



**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑΤΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε

ΔΜΠΣ: ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΑΡΑΓΩΓΗΣ, ΑΥΤΟΜΑΤΙΣΜΟΥ & ΡΟΜΠΟΤΙΚΗΣ

## ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη και αυτόνομη πλοήγηση ρομποτικής  
πλατφόρμας αγρού σε καλλιέργεια σειρών

Καλυκάκης Γ. Εμμανουήλ

Επιβλέπων: Δρ. Καβουσσάνος Εμμανουήλ

Ηράκλειο Κρήτης, Απρίλιος 2016



**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑΤΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε

ΔΜΠΣ: ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΑΡΑΓΩΓΗΣ, ΑΥΤΟΜΑΤΙΣΜΟΥ & ΡΟΜΠΟΤΙΚΗΣ

## ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

### Ανάπτυξη και αυτόνομη πλοήγηση ρομποτικής πλατφόρμας αγρού σε καλλιέργεια σειρών

Καλυκάκης Γ. Εμμανουήλ

Επιβλέπων: Δρ. Καβουσσάνος Εμμανουήλ

Εγκρίθηκε από την τριμελή επιτροπή στις 22/04/2016

Δρ. Καβουσσάνος Εμμανουήλ, Καθηγητής .....  
(Υπογραφή)

Δρ. Σφακιωτάκης Μιχαήλ, Επίκουρος Καθηγητής .....  
(Υπογραφή)

Δρ. Φασουλάς Ιωάννης, Επίκουρος Καθηγητής .....  
(Υπογραφή)

Ηράκλειο Κρήτης, Απρίλιος 2016



## Ευχαριστίες

Η παρούσα μεταπτυχιακή εργασία εκπονήθηκε στα πλαίσια του Διατμηματικού Προγράμματος Μεταπτυχιακών Σπουδών «Προηγμένα Συστήματα Παραγωγής, Αυτοματισμού & Ρομποτικής» του ΤΕΙ Κρήτης υπό την επίβλεψη του καθηγητή Δρ. Καβουσσανού Εμμανουήλ τον οποίο και ευχαριστώ για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον αντικείμενο που ανταποκρίνεται στα επιστημονικά μου ενδιαφέροντα καθώς και για την εμπιστοσύνη που έδειξε στο πρόσωπο μου.

Θα ήθελα να ευχαριστήσω τον Δρ. Σφακιωτάκη Μιχαήλ επίκουρο καθηγητή του τμήματος Ηλεκτρολόγων Μηχανικών για την συμμετοχή του στην εξεταστική επιτροπή, για τις γνώσεις που μου μετέδωσε κατά την διάρκεια των μαθημάτων αλλά και για τις συμβουλές που μου έδωσε κατά την διάρκεια εκπόνησης της εργασίας. Ευχαριστώ επίσης και τον Δρ. Φασουλά Ιωάννη επίκουρο καθηγητή του τμήματος Μηχανολόγων Μηχανικών για την συμμετοχή του στην τριμελή επιτροπή καθώς και για τις εύστοχες παρατηρήσεις του που αποσκοπούσαν στην βελτίωση της παρούσας εργασίας.

Αισθάνομαι την υποχρέωση να εκφράσω την ευγνωμοσύνη μου στους προπτυχιακούς φοιτητές του τμήματος Μηχανολογίας με τους οποίους συνεργάστηκα για την παραγωγή των εξαρτημάτων της ρομποτικής πλατφόρμας: τον Μουντάκη Νικόλαο ο οποίος αφιέρωσε άπλετο χρόνο και κατασκεύασε τα περισσότερα από αυτά, τον Κοντέ Γεώργιο και τέλος τον συμφοιτητή μου Φρουδαράκη Κωνσταντίνο. Χωρίς την συμβολή τους το ρομποτικό όχημα απλά δεν θα είχε φυσική υπόσταση.

Ευχαριστώ επίσης την συμφοιτήτρια μου Γκλίβα Ρόζα για την υποστήριξη που μου παρείχε, τον συμφοιτητή μου Δεδούση Κωνσταντίνο για τις συμβουλές που μου έδωσε για τον τρόπο κατασκευής των τυπωμένων κυκλωμάτων, τον φίλο Παρασύρη Σταύρο που μου επέτρεψε να χρησιμοποιήσω το αγρόκτημα του για την διεξαγωγή των πειραμάτων και τέλος τον κ. Συλλιγνάκη Ιωάννη για την συνδρομή του στην διαδικασία αγοράς των εξαρτημάτων του οχήματος.

Τέλος θα ήθελα να ευχαριστήσω την σύζυγο μου Πηνελόπη για την ηθική στήριξη που μου παρείχε και κυρίως για την υπομονή που έδειξε όλο αυτό το διάστημα.

## Περίληψη

Η διείσδυση των επιστημών της μηχανικής και της πληροφορικής στην γεωργική παραγωγή και γενικότερα η ενσωμάτωση ολοένα και περισσότερων τεχνολογιών αιχμής αποσκοπεί στην καλύτερη διαχείριση του αγροκτήματος, στην μείωση του κόστους παραγωγής, την προστασία του φυσικού περιβάλλοντος και στην αυτοματοποίηση των γεωργικών διεργασιών. Τα τελευταία χρόνια παρατηρείται μια αυξανόμενη τάση στην μελέτη και την κατασκευή ευέλικτων, ελαφριών και μικρών σε διαστάσεις αυτόνομων ρομποτικών οχημάτων. Η «ευφυΐα» που προσδίδεται σε αυτές τις ηλεκτρομηχανολογικές διατάξεις ώστε να χαρακτηριστούν ως ρομπότ έγκειται στην χρήση κατάλληλων αισθητήριων συνδυασμένων με ανάλογους αλγόριθμους επεξεργασίας και στην αξιοποίηση της παρεχόμενης πληροφορίας.

Η παρούσα εργασία παρουσιάζει και περιγράφει την εξ αρχής ανάπτυξη και κατασκευή ενός τετράτροχου ρομποτικού οχήματος χαμηλού κόστους και εν συνεχεία πραγματεύεται τον έλεγχο κίνησης του κατά την αυτόνομη οδήγηση του σε καλλιέργεια σειρών (row crop). Το όχημα διαθέτει σύστημα παθητικής ανάρτησης αρθρωτού τύπου (passive articulated suspension) ώστε το σασί του να προσαρμόζεται στη μορφολογία του εδάφους και οι τέσσερις κινητήριοι τροχοί του να μην χάνουν την πρόσφυση τους κατά την υπέρβαση εμποδίων. Για λόγους απλότητας επιλέχθηκε η λύση της διαφορικής οδήγησης (differential drive) συνεπώς σε κάθε αλλαγή κατεύθυνσης το όχημα πλαγιολισθάνει. Η ρύθμιση της ταχύτητας περιστροφής των τροχών υλοποιείται μέσω κλειστού βρόγχου με PID αντιστάθμιση. Η τηλεκατεύθυνση του οχήματος και η λήψη δεδομένων τηλεμετρίας επιτυγχάνεται μέσω Bluetooth πρωτοκόλλου. Η πλατφόρμα είναι εξοπλισμένη με πληθώρα αισθητήριων εσωτερικής κατάστασης και αντιλαμβάνεται το εξωτερικό περιβάλλον της χρησιμοποιώντας μόνο μια web κάμερα η οποία είναι εγκατεστημένη σε έναν μηχανισμό περιστροφής και κλίσης. Η αυτόνομη πλοήγηση της ρομποτικής πλατφόρμας κατά την διάσχιση διαδρομής εκατέρωθεν της οποίας υπάρχουν συστοιχίες φυτών επιτυγχάνεται με έναν ελεγκτή κίνησης ασαφούς λογικής κάνοντας χρήση αποκλειστικά οπτικής πληροφορίας (visual servoing).

Στους κάτωθι υπερσύνδεσμους διατίθενται σχετικά βίντεο στα οποία αποτυπώνεται η λειτουργία και οι δυνατότητες κίνησης της ρομποτικής πλατφόρμας τόσο κατά την απομακρυσμένη διαχείριση της όσο και κατά την αυτόνομη λειτουργία της.

[Τηλεχειρισμός](#)

[Αυτόνομη πλοήγηση](#)

## **Abstract**

The penetration of the mechatronics science and informatics on the agricultural production and generally the embedding of more cutting edge technologies aims to the better management of the farm, to the reduction of the production cost, to the protection of the natural environment and to automation of the agricultural procedures. During recent years an increasing tendency is observed to the search and construction of flexible, light and small in dimensions autonomous robotics vehicles. The intelligence that is given to these electromechanical devices in order to be characterized as robots has to do with the usage of appropriate sensors which are combined to relative processing algorithms and to the utilization of the information that is resulted.

This present project presents and describes the development and construction of the very beginning of a low cost four wheel robotic vehicle and continuously deals with its motion control during its autonomous navigation trough the row crop. The vehicle has a passive articulated suspension system so its chassis to be adjusted to the soil's morphology and its four drive wheel to remain in contact all the time with it during its effort to climb over obstacles. For reasons of simplicity the solution of the differential drive has been chosen. Skid-steering is achieved by imposing different wheel velocities on opposite sides of the rover. The adjusting of the velocity of the wheels' rotation is implemented via a closed loop system with PID compensation. The teleoperation of the vehicle and the receiving of the telemetry's data is succeed by Bluetooth protocol. The platform is equipped by various internal state sensors and perceives its external environment by using only one web camera which is installed on pan/tilt mechanism. The autonomous navigation of the robotic platform during crossing the route which is a row crop is succeed through a fuzzy logic motion controller which uses exclusively visual information.

# Πίνακας περιεχομένων

<b>1. Εισαγωγικά στοιχεία</b> .....	<b>1</b>
1.1 Γεωργία Ακριβείας .....	1
1.2 Η ρομποτική τεχνολογία στην γεωργία .....	3
1.2.1 Αγροτικά ρομπότ.....	4
1.2.1.1 VineRobot: Η τεχνολογία στην υπηρεσία της αμπελοκαλλιέργειας.....	5
1.2.1.2 Οι ρομποτικές πλατφόρμες Mantis και Shrimp .....	6
1.2.1.3 Ο ρομποτικός αγρότης Ladybird .....	7
1.3 Πληροφορίες για τον διαγωνισμό «Robot Field Event» .....	8
1.3.1 1 <sup>η</sup> δραστηριότητα - βασική πλοήγηση.....	8
1.3.2 2 <sup>η</sup> δραστηριότητα - αναβαθμισμένη πλοήγηση .....	10
1.3.3 3 <sup>η</sup> δραστηριότητα - εντοπισμός άνδρων φυτών και κατασκευή χάρτη .....	11
1.3.4 4 <sup>η</sup> δραστηριότητα - freestyle.....	13
1.4 Συστήματα ανάρτησης .....	15
1.4.1 Σύστημα ανάρτησης τύπου Rocker-Bogie .....	16
1.4.1.1 Το ρομποτικό ρόβερ Dune .....	17
1.4.1.2 Η ρομποτική πλατφόρμα εξερεύνησης Oryx 2.0 .....	19
1.5 Αλλαγή κατεύθυνσης με πλαγιολίσθηση (Skid-Steering) .....	21
1.6 Συνεισφορά της παρούσας εργασίας .....	22
<b>2. Δομικά εξαρτήματα οχήματος - Συναρμολόγηση</b> .....	<b>27</b>
2.1 Δομικά εξαρτήματα του οχήματος .....	27
2.1.1 Σφαιρικοί σύνδεσμοι .....	27
2.1.2 Τροχοί.....	28
2.1.3 Εξαρτήματα Actobotics .....	28
2.1.4 Κατασκευή εξαρτημάτων.....	29
2.1.4.1 Κουτί ηλεκτρονικών ελέγχου .....	29
2.1.4.2 Καπάκι κουτιού των ηλεκτρονικών ελέγχου .....	30
2.1.4.3 Μπάρα διαφορικού.....	30
2.1.4.4 Προσαρμογέας σύνδεσης του κινητήρα με τον τροχό του οχήματος .....	30
2.1.4.5 Βάση στήριξης κινητήρα και σύνδεσης του με το σασί.....	31
2.1.4.6 Βάση στήριξης του καπακιού των ηλεκτρονικών ελέγχου .....	31
2.1.4.7 Βάση σταθεροποίησης του στύλου για την κάμερα.....	32
2.1.4.8 Αποστάτες .....	32
2.2 Συναρμολόγηση .....	33
2.2.1.1 Σύνδεση κινητήρων με τη βάση στήριξης τους.....	33
2.2.1.2 Συναρμολόγηση των πλευρικών βραχιόνων (rocker arms).....	34
2.2.1.3 Συναρμολόγηση εξαρτημάτων στο κουτί των ηλεκτρονικών ελέγχου .....	38

2.2.1.4	Συναρμογή του σασί με το κουτί ελέγχου .....	41
2.3	3D γραφικές απεικονίσεις του ρόβερ .....	42
<b>3.</b>	<b>Κατανομή ηλεκτρικής ενέργειας - Τροφοδοσία .....</b>	<b>44</b>
3.1	Ηλεκτρονικά ισχύος - Ρυθμιστές τάσεις.....	44
3.1.1	Γραμμικοί ρυθμιστές.....	44
3.1.2	Διακοπτικοί ρυθμιστές τάσης .....	46
3.1.2.1	Διακοπτικού τύπου κύκλωμα μετατροπής DC σε DC.....	47
3.2	Ηλεκτρολογικό διάγραμμα συνδέσεων .....	49
3.3	Μπαταρίες Ιόντων Λιθίου Πολυμερών (LiPo).....	51
3.3.1	Γενικά στοιχεία.....	51
3.3.2	Φόρτιση.....	52
3.3.3	Προφυλάξεις .....	53
3.3.4	Αδιάλειπτος έλεγχος του επιπέδου της τάσης του συσσωρευτή.....	54
3.4	Εκτίμηση απαιτούμενης ισχύος και χωρητικότητας συσσωρευτή .....	54
<b>4.</b>	<b>Μονάδες εισόδου/εξόδου - Αισθητήρια .....</b>	<b>56</b>
4.1	Μονάδες εισόδου/εξόδου .....	56
4.1.1	Διασύνδεση και έλεγχος οθόνης LCD .....	56
4.1.2	Διασύνδεση και ανάγνωση ηλεκτρολογίου 4x4.....	58
4.2	Αισθητήρια εσωτερικής κατάστασης.....	61
4.2.1	Χρησιμοποιούμενα ψηφιακά φίλτρα .....	61
4.2.1.1	Απλό φίλτρο κινούμενου μέσου .....	62
4.2.1.2	Φίλτρο εκθετικής εξομάλυνσης .....	62
4.2.2	Διαιρέτης τάσης .....	63
4.2.3	Αισθητήρας θερμοκρασίας TMP36.....	65
4.2.3.1	Οδήγηση ανεμιστήρα.....	66
4.2.4	Αισθητήριο ρεύματος .....	67
4.2.5	Αισθητήρια για τον υπολογισμό του προσανατολισμού .....	69
4.2.5.1	Μαγνητόμετρο 3 αξόνων .....	69
4.2.5.2	Επιταχυνσιόμετρο & Γυροσκόπιο 3 αξόνων .....	70
<b>5.</b>	<b>Έλεγχος κίνησης τροχών.....</b>	<b>71</b>
5.1	Ηλεκτρικός κινητήρας EMG49.....	72
5.2	Μοντελοποίηση κινητήριου συστήματος.....	73
5.3	Χάραξη χαρακτηριστικών ροπής.....	76
5.4	Περιστροφικός κωδικοποιητής αυξητικού τύπου .....	79
5.4.1	Εισαγωγικά στοιχεία .....	79
5.4.2	Ανάγνωση κωδικοποιητή .....	80
5.5	Οδήγηση του σερβοκινητήρα EMG49 μέσω Η-γέφυρας.....	82
5.6	Εκτίμηση γωνιακής ταχύτητας περιστροφής.....	84



5.7	Πειραματική ταυτοποίηση συστήματος .....	87
5.7.1	Πειραματικά αποτελέσματα ταχύτητας περιστροφής dc κινητήρα-τροχού ανοικτού βρόγχου .....	89
5.7.2	Συναρτήσεις μεταφοράς ανοικτού βρόγχου .....	96
5.7.3	Προσομοίωση συστήματος ανοικτού βρόγχου .....	97
5.7.4	Αποτελέσματα προσομοίωσης .....	99
5.8	Αναλογικός-ολοκληρωτικός-διαφορικός ελεγκτής.....	101
5.8.1	Διακριτοποίηση και ψηφιακή υλοποίηση του PID ελεγκτή .....	101
5.8.2	Ρύθμιση ελεγκτή PID με τη μέθοδο των Ziegler- Nichols .....	103
5.8.3	Σχεδίαση ελεγκτή PID με την μέθοδο άμεσης σύνθεσης .....	108
5.8.3.1	Εφαρμογή μεθόδου στο σύστημα του εμπρόσθιου αριστερού τροχού ..	110
5.9	Ο διάυλος I <sup>2</sup> C.....	113
5.9.1	Η βιβλιοθήκη Wire .....	117
5.10	Τυπωμένο κύκλωμα ελέγχου της ταχύτητας περιστροφής των τροχών .....	118
5.11	Η αναπτυξιακή πλακέτα Arduino Mega 2560.....	122
5.12	Πειραματικά αποτελέσματα .....	127
5.13	Εκτύπωση γραφημάτων σε πραγματικό χρόνο .....	131
<b>6.</b>	<b>Σύστημα μηχανικής όρασης - Υλικό &amp; Λογισμικό .....</b>	<b>134</b>
6.1	Μηχανισμός περιστροφής/κλίσης της κάμερας.....	134
6.1.1	Μικρό-σερβοκινητήρες Batao 2122 - Βαθμονόμηση .....	136
6.2	Ο μικροϋπολογιστής Odroid-XU3 .....	139
6.3	Θεωρητικό υπόβαθρο - Χρησιμοποιούμενοι αλγόριθμοι μηχανικής όρασης ...	144
6.3.1	Κανονικοποίηση χρωματικών τιμών στον χώρο RGB .....	144
6.3.2	Τμηματοποίηση εικόνας - Μέθοδος κατωφλίωσης Otsu.....	145
6.3.2.1	Παράδειγμα εφαρμογής της μεθόδου Otsu .....	147
6.3.3	Χρωματικοί δείκτες βλάστησης στο ορατό φάσμα .....	148
6.3.4	Μορφολογική επεξεργασία δυαδικών εικόνων .....	156
6.3.5	Υπολογισμός κέντρου μάζας (centroid).....	158
6.3.6	Διαχωριστική συσταδοποίηση σημείων - Αλγόριθμος k-means .....	158
6.4	Αλγόριθμος μηχανικής όρασης.....	160
6.5	Αυτόματη εκτέλεση εφαρμογών κατά την έναρξη του Λ.Σ. Ubuntu .....	170
<b>7.</b>	<b>Ασαφής ελεγκτής αυτόνομης πλοήγησης .....</b>	<b>172</b>
7.1	Ασαφής έλεγχος - εισαγωγικά στοιχεία.....	172
7.2	Σχεδίαση ασαφούς ελεγκτή .....	173
<b>8.</b>	<b>Ασύρματη επικοινωνία .....</b>	<b>182</b>
8.1	Διαχείριση του Odroid-XU3 εξ' αποστάσεως μέσω Wi-Fi.....	182
8.2	Διαχείριση και έλεγχος οχήματος μέσω Bluetooth .....	185

<b>9. Προσανατολισμός οχήματος .....</b>	<b>190</b>
9.1  Υλοποιηθείσα διάταξη .....	190
9.2  Μαγνητόμετρο .....	191
9.3  Εύρεση γωνιών roll και pitch από το επιταχυνσιόμετρο .....	193
9.4  Διακριτό φίλτρο Kalman.....	195
9.4.1  Εξισώσεις αναπροσαρμογών .....	196
9.5  Εφαρμογή του διακριτού φίλτρου Kalman για τον συγκερασμό των δεδομένων του επιταχυνσιόμετρου και του γυροσκόπιου .....	197
<b>10.Συμπεράσματα - Προτάσεις για μελλοντική εξέλιξη.....</b>	<b>202</b>
Βιβλιογραφία .....	205
Παράρτημα Α - Φωτογραφίες οχήματος .....	208
Παράρτημα Β - Μηχανολογικά σχέδια .....	212
Παράρτημα Γ - Τυπωμένα κυκλώματα .....	221
Παράρτημα Δ - Ρυθμίσεις μεταγλωττιστών.....	239
Παράρτημα Ε - Πηγαίος κώδικας .....	242

# Κεφάλαιο

# 1

## 1. Εισαγωγικά στοιχεία

Στο κεφάλαιο αυτό αναπτύσσεται η έννοια της Γεωργίας Ακριβείας, αναφέρονται οι λόγοι για την επιβεβλημένη μετάβαση από τις συμβατικές μεθόδους αγροτικής παραγωγής σε νέες μορφές διαχείρισης. Εξηγείται γιατί η είσοδο της ρομποτικής στον γεωργικό τομέα θα έχει μακροπρόθεσμα πολλαπλά οφέλη τόσο για την παραγωγή όσο και το περιβάλλον και θα είναι τελικά οικονομικά βιώσιμη. Κατόπιν αναφέρονται οι δυνατότητες των σύγχρονων ρομποτικών οχημάτων και παρουσιάζονται κάποιες αξιόλογες ερευνητικές προσπάθειες. Δίνονται πληροφορίες για τον διεθνή διαγωνισμό "Robot Field Event" η διεξαγωγή του οποίου έχει κίνητρο την ανάπτυξη ρομποτικών οχημάτων για χρήση σε γεωργικές εφαρμογές και ωθεί την ενσωμάτωση τεχνολογιών αιχμής στον αγροτικό τομέα. Περιγράφονται οι ζητούμενες ενέργειες που θα έπρεπε να επιτελέσουν οι ρομποτικές πλατφόρμες μέσω των δραστηριοτήτων του διαγωνισμού για το έτος 2015 και παρουσιάζονται οι νικητές σε κάθε περίπτωση.

Στις επόμενες ενότητες τεκμηριώνονται οι λόγοι καταλληλότητας των συστημάτων παθητικής ανάρτησης για τα τροχοφόρα ρομποτικά οχήματα που προορίζονται να κινούνται σε ανώμαλο έδαφος και καθορίζονται τα κριτήρια επιλογής για την χρήση ενός τέτοιου δικτύωματος. Περιγράφεται το σύστημα ανάρτησης τύπου Rocker-Bogie καθώς και τα πλεονεκτήματα των οχημάτων που φέρουν σύστημα αναρτήσεως τέτοιου είδους, εν συνεχεία παρουσιάζονται δυο ερευνητικά τετράτροχα οχήματα με παρεμφερές σύστημα ανάρτησης αλλά απλούστερης σχεδίασης. Ακολουθως αναφέρονται οι εφικτοί ελιγμοί και οι ιδιότητες των τετράτροχων οχημάτων με διαφορεική οδήγηση. Κλείνοντας δίνεται μια σύντομη περιγραφή του ρομποτικού οχήματος που αναπτύχθηκε κατά την εκπόνηση της παρούσας εργασίας.

### 1.1 Γεωργία Ακριβείας

Χιλιάδες χρόνια πριν έως και σήμερα γίνεται μεγάλη προσπάθεια από την ανθρωπότητα για την διασφάλιση ικανής γεωργικής παραγωγής, τόσο σε ποσότητα όσο και σε ποιότητα.

Στο παρελθόν οι παραγωγοί γεωργικών προϊόντων εκτελούσαν τις καλλιεργητικές τους εργασίες χειρωνακτικά και είχαν άμεση επαφή με το έδαφος, την καλλιέργεια και τις διάφορες παραμέτρους που επηρέαζαν την παραγωγή τους. Έτσι γνώριζαν με μεγαλύτερη ακρίβεια τα τμήματα στο χωράφι τους με διαφορετικά χαρακτηριστικά και είχαν την δυνατότητα να διαφοροποιήσουν τη διαχείριση του κάθε τμήματος του αγρού τους αλλά και να ρυθμίσουν ανάλογα τις εισροές του, όπως σπόρο, νερό, οργανική ουσία κλπ.

Με την ανάπτυξη και εξέλιξη των γεωργικών μηχανημάτων, οι παραγωγοί έχουν τη δυνατότητα να εκμεταλλεύονται μεγαλύτερες εκτάσεις χάνοντας όμως την άμεση επαφή με τον αγρό και εφαρμόζουν τις καλλιεργητικές εισροές τους σε δοσολογίες που αντιπροσωπεύουν μέσους όρους. Ακόμα και στις περιπτώσεις που οι παραγωγοί γνωρίζουν

κάποιες διαφοροποιήσεις στο έδαφος των αγρών τους, η τεχνολογία των μηχανημάτων εφαρμογής των εισροών που χρησιμοποιούν δεν τους βοηθά στην διαφορετική αντιμετώπιση των τμημάτων αυτών.

Η αύξηση του ανταγωνισμού στη γεωργία τα τελευταία χρόνια, έχει οδηγήσει στην ανάγκη για δραστική μείωση του κόστους στη γεωργική παραγωγή, ενώ τα οξυμένα προβλήματα του περιβάλλοντος έχουν κάνει επιτακτική τη μεγαλύτερη δυνατή μείωση των επιβλαβών εισροών στον αγρό. Οι τιμές των εισροών αυξάνονται, ενώ οι πραγματικές τιμές των γεωργικών προϊόντων μένουν σταθερές ή μειώνονται λόγω του διεθνούς ανταγωνισμού με συνέπεια τον περιορισμό του καθαρού εισοδήματος ανά καλλιεργήσιμο στρέμμα. Οι αυστηρότεροι νόμοι για τις περιβαλλοντικές επιπτώσεις των καλλιεργειών και οι κανονισμοί για τις διάφορες πιστοποιήσεις καθιστούν πιο αναγκαία από ποτέ τη χρήση νέων τεχνολογιών.

Σήμερα με την βελτίωση των δυνατοτήτων των ηλεκτρονικών υπολογιστών και με την ανάπτυξη της τεχνολογίας γενικότερα είναι δυνατή η αντιμετώπιση της παραλλακτικότητας<sup>1</sup> ενός αγρού. Όταν η παραλλακτικότητα του γεωργικού περιβάλλοντος λαμβάνεται υπόψη σε κλίμακα μικρότερη του αγροτεμαχίου τότε η μέθοδος γεωργικής διαχείρισης ονομάζεται **Γεωργία Ακριβείας** (Precision Agriculture). Η γεωργία ακριβείας (ή καλλιέργεια ακριβείας) είναι το σύνολο των γεωργικών πρακτικών που εστιάζονται σε συγκεκριμένες περιοχές του αγρού και σε συγκεκριμένη χρονική στιγμή. Βασίζεται στην παρατήρηση και τις μετρήσεις ώστε να είναι γνωστά π.χ. όλα τα χαρακτηριστικά του εδάφους όπως η περιεκτικότητα του σε διάφορα στοιχεία, ο τύπος του, τα φυσικά χαρακτηριστικά του, η προσβολή από ζιζάνια κλπ. Οι καλλιεργητικές πρακτικές περιλαμβάνουν την επιλογή ποικιλίας/υβριδίου, την προσαρμοσμένη εφαρμογή λίπανσης (π.χ. διαφορετικές ποσότητες σε διαφορετικά σημεία του αγρού), την ορθολογική χρήση των χημικών εισροών, τις αποφάσεις για τύπο άρωσης και άρδευσης κλπ. Αυτό έρχεται σε αντίθεση με τις πιο παραδοσιακές πρακτικές στις οποίες οι διάφορες γεωργικές δραστηριότητες, όπως η άρδευση, η εφαρμογή λιπασμάτων, εντομοκτόνων και ζιζανιοκτόνων εφαρμόζονται ομοιόμορφα σε ολόκληρο το χωράφι, αγνοώντας κάθε μεταβλητότητα μέσα στο χωράφι. Η παραδοσιακή γεωργική διαχείριση αντιμετωπίζει τα αγροτεμάχια ως ομοιόμορφα (βασιζόμενη σε μέσους όρους) και αγνοεί την εγγενή ή επίκτητη ως προς το χώρο και το χρόνο παραλλακτικότητα τους. Στην καρδιά της Γεωργίας Ακριβείας βρίσκεται η διαχείριση αυτής ακριβώς της παραλλακτικότητας.

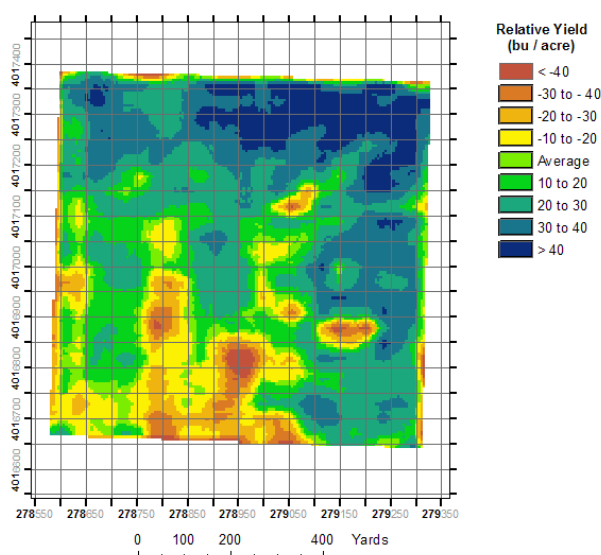
Με την Γεωργία Ακριβείας επιτυγχάνεται η εφαρμογή της κατάλληλης εισροής, στο σημείο που χρειάζεται, στην κατάλληλη δόση και σωστό χρόνο, με την ελάχιστη δυνατή επιβάρυνση του περιβάλλοντος. Κάθε σημείο του αγρού ερευνάται και ανάλογα τον επιδιωκόμενο σκοπό, δέχεται την απόλυτα αναγκαία ποσότητα της κατάλληλης εισροής. Το αποτέλεσμα είναι η ακριβέστερη εφαρμογή των κανόνων της Ορθής Γεωργικής Πρακτικής (π.χ. EUREPGAP), η μεγιστοποίηση της οικονομικής απόδοσης της γεωργικής εκμετάλλευσης και η βελτίωση της ανταγωνιστικότητας των γεωργικών προϊόντων μέσω της μείωσης της υπερβολικής χρήσης χημικών και της υπέρμετρης χρήσης γεωργικών μηχανημάτων. Τα οφέλη είναι προφανή και για το περιβάλλον, όπου σταματά η αλόγιστη χρήση εισροών σε δοσολογίες πολλαπλάσιων των απαιτούμενων.

Θεμέλιο λίθο στην Γεωργία Ακριβείας αποτελεί το Παγκόσμιο Σύστημα Καθορισμού Θέσης (Global Positioning System) ένα δίκτυο 24 δορυφόρων της Γης. Ένας δέκτης GPS τοποθετείται σε ένα γεωργικό μηχάνημα και με βάση το γεωγραφικό στίγμα είναι δυνατή η καταγραφή δεδομένων αλλά και η εφαρμογή εισροών ελεγχόμενα και συγκεκριμένα

---

<sup>1</sup> Ως παραλλακτικότητα (variability) του γεωργικού περιβάλλοντος εννοείται η διαφοροποίηση που παρουσιάζουν οι παράμετροι της καλλιέργειας, είτε ως προς τη θέση, είτε ως προς τη χρονική στιγμή κατά την οποία μετρώνται και μπορεί να είναι εγγενής ή επίκτητη.

σημεία. Με το κατάλληλο λογισμικό δημιουργούνται γεωστατικοί χάρτες διάφορων παραμέτρων του αγρού, όπως της εδαφικής ηλεκτρικής αγωγιμότητας, εδαφικών αναλύσεων, της παραγωγής και των ποιοτικών χαρακτηριστικών αυτής. Η συλλογή πληροφοριών υπό μορφή χαρτών βοηθά στην λήψη αποφάσεων για τον εντοπισμό της παραλλακτικότητας του αγρού και το διαχωρισμό σε ομοιόμορφες ζώνες. Σκοπός του διαχωρισμού της έκτασης του αγρού σε μικρότερα ομοιογενή τμήματα (ζώνες διαχείρισης) είναι η διαχείριση αυτών σύμφωνα με τα ιδιαίτερα χαρακτηριστικά τους και η εφαρμογή των καλλιεργητικών εισροών με διαφοροποιημένη δόση.



Εικόνα 1.1. Χάρτης απόδοσης αγρού.

## 1.2 Η ρομποτική τεχνολογία στην γεωργία

Η συντριπτική πλειονότητα των ρομπότ που λειτουργούν σήμερα αποτελείται από προγραμματιζόμενους βιομηχανικούς βραχίονες εγκατεστημένους σε παραγωγικές μονάδες, όπου εκτελούν αυστηρά επαναλαμβανόμενες εργασίες, σε κατάλληλα δομημένο, απόλυτα ελεγχόμενο περιβάλλον, όπως για παράδειγμα αυτό της αυτοκινητοβιομηχανίας.

Στη γεωργία, εργασίες που είναι σχετικά απλές για τον άνθρωπο, όπως π.χ. ο εντοπισμός και η συλλογή ενός καρπού από ένα δένδρο ακόμη και αν εμποδίζεται από το φύλλωμα ή κλαδιά, για ένα ρομπότ είναι μια πολύπλοκη και πολυσύνθετη διαδικασία, η οποία βρίσκεται ακόμα σε πειραματικό επίπεδο. Τέτοιες ενέργειες παραμένουν ακόμα εκτός των δυνατοτήτων των σύγχρονων ρομπότ σε επίπεδο παραγωγής. Η λειτουργία ρομπότ σε γεωργικό περιβάλλον αποτελεί ιδιαίτερη τεχνολογική πρόκληση, κυρίως λόγω της αδυναμίας μας να τα σχεδιάσουμε και να τα προγραμματίσουμε ώστε να συνυπάρχουν και να αλληλεπιδρούν αξιόπιστα, με ασφάλεια και ταχύτητα με ανθρώπους, ζώα και φυτά. Οι γεωργικές εργασίες συχνά είναι επίπονες, επαναλαμβανόμενες αλλά όχι πανομοιότυπες, ενώ εκτελούνται συνήθως κάτω από αντίξοες καιρικές συνθήκες σε απρόβλεπτο και δύσκολο περιβάλλον. Αυτό είναι όμως και μια πρόκληση αφού είτε λόγω της υφιστάμενης τεχνολογίας, είτε λόγω του δυναμικού περιβάλλοντος (π.χ. σκίαση, φυσικός φωτισμός, φυσικά ή τεχνητά εμπόδια, κλπ), τα αυτόνομα ρομπότ δύσκολα αντεπεξέρχονται των περίπλοκων και συχνά επικίνδυνων γεωργικών ασχολιών.

