



**Τ.Ε.Ι ΚΡΗΤΗΣ - ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.**

**Έλεγχος ρομποτικού οχήματος με την χρήση ασαφούς λογικής**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Μαρκαντωνάκης Γεώργιος**

**ΧΑΝΙΑ 2014**

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα εργασία εκπονήθηκε στο *Εργαστήριο Ευφρών συστημάτων και ρομποτικής* του Τμήματος Ηλεκτρονικών μηχανικών, κατά το χρονικό διάστημα Σεπτέμβριος 2013 – Αύγουστος 2014, υπό την επίβλεψη του Επικ. Καθηγητή Δοϊτσίδα Ελευθέριου.

Με την ολοκλήρωση της εργασίας αυτής, θα ήθελα να εκφράσω την ειλικρινή μου εκτίμηση στον επιβλέποντα καθηγητή μου Δρ Δοϊτσίδα Ελευθέριο για την καθοδήγηση και ενθάρρυνση του κατά τη διάρκεια της πτυχιακής μου εργασίας.

# Πίνακας περιεχομένων

1	Εισαγωγή.....	7
2	LEGO Mindstorms.....	8
2.1	Η σειρά Lego Mindstorms.....	8
2.2	Εξαρτήματα.....	8
2.2.1	Brick NXT.....	8
2.2.2	Σερβομηχανή.....	9
2.2.3	Αισθητήρα επαφής.....	9
2.2.4	Αισθητήρα φωτός.....	10
2.2.5	Αισθητήρα ήχου.....	10
2.2.6	Αισθητήρα υπερήχων.....	11
2.2.7	Αισθητήρα πυξίδας.....	11
2.2.8	Γυροσκόπιο.....	11
2.3	Περιβάλλον προγραμματισμού του NXT.....	13
2.4	Παράδειγμα ανάπτυξης, μεταφοράς και εφαρμογής προγράμματος.....	14
	LEGO DIGITAL DESIGNER.....	18
2.5	Σχεδιασμός και ανάπτυξη πειραματικού οχήματος.....	20
3	Η εργαλειοθήκη RWTH- Mindstorms NXT του MATLAB.....	23
3.1	Γενικά για την εργαλειοθήκη.....	23
3.2	Παραδείγματα χρήσης της εργαλειοθήκης RWTH-Mindstorms NXT.....	24
4	Ασαφής Λογική.....	28
4.1	Γενικά περί ασαφούς Λογικής.....	28
4.2	Κλασικό και Ασαφές σύνολο.....	28
4.3	Λεκτικοί τελεστές.....	31
4.4	Ασαφή συστήματα.....	32
5	Ασαφής έλεγχος έντροχου ρομποτικού οχήματος.....	38
5.1	Αντίληψη του χώρου.....	38
5.1.1	Οδομετρία.....	38
5.1.2	Πλοήγηση.....	40
5.2	Ασαφείς ελεγκτές.....	40
5.2.1	Ελεγκτής αποφυγής εμποδίων.....	40
5.2.2	Ελεγκτής αποφυγής εμποδίων και διόρθωση κατεύθυνσης.....	42
6	Συμπεράσματα.....	47
7	Παράρτημα.....	48
8	BIBΛΙΟΓΡΑΦΙΑ.....	53

## Ευρετήριο Εικόνων

Εικόνα 1 NXT-BRICK Πηγή:lego.com.....	7
Εικόνα 2: Σερβοκινητήρας. (Πηγή:lego.com).....	8
Εικόνα 3: Αισθητήρας επαφής (Πηγή:lego.com).....	8
Εικόνα 4: Αισθητήρας φωτός (Πηγή:lego.com).....	9
Εικόνα 5: Αισθητήρας ήχου Πηγή:lego.com.....	9
Εικόνα 6: Αισθητήρας υπερήχων (Πηγή:lego.com).....	10
Εικόνα 7: Ψηφιακή πυξίδα Πηγή:lego.com.....	10
Εικόνα 8: Mindstorm NXT kit. Πηγή: RoboRAVE.....	11
Εικόνα 9: Το γραφικό περιβάλλον του NXT-G. Αριστερά βρίσκεται η παλέτα προγραμματισμού ενώ στο μέσο η περιοχή εργασίας.....	12
Εικόνα 10:Κίνηση οχήματος εμπρός και επιστροφή στην αρχική του θέση.....	13
Εικόνα 11:Επιλογή κίνησης της παλέτας NXT-G.....	14
Εικόνα 12: Ρύθμιση παραμέτρων κίνησης.....	14
Εικόνα 13: Δύο εντολές κίνησης σε σειρά.....	15
Εικόνα 14: Ρύθμιση παραμέτρων κίνησης δεύτερης εντολής.....	15
Εικόνα 15: Μέθοδος εισαγωγής του προγράμματος στο NXT-BRICK.....	16
Εικόνα 16:Οθόνη του NXT-BRICK. ....	16
Εικόνα 17: Παλέτα του Lego Digital Designer (LDD).....	17
Εικόνα 18: Επιλογή εξαρτημάτων και τοποθέτηση τους στο παράθυρο δημιουργίας.....	18
Εικόνα 19: Κατασκευή αντικειμένου.....	18
Εικόνα 20: Κατασκευή αντικειμένου.....	19
Εικόνα 21: Δύο κινητήρες περιστροφής τοποθετημένοι παράλληλα και σε αντίθετη φορά.....	19
Εικόνα 22: Αισθητήρες υπερήχων.....	20
Εικόνα 23: Αυτοκινούμενο όχημα σε διάδρομο.....	20
Εικόνα 24:Τυφλό σημείο.....	20
Εικόνα 25: Τελική υλοποίηση οχήματος.....	21
Εικόνα 26: Τελική υλοποίηση στο πρόγραμμα Lego digital designer.....	21
Εικόνα 27: Γενική επισκόπηση της επικοινωνίας μεταξύ MATLAB και υλικού NXT. ....	23
Εικόνα 28:Βαθμός συμμετοχής $\mu A(\chi)=0.5$ .....	29
Εικόνα 29: Ασαφή σύνολα: Κρύο, Χλιαρό, Ζεστό.....	29
Εικόνα 30:Τομή και ένωση ασαφών συνόλων.....	30
Εικόνα 31:Ασαφή σύνολα Ζεστό, Κρύο, Καυτό.....	31
Εικόνα 32:Ασαφή σύνολα Κλειστό, Μέτριο, Μέγιστο.....	32
Εικόνα 33: ΟΙ είσοδοι και οι έξοδοι μετά την ενεργοποίηση των κανόνων.....	33
Εικόνα 34:Ένωση των συνόλων εξόδου Μέτριο και Μέγιστο.....	33
Εικόνα 35:Διαδικασία αποασαφοποίησης εξόδου.....	33
Εικόνα 36:Αποτελέσματα εξόδου με τη μέθοδο του κέντρου βάρους και του μεγίστου.....	35
Εικόνα 37: Το μεταξόνιο (b), τα διαστήματα που έχουν κινηθεί οι τροχοί (UR και UL) και το όχημα συνολικά(UD).....	37
Εικόνα 38Εύρεση $\chi'$ και $\gamma'$ .....	38
Εικόνα 39:Προσδιορισμός του σφάλματος κατεύθυνσης.....	39
Εικόνα 40Σχεδιάγραμμα ελεγκτή.....	40
Εικόνα 41:Λεκτική μεταβλητή εισόδου.....	40
Εικόνα 42:Λεκτική μεταβλητή εξόδου.....	41
Εικόνα 43:Σχεδιάγραμμα ελεγκτή.....	42
Εικόνα 44:Μεταβλητή εισόδου “σφάλμα γωνίας”.....	42
Εικόνα 45:Αποφυγή εμποδίων και διόρθωση πορείας 1.....	43
Εικόνα 46:Αποφυγή εμποδίων και διόρθωση πορείας 2.....	44
Εικόνα 47:Πρώτη τροχιά οχήματος.....	44
Εικόνα 48:Εισαγωγή δεύτερης τροχιάς.....	45
Εικόνα 49:Εισαγωγή τρίτης τροχιάς.....	45

Εικόνα 50	Στιγμιότυπο από το lego Degital Designer 1	50
Εικόνα 51	Στιγμιότυπο από το lego Degital Designer 2	50
Εικόνα 52	Στιγμιότυπο από το lego Degital Designer 3	50
Εικόνα 53	Στιγμιότυπο από το lego Degital Designer 4	50
Εικόνα 54	Στιγμιότυπο από το lego Degital Designer 5	51

# 1 Εισαγωγή

Η παρούσα πτυχιακή μελετάει ένα σύστημα ελέγχου ασαφούς λογικής για ένα έντροχο όχημα. Το όχημα είναι σε θέση να πλοηγείται από ένα σημείο του χώρου σε ένα άλλο αποφεύγοντας εμπόδια.

Για την κατασκευή του οχήματος χρησιμοποιήθηκε η ρομποτική σειρά NXT Mindstorm της Lego. Το όχημα περιλαμβάνει την προγραμματιζόμενη μονάδα NXT Brick, τρεις αισθητήρες υπερήχων για τον εντοπισμό αντικειμένων και δύο σερβοκινητήρες.

Για την εύρεση της θέσης του οχήματος στο χώρο χρησιμοποιήθηκε η μέθοδος της οδομετρίας. Με τον όρο οδομετρία εννοούμε την χρήση των δεδομένων από τους αισθητήρες στους σερβοκινητήρες ώστε με την χρήση των κατάλληλων εξισώσεων κίνησης να εκτιμηθεί η αλλαγή της θέσης του οχήματος στη πάροδο του χρόνου. Πρέπει να σημειωθεί πως αυτή η μέθοδος είναι ευαίσθητη σε σφάλματα λόγο του τζόγου των κινητήρων και των τροχών.

Ο ασαφής ελεγκτής έχει ως είσοδο την απόσταση από τα τυχόν εμπόδια και το σφάλμα κατεύθυνσης του οχήματος σε σχέση με το σημείο-στόχο. Μια σειρά από κανόνες πλοήγησης ενεργοποιούνται και ο ελεγκτής δίνει σαν έξοδο τιμές ισχύος για τους κινητήρες του οχήματος.

## 2 LEGO Mindstorms

### 2.1 Η σειρά Lego Mindstorms

Το [LEGO Mindstorms NXT](#) είναι ένα προγραμματιζόμενο σετ ρομποτικής που κυκλοφόρησε από την LEGO στα τέλη Ιουλίου του 2006 και αποτελεί μία βελτιωμένη έκδοση του LEGO Mindstorms RCX (Robotic Command eXplorers). Το σετ περιλαμβάνει λογισμικό (software) και υλικό (hardware) για την δημιουργία προγραμματιζόμενων ρομπότ, προσαρμοζόμενα στην επιθυμία του χρήστη. Περιλαμβάνει υλικά κατασκευής για τη σύνθεση κατασκευών, όπως πλαστικά τουβλάκια (τουβλάκια - σύνδεσμοι), άξονες, γρανάζια, και άλλα πλαστικά εξαρτήματα σε διάφορα μεγέθη και χρώματα. Τα δομικά υλικά κατασκευάζονται με υψηλό βαθμό ακρίβειας και αντοχής, για τη συναρμολόγηση ανθεκτικών κατασκευών. Έχει τη δυνατότητα να εξομοιώσει σχεδόν όλους τους σύγχρονους αυτοματισμούς και να προσεγγίσει - ικανοποιητικά - συστήματα αυτόματου ελέγχου. Επίσης, το βασικό σετ περιλαμβάνει και κάποια ηλεκτρονικά μέρη, τα οποία δίνουν τη δυνατότητα σε ένα ρομπότ MINDSTORMS να ζωντανέψει και να εκτελέσει διάφορες διαδικασίες.

Μπορεί να μοιάζει ως ένα ενδιαφέρον παιχνίδι αλλά στην πραγματικότητα είναι ένα ολοκληρωμένο περιβάλλον ρομποτικής. Μπορεί να χρησιμοποιηθεί από σχολεία, πανεπιστήμια αλλά και χομπίστες. Με αυτό μπορείς να κατασκευάσεις από απλά ρομπότ μέχρι ότι πιο πολύπλοκο με μόνο περιορισμό τη φαντασία του χρήστη.

### 2.2 Εξαρτήματα

Τα Lego Mindstorms περιλαμβάνουν ένα σύνολο αισθητήρων, επενεργητών και μονάδων ελέγχου που μπορούν να προγραμματιστούν. Στη συνέχεια θα παρουσιαστούν συνοπτικά, τα βασικά τους χαρακτηριστικά.

#### 2.2.1 Brick NXT

Το προγραμματιζόμενο τούβλο NXT αποτελεί τη βασική μονάδα ελέγχου της κατασκευής που υλοποιείται με τη βοήθεια του Lego Mindstorms Kit. Πρόκειται για ένα ευφύες, ελεγχόμενο από υπολογιστή τούβλο, το οποίο δίνει την δυνατότητα στη ρομποτική μας κατασκευή να εκτελέσει μια σειρά από διαφορετικές διαδικασίες.

Το προγραμματιζόμενο τούβλο έχει τέσσερις θύρες εισόδου για την ένωση των αισθητήρων - θύρες 1, 2, 3 και 4 και διαθέτει τρεις θύρες εξόδου για των έλεγχο των μηχανών- θύρες A, B και C. Το brick διαθέτει μια μονόχρωμη LCD



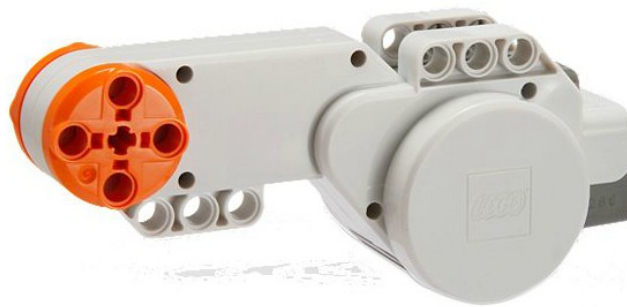
Εικόνα 1 NXT-BRICK Πηγή:lego.com

οθόνη, μία θύρα usb, μεγάφωνο ηχητικής ποιότητας 8 kHz με εύρος καναλιού 8-bit και συχνότητα δειγματοληψίας 2-16 KHz και τέσσερα κουμπιά που μπορούν να χρησιμοποιηθούν για περιήγηση στο μενού. Διαθέτει έναν 32-bit μικροελεγκτή με 256 KB FLASH μνήμη και 64 KB μνήμη RAM, καθώς και ένα 8-Bit μικροελεγκτή και υποστήριξη bluetooth (Εικόνα 1).

### 2.2.2 Σερβομηχανή

Οι βασικοί κινητήρες της σειράς NXT, είναι οι σερβομηχανές που περιλαμβάνουν ένα ενσωματωμένο αισθητήρα περιστροφής, ο οποίος μπορεί να πληροφορήσει το μικροεπεξεργαστή ανα πάσα στιγμή, για τη θέση στην οποία έχει περιστραφεί ένας κινητήρας.

Οι αισθητήρες αυτοί μπορούν να μετρήσουν τη περιστροφή του κινητήρα σε μοίρες (degrees) ή σε πλήρεις περιστροφές (μία πλήρη περιστροφή αντιστοιχεί σε 360 μοίρες) με ακρίβεια +/- 1 μοίρα (Εικόνα 2).



Εικόνα 2: Σερβοκινητήρας. (Πηγή:lego.com)

### 2.2.3 Αισθητήρα επαφής

Με τον αισθητήρα επαφής η ρομποτική κατασκευή είναι σε θέση να αντιληφθεί πότε έχει έρθει σε επαφή ή σταμάτησε να έχει επαφή με κάποιο αντικείμενο στο κοντινό του περιβάλλον όπου και δραστηριοποιείται. Ο τρόπος λειτουργίας του είναι αρκετά απλός. Με το πάτημα του πλήκτρου ένα κύκλωμα κλείνει και διέρχεται από ηλεκτρικό ρεύμα. Το NXT είναι σε θέση να ανιχνεύσει αυτή τη ροή, και το πρόγραμμα μπορεί να διαβάσει την κατάσταση του αισθητήρα αφής (Εικόνα 3)[[Dave Astolfo et al.](#)].



Εικόνα 3: Αισθητήρας επαφής (Πηγή:lego.com)



## 2.2.4 Αισθητήρα φωτός

Η λειτουργία του αισθητήρα φωτός είναι να ανιχνεύει το φως του περιβάλλοντός του και να μετράει την τιμή της έντασης του. Μπορεί να χρησιμοποιηθεί για ένα ευρύ φάσμα εφαρμογών. Η πιο σημαντική διαφορά ανάμεσα στον αισθητήρα αφής και αισθητήρα φωτός είναι ότι ο τελευταίος επιστρέφει πολλές πιθανές τιμές αντί για μια απλή on / off κατάσταση. Αυτές οι τιμές εξαρτώνται από την ένταση του φωτός που χτυπά τον αισθητήρα κατά τη στιγμή που παίρνεται η μέτρηση. Οι προεπιλεγμένες ρυθμίσεις επιστρέφουν τιμή σε μορφή ποσοστού από 0 μέχρι 100.

Επίσης διαθέτει ένα μικρό κόκκινο LED το οποίο παρέχει μια σταθερή πηγή φωτός, επιτρέποντας έτσι να μετρηθεί το ανακλώμενο φως που επιστρέφει στον αισθητήρα. Με αυτό το τρόπο μπορεί να μετρηθεί η απόσταση από ένα αντικείμενο (Εικόνα 4)[[Dave Astolfo et al.](#)].



Εικόνα 4: Αισθητήρας φωτός  
(Πηγή:lego.com)

## 2.2.5 Αισθητήρα ήχου

Ο αισθητήρας ήχου έχει την ικανότητα να μετράει τα επίπεδα του θορύβου σε db (decibels) και σε dBA (συχνότητες περίπου 3-6 kHz όπου το ανθρώπινο αυτί είναι πιο ευαίσθητο). Μπορεί να ξεχωρίσει θορύβους ανάλογα με το επίπεδο έντασης τους. Ο τρόπος κατασκευής του είναι παρόμοιος με τον αισθητήρα επαφής (Εικόνα 5)[[lego mindstorm](#), [Michael Gasperi et al.](#)].



Εικόνα 5: Αισθητήρας ήχου  
Πηγή:lego.com

## 2.2.6 Αισθητήρα υπερήχων

Η αρχή λειτουργίας των αισθητήρων υπερήχων βασίζεται στην εκπομπή υπερήχων υψηλής συχνότητας (πέρα από το όριο της ανθρώπινης ακοής), που ανακλώνται στα αντικείμενα, επιστρέφουν στον αισθητήρα και παίρνεται η μέτρηση. Ο χρόνος που χρειάζεται κάθε παλμός από την στιγμή της εκπομπής του να επιστρέψει πίσω στον αισθητήρα καθορίζει την απόσταση από αυτό. Όσο μεγαλύτερο το χρονικό διάστημα, τόσο μακρύτερα βρίσκεται το αντικείμενο και αντιστρόφως. Οι νυχτερίδες χρησιμοποιούν την ίδια αρχή ως μέσο για την πλοήγηση και τον εντοπισμό της λείας τους. Η τεχνική αυτή ονομάζεται ηχοεντοπισμός και επιτρέπει στις νυχτερίδες να διακρίνουν ένα έντομο από ένα φύλλο που πέφτει (Εικόνα 6) [[Dave Astolfo et al.](#)].



Εικόνα 6: Αισθητήρας υπερήχων  
(Πηγή:lego.com)

## 2.2.7 Αισθητήρα πυξίδας

Ο NXT αισθητήρας πυξίδας ανιχνεύει μαγνητικά πεδία για την πλοήγηση. Χρησιμοποιώντας μια ψηφιακή πυξίδα, ο αισθητήρας είναι σε θέση να μετρήσει το μαγνητικό πεδίο της Γης και να επιστρέφει ως ένδειξη την κατεύθυνση του οχήματος.

Η συσκευή μπορεί να χρησιμοποιηθεί χωρίς βαθμονόμηση ή μπορεί να βαθμονομείται με το λογισμικό για να εξισώνει τα μαγνητικά πεδία και να προσδιορίζει μια εξαιρετικά ακριβή ένδειξη πυξίδας (Εικόνα 7) [[Πανεπιστήμιο Κρήτης](#)].



Εικόνα 7: Ψηφιακή πυξίδα  
Πηγή:lego.com

## 2.2.8 Γυροσκόπιο

Ο συγκεκριμένος αισθητήρας επιστρέφει ως έξοδο τον αριθμό των μοιρών ανά δευτερόλεπτο περιστροφής καθώς επίσης δείχνει και την κατεύθυνση της. Μετρά +/- 360° ανά δευτερόλεπτο και μπορεί να χρησιμοποιηθεί σε ρομπότ που πρέπει να ισορροπήσουν, ταλαντευτούν ή σε άλλες λειτουργίες όπου η μέτρηση της περιστροφής είναι απαραίτητη [[Hitechnic](#)].

Εκτός από τις συσκευές που αναφέρθηκαν, υπάρχει πλήθος δομικών στοιχείων που μπορούν να χρησιμοποιηθούν για την επίτευξη διάφορων συστημάτων. Ενδεικτικά αναφέρω τα εξής:

- Τουβλάκια (Bricks)
- Πλακίδια (Plates)

- Δοκοί (Beams) καθώς και δοκοί με δόντα (Rack and Pinion).
- Άξονες (Axles)
- Δακτύλιοι (Bushings)
- Πιράκια (Pegs)
- Συνδέσμους (Connectors)
- Ρόδες (Wheels)
- Γρανάζια (Gears) καθώς και γρανάζια με λοξά δόντα (Bevel gears), Κοχλιωτά γρανάζια (worm gears), Γρανάζι με ολίσθηση (Slip clutch), διαφορικό κ.α.
- Ιμάντες και Τροχαλίες (Belts and Pulleys)
- Καλώδια

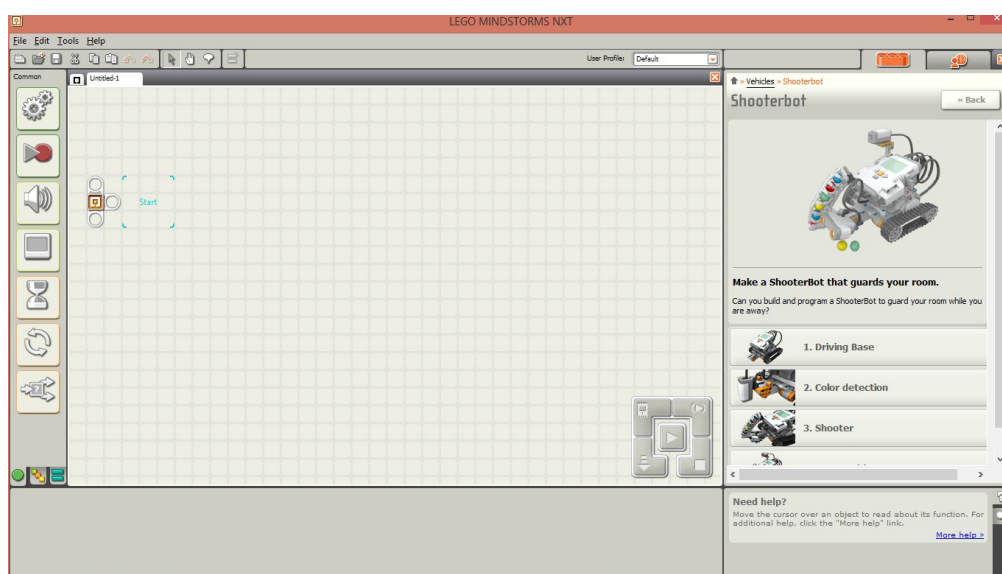
Όλα τα παραπάνω είναι σε διάφορα μεγέθη.(Εικόνα 8)



*Εικόνα 8: Mindstorm NXT kit. Πηγή: RoboRAVE*

## 2.3 Περιβάλλον προγραμματισμού του NXT

Για τον έλεγχο των κατασκευών που δημιουργούνται με τα δομικά στοιχεία των LEGO Mindstorms, μπορεί να χρησιμοποιηθεί το λογισμικό NXT-G. Το λογισμικό αυτό βασίζεται στο προγραμματιστικό περιβάλλον LabVIEW, το οποίο έχει αναπτυχθεί από τη National Instruments. Το λογισμικό διαθέτει μια Drag and drop φιλοσοφία προγραμματισμού η οποία βασίζεται στη φαντασία του χρήστη και όχι στην λογική γραμμών κώδικα που έχουν οι γλώσσες προγραμματισμού. Τα εικονίδια ενώνονται αυτόματα μόλις τα τοποθετήσουμε πάνω στη δοκό σύνδεσης (sequence beam). Το προγραμματιστικό περιβάλλον το καθιστά αρκετά εύκολο για ένα αρχάριο αλλά εξίσου ισχυρό για έναν έμπειρο χρήστη. ([NXT user guide](#)).



Εικόνα 9: Το γραφικό περιβάλλον του NXT-G. Αριστερά βρίσκεται η παλέτα προγραμματισμού ενώ στο μέσο η περιοχή εργασίας.

Η παλέτα προγραμματισμού βρίσκεται στην αριστερή πλευρά της επιφάνειας εργασίας και διαθέτει όλα τα μπλοκ εντολών που χρειάζονται για την δημιουργία προγραμμάτων. Οι καρτέλες στο κάτω μέρος της παλέτας επιτρέπουν την αλλαγή μεταξύ της κοινής παλέτας (που περιλαμβάνει τα πιο συχνά χρησιμοποιούμενα μπλοκ), της πλήρους παλέτας (που περιέχει όλα τα μπλοκ) και της προσαρμοσμένης παλέτας (που περιέχει τα μπλοκ της επιλογής του χρήστη για την ανάπτυξη εφαρμογών) ([NXT user guide](#)) (Εικόνα 9).

Το χειριστήριο το οποίο βρίσκεται στο δεξί κάτω μέρος της περιοχής εργασίας διαθέτει πέντε κουμπιά και επιτρέπει την μεταμόρφωση των προγραμμάτων από τον υπολογιστή στο NXT. Επίσης από το χειριστήριο μπορούν να αλλαχθούν οι ρυθμίσεις του NXT ([NXT user guide](#)).

Η διαδικασία για την ανάπτυξη, μεταφορά και χρήσης του προγράμματος στο NXT είναι αρκετά απλή και περιγράφεται συνοπτικά στα παρακάτω βήματα:

1. Μεταφορά των μπλοκ από την παλέτα στην επιφάνεια εργασίας
2. Σύνδεση σε σειρά

3. Ρύθμιση των παραμέτρων του κάθε μπλοκ εντολής
4. Σύνδεση του NXT με τον υπολογιστή μέσω καλωδίου usb ή με την χρήση Bluetooth
5. Λήψη του προγράμματος στο NXT
6. Από το περιβάλλον χρήσης του NXT μπορεί να τεθεί το πρόγραμμα σε εφαρμογή.

Υπάρχουν τρεις διαφορετικές καρτέλες στην παλέτα εντολών:

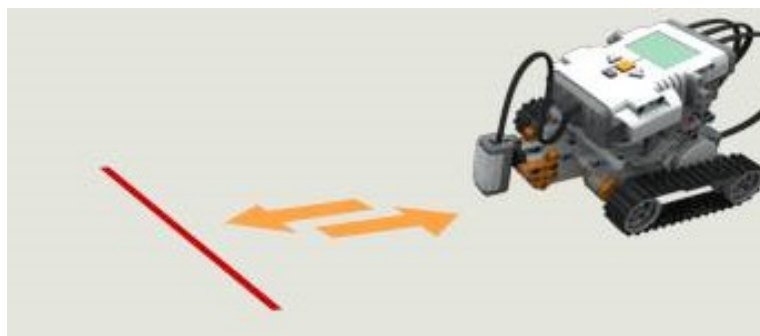
Η **βασική παλέτα** εντολών που όπως υποδηλώνει και το όνομα της περιέχει τις βασικές εντολές, αυτές δηλαδή που μας επιτρέπουν να δημιουργήσουμε το πρόγραμμα που θέλουμε για την πλειοψηφία των προβλημάτων που επιθυμούμε να επιλύσουμε. Στην παλέτα αυτή συναντούμε τις εντολές για να κινήσουμε τη ρομποτική μας κατασκευή, να βάλουμε το τούβλο NXT να αναπαράγει κάποιο ήχο ή μελωδία, να μας εμφανίσει κάποιο κείμενο ή γράφημα στην οθόνη του αλλά και τις πιο σύνθετες δομές προγραμματισμού όπως η αναμονή μέχρι να συμβεί κάποιο γεγονός, ή επανάληψη μιας διαδικασίας και η λήψη απόφασης από το ρομπότ μας ανάλογα με τις επικρατούσες συνθήκες.

Η **πλήρης παλέτα** εντολών από την οποία μπορούμε να χρησιμοποιήσουμε το σύνολο των δυνατοτήτων που υποστηρίζει ο μικροεπεξεργαστής NXT για την απόδοση της κατάλληλης συμπεριφοράς στη ρομποτική κατασκευή μας.

Η **προσαρμοσμένη παλέτα** εντολών στην οποία μπορούμε να βάλουμε κάποιες δικές μας εντολές. Τις εντολές αυτές μπορούμε να της δημιουργήσουμε συνθέτοντας μια διαδικασία από ένα συνδυασμό εντολών από αυτές που διαθέτει το προγραμματιστικό μας περιβάλλον. Με αυτό τον τρόπο δημιουργούμε μια δική μας βιβλιοθήκη με ένα αριθμό από διαθέσιμες υπορουτίνες.

## 2.4 Παράδειγμα ανάπτυξης, μεταφοράς και εφαρμογής προγράμματος

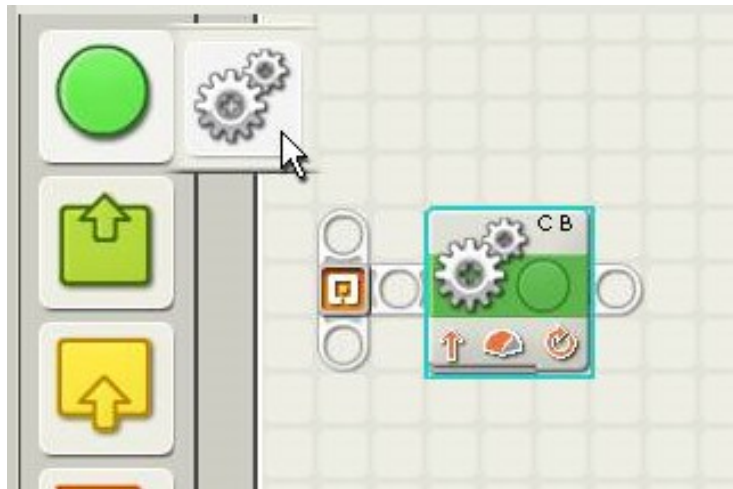
Στο παρακάτω παράδειγμα το οποίο είναι ένα από τα έτοιμα παραδείγματα που περιλαμβάνονται στο λειτουργικό NXT-G, θα προγραμματίσουμε το ρομπότ ώστε να μετακινηθεί μπροστά και πίσω στην αρχική του θέση (Εικόνα 10).



Εικόνα 10: Κίνηση οχήματος εμπρός και επιστροφή στην αρχική του θέση.

## Βήμα 1

Επιλέγουμε με τον κέρσορα την εντολή κίνησης και την σέρνουμε στην επιφάνεια εργασίας (Εικόνα 11).

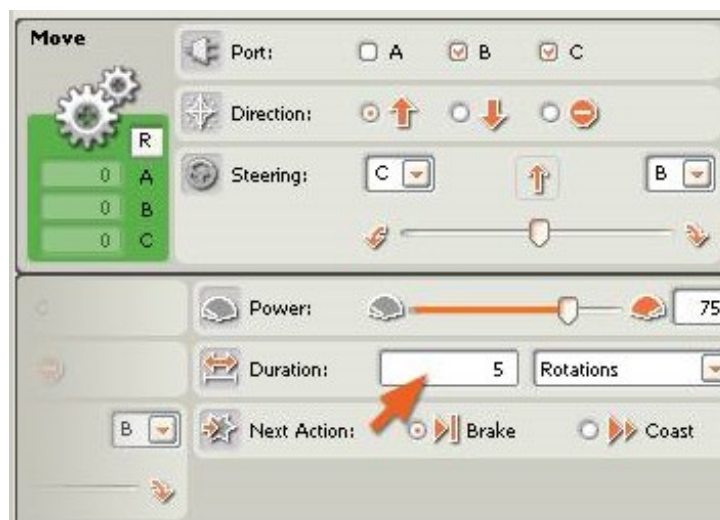


Εικόνα 11:Επιλογή κίνησης της παλέτας NXT-G.

## Βήμα 2

Ρυθμίζουμε τις παραμέτρους (Εικόνα 12).

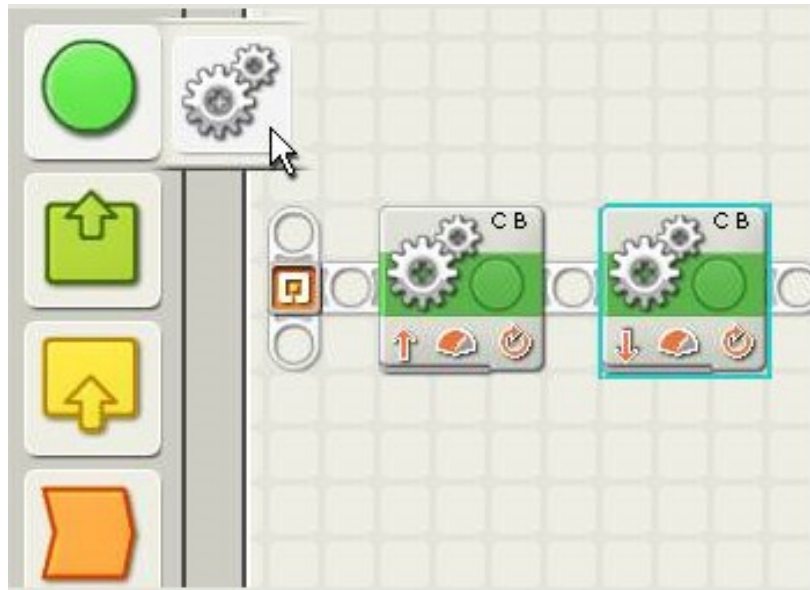
- Port, έξοδος brick προς κινητήρα, B και C
- Direction, διεύθυνση: μπροστά
- Steering, διεύθυνση τιμονιού: ευθεία
- Power, δύναμη κινητήρα: 75/100
- Duration, διάρκεια: 5 πλήρεις περιστροφές του κινητήρα (1800 μοίρες)
- Next action, επόμενη ενέργεια: φρένο



Εικόνα 12: Ρύθμιση παραμέτρων κίνησης.

### Βήμα 3

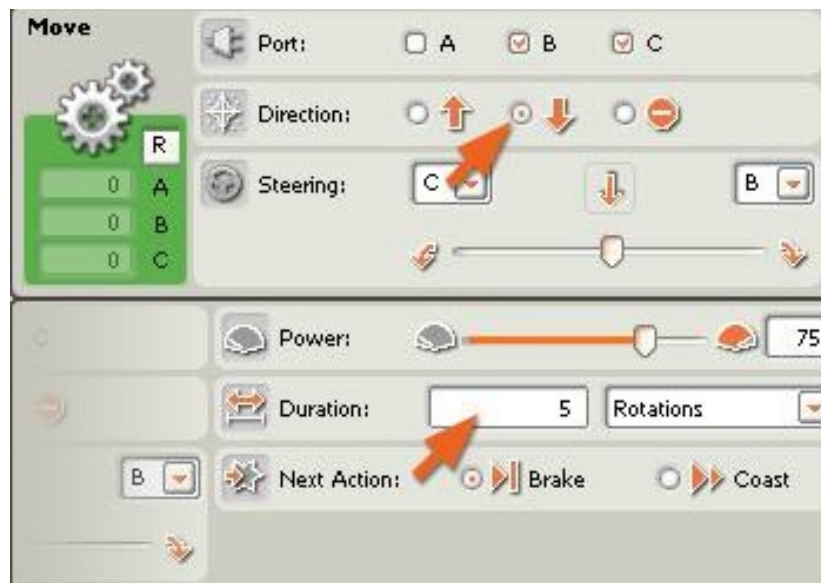
Επιλέγουμε με τον κέρσορα πάλι την εντολή κίνησης και την τοποθετούμε αμέσως δεξιά από την πρώτη (Εικόνα 13).



Εικόνα 13: Δύο εντολές κίνησης σε σειρά.

### Βήμα 4

Ρυθμίζουμε τις παραμέτρους για την δεύτερη κίνηση όπως και πριν όμως στο Direction επιλέγουμε: πίσω (Εικόνα 14).



Εικόνα 14: Ρύθμιση παραμέτρων κίνησης δεύτερης εντολής.

## Βήμα 5

Εισάγουμε το πρόγραμμα στο NXT (εικόνα 15)



*Εικόνα 15: Μέθοδος εισαγωγής του προγράμματος στο NXT-BRICK.*

## Βήμα 6

Από την οθόνη χρήσης του NXT-Brick ενεργοποιείτε το πρόγραμμα που μεταφέρθηκε (Εικόνα 16)



*Εικόνα 16:Οθόνη του NXT-BRICK.*

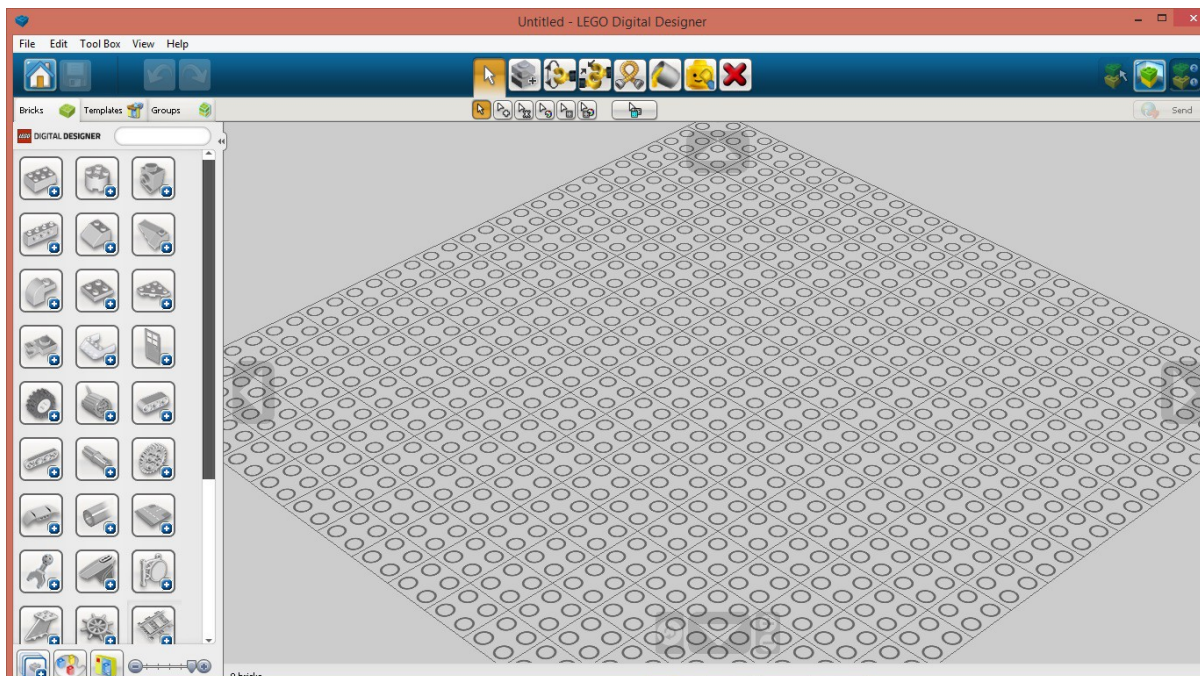
*(Πηγή:NXT user guide)*



## LEGO DIGITAL DESIGNER

Το Lego Digital Designer (LDD) είναι ένα πρόγραμμα από την Lego, που επιτρέπει στους χρήστες να δημιουργήσουν μοντέλα χρησιμοποιώντας εικονικά κομμάτια Lego. Πρόκειται για ένα σχεδιαστικό πρόγραμμα που επιτρέπει τον σχεδιασμό διάφορων συσκευών.

Το πρόγραμμα διαθέτει μια παλέτα με τούβλα και κομμάτια σε διάφορα χρώματα, όπου μπορούν να χρησιμοποιηθούν για να φτιαχτεί οποιοδήποτε μοντέλο θέλουμε. Υπάρχουν τρεις επιλογές προβολής. Στην πρώτη μπορούμε να φτιάξουμε το μοντέλο μας. Στη δεύτερη μπορούν να ληφθούν στιγμιότυπα (screenshots) του μοντέλου μας από διάφορες γωνίες θέασης και στη τρίτη επιλογή αφού έχουμε ολοκληρώσει το μοντέλο μας το πρόγραμμα παράγει αυτόματα οδηγίες για το πώς θα το φτιάξουμε, βήμα βήμα με τα πραγματικά κομμάτια Lego (Εικόνα 17).

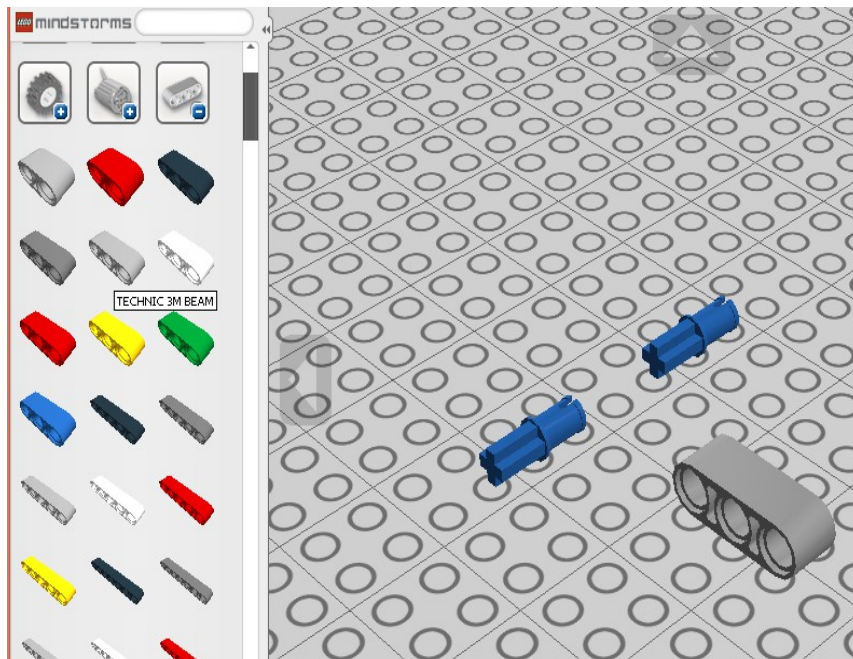


Εικόνα 17: Παλέτα του Lego Digital Designer (LDD).

Στο παρακάτω παράδειγμα θα συνδέσουμε δύο πιράκια με μία δοκό.

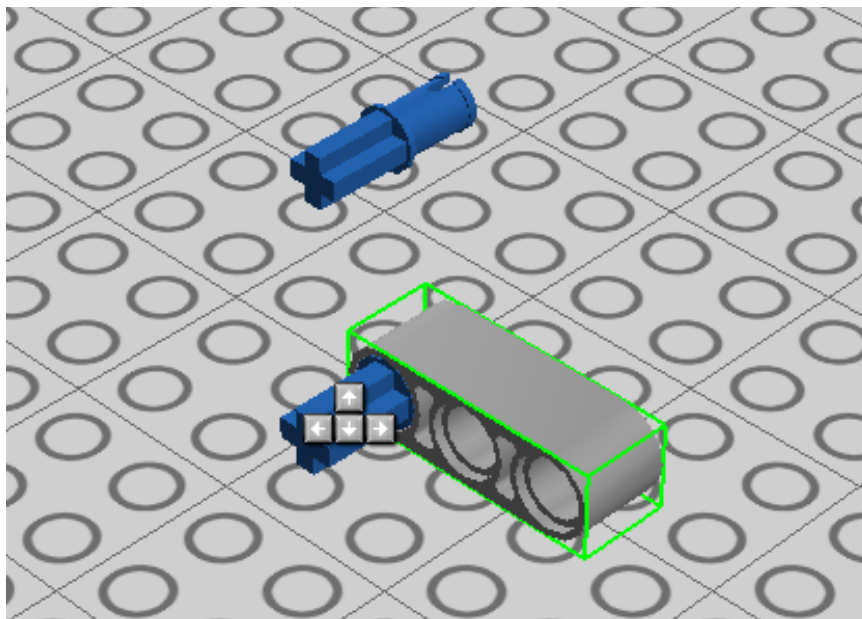
### Βήμα 1

Επιλέγουμε τα εξαρτήματα ένα ένα. Το πρόγραμμα τα τοποθετεί αυτόματα στην επιφάνεια εργασίας (Εικόνα 18).



Εικόνα 18: Επιλογή εξαρτημάτων και τοποθέτηση τους στο παράθυρο δημιουργίας.

## Βήμα 2

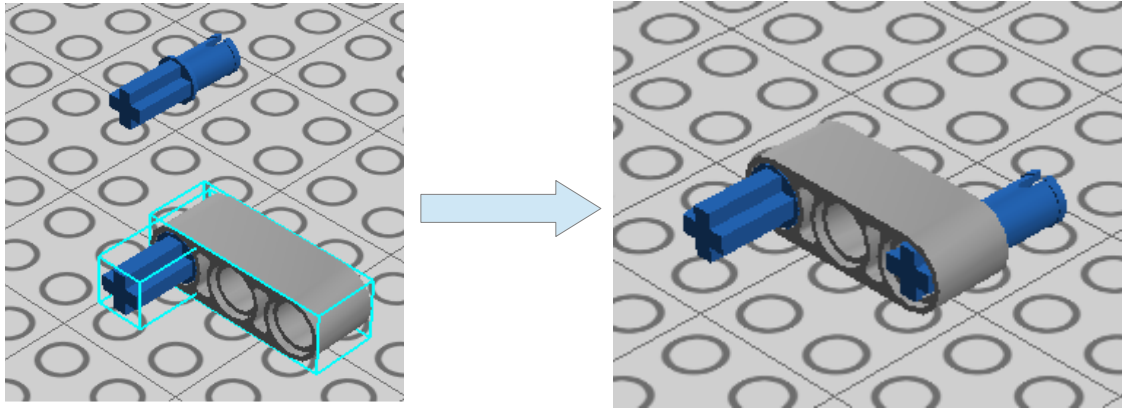


Εικόνα 19: Κατασκευή αντικειμένου.

Επιλέγουμε το πιράκι και σέρνουμε μέχρι να εφαρμόσει στη δοκό, στην οπή που επιθυμούμε (Εικόνα 19).

## Βήμα 3

Επιλέγουμε τη δοκό που είναι ενωμένη με το πιράκι έχοντας πατημένο το control ώστε να θεωρηθεί από το πρόγραμμα ως ένα σώμα και έπειτα σέρνουμε στο δεύτερο πιράκι (Εικόνα 19).



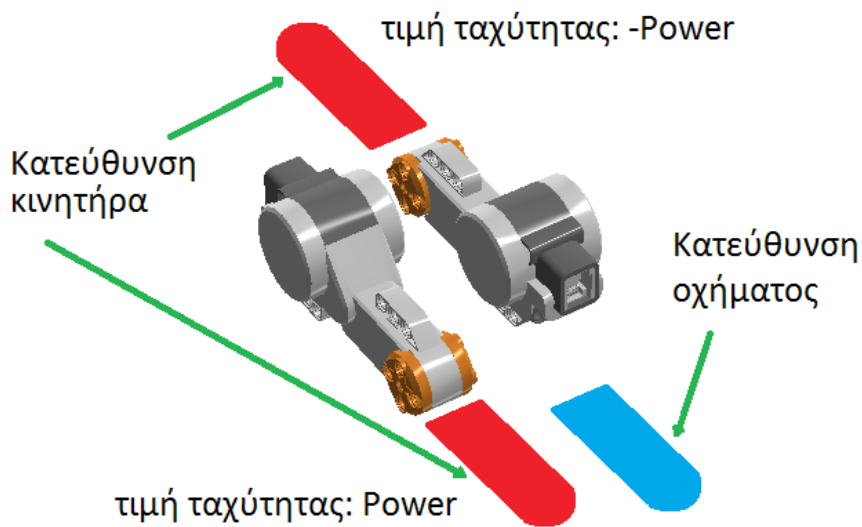
Εικόνα 20: Κατασκευή αντικειμένου.

#### Βήμα 4

Επιλέγουμε τον οδηγό σχεδίασης και έπειτα το πρόγραμμα μας καθοδηγεί πώς να δημιουργήσουμε το πραγματικό μας μοντέλο.

### 2.5 Σχεδιασμός και ανάπτυξη πειραματικού οχήματος

Το σύστημα που θέλουμε να δημιουργήσουμε θα πρέπει να κινείται στο χώρο να αντιλαμβάνεται τα εμπόδια που βρίσκονται στη πορεία του να τα αποφεύγει και να χαρτογραφεί την πορεία του.



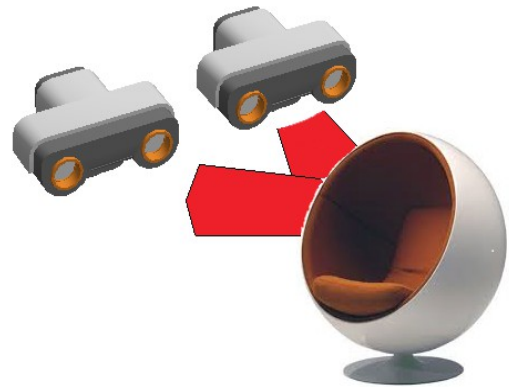
Εικόνα 21: Δύο κινητήρες περιστροφής τοποθετημένοι παράλληλα και σε αντίθετη φορά.

Στην αρχική υλοποίηση (όπου παρουσιάζεται παρακάτω) το όχημα αποτελούταν από δύο κινητήρες τοποθετημένους παράλληλα και σε αντίθετη φορά όπου για να κινηθεί το όχημα σε ευθεία γραμμή η τιμή 'Power' στην εντολή κίνησης θα πρέπει να είναι ίση κατά απόλυτη τιμή και αντίθετου πρόσημου (Εικόνα 21).

Η τοποθέτηση αυτή όμως των κινητήρων είχε σαν αποτέλεσμα να μην υπάρχει καλή κατανομή του φορτίου και να παρουσιαστεί πρόβλημα στην οδομέτρηση, οπότε προτιμήθηκε η παράλληλη τοποθέτηση προς στην ίδια φορά.

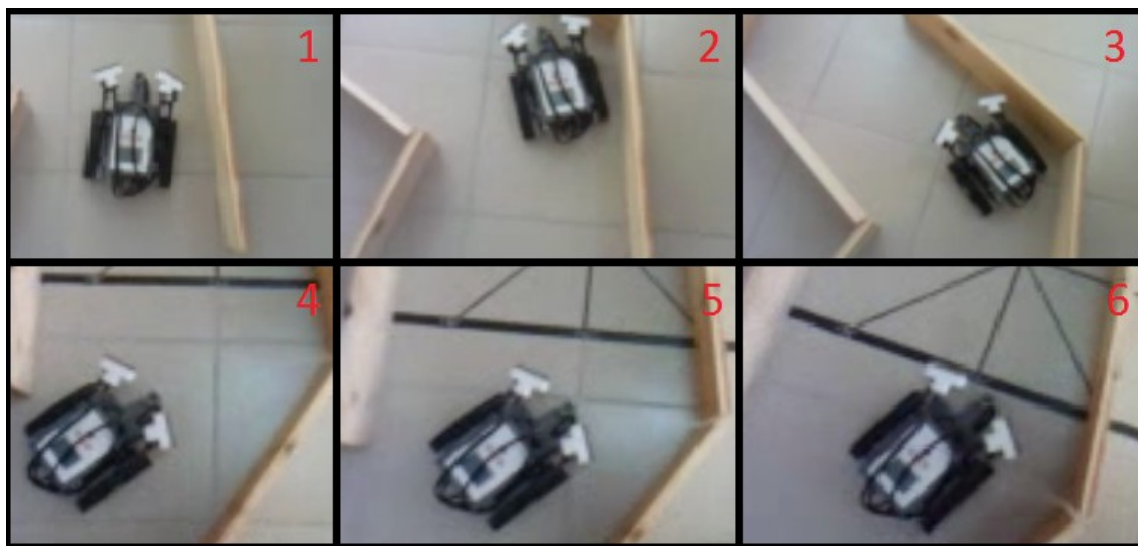
Άλλο πρόβλημα που παρουσιάστηκε ήταν στη τοποθέτηση των αισθητήρων υπερήχων.

Αρχικά οι αισθητήρες υπερήχων τοποθετήθηκαν και αυτοί παράλληλα από την δεξιά και αριστερή πλευρά του οχήματος, όμως μη μπορώντας να ρυθμίσουμε διαφορετική συχνότητα λειτουργίας στο καθένα είχε σαν αποτέλεσμα η ανάκλαση της ηχούς του ενός αισθητήρα να λαμβάνεται στο δεύτερο με αποτέλεσμα να υπάρχει εσφαλμένη μέτρηση (Εικόνα 22).



Εικόνα 22: Αισθητήρες υπερήχων.

Το πρόβλημα ξεπεράστηκε εν μέρη τοποθετώντας τους αισθητήρες υπό ελαφρά γωνία προς τα αριστερά και δεξιά αντίστοιχα. Στη καινούργια μορφή του το όχημα πλέον μπορούσε να περάσει μέσα από ένα διάδρομο αρκετά ευέλικτα, χωρίς να υπάρχει επιπλοκή με τις ηχούς των αισθητήρων (Εικόνα 23).



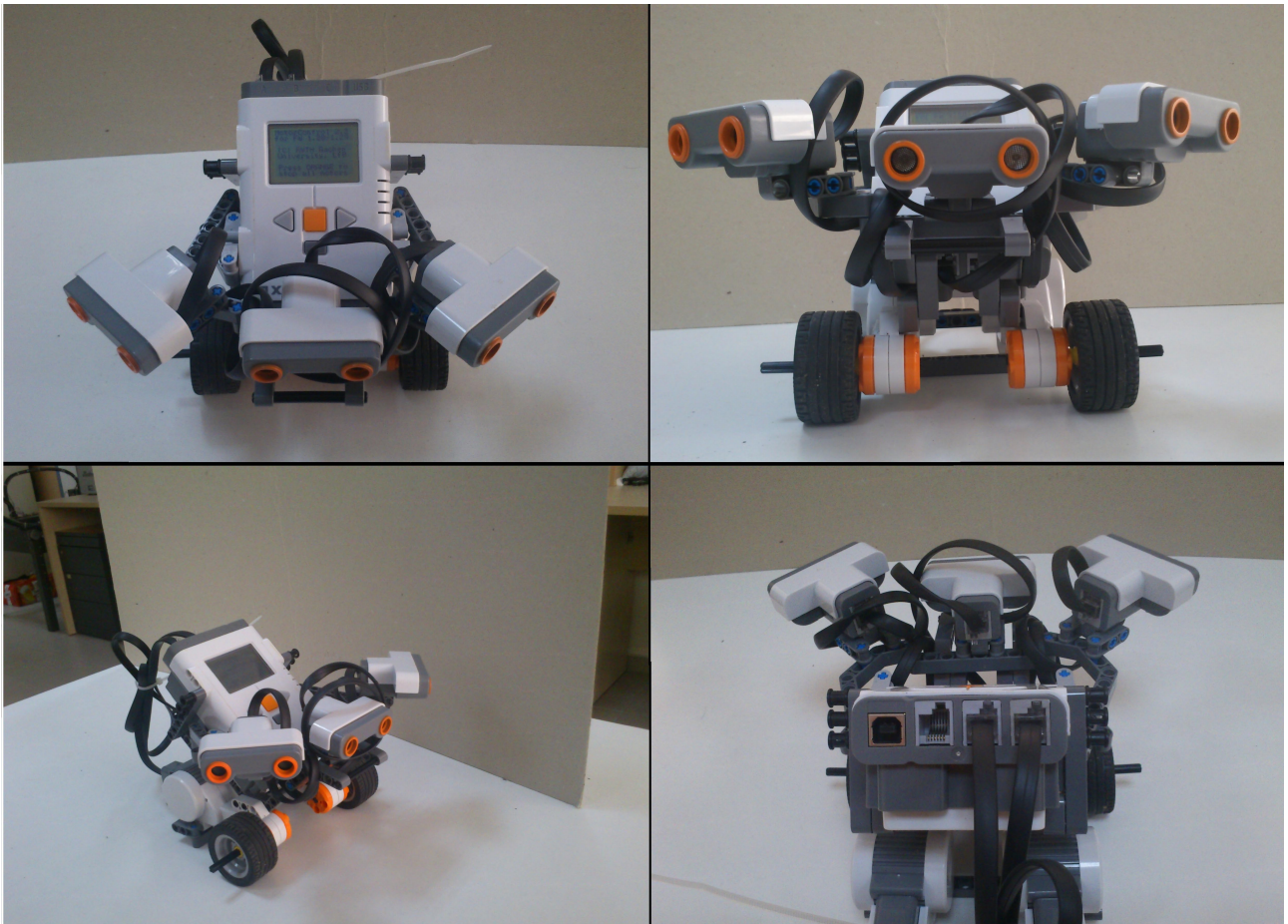
Εικόνα 23: Αυτοκινούμενο όχημα σε διάδρομο.

Αυτή η σχεδίαση λειτουργεί ικανοποιητικά για την αποφυγή της σύγκρουσης σε ένα διάδρομο αλλά με την προϋπόθεση πως δεν θα υπάρχουν άλλα εμπόδια διάσπαρτα στη διαδρομή του. Για λόγο της συγκεκριμένης τοποθέτησης των αισθητήρων το όχημα αδυνατούσε αρκετές φορές να αποφύγει τη σύγκρουση με αντικείμενα στο μέγεθος ενός ποτηριού καθώς δεν γινόταν αντιληπτό από τους αισθητήρες (Εικόνα24).

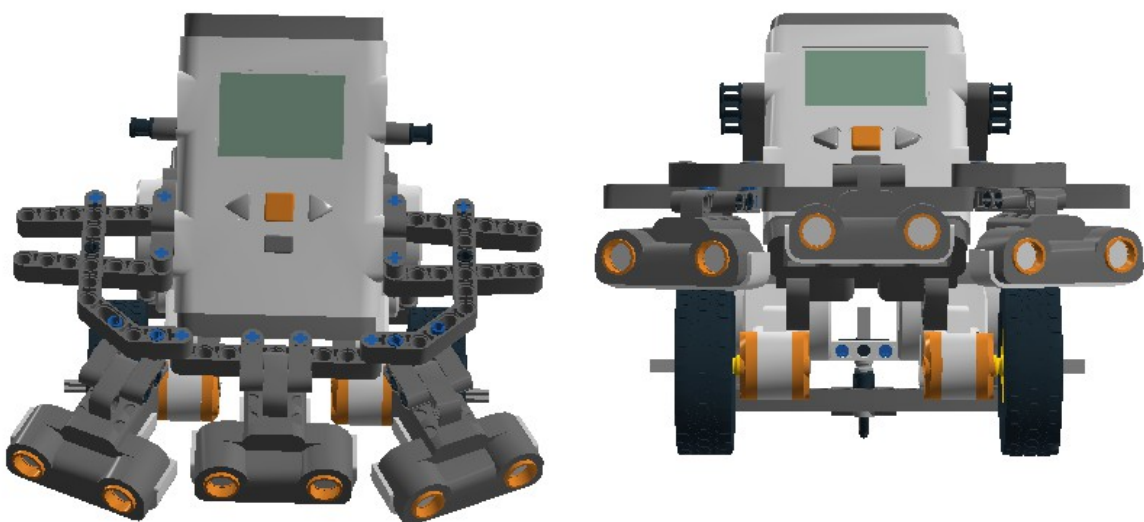


Εικόνα 24: Tuφλό σημείο.

Τελικά το σχέδιο κατέληξε στην ύπαρξη τριών αισθητήρων ώστε να μην υπάρχει τυφλό σημείο σε περίπου 180 μοίρες μπροστά από το όχημα. Το τελικό σχέδιο περιλαμβάνει 107 κομμάτια lego τρεις αισθητήρες υπερήχων (δύο στις μπροστά γωνίες του οχήματος με κατεύθυνση ελαφρώς δεξιά και αριστερά αντίστοιχα καθώς και ένα τρίτο αισθητήρα ακριβώς μπροστά από το από το όχημα), δύο μοτέρ για την κίνηση του οχήματος παράλληλα τοποθετημένα και με την ίδια φορά (Εικόνα 25).



Εικόνα 25: Τελική υλοποίηση οχήματος



Εικόνα 26: Τελική υλοποίηση στο πρόγραμμα *Lego digital designer*.

## 3 Η εργαλειοθήκη RWTH- Mindstorms NXT του MATLAB.

### 3.1 Γενικά για την εργαλειοθήκη

Αυτή η εργαλειοθήκη έχει αναπτυχθεί για τον έλεγχο του LEGO MINDSTORMS NXT με το MATLAB μέσω μιας ασύρματης Bluetooth σύνδεσης ή μέσω USB.

Το λογισμικό αυτό είναι ένα δωρεάν, ανοικτού κώδικα προϊόν. Η ανάπτυξη της εργαλειοθήκης αυτής έγινε για τις ανάγκες του προπτυχιακού project -MATLAB meets LEGO Mindstorms-στο Πολυτεχνείο του Άαχεν για φοιτητές της σχολής ηλεκτρολόγων μηχανικών, και ως εκ τούτου έχει σχεδιαστεί κυρίως για εκπαιδευτικούς σκοπούς ([RWTH Mindstorms NXT Toolbox](#)).

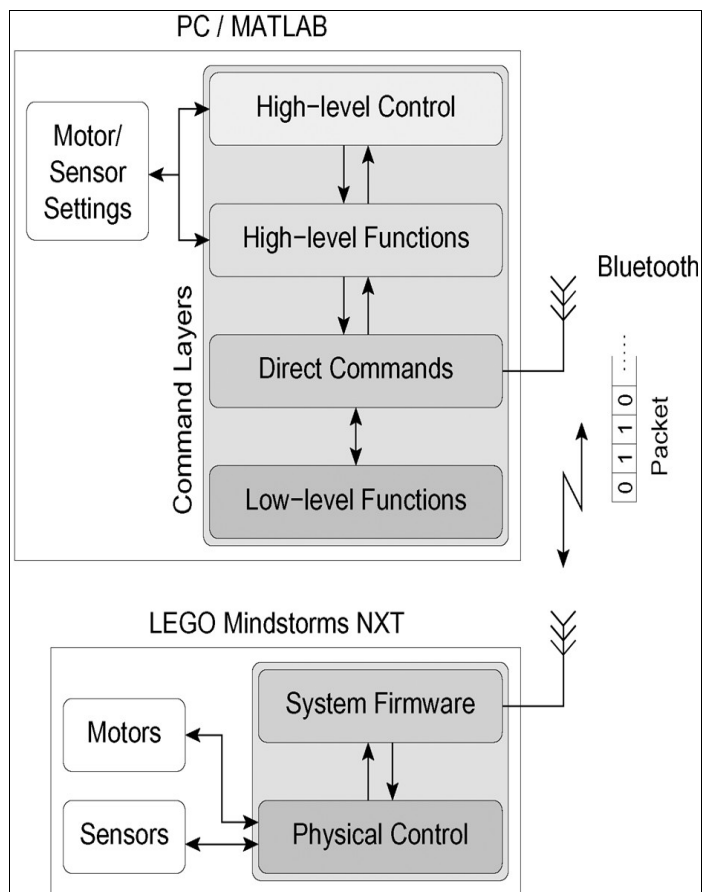
Αυτή η εργαλειοθήκη δεν είναι η μοναδική που υπάρχει στο Matlab για τα Lego mindstorms, όμως επειδή έχει σχεδιαστεί για πρωτοετείς φοιτητές δεν χρειάζεται να έχει κανείς ανεπτυγμένες γνώσεις θεωρίας ελέγχου ενώ ταυτόχρονα έχει μεγάλη λειτουργικότητα και πληθώρα επιλογών. Επιπλέον η εργαλειοθήκη παρέχει υψηλού επιπέδου εντολές και συναρτήσεις ελέγχου ώστε να υπάρχει ένα εύκολο περιβάλλον εργασίας, το οποίο είναι πολύ σημαντικό για ένα νέο χρήστη. Η εργαλειοθήκη περιλαμβάνει ακόμα λεπτομερείς καταλόγους με επεξηγήσεις, οδηγίες και παραδείγματα([Alexander Behrens et al.](#)).

Οι συναρτήσεις της εργαλειοθήκης είναι βασισμένες πάνω στο LEGO MINDSTORMS NXT Bluetooth πρωτόκολλο επικοινωνίας. Παρόλο που μια Bluetooth σύνδεση δεν είναι προτεινόμενη για έλεγχο ρομπότ σε πραγματικό χρόνο λόγω της υψηλής καθυστέρησης μετάδοσης της πληροφορίας, αυτή η εργαλειοθήκη παρέχει συναρτήσεις για αλληλεπίδραση με το ρομπότ απευθείας ([Alexander Behrens et al.](#)).

Το κύριο πλεονέκτημα της συγκεκριμένης υλοποίησης τηλεχειρισμού είναι ότι επιτρέπει να συνδυάσετε τις εφαρμογές ρομπότ με σύνθετες μαθηματικές πράξεις και απεικονίσεις σε περιβάλλον MATLAB ([Alexander Behrens et al.](#)).

Παρά το γεγονός ότι υπάρχει μια ευρεία επιλογή δωρεάν και μη λογισμικών για διαφορετικές γλώσσες προγραμματισμού, μόνο δύο εφαρμογές χρησιμοποιούν το προγραμματιστικό περιβάλλον ανάπτυξης MATLAB. Το πρώτο πακέτο λογισμικού που αναπτύχθηκε από τον T. Chikamasa, βασίζεται σε βρόχους ελέγχου σε περιβάλλον MATLAB και Simulink. Παρέχει μια λειτουργία προσομοίωσης και ένα ενσωματωμένο κωδικοποιητή, όπου δεν επιτρέπει η διόρθωση του κώδικα να γίνει κατά στάδια. Επίσης εστιάζει σε προηγμένη θεωρία ελέγχου και απαιτεί μια υπάρχουσα εξοικείωση με το Simulink, το οποίο δύσκολα μπορεί να αναμένεται από πρωτοετείς φοιτητές. Το δεύτερο πακέτο λογισμικού έχει αναπτυχθεί από τον Gregory Gutt και χρησιμοποιεί την Bluetooth επικοινωνία μεταξύ ενός ρομπότ NXT και τον υπολογιστή ως τηλεχειριστήριο. Χρειάζεται ωστόσο πρόσθετο λογισμικό για να επιτύχει την σύνδεση Bluetooth μέσω MATLAB. Δεδομένου ότι τα ληφθέντα αρχεία αποθηκεύονται αρχικά στον ενταμιευτή του NXT (Buffered in files) καθιστά την εφαρμογή αυτή μη κατάλληλη για άμεση και διαισθητική επικοινωνία υπολογιστή-ρομπότ ([Alexander Behrens et al.](#)).

Ως αποτέλεσμα των παραπάνω αναπτύχθηκε η καινοτόμα RWTH-Mindstorms NXT εργαλειοθήκη για το MATLAB, η οποία παρέχει απευθείας επικοινωνία μεταξύ των ρομπότ NXT και του MATLAB χωρίς τη χρήση πρόσθετου λογισμικού. Για αυτό το σκοπό εφαρμόστηκε στο MATLAB το Mindstorm Bluetooth πρωτόκολλο επικοινωνίας, το οποίο επιτρέπει στους χρήστες να ελέγχουν τους κινητήρες του NXT, να παίρνουν μετρήσεις από τους αισθητήρες του (αφής, ήχου, φωτός και υπερήχων), ή να ενεργοποιούν άλλες λειτουργίες του συστήματος όπως τον έλεγχο της στάθμης φόρτισης της μπαταρίας ή να ενεργοποιήσουν έναν ήχο. Αυτές οι απευθείας εντολές που καθορίζονται από το πρωτόκολλο, αντιπροσωπεύουν ένα στρώμα εντολών της δομής λειτουργιών της εργαλειοθήκης. Μια γενική επισκόπηση της επικοινωνίας μεταξύ MATLAB και του NXT χρησιμοποιώντας την πολυεπίπεδη δομή της εργαλειοθήκης, η οποία παρέχει χαμηλού καθώς και υψηλού επιπέδου λειτουργίες, φαίνεται στην εικόνα 27. Η δομή λειτουργιών της εργαλειοθήκης παρέχει τέσσερα επίπεδα εντολών. Ο χρήστης του MATLAB μπορεί να εφαρμόσει ρυθμίσεις στους κινητήρες και στους αισθητήρες καθώς και ολόκληρους βρόγχους ελέγχου μέσω λειτουργιών υψηλού επιπέδου. Αυτές οι διαμορφώσεις στη συνέχεια διέρχονται στο επίπεδο των απευθείας εντολών, όπου δημιουργεί ένα έγκυρο δυαδικό πακέτο Bluetooth χρησιμοποιώντας λειτουργίες χαμηλού επιπέδου. Στο επόμενο βήμα, το πακέτο μεταδίδεται μέσω του ασύρματου καναλιού Bluetooth στο NXT. Το firmware του NXT αποκωδικοποιεί το μήνυμα και ενεργοποιεί το φυσικό έλεγχο των κινητήρων και των αισθητήρων. Η αντίστροφη σειρά γίνεται για την μεταφορά πληροφοριών από το υλικό προς το MATLAB ([Alexander Behrens et al.](#)).



Εικόνα 27: Γενική επισκόπηση της επικοινωνίας μεταξύ MATLAB και υλικού NXT.

### 3.2 Παραδείγματα χρήσης της εργαλειοθήκης RWTH-Mindstorms NXT

Η ευχρηστία του περιβάλλοντος ανάπτυξης είναι ουσιώδης για την επιτυχία του έργου ενός καινούργιου χρήστη. Για αυτό το λόγο οι χαμηλού επιπέδου συναρτήσεις και οι απευθείας εντολές έχουν ενσωματωθεί στις υψηλού επιπέδου συναρτήσεις και στις εντολές ελέγχου, και ένα περιβάλλον χρήστη στα πρότυπα των υπόλοιπων εργαλειοθηκών του MATLAB έχει

αναπτυχθεί. Παρακάτω θα δούμε μερικά παραδείγματα για το πως πραγματοποιείται η κίνηση ενός κινητήρα καθώς και λήψη μετρήσεων από τους αισθητήρες

Αρχικά θα πρέπει να δημιουργήσουμε μια σύνδεση μεταξύ NXT και MATLAB μέσω Bluetooth και έπειτα την δημιουργία ενός αντικειμένου.

```
% Προετοιμασία
COM_CloseNXT('all');
Close all;
Clear all;

% Σύνδεση στο NXT, μέσω USB ή Bluetooth
h = COM_OpenNXT('bluetooth.ini');
COM_SetDefaultNXT(h);

%Δημιουργία ενός Αντικειμένου (Object) για τον κινητήρα
Forward = NXTMotor();

% Θέτουμε παραμέτρους
Forward.Power = 100;
%Το ποσοστό δύναμης του κινητήρα να είναι 100%
Forward.Port = MOTOR_A

%θέτουμε ως θύρα εξόδου/εισόδου του κινητήρα την 'A'
Forward.TachoLimit = 360

%θέτουμε τις μοίρες περιστροφής του κινητήρα στις 360, μια πλήρη
%περιστροφή.
Forward.SendToNXT();
%Στέλνει το πακέτο εντολών στο NXT και ξεκινάει η κίνηση του
%κινητήρα.
```



Η παραπάνω σειρά εντολών/παραμετροποιήσεων μπορεί να γραφεί και ως εξής:

```
% Προετοιμασία
COM_CloseNXT('all');
Close all;
Clear all;

% Σύνδεση στο NXT, μέσω USB ή Bluetooth
h = COM_OpenNXT('bluetooth.ini');
COM_SetDefaultNXT(h);

Forward = NXTMotor('A','Power',100,'TackoLimit',360);
Forward = SendToNXT();
```

Αντίστοιχα για την ενεργοποίηση και τη λειτουργία του αισθητήρα υπερήχων χρησιμοποιείται η παρακάτω διαδικασία.

```
OpenUltrasonic(SENSOR_1)
% Ενεργοποιούμε τον αισθητήρα υπερήχων που βρίσκεται στην έξοδο 1
% του Brick, αυτή την στιγμή ο αισθητήρας ξεκινάει να στέλνει
% υπερήχους.

Pause(0.1);
% Δίνουμε χρόνο στον αισθητήρα να μετρήσει τις ηχούς.

GetUltrasonic(SENSOR_1)
% Πάρε μια μέτρηση.

CloseUltrasonic(SENSOR_1)
% Απενεργοποιούμε τον αισθητήρα μετά την χρήση.
```

Στο παράδειγμα που ακολουθεί το πρόγραμμα λειτουργεί ως εξής, αρχικά διαγράφουμε όλες τις αποθηκευμένες τιμές και μεταβλητές, έπειτα πραγματοποιούμε σύνδεση μεταξύ MATLAB και NXT και ενεργοποιούμε των αισθητήρα υπερήχων. Στη συνέχεια για δέκα επαναλήψεις το πρόγραμμα παίρνει μέτρηση από τον αισθητήρα και εάν δεν υπάρχει αντικείμενο σε απόσταση 20 εκατοστών τότε προχωράει ευθεία για 3 περιστροφές του κινητήρα, αλλιώς στρίβει 90 μοίρες.

```
COM_CloseNXT('all');  
  
close all;  
  
clear all;  
  
Connection=COM_OpenNXT('bluetooth.ini');  
COM_SetDefaultNXT(Connection);  
  
OpenUltrasonic(SENSOR_1);  
I=0  
  
while i<10;  
    distance = GetUltrasonic(SENSOR_1);  
    if distance>20  
        forward = NXTMotor('AB','Power',70,'TachoLimit',1080);  
        forward.SendToNXT();  
    else  
        steering = NXTMotor('A','Power',40,'TachoLimit',570);  
        steering.SendToNXT();  
    end  
  
    i=i+1;  
end  
CloseUltrasonic(SENSOR_1);  
end
```

## 4 Ασαφής Λογική

### 4.1 Γενικά περί ασαφούς Λογικής

Η εισαγωγή της ασαφούς Λογικής έγινε από τον Περσικής καταγωγής, Αμερικανό καθηγητή Lotfi Zadeh. Η σύλληψη της σχετικά απλής ιδέας, έγινε ένα βράδυ του Ιούλη του 1964 στη Νέα Υόρκη και δημοσιεύθηκε στη διάσημη πια εργασία του το 1965 ([Γ. Α. Θεοδώρου](#)).

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 ([Γ. Α. Θεοδώρου](#)).

Σύμφωνα με την κλασσική δίτιμη λογική μια λογική πρόταση μπορεί να πάρει μόνο δύο τιμές, δηλαδή μπορεί να είναι αληθής ή ψευδής (1 ή 0), αποκλείοντας τρίτη λύση. Έτσι αν μια λογική πρόταση δεν είναι αληθής τότε θα είναι αναγκαία ψευδής ([Γ. Α. Θεοδώρου](#)).

Η Ασαφής Λογική είναι μια επέκταση της κλασσικής-παραδοσιακής δίτιμης Αριστοτέλειας λογικής, που μας εφοδιάζει με έναν αποτελεσματικό μαθηματικό σκελετό, για την έκφραση γλωσσικών εννοιών και για την παράσταση γνώσεων και πληροφοριών σε περιβάλλον ασάφειας και αβεβαιότητας. 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 ([Γ. Α. Θεοδώρου](#)).

Η ασάφεια (fuzziness) είναι μια έννοια που σχετίζεται με την ποσοτικοποίηση της ποιοτικής πληροφορίας και οφείλεται κυρίως σε μη-ακριβή δεδομένα. Για παράδειγμα, η φράση “Ο Νίκος είναι ψηλός” αν και δεν προσδιορίζει με ακρίβεια το ύψος του Νίκου, επιτρέπει να βγουν κάποια συμπεράσματα ή να ληφθούν ορισμένες αποφάσεις για θέματα που σχετίζονται με το ύψος του Νίκου ([Βλαχάβας Ιωάννης et al.](#)).

Αν και είναι δυνατό να αποδοθούν συγκεκριμένες τιμές σε λεκτικά προσδιορισμένα μεγέθη για να περιοριστεί η ασάφεια, αυτό οδηγεί πολλές φορές σε λάθος κρίσεις. Αν για παράδειγμα, θεωρηθεί ότι ψηλός είναι όποιος έχει ύψος πάνω από 1.95 μέτρα, δεν είναι απόλυτα σωστό να βγει το συμπέρασμα ότι κάποιος με ύψος 1.94 δεν είναι ψηλός ([Βλαχάβας Ιωάννης et al.](#)).

### 4.2 Κλασσικό και Ασαφές σύνολο

Αρχικός σκοπός του γνωστού κλασσικού συνόλου  $A$  είναι να μοντελοποιήσει μαθηματικά, δηλαδή να αναπαραστήσει τυπολογικά, κάποια λογική έννοια.

πχ η έννοια “οι ακέραιοι αριθμοί που είναι μεταξύ 2 και 6 ” μπορεί να παρασταθεί με το σύνολο:  $A = \{x \in \mathbb{Z} | 2 \leq x \leq 6\} = \{3, 4, 5\}$ , όπου  $\mathbb{Z}$  συμβολίζει το σύνολο των ακεραίων αριθμών

Έτσι πρώτα χρειαζόμαστε ένα σύνολο αναφοράς που περιέχει όλα τα στοιχεία τα σχετικά με την έννοια που θέλουμε να εκπροσωπήσουμε μέσω του συνόλου  $A$ , (όπως στο ανωτέρω

παράδειγμα όπου το σύνολο αναφοράς είναι το  $\mathbb{Z}$ ) ([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 ([Γ. Α. Θεοδώρου](#)):

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

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

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

Από τα παραπάνω είναι σαφές πως για τα κλασικά σύνολα υπάρχουν δύο κατηγορίες ([Γ. Α. Θεοδώρου](#))

- Στην κατηγορία με τα στοιχεία του  $X$  που ανήκουν στο  $A$  (ζεστό)
- 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 ([Γ. Α. Θεοδώρου](#)).

Όμως αν επιχειρούσαμε να εκφράσουμε μαθηματικά μέσω του παραπάνω κλασικού συνόλου, μια απλή καθημερινή λεκτική έννοια της μορφής <<το νερό είναι περίπου ζεστό>>, θα ήταν αδύνατο, αφού μια τέτοια καθημερινή μεν αλλά αόριστη-προσεγγιστική έννοια δεν μπορεί να διαχωρίσει σαφώς το σύνολο αναφοράς των ακεραίων  $\mathbb{Z}$  σε δύο καθαρές κατηγορίες στοιχείων (σε αυτά που ανήκουν και σε αυτά που δεν ανήκουν στο  $\mathbb{Z}$ ), δεδομένου ότι το “περίπου ζεστό” δεν έχει καθαρά-σαφή όρια, όπως απαιτούν τα κλασικά μαθηματικά.

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 ([Γ. Α. Θεοδώρου](#))

Πιο συγκεκριμένα η Ασαφής Συνολοθεωρία εισάγει την συνάρτηση συμμετοχής ( $\mu_A(x)$ ) ενός ασαφούς συνόλου  $A$  ([Γ. Α. Θεοδώρου](#)).

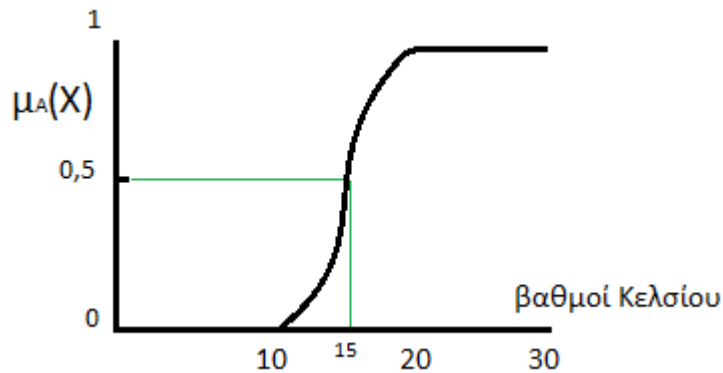
Ο αριθμός  $\mu_A(x) \in [0, 1]$  δηλώνει το βαθμό συμμετοχής με τον οποίο το στοιχείο  $x$  ανήκει στο  $X$  ανήκει (συμμετέχει) στο ασαφές υποσύνολο  $A$  του  $X$  ([Γ. Α. Θεοδώρου](#)).

$\mu_A(x) = 1$ , σημαίνει ότι το  $x$  ανήκει ολοκληρωτικά στο  $A$

$\mu_A(x) = 0$ , σημαίνει ότι το  $x$  δεν ανήκει καθόλου στο  $A$

$0 < \mu_A(x) < 1$ , σημαίνει ότι το  $x$  ανήκει μερικά, δηλαδή κατά κάποιον βαθμό στο  $A$

Στο διάγραμμα της εικόνας 28 απεικονίζεται ο βαθμός συμμετοχής που προσδιορίζει πότε το νερό είναι ζεστό.

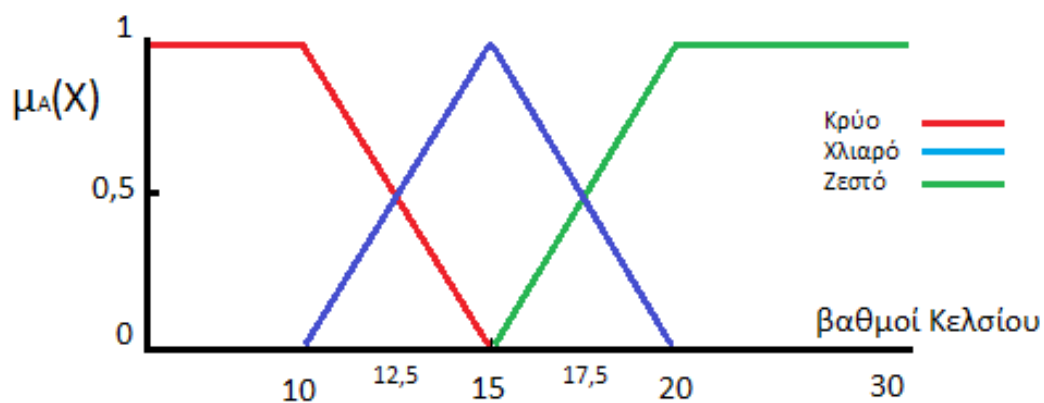


Εικόνα 28: Βαθμός συμμετοχής  $\mu_A(x)=0.5$

Έτσι οι βαθμοί Κελσίου που είναι

- από 20 και πάνω θεωρείται ζεστό με  $\mu_A(20) = 1$
- ακριβώς 15 θεωρείται ζεστό με  $\mu_A(15) = 0,5$
- από δέκα και κάτω θεωρείται ζεστό με  $\mu_A(10) = 0$

Στο διάγραμμα της εικόνας 29 απεικονίζεται η μεταβλητή θερμοκρασία με τα ασαφή σύνολα/τιμές “κρύο”, “χλιαρό”, “ζεστό” με υπεрсύνολο αναφοράς τους βαθμούς Κελσίου.



Εικόνα 29: Ασαφή σύνολα: Κρύο, Χλιαρό, Ζεστό

Σε αυτή την περίπτωση για τους βαθμούς Κελσίου 12,5 και 17,5 ισχύει:

- Κρύο:  $\mu_A(12,5) = 0,5$ ,  $\mu_A(17,5) = 0$
- Χλιαρό:  $\mu_A(12,5) = 0,5$ ,  $\mu_A(17,5) = 0,5$
- Ζεστό:  $\mu_A(12,5) = 0$ ,  $\mu_A(17,5) = 0$

Όπως μπορούμε να παρατηρήσουμε τα σύνολα κρύο με χλιαρό και χλιαρό με ζεστό αλληλοκαλύπτονται. Αυτό είναι βασικό χαρακτηριστικό της ασαφής λογικής και πάνω σε αυτό στηρίζεται η ασαφής συλλογιστική.

Με λίγα λόγια έχουμε μια γλωσσική μεταβλητή (θερμοκρασία) να περιγράφεται από τις τρεις γλωσσικές τιμές (κρύο, χλιαρό, ζεστό) σε ένα υπερσύνολο αναφοράς (βαθμοί Κελσίου)

Να σημειώσουμε πως ο βαθμός συμμετοχής δεν θα πρέπει να συγχέεται με τον όρο “πιθανότητα”.

Το  $P(A) = 0,7$  σημαίνει την πιθανότητα κατά 0,7 να ισχύσει, ενώ ο βαθμός συμμετοχής  $\mu_A(X) = 0,7$  σημαίνει πως ισχύει κατά 0,7.

### 4.3 Λεκτικοί τελεστές

Η λεκτική μεταβλητή Θερμοκρασία αφού την ορίσαμε με ασαφή σύνολα είναι πλέον μια ασαφή μεταβλητή. Οι τρεις λεκτικές τιμές που περιγράφονται από τα τρία ασαφή σύνολα μπορούν να αλληλεπιδράσουν μεταξύ τους με πράξεις με την χρήση λεκτικών τελεστών.

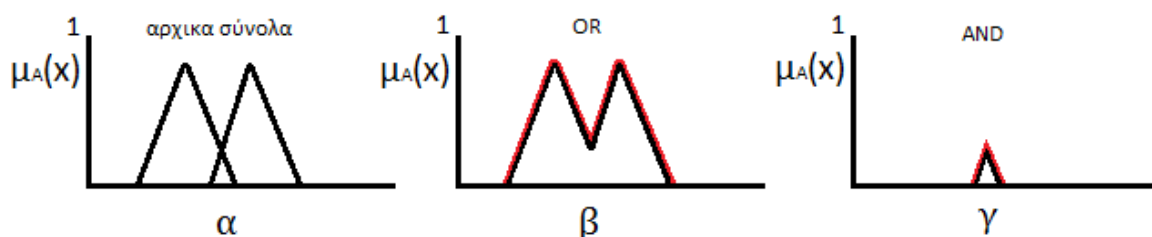
Οι βασικοί λεκτικοί τελεστές είναι οι εξής([Βλαχάβας Ιωάννης et al.](#)):

$$VERY A = (\mu_A(x))^2$$

$$A \text{ AND } B = \mu_A(x) \cap \mu_B(x), \text{ πράξη ελαχίστου (min)}$$

$$A \text{ OR } B = \mu_A(x) \cup \mu_B(x), \text{ πράξη μεγίστου (max)}$$

$$NOT A = [1 - \mu_A(x)]$$



Εικόνα 30: Τομή και ένωση ασαφών συνόλων.

Στα διαγράμματα της εικόνας 30 παρουσιάζονται ενδεικτικά τα αρχικά σύνολα (α) η ένωση (β) και η τομή (γ) δύο συνόλων

## 4.4 Ασαφή συστήματα

Ασαφές σύστημα είναι ένα σύνολο από <<IF-THEN>> κανόνες που αντιστοιχίζει λεκτικές-τιμές εισόδους σε λεκτικές-τιμές εξόδους. Στην πιο απλή του μορφή:

**IF** x is A **THEN** y is B

παράδειγμα:

***IF** Θερμοκρασία is κρύο **THEN** Boiler is Μέγιστο*

Ο παραπάνω ασαφής κανόνας αντιστοιχίζει την ασαφή είσοδο/τιμή -κρύο- της μεταβλητής -Θερμοκρασία- στην ασαφή έξοδο/τιμή -Μέγιστο- της μεταβλητής -Boiler-.

Ας υποθέσουμε ότι θέλουμε η θερμοκρασία του νερού να είναι “χλιαρή”, η λογική του ασαφούς ελεγκτή στο παράδειγμα μας θα είναι:

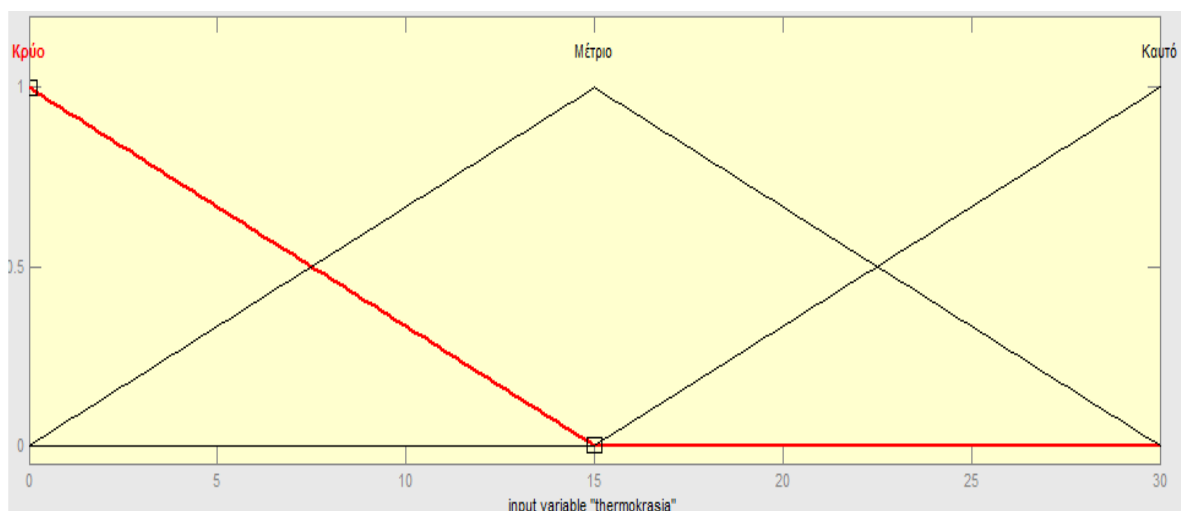
***IF** Θερμοκρασία is κρύο **THEN** Boiler is Μέγιστο*

***IF** Θερμοκρασία is μέτριο **THEN** Boiler is Μέτριο*

***IF** Θερμοκρασία is καυτό **THEN** Boiler is Κλειστό*

Έχουμε ορίσει στη λεκτική μεταβλητή “Θερμοκρασία” τις ακόλουθες λεκτικές-υποκειμενικές τιμές με την κάθε μια να ορίζεται από ένα σύνολο, στο υπερσύνολο αναφοράς “Βαθμοί Κελσίου” (Εικόνα 31):

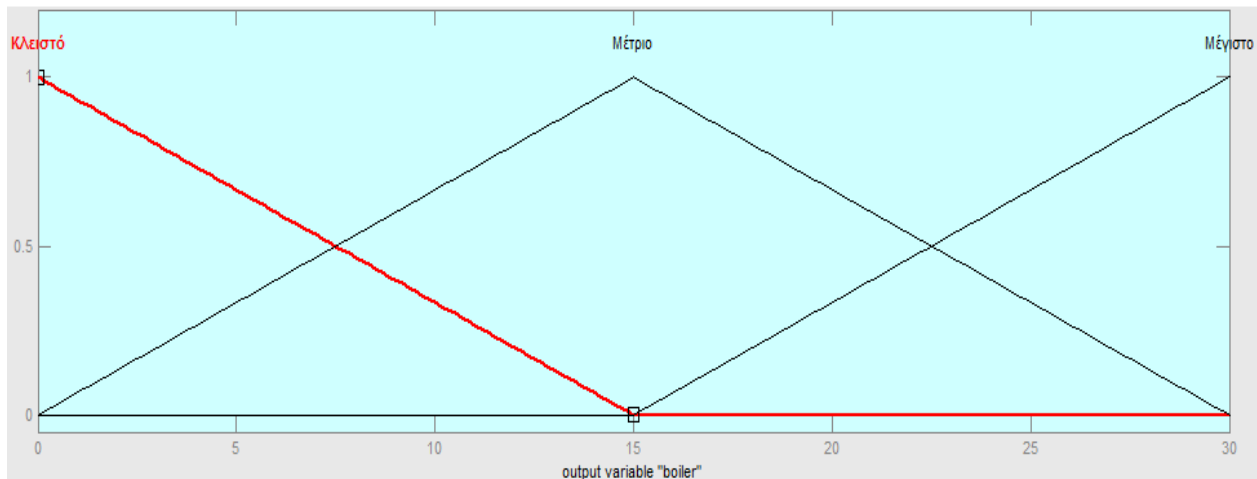
- Κρύο = (0,1,...,15)
- Μέτριο = (0,1,...,30)
- Καυτό = (15,16,...,30)



Εικόνα 31: Ασαφή σύνολα Ζεστό, Κρύο, Καυτό.

Και αντίστοιχα για την λεκτική μεταβλητή “Boiler” τις ακόλουθες τιμές (Εικόνα 32):

- Κλειστό = (0,1,...,15)
- Μέτριο = (0,1,...,30)
- Μέγιστο = (15,16,...,30)



Εικόνα 32: Ασαφή σύνολα Κλειστό, Μέτριο, Μέγιστο.

Στην περίπτωση που η είσοδος της θερμοκρασίας είναι 5 °C τότε ισχύει:

- Κρύο:  $\mu_A(5) = 0,67$
- Μέτριο:  $\mu_A(5) = 0,33$
- Καυτό  $\mu_A(5) = 0$

και οι κανόνες ενεργοποιούνται ως εξής (Εικόνα 33):

**IF Θερμοκρασία=5 is (0.67)κρύο THEN Boiler is (0.67)Μέγιστο**

**IF Θερμοκρασία=5 is (0.33)μέτριο THEN Boiler is (0.33)Μέτριο**

**IF Θερμοκρασία=5 is (0)καυτό THEN Boiler is (0)Κλειστό**

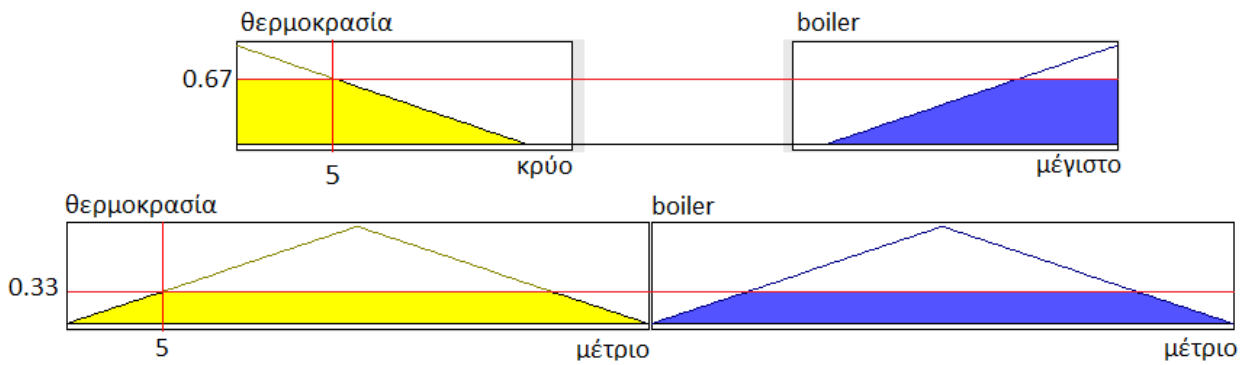
Για ευκολία θέτουμε  $B'_2 = (0.33)\text{Μέτριο}$  και  $B'_3 = (0.67)\text{Μέγιστο}$ .

Στη συνέχεια βρίσκουμε την συνολοθεωρητική ένωση-έξοδο-ασαφές σύνολο Boiler, δηλαδή το συνολοθεωρητικό άθροισμα:

$$\text{BOILER} = B'_2 \cup B'_3 = B'_2 + B'_3 \text{ (άθροισμα συνόλων)}$$

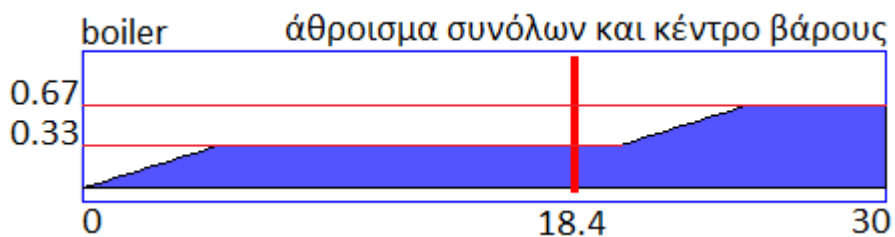
Η παραπάνω σχέση εκφράζεται και ως:  $\text{BOILER} = \max \{ B'_2, B'_3 \}$  .





Εικόνα 33: Οι είσοδοι και οι έξοδοι μετά την ενεργοποίηση των κανόνων.

Στην εικόνα 34 παρουσιάζεται το άθροισμα συνόλων και κέντρο βάρους.



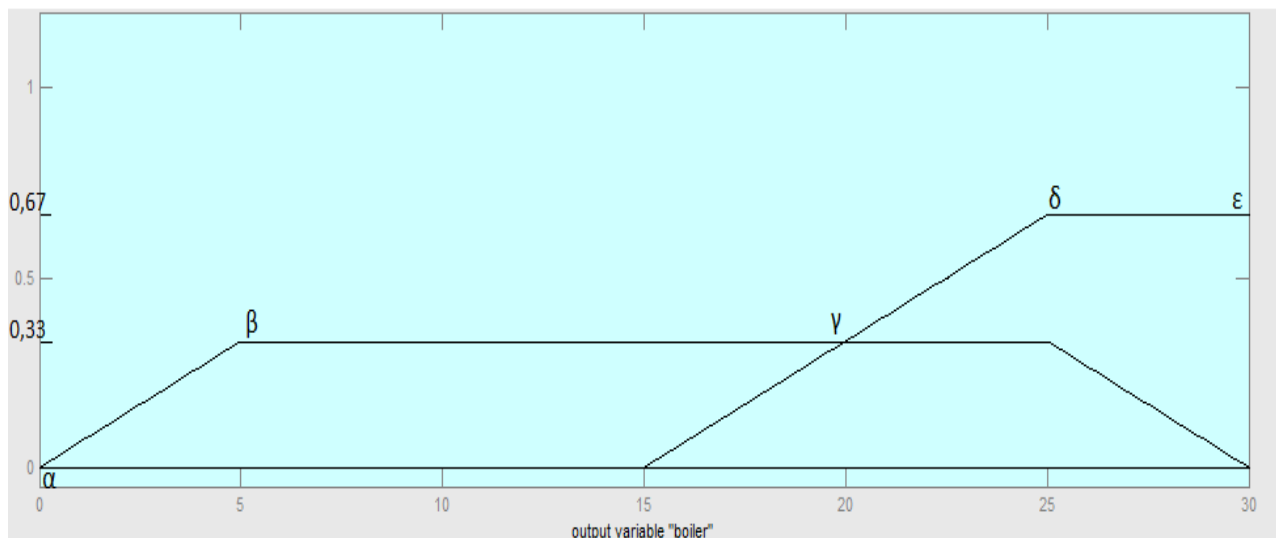
Εικόνα 34: Ένωση των συνόλων εξόδου Μέτριο και Μέγιστο.

Αν θέλουμε να γίνουμε λίγο πιο αναλυτικοί για το πως γίνεται η αποασαφοποίηση τότε στο προηγούμενο παράδειγμα η ένωση-έξοδος εκφράζεται γραφικά από την ένωση του εξωτερικού περιγράμματος (α,β,γ,δ,ε) (Εικόνα 35) των ασαφών  $B'_2, B'_3$  συνόλων

(Να σημειώσουμε πως στο παρακάτω παράδειγμα ακολουθείται η μεθοδολογία επίλυσης όπως παρουσιάζεται στο βιβλίο του [Γ. Α. Θεοδώρου](#))

Με σημεία:

$$\alpha = (0, 0), \beta = (5, 0.33), \gamma = (20, 0.33), \delta = (25.10, 0.67), \epsilon = (30, 0.67)$$



Εικόνα 35: Διαδικασία αποασαφοποίησης εξόδου.

Η παραπάνω ένωση έχει προφανώς σύνολο αναφοράς  $y \in [0,30] \cup [15,30] = [0,30]$ , προσδιορίζεται σύμφωνα με την ένωση του εξωτερικού διαγράμματος (αβγδε) των ασαφών συνόλων  $B'_2, B'_3$ , ως εξής

$$\max_{y \in [0,5]} \{B'_2(y), B'_3(y)\} = B'_2(y \in [0,5]) = 0.066y$$

Για  $y$  στο διάστημα από 0 μέχρι 5, (δηλαδή στο περίγραμμα αβ), έχουμε:

αφού στο διάστημα από 0 μέχρι 5 έχουμε μόνο την αριστερή πλευρά του τραπεζίου  $B'_2$  που είναι ένα ευθύγραμμο τμήμα που διέρχεται από τα σημεία (0, 0) και (5, 0.33) και άρα από την γνωστή εξίσωση:

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

της αναλυτικής γεωμετρίας για μια ευθεία που διέρχεται από δύο γνωστά σημεία  $(x_1, y_1)$ ,  $(x_2, y_2)$  αντικαθιστώντας βρίσκουμε  $\frac{y-0}{0.33-0} = \frac{x-0}{5-0} \dots y = 0.66 \cdot x$ , δηλαδή με τα εδώ σύμβολα είναι:  $B'_2(y \in [0,5]) = 0.33$

Όμοια για  $y$  στο διάστημα από 5 μέχρι 20 (δηλαδή στο περίγραμμα βγ), βρίσκουμε

$$\max_{y \in [5,20]} \{B'_2(y), B'_3(y)\} = B'_2(y \in [5,20]) = 0.33$$

και για  $y$  στο διάστημα από 20 μέχρι 25.10 (δηλαδή στο γδ) είναι:

$$\max_{y \in [20,25.10]} \{B'_2(y), B'_3(y)\} = B'_3(y \in [20,25.10]) = 0.067y - 1.003$$

αφού η πάνω βάση του τραπεζίου  $B'_2$  που είναι  $y = 0.33$  τέμνει την αριστερή πλευρά του τραπεζίου  $B'_3$  που διέρχεται από τα σημεία (20, 0.33) και (25.10, 0.67) και έχει εξίσωση:

$$\frac{y - 0.33}{0.67 - 0.33} = \frac{x - 20}{25.10 - 20} \Leftrightarrow y = 0.067 \cdot x - 1.003$$

στο σημείο  $y = 0.33 = 0.067 \cdot x - 1.003 \Rightarrow x \approx 20$

Όμοια,

$$\max_{y \in [25.10, 30]} \{B'_2(y), B'_3(y)\} = B'_3(y \in [25.10, 30]) = 0.67$$

Έτσι λοιπόν προσδιορίζουμε τελικά ότι η ένωση-ασαφές σύνολο  $B = B'_2 \cup B'_3$ , είναι το χωρίο που περιλαμβάνεται από το περίγραμμα (αβγδε) και τον οριζόντιο άξονα των y.

Ακολούθως υπολογίζουμε την αποασαφοποίηση του B.

### 1. Μέσω της μεθόδου "κέντρου βάρους-centroid defuzzification"

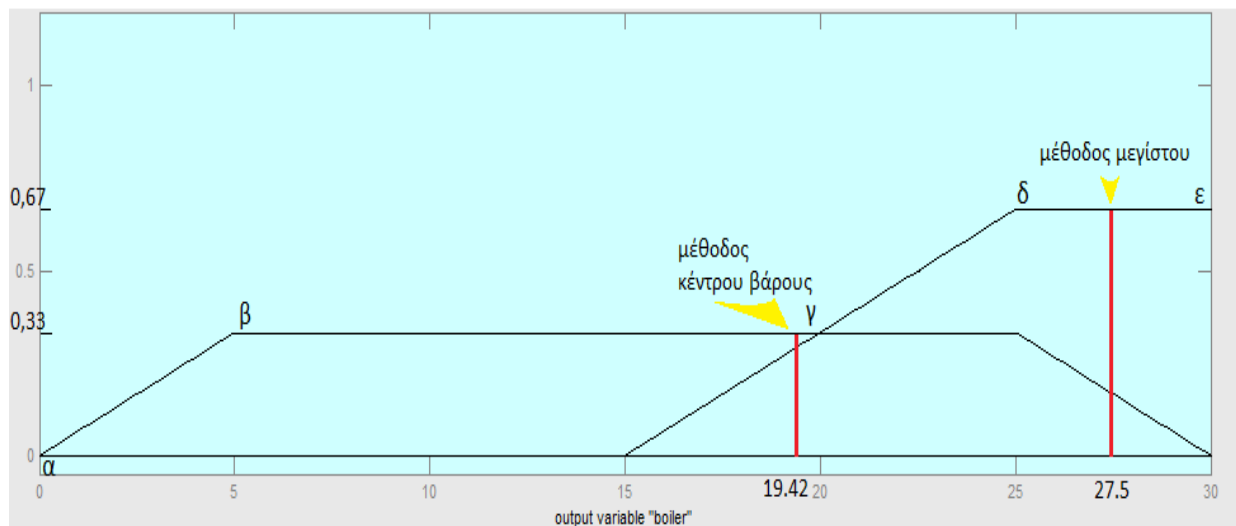
βρίσκουμε μέσω απλών ολοκληρώσεων:

$$F(x) = b = \frac{\int \mu_B(y) \cdot y dy}{\int \mu_B(y) dy} =$$

$$\frac{\int_0^5 (0.0066y) \cdot y dy + \int_5^{20} (0.33) \cdot y dy + \int_{20}^{25.10} (0.67y - 1.003) \cdot y dy + \int_{25.10}^{30} (0.67) \cdot y dy}{\int_0^5 (0.0066y) dy + \int_5^{20} 0.33 dy + \int_{20}^{25.10} (0.067y - 1.003) dy + \int_{25.10}^{30} (0.67) dy}$$

$$= \dots \approx 19.42$$

Το αποτέλεσμα αυτής της μεθόδου απεικονίζεται στο διάγραμμα της εικόνας 36.



Εικόνα 36: Αποτελέσματα εξόδου με τη μέθοδο του κέντρου βάρους και του μεγίστου.

### 2. Μέσω της μεθόδου του μεγίστου, δηλαδή μέσω της σχέσης:

$$F(X) = b = \max \{\mu_B(y_i)\}$$

είναι προφανές από το παραπάνω σχήμα ότι όλα τα σημεία του διαστήματος  $[25.10, 30]$  είναι μέγιστα και επομένως το αντιπροσωπευτικότερο σημείο είναι το μέσον του διαστήματος, δηλαδή  $F(x) = b = 27.55$ . Το αποτέλεσμα αυτής της μεθόδου απεικονίζεται στο διάγραμμα της εικόνας 36.

Τέλος επισημαίνεται ότι η μέθοδος του κέντρου βάρους που δίνει τιμή  $F(x) = 19.42$  είναι πιο αξιόπιστη και πιο αντιπροσωπευτική του B αφού λαμβάνει υπόψιν και τα δύο σύνολα εξόδου που ενεργοποιούνται, έναντι της τιμής  $F(x) = 27.55$  που δίνει η μέθοδος του μεγίστου.

## 5 Ασαφής έλεγχος έντροχου ρομποτικού οχήματος.

Σε αυτό το κεφάλαιο θα γίνει αναφορά στην ανάπτυξη και στην υλοποίηση του ασαφούς ελεγκτή που δημιουργήθηκε για την πλοήγηση του ρομποτικού οχήματος. Η εύρεση της θέσης του οχήματος στο χώρο επιτυγχάνεται με την μέθοδο της οδομετρίας και με την χρήση κατάλληλων εξισώσεων.

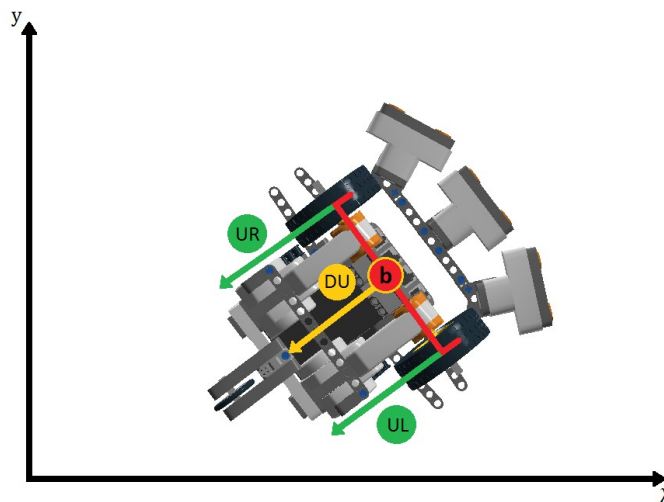
Η ανάπτυξη του ελεγκτή έγινε σε δύο στάδια. Αρχικά για την κατανόηση της ορθής λειτουργίας ενός ασαφούς ελεγκτή για την πλοήγηση ενός ρομποτικού οχήματος δημιουργήθηκε ένας απλός ελεγκτής αποφυγής αντικειμένων, χωρίς να διαθέτει την ικανότητα εύρεσης συγκεκριμένου σημείου στο χώρο και πλοήγησης μέχρι αυτό.

Έπειτα επανασχεδιάστηκε με την προσθήκη του σφάλματος κατεύθυνσης δίνοντας την δυνατότητα στο όχημα την αποφυγή αντικειμένων και ταυτόχρονα την πλοήγηση σε προκαθορισμένο σημείο στο χώρο.

### 5.1 Αντίληψη του χώρου

#### 5.1.1 Οδομετρία

Για την χαρτογράφηση του οχήματος στο χώρο ώστε να γνωρίζει το πρόγραμμα ελέγχου που βρίσκεται κάθε στιγμή σε ένα καρτεσιανό μοντέλο αναφοράς  $(x,y)$  χρησιμοποιήθηκε η μέθοδος της οδομετρίας η οποία επιτρέπει να εκτιμηθεί η αλλαγή θέσης στη πάροδο του χρόνου ενός έντροχου οχήματος χρησιμοποιώντας δεδομένα από αισθητήρες κίνησης.



Εικόνα 37: Το μεταξόνιο ( $b$ ), τα διαστήματα που έχουν κινηθεί οι τροχοί ( $U_R$  και  $U_L$ ) και το όχημα συνολικά ( $U_D$ )

Εάν είναι γνωστή η απόσταση που έχει διανύσει ο κάθε τροχός ( $U_L$  και  $U_R$ ) (Εικόνα 37) τότε η συνολική μετατόπιση του οχήματος θα είναι:

$$U_D = \frac{U_R + U_L}{2}$$

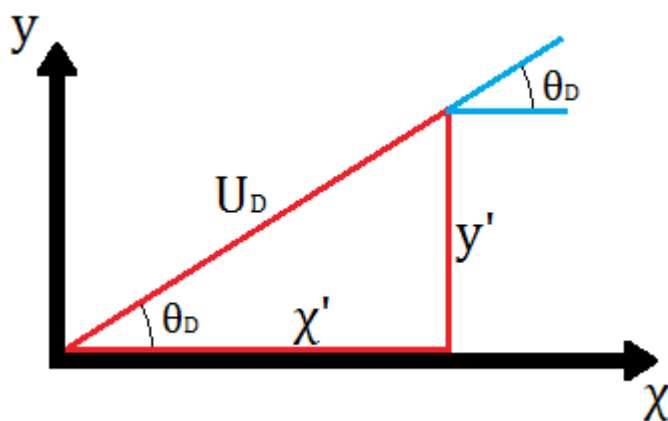
και για να βρεθεί η γωνία του οχήματος εκείνη την δεδομένη στιγμή:

$$\theta_D = \frac{U_R - U_L}{b} \text{ (rad)}$$

$$\cos(\theta_D) = \frac{x'}{U_D} \Rightarrow x' = \cos(\theta_D) \cdot U_D$$

$$\sin(\theta_D) = \frac{y'}{U_D} \Rightarrow y' = \sin(\theta_D) \cdot U_D$$

Η εύρεση των καινούργιων σημείων  $x$  και  $y$  γίνεται με την χρήση των γνωστών τριγωνομετρικών εξισώσεων (Εικόνα 38):



Εικόνα 38 Εύρεση  $x'$  και  $y'$

Το NXT έχει τη δυνατότητα μέτρησης των μοιρών που έχει περιστραφεί ο κάθε κινητήρας. Επίσης είναι γνωστό πως ο λόγος της περιφέρειας ενός κύκλου προς την διάμετρό του

ισούται με  $\pi$  ( $\pi = \frac{C}{d}$ )

η διάμετρος του κάθε τροχού του οχήματος είναι  $d=4.30$  cm

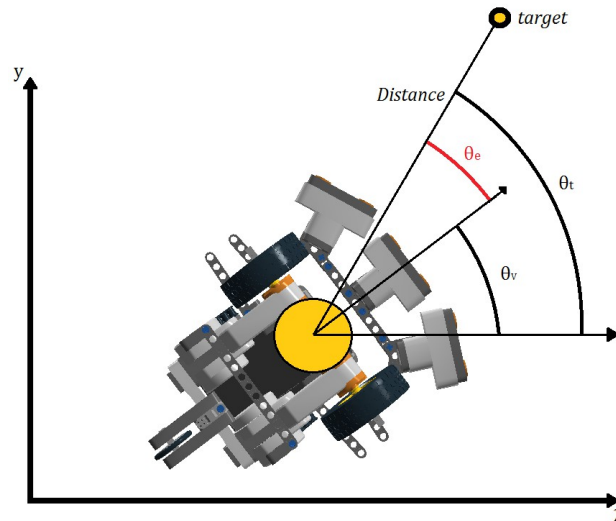
άρα η περιφέρεια του κάθε τροχού (C) κάνοντας χρήση του προηγούμενου τύπου θα είναι: 13.50cm.

Γνωρίζοντας ότι όταν ο κινητήρας θα κάνει μια πλήρη περιστροφή ο αισθητήρας από τους κινητήρες θα έχει καταγράψει  $360^\circ$  η μετατόπιση για το κάθε τροχό μπορεί να υπολογιστεί από την ακόλουθη συνάρτηση:

$$U_{LID}(t) = (\text{Degrees Sensor Reading}(t) - \text{Degrees Sensor Reading}(t-1)) \cdot \frac{C}{360}$$

## 5.1.2 Πλοήγηση

Ο ελεγκτής πλοήγησης τείνει να το οδηγεί το όχημα από μια δεδομένη αρχική θέση σε ένα σταθερό ή κινούμενο σημείο-στόχος. Η απόκριση του ελεγκτή οδηγεί το όχημα προς τα αριστερά, προς τα δεξιά ή προς τα εμπρός, ανάλογα με τις δύο μεταβλητές εισόδου: την απόσταση ( $d$ ) μεταξύ του οχήματος και του στόχου και το σφάλμα κατεύθυνσης ( $\theta_e$ ). Το σφάλμα κατεύθυνσης είναι η διαφορά της επιθυμητής κατεύθυνσης ( $\theta_t$ ) με την τρέχουσα κατεύθυνση του οχήματος ( $\theta_v$ ).



Εικόνα 39: Προσδιορισμός του σφάλματος κατεύθυνσης

$$Distance = \sqrt{(\chi_t - \chi_v)^2 + (y_t - y_v)^2}$$
$$\theta_e = \theta_t - \theta_v = \text{atan2}\left(\frac{y_t - y_v}{\chi_t - \chi_v}\right) - \theta_v$$

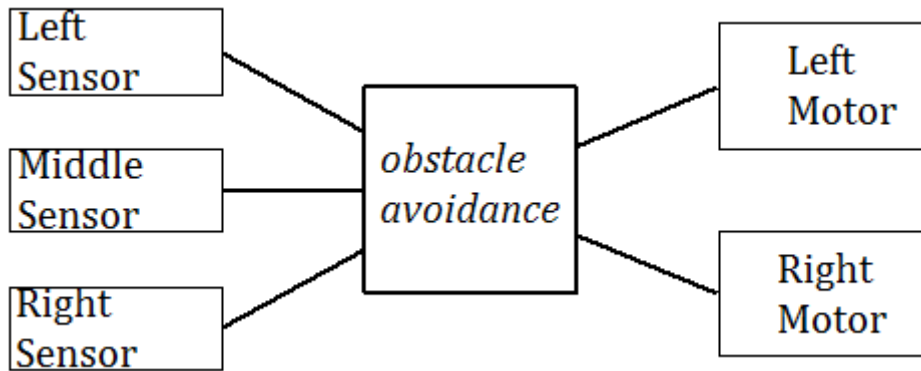
## 5.2 Ασαφείς ελεγκτές

Κατά την διάρκεια της πτυχιακής αναπτύχθηκαν δύο ασαφείς ελεγκτές για τον έλεγχο του οχήματος, η μεθοδολογία τους είναι παρόμοια και θα αναλυθούν παρακάτω.

### 5.2.1 Ελεγκτής αποφυγής εμποδίων.

Για την κατανόηση της ανάπτυξης και της λειτουργίας ενός ασαφούς ελεγκτή αναπτύχθηκε αρχικά ένα ελεγκτής αποφυγής αντικειμένων χωρίς την δυνατότητα κατεύθυνσης προς ένα σημείο-στόχο.

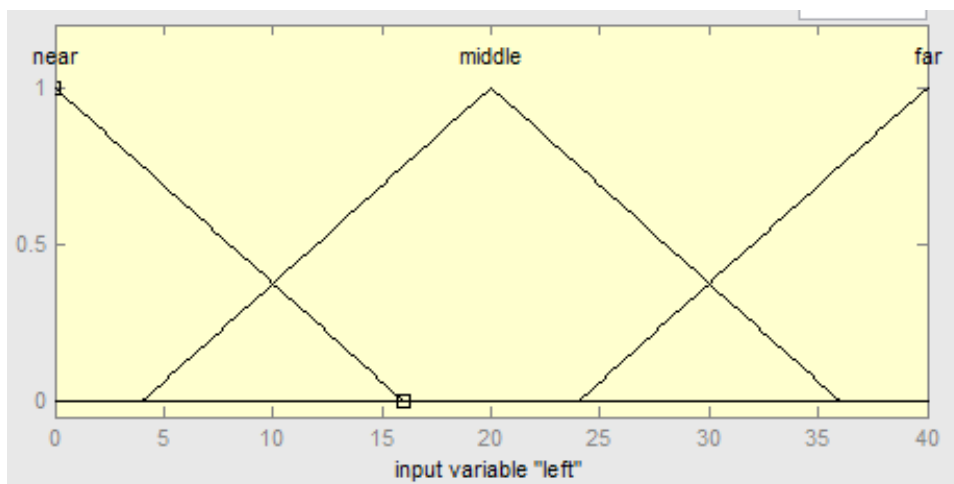
Ο ελεγκτής αποτελείται από τρεις εισόδους, μία είσοδο για κάθε αισθητήρα υπερήχων και δύο εξόδους μια κάθε κινητήρα (Εικόνα 40).



Εικόνα 40 Σχεδιάγραμμα ελεγκτή

Η κάθε μεταβλητή εισόδου δέχεται ως δεδομένα εκατοστά που αντιστοιχούν την απόσταση από κάποιο εμπόδιο και χωρίζεται σε τρία ασαφή σύνολα (Εικόνα 41):

- near = [0,1,...,16]
- middle = [4,5,...,36]
- far = [24,26,...,40]

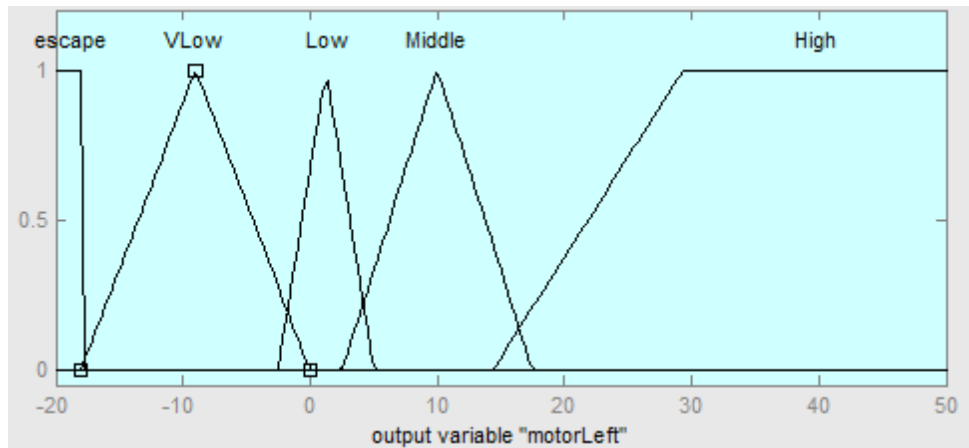


Εικόνα 41: Λεκτική μεταβλητή εισόδου

Η κάθε μεταβλητή εξόδου εξάγει το ποσοστό της δύναμης που θα δοθεί στο κινητήρα και χωρίζεται σε πέντε ασαφή σύνολα (Εικόνα 42).

- Escape = [-20,...,-18]
- Very Low = [-18, -17,..., 0]
- Low = [-2.5,...,5]
- Middle = [2.5,..., 17.5]
- High = [14.38,...,50]
- Για τον δεξί κινητήρα το Escape ορίζεται: [48,...,50]





Εικόνα 42:Λεκτική μεταβλητή εξόδου.

Οι κανόνες τους ασαφούς ελεγκτή ορίζονται ενδεικτικά:

*if (left is near) and (middle is near) and (right is middle)*  
*then (motorLeft is High)(motorRight is Vlow)*  
*if (left is near) and (middle is middle) and (right is far)*  
*then (motorLeft is High)(motorRight is Low)*  
*if (left is near) and (middle is far) and (right is near)*  
*then (motorLeft is Middle)(motorRight is Middle)*  
*if (left is near) and (middle is near) and (right is near)*  
*then (motorLeft is Escape)(motorRight is Escape)*

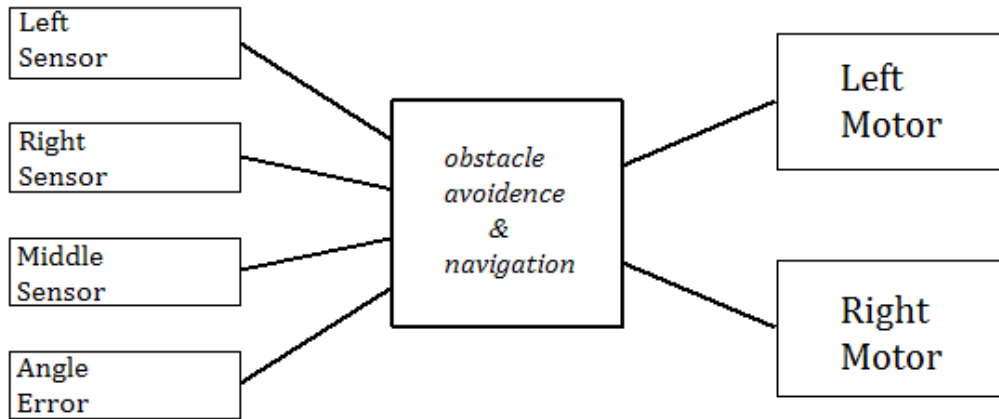
Όσο δεν υπάρχει εμπόδιο το όχημα θα κινείται ευθείας όταν ανιχνευθεί κάποιο εμπόδιο η τιμή που θα λάβει ο αισθητήρας θα είναι κάτω από τα 40 cm και ανάλογα με την απόσταση του εμποδίου και το πόσοι αισθητήρες το έχουν αντιληφθεί θα ενεργοποιηθούν και οι αντίστοιχοι κανόνες.

### 5.2.2 Ελεγκτής αποφυγής εμποδίων και διόρθωση κατεύθυνσης.

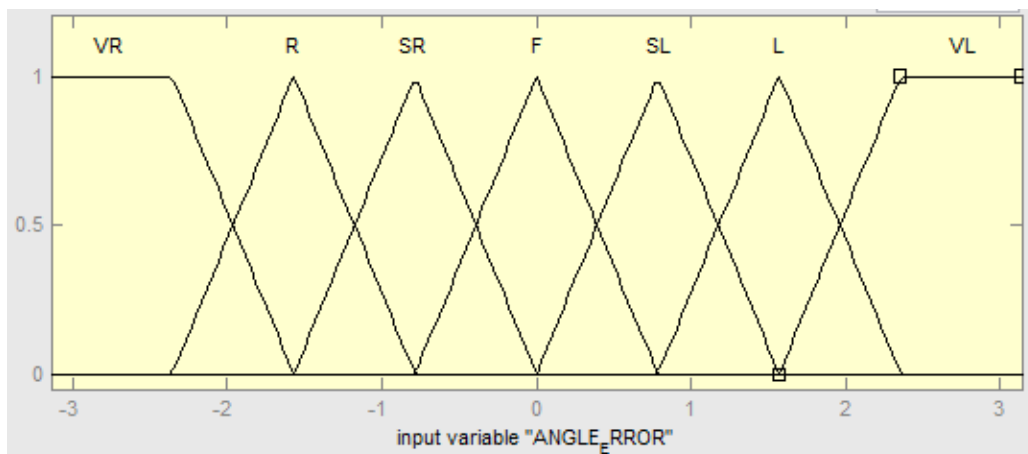
Ο δεύτερος ελεγκτής που αναπτύχθηκε διαθέτει ως τέταρτη είσοδο το σφάλμα κατεύθυνσης. όπου το υπερσύνολο αναφοράς των τιμών είναι από  $-\pi$  έως  $\pi$  και χωρίζονται σε εφτά ασαφή σύνολα (Εικόνα 43 Εικόνα 44):

- $VR = [-3.14, \dots, -1.57]$  ( $-180^\circ$  έως  $-90^\circ$ )
- $R = [-2.356, \dots, -0.7854]$  ( $-135^\circ$  έως  $-45^\circ$ )
- $SR = [-1.57, \dots, 0]$  ( $-90^\circ$  έως  $0^\circ$ )
- $F = [-0.7854, \dots, 0.7854]$  ( $-45^\circ$  έως  $45^\circ$ )

- $SL = [0, \dots, 1.57]$  ( $0^\circ$  έως  $-90^\circ$ )
- $L = [0.7854, \dots, 2.356]$  ( $45^\circ$  έως  $135^\circ$ )
- $VL = [1.57, \dots, 3.14]$  ( $90^\circ$  έως  $180^\circ$ )



Εικόνα 43: Σχεδιάγραμμα ελεγκτή.



Εικόνα 44: Μεταβλητή εισόδου “σφάλμα γωνίας”

Επίσης για λόγους ευκολίας οι τιμές των εισόδων-αισθητήρων έχουν μειωθεί σε δύο ασαφή σύνολα:

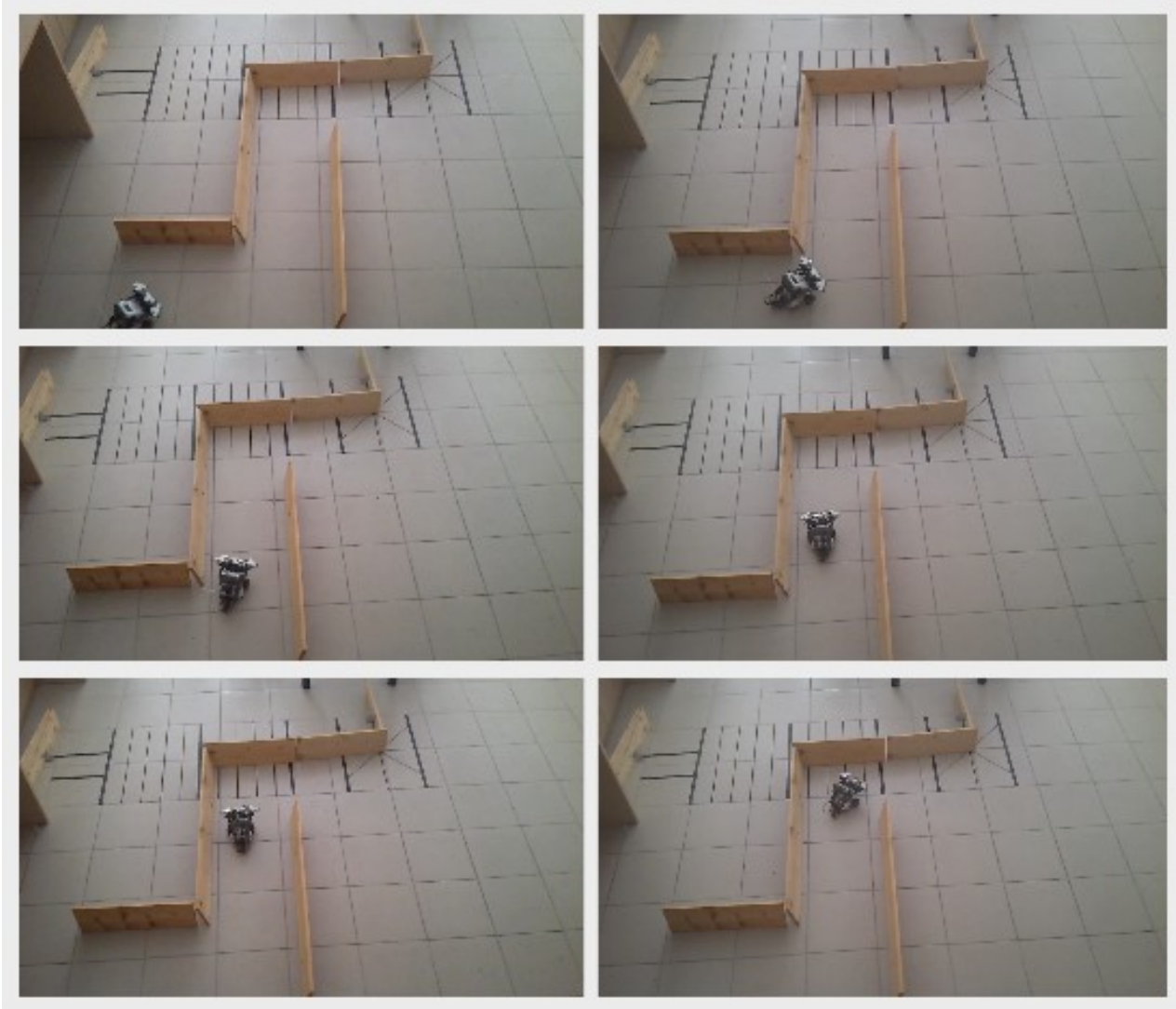
- $near = [0, \dots, 35]$
- $far = [20, \dots, 50]$

Όπως και πριν όσο δεν υπάρχει εμπόδιο το όχημα θα κινείται διορθώνοντας την πορεία του ώστε να προσεγγίσει το σημείο-στόχο, όταν ανιχνευθεί κάποιο εμπόδιο η τιμή που θα λάβει ο αισθητήρας θα είναι κάτω από τα 40 cm και ανάλογα με την απόσταση του εμποδίου, το πόσοι αισθητήρες το έχουν αντιληφθεί αλλά σε συνδυασμό με τη γωνία σφάλματος θα ενεργοποιηθούν και οι αντίστοιχοι κανόνες αποφυγής.

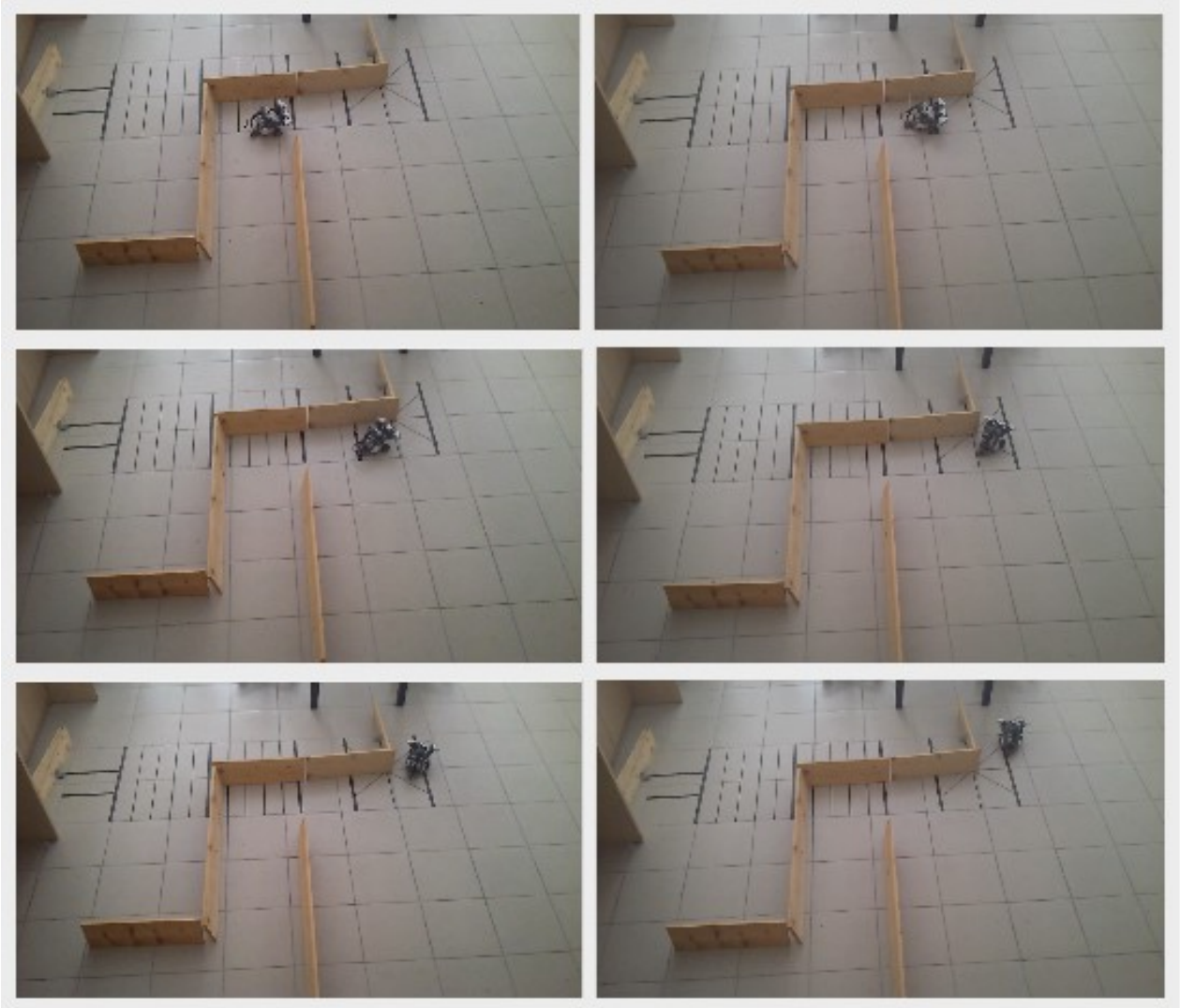
Οι τωρινοί κανόνες ακολουθούν παρόμοια λογική με τους προηγούμενους:

*if (left is near) and (middle is near) and (right is far) and (angle\_error is SL)  
then (motorLeft is High)(motorRight is Vlow)*

Στιγμιότυπα από την διεξαγωγή του πειράματος.

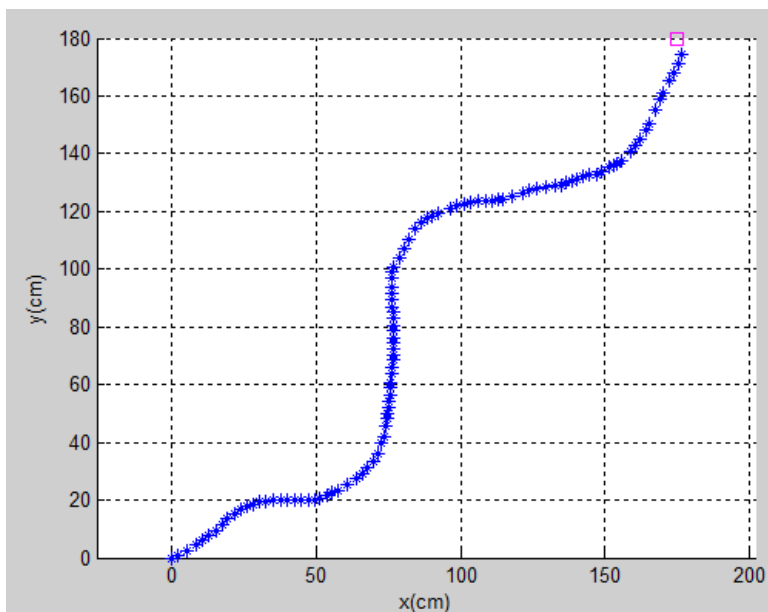


Εικόνα 45:Αποφυγή εμποδίων και διόρθωση πορείας 1

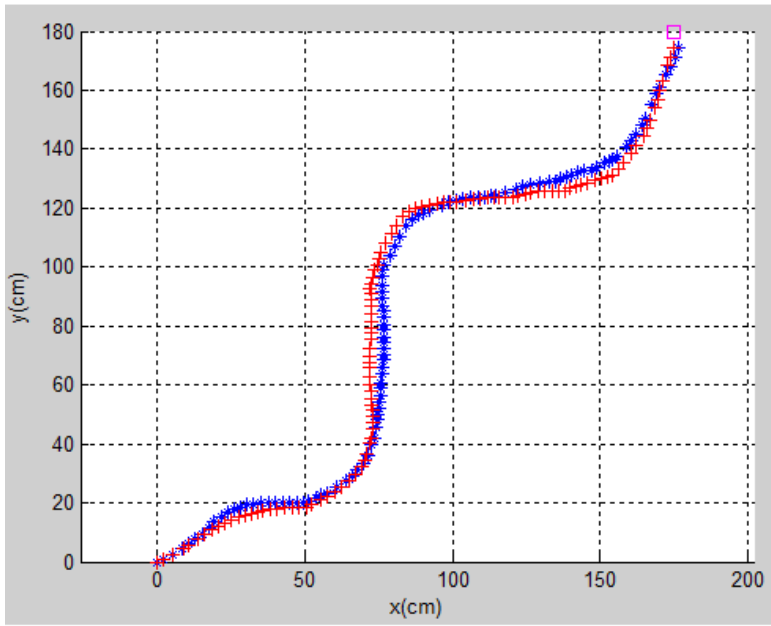


Εικόνα 46:Αποφυγή εμποδίων και διόρθωση πορείας 2

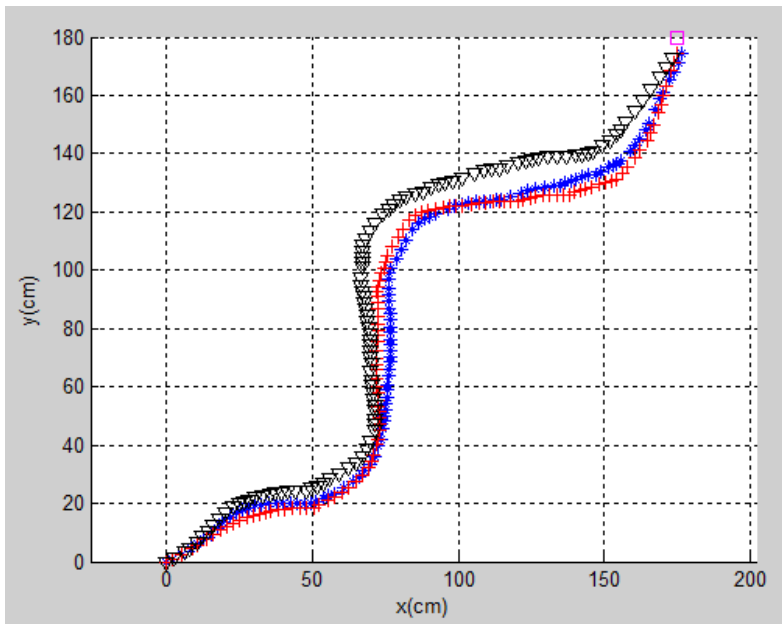
Παρακάτω απεικονίζονται οι τροχιές από τρεις διεξαγωγές του πειράματος.



Εικόνα 47:Πρώτη τροχιά οχήματος



Εικόνα 48:Εισαγωγή δεύτερης τροχιάς



Εικόνα 49:Εισαγωγή τρίτης τροχιάς

## 6 Συμπεράσματα

Το παραπάνω όχημα κατάφερε να εκτελέσει τις εργασίες για τις οποίες σχεδιάστηκε. Η ρομποτική σειρά Lego Mindstorm αποδείχθηκε ένα εύχρηστο και ολοκληρωμένο σετ ρομποτικής όπου ο καθένας μπορεί μέσα σε λίγες ώρες να σχεδιάσει και να θέσει σε κίνηση το πρώτο του ρομπότ.

Ο ασαφής έλεγχος αποδείχτηκε αρκετά εύκολος στο σχεδιασμό του και χωρίς περίπλοκους κανόνες το όχημα ήταν σε θέση να πλοηγηθεί ανάμεσα σε εμπόδια και να φτάσει στο τελικό του στόχο. Τα εμπόδια δοκιμάστηκαν σε διάφορες θέσεις στο χώρο και η απόκριση του ρομπότ ήταν συνήθως αρκετά καλή. Πρόβλημα παρουσιαζόταν όταν έφτανε σε αδιέξοδο και οι μετρήσεις στους τρεις αισθητήρες ήταν ταυτόχρονα παρόμοιες. Γενικά ο ασαφής έλεγχος είναι αρκετά ευέλικτος και με αποκρίσεις κοντά στη συμπεριφορά και στην ακρίβεια των ανθρώπων εφόσον έχει γίνει σωστή μελέτη των κανόνων.

Πολλά σφάλματα είχαν οι τρεις αισθητήρες υπερήχων καθώς αρκετές φορές οι τιμές που ελάμβαναν ήταν εσφαλμένες. Το πρόβλημα επιλύθηκε εν μέρη με το να τοποθετηθούν οι αισθητήρες σε διαφορετικές γωνίες. Επίσης άλλο ένα πρόβλημα ήταν ο τζόγος στους σερβοκινητήρες καθώς στην εσωτερική τους κατασκευή υπάρχουν πολλά επίπεδα γραναζιών.

## 7 Παράρτημα

```
function control3
OpenUltrasonic(SENSOR_1)
OpenUltrasonic(SENSOR_2)
OpenUltrasonic(SENSOR_4)

port1=SENSOR_4;
port2=SENSOR_2;
port3=SENSOR_1;

fis = readfis('navigator2');
b = 10.9;
UL = 0;
UR = 0;

data1.Position = 0;
data2.Position = 0;

x_vehicle = 0;
y_vehicle = 0;
x_old = 0;
y_old = 0;
targetx = 200;
Targety = 150;

apostasi = 150;
DTheta = 0;
vehicle_theta = 0;
Old_Theta = 0;

motor1 = NXTMotor('A');
motor2 = NXTMotor('B');

motor1.ResetPosition();
motor2.ResetPosition();

while apostasi>5

    UL = data1.Position/26.67;
    UR = data2.Position/26.67;
    motor1.ResetPosition();
    motor2.ResetPosition();
```

```

DU = (UR+UL)/2;
DTheta = ((UR-UL)/b);
vehicle_theta = DTheta + Old_Theta;
Old_Theta = vehicle_theta;

hold on
plot(x_vehicle,y_vehicle, '*')
plot(targetx,targety)
axis equal

x_vehicle = (x_old + DU*cos(vehicle_theta));
y_vehicle = (y_old + DU*sin(vehicle_theta));

x_old = x_vehicle;
y_old = y_vehicle;

new_x = targetx-x_vehicle;
new_y = targety-y_vehicle;

apostasi = sqrt(new_x^2 + new_y^2);
new_theta = atan2(new_y,new_x);
angle_error = new_theta - vehicle_theta

while abs(angle_error)>=pi

    if angle_error<(-1*pi);
        angle_error = angle_error+2*pi;
    end

    if angle_error>pi
        angle_error = angle_error-2*pi;
    end

end

motor1.ResetPosition();
motor2.ResetPosition();

scanLeft = GetUltrasonic(SENSOR_4)
scanMiddle = GetUltrasonic(SENSOR_2)
scanRight = GetUltrasonic(SENSOR_1)

```



```

    if scanLeft > 50
        scanLeft = 50
    end

    if scanRight > 50
        scanRight = 50
    end

    if scanMiddle > 50
        scanMiddle = 50
    end

    Y = evalfis([scanLeft,scanMiddle,scanRight,angle_error],fis);

    mPowerLeft = round(Y(1));
    mPowerRight = round(Y(2));

    motor1 = NXTMotor('A', 'Power', mPowerLeft, 'SmoothStart',
        true);
    motor2 = NXTMotor('B', 'Power', mPowerRight,
        'SmoothStart',true);

    motor1.SendToNXT();
    motor2.SendToNXT();

    pause(.3)
    StopMotor('all', 'off');

    motor1.WaitFor();
    motor2.WaitFor();

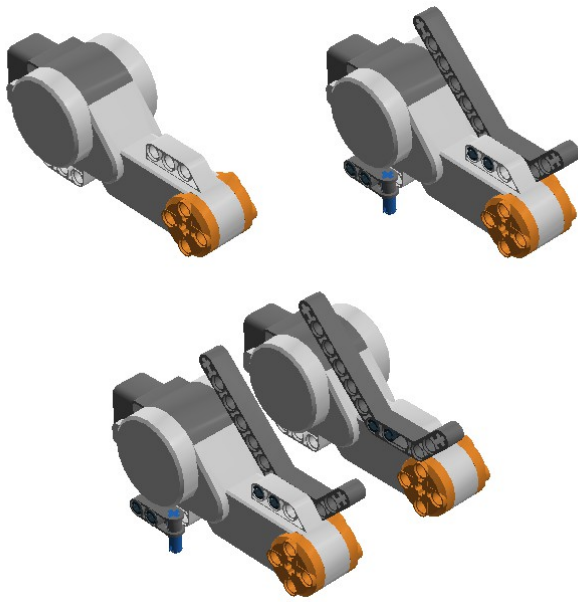
    data1 = motor1.ReadFromNXT();
    data2 = motor2.ReadFromNXT();
end

CloseSensor(SENSOR_1)
CloseSensor(SENSOR_4)
CloseSensor(SENSOR_2)

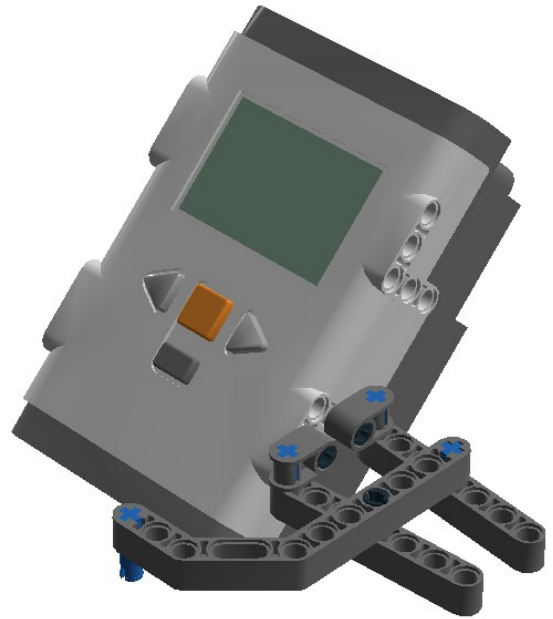
StopMotor('all', 'off');

end

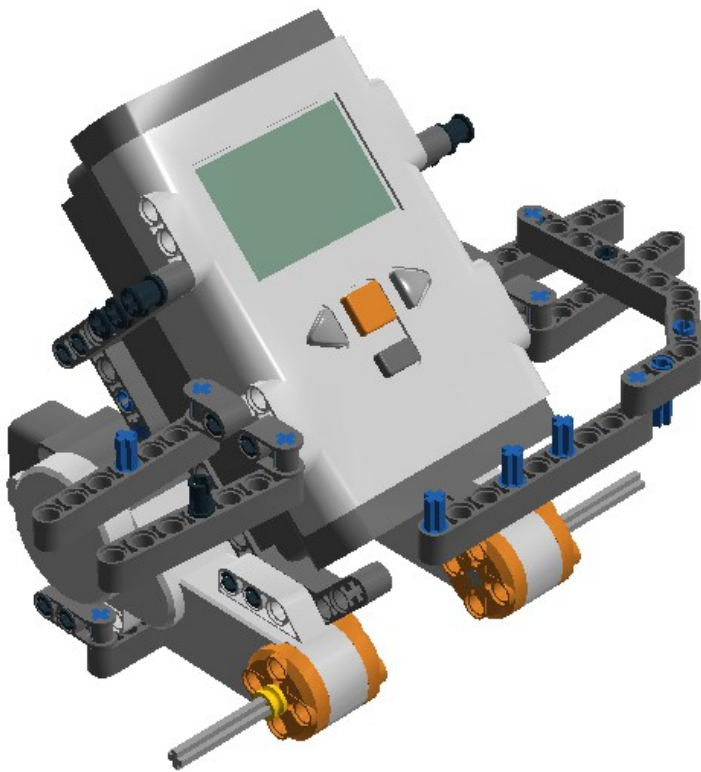
```



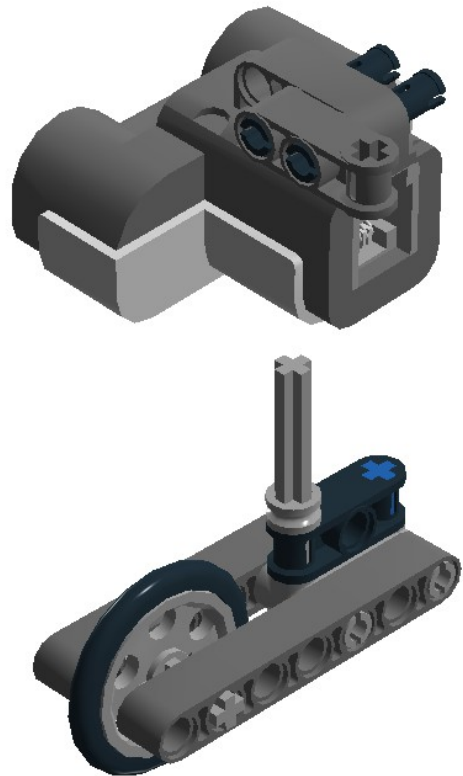
Εικόνα 50 Στιγμιότυπο από το lego Digital Designer 1



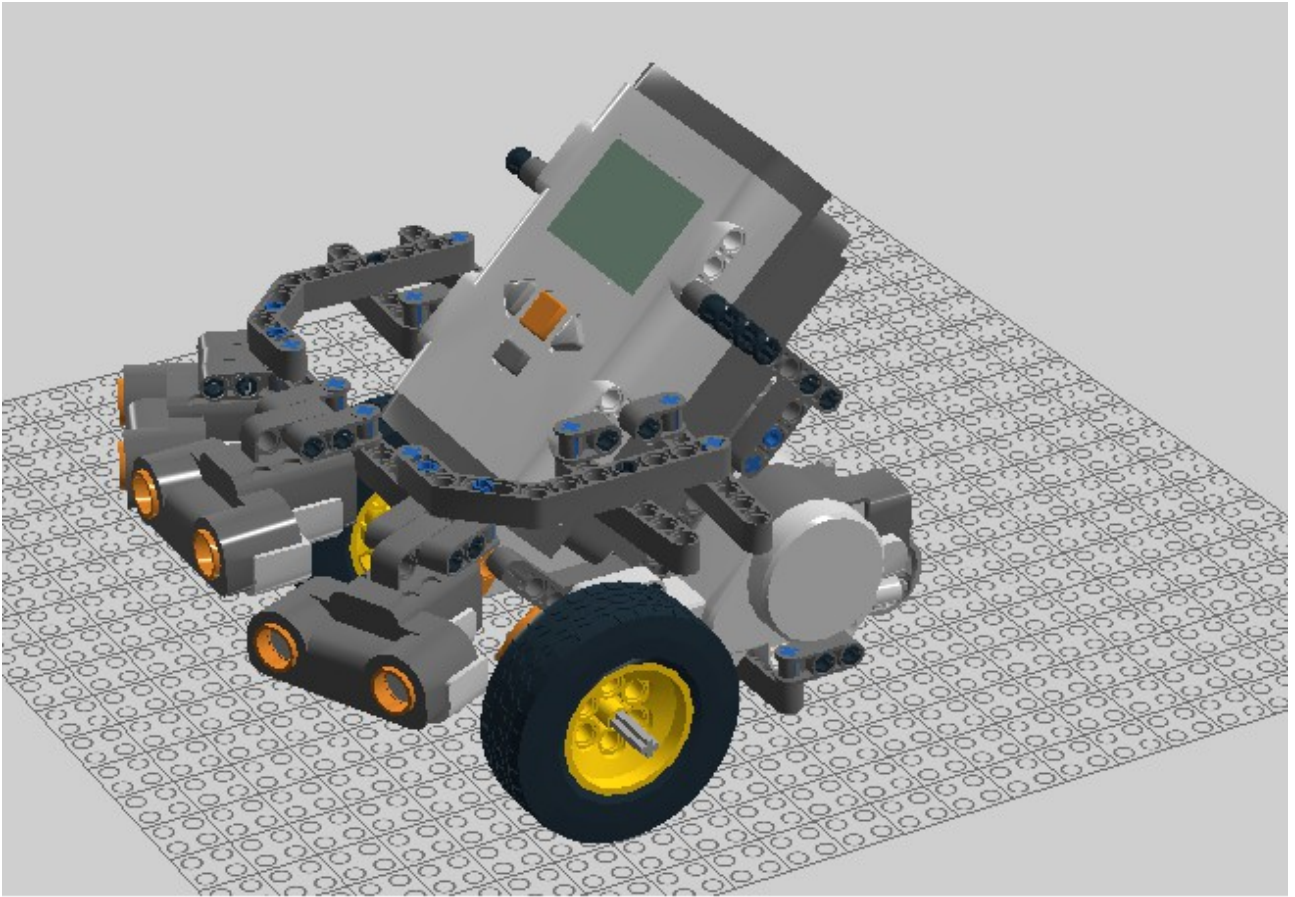
Εικόνα 51 Στιγμιότυπο από το lego Digital Designer 2



Εικόνα 52 Στιγμιότυπο από το lego Digital Designer 3



Εικόνα 53 Στιγμιότυπο από το lego Digital Designer 4



*Εικόνα 54 Στιγμιότυπο από το lego Digital Designer 5*

## 8 ΒΙΒΛΙΟΓΡΑΦΙΑ

Building Robots with LEGO Mindstorms NXT(Dave Astolfo, Mario Ferrari, Giulio Ferrari).

LEGO Mindstorm Guide Manual.

Extreme NXT: Extending the LEGO MINDSTORMS NXT to the Next Level, Second Edition(Michael Gasperi, Philippe Hurbain).

Ιστολόγιο εκπαιδευτικής Ρομποτικής του Παιδαγωγικού Τμήματος Δημοτικής Εκπαίδευσης του Πανεπιστημίου Κρήτης.

Εταιρία παραγωγής εγκεκριμένων αισθητήρων για LEGO Mindstorms-Hitechniks.

NXT user guide

Teaching Practical Engineering for Freshman Students using the RWTH-Mindstorms NXT Toolbox for MATLAB

Εισαγωγή στην ΑΣΑΦΗ ΛΟΓΙΚΗ-Δρ. Γιάννης Θεοδώρου

ΕΥΦΥΗΣ ΕΛΕΓΧΟΣ-Ροβέρτος-Ε.Κινγκ

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ Γ Έκδοση

RWTH – Mindstorms NXT Toolbox -RWTH AACHEN UNIVERSITY

Σχεδιασμός ασαφούς ελεγκτή για πλοήγηση ρομποτικών οχημάτων – Μεταπτυχιακή εργασία Δεδούση Κωνσταντίνου.

Έλεγχος ρομποτικού οχήματος και αποφυγή δυναμικών, στατικών εμποδίων με τη βοήθεια ελεγκτών ασαφούς λογικής – Μεταπτυχιακή εργασία Καλυκάκη Εμμανουήλ