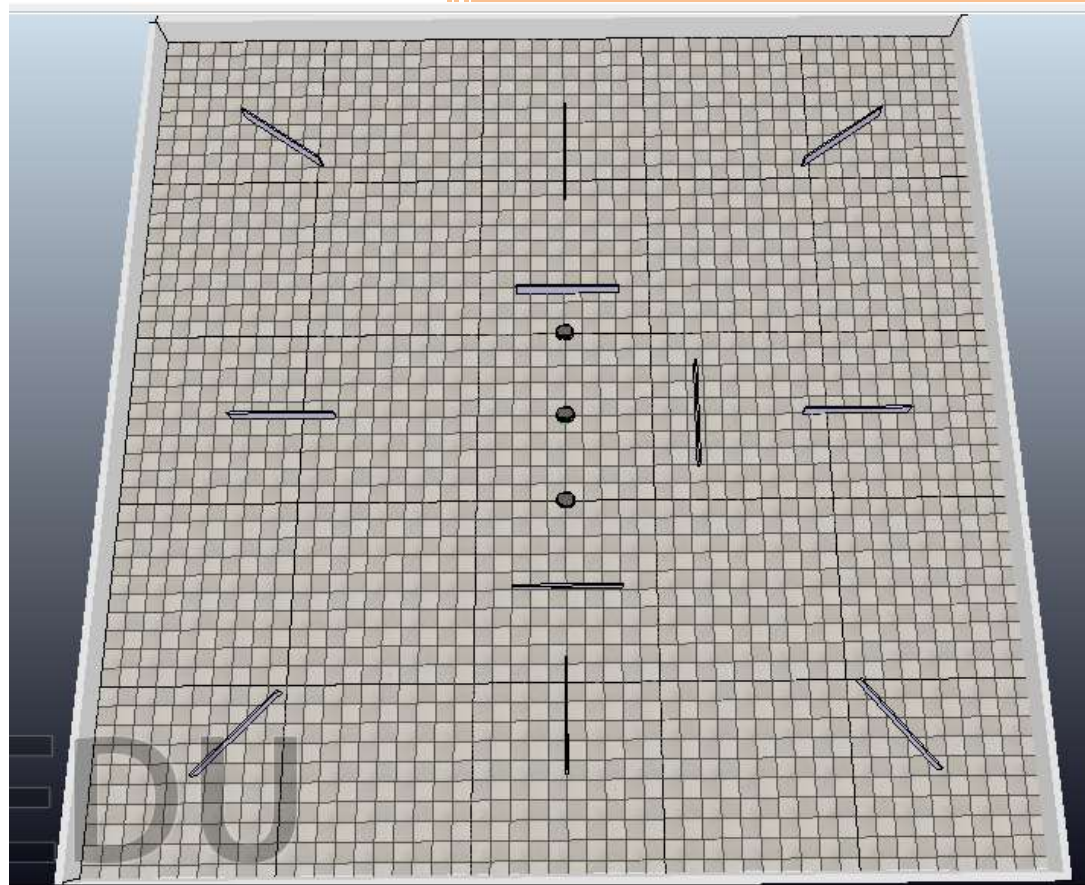




**Τεχνολογικό Εκπαιδευτικό Ίδρυμα  
Κρήτης**  
*Διατμηματικό Πρόγραμμα Προηγμένων  
Συστημάτων Παραγωγής,  
Αυτοματισμού και Ρομποτικής*

2019



**Βελτιστοποίηση της Συνεργατικής Συμπεριφοράς Ομάδων  
Ρομποτικών Οχημάτων με τη βοήθεια Εξελικτικών Στρατηγικών**

Αθανάσιος Μαρτίνης





Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης  
Διατμηματικό Πρόγραμμα Προηγμένων  
Συστημάτων Παραγωγής, Αυτοματισμού  
και Ρομποτικής

## Βελτιστοποίηση της Συνεργατικής Συμπεριφοράς Ομάδων Ρομποτικών Οχημάτων με τη βοήθεια Εξελικτικών Στρατηγικών

Αθανάσιος Μαρτίνης

Επιβλέπων Καθηγητής:  
Αναπληρωτής Καθηγητής Ε. Δοϊτσίδης  
Τμήμα Ηλεκτρονικών Μηχανικών Τ.Ε

*Ηράκλειο, Ιανουάριος 2019*



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών (ΔΠΜΣ)

«Προηγμένα Συστήματα Παραγωγής, Αυτοματισμού και Ρομποτικής»

## ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

### Βελτιστοποίηση της Συνεργατικής Συμπεριφοράς Ομάδων Ρομποτικών Οχημάτων με τη βοήθεια Εξελικτικών Στρατηγικών

Μεταπτυχιακός Φοιτητής: Αθανάσιος Μαρτίνης

ΑΜ: ΜΤΗ31

Επιβλέπων Καθηγητής: Αναπληρωτής Καθηγητής Ε. Δοϊτσίδης

Παρουσιάστηκε δημόσια και εξετάστηκε στο  
Τμήμα.....στις...../...../.....

Εξεταστική Επιτροπή

Δοϊτσίδης Ελευθέριος, Αναπληρωτής Καθηγητής ΤΕΙ Κρήτης .....

Φασουλάς Ιωάννης, Επίκουρος Καθηγητής ΤΕΙ Κρήτης .....

Παπαδουράκης Γεώργιος, Καθηγητής ΤΕΙ Κρήτης .....



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να ευχαριστήσω την οικογένεια μου για την ανοχή και την υποστήριξη που μου πρόσφεραν ώστε να καταφέρω να ολοκληρώσω το μεταπτυχιακό πρόγραμμα. Τους καθηγητές του μεταπτυχιακού προγράμματος για την συνεργασία που είχαμε και ιδιαιτέρως τον επιβλέποντα καθηγητή κ. Δοϊτσίδα Ελευθέριο για την άριστη συνεργασία, την καθοδήγησή του και την υπομονή του για την εκπόνησης της παρούσας διπλωματικής εργασίας.





## ΠΕΡΙΛΗΨΗ

Ο σχεδιασμός ελεγκτών ρομποτικών οχημάτων, είναι ένα αντικείμενο εξαιρετικού ενδιαφέροντος καθώς οι αυτόνομες συσκευές παίζουν ολοένα και σημαντικότερο ρόλο στην καθημερινότητα μας. Ήδη μια πληθώρα εταιριών έχει παρουσιάσει αυτόνομα ρομποτικά συστήματα ικανά να κινούνται σε άγνωστο περιβάλλον.

Στόχος της παρούσας εργασίας είναι να μελετηθεί η δυνατότητα σχεδίασης, με βέλτιστο τρόπο, ελεγκτών σχεδιασμένων με την βοήθεια τεχνικών ευφυούς ελέγχου για την πλοήγηση αυτόνομων ρομποτικών συσκευών. Συγκεκριμένα σχεδιάστηκαν ασαφής ελεγκτές για τη πλοήγηση ρομποτικών οχημάτων που κινούνταν σε άγνωστο περιβάλλον. Οι ελεγκτές βελτιστοποιήθηκαν με τη βοήθεια γεννητικών αλγορίθμων, μιας διαδικασίας βελτιστοποίησης εμπνευσμένη από τη διαδικασία της φυσικής επιλογής, που βασίζεται στην αξιολόγηση των αποτελεσμάτων με τη βοήθεια μιας συνάρτησης καταλληλότητας (fitness function).

Στόχος ήταν να μελετηθεί η επίδραση διαφορετικών τύπων συναρτήσεων καταλληλότητας στην εξελικτική διαδικασία, καθώς και στη συμπεριφορά των οχημάτων. Προκειμένου να αξιολογηθεί η προτεινόμενη προσέγγιση προσομοιώθηκε η συμπεριφορά μιας ομάδας οχημάτων αποτελούμενη από 3 μέλη που είχαν κοινό στόχο.

Μελετήθηκαν 3 διαφορετικοί τύποι συναρτήσεων καταλληλότητας σύμφωνα με τη προσέγγιση που έχει προταθεί στην [8], με κύριο χαρακτηριστικό τη πρότερη γνώση που ενσωματώνεται στην εξελικτική διαδικασία μέσω της συνάρτησης προσαρμογής. Όλα τα πειράματα πραγματοποιήθηκαν σε περιβάλλον προσομοίωσης με χρήση του προσομοιωτή ρομποτικών οχημάτων VREP. Η εξελικτική διαδικασία υλοποιήθηκε με χρήση του λογισμικού Matlab, που διασυνδέθηκε με το περιβάλλον προσομοίωσης VREP με μια προσέγγιση client – server.

Τα αποτελέσματα που προέκυψαν αξιολογήθηκαν συγκριτικά και εξάχθησαν χρήσιμα συμπεράσματα.

## ABSTRACT

The core component of autonomous vehicles is the actual controller of the vehicle itself, which is responsible for the operation in an unknown environment. Currently there are several approaches on the actual type of the controller used and on the way that it is formulated.

The scope of this thesis, was to study how we can derive working, robust controllers using computational intelligence techniques, so that we can autonomously navigate a robotic vehicle in an unknown environment with static and dynamic obstacles. Our approach was based on fuzzy logic, a technique which allows the modelling of human knowledge in a meaningful way, in order to be used on a mobile robot. The aforementioned fuzzy logic based controllers were optimized using genetic algorithms, a nature inspired methodology, which is based on fitness function which is metric that evaluates the overall performance of the proposed controller.

A key factor to the optimization procedure is the a priori knowledge, that the fitness function provides, to the procedure. Based on the classification proposed in [8] we selected three different type of functions and used as a test case a team of three robots with a common goal. We performed detailed simulations and compared the performance of the different approaches.

Based on the results we provide some insight about the influence of the different functions to the overall performance of the team and to the actual behavior of the robots.

# Περιεχόμενα

## ΚΕΦΑΛΑΙΟ 1

|               |    |
|---------------|----|
| ΕΙΣΑΓΩΓΗ..... | 15 |
|---------------|----|

## ΚΕΦΑΛΑΙΟ 2

|                               |    |
|-------------------------------|----|
| ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ..... | 18 |
|-------------------------------|----|

|   |    |
|---|----|
| 2.1 ΕΙΣΑΓΩΓΗ .....  | 18 |
| 2.2 ΠΡΟΗΓΟΥΜΕΝΕΣ ΜΕΛΕΤΕΣ.....   | 18 |
| 2.3 ΑΣΑΦΗΣ ΈΛΕΓΧΟΣ.....   | 21 |
| 2.3.1 Εισαγωγή.....   | 21 |
| 2.3.2 Ευφυής έλεγχος.....   | 22 |
| 2.3.3 Ασαφής έλεγχος.....   | 23 |
| 2.3.4 Ασαφή σύνολα.....   | 25 |
| 2.3.4.1 Ορισμοί.....  | 25 |
| 2.3.4.2 Τύποι συναρτήσεων ασαφών συνόλων.....   | 27 |
| 2.3.4.3 Πράξεις ασαφών συνόλων.....   | 29 |
| 2.3.4.4 Σχέσεις μεταξύ ασαφών συνόλων.....  | 30 |
| 2.3.5 Ασαφείς Αριθμοί.....  | 30 |
| 2.3.6 Συνεπαγωγές.....  | 31 |
| 2.3.7 Προσεγγιστικός συλλογισμός.....   | 32 |
| 2.3.8 Σχεδιασμός Συστήματος Ασαφούς Ελέγχου.....  | 33 |
| 2.3.8.1 Ζητήματα μοντελοποίησης και στόχοι απόδοσης.....                                      | 35 |
| 2.3.8.2 Σχεδιασμός Ασαφούς Ελεγκτή.....   | 36 |
| 2.3.8.3 Αξιολόγηση επιδόσεων ασαφή Ελεγκτή.....   | 36 |
| 2.3.8.4 Ασαφοποίηση εισόδων.....  | 37 |
| 2.3.8.5 Μηχανισμός συμπερασμού.....   | 37 |
| 2.3.8.6 Αποασαφοποίηση εξόδων.....  | 38 |
| 2.3.8.7 Γνωστοί μηχανισμοί ασαφών ελεγκτών.....   | 38 |
| 2.4 ΈΛΕΓΧΟΣ ΕΝΤΡΟΧΟΥ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ ΜΕ ΑΣΑΦΗ ΛΟΓΙΚΗ.....                                 | 40 |
| 2.5 ΕΞΕΛΙΚΤΙΚΟΙ ΥΠΟΛΟΓΙΣΜΟΙ (EVOLUTIONARY COMPUTATION).....                                   | 40 |
| 2.5.1 Γενετικοί αλγόριθμοι.....   | 42 |
| 2.5.2 Γενετικός Προγραμματισμός.....  | 44 |
| 2.5.3 Εξελικτική Στρατηγική.....  | 45 |
| 2.5.4 Εξελικτικός προγραμματισμός.....  | 45 |
| 2.6 ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ.....   | 45 |
| 2.6.1 Ιστορική αναδρομή.....  | 45 |
| 2.6.2 Βασικές έννοιες.....  | 47 |
| 2.6.2.1 Γονίδια και χρωμοσώματα.....  | 47 |
| 2.6.2.2 Γονείς.....   | 47 |
| 2.6.2.3 Πληθυσμοί και γενιές.....   | 49 |
| 2.6.2.4 Παιδιά.....   | 49 |
| 2.6.2.5 Συνάρτηση Κόστους ή Καταλληλότητας.....   | 50 |
| 2.6.2.6 Αποτελεσματικότητα και Αποδοτικότητα [65].....  | 51 |
| 2.6.3 Τεχνικές Αναπαραγωγής.....  | 52 |
| 2.6.3.1 Διασταύρωση.....  | 52 |
| 2.6.3.2 Μετάλλαξη.....  | 54 |
| 2.6.4 Γενετική επιλογή.....   | 55 |
| 2.6.4.1 Μέθοδοι επιλογής.....   | 56 |
| 2.6.5 Πεδίο Καταλληλότητας.....   | 59 |
| 2.7 Η ΣΥΝΑΡΤΗΣΗ ΚΑΤΑΛΛΗΛΟΤΗΤΑΣ.....   | 62 |
| 2.7.1 Κατηγορίες Συναρτήσεων καταλληλότητας.....  | 63 |
| 2.7.1.1 Συνάρτηση Καταλληλότητας Δεδομένων Εκπαίδευσης (Training data fitness functions)..... | 64 |
| 2.7.1.2 Συνάρτηση Καταλληλότητας Συμπεριφοράς (Behavioral fitness functions).....             | 64 |

|  |    |
|--|----|
| 2.7.1.3 Συνάρτηση Καταλληλότητας Αυξανομένης Λειτουργικότητας (Functional incremental fitness functions)           | 65 |
| 2.7.1.4 Προσαρμοσμένη Συνάρτηση Καταλληλότητας (Tailored fitness functions)  | 65 |
| 2.7.1.5 Συνάρτηση καταλληλότητας αυξανομένης δυσκολίας περιβάλλοντος (Environmental incremental fitness Functions) | 66 |
| 2.7.1.6 Ανταγωνιστική ή Συνεργατική Συνάρτηση Καταλληλότητας (Competitive and co-competitive selection)            | 66 |
| 2.7.1.7 Αθροιστική ή Ολοκληρωτική Συνάρτηση Καταλληλότητας (Aggregate fitness functions)                           | 67 |
| 2.7.2 Αναλυτικά παραδείγματα συναρτήσεων καταλληλότητας [8]  | 68 |
| 2.7.2.1 Εφαρμοσμένες συναρτήσεις καταλληλότητας συμπεριφοράς   | 68 |
| 2.7.2.2 Εφαρμοσμένες Προσαρμοσμένες συναρτήσεις καταλληλότητας   | 72 |
| 2.7.2.3 Εφαρμοσμένες Αθροιστικές ή Συγκεντρωτικές Συναρτήσεις Καταλληλότητας (Aggregate fitness functions)         | 76 |
| 2.8 ΑΣΚΗΣΕΙΣ Η ΑΠΟΣΤΟΛΕΣ ΡΟΜΠΟΤ ΣΤΗΝ ΕΞΕΛΙΚΤΙΚΗ ΡΟΜΠΟΤΙΚΗ  | 78 |

### ΚΕΦΑΛΑΙΟ 3

|  |           |
|--|-----------|
| <b>ΈΛΕΓΧΟΣ ΕΝΤΡΟΧΟΥ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ ΜΕ ΧΡΗΣΗ ΑΣΑΦΗΣ ΛΟΓΙΚΗΣ</b>                          | <b>82</b> |
| 3.1 ΕΙΣΑΓΩΓΗ   | 82        |
| 3.2 ΚΙΝΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ ΔΙΑΦΟΡΙΚΗΣ ΚΙΝΗΣΗΣ [62]                           | 82        |
| 3.3 ΑΣΑΦΗΣ ΕΛΕΓΧΟΣ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ ΔΙΑΦΟΡΙΚΗΣ ΚΙΝΗΣΗΣ                                    | 85        |
| 3.3.1 Προσομοίωση της λειτουργίας του οχήματος διαφορικής οδήγησης με χρήση MatLab και V-rep | 86        |
| 3.3.2 V-rep, Virtual robot experimentation platform  | 87        |
| 3.3.2.1 Vrep 's Μεθοδολογία Ελέγχου  | 89        |
| 3.3.2.2 Η Αρχιτεκτονική ελέγχου στο V-REP  | 89        |
| 3.3.2.3 V-REP Περιβάλλον Προγραμματισμού - Τρόπος λειτουργίας                                | 90        |
| 3.3.2.4 Απομακρυσμένο Περιβάλλον Προγραμματισμού (Remote API)                                | 91        |
| 3.3.3 Επικοινωνία Vrep - Matlab μέσω απομακρυσμένου API (Remote API modus operandi)          | 92        |
| 3.3.4 Το Pioneer στο V-rep   | 93        |
| 3.3.5 Ο Ασαφής Ελεγκτής  | 96        |
| 3.3.6 Προσομοίωση Ομάδας Ρομποτικών Οχημάτων στο Vrep  | 100       |
| 3.4.3.1 Σκηνή ορισμού τιμών ταχύτητας μοτέρ  | 101       |
| 3.4 ΕΠΙΒΕΒΑΙΩΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΑΣΑΦΗ ΕΛΕΓΚΤΗ - ΣΕΝΑΡΙΟΥ   | 102       |
| 3.4.1 Αποτελέσματα προσομοίωσης 1 <sup>ου</sup> Οχήματος                                     | 106       |
| 3.4.2 Αποτελέσματα προσομοίωσης 2 <sup>ου</sup> Οχήματος                                     | 108       |
| 3.4.3 Αποτελέσματα προσομοίωσης 3 <sup>ου</sup> Οχήματος                                     | 110       |

### ΚΕΦΑΛΑΙΟ 4

|   |            |
|---|------------|
| <b>ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΑΣΑΦΟΥΣ ΕΛΕΓΚΤΗ ΡΟΜΠΟΤΙΚΩΝ ΟΧΗΜΑΤΩΝ ΜΕ ΤΗ ΧΡΗΣΗ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ</b>                | <b>112</b> |
| 4.1 ΕΙΣΑΓΩΓΗ  | 112        |
| 4.2 ΔΕΔΟΜΕΝΑ ΓΕΝΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ   | 113        |
| 4.3 ΔΗΜΙΟΥΡΓΙΑ ΧΡΩΜΟΣΩΜΑΤΟΣ ΑΠΟ ΕΛΕΓΚΤΗ ΑΣΑΦΟΥΣ ΛΟΓΙΚΗΣ   | 114        |
| 4.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΛΙΚΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ : ΣΥΝΑΡΤΗΣΗ ΚΑΤΑΛΛΗΛΟΤΗΤΑΣ ΣΥΜΠΕΡΙΦΟΡΑΣ (BEHAVIORAL)             | 116        |
| Ελεγκτής : καλύτερη τιμή συνάρτηση καταλληλότητας συμπεριφοράς  | 118        |
| Αποτελέσματα προσομοίωσης 1 <sup>ου</sup> Οχήματος  | 122        |
| Αποτελέσματα προσομοίωσης 2 <sup>ου</sup> Οχήματος  | 124        |
| Αποτελέσματα προσομοίωσης 3 <sup>ου</sup> Οχήματος  | 126        |
| Ελεγκτή : Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς          | 128        |
| Αποτελέσματα προσομοίωσης 1 <sup>ου</sup> Οχήματος  | 132        |
| Αποτελέσματα προσομοίωσης 2 <sup>ου</sup> Οχήματος  | 134        |
| Αποτελέσματα προσομοίωσης 3 <sup>ου</sup> Οχήματος  | 136        |
| 4.5 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΛΙΚΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ : ΑΘΡΟΙΣΤΙΚΗ Η ΟΛΟΚΛΗΡΩΤΙΚΗ ΣΥΝΑΡΤΗΣΗ ΚΑΤΑΛΛΗΛΟΤΗΤΑΣ (AGGREGATE) | 138        |
| Ελεγκτή : με την καλύτερη τιμή στην ολοκληρωτική συνάρτηση καταλληλότητας                                 | 139        |

|   |     |
|---|-----|
| Αποτελέσματα προσομοίωσης 1 <sup>ου</sup> Οχήματος .....  | 143 |
| Αποτελέσματα προσομοίωσης 2 <sup>ου</sup> Οχήματος .....  | 145 |
| Αποτελέσματα προσομοίωσης 3 <sup>ου</sup> Οχήματος .....  | 147 |
| <i>Ελεγκτή : Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή ολοκληρωτικής συνάρτησης καταλληλότητας</i> .....  | 149 |
| Αποτελέσματα προσομοίωσης 1 <sup>ου</sup> Οχήματος .....  | 153 |
| Αποτελέσματα προσομοίωσης 2 <sup>ου</sup> Οχήματος .....  | 155 |
| Αποτελέσματα προσομοίωσης 3 <sup>ου</sup> Οχήματος .....  | 157 |
| <b>4.6 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΛΙΚΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ : ΠΡΟΣΑΡΜΟΣΜΕΝΗ ΣΥΝΑΡΤΗΣΗ ΚΑΤΑΛΛΗΛΟΤΗΤΑΣ (TAILORED)</b> .....       | 159 |
| <i>Ελεγκτή : με την καλύτερη τιμή στην προσαρμοσμένη συνάρτησης καταλληλότητας</i> .....                        | 160 |
| Αποτελέσματα προσομοίωσης 1 <sup>ου</sup> Οχήματος .....  | 164 |
| Αποτελέσματα προσομοίωσης 2 <sup>ου</sup> Οχήματος .....  | 166 |
| Αποτελέσματα προσομοίωσης 3 <sup>ου</sup> Οχήματος .....  | 168 |
| <i>Ελεγκτή : Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή προσαρμοσμένης συνάρτησης καταλληλότητας</i> ..... | 170 |
| Αποτελέσματα προσομοίωσης 1 <sup>ου</sup> Οχήματος .....  | 174 |
| Αποτελέσματα προσομοίωσης 2 <sup>ου</sup> Οχήματος .....  | 176 |
| Αποτελέσματα προσομοίωσης 2 <sup>ου</sup> Οχήματος .....  | 178 |

## ΚΕΦΑΛΑΙΟ 5

|  |     |
|--|-----|
| <b>ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΡΓΑΣΙΕΣ</b> ..... | 180 |
|--|-----|

|                           |     |
|---------------------------|-----|
| <b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b> ..... | 181 |
|---------------------------|-----|

|                        |     |
|------------------------|-----|
| <b>ΠΑΡΑΡΤΗΜΑ</b> ..... | 186 |
|------------------------|-----|

|   |     |
|---|-----|
| A. ΔΟΜΗ ΑΠΟΘΗΚΕΥΣΗΣ ΔΕΔΟΜΕΝΩΝ ΠΡΟΣΟΜΟΙΩΣΗΣ.....                         | 186 |
| B. ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ROBOT SIMULATION .....                                | 187 |
| Γ. ΔΟΜΗ ΤΩΝ ΡΟΥΤΙΝΩΝ ROBOT SIMULATION .....                             | 188 |
| <i>RGSMoveToPoint_v1</i> .....  | 189 |
| CreateObjectHandles_v1 .....  | 193 |
| StreamSimValues_v2 .....  | 196 |
| SetVehicleSpeed_v1 .....  | 200 |
| calculateStep_v1 .....  | 201 |
| readsensorData_v6 .....   | 202 |
| distcalc .....  | 207 |
| normaliseInCircle_v2 .....  | 207 |
| randompoint .....   | 207 |
| readvehicleSpeed_v3 .....   | 208 |
| storevehicleData_v1 .....   | 210 |
| B. ΡΟΥΤΙΝΕΣ ΕΞΕΛΙΚΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ .....                               | 213 |
| <i>LocoFuzzyOptim_v2.m</i> .....  | 213 |
| <i>Fit_LM_SU_v_1</i> .....  | 214 |
| <i>Fit_LM_P_v_2</i> .....   | 216 |
| <i>Fit_LM_PSU_v_4</i> .....   | 217 |
| <i>LocoMoveoutputfcn_v4</i> .....                                       | 219 |
| E. ΤΥΠΟΙ ΛΕΙΤΟΥΡΓΙΑΣ REMOTE API [64] .....                              | 220 |
| Αποκλεισμός κλήσεων λειτουργίας – (Blocking function calls).....        | 220 |
| Κλήσεις λειτουργίας μη αποκλεισμού - (Non-blocking function calls)..... | 221 |
| Ροή δεδομένων - (Data streaming) .....                                  | 222 |
| Σύγχρονη λειτουργία - (Synchronous operation).....                      | 222 |
| Πρόσθετες λεπτομέρειες.....   | 224 |





# ΚΕΦΑΛΑΙΟ 1

## Εισαγωγή

Η πλοήγηση και ο έλεγχος ρομποτικών οχημάτων είναι ένα θέμα που τα τελευταία χρόνια προσελκύει όλο και μεγαλύτερο ενδιαφέρον. Καθοριστικό ρόλο για αυτό έπαιξε η τεχνολογική εξέλιξη που επέτρεψε τη χρήση ρομποτικών συστημάτων για σκοπούς που μέχρι πριν από λίγο καιρό, λόγω των περιορισμένων τους δυνατοτήτων, δεν ήταν δυνατόν να χρησιμοποιηθούν. Σήμερα όλο και συχνότερα εμφανίζονται ρομποτικά συστήματα με δυνατότητες παραπλήσιες αυτών που έχουν οι άνθρωποι-χειριστές, όπως για παράδειγμα τα αυτόνομα αυτοκίνητα που παρουσιάζονται από διάφορες εταιρίες.

Στην ρομποτική χρησιμοποιείται η γνώση από διάφορους κλάδους, όπως η μηχανική, η ηλεκτρονική και η μηχανική υπολογιστών, με σκοπό την μετακίνηση ενός τροχήλατου ρομπότ σε ένα καθορισμένο περιβάλλον με κάποιο βαθμό αυτονομίας. Μία από τις σημαντικότερες προκλήσεις της αυτόνομης πλοήγησης για τα αυτοκινούμενα ρομπότ είναι η ανίχνευση και η αποφυγή εμποδίων. Επειδή οι πληροφορίες αυτές συνήθως δεν είναι γνωστές με ακρίβεια, η ασαφής λογική είναι μια κατάλληλη τεχνική για τον χειρισμό αυτών των δεδομένων.

Απώτερος σκοπός είναι να αναπτυχθούν συστήματα με υψηλή προσαρμοστικότητα σε μεταβαλλόμενες συνθήκες και με όσο το δυνατόν μικρότερη εξάρτηση από τον αρχικό σχεδιασμό. Μια προσέγγιση για την βελτιστοποίηση του ελεγκτή είναι οι γενετικοί αλγόριθμοι που χρησιμοποιούν την φυσική επιλογή ως μια μηχανή αναζήτησης. Ενεργούν πάνω σε έναν πληθυσμό ατόμων ή χρωμοσωμάτων. Εντός αυτού του πληθυσμού, τα χρωμοσώματα είναι οι πιθανές υποψήφιες λύσεις για το πρόβλημα που χρειάζεται να επιλυθεί. Τα χρωμοσώματα αποτελούνται από διάφορες μορφές όπως bits, αριθμούς ή ακολουθίες παραμέτρων, ανάλογα με το πρόβλημα.

Για την πραγματοποίηση της εργασίας προτιμήθηκε να γίνει χρήση λογισμικών Η/Υ και η διαδικασία να είναι μια προσομοίωση σε αυτόν. Η δημιουργία ενός πλήρους εικονικού μοντέλου ρομπότ ή συστήματος, με την προσομοίωση εξαρτημάτων και των προγραμμάτων ελέγχου μπορεί να επηρεάσει σημαντικά τη γενική απόδοση του έργου. Ανάλογα με το επίπεδο λεπτομέρειας και ακρίβειας του περιβάλλοντος προσομοίωσης υπάρχουν διάφορες περιοχές που μπορούν να αναλυθούν, οι οποίες επηρεάζουν τον κύκλο ζωής ανάπτυξης σε κάποιο βαθμό καθώς και το κόστος.

Τα οφέλη της προσομοίωσης είναι:

- Μείωση του κόστους που απορρέει από την παραγωγή ρομπότ.
- Έλεγχος του πηγαίου κώδικα που ελέγχει έναν συγκεκριμένο πόρο ή ένα μίγμα από πόρους.
- Προσομοίωση διάφορων εναλλακτικών λύσεων χωρίς τη συμμετοχή πραγματικού κόστους.
- Το ρομπότ ή συστήματα αυτού μπορούν να ελεγχθούν πριν από την εφαρμογή.



- Η προσομοίωση μπορεί να γίνει σε στάδια, το οποίο είναι ευεργετικό για πολύπλοκα έργα.
- Επίδειξη συστήματος για να προσδιοριστεί αν είναι βιώσιμο ή όχι?
- Συμβατότητα με ένα ευρύ φάσμα των γλωσσών προγραμματισμού.
- Μικρότερους χρόνους παράδοσης.

Βέβαια υπάρχουν και τα μειονεκτήματα της προσομοίωσης:

- Μια εφαρμογή μπορεί να προσομοιώσει ακριβώς ότι είναι προγραμματισμένο να προσομοιωθεί. Δεν θα προσομοιωθούν εσωτερικοί ή εξωτερικοί παράγοντες, οι οποίοι αγνοούνται στη φάση ανάπτυξης.
- Ένα ρομπότ μπορεί να αντιμετωπίσετε πολλά περισσότερα σενάρια στον πραγματικό κόσμο από ό, τι μπορεί να προσομοιωθεί.

Οι νέες εκδόσεις από τις πλατφόρμες λογισμικού προσομοίωσης προσφέρουν όλο και περισσότερα χαρακτηριστικά που κάνουν μια προσομοίωση ευκολότερη και επίσης πολύ κοντά στην πραγματικότητα. Τα περισσότερα εργαλεία προσομοίωσης είναι συμβατά με προγραμματισμού γλώσσες όπως η C/C++, Perl, Python, Java, LabVIEW, URBI ή MATLAB, ωστόσο σε γενικές γραμμές προσφέρουν ποικίλες δυνατότητων ανάλογα με τους τομείς που η κάθε εφαρμογή στοχεύει ή εστιάζει. Στην συγκεκριμένη παρουσίαση έχει γίνει χρήση των προγραμμάτων Matlab και V-rep.

Το Matlab χρησιμοποιήθηκε για :

- για την δημιουργία του ελεγκτή ασαφούς λογικής
- Έλεγχο του ρομπότ που προσομοιώνουμε (σε συνεργασία με πρόγραμμα προσομοίωσης ρομπότ).
- και την χρήση των γενετικών αλγόριθμων για την εξέλιξη του ελεγκτή με διάφορα σενάρια.

Ενώ το V-rep,

- Προσομοίωση του τροχήλατου Ρομπότ με την χρήση εξωτερικού προγράμματος ελέγχου μέσα από το Matlab.

Αναλυτικά, στο δεύτερο κεφάλαιο παρουσιάζονται μετά από βιβλιογραφική έρευνα οι εργασίες που είναι σχετικές με την παρούσα εργασία, παρουσιάζονται οι βασικές αρχές της ασαφούς λογικής καθώς και ο τρόπος που σχεδιάζονται οι ασαφείς ελεγκτές για την πλοήγηση εντρόχων ρομποτικών οχημάτων. Ακολουθεί η ανάλυση της διαδικασίας βελτιστοποίησης με χρήση γενετικών αλγόριθμων.

Στο τρίτο κεφάλαιο γίνεται η παρουσίαση του τρόπου κίνησης των διαφορικών ρομποτικών οχημάτων και παρουσιάζεται ο αρχικός ασαφής ελεγκτή που δημιουργήθηκε. Παρουσιάζεται το περιβάλλον προσομοίωσης και το έντροχο ρομποτικό όχημα που χρησιμοποιήθηκε για την εργασία. Επίσης γίνεται αναλυτική παρουσίαση του τρόπου που συνεργάζονται τα δύο προγράμματα Vrep και Matlab.

Στο τέταρτο κεφάλαιο παρουσιάζονται η διαδικασία βελτιστοποίησης του ασαφή ελεγκτή και τα αποτελέσματα από την χρήση των 3 διαφορετικών συναρτήσεων

καταλληλότητας. Αναλύονται τα αποτελέσματα και παρουσιάζονται οι καλύτεροι ελεγκτές καθώς και μια συγκριτική ανάλυση των αποτελεσμάτων.

Στο πέμπτο κεφάλαιο γίνεται μια σύνοψη της εργασίας και παρουσιάζονται σκέψεις για μελλοντική έρευνα στο συγκεκριμένο πεδίο.

Στο παράρτημα παρουσιάζονται αναλυτικά οι κώδικες που αναπτύχθηκαν στα πλαίσια της συγκεκριμένης εργασίας με χρήση του λογισμικού MATLAB

.

## ΚΕΦΑΛΑΙΟ 2

### ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

#### 2.1 Εισαγωγή

Στο κεφάλαιο παρουσιάζονται εργασίες που ασχολήθηκαν με τους ασαφείς ελεγκτές για έντροχα ρομποτικά οχήματα, το πρόγραμμα προσομοίωσης ρομπότ V-ger σε συνεργασία με το Matlab και την μελέτη της συνάρτησης καταλληλότητας στην εξελικτική διαδικασία των ασαφών ελεγκτών. την συνέχεια παρουσιάζονται οι βασικές αρχές της ασαφούς λογικής και πως χρησιμοποιείται στον έλεγχο συστημάτων και ειδικότερα στα έντροχα ρομποτικά οχήματα. Ακολούθως γίνεται μια γενική αναφορά στην εξελικτική διαδικασία, τα είδη βελτιστοποίησης και μια αναλυτικότερη παρουσίαση των Γενετικών Αλγόριθμων με την συνάρτηση καταλληλότητας.

#### 2.2 Προηγούμενες Μελέτες

Στην εργασία [1] γίνεται έλεγχος ρομποτικού οχήματος με την βοήθεια ασαφή ελεγκτή, ώστε να λειτουργεί σ' ένα περιβάλλον που περιλαμβάνει αντικείμενα και άλλα ρομποτικά οχήματα. Ο ελεγκτής ήταν 2 επιπέδων. Το 1<sup>ο</sup> επίπεδο αποτελούνταν από έναν εκλεκτή που οδηγούσε το όχημα και βοηθούσε στην αποφυγή των εμποδίων τα οποία δεν ήταν ρομποτικά οχήματα και το 2<sup>ο</sup> επίπεδο από ένα Petri Net μοντέλο που δημιουργεί κανόνες if – then και ήταν για την αποφυγή των ρομποτικών οχημάτων. Τα οχήματα είχαν την δυνατότητα της μεταξύ τους επικοινωνίας. Το όχημα που χρησιμοποιήθηκε ήταν ένα τρίτροχο με διαφορική κίνηση στους 2 τροχούς. Είχε αισθητήρες υπερήχων για την μέτρηση της απόστασης από το αντικείμενο και αισθητήρες υπέρυθρων για τον εντοπισμό του στόχου. Ο ασαφής ελεγκτής είχε είσοδο τα δεδομένα από τους αισθητήρων και έξοδο την ταχύτητα των μοτέρ. Οι λεκτικές μεταβλητές 'αργά', 'γρήγορα', 'μέτρια' καθορίζουν την ταχύτητα του μοτέρ. Οι λεκτικές μεταβλητές 'μακριά', 'μέτρια' και 'κοντά' καθορίζουν την απόσταση του στόχου. Οι λεκτικές μεταβλητές 'θετική', 'αρνητική' και 'μηδενική' καθορίζουν την διεύθυνση του στόχου. Οι συναρτήσεις συμμετοχής ήταν τραπέζια και τρίγωνα. Είχαν δημιουργηθεί 24 κανόνες. Οι 1-12 αφορούν την αντίδραση για την αποφυγή αντικειμένου, οι 13-18 αφορούν την αποφυγή εμποδίου και ακολουθήσει τοιχίου και οι 19-24 αφορούν την προσέγγιση στόχου. Προκειμένου να ταυτοποιηθεί η καλή λειτουργία της προτεινόμενης προσέγγισης έγινε προσομοίωση με χρήση του λογισμικού ROBNV, που είχε την δυνατότητα να προσομοιώσει 1000 ρομποτικά οχήματα. Μελετήθηκαν 5 διαφορετικά σενάρια. Συγκεκριμένα : (1) Ακολούθηση και αναζήτηση στόχου. (2) Αποφυγή αντικειμένου – Αναζήτηση στόχου από πολλαπλά ρομπότ. (3) Μετακίνηση, ελεύθερη από σύγκρουση σ ένα ασαφές περιβάλλον. (4) Αποφυγή αντικειμένων από μεγάλο αριθμό ρομπότ. (5) Διαφυγή από αδιέξοδο.

Τα αποτελέσματα έδειξαν ότι το σύστημα λειτουργεί και επιτυγχάνει αυτό για το οποίο σχεδιάστηκε. Όλοι οι κανόνες ήταν εμπειρικοί και αποτέλεσμα μελέτης του σχεδιαστή. Δεν έγινε καμία προσπάθεια βελτιστοποίησης μέσω κάποιας εξελικτικής διαδικασίας.

Η εργασία [2] ασχολείται με την ανάπτυξη μιας τεχνικής καθοδήγησης (on-line) για τον έλεγχο ενός αυτόνομου τροχοφόρου ρομπότ σε ένα άγνωστο (δυναμικό) περιβάλλον με την χρήση τεχνικών ασαφούς λογικής. Έγινε χρήση 2 διαφορετικών Ασαφών Ελεγκτών με διαφορετικές συναρτήσεις συμμετοχής. (1) Καθοδήγηση στο στόχο (TFLC) και (2) αποφυγή εμποδίων (OAFLC). Το όχημα ήταν το Scout-II, ένα όχημα με διαφορεική κίνηση και 3 τροχούς. Η οδήγηση γίνεται ασύρματα μέσω WiFi και το πρόγραμμα οδήγησης είναι σε γλώσσα προγραμματισμού C++. Το όχημα είναι εξοπλισμένο με σύστημα Dead-Reckoning για την εύρεση της θέσης και της διεύθυνση, χωρίς όμως μεγάλη ακρίβεια. Ήταν εξοπλισμένο με 3 αισθητήρες υπέρηχων. Ο εκλεκτής TFLC έχει 2 εισόδους (α) Απόσταση, (β) Γωνιακή απόκλιση στόχου – οχήματος και 2 εξόδους (α) Ταχύτητα δεξιού και (β) ταχύτητα αριστερού τροχού. Ο ελεγκτής OAFLC είχε 3 εισόδους την απόσταση μέσω του κάθε αισθητήρα υπέρηχων και 2 εξόδους (α) ταχύτητα δεξιού και (β) ταχύτητα αριστερού τροχού. Ο κάθε ελεγκτής επεμβαίνει ανάλογα την περίπτωση, ο TFLC ήταν υπεύθυνος για την κίνηση σε ελεύθερο περιβάλλον ενώ ο OAFLC ήταν υπεύθυνος για την αποφυγή αντικειμένου. Το σύστημα δοκιμάστηκε σε τρία σενάρια (α) Καθαρό περιβάλλον, (β) με στατικά εμπόδια και (γ) με δυναμικά εμπόδια.

Η εργασία [3] παρουσιάζει την βελτιστοποίηση συστήματος που οδηγεί ένα τροχήλατο ρομπότ. Το σύστημα ελέγχου αποτελείται από δύο FIS (Fuzzy Inference Systems - ασαφή ελεγκτή), ένα για τον έλεγχο της οδήγησης και ένα για τον έλεγχο της αντίδρασης και ένα WFIS (Weighted Fuzzy Inference System) για έλεγχο ενεργοποίησης του κατάλληλου FIS και στο ανάλογο ποσοστό. Το όχημα είναι διαφορετικής κίνησης με τον πίσω τροχό ελεύθερο απλά για στήριξη. Είναι εξοπλισμένο με αισθητήρια και κάμερες. Τα στοιχεία συλλέγονται από τα αισθητήρια και τις κάμερες και είναι οι εισοδοί ενώ η έξοδος είναι οι ταχύτητες των τροχών και είναι συνδυασμός τιμών και από τους 2 FIS. Το σύστημα βελτιστοποιείται μέσω γενετικού αλγόριθμου στον WFIS ελεγκτή. Ο γενετικός αλγόριθμος είναι HGA (Hierarchical Genetic Algorithm) που έχει χαρακτηριστικά για την επίλυση πολλαπλών αντικειμενικών προβλημάτων. Βελτιστοποίηση γίνεται στις (α) παραμέτρους των συναρτήσεων συμμετοχής, (β) το πλήθος των συναρτήσεων συμμετοχής και (γ) στον αριθμό των κανόνων. Η βελτιστοποίηση λαμβάνει χώρα σε 2 στάδια (α) την επιλογή και των πολλαπλασιασμό των χρωμοσωμάτων και (β) την αξιολόγηση του κάθε ελεγκτή με κριτήριά (α) την απόσταση που διανύθηκε, (β) τον χρόνο που καλύφθηκε η απόσταση και (γ) τον χρόνο ζωής της μπαταρίας. Η στρατηγική δεν μπορεί να εξασφαλίσει ότι θα έχουμε οριζόντια μετατόπιση χωρίς βρόχους επανάληψης. Γι' αυτό έγινε χρήση ενός νευρωνικό δίκτυο (NN) που χρησιμοποιείται για να ανιχνεύσει κυκλικές τροχιές που δεν έχουν την επιθυμητή συμπεριφορά μετακίνησης δίνοντάς τους χαμηλή τιμή ενεργοποίησης ενώ επιβραβεύει με υψηλότερη τιμή ενεργοποίησης τις υπόλοιπες. Το NN έχει δύο εισόδους και μία έξοδο και δύο κρυμμένα επίπεδα. Η συνάρτηση καταλληλότητας βγαίνει σε συνδυασμό από το ένα ασαφή ελεγκτή και το νευρωνικό δίκτυο. Ο Γενετικός αλγόριθμος έχει αναπτυχθεί σε C# ενώ οι προσομοιώσεις έγιναν με χρήση του λογισμικού Matlab που αλληλοεπιδρά με το λογισμικό σε C#. Στα αποτελέσματα αναφέρεται ότι έχει βελτιωθεί η συμπεριφορά του ρομπότ μέσω της βελτίωσης του WFIS.

Η εργασία [4] αναφέρεται στην καθοδήγηση ενός μηχανοκίνητου οχήματος βάση της συμπεριφοράς του με ασαφή ελεγκτή. Ο ελεγκτής είναι δομημένος έτσι που στην

ουσία να είναι σαν 2 ξεχωριστοί ελεγκτές που συνεργάζονται. Έχει δυο τμήματα (α)την αναζήτηση του στόχου που παίρνει δεδομένα από την απόσταση του στόχου και την γωνία απόκλισης και δίνει έξοδο την ταχύτητα των μοτέρ και (β)την αποφυγή εμποδίου που παίρνει δεδομένα από τα στοιχεία των αισθητήρων και δίνει έξοδο την ταχύτητα των μοτέρ. Η κάθε έξοδος παίρνει ένα βαθμό βαρύτητας μέσω μιας εξίσωσης. Ο συνδυασμός των τιμών είναι το τελικό αποτέλεσμα. Οι προσομοιώσεις έγιναν με χρήση του λογισμικού Matlab και ταυτοποίηση με την χρήση οχήματος Voyager 2. Δεν έγινε καμία προσπάθεια να γίνει βελτιστοποίηση του συστήματος μέσω κάποιας εξελεγκτικής διαδικασίας.

Η εργασία [5] μελετά ένα σύστημα που χρησιμοποιεί ένα ασαφή ελεγκτή για την αποφυγή της σύγκρουσης και παράλληλα να μπορεί να ακολουθεί μια συγκεκριμένη πορεία. Το όχημα είναι εφοδιασμένο με αισθητήρες απόστασης τύπου υπερήχων, κάμερα για αποφυγή σύγκρουσης, αισθητήρες εδάφους για την ακολούθηση του μονοπατιού και GPS για την εύρεση της θέσης του οχήματος. Ο ασαφής ελεγκτής είναι από το Matlab Fuzzy Logic. Ο σχεδιασμός είχε στόχο τον εντοπισμό των αντικειμένων και την αποφυγή αυτών με χρήση όλων των αισθητήρων. Τα δεδομένα των αισθητήρων ήταν οι είσοδοι και έξοδοι οι ταχύτητες των μοτέρ ενώ έγινε χρήση 24 κανόνων. Η προσομοίωσή έγινε στο λογισμικό Webots Pro σε μια πίστα με μια μαύρη γραμμή και σταθερά εμπόδια, το όχημα ήταν το E-ruck robot με διαφορεική κίνηση. Έγιναν 3 σενάρια (α)απλό περιβάλλον, ένα αντικείμενο και ένα ρομπότ, (β)πολλά αντικείμενα και δύο ρομπότ και (γ)πολλά αντικείμενα σε διάφορες θέσεις και ένα ρομπότ. Η χρήση των αισθητηρίων βελτίωσε την απόσταση που το όχημα απέφυγε τα εμπόδια και υπήρξε μείωση της διαδρομής που έκανε το ρομπότ. Δεν υπήρχε καμία προσπάθεια βελτιστοποίησης του ελεγκτή μέσω κάποιας διαδικασίας.

Στην εργασία [6] δημιουργήθηκε ένας ασαφής ελεγκτής (FLC) για την καθοδήγηση ενός αυτόνομου τροχήλατου οχήματος. Το όχημα είναι τρίτροχο με διαφορεική κίνηση. Στο σενάριο μελετήθηκαν 2 διαφορετικές εκδόσεις στους ελεγκτές. Η πρώτη προσέγγιση ήταν ένας ελεγκτής μόνο για καθοδήγηση (χωρίς την δυνατότητα αποφυγής εμποδίων) όπου τα δεδομένα της απόστασης και της γωνίας απόκλισης από στόχο ήταν οι είσοδοι στον ελεγκτή, ενώ έξοδος ήταν η ταχύτητα των τροχών. Σχηματίστηκαν 35 κανόνες συμπεριφοράς. Η δεύτερη προσέγγιση ήταν ένας ελεγκτής για καθοδήγηση και αποφυγή εμποδίων, όπου τα δεδομένα της απόστασης, της γωνίας απόκλισης από στόχο και τα δεδομένα από τα αισθητήρια ήταν οι είσοδοι στον ελεγκτή, ενώ έξοδος ήταν η ταχύτητα των τροχών. Σχηματίστηκαν 62 κανόνες συμπεριφοράς. Η προσομοίωση έγινε στο λογισμικό Matlab με την χρήση μαθηματικού μοντέλου του οχήματος.

Η εργασία [7] αφορά την μοντελοποίηση ενός ειδικού οχήματος το οποίο στηριζόμενο στην φιλοσοφία του εκκρεμούς μπορεί να αλλάξει σχήμα (μέγεθος) ώστε να μπορεί να χωράει σε διαφορετικές διατομές σωληνώσεων. Ο σκοπός του οχήματος αυτού είναι να επιβλέπει τις σωληνώσεις που μεταφέρουν πετρελαιοειδή. Το μοντέλο και οι υπολογισμοί έγιναν στο Matlab. Η προσομοίωση έγινε στο Vrep με έλεγχο του οχήματος μέσω του Matlab. Το αποτέλεσμα ήταν πολύ καλό και δείχνει ότι ο συνδυασμός των δύο αυτών προγραμμάτων μπορεί να βοηθήσει στην ανάλυση και μελέτη διαφόρων ρομποτικών συστημάτων με την άνεση και την σιγουριά που προσφέρει ο υπολογιστής.

Στην εργασία [8] των Nelson, Barlow και Doitsidis γίνεται ερευνά για τις συναρτήσεις καταλληλότητας που χρησιμοποιούνται στον τομέα της εξελικτικής Ρομποτικής (ER). Η εξελικτική ρομποτική είναι ένας τομέας έρευνας που εφαρμόζει την τεχνητή εξέλιξη ώστε να παράγει συστήματα ελέγχου για αυτόνομα ρομπότ. Κατά τη διάρκεια της εξέλιξης, τα ρομπότ επιχειρούν να εκτελέσουν μια συγκεκριμένη εργασία σε ένα δεδομένο περιβάλλον. Οι ελεγκτές με την καλύτερη εκτέλεση επιλέγονται, τροποποιούνται και πολλαπλασιάζονται για να εκτελέσουν ξανά την εργασία σε μια επαναληπτική διαδικασία που μιμείται ορισμένες πτυχές της φυσικής εξέλιξης. Το βασικό στοιχείο αυτής της διαδικασίας είναι η μέτρηση της ικανότητας των εξελισσόμενων ελεγκτών. Προκειμένου να επιτευχθεί αυτό, πρέπει να αντιμετωπιστούν τα θέματα που αφορούν την αξιολόγηση της καταλληλότητας. Η εργασία αυτή Το έγγραφο αυτό καταγράφει τις έρευνες ER και έχει επικεντρωθεί σε εργασίες που εμπλέκονται πραγματικά ρομπότ. Η έρευνα κατηγοριοποιεί της συναρτήσεις καταλληλότητας σύμφωνα με το βαθμό μιας εκ των προτέρων γνώσης που χρησιμοποιείται για να διαμορφώσει τις διάφορες συναρτήσεις καταλληλότητας που χρησιμοποιούνται κατά τη διάρκεια της εξέλιξης.

Στην κατηγοριοποίηση που προτείνει η εργασία [8] έχει στηριχθεί η τρέχουσα μελέτη. Γίνεται μια εκτενής αναφορά στα αποτελέσματά της και παρουσιάζονται οι επιμέρους κατηγορίες σε επόμενο κεφάλαιο.

Η εργασία [9] έχει συνοψίσει και κατηγοριοποιήσει όλες τις αποστολές που έχουν χρησιμοποιηθεί στην ER, που μελετά τη χρήση της εξελικτικής πληροφορικής για την παραγωγή ελεγκτών για τα ρομπότ. Έτσι αναλύονται ένα μεγάλο μέρος από τις εργασίες που έχουν παρουσιαστεί και ανάλογα την αποστολή που επέλεξαν οι μελετητές γίνεται μια κατηγοριοποίηση και ανάλυση αυτών. Επίσης ανάλογα την περίπτωση γίνεται και παρουσίαση των συναρτήσεων καταλληλότητας που χρησιμοποιήθηκε ανά περίπτωση.

## 2.3 Ασαφής Έλεγχος

### 2.3.1 Εισαγωγή

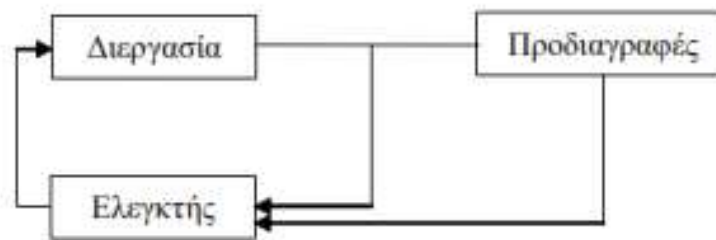
Η κλασική θεωρία συστημάτων αυτομάτου ελέγχου, συνέβαλε αποφασιστικά στην ανάπτυξη και υλοποίηση πολλών αγαθών την λειτουργία των οποίων τώρα θεωρούμε δεδομένη, όπως τα τηλέφωνα, ο αυτόματος πιλότος ενός αεροπλάνου κλπ. Παρά όμως την συνεχόμενη αυτή πρόοδο, το χάσμα μεταξύ θεωρίας και πράξης κυρίως στην βιομηχανία γίνεται όλο και μεγαλύτερο. Η μεγάλη ανάγκη της βιομηχανίας για ανάπτυξη μιας καινούριας θεωρίας ελέγχου που θα ανταποκρίνεται στις ανάγκες της, οδήγησε πολλούς επιστήμονες στην έρευνα νέων μη συμβατικών τεχνικών αυτομάτου ελέγχου, κάποιες από τις οποίες περιγράφονται από τον όρο "Ευφυής Έλεγχος".

Η εφαρμογή τεχνικών της συμβατικής θεωρίας ελέγχου για τον έλεγχο ενός συστήματος ή μιας διαδικασίας προϋποθέτει την ύπαρξη ενός πλήρους αναλυτικού μοντέλου του ελεγχόμενου συστήματος. Κάτι τέτοιο είναι συχνά αδύνατο λόγω της πολυπλοκότητας των βιομηχανικών διεργασιών ή και της αδυναμίας μετρήσεων. Αν υπάρχει ένα τέτοιο μοντέλο τότε η συνηθέστερος τύπος ελεγκτή που χρησιμοποιείται είναι αυτός των τριών όρων (PID) με υλοποίηση σε Programmable Logic Controllers (PLC).

Η θεωρία του μη συμβατικού ελέγχου, αντί να προσπαθεί να μοντελοποιήσει το ελεγχόμενο σύστημα, ψάχνει να βρει ένα σύνολο λεκτικών προτάσεων που να περιγράφουν τις αντιδράσεις ενός επιτυχημένου ανθρώπου χειριστή του συστήματος, τις οποίες και προσπαθεί να περιγράψει με διάφορες τεχνικές όπως την ασαφή λογική και τα νευρωνικά δίκτυα.

### 2.3.2 Ευφυής έλεγχος

Το πρόβλημα ελέγχου ενός συστήματος ή διεργασίας περιγράφεται από το ακόλουθο σχήμα.

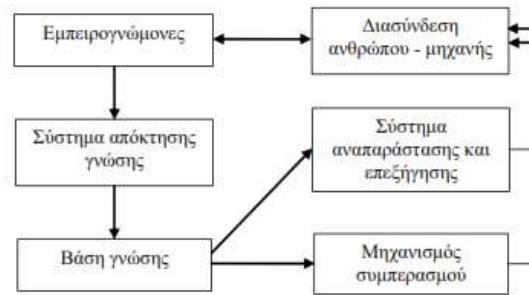


Εικόνα 1: Το πρόβλημα ελέγχου ενός συστήματος [10]

Στον συμβατικό έλεγχο η διεργασία και ο ελεγκτής θεωρούνται γνωστά και μάλιστα γραμμικά με την έννοια ότι υπάρχουν γραμμικά μαθηματικά μοντέλα διαφορικών εξισώσεων που τα περιγράφουν. Οι προδιαγραφές είναι ένα σύνολο κριτηρίων όπως ευστάθεια, ταχύτητα απόκρισης, υπερύψωσης κλπ που αν πληρούνται ο ελεγκτής θεωρείται επιτυχημένος.

Ο όρος "Ευφυής Έλεγχος" αντλεί τεχνικές από διάφορες επιστήμες όπως η νευρολογία, η ψυχολογία τα μαθηματικά κλπ. Ο στόχος ενός ευφυούς ελεγκτή είναι να λειτουργεί όπως ένας επιτυχημένος άνθρωπος ελεγκτής με τους ίδιους κανόνες χωρίς όμως τα μειονεκτήματά του. Το πλεονέκτημα των ανθρώπων σαν ελεγκτές μιας διεργασίας είναι ότι μπορούν να ανταπεξέλθουν και να πάρουν αποφάσεις κάτω από συνθήκες αβεβαιότητας και να αντιδράσουν άμεσα σε απρόβλεπτες καταστάσεις. Ένας καλά σχεδιασμένος ευφυής ελεγκτής, πρέπει να μπορεί να "μιμηθεί" τον καλύτερο άνθρωπο ελεγκτή της συγκεκριμένης διαδικασίας. Έτσι ένα πρώτο πρόβλημα που πρέπει να απαντηθεί από τον σχεδιαστή ενός ευφυούς ελεγκτή είναι η καταγραφή των κανόνων με βάση τους οποίους λειτουργεί ένας επιτυχημένος άνθρωπος ελεγκτής της διεργασίας.

Η εξόρυξη αυτής της γνώσης (data mining) γίνεται είτε με συνέντευξη του χειριστή είτε με τεχνικές pattern association, γενετικών αλγορίθμων κλπ. Δεύτερο βήμα είναι η αποθήκευση αυτών των κανόνων σε μια βάση γνώσης χρησιμοποιώντας είτε συμβολική μορφή (LISP, C++, κλπ) είτε αριθμητική μορφή (ασαφής λογική, νευρωνικά δίκτυα). Έπειτα ακολουθεί η επιλογή και υλοποίηση ενός μηχανισμού συμπερασμού ο οποίος παίρνοντας σαν είσοδο κάποιες μετρήσεις από την ελεγχόμενη διαδικασία και χρησιμοποιώντας την βάση γνώσης που έχει δημιουργηθεί, βγάζει κάποια έξοδο που ανατροφοδοτείται στην ελεγχόμενη διαδικασία. Τα παραπάνω βήματα φαίνονται στο εικόνα 2.



Εικόνα 2: Διαδικασία σχεδίασης ευφυής ελεγκτή [10]

Ο ευφυής έλεγχος συνήθως χρησιμοποιείται σε εφαρμογές μεγάλης κλίμακας και πολυπλοκότητας, μιας και τότε είναι πρακτικά αδύνατη η εφαρμογή συμβατικών τεχνικών.

Η βασικές αρχές πάνω στις οποίες σχεδιάζεται ένας ευφυής ελεγκτής είναι οι ακόλουθες.

**Ορθότητα:** Η ικανότητα εκτέλεσης των λειτουργικών απαιτήσεων του συστήματος με ασφάλεια.

**Ευρωστία:** Η ικανότητα του συστήματος να παραμένει λειτουργικό κάτω από μη αναμενόμενες συνθήκες.

**Επεκτασιμότητα:** Η δυνατότητα επέκτασης του υλικού και του λογισμικού χωρίς επανασχεδίαση του συστήματος από την αρχή.

Στα ασαφή συστήματα η αναπαράσταση της γνώσης γίνεται μέσω ασαφών συνόλων και ασαφούς Λογικής.

### 2.3.3 Ασαφής έλεγχος

Στα μέσα του 1960 ο Lotfi A. Zadeh του πανεπιστημίου Berkeley της Καλιφόρνια εισήγαγε την θεωρία των ασαφών συνόλων, η οποία λέει ότι συνήθως στον κόσμο που ζούμε τα αντικείμενα γύρω μας ανήκουν σε διάφορα σύνολα με διαφορετικούς βαθμούς συμμετοχής.

Πχ. η κλάση των "ψηλών ανθρώπων" δεν έχει αυστηρό κριτήριο συμμετοχής.

Ο ασαφής ορισμός κλάσεων παίζει πολύ μεγάλο ρόλο στην ανθρώπινη επικοινωνία. Το 1965 ο Zadeh θεμελίωσε πλήρως την θεωρία των ασαφών συνόλων και της ασαφούς λογικής ολοκληρώνοντας την δουλειά αρκετών άλλων μαθηματικών μέχρι τότε. Η θεωρία του Zadeh δέχθηκε μεγάλη αμφισβήτηση κυρίως στην Αμερική. Την δεκαετία του 1970 ο Ebrahim H. Mamdani, μηχανικός στο πανεπιστήμιο Queen Mary του Λονδίνου δοκίμασε για πρώτη φορά την ασαφή λογική για την ανάπτυξη ενός ελεγκτή ατμομηχανής. Η επιτυχία τους οδήγησε στην αναγνώριση της ασαφούς λογικής σαν ένα σημαντικό εργαλείο αυτομάτου ελέγχου κάτι που φαίνεται και από την πληθώρα επιστημονικών δημοσιεύσεων πάνω στο θέμα.

Η έννοια 'ασαφής λογική' έχει δυο διαφορετικές ερμηνείες.[11] Στη στενή ερμηνεία, ασαφής λογική είναι ένα λογικό σύστημα, το οποίο είναι μια επέκταση της πολυβάθμιας λογικής. Ωστόσο σε μια πιο διευρυμένη έννοια, η ασαφής λογική είναι σχεδόν συνώνυμη με την θεωρία των ασαφών συνόλων. Μια θεωρία η οποία



στηρίζεται σε ομάδες αντικειμένων τα όρια των οποίων δεν είναι αυστηρά προκαθορισμένα και οι συμμετοχή αυτών εξαρτάται από μια σχέση βαθμίδας. Με την οπτική αυτή, η ασαφής λογική, στην στενή έννοια, είναι εάν παρακλάδι της ασαφούς λογικής. Ακόμα όμως και στην ποιο στενή του έννοια, η ασαφής λογική διαφέρει τόσο στην έννοια αλλά και την ουσία από τα κλασικά πολυβάθμια λογικά συστήματα.

Ακόμα ένα σημαντικό στοιχείο της ασαφούς λογικής που παίζει κεντρικό ρόλο σε αυτή την εφαρμογή είναι ο κανόνας **Εάν - Τότε (if - then)**. Τα συστήματα που στηρίζονται σε κανόνες έχουν μια μακριά ιστορία στην τεχνητή νοημοσύνη. Όμως το πρόβλημα είναι ότι σε τέτοια συστήματα λείπει ο μηχανισμός που θα αντιμετωπίσει τα λογικά προηγούμενα και τις λογικές συνέπειες. Στην ασαφή λογική, ο μηχανισμός αυτός παρέχεται από το λογισμό των ασαφών κανόνων.

Ο λογισμός των ασαφών κανόνων εξυπηρετεί ως βάση και για αυτό θα μπορούσε να ονομαστεί η Ασαφής Εξάρτηση και Γλώσσα Εντολών (Fuzzy Dependency and Command Language, FDCL). Στις περισσότερες εφαρμογές της ασαφούς λογικής, στην πραγματικότητα η ασαφή λογική λύση είναι μια μετάφραση της ανθρώπινης αντίδρασης (λύση) σε FDCL.

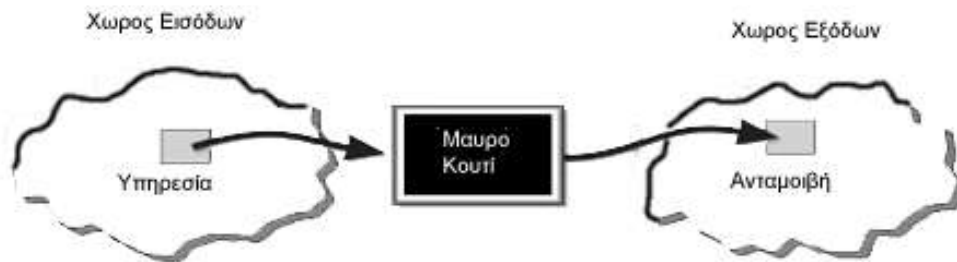


Εικόνα 3: Ανθρώπινη Αντίδραση [11]

Η Ασαφής λογική είναι ένας βολικός τρόπος για να περιγραφεί η μετάβαση από ένα σύνολο εισόδων σε ένα σύνολο εξόδων. Η δυνατότητα να περιγραφεί ο τρόπος μετάβασης από τα στοιχεία εισόδου στα στοιχεία εξόδου είναι το σημείο εκκίνησης για όλα. Εξετάστε τα ακόλουθα παραδείγματα: (α) Με πληροφορίες σχετικά με το πόσο καλή ήταν η υπηρεσία σε ένα εστιατόριο, ένα ασαφούς λογικής σύστημα μπορεί να σας πει πόση πρέπει να είναι η ανταμοιβή του σερβιτόρου. (β) Με τις προδιαγραφές σας για το ποσό ζεστό επιθυμείτε το νερό, η ασαφής λογική μπορεί να ρυθμίσει τη βαλβίδα της βρύσης. (γ) Με πληροφορίες σχετικά με το πόσο μακριά είναι το αντικείμενο της φωτογραφίας σας, ένα ασαφούς λογικής σύστημα μπορεί να εστιάσει το φακό για σας. (δ) Με πληροφορίες σχετικά με το πόσο γρήγορα το αυτοκίνητο πηγαίνει και πόσο σκληρά ο κινητήρας λειτουργεί, η ασαφής λογική του συστήματος μπορεί να αλλάζει ταχύτητες για σας.

Για να καθορίσει το κατάλληλο ποσό φιλοδωρήματος απαιτείται να οροθετηθεί η σχέση των διαφόρων εισόδων με την κατάλληλη έξοδο. Μεταξύ του συνόλου των εισόδων και του συνόλου των εξόδων, όπως παρατηρούμε στην εικόνα 4 μεσολαβεί ένα μαύρο κουτί που μπορεί να περιέχει οποιοδήποτε αριθμό πραγμάτων: ασαφή συστήματα, γραμμικά συστήματα, έμπειρα συστήματα, νευρωνικά δίκτυα, διαφορικές εξισώσεις, παρεμβολικούς πολυδιάστατους πίνακες αναζήτησης, για να αναφέρουμε

μόνο μερικές από τις πιθανές επιλογές. Είναι σαφές ότι ο κατάλογος θα μπορούσε να συνεχιστεί χωρίς τέλος.



Εικόνα 4: Σχέση Εισόδων - Εξόδων στον Ασαφή Έλεγχο [11]

Από τις δεκάδες τρόπους με τους οποίους το μαύρο κουτί μπορεί να δουλέψει, αποδεικνύεται ότι ο καλύτερος είναι η Ασαφή Λογική. Όπως ο Lotfi Zadeh, ο οποίος θεωρείται ότι είναι ο πατέρας της ασαφούς λογικής, παρατήρησε ότι:

"Σχεδόν σε κάθε περίπτωση μπορείτε να χτίσετε το ίδιο προϊόν χωρίς ασαφή λογική, αλλά η ασαφής λογική είναι ταχύτερη και φθηνότερη."

Η Ασαφής λογική είναι ευκολότερα κατανοητή. Οι μαθηματικές έννοιες πίσω από ασαφή συλλογιστική είναι πολύ απλές. Η Ασαφής λογική είναι μια πιο διαισθητική προσέγγιση, χωρίς την εκτεταμένη πολυπλοκότητα. Η Ασαφής λογική είναι πολύ ευέλικτη. Σε οποιοδήποτε δεδομένο σύστημα, είναι εύκολο να προσθέσουμε περισσότερες λειτουργίες χωρίς να απαιτείται να ξεκινήσουμε από την αρχή. Ασαφής λογική είναι ανεκτική σε ανακριβή στοιχεία.

Τα πάντα είναι ασαφή, αλλά περισσότερο και από αυτό, τα περισσότερα πράγματα είναι ανακριβής ακόμη και σε μια προσεκτική επιθεώρηση. Η Ασαφή συλλογιστική χτίζει αυτή την κατανόηση στη διαδικασία παρά στην εφαρμογή στο τέλος. Η Ασαφής λογική μπορεί να μοντελοποιήσει μη γραμμικές λειτουργίες αυθαίρετης πολυπλοκότητας. Μπορείτε να δημιουργήσετε ένα ασαφές σύστημα για να ταιριάζει με κάθε σύνολο δεδομένων εισόδου-εξόδου. Αυτή η διαδικασία είναι ιδιαίτερα εύκολη από προσαρμοστικές τεχνικές όπως Adaptive Neuro-Fuzzy Συστήματα (ANFIS). Η Ασαφής λογική μπορεί να δομηθεί πάνω στην εμπειρία των στελεχών της.

## 2.3.4 Ασαφή σύνολα

### 2.3.4.1 Ορισμοί

**Σύνολο** είναι οποιαδήποτε συλλογή - ομάδα ομοειδών πραγμάτων που έχουν ή ικανοποιούν μία συγκεκριμένη ιδιότητα. Τα μέλη της ομάδας αυτής καλούνται **στοιχεία** του συνόλου. Το πλήθος των στοιχείων ενός συνόλου καλείται πληθικός αριθμός του συνόλου (συμβολίζεται συνήθως με  $N$ ). Υπάρχουν **πεπερασμένα** και **άπειρα** σύνολα, ανάλογα με το αν ο πληθικός τους αριθμός είναι πεπερασμένος ή άπειρος.

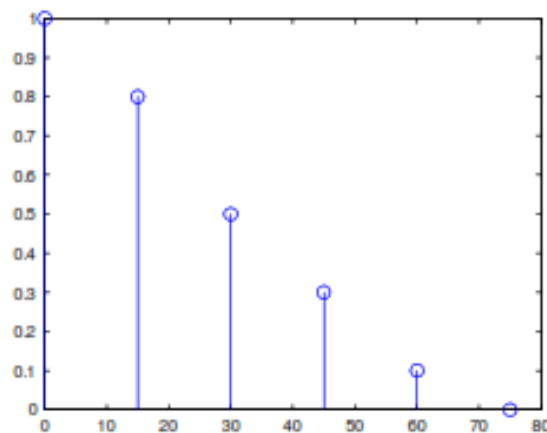
Έστω  $X$  ένα μη μηδενικό σύνολο. Ένα **ασαφές σύνολο  $A$**  του  $X$  χαρακτηρίζεται από την συνάρτηση συμμετοχής του  $\mu_A: X \rightarrow [0; 1]$  όπου  $\mu_A(x)$  είναι ο βαθμός συμμετοχής του στοιχείου  $x \in X$  στο **ασαφές σύνολο  $A$** . Το **ασαφές σύνολο  $A$**  χαρακτηρίζεται πλήρως από το σύνολο των ζευγαριών  $A = \{ (x; \mu_A(x)) \}$  όπου  $x \in X$ .

Αν  $\mathbf{X} = \{x_1; x_2; \dots; x_n\}$  ένα πεπερασμένο σύνολο και  $\mathbf{A}$  ένα ασαφές σύνολο του  $\mathbf{X}$  τότε χρησιμοποιούμε τον συμβολισμό  $\mathbf{A} = \mu_1/x_1 + \dots + \mu_n/x_n$  όπου  $\mu_i/x_i$  συμβολίζει ότι  $\mu_i$  είναι ο βαθμός συμμετοχής του  $x_i$  στο  $\mathbf{A}$  και το (+) συμβολίζει την ένωση. Όσο μεγαλύτερο βαθμό συμμετοχής έχει ένα στοιχείο τόσο περισσότερο "ανήκει" στο σύνολο.

Έστω ότι έχουμε το πεπερασμένο σύνολο  $\mathbf{X} = \{0; 15; 30; 45; 60; 75\}$  όπου τα στοιχεία του  $\mathbf{X}$  είναι οι ανθρώπινες ηλικίες σε χρόνια. Έστω επίσης ότι θέλουμε να ορίσουμε το σύνολο των νέων ανθρώπων πάνω σε ένα σύνολο  $\mathbf{X}$ . Η παραπάνω έννοια μπορεί να εκφραστεί από το ακόλουθο ασαφές σύνολο  $\mathbf{A}$ :

$$A = 1/0 + 0,8/15 + 0,5/30 + 0,3/45 + 0,1/60 + 0/75$$

και γραφικά από το ακόλουθο:

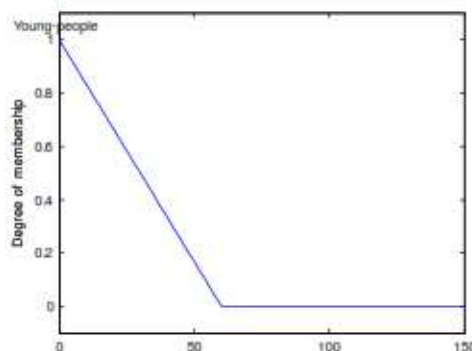


Εικόνα 5 :Γραφική παράσταση ασαφούς συνόλου A πάνω σε ένα σύνολο X [10].

Αν τώρα ορίσουμε το  $\mathbf{X}$  να είναι το σύνολο αριθμών από 0 έως 150. Το ασαφές σύνολο ορίζεται από την παρακάτω σχέση:

$$\mu_A(x) = \begin{cases} \frac{-1}{60}x + 1, & 0 \leq x \leq 60 \\ 0, & 60 \leq x \leq 150 \end{cases}$$

Και γραφικά θα παρουσιάζεται :



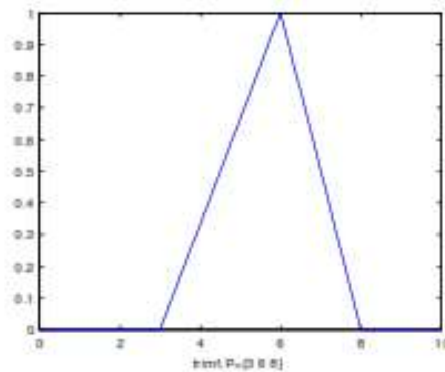
Εικόνα 6 : Ασαφές Σύνολο αριθμών 0 έως 150 [10]

Όταν το σύνολο  $\mathbf{X}$  είναι συνεχές τότε και η συνάρτηση συμμετοχής που αντιστοιχεί σε ένα ασαφές σύνολο  $\mathbf{A}$  είναι και αυτή με τη σειρά της συνεχής.

### 2.3.4.2 Τύποι συναρτήσεων ασαφών συνόλων

**Τριγωνική.** Έχει σαν παραμέτρους τρεις πραγματικούς αριθμούς  $a$ ;  $b$ ;  $c$ .

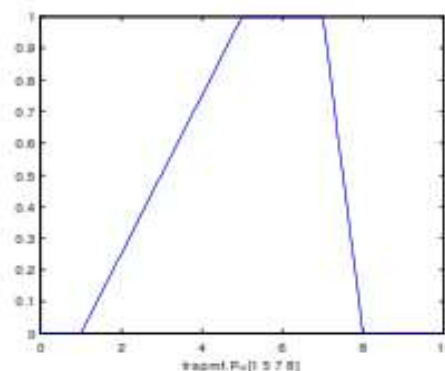
$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & x \geq c \end{cases}$$



Εικόνα 7 : Τριγωνική συνάρτηση συμμετοχής [10]

**Τραπεζοειδής.** Έχει τέσσερις πραγματικούς αριθμούς σαν παραμέτρους  $a$ ;  $b$ ;  $c$ ;  $d$ :

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x \geq d \end{cases}$$

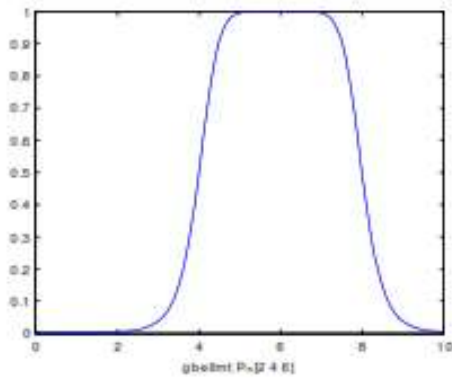


Εικόνα 8 : Τραπεζοειδής συνάρτηση συμμετοχής [10]

**Καμπανοειδής.** Έχει σαν παραμέτρους τρεις πραγματικούς αριθμούς  $a$ ;  $b$ ;  $c$ .

Η παράμετρος  $c$  δείχνει το κέντρο της καμπύλης, η  $b$  το σημείο που έχει σαν τιμή 0:5 και η  $a$  το σημείο που από 0 γίνεται αυστηρά θετική.

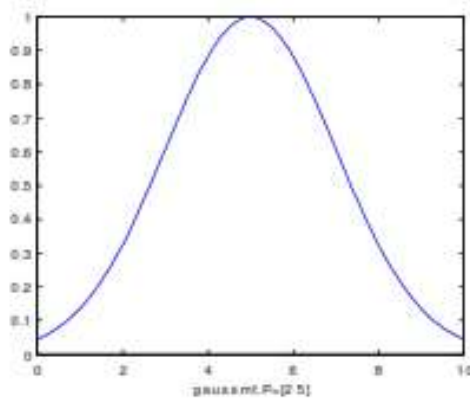
$$\mu_A(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}}$$



Εικόνα 9 : Καμπανοειδής συνάρτηση συμμετοχής [10]

**Συμμετρική συνάρτηση του Gauss.** Έχει σαν παραμέτρους δύο πραγματικούς αριθμούς  $a$  και  $c$ . Η  $c$  δείχνει το κέντρο της καμπύλης.

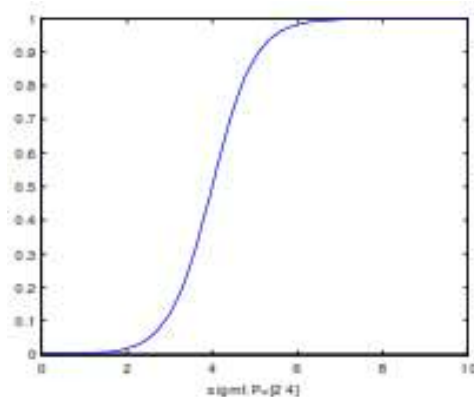
$$\mu_A(x) = e^{-\frac{(x-c)^2}{2a^2}}$$



Εικόνα 10 : Συμμετρική συνάρτηση συμμετοχής του Gauss [10]

**Σιγμοειδής.** Έχει σαν παραμέτρους δύο πραγματικούς αριθμούς  $a$ ;  $c$ :

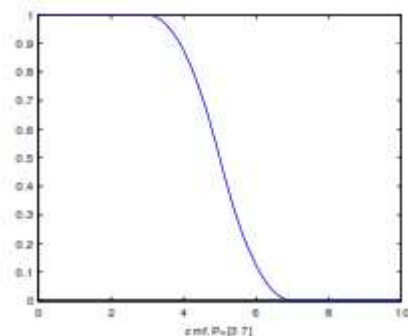
$$\mu_A(x) = \frac{1}{1 + e^{-a(x-c)}}$$



Εικόνα 11 : Σιγμοειδής συνάρτηση συμμετοχής [10]

**Z συνάρτηση.** Έχει σαν παραμέτρους δύο πραγματικούς αριθμούς  $a$ ,  $b$ . Τα  $a$ ,  $b$  δείχνουν τα άκρα της καμπύλης.

$$\mu_A(x) = \begin{cases} 1, & x \leq a \\ 1 - 2 \left( \frac{x-a}{b-a} \right)^2, & a \leq x \leq \frac{a+b}{2} \\ 2 \left( b - \frac{x}{b-a} \right)^2, & \frac{a+b}{2} \leq x \leq b \\ 0, & x \geq b \end{cases}$$

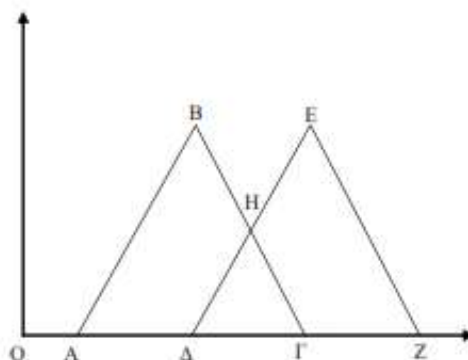


Εικόνα 12: Z συνάρτηση συμμετοχής [10]

### 2.3.4.3 Πράξεις ασαφών συνόλων

Οι πράξεις που μπορούν να γίνουν με τα ασαφή σύνολα είναι η **τομή** (λογικό OR) και η **ένωση** (λογικό AND).

Έστω δύο ασαφή σύνολα **A** και **B** ορισμένα πάνω στο ίδιο μη πεπερασμένο σύνολο **X**: Οι συναρτήσεις συμμετοχής τους φαίνονται στο παρακάτω γράφημα



Εικόνα 13: Ένωση ή Τομή Συνόλων [10]

όπου η συνάρτηση συμμετοχής του **A** είναι αυτή που σχηματίζεται από τα ευθύγραμμα τμήματα **OABΓZ** και του **B** από τα **OΔEZ**. Η τομή **A ∩ B** θα έχει συνάρτηση συμμετοχής την **OΔΗΓZ** ενώ η ένωση **A ∪ B** την **OABHEZ**. Οι πράξεις πριν είναι ορισμένες και για συνεχή και για διακριτά ασαφή σύνολα. Υπάρχουν και κάποιες πράξεις που χρησιμοποιούμε συχνά για να περιγράψουμε κάποιες έννοιες.

**Λεκτικός μετατροπέας** ονομάζεται μια πράξη πάνω σε ένα ασαφές σύνολο που μετατρέπει την λεκτική έννοια αυτού του συνόλου. Χαρακτηριστικό παράδειγμα ενός λεκτικού μετατροπέα είναι το "ΠΟΛΥ".

Στις πράξεις της ένωσης και της τομής συνόλων υπάρχουν και οι ιδιότητες. Αυτές είναι η μεταβατική ιδιότητα, προσεταιριστική ιδιότητα, επιμεριστική ιδιότητα, Ιδιότητα του De Morgan και η απορροφητική ιδιότητα.

### 2.3.4.4 Σχέσεις μεταξύ ασαφών συνόλων

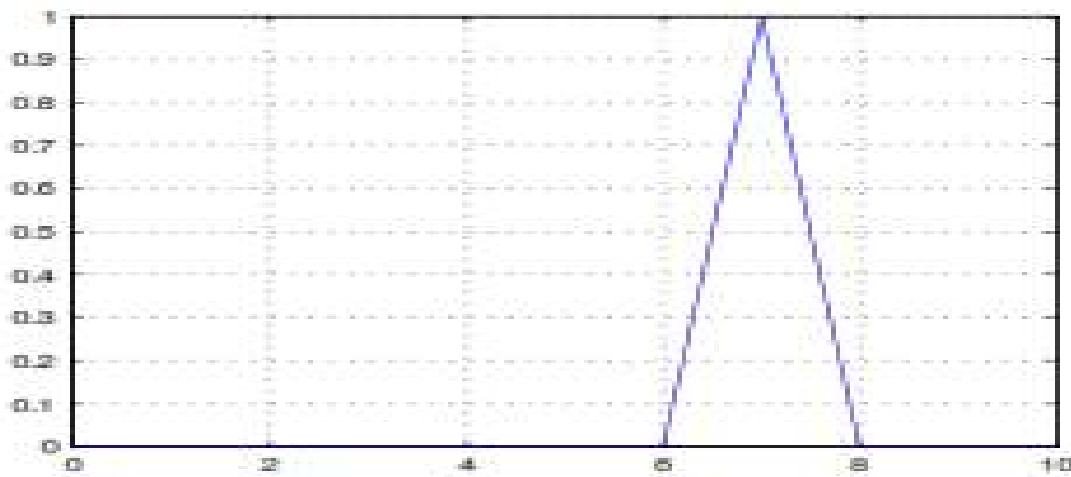
Στον ασαφή έλεγχο η σχέσεις μεταξύ αντικειμένων παίζουν σημαντικό ρόλο. Μερικές σχέσεις αφορούν στοιχεία μέσα στο ίδιο σύνολο, πχ μια μέτρηση είναι μεγαλύτερη από κάποια άλλη. Άλλες σχέσεις ορίζονται μεταξύ διαφόρων συνόλων, πχ μια μέτρηση έχει μεγάλη τιμή και ταχύτητα μεταβολής της είναι θετική κλπ. Ένα απλό παράδειγμα μιας ασαφούς σχέσης είναι η ομοιότητα δύο ανθρώπων. Πχ ο Τάσος μοιάζει με τον Κώστα με βαθμό 0,7 ενώ ο Κώστας με τον Θανάση σε βαθμό 0,3. Το παραπάνω παράδειγμα είναι μια σχέση μεταξύ δύο στοιχείων αλλά γενικά είναι δυνατόν να ορίσουμε σχέσεις με περισσότερα από δύο στοιχεία.

Αντίστοιχα με τις πράξεις μεταξύ ασαφών συνόλων, υπάρχουν και οι πράξεις μεταξύ σχέσεων ασαφών συνόλων. **Τομή** μεταξύ δύο ασαφών σχέσεων  $R_1$  και  $R_2$ , **Ένωση** μεταξύ δύο ασαφών σχέσεων  $R_1$  και  $R_2$  και το **Καρτεσιανό γινόμενο**  $A \times B$  μεταξύ δυο ασαφών συνόλων  $A$  και  $B$  ορίζεται ως η ασαφής σχέση με συνάρτηση συμμετοχής για όλα τα  $x \in A$  και  $y \in B$ .  $(A \times B)(x; y) = \min \{\mu_A(x); \mu_B(y)\}$

Υπάρχουν οι συνθέσεις συνόλων (α)  $\text{Sup} - \min$  ή αλλιώς  $\text{max} - \min$  σύνθεση  $A \circ R$ , (β)  $\text{max} - \text{prod}$  σύνθεση  $A \cdot R$  ενός ασαφούς συνόλου  $A$  και (γ)  $\text{max} - \text{average}$  σύνθεση  $A < + > R$  ενός ασαφούς συνόλου  $A$  του  $X$

### 2.3.5 Ασαφείς Αριθμοί

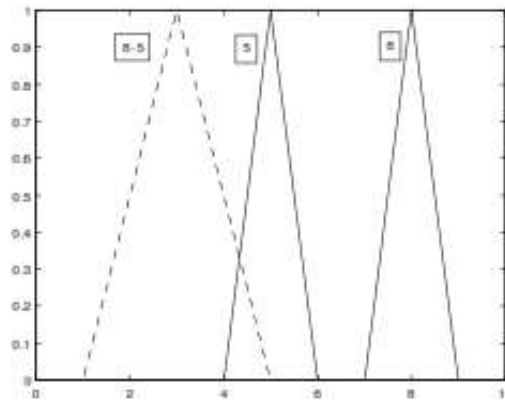
Οι ασαφείς αριθμοί χρησιμοποιούνται σε εφαρμογές όπου είναι επιθυμητή η αναπαράσταση της αβεβαιότητας των αριθμητικών δεδομένων. Ένας εύκολος τρόπος να φανταστούμε ένα ασαφή αριθμό είναι η έκφραση "περίπου 7".



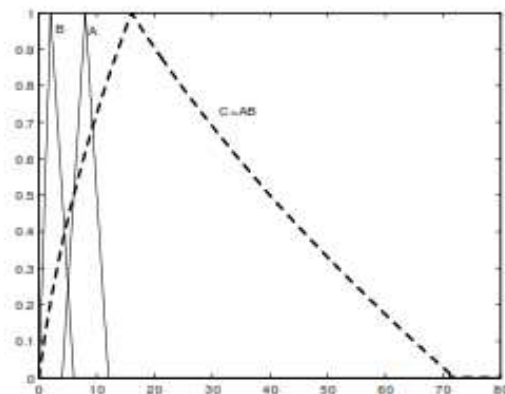
Εικόνα 14: Ο Ασαφής αριθμός 7 [10]

Αντίστοιχα μπορεί να οριστεί ένας ασαφής ορισμός σε διακριτά απειροσύνολα όπως οι ακέραιοι αριθμοί. Όλες οι πράξεις ασαφών συνόλων που έχουμε ορίσει πιο πριν ισχύουν προφανώς και για τους ασαφείς αριθμούς. Επίσης ανάλογα με την εφαρμογή, ο καθένας μπορεί να διαλέξει διαφορετικές συναρτήσεις συμμετοχής για τον ίδιο ασαφή αριθμό.

Οι πράξεις μεταξύ ασαφών αριθμών είναι πρόσθεση, αφαίρεση, πολλαπλασιασμός και η διαίρεση.



Εικόνα 15: Αφαίρεση ασαφών αριθμών [10]



Εικόνα 16: Πολλαπλασιασμός ασαφών αριθμών [10]

### 2.3.6 Συνεπαγωγές

Έστω οι προτάσεις  $p = "x \text{ ανήκει στο σύνολο } A"$  και  $q = "y \text{ ανήκει στο σύνολο } B"$  όπου  $A$  και  $B$  είναι κλασικά σύνολα. Η πρόταση " $p$  συνεπάγεται  $q$ " θα συμβολίζεται  $R : p \rightarrow q$ , ερμηνεύεται ως ότι δεν μπορεί να αληθεύει το  $p$  και να μην αληθεύει το  $q$ .

$$R : q \rightarrow p = \begin{cases} 1, & \tau(p) \leq \tau(q) \\ 0, & \tau(p) > \tau(q) \end{cases}$$

Η ερμηνεία της συνεπαγωγής είναι ότι ο βαθμός αλήθειας της  $p > q$  καθορίζει πόσο το  $q$  αληθεύει κατά τον ίδιο βαθμό όσο το  $p$  που ονομάζεται **αυστηρή συνεπαγωγή**

$$\mu_R(x, y) = \begin{cases} 1, & \mu_A(x) \leq \mu_B(y) \\ 0, & \mu_A(x) > \mu_B(y) \end{cases}$$

**η συνεπαγωγή Godel**

$$\mu_R(x, y) = \begin{cases} 1, & \mu_A(x) \leq \mu_B(y) \\ \mu_B(y), & \mu_A(x) > \mu_B(y) \end{cases}$$

**η συνεπαγωγή Larsen.**

$$\mu_R(x, y) = \mu_A(x)\mu_B(y)$$

και ο πιο διαδεδομένος τελεστής συνεπαγωγής στα ασαφή σύνολα είναι του **Mamdani**

$$\mu_R(x, y) = \min\{\mu_A(x), \mu_B(y)\}$$



### 2.3.7 Προσεγγιστικός συλλογισμός

Στην συμπερασματική συλλογιστική, δεδομένων αληθών προτάσεων το συμπέρασμα που βγαίνει δεν μπορεί να είναι ψευδές. Κλασικό παράδειγμα συμπερασματικής συλλογιστικής είναι το ακόλουθο:

Πρόταση: Όλοι οι άνθρωποι είναι θνητοί

Γεγονός: Ο Σωκράτης είναι άνθρωπος

Συμπέρασμα: Ο Σωκράτης είναι θνητός

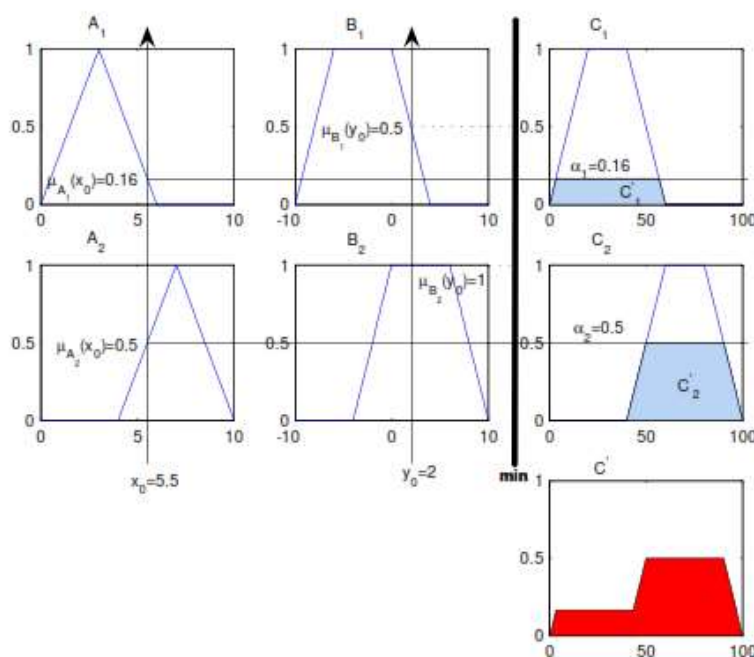
Ο έλεγχος διαδικασιών ή συστημάτων με ασαφείς ελεγκτές προϋποθέτει την ύπαρξη κάποιων λεκτικών κανόνων που περιγράφουν τις αντιδράσεις ενός ανθρώπου χειριστή. Αυτοί οι κανόνες περιγράφονται από ένα σύνολο προτάσεων της μορφής "ΑΝ Α τότε Β". Είναι προφανές ότι σε πολύπλοκες διαδικασίες δεν είναι γνωστοί όλοι οι κανόνες εκ των προτέρων. Άρα ζητείτε ένας μηχανισμός που μπορεί να παίρνει αποφάσεις με ελλιπή στοιχεία, κάτι που η ασαφής λογική αποδεικνύεται ότι μπορεί να κάνει. Στον προσεγγιστικό συλλογισμό και την ασαφή λογική ο σημαντικότερος κανόνας συνεπαγωγής είναι ο **Generalized Modus Ponens (GMP)** για τον οποίο ισχύει :

Πρόταση: R : ΑΝ x είναι Α ΤΟΤΕ y είναι Β

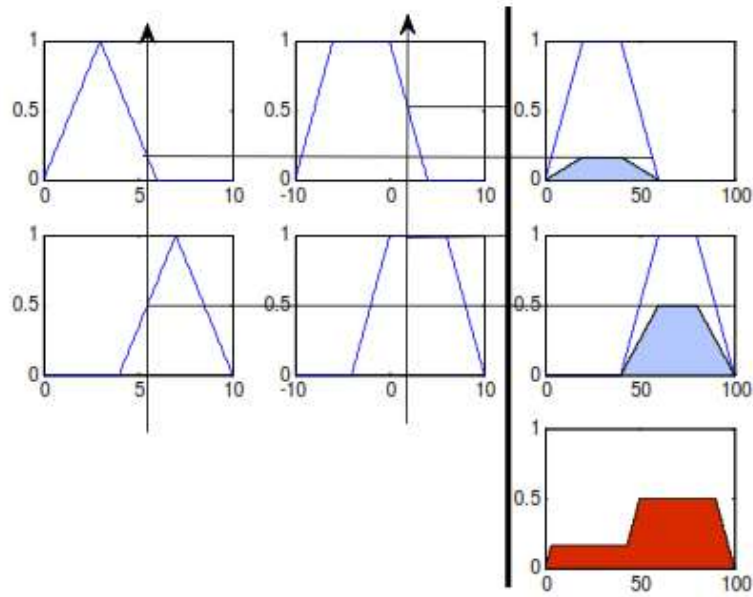
Γεγονός: x είναι Α'

Συμπέρασμα: y είναι Β'

Στόχος είναι η εύρεση ενός συμπεράσματος έχοντας σαν δεδομένα τα αίτια. Το συμπέρασμα Β προκύπτει από την σύνθεση του Α και τον πίνακα αλήθειας της συνεπαγωγής. Η σύνθεση όπως έχουμε ήδη πει μπορεί να οριστεί με διάφορους τελεστές.



Εικόνα 17 : Βαθμός εκπλήρωσης κανόνων με max-min σύνθεση. [10]



Εικόνα 18 : Βαθμός εκπλήρωσης κανόνων με max – prod σύνθεση. [10]

Το "ψαλίδισμα" της συνάρτησης συμμετοχής εξαρτάται από το τελεστή συνεπαγωγής Mamdani που θα επιλέξουμε.

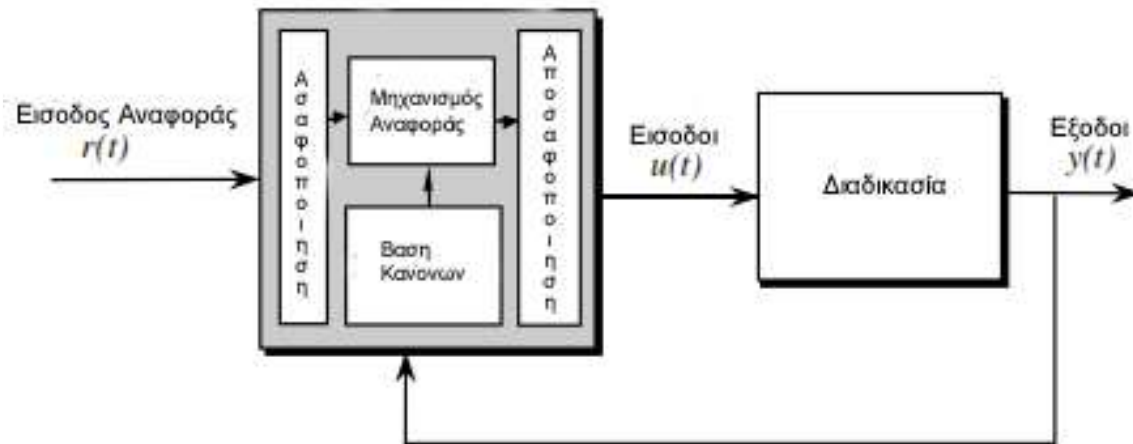
### 2.3.8 Σχεδιασμός Συστήματος Ασαφούς Ελέγχου

Το κυριότερο κίνητρο για την χρήση του ασαφούς ελέγχου είναι η δυσκολία που παρουσιάζεται στο να μοντελοποιήσουμε και να προσομοιώσουμε πολύπλοκα συστήματα για την ανάπτυξη συστημάτων ελέγχου, ειδικά κατά την εφαρμογή. Ακόμη και αν μπορέσει να αναπτυχθεί ένα σχετικά 'ακριβές' μοντέλο ενός δυναμικού συστήματος, είναι συχνά υπερβολικά πολύπλοκο ώστε να χρησιμοποιηθεί στην ανάπτυξη ενός ελεγκτή. Ειδικά για πολλές διαδικασίες σχεδιασμού συμβατικού ελέγχου που απαιτούν περιοριστικά παραδοχές για το σύστημα (π.χ., γραμμικότητα). Αυτός είναι και ο λόγος που στην πράξη συμβατικοί ελεγκτές αναπτύσσονται συχνά μέσω απλών μοντέλων που παρουσιάζουν την συμπεριφορά του συστήματος και ικανοποιούν τις απαραίτητες υποθέσεις, και τον συντονισμό ή την βελτιστοποίηση για την συγκεκριμένη διαδικασία (ad hoc tuning) σχετικά απλών γραμμικών ή μη γραμμικών ελεγκτών.

Είναι κατανοητό ότι η στοχευμένη, στο πρόβλημα επίλυση, εισάγετε στην διαδικασία σχεδιασμού για όσο χρονικό διάστημα ασχολούνται με την πραγματική εφαρμογή του συστήματος ελέγχου. Πρέπει, εξάλλου, να αναγνωριστεί ότι στις προσεγγίσεις συμβατικού ελέγχου της εφαρμοσμένης μηχανικής που χρησιμοποιούν κατάλληλα συστήματα στοχευμένη στο πρόβλημα επίλυση έχουν αποδειχθεί σχετικά επιτυχημένα. Πόσο ποσοστό της επιτυχίας μπορεί να αποδοθεί στη προσέγγιση με την χρήση του μαθηματικού μοντέλο για τον σχεδιασμό του συμβατικού ελέγχου και πόσο πρέπει να αποδοθεί στην στοχευμένη ευρηματική βελτιστοποίηση που έχει χρησιμοποιεί κατά την υλοποίηση του σχεδίου; Και αν μπορούμε να εκμεταλλευτούμε τη χρήση ευεργετικών πληροφοριών που παίρνουμε σε όλη την διαδικασία σχεδιασμού, θα μπορέσουμε να αποκτήσουμε συστήματα ελέγχου υψηλότερης απόδοσης; Ο ασαφής έλεγχος παρέχει μια τυπική μεθοδολογία που παρουσιάζει,

χειρίζεται και υλοποιεί την ευεργετική γνώση του ανθρώπου χειριστή σχετικά με το πώς ελέγχουν ένα σύστημα.

Ένα διάγραμμα ασαφούς ελεγκτή δίνεται στην εικόνα 19, όπου εμφανίζετε ένας ασαφής ελεγκτής ενσωματωμένος σε ένα σύστημα ελέγχου κλειστού βρόχου. Οι εξοδοι του συστήματος δηλώνονται με  $y(t)$ , οι εισοδοι χαρακτηρίζονται με  $u(t)$ , και οι εισοδοι των δεδομένων στον ασαφές ελεγκτή συμβολίζεται με  $r(t)$ .



Εικόνα 19 : Αρχιτεκτονική Ασαφούς Ελεγκτή [59]

Ο ασαφής ελεγκτής έχει τέσσερα κύρια συστατικά στοιχεία:

- η «**βάση των κανόνων**» όπου είναι αποθηκευμένη η γνώση, με τη μορφή της να είναι ένα σύνολο κανόνων, για τον έλεγχο της διαδικασίας. Με την θεωρία των ασαφών συνόλων είναι δυνατή η μετάφραση των λεκτικών κανόνων της βάσης γνώσης σε μαθηματικούς κανόνες.
- Το περιβάλλον **ασαφοποίησης** που μετατρέπει τις εισροές, έτσι ώστε να μπορούν να ερμηνευτούν και να συγκριθούν με τους κανόνες της βάσης. Δηλαδή μετατρέπει τις πραγματικών τιμών των μεταβλητών εισόδου του ελεγκτή σε ασαφή σύνολα.
- τον **μηχανισμό συμπερασμού** που αξιολογεί ποιος κανόνας ελέγχου είναι σχετικός με την τρέχουσα στιγμή και στη συνέχεια να αποφασίσει ποια θα πρέπει να είναι η είσοδος στο σύστημα. Εκεί παράγονται μέσω συνεπαγωγών τα ασαφή σύνολα των συμπερασμάτων.
- Το περιβάλλον **αποσαφοποίησης** που μετατρέπει τα συμπεράσματα που εξάχθηκαν από τον μηχανισμό συμπερασμού σε τιμές για τις εισόδους του συστήματος. Τα ασαφή σύνολα των συμπερασμάτων μετατρέπονται σε πραγματικούς αριθμούς έτσι ώστε να είναι δυνατή η μετάδοση της δράσης ελέγχου στην διαδικασία.

Ο ασαφής ελεγκτής μπορεί να θεωρηθεί ως μια τεχνητή οντότητα που παίρνει αποφάσεις και που λειτουργεί σε ένα σύστημα κλειστού βρόχου σε πραγματικό χρόνο. Συγκεντρώνει τα δεδομένα εξόδου του συστήματος  $y(t)$ , συγκρίνει με τις εισόδους  $r(t)$  αναφοράς και στη συνέχεια αποφασίζει ποια θα είναι η είσοδος στο σύστημα  $u(t)$  ώστε να διασφαλίζεται ότι τηρούνται οι στόχοι απόδοσης.

### 2.3.8.1 Ζητήματα μοντελοποίησης και στόχοι απόδοσης

Η σχεδιαστές ασαφών ελεγκτών υποστηρίζουν την άποψη ότι :

«ένα μοντέλο δεν είναι απαραίτητη για την ανάπτυξη ενός ασαφή ελεγκτή, και αυτό είναι το κύριο πλεονέκτημα της προσέγγισης.». [59]

Ωστόσο, η σωστή κατανόηση της δυναμικής του συστήματος μπορεί να επιτευχθεί χωρίς να χρησιμοποιηθούν οι αρχές της φυσικής για να αναπτυχθεί ένα μαθηματικό μοντέλο; Και μια ακριβής κατανόηση στο πώς να ελέγχετε το σύστημα θα επιτευχθεί χωρίς να βασίζεται σε αξιολόγηση μέσω της προσομοίωσης που επίσης χρειάζεται ένα πρότυπο; Πάντα ξέρουμε περίπου σε βασικές γραμμές την διαδικασία που θέλουμε να ελέγξουμε (π.χ., ξέρουμε ότι πρόκειται για ένα όχημα ή έναν πυρηνικό αντιδραστήρα), και συχνά είναι δυνατόν να παραχθεί τουλάχιστον ένα κατά προσέγγιση μοντέλο, οπότε γιατί να μην το κάνουμε αυτό; Για μια εφαρμογή που η ασφάλεια είναι κρίσιμο στοιχείο, αν δεν θέλουμε να χρησιμοποιήσουμε ένα τυπικό μοντέλο, τότε δεν είναι δυνατό να χρησιμοποιηθεί μια μαθηματική ανάλυση ή εξέλιξη που στηρίζετε σε προσομοίωση και περιγράφει την εφαρμογή.

Είναι σωστό να αγνοηθούν όλα αυτές οι αναλυτικές προσεγγίσεις για μια τέτοια εφαρμογή; Σαφώς, θα υπάρξουν ορισμένες εφαρμογές όπου μπορεί απλά να επιλέξετε έναν ελεγκτή (ασαφή ή συμβατικό) και να γίνει κατευθείαν εφαρμογή. Σε μια τέτοια περίπτωση δεν υπάρχει καμία ανάγκη για ένα τυπικό μοντέλο της διαδικασίας. Ωστόσο, είναι πραγματικά ο τύπος ελέγχου σε αυτό το πρόβλημα τόσο απαιτητικός που να απαιτείται ο ασαφής έλεγχος; Θα μπορούσε μια συμβατική προσέγγιση (όπως ο έλεγχος PID) ή ένα σύστημα «πίνακα παρακολούθησής» να λειτουργεί εξίσου καλά ή μπορεί και καλύτερα, λαμβάνοντας ιδιαίτερα υπόψη την πολυπλοκότητα της υλοποίησης;

Συνολικά, όταν σκέφτεστε τη δυνατότητα να αγνοήσετε τις πληροφορίες που είναι συχνά διαθέσιμες σε ένα μαθηματικό μοντέλο, είναι σαφές ότι αυτό δεν θα είναι και το πιο σοφό μπορεί να γίνει. Ο ρόλος της μοντελοποίησης στην σχεδίαση του ασαφή ελέγχου είναι αρκετά παρόμοια με το ρόλο που αυτή έχει στο σχεδιασμό ενός συστήματος συμβατικού ελέγχου. Στον Ασαφή έλεγχο υπάρχει μια πιο σημαντική έμφαση στη χρήση των heuristics, αλλά και σε πολλές άλλες προσεγγίσεις ελέγχου (π.χ. έλεγχος PID για τον έλεγχο της διαδικασίας), υπάρχει μια παρόμοια έμφαση.

Βασικά, στον ασαφή έλεγχο υπάρχει εστίαση στη χρήση κανόνων που αντιπροσωπεύουν πώς μπορείτε να ελέγχετε το σύστημα και όχι στον έλεγχο των απλών διαφορικών εξισώσεων (ODE). Αυτή η προσέγγιση μπορεί να προσφέρει κάποια πλεονεκτήματα στο ότι η αναπαράσταση της γνώσης σε κανόνες είναι πιο ξεκάθαρη και φυσική σε κάποιους ανθρώπους. Για άλλους, όμως, η χρήση των διαφορικών εξισώσεων είναι πιο κατανοητή και φυσική. Βασικά, υπάρχει απλά μια «διαφορά γλώσσας» μεταξύ ασαφή και συμβατικού ελέγχου:

<<Οι διαφορικές εξισώσεις είναι η γλώσσα του συμβατικού ελέγχου, και οι κανόνες είναι η γλώσσα του ασαφή ελέγχου.>>

Οι στόχοι της απόδοσης και οι περιορισμοί σχεδίασης είναι οι ίδιες με εκείνες για τον συμβατικό έλεγχο που συνοψίζονται παραπάνω, δεδομένου ότι εξακολουθούμε να θέλουμε να πληρούν τα ίδια χαρακτηριστικά με του κλειστού βρόχου. Θεμελιώδη

περιορισμοί που το σύστημα μας τροφοδοτεί και επηρεάζουν στο να επιτύχει έλεγχο υψηλής απόδοσης, εξακολουθούν να παρουσιάζονται όπως ακριβώς και στον συμβατικό έλεγχο (π.χ., η μη-μηδενική φάση ή ασταθής συμπεριφορά παρουσιάζονται ακόμα ως προκλήσεις για τον ασαφή Έλεγχο).

### 2.3.8.2 Σχεδιασμός Ασαφούς Ελεγκτή

Ο σχεδιασμός ενός συστήματος Ασαφή ελέγχου ισοδυναμεί ουσιαστικά με :

- την επιλογή των εισόδων και εξόδων του ασαφούς ελεγκτή,
- να επιλεγεί η προ επεξεργασία που χρειάζεται για τις εισόδους του ελεγκτή και ενδεχομένως την επεξεργασία που απαιτείται για τις εξόδους, και
- σχεδιασμό καθενός από τα τέσσερα συστατικά στοιχεία του ασαφούς ελεγκτή που παρουσιάζονται στην εικόνα 20.

Υπάρχουν τυπικές επιλογές για τις διασυνδέσεις ασαφοποίησης - αποσαφοποίησης. Επιπλέον, πιο συχνά ο σχεδιαστής εγκαθιστά ένα μηχανισμό συμπερασμού που θα μπορεί να χρησιμοποιηθεί σε πολλές διαφορετικές διαδικασίες. Ως εκ τούτου, το κύριο μέρος του ασαφούς ελεγκτή που εστιάζουμε στον σχεδιασμό είναι **η βάση των κανόνων**. Η βάση των κανόνων είναι κατασκευασμένη έτσι ώστε να μοιάζει σαν να υπάρχει ένας ανθρώπινος εμπειρογνώμονα «μέσα στον βρόχο». Οι πληροφορίες που θα φορτωθούν στους κανόνες της βάσης μπορεί να προέρχονται από έναν άνθρωπο, εμπειρογνώμονα που ξόδεψε πολύ χρόνο ώστε να μάθει τον καλύτερο τρόπο για να ελέγχει τη διαδικασία. Σε άλλες περιπτώσεις που δεν υπάρχει η δυνατότητα να χρησιμοποιηθεί η εμπειρία ενός ανθρώπου, ο σχεδιαστής μηχανικός απλά θα μελετήσει την δυναμική του συστήματος (ίσως με τη χρήση μοντέλου και προσομοίωσης) και θα δημιουργήσει ένα σύνολο κανόνων ελέγχου που να προσεγγίζει τον έλεγχο της διαδικασίας.

Για παράδειγμα, ένας κανόνας μπορεί να έχει ένα ανθρώπινο πρόγραμμα οδήγησης είναι «Εάν η ταχύτητα είναι μικρότερη από ό,τι το επιθυμητό, πιάστε προς τα κάτω το πεντάλ του γκαζιού.» Ένας κανόνας που θα αντιπροσώπευε ακόμα πιο λεπτομερείς πληροφορίες σχετικά με τον τρόπο για να ρυθμίσει την ταχύτητα θα ήταν «αν η ταχύτητα είναι μικρότερη από το ορισμένο όριο και η ταχύτητα πλησιάζει το όριο πολύ γρήγορα, τότε αφήστε το για μικρό ποσοστό το πεντάλ του γκαζιού». Στον δεύτερο κανόνα παρουσιάζονται και οι γνώσεις μας σχετικά με τον τρόπο που θα βεβαιωθούμε ότι δεν θα υπερβούμε τον επιθυμητό στόχο μας (το σημείο ταχύτητα). Σε γενικές γραμμές, εάν δημιουργήσουμε μια βάση κανόνων με πολύ λεπτομερείς στοιχεία των γνώσεων, ενισχύουμε τις πιθανότητές για μία καλύτερη απόδοση.

### 2.3.8.3 Αξιολόγηση επιδόσεων ασαφή Ελεγκτή

Κάθε μέθοδος που χρησιμοποιείται για την αξιολόγηση των επιδόσεων των συμβατικών ελεγκτών εφαρμόζεται με ανάλογο τρόπο στην ασαφή λογική. Ο ασαφής ελεγκτής είναι μη γραμμικός ελεγκτής, οπότε πολλές ιδέες συμβατικής μοντελοποίησης, ανάλυσης (μέσω μαθηματικών, προσομοίωσης, ή πειραματισμού) και σχεδιαστικές μπορούν να εφαρμοστούν άμεσα.

Συχνά είναι αρκετά σημαντικό να προσδιοριστούν οι τιμές σε σχέση με τις συμβατικές μεθόδους. Δυστυχώς, πολύ λίγοι έχουν εκτελέσει αναλυτικές και συγκριτικές δοκιμές

μεταξύ συμβατικού και ευφυή ελέγχου και να έχουν λάβει υπόψιν τους ένα ευρύ φάσμα από τις διαθέσιμες συμβατικές μεθόδους (γραμμικό, μη γραμμικό, προσαρμοστικό, κλπ.) καθώς και μεθόδους ασαφών ελέγχων (άμεσο, προσαρμοστικό, εποπτικό), σε θεωρητικό επίπεδο, με προσομοίωση και πειραματικές αναλύσεις, θέματα υπολογιστικής και ούτω καθεξής. Επιπλέον, οι περισσότερες εργασίες πάνω στον ασαφή έλεγχο επικεντρώνονται μόνο στα πλεονεκτήματα του και δεν έχει υπάρξει μια κριτική ματιά στα ποια ενδεχόμενα μειονεκτήματα θα μπορούσαν να υπάρξουν όταν αυτός χρησιμοποιείται. Σαφώς, για ορισμένες εφαρμογές υπάρχει ανάγκη για ανάπτυξη μιας μεθοδολογίας για την ανάπτυξη, εφαρμογή και αξιολόγηση ασαφών ελεγκτών για να εξασφαλιστεί ότι είναι αξιόπιστοι σε σχέση με τα χαρακτηριστικά απόδοσης του.

#### 2.3.8.4 Ασαφοποίηση εισόδων

Οι εισοδοί σε έναν ασαφή ελεγκτή είναι σήματα άρα σαφείς μεταβλητές, γι' αυτό και απαιτείται η ασαφοποίηση τους. Έστω ένας ελεγκτής με δύο εισόδους  $x_1(t); \dots ; x_n(t)$  και μία έξοδο  $y(t)$ . Θεωρούμε ότι στην χρονική στιγμή  $t_0$  έρχονται σαν εισόδος πραγματικές τιμές  $x_1; \dots ; x_n$ . Ο στόχος είναι να παράγουμε με βάση αυτούς τους αριθμούς ασαφή σύνολα  $A_1; \dots ; A_n$ .

Ένας τρόπος να παραχθούν οι ασαφοποιημένες εισοδοί είναι να ορίσουμε τα  $A_i$  σαν ασαφή σημεία με την ακόλουθη συνάρτηση συμμετοχής:

$$\mu_{A'_i}(x_i(t)) = \begin{cases} 1, & x_i(t) = x_i \\ 0, & x_i(t) \neq x_i \end{cases}$$

Ένας δεύτερος τρόπος ασαφοποίησης είναι να λάβουμε υπόψη μας την αβεβαιότητα στα σήματα εισόδου θεωρώντας ότι έχουμε σαν εισόδο έναν ασαφή αριθμό.

#### 2.3.8.5 Μηχανισμός συμπερασμού

Έστω  $n$  το πλήθος κανόνες που αντιστοιχούν στον ελεγκτή της μορφής

"ΑΝ  $x_1$  ΕΙΝΑΙ  $A_1$  ΚΑΙ  $x_2$  ΕΙΝΑΙ  $A_2$  ΤΟΤΕ  $y$  ΕΙΝΑΙ  $B_i$ ".

Μεταξύ των κανόνων υπονοείται το συνδυαστικό "επίσης" που ερμηνεύεται σαν διάζευξη (OR). Οι κανόνες αυτοί αντιστοιχούν σε ασαφείς συνεπαγωγές  $R_i$ . Ο μηχανισμός συμπερασμού για να οριστεί πλήρως χρειάζεται να οριστεί ο τελεστής συνεπαγωγής, ο τελεστής σύνθεσης που χρησιμοποιείται, το συνδυαστικό μεταξύ των  $n$  κανόνων, και ο τελεστής "ΚΑΙ" που ενώνει τις προϋποθέσεις των κανόνων.

| Τελεστής "Η"  | Τελεστής "ΚΑΙ" | Τελεστής συνεπαγωγής | Τελεστής σύνθεσης   |
|---------------|----------------|----------------------|---------------------|
| Mamdani (max) | Mamdani (min)  | Αυστηρή              | Mamdani (max - min) |
| Probor        | Larsen (prod)  | Godel                | Larsen (max prod)   |
|               |                | Larsen (prod)        | max - average       |
|               |                | Mamdani (min)        |                     |

Ο στόχος είναι να παραχθεί ένα ασαφές σύνολο σαν απόφαση του ελεγκτή με την βοήθεια τεχνικών που αναφέρονται παραπάνω

### 2.3.8.6 Αποασαφοποίηση εξόδων

Για να προκύψει τελικά μια σαφής ενέργεια ελέγχου πρέπει σε ασαφές σύνολο A να εφαρμοστεί μια από τις παρακάτω τεχνικές αποασαφοποίησης (εικόνα 21).

Κέντρου βάρους (Center of area - Centroid). Η έξοδος υπολογίζεται από τον τύπο

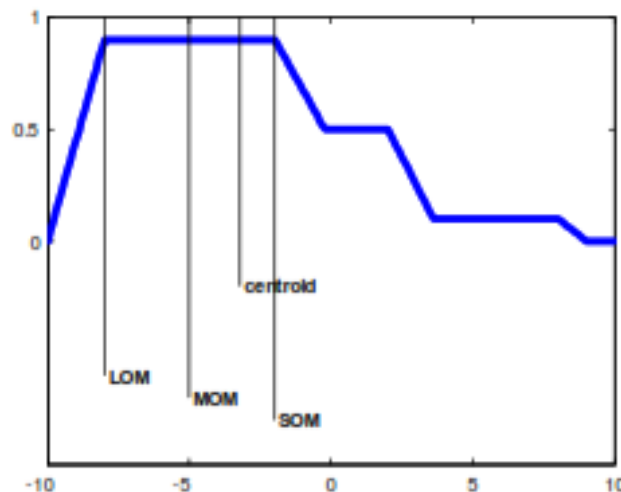
$$z = \frac{\sum x_i \mu_A(x_i)}{\sum \mu_A(x_i)}$$

**Αποασαφοποίηση μικρότερου των μεγίστων (Smallest of maxima - SOM).** Είναι το μικρότερο σε απόλυτη τιμή από τα  $x_i$  που έχουν την μέγιστη τιμή συμμετοχής στο A.

$$z = \frac{1}{N} \sum_{i=1}^N x_i$$

**Αποασαφοποίηση μεγαλύτερου των μεγίστων (Largest of maxima - LOM).** Είναι το μεγαλύτερο σε απόλυτη τιμή από τα  $x_i$  που έχουν την μέγιστη τιμή συμμετοχής στο A.

**Αποασαφοποίηση μέσου των μεγίστων (Middle of maxima - MOM).** Είναι ο μέσος όρος όλων των στοιχείων  $x_i$  που παίρνουν την μέγιστη τιμή στο A.



Εικόνα 20: Διάφοροι μέθοδοι αποασαφοποίησης [10]

### 2.3.8.7 Γνωστοί μηχανισμοί ασαφών ελεγκτών

Οι πιο γνωστοί μηχανισμοί ασαφών ελεγκτών είναι οι **Mamdani**, **Larsen**, **Tsukamoto** και ο Απλοποιημένος **Sugeno-Takagi**.

Ο ασαφής ελεγκτής Mamdani [10] έχει τα παρακάτω χαρακτηριστικά

|                      |                        |
|----------------------|------------------------|
| Ασαφοποίηση          | ασαφές σημείο          |
| Τελεστής "ΚΑΙ"       | min                    |
| Τελεστής "OR"        | max                    |
| Τελεστής συνεπαγωγής | Mamdani ( <i>min</i> ) |
| Τελεστής σύνθεσης    | <i>max - min</i>       |
| Αποασαφοποίηση       | Οτιδήποτε              |

Στο παρακάτω παράδειγμα [10] εξηγείται η λειτουργία του ελεγκτή. Παίρνουμε δύο κανόνες της μορφής:

**R<sub>1</sub>** : AN **x** είναι **A<sub>1</sub>** ΚΑΙ **y** είναι **B<sub>1</sub>** ΤΟΤΕ **z** είναι **C<sub>1</sub>**

**R<sub>2</sub>** : AN **x** είναι **A<sub>2</sub>** ΚΑΙ **y** είναι **B<sub>2</sub>** ΤΟΤΕ **z** είναι **C<sub>2</sub>**

Ορίζουμε σαν εισόδους στον ελεγκτή τα **x = x<sub>0</sub>** και **y = y<sub>0</sub>**. Στόχος είναι να βρεθεί η αριθμητική έξοδος **z**. Μια και έχουμε τελεστή "ΚΑΙ" το **min**, τα επίπεδα ενεργοποίησης των κανόνων υπολογίζονται ως εξής:

$$\alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0)$$

$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0)$$

Ο κάθε κανόνας δίνει έξοδο

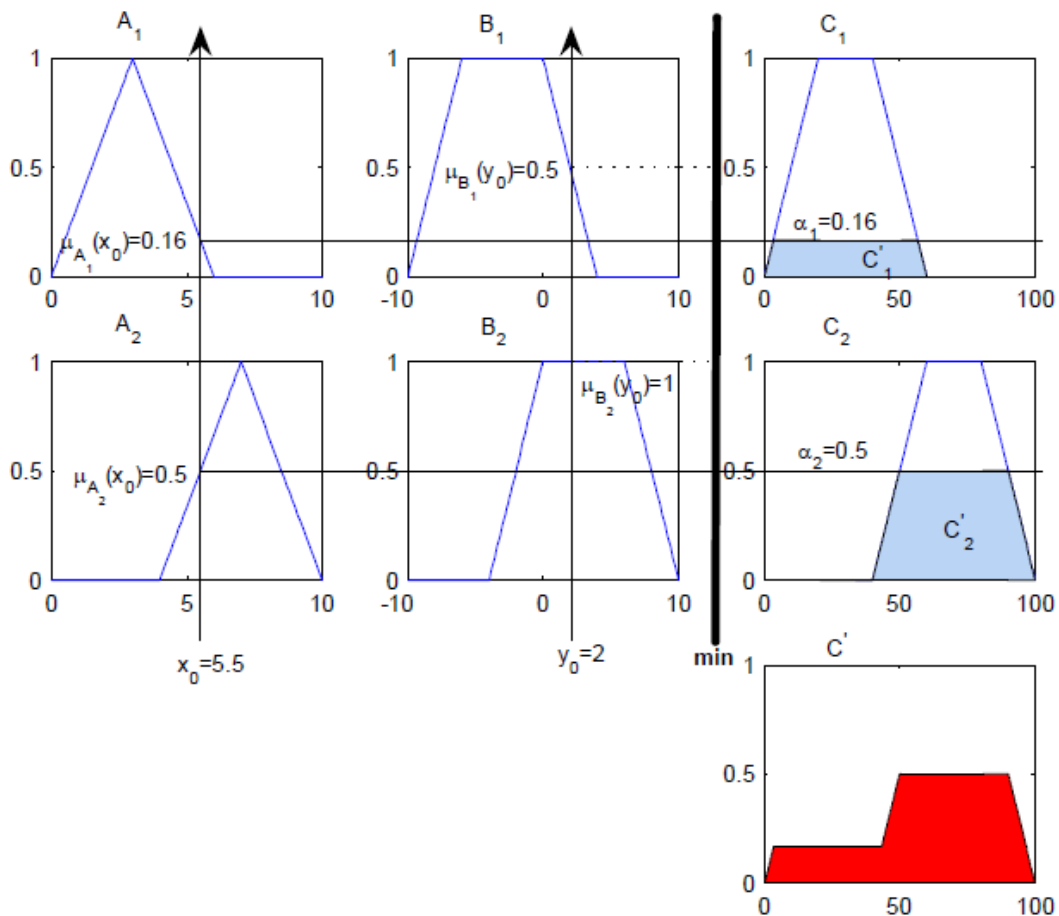
$$\mu_{C_1}(z) = \alpha_1 \wedge \mu_{C_1}(z)$$

$$\mu_{C_2}(z) = \alpha_2 \wedge \mu_{C_2}(z)$$

Και η συνολική έξοδος θα είναι

$$\mu_{C'}(z) = \mu_{C_1}(z) \vee \mu_{C_2}(z)$$

Γραφικά η παραπάνω διαδικασία φαίνεται στο σχήμα 21.

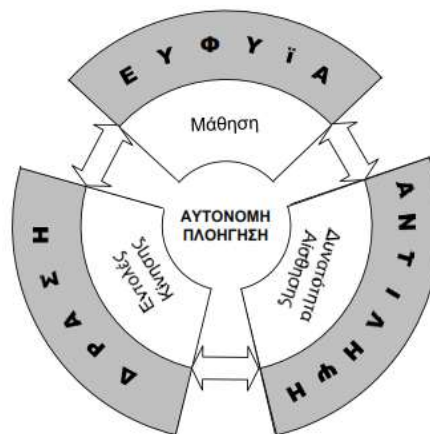


Εικόνα 21 :Ελεγκτής Mamdani [10]



## 2.4 Έλεγχος έντροχου ρομποτικού οχήματος με ασαφή λογική

Τα αυτόνομα έντροχα ρομποτικά οχήματα έχουν τη δυνατότητα να δρουν σε εχθρικό και μη δομημένο περιβάλλον με διαφορετικό βαθμό πολυπλοκότητας. Μέσα σε αυτό το περιβάλλον καλούνται να εκτελέσουν μια αποστολή που τους έχει ανατεθεί, όπως εξερεύνηση, χαρτογράφηση, διάσωση, κ.α. Για να είναι ικανά να εκτελέσουν την αποστολή τους θα πρέπει να έχουν κάποια χαρακτηριστικά, με κύριο την *αντίληψη* του περιβάλλοντος στο οποίο βρίσκονται και τους κινδύνους που μπορεί να αντιμετωπίσουν. Τις πληροφορίες για το περιβάλλον τις λαμβάνουν από κατάλληλα αισθητήρια με τα οποία είναι εξοπλισμένα και συνήθως τις χρησιμοποιούν για τον υπολογισμό της θέσης τους για την κίνησή τους προς κάποιο στόχο, ή την αποφυγή εμποδίων κ.α. Άλλο χαρακτηριστικό που πρέπει να έχουν είναι η *ευφυΐα* ώστε να μπορούν να διαχειριστούν τα δεδομένα από τα αισθητήρια, να μαθαίνουν και να προσαρμόζονται στο άγνωστο περιβάλλον και να κινούνται αυτόνομα μέσα σ' αυτό. Επίσης θα πρέπει να *δρουν* χωρίς τη βοήθεια του ανθρώπου και να φέρουν εις πέρας τις αποστολές που τους έχουν δοθεί [61].



Εικόνα 22 : Αλληλεπίδραση των απαιτούμενων χαρακτηριστικών για αυτόνομη πλοήγηση [61]

## 2.5 Εξελικτικοί υπολογισμοί (Evolutionary computation)

Οι Εξελικτικοί υπολογισμοί είναι μια ομάδα τεχνικών, επίλυσης προβλημάτων συνδυαστικής βελτιστοποίησης. Βασίζονται στην βιολογική θεωρία της εξέλιξης των ειδών του Charles Darwin's, όπως η γενετική κληρονομικότητα και η φυσική επιλογή. Ο εξελικτικός υπολογισμός εφαρμόζεται συνήθως σε συστήματα υπολογιστών και χρησιμοποιούνται για την επίλυση προβλημάτων που έχουν πάρα πολλές μεταβλητές και δεν μπορούν οι παραδοσιακοί αλγόριθμοι να εξετάσουν καθώς και τις περιπτώσεις όπου η προσέγγιση για την επίλυση ενός συγκεκριμένου προβλήματος δεν είναι εύκολα κατανοητή. Χρησιμοποιούν μια επαναληπτική διαδικασία όπου κάποιοι πληθυσμοί ατόμων εξελίσσονται ώστε να ταιριάζουν καλύτερα με το περιβάλλον τους. Χρησιμοποιείται για αυτόνομη αναζήτηση μέσα από μια σειρά πιθανών λύσεων για ένα πρόβλημα ώστε να βρει τελικά μια λύση η οποία θα λύνει επαρκώς το συγκεκριμένο πρόβλημα. Δεν θα βρει όμως απαραίτητα την πιο βέλτιστη λύση!

Εν συντομία οι αρχές της βιολογικής εξέλιξης: υπάρχουν πληθυσμοί οι οποίοι είναι οι ομάδες των ατόμων ενός είδους. Κάθε άτομο κατέχει ένα γονιδίωμα (genome) που αποτελείται από χρωμοσώματα, τα οποία με τη σειρά τους περιέχουν ένα μεγάλο

αριθμό γονιδίων (genes). Τα γονίδια αυτά κωδικοποιούν ή ελέγχουν κληρονομικά χαρακτηριστικά. Το πλήρες γονιδίωμα για ένα άτομο είναι γνωστό ως γενότυπος, το πλήρες σύνολο των παρατηρήσιμων χαρακτηριστικών του ατόμου είναι ο φαινότυπος.

Όταν ανεξάρτητα άτομα ζευγαρώνουν και αναπαράγουν απογόνους μεταβιβάζουν κάποια από τα γνωρίσματα τους από τη μία γενιά στην επόμενη. Κατά τη διαδικασία της αναπαραγωγής, συγκεκριμένοι μηχανισμοί όπως η φυσική επιλογή, η γενετική μετάλλαξη, ο γενετικός ανά-συνδυασμός ή η ροή γονιδίων επιτρέπουν την γενετική παραλλαγή ή αλλαγή. Οποιαδήποτε αλλαγή που ενισχύει τα χαρακτηριστικά του ατόμου γίνεται και παραμένει και στις επόμενες διαδοχικές γενιές αυτού του πληθυσμού.

Συνήθως πολλοί περισσότεροι απόγονοι παράγονται από τον αριθμό που το περιβάλλον μπορεί να υποστηρίξει και αυτοί οι απόγονοι διαφέρουν ως προς την ικανότητά τους να επιβιώσουν ή να αναπαραχθούν. Ως αποτέλεσμα αυτού, επέρχεται ο ανταγωνισμός για την επιβίωση και την αναπαραγωγή. Άτομα με ευνοϊκά χαρακτηριστικά τα οποία είναι τα πλέον καλύτερα προσαρμοσμένα στο περιβάλλον τους, στην εξελικτική διαδικασία, έχουν μεγαλύτερη «ικανότητα» και είναι σε θέση να επιβιώσουν, να αναπαραχθούν και να μεταδώσουν τα γενετικά χαρακτηριστικά τους σε αυξανόμενους αριθμούς στις επόμενες γενιές. Με την πάροδο του χρόνου, αυτή η χαρακτηριστική «ικανότητα» γίνεται κυρίαρχη ανάμεσα στον πληθυσμό.

Η βιολογική διαδικασία της εξέλιξης, αναπαράγεται στον εξελικτικό υπολογισμό στον οποίο το χρωμόσωμα μπορεί να περιγράψει μηχανικές ή εμπορικές παραμέτρους, όπως το χρονοδιάγραμμα προσγειώσεων ενός αεροσκάφους. Αλλάζοντας το χρωμόσωμα, στην ουσία τροποποιείτε το χρονοδιάγραμμα προσγειώσεων, και με την εφαρμογή της εξελικτικής διαδικασίας, εξελίσσεται το χρωμόσωμα και κατ' επέκταση εξελίσσεται το χρονοδιάγραμμα των προσγειώσεων.

Οι τεχνικές του εξελικτικού υπολογισμού χρησιμοποιούνται σε ένα ευρύ φάσμα εμπορικών εφαρμογών όπως χρονικός προγραμματισμός, βίο-Πληροφορική, οικονομικά, ανάλυση χρηματοπιστωτικών αγορών, τηλεπικοινωνίες, απεικόνιση, σχεδίαση ολοκληρωμένων κυκλωμάτων και σχεδιασμός φαρμάκων. Ακόμα βρίσκουν πεδίο εφαρμογής και στην επιστημονική έρευνα αιχμής όπως είναι η ανασύσταση πρωτεϊνών.

Ο Εξελικτικός Υπολογισμός αποτελείται από πολλούς τομείς με την ευρεία έννοια του όρου:

- Βελτιστοποίηση στηριζόμενη στη δομή μιας αποικίας μυρμηγκιών
- Τεχνητό ανοσοποιητικό σύστημα
- Τεχνητή ζωή (ή ψηφιακός οργανισμός)
- Αλγόριθμοι στηριζόμενοι στην πολιτιστική συμπεριφορά.
- Διαφορική εξέλιξη
- Εξέλιξη διπλής-φάσης
- Αλγόριθμος με εκτίμηση της διασποράς.

- Εξελικτικοί αλγόριθμοι
- Εξελικτικός προγραμματισμός
- Εξελικτική στρατηγική
- Προγραμματισμός γονιδιακής έκφρασης
- Γενετικοί αλγόριθμοι
- Γενετικός προγραμματισμός
- Εξέλιξη της γραμματικής
- Μοντέλο Μαθησιακής Εξέλιξης
- Μιμητικός αλγόριθμος
- Βελτιστοποίηση βάση σμήνους σωματιδίων
- Αυτό-οργάνωση όπως η αυτό-οργάνωση χαρτών, ανταγωνιστική μάθηση
- Νοημοσύνη σμήνους

Ποιο διαδεδομένοι είναι οι τέσσερις ευρείς τομείς:

- Γενετικοί αλγόριθμοι,
- Γενετικός προγραμματισμός,
- Εξελικτικές στρατηγικές και
- Εξελικτικός προγραμματισμός.

### 2.5.1 Γενετικοί αλγόριθμοι

Οι γενετικοί αλγόριθμοι χρησιμοποιούνται για να βρεθούν λύσεις σε ένα πρόβλημα χρησιμοποιώντας την φυσική επιλογή ως μια μηχανή αναζήτησης. Για παράδειγμα ένα πρόβλημα θα μπορούσε να είναι η δημιουργία ενός χρονοδιαγράμματος προσγειώσεων αεροσκαφών στο αεροδρόμιο. Ενεργούν πάνω σε έναν πληθυσμό ατόμων ή χρωμοσωμάτων. Εντός αυτού του πληθυσμού, τα χρωμοσώματα είναι οι πιθανές υποψήφιες λύσεις για το πρόβλημα που χρειάζεται να επιλυθεί. Τα χρωμοσώματα αποτελούνται από διάφορες μορφές όπως bits, αριθμούς ή ακολουθίες παραμέτρων, ανάλογα με το πρόβλημα.

Ο Γενετικός Αλγόριθμος είναι μια επαναληπτική διαδικασία και αποτελείται από τρεις κύριες διεργασίες: αναπαραγωγή, αξιολόγηση καταλληλότητας και επιλογής γονέων.

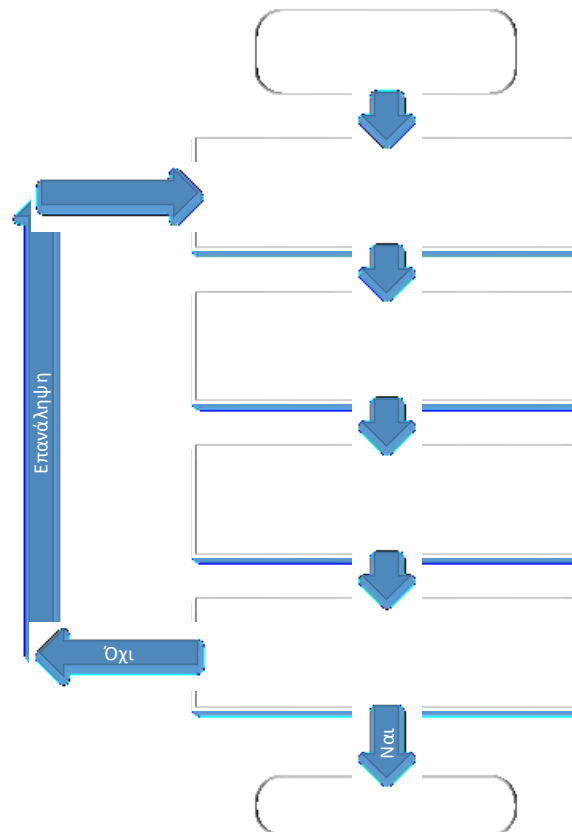
**Αναπαραγωγή** είναι η γενιά των απογόνων από τον επιζήσαντα πληθυσμό των ατόμων ή των χρωμοσωμάτων. Η αναπαραγωγή χρησιμοποιεί δύο γενετικούς τελεστές:

- **crossover**, όπου τα χρωμοσώματα ανταλλάσσονται μεταξύ των γονέων, και
- **μετάλλαξη**, όπου μεταβάλλονται τυχαία τμήματα των χρωμοσωμάτων των γονιών.

**Αξιολόγηση καταλληλότητας** όπου καθορίζετε πόσο καλά το κάθε άτομο ή χρωμόσωμα από τον υπάρχον πληθυσμό λειτουργεί ως μια πιθανή λύση στο πρόβλημα.

**Η επιλογή γονέων** καθορίζει ποια άτομα ή χρωμοσώματα από των υπάρχων πληθυσμό τα οποία θα επιβιώσουν και θα γίνουν γονείς, βάση της καταλληλότητας τους, ώστε να παράγουν τα παιδιά που θα αποτελούν την επόμενη γενιά.

Τα βήματα της γενετικών αλγορίθμων φαίνονται στην εικόνα 23.



Εικόνα 23: Διαδικασία Γενετικού Αλγόριθμου [60]

Αρχικά, δημιουργείται ένας τυχαίος πληθυσμός χρωμοσωμάτων (υποψήφιες λύσεις). Ο πληθυσμός αυτός χρησιμοποιείται για να αναπαραχθούν απόγονοι, συνδυάζοντας τα χρωμοσώματα αυτά χρησιμοποιώντας συνδυασμό (crossover) και μετάλλαξη (mutation). Το κάθε νέο χρωμόσωμα αξιολογείται και καθορίζεται πόσο καλά αυτό λύνει το πρόβλημα που του έχει ανατεθεί, και στη συνέχεια γίνεται η βαθμολόγηση καταλληλότητας ανάλογα του των αποτελεσμάτων της επίλυσης της λύσης. Τα χρωμοσώματα με το καλύτερο βαθμό καταλληλότητας διατηρούνται (επιλογή) και γίνονται οι γονείς που χρησιμοποιούνται για τη δημιουργία νέων απογόνων.

Οι διαδικασίες της αναπαραγωγής, αξιολόγησης καταλληλότητας και επιλογής γονέων επαναλαμβάνονται έως ότου είτε επιτυγχάνεται ο απαιτούμενος βαθμός καταλληλότητας ή όταν έχει ολοκληρωθεί ένας συγκεκριμένος αριθμός γενεών.

Για την πλήρη κατανόηση παρουσιάζεται ένα παράδειγμα [60] ορισμού ωραρίου εργασίας Νοσηλευτών το οποίο χρησιμοποιεί έναν γενετικό αλγόριθμο για να δημιουργήσει ένα κατάλογο ονομάτων για τους νοσηλευτές που εργάζονται σε βάρδιες (δώδεκα ωρών) σε ένα νοσοκομείο. Οι δύο μεταβλητές που εμπλέκονται είναι

(1) τα ονόματα των νοσοκόμων και (2) τους χρόνους που είναι υποχρεωμένοι να εργάζονται. Το χρωμόσωμα είναι μια σειρά από γράμματα, όπου το γράμμα αντιπροσωπεύει το νοσοκόμο/α, ενώ η σειρά κατά την ακολουθία δηλώνει το χρόνο που είναι υποχρεωμένοι να εργάζονται. Στο παράδειγμα, υπάρχουν πέντε νοσοκόμοι/ες, Σάσσα (S), Μιχάλης (M), Βασίλης (B) και Τάκης (T). Οι χρόνοι που απαιτούνται εμφανίζονται στον Πίνακας 1.

Ένα πιθανό χρωμόσωμα είναι το [B, R, S, M, T, B, R, S, M, T, B, R, S, M].

| Θέση στο Χρωμόσωμα | 1                    | 2                    | 3                     | ..... | 12                   | 13                   |
|--------------------|----------------------|----------------------|-----------------------|-------|----------------------|----------------------|
| Χρόνος Εργασίας    | Δευτέρα<br>7πμ – 7μμ | Δευτέρα<br>7μμ – 7πμ | Τρίτη<br>7πμ –<br>7μμ |       | Κυριακή<br>7πμ – 7μμ | Κυριακή<br>7μμ – 7πμ |

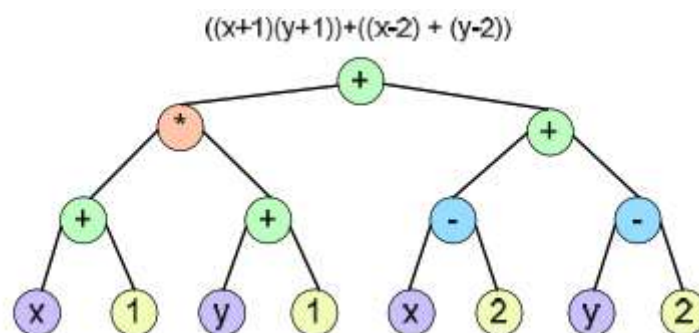
Πίνακας 1 Χρωμόσωμα Ωραρίου Νοσηλευτών [60]

Η αξιολόγηση της καταλληλότητας θα καθορίζει πόσο καλά το χρωμόσωμα πληρούνει τα κριτήρια του καταλόγου. Αυτά τα κριτήρια εξαρτώνται από τις απαιτήσεις του χρονοπρογραμματισμού, για παράδειγμα, μια νοσοκόμα δεν θα μπορεί να εργαστεί περισσότερο από ένα 12 ώρες σε μια μέρα, μια νοσοκόμα δεν θα μπορεί να εργαστεί περισσότερο από τρεις βάρδιες την εβδομάδα και θα ήταν προτιμότερο οι νοσοκόμοι/ες να εργάζονται τρεις βάρδιες μέρα ή τρεις βάρδιες νύχτα στη σειρά. Μια πιο προηγμένη αξιολόγηση καταλληλότητας θα επιτρέπει στους νοσηλευτές να μπορούν να κάνουν αίτηση για ημέρες ή εβδομάδες των προγραμματισμένων διακοπών.

### 2.5.2 Γενετικός Προγραμματισμός

Ο Γενετικός προγραμματισμός αναπτύχθηκε αρχικά από τον John Koza [12]. Η διαδικασία είναι παρόμοια με αυτή των γενετικών αλγορίθμων, εκτός από το ότι το χρωμόσωμα είναι ένα πρόγραμμα υπολογιστή και όχι μια πιθανή λύση, όπως γίνεται στους γενετικούς αλγορίθμους. Ως εκ τούτου ο γενετικός προγραμματισμός χρησιμοποιείται για να εξελιχθεί ένα πρόγραμμα υπολογιστή.

Το πρόγραμμα παρουσιάζεται ως μια δενδροειδής δομή με τους κλάδους του δέντρου να αντιπροσωπεύουν τις ρουτίνες του προγράμματος, ενώ τα φύλλα στα κλαδιά αντιπροσωπεύουν τις μεταβλητές και τις σταθερές. Η δομή του δέντρου μεταβάλλεται κατά την εξελικτική διαδικασία. Η καταλληλότητα αξιολογείται εκτελώντας το πρόγραμμα κατά πόσο καλά αποδίδει.



Εικόνα 24 : Δενδροειδής Παρουσίαση του Γενετικού Προγραμματισμού[60]

Η εικόνα 24 δείχνει πώς ένα αρχικό πρόγραμμα  $((x+1)(y+1)) + ((x-2) + (y-2))$  μπορεί να παρουσιαστεί σε δομή δέντρου. Οι μεταβλητές και οι σταθερές παρατίθενται στο κάτω μέρος του δέντρου με τους αριθμητικούς τελεστές στα κλαδιά.

Σε αυτό το παράδειγμα το 'x' και το '1' προστίθενται μαζί, το 'y' και το '1' προστίθενται μαζί και στη συνέχεια πολλαπλασιάζονται οι δύο αυτοί κλάδοι να δώσει  $(x+1)(y+1)$ .

Ομοίως, το '2' αφαιρείται από το 'x' και το '2' αφαιρείται από το 'y'. Οι δύο αυτοί κλάδοι προστίθενται να δώσουν το  $(x-2) + (y-2)$ .

Αυτή είναι η δομή του δέντρου που έχει εξελιχθεί για να παράγουν ένα νέο πρόγραμμα σε κάθε γενιά.

### 2.5.3 Εξελικτική Στρατηγική

Η εξελικτική στρατηγική αναπτύχθηκε από τους Rechenberg [13, 14] και Schwefel [15] στο τεχνικό Ινστιτούτο του Βερολίνου. Έχει σχεδιαστεί για να δίνει λύσεις σε προβλήματα τεχνικής βελτιστοποίησης (αρχικά σε πειράματα αεροσήραγγας) όπου τα άτομα αντιπροσωπεύουν το φαινότυπο που αποτελείται από τα χαρακτηριστικά του αντικειμένου που επιθυμούμε να βελτιστοποιήσουμε. Αυτά τα άτομα είναι πραγματικοί αριθμοί και εξελίσσονται χρησιμοποιώντας της ίδιες τεχνικές με τους γενετικούς αλγόριθμους (διασταύρωσης και επιλογής). Το ποσό της μετάλλαξης καθορίζεται εντούτοις από μια Gaussian τυχαία τιμή.

Σε γενικές γραμμές, οι εξελικτικές στρατηγικές χρησιμοποιούν προσαρμοσμένο ρυθμό μετάλλαξης, όπου ο ρυθμός μετάλλαξης είναι μέρος του χρωμοσώματος και έχει την δυνατότητα να αλλάζει μετά από κάθε γενιά. Εδώ εφόσον ο απόγονος έχει καλύτερη τιμή καταλληλότητας θα αντικαταστήσει τους γονείς.

### 2.5.4 Εξελικτικός προγραμματισμός

Ο Εξελικτικός προγραμματισμός αναπτύχθηκε από τους *Fogel et al* στην πρόωρη δεκαετία του 1960 [16, 17] κυρίως ως μέσο για την επίτευξη τεχνητής νοημοσύνης μέσω της εξέλιξης, αλλά χρησιμοποιήθηκε έκτοτε για προβλήματα βελτιστοποίησης. Ο όρος εξελικτικός προγραμματισμός προήλθε από τα αρχικά πειράματα, όπου ένας πληθυσμός μηχανών πεπερασμένων καταστάσεων εμπλέκονταν στο να περιγράψουν τη συμπεριφορά ενός προγράμματος λογισμικού. Αργότερα, τα άτομα αυτά επεκτάθηκαν σε τομείς με πραγματικά πρόβλημα όπως διανύσματα πραγματικών τιμών, λίστες διάταξης ή παραγγελίας, δέντρα ή μηχανές πεπερασμένων καταστάσεων.

## 2.6 Γενετικοί Αλγόριθμοι

### 2.6.1 Ιστορική αναδρομή

Τις δεκαετίες του 1950–1960 πολλοί επιστήμονες της επιστήμης των υπολογιστών, πραγματοποίησαν μελέτες σε εξελικτικά συστήματα με την σκέψη ότι η εξέλιξη θα μπορούσε να χρησιμοποιηθεί ως εργαλείο βελτιστοποίησης για προβλήματα που απασχολούσαν μηχανικούς. Η ιδέα σε όλα αυτά τα συστήματα ήταν να αναπτύξουν έναν πληθυσμό «υποψηφίων» λύσεων σε ένα δεδομένο πρόβλημα χρησιμοποιώντας

τελεστές παρόμοιους με αυτούς που χρησιμοποιεί η φύση κατά την διάρκεια γενετικών παραλλαγών και φυσικής επιλογής.

Η φύση έχει έναν πολύ ισχυρό μηχανισμό εξέλιξης των οργανισμών, που βασίζεται στον ακόλουθο κανόνα της φυσικής επιλογής: οι οργανισμοί που δε μπορούν να επιβιώσουν στο περιβάλλον τους πεθαίνουν, ενώ οι υπόλοιποι πολλαπλασιάζονται μέσω της αναπαραγωγής. Οι απόγονοι παρουσιάζουν μικρές διαφοροποιήσεις από τους προγόνους τους, ενώ συνήθως υπερισχύουν αυτοί που συγκεντρώνουν τα καλύτερα χαρακτηριστικά.

Αν το περιβάλλον μεταβάλλεται με αργούς ρυθμούς, τα διάφορα είδη μπορούν να εξελίσσονται σταδιακά ώστε να προσαρμόζονται σε αυτό. Αν όμως συμβούν ραγδαίες μεταβολές, αρκετά είδη οργανισμών θα εξαφανιστούν. Σποραδικά, συμβαίνουν τυχαίες μεταλλάξεις, από τις οποίες οι περισσότερες οδηγούν τα μεταλλαγμένα άτομα στο θάνατο, αν και είναι πιθανό, πολύ σπάνια όμως, να οδηγήσουν στη δημιουργία νέων «καλύτερων» οργανισμών.

Η θεωρία της εξέλιξης (evolution) έχει χρησιμοποιηθεί σε μία κατηγορία αλγορίθμων επίλυσης προβλημάτων, που ονομάζονται *γενετικοί αλγόριθμοι (genetic algorithms)*. Η εισαγωγή των αλγορίθμων αυτών έγινε το 1958 από τον *Friedberg*, ο οποίος επιχείρησε να συνδυάσει μικρά προγράμματα FORTRAN, ωστόσο τα προγράμματα που προέκυψαν τις περισσότερες φορές δεν ήταν εκτελέσιμα. Οι γενετικοί αλγόριθμοι τελικά επινοήθηκαν από τον John Holland την δεκαετία του 1960 και αναπτύχθηκαν από τους φοιτητές του και τους συνεργάτες του στο πανεπιστήμιο του Μίσιγκαν.

Είναι μια μέθοδος για μεταφορά από έναν πληθυσμό «χρωμοσωμάτων» σε έναν άλλο πληθυσμό χρησιμοποιώντας ένα είδος φυσικής επιλογής μαζί με γενετικούς τελεστές όπως:

- Διασταύρωσης (crossover), δηλαδή ανταλλαγή γενετικού υλικού μεταξύ δύο χρωμοσωμάτων από απλοειδείς γονείς.
- Μετάλλαξης (mutation), δηλαδή αναστροφή γενετικού υλικού σε έναν τυχαία επιλεγμένο τόπο.
- Αντιστροφής (inversion), δηλαδή ανακατανομή της διάταξης των γονιδίων σε ένα χρωμόσωμα.
- Επιλογής (selection), δηλαδή του διαχωρισμού των χρωμοσωμάτων που θα χρησιμοποιηθούν για την παραγωγή της επόμενης γενεάς.

Ο γενετικός αλγόριθμος εκτελεί μία αναζήτηση στο χώρο των υποψηφίων λύσεων, με στόχο την εύρεση κάποιας λύσης που μεγιστοποιεί τη συνάρτηση καταλληλότητας (αντικειμενική συνάρτηση κόστους). Η αναζήτηση αυτή είναι παράλληλη, καθώς σε κάθε υποψήφια λύση μπορεί να εκτελεστεί ξεχωριστή αναζήτηση. Η μέθοδος της αναζήτησης μπορεί να θεωρηθεί σαν *αναρρίχηση λόφου*, καθώς γίνονται μικρές αλλαγές στις υποψήφιες λύσεις του πληθυσμού και επιλέγονται πάντα οι καλύτερες, βάση της συνάρτησης καταλληλότητας. Η αναζήτηση επικεντρώνεται στις περισσότερο κατάλληλες λύσεις, χωρίς όμως να αγνοούνται οι υπόλοιπες, καθώς υπάρχει πάντα ο κίνδυνος να παγιδευτεί η διαδικασία σε τοπικό μέγιστο.

Μέχρι τις αρχές της δεκαετίας του 1980 η μελέτη των γενετικών αλγορίθμων περιοριζόταν σε θεωρητικό επίπεδο, με λίγες πρακτικές εφαρμογές. Στην περίοδο αυτή οι γενετικοί αλγόριθμοι χρησιμοποιούνταν κυρίως για προβλήματα βελτιστοποίησης συναρτήσεων, χρησιμοποιώντας χρωμοσώματα συγκεκριμένου μήκους, δυαδικής κωδικοποίησης. Χαρακτηριστικές είναι οι μελέτες των Hollstien και De Jong την περίοδο αυτή.

Ο Hollstien πραγματοποίησε ανάλυση για το πώς επιδρούν οι τελεστές της επιλογής και της διασταύρωσης στην επίδοση του αλγορίθμου. Ο De Jong προσπάθησε να προσδιορίσει τα στοιχεία εκείνα που εμποδίζουν τους αλγορίθμους να φτάσουν γρήγορα σε σύγκλιση.

Από την αρχή της δεκαετίας του '80, η επιστημονική κοινότητα που ασχολείται με τη θεωρία και εφαρμογή των γενετικών αλγορίθμων, έχει δημιουργήσει μία πληθώρα πρακτικών εφαρμογών που εκτείνονται σε πολλούς τομείς ερευνητικής και όχι μόνο δραστηριότητας. Βελτιώνοντας την απόδοση των γενετικών με τη ρύθμιση και βελτιστοποίηση των γενετικών τελεστών, αποδεικνύεται ότι οι γενετικοί αλγόριθμοι μπορούν να χρησιμοποιηθούν σε διάφορες κατηγορίες προβλημάτων και αποτελούν μία ισχυρή μέθοδο βελτιστοποίησης.

## 2.6.2 Βασικές έννοιες

### 2.6.2.1 Γονίδια και χρωμοσώματα

Το γονίδιο αποτελεί τη βασική δομική μονάδα στην γενετική αλλά και στην μέθοδο βελτιστοποίησης με τη χρήση γενετικών αλγορίθμων. Τα γονίδια αποτελούν την κωδικοποιημένη παράσταση των παραμέτρων βελτιστοποίησης ενώ τα χρωμοσώματα περιέχουν το σύνολο των γονιδίων.

Πιο συγκεκριμένα, κάθε υποψήφια λύση παριστάνεται με μια *συμβολοσειρά* ενός πεπερασμένου αλφαβήτου. Κάθε στοιχείο της συμβολοσειράς είναι ένα γονίδιο και η ίδια η συμβολοσειρά είναι ένα χρωμόσωμα.

Στην φύση (DNA) τα αλφάβητο έχει μήκος τέσσερα και αποτελείται από τα στοιχεία A,G,T και C ενώ στους γενετικούς αλγορίθμους χρησιμοποιείται συνήθως δυαδικό αλφάβητο. Ωστόσο υπάρχουν περιπτώσεις γενετικών αλγορίθμων που χρησιμοποιούν πιο σύνθετες μορφές αναπαράστασης (π.χ. πραγματικούς αριθμούς, αλφαριθμητικά στοιχεία κλπ).

### 2.5.2.2 Γονείς

Η διαδικασία επιλογής γονέων σχετίζεται με την απόδοση πιθανοτήτων επιλογής προς αναπαραγωγή στα μέλη ενός πληθυσμού υποψηφίων λύσεων. Κατά τη διαδικασία αυτή, κάποιιοι γονείς με υψηλή τιμή στη συνάρτηση καταλληλότητας ενδέχεται να επιλεγούν προς αναπαραγωγή περισσότερες από μία φορές, ενώ κάποιιοι γονείς με χαμηλή καταλληλότητα ενδέχεται να μην επιλεγούν καθόλου.

Κατά τη διαδικασία επιλογής, αρχικά οι υποψήφιοι λύσεις αντιγράφονται σε μια *δεξαμενή ζευγαρώματος (mating pool)*. Η δεξαμενή αυτή έχει μέγεθος ίσο με τον αρχικό πληθυσμό, και σε αυτήν αντιγράφονται μέλη του αρχικού πληθυσμού, με πιθανότητα ανάλογη της καταλληλότητας τους.



Έπειτα από την διαδικασία αρχικοποίησης του πληθυσμού, όπου δημιουργείται η πρώτη γενιά, επιλέγονται ζευγάρια χρωμοσωμάτων από τον πληθυσμό τα οποία ονομάζονται *γονείς*. Η επιλογή γίνεται με πιθανότητά και σταθμισμένα από την συνάρτηση κόστους.

Για παράδειγμα ένα τυπικό σενάριο επιλογής γονέων αποτελεί η μέθοδος «ρουλέτα», κατά την οποία σε κάθε χρωμόσωμα του πληθυσμού αντιστοιχίζεται σε τμήμα της ρουλέτας. Το μέγεθος του τμήματος αυτού είναι αντίστοιχο της τιμής της σχετικής συνάρτησης κόστους του χρωμοσώματος. Η ρουλέτα περιστρέφεται κάθε φορά που αναζητούμε ένα γονέα. Χρωμοσώματα τα οποία κατέχουν μεγάλο τμήμα της ρουλέτας, έχουν μεγαλύτερη πιθανότητα να εκλεγούν και να περάσουν τα χαρακτηριστικά τους στην επόμενη γενιά.

Ο τρόπος επιλογής των γονέων που θα ζευγαρώσουν επηρεάζει σημαντικά την απόδοση των γενετικών αλγορίθμων. Δυο προβλήματα που εμφανίζονται συχνά στους γενετικούς αλγόριθμους και τα οποία μπορούν να αντιμετωπιστούν με τροποποίηση της διαδικασίας επιλογής, είναι η *πρόωρη σύγκλιση* (*premature convergence*) και η *αργή σύγκλιση* (*slow convergence*). Με τον όρο «σύγκλιση» εννοούμε την επικράτηση ενός χρωμοσώματος ή μικρών παραλλαγών του, σε μεγάλο ποσοστό στον πληθυσμό.

Κατά την *πρόωρη σύγκλιση*, ο πληθυσμός πολύ γρήγορα συγκλίνει γύρω από κάποιο χρωμόσωμα, το οποίο όμως αποτελεί τοπικό μέγιστο (ή ελάχιστο). Το αποτέλεσμα είναι να μην μπορεί πλέον ο γενετικός αλγόριθμος να ξεφύγει από αυτό το τοπικό μέγιστο, παρά μόνο με τη διαδικασία της μετάλλαξης, το οποίο όμως έχει πρακτικά μηδενική πιθανότητα να συμβεί. Το φαινόμενο αυτό εμφανίζεται σε περιπτώσεις που η συνάρτηση καταλληλότητας εμφανίζει πολύ απότομες μεταβολές και έντονα τοπικά μέγιστα και μπορεί να αντιμετωπιστεί με δύο τρόπους. Ο πρώτος είναι η απεικόνιση της συνάρτησης καταλληλότητας σε μία νέα συνάρτηση, λιγότερο απότομη (*fitness remapping*). Ο δεύτερος είναι ο καθορισμός ελάχιστων και μέγιστων ορίων, όσον αφορά το πόσες φορές επιλέγεται ένα χρωμόσωμα προς αναπαραγωγή σε κάθε κύκλο ανανέωσης του πληθυσμού.

Η *αργή σύγκλιση* είναι ουσιαστικά το ακριβώς αντίθετο φαινόμενο της πρόωρης σύγκλισης. Κατά αυτήν, μετά από ένα μεγάλο αριθμό επαναλήψεων, ο πληθυσμός εξακολουθεί να μην συγκλίνει. Το φαινόμενο εμφανίζεται όταν η συνάρτηση καταλληλότητας έχει μικρές κλίσεις, με αποτέλεσμα τα μέγιστα και τα ελάχιστα της να έχουν μικρές διαφορές. Η λύση είναι και πάλι η απεικόνιση της συνάρτησης σε μια νέα, η οποία πρέπει να έχει πιο έντονες διακυμάνσεις.

Τέλος μια εναλλακτική προσέγγιση στο θέμα της εξέλιξης του πληθυσμού είναι η μη ανανέωση ολόκληρου του πληθυσμού σε κάθε γενιά, αλλά ένα μέρους αυτού. Ορίζεται ως *χάσμα γενεών* (*generation gap*) το ποσοστό των χρωμοσωμάτων κάθε γενιάς που ανανεώθηκε, προς το σύνολο των χρωμοσωμάτων. Στους τυπικούς γενετικούς αλγορίθμους ο συντελεστής αυτός ισούται με τη μονάδα. Ωστόσο η νέα τάση είναι η ανανέωση μέρους του πληθυσμού και μάλιστα, στην πιο ακραία μορφή μόνο δύο μελών. Η μέθοδος αυτή ονομάζεται *μέθοδος μερικής ανανέωσης* (*Steady state replacement*).

Η μέθοδος της μερικής ανανέωσης προσεγγίζει πιο πολύ στην πραγματικότητα, αφού εκεί συνυπάρχουν πάντα σε κάποιο βαθμό οι διαφορετικές γενεές. Μάλιστα δίνεται η δυνατότητα στους απογόνους να ανταγωνιστούν τους γονείς τους, επικρατώντας και πάλι ο καλύτερος.

Υπάρχουν δύο θέματα στην προσέγγιση της μερικής ανανέωσης: πώς θα επιλεγούν οι γονείς που θα ζευγαρώσουν και πώς θα επιλεγεί ισάριθμος αριθμός γονέων που θα αποχωρήσουν, ώστε να αφήσουν χώρο για τους απογόνους.

Οι προσεγγίσεις που ακολουθούνται είναι οι ακόλουθες τρεις:

- Επιλογή των γονέων προς αναπαραγωγή με πιθανότητα ανάλογη προς την καταλληλότητα τους και τυχαία επιλογή των γονέων που θα αποχωρήσουν.
- Επιλογή των γονέων προς αναπαραγωγή τυχαία και επιλογή των γονέων που θα αποχωρήσουν με πιθανότητα αντιστρόφως ανάλογη προς την καταλληλότητα τους.
- Επιλογή των γονέων προς αναπαραγωγή με πιθανότητα ανάλογη προς την καταλληλότητα τους και επιλογή των γονέων που θα αποχωρήσουν με πιθανότητα αντιστρόφως ανάλογη προς την καταλληλότητα τους.

### 2.6.2.3 Πληθυσμοί και γενιές

Ο πληθυσμός αποτελείται από μία ομάδα πιθανών λύσεων σε μορφή χρωμοσωμάτων και χρησιμοποιείται από τον γενετικό αλγόριθμο για την εύρεση της βέλτιστης λύσης. Οι επαναλήψεις του γενετικού αλγόριθμου ονομάζονται γενιές.

Η αναπαραγωγή η οποία αποτελείται από την επιλογή (*selection*), την διασταύρωση (*crossover*) και την μετάλλαξη (*mutation*) συνεχίζεται, έως ότου να δημιουργηθεί μία νέα γενιά η οποία θα αντικαταστήσει την προηγούμενη. Εκείνα τα χρωμοσώματα της εκάστοτε γενιάς που έχουν τα καλύτερα χαρακτηριστικά, θα παράγουν περισσότερα αντίγραφα τους στην επόμενη γενιά με αποτέλεσμα την σύγκλιση του πληθυσμού προς την βέλτιστη λύση. Η διαδικασία ολοκληρώνεται όταν τα κριτήρια τερματισμού έχουν ικανοποιηθεί.

Με έναν αποδοτικό γενετικό αλγόριθμο, ο πληθυσμός θα πρέπει μετά από αρκετές επαναλήψεις να συγκλίνει προς το ολικό μέγιστο. Η σύγκλιση αυτή αφορά είτε το καλύτερο στοιχείο του πληθυσμού ή το μέσο όρο αυτού. Ένα γονίδιο θεωρείται ότι έχει συγκλίνει, όταν έχει την ίδια τιμή στο 95% των χρωμοσωμάτων. Ένας πληθυσμός θεωρείται ότι έχει συγκλίνει, όταν όλα τα γονίδια του έχουν συγκλίνει.

### 2.6.2.4 Παιδιά

Οι απόγονοι δημιουργούνται από την εφαρμογή στοχαστικών τελεστών στο επιλεγμένο ζεύγος γονέων, όπως η διασταύρωση και η μετάλλαξη. Η διασταύρωση (*crossover*) αποτελεί τελεστή επανασύνδεσης καθώς αφορά την τυχαία επιλογή του σημείου του χρωμοσώματος όπου θα γίνει η αντιστοίχιση και η συνένωση του γενετικού υλικού των δύο γονέων. Αποτέλεσμα είναι τα δύο παιδιά να φέρουν χαρακτηριστικά και των δύο γονέων. Η διασταύρωση πραγματοποιείται με πιθανότητα  $P_{\text{cross}}$  που κυμαίνεται μεταξύ 0,6 και 0,8.

Η μετάλλαξη αλλάζει την τιμή ενός τυχαία επιλεγμένου γονιδίου του χρωμοσώματος και αποτελεί ένα μηχανισμό ο οποίος διασφαλίζει ότι η επιλογή δεν συγκλίνει σε ένα τοπικό μέγιστο. Επίσης διευρύνει την περιοχή αναζήτησης, καθώς εισάγει γενετικό υλικό το οποίο δεν υπάρχει στην προηγούμενη γενιά, ούτε είναι δυνατόν να προκύψει από τον τελεστή διασταύρωσης. Η μετάλλαξη δεν είναι τόσο σημαντική όσο η διασταύρωση και συνήθως πραγματοποιείται με πιθανότητα  $P_{\text{mutation}}$  της τάξης του 0,05.

### 2.6.2.5 Συνάρτηση Κόστους ή Καταλληλότητας

Η αντικειμενική συνάρτηση η οποία καθορίζει τον στόχο βελτιστοποίησης, ονομάζεται *συνάρτηση κόστους* ή *συνάρτηση καταλληλότητας*. Η συνάρτηση κόστους δέχεται ως είσοδο ένα χρωμόσωμα και επιστρέφει έναν αριθμό που υποδηλώνει πόσο κατάλληλο είναι, δηλαδή αναθέτει μία τιμή σε κάθε χρωμόσωμα του πληθυσμού. Η τιμή αυτή αποτελεί κριτήριο για το πόσο ικανοποιητική είναι η λύση που αντιπροσωπεύει το κάθε χρωμόσωμα στην συγκεκριμένη φάση.

Συγκεκριμένα το πεδίο τιμών της συνάρτησης είναι συνήθως από το 0 έως το 1. Τιμή ίση με 1 σημαίνει ότι το συγκεκριμένο χρωμόσωμα είναι τέλειο, δηλαδή ικανοποιεί όλες τις προϋποθέσεις του προβλήματος και αποτελεί αποδεκτή λύση.

Ανάλογα με το πρόβλημα, η κατασκευή της συνάρτησης καταλληλότητας μπορεί να είναι από απλή, έως εξαιρετικά πολύπλοκη. Η ιδανική συνάρτηση καταλληλότητας θα πρέπει να είναι συνεχής και μονότονη. Ωστόσο αυτό σπάνια συμβαίνει (εάν συνέβαινε, θα αρκούσε μια απλή αναζήτηση αναρρίχησης λόφου για να βρεθεί η βέλτιστη λύση), οπότε αυτό που επιζητείται είναι μια συνάρτηση καταλληλότητας που δεν έχει πολλά τοπικά μέγιστα ή ένα απομονωμένο ολικό μέγιστο.

Ο γενικός κανόνας στην κατασκευή της συνάρτησης καταλληλότητας είναι ότι αυτή πρέπει να αντικατοπτρίζει ρεαλιστικά την αξία του χρωμοσώματος. Ωστόσο η έννοια της αξίας ενός χρωμοσώματος δεν είναι πάντα ιδιαίτερα χρήσιμη. Για παράδειγμα, σε προβλήματα βελτιστοποίησης με περιορισμούς υπάρχουν πολλά χρωμοσώματα, δηλαδή πολλές υποψήφια λύσεις που παραβιάζουν περιορισμούς, άρα η αξία τους είναι μηδενική. Κλασικό παράδειγμα είναι το πρόβλημα της δημιουργίας του ωρολογίου προγράμματος, όπου οι αίθουσες και οι διδάσκοντες πρέπει να κατανομηθούν στα τμήματα, με περιορισμούς όπως ότι μια αίθουσα ή ένας διδάσκων δεν μπορεί να συνδεθεί την ίδια ώρα με δύο διαφορετικά τμήματα. Σε αυτή την περίπτωση η συνάρτηση καταλληλότητας δεν πρέπει να εκφράζει την ποιότητα της λύσης, αλλά το πόσο κοντά αυτή βρίσκεται σε μια έγκυρη λύση. Έτσι υπολογίζεται ο αριθμός των περιορισμών που παραβιάζει η λύση και γίνεται μια εκτίμηση για την προσπάθεια που απαιτείται για να διορθωθεί.

Μια άλλη προσέγγιση που ακολουθείται πολλές φορές είναι αυτή της *προσεγγιστικής συνάρτησης καταλληλότητας* (*approximate fitness function*). Το κρίσιμο θέμα είναι η επιθυμητή ακρίβεια της συνάρτησης καταλληλότητας και το υπολογιστικό κόστος που θεωρείται αποδεκτό για μια συνάρτηση καταλληλότητας που δίνει ενδεχομένως άριστα αποτελέσματα. Εάν λοιπόν υπάρχουν διαθέσιμες δύο συναρτήσεις καταλληλότητας, μια ακριβής και μια προσεγγιστική, όπου όμως η ακριβής έχει δεκαπλάσιο χρόνο υπολογισμού από την προσεγγιστική, ενδεχομένως να συμφέρει η χρήση της προσεγγιστικής, αφού αυτή θα επιτρέψει την πραγματοποίηση

δεκαπλάσιας ποσότητας υπολογισμών (δηλαδή αναπαραγωγών) στον ίδιο χρόνο, σε σχέση με τη χρονοβόρα συνάρτηση.

### 2.6.2.6 Αποτελεσματικότητα και Αποδοτικότητα [65]

Ο Holland το 1975 ήταν ο πρώτος που έδωσε μια τεκμηριωμένη εξήγηση για το πώς και γιατί οι γενετικοί αλγόριθμοι λειτουργούν, με το *θεώρημα των σχημάτων (schema theorem)*. Για την κατανόηση του θεωρήματος θα δοθούν αρχικά μερικοί ορισμοί, θεωρώντας ότι το αλφάβητο στο οποίο παριστάνονται τα γονίδια είναι το δυαδικό  $\{0,1\}$ , τότε ένα σχήμα (*schema*) είναι ένα χρωμόσωμα που δημιουργείται από το αλφάβητο  $\{0, 1, \#\}$ , όπου το σύμβολο  $\#$  υποδηλώνει οποιαδήποτε τιμή. Ένα χρωμόσωμα λέγεται ότι περιέχει ένα σχήμα, εάν οι δύο συμβολοσειρές ταυτίζονται για όλα τα διαφορετικά από το  $\#$  στοιχεία του σχήματος. Για παράδειγμα, το χρωμόσωμα 11001 περιέχει τα σχήματα  $\#10\#\#$ ,  $1\#00\#$ ,  $\#\#001$  και  $1\#\#\#1$ .

Ορίζεται ως *τάξη (order)* ενός σχήματος το πλήθος των συμβόλων που είναι διάφορα από το  $\#$ . Ορίζεται ως *μήκος (length)* ενός σχήματος η απόσταση των δύο πιο εξωτερικών μη- $\#$  συμβόλων. Για τα προαναφερθέντα τέσσερα σχήματα, οι τάξεις τους είναι 2, 3, 3, 2 και τα μήκη τους 2, 4, 3, 5 αντίστοιχα.

Ο Holland θεωρεί ότι η υψηλή τιμή καταλληλότητας ενός χρωμοσώματος οφείλεται στο γεγονός ότι αυτό περιέχει «καλά» σχήματα. Επιλέγοντας για αναπαραγωγή τα χρωμοσώματα με υψηλή τιμή καταλληλότητας αυξάνουμε την πιθανότητα επικράτησης των καλών σχημάτων στον πληθυσμό. Το θεώρημα των σχημάτων καθορίζει ότι αποδίδοντας πιθανότητα αναπαραγωγής στα χρωμοσώματα ανάλογη προς την καταλληλότητά τους, τα «καλά» σχήματα τυγχάνουν εκθετικά αυξανόμενου αριθμού αναπαραγωγών στις επόμενες γενεές. Επίσης καθορίζει ότι ο αριθμός των σχημάτων τα οποία μεταφέρονται από γενεά σε γενεά είναι της τάξης του  $N^3$ , όπου  $N$  το μέγεθος του πληθυσμού. Η τελευταία ιδιότητα αναφέρεται και ως *έμμεσος παραλληλισμός (implicit parallelism)* και θεωρείται ως η εξήγηση για την αποτελεσματικότητα των γενετικών αλγορίθμων στην επίλυση προβλημάτων.

Το βασικότερο ερώτημα που έχει απασχολήσει την επιστημονική κοινότητα σε σχέση με τους γενετικούς αλγορίθμους είναι το πώς πρέπει να κωδικοποιηθεί ένα πρόβλημα, ώστε η αντιμετώπισή του να είναι αποδοτική. Μια καλή κωδικοποίηση πρέπει να έχει τις παρακάτω ιδιότητες:

- Τα σχετικά γονίδια να είναι γειτονικά στα χρωμοσώματα.
- Να υπάρχει μικρή αλληλεπίδραση μεταξύ των γονιδίων.

*Σχετικά γονίδια* είναι αυτά τα οποία συνδυασμένα επηρεάζουν τη συνάρτηση καταλληλότητας. Αυτό σημαίνει ότι η συνάρτηση καταλληλότητας δεν εξαρτάται από κάθε ένα γονίδιο ξεχωριστά, αλλά από συνδυασμούς αυτών. Εάν όλα τα γονίδια ήταν ανεξάρτητα μεταξύ τους (η ιδανική περίπτωση), τότε η επίλυση ενός προβλήματος θα αναγόταν στην επίλυση  $N$  ανεξάρτητων υπό-προβλημάτων, όπου  $N$  ο αριθμός των γονιδίων. Στην πράξη, αυτό που συμβαίνει είναι ότι υπάρχει μερική αλληλεπίδραση μεταξύ των γονιδίων. Σε αυτή την περίπτωση, μια αποδοτική κωδικοποίηση θα είχε τα σχετικά γονίδια κοντά, δημιουργώντας καλά σχήματα όσο το δυνατόν μικρότερης τάξης και μικρότερου μήκους. Η δημιουργία καλών σχημάτων μικρής τάξης και μήκους

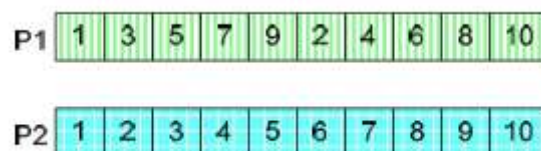
αυξάνει την πιθανότητα επιβίωσης τους κατά τη φάση της διασταύρωσης στη διαδικασία της αναπαραγωγής.

Σε περίπτωση που υπάρχει ισχυρή αλληλεπίδραση μεταξύ των γονιδίων μιας κωδικοποίησης, με αποτέλεσμα να μην είναι δυνατή η τοποθέτηση σχετικών γονιδίων σε κοντινές θέσεις και η δημιουργία μικρών σχημάτων, τότε η απόδοση του γενετικού αλγορίθμου είναι αμφίβολη. Σε αυτήν την περίπτωση πρέπει είτε να επιχειρηθεί εναλλακτική κωδικοποίηση, επιλέγοντας διαφορετικές μεταβλητές για την αναπαράσταση του φαινοτύπου ή να εγκαταλειφθεί η ιδέα της χρήσης γενετικών αλγορίθμων για την επίλυση του προβλήματος.

### 2.6.3 Τεχνικές Αναπαραγωγής

Οι τεχνικές αναπαραγωγής είναι η διασταύρωση και η μετάλλαξη που χρησιμοποιούνται για να μεταβληθεί το χρωμόσωμα.

Τα στοιχεία για την εξήγηση της διαδικασίας αφορούν το παραδείγματα υπολογισμού του ταξιδιού του πωλητή, όπου το χρωμόσωμα είναι ρυθμισμένο για δέκα πόλεις. Οι αρχικοί γονείς φαίνονται στην εικόνα 25.



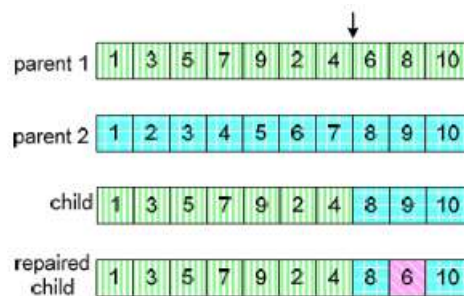
Εικόνα 25 : Χρωμόσωμα για βελτιστοποίηση ταξιδιού 10 πόλεων [60]

#### 2.6.3.1 Διασταύρωση

Διασταύρωση είναι ή μέθοδος που χρησιμοποιείται για να χωρίσει και να «ανασύνθεση» τα χρωμοσώματα από δύο ή περισσότερους γονείς σε έναν απόγονο. Οι κύριοι αλγόριθμοι διασταύρωσης παρουσιάζονται στην συνέχεια.

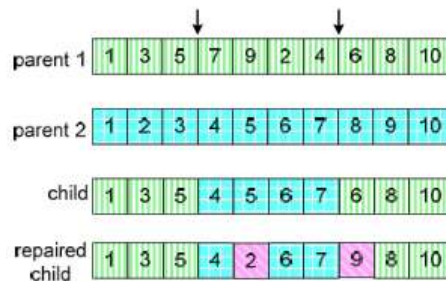
Στην **διασταύρωση μονού σημείου** επιλέγεται ένα σημείο τυχαία στο χρωμόσωμα του ενός γονέα, χωρίζοντας το χρωμόσωμα σε 2 μέρη. Οι παράμετροι του χρωμοσώματος μετά από αυτό το σημείο γίνεται αντιμετάθεση με τις παραμέτρους του χρωμοσώματος του άλλου γονέα. Αυτή η διαδικασία δημιουργεί έναν απόγονο με τον συνδυασμό των τμημάτων με χρωμοσωμάτων των δύο γονέων, όπως φαίνεται στην εικόνα 26.

Στο παράδειγμα σε αυτή την περίπτωση το προκύπτουν χρωμόσωμα (παιδί) δεν είναι έγκυρο, καθώς η ίδια πόλη (9) παρουσιάζεται δύο φορές μέσα σε αυτό. Αυτό το χρωμόσωμα θα πρέπει να επισκευαστεί (επισκευασμένο παιδί), έτσι ώστε να γίνει έγκυρο. Το μειονέκτημα στην διασταύρωση μονού σημείου είναι ότι τόσο η αρχή και το τέλος των χρωμοσωμάτων του ενός γονέα μπορεί να περιέχει καλές ιδιότητες, αλλά δεν μπορούν τόσο να περάσουν σε έναν απόγονο γιατί μόνο το πρώτο ή το τελευταίο τμήμα του κάθε γονέα θα μεταφερθούν στους απογόνους.



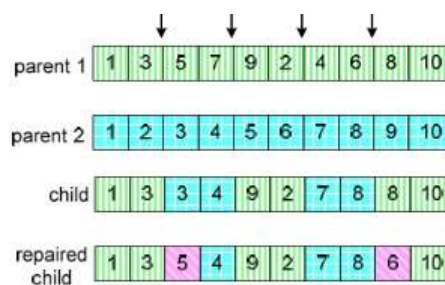
Εικόνα 26 : Διασταύρωση μονού σημείου [60]

Στη **διασταύρωση διπλού σημείου** επιλέγονται τυχαία δύο σημεία στο χρωμόσωμα και γίνεται αντιμετάθεση μεταξύ των χρωμοσώματα των δύο γονέων, όπως φαίνεται στην εικόνα 27. Αυτό θεωρείται καλύτερη τεχνική από την διασταύρωση μονού σημείου, καθώς υπερνικά το πρόβλημα που έχει το crossover μονού σημείου στο να μην είναι σε θέση να μεταφέρει το πρώτο και το τελευταίο τμήμα του χρωμοσώματος.



Εικόνα 27 : Διασταύρωση διπλού σημείου [60]

Στην **διασταύρωση πολλαπλών σημείων** επιλέγονται τυχαία πολλαπλά σημεία και ανταλλάσσονται τα χρωμοσώματα μεταξύ αυτών των σημείων, όπως φαίνεται στην εικόνα 28. Το πλεονέκτημα αυτής της τεχνικής είναι ότι ο χώρος αναζήτησης διερευνάται λεπτομερέστερα. Ωστόσο, ακολουθίες ή καλά δομικά στοιχεία εντός του χρωμοσώματος μπορεί να καταστραφούν όταν αυτό χωρίζεται.



Εικόνα 28: Διασταύρωση πολλαπλών σημείων [60]

Στην **ομοιόμορφη διασταύρωση** κάθε παράμετρος εντός των χρωμοσωμάτων των γονέων θα έχει μια πιθανότητα 50% να περάσει στους απογόνους, όπως φαίνεται στην εικόνα 29. Αν και αυτό μπορεί να προκαλέσει αναστάτωση, το χρωμόσωμα από ομοιόμορφη διασταύρωση έχει μια ισχυρή προκατάληψη εξερεύνησης του χώρου αναζήτησης. Αυτή η μέθοδος είναι ιδιαίτερα χρήσιμη, εάν το μέγεθος του πληθυσμού είναι μικρό, δεδομένου ότι παράγει μια υψηλή διακύμανση μεταξύ των γονέων και των απογόνων τους, λόγω του ότι το χρωμόσωμα μεταβάλλεται σε όλη την πλήρη γκάμα.



Εικόνα 29: Ομοιόμορφη Διασταύρωση [60]

Εκτός των αναφερθέντων περιπτώσεων στην βιβλιογραφία έχουν αναφερθεί και οι ακόλουθοι μέθοδοι διασταύρωσης:

α) **αριθμητική**, όπου ένας αριθμητικός δείκτης διασταύρωσης εφαρμόζεται στους δύο γονείς να σχηματιστεί ένας απόγονος,

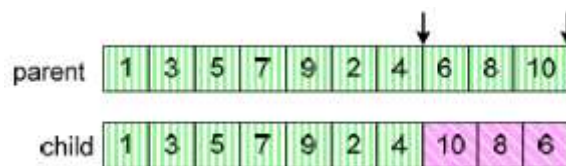
β) **ευρηματική με πολλούς γονείς** χρησιμοποιώντας διαγώνια διασταύρωση και διασταύρωση με σάρωση [18], και

γ) πολλούς γονείς με **χρήση πολυωνύμου και κανονική λογαριθμική κατανομή** [19].

### 2.6.3.2 Μετάλλαξη

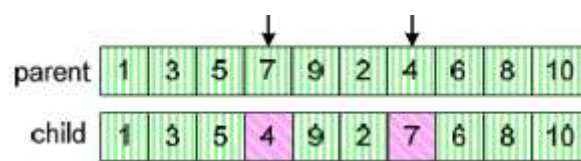
Μετάλλαξη είναι μια μέθοδος αναπαραγωγής που λειτουργεί σε ένα ενιαίο χρωμόσωμα. Είναι μια τυχαία διαδικασία και έχει μια χαμηλή πιθανότητα που συμβαίνει που συνήθως είναι μεταξύ 0,1 έως 2 τοις εκατό. Σκοπός της είναι να αυξήσει την ποικιλομορφία του πληθυσμού. Υπάρχουν πολλές τεχνικές μετάλλαξης, μερικές από αυτές περιγράφονται παρακάτω.

Η **μετάλλαξη αναστροφής** όπου δύο τυχαία σημεία επιλέγονται στο χρωμόσωμα του γονέα και οι παράμετροι μεταξύ των επιλεγμένων σημείων είναι αναστρέφονται. Αυτό μμείται ένα φυσικό φαινόμενο βιολογικής μετάλλαξης (εικόνα 30).



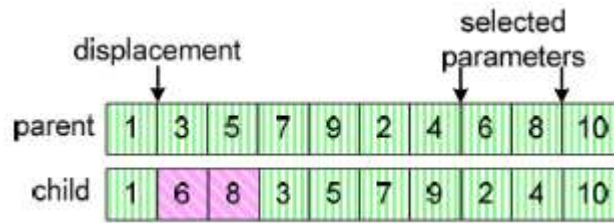
Εικόνα 30 : Μετάλλαξη αναστροφής [60]

Στην **μετάλλαξη αντιμετάθεσης** επιλέγονται δύο τυχαία σημεία στο χρωμόσωμα του γονέα και οι παράμετροι σε αυτό το σημείο αντιμετατίθενται (εικόνα 31).



Εικόνα 31: Μετάλλαξη Αντιμετάθεσης [60]

Στην **μετάλλαξη μετατόπισης** επιλέγονται δύο τυχαία σημεία και οι παράμετροι μεταξύ αυτών των σημείων μετακινούνται αυτούσιες σε ένα άλλο τυχαία επιλεγμένο τμήμα του χρωμοσώματος (εικόνα 32).



Εικόνα 32: Μετάλλαξη μετατόπισης [60]

Σε γενικές γραμμές έχει διαπιστωθεί ότι το ποσοστό μεταλλαγών πρέπει να είναι υψηλό στην αρχή αλλά στη συνέχεια θα πρέπει να μειώνεται καθώς η εξέλιξη πλησιάζει σε ένα υψηλό βαθμό καταλληλότητας. Ο Muhlenbein [20] μελέτησε χρωμοσώματα αποτελούμενα από bit και διαπίστωσε ότι όσα περισσότερα bit εντός της συμβολοσειράς του χρωμοσώματος διορθώνονταν τότε η πιθανότητα μιας μετάλλαξης να δημιουργήσει καλύτερους απογόνους μειώθηκαν. Αντιστρόφως, όταν περισσότερα bit εντός της συμβολοσειράς χρωμοσώματος διορθώνονταν αυξάνονταν η πιθανότητα της διασταύρωσης να παράγει καλύτερους απογόνους. Διαπιστώθηκε ότι εάν χρησιμοποιηθεί ένα ποσοστό συνεχής μετάλλαξης, τότε το ποσοστό βέλτιστης μετάλλαξης είναι το  $1/L$  όπου  $L$  το μήκος της συμβολοσειράς σε bit. Η χρήση του τελεστή μετάλλαξης γίνεται σε συνδυασμό με τον τελεστή διασταύρωσης. Ωστόσο, υπάρχουν αλγόριθμοι που χρησιμοποιούν μόνο τον τελεστή μετάλλαξης.

#### 2.6.4 Γενετική επιλογή

Γενετική επιλογή ονομάζουμε την μέθοδος που χρησιμοποιείται για να επιλέξουμε ποιοι απόγονοι θα διατηρηθούν για να γίνουν γονείς στην επόμενη γενιά. Σκοπός της μεθόδου αυτής είναι να μετατοπίσει τον πληθυσμό σε ένα υψηλότερο επίπεδο καταλληλότητας. Η γενετική επιλογή χρησιμοποιεί την τιμή της συνάρτησης καταλληλότητας κάθε απογόνου για να προσδιορίσει εάν θα διατηρηθεί. Κάθε απόγονος αξιολογείται και του δίνεται μια τιμή καταλληλότητας, η οποία χρησιμοποιείται από τη διαδικασία επιλογής.

**Η πίεση επιλογής είναι μία σημαντική παράμετρος στην γενετική επιλογή.** Η μέθοδος επιλογής έχει μια υψηλή πίεση επιλογής εφόσον μειώνει την διαφορά της τιμής καταλληλότητας μεταξύ των ατόμων ή μία χαμηλή πίεση επιλογής αν επιτρέπει σε πολλά και διαφορετικά άτομα να επιβιώσουν. Μια επιλογή με χαμηλή πίεση θα έχει αργό ρυθμό σύγκλισης προς την βέλτιστη λύση και ενδεχομένως να κολλήσει γύρω από κάποιο σημείο, ενώ μια πίεση επιλογής πολύ υψηλή μπορεί να παγιδευτεί σε τοπικά μέγιστα λόγω απώλειας της ποικιλομορφίας. Η επιλογή του ποια μέθοδος επιλογής θα χρησιμοποιηθεί εξαρτάται από το είδος του προβλήματος που καλείται να επιλύσει. Για παράδειγμα, αναλογική επιλογή, γραμμική κατάταξη και επιλογή τουρνουά έχουν πίεση συγκριτικής επιλογής που αυξάνει με τη σειρά που παρατίθενται [21].

Η γενετική επιλογή εξαρτάται από μια σειρά από μεταβλητές όπως η πίεση επιλογής, η απώλεια της πολυμορφίας, διακύμανση επιλογής, επιλογή έντασης, και το χρόνο της διαδικασίας.

**Επιλεκτική πίεση:** είναι ο λόγος της πιθανότητας του καλύτερου ατόμου να επιλεγεί, προς την μέση πιθανότητα επιλογής όλων των ατόμων εντός του πληθυσμού. Δηλώνει



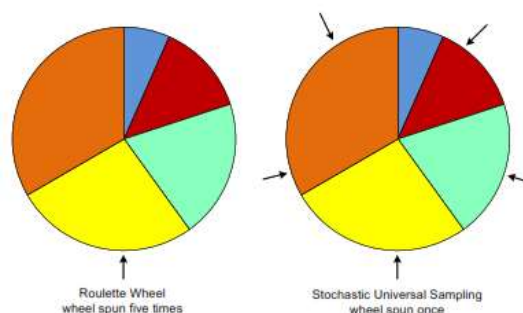
την ποικιλομορφία του πληθυσμού μετά από τη διαδικασία επιλογής. Εάν η πίεση επιλογής είναι πολύ μικρή, τότε θα επέλθει ελάχιστη ή καθόλου βελτίωση στην συνάρτηση καταλληλότητας του πληθυσμού. Εάν η επιλεκτική πίεση είναι πολύ υψηλή, είναι δυνατόν να πάρει πρόωρη σύγκλιση προς ένα σημείο, με τα άτομα που επικεντρώνεται σε αυτό το οποίο θα είναι ένα τοπικό μέγιστο.

- Προκατάληψη: είναι η πιθανότητα ενός ατόμου που έχει επιτύχει πολύ υψηλή τιμή καταλληλότητας να διατηρηθεί μετά από την επιλογή.
- Απώλεια της ποικιλομορφίας: αντανακλά τη μείωση της ποικιλομορφίας των ατόμων εντός του πληθυσμού λόγω της διαδικασίας επιλογής.
- Ένταση επιλογής: ορίζει την αναμενόμενη μέση τιμή καταλληλότητας του πληθυσμού, αφού έχει εφαρμοστεί η διαδικασία επιλογής. Ένα σημάδι που μας δείχνει την απώλεια στη διαφορετικότητα σε έναν πληθυσμό είναι όταν ο πληθυσμός έχει μια υψηλή μέση τιμή καταλληλότητας.
- Επιλογή διακύμανση: εκφράζει την μεταβολή της κατανομής της τιμής καταλληλότητας του πληθυσμού σε σύγκριση με την κανονικοποιημένη Gaussian κατανομή μετά τη διαδικασία επιλογής.
- Χρόνος διαδικασίας : ο χρόνος που απαιτείται από τον πλήρη πληθυσμό να αναπαραχθεί με τα καλύτερα άτομα.

#### 2.6.4.1 Μέθοδοι επιλογής

Υπάρχουν πολλές και διαφορετικές μέθοδοι επιλογής, κάθε μια με τα δικά της πλεονεκτήματα και μειονεκτήματα τόσο για την πίεση επιλογής, ή την διατήρηση της πολυμορφίας. Μερικά από αυτά τα συστήματα θα τα παρουσιάσουμε περιγραφικά παρακάτω:

Η **Αναλογική επιλογή ή Επιλογή Ρουλέτας** είναι μια μέθοδος όπου η πιθανότητα ενός ατόμου για να επιλεγεί εξαρτάται από την τιμή καταλληλότητας του προς την μέση τιμή καταλληλότητας όλων των ατόμων. Επομένως, ένα άτομο με υψηλή τιμή καταλληλότητας έχει υψηλότερη πιθανότητα να επιλεγεί. Αυτή η μέθοδος της επιλογής μπορεί να απεικονιστεί με την περιστροφή ενός τροχού ρουλέτας ή έναν περιστρεφόμενο καθολικό δειγματοληπτικό τροχό, όπως φαίνεται στην εικόνα 33. Παρά το γεγονός ότι τα άτομα με υψηλότερη τιμή καταλληλότητας θα έχουν περισσότερους αριθμούς στον τροχό της ρουλέτας και έτσι θα έχουν περισσότερες πιθανότητες να επιλεγούν, άτομα με μικρότερη τιμή καταλληλότητας εξακολουθούν να διατηρούν μια πιθανότητα να επιλεγούν.



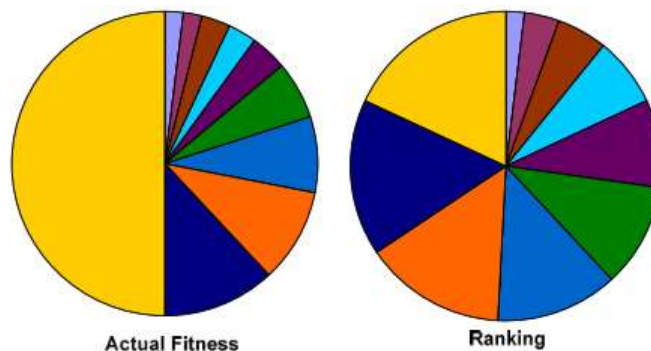
Εικόνα 33 : Αναλογική μέθοδος επιλογής ή Επιλογή ρουλέτας [60]

Με την τεχνική της δειγματοληψίας με τον τροχό της ρουλέτας, ο τροχός περιστρέφεται επανειλημμένα, σε κάθε περιστροφή επιλέγεται και ένα άτομο, μέχρι να επιτευχθεί ο απαιτούμενος αριθμός των απογόνων. Όταν έχουμε ένα τροχό στοχαστικής καθολικής δειγματοληψίας απαιτεί μόνο ένα γύρισμα του τροχού, με τα πολλαπλά σημεία επιλογής να είναι ισοκατανεμημένα και το σημείο εκκίνησης να έχει επιλεγεί τυχαία. Η μέθοδος αναλογικής επιλογής διατηρεί την ποικιλομορφία του πληθυσμού για αρκετές γενεές και παρουσιάζει μια γρήγορη αύξηση της τιμής καταλληλότητας των καλύτερων ατόμων παρότι η μέση τιμή της καταλληλότητας παρουσιάζει μια πιο αργή εξέλιξη.

Η **επιλογή βάση της βαθμολογίας** χρησιμοποιεί μια διαδικασία παρόμοια με τον τροχό της ρουλέτας. Ωστόσο, το κάθε άτομο παίρνει μια εκτίμηση καταλληλότητας που εξαρτάται από το βαθμό εντός του πληθυσμού, παρά στην τιμή καταλληλότητας. Μετά την κατάταξη, η τιμή καταλληλότητας θα κυμαίνεται από το ένα έως το μέγεθος του πληθυσμού. Όλα τα άτομα θα έχουν μια διαφορετική κατάταξη, ακόμη και αν έχουν την ίδια τιμή καταλληλότητας, ως εκ τούτου, κάθε άτομο θα έχει μια διαφορετική πιθανότητα επιλογής.

Αυτό εισάγει μια ομοιόμορφη κλιμάκωση, και ελέγχει την πίεση επιλογής, δηλαδή η πίεση επιλογής μειώνεται όταν η διακύμανση της τιμής καταλληλότητας είναι υψηλή και αυξάνεται όταν η διακύμανση της τιμής καταλληλότητας είναι χαμηλή, αποτρέποντας κατά συνέπεια ένα άτομο να αποκτήσει δεσπόζουσα θέση στην διαδικασία επιλογής.

Η **επιλογή με βάση το βαθμό** μπορεί να είναι γραμμική ή εκθετική [22]. Επιλογή με βάση το βαθμό παρέχει έναν απλό τρόπο για τον έλεγχο της πίεσης επιλογής. Στα γραφήματα πίτας στην εικόνα 34, ένα άτομο δεσπόζει με τιμή καταλληλότητας στο 50% · Ωστόσο, μετά την κατάταξη, το άτομο αυτό έχει μόνο μια ελαφρώς καλύτερη πιθανότητα να επιλεγεί από το δεύτερο ή τον τρίτο στην ατομική κατάταξη.

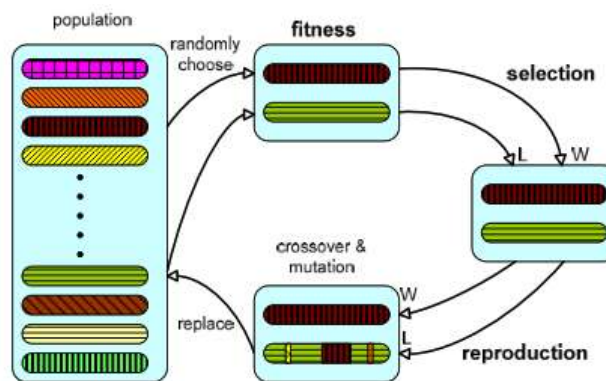


Εικόνα 34 :Επιλογή βάση βαθμολογίας [60]

Στην **επιλογή τουρνουά** η διαδικασία χωρίζει τα άτομα εντός του πληθυσμού σε υποομάδες και μέσα σε κάθε γενιά διατηρείται το καλύτερο άτομο για κάθε υποομάδα. Οι υποομάδες μπορούν να ποικίλουν σε μέγεθος από δύο ή περισσότερα άτομα. Όσο μεγαλύτερο το μέγεθος της υποομάδας, τόσο μεγαλύτερη και η πίεση επιλογής. Με τη μέθοδο αυτή, γενικά επιλέγονται τα άτομα με την υψηλότερη τιμή καταλληλότητας. Ωστόσο, ορισμένα άτομα με χαμηλή τιμή καταλληλότητας επίσης διατηρούνται για να μπορέσει να διατηρηθεί η πολυμορφία. Η επιλογή τουρνουά είναι κατάλληλη τεχνική

για την εξέλιξη του υλικού καθώς είναι αποδοτικό, εύκολη να εφαρμοστεί και ικανό για παραλληλισμό.

Η **μικροβιακή επιλογή γενετικών αλγόριθμων** [23] (**Microbial Genetic Algorithms**) είναι μια εξειδικευμένη μορφή της επιλογής με τουρνουά, η οποία επιδιώκει να βελτιώσει την αποτελεσματικότητα της επιλογής του τυπικού τουρνουά. Αντί για δημιουργία και αξιολόγηση ενός νέου πληθυσμού σε ένα βήμα, από τον υπάρχων πληθυσμό επιλέγονται τυχαία δύο άτομα. Αυτά τα δύο άτομα χρησιμοποιούνται για τη δημιουργία ενός νέου απογόνου ο οποίος θα αντικαταστήσει έναν εκ των γονέων που τουλάχιστον του ταιριάζει εντός του πληθυσμού, όπως φαίνεται στην εικόνα 35.



Εικόνα 35 : Μικροβιακή Επιλογή [60]

Αυτός ο αλγόριθμος είναι χρήσιμος για την εξέλιξη υλικού, καθώς είναι απλός στην εφαρμογή.

Ο **ελιτισμός** επιτρέπει σ' έναν περιορισμένο αριθμό από τα ισχυρότερα άτομα να περάσουν στην επόμενη γενιά χωρίς τροποποίηση από τις ρουτίνες διασταύρωσης και μετάλλαξης. Αυτό εξασφαλίζει ότι το άτομο με την καλύτερη τιμή καταλληλότητας δεν χάνεται ποτέ μέσα στις διαδικασίες διασταύρωσης και μετάλλαξης που σε κάποιες περιπτώσεις οδηγούν στη καταστροφή κάποιων από τα καλύτερα άτομα. Ωστόσο, ένας υψηλό αριθμός ελίτ ατόμων θα προκαλέσει απώλεια στην ποικιλομορφία του πληθυσμού. Ο Ελιτισμός χρησιμοποιείται συχνά σε συνδυασμό με άλλα συστήματα επιλογής.

Η **κλιμακωτή επιλογή** αλλάζει την πίεση επιλογής του πληθυσμού, καθώς εξελίσσεται. Αρχικά, η πίεση επιλογής είναι χαμηλή, επιτρέποντας σε ένα ευρύ φάσμα ατόμων να επιβιώσουν στην επόμενη γενιά, αλλά όσο η μέση τιμή καταλληλότητας αυξάνει, η πίεση επιλογής αυξάνει και έτσι μειώνεται και ο αριθμός των ατόμων που θα επιβιώσουν. Η κλιμακωτή επιλογή αρχικά έχει μια χαμηλή πίεση επιλογή αλλά όσο η τιμή καταλληλότητας των ατόμων φτάσει σε προχωρημένο επίπεδο, η διαδικασία επιλογής γίνεται περισσότερο επιλεκτική. Αυτή η μέθοδος μπορεί να χρησιμοποιηθεί ως μέρος των αναλογικών συστημάτων επιλογής καταλληλότητας που είδαμε προηγουμένως, αλλάζοντας την πίεση επιλογής από ένα σταθερό σε ένα μεταβλητό ρυθμό. Αν η πίεση επιλογής είναι πάρα πολύ υψηλή κατά την εκκίνηση, η σύγκλιση θα είναι πιο γρήγορη, όμως χάνεται γρήγορα η ποικιλομορφία του πληθυσμού και η διαδικασία μπορεί να παγιδευτεί σε ένα τοπικό μέγιστο. Πολλές μέθοδοι είναι σε θέση

να καθορίσουν την κλιμάκωση: γραμμική κλιμάκωση, σιγμοειδής κλιμάκωση, κλιμάκωση παραθύρου [24] και σχετική καταλληλότητα [25].

Η **γενεαλογική επιλογή** μιμείται μια πραγματική κατάσταση όπου οι γονείς δεν διατηρούνται και οι καλύτεροι απόγονοι μετακαλούνται στην επόμενη γενιά.

**Σταθερή κατάσταση επιλογής**, όπου ένα μεγάλο ποσοστό από τον υπάρχοντα πληθυσμό διατηρείται μετά τη διαδικασία επιλογής. Μόνο μερικοί από τους νέους απογόνους (οι καλύτεροι) διατηρούνται για να αντικαταστήσουν έναν μικρό αριθμό από τους χειρότερους γονείς του υπάρχοντα πληθυσμού.

Η **ιεραρχική επιλογή** γίνεται πολλές φορές μέσα σε κάθε γενεά, οι πρώτες επιλογές είναι απλές και γρήγορες, και εξαλείφονται πολλοί από τους ασθενέστερους απογόνους. Η διαδικασία επιλογής στην πορεία γίνεται πιο περίπλοκη, καθώς ο αριθμός των απογόνων μειώνεται. Αυτό επιταχύνει τη συνολική διαδικασία επιλογής όπως η διαδικασία αρχικής επιλογής παίρνει λιγότερο χρόνο.

**Επιλογή με Ομοιόμορφη κατανομή καταλληλότητας** [26, 27] έχει μια πίεση επιλογής που ευνοεί τα άτομα που βάση της τιμής καταλληλότητας εντάσσονται στην περιφέρεια, αντί για απευθείας επιλογή ενός ατόμου με υψηλή τιμή καταλληλότητας. Η μέθοδος επιλογής βρίσκει την υψηλότερη τιμή καταλληλότητας ( $f_{max}$ ) και την χαμηλότερη τιμή καταλληλότητας ( $f_{min}$ ) στον πληθυσμό. Γίνεται ομοιόμορφη κατανομή μεταξύ των τιμών του  $f_{max}$  και  $f_{min}$ , επιλέγεται μια τυχαία τιμή ανάμεσα στις τιμές του  $f_{max}$  και  $f_{min}$  και το άτομο του οποίου η τιμή καταλληλότητας είναι πλησιέστερη στην τυχαία τιμή, επιλέγεται. Επιλογή με ομοιόμορφη κατανομή της καταλληλότητας θα έχει υψηλότερη πίεση επιλογής σε ένα πληθυσμό με χαμηλή μέση τιμή καταλληλότητας, και μια χαμηλότερη πίεση επιλογής σε ένα πληθυσμό με υψηλή μέση τιμή καταλληλότητας. Η επιλογή αυτή ταιριάζει σε ένα πληθυσμό με λίγα άτομα, ωστόσο, απαιτείται μόνο ένα άτομο με καλή τιμή καταλληλότητας και υπάρχει και μεγάλη ποικιλομορφία που διατηρείται στον πληθυσμό.

**Επιλογή με Μοντέλο Νησιού** είναι κατάλληλο για εφαρμογές που τρέχουν παράλληλα και της συναντάμε συνήθως σε FPGA όπου μπορούν να τρέχουν ταυτόχρονα πολλαπλοί γενετικοί αλγόριθμοι. Χρησιμοποιεί υποσύνολα πληθυσμού σε ξεχωριστά νησιά, τα οποία εξελίσσονται παράλληλα, σε παράλληλες μηχανές γενετικών αλγορίθμων. Περιοδική μετεγκατάσταση παρουσιάζεται όταν άτομα ανταλλάσσονται μεταξύ των υπό-πληθυσμών από τα διάφορα νησιά. Έτσι, εάν ο συνολικός πληθυσμός ήταν  $T_{total}$ , και ο αριθμός των νησιών ήταν  $I_{island}$ , τότε ο υπό-πληθυσμός του κάθε νησιού είναι  $I_{subpopulation} = T_{total} / I_{island}$ , όπου ο συνολικός πληθυσμός διαιρείται εξίσου μεταξύ των νήσων.

Δύο σημαντικές παράμετροι του μοντέλου νησιού είναι το **μέγεθος μετανάστευσης**, το οποίο είναι ο αριθμός των ατόμων που θα μεταφερθούν όταν παρουσιαστεί η μετανάστευση, και το **διάστημα μετανάστευσης**, το οποίο είναι ο αριθμός των γενεών που συμβαίνουν μεταξύ μετανάστευσης [28].

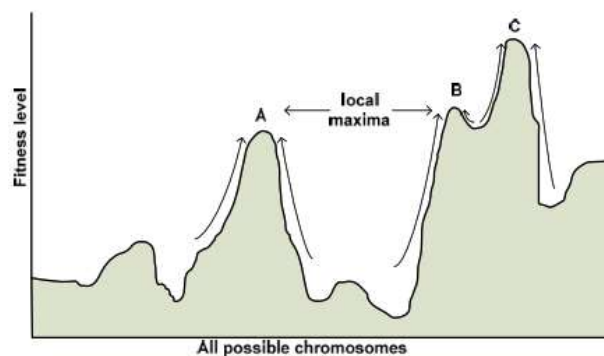
### 2.6.5 Πεδίο Καταλληλότητας

Το τοπίο καταλληλότητας είναι μια γραφική παράσταση που αποτυπώνεται η τιμή καταλληλότητας που έχουν όλοι οι πιθανοί συνδυασμοί των χρωμοσωμάτων. Αυτό

είναι ένα διδιάστατο γράφημα με την καταλληλότητα (fitness) να είναι στον άξονα y και όλα τα πιθανά χρωμοσώματα είναι στον άξονα x, όπως φαίνεται στην εικόνα 29. Το μέγεθος του πεδίου καταλληλότητας (fitness) είναι ανάλογο με το μέγεθος του χρωμοσώματος, με ένα μεγαλύτερο χρωμόσωμα να έχει ένα μεγαλύτερο τοπίο καταλληλότητας.

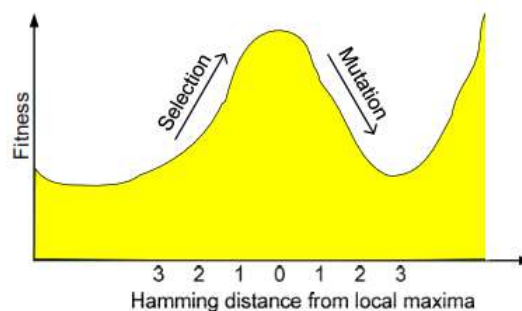
Πιθανές λύσεις ή επιτυχημένα χρωμοσώματα, είναι γνωστά ως μέγιστα (maxima) στο γράφημα. Τα περισσότερα προβλήματα θα έχουν πολλαπλά μέγιστα (maxima). Η καταλληλότητα των χρωμοσωμάτων μεταξύ των μεγίστων (maxima) είναι μη γραμμική, οπότε το πεδίο καταλληλότητας γίνεται μια σειρά από κορυφές και κοιλάτες. Καθώς εξελίσσονται τα χρωμοσώματα, θα ανεβάσουν το πεδίο καταλληλότητας προς ένα μέγιστο.

Ωστόσο, συχνά κινείται σε κορυφές που είναι γνωστές ως τοπικά μέγιστα (maxima), όπου βρίσκεται μια βελτίωση της τιμής καταλληλότητας, αλλά δεν είναι η τελική λύση. Η εξέλιξη ενδέχεται να τερματιστεί σε τοπικά μέγιστα, εκτός αν η διαδικασία επιλογής και αναπαραγωγής μπορεί να μετακινηθεί το χρωμόσωμα σε παρακείμενη κοιλάτητα και μέχρι την επόμενη κορυφή.



Εικόνα 36 : Πεδίο καταλληλότητας [60]

Στην εικόνα 36, σημεία A και B είναι τοπικά μέγιστα (maxima) με το C είναι η βέλτιστη λύση. Τα βέλη υποδεικνύουν την κατεύθυνση της εξελικτικής διαδικασίας, καθώς ο πληθυσμός μετακινείται από μια πολύ χαμηλή τιμή καταλληλότητας προς μια κορυφή. Προκειμένου για ένα χρωμόσωμα να προχωρήσει πέρα από τα τοπικά μέγιστα στο A, απαιτείται μια μεγάλη ποικιλομορφία του πληθυσμού και εξαρτάται από την διαδικασία επιλογής που χρησιμοποιείται και σε ένα ρυθμό μετάλλαξης αρκετά υψηλό που θα επιτρέψει στα χρωμοσώματα να κινούνται κάτω από τα τοπικά ακρότατα και μέχρι την επόμενη κορυφή.



Εικόνα 37 : Απόσταση επενέργειας από το τοπικό μέγιστο. [60]

Η εικόνα 37 παρουσιάζει πώς η διαδικασία επιλογής διατηρεί τα χρωμοσώματα με την υψηλότερη τιμή καταλληλότητας, όταν η συνάρτηση καταλληλότητας κινείται επάνω στο πεδίο καταλληλότητας. Η λειτουργία της μετάλλαξης σκοπό έχει να αποτρέψει τα χρωμοσώματα να παγιδευτούν σε τοπικά μέγιστα. Το επίπεδο της μετάλλαξης που απαιτείται για να κινηθούν τα χρωμοσώματα μακριά από τα τοπικά maxima μπορεί να υπολογιστεί από την απόσταση επενέργειας (hamming). Η απόσταση επενέργειας (hamming) περιγράφει τον ελάχιστο αριθμό των αλλαγών που απαιτούνται εντός του χρωμοσώματος για να μετακινηθεί από την κορυφή του τοπικού μέγιστου σε μια κοιλάτητα διορθώσεως.

## 2.7 Η συνάρτηση καταλληλότητας

Μια επιτυχημένη διαδικασία εξέλιξης του ελεγκτή ενός αυτόνομου ρομπότ στηρίζεται στην επιτυχημένη σύνταξη της κατάλληλης συνάρτησης καταλληλότητας η οποία θα είναι με τέτοιο τρόπο δομημένη που θα συλλέγει της επιτυχημένες συμπεριφορές του οχήματος χωρίς όμως την απαίτηση να γίνεται μεγάλη ανάλυση των συμπεριφορών αυτών.

Η συνάρτηση καταλληλότητας είναι στην καρδιά της διαδικασίας της εξελικτικής διαδικασίας. Είναι υπεύθυνη να αποφασίσει ποια λύση (εκλεκτής στην προκειμένη περίπτωση) ανάμεσα στις προτεινόμενες είναι η καλύτερη για το συγκεκριμένο πρόβλημα. Όμως στην πραγματικότητα όταν γίνεται μια εξελικτική διαδικασία κατά την οποία γίνεται προσπάθεια να εξελιχτεί ένας εκλεκτής για μια ποιο περιπλοκή εργασία – διεργασία τότε η συνάρτηση καταλληλότητας υπάρχει περίπτωση να λειτουργεί ως περιοριστικός παράγοντας στην εύρεση ενός ποιοτικά καλού ελεγκτή.

Ο περιορισμός γίνεται εμφανής με ένα δισταγμό στην συνάρτηση καταλληλότητας στις επόμενες γενεές όπου αδυνατεί να βελτιώσει ή τουλάχιστον μεταβάλει την τιμή της ανάμεσα στα διαφορετικά άτομα του πληθυσμού που γίνεται η εξελικτική διαδικασία. Παρότι η ανάπτυξη μια πειραματική πλατφόρμα η οποία να είναι ικανή να υποστηρίξει την εξελικτική διαδικασία παραμένει μια μη τετριμμένη εργασία, πολλές από τις αρχικές ανησυχίες και επικρίσεις σχετικά με τον τρόπο που αυτό θα πραγματοποιηθεί και θα μεταβιβαστεί από την προσομοίωση σε ένα πραγματικό ρομπότ έχουν υπολογιστεί. Υπάρχουν επαρκή παραδείγματα από ερευνητικές πλατφόρμες της εξελικτικής ρομποτικής που έχουν αποδείξει με επιτυχία την παραγωγή ενός ελεγκτή εργασίας σε πραγματικό ρομπότ. Επίσης, έχουν υπάρξει πολλά παραδείγματα ελεγκτών που με επιτυχία εξελίχθηκαν σε περιβάλλον προσομοίωσης και με επιτυχία μεταφέρθηκαν σε πραγματικό ρομπότ.

Η εξελικτική ρομποτική (ER) είναι η έρευνα για την ανάπτυξη αυτόματων μεθόδων για τη δημιουργία ευφύων αυτόνομων ελεγκτών ρομπότ, και το πράτει με τρόπο που δεν απαιτεί άμεσο προγραμματισμό από τον άνθρωπο. Ένα από τα κύρια επιτεύγματα του πεδίου της ER, στο σύνολό της, είναι ότι εξελιγμένες δομές ελέγχου ρομπότ (όπως είναι τα νευρωνικά δίκτυα) μπορούν να εκπαιδευθούν για να παράγουν λειτουργικές συμπεριφορές σε πραγματικά (ενσωματωμένα) αυτόνομα ρομπότ. Αυτό δεν έχει αποδειχθεί είναι ότι οι μέθοδοι ER μπορούν να επεκταθούν και να παράγουν ελεγκτές ρομπότ οι οποίοι είναι ικανοί για πολύπλοκες αυτόνομες συμπεριφορές. Ειδικότερα, καμία ER εργασία δεν έχει δείξει ακόμη ότι είναι δυνατόν να εξελιχθεί ένας πολύπλοκος ελεγκτής στη γενική περίπτωση ή για γενικευμένη εργασία.

Ανησυχίες που σχετίζονται με την αξιολόγηση της καταλληλότητας της συνάρτησης καταλληλότητας παραμένουν σε μεγάλο βαθμό χωρίς απάντηση. Μεγάλο μέρος της έρευνας του ER που παρουσιάζονται στη βιβλιογραφία παρουσιάζει κάποια μορφή συνάρτησης που έχει σχεδιαστεί όχι αυτόματα και είναι προσανατολισμένη για συγκεκριμένο έργο και που περισσότερο ή λιγότερο καθορίζει πώς να επιτύχει το έργο ή την συμπεριφορά για την οποία προορίζεται. Οι πιο σύνθετες συμπεριφορές αφορούν συνδυασμό 3 ή 4 συντονισμένων συνεργαζόμενων συμπεριφορών. Σε κάποιες μελέτες η μέθοδος της εξέλιξης μέσω της συνάρτησης καταλληλότητας έχει επιλεγεί για μια πιο προκαθορισμένη ή γνωστή από πριν λύση. Σε άλλες η συνάρτηση

καταλληλότητας περιλαμβάνει λειτουργίες που απαιτούν σχετικά λίγες εκ των προτέρων γνώσεις και η εξέλιξη μπορεί να προχωρήσει με σχετικά αμερόληπτο τρόπο. Αυτή είναι μια ενδιαφέρουσα αντίθεση στις μελέτες με στόχο την εξέλιξη μέσω απλής παλιννόστησης ή συμπεριφοράς αποφυγής αντικείμενου, πολλά από τα οποία χρησιμοποιούν σύνθετες και περίπλοκες συναρτήσεις συμπεριφοράς που σε μεγάλο βαθμό οδηγούν, με προκατάληψη, τον εξελιχθείσα ελεγκτή προς μια λύση που είναι σχεδόν προκαθορισμένη.

### 2.7.1 Κατηγορίες Συναρτήσεων καταλληλότητας.

Στην εξελικτική ρομποτική έχουν χρησιμοποιηθεί διάφορες μορφές συναρτήσεων καταλληλότητας για την εξέλιξη των ελεγκτών. Ο κάθε χρήστης έχει την άνεση αλλά και το δικαίωμα να συνθέσει την συνάρτηση καταλληλότητας που θεωρεί ότι θα βοηθήσει καλύτερα στην ανάπτυξη του εκάστοτε ελεγκτή.

Η εργασία των ***Nelson et al [8], Fitness functions in evolutionary robotics: A survey and analysis*** ταξινομεί τις συναρτήσεις καταλληλότητας και επίσης παρουσιάζονται και παραδείγματα ώστε αυτές να είναι πιο κατανοητές. Η ταξινόμηση βασίζεται στο ποσοστό με το οποίο η πρότερη γνώση κάποιων λειτουργιών αντικατοπτρίζεται στις συναρτήσεις καταλληλότητας που χρησιμοποιούνται για να εξελιχθούν συμπεριφορές ή ικανότητες των ρομπότ. Η αιτιολόγηση για τη χρήση της εκ των προτέρων γνώσης ως βάση για την ταξινόμηση είναι ότι αντικατοπτρίζει το επίπεδο της πραγματικά καινούργιας μάθησης που έχει επιτευχθεί από τον ελεγκτή.

Υπάρχουν βέβαια και άλλα μέσα με τα οποία σχεδιαστές εισάγουν τη δική τους εκ των προτέρων γνώση σχετικά με τον βέλτιστο τρόπο επιτεύξεις των εργασιών στο σχεδιασμό των πειραματικών συστημάτων που προορίζονται για εξέλιξη / μελέτη (ή μάθηση) σε αυτόνομα ρομπότ. Αυτές είναι η επιλογή κατάλληλων αισθητήρων και ενεργοποιητών, σχεδιασμός του περιβάλλοντα εκπαίδευσης, και η επιλογή των συνθηκών που θα γίνει η εξέλιξη. Η συνάρτηση καταλληλότητας είναι αυτή που περιέχει την πιο ρητή και πολυμορφική γνώση στις λύσεις των εργασιών. Τελικά ορίζονται επτά κατηγορίες τις συναρτήσεις καταλληλότητας όπως παρουσιάζονται στον παρακάτω πίνακα [8].

| Τύπος Συνάρτησης Καταλληλότητας  | Ποσοστό Πρότερης γνώσης   |
|--|---------------------------|
| ○ Συνάρτηση Καταλληλότητας Δεδομένων Εκπαίδευσης (Training data fitness functions)                           | ○ Πολύ Υψηλό              |
| ○ Συνάρτηση Καταλληλότητας Συμπεριφοράς (Behavioral fitness functions)                                       | ○ Υψηλό                   |
| ○ Συνάρτηση Καταλληλότητας αυξανόμενης Λειτουργικότητας (Functional incremental fitness functions)           | ○ Μεσαίο προς Υψηλό       |
| ○ Προσαρμοσμένη Συνάρτηση Καταλληλότητας (Tailored fitness functions)  | ○ Μεσαίο                  |
| ○ Συνάρτηση καταλληλότητας αυξανόμενης δυσκολίας περιβάλλοντος (Environmental incremental fitness Functions) | ○ Μεσαίο                  |
| ○ Ανταγωνιστική ή Συνεργατική Συνάρτηση Καταλληλότητας (Competitive and co-competitive selection)            | ○ Πολύ Χαμηλό προς Μεσαίο |
| ○ Αθροιστική ή Ολοκληρωτική Συνάρτηση Καταλληλότητας (Aggregate fitness functions)                           | ○ Πολύ Χαμηλό             |

Πίνακας 1: Κατηγορίες Συναρτήσεων Καταλληλότητας [8]



### 2.7.1.1 Συνάρτηση Καταλληλότητας Δεδομένων Εκπαίδευσης (Training data fitness functions)

Οι συναρτήσεις καταλληλότητας που χρησιμοποιούνται με σύνολα δεδομένων, δεν είναι αποκλειστικά για εξελικτική υπολογιστικών μεθόδων. Οι συναρτήσεις καταλληλότητας με δεδομένα εκπαίδευσης χρησιμοποιούνται σε διαδικασίες κατάρτισης βαθμιαία μειούμενες όπως μέθοδος επιστροφής σφάλματος πολλαπλασιασμού για εκπαίδευση νευρωνικών δικτύων και διάφορες προσαρμογές σε καμπύλη ή αριθμητικές μεθόδους.

Εδώ, η συνάρτηση καταλληλότητας μεγιστοποιείται, όταν σύστημα παράγει μια ελάχιστη τιμή εξόδου σφάλματος όταν ένα συγκεκριμένο σύνολο δεδομένων εισόδου συνδυάζεται με ένα σύνολο βέλτιστων εξόδων. Για ένα συγκεκριμένο πρόβλημα, ένα σύνολο δεδομένων εκπαίδευσης πρέπει να περιλαμβάνει επαρκή παραδείγματα, έτσι ώστε το σύστημα εκμάθησης να μπορεί να προεκτείνει ένα έγκυρο στοιχείο ελέγχου σε γενικευμένο νόμο. Έτσι, τουλάχιστον εμμέσως, το ιδανικό σύνολο δεδομένων εκπαίδευσης περιέχει γνώση από όλα τα κύρια χαρακτηριστικά του στοιχείου ελέγχου.

Η κύρια χρήση της συνάρτησης καταλληλότητας με δεδομένα κατάρτισης στο πεδίο της αυτόνομης μάθησης είναι στην περιοχή της μιμητικής μάθησης, όπου ένα ρομποτικό σύστημα μαθαίνει να μιμείται τη συμπεριφορά που δημιουργείται από έναν εκπαιδευτή άνθρωπο ή άλλου είδους.

### 2.7.1.2 Συνάρτηση Καταλληλότητας Συμπεριφοράς (Behavioral fitness functions)

Η συνάρτηση καταλληλότητας συμπεριφοράς είναι εστιασμένη στην αποστολή και την συντάσσει ο μελετητής με τρόπο που να μετράει – καταγράφει διάφορα στοιχεία του τι κάνει το ρομπότ και με ποιο τρόπο αυτό τα κάνει.

Οι συναρτήσεις αυτές γενικά περικλείουν πολλές υπό-συναρτήσεις ή όρους που στο τέλος συνδυάζονται σε ένα ισορροπημένο άθροισμα ή πηλίκο. Οι υπό-συναρτήσεις ή οι όροι αυτοί έχουν ως σκοπό να μετρήσουν την αντίδραση σε απλές ενέργειες και να καταγράψουν, σε «χαμηλό επίπεδο», των σημάτων αισθητήρων, ενεργοποιητών ή άλλων χαρακτηριστικών στοιχείων του ρομπότ. Αυτές αναφέρονται σαν όροι συμπεριφοράς και μετρούν μερικά στοιχεία της αντίδρασης του ρομπότ και όχι τι έχει αυτό καταφέρει.

Η ποιότητα που χαρακτηρίζει τις συναρτήσεις στην κατηγορία αυτή είναι ότι είναι φτιαγμένες μόνο από συνθήκες ή στοιχεία τα οποία διαλέγονται ώστε να δηλώνουν μια συμπεριφορά για μια προορισμένη λύση σε μια συγκεκριμένη αποστολή.

Για παράδειγμα, εκπαίδευση οχήματος να κινείται και να αποφεύγει εμπόδια. Η συνάρτηση καταλληλότητας θα περιέχει έναν όρο ο οποίος θα μεγιστοποιείται όταν οι μπροστινοί αισθητήρες του οχήματος ενεργοποιούνται σε μικρή απόσταση. Έτσι το σύστημα με τον τρόπο αυτό ρυθμίζεται ώστε το όχημα θα παράγει μια συγκεκριμένη έξοδο του ενεργοποιητή για μια συγκεκριμένη είσοδο των αισθητήρων. Η επιλογή αυτή έγινε για μια συμπεριφορά η οποία, κατά τον σχεδιαστή, θα δώσει την λειτουργία της αποφυγής των εμποδίων. Το όχημα όμως δεν έχει εξελιχθεί να αποφεύγει τα

εμπόδια απλά έτσι μαθαίνει να στρίβει όταν οι μπροστινοί αισθητήρες ενεργοποιούνται.

Κάποια στοιχεία στις συναρτήσεις καταλληλότητας συμπεριφοράς δεν επιλέγονται με στόχο την ακριβή σχέση σύνδεσης του αισθητήρα με τον αντίστοιχο ενεργοποιητή αλλά περισσότερο για τον σχεδιασμό των στοιχείων ελέγχου. Έστω ότι θέλουμε να αναπτυχθεί ένας ελεγκτής οχήματος ο οποίος να περνάει τον περισσότερο χρόνο του σε κίνηση. Πρέπει η συνάρτηση καταλληλότητας να μεγιστοποιείται όταν υπάρχει συνεχής κίνηση ως προς τον χρόνο. Όμως αυτή η εξέλιξη δεν θα δώσει καλό αποτέλεσμα αν πρέπει να γίνει συνδυασμός της σχέσης αισθητήρα – ενεργοποιητή.

Υπάρχουν βέβαια και άλλοι πιθανοί συνδυασμοί που παράγουν τα επιθυμητά στοιχεία ελέγχου, όπως ένας όρος που μεγιστοποιεί τον ρυθμό κίνησης προς τα εμπρός σε σχέση με την δραστηριότητα των μπροστινών αισθητήρων .

### **2.7.1.3 Συνάρτηση Καταλληλότητας Αυξανομένης Λειτουργικότητας (Functional incremental fitness functions)**

Στην συνάρτηση καταλληλότητας αυξανομένης λειτουργικότητας η εκπαίδευση του ρομπότ γίνεται τμηματικά. Αρχικά γίνεται εκπαίδευση σε μια απλή ικανότητα και μετά στηριζόμενη πάνω σε αυτήν την ικανότητα προχωράμε και χτίζουμε μια πιο πολύπλοκη συμπεριφορά.

Η διαδικασία αυτή της εξέλιξης όπως αυτή ακολουθήθηκε, δηλαδή προσθήκη μια ικανότητας μετά αύξηση της δυσκολίας και μετά ενσωμάτωση της νέας ικανότητας συνεχίζετε μέχρι να επιτευχθεί η τελικά προσδοκώμενη συμπεριφορά. Η όλη διαδικασία μπορεί να θεωρηθεί ως εκπαίδευση σε απλές ύπο συμπεριφορές και μετά στην συνέχεια εκπαιδεύετε σε πιο πολύπλοκες συμπεριφορές.

Η συνάρτηση καταλληλότητας με αυξανομένης λειτουργικότητα μας αποκαλύπτει μια μεγάλη δυσκολία στην εξελικτική ρομποτική. Για μια δύσκολη αποστολή υπάρχει μεγάλη πιθανότητα κάποιος ή όλοι οι ελεγκτές σε ένα πληθυσμό να μην επιτύχουν την αποστολή. Οι εκλεκτές αυτοί ορίζονται ως κάτω των προσδοκιών. Αν όλος ο πληθυσμός είναι κάτω τον προσδοκιών μπορεί για μια αποστολή η συγκεκριμένη συνάρτηση να μην δώσει καλό σκορ και έτσι ο πληθυσμός να μην εξελιχθεί. Η λύση για να αντιμετωπιστεί το συγκεκριμένο πρόβλημα είναι να υπολογίζεται και ο βαθμός δυσκολίας της εκάστοτε αποστολής ή ικανότητας.

Η κριτική που γίνεται στην συνάρτηση καταλληλότητας αυξανόμενης δυσκολίας λειτουργικότητας είναι ότι περιορίζουν την διαδρομή της εξέλιξης σε τέτοιο βαθμό που θεωρείται ότι ο ελεγκτής δεν έχει μάθει ή αποκτήσει πραγματικά νέες συμπεριφορές ή ικανότητες. Ο σχεδιαστής πρέπει να περικλείει στοιχεία της επιθυμητής λύσης αλλά και να κατασκευάσει τη διαδρομή ανίχνευσης μέσα στο σύνολο του προγραμματισμού του ελεγκτή. Για μη τετριμμένες αποστολές, δεν υπάρχει εγγύηση ότι το σχεδιαστικό αυτό πρόβλημα θα είναι βατό (trackable).

### **2.7.1.4 Προσαρμοσμένη Συνάρτηση Καταλληλότητας (Tailored fitness functions)**

Στην προσαρμοσμένη συνάρτηση καταλληλότητας πέρα από τους όρους μετρήσεως συμπεριφοράς περιλαμβάνονται και στοιχεία ολοκληρωτισμού τα οποία μετρούν το

βαθμό ή την προοπτική της ολοκλήρωσης της αποστολής. Το τμήμα αυτό μπορεί να είναι ανεξάρτητο ή όχι, συνδεδεμένο με οποιαδήποτε συμπεριφορά ή μέθοδο.

Έτσι είναι προφανές ότι η προσαρμοσμένη συνάρτηση είναι συνδυασμός δυο συναρτήσεων της συμπεριφοράς και της αθροιστικής ή ολοκληρωτικής. Παράδειγμα: Όχημα που εξελίσσετε να αντιδράει στο φως. Η πιο πιθανή συνθήκη στην συνάρτηση θα είναι να πάει το όχημα στο φως, χωρίς να ληφθεί υπόψιν κανένα άλλο στοιχείο. Έτσι θα έχουμε συντάξει μια ολοκληρωτική ή αθροιστική συνάρτηση καταλληλότητας. Προσθέτοντας όμως έναν όρο ο οποίος να μεγιστοποιεί το χρόνο που το όχημα κινείται προ της πηγής του φωτός τότε ο συνδυασμός των 2 αυτών όρων δημιουργεί μια προσαρμοσμένη συνάρτηση καταλληλότητας.

Σε αντίθεση με τις αθροιστικές ή ολοκληρωτικές συναρτήσεις καταλληλότητας οι ολοκληρωτικοί όροι στις προσαρμοσμένες συναρτήσεις μπορούν και μετρούν κάποιο βαθμό δυσκολίας στις επιμέρους αποστολές με τέτοιο τρόπο που να επιβάλει ή να εισάγει κάποια πρότερη γνώση στον ελεγκτή που εξελίσσεται.

Χρησιμοποιείται ο όρος 'προσαρμοσμένες' ώστε να δώσουμε έμφαση ότι αυτοί οι τύποι συναρτήσεων καταλληλότητας είναι προσαρμοσμένη στην αποστολή, διατυπωμένες στο χέρι και περιέχουν πολλούς τύπους μετρήσεων, προσαρμοσμένη ή φτιαγμένη από τον σχεδιαστή για το συγκεκριμένο πρόβλημα που συνήθως περιέχουν πληροφορίες της λύσης έμμεσα ή άμεσα.

#### 2.7.1.5 Συνάρτηση καταλληλότητας αυξανόμενης δυσκολίας περιβάλλοντος (Environmental incremental fitness Functions)

Μια άλλη περίπτωση είναι αντί απλά να αυξάνουμε την πολυπλοκότητα της συνάρτησης επιλογής, να γίνει σταδιακή εξέλιξη αυξάνοντας τη δυσκολία του περιβάλλοντος στο οποίο θα πρέπει να λειτουργούν τα ρομπότ. Αυτό αναφέρεται ως αυξανόμενης δυσκολίας περιβάλλοντος εξέλιξη.

Η αυξανόμενης δυσκολίας περιβάλλοντος εξέλιξη δεν περιορίζει το χώρο αναζήτησης στο βαθμό που η εξέλιξη πρέπει να συγκλίνει σε μια συγκεκριμένη και προκαθορισμένη λύση.

Σχετικά λίγες εργασίες έχουν γίνει με χρήση αυξανόμενης δυσκολίας περιβάλλοντος εξέλιξης. Από την έρευνα φαίνεται ότι η περιβαλλοντική σταδιακή εξέλιξη μπορεί να παράγει ελεγκτές ρομπότ που να είναι σε θέση να εκφράσουν σύνθετες συμπεριφορές. Ωστόσο, δεν ήταν σαφές σε ποιο βαθμό η επιλογή ή η επαύξηση της δυσκολίας του περιβάλλοντα εκπαίδευσης διαμόρφωσαν τους τελικούς ελεγκτές.

#### 2.7.1.6 Ανταγωνιστική ή Συνεργατική Συνάρτηση Καταλληλότητας (Competitive and co-competitive selection)

Στην κατηγορία αυτή χρησιμοποιείται άμεσος ανταγωνισμός μεταξύ των μελών ενός πληθυσμού που εξελίσσεται. Γενικά σε όλες σχεδόν τις ER έρευνες οι ελεγκτές ανταγωνίζονται υπό την έννοια ότι βαθμολογούνται τα επιτεύγματα τους και αυτοί οι βαθμοί συγκρίνονται κατά τη διάρκεια της επιλογής και του πολλαπλασιασμού. Ωστόσο, στην ανταγωνιστική εξέλιξη οι ελεγκτές των ρομπότ ανταγωνίζονται μέσα στο ίδιο περιβάλλον έτσι ώστε η συμπεριφορά του ενός ρομπότ να επηρεάζει άμεσα τη συμπεριφορά και, ως εκ τούτου, την συνάρτηση καταλληλότητας του άλλου.

Για παράδειγμα, σε μια εργασία αναζήτησης στόχου, ένα ρομπότ μπορεί να εμποδίσει το άλλο από την εκτέλεση της εργασίας απλά σπρώχνοντάς το μακριά από το στόχο. Έτσι το δεύτερο ρομπότ θα μπορούσε να λάβει καλύτερη βαθμολογία στην συνάρτηση καταλληλότητας αν δεν είχε εμποδιστεί από το πρώτο ρομπότ.

Σε μια συνεργατική εξέλιξη δύο ξεχωριστοί πληθυσμοί (εκτελώντας διαφορετικές εργασίες) ανταγωνίζονται μεταξύ τους μέσα στο ίδιο περιβάλλον. Παραδείγματα της συνεργατικής εξέλιξης αφορούν πληθυσμούς ρομπότ σε ρόλους θηρευτών και θηραμάτων.

Δύο συν-εξελισσόμενοι πληθυσμοί, εάν έχουν προετοιμαστεί ταυτόχρονα, είναι καλή ευκαιρία για την προώθηση της εξέλιξης πιο σύνθετων συμπεριφορών από το ένα στο άλλο. Καθώς ο ένας πληθυσμός εξελίσσει κάποιες δεξιότητες, ο άλλος απαντά με την βελτίωση και εφαρμογή καλύτερων συμπεριφορών. Αποτελέσματα από άλλους τομείς της εξελικτικής διαδικασίας δείχνουν ότι λαμβάνοντας υπόψη τις σωστές συνθήκες εξέλιξης, η επιλογή συγκεντρωτικών συναρτήσεων καταλληλότητας σε συνδυασμό με συναρτήσεις ενδό-πληθυσμιακού ανταγωνισμού μέσα σε έναν ενιαίο πληθυσμό, η εκτέλεση άμεσα ανταγωνιστικών εργασιών, μπορεί να παράγει πολύ ικανά και ανταγωνιστικά συστήματα.

#### 2.7.1.7 Αθροιστική ή Ολοκληρωτική Συνάρτηση Καταλληλότητας (Aggregate fitness functions)

Οι αθροιστικών αποτελεσμάτων ή ολοκληρωτικές συναρτήσεις καταλληλότητας χρησιμοποιούνται και καταγράφουν, σε υψηλό επίπεδο την επιτυχία ή την αποτυχία στην ολοκλήρωση μιας εργασίας χωρίς όμως να λαμβάνεται υπόψη πώς αυτή η εργασία ολοκληρώθηκε. Αυτού του είδους η βαθμολόγηση μειώνει ή εξαλείφει την ανθρώπινη επιρροή ή προκατάληψη στο εξελισσόμενο σύστημα και επικεντρώνεται μόνο στην αξιολόγηση του όφελος (ή του ελλείμματος) από τις συμπεριφορές των ρομπότ σε απλά επιτυχία/αποτυχία. Αυτό μερικές φορές ονομάζεται αξιολόγησης όλα-σε-ένα.

Παράδειγμα: ένα ρομπότ πρέπει να εντοπίζει και να συλλέγει αντικείμενα τα οποία στη συνέχεια να τα τοποθετεί σε μια συγκεκριμένη θέση (ή μια "φωλιά"). Η συνάρτηση καταλληλότητας συγκεντρωτικών αποτελεσμάτων θα αποτελείται μόνο από πληροφορίες που σχετίζονται μόνο με την ολοκλήρωση των εργασιών. Ας υποθέσουμε ότι η εργασία θεωρείται πλήρης όταν το αντικείμενο έχει τοποθετηθεί στην 'φωλιά'. Μια χαρακτηριστική συνάρτησης συγκεντρωτικών αποτελεσμάτων για το έργο αυτό θα ήταν να υπολογίζεται ο αριθμός των αντικειμένων που βρίσκονται στην φωλιά μετά το τέλος της δοκιμασίας.

Γενικά υπάρχει ένας δισταγμός στην χρήση συναρτήσεων καταλληλότητας συγκεντρωτικών αποτελεσμάτων. Ο λόγος του δισταγμού αυτού είναι ότι οι αρχικοί πληθυσμοί των ελεγκτών δεν έχουν γενικά κανένα ανιχνεύσιμο επίπεδο γενικής αρμοδιότητας για να εκτελέσουν απλές εργασίες (δηλαδή είναι κάτω των προσδοκιών, sub-minimal). Στο παράδειγμα που παρουσιάσαμε, εάν τα αντικείμενα ήταν αραιά τοποθετημένα σε ένα σύνθετο περιβάλλον και οι ελεγκτές του αρχικού μη εξελιγμένου πληθυσμού, είναι τυχαία ρυθμισμένοι χωρίς οποιαδήποτε ικανότητα πλοήγησης, αναγνώρισης αντικείμενου ή παλιννόστησης, οι πιθανότητες ένας από αυτούς τους

ελεγκτές να ολοκληρώσει την αποστολή είναι πραγματικά πολύ μικρή. Η κατάσταση αυτή αναφέρεται συχνά ως το πρόβλημα εκκίνησης.

Πάντως παρότι υπάρχει αυτή η δυσκολία, η χρήση αθροιστικών συναρτήσεων καταλληλότητας με τον ένα ή τον άλλο τρόπο φαίνεται να είναι αναγκαία προκειμένου να δημιουργηθούν πολύπλοκοι ελεγκτές στη γενική περίπτωση, ή αν κάποιος θέλει να αποφευχθεί η επιρροή από περιορισμούς μέσω της ανθρώπινης ή σχεδιαστής προκατάληψης στους ελεγκτές.

Για την εξέλιξη πραγματικά πολύπλοκων συμπεριφορών, η επιλογή χρήσης συναρτήσεων καταλληλότητας συμπεριφοράς σε συνδυασμό με την συνάρτηση αυξητικής λειτουργικότητας έχουν ως αποτελέσματα κυρίως την βελτιστοποίηση στοιχείων των ελεγκτών που έχουν ενσωματωθεί από τον άνθρωπο-σχεδιαστή, σε αντίθεση με την εξέλιξη ή εκμάθηση πραγματικά νέων ευφυών συμπεριφορών.

Υπάρχει τρόπος να ξεπεραστούν ορισμένα από τα προβλήματα που δημιουργούνται με την επιλογή των αθροιστικών συναρτήσεων συμπεριφοράς. Αρχικά μπορεί να γίνει χρήση προσαρμοσμένης συνάρτησης για να εκπαιδευτεί το ρομπότ μέχρι το σημείο στο οποίο έχει τουλάχιστον τη δυνατότητα επίτευξης μια συγκεκριμένη περίπλοκη εργασία σε κάποιο φτωχό αλλά ανιχνεύσιμο επίπεδο, και στη συνέχεια να εφαρμοστεί μια αθροιστικών αποτελεσμάτων συνάρτηση σε συνδυασμό με ένδον-πληθυσμιακό ανταγωνισμό και έτσι να οδηγήσει την εξελικτική διαδικασία να αναπτυχθούν οι ελεγκτές.

Ο ένδον-πληθυσμιακός ανταγωνισμός παρουσιάζει μια συνεχώς αυξανόμενη δυσκολία την εργασία σε ένα εξελισσόμενο πληθυσμό και είναι σε θέση να παράγει ελεγκτές που έχουν ικανότητες οι οποίες δεν έχουν οραματιστεί από ανθρώπινους σχεδιαστές

## 2.7.2 Αναλυτικά παραδείγματα συναρτήσεων καταλληλότητας [8]

Παρακάτω γίνεται μια παρουσίαση συναρτήσεων καταλληλότητας (μαθηματικός τύπος) σε σχέση με διάφορες εργασίες που έχουν καταγραφεί στην βιβλιογραφία.

### 2.7.2.1 Εφαρμοσμένες συναρτήσεις καταλληλότητας συμπεριφοράς

Στην εργασία [30] οι *D. Floreano et al.* Χρησιμοποίησαν ένα Khepera ρομπότ και εξελίχθηκε να εκτελεί μια διεργασία κινήσεως και αποφυγής εμποδίων. Κατά τη διάρκεια του πειράματος, το ρομπότ έμαθε να περιηγηθείτε γύρω από ένα περιβάλλον τύπου λαβύρινθο σε ένα ενιαίο κλειστό βρόχο, και να αποφύγει πρόσκρουση σε τοίχους. Το ρομπότ ήταν εξοπλισμένο με IR αισθητήρες για την ανίχνευση του περιβάλλοντός του.

Η συνάρτηση καταλληλότητας ήταν :

$$f = \text{mean}(u_l, u_r)(1 - \sqrt{|u_l - u_r|})(1 - S_{ir})$$

Όπου  $u_l$ ,  $u_r$  οι ταχύτητες των τροχών (αριστερών – δεξιών) και  $s_{ir}$  είναι το μέγιστο ρεύμα ενεργοποιήσεις των αισθητήρων. Θεωρείται μια συνάρτηση καταλληλότητας συμπεριφοράς επειδή βασίζει την συνάρτηση στις τοπικές κινητικές συμπεριφορές και τα σήματα του αισθητήρα και δεν μετρά άμεσα την ολοκλήρωση της εργασίας μερική ή συνολική.

Στην εργασία [30] οι *D. Floreano et al.* πραγματοποιούν ένα ακόμα πείραμα με το ρομπότ Khepera το οποίο είχε ως στόχο την περιοδική κίνηση προς ένα στόχο. Το ρομπότ ήταν να ταξιδέψει σε μια αρένα για ένα χρονικό διάστημα και στη συνέχεια να μετακίνηση σε μια θέση του στόχου και να παραμείνει εκεί για ένα σύντομο χρονικό διάστημα. Ο στόχος σημαδεύτηκε από μια πηγή φωτός και το ρομπότ ήταν εξοπλισμένο με φώτο-ανιχνευτές και IR αισθητήρες. Ήταν μια προσομοίωση φόρτισης της μπαταρίας του ρομπότ. Η συνάρτηση καταλληλότητας ήταν:

$$f = \text{mean}(u_l, u_r)(1 - S_{ir})$$

Η εξέλιξη του επιτυχημένου ελεγκτή πήρε 10 h έλεγχο του χρόνου με το πραγματικό ρομπότ και αναπτύχθηκαν 240 γενιές με πληθυσμό 100 ελεγκτές.

Στην εργασία [31] οι *H.H Lund et al.* πραγματοποίησαν πειράματα με μετακίνηση και αποφυγή αντικείμενου. Έγινε χρήση του ρομπότ Khepera και η συνάρτηση ήταν:

$$f = \text{mean}(u_l, u_r)(1 - (u_l - u_r)^2(1 - S_{ir}))$$

Όπου  $u_l$ ,  $u_r$  οι ταχύτητες των τροχών (αριστερών - δεξιών) και  $S_{ir}$  είναι το μέγιστο ρεύμα ενεργοποιήσεων των αισθητήρων.

Στην εργασία [32] οι *W. Banzhaf et al* εξέλιξαν τέσσερις χωριστές συμπεριφορές χρησιμοποιώντας ενσώματη εξέλιξη και γενετικό προγραμματισμό (GP). Χρησιμοποιήθηκε ρομπότ Khepera εξοπλισμένο με 8 IR αισθητήρες. Όλες οι συναρτήσεις καταλληλότητας ήταν συμπεριφοράς και είχαν διατυπωθεί με τέτοιο τρόπο ώστε να γίνεται χαρτογράφηση των εισόδων των αισθητήρων με τις εξόδους του ενεργοποιητή.

Οι συμπεριφορές που εξελίχθηκαν ήταν (α) κίνηση προς τα εμπρός με αποφυγή αντικείμενου, (β) παλιννόστηση / παρακολούθηση αντικείμενου, (γ) παρακολούθηση τοίχου και (δ) 'κρύβομαι στο σκοτάδι' (αποφεύγοντας το φως). Οι τέσσερις συναρτήσεις καταλληλότητας για τις τέσσερις συμπεριφορές είναι αντίστοιχα:

$$(α) \quad f_{(-)} = S_{ir} - (u_l + u_r - |u_l - u_r|)$$

Όπου  $u_l$ ,  $u_r$  οι ταχύτητες των τροχών (αριστερών - δεξιών) και  $S_{ir}$  είναι το άθροισμα ενεργοποιήσεων όλων των αισθητήρων.

$$(β) \quad f_{(-)} = (S_{ir1} + S_{ir2} + S_{ir3} + S_{ir4} - c)^2$$

Όπου  $c$  είναι μια σταθερά η οποία την επέλεξαν ώστε το ρομπότ να μάθει να ακολουθεί το αντικείμενο σε απόσταση τέτοια ώστε το άθροισμα των τεσσάρων μπροστινών αισθητήρων να είναι σχεδόν η τιμή της σταθερά  $c$ .

$$(γ) \quad f_{(-)} = (S_{ir1} - c_1)^2 + (S_{ir2} - c_2)^2 + S_{ir3}^2 - (u_l + u_r)^2$$

Όπου  $S_{ir1}$  και  $S_{ir2}$  είναι οι τιμές των αισθητήρων στα πλαϊνά τοιχώματα του ρομπότ,  $S_{ir3}$  ο αισθητήρας στην μπροστινή πλευρά, ενώ τα  $c_1$  και  $c_2$  είναι σταθερές και  $u_l$ ,  $u_r$  οι ταχύτητες των τροχών (αριστερών - δεξιών).

$$(δ) \quad f_{(-)} = S_{photo} - (u_l + u_r - |u_l - u_r|)$$

όπου  $S_{\text{photo}}$  είναι η ενεργοποίηση του φώτο-αισθητήρα και  $u_l$ ,  $u_r$  οι ταχύτητες των τροχών (αριστερών – δεξιών).

Στην εργασία [33] ο N. Jakobi εξέλιξε ρομπότ με πόδια και να μαθαίνει να μετακινείται και να αποφύγει εμπόδια. Το ρομπότ είχε IR αισθητήρες στην δεξιά και την αριστερή πλευρά και ένα προφυλακτήρα στο μπροστινό μέρος. Η συνάρτηση καταλληλότητας συμπεριφοράς σε αυτό το πείραμα έχει τέσσερα τμήματα, κάθε ένα σχεδιασμένο για τον υπολογισμό της ικανότητας να εκτελεί μια επιθυμητή πτυχή της εργασίας. Αυτό το ορίζει ο συντάκτης ανά περίπτωση με την χρήση τεσσάρων όρων με αμοιβαία αποκλεισμένων Boolean συντελεστών ως εξής:

$$f_{(-)} = B_1(u_l + u_r) + B_2(u_l - u_r) + B_3(-u_l + u_r) + B_4(-u_l - u_r)$$

Όπου έχουμε  $u_l$ ,  $u_r$  οι ταχύτητες των ποδιών (αριστερού – δεξιού) και  $B_1 - B_2$  είναι αμοιβαία αποκλεισμένοι Boolean συντελεστές που είναι μη μηδενικοί υπό τις ακόλουθες προϋποθέσεις:  $B_1$  είναι μη μηδενική, όταν υπάρχουν εμπόδια τα οποία βρίσκονται στο εύρος των υπέρυθρων αισθητήρων και ο αισθητήρας χτύπημα δεν εμπλέκεται,  $B_2$  είναι μη μηδενική, όταν υπάρχει αντικείμενο στο εύρος του δεξιού αισθητήρα,  $B_3$  είναι μη μηδενική, όταν υπάρχει ένα αντικείμενο στο εύρος του αριστερού αισθητήρα, και  $B_4$  είναι μη μηδενική, όταν εμπλέκεται ο αισθητήρας του προφυλακτήρα.

Η επιθυμητή συμπεριφορά καθορίζεται από τις ενδείξεις των αισθητήρων και το σώμα του ρομπότ αντιδράει ανάλογα σε κάθε μία από τις περιπτώσεις και η συνάρτηση συμπεριφοράς είναι σχεδιασμένη για να επιλέγεται για αυτά. Το σύστημα έχει προγραμματιστεί με έμμεσο τρόπο να αναπαράγει μια συγκεκριμένη εκ των προτέρων γνωστή λύση. Επίσης το ασυνήθιστο σε αυτή την εργασίας είναι ότι χρησιμοποιήθηκαν 3500 γενεές για την ανάπτυξη ελεγκτών ώστε να έχει αποτελεσματική μετακίνηση το πραγματικό ρομπότ. Αυτό είναι μεταξύ 10 και 100 φορές ο αριθμός που απαιτείται για άλλα παρόμοια πειράματα. Ωστόσο, πρέπει να σημειωθεί ότι έχουν γίνει ελάχιστες ερευνητικές προσπάθειες να διερευνηθεί ο ευφυής έλεγχος την μάθησης σε ένα octopod.

Στην εργασία [34] οι T. Gomi et al. σε πραγματικό εξάποδο ρομπότ εξέλιξαν τους βηματισμούς του. Η συνάρτηση καταλληλότητας συμπεριφοράς που χρησιμοποιείται είναι μία από τις πιο περίπλοκες που εντοπίστηκαν στη βιβλιογραφία για ρομπότ με πόδια. Η συνάρτηση είναι:

$$f = (\text{strides})(1 - \text{overcurrent})(\text{balanced})(1 - \text{bellytouch})$$

όπου κάθε όρος είναι μια συνάρτηση που βασίζεται σε ένα συνδυασμό της συμπεριφορά του ρομπότ και εισόδων των αισθητήρων. **Strides** μετράει τις κυκλικές κινήσεις των ποδιών, **overcurrent** μετράει τις εντολές του ενεργοποιητή που υπερβαίνουν την ικανότητα του κινητήρα σε ρεύμα, **balanced** ο βαθμός κλίσης του σώματος του ρομπότ, και το **bellytouch** μετράει τον αριθμό που το σώμα του ρομπότ πέφτει αρκετά χαμηλή ώστε να ξύσει στο πάτωμα. Το hexapod μπόρεσε να εξελιχθεί αποτελεσματικά εντός πενήντα γενιών. Όπως συμβαίνει με αρκετά από τα άλλα παραδείγματα έρευνας μάθησης βαδίσματος, αυτό το ρομπότ δεν είχε αισθητήρες και έτσι δεν μαθαίνουν να αντιδρούν έξυπνα ή δυναμικά στο περιβάλλον.

Οι *V. Mattellan et al* [35] πραγματοποίησαν μελέτη με πληθυσμό ελεγκτών ασαφούς λογικής. Η αποστολή του ρομπότ ήταν μετακίνηση και αποφυγή αντικείμενου. Η συνάρτηση καταλληλότητας περιέχει και ένα όρο έκφραση (φιλαργυρίας) που μειώνει τον αριθμό των κανόνων στο σύνολο κανόνων του ελεγκτή:

$$f = \frac{\text{mean}(u_l, u_r)(1 - |u_l - u_r|)(1 - S_{ir})}{|\text{rules}|}$$

Όπου  $u_l$ ,  $u_r$  οι ταχύτητες των τροχών (αριστερών - δεξιών) και  $S_{ir}$  είναι το μέγιστο ρεύμα ενεργοποιήσεις των αισθητήρων ανά δοκιμή και **rules** είναι ο αριθμός των κανόνων του ασαφούς ελεγκτή. Ένας πληθυσμός 100 ατόμων και είχε επιτυχία στην 60 γενεά να κινείται και παράλληλα να ξεφεύγει από τα εμπόδια.

Στην εργασία των

Στην εργασία των *P. Nordin et al.* [36] έγινε χρήση γενετικού προγραμματισμού για να εξελιχθεί Khepera ρομπότ για μετακίνηση και αποφυγή αντικείμενου. Η συνάρτηση καταλληλότητας ήταν :

$$f_{(-)} = c(|u_l - u_r| + |u_l| + |u_r| - (u_l - u_r)) + \sum S_{ir}$$

Όπου  $u_l$ ,  $u_r$  οι ταχύτητες των τροχών (αριστερών - δεξιών) και  $\sum S_{ir}$  είναι το άθροισμα του ρεύματος ενεργοποίησης των αισθητήρων. Ένας πληθυσμός του τεράστιου αριθμού των 10000 ατόμων χρησιμοποιήθηκε για 250 γενεές.

Οι *J. Liu et al.* [37] περιγράφουν μια αποστολή ώθησης αντικείμενου που αφορά μια συμπεριφορά sumo-ρομπότ. Οι Ελεγκτές εξελίχθηκαν να ωθήσουν ένα αντικείμενο έξω από την αρένα. Έγινε χρήση GP για να εξελιχθούν ελεγκτές που αποτελούνται από συμπεριφορές πρωτόγονες όπως " πιο μπροστά " και " αριστερή στροφή ".

$$f = \sum S_{active} + \sum arms_{holding} + \sum arms_{touching}$$

όπου  $S_{active}$  είναι ο αριθμός των ενεργών αισθητήρες,  $arms_{holding}$  είναι ο αριθμός των βραχιόνων που ασκούν πίεση στα πλευρά του αντικείμενου και  $arms_{touching}$  είναι ο αριθμός των βραχιόνων σε επαφή με το αντικείμενο. Οι ερευνητές ανέφεραν ότι το ρομπότ τελικά ήταν σε θέση να μάθει να σπρώχνει αντικείμενα χρησιμοποιώντας τη παραπάνω συνάρτηση.

Οι *H. S.Seok et al* [38] παρουσιάζουν την εξέλιξη μιας συμπεριφοράς αποφυγής φωτός (phototaxis) και το αντικείμενο σε ένα ρομπότ εξοπλισμένο με sonar και φωτοανιχνευτών. Η συνάρτηση συμπεριφοράς που χρησιμοποιείται εδώ έχει το ασυνήθιστο ότι περιλαμβάνει τιμές που μετρούνται σε προηγούμενα χρονικά βήματα. Η συνάρτηση συνοψίζεται ως:

$$\varphi(t + 1) = c_1(\varphi(t) + c_2(S_{photo\_max} - S_{photo})) + \frac{\sum S_{sonar}}{S_{sonar\_max}} + c_4$$

όπου  $S_{photo}$ ,  $S_{photo\_max}$ ,  $\sum S_{sonar}$ ,  $S_{sonar\_max}$  είναι η διέγερση του μπροστινού φωτοανιχνευτή, μέγιστη διέγερση φωτοανιχνευτή, το άθροισμα των διεγέρσεων αισθητήρα σόναρ, και η μέγιστη διέγερση του σόναρ αντίστοιχα.



### 2.7.2.2 Εφαρμοσμένες Προσαρμοσμένες συναρτήσεις καταλληλότητας

Οι προσαρμοσμένες συναρτήσεις καταλληλότητας περιέχουν συγκεντρωτικούς όρους που μετρούν κάποιο επίπεδο της ολοκλήρωσης της εργασίας, αλλά επίσης περιέχουν και όρους συμπεριφοράς.

Οι *F. Hoffmann et al.* [39] εξελίσσουν ρομπότ με ελεγκτή ασαφούς λογικής να εκτελεί εργασία παλιννόστησης και αποφυγής αντικείμενου. Χρησιμοποιήθηκε μια προσαρμοσμένη συνάρτηση καταλληλότητας χωρίς όμως να περικλείει όρους συμπεριφοράς. Η συνάρτηση που δηλώθηκε είχε δύο αλληλοαναιρούμενες περιπτώσεις, η καθένα μια με την δική της υπό-λειτουργία:

Περίπτωση 1<sup>η</sup> που έχουμε επαφή με αντικείμενο  $F_{collision} = \frac{t_{collision}}{t_{max}}$

$t_{collision}$  είναι ο χρόνος που έγινε η επαφή και  $t_{max}$  ο μέγιστος χρόνος που έχει οριστεί για την εργασία.

Περίπτωση 2<sup>η</sup> που δεν έχουμε επαφή με αντικείμενο  $F_{free} = 1 - \frac{d}{d_0}$

Όπου  $d$  είναι η εναπομείναν απόσταση του ρομπότ από τον στόχο και  $d_0$  είναι η αρχική απόσταση μεταξύ ρομπότ και στόχου όταν ξεκινάει η διαδικασία.

Πρέπει να σημειώσουμε ότι η διεργασία σταματάει όταν υπάρχει επαφή ή το ρομπότ φτάνει στο στόχο του ή όταν περάσει ο μέγιστος ορισμένος χρόνος.

Μετά από 30 γενιές είχαν εξελιχθεί ελεγκτές ικανοί να φτάσουν τη θέση στόχου χωρίς σύγκρουση σε περίπου το 50% των δοκιμών. Ίσως ο αριθμός των 30 γενεών να είναι μικρός και οι πληθυσμοί να μην είχαν συγκλίνει.

Ο A.Thompson [40] εξελίσσει μια απλή μετακίνηση με αποφυγή τοίχου σε ένα ρομπότ χρησιμοποιώντας ένα εξελίξιμο σύστημα ελέγχου. Χρησιμοποιήθηκε η παρακάτω έκφραση υπό ολοκλήρωση προσαρμοσμένη συνάρτηση καταλληλότητας:

$$f = e^{-c_1(dx)^2} + e^{-c_2(dy)^2} + B$$

όπου  $d_x$  και  $d_y$  είναι οι αποστάσεις από το κέντρο της αρένας για την τρέχουσα θέση του ρομπότ, και  $B$  είναι μια δυαδική τιμή η οποία παίρνει τιμές : 1 αν το ρομπότ κινείται και 0 αλλιώς. Η συνάρτηση έχει στόχο να κρατάει το ρομπότ μακριά από την αρχική του θέση (μέση της αρένας) και επίσης να είναι συνέχεια εν κινήσει. Το έργο αυτό πραγματοποιήθηκε σε ένα εργαστήριο ρομπότ χρησιμοποιώντας αισθητήρες sonar. Λειτουργικοί ελεγκτές εξελίχθηκαν σε 35 γενιές.

Η εξέλιξη ενός μικρού ρομπότ εφοδιασμένου με 2 κάμερες παρουσιάζουν οι D.keymeulen [41]. Η αποστολή ήταν να πάει σε ένα σημείο στόχο μέσα από μια διαδρομή με εμπόδια. Η διαδικασία απαιτούσε να γίνουν 64 δοκιμές και μετά εξαγονταν η τιμή της συνάρτησης. Να τονιστεί ότι περιέχει και ένα όρο ολοκληρωτισμού που είναι η μέτρηση των οριστικών δοκιμών.

$$F_{64trials} = \sum_1^{64} (B_{goalfound} + c_1 \left(1 - \frac{d_{goal}}{d_{max}}\right) + c_2 \left(1 - \left(1 - \frac{t_{goal}}{t_{max}}\right)\right))$$

όπου  $B_{\text{goal\_found}}$  είναι μια τιμή Boolean που είναι true εάν το ρομπότ βρεθεί στο αντικείμενο-στόχο κατά τη διάρκεια της εν λόγω δοκιμής,  $d_{\text{goal}}$  είναι η υπολειπόμενη απόσταση μεταξύ του ρομπότ και του στόχου στο τέλος της δοκιμαστικής περιόδου,  $t_{\text{goal}}$  είναι ο αριθμός του χρόνου (βήματα) που απαιτούνται για να βρει στο στόχο και  $d_{\text{max}}$  είναι η μεγαλύτερη απόσταση (γραμμική) που μπορεί να επιτευχθεί στο εκπαιδευτικό περιβάλλον. Επιτυχία επηρξε από έναν πληθυσμό των 20 ατόμων που εξελίχθηκε για 650 γενιές.

Στην εργασία των *A. Ishiguro et al* [42] με ρομπότ Khepera όπου εξελίχθηκαν ελεγκτές με αποστολή σπρώξιμο αντικείμενου και κίνηση στο φως. Στο έργο αυτό, το ρομπότ εντόπιζε αντικείμενα (μικροί κύλινδροι) μέσα σε ένα περιβάλλον και στη συνέχεια τους ωθούσε σε θέση στόχο που χαρακτηρίζεται από μια πηγή φωτός. Για αυτό το πείραμα, οι ελεγκτές είχαν εξελιχθεί σε προσομοίωση και στη συνέχεια μεταφέρθηκε στο πραγματικό ρομπότ. Οι συγγραφείς χρησιμοποίησαν την ακόλουθη συνάρτηση καταλληλότητας:

$$F = \left( 1 - \frac{d_f(\text{peg}, \text{goal})}{d_o(\text{peg}, \text{goal})} \right)^2$$

όπου  $d_o$  και  $d_f$  είναι η μέτρηση της απόστασης μεταξύ της αντικείμενου και της πηγής φωτός στην αρχή και στο τέλος μιας δοκιμαστικής περιόδου αξιολόγησης. Η F εμπίπτει στην κατηγορία των προσαρμοσμένων συναρτήσεων καταλληλότητας επειδή μετρά κάποιο βαθμό επιτυχίας. Η συνάρτηση δεν εισφέρει υψηλού βαθμού εκ των προτέρων γνώση των στρατηγικών του ελεγκτή. Η εξέλιξη επήλθε στις 200 γενιές, όπου ο καλύτερος ελεγκτής ήταν σε θέση να ολοκληρώσει το έργο στο 92% των δοκιμών.

Ελεγκτής με αντικείμενο την μετακίνηση και αποφυγή αντικείμενου εξελίχθηκε σε προσομοίωση και μεταφέρθηκε σε ένα RWI B21 ρομπότ υπηρεσίας εξοπλισμένο με σόναρ στην εργασία των *M. Ebner et al.* [43]. Αυτό το ρομπότ είναι μεγαλύτερο από τα περισσότερα ρομπότ που χρησιμοποιούνται σε διαδικασίες ER (πάνω από ένα μέτρο ψηλό) και δοκιμάστηκε σε ένα διάδρομο του κτιρίου. Οι συγγραφείς διατύπωσαν μια σύνθετη συνάρτηση για μια απλή εργασία. Η συνάρτηση προσπαθεί να μεγιστοποιήσει το χρόνο κίνησης ελαχιστοποιώντας την περιστροφή του ρομπότ. Αυτό μπορεί να γραφτεί ως:

$$F = 1 - \frac{t}{t_{\text{max}}} \sqrt{\frac{|\sum r|}{\omega_{\text{net}} t_{\text{max}}}}$$

Όπου  $t$  είναι ο χρόνος της πρώτης σύγκρουσης,  $t_{\text{max}}$  είναι ο μέγιστος χρόνος για μια περίοδο αξιολόγησης,  $\sum r$  είναι το άθροισμα των όλα των διακριτών περιστροφών που πραγματοποιούνται από το ρομπότ, και  $\omega_{\text{net}}$  είναι η μέση γωνιακή ταχύτητα κατά τη διάρκεια μιας περιόδου αξιολόγησης. Η εξέλιξη πραγματοποιήθηκε με το πραγματικό ρομπότ RWI B21 και απαιτήθηκαν περίπου 200 h εξέλιξης για 50 γενιές με ένα μέγεθος πληθυσμού 75. Αυτό τονίζει την ανάγκη για ύπαρξη υψηλής ποιότητας προσομοιωτές στην ER που πρόκειται να χρησιμοποιηθούν για τη δημιουργία ελεγκτών ισχυρών ρομπότ που θα λειτουργούν σε πραγματικό περιβάλλον χώρο και όχι έναν εργαστηριακό.

Στην εργασία των *D. Keymeulen et al.* [44] όπου ένα ρομπότ Khepera μετακινείται σε ένα ενδιάμεσο σημείο και από εκεί στο τελικό σημείο. Όταν το ρομπότ φτάσει στο ενδιάμεσο σημείο μια πηγή φωτός ενεργοποιείται. Γίνεται χρήση μια απλής προσαρμοσμένης συνάρτησης καταλληλότητας.

$$F = \frac{t_{goal}}{t_{max}}$$

Όπου  $t_{goal}$  είναι ο χρόνος που απαιτήθηκε από τη στιγμή που ενεργοποιήθηκε η πηγή φωτός μέχρι το τελικό σημείο. Ενώ  $t_{max}$  είναι ο ολικός χρόνος της δοκιμής.

Στην εργασία των *I. G. Sprinhuizen et al.* [45], όπου ένα ρομπότ Khepera με IR αισθητήρες έπρεπε να σπρώξει ένα αντικείμενο προς μια φωτεινή πηγή, ο συγγραφέας έκανε χρήση ξεχωριστών πληθυσμών ελεγκτών χρησιμοποιώντας τέσσερις διαφορετικές συναρτήσεις καταλληλότητας, όπου μετά έκανε σύγκριση των αποτελεσμάτων. Οι συναρτήσεις είναι :

$$(α) \quad f = d_{box} - \frac{1}{2} d_{box,robot}$$

Όπου  $d_{box}$  είναι η απόσταση που έκανε το αντικείμενο και  $d_{box, robot}$  είναι η τελική απόσταση μεταξύ ρομπότ και αντικειμένου.

$$(β) \quad f = \Delta d_{box} - \frac{1}{2} \Delta d_{box,robot}$$

Όπου το  $\Delta$  είναι η μεταβολή ως προς τον χρόνο.

$$(γ) \quad f = c_1(s_{ir2} + s_{ir3}) + (1 - c_2 \sum s_{photo})$$

Όπου  $s_{ir2}$  και  $s_{ir3}$  είναι οι εμπρόσθιοι αισθητήρες του ρομπότ και  $\sum s_{photo}$  είναι το άθροισμα των τιμών ενεργοποίησης των φώτο-αισθητήρων του ρομπότ.

$$(δ) \quad f = c_1(s_{ir2} + s_{ir3}) + c_2|u_l + u_r| - c_3|u_l - u_r|$$

Όπου  $u_l$ ,  $u_r$  οι ταχύτητες των τροχών (αριστερών - δεξιών).

Με τον συνδυασμό των παραπάνω συναρτήσεων έγινε εξέλιξη για 250 γενεές με πληθυσμό 30 ατόμων. Πάντως η 1<sup>η</sup> συνάρτηση αναφέρθηκε να δίνει τον καλύτερο ελεγκτή.

Στην εργασία του *U. Nehmzow* [46] όπου εξετάστηκε η ανάπτυξη τριών συμπεριφορών: προσανατολισμού σε φωτεινή πηγή, αποφυγή αντικειμένου και μεταφορά σε θέση ηρεμίας (ήταν σε συνδυασμό μ' ένα άλλο ακόμα ρομπότ στο πεδίο δοκιμών. Το ρομπότ ήταν κατασκευασμένο κυρίως από LEGOs και εξοπλισμένα με IR αισθητήρες, επαφής και φώτο-ανιχνευτών. Για κάθε εργασία, χρησιμοποιήθηκε μια απλή προσαρμοσμένη συνάρτηση καταλληλότητας.

(α) προσανατολισμός σε φωτεινή πηγή:

$$F_{phototaxis} = \frac{\sum t_{light}}{t_{max}}$$

Όπου  $\sum t_{light}$  είναι το άθροισμα των χρονικών στιγμών που το ρομπότ έχει διεύθυνση προς την φωτεινή πηγή και  $t_{max}$  είναι ο ολικός χρόνος της δοκιμής.

(β) αποφυγή αντικειμένου:

$$F_{avoidance} = \frac{\sum t_s}{t_{max}}$$

Όπου  $\sum t_s$  είναι το άθροισμα των χρονικών στιγμών που κανένας από τους αισθητήρες δεν δίνει σήμα επαφής και  $t_{max}$  είναι ο ολικός χρόνος της δοκιμής.

(γ) μεταφορά σε θέση ηρεμίας

$$F_{homing} = t_{max} - t_{complete}$$

Όπου  $t_{complete}$  είναι ο χρόνος που απαιτείται για να έρθει το ρομπότ κοντά σε ένα άλλο ρομπότ που είναι μέσα στο πεδίο.

Αναφέρθηκε εξέλιξη μέσα σε 30 γενεές.

Οι *R.A. Watson et al.*[47] εξελίξαν ένα ρομπότ να προσεγγίζει μια φωτεινή πηγή. Η εξέλιξη πραγματοποιήθηκε σε έναν πληθυσμό οκτώ ρομπότ χρησιμοποιώντας έναν ασύγχρονο αλγόριθμο σε συνδυασμό με μια προσαρμοσμένη συνάρτηση καταλληλότητας. Ασύγχρονη εννοούμε ότι ο εξελικτικός αλγόριθμος δεν χρησιμοποίησε συγκεκριμένη γενιά ή περίοδο. Απλά, ένας ελεγκτής μόλις είχε επιτύχει μια πολύ υψηλή τιμή στην συνάρτηση, μετέδιδε τον κώδικα του στα άλλα οχήματα ανεξάρτητα από το τι είχαν επιτύχει οι άλλοι. Τότε το ποσό ενέργειας του έπεφτε σε πολύ χαμηλό επίπεδο. Η συνάρτηση υπολογίζει το επίπεδο ενέργειας και ενημερώνεται σε κάθε χρονικό βήμα.

$$\varphi(t) = \varphi(t - 1) + c_1 B_{reward} - c_2 B_{penalty}$$

Όπου  $t$  είναι ο χρόνος,  $B_{reward}$  είναι μια μεταβλητή boolean που είναι αληθής κάθε φορά που το ρομπότ πλησιάζει την φωτεινή πηγή και  $B_{penalty}$  η οποία γίνεται αληθής κάθε φορά που ελεγκτής μεταδίδει τα στοιχεία του.

Ο αλγόριθμος είναι πλήρως ασύγχρονος και δεν υπάρχει η κλασική διαδικασία ελεγχος πλυθισμού οπότε μετά παμε σε νέα γενεά.

Στην εργασία των *M. Quinn et al.* [48] όπου ομάδα τριών ρομπότ πρέπει να κινηθεί σε σχηματισμό. Τα ρομπότ χρησιμοποιούν ένα τυπικό σύστημα διαφορικής κίνησης και το κάθε ένα έχει τέσσερις IR αισθητήρες για την ανίχνευση του περιβάλλοντος. Χρησιμοποιείται μια προσαρμοσμένη συνάρτηση καταλληλότητας σχετικά περιπλοκή που δίνεται από:

$$f = p \sum_1^{T_{max}} (D_{gain}(d, d_{best}) (1 + \tanh\left(\frac{S}{20}\right)))$$

όπου  $P$  είναι ένας όρος ποινής εφόσον υπάρχει σύγκρουση, μάλιστα είχε τάση να μειώνεται προς το 0 καθώς ο αριθμός των συγκρούσεων αυξάνεται,  $D_{gain}$  είναι μια συνάρτηση με όρους την τρέχουσα απόσταση  $d$  και την μετρημένη καλύτερη απόσταση  $d_{best}$ . Ο όρος  $S$  είναι ένα μέτρο που χαρακτηρίζει την διασπορά της ομάδας. Ο όρος της άθροισης έχει εισέλθει στην συνάρτησης καταλληλότητας επειδή ο παράγοντας ποινής σύγκρουσης  $P$ , εφαρμόζεται μετά την ολοκλήρωση της αξιολόγησης στο προκαθορισμένο χρονικό διάστημα και από τα άλλα στοιχεία της

αξιολόγησης. Οι συγγραφείς χρησιμοποίησαν ένα πληθυσμό 50 ελεγκτών που έτρεξε 100 διαφορετικές εξελίξεις. Μάλιστα αναφέρεται ότι σε κάθε γενεά του εξελικτικού, αναδεικνύονταν και ένας ελεγκτής κατάλληλος που μπορούσε να επιτύχει τον στόχο της μετακίνησης της ομάδας στην προσομοίωση αλλά και σε τρία πραγματικά ρομπότ.

11) Η εργασία των *A.L. Nelson et al.* [49] ασχολείται με την εξέλιξη ενός ρομπότ ΕνBot για άσκηση μετακίνησης και αποφυγή αντικειμένου. Το ρομπότ ΕνBot είναι μικρό σε μέγεθος και έχει 5 δυαδικούς αισθητήρες επαφής. Η προσαρμοσμένη συνάρτηση καταλληλότητας είναι:

$$F = c_1 d_{net} + c_2 d_{max} + c_3 d_{arc\_length} - c_4 B_{stuck}$$

όπου  $d_{net}$  μετρά την απόσταση μετατόπισης του ρομπότ αρχική με τελική θέση,  $d_{max}$  μετρά την μεγαλύτερη απόσταση που επιτεύχθηκε από το ρομπότ ανά πάσα στιγμή κατά τη διάρκεια της δοκιμής,  $d_{arc\_length}$  είναι το μήκος τόξου της γραμμής της διαδρομής του ρομπότ κατά τη διάρκεια μιας δοκιμής και  $B_{stuck}$  είναι μια τιμή Boolean που είναι αληθής, εάν το ρομπότ είναι μόνιμα ακινητοποιημένο κατά τη διάρκεια της δοκιμής.

Οι συντάκτες αναφέρουν ότι αποτελεσματικός ελεγκτής εξελίχθηκε σε προσομοίωση και δοκιμάστηκε σε πραγματικό ρομπότ χρησιμοποιώντας ένα πλήθος 20 ελεγκτών για 3000 γενιές. Αυτός ο αριθμός των γενεών είναι αρκετά υψηλή σε σύγκριση με άλλες εργασίες ER.

12) Η εργασία των *G. Gari et al.* [50] όπου ένα ρομπότ πρέπει να περάσει από τρία σημεία με συγκεκριμένη σειρά. Η συνάρτηση καταλληλότητας ανανεώνονταν για κάποιο χρονικό διάστημα

$$\varphi_{goal\_position} = \begin{cases} 1 & \text{σε ακολουθία} \\ -1 & \text{υπολοιπες περ.} \end{cases}$$

Η συνάρτηση ανανεώνονταν μόνο όταν το ρομπότ έφτανε σε ένα από τα σημεία στόχους.

Στην εργασία των *G.B. Parker et al.* [51], αναπτύσσει ελεγκτή για εργασία ακολουθίας φωτός σε ένα περιβάλλον με πολλά εμπόδια. Ρομπότ κατασκευασμένο από LEGO Mindstorm, εξοπλισμένο με δύο φώτο-ανιχνευτές και ενός μεμονωμένου αισθητήρα αφής. Έγινε χρήση μιας απλής προσαρμοσμένη συνάρτηση καταλληλότητας και δίνεται από:

$$F = d_{max}^2 - d^2$$

Όπου  $d_{max}$  είναι η μεγαλύτερη διάσταση του πεδίου εκπαίδευσης και  $d$  η απόσταση μεταξύ του ρομπότ και της πηγής φωτός.

### 2.7.2.3 Εφαρμοσμένες Αθροιστικές ή Συγκεντρωτικές Συναρτήσεις Καταλληλότητας (Aggregate fitness functions)

Οι Αθροιστικές ή συγκεντρωτικές συναρτήσεις καταλληλότητας υπολογίζουν μόνο την ολοκλήρωση των εργασιών χωρίς εξάρτηση από οποιοδήποτε συγκεκριμένη Αισθητήρα - ενεργοποιητή συμπεριφορά. Οι αθροιστικές ή συγκεντρωτικές

συναρτήσεις καταλληλότητας συλλέγουν (ή αθροίζουν) το όφελος (ή το έλλειμμα) όλων των πτυχών του ρομπότ.

Η εργασία των *G.S.Hornby et al.* [52], όπου εξελίσσεται η συμπεριφορά ενός ρομπότ Sony AIBO να σπρώχνει μια μπάλα.

$$f = d_{ball}$$

Η συνάρτηση μετράει μόνο την απόσταση που μετακινήθηκε η μπάλα.

Στην εργασία των *E.J.P Earon et al.* [53], όπου εξελίσσουν τους βηματισμούς για ένα ξάποδο ρομπότ. Οι ελεγκτές βιάδισης ήταν με τη μορφή πίνακα αναζήτησης.

$$f = d$$

Η συνάρτηση συγκεντρωτικών αποτελεσμάτων μετρούσε την απόσταση που διάνυσε το ρομπότ, ενώ περπατούσε σε διάδρομο.

Εργασία των *F. Hoffmann et al.* [54], με εξέλιξη της συμπεριφοράς μετακίνησης ενός ρομπότ (LEGO Mindstorm kit) με αισθητήρες αφής για ανίχνευση αντικειμένων που πρέπει να ταξιδεύει σε ένα περιβάλλον το οποίο περιέχει μια πίστα και μικρά εμπόδια που τοποθετούνται κατά μήκος των τοίχων. Η αρχιτεκτονική του ελεγκτή ήταν μια απλή χαρτογράφηση τιμές αισθητήρα σε ενεργοποίηση μηχανικών μελών.

$$f = d_{arc\_length}$$

Η λειτουργία γυμναστηρίου απλά μετράει την απόσταση που ταξίδεψε από το ρομπότ (όπως αναφέρθηκε από ένα περιστρεφόμενο τροχό και μετρητή χιλιομετρικών αποστάσεων που ήταν συνδεδεμένος στο ρομπότ) για μια χρονική περίοδο στην εκάστοτε περίοδο αξιολόγησης. Αυτή η συνάρτηση μπορεί να θεωρηθεί αθροιστικών ή συγκεντρωτικών αποτελεσμάτων εφόσον η εργασία θεωρείται απλή μετακίνηση σε ένα άγνωστο περιβάλλον. Μετά από 20 γενιές, ο καλύτερος ελεγκτής ήταν σε θέση διατηρεί συνεχή μετακίνηση με ταχύτητα 3.5 m/s χωρίς να συγκρούεται με εμπόδια.

Η εργασία των *P. Augustsson et al.* [55] όπου εξελίσσεται η συμπεριφορά ανύψωσης σε ένα ιπτάμενο ρομπότ. Η συνάρτηση που ήταν πολύ απλή:

$$f = h$$

Όπου  $h$  είναι το ύψος που καταφέρνει το ρομπότ να ανυψώσει το αντικείμενο. Ένας πολύ μικρός πληθυσμός 4 ατόμων εξελίχθηκε με επιλογή τουρνουά γενετικών αλγορίθμων. Το ρομπότ έμαθε να αναπαράγει ανέλκυση ανελκυστήρα, αλλά το ρομπότ δεν ήταν σε θέση να παράγει επαρκή ανέλκυση υπό τη δική του ενέργεια. Ως εκ τούτου, η διαδικασία εξέλιξης δεν ήταν απόλυτα επιτυχής.

Στην εργασία των *I. Zufferey et al.* [56], όπου ένα μικρό ρομποτικό αερόστατο εξοπλισμένο με μια φωτογραφική μηχανή και τοποθετήθηκε σε ένα μικρό δωμάτιο με σημάδια σε μορφή γραμμικού κώδικα στους τοίχους και εξελίχθηκε να κινείται και αποφεύγει τον τοίχο. Η συνάρτηση καταλληλότητας έλεγχε τη μέση της ταχύτητας σε κάθε δοκιμαστική περίοδο:

$$f = u$$

όπου  $u$  είναι η τρέχουσα ταχύτητα του ρομποτικού αερόστατου σε κάθε χρονική στιγμή. Η συνάρτηση θεωρείται συγκεντρωτικών αποτελεσμάτων εφόσον αφορά την συμπεριφορά της κίνησης. Χρησιμοποιώντας έναν πληθυσμό των 60, επιτυχημένοι ελεγκτές είχαν εξελιχθεί σε 20 γενιές με εφαρμογή και σε φυσικό ρομπότ.

Στην εργασία των *I. Macinnes et al.* [57], ρομπότ κατασκευασμένο από LEGO και μονάδες σέρβο. Η εξέλιξη ήταν ικανότητες στην μετακίνηση σε περιβάλλον προσομοίωσης και, στη συνέχεια, να κατασκευαστεί στο εργαστήριο με πραγματικό υλικό. Η συνάρτηση μετρούσε την συνολική καθαρή απόσταση μετακίνησης  $d$  κατά τη διάρκεια της δοκιμαστικής περιόδου:

$$f = d$$

Υπήρχε μια αρχική περίοδο διευθέτησης πριν από κάθε περίοδο υπολογισμού της συνάρτησης καταλληλότητας. Αυτό έγινε για να αποφευχθεί η επιλογή για το ρομπότ που κινείται και έτσι η συνάρτηση να είναι τεχνικά ως προσαρμοσμένη, σε ένα μικρό βαθμό. Η εξέλιξη ενός λειτουργικού ελεγκτή έγινε σε 2000 γενιές. Το καλύτερο ρομπότ και ελεγκτής που κατασκευάστηκε αναφέρθηκε για να είναι σε θέση να μετακινήσει περίπου 14 εκατοστά ανά λεπτό.

Στην εργασία των *S. Chernova et al.* [58], όπου έγινε εκμάθηση βάρδισης χρησιμοποιώντας μια ελαφρώς διαφορετική συνάρτηση. Χρησιμοποιήθηκε ένα τετράποδο ρομπότ AIBO της Sony. Εξελίχθηκαν ελεγκτές οι οποίοι είχαν την μορφή ενός συνόλου 12 παραμέτρων βάρδισης. Η συνάρτηση μετρούσε την μέση ταχύτητα που επιτύγχανε το ρομπότ:

$$f = \frac{d}{t_{max}}$$

όπου  $d$  είναι η δανεισθείσα απόσταση και το  $t_{max}$  είναι η χρονική διάρκεια μιας συγκεκριμένη αξιολόγησης. Σημειώστε ότι από το  $t_{max}$  είναι σταθερή, τότε η συνάρτηση είναι παρόμοια άλλες που χρησιμοποιήθηκαν σε άλλα πειράματα. Η διαδικασία της εξέλιξης επέτρεψε στο ρομπότ να ταξιδέψει 20% πιο γρήγορα από ό, τι το καλύτερο σύστημα βαδίσματος που προκύπτει από την καλύτερη μη αυτόματη ρύθμιση των παραμέτρων.

Για την εξέλιξη ενός ελεγκτή ασαφούς λογικής ενός τροχήλατου ρομπότ, γίνεται χρήση ενός πεδίου δοκιμών και επί αυτού του πεδίου στο ρομπότ ανατίθεται μια αποστολή ή άσκηση. Από το αποτέλεσμα αυτής της άσκησης μέσω της συνάρτησης καταλληλότητας βαθμολογείται η απόδοση. Μέσω της βαθμολογίας αυτής γίνεται και η αξιολόγηση των γονέων για την συνέχιση της διαδικασίας εξέλιξης.

## 2.8 Ασκήσεις ή Αποστολές ρομπότ στην Εξελικτική ρομποτική

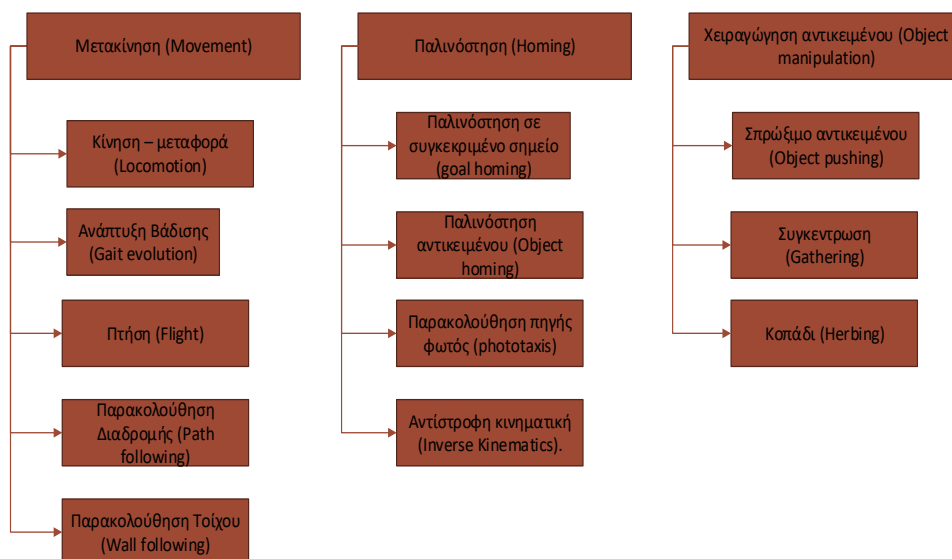
Στο πεδίο της εξελικτικής Ρομποτικής (ER) υπάρχουν πάρα πολλές περιπτώσεις όπου ο κάθε μελετητής επιλέγει την δική του προσέγγιση για την διαδικασία της εξέλιξης του ελεγκτή του ρομπότ του. Στην βιβλιογραφία έχουν μελετηθεί διαφορετικού τύπου αποστολές ή εργασίες. Μια παρουσίαση των αποστολών που έχουν χρησιμοποιηθεί στην εξελικτική ρομποτική και παράλληλα παρουσιάζεται και μια κατηγοριοποίηση των αποστολών αυτών [9]. Οι εργασίες μπορεί να κατηγοριοποιηθούν ανάλογα με

κάποιες ιδιότητες, όπως ο βαθμός αλληλεπίδρασης με το περιβάλλον ή κατά πόσον εμπλέκονται ομάδες ρομπότ ή απλά μονάχα ένα.

Η πρώτη κατηγορία είναι αυτή στην οποία ο στόχος είναι να μάθει το ρομπότ να κινείται. Στην κατηγορία αυτή περιλαμβάνονται διάφορων τύπων ρομπότ, τροχοφόρα, ρομπότ με πόδια ή άκρα και ρομπότ με φτερά. Η δεύτερη κατηγορία είναι αυτή στην οποία οι εργασίες έχουν ένα συγκεκριμένο προορισμό ή προσανατολισμό, όπως το αντικείμενο της παλιννόστησης ή της όδευσης προς μια φωτεινή πηγή (phototaxis). Τέλος η τρίτη κατηγορία είναι οι εργασίες που απαιτούν από το ρομπότ να χειραγωγήσουν το περιβάλλον στο οποίο δρουν, για παράδειγμα να ωθήσουν ένα αντικείμενο ή την συγκέντρωση. Η ανάλυση των κατηγοριών παρουσιάζεται στην εικόνα 38. Αυτή η κατηγοριοποίηση σε αύξουσα πολυπλοκότητα, επειδή οι εργασίες της μίας κατηγορίας είναι δευτερεύουσες εργασίες των εργασιών στην επόμενη κατηγορία. Για παράδειγμα μια εργασία, όπως η πίεση ενός πλαισίου απαιτεί επίσης την εκμάθηση της μετακίνησης.

Μερικές φορές οι εργασίες αφορούν ομάδα ρομπότ τα οποία συνεργάζονται ή/και να ανταγωνίζονται. Τα ρομπότ πάντα σιωπηρά ανταγωνίζονται μεταξύ τους μέσω της συνάρτησης καταλληλότητας. Όμως οι εργασίες θεωρούνται συνεργατικές ή ανταγωνιστικές μόνο όταν το ρομπότ που λειτουργούν ταυτόχρονα, πράγμα που δυνητικά επηρεάζει το ένα το άλλο.

Για τις εργασίες αυτές τα ρομπότ θα μπορούσαν να λειτουργήσουν σε μια ομάδα προς ένα κοινό στόχο, ή πολλές ομάδες μπορούν να είναι ενεργές την ίδια στιγμή, ενδεχομένως με διακριτά (υπο) καθήκοντα. Παράδειγμα από μια ανταγωνιστική εργασία είναι ένα σύστημα θηρευτών-θηραμάτων (predator – prey), όπου μια ομάδα ρομπότ καταδιώξει και η άλλη αποφεύγει.



Εικόνα 38 : Εργασίες στην Εξελικτική Ρομποτική

Μετακίνηση (movement) είναι το συλλογικό όνομα για τις διάφορες μεθόδους που χρησιμοποιούνται από τα ρομπότ να μεταφέρουν τον εαυτό τους από τόπο σε τόπο. Τα τροχοφόρα ρομπότ είναι συνήθως αρκετά αποδοτικά ενεργειακά και απλά στον



έλεγχο. Ωστόσο, άλλες μορφές μετακίνησης μπορεί να είναι πιο κατάλληλες για μια σειρά από λόγους, όπως για παράδειγμα η διέλευση μέσα από ανώμαλο έδαφος, καθώς και η μετακίνηση με παράλληλη αλληλεπίδραση στο ανθρώπινο περιβάλλον.

Επίσης η μελέτη δίποδων ή ρομπότ σε μορφή εντόμου μπορεί να επηρεάσει ευεργετικά την βιομηχανική. Ένας σημαντικός στόχος σε αυτό το πεδίο είναι η ανάπτυξη δυνατοτήτων από το ρομπότ να αποφασίζει αυτόνομα πώς, πότε, και από πού να προχωρήσει. Ωστόσο, ο συντονισμός ενός μεγάλου αριθμού αρθρώσεων σε ένα ρομπότ ακόμα και για απλές αποστολές, όπως η κίνηση σε, είναι δύσκολο. Αυτόνομη ρομποτική μετακίνηση είναι ένα σημαντικό τεχνολογικό εμπόδιο για πολλούς τομείς της ρομποτικής, όπως τα ανθρωποειδή (Asimo της Honda).

Η Τροχοφόρα μετακίνηση είναι μια ενδιαφέρουσα ιδέα που συγχωνεύει την έμπνευση από τεχνητούς τροχούς και το βιολογικό σύστημα που αποτελεί τη βάση για την ανάπτυξη ρομπότ που κινούνται με άκρα. Ωστόσο, η επαφή με το περιβάλλον και η μορφολογία του εδάφους είναι δύο βασικοί παράγοντες στην πρόκληση αυτών των μηχανισμών.

Τα παρακάτω είναι χαρακτηριστικά του τύπου του περιβάλλοντος:

- Το μέσο (π.χ., μαλακά ή σκληρά εδάφη, αέρας ή νερό)
- δομή του μέσου (π.χ., τραχύ ή επίπεδη σε περίπτωση σκληρού εδάφους).

Τα χαρακτηριστικά της επαφής με το έδαφος είναι:

- τριβής μεταξύ της επιφάνειας και το ρομπότ
- γωνία επαφής στο έδαφος τύπου σημείου επαφής (π.χ., το αποτύπωμα σε περίπτωση μετακίνησης πόδια)

τα χαρακτηριστικά σταθερότητας είναι:

- κλίση του εδάφους
- το κέντρο βάρους του ρομπότ
- Γεωμετρία και ο αριθμός των σημείων επαφής

Η τροχοφόρα μετακίνηση είναι ο πιο δημοφιλής μηχανισμός που έχει ενσωματωθεί στο σχεδιασμό και την ανάπτυξη των αυτόνομων μετακινούμενων ρομπότ, με τον κύριο λόγο να είναι ότι η δυνατότητα αυτή είναι αποτελεσματική και με απλό έλεγχο. Η ισορροπία δεν είναι ένα πρόβλημα με αυτό του είδους την κίνηση καθώς οι τροχοί είναι συνέχεια σε επαφή με το έδαφος. Επαρκής ισορροπία είναι εγγυημένη σε περίπτωση δίκυκλων και τρίκυκλων ρομπότ. Ωστόσο, ένα σύστημα ανάρτησης είναι απαραίτητο για την ισορροπία σε ρομπότ με περισσότερες από δύο τροχούς. Θέματα ευελιξία, ελέγχου και τριβής (έλξης) του τροχοφόρου ρομπότ είναι πολύ πιο σημαντικά να επιλυθούν από πρόβλημα της ισορροπίας επειδή ένα τροχοφόρο ρομπότ πρέπει να είναι ικανό να διατηρεί πλήρη λειτουργική ικανότητα κατά την εκτέλεση της αποστολής του σε όλες της συνθήκες και την ποικιλομορφία του εδάφους.

Παλινόστηση στην ρομποτική μπορεί να οριστεί το πρόβλημα να βρεθεί η διαδρομή που επιτρέπει σε ένα ρομπότ να επιστρέψει στην αρχική, «σπίτι», θέση στο χώρο, μετά την εκτέλεση μιας τυχαίας διαδρομής. Πληροφορίες οδομετρίας έχουν χρησιμοποιηθεί παραδοσιακά για την επίλυση αυτού του προβλήματος, αλλά έχει αποδειχθεί να είναι

μια αναξιόπιστη πηγή. Αλλή μέθοδος είναι η οπτική παλινόστησης βάσει μια πανοραμικής κάμερας.

Στα αυτοκινούμενα και ανταγωνιστικά ρομπότ, η χειραγώγηση αντικείμενου είναι ένας μηχανισμός που επιτρέπει σε ένα ρομπότ να αλληλοεπιδράει με αντικείμενα στο περιβάλλον του. Υπάρχουν τρεις βασικές κατηγορίες χειραγώγησης αντικειμένων: άροτρα, σκούπες και συγκράτηση τριβής. Η πρώτη κατηγορία χειραγώγηση αντικείμενου, τα **Άροτρα (Plows)** στην οποία ασκείται μια ενιαία δύναμη στην πλευρά του αντικείμενου. Η μετακίνηση αντικειμένων τύπου άροτρα χωρίς στην πραγματικότητα να τα σηκώνει και είναι μακράν ο ευκολότερος τύπος χειρισμού. Η δεύτερη κατηγορία χειραγώγησης αντικείμενου, **την Σκούπα - Σέσουλα (Scoops)**, εφαρμόζεται μια δύναμη κάτω από ένα αντικείμενο έτσι ώστε να ανασηκωθεί και μετά να μπορεί μετακινηθεί. Μόλις ένα αντικείμενο είναι μέσα σε μια σέσουλα, τότε αυτό μπορεί να ανυψωθεί ή να χαμηλώσει βασιζόμενη στην βαρύτητα και γίνεται με τέτοιο τρόπο για να διατηρείτε το αντικείμενο μέσα στη σέσουλα. Στην τρίτη κατηγορία χειραγώγησης αντικείμενου, την **συγκράτηση μέσω τριβής (Friction Grabbers)**, εφαρμόζεται μια δύναμη σε ένα αντικείμενο σε δύο τουλάχιστον σημεία, έτσι ώστε το αντικείμενο να είναι τσιμπημένο ή αρπαγμένο. Η συγκράτηση μέσω τριβής συγκρατεί τα αντικείμενα με ασφάλεια και είναι γενικά ο τύπος χειρισμού που παρέχει τον καλύτερο έλεγχο στα αντικείμενα. Η πιο κοινή μορφή εξαρτήματος για αυτού του τύπου χειραγώγησης είναι ένα νύχι τσιμπήματος (συγκράτησης).

## ΚΕΦΑΛΑΙΟ 3

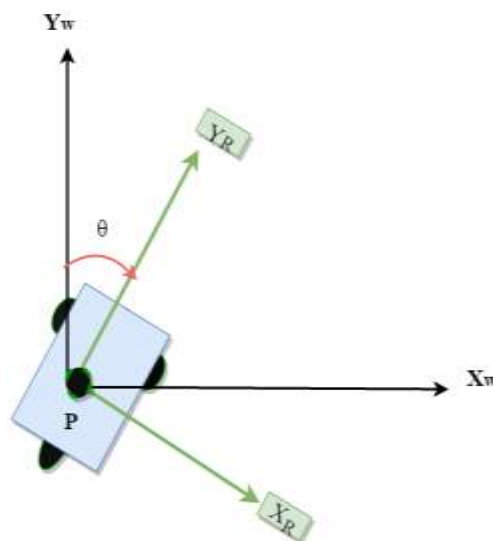
### “Έλεγχος έντροχου ρομποτικού οχήματος με χρήση ασαφής λογικής

#### 3.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται ανάλυση του μοντέλου κίνησης ενός οχήματος που έχει διαφορική κίνηση. Στη συνέχεια παρουσιάζεται ο πρώτος ασαφής ελεγκτής που αρχικά αναπτύχθηκε για οδήγηση ενός ρομποτικού διαφορικού οχήματος. Μετά έγινε προσομοίωση με την χρήση του λογισμικού V-ger για την πιστοποίηση καλής λειτουργίας. Προσομοιώνουμε το πραγματικό ρομποτικό όχημα τύπου Pioneer.

#### 3.2 Κινηματικό μοντέλο ρομποτικού οχήματος διαφορικής κίνησης [62]

Για να προσδιοριστεί η θέση ενός ρομποτικού οχήματος σε ένα δισδιάστατο επίπεδο, είναι απαραίτητο να προσδιοριστεί μια σχέση μεταξύ του παγκόσμιου (global) πλαισίου αναφοράς του επιπέδου και του σταθερού πλαισίου του ρομπότ, όπως φαίνεται στην εικόνα 39.



Εικόνα 39: Θέση και προσανατολισμός κινούμενου ρομπότ στο επίπεδο [62]

Για να καθορίσουμε τη θέση που βρίσκεται το όχημα σε σχέση με το παγκόσμιο πλαίσιο αναφοράς  $O: \{X_i, Y_i\}$ , θεωρούμε ένα σημείο  $P$  στο σταθερό πλαίσιο ως σημείο αναφοράς για τη θέση του ρομπότ σε σχέση με το τοπικό πλαίσιο αναφοράς του  $\{X_R, Y_R\}$ . Η θέση του σημείου  $P$  ως προς το παγκόσμιο πλαίσιο αναφοράς μπορεί να καθοριστεί από τις συντεταγμένες  $x$  και  $y$ , και η γωνιακή διαφορά μεταξύ του παγκόσμιου και του τοπικού πλαισίου αναφοράς δίνεται από τη γωνία  $\theta$ .

Επομένως η θέση και ο προσανατολισμός του ρομπότ δίνεται από το διάνυσμα:

$$\xi_1 = [x \ y \ \theta]^T \quad (3.1)$$

Τα δύο πλαίσια συσχετίζονται με τον πίνακα στροφής:

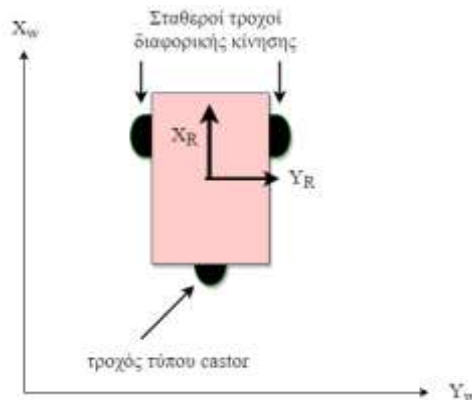
$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

Για τη θέση και την ταχύτητα κίνησης του ρομπότ εκφρασμένες στο παγκόσμιο σύστημα, σύμφωνα με τις σχέσεις 3.1 & 3.2 θα ισχύει :

$$\xi_R = R(\theta) \cdot \xi_1 = R(\theta) \cdot [x \ y \ \theta]^T \quad (3.3)$$

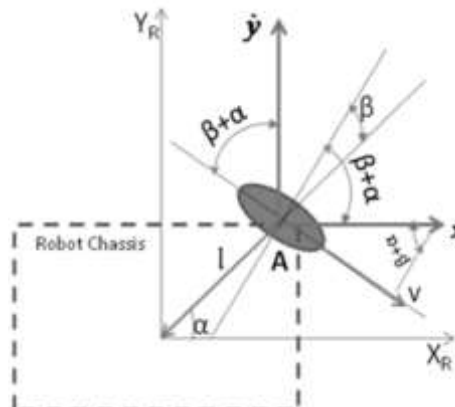
$$\dot{\xi}_R = R(\theta) \cdot \dot{\xi}_1 = R(\theta) \cdot [\dot{x} \ \dot{y} \ \dot{\theta}]^T \quad (3.4)$$

Υπάρχουν διάφοροι τρόποι με τους οποίους μπορεί να κινηθεί ένα έντροχο ρομπότ ανάλογα με τη διαμόρφωση που έχουν οι τροχοί του και εάν έχει κατευθυντήριο τροχό ή όχι. Συγκεκριμένα για την περίπτωση της διαφορικής κίνησης έχουμε δύο κινητήριους σταθερούς τροχούς και έναν τροχό τύπου caster όπως φαίνεται στην εικόνα 40, ο σταθερός τροχός δεν έχει κατακόρυφο άξονα περιστροφής για την οδήγηση, η γωνία του ως προς το σταθερό πλαίσιο είναι σταθερή και η κίνησή του περιορίζεται σε εμπρός και πίσω κατά μήκος του επιπέδου του τροχού και σε περιστροφή γύρω από το σημείο επαφής του με το επίπεδο του εδάφους.

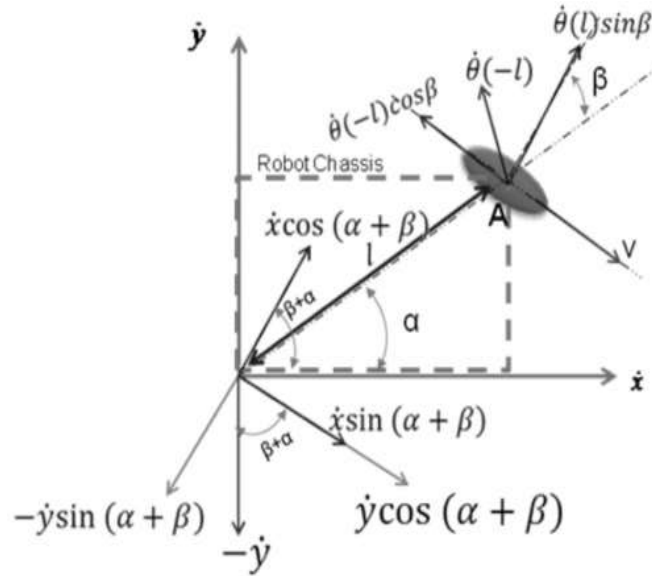


Εικόνα 40 : Ρομποτικό όχημα με διαφορική κίνηση [62]

Στην εικόνα 41 απεικονίζεται ένας σταθερός τροχός και η θέση του είναι εκφρασμένη σε πολικές συντεταγμένες με απόσταση  $l$  και γωνία  $\alpha$  από την αρχή αξόνων του πλαισίου αναφοράς του σταθερού πλαισίου του ρομποτικού οχήματος. Η γωνία του επιπέδου του τροχού σε σχέση με το σταθερό πλαίσιο συμβολίζεται με  $\beta$ .



Εικόνα 41 : Οι γωνίες που σχηματίζει ο τροχός με το πλαίσιο συντεταγμένων του σταθερού πλαισίου[63]



Εικόνα 42: 4 Ανάλυση των διανυσμάτων ταχυτήτων του τροχού [63]

Ενώ ένα διαφορικό ρομπότ είναι σε κίνηση, δύο περιορισμοί είναι δυνατόν να συμβαίνουν. Ο πρώτος περιορισμός επιβάλλεται από την επαφή κύλισης και ο δεύτερος από την πλευρική ολίσθηση. Σκοπός είναι να αποφευχθεί η πλευρική ολίσθηση, ενώ το ρομπότ είναι σε κίνηση.

Ο περιορισμός κύλισης για αυτόν τον τροχό επιβάλλει ότι όλες οι κινήσεις κατά μήκος της διεύθυνσης του επιπέδου του τροχού πρέπει να συνοδεύονται από την κατάλληλη ποσότητα περιστροφής έτσι ώστε να υπάρχει καθαρή κύλιση στο σημείο επαφής:

$$\dot{x}_R \sin(\alpha + \beta) - \dot{y}_R \cos(\alpha + \beta) - \dot{\theta}_R l \cos(\beta) = \dot{\varphi} r \quad (3.5)$$

Όπου  $\dot{\varphi}$ ,  $r$ , η γωνιακή ταχύτητα και η ακτίνα του τροχού. Ο περιορισμός ολίσθησης για αυτόν τον τροχό επιβάλλει ότι η συνιστώσα της κίνησης του τροχού που είναι κάθετη προς το επίπεδο του τροχού πρέπει να είναι μηδέν:

$$\dot{x}_R \sin(\alpha + \beta) - \dot{y}_R \cos(\alpha + \beta) - \dot{\theta}_R l \cos(\beta) = 0 \quad (3.6)$$

Στα διαφορικά κινούμενα ρομπότ για τον αριστερό τροχό ισχύει ότι,  $\alpha = 90^\circ$ ,  $\beta = 0^\circ$  και  $l = b/2$ , οπότε :

$$\dot{x}_R \sin(90) - \dot{y}_R \cos(90) - \dot{\theta}_R l \cos(0) = \dot{\varphi}_1 r \quad (3.7) \text{ και}$$

$$\dot{x}_R \sin(90) - \dot{y}_R \cos(90) - \dot{\theta}_R l \cos(0) = 0 \quad (3.8)$$

Λύνοντας το σύστημα των δύο εξισώσεων προκύπτει ότι:

$$\dot{x}_R - \dot{\theta}_R b/2 = \dot{\varphi}_1 r \quad (3.9) \text{ και } \dot{y}_R = 0 \quad (3.10)$$

Για τον δεξιό τροχό ισχύει ότι,  $\alpha = -90^\circ$ ,  $\beta = 0^\circ$  και  $l = b/2$ , οπότε :

$$\dot{x}_R - \dot{\theta}_R b/2 = \dot{\varphi}_2 r \quad (3.11) \text{ και } \dot{y}_R = 0 \quad (3.12)$$

λύνοντας ως προς  $\dot{x}_R$  και  $\dot{y}_R$  παίρνουμε το κινηματικό μοντέλο του ρομπότ όπου:

$$\dot{x}_R = (\dot{\varphi}_1 + \dot{\varphi}_2) r/2 \quad (3.13) \quad \dot{y}_R = 0 \quad (3.14) \quad \dot{\theta}_R = (\dot{\varphi}_2 - \dot{\varphi}_1) r/b \quad (3.15) \quad [63]$$

### 3.3 Ασαφής έλεγχος ρομποτικού οχήματος διαφορικής κίνησης.

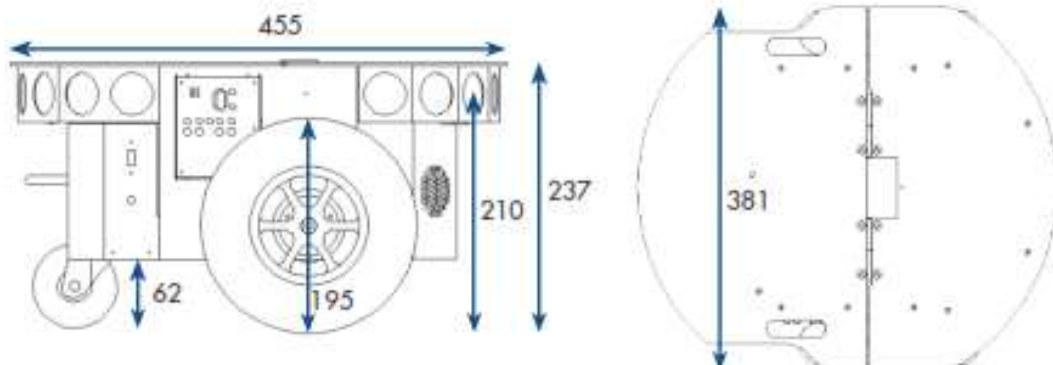
Στα πλαίσια της εργασίας αρχικά σχεδιάστηκε ένας ασαφής ελεγκτής τύπου Mamdani που ήταν σε θέση να οδηγήσει το ρομποτικό όχημα προς ένα σημείο στόχο με παράλληλη δυνατότητα αποφυγής εμποδίων (σταθερών ή κινητών). Για την πιστοποίηση λειτουργίας χρησιμοποιήθηκε το προσομοιωμένο μοντέλο του έντροχου ρομποτικού οχήματος Pioneer 3rdx (εικόνα 43).



Εικόνα 43 : Το ρομποτικό όχημα Pioneer 3-DX

Το Pioneer 3-DX είναι ένα μικρό ελαφρύ ρομποτικό όχημα διαφορικής κίνησης με δύο τροχούς και δύο κινητήρες. Είναι ιδανικό για χρήση σε εσωτερικούς χώρους εργαστηρίου ή σε χώρους μιας τάξης. Το ρομποτικό όχημα φέρει μπροστινό σόναρ, μια μπαταρία, encoder στους τροχούς, ένα μικροελεγκτή με firmware ARCOS, και το πακέτο ανάπτυξης λογισμικού Pioneer SDK για αυτοκινούμενα ρομπότ. Τα ερευνητικά ρομπότ τύπου Pioneer είναι τα πιο δημοφιλή έξυπνα κινητά ρομπότ στον κόσμο για την εκπαίδευση και την έρευνα. Η Ευελιξία τους, αξιοπιστία και αντοχή τους τα έχει καταστήσει την προτιμώμενη πλατφόρμα για προηγμένη ευφυή ρομποτική. Τα Pioneer είναι προ-συναρμολογημένα, προσαρμόσιμα, αναβαθμίσιμα, και αρκετά αξιόπιστο να διαρκέσει για χρόνια στα εργαστηρίου και την χρήση της διδασκαλίας.

#### Dimensions (mm)



Core Software - included with all research platforms

Εικόνα 44: Όχημα Pioneer 3-DX Διαστάσεις

|                               |  |  |
|-------------------------------|--|--|
| <i>Κατασκευή</i>              | Σώμα κατασκευής                        | 1.6 mm Αλουμινίου (Επικαλυμμένο).                      |
|                               | Ρόδες                                  | Λάστιχο γεμισμένο με αφρό                              |
| <i>Λειτουργία</i>             | Βάρος Ρομπότ                           | 9 Kg   |
|                               | Ωφέλιμο φορτίο λειτουργίας             | 17 Kg  |
| <i>Διαφορική Κίνηση</i>       | Ακτίνα στροφής κίνησης                 | 0 cm   |
|                               | Ακτίνα ταλάντευσης                     | 26.7 cm  |
|                               | Μέγιστη Ταχύτητα εμπρός/πίσω           | 1.2 m/s  |
|                               | Ταχύτητα περιστροφής                   | 300 °/s  |
|                               | Μέγιστο βήμα Μετάβασης                 | 2.5 cm   |
|                               | Μέγιστο εγκάρσιο χάσμα μετάβασης       | 5 cm Max   |
|                               | Μέγιστος βαθμός μετάβασης              | 25%  |
|                               | Επιφάνεια Μετάβασης:                   | εσωτερικός, αναπηρικών καρεκλών                        |
| <i>Ισχύς</i>                  | Χρόνος εκτέλεσης                       | 8-10 ώρες / 3 μπαταρίες (χωρίς τα εξαρτήματα)          |
|                               | Χρόνος Φόρτισης                        | 12 ώρες (πρότυπα)                                      |
|                               |  | 2.4 ώρες (προαιρετικός υψηλής χωρητικότητας φορτιστής) |
|                               | Διαθέσιμες παροχές ηλεκτρικού ρεύματος | 5 v @ 1.5 A (με διακόπτη)                              |
| 12 v @ 2.5 A (με διακόπτη)    |  |  |
| <i>Μπαταρίες</i>              | Υποστηρίζουν                           | μέχρι 3  |
|                               | Τάση                                   | 12 V   |
|                               | Χωρητικότητα                           | 7.2 Ah (κάθε μια)                                      |
| <i>Μικροελεγκτής I/O</i>      | Σειριακό σύστημα                       |  |
|                               | 32 ψηφιακές είσοδοι                    |  |
|                               | 8 ψηφιακές εξόδους                     |  |
|                               | 7 αναλογική είσοδοι                    |  |
|                               | 3 σειριακές πόρτες επέκτασης           |  |
|                               | MIDI προγραμματιζόμενη piezo σειρήνα   |  |
| <i>Πίνακας ελέγχου Χρήστη</i> | Ένδειξη κύρια ισχύος                   |  |
|                               | Δείκτης φόρτισης μπαταρίας             |  |
|                               | 2 Εφεδρικούς διακόπτες τροφοδοσίας     |  |
|                               | Επαναφορά συστήματος                   |  |
|                               | Μπουτόν Ενεργοποίηση Κινητήρα          |  |

Πίνακας 2: Στοιχεία ρομποτικού οχήματος Pioneer 3-DX

### 3.3.1 Προσομοίωση της λειτουργίας του οχήματος διαφορικής οδήγησης με χρήση MatLab και V-rep

Προκειμένου να προσομοιώσουμε τον τρόπο λειτουργίας ενός ρομποτικού οχήματος επιλέχθηκε η λύση του λογισμικού MatLab και Vrep. Το MatLab χρησιμοποιήθηκε για :

- για την δημιουργία του ελεγκτή ασαφούς λογικής
- Έλεγχο του ρομπότ που προσομοιώνουμε (σε συνεργασία με πρόγραμμα προσομοίωσης ρομπότ).

Ενώ το V-rep χρησιμοποιήθηκε για :

- Προσομοίωση του τροχήλατου Ρομπότ με την χρήση κώδικα ελέγχου μέσω του λογισμικού MatLab.

Στην συνέχεια παρουσιάζουμε τα βασικά χαρακτηριστικά του Vrep καθώς και τον τρόπο επικοινωνίας του με το MatLab.

### 3.3.2 V-rep, Virtual robot experimentation platform

Το Vrep είναι ένας ισχυρός τρισδιάστατος προσομοιωτής ρομπότ, που διαθέτει αρκετές ενότητες με ευέλικτους υπολογισμούς (Αντίστροφη κινηματική, φυσικής/δυναμικής, ανίχνευση σύγκρουσης, υπολογισμός ελάχιστης απόστασης, σχεδιασμός διαδρομή, κ.λπ.), μια αρχιτεκτονική κατανεμημένων ελέγχων (δυνατότητα απεριόριστου αριθμού ρουτίνων ελέγχου, διαφορετικών τύπων) και διάφορους μηχανισμούς επέκταση (πρόσθετα - plug-ins, εφαρμογή προσαρμοσμένου προγράμματος-πελάτη, κλπ.).

Προσφέρει ένα πλήθος λειτουργιών που μπορούν εύκολα να ενσωματωθούν και να συνδυάζονται μέσα από λειτουργίες API και δέσμη ενεργειών. Το V-REP είναι το τέλειο εργαλείο για την δημιουργία γρήγορων πρωτότυπων και να γίνει επαλήθευση, απομακρυσμένη παρακολούθηση, γρήγορη ανάπτυξη αλγορίθμου, εκπαίδευση στηριζόμενη στην Ρομποτική και προσομοίωση βιομηχανικών συστημάτων αυτοματισμού. Έχει αναπτυχθεί ένας ισχυρός μηχανισμός που επιτρέπει την εύκολη πρόσβαση σε λειτουργίες V-REP API από εξωτερικές εφαρμογές (π.χ. ρομπότ): το απομακρυσμένο API. Αυτό μπορεί να συνδεθεί με Python, Java, Matlab, Urbi. Το λογισμικό είναι διαθέσιμο για PC, Mac και Linux πλατφόρμες. Το Vrep προσφέρει ένα μεγάλο εύρος δυνατοτήτων που του επιτρέπουν να:

**Αλληλεπίδραση κατά τη διάρκεια της προσομοίωσης.** Η αλληλεπίδραση σε μια προσομοίωση της λειτουργίας είναι δυνατή ανά πάσα στιγμή, τα μοντέλα, μαζί με τα στοιχεία τους που σχετίζεται με συμπεριφορά (δηλαδή scripts) μπορούν να μετατοπιστούν, περιστρέφουν, αντιγραφή/επικόλληση, διαγραφή, κλιμάκωση, κλπ. χωρίς να χρειάζεται να ρυθμίσετε οποιονδήποτε κώδικα.

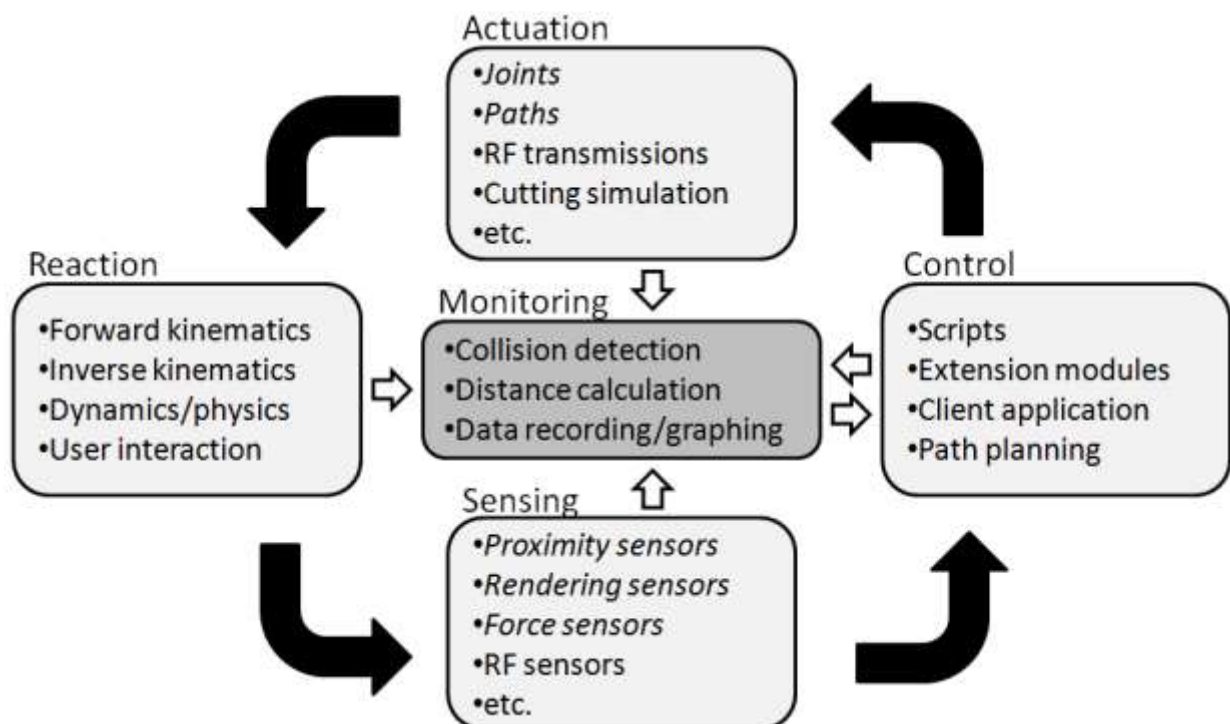
Απεριόριστος αριθμός πλήρως προσαρμόσιμων από τον χρήστη στοιχείων με ενσωματωμένη κατάσταση επεξεργασίας. Προσομοίωση της επιφάνειας κοπής χρησιμοποιώντας διάφορα εργαλεία κοπής. Προσομοίωση αισθητήρων ισχύος/ροπής, ικανό να καταγράφει τις εφαρμοζόμενες δυνάμεις ή ροπές και υπό όρους το σπάσιμο σε τμήματα. Διάφοροι τύποι αρθρώσεων: revolute αρθρώσεις, πρισματικές αρθρώσεις, βίδες και σφαιρικές αρθρώσεις. Διαδρομή ή τροχιά λειτουργικότητας για την επεξεργασία κινηματικών κινήσεων. Στην απλή σκηνή, είναι διαθέσιμες διάφορες λειτουργίες ολοκληρωμένης επεξεργασίας: τρίγωνο, vertex, άκρη, διαδρομή/τροχιάς, και προσωποποιημένη λειτουργία επεξεργασίας. Απαιτείται ελάχιστος ή καθόλου προγραμματισμός για απλές προσομοιώσεις Μοντέλο με δυνατότητα αυτό-επανάληψης ή αυτοκαταστροφής. Πρόγραμμα περιήγησης των μοντέλων με λειτουργικότητα drag-and-drop.

Ισχυρό API με περισσότερες από 280 διακριτές και λεπτομερείς λειτουργίες, διαθέσιμες σε C/C++ και Lua (και σε άλλες γλώσσες προγραμματισμού με ικανότητα κλήσης API επίσης να είναι χρησιμοποιήσιμο). Υποστηριζόμενες μορφές δεδομένων CAD: DXF, 3DS, OBJ και STL. Πλήρως προσαρμόσιμος προσομοιωτής, σε διάφορα επίπεδα, μέσω της δυνατότητας να γραφεί ένα προσαρμοσμένο πρόγραμμα-πελάτη ή με το γράψιμο κάποιων πρόσθετων.



Συμπαγές και ελαφριά εφαρμογή και αρχεία: V-REP είναι μεγέθους 5MB, χωρίς να απαιτείται κάποια εγκατάσταση έξτρα στοιχείων. Σκηνές προσομοίωση και μοντέλα αποθηκεύονται ως ένα ενιαίο συμπιεσμένο αρχείο, η ανάκτηση των εργασιών γίνεται πολύ γρήγορα. Πλήρης και χαρακτηρισμένη προβολή της ιεράρχησης των αντικειμένων, αναφέροντας τα ονόματα και τύπους, δέσμες ενεργειών ελέγχου, περατώσεις βρόχων, επιλογή και ορατότητα μελών, προειδοποιήσεις, κλπ. Πλήρως προσαρμόσιμη προβολή διαμορφώσεις, με 8 σελίδες με ταυτότητα, όπου καθένας μπορεί να προσαρμόσει με απεριόριστο αριθμό όψεων. Παρέχεται μια δωρεάν έκδοση του προγράμματος αναπαραγωγής V-REP, επιτρέποντας να τρέχει και να αλληλοεπιδρά με σκηνές ή μοντέλα που έχουν δημιουργηθεί σε προηγούμενες εκδόσεις. Υπάρχει η δυνατότητα να τρέχουν τέσσερις προσομοιώσεις σε παράλληλη εκτέλεση και να μοιράζονται το ίδιο αντίγραφο του buffer. Προσομοίωση και οπτικοποίηση των ασύρματων επικοινωνιών, με δωρεάν προσδιορισμό του εύρους εκπομπής, την κατεύθυνση και το σχήμα αυτών.

Εκτός από τις λειτουργικές μονάδες δυναμικής ή φυσικής που ενεργούν άμεσα σε όλα τα δυναμικά αντικείμενα μέσα στην σκηνή λειτουργίας, υπάρχουν και άλλες ενότητες υπολογισμών που απαιτούν τη εκτέλεση κάποιων εργασιών υπολογισμού ή υπολογισμών σε αντικείμενα, που καθορίζουν ποια αντικείμενα στην λειτουργική μονάδα θα λειτουργήσουν και πώς. Εάν για παράδειγμα ο χρήστης επιθυμεί να έχει την ελάχιστη απόσταση μεταξύ δύο σχημάτων A και B να υπολογίζεται αυτόματα και ίσως και να καταγράφεται, τότε ένα αντικείμενο ελάχιστης απόσταση πρέπει να οριστεί, έχοντας ως παραμέτρους τα σχήματα A και B. Η εικόνα 45 δείχνει μια τυπική προσομοίωση βρόχου του Vrep, συμπεριλαμβανομένων των αντικειμένων της κύριας σκηνής και ενότητες υπολογισμού.



Εικόνα 45 : Τυπικός βρόχος λειτουργίας μιας V-rep σκηνής [64]

### 3.3.2.1 Vrep 's Μεθοδολογία Ελέγχου

Το Vrep μας δίνει τη δυνατότητα μέσω διαφόρων τρόπων του έλεγχου των προσομοιώσεων. Το V-REP αποτελείται από μια βιβλιοθήκη συναρτήσεων και απαιτεί μια εφαρμογή διακομιστή-πελάτη (server-client) να εκτελεστεί. Η εφαρμογή-πελάτη που έχει ως προεπιλογή το V-REP είναι αρκετά απλή και φροντίζει να φορτωθούν οι ενότητες επέκτασης, να καταγράφει επιστροφές συμβάντων (ή επιστροφές μηνυμάτων), τοποθετώντας σε λειτουργικές μονάδες που είναι είδη φορτωμένες στην μνήμη του συστήματος, προετοιμάζει τον προσομοιωτή και χειρίζεται τις αιτήσεις και την προσομοίωση του βρόχο.

Επιπλέον προσαρμοσμένες λειτουργίες στην προσομοίωση μπορούν να προστεθούν μέσω:

1. Ρουτίνων στη γλώσσα Lua που είναι μια ελαφριά γλώσσα προγραμματισμού και έχει σχεδιαστεί για να υποστηρίξει τον διαδικαστικό προγραμματισμό. Ο μεταγλωττιστής δεσμών ενεργειών Lua είναι ενσωματωμένος στο V-REP, και έχει την δυνατότητα να επεκτείνεται με αρκετές εκατοντάδες ειδικές για το V-REP εντολές. Οι δέσμες ενεργειών είναι το κύριο στοιχείο ελέγχου για τον μηχανισμό προσομοίωσης του V-REP.
2. Ενότητες επέκτασης για το V-REP (plugins). Οι ενότητες επέκτασης επιτρέπουν την εγγραφή και τον χειρισμό μέσω προσαρμοσμένων εντολών. Μια εντολή δέσμης ενεργειών υψηλού επιπέδου (π.χ., `οχημαΜετακινησουΚαιΑπεφυγεΑντικειμενα-robotMoveAndAavoidObstacles (duration)`), τότε μια ρουτίνα μπορεί να χειριστεί αυτήν την υψηλού επιπέδου εντολή εκτελώντας την αντίστοιχη λογική ενώ μια χαμηλού επιπέδου API συνάρτηση να καλείται σε μια γρήγορη και κρυφό μονάδα.

### 3.3.2.2 Η Αρχιτεκτονική ελέγχου στο V-REP.

Η αρχιτεκτονική ελέγχου στο V-Rep φαίνεται στην εικόνα 46. (Οι περιοχές που είναι σκιασμένο με γκριζό χρώμα μπορούν να προσαρμοστούν από το χρήστη).

Η εφαρμογή πελάτη (1) μέσω ρουτινών σε C/C++ API μπορεί να πραγματοποιούν κλήσεις στο Vrep ή από ενότητες επέκτασης.

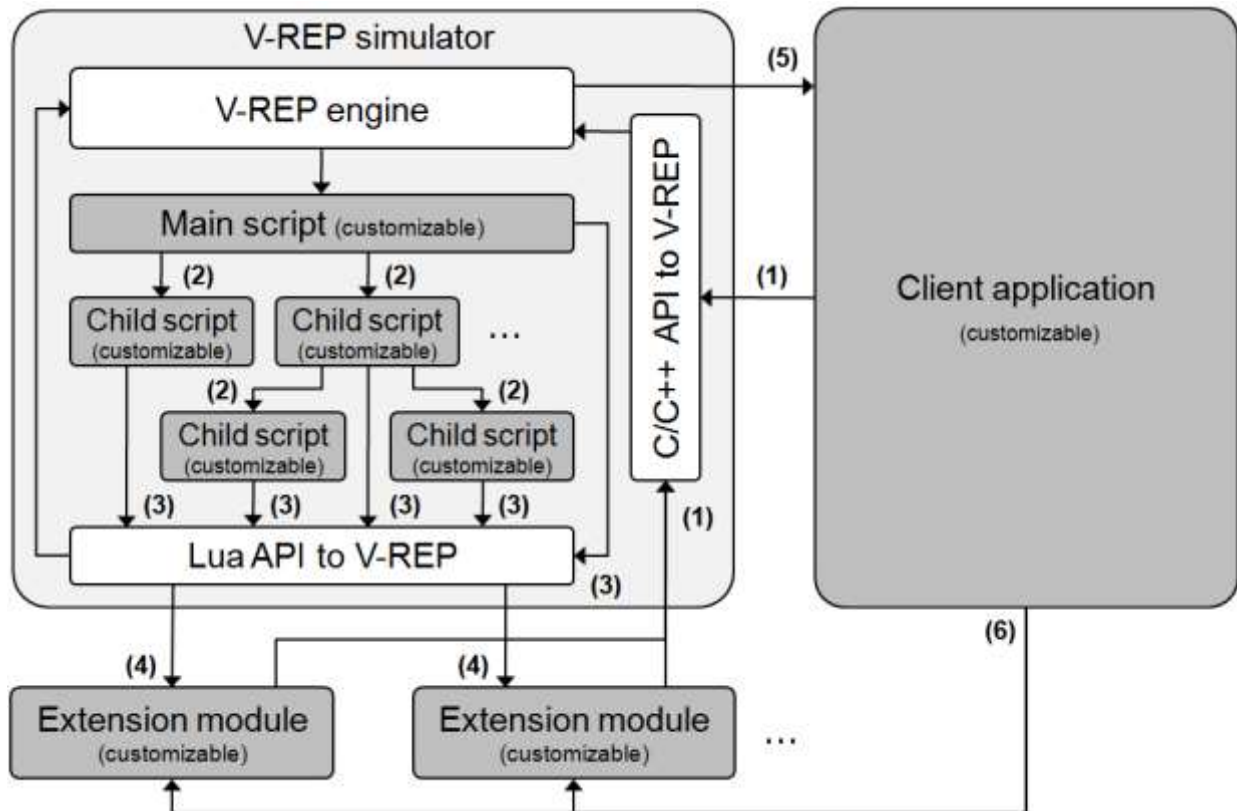
Σενάριο χειρισμών κλήσεων.(2). Συνήθως `simHandleChildScript` (sim διαχειρίσου τα πάντα). Εκτελεί όλες τις υπόλοιπες υπορουτίνες στην τρέχουσα ιεραρχία.

Κλήσεις των εντολών Lua API (3) μέσα στο V-REP από δέσμες ενεργειών.

Επιστροφή δεδομένων από κλήσεις σε ενότητες επέκτασης.(4) Προέρχονται όταν μια δέσμη ενεργειών καλεί μια προσαρμοσμένη συνάρτηση, που είχαν προηγουμένως ταξινομηθεί από μια λειτουργική μονάδα επέκτασης.

Επιστροφή δεδομένων (5) από κάποιο γεγονός στην εφαρμογή-πελάτη.

Αναμεταδίδονται κλήσεων συμβάντων (6) προς την ενότητα επέκτασης.



Εικόνα 46 : Αρχιτεκτονική Ελέγχου στο V-rep.[64]

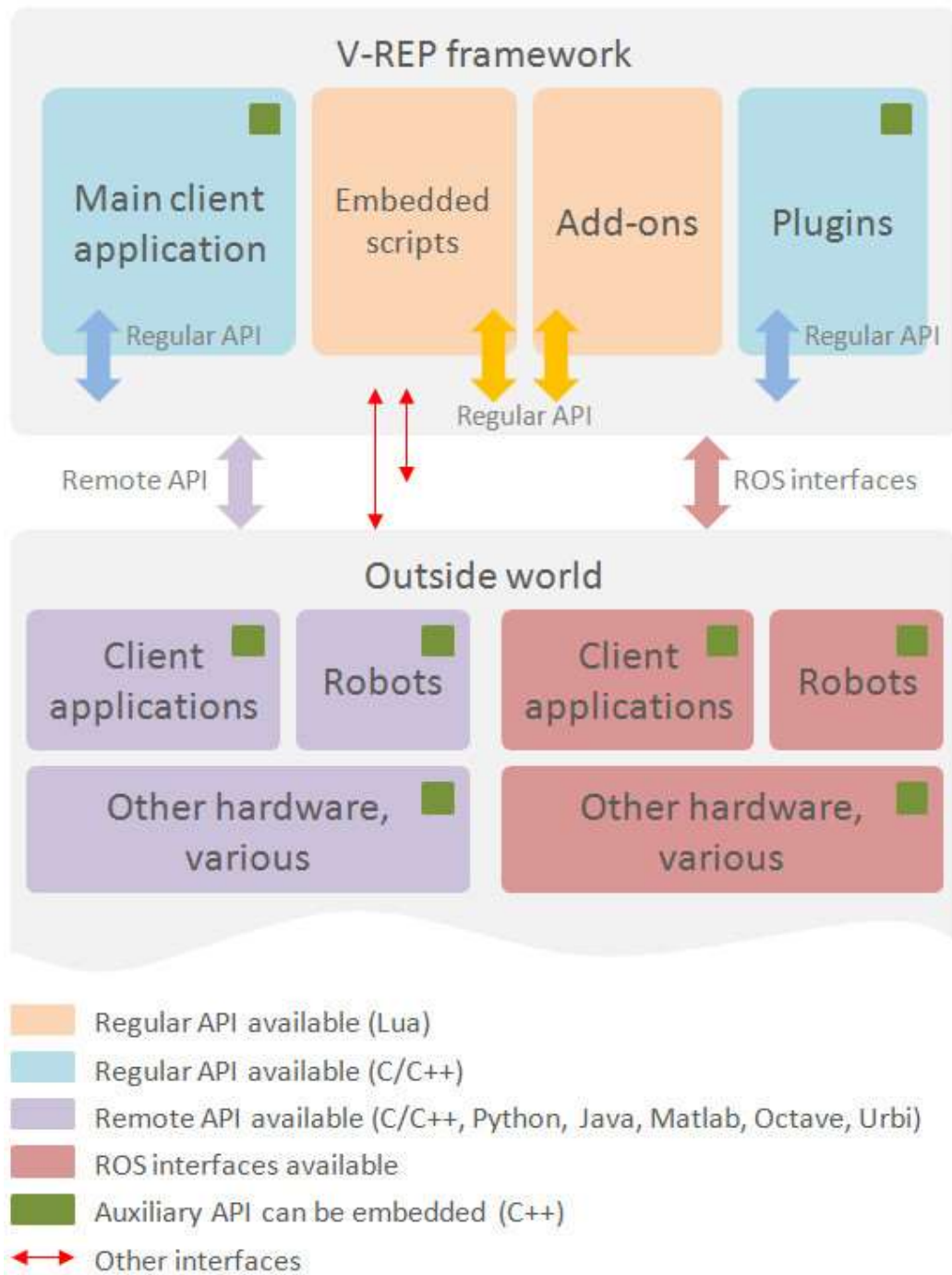
### 3.3.2.3 V-REP Περιβάλλον Προγραμματισμού - Τρόπος λειτουργίας

Το πλαίσιο λειτουργίας V-REP API ομαδοποιεί όλες τις διασυνδέσεις γύρω από το V-REP.

Παρέχεται σε 4 + 1 διαφορετικές εκδόσεις:

1. Το κανονικό API (**The regular API**)
2. Το απομακρυσμένο API (**The remote API**)
3. Το ROS περιβάλλον (**The ROS interfaces**)
4. Το βοηθητικό API (**The auxiliary API**)
5. Άλλα περιβάλλοντα (**Other interfaces**)

Ενώ η κανονική API μπορεί να προσπελαστεί από μέσα από τον προσομοιωτή (π.χ. από μια ενσωματωμένη δέσμη ενεργειών, ένα πρόσθετο, ένα plugin ή η εφαρμογή κύριου πελάτη), το απομακρυσμένο API και των διεπαφών ROS μπορεί να προσπελαστεί από σχεδόν οποιαδήποτε πιθανή εξωτερική εφαρμογή ή υλικό (συμπεριλαμβανομένων πραγματικά ρομπότ, απομακρυσμένους υπολογιστές, κλπ.). Το βοηθητικό API δεν είναι μια διασύνδεση ως έχει, αλλά περισσότερο μια συλλογή από λειτουργίες ενός βοηθητικού προγράμματος που μπορούν να ενσωματωθούν και να λειτουργήσουν από μόνα τους. Το άλλο περιβάλλον διασυνδέσεις ομαδοποιεί όλες τις πιθανές δυνατότητες για τον χρήστη και επεκτείνει τις διαθέσιμες διασυνδέσεις. Η εικόνα 47 παρουσιάζει μια επισκόπηση των διαφόρων διεπαφών του Vrep.



Εικόνα 47 : Διεπαφές του V-Rep [64]

### 3.3.2.4 Απομακρυσμένο Περιβάλλον Προγραμματισμού (Remote API)

Το V-REP προσφέρει ένα απομακρυσμένο API που επιτρέπει τον έλεγχο μια προσομοίωσης (ή του προσομοιωτή μόνο) από μια εξωτερική εφαρμογή ή απομακρυσμένο υπολογιστικό σύστημα (π.χ. πραγματικό ρομπότ, απομακρυσμένο υπολογιστή, κλπ.).

Το απομακρυσμένο API του V-REP αποτελείται από περίπου εκατό συγκεκριμένες εντολές και μία γενική συνάρτηση, που μπορεί να κληθεί από μια εφαρμογή C/C++, μια δέσμη ενεργειών Python, μια εφαρμογή Java, ένα πρόγραμμα Matlab/Octave, μια δέσμη ενεργειών Urbi ή μια δέσμη ενεργειών Lua. Οι απομακρυσμένες λειτουργίες API αλληλοεπιδρούν με το V-REP μέσω υποδοχών επικοινωνίας με τέτοιο τρόπο που να

μειώνεται η καθυστέρηση και το φόρτο δικτύου. Όλα αυτά συμβαίνουν με ένα κρυφό τρόπο για τον χρήστη.

Το απομακρυσμένο API μπορεί να αφήσει μία ή περισσότερες εξωτερικές εφαρμογές να αλληλοεπιδρούν με το V-REP με έναν σύγχρονο ή ασύγχρονο τρόπο (η προεπιλογή είναι ασύγχρονη), και ακόμη και απομακρυσμένος έλεγχος του προσομοιωτή υποστηρίζεται (π.χ. φόρτωση εξ αποστάσεως μια σκηνης, Έναρξη, παύση ή η διακοπή ενός προσομοίωση για παράδειγμα).

Ο όρος σύγχρονη χρησιμοποιείται με την έννοια ότι κάθε πέρασμα του προσομοίωση τρέχει συγχρονισμένα με την απομακρυσμένη εφαρμογή API (δηλαδή ο προσομοιωτής θα περιμένει για ένα σήμα ενεργοποίησης από έναν υπολογιστή-πελάτη να ξεκινήσει το επόμενο πέρασμα προσομοίωσης σε μια χρονική στιγμή  $t + dt$ ). Αυτό είναι διαφορετικό από τη σύγχρονη/ασύγχρονη κατά την έννοια του αποκλεισμού/μη παρεμπόδιση λειτουργίας. Επίσης, η απομακρυσμένη API υποστηρίζει λειτουργίες αποκλεισμού και μη αποκλεισμού.

Η απομακρυσμένη λειτουργικότητα API έρχεται σε 2 ξεχωριστές οντότητες, που αλληλοεπιδρούν μέσω υποδοχών επικοινωνίας:

1. **Πρόγραμμα-πελάτη** (δηλαδή η εφαρμογή μας): το απομακρυσμένο API στην πλευρά του προγράμματος-πελάτη είναι διαθέσιμο σε πολλές διαφορετικές γλώσσες προγραμματισμού. Προς το παρόν υποστηρίζει τις ακόλουθες γλώσσες: C/C++, Python, Java, Matlab, οκτάβα, Urbi και Lua. Συνδέσεις για άλλες γλώσσες μπορούν να δημιουργηθούν εύκολα από τον προγραμματιστή.
2. **Διακομιστής** (δηλαδή το V-REP): το απομακρυσμένο API στην πλευρά του διακομιστή υλοποιείται μέσω ενός V-REP προσθέτου που θα πρέπει να φορτωθεί από το V-REP ως προεπιλογή. Ανάλογα το API είναι και το αντίστοιχο αρχείο που φορτώνεται.

### 3.3.3 Επικοινωνία Vrep - Matlab μέσω απομακρυσμένου API (Remote API modus operandi)

Στο Vrep μια απομακρυσμένη συνάρτηση API καλείται με παρόμοιο τρόπο όπως μια τακτική συνάρτηση API, ωστόσο έχουμε 2 σημαντικές διαφορές:

1. Σχεδόν όλες οι απομακρυσμένες συναρτήσεις API επιστρέφουν μια παρόμοια τιμή: **έναν κωδικό επιστροφής (a return code)**.
2. Σχεδόν όλες οι απομακρυσμένες συναρτήσεις API απαιτούν δύο επιπλέον ορίσματα:
  - τον τρόπο λειτουργίας (operation mode ) και το
  - clientID (Αναγνωριστικό που επιστρέφεται από τη συνάρτηση simxStart)

Η ανάγκη για τον τρόπο λειτουργίας και ένα συγκεκριμένο κωδικό επιστροφής (return code) προέρχεται από το γεγονός ότι η απομακρυσμένη συνάρτηση API πρέπει να ταξιδέψει μέσω της υποδοχής επικοινωνίας στο διακομιστή (V-REP), να εκτελέσει μια εργασία, και στην συνέχεια, να επιστρέψει στον πελάτη (MatLab). Μια απλή (ή τακτική) προσέγγιση θα ήταν ο πελάτης να στέλνει ένα αίτημα, και να περιμένει μέχρι ο διακομιστής να κάνει επεξεργασία του αιτήματος και να απαντήσει. Στις

περισσότερες περιπτώσεις αυτό θα πάρει πολύ χρόνο και θα απαξιώσει την εφαρμογή του πελάτη.

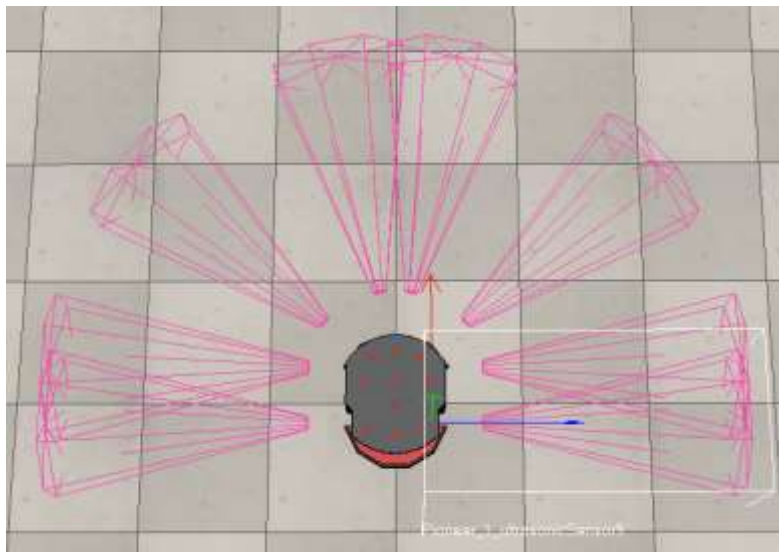
Αντί αυτού, το Remote API επιτρέπει στο χρήστη να επιλέξει τον τύπο της λειτουργίας και τον τρόπο που η προσομοίωση δουλεύει, παρέχοντας τέσσερις βασικούς μηχανισμούς για την εκτέλεση κλήσεων λειτουργίας ή για τον έλεγχο της προόδου προσομοίωσης:

1. Αποκλεισμός κλήσεων λειτουργίας - (Blocking function calls)
2. Κλήσεις λειτουργίας μη αποκλεισμού - (Non-blocking function calls)
3. Ροή δεδομένων - (Data streaming)
4. Σύγχρονη λειτουργία - (Synchronous operation)

Στο παράρτημα Ε, αναλύονται οι μηχανισμοί εκτέλεση κλήσεων λειτουργίας ή για τον έλεγχο της προόδου προσομοίωσης.

### 3.3.4 Το Pioneer στο V-rep

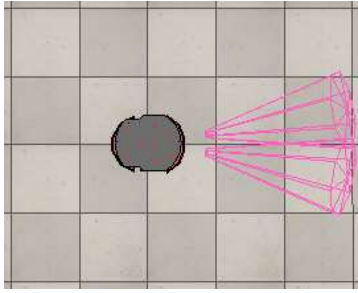
Στο Vrep υπάρχει η δυνατότητα προσομοίωσης του οχήματος Pioneer 3-DX. Το όχημα με τους 8 προσομοιωμένους αισθητήρες υπέρυθρων παρουσιάζεται στην εικόνα 48.



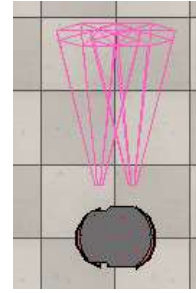
Εικόνα 48 : Αισθητήρες Οχήματος Pioneer 3-DX

Προκειμένου να μειώσουμε την πολυπλοκότητα του εκλεκτής μας, ομαδοποιούμε τους αισθητήρες ανάλογα την θέση τους στις εξής κατηγορίες : (1) Εμπρόσθιοι Αισθητήρες (fr), (2) Αριστεροί Αισθητήρες (lf), (3) Δεξιόι Αισθητήρες (rt), (4) Εμπρόσθιοι Αριστερά (frlf) και (5) Εμπρόσθιοι Δεξιά (frrt).

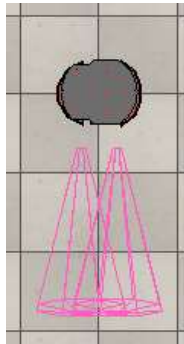
Παρακάτω ακολουθεί η παρουσίαση της θέσης κάθε ομάδας αισθητήρων.



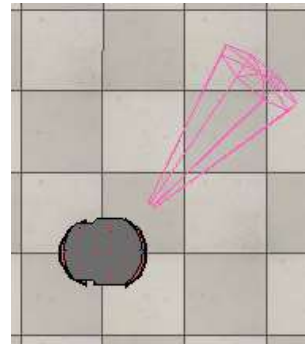
Εικόνα 49: Εικόνα 46 : Εμπρόσθιοι Αισθητήρες (fr)



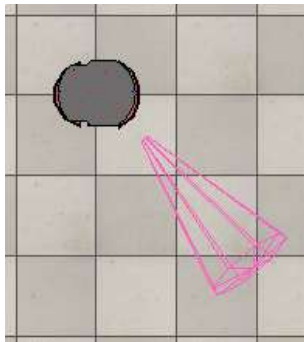
Εικόνα 50: Αριστεροί Αισθητήρες (lf)



Εικόνα 51: Δεξιοί Αισθητήρες (rt)



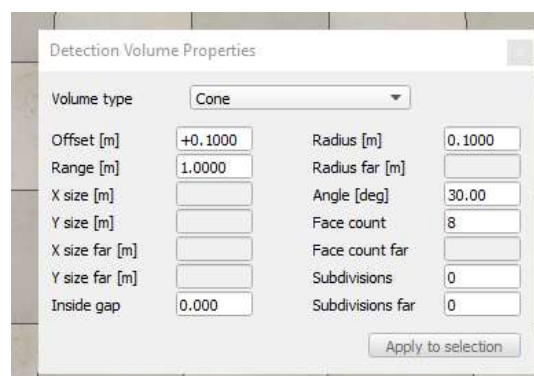
Εικόνα 52: Εμπρόσθιοι Αριστεροί (frlf)



Εικόνα 53: Εμπρόσθιοι Δεξιά (frtr)

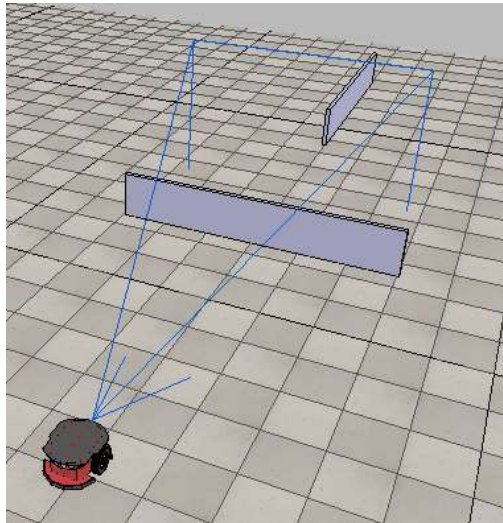
Στο συγκεκριμένο όχημα υπάρχει η δυνατότητα να ενεργοποιηθούν και άλλοι αισθητήρες στις εμπρόσθιες πλευρές και στο οπίσθιο μέρος με σκοπό την πύκνωση του πεδίου εντοπισμού. Οι οπίσθιοι αισθητήρες αφού δεν έγινε χρήση αρνητικής ταχύτητας, δεν ενεργοποιήθηκαν. Τα στοιχεία των αισθητήρων (ίδια για όλους) όπως αυτά δηλώνονται στο λογισμικό Vrep (εικόνα 54) είναι:

- Σχήμα αισθητήρα: κώνος,
- Δραστικό μήκος αισθητήρα: 1 m και
- Γωνία δέσμης 30°.



Εικόνα 54 : Στοιχεία Αισθητήρων Στο Λογισμικό Vrep

Εκτός από τους αισθητήρες υπερήχων υπάρχει και η δυνατότητα ανίχνευσης εμποδίων μέσω οπτικών αισθητήρων (εικόνα 55).



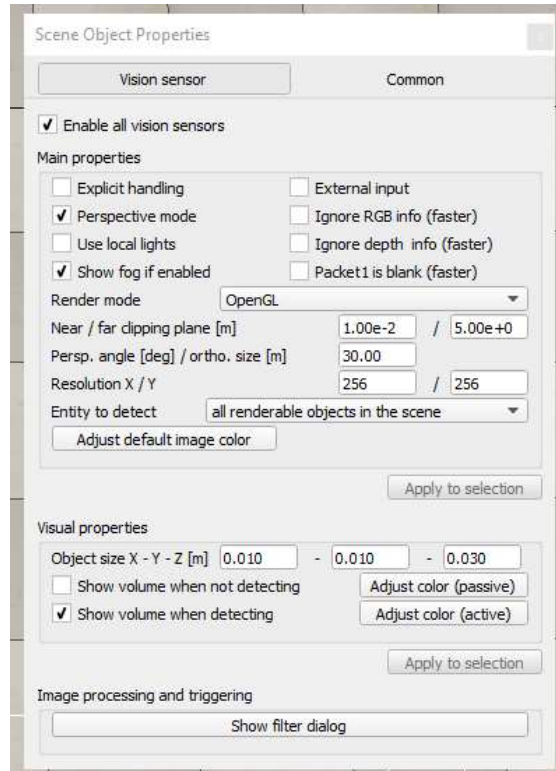
Εικόνα 55 : Οπτικός Αισθητήρες – Κάμερα

Οι παράμετροι της κάμερας είναι :

Αποστάσεις πεδίου ελέγχου : κοντά 0,01m και μακριά 5m

Γωνία προοπτικού : 30°

Ανάλυση X/Y : 256 x 256



Εικόνα 56 : Στοιχεία Οπτικού Αισθητήρα Στο Λογισμικό Vrep

Στην μελέτη δεν χρησιμοποιήθηκαν οι οπτικοί αισθητήρες.

Τα υπόλοιπα στοιχεία του μοντέλου είναι οι τιμές που παρέχει το λογισμικό Vrep.

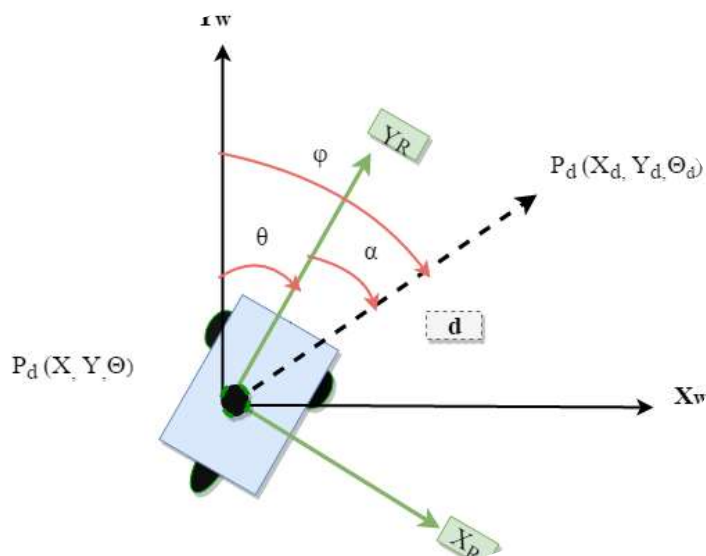


### 3.3.5 Ο Ασαφής Ελεγκτής

Ο ασαφής ελεγκτής πλοήγησης οδηγεί το όχημα από μια δεδομένη αρχική θέση σε ένα σημείο-στόχο. Η απόκριση του ελεγκτή οδηγεί το όχημα προς τα εμπρός είτε προς τα αριστερά είτε προς τα δεξιά, ανάλογα με τη μεταβλητή εισόδου. Η βασική λογική πάνω στην οποία στηρίζεται ο ελεγκτής είναι: Εφόσον δεν υπάρχει εμπόδιο τότε αν η επιθυμητή θέση είναι αριστερά της τρέχουσας, το ρομπότ στρίβει αριστερά, αν η επιθυμητή θέση είναι δεξιά της τρέχουσας στρίβει δεξιά, αν είναι στο κέντρο της τρέχουσας πηγαίνει ευθεία. Για να μπορέσουμε να κατευθύνουμε το όχημα είναι απαραίτητο να υπολογίσουμε το σφάλμα κατεύθυνσης. Αν υπάρχει εμπόδιο τότε το όχημα πρώτα αποφεύγει το εμπόδιο και μετά προσπαθεί να προσεγγίσει το σημείο στόχο.

Οι δύο τροχοί ενός διαφορικά οδηγούμενου ρομποτικού οχήματος είναι σταθεροί και κάθε τροχός ελέγχεται ανεξάρτητα από τον κινητήρα του, έτσι ώστε η διαδρομή του ελεγχόμενου ρομπότ να καθορίζεται από τις ταχύτητες των δύο κινητήρων. Το αντικείμενο του σχεδιασμού ενός ασαφούς ελεγκτή είναι να καθοριστούν οι ταχύτητες του αριστερού και δεξιού τροχού, έτσι ώστε το ρομπότ να μπορεί να κινείται αποτελεσματικά από την τρέχουσα θέση του  $(x,y)$ , σε μια επιθυμητή θέση  $(x_d,y_d)$ .

Το σχηματικό διάγραμμα ενός ρομποτικού οχήματος με διαφορική οδήγηση φαίνεται στην εικόνα 57, όπου  $X_w-Y_w$  είναι οι παγκόσμιες (global) συντεταγμένες και  $X_R-Y_R$  είναι οι τοπικές (local) συντεταγμένες, όπου η αρχή των αξόνων του τοπικού πλαισίου συντεταγμένων βρίσκεται στο κέντρο  $p$  του σταθερού πλαισίου του ρομποτικού οχήματος. Το σώμα του ρομπότ είναι συμμετρικό και το κέντρο μάζας βρίσκεται στο γεωμετρικό κέντρο  $p$  του σώματος. Τα σημεία  $(x,y)$  αντιπροσωπεύουν τη θέση του γεωμετρικού κέντρου  $p$  στο παγκόσμιο σύστημα συντεταγμένων  $X_w-Y_w$  και  $\theta$  είναι η γωνία που δείχνει τον προσανατολισμό του ρομπότ. Η  $\theta$  γωνία λαμβάνεται δεξιόστροφα από τον  $Y_w$  προς  $X_w$  άξονα.



Εικόνα 57 : Θέση και προσανατολισμός ρομποτικού οχήματος στην αρχική και στην επιθυμητή [62]

Το ρομπότ έχει δύο βαθμούς ελευθερίας σε σχέση με τη θέση και τον προσανατολισμό του και περιγράφονται από το διάνυσμα  $p(x, y, \theta)$ . Στην εικόνα 53 τα διανύσματα

$p(x,y,\theta)$  και  $p_d(x_d,y_d,\theta_d)$  περιγράφουν την τρέχουσα  $(x, y)$  και την επιθυμητή θέση  $(x_d, y_d)$  του ρομπότ,  $\theta$  είναι ο τρέχον προσανατολισμός και  $\theta_d$  είναι ο προσανατολισμός του ρομπότ στην επιθυμητή θέση. Η ευκλείδεια απόσταση μεταξύ της τρέχουσας  $(x, y)$  και επιθυμητής θέσης  $(x_d, y_d)$  είναι  $d$ , όπου

$$d = \sqrt{(x_d - x)^2 + (y_d - y)^2}$$

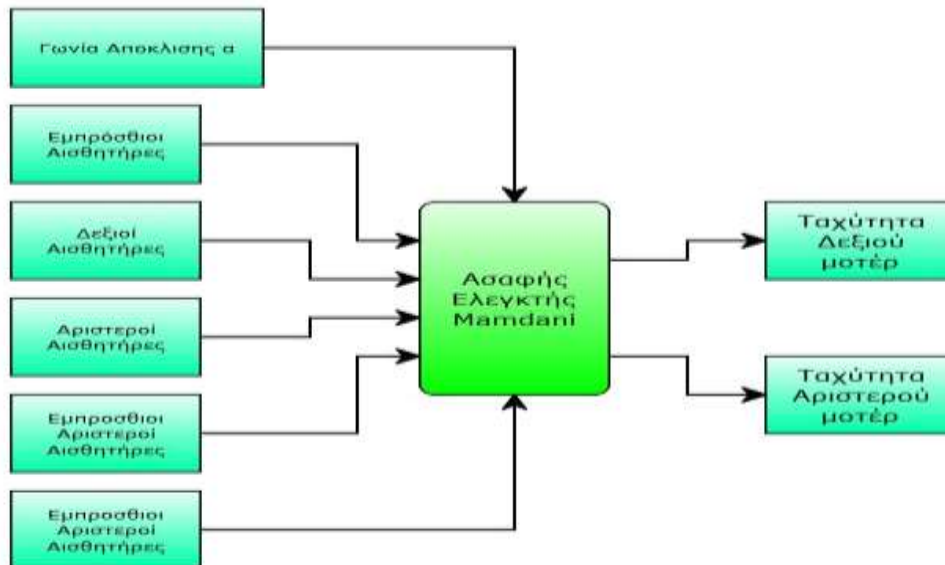
$\theta$  είναι η γωνία μεταξύ του  $Y_w$  άξονα και του προσανατολισμού του ρομπότ στην τρέχουσα θέση,  $\varphi$  είναι η γωνία μεταξύ του  $Y_w$  άξονα και του προσανατολισμού του ρομπότ στην επιθυμητή θέση και υπολογίζεται από τη σχέση:

$$\varphi = \tan^{-1} \frac{y_d - y}{x_d - x}$$

Η γωνία  $\alpha$  είναι η γωνία μεταξύ του προσανατολισμού του ρομπότ και μεταξύ της επιθυμητής θέσης και ονομάζεται και σφάλμα της γωνίας κατεύθυνσης και δίνεται από τη σχέση:

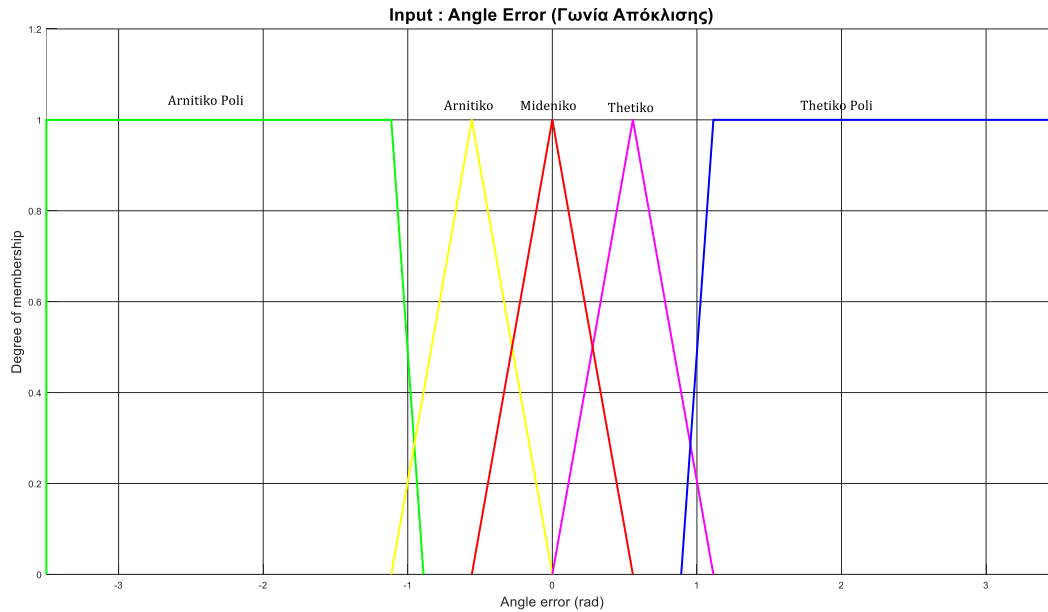
$$\alpha = \begin{cases} \varphi - \theta, & \text{εάν } \varphi - 180 < \theta < 180 \\ (\varphi - \theta) - 360, & \text{εάν } -180 < \theta < \varphi - 180 \end{cases}$$

Τα στοιχεία του ασαφή ελεγκτή παρουσιάζονται στην εικόνα 58. Υπάρχουν 6 είσοδοι. Γωνία απόκλισης και 5 ομάδες αισθητήρων. Υπάρχουν 2 έξοδοι. Ταχύτητα των μοτέρ του εκάστοτε τροχού



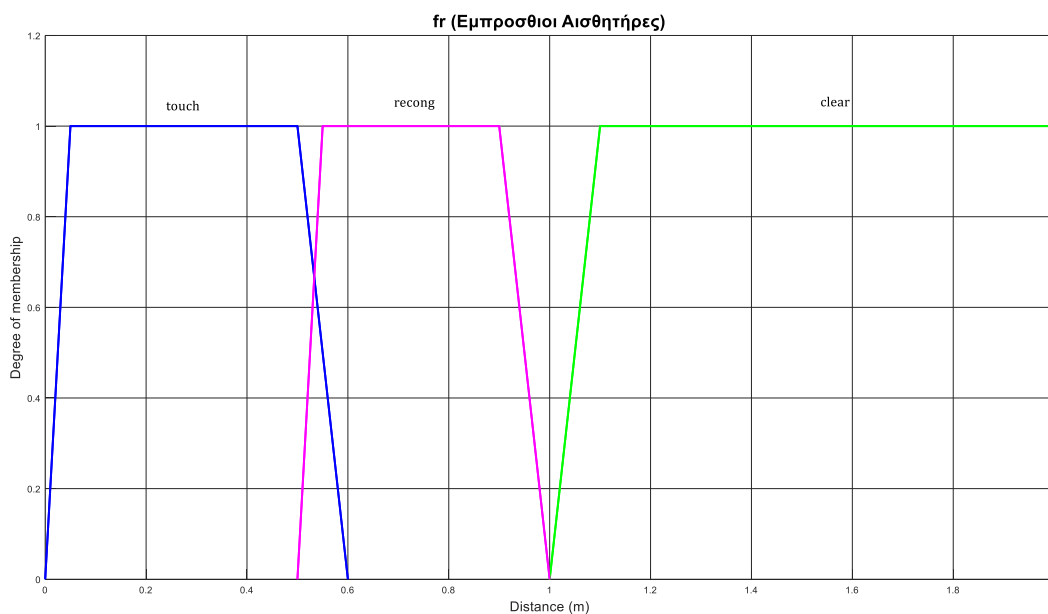
Εικόνα 58: Είσοδοι & Έξοδοι του Ασαφή Ελεγκτή

Η πρώτη είσοδος του ασαφή ελεγκτή είναι η γωνία απόκλισης. Χωρίζουμε την είσοδο σε 5 ασαφή σύνολα που χαρακτηρίζονται από τις συναρτήσεις συμμετοχής τους: Αριστερά Πολύ, αριστερά, μηδενικό, δεξιά, δεξιά πολύ και έχει εύρος από  $-3,14$  έως  $3,14$  rad που ισοδυναμούν σε μία πλήρη περιστροφή είτε αριστερόστροφα είτε δεξιόστροφα, όπως φαίνεται στην εικόνα 59.



Εικόνα 59 : Είσοδος Γωνία Απόκλισης

Επόμενη είσοδος του ασαφή ελεγκτή είναι οι αισθητήρες. Η είσοδος του ασαφή ελεγκτή έχει χωριστεί σε 3 ασαφή σύνολα που χαρακτηρίζονται από τις συναρτήσεις συμμετοχής: touch (επαφή), recong (αναγνώριση) και clear (άδειο ή καθαρό) με εύρος τιμών 0 έως και 2 m όπως φαίνεται στην εικόνα 60.

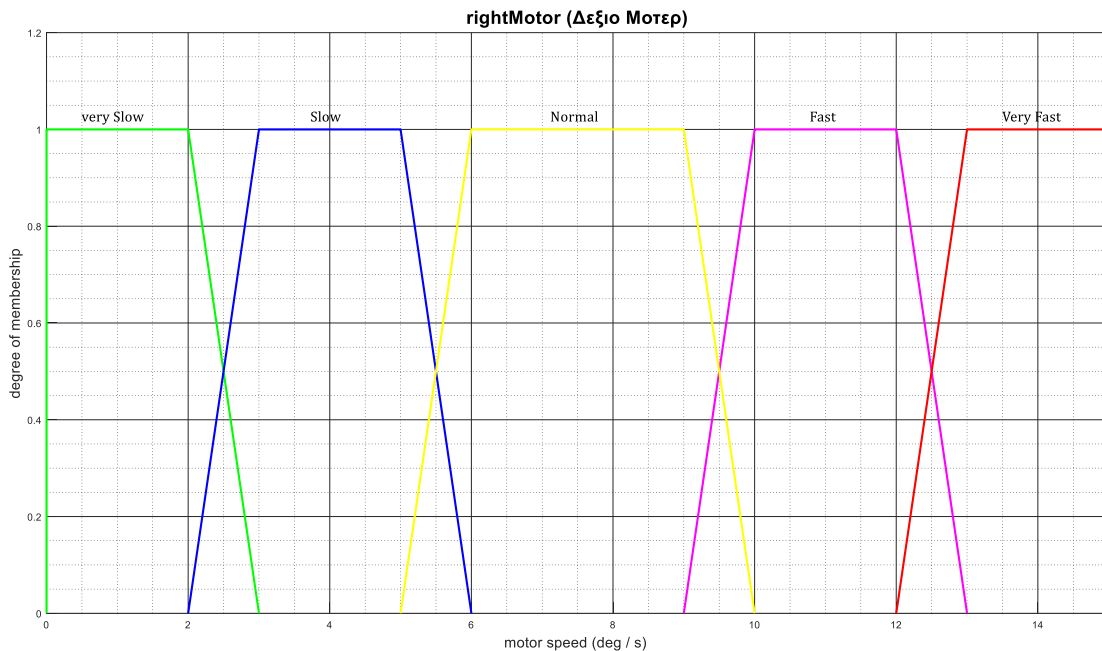


Εικόνα 60: Είσοδος Εμπρόσθιοι Αισθητήρες

Οι υπόλοιπες είσοδοι του ασαφή που αναφέρονται στις υπόλοιπες ομάδες Αριστεροί Αισθητήρες (lf), Δεξιοί Αισθητήρες (rt), Εμπρόσθιοι Αριστερά (frlf), Εμπρόσθιοι Δεξιά (frrt) είναι ακριβώς ίδιες με των εμπρόσθιων αισθητήρων όπως αυτή φαίνεται στην εικόνα 60.

Η πρώτη έξοδος του ασαφή ελεγκτή είναι η ταχύτητα του δεξιού μοτέρ. Έχει χωριστεί σε 5 ασαφή σύνολα που χαρακτηρίζονται από την συνάρτηση συμμετοχής: very slow (πολύ αργά), slow (αργά), normal (κανονική), fast (γρήγορα) και very fast (πολύ

γρήγορα). Το εύρος των τιμών είναι από 0 έως και 15 μοίρες ανά δευτερόλεπτο (deg/s).



Εικόνα 61 : Έξοδος ταχύτητα Δεξιού Μοτέρ

Η δεύτερη έξοδος αναφέρεται στην ταχύτητα του αριστερού μοτέρ. Είναι ίδια ακριβώς με την είσοδο του δεξιού μοτέρ όπως αυτή φαίνεται στην εικόνα 61.

Η βάση κανόνων αποτελείται από 275 λεκτικούς κανόνες και στηρίζεται στην παρακάτω φιλοσοφία:

1. Έλεγχος θέσης και απόκλισης από το σημείο που βρίσκεται το όχημα με το σημείο που του ορίζουμε να κινηθεί.
2. Έλεγχος αισθητήρων για το αν υπάρχει κάποιο εμπόδιο κοντά στο όχημα.

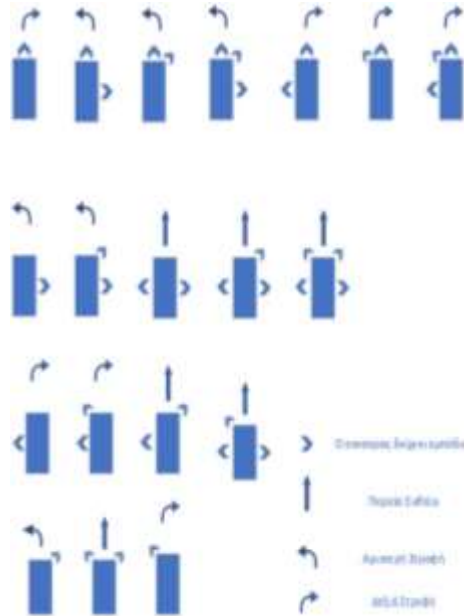
Αν δεν υπάρχει εμπόδιο, τότε το όχημα θα στρίψει ώστε να μηδενίσει την γωνία απόκλισης που έχει η διεύθυνση του οχήματος σε σχέση με την ευθεία που σχηματίζει το σημείο που είναι το όχημα με το επιθυμητό σημείο. Στον πίνακα 3 φαίνονται ένα μέρος από την βάση κανόνων και συγκεκριμένα οι λεκτικοί κανόνες όταν δεν υπάρχουν εμπόδια.

| Είσοδοι Ασαφή Ελεγκτή |       |       |       |       |       | Έξοδοι Ασαφή Ελεγκτή |           |
|-----------------------|-------|-------|-------|-------|-------|----------------------|-----------|
| Angle                 | fr    | Rt    | Lf    | frft  | frlf  | Right                | Left      |
| Mideniko              | clear | clear | Clear | clear | clear | Very Fast            | Very Fast |
| Thetiko               | clear | clear | Clear | clear | clear | Normal               | Fast      |
| Thetiko Poli          | clear | clear | Clear | clear | clear | Slow                 | Normal    |
| Arnhhtiko             | clear | clear | Clear | clear | clear | Fast                 | Normal    |
| Arnhhtiko Poli        | clear | clear | Clear | clear | clear | Normal               | Slow      |

Πίνακας 3 : Λεκτική κανόνες Ασαφούς Ελεγκτή

Γενικά το όχημα εφόσον κινείται ευθεία ή σε μικρή γωνία απόκλισης προσπαθεί να κινείται στην μέγιστη ταχύτητα. Αν βρει εμπόδιο, ανάλογα την θέση του εμποδίου ως προς το όχημα αλλά και την απόσταση από αυτό, ρυθμίζει την πορεία του και την

ταχύτητα του. Αν υπάρχει κάποιο εμπόδιο, το πρώτο μέλημα του ασαφή ελεγκτή είναι η αποφυγή του εμποδίου. Η ταχύτητα μειώνεται ανάλογα τις τιμές που δίνουν οι αισθητήρες και αφορούν την απόσταση του οχήματος από το εμπόδιο. Παρακάτω παρουσιάζεται η φιλοσοφία που δομήθηκε ο τρόπος που το όχημα θα αποφύγει τα εμπόδια.



Εικόνα 62: Φιλοσοφία Κίνησης Οχήματος σε εμπόδια

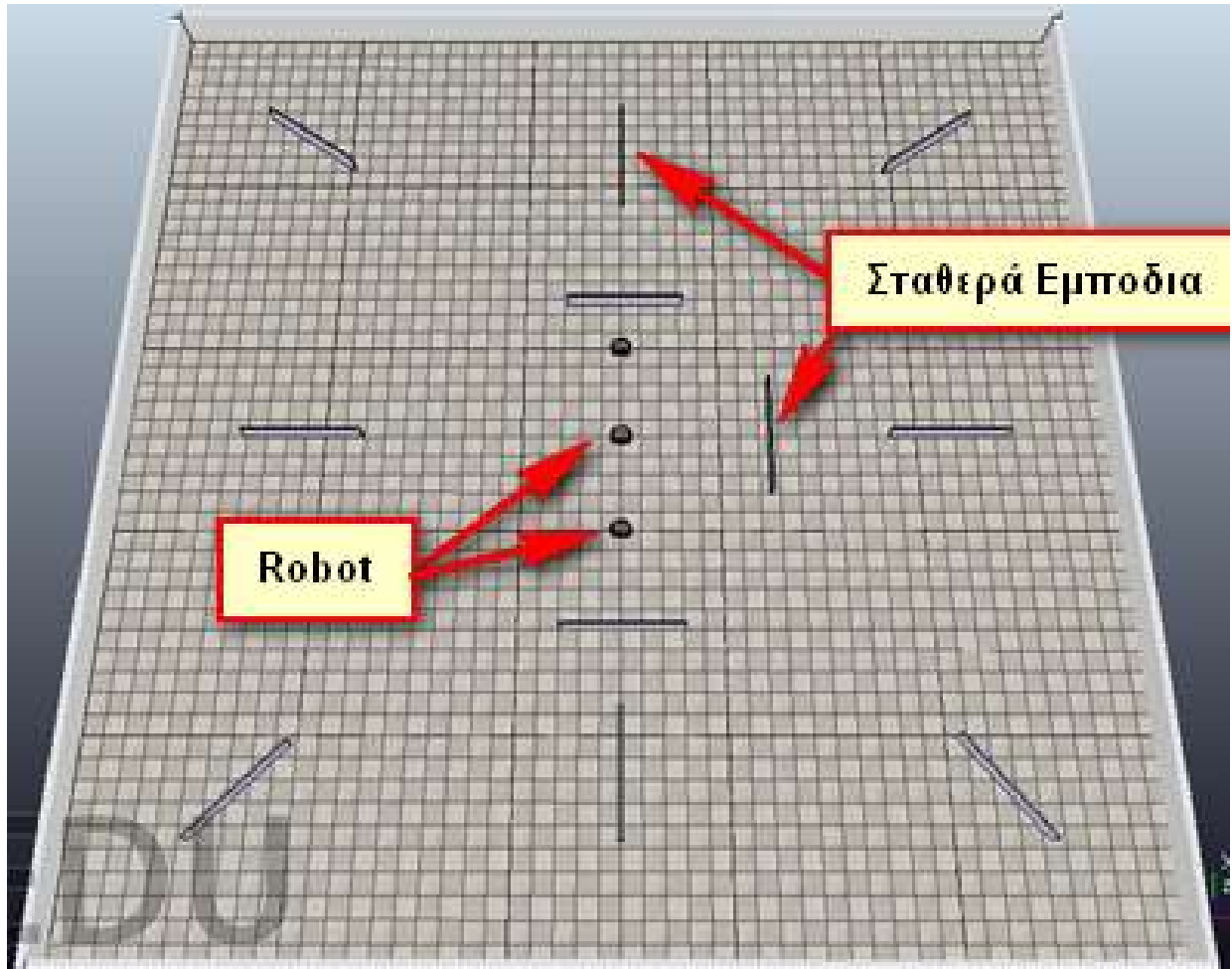
### 3.3.6 Προσομοίωση Ομάδας Ρομποτικών Οχημάτων στο Vrep.

Για τη προσομοίωση του ρομποτικού οχήματος Pioneer για τις δοκιμές του στρατηγικού ελέγχου στο V-REP αρχικά δημιουργήθηκε ο χώρος κίνησης των ρομποτικών οχημάτων που ονομάζεται **σκηνή** και μπορεί να έχει διάφορα στατικά ή κινούμενα εμπόδια, καθώς και σημεία-στόχους που θα προσεγγίζει το όχημα. Αρχικά δημιουργήθηκαν διάφορες σκηνές με ποικίλες διαμορφώσεις σε εμπόδια, ανάλογα με το σκοπό της προσομοίωσης. Στην συγκεκριμένη μελέτη, λόγω ότι υπάρχει φόρτο εργασίας με την χρήση τριών διαφορετικών συναρτήσεων καταλληλότητας, πάρθηκε η απόφαση να γίνει χρήση μόνο ενός χώρου πειραματισμού (εικόνα 63). που προσομοιώνει ένα γήπεδο 25 x 25 μέτρων με σταθερά εμπόδια. Ως κινητά εμπόδια θεωρούνται τα κινούμενα ρομποτικά οχήματα που υπάρχουν στην σκηνή.

Η αποστολή που καλούνται να εκτελέσουν είναι μετακίνηση (movement) και εντάσσεται στην κατηγορία: κίνηση – μεταφορά (Locomotion). Η άσκηση ήταν τα ρομπότ να προσπαθήσουν να προσεγγίσουν 3 σημεία (σταθερά) διαδοχικά. Οποιοδήποτε από τα ρομποτικά οχήματα προσέγγιζε το σημείο τότε αυτόματα τα ρομποτικά οχήματα προσπαθούσαν να πάνε στο επόμενο σημείο.

Στη συνέχεια αναπτύχθηκε ο κατάλληλος κώδικας ώστε κάθε αντικείμενο που υπάρχει στην αρένα της προσομοίωσης του V-REP να μπορεί να αναγνωριστεί από το MATLAB, να γίνει η λήψη των δεδομένων για τη θέση και τον προσανατολισμό των οχημάτων, τη θέση του στόχου, τις μετρήσεις των αισθητηρίων. Κατά τη λήψη των δεδομένων έχει εισαχθεί κατάλληλη χρονική καθυστέρηση για την σωστή ενημέρωση των δεδομένων. Στη συνέχεια ακολουθούν οι μαθηματικοί υπολογισμοί όπως της

απόστασης από το στόχο, του σφάλματος προσανατολισμού σε σχέση με το στόχο, και ξεκινάει η διαδικασία για τον ασαφή έλεγχο για κάθε ένα ρομποτικό όχημα διαδοχικά. Αφού εισαχθούν τα δεδομένα που αποτελούν τις εισόδους του ελεγκτή, το MATLAB θα κάνει τους υπολογισμούς θα βγάλει τις τιμές εξόδου και θα στείλει τις τιμές των ταχυτήτων των τροχών στο V-REP ώστε να κινηθεί κατάλληλα το όχημα.



Εικόνα 63 : Σκηνή ασκήσεων Ρομποτικών Οχημάτων

Επίσης δημιουργήθηκε κατάλληλη δομή όπου καταγράφονται όλα τα δεδομένα της προσομοίωσης. Στο παράρτημα Δ γίνεται παρουσίαση της δομής.

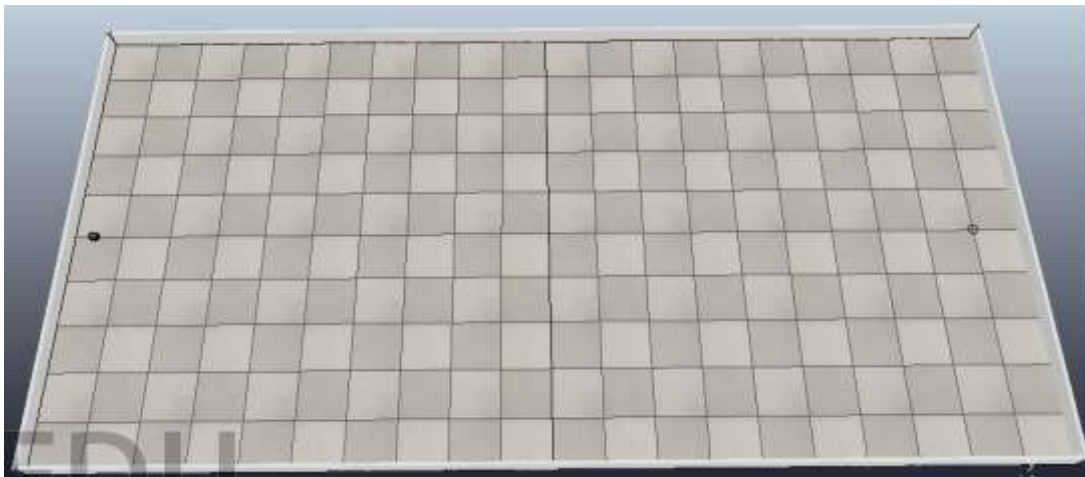
#### 3.4.3.1 Σκηνή ορισμού τιμών ταχύτητας μοτέρ

Ένα ερώτημα που τέθηκε ήταν το εύρος των τιμών που πρέπει να δίνει ο ελεγκτής στα μοτέρ. Από τα στοιχεία του λογισμικού Vrep που προσομοιώνει το ρομποτικό όχημα για τα μοτέρ η ταχύτητα δίνεται σε deg/s (εικόνα 64). Οπότε έπρεπε να γίνει αντιστοίχιση των τιμών που θα δίνονταν στα μοτέρ ώστε να μην ξεπεράσει το μοντέλο την μέγιστη ταχύτητα των 1.2 m/s όπως αυτή δίνεται από τα δεδομένα του κατασκευαστή.

Για τον σκοπό αυτόν γράφτηκε κώδικας στο λογισμικό MatLab, όπου σε ειδική για τον σκοπό αυτό σκηνή του Vrep (εικόνα 65) δίνονται διάφορες τιμές γωνιακής ταχύτητας στα μοτέρ και καταγράφονταν η ευθύγραμμη ταχύτητα του οχήματος, με σκοπό να γίνει η αντιστοιχία των τιμών.



Εικόνα 64 : Φόρμα Στοιχείων Μοτέρ στο λογισμικό Vrep



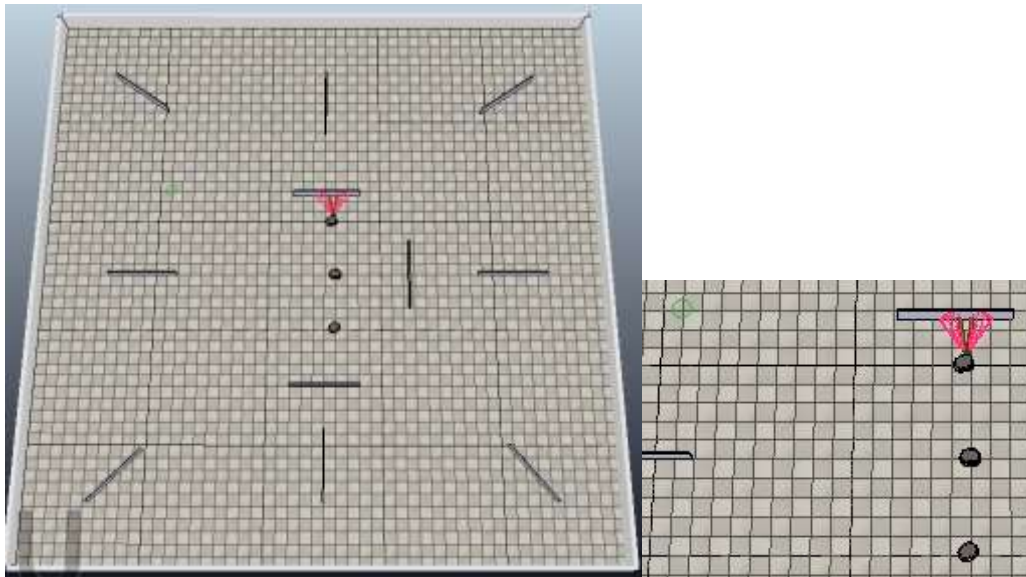
Εικόνα 65 : Πίστα υπολογισμού τιμών ταχύτητας μοτέρ

Η σκηνή είναι μια επίπεδη επιφάνεια χωρίς εμπόδια, διαστάσεων 50 x 25 μέτρων. Το μέγεθος εξήχθη πειραματικά ώστε να προλαβαίνουμε να πιάσει την μέγιστη τιμή που ορίζαμε στην ρουτίνα στο MatLab.

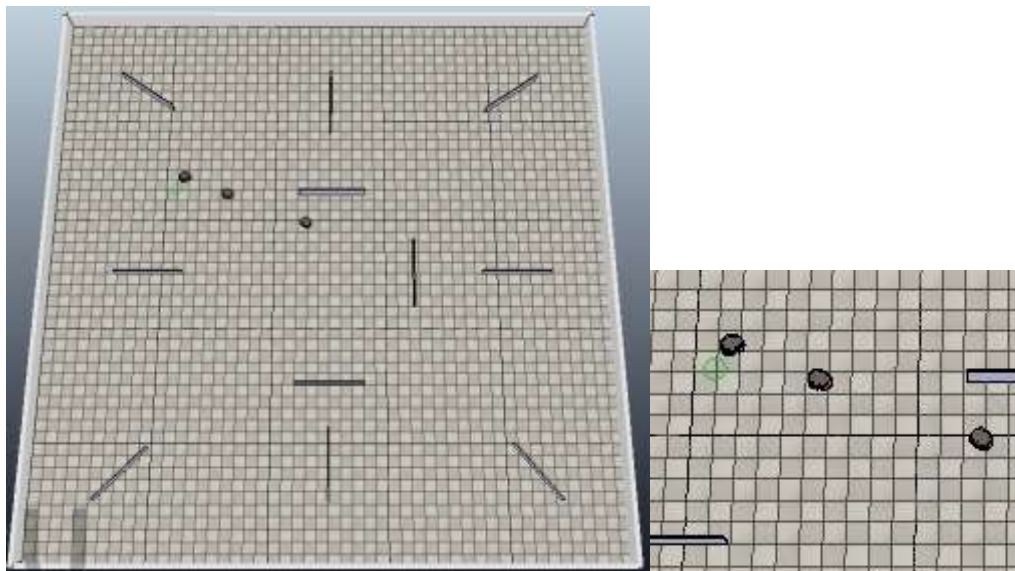
Το όχημα το έχουμε στην αρχή την πίστας και σταδιακά αυξάνουμε την τιμή ταχύτητας στα μοτέρ. Στην συνέχεια παίρνουμε από την προσομοίωση τις ταχύτητες σε όλους τους άξονες(x, y, z) και υπολογίζουμε την ευθύγραμμη ταχύτητα του οχήματος. Στο MatLab παίρνουμε τον πίνακα τιμών,

### 3.4 Επιβεβαίωση λειτουργίας ασαφή ελεγκτή - σεναρίου

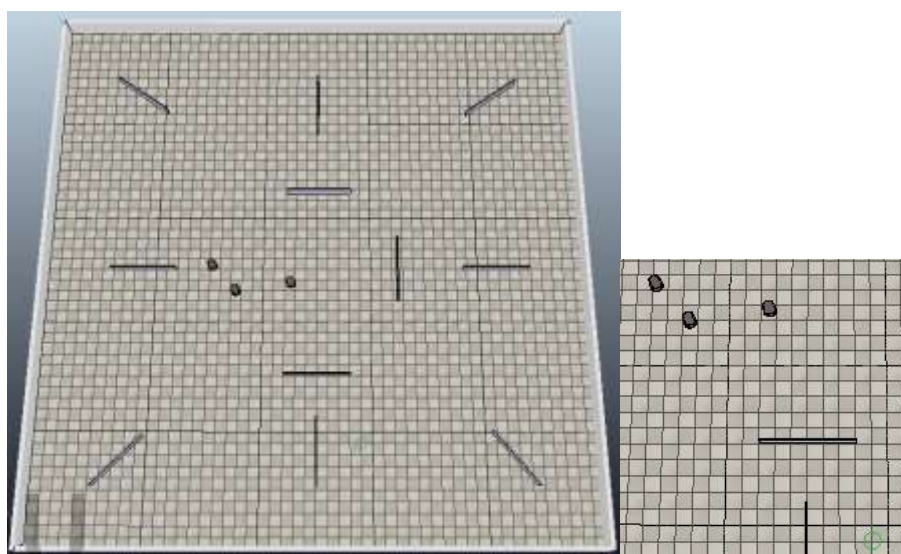
Στη συνέχεια έγινε η προσομοίωση του ρομποτικού οχήματος Pioneer στο Vrep με τον ασαφή ελεγκτή που παρουσιάσαμε προηγουμένως. Σκοπός ήταν επαληθεύσουμε ότι ο ελεγκτής που σχεδιάσαμε πραγματικά μπορούσε να οδηγήσει τα ρομποτικά οχήματα (όχι διαδοχικά, αλλά τυχαία) να μεταβούν από ένα αρχικό σημείο σε όλα τα σημεία που είχαμε ορίσει. Στις εικόνες 66 έως 68 παρουσιάζονται στιγμιότυπα της κίνησης του οχήματος.



Εικόνα 66 & 66α : Οχήματα στο αρχικό σημείο εκκίνησης

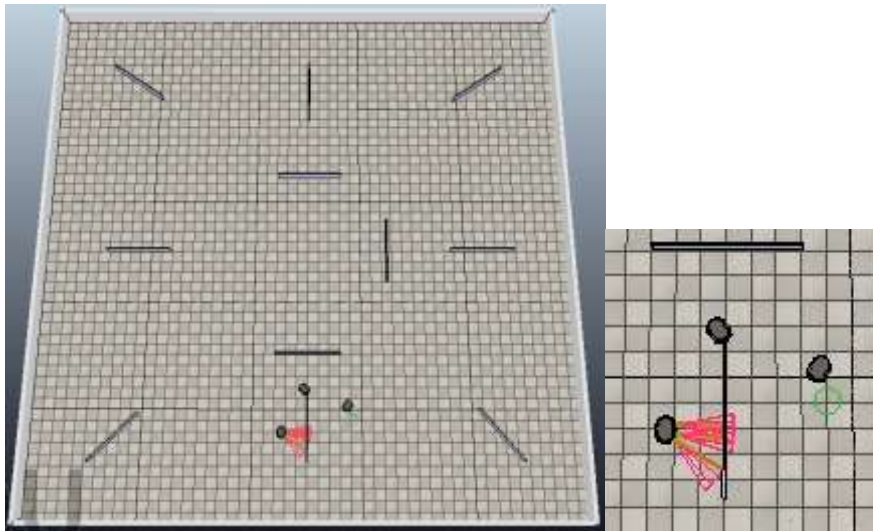


Εικόνα 67 & 67α : Οχήματα στο πρώτο σημείο στόχος

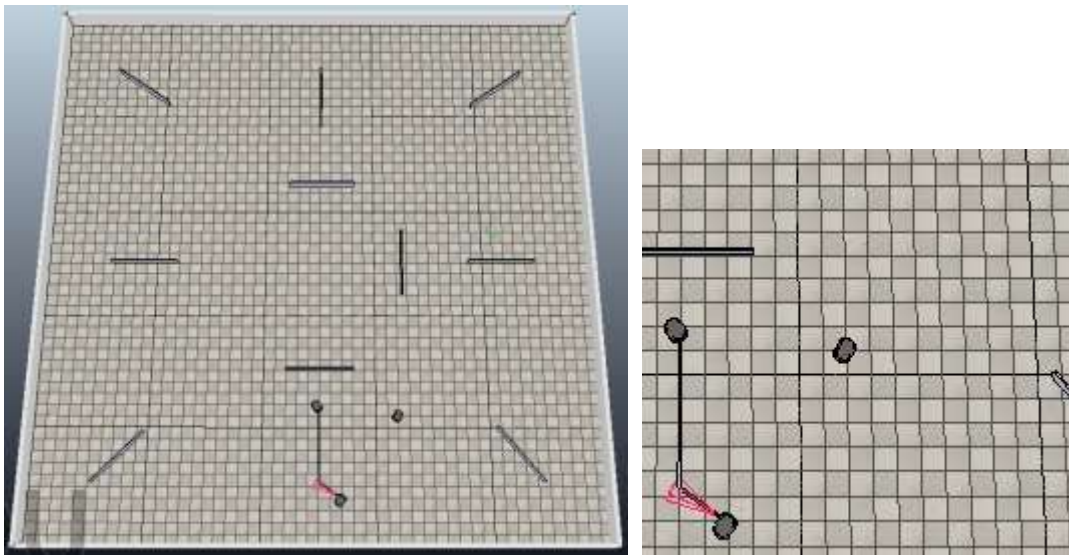


Εικόνα 68 & 68α: Τα οχήματα κινούνται προς το δεύτερο σημείο στόχος

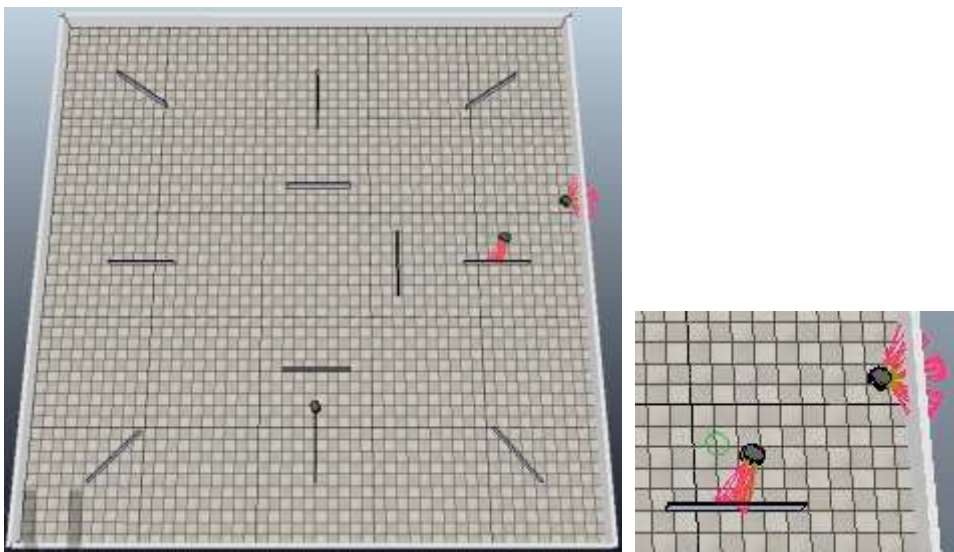




Εικόνα 69 & 69α: Τα οχήματα στο δεύτερο σημείο στόχος



Εικόνα 70 & 70α: Τα οχήματα κινούνται προς το τρίτο σημείο στόχος



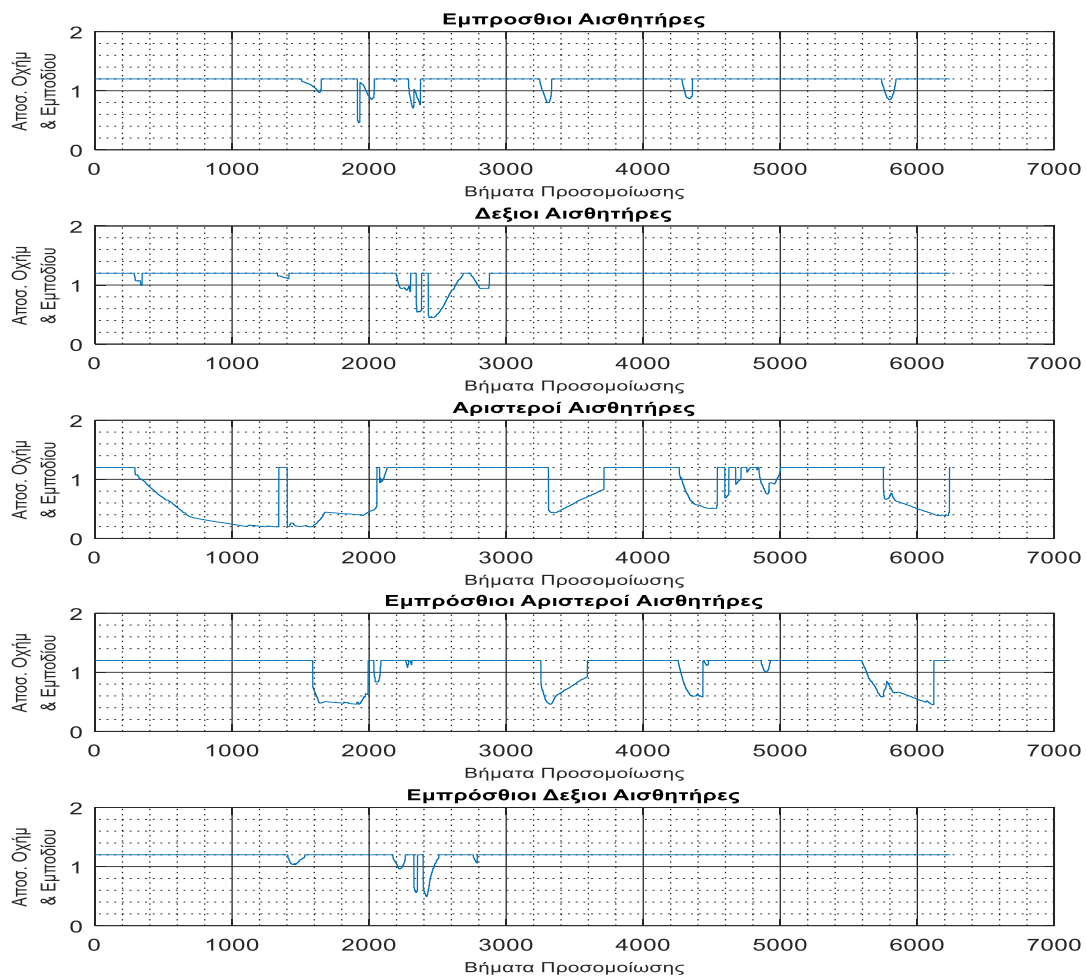
Εικόνα 71 & 71α : Τα οχήματα στο τρίτο σημείο στόχος

Παρατηρούμε ότι τα όχημα διανύουν μια ομαλή τροχιά και καταφέρνουν να κινηθούν και να προσεγγίσουν το σημείο-στόχο, το σφάλμα της γωνίας κατεύθυνσης μειώνεται έως ότου φτάσει κοντά στις 0 μοίρες, οι ταχύτητες των τροχών φτάνουν κοντά στα 1,2 m/s την μέγιστη δηλαδή ταχύτητα του οχήματος. Τα στοιχεία για τα οποία σχεδιάστηκε ο ασαφής ελεγκτής τα επιτυγχάνει.

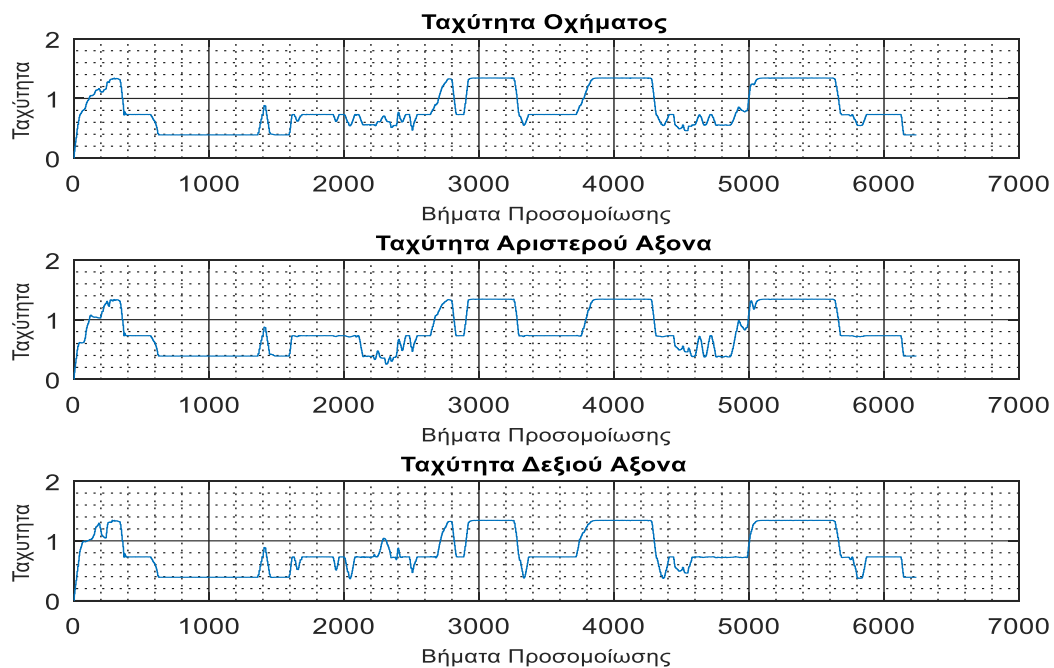
Η συγκεκριμένη προσομοίωση αποδεικνύει τόσο την λειτουργικότητα του ασαφούς ελεγκτή όσο και την ορθή λειτουργία του περιβάλλοντος προσομοίωσης που έχει επιλεγεί για τα πειράματα στα πλαίσια της συγκεκριμένης εργασίας.

Τα δεδομένα της προσομοίωσης παρουσιάζονται, σε γραφήματα για κάθε όχημα ξεχωριστά, στις παρακάτω εικόνες. Το επιθυμητό για το γράφημα των αισθητήρων είναι να είναι πάντα στην τιμή του 1.2m. Οι ταχύτητες των αξόνων να είναι ίδιες και να προσεγγίζει την μέγιστη τιμή των 1.2 m/sec που σημαίνει ότι το όχημα δεν στρίβει συνέχεια. Η διαδρομή του οχήματος μας δείχνει ότι το όχημα προσέγγισε ή όχι τα σημεία.

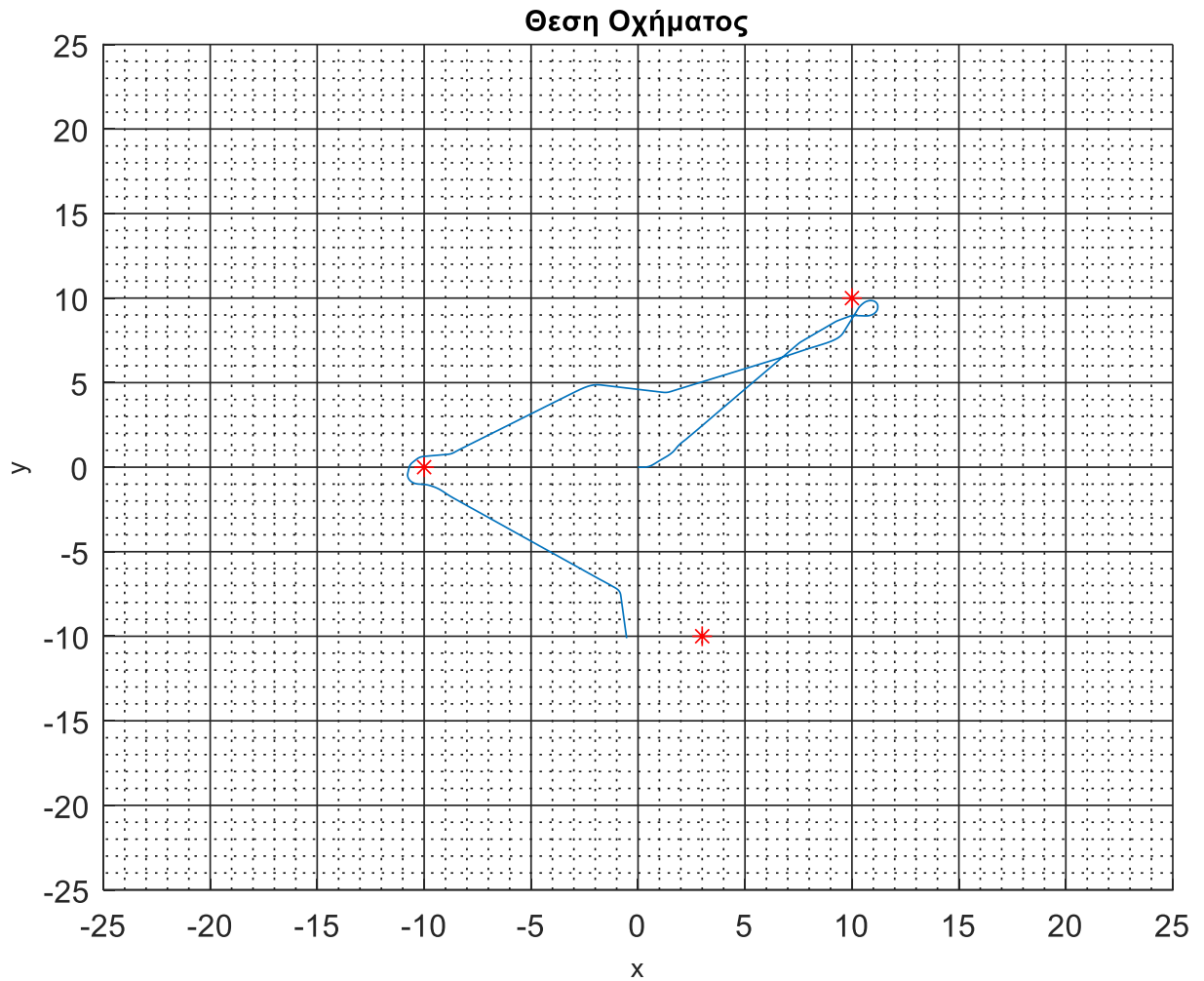
### 3.4.1 Αποτελέσματα προσομοίωσης 1<sup>ου</sup> Οχήματος



Εικόνα 72 : Δεδομένα Αισθητήρων 1ου οχήματος+



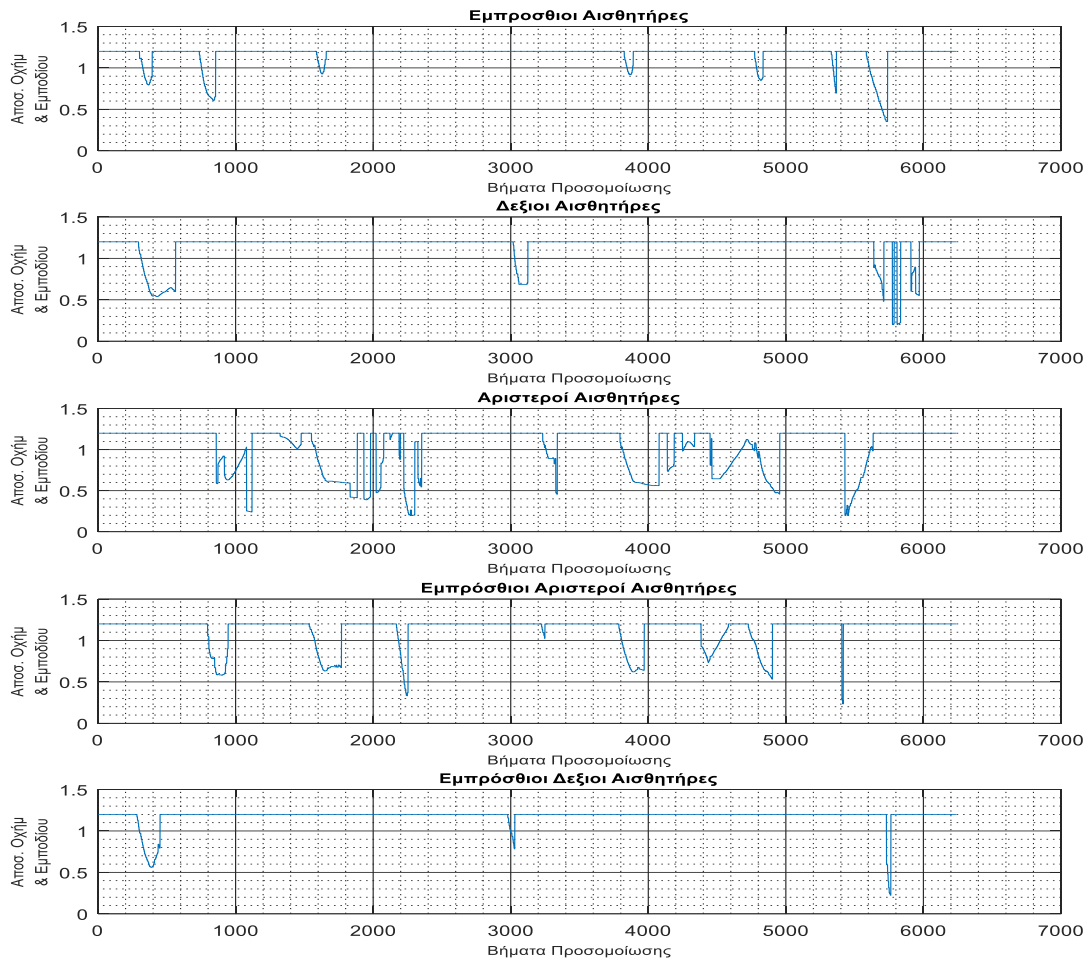
Εικόνα 73 : Ταχύτητες 1ου οχήματος



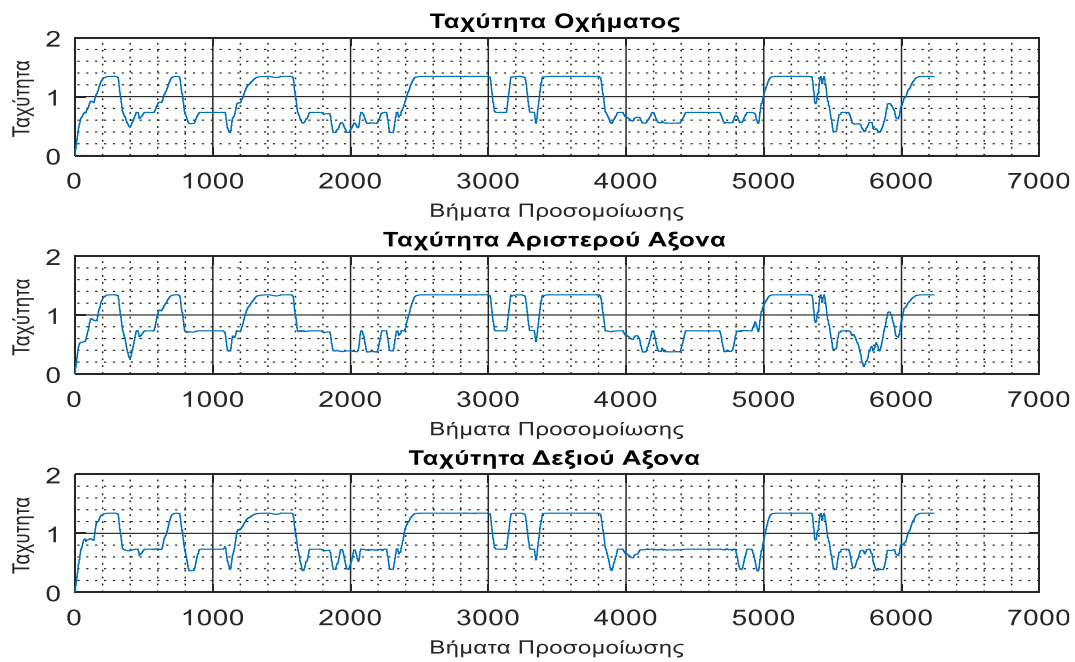
Εικόνα 74 : Διαδρομή 1ου Οχήματος

Στο διάγραμμα των αισθητήρων του 1<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται κατά την διάρκεια του πειράματος όλοι σχεδόν οι αισθητήρες. Άλλοι για μεγαλύτερη διάρκεια και άλλοι λιγότερο. Ο ελεγκτής καταφέρει να αποφεύγει τα εμπόδια αυτά και προσπαθεί να διατηρεί το όχημα στην μεγαλύτερη απόσταση από το εμπόδιο. Αντίστοιχα η ταχύτητα του οχήματος προσαρμόζεται στα στοιχεία των αισθητήρων και μειώνεται ανάλογα αλλά όταν ελευθερώνεται από εμπόδια καταφέρει να την διατηρεί σε υψηλά επίπεδα. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία και καταφέρει να τα προσεγγίσει.

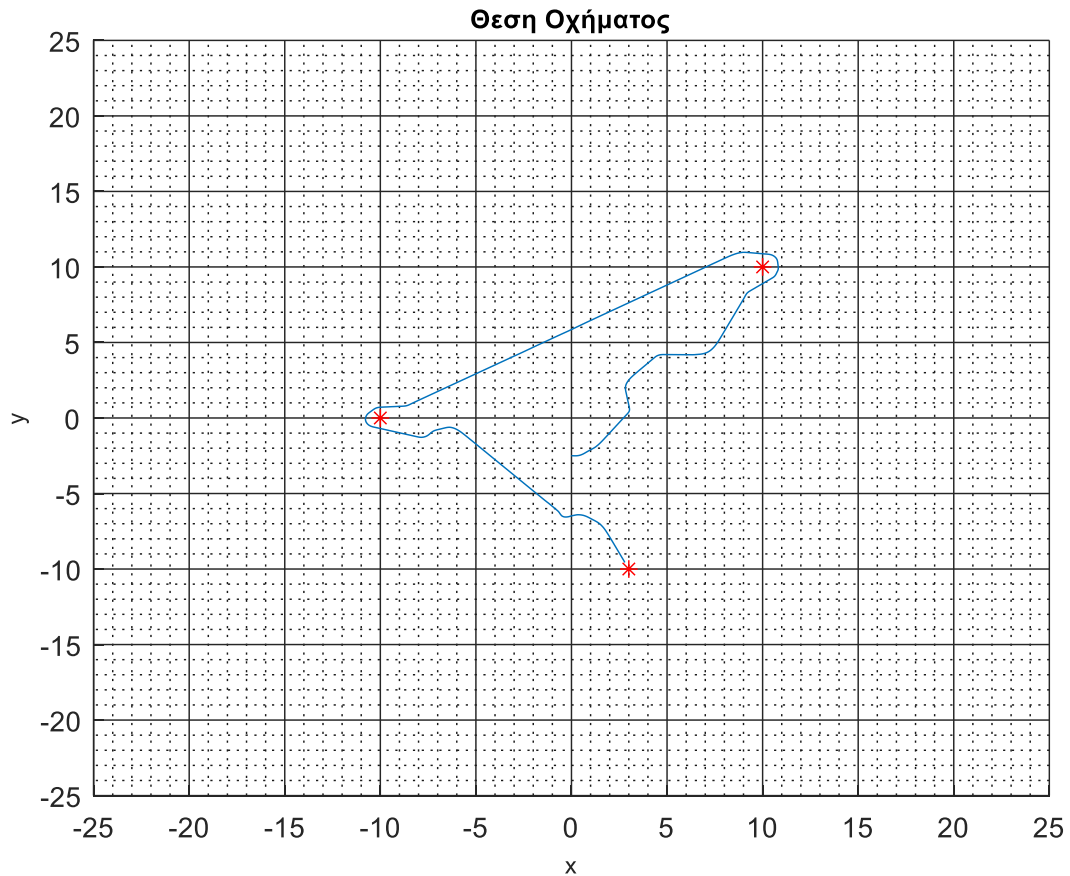
### 3.4.2 Αποτελέσματα προσομοίωσης 2<sup>ου</sup> Οχήματος



Εικόνα 75: Δεδομένα Αισθητήρων 2ου οχήματος



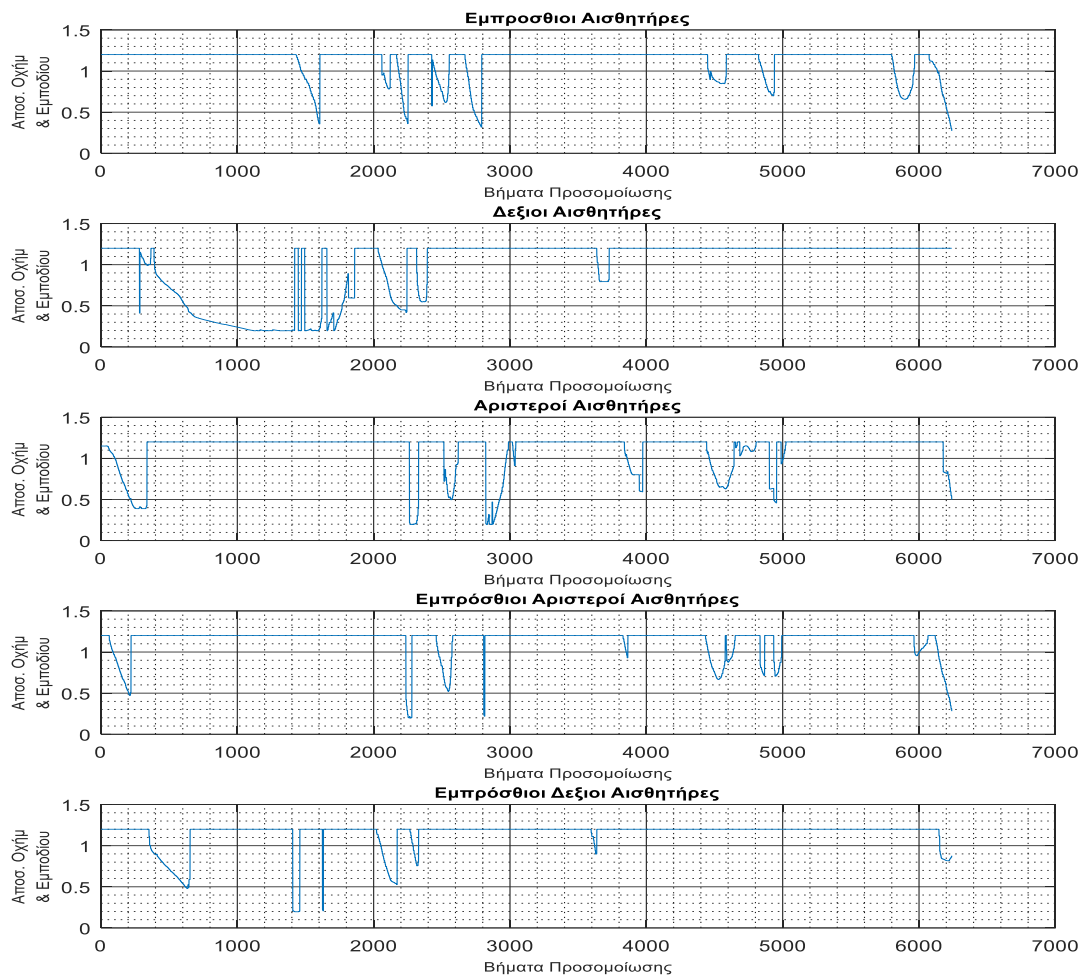
Εικόνα 76 : Ταχύτητες 2ου οχήματος



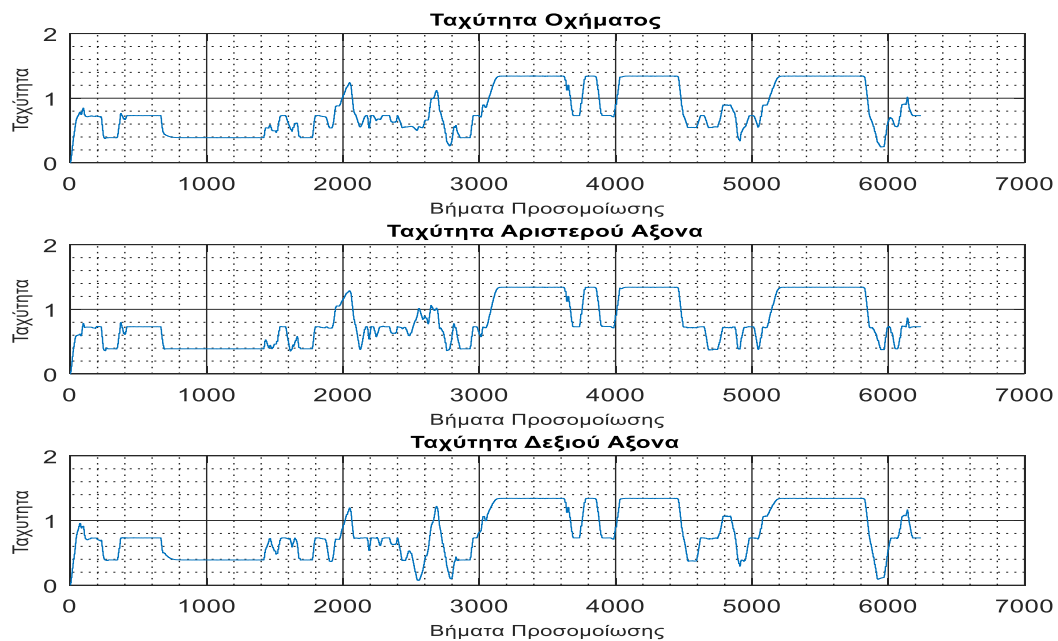
Εικόνα 77: Διαδρομή 2ου οχήματος

Στο διάγραμμα των αισθητήρων του 2<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται κατά την διάρκεια του πειράματος όλοι σχεδόν οι αισθητήρες. Άλλοι για μεγαλύτερη διάρκεια και άλλοι λιγότερο. Ο ελεγκτής καταφέρει να αποφεύγει τα εμπόδια αυτά και προσπαθεί να διατηρεί το όχημα στην μεγαλύτερη απόσταση από το εμπόδιο. Αντίστοιχα η ταχύτητα του οχήματος προσαρμόζεται στα στοιχεία των αισθητήρων και μειώνεται ανάλογα. Γενικά όμως προσπαθεί να διατηρηθεί σε επίπεδα πάνω από το 50% της δυναμικότητας του εφόσον δεν έχει εμπόδια. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία και καταφέρει να τα προσέγγιση.

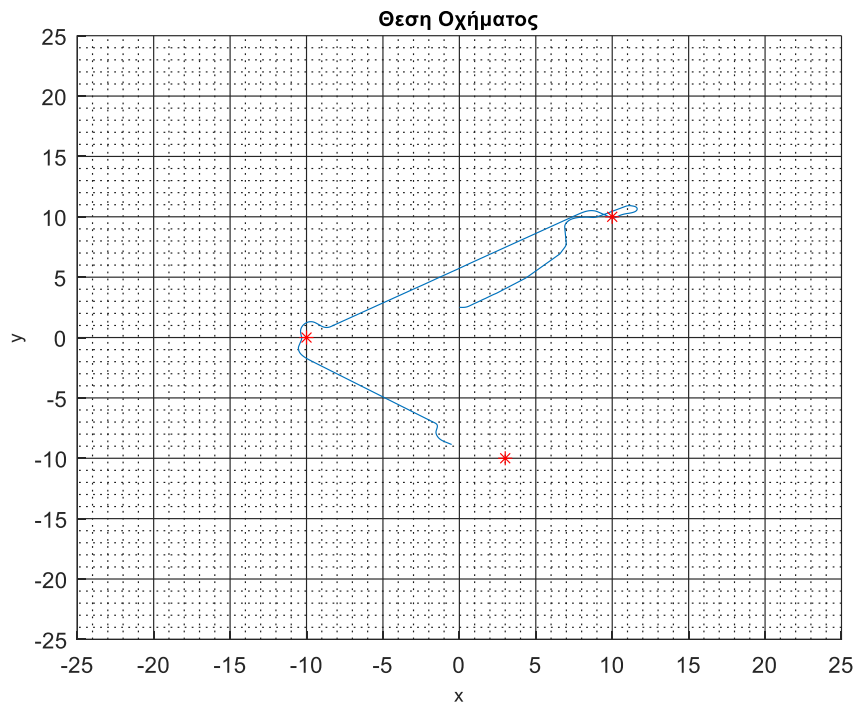
### 3.4.3 Αποτελέσματα προσομοίωσης 3ου Οχήματος



Εικόνα 78 : Δεδομένα Αισθητήρων 3ου οχήματος



Εικόνα 79 : Ταχύτητες 3ου Οχήματος



Εικόνα 80: Διαδρομή 3ου Οχήματος

Στο διάγραμμα των αισθητήρων του 3<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται κατά την διάρκεια του πειράματος όλοι σχεδόν οι αισθητήρες. Άλλοι για μεγαλύτερη διάρκεια και άλλοι λιγότερο. Ο ελεγκτής καταφέρνει να αποφεύγει τα εμπόδια αυτά και προσπαθεί να διατηρεί το όχημα στην μεγαλύτερη απόσταση από το εμπόδιο. Αντίστοιχα η ταχύτητα του οχήματος προσαρμόζεται στα στοιχεία των αισθητήρων και μειώνεται ανάλογα. Γενικά όμως προσπαθεί να διατηρηθεί σε επίπεδα πάνω από το 50% της δυναμικότητας του εφόσον δεν έχει εμπόδια. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία και καταφέρνει να τα προσεγγίσει.

Από τα διαγράμματα παρατηρούμε ότι τα όχημα εκτελούν ομαλή τροχιά και καταφέρνουν να προσεγγίσει διαδοχικά τα σημεία-στόχους, οι ταχύτητες των τροχών φτάνουν κοντά στα 1,2 m/s όπως ήταν αναμενόμενο σύμφωνα με τον σχεδιασμό του ασαφούς ελεγκτή. Η συγκεκριμένη προσομοίωση αποδεικνύει τόσο την λειτουργικότητα του ασαφούς ελεγκτή όσο και την ορθή λειτουργία του περιβάλλοντος προσομοίωσης που έχει επιλεγεί για τα πειράματα στα πλαίσια της συγκεκριμένης εργασίας.

Αποδεικνύεται επίσης και η ελεγκτής παρότι σχεδιάστηκε εμπειρικά είναι λειτουργικός και αποτελεσματικός. Ο μελετητής έχει περάσει την εμπειρία του σας χειριστής οχήματος πάνω στον ασαφή ελεγκτή. Η παρατήρηση αυτή δεν καταστά μη αναγκαία την ανάγκη βελτιστοποίησης του ελεγκτή απλά θέτει ποιο υψηλά τον πήχη για τον αλγόριθμο βελτιστοποίησης.



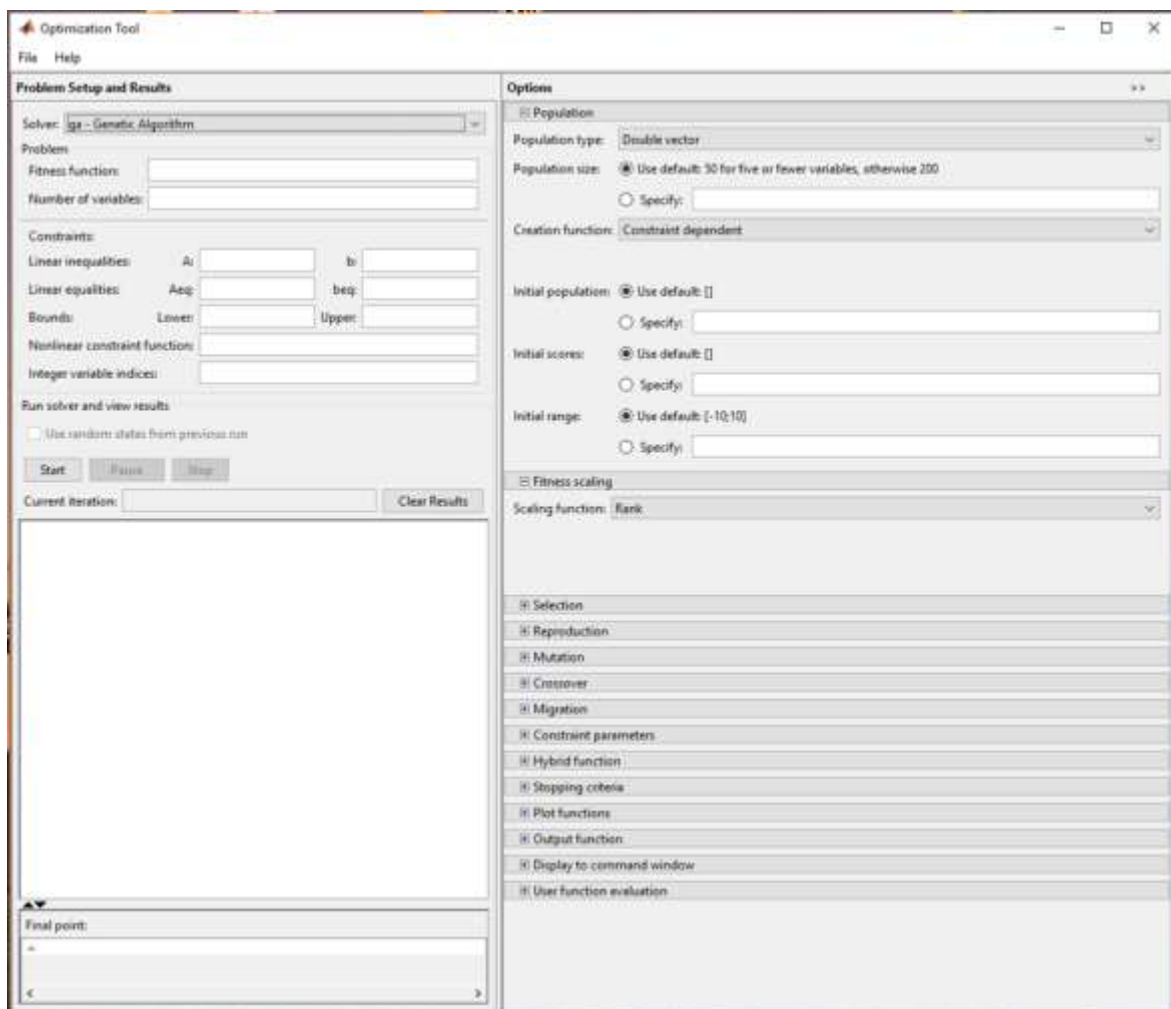
## ΚΕΦΑΛΑΙΟ 4

### Βελτιστοποίηση ασαφούς Ελεγκτή ρομποτικών οχημάτων με τη χρήση γενετικών αλγορίθμων.

#### 4.1 Εισαγωγή

Στο κεφάλαιο αυτό μελετάται η επίδραση της εφαρμογής του γενετικού αλγορίθμου βελτιστοποίησης σε ασαφείς ελεγκτές με στόχο να βελτιστοποιηθεί η συμπεριφορά τους. Γίνεται μελέτη σε 3 διαφορετικές περιπτώσεις ως προς την συνάρτηση καταλληλότητας, με σταθερές τις μεταβλητές του γενετικού αλγόριθμου, της ασκήσεως που εκτελούν τα οχήματα και της σκηνης που γίνεται η άσκηση αυτή. Οι προσομοιώσεις έγιναν στον ίδιο υπολογιστή, αρχικά με επεξεργαστή AMD Athlon 64 στα 1.70GHz 64bit και μετά από την αδυναμία του συστήματος να εκτελεί τις προσομοιώσεις σε λογικά χρονικά πλαίσια έγινε αλλαγή σε AMD Ryzen 5 στα 1.60GHz 64bit με 6 Core.

Η εξελικτική διαδικασία έγινε με την χρήση του MatLab και του εργαλείου που μας παρέχει το Optimization Tool. Το γραφικό περιβάλλον δ επαφής παρουσιάζεται στην εικόνα 81.



Εικόνα 81 : Εργαλειοθήκη Εξελικτικών υπολογισμών OptimTool

Υπάρχει δυνατότητα υποστήριξης διαφορετικών ειδών βελτιστοποιήσεων, τα βασικότερα των οποίων παρουσιάζονται στον πίνακα 4.

|                |  |
|----------------|--|
| ga             | Εύρεση του ελαχίστου μιας συνάρτησης με την χρήση γενετικών αλγορίθμων             |
| gamultiobj     | Εύρεση του ελαχίστου πολλαπλών συναρτήσεων με την χρήση γενετικών αλγορίθμων       |
| patternsearch  | Εύρεση του ελαχίστου μιας συνάρτησης με την αναζήτησης μοτίβων                     |
| particleswarm  | Βελτιστοποίηση με χρήση σμήνους σωματιδίων   |
| simulannealbnd | Εύρεση ελαχίστου μιας συνάρτησης χρησιμοποιώντας προσομοιωμένο αλγόριθμο ανόπτωσης |
| GlobalSearch   | Εύρεση καθολικών ελαχίστων.  |
| MultiStart     | Εύρεση πολλαπλών τοπικών ελαχίστων.  |

Πίνακας 4:Είδη Βελτιστοποιήσεων που υποστηρίζει το Optimization Tool

Στην συγκεκριμένη μελέτη έγινε χρήση του Optimization Tool και ως αλγόριθμος βελτιστοποίησης χρησιμοποιήθηκε ο γενετικός αλγόριθμος ga.

## 4.2 Δεδομένα Γενετικού Αλγόριθμου

Επιλέξαμε να χρησιμοποιήσουμε 3 διαφορετικές συναρτήσεις καταλληλότητας, ώστε να δοκιμάσουμε την διαδικασία σε μια ποιο διευρυμένη βάση. Από την προηγούμενη ανάλυση που έχει περιγράψει με λεπτομέρεια στην παράγραφο 2,7, επιλέχθηκαν 3 κατηγορίες συνάρτησης καταλληλότητας :

1. Συνάρτηση Καταλληλότητας Συμπεριφοράς
2. Προσαρμοσμένη Συνάρτηση Καταλληλότητας
3. Ολοκληρωτική ή Αθροιστική Συνάρτηση Καταλληλότητας

Πέρα από την διαφορετική συνάρτηση καταλληλότητας όλες οι υπόλοιπες μεταβλητές στον γενετικό αλγόριθμο παρέμειναν ίδιες ώστε να μπορούμε να έχουμε την δυνατότητα της σύγκρισης. Με τον τρόπο αυτό μας παρέχεται και η δυνατότητα να καταλήξουμε και σε ένα συμπέρασμα για το ποια είναι η καλύτερη συνάρτηση καταλληλότητας για την βελτιστοποίηση ασαφή ελεγκτή τροχήλατου ρομποτικού οχήματος.

Οι μεταβλητές του γενετικού αλγόριθμου παρουσιάζονται παρακάτω στον πίνακα 5. Η παρούσα μορφή είναι αυτή που χρησιμοποιείται στον MatLab. Στην βοήθεια που παρέχεται μαζί με το OptimTool μας παρέχονται οι πληροφορίες πως να συνταχθούν οι τιμές και ο τρόπος που αυτά περνάνε στο OptimTool.

Όπως παρατηρείτε στον πίνακα υπάρχει και μια μεταβλητή που αναφέρεται στα όρια των τιμών που θα κινηθεί ο αλγόριθμός για να βελτιστοποιήσει τις τιμές του ασαφή

ελεγκτή. Οι τιμές παρουσιάζονται στον πίνακα 6 και επεξηγούνται στην παράγραφο 4,3

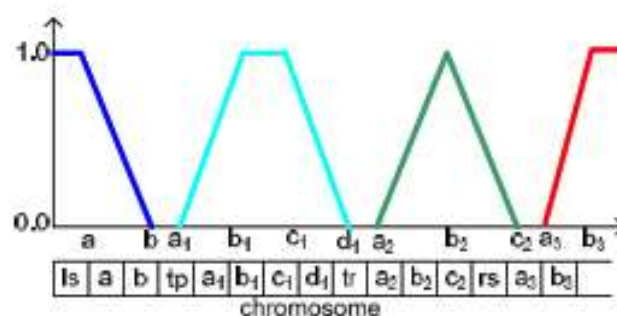
|  |   |
|--|---|
| Μέγεθος πληθυσμού (PopulationSize):  | 117   |
| Αρχικός Πληθυσμός (InitialPopulationMatrix)  | 1x117   |
| Αρχικός πίνακας ορίων (InitialPopulationRange)   | 2x117   |
| Αρχικός πίνακας βαθμολόγησης (InitialScoresMatrix)   | Κενό  |
| Μέγιστος αριθμός γενεών που θα τρέξει η ρουτίνα (MaxGenerations)                           | 200   |
| Μέγιστος αριθμός γενεών χωρίς μεταβολή στην συνάρτηση καταλληλότητας (MaxStallGenerations) | 10  |
| Συνάρτηση δημιουργίας (CreationFcn)  | @gacreationuniform                                |
| Συνάρτηση επιλογής (SelectionFcn)  | @selectionroulette                                |
| Συνάρτηση διασταύρωσης (CrossoverFcn)  | Διασταύρωση μονού σημείου (@crossoversinglepoint) |
| Ποσοστό διασταύρωσης (CrossoverFraction)   | 0.8000  |
| Συνάρτηση Μετάλλαξης (MutationFcn)   | {@mutationuniform [0.0700]}                       |
| Ελιτισμός (EliteCount):  | '0.05*Μέγεθος πληθυσμού                           |
| Συνάρτηση κατάταξης των συναρτήσεων καταλληλότητας (FitnessScalingFcn)                     | @fitscalingrank                                   |
| Συνάρτηση εκτύπωσης (PlotFcn)  | Εκτύπωση καλύτερου και μέσης τιμής {@gaplotbestf} |

Πίνακας 5: Μεταβλητές Γενετικού Αλγόριθμου

### 4.3 Δημιουργία χρωμοσώματος από ελεγκτή ασαφούς λογικής,

Οι παράμετροι ενός ελεγκτή ασαφούς λογικής μπορούν να ομαδοποιηθούν σε τρεις τομείς. Αυτά είναι: (α) Η ασαφοποίηση των εισροών, (β) ασαφείς κανόνες και την (γ) ασαφοποίηση της εξόδου. Αυτοί οι τομείς συνήθως ρυθμίζονται από ένα πείραμα ή το σχεδιασμό. Ωστόσο ένας γενετικός αλγόριθμος μπορεί να εφαρμοστεί σε αυτές τις παραμέτρους ώστε να εξελιχθεί ο ελεγκτής ασαφούς λογικής. Αυτές οι παράμετροι μπορούν να παρουσιαστούν σε ένα χρωμόσωμα με διάφορους τρόπους. Ένα χρωμόσωμα μιας συνάρτησης συμμετοχής απεικονίζεται στην εικόνα 82.

Οι παράμετροι στο χρωμόσωμα είναι οι τιμές των ορίων του εκάστοτε σχήματος. Στο συγκεκριμένο παράδειγμα τα σχήματα που έχουμε (από τα αριστερά προς τα δεξιά) είναι: **ls** (left shoulder - αριστερός ώμος), **tp** (trapezoid -τραπεζοειδής), **tr** (triangle - τρίγωνο) και **rs** (right shoulder - δεξιός ώμος).



Εικόνα 82 : Δημιουργία Χρωμοσώματος Ασαφούς Ελεγκτή

Οι παράμετροι του σχήματος δίνονται από τα σημεία στα οποία το σχήμα αλλάζει σε σχέση με το μηδέν στον άξονα x. Οι ασαφείς κανόνες μπορούν επίσης να κωδικοποιηθούν είτε ως αριθμητικές τιμές ή σύμβολα που μετατρέπονται σε γλωσσικούς κανόνες.

Στην προκειμένη μελέτη, δεν γίνεται βελτιστοποίηση των σχημάτων. Δηλαδή δεν μεταβάλλουμε τα αρχικά στοιχεία του ελεγκτή μέσω του γενετικού αλγόριθμου, υπάρχει μια μεταβολή στο εύρος των σχημάτων, μέσω της μεταβολής των τιμών, αλλά όχι του τύπου του σχήματος. Ο κανόνας που καθορίζει το σχήμα είναι ότι :  $a < b < a_1 \dots < b_3$ .

Επίσης ο γενετικός αλγόριθμος δεν μεταβάλλει τους κανόνες που έχουμε θεσπίσει στον ασαφή ελεγκτή.

Το χρωμόσωμα που προκύπτει από τον ασαφή εκλεκτή είναι :

Angle: 17 τιμές

|      |      |        |         |        |        |   |        |   |       |   |       |       |        |       |     |     |
|------|------|--------|---------|--------|--------|---|--------|---|-------|---|-------|-------|--------|-------|-----|-----|
| -3.5 | -3.5 | -1.114 | -0.8913 | -1.114 | -0.557 | 0 | -0.557 | 0 | 0.557 | 0 | 0.557 | 1.114 | 0.8913 | 1.114 | 3.5 | 3.5 |
|------|------|--------|---------|--------|--------|---|--------|---|-------|---|-------|-------|--------|-------|-----|-----|

Sensor : 12 τιμές x 5

|   |      |     |     |     |      |     |   |   |     |   |   |
|---|------|-----|-----|-----|------|-----|---|---|-----|---|---|
| 0 | 0,05 | 0,5 | 0,6 | 0,5 | 0,55 | 0,9 | 1 | 1 | 1,1 | 2 | 2 |
|---|------|-----|-----|-----|------|-----|---|---|-----|---|---|

Motor Speed : 20 τιμές x 2

|   |   |   |   |   |   |   |   |   |   |   |    |   |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|---|----|---|----|----|----|----|----|----|----|
| 0 | 0 | 2 | 3 | 2 | 3 | 5 | 6 | 5 | 6 | 9 | 10 | 9 | 10 | 12 | 13 | 12 | 13 | 15 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|----|---|----|----|----|----|----|----|----|

Άρα ο τελικός αριθμός των στοιχείων του χρωμοσώματος είναι:

$$17 + (12 \times 5) + (20 \times 2) = 17 + 60 + 40 = 117.$$

Επίσης ορίστηκαν και τα όρια τα οποία θα κινούνταν οι τιμές των μεταβλητών. Οι τιμές αυτές μας διασφαλίζουν ότι η γωνία είναι από π έως -π, οι αισθητήρες θα έχουν δραστικό πεδίο έως 2m και οι μέγιστη ταχύτητα του οχήματος δεν θα ξεπεράσει το 1.2m/sec που είναι και η πραγματική του μέγιστη ταχύτητα. Οι τιμές αυτές παρουσιάζονται στον πίνακα 6.

|                                     |              |
|-------------------------------------|--------------|
| <u>Angle - Γωνία Απόκλισης</u>      | -3,5 εως 3,5 |
| <u>Sensor - Αισθητήρες</u>          | 0 εως 2      |
| <u>Motor Speed - Ταχύτητα Μοτέρ</u> | 0 εως 15     |

Πίνακας 6 : Όρια τιμών μεταβλητών χρωμοσώματος

#### 4.4 Αποτελέσματα Εξελικτικής Διαδικασίας : Συνάρτηση καταλληλότητας Συμπεριφοράς (Behavioral)

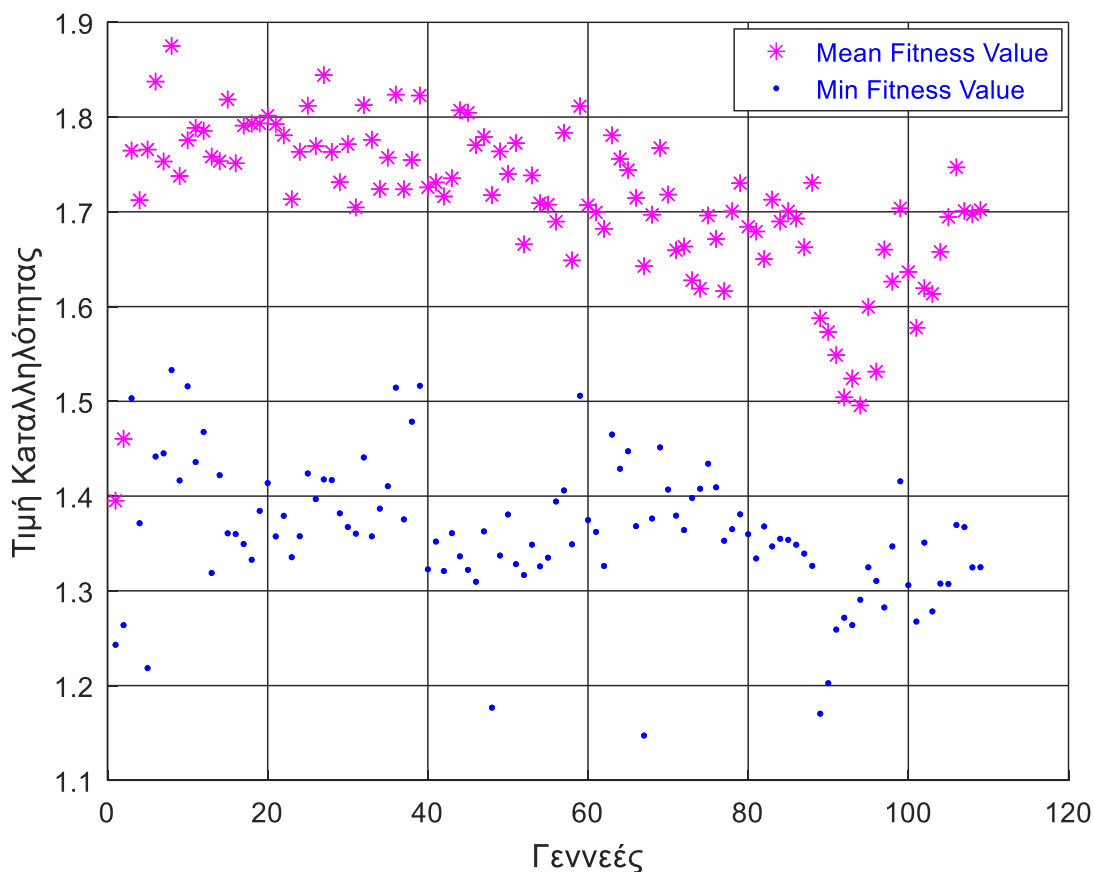
Η συνάρτηση καταλληλότητας συμπεριφοράς αξιολογεί μόνο τα στοιχεία από την συμπεριφορά των οχημάτων κατά την διάρκεια της προσομοίωσης. Έτσι μετρείται ο μέσος όρος των τιμών που δίνουν οι αισθητήρες  $S$  και η μέση ταχύτητα των οχημάτων. Προσβλέποντας ότι το μικρότερο είναι το καλύτερο (ελαχιστοποίηση της συνάρτησης) η συνάρτηση έχει την ακόλουθη μορφή:

$$f = \frac{2.4}{(S + U)}$$

2.4 είναι η μέγιστη τιμή του αθροίσματος των μέγιστων τιμών που μπορεί να λάβουν οι αισθητήρες  $S = 1.2$  & η μέγιστη τιμή που μπορεί να λάβει η ταχύτητα του οχήματος  $U = 1.2$

$S$  = μέση τιμή των τιμών που επιστρέφουν όλοι η αισθητήρες frD, lfD, rtD, frlfD, frrtD

$U$  = μέση τιμή ταχύτητας οχήματος



Εικόνα 83 : Εξέλιξη με χρήση της συνάρτησης συμπεριφοράς (Μέση τιμή - Καλύτερη τιμή)

Στο συγκεκριμένο πείραμα τα τρία οχήματα παίρνουν εντολή να κινηθούν διαδοχικά προς τρία σημεία-στόχο. Όποιο από τα οχήματα προσεγγίσει πρώτο το σημείο-στόχο δίνει εντολή στα άλλα οχήματα να κινηθούν προς το επόμενο σημείο-στόχο μέχρι όλα τα σημεία-στόχος να προσεγγιστούν από κάποιο όχημα.

Παρατηρούμε ότι κατά την διάρκεια του πειράματος ο αλγόριθμος βελτιώνει την τιμή της συνάρτησης καταλληλότητας, όμως η διακύμανση που παρουσιάζει η καλύτερη τιμή της οφείλεται στην δυναμική του πειράματος, μέσω των κινούμενων οχημάτων που επηρεάζουν το ένα το άλλο. Η συνάρτηση μετράει τα στοιχεία λειτουργίας των 3 οχημάτων και μας δίνει την μέση τιμή.

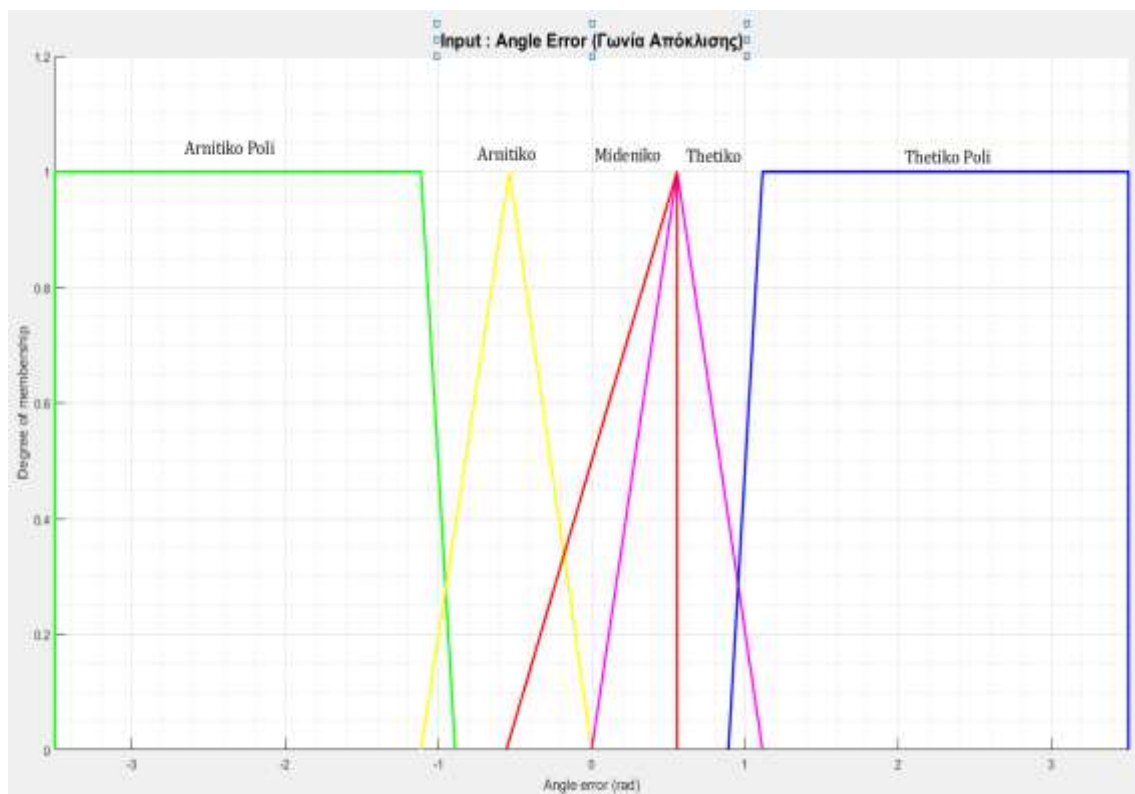
Η τάση είναι προς την ελαχιστοποίηση της τιμής της συνάρτησης καταλληλότητας. Επίσης και ο μέσος όρος των τιμών της συνάρτησης καταλληλότητας παρουσιάζει τάση βελτίωσης. Ο αριθμός των γενεών που έτρεξε ο αλγόριθμος είναι μικρός με αποτέλεσμα το γράφημα μα μην έχει την επιθυμητή μορφή.

Επίσης φαίνεται και το πρόβλημα της ασυνέχειας του αλγορίθμου λόγω διακοπής και επανεκκίνησης της διαδικασίας βελτιστοποίησης.

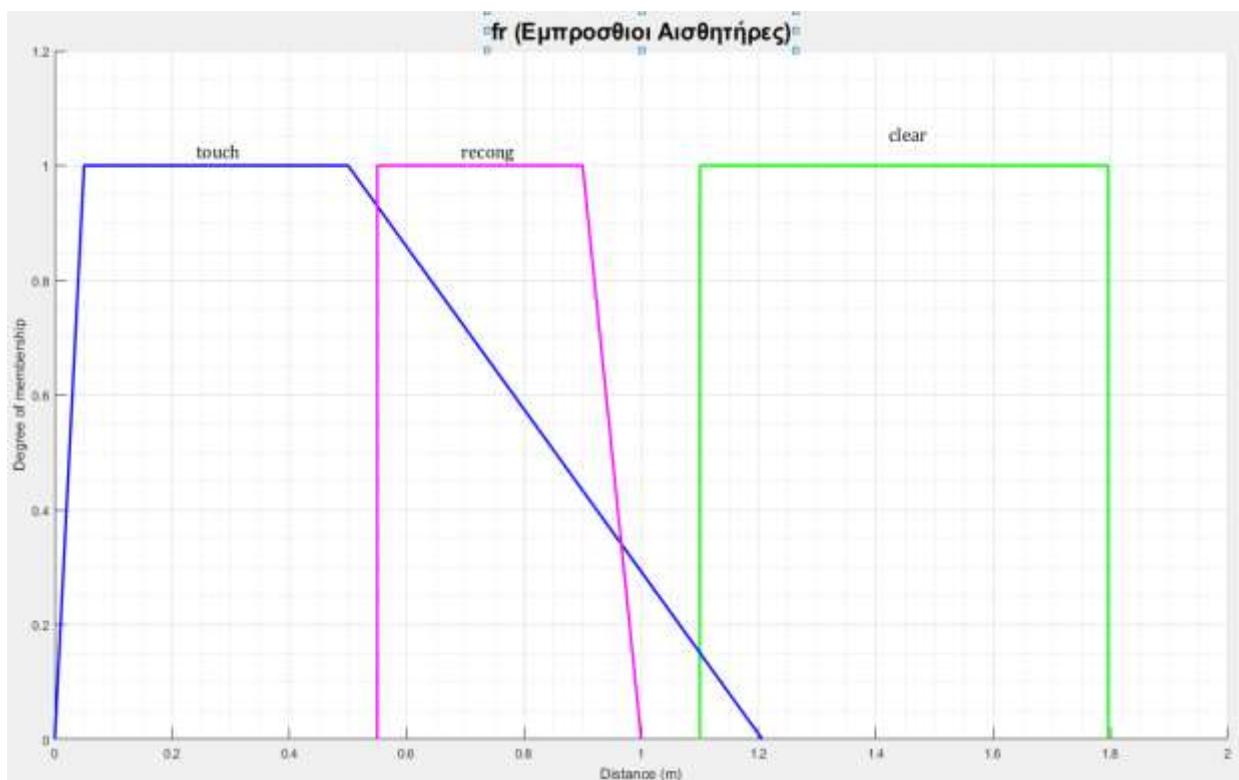
Παρότι όμως δεν έχει ολοκληρωθεί ο αλγόριθμος, εξάγουμε τους κάτωθι ελεγκτές:

1. Τον ελεγκτή με την καλύτερη τιμή της συνάρτησης καταλληλότητας.
2. Τον καλύτερο ελεγκτή στην γενεά με τον καλύτερο μέσο όρο της τιμής της συνάρτησης καταλληλότητας.

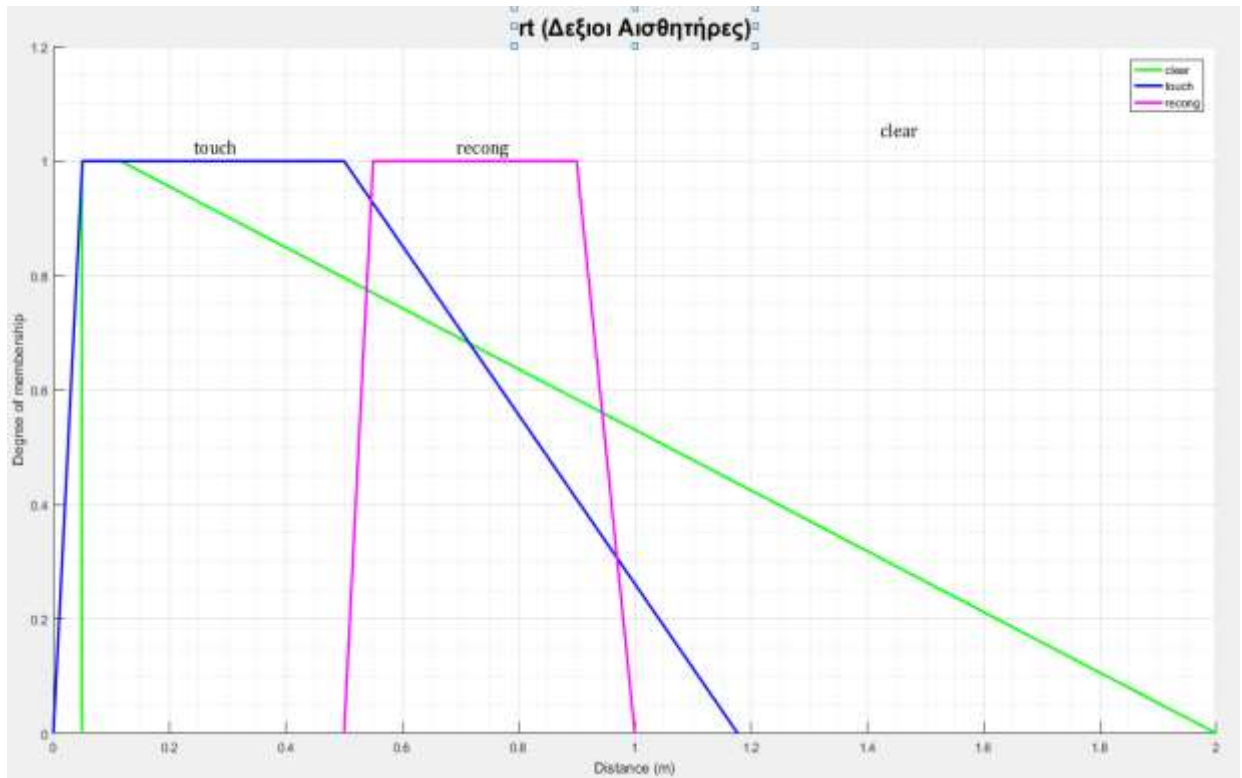
Ελεγκτής : καλύτερη τιμή συνάρτηση καταλληλότητας συμπεριφοράς



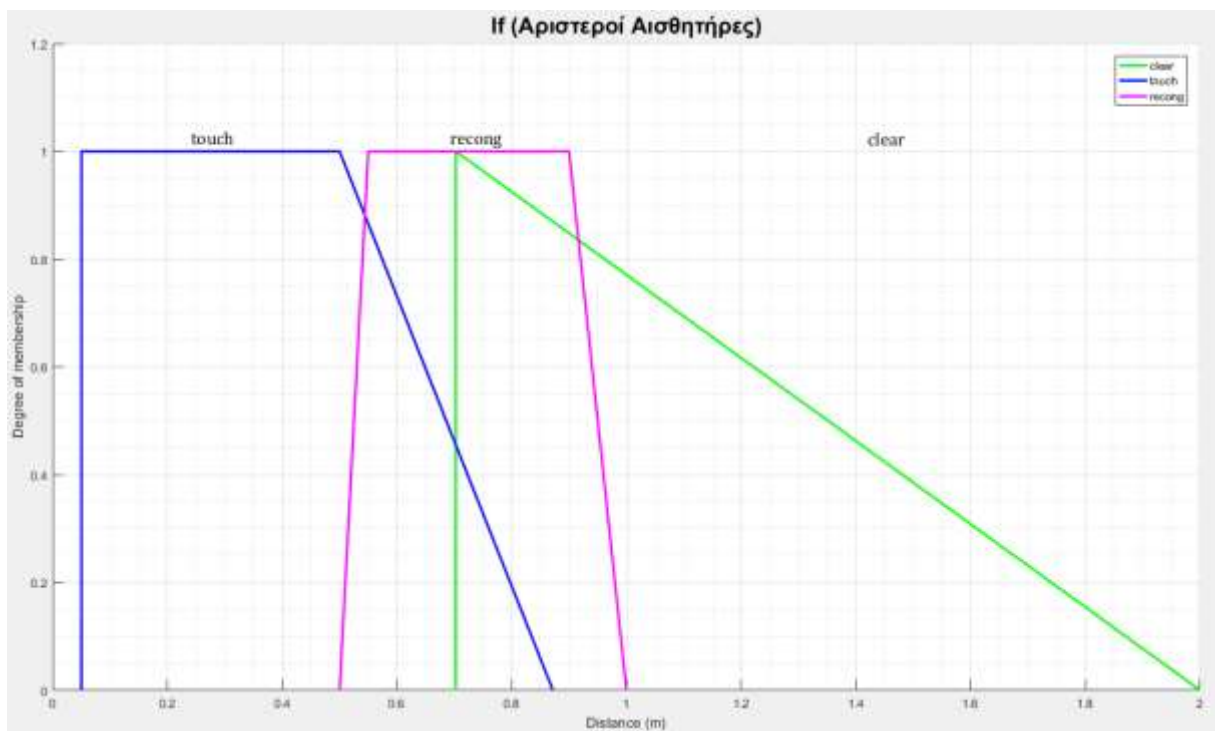
Εικόνα 84 : Είσοδος Γωνία Απόκλισης - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς



Εικόνα 85 : Είσοδος Εμπρόσθιοι Αισθητήρες - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς

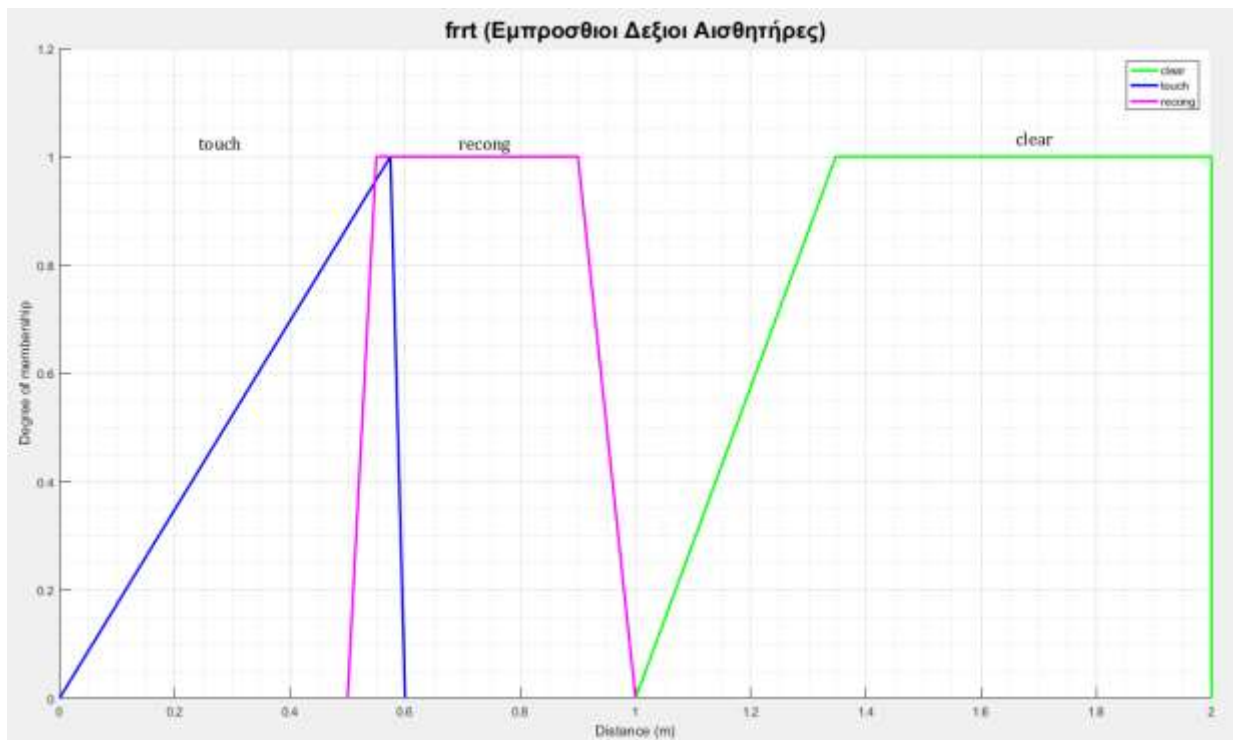


Εικόνα 86 : Είσοδος Δεξιοί Αισθητήρες - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς

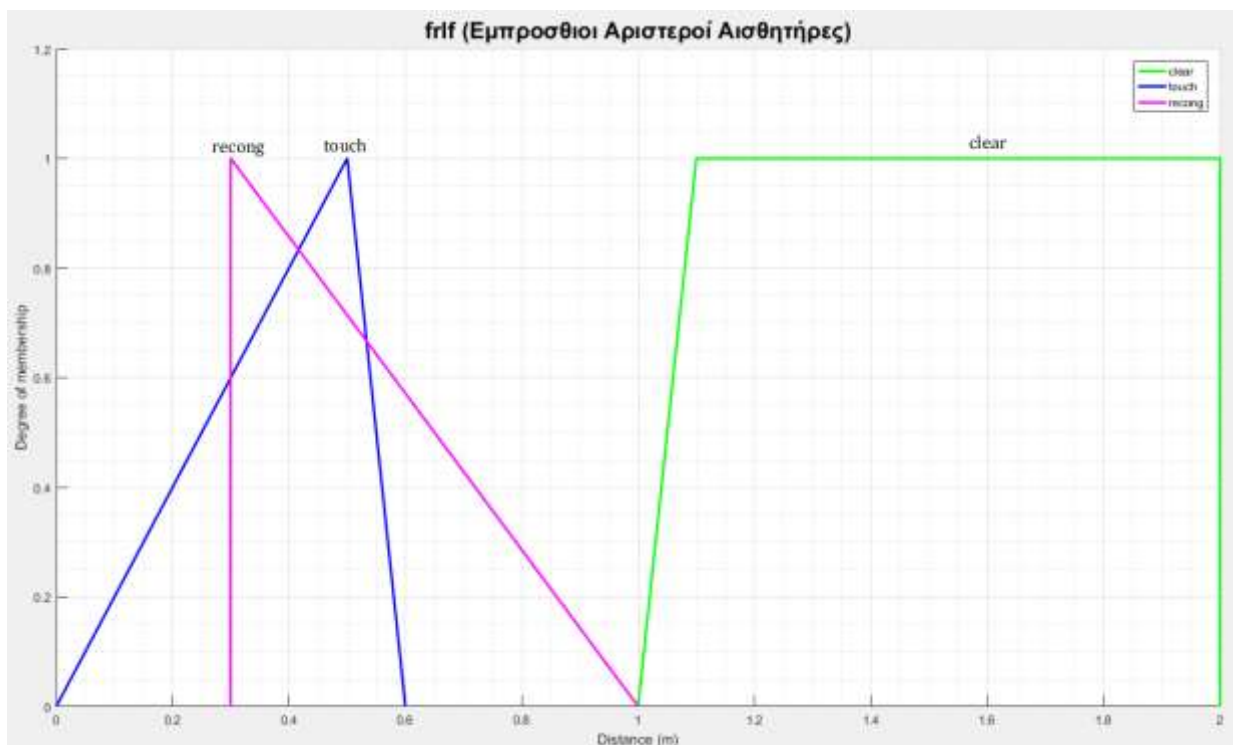


Εικόνα 87: Είσοδος Αριστεροί Αισθητήρες - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς

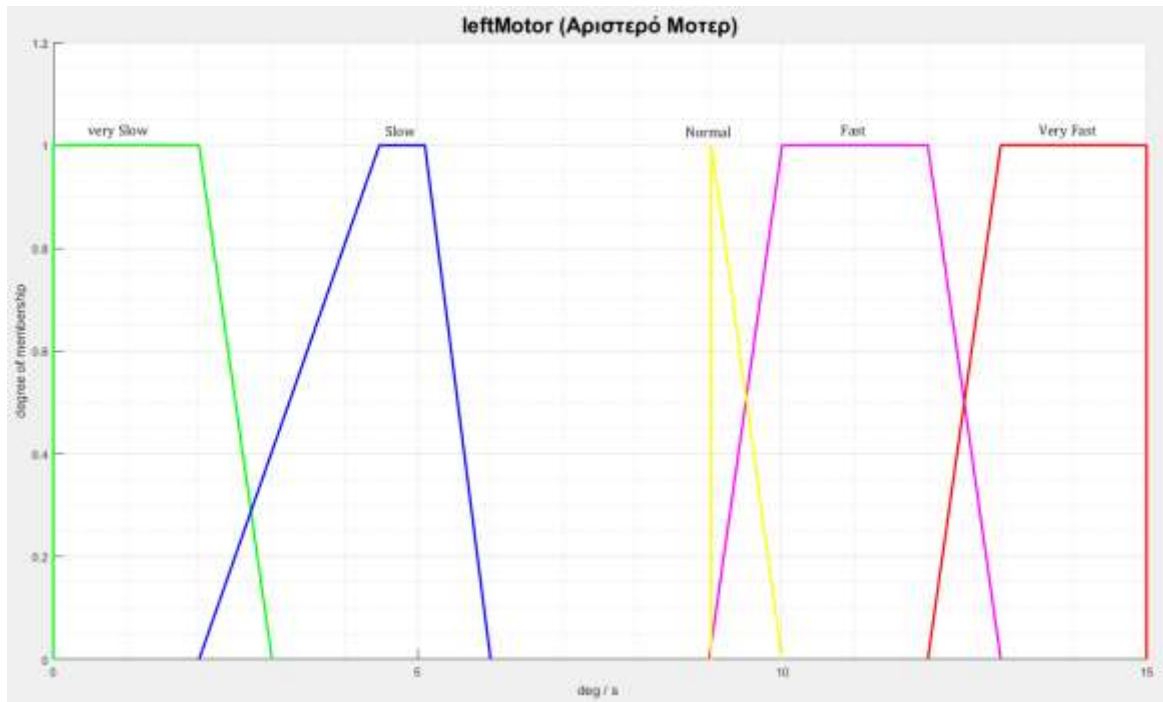




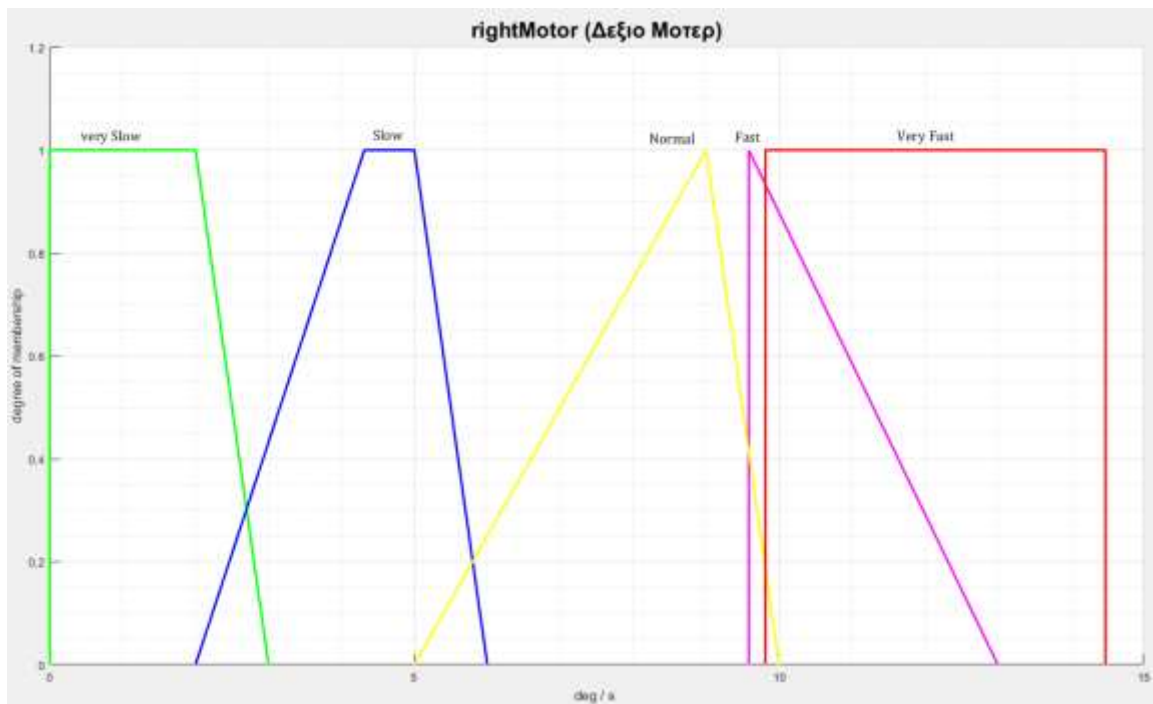
Εικόνα 88: Είσοδος Εμπρόςθιοι Δεξιοί Αισθητήρες - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς



Εικόνα 89: Είσοδος Εμπρόςθιοι Αριστεροί Αισθητήρες - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς

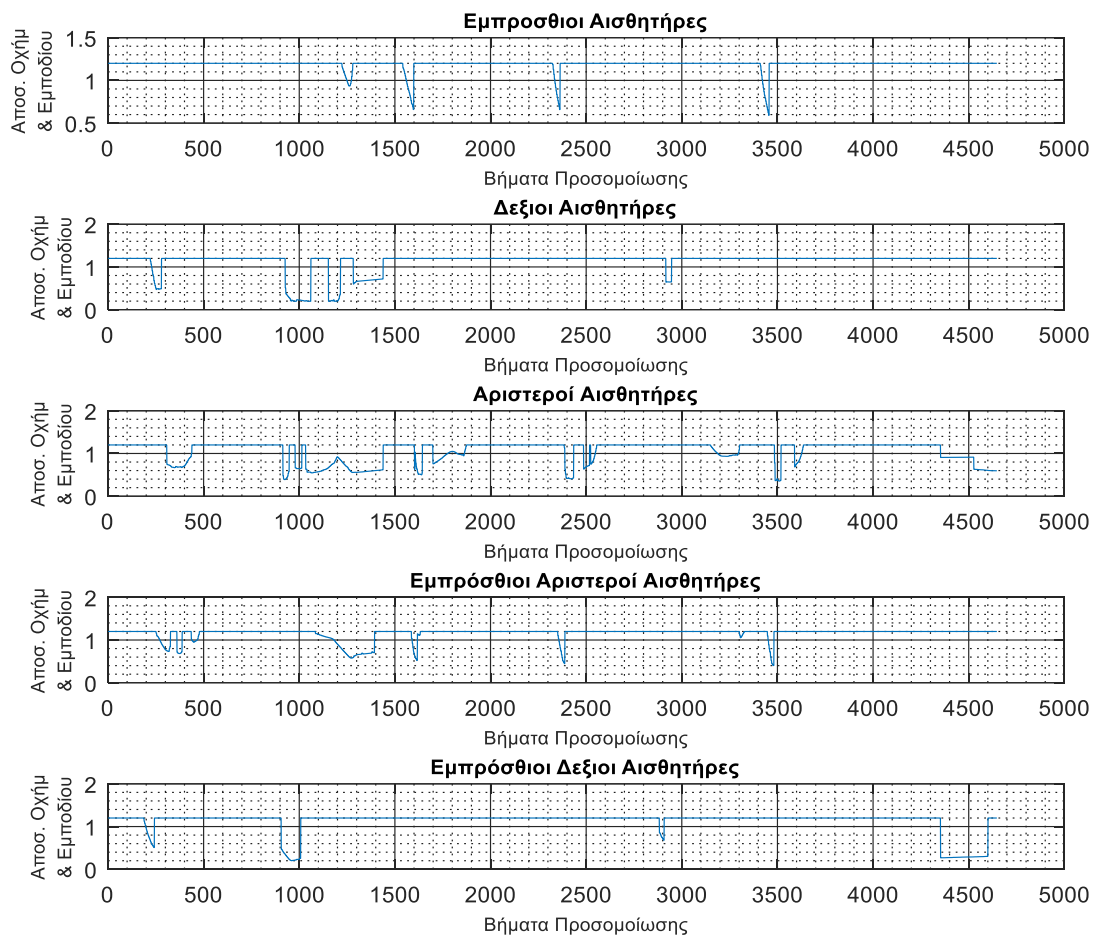


Εικόνα 90: Έξοδος Ταχύτητες Αριστερό Μοτέρ - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς

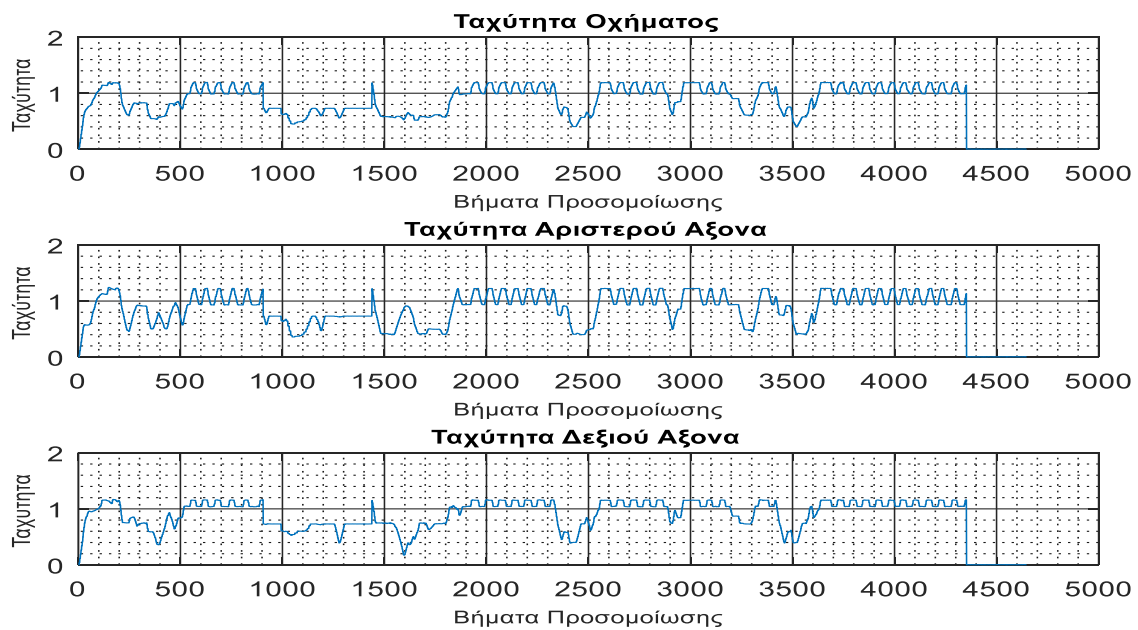


Εικόνα 91: Έξοδος Ταχύτητες Δεξιού Μοτέρ - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς

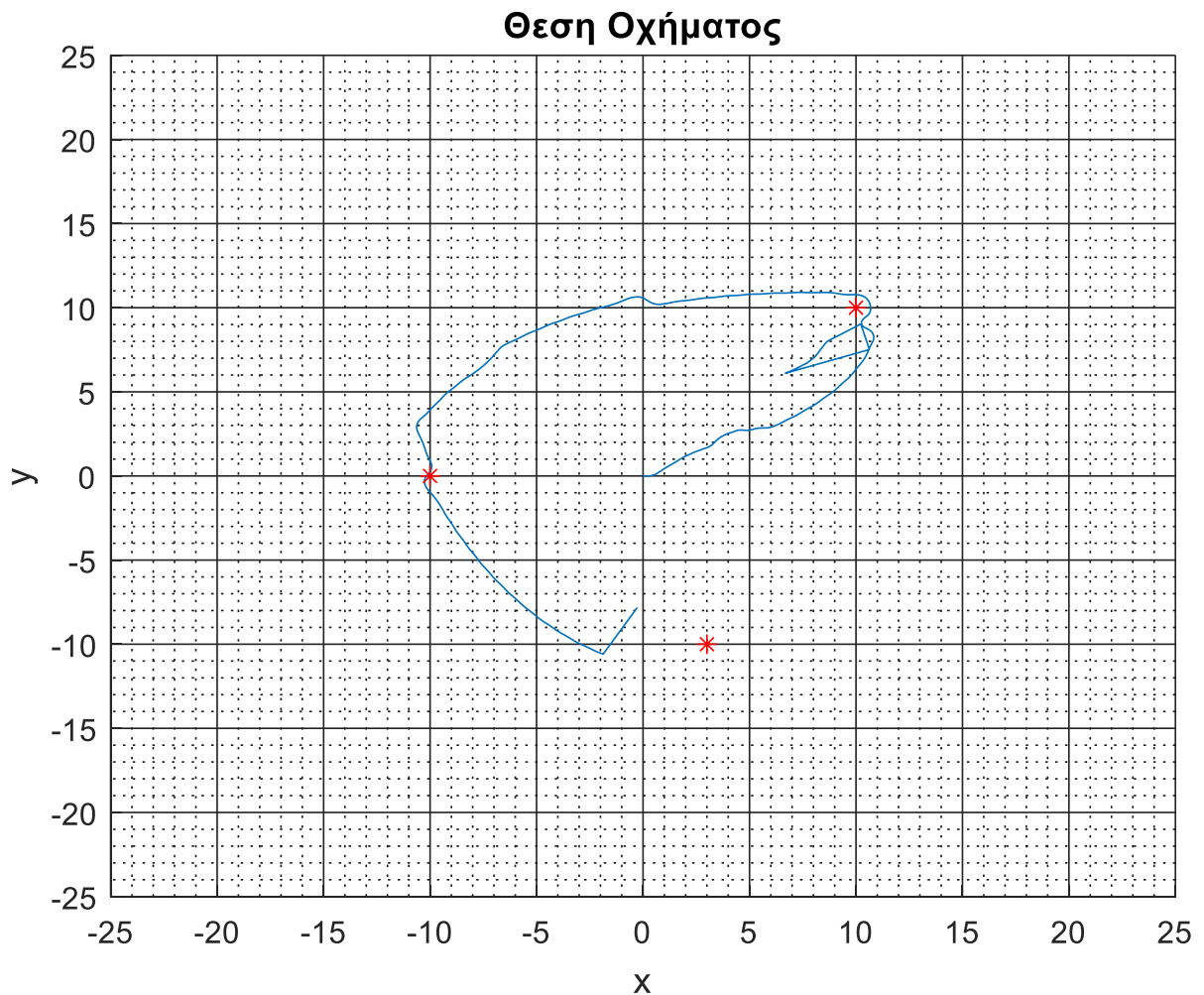
## Αποτελέσματα προσομοίωσης 1<sup>ου</sup> Οχήματος



Εικόνα 92: Στοιχεία Αισθητήρων 1ου Οχήματος - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς



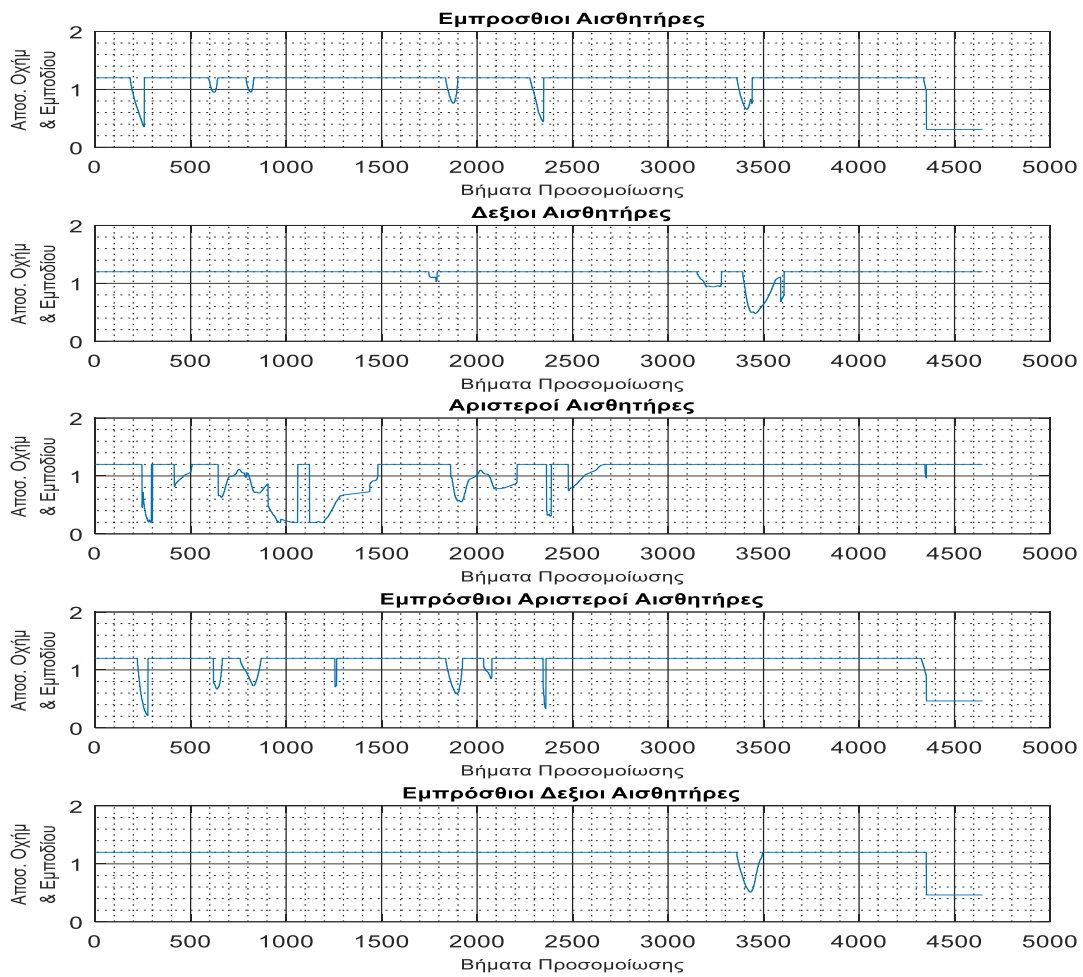
Εικόνα 93 : Ταχύτητες 1ου Οχήματος - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς



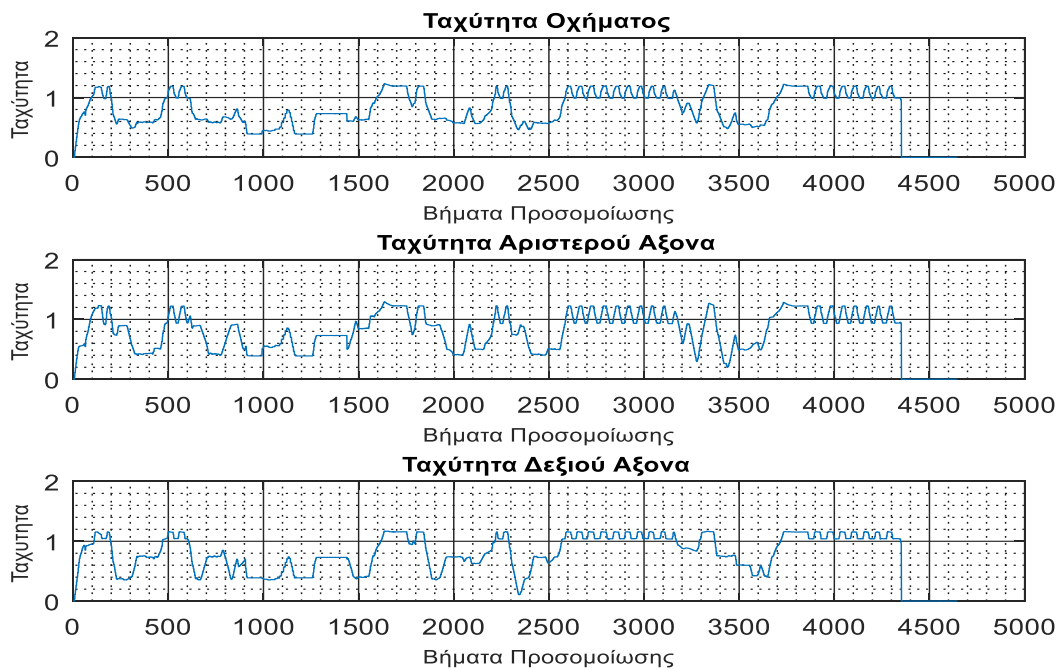
Εικόνα 94: Διαδρομή 1ου οχήματος - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς

Στο διάγραμμα των αισθητήρων του 1<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται όλοι οι αισθητήρες άλλα για μικρότερο χρονικό διάστημα από το 1<sup>ο</sup> όχημα του αρχικού ελεγκτή και αποφεύγει τα εμπόδια, που σημαίνει ότι ο ελεγκτής έχει βελτιστοποιηθεί στο κομμάτι της αποφυγής των εμποδίων. Αντίστοιχα παρατηρούμε και ότι η ταχύτητα του οχήματος διατηρείται σε υψηλότερη τιμή από το 1<sup>ο</sup> όχημα της αρχικής έκδοσης του ελεγκτή παρότι παρουσιάζει αυξομειώσεις αλλά σε πολύ υψηλές τιμές. Στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία.

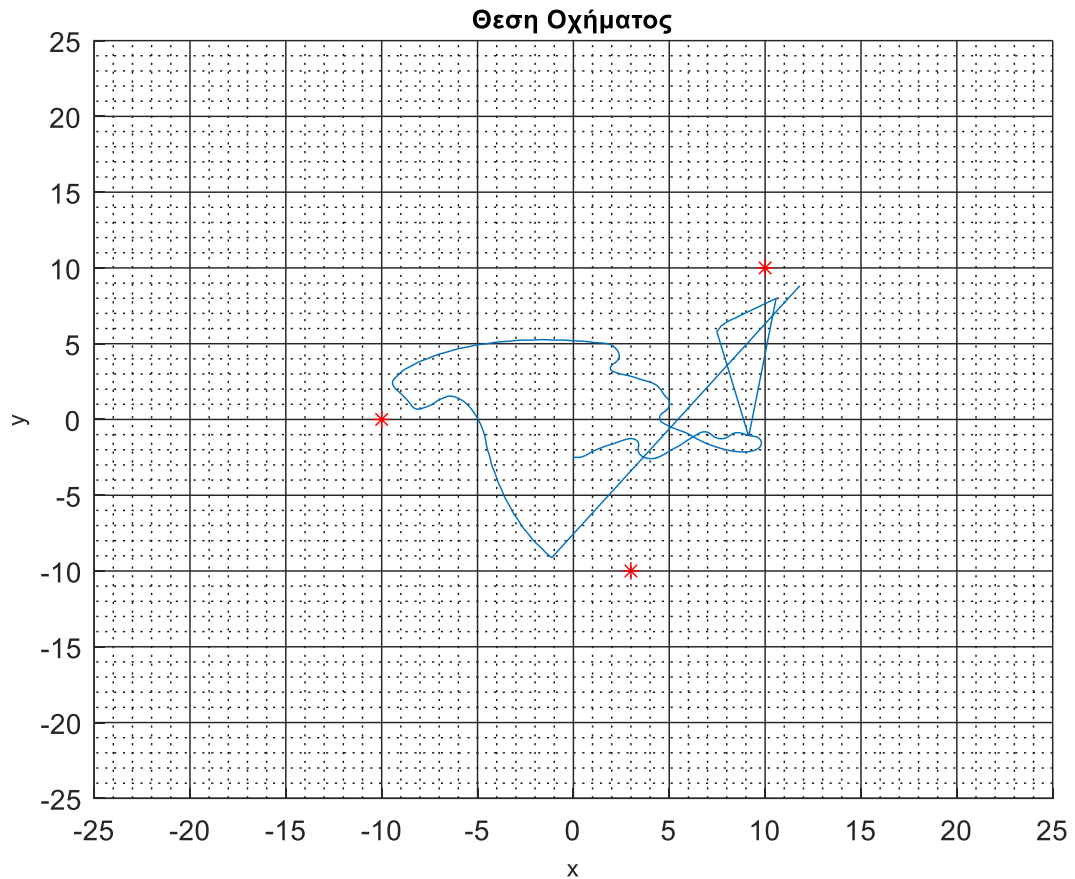
Αποτελέσματα προσομοίωσης 2<sup>ου</sup> Οχήματος



Εικόνα 95: Στοιχεία Αισθητήρων 2ου οχήματος - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς



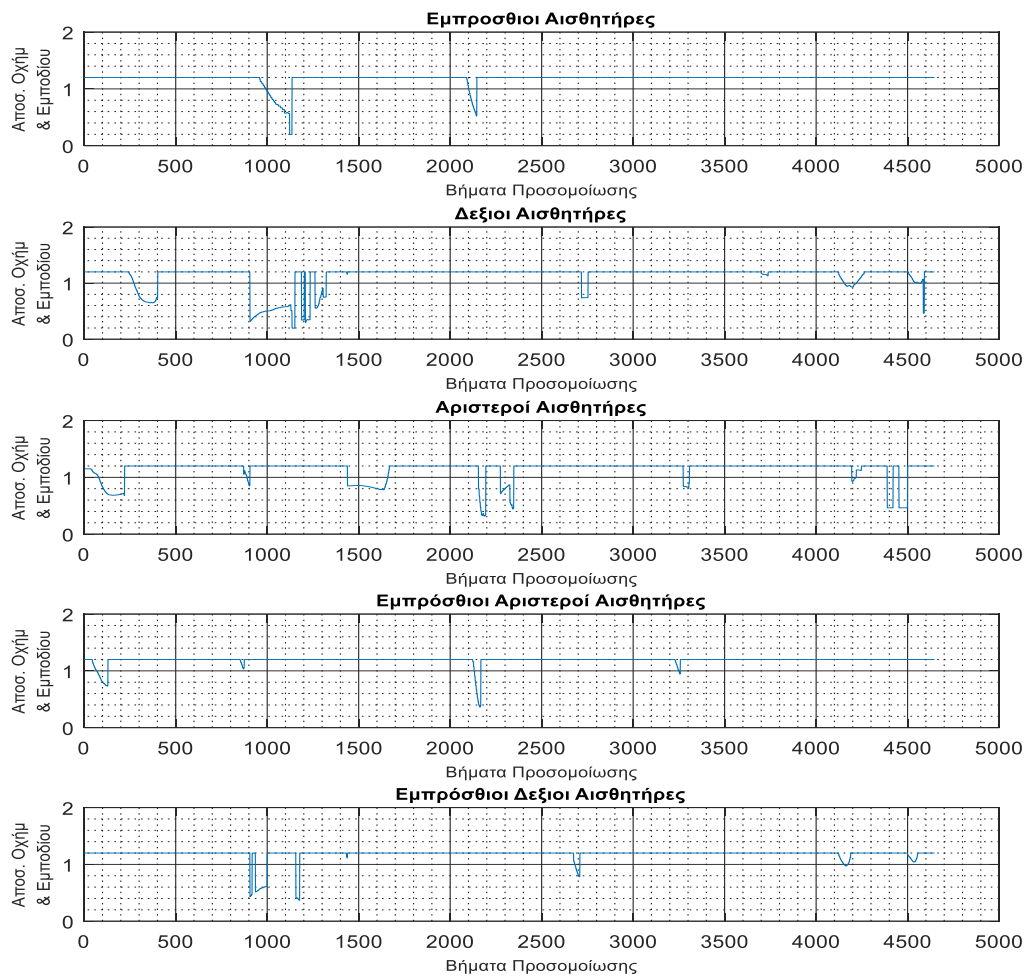
Εικόνα 96: Ταχύτητες 2ου οχήματος - Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς



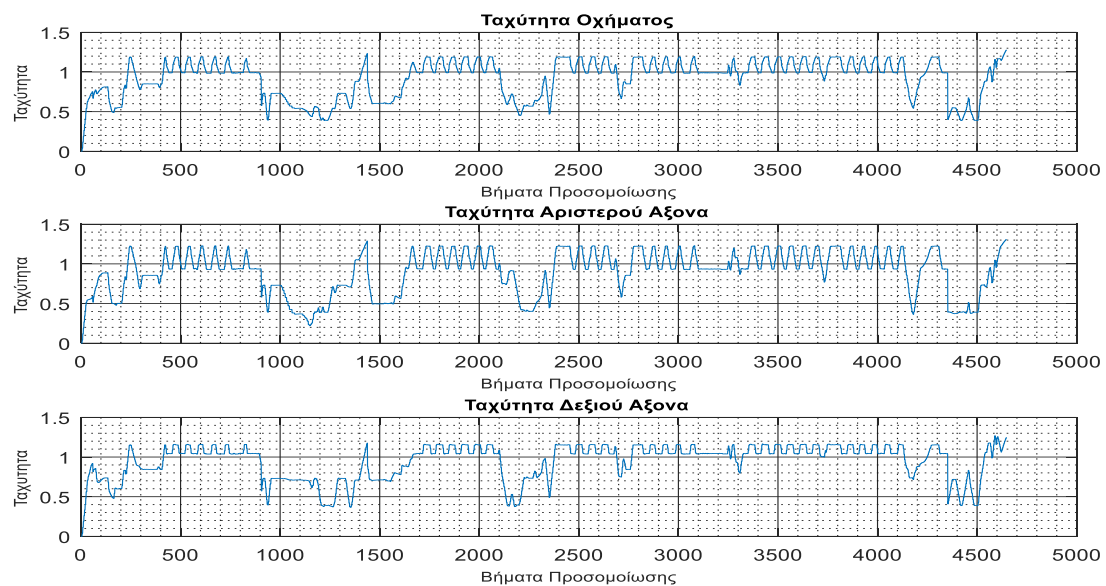
Εικόνα 97: Διαδρομή 2ου οχήματος – Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς

Στο διάγραμμα των αισθητήρων του 2<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται όλοι οι αισθητήρες άλλα για μικρότερο χρονικό διάστημα από το 1<sup>ο</sup> όχημα του αρχικού ελεγκτή και αποφεύγει τα εμπόδια, που σημαίνει ότι ο ελεγκτής έχει βελτιστοποιηθεί στο κομμάτι της αποφυγής των εμποδίων. Αντίστοιχα παρατηρούμε και ότι η ταχύτητα του οχήματος διατηρείται σε υψηλότερη τιμή από το 2<sup>ο</sup> όχημα της αρχικής έκδοσης του ελεγκτή παρότι παρουσιάζει αυξομειώσεις αλλά σε πολύ υψηλές τιμές. Στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία.

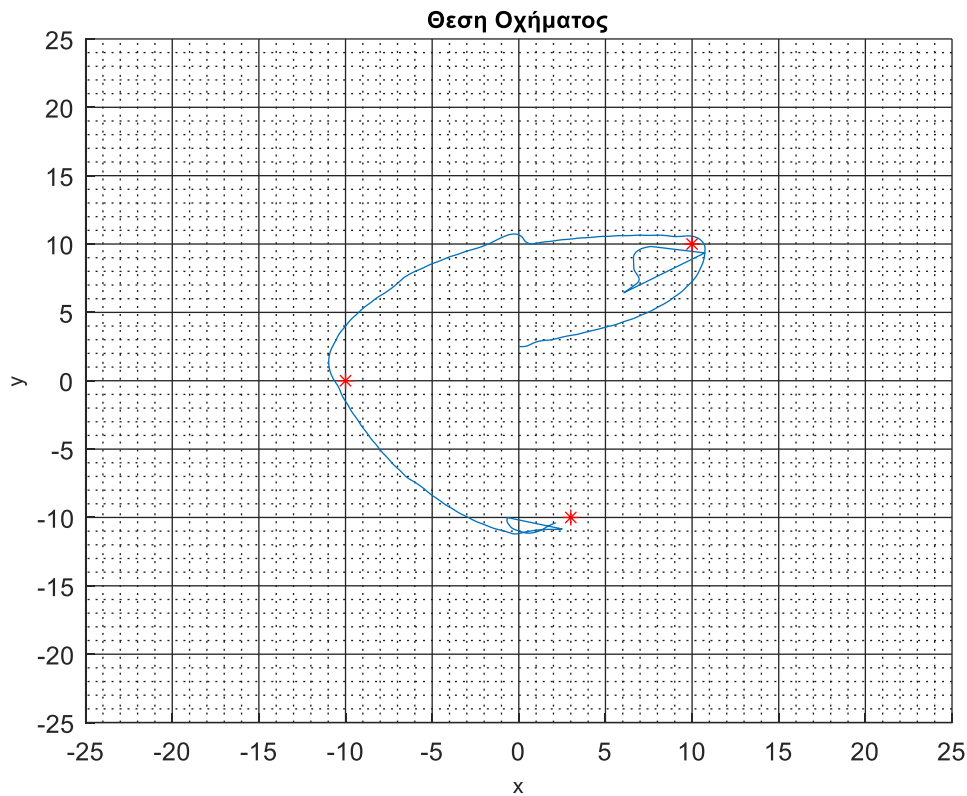
## Αποτελέσματα προσομοίωσης 3<sup>ου</sup> Οχήματος



Εικόνα 98: Δεδομένα Αισθητήρων 3ου οχήματος – Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς



Εικόνα 99: Ταχύτητες 3ου οχήματος – Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς

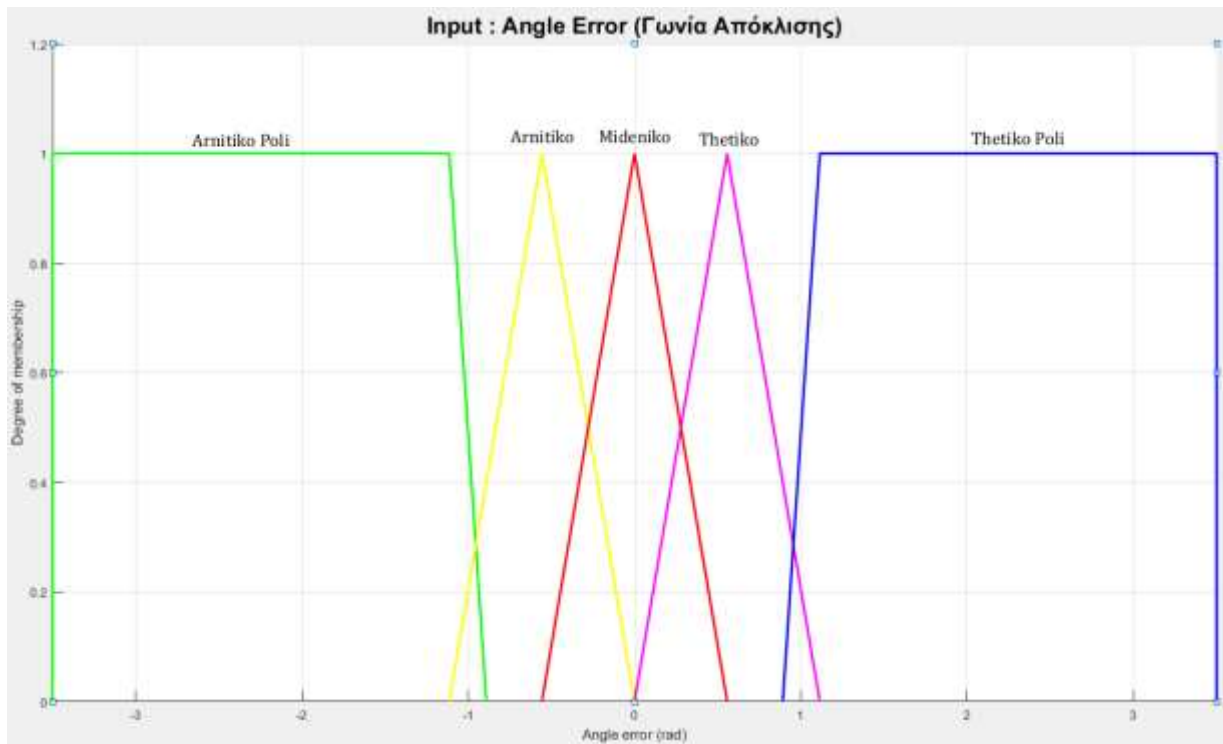


Εικόνα 100: Διαδρομή 3ου οχήματος – Ελεγκτής με την καλύτερη τιμή στη συνάρτηση καταλληλότητας συμπεριφοράς

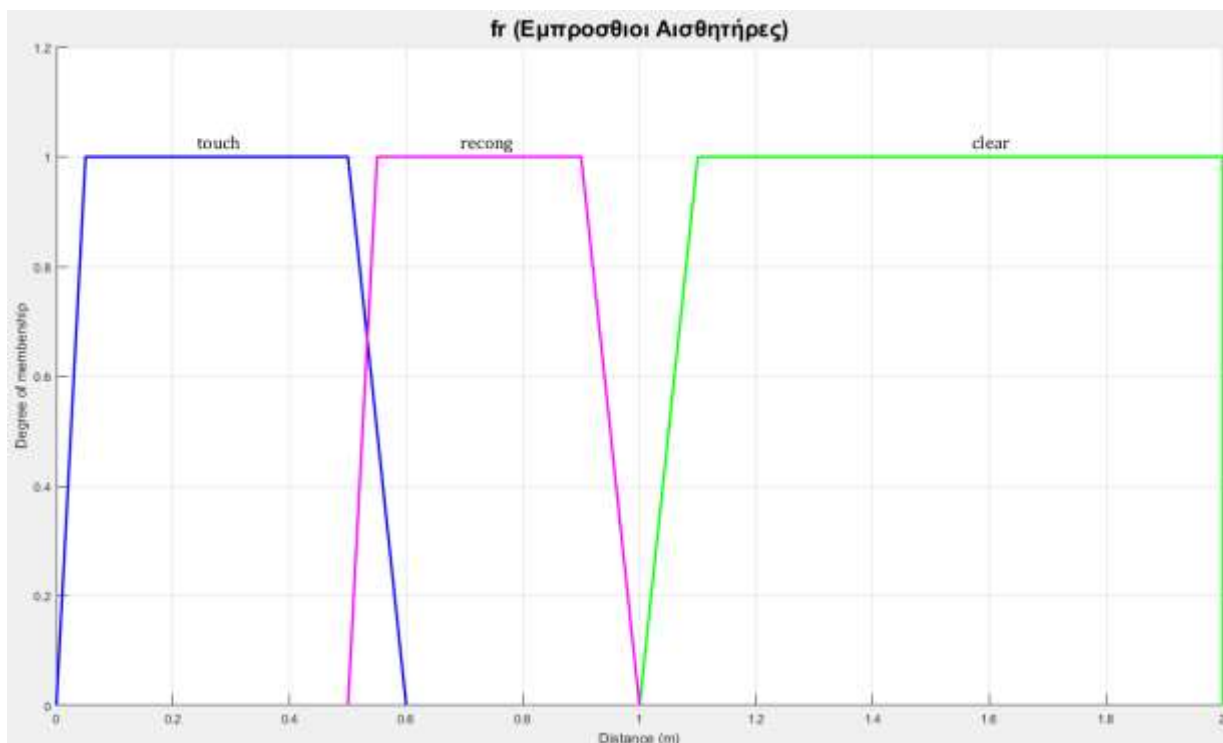
Στο διάγραμμα των αισθητήρων του 3<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται όλοι οι αισθητήρες άλλα για πολύ μικρότερο χρονικό διάστημα από το 3<sup>ο</sup> όχημα του αρχικού ελεγκτή και αποφεύγει τα εμπόδια, που σημαίνει ότι ο ελεγκτής έχει βελτιστοποιηθεί στο κομμάτι της αποφυγής των εμποδίων. Αντίστοιχα παρατηρούμε και ότι η ταχύτητα του οχήματος διατηρείται σε υψηλότερη τιμή από το 3<sup>ο</sup> όχημα της αρχικής έκδοσης του ελεγκτή παρότι παρουσιάζει αυξομειώσεις αλλά σε πολύ υψηλές τιμές. Στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία.



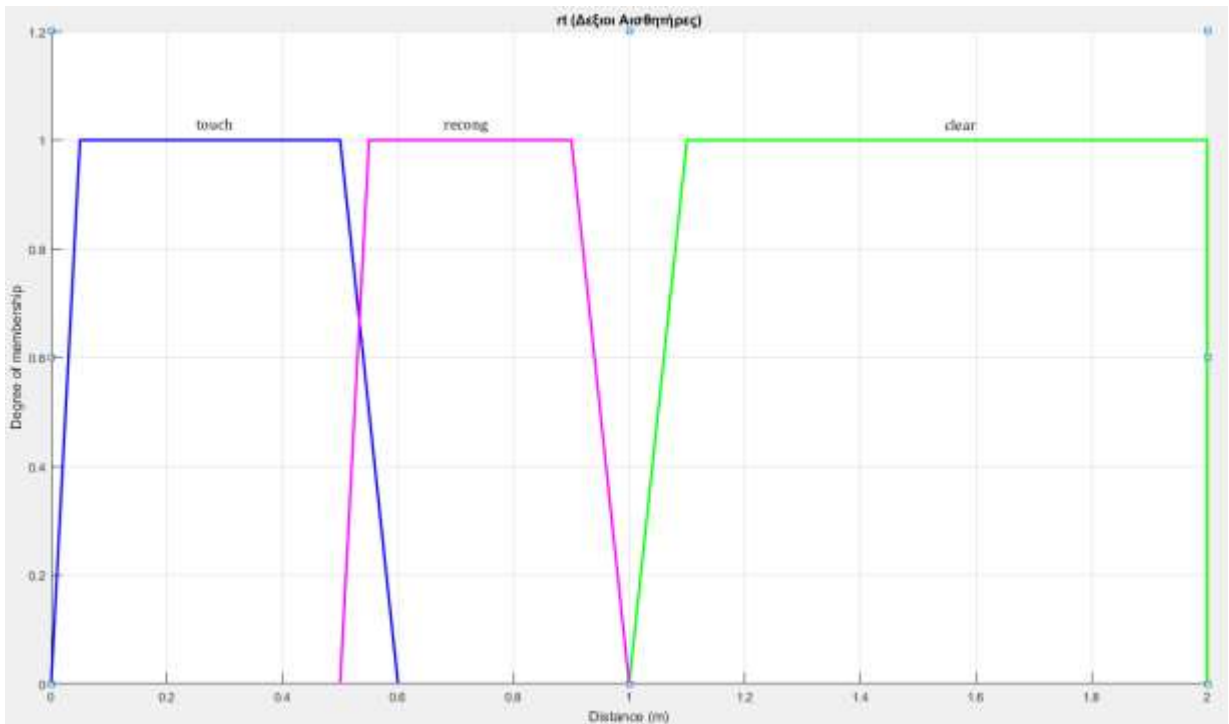
Ελεγκτή : Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς



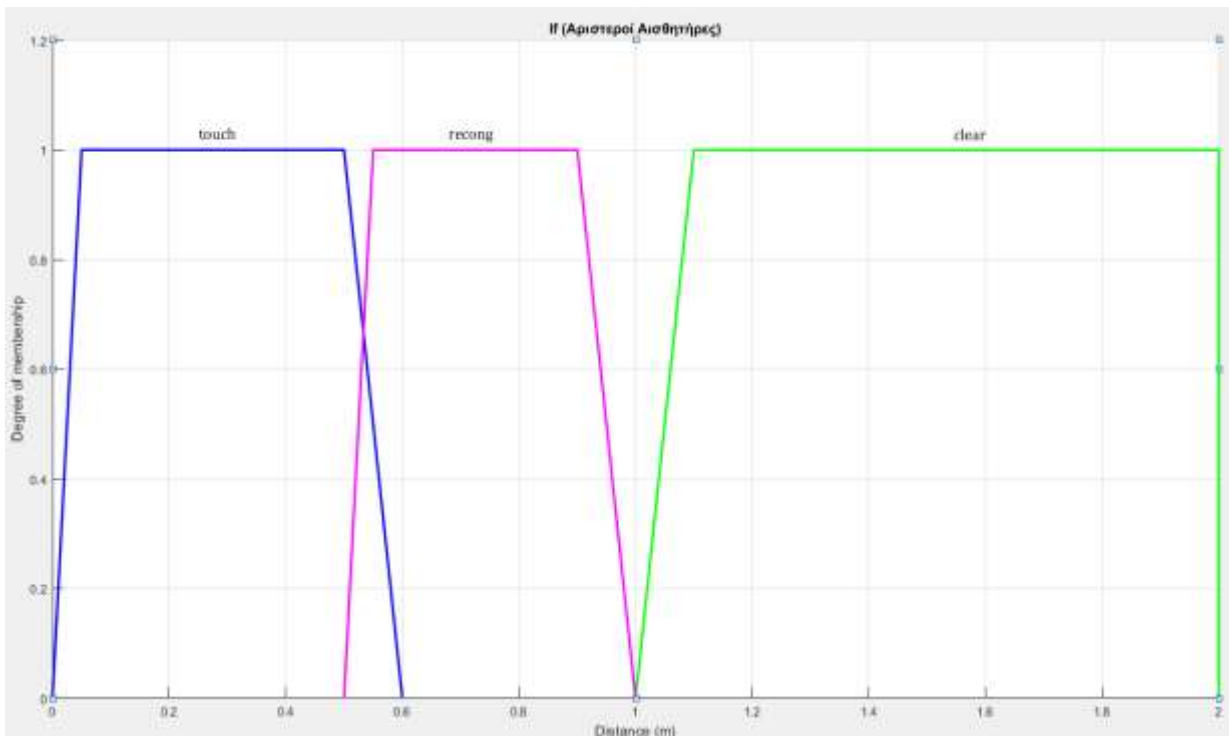
Εικόνα 101 : Είσοδος Γωνία Απόκλισης – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς



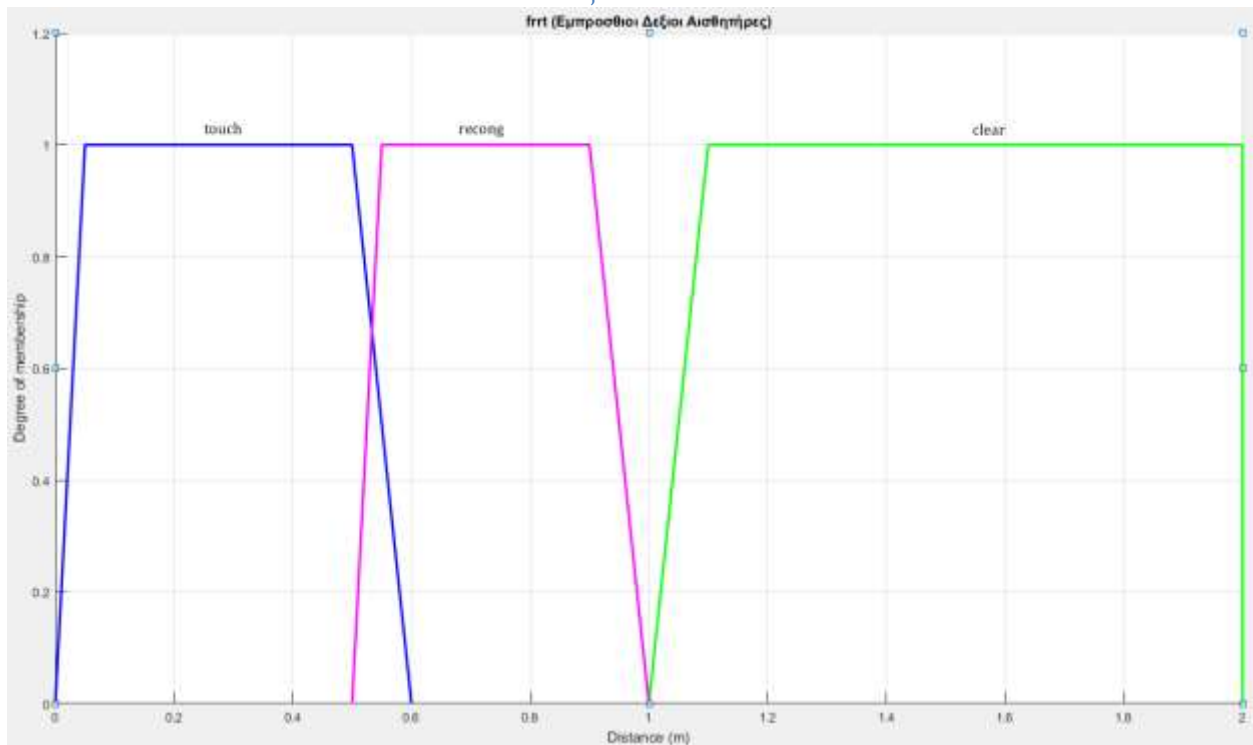
Εικόνα 102 : Είσοδος Εμπροσθιοι Αισθητήρες – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς



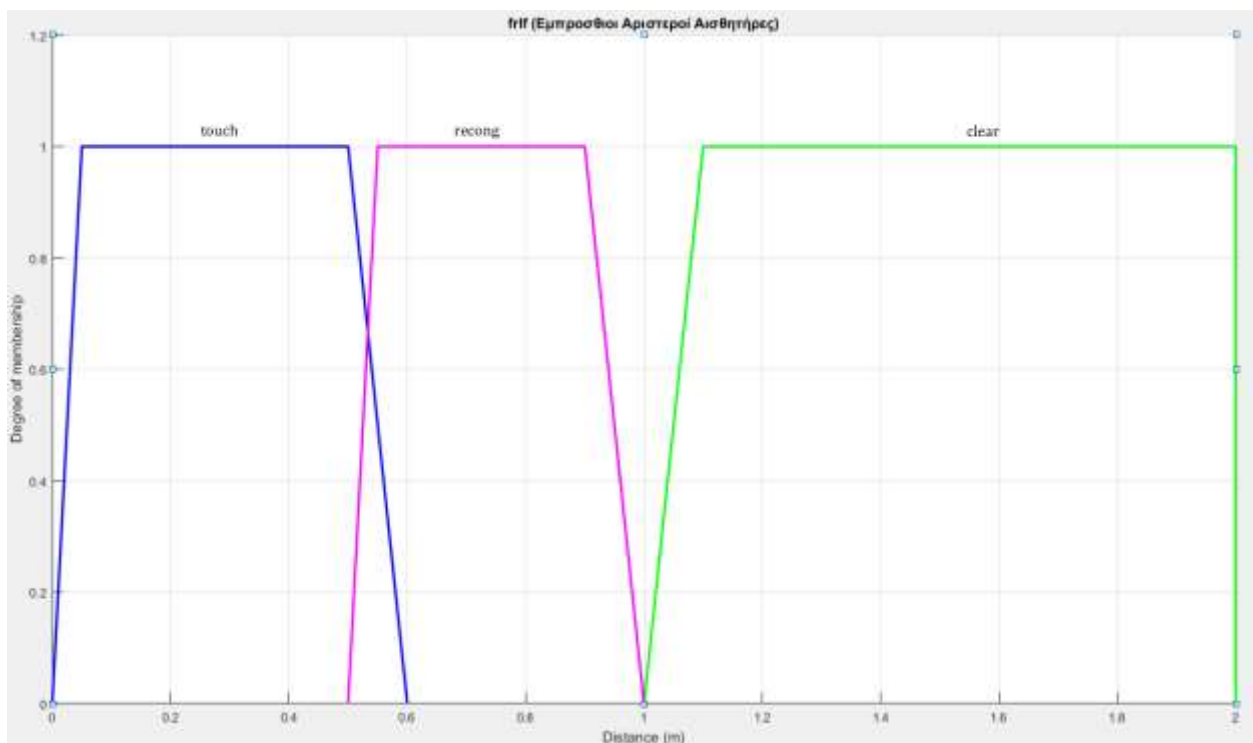
Εικόνα 103: Είσοδος Δεξιοί Αισθητήρες – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφορά



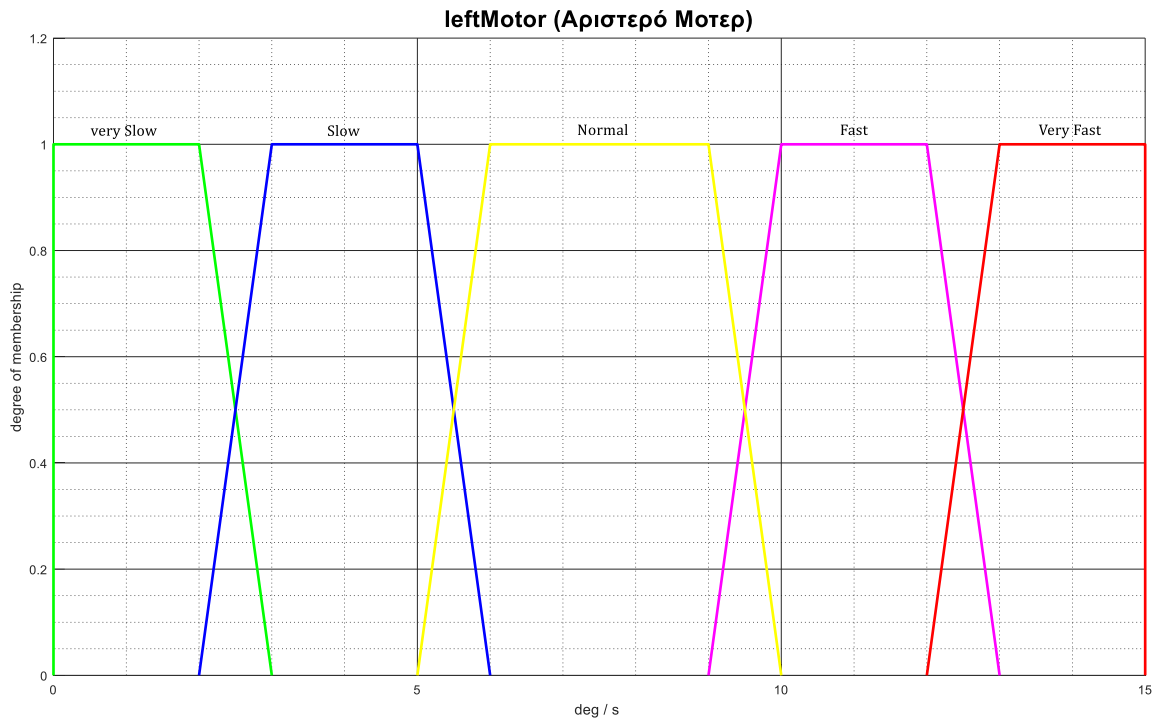
Εικόνα 104: Είσοδος Αριστεροί Αισθητήρες – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς



Εικόνα 105: Είσοδος Εμπρόσθιοι Δεξιοί Αισθητήρες – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς



Εικόνα 106 : Είσοδος Εμπρόσθιοι Αριστεροί Αισθητήρες – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς

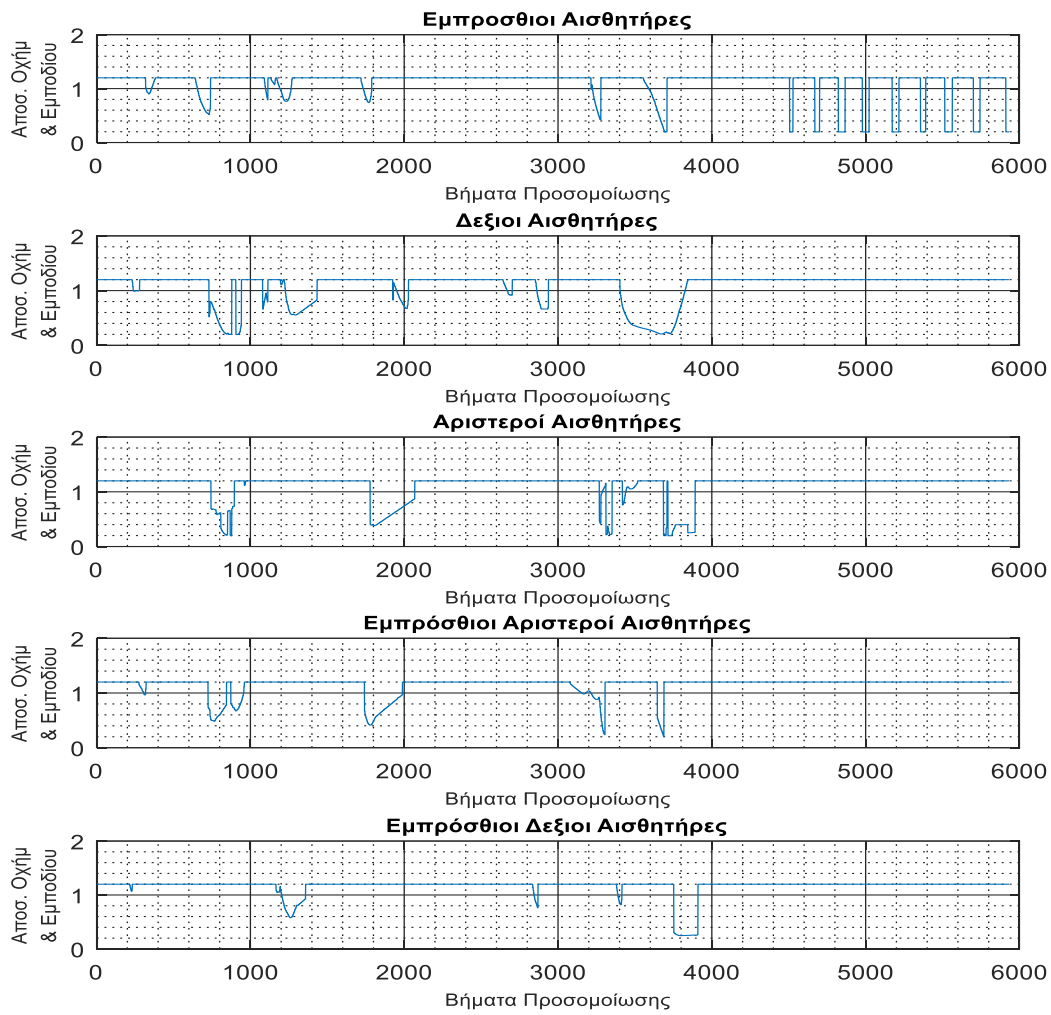


Εικόνα 107: Έξοδος Ταχύτητα Αριστερού μοτέρ – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς

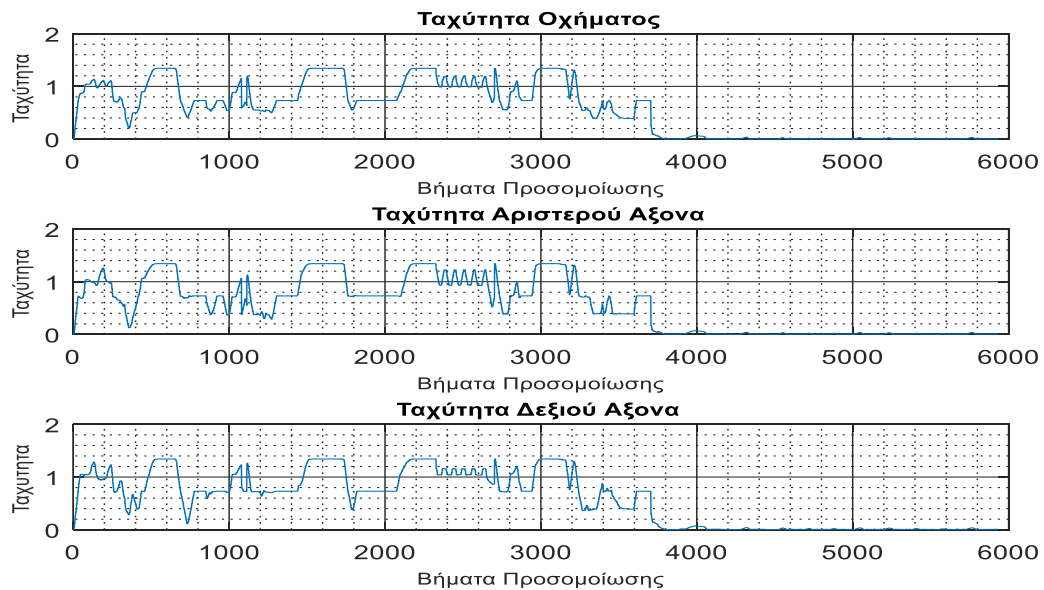


Εικόνα 108: Έξοδος Ταχύτητα Δεξιού μοτέρ – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς

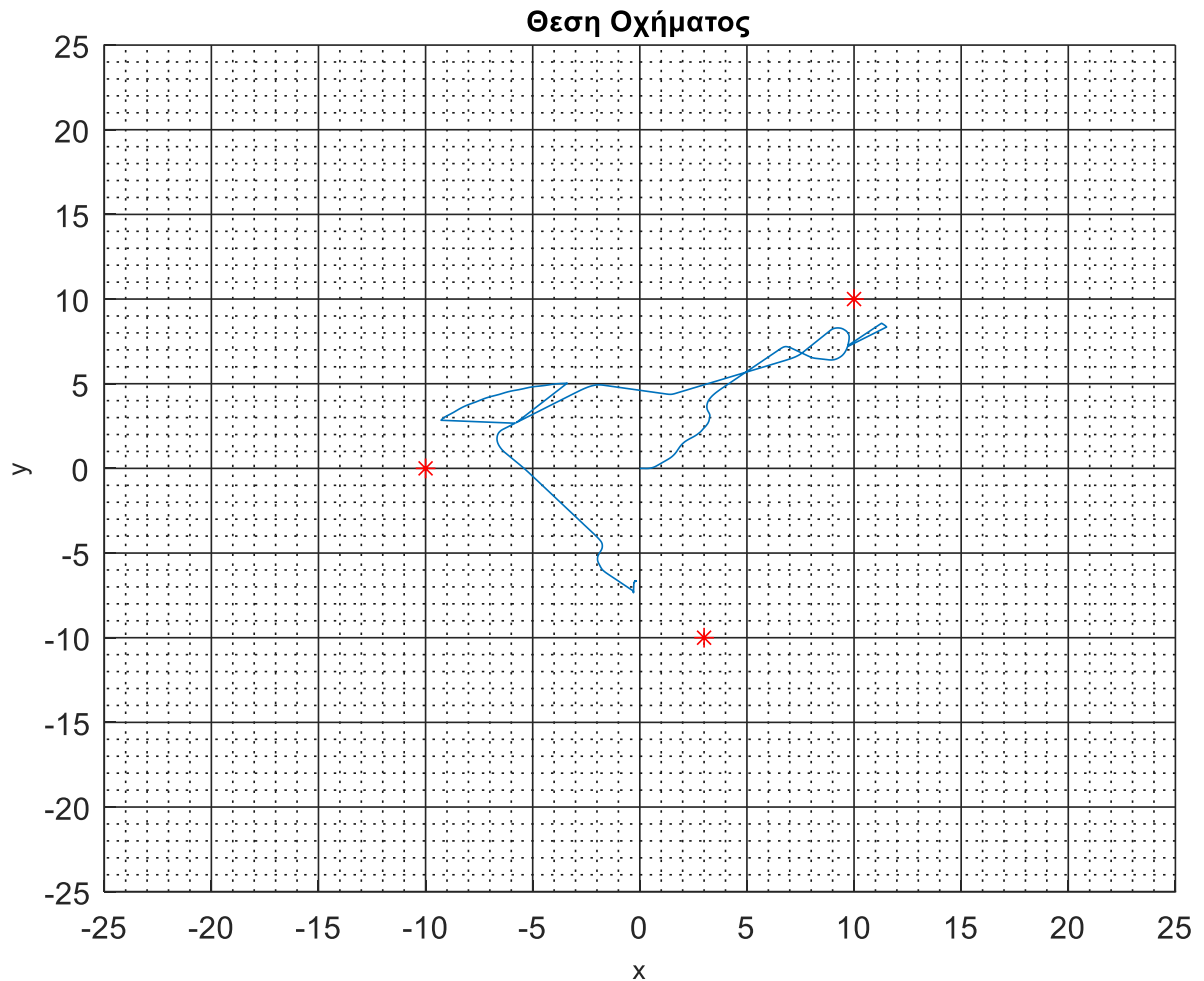
### Αποτελέσματα προσομοίωσης 1<sup>ου</sup> Οχήματος



Εικόνα 109: Δεδομένα αισθητήρων 1ου οχήματος – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς



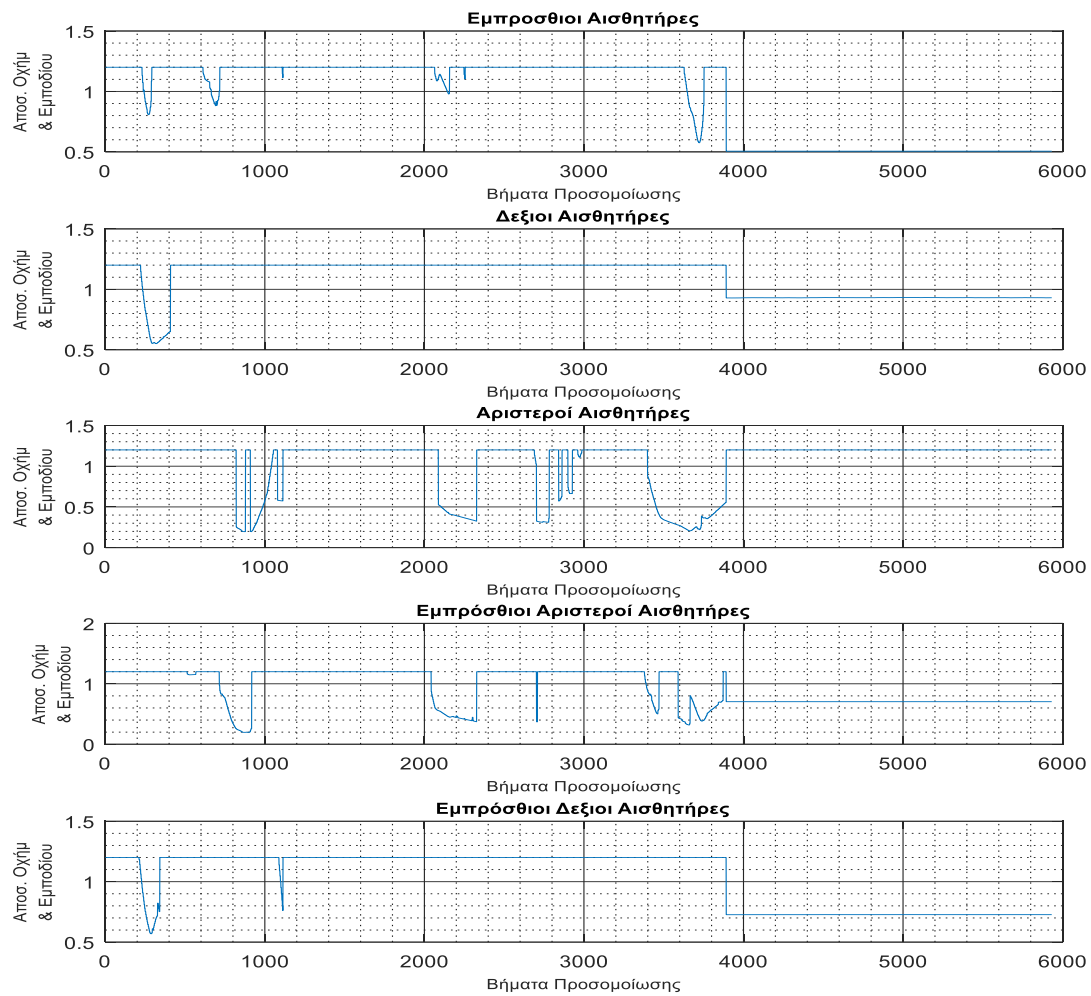
Εικόνα 110: Ταχύτητες 1ου οχήματος – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς



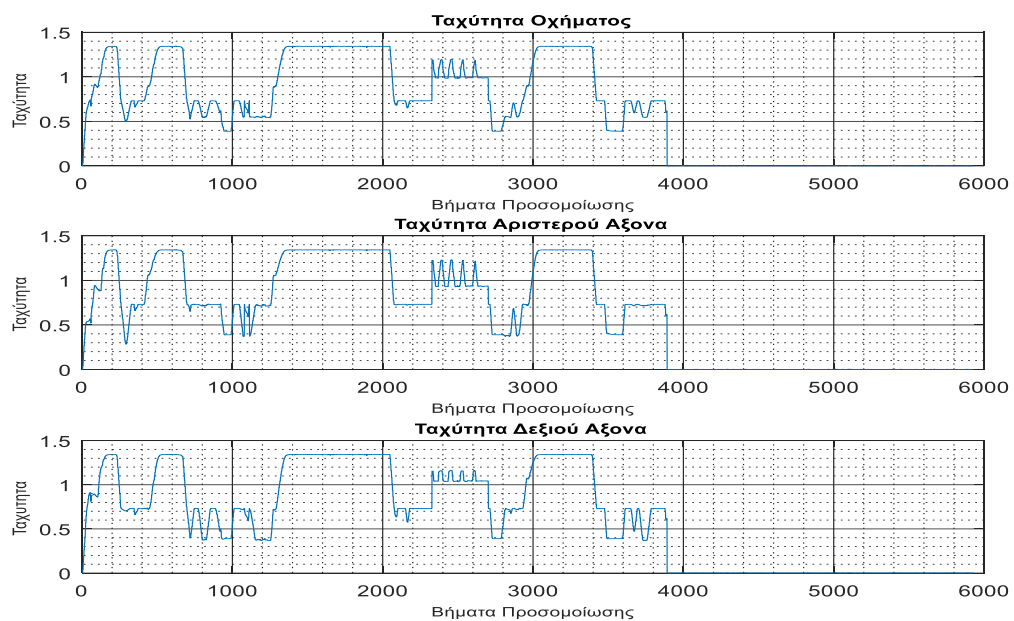
Εικόνα 111: Διαδρομή 1ου οχήματος – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς

Στο διάγραμμα των αισθητήρων του 1<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται όλοι οι αισθητήρες άλλα για μικρότερο χρονικό διάστημα από το 1<sup>ο</sup> όχημα του αρχικού ελεγκτή και αποφεύγει τα εμπόδια. Αντίστοιχα παρατηρούμε και ότι η ταχύτητα του οχήματος διατηρείται σε υψηλότερη τιμή από το 1<sup>ο</sup> όχημα της αρχικής έκδοσης του ελεγκτή. Στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία όμως δεν καταφέρνει να προσεγγίσει το 3<sup>ο</sup> σημείο.

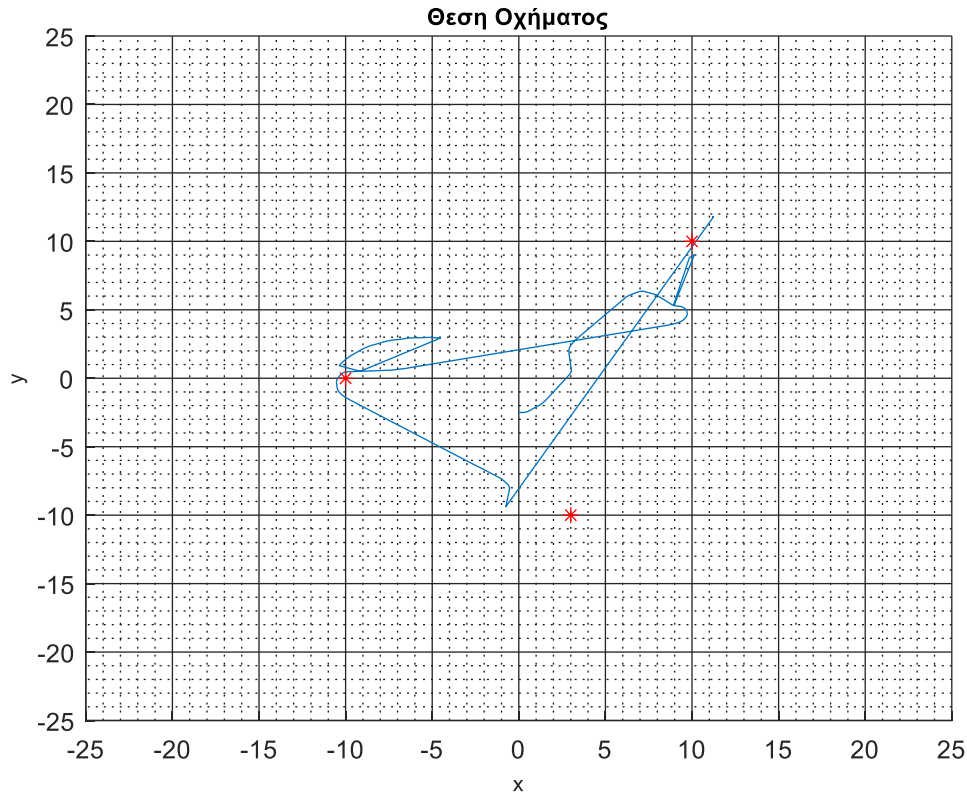
## Αποτελέσματα προσομοίωσης 2<sup>ου</sup> Οχήματος



Εικόνα 112: Δεδομένα αισθητήρων 2ου οχήματος – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς



Εικόνα 113: Ταχύτητες 2ου οχήματος – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς

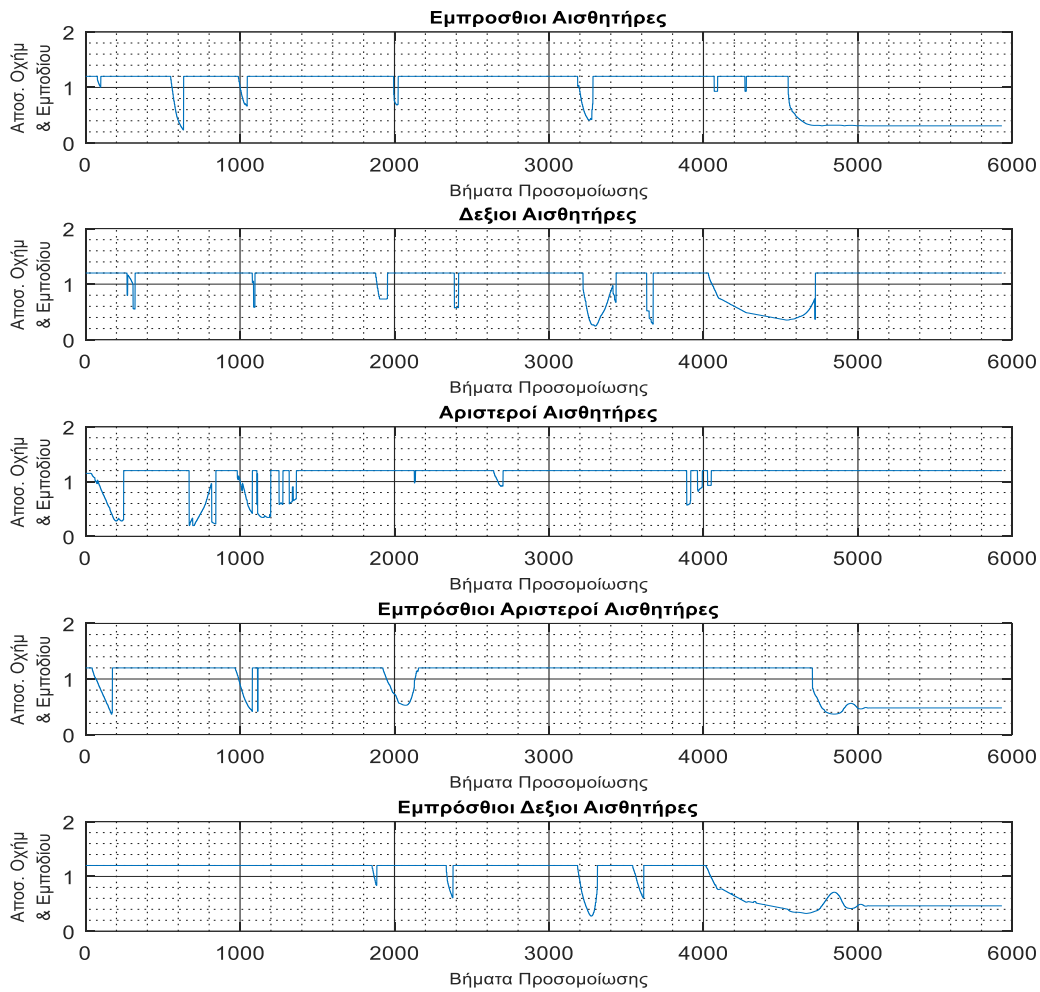


Εικόνα 114 : Διαδρομή 2ου οχήματος – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς

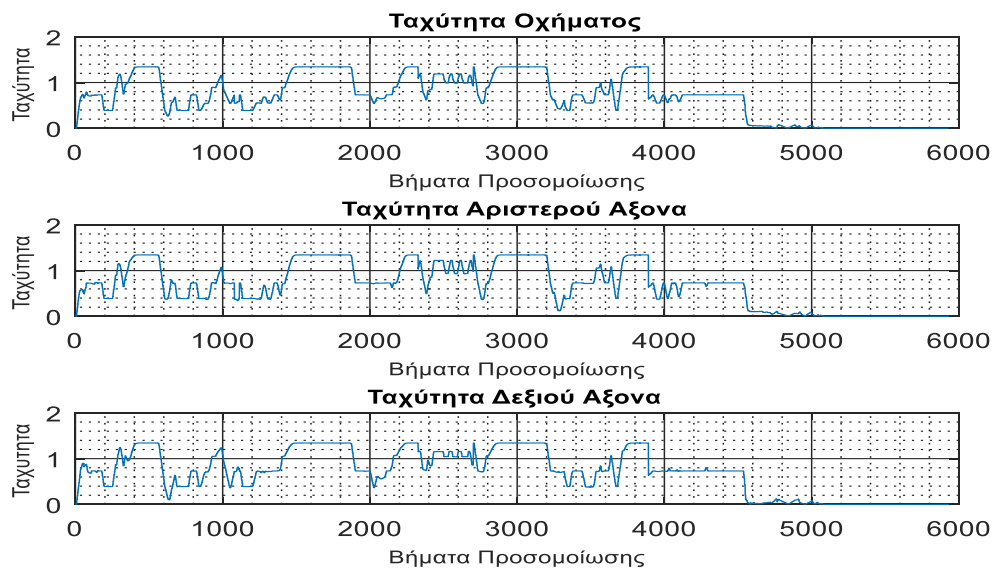
Στο διάγραμμα των αισθητήρων του 2<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται όλοι οι αισθητήρες αλλά για μικρότερο χρονικό διάστημα από το 2<sup>ο</sup> όχημα του αρχικού ελεγκτή και αποφεύγει τα εμπόδια. Αντίστοιχα παρατηρούμε και ότι η ταχύτητα του οχήματος διατηρείται σε υψηλότερη τιμή από το 2<sup>ο</sup> όχημα της αρχικής έκδοσης του ελεγκτή παρότι παρουσιάζει αυξομειώσεις αλλά σε πολύ υψηλές τιμές. Στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία.



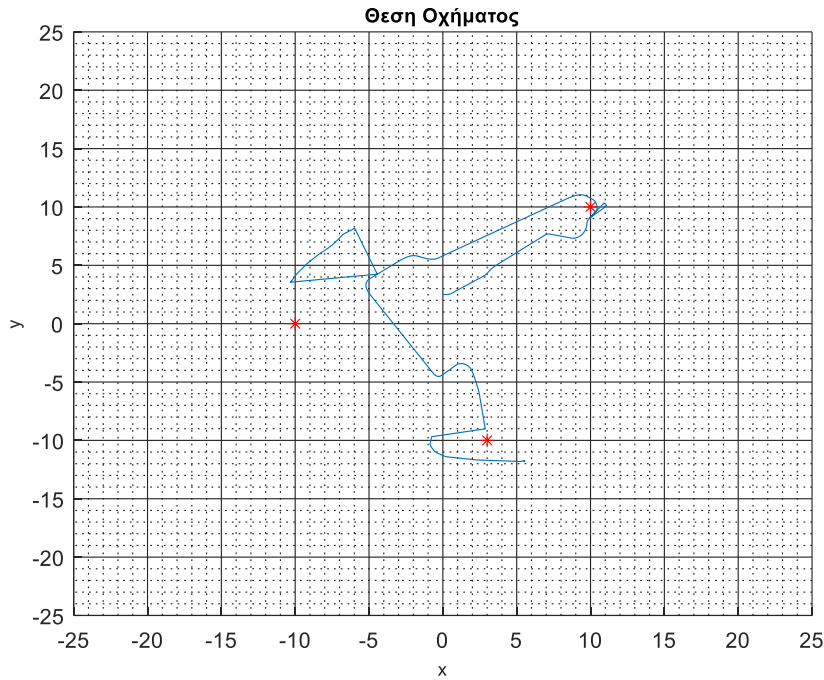
**Αποτελέσματα προσομοίωσης 3<sup>ου</sup> Οχήματος**



Εικόνα 115: Δεδομένα αισθητήρων 3ου οχήματος – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς



Εικόνα 116: Ταχύτητες 3ου οχήματος – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς



Εικόνα 117: Διαδρομή 3ου οχήματος – Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή συνάρτησης καταλληλότητας συμπεριφοράς

Στο διάγραμμα των αισθητήρων του 3ου οχήματος παρατηρούμε ότι ενεργοποιούνται όλοι οι αισθητήρες άλλα για μικρότερο χρονικό διάστημα από το 3ο όχημα του αρχικού ελεγκτή και αποφεύγει τα εμπόδια. Αντίστοιχα παρατηρούμε και ότι η ταχύτητα του οχήματος διατηρείται σε υψηλότερη τιμή από το 3ο όχημα της αρχικής έκδοσης του ελεγκτή. Στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία.

Στον πίνακα 7 παρουσιάζονται συγκεντρωτικά τα αποτελέσματα των προσομοιώσεων οι οποίες έγιναν με χρήση των ελεγκτών που εξήχθησαν από την εξελικτική διαδικασία με χρήση της συνάρτησης καταλληλότητας τύπου συμπεριφοράς. Παρατηρήσουμε ότι στον καλύτερο ελεγκτή το όχημα κινείται πιο μακριά από τα εμπόδια (σταθερά και μη σταθερά) και υπάρχει βελτιστοποίηση στις τιμές του χρόνου ολοκλήρωσης της προσομοίωσης. Επίσης ενώ ο ελεγκτής που με την καλύτερη τιμή συνάρτησης καταλληλότητας στην γενεά με την μέση τιμή είναι σχεδόν ίδιος με τον αρχικό ελεγκτή παρουσιάζει βελτιστοποίηση στον χρόνο που εκτελείται η προσομοίωση

|                 | <i>S</i> | <i>U</i> | <i>Ul</i> | <i>Ur</i> | <i>Th</i> | <i>Total Time</i> |
|-----------------|----------|----------|-----------|-----------|-----------|-------------------|
| <i>Original</i> | 1,1029   | 0.8713   | 0,8660    | 0,8760    | 0         | 123,43            |
| <i>min</i>      | 1.127    | 0.843    | 0.832     | 0.8539    | 0         | 73.042            |
| <i>mean</i>     | 1.096    | 0.734    | 0.726     | 0.743     | 0         | 66.075            |

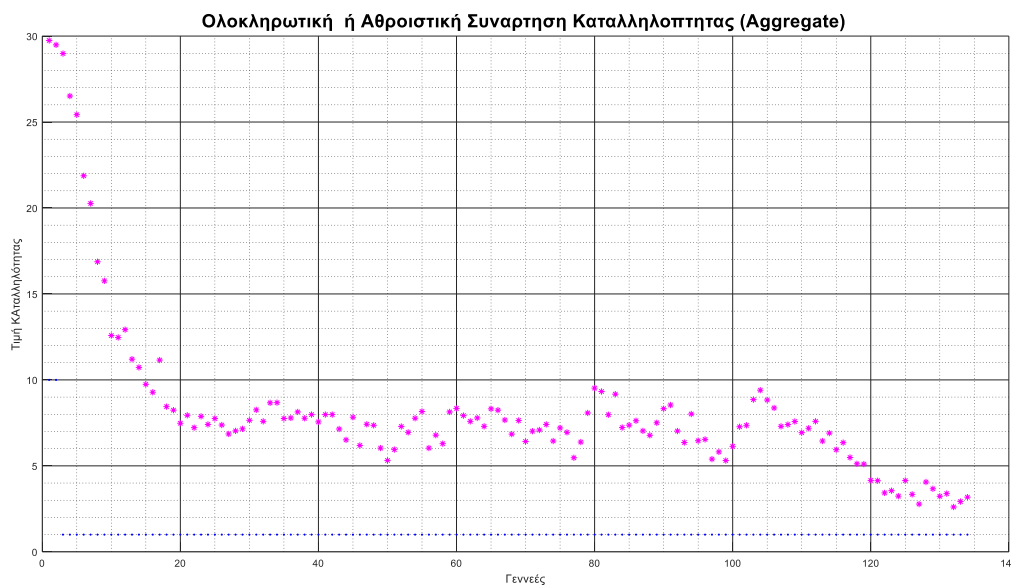
|             |        |          |          |          |  |          |
|-------------|--------|----------|----------|----------|--|----------|
| <i>min</i>  | 2,19 % | -3,25 %  | -3,93 %  | -2,52 %  |  | -40,82 % |
| <i>mean</i> | -0,63% | -18,71 % | -19,28 % | -17,90 % |  | -86,81%  |

Πίνακας 7: Σύγκριση αποτελεσμάτων εξέλιξης με χρήση συνάρτησης καταλληλότητας συμπεριφοράς

## 4.5 Αποτελέσματα Εξελικτικής Διαδικασίας : Αθροιστική ή ολοκληρωτική συνάρτηση καταλληλότητας (Aggregate)

Στην συγκεκριμένη κατηγορία συναρτήσεων καταλληλότητας μετρούμε μόνο το αποτέλεσμα της διαδικασίας, χωρίς να ελέγχουμε πως αυτό έχει επιτευχθεί. Έτσι στην συγκεκριμένη περίπτωση ελέγχουμε και βαθμολογούμε αν τα οχήματα έχουν προσεγγίσει τα σημεία ή όχι.

Άρα όταν τα οχήματα έχουν προσεγγίσει και τα 3 σημεία λαμβάνουν βαθμολογία 1, ενώ αν δεν προσεγγίσουν κάποιο σημείο παίρνουν βαθμολογία 30. Προσέγγιση 2 σημείων βαθμολογείται με 10, ενώ προσέγγιση ενός σημείου 20. Ο στόχος είναι το μικρότερο να είναι το καλύτερο.



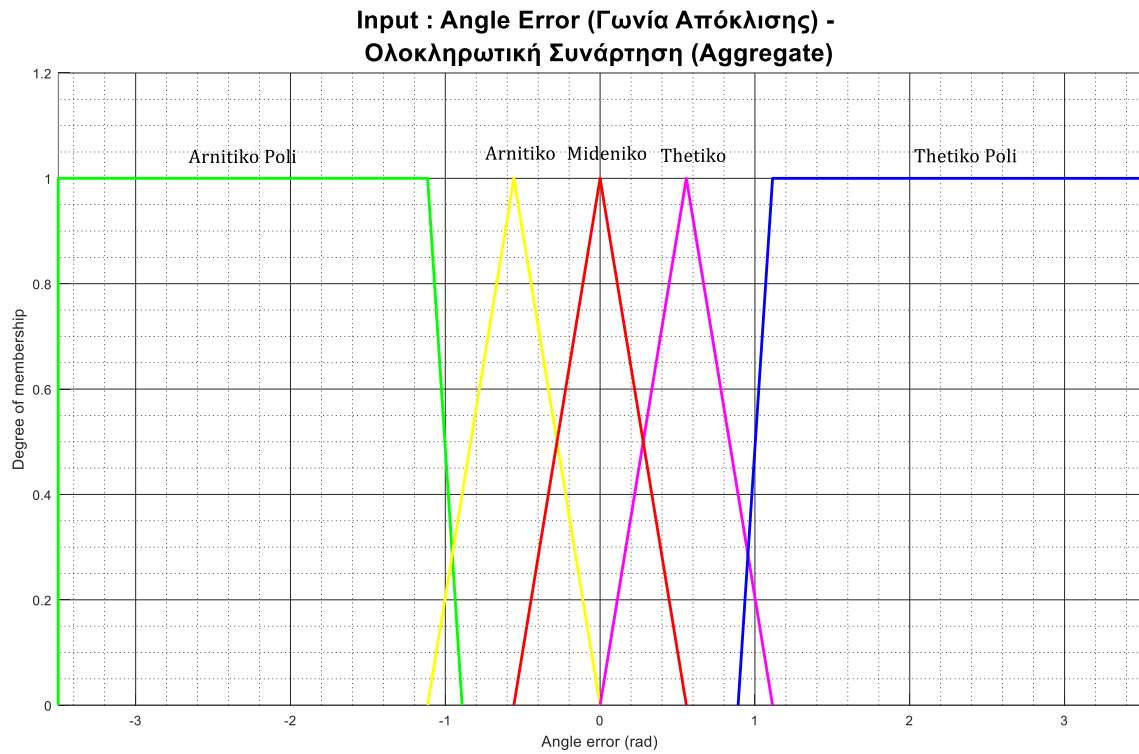
Εικόνα 118 : Εξέλιξη βελτιστοποίηση ασαφούς ελεγκτή με ολοκληρωτική συνάρτηση καταλληλότητας

Παρατηρούμε ότι υπάρχει βελτιστοποίηση της μέση τιμής της συνάρτησης ώστε ο μέσος όρος όλων των συναρτήσεων ανά γενεά να προσεγγίσει την τιμή του καλύτερου ελεγκτή ανά γενεά, Ο αριθμός των γενεών που έγινε η εξελικτική διαδικασία ήταν ικανοποιητικός. Απλά δεν υπάρχει μια κανονική συνέχεια λόγω της αδυναμίας του συστήματος που έγινε η μελέτη (μικρή μνήμη). Με αποτέλεσμα σε κάθε διακοπή να παρουσιάζεται μια υστέρηση και χάσιμο μερικών γενεών πριν να αρχίσει να βελτιώνεται ο ελεγκτής. Το φαινόμενο μπορεί να εντοπιστεί στην 30η γενεά, 80η γενεά και την 100<sup>η</sup>.

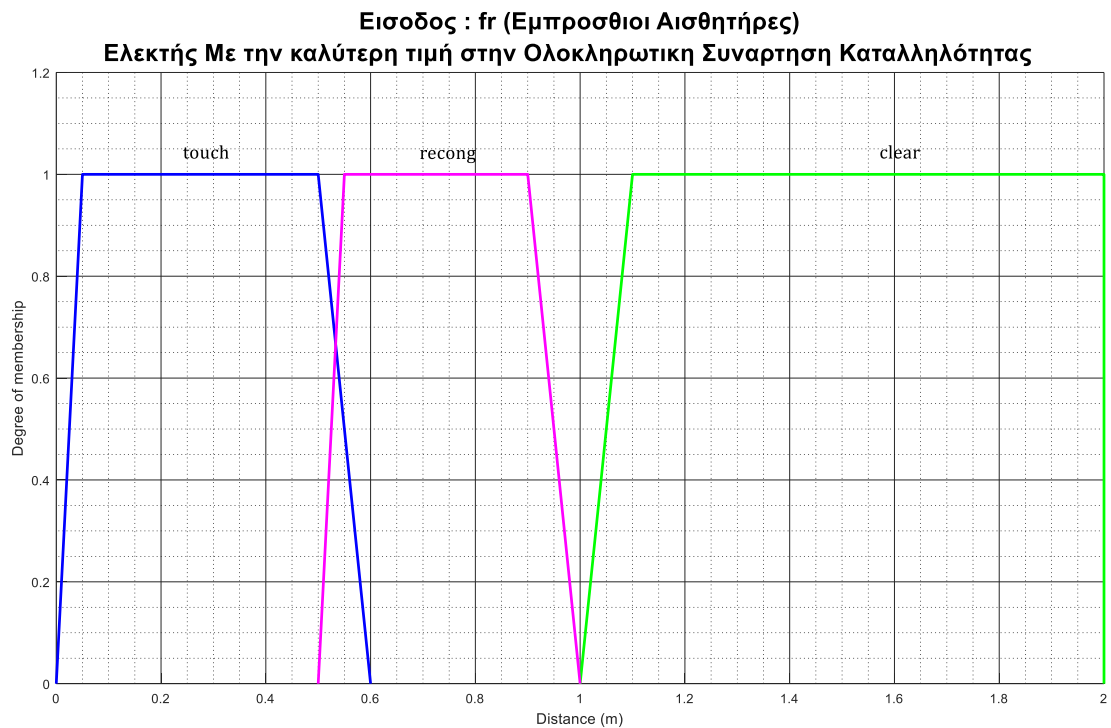
Παρόλα αυτά εξάγουμε δυο ελεγκτές:

1. Τον ελεγκτή με την καλύτερη τιμή της συνάρτησης καταλληλότητας.
2. Τον καλύτερο ελεγκτή στην γενεά με τον καλύτερο μέσο όρο της τιμής της συνάρτησης καταλληλότητας.

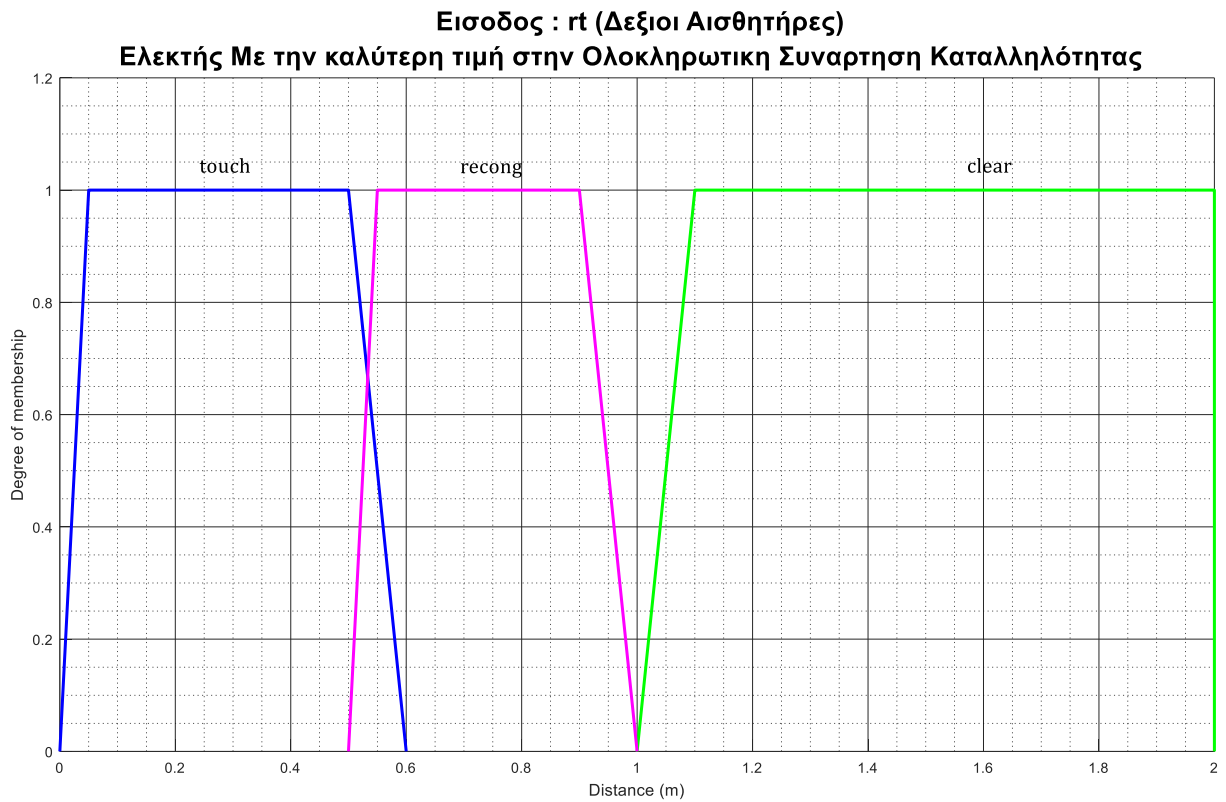
Ελεγκτή : με την καλύτερη τιμή στην ολοκληρωτική συνάρτηση καταλληλότητας



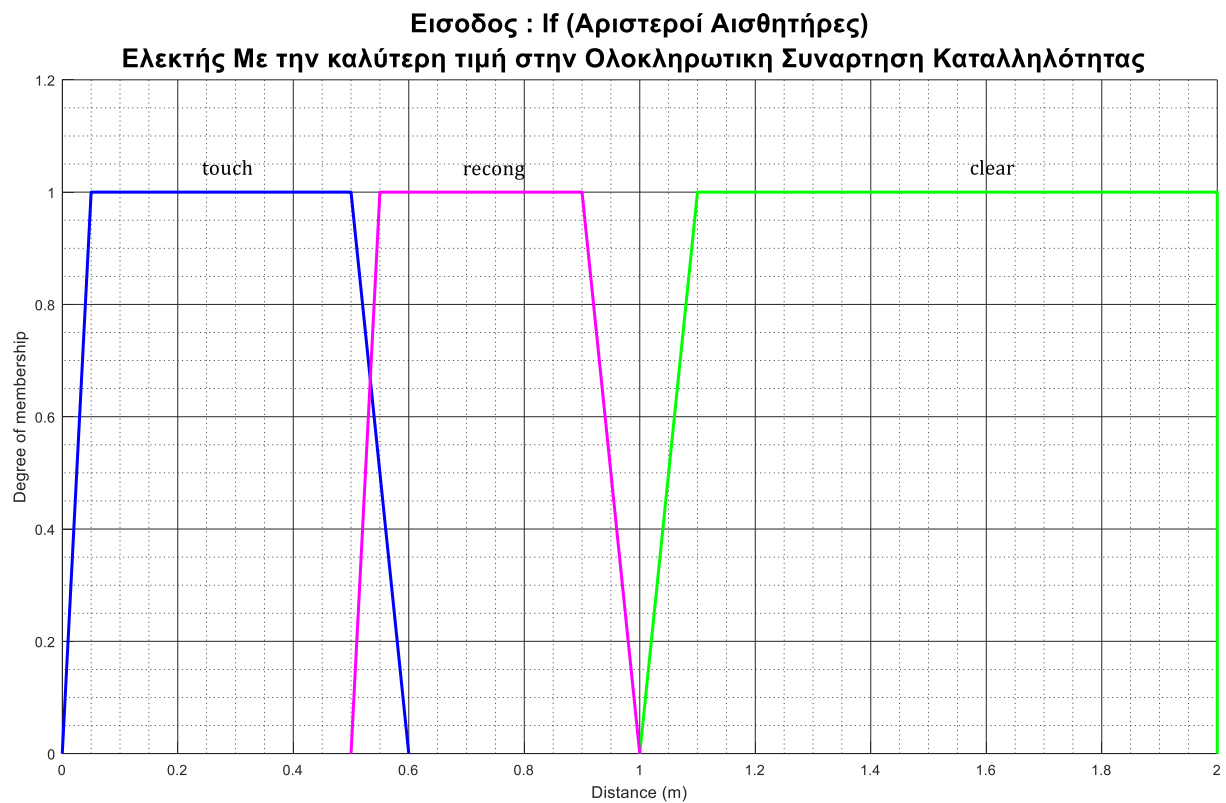
Εικόνα 119 : Είσοδος Γωνία Απόκλισης – Ελεγκτής Με την καλύτερη τιμή στην Ολοκληρωτική συνάρτησης καταλληλότητας



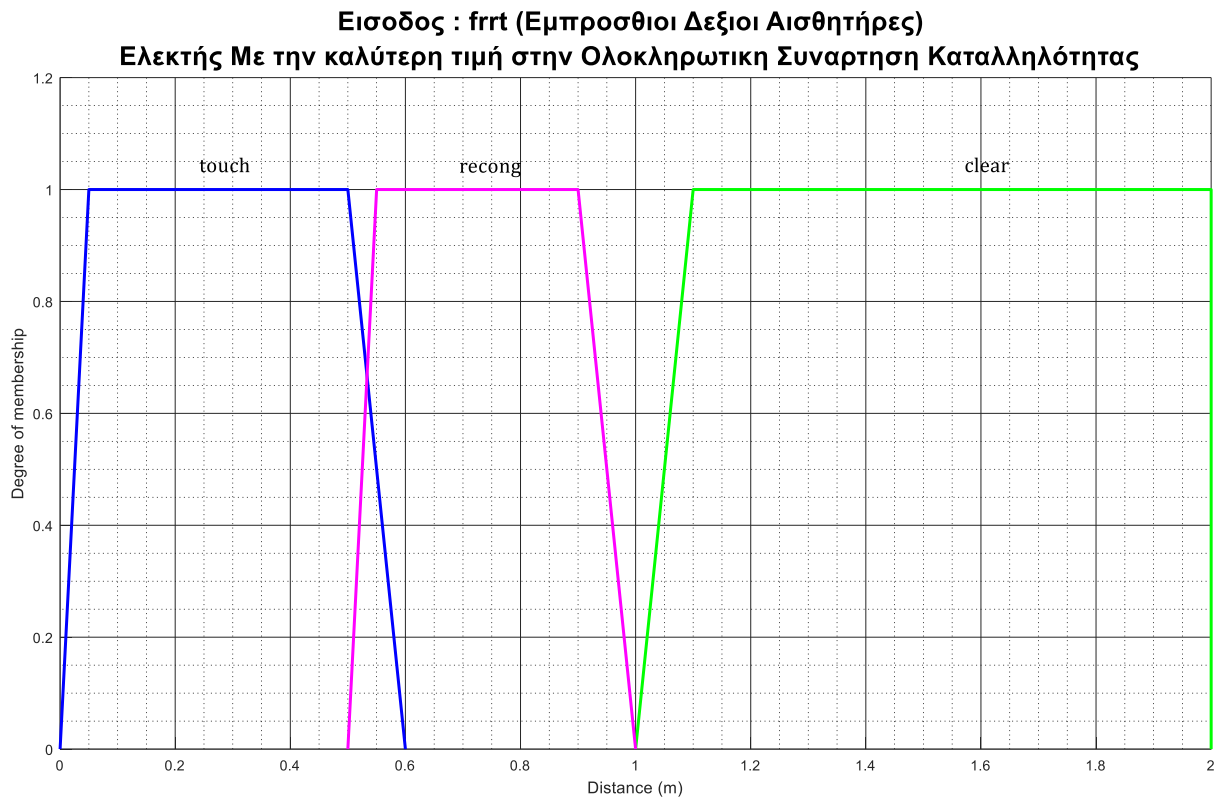
Εικόνα 120 : Είσοδος Εμπρόσθοι Αισθητήρες –Ελεγκτής Με την καλύτερη τιμή στην Ολοκληρωτική συνάρτηση καταλληλότητας



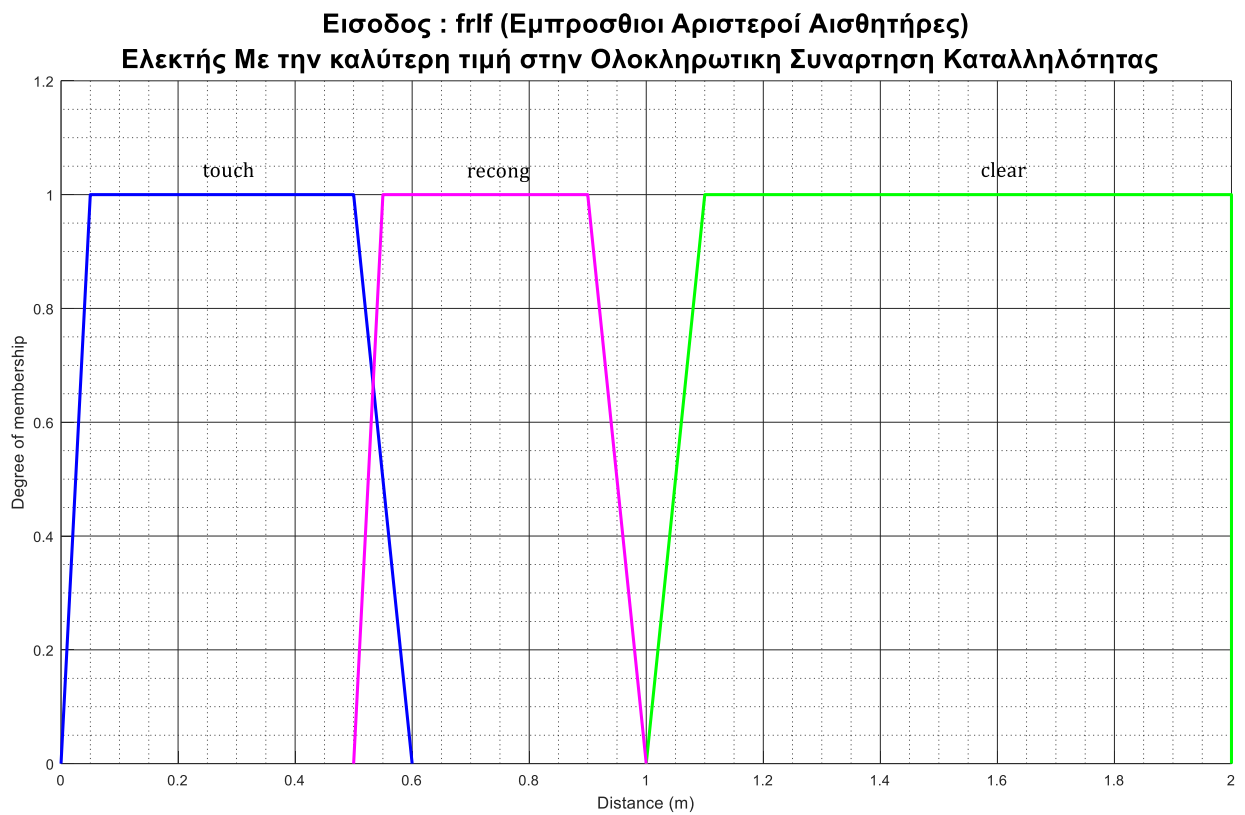
Εικόνα 121 : Είσοδος Δεξιοί Αισθητήρες –Ελεγκτής Με την καλύτερη τιμή στην Ολοκληρωτική συνάρτηση καταλληλότητας



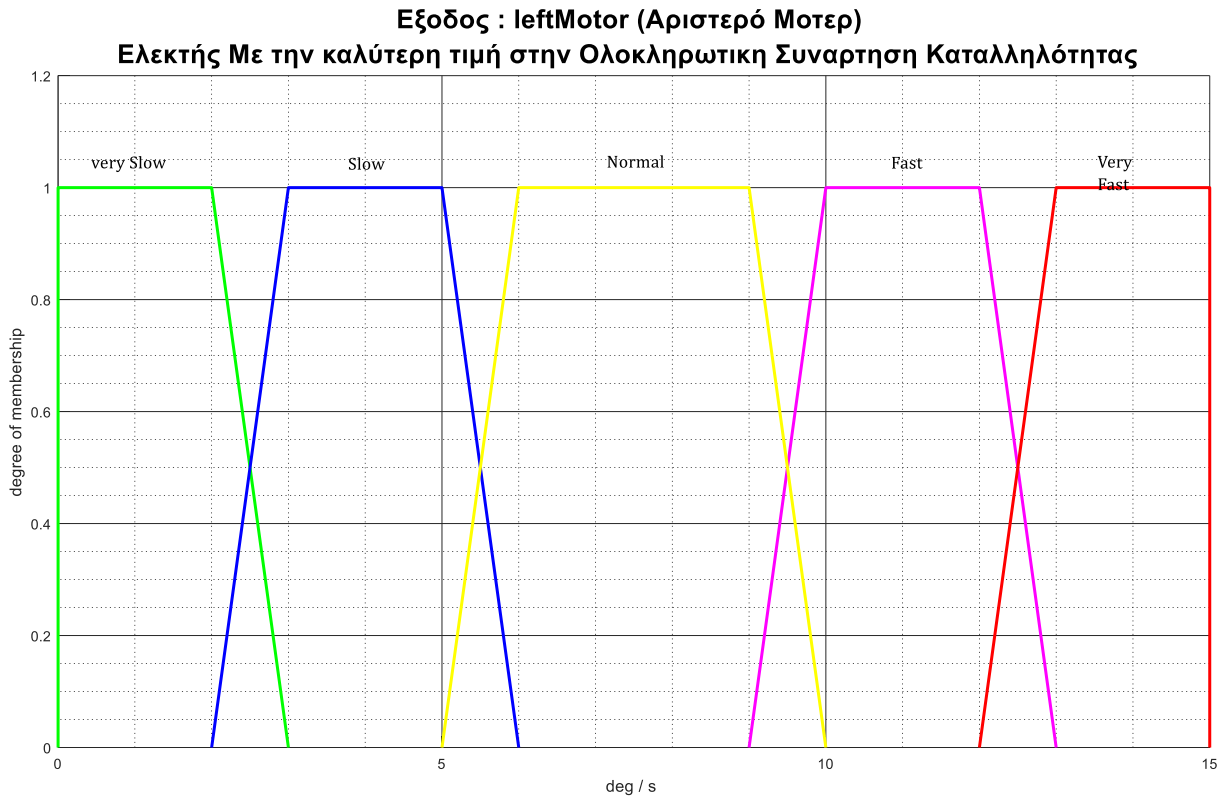
Εικόνα 122 : Είσοδος Αριστεροί Αισθητήρες – Ελεγκτής Με την καλύτερη τιμή στην Ολοκληρωτική συνάρτηση καταλληλότητας



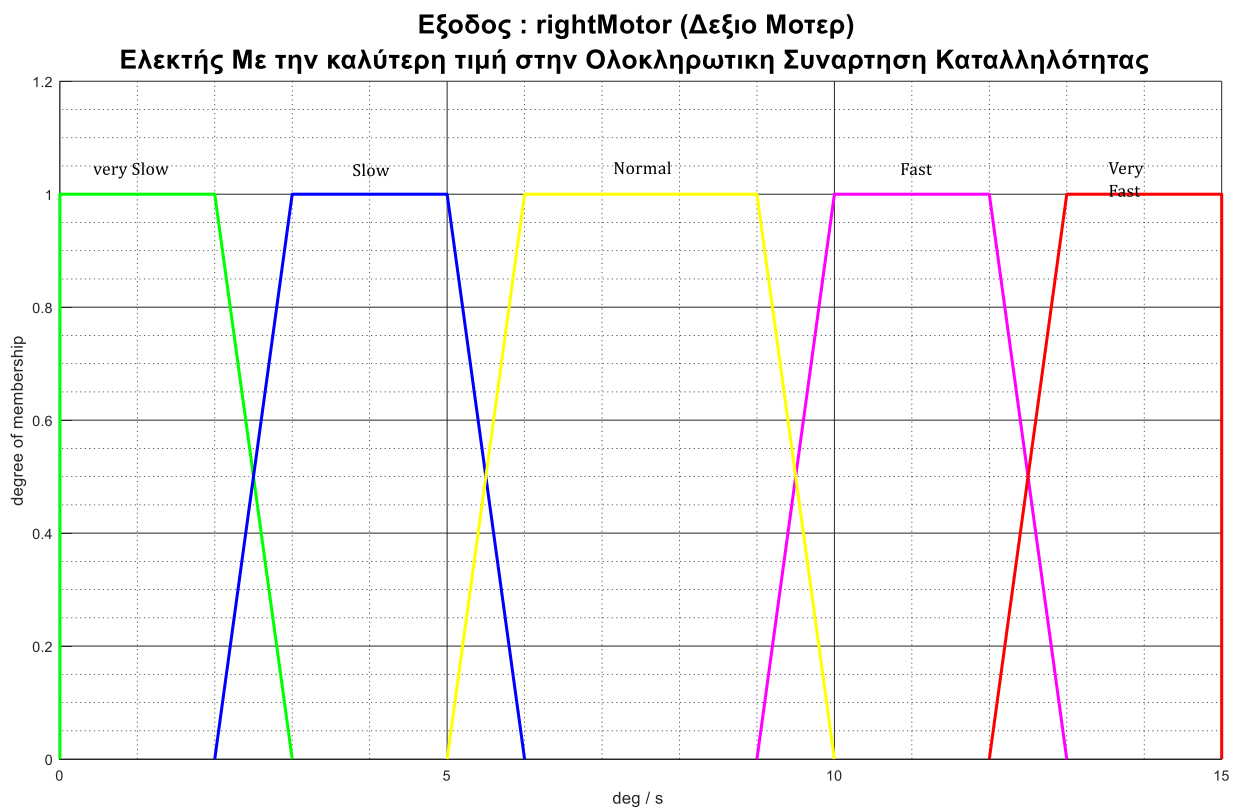
Εικόνα 123 :Είσοδος Εμπρόσθιοι Δεξιοί Αισθητήρες -Ελεγκτής Με την καλύτερη τιμή στην Ολοκληρωτική συνάρτηση καταλληλότητας



Εικόνα 124 : Είσοδος Εμπρόσθιοι Αριστεροί Αισθητήρες -Ελεγκτής Με την καλύτερη τιμή στην Ολοκληρωτική συνάρτηση καταλληλότητας

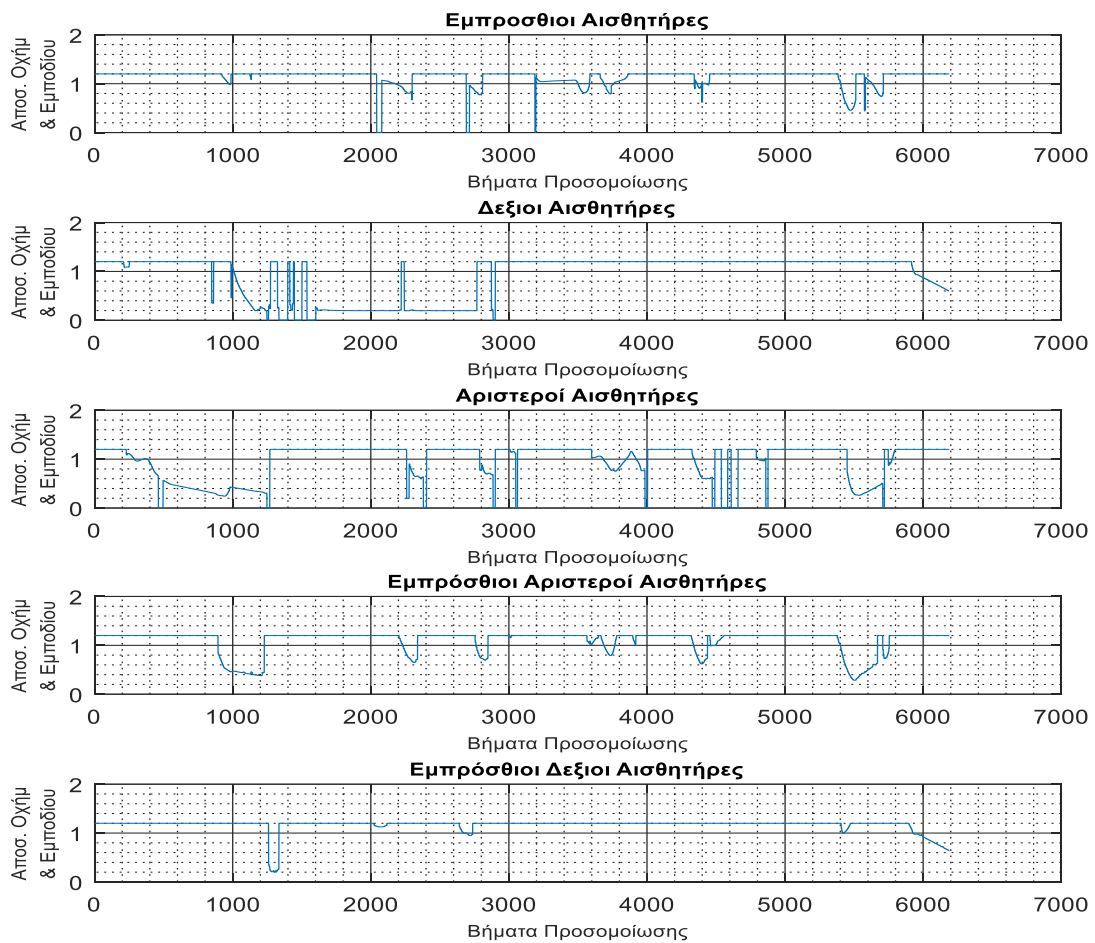


Εικόνα 125 : Έξοδος ταχύτητες αριστερού μοτέρ –Ελεγκτής Με την καλύτερη τιμή στην Ολοκληρωτική συνάρτησης καταλληλότητας

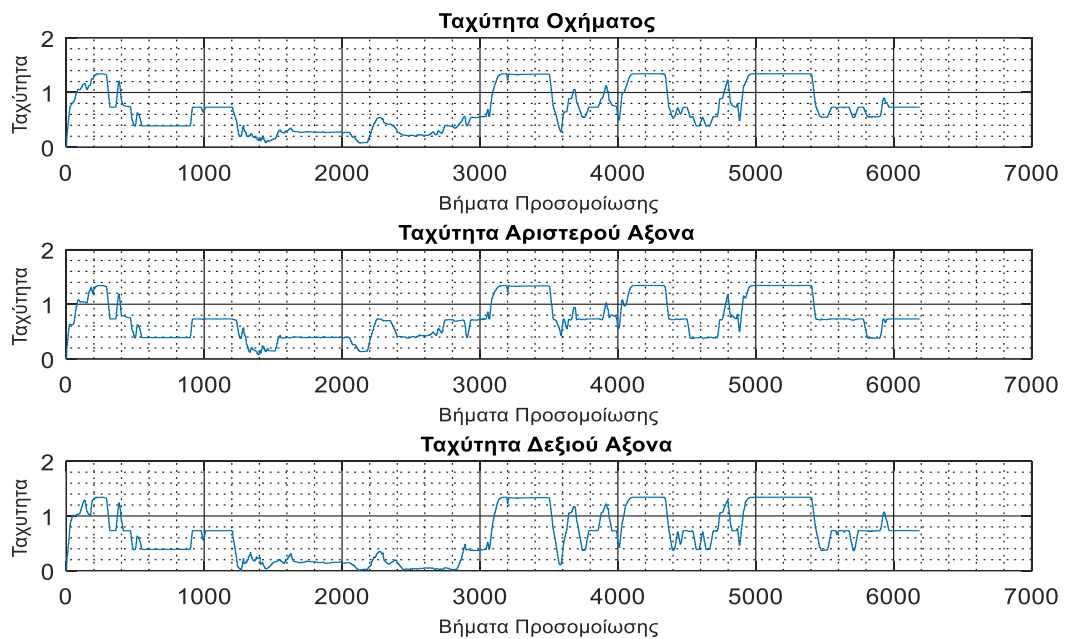


Εικόνα 126 : Έξοδος Ταχύτητες δεξιού μοτέρ –Ελεγκτής Με την καλύτερη τιμή στην Ολοκληρωτική συνάρτηση καταλληλότητας

## Αποτελέσματα προσομοίωσης 1<sup>ου</sup> Οχήματος

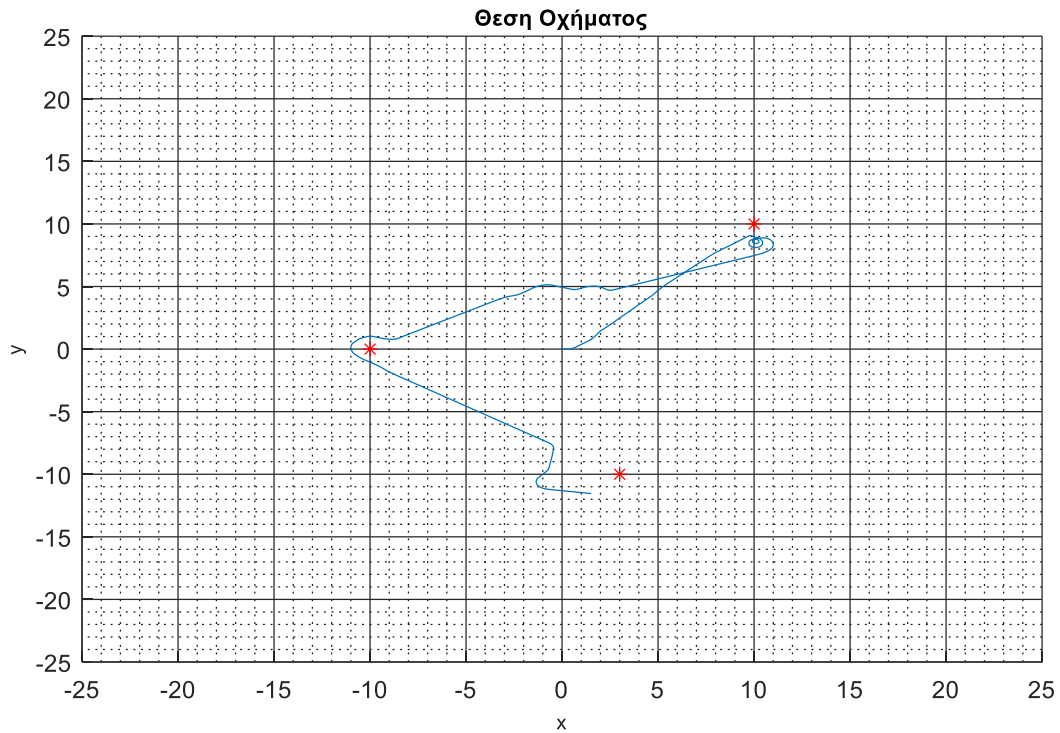


Εικόνα 127 : Δεδομένα Αισθητήρων 1ου Οχήματος – Ελεγκτής με την καλύτερη τιμή Ολοκληρωτική Συνάρτηση Καταλληλότητας



Εικόνα 128: Δεδομένα Ταχυτήτων 1ου Οχήματος – Ελεγκτής με την καλύτερη τιμή Ολοκληρωτική Συνάρτηση Καταλληλότητας

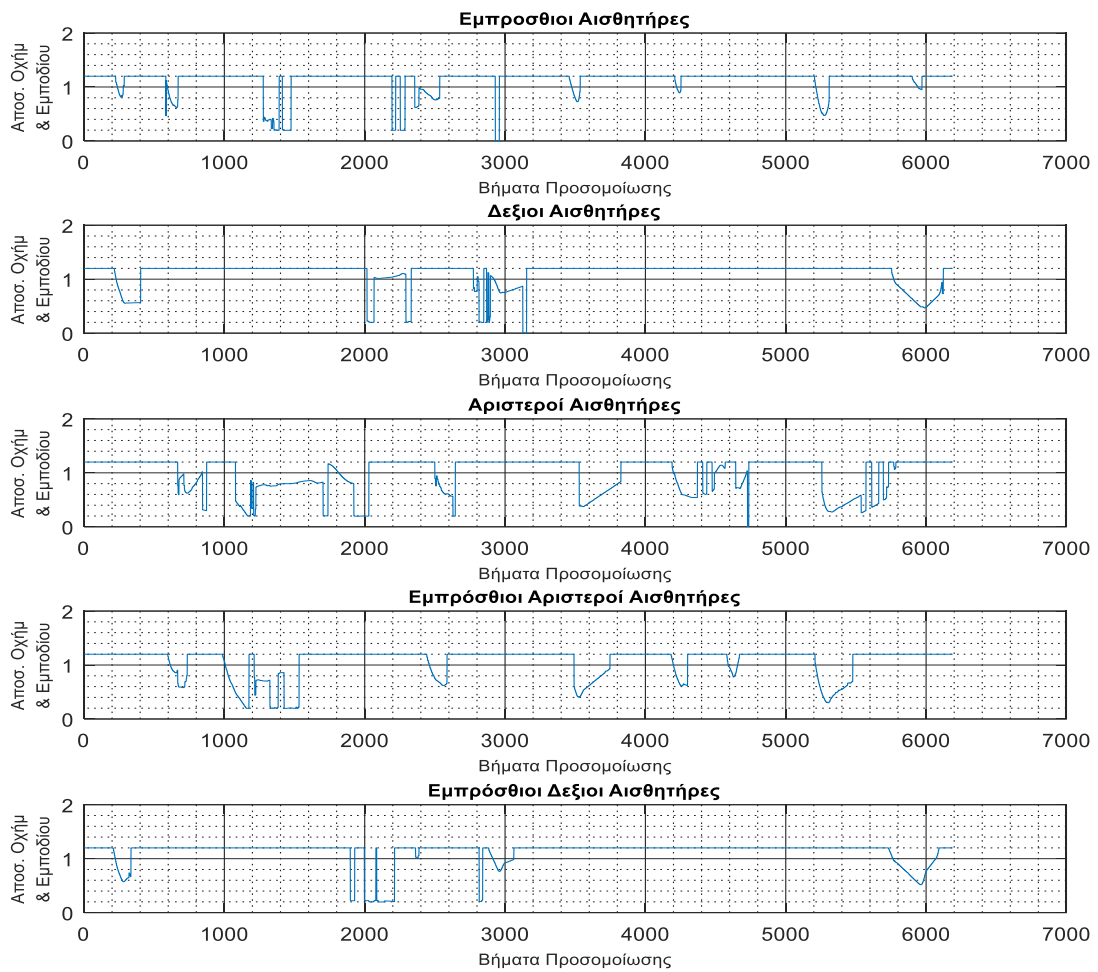




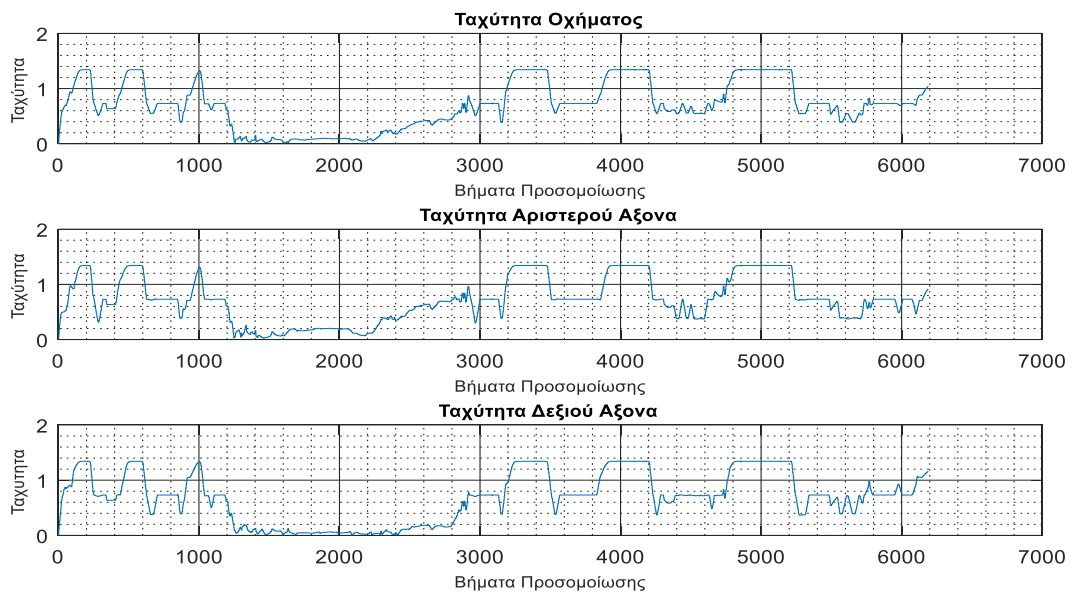
Εικόνα 129 : Διαδρομή 1ου Οχήματος – Ελεγκτής με την καλύτερη τιμή Ολοκληρωτική Συνάρτηση καταλληλότητας

Στο διάγραμμα των αισθητήρων του 1<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Η ταχύτητα του έχει την αναμενόμενη αυξομείωση συνέπεια της ενεργοποιήσεως των αισθητήρων. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει. Γενικά δεν έχει μεγάλες διαφοροποιήσεις από την συμπεριφορά του 1<sup>ου</sup> οχήματος του αρχικού ελεγκτή.

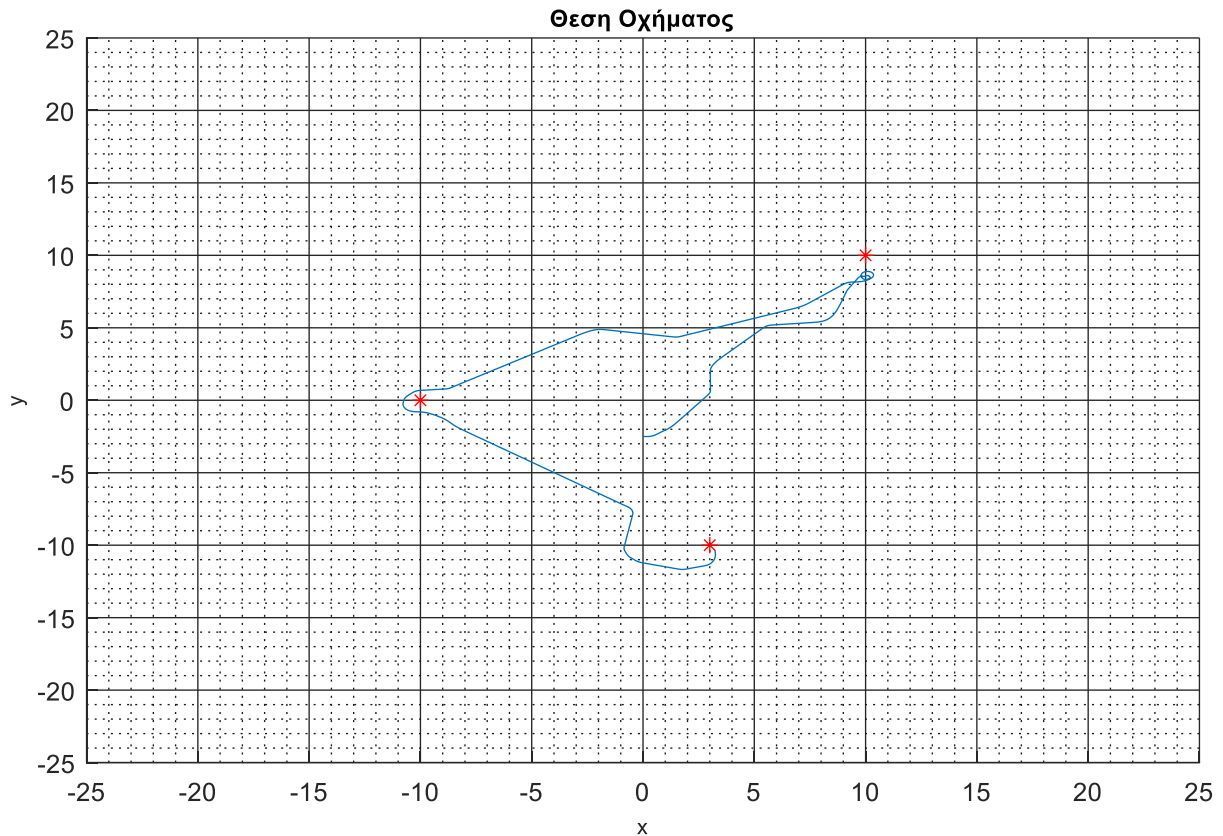
## Αποτελέσματα προσομοίωσης 2<sup>ου</sup> Οχήματος



Εικόνα 130 : Δεδομένα Αισθητήρων 2ου Οχήματος – Ελεγκτής με την καλύτερη τιμή Ολοκληρωτική Συνάρτηση Καταλληλότητας



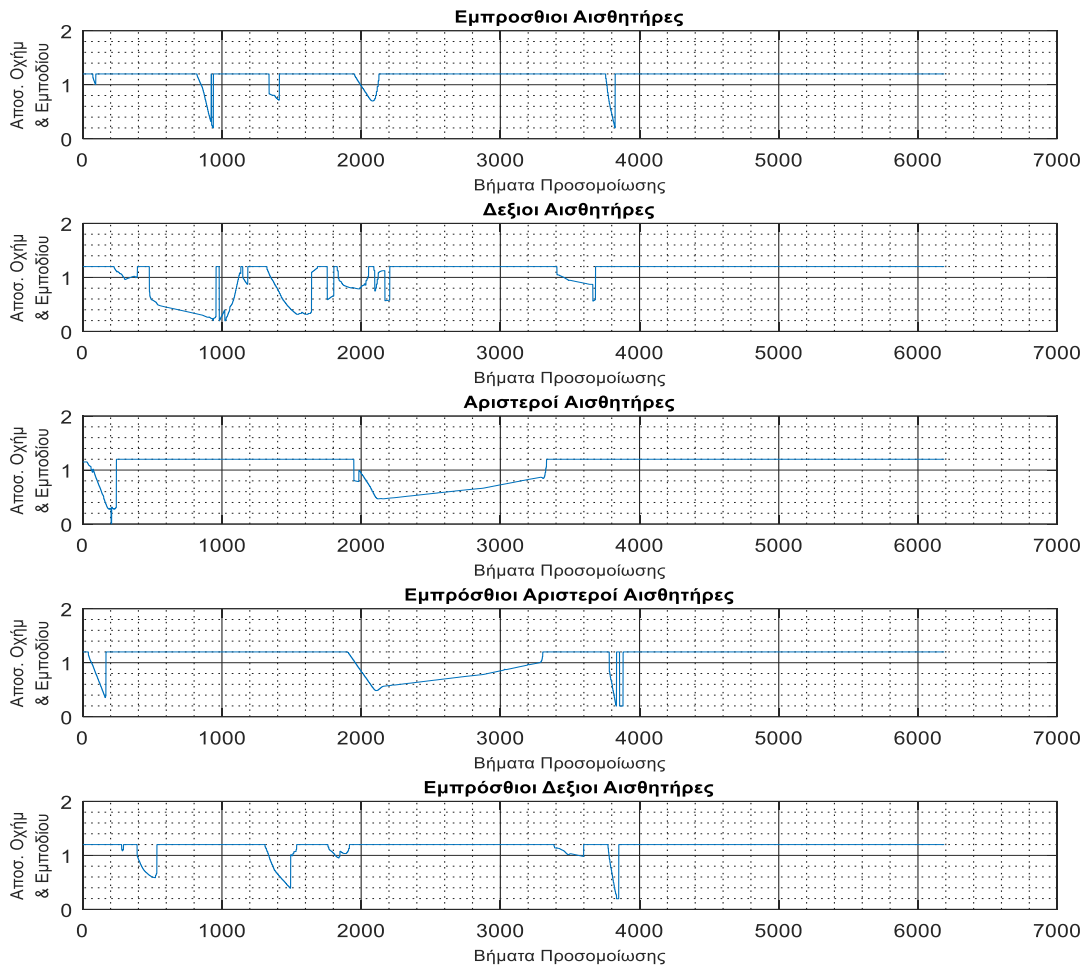
Εικόνα 131 : Ταχύτητες 2ου Οχήματος –λεγκτής με την καλύτερη τιμή Ολοκληρωτική Συνάρτηση καταλληλότητας



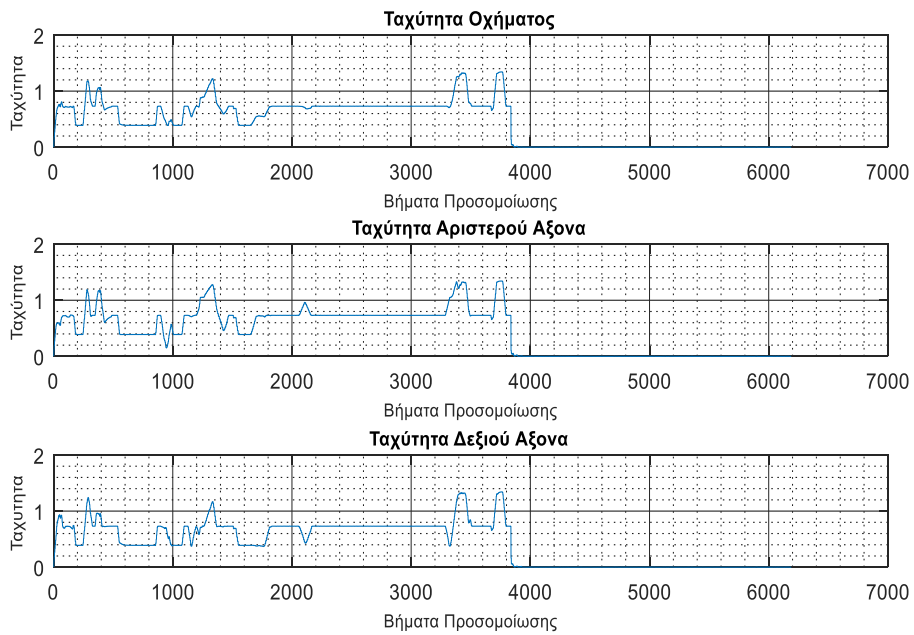
Εικόνα 132 : Διαδρομή 2ου Οχήματος – Ελεγκτής με την καλύτερη τιμή Ολοκληρωτική Συνάρτησης

Στο διάγραμμα των αισθητήρων του 2<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Παρατηρούμε ότι το συγκεκριμένο όχημα παρουσιάζει μεγαλύτερη δυσκολία να αποφύγει τα εμπόδια. Η ταχύτητα του έχει την αναμενόμενη αυξομείωση συνέπεια της ενεργοποίησης των αισθητήρων και λόγω της δυσκολίας που αναφέραμε προηγουμένως εμφανίζει και σχεδόν μηδενισμό αυτής. Τα διαγράμματα των αξόνων δεν διαφέρουν οπότε και το όχημα δεν στρίβει πολύ. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει. Υπάρχει μια μικρή διαφοροποίηση ως προς τις ταχύτητες σε σχέση με το 2<sup>ο</sup> όχημα του αρχικού ελεγκτή.

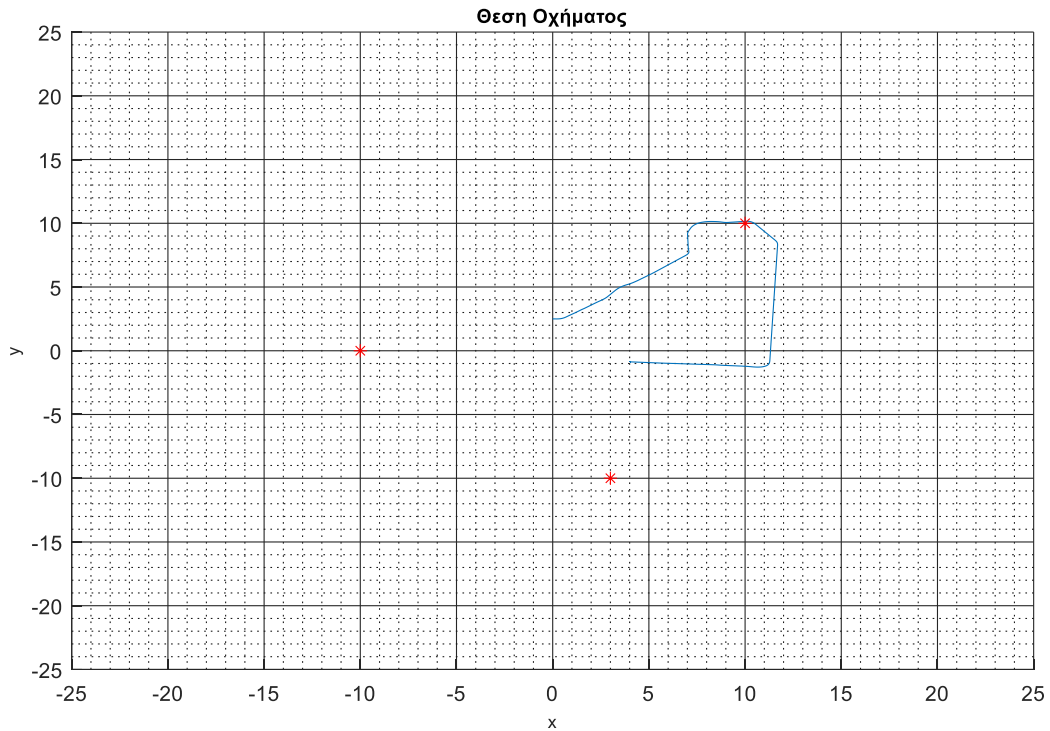
### Αποτελέσματα προσομοίωσης 3<sup>ου</sup> Οχήματος



Εικόνα 133: Δεδομένα Αισθητήρων 3ου Οχήματος –Ελεγκτής με την καλύτερη τιμή Ολοκληρωτική Συνάρτησης



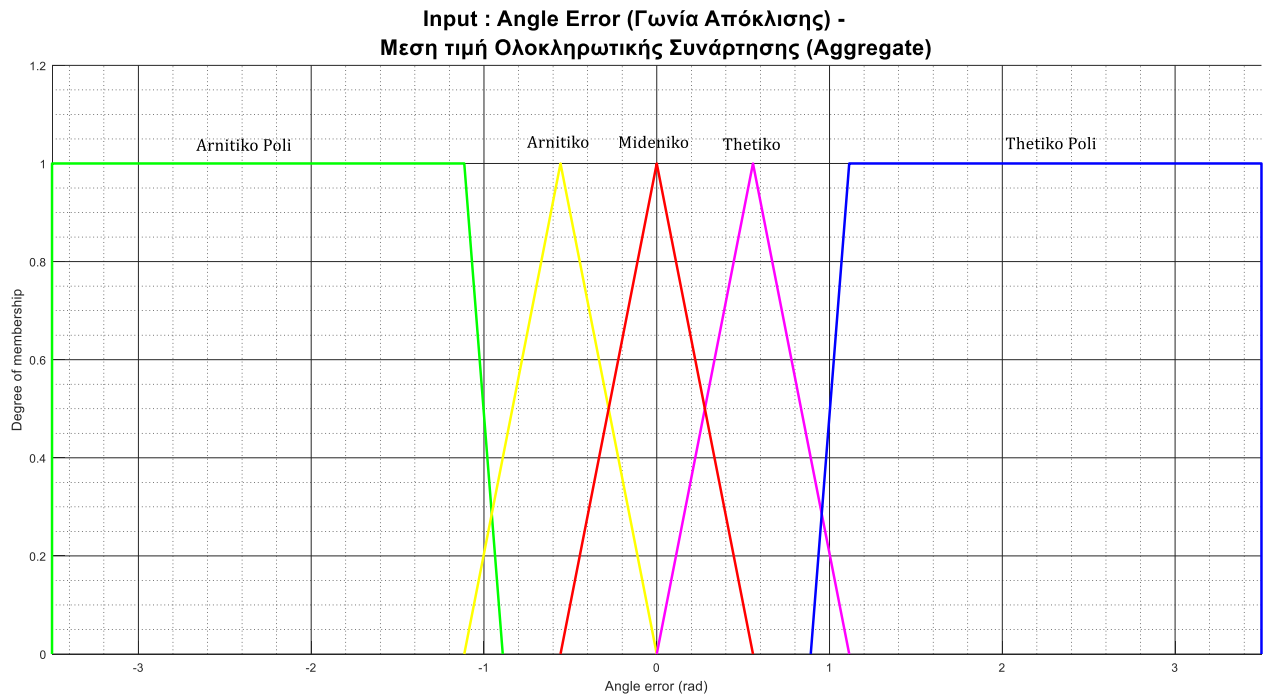
Εικόνα 134 : Ταχύτητες 3ου Οχήματος – Ελεγκτής με την καλύτερη τιμή Ολοκληρωτική Συνάρτηση Καταλληλότητας



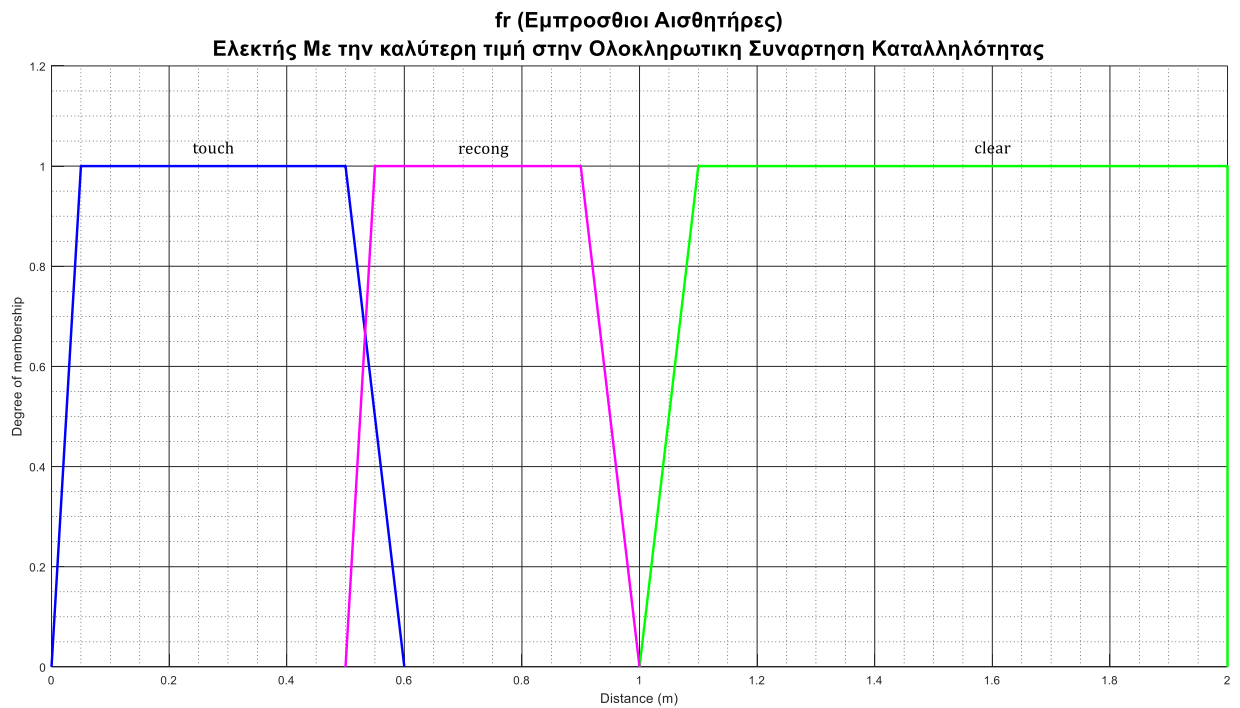
Εικόνα 135 : Διαδρομή 3ου Οχήματος – Ελεγκτής με την καλύτερη τιμή Ολοκληρωτική Συνάρτηση Καταλληλότητας

Στο διάγραμμα των αισθητήρων του 3<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Παρατηρούμε ότι το συγκεκριμένο όχημα δεν παρουσιάζει δυσκολία να αποφύγει τα εμπόδια. Η ταχύτητα του διατηρείται σε υψηλά επίπεδα. Στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται για τα 2 σημεία στόχους αλλά δεν καταφέρνει να τα προσεγγίσει.

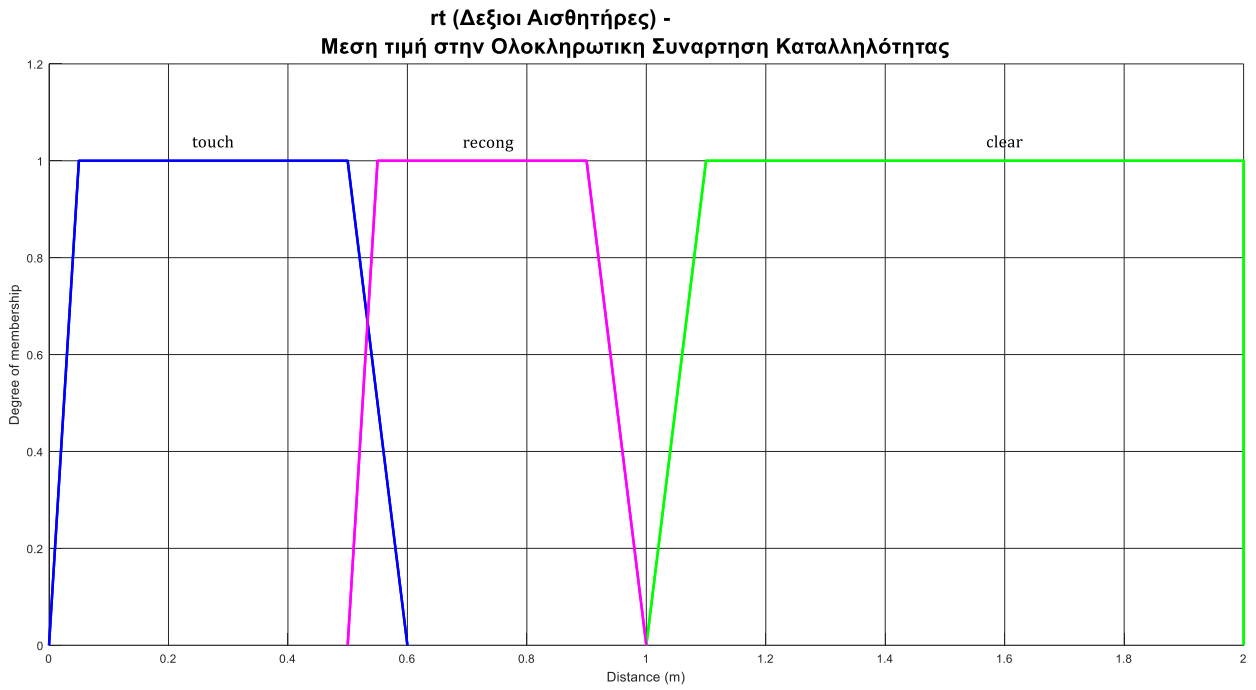
Ελεγκτή : Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή ολοκληρωτικής συνάρτησης καταλληλότητας



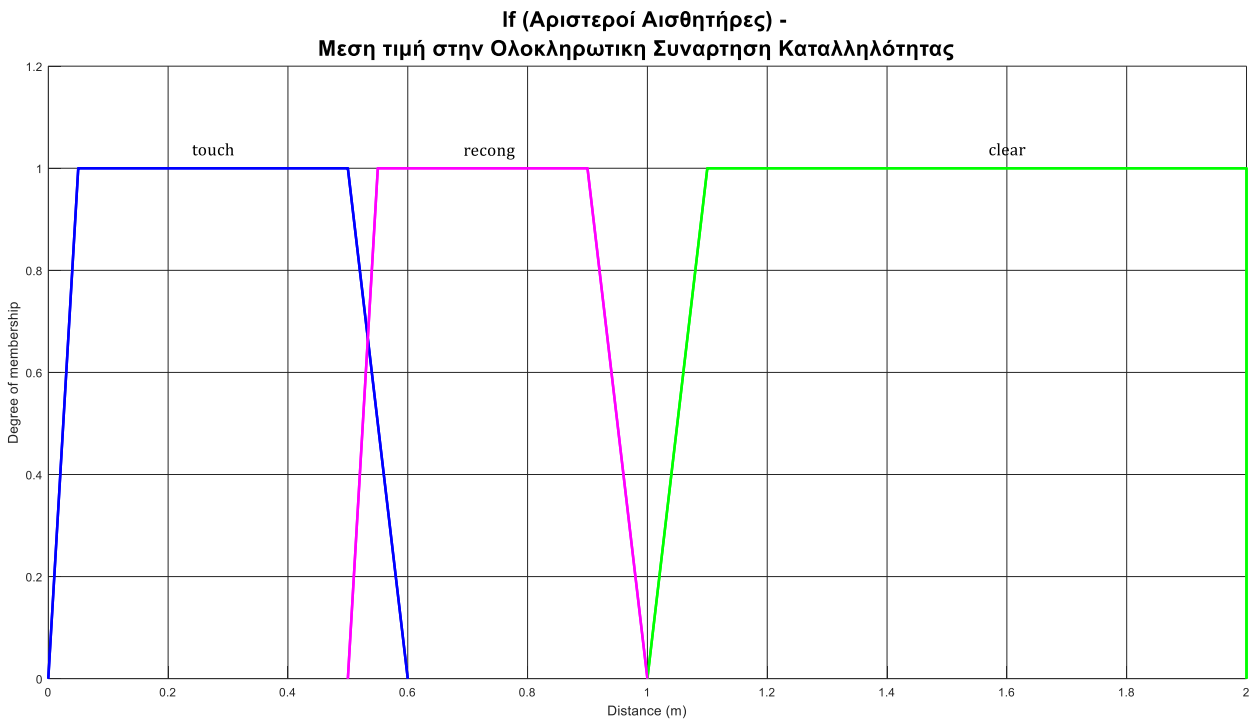
Εικόνα 136: Είσοδος Γωνία Απόκλισης – Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας



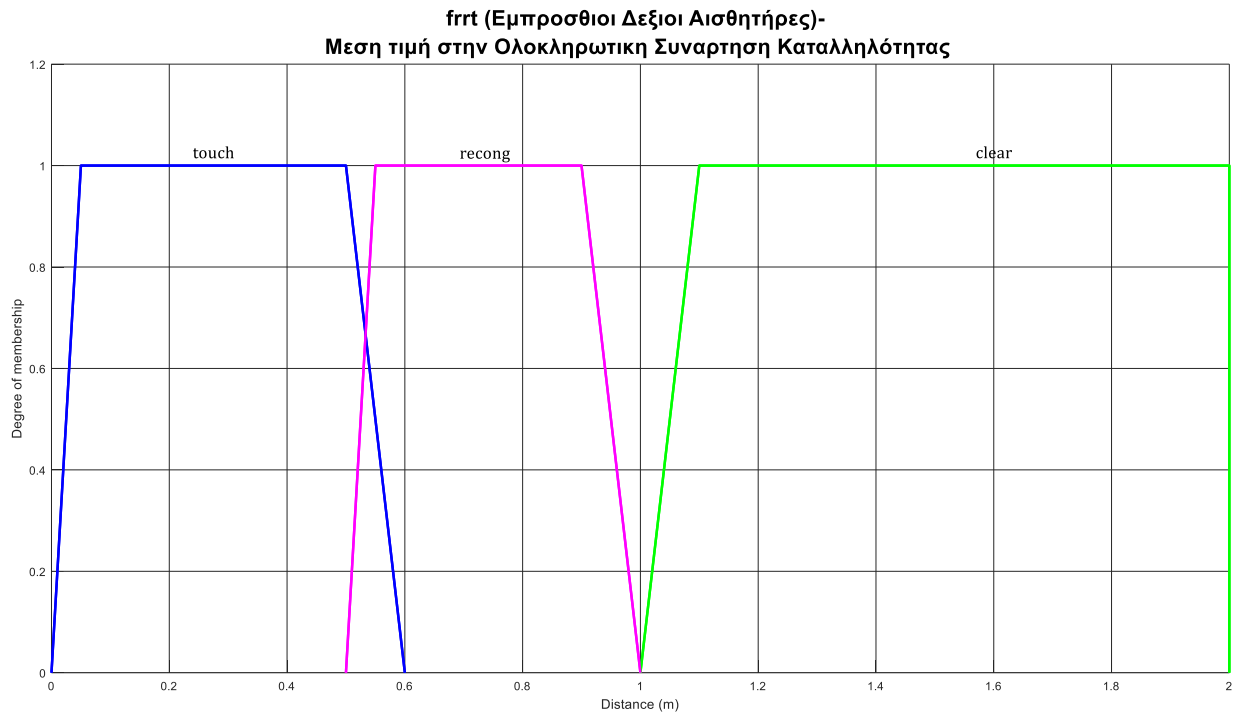
Εικόνα 137: Είσοδος Εμπρόσθοι Αισθητήρες – Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας



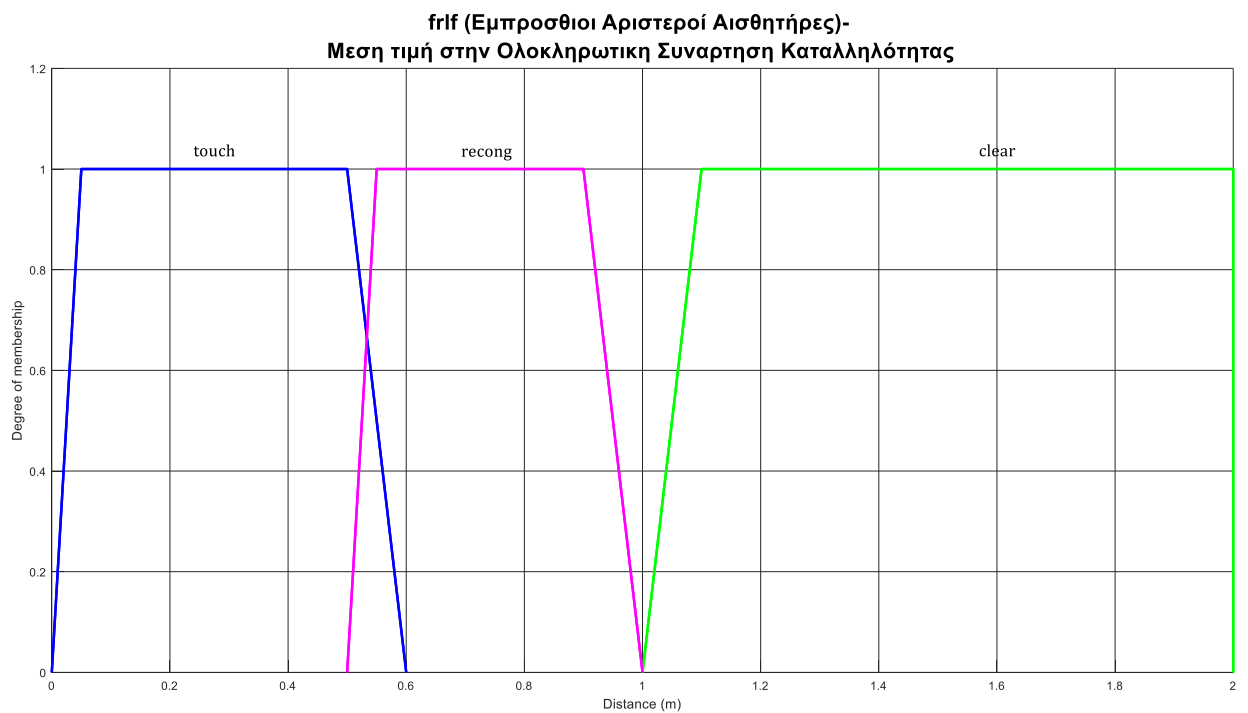
Εικόνα 138 : Είσοδος Δεξιοί Αισθητήρες - Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας



Εικόνα 139: Είσοδος Αριστεροί Αισθητήρες - Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας

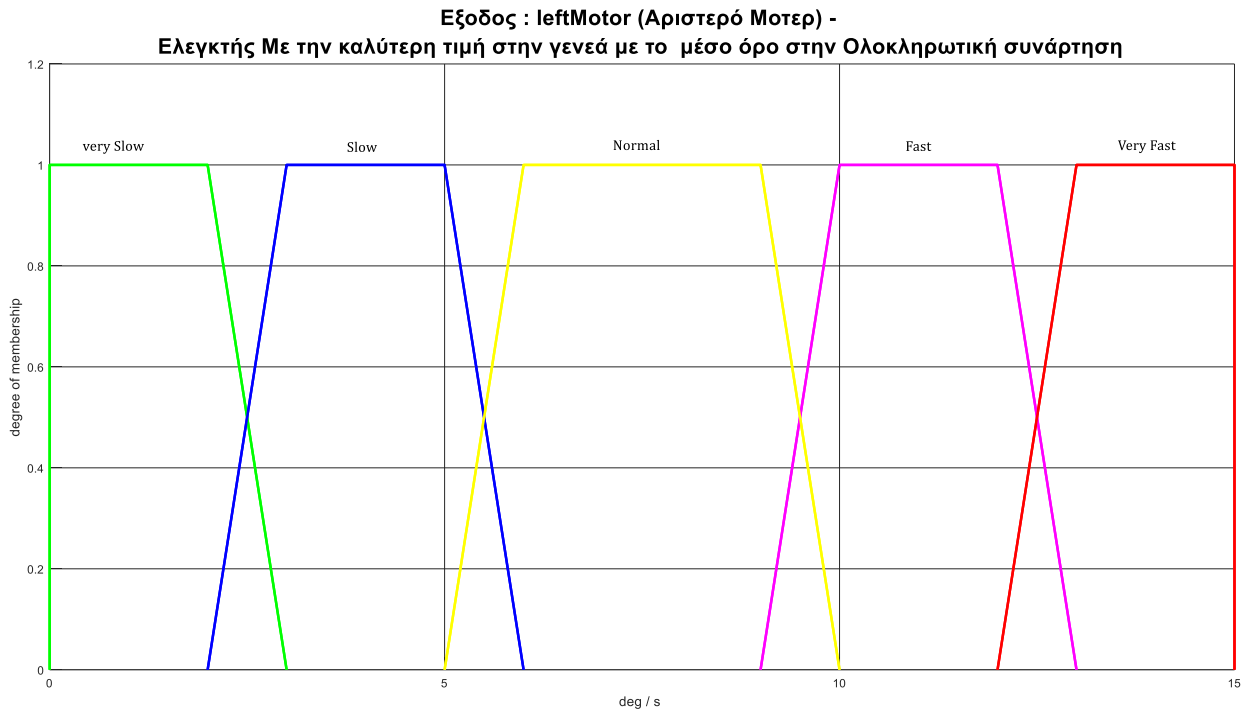


Εικόνα 140 : Είσοδος Εμπρόσθιοι Δεξιοί Αισθητήρες – Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας

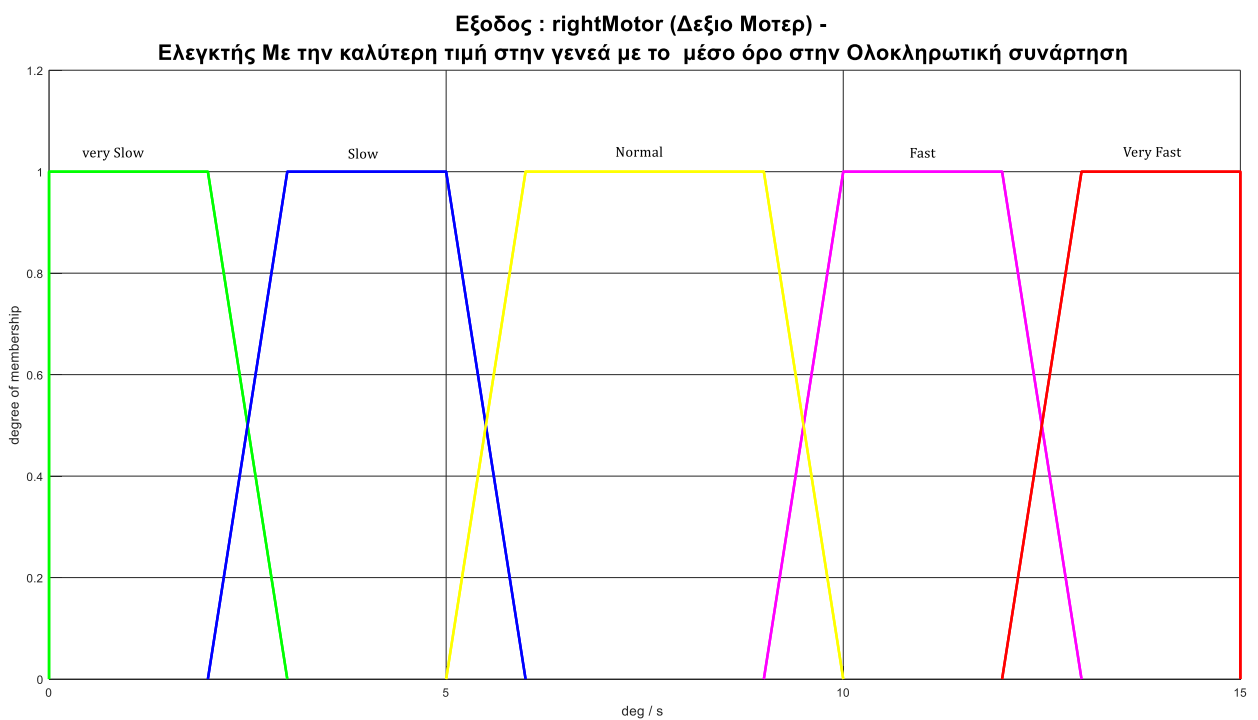


Εικόνα 141: Είσοδος Εμπρόσθιοι Αριστεροί Αισθητήρες – Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας



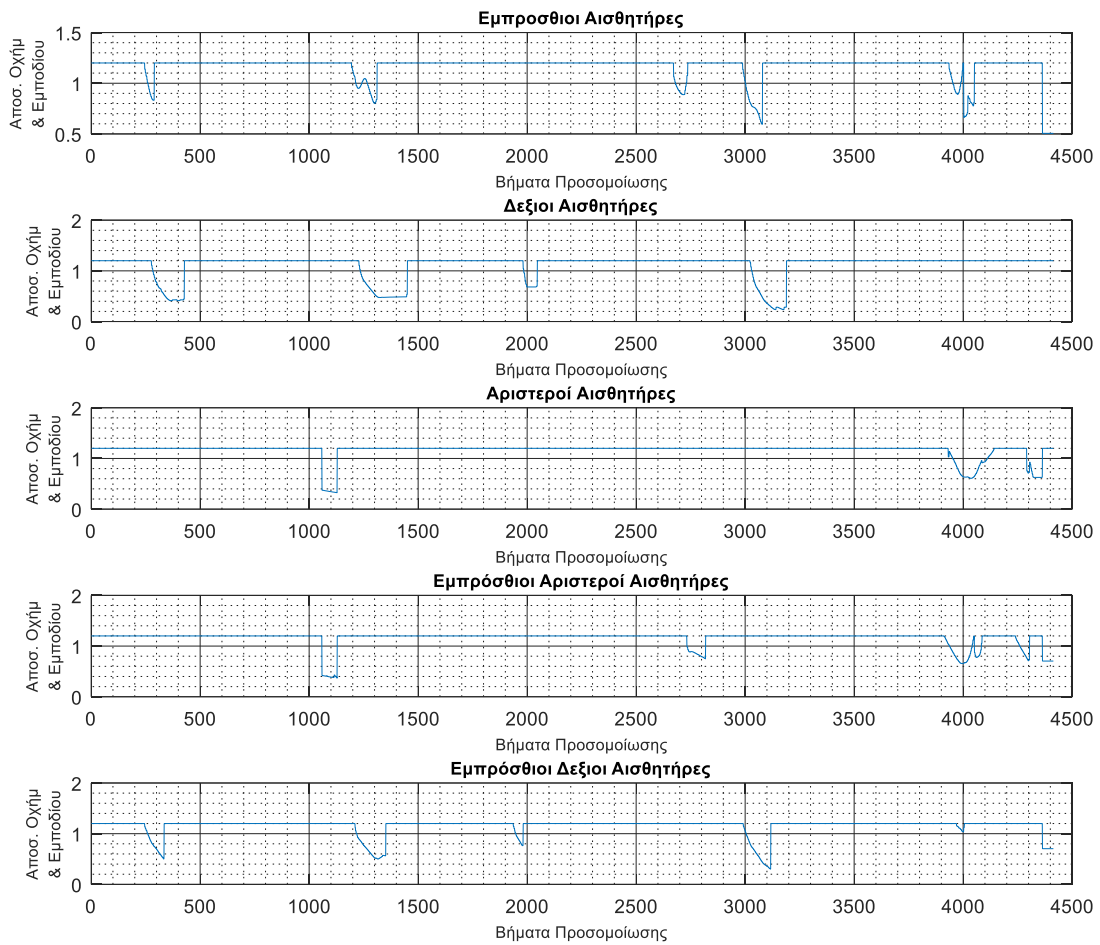


Εικόνα 142 : Έξοδος ταχύτητες αριστερού μοτέρ – Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας

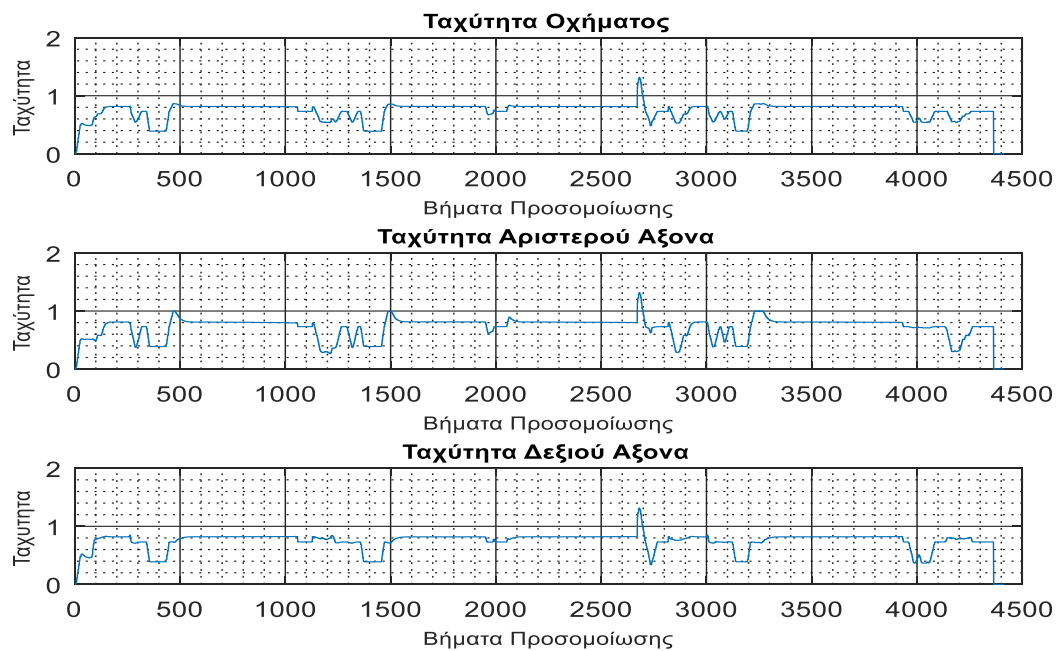


Εικόνα 143 : Έξοδος ταχύτητες δεξιού μοτέρ – Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας

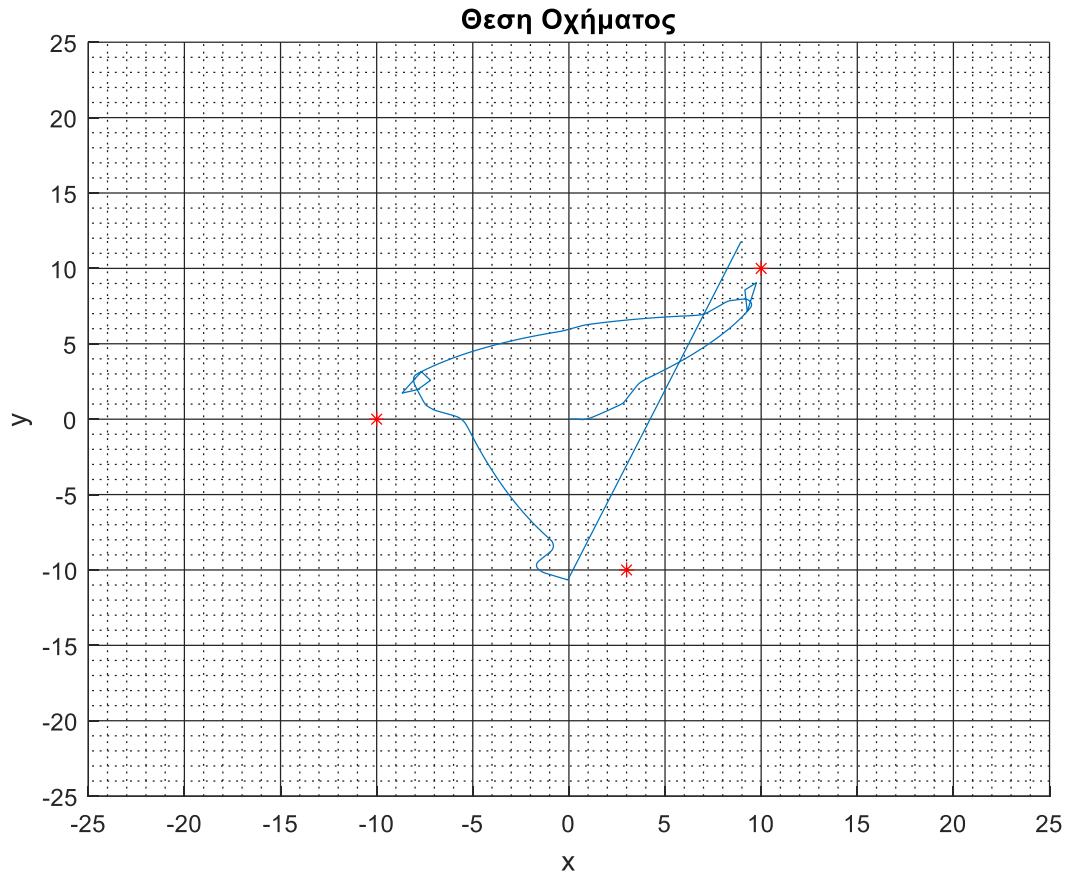
## Αποτελέσματα προσομοίωσης 1<sup>ου</sup> Οχήματος



Εικόνα 144 : Δεδομένα Αισθητήρων 1ου Οχήματος – Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας



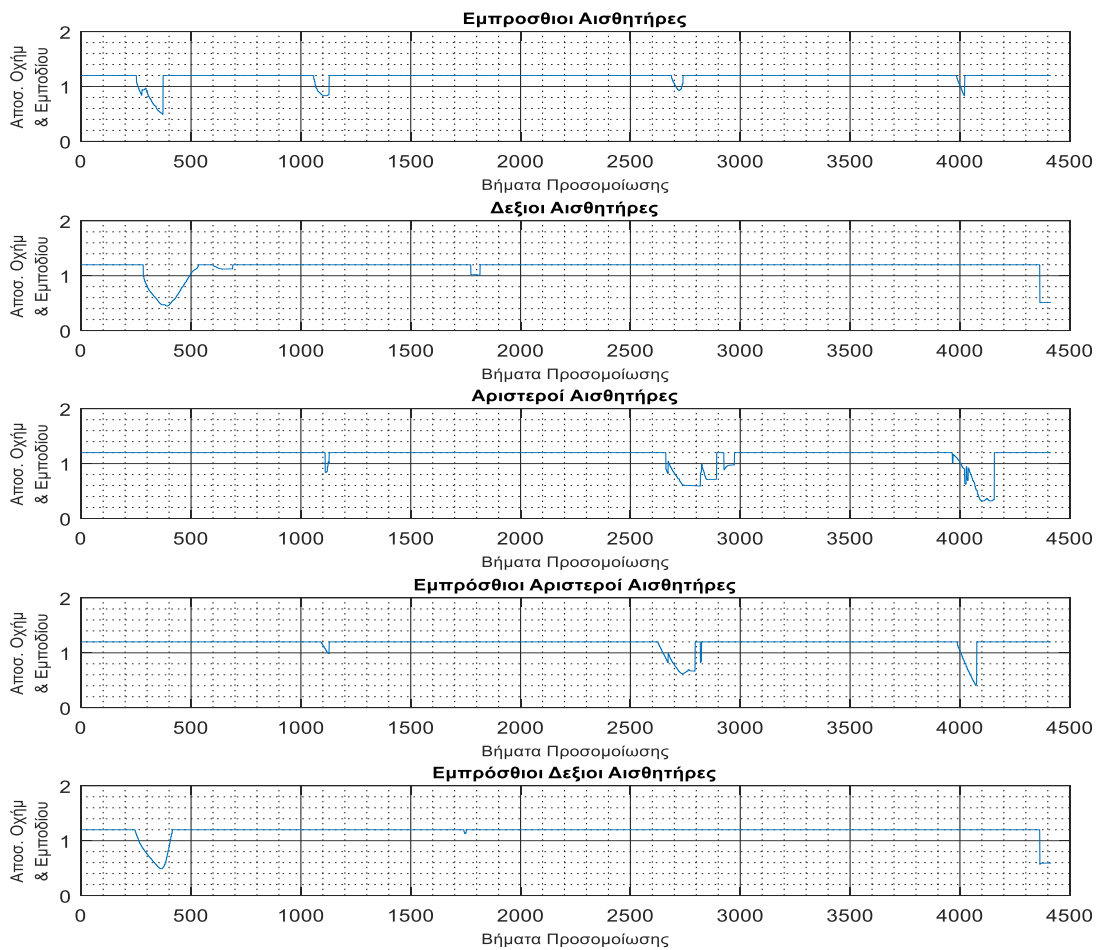
Δεδομένα Ταχυτήτων 1ου Οχήματος – Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας



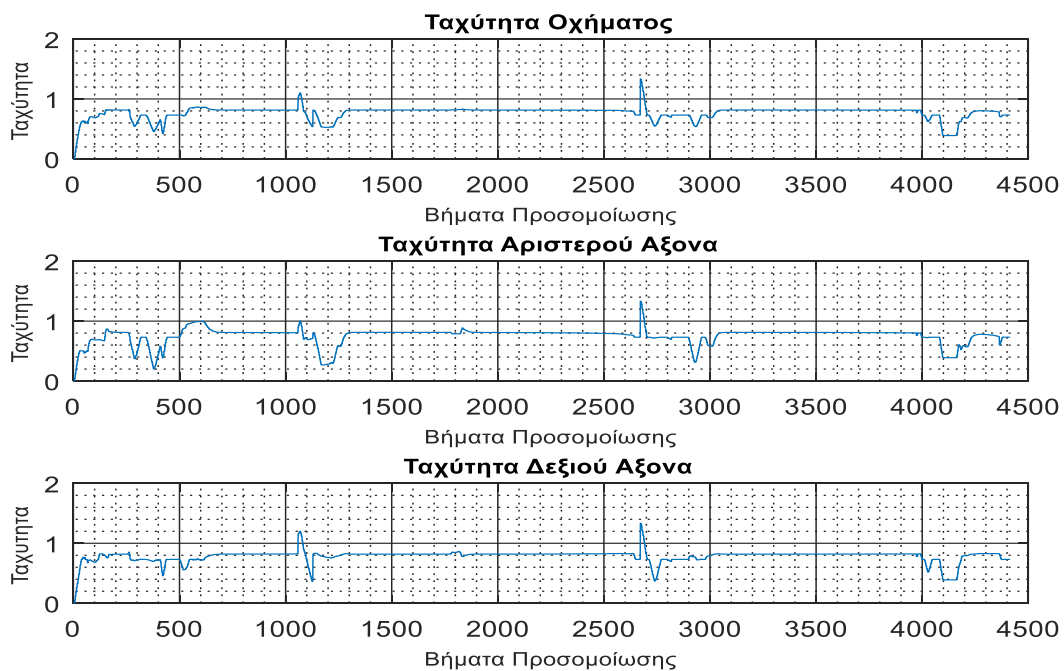
Εικόνα 145 : Διαδρομή 1ου Οχήματος – Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας

Στο διάγραμμα των αισθητήρων του 1<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και ο ελεγκτής καταφέρνει πολύ εύκολα και γρήγορα να αποφεύγει την αιτία ενεργοποίησης. Η ταχύτητα του έχει σταθερή σχεδόν υψηλή τιμή, σχεδόν γραμμική. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει. Παρουσιάζει πολύ πιο γραμμική λειτουργία σε σχέση με το 2<sup>ο</sup> όχημα του αρχικού ελεγκτή.

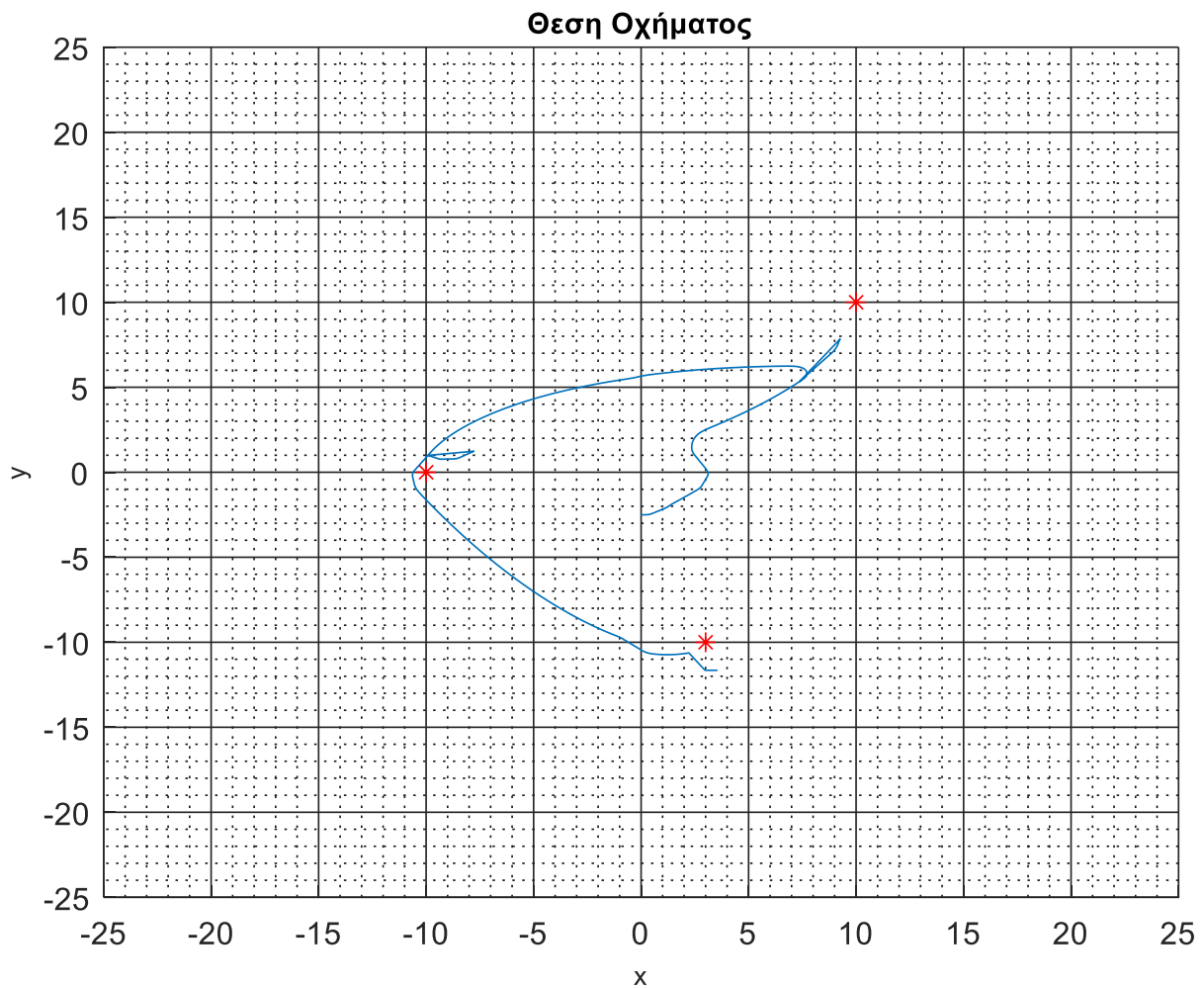
## Αποτελέσματα προσομοίωσης 2<sup>ου</sup> Οχήματος



Εικόνα 146 : Δεδομένα Αισθητήρων 2ου Οχήματος – Ελεγκτής Με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας



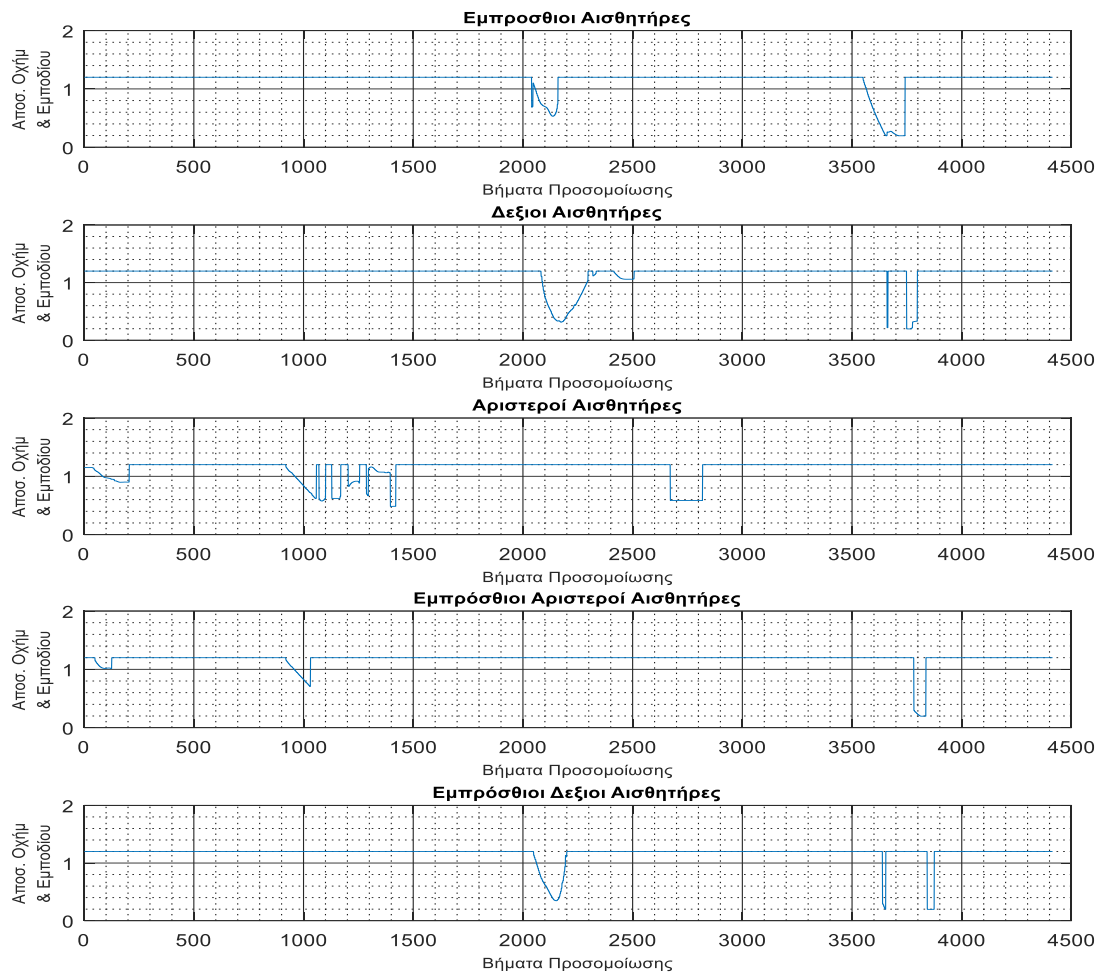
Εικόνα 147 : Δεδομένα Ταχύτητας 2ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας



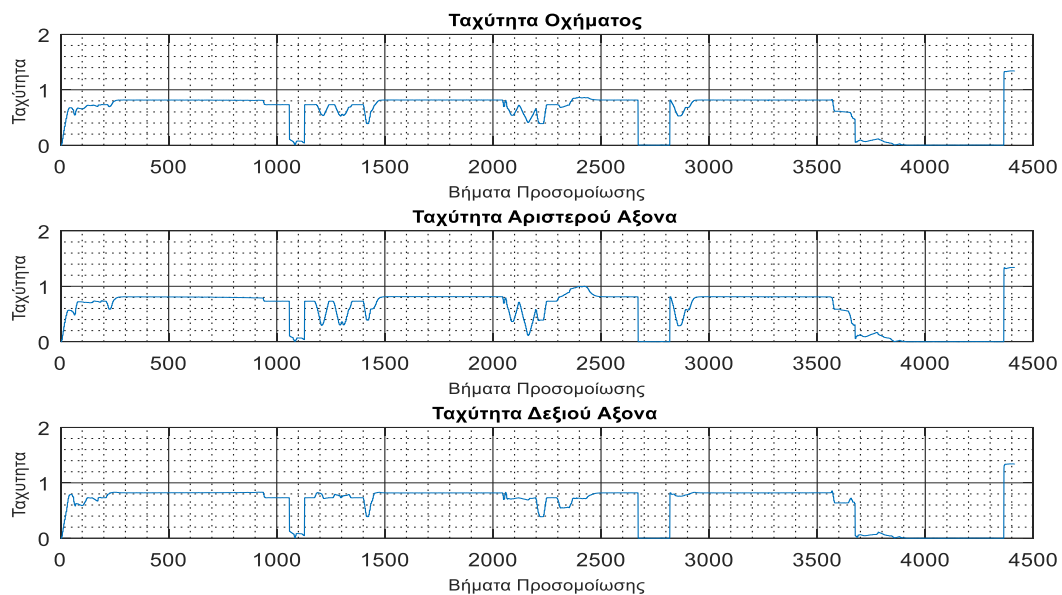
Εικόνα 148 : Διαδρομή 2ου οχήματος – Ελεγκτής με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας

Στο διάγραμμα των αισθητήρων του 2<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Η ταχύτητα του έχει την αναμενόμενη αυξομείωση συνέπεια της ενεργοποίησης των αισθητήρων αλλά διατηρείται σε υψηλή τιμή και έχει μια γραμμική αίσθηση. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει. Παρουσιάζει πολύ πιο γραμμική λειτουργία σε σχέση με το 2<sup>ο</sup> όχημα του αρχικού ελεγκτή.

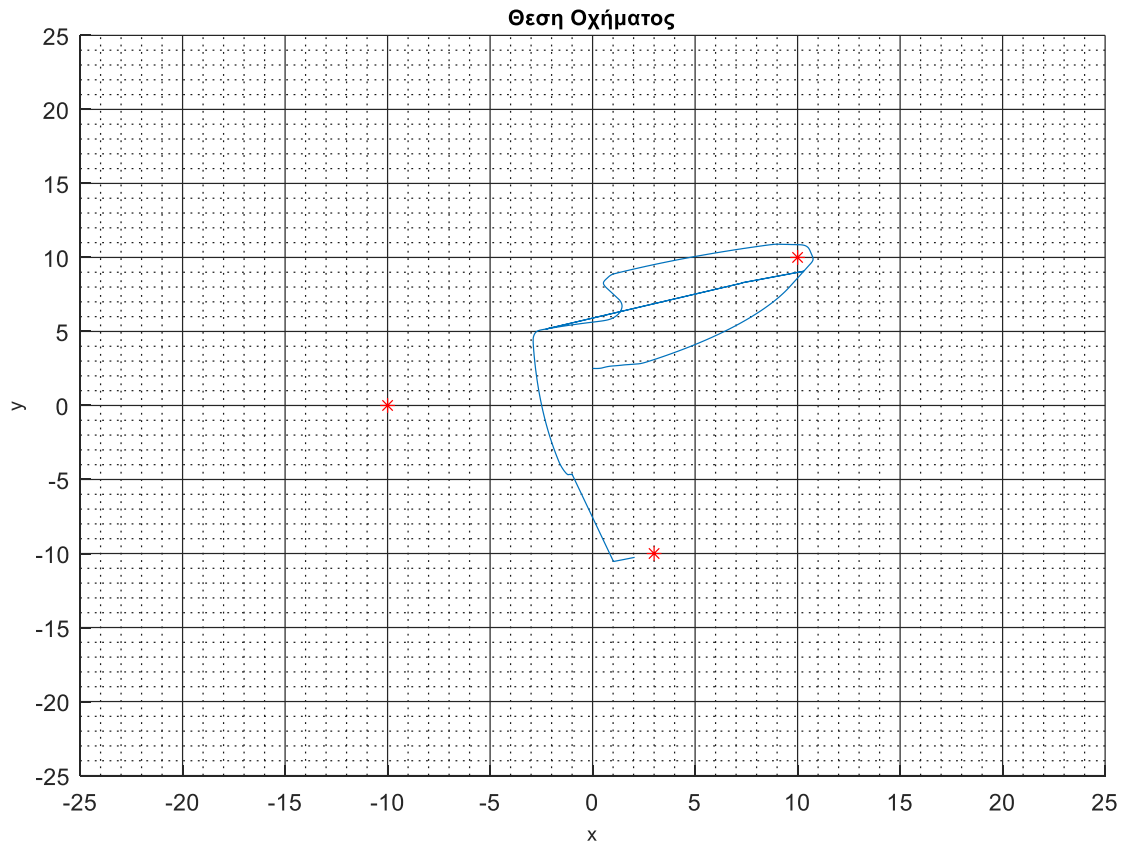
## Αποτελέσματα προσομοίωσης 3<sup>ου</sup> Οχήματος



Εικόνα 149 : Δεδομένα Αισθητήρων 3<sup>ου</sup> Οχήματος – Ελεγκτής με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας



Εικόνα 150 : Δεδομένα ταχύτητας 3<sup>ου</sup> Οχήματος –Ελεγκτής με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας



Εικόνα 151 : Διαδρομή 3ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην γενεά με το μέσο όρο στην Ολοκληρωτική συνάρτηση καταλληλότητας

Στο διάγραμμα των αισθητήρων του 3<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Η ταχύτητα του έχει την αναμενόμενη αυξομείωση συνέπεια της ενεργοποίησης των αισθητήρων. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει.

Οι ελεγκτές που εξήχθησαν με την ολοκληρωτική συνάρτηση καταλληλότητας παρουσιάζουν βελτιστοποίησης στον χρόνο της προσομοίωσης. Οι ελεγκτές δεν διαφοροποιούνται επαρκώς από τον αρχικό και αυτό μας οδηγεί στο συμπέρασμα ότι ο αριθμός των γενεών που έτρεξε ο αλγόριθμος ήταν μικρός με αποτέλεσμα την μικρή διαφοροποίηση των ελεγκτών. Στον πίνακα 8 παρουσιάζονται συγκεντρωτικά τα αποτελέσματα των προσομοιώσεων που έγιναν με χρήση των ελεγκτών που εξήχθησαν από την εξελικτική διαδικασία με την ολοκληρωτική ή αθροιστική συνάρτηση καταλληλότητας.

|                 | <i>S</i> | <i>U</i> | <i>Ul</i> | <i>Ur</i> | <i>Th</i> | <i>TotalTime</i> |
|-----------------|----------|----------|-----------|-----------|-----------|------------------|
| <i>Original</i> | 1,1029   | 0,8713   | 0,8660    | 0,8760    | 0         | 123,43           |
| <i>min</i>      | 1.094    | 0.646    | 0.641     | 0.652     | 0         | 85.12            |
| <i>mean</i>     | 1.154    | 0.706    | 0.693     | 0.718     | 0         | 84.40            |

|             |        |         |         |         |  |         |
|-------------|--------|---------|---------|---------|--|---------|
| <i>min</i>  | -0,81% | -25,86% | -25,98% | -25,57% |  | -31,04% |
| <i>mean</i> | 4,43 % | -23,41% | -24,96% | -22,01% |  | -46,25% |

Πίνακας 8: Σύγκριση αποτελεσμάτων εξέλιξης με ολοκληρωτική ή αθροιστική συνάρτηση συμπεριφοράς

## 4.6 Αποτελέσματα Εξελικτικής Διαδικασίας : Προσαρμοσμένη συνάρτηση καταλληλότητας (Tailored)

Η προσαρμοσμένη συνάρτηση καταλληλότητας μετράει τα δεδομένα της προσομοίωσης αλλά και αν έχει επιτευχθεί ο στόχος του πειράματος, στην προκειμένη περίπτωση αν έχουν προσεγγιστεί τα σημεία - στόχοι ή όχι.

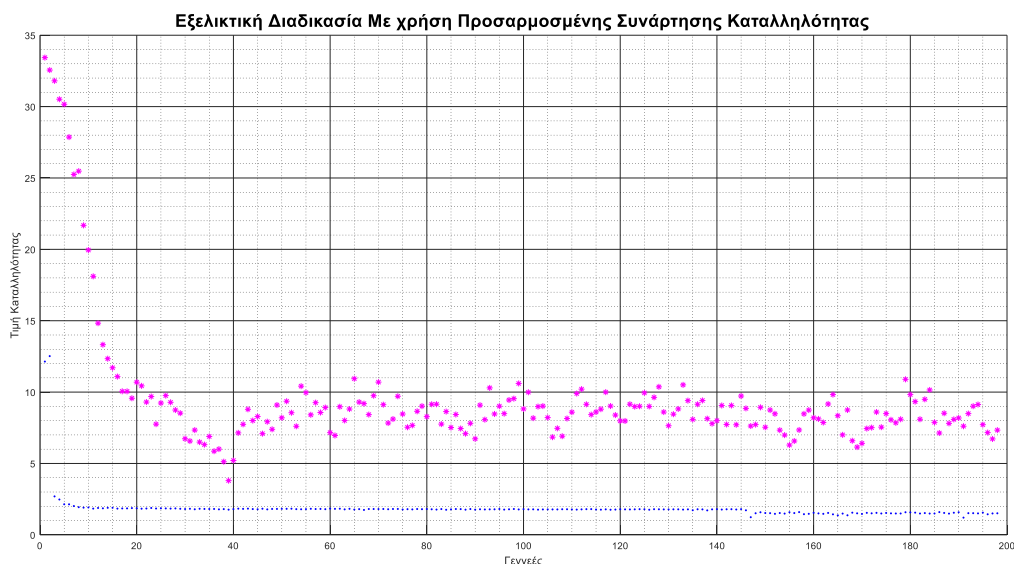
$$f = th + (2.4 / (S + U));$$

Ανάλογα τον αριθμό των σημείων που έχουν προσεγγίσει είναι και η βαθμολογία του  $th$ . Προσέγγιση 3 σημείων ποινή 0, προσέγγιση 2 σημείων ποινή 10, προσέγγιση 1 σημείου ποινή 20 και προσέγγιση 0 σημείων ποινή 30. Με στόχο το μικρότερο είναι το καλύτερο.

2.4 είναι η μέγιστη τιμή του αθροίσματος των μέγιστων τιμών που μπορεί να λάβουν οι αισθητήρες  $S = 1.2$  και η μέγιστη τιμή της ταχύτητας του οχήματος  $U = 1.2$

$S$  = μέση τιμή των τιμών που επιστρέφουν όλοι οι αισθητήρες  $frD, lfD, rtD, frlfD, frrtD$

$U$  = μέση τιμή ταχύτητας οχήματος



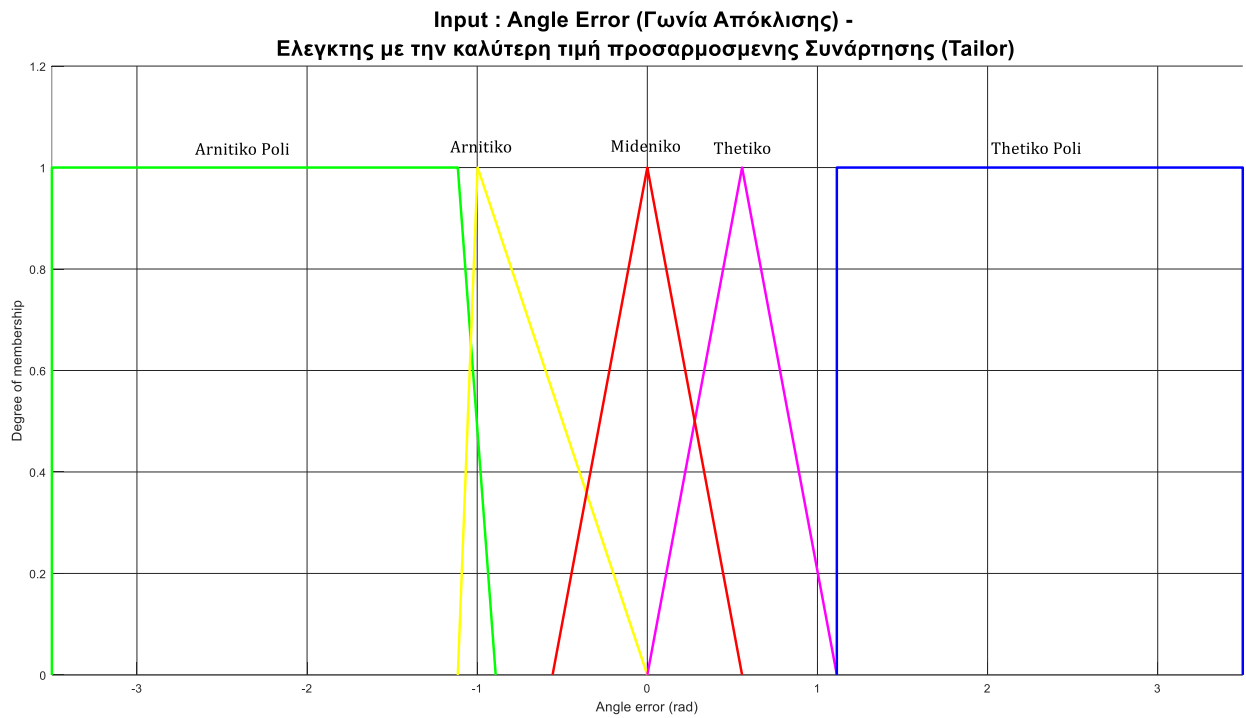
Εικόνα 152: Εξέλιξη βελτιστοποίηση ασαφούς ελεγκτή με προσαρμοσμένη συνάρτηση καταλληλότητας

Παρατηρούμε ότι υπάρχει βελτιστοποίηση της μέσης τιμής της συνάρτησης ώστε ο μέσος όρος όλων των συναρτήσεων ανά γενεά να προσεγγίσει την τιμή του καλύτερου ελεγκτή ανά γενεά και παράλληλα βλέπουμε ότι βελτιώνεται και η τιμή του fitness. Ο αριθμός των γενεών που έγινε η εξελικτική διαδικασία ήταν ικανοποιητικός. Παρουσιάζεται το φαινόμενο της καθυστέρησης κάθε φορά που σταματούσε η διαδικασία και ξεκινούσαμε από την τελευταία γενεά. Το φαινόμενο μπορεί να εντοπιστεί στην 40η γενεά, 90η γενεά και την 160η. Παρόλα αυτά εξάγουμε δυο ελεγκτές:

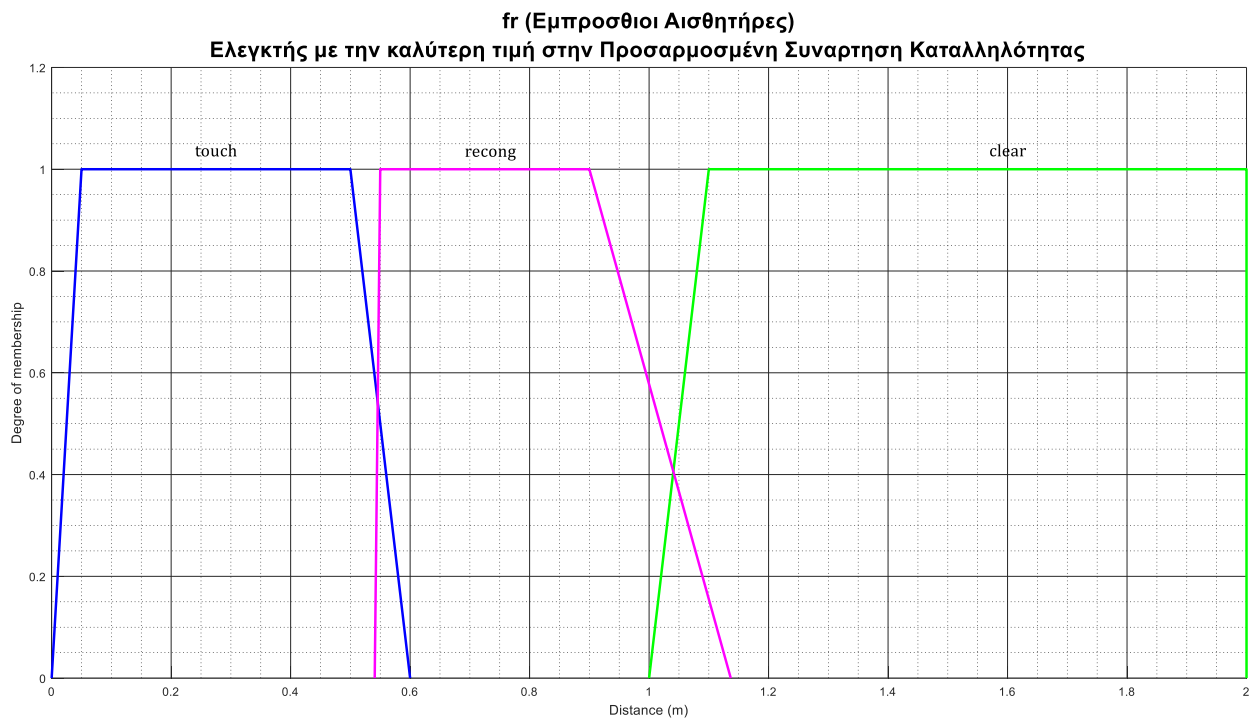
1. Τον ελεγκτή με την καλύτερη τιμή της συνάρτησης καταλληλότητας.
2. Τον καλύτερο ελεγκτή στην γενεά με τον καλύτερο μέσο όρο της τιμής της συνάρτησης καταλληλότητας.



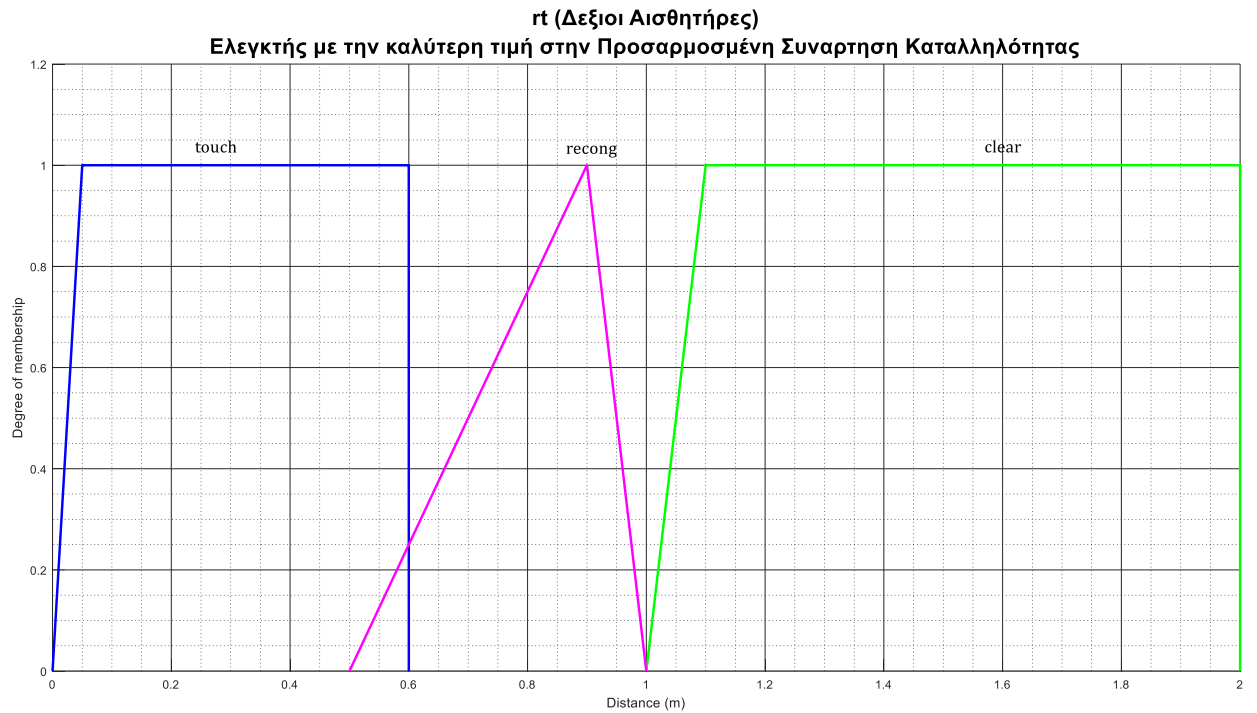
Ελεγκτή : με την καλύτερη τιμή στην προσαρμοσμένη συνάρτησης καταλληλότητας



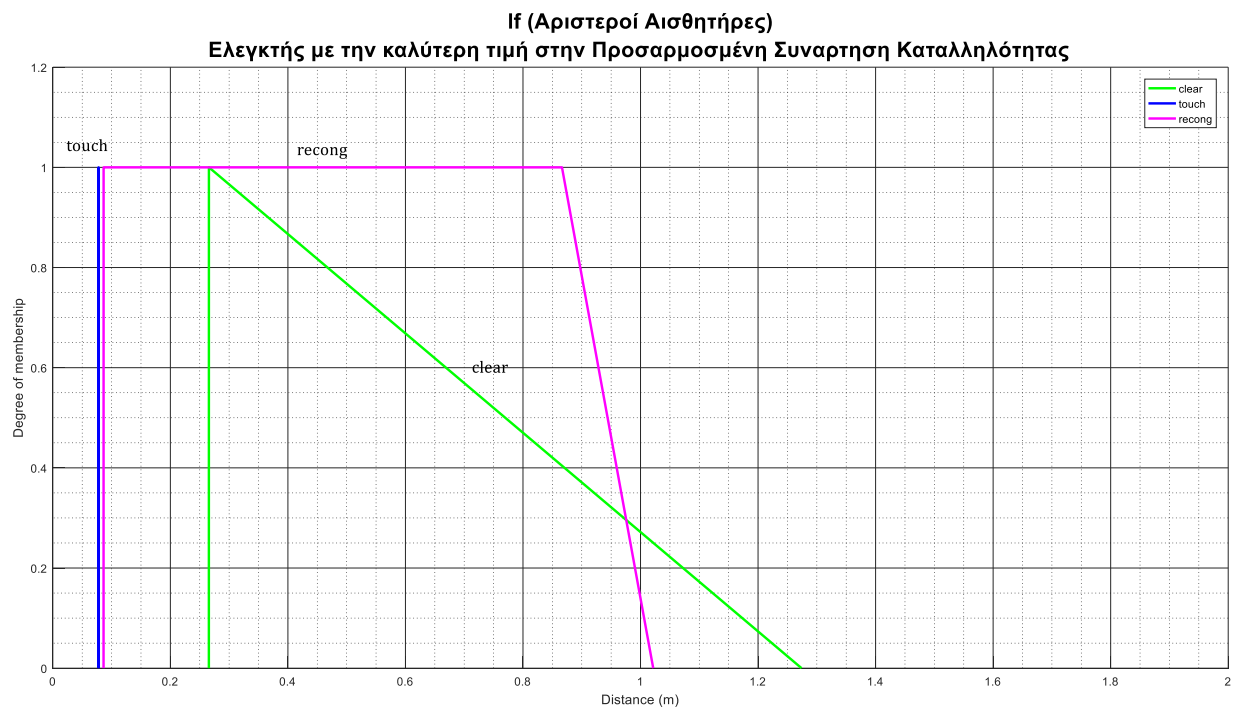
Εικόνα 153 : Είσοδος Γωνία Απόκλισης – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



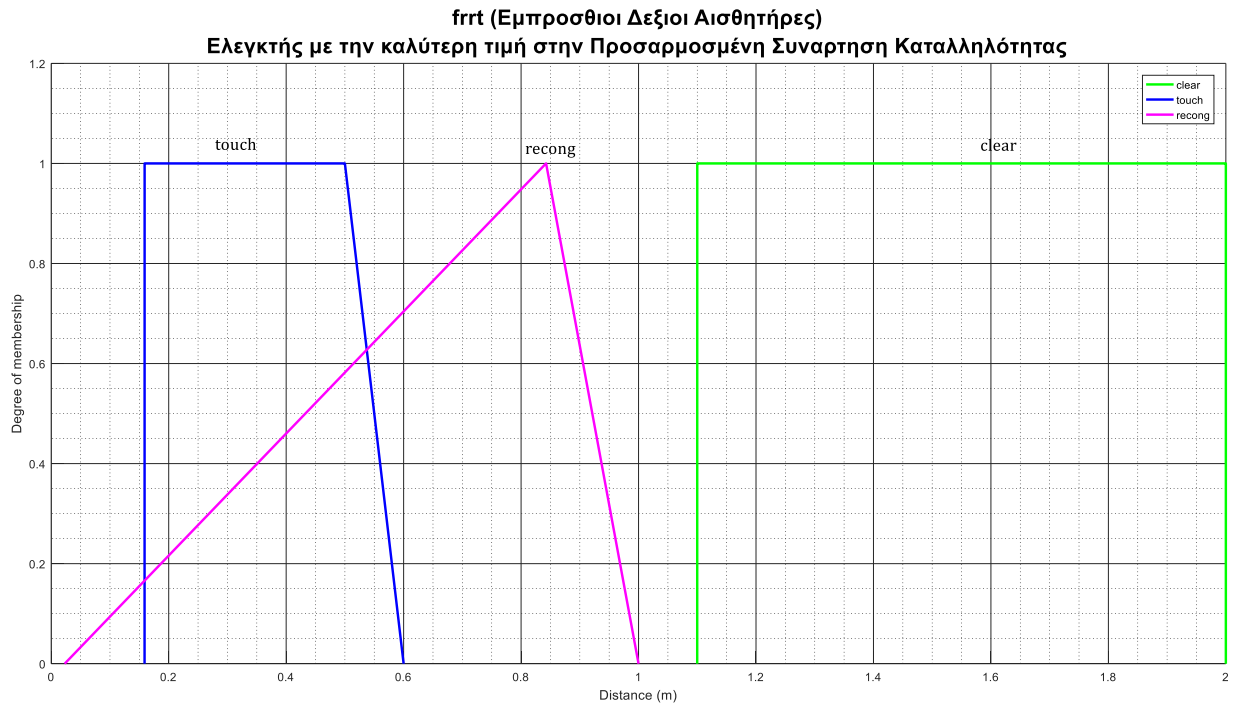
Εικόνα 154: Είσοδος Εμπρόσθιοι Αισθητήρες – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



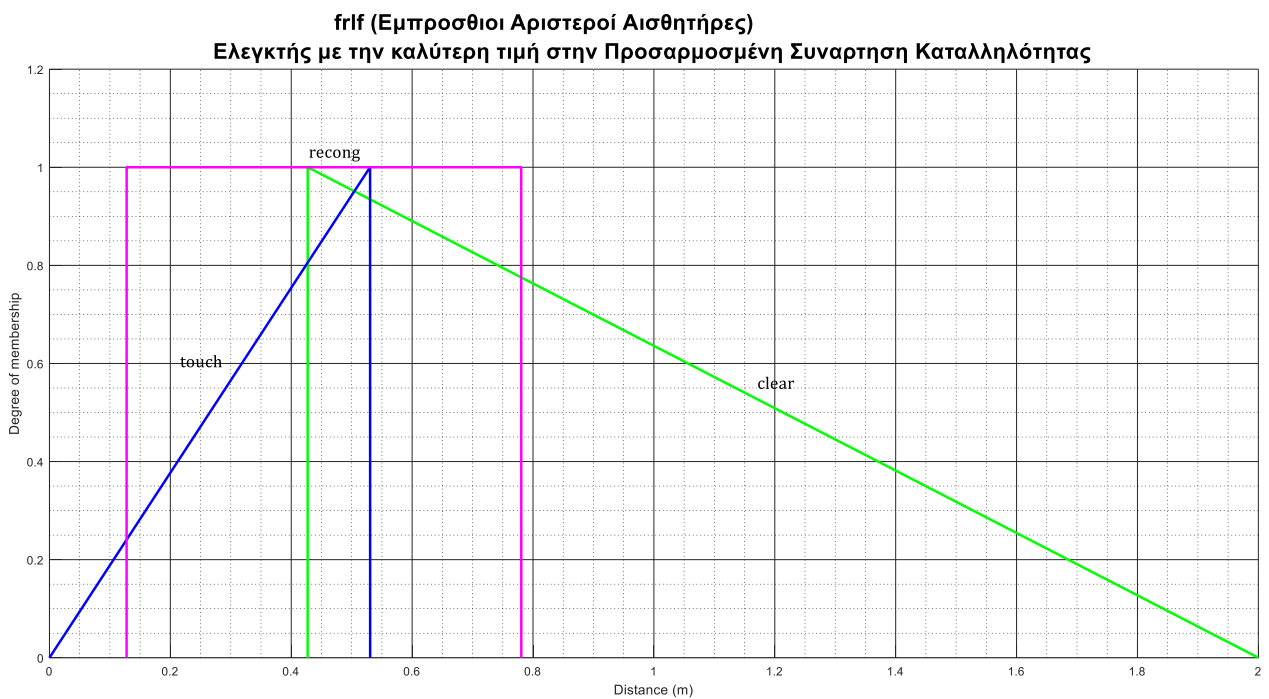
Εικόνα 155: Είσοδος Δεξιοί Αισθητήρες – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



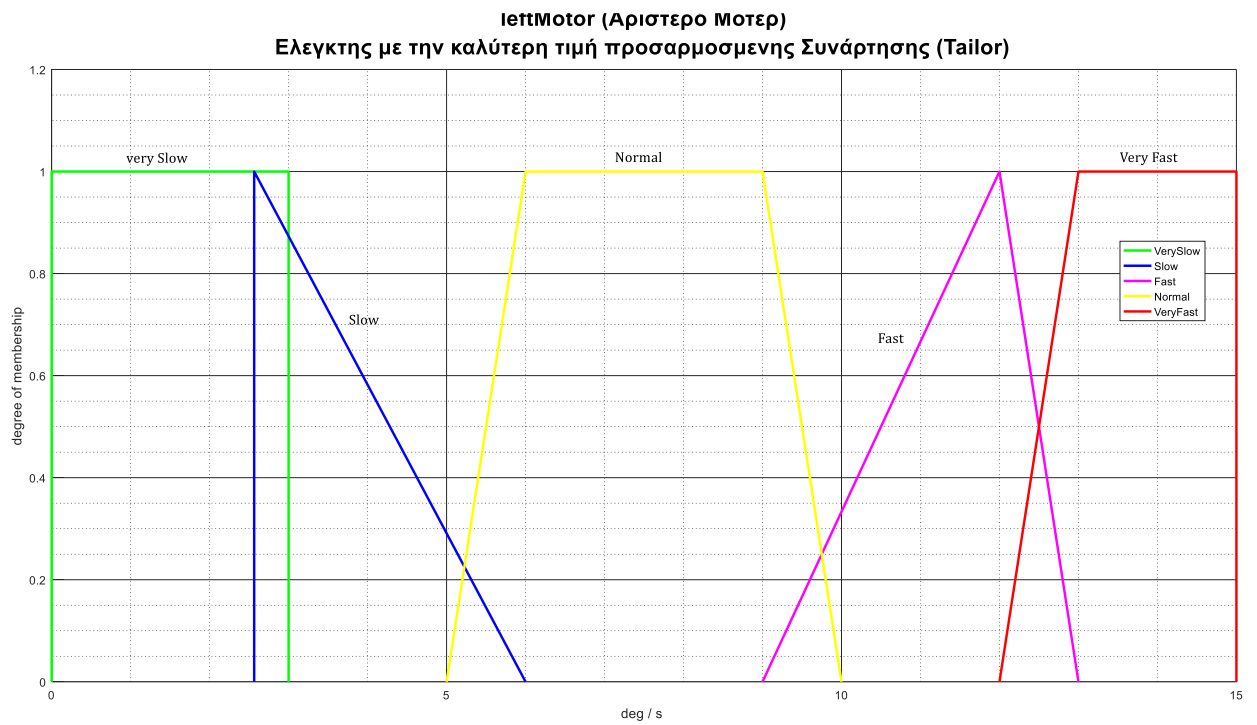
Εικόνα 156: Είσοδος Αριστεροί Αισθητήρες – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



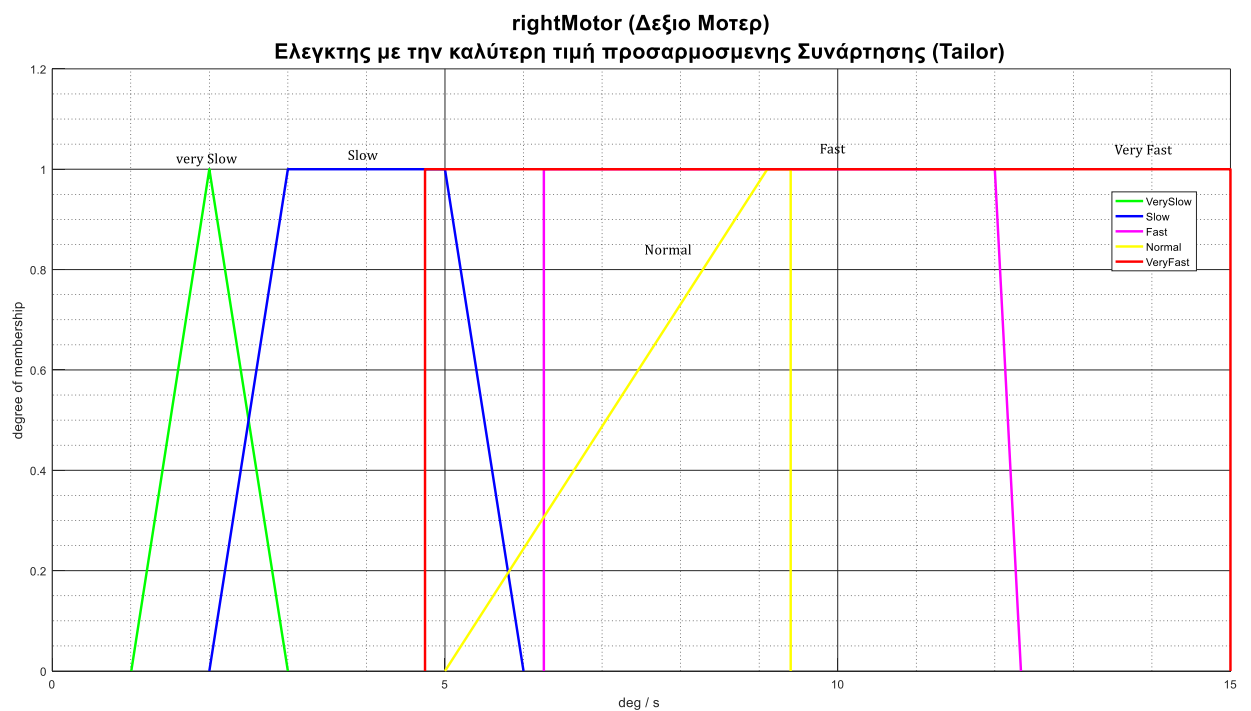
Εικόνα 157: Είσοδος Εμπρόσθοι Δεξιοί Αισθητήρες – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



Εικόνα 158: Είσοδος Εμπρόσθοι Αριστεροί Αισθητήρες – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

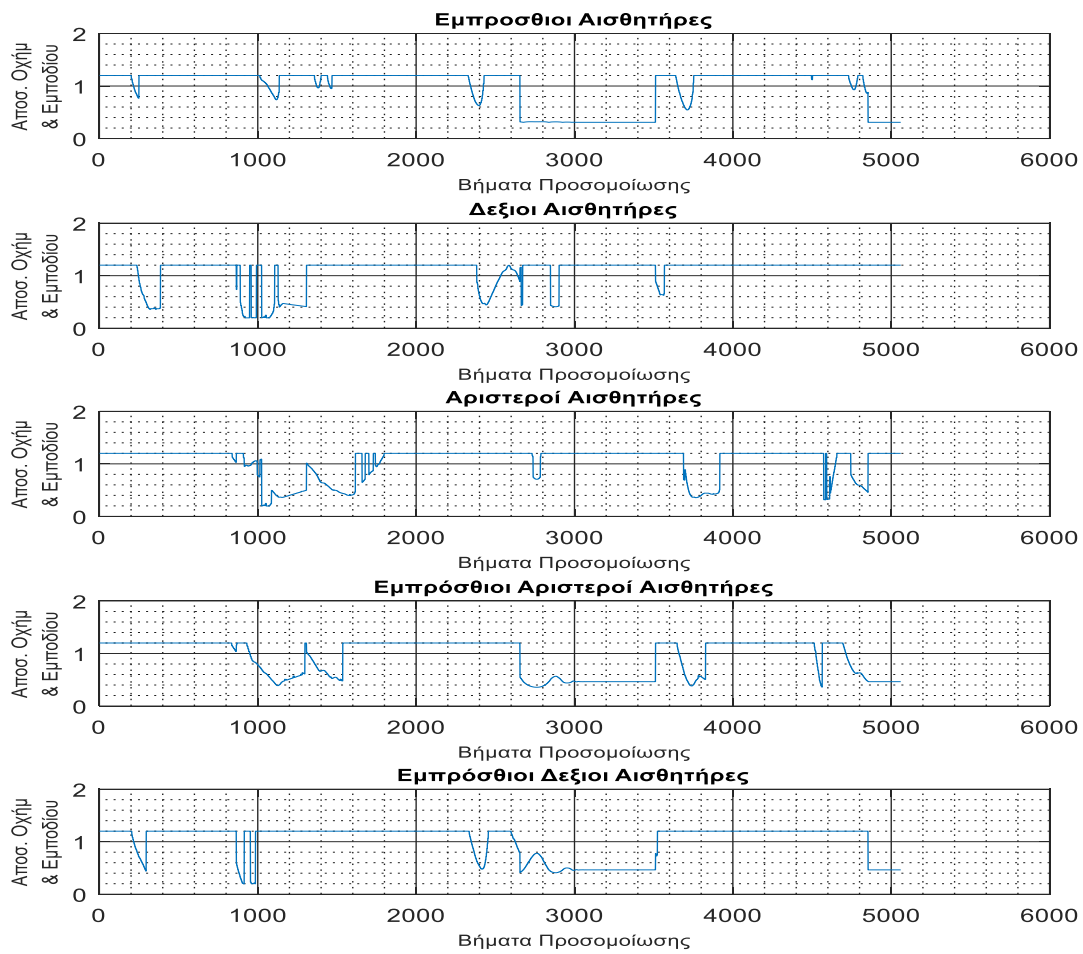


Εικόνα 159 : Έξοδος Ταχύτητα Αριστερού Μοτέρ – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

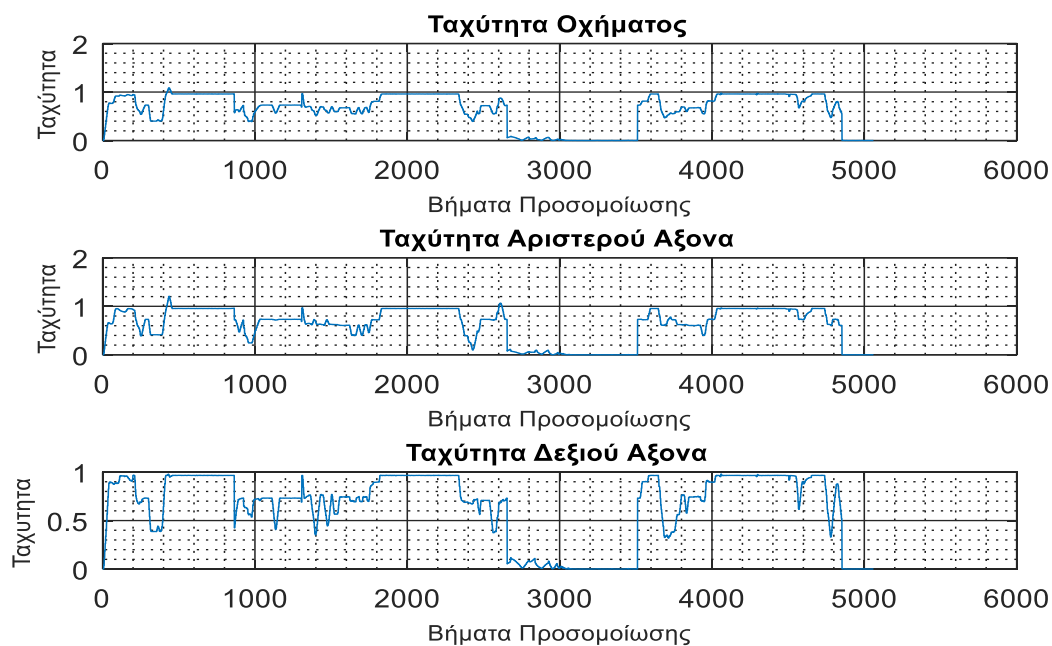


Εικόνα 160 : Έξοδος Ταχύτητα Δεξιού Μοτέρ – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

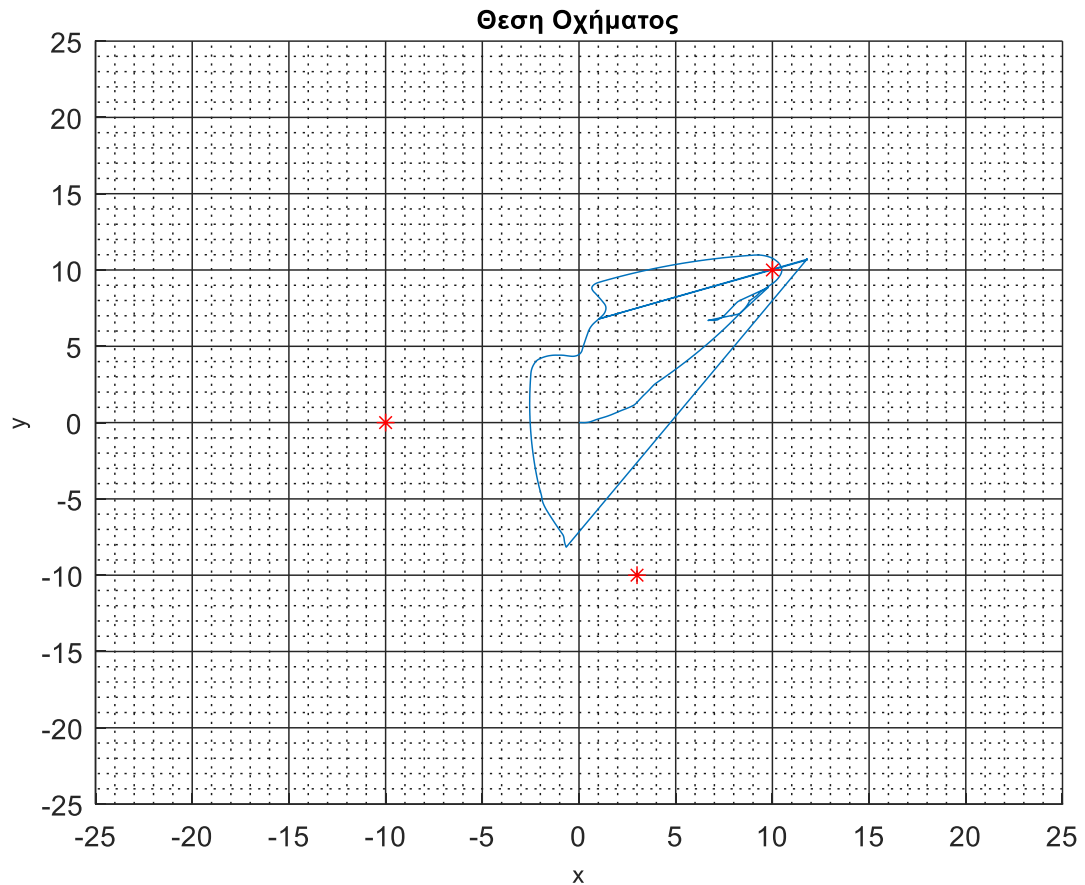
## Αποτελέσματα προσομοίωσης 1<sup>ου</sup> Οχήματος



Εικόνα 161: Δεδομένα Αισθητηρίων 1ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



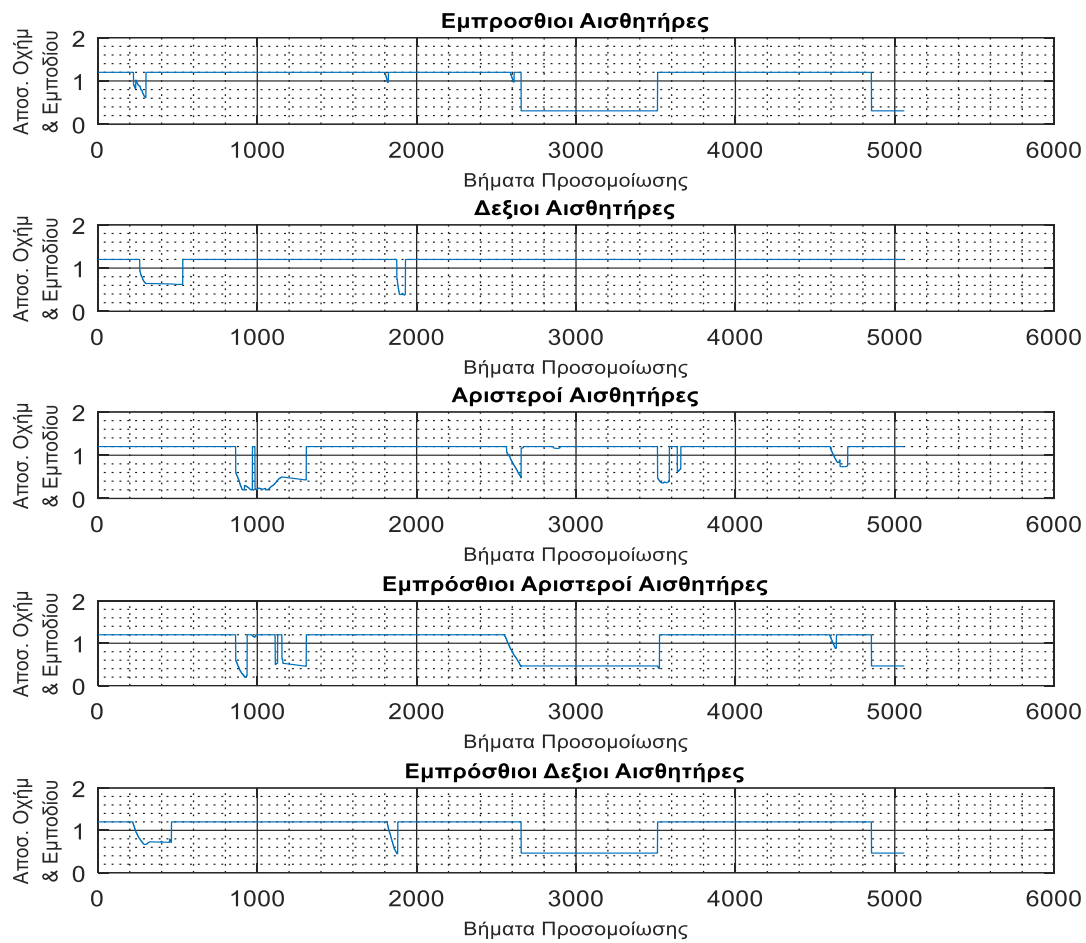
Εικόνα 162 : Δεδομένα Ταχύτητας 1ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



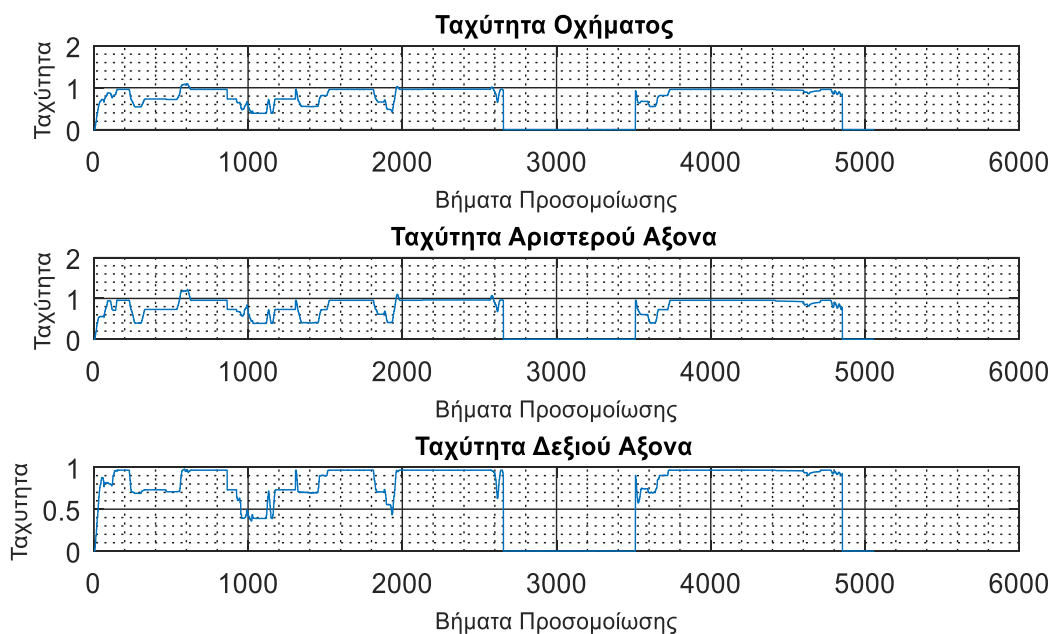
Εικόνα 163 : Διαδρομή 1ου Οχήματος - Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

Στο διάγραμμα των αισθητήρων του 1<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Η ταχύτητα του έχει την αναμενόμενη αυξομείωση συνέπεια της ενεργοποίησής των αισθητήρων αλλά γίνεται προσπάθεια να διατηρηθεί σε υψηλή τιμή. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει.

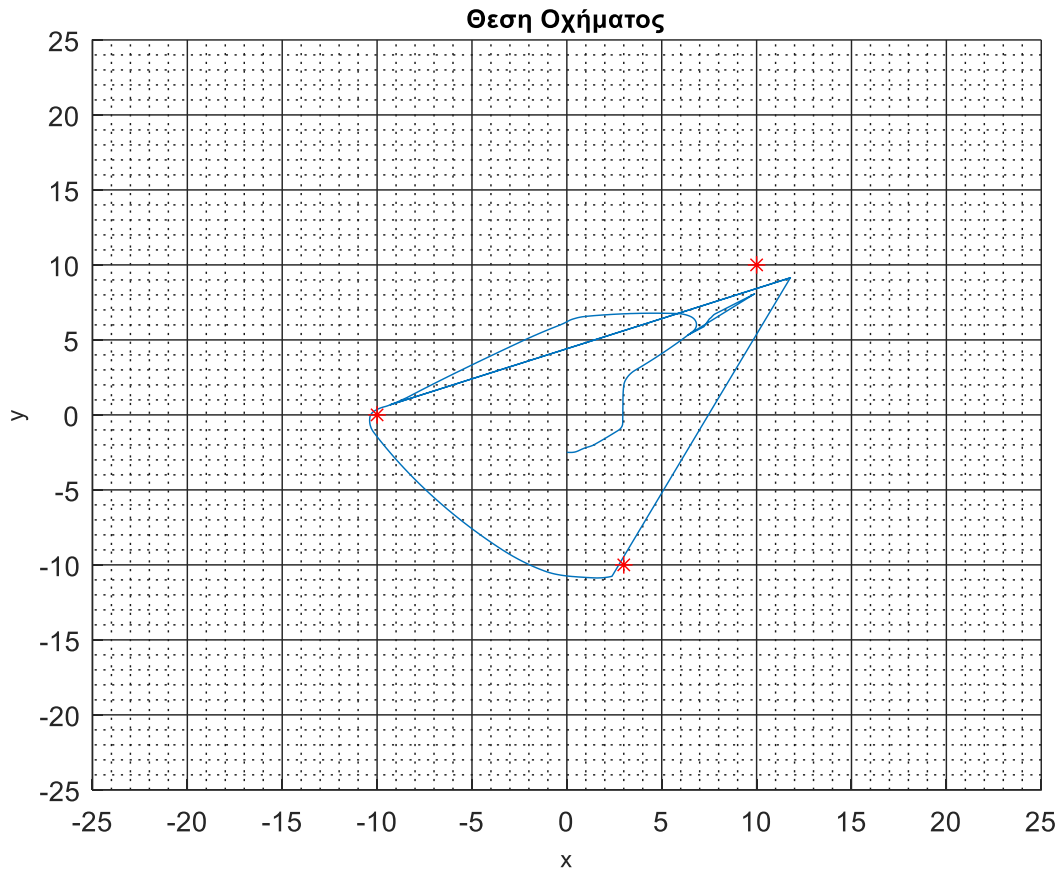
## Αποτελέσματα προσομοίωσης 2<sup>ου</sup> Οχήματος



Εικόνα 164: Δεδομένα Αισθητηρίων 2ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



Εικόνα 165 : Δεδομένα Ταχύτητας 2ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

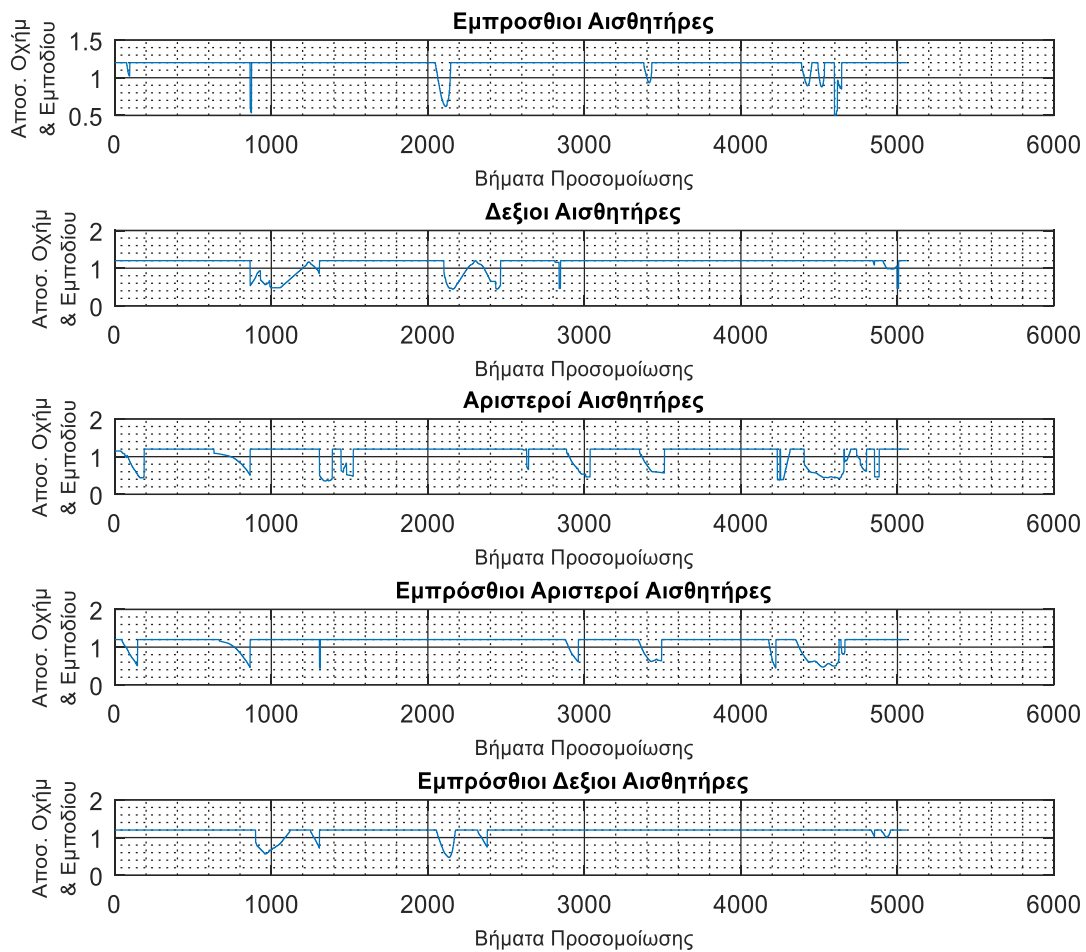


Εικόνα 166 : Διαδρομή 2ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

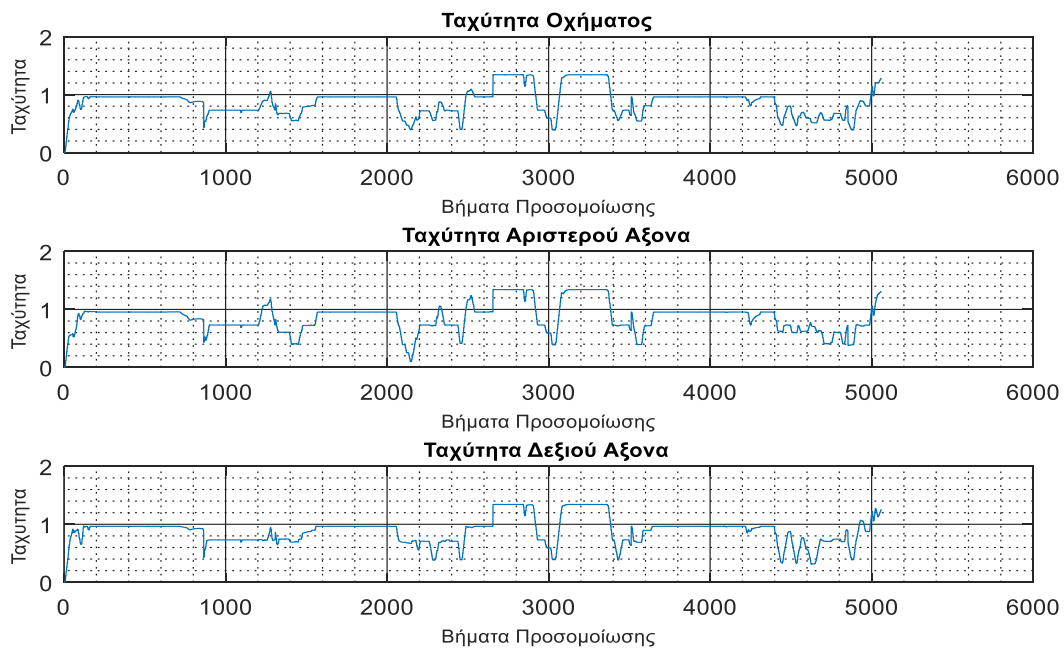
Στο διάγραμμα των αισθητήρων του 2<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Η ταχύτητα του έχει την αναμενόμενη αυξομείωση συνέπεια της ενεργοποίησής των αισθητήρων αλλά διατηρείται σε υψηλή τιμή. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει.



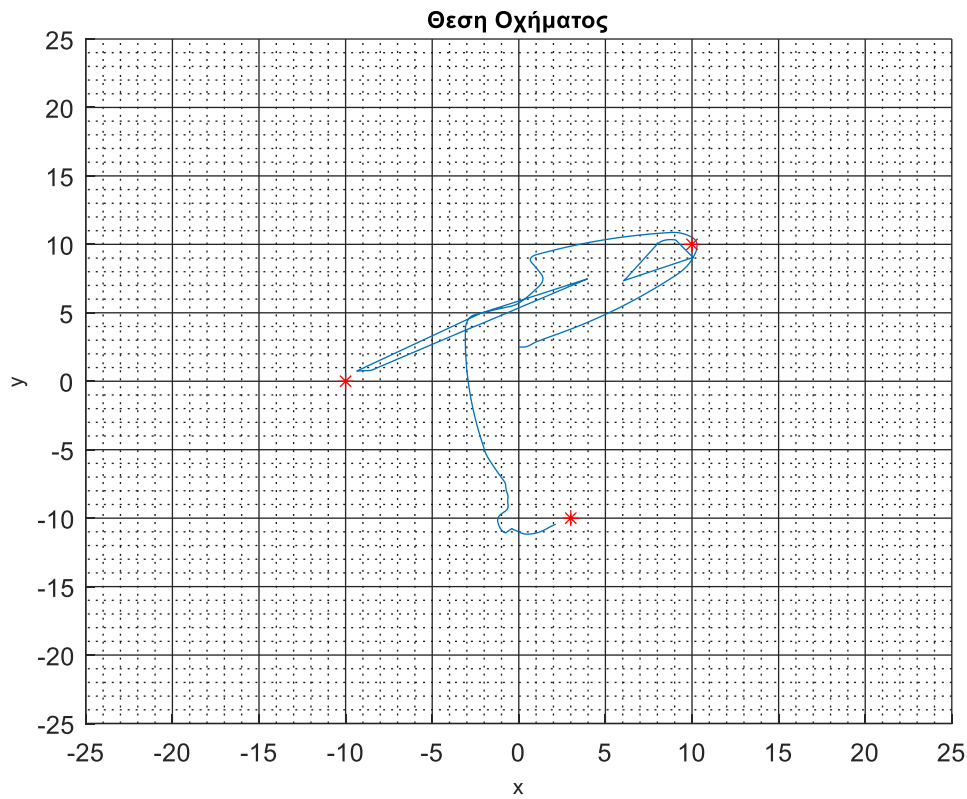
## Αποτελέσματα προσομοίωσης 3<sup>ου</sup> Οχήματος



Εικόνα 167: Δεδομένα Αισθητηρίων 3ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



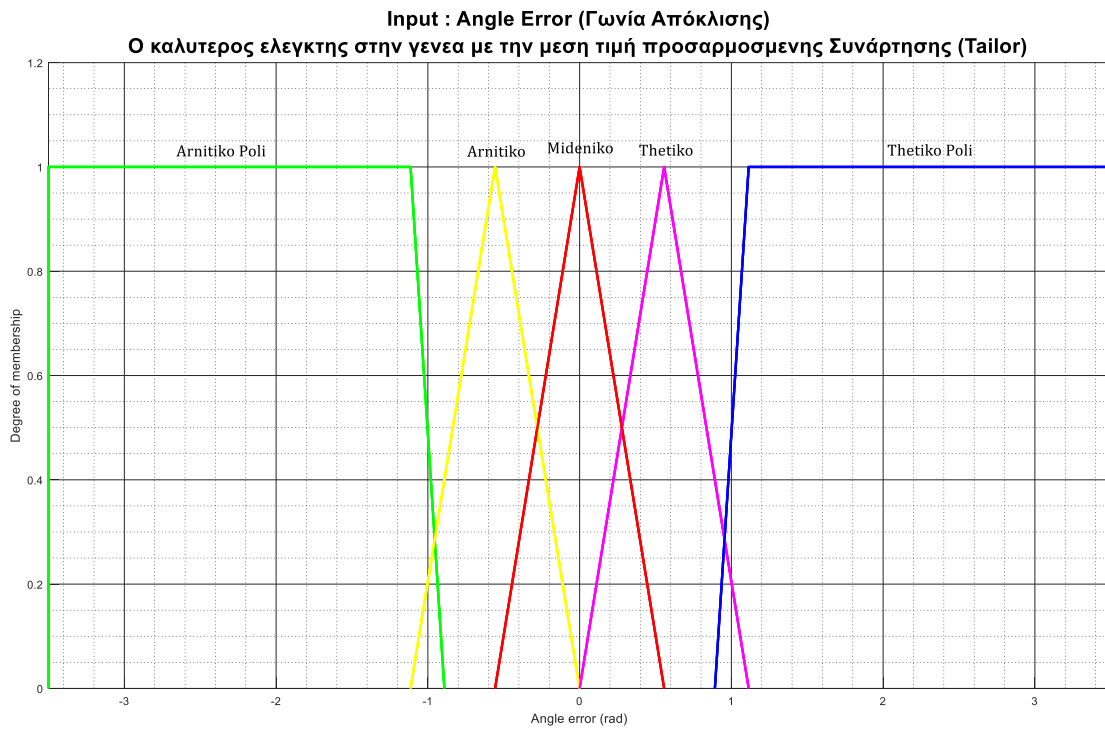
Εικόνα 168 : Δεδομένα Ταχύτητας 3ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



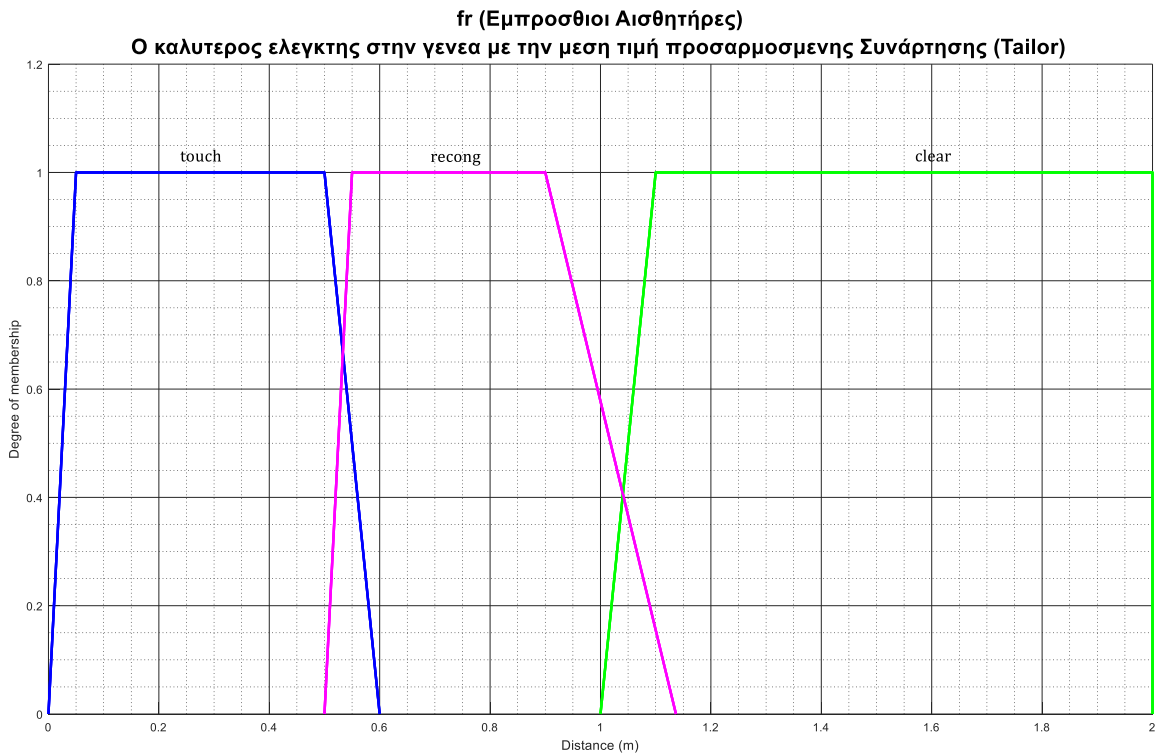
Εικόνα 169 : Διαδρομή 3ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

Στο διάγραμμα των αισθητήρων του 3ου οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Η ταχύτητα του έχει την αναμενόμενη αυξομείωση συνέπεια της ενεργοποίησής των αισθητήρων αλλά διατηρείται σε υψηλή τιμή. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει.

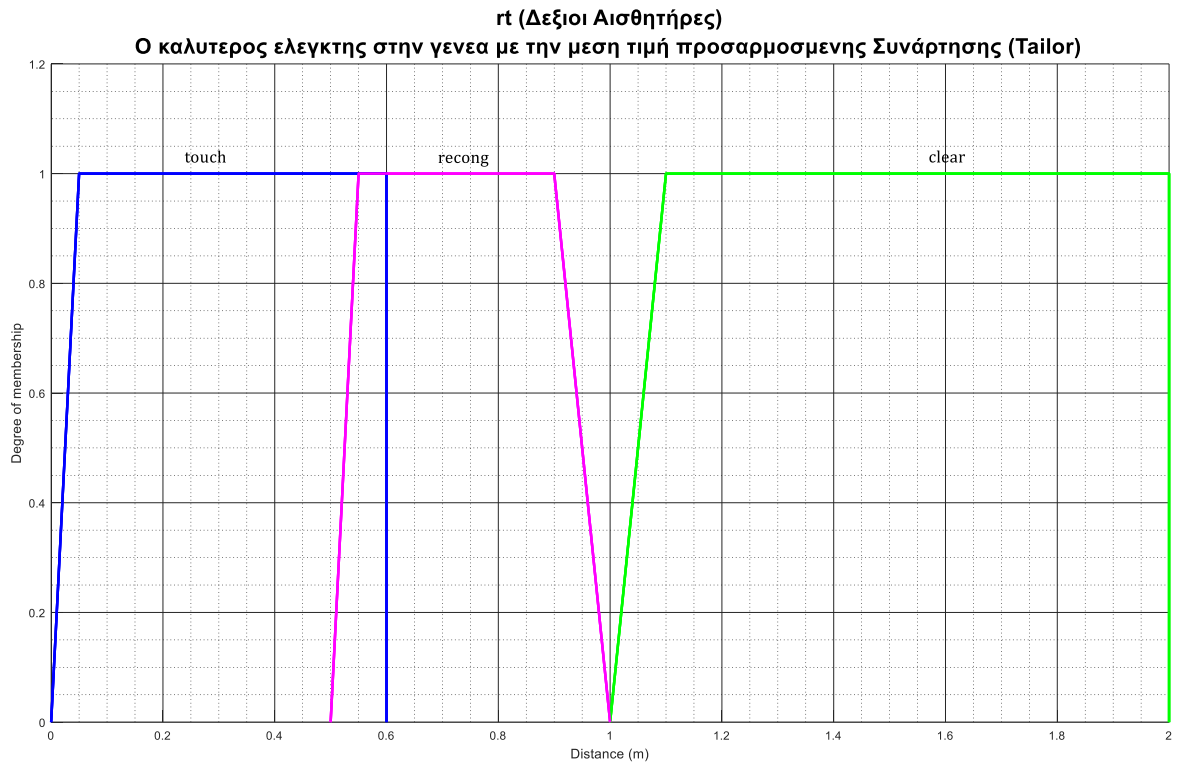
Ελεγκτή : Ο καλύτερος ελεγκτής στη γενεά με την μέση τιμή προσαρμοσμένης συνάρτησης καταλληλότητας



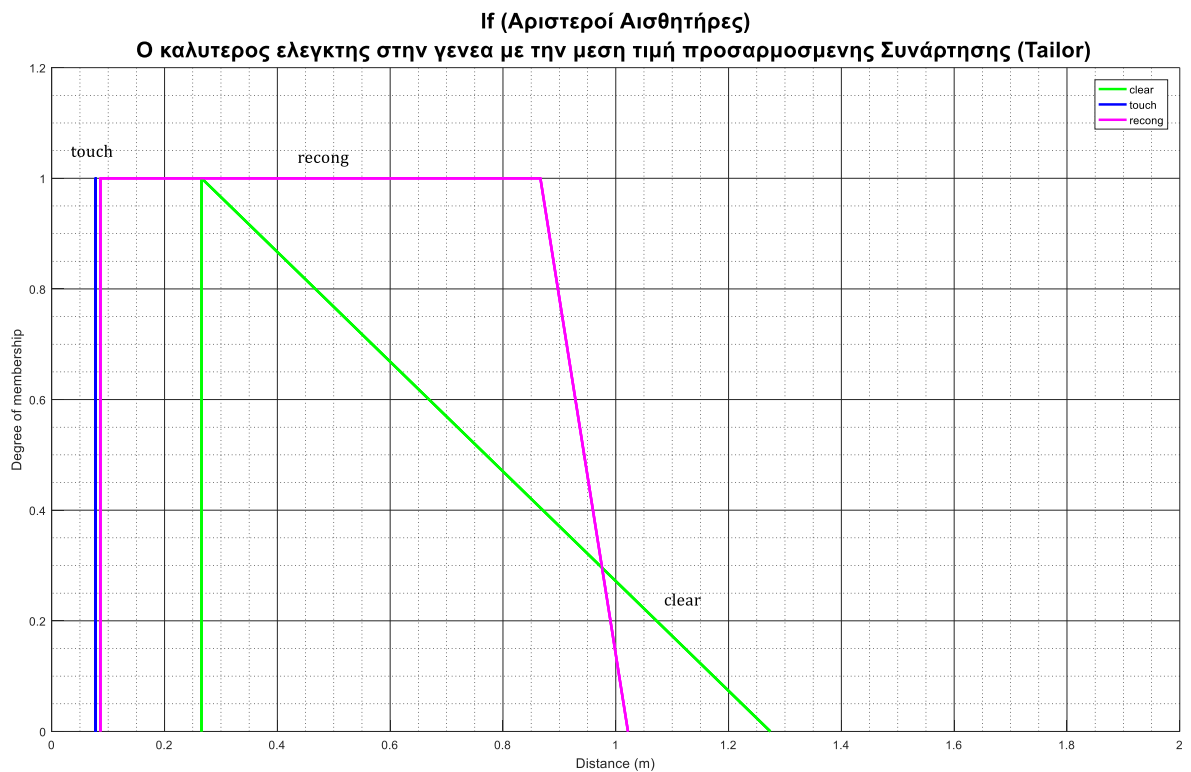
Εικόνα 170 : Είσοδος Γωνία Απόκλισης – Ελεγκτής με την καλύτερη τιμή στην γενεά μέσης τιμής συνάρτησης καταλληλότητας



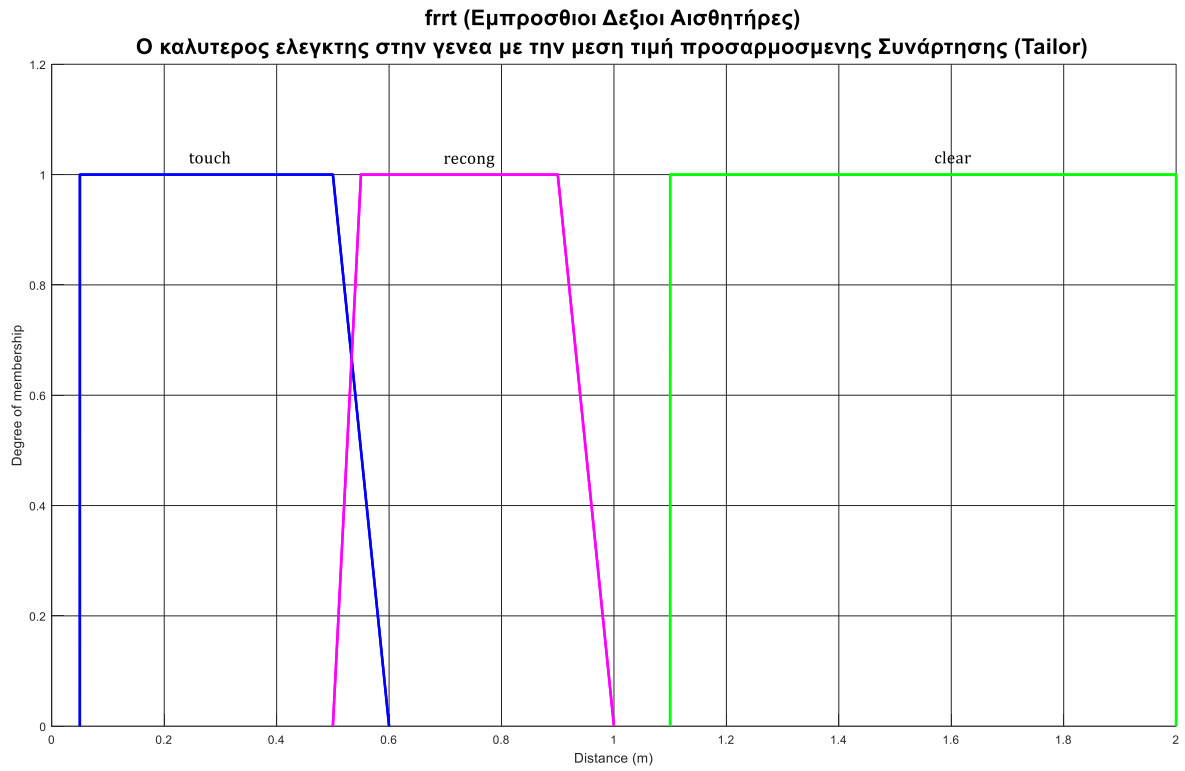
Εικόνα 171: Είσοδος Εμπρόσθιοι Αισθητήρες – Ελεγκτής με την καλύτερη τιμή στην γενεά μέσης τιμής συνάρτησης καταλληλότητας



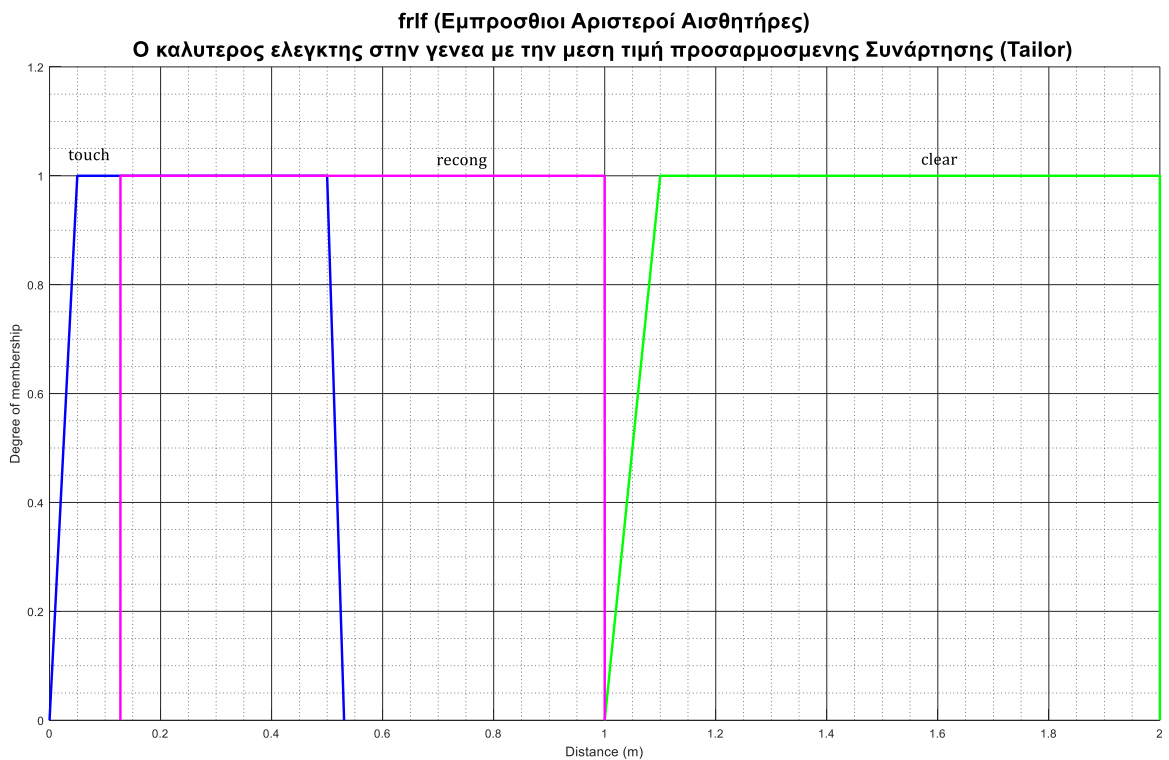
Εικόνα 172: Είσοδος Δεξιοί Αισθητήρες –Ελεγκτής με την καλύτερη τιμή στην γενεά μέσης τιμής συνάρτησης καταλληλότητας



Εικόνα 173: Είσοδος Αριστεροί Αισθητήρες – Ελεγκτής με την καλύτερη τιμή στην γενεά μέσης τιμής συνάρτησης καταλληλότητας



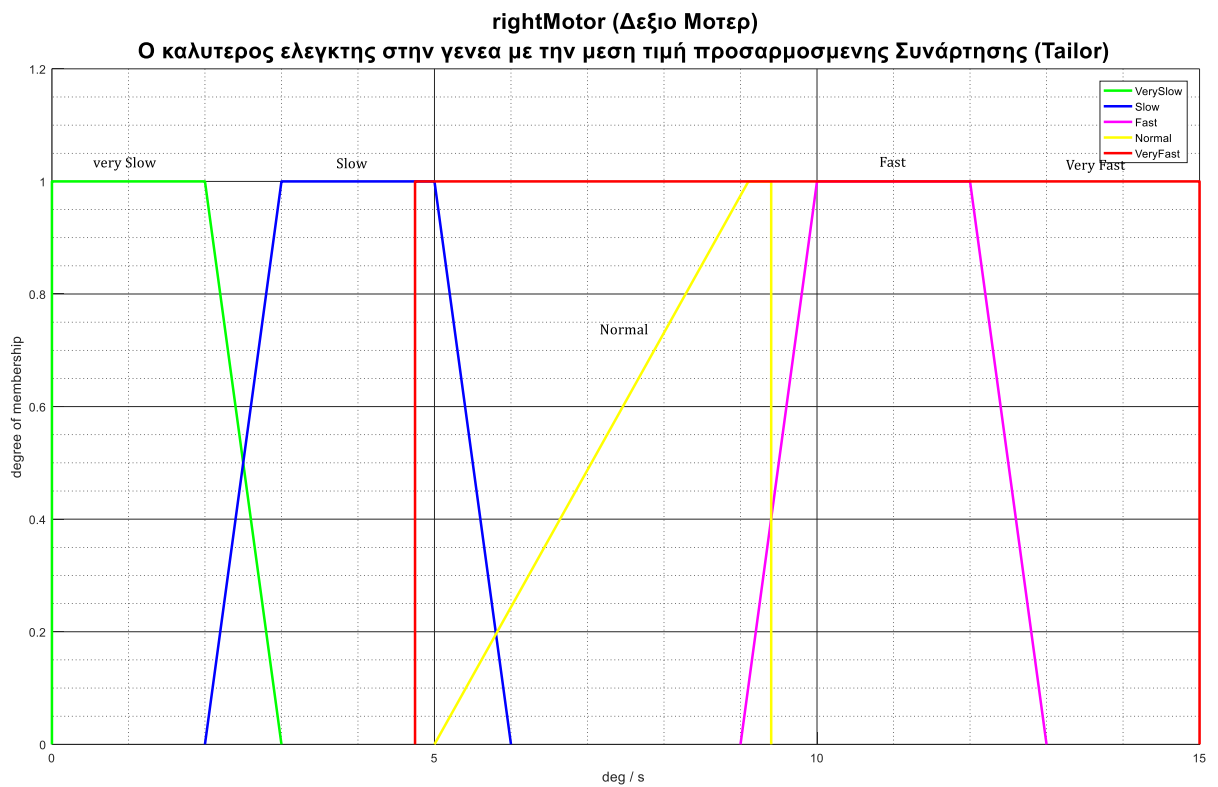
Εικόνα 174: Είσοδος Εμπρόσθιοι Δεξιοί Αισθητήρες – Ελεγκτής με την καλύτερη τιμή στην γενεά μέσης τιμής συνάρτηση καταλληλότητας



Εικόνα 175: Είσοδος Εμπρόσθιοι Αριστεροί Αισθητήρες – Ελεγκτής με την καλύτερη τιμή στην γενεά μέσης τιμής συνάρτηση καταλληλότητας

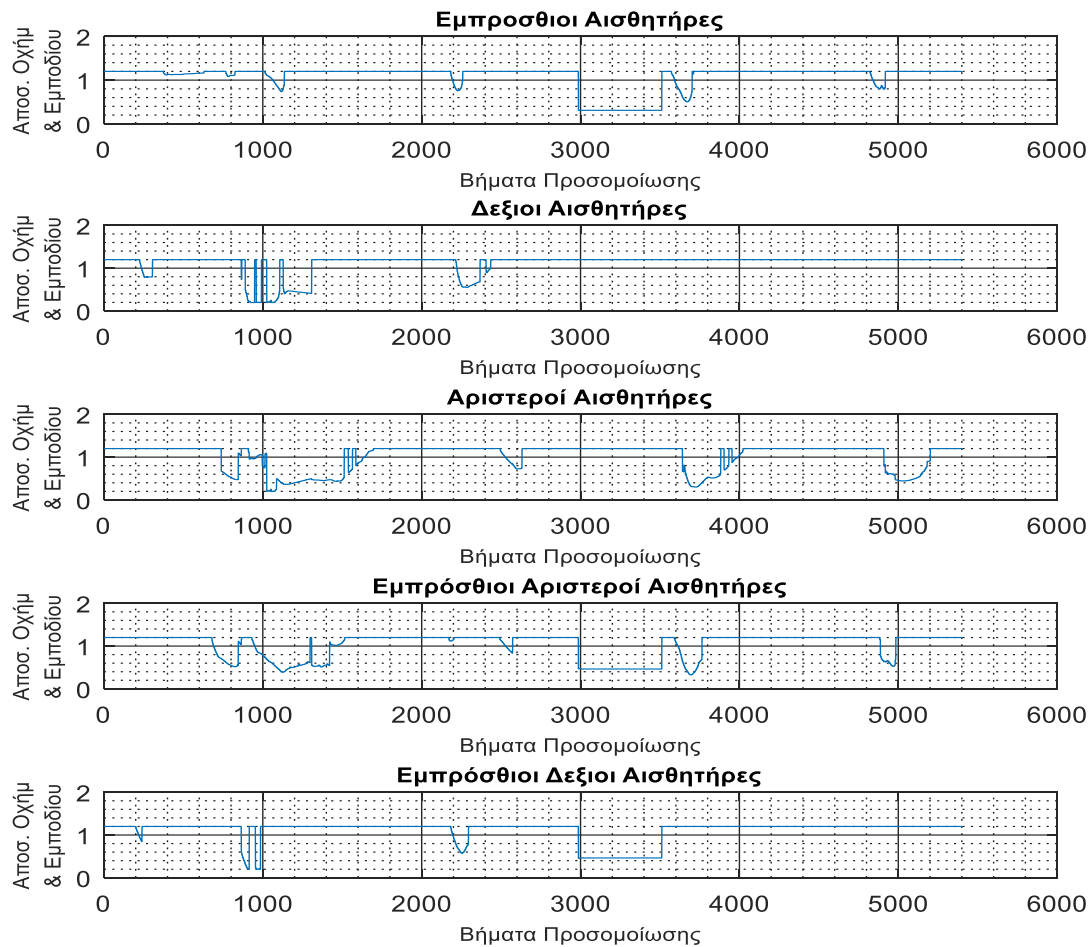


Εικόνα 176 : Έξοδος Ταχύτητα Αριστερού Μοτέρ – Ελεγκτής με την καλύτερη τιμή στην γενεά μέσης τιμής συνάρτηση καταλληλότητας

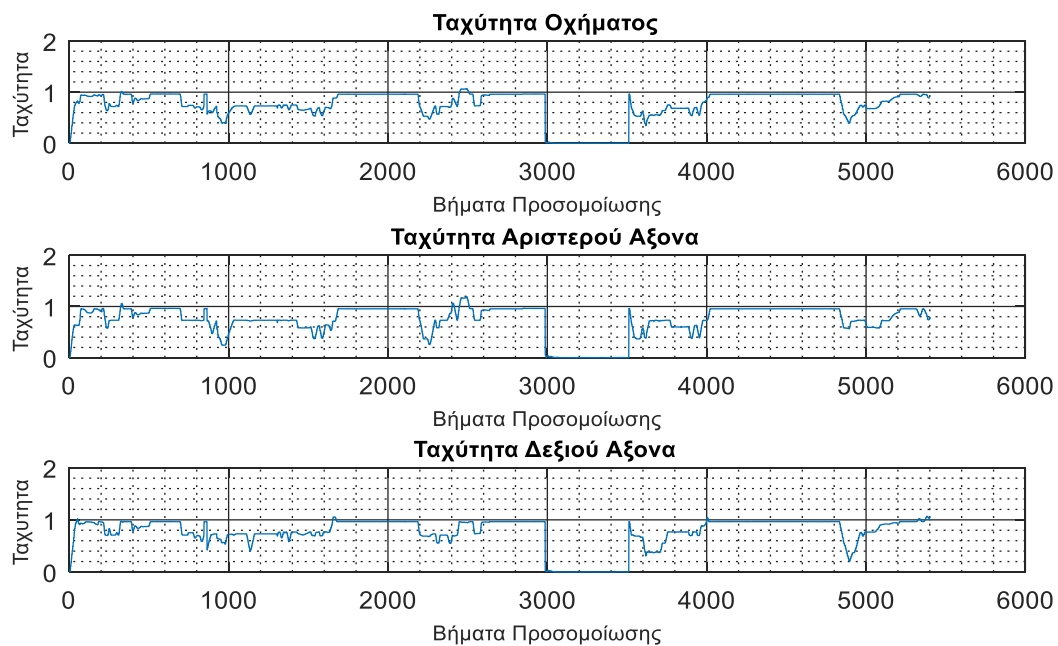


Εικόνα 177: Έξοδος Ταχύτητα Δεξιού Μοτέρ – Ελεγκτής με την καλύτερη τιμή στην γενεά μέσης τιμής συνάρτηση καταλληλότητας

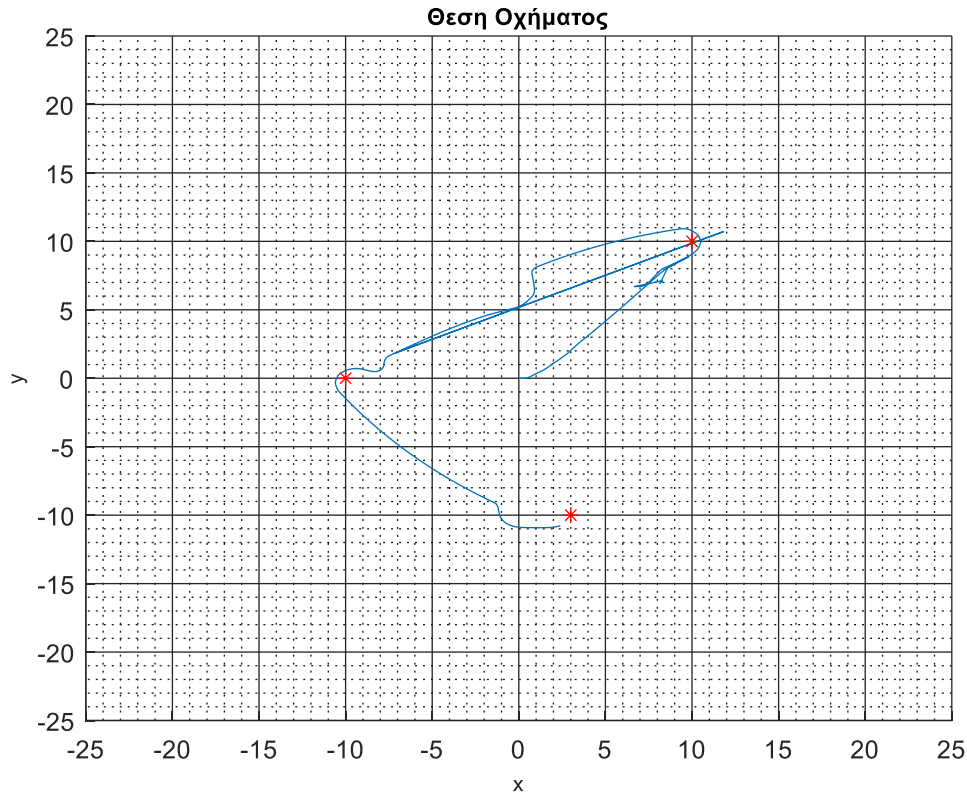
## Αποτελέσματα προσομοίωσης 1<sup>ου</sup> Οχήματος



Εικόνα 178: Δεδομένα Αισθητηρίων 1ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



Εικόνα 179 : Δεδομένα Ταχύτητας 1ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

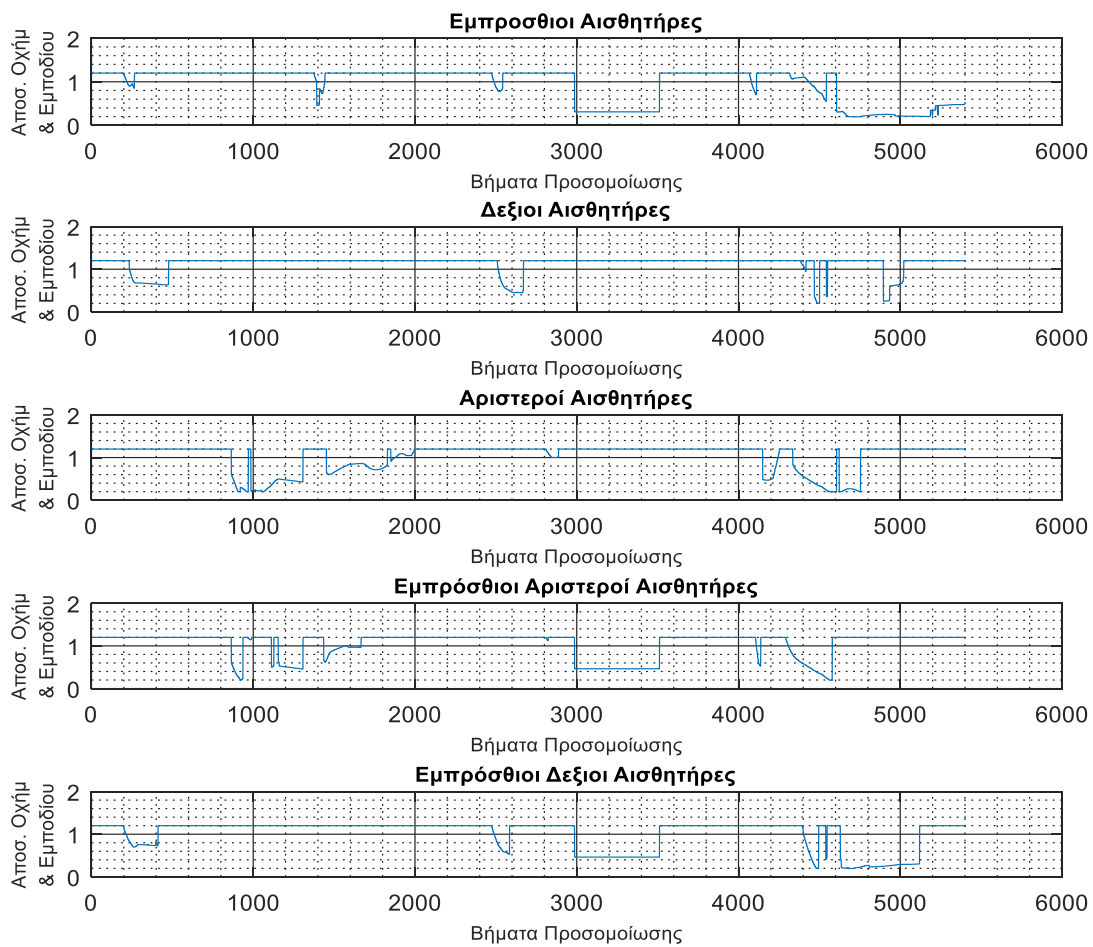


Εικόνα 180 : Διαδρομή 1ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

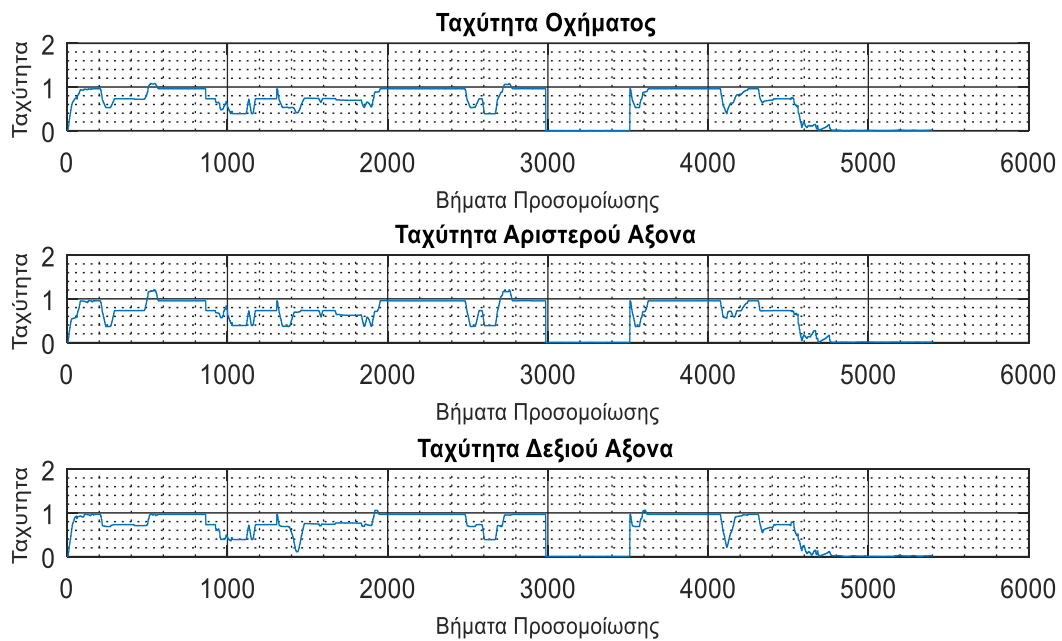
Στο διάγραμμα των αισθητήρων του 1<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Η ταχύτητα του έχει την αναμενόμενη αυξομείωση συνέπεια της ενεργοποίησης των αισθητήρων αλλά διατηρείται σε υψηλή τιμή. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει.



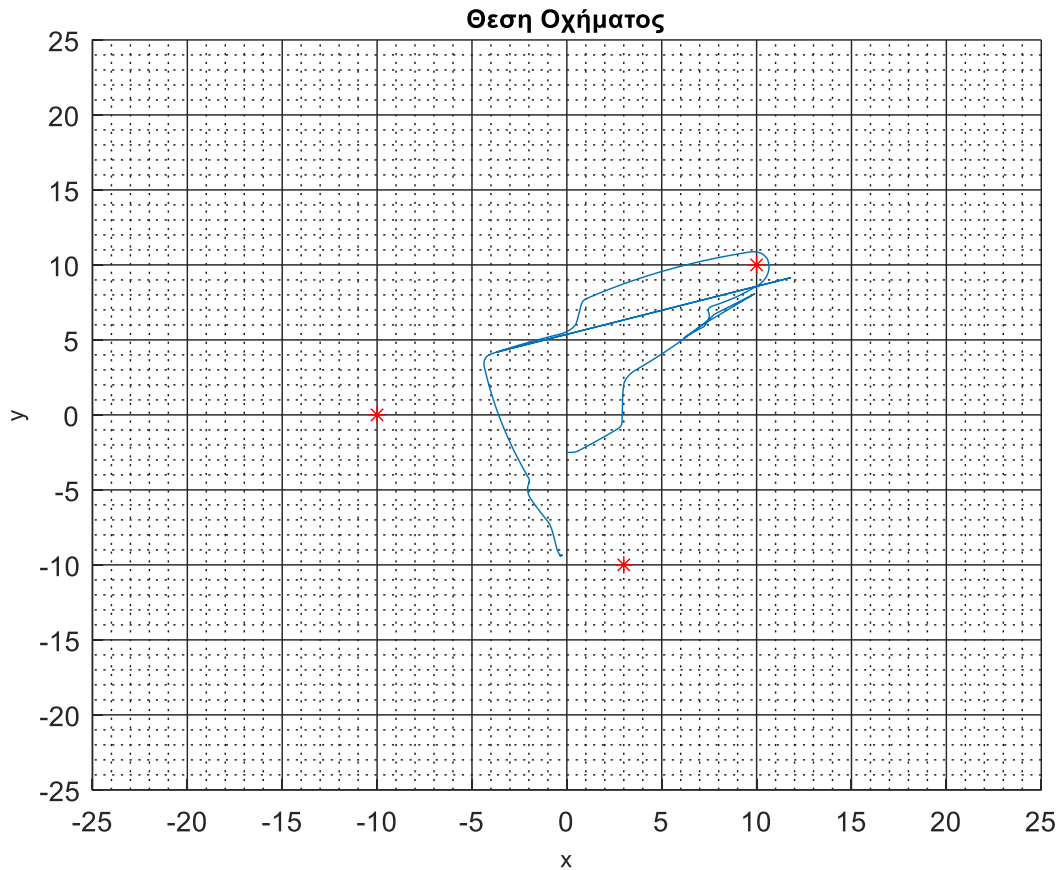
**Αποτελέσματα προσομοίωσης 2<sup>ου</sup> Οχήματος**



Εικόνα 181: Δεδομένα Αισθητηρίων 2ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



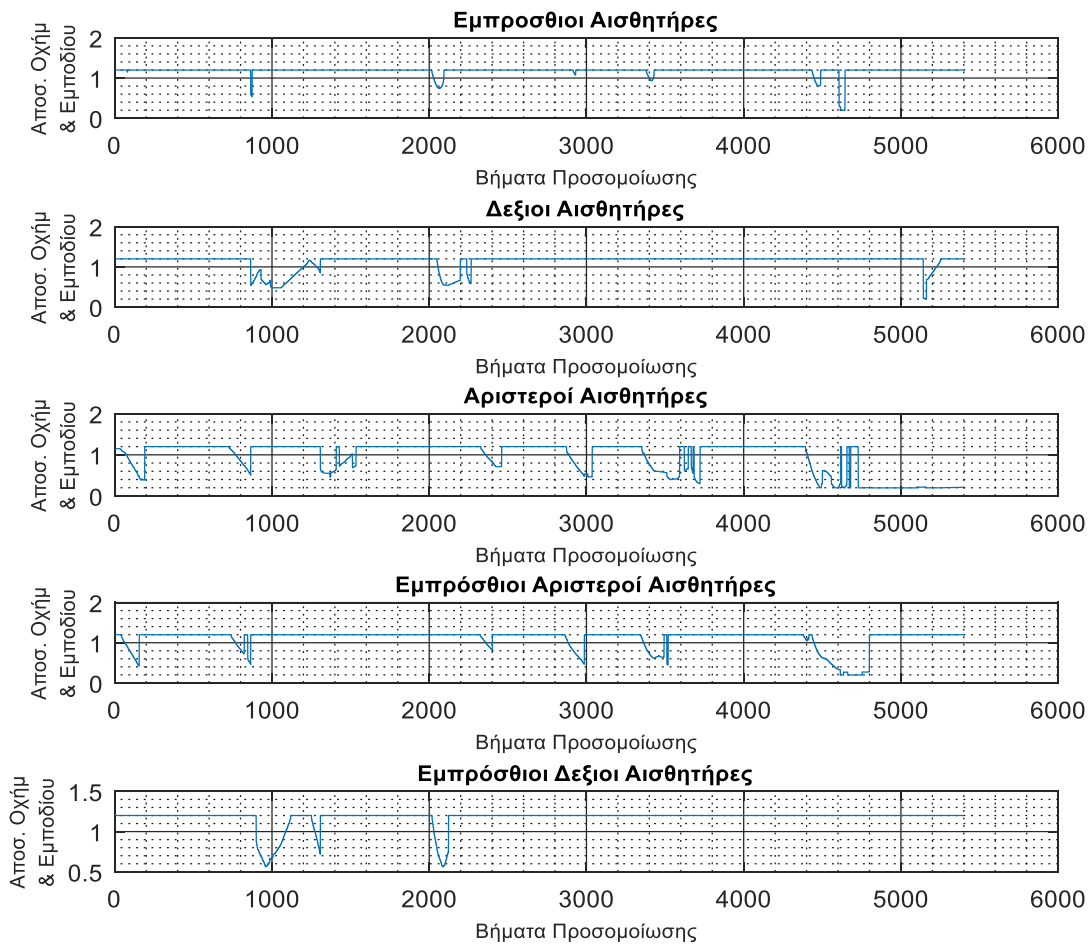
Εικόνα 182 : Δεδομένα Ταχύτητας 2ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



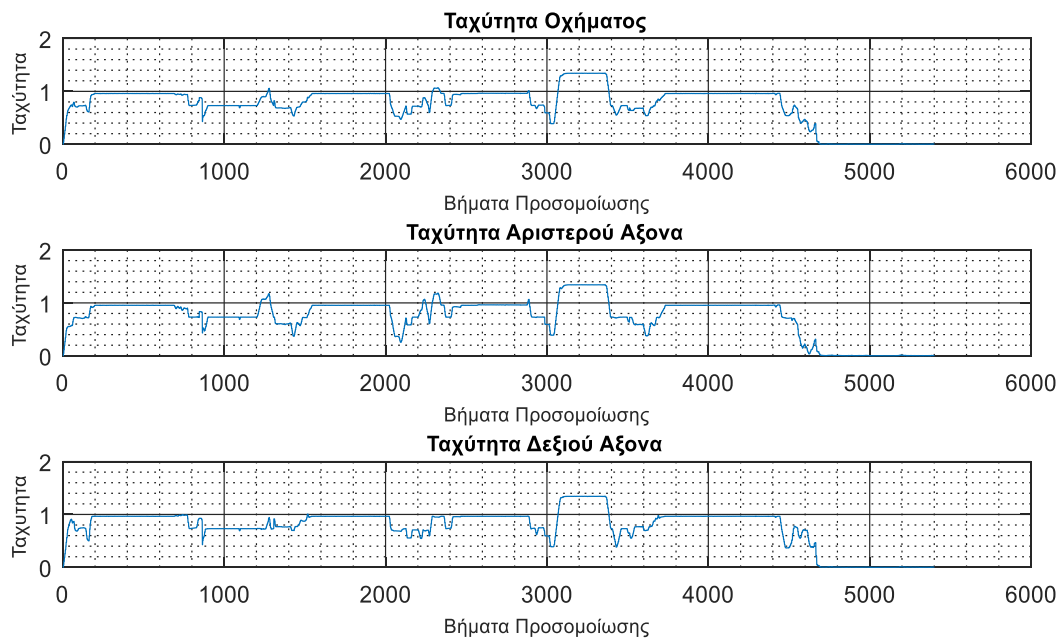
Εικόνα 183 : Διαδρομή 2ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

Στο διάγραμμα των αισθητήρων του 2<sup>ου</sup> οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Η ταχύτητα του έχει την αναμενόμενη αυξομείωση συνέπεια της ενεργοποίησής των αισθητήρων αλλά διατηρείται σε υψηλή τιμή. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει.

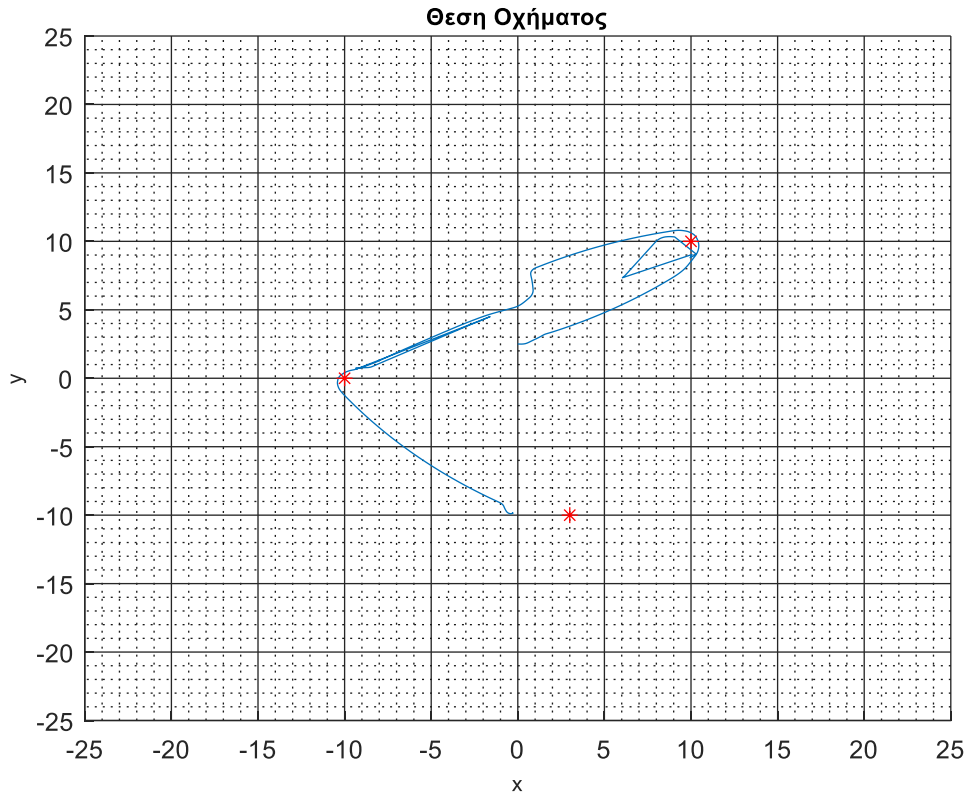
**Αποτελέσματα προσομοίωσης 2<sup>ου</sup> Οχήματος**



Εικόνα 184: Δεδομένα Αισθητηρίων 3ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



Εικόνα 185 : Δεδομένα Ταχύτητας 3ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας



Εικόνα 186 : Διαδρομή 3ου Οχήματος – Ελεγκτής με την καλύτερη τιμή στην προσαρμοσμένη συνάρτηση καταλληλότητας

Στο διάγραμμα των αισθητήρων του 3ου οχήματος παρατηρούμε ότι ενεργοποιούνται οι αισθητήρες και προσπαθεί να αποφύγει την αιτία ενεργοποίησης. Η ταχύτητα του έχει την αναμενόμενη αυξομείωση συνέπεια της ενεργοποίησης των αισθητήρων αλλά διατηρείται σε υψηλή τιμή. Τέλος στο διάγραμμα θέσης παρατηρούμε ότι το όχημα κινείται και για τα 3 σημεία, και καταφέρνει να τα προσεγγίσει.

Στον πίνακα 9 παρουσιάζονται συγκεντρωτικά τα αποτελέσματα των προσομοιώσεων που έγιναν με χρήση των ελεγκτών που εξήχθησαν από την εξελικτική διαδικασία με την χρήση προσαρμοσμένης συνάρτησης καταλληλότητας. Μπορούμε να παρατηρήσουμε ότι τα όχημα κινείται πιο κοντά στα εμπόδια (σταθερά και μη σταθερά) το οποίο έχει άμεση επίδραση στην ταχύτητα που κινούνται τα οχήματα (μείωση) αλλά υπάρχει βελτιστοποίηση στις τιμές του χρόνου ολοκλήρωσης της προσομοίωσης. Παρατηρούμε επίσης ότι και οι δυο ελεγκτές έχουν την ίδια τάση βελτιστοποίησης.

|                 | <i>S</i> | <i>U</i> | <i>Ul</i> | <i>Ur</i> | <i>th</i> | <i>TotalTime</i> |
|-----------------|----------|----------|-----------|-----------|-----------|------------------|
| <i>Original</i> | 1,1029   | 0,8713   | 0,8660    | 0,8760    | 0         | 123,43           |
| <i>min</i>      | 1.045    | 0.737    | 0.726     | 0.748     | 0         | 72.37            |
| <i>mean</i>     | 1.064    | 0.706    | 0.69      | 0.716     | 0         | 90.05            |

|             |         |          |          |          |  |         |
|-------------|---------|----------|----------|----------|--|---------|
| <i>min</i>  | -5,25 % | -15,41 % | -16,17%  | -14,61%  |  | -41.37% |
| <i>mean</i> | -3,66 % | -23,41 % | -25,51 % | -22,35 % |  | -37,07% |

Πίνακας 9 : Σύγκριση αποτελεσμάτων εξέλιξης με προσαρμοσμένη συνάρτηση συμπεριφοράς

Οι τιμές είναι ο μέσος όρος όλων των οχημάτων.

## ΚΕΦΑΛΑΙΟ 5

### Συμπεράσματα – Μελλοντικές Εργασίες

Σκοπός της παρούσας εργασίας ήταν να μελετηθεί η επίδραση της συνάρτησης καταλληλότητας στην εξελικτική διαδικασία με γενετικούς αλγόριθμους σε ασαφής ελεγκτή ενός τροχήλατου ρομποτικού οχήματος όταν αυτό λειτουργεί σε ένα περιβάλλον με εμπόδια και είναι ομάδα με άλλα ρομποτικά οχήματα.

Από τα αποτελέσματα των προσομοιώσεων μπορούμε να παρατηρήσουμε ότι καλύτερη βελτιστοποίηση παρουσίασε η εξελικτική διαδικασία με την χρήση συνάρτησης καταλληλότητας συμπεριφοράς. Βέβαια με πολύ μικρή διαφορά σε σχέση με την διαδικασία που έγινε χρήση της προσαρμοσμένης συνάρτησης καταλληλότητας. Βέβαια ποιο ολοκληρωμένη είναι η διαδικασία με την προσαρμοσμένη συνάρτηση καταλληλότητας διότι αυτή μας εξασφαλίζει ότι έχει ολοκληρωθεί η αποστολή. Η εξελικτική διαδικασία με την ολοκληρωτική συνάρτηση καταλληλότητας δεν παρουσίασε βελτιστοποίηση παρά μόνο στον χρόνο ολοκλήρωσης της προσομοίωσης. Η συναρτήσεις συμμετοχής στο ασαφή ελεγκτή δεν είχαν καμία μεταβολή σε σχέση με τον αρχικό.

Βέβαια πρέπει να τονίσουμε ότι ο αριθμός των γενεών σε όλες τις εξελικτικές διαδικασίες είναι μικρός. Αυτό οφείλεται στην αδυναμία του συστήματος που έγινε η μελέτη, αλλά και στον παράγοντα της μεταβλητότητας που είχε η προσομοίωση με τα μη σταθερά εμπόδια (κινούμενα ρομπότ).

|                       | <i>S</i> | <i>U</i> | <i>Ul</i> | <i>Ur</i> |   | <i>TotalTime</i> |
|-----------------------|----------|----------|-----------|-----------|---|------------------|
| <i>Original</i>       | 1.1584   | 0.8514   | 0.8287    | 0.8741    |   | 84.2192          |
| <i>Behavioral min</i> | 1.127    | 0.843    | 0.832     | 0.8539    |   | 73.042           |
| <i>Tailor min</i>     | 1.045    | 0.737    | 0.726     | 0.748     | 0 | 72.37            |
| <i>Aggregate Mean</i> | 1.154    | 0.706    | 0.693     | 0.718     | 0 | 84.40            |

Πίνακας 10 : Σύγκριση αποτελεσμάτων εξέλιξης

Συγκεκριμένα με χρήση των νέων ελεγκτών το όχημα είχε τη δυνατότητα να αποφεύγει τα εμπόδια σε μικρότερη απόσταση και αυτό του έδινε το πλεονέκτημα να διανύει μικρότερη διαδρομή, με αποτέλεσμα να φτάνει στο στόχο πιο γρήγορα, δεν έστριβε τόσο απότομα και προσπαθούσε να κάνει πορεία όσο το δυνατόν ευθύγραμμη ώστε να μπορεί να αναπτύξει μεγαλύτερη ταχύτητα.

Σε μελλοντική εργασία θα πρέπει να γίνει προσπάθεια να αυξηθεί ο αριθμός των γενεών. Με παράλληλη βελτιστοποίηση του κώδικα στο λογισμικό MatLab ώστε να μειωθεί ο συνολικός χρόνος της εργασίας που αθροιστικά είναι πολύ μεγάλος. Χρόνος γενεάς (μέση τιμή) :  $117 \times 2 = 234 \text{ min}$  ή 3:54:00.

Επίσης στην παρούσα εργασία η ομάδα των οχημάτων δεν είχε ρόλους, η μονη συνεργασία ήταν ένα σήμα ότι κάποιο από τα οχήματα προσέγγισε το στόχο ώστε να κινηθούν προς τον επόμενο.

## Βιβλιογραφία

- [1] NAVIGATION OF MOBILE ROBOTS USING A FUZZY LOGIC CONTROLLER, DAYAL R. PARHI, 2004
- [2] Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment. Mohammed Faisal, Ramdane Hedjar, Mansour Al Sulaiman and Khalid Al-Mutib, 2012
- [3] Optimal Design of the Fuzzy Navigation System for a Mobile Robot Using Evolutionary Algorithms, Abraham Meléndez, Oscar Castillo, Fevrier Valdez, Jose Soria and Mario Garcia, 2012.
- [4] Behavior-Based Fuzzy Control for Mobile Robot Navigation, Hongwei Mo, Qirong Tang and Longlong Meng, 2013
- [5] Sensor Fusion Based Model for Collision Free Mobile Robot Navigation, Marwah Almasri, Khaled Elleithy and Abrar Alajlan, 2015
- [6] Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation, Hajer Omrane, Mohamed Slim Masmoudi, and Mohamed Masmoudi, 2016
- [7] Modeling and Simulation of a Tracked Mobile Inspection Robot in MATLAB and V-REP Software, Michał Ciszewski, Łukasz Mitka, Tomasz Buratowski, Mariusz Giergiel, 2017
- [8] Fitness functions in evolutionary robotics: A survey and analysis, Andrew L. Nelson a, Gregory J. Barlow b, Lefteris Doitsidis c, 2007
- [9] Common tasks in Evolutionary Robotics, an overview, Corn\_e Sprong, 2011.
- [10] Ευφυής Έλεγχος, Θεωρία και Εφαρμογές, Δρ. Βολογιαννίδης Σταύρος, (svol@teiser.gr), 2009.
- [11] Fuzzy Inference System Modeling, Fuzzy Logic Toolbox, Matlab Help Documentetion, 2016.
- [12] J. R. Koza, Genetic Programming: on the programming of computers by means of natural selection: The MIT Press, 1992.
- [13] I. Rechenberg, "Cybernetic solution path of an experimental problem," Royal Aircraft Establishment (UK), Ministry of Aviation, Farnborough 1965.
- [14] I. Rechenberg, Evolutionsstrategie; Optimierung technischer Systeme nach Prinzipien der biologischen Evolution: Frommann-Holzboog, 1973.
- [15] H.-P. Schwefel, Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: Mit einer vergleichenden Einführung in die Hill-Climbing und Zufallsstrategie: Birkhauser; 1. Aufl edition, 1977.
- [16] L. J. Fogel, A. J. Owens, and M. J. Walsh, Artificial Intelligence Through Simulated Evolution. New York: Wiley Publishing, 1966.
- [17] L. J. Fogel, Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming: Wiley-Interscience 1999.
- [18] A. E. Eiben and C. H. M. van Kemende, "Diagonal crossover in genetic algorithms for numerical optimization," Control and Cybernetics, vol. 26, pp. 447-65, 1997.

- 
- [19] P. Rahila and M. M. Raghuwanshi, "Multi-Objective Optimization Using Multi Parent Crossover Operators," *Emerging Trends in Computing and Information Sciences*, vol. 2, pp. 99-105 2011.
- [20] H. Muhlenbein, "How genetic algorithms really work: I. Mutation and hillclimbing," in *Proceedings of the Second Conference on Parallel Problem Solving from nature*, Brussels 1992, pp. 15-26.
- [21] T. Back, "Selective pressure in evolutionary algorithms: a characterization of selection mechanisms," in *Evolutionary Computation*, 1994. IEEE World Congress on Computational Intelligence., *Proceedings of the First IEEE Conference on*, 1994, pp. 57-62 vol.1.
- [22] B. A. Julstrom, "It's all the same to me: revisiting rank-based probabilities and tournaments," in *Evolutionary Computation*, 1999. CEC 99. *Proceedings of the 1999 Congress on*, 1999, p. 1505 Vol. 2.
- [23] I. Harvey, "Artificial Evolution: A Continuing SAGA," in *Evolutionary robotics : from intelligent robotics to artificial life ER 2001 : evolutionary robotics. International symposium*, , Tokyo , JAPAN,, 2001, pp. 94-109.
- [24] P. J. Hancock, "An Empirical Comparison of Selection Methods in Evolutionary Algorithms," *Selected Papers From AISB Workshop on Evolutionary Computing*, SpringerVerlag, pp. 865: 80-94., 1994.
- [25] S. Gupta, "Relative Fitness Scaling for Improving Efficiency of Proportionate Selection in Genetic Algorithms," presented at the GECCO Genetic and Evolutionary Computation Conference, Montreal Canada, 2009.
- [26] S. Legg, M. Hutter, and A. Kumar, "Tournament versus fitness uniform selection," in *Evolutionary Computation*, CEC2004. *Congress on*, 2004, pp. 2144-2151 Vol.2.
- [27] M. Hutter, "Fitness uniform selection to preserve genetic diversity," in *Evolutionary Computation*, CEC '02. *Proceedings of the Congress on*, 2002, pp. 783-788.
- [28] D. Whitley, S. Rana, and R. Heckendorn, "Island model genetic algorithms and linearly separable problems," in *Evolutionary Computing*, ed, 1997, pp. 109-125.
- [29] *Evolution of Robotic Controllers Using Genetic Algorithms*, Mark Beckerleg, A thesis submitted to AUT University, School of Engineering, 2012.
- [30] D. Floreano, F. Mondada, Evolution of homing navigation in a real mobile robot, *IEEE Transactions on Systems, Man, Cybernetics Part B: Cybernetics* 26 (3) (1996) 396\_407.
- [31] H.H. Lund, O. Miglino, From simulated to real robots, in: *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 362\_365.
- [32] W. Banzhaf, P. Nordin, M. Olmer, Generating adaptive behavior using function regression within genetic programming and a real robot, in: *Proceedings of the Second International Conference on Genetic Programming*, San Francisco, 1997, pp. 35\_43.
- [33] N. Jakobi, Running across the reality gap: Octopod locomotion evolved in a minimal simulation, in: P. Husbands, J.A. Meyer (Eds.), *Evolutionary Robotics: First European Workshop, EvoRobot98*, Springer-Verlag, 1998, pp. 39\_58.
- [34] T. Gomi, K. Ide, Evolution of gaits of a legged robot, in: *The 1998 IEEE International Conference on Fuzzy Systems*, *Proceedings of the 1998 IEEE World Congress on Computational Intelligence*, vol. 1, 4\_9 May, 1998, pp. 159\_164.

- [35] V. Matellán, C. Fernández, J.M. Molina, Genetic learning of fuzzy reactive controllers, *Robotics and Autonomous Systems* 25 (1\_2) (1998) 33\_41.
- [36] P. Nordin, W. Banzhaf, M. Brameier, Evolution of a world model for a miniature robot using genetic programming, *Robotics and Autonomous Systems* 25 (1\_2) (1998) 105\_116.
- [37] J. Liu, C.K. Pok, H.K. Keung, Learning coordinated maneuvers in complex environments: A sumo experiment, in: *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99*, vol. 1, 6\_9 July, 1999, pp. 343\_349.
- [38] H.-S. Seok, K.-J. Lee, J.-G. Joung, B.-T. Zhang, An on-line learning method for object-locating robots using genetic programming on evolvable hardware, in: *Proceedings of the Fifth International Symposium on Artificial Life and Robotics, AROB'00*, vol. 1, 2000, pp. 321\_324.
- [39] F. Hoffmann, G. Pfister, Evolutionary learning of a fuzzy control rule base for an autonomous vehicle, in: *Proceedings of the Fifth International Conference IPMU: Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain, July 1996*, pp. 659\_664.
- [40] A. Thompson, Evolving electronic robot controllers that exploit hardware resources, *Advances in Artificial Life: Proceedings of the 3rd European Conference on Artificial Life, ECAL95, Lausanne*, vol. 929, Springer-Verlag, 1995, pp. 640\_656.
- [41] D. Keymeulen, M. Iwata, Y. Kuniyoshi, T. Higuchi, Online evolution for a selfadapting robotic navigation system using evolvable hardware, *Artificial Life4* (4) (1998) 359\_393.
- [42] A. Ishiguro, S. Tokura, T. Kondo, Y. Uchikawa, Reduction of the gap between simulated and real environments in evolutionary robotics: A dynamically rearranging neural network approach, in: *Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, 1999, pp. 239\_244.
- [43] M. Ebner, A. Zell, Evolving a behavior-based control architecture \_ From simulations to the real world, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 2, 13\_17 July 1999, pp. 1009\_1014.
- [44] D. Keymeulen, M. Iwata, Y. Kuniyoshi, T. Higuchi, Online evolution for a selfadaptingrobotic navigation system using evolvable hardware, *Artificial Life 4* (4) (1998) 359\_393.
- [45] I.G. Sprinkhuizen-Kuyper, R. Kortmann, E.O. Postma, Fitness functions for evolving box-pushing behaviour, in: A. van den Bosch, H. Weigand (Eds.), *Proceedings of the Twelfth Belgium\_Netherlands Artificial Intelligence Conference, 2000*, pp. 275\_282.
- [46] U. Nehmzow, Physically embedded genetic algorithm learning in multi-robot scenarios: The PEGA algorithm, in: *Proceedings of the Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, Edinburgh, 2002*.
- [47] R.A. Watson, S.G. Ficici, J.B. Pollack, Embodied evolution: Distributing an evolutionary algorithm in a population of robots, *Robotics and Autonomous Systems* 39 (1) (2002) 1\_18.
- [48] M. Quinn, L. Smith, G. Mayley, P. Husbands, Evolving team behaviour for real robots, in: *EPSRC/BBSRC International Workshop on Biologically-Inspired Robotics: The Legacy of W. Grey Walter, WGW'02, 14\_16 Aug., 2002, HP Bristol Labs, UK*.
- [49] A.L. Nelson, E. Grant, J.M. Galeotti, S. Rhody, Maze exploration behaviors using an integrated evolutionary robotics environment, *Journal of Robotics and Autonomous Systems* 46 (3) (2004) 159\_173.

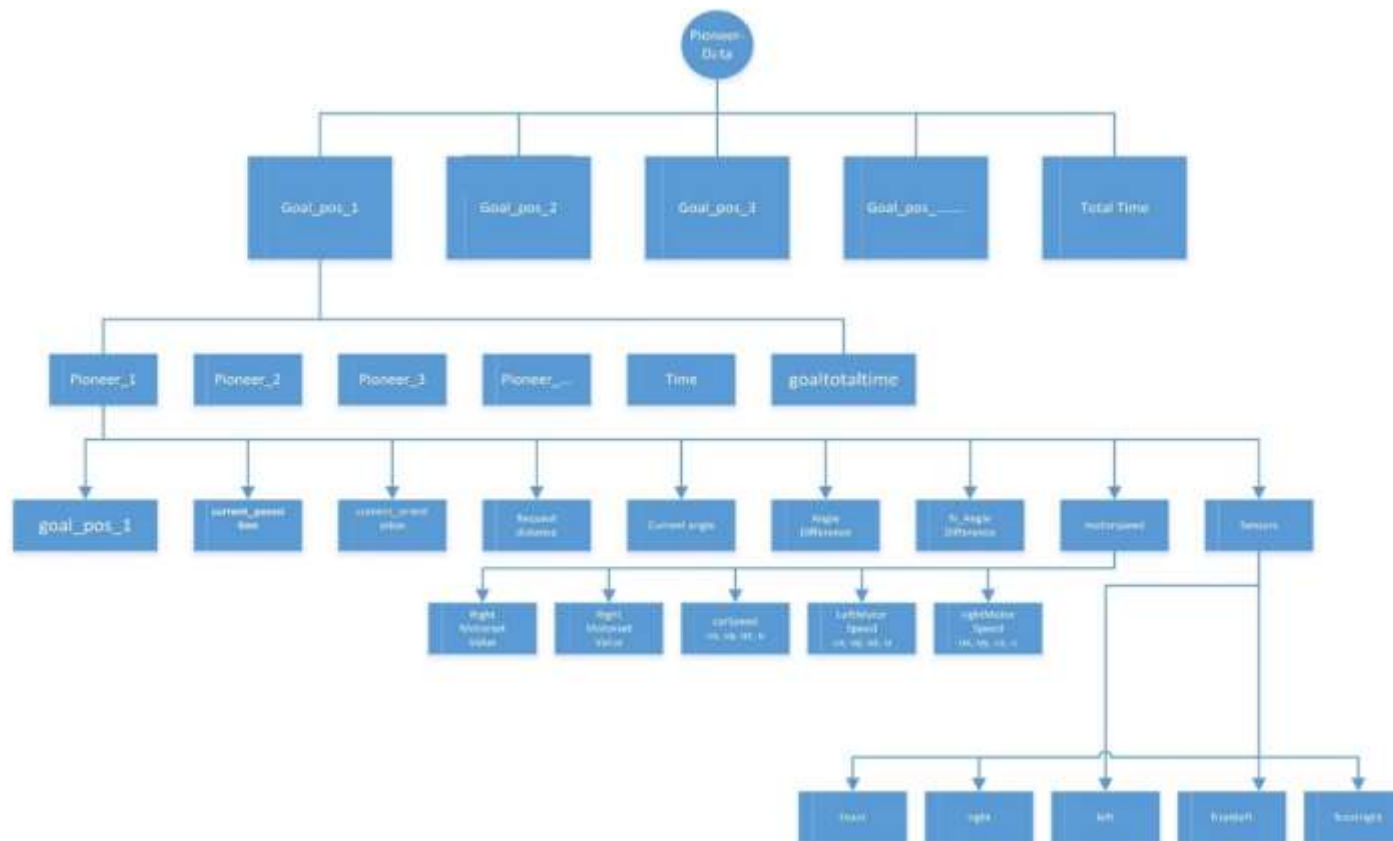


- [50] G. Capi, K. Doya, Evolution of recurrent neural controllers using an extended parallel genetic algorithm, *Robotics and Autonomous Systems* 52 (2\_3) (2005) 148\_159.
- [51] G.B. Parker, R. Georgescu, Using cyclic genetic algorithms to evolve multiloop control programs, in: *Proceedings of the 2005 IEEE International Conference on Mechatronics and Automation, ICMA 2005*, July 2005, Niagara Falls, Ontario, Canada
- [52] G.S. Hornby, S. Takamura, J. Yokono, O. Hanagata, M. Fujita, J. Pollack, Evolution of controllers from a high-level simulator to a high dof robot, in: J. Miller (Ed.), *Evolvable Systems: From Biology to Hardware; Proceedings of the Third International Conference, ICES 2000*, in: *Lecture Notes in Computer Science*, vol. 1801, Springer, 2000, pp. 80\_89.
- [53] E.J.P. Earon, T.D. Barfoot, G.M.T. D'Eleuterio, From the sea to the sidewalk: the evolution of hexapod walking gaits by a genetic algorithm, in: *Proceedings of the International Conference on Evolvable Systems, ICES, Edinburgh, Scotland, 17\_19 April, 2000*.
- [54] F. Hoffmann, J.C.S. Zagal Montealegre, Evolution of a tactile wall-following behavior in real time, in: *The 6th Online World Conference on Soft Computing in Industrial Applications, WSC6, 10\_24 Sept., 2001*.
- [55] P. Augustsson, K. Wolff, P. Nordin, Creation of a learning, flying robot by means of evolution, in: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*, New York, 9\_13 July, 2002, Morgan Kaufmann, 2002, pp. 1279\_1285.
- [56] J. Zufferey, D. Floreano, M. van Leeuwen, T. Merenda, Evolving vision based flying robot, in: Bülthoff, Lee, oggio, Wallraven (Eds.), *Proceedings of the 2nd International Workshop on Biologically Motivated Computer Vision*, in: *LNCS*, vol. 2525, Springer-Verlag, Berlin, pp. 592\_600.
- [57] I. Macinnes, E. Di Paolo, Crawling out of the simulation: Evolving real robot morphologies using cheap, reusable modules, in: *Proceedings of the International Conference on Artificial Life, ALIFE9, Boston, MA, 12\_15 Sept., MIT Press, 2004*, pp. 94\_99.
- [58] S. Chernova, M. Veloso, An evolutionary approach to gait learning for fourlegged robots, in: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems, IROS'04*, vol. 3, Sendai, Japan, Sept. 28\_Oct. 2, 2004, pp. 2562\_2567
- [59] *Fuzzy Control*, Kevin M. Passino, Stephen Yurkovich, 1998 Addison Wesley Longman, Inc.
- [60] *Evolution of Robotic Controllers Using Genetic Algorithms*, Mark Beckerleg, A thesis submitted to AUT University, School of Engineering in fulfilment of the requirements for the degree of Doctor of Philosophy (PhD), March 13, 2012.
- [61] Ε. Δοϊτσίδης, “Μεθοδολογία Μοντελοποίησης και Βελτιστοποίησης Υπολογιστικής Νοημοσύνης Ομάδας Αυτόνομων Οχημάτων”, Διδακτορική Διατριβή, Τμήμα Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης, 2008.
- [62] Θ Κοντοζούδης, «Βελτιστοποίηση ελεγκτών ρομποτικών οχημάτων με τη μέθοδο Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization)», ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ, Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών (ΔΠΜΣ) «Προηγμένα Συστήματα Παραγωγής, Αυτοματισμού και Ρομποτικής», ΑΤΕΙ Κρήτης
- [63] Δρ. Μ. Καββουσάνος, “Κινηματική ανάλυση τροχοφόρων ρομπότ”, Σημειώσεις μαθήματος: Προηγμένα Ρομποτικά Συστήματα, Διατμηματικό Μεταπτυχιακό Πρόγραμμα Προηγμένα Συστήματα Παραγωγής, Αυτοματισμού και Ρομποτικής, ΤΕΙ Κρήτης, 2016.
- [64] *Virtual Robot Experimentation Platform (V-rep) USER MANUAL Version 3.5*.

[65] Βελτιστοποίηση Προγραμματισμού παραγωγής με χρήση γενετικών αλγόριθμων, Νικόλαος Μιχ. Ουλκέρογλου, Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π., 2004.

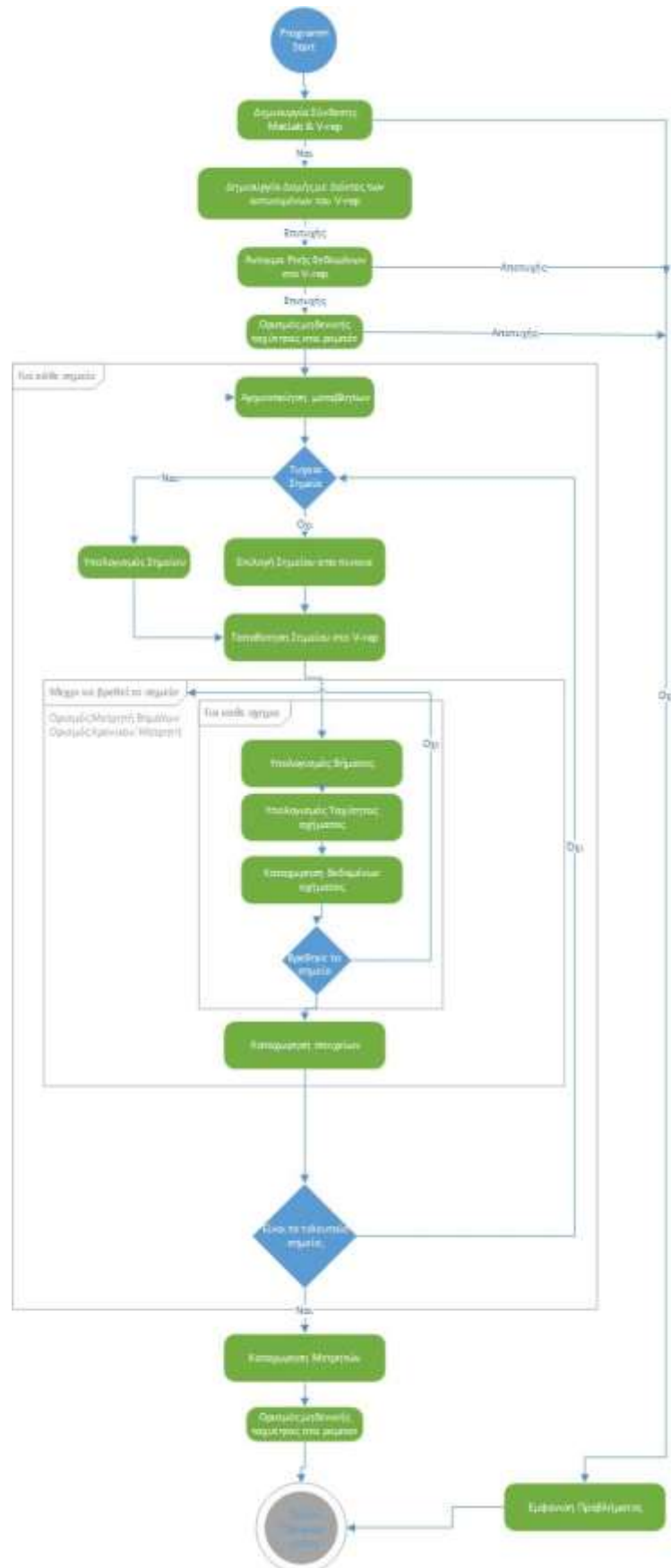
## ΠΑΡΑΡΤΗΜΑ

### Α. Δομή Αποθήκευσης Δεδομένων Προσομοίωσης



- 
- Η δομή όπου καταχωρούνται όλα τα δεδομένα της προσομοίωσης, ώστε να μπορούμε να υπολογίσουμε τα στοιχεία που επιθυμούμε.

## Β. Διάγραμμα Ροής Robot Simulation



---

## Γ. Δομή των ρουτινών Robot Simulation

1. RGSMoveToPoint\_v1
  - 1.1 CreateObjectHandles\_v1
  - 1.2 randompoint
  - 1.3 StreamSimValues\_v2
  - 1.4 SetVehicleSpeed\_v1
  - 1.5 calculateStep\_v1
    - 1.5.1 readsensorData\_v6
    - 1.5.2 distcalc
    - 1.5.3 normaliseInCircle\_v2
  - 1.6 readvehicleSpeed\_v3
  - 1.7 storevehicleData\_v1

## RGSMoveToPoint\_v1

Η βασική ρουτίνα για την προσομοίωση των οχημάτων, για την μια διαδικασία προσέγγισης ενός συγκεκριμένου σημείου.

Συλλέγει όλα τα δεδομένα και εξάγει μια δομή δεδομένων από την προσομοίωση.

- 
- `function [ pioneerData ] = RGSMoveToPoint_v1 (vehicleName, vehiclemajority, points, pointmajority, random, totalstepTime, FuzzyCont_1 )`

```
% RGSMoveToPoint_v1. Ελεγχει μια προσομοίωση του Vrep.
% Στοιχεία εισόδου της ρουτίνας:
% 1) vehicleName : Το όνομα που έχει δοθεί στο όχημα που προσομοιώνουμε στην σκηνή του Vrep.
% 2) vehiclemajority: ο αριθμός των οχημάτων που προσομοιώνουμε στην σκηνή του Vrep.
% 3) points, Ένας πίνακας :
%         - Οπου υπάρχουν οι συντεταγμένες των σημείων, αν είναι σταθερά.
%         - Τα ορια για να βγουν τα σημεία αν είναι τυχαία.
% 4) pointmajority : Ο αριθμός των σημείων που θα χρησιμοποιήσουμε.
% 5) random : Ένας δείκτης παίρνει τιμές 0 ή 1.
%         0 - όχι τυχαία σημεία αλλά σταθερά.
%         1 - τυχαία σημεία οπότε χρησιμοποιούμε μια γεννήτρια σημείων.
% 6) totalstepTime : ο μέγιστος χρόνος ανά βήμα στην προσομοίωση.
% 7) FuzzyCont_1 : Ο ασαφής εκλεκτής που δουλεύουμε για την συγκεκριμένη προσομοίωση.
```

```
% Στοιχεία Εξόδου της ρουτίνας :
% 1) Pioneer : Η δομή με τα στοιχεία της προσομοίωσης.
% 2) clientID: : Δείκτης που δηλώνει την σύνδεση με το Vrep
○
warning ('off'); % Απενεργοποίηση μηνυμάτων προειδοποίησης
○ try
%% Ενεργοποίηση σύνδεσης μεταξύ MatLab και V-rep
% Χρήση του αρχείου (remoteApiProto.m)
○ vrep = remApi('remoteApi');
% Κλείσιμο όλων των ανοικτών συνδέσεων.
○ vrep.simxFinish(-1);
% Ξεκινάει η προσομοίωση
○ clientID = vrep.simxStart('127.0.0.1',19997,true,true,2000,5);
○ if clientID== -1
○ error('Connection Error: Failed to Connect');
○ else
○ disp('Connected to remote API server');
○
○ [errorCode]=vrep.simxStartSimulation(clientID,vrep.simx_opmode_oneshot_wait);
○ if errorCode ~= 0
○ error('Vrep error starting the simulation!');
○ else
○ disp('OK, starting the simulation');
○ end
○ end
○
%% Δημιουργία Δομής στοιχείων αντικειμένων της προσομοίωσης στο Matlab & ενεργοποίηση σε
% πραγματικό χρόνο των μεταβλητών στο Vrep.
○
% Δημιουργία Δομής στο MatLab με τα στοιχεία των αντικειμένων του Vrep.
○ [v] = CreateObjectHandles_v1 ( clientID, vrep, vehicleName, vehiclemajority);
```

```

% Ενεργοποίηση των μεταβλητών του Vrep σε πραγματικό χρόνο. streaming - buffer
o StreamSimValues_v2(clientID, vrep, v, vehicleName, vehiclemajority);
% Ορισμός την ταχύτητας των οχημάτων σε 0, ακινησία.
o motorCurrent = [0 0];
o SetVehicleSpeed_v1 (clientID, vrep, v, vehicleName, 3, 3 , motorCurrent);
o
%% Κυρίως ρουτίνα των robot
% timerforTask -----> Ο χρονομετρητής για όλη την διεργασία.
o timerforTask = tic;

% Ξεκινάει η διαδικασία, Ένα - Ένα για όλα τα σημεία.
o for pointnumber = 1:1:pointmajority;
  % Βοηθητικές Μεταβλητές
  % Διακόπτης που ελέγχει την διαδικασία του while.
o runswitch = 1;
  % Μεταβλητή που ελέγχει αν προσέγγισαν το σημείο.
o doneIt = 0;
  % Μετρητής των βημάτων που κάνει η ρουτίνα (βοηθάει στις καταχωρήσεις των δεδομένων
o stepcounter = 0 ;
  % Ορισμός πίνακα που καταχωρούμε τους χρόνους ανά βήμα ώστε να μπορούμε να
υπολογίσουμε
  % τον συνολικό χρόνο. Χρησιμοποιείται για την αποφυγή ατέρμονων διαδικασιών.
o stepTimer = [];
  % Ορισμός αρχικής τιμής για τον μετρητή βημάτων;
o currentstepTime = 0;
o
% Ορισμός του ονόματος του nN σημείου. Χρησιμοποιείται για την καταχώρηση των δεδομένων
% στην δομή των στοιχείων της προσομοίωσης.
o Goalnamels = strcat ('goal_pos_', int2str(pointnumber));
o
  % Εκκίνηση του χρονομετρητή για το συγκεκριμένο σημείο
o timerforGoal = tic;
o
  %% Επιλογή του τρόπου που θα βγει το σημείο που πρέπει να προσεγγίσουν τα οχήματα.
  % Αν έχουμε ορίσει τυχαία σημεία.
o if random == 1
  % Γίνεται επιλογή τυχαίου σημείου ανάλογα τα όρια που έχουμε ορίσει στο πίνακα 'points';
o xmin = points(1);
o xmax = points (2);
  % Το σημείο προσέγγισης υπολογίζεται από την ρουτίνα
o goal_pos = randompoint(xmin, xmax);
o else
  % Η επιλογή είναι το n σημείο από τα σταθερά σημεία που δίνουμε μέσω του πίνακα 'points'.
o goal_pos = points ( pointnumber,:);
o end
o
% Μετακινούμε το σημείο στην σκηνή του Vrep - Προσοχή στο όνομα της δομής των στοιχείων
% της προσομοίωσης.
% Αν αλλάξει το ονομα πρέπει να γίνει αλλαγή και στην μεταβλητή 'v.goal' στην παρακάτω
% εντολή.
o [errorCode]=vrep.simxSetObjectPosition(clientID, v.goal, -1, goal_pos,
vrep.simx_opmode_oneshot);
o
%% Η κυρίως ρουτίνα που τα οχήματα μετακινούνται για την εύρεση του σημείου.
o while(vrep.simxGetConnectionId(clientID)~-=-1) & (runswitch ~= 0)
  %Αύξηση του μετρητή βημάτων κατά 1
o stepcounter=stepcounter+1;

```

```

○
○ % Εκκίνηση του μετρητή για το συγκεκριμένο βήμα. Ελέγχουμε και τις περιπτώσεις που το
  % οχήματα κολλάνε.
○ timerforStep = tic;

  % Η διαδικασία είναι για κάθε όχημα χωριστά.
○ for vehiclenumber = 1:1:vehiclemajority
  % Υπολογισμός της στοιχείων για την προσέγγιση του σημείου.
○ [pointreached, simstepData] = calculateStep_v1 (clientID, vrep, v,
vehicleName, vehiclenumber, FuzzyCont_1, goal_pos);

  % Διαβάζουμε την ταχύτητα του οχήματος
○ [speedData] = readvehicleSpeed_v3 (clientID, vrep, v, vehicleName,
vehiclenumber);

  % Αποθήκευση των δεδομένων της προσομοίωσης
  % Εντός της ρουτίνας υπάρχει μια 'presistet' μεταβλητή για να κρατάει τα δεδομένα
  % της προσομοίωσης.
○ [data] = storevehicleData_v1 (v, vehicleName, vehiclenumber, pointnumber,
stepcounter, simstepData,speedData, goal_pos);

  % Ελεγχος εάν καποιο οχημα έδωσε μέσω της μεταβλητής 'doneIt'
στοιχεία ότι προσέγγισε το

  % σημείο.

○ if pointreached == 1
○ doneIt = 1;
○ runswitch = 0;
○ disp ('One of the Vehicles reached the point')
○ end
○ end

  % Σταματάει ο χρονοδιακόπτης του βήματος
○ stopthetimerforstep = toc(timerforStep);
  % Ανανεωση του πινακα των χρόνων
○ stepTimer(stepcounter) = stopthetimerforstep;
  % Υπολογισμός του ολικού χρόνου
○ currentstepTime = sum(stepTimer);
○ if currentstepTime > totalstepTime
○ disp (' Maximun Time for Goal - Time Break');
  % Καταχώρηση δεδομένων στην δομή
○ pioneerData.(GoalnameIs) = data.(GoalnameIs);
  % Καταχώρηση δεδομένων στην δομή
○ pioneerData.(GoalnameIs).stepTimer = stepTimer;
  % Αλλαγή τιμής στο διακόπτη για την διακοπή του 'while'
  runswitch = 0;
○ end
  % Έλεγχος αν το όχημα προσέγγισε το σημείο.
○ if doneIt == 1
  % Ενημέρωση της δομής των δεδομένων
○ pioneerData.(GoalnameIs) = data.(GoalnameIs);
  % Ενημέρωση της δομής των δεδομένων
○ pioneerData.(GoalnameIs).stepTimer = stepTimer;
○ end
○ end

  % Σταματημα και καταχώρηση του χρόνου για το συγκεκριμένο σημείο

```



```

o   stopthetimerforGoal = toc(timerforGoal);
o   pioneerData.(GoalnameIs).GoalTime = stopthetimerforGoal;
o   pioneerData.(GoalnameIs).doneIt = doneIt;
o   if ((pointnumber > pointmajority) || (pointnumber == pointmajority))
o       disp ('Task completed!');
o   else
o       disp ('Going to next point!');
o       runswitch = 1;
o   end
end

%Σταμάτημα και καταχώρηση του χρονομετρητή του ολικού διαδικασίας
o   stopthetimerforTask = toc(timerforTask);
o   pioneerData.TotalTime = stopthetimerforTask;
o
%Σταμάτημα των οχημάτων στην θέση που βρίσκονται.
o   motorCurrent = [0 0];
o   SetVehicleSpeed_v1 (clientID, vrep, v, vehicleName, 3, 3 , motorCurrent);

%Stop the simulation and close the connection
o   [errorCode]= vrep.simxStopSimulation(clientID,vrep.simx_opmode_oneshot);
o   pause(0.5);
o   vrep.simxFinish(-1);
o   vrep.delete();
o   disp('program end')
o
% Η εντολή catch..... αναγνωρίζει αν υπάρχει λάθος στην εκτέλεση του προγράμματος
% το εκτυπώνει το σφάλμα και διακόπτει της λειτουργίες.
o   catch err %if an error occurred the programm stops
o       disp(err.message)
o       load('C:\MVRS\Robot
Evaluation\Movement\Locomotion\Pioneer_Data_Bad.mat')
o       pause(1);
o   % close the line if still open
o       vrep.simxFinish(clientID);

%call the destructor!
o       vrep.delete();
o       disp('An Error has been occurred - Program termination ! ');
o   end %of try
o   end %of function
o

```

## CreateObjectHandles\_v1

```

o   function [ VrepObjectStructure ] = CreateObjectHandles_v1 (clientID, vrepFuncStruct,
vehicleName, vehiclemajority)
o
% [ VrepObjectStructure ] = CreateObjectHandles_v1 (clientID, vrepFuncStructure, vehicleName,
vehiclemajority)
% Δημιουργεί μια δομή όπου υπάρχουν οι δείκτες (Handles) των αντικειμένων της σκηνής του
Vrep.
% Δεν είναι αυτόματο. Το καθοδηγούμε τι θα διαβάσει.

% clientID : Vrep cliend, ορίζεται όταν γίνεται η σύνδεση MatLab με Vrep
% vrepFuncStructure : Η δομή όπου υπάρχουν όλες οι εντολές του Vrep remoteApiProto.m)
% vehicleName : Το όνομα που χρησιμοποιούμε στην σκηνή του Vrep.
% vehiclemajority: Ο αριθμός των οχημάτων που θα προσομοιώσουμε στην σκηνή.

% Τα αντικείμενα που ορίζουμε τους δείκτες είναι:
% Goal
% Pioneer Body , Pioneer Left Motor, Pioneer Right Motor
% Pioneer ultrasonic sensors :   Sensor 1, Sensor 2, Sensor 4, Sensor 5, Sensor 7, % Sensor 8, Sensor
9,
%
%                               Sensor 11, Sensor 14, Sensor 16
% Pioneer Vision sensors : Vision Sensor Front, Vision Sensor Back

% Ορίζουμε χρόνο καθυστέρησης διότι παρατηρήθηκε πρόβλημα με το ανάκληση του 1ου δείκτη
% από τα αντικείμενα.
o   pause (2)
o
% Ανάγνωση του δείκτη για το αντικείμενο Goal
o   [errorCode, v.goal] = vrepFuncStruct.simxGetObjectHandle(clientID,'goal',
vrepFuncStruct.simx_opmode_oneshot_wait);
o   if errorCode ~= 0
o       error('Error with object Goal');
o   end
o
% Ανάγνωση όλων των στοιχείων του οχήματος, σε επαναλαμβανόμενη διαδικασία ανάλογα των
% αριθμό οχημάτων που δόθηκε
o   for vehicleNumber = 1 : vehiclemajority
o       vehicleNameIs = strcat ( vehicleName, '_', int2str(vehicleNumber));
o       leftMotorNameIs = strcat ( vehicleNameIs,'_','leftMotor');
o       rightMotorNameIs = strcat ( vehicleNameIs,'_','rightMotor');
o       sensorNameIs = strcat ( vehicleName, '_', int2str(vehicleNumber), '_ultrasonicSensor');
o
o   %% Pioneer
o   % Pioneer Σώμα
o       [errorCode, v.(vehicleNameIs) ] = vrepFuncStruct.simxGetObjectHandle(clientID,
vehicleNameIs, vrepFuncStruct.simx_opmode_oneshot_wait);
o       if errorCode ~= 0
o           error(strcat('Error with :!', vehicleNameIs, ' handle'));
o       end
o
o   % Pioneer Αριστερό Μοτέρ
o       [errorCode, v.(leftMotorNameIs)] = vrepFuncStruct.simxGetObjectHandle(clientID,
leftMotorNameIs, vrepFuncStruct.simx_opmode_oneshot_wait);
o       if errorCode ~= 0
o           error(strcat('Error with :!', vehicleNameIs, ' left motor'));
o       end
o

```

```

○
○ % Pioneer Δεξιός Τροχός
○ [errorCode, v.(rightMotorNameIs)] =
vrepFuncStruct.simxGetObjectHandle(clientID, rightMotorNameIs,
vrepFuncStruct.simx_opmode_oneshot_wait);
○ if errorCode ~= 0
○ error(strcat('Error with :!', vehicleNameIs, ' right motor'));
○ end
○
○ %% Pioneer Αισθητήρες υπερήχων
○ % Sensor 1
○ sens1 = strcat (sensorNameIs, '1');
○ [errorCode, v.(sens1)] = vrepFuncStruct.simxGetObjectHandle(clientID, sens1,
vrepFuncStruct.simx_opmode_oneshot_wait);
○ if errorCode ~= 0
○ error(strcat('Error with :!', vehicleNameIs, ' ultrasonic sensor 1 object'));
○ end
○
○ % Sensor 2
○ sens2 = strcat (sensorNameIs, '2');
○ [errorCode, v.(sens2)] = vrepFuncStruct.simxGetObjectHandle(clientID, sens2,
vrepFuncStruct.simx_opmode_oneshot_wait);
○ if errorCode ~= 0
○ error(strcat('Error with :!', vehicleNameIs, ' ultrasonic sensor 2 object'));
○ end
○
○ % Sensor 4
○ sens4 = strcat (sensorNameIs, '4');
○ [errorCode, v.(sens4)] = vrepFuncStruct.simxGetObjectHandle(clientID, sens4,
vrepFuncStruct.simx_opmode_oneshot_wait);
○ if errorCode ~= 0
○ error(strcat('Error with :!', vehicleNameIs, ' ultrasonic sensor 4 object'));
○ end
○
○ % Sensor 5
○ sens5 = strcat (sensorNameIs, '5');
○ [errorCode, v.(sens5)] = vrepFuncStruct.simxGetObjectHandle(clientID, sens5,
vrepFuncStruct.simx_opmode_oneshot_wait);
○ if errorCode ~= 0
○ error(strcat('Error with :!', vehicleNameIs, ' ultrasonic sensor 5 object'));
○ end
○
○ % Sensor 7
○ sens7 = strcat (sensorNameIs, '7');
○ [errorCode, v.(sens7)] = vrepFuncStruct.simxGetObjectHandle(clientID, sens7,
vrepFuncStruct.simx_opmode_oneshot_wait);
○ if errorCode ~= 0
○ error(strcat('Error with :!', vehicleNameIs, ' ultrasonic sensor 5 object'));
○ end
○
○ % Sensor 8
○ sens8 = strcat (sensorNameIs, '8');
○ [errorCode, v.(sens8)] = vrepFuncStruct.simxGetObjectHandle(clientID, sens8,
vrepFuncStruct.simx_opmode_oneshot_wait);
○ if errorCode ~= 0
○ error(strcat('Error with :!', vehicleNameIs, ' ultrasonic sensor 8 object'));
○ end

```

```
% Sensor 9
o      sens9 = strcat (sensorNameIs, '9');
o      [errorCode, v.(sens9)] = vrepFuncStruct.simxGetObjectHandle(clientID, sens9,
vrepFuncStruct.simx_opmode_one-shot_wait);
o      if errorCode ~= 0
o          error(strcat('Error with :!', vehicleNameIs, ' ultrasonic sensor 9 object'));
o      end

% Sensor 11
o      sens11 = strcat (sensorNameIs, '11');
o      [errorCode, v.(sens11)] = vrepFuncStruct.simxGetObjectHandle(clientID, sens11,
vrepFuncStruct.simx_opmode_one-shot_wait);
o      if errorCode ~= 0
o          error(strcat('Error with :!', vehicleNameIs, ' ultrasonic sensor 11 object'));
o      end
o
% Sensor 14
o      sens14 = strcat (sensorNameIs, '14');
o      [errorCode, v.(sens14)] = vrepFuncStruct.simxGetObjectHandle(clientID, sens14,
vrepFuncStruct.simx_opmode_one-shot_wait);
o      if errorCode ~= 0
o          error(strcat('Error with :!', vehicleNameIs, ' ultrasonic sensor 14 object'));
o      end
o
% Sensor 16
o      sens16 = strcat (sensorNameIs, '16');
o      [errorCode, v.(sens16)] = vrepFuncStruct.simxGetObjectHandle(clientID, sens16,
vrepFuncStruct.simx_opmode_one-shot_wait);
o      if errorCode ~= 0
o          error(strcat('Error with :!', vehicleNameIs, ' ultrasonic sensor 16 object'));
o      end
o
%% Pioneer Αισθητήρες εικόνας
% Εμπρόσθιος Αισθητήρας Οράσεως
o      visFront = strcat (vehicleNameIs, '_vsFront');
o      [errorCode, v.(visFront)] = vrepFuncStruct.simxGetObjectHandle(clientID,
visFront, vrepFuncStruct.simx_opmode_one-shot_wait);
o      if errorCode ~= 0
o          error(strcat('Error with :!', vehicleNameIs, ' Front Vision Sensor'));
o      end
o
% Οπίσθιος Αισθητήρας Οράσεως
o      visBack = strcat (vehicleNameIs, '_vsBack');
o      [errorCode, v.(visBack)] = vrepFuncStruct.simxGetObjectHandle(clientID, visBack,
vrepFuncStruct.simx_opmode_one-shot_wait);
o      if errorCode ~= 0
o          error(strcat('Error with :!', vehicleNameIs, ' Back Vision Sensor'));
o      end
o      end
o
% Εξαγωγή της τελικής δομής όπου έχουν καταχωρηθεί οι δείκτες των αντικειμένων.
o      VrepObjectStructure = v;
```

## StreamSimValues\_v2

```

o    function StreamSimValues_v2(clientID, vrepFuncStruct, HandleStruct, vehicleName,
vehiclemajority )
o
% StreamSimValues_v1. Εναρξη της ροής δεδομένων της προσομοίωσης. Δεν εξάγει στοιχεία η
% ρουτίνα.
% Εναρξη της ροής δεδομένων από την προσομοίωση, και διαβάζει την τιμή από την
% ενεργοποιημένη αποικηκη δεδομένων. (buffer)
o
% Μη αυτοματοποιημένη διαδικασία, ορίζουμε τις μεταβλητές που θα ενεργοποιηθούν.
o
% clientID : Vrep πελάτης, ορίζεται όταν σταθεροποιηθεί η σύνδεση MAtLab με Vrep.
% vrepFuncStruct : Η δομή όπου είναι όλες οι εντολές του Vrep (remoteApiProto.m),
% HandleStruct : Η δομή με τους δείκτες των αντικειμένων της σκηνής του Vrep.
% vehicleName : Το όνομα του οχήματος που έχει δοθεί στην σκηνή του Vrep.
% vehiclemajority: Ο αριθμός των ενεργοποιημένων οχημάτων στην σκηνή του Vrep.
o
% Τα ονόματα των αντικειμένων είναι:
% Goal
% Pioneer Body , Pioneer Left Motor, Pioneer Right Motor
% Pioneer ultrasonic sensors :   Sensor 1, Sensor 2, Sensor 4, Sensor 5, Sensor 7,
%                               Sensor 8, Sensor 9, Sensor 11, Sensor 14, Sensor 16
% Pioneer Vision sensors : Vision Sensor Front, Vision Sensor Back

% Goal
% 1ο καλεσμα: Εναρξη ροής (streaming)
o    [errorCode, goal_pos] = vrepFuncStruct.simxGetObjectPosition(clientID,
HandleStruct.goal, -1, vrepFuncStruct.simx_opmode_streaming);
o    pause(0.2);
% Επόμενο καλεσμα, διαβάζουμε την τιμή που δίνει η μεταβλητή: buffering.
o    [errorCode, goal_pos] = vrepFuncStruct.simxGetObjectPosition(clientID,
HandleStruct.goal, -1, vrepFuncStruct.simx_opmode_buffer);
o    if errorCode ~= 0
o        error('Error with simxGetObjectPosition for goal!');
o    end
o
o    for vehicleNumber= 1:1:vehiclemajority
% Δημιουργία ονομάτων, ώστε να διαβάζουμε τους δείκτες των αντικειμένων.
o        vehicleNameIs = eval(strcat ( 'HandleStruct.',vehicleName, '_ ',
int2str(vehicleNumber)));
o        leftMotorNameIs = eval(strcat ('HandleStruct.',vehicleName, '_ ',
int2str(vehicleNumber), '_leftMotor'));
o        rightMotorNameIs = eval(strcat ('HandleStruct.',vehicleName, '_ ',
int2str(vehicleNumber), '_rightMotor'));
% Δημιουργία ονομάτων των αισθητήρων ώστε να διαβάσουμε τους δείκτες των αισθητήρων
o        sensornameIs = strcat ( vehicleName, '_ ', int2str(vehicleNumber),
'_ultrasonicSensor');
o
% Θεση του Οχήματος
[errorCode, actu_pos] = vrepFuncStruct.simxGetObjectPosition(clientID,
vehicleNameIs, -1, vrepFuncStruct.simx_opmode_streaming);

o        pause(0.2);
o        [errorCode, actu_pos] = vrepFuncStruct.simxGetObjectPosition(clientID,
vehicleNameIs, -1, vrepFuncStruct.simx_opmode_buffer);
o        if errorCode ~= 0

```

```

○          error('Error, can not find the actual position of the vehicle');
○          end
○          % Διευθυνση - γωνία απόκλισης
○          [errorCode, actu_orien]=vrepFuncStruct.simxGetObjectOrientation(clientID,
vehicleNameIs, -1, vrepFuncStruct.simx_opmode_streaming);
○          pause(0.2);
○          [errorCode, actu_orien]=vrepFuncStruct.simxGetObjectOrientation(clientID,
vehicleNameIs, -1, vrepFuncStruct.simx_opmode_buffer);
○          if errorCode ~= 0
○              error('Error, can not find the actual orientation of the vehicle');
○          end
○
○          %% Ταχύτητα
○          % Οχημα (Δεν διαβάζουμε την γωνιακή ταχύτητα του οχήματος)
○          [errorCode, linear_car_Velocity] = vrepFuncStruct.simxGetObjectVelocity
(clientID, vehicleNameIs, vrepFuncStruct.simx_opmode_streaming);
○          pause(0.2);
○          [errorCode, linear_car_Velocity]= vrepFuncStruct.simxGetObjectVelocity (clientID,
vehicleNameIs, vrepFuncStruct.simx_opmode_buffer);
○          if errorCode ~= 0
○              error('Error, can not read the vehicle velocity');
○          end
○
○          % Ταχύτητα αριστερού μοτέρ (Δεν διαβάζουμε την γωνιακή ταχύτητα του μοτέρ)
○          [errorCode, linear_left_Velocity] = vrepFuncStruct.simxGetObjectVelocity
(clientID, leftMotorNameIs, vrepFuncStruct.simx_opmode_streaming);
○          pause(0.2);
○          [errorCode, linear_left_Velocity]= vrepFuncStruct.simxGetObjectVelocity
(clientID, leftMotorNameIs, vrepFuncStruct.simx_opmode_buffer);
○          if errorCode ~= 0
○              error('Error, can not read the leftMotor velocity');
○          end
○
○          % Ταχύτητα δεξιού μοτέρ (Δεν διαβάζουμε την γωνιακή ταχύτητα)
○          [errorCode, linear_right_Velocity] = vrepFuncStruct.simxGetObjectVelocity
(clientID, rightMotorNameIs, vrepFuncStruct.simx_opmode_streaming);
○          pause(0.2);
○          [errorCode, linear_right_Velocity]= vrepFuncStruct.simxGetObjectVelocity
(clientID, rightMotorNameIs, vrepFuncStruct.simx_opmode_buffer);
○          if errorCode ~= 0
○              error('Error, can not set the rightMotor velocity');
○          end
○
○          %% Αισθητήρες
○          % Ενεργοποίηση των αισθητήρων. Ανοιγματος ροης των δεδομένων και διαβάζουμε την τιμή
% που στέλνει
% Οι αισθητήρες που είναι ενεργει είναι οι 16,1,2,4,5,7,8,9.
○
○          % Sensor 1
○          sens1 = eval(strcat ('HandleStruct.', sensornames, '1'));
○          [errorCode, det1, detP1] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens1, vrepFuncStruct.simx_opmode_streaming);
○          pause(0.2);
○          [errorCode, det1, detP1] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens1, vrepFuncStruct.simx_opmode_buffer);
○          if errorCode ~= 0
○              error('Error, with sensor 1');

```

```
○         end
○
○     % Sensor 2
○         sens2 = eval(strcat ('HandleStruct.', sensornameIs, '2'));
○
○         [errorCode, det2, detP2 ] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens2, vrepFuncStruct.simx_opmode_streaming);
○         pause(0.2);
○         [errorCode, det2, detP2] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens2, vrepFuncStruct.simx_opmode_buffer);
○         if errorCode ~= 0
○             error('Error, with sensor 2');
○         end
○
○     % Sensor 4
○         sens4 = eval(strcat ('HandleStruct.', sensornameIs, '4'));
○
○         [errorCode, det4, detP4 ] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens4, vrepFuncStruct.simx_opmode_streaming);
○         pause(0.2);
○         [errorCode, det4, detP4] = vrepFuncStruct.simxReadProximitySensor (clientID,
sens4, vrepFuncStruct.simx_opmode_buffer);
○         if errorCode ~= 0
○             error('Error, with sensor 4');
○         end
○
○     % Sensor 5
○         sens5 = eval(strcat ('HandleStruct.', sensornameIs, '5'));
○
○         [errorCode, det5, detP5] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens5, vrepFuncStruct.simx_opmode_streaming);
○         pause(0.2);
○         [errorCode, det5, detP5] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens5, vrepFuncStruct.simx_opmode_buffer);
○         if errorCode ~= 0
○             error('Error, with sensor 5');
○         end
○
○     % Sensor 7
○         sens7 = eval(strcat ('HandleStruct.', sensornameIs, '7'));
○
○         [errorCode, det7, detP7] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens7, vrepFuncStruct.simx_opmode_streaming);
○         pause(0.2)
○         [errorCode, det7, detP7] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens7, vrepFuncStruct.simx_opmode_buffer);
○         if errorCode ~= 0
○             error('Error, with sensor 7');
○         end
○
○     % Sensor 8
○         sens8 = eval(strcat ('HandleStruct.', sensornameIs, '8'));
○         [errorCode, det8, detP8] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens8, vrepFuncStruct.simx_opmode_streaming);
○         pause(0.2);
```

```

○           [errorCode, det8, detP8] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens8, vrepFuncStruct.simx_opmode_buffer);
○           if errorCode ~= 0
○               error('Error, with sensor 8');
○           end

% Sensor 9
○           sens9 = eval(strcat ('HandleStruct.', sensornameIs, '9'));
○
○           [errorCode, det9, detP9]=vrepFuncStruct.simxReadProximitySensor(clientID,sens9,
vrepFuncStruct.simx_opmode_streaming);
○           pause(0.2);
○           [errorCode, det9, detP9]=vrepFuncStruct.simxReadProximitySensor(clientID,sens9,
vrepFuncStruct.simx_opmode_buffer);
○           if errorCode ~= 0
○               error('Error, with sensor 9');
○           end

% Sensor 16
○           sens16 = eval(strcat ('HandleStruct.', sensornameIs, '16'));

○           [errorCode, det16, detP16] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens16, vrepFuncStruct.simx_opmode_streaming);
○           pause(0.2);
○           [errorCode, det16, detP16] = vrepFuncStruct.simxReadProximitySensor
(clientID, sens16, vrepFuncStruct.simx_opmode_buffer);
○           if errorCode ~= 0
○               error('Error, with sensor 16');
○           end

%% Den einai ftiagmeno .....

% %% Vision Sensors
% [number returnCode,boolean detectionState,array auxData,array auxPacketInfo]= ...
% simxReadVisionSensor(number clientID, number sensorHandle,number operationMode)
%
% [number returnCode,array resolution,libpointer image]= ...
% simxGetVisionSensorImage(number clientID, number sensorHandle, number options,
% number operationMode)
○           end

```



## SetVehicleSpeed\_v1

```

○
○   function SetVehicleSpeed_v1 (clientID, vrepFuncStruct, HandleStruct, vehicleName,
vehiclemajority, vehicleNumber, motorCurrent)
○
% SetVehicleSpeed_v1 : Ορίζει την γωνιακής ταχύτητας στα μοτέρ (αριστερό & Δεξιό)

% clientID : Πελάτης Vrep, καθορίζεται όταν γίνεται η σύνδεση MAtLab με Vrep.
% vrepFuncStruct : η δομή που περιέχει όλες της ρουτίνες του Vrep (remoteApiProto.m),
% HandleStruct : η δομή που περιέχει όλους τους δείκτες των αντικειμένων της σκηνης.
% vehicleName : Το όνομα του οχήματος που είναι στην σκηνή του Vrep.
% vehiclemajority: Ο αριθμός των οχημάτων στην σκηνή.
% vehicleNumber: Ο αριθμός των οχημάτων που προσομοιώνουμε στην σκηνή και ορίζουμε την
%                 ταχύτητα.
% motorCurrent : Ο πίνακας που ορίζουμε τις τιμές της γωνιακής ταχύτητας.

% Χωρίς Έξοδο, στέλνει την τιμή στο Vrep. Δεν υπάρχει επιστροφή από το Vrep.
○
○   if vehiclemajority == 1
% Δημιουργία ονομάτων ώστε να μπορούμε να διαβάσουμε τους δείκτες των αντικειμένων
○       leftMotorNames = eval(strcat ('HandleStruct.',vehicleName, '_',
int2str(vehicleNumber), '_leftMotor'));
○       rightMotorNames = eval(strcat ('HandleStruct.',vehicleName, '_',
int2str(vehicleNumber), '_rightMotor'));
○
% Ορίζει την τιμή της γωνιακής ταχύτητας, χωρίς απάντηση από το Vrep.
○       [errorCode]=vrepFuncStruct.simxSetJointTargetVelocity(clientID,
leftMotorNames, motorCurrent(1), vrepFuncStruct.simx_opmode_oneshot);
○       [errorCode]=vrepFuncStruct.simxSetJointTargetVelocity(clientID,
rightMotorNames, motorCurrent(2), vrepFuncStruct.simx_opmode_oneshot);
○       else
○           for vehicleNumber = 1:1:vehiclemajority
% Δημιουργία ονομάτων ώστε να μπορούμε να διαβάσουμε τους δείκτες των αντικειμένων
○               leftMotorNames = eval(strcat ('HandleStruct.',vehicleName, '_',
int2str(vehicleNumber), '_leftMotor'));
○               rightMotorNames = eval(strcat ('HandleStruct.',vehicleName, '_',
int2str(vehicleNumber), '_rightMotor'));
○
% Ορίζει την τιμή της γωνιακής ταχύτητας, χωρίς απάντηση από το Vrep.
○               [errorCode]=vrepFuncStruct.simxSetJointTargetVelocity(clientID,
leftMotorNames, motorCurrent(1), vrepFuncStruct.simx_opmode_oneshot);
○               [errorCode]=vrepFuncStruct.simxSetJointTargetVelocity(clientID,
rightMotorNames, motorCurrent(2), vrepFuncStruct.simx_opmode_oneshot);
○                   end
○           end
○       end
○   end

```

**calculateStep\_v1**

```

o   function [doneIt, SimulationData ] = calculateStep_v1 (clientID, vrepFuncStruct,
HandleStruct, vehicleName, vehicleNumber, FuzzyCont, goal_pos)
% calculateStep_v1. Δεδομένα από την κίνηση του οχήματος(ένα βήμα κάθε φορά)
% Είσοδος:
% clientID : Πελάτης Vrep, καθορίζεται όταν γίνεται η σύνδεση MATLAB με Vrep.
% vrepFuncStruct : η δομή που περιέχει όλες της ρουτίνας του Vrep (remoteApiProto.m),
% HandleStruct : η δομή που περιέχει όλους τους δείκτες των αντικειμένων της σκηνής.
% vehicleName : Το όνομα του οχήματος που είναι στην σκηνή του Vrep.
% vehiclenumber : Ο αριθμός των οχημάτων στην σκηνή.
% FuzzyCont: ο ασαφής εκλεκτής που χρησιμοποιούμε.
% goal_pos : Το σημείο που πρέπει να πιάσει το όχημα
%
% Εξοδος:      a) Δεδομένα της προσομοίωσης του οχήματος
%              b) Ένας δείκτης που μας καθορίζει αν είμαστε στο σημείο ή όχι
% Δημιουργία ονομάτων για την ανάγνωση των δεικτών των αντικειμένων
o     names = eval(strcat ('HandleStruct.',vehicleName, '_', int2str(vehicleNumber)));
o
% Διαβάζει την θέση XYZ
o     [errorCode, curr_pos] = vrepFuncStruct.simxGetObjectPosition(clientID, names, -1,
vrepFuncStruct.simx_opmode_buffer);
% Διαβάζει την διεύθυνση στο καρτεσιανό σύστημα
o     [errorCode, curr_orien] = vrepFuncStruct.simxGetObjectOrientation(clientID, names, -
1, vrepFuncStruct.simx_opmode_buffer);
% Υπολογισμός της απόστασης του οχήματος από το σημείο.
o     [req_dist, req_ang] = distcalc (curr_pos, goal_pos);
o     AngleDiff = req_ang - curr_orien(3);
o     N_AngleDiff = normaliseInCircle_v2(AngleDiff);
% Μετατροπή των τιμών σε double για χρήση με τον ασαφή ελεκτη
o     N_AngleDiff = double(N_AngleDiff);
o     req_dist = double(req_dist);
% Διαβάζει τα δεδομένα των αισθητήρων
o     [frD, rtD, lfD, frtD, frlfD, sensorData] = readsensorData_v6(clientID, vrepFuncStruct,
HandleStruct, vehicleName, vehicleNumber);
% Υπολογισμός της γωνιακής ταχύτητας
o     [motorCurrent] = evalfis([N_AngleDiff, frD, rtD, lfD, frtD, frlfD], FuzzyCont);
% Ορίζει την ταχύτητα των μοτερ
o     SetVehicleSpeed_v1 (clientID, vrepFuncStruct, HandleStruct, vehicleName, 1,
vehicleNumber, motorCurrent);
% Ελεγχος αν έχει φτάσει στο σημείο.
o     if req_dist < 1
o         doneIt = 1;
o     else
o         doneIt = 0;
o     end
% Κρατάμε τα δεδομένα από το βήμα.
o     SimulationData.current_position = curr_pos;
o     SimulationData.current_orientation = curr_orien;
o     SimulationData.request_distance = req_dist;
o     SimulationData.current_angle = req_ang;
o     SimulationData.AngleDifference = AngleDiff;
o     SimulationData.N_AngleDifference = N_AngleDiff;
o     SimulationData.motorCurrent = motorCurrent;
o     SimulationData.sensors = sensorData;
o
o     end

```

## readsensorData\_v6

```

○
○ function [frD, rtD, lfD, frrtD, frlfD, SensorData] = readsensorData_v6(clientID,
vrepFuncStruct, HandleStruct, vehicleName, vehicleNumber)

% readsensorData_v6. Καταγράφει τα δεδομένα από τους αισθητήρες και εξαγει τα στοιχεία
% που απαιτούνται από τον ασαφή ελεκτη

% Είσοδος:
% clientID : Πελάτης Vrep, καθορίζεται όταν γίνεται η σύνδεση MAtLab με Vrep.
% vrepFuncStruct : η δομή που περιέχει όλες της ρουτίνας του Vrep (remoteApiProto.m),
% HandleStruct : η δομή που περιέχει όλους τους δείκτες των αντικειμένων της σκηνής.
% vehicleName : Το όνομα του οχήματος που είναι στην σκηνή του Vrep.
% vehiclenumber : Ο αριθμός των οχημάτων στην σκηνή.
% FuzzyCont: ο ασαφής εκλεκτής που χρησιμοποιούμε.
% goal_pos : Το σημείο που πρέπει να πιάσει το όχημα

% Εξοδος:      1. frD:  Εμπροσθιος ελεγχος
%              2. rtD:  Δεξιος ελεγχος
%              3. lfD:  Αριστερός ελεγχος
%              4. frrtD: Μπροστινο δεξιά ελεγχος
%              5. frlfD: Μπροστά αριστερός έλεγχος
%              6. sensorData: μια δομή με όλα τα στοιχεία των αισθητήρων.
○
% Μόνο η μπροστινοι και πλαγιοι αισθητήρες είναι ενεργει 16,1,2,4,5,7,8,9.
% Ο κωδικας εχει και του οπισθιους αισθητήρες και 2 οπτικους αισθητήρες αλλα δεν είναι
% ενεργει.
○
%% Αισθητήρες
% Αισθητήρας 1
○ sens1 = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_ultrasonicSensor', '1'));
○ [errorCode,det1,detP1]=vrepFuncStruct.simxReadProximitySensor(clientID, sens1,
vrepFuncStruct.simx_opmode_buffer);
○
% Αισθητήρας 2
○ sens2 = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_ultrasonicSensor', '2'));
○ [errorCode,det2,detP2]=vrepFuncStruct.simxReadProximitySensor(clientID,sens2,
vrepFuncStruct.simx_opmode_buffer);
○
% Αισθητήρας 4
○ sens4 = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_ultrasonicSensor', '4'));
○ [errorCode,det4,detP4]=vrepFuncStruct.simxReadProximitySensor(clientID,sens4,
vrepFuncStruct.simx_opmode_buffer);

% Αισθητήρας 5
○ sens5 = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_ultrasonicSensor', '5'));
○ [errorCode,det5,detP5]=vrepFuncStruct.simxReadProximitySensor(clientID,sens5,
vrepFuncStruct.simx_opmode_buffer);
○

```

```

% Αισθητήρας 7
o   sens7 = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_ultrasonicSensor', '7'));
o   [errorCode,det7,detP7]=vrepFuncStruct.simxReadProximitySensor(clientID,sens7,
vrepFuncStruct.simx_opmode_buffer);
o
% Αισθητήρας 8
o   sens8 = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_ultrasonicSensor', '8'));
o   [errorCode,det8,detP8]=vrepFuncStruct.simxReadProximitySensor(clientID,sens8,
vrepFuncStruct.simx_opmode_buffer);
o
% Αισθητήρας 9
o   sens9 = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_ultrasonicSensor', '9'));
o   [errorCode,det9,detP9]=vrepFuncStruct.simxReadProximitySensor(clientID,sens9,
vrepFuncStruct.simx_opmode_buffer);
o
% % Αισθητήρας 11
% sens11 = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_ultrasonicSensor', % '11'));
% [errorCode,det11,detP11]=vrepFuncStruct.simxReadProximitySensor(clientID,sens11,
% vrepFuncStruct.simx_opmode_buffer);
% pause(0.1);
%
% Αισθητήρας 14
% sens14 = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_ultrasonicSensor', % '14'));
% [errorCode,det14,detP14]=vrepFuncStruct.simxReadProximitySensor(clientID,sens14,
% vrepFuncStruct.simx_opmode_buffer);
% pause(0.1);

% Αισθητήρας 16
o   sens16 = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_ultrasonicSensor', '16'));
o   [errorCode,det16,detP16] = vrepFuncStruct.simxReadProximitySensor (clientID, sens16,
vrepFuncStruct.simx_opmode_buffer);

% % % Οπτικός Αισθητήρας
% [number returnCode,boolean detectionState,array auxData,array auxPacketInfo]= ...
%   simxReadVisionSensor(number clientID, number sensorHandle,number operationMode)
%
% [number returnCode,array resolution,libpointer image]= ...
%   simxGetVisionSensorImage(number clientID,number sensorHandle,number
options,number operationMode)
o
% Υπολογισμός Αναγνώρισης
% Εμπροσθιος ελεγχος
o   front = det4 | det5;
% Δεξιός ελεγχος
o   right = det8 | det9;
% Αριστερός ελεγχος
o   left = det1 | det16;
% Μπροστινο δεξιά ελεγχος
o   frontright = det7;
o
% Μπροστινο αριστερο ελεγχος

```

```

○   frontleft = det2;
○
% For future use
% back = det11 | det14; %back detect
% backright = det9 && det11;
% backleft = det14 && det16;
○
% Εμπρόσθιος έλεγχος frD, επιλέγουμε την μικρότερη τιμή
○   if front == 1
○       if detP4(3) > 0 && detP5(3) > 0
○           if detP4(3) > detP5(3)
○               frD = detP5(3);
○           else
○               frD = detP4(3);
○           end
○       elseif detP5(3) < 0
○           frD = detP4(3);
○       elseif detP4(3) < 0
○           frD = detP5(3);
○       end
○   else
○       frD = 1.2;
○   end
○
% Δεξιός έλεγχος rtD, επιλέγουμε την μικρότερη τιμή
○   if right == 1
○       if detP8(3) > 0 && detP9(3) > 0
○           if detP8(3) > detP9(3)
○               rtD = detP9(3);
○           else
○               rtD = detP8(3);
○           end
○       elseif detP8(3) < 0
○           rtD = detP9(3);
○       elseif detP9(3) < 0
○           rtD = detP8(3);
○       end
○   else
○       rtD = 1.2;
○   end
○
% Αριστερός έλεγχος lfD, taking the closest value
○   if left == 1
○       if detP1(3) > 0 && detP16(3) > 0
○           if detP1(3) > detP16(3)
○               lfD = detP16(3);
○           else
○               lfD = detP1(3);
○           end
○       elseif detP1(3) < 0
○           lfD = detP16(3);
○       elseif detP16(3) < 0
○           lfD = detP1(3);
○       end
○   else
○       lfD = 1.2;
○   end

```

```
○
% Front right Detection frrtD, taking the closest value
○   if frontright == 1
○     frrtD = detP7(3);
○   else
○     frrtD = 1.2;
○   end
○
% Front left Detection frltD, taking the closest value
○   if frontleft == 1
○     frlfD = detP2(3);
○   else
○     frlfD = 1.2;
○   end
○
% % Back Detection bkD, taking the closest value
% if back == 1
%   if sensors.(detP11name)(3) > 0 && sensors.(detP14name)(3) > 0
%     if sensors.(detP11name)(3) > sensors.(detP14name)(3)
%       bkD = sensors.(detP14name)(3);
%     else
%       bkD = sensors.(detP11name)(3);
%     end
%   elseif sensors.(detP11name)(3) < 0
%     bkD = sensors.(detP14name)(3);
%   elseif sensors.(detP14name)(3) < 0
%     bkD = sensors.(detP11name)(3);
%   end
% else
%   bkD = 1.2;
% end

% % Back Right Detection bkrtD, taking the closest value
% if backright == 1
%   if sensors.(detP9name)(3) > 0 && sensors.(detP11name)(3) > 0
%     if sensors.(detP9name)(3) > sensors.(detP11name)(3)
%       bkrtD = sensors.(detP11name)(3);
%     else
%       bkrtD = sensors.(detP9name)(3);
%     end
%   elseif sensors.(detP9name)(3) < 0
%     bkrtD = sensors.(det11Pname)(3);
%   elseif sensors.(detP11name)(3) < 0
%     bkrtD = sensors.(detP9name)(3);
%   end
% else
%   bkrtD = 1.2;
% end

% % Back Left Detection bkltD, taking the closest value
% if backleft == 1
%   if sensors.(detP14name)(3) > 0 && sensors.(detP16name)(3) > 0
%     if sensors.(detP14name)(3) > sensors.(detP16name)(3)
%       bkltD = sensors.(detP16name)(3);
%     else
%       bkltD = sensors.(detP14name)(3);
%     end
%   elseif sensors.(detP14name)(3) < 0
```

```

%     bklfD = sensors.(det16Pname)(3);
%     elseif sensors.(detP16name)(3) < 0
%         bklfD = sensors.(detP14name)(3);
%     end
% else
%     bklfD = 1.2;
% end
% %*****
o
o     frD = double(frD);
o     rtD = double(rtD);
o     lfD = double(lfD);
o     frrtD = double(frrtD);
o     frlfD = double(frlfD);
% bk = double(bkD);
% bkrt = double(bkrtD);
% bklf = double(bklfD);
o

% Output the Sensors Data
%Front
o     SensorData.front.front = front;
o     SensorData.front.frD = frD;
o     SensorData.front.det4 = det4;
o     SensorData.front.detP4 = detP4;
o     SensorData.front.det5 = det5;
o     SensorData.front.detP5 = detP5;
o
%right
o     SensorData.right.right = right;
o     SensorData.right.rtD = rtD;
o     SensorData.right.det8 = det8;
o     SensorData.right.detP8 = detP8;
o     SensorData.right.det9 = det9;
o     SensorData.right.detP9 = detP9;
o
%left
o     SensorData.left.left = left;
o     SensorData.left.lfD = lfD;
o     SensorData.left.det1 = det1;
o     SensorData.left.detP1 = detP1;
o     SensorData.left.det16 = det16;
o     SensorData.left.detP16 = detP16;
o
%frontleft
o     SensorData.frontleft.frontleft = frontleft;
o     SensorData.frontleft.frlfD = frlfD;
o     SensorData.frontleft.det2 = det2;
o     SensorData.frontleft.detP2 = detP2;
o
%frontright
o     SensorData.frontright.frontright = frontright;
o     SensorData.frontright.frrtD = frrtD;
o     SensorData.frontright.det7 = det7;
o     SensorData.frontright.detP7 = detP7;
o
o     End

```

**distcalc**

```

○ function [dist, ang] = distcalc (startpoint, targetpoint)
% distcalc, Υπολογισμός της απόστασης και γωνίας μεταξύ 2 σημείων
% Παίρνει το δηλωθέν αρχικό σημείο και το τελικό σημείο. Υπολογίζει την αποσταση και την
γωνία % που σχηματίζει με τον άξονα των x.
○ distx = targetpoint(1) - startpoint(1);
○ disty = targetpoint(2) - startpoint(2);
○ dist = sqrt(distx^2 + disty^2);
○ ang = atan2(disty, distx);
○ end

```

**normaliseInCircle\_v2**

```

○ function [N_angle] = normaliseInCircle_v2(radIn)
% normaliseInCircle Μετατροπή των τιμών της γωνίας.
% Μετατροπή των τιμών της γωνίας ώστε να είναι από -180 έως 180

% Ελεγχος είναι ίση ή μικρότερη από -180 μοίρες
○ if radIn <= -pi
% μετατροπή της γωνίας στο εύρος -180 to 180
○ N_angle = radIn + 2*pi;
% Η γωνιά είναι μεγαλύτερη ή ίση από 180 μοίρες
○ elseif radIn >= pi
%μετατροπή της γωνίας στο εύρος -180 to 180
○ N_angle = radIn - 2*pi;
○ else
% Διατηρούμε την τρεχούσα τιμή
○ N_angle = radIn;
○ end
○ end

```

**randompoint**

```

○ function [randnum] = randompoint(xmin, xmax )
% Επιστρέφει ένα ζεύγος τυχαίων αριθμών που αντιπροσωπεύουν τις συντεταγμένες
% στο χώρο xyz. Το z = 0.
% Τα νούμερα σε δεδομένο εύρος τιμών που δίνεται είτε από τον χρήστη
% είτε απο άλλα στοιχεία του προγράμματος.
○ randnumxy = xmin + rand(1,2) * (xmax-xmin);
○ randnum = [randnumxy 0];
○ end

```



## readvehicleSpeed\_v3

```

○ function [speedData] = readvehicleSpeed_v3 (clientID, vrepFuncStruct, HandleStruct,
vehicleName, vehicleNumber)

% readvehicleSpeed_v3 Διαβάζει την ταχύτητα του οχήματος (χωρίς διεύθυνση)

% Διαβάζει την ταχύτητα του οχήματος (γραμμική και γωνιακή).
% Εξαγωγή των δεδομένων για επιπλέον υπολογισμούς.
% Vrep επιστρέφει την ταχύτητα στους άξονες x, y & z
% Εξαγωγή μόνο την γραμμική ταχύτητα, όχι την γραμμική ούτε την διεύθυνση.

% Εισοδος:
% clientID : Πελάτης Vrep, καθορίζεται όταν γίνεται η σύνδεση MAtLab με Vrep.
% vrepFuncStruct : η δομή που περιέχει όλες της ρουτίνες του Vrep (remoteApiProto.m),
% HandleStruct : η δομή που περιέχει όλους τους δείκτες των αντικειμένων της σκηνής.
% vehicleName : Το όνομα του οχήματος που είναι στην σκηνή του Vrep.
% vehiclenumber : Ο αριθμός των οχημάτων στην σκηνή.

% Εξοδος: Τα στοιχεία ταχύτητας του οχήματος.

%% Ταχύτητα
% Διαβάζουμε τον δείκτη του οχήματος
○ vehicleNameIs = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber)));
○ leftMotorNameIs = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_leftMotor'));
○ rightMotorNameIs = eval(strcat ('HandleStruct.', vehicleName, '_', int2str(vehicleNumber),
'_rightMotor'));

% Οχημα
% Δεν υπολογίζεται η γωνιακή ταχύτητα.
○ [errorCode, u]= vrepFuncStruct.simxGetObjectVelocity(clientID, vehicleNameIs,
vrepFuncStruct.simx_opmode_buffer);
○ if errorCode ~= 0
○ error('Error, can not read the vehicle velocity');
○ end
○ linear_car_Velocity = sqrt(u(1)^2 + u(2)^2);

% Αριστερό μοτέρ
% Δεν υπολογίζεται η γωνιακή ταχύτητα.
○ [errorCode, uL]= vrepFuncStruct.simxGetObjectVelocity(clientID, leftMotorNameIs,
vrepFuncStruct.simx_opmode_buffer);
○ if errorCode ~= 0
○ error('Error, can not read the leftMotor velocity');
○ end
○ linear_left_Velocity = sqrt(uL(1)^2 + uL(2)^2);

% Δεξιο μοτέρ
% Δεν υπολογίζεται η γωνιακή ταχύτητα.
○ [errorCode, uR]= vrepFuncStruct.simxGetObjectVelocity(clientID, rightMotorNameIs,
vrepFuncStruct.simx_opmode_buffer);
○ if errorCode ~= 0
○ error('Error, can not set the rightMotor velocity');
○ end
○ linear_right_Velocity = sqrt(uR(1)^2 + uR(2)^2);
○

```

% Καταχώρηση των τιμών που αναγνώσαμε και υπολογίσαμε

- speedData.linear\_car\_Velocity = linear\_car\_Velocity;
- speedData.u = u;
- speedData.linear\_left\_Velocity = linear\_left\_Velocity;
- speedData.uL = uL;
- speedData.linear\_right\_Velocity = linear\_right\_Velocity;
- speedData.uR = uR;
- end

## storevehicleData\_v1

```

○ function [finalData] = storevehicleData_v1 (HandleStruct, vehicleName, vehicleNumber,
pointnumber, stepcounter, simstepData, speedData, goal_pos)
○
% keepsimulationData_v1, Δημιουργεί και καταχωρεί τα δεδομένα της προσομοίωσης
○
% Εισοδος:
    %HandleStruct : η δομή που περιέχει όλους τους δείκτες των αντικειμένων της σκηνής.
    %vehicleName : Το όνομα του οχήματος που είναι στην σκηνή του Vrep.
    %vehiclenumber : Ο αριθμός των οχημάτων στην σκηνή.
    %pointnumber: αριθμός του σημείου
    %stepcounter: αριθμός των βημάτων,
    %simstepData: τα δεδομένα του βήματος,
    %speedData: στοιχεία ταχύτητας,
    %goal_pos : το δεδομένο σημείο προσέγγισης

% Εξοδος: Μια δομή με τα στοιχεία της προσομοίωσης
○
○ persistent SimData;
○
% Δημιουργία ονοματων για την καταχώρηση στη δομή.
○ GoalnameIs = strcat ('goal_pos_', int2str(pointnumber));
○ PionernameIs = strcat (vehicleName, '_', int2str(vehicleNumber));
○
○ SimData.(GoalnameIs).goal_position = goal_pos;
○
% Δεδομένα της προσομοίωσης για το βήμα
○ SimData.(GoalnameIs).(PionernameIs).current_possition(stepcounter,:) =
simstepData.current_possition;
○ SimData.(GoalnameIs).(PionernameIs).current_orientation(stepcounter,:) =
simstepData.current_orientation;
○ SimData.(GoalnameIs).(PionernameIs).requert_distance(stepcounter,:) =
simstepData.requert_distance;
○ SimData.(GoalnameIs).(PionernameIs).current_angle(stepcounter,:) =
simstepData.current_angle;
○ SimData.(GoalnameIs).(PionernameIs).AngleDifference(stepcounter,:) =
simstepData.AngleDifference;
○ SimData.(GoalnameIs).(PionernameIs).N_AngleDifference(stepcounter,:) =
simstepData.N_AngleDifference;
○ SimData.(GoalnameIs).(PionernameIs).N_AngleDifference(stepcounter,:) =
simstepData.N_AngleDifference;
○
% Δεδομένα ταχύτητας για το βήμα
○ SimData.(GoalnameIs).(PionernameIs).motorSpeed.leftMotorsetvalue(stepcounter,:) =
simstepData.motorCurrent(1);
○ SimData.(GoalnameIs).(PionernameIs).motorSpeed.rightMotorsetvalue(stepcounter,:) =
simstepData.motorCurrent(2);
○ SimData.(GoalnameIs).(PionernameIs).motorSpeed.carSpeed(stepcounter,:) =
speedData.linear_car_Velocity;
○ SimData.(GoalnameIs).(PionernameIs).motorSpeed.axecarSpeed(stepcounter,:) =
speedData.u;
○ SimData.(GoalnameIs).(PionernameIs).motorSpeed.leftmotorSpeed(stepcounter,:) =
speedData.linear_left_Velocity;
○ SimData.(GoalnameIs).(PionernameIs).motorSpeed.axeleftmotorSpeed(stepcounter,:) =
speedData.uL;

```

```

○ SimData.(GoalnameIs).(PioneernameIs).motorSpeed.rightmotorSpeed(stepcounter,:) =
speedData.linear_right_Velocity;
○ SimData.(GoalnameIs).(PioneernameIs).motorSpeed.axerightmotorSpeed(stepcounter,:) =
speedData.uR;
○
% Δεδομένα των αισθητήρων από το βήμα
% Μπροστά
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.front.front(stepcounter,:) =
simstepData.sensors.front.front;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.front.frD(stepcounter,:) =
simstepData.sensors.front.frD;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.front.det4(stepcounter,:) =
simstepData.sensors.front.det4;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.front.detP4(stepcounter,:) =
simstepData.sensors.front.det4;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.front.det5(stepcounter,:) =
simstepData.sensors.front.det5;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.front.detP5(stepcounter,:) =
simstepData.sensors.front.detP5;
○
% Δεξιά
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.right.right(stepcounter,:) =
simstepData.sensors.right.right;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.right.rtD(stepcounter,:) =
simstepData.sensors.right.rtD;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.right.det8(stepcounter,:) =
simstepData.sensors.right.det8;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.right.detP8(stepcounter,:) =
simstepData.sensors.right.detP8;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.right.det9(stepcounter,:) =
simstepData.sensors.right.det9;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.right.detP9(stepcounter,:) =
simstepData.sensors.right.detP9;
○
% Αριστερά
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.left.left(stepcounter,:) =
simstepData.sensors.left.left;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.left.lfD(stepcounter,:) =
simstepData.sensors.left.lfD;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.left.det1(stepcounter,:) =
simstepData.sensors.left.det1;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.left.detP1(stepcounter,:) =
simstepData.sensors.left.detP1;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.left.det16(stepcounter,:) =
simstepData.sensors.left.det16;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.left.detP16(stepcounter,:) =
simstepData.sensors.left.detP16;
○
% Μπροστά Αριστερά
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.frontleft.frontleft(stepcounter,:) =
simstepData.sensors.frontleft.frontleft;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.frontleft.frlfD(stepcounter,:) =
simstepData.sensors.frontleft.frlfD;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.frontleft.det2(stepcounter,:) =
simstepData.sensors.frontleft.det2;
○ SimData.(GoalnameIs).(PioneernameIs).Sensors.frontleft.detP2(stepcounter,:) =
simstepData.sensors.frontleft.detP2;

```

---

- 
- 
- **%Μπροστά Δεξιά**
- SimData.(GoalnameIs).(PioneernameIs).Sensors.frontright.frontright(stepcounter,:) = simstepData.sensors.frontright.frontright;
- SimData.(GoalnameIs).(PioneernameIs).Sensors.frontright.frrtD(stepcounter,:) = simstepData.sensors.frontright.frrtD;
- SimData.(GoalnameIs).(PioneernameIs).Sensors.frontright.det7(stepcounter,)= simstepData.sensors.frontright.det7;
- SimData.(GoalnameIs).(PioneernameIs).Sensors.frontright.detP7(stepcounter,)= simstepData.sensors.frontright.detP7;
- 
- finalData = SimData;
- End

## B. Ρουτίνες Εξελικτικής Διαδικασίας

- Η βασική ρουτίνα που ορίζουμε τις μεταβλητές για να ξεκινήσει η εξελικτική διαδικασία.

### LocoFuzzyOptim\_v2.m

```

○ clear all; close all; clc;
○ warning ('off');
○
%% Ορισμός Βασικών Μεταβλητών
% Ορισμός σταθερού αριθμού στην γεννήτρια τυχαίων αριθμών. Έχουμε σε κάθε τρέξιμο την
% ίδια ακολουθία τυχαίων αριθμών.
○
○ rng (10);
○ vehiclemajority = 3;
○ pointmajority = 3;
○ vName = 'Pioneer';
○
%% Δημιουργία του αρχικού πληθυσμού
%Φορτώνουμε τον ασαφή ελεγκτή
○ fuzzy_name = 'C:\MVRs\Robot evaluation\Movement\Locomotion\
Pioneer_Sensor_Loco_v6.fis';
○
% Δημιουργία της δομής που καταχωρείτε ο πληθυσμός από τον ασαφή ελεγκτή.
○ [pop] = CreatePopulationStructure (fuzzy_name, 1);
% Δημιουργία του 1ου χρωμοσωματος
○ genome = pop.controller_genes.controller_childs.genome;
○
% Καταχώρηση των βασικών στοιχείων του ελεγκτή ώστε να μπορούμε να δημιουργούμε
% καινούργιους για την αξιολόγηση.
○ Fuzzycontroller = pop.controller_genes.controller_childs(1).controller;
○
%Καθαρισμός μεταβλητών
○ clear pop fuzzy_name;
○
% Φορτώνουμε τις μεταβλητές για το optintool -
% Για κάθε περίπτωση φορτώνουμε και διαφορετικό αρχείο:
○ load('C:\MVRs\Robot Evaluation\Movement\Locomotion\LocoProblemTailor.mat');
% Φορτώνουμε τα όρια που μπορεί να ψάξει η εξελικτική διαδικασία.
○ load('C:\MVRs\Robot Evaluation\Movement\Locomotion\LocoFuzzyLimits.mat');
%Καθορίζουμε της αρχικές μεταβλητές στο optintool
○ LocoProblemTailor.options.InitialPopulation = genome;
○ LocoProblemTailor.options.InitialScores = [];
○ LocoProblemTailor.options.PopInitRange = LocoFuzzyLimits;
%% Σταθερά 3 σημεία για το simulation του Controller
○ shmeia = [10, 10, 0.2; -10, 0, 0.2; 3, -10, 0.2];
% Φορτωνουμε το optintool με τα δεδομένα στοιχεία.
○ optintool (LocoProblemTailor)

```

Πρέπει να τονίσουμε ότι ανάλογα την Συνάρτηση Καταλληλότητας φορτώνουμε και τα κατάλληλα δεδομένα. Έτσι έχουμε τρεις περιπτώσεις:

1. LocoProblemBehavioral για την Behavioral εξέλιξη.
2. LocoProblemAggregate για την Aggregate εξέλιξη.
3. LocoProblemTailor για την Tailor εξέλιξη.

## Fit\_LM\_SU\_v\_1

```
function [fmean] = Fit_LM_SU_v_1 (x)
```

```
% Fit_LM_SU_v_1 - Υπολογισμός της συνάρτησης καταλληλότητας Συμπεριφοράς
```

```
% f = 2.4 / (S + U);
```

```
% 2.4 είναι η μέγιστη τιμή του αθροίσματος μέγιστου S = 1.2 & μέγιστου U = 1.2
```

```
% S = μέση τιμή των τιμών που επιστρέφουν ολοι η αισθητήρες frD, lfD, rtD, frlfD, frtD
```

```
% U = μέση τιμή ταχύτητας οχήματος
```

```
% Ενσωμάτωση στην ρουτίνα των μεταβλητών που απαιτούνται για την ρουτίνα
```

```
○ shmeia = evalin('base', 'shmeia');
○ vehiclemajority = evalin('base', 'vehiclemajority');
○ pointmajority = evalin('base', 'pointmajority');
○ vName = evalin('base', 'vName');
```

```
%% Δημιουργία του ασαφή ελεγκτή που θα αξιολογηθεί
```

```
○ Fuzzycontroller = evalin('base', 'Fuzzycontroller');
○ [TrexonElextis] = CreateFuzzyForEvaluation_v3 (Fuzzycontroller, x);
% Ενσωμάτωση στην ρουτίνα του ελεγκτή
○ assignin ('base', 'TrexonElextis', TrexonElextis);
```

```
% Εκτέλεση της προσομοίωσης
```

```
○ disp (** Running the robot simulation **);
○ Pioneer_Data = RGSMoveToPoint_v1 (vName, 3, shmeia, 3, 0, 60, TrexonElextis );
```

```
% Ενσωμάτωση στην ρουτίνα των αποτελεσμάτων της προσομοίωσης
```

```
○ assignin ('base', 'Pioneer_Data', Pioneer_Data);
○
```

```
%% Υπολογισμός της συνάρτησης καταλληλότητας
```

```
% Αρχικοποίηση μεταβλητών
```

```
○ sall=[];
○ uall=[];
○ fall = [];
```

```
% Ελεγχος όλων των οχημάτων
```

```
○ for i = 1:1:vehiclemajority
%ορισμός ονομάτων για την ολοκλήρωση των υπολογισμών
○ pionernameIs = strcat('Pioneer_', int2str(i));
```

```
%% Υπολογισμός μέσης τιμής των μετρήσεων από τους αισθητήρες.
```

```
○ for j = 1:1:pointmajority
○ goalposIs = strcat('goal_pos_', int2str(j));
○ sF = mean(Pioneer_Data.(goalposIs).(pionernameIs).Sensors.front.frD);
○ sR = mean(Pioneer_Data.(goalposIs).(pionernameIs).Sensors.right.rtD);
○ sL = mean(Pioneer_Data.(goalposIs).(pionernameIs).Sensors.left.lfD);
○ sFL =
mean(Pioneer_Data.(goalposIs).(pionernameIs).Sensors.frontleft.frlfD);
○ sFR =
mean(Pioneer_Data.(goalposIs).(pionernameIs).Sensors.frontright.frrtD);
○ sall = [sall, sF, sR, sL, sFL, sFR];
```

```
○ end
○ S = mean(sall);
○ % Αρχικοποίηση & καθαρισμός μεταβλητών
○ clear goalposIs j;
○ sall = [];

○ %% Υπολογισμός μέσης τιμή ταχύτητας οχήματος
○ for j = 1:1:pointmajority
○ goalposIs = strcat('goal_pos_', int2str(j));
○ u =
mean(Pioneer_Data.(goalposIs).(pioneernameIs).motorSpeed.carSpeed(:));
○ uall = [uall, u];
○ end
○ U = mean(uall);
○ % Αρχικοποίηση & καθαρισμός μεταβλητών
○ clear goalposIs j;
○ uall = [];

○ %% Υπολογισμός της συνάρτησης καταλληλότητας για το όχημα
○ f = 2.4 / (S + U);
○ fall = [fall, f];
○ end
○ % Υπολογισμός της μέσης τιμής από όλες τις τιμές των συναρτήσεων καταλληλότητας
○ fmean = mean(fall);

○ %Καθαρισμός μεταβλητών
○ clear fall
○ end
```



## Fit\_LM\_P\_v\_2

```

○ function [f] = Fit_LM_P_v_2(x)
% Fit_LM_P_v_2 - Αθροιστική ή Ολοκληρωτική Συνάρτηση καταλληλοτητας.
% f = th ;
% th είναι η τιμή που μας δείχνει ποσα σημεία επιτύχαμε.

% Ενσωμάτωση στην ρουτίνα των μεταβλητών που απαιτούνται για την ρουτίνα
○ shmeia = evalin('base', 'shmeia');
○ vehiclemajority = evalin('base', 'vehiclemajority');
○ pointmajority = evalin('base', 'pointmajority');
○ vName = evalin('base', 'vName');

% Δημιουργούμε τον ασαφή ελεκτη που θα δοκιμάσουμε
○ Fuzzycontroller = evalin('base', 'Fuzzycontroller');
○ [TrexonElextis] = CreateFuzzyForEvaluation_v3 (Fuzzycontroller, x);
% Ενσωμάτωση στην ρουτίνα του ελεγκτή
○ assignin ('base', 'TrexonElextis', TrexonElextis);
○
% Εκτελούμε την προσομοίωση
○ disp (** Running the robot simulation **);
○ Pioneer_Data = RGSMoveToPoint_v1 (vName, 3, shmeia, 3, 0, 60, TrexonElextis );
% Ενσωμάτωση στην ρουτίνα των αποτελεσμάτων της προσομοίωσης
○ assignin ('base', 'Pioneer_Data', Pioneer_Data);
○
%% Υπολογισμός της τιμής της συναρτησης καταλληλότητας
% starting values
○ th = 0;
○ doneItcounter = 0;
○
% Ελέγχουμε ποσα σημεία προσέγγισαν τα οχήματα.
○ for j=1:1:pointmajority
○ goalposIs = strcat('goal_pos_', int2str(j));
○ if Pioneer_Data(goalposIs).doneIt == 1
○ doneItcounter = doneItcounter + 1;
○ end
○ end
○
○ if doneItcounter == 3
○ th = 1;
○ elseif doneItcounter == 2
○ th = 10;
○ elseif doneItcounter == 1
○ th = 20;
○ elseif doneItcounter == 0
○ th = 30;
○ end
○
○ f = th;
○
% Επαναφορά των μεταβλητών στις αρχικές τιμές ή καθαρισμός αυτών
○ doneItcounter = 0;
○ clear goalposIs j;
○ clear Pioneer_Data th
○ end

```

## Fit\_LM\_PSU\_v\_4

```

○ function [fmean] = Fit_LM_PSU_v_4 (x)
○
% f = th +(2.4 / (S + U ));
% th ποινή, ανάλογη των επιτυχημένων σημείων.
% 2.4 είναι η μεγιστη τιμή του αθροίσματος S = 1.2 & max U = 1.2
% S = μεση τιμή των ,ετρήσεων από όλους τους αισθητήρες frD, lfD, rtD, frlfD, frrtD
% U = μεση τιμή της ταχύτητας

% Ενσωμάτωση στην ρουτίνα των μεταβλητών που απαιτούνται για την ρουτίνα
○ shmeia = evalin('base', 'shmeia');
○ vehiclemajority = evalin('base', 'vehiclemajority');
○ pointmajority = evalin('base', 'pointmajority');
○ vName = evalin('base', 'vName');
○
%% Δημιουργία του ασαφή ελεγκτή που θα μελετηθεί
○ Fuzzycontroller = evalin('base', 'Fuzzycontroller');
○ [TrexonElextis] = CreateFuzzyForEvaluation_v3 (Fuzzycontroller, x);
% Ενσωμάτωση στην ρουτίνα του ελεγκτή
○ assignin ('base', 'TrexonElextis', TrexonElextis);
○
% Εκτελούμε τη προσομοίωση
○ disp (** Running the robot simulation **);
○ Pioneer_Data = RGSMoveToPoint_v1 (vName, 3, shmeia, 3, 0, 60, TrexonElextis );
% Ενσωμάτωση στην ρουτίνα των αποτελεσμάτων της προσομοίωσης
○ assignin ('base', 'Pioneer_Data', Pioneer_Data);
○
%% Υπολογισμός της συνάρτησης καταλληλότητας
% Αρχικές τιμές των μεταβλητών
○ th = 0;
○ doneItcounter = 0;
○ sall=[];
○ uall=[];
○ fall = [];
○
%% Υπολογισμός των σημείων που προσέγγισαν τα οχήματα.
○ for j=1:1:pointmajority
○ goalposIs = strcat('goal_pos_', int2str(j));
○ if Pioneer_Data.(goalposIs).doneIt == 1
○ doneItcounter = doneItcounter + 1;
○ end
○ end
○ if doneItcounter == 3
○ th = 0;
○ end
○ if doneItcounter == 2
○ th = 10;
○ end
○ if doneItcounter == 1
○ th = 20;
○ end
○ if doneItcounter == 0
○ th = 30;
○ end

```

```

% Αρχικοποίηση και καθαρισμός μεταβλητών
○      clear goalposIs j;
○      doneItcounter = 0;
○
% Ελέγχουμε όλα τα οχήματα
○      for i = 1:1:vehiclemajority
○          % Δημιουργία ονόματος για την ανάγνωση των τιμών
○              pioneernameIs = strcat('Pioneer_', int2str(i));
○          %% Διαβάζουμε τις τιμές όλων των αισθητήρων και εξάγουμε την μέση τιμή
○              for j = 1:1:pointmajority
○                  goalposIs = strcat('goal_pos_', int2str(j));
○                  sF = mean(Pioneer_Data.(goalposIs).(pioneernameIs).Sensors.front.frD);
○                  sR = mean(Pioneer_Data.(goalposIs).(pioneernameIs).Sensors.right.rtD);
○                  sL = mean(Pioneer_Data.(goalposIs).(pioneernameIs).Sensors.left.lfD);
○                  sFL =
mean(Pioneer_Data.(goalposIs).(pioneernameIs).Sensors.frontleft.frlfD);
○                  sFR =
mean(Pioneer_Data.(goalposIs).(pioneernameIs).Sensors.frontright.frftD);
○                  sall = [sall, sF, sR, sL, sFL, sFR];
○                  end
○                  S = mean(sall);
○          % καθαρισμός των μεταβλητών
○              clear goalposIs j;
○              sall = [];
○          %% Εξάγουμε την μέση τιμή της ταχύτητας του οχήματος
○              for j = 1:1:pointmajority
○                  goalposIs = strcat('goal_pos_', int2str(j));
○                  u =
mean(Pioneer_Data.(goalposIs).(pioneernameIs).motorSpeed.carSpeed(:));
○                  uall = [uall, u];
○                  end
○                  U = mean(uall);

○          % καθαρισμός των μεταβλητών
○              clear goalposIs j;
○              uall = [];
○
○          % Υπολογισμός της συνάρτησης καταλληλότητας για το όχημα
○              f = th + (2.4 / (S + U));
○              fall = [fall, f];
○          end

% Υπολογισμός της μέσης τιμής όλων των συναρτήσεων από τα οχήματα
○      fmean = mean(fall);

% καθαρισμός των μεταβλητών
○      clear fall
○      end

```

Η ρουτίνα του MatLab, OptimTool, μας δίνεται η δυνατότητα να συντάξουμε (με συγκεκριμένο τρόπο) μια ρουτίνα μέσω της οποίας ο χρήστης να παίρνει στοιχεία από την διαδικασία της εξέλιξης που δεν τα παρέχει η ρουτίνα.

Έγινε χρήση της ρουτίνας αυτής ώστε να μπορέσουμε να κρατήσουμε σε μία δομή, όλους τους γονείς από κάθε γενεά και την βαθμολογία που αυτοί επιτύγχαναν.

Έτσι μπορεί να γίνει και αξιολόγηση των αποτελεσμάτων.

### LocoMoveoutputfcn\_v4

```

○ function [state, options, optchanged] = LocoMoveoutputfcn_v4 (options, state, flag)
○     optchanged = false;
○     persistent GenerationStruct
○     switch flag
○         case 'init'
○             disp('Init ...');
○             disp('Saving Population Data');
○         % Create the generation structure
○             gen_num = state.Generation;
○             GenerationStruct.Population = state.Population;
○             GenerationStruct.Fitness = state.Score;
○         % Save the structure with the data of the simulations
○             Dname = ['C:\MVRs\Robot
Evaluation\Movement\Loco_Eval_MotionData\'];
○             FileName = ['GenerationStruct_'];
○             t = date;
○             Sname = strcat(Dname, FileName, t);
○             save (Sname, 'GenerationStruct');
○         % Clear the unneeded values
○             clear Dname FileName t Sname;
○
○         case {'iter','interrupt'}
○             disp('Iterating ...');
○             disp('Saving Population Data');
○         % Update the generation structure
○             gen_num = state.Generation;
○             GenerationStruct(gen_num+1).Population = state.Population;
○             GenerationStruct(gen_num+1).Fitness = state.Score;
○         % Save the structure with the data of the simulations
○             Dname = ['C:\MVRs\Robot
Evaluation\Movement\Loco_Eval_MotionData\'];
○             FileName = ['GenerationStruct_'];
○             t = date;
○             Sname = strcat(Dname, FileName, t);
○             save (Sname, 'GenerationStruct');
○         % Clear the unneeded values
○             clear Dname FileName t Sname;
○         case 'done'
○             disp('Ok, Finished!');
○     end
○ end
○

```

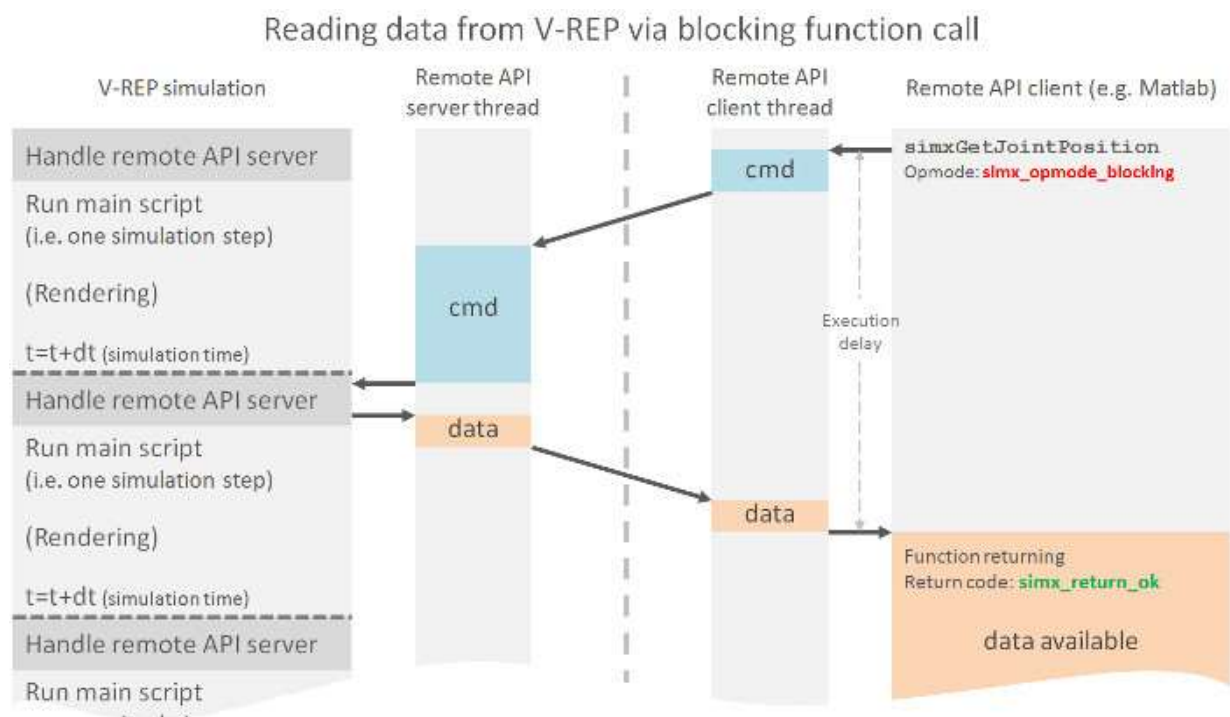
Η συνάρτηση αυτή είναι κοινή για όλες τις περιπτώσεις εξέλιξης που χρησιμοποιήσαμε.

## Ε. Τύποι λειτουργίας Remote API [64]

Το Remote API επιτρέπει στο χρήστη να επιλέξει τον τύπο της λειτουργίας και τον τρόπο που ο προσομοίωτης δουλεύει παρέχοντας τέσσερις βασικούς μηχανισμούς για την εκτέλεση κλήσεων λειτουργίας ή για τον έλεγχο της προόδου προσομοίωσης:

- Αποκλεισμός κλήσεων λειτουργίας - (Blocking function calls)
- Κλήσεις λειτουργίας μη αποκλεισμού - (Non-blocking function calls)
- Ροή δεδομένων - (Data streaming)
- Σύγχρονη λειτουργία - (Synchronous operation)

### Αποκλεισμός κλήσεων λειτουργίας - (Blocking function calls)

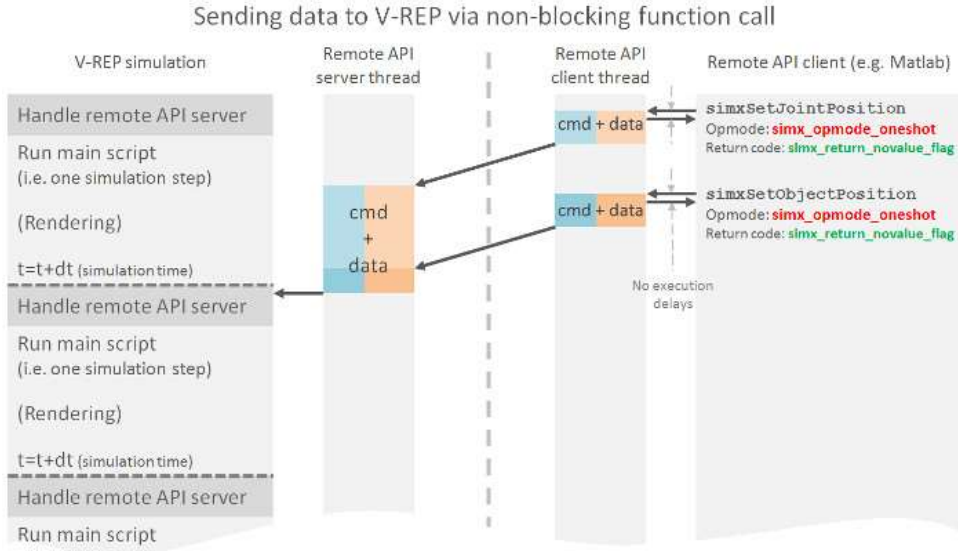


Εικόνα 187: Λειτουργία Αποκλεισμού Κλήσεων

Μια κλήση λειτουργία αποκλεισμού είναι η απλή ή τακτική προσέγγιση, και συμβαίνει για καταστάσεις όπου δεν έχουμε την πολυτέλεια να περιμένουμε μια απάντηση από το διακομιστή.

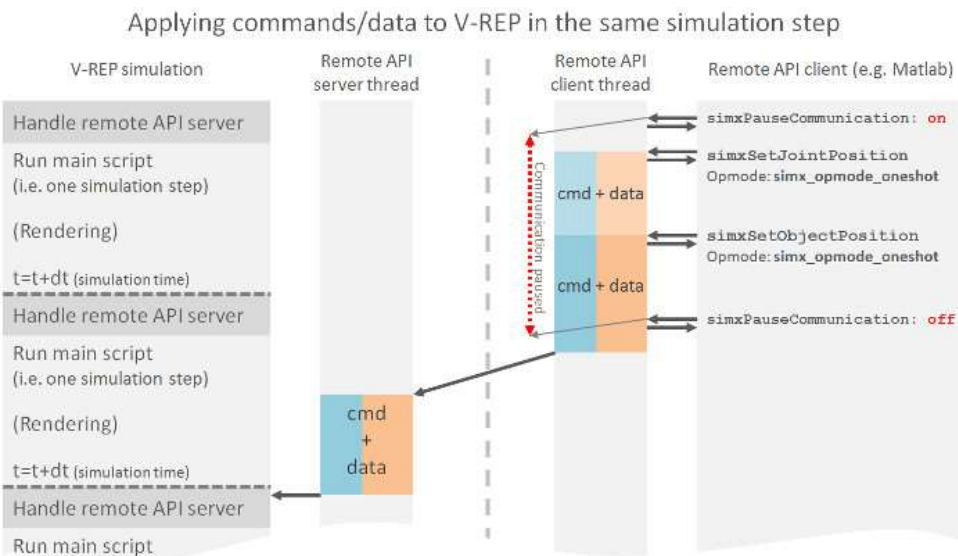
## Κλήσεις λειτουργίας μη αποκλεισμού - (Non-blocking function calls)

Κλήσεις λειτουργίας μη αποκλεισμού: μια κλήση λειτουργία μη αποκλεισμού προορίζεται για καταστάσεις όπου θέλουμε απλά να στείλουμε δεδομένα στο V-Rep χωρίς την ανάγκη για μια απάντηση.



Εικόνα 188 : Κλήση Λειτουργίας μη αποκλεισμού

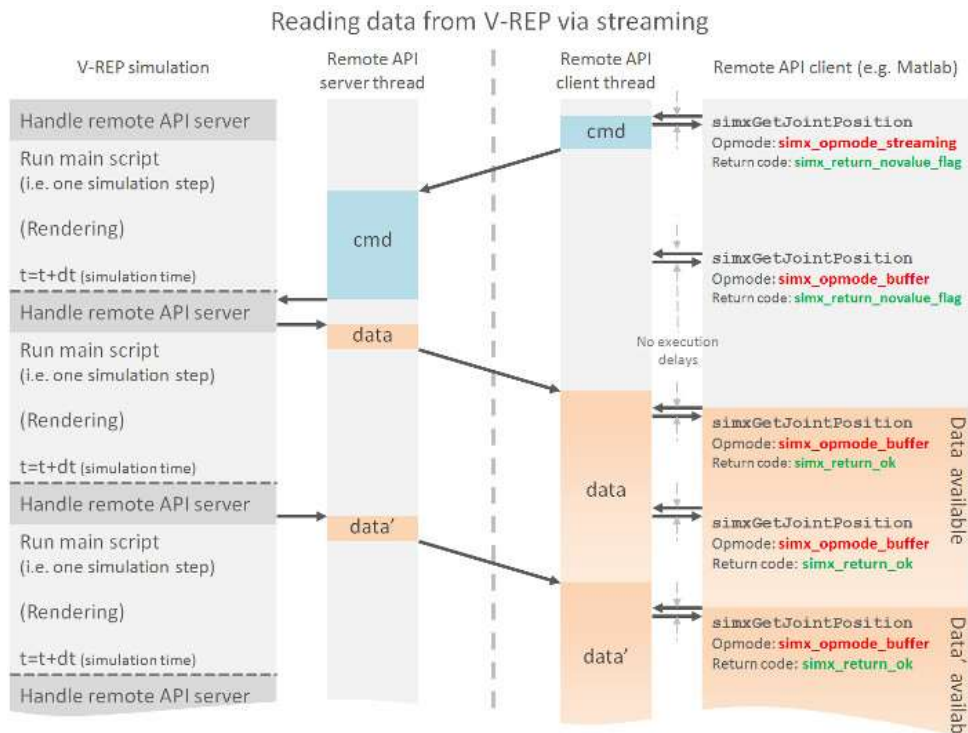
Σε ορισμένες περιπτώσεις, είναι σημαντικό να σταλούν διάφορα δεδομένα μέσα στο ίδιο μήνυμα, προκειμένου να έχουν τα δεδομένα που εφαρμόζονται στην ίδια χρονική στιγμή στην πλευρά του διακομιστή (π.χ. θέλουμε οι 3 αρθρώσεις ενός ρομπότ να πρέπει να κινηθούν στο μοντέλο του V-Rep την ίδια ακριβώς στιγμή, δηλαδή στο ίδιο βήμα της προσομοίωσης). Στην περίπτωση αυτή, ο χρήστης μπορεί να σταματήσει προσωρινά το νήμα επικοινωνίας, προκειμένου να επιτευχθεί αυτό, όπως φαίνεται στο ακόλουθο παράδειγμα:



Εικόνα 189: Διακοπή Επικοινωνίας για αποστολή μαζικών εντολών

## Ροή δεδομένων - (Data streaming)

Ο διακομιστής μπορεί να προβλέψει τον τύπο των δεδομένων που απαιτεί ο υπολογιστής-πελάτης. Για να συμβεί αυτό, ο πελάτης πρέπει να σηματοδοτήσει αυτή την επιθυμία στο πρόγραμμα διαχειριστή με ένα "streaming" ή μια σημαία 'συνεχούς λειτουργίας' (δηλαδή η λειτουργία είναι αποθηκευμένη στην πλευρά του διακομιστή, εκτελείται και στέλνει δεδομένα σε τακτική βάση, χωρίς την ανάγκη για νέο αίτημα από τον πελάτη). Αυτό μπορεί να θεωρηθεί ως μια συνδρομή εντολή/μήνυμα από τον υπολογιστή-πελάτη στο διακομιστή, ο διαχειριστής θα έχει συνεχή ροή των δεδομένων στον υπολογιστή-πελάτη.



Εικόνα 190: Λειτουργία συνεχούς ροής (streaming)

Μόλις τελειώσει η ροή των δεδομένων, ο απομακρυσμένος πελάτης API θα πρέπει να ενημερώνει πάντα το διακομιστή (δηλαδή V-REP) για να σταματήσει η ροή δεδομένων, αλλιώς ο διαχειριστής θα συνεχίσει να ροή μη αναγκαίων δεδομένων με αποτέλεσμα την αργή επικοινωνία. Χρησιμοποιήστε τον τρόπο λειτουργίας `simx_opmode_discontinue` για την διακοπή της αποστολής των δεδομένων.

## Σύγχρονη λειτουργία - (Synchronous operation)

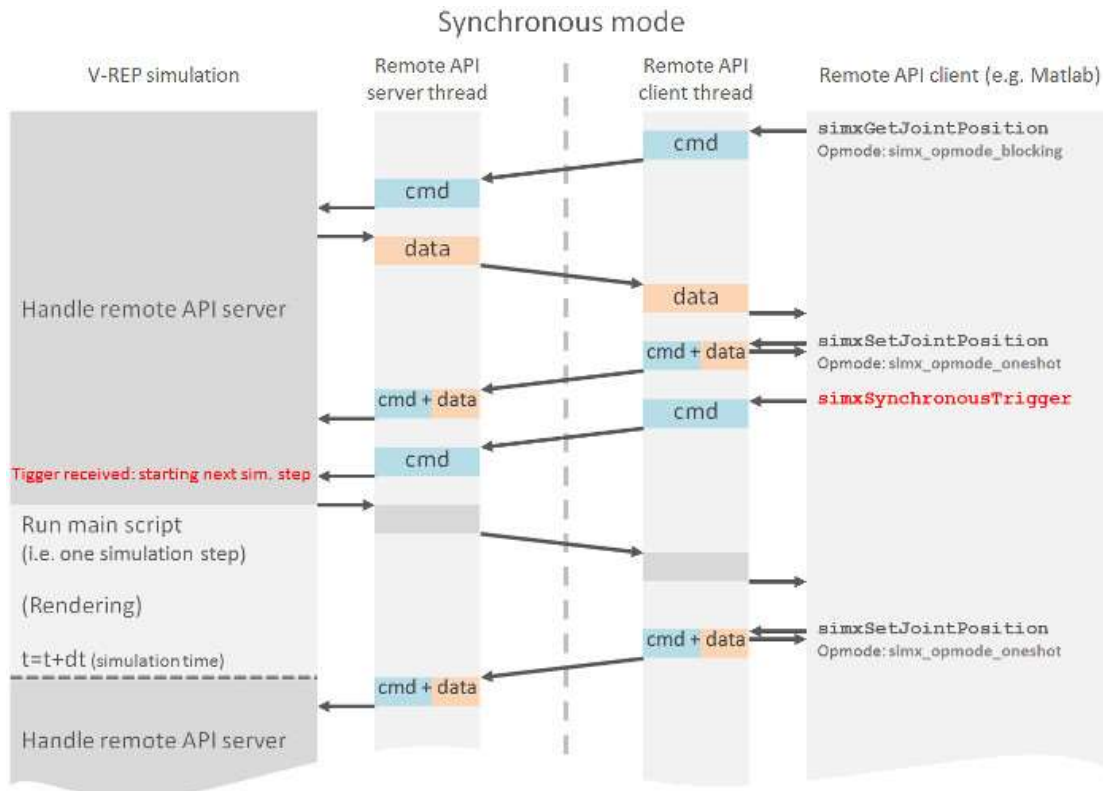
Σύγχρονη λειτουργία: από τις παραπάνω κλήσεις λειτουργίας παρατηρούμε ότι μια προσομοίωση θα προωθηθεί ή θα προχωρήσει χωρίς να ληφθεί υπόψη η πρόοδος του απομακρυσμένου προγράμματος-πελάτη API.

Οι κλήσεις των απομακρυσμένων συναρτήσεων API θα εκτελεστούν ασύγχρονα από προεπιλογή. Υπάρχουν ωστόσο καταστάσεις όπου ο απομακρυσμένος υπολογιστής-πελάτης API πρέπει να συγχρονιστεί με την πρόοδο της προσομοίωσης, ελέγχοντας

την προσομοίωση από την πλευρά του απομακρυσμένου πελάτη API. Αυτό μπορεί να επιτευχθεί με τη χρήση του απομακρυσμένου API συγχρονισμένης λειτουργίας.

Η υπηρεσία απομακρυσμένου διακομιστή API χρειάζεται σε αυτήν την περίπτωση να είναι ενεργοποιημένη για συγχρονισμένη λειτουργία (αυτό μπορεί να επιτευχθεί μέσω της συνάρτησης `simRemoteApi.Start` ή μέσω του συνεχούς απομακρυσμένου αρχείου ρύθμισης παραμέτρων της υπηρεσίας διακομιστή API `remoteApiConnections.txt`).

Ακολουθεί ένα παράδειγμα της συγχρονισμένης λειτουργίας:



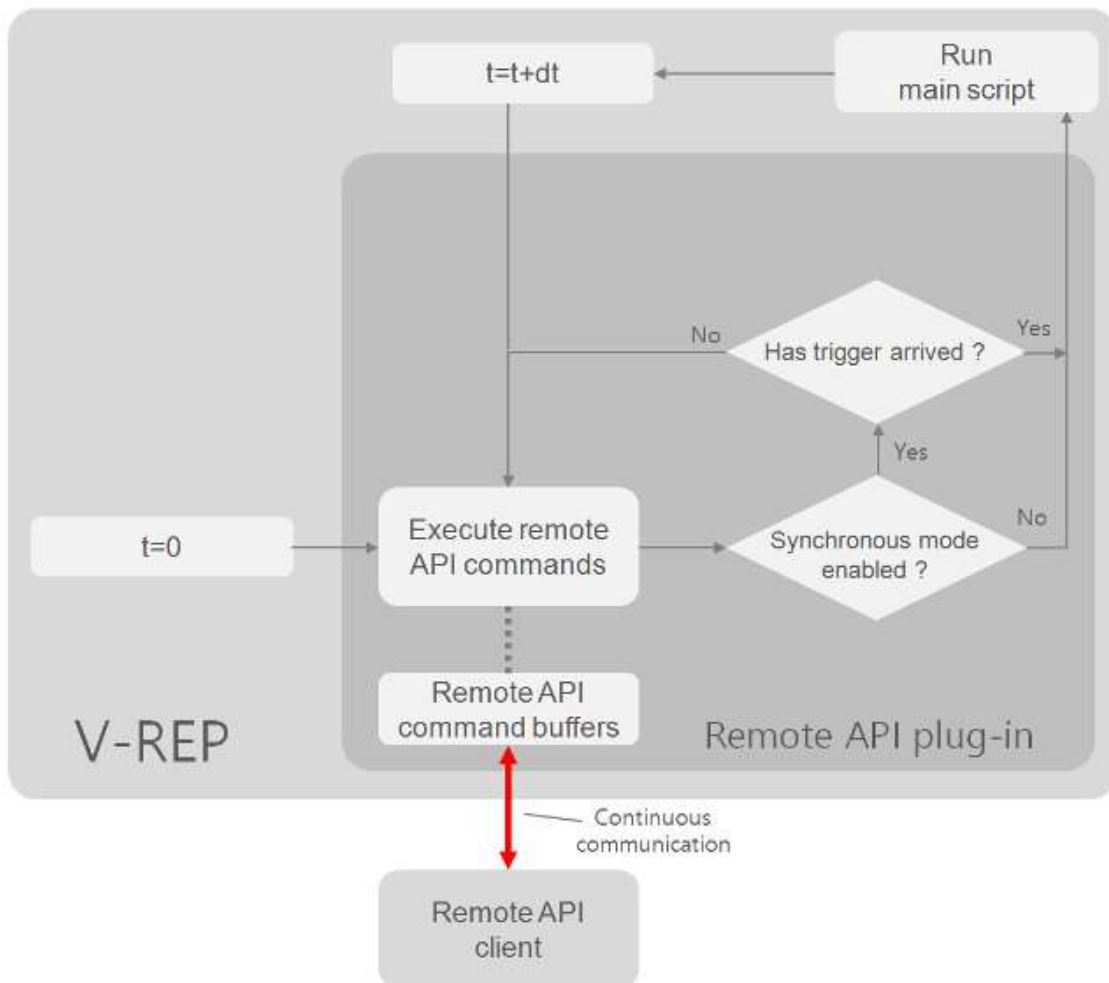
Εικόνα 191: Συγχρονισμένη Λειτουργία

Κατά την κλήση της `simxSynchronousTrigger`, στο επόμενο βήμα της προσομοίωσης θα ξεκινήσει ο υπολογισμός. Αυτό δεν σημαίνει ότι όταν επιστρέψει η κλήση λειτουργίας στο επόμενο βήμα της προσομοίωσης θα έχει ολοκληρωθεί και ο υπολογισμός της εντολής που εστάλη. Για το λόγο αυτό θα πρέπει να είμαστε σίγουροι ότι διαβάζονται τα σωστά δεδομένα. Εάν δεν ληφθεί κάποιο ειδικό μέτρο, μπορεί να διαβάζονται δεδομένα από ένα προηγούμενο βήμα προσομοίωσης, ή και από το τρέχον βήμα προσομοίωσης.

Όταν υπάρχουν πολλοί απομακρυσμένοι πελάτες API που ο κάθε ένας έχει την ανάγκη να στείλει ένα σήμα για να ξεκινήσει το επόμενο βήμα προσομοίωσης, τότε καλό είναι να γίνει χρήση μίας μη-threaded υπορουτίνας στην προσομοίωση σας. Στο κώδικα αυτό απλά πρέπει η τελευταία εντολή ροής να είναι για την επανάληψη του σήματος, αλλιώς η επανάληψη δεν θα είναι η τελευταία ενημερωμένη τιμή.



Το παρακάτω διάγραμμα απεικονίζει τον τρόπο που οι απομακρυσμένες εντολές API αντιμετωπίζονται από την πλευρά του διακομιστή (δηλαδή στην V-rep Remote API plugin πλευρά):

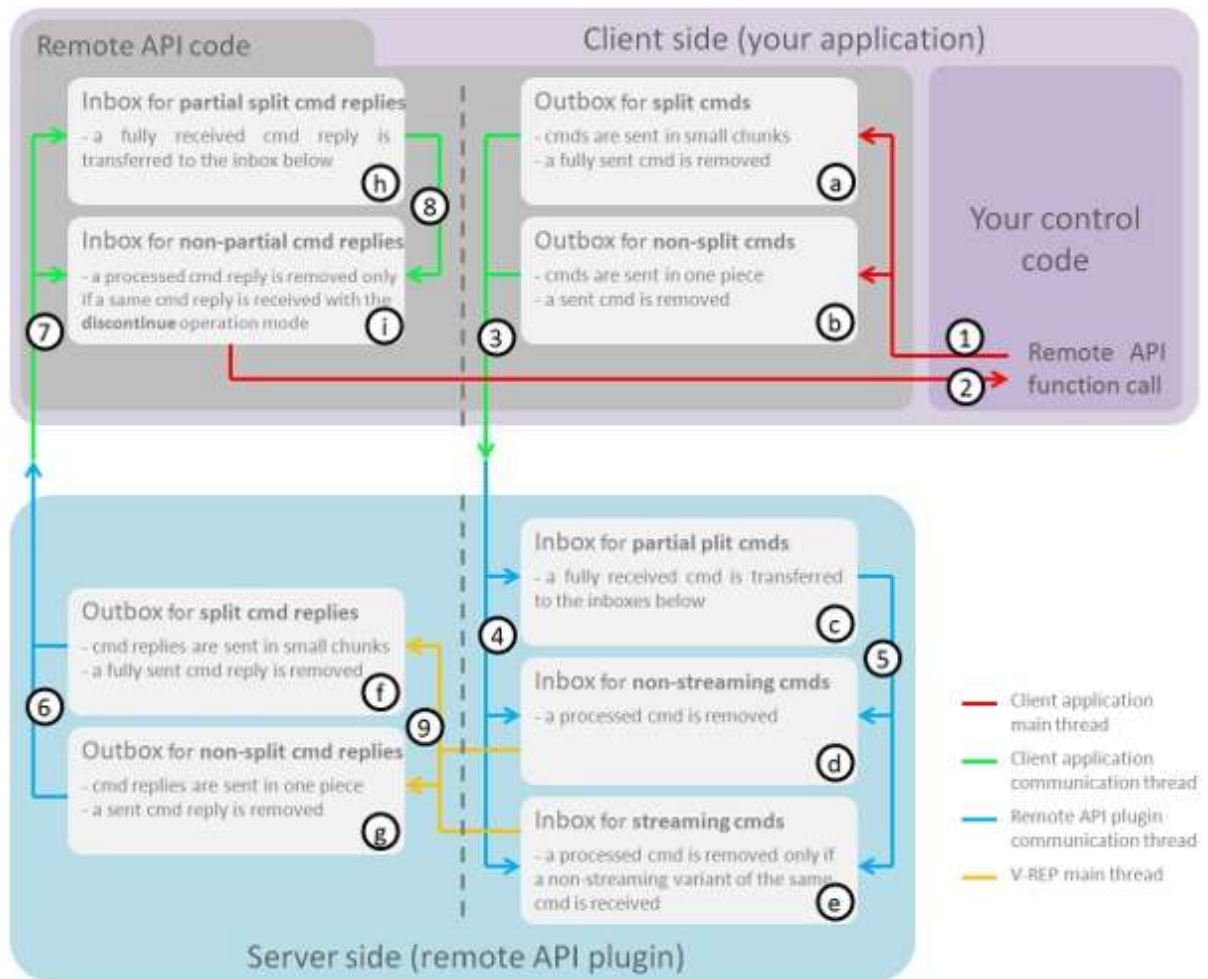


### Πρόσθετες λεπτομέρειες

Στη μεριά του πελάτη (δηλαδή την αίτησή σας), τουλάχιστον 2 νήματα θα πρέπει να τρέχουν:

- το κύριο νήμα (το ένα από το οποίο θα καλείτε το Remote API functions) και
- το νήμα επικοινωνίας (αυτό που θα χειρίζεται τις μεταφορές δεδομένων πίσω από το σκηνές).

Μπορεί να υπάρχουν όσα νήματα επικοινωνίας (δηλ. γραμμές επικοινωνίας) απαιτούνται από την πλευρά του πελάτη: Φροντίστε να καλέσετε `simxStart` για κάθε ένα από αυτά. Η πλευρά του διακομιστή, η οποία υλοποιείται με ένα plugin V-rep, λειτουργεί με παρόμοιο τρόπο. Το παρακάτω σχήμα απεικονίζει την απομακρυσμένη μέθοδο λειτουργίας API:



Εικόνα 192: Απομακρυσμένη μεθοδος λειτουργία API

Παρακάτω περιγράφονται οι διάφοροι τρόποι λειτουργίας που υποστηρίζονται και φαίνονται μέσω τον παραπομπών στο παραπάνω σχήμα :

1. **simx\_opmode\_oneshot**: λειτουργία μη αποκλεισμού.

Μια εντολή αποστέλλεται στο διακομιστή για να εκτελεστεί (1)-(b)-(3). Μια απάντηση από μια ίδια εντολή, η οποία είχε προηγούμενος εκτελεστεί, επιστρέφεται από ένα τοπικό buffer, εφόσον υπάρχει (i)-(2).

Η λειτουργία δεν αναμένει απάντηση από το διακομιστή (7)-(i). Από την πλευρά του διακομιστή, η εντολή αποθηκεύεται προσωρινά (4)-(d) και εκτελείται μία φορά (d)-(9)-(g) και μια απάντηση χει στάλει πίσω (g)-(6).

Αυτή η λειτουργία χρησιμοποιείται συχνά ως "set-functions" (π.χ. `simxSetJointPosition`), όπου ο χρήστης δεν ενδιαφέρεται για την τιμή επιστροφής.

2. **simx\_opmode\_blocking**: λειτουργία αποκλεισμού.

Μια εντολή αποστέλλεται στο διακομιστή για εκτέλεση (1)-(β)-(3), και η εντολή αυτή περιμένει την απάντηση του διακομιστή (7)-(i)-(2).

Στη συνέχεια, η απάντηση θα διαγραφεί από το buffer εισερχομένων (i), το οποίο δεν συμβαίνει με το άλλο operationmodes.

Από την πλευρά του διακομιστή, η εντολή αποθηκεύεται προσωρινά (4)-(d), εκτελείται μία φορά (d)-(9)-(ζ) και μια απάντηση στάλθηκε πίσω (g)-(6).

Αυτό η κατάσταση λειτουργίας χρησιμοποιείται με "λειτουργίες ανάληψης" (π.χ. simxGetObjectHandle), όπου ο χρήστης απαιτεί απάντηση στην εντολή που στάλθηκε.

3. **simx\_opmode\_streaming**: λειτουργία χωρίς αποκλεισμούς.

Μια εντολή αποστέλλεται στο διακομιστή για εκτέλεση (1)-(β)-(3). Μια απάντηση σε μια ίδια εντολή, που εκτελέστηκε προηγουμένως, επιστρέφεται από ένα τοπικό buffer, εάν είναι διαθέσιμο (i)-(2). Η λειτουργία δεν αναμένει απάντηση από το διακομιστή (7)-(i).

Similar `simx_opmode_oneshot`, αλλά με τη διαφορά ότι η εντολή θα αποθηκευτεί στην πλευρά του διακομιστή (4)-(e) και εκτελείται συνεχώς (e)-(9) και συνεχώς αποστέλλονται πίσω στον πελάτη (g)-(6). Αυτός ο τρόπος χρησιμοποιείται συχνά με «πάρε μια ενέργεια από συνάρτηση» (π.χ. `simxGetJointPosition`), όπου ο χρήστης απαιτεί μια συγκεκριμένη αξία συνεχώς.

4. **simx\_opmode\_oneshot\_split** (δεν συνιστάται): λειτουργία χωρίς αποκλεισμούς.

Μια εντολή αποστέλλεται σταδιακά (σε μικρά κομμάτια δεδομένων) για την εκτέλεση στον διακομιστή (1)-(α)-(3). Μια απάντηση σε μια ίδια εντολή, η οποία προηγουμένως έχει εκτελεστεί, επιστρέφεται από ένα τοπικό buffer, εάν υπάρχει (i)-(2). Η συνάρτηση δεν περιμένει για μια απάντηση από το διακομιστή (7)-(h)-(i). Όταν η εντολή σταλεί πλήρως, καταργείται από (α).

Στην πλευρά του διακομιστή, τα κομμάτια της εντολής αποθηκεύονται προσωρινά (4)-(γ) και όταν η εντολή ληφθεί πλήρως, η εντολή θα εκτελεστεί μία φορά (5)-(d)-(9) και μια απάντηση σταδιακά θα αποστέλλει πίσω στον πελάτη (f)-(6).

Ο πελάτης λαμβάνει την απάντηση σε μικρά κομμάτια (7)-(η) και όταν η απάντηση είναι πλήρης, την αποθηκεύει στο τοπικό buffer (8)-(i). Αυτή η λειτουργία χρησιμοποιείται συχνά με "set-functions" που σχετίζονται με μεγάλες ποσότητες δεδομένων (π.χ. `simxSetVisionSensorImage`), προκειμένου να μην υπερφορτώσει το δίκτυο επικοινωνίας.

5. **simx\_opmode\_streaming\_split** (δεν προτείνεται): λειτουργία χωρίς αποκλεισμούς.

Μια εντολή αποστέλλεται σταδιακά (σε μικρά κομμάτια δεδομένων) για να εκτελεστεί στον διακομιστή (1)-(α)-(3). Μια απάντηση σε μια ίδια εντολή που έχει προηγουμένως εκτελεστεί, επιστρέφεται από ένα τοπικό buffer, εάν υπάρχει (i)-(2). Η συνάρτηση δεν περιμένει απάντηση από το διακομιστή (7)-(h)-(i). Όταν η εντολή σταλεί πλήρως, καταργείται από (α).

Από την πλευρά του διακομιστή, τα κομμάτια της εντολής προσωρινά αποθηκεύονται (4)-(γ) και όταν η εντολή ελήφθη πλήρως (5)-(e), η εντολή θα εκτελεστεί συνεχώς (e)-(9)-(f) και μια απάντηση θα σταλεί σταδιακά πίσω

στον πελάτη (f)-(6). Ο πελάτης λαμβάνει την απάντηση σε μικρά κομμάτια (7)-(h) και όταν η απάντηση είναι πλήρης, αποθηκεύεται στο τοπικό buffer (8)-(i).

Αυτή η λειτουργία χρησιμοποιείται συχνά με "λειτουργίες ανάληψης" που σχετίζονται με μεγάλες ποσότητες δεδομένων (π.χ. `simxGetVisionSensorImage`), όπου ο χρήστης απαιτεί δεδομένα συνεχώς χωρίς να υπερφορτώνεται το δίκτυο επικοινωνίας.

6. **simx\_opmode\_discontinue**: λειτουργία μη αποκλεισμού.

Μια εντολή αποστέλλεται στο διακομιστή (1)-(β)-(3). Μια απάντηση από μια ίδια εντολή που εκτελέστηκε προηγουμένως επιστρέφεται από ένα τοπικό buffer, εάν υπάρχει (i)-(2). Η λειτουργία δεν αναμένει απάντηση από το διακομιστή (7)-(i).

Στο διακομιστή η εντολή απλά καθαρίζει μια παρόμοια εντολή που βρίσκεται στο (ε). Μια απάντηση αποστέλλεται στον πελάτη (6)-(7) που θα καθαρίσει επίσης μια παρόμοια απάντηση που βρίσκεται στο (i).

Αυτή η λειτουργία χρησιμοποιείται για να απελευθερωθεί κάποια μνήμη στο (i) (παρόμοια με `simx_opmode_remove`), ή να διακόψει τις εντολές ροής (την κατάργησή τους από (ε)).

7. **simx\_opmode\_buffer**: λειτουργία χωρίς αποκλεισμούς.

Καμία εντολή δεν αποστέλλεται στο διακομιστή, αλλά λαμβάνετε μια απάντηση από μια ίδια εντολή, που προηγουμένως εκτελείται από ένα τοπικό buffer, εάν υπάρχει (i)-(2). Αυτή η λειτουργία χρησιμοποιείται συχνά σε συνδυασμό με το `simx_opmode_streaming` ή `simx_opmode_streaming_split` λειτουργία: πρώτον, μια σταθερή εντολή στέλνεται και εκτελείται με μια εντολή ροής και τότε μόνο απαντήσεις από την εντολή λαμβάνονται.

8. **simx\_opmode\_remove**: λειτουργία μη αποκλεισμού.

Καμία εντολή δεν στέλνεται στο διακομιστή, αλλά μόνο μια απάντηση από μια ίδια εντολή που προηγουμένως είχε εκτελεστεί καθαρίζεται από το τοπικό buffer, αν υπάρχει (i). Η συνάρτηση δεν επιστρέφει τιμές, εκτός από έναν κωδικό επιστροφής.

Αυτή η λειτουργία μπορεί να χρησιμοποιηθεί για να απελευθερώσει κάποια μνήμη από την πλευρά του πελάτη, αλλά είναι σπάνια απαραίτητη.