

Διεθνές Πανεπιστήμιο της Ελλάδος

Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών

---

Διπλωματική Εργασία

Πρόγραμμα Μεταπτυχιακών Σπουδών στη Ρομποτική

---

## Μελέτη, σχεδίαση και κατασκευή πανκατευθυντικού ρομπότ με σουηδικούς τροχούς και ελεγκτή Raspberry Pi

Πάρης Χοράτσιου-Ράντου

*Διεθνές Πανεπιστήμιο της Ελλάδος*

Σέρρες, 29-9-2023

Εργασία που υποβλήθηκε στο

Πρόγραμμα Μεταπτυχιακών Σπουδών στη Ρομποτική,

του Διεθνούς Πανεπιστημίου της Ελλάδος,

για τη μερική εκπλήρωση υποχρεώσεων για το Δίπλωμα Ειδίκευσης στη

Ρομποτική

Επιβλέπων Καθηγητής: **Ιωάννης Καλόμοιρος**



*Ο κάτωθι υπογεγραμμένος φοιτητής, έχοντας επίγνωση των συνεπειών του Νόμου περί λογοκλοπής, δηλώνω υπεύθυνα ότι είμαι συγγραφέας αυτής της Διπλωματικής Εργασίας, αναλαμβάνοντας την ευθύνη επί ολοκλήρου του κειμένου. Έχει γίνει αναφορά σε όλες τις πηγές που έχω συμπεριλάβει. Όλα τα κείμενα και τα διάφορα στοιχεία που υπάρχουν μέσα στην εργασία, είτε είναι από βιβλία, είτε είναι από το διαδίκτυο, ή άλλες εργασίες είναι αναγνωρισμένο από εμένα ως πνευματικό έργο άλλου συγγραφέα. Έχω δε αναφέρει τα ονόματα τους και την πηγή από την οποία προέρχονται.*

*Πάρης Χοράτσιου - Ράντου*

# Περίληψη

Τα τροχήλατα ρομποτικά οχήματα είναι αυτόνομα ή τηλεκατευθυνόμενα οχήματα που χρησιμοποιούνται για διάφορους σκοπούς. Αυτά τα ρομποτικά οχήματα μπορούν να εκτελούν εργασίες όπως η μεταφορά εμπορευμάτων σε βιομηχανικές εγκαταστάσεις, για εξερεύνηση ανεξερεύνητων περιοχών, για παροχή ανθρωπιστικής βοήθειας σε επικίνδυνες καταστάσεις κλπ. Τα οχήματα αυτά έχουν γίνει ένα αναπόσπαστο κομμάτι της ζωής μας. Προσφέρουν αυξημένη αποδοτικότητα και ασφάλεια σε διάφορους τομείς και συνεχώς εξελίσσονται με τη χρήση προηγμένων τεχνολογιών όπως οι αισθητήρες, η τεχνητή νοημοσύνη και η αυτοματοποίηση.

Στα πλαίσια της παρούσας μεταπτυχιακής διατριβής μελετήθηκε σε βάθος η κίνηση ενός οχήματος με τέσσερις σουηδικούς τροχούς 45 μοιρών. Πραγματοποιήθηκε μελέτη της κινηματικής του συγκεκριμένου οχήματος, αναλύοντας τόσο το ευθύ κινηματικό όσο και το αντίστροφο κινηματικό πρόβλημα. Στη συνέχεια ακολούθησε η διαδικασία των πειραμάτων.

Συγκεκριμένα, ολοκληρώθηκαν κάποια πειράματα, τα οποία είχαν ως σκοπό την μελέτη της κίνησης του οχήματος, πραγματοποιώντας μια προσομοίωση με την βοήθεια ηλεκτρονικού υπολογιστή. Τέλος, έγινε προσπάθεια κατασκευής ενός λειτουργικού οχήματος με 4 σουηδικούς τροχούς χρησιμοποιώντας σαν εργαλείο, όλες τις γνώσεις που αποκτήθηκαν κατά την διάρκεια των σπουδών.

Ακόμη, επειδή το όχημα πρέπει να υποστηρίζει ένα σύστημα τηλεκατεύθυνσης, δημιουργήσαμε ένα γραφικό περιβάλλον με την βοήθεια της γλώσσας προγραμματισμού Python το οποίο μας δίνει την δυνατότητα σε συνδυασμό με το ηλεκτρονικό κύκλωμα και τον ελεγκτή του οχήματος να μπορούμε απομακρυσμένα να ελέγχουμε το όχημα.

Τέλος, ολοκληρώνοντας όλα τα πειράματα με επιτυχία μπορούμε να αποδείξουμε ότι όντως ένα όχημα με 4 σουηδικούς τροχούς είναι πανκατευθυντικό και μπορεί να κινηθεί προς όλες τις κατευθύνσεις χωρίς να χρειαστεί να αλλάξει φορά ή να περιστραφεί.

# Abstract

Wheeled robots are autonomous or remote-controlled vehicles used for various purposes. These robotic vehicles can perform tasks such as transporting goods to industrial facilities, exploring unexplored areas, providing humanitarian aid in dangerous situations, etc. These vehicles have become an integral part of our lives since they offer increased efficiency and safety in various areas and are constantly evolving with the use of advanced technologies such as sensors, artificial intelligence and automation.

In the context of this master's thesis, the movement of a vehicle with four Swedish wheels of 45 degrees was studied in depth. A study of the kinematics of the specific vehicle was carried out, analyzing both the forward kinematic and the inverse kinematic problem. Moving forward, the process of experiments followed.

In particular, some experiments were completed, which aim was to study the movement of the vehicle carrying out a simulation with the help of a computer.

Finally, an attempt was made to build a functional vehicle with 4 Swedish wheels, using as a tool all the knowledge acquired during my studies.

Moreover, because the vehicle must support a remote control system, we created a graphical environment with the help of the Python programming language in combination with an electronic circuit and a vehicle controller, which allow us to remotely control the vehicle.

Lastly, by completing all the experiments successfully we can prove that indeed a vehicle with 4 Swedish wheels is omnidirectional and can move in all directions its orientation.

# Ευχαριστίες

Με την ολοκλήρωση της μεταπτυχιακής διπλωματικής μου εργασίας, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους τους καθηγητές του τμήματος Ρομποτικής του Διεθνούς Πανεπιστημίου Ελλάδος για όλα όσα μου έμαθαν και συνέβαλλαν στην εκπόνησή της.

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή μου, κύριο Ιωάννη Καλόμοιρο, για την εμπιστοσύνη που μου έδειξε εξ' αρχής, την επιστημονική του καθοδήγηση, τις υποδείξεις του, την επιμονή του, το αμείωτο ενδιαφέρον του, τη συμπαράστασή του, και τη συνεχή τους υποστήριξη από την αρχή μέχρι το τέλος.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένειά μου, την σύντροφο μου και τους φίλους μου για όλη τη στήριξη, τη συμπαράσταση και την κατανόησή τους, καθ' όλη τη διάρκεια των σπουδών μου.

# Περιεχόμενα

Περίληψη.....	4
Abstract .....	5
Ευχαριστίες .....	6
Περιεχόμενα.....	7
Ευρετήριο Σχημάτων .....	9
Ευρετήριο πινάκων.....	11
<b>Κεφάλαιο I. Περιγραφή του προβλήματος.....</b>	<b>12</b>
1.1    Αυτόνομα Ρομπότ .....	12
1.2    Περιγραφή του προβλήματος.....	13
1.3    Στόχοι και αντικείμενα έρευνας .....	14
1.4    Συνεισφορά της διπλωματικής εργασίας.....	15
1.5    Περίγραμμα της μεταπτυχιακής εργασίας.....	15
<b>Κεφάλαιο II. Κατάσταση της τέχνης .....</b>	<b>17</b>
2.1    Τροχήλατα ρομποτικά συστήματα.....	17
2.2    Ο τροχός .....	19
2.3    Ο σουηδικός τροχός.....	21
2.4    Κατάσταση της τέχνης.....	21
2.5    Συνεισφορά στην κατάσταση της τέχνης.....	25
<b>Κεφάλαιο III. Η κινηματική του οχήματος.....</b>	<b>27</b>
3.1    Γενικοί ορισμοί για την κίνηση των τροχήλατων ρομπότ. ....	27
3.2    Ευθύ και αντίστροφο κινηματικό πρόβλημα οχήματος με 4 σουηδικούς τροχούς .....	32
3.3    Ευθύ και αντίστροφο κινηματικό πρόβλημα οχήματος με 4 σουηδικούς τροχούς: ένας εναλλακτικός υπολογισμός .....	36
<b>Κεφάλαιο IV. Μέθοδοι και υλικά.....</b>	<b>40</b>
4.1    Εισαγωγή .....	40
4.2    Εργαλεία ανάπτυξης .....	40
4.3    Προγραμματισμός του ελεγκτή – η γλώσσα προγραμματισμού Python .....	41
4.4    Τομέας Ηλεκτρονικών Κυκλωμάτων .....	43
4.4.1    Raspberry Pi.....	43
4.4.2    Υπόλοιπα ηλεκτρονικά εξαρτήματα.....	44
4.5    Σχεδιασμός 3D και 3D Εκτύπωση.....	48

4.5.1 Σχεδιαστικό πρόγραμμα Fusion 360.....	48
4.5.2 Τρισδιάστατοι εκτυπωτές .....	48
4.5.3 Λειτουργία 3D εκτύπωσης.....	49
4.5.4 Οι πιο δημοφιλείς μέθοδοι 3D εκτύπωσης .....	50
4.6 Έρευνα αγοράς υλικών .....	52
<b>Κεφάλαιο V. Ερευνητική μεθοδολογία.....</b>	<b>55</b>
5.1 Εισαγωγή .....	55
5.2 Σχεδιασμός 3D μοντέλου και 3D εκτύπωση .....	55
5.3 Ηλεκτρονικό κύκλωμα.....	59
5.4 Ανάλυση μαθηματικού μοντέλου στο MATLAB.....	61
5.5 Προγραμματισμός Server - Client .....	64
5.5.1 Δημιουργία Server-Client και GUI (Graphical user interface) .....	64
5.6 Πειραματικές δοκιμές. ....	68
6.1 Συμπεράσματα. ....	71
6.2 Μελλοντικές βελτιώσεις. ....	72
<b>Βιβλιογραφία.....</b>	<b>73</b>
<b>Παράρτημα I.....</b>	<b>75</b>
<b>Παράρτημα II.....</b>	<b>87</b>



# Ευρετήριο Σχημάτων

<i>Σχήμα 2.1 – Απεικόνιση διάφορων τύπων ρόδας</i>	<i>20</i>
<i>Σχήμα 2.2 Αριστερά διακρίνουμε τον σουηδικό τροχό με κυλινδράκια υπό 45 μοίρες και στα δεξιά ο σουηδικός τροχός με κυλινδράκια υπό 90 μοίρες</i>	<i>21</i>
<i>Σχήμα 2.3. Ολοκληρωμένο όχημα της ομάδας του ιδρύματος της Ρουμανίας</i>	<i>23</i>
<i>Σχήμα 2.4 – Τελικό όχημα της ομάδας του πανεπιστημίου της Κίνας</i>	<i>24</i>
<i>Σχήμα 2.5 – Κινήσεις στον χώρο μελέτης του οχήματος της ομάδας του ιδρύματος Nanjing</i>	<i>24</i>
<i>Σχήμα 2.6 – Απεικονίσεις οχημάτων με περισσότερους από τέσσερις τροχούς</i>	<i>25</i>
<i>Σχήμα 3.1 – Απεικόνιση του οχήματος στο πλαίσιο αναφοράς</i>	<i>27</i>
<i>Σχήμα 3.2 – Απεικόνιση οχήματος που περιστράφηκε κατά γωνία <math>\theta</math></i>	<i>28</i>
<i>Σχήμα 3.3 – Απεικόνιση τροχού mecanum στο πλαίσιο αναφοράς του οχήματος</i>	<i>32</i>
<i>Σχήμα 3.4 – Απεικόνιση οχήματος στο παγκόσμιο πλαίσιο αναφοράς</i>	<i>33</i>
<i>Σχήμα 3.5 – Το ρομπότ mecanum με τέσσερις σουηδικούς τροχούς</i>	<i>36</i>
<i>Σχήμα 4.1 - Γραναζωτός κινητήρας DC</i>	<i>44</i>
<i>Σχήμα 4.2 - Αναπαράσταση σχέσης ροπής και ταχύτητας</i>	<i>45</i>
<i>Σχήμα 4.3 - Μπαταρία Li-Po 2700 mAh</i>	<i>45</i>
<i>Σχήμα 4.4 - Διάταξη Dual H Bridge</i>	<i>46</i>
<i>Σχήμα 4.5 – Διάταξη κίνησης του ηλεκτρικού φορτίου στις δυο εναλλακτικές διαδρομές</i>	<i>47</i>
<i>Σχήμα 4.6 – Αναπαράσταση διάφορων παλμών σήματος PWM</i>	<i>47</i>
<i>Σχήμα 4.7 – Ολοκληρωμένη τρισδιάστατη εκτύπωση</i>	<i>49</i>
<i>Σχήμα 4.8 – Μέθοδος τρισδιάστατης εκτύπωσης SLS</i>	<i>50</i>
<i>Σχήμα 4.9 – Μέθοδος τρισδιάστατης εκτύπωσης SLA</i>	<i>51</i>

<i>Σχήμα 4.10 – Σασί αλουμινίου και σουηδικοί τροχοί</i>	<i>52</i>
<i>Σχήμα 4.11 – Γραναζωτοί κινητήρες συνεχούς ρεύματος</i>	<i>52</i>
<i>Σχήμα 4.12 – Raspberry Pi 4</i>	<i>52</i>
<i>Σχήμα 4.13 – Controller Board L298N</i>	<i>52</i>
<i>Σχήμα 4.14 – Μπαταρία Li-Po</i>	<i>52</i>
<i>Σχήμα 4.15 – Φορτιστής μπαταρίας Li-Po IMAX B6</i>	<i>52</i>
<i>Σχήμα 4.16 – Πλακέτα LM2596 DC-DC converter</i>	<i>52</i>
<i>Σχήμα 5.1 - Το σασί της κατασκευής</i>	<i>55</i>
<i>Σχήμα 5.2 – Απεικονίζονται τα τρισδιάστατα σχέδια που καλύπτουν το σασί περιφερειακά</i>	<i>56</i>
<i>Σχήμα 5.3 – Απεικόνιση της τρισδιάστατης εκτύπωσης με την κοιλότητα</i>	<i>56</i>
<i>Σχήμα 5.4 – Τελική μορφή της τρισδιάστατης εκτύπωσης</i>	<i>57</i>
<i>Σχήμα 5.5 – Τελική κατασκευή του οχήματος που μελετάμε</i>	<i>58</i>
<i>Σχήμα 5.6 – Συνδεσμολογία του ρομποτικού οχήματος</i>	<i>59</i>
<i>Σχήμα 5.7 – Graphical user interface που αναπτύχθηκε με την βιβλιοθήκη Tkinter της Python</i>	<i>65</i>
<i>Σχήμα 5.8 – Αντιστοιχίες κινήσεων στα πλήκτρα του πληκτρολογίου</i>	<i>65</i>
<i>Σχήμα 5.9 – Δημιουργία σύνδεσης Client-Server στον υπολογιστή</i>	<i>66</i>
<i>Σχήμα 5.10 – Συνάρτηση υπεύθυνη για το πλήκτρο ‘w’ μεριά Server (Υπολογιστής)</i>	<i>67</i>
<i>Σχήμα 5.10 – Συνάρτηση υπεύθυνη για το πλήκτρο ‘w’ μεριά Server (Υπολογιστής)</i>	<i>67</i>
<i>Σχήμα 5.11 – Συνάρτηση υπεύθυνη για το πλήκτρο ‘w’ μεριά Client (Οχήματος)</i>	<i>66</i>

## Ευρετήριο πινάκων

<i>Πίνακας 3.1- Γενική μορφή υπολογισμού γωνιών <math>\alpha</math>, <math>\beta</math> και <math>\gamma</math></i>	<i>34</i>
<i>Πίνακας 3.1- Τιμές των γωνιών στη διαμόρφωση του σχήματος 3.5</i>	<i>37</i>
<i>Πίνακας 4.1- Κοστολόγηση της διπλωματικής εργασίας</i>	<i>54</i>
<i>Πίνακας 5.1 – Κίνηση οχήματος ανάλογα με τις ταχύτητες του κάθε τροχού</i>	<i>62</i>
<i>Πίνακας 5.2 – Αποτύπωση των αποτελεσμάτων της προσομοίωσης στον πρόγραμμα MATLAB</i>	<i>63</i>
<i>Πίνακας 5.3 – Αποτύπωση των αποτελεσμάτων του πειραματικού σταδίου μέρος Α</i>	<i>69</i>
<i>Πίνακας 5.3 – Αποτύπωση των αποτελεσμάτων του πειραματικού σταδίου μέρος Β</i>	<i>70</i>

# Κεφάλαιο I. Περιγραφή του προβλήματος

## 1.1 Αυτόνομα Ρομπότ

Ρομποτική είναι η επιστήμη η οποία συνδυάζει αρχές από τις επιστήμες των μαθηματικών, της φυσικής, της πληροφορικής, της μηχανολογίας, της ηλεκτρολογίας, των τηλεπικοινωνιών. Εμπεριέχει επίσης την τεχνολογία συστημάτων αυτομάτου ελέγχου, την επεξεργασία ψηφιακών σημάτων, την υπολογιστική όραση, τα νευρωνικά δίκτυα, μηχανική μάθηση και την τεχνητή νοημοσύνη.

Ο κλάδος της ρομποτικής παρόλο που μελετάται για σχετικά μικρό διάστημα, έχει πολύ μεγάλη τεχνολογική ανάπτυξη και εξελίσσεται καθημερινά με γρήγορους ρυθμούς. Η εξέλιξη αυτή, σε συνδυασμό με την καθημερινή μελέτη, έχουν οδηγήσει σε τεχνολογικά επιτεύγματα τα οποία πριν είκοσι χρόνια θα μπορούσαν να είχαν θεωρηθεί σενάρια επιστημονικής φαντασίας.

Οι εφαρμογές της ρομποτικής έχουν συμβάλει σημαντικά στην ζωή μας ακόμα και αν δεν μπορούμε να το αντιληφθούμε. Τα ρομπότ έχουν μεγάλο φάσμα εφαρμογών και μία από τις πιο διαδεδομένες χρήσεις τους, είναι οι ρομποτικοί βραχίονες τους οποίους μπορεί κανείς να τους συναντήσει σχεδόν σε όλα τα μεγάλα εργοστάσιο και τις γραμμές παραγωγής, δίνοντας την δυνατότητα να λειτουργήσουν όλο το εικοσιτετράωρο πραγματοποιώντας γρήγορα και με ακρίβεια την εργασία που τους έχει ανατεθεί.

Η συνεχής εξέλιξη του κλάδου έχει δώσει την δυνατότητα να κατασκευάσουμε ρομποτικά συστήματα τα οποία είναι πλήρως αυτόνομα. Η ικανότητα ενός συστήματος να λαμβάνει τις δικές του αποφάσεις ανάλογα με τα εξωτερικά ερεθίσματα που δέχεται από το περιβάλλον στο οποίο βρίσκεται, ονομάζεται αυτονομία και παρέχει σε αυτά τα ρομπότ την δυνατότητα να λειτουργούν και να πλοηγούνται χωρίς άμεση ανθρώπινη παρέμβαση. Είναι εξοπλισμένα με διάφορους αισθητήρες, μονάδες επεξεργασίας και ενεργοποιητές. Ο αριθμός και ο τύπος των αισθητήρων είναι άρρηκτα συνδεδεμένος με την εργασία την οποία θα εκτελούν. Βασικά στοιχεία και χαρακτηριστικά των αυτόνομων κινητών ρομπότ περιλαμβάνουν:

- **Αισθητήρες:** Τα αυτόνομα κινητά ρομπότ είναι εξοπλισμένα με αισθητήρες για τη συλλογή πληροφοριών σχετικά με το περιβάλλον τους. Αυτοί οι αισθητήρες μπορούν να περιλαμβάνουν κάμερες, lidar (Light Detection and Ranging), αισθητήρες υπερήχων, ραντάρ και πολλά άλλα. Αυτοί οι αισθητήρες παρέχουν δεδομένα σχετικά με τη θέση, τον προσανατολισμό, τα εμπόδια και άλλους περιβαλλοντικούς παράγοντες.
- **Μονάδες επεξεργασίας:** Τα σύγχρονα αυτόνομα ρομπότ χρησιμοποιούν ισχυρούς υπολογιστές ή μικροελεγκτές για την επεξεργασία δεδομένων και τη λήψη αποφάσεων σε πραγματικό

χρόνο. Αυτές οι μονάδες επεξεργασίας χρησιμοποιούν αλγόριθμους και τεχνητή νοημοσύνη για να ερμηνεύσουν πληροφορίες αισθητήρων, να σχεδιάσουν τροχιές, να αποφύγουν εμπόδια και να καθορίσουν την καλύτερη πορεία δράσης για την επίτευξη των στόχων τους.

- **Χαρτογράφηση και εντοπισμός:** Τα αυτόνομα ρομπότ δημιουργούν χάρτες του περιβάλλοντός τους και τους χρησιμοποιούν για πλοήγηση. Έχουν την δυνατότητα να δημιουργούν χάρτες ενώ ταυτόχρονα παρακολουθούν τη δική τους θέση μέσα στο περιβάλλον. Αυτές οι πληροφορίες είναι ζωτικής σημασίας για την ασφαλή και την ασφαλή πλοήγηση.
- **Σχεδιασμός διαδρομής και πλοήγηση:** Εφόσον έχει ολοκληρωθεί η χαρτογράφηση, το ρομπότ μπορεί να σχεδιάσει βέλτιστες διαδρομές για πλοήγηση από το ένα σημείο στο άλλο, αποφεύγοντας παράλληλα εμπόδια. Οι αλγόριθμοι σχεδιασμού διαδρομής λαμβάνουν υπόψη παράγοντες όπως εμπόδια, έδαφος και περιορισμούς για να βρουν την καλύτερη διαδρομή.
- **Αποφυγή εμποδίων:** Τα αυτόνομα κινητά ρομπότ έχουν σχεδιαστεί για να πλοηγούνται γύρω από εμπόδια για να εξασφαλίζουν ασφαλή και αποτελεσματική κίνηση. Χρησιμοποιούν τα δεδομένα αισθητήρων και τους αλγόριθμους σχεδιασμού διαδρομής για να εντοπίσουν πιθανά εμπόδια και να προσαρμόσουν ανάλογα τις τροχιές τους.

Τέτοιου τύπου ρομπότ συναντάμε πλέον αρκετά συχνά με την μορφή μη επανδρωμένου οχήματος παραδείγματος χάρι: στα σπίτια μας με την μορφή της ηλεκτρικής σκούπας, πειραματικά οχήματα στους δρόμους μας, drones στον αέρα, υποβρύχια οχήματα στις θάλασσες, οχήματα διαστημικής μελέτης όπως αυτά που κινούνται στην επιφάνεια του πλανήτη Άρη και πολλά άλλα.

## 1.2 Περιγραφή του προβλήματος

Η παρούσα εργασία έχει ως σκοπό την μελέτη και κατασκευή ενός ρομποτικού οχήματος το οποίο θα έχει την δυνατότητα να κινείται πανκατευθυντικά, δηλαδή προς όλες τις κατευθύνσεις, χωρίς να παρουσιάζει τους κλασικούς περιορισμούς κίνησης που έχουν τα περισσότερα οχήματα με 4 τροχούς. Για τον σκοπό αυτόν θα χρειαστεί να ασχοληθούμε και να μελετήσουμε τρία διαφορετικά προβλήματα.

- Η επίλυση των εξισώσεων κίνησης.
- Η σχεδίαση και η κατασκευή του οχήματος.
- Το σύστημα ελέγχου και ο προγραμματισμός του οχήματος

Η επίλυση των εξισώσεων της κίνησης είναι το πρώτο με το οποίο θα ασχοληθούμε κατά την εκπόνηση της εργασίας. Συγκεκριμένα, θα χρειαστεί να γίνει επίλυση του ευθέως κινηματικού προβλήματος καθώς και του αντίστροφου κινηματικού προβλήματος.

Η μελέτη και η υλοποίηση του κατασκευαστικού μέρους της εργασίας απαιτεί τη δημιουργία μιας ρομποτικής πλατφόρμας, η οποία θα έχει τα χαρακτηριστικά που αναφέραμε παραπάνω, η οποία πέρα

από την πανκατευθυντική κίνηση θα έχει ενσωματωμένο και όλο το ηλεκτρονικό κύκλωμα και την μπαταρία.

Στη συνέχεια, θα υλοποιηθεί το σύστημα ελέγχου του οχήματος με γνώσεις που αποκτήθηκαν κατά την φοίτηση στο μεταπτυχιακό πρόγραμμα στη ρομποτική, καθώς θα χρειαστεί να επιλεχθούν τα σωστά ηλεκτρονικά εξαρτήματα και ένας ελεγκτής ο οποίος θα μπορεί να λειτουργήσει με τις προϋποθέσεις που απαιτούνται για τον έλεγχο του συγκεκριμένου οχήματος. Μέρος του ελέγχου φυσικά είναι και ο προγραμματισμός του ελεγκτή, ο οποίος θα δώσει την δυνατότητα να ελέγχεται το όχημα απομακρυσμένα, μέσω τοπικού δικτύου Wi-Fi. Η γλώσσα προγραμματισμού Python παρέχει τα εργαλεία που χρειάζονται για την δημιουργία ενός γραφικού περιβάλλοντος (GUI-Graphical user interface) μέσω του οποίου θα ελέγχεται το όχημα με τρόπο που θα είναι εύκολο για τον χρήστη να τον χρησιμοποιεί.

### 1.3 Στόχοι και αντικείμενα έρευνας

Για να μπορέσει να κινηθεί το όχημα με τον τρόπο που περιγράψαμε στην παραπάνω ενότητα θα χρειαστεί να είναι εξοπλισμένο με ειδικούς τροχούς, τους λεγόμενους «Σουηδικούς Τροχούς». Ο σουηδικός τροχός είναι ένας εξειδικευμένος τύπος τροχού ο οποίος χρησιμοποιείται σε διάφορες εφαρμογές. Πρόκειται για έναν τροχό ο οποίος, στην πιο συνηθισμένη του σχεδιαστική μορφή, αποτελείται από μία σειρά με ειδικούς κυλίνδρους που είναι τοποθετημένοι σε μια διάταξη 45 μοιρών ως προς τον άξονα περιστροφής του τροχού. Αυτός ο πρωτότυπος σχεδιασμός επιτρέπει στα συστήματα να κινούνται μπρος, πίσω, δεξιά, αριστερά, ακόμα και διαγώνια προσφέροντας μεγάλη ευελιξία.

Ο σχεδιασμός του ρομποτικού οχήματος πραγματοποιείται στον ηλεκτρονικό υπολογιστή χρησιμοποιώντας πρόγραμμα τρισδιάστατης σχεδίασης. Στη συνέχεια, αφού ολοκληρωθεί το σχεδιαστικό κομμάτι όλα τα εξαρτήματα εκτυπώνονται με την βοήθεια ενός τρισδιάστατου εκτυπωτή, ώστε να πάρει μορφή το όχημα μας.

Ένα από το πιο σημαντικό μέρος της διπλωματικής αυτής θα είναι η επιλογή των απαραίτητων ηλεκτρονικών εξαρτημάτων, ώστε να λειτουργήσει ομαλά το όχημα δίχως να εμφανιστούν προβλήματα στην κίνησή του, καθώς και για την αποφυγή πυρκαγιάς, μιας και θα χρειαστεί το όχημα να φέρει μπαταρία τύπου **Li-Po** η οποία μπορεί να είναι επικίνδυνη αν χρησιμοποιηθεί με λάθος τρόπο.

Για τον έλεγχο της κατασκευής μας θα χρειαστούν γνώσεις της γλώσσας προγραμματισμού **Python** η οποία θα μας δώσει την δυνατότητα να ελέγχουμε το ρομπότ απομακρυσμένα, μιας και θα υπάρχει ασύρματη σύνδεση μέσω τοπικού δικτύου Wi-Fi. Ο ελεγκτής του οχήματος είναι ένα Raspberry Pi το οποίο θα το προγραμματίσουμε ώστε να κινεί το όχημα με τις εντολές που θα δίνει ο χρήστης στο ρομπότ. Αυτό συνεπάγεται ότι η ο έλεγχος του οχήματος θα πραγματοποιηθεί σε 2 μέρη:

1. Θα δημιουργήσουμε εφαρμογή η οποία βρίσκεται στον σταθερό μας υπολογιστή και δουλεύει ως server και θα στέλνει τις εντολές στο όχημα.
2. Το **Raspberry Pi** θα είναι προγραμματισμένο έτσι ώστε να συνδέεται στον **Server** που δημιουργήσαμε με την μορφή **Client** και να δέχεται τις εντολές που στέλνει η εφαρμογή και στη συνέχεια να τις εκτελεί με τέτοιο τρόπο, ώστε να κινείται σωστά το όχημα.

## 1.4 Συνεισφορά της διπλωματικής εργασίας

Η συγκεκριμένη εργασία αρχικά επιχειρεί να εξετάσει τον σουηδικό τροχό για τα πλεονεκτήματα και τα μειονεκτήματα που εμφανίζει και τις δυνατότητες που παρέχει σε ένα όχημα. Επιπλέον, η παρούσα διπλωματική εργασία είναι μια προσπάθεια μελέτης, κατασκευής και προγραμματισμού ενός πανκατευθυντικού οχήματος, με τις γνώσεις που αποκτήθηκαν κατά την διάρκεια παρακολούθησης του μεταπτυχιακού προγράμματος σπουδών στον κλάδο της ρομποτικής.

Έτσι, θα διαπιστωθεί αν το μοντέλο που θα μελετήσουμε και θα υλοποιήσουμε μπορεί να χρησιμοποιηθεί ως ένα πλήρες εκπαιδευτικό εργαλείο για τη μελέτη πανκατευθυντικού οχήματος. Επίσης θα διερευνηθεί σε τι είδους αποστολές μπορεί να χρησιμοποιηθεί ένα τέτοιο όχημα.

## 1.5 Περίγραμμα της μεταπτυχιακής εργασίας

Στο **κεφάλαιο 1** γίνεται η εισαγωγή στο θέμα της μεταπτυχιακής εργασίας και πραγματοποιείται μια σύντομη περιγραφή του πανκατευθυντικού οχήματος που θα εξετάσουμε. Δίνονται επίσης συνοπτικά το περίγραμμα του προβλήματος, οι στόχοι και τα αντικείμενα έρευνας, καθώς και η συνεισφορά της παρούσας εργασίας.

Στο **Κεφάλαιο 2** αναφέρονται μερικές σημαντικές πληροφορίες για τα τροχήλατα ρομποτικά συστήματα και για τους τύπους των τροχών. Στη συνέχεια παρουσιάζονται μερικές προσπάθειες άλλων φοιτητών για την μελέτη και κατασκευή παρόμοιου οχήματος με αυτό που θα μελετήσουμε (κατάσταση της τέχνης).

Στο **Κεφάλαιο 3** παρουσιάζεται η ολοκληρωμένη θεωρητική μελέτη για την κινηματική του οχήματος με 4 σουηδικούς τροχούς, αναλύοντας το ευθύ και το αντίστροφο κινηματικό πρόβλημα και τις σχέσεις μεταξύ του οχήματος και του χώρου στον οποίο κινείται.

Στο **Κεφάλαιο 4** έχουμε μια λεπτομερή περιγραφή της μεθόδου που ακολουθήθηκε για την κατασκευή του οχήματος καθώς και όλων των υλικών που χρειάστηκε για να ολοκληρωθεί. Επίσης γίνεται ανάλυση των απαραίτητων γνώσεων που χρειάστηκαν για την κατασκευή.

Στο **Κεφάλαιο 5** αναπτύσσεται η διαδικασία της συναρμολόγησης του ρομποτικού οχήματος βήμα-βήμα. Επίσης περιγράφεται ο τρόπος με τον οποίον πραγματοποιήθηκε η διαδικασία των

πειραμάτων ξεκινώντας στην αρχή με ένα θεωρητικό κομμάτι το οποίο πραγματοποιήθηκε στο Matlab. Στη συνέχεια, εφαρμόσαμε όλα τα αποτελέσματα που εξαγάγαμε από το Matlab και τα περάσαμε στο ρομποτικό όχημα για να παρατηρήσουμε αν αυτά τα δύο θα έχουν το ίδιο τελικό αποτέλεσμα.

Στο **Κεφάλαιο 6** γίνεται περιγραφή των συνολικών αποτελεσμάτων και των συμπερασμάτων που προέκυψαν από όλη την εργασία. Τέλος επισημαίνονται κάποιες μελλοντικές βελτιώσεις οι οποίες θα μπορούσαν να εφαρμοστούν για ακόμα καλύτερα αποτελέσματα.



# Κεφάλαιο II. Κατάσταση της τέχνης

## 2.1 Τροχήλατα ρομποτικά συστήματα

Τα περισσότερα ρομποτικά συστήματα που έχουν την δυνατότητα να κινούνται στον χώρο, συνήθως φέρουν ρόδες. Εφόσον μας το επιτρέπουν οι συνθήκες του εδάφους, είναι προτιμότερο να επιλέξουμε τροχούς παρά αρθρωτά μέρη για την κίνηση του ρομπότ. Ο τροχός είναι μια ανθρώπινη εφεύρεση που επιτυγχάνει εξαιρετικά υψηλή απόδοση σε επίπεδο και λείο έδαφος.

Επιπλέον, δεδομένου ότι τα τροχοφόρα ρομπότ κατασκευάζονται συνήθως με όλους τους τροχούς σε συνεχή επαφή με το έδαφος, η ισορροπία δεν είναι συνήθως ένα ερευνητικό ζήτημα, καθώς επιτυγχάνεται όταν το ρομπότ διαθέτει από τρεις τροχούς και άνω. Ωστόσο, σε κάποιες περιπτώσεις δύο τροχοί μπορούν επίσης να κάνουν ένα ρομπότ σταθερό. Όταν επιθυμούμε να χρησιμοποιήσουμε περισσότερους από τρεις τροχούς, ένα σύστημα ανάρτησης βοηθάει στο να επιτρέψει σε όλους τους τροχούς να παραμείνουν σε επαφή με το έδαφος, όταν το ρομπότ ταξιδεύει σε ανώμαλο έδαφος.

Όπως θα δούμε, υπάρχουν διαφορετικές διατάξεις τροχών που μπορούν να χρησιμοποιηθούν για την κινητικότητα κινητών ρομπότ. Υπάρχουν διαφορετικοί τύποι τροχών, ο καθένας με μοναδικά πλεονεκτήματα και μειονεκτήματα.

Τα κινητά συστήματα είναι συστήματα που δεν είναι μόνιμα στερεωμένα, μπορούν δηλαδή να μετακινούνται σε ένα συγκεκριμένο περιβάλλον και σε μια συγκεκριμένη περιοχή. Κατηγοριοποιούνται σε διάφορες κατηγορίες οι οποίες εξαρτώνται από το περιβάλλον που κινούνται [1]:

**Εδαφικά συστήματα κίνησης:** διάφοροι τύποι κινητών συστημάτων όπως κινητά οχήματα με τροχούς ή ρομπότ με πόδια (που μοιάζουν με ανθρώπους ή ζώα) και κινητά ρομπότ. Τα ρομπότ με πόδια μιμούνται την κίνηση διάφορων θηλαστικών. Τα περισσότερα κινητά συστήματα στο έδαφος που έχουν τροχούς ανήκουν σε αυτήν την ομάδα. Τις περισσότερες φορές σχεδιάζονται ώστε να είναι αυτόνομα ώστε σε κάποιο βαθμό δεν χρειάζεται να τα ελέγχει ο άνθρωπος.

**Εναέρια συστήματα κίνησης:** είναι συστήματα τα οποία κινούνται στον αέρα και δεν έχουν σταθερή βάση. Σε αυτήν την κατηγορία ανήκουν διάφορα αεροσκάφη, ελικόπτερα, μη επανδρωμένα αεροσκάφη, πύραυλοι, ιπτάμενα συστήματα που μιμούνται ιπτάμενα ζώα.

**Υδάτινα και υποβρύχια συστήματα κίνησης:** Εδώ υπάρχουν διάφοροι τύποι σκαφών, όπως βάρκες, υποβρύχια, υποβρύχια οχήματα κ.λπ.

Τα εδαφικά συστήματα κίνησης θεωρούνται αυτόνομα όταν είναι ικανά να λειτουργούν χωρίς ανθρώπινη επίβλεψη και να εκτελούν ανεξάρτητη κίνηση στο περιβάλλον τους. Η αυτονομία πρέπει να διασφαλίζεται από τα ακόλουθα:

- **Ενεργειακή αυτονομία:** το ρομπότ πρέπει να διαθέτει μια πηγή ενέργειας.
- **Αυτονομία λήψης αποφάσεων:** το ρομπότ πρέπει να έχει τη δυνατότητα να λαμβάνει ορισμένες αποφάσεις και να προβαίνει στις κατάλληλες ενέργειες.

Στην πραγματικότητα, αυτό σημαίνει ότι το κινητό σύστημα είναι υπεύθυνο για την παροχή εντολών με βάση το βαθμό αυτονομίας που διαθέτει το σύστημα. Με βάση το επίπεδο αυτονομίας του ρομπότ, θα πρέπει να οριστούν τιμές από τον ελεγκτή του ρομπότ ή από τον χειριστή, για τις ακόλουθες τυπικές παραμέτρους:

- **Οι επιθυμητές ταχύτητες των τροχών.**

Το ρομπότ λαμβάνει εντολές ως προς τις επιθυμητές ταχύτητες των τροχών. Επιπλέον, χρησιμοποιούνται βασικοί αλγόριθμοι ελέγχου με κατάλληλους αισθητήρες (συνήθως περιστροφικούς κωδικοποιητές) για να διασφαλιστεί ότι η κίνηση είναι η προβλεπόμενη.

- **Οι επιθυμητές διαμήκεις και γωνιακές ταχύτητες του ρομπότ.**

Ο ελεγκτής του ρομπότ γνωρίζει το κινηματικό του μοντέλο, το οποίο του επιτρέπει να υπολογίζει τις σωστές περιστροφικές ταχύτητες για τους τροχούς, ώστε να επιτυγχάνονται οι επιθυμητές ταχύτητες του οχήματος, στο αδρανειακό σύστημα αναφοράς. Είναι ικανό να προσδιορίσει και να ακολουθήσει μια επιθυμητή διαδρομή ή τροχιά καθώς και να γνωρίζει τη θέση που παίρνει μέσα στο περιβάλλον του. Για τον εντοπισμό χρησιμοποιούνται διάφοροι αισθητήρες οι οποίοι είτε είναι προσαρτημένοι σε διαφορετικά μέρη του σώματος του ρομπότ είτε στο περιβάλλον. Αν για τον οποιοδήποτε λόγο παρουσιαστούν στο σύστημα ανακριβή δεδομένα αισθητήρων, ολίσθηση τροχών, ανακριβή μοντέλα, καθυστερήσεις κ.λπ., τότε παρουσιάζονται διάφορα προβλήματα, με αποτέλεσμα το σύστημα να οδηγηθεί σε άλλη θέση από αυτήν που επιθυμούμε.

- **Επιθυμητή λειτουργία στο γνωστό περιβάλλον με πιθανά εμπόδια.**

Το ρομπότ καλείται να εκτελέσει μια σειρά από εργασίες στην προκαθορισμένη περιοχή του, αλλά στην πορεία του υπάρχουν κάποια στατικά ή φυσικά εμπόδια ή ακόμα μπορεί να εμφανιστεί ένα καινούργιο εμπόδιο το οποίο αρχικά δεν βρισκόταν στην πορεία του ρομπότ, όπως για παράδειγμα ένα εργαλείο που μπορεί να έπεσε. Σε αυτήν την περίπτωση, το ρομπότ θα πρέπει να είναι σε θέση να σχεδιάσει τη διαδρομή του και αν αυτά τα εμπόδια που αναφέραμε βρίσκονται στην πορεία του τότε να μπορεί να κάνει επανασχεδιασμό της πορείας, ώστε να εκτελέσει την αποστολή του, χωρίς πρόβλημα.

- **Επιθυμητή συμπεριφορά στο άγνωστο περιβάλλον.**
- Το ρομπότ θα πρέπει να γνωρίζει την θέση του στο χώρο, ώστε να μπορέσει να δουλέψει σωστά. Μια μέθοδος είναι το SLAM (ταυτόχρονος εντοπισμός και χαρτογράφηση). Είναι μια μέθοδος που χρησιμοποιείται για αυτόνομα οχήματα, η οποία επιτρέπει την κατασκευή ενός χάρτη και τον εντοπισμό του οχήματος μέσα σε αυτόν ταυτόχρονα. Οι αλγόριθμοι SLAM επιτρέπουν στο όχημα να χαρτογραφεί άγνωστα περιβάλλοντα. Οι μηχανικοί χρησιμοποιούν τις πληροφορίες του χάρτη για την εκτέλεση εργασιών όπως ο σχεδιασμός διαδρομής και η αποφυγή εμποδίων.

Τα σημαντικότερα μηχανικά και ηλεκτρονικά εξαρτήματα ενός κινητού αυτόνομου οχήματος, είναι τα εξής:

**Μηχανικά μέρη:** Κινούμενα μέρη όπως το σασί, το σώμα, οι τροχοί, οι ερπύστριες, τα πόδια κ.λπ.

**Ενεργοποιητές:** Ηλεκτρικοί κινητήρες όπως κινητήρες συνεχούς ρεύματος, βηματικοί κινητήρες, σερβοκινητήρες κ.λπ.

**Αισθητήρες:** Υπάρχει μεγάλη πληθώρα στην χρήση ηλεκτρονικών αισθητήρων όπως κωδικοποιητές περιστροφής, αισθητήρες προσέγγισης και απόστασης, GPS κλπ.

**Υπολογιστή ισχύ:** Μικροεπεξεργαστές και μικροελεγκτές, φορητοί υπολογιστές, ενσωματωμένα συστήματα κ.λπ.

**Πηγή ενέργειας:** Μπαταρίες, ηλιακοί συλλέκτες, κ.λπ.

**Ηλεκτρονικά εξαρτήματα:** Οδηγοί (drives) των επενεργητών, τηλεπικοινωνιακά συστήματα, μονάδες διαμερισμού ενέργειας κλπ.

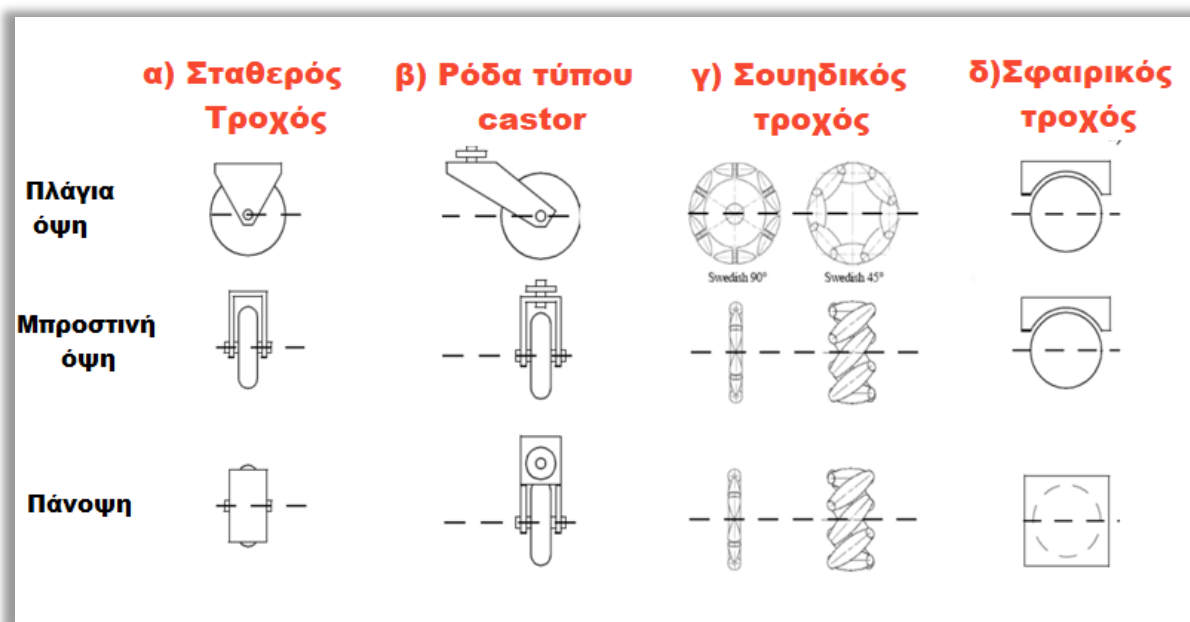
## 2.2 Ο τροχός

Ο τροχός χρησιμοποιείται πιο συχνά ως μηχανισμός μετάδοσης κίνησης στην κινητή ρομποτική. Έχει εντυπωσιακή απόδοση και μάλιστα με απλή μηχανική διαμόρφωση. Επιπλέον, το ζήτημα της ισορροπίας δεν αποτελεί συνήθως το επίκεντρο της έρευνας στα ρομπότ με τροχούς, διότι είναι τις περισσότερες φορές σχεδιασμένα έτσι ώστε όλοι οι τροχοί να έχουν καλή επαφή ανά πάσα στιγμή. Τα ρομπότ με δύο τροχούς μπορεί να εμφανίσουν σταθερότητα, αλλά τουλάχιστον τρεις τροχοί είναι απαραίτητοι για να έχουν μια σταθερή βάση. Όταν εμπλέκονται τρεις ή περισσότεροι τροχοί, είναι απαραίτητο ένα σύστημα ανάρτησης για να επιτρέπεται σε όλους τους τροχούς να διατηρούν επαφή με το έδαφος [2].

Τέσσερις κύριοι τύποι τροχών απεικονίζονται στο **Σχήμα 2.1**. Έχουν διαφορετική κινηματική συμπεριφορά, και έτσι η επιλογή του τύπου του τροχού έχει σημαντικό αντίκτυπο στη συνολική

κινηματική του κινητού ρομπότ. Ο κοινός τροχός και ο castor τροχός (σχέδιο a και σχέδιο b, σχήμα 2.1) έχουν έναν πρωτεύοντα άξονα περιστροφής, γι' αυτό και έχουν μια ιδιαίτερα κατευθυνόμενη φύση. Για να μπορέσει να ακολουθήσει μια διαφορετική διαδρομή, ο τροχός πρέπει πρώτα να στραφεί γύρω από τον κατακόρυφο άξονα. Η κύρια διάκριση μεταξύ των δύο τροχών είναι ότι ο τυπικός τροχός μπορεί να εκτελέσει αυτήν την κίνηση αλλαγής διεύθυνσης χωρίς παρενέργειες, ενώ αντίθετα, ο τροχός castor περιστρέφεται γύρω από έναν πρόσθετο άξονα, καθώς το κέντρο περιστροφής μετατοπίζεται από το σημείο επαφής του τροχού με το έδαφος.

Ο σουηδικός τροχός και ο σφαιρικός τροχός (σχέδιο c και σχέδιο d σχήμα 2.1), είναι και οι δύο παραλλαγές που περιορίζονται λιγότερο από τον συμβατικό τυπικό τροχό. Ο σουηδικός τροχός έχει την ίδια αντίσταση προς όλες τις κατευθύνσεις με έναν τυπικό τροχό, αλλά παρέχει μειωμένη αντίσταση και σε διαφορετική κατεύθυνση, μερικές φορές κάθετα στη συμβατική κατεύθυνση, όπως με το Swedish 90, και μερικές φορές σε ενδιάμεση γωνία, όπως με το Swedish 45. Οι μικροί κύλινδροι που είναι στερεωμένοι γύρω από την περιφέρεια του τροχού είναι παθητικοί και ο κύριος άξονας του τροχού είναι το μόνο ενεργό μέρος. Το κύριο πλεονέκτημα αυτής της σχεδίασης είναι πως, παρά το γεγονός ότι η περιστροφή του τροχού γίνεται μόνο γύρω από έναν πρωτεύοντα άξονα, ο τροχός είναι ικανός να κινείται κινηματικά με ελάχιστη τριβή σε πολλές πιθανές τροχιές, όχι μόνο προς τα εμπρός και προς τα πίσω [2].



**Σχήμα 2.1 – Απεικόνιση διαφόρων τύπων τροχών**

Ένας σφαιρικός τροχός είναι πραγματικά πανκατευθυντικός και συχνά σχεδιάζεται έτσι ώστε να μπορεί να οδηγηθεί ενεργά για να περιστρέφεται προς οποιαδήποτε κατεύθυνση. Ένας μηχανισμός που υλοποιεί αυτό το σφαιρικό σχέδιο μοιάζει με ένα ποντίκι υπολογιστή, ο μηχανισμός έχει κυλίνδρους που τροφοδοτούνται ενεργά και που στηρίζονται στην κορυφή της σφαίρας. Αυτοί οι κύλινδροι δέχονται μια περιστροφική δύναμη που ασκείται πάνω τους.

### 2.3 Ο σουηδικός τροχός

Οι σουηδικοί τροχοί υπάρχουν εδώ και πολύ καιρό και είναι αυτοί που μας έδωσαν την δυνατότητα για το πανκατευθυντικό σύστημα κίνησης που εξετάζουμε. Η σουηδική επιχείρηση Mecanum AB δημιούργησε και κατοχύρωσε με δίπλωμα ευρεσιτεχνίας αυτό το σχέδιο τροχού με τον Bengt Ilon το 1973 [3].



**Σχήμα 2.2** Αριστερά διακρίνουμε τον σουηδικό τροχό με κυλινδράκια υπό 45 μοίρες και στα δεξιά ο σουηδικός τροχός με κυλινδράκια υπό 90 μοίρες

Υπάρχουν διάφορες παραλλαγές του σουηδικού τροχού. Η επιφάνεια κάθε τροχού καλύπτεται από πολλαπλούς περιστρεφόμενους κυλίνδρους οι οποίοι έχουν την δυνατότητα να περιστραφούν γύρω από τον άξονα τους. Ο τροχός τροφοδοτείται από έναν κινητήρα, αλλά οι κύλινδροι του τροχού δεν έχουν κάποιο σύστημα περιστροφής. Αυτοί στηρίζονται από ρουλεμάν και έχουν τη δυνατότητα να περιστρέφονται ελεύθερα γύρω από τον κεντρικό τους άξονα. Η τοποθέτηση των κυλίνδρων γίνονται στις  $\pm 45^\circ$  και υπάρχουν τόσο αριστερές όσο και δεξιές εκδόσεις αυτού του τροχού (Σχήμα 2.2). Ο τροχός Mecanum με κυλινδράκια τοποθετημένα σε γωνία  $90^\circ$  δεν απαιτούν αριστερή ή δεξιά έκδοση (Σχήμα 2.2).

Ένα ρομπότ που χρησιμοποιεί σουηδικούς τροχούς για την κίνηση του, μπορεί να έχει τρεις είτε τέσσερις ρόδες. Τα οχήματα με τρεις τροχούς έχουν κυλίνδρους που είναι τοποθετημένοι υπό γωνία  $90^\circ$  μοιρών ως προς τον άξονα του τροχού. Σε αυτήν την διπλωματική θα εξετάσουμε την κίνηση του ρομπότ που βασίζεται σε τέσσερις σουηδικούς τροχούς. Για την κατασκευή ενός ρομπότ με τέσσερις τροχούς απαιτούνται δύο αριστερόστροφοι τροχοί (κύλινδροι σε  $+45^\circ$  ως προς τον άξονα του τροχού) και δύο δεξιόστροφοι τροχοί (κύλινδροι σε  $-45^\circ$  ως προς τον άξονα του τροχού).

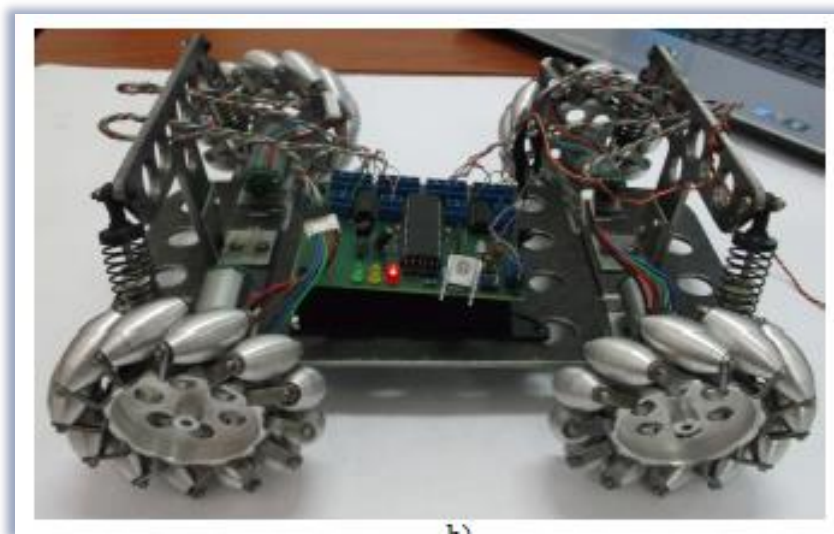
Αν και οι κύλινδροι περιστρέφονται ελεύθερα, αυτό δεν σημαίνει ότι το ρομπότ περιστρέφει τους τροχούς του και δεν κινείται. Αυτό θα συνέβαινε μόνο εάν οι κύλινδροι τοποθετούνταν παράλληλα με τον άξονα του τροχού. Ωστόσο, οι τροχοί έχουν τους κυλίνδρους τοποθετημένους υπό γωνία. Μελετώντας έναν τροχό μπορούμε να παρατηρήσουμε ότι η δύναμη που παράγεται από την περιστροφή του, δρα στο έδαφος μέσω του ενός κυλίνδρου που έχει επαφή με το έδαφος. Σε αυτόν τον κύλινδρο, η δύναμη μπορεί να χωριστεί σε ένα διάνυσμα παράλληλο προς τον άξονα του κυλίνδρου και ένα διάνυσμα κάθετο στον άξονα του κυλίνδρου. Η δύναμη κάθετη στον άξονα του κυλίνδρου θα έχει ως αποτέλεσμα μια μικρή περιστροφή του κυλίνδρου, ενώ η δύναμη παράλληλη προς τον άξονα του κυλίνδρου θα ασκήσει δύναμη στον τροχό και επομένως στο όχημα.

Δεδομένου ότι οι σουηδικοί τροχοί για να λειτουργήσουν πρέπει να συνεργαστούν π.χ. σε ένα συγκρότημα τεσσάρων τροχών, οι προκύπτουσες δυνάμεις των τροχών σε γωνία  $45^\circ$  από κάθε τροχό πρέπει να συνδυαστούν για να προσδιοριστεί η συνολική κίνηση του οχήματος. Εάν οι δύο τροχοί είναι οι μπροστινοί τροχοί του ρομπότ και οι δύο περιστρέφονται προς τα εμπρός, τότε κάθε ένας από τους δύο προκύπτοντες φορείς δύναμης  $45^\circ$  μπορεί να χωριστεί σε δύναμη προς τα εμπρός και πλάγια. Οι δύο δυνάμεις προς τα εμπρός αθροίζονται, ενώ οι δύο πλάγιες δυνάμεις (μία προς τα αριστερά και μία προς τα δεξιά) αλληλοεξουδετερώνονται.

## 2.4 Κατάσταση της τέχνης

Η κατασκευή ενός οχήματος με σουηδικό τροχό έχει κερδίσει το ενδιαφέρον σε αρκετές εταιρίες παγκοσμίως καθώς και σε πολλά τεχνολογικά ιδρύματα ή πανεπιστήμια. Κάνοντας μια σχετικά σύντομη αναζήτηση μπορούμε να δούμε πως έχουν γίνει πάρα πολλές προσπάθειες από φοιτητές και υποψήφιους διδάκτορες για μελέτη και βελτίωση των οχημάτων αυτών.

Μια από αυτές τις προσπάθειες είναι η έρευνα που έκανε το 2014 η ομάδα των **Mihai Olimpiu Tătar, Cătălin Popovici, Dan Mândru, Ioan Ardelean, Alin Pleșa** του ιδρύματος τεχνολογίας της **Cluj-Napoca** στην Ρουμανία [4]. Η μελέτη της ομάδας του συγκεκριμένου πανεπιστημίου ήταν να



**Σχήμα 2.3. Ολοκληρωμένο όχημα της ομάδας του ιδρύματος της Ρουμανίας [4]**

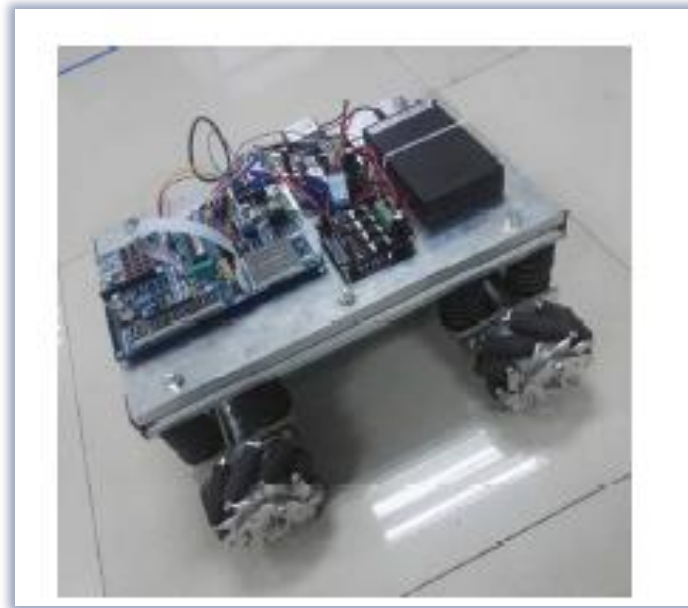
κατασκευάσουν ένα όχημα το οποίο διαθέτει τέσσερις σουηδικούς τροχούς των 45 μοιρών και στην συνέχεια να μελετήσουν την πραγματική κίνηση και την συμπεριφορά του οχήματος στον χώρο.

Η ομάδα του πανεπιστημίου κατάφερε να κατασκευάσει ένα όχημα το οποίο αποτελείται από ένα αλουμινένιο σασί και διαθέτει σύστημα αναρτήσεων σε όλες τις ρόδες. Το όχημα ελέγχεται από έναν μικροελεγκτή ATmega 644 και διαθέτει 2 ελεγκτές L293D οι οποίοι ελέγχουν από 2 ρόδες το καθένα. Στη συνέχεια ασχολήθηκε με το κινηματικό πρόβλημα, ώστε να μπορέσει να συγκρίνει τα αποτελέσματα που είχαν τόσο στο θεωρητικό μέρος όσο και στο πρακτικό. Στα **σχήμα 2.3** μπορούμε να δούμε την τελική κατασκευή το οχήματος.

Μια άλλη προσπάθεια είναι η έρευνα που έκανε το 2016 η ομάδα των φοιτητών **Qian Jia, Mulan Wang, Shuqing Liu, Jianjing Ge, Chen Gu** του ιδρύματος τεχνολογίας της **Nanjing** της Κίνας [5].

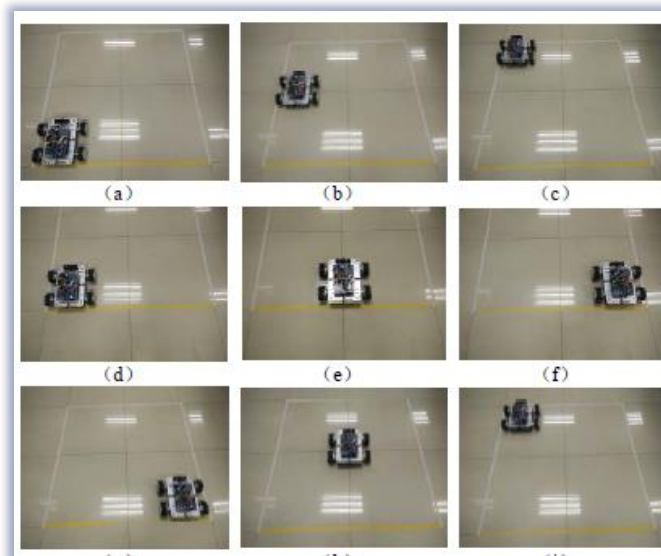
Η συγκεκριμένη ομάδα έχει ως σκοπό την μελέτη του μαθηματικού μοντέλου και στη συνέχεια, την κατασκευή ενός οχήματος με 4 σουηδικούς τροχούς των 45 μοιρών. Αφού ολοκλήρωσε τη μελέτη και τη λύση του κινηματικού προβλήματος η ομάδα κατασκεύασε ένα όχημα, το οποίο παρουσιάζει κάποιες ομοιότητες με το δικό μας. Το όχημα διαθέτει έναν ελεγκτή που δέχεται πληροφορίες και εντολές και ελέγχει τους επενεργητές του οχήματος, ώστε να κινηθεί. Διαθέτει αισθητήρες με υπέρυθρες ακτίνες, για την μέτρηση της απόστασης από διάφορα εμπόδια. Ο έλεγχος του οχήματος πραγματοποιείται από ένα τηλεχειριστήριο, το οποίο συνδέεται με το πρωτόκολλο BLUETOOTH





Σχήμα 2.4 – Τελικό όχημα της ομάδας του πανεπιστημίου της Κίνας.

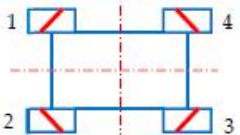



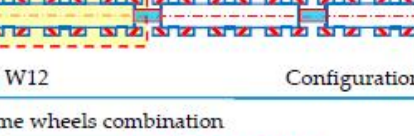



καθώς και επικοινωνία μέσω εφαρμογής κινητού. Στο **σχήμα 2.5** μπορούμε να δούμε κάποιες κινήσεις του οχήματος στον χώρο της μελέτης.



Σχήμα 2.5 – Κινήσεις στον χώρο μελέτης του οχήματος της ομάδας του ιδρύματος Nanjing



Το 2019 μια ομάδα φοιτητών από διαφορετικά πανεπιστήμια του κόσμου μελέτησαν την δημιουργία ενός οχήματος με οχτώ ή περισσότερους σουηδικούς τροχούς, με σκοπό να βοηθήσει στην μεταφορά διάφορων υπέρβαρων αντικειμένων. Η ομάδα αποτελείται από τους φοιτητές **Yunwang Li, Sumei Dai, Lala Zhao, Xucong Yan** και **Yong Shi** [6]. Οι συγκεκριμένη μελέτη είναι μεγάλη και αρκετά λεπτομερής. Ξεκινάει με την ανάλυση του οχήματος με τρεις τροχούς και στη συνέχεια προχωράει στην ανάλυση με τέσσερις τροχούς. Τέλος, η εργασία ολοκληρώνεται με την παρουσίαση διαφόρων μοντέλων οχημάτων με πολλούς σουηδικούς τροχούς, όπως απεικονίζονται στο **σχήμα 2.6**.

	Schematic Diagram of Wheel Configurations	Application Examples
(a)	 <p>Configuration W4</p>	
(b)	 <p>3 W4 Configuration W12</p>	
(c)	 <p>3 W12 Configuration W36</p>	
(d)	<p>Same wheels combination</p>  <p>6 W4 Configuration</p>	

Σχήμα 2.6 – Απεικονίσεις οχημάτων με περισσότερους από τέσσερις τροχούς

## 2.5 Συνεισφορά στην κατάσταση της τέχνης

Όπως μπορούμε να παρατηρήσουμε, έχουν γίνει πολλές μελέτες ανά τον κόσμο για τους σουηδικούς τροχούς και τα οχήματα που διαθέτουν τέτοιο είδος τροχού. Ένα μεγάλο ποσοστό αυτών των ερευνών και των μελετών αρκούνται στο θεωρητικό μέρος, επομένως η δική μας εργασία και μελέτη θα μας προσφέρει μια πιο ολοκληρωμένη εικόνα για τα συγκεκριμένα οχήματα, καθώς με την βοήθεια της κατασκευής που θα κάνουμε θα μπορούμε να μελετήσουμε σε πραγματικό χρόνο την κίνηση του οχήματος στις εντολές που λαμβάνει. Επίσης, λόγω της ύπαρξης τρισδιάστατου εκτυπωτή και των γνώσεων σχεδιασμού μπορούμε να δημιουργήσουμε ένα όχημα το οποίο θα μπορεί να επαναληφθεί από κάθε ενδιαφερόμενο και το οποίο θα περιέχει και πλαστικές θήκες, ώστε να μην

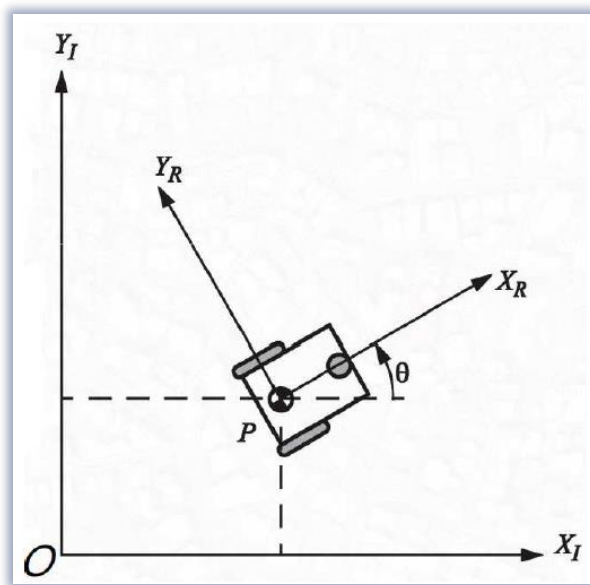
έχουμε εκτεθειμένα καλώδια και εκτεθειμένες πλακέτες. Εξίσου σημαντική είναι η χρήση της γλώσσας προγραμματισμού Python με την οποία θα κατασκευάσουμε από την αρχή ένα πλαίσιο απομακρυσμένου ελέγχου του οχήματος, με βάση το Raspberry Pi και δεν θα βασιστούμε σε κάποιο έτοιμο τηλεχειριστήριο. Το όχημα θα μπορεί να χρησιμοποιηθεί για περαιτέρω μελέτη, προσθέτοντας αισθητήρες.

# Κεφάλαιο III. Η κινηματική του οχήματος

## 3.1 Γενικοί ορισμοί για την κίνηση των τροχήλατων ρομπότ.

Στην παράγραφο αυτή θα δώσουμε κάποιους γενικούς ορισμούς, για την κίνηση των τροχήλατων ρομπότ. Αρχικά μοντελοποιούμε το ρομπότ ως άκαμπτο σώμα σε τροχούς, που λειτουργούν σε οριζόντιο επίπεδο. Για να μπορέσουμε να καθορίσουμε τη θέση του ρομπότ στο επίπεδο, έχουμε ένα γενικό πλαίσιο αναφοράς  $\mathbf{I}$ , του οποίου οι άξονες  $X_I$  και  $Y_I$  ορίζουν μια αυθαίρετη αδρανειακή βάση στο επίπεδο. Για να μπορέσουμε να καθορίσουμε τη θέση του ρομπότ στο επίπεδο δημιουργούμε μια σχέση μεταξύ του γενικού πλαισίου αναφοράς  $\mathbf{I}$  και του τοπικού πλαισίου αναφοράς  $\mathbf{R}$  του ρομπότ.

Το σημείο  $\mathbf{P}$  του πλαισίου του ρομπότ επιλέγεται ως σημείο αναφοράς της θέσης του ρομπότ και δημιουργούμε δυο άξονες  $X_R$  και  $Y_R$  με αρχή των αξόνων το σημείο  $\mathbf{P}$ . Η θέση του  $\mathbf{P}$  στο γενικό πλαίσιο αναφοράς  $\mathbf{I}$  καθορίζεται από τις συντεταγμένες  $X$  και  $Y$ , καθώς και τη γωνιακή διαφορά μεταξύ του γενικού  $\mathbf{I}$  και του τοπικού συστήματος αναφοράς, που συμβολίζεται με  $\theta$ .



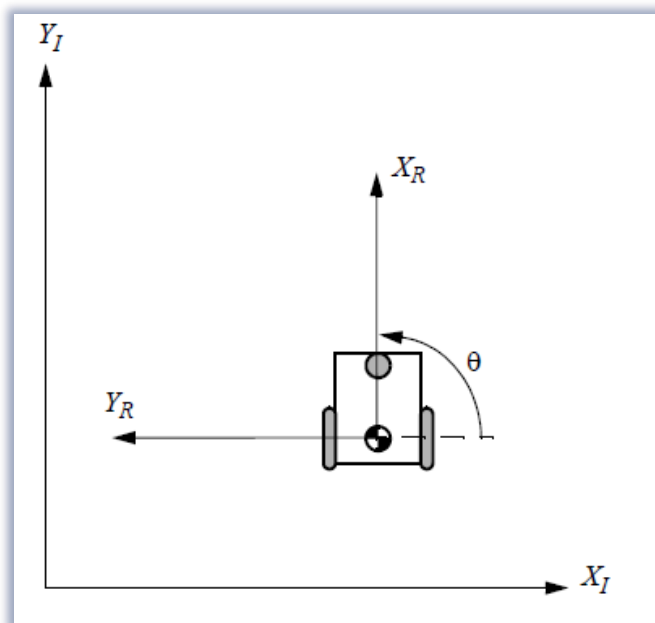
Σχήμα 3.1 – Απεικόνιση του οχήματος στο πλαίσιο αναφοράς [7]

Τα στοιχεία αυτά μας περιγράφουν την θέση του ρομπότ ως διάνυσμα:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = [x \ y \ \theta]^T$$

Σχέση 3.1

Η συσχέτιση της κίνησης του ρομπότ στο σύστημα του ρομπότ  $\mathbf{R}$  με το γενικό πλαίσιο αναφοράς  $\mathbf{I}$  περιγράφει την κίνηση του ρομπότ στο  $\mathbf{I}$ . Συνεπώς η συσχέτιση αυτή θα είναι κάθε φορά συνάρτηση της τρέχουσας θέσης του ρομπότ.



Σχήμα 3.2 – Απεικόνιση οχήματος που περιστράφηκε κατά γωνία  $\theta$  [7]

Αυτή η αντιστοίχιση ανάμεσα στο  $\mathbf{R}$  και στο  $\mathbf{I}$  επιτυγχάνεται χρησιμοποιώντας τον παρακάτω πίνακα περιστροφής:

Πίνακας περιστροφής:

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

Έχοντας τον πίνακα περιστροφής μπορούμε να μετατρέψουμε και να υπολογίσουμε την κίνηση στο γενικό πλαίσιο αναφοράς  $\mathbf{I} (\mathbf{X}_I, \mathbf{Y}_I)$  σε κίνηση στο πλαίσιο αναφοράς του ρομπότ  $\mathbf{R} (\mathbf{X}_R, \mathbf{Y}_R)$  και δίνεται από την σχέση:

$$\dot{\xi}_R = \mathbf{R}(\theta)\dot{\xi}_I \quad \text{Σχέση 3.2}$$

Επομένως η ταχύτητα του ρομπότ στο τοπικό σύστημα αναφοράς  $\mathbf{R}$ , συνδέεται με την συνολική ταχύτητα που έχει το ρομπότ στο γενικό πλαίσιο αναφοράς  $\mathbf{I}$  με την σχέση:

$$\dot{\xi}_I = \mathbf{R}(\theta)^{-1} \dot{\xi}_R \quad \text{Σχέση 3.3}$$

όπου το  $\mathbf{R}(\theta)^{-1}$  είναι ο αντίστροφος πίνακας του πίνακα περιστροφής  $\mathbf{R}(\theta)$  που αναλύσαμε παραπάνω και θα ισούται με:

**Αντίστροφος πίνακας περιστροφής:**

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

Άρα έχοντας τον πίνακα στροφής μπορούμε εύκολα να υπολογίσουμε είτε την ταχύτητα στο γενικό πλαίσιο αναφοράς  $\xi_I$ , είτε την ταχύτητα του τοπικού πλαισίου αναφοράς  $\xi_R$  γνωρίζοντας μία από τις δύο.

### Ευθύ κινηματικό πρόβλημα.

Το ευθύ κινηματικό πρόβλημα μελετάει την κίνηση του οχήματος όταν είναι γνωστές οι γωνιακές ταχύτητες των τροχών για όλες τις χρονικές στιγμές που ορίσαμε στον πίνακα δειγματοληψίας. Με βάση τις γωνιακές ταχύτητες περιστροφής των τροχών  $\dot{\phi}$ , μπορούμε να υπολογίσουμε τις ταχύτητες του οχήματος στο πλαίσιο του ρομπότ  $\mathbf{R}$  χρησιμοποιώντας τις κινηματικές εξισώσεις:

$$\dot{\xi}_R = f(\dot{\phi}) \quad \text{Σχέση 3.4}$$

Στη συνέχεια, μπορούμε να υπολογίσουμε τις ταχύτητες στο αδρανειακό σύστημα αναφοράς **I**:

$$\xi_I = R(\theta)^{-1} \xi_R$$

**Σχέση 3.3**

Ξέροντας τις ταχύτητες  $\xi_I$  στο **I**, μπορούμε να υπολογίσουμε τις διαδοχικές θέσεις  $\xi_I$ , δηλαδή την τροχιά του ρομπότ, για να βρούμε τον πίνακα  $\xi_I$  πρέπει να υπολογίσουμε κάθε φορά τις τιμές του  $\theta$  για κάθε χρονική στιγμή ακολουθώντας τα παρακάτω βήματα:

$$\Delta t = (\text{Νέα χρονική στιγμή}) - (\text{Προηγούμενη χρονική στιγμή})$$

Οι τιμές του  $\theta$  σε κάθε βήμα προκύπτουν από την σχέση:

$$\Delta \theta = \dot{\theta}_n \Delta t$$

Άρα η καινούργια γωνία  $\theta$  θα ισούται με την προηγούμενη γωνία  $\theta$  συν την καινούργια:

$$\theta_n = \theta_{n-1} + \Delta \theta$$

Επομένως γνωρίζοντας αρχικά το  $\theta$  για κάθε χρονική στιγμή προχωράμε στον πολλαπλασιασμό του αντίστροφου πίνακα περιστροφής  $R(\theta)^{-1}$  με το  $\xi_R$  και με αυτόν τον τρόπο παίρνουμε το  $\xi_I$  για κάθε χρονική στιγμή (**Σχέση 3.3**).

Γνωρίζοντας της ταχύτητες του οχήματος στο αδρανειακό πεδίο **I** για κάθε χρονική στιγμή μπορούμε να υπολογίσουμε την θέση του, ώστε να αποτυπώσουμε την κίνηση του σχεδιάζοντας όλα τα σημεία που έχουμε βρει.

Με τον ίδιο τρόπο που υπολογίσαμε τις τιμές του  $\theta$  για την κάθε χρονική στιγμή μπορούμε να βρούμε και τις συντεταγμένες  $x$  και  $y$ . Για τον υπολογισμό του κάθε σημείου  $x_n$  πρέπει να αφαιρέσουμε κάθε φορά την προηγούμενη χρονική στιγμή για να βρούμε το χρονικό διάστημα  $\Delta t$  και στη συνέχεια να προσθέσουμε στο προηγούμενο  $x$  το καινούργιο  $\Delta x$  που υπολογίσαμε στο καθολικό πλαίσιο αναφοράς.

Δηλαδή:

$$\Delta x = \dot{x}_n \Delta t$$

Άρα το καινούργιο  $x_n$  θα ισούται με:

$$x_n = x_{n-1} + \Delta x$$

Παρόμοια βρίσκουμε και τις τιμές του  $y_n$ :

$$\Delta y = \dot{y}_n \Delta t$$

Άρα το καινούργιο  $\mathbf{y}_n$  θα ισούται με:

$$\mathbf{y}_n = \mathbf{y}_{n-1} + \Delta\mathbf{y}$$

Πλέον είμαστε σε θέση να κάνουμε μια προσομοίωση της κίνησης του οχήματος, εφόσον έχουμε όλα τα στοιχεία του  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\theta$  για κάθε χρονική στιγμή που υπολογίσαμε λύνοντας το ευθύ κινηματικό πρόβλημα.

### Αντίστροφο κινηματικό πρόβλημα.

Το αντίστροφο κινηματικό πρόβλημα, όπως το λέει και η ονομασία του, μας βοηθάει να βρούμε τις περιστροφικές ταχύτητες του κάθε τροχού σε συγκεκριμένες χρονικές στιγμές. Το αντίστροφο κινηματικό πρόβλημα ορίζεται από τις κινηματικές εξισώσεις ως εξής:

$$\dot{\phi} = f'(\xi_I) \quad \text{Σχέση 3.5}$$

Για να μπορέσουμε να λύσουμε ως προς το αντίστροφο κινηματικό πρόβλημα, θα χρειαστεί να γνωρίζουμε τον πίνακα  $\xi_I$  με τις ταχύτητες του ρομπότ στο γενικό πλαίσιο αναφοράς.

Για τον υπολογισμό του  $\xi_I$  για την κάθε χρονική στιγμή θα χρησιμοποιούμε την **σχέση 3.6**:

$$\xi_I = \begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{bmatrix} = \begin{bmatrix} \Delta x / \Delta t \\ \Delta y / \Delta t \\ \Delta \theta / \Delta t \end{bmatrix} \quad \text{Σχέση 3.6}$$

Οι τιμές  $\dot{x}$ ,  $\dot{y}$  και  $\dot{\theta}$ , θα προκύψουν από την τροχιά, που τη θεωρούμε γνωστή. Για το σκοπό αυτό, θα αφαιρέσουμε από τα  $\mathbf{x}$ ,  $\mathbf{y}$  και  $\theta$  σε κάθε σημείο της τροχιάς, την τιμή τους κατά το προηγούμενο βήμα και θα διαιρέσουμε με το  $\Delta t$ :

$$\Delta t = (\text{Νέα χρονική στιγμή}) - (\text{Προηγούμενη χρονική στιγμή})$$

Δηλαδή οι τιμές του πίνακα ταχυτήτων θα ισούνται με :

$$\dot{x}_I = (\mathbf{x}_n - \mathbf{x}_{n-1}) / \Delta t$$

$$\dot{y}_I = (\mathbf{y}_n - \mathbf{y}_{n-1}) / \Delta t$$

$$\dot{\theta}_I = (\theta_n - \theta_{n-1}) / \Delta t$$

Όπως αναφέραμε νωρίτερα με την χρήση της **σχέσης 3.2** μπορούμε να υπολογίσουμε την ταχύτητα του τοπικού πλαισίου αναφοράς  $\xi_R$  πολλαπλασιάζοντας το  $\xi_I$  με τον πίνακα περιστροφής  $R(\theta)$ .

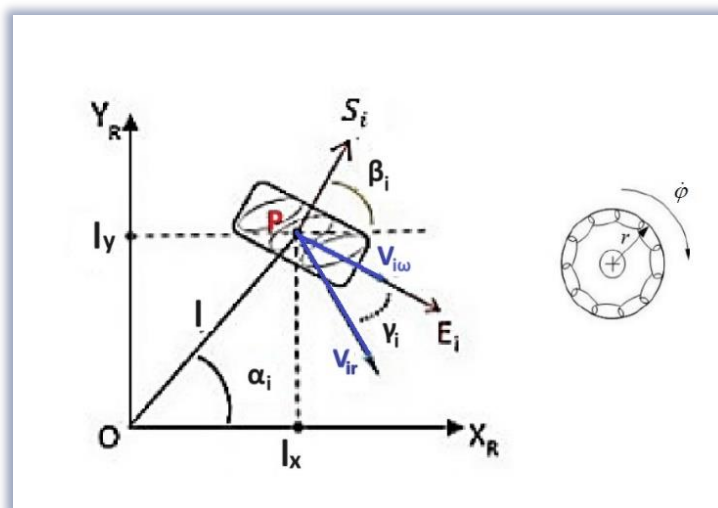
$$\xi_R = R(\theta)\xi_I$$

Για τον υπολογισμό του πίνακα  $R(\theta)$  θα χρειαστεί κάθε φορά να λάβουμε υπόψιν το  $\theta$  της χρονικής στιγμής.

Στη συνέχεια, μπορούμε να βρούμε τις περιστροφικές ταχύτητες των τροχών, χρησιμοποιώντας την αντίστροφη κινηματική εξίσωση (3.5).

### 3.2 Ευθύ και αντίστροφο κινηματικό πρόβλημα οχήματος με 4 συνηθισμένους τροχούς

Όλοι οι τροχοί εμφανίζουν κάποιους περιορισμούς όσον αφορά την κίνηση τους. Αυτοί οι περιορισμοί χωρίζονται σε περιορισμοί κύλισης και περιορισμού ολίσθησής. Η στάση (πόζα) ενός τροχού Swedish, εκφράζεται όπως η στάση του σταθερού τροχού, με τη διαφορά ότι η γωνία γύρω από το σύστημα R του ρομπότ περιέχει ακόμη μία γωνία  $\gamma$ , που παριστάνει τη γωνία ανάμεσα στο επίπεδο του τροχού και τον άξονα περιστροφής των κυλίνδρων (rollers). Στο **σχήμα 3.3** απεικονίζεται η θέση του τροχού ως προς το σασί του οχήματος.



Σχήμα 3.3 – Απεικόνιση τροχού mecanum στο πλαίσιο αναφοράς του οχήματος [9]



Οι παράμετροι του παραπάνω συστήματος αντιστοιχούν στους παρακάτω συμβολισμούς:

$S_i P_i E_i$ : Συντεταγμένες του κάθε τροχού με κέντρο το σημείο  $P$  του κάθε τροχού.

$OP_i$ : Συμβολίζει το διάνυσμα που δείχνει την απόσταση μεταξύ του κέντρου του ρομπότ και του κέντρου του τροχού .

$l_x$ : Η μισή απόσταση ανάμεσα στους δύο μπροστινούς τροχούς.

$l_y$ : Η μισή απόσταση ανάμεσα στους μπροστινούς και τους πίσω τροχούς.

$l$ : Η απόσταση από το κέντρο του ρομπότ (σημείο  $O$ ) έως την βάση του τροχού(σημείο  $P$ ).

$r$ : Ισούται με την ακτίνα του τροχού.

$\alpha_i$ : Η γωνία ανάμεσα στο  $OP_i$  και το  $X_R$  .

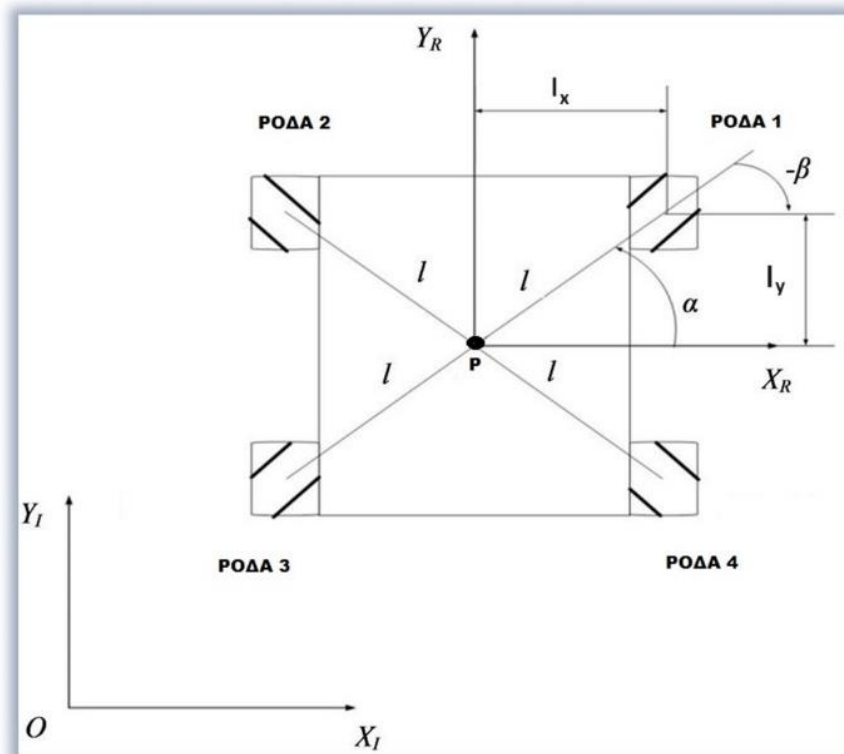
$\beta_i$ : Η γωνία ανάμεσα στο  $S_i$  και το  $X_R$ .

$\gamma_i$ : Η γωνία ανάμεσα στο  $vir$  και το  $E_i$ .

$vi\omega$  [m/s]: το διάνυσμα ταχύτητας που αντιστοιχεί στις στροφές του τροχού.

$vir$ : Η ταχύτητα του των κυλίνδρων για τον κάθε τροχό.

$\varphi$ : Η γωνιακή ταχύτητα του κάθε τροχού.



Σχήμα 3.4 – Απεικόνιση οχήματος στο παγκόσμιο πλαίσιο αναφοράς, όπως το πραγματεύεται η αναφορά [8]

Όλοι τροχοί εμφανίζουν κάποιους περιορισμούς στην κίνηση τους και αυτούς τους χωρίζουμε σε δύο μέρη. Στον περιορισμό κύλισης και τον περιορισμό ολίσθησης.

Οι περιορισμοί του σουηδικού τροχού ισούνται με:

#### Περιορισμός κύλισης:

$$[\sin(\alpha + \beta + \gamma) - \cos(\alpha + \beta + \gamma) (-l)\cos(\beta + \gamma)] \cdot R(\theta) \cdot \dot{\xi}_I - r \cdot \dot{\varphi} \cdot \cos(\gamma) = 0$$

#### Περιορισμός ολίσθησης:

$$[\cos(\alpha + \beta + \gamma) \cos(\alpha + \beta + \gamma) (l)\sin(\beta + \gamma)] \cdot R(\theta) \cdot \dot{\xi}_I - r \cdot \dot{\varphi} \cdot \sin(\gamma) = r_{sw} \cdot \dot{\varphi}_{sw}$$

Ο παρακάτω πίνακας (**Πίνακας 3.1**) μας εμφανίζει την γενική μορφή με την οποία μπορούμε να υπολογίσουμε την τις γωνίας  $\alpha$ ,  $\beta$  και  $\gamma$ , ώστε να μπορέσουμε να λύσουμε τους περιορισμούς που αναφέραμε καθώς και τις παρακάτω εξισώσεις.

**Πίνακας 3.1- Γενική μορφή υπολογισμού γωνιών  $\alpha$ ,  $\beta$  και  $\gamma$ . [8]**

Ρόδα	$\alpha_i$	$\beta_i$	$\gamma_i$	$l_i$	$l_{ix}$	$l_{iy}$
1	$\tan^{-1}\left(\frac{l_{y1}}{l_{x1}}\right)$	$-\tan^{-1}\left(\frac{l_{y1}}{l_{x1}}\right)$	$(\pi/2+\pi/4)$	$l_1$	$l_{x1}$	$l_{y1}$
2	$\pi - \tan^{-1}\left(\frac{l_{y2}}{l_{x2}}\right)$	$\tan^{-1}\left(\frac{l_{y2}}{l_{x2}}\right)$	$-(\pi/2+\pi/4)$	$l_2$	$l_{x2}$	$l_{y2}$
3	$\pi + \tan^{-1}\left(\frac{l_{y3}}{l_{x3}}\right)$	$-\tan^{-1}\left(\frac{l_{y3}}{l_{x3}}\right)$	$(\pi/2+\pi/4)$	$l_3$	$l_{x3}$	$l_{y3}$
4	$2\pi - \tan^{-1}\left(\frac{l_{y4}}{l_{x4}}\right)$	$\tan^{-1}\left(\frac{l_{y4}}{l_{x4}}\right)$	$-(\pi/2+\pi/4)$	$l_4$	$l_{x4}$	$l_{y4}$

Λαμβάνουμε τις εξισώσεις για όλους τους τροχούς, με βάση τον περιορισμό και δημιουργούμε ένα σύστημα ανεξάρτητων εξισώσεων όπου N ο αριθμός των τροχών του συστήματος [7]

$$J = \begin{bmatrix} \sin(\alpha_1 + \beta_1 + \gamma_1) & -\cos(\alpha_1 + \beta_1 + \gamma_1) & (-l_1)\cos(\beta_1 + \gamma_1) \\ \sin(\alpha_2 + \beta_2 + \gamma_2) & -\cos(\alpha_2 + \beta_2 + \gamma_2) & (-l_2)\cos(\beta_2 + \gamma_2) \\ \sin(\alpha_3 + \beta_3 + \gamma_3) & -\cos(\alpha_3 + \beta_3 + \gamma_3) & (-l_3)\cos(\beta_3 + \gamma_3) \\ \sin(\alpha_4 + \beta_4 + \gamma_4) & -\cos(\alpha_4 + \beta_4 + \gamma_4) & (-l_4)\cos(\beta_4 + \gamma_4) \end{bmatrix}$$

Ο πίνακας  $J$  που θα δημιουργήσουμε, συμβάλει στον υπολογισμό των τελικών ταχυτήτων του ρομποτικού μας οχήματος με την **σχέση 3.7** που βλέπουμε παρακάτω:

$$\begin{bmatrix} \sin(a_1 + \beta_1 + \gamma_1) & -\cos(a_1 + \beta_1 + \gamma_1) & (-l_1)\cos(\beta_1 + \gamma_1) \\ \sin(a_2 + \beta_2 + \gamma_2) & -\cos(a_2 + \beta_2 + \gamma_2) & (-l_2)\cos(\beta_2 + \gamma_2) \\ \sin(a_3 + \beta_3 + \gamma_3) & -\cos(a_3 + \beta_3 + \gamma_3) & (-l_3)\cos(\beta_3 + \gamma_3) \\ \sin(a_4 + \beta_4 + \gamma_4) & -\cos(a_4 + \beta_4 + \gamma_4) & (-l_4)\cos(\beta_4 + \gamma_4) \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} r_1\dot{\phi}_1\cos\gamma_1 \\ r_2\dot{\phi}_2\cos\gamma_2 \\ r_3\dot{\phi}_3\cos\gamma_3 \\ r_4\dot{\phi}_4\cos\gamma_4 \end{bmatrix}$$

**Σχέση 3.7**

Εφόσον όλες οι ρόδες έχουν ίδια διάμετρο και η απόσταση της τοποθέτησης είναι ίσες μεταξύ τους και χρησιμοποιήσουμε της μεταβλητές  $\alpha$ ,  $\beta$  και  $\gamma$  για κάθε ρόδα, μπορούμε να λύσουμε ως προς τις περιστροφικές ταχύτητες των τροχών και να βρούμε το αντίστροφο κινηματικό πρόβλημα.

Η κινηματική εξίσωση θα είναι [8]:

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} = -\left(\frac{\sqrt{2}}{r}\right) \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & l\sin(\frac{\pi}{4} - \psi) \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & l\sin(\frac{\pi}{4} - \psi) \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & l\sin(\frac{\pi}{4} - \psi) \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & l\sin(\frac{\pi}{4} - \psi) \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{bmatrix}$$

**Σχέση 3.8**

Όπου:  $\psi = \tan^{-1}(l_y/l_x)$ , όπου τα  $l_x$  και τα  $l_y$  φαίνονται στο Σχήμα 3.4.

Ο υπολογισμός του ευθέως κινηματικού προβλήματος σύμφωνα με την παραπάνω μέθοδο προκύπτει λύνοντας ως προς τις ταχύτητες στο αδρανειακό σύστημα  $I$  και δίνεται από την παρακάτω σχέση (βλέπε αναφορά [8]):

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{bmatrix} = -\left(\frac{\sqrt{2}}{2}\right) r J^+ \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix}$$

**Σχέση 3.9**

όπου:

$$J^+ = (J^T J)^{-1} J^T$$

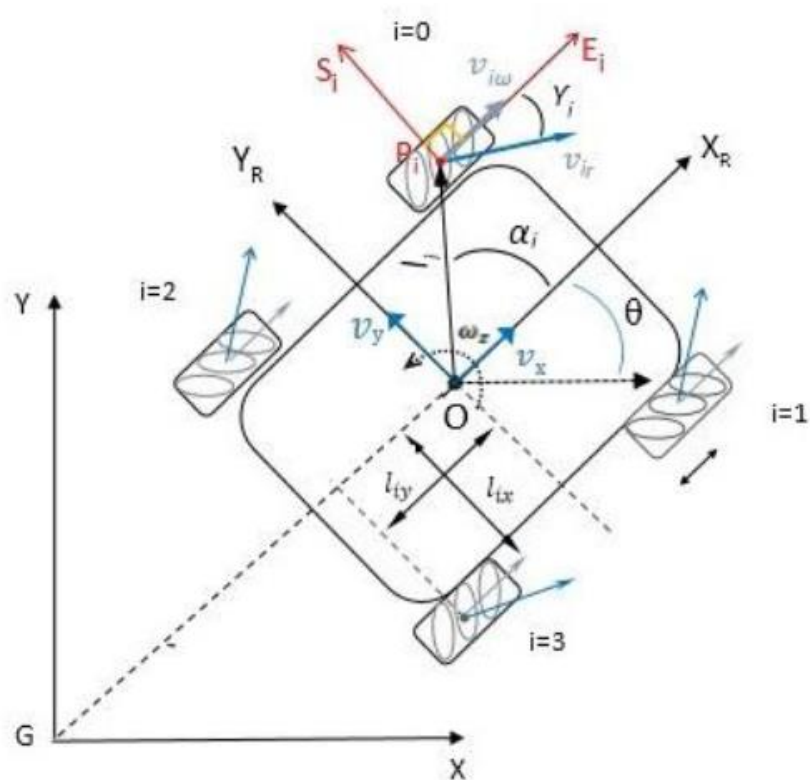
**Σχέση 3.10**

Και ο πίνακας  $J$  ισούται με:

$$J = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & l \sin(\frac{\pi}{4} - \psi) \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & l \sin(\frac{\pi}{4} - \psi) \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & l \sin(\frac{\pi}{4} - \psi) \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & l \sin(\frac{\pi}{4} - \psi) \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

### 3.3 Ευθύ και αντίστροφο κινηματικό πρόβλημα οχήματος με 4 σουηδικούς τροχούς: ένας εναλλακτικός υπολογισμός

Μια διαφορετική μελέτη του ίδιου θέματος, λύνει το ευθύ κινηματικό πρόβλημα με έναν διαφορετικό τρόπο από τον παραπάνω. Εδώ, θεωρούμε μια διαμόρφωση του ρομπότ mecanum με τέσσερις σουηδικούς τροχούς, όπως φαίνεται στο παρακάτω σχήμα.



Σχήμα 3.5 Το ρομπότ mecanum με τέσσερις σουηδικούς τροχούς, όπως το πραγματεύεται η αναφορά [9]

Η συγκεκριμένη μελέτη καταλήγει ως τελικό συμπέρασμα στις παρακάτω σχέσεις για το αντίστροφο κινηματικό πρόβλημα (δες αναφορά [9]):

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} = \frac{-1}{r} \begin{bmatrix} \frac{\cos(\beta_1-\gamma_1)}{\sin \gamma_1} & \frac{\sin(\beta_1-\gamma_1)}{\sin \gamma_1} & \frac{l_1 \sin(\beta_1-\gamma_1-a_1)}{\sin \gamma_1} \\ \frac{\cos(\beta_2-\gamma_2)}{\sin \gamma_2} & \frac{\sin(\beta_2-\gamma_2)}{\sin \gamma_2} & \frac{l_2 \sin(\beta_1-\gamma_1-a_1)}{\sin \gamma_1} \\ \frac{\cos(\beta_3-\gamma_3)}{\sin \gamma_3} & \frac{\sin(\beta_3-\gamma_3)}{\sin \gamma_3} & \frac{l_3 \sin(\beta_3-\gamma_3-a_3)}{\sin \gamma_3} \\ \frac{\cos(\beta_4-\gamma_4)}{\sin \gamma_4} & \frac{\sin(\beta_4-\gamma_4)}{\sin \gamma_4} & \frac{l_4 \sin(\beta_4-\gamma_4-a_4)}{\sin \gamma_4} \end{bmatrix} \begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{bmatrix} \quad \text{Σχέση 3.11}$$

Η γενική μορφή του ευθέως κινηματικού προβλήματος είναι:

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{bmatrix} = \mathbf{T}^+ \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} \quad \text{Σχέση 3.12}$$

Όπου ο πίνακας T ισούται με:

$$\mathbf{T} = \frac{-1}{r} \begin{bmatrix} \frac{\cos(\beta_1-\gamma_1)}{\sin \gamma_1} & \frac{\sin(\beta_1-\gamma_1)}{\sin \gamma_1} & \frac{l_1 \sin(\beta_1-\gamma_1-a_1)}{\sin \gamma_1} \\ \frac{\cos(\beta_2-\gamma_2)}{\sin \gamma_2} & \frac{\sin(\beta_2-\gamma_2)}{\sin \gamma_2} & \frac{l_2 \sin(\beta_1-\gamma_1-a_1)}{\sin \gamma_1} \\ \frac{\cos(\beta_3-\gamma_3)}{\sin \gamma_3} & \frac{\sin(\beta_3-\gamma_3)}{\sin \gamma_3} & \frac{l_3 \sin(\beta_3-\gamma_3-a_3)}{\sin \gamma_3} \\ \frac{\cos(\beta_4-\gamma_4)}{\sin \gamma_4} & \frac{\sin(\beta_4-\gamma_4)}{\sin \gamma_4} & \frac{l_4 \sin(\beta_4-\gamma_4-a_4)}{\sin \gamma_4} \end{bmatrix}$$

Ο ψευδοαντίστροφος πίνακας  $T^+$  δίνεται από τη σχέση

$$\mathbf{T}^+ = (\mathbf{T}^T \mathbf{T})^{-1} \mathbf{T}^T$$

**Πίνακας 3.2 Τιμές των γωνιών στη διαμόρφωση του σχήματος 3.5**

$i$	Wheels	$\alpha_i$	$\beta_i$	$\gamma_i$	$l_i$	$l_{ix}$	$l_{iy}$
0	1sw	$\pi/4$	$\pi/2$	$-\pi/4$	$l$	$l_x$	$l_y$
1	2sw	$-\pi/4$	$-\pi/2$	$\pi/4$	$l$	$l_x$	$l_y$
2	3sw	$3\pi/4$	$\pi/2$	$\pi/4$	$l$	$l_x$	$l_y$
3	4sw	$-3\pi/4$	$-\pi/2$	$-\pi/4$	$l$	$l_x$	$l_y$

Αντικαθιστώντας τις τιμές των γωνιών  $\alpha$ ,  $\beta$  και  $\gamma$  από τον παραπάνω **πίνακα 3.2** και κάνοντας τις πράξεις στον πίνακα **T** έχουμε μία τελική μορφή, όπου πλέον το **T** ισούται με:

$$T = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(l_x + l_y) \\ 1 & 1 & (l_x + l_y) \\ 1 & 1 & -(l_x + l_y) \\ 1 & -1 & (l_x + l_y) \end{bmatrix} \quad \text{Σχέση 3.13}$$

$$T^+ = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} & -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} \end{bmatrix} \quad \text{Σχέση 3.14}$$

Σύμφωνα λοιπόν με την σχέση 3.2 το αντίστροφο κινηματικό πρόβλημα μπορεί να εκφραστεί και ως εξής (βλέπε αναφορές [9], [10]):

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} = T \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(l_x + l_y) \\ 1 & 1 & (l_x + l_y) \\ 1 & 1 & -(l_x + l_y) \\ 1 & -1 & (l_x + l_y) \end{bmatrix} \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta} \end{bmatrix} \quad \text{Σχέση 3.15}$$

Το ευθύ κινηματικό πρόβλημα με αυτήν την μέθοδο μπορεί να γραφεί ως εξής:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta} \end{bmatrix} = T^+ \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} & -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} \end{bmatrix} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} \quad \text{Σχέση 3.16}$$

Άρα για να υπολογίσουμε τις γωνιακές ταχύτητες του κάθε τροχού μπορούμε να χρησιμοποιήσουμε την παρακάτω σχέση για κάθε τροχό:

$$\begin{cases} \dot{\phi}_1 = \frac{1}{r} (\dot{x} - \dot{y} - (l_x + l_y)\dot{\theta}) \\ \dot{\phi}_2 = \frac{1}{r} (\dot{x} + \dot{y} + (l_x + l_y)\dot{\theta}) \\ \dot{\phi}_3 = \frac{1}{r} (\dot{x} + \dot{y} - (l_x + l_y)\dot{\theta}) \\ \dot{\phi}_4 = \frac{1}{r} (\dot{x} - \dot{y} + (l_x + l_y)\dot{\theta}) \end{cases} \quad \text{Σχέση 3.17}$$

Επίσης μπορούμε να υπολογίσουμε το διάνυσμα των ταχυτήτων  $\dot{\xi}$  [11]:

$$\begin{cases} \dot{x}(t) = (\dot{\phi}_1 + \dot{\phi}_2 + \dot{\phi}_3 + \dot{\phi}_4) \frac{r}{4} \\ \dot{y}(t) = (-\dot{\phi}_1 + \dot{\phi}_2 + \dot{\phi}_3 - \dot{\phi}_4) \frac{r}{4} \\ \dot{\theta}(t) = (-\dot{\phi}_1 + \dot{\phi}_2 - \dot{\phi}_3 + \dot{\phi}_4) \frac{r}{4(l_x + l_y)} \end{cases} \quad \text{Σχέση 3.18}$$

Η **σχέση 3.18** μας βοηθάει να αντιληφθούμε τον τρόπο με τον οποίο θα κινηθεί το όχημα ανάλογα πάντα με τις περιστροφικές ταχύτητες των τροχών.

# Κεφάλαιο IV. Μέθοδοι και υλικά

## 4.1 Εισαγωγή

Σε αυτή την ενότητα θα μελετήσουμε τα εργαλεία που θα χρησιμοποιήσουμε για να κατασκευάσουμε ένα όχημα με τέσσερις σουηδικούς τροχούς, το οποίο θα μπορούμε να το κατευθύνουμε απομακρυσμένα μέσω του ασύρματου τοπικού δικτύου Wi-Fi, στον χώρο στον οποίο βρισκόμαστε.

Όπως είδαμε στις προηγούμενες ενότητες οι σουηδικοί τροχοί είναι ικανοί να κινούνται προς όλες τις κατευθύνσεις με αποτέλεσμα να είναι σημαντικοί στη ρομποτική για τον ξεχωριστό σχεδιασμό και τις βελτιωμένες ικανότητές τους, όπως το γεγονός ότι παρέχουν μεγαλύτερο εύρος κινητικότητας και ευελιξίας, επομένως προσφέρουν πολλά πλεονεκτήματα.

Η κατασκευή ενός τέτοιου οχήματος έχει αναμφίβολα τα οφέλη του, αλλά παρουσιάζει επίσης ορισμένα εμπόδια και περιορισμούς. Για παράδειγμα, η κατασκευή σουηδικών τροχών απαιτεί περισσότερη προσπάθεια και δεξιότητα από την κατασκευή συμβατικών τροχών και μπορεί να οδηγήσει σε αύξηση του προϋπολογισμού και της συντήρησης. Σημαντικός περιορισμός είναι και το βάρος που μπορεί να συγκρατήσουν αυτοί οι τροχοί ώστε να λειτουργούν ομαλά και να μην ζορίζεται το ρομπότ.

Ένα από τα μειονεκτήματα που παρουσιάζεται σε τέτοια οχήματα είναι πως το σύστημα ελέγχου των τροχών θα χρειαστεί να καταναλώσει περισσότερους πόρους για να κατευθύνει το όχημα σε σύγκριση με άλλα οχήματα με συμβατικούς τροχούς, για αυτόν τον λόγο θα χρειαστεί περισσότερη μελέτη και περισσότερος χρόνος για τον σωστό προγραμματισμό του οχήματος, ώστε να λειτουργεί σωστά. Εξίσου σημαντικό θα είναι και ο χειριστής που θα το λειτουργήσει να είναι κατάλληλα εκπαιδευμένος, ώστε να μην έχουμε ατυχήματα.

## 4.2 Εργαλεία ανάπτυξης

Η κατασκευή ενός οχήματος είναι αρκετά περίπλοκη καθώς χρειάζονται γνώσεις από πολλά επιστημονικά πεδία για παράδειγμα:

- **γνώσεις Προγραμματισμού**, για τον προγραμματισμό του οχήματος καθώς και για την δημιουργία πλατφόρμας ελέγχου κάνοντας χρήση της γλώσσας προγραμματισμού Python.
- **γνώσεις Ηλεκτρονικών κυκλωμάτων**, για την σωστή επιλογή μοτέρ, οδηγών ισχύος των μοτέρ, της μπαταρίας η οποία θα τροφοδοτήσει όλο το σύστημα και φυσικά του μικροελεγκτή ή μικροεπεξεργαστή που θα ελέγχει όλο το όχημα.



- γνώσεις **Μηχανολογίας**, ώστε να μπορέσουμε να κατασκευάσουμε και να συναρμολογήσουμε όλα τα εξαρτήματα μας.
- γνώσεις **3D Σχεδιασμού και 3D Εκτύπωσης**, ώστε να σχεδιάσουμε όλα τα πλαστικά μέρη του οχήματός μας κυρίως για αισθητικούς λόγους και στη συνέχεια να τα εκτυπώσουμε με την βοήθεια ενός 3D εκτυπωτή.

### 4.3 Προγραμματισμός του ελεγκτή – η γλώσσα προγραμματισμού Python

Η Python έχει καθιερωθεί ως μια από τις προτιμώμενες γλώσσες για προγραμματιστές και ερευνητές που εργάζονται στη βιομηχανία ρομποτικής, χάρη στην ευκολία χρήσης της, το τεράστιο οικοσύστημα βιβλιοθηκών και την ισχυρή υποστήριξη της κοινότητας. Είναι μια ευέλικτη και φιλική προς το χρήστη γλώσσα προγραμματισμού. Στις ενότητες που ακολουθούν, θα εξετάσουμε τη χρήση της στη ρομποτική, τονίζοντας τα πλεονεκτήματα, τις χρήσεις και τη συμβολή της στις εξελίξεις του τομέα [12].

- **Απλότητα και αναγνωσιμότητα:** Η καθαρή και ευανάγνωστη σύνταξη της την καθιστά ιδανική επιλογή για προγραμματισμό ρομποτικής. Η απλότητά της, επιτρέπει στους προγραμματιστές να επικεντρωθούν στη λογική και τους αλγόριθμους αντί να κολλήσουν σε περίπλοκη σύνταξη. Επιπλέον, η κατανοητή βάση κώδικα της Python ενισχύει τη συνεργασία και την ανταλλαγή γνώσεων μεταξύ των ερασιτεχνών της ρομποτικής και των ερευνητών.
- **Εκτεταμένο οικοσύστημα βιβλιοθηκών:** Διαθέτει ένα τεράστιο οικοσύστημα βιβλιοθηκών που παρέχει έτοιμα προς χρήση εργαλεία και ενότητες για διάφορες ρομποτικές εφαρμογές. Βιβλιοθήκες όπως οι NumPy, SciPy και OpenCV προσφέρουν ισχυρές λειτουργίες για αριθμητικούς υπολογισμούς, επιστημονικούς υπολογισμούς και υπολογιστική όραση, αντίστοιχα.
- **Module ρομποτικής:** Έχει κερδίσει δημοτικότητα στον τομέα της ρομποτικής λόγω της διαθεσιμότητας ισχυρών βιβλιοθηκών ειδικά σχεδιασμένες για ανάπτυξη ρομποτικής. Το ROS (Robot Operating System) παρέχει ένα ολοκληρωμένο πλαίσιο για την κατασκευή και τον έλεγχο ρομπότ. Παρέχει εργαλεία επικοινωνίας, αντίληψης, σχεδιασμού κίνησης και πολλά άλλα, καθιστώντας την ανάπτυξη σύνθετων ρομπότ πιο προσιτή.
- **Δημιουργία πρωτοτύπων και ταχεία ανάπτυξη:** Η ευκολία χρήσης της και οι γρήγορες δυνατότητες δημιουργίας πρωτοτύπων την καθιστούν εξαιρετική επιλογή για ταχεία ανάπτυξη

στη ρομποτική. Η διερμηνευόμενη φύση της επιτρέπει ταχύτερες επαναλήψεις κώδικα, επιτρέποντας στους προγραμματιστές να πειραματιστούν, να δοκιμάσουν και να βελτιώσουν αποτελεσματικά τους αλγορίθμους και τις στρατηγικές ελέγχου τους. Το διαδραστικό περιβάλλον ανάπτυξης και τα εργαλεία εντοπισμού σφαλμάτων ενισχύουν περαιτέρω τη διαδικασία ανάπτυξης.

- **Μηχανική μάθηση και ενσωμάτωση AI:** Οι εκτεταμένες βιβλιοθήκες μηχανικής μάθησης και τεχνητής νοημοσύνης της, όπως το TensorFlow και το PyTorch, έχουν φέρει επανάσταση στον τομέα της ρομποτικής. Αυτές οι βιβλιοθήκες διευκολύνουν την ενσωμάτωση προηγμένων αλγορίθμων μάθησης, επιτρέποντας στα ρομπότ να μαθαίνουν από δεδομένα, να προσαρμόζονται σε δυναμικά περιβάλλοντα και να λαμβάνουν έξυπνες αποφάσεις. Η ευελιξία της στην εργασία με διαφορετικούς τύπους δεδομένων και η υποστήριξη της για νευρωνικά δίκτυα έχουν επιταχύνει την ενσωμάτωση της τεχνητής νοημοσύνης σε εφαρμογές ρομποτικής.
- **Εκπαιδευτική χρήση και υποστήριξη από την κοινότητα:** Η δημοτικότητα και η ευκολία μάθησης της Python την έχουν καταστήσει προτιμώμενη γλώσσα για την εκπαίδευση και την έρευνα στη ρομποτική. Οι τεράστιοι κοινοτικοί πόροι, τα διαδικτυακά σεμινάρια και η τεκμηρίωση την καθιστούν προσβάσιμη τόσο σε αρχάριους όσο και σε έμπειρους προγραμματιστές. Η υποστηρικτική κοινότητα προωθεί την ανταλλαγή γνώσεων, τα συνεργατικά έργα και την ανάπτυξη πλαισίων ρομποτικής ανοιχτού κώδικα και βιβλιοθηκών.

Οι εφαρμογές για αυτόνομα αυτοκίνητα, βιομηχανικό αυτοματισμό, τεχνολογία drone και αλληλεπίδραση ανθρώπου με ρομπότ είναι μόνο μερικοί από τους τομείς όπου η επιρροή της Python στη ρομποτική είναι σαφής.

Για τους λόγους που αναφέραμε παραπάνω χρησιμοποιήθηκε στην συγκεκριμένη εργασία η γλώσσα Python για την κατασκευή ενός GUI (Graphical User Interface) με το οποίο θα μπορούμε να ελέγχουμε το όχημα μας από τον υπολογιστή. Το GUI δημιουργήθηκε με την χρήση της βιβλιοθήκης Tkinter.

Η Python διαθέτει πολλά πλαίσια GUI, αλλά το Tkinter είναι το μόνο πλαίσιο που είναι ενσωματωμένο στην πρότυπη βιβλιοθήκη της Python. Το Tkinter έχει πολλά πλεονεκτήματα. Λειτουργεί σε όλα τα λειτουργικά συστήματα (cross-platform), οπότε ο ίδιος κώδικας λειτουργεί σε Windows, macOS και Linux. Τα οπτικά στοιχεία απεικονίζονται με τη χρήση εγγενών στοιχείων του λειτουργικού συστήματος, οπότε οι εφαρμογές που έχουν δημιουργηθεί με το Tkinter μοιάζουν σαν να ανήκουν στην πλατφόρμα όπου εκτελούνται.

Το μειονέκτημα που έχουν τα GUIs που κατασκευάζονται με το Tkinter είναι πως φαίνονται ξεπερασμένα. Παρόλα αυτά, το Tkinter είναι ελαφρύ στη χρήση σε σύγκριση με άλλα πλαίσια. Αυτό

το καθιστά μια συναρπαστική επιλογή για τη δημιουργία εφαρμογών GUI στην Python, ειδικά για εφαρμογές όπου η σύγχρονη λάμψη δεν είναι απαραίτητη και η βασική προτεραιότητα είναι να δημιουργήσουμε γρήγορα κάτι που να είναι λειτουργικό και cross-platform.

## 4.4 Τομέας Ηλεκτρονικών Κυκλωμάτων

### 4.4.1 Raspberry Pi

Σε μια εποχή που χαρακτηρίζεται από τεχνολογικές εξελίξεις, το Raspberry Pi αποτελεί ένα επαναστατικό εργαλείο που έχει αιχμαλωτίσει τη φαντασία όσων ενδιαφέρονται για την τεχνολογία, των εκπαιδευτικών και των ερασιτεχνών. Αναπτύχθηκε με το όραμα να κάνει τους υπολογιστές προσβάσιμους σε όλους. Είναι ένας μικρός υπολογιστής μονής πλακέτας (single board) που έχει αποκτήσει τεράστια δημοτικότητα λόγω της προσιτής τιμής, της ευελιξίας και της ευκολίας χρήσης του[13].

Το Raspberry Pi, σχεδιάστηκε και αναπτύχθηκε από το Ίδρυμα Raspberry Pi, εισήχθη το 2012 με στόχο να φέρει επανάσταση στον τρόπο με τον οποίο οι άνθρωποι μαθαίνουν, πειραματίζονται και δημιουργούν με υπολογιστές. Πρόκειται για μια πλακέτα μεγέθους πιστωτικής κάρτας εξοπλισμένη με επεξεργαστή, μνήμη, θύρες εισόδου / εξόδου και ισχυρό λειτουργικό σύστημα. Παρά το μικρό του μέγεθος, το Raspberry Pi διαθέτει σημαντική υπολογιστική ισχύ που ανταγωνίζεται τα παραδοσιακά συστήματα.

Η ευελιξία του είναι ένα από τα καθοριστικά χαρακτηριστικά του. Υποστηρίζει μια σειρά λειτουργικών συστημάτων, συμπεριλαμβανομένων διανομών που βασίζονται σε Linux, οι οποίες προσφέρουν ένα τεράστιο οικοσύστημα εφαρμογών και εργαλείων. Επιπλέον, διαθέτει πολλές επιλογές συνδεσιμότητας όπως USB, HDMI, Ethernet, Wi-Fi και Bluetooth, επιτρέποντας την απρόσκοπτη ενσωμάτωση με εξωτερικές συσκευές και δίκτυα. Οι ακίδες GPIO (General Purpose Input/Output) επιτρέπουν τη σύνδεση διαφόρων αισθητήρων, επενεργητών και άλλων εξαρτημάτων, καθιστώντας το ιδανικό για πρωτότυπα και projects με βάση το Internet of Things (IoT).

Το Raspberry Pi έχει βρει ευρεία χρήση σε διάφορους τομείς, επηρεάζοντας τις βιομηχανίες και προωθώντας την καινοτομία με αδιανόητους τρόπους. Στην εκπαίδευση, έχει γίνει ένα πολύτιμο εργαλείο για τη διδασκαλία της επιστήμης των υπολογιστών, του προγραμματισμού και της ηλεκτρονικής. Η οικονομική προσιτότητά του, επιτρέπει στα σχολεία να ενσωματώσουν πρακτικές μαθησιακές εμπειρίες στο πρόγραμμα σπουδών τους, δίνοντας τη δυνατότητα στους μαθητές να εξερευνήσουν την κωδικοποίηση, τη ρομποτική και την ψηφιακή δημιουργία.

Στον τομέα του οικιακού αυτοματισμού, λειτουργεί ως κεντρικός κόμβος, ελέγχοντας και παρακολουθώντας διάφορες έξυπνες συσκευές, που κυμαίνονται από φώτα και θερμοστάτες έως συστήματα ασφαλείας. Η χαμηλή κατανάλωση ενέργειας και το μικρό αποτύπωμα το καθιστούν μια ενεργειακά αποδοτική λύση που μπορεί να μετατρέψει οποιαδήποτε κατοικία σε έξυπνο σπίτι.

Το Raspberry Pi έχει γίνει καταλύτης για την καινοτομία, εμπνέοντας τα άτομα να μετατρέψουν τις ιδέες τους σε πραγματικότητα. Η προσβασιμότητά του έχει μειώσει το εμπόδιο εισόδου για τους εκκολαπτόμενους επιχειρηματίες και εφευρέτες, επιτρέποντάς τους να πρωτοτυπήσουν και να δοκιμάσουν τις ιδέες τους χωρίς σημαντική οικονομική επένδυση. Η ακμάζουσα κοινότητα που το περιβάλλει προσφέρει μια πλατφόρμα για ανταλλαγή γνώσεων, συνεργατικά έργα και ανάπτυξη λογισμικού και υλικού ανοιχτού κώδικα.

#### 4.4.2 Υπόλοιπα ηλεκτρονικά εξαρτήματα

##### Γρανάζωτός κινητήρας συνεχούς ρεύματος

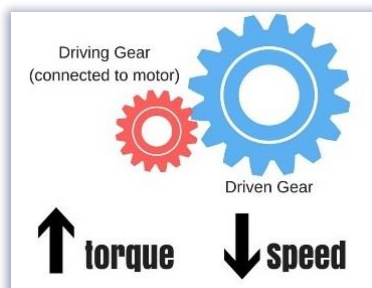


Σχήμα 4.1 - Γρανάζωτός κινητήρας

Οι κινητήρες συνεχούς ρεύματος είναι πολύ εύκολοι στην εφαρμογή και οι πιο συχνά χρησιμοποιούμενοι επενεργητές σε ρομπότ. Υπάρχουν διάφοροι τύποι κινητήρων συνεχούς ρεύματος διαθέσιμοι. Ο κινητήρας συνεχούς ρεύματος παρέχει καλή ταχύτητα χωρίς αρκετή ροπή. Για να ξεπεραστεί αυτό, ο κινητήρας DC συχνά συνδυάζεται με γρανάζια που παρέχουν μεγαλύτερη ροπή, αλλά μειώνουν την ταχύτητα.

Η αλλαγή του λόγου ταχυτήτων είναι ένας από τους ευκολότερους τρόπους αλλαγής του μηχανικού πλεονεκτήματος ενός μηχανισμού ή συστήματος για την επίτευξη της επιθυμητής ταχύτητας ή ροπής. Η χρήση γραναζιών διαφορετικού μεγέθους εμπλέκονται μεταξύ τους όπως μπορούμε να δούμε στο **σχήμα 4.1**, και μπορούν να αλλάξουν την ταχύτητα ή τη ροπή που αποδίδει ο κινητήρας.

Η σχέση ταχύτητας/ροπής είναι αντίστροφη μεταξύ τους. Όταν αυξάνετε η ταχύτητα, μειώνετε η ροπή και όταν αυξάνετε η ροπή, μειώνετε την ταχύτητα. Η ροπή μπορεί να θεωρηθεί ως η ποσότητα ισχύος που παρέχεται ως δύναμη στρέψης στον άξονα κίνησης.



**Σχήμα 4.2 - Αναπαράσταση σχέσης ροπής και ταχύτητας.**

Σε μια εφαρμογή όπως στο παραπάνω **σχήμα 4.2**, όταν ένα μικρότερο γρανάτζι συνδέεται με έναν κινητήρα και το μεγαλύτερο γρανάτζι κινείται, η ταχύτητα μειώνεται και όπως αναφέραμε, μεγαλώνει η ροπή. Αυτό λειτουργεί επειδή το μεγαλύτερο γρανάτζι περιστρέφεται πιο αργά από το μικρότερο γρανάτζι, καθώς έχει περισσότερα δόντια.

### **Μπαταρίες Λιθίου-Πολυμερών (Li-Po)**

Η μπαταρία λιθίου-πολυμερών, ευρέως γνωστή ως μπαταρία Li-Po, λειτουργεί με την τεχνολογία ιόντων λιθίου, αντί του συνήθως χρησιμοποιούμενου υγρού ηλεκτρολύτη. Αυτού του είδους οι μπαταρίες είναι επαναφορτιζόμενες, παρέχοντας έτσι στους χρήστες τεράστια εξοικονόμηση κόστους.

Χρησιμοποιούνται ειδικά σε προϊόντα και συσκευές όπου το βάρος είναι ένα κρίσιμο χαρακτηριστικό και έχει ιδιαίτερη σημασία. Επίσης, δεν υπάρχει περιορισμός στο σχήμα της μπαταρίας, είναι ανθεκτικές στην υπερφόρτιση ενώ ο κίνδυνος διαρροής χημικών είναι μικρός. Όλα τα παραπάνω



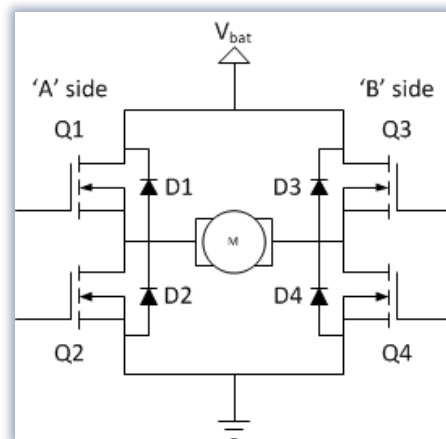
**Σχήμα 4.3 - Μπαταρία Li-Po 2700mAh.**

χαρακτηριστικά κάνουν τις μπαταρίες Li-Po ιδιαίτερα δημοφιλείς στις εφαρμογές ρομποτικής. Στο **σχήμα 4.3** απεικονίζεται μία κλασική μπαταρία Li-Po.

### Motor Driver Controller Board L298N Dual H Bridge

Το κύκλωμα οδήγησης κινητήρων L298N χρησιμοποιείται για τον έλεγχο της ταχύτητας και της κατεύθυνσης των κινητήρων. Το κύκλωμα L298N μπορεί να οδηγήσει ένα σερβοκινητήρα δύο φάσεων, ένα σερβοκινητήρα τεσσάρων φάσεων ή δύο κινητήρες συνεχούς ρεύματος. Για τους κινητήρες, η μόνη προϋπόθεση είναι η σύνδεση εξωτερικής πηγής τροφοδοσίας, μιας και οι μικροελεγκτές συνήθως έχουν μέγιστη έξοδο στα 5VDC και τα μοτέρ χρειάζονται περισσότερα Volt.

Διαθέτει τέσσερις εισόδους αντίστοιχες των τεσσάρων διακοπών της διάταξης της «γέφυρας τύπου H» (OUT1, OUT2, OUT3, OUT4). Ο πίνακας ελέγχου έχει τερματικούς ακροδέκτες των 12V και 5V. Το τερματικό των 12V μπορεί να δεχτεί τάσεις από 7V έως 12V όταν ο συνδετήρας (jumper) των +12V είναι στην θέση του, ενώ από 12V έως 35V χωρίς το jumper. Στην πρώτη περίπτωση ενεργοποιείται ο ρυθμιστής τάσης του κυκλώματος και μπορεί να δεχθεί 5V. Οι ακροδέκτες των κινητήρων συνδέονται με τους ακροδέκτες IN1, IN2, IN3 και IN4 του κυκλώματος L298N. Για να χρησιμοποιήσουμε σερβοκινητήρες διατηρούμε τα jumpers στις ακίδες ENA και ENB. Διαφορετικά, για να χρησιμοποιήσουμε κινητήρες συνεχούς ρεύματος, αφαιρούμε τα jumper και αυτές τις ακίδες τις συνδέουμε με ακίδες PWM του μικροεπεργαστή [14].

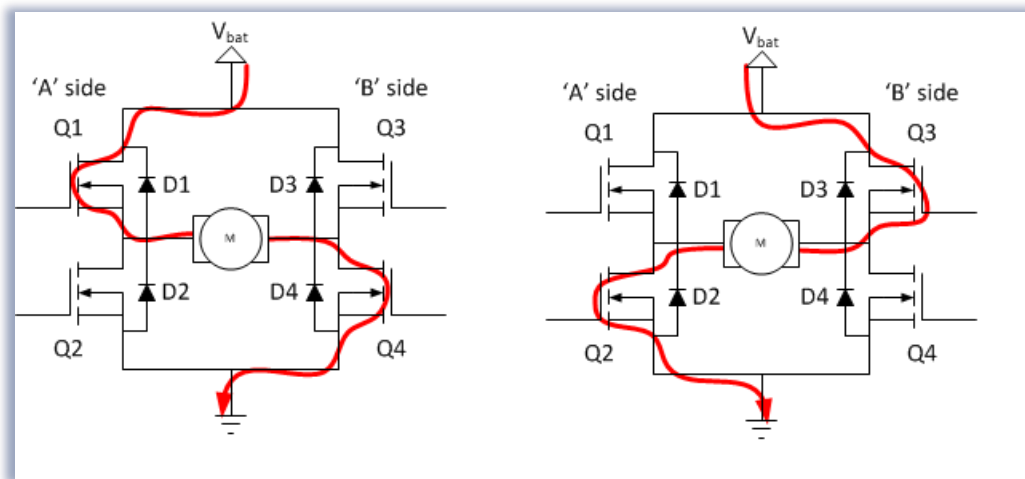


**Σχήμα 4.4 - Διάταξη Dual H Bridge.**

Το κύκλωμα οδήγησης κινητήρων L298N χρησιμοποιεί την διάταξη της «γέφυρας τύπου H» (H-Bridge) για τον έλεγχο της κατεύθυνσης περιστροφής ενός DC κινητήρα.

Οι διακόπτες στην διάταξη «γέφυρας τύπου H» ορίζουν την κατεύθυνση περιστροφής του κινητήρα. Με κλειστούς τους διακόπτες Q1 και Q4 ο κινητήρας περιστρέφεται προς μία κατεύθυνση ενώ με κλειστούς τους διακόπτες Q2 και Q3 ο κινητήρας περιστρέφεται προς άλλη κατεύθυνση.

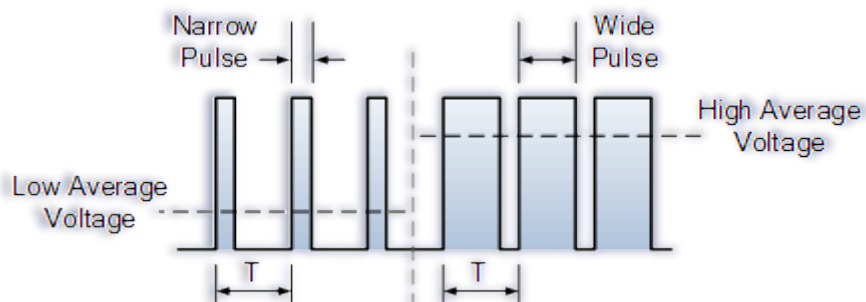
Για να περιστρέψουμε τον αριστερό κινητήρα προς μία κατεύθυνση αρκεί να εφαρμόσουμε έναν υψηλό παλμό στην ακίδα IN1 και έναν χαμηλό παλμό στην ακίδα IN2. Για να αντιστρέψουμε την κατεύθυνση, αρκεί να εφαρμόσουμε αντίστροφα τους παλμούς τις ακίδες IN1 και IN2. Ομοίως, περιστρέφουμε και τον δεξιό κινητήρα με τις ακίδες IN3 και IN4.



Σχήμα 4.5 – Διάταξη κίνησης του ηλεκτρικού φορτίου στις δυο εναλλακτικές διαδρομές.

#### Διαμόρφωση πλάτους παλμού (PWM)

Η ροπή σε έναν κινητήρα συνεχούς ρεύματος είναι ανάλογη του ρεύματος οδήγησης. Για τον έλεγχο της ταχύτητας ενός κινητήρα συνεχούς ρεύματος, η ροπή του διαμορφώνεται χρησιμοποιώντας κύκλωμα διαμόρφωσης πλάτους παλμού (PWM ή Pulse Width Modulation). Ο PWM ενεργοποιεί και να απενεργοποιεί έναν διακόπτη με πολύ γρήγορο ρυθμό, που λέγεται Duty Cycle. Όσο περισσότερο χρόνο ο διακόπτης είναι ON, τόσο μεγαλύτερη είναι η τάση που στέλνεται στον δέκτη δημιουργώντας μεγαλύτερες ταχύτητες.



Σχήμα 4.6 – Αναπαράσταση διάφορων παλμών σήματος PWM.

## 4.5 Σχεδιασμός 3D και 3D Εκτύπωση

### 4.5.1 Σχεδιαστικό πρόγραμμα Fusion 360

Στον κόσμο του σχεδιασμού και της μηχανικής το Fusion 360 είναι ένα λογισμικό 3D CAD που κερδίζει δημοτικότητα τα τελευταία χρόνια, λόγω του ολοκληρωμένου συνόλου χαρακτηριστικών και δυνατοτήτων του. Αναπτύχθηκε από την Autodesk, και έχει αναδειχθεί ως ένα εργαλείο λογισμικού που έφερε την επανάσταση στον τρόπο με τον οποίο οι σχεδιαστές, οι μηχανικοί και οι κατασκευαστές ζωντανεύουν τις ιδέες τους καθώς μπορεί να χρησιμοποιηθεί για σχεδιασμό προϊόντων, δημιουργία πρωτοτύπων, κινούμενα σχέδια και άλλα.

Βασίζεται στην τεχνολογία του Cloud και επιτρέπει τη συνεργασία και τη διαχείριση δεδομένων σε πραγματικό χρόνο. Ξεκίνησε το 2013 και έχει αποκτήσει τεράστια δημοτικότητα λόγω της φιλικής διεπαφής προς το χρήστη, των συνεργατικών δυνατοτήτων και της ενσωμάτωσης πολλαπλών λειτουργιών σχεδιασμού και μηχανικής σε μια ενιαία πλατφόρμα. Παρέχει στους σχεδιαστές και τους μηχανικούς ένα ευρύ φάσμα ισχυρών εργαλείων και χαρακτηριστικών.

Από τον σχεδιασμό έως την κατασκευή, το Fusion 360 παρέχει μια ολοκληρωμένη ροή εργασίας που βελτιστοποιεί τη διαδικασία ανάπτυξης προϊόντων. Προσφέρει ισχυρή παραμετρική μοντελοποίηση, επιτρέποντας στους χρήστες να δημιουργούν πολύπλοκα 3D σχέδια με ευελιξία σχεδιασμού. Με ενσωματωμένα εργαλεία προσομοίωσης και ανάλυσης, οι σχεδιαστές μπορούν να επικυρώσουν τα σχέδιά τους για δομική ακεραιότητα, κίνηση και θερμική απόδοση, βελτιστοποιώντας τα προϊόντα τους πριν από την κατασκευή πρωτοτύπων ή την κατασκευή.

Η ενσωμάτωση των δυνατοτήτων κατασκευής με τη βοήθεια υπολογιστή (CAM), προσθέτει μια άλλη διάσταση στη λειτουργικότητά του. Οι χρήστες μπορούν να δημιουργήσουν διαδρομές εργαλείων, να προσομοιώσουν τις λειτουργίες κατεργασίας και να δημιουργήσουν κώδικα G για μηχανές CNC απευθείας μέσα στο λογισμικό. Αυτή η απρόσκοπτη ενοποίηση μεταξύ σχεδιασμού και κατασκευής μειώνει την ανάγκη μετάφρασης δεδομένων, διασφαλίζοντας ακρίβεια και αποτελεσματικότητα σε όλη τη διαδικασία παραγωγής.

### 4.5.2 Τρισδιάστατοι εκτυπωτές

Η τρισδιάστατη εκτύπωση (3D printing) είναι μια μέθοδος προσθετικής κατασκευής στην οποία κατασκευάζονται αντικείμενα μέσω της διαδοχικής πρόσθεσης επάλληλων στρώσεων υλικού. Στη τρισδιάστατη εκτύπωση μπορούν να χρησιμοποιηθούν διάφοροι τύποι υλικού, κυρίως κεραμικά και πολυμερή. Σε σύγκριση με άλλες τεχνολογίες και εξοπλισμό προσθετικής κατασκευής, οι τρισδιάστατοι εκτυπωτές είναι συνήθως ταχύτεροι, φθηνότεροι και ευκολότεροι στη χρήση. Για τον λόγο αυτό πολλοί



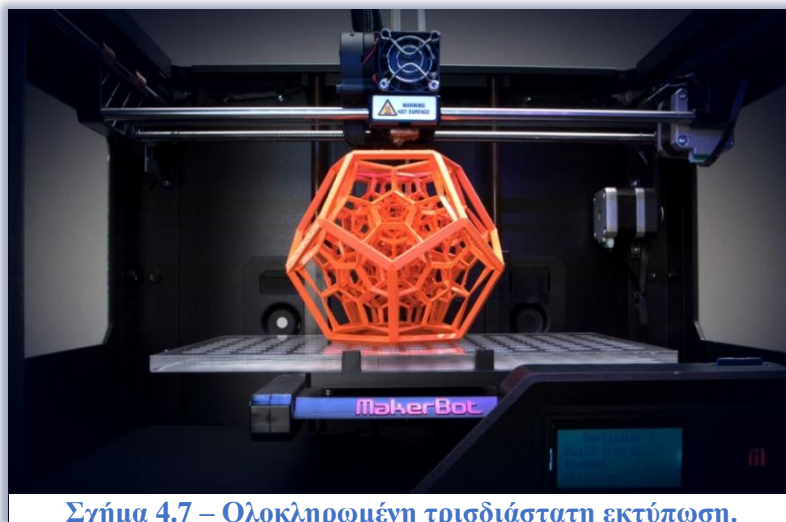
πιστεύουν ότι στα επόμενα χρόνια η παγκόσμια παραγωγή αγαθών θα στραφεί προς αυτή την κατεύθυνση, αντικαθιστώντας σταδιακά τις παραδοσιακές τεχνικές. Δεν είναι λίγοι αυτοί που πιστεύουν ότι η τρισδιάστατη εκτύπωση θα αποτελέσει μία «νέα βιομηχανική επανάσταση», καθώς θα φέρει αποκέντρωση των παραγωγικών διαδικασιών, ανοίγοντας τον δρόμο για παραγωγή τοπική και μικρής κλίμακας, προσαρμοσμένη στις τρέχουσες ανάγκες.

Οι τρισδιάστατοι εκτυπωτές χρησιμοποιούνται κυρίως για την κατασκευή φυσικών μοντέλων και πρωτοτύπων από σχεδιαστές, μηχανικούς και ομάδες ανάπτυξης νέων προϊόντων, έχουν τη δυνατότητα να εκτυπώνουν μέρη και εξαρτήματα από διάφορα υλικά, με διαφορετικές μηχανικές και φυσικές ιδιότητες και συχνά σε μια ενιαία διαδικασία κατασκευής. Η νέα τεχνολογία διαχείρισης και μετακίνησης υλικών (ως έχουν ή με αναπαραγωγή τους), ονομάζεται (ψηφιακό) MatterNet, κατά αναλογία της τεχνολογίας του διαδικτύου (internet), που επιτρέπει την διαχείριση και μεταφορά των πληροφοριών (κειμένων, σταθερών ή κινούμενων εικόνων και ήχου).

#### 4.5.3 Λειτουργία 3D εκτύπωσης

Η 3D εκτύπωση αφορά στη δημιουργία ενός πραγματικού αντικειμένου στον τρισδιάστατο κόσμο μας μέσα από διάφορες μεθόδους. Αυτό που χρειάζεται για να πραγματοποιηθεί είναι ένας 3D εκτυπωτής, το υλικό για την κατασκευή και ένα ψηφιακό μοντέλο 3D (ηλεκτρονικό αρχείο). Το τελικό αντικείμενο δημιουργείται με την αλληπάλληλη εναπόθεση στρωμάτων του υλικού εκτύπωσης, το ένα πάνω στο άλλο.

Αρχικά, θα πρέπει να έχουμε ένα τρισδιάστατο-μοντέλο. Αυτό είναι ένα ψηφιακό σχέδιο, σε μορφή αρχείου (file format) STL ή OBJ που είναι και τα πιο συνηθισμένα και τα οποία δημιουργούνται είτε από εφαρμογή CAD ή με τη χρήση ενός 3D-σαρωτή, ή ακόμα και από ψηφιακές φωτογραφικές μηχανές με τη βοήθεια κατάλληλου λογισμικού.



Σχήμα 4.7 – Ολοκληρωμένη τρισδιάστατη εκτύπωση.

Σε μια διαδικασία που ονομάζεται fix-up το αρχείο προς εκτύπωση ελέγχεται για πιθανά σφάλματα. Στη συνέχεια, το αρχείο υπόκειται σε μια επεξεργασία όπου έχει σαν αποτέλεσμα τον τεμαχισμό του μοντέλου σε μία σειρά από λεπτές στρώσεις. Τέλος, το αρχείο αναλύεται σε «οδηγίες» για ένα συγκεκριμένο 3D εκτυπωτή και αποστέλλεται στο μηχάνημα όπου θα ξεκινήσει δημιουργία του αντικείμενου σε στρώσεις. Τα στρώματα αυτά είναι εξαιρετικά λεπτά (τυπικά 0,1 χιλιοστά ή και λιγότερο) και όσο λεπτότερες είναι οι στρώσεις, τόσο λιγότερο φαίνονται οι διατομές στο τελικό αντικείμενο. Πάντως, σε μερικές περιπτώσεις, ένα τελικό φινίρισμα απαιτείται προκειμένου να απομακρυνθούν μικρό ατέλειες και να βελτιωθεί με γυάλισμα η επιφάνεια.

#### 4.5.4 Οι πιο δημοφιλείς μέθοδοι 3D εκτύπωσης

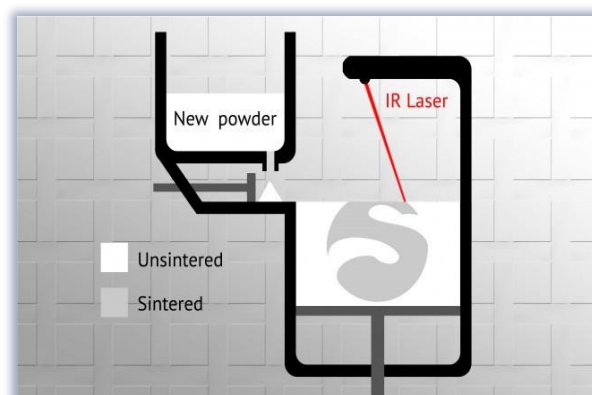
Η μέθοδος που χρησιμοποιείται από τους περισσότερους επιτραπέζιους 3D εκτυπωτές είναι γνωστή ως Εναπόθεση Λιωμένου Υλικού ( Fused Deposition Modeling – FDM). Στην τεχνική αυτή, μία θερμαινόμενη κεφαλή σε σχήμα σωληνάριου χρησιμοποιείται για να λιώσει το υλικό εκτύπωσης και να το εξωθήσει μέσω μιας κεφαλής (ακροφύσιο). Το υλικό, συνήθως κάποιο είδος πλαστικού, μερικές φορές πορσελάνη ακόμη και πηλός, τροφοδοτείται με τη μορφή λεπτού νήματος.

Υπάρχουν και άλλες μέθοδοι «εκτύπωσης» όπου χρησιμοποιούνται διαφορετικά υλικά. Η βασική αρχή παραμένει λίγο πολύ η ίδια σε όλες τις τεχνικές: Το αντικείμενο είναι χτισμένο στρώμα-στρώμα από τη βάση προς την κορυφή. Πάντως δε συμβαίνει σε όλες τις μεθόδους το υλικό να θερμαίνεται και να εξάγεται μέσω ακροφυσίου. Ορισμένες μέθοδοι περιλαμβάνουν λέιζερ για την τήξη και τη στερεοποίηση και κάποιο υλικό σε μορφή σκόνης.

**Άλλες κοινές μέθοδοι είναι:**

- **SLS: Selective Laser Sintering**

Στην επιλεκτική πυροσυσσωμάτωση με λέιζερ χρησιμοποιείται λέιζερ, ώστε να λιώσουν επιλεκτικά κόκκοι από κονιοποιημένο υλικό (π.χ. μέταλλο) και στη συνέχεια να ενοποιηθούν σε ένα

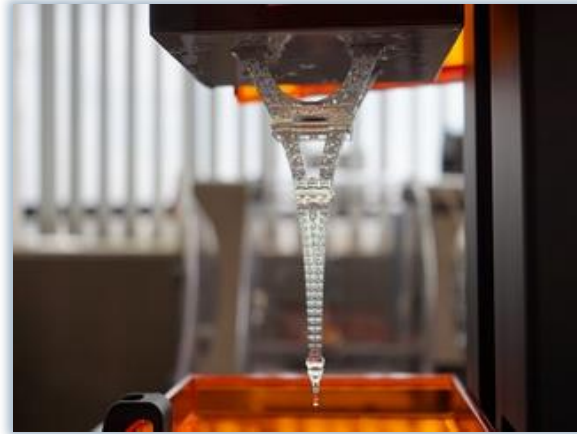


Σχήμα 4.8 – Μέθοδος τρισδιάστατης εκτύπωσης SLS

στρώμα, για το σταδιακό χτίσιμο του τελικού αντικειμένου. Το λέιζερ στοχεύει σε σημεία που ορίζονται από το ηλεκτρονικό αρχείο του 3D σχεδίου. Στο σχήμα 4.8 απεικονίζεται η μέθοδος εκτύπωση SLS.

- **SLA: Stereolithography**

Στη στερεολιθογραφία το μοντέλο χτίζεται στρώμα – στρώμα με τη χρήση υγρής ρητίνης η οποία στερεοποιείται αν εκτεθεί σε συγκεκριμένο φωτισμό (UV-laser ή ψηφιακό προβολέα). Στο σχήμα 4.9 απεικονίζεται η μέθοδος εκτύπωσης SLA.



**Σχήμα 4.9 – Μέθοδος τρισδιάστατης εκτύπωσης SLA**

## 4.6 Έρευνα αγοράς υλικών

Ύστερα από την μελέτη που πραγματοποιήθηκε έχουμε πλέον τα εργαλεία και την γνώση για να επιλέξουμε όλα τα υλικά που θα μας χρειαστούν για την κατασκευή του συγκεκριμένου οχήματος. Όλα τα υλικά αγοράστηκαν από την τοπική αγορά με γνώμονα να είναι όλα συμβατά μεταξύ τους και να μπορούν να λειτουργήσουν με τον τρόπο που επιθυμούμε. Είναι σημαντικό να είναι όλα τα υλικά σχετικά απλά και ετοιμοπαράδοτα, ώστε αν έχουμε κάποιο ελαττωματικό ηλεκτρονικό ή κάτι χαλάσει να μπορεί να αλλαχτεί εύκολα. Παρακάτω φαίνονται όλα τα εξαρτήματα που χρησιμοποιήθηκαν, καθώς και ένας πίνακας με τα κόστη τους.

- Αλουμινένιο σασί ρομποτικού οχήματος με 4 σουηδικούς τροχούς διαμέτρου 60mm



Σχήμα 4.10 – Σασί αλουμινίου και σουηδικοί τροχοί

- Τέσσερις γραναζωτοί κινητήρες συνεχούς ρεύματος (48/1)



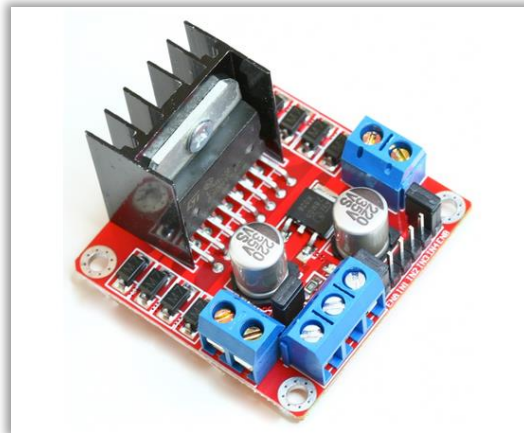
Σχήμα 4.11 – Γραναζωτοί κινητήρες συνεχούς ρεύματος

- Raspberry Pi 4



Σχήμα 4.12 – Raspberry Pi 4

- Stepper Motor Driver Controller Board L298N Dual H Bridge



Σχήμα 4.13 – Controller Board L298N

- Μπαταρία Li-PO 3S 2200mAh/35C Lipo Battery with XT60



Σχήμα 4.14 – Μπαταρία Li-Po

- IMAX B6 DC Charger 5A 50W



Σχήμα 4.15 –Φορτιστής μπαταρίας Li-Po IMAX B6

- LM2596 3~40V DC-DC Buck Adjustable Converter / Step Down



Σχήμα 4.16 – Πλακέτα LM2596 DC-DC converter

Στον παρακάτω **πίνακα 4.1** εμφανίζονται οι τιμές αγοράς όλων των περιφερειακών ηλεκτρονικών εξαρτημάτων που χρησιμοποιήθηκαν για την ολοκλήρωση της διπλωματικής εργασίας.

**Πίνακας 4.1- Κοστολόγηση της διπλωματικής εργασίας**

A/A	ΠΕΡΙΓΡΑΦΗ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ	ΣΥΝΟΛΟ ΕΙΔΟΥΣ
1	Αλουμινένιο σασί και ρόδες	1	35.00€	35.00€
2	Γραναζωτοί κινητήρες συνεχούς ρεύματος	4	2.50€	10.00€
3	Raspberry Pi 4	1	90.00€	90.00€
4	Stepper Motor Driver Controller Board L298N	2	6.00€	12.00€
5	Μπαταρία Li-PO 3S 2200mAh/35C	2	26.00€	52.00€
6	IMAX B6 DC Charger 5A 50W	1	35.00€	35.00€
7	LM2596 3~40V DC-DC Buck Adjustable Converter	1	3.00€	3.00€
8	Filament 3D Εκτυπωτή	500gr	12.00€	12.00€
Συνολική αξία				250.00€

# Κεφάλαιο V. Ερευνητική μεθοδολογία

## 5.1 Εισαγωγή

Στο κεφάλαιο αυτό θα αναλύσουμε και θα παρουσιάσουμε την διαδικασία η οποία ακολουθήθηκε για την κατασκευή του ρομποτικού οχήματος. Θα γίνει επίσης μια περιγραφή ο τρόπος με τον οποίον συνδέθηκαν όλα τα ηλεκτρονικά κυκλώματα και τέλος θα περιγράψουμε τον κώδικα της εφαρμογής που ελέγχει το όχημα και στη συνέχεια τον κώδικα που κινεί το όχημα.

## 5.2 Σχεδιασμός 3D μοντέλου και 3D εκτύπωση

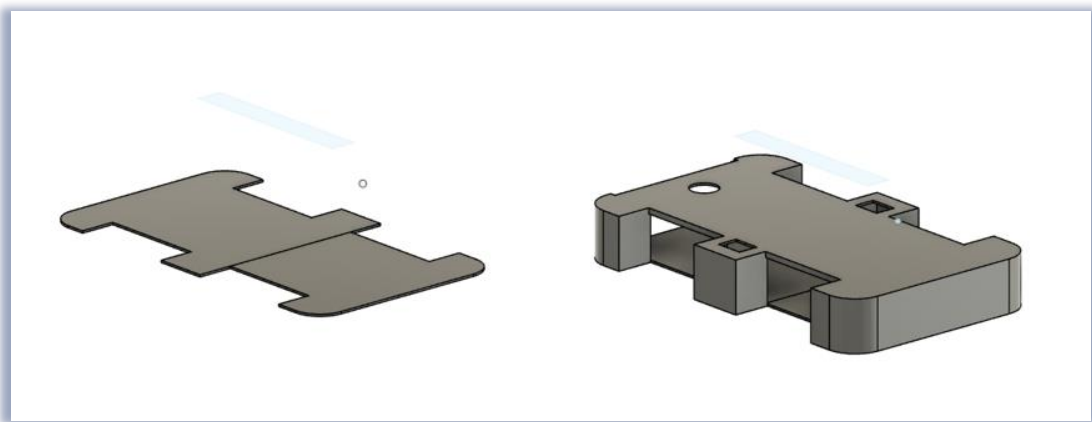
Όπως αναφέραμε και παραπάνω για τον σχεδιασμό των κατάλληλων εξαρτημάτων για το ρομποτικό όχημα χρησιμοποιήθηκε το πρόγραμμα Fusion 360. Συναρμολογώντας το μεταλλικό σασί μαζί με τις ρόδες έχουμε μια αρχική εικόνα του σκελετού της κατασκευής μας. Μελετώντας λοιπόν το σασί σχεδιάσαμε γύρω του κομμάτια τα οποία όταν συναρμολογηθούν θα δώσουν μια όμορφη εικόνα ενός οχήματος. Στο παρακάτω σχήμα 5.1 μπορούμε να δούμε το σασί γυμνό, χωρίς κάποια παρέμβαση. Τα μοτέρ μπηκαν ανάμεσα στις 2 πλάκες από αλουμίνιο ώστε το ολικό ύψος να είναι σχετικά μικρό.



Σχήμα 5.1 - Το σασί της κατασκευής.

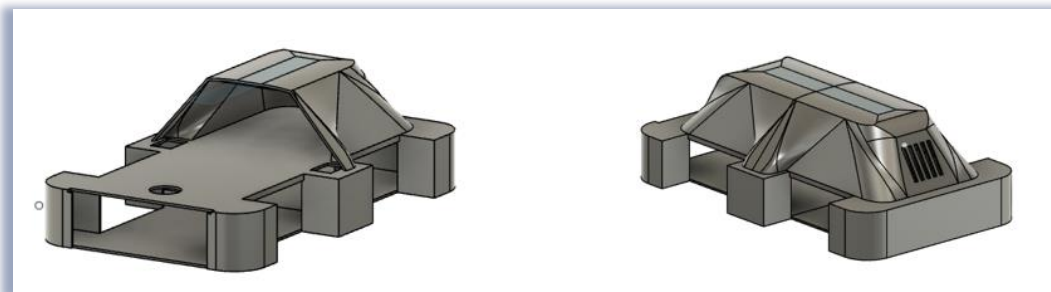
Ο σχεδιασμός 3D θα μας βοηθήσει στο να κάνουμε το ρομποτικό όχημα όσο το δυνατόν πιο όμορφο εμφανισιακά. Για να πετύχουμε το επιθυμητό αποτέλεσμα θα χρειαστεί να χωριστεί τμηματικά. Το πρώτο μέρος θα απαρτίζεται από τα πλαστικά μέρη για το σασί. Το δεύτερο μέρος θα περιλαμβάνει τα πλαστικά μέρη για τα ηλεκτρονικά μέρη του οχήματος, ώστε να μην φαίνονται και να είναι τακτοποιημένα. Τρίτο και τελευταίο μέρος θα είναι το εσωτερικό κομμάτι που θα συγκρατεί την μπαταρία και θα μας δίνει την δυνατότητα να την βγάζουμε όποτε το επιθυμούμε.

Ξεκινώντας τη σχεδίαση θα επιχειρήσουμε να κλείσουμε το σασί του οχήματος γύρω-γύρω ώστε να μην έχουμε κενά, για να μην φαίνεται η καλωδίωση. Στο **σχήμα 5.2** μπορούμε να δούμε το κάτω μέρος του οχήματος και στην εικόνα μπορούμε να δούμε τα πλαστικά που θα τοποθετηθούν γύρω γύρω στο σασί.



**Σχήμα 5.2 – Τα τρισδιάστατα σχέδια που καλύπτουν το σασί περιφερειακά.**

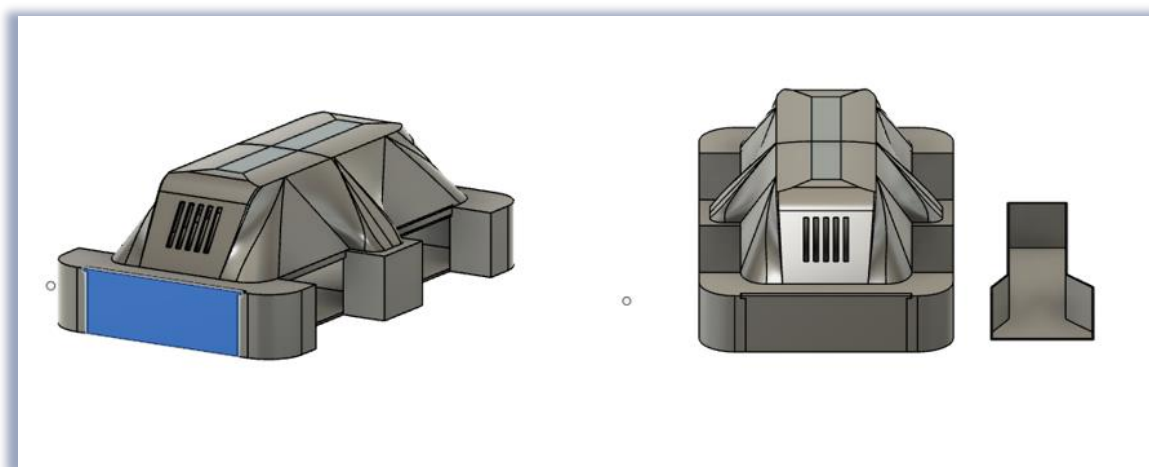
Έχοντας ολοκληρώσει το πρώτο στάδιο του σχεδιασμού μας προχωράμε στο επόμενο κομμάτι. Για να μπορέσουμε να περιβάλουμε όλα τα ηλεκτρονικά μέρη του οχήματος. Μπροστά και πίσω έχουμε δυο αεραγωγούς και με την βοήθεια δυο ανεμιστήρων έχουμε κυκλοφορία του αέρα, ώστε να ψύχονται τα εξαρτήματα χωρίς να κινδυνεύουν να καούν. Στις **σχήμα 5.3** φαίνεται η κοιλότητα η οποία υπολογίστηκε και σχεδιάστηκε έτσι ώστε να χωρέσουν εσωτερικά όλα τα ηλεκτρονικά μέρη.



**Σχήμα 5.3 – Απεικόνιση της τρισδιάστατης εκτύπωσης με την κοιλότητα**

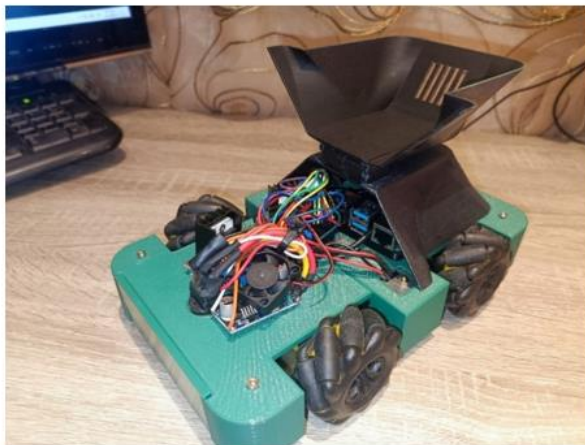


Για να μπορέσουμε να αποκρύψουμε την μπαταρία αλλά και να μπορεί να βγαίνει ώστε να την φορτίζουμε δημιουργήθηκε μια ειδική υποδοχή στη διάσταση της μπαταρίας. Βρισκόμαστε ήδη στο τελικό στάδιο της σχεδίασης μας και όπως θα δούμε στο **σχήμα 5.4**, έχει επισημανθεί το ειδικό πορτάκι που ανοίγει ώστε να βγάζουμε την μπαταρία όποτε αυτό είναι επιθυμητό. Στην εικόνα β στα δεξιά του σχήματος μπορούμε να παρατηρήσουμε την ειδική υποδοχή της μπαταρίας. Έχοντας στην αρχή στα τοιχώματα μια κωνικότητα μας βοηθάει στο να μπει εύκολα η μπαταρία στην θέση της και στην συνέχεια μένει σταθερή στο σημείο καθώς δεν μπορεί να προχωρήσει ή να φύγει από την θέση της χάρη στα εσωτερικά τοιχώματα που διαθέτει η αποδοχή.



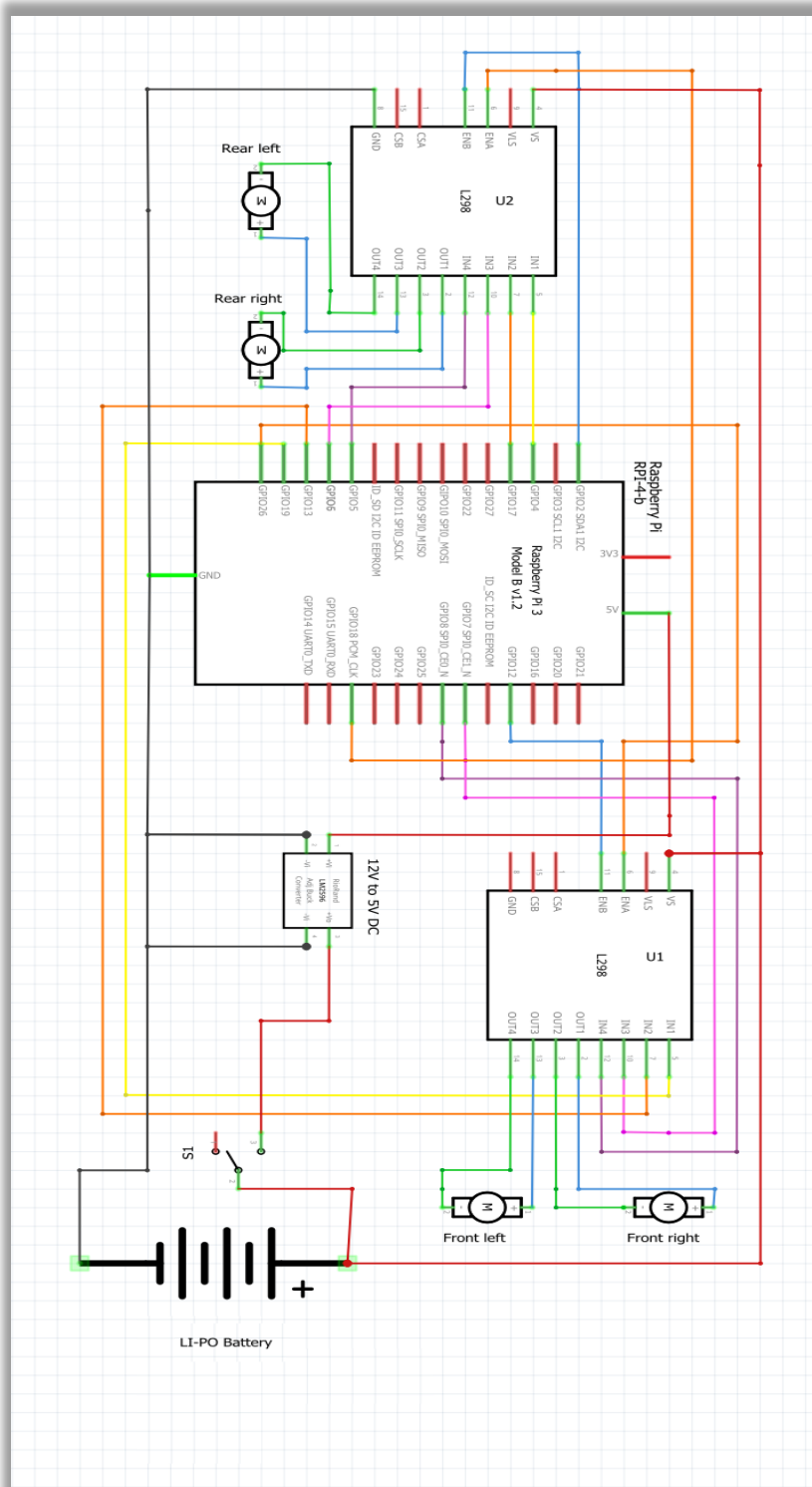
**Σχήμα 5.4 – Τελική μορφή της τρισδιάστατης εκτύπωσης**

Το τελικό αποτέλεσμα του 3D σχεδιασμού έχει ολοκληρωθεί ύστερα από κάποιες ώρες σχεδιασμού και διορθώσεων όσον αφορά της λεπτομέρειες. Στη συνέχεια, όπως αναφέραμε παραπάνω, με την βοήθεια ενός 3D εκτυπωτή θα εκτυπωθούν όλα τα μέρη που σχεδιάσαμε. Συνολικά το όχημα απαρτίζεται από οκτώ διαφορετικά εξαρτήματα, τα οποία ανάλογα με το σχέδιο και το μέγεθος του καθενός μπορεί να χρειαστεί από τέσσερις έως δώδεκα ώρες εκτύπωσης για να ολοκληρωθεί το καθένα ξεχωριστά. Επομένως, ο λεπτομερής σχεδιασμός είναι πολύ σημαντικός ώστε να μην βγουν λάθος τα εξαρτήματα και χρειαστεί μετά ξανά εκτύπωση από την αρχή. Φυσικά, για αυτήν την μεγάλη καθυστέρηση στην εκτύπωση οφείλεται το γεγονός πως ο εκτυπωτής είναι ερασιτεχνικός και φτηνός. Με ένα εξειδικευμένο εκτυπωτή οι χρόνοι αυτοί θα μπορούσαν να ελαττωθούν σημαντικά. Παρόλα αυτά, ευτυχώς για την εργασία αυτή, ο σχεδιασμός ήταν σωστός και έτσι δεν είχαμε κάποια χρονική απώλεια. Στο παρακάτω **σχήμα 5.5** εμφανίζεται το όχημα στην τελική του μορφή.



Σχήμα 5.5 – Τελική κατασκευή του οχήματος που μελετάμε.

### 5.3 Ηλεκτρονικό κύκλωμα.



Σχήμα 5.6 – Συνδεσμολογία του ρομποτικού οχήματος.

Στο **σχήμα 5.6** μπορούμε να δούμε την συνδεσμολογία του ρομποτικού οχήματος. Είναι αρκετά σημαντικό κομμάτι της συγκεκριμένης διπλωματικής εργασίας και χρειάζεται σωστή επιλογή των ηλεκτρονικών εξαρτημάτων που θα χρησιμοποιήσουμε για την υλοποίηση της. Όπως μπορούμε να δούμε το σχέδιο δεν είναι πάρα πολύ περίπλοκο, επομένως και οι φυσικές συνδεσμολογίες είναι σχετικά απλές. Εφαρμόζουμε αρκετά από όσα διδαχθήκαμε κατά την διάρκεια των σπουδών μας στο Μεταπτυχιακό της Ρομποτικής, ώστε να συνδεθούν όλα με τον κατάλληλο τρόπο.

Είναι σημαντικό να χρησιμοποιηθούν τα σωστά εξαρτήματα διότι μια λάθος επιλογή τους πέρα από το γεγονός ότι δεν θα λειτουργήσει σωστά το όχημα (ίσως και να μην λειτουργήσει καθόλου), υπάρχει και ο κίνδυνος να καούν ολοκληρωμένα, να χαλάσουν οι θύρες του Raspberry Pi με αποτέλεσμα να μην μπορούμε πλέον να χρησιμοποιήσουμε τις GPIO θύρες. Ο μεγαλύτερος κίνδυνος είναι αυτός της πυρκαγιάς, μιας και ένα βραχυκύκλωμα ή μία υπέρταση μπορεί να χαλάσει την μπαταρία και σε ακραίες περιπτώσεις να έχουμε ακόμα και φωτιά.

Η μπαταρίας είναι ζωτικό εξάρτημα για την εργασία. Ο λόγος που χρησιμοποιήσαμε μπαταρία είναι για να μπορέσει το όχημα να μην έχει κάποιους περιορισμούς εξαιτίας των καλωδίων και με αυτόν τον τρόπο μπορεί να κατευθύνεται προς όλες της πλευρές. Η μπαταρία τροφοδοτεί όλα τα ηλεκτρονικά εξαρτήματα. Είναι Li-Po Battery τριών κυψελών (S3) στα 11.1 Volt και έχει ρυθμό εκ φόρτισης 35C. Την συγκεκριμένη μπαταρία την φορτίζουμε με ειδικό φορτιστή ο οποίος είναι ο **IMAX B6 DC Charger 5A 50W** που αναφέραμε στον πίνακα με τα εξαρτήματα που αγοράσαμε.

Το Raspberry Pi για την λειτουργία του χρειάζεται γύρω στα 5 Volt DC, επομένως τα σχεδόν 12 Volt που μας αποδίδει η μπαταρία είναι υπερτροφοδότηση και η πιθανότητα να χαλάσει είναι βέβαιη. Για τον λόγο αυτόν χρησιμοποιήσαμε μία έτοιμη πλακέτα την **LM2596 DC to DC Buck Converter** με την οποία κατεβάζουμε την τάση από τα 11.1V στα ασφαλή 5V λειτουργίας του..

Το Raspberry Pi θα είναι ο κεντρικός επεξεργαστής του οχήματος μας και βλέποντας το σχέδιο των συνδεσμολογιών, έχουν γίνει όλες οι απαραίτητες συνδέσεις των θυρών GPIO, ώστε να μπορέσει με τον κατάλληλο προγραμματισμό να λειτουργήσει σωστά. Τα απαραίτητα σήματα δίνουν εντολή στους **Driver Controller Board L298N**, τα οποία πέρα από την φορά του τροχού, παίρνοντας το κατάλληλο PWM σήμα θα ρυθμίζει και την ταχύτητα περιστροφής των μοτέρ.

Όσον αφορά τα μοτέρ μπορούμε να δούμε πως η τροφοδοσία τους γίνεται με 12 Volt DC καθώς τα 5 Volt που μας παρέχει το Raspberry δεν είναι αρκετά. Χρησιμοποιώντας κατάλληλα όργανα μέτρησης παρατηρήθηκε πως το όχημα σε λειτουργία αδράνειας χρειάζεται 0.5 A. Αν επιθυμούμε να κάνουμε κίνηση για την οποία χρειάζεται να λειτουργήσουν όλοι οι τροχοί, τότε θα χρειαστεί περίπου 1A όταν λειτουργούν και οι 4 κινητήρες με την μέγιστη ταχύτητα. Αν επιλέξουμε να κάνουμε κίνηση η οποία θα χρησιμοποιηθούν μόνο 2 τροχοί, τότε θα χρειαστεί περίπου 0.75 A για να λειτουργήσει.

## 5.4 Ανάλυση μαθηματικού μοντέλου στο MATLAB

Με την βοήθεια του προγράμματος MATLAB δημιουργήσαμε ένα σύντομο πρόγραμμα για να μπορέσουμε να αναλύσουμε την κίνηση του οχήματος κάνοντας αρκετές προσομοιώσεις. Στόχος ήταν να παρατηρήσουμε αν όντως το όχημα μας θα κινηθεί με τον τρόπο που αναφέραμε πριν ακόμα περάσουμε σε πραγματικές δοκιμές με το ίδιο το όχημα που κατασκευάσαμε.

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta} \end{bmatrix} = T^+ \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} & -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} \end{bmatrix} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix}$$

Κάνουμε χρήση των εξισώσεων του ευθέως κινηματικού προβλήματος, που αναφέραμε στο κεφάλαιο **III**. Με βάση την **κινηματική του οχήματος**, μπορούμε να υπολογίσουμε την κίνηση. Στη συνέχεια μπορούμε να εμφανίσουμε σε διάγραμμα το πως θα κινηθεί το ρομπότ, χρησιμοποιώντας συγκεκριμένες ταχύτητες. Οι επιθυμητές κινήσεις υπολογίστηκαν με την **σχέση 3.18** (σελ. 40), εφαρμόζοντας κάθε φορά διαφορετικές γωνιακές ταχύτητες.

- Κάνοντας όλους του υπολογισμούς για τις πιθανές κινήσεις του οχήματος διαπιστώθηκε πως αν όλοι οι τροχοί έχουν ίδια ταχύτητα τότε το όχημα θα κινηθεί θετικά ή αρνητικά ως προς τον άξονα X.
- Αν οι τροχοί από την αριστερή μεριά του οχήματος έχουν θετική ταχύτητα και οι τροχοί από την δεξιά πλευρά έχουν ίδια ταχύτητα αλλά αντίθετης φοράς, τότε το όχημα θα κάνει περιστροφή ως προς τον εαυτό του είτε δεξιόστροφα, είτε αριστερόστροφα ανάλογα με την φορά περιστροφής.
- Αν οι δύο τροχοί στις αντίθετες διαμετρικά θέσεις, έχουν ίδια ταχύτητα περιστροφής και οι άλλοι 2 τροχοί έχουν ίδια ταχύτητα αντίθετης φοράς, τότε το όχημα θα μετακινηθεί ως προς τον άξονα Y. Θα κάνει δηλαδή ολίσθηση είτε δεξιά είτε αριστερά.

Ολοκληρώνοντας όλες τις πράξεις, με τον τρόπο που είδαμε στην σελίδα 40, καταλήγουμε στον **πίνακα 5.1**. Ο πίνακας αναπαριστά την τελική κίνηση του οχήματος, εάν σε αυτό εφαρμόσουμε συγκεκριμένες ταχύτητες στους ενεργοποιητές από την κάθε ρόδα.

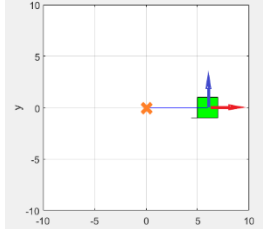
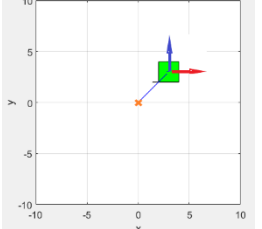
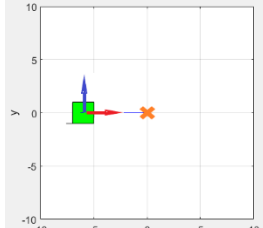
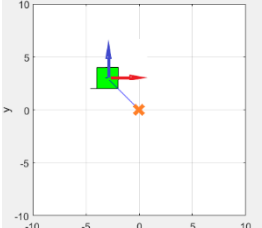
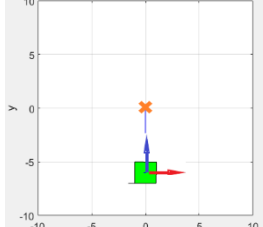
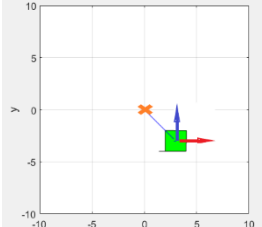
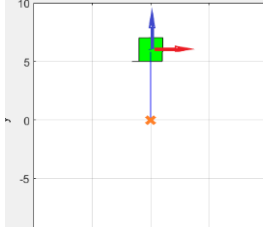
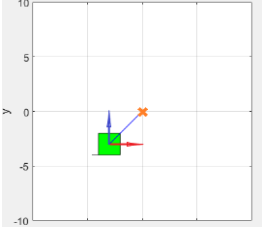
Πίνακας 5.1 – Κίνηση οχήματος ανάλογα με τις ταχύτητες του κάθε τροχού.

Κατεύθυνση	Vx	Vy	$\Omega$	Ρόδα 1	Ρόδα 2	Ρόδα 3	Ρόδα 4
<b>Μπροστά</b>	<b>x</b>	<b>0</b>	<b>0</b>	+	+	+	+
<b>Πίσω</b>	<b>-x</b>	<b>0</b>	<b>0</b>	-	-	-	-
<b>Αριστερά</b>	<b>0</b>	<b>y</b>	<b>0</b>	-	+	+	-
<b>Δεξιά</b>	<b>0</b>	<b>-y</b>	<b>0</b>	+	-	-	+
<b>Αριστερά μπροστά διαγώνια</b>	<b>x</b>	<b>y</b>	<b>0</b>	<b>0</b>	+	+	<b>0</b>
<b>Αριστερά πίσω διαγώνια</b>	<b>-x</b>	<b>y</b>	<b>0</b>	-	<b>0</b>	<b>0</b>	-
<b>Δεξιά μπροστά διαγώνια</b>	<b>x</b>	<b>-y</b>	<b>0</b>	+	<b>0</b>	<b>0</b>	+
<b>Δεξιά πίσω διαγώνια</b>	<b>-x</b>	<b>-y</b>	<b>0</b>	<b>0</b>	-	-	<b>0</b>

Για να μπορέσουμε να κάνουμε αναπαράσταση της κίνησης του οχήματος δημιουργήσαμε ένα σύντομο κώδικα στο πρόγραμμα Matlab. Δημιουργήσαμε και ένα plot στο οποίο θα σχεδιάζεται η κίνηση του οχήματος ώστε με αυτόν τον τρόπο να έχουμε μια προσομοίωση και να μπορέσουμε να συγκρίνουμε τα αποτελέσματα που είχαμε στον **πίνακα 5.1** με τα αποτελέσματα που θα έχουμε από την προσομοίωση.

Εφαρμόζοντας λοιπόν στο Matlab τις ταχύτητες που υποδείξαμε στον **πίνακα 5.1** παρατηρούμε πως στο θεωρητικό μέρος της εργασίας, ισχύουν οι μαθηματικές σχέσεις για την κίνηση. Στον **πίνακα 5.2** (παρακάτω σελίδα), μπορούμε να δούμε πως τα αποτελέσματα που είχαμε από την προσομοίωση είναι ίδια με του **πίνακα 5.1**, επομένως μπορούμε να πούμε πως οι σχέσεις που αναφέραμε στο **κεφάλαιο III** είναι σωστές και εκφράζουν το κινηματικό μοντέλο του οχήματος.

Πίνακας 5.2 – Αποτύπωση των αποτελεσμάτων της προσομοίωσης στον πρόγραμμα MATLAB

Μπροστά		$V_x = 0.2$ $V_y = 0$ $\omega = 0$	Αριστερά μπροστά διαγώνια		$V_x = 0.1$ $V_y = 0.1$ $\omega = 0$
Πίσω		$V_x = -0.2$ $V_y = 0$ $\omega = 0$	Αριστερά πίσω διαγώνια		$V_x = -0.1$ $V_y = 0.1$ $\omega = 0$
Δεξιά		$V_x = 0$ $V_y = -0.2$ $\omega = 0$	Δεξιά μπροστά διαγώνια		$V_x = 0.1$ $V_y = -0.1$ $\omega = 0$
Αριστερά		$V_x = 0$ $V_y = 0.2$ $\omega = 0$	Δεξιά πίσω διαγώνια		$V_x = -0.1$ $V_y = 0.1$ $\omega = 0$



## 5.5 Προγραμματισμός Server - Client

Η δυνατότητα σύνδεσης και κοινής χρήσης δεδομένων σε πολλαπλές συσκευές μέσω δικτύου είναι ζωτικής σημασίας για ένα ευρύ φάσμα εφαρμογών, από διαδικτυακές υπηρεσίες έως συσκευές IoT και καταναμημένα συστήματα. Παρέχει ισχυρές δυνατότητες για την κατασκευή αλληλεπιδράσεων τύπου Server-Client, επιτρέποντας στους υπολογιστές να επικοινωνούν μεταξύ τους.

Η Python περιλαμβάνει μια ενότητα socket που επιτρέπει τον προγραμματισμό δικτύου. Τα Sockets είναι τα βασικά δομικά στοιχεία της δικτυακής επικοινωνίας, επιτρέποντας σε διεργασίες σε ξεχωριστές συσκευές να επικοινωνούν και να ανταλλάσσουν δεδομένα. Τα sockets, δημιουργούν συνδέσεις, παρέχοντας χαρακτηριστικά από την πλευρά του Client και του Server. Ενώ ο Client ξεκινά τη σύνδεση με τον Server, ο Server χρησιμεύει ως ακροατής - δέκτης, περιμένοντας να λάβει εισερχόμενες συνδέσεις.

Υπάρχουν πολλές κρίσιμες φάσεις στη διαδικασία επικοινωνίας Server-Client. Πρώτον, ο Server χρησιμοποιεί τη μέθοδο socket() για να κατασκευάσει ένα socket, δίνοντας τις διευθύνσεις (IPv4 ή IPv6) και το πρωτόκολλο υποδοχής (TCP ή UDP). Στη συνέχεια, χρησιμοποιώντας τη μέθοδο bind(), ο Server συνδέει το socket σε μια συγκεκριμένη διεύθυνση και θύρα. Αφού συνδεθεί, ο Server χρησιμοποιεί τη μέθοδο listen() για να ακούσει για εισερχόμενες συνδέσεις, με τον μέγιστο αριθμό συνδέσεων στην ουρά να καθορίζεται.

Ένα Socket διαμορφώνεται με παρόμοιο τρόπο στην πλευρά του Client χρησιμοποιώντας την socket(). Ο Client ξεκινά τη σύνδεση χρησιμοποιώντας τη μέθοδο connect() και περνώντας τη διεύθυνση και τον αριθμό θύρας του Server. Μετά τη δημιουργία μιας επιτυχημένης σύνδεσης, τόσο ο Server όσο και ο Client μπορούν να ανταλλάσσουν δεδομένα χρησιμοποιώντας τις μεθόδους send() και recv().

Έχοντας συμβατότητα σε πολλαπλές πλατφόρμες τα socket της είναι προσβάσιμα και με αυτόν τον τρόπο επιτρέπουν τη δημιουργία συνδέσεων Server-Client μεταξύ διαφορετικών λειτουργικών συστημάτων. Τα socket της Python υποστηρίζουν τόσο τα πρωτόκολλα TCP όσο και τα πρωτόκολλα UDP, επιτρέποντας στους προγραμματιστές να επιλέξουν το κατάλληλο πρωτόκολλο ανάλογα με τις ανάγκες της εφαρμογής τους.

### 5.5.1 Δημιουργία Server-Client και GUI (Graphical user interface)

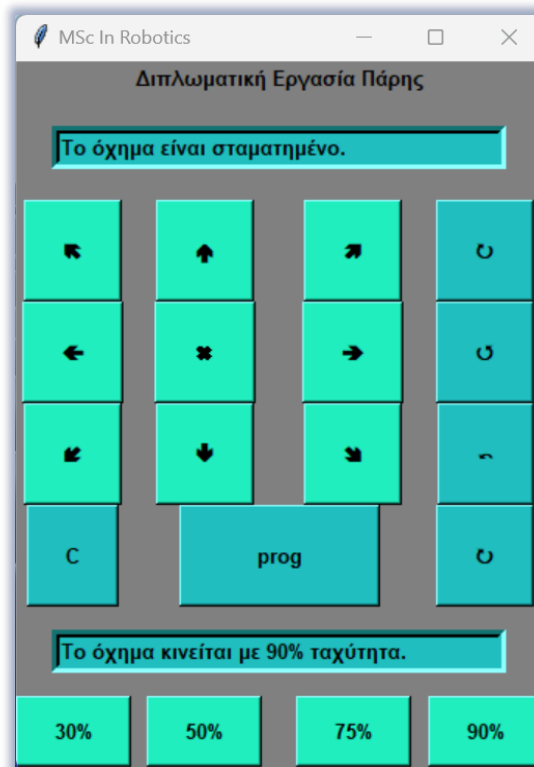
Η συγκεκριμένη διπλωματική εργασία έχει ως σκοπό την δημιουργία ενός οχήματος το οποίο να μπορεί να ελέγχεται μέσω του τοπικού δικτύου το οποίο βρίσκεται. Με την βοήθεια των δυνατοτήτων της επικοινωνίας Server-Client δημιουργήσαμε ένα GUI με την βιβλιοθήκη **Tkinter** της Python, το



οποίο έχει την δυνατότητα να ελέγχει απομακρυσμένα το όχημα μας ακριβώς με τον τρόπο που επιθυμούμε. Στο **σχήμα 5.7** μπορούμε να δούμε το GUI που δημιουργήσαμε.

Το GUI έχει σχεδιαστεί έτσι ώστε να κινείται με όλους σχεδόν τους δυνατούς τρόπους. Μόλις ανοίξει η εφαρμογή και έχουμε σύνδεση με το όχημα μπορούμε να το κατευθύνουμε είτε κάνοντας κλικ με το ποντίκι πάνω στο πλήκτρο που έχουμε αναθέσει στην κάθε κίνηση. Μπορούμε επίσης να το κατευθύνουμε με την βοήθεια του πληκτρολογίου μιας και έχει γίνει ανάθεση κάποιων πλήκτρων του πληκτρολογίου, ώστε να κάνει την κίνηση που επιθυμούμε.

Στο **σχήμα 5.8** απεικονίζονται οι αναθέσεις των κινήσεων στα αντίστοιχα πλήκτρα του πληκτρολογίου.



**Σχήμα 5.7 – Graphical user interface που αναπτύχθηκε με την βιβλιοθήκη Tkinter της Python.**



**Σχήμα 5.8 – Αντιστοιχίες κινήσεων στα πλήκτρα του πληκτρολογίου**

Πέρα από την κατεύθυνση του οχήματος μπορούμε να ελέγξουμε και την ταχύτητα περιστροφής των τροχών του αλλάζοντας το Duty Cycle από το PWM signal του Raspberry Pi. Η προκαθορισμένη ταχύτητα εκκίνησης είναι στο 90% και μπορεί να αλλαχτεί σε 30%, 50%, 70% και ξανά στα 90%.

Πατώντας τα πλήκτρα που επιθυμούμε για να κινηθεί το όχημα η εφαρμογή αλλάζει χρώμα στα συγκεκριμένα πλήκτρα που πατάμε και τα κάνει κίτρινα για όση ώρα είναι πατημένα. Για όσο χρονικό διάστημα πατιέται ένα πλήκτρο, για τόσο χρονικό διάστημα το όχημα κινείται προς την κατεύθυνση εκείνη. Η κίνηση σταματάει και το όχημα ακινητοποιείται μόλις αφήσουμε το κουμπί που πατούσαμε.

Η συγκεκριμένη διπλωματική εργασία έχει ως σκοπό την δημιουργία ενός οχήματος το οποίο να μπορεί να ελέγχεται μέσω του τοπικού δικτύου το οποίο βρίσκεται. Με την βοήθεια των δυνατοτήτων της επικοινωνίας Server-Client δημιουργήσαμε μια σύνδεση τέτοιας μορφής, εφαρμόζοντας κανονικά όσα αναλύσαμε παραπάνω.

Ακολουθώντας όσα διδαχτήκαμε δημιουργήσαμε μία σύνδεση Server-Client. Στο **σχήμα 5.9** παρουσιάζονται οι αρχικές γραμμές του κώδικα μας στο οποίο ακολουθώντας όσα διδαχτήκαμε, δημιουργούμε το παράθυρο του Gui καθώς και μία σύνδεση Server-Client η οποία ανοίγει ένα Socket και όπως μπορούμε να δούμε, "ακούει" και περιμένει μια μόνο συσκευή και όχι παραπάνω. Η λογική της επικοινωνίας είναι αρκετά απλή. Το κάθε πλήκτρο που ορίσαμε για τις κινήσεις του οχήματος έχει ένα χαρακτήρα σαν μεταβλητή.

```
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('0.0.0.0', 8011))
s.listen(1)

root = Tk()
root.geometry("340x460")
root.configure(bg='gray')
root.title("MSc In Robotics")
myFont = font.Font(family='Helvetica', size=10, weight='bold')

e = Entry(root, width=40, borderwidth=5,bg="#20bebe",fg="black")
e.grid(row=1, column=0, columnspan=4, padx=20, pady=20)
e1 = Entry(root, width=40, borderwidth=5,bg="#20bebe",fg="black")
e1.grid(row=6, column=0, columnspan=4, padx=15, pady=15)

clientsocket, address = s.accept()

e.delete(0,END)
e.insert(0, "Πραγματοποιήθηκε σύνδεση με το όχημα!")
e1.delete(0,END)
e1.insert(0, "Το όχημα κινείται με 90% ταχύτητα.")
```

**Σχήμα 5.9 – Δημιουργία σύνδεσης Client-Server στον υπολογιστή**

Ανάλογα με το ποιο πλήκτρο πατάμε, φορτώνουμε την αντίστοιχη μεταβλητή και την στέλνουμε μέσω του socket στο Client (στο όχημα, στην συγκεκριμένη περίπτωση). Η κάθε μεταβλητή που θα φτάσει μέσω της σύνδεσης, σημαίνει και μια συγκεκριμένη πορεία, οπότε μετά στον κώδικα του οχήματος ορίζουμε τι πρέπει να κάνει το όχημα.

Για παράδειγμα για την κίνηση προς τα εμπρός, μόλις πατηθεί το πλήκτρο αλλάζει χρώμα το κουμπί στο GUI και γίνεται κίτρινο. Εμφανίζεται το μήνυμα πως το όχημα κινείται προς τα εμπρός και μέσω του socket στέλνουμε την μεταβλητή 'x' η οποία ισούται με 'w'. Από την μεριά του Client μόλις διαβάσει την μεταβλητή που στάλθηκε θα κάνει την αντίστοιχη κίνηση που ορίσαμε να κάνει αν διαβάσει w. Επομένως θα γυρίσουν όλοι οι τροχοί προς τα εμπρός και το όχημα θα κινηθεί μπροστά. Στο **σχήμα 5.10** απεικονίζεται το κομμάτι του κώδικα που είναι υπεύθυνο για το πλήκτρο 'w'.

```
def mpros(event):
    x='w'
    MPROS.configure(bg='yellow')
    e.delete(0,END)
    e.insert(0, "Το όχημα κινείται προς τα εμπρος.")
    clientsocket.send(bytes(x, "utf-8"))
```

**Σχήμα 5.10** – Συνάρτηση υπεύθυνη για το πλήκτρο 'w', μεριά Server (Υπολογιστής)

```
elif x=='w':
    print("Mrosta")
    GPIO.output(RightFront1,GPIO.HIGH)
    GPIO.output(RightFront2,GPIO.LOW)
    GPIO.output(LeftFront1,GPIO.HIGH)
    GPIO.output(LeftFront2,GPIO.LOW)
    GPIO.output(RightBack1,GPIO.HIGH)
    GPIO.output(RightBack2,GPIO.LOW)
    GPIO.output(LeftBack2,GPIO.HIGH)
    GPIO.output(LeftBack1,GPIO.LOW)
```

**Σχήμα 5.11** – Συνάρτηση υπεύθυνη για το πλήκτρο 'w', μεριά Client (Οχήματος)

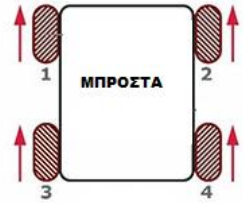
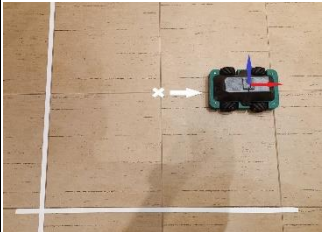
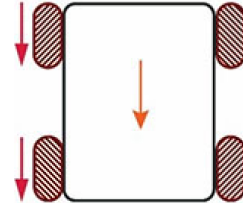
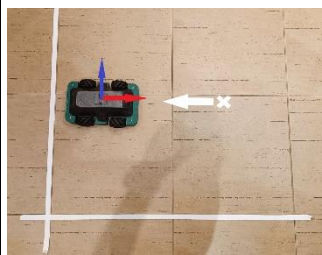
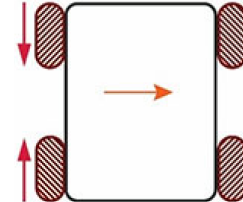
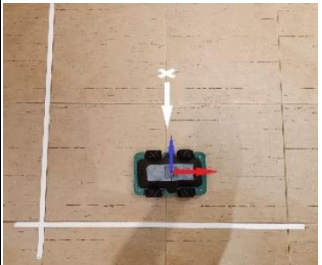
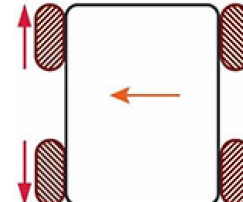
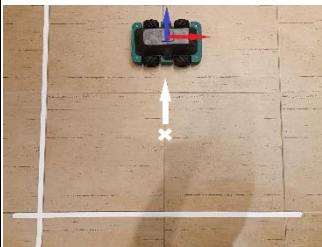
Στο **σχήμα 5.11** αναγράφεται ο κώδικας από το μέρος του Client. Δηλαδή, μόλις λάβει το 'w' θα ορίσει σε όλους τους τροχούς να έχουν ίδια ταχύτητα και ίδια φορά, επομένως το όχημα θα κινηθεί σε ευθεία.

## 5.6 Πειραματικές δοκιμές.

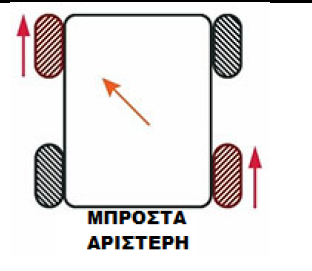
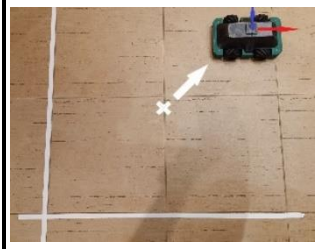
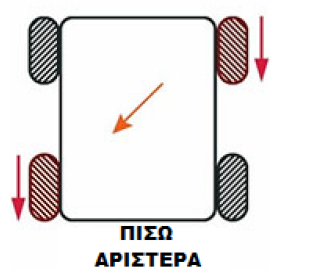
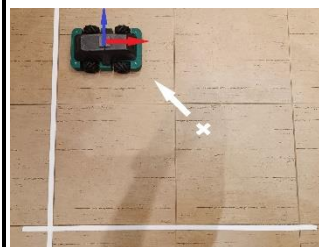
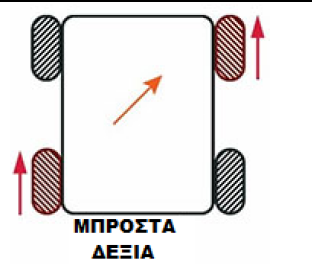
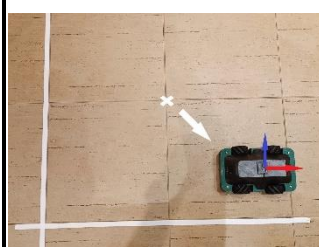
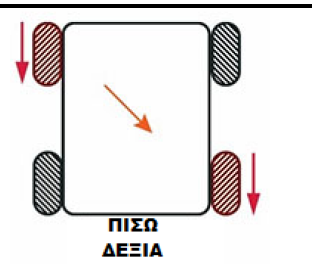
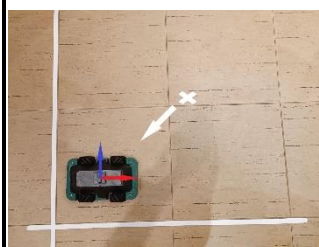
Έχοντας ολοκληρώσει τόσο το θεωρητικό μέρος όσο και το κατασκευαστικό κομμάτι της διπλωματικής εργασίας, μπορούμε να κάνουμε και μελέτη της κινηματικής του οχήματος, σε πρακτικό πείραμα. Για να μπορέσουμε να διαπιστώσουμε αν όλα όσα μελετήσαμε και αναφέραμε κατά την διάρκεια των προηγούμενων κεφαλαίων έχουν σωστά αποτελέσματα, δημιουργήσαμε έναν κώδικα ο οποίος θα εφαρμόζει τις ταχύτητες του κάθε τροχού σύμφωνα με τους υπολογισμούς της παραγράφου **5.4**. Αν, για παράδειγμα, θέλουμε να κινηθεί το όχημα μπροστά, θα πρέπει όλοι οι τροχοί να γυρνάνε με την ίδια ταχύτητα και να έχουν όλοι τους ίδια φορά προς τα εμπρός. Αν επιθυμούμε το όχημα να κινηθεί προς τα πίσω, τότε θα πρέπει οι τροχοί να έχουν την ίδια ταχύτητα όλοι μεταξύ τους, αλλά αντίθετη φορά σε σχέση με την κίνηση προς τα μπροστά.

Στους **πίνακες 5.3** και **5.4** μπορούμε να δούμε τα αποτελέσματα από όλες κατευθύνσεις που κινήθηκε το όχημα, όταν εφαρμόσαμε στους τροχούς του τις ταχύτητες που αναφέραμε. Όπως μπορούμε να παρατηρήσουμε από τους πίνακες, το όχημα πραγματικά κινείται με τον τρόπο που υπολογίσαμε και προβλέψαμε.

Πίνακας 5.3 – Αποτύπωση των αποτελεσμάτων του πειραματικού σταδίου μέρος Α

Κατεύθυνση	Κίνηση τροχών		Αναπαράσταση κίνησης οχήματος	
Μπροστά				$V_x = x$ $V_y = 0$ $\omega = 0$
Πίσω				$V_x = -x$ $V_y = 0$ $\omega = 0$
Δεξιά				$V_x = 0$ $V_y = -y$ $\omega = 0$
Αριστερά				$V_x = 0$ $V_y = y$ $\omega = 0$

Πίνακας 5.4 – Αποτύπωση των αποτελεσμάτων του πειραματικού σταδίου μέρος Β

Κατεύθυνση	Κίνηση τροχών		Αναπαράσταση κίνησης οχήματος	
Αριστερά μπροστά διαγώνια	 <p>ΜΠΡΟΣΤΑ ΑΡΙΣΤΕΡΗ</p>			$V_x = x$ $V_y = y$ $\omega = 0$
Αριστερά πίσω διαγώνια	 <p>ΠΙΣΩ ΑΡΙΣΤΕΡΑ</p>			$V_x = -x$ $V_y = y$ $\omega = 0$
Δεξιά μπροστά διαγώνια	 <p>ΜΠΡΟΣΤΑ ΔΕΞΙΑ</p>			$V_x = x$ $V_y = -y$ $\omega = 0$
Δεξιά πίσω διαγώνια	 <p>ΠΙΣΩ ΔΕΞΙΑ</p>			$V_x = -x$ $V_y = -y$ $\omega = 0$

# Κεφάλαιο VI. Συμπέρασμα και βελτιώσεις.

## 6.1 Συμπεράσματα.

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκε η μελέτη των οχημάτων με τέσσερις σουηδικούς τροχούς, με σκοπό την κατασκευή ενός ρομποτικού οχήματος το οποίο μπορεί να κινείται προς όλες τις κατευθύνσεις, χωρίς κάποιους περιορισμούς, να κινείται δηλαδή πανκατευθυντικά. Επιπλέον, μελετήθηκε η κινηματική του οχήματος τόσο από θεωρητική άποψη, όσο και από πρακτική άποψη.

Πιο αναλυτικά, όσον αφορά το θεωρητικό μέρος, παρουσιάστηκε το κινηματικό μοντέλο λύνοντας το ευθύ και το αντίστροφο κινηματικό πρόβλημα. Τα εργαλεία που χρησιμοποιούνται είναι η θεωρία της κίνησης των διάφορων τύπων τροχού και συγκεκριμένα του σουηδικού τροχού.

Στη συνέχεια, μελετήθηκαν οι κινήσεις που μπορεί να πραγματοποιηθούν με το συγκεκριμένο ρομποτικό μοντέλο κάνοντας προσομοιώσεις μέσω υπολογιστικών προγραμμάτων, αφού πρώτα αποδείχτηκαν μαθηματικά όλες οι απαραίτητες εξισώσεις κίνησης. Μόλις ολοκληρώθηκε η κατασκευή του οχήματος, πραγματοποιήθηκαν και πρακτικές μελέτες, ώστε να μπορέσουμε να επιβεβαιώσουμε τα στοιχεία που είχαμε από το θεωρητικό μέρος και να τα κάνουμε σύγκριση μεταξύ τους.

Όλα τα αποτελέσματα που είχαμε και από τις 3 διαφορετικές μελέτες υποδεικνύουν πως η μελέτη και το πείραμα μας ολοκληρώθηκε με επιτυχία, καθώς όλα συμφωνούσαν μεταξύ τους. Η σημαντικότερη μελέτη είναι σίγουρα αυτή του κατασκευαστικού μέρους, καθώς ολοκληρώθηκε με μεγάλη επιτυχία η κατασκευή και λειτουργεί σωστά. Αυτό μας αποδεικνύει ότι το μεταπτυχιακό πρόγραμμα της ρομποτικής μας έδωσε τα κατάλληλα εφόδια και τις απαραίτητες γνώσεις ώστε να φέρουμε εις πέρας την μελέτη του προβλήματος που μας ανατέθηκε.

## 6.2 Μελλοντικές βελτιώσεις.

Το σύστημα που μελετήθηκε και αναπτύχθηκε στα πλαίσια της παρούσας εργασίας δίνει δυνατότητες για περαιτέρω διερεύνηση ως προς τα ακόλουθα μέρη:

- Ενσωμάτωση αισθητήρων για καλύτερο έλεγχο τροχών έχοντας feedback από ταχύτητα περιστροφή τους, ώστε να γυρνάνε πραγματικά με την ίδια ταχύτητα.
- Προσθήκη αισθητήρων για να μπορέσει να κινείται αυτόνομα το όχημα, καθώς και για να παρέχεται καλύτερος έλεγχος του οχήματος.
- Τοποθέτηση κάμερας και ενσωμάτωση της εικόνας στο GUI ώστε να μπορεί να ελέγχεται το όχημα χωρίς να υπάρχει με αυτό οπτική επαφή.
- Δημιουργία καλύτερου και πιο σωστού κώδικα ώστε να μην εμφανίζονται διάφορα bugs.



# Βιβλιογραφία

- [1] Gregor Klancar, Andrej Zdesar, Saso Blazic, Igor Skrjanc, “Wheeled Mobile Robotics – From Fundamentals Towards Autonomous Systems”. 2017.
- [2] Roland Siegwart, Illah R. Nourbakhsh, Davide Scaramuzza , “Introduction to Autonomous Mobile Robots”, 2004.
- [3] Thomas Bräunl, “Embedded Robotics - Mobile Robot Design and Applications with Embedded Systems”, 2003.
- [4] Mihai Olimpiu Tătar, Cătălin Popovici, Dan Mândru, Ioan Ardelean, Alin Pleșa, “Design and development of an autonomous Omni-Directional Mobile robot with Mecanum wheels”. 2014.
- [5] Qian Jia, Mulan Wang, Shuqing Liu, Jianjing Ge, Chen Gu. “Research and Development of Mecanum-wheeled Omnidirectional Mobile Robot Implemented by Multiple Control Methods.”, 2016.
- [6] Yunwang Li, Sumei Dai, Lala Zhao, Xucong Yan, Yong Shi. “Topological Design Methods for Mecanum Wheel Configurations of an Omnidirectional Mobile Robot”, 2019.
- [7] Ιωάννης Καλόμοιρος. “Σημειώσεις μαθήματος: «Αυτόνομα ρομποτικά συστήματα»”. 2021.
- [8] Lih-Chang Lin, Hao-Yin Shih., “Modeling and Adaptive Control of an Omni-Mecanum-Wheeled Robot”, 2013.
- [9] Hamid Taheri, Bing Qiao, Nurallah Ghaeminezhad, “Kinematic Model of a Four Mecanum Wheeled Mobile Robot”, 2015.

[10] Daniil S. Alhanov, Vasily I. Rubtsov, “Development of the Laboratory Work: “Modeling of a Mobile Robot on Mecanum Wheels Kinematics”. 2019.

[11] <https://docs.python.org/3/library/index.html>

[12] <https://www.raspberrypi.org/about/>

[13] Yin Liuliu, Wang Fang, Han Sen, Li Yuchen, Sun Hao, Lu Qingjie, Yang Cheng, Wang Quanzhao, “Application of Drive Circuit Based on L298N in Direct Current Motor Speed Control System” , 2016.

## Παράρτημα Ι. Κώδικας Server (Υπολογιστή)

```
from tkinter import*
import socket
import tkinter.font as font
import time
import subprocess

s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('0.0.0.0', 8001))
s.listen(1)

root = Tk()
root.geometry("550x750")
root.configure(bg='gray')
root.title("MSc In Robotics")
myFont = font.Font(family='Helvetica', size=10, weight='bold')

e = Entry(root, width=40, borderwidth=5,bg="#20bebe",fg="black")
e.grid(row=1, column=0, columnspan=4, padx=20, pady=20)
e1 = Entry(root, width=40, borderwidth=5,bg="#20bebe",fg="black")
e1.grid(row=6, column=0, columnspan=4, padx=15, pady=15)

clientsocket, address = s.accept()

e.delete(0,END)
e.insert(0, "Πραγματοποιήθηκε σύνδεση με το όχημα!")
e1.delete(0,END)
e1.insert(0, "Το όχημα κινείται με 90% ταχύτητα.")
```

```
def mpros(event):  
    x='w'  
    MPROS.configure(bg='yellow')  
    e.delete(0,END)  
    e.insert(0, "Το όχημα κινείται προς τα εμπρος.")  
    clientsocket.send(bytes(x, "utf-8"))
```

```
def piso(event):  
    x='b'  
    PISO.configure(bg='yellow')  
    e.delete(0,END)  
    e.insert(0, "Το όχημα κινείται προς τα πίσω.")  
    clientsocket.send(bytes(x, "utf-8"))
```

```
def stop(event):  
    x='s'  
    MPROS.configure(bg="#20eebe")  
    PISO.configure(bg="#20eebe")  
    DEKSIA.configure(bg="#20eebe")  
    ARISTERA.configure(bg="#20eebe")  
    KDA.configure(bg="#20eebe")  
    KDD.configure(bg="#20eebe")  
    PDA.configure(bg="#20eebe")  
    PDD.configure(bg="#20eebe")  
    GUROA.configure(bg="#20bebe")  
    GUROB.configure(bg="#20bebe")  
    CYCLEA.configure(bg="#20bebe")  
    CYCLED.configure(bg="#20bebe")  
    PROG.configure(bg="#20bebe")  
    CLEAR.configure(bg="#20bebe")  
    SPEED.configure(bg="#20eebe")
```

```
SPEED1.configure(bg="#20eebe")
SPEED2.configure(bg="#20eebe")
SPEED3.configure(bg="#20eebe")
e.delete(0,END)
e.insert(0, "Το όχημα είναι σταματημένο.")
clientsocket.send(bytes(x, "utf-8"))
```

```
def clear(event):
    x='o'
    CLEAR.configure(bg='yellow')
    e.delete(0,END)
    e.insert(0, "Εγινε καθαρισμός του GPIO.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def aristera(event):
    x='a'
    ARISTERA.configure(bg='yellow')
    e.delete(0,END)
    e.insert(0, "Το όχημα κινείται προς τα αριστερά.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def deksia(event):
    x='d'
    DEKSIA.configure(bg='yellow')
    e.delete(0,END)
    e.insert(0, "Το όχημα κινείται προς τα δεξιά.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def pda(event=None):
    x='q'
    PDA.configure(bg='yellow')
```

```
e.delete(0,END)
e.insert(0, "Το όχημα κινείται διαγώνια πάνω αριστερά.")
clientsocket.send(bytes(x, "utf-8"))
```

```
def pdd(event):
    x='e'
    PDD.configure(bg='yellow')
    e.delete(0,END)
    e.insert(0, "Το όχημα κινείται διαγώνια πάνω δεξιά.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def kda(event):
    x='z'
    KDA.configure(bg='yellow')
    e.delete(0,END)
    e.insert(0, "Το όχημα κινείται διαγώνια κάτω αριστερά.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def kdd(event):
    x='c'
    KDD.configure(bg='yellow')
    e.delete(0,END)
    e.insert(0, "Το όχημα κινείται διαγώνια κάτω δεξιά.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def cycled(event):
    x='r'
    CYCLED.configure(bg='yellow')
```

```
e.delete(0,END)
e.insert(0, "Δεξιόστροφο κύκλο.")
clientsocket.send(bytes(x, "utf-8"))
```

```
def cyclea(event):
    x='t'
    CYCLEA.configure(bg='yellow')
    e.delete(0,END)
    e.insert(0, "Αριστερόστροφο κύκλο.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def guroa(event):
    x='f'
    GUROA.configure(bg='yellow')
    e.delete(0,END)
    e.insert(0, "Κύκλο γυρο άξοχα X.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def gurob(event):
    x='g'
    GUROB.configure(bg='yellow')
    e.delete(0,END)
    e.insert(0, "Κύκλο γύρω άξονα Y.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def prog(event):
    e.delete(0,END)
    e.insert(0, "Εκτελείται πρόγραμμα παρουσίασης.")
    x='w'
    PROG.configure(bg='yellow')
    clientsocket.send(bytes(x, "utf-8"))
```

```
time.sleep(2)
x='s'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(0.2)
x='b'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(2)
x='s'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(0.2)
x='a'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(2)
x='s'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(0.2)
x='d'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(2)
x='s'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(0.2)
x='q'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(2)
x='s'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(0.2)
x='c'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(2)
x='s'
```



```
clientsocket.send(bytes(x, "utf-8"))
time.sleep(0.2)
x='e'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(2)
x='s'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(0.2)
x='z'
clientsocket.send(bytes(x, "utf-8"))
time.sleep(2)
x='s'
clientsocket.send(bytes(x, "utf-8"))
```

```
def speed(event):
    x='m'
    SPEED.configure(bg='yellow')
    e1.delete(0,END)
    e1.insert(0, "Το όχημα κινείται με 30% ταχύτητα.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def speed1(event):
    x='n'
    SPEED1.configure(bg='yellow')
    e1.delete(0,END)
    e1.insert(0, "Το όχημα κινείται με 50% ταχύτητα.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
def speed2(event):
    x='k'
    SPEED2.configure(bg='yellow')
    e1.delete(0,END)
```

```
e1.insert(0, "Το όχημα κινείται με 75% ταχύτητα.")
clientsocket.send(bytes(x, "utf-8"))
```

```
def speed3(event):
    x='v'
    SPEED3.configure(bg='yellow')
    e1.delete(0,END)
    e1.insert(0, "Το όχημα κινείται με 90% ταχύτητα.")
    clientsocket.send(bytes(x, "utf-8"))
```

```
myLabel = Label(root,text="Διπλωματική Εργασία Πάρης",bg=('gray'))
```

```
PDA = Button(root, text="P", pady=20, padx=20
,bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<q>",pda)
PDA.bind("<ButtonPress-1>",pda)
```

```
MPROS = Button(root, text="W", pady=20, padx=20
,bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<w>",mpros)
MPROS.bind("<ButtonPress-1>",mpros)
```

```
PDD = Button(root, text="E", pady=20, padx=20
,bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<e>",pdd)
PDD.bind("<ButtonPress-1>",pdd)
```

```
ARISTERA = Button(root, text="A", pady=20, padx=20
,bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<a>",aristera)
ARISTERA.bind("<ButtonPress-1>",aristera)
```

```
DEKSIA = Button(root, text="☒", pady=20, padx=20,
bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<d>",deksia)
DEKSIA.bind("<ButtonPress-1>",deksia)
```

```
STOP = Button(root, text="✕", pady=20, padx=20
,bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<KeyRelease>",stop)
root.bind("<ButtonRelease-1>",stop)
```

```
KDA = Button(root, text="☒", pady=20, padx=20
,bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<z>",kda)
KDA.bind("<ButtonPress-1>",kda)
```

```
PISO = Button(root, text="☒", pady=20, padx=20
,bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<x>",piso)
PISO.bind("<ButtonPress-1>",piso)
```

```
KDD = Button(root, text="☒", pady=20, padx=20
,bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<c>",kdd)
KDD.bind("<ButtonPress-1>",kdd)
```

```
CLEAR = Button(root, text="C", pady=20, padx=20
,bg="#20bebe",fg="black",activebackground="yellow")
root.bind("<o>",clear)
CLEAR.bind("<ButtonPress-1>",clear)
```

```
CYCLED = Button(root, text="Ϸ", pady=20, padx=20
, bg="#20bebe", fg="black", activebackground="yellow")
root.bind("<t>", cycled)
CYCLED.bind("<ButtonPress-1>", cycled)
```

```
CYCLEA = Button(root, text="ϸ", pady=20, padx=20
, bg="#20bebe", fg="black", activebackground="yellow")
root.bind("<r>", cyclea)
CYCLEA.bind("<ButtonPress-1>", cyclea)
```

```
GUROA = Button(root, text="Ϲ", pady=20, padx=20
, bg="#20bebe", fg="black", activebackground="yellow")
root.bind("<f>", guroa)
GUROA.bind("<ButtonPress-1>", guroa)
```

```
GUROB = Button(root, text="Ϻ", pady=20, padx=20
, bg="#20bebe", fg="black", activebackground="yellow")
root.bind("<g>", gurob)
GUROB.bind("<ButtonPress-1>", gurob)
```

```
PROG = Button(root, text="prog", pady=20,
padx=45, bg="#20bebe", fg="black", activebackground="yellow")
root.bind("<m>", prog)
PROG.bind("<ButtonPress-1>", prog)
```

```
SPEED = Button(root, text="30%", pady=10, padx=20
, bg="#20eebe", fg="black", activebackground="yellow")
root.bind("<Key-1>", speed)
SPEED.bind("<ButtonPress-1>", speed)
```

```
SPEED1 = Button(root, text="50%", pady=10, padx=20
, bg="#20eebe", fg="black", activebackground="yellow")
```

```

root.bind("<Key-2>",speed1)
SPEED1.bind("<ButtonPress-1>",speed1)

SPEED2 = Button(root, text="75%", pady=10, padx=20
,bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<Key-3>",speed2)
SPEED2.bind("<ButtonPress-1>",speed2)

SPEED3 = Button(root, text="90%", pady=10, padx=20
,bg="#20eebe",fg="black",activebackground="yellow")
root.bind("<Key-4>",speed3)
SPEED3.bind("<ButtonPress-1>",speed3)

if prog==True:
    prog

myLabel['font'] = myFont
PDA['font'] = myFont
MPROS['font'] = myFont
PDD['font'] = myFont
ARISTERA['font'] = myFont
DEKSIA['font'] = myFont
STOP['font'] = myFont
KDA['font'] = myFont
PISO['font'] = myFont
KDD['font'] = myFont
CLEAR['font'] = myFont
CYCLED['font'] = myFont
CYCLEA['font'] = myFont
e['font'] = myFont
e1['font'] = myFont

```

```
GUROA['font'] = myFont
GUROB['font'] = myFont
PROG['font'] = myFont
SPEED['font'] = myFont
SPEED1['font'] = myFont
SPEED2['font'] = myFont
SPEED3['font'] = myFont
```

```
myLabel.grid(row=0,column=1,columnspan=2)
PDA.grid(row=2,column=0)
MPROS.grid(row=2,column=1)
PDD.grid(row=2,column=2)
CYCLEA.grid(row=2,column=3)
ARISTERA.grid(row=3,column=0)
DEKSIA.grid(row=3,column=2)
STOP.grid(row=3,column=1)
CYCLED.grid(row=3,column=3)
KDA.grid(row=4,column=0)
PISO.grid(row=4,column=1)
KDD.grid(row=4,column=2)
CLEAR.grid(row=5,column=0)
PROG.grid(row=5,column=1,columnspan=2)
GUROA.grid(row=4,column=3)
GUROB.grid(row=5,column=3)
SPEED.grid(row=7,column=0)
SPEED1.grid(row=7,column=1)
SPEED2.grid(row=7,column=2)
SPEED3.grid(row=7,column=3)
```

```
root.mainloop()
```

## Παράρτημα II. Κώδικας Client (Οχήματος)

```
import RPi.GPIO as GPIO
from time import sleep

import socket

s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.1.15', 8001))

RightFront1 = 7
RightFront2 = 8
en = 12
LeftFront1= 13
LeftFront2= 19
en2=26

RightBack1 = 17
RightBack2 = 4
en3 = 18

LeftBack1= 6
LeftBack2= 5
en4=2

GPIO.setmode(GPIO.BCM)
GPIO.setup(RightFront1,GPIO.OUT)
GPIO.setup(RightFront2,GPIO.OUT)
GPIO.setup(en,GPIO.OUT)

GPIO.setup(LeftFront1,GPIO.OUT)
```

```
GPIO.setup(LeftFront2,GPIO.OUT)
GPIO.setup(en2,GPIO.OUT)
```

```
GPIO.setup(RightBack1,GPIO.OUT)
GPIO.setup(RightBack2,GPIO.OUT)
GPIO.setup(en3,GPIO.OUT)
```

```
GPIO.setup(LeftBack1,GPIO.OUT)
GPIO.setup(LeftBack2,GPIO.OUT)
GPIO.setup(en4,GPIO.OUT)
```

```
GPIO.output(RightFront1,GPIO.LOW)
GPIO.output(RightFront2,GPIO.LOW)
GPIO.output(LeftFront1,GPIO.LOW)
GPIO.output(LeftFront2,GPIO.LOW)
```

```
GPIO.output(RightBack1,GPIO.LOW)
GPIO.output(RightBack2,GPIO.LOW)
GPIO.output(LeftBack1,GPIO.LOW)
GPIO.output(LeftBack2,GPIO.LOW)
```

```
p=GPIO.PWM(en,1000)
q=GPIO.PWM(en2,1000)
o=GPIO.PWM(en3,1000)
i=GPIO.PWM(en4,1000)
```

```
p.start(90)
q.start(90)
o.start(90)
i.start(90)
```



```
while(1):
```

```
    msg=s.recv(1024)
```

```
    x=(msg.decode("utf-8"))
```

```
    if x=='s':
```

```
        print("stop")
```

```
        GPIO.output(RightFront1,GPIO.LOW)
```

```
        GPIO.output(RightFront2,GPIO.LOW)
```

```
        GPIO.output(LeftFront1,GPIO.LOW)
```

```
        GPIO.output(LeftFront2,GPIO.LOW)
```

```
        GPIO.output(RightBack1,GPIO.LOW)
```

```
        GPIO.output(RightBack2,GPIO.LOW)
```

```
        GPIO.output(LeftBack2,GPIO.LOW)
```

```
        GPIO.output(LeftBack1,GPIO.LOW)
```

```
    elif x=='w':
```

```
        print("Mrosta")
```

```
        GPIO.output(RightFront1,GPIO.HIGH)
```

```
        GPIO.output(RightFront2,GPIO.LOW)
```

```
        GPIO.output(LeftFront1,GPIO.HIGH)
```

```
        GPIO.output(LeftFront2,GPIO.LOW)
```

```
        GPIO.output(RightBack1,GPIO.HIGH)
```

```
        GPIO.output(RightBack2,GPIO.LOW)
```

```
        GPIO.output(LeftBack2,GPIO.HIGH)
```

```
        GPIO.output(LeftBack1,GPIO.LOW)
```

```
    elif x=='b':
```

```
print("Piso")
GPIO.output(RightFront1,GPIO.LOW)
GPIO.output(RightFront2,GPIO.HIGH)
GPIO.output(LeftFront1,GPIO.LOW)
GPIO.output(LeftFront2,GPIO.HIGH)
GPIO.output(RightBack1,GPIO.LOW)
GPIO.output(RightBack2,GPIO.HIGH)
GPIO.output(LeftBack2,GPIO.LOW)
GPIO.output(LeftBack1,GPIO.HIGH)
```

```
elif x=='d':
```

```
print("deksia")
GPIO.output(RightFront1,GPIO.LOW)
GPIO.output(RightFront2,GPIO.HIGH)
GPIO.output(LeftFront1,GPIO.HIGH)
GPIO.output(LeftFront2,GPIO.LOW)
GPIO.output(RightBack1,GPIO.HIGH)
GPIO.output(RightBack2,GPIO.LOW)
GPIO.output(LeftBack2,GPIO.LOW)
GPIO.output(LeftBack1,GPIO.HIGH)
```

```
elif x=='a':
```

```
print("Aristera")
GPIO.output(RightFront1,GPIO.HIGH)
GPIO.output(RightFront2,GPIO.LOW)
GPIO.output(LeftFront1,GPIO.LOW)
GPIO.output(LeftFront2,GPIO.HIGH)
GPIO.output(RightBack1,GPIO.LOW)
GPIO.output(RightBack2,GPIO.HIGH)
GPIO.output(LeftBack2,GPIO.HIGH)
```

```
GPIO.output(LeftBack1,GPIO.LOW)
```

```
elif x=='r':
```

```
    print("Deksiostrofa")
    GPIO.output(RightFront1,GPIO.HIGH)
    GPIO.output(RightFront2,GPIO.LOW)
    GPIO.output(LeftFront2,GPIO.HIGH)
    GPIO.output(LeftFront1,GPIO.LOW)
    GPIO.output(RightBack1,GPIO.HIGH)
    GPIO.output(RightBack2,GPIO.LOW)
    GPIO.output(LeftBack1,GPIO.HIGH)
    GPIO.output(LeftBack2,GPIO.LOW)
```

```
elif x=='t':
```

```
    print("Aristerostrofa")
    GPIO.output(RightFront1,GPIO.LOW)
    GPIO.output(RightFront2,GPIO.HIGH)
    GPIO.output(LeftFront2,GPIO.LOW)
    GPIO.output(LeftFront1,GPIO.HIGH)
    GPIO.output(RightBack1,GPIO.LOW)
    GPIO.output(RightBack2,GPIO.HIGH)
    GPIO.output(LeftBack1,GPIO.LOW)
    GPIO.output(LeftBack2,GPIO.HIGH)
```

```
elif x=='q':
```

```
    print("Pano diagonia aristera")
    GPIO.output(RightFront1,GPIO.HIGH)
    GPIO.output(RightFront2,GPIO.LOW)
    GPIO.output(LeftFront1,GPIO.LOW)
    GPIO.output(LeftFront2,GPIO.LOW)
    GPIO.output(RightBack1,GPIO.LOW)
```

```
GPIO.output(RightBack2,GPIO.LOW)
GPIO.output(LeftBack2,GPIO.HIGH)
GPIO.output(LeftBack1,GPIO.LOW)
```

elif x=='e':

```
print("Pano diagonia dekcia")
GPIO.output(RightFront1,GPIO.LOW)
GPIO.output(RightFront2,GPIO.LOW)
GPIO.output(LeftFront1,GPIO.HIGH)
GPIO.output(LeftFront2,GPIO.LOW)
GPIO.output(RightBack1,GPIO.HIGH)
GPIO.output(RightBack2,GPIO.LOW)
GPIO.output(LeftBack2,GPIO.LOW)
GPIO.output(LeftBack1,GPIO.LOW)
```

elif x=='c':

```
print("Kato diagonia aristera")
GPIO.output(RightFront1,GPIO.LOW)
GPIO.output(RightFront2,GPIO.HIGH)
GPIO.output(LeftFront1,GPIO.LOW)
GPIO.output(LeftFront2,GPIO.LOW)
GPIO.output(RightBack1,GPIO.LOW)
GPIO.output(RightBack2,GPIO.LOW)
GPIO.output(LeftBack2,GPIO.LOW)
GPIO.output(LeftBack1,GPIO.HIGH)
```

elif x=='z':

```
print("Kato diagonia dekcia")
GPIO.output(RightFront1,GPIO.LOW)
GPIO.output(RightFront2,GPIO.LOW)
GPIO.output(LeftFront1,GPIO.LOW)
GPIO.output(LeftFront2,GPIO.HIGH)
```

```
GPIO.output(RightBack1,GPIO.LOW)
GPIO.output(RightBack2,GPIO.HIGH)
GPIO.output(LeftBack2,GPIO.LOW)
GPIO.output(LeftBack1,GPIO.LOW)
```

```
elif x=='f':
```

```
print("GUROA")
GPIO.output(RightFront1,GPIO.LOW)
GPIO.output(RightFront2,GPIO.LOW)
GPIO.output(LeftFront1,GPIO.HIGH)
GPIO.output(LeftFront2,GPIO.LOW)
GPIO.output(RightBack1,GPIO.LOW)
GPIO.output(RightBack2,GPIO.LOW)
GPIO.output(LeftBack2,GPIO.HIGH)
GPIO.output(LeftBack1,GPIO.LOW)
```

```
elif x=='g':
```

```
print("GUROB")
GPIO.output(RightFront1,GPIO.HIGH)
GPIO.output(RightFront2,GPIO.LOW)
GPIO.output(LeftFront1,GPIO.LOW)
GPIO.output(LeftFront2,GPIO.HIGH)
GPIO.output(RightBack1,GPIO.LOW)
GPIO.output(RightBack2,GPIO.LOW)
GPIO.output(LeftBack2,GPIO.LOW)
GPIO.output(LeftBack1,GPIO.LOW)
```

```
elif x=='m':
```

```
print("30")
p.ChangeDutyCycle(30)
```

```
q.ChangeDutyCycle(30)
o.ChangeDutyCycle(30)
i.ChangeDutyCycle(30)
x='z'
```

```
elif x=='n':
```

```
print("50")
p.ChangeDutyCycle(50)
q.ChangeDutyCycle(50)
o.ChangeDutyCycle(50)
i.ChangeDutyCycle(50)
x='z'
```

```
elif x=='k':
```

```
print("75")
p.ChangeDutyCycle(75)
q.ChangeDutyCycle(75)
o.ChangeDutyCycle(75)
i.ChangeDutyCycle(75)
x='z'
```

```
elif x=='v':
```

```
print("high")
p.ChangeDutyCycle(90)
q.ChangeDutyCycle(90)
o.ChangeDutyCycle(90)
i.ChangeDutyCycle(90)
x='z'
```

```
elif x=='o':
```

```
GPIO.cleanup()
print("GPIO Clean up")
```

```
break

elif s.close():
    GPIO.cleanup()
    print("GPIO Clean up")
    break
```