Ήδη, ρομποτικές τεχνολογίες έχουν ενσωματωθεί στα σύγχρονα γεωργικά μηχανήματα (π.χ., τρακτέρ, θεριζοαλωνιστικές) και ολοένα και περισσότερο η γεωργία

λειτουργεί με γνώμονα τη συλλογή και την επεξεργασία δεδομένων υλοποιώντας τεχνικές γεωργίας ακριβείας. Παρά το σχετικά υψηλό κόστος, την περιορισμένη ταχύτητα και αξιοπιστία των πειραματικών γεωργικών ρομπότ, εκτιμάται από ειδικούς ότι εντός της επόμενης δεκαετίας θα εισαχθούν στη γεωργική παραγωγή ρομπότ, ικανά να εργασθούν αυτόνομα σε πραγματικές συνθήκες παραγωγής. Γενικότερα θα υπάρχει μια σημαντική στροφή προς την υιοθέτηση νέων τεχνικών όπως επίσης και αλλαγή νοοτροπίας ώστε ο γεωργός να υιοθετεί καινούργιες τεχνικές και τεχνολογίες.

Η χρήση ρομποτικής τεχνολογίας στη γεωργία μπορεί να αποτελέσει μία αποτελεσματική λύση στα υφιστάμενα προβλήματα, αφού ως γνωστό μεγάλο τμήμα του κόστους παραγωγής προέρχεται από τη χειρωνακτική εργασία. Το γεωργικό επάγγελμα φθίνει, ενώ συχνά παρατηρείται δυσκολία εξεύρεσης μόνιμων ή εποχιακών εργατών ιδιαίτερα τις περιόδους συγκομιδής κυρίως στις αναπτυσσόμενες χώρες. Το αυξανόμενο ενδιαφέρον για τους μηχανισμούς αυτοματισμού πηγάζει και από το γεγονός ότι δεν υπάρχει αρκετή θέληση, ιδιαίτερα των νέων για εργασία στα χωράφια και κατ' επέκταση αυξάνεται το κόστος συγκομιδής. Από την άλλη, υπάρχει η ανάγκη για αύξηση της γεωργικής παραγωγής έτσι ώστε να ικανοποιηθούν οι αυξανόμενες ανάγκες παραγωγής τροφίμων. Στο μέλλον λόγω των κανονισμών που θα επιβληθούν (π.χ. νέα Κοινή Αγροτική Πολιτική) όσο και λόγω των διαθέσιμων τεχνολογιών που θα το καθιστούν δυνατό θα υπάρξουν αυστηρότερα κριτήρια για την παραγωγή αγροτικών αγαθών. Θεωρείται λοιπόν επιβεβλημένη η χρήση των νέων τεχνολογιών και θα δοθεί ώθηση στη χρήση νέων συστημάτων που θα αυτοματοποιούν τη γεωργική παραγωγή. Στόχος αποτελεί η μείωση της χρήσης των φυτοφαρμάκων και η ορθολογικότερη χρήση τους με σημαντικό οικονομικό και περιβαλλοντολογικό όφελος, η βέλτιστη χρήση του νερού και εξοικονόμηση καυσίμου στις καλλιέργειες επιτυγχάνοντας παράλληλα το παραγόμενο προϊόν να έχει καλύτερη ποιότητα με ταυτόχρονη μεγιστοποίηση της παραγωγής.

Σε ερευνητικό επίπεδο έχουν κατασκευαστεί ποικίλα γεωργικά ρομπότ για εργασίες όπως μεταφυτεύσεις σε φυτώρια, συλλογή καρπών, σπορά, εντοπισμός ζιζανίων και επιλεκτικό ψεκάσμο ακριβείας καθώς και μια σειρά από άλλες καλλιεργητικές φροντίδες. Η συγκομιδή έχει τραβήξει το μεγαλύτερο ερευνητικό ενδιαφέρον, κυρίως λόγω οικονομικών μεγεθών και της έλλειψης εργατικού δυναμικού. Τα γεωργικά μηχανήματα θα έχουν ολοένα και περισσότερες δυνατότητες αυτόνομης λειτουργίας, συλλογής και καταγραφής πληροφοριών. Οι δυνατότητες αυτές θα επεκτείνονται, μάλιστα, και στην επικοινωνία με προγράμματα διαχείρισης αγροκτήματος και μελλοντικά με άλλα συνεργαζόμενα γεωργικά μηχανήματα.

### **1.2.1 Αγροτικά ρομπότ**

Αρκετές εταιρίες και ερευνητικά ιδρύματα με ειδίκευση στην ανάπτυξη ρομποτικών συστημάτων αναπτύσσουν ολοκληρωμένα συστήματα (υλικό και λογισμικό) για μη επανδρωμένα οχήματα για χρήση σε γεωργικές εφαρμογές. Τα αγροτικά ρομπότ είναι ηλεκτροκίνητα οχήματα μικρού βάρους, διαθέτουν κίνηση και διεύθυνση και στους τέσσερις τροχούς είναι εφοδιασμένα με συστήματα αυτόνομης πλοήγησης, ασύρματης δικτύωσης, απομακρυσμένης επίβλεψης και με συστήματα ελέγχου και διάγνωσης βλαβών. Τα χαρακτηριστικά αυτά επιτρέπουν την εύκολη μετακίνηση σε οποιοσδήποτε συνθήκες εδαφών ανεξάρτητα από την υγρασία και το ανάγλυφο του εδάφους. Η ηλεκτροκίνηση εξασφαλίζει την οικονομική και περιβαλλοντικά φιλική λειτουργία τους ενώ τα ηλεκτρονικά τους συστήματα επιτρέπουν την αυτόνομη και αυτόματη μετακίνηση τους με τη δυνατότητα επίβλεψης και προβολής των πληροφοριών λειτουργίας σε πραγματικό χρόνο καθώς και την ικανότητα προφύλαξης τους από πιθανές βλάβες ή δυσλειτουργίες.

Τα συστήματα αυτά συλλέγουν πληροφορίες από αισθητήρες τόσο για το ευρύτερο περιβάλλον, όσο και για την ίδια τη λειτουργία τους, και στη συνέχεια τις επεξεργάζονται και διαμορφώνουν διάφορες στρατηγικές πλοήγησης. Η διαδρομή κίνησης τους επιλέγεται δυναμικά και επανασχεδιάζεται η πορεία τους λαμβάνοντας υπόψη εμπόδια και τη γενικότερη μορφολογία του εδάφους. Συνδυάζουν πληροφορίες από αισθητήρες όρασης με στοιχεία που προέρχονται από άλλους αισθητήρες IMU, GPS, πυξίδες κτλ. διαμορφώνοντας έτσι ένα χάρτη του περιβάλλοντος χώρου. Στις παρακάτω ενότητες παρουσιάζονται κάποιες αξιόλογες ερευνητικές προσπάθειες τέτοιων ρομποτικών συστημάτων.

### 1.2.1.1 VineRobot: Η τεχνολογία στην υπηρεσία της αμπελοκαλλιέργειας

Μέχρι και σήμερα οι αμπελουργοί εκτιμούν οπτικά την παραγωγή τους, μετρούν ορισμένες παραμέτρους με το χέρι ή παίρνουν δείγματα καρπών και φύλλων για ανάλυση σε εργαστήρια.

Ευρωπαίοι μηχανικοί κατασκεύασαν ένα ρομπότ το οποίο έχει ως σκοπό να συλλέγει πληροφορίες για την κατάσταση των φύλλων και των καρπών στους αμπελώνες βοηθώντας έτσι τους αγρότες να παίρνουν πιο εύκολα αποφάσεις. Προορίζεται να είναι γρηγορότερο και ακριβέστερο από τους ανθρώπους που κάνουν την ίδια δουλειά της «επιθεώρησης» του αμπελώνα. Στο πρόγραμμα εμπλέκονται οκτώ φορείς από τέσσερις διαφορετικές οινοπαραγωγές χώρες (Γαλλία, Γερμανία, Ιταλία και Ισπανία), και έχει λάβει χρηματοδότηση άνω των δύο εκατ. ευρώ από την Ε.Ε. Στο πρόγραμμα ηγείται το Universidad de La Rioja (Ισπανία).



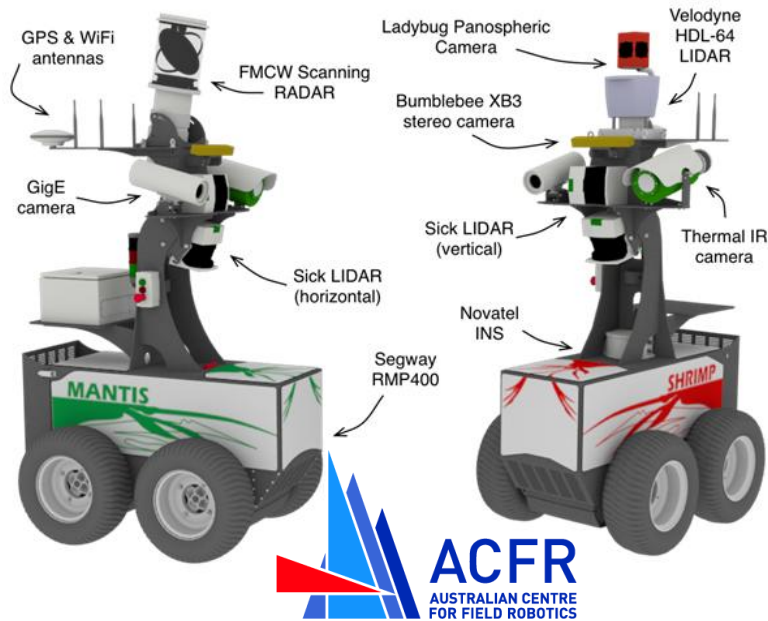
**Εικόνα 1.2** Ρομποτική πλατφόρμα αυτόματης ανιχνευτής ποιότητας και ωρίμανσης σε αμπελώνα.

Το έργο έχει σκοπό τον σχεδιασμό, την ανάπτυξη και τη χρήση ενός εξειδικευμένου μη επανδρωμένου οχήματος που θα κινείται με ηλιακή ενέργεια και θα είναι εξοπλισμένο με μη επεμβατικές τεχνολογίες εντοπισμού ώστε να λειτουργήσει ως αυτόματο σύστημα ιχνηλασιμότητας προκειμένου να καταγράφονται απαραίτητες πληροφορίες. Το VineRobot συνδυάζει προηγμένους αισθητήρες αναγνώρισης λάμψης της χλωροφύλλης, χρησιμοποιεί τεχνολογίες στερεοσκοπικής όρασης, θερμογράφο υπερύθρων, GPS. Αυτό σημαίνει πως μπορεί να παρακολουθήσει την παραγωγή, την ανάπτυξη των φυτών, το επίπεδο του νερού και τη σύσταση των καρπών χωρίς να έρχεται σε επαφή με αυτούς. Οι αγρότες λαμβάνουν αυτές τις πληροφορίες μέσω κινητού, tablet ή φορητού υπολογιστή. Το ρομπότ αυτό αποτελεί ένα ελκυστικό εργαλείο για χρήση στη γεωργία ακριβείας.

### 1.2.1.2 Οι ρομποτικές πλατφόρμες Mantis και Shrimp

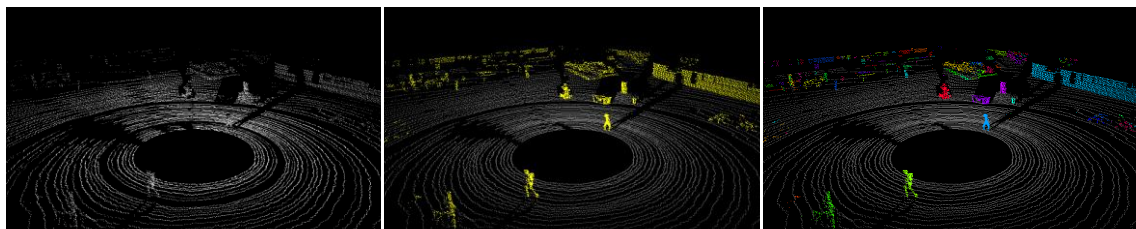
Πρόκειται για ρομποτικές πλατφόρμες γενικού σκοπού με εφαρμογές και στην γεωργία, κατάλληλες για έρευνα η οποία σχετίζεται με την αντίληψη του περιβάλλοντος χώρου. Ανάλογα με την εφαρμογή για την οποία προορίζονται υπάρχει ευελιξία ως προς το ποια αισθητήρια θα χρησιμοποιηθούν ώστε να μειωθεί το κόστος απόκτησης τους.

Εκτός από τους αισθητήρες που εικονίζονται (κάμερες, σαρωτές Laser, ραντάρ) έχει προστεθεί ένας αισθητήρας ηλεκτρικής αγωγιμότητας του εδάφους ο οποίος είναι τοποθετημένος στο πίσω μέρος της πλατφόρμας και σύρεται, έναν αισθητήρα ακτινοβολίας γ για την μέτρηση ιδιοτήτων του εδάφους και έναν υπερφασματικής απεικόνισης.



Εικόνα 1.3 Ρομποτικές πλατφόρμες Mantis και Shrimp

Με τον προηγμένο τεχνολογικά εξοπλισμό που διαθέτουν έχουν την δυνατότητα καταγραφής τέρασσιου όγκου δεδομένων (~500GB/hour ή 150MB/sec). Η πλατφόρμα Shrimp είναι εξοπλισμένη με τον αισθητήρα HDL-64E - Velodyne LIDAR ο οποίος πραγματοποιεί ένα εκατομμύριο μετρήσεις το δευτερόλεπτο και μπορεί να χρησιμοποιηθεί για την τρισδιάστατη χαρτογράφηση του χώρου. Από αυτά τα πρωτογενή δεδομένα και για την τμηματοποίηση (segmentation) και τον διαχωρισμό της εικόνας σε αντικείμενα (όπως δέντρα, ανθρώπους, κτήρια) από το έδαφος έχουν αναπτυχθεί κατάλληλοι αλγόριθμοι π.χ. mesh-voxel segmentation (Douillard 2011). Μετά την αναγνώριση του εδάφους τα αντικείμενα κατηγοριοποιούνται (clustering).



Εικόνα 1.4 Πρωτογενή δεδομένα από τον αισθητήρα LIDAR και επεξεργασία τους με αλγόριθμο για την τμηματοποίηση, κατηγοριοποίηση των αντικειμένων.



Το έργο των επίγειων ρομπότ συνεπικουρείται από εναέρια πλατφόρμα παρέχοντας τους σημαντικές πληροφορίες. Τα εναέρια οχήματα επιφορτίζονται και με ενέργειες όπως π.χ. ο ψεκασμός για την καταπολέμηση ασθενειών.



Εικόνα 1.5 Πειραματικές δοκιμές που αποσκοπούν στη συνεργασία των επίγειων οχημάτων με εναέρια πλατφόρμα.

### 1.2.1.3 Ο ρομποτικός αγρότης Ladybird

Στο πανεπιστήμιο του Σίνδνεϋ σχεδιάστηκε και αναπτύχθηκε μια αυτόνομη ηλεκτροκίνητη ρομποτική πλατφόρμα κατάλληλη για την έρευνα και την ανάπτυξη τεχνολογιών που σχετίζονται με την καλλιέργεια των λαχανικών σε μεγάλες εκτάσεις. Η δυνατότητα κίνησης και διεύθυνσης και των 4 τροχών της, και το χαμηλό βάρος της, την κάνουν ικανή να κινείται πανκατευθυντικά, ελαχιστοποιώντας έτσι και τις ζημιές οι οποίες προκαλούνται από την μετακίνηση της στην καλλιεργημένη περιοχή μέσα στην οποία βρίσκεται. Στο πλαίσιο (σασί) του οχήματος βρίσκεται τοποθετημένη μια μεγάλη γκάμα αισθητηρίων (στερεοσκοπικής όρασης, υπερφασματικής απεικόνισης κ.α.) και ένας βραχίονας έξι βαθμών ελευθερίας. Στόχος αποτελεί μια πληθώρα καλλιεργητικών ενεργειών όπως ο ψεκασμός, η καταστροφή ζιζανίων με μηχανικό τρόπο και φυσικά η συγκομιδή αλλά και η εποπτεία της καλλιεργημένης έκτασης, η χαρτογράφηση της, ο εντοπισμός και η κατάταξη των λαχανικών και των ζιζανίων. Η πλατφόρμα έχει την δυνατότητα της αυτόματης πλοήγησης άναμεσα στις καλλιεργειμένες σειρές (row crop) και μπορεί να παραμείνει σε πλήρη λειτουργία για τρεις συνεχόμενες ημέρες χάρη στις ηλιακού συλλέκτες που διαθέτει.



Εικόνα 1.6 Το αγροτικό ρομποτικό όχημα Ladybird του πανεπιστημίου του Σίνδνεϋ (ACFR).

### 1.3 Πληροφορίες για τον διαγωνισμό «Robot Field Event»



Εικόνα 1.7 Το λογότυπο του διαγωνισμού

Ο διαγωνισμός "Field Robot Event" (<http://www.fieldrobot.nl/>) προωθεί το όραμα για την ανάπτυξη των μοντέρνων τεχνολογιών που σχετίζονται με τη Γεωργία Ακρίβειας. Διεξάγεται και οργανώνεται από το 2003 κάθε Ιούνιο σε ετήσια βάση διεθνώς σε διάφορες πόλεις του Ευρωπαϊκού Βορρά και αποτελεί τον μοναδικό υπαίθριο διαγωνισμό αυτόνομων ρομποτικών οχημάτων. Απευθύνεται σε μαθητές λυκείου, φοιτητές, ερευνητές, εταιρίες και γενικότερα σε οποιονδήποτε ενδιαφέρεται για την εισαγωγή του αυτοματισμού στον τομέα της γεωργίας. Δεν υπάρχουν σχεδιαστικοί και κατασκευαστικοί περιορισμοί των αγροτικών ρομπότ αφού στόχος του διαγωνισμού είναι η ενίσχυση της δημιουργικότητας των συμμετεχόντων. Στα πλαίσια του διαγωνισμού επιδιώκεται η ανταλλαγή ιδεών, απόψεων και εμπειριών για την ανάπτυξη αγροτικών ρομποτικών συστημάτων.

Οι ρομποτικές πλατφόρμες διαγωνίζονται σε τέσσερις διαφορετικού τύπου δυσκολίας δραστηριότητες: βασική πλοήγηση και αναβαθμισμένη, εντοπισμός αγριόχορτων και ελεύθερου τύπου δραστηριότητα (freestyle). Κάθε χρόνο υπάρχουν αρκετές αλλαγές στους κανόνες που περιγράφουν τον τρόπο με τον οποίο θα πρέπει να εκτελείται μια δραστηριότητα ούτως ώστε να αναβαθμίζεται το επίπεδο δυσκολίας και να διατηρείται το ενδιαφέρον των διαγωνιζόμενων. Τα πρώτα τρία οχήματα από κάθε δραστηριότητα επιβραβεύονται καθώς και το καλύτερο εξ' αυτών. Στις δυο πρώτες δραστηριότητες μια ομάδα μπορεί να λάβει έως και 5 πόντους, όπου 5 είναι ο αριθμός των ομάδων που συμμετέχουν π.χ. αν αυτές είναι 20, το καλύτερο ρομπότ θα λάβει 20 πόντους, το δεύτερο 19 κ.ο.κ. Αν υπάρξει ισοπαλία ανάμεσα σε δυο ομάδες για παράδειγμα στην τρίτη θέση και οι δυο ομάδες θα λάβουν 18 πόντους ενώ η 5<sup>η</sup> ομάδα θα βαθμολογηθεί με 16. Για την τρίτη και τέταρτη δραστηριότητα κάθε συμμετέχουσα ομάδα μπορεί να λάβει μέχρι 20 πόντους αλλά μόνο η τρίτη προσμετράτε στη συνολική βαθμολογία της ομάδας. Κάθε διαγωνιζόμενη ομάδα θα λάβει τουλάχιστον ένα πόντο για κάθε δραστηριότητα στην οποία συμμετέχει. Παρόλα αυτά αν το σύνολο των πόντων είναι αρνητικό η ομάδα θα βαθμολογηθεί με έναν πόντο για τη συμμετοχή της.

#### 1.3.1 1<sup>η</sup> δραστηριότητα - βασική πλοήγηση

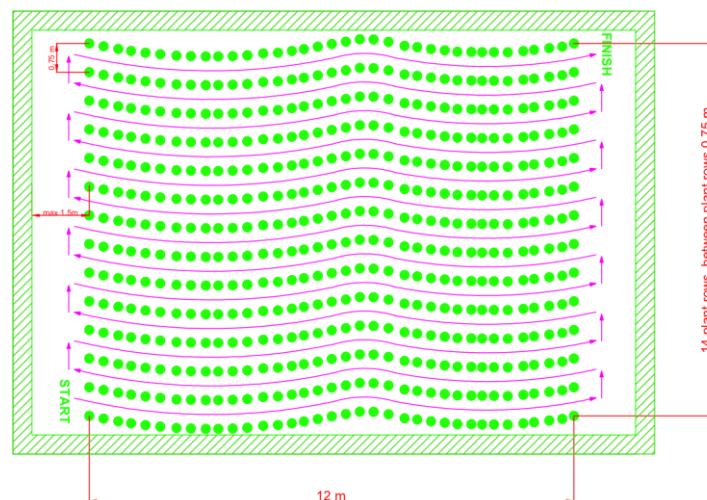
Τον Ιούνιο του 2015 στόχος σε αυτήν την δραστηριότητα ήταν να ελεγχθούν οι βασικές δυνατότητες πλοήγησης του οχήματος σε καλλιεργημένη έκταση. Τα φυτά φυτεύονται σε σταθερές αποστάσεις από τα γειτονικά τους σε ένα προκαθορισμένο σχέδιο (grid pattern). Το όχημα ξεκινά από την αφετηρία ανάμεσα στις συστοιχίες φυτών της πρώτης σειράς και θα πρέπει να κινείται ενδιάμεσα τους. Μόλις φτάσει στο τέλος του φυσικού 'διαδρόμου' που σχηματίζουν τα φυτά θα πρέπει να στρίψει δεξιά (ή αριστερά) σύμφωνα με τις οδηγίες των κριτών, να εισέλθει και να συνεχίσει στη δεύτερη σειρά και στο τέλος της να στρίψει αριστερά (ή δεξιά). Η προαναφερθείσα διαδικασία επαναλαμβάνεται μέχρι να φτάσει στον τερματισμό της καλλιεργημένης περιοχής έχοντας διασχίσει όλες τις σειρές είτε έχει παρέλθει το χρονικό διάστημα των τριών λεπτών από την στιγμή που ξεκίνησε.



**Εικόνα 1.8** Διάταξη φυτών και ο σχηματιζόμενος φυσικός διάδρομος μέσα στον οποίο μετακινούνται τα οχήματα.

Legend:  
● - Corn, height of corn 0.25 - 0.5 m  
→ - Driving direction

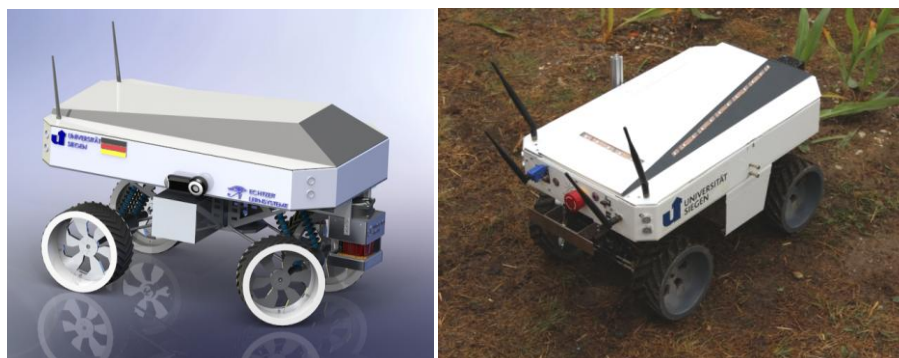
### Task1



**Εικόνα 1.9** Σχέδιο πορείας ρομποτικού οχήματος κατά την 1<sup>η</sup> δραστηριότητα.

Στο χρόνο των τριών λεπτών το όχημα θα πρέπει να έχει διανύσει τη μεγαλύτερη δυνατή απόσταση. Λαμβάνονται θετικοί πόντοι ανάλογα με την διανυθείσα απόσταση εντός των τριών λεπτών. Για κάθε ζημιά σε φυτό αφαιρείται και ένα μέτρο από την απόσταση που έχει διανύσει, ενώ τρία μέτρα αφαιρούνται αν ο συνοδός που εποπτεύει το όχημα κατά την λειτουργία του αναγκαστεί να το αγγίξει. Σε περίπτωση ισοβαθμίας για οχήματα που έχουν καλύψει την ίδια απόσταση βαθμολογείται καλύτερα το όχημα που ολοκλήρωσε την προσπάθεια γρηγορότερα.

Νικητής 1<sup>ης</sup> δραστηριότητας για το έτος 2015 - University of Siegen (Γερμανία)



**Εικόνα 1.10** Ρομποτικό όχημα της ομάδας Zephyr του πανεπιστημίου Siegen με χώρα προέλευσης την Γερμανία

## Βασικές πληροφορίες πλατφόρμας - τεχνικά χαρακτηριστικά

Διαστάσεις (ΠxΜxΥ)	40 x 65 x 50 cm
Βάρος	15 Kg
Σύστημα διεύθυνσης	κίνηση και διεύθυνση και στους τέσσερις τροχούς
Ακτίνα στροφής	75 cm
Μπαταρία	4 cell LiFePo battery (20 Ah)
Κινητήρες	2 x Vector K4 χωρίς ψύκτρες
Ελεγκτής	2 x LPR Truck Puller
Αισθητήρες	Κάμερα, Υπερήχων, Σαρωτής laser, Πυξίδα, Γυροσκόπιο, Επιταχυνσιόμετρο, Kinect V2
Επικοινωνία	TCP/IP
Λειτουργικό Σύστημα	Ubuntu
Προγραμματίστηκε σε	C / C ++ και Python
Προγραμματιστικά εργαλεία - βιβλιοθήκες	Robot Operating System (ROS), OpenCV, Point Cloud Library (PCL)
Κόστος εξαρτημάτων	7000€

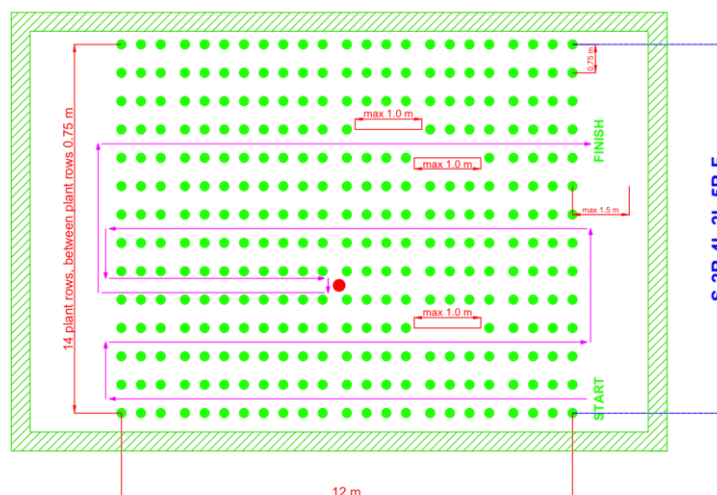
1.3.2 2<sup>η</sup> δραστηριότητα - αναβαθμισμένη πλοήγηση

Δεκαπέντε λεπτά πριν από την έναρξη δίνεται το σχέδιο πορείας βάσει του οποίου το όχημα θα πρέπει να πλοηγηθεί. Στην δραστηριότητα αυτή το όχημα θα συναντήσει εμπόδια (πλαστικοί κώνοι σήμανσης που χρησιμοποιούνται για την προσωρινή εκτροπή της κυκλοφορίας των οχημάτων και την επισήμανση επικίνδυνων σημείων) και θα πρέπει να τα αποφύγει αλλάζοντας πορεία προς την αντίθετη κατεύθυνση καθώς και ενδέχεται σκόπιμα να έχουν αφαιρεθεί φυτά από μια από τις πλευρές του φυσικού διαδρόμου των φυτών σε μέγιστη απόσταση του ενός μέτρου. Η όλη διαδικασία θα πρέπει να ολοκληρωθεί σε τρία λεπτά.

Πόντοι δίνονται μόνο στην περίπτωση που ακολουθηθεί διαδρομή που καθορίζεται στο σχέδιο πορείας. Αν το ρομπότ χάσει μια σειρά ή στρίψει σε λανθασμένη κατεύθυνση δεν λαμβάνονται πόντοι. Σε περίπτωση ισοβαθμίας είτε αφαίρεσης πόντων ισχύουν οι ίδιοι κανόνες που ισχύουν και στην πρώτη δραστηριότητα.

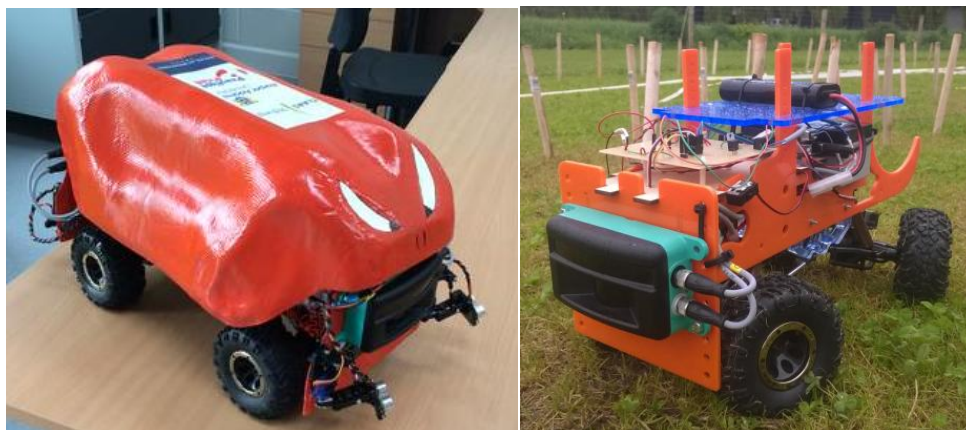
Legend:  
 ● - Corn, height of corn 0.25 - 0.5 m  
 - - Driving direction  
 ● - Obstacle

## Task2



Εικόνα 1.11 Παράδειγμα σχεδίου πορείας ρομπωτικού οχήματος κατά την 2<sup>η</sup> δραστηριότητα

Νικητής 2<sup>ης</sup> δραστηριότητας για το έτος 2015 - Harper Adams University (Ηνωμένο Βασίλειο)



Εικόνα 1.12 Η ρομποτική πλατφόρμα της ομάδας η οποία κατέλαβε την πρώτη θέση στην 2<sup>η</sup> δραστηριότητα

#### Βασικές πληροφορίες πλατφόρμας - τεχνικά χαρακτηριστικά

Διαστάσεις (ΠxΜxΥ)	30 x 50 x 30 cm
Βάρος	7 Kg
Σύστημα διεύθυνσης	Δυνατότητα στροφής του άξονα κάθε ζεύγους τροχών
Ακτίνα στροφής	0.5 m
Μπαταρίες	12V NiMh και 7.2V NiMh
Κινητήρες	Maverick scout drive motor
Ελεγκτής	Arduino Mega 2560
Αισθητήρες	Υπερήχων, Πυξίδα, Multi-ray LED scanner
Επικοινωνία	RS232 με τον σαρωτή
Προγραμματίστηκε σε	Arduino IDE

### 1.3.3 3<sup>η</sup> δραστηριότητα - εντοπισμός άνυδρων φυτών και κατασκευή χάρτη

Η ρομποτική πλατφόρμα πρέπει να εντοπίσει τόσο τα υγιή φυτά όσο και εκείνα που έχουν αποξηραθεί δημιουργώντας ταυτόχρονα ένα χάρτη που να απεικονίζει τη διάταξη των φυτών καλύπτοντας τη μέγιστη δυνατή απόσταση στην καλλιεργημένη έκταση. Όταν ανιχνευθεί ένα άνυδρο φυτό θα πρέπει να παραχθεί ένα οπτικό ή ηχητικό σήμα. Ο χάρτης του αγρού θα πρέπει να είναι αρχείο εικόνας με ελάχιστη ανάλυση δυο εικονοστοιχεία ανά εκατοστό έτσι ώστε να είναι δυνατή η αξιολόγηση του χάρτη κάθε ομάδας. Τα άνυδρα φυτά στην πραγματικότητα είναι υγιή φυτά τα οποία έχουν βαφτεί με καφέ χρώμα λίγο πριν την έναρξη.

Για την κατασκευή του χάρτη θα χρησιμοποιούνται τα χρώματα που παρατίθενται στον παρακάτω πίνακα.

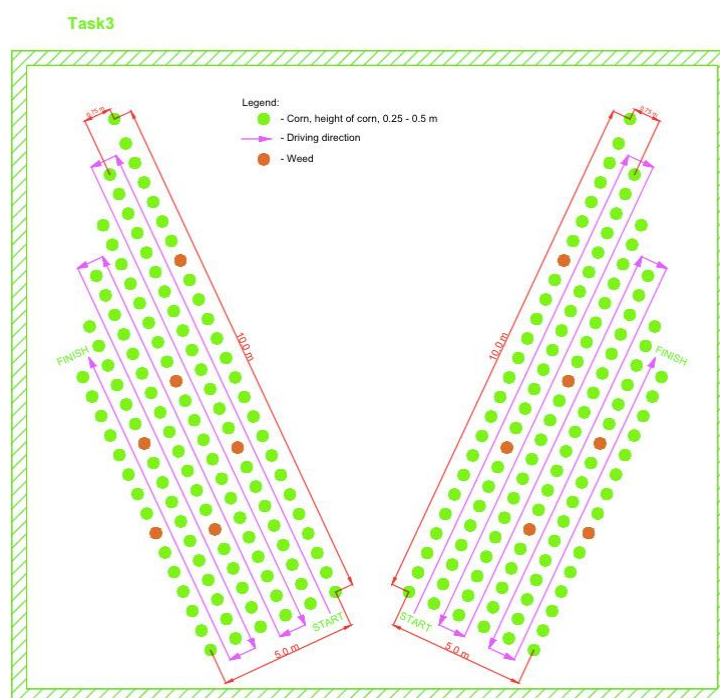
1. Άγνωστη/ανεξερεύνητη περιοχή : γκρι χρώμα
2. Φυτά/εμπόδια/τείχη: μαύρο
3. Εξερευνηθείσα περιοχή: άσπρο
4. Περιοχή άνυδρων φυτών: κόκκινο

Η όλη διαδικασία θα πρέπει να έχει ολοκληρωθεί σε τρία λεπτά συγκεκριμένα να έχει κατασκευαστεί ο χάρτης του περιβάλλοντα χώρου να έχουν ανιχνευθεί τα αποξηραμένα φυτά παράγοντας ταυτόχρονα οπτικό ή ηχητικό σήμα. Η κατανομή της βαθμολογίας γίνεται σε δυο μέρη. Κατά το πρώτο για κάθε ορθή ανίχνευση μη υγιούς φυτού δίνονται δυο πόντοι και συνολικά υπάρχουν 5 τομείς με προβληματικά φυτά προς

εντοπισμό αποδίδοντας συνολικά 10 πόντους. Άλλοι δέκα πόντοι δίνονται στην κατασκευή του χάρτη ο οποίος πληροί τις προαναφερθείσες προδιαγραφές, πέντε για την σωστή αποτύπωση των σειρών και πέντε για τις περιοχές με αποξηραμένο φυτό.

Επιτυχή ανίχνευση θεωρείται όταν το ρομπότ εντοπίσει το άνυδρο φυτό δύο φυτά πριν (περίπου 25cm) και σταματήσει το πολύ δύο φυτά μετά. Ο χάρτης πρέπει να δοθεί το αργότερο μέσα σε 15 λεπτά αφότου το όχημα έχει ολοκληρώσει την πλοήγηση του.

Αφαιρείται μισός βαθμός από τη διανυθείσα απόσταση σε περίπτωση καταστροφής ενός φυτού. Αν αναγκαστεί ο συνοδός να παρέμβει αγγίζοντας το όχημα αφαιρείται ένας πόντος. Η ομάδα χάνει ένα πόντο από τη συνολική της βαθμολογία αν κατά τη διάρκεια της εκτέλεσης το ρομπότ παράξει το οπτικό ή ηχητικό ερέθισμα είτε πολύ νωρίς είτε πολύ αργά. Χάνεται επίσης μισός πόντος στην περίπτωση που δεν απεικονιστεί σωστά στο χάρτη η περιοχή με το προβληματικό φυτό σε σχέση με τον ορθό χάρτη αναφοράς.

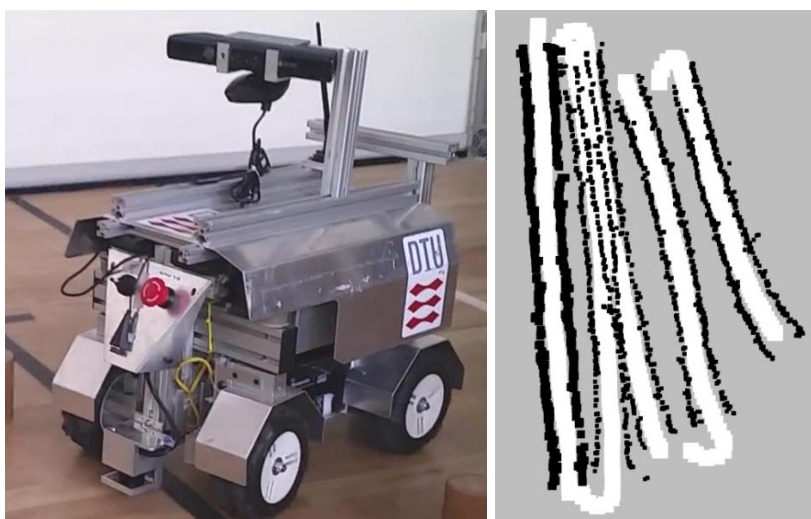


Εικόνα 1.13 3<sup>η</sup> δραστηριότητα, εντοπισμός προβληματικών φυτών και κατασκευή χάρτη της περιοχής.



Εικόνα 1.14 Δείγματα των χρωματισμένων φυτών που θα ανιχνεύονταν ως προβληματικά

Νικητής 3<sup>ης</sup> δραστηριότητας για το έτος 2015 - Πολυτεχνείο Δανίας



Εικόνα 1.15 Ρομποτικό όχημα της ομάδας DTUπi-Corn και ο χάρτης που παράχθηκε

Βασικές πληροφορίες πλατφόρμας - τεχνικά χαρακτηριστικά

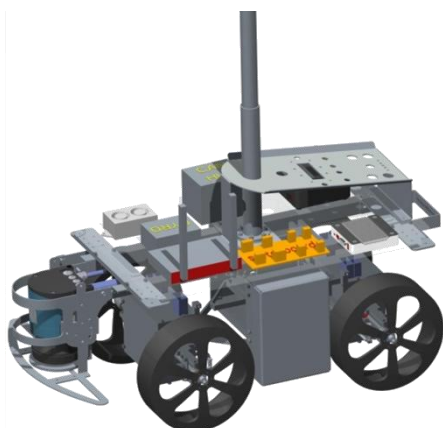
Διαστάσεις (ΠxΜxΥ)	37 x 82 x 47 cm
Βάρος	20 Kg
Σύστημα διεύθυνσης	Ackerman
Μπαταρίες	Δύο 12V συνδεδεμένες παράλληλα
Κινητήρες	Σερβοκινητήρες για τις μπροστινούς κατευθυντήριους τροχούς και δυο ηλεκτροκινητήρες για τους πίσω (differential drive)
Αισθητήρες	Kinect, 2 σαρωτές Laser (ένας μπροστά και ο άλλος πίσω), Γυροσκόπιο
Επικοινωνία	Ethernet
Λειτουργικό Σύστημα	Mobotware
Προγραμματίστηκε σε	SMR-CL

### 1.3.4 4<sup>η</sup> δραστηριότητα - freestyle

Η δραστηριότητα αυτή επαφίεται στην φαντασία, δημιουργικότητα και την εφευρετικότητα κάθε ομάδας. Στόχος είναι να αναβαθμιστεί και να βελτιωθεί η πλατφόρμα ώστε να επιτελεί έργο που να σχετίζεται με τη γεωργία ακριβείας.

Κατανομή πόντων

- μέχρι 5 πόντους για την πρωτοτυπία της ιδέας
- μέχρι 5 για την χρηστικότητα
- έως 5 για την πολυπλοκότητα
- και έως 5 πόντους για την βέλτιστη απόδοση



Εικόνα 1.16 CAD μοντέλο και η ομάδα GroundBreaker του Πανεπιστημίου του Ελσίνκι.

Η ομάδα αυτή βραβεύτηκε σ' αυτήν την δραστηριότητα για το ρυμουλκούμενο όχημα το οποίο ανέπτυξε και το οποίο έφερε απαραίτητο αυτοματοποιημένο σύστημα ούτως ώστε να πραγματοποιείται δειγματοληψία του εδάφους και να μετρούνται οι ιδιότητες του. Τα αποτελέσματα των μετρήσεων λαμβάνονται σε tablet σε πραγματικό χρόνο. Το ρυμουλκούμενο όχημα έφερε εισδυσιόμετρο ώστε να μετράται ο δείκτης κώνου και αυτοματοποιημένο σύστημα με μάντα περισυλλογής για την λήψη δειγμάτων εδάφους και τοποθέτησης του σε ειδικά δοχεία.

#### Βασικές πληροφορίες πλατφόρμας - τεχνικά χαρακτηριστικά

Διαστάσεις (ΠxΜxΥ)	39 x 75 x 56 cm
Βάρος	19 Kg
Σύστημα διεύθυνσης	4 τροχοί με σερβοκινητήρα σε κάθε άξονα κίνησης
Ακτίνα στροφής	0.45 m
Μπαταρία	3 x 5000mAh LiPo
Κινητήρες	2 x 12V DC motors
Ελεγκτής	Υβριδικός έλεγχος, συνδυασμός PC και μικροελεγκτών
Αισθητήρες	3 x Κάμερα, 4 x υπερύχων, 4 x υπερύθρων, Σαρωτής Laser, Γυροσκόπιο 3 αξόνων, GPS, κωδικοποιητές θέσης στα μοτέρ κίνησης
Επικοινωνία	CAN, RS485 και Ethernet
Λειτουργικό Σύστημα	Windows CE και 7
Προγραμματίστηκε σε	C#, Simulink code generation, C++
Προγραμματιστικά εργαλεία - βιβλιοθήκες	Visual Studio, Matlab, CodeVision AVR, SVN
Κόστος εξαρτημάτων	4000€
Ώρες εργασίας	6000h



## 1.4 Συστήματα ανάρτησης

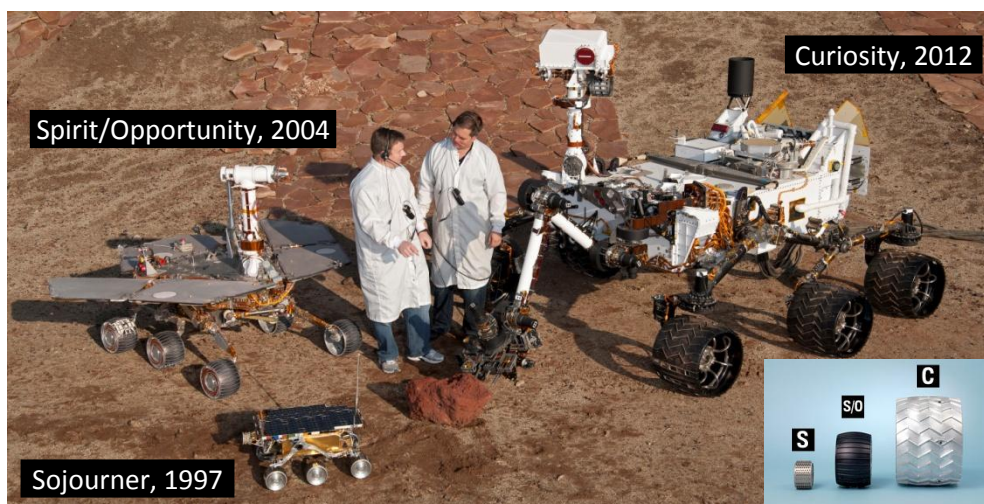
Ένα τροχοφόρο όχημα δυσκολεύεται να υπερβεί εμπόδιο του οποίου το ύψος του είναι μεγαλύτερο από την ακτίνα του τροχού του. Για να αντιμετωπιστεί αυτό το πρόβλημα θα πρέπει να χρησιμοποιηθεί ένα κατάλληλο σύστημα ανάρτησης. Αν και υπάρχει μια μεγάλη ποικιλομορφία στους τύπους των αναρτήσεων των επιβατικών αυτοκινήτων, σε όλα χρησιμοποιούνται κάποιες μορφής ελατήρια (σούστες) και αποσβεστήρες κραδασμών (αμορτισέρ) ώστε το όχημα εκτός από την άνεση που παρέχει στους επιβαίνοντες να διατηρεί και την ευστάθεια του όταν π.χ. το οδόστρωμα παρουσιάζει ανωμαλίες ή κατά την αλλαγή πορείας. Τα περισσότερα συστήματα αναρτήσεων έχουν σχεδιαστεί ώστε να φροντίζουν την ομαλή και ασφαλή οδική συμπεριφορά του αμαξώματος σε μεγάλες ταχύτητες και κυρίως σε ομαλές επιφάνειες. Παρά την ευρεία χρήση τους και τις αξιόλογες επιδόσεις τους στο χώρο της αυτοκινητοβιομηχανίας ελάχιστα χρησιμοποιούνται σε ρομποτικές εφαρμογές. Οι τύποι αυτοί των αναρτήσεων είναι κατάλληλοι για οχήματα που κινούνται με ταχύτητα μεγαλύτερη των 8 m/sec περίπου. Κάτω από αυτό το όριο δυσχεραίνουν την κίνηση αφού αλλάζουν τη δύναμη που ασκεί κάθε τροχός στο έδαφος κατά το πέρασμα ενός εμπόδιου. Σ' ένα τετράτροχο όχημα με συμβατικού τύπου σύστημα αναρτήσεων όλοι οι τροχοί εμφανίζονται να "πατάνε" καλά στο έδαφος, αλλά αν κάποιος τροχός συναντήσει εμπόδιο ανασηκώνεται και φέρει μεγαλύτερο βάρος από τους υπόλοιπους με αποτέλεσμα οι λιγότερο επιβαρυσμένοι να χάνουν και την πρόσφυση τους με το οδόστρωμα. Μια καλύτερη λύση στις χαμηλές ταχύτητες είναι να επιτραπεί η ανύψωση σε κάποιους από τους τροχούς σε σχέση με το πλαίσιο (σασί) κατά την υπέρβαση του εμποδίου χωρίς να αλλάξει η κατανομή βάρους ή αν αλλάξει, να αλλάξει το λιγότερο δυνατό. Αυτό μπορεί να επιτευχθεί χρησιμοποιώντας ένα σύστημα παθητικής ανάρτησης αρθρωτού τύπου (passive articulated suspension). Ένα αξιοσημείωτο παράδειγμα αποτελεί η ανάρτηση τύπου Rocker-Bogie (Bickler, 1998) η οποία παρουσιάστηκε επιτυχώς από τη NASA στα ρόβερ Spirit και Opportunity για την εξερεύνηση πλανητών (Maimone et al., 2004).

Οι παθητικές αναρτήσεις αποτελούνται από άκαμπτες ράβδους αλλά και συνδέσμους ελεύθερης περιστροφής προσδίδοντας στο δικτύωμα έναν ή περισσότερους βαθμούς ελευθερίας (DoF). Στα συστήματα αυτά το σασί προσαρμόζεται στο έδαφος με βάση βαρυτικές δυνάμεις που κατανέμουν ομαλά το φορτίο. Τέτοιες διατάξεις είναι ιδανικές όταν εξετάζουμε ένα σύστημα στη μόνιμη κατάσταση (steady state) αγνοώντας την δυναμική του όπως συχνά συμβαίνει με οχήματα που κινούνται με χαμηλή ταχύτητα. Σε υψηλές ταχύτητες οι δυναμικές αναρτήσεις χρησιμοποιούνται για την απορρόφηση των κραδασμών ωστόσο οι βαθμοί ελευθερίας τους καθορίζονται με κριτήριο τα ελατήρια και τους αποσβεστήρες που διαθέτουν ή και από τον ενεργό έλεγχο τους.

Δείκτη για την επιλογή ανάμεσα σε παθητικό ή δυναμικό σύστημα ανάρτησης αποτελεί ο λόγος μεταξύ της κινητικής και της δυναμικής ενέργειας:

$$\delta = \frac{\frac{1}{2}mv^2}{mgh} = \frac{v^2}{2gh}$$

όπου  $v$  η ταχύτητα του οχήματος σε  $m/s$ ,  $g$  η επιτάχυνση της βαρύτητας σε  $m/s^2$  και  $h$  είναι η διαφορά στο ύψος του κέντρου βάρους του οχήματος όταν αυτό υπερπηδά εμπόδιο μέγιστου ύψους ίσου σχεδόν το μισό της διαμέτρου του τροχού. Όταν ο δείκτης αυτός είναι μικρότερος από ένα θα πρέπει να χρησιμοποιείται σύστημα παθητικής ανάρτησης.

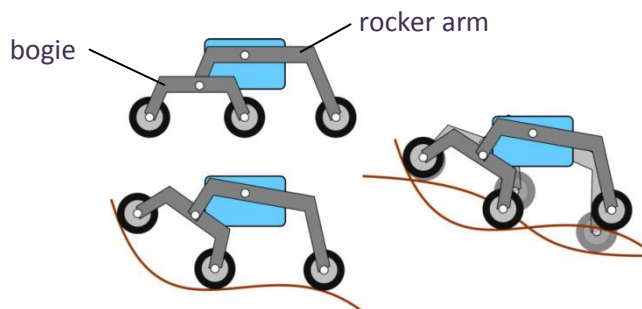


Εικόνα 1.17 Ρόβερ τα οποία έχει αποστείλει η NASA στον πλανήτη Άρη για την εξερεύνηση του.

#### 1.4.1 Σύστημα ανάρτησης τύπου Rocker-Bogie

Το ιδιαίτερο χαρακτηριστικό της είναι ότι δεν ενσωματώνει κανένα από τα τυπικά στοιχεία που συναντούμαι σε συνήθη συστήματα ανάρτησης. Η ανάρτηση αποτελείται από ένα δικτύωμα το οποίο έχει δυο βραχίονες (rocker arms), έναν σε κάθε μεριά του κυρίου άξονα (ή μπάρα) της ανάρτησης ο οποίος διέρχεται από το κέντρο βάρους του οχήματος κάθετα στον διαμήκη του άξονα. Κάθε rocker arm έχει ένα τροχό συνδεδεμένο με κινητήρα στο ένα άκρο του και στο άλλο άκρο συνδεδεμένο ένα δευτερεύων "μπράτσο" που ονομάζεται bogie. Σε κάθε άκρο του βραχίονα bogie είναι τοποθετημένοι τροχοί συνδεδεμένοι με κινητήρες και το bogie είναι συνδεδεμένο με το αντίστοιχο rocker arm με ένα σύνδεσμο ελεύθερης περιστροφής. Η ανάρτηση επιτρέπει και στους έξι τροχούς του οχήματος να αλλάζουν διεύθυνση, ενώ σε μια συντηρητική εκδοχή της εκ των έξι, οι τέσσερις τροχοί (στο εμπρός και πίσω μέρος) έχουν δυνατότητα αλλαγής κατεύθυνσης.

Ο κύριος άξονας της ανάρτησης περιέχει αντιστροφόμενο διαφορικό (counter rotating differential) ώστε η γωνία κλίσης του κυρίου σώματος να είναι ο μέσος όρος των γωνιών των δυο rocker arms. Ο παθητικός μηχανισμός του διαφορικού που συνδέει τα μπράτσα (rocker arms) των δύο πλευρών και ο σύνδεσμος ελεύθερης περιστροφής τους με το bogie δίνουν 3 βαθμούς ελευθερίας συνολικά. Οι σύνδεσμοι της ανάρτησης επιτρέπουν την περιστροφή ώστε οι τροχοί να ακολουθούν το έδαφος προσδίδοντας στο όχημα τη μέγιστη δυνατή σταθερότητα σε ανώμαλες επιφάνειες. Έτσι η δύναμη που ασκείται σε κάθε τροχό είναι σχεδόν ίση. Σε ένα όχημα με τυπική ελαστική ανάρτηση αν ένας από τους τροχούς βρεθεί πιο ψηλά από τους υπόλοιπους λόγω ενός εμποδίου έχει ως συνέπεια την μείωση της εφαρμοζόμενης δύναμης στους υπόλοιπους με αποτέλεσμα την μείωση της πρόσφυσης τους. Το αποτέλεσμα της εξισορρόπησης και σταθεροποίησης των ασκούμενων δυνάμεων στους τροχούς είναι η ελαχιστοποίηση των μετατοπίσεων του κυρίου σώματος. Η κάθετη μετατόπιση του γεωμετρικού κέντρου του κυρίου σώματος είναι ίση με το μέσο όρο των μετατοπίσεων των τροχών. Η ανάρτηση αυτού του τύπου δίνει στο όχημα τη μέγιστη δυνατή απόσταση μεταξύ του κυρίου σώματος και του εδάφους (ground clearance) πράγμα που καθιστά εύκολη την υπέρβαση εμποδίων με ύψος αρκετά μεγαλύτερο των τροχών του και τη δυνατότητα ελιγμών που δύσκολα πραγματοποιούνται με τα συνήθη συστήματα.



Εικόνα 1.18 Σύστημα ανάρτησης τύπου Rocker-Bogie.

Στην πραγματικότητα βέβαια τα ακριβή πλεονεκτήματα των εξάτροχων οχημάτων με σύστημα ανάρτησης τύπου rocker bogie παραμένουν αδιευκρίνιστα. Οι προηγούμενες αποστολές της NASA δεν έχουν αξιοποιήσει πλήρως τις δυνατότητες του συστήματος και κάποιες μελέτες αποδεικνύουν ότι τα τετράτροχα ρόβερ μπορούν να επιτύχουν παρεμφερείς στόχους απλά αποφεύγοντας διαδρομές με ογκώδη εμπόδια (Roman, 2005).

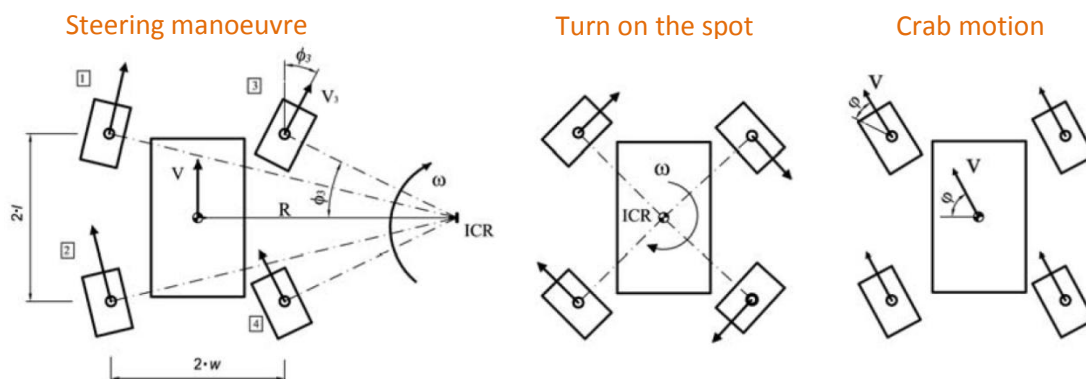
Ένα σύστημα ανάρτησης τέτοιου τύπου με απλούστερη σχεδίαση μπορεί να υλοποιηθεί και σε τετράτροχα οχήματα με το πλεονέκτημα της χρήσης λιγότερων τροχών, κινητήρων, συνδέσμων και αξόνων κτλ. Ακολούθως παρουσιάζονται δυο τέτοιες ερευνητικές προσπάθειες όπως αυτές καταγράφονται στην διεθνή βιβλιογραφία.

#### 1.4.1.1 Το ρομποτικό ρόβερ Dune

Το ρόβερ Dune κατασκευάστηκε στο Πανεπιστήμιο του Salento σε συνεργασία το Πολυτεχνείο του Bari (Ιταλία) και πρόκειται για ένα τετράτροχο όχημα με 4 κινητήριους τροχούς καθένας εκ των οποίων μπορεί να αλλάζει κατεύθυνση με συνέπεια το όχημα να έχει τη δυνατότητα εκτέλεσης μεγάλης γκάμας ελιγμών.

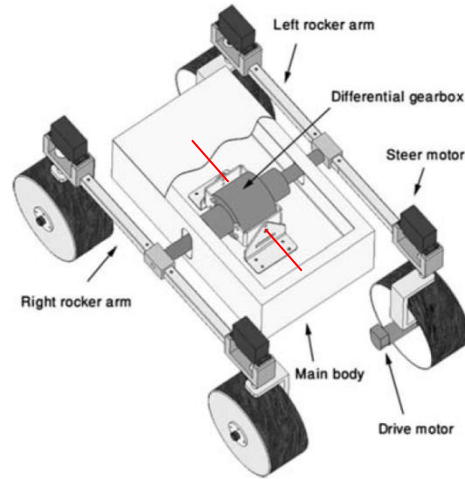
##### Βασικά χαρακτηριστικά οχήματος

Διαστάσεις	74 x 45 x 32 cm
Διάμετρος τροχού	20 cm
Απόσταση από το έδαφος (ground clearance)	21 cm
Βάρος	16 Kg
Μέγιστη ταχύτητα	150 cm/s
Μπαταρία	2 Kg NiCad 100Wh

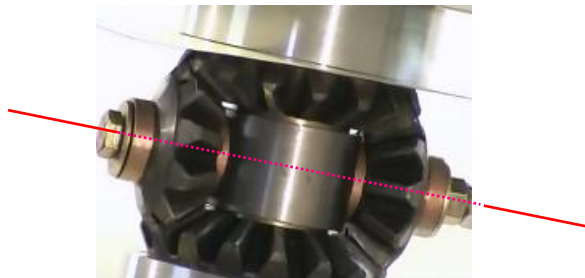


Εικόνα 1.19 Εφικτοί ελιγμοί του ρόβερ Dune

Το συγκεκριμένο ρόβερ φέρει μια απλουστευμένη εκδοχή του συστήματος ανάρτησης rocker-bogie. Πρωταρχικός σκοπός του συστήματος ανάρτησης είναι η εξασφάλιση ότι και οι τέσσερις τροχοί θα παραμένουν συνεχώς σε επαφή με το έδαφος κατά την κίνηση του οχήματος σε επιφάνειες που παρουσιάζουν ανωμαλίες. Ανά δυο οι κινητήριοι τροχοί σε κάθε πλευρά και κατά μήκος του οχήματος συνδέονται μέσω ενός άκαμπτου συνδέσμου (rocker arm). Κάθε μπράτσο περιστρέφεται σε σχέση με το κύριο πλαίσιο μέσω ενός άξονα που περιέχει διάταξη με αντιπεριστρεφόμενο διαφορικό. Ο μηχανισμός (γρναζοκιβώτιο) αυτός είναι σταθεροποιημένος στο κυρίως σώμα του οχήματος. Ο τρόπος σύνδεσης επιτρέπει σε κάθε rocker arm να υφίσταται περιστροφή όταν ένας από τους τροχούς ανέρθει ή κατέρθει λόγω εμποδίου.



Εικόνα 1.20 Το ρόβερ Dune. Στο CAD μοντέλο απεικονίζονται τα βασικά δομικά μέρη του συστήματος ανάρτησης.



Εικόνα 1.21 Παράδειγμα μηχανισμού αντιπεριστρεφόμενου διαφορικού (counter rotating differential)

Η παθητική κίνηση περιστροφής διατηρεί το φορτίο και στους τέσσερις τροχούς σχεδόν ίσο προσδίδοντας στο όχημα ευστάθεια αφού κάθε τροχός δεν χάνει την πρόσφυση του με το έδαφος και δίνει την ικανότητα στο όχημα να σκαρφαλώνει εμπόδια με ύψος αρκετά μεγαλύτερο των τροχών του. Επιπλέον ο περιορισμός που υπεισέρχεται με την ύπαρξη του διαφορικού προσδίδει κλίση (pitch) στο κυρίως σώμα του οχήματος ίση με το μέσο όρο των γωνιών κλίσης κάθε μπράτσου με συνέπεια καθώς κάθε rocker arm κινείται πάνω σε εμπόδια το κυρίως σώμα του οχήματος διατηρείται σε περισσότερο επίπεδη θέση σε σχέση με αυτά. Συνεπώς αν η γωνία κλίσης για το δεξί μπράτσο είναι  $\theta_R$  και για το αριστερό  $\theta_L$  τότε η κλίση του αμαξώματος θα είναι:

$$\theta = \frac{\theta_R + \theta_L}{2}$$

Η συμπεριφορά αυτή είναι επωφελής για οχήματα που πλοηγούνται μέσω κάμερας (vision based control) και γενικότερα σε αυτόνομα ρομποτικά οχήματα οδηγούμενα από αισθητήρες λόγω των μειωμένων ταλαντώσεων του κυρίως σώματος τους σε σχέση με το σασί τους.

#### 1.4.1.2 Η ρομποτική πλατφόρμα εξερεύνησης Oryx 2.0



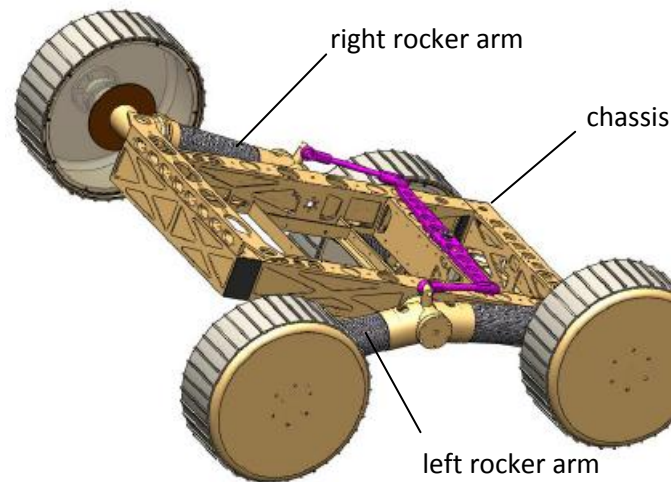
Εικόνα 1.22 Το ρόβερ Oryx 2.0

Το Oryx 2 είναι δημιούργημα μιας ομάδας φοιτητών του Πολυτεχνείου Worcester (Η.Π.Α.) (<http://wpirover.com/>). Σχεδιάστηκε ως ερευνητική πλατφόρμα ικανή να φέρει κατάλληλο εξοπλισμό για εξερεύνηση επιφανείας με σχεδιαστικές προδιαγραφές τέτοιες ώστε να μπορεί να ανταπεξέλθει σε άγνωστο, εχθρικό και δύσβατο έδαφος όπως εκείνα του πλανήτη Άρη και της Σελήνης αλλά και σε γήινο περιβάλλον κάτω από δύσκολες συνθήκες.

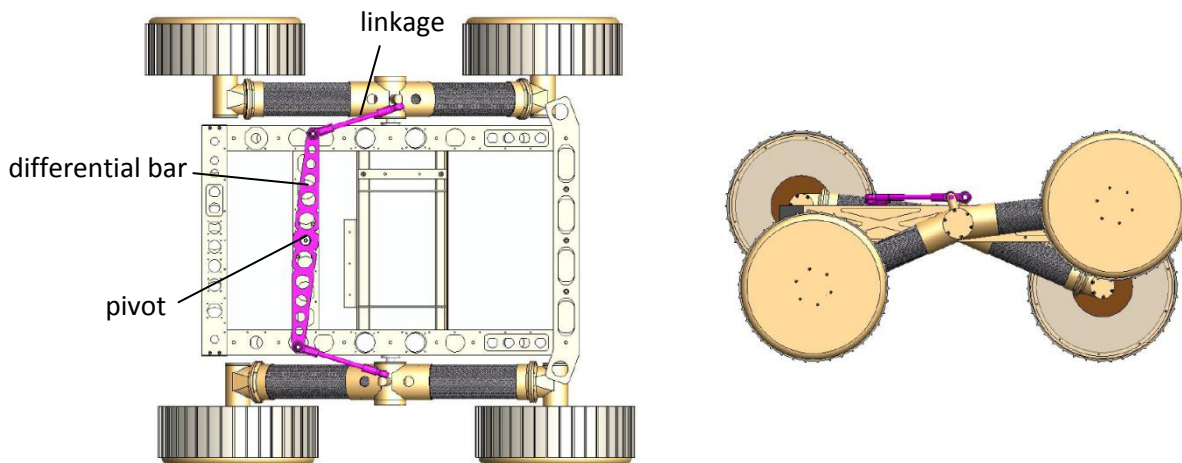
Διαθέτει τέσσερις κινητήριους τροχούς οι οποίοι παραμένουν ευθυγραμμισμένοι με το διαμήκη άξονα του ρόβερ και οποιαδήποτε στροφή επιτυγχάνεται με διαφορετική ταχύτητα στα δυο ζεύγη τροχών και αναγκαστική πλαγιολίσθηση τους (skid-steering). Όταν θα πρέπει να ακολουθηθεί μια συγκεκριμένη τροχιά για την βελτιστοποίηση της πορείας του μια διαφορετική στρατηγική ελέγχου είναι σε κάθε τροχό μεμονωμένα να εφαρμόζεται και διαφορετική ταχύτητα.

Το δικτύωμα της ανάρτησης που επιλέχθηκε αποτελείται μια μπάρα (differential bar) η οποία στο μέσο της ενώνεται με το υπόλοιπο σασί με σύνδεσμο ελεύθερης περιστροφής και είναι κάθετη στον διαμήκη άξονα του οχήματος και παράλληλη με το πάνω μέρος του πλαισίου. Κάθε απόληξη της μπάρας αυτής ενώνεται μέσω συνδετικής ράβδου με το μπράτσο (rocker arm) της αντίστοιχης πλευράς (Εικόνα 1.23, 1.24). Οι συνδετικές ράβδοι διαθέτουν στα άκρα τους κατάλληλες σφαιρικές αρθρώσεις. Κάθε μπράτσο φέρει δυο κινητήρες, έναν σε κάθε άκρο.

Το όχημα διαθέτει δικτύωμα παθητικής ανάρτησης με ένα βαθμό ελευθερίας (όταν το ένα μπράτσο ανέρχεται το άλλο κατέρχεται) ώστε το σασί να προσαρμόζεται ανάλογα στη μορφολογία του εδάφους. Χωρίς την ομαλή κατανομή του φορτίου που παρέχει το παθητικό σύστημα ανάρτησης κάποιοι τροχοί θα δέχονταν υπέρμετρη κάθετη δύναμη με αποτέλεσμα την βύθιση τους σε μαλακό έδαφος και κάποιοι άλλοι ίσως να έχαναν την πρόσφυση τους με το έδαφος κατά την διέλευση του οχήματος από εμπόδιο. Επιπρόσθετα λόγω του διαφορεικού το κυρίως σώμα του οχήματος σταθμίζεται μεταξύ του μέσου όρου κλίσης των δυο πλευρικών μπράτσων του σασί.



Εικόνα 1.23 Θέση του συστήματος της παθητικής ανάρτησης με τον μπροστά δεξιά τροχό πάνω σε εμπόδιο.



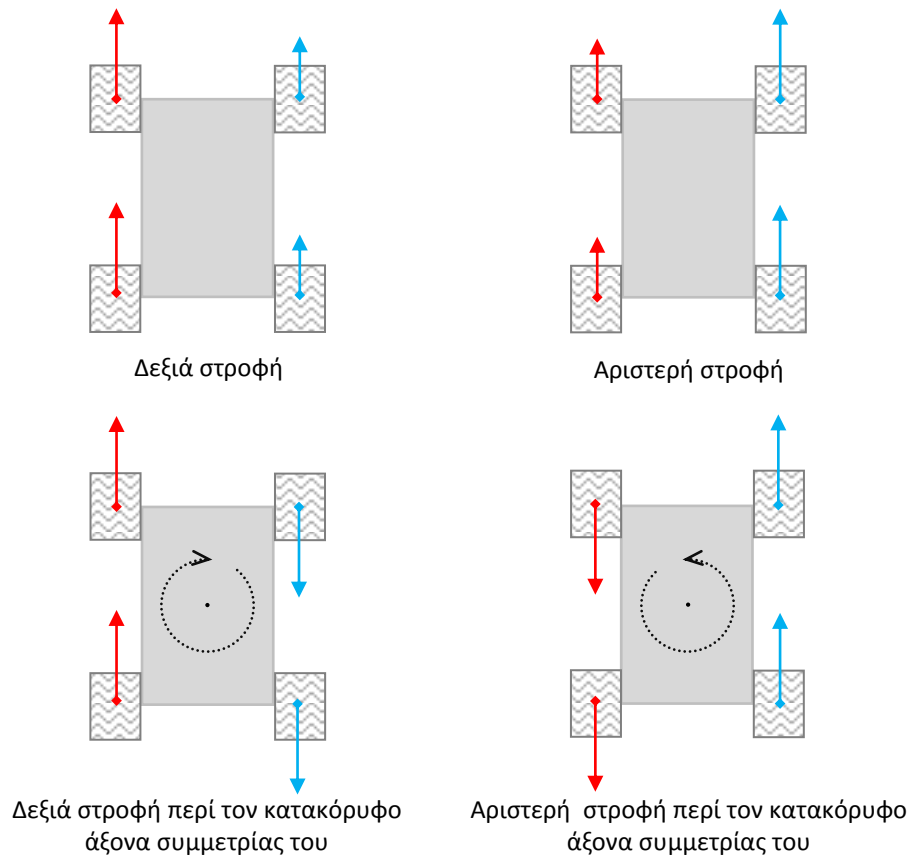
Εικόνα 1.24 Μέγιστη επιτρεπόμενη θέση του rocker arm (κάτοψη και δεξιά όψη)

#### Κατασκευαστικά χαρακτηριστικά - τεχνικές προδιαγραφές

Διαστάσεις (L x W x H)	96 x 89 x 31 cm
Βάρος	35 Kg
Ονομαστικό βάρος φορτίου	15 Kg
Μέγιστη ταχύτητα	1.2 m/s
Μέγιστο ύψος εμπόδιου	15 cm
Χρόνος λειτουργίας	~ 3hrs
Κινητήρια ισχύ	Έως 400W συνεχόμενα
Αισθητήρες	Τάση μπαταριών, Θερμοκρασιών συστήματος, Προσανατολισμού ρόβερ, Οδομετρίας, Θέσης βραχιόνων (rocker arms), Χειρισμού σφαλμάτων
Μπαταρία	22.5V 36Ah Lithium Ion w/BMS
On board computer	Water-cooled Intel Quad-core i5 processor on Mini-ITX motherboard
Επικοινωνία	USB 2.0 (8) / Gigabit Ethernet (2) /Wi-Fi / 4G LTE
Λογισμικό	ROS

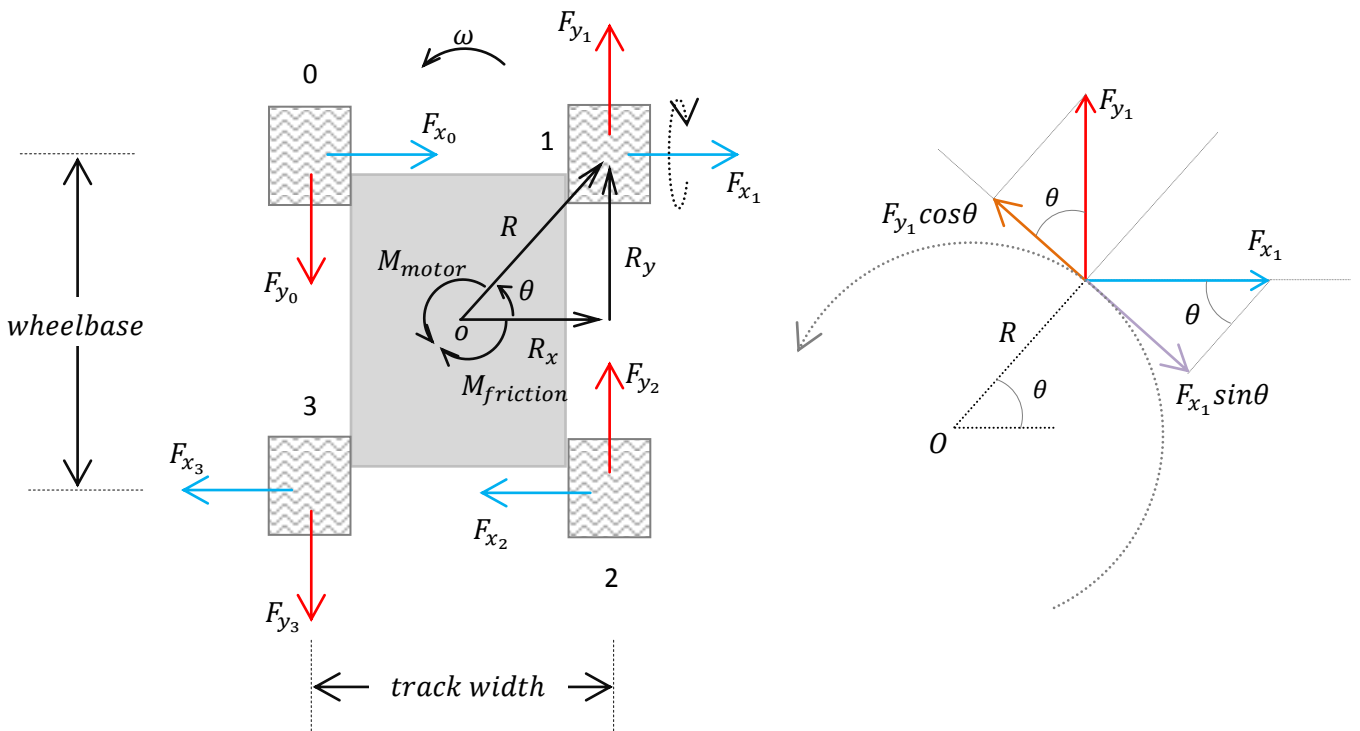
## 1.5 Αλλαγή κατεύθυνσης με πλαγιολίσθηση (Skid-Steering)

Ελιγμοί οχημάτων των οποίων όλοι οι τροχοί τους είναι ευθυγραμμισμένοι με τον διαμήκη άξονα τους και δεν υπάρχουν μηχανισμοί ή επενεργητές που να τους δίνουν την δυνατότητα αλλαγής κατεύθυνσης, επιτυγχάνεται επιβάλλοντας διαφορετικές ταχύτητες σε κάθε πλευρά τους. Στροφές διαφόρων ακτινών πραγματοποιούνται ελέγχοντας τη διαφορά ταχυτήτων μεταξύ των δυο πλευρών του. Όταν οι ταχύτητες είναι ίσες σε κάθε πλευρά αλλά αντίθετης κατεύθυνσης από πλευρά σε πλευρά, επιτυγχάνεται περιστροφή του οχήματος γύρω από το γεωμετρικό του κέντρο, πραγματοποιώντας συνεπώς αλλαγή κατεύθυνσης χωρίς μεταφορική κίνηση.



**Εικόνα 1.25** Εφικτοί ελιγμοί τετράτροχου οχήματος με πλαγιολίσθηση

Πλεονεκτήματα αποτελεί η απλοϊκή μηχανολογική σχεδίαση με αποτέλεσμα την ευρωστία και το λιγότερο βάρος. Η αλλαγή κατεύθυνσης με πλαγιολίσθηση έχει δυο μειονεκτήματα σε σχέση με άλλες μεθόδους ελέγχου, την δυσκολία στην ακριβή παρακολούθηση συγκεκριμένης τροχιάς καθώς δεν είναι εφικτό να προσδιοριστεί η μετακίνηση λόγω των ακαθόριστων συμπεριφορών και αλληλεπιδράσεων μεταξύ των τροχών και του εδάφους αλλά και η σημαντική υψηλή κατανάλωση ενέργειας κατά την διενέργεια ελιγμών που οφείλεται στην σύμπτυξη και ισοπέδωση του εδάφους που υπάρχει κάτω από τους τροχούς από την πλευρική ολίσθηση τους. Η κατανάλωση ενέργειας αυξάνει όσο μικραίνει η ακτίνα στροφής και είναι δύσκολο να καθοριστεί το ποσό της. Κατά την περιστροφή γύρω από τον άξονα συμμετρίας του βρέθηκε ότι το όχημα που πλαγιολισθαίνει καταναλώνει ενέργεια δυο φορές περισσότερη από τα εκείνα τα οχήματα που διαθέτουν κατάλληλους μηχανισμούς για περιστροφή των τροχών τους. Ωστόσο για κινήσεις χωρίς αλλαγή πορείας ή στροφών με μεγάλη ακτίνα η κατανάλωση είναι συγκρίσιμη.



Εικόνα 1.26 Ασκούμενες δυνάμεις στο όχημα κατά την περιστροφική κίνηση

Στη μόνιμη κατάσταση (steady state) και κατά την περιστροφή περί τον άξονα συμμετρίας στο όχημα εμφανίζονται δυο κατηγορίες δυνάμεων. Οι διαμήκεις δυνάμεις ( $F_y$ ) που παράγονται από τους κινητήριους τροχούς και δημιουργούν την ροπή στρέψης και οι πλευρικές δυνάμεις ( $F_x$ ) που αντιστέκονται στην περιστροφική κίνηση. Στην ισορροπία τα αλγεβρικά αθροίσματα των ροπών των δυνάμεων αυτών είναι μηδέν οπότε ισχύει η σχέση

$$\sum_l F_{y_i} \cos\theta = \sum_l F_{x_i} \sin\theta$$

Η αναλογία μεταξύ του μετατρόχιου (track width) και του μεταξονίου (wheelbase)<sup>2</sup> είναι συνάρτηση της γωνίας  $\theta$ . Μικρή γωνία  $\theta$  αυξάνει τις διαμήκεις δυνάμεις ( $F_y$ ) σε σχέση με τις πλευρικές δυνάμεις ( $F_x$ ) αντίστασης κάνοντας έτσι την περιστροφή πιο αποδοτική.

## 1.6 Συνεισφορά της παρούσας εργασίας

Αποτέλεσμα αυτής της εργασίας αυτής ήταν η ανάπτυξη και κατασκευή ενός τετράτροχου ρομποτικού οχήματος διαφορικής οδήγησης (differential drive) με σύστημα παθητικής ανάρτησης αρθρωτού τύπου (passive articulated suspension) το οποίο αποτελεί απλουστευμένη εκδοχή των εξάτροχων οχημάτων με σύστημα ανάρτησης τύπου rocker-bogie. Πληθώρα φωτογραφιών της ρομποτικής πλατφόρμας παρατίθενται στο παράρτημα Α της παρούσας.

Εκτός από το χαμηλό κόστος πρωτεύοντα ρόλο στην διαδικασία ανάπτυξης έπαιξε και η διατήρηση του συνολικού βάρους της πλατφόρμας σε όσο το δυνατόν χαμηλά επίπεδα ώστε να υπάρχει αυξημένη ενεργειακή αυτονομία και να επιτυγχάνεται η μικρότερη δυνατή συμπίεση του εδάφους. Τα σχεδιαστικά κριτήρια καθορίστηκαν επίσης και από τις προδιαγραφές του διεθνούς διαγωνισμού "Robot Field Event".

<sup>2</sup> μετατρόχιο καλείται η απόσταση μεταξύ των δυο τροχών ενώ μεταξόνιο καλείται η απόσταση μεταξύ των αξόνων



Το όχημα διαθέτει τέσσερις κινητήριους τροχούς και κάθε προσπάθεια αλλαγής στην κατεύθυνση του επιτυγχάνεται με την επιβολή διαφορετικών ταχυτήτων στο ζεύγος των κινητήρων κάθε πλευράς. Το ρόβερ είτε κινείται προς τα εμπρός είτε με την όπισθεν έχει την δυνατότητα να πραγματοποιήσει οποιοδήποτε ελιγμό δύναται να επιτύχει ένα τετράτροχο όχημα με διαφορεική οδήγηση.

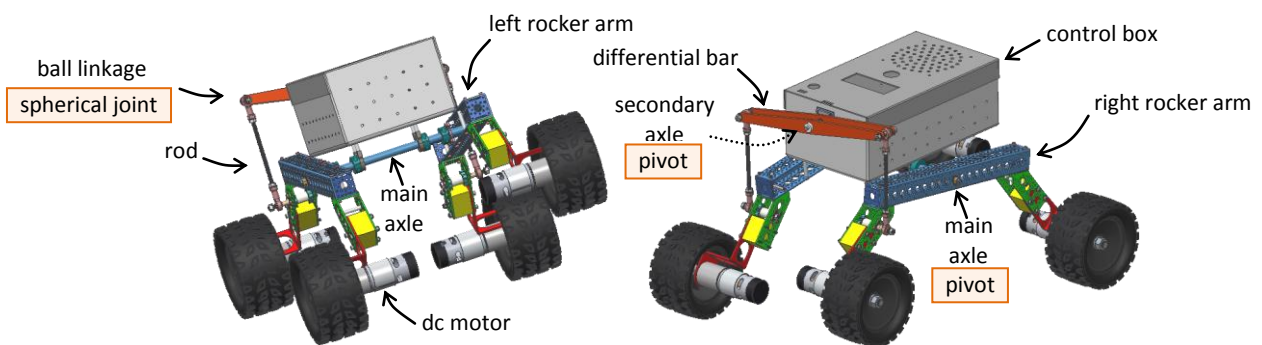
Η ρομποτική πλατφόρμα κινείται είτε με τηλεχειρισμό μέσω ασύρματης ζεύξης Bluetooth οπότε οι αντίστοιχες εντολές κίνησης αποστέλλονται από συσκευή (π.χ. κινητό τηλέφωνο, Η/Υ) η οποία διαθέτει ανάλογο πομποδέκτη είτε αυτόνομα κατά την διάσχιση διαδρομής εκατέρωθεν της οποίας υπάρχουν συστοιχίες φυτών κάνοντας αποκλειστικά χρήση οπτικής πληροφορίας (vision based control).

Το δικτύωμα της ανάρτησης αποτελείται από δυο βραχίονες (rocker arms) τα άκρα των οποίων φέρουν τους κινητήρες και συνδέονται μεταξύ τους μέσω κυλινδρικού συμπαγούς άξονα ελεύθερης περιστροφής. Στον κύριο αυτό άξονα εδράζει το κουτί των ηλεκτρονικών έλεγχου στο πίσω μέρος του οποίου διέρχεται δευτερεύων κυλινδρικός άξονας μικρότερης διαμέτρου επίσης ελεύθερης περιστροφής και στηρίζει δυο μπάρες (differential bars) οι απολήξεις των οποίων ενώνονται μέσω σφαιρικών αρθρώσεων και συνδετικών ράβδων με τα άκρα των πλευρικών βραχιόνων. Στο σύστημα αυτό προσδίδεται ένας βαθμός ελευθερίας αφού όταν ο ένας βραχίονας ανέρχεται ο άλλος κατέρχεται.

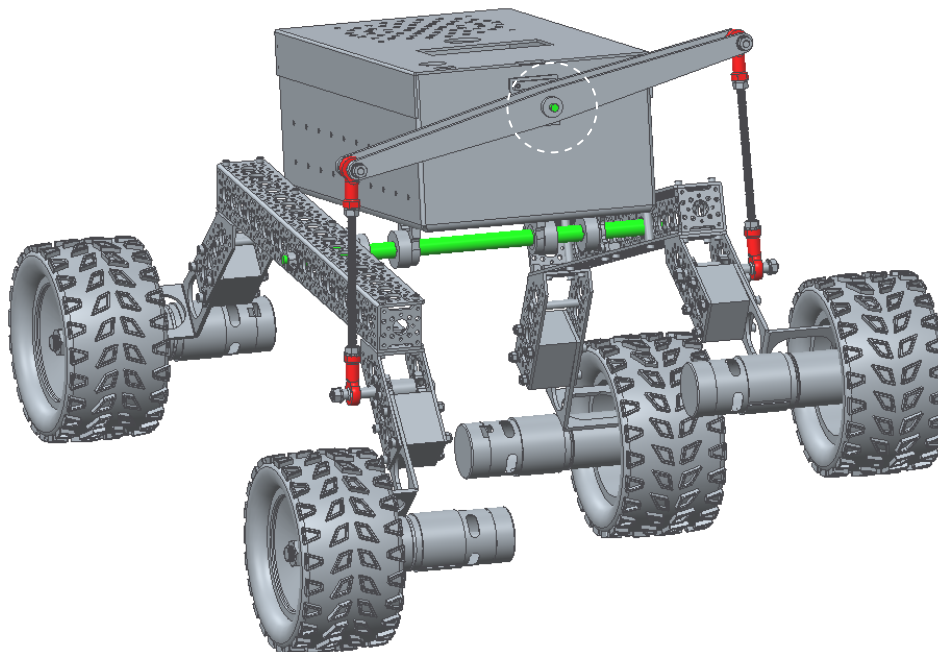
Έτσι το σασί του οχήματος προσαρμόζεται στη μορφολογία του εδάφους και οι τέσσερις τροχοί του παραμένουν συνεχώς σε επαφή με αυτό κατά την κίνηση του. Το παθητικό σύστημα ανάρτησης δίνει επίσης την ικανότητα στο όχημα να σκαρφλώνει εμπόδια με ύψος αρκετά μεγαλύτερο της διαμέτρου των τροχών του, παρέχει στο όχημα ευστάθεια και προσφέρει ομαλή κατανομή του φορτίου του ώστε κατά την διέλευση εμποδίων κανένας του τροχός να μην βυθίζεται σε μαλακό έδαφος ή να χάνει την πρόσφυση του. Η συμπεριφορά του δικτύωματος της ανάρτησης παρέχει σταθερότητα στην κάμερα η οποία βρίσκεται εγκατεστημένη στο κυρίως σώμα του λόγω των μειωμένων ταλαντώσεων του κυρίως αυτού σώματος σε σχέση με το σασί του οχήματος.

Κατά την αυτόνομη πλοήγηση το ρομποτικό όχημα αντιλαμβάνεται το περιβάλλον του και παίρνει ενδείξεις από αυτό χρησιμοποιώντας την web camera. Τα δεδομένα από την κάμερα του οχήματος χρησιμοποιούνται για να εξαχθούν οι εντολές κίνησης του. Για να έχει το ρομποτικό όχημα την δυνατότητα αυτόνομης πλοήγησης σε καλλιέργεια σειρών (row crop) απαιτεί ελεγκτή που να μπορεί να διαχειριστεί την αβεβαιότητα στο ανοιχτό περιβάλλον του αγρού με τις απρόβλεπτες κλιματικές συνθήκες και να δρα γρήγορα και αποτελεσματικά με τρόπο ανάλογο με αυτόν που θα δρούσε ένας χειριστής. Η ικανότητα της ασαφούς λογικής να μιμηθεί τον τρόπο δράσης ενός ανθρώπου χειριστή αποτελεί τον κύριο λόγο για τον οποίο σχεδιάστηκε και υλοποιήθηκε ασαφής ελεγκτής κίνησης για την αυτόνομη πλοήγηση του συγκεκριμένου ρομποτικού οχήματος.

Στο διάγραμμα της εικόνας 1.29 απεικονίζεται η διασύνδεση των ηλεκτρονικών υποσυστημάτων και η αρχιτεκτονική έλεγχου του ρομποτικού πρωτότυπου. Το όλο σύστημα τροφοδοτείται από μια μπαταρία LiPo. Η αναπτυξιακή πλακέτα Arduino Mega 2560 έχει τον ρόλο μητρικής κάρτας και προσφέρει την διασύνδεση όλων των επιμέρους υποσυστημάτων και διατάξεων. Σε κάθε κινητήριο τροχό αντιστοιχεί και ένας μικροελεγκτής ο οποίος είναι επιφορτισμένος με την ανάγνωση-αποκωδικοποίηση των σημάτων του αντίστοιχου κωδικοποιητή θέσης του σερβομηχανισμού και την υλοποίηση του ελέγχου κλειστού βρόχου με PID αντιστάθμιση. Ο μικροϋπολογιστής Odroid-XU3 εκτελεί το πρόγραμμα επεξεργασίας εικόνας και παράγει τα απαραίτητα δεδομένα εισόδου για τον ασαφή ελεγκτή κίνησης κατά την αυτόνομη πλοήγηση της ρομποτικής πλατφόρμας στην καλλιέργεια σειρών.



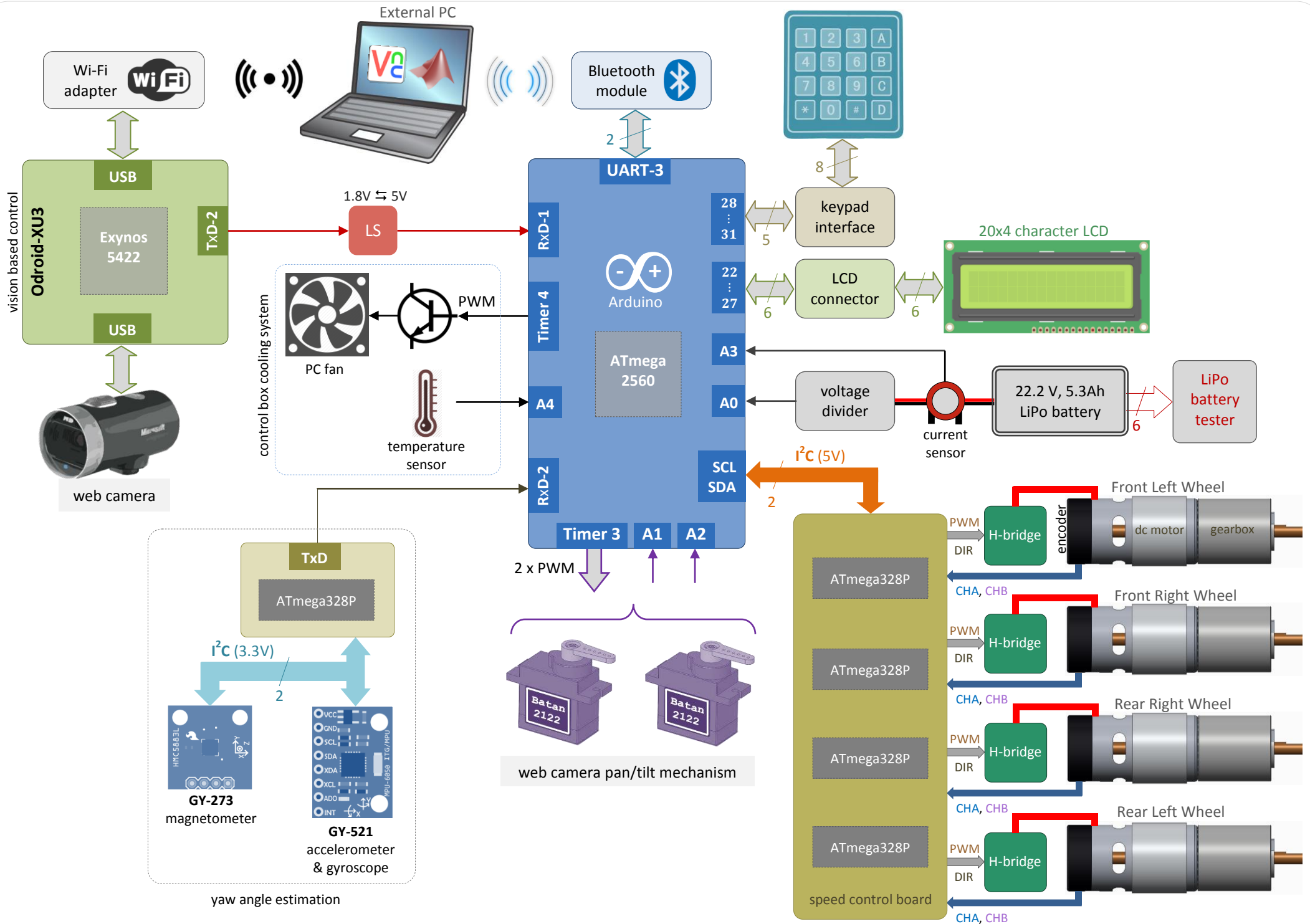
Εικόνα 1.27 Φωτογραφίες και cad απεικονίσεις του rover



Εικόνα 1.28 3D απεικόνιση της ρομποτικής πλατφόρμας. Με κόκκινο χρώμα απεικονίζονται οι σφαιρικές αρθρώσεις και πράσινο οι άξονες ελεύθερης περιστροφής του δικτυώματος.

Πληροφορίες σχετικά με την ρομποτική πλατφόρμα και τα τεχνικά της χαρακτηριστικά συνοψίζονται στον κάτωθι πίνακα:

Σύστημα ανάρτησης	παθητική, αρθρωτού τύπου
Σύστημα διεύθυνσης	διαφορική οδήγηση
Διαστάσεις (ΠxΜxΥ)	49x71x86 cm
Βάρος	10.5 Kg
Μέγιστη ταχύτητα	≈ 1.2 m/s
Κινητήρες	4x24V EMG49 DC κινητήρες μόνιμου μαγνήτη με ψήκτρες και μειωτήρα στροφών με λόγο μείωσης 49:1 2 x micro servo Batan 2122 στον μηχανισμό περιστροφής/κλίσης της κάμερας
Συσσωρευτής	22.2V, 5.3Ah 6 cell LiPo
Μικροϋπολογιστής	Odroid-XU3 της Hardkernel με Λ.Σ Ubuntu 14.04
Μικροελεγκτές	Arduino Mega 2560 , 5x ATmega328
Μονάδες εισόδου/εξόδου	4x4 ηλεκτρολόγιο μεμβράνης, 20x4 LCD οθόνη χαρακτήρων
Αισθητήρες	Microsoft LifeCam Cinema 720p HD, 4x κωδικοποιητές θέσης αυξητικού τύπου, μαγνητόμετρο HMC5883L, επιταχυνσιόμετρο & γυροσκόπιο MPU6050, θερμοκρασίας TMP36, ρεύματος Phidgets 1119, τάσης συσσωρευτή
Επικοινωνία	WiFi, Bluetooth module HC-06 (για απομακρυσμένη διαχείριση) I <sup>2</sup> C, ασύγχρονη σειριακή (μεταξύ των μC)
Προγραμματίστηκε σε	Wiring, C/C++
Προγραμματιστικά εργαλεία, βιβλιοθήκες	Arduino IDE, Atmel Studio 6 + Visual Micro plugin, Microsoft Visual Studio 2013, Eclipse IDE, Matlab/Simulink, OpenCV 2.4.9
Κόστος εξαρτημάτων	≈ 1500€



Εικόνα 1.29 Σχηματικό διάγραμμα της διασύνδεσης των ηλεκτρονικών υποσυστημάτων και της αρχιτεκτονικής ελέγχου του ρομπωτικού οχήματος.

## Κεφάλαιο

## 2

## 2. Δομικά εξαρτήματα οχήματος - Συναρμολόγηση

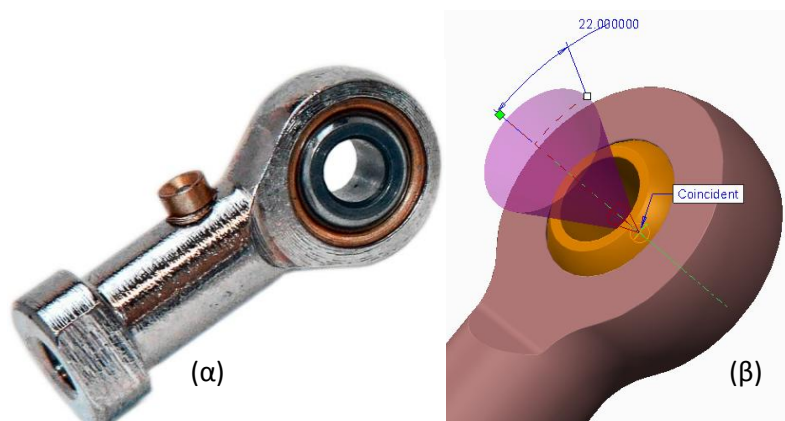
Στο κεφάλαιο αυτό παρουσιάζονται όλα τα δομικά μέρη από τα οποία απαρτίζεται το όχημα: τα εξαρτήματα Actobotics που χρησιμοποιούνται καθώς και εκείνα που κατασκευάστηκαν εκ' νέου. Δίνονται επίσης τα στάδια συναρμολόγησης της ρομποτικής πλατφόρμας και σύνθεσης της ως ένα ενιαίο σύνολο.

### 2.1 Δομικά εξαρτήματα του οχήματος

#### 2.1.1 Σφαιρικοί σύνδεσμοι

Η σύνδεση των βραχιόνων που φέρουν τους ηλεκτροκινητήρες (rocker arms) με τον μηχανισμό του διαφορικού πραγματοποιήθηκε μέσω κοχλιωτών ράβδων στα άκρα των οποίων βιδώθηκαν σφαιρικοί σύνδεσμοι. Κάθε τέτοιος σύνδεσμος βασίζεται στη γωνιακή μετατόπιση της σφαίρας και επιτρέπει τη δημιουργία σφαιρική άρθρωσης.

Κάθε διάταξη σφαιρικού συνδέσμου αποτελείται από τρία μέρη α) κυλινδρικό περίβλημα με σφαιρικό άκρο και διάτρηση με εσωτερικό σπείρωμα στο άλλο άκρο β) δακτύλιο και γ) διάτρητη σφαίρα. Ο δακτύλιος εσωκλείεται στο σφαιρικό άκρο του περιβλήματος και η σφαίρα είναι προσαρμοσμένη στον δακτύλιο το εσωτερικό του οποίου φέρει αυλάκι πάνω στο οποίο κυλιέται η σφαίρα. Κάθε σύνδεσμος διαθέτει ειδικό στόμιο λίπανσης. Η μέγιστη γωνία κάμψης ενός άξονα ο οποίος διέρχεται από την σφαίρα είναι  $22^\circ$ .



Εικόνα 2.1 α) Σφαιρικός σύνδεσμος PHS 6 β) αξονική μετατόπιση

### 2.1.2 Τροχοί



Εικόνα 2.2 Οι τροχοί του rover της εταιρείας Absima

Τα ελαστικά του ρόβερ είναι της εταιρείας Absima και έχουν διάμετρο 154 mm και πλάτος 78 mm. Η ζάντα έχει εσωτερική διάμετρο 89 mm και φέρει εξάγωνη υποδοχή σύνδεσης 17mm.

### 2.1.3 Εξαρτήματα Actobotics

Για την κατασκευή του ρόβερ χρησιμοποιήθηκαν τα κάτωθι εξαρτήματα *Actobotics*. Στο δικτυακό τόπο [www.servocity.com](http://www.servocity.com) δίνονται φωτογραφίες, τα μηχανολογικά σχέδια των εξαρτημάτων καθώς και τα 3D γραφικά τους μοντέλα (step files).

α/α	Ποσότητα	Κωδικός	Είδος	Παρατηρήσεις
1	2	585458	15"L Aluminum Channel	Δοκοί αλουμινίου σε σχήμα Π
2	1	585440	1.5"L Aluminum Channel	
3	4	585482	Channel Bracket B (45°)	
4	1	585468	Flat Single Bracket	
5	4	545360	90° Quad Hub Mount C	
6	4	545576	0.77" Set Screw Hub (6mm Bore)	
7	1	634244	Solid Shaft (1/2"D, 12"L)	
8	1	634164	Solid Shaft (1/4"D, 4"L)	Συμπαγείς άξονες ανοξείδωτου ατσάλιού
9	2	535158	1/2" Bore Pillow Block	Έδρανα
10	2	535118	1/2" Bore Flat Bearing Mount	
11	1	535130	1/4" Ball Bearing Quad Pillow Block	
12	6	535198	1/4" Bore Flanged Ball Bearing	Σφαιρικό ρουλεμάν με φλάντζα
13	6	625102	1/4" Shaft Clamp Collar	Σφιγκτήρες
14	4	6435K14	1/2" Bore Steel Clamp Collar	
15	4	585470	90° Dual Side Mount	
16	8	585472	Dual Screw Plate	
17	6	633136	1.32"L 6-32 Aluminum Threaded Standoff	
18	2	633122	1/2"L 6-32 Aluminum Threaded Standoff	Αποστάτες αλουμινίου με εσωτερικό σπειρώμα

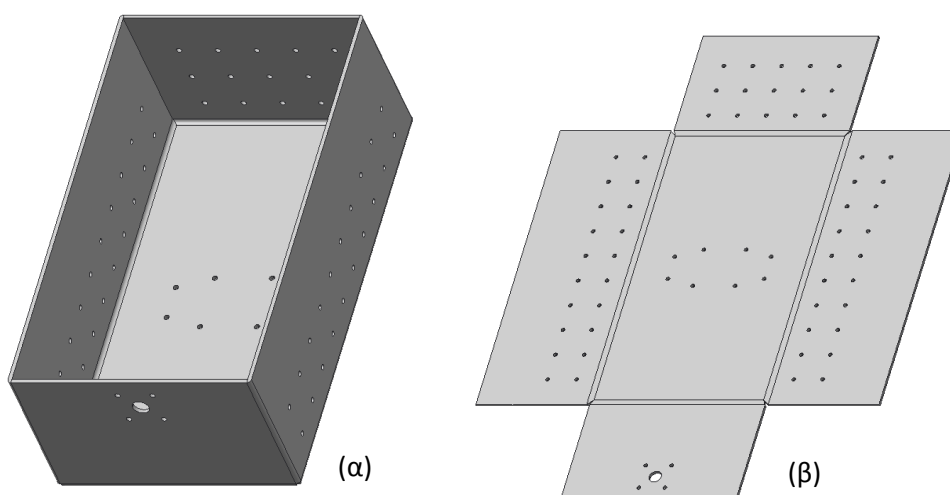
Το Actobotics είναι ένα σύστημα δόμησης το οποίο δίνει στον χρήστη την δυνατότητα της ανάπτυξης και σύνθεσης πρότυπων και σύνθετων ηλεκτρομηχανολογικών κατασκευών όπως π.χ. ρομποτικών εφαρμογών και άλλων αυτοματοποιημένων ή αλληλεπιδραστικών συστημάτων. Βασίζεται σε μια πληθώρα εξειδικευμένων εξαρτημάτων όπως π.χ. δοκών αλουμινίου, ράβδων, σωλήνων, εδράνων, γραναζιών, σφικτήρων, συνδέσμων κ.α. τα οποία ακολουθούν συγκεκριμένο μοτίβο διαστάσεων με μεγάλη ακρίβεια. Χάρη στα τυποποιημένα σχήματα των οπών που διαθέτουν τα δομικά εξαρτήματα, όλα σχεδόν μπορούν να συνδεθούν μεταξύ τους με απεριόριστες δυνατότητες τοποθέτησης. Το τελικό αποτέλεσμα είναι μηχανολογικές κατασκευές με μικρά όρια ανοχής και ελάχιστες τριβές μεταξύ των κινούμενων μερών.

#### 2.1.4 Κατασκευή εξαρτημάτων

Τα 3D γραφικά μοντέλα όλων των εξαρτημάτων δημιουργήθηκαν με την βοήθεια της εφαρμογής PTC Creo Parametric 2.0 και στη συνέχεια για την κατασκευή τους κυρίως σε μηχανές ψηφιακής καθοδήγησης (CNC) έγινε χρήση λογισμικού CAM (PTC-Creo, CGTech-Vericut) για την παραγωγή του κώδικα G και την προσομοίωση της κίνησης του κοπτικού εργαλείου της εργαλειομηχανής ώστε να ελεγχθεί η μορφή του μοντέλου και η ακρίβεια της κατεργασίας πριν από την πραγματική εκτέλεση των κατεργασιών. Όλα τα μηχανολογικά σχέδια των εξαρτημάτων διατίθενται στο παράρτημα Β.

##### 2.1.4.1 Κουτί ηλεκτρονικών ελέγχου

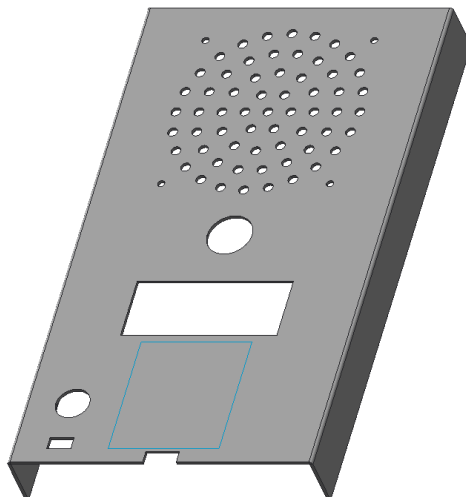
Το κουτί το οποίο φιλοξενεί το μεγαλύτερο μέρος του ηλεκτρονικού εξοπλισμού και τον συσσωρευτή του ρόβερ κατασκευάστηκε από φύλλο αλουμινίου πάχους 3 mm και έχει διαστάσεις 188 x 332 x 113 mm (ΠxΜxΥ). Το βάρος του είναι 1614 γρ. και παρέχει χωρητικότητα περίπου 6485 cm<sup>3</sup>. Το ανάπτυγμα της λαμαρίνας αφού μορφοποιήθηκε και κάμφθηκε ανάλογα συγκολλήθηκε στις ακμές του. Στο πίσω μέρος του κουτιού λόγω των εγκάρσιων μηχανικών τάσεων που ασκούνται από τον άξονα του διαφορικού έχει τοποθετηθεί διπλό φύλλο λαμαρίνας ίδιου πάχους και αναλόγων διαστάσεων για περαιτέρω ενίσχυση του.



Εικόνα 2.3 α) Κουτί ηλεκτρονικών ελέγχου β) ανάπτυγμα κάμψεων

#### 2.1.4.2 Καπάκι κουτιού των ηλεκτρονικών ελέγχου

Το καπάκι του κουτιού των ηλεκτρονικών ελέγχου δημιουργήθηκε από φύλλο αλουμινίου πάχους 2 mm. Εκτός από την προστασία που παρέχει στον εξοπλισμό έχει διαμορφωθεί κατάλληλα ώστε να αποτελεί και βάση στήριξης για τον ανεμιστήρα, την LCD οθόνη χαρακτήρων, του κουμπιού τροφοδοσίας του οχήματος, του πληκτρολόγιου τύπου μεμβράνης και της υποδοχής για USB θύρα. Στο μέσο του υπάρχει κυκλικό άνοιγμα μέσα από το οποίο διέρχεται ο σωλήνας υποδοχής της κάμερας.



Εικόνα 2.4 3D γραφικό μοντέλο του καπακιού

#### 2.1.4.3 Μπάρα διαφορικού

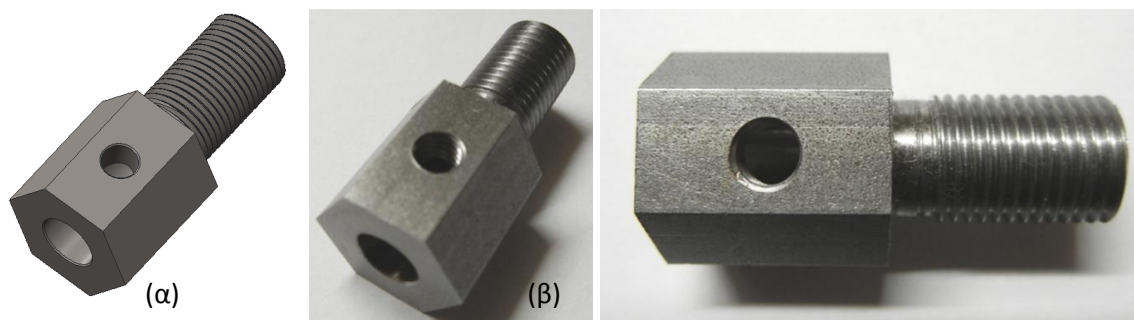
Σε CNC επίσης κατασκευάστηκαν οι δυο βραχίονες του διαφορικού από μια μπάρα αλουμινίου πάχους 5 mm και πλάτους 20 mm. Το βάρος κάθε μιας είναι 87 γρ.



Εικόνα 2.5 Ο βραχίονας διαφορικού (differential arm).

#### 2.1.4.4 Προσαρμογέας σύνδεσης του κινητήρα με τον τροχό του οχήματος

Κατασκευάστηκαν σε τόρνο με πρώτη ύλη μια εξαγωνη μπάρα μολυβδούχου χάλυβα 17mm και καθένας έχει βάρος 44 γρ.

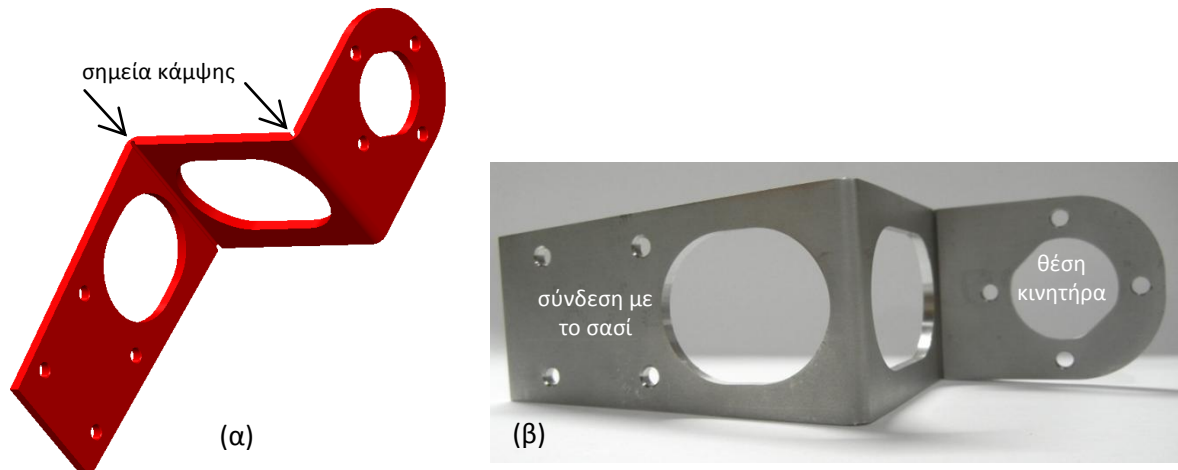


Εικόνα 2.6 Προσαρμογέας σύνδεσης κινητήρα με τον τροχό α) CAD μοντέλο β) φωτογραφίες του εξαρτήματος.



#### 2.1.4.5 Βάση στήριξης κινητήρα και σύνδεσης του με το σασί

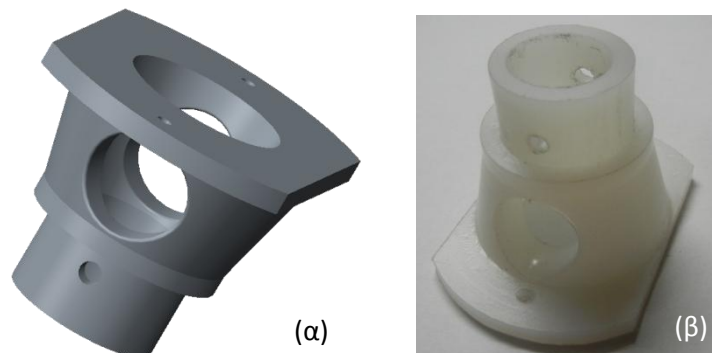
Οι τέσσερις βάσεις κατασκευάστηκαν από ανοξείδωτο ατσάλι πάχους 3 mm σε CNC. Κάθε κομμάτι μετά την κατεργασία του κάμφθηκε κατά 90° στα σημεία που απεικονίζονται στην παρακάτω 3D μοντέλο. Το βάρος κάθε βάσης είναι ίσο με 127 γρ.



Εικόνα 2.7 Βάση στήριξης του κινητήρα και σύνδεσης του με το σασί α) CAD μοντέλο β) φωτογραφία

#### 2.1.4.6 Βάση στήριξης του καπακιού των ηλεκτρονικών ελέγχου

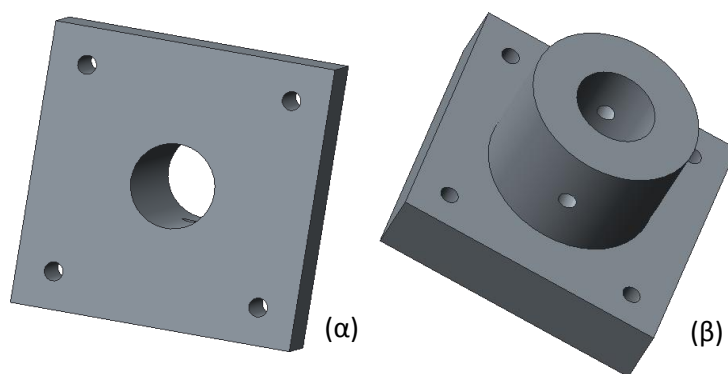
Για να είναι προσιτά τα ηλεκτρονικά ελέγχου τα οποία εμπεριέχονται στο κουτί (control box) του ρόβερ και για να μην δυσχεραίνεται η πρόσβαση σε αυτά όταν χρειάζεται να γίνουν παρεμβάσεις κατασκευάστηκε από πλαστικό (ertalon) βάση στήριξης του καπακιού ώστε αυτό να σταθεροποιείται σε σταθερή απόσταση.



Εικόνα 2.8 Βάση στήριξης του καπακιού α) 3D CAD μοντέλο β) φωτογραφία

### 2.1.4.7 Βάση σταθεροποίησης του στύλου για την κάμερα

Η κάμερα του ρόβερ τοποθετείται στο ένα άκρο μιας σωλήνας αλουμινίου εξωτερικής διαμέτρου 18 mm και μήκους 55 cm. Το άλλο άκρο της σωλήνας σταθεροποιείται στον πάτο του κουτιού των ηλεκτρονικών ελέγχου μέσω μια βάσης στήριξης η οποία κατασκευάστηκε από πλαστικό για αυτόν το σκοπό.

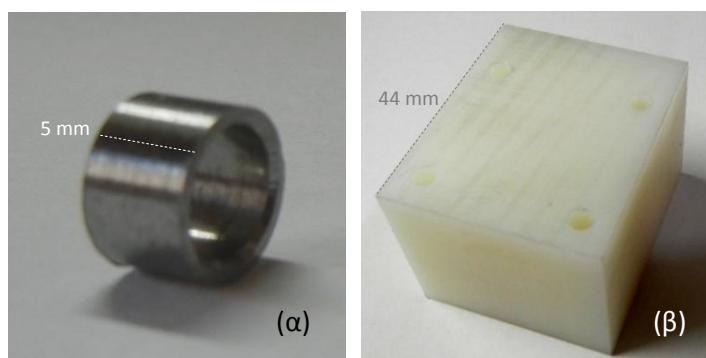


Εικόνα 2.9 3D CAD μοντέλο της βάσης για την σταθεροποίηση του στύλου της κάμερας α) κάτω β) πάνω όψη

### 2.1.4.8 Αποστάτες

Κατασκευάστηκαν τέσσερις ορθογώνιοι παραλληλεπίπεδοι αποστάτες από πλαστικό (ertalon) με βάρος 59 γρ. ο καθένας ώστε να συνδεθεί η βάση στήριξης του κινητήρα με το channel bracket B.

Για την αποφυγή συγκρούσεων μεταξύ των σφαιρικών συνδέσμων (PHS6 Female rod end bearing) με την differential bar και τα rocker arms κατασκευάστηκαν στον τόρνο έξι αποστάτες από εξάγωνη μπάρα μολυβδούχου χάλυβα. Το μήκος κάθε αποστάτη είναι 5 mm, η εξωτερική διάμετρος τους 8 mm και η εσωτερική τους 6 mm.

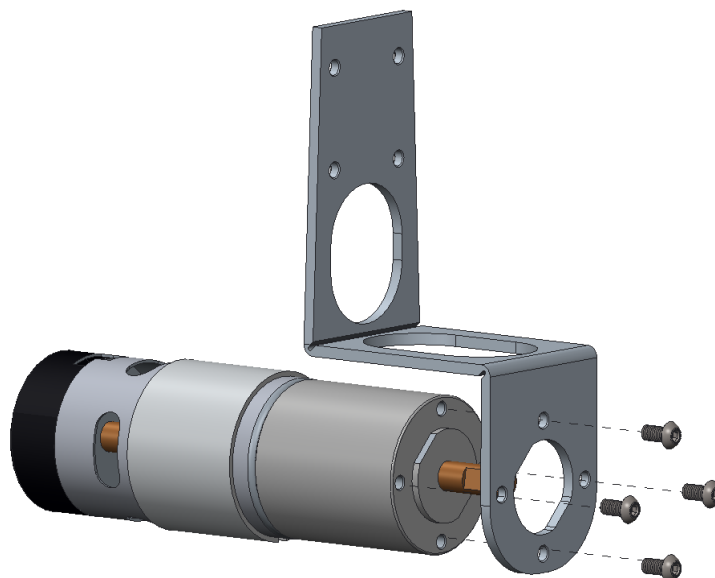


Εικόνα 2.10 Αποστάτες α) μεταλλικός β) πλαστικός

## 2.2 Συναρμολόγηση

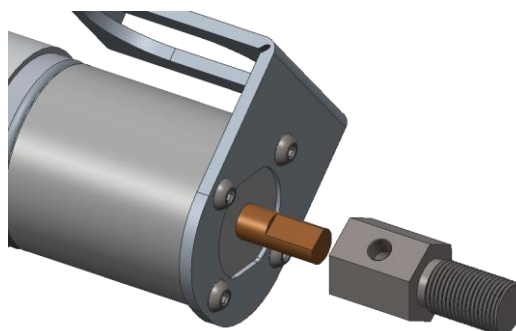
### 2.2.1.1 Σύνδεση κινητήρων με τη βάση στήριξης τους

Ο κινητήρας τοποθετείται στην κατάλληλη εγκοπή της βάσης με προσανατολισμό τέτοιο ώστε τα καλώδια της τροφοδοσίας και του κωδικοποιητή θέσης να έχουν φορά προς άνω. Οι κοχλίες για την σταθεροποίηση του κινητήρα είναι μήκους 8 mm με στρογγυλό κεφάλι και εξαγωνική εγκοπή.



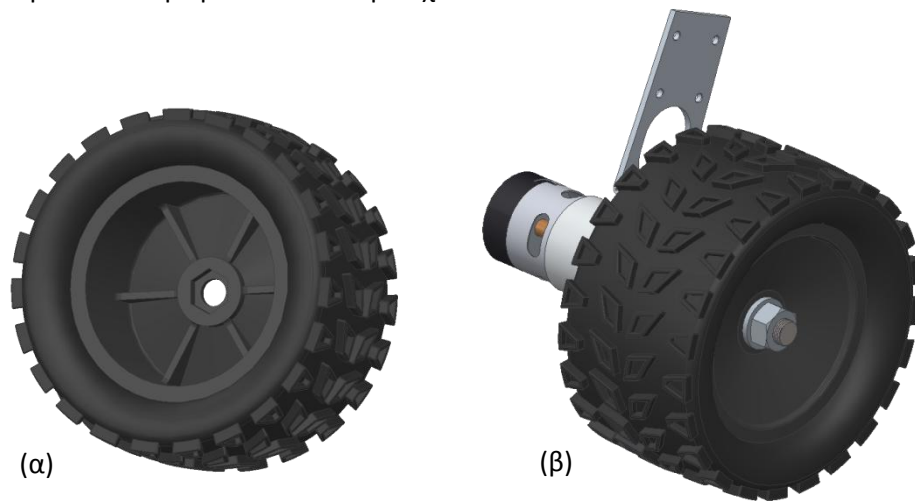
Εικόνα 2.11 Σύσφιξη κινητήρα στην βάση του

Στον άξονα του κινητήρα, το άκρο του οποίου είναι σχήματος D, συσφίγγεται κατάλληλα ο προσαρμογέας σύνδεσης του με τον τροχό μέσω ακέφαλου κοχλία (σκουλικόβιδα). Ο κοχλίας αυτός πρέπει να βιδωθεί κατά τέτοιο τρόπο ώστε η άκρη του να πιέζει το επίπεδο τμήμα επιφάνειας του άξονα.



Εικόνα 2.12 Εισαγωγή προσαρμογέα σύνδεσης κινητήρα-τροχού

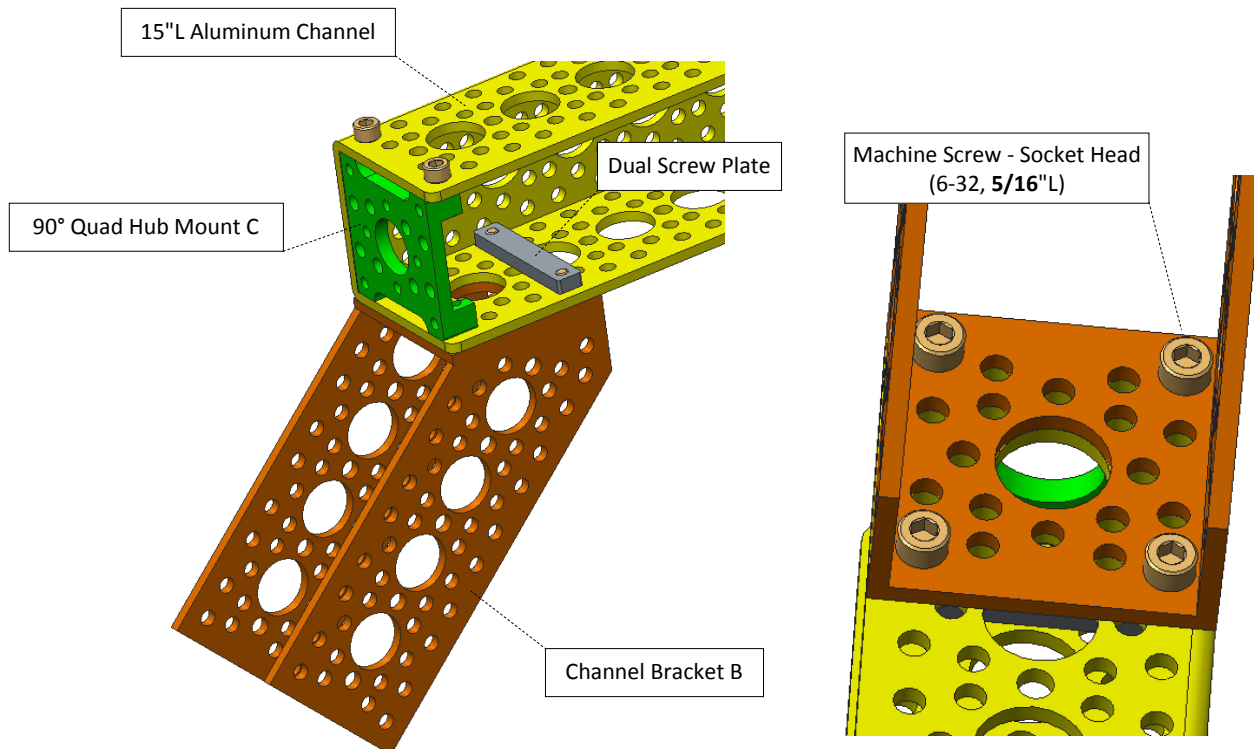
Η εξάγωνη υποδοχή της ζάντας του τροχού εφαρμόζεται στον προσαρμογέα και ο τροχός σταθεροποιείται με ροδέλα και περικόχλιο M10.



Εικόνα 2.13 α) Πίσω όψη του τροχού β) Προσάρτηση τροχού στον κινητήρα

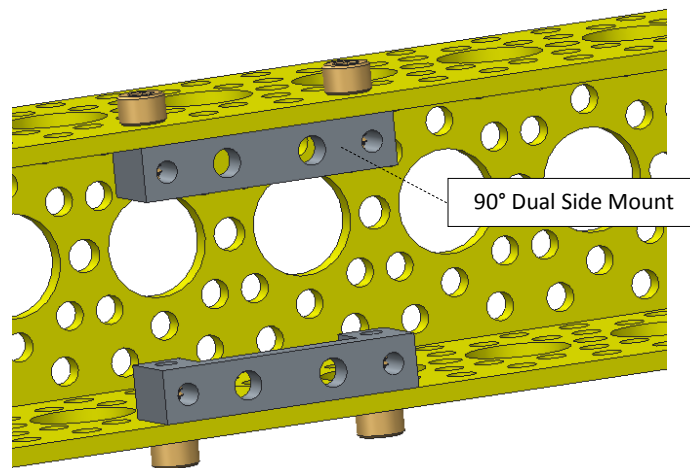
### 2.2.1.2 Συναρμολόγηση των πλευρικών βραχιόνων (rocker arms)

Στο κάθε άκρο της δοκού με σχήμα Π και μήκους 15" βιδώνεται μέσω κοχλιών κυλινδρικής κεφαλής με εξαγωνική εγκοπή (άλλεν) η υπό γωνία 45° αλουμινένια δοκός με την βοήθεια κατάλληλου τετράγωνου προσαρμογέα σύνδεσης (εδώ χρησιμοποιούμενου και ως τερματικού αποστάτη) και ειδικού μακρόστενου πλακιδίου σύνδεσης.



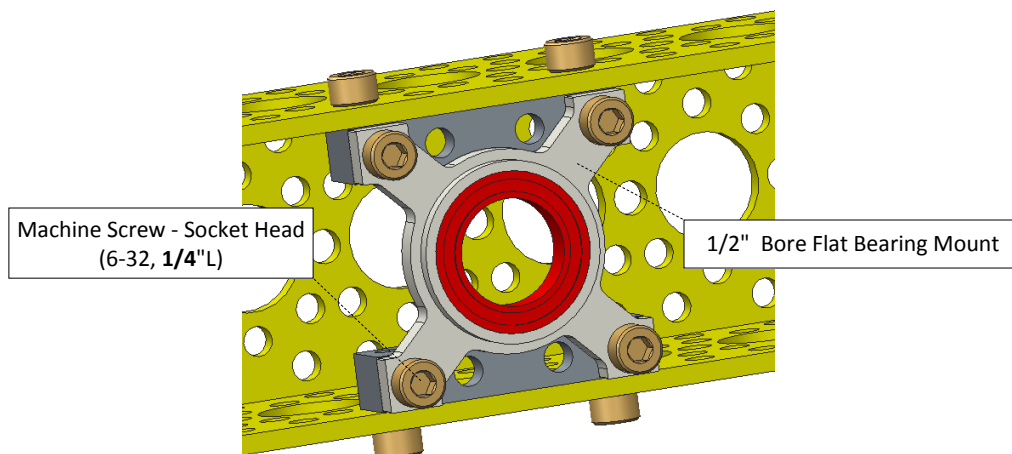
Εικόνα 2.14 Συναρμολόγηση βραχίονα (rocker arm). Προσάρτηση στο άκρο της οριζόντιας δοκού της υπό γωνίας 45° δοκού

Στο μέσο της οριζόντιας δοκού βιδώνονται στηρίγματα διπλής όψης.



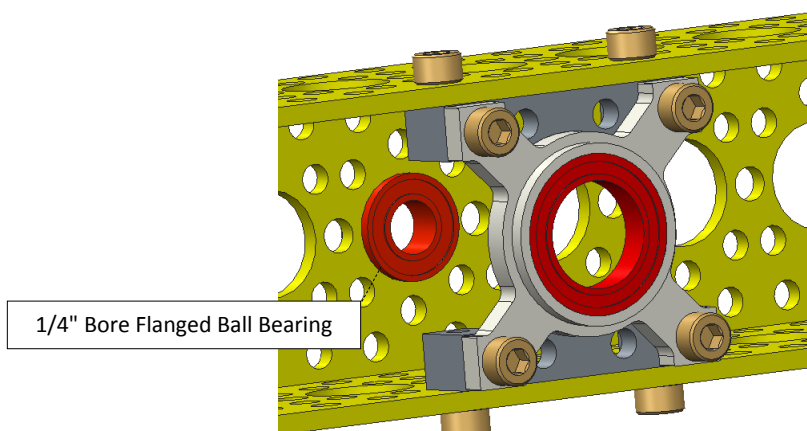
Εικόνα 2.15 Τοποθέτηση βάσεων για την στήριξη εδράνου

Στις βάσεις αυτές βιδώνεται αξονικό έδρανο ολίσθησης.



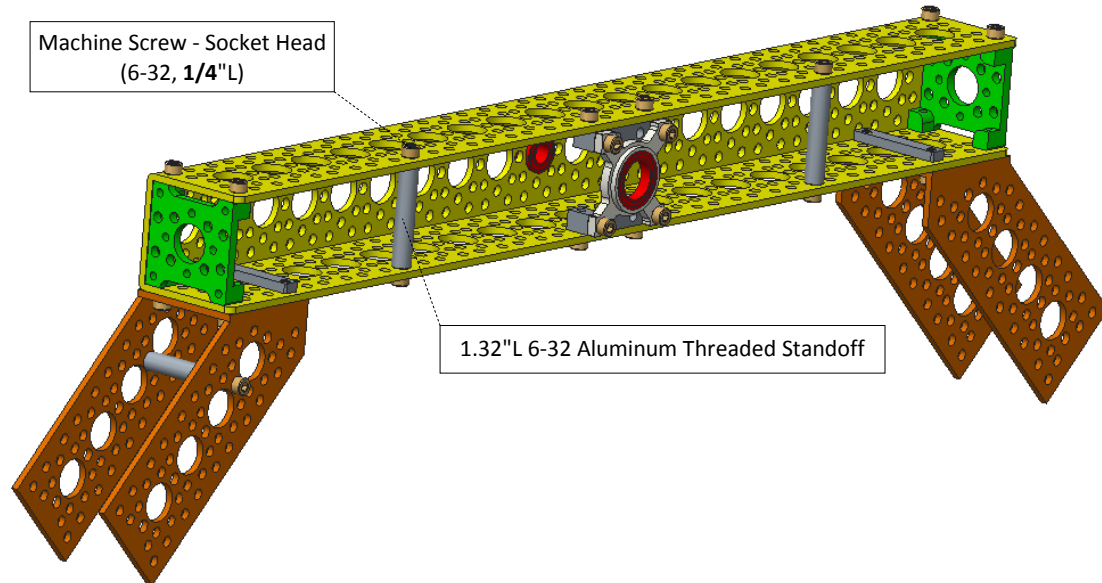
Εικόνα 2.16 Σύσφιξη εδράνου ολίσθησης στην οριζόντια δοκό

Απέναντι από το έδρανο και στην οπή της δοκού με διάμετρο 0.5" τοποθετείται σφαιρικό ρουλεμάν με φλάντζα η οποία εφάπτεται στην εξωτερική επιφάνεια της δοκού.



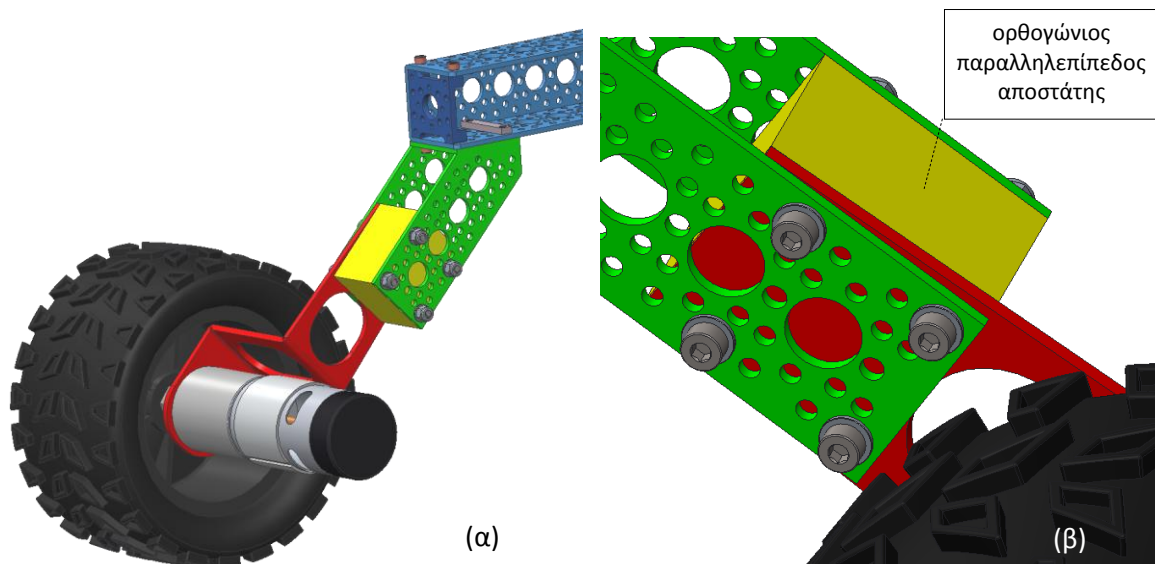
Εικόνα 2.17 Τοποθέτηση ρουλεμάν με φλάντζα

Βιδώνονται τρεις κυλινδρικοί αποστάτες αλουμινίου στις κατάλληλες θέσεις. Δυο στην οριζόντια δοκό και ένας στην προσαρτημένη σε αυτήν μπροστινή υπό γωνία 45° δοκό.



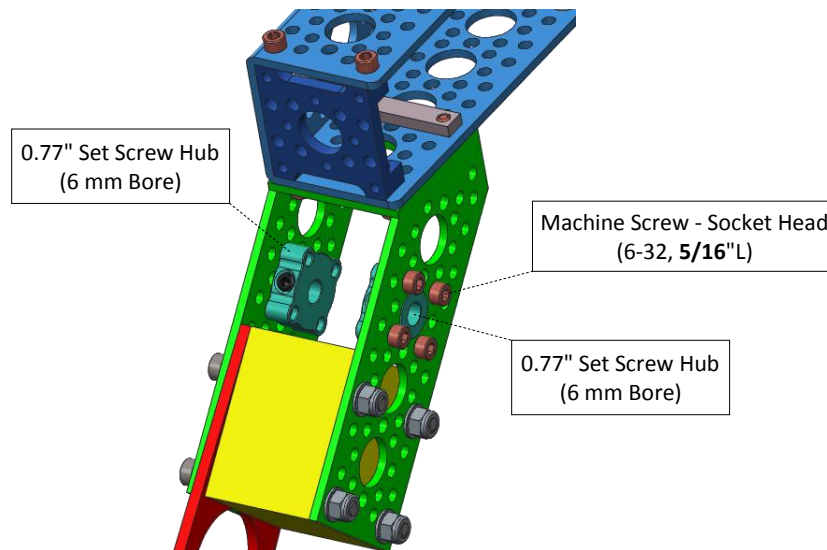
Εικόνα 2.18 Εισαγωγή αποστατών σε συγκεκριμένες θέσεις του πλευρικού βραχίονα

Το κάθε άκρο του βραχίονα ενώνεται με την βάση στήριξης κινητήρα-τροχού μέσω του ορθογώνιου παραλληλεπίπεδου αποστάτη από πλαστικό και 4 κοχλίες τύπου άλλεν M4x45mm με περικόχλια ασφαλείας M4. Για την προστασία της επιφάνεια της αλουμινένιας δοκού και την ομαλότερη κατανομή της δύναμης συμπίεσως χρησιμοποιήθηκαν ανάλογες ροδέλες τόσο στην κεφαλή αλλά και στο άκρο των κοχλιών.



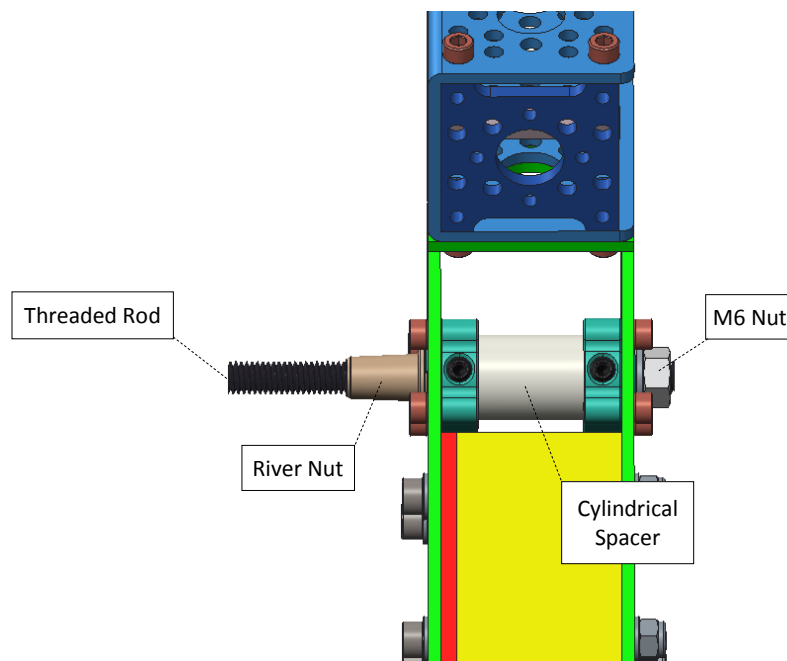
Εικόνα 2.19 α) Συναρμολογημένο άκρο του βραχίονα β) μεγεθυσμένο του τμήμα

Στο πίσω μέρος της οριζόντιας δοκού και στην προσαρτημένη σε αυτήν υπό γωνία 45° δοκό βιδώνονται οι βάσεις στήριξης κοχλιωτού άξονα διαμέτρου 6mm. Οι βάσεις τοποθετούνται εσωτερικά της δοκού και απέναντι η μια από την άλλη με την κυλινδρική εγκοπή τους διαμέτρου 0.5" να εισέρχεται σε αντίστοιχη οπή της δοκού.



Εικόνα 2.20 Τοποθέτηση βάσεων στήριξης κοχλιωτού άξονα

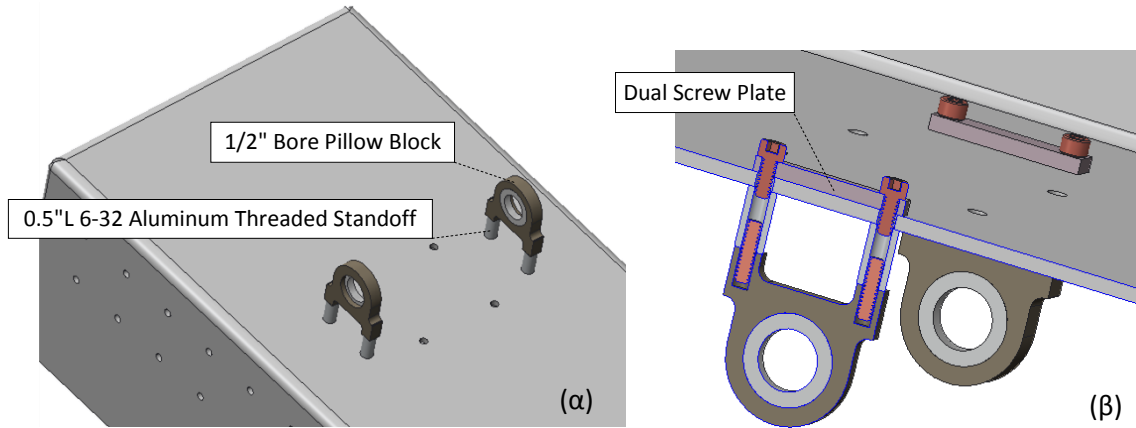
Ανάμεσα στις δυο αυτές βάσεις παρεμβάλλεται διάτρητος κυλινδρικός αποστάτης από πλαστικό. Ο κοχλιωτός άξονας συσφίγγεται με περικόχλιο M6 από την μια πλευρά και με μεταλλικό αποστάτη με εσωτερικό σπείρωμα (river nut) από την άλλη αλλά και εγκάρσια από τις σκουλικόβιδες που διαθέτουν οι βάσεις.



Εικόνα 2.21 Σύσφιξη κοχλιωτού άξονα

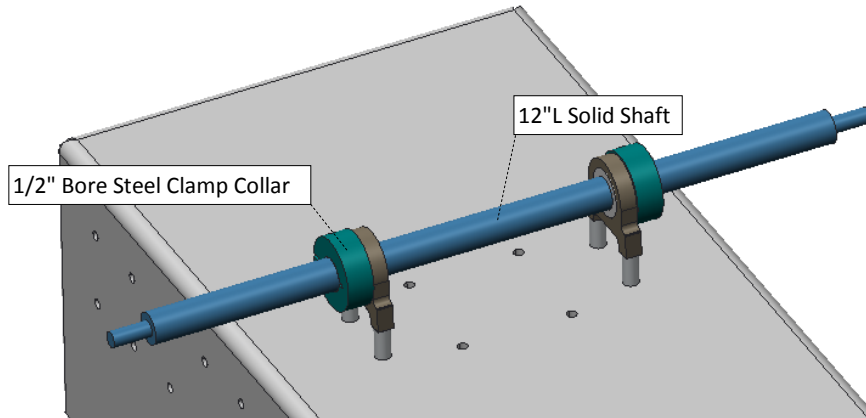
### 2.2.1.3 Συναρμολόγηση εξαρτημάτων στο κουτί των ηλεκτρονικών ελέγχου

Εξωτερικά και στην κάτω έδρα του κουτιού βιδώνονται μέσω ειδικού πλακιδίου σύνδεσης δυο έδρανα ολίσθησης τύπου Ω. Ανάμεσα στα έδρανα και στο κουτί παρεμβάλλονται αλουμιμένιοι κυλινδρικοί αποστάτες μήκους 0.5".



Εικόνα 2.22 α) Κάτω όψη κουτιού ελέγχου β) Τομή στο επίπεδο του εδράνου

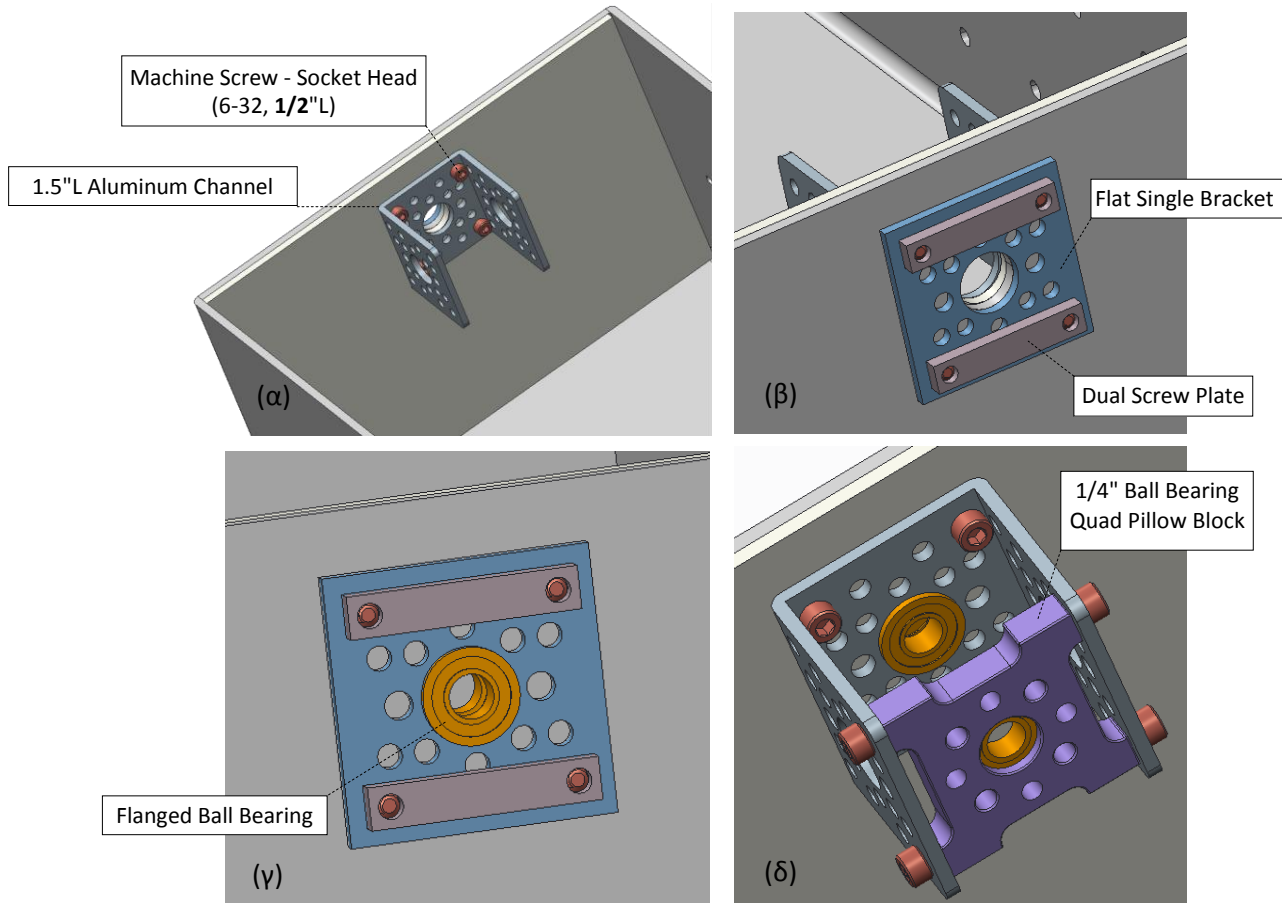
Μέσα από τα κουζινέτα βάσης διέρχεται συμπαγής άξονας διαμέτρου 0.5" από ατσάλι του οποίου τα άκρα έχουν κατεργαστεί και λεπτυνθεί σε μήκος 0.6" ώστε η διάμετρος τους να είναι 0.25". Για την συγκράτηση του άξονα χρησιμοποιούνται δακτύλιοι σύσφιξης.



Εικόνα 2.23 Τοποθέτηση και σύσφιξη του κύριου άξονα

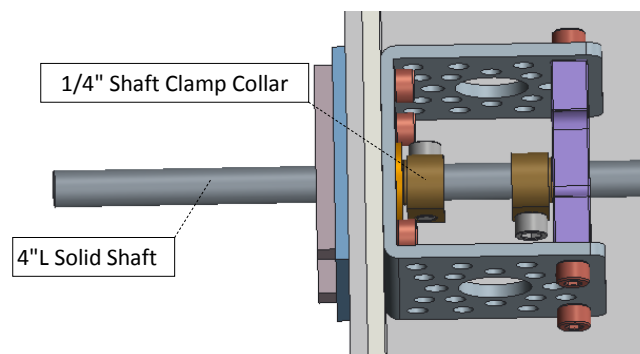


Στο πίσω μέρος του κουτιού των ηλεκτρονικών ελέγχου και στην διάτρηση με διάμετρο 0.5" στηρίζεται και βιδώνεται η δοκός αλουμινίου μήκους 1.5" μέσω των ειδικών πλακιδίων σύνδεσης και ενός επίπεδου υποστηρίγματος. Στην διαμπερή τρύπα διαμέτρου 0.5" τοποθετούνται δυο σφαιρικά ρουλεμάν με φλάντζα ένα σε κάθε πλευρά. Στην άκρη της δοκού βιδώνεται τετράγωνη βάση με έδρανο κύλισης εσωτερικής διαμέτρου 1/4".



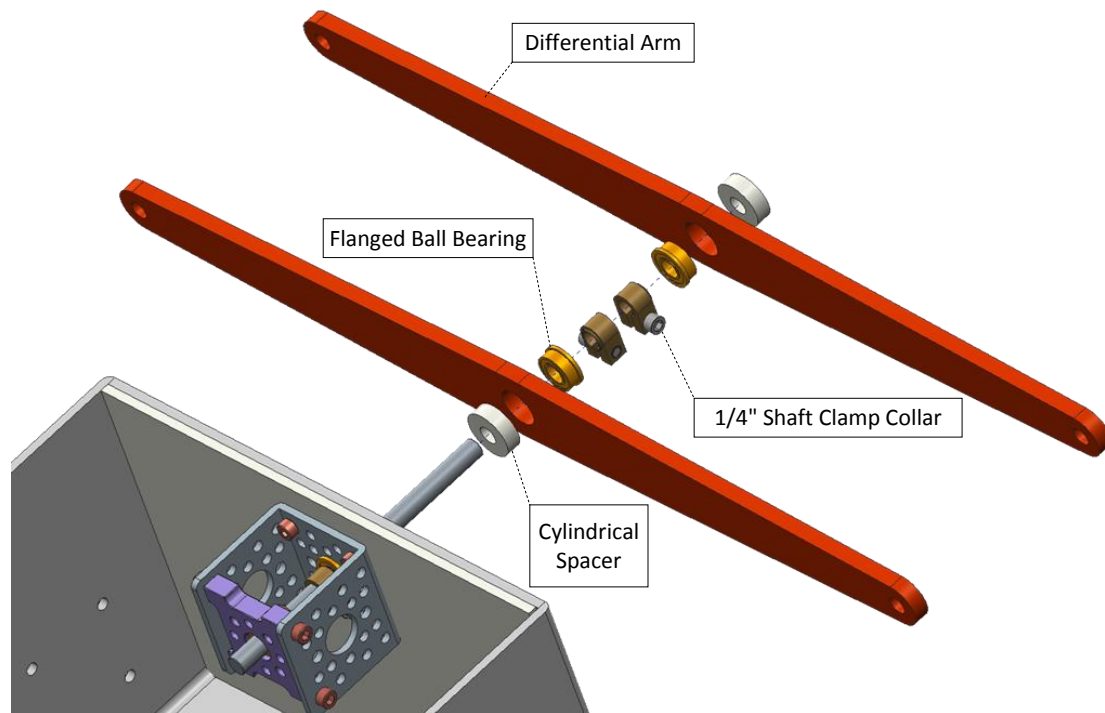
**Εικόνα 2.24** Στάδια συναρμολόγησης της βάσης στήριξης του άξονα του διαφορικού

Ο άξονας του διαφορικού συγκρατείται με σφικτήρες οι οποίοι τοποθετούνται με το κυκλικό εξόγκωμα που φέρουν να εφάπτεται στο ρουλεμάν, ώστε να ελαχιστοποιούνται οι τριβές μεταξύ τους.



**Εικόνα 2.25** Τοποθέτηση του άξονα διαμέτρου 1/4"

Στον άξονα αυτόν προσαρτούνται οι βραχίονες του διαφορικού και τα παρελκόμενα τους (κυλινδρικοί αποστάτες, σφικτήρες, ρουλεμάν με φλάντζα) με την σειρά και τον προσανατολισμό που εμφανίζεται στην κάτωθι 3Δ γραφική απεικόνιση.



Εικόνα 2.26 Συναρμολόγηση βραχιόνων διαφορικού

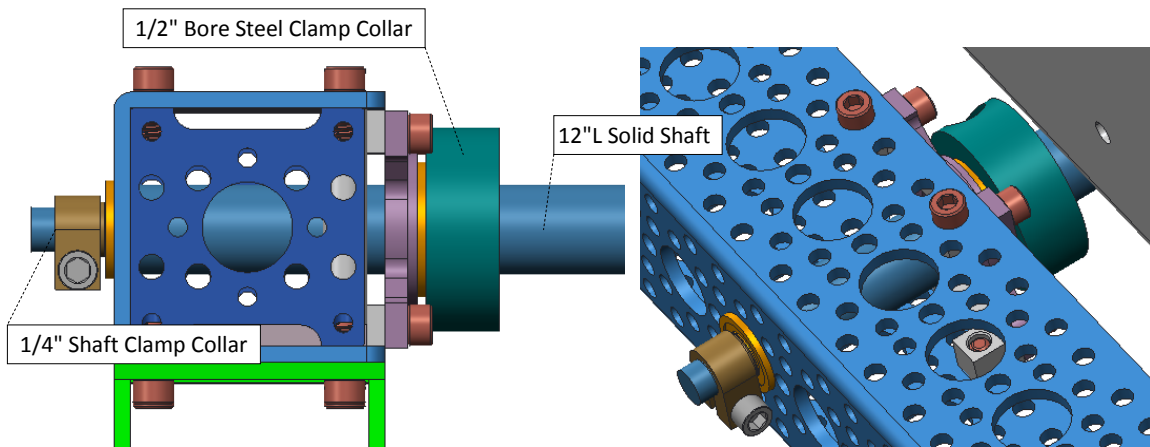
Για την συναρμολόγηση των μπαρών που ενώνουν του βραχίονες του διαφορικού με τους πλευρικούς βραχίονες (rocker arms) χρησιμοποιούμε δυο κοχλιωτές ράβδους στα άκρα των οποίων βιδώνονται οι σφαιρικοί σύνδεσμοι. Πρώτα βιδώνονται στο κάθε άκρο της κοχλιωτής ράβδου τα περικόχλια M6 κατόπιν εισάγονται οι ροδέλες και τελευταίοι βιδώνονται οι δυο σφαιρικοί σύνδεσμοί με τρόπο τέτοιο ώστε οι άξονες που θα διέρχονται από αυτούς να είναι κάθετοι.



Εικόνα 2.27 Μπάρα διαφορικού

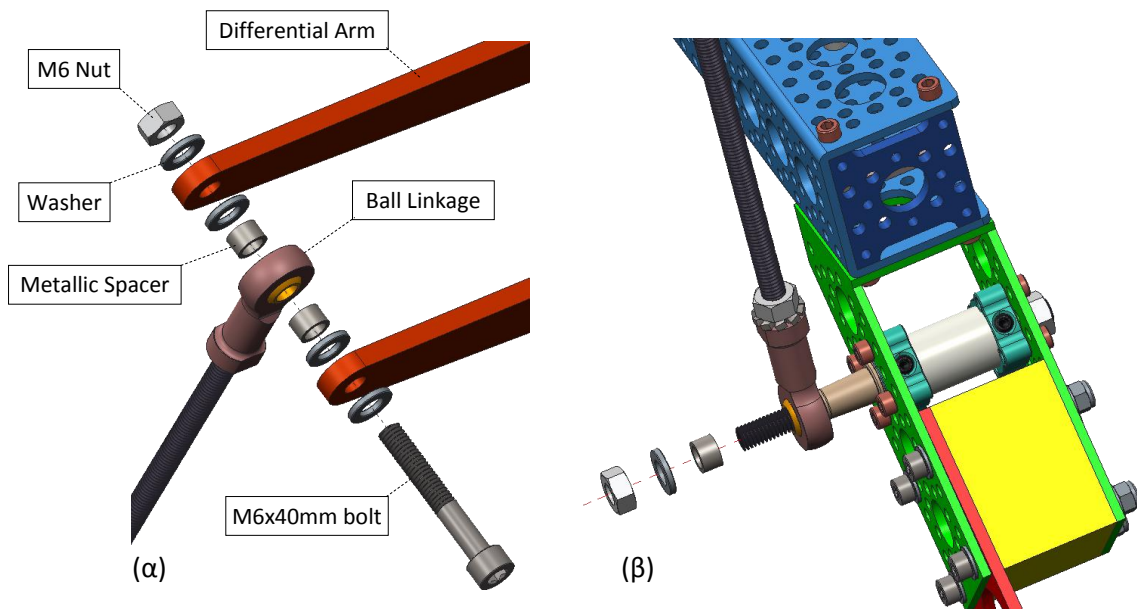
### 2.2.1.4 Συναρμολογή του σασί με το κουτί ελέγχου

Για την σύνδεση του κουτιού ελέγχου με τους πλευρικούς βραχίονες εισάγεται πρώτα ένας σφικτήρας εσωτερικής διαμέτρου 1/2" σε κάθε άκρο του κυρίως άξονα και έπειτα κάθε άκρο του άξονα εισέρχεται στον αντίστοιχο βραχίονα κάθε πλευράς μέσω του εγκατεστημένου (στο μέσο κάθε οριζόντιας δοκού) εδράνου ολίσθησης. Τα κατεργασμένα άκρα διαμέτρου 1/4" του κυρίως άξονα διέρχονται από τα ρουλεμάν με την φλάντζα και συσφίγγονται με τα αντίστοιχα κολάρα.



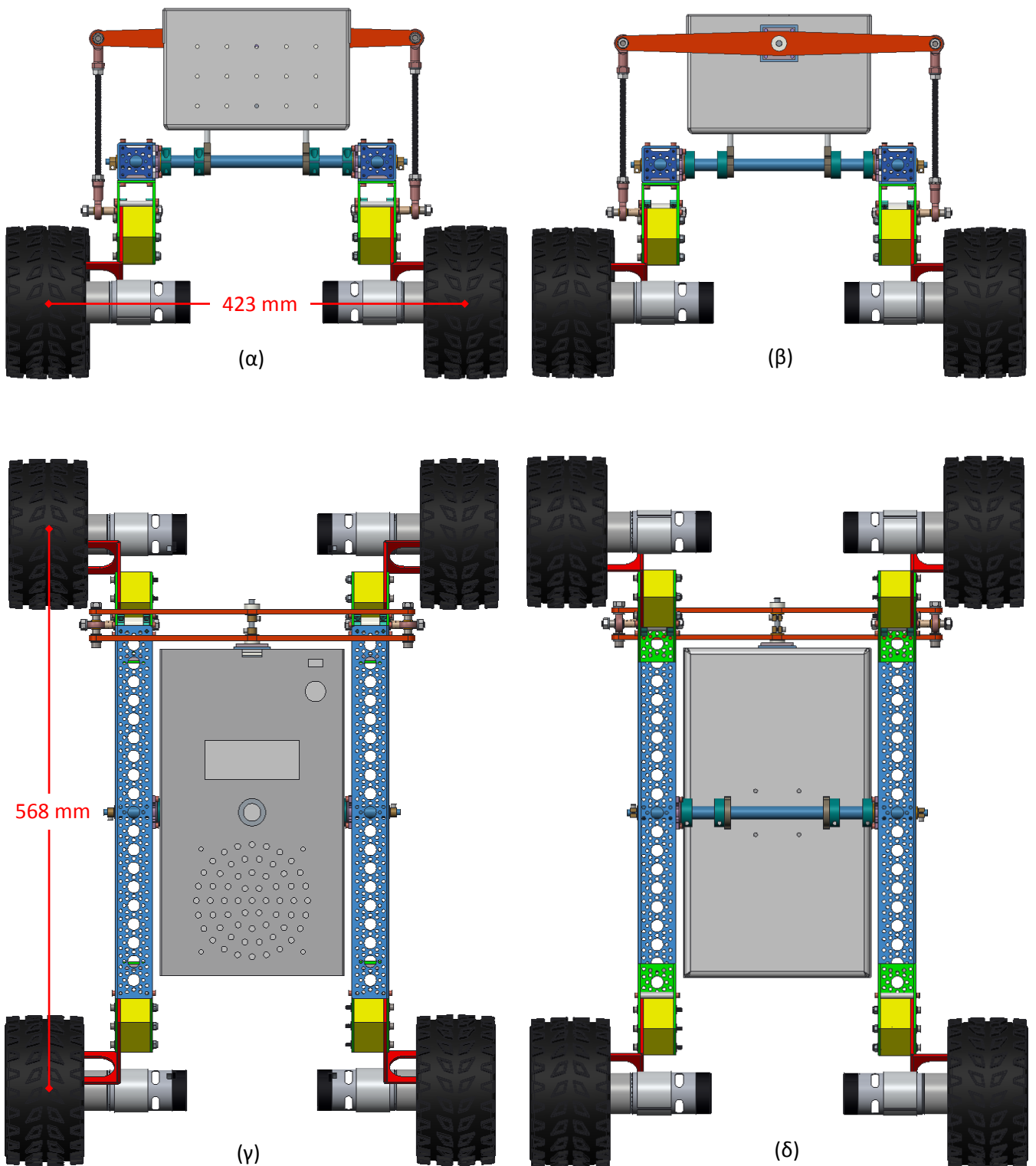
Εικόνα 2.28 Εισαγωγή κυρίως άξονα στους πλευρικούς βραχίονες

Ο τρόπος σύνδεσης της διάταξης του διαφορικού με το σασί του οχήματος μέσω των άκρων κάθε κοχλιωτής μπάρας που φέρει σφαιρικούς συνδέσμους εικονίζεται στις κάτωθι 3D γραφικές απεικονίσεις.

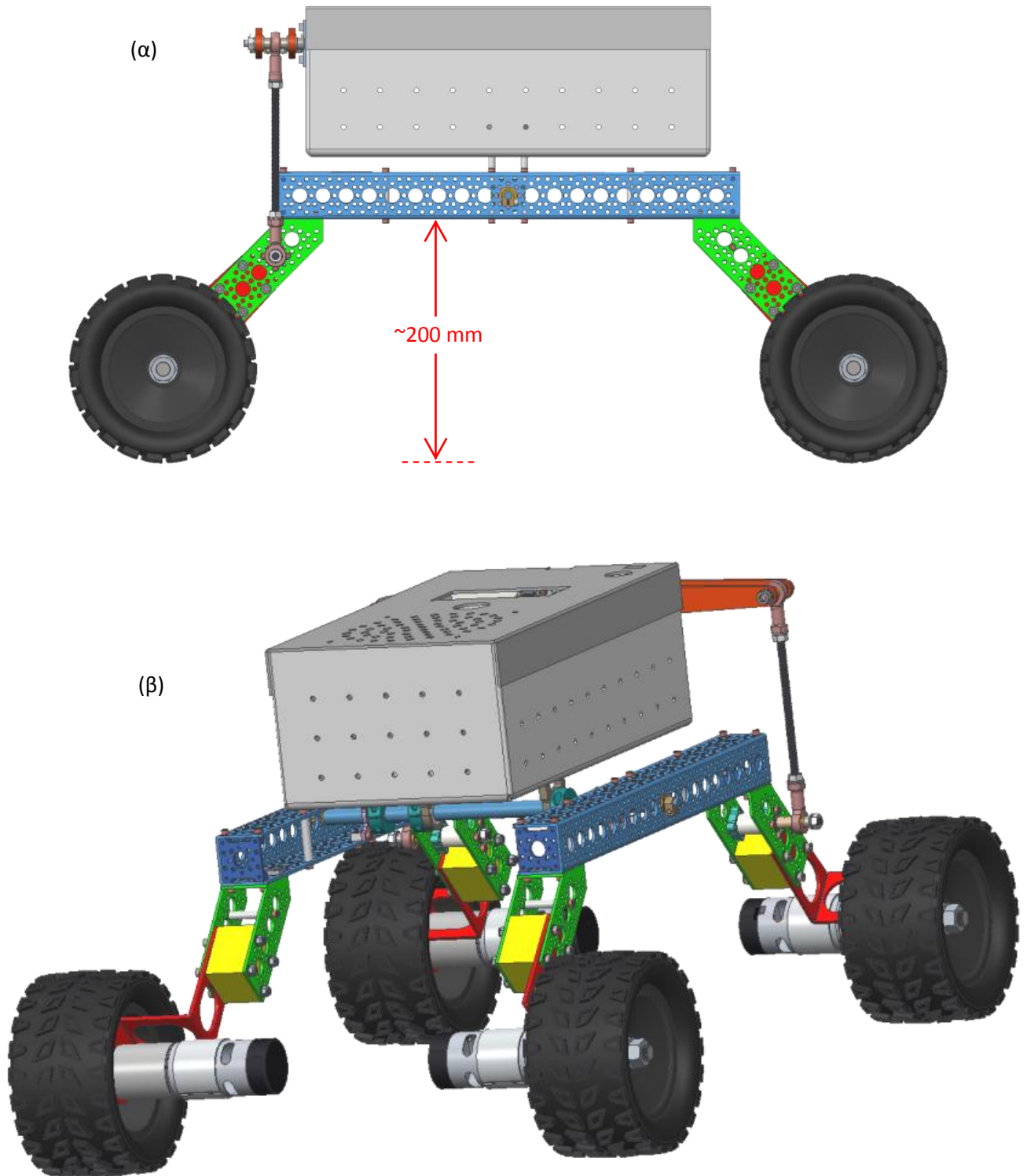


Εικόνα 2.29 Σύνδεση διάταξης διαφορικού με το σασί α) άνω και β) κάτω άκρο της μπάρας

### 2.3 3Δ γραφικές απεικονίσεις του ρόβερ



Εικόνα 2.30 α) εμπρός β) οπίσθια γ) άνω και δ) κάτω όψη του ρόβερ



Εικόνα 2.31 α) Δεξιά πλευρά και β) πλάγια όψη του ρόβερ

## Κεφάλαιο

## 3

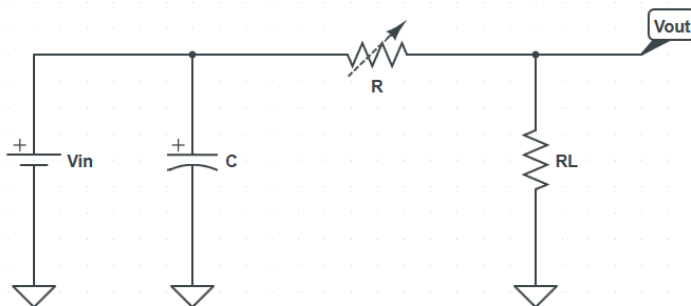
### 3. Κατανομή ηλεκτρικής ενέργειας - Τροφοδοσία

Στο παρόν κεφάλαιο γίνεται μια εκτενής αναφορά στους ρυθμιστές τάσης που χρησιμοποιούνται, περιγράφεται η συνδεσμολογία και τα απαραίτητα τυπωμένα κυκλώματα που κατασκευάστηκαν για την διανομή της ηλεκτρικής ενέργειας από τον συσσωρευτή στους κινητήρες και στον λοιπό ηλεκτρονικό εξοπλισμό του οχήματος. Αναφέρονται τα βασικά χαρακτηριστικά της μπαταρίας ιόντων πολυμερών (LiPo) που φέρει το όχημα και γενικότερα η αρχή λειτουργίας των μπαταριών αυτών καθώς και ο τρόπος φόρτισης τους. Επισημαίνονται οι απαραίτητες προφυλάξεις που απαιτεί η χρήση των μπαταριών LiPo και παρουσιάζεται η χρησιμοποιούμενη συσκευή για την συνεχή επίβλεψη του επιπέδου τάσης κάθε στοιχείου της μπαταρίας. Τέλος αποδεικνύεται η επάρκεια τόσο των επιλεγμένων κινητήρων ως προς την απαιτούμενη ροπή όσο και της χρησιμοποιούμενης μπαταρίας LiPo για τις απαιτήσεις της εφαρμογής.

#### 3.1 Ηλεκτρονικά ισχύος - Ρυθμιστές τάσεις

##### 3.1.1 Γραμμικοί ρυθμιστές

Οι γραμμικοί ρυθμιστές τάσης (voltage regulator) είναι κατάλληλοι για την τροφοδοσία συσκευών χαμηλής κατανάλωσης ενέργειας. Είναι εύκολοι στην χρήση, φθηνοί και ως εκ τούτου είναι αρκετά δημοφιλείς. Ωστόσο λόγω του τρόπου λειτουργία τους σε κάποιες εφαρμογές η χρήση τους είναι εξαιρετικά ενεργοβόρα και αυτό γιατί η διαφορά τάσης μεταξύ εισόδου και εξόδου αποβάλλεται ως θερμότητα αναγκαστικά μέσω ογκωδών ψηκτρών. Χονδρικά ένας γραμμικός ρυθμιστής μπορεί να παρομοιαστεί με μια μεταβλητή αντίσταση η τιμή της οποίας αναπροσαρμόζεται ανάλογα με το φορτίο ώστε να διατηρείται σταθερή η τάση εξόδου.



Εικόνα 3.1 Ισοδύναμο κύκλωμα γραμμικού ρυθμιστή τάσης

Πρέπει να διερευνάται για το αν σε μια εφαρμογή σπαταλείται περισσότερη ενέργεια για τον υποβιβασμό της τάσης από ότι τελικά διανέμεται στην συσκευή αποδέκτη. Η καταναλισκόμενη ενέργεια σε ένα γραμμικό ρυθμιστή είναι:

$$P = (V_{in} - V_{out})I_L$$

Συνεπώς όσο μεγαλύτερη είναι η διαφορά της τάσης εισόδου με την τάση εξόδου για δεδομένο καταναλισκόμενο ρεύμα τόσο μεγαλύτερες είναι οι απώλειες ισχύος που εμφανίζονται υπό μορφή θερμότητας.

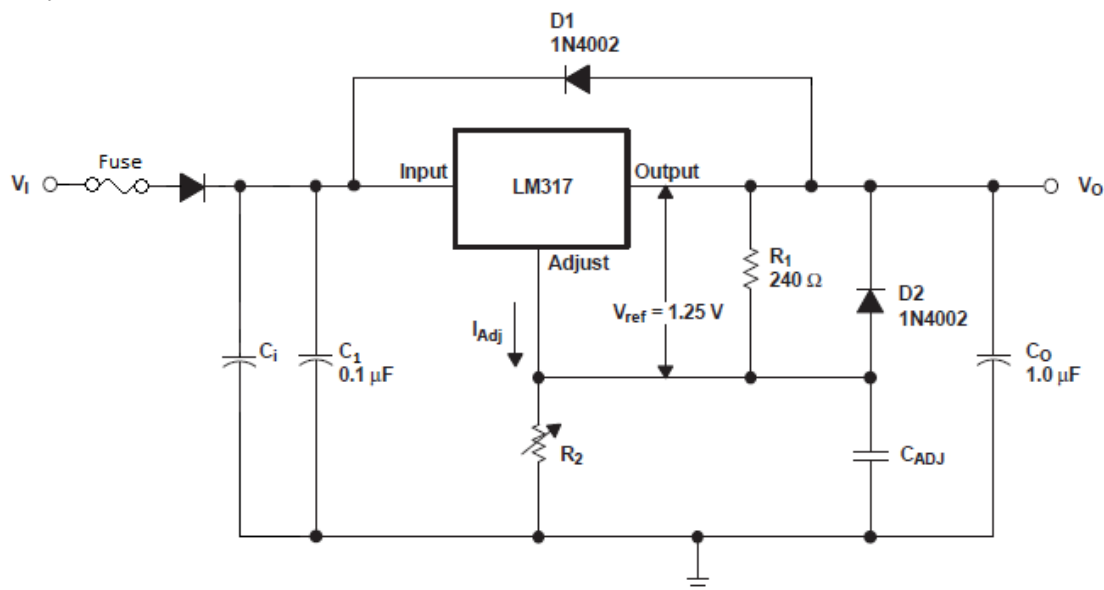
Για την ρύθμιση της τάσης στα επιθυμητά επίπεδα χρησιμοποιήθηκε ο γραμμικός σταθεροποιητής τάσης LM317 ο οποίος διαθέτει τρεις ακροδέκτες και είναι κατάλληλος για εφαρμογές χαμηλής ισχύος. Το συγκεκριμένο ολοκληρωμένο δίνει στην έξοδο του τάση απαλλαγμένη από διακυμάνσεις με εύρος 1.25V έως 37V και παροχή ρεύματος μεγαλύτερη του 1.5A κάτω από συγκεκριμένες συνθήκες εργασίας και με κατάλληλη ψήκτρα αλουμινίου. Διαθέτει θερμική προστασία και εσωτερική ασφάλεια για τον περιορισμό του ρεύματος σε περίπτωση βραχυκυκλώματος. Επιτυγχάνει κατά 80dB μείωση της κυμάτωσης (ripple) στην έξοδο το οποίο σημαίνει ότι η διακύμανση της τάσης εισόδου είναι 10000 μικρότερη στην έξοδο αποφεύγοντας τη χρήση ογκωδών LC φίλτρων. Συμφώνα με το datasheet για κάθε ένα volt που τροποποιείται η τάση εισόδου η ποσοστιαία μεταβολή της τάσης εξόδου είναι 0.02% (line regulation). Όταν το ρεύμα στο φορτίο μεταβληθεί από 10mA σε 1.5A η μεταβολή της τάσης εξόδου είναι 0.3% (load regulation).

Για την τροφοδοσία του Arduino Mega 2560 με 7V, των δυο μικρό-σερβοκινητήρων για την προσανατολισμό της κάμερας με 5V και όλων των υπολοίπων ηλεκτρονικών διατάξεων επίσης με 5V κατασκευάστηκαν τρία παρεμφερή τυπωμένα κυκλώματα με βάση το κυκλωματικό διάγραμμα της εικόνας 3.2. Τάση εισόδου για κάθε ένα είναι τα 12V και διαφοροποιούνται μεταξύ τους στην τιμή της ασφάλειας, της χωρητικότητας του πυκνωτή εισόδου  $C_i$  και την τιμή της αντίστασης  $R_2$ . Τα σχέδια των κυκλωμάτων αυτών παρατίθενται στο παράρτημα Γ της παρούσας εργασίας.

Η τάση εξόδου στο κάτωθι κύκλωμα εξαρτάται από το δίκτυο ανάδρασης που εφαρμόζεται στον ακροδέκτη Adj. Η τιμή της τάσης εξόδου δίνεται από τη σχέση:

$$V_{out} = \left(1 + \frac{R_2}{R_1}\right)V_{ref} + I_{Adj}R_2$$

Με την  $R_2$  να είναι η αντίσταση προς τη γη και η τάση στα άκρα της  $R_1$  να είναι σταθερή  $V_{ref} = 1.25V$ . Το ρεύμα  $I_{Adj}$  τυπικά είναι 50μΑ και θεωρείται αμελητέο.



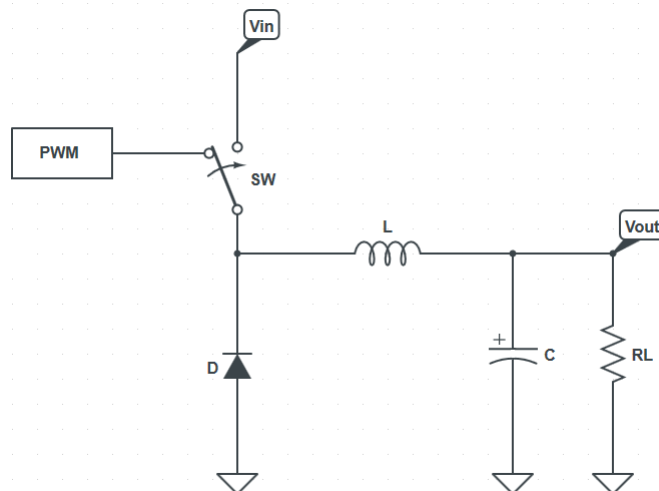
Εικόνα 3.2 Κύκλωμα ρύθμισης της τάσης με βάση το γραμμικό σταθεροποιητή LM317

Ο πυκνωτής εισόδου  $C_i$  διασφαλίζει ότι η τάση εισόδου θα διατηρείται το ελάχιστο 2 με 3V υψηλότερη από την τάση εξόδου όταν η τελευταία μεγιστοποιείται σε συνδυασμό με αύξηση του ρεύματος στο φορτίο. Ο πυκνωτής εξόδου  $C_o$  βελτιώνει τη μεταβατική απόκριση και ο  $C_{Adj}$  αποτρέπει την ενίσχυση της κυμάτωσης καθώς η τάση εξόδου ρυθμίζεται υψηλότερα. Οι δίοδοι προστασίας D1, D2 εξασφαλίζουν αγωγίμο δρόμο κατά την εκφόρτιση των πυκνωτών  $C_o$  και  $C_{Adj}$  αποτρέποντας έτσι τη διέλευση του ρεύματος μέσα από διαδρομές χαμηλής αντίστασης του ολοκληρωμένου.

Η ασφάλεια (resettable fuse PTC) στην είσοδο του κυκλώματος σε κανονικές συνθήκες έχει πολύ χαμηλή αντίσταση. Όταν όμως ξεπεράσει το όριο αυξάνει η θερμοκρασία της και αποκτά πολύ υψηλή αντίσταση. Έτσι μέχρι την πιθανή άρση της ανωμαλίας που προκάλεσε την αρχική υπερ-ένταση το κύκλωμα διαρρέεται από χαμηλό ρεύμα και η διάταξη προστατεύεται από μόνιμες βλάβες.

### 3.1.2 Διακοπτικοί ρυθμιστές τάσης

Οι διακοπτικοί ρυθμιστές τάσης (switching regulators) είναι πιο σύνθετοι αλλά υψηλότερης απόδοσης και έχουν χαμηλή απαίτηση ισχύος. Τα κυκλώματα αυτά χρησιμοποιούν έναν τουλάχιστον ημιαγωγικό διακόπτη ισχύος (κυρίως MOSFET) ο οποίος βρίσκεται σε είτε σε αγωγίμη κατάσταση είτε σε κατάσταση αποκοπής. Λαμβάνονται έτσι μικρές ποσότητες ενέργειας διακοπτικά από την πηγή εισόδου και μεταφέρονται στην έξοδο. Αυτό επιτυγχάνεται με τη βοήθεια ενός ελεγκτή ο οποίος ρυθμίζει τον ρυθμό μεταφοράς της ενέργειας.



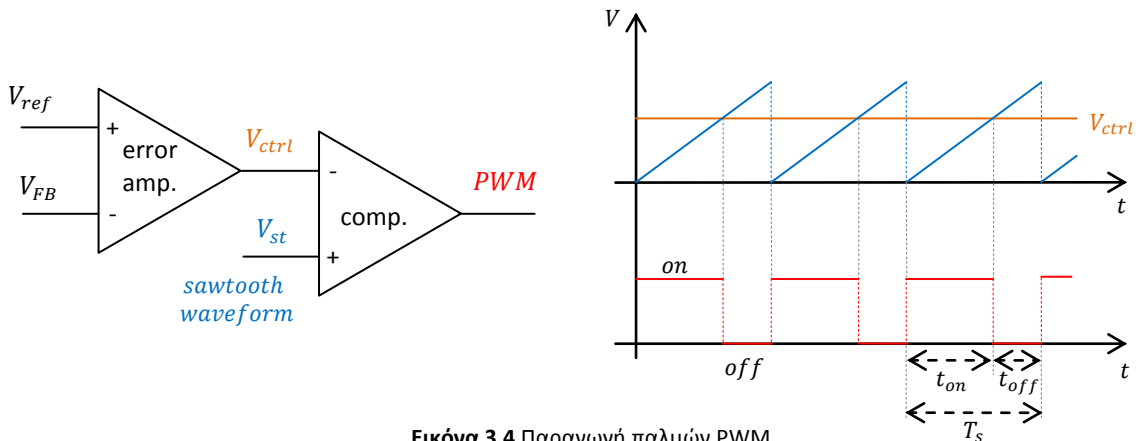
Εικόνα 3.3 Ισοδύναμο κύκλωμα διακοπτικού ρυθμιστή τάσης

Για τον έλεγχο αυτών των μετατροπέων χρησιμοποιείται η τεχνική της διαμόρφωσης εύρους παλμού (PWM). Διατηρείται σταθερή η συχνότητα και μεταβάλλεται ο χρόνος κατά τον οποίο είναι αγωγίμο το ημιαγωγικό στοιχείο ώστε να καθορίζεται η μέση τιμή της τάσης εξόδου. Τέτοιες μορφές παλμοί παράγονται με τη σύγκριση μιας περιοδικής κυματομορφής η οποία μεταβάλλεται γραμμικά με το χρόνο (συνήθως τριγωνική) με μια τάση η οποία προκύπτει από κατάλληλη ενίσχυση του σφάλματος μεταξύ της επιθυμητής και της πραγματικής εξόδου. Όταν η συγκεκριμένη τάση είναι μεγαλύτερη από την τριγωνική το σήμα ελέγχου του ημιαγωγικού στοιχείου γίνεται high με αποτέλεσμα να άγει το ημιαγωγικό στοιχείο ενώ αν είναι μικρότερη το σήμα ελέγχου μηδενίζεται και οδηγείται στην αποκοπή.

Η διακύμανση της τάσης εξόδου μπορεί να ελαττωθεί αρκετά με τη χρήση ενός χαμηλοπερατού φίλτρου πρώτου βαθμού που αποτελείται από μια επαγωγή σε σειρά και ένα πυκνωτή συνδεδεμένο παράλληλα στο φορτίο. Η συχνότητα αποκοπής του φίλτρου



επιλέγεται πολύ μικρότερη από τη συχνότητα λειτουργίας του μετατροπέα εξαλείφοντας ουσιαστικά την κυμάτωση στην τάση εξόδου.

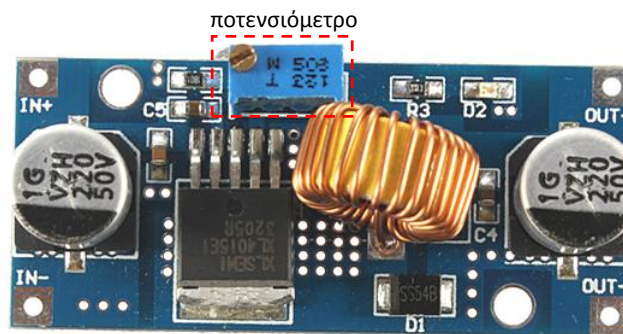


Εικόνα 3.4 Παραγωγή παλμών PWM

Κατά τη χρονική στιγμή που το ημιαγωγικό στοιχείο οδηγείται σε κατάσταση αποκοπής εξαιτίας της απότομης μεταβολής του ρεύματος της επαγωγής θα εμφανιστεί στα άκρα του μια υπέρταση με κίνδυνο αυτό να καταστραφεί. Για την αντιμετώπιση αυτού του προβλήματος συνήθως τοποθετείται μια διόδος η οποία είναι πολωμένη ανάστροφα από την  $V_{in}$  όταν άγει το ημιαγωγικό στοιχείο ενώ όταν ο ημιαγωγικός διακόπτης είναι ανοιχτός το ρεύμα βρίσκει αγωγίμο δρόμο μέσω της διόδου και ένα μέρος της αποθηκευμένης ενέργειας του πηνίου μεταφέρεται στο φορτίο. Το ρεύμα στο πηνίο δεν μπορεί να ελαττωθεί απότομα έτσι την στιγμή που ανοίγει ο διακόπτης το ρεύμα μένει σταθερό και η διαδρομή του είναι μέσα από την διόδο και την επαγωγή στο φορτίο.

### 3.1.2.1 Διακοπτικού τύπου κύκλωμα μετατροπής DC σε DC

Για την μετατροπή της τάσης του συσσωρευτή (ελάχιστα επιτρεπόμενη: 20.4V, μέγιστη: 25.2V) και τον σταδιακό υποβιβασμό της σε πρώτο επίπεδο στα 12V για την παροχή ενέργειας στον ανεμιστήρα και την περαιτέρω μετατροπή της για την τροφοδοσία των ηλεκτρονικών ελέγχου αλλά και για την παροχή τάσης 5V στο Odroid XU3 του οποίου η απαίτηση σε ρεύμα μπορεί να φθάσει έως και τα 4A χρησιμοποιήθηκαν δυο μετατροπείς dc σε dc με βάση το ολοκληρωμένο κύκλωμα XL4015. Το εύρος της τάσης εισόδου της διάταξης είναι από 8V έως 36V και η τάση εξόδου ρυθμίζεται μέσω του ποτενσιόμετρου (αντίσταση  $R_2$ ) από 1.25V έως 32V. Η διάταξη διαθέτει ικανότητα αδιάλειπτης παροχής ρεύματος 5A.



Εικόνα 3.5 DC σε DC μετατροπέας υποβιβασμού τάσης με βάση το ολοκληρωμένο XL4015

Το chip XL4015 περιέχει κύκλωμα για την παραγωγή σταθεροποιημένης τάσης αναφοράς  $V_{ref} = 1.25V$  η οποία συγκρίνεται με ένα κλάσμα  $R_1/(R_1 + R_2)$  της εξόδου  $V_{out}$  από έναν διαφορικό ενισχυτή σφάλματος. Στο chip υπάρχει ακόμη μια γεννήτρια τριγωνικής κυματομορφής με συχνότητα 180 KHz της οποία η έξοδος εφαρμόζεται στον ακροδέκτη μη αναστροφής ενός κυκλώματος σύγκρισης που λειτουργεί σαν διαμορφωτής εύρους παλμού (**Pulse Width Modulator**). Η τάση εξόδου του ενισχυτή σφάλματος εφαρμόζεται στην αναστρέφουσα είσοδο του συγκριτή όπως φαίνεται στο σχήμα της εικόνας 3.4. Αυτός ο διαμορφωτής παράγει τετραγωνική κυματομορφή με περίοδο  $T = 1/180\text{ms}$  της οποίας το τμήμα on μεταβάλλεται γραμμικά με την τάση ελέγχου. Η έξοδος του PWM οδηγεί μεταγωγικό διακόπτη ισχύος που παράγει μια τετραγωνική κυματομορφή περιόδου  $T$  της οποίας η ελάχιστη τιμή είναι 0 και η μέγιστη η τάση εισόδου  $V_{in}$ . Αυτή η τετραγωνική κυματομορφή φιλτράρεται από τον συνδυασμό  $LC$  που λειτουργεί σαν βαθυπερατό φίλτρο. Η σύνθετη αντίσταση του  $C$  είναι πολύ μικρότερη από την σύνθετη αντίσταση του  $L$  οπότε  $\frac{T}{2\pi C} \ll \frac{2\pi L}{T} \Rightarrow \sqrt{LC} = 0.3195 \gg \frac{T}{2\pi} = 8.842 * 10^{-7}$  άρα όλες οι συνιστώσες Fourier της τετραγωνικής κυματομορφής υφίστανται μεγάλη εξασθένηση και η  $V_{out}$  θα είναι σταθερή ίση με την μέση τιμή της τετραγωνικής κυματομορφής. Επειδή υπάρχει φαινομενικό βραχυκύκλωμα μεταξύ των ακροδεκτών εισόδου του ενισχυτή σφάλματος θα είναι

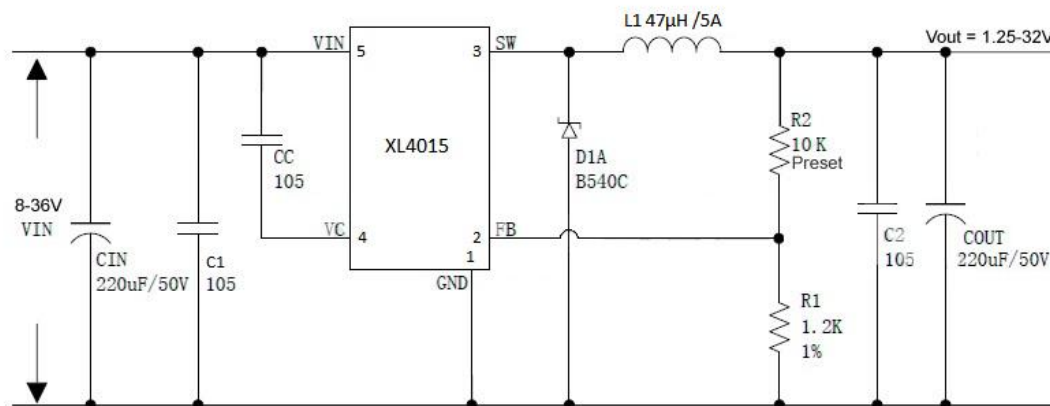
$$V_{ref} = \frac{R_1}{R_1 + R_2} V_{out} \Rightarrow V_{out} = \left(1 + \frac{R_2}{R_1}\right) V_{ref}$$

Η σταθεροποιημένη τάση εξόδου είναι ανεξάρτητη από διακυμάνσεις της τάσης εισόδου  $V_{in}$  και από τις μεταβολές του ρεύματος φορτίου, εξαρτάται μόνο από την τάση  $V_{ref}$  και το λόγο των αντιστάσεων  $R_2/R_1$ . Η συνδεσμολογία αυτή μπορεί να χρησιμοποιηθεί μόνον αν  $V_{in} > V_{out}$ .

Η απόδοση της μετατροπής ξεπερνά το 90% γιατί χρησιμοποιείται ημιαγωγικός διακόπτης με μικρές απώλειες (μικρή  $V_{CE}$  στον κόρο και μεγάλη ταχύτητα μεταγωγής) και στο φίλτρο η επαγωγή έχει μικρή ωμική αντίσταση.

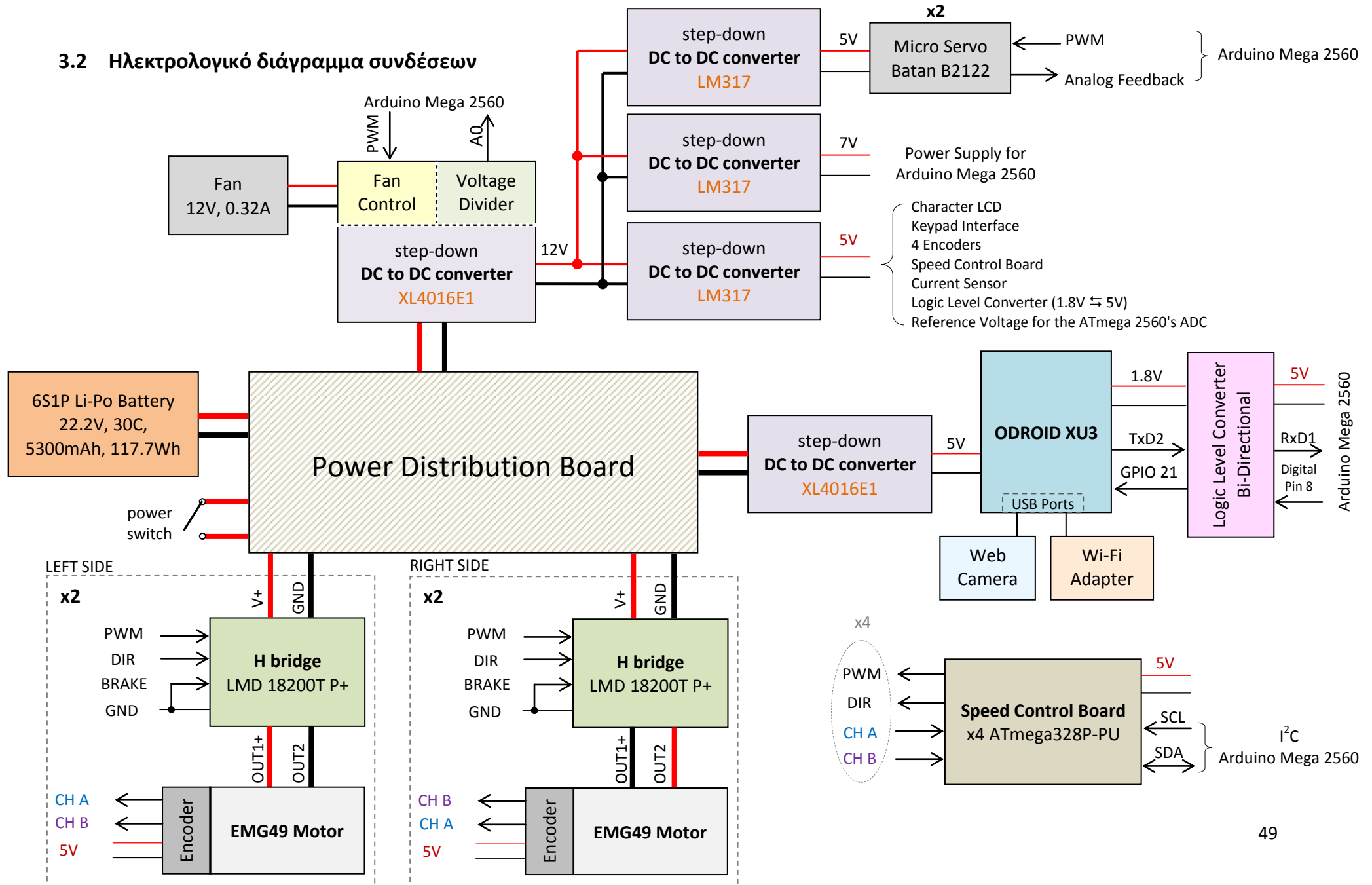
Input	In current	Out	Out current	Power	Efficiency
24V	1A	5V	4.5A	22.5W	93%
24V	2.4A	12V	4.5A	54W	93.7%

Χρησιμοποιείται διόδος Schottky (B540C) αφού προσφέρει ταχεία μετάβαση από την αγωγιμότητα στην αποκοπή και χαμηλό φραγμό δυναμικού (0.3V).



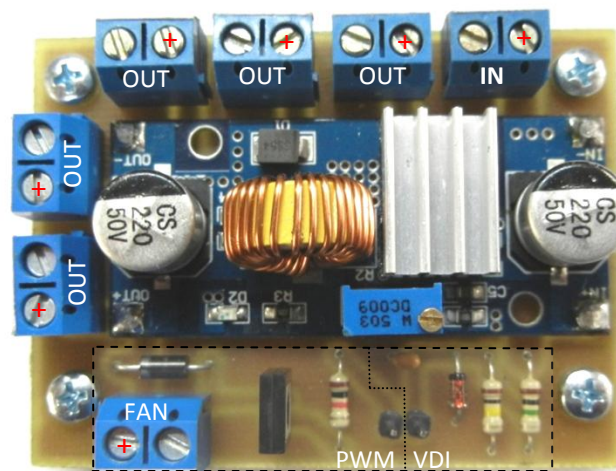
Εικόνα 3.6 Κυκλωματικό διάγραμμα του DC σε DC μετατροπέα

### 3.2 Ηλεκτρολογικό διάγραμμα συνδέσεων



Το τυπωμένο κύκλωμα διανομής της ηλεκτρικής ισχύος περιλαμβάνει κατάλληλο αριθμό κλεμών για την κατανομή της ηλεκτρικής ενέργειας στους ηλεκτροκινητήρες (μέσω των Η-γεφυρών) και στον λοιπό ηλεκτρονικό εξοπλισμό του οχήματος (μέσω των ρυθμιστών τάσεως). Σε αυτό συνδέεται μέσω μιας κλέμας ο κεντρικός διακόπτης ηλεκτρικής τροφοδοσίας (On-Off) του οχήματος και ενσωματώνονται πέντε ασφαλειοθήκες (ασφάλειες τήξεως με γυάλινο περίβλημα διαστάσεως 5x20mm) με τοποθετημένες σε αυτές ασφάλειες των 5A για την προστασία κάθε μιας εκ των τεσσάρων Η-γέφυρων (και κατ' επέκταση των κινητήρων) και μια κεντρική των 10A.

Ο ένας DC σε DC μετατροπέας με βάση το ολοκληρωμένο XL4015 προσαρμόστηκε σε κατάλληλα κατασκευασμένο τυπωμένο κύκλωμα το οποίο διαθέτει έξι κλέμες για την διανομή των 12V μια εκ των οποίων αφιερώνεται στην τροφοδοσία του ανεμιστήρα. Στο συγκεκριμένο PCB ενσωματώθηκε ο διαιρέτης τάσης για την μέτρηση της τάσης του συσσωρευτή και το κύκλωμα οδήγησης του ανεμιστήρα.



Εικόνα 3.7 Τυπωμένο κύκλωμα διανομής των 12V



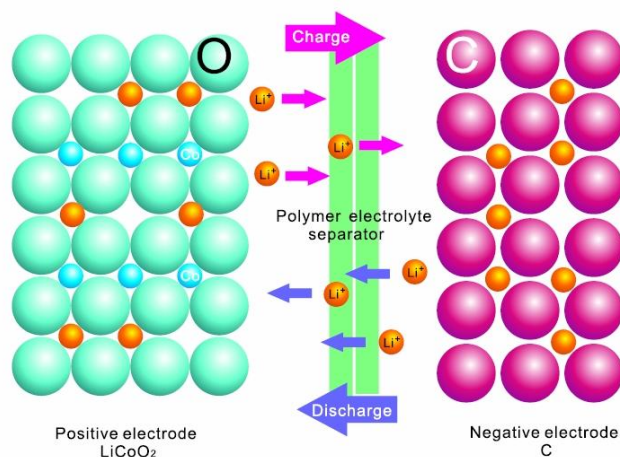
Εικόνα 3.8 DC σε DC μετατροπέας στα 5V με βάση τον γραμμικό σταθεροποιητής τάσης LM317 για την τροφοδοσία των μικροσερβοκινητήρων

### 3.3 Μπαταρίες Ιόντων Λιθίου Πολυμερών (LiPo)

#### 3.3.1 Γενικά στοιχεία

Οι επαναφορτιζόμενες μπαταρίες LiPo καθιερώθηκαν στο χώρο του τηλεκατευθυνόμενου μοντελισμού χάρη στον υψηλό λόγο αποθηκευμένης ενέργειας/βάρους (mAh/g) και τον υψηλό ρυθμό εκφόρτισης με μεγάλες εντάσεις ρεύματος. Είναι επίσης δημοφιλείς λόγω των flat καμπυλών εκφόρτισης και της μεγάλης πυκνότητας ισχύος που διαθέτουν. Το κύριο μειονέκτημα τους είναι το υψηλό κόστος αγοράς.

Τα θετικά ηλεκτρόδια τα οποία χρησιμοποιούνται στις μπαταρίες Li-ion ως ενεργά συστατικά είναι μεταλλικά οξείδια του λιθίου π.χ.  $\text{LiCoO}_2$  (οξείδιο του κοβαλτιούχου λιθίου) και ως υλικό αρνητικού ηλεκτροδίου η πλειοψηφία των μπαταριών χρησιμοποιεί διάφορους τύπους άνθρακα. Τα ηλεκτρόδια είναι ηλεκτρικά απομονωμένα μέσω συμπαγούς φύλλου πολυμερούς υλικού το οποίο είναι μη αγώγιμο αλλά επιτρέπει τη διέλευση ιόντων. Στα στοιχεία (cell) LiPo χρησιμοποιούνται επίπεδες απανωτές στρώσεις λεπτών φύλλων ηλεκτροδίων και διαχωριστικών φύλλων πολυμερούς ηλεκτρολύτη.



Εικόνα 3.9 Σχηματικό διάγραμμα αρχής λειτουργίας μπαταριών LiPo

Αποτελούνται από πολλαπλά στοιχεία (cell) συνδεδεμένα σε σειρά. Το κάθε στοιχείο έχει ονομαστική τιμή τάσης 3.7V. Σειρές στοιχείων δίνουν πολλαπλάσιες τιμές τάσης των 3.7V. Έτσι υπάρχουν μπαταρίες με 1 cell (1S) στα 3.7V, 2 cell (2S) στα 7.4V, 3 cell (3S) στα 11.1V, 4 cell (4S) στα 14.8V και ούτω καθεξής. Όταν ένα στοιχείο είναι πλήρως φορτισμένο έχει μέγιστη τάση 4.2V. Οπότε μια 3 cell μπαταρία πλήρως φορτισμένη θα έχει τάση  $3 \times 4.2V = 12.6V$ . Κατά την εκφόρτιση της μπαταρίας πρέπει να παρακολουθείται η τάση κάθε στοιχείου καθώς κανένα από αυτά δεν πρέπει να πέσει κάτω από τα 3V διαφορετικά αυτή αχρηστεύεται.

Εκτός από την τάση τους οι μπαταρίες LiPo χαρακτηρίζονται και από το ρεύμα που μπορούν να δώσουν. Η χωρητικότητα μιας μπαταρίας ή το πόση ενέργεια μπορεί να αποθηκεύσει μετράται σε mAh. Η τιμή αυτή δηλώνει την ένταση του ρεύματος που η μπαταρία μπορεί να δώσει για μια ώρα. Για να ικανοποιηθούν οι απαιτήσεις χρήσης ενδέχεται να χρησιμοποιείται ένας συνδυασμός εν σειρά στοιχείων σε παράλληλη διάταξη αυξάνοντας έτσι την χωρητικότητα ενώ η τάση παραμένει σταθερή. Για παράδειγμα μια μπαταρία μπορεί να χαρακτηρίζεται ως 3S2P δηλαδή διαθέτει 2 παράλληλα στελέχη με 3 στοιχεία συνδεδεμένα σε σειρά το κάθε στέλεχος.

Ο ρυθμός εκφόρτισης C περιγράφει το πόσο γρήγορα μπορεί να εκφορτιστεί η μπαταρία με ασφάλεια. Για παράδειγμα μια μπαταρία 5000mAh και 30C μπορεί να αποφορτιστεί με ασφάλεια με ρυθμό ίσο με 30 φορές την χωρητικότητας της συνεπώς μπορούν να συνδεθούν φορτία μέχρι  $5 \times 30 = 150A$ . Καθώς η μπαταρία εκφορτίζεται το ρεύμα που έχει αποθηκευμένο μειώνεται έτσι η μπαταρία του παραδείγματος όταν είναι στο 50% της φόρτισης της θα μπορεί να δώσει μόνο  $2.5 \times 30 = 75A$ . Υψηλότερος βαθμός C σημαίνει και υψηλότερο κόστος αγοράς. Στα τεχνικά χαρακτηριστικά καθορίζεται επίσης και το μέγιστο ρεύμα (burst rate) που μπορεί να δώσει μια μπαταρία LiPo για πολύ σύντομο διάστημα συνήθως 10-20 δευτερολέπτων π.χ.60C

Στην συγκεκριμένη εφαρμογή χρησιμοποιείται η μπαταρία LiPo της εταιρείας Gens ace με 6 στοιχεία σε σειρά (6S1P) και τα εξής τεχνικά χαρακτηριστικά:

Χωρητικότητα	5300mAh
Τάση	22.2V
Μέγιστο συνεχές ρεύμα εκφόρτισης	30C (159A)
Μέγιστο στιγμιαίο ρεύμα εκφόρτισης	60C (318A)
Βάρος	800g
Διαστάσεις (ΜxΠxΥ)	139x41x61mm
Ρυθμός φόρτισης	Συνίσταται 1-3C, 5C μέγιστο



Εικόνα 3.10 Gens ace 6S1P LiPo Battery Pack

### 3.3.2 Φόρτιση

Υπάρχουν ειδικοί φορτιστές οι οποίοι φροντίζουν τα στοιχεία της μπαταρίας να έχουν το ίδιο φορτίο (ισοστάθμιση, balancing). Η μπαταρία ποτέ δεν θα πρέπει να φορτίζεται με ρυθμό φόρτισης μεγαλύτερο του 1C εκτός αν αυτό αναφέρεται ξεκάθαρα από τον κατασκευαστή. Ο τύπος για τον υπολογισμό είναι:

$$XXXX \text{ mAh} / 1000 \times (\text{Charging C rating}) = XX \text{ amps}$$

Γενικότερα με όσο λιγότερα Ampere φορτίζεται μια μπαταρία τόσο μεγαλύτερο είναι το χρονικό διάστημα που διαρκεί η φόρτιση της. Βέβαια ο αργός ρυθμός φόρτισης διασφαλίζει μακροπρόθεσμα μεγαλύτερη διάρκεια ζωής για την μπαταρία.

Στην προκειμένη περίπτωση χρησιμοποιήθηκε ο φορτιστής iMax B6AC ο οποίος μπορεί να φορτίσει ή εκφορτίσει μπαταρίες NiCd/NiMH, Li-ion, LiPo, LiFe, Pb. Διαθέτει μικροεπεξεργαστή με εξειδικευμένες λειτουργίες και διάφορες μεθόδους φόρτισης και φωτιζόμενη οθόνη πολλαπλών ενδείξεων. Διαθέτει εξισορροπιστή και μπορεί να παρακολουθεί κάθε στοιχείο της μπαταρίας ξεχωριστά. Όταν η τάση της μπαταρίας υπερβεί το όριο ή το μέγιστο χρόνο φόρτισης που έχει ορίσει ο χρήστης η διαδικασία διακόπτεται αυτόματα. Μπορεί επίσης να ρυθμιστεί το ανώτερο όριο του ρεύματος φόρτισης και ο χρήστης να αποθηκεύσει τις παραμέτρους φόρτισης/εκφόρτισης για συγκεκριμένο τύπο μπαταρίας.



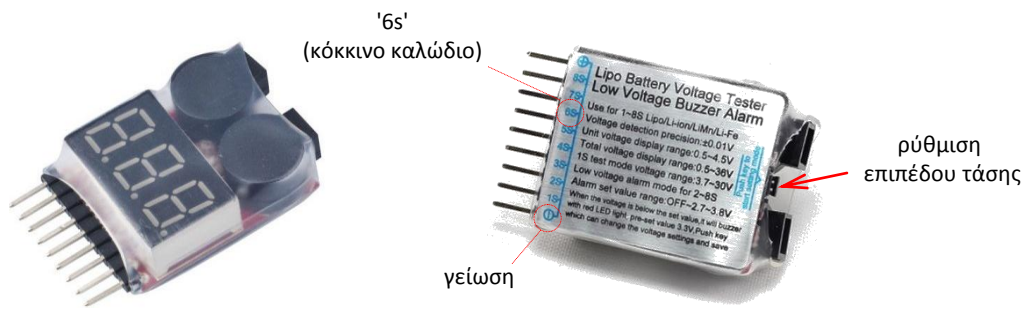
Εικόνα 3.11 Φορτιστής μπαταριών iMax B6AC

### 3.3.3 Προφυλάξεις

Αν βραχυκυκλωθούν οι ακροδέκτες ή ξεπεραστούν τα όρια που θέτει ο κατασκευαστής μπορεί οι μπαταρίες LiPo να γίνουν εξαιρετικά επικίνδυνες. Σε καμιά περίπτωση δεν πρέπει να κοπεί ή να τρυπηθεί το εξωτερικό κέλυφος της μπαταρίας. Τα ενεργά χημικά συστατικά που περιέχουν οι LiPo διαχωρίζονται από μονωτικό υλικό εσωτερικά των στοιχείων το οποίο αν διαρραγεί και αυτά έρθουν απότομα σε επαφή τότε η αντίδραση είναι βίαιη και παράγεται μεγάλη ποσότητα αερίων με παράλληλη αύξηση της θερμοκρασίας, συνδυασμός που ενδέχεται να οδηγήσει σε βίαιη εκτόνωση και έκρηξη. Πυρκαγιά επίσης μπορεί να προκληθεί και από την επαφή του λιθίου με το οξυγόνο του αέρα.

Δεν πρέπει να μένουν αποθηκευμένες πλήρως φορτισμένες για μεγάλο χρονικό διάστημα διότι μειώνεται η δυνατότητα τους να αποκτούν όλη την ονομαστική χωρητικότητά τους. Αν η μπαταρία δεν πρόκειται να χρησιμοποιηθεί άμεσα καλό είναι να φορτίζονται μέχρι το 60%. Αρκετοί φορτιστές διαθέτουν για τον σκοπό αυτό ειδικό πρόγραμμα φόρτισης για μακροπρόθεσμη αποθήκευση. Αν οι μπαταρίες χτυπηθούν με βίαιο τρόπο ή αν κάποιο στοιχείο φουσκώσει κατά την διάρκεια της φόρτισης θα πρέπει να απομονωθούν σε ασφαλές σημείο γιατί ενδέχεται να αναφλεγούν. Κατά την φόρτιση ποτέ μια μπαταρία LiPo να φορτίζεται χωρίς επίβλεψη.

### 3.3.4 Αδιάλειπτος έλεγχος του επιπέδου της τάσης του συσσωρευτή



Εικόνα 3.12 Άνω και κάτω όψη της συσκευής LiPo Tester

Για την αποτροπή μόνιμης βλάβης λόγω χαμηλής τάσης σε κάποιο από τα εν σειρά στοιχεία (cell) της μπαταρίας χρησιμοποιείται κατάλληλη διάταξη ελέγχου (LiPo battery low voltage tester) για την συνεχή επίβλεψη του επιπέδου της τάσης σε κάθε ένα από αυτά. Η συσκευή διαθέτει δυνατότητα προγραμματισμού του επιθυμητού επιπέδου τάσης μέσω του κατάλληλου κουμπιού που αυτή διαθέτει.

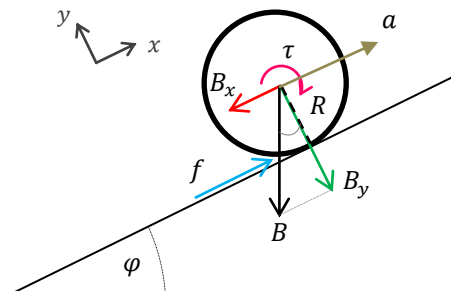
Όταν η τάση της μπαταρίας σε κάποιο από τα εν σειρά στελέχη της ανιχνευθεί με τιμή χαμηλότερη από το επίπεδο τάσης που έχει προρυθμιστεί παράγεται ανάλογο προειδοποιητικό ηχητικό σήμα από τον διατιθέμενο σε αυτήν βομβητή.

#### Χαρακτηριστικά συσκευής

Κατάλληλη για μπαταρίες LiPo	1~8S
Ακριβής ανίχνευση τάσης	$\pm 0.01V$
Εύρος τάσης για κάθε στοιχείο	0.5 ~ 4.5V
Συνολικό φάσμα ένδειξης τάσης	0.5 ~ 36V
Εύρος τάσης ενεργοποίησης βομβητή	απενεργοποίηση ~ 2.7 ~ 3.8V
Διαστάσεις	39x24x11 mm
Βάρος	9g

### 3.4 Εκτίμηση απαιτούμενης ισχύος και χωρητικότητας συσσωρευτή

Για τον υπολογισμό της απαιτούμενης ισχύος για την μετακίνηση της ρομποτικής πλατφόρμας θεωρείται ότι το βάρος της  $B$  είναι ομοιόμορφα κατανομημένο στους 4 τροχούς της και ότι αυτοί ισαπέχουν από το κέντρο μάζας  $M$  του οχήματος (και οι δυο υποθέσεις είναι αρκετά κοντά στην πραγματικότητα). Υποθέτουμε ότι το όχημα ανέρχεται σε κεκλιμένο επίπεδο κλίσεως  $\varphi$  χωρίς να ολισθαίνει έχοντας επιτάχυνση  $a$ . Η μοναδική δύναμη της οποίας η ροπή είναι υπεύθυνη για την στροφική κίνηση κάθε τροχού ακτίνας  $R$  είναι η τριβή  $f$ . Η φορά της τριβής πρέπει να είναι τέτοια ώστε η ροπή της να τείνει να περιστρέψει τον τροχό κατά αντίθετη φορά από την φορά περιστροφής του προκαλώντας την μείωση της γωνιακής ταχύτητας του  $\omega$ .



Εικόνα 3.13 Ανάλυση δυνάμεων σε κινητήριο τροχό κατά την άνοδο του οχήματος σε κεκλιμένο επίπεδο



Η απαιτούμενη ροπή είναι

$$\tau = fR \rightarrow f = \frac{\tau}{R}$$

Για την μεταφορική κίνηση του τροχού έχουμε

$$\sum F_x = ma \rightarrow f - mg\sin\varphi = ma$$

αντικαθιστώντας στην προηγούμενη σχέση την μάζα  $m$  του τροχού, η οποία θεωρήθηκε ότι είναι ίση με  $M/4$ , και την δύναμη της τριβής προκύπτει ότι

$$\tau = \frac{MR(\alpha + g\sin\varphi)}{4}$$

Η κινητήριος ισχύς είναι

$$P_{out} = \tau\omega$$

όπου  $\omega$  η γωνιακή ταχύτητα του τροχού

Ως συντελεστής απόδοσης καθορίζεται ως ο λόγος

$$e = \frac{P_{out}}{P_{in}} 100$$

Οπότε η προσφερθείσα ισχύς εισόδου ανά κινητήρα θα πρέπει να είναι

$$P_{in} = \frac{100}{e} \frac{MR(\alpha + g\sin\varphi)}{4} \omega$$

θέτοντας ενδεικτικά

$$e = 50\%$$

$$M = 10Kg$$

$$R = \frac{154}{2} mm = 0.077m$$

$$a = 0 \frac{m}{sec^2} \text{ (σταθερή ταχύτητα)}$$

$$g = 9.81 m/sec^2$$

$$\varphi = 15^\circ$$

$$v = 1 \frac{m}{sec} \rightarrow v = \omega R \rightarrow \omega = 13 \frac{rad}{s}$$

υπολογίζεται ότι  $P_{in} = 12.71 W$  και  $\tau = 0.98 Nm$

Αν η τάση τροφοδοσίας είναι  $V_{cc} = 24V$  τότε το καταναλισκόμενο ρεύμα ανά κινητήρα είναι

$$P_{in} = IV_{cc} \rightarrow I = \frac{P_{in}}{V_{cc}} = 0.53A$$

Για τον υπολογισμό της χωρητικότητας της μπαταρίας είναι γνωστό ότι

$$Battery\ Life\ (hours) = \frac{Battery\ Capacity\ (Ah)}{Current\ Draw\ (A)}$$

συνεπώς για 60 min συνεχούς λειτουργίας του οχήματος θα έχουμε

$$1h * 4 * 0.53A = 2.12Ah$$

Κατά συνέπεια οι κινητήρες και η μπαταρία LiPo που επιλέχθηκαν κρίνονται ικανοποιητικές ως προς τις επιδόσεις τους και τα τεχνικά τους χαρακτηριστικά για τις απαιτήσεις της συγκεκριμένης εφαρμογής.

## Κεφάλαιο

# 4

## 4. Μονάδες εισόδου/εξόδου - Αισθητήρια

Στο παρόν κεφάλαιο περιγράφεται ο τρόπος σύνδεσης και το ρεπερτόριο εντολών της βιβλιοθήκης προγραμματισμού της οθόνης χαρακτήρων μέσω της οποίας προβάλλονται τόσο οι τρέχουσες μετρούμενες ποσότητες από τα αντίστοιχα αισθητήρια, όπως το επίπεδο τάσης του συσσωρευτή, το καταναλισκόμενο ρεύμα, η θερμοκρασία εντός του κουτιού των ηλεκτρονικών ελέγχου και η εκτιμώμενη γωνία γαω του οχήματος αλλά και τα δεδομένα κατά την καταχώριση τιμών μέσω του πληκτρολογίου για την εν λειτουργία παραμετροποίηση του συστήματος χωρίς να απαιτείται η τροποποίηση του firmware του μικροελεγκτή ATmega2560. Παρουσιάζεται επίσης ο τρόπος διασύνδεσης και ανάγνωσης του πληκτρολογίου 4x4 μέσω ενός εξειδικευμένου ολοκληρωμένου κυκλώματος καθώς και η αντίστοιχη προγραμματισμένη λειτουργία που έχει αποδοθεί σε συγκεκριμένα πλήκτρα.

Στην δεύτερη ενότητα αρχικά αναφέρονται τα χρησιμοποιούμενα χαμηλοπερατά ψηφιακά φίλτρα για την εξομάλυνση των διακριτοποιημένων τιμών εξόδου των αισθητηρίων και ακολούθως οι αισθητήριες διατάξεις εσωτερικής κατάστασης (internal state sensors) με τις οποίες είναι εξοπλισμένη η ρομποτική πλατφόρμα:

- ο διαιρέτης τάσης με τον οποίο μετράται το επίπεδο της τάσης του συσσωρευτή,
- το αισθητήριο θερμοκρασίας μέσω του οποίου μετράται η θερμοκρασία εντός του κουτιού των ηλεκτρονικών ελέγχου
- ο αισθητήρας ρεύματος μέσω του οποίου μετράται κάθε χρονική στιγμή η συνολική κατανάλωση ρεύματος
- η ηλεκτρονική πυξίδα και τα αισθητήρια μέτρησης αδρανειακών μεγεθών από τον συγκερασμό των δεδομένων των οποίων προκύπτει η εκτίμηση της γωνίας γαω του οχήματος (η τεκμηρίωση δίνεται στο Κεφάλαιο 10).

Περιγράφεται τέλος ο τρόπος σύνδεσης και οδήγησης του ανεμιστήρα του οποίου η ταχύτητα περιστροφής ρυθμίζεται βάση της μετρούμενης θερμοκρασίας ώστε να ψύχονται επιτυχώς τα ηλεκτρονικά κυκλώματα ισχύος.

### 4.1 Μονάδες εισόδου/εξόδου

#### 4.1.1 Διασύνδεση και έλεγχος οθόνης LCD

Για την απεικόνιση πληροφοριών και την επικοινωνία του χρήστη με το ρομποτικό όχημα χρησιμοποιείται μια οθόνη χαρακτήρων υγρών κρυστάλλων (LCD) τύπου 20x4, διαθέτει δηλαδή 4 γραμμές των 20 χαρακτήρων. Κάθε χαρακτήρας απαρτίζεται από ένα block 5x8 εικονοστοιχείων τα οποία συνδυάζονται κατάλληλα για την απεικόνιση του εκάστοτε συμβόλου. Η μονάδα ενσωματώνει τον κυρίως ελεγκτή ο οποίος είναι συμβατός

με τον HDD44780 της Hitachi και μνήμη ROM στην οποία βρίσκεται κωδικοποιημένο το σύνολο των χαρακτήρων της.

Για εξοικονόμηση ψηφιακών ακροδεκτών I/O του Arduino η οθόνη λειτουργεί σε 4 bit mode χρησιμοποιούνται δηλαδή 4 γραμμές δεδομένων οπότε κάθε byte που αποστέλλεται από τον μικροελεγκτή χωρίζεται σε δυο nibbles με συνέπεια να απαιτείται ελαφρώς περισσότερος χρόνος για την αλληλεπίδραση μεταξύ μικροελεγκτή και της οθόνης LCD.

Σύνδεση ακροδεκτών LCD με τον Arduino Mega 2560

Ακροδέκτης LCD		Περιγραφή	Ακροδέκτης Arduino
1	$V_{SS}$	Ground	γείωση
2	$V_{DD}$	Supply voltage	5V
3	$V_O$	Contrast adjust	μεσαία λήψη ποτενσιόμετρου
4	$RS$	Register Select signal High: data, Low: instruction code	22
5	$R/W$	H: Read (LCD →MPU) L: Write (MPU→LCD)	γείωση
6	$E$	Chip enable signal	23
7-10	$DB0 - 3$	Data bus lines	-
11	$DB4$	Data bus line	24
12	$DB5$	Data bus line	25
13	$DB6$	Data bus line	26
14	$DB7$	Data bus line	27
15	$A$	Backlight +	5V
16	$K$	Backlight -	γείωση



Εικόνα 4.1 Η χρησιμοποιούμενη 20x4 LCD οθόνη χαρακτήρων με μπλέ οπίσθιο φωτισμό

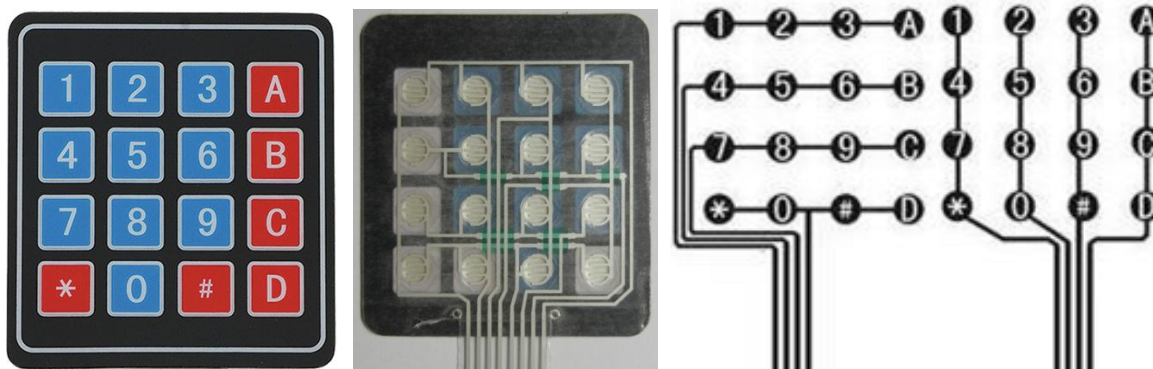
Για την διευκόλυνση της διασύνδεσης της με τον ATmega2560 κατασκευάστηκε τυπωμένο κύκλωμα το οποίο φέρει εκτός των κατάλληλων υποδοχών-ακροδεκτών και ποτενσιόμετρο (vertical trimmer 10KΩ) ώστε να υπάρχει δυνατότητα ρύθμισης της αντίθεσης ανάλογα με τις συνθήκες φωτισμού. Για την ενεργοποίηση του οπίσθιου φωτισμού και για την προστασία του τοποθετήθηκε αντίσταση 330Ω.

Στη βασική έκδοση του Arduino IDE περιλαμβάνεται η βιβλιοθήκη LiquidCrystal.h με μια σειρά από συναρτήσεις οι οποίες διευκολύνουν σε μεγάλο βαθμό τον προγραμματισμό των μονάδων LCD που βασίζονται στον ελεγκτή HDD44780.

Οι συναρτήσεις της βιβλιοθήκης που χρησιμοποιήθηκαν είναι:

- `LiquidCrystal(rs, enable, d4, d5, d6, d7)`  
Δημιουργία αντικειμένου τύπου `LiquidCrystal`. Δήλωση χρησιμοποιούμενων ψηφιακών ακροδεκτών αφιερωμένων στην λειτουργία της οθόνης.
- `begin(cols,rows)`  
Συνάρτηση αρχικοποίησης και καθορισμός πλήθους στηλών, γραμμών οθόνης.
- `clear()`  
Καθαρισμός οθόνης και τοποθέτηση θέσης εκτύπωσης στη γραμμή 0, στήλη 0.
- `setCursor(col,row)`  
Καθορισμός θέσης εκτύπωσης στήλη, γραμμή.
- `write(character)`  
Συνάρτηση η οποία εμφανίζει στην οθόνη στην τρέχουσα θέση εκτύπωσης το ASCII σύμβολο του χαρακτήρα
- `print(data, [num])`  
Συνάρτηση η οποία εμφανίζει στην οθόνη στην τρέχουσα θέση έναν αριθμό ή έναν χαρακτήρα ή μια συμβολοσειρά.
- `createChar(num,array)`  
Με την συνάρτηση `createChar` δίνεται η δυνατότητα ορισμού και απεικόνισης έως και 8 εξειδικευμένων χαρακτήρων - συμβόλων (`glyphs`) οι οποίοι δεν ανήκουν στο σύνολο χαρακτήρων της οθόνης. Για τον σχηματισμό και την εύρεση της μήτρας των δυαδικών ψηφίων (εικονοστοιχείων) διατίθεται πληθώρα βοηθητικών εφαρμογών στο διαδίκτυο μιας εκ των οποίων είναι η [5x8 LCD Display Custom Character Designer by ETC](#)

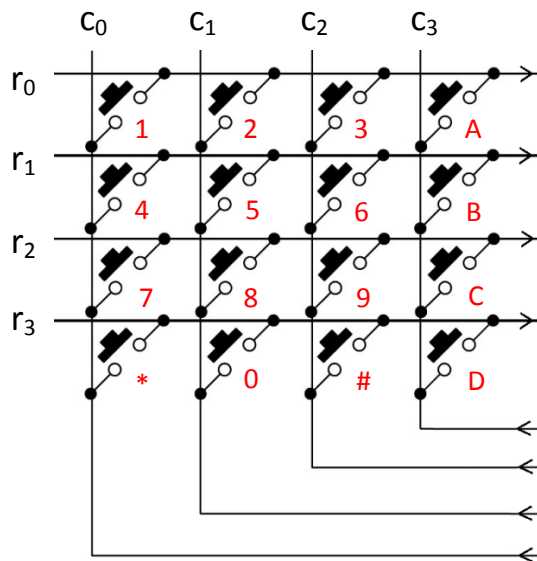
#### 4.1.2 Διασύνδεση και ανάγνωση πληκτρολογίου 4x4



Εικόνα 4.2 Άνω και κάτω όψη του πληκτρολογίου. Διασύνδεση γραμμών και στηλών.

Το πληκτρολόγιο μεμβράνης που χρησιμοποιήθηκε είναι ανθεκτικό στη σκόνη, αδιάβροχο και αποτελεί ένα πίνακα διακοπών (4x4 key matrix) με κάθε πλήκτρο όταν πιέζεται να δημιουργεί μια ηλεκτρική σύνδεση μεταξύ της γραμμής και της στήλης στις οποίες ανήκει.

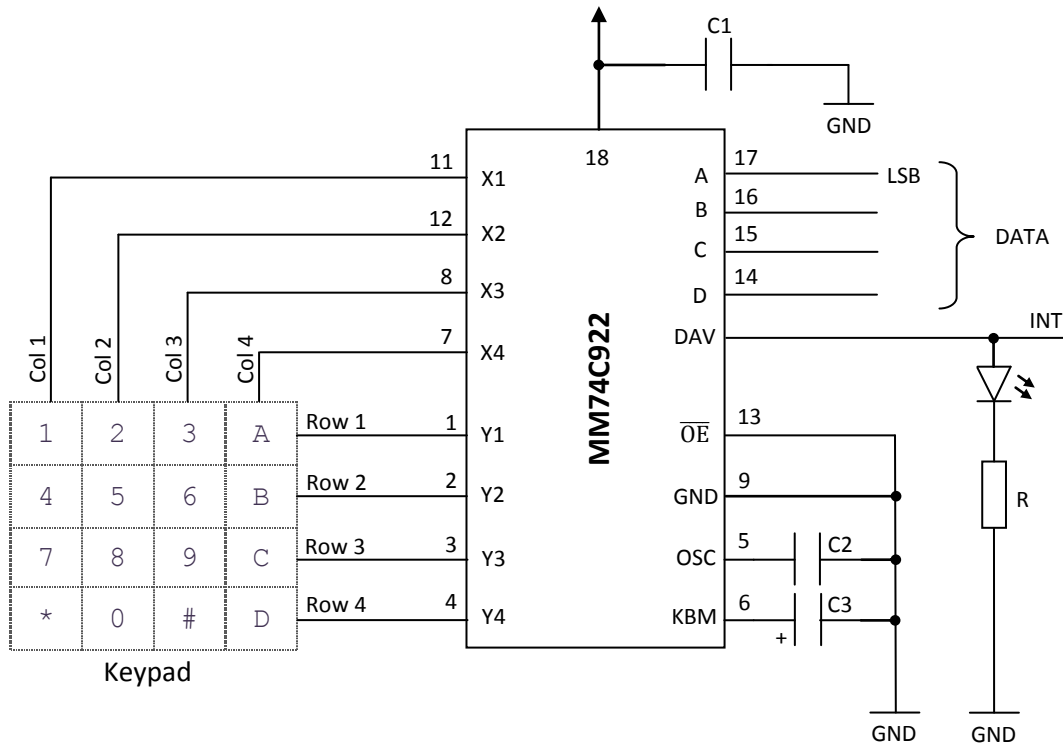
Προκειμένου να ανιχνευθεί τυχόν πάτημα κάποιου πλήκτρου συνήθως ακολουθείται η μέθοδος της συνεχούς σάρωσης (rolling) του πληκτρολογίου. Οι ακροδέκτες των γραμμών και των στηλών συνδέονται σε μια 8-bit θύρα του μικροελεγκτή. Οι ακροδέκτες των στηλών ενεργοποιούνται ως έξοδοι και γράφεται σε αυτούς 1 και οι ακροδέκτες των γραμμών καθίστανται ως εισοδοί με ενεργοποιημένες τις εσωτερικές αντιστάσεις pull-up. Κατά την διαδικασία της σάρωσης εφαρμόζεται διαδοχικά σε κάθε ακροδέκτη εξόδου χαμηλό δυναμικό (0), όπου αν δεν πατηθεί κάποιο πλήκτρο όλες οι εισοδοί είναι σε υψηλό (1) λόγω των αντιστάσεων pull-up, αν όμως υπάρξει ηλεκτρική σύνδεση μεταξύ της στήλης της οποίας ο ακροδέκτης τέθηκε 0 και κάποιας γραμμής λόγω πατήματος του αντίστοιχου πλήκτρου τότε θα εμφανιστεί 0 και στον συγκεκριμένο ακροδέκτη εισόδου. Η σύνθεση των 4 bit που αποστέλλονται στις στήλες π.χ. 1011 με των 4 bit που διαβάζονται στις γραμμές δημιουργεί έναν κωδικό ο οποίος είναι μοναδικός για κάθε πλήκτρο.



Εικόνα 4.3 Σάρωση πληκτρολογίου

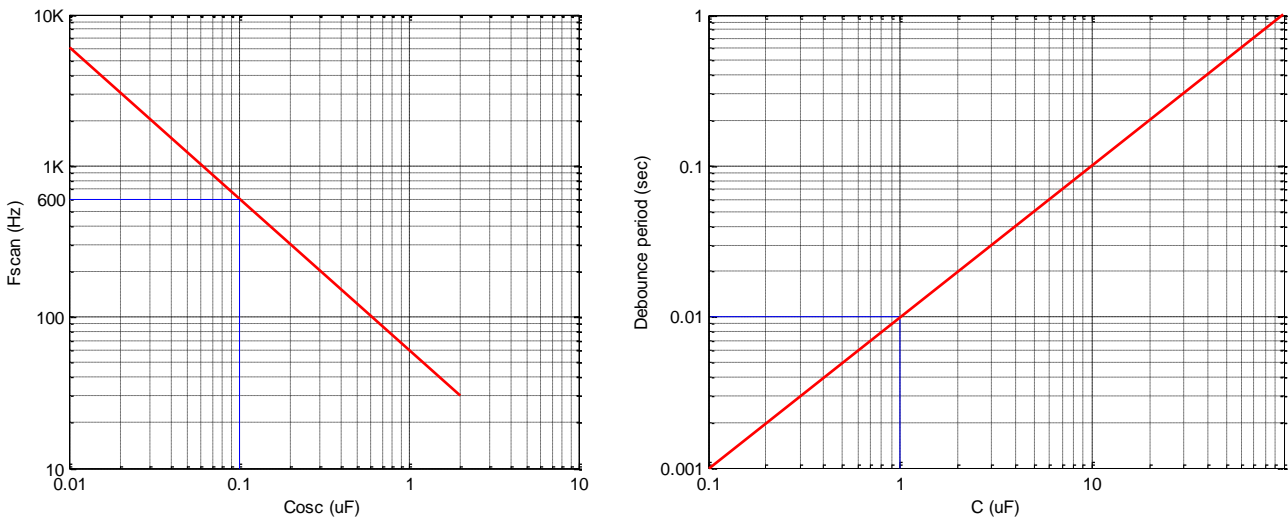
Αυτή η συνεχής παρακολούθηση συνεπάγεται χρονική επιβάρυνση, η οποία γίνεται μεγαλύτερη με την αύξηση της συχνότητας σάρωσης, αφού σημαντικό μέρος της επεξεργαστικής ισχύς αναλώνεται στην διαδικασία αυτή.

Μια άλλη προσέγγιση της διασύνδεσης του πληκτρολογίου στον  $\mu\text{E}$  είναι η χρήση ενός εξειδικευμένου ολοκληρωμένου κυκλώματος. Το IC το οποίο χρησιμοποιήθηκε είναι το **MM74C922** της εταιρείας Fairchild, διατίθεται σε συσκευασία DIP 18 ακροδεκτών και είναι επιφορτισμένο τόσο με τη σάρωση του πληκτρολογίου όσο και με την αντιμετώπιση της εμφάνισης των αναπηδήσεων (button bounce) όταν αλλάζει η κατάσταση ενός εκ των διακόπτων (τη στιγμή που πατιέται ή αφήνεται το πλήκτρο). Όταν πατηθεί κάποιο πλήκτρο παράγεται κατάλληλο ηλεκτρικό σήμα στον ακροδέκτη Data Available το οποίο μπορεί να χρησιμοποιηθεί για την πρόκληση εξωτερικής διακοπής (interrupt) σε  $\mu\text{E}$  και αποστέλλεται ο ανάλογος 4-bit κωδικός που αντιστοιχεί στο πλήκτρο. Ο ακροδέκτης εξόδου Data Available του chip αλλάζει κατάσταση και γίνεται 1 όταν επικαιροποιηθεί το πάτημα ενός πλήκτρου και επιστρέφει σε 0 όταν το πλήκτρο απελευθερώνεται ακόμη και αν έχει κατά τη διάρκεια πατηθεί ένα διαφορετικό πλήκτρο. Μετά το χρονικό διάστημα που απαιτείται για την παρέλευση των αναπηδήσεων ο ακροδέκτης επιστρέφει σε 1 για δηλωθεί η αποδοχή του νέου πλήκτρου (two-key roll-over). Η συχνότητα σάρωσης μπορεί να καθοριστεί είτε από εξωτερικό ρολόι ή εξωτερικό πυκνωτή. Το ενσωματωμένο κύκλωμα για το debouncing απαιτεί σύνδεση εξωτερικού πυκνωτή η χωρητικότητα του οποίου καθορίζεται από την χρονική διάρκεια των αναπηδήσεων.



Εικόνα 4.4 Συνδεσμολογία του MM74C922 με το keypad

Το κύκλωμα το οποίο κατασκευάστηκε εμφανίζεται στην εικόνα 4.4. Οι τιμές των πυκνωτών για το debouncing και την συχνότητα σάρωσης είναι 1  $\mu\text{F}$  και 0.1  $\mu\text{F}$  αντίστοιχα. Όποτε σύμφωνα με τα γραφήματα τα οποία διατίθενται στο datasheet ο χρόνος αναμονής για τον τερματισμό των αναπηδήσεων ορίζεται στα 10 ms και η συχνότητα σάρωσης στα 600 Hz. Η ενεργοποίηση της φωτοδιόδου αποτελεί οπτική ένδειξη και δηλώνει την έγκυρη καταχώρηση ενός πλήκτρου.



Εικόνα 4.5 Καθορισμός συχνότητας σάρωσης πληκτρολογίου και χρονικού διαστήματος για το debouncing

Ο παρακάτω πίνακας δίνει την αντιστοιχία του 4-bit κωδικού των δεδομένων που αποστέλλεται από το IC και του αναγραφόμενου συμβόλου στο πλήκτρο.

nibble	D	C	B	A	Key
1	0	0	0	0	1
2	0	0	0	1	2
3	0	0	1	0	3
4	0	0	1	1	A
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	B
9	1	0	0	0	7
10	1	0	0	1	8
11	1	0	1	0	9
12	1	0	1	1	C
13	1	1	0	0	*
14	1	1	0	1	0
15	1	1	1	0	#
16	1	1	1	1	D

Πλήκτρο	Λειτουργία
0	Τερματισμός Odroid-XU3
1	Επαναλαμβανόμενο τραπεζοειδές προφίλ κίνησης
A	Εκκίνηση/Ακινητοποίηση οχήματος
B	Καθορισμός μέγιστης γωνιακής ταχύτητας κινητήρων σε rad/s
C	Αυτόνομη πλοήγηση εντός καλλιέργειας σειρών
D	Καθορισμός γωνίας κλίσης της κάμερας από 0 έως και 45°
*	Επανεκκίνηση του μικροεπεξεργαστή ATmega328 ο οποίος είναι επιφορτισμένος με τον υπολογισμό του προσανατολισμού (heading) του οχήματος

Η καταχώριση των αριθμητικών δεδομένων (γωνία κλίσης κάμερας, μέγιστη γωνιακή ταχύτητα κινητήρων) επιτυγχάνεται με το πλήκτρο # ενώ σε τυχόν εσφαλμένη πληκτρολόγηση η διαγραφή των ήδη εισαχθέντων αριθμών πραγματοποιείται με το πλήκτρο C.

## 4.2 Αισθητήρια εσωτερικής κατάστασης

### 4.2.1 Χρησιμοποιούμενα ψηφιακά φίλτρα

Για την απομάκρυνση των ανεπιθύμητων υψίσυχων συνιστωσών του δειγματοληπτημένου σήματος εξόδου των αισθητηρίων ρεύματος, θερμοκρασίας αλλά της μεσαίας λήψης του διαιρέτη τάσης χρησιμοποιήθηκαν τα κάτωθι ψηφιακά χαμηλοπερατά φίλτρα.

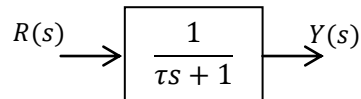
#### 4.2.1.1 Απλό φίλτρο κινούμενου μέσου

Το φίλτρο κινούμενου μέσου όρου είναι ένα FIR βαθυπερατό φίλτρο το οποίο εξαλείφει τις υψηλές συχνότητες και εισάγει καθυστέρηση φάσης. Στην έξοδο του φίλτρου παίρνουμε τον μέσο όρο των  $n$  πρόσφατων δειγμάτων της εισόδου. Οπότε:

$$y_k = \frac{1}{n} \sum_{i=0}^{n-1} x_{k-i} = \frac{x_k + x_{k-1} + x_{k-2} + \dots + x_{k-(n-1)}}{n}$$

Η είσοδος μετράται σε τακτά χρονικά διαστήματα. Η έξοδος του φίλτρου εξομαλύνει το σήμα εισόδου και ανταποκρίνεται στις μεταβολές της με ορισμένη χρονική καθυστέρηση. Όσο αυξάνει το  $n$  αυξάνει και ο χρόνος επεξεργασίας του σήματος εισόδου για να υπολογιστεί η έξοδος και εισάγεται μεγαλύτερη χρονική καθυστέρηση. Όσο μειώνεται το  $n$  τόσο μικραίνει το σφάλμα στην έξοδο αλλά αυξάνεται η ευαισθησία της εξόδου σε τυχαίες διακυμάνσεις της εισόδου. Ο λόγος SNR (signal to noise ratio) μπορεί να βελτιωθεί περαιτέρω με πολλαπλά περάσματα του φίλτρου.

#### 4.2.1.2 Φίλτρο εκθετικής εξομάλυνσης



Η συνάρτηση μεταφοράς στο πεδίο του χρόνου για φίλτρο 1<sup>ης</sup> τάξης είναι:

$$\frac{Y(s)}{R(s)} = \frac{1}{\tau s + 1} \Rightarrow (\tau s + 1)Y(s) = R(s) \xrightarrow{\mathcal{L}^{-1}} \tau \dot{y}(t) + y(t) = r(t)$$

Η διακριτοποίηση της προηγούμενη σχέσης δίνει

$$\begin{aligned} \tau \frac{y_k - y_{k-1}}{T} + y_k &= r_k \Rightarrow \\ \left(\frac{\tau}{T} + 1\right) y_k &= \frac{\tau}{T} y_{k-1} + r_k \Rightarrow \\ y_k &= \frac{\tau}{T + \tau} y_{k-1} + \frac{T}{T + \tau} r_k \end{aligned}$$

θέτουμε

$$a = \frac{T}{T + \tau}$$

οπότε προκύπτει

$$y_k = (1 - a)y_{k-1} + ar_k \Rightarrow$$

$$\boxed{y_k = y_{k-1} + a(r_k - y_{k-1})}$$

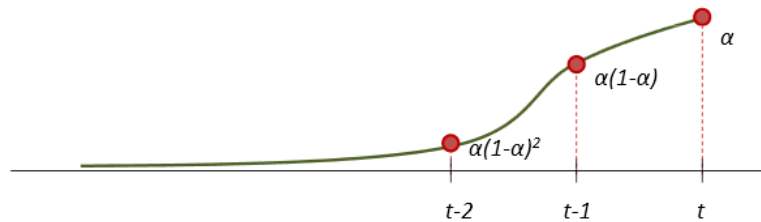
όπου  $T$  ο χρόνος δειγματοληψίας και  $0 < a < 1$ .



Όσο μεγαλύτερη τιμή επιλεχθεί για το  $\alpha$  τόσο μεγαλύτερη εμπιστοσύνη δείχνουμε στην τελευταία μέτρηση. Αν το  $\alpha \cong 1$  ουσιαστικά μόνο οι πολύ πρόσφατες μετρήσεις χρησιμοποιούνται. Με επαναλαμβανόμενη εφαρμογή της τελευταίας σχέσης προκύπτει ότι έχουμε ένα σταθμισμένο<sup>3</sup> (weighted) μέσο όρο με βάρη  $\alpha(1 - \alpha)^i$  όπου  $i = 0, 1, \dots, n - 1$ . Η τιμή της εξόδου εκτιμάται από το σταθμισμένο άθροισμα των προηγούμενων μετρήσεων.

$$y_k = (1 - \alpha)^n y_{k-n} + \alpha \sum_{i=0}^{n-1} (1 - \alpha)^i r_{k-i}$$

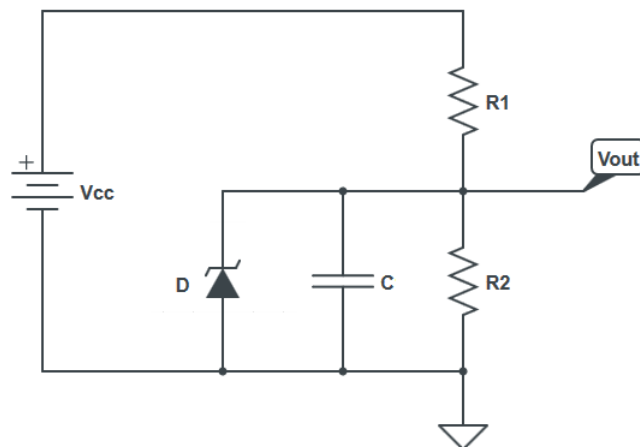
Ο συντελεστής βαρύτητας των παλαιότερων τιμών των δεδομένων μειώνεται εκθετικά και η επιλογή του  $\alpha$  καθορίζει το πόσο γρήγορα φθίνουν.



Εικόνα 4.6 Εκθετική ομαλοποίηση

#### 4.2.2 Διαιρέτης τάