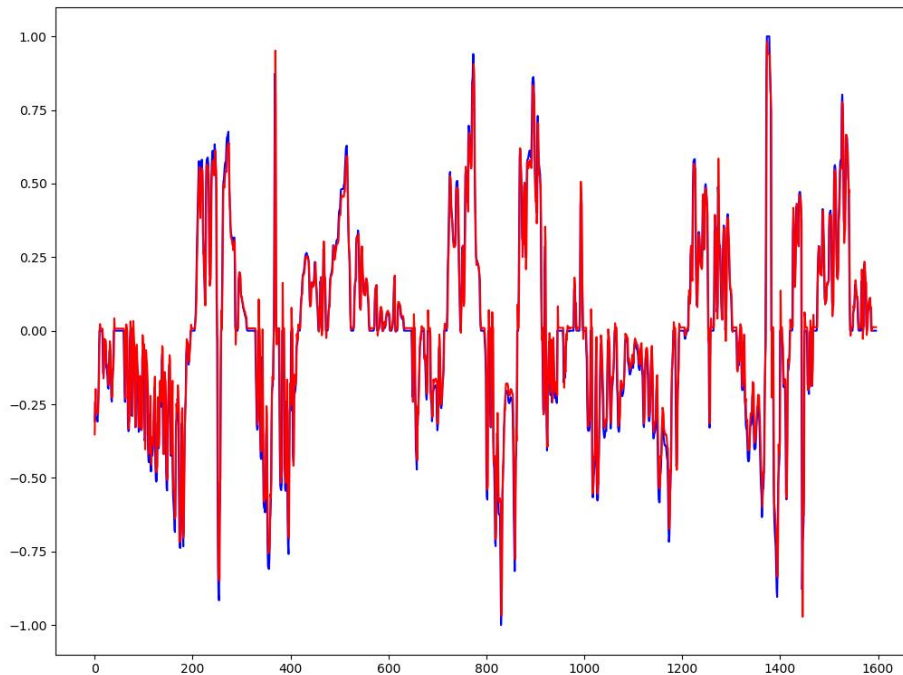
Εικόνα 61. Πρόβλεψη μοντέλου SVR

5.6 Νευρωνικά δίκτυα

Όπως και με τις μηχανές διανυσμάτων υποστήριξης, έτσι και εδώ οι παράμετροι των μοντέλων δεν μπορούν να πάρουν τις καλύτερες δυνατές τιμές με αποτέλεσμα τα μοντέλα να μην είναι τόσο αποδοτικά όσο θα μπορούσαν να είναι.

Πίνακας 28. Αρχιτεκτονική νευρωνικού δικτύου

Κρυφά επίπεδα	1
Νευρώνες εισόδου	2
Νευρώνες εξόδου	1
Αρχικοποίηση βαρών	Τυχαία
Συνάρτηση ενεργοποίησης	Σιγμοειδής
Εποχές εκπαίδευσης	50



Εικόνα 62. Πρόβλεψη νευρωνικού δικτύου στο σετ εκπαίδευσης με μικρό σφάλμα

5.7 Κυριότερα ευρήματα/ αποτελέσματα

Αρχικά πρέπει να ειπωθούν κάποια πράγματα για τα δεδομένα. Η γραμμική ταχύτητα και η εμπρός/πίσω εντολή του χειριστή έχουν πολύ απλή μορφή. Οι περισσότερες μετρήσεις είναι είτε στην μπροστά είτε στην πίσω θέση του χειριστηρίου με αποτέλεσμα οι γραφικές να έχουν μια απλή σχεδόν βηματική μορφή. Γι' αυτό και τα περισσότερα μοντέλα δεν πάσχιζαν καθόλου να προσεγγίσουν αυτή την μορφή. Προφανώς όσον αφορά τις δεξιά/αριστερά εντολές και την γωνιακή ταχύτητα δεν ισχύει αυτό. Η μορφή τους είναι σχεδόν μη γραμμική πράγμα που καθιστά την προσέγγιση τους λίγο πιο δύσκολη από ότι υπόλοιπα δεδομένα.

Το μοντέλο κατάβασης δυναμικού και το αυτοπαλίνδρομο μοντέλο παρουσίασαν τα μικρότερα μέσα τετραγωνικά σφάλματα της τάξης του 0,003 για τις γωνιακές ταχύτητες και τις δεξιά/αριστερά εντολές χειριστή. Τόσο το αυτοπαλίνδρομο μοντέλο, όσο και το μοντέλο κατάβασης δυναμικού συμπεριλάμβαναν και τις προηγούμενες τιμές των δεδομένων. Αυτό είναι ένας από τους λόγους που αυτά τα μοντέλα μπόρεσαν να προσεγγίσουν τόσο καλά τα δεδομένα των πειραμάτων.

Δυστυχώς τα μοντέλα με την χαμηλότερη απόδοση ήταν το μοντέλο Τέιλορ, το μοντέλο random forest και το νευρωνικό δίκτυο. Για το μοντέλο Τέιλορ ο κύριος λόγος της χαμηλής του απόδοσης είναι η αδυναμία της εκπαίδευσης, δηλαδή της προσαρμογής κάποιων παραμέτρων. Το μοντέλο αυτό σε αντίθεση με όλα τα άλλα, δημιουργεί όρους διαφοράς μεταξύ προηγούμενων τιμών. Δεν είναι όμως αρκετό για να προσεγγίσει σε ικανοποιητικό βαθμό τα δεδομένα. Όσον αφορά το νευρωνικό δίκτυο και το μοντέλο random forest η αδυναμία για ακόμα μια φορά βρίσκεται στις παραμέτρους. Αυτά τα δύο μοντέλα λόγω της πολυπλοκότητας τους και των πολλών παραμέτρων τους είναι δύσκολο να εκπαιδευτούν. Για παράδειγμα ένα νευρωνικό μπορεί να έχει από κανένα μέχρι x αριθμούς κρυφών επιπέδων ή μπορεί να έχει σε κάθε επίπεδο μια από τις πολλές συναρτήσεις ενεργοποίησης. Αυτό καθιστά πολύ δύσκολη την επιλογή των σωστών παραμέτρων.

Προφανώς υπάρχουν τρόποι διαφυγής αυτού του προβλήματος. Υπάρχουν αλγόριθμοι που δοκιμάζουν πολυάριθμες παραμέτρους και στο τέλος εμφανίζουν τις σωστές παραμέτρους. Αυτοί οι αλγόριθμοι χρειάζονται τρομερή υπολογιστική δύναμη και πολύ χρόνο για να παρουσιάσουν κάποια αποτελέσματα. Το μοντέλο των μηχανών διανυσμάτων υποστήριξης, αν και εξίσου δύσκολο να βρεθούν οι παράμετροί του, εμφάνισαν σχετικά ικανοποιητικά αποτελέσματα. Μπορούν σε μεγάλο βαθμό να προσεγγίσουν την αποδοτική και ομαλή οδήγηση του ρομπότ (Εικόνα 58).

Κεφάλαιο 6. Συζήτηση – Συμπεράσματα – Μελλοντικές επεκτάσεις

6.1 Ανακεφαλαίωση

Αυτή η πτυχιακή εργασία πραγματεύεται αλγόριθμους για την πρόβλεψη των σωστών ενεργειών των χειριστών σε προσομοιωμένο ρομπότ. Ουσιαστικά οι σωστές ενέργειες είναι δεδομένα από πειράματα αποδοτικής οδήγησης. Σκοπός ήταν η δημιουργία μαθηματικών και μοντέλων μηχανικής μάθησης και η εύρεση του πιο αποδοτικού μοντέλου. Δηλαδή του μοντέλου που είναι το πιο απλό να υλοποιηθεί, με όσο το δυνατόν λιγότερες παραμέτρους, για να μην έχει μεγάλη καθυστέρηση όταν υπολογίζετε το σφάλμα πραγματικής και προβλεπόμενης οδήγησης σε πραγματικό χρόνο. Με την βοήθεια του *ros* ήταν δυνατή η μεταφορά των δεδομένων καθώς και η ίδια η συλλογή δεδομένων. Μετά από μια προεργασία, τα δεδομένα χωρίστηκαν και δοκιμάστηκαν σε μαθηματικά μοντέλα και σε μοντέλα μηχανικής μάθησης.

6.2 Συζήτηση / Συμπεράσματα

Τα αποτελέσματα για τα περισσότερα μοντέλα ήταν καθοριστικά. Όμως, η δυσκολία εύρεσης των κατάλληλων παραμέτρων για μερικά μοντέλα ήταν μεγάλη, με αποτέλεσμα τα να μην μπορέσουν να προσεγγίζουν τα δεδομένα. Αυτό συμβαίνει για δύο λόγους. Αρχικά όπως προαναφέρθηκε μερικά από τα παραπάνω μοντέλα έχουν πάρα πολλές παραμέτρους. Για παράδειγμα ένα νευρωνικό, ανάλογα με την αρχιτεκτονική του έχει συγκεκριμένο αριθμό νευρώνων, επιπέδων και συναρτήσεων ενεργοποίησης εκ των οποίων όλα συμβάλουν στην ακρίβεια του μοντέλου. Επιπρόσθετα για να μπορούν να εκπαιδευτούν τέτοια μοντέλα πρέπει να υπάρχει και ο κατάλληλος αριθμός δεδομένων εκπαίδευσης. Πρακτικά αυτά τα μοντέλα υλοποιήθηκαν με 5000 χρονικές στιγμές δύο δεδομένων, την ταχύτητα ή την εντολή από το χειριστήριο, την χρονική στιγμή $t-1$ και $t-2$.

Από την άλλη τα αποτελέσματα των υπόλοιπων μοντέλων έδωσαν πολύ θετικά αποτελέσματα και επιβεβαίωσαν την υπόθεση ότι η διαφορά μεταξύ των σφαλμάτων θα είναι πολύ μεγάλη και συγκεκριμένα πάνω από πενταπλάσια. Αυτή η διαφορά μεταξύ των σφαλμάτων θα μπορούσε να χρησιμοποιηθεί για να μπορεί ο χειριστής ή και το ίδιο το ρομπότ να αξιολογήσει την εκάστοτε οδήγηση/χειρισμό.

6.3 Πρακτικές προεκτάσεις της έρευνας

Τα αποτελέσματα της παραπάνω ανάλυσης μπορούν να χρησιμοποιηθούν με πολλούς τρόπους. Στην πράξη η μοντελοποίηση αυτών των συγκεκριμένων διαδικασιών είναι μεγάλης σημασίας διότι μπορούν, να χρησιμοποιηθούν για την ανάπτυξη εφαρμογών μεικτής αυτονομίας, σε ρομποτικές εφαρμογές διάσωσης, πράγμα που σε πραγματικές καταστάσεις θα μπορούσε να σώσει ζωές .

Η γνώση που προσκομίσθηκε από τα μοντέλα μπορεί να χρησιμοποιηθεί μελλοντικά για την δημιουργία πληροφορίας που ο υπολογιστής μπορεί να επεξεργαστεί. Όπως έχει γίνει σε αρκετές έρευνες (Boer, 2000) (Okihiko Nakayama, 1999) αυτά τα μοντέλα μπορούν, ανάλογα με την εφαρμογή να χρησιμοποιηθούν, ώστε να είναι δυνατή η δημιουργία κάποιας νέας μεθόδου, παρόμοια της εντροπίας (Boer, 2000).

Μελλοντικές προεκτάσεις αυτής της έρευνας θα μπορούσαν να είναι αρκετές, αλλά η βασική ιδέα είναι η δημιουργία ρομπότ το οποίο θα χρησιμοποιεί τα μοντέλα αυτά ώστε να μπορεί να αξιολογήσει τον χειριστή με βάση το πόσο κοντά είναι στην προβλεπόμενη αποδοτική οδήγηση. Πρακτικά το σφάλμα που θα έχει η πραγματική οδήγηση με την προβλεπόμενη αποδοτική οδήγηση είναι αυτό που δείχνει και τα αποτελέσματα. Με μια κατηγοριοποίηση των σφαλμάτων με την χρήση συναρτήσεων πυρήνα θα ήταν δυνατό να εμφανίζεται σε πραγματικό χρόνο μια αξιολόγηση που θα κάνει το ρομπότ στον άνθρωπο (ή ακόμα και το ρομπότ για τον εαυτό του), μέσω μιας διεπαφής ανθρώπου μηχανής.

Βιβλιογραφικές Αναφορές

- Anthony, M. (1997). *Artificial Neural Networks*.
- AtsushiSakai. (χ.χ.). *Rosbag to csv*. Ανάκτηση από GitHub.
- Baker, M. C. (2004). *Improved interfaces for human-robot*.
- Beer, J. M. (2014). *Toward a Framework for Levels of Robot Autonomy in Human-Robot Interaction*.
- Boer, E. R. (2000). *Behavioral Entropy as an Index of Workload*.
- Brownlee, J. (χ.χ.). *How to Prepare Data For Machine Learning*. Ανάκτηση από Machine learning mastery.
- Dyer, C. (2002, 04 24). *Theory of the Taylor Expansion*. Ανάκτηση από <http://pathfinder.scar.utoronto.ca>.
- Emmanouil, C. (2017). *FLEXIBLE ROBOTIC CONTROL VIA CO-OPERATION*. Birmingham.
- Endsley, M. R. (1999). *Level of automation effects on performance, situation awareness and workload in a dynamic control task*.
- Eric Farmer & Adam Brownson, Q. (2003). *Review of Workload Measurement, Analysis and Interpretation*.
- Fong, T. T. (2003). *Robot, asker of questions. Robotics and Autonomous*.
- Hart, S. G. (1988). *Development of NASA-TLX (Task Load Index): Results*.
- Louppe, G. (2014). *Understanding Random Forests*.
- Manolis Chiou, G. B. (2017). *Human-Initiative Variable Autonomy: an Experimental Analysis of the*.
- Manolis Chiou, N. H. (2015). *Towards the Principled Study of Variable Autonomy*. Birmingham.
- Manolis Chiou, R. S. (2016). *Experimental Analysis of a Variable Autonomy Framework for*. Birmingham.
- O’Kane, J. M. (2014). *A Gentle Introduction to ROS*.
- Okihiko Nakayama, T. F. (1999). *Development of a Steering Entropy Method*.
- Ros Wiki. (2018). Ανάκτηση από wiki.ros.org.
- Schindler, A. (2017). *Comparing Shallow versus Deep Neural Network*.

- Scholkopf, A. J. (2003). *A Tutorial on Support Vector Regression*.
- Sheridan, T. B. (1978). *Human and computer control of undersea teleoperators*.
- Sheridan, T. B. (1989). *Telerobotics*.
- Support Vector Machine - Regression (SVR)*. (χ.χ.). Ανάκτηση από Saedsayad.com.
- Thomas B Sheridan, L. C. (1989). *Supervisory control, mental models and decision aids*.
- What is ROS?* (χ.χ.). Ανάκτηση από roboticsbackend.com.
- Yanco, H. D. (2004). *Beyond Usability Evaluation: Analysis of Human-Robot Interaction at a Major Robotics Competition*.
- Γαζή, Σ. (2015). *Γραμμικά Μοντέλα Χρονοσειρών και Αυτοσυσχέτισης*.
- Διαμαντάρας, Κ. (2007). *Τεχνητά Νευρωνικά Δίκτυα*. Αθήνα: Εκδόσεις Κλειδάριθμος.