



Τεχνολογικό
Εκπαιδευτικό
Ίδρυμα
Κρήτης

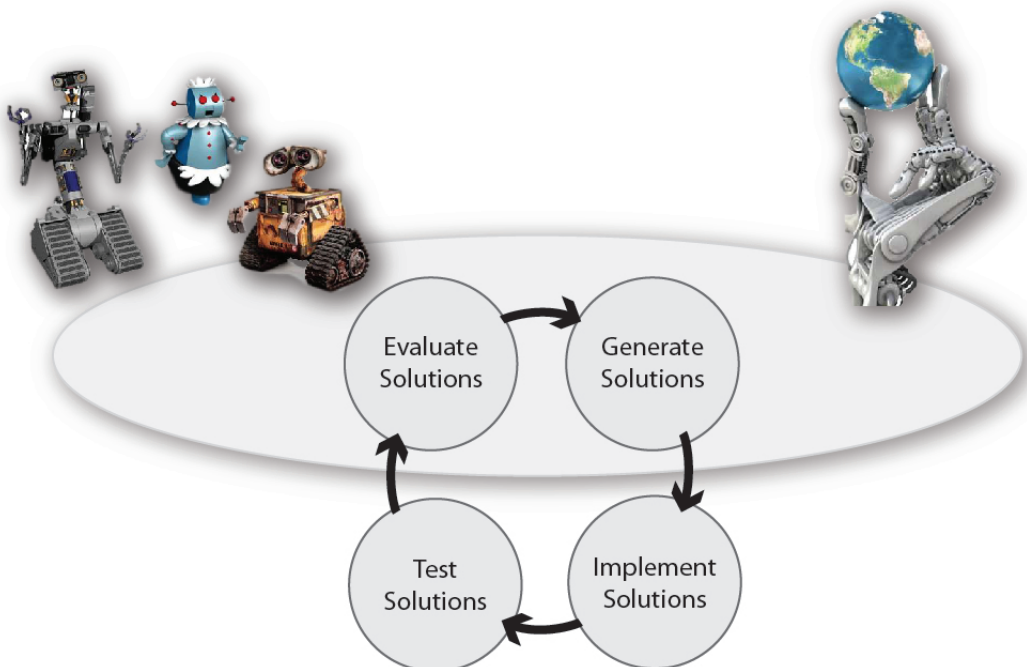
ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ρομποτικά Οχήματα και Περιβάλλοντα Προσομοίωσης

Συγγραφή-Επιμέλεια:
Ιωάννης ΣΟΡΚΟΣ

Επιβλέπων Καθηγητής:
Δρ. Ελευθέριος ΔΟΪΤΣΙΔΗΣ



15 Δεκεμβρίου 2014

Ευχαριστώ θερμά την Ελένη Λιονάκη.

Περιεχόμενα

1	Ρομποτική-Robotics	7
1.0.1	Εισαγωγή	7
2	Έντροχα ρομποτικά οχήματα.	11
2.1	Γενικά.	11
2.2	Δίτροχα οχήματα.	12
2.3	Οχήματα με τρεις τροχούς.	12
2.4	Οχήματα με τέσσερις τροχούς.	13
2.5	Οχήματα με περισσότερους απο 5 τροχούς.	15
2.6	Omni-Wheels.	16
3	KheperaIII.	17
3.1	Γενικά χαρακτηριστικά.	17
3.1.1	Κινητήρες και έλεγχος.	18
3.1.2	Αισθητήρες.	18
4	Software.	21
4.1	V-rep.	21
4.1.1	Χώρος εργασίας-χειρισμός.	22
5	Ασαφής Λογική	25
5.1	Η έννοια της Ασαφούς Λογικής- <i>Fuzzy Logic</i>	25
5.1.1	Εισαγωγή.	25
5.1.2	Ασάφεια και Λογική.	25
5.1.3	Δομή.	27
6	Υλοποίηση	30
6.1	Κίνηση ρομποτικού οχήματος σε καθορισμένα σημεία.	30
6.1.1	Κίνηση του οχήματος σε δοσμένο σημείο.	31
6.2	Σχεδίαση Ασαφούς Ελεκτηή.	32
7	Παράρτημα	39

1 Ρομποτική-Robotics

1.0.1 Εισαγωγή

Η ρομποτική είναι ένας επιστημονικός κλάδος που βρίσκεται στην τομή πολλών διαφορετικών επιστημονικών πεδίων. Σαν παραδείγματα, μόνο, μπορούμε να αναφέρουμε τη μηχανολογία, την ηλεκτρονική και την επιστήμη υπολογιστών. Η ρομποτική ασχολείται με το σχεδιασμό, την κατασκευή, τη λειτουργία και τις εφαρμογές των robots.

Η ρομποτική ξεφεύγει από τα γνωστικά πεδία των παραπάνω επιστημών και προκαλεί το ενδιαφέρον επιστημονικών πεδίων όπως η ανθρωπολογία και η ψυχολογία.

Ερωτήματα όπως το τι είναι ένα robot· ποιά είναι τα χαρακτηριστικά του και ποιές οι χρήσεις του επιδέχονται πολλές και διαφορετικές απαντήσεις.

Το robot υπονοεί τη μηχανική κατασκευή καθώς και την απουσία του ανθρώπινου-έμφυχου στοιχείου. Το μηχανικό ανθρωποειδές είναι μια ιδέα που ξεκινάει από πολύ νωρίς στην ιστορία του ανθρώπου. Σε παραμύθια, θρύλους και μύθους εμφανίζονται μεταλλικές κατασκευές, φτιαγμένες να υπηρετούν ή να τιμωρούν.



Σχήμα 1: Αφίσα από την πρώτη παράσταση του έργου Rossum's Universal Robots-1920.

Το όνομα ROBOT ανήκει στον Τσέχο συγγραφέα *Karel Čapek*. Εμφανίστηκε πρώτη φορά στο θεατρικό δράμα *Rossum's Universal Robots* το 1920. Στο θεατρικό του K.Čapek, συμβαίνει το εξής παράδοξο. Στο έργο φαίνεται να ορίζει με ακρίβεια τη "ρομποτική συμπεριφορά" παρόλο που δεν υπάρχει ούτε μία μηχανική κατασκευή σε ολόκληρο το κείμενο.

Πρώτη απόπειρα μηχανικής κατασκευής, που πλησιάζει τη σημερινή έννοια του robot ανήκει στον *Leonardo da Vinci* και στο *Mechanical Knight*, του 1495.

Σημεία αναφοράς στην ιστορία των ρομποτικών κατασκευών, είναι ο Ούγγρος *Wolfgang von Kempelen*, που κατασκεύασε μια αυτόματη μηχανική κατασκευή που έπαιζε σκάκι το 1770, με σκοπό να εντυπωσιάσει την Αυτοκράτειρα Μαρία Τερέζα της Αυστρίας. Η συσκευή του W.von Kempelen έμεινε γνωστή ως "Ο Τούρκος" (*Schachtürke*- Σχήμα 2β).

Επίσης ο Γάλλος *Jacques de Vaucanson* με τη μηχανική πάπια που κατασκεύασε το 1739. Ο J. de Vaucanson ίσως να μην επηρέασε όσο ήθελε τη ρομποτική σε τεχνικό επίπεδο, αλλά με το όραμα του παρακίνησε πολλούς μεταγενέστερους επιστήμονες και εφευρέτες.

Παρόλη την κατασκευαστική δυσκολία που χαρακτηρίζει αυτές τις κατασκευές, δύσκολα θα έπειθαν κάποιον ότι όντως πρόκειται για robot.

Η αλλαγή φάσης στον τρόπο που φαντάζονται οι άνθρωποι ένα robot γίνεται το 1942 από τον *Isaac Asimov*. Στο διήγημα "Runaround" (Το οποίο διαδραματίζεται το 2015) γράφει τους 3 κανόνες της ρομποτικής.

1. Ένα robot δε μπορεί να βλάψει ανθρώπινο ον, ούτε μέσω απραγίας, να επιτρέψει σε ανθρώπινο ον να πληγωθεί.
2. Ένα robot οφείλει να υπακούει τις διαταγές που του δίνονται από ανθρώπινα όντα, εκτός αν

έρχονται σε ρήξη με τον Πρώτο Νόμο

3. Ένα robot οφείλει να προστατεύει την ύπαρξή του, όσο, η προστασία αυτή, δεν έρχεται σε ρήξη με τον Πρώτο και Δεύτερο Νόμο.

Ο I. Asimov δεν κατασκεύασε κάποιο robot. Κατάφερε όμως μέσα από τη λογοτεχνία να μας προκαλέσει να οραματιστούμε μια σαφή εικόνα για αυτά. Μας έδειξε πως θα ήταν η συνύπαρξη ανθρώπων και robot, και εισήγαγε την έννοια της βούλησης και της ύπαρξης για το robot, θέτοντας προβληματισμούς πάνω σε θέματα “Δημιουργού-Δημιουργήματος” και “Ηθικής-Τεχνολογίας”.

Αυτές όμως είναι λογοτεχνικές προσεγγίσεις, ίσως, και μεμονωμένες τεχνικές απόπειρες. Με εξαίρεση τον Ιππότη του da Vinci που όντως ήταν μια σύγχρονη ρομποτική προσέγγιση εγκλωβισμένη το 1495, όλες οι άλλες κατασκευές ήταν περισσότερο παιχνίδια, παρά robot με την έννοια της “κατασκευής που εκτελεί κάποιες εργασίες”. Σίγουρα έθεσαν τις βάσεις στη φαντασία μας. Σίγουρα έδωσαν κάποιες κατασκευαστικές κατευθύνσεις. Πότε όμως γίνεται η αρχή της σύγχρονης ρομποτικής; Πότε το robot παύει να είναι παιχνίδι και γίνεται εργαλείο;

Η πρώτη προσπάθεια κατασκευής Robot με τη σημερινή προσέγγιση και σημείο αναφοράς για τη σύγχρονη ρομποτική, το robot Unimate του George Charles Devol, Jr.

Ο βραχίονας Unimate χρησιμοποιήθηκε στην εταιρία General Motors για να ταξινομεί και στιβάξει χυτό μέταλλο. Έχοντας τις εντολές αποθηκευμένες σε ένα μαγνητικό τύμπανο. Ο λόγος κατασκευής του ήταν ότι το περιβάλλον ενός χυτηρίου είναι από τα πιο επιβλαβή για την υγεία του ανθρώπου.

Η ρομποτική έχει πολύ μεγάλο εύρος πρακτικών εφαρμογών. Και μπορεί να βοηθήσει στην επίλυση πολλών και διαφορετικών προβλημάτων. Ενδεικτικά αναφέρονται οι ακόλουθες εφαρμογές.

- Ρομποτικά τεχνητά μέλη.
- Ρομποτικοί μηχανισμοί σε χώρους εργασίας με μεγάλο βαθμό επικινδυνότητας.
- Διαστημικές εφαρμογές.
- Ιατρικά μηχανήματα.
- Εκπαιδευτικές εφαρμογές.

Για να αναλύσει κανείς έστω και μία από τις προαναφερθείσες εφαρμογές της ρομποτικής θα αναγκάζοταν, σίγουρα πάνω από ένα βιβλίο. Στην παρούσα εργασία θα σταθούμε μόνο στις εκπαιδευτικές εφαρμογές. Όχι φυσικά με πρόθεση να αναλύσουμε το θέμα, αλλά έχοντας σαν στόχο να αναδείξουμε τη βαρύτητα του συγκεκριμένου τομέα και την επίδραση του.

Ας προσπαθήσουμε να εμβαθύνουμε στις πτυχές της ρομποτικής μιας και υπάρχουν πολλά διαφορετικά ήδη robot. Χρησιμοποιούνται σε πολλά διαφορετικά περιβάλλοντα, με πολύ διαφορετικό τρόπο κάθε φορά. Παρόλο που οι εφαρμογές της ρομποτικής παρουσιάζουν μεγάλες διαφορές,



(α') Μοντέλο του Mechanical Knight.
Εμφανίζεται και μια εκδοχή του εσωτερικού μηχανισμού.

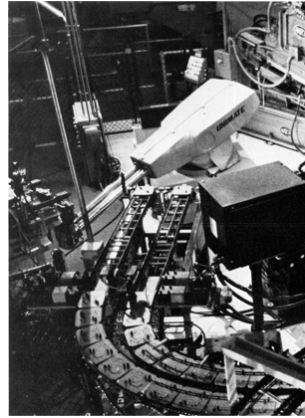


(β') Ρομποτική κατασκευή που παίζει σκάκι

Σχήμα 2: Μηχανικές ρομποτικές κατασκευές.



(α) Ρομποτικό πόδι κινούμενο με χρήση αέρα.



(β') Unimate Robot-1954



(γ') Ρομποτικός βραχίονας, KUKA ROBOTICS.



(δ') Robots με μορφή φιδιού.

Σχήμα 3: Μορφές σύγχρονων Robots.

όλες μοιράζονται τρεις βασικές ομοιότητες στην προσέγγιση.

- Όλα τα robot αποτελούν κάποιου είδους μηχανική κατασκευή με ορισμένο σχήμα, μορφή και δομή, ανάλογα με τη χρήση. Για παράδειγμα: Ας υποθέσουμε ένα robot πρέπει να μετακινείται μέσα σε παχύ χώμα ή λάσπη. Οι περισσότεροι θα σκεφτούν να χρησιμοποιήσουν ερπύστριες. Αυτό σημαίνει ότι θα χρησιμοποιηθεί η κίνηση 2×2 (Περισσότερες πληροφορίες για την 2×2 κίνηση στην Ενότητα 2). Ότι η δομή και το σχήμα του robot θα ακολουθήσουν περιορισμούς. Περιορισμούς που οφείλονται στο περιβάλλον, και σε λειτουργία σε πραγματικές συνθήκες. **Εμπλεκόμενα επιστημονικά πεδία: Μηχανική-Μηχανολογία, Φυσική Υλικών.**

- Όλες οι ρομποτικές κατασκευές έχουν μια πηγή τροφοδοσίας, κάποιους αισθητήρες, μπαταρίες, κυκλώματα και καλώδια. Στο παράδειγμά μας κάποιος κινητήρας θα κινεί τις ερπύστριες, θα υπάρχουν μετρητικά όργανα για τις εκάστοτε ενδείξεις κλπ. Η ηλεκτρική/ηλεκτρονική πτυχή θα αρχίσει να κινεί την μηχανική κατασκευή.

Εμπλεκόμενα επιστημονικά πεδία: Ηλεκτρολογία-Ηλεκτρονική και Χημική Μηχανική.

- Όλα τα robot δέχονται κάποιες εντολές με μορφή κώδικα. Προγραμματισμός. Με αυτό τον τρόπο το robot μπορεί να παίρνει αποφάσεις. Ας υποθέσουμε στο παραδειγμά μας, ότι η κατασκευή μας είναι άρτια κατασκευασμένη και ότι έχει τα σωστά επίπεδα ενέργειας για την τροφοδοσία του.

Γιατί δεν κάνει τίποτα; Τι κινεί το robot; Ακόμα και να χρησιμοποιούμε ένα τηλεχειριστήριο με μοχλό και απλά να πιέζουμε μπροστά το μοχλό για να προχωρήσει το οχήμά μας, το robot χρειάζεται ένα βασικό πρόγραμμα που να το δίνει την ικανότητα να "ύπακούει".

Εμπλεκόμενα επιστημονικά πεδία: Μαθηματικά, Προγραμματισμός, Τεχνητή Νοημοσύνη.

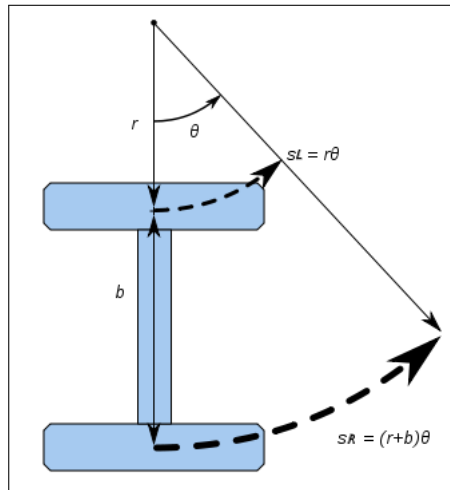
Στη βάση του όμως το robot είναι σαν μία κατασκευή, σαν ένα παιδικό παιχνίδι που χρειάζεται συναρμολόγηση. Έχει βήματα που αν τα ακολουθήσεις μπορείς να ολοκληρώσεις την κα-

2 Έντροχα ρομποτικά οχήματα.

2.1 Γενικά.

Διαφορικό είναι ο μηχανισμός που σκοπό έχει να επιτρέπει στους κινητήριους τροχούς να περιστρέφονται με διαφορετικές γωνιακές ταχύτητες και κατανέμει τη ροπή στρέψης, ανάλογα με τον συντελεστή πρόσφυσης των κινητήριων τροχών.

Στα έντροχα ρομποτικά οχήματα συναντάμε διάφορους αριθμούς τροχών, ανάλογα με τις ανάγκες πλοήγησης και της χρήσης του οχήματος. Παρόλαυτά για την κίνηση του οχήματος χρησιμοποιούμε τη μέθοδο της διαφορικής κίνησης των τροχών ή πιο σύντομα του διαφορικού.



Σχήμα 5: Διαφορική κίνηση ανάμεσα σε αριστερό και δεξί τροχό.

Η προσέγγιση της κίνησης ενός ρομποτικού οχήματος με τροχούς και όχι με πέλματα ή πόδια, επιτρέπει ευκολότερο προγραμματισμό, σχεδιασμό και κατασκευή, έχοντας σαν προϋπόθεση την κίνηση του οχήματος σε σχετικά λείες επιφάνειες, χωρίς μεγάλες κλίσεις ή έντονα εξογκώματα.

Προφανώς το μειονέκτημα που παρουσιάζουν τα έντροχα ρομποτικά οχήματα είναι οι δυσκολίες προκύπτουν στην πλοήγηση του οχήματος, σε επιφάνειες που ξεφεύγουν από τους περιορισμούς που αναφέρουμε παραπάνω.

Στα έντροχα ρομποτικά οχήματα συναντάμε διάφορους αριθμούς τροχών, ανάλογα με τις ανάγκες πλοήγησης και των χρήσεων του οχήματος.

Η αρχή λειτουργίας του διαφορικού είναι ένα μηχανικό σύστημα που επιτρέπει τους τροχούς ενός οχήματος να κινούνται με διαφορετική ταχύτητα, γεγονός που μας επιτρέπει να στρίβουμε το όχημα προς την κατεύθυνση που επιθυμούμε.

Παρατηρώντας το (Σχήμα 5) βλέπουμε ότι για να στρίψει το σύστημα των τροχών αριστερά, πρέπει ο αριστερός τροχός να κινηθεί κατά μία απόσταση:

$$S_L = r\theta$$

ενώ ο δεξιός τροχός πρέπει να κάνει απόσταση μεγαλύτερη όσο είναι το μεταξόνιο των δύο τροχών [b],

$$S_R = (r + b)\theta$$

Συμπεραίνουμε λοιπόν ότι για να στρίψει ένα όχημα πρέπει οι τροχοί του να κινηθούν με διαφορετικές ταχύτητες, ώστε να καταλήξουν στο ίδιο σημείο διανύοντας διαφορετικές αποστάσεις. Οι απόσταση που διανύει ο κάθε τροχός είναι το μήκος του τόξου που εμφανίζεται με διακεκομμένες γραμμές στο (Σχήμα 5).

2.2 Δίτροχα οχήματα.

Μία από τις πιο απλές κατασκευές εντρόχων οχημάτων είναι αυτά με τους δύο τροχούς. Έχουμε διάφορες εκδοχές στη γεωμετρία των τροχών.



(α') Dicycle



(β') Bicycle



(γ') Tandem.jpg

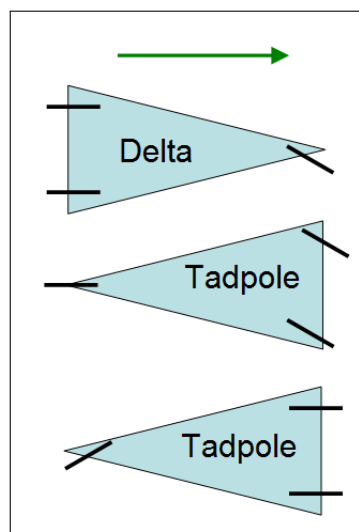
Σχήμα 6: Δίτροχα οχήματα και οι θέσεις των τροχών.

- Τροχοί τοποθετημένοι παράλληλα [dicycles, (Σχήμα 6α)]
- Τροχοί ο ένας πίσω από τον άλλο [bicycle, (Σχήμα 6β)]
- Τροχοί τοποθετημένοι επάλληλα [tandemly, (Σχήμα 6γ)].

Στα bicycles (Σχήμα 6β) και σε ορισμένα dicycles (Σχήμα 6α) για να μπορεί το όχημα να βρίσκεται σε όρθια θέση πρέπει να κινείται, ώστε το κέντρο μάζας του οχήματος να μένει μέσα στα όρια της βάσης του. Συγκεκριμένα στα dicycles (Σχήμα 6α) φροντίζουμε το κέντρο βάρους του οχήματος να βρίσκεται κάτω από τη βάση του (δλδ: το επίπεδο που ανάμεσα στους δύο τροχούς και κάτω από τη θέση του αναβάτη). Επίσης για λόγους βελτίωσης της στατικότητας συνήθως η μπαταρία τοποθετείται στη βάση του οχήματος.

2.3 Οχήματα με τρεις τροχούς.

Αρχικά, η δομή ενός τέτοιου οχήματος ακολουθεί κάποια από τις τοπολογίες που φαίνεται στο (Σχήμα 7).



Σχήμα 7: Τοπολογίες των τροχών σε οχήματα τριών τροχών.

Παρατηρούμε ότι τα οχήματα με 3 τροχούς έχουν αρκετούς τρόπους στη χρήση και τοποθέτηση των τροχών τους. Αυτό έχει να κάνει κυρίως με τις συνθήκες στις οποίες θέλουμε να χρησιμοποιήσουμε το όχημα, αλλά και από ενδεχόμενους κατασκευαστικούς και οικονομικούς περιορισμούς που θα προκύψουν.

Τα ρομποτικά οχήματα με 3 τροχούς χωρίζονται σε 2 μεγάλες κατηγορίες.

1. Δύο κινητήριους τροχούς και ο τρίτος τροχός έχει ρόλο στήριξης.

Σε αυτά τα οχήματα οι δύο κινητήριοι τροχοί χρησιμοποιούν τον τρόπο λειτουργίας της διαφορικής κίνησης που περιγράφουμε στην αρχή της ενότητας. Επομένως έχουμε δύο κινητήριους τροχούς, ελεγχόμενους από δυο κινητήρες/ελεκτές και έναν ελεύθερο τροχό για λόγος στήριξης του οχήματος.

2. Οχήματα που οι δύο τροχοί τροφοδοτούνται από έναν κινητήρα, ενώ ο τρίτος τροχός ευθύνεται για την κατεύθυνση του οχήματος, ελεγχόμενος είτε από δεύτερο κινητήρα, είτε από κάποιο σύστημα ελέγχου κατεύθυνσης (πχ: τιμόνι, πηδάλιο).

Τα πλεονεκτήματα των οχημάτων με τρεις τροχούς είναι η σχετικά καλή στατικότητα και κατευθυντικότητα του οχήματος, τηρώντας βέβαια κάποιες προϋποθέσεις, όπως η τραχύτητα της επιφάνειας του εδάφους και η κλίση του, καθώς και το αν υπάρχουν μεγάλες ανομοιομορφίες (πχ: πέτρες, λάσπη κλπ.). Πλεονέκτημα μπορεί να θεωρηθεί και το χαμηλότερο κόστος κατασκευής σε σχέση με τα οχήματα τεσσάρων τροχών.

2.4 Οχήματα με τέσσερις τροχούς.

Αρχικά θα παρουσιάσουμε μερικές από τις πιο συνηθισμένες περιπτώσεις οχημάτων με τέσσερις τροχούς και τους τρόπους κίνησης/πλοήγησης.

1. Δύο κινητήριοι τροχοί και δύο ελεύθεροι.

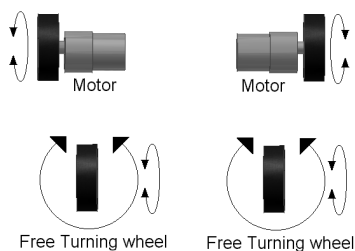
Αυτή η δομή μοιάζει ουσιαστικά με ένα δίτροχο όχημα (dicycle) στο οποίο προσπαθήσαμε να εξαλείψουμε τα προβλήματα στατικότητας, τα οποία υπήρχαν, με ένα τρόπο τεχνικά, κατασκευαστικά και οικονομικά, εύκολο.

Όπως φαίνεται και στο (Σχήμα 8α) το όχημα αποτελείται από δύο κινητήριους τροχούς, που περιστρέφονται με διαφορετική κίνηση, και δύο ελεύθερους τροχούς ή τροχούς στήριξης, που απλώς στηρίζουν το όχημα και στρέφονται αναλόγως με την κίνηση του οχήματος.

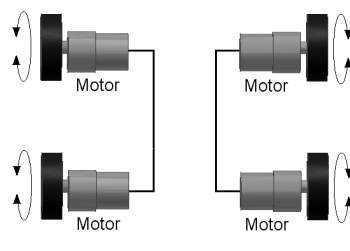
2. Δύο επί Δύο (2x2).

Στην περίπτωση αυτή οι τροχοί κινούνται ανα δύο, από την ίδια πηγή κίνησης όπως φαίνεται χαρακτηριστικά στο (Σχήμα 8β). Παρατηρούμε ότι οι τροχοί που βρίσκονται στην ίδια πλευρά του οχήματος έχουν την ίδια πηγή κίνησης.

Αυτή η δομή παρουσιάζει μεγάλα μειονεκτήματα σε επίπεδο υλοποίησης επειδή αναγκάζομαστε να τηρούμε πολύ αυστηρούς περιορισμούς στην τήρηση της ίδιας ταχύτητας στους τροχο-

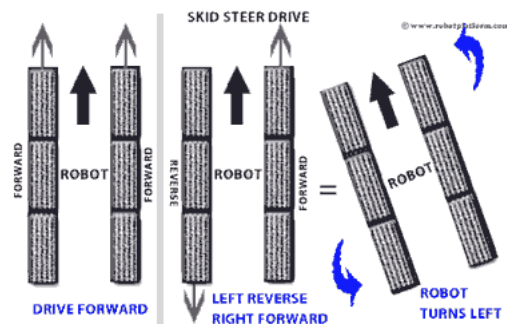


(α') Δύο κινητήριοι τροχοί-Δύο ελεύθεροι.



(β') Δύο επί Δύο 2x2

Σχήμα 8: Διαφορετικοί τρόποι κίνησης οχημάτων με τέσσερις τροχούς.



Σχήμα 9: Skid Steer Drive-Τρόπος πλοήγησης 2x2.

ύς του οχήματος. Το πρόβλημα που προκύπτει στην πράξη είναι ότι η πρόσφυση των τροχών μπορεί να έχει μικροδιαφορές, που οφείλονται σε μικροδιαφορές υψής της επιφάνειας, πάνω στην οποία κινείται το όχημα. Αυτή η διαφορά στην κίνηση κάνει τον «πιο αργό» τροχό να «σέρνεται» (ολισθαίνει) μη επιτρέποντας έτσι, την ευθεία κίνηση του οχήματος.

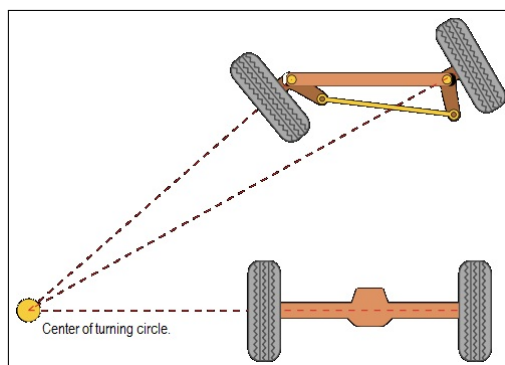
Παρόλα αυτά η 2x2 κίνηση βρήκε εφαρμογή στην κατασκευή αρμάτων μάχης (ερπυστριοφόρα ή τροχοφόρα), και εκεί οφείλει τη μεγάλη αναγνώριση που έχει αυτός ο τρόπος πλοήγησης. Ο όρος που χρησιμοποιείται είναι *skid steer drive* που μπορούμε να τον αποδόσουμε ως πλοήγηση μέσω ολίσθησης, παίρνοντας την ονομασία από την αίσθηση, ότι το όχημα δεν στρίβει, αλλά ολισθαίνει προς την κατεύθυνση που επιθυμούμε. Προσπαθώντας λίγο να γενικεύσουμε την 2x2 κίνηση να σημειώσουμε ότι πέρα από το μεγάλο αριθμό κινητήρων που χρειάζεται αυτή η δομή για να λειτουργεί (υπολογίζοντας ότι δεν σταματάμε απαραίτητα στους 4 τροχούς, αλλά μπορεί να έχουμε 6 ή 8 ή ερπύστριες), η 2x2 κίνηση μπορεί να προκαλέσει προβλήματα στις δυνατότητες πλοήγησης του οχήματος.

Επίσης έχει έναν περιορισμένο αριθμό επιφανειών που μπορεί να κινήσει και αυτό γιατί επιφάνειες με μεγάλες ανωμαλίες αυξάνουν τη διαφορά ταχύτητας στους τροχούς της ίδιας πηγής κίνησης, λόγω διαφοράς πρόσφυσης μεταξύ των τροχών. Με αποτέλεσμα το όχημα να μην κινείται ευθεία.

Προσπαθώντας να εξαλείψουμε τα μειονεκτήματα της 2x2 κίνησης καταλλήγουμε στην τρίτη κατηγορία κίνησης οχημάτων με τέσσερις τροχούς.

3. Γεωμετρία στρέψης Ackermann.

Όταν ένα όχημα επιχειρεί να στρίψει προς κάποια κατεύθυνση, αρχικά οι τροχοί διαγράφουν τόξα κύκλων με διαφορετική ακτίνα, έτσι ο εσωτερικός τροχός διανύει μικρότερη απόσταση (τόξο κύκλου) από ότι ο εξωτερικός. Αυτό έχει σαν αποτέλεσμα το όχημα να μην καταφέρνει να αλλάξει κατεύθυνση αποτελεσματικά.



Σχήμα 10: Γεωμετρία στρέψης Ackermann

Ο R. Ackermann έλυσε αυτό το πρόβλημα σχεδιάζοντας ένα σύστημα στρέψης που ο κάθε τροχός διαγράφει τόξο κύκλου διαφορετικής ακτίνας. Με τη βοήθεια του (Σχήματος 10) παρατηρούμε ότι ο εσωτερικός τροχός είναι περισσότερο στριμμένος απο τον εξωτερικό.

Όσο δύσκολο και άν είναι να αποδώσεις τα εύσημα της καινοτομίας, είναι ωστόσο ο πιο διαδεδομένος και γνώριμος τρόπος κίνησης-πλοήγησης ενός οχήματος, και τα σημερινά αυτοκίνητα χρησιμοποιούν αυτό τον τρόπο με λίγες αλλαγές που δεν επηρεάζουν ουσιαστικά τη δομή, απλώς βελτιώνουν και εξομαλύνουν τη λειτουργία της.

Τα ρομποτικά οχήματα που χρησιμοποιούν αυτό τον τρόπο κίνησης-πλοήγησης είναι μια μικρογραφία του τρόπου κίνησης του αυτοκινήτου. Δύο τροχοί κίνησης, λοιπόν, που χρησιμοποιούν διαφορεικό τρόπο μετάδοσης και δύο τροχοί πλοήγησης που ελέγχονται απο κάποιο έλεγχο πλοήγησης (πχ:τιμόνι). Στα ρομποτικά οχήματα το τιμόνι, συνήθως είναι ένας σερβομηχανισμός (*servomechanism*) που ελέγχει την κατεύθυνση των τροχών. Μία δύσκολη δομή για υλοποίηση μιας και συνδιάζει μετάδοση μέσω διαφορικού, απο μία πηγή κίνησης, έχοντας δύο ανεξάρτητους άξονες, έναν για ταχύτητα και έναν για κατεύθυνση κύλισης του οχήματος.

Ωστόσο ένα μεγάλο πλεονέκτημα του συγκεκριμένου συστήματος σε σχέση με τις άλλες δομές τετράτροχων οχημάτων είναι όταν το όχημά μας χρησιμοποιεί *κινητήρα εσωτερικής καύσης* γιατί μπορούμε να έχουμε μεγάλη οικονομία σε κινητήρες, μιας και μπορούμε να μεταδώσουμε κίνηση σε όλους τους τροχούς απο τον *κινητήρα εσωτερικής καύσης*. Καθώς και κάποιο μηχανισμό για τον έλεγχο κατεύθυνσης.

“Η ανακάλυψη αποδίδεται στον Georg Lankensperger στο Μόναχο το 1817 και αργότερα κατοχύρωσε την πατέντα ο Rudolph Ackermann το 1818, στην Αγγλία, για οχήματα που τα έσερναν άλογα. Διεκδικητής της ανακάλυψης εμφανίζεται και ο Erasmus Darwin παππούς, του Charles Darwin θεμελιωτή της εξέλιξης των ειδών.”

2.5 Οχήματα με περισσότερους απο 5 τροχούς.

Ο πιο συνηθισμένος αριθμός τροχών σ'αυτή την περίπτωση είναι 6. Είναι οχήματα κατασκευασμένα για πολύ συγκεκριμένες χρήσεις, λόγω της κατασκευαστικής και προγραμματιστικής δυσκολίας που παρουσιάζουν. Η κίνηση σε οχήματα με πολλούς τροχούς επιτυγχάνεται μέσω διαφορικού. Λόγω των πολλών τροχών είναι πιθανό να υπάρξει κάποια τοπική διαφορά στην ταχύτητα κάποιου τροχού, σε σχέση με τους υπόλοιπους. Η διαφορά στην ταχύτητα προκαλεί δύο προβλήματα.

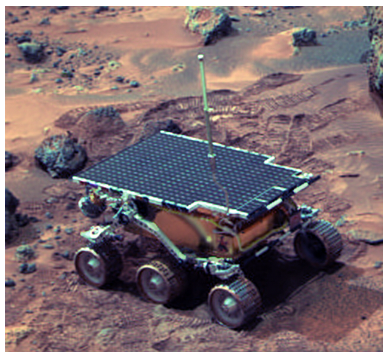
1. Στους τροχούς που βρίσκονται στην ίδια πλευρά του οχήματος, παρατηρούμε ότι ο τροχός που κινείται πιο αργά, σε σχέση με τους διπλανούς του, «ολισθαίνει».
2. Το όχημα συνολικά κινείται «πλάγια» αντί για «ευθεία».

Για να βελτιώσουμε τον έλεγχο του οχήματος προσθέτουμε συνήθως έναν ακόμη τροχό, συνήθως στην μπροστινή ή πίσω όψη του οχήματος, χρησιμοποιώντας τον για οδομετρία.

“Οδομετρία είναι η χρήση δεδομένων απο αισθητήρες κίνησης με σκοπό τον υπολογισμό της αλλαγής θέσης ενός σώματος προς το χρόνο.”

Μέσω αυτού του τροχού καταφέρνουμε να έχουμε μία πιο συνολική εικόνα για την επιφάνεια στην οποία κινείται το όχημα. Απο τη στιγμή που αυτός ο τροχός δεν συμμετέχει στην κίνηση του οχήματος, είναι ένα πολύ χρήσιμο μέσο ανάδρασης, ώστε να επιβεβαιώνουμε άν το όχημά μας κινείται στην κατεύθυνση που θέλουμε ή αν λόγω κάποιου σφάλματος κινείται σε λανθασμένη κατεύθυνση.

Στο (Σχήμα 11) παρουσιάζονται διαστημικά ρομποτικά οχήματα 6 τροχών.



(α') Sojourner.



(β') Πρωτότυπα οχήματα διαστημικών αποστολών.

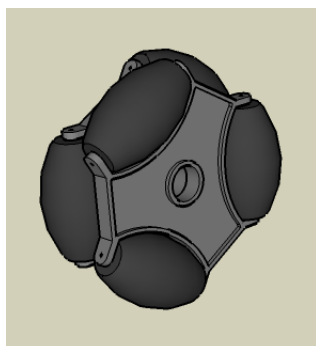


(γ') Όχημα διαστημικής αποστολής στον πλανήτη Άρη 2008.

Σχήμα 11: Οχήματα με έξι τροχούς.

2.6 Omni-Wheels.

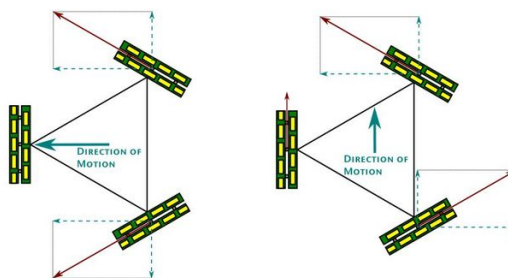
Αυτή η περίπτωση δεν έχει να κάνει με τον αριθμό τροχών του οχήματος. Αφορά ξεκάθαρα τον τροχό. Η ιδέα των *omni-wheels* είναι να δημιουργηθεί ένας τροχός από άλλους μικρότερους. Οι μικρότεροι τροχοί έχουν τον άξονα περιστροφής τους κάθετο, στον άξονα περιστροφής του κεντρικού τροχού. Στο (Σχήμα 13) οι μικροί τροχοί που έχουν σχήμα έλλειψης (Σκούρο γκρι) επιτρέπουν στο όχημα να κινείται πλάγια.



Σχήμα 12: Simple Omniwheel.

Αυτοί οι τροχοί μας επιτρέπουν να κινούμε το όχημα προς δύο κατευθύνσεις ταυτόχρονα. Άρα τελικά το όχημα θα μετακινηθεί στην συνισταμένη των κατευθύνσεων, από τα *Omniwheels*. Μπορεί επομένως να στρίβει **χωρίς** να διαγράφει τόξο από την ευθεία **πριν** τη στροφή, στην ευθεία **μετά** τη στροφή.

KIWI DRIVE ROBOT



2

Σχήμα 13: Όχημα τριών τροχών με *Omniwheels*.

Στη ρομποτική όταν ένα σύστημα μπορεί να ελέγχει τόσους βαθμούς ελευθερίας, όσους διαθέτει, ονομάζεται *Holonomic*. Στην περίπτωση των ρομποτικών οχημάτων με omni-wheels, holonomic move ονομάζεται η ικανότητα του οχήματος να κινείται στιγμιαία προς οποιαδήποτε κατεύθυνση θελήσουμε.

Ο αριθμός των τροχών στα οχήματα που χρησιμοποιούν *Omniwheels*, διαφέρει και συνθήτως έχει να κάνει με το μέγεθος του οχήματος. Οι *Omniwheels* έχουν χαμηλή απόδοση. Επίσης επηρεάζονται αρκετά από το έδαφος στο οποίο κινούνται. Αρκεί να φανταστούμε ότι οι τροχοί *Omniwheels* σπάνια κινούνται προς την κατεύθυνση που θα κινηθεί το όχημα. Κάθε ένας τροχός, σε τοπικό επίπεδο, πραγματοποιεί πολλές κινήσεις ταυτόχρονα, αλλά το όχημα τελικά καταλήγει να κατευθυνθεί σε μια διαφορετική πορεία. Στο (Σχήμα 12) συμβολίζουμε με κόκκινα βέλη την κίνηση των τροχών, ενώ με μπλέ δηλώνουμε την τελική κατεύθυνση του οχήματος.

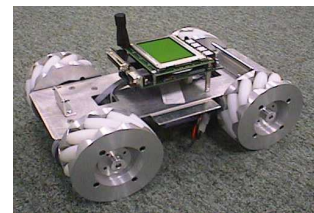
Στο (Σχήμα 14) παρουσιάζονται οχήματα εφοδιασμένα με omniwheels.



(α') Όχημα οκτώ τροχών εφοδιασμένο με *Omniwheels* και βραχίονα εργασίας (KUKA).



(β') *Omniwheel*-Τα κίτρινα βέλη υποδεικνύουν τις πορείες που μπορεί να μετακινηθεί το όχημα.



(γ') Όχημα τεσσάρων τροχών εφοδιασμένο με *Omniwheels*.

Σχήμα 14

3 KheperaIII.

3.1 Γενικά χαρακτηριστικά.

Το *KheperaIII* είναι ένα μικρό *robot* που αναπτύχθηκε από μία ομάδα Ελβετών στη δεκαετία του '90 και μάλιστα είχε και μεγάλη εμπορική επιτυχία σε ερευνητικούς και εκπαιδευτικούς κύκλους. Μερικοί από τους λόγους που οδήγησαν στην επιτυχία τόσο κατασκευαστικά αλλά και εμπορικά θα αναπτυχθούν στη συνέχεια.

1. Αισθητήρες υπερύθρων.
2. Αισθητήρες υπερήχων.
3. Σειριακή θύρα.
4. Mini-USB θύρα.
5. Θύρα τροφοδοσίας.
6. Reset.
7. Διακόπτης On/Off.
8. Ροδέλα καλωδίου.

3.1.1 Κινητήρες και έλεγχος.

Το KheperaIII είναι ένα όχημα με δύο τροχούς, τύπου *Dicycle*. Στην πραγματικότητα το όχημά μας χρησιμοποιεί μία αναλογία του διαφορικού και του κιβωτίου ταχυτήτων, με τη διαφορά, ότι αυτό δεν επιτυγχάνεται με μηχανικό αλλά με ηλεκτρονικό τρόπο.

Συγκεκριμένα, το όχημα καταφέρνει να στρίβει μεταβάλλοντας τη γωνιακή ταχύτητα των τροχών. Πχ: για να στρίψει το όχημα αριστερά, θα πρέπει ο αριστερός τροχός να περιστραφεί αργότερα από τον δεξιό. Ο κάθε τροχός κινείται όταν, με έναν βηματικό κωδικοποιητή (*incremental encoder*) στέλνουμε παλμούς ελέγχου, στον PID ελεκτή του κάθε κινητήρα. Ο βηματικός κωδικοποιητής χρησιμοποιεί διαμόρφωση εύρους παλμού (“*pulse width modulation*”, PWR). Αυτό μας βοηθάει να κατανοήσουμε καλύτερα τον τρόπο ελέγχου κίνησης του οχήματος, όπως φαίνεται στο (Σχήμα 16) βλέποντας το εύρος του παλμού έχουμε μια εικόνα για το πως περίπου θα καταναμηθεί η κίνηση στους τροχούς.

Στο συγκεκριμένο παράδειγμα (Σχήμα 16) βλέπουμε πως θα ήταν μία παλμοσειρά που θα ανάγκαζε το όχημα σε μία απότομη δεξιά στροφή. Το ένα σήμα θα είχε ενεργή έξοδο για το 95% του χρόνου που διαρκεί ο παλμός, ενώ το άλλο σήμα θα είχε ενεργή έξοδο μόνο για το 5% του χρόνου διάρκειας του παλμού.

Ο τροχός στα δεξιά θα κινούταν ελάχιστα, ενώ ο τροχός στα αριστερά θα δούλευε, σχεδόν, όλο το εύρος του παλμού. Ο κάθε βηματικός κινητήρας του KheperaIII, ελέγχεται από έναν PID (*proportional, integral, derivative*) ελεκτή, του οποίου οι όροι είναι αντιστοιχισμένοι στις παρακάτω μεταβλητές (K_p, K_i, K_d).

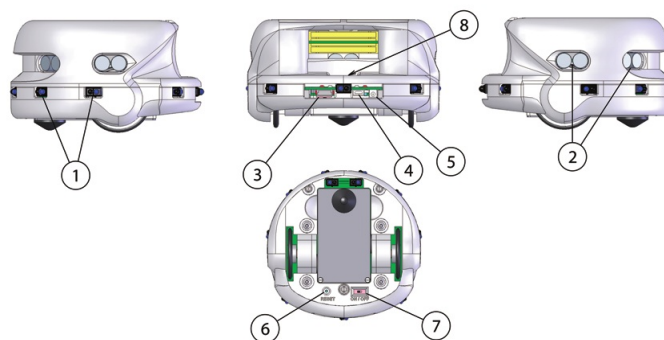
Ένα πολύ ενδιαφέρον χαρακτηριστικό του συγκεκριμένου οχήματος είναι ότι ο ελεκτής του μπορεί να μας δώσει δύο τρόπους ελέγχου.

- Έλεγχος ταχύτητας.
- Έλεγχος θέσης.

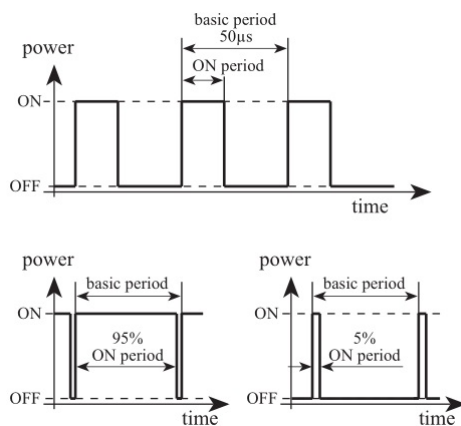
Ο τρόπος λειτουργίας του ελεκτή εξαρτάται από την εντολή που θα λάβει. Αν είναι εντολή ελέγχου ταχύτητας, ο ελεκτής επηρεάζει την γωνιακή ταχύτητα των τροχών και κατ'επέκταση την ταχύτητα του οχήματος. Αν είναι εντολή ελέγχου θέσης τότε ελεκτής εμφανίζει τη θέση στην οποία βρίσκεται το όχημα, σε ορισμένο, χαρτογραφημένο χώρο.

3.1.2 Αισθητήρες.

Το KheperaIII, είναι εφοδιασμένο με δύο διαφορετικά είδη αισθητήρων, υπερήχων (*Supersonic*) και υπέρυθρων (*Infrared, IR*). Οι αισθητήρες υπέρυθρων είναι συνολικά 11. Αποτελούνται από έναν εκπομπό και ένα δέκτη υπέρυθρου φωτός. Στην περιφέρεια του οχήματος βρίσκονται 9 από αυτούς,



Σχήμα 15: Το όχημα KheperaIII από διαφορετικές οπτικές γωνίες.



Σχήμα 16: Παράδειγμα παλμοσειράς ελέγχου των βηματικών κινητήρων ενός KheperaIII.

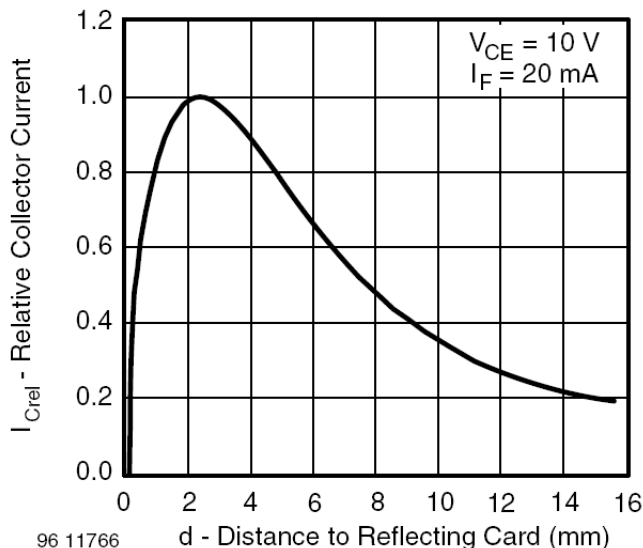
ενώ 2 βρίσκονται κάτω απο το όχημα (Σχήμα 18α.). Οι υπέρυθροι αισθητήρες πραγματοποιούν νέα μέτρηση, κάθε 33ms, ενώ κάθε 3ms οι αισθητήρες διαβάζονται με τη σειρά (1-9), απο τον ελεκτή. Στο συγκεκριμένο όχημα μπορούμε να πάρουμε δύο διαφορετικές μετρήσεις:

1. Μέτρηση του περιβάλλοντος φωτός, (*ambient light measurement*).

Αυτή η μέτρηση εξαρτάται πάρα πολύ απο τις συνθήκες που επικρατούν κάθε δεδομένη στιγμή γύρω απο το όχημά και καλό είναι να λαμβάνεται υπόψη οτι το σφάλμα της μέτρηση μπορεί να είναι μεγάλο δεδομένων των συνθηκών.

2. Μέτρηση ανακλώμενου φωτός – εγγύτητα (*reflected light measurement – proximity*).

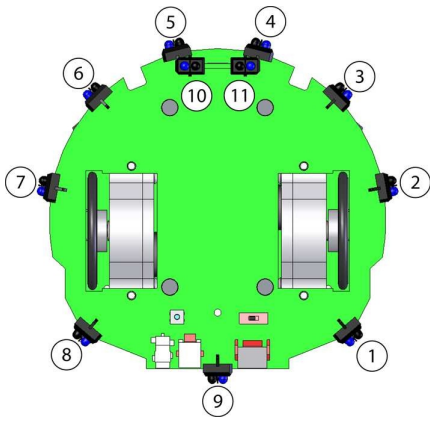
Η κύρια χρήση των υπέρυθρων αισθητήρων είναι η ανίχνευση αντικειμένων γύρω απο το όχημα. Οι μετρήσεις του ανακλώμενου φωτός εξαρτώνται απο την ανακλαστικότητα (*reflectivity*) του αντικειμένου και απο τις συνθήκες του περιβάλλοντος φωτός.



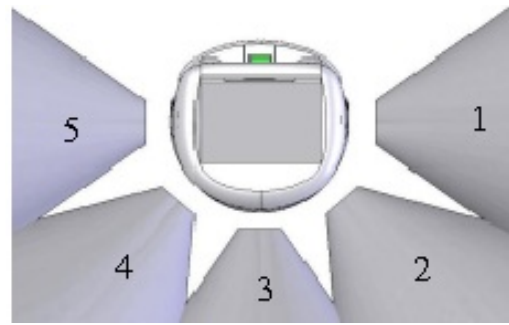
Σχήμα 17: Καμπύλη ρεύματος δέκτη (*receiver*) προς απόσταση του αντικειμένου.

Το πόσο φως θα ανακλαστεί στο αντικείμενο και τελικά θα επιστρέψει στον αισθητήρα, εξαρτάται απο το χρώμα, το υλικό και την επιφάνεια του αντικειμένου, επομένως η γραφική παράσταση μπορεί να διαφέρει απο αντικείμενο σε αντικείμενο. Σκοπός της είναι να έχουμε μια εικόνα για τη συμπεριφορά των αισθητήρων και δεν πρέπει να θεωρηθεί σημείο αναφοράς (Σχήμα 17).

Οι αισθητήρες υπερήχων βρίσκονται στην περιφέρεια του οχήματος όπως φαίνεται στο (Σχήμα 18β.) και είναι συνολικά 5 ζεύγη (πομπός-δέκτης). Το εύρος που καλύπτουν φαίνε-



(α') Θέσεις των υπέρυθρων αισθητήρων, 9 στην περιφέρεια και 2 κάτω απο το όχημα.



(β') Θέσεις υπέρηχων αισθητήρων και ο κώνος σάρωσης. Τα λευκά σημεία είναι περιοχές που δεν μπορεί να σαρώσει ο αισθητήρας.

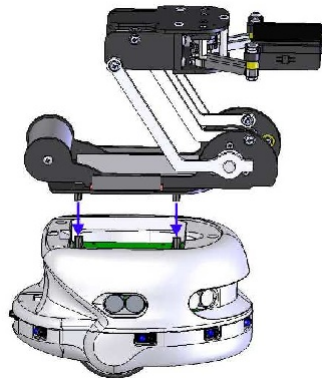
Σχήμα 18: Αισθητήρες του KheperaIII και η θέση τους στο όχημα.

ται στο (Σχήμα 18β'). Η περιοχές κάλυψης είναι χρωματισμένες με γκρι χρώμα, ενώ τα λευκά τμήματα είναι περιοχές που οι αισθητήρες δεν μπορούν να ανιχνεύσουν.

Η περιοχή που καλύπτει ο κάθε αισθητήρας έχει κωνικό σχήμα και μπορεί να ανιχνεύει αντικείμενα που βρίσκονται σε απόσταση $[20cm-4m]$. Στην περίπτωση που κάποιο αντικείμενο βρίσκεται σε απόσταση μικρότερη των $20cm$, επομένως βρίσκεται εκτός του κώνου σάρωσης των αισθητήρων υπέρηχων, χρησιμοποιούμε τους υπέρυθρους αισθητήρες οι οποίοι μπορούν να ανιχνεύουν αντικείμενα απο $[1cm-30cm]$. Η ονομαστική συχνότητα που εκπέμπεται απο τους συγκεκριμένους μετατροπείς (*transducers*), είναι $40kHz[\pm 1kHz]$.

Το robot KheperaIII είναι εξοπλισμένο με μπαταρία πολυμερών λιθίου $[7.4volt]$ χωρητικότητας $[1400mAh]$. Επίσης λειτουργεί με συνεχές ρεύμα $[9volt-2A]$.

Στο όχημα προσαρμόζεται βραχίονας εργασίας 2 βαθμών ελευθερίας, οποίος μπορεί να ελεγχθεί και αυτός, μεσω εντολών από τον μικροεπεξεργαστή του KheperaIII.



Σχήμα 19: KheperaIII εφοδιασμένο με το βραχίονα εργασίας.

4 Software.

4.1 V-rep.

Το V-rep είναι ένα πρόγραμμα προσομοίωσης ρομποτικών οχημάτων και συστημάτων.

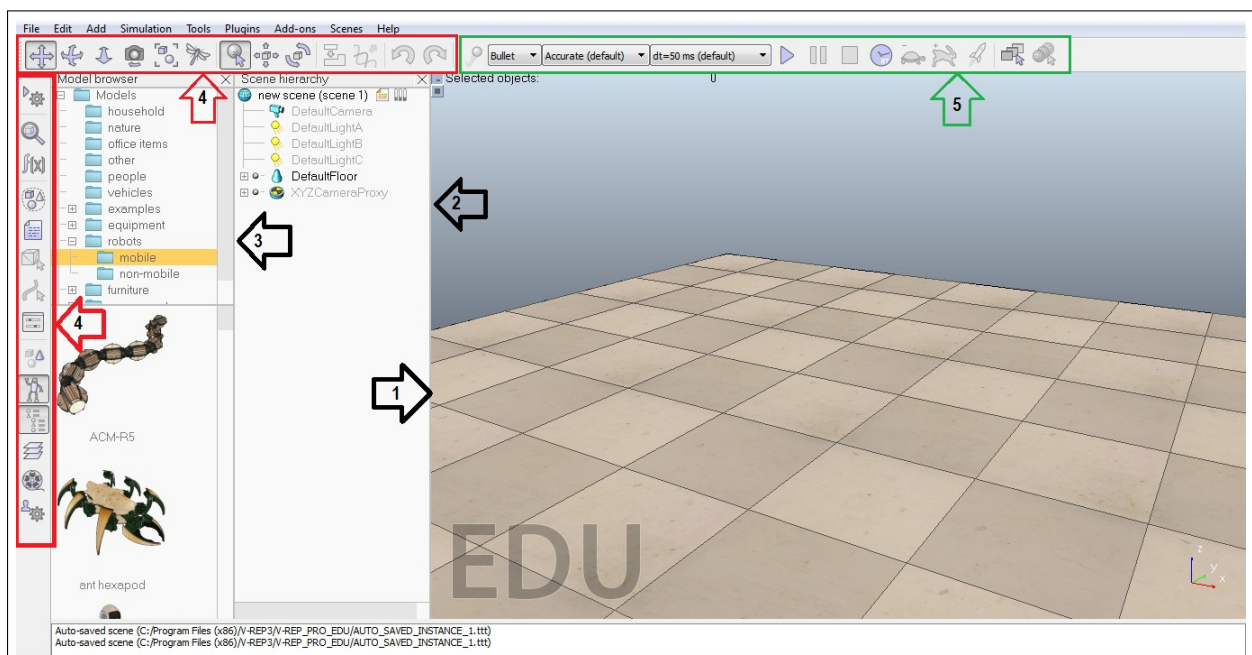


The robot simulator V-REP, with integrated development environment, is based on a distributed control architecture: each object/model can be individually controlled via an embedded script, a plugin, a ROS node, a remote API client, or a custom solution. This makes V-REP very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, Matlab, Octave or Urbi. V-REP is used for fast algorithm development, factory automation simulations, fast prototyping and verification, robotics related education, remote monitoring,

safety double-checking, etc.

Σημείωση: Παρατηρείται το αγγλικό κείμενο καθώς ορισμένες έννοιες (πχ: embedded script κα.) είναι δύσκολο να αποδοθούν στα Ελληνικά χωρίς να προκαλέσουν σύγχυση. Ο λόγος της σύγχυσης έχει να κάνει με την εξοικείωση του αναγνώστη σε συγκεκριμένες ξενικές εκφράσεις, την έλλειψη βιβλιογραφίας στα Ελληνικά κα. Δεν υπάρχει καμία υπόνοια για την αδυναμία της Ελληνικής γλώσσας σαν μέσο απόδοσης. Ο σκοπός αυτού του κειμένου δεν είναι η μετάφραση.

Στο περιβάλλον αυτό, θα προσομοιώσουμε την κίνηση ενός οχήματος KheperaIII, ελεγχόμενο από ένα ασαφή ελεγκτή (fuzzy controller) που θα σχεδιάσουμε σε περιβάλλον Matlab, όπως θα εξηγήσουμε παρακάτω. Λίγα λόγια για το περιβάλλον V-rep και το user interface του.



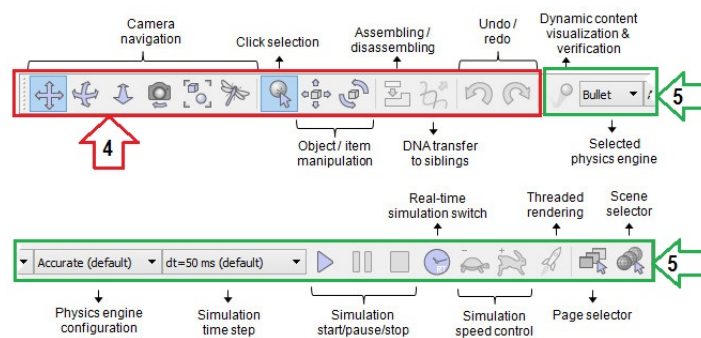
Σχήμα 20: Συνολική εικόνα του αρχικού χώρου εργασίας της V-rep.

1. Παράθυρο προσομοίωσης *User Interface*(1).

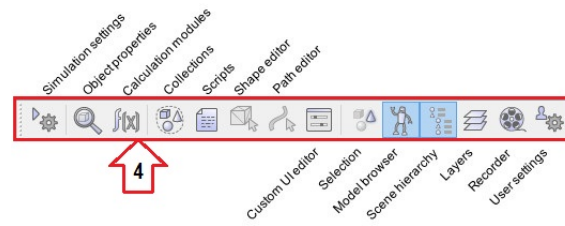
Ο βασικός χώρος εργασίας του προγράμματος. Στο χώρο αυτό εμφανίζονται τα μοντέλα και αντικείμενα που εισάγουμε και σε αυτό το χώρο πραγματοποιούνται οι προσομοιώσεις. Στην "όρολογία" της V-rep τα αρχεία που ανοίγουμε ονομάζονται *scenes*.

2. Παράθυρο ιεραρχίας αντικειμένων *Scene Hierachy* (2).

Στο χώρο αυτό εμφανίζονται όλα τα αντικείμενα που προσθέτουμε στο παράθυρο προσομοίωσης ιεραρχικά, με δενδροειδή δομή για καλύτερη περιήγηση του σχεδιαστή μεταξύ των αντικειμένων. Επιτρέπει επίσης την ομαδοποίηση των αντικειμένων, κάτι που διευκολύνει το σχεδιαστή καθώς προσφέρει συνολική εικόνα της προόδου μέχρι εκείνο το σημείο.



(α') Γραμμές Εργαλείων.



(β') Εργαλεία προσωμοίωσης.

Σχήμα 21: Παρουσίαση των λειτουργιών, των γραμμών εργαλείων.

3. Περιηγητής Μοντέλων *Model Browser* (3).

Σε αυτό το παράθυρο μπορούμε να προσπελάσουμε τα προεπιλεγμένα μοντέλα που υπάρχουν στο V-ger, πχ: αντικείμενα, άνθρωποι, οχήματα, κινούμενα και ακίνητα ρομποτ. Είναι ομαδοποιημένα σε μεγάλες κατηγορίες και αρχειοθετημένα σε δενδροειδή δομή για γρήγορη περιήγηση. Ενώ κάτω απο τους φακέλους υπάρχει και προεπισκόπηση των μοντέλων ανα κατηγορία.

4. Γραμμές εργαλείων (4).

Εργαλεία που έχουν να κάνουν με μετακίνηση ή περιστροφή των αντικειμένων που βρίσκονται στο παράθυρο προσομοίωσης (1) καθώς και με τη θέση και τις ρυθμίσεις της κάμερας, υπάρχουν εργαλεία σχεδιασμού, εργαλεία συναρτήσεων και υπολογισμών, περιήγηση μεταξύ των διαφορετικών επιπέδων (layers) του project που σχεδιάζουμε. Να σημειώσουμε ότι στο περιβάλλον V-ger μπορούμε να έχουμε εικόνα απο τις κάμερες που θα τοποθετήσουμε στο παράθυρο προσομοίωσης και να εμφανίζουμε τα διαφορετικά video σε επιμέρους παράθυρα.

5. Εργαλεία προσωμοίωσης *Simulation tools*(5).

Επιλέγουμε τη μηχανή φυσικής που θα τρέξει ο προσομοιωτής (*V-ger*) για να κάνει τους απαραίτητους υπολογισμούς και να έχει την αντίστοιχη συμπεριφορά κατά την προσομοίωση. Πχ: συγκρουση, αναπήδηση, πτώση κλπ. Επίσης επιλέγουμε την ταχύτητα της προσομοίωσης, καθώς και τη διάσταση απο την οποία βλέπουμε το παράθυρο προσομοίωσης (1).

4.1.1 Χώρος εργασίας-χειρισμός.

Σημείωση: Σε ένα πρόγραμμα με γραφικό περιβάλλον είναι δυνατό να γίνει η ίδια εργασία με πάνω απο έναν τρόπους. Παράδειγμα: Μπορούμε να κάνουμε save κάποιο αρχείο:

- File→ Save.
- Πατώντας το εικονίδιο (συντόμευση) της δισκέτας.
- Πατώντας Ctrl+S

Το V-ger δεν αποτελεί εξαίρεση. Οι παρακάτω υποδείξεις δεν αποτελούν εγχειρίδιο χρήσης της εφαρμογής. Είναι παραδείγματα που δίνουν μια μόνο πτυχή του προγράμματος.

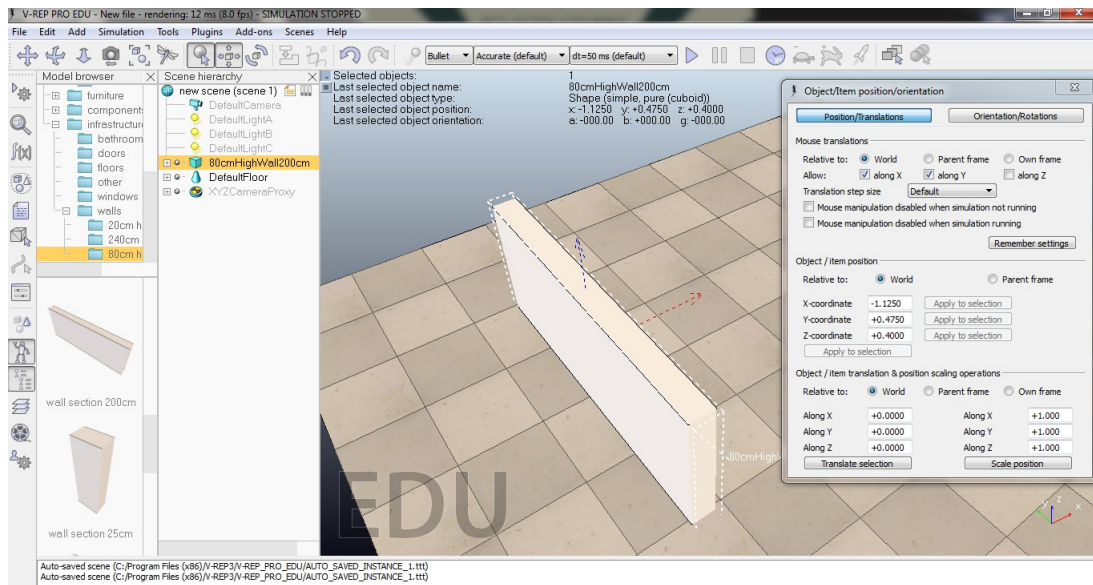
Ας επιχειρήσουμε να δημιουργήσουμε ένα χώρο εργασίας.

Ξεκινάμε τρέχοντας το V-Rep. Επιλέγουμε File→ New Scene. Ανοίξαμε μια νέα σκηνή. Παρατηρούμε το επίπεδο που υπάρχει είναι ουσιαστικά το “έδαφος”.

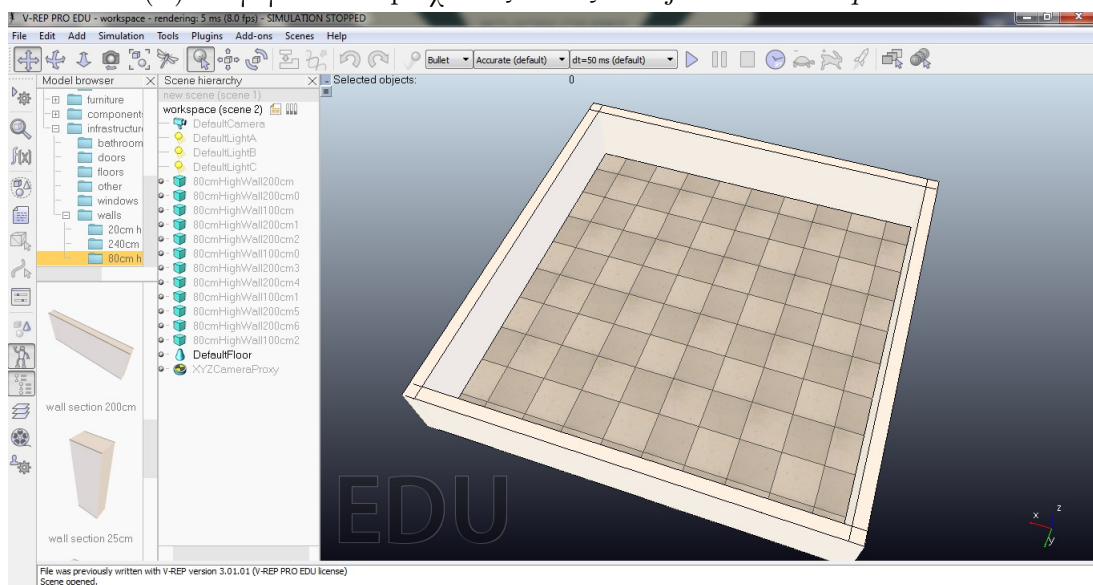
Πατώντας και κρατώντας πατημένο δεξί-κλικ μπορούμε να μετακινήσουμε/περιστρέψουμε το χώρο εργασίας ώστε να δουλέψουμε στις περιοχές που επιθυμούμε.

Με *scroll-in/scroll out* κάνουμε *zoom-in/zoom-out*, ώστε να πλησιάσουμε με την κάμερα ή να απομακρυνθούμε αντίστοιχα.

Ας προσπαθήσουμε τώρα να περιστοιχίσουμε το επίπεδο με τοίχο, ώστε να δημιουργήσουμε ένα ορισμένο χώρο εργασίας.



(α') Σκηνή του V-Rep έχοντας επιλέξει *Object-item manipulation*.



(β') Απλό παράδειγμα οριοθετημένου χώρου εργασίας χρησιμοποιώντας 8 *wall sections 200cm* και 4 *wall sections 100cm*.

Σχήμα 22: V-Rep User Interface.

Στο παράθυρο *Model Browser* (3), στο δένδροειδές μενού, επιλέγουμε *infrastructure*. Παρατηρούμε οτι περιέχει υποφακέλους. Ανοίγουμε το φάκελο "walls". Στα παραδείγματά μας επιλέγουμε τοίχους απο το φάκελο *80cm high walls*. Είναι ένα αρκετά πρακτικό ύψος τοίχου, για να μπορέσει κανείς να δουλέψει με σχετική ευκολία στο πρόγραμμα. Πχ: οι τοίχοι απο τον φάκελο

240cm high walls δυσκολεύουν αρκετά την ορατότητα κατά τη διάρκεια της προσομοίωσης.

Έχουμε λοιπόν ανοιχτό το φάκελο 80cm high walls. Από εκεί επιλέγουμε wall section 200cm. Με drag& drop εισάγουμε στη σκηνή μας τον τοίχο.

Στη συνέχεια για να μετακινήσουμε/περιστρέψουμε τον τοίχο, κάνουμε "άριστερό-κλίκ" στη Γραμμή εργαλείων (4) στις επιλογές, Object-item manipulation Σχήμα 17α. Είτε με σαφής συντεταγμένες, είτε με απλή μετακίνηση με το ποντίκι, τοποθετούμε τον τοίχο, όπου επιθυμούμε στο επίπεδο.

Κάνοντας το ίδιο με τα διαθέσιμα κομμάτια τοίχων wall sections, μπορούμε να κατασκευάσουμε το χώρο εργασίας, όπως επιθυμούμε.

Ας δείξουμε στη συνέχεια πώς μπορούμε να εισάγουμε γεωμετρικά στερεά, είτε ως μορφοποίηση του χώρου είτε ως εμπόδια. Με Δεξί-κλίκ μέσα στο χώρο εργασίας, εμφανίζεται ένα παράθυρο. Εκεί επιλέγουμε Add→ Primitive shape→ διαλέγουμε το σχήμα που θέλουμε να εισάγουμε. Στο παράθυρο που εμφανίζεται ορίζουμε τις επιθυμητές διαστάσεις και πατάμε OK. Κοιτάζουμε στη σκηνή για να διαπιστώσουμε ότι το αντικείμενό εμφανίστηκε στη σκηνή επιτυχώς.

Με τα πλήκτρα Object-item manipulation, από τη Γραμμή εργαλείων (4), μπορούμε να μετακινήσουμε/περιστρέψουμε το αντικείμενο.

5 Ασαφής Λογική

5.1 Η έννοια της Ασαφούς Λογικής-*Fuzzy Logic*.

5.1.1 Εισαγωγή.

Για να ορίσουμε επιτυχημένα την ασαφή λογική πρέπει να ανατρέξουμε στους λόγους που οδήγησαν στην ανακάλυψή της.

Η Ασαφής Λογική (*Fuzzy Logic*) είναι η προσπάθεια των επιστημόνων και κυρίως αυτών που ασχολούνται με την “τεχνητή νοημοσύνη” να μελετήσουν και να κατανοήσουν τη δομή της φυσικής γλώσσας του ανθρώπου. [1]

Σύμφωνα με την κλασσική δίτιμη λογική μια λογική πρόταση μπορεί να πάρει μόνο δύο τιμές, δηλαδή μπορεί να είναι αληθής ή ψευδής (1ή0), αποκλείοντας τρίτη λύση. Έτσι αν μια λογική πρόταση δεν είναι αληθής τότε θα είναι αναγκαία ψευδής.[1]

Η ιδέα πίσω από μία γλώσσα προγραμματισμού, υψηλού ή όχι επιπέδου, είναι να δώσει σαφείς, κατηγορηματικές εντολές σε έναν υπολογιστή. Όσο πιο σαφείς είναι οι εντολές που δίνουμε, μειώνουμε τις πιθανότητες ο υπολογιστής να συναντήσει καταστάσεις που δεν προκύπτουν από τους κανόνες που εμείς θέτουμε. Όσο ο υπολογιστής συναντά ερωτήματα, των οποίων οι απαντήσεις προκύπτουν ως κατηγορηματικά συμπεράσματα από τους κανόνες που εισάγουμε, συνεχίζει να δίνει σωστές απαντήσεις. Άρα για έναν υπολογιστή η καταστάσεις, «περισσότερο-λιγότερο», «πιο ζεστό-πιο κρύο», «πιο αργά-πιο γρήγορα» είναι ουσιαστικά ακατανοήτες. Στην πραγματικότητα για να καταφέρει ένας υπολογιστής να επεξεργαστεί τέτοιες έννοιες, πρέπει να του ορίσουμε ένα μέτρο σύγκρισης πρώτα. Πχ: «περισσότερο ή λιγότερο» σε σχέση με τι; Πιο απλά, κατά μια έννοια η “*Θεωρία των Ασαφών Συνόλων*” μπορεί να θεωρηθεί ότι στοχεύει να μαθηματικοποιήσει, δηλαδή να μοντελοποιήσει τη φυσική-καθομιλουμένη γλώσσα και την κοινή λογική. [1]

Ο άνθρωπος όταν περιγράφει ή σκέφτεται δεν θέτει ορισμούς και μετά συζητάει. Μορφοποιεί το αποτέλεσμα με βάση τα ερεθίσματά του. Για παράδειγμα, κάποιος που θέλει να διαπιστώσει πόσο ζεστό είναι το νερό πριν κάνει μπάνιο, θα ανοίξει τη βρύση και θα δοκιμάσει με το χέρι του τη θερμοκρασία.

Αν τον ρωτήσουν πόσους βαθμούς έχει το νερό σίγουρα δεν θα ξέρει. Όταν όμως, κάποιος τον ρωτήσει αν είναι «ζεστό» το νερό, τότε θα του απαντήσει με βάση την *αίσθηση* που είχε στο χέρι του.

Ένας υπολογιστής λοιπόν, θα μπορεί να απαντήσει αν το νερό είναι «ζεστό» μόνο αν πάρει μέτρηση της θερμοκρασίας και συγκρίνει τη μέτρηση με κάποια προκαθορισμένη τιμή που του όρισε ο προγραμματιστής. Επομένως αν η ορισμένη τιμή είναι $30^{\circ}C$ (βαθμοί Κελσίου) και το νερό είναι $29^{\circ}C$, ο υπολογιστής θα βγάζει έξοδο ότι το νερό «δεν είναι ζεστό» μέχρι να πάρει μέτρηση $30^{\circ}C$.

Σίγουρα, ένας άνθρωπος που θέλει να κάνει μπάνιο θα κρίνει ότι το νερό είναι ζεστό και στις δύο περιπτώσεις. Μέσω της ασαφούς λογικής καταφέρνουμε να προσεγγίσουμε καλύτερα καταστάσεις «περισσότερο-λιγότερο» χρησιμοποιώντας και στον τρόπο που προγραμματίζουμε μία πιο αφαιρετική προσέγγιση.

5.1.2 Ασάφεια και Λογική.

Γίνεται η «ασάφεια» να είναι λογική; Μπορούμε να συνδιάσουμε δύο «ασαφή» συμπεράσματα σε ένα νέο «λογικό» συμπέρασμα, όπως ίσως θα κάναμε με μεγάλη ασφάλεια σε δύο αντίστοιχα «σαφή» συμπεράσματα; Στην καθημερινότητά ακούμε συχνά τις φράσεις:

«Ο ήλιος καίει».

«Μην πιείς νερό, είναι παγωμένο».

«Ο καφές είναι πολύ γλυκός».

«Πήγαινε πιά αργά, με το αυτοκίνητο».

Όλες αυτές οι φράσεις όχι μόνο είναι δύσκολο να τις επεξεργαστεί ένας υπολογιστής, είναι τόσο «ασαφείς» που κανένας προγραμματιστής δε θα μπει στον κόπο να κωδικοποιήσει κάν τέτοιες προτάσεις. Η σαφήνεια αυτών των φράσεων λοιπόν είναι ξεκάθαρα ελάχιστη.



Σχήμα 23: Ακριβής (*precise*) περιγραφή ενός γεγονότος.

Σπάνια όμως δύο άνθρωποι κάνουν τον ακόλουθο διάλογο.

-A: Μην πιείς νερό, είναι παγωμένο.

-B: Αν το νερό είναι όντως παγωμένο, δεν μπορώ να το πιώ.

Είναι όμως λογική η απάντηση. Χαρακτηρίζεται επίσης από απόλυτη σαφήνεια. Δεν είναι όμως απάντηση που θα έδιναν άνθρωποι, χρησιμοποιώντας φυσική γλώσσα.

Οι άνθρωποι περιγράφουν την πραγματικότητα που βλέπουν. Αντιδρούν με βάση το περιβάλλον τους, την εκάστοτε χρονική στιγμή. Χρησιμοποιούν τη γλώσσα για να εκφράσουν τη σκέψη τους. Τροποποιούν τις προτάσεις τους με βάση το σύστημα αναφοράς που επιλέγουν.

Η πρόταση «Μην πιείς νερό, είναι παγωμένο», είναι λάθος τελικά;

Εδώ αναγκαζόμαστε να εξετάσουμε την πρόταση όχι σαν μια αλληλουχία λέξεων, αλλά σαν νόημα. Επομένως υπονοούμε ότι υπάρχει μία άλλη «οντότητα» που την ονομάζουμε νόημα και αυτή η οντότητα δεν είναι κατηγορηματικά (*explicitly*) ορισμένη από τις λέξεις. Σ' αυτό το σημείο ως θεωρήσουμε ότι τα δομικά στοιχεία μίας πρότασης είναι οι λέξεις, (*Αγνοώντας, συνειδητά, ότι οι λέξεις αποτελούνται από γράμματα*).

Ίσως οι λέξεις να μην είναι μονοσήμαντα ορισμένες. Όμως αν ρωτήσουμε έναν άνθρωπο να ερμηνεύσει μια-μια τις λέξεις της πρότασης «Μην πιείς νερό, είναι παγωμένο», θα έχουμε μεγάλη συμφωνία στην επιμέρους ερμηνεία των λέξεων.

Άρα θα μάντευε κανείς ότι πολλές μικρές οντότητες, που τείνουν να είναι μονοσήμαντα ορισμένες (οι λέξεις), συνδιάσσοντας 'τες με μοναδική αλληλουχία (σύνταξη πάντα με την ίδια σειρά) θα είχαν σαν αποτέλεσμα μία πρόταση που θα σημαίνει κυριολεκτικά ένα και μόνο ένα πράγμα.

Ας δούμε τη πρόταση όχι σαν άθροισμα οντοτήτων. Αλλά σαν οντότητα εξολοκλήρου.

Εύκολα ένας άνθρωπος θα καταλάβαινε ότι αυτή η πρόταση δεν είχε σαν «σκοπό» να κυριολεκτήσει, αλλά να ενημερώσει, να προειδοποιήσει, να προστατέψει.

Για να το κάνει αυτό πρέπει όχι μόνο να δομήσει μία λογικά ορθή και σαφή πρόταση, αλλά να προλάβει να προφέρει αυτή την πρόταση σε εκείνον που θέλει να προειδοποιήσει-προστατέψει.

Ας δώσουμε ένα παράδειγμα που ο χρόνος είναι πιο σημαντικός από ότι στην περίπτωση με το «παγωμένο νερό».



Σχήμα 24: Ασαφής (Fuzzy) περιγραφή ενός γεγονότος.

Στην κλασσική πια σκηνή των cartoon, που τους πέφτει το αμόνι στο κεφάλι. Η σαφής και λογική πρόταση σ'αυτη την περίπτωση είναι ίσως κάπως έτσι:

«Ένα αμόνι διαγράφει ελεύθερη πτώση με ταχύτητα $43.5m/s$ και στόχο το κεφάλι σου, σύμφωνα με το βάρος του, σ'αυτό το σημείο του πλανήτη, η μάζα του τη στιγμή που θα πέσει στο κεφάλι σου θα είναι $1500Kg \dots$ »

Καταλαβαίνουμε όλοι οτι μέχρι να ακούσει ο φίλος μας στο (Σχήμα 23.) τη λέξη «διαγράφει» θα έχει πέσει το αμόνι στο κεφάλι του.

Αντιθέτως η πρόταση «Φύγε!», «Πρόσεξε!» δεν έχουν καμία σαφήνεια,αλλά θα έχουν το επιθυμητό αποτέλεσμα. Να γλυτώσεις το αμόνι.

Γυρίζοντας στο παράδειγμα με το «παγωμένο νερό». Ο σκοπός είναι πάλι να «προειδοποιήσεις-προστατέψεις». Άρα, επιτυχημένη «έξοδος» του συστήματος είναι να μην πεί κάποιος το νερό. Σίγουρα το νερό δεν βρίσκεται σε θερμοκρασία $0^{\circ}C$. Άρα είναι ακόμα σε υγρή κατάσταση και κατ'επέκταση πίνεται. Τότε ακριβώς είναι που έχει νόημα να χρησιμοποιήσεις τη λάθος πρόταση (ανακριβή και υπερβολική στο παράδειγμά μας) για να πετύχεις την επιθυμητη έξοδο. Αν ήταν όντως πάγος, δεν θα υπήρχε λόγος προειδοποίησης, στη φυσική γλώσσα του ανθρώπου. Δεν μπορείς να πιείς κάτι στερεό. Αντιθέτως η «μη παγωμένη», άρα πόσιμη, κατάσταση του νερού μας «αναγκάζει» να πούμε λάθος πρόταση για να έχουμε σωστή έξοδο. Ενώ αν επιλέγαμε να πούμε τη «σωστή» πρόταση, να “μη είχαμε καν έξοδο”.

Επομένως στο παράδειγμά μας, το νερό είναι τόσο χύο που «τείνει» στο παγωμένο. Είναι πιο κοντά στο «παγωμένο» (A) παρά στο «όχι παγωμένο” ($\neg A$).

Πιο απλά, κατά μια έννοια η “Θεωρία των Ασαφών Συνόλων” μπορεί να θεωρηθεί ότι στοχεύει να μαθηματικοποιήσει, δηλαδή να μοντελοποιήσει τη φυσική-καθομιλουμένη γλώσσα και την κοινή λογική [1].

Η ασάφεια (fuzziness) είναι μια έννοια που σχετίζεται με την ποσοτικοποίηση της ποιοτικής πληροφορίας και οφείλεται κυρίως σε μη-ακριβή δεδομένα.[6]

5.1.3 Δομή.

Αρχικός σκοπός του γνωστού κλασσικού συνόλου A είναι να μοντελοποιήσει μαθηματικά, δηλαδή να αναπαραστήσει τυπολογικά, κάποια λογική έννοια. πχ: η έννοια “οι ακέρατοι αριθμοί που είναι μεταξύ 2 και 6” μπορεί να παρασταθεί με το σύνολο: $A = \{x \in \mathbb{Z} | 2 \leq x \leq 6\} = \{3, 4, 5\}$ Έτσι

πρώτα χρειαζόμαστε ένα σύνολο αναφοράς που περιέχει όλα τα στοιχεία τα σχετικά με την έννοια που θέλουμε να εκπροσωπήσουμε μέσω του συνόλου A . (Γ. Α. Θεοδώρου).

Είναι γνωστό επίσης, ότι γενικά ένα κλασικό (ή σύνηθες) σύνολο A ως προς σύνολο αναφοράς X , μπορεί να παρασταθεί ισοδύναμα μέσω της χαρακτηριστικής συνάρτησής του.

$$I_A : x \in X \Rightarrow$$

$$I_A(x) = \begin{cases} 1, & \text{όταν } x \in A. \\ 0, & \text{όταν } x \notin A. \end{cases}$$

Όπως στο παραπάνω παράδειγμα, όπου η χαρακτηριστική συνάρτηση του συνόλου $A = \{3, 4, 5\}$, είναι:

$$I_A : x \in X \Rightarrow$$

$$I_A(x) = \begin{cases} 1, & \text{όταν } x \in A = \{3, 4, 5\}. \\ 0, & \text{όταν } x \notin A. \end{cases}$$

Από τα παραπάνω είναι σαφές πως για τα κλασικά σύνολα υπάρχουν δύο κατηγορίες:

- Στην κατηγορία με τα στοιχεία του X που ανήκουν στο A .
- Στην κατηγορία με τα στοιχεία του X που δεν ανήκουν στο A .

Η μετάβαση από την μια κατηγορία (των στοιχείων του X που ανήκουν στο A), στην άλλη κατηγορία (των στοιχείων του X που δεν ανήκουν στο A), είναι απολύτως σαφής, απότομη και ξεκάθαρη.

Στην Ασαφή Λογική δεν υπάρχει το δίτιμο σύνολο, αλλά ένα νέο πλειότιμο σύνολο A , όπου η μετάβαση από την κατηγορία των στοιχείων του X που ανήκουν στο ασαφές σύνολο A , στην άλλη κατηγορία των στοιχείων του X που δεν ανήκουν στο A , δεν είναι απότομη-σαφής αλλά είναι βαθμιαία-ασαφής όπως συνήθως συμβαίνει στην πραγματικότητα. Πιο συγκεκριμένα η Ασαφής Συνολοθεωρία εισάγει την συνάρτηση συμμετοχής $\mu_A(x)$ ενός ασαφούς συνόλου A .

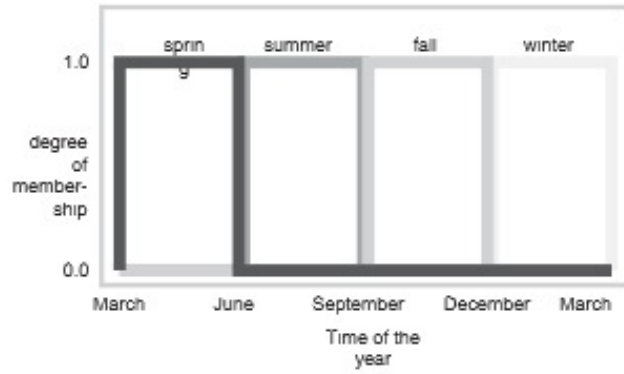
Ορισμός 1. Ασαφές σύνολο. Έστω Ξ ένα μη μηδενικό σύνολο. Ένα ασαφές σύνολο A του Ξ χαρακτηρίζεται από την συνάρτηση συμμετοχής του $\mu_A : X \rightarrow [0, 1]$ όπου $\mu_A(x)$ είναι ο βαθμός συμμετοχής του στοιχείου $x \in X$ στο ασαφές σύνολο A .

Το ασαφές σύνολο A χαρακτηρίζεται πλήρως από το σύνολο των ζευγαριών $\{(x, \mu_A(x))\}$, όπου $x \in X$. Αν $X = \{x_1, x_2, \dots, x_n\}$ ένα πεπερασμένο σύνολο και A ένα ασαφές σύνολο του X τότε χρησιμοποιούμε τον συμβολισμό:

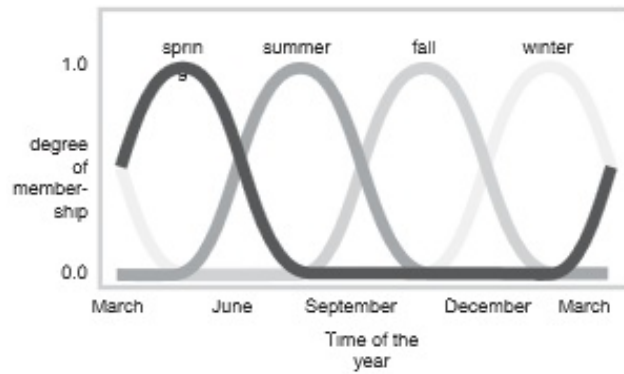
$$A = \frac{\mu_1}{x_1} + \frac{\mu_2}{x_2} + \dots + \frac{\mu_n}{x_n}$$

όπου $\frac{\mu_i}{x_i}$ συμβολίζει ότι μ_i είναι ο βαθμός συμμετοχής του x_i στο A και το $+$ συμβολίζει την ένωση (Σ. Βολογιαννίδης).

Είναι προφανές ότι όσο μεγαλύτερο βαθμό συμμετοχής έχει ένα στοιχείο τόσο περισσότερο "ανήκει" στο σύνολο.



(α') Γραφική παράσταση βαθμού συγγένειας εποχών προς μήνες του χρόνου. Χρησιμοποιώντας τον ορισμό του κλασικού συνόλου.



(β') Γραφική παράσταση βαθμού συγγένειας εποχών προς μήνες του χρόνου. Χρησιμοποιώντας τον ορισμό του Ασαφούς συνόλου.

Σχήμα 25: Μια γραφική προσέγγιση του 'πόσο' ένα στοιχείο ενός συνόλου ανήκει στο σύνολο.

Η γωνιακή ταχύτητα του οχήματος δίνεται:

$$\dot{\theta} = \frac{v}{R_1}$$

Όμως

$$R_1 = \frac{L}{\tan\gamma}$$

, όπου L : το μήκος μεταξύ των τροχών του οχήματος ή μεταξόνιο. Επομένως το όχημά όταν στρίβει πραγματοποιεί ένα κομμάτι μια κυκλικής τροχιάς ή ένα τόξο. Από την ισότητα αυτή παρατηρούμε ότι το τόξο που διαγράφει ένα όχημα που στρίβει εξαρτάται από το μήκος του μεταξονίου. Να σημειώσουμε επίσης ότι η γωνία στρέψης γ έχει μηχανικούς περιορισμούς και η μέγιστη τιμή της καθορίζει την ελάχιστη τιμή της R_1 .

Παρατηρούμε επίσης ότι $R_2 > R_1$, που σημαίνει ότι ο μπροστινός τροχός διαγράφει μεγαλύτερο τόξο άρα πρέπει να κινείται με μεγαλύτερη ταχύτητα από τον πίσω.

Στην περίπτωση του τετράτροχου οχήματος η απόσταση μεταξύ αριστερών και δεξιών τροχών είναι συγκεκριμένη και μη αμελητέα στους υπολογισμούς μας. Υποθέτοντας ότι οι γωνίες στρέψης είναι γ_L και γ_R για αριστερό και δεξί τροχό, συμπεραίνουμε ότι για να στρίβει ένα τετράτροχο όχημα με επιτυχία και χωρίς να προκαλεί έντονες φθορές στα ελαστικά και στο σύστημα στρέψης πρέπει $\gamma_R \neq \gamma_L$. Τη λύση σ αυτή την περίπτωση δίνει η γεωμετρία στρέψης *Ackermann* (ενότητα 2.4). Η γεωμετρία του σχήματος 27, μας βοηθά να κατανοήσουμε την ανάγκη να περιστρέφονται οι τροχοί σε διαφορετικές ταχύτητες, βοηθώντας να καταλάβουμε τη χρήση του διαφορικού (ενότητα 2.0).

Τέλος να σημειώσουμε ότι όταν $\gamma \rightarrow \frac{\pi}{2}$ σημαίνει ότι ο άξονας κίνησης του μπροστινού τροχού είναι κάθετος στον άξονα κίνησης του πίσω, επομένως προσεγγίζουμε τους κατασκευαστικούς περιορισμούς του οχήματος και θα προσπαθήσουμε στην σχεδίαση να αποφύγουμε αυτές τις τιμές. Σε επίπεδο προσωμοίωσης η ύπαρξη μη επιτρεπτών τιμών στη γωνία στρέψης αυξάνει το τόξο που διαγράφει το όχημα μέχρι να φτάσει στον επιθυμητό στόχο.[7]

6.1.1 Κίνηση του οχήματος σε δοσμένο σημείο.

Για να οδηγήσουμε το όχημα σε μία συγκεκριμένη θέση (x^*, y^*, θ^*) Ξεκινώντας από την σχέση:

$$\dot{x} = v \cos\theta \quad (1\alpha')$$

$$\dot{y} = v \sin\theta \quad (1\beta')$$

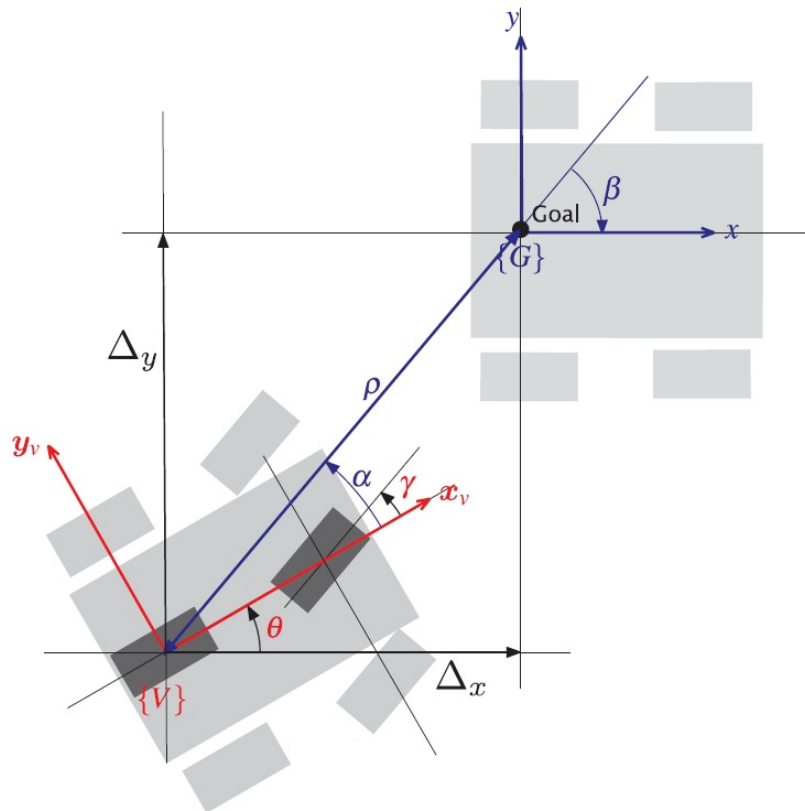
$$\dot{\theta} = \frac{v}{L} \tan\gamma \quad (1\gamma')$$

Με σκοπό να ελέγξουμε τον τελικό προσανατολισμό του οχήματος, γράφουμε την εξίσωση (1) με μορφή πινάκων [7]:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \gamma \end{pmatrix} \quad (2)$$

Στη συνέχεια μετασχηματίζουμε τις εξισώσεις σε πολική μορφή ως εξής: βλ. Σχήμα 27

$$\rho = \sqrt{\Delta_x^2 + \Delta_y^2}$$



Σχήμα 27: Σύστημα πολικών συντεταγμένων του οχήματος κινούμενο προς τυχαίο στόχο. ρ είναι η απόσταση από το στόχο, β είναι η γωνία του διανύσματος του στόχου σε σχέση με το περιβάλλον σύστημα αναφοράς και α είναι η γωνία του διανύσματος του στόχου σε σχέση με το πλαίσιο του οχήματος.

$$\alpha = \tan^{-1} \frac{\Delta y}{\Delta x} - \theta$$

$$\beta = -\theta - \alpha$$

Μετασχηματίζοντας την εξίσωση 2 καταλλήγουμε:

$$\begin{pmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{pmatrix} = \begin{pmatrix} -\cos\alpha & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ -\frac{\sin\alpha}{\rho} & 0 \end{pmatrix} \begin{pmatrix} v \\ \gamma \end{pmatrix}, \alpha \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right] \quad (3)$$

Ορίζουμε αυτό το πεδίο ορισμού λόγω των κατασκευαστικών περιορισμών του οχήματος. Για παράδειγμα αν κάποιος στόχος δεν ανήκει στο π.ο. σημαίνει ότι ο στόχος που στέλνουμε το όχημα βρίσκεται πίσω από το όχημα. Περισσότερα στοιχεία θα παρατεθούν στην ενότητα της σχεδίασης του ελεκτή και της προσωμοίωσης.

6.2 Σχεδίαση Ασαφούς Ελεκτή.

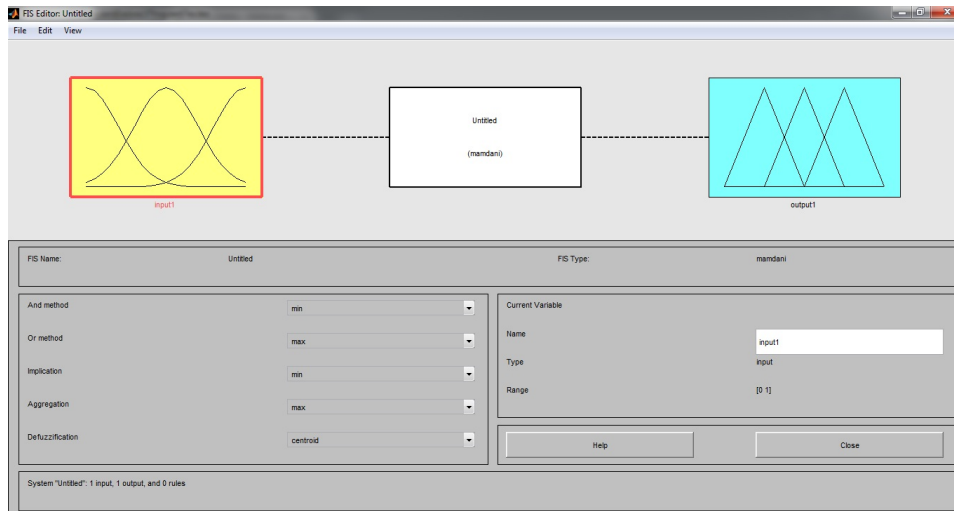
Θα δείξουμε στη συνέχεια ένα παράδειγμα σχεδίασης ενός ασαφούς ελεκτή σε περιβάλλον MATLAB.

Γράφοντας στο *command window*

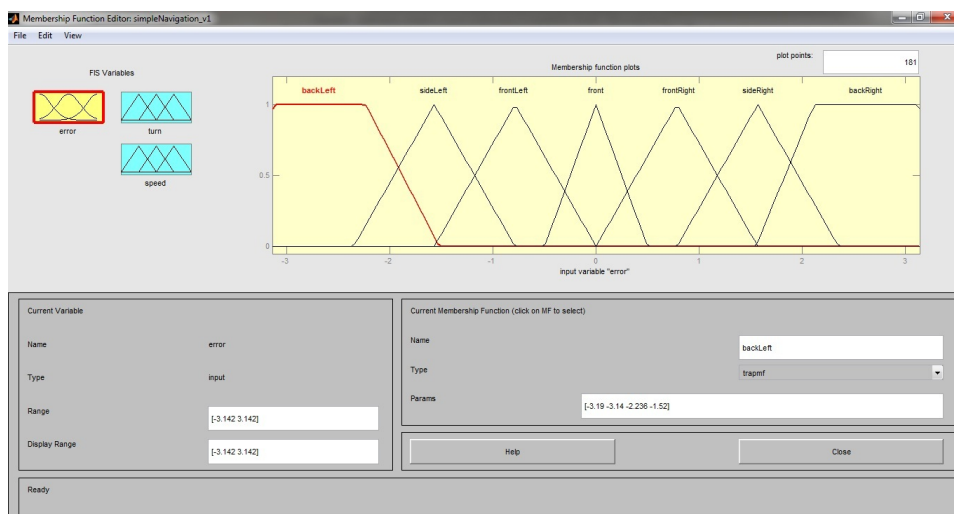
```
>> fuzzy
```

ανοίγουμε το παράθυρο του *Fuzzy Logic Editor (FIS.)* της MATLAB (Σχήμα 28).

Χρησιμοποιώντας τη γραμμή εργαλείων επιλέγουμε



Σχήμα 28: FUZZY LOGIC EDITOR-FIS.



Σχήμα 29: FIS-Σύνολο συναρτήσεων συμμετοχής εισόδου.

▷ EDIT→ADD VARIABLES→INPUT.

Παρατηρούμε ότι εμφανίζεται δεύτερη είσοδος στο σύστημά.

Ακολουθώντας την ίδια διαδικασία, εισάγουμε συνολικά 1 είσοδο. Ο αριθμός των εισόδων μπορεί να ποικίλει ανάλογα με την εφαρμογή που σκοπεύει να προσομοιώσει ο προγραμματιστής. Στο παράδειγμά μας οι εισοδοί είναι οι αισθητήρες γύρω από το όχημα, ομαδοποιημένοι ως "error".

Ονομάζουμε τις Εισόδους (αυτό γίνεται επιλέγοντας την είσοδο που θέλουμε στο Block διάγραμμα του FIS, και αλλάζοντας το όνομα στο πεδίο "Name"). Ξεκινώντας από πίσω αριστερά, κινούμαστε περιμετρικά του οχήματος, με τη φορά των δεικτών του ρολογιού (clockwise): τα ονόματα που χρησιμοποιούμε είναι *backLeft*, *sideLeft*, *frontLeft*, *Front*, *frontRight*, *sideRight*, *backRight*.

Στη συνέχεια θα ρυθμίσουμε τον τύπο συναρτήσεων συμμετοχής που θα χρησιμοποιήσουμε και το εύρος τιμών μέσα στο οποίο κινούνται οι συναρτήσεις συμμετοχής.

Αρχικά επιλέγουμε τον τύπο των συναρτήσεων συμμετοχής που θα χρησιμοποιήσουμε στο πεδίο *Type*. Στο *dropdown menu* που εμφανίζεται επιλέγουμε τον τύπο συναρτήσεων όπως εμφανίζονται στον Πίνακα 1.

Στο πεδίο *Range* εισάγουμε τις τιμές [-3.14 3.14] και στο πεδίο *Params* εισάγουμε τις τιμές του Πίνακα 2.

MEMBER FUNCTION	Type
backLeft, backRight	trapmf
sideLeft, sideRight, frontLeft, frontRight, front	trimf

Πίνακας 1: Οι τύποι των συναρτήσεων συμμετοχής που χρησιμοποιήθηκαν στην είσοδο "error".

MEMBER FUNCTION	Params
backLeft	[-3.19 -3.14 -2.236 -1.52]
sideLeft	[-2.37 -1.59 -0.802]
frontLeft	[-1.571 -0.7854 0]
front	[-0.5 0 0.5]
frontRight	[0 0.7854 1.571]
sideRight	[0.7854 1.571 2.356]
backRight	[1.55 2.12 3.14 3.2]

Πίνακας 2: Παράμετροι που χρησιμοποιήθηκαν στις συναρτήσεις συμμετοχής της εισόδου "error".

Ως αυτό το σημείο έχουμε δώσει εύρος τιμών (*Range*) στις συναρτήσεις συμμετοχής των εισόδων του FUZZY LOGIC CONTROLLER, έχουμε καθορίσει τον τύπο των συναρτήσεων συμμετοχής (*Member Functions*) και έχουμε καθορίσει τις παραμέτρους (*Params*) των συναρτήσεων συμμετοχής.

Περνάμε στο σχεδιασμό των εξόδων του ελεκτή. Χρησιμοποιούμε δύο συναρτήσεις συμμετοχής ως εξόδους του συστήματος.

Αρχικά χρησιμοποιώντας τη γραμμή εργαλείων επιλέγουμε

▷ EDIT→ADD VARIABLES→OUTPUT.

Παρατηρούμε ότι εμφανίζεται δεύτερη έξοδος στο σύστημά.

Στο πεδίο *Name*, μετονομάζουμε τις δύο εξόδους του ελεκτή σε *Turn* και *Speed*.

Ο τύπος συναρτήσεων συμμετοχής που αποτελούν την έξοδο *Turn* εμφανίζονται στον Πίνακα 3.

MEMBER FUNCTION	Type
turnLeft, turnRight	trapmf
littleLeft, littleRighth, straight	trimf

Πίνακας 3: Τύποι συναρτήσεων συμμετοχής που αποτελούν την έξοδο *Turn*.

Στο πεδίο *Range* εισάγουμε τις τιμές [-0.5 0.5] και στο πεδίο *Params* τις τιμές του Πίνακα 4.

Ακολουθώντας την ίδια διαδικασία δημιουργούμε και τις συναρτήσεις συμμετοχής που αποτελούν την έξοδο "speed", Πίνακας 5.

Στο πεδίο *Range* εισάγουμε τις τιμές [0 1] και στο πεδίο *Params* τις τιμές του Πίνακα 6.

Στα Σχήματα 30,31 εμφανίζονται οι συναρτήσεις μεταφοράς που αποτελούν τις εξόδους "Turn" και "Speed" αντίστοιχα.

Σ' αυτό το σημείο έχουμε ολοκληρώσει την σχεδίαση των εισόδων και των εξόδων του ασαφή ελεκτή που σχεδιάζουμε. Ο συγκεκριμένος ελεκτή έχει μία είσοδο· το σφάλμα θέσης του οχήματος ("error") και δύο εξόδους την στροφή και την ταχύτητα ("Turn" και "Speed" αντίστοιχα.)

Το επόμενο βήμα είναι να ορίσουμε τους κανόνες (rules) του ελεκτή ώστε να πετύχουμε το επιθυμητό αποτέλεσμα κατά τη διάρκεια της προσομοίωσης. Πιο συγκεκριμένα στο παράθυρο *Membership Function Editor* επιλέγουμε:

MEMBER FUNCTION	Params
turnLeft	[-0.5443 -0.5 -0.35 -0.1]
littleLeft	[-0.4 -0.25 0]
straight	[-0.1627 0 0.1601]
littleRight	[0 0.25 0.4]
turnRight	[0.1 0.35 0.5 0.5443]

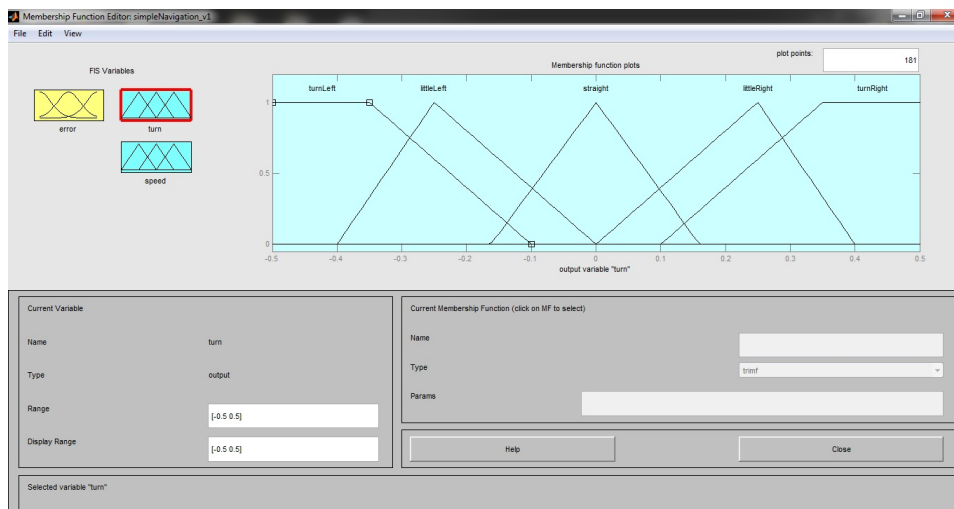
Πίνακας 4: Παράμετροι που χρησιμοποιήθηκαν στις συναρτήσεις συμμετοχής της εξόδου "error."

MEMBER FUNCTION	Type
slow, medium, fast	trimf

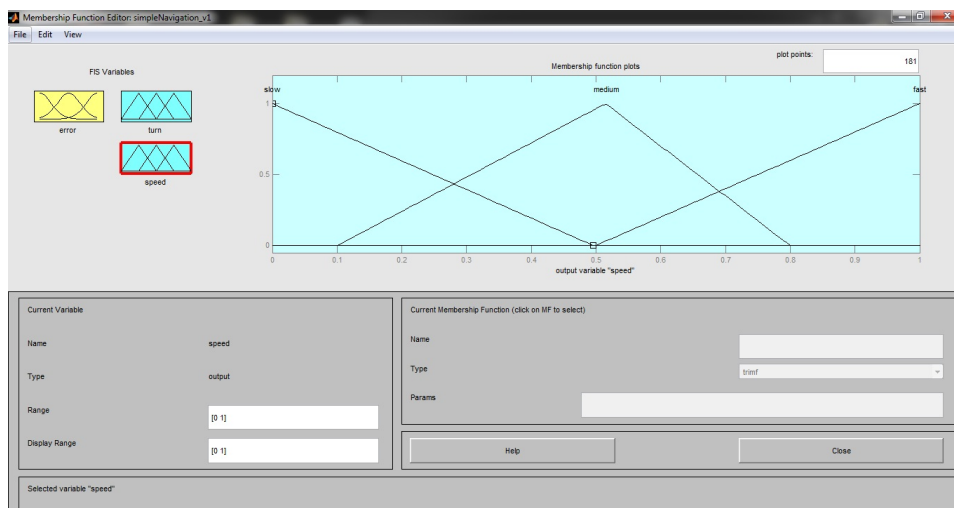
Πίνακας 5: Τύποι συναρτήσεων συμμετοχής που αποτελούν την έξοδο "Speed".

MEMBER FUNCTION	Params
slow	[-0.4 0 0.496]
medium	[0.1 0.5146 0.8]
fast	[0.499 1 1.01]

Πίνακας 6: Παράμετροι που χρησιμοποιήθηκαν στις συναρτήσεις συμμετοχής εισόδου "Speed."



Σχήμα 30: Συναρτήσεις συμμετοχής της εξόδου "Turn".



Σχήμα 31: Συναρτήσεις συμμετοχής της εξόδου "Speed."

▷ EDIT→RULES ή πατώντας Ctrl+3

Το παράθυρο που ανοίγει μας επιτρέπει να ορίσουμε αναλυτικά τους κανόνες που διέπουν τον ελεκτή που σχεδιάζουμε.

Ουσιαστικά, με απλούς προγραμματιστικούς κανόνες, όπως IF [input argument] THEN [output arguments] σε συνδυασμό με AND/OR τελεστές και με “βάρος” (Weight) κάθε κανόνα, μπορούμε να δημιουργήσουμε σύντομους κανόνες.

Ενδέχεται βέβαια, σε συστήματα με μεγάλο αριθμό εισόδων/εξόδων ή με μεγάλο αριθμό συναρτήσεων συμμετοχής, να προκύπτουν δυσκολίες στην ομαδοποίηση των κανόνων και στην επιλογή βάρους του κάθε κανόνα. Παρόλαυτά η ευκολίες που μας προσφέρει το περιβάλλον προσομοίωσης είναι πολλές γι’ αυτό και επιλέγεται. Επίσης σημαντική είναι η δυνατότητα να μελετήσουμε τη συμπεριφορά ενός συστήματος σε βάθος, χρησιμοποιώντας πολλές διαφορετικές παραμέτρους.

Αναλύοντας για παράδειγμα τον πρώτο κανόνα, με βάση το γενικό κανόνα:

```
IF(input argument)THEN(output argument)AND/OR(output argument)(weight)
```

```
IF(error is frontLeft)/THEN(turn is Left)AND(speed is medium)(1)
```

Δηλαδή όταν το σφάλμα κίνησης του οχήματος είναι (εμπρός αριστερά) ΤΟΤΕ (στρίψε αριστερά) ΚΑΙ (η ταχύτητα είναι μεσαία.) (βάρος κανόνα 1)

Με ανάλογο τρόπο ερμηνεύουμε όλες τις εντολές του ελεκτή μίας και έχουν όλες παρόμοιο τρόπο σύνταξης.

Έχοντας τελειώσει τη σύνταξη κανόνων του ελεκτή μπορούμε να ελέγξουμε τη συμπεριφορά των εξόδων σε σχέση με την είσοδο, επιλέγοντας:

▷ VIEW→SURFACE ή πατώντας Ctrl+6.

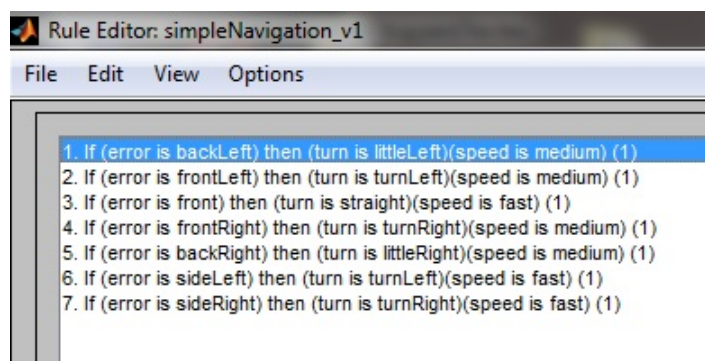
Εμφανίζεται το παράθυρο Surface Viewer, στο οποίο επιλέγοντας την είσοδο και έξοδο που θέλουμε, βλέπουμε την γραφική παράσταση εισόδου/εξόδου. Στο (Σχήμα 33) βλέπουμε πώς συμπεριφέρεται η Στρέψη του οχήματος (έξοδος) ανάλογα με το σφάλμα θέσης (είσοδος). Παρατηρούμε ότι όταν το όχημα έχει μηδενικό σφάλμα θέσης, τότε κινείται με μηδενική στρέψη.

Στο Σχήμα 34 βλέπουμε πως συμπεριφέρεται η ταχύτητα του οχήματος σε σχέση με το σφάλμα θέσης, παρατηρούμε ότι η μέγιστη ταχύτητα επιτυγχάνεται όταν το σφάλμα θέσης είναι μηδενικό.

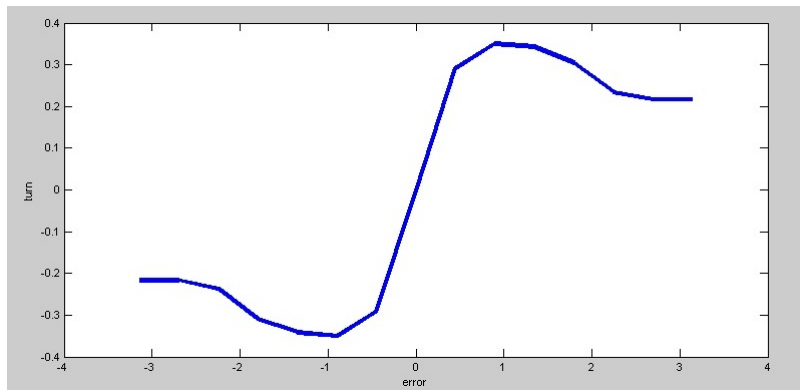
Παρατηρούμε ότι η γραφικές παραστάσεις εξόδου, ακολουθούν το θεωρητικό υπόβαθρο της παραγράφου 4.3.1 και συγκεκριμένα τείνουν να μοιάζουν με τις γραφικές παραστάσεις του Σχήματος 25β'. Δηλαδή, πλησιάζουν την συμπεριφορά των ασαφών συνόλων, δηλαδή το βαθμό συμμετοχής κάθε στοιχείου του συνόλου με το σύνολο· αυτό εμφανίζεται με τη μορφή “καμπυλότητας” στο γράφημα του Σχήματος 33.

Θεωρώντας ότι έχουμε ικανοποιητική συμπεριφορά στον ελεκτή, κάτι που ορίζεται από τις συνθήκες στις οποίες θα λειτουργεί ο ελεκτή, θα συνεχίσουμε χρησιμοποιώντας τον για την πλοήγηση ενός οχήματος σε κάποιο στόχο.

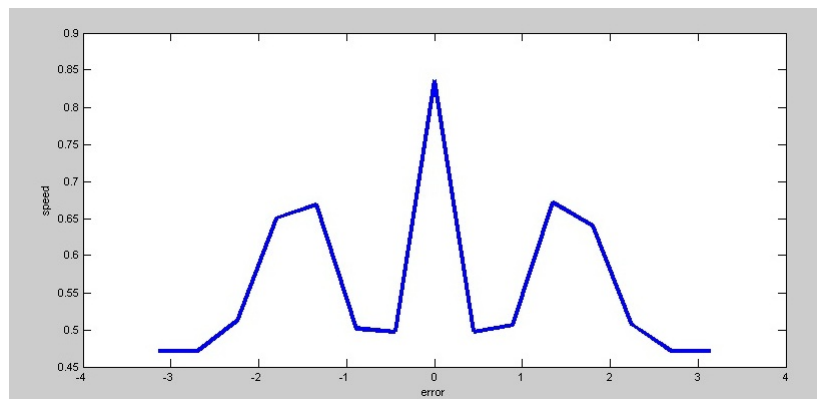
Μέσα σε περιβάλλον MATLAB θα χρησιμοποιήσουμε το ROBOTICS TOOLBOX του Peter



Σχήμα 32: Κανόνες του Ασαφή Ελεκτή.



Σχήμα 33: Γραφική παράσταση Σφάλματος/Στρέψης.



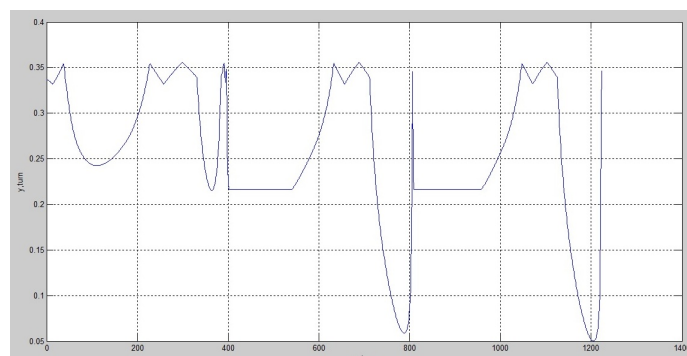
Σχήμα 34: Γραφική παράσταση Σφάλματος/Ταχύτητας.

Corke απο το οποίο θα χρησιμοποιήσουμε την ρουτίνα *Vehicle*, στη συνέχεια θα γράψουμε ένα M-FILE χρησιμοποιώντας τον Ασαφή Ελεκτή που σχεδιάσαμε σε συνδιασμό με την *Vehicle* του Peter Corke. Για να δοκιμάσουμε την ακρίβεια πλοήγησης δώσαμε με γεννήτριες τυχαίων τιμών (`rand[...]`), τα σημεία (x, y) του καρτεσιανού επιπέδου. Στην πρώτη δοκιμή δώσαμε 3 ζεύγη σημείων και στη δεύτερη δοκιμή 5 ζεύγη σημείων. Ο κώδικας που χρησιμοποιήθηκε παρατίθεται αναλυτικά στο Παράρτημα.

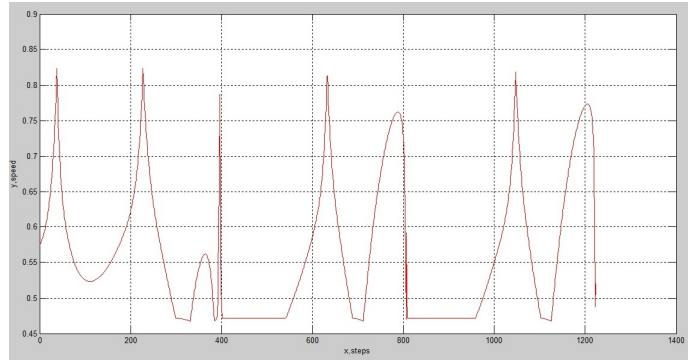
Τα αποτελέσματα που συλλέξαμε μετά την προσομοίωση.

Στο σχήμα 37 μπορούμε να παρατηρήσουμε και σημεία που χρειάζονται βελτίωση για παράδειγμα διαδρομές που θεωρούμε ότι θα έπρεπε να είχε πάρει το όχημά μας και δεν επιβεβαιώθηκε πειραματικά.

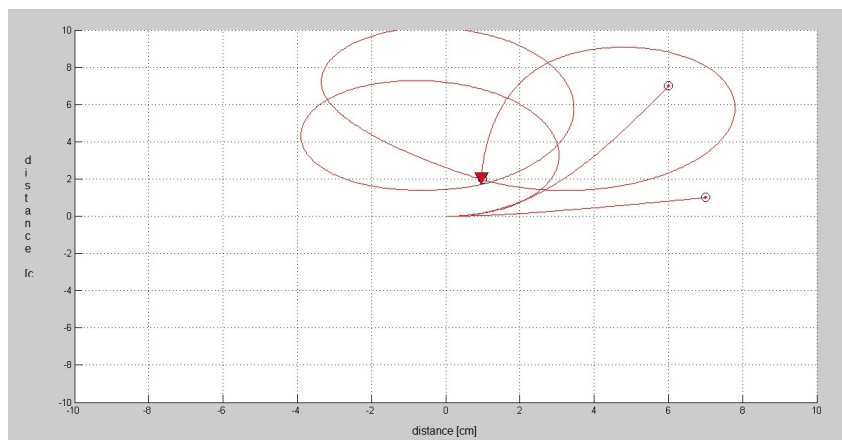
Στη συνέχεια θα τρέξουμε ακόμα μία προσομοίωση, αυτή τη φορά με 5 ζεύγη σημείων. Τα αποτελέσματα εμφανίζονται στο σχήμα 38.



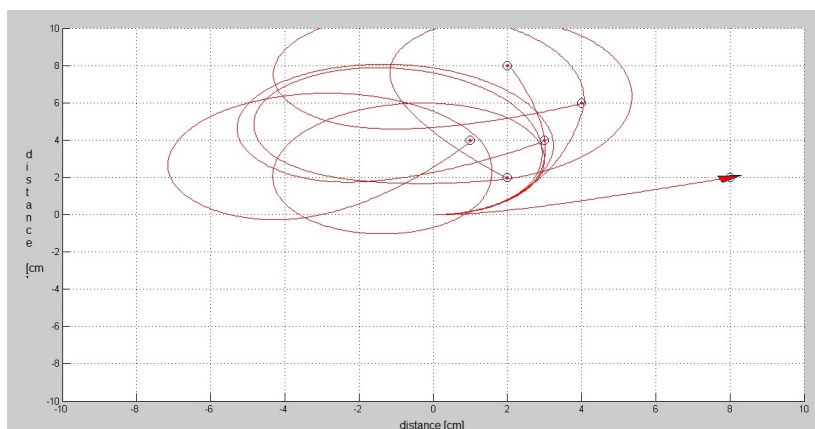
Σχήμα 35: Γραφική παράσταση στρέψης του Οχήματος προς βήματα (steps/μονάδα χρόνου).



Σχήμα 36: Γραφική παράσταση ταχύτητας του Οχήματος προς βήματα (steps/μονάδα χρόνου).



Σχήμα 37: Αποτελέσματα προσομοίωσης με 3 ζεύγη σημείων. Στην εικόνα εμφανίζουμε το ίχνος της κίνησης που ακολούθησε το όχημα μέχρι να φτάσει στο στόχο. Παρατηρήστε το μέγεθος της διαδρομής για να φτάσει στο 3ο σημείο. Αυτό οφείλεται στο ότι οι τροχοί έχουν συγκεκριμένη μέγιστη γωνία στρέψης και ότι το οχημά μας πρέπει να πλησιάσει το στόχο χωρίς να αλλάξει διεύθυνση κίνησης· στην προκειμένη περίπτωση κινείται μόνο προς τα εμπρός.



Σχήμα 38: Αποτελέσματα προσομοίωσης με 5 ζεύγη σημείων. Το όχημα επιδιώκει να φτάσει στο δοσμένο σημείο ακολουθώντας τη συντομότερη διαδρομή, εφόσον τα μηχανικά μέρη του οχήματος το επιτρέπουν. Χαρακτηριστικό παράδειγμα η τιμή (8,2) που το όχημα κατευθύνεται σχεδόν ευθεία.

7 Παράρτημα

```
function [history_v1]=move_v2(xt, yt)

oxima=Vehicle(diag([0.005,0.5*pi/180].^2));
oxima.maxspeed=1;
targetx = xt;
targety = yt;
distance=100;

%Read Fuzzy controller
initfis=readfis('simpleNavigation_v1.fis');

for targetx = randi(8,1,5);
    for targety = randi(8,1,5);
        counter=1;

while distance > 0.05,           %Starting Control Loop
    output=oxima.x;             %Position and orientation of vehicle
    oximaX=output(1);          %Y coordinate
    oximaY=output(2);          %X coordinate
    oximatheta=output(3);      %Orientation of vehicle
    history_v1.x(counter)=oximaX;
    history_v1.y(counter)=oximaY;
    history_v1.th(counter)=oximatheta;
    neoX=targetx-oximaX;
    neoY=targety-oximaY;

    %Calculate target's distance
    distance=sqrt(neoX^2 + neoY^2);

    %Calculate Vehicle-Target angle
    newTheta=atan2(neoY, neoX);
    err=newTheta - oximatheta    %Calculate angle error

        %Angle error normalization loop
        while abs(err)>=pi,
            if err < (-1*pi),
                err=err+2*pi;
            end
            if err >=pi,
                err=err-2*pi;
            end
        end
    end

f_command=evalfis(err, initfis); %Fuzzy controller calculations
    history_v1.v1(counter)=f_command(1);
    history_v1.v2(counter)=f_command(2);
    oxima.update(fliplr(f_command)); %Send output to robot
    %Plot results
```

```

oxima.plot(output,'scale',0.15,'colour','b','fill');
hold on
plot(targetx,targety,'bo','MarkerSize',10); %Plot target
plot(targetx,targety,'r.','MarkerSize',6); %Plot target
hold off
grid on; axis([-10 10 -10 10])
drawnow;
counter=counter+1;
end %End loop
end
end

hold on
oxima.plot_xy('r'); %Plot trajectory line
figure

plot(history_v1.v1)
grid
xlabel('x, steps')
ylabel('y, turn')
figure
plot(history_v1.v2,'r')
grid
xlabel('x, steps')
ylabel('y, speed')

```


Αναφορές

- [1] Γιάννης Θεοδώρου, *Εισαγωγή στην Ασαφή Λογική*, Εκδόσεις Τζιόλα 2010
- [2] Γεώργιος Μαρχαντωνάκης, *Κίνηση ρομποτικού οχήματος με χρήση ασαφούς λογικής*, ΤΕΙ Κρήτης 2014
- [3] Εμμανουήλ Καλυκάκης, *Έλεγχος ρομποτικού οχήματος και αποφυγή δυναμικών και στατικών εμποδίων με τη βοήθεια ελεγκτών ασαφούς λογικής*, ΤΕΙ Κρήτης 2014
- [4] Κωνσταντίνος Δεδούσης, *Σχεδιασμός Ασαφούς Ελεκτή για πλοήγηση ρομποτικών οχημάτων*, ΤΕΙ Κρήτης 2013
- [5] Βολογιαννίδης Σταύρος, *Ευφυής Έλεγχος, Θεωρία και Εφαρμογές*, ΤΕΙ Σερρών 2009
- [6] Ιωάννης Βλαχάβας, Πέτρος Κεφάλας, Νικόλαος Βασιλειάδης, Φώτης Κόκκορας, Ηλίας Σακελλαρίου, *Ευφυής Έλεγχος*, Εκδόσεις Πανεπιστημίου Μακεδονίας 2011
- [7] Peter Corke, *Robotics Vision and Control: Fundamental Algorithms in Matlab*, Springer-Verlag Berlin Heidelberg 2011
- [8] <http://en.wikipedia.org/wiki/Portal:Robotics>
- [9] <http://en.wikipedia.org/wiki/Portal:Electronics>
- [10] <http://www.coppeliarobotics.com>
- [11] Andrian Biran , Moshe Breiner, *Matlab 6 for Engineers* Εκδόσεις Τζιόλα 2003
- [12] *Khepera III robot Manual*, K-Team 2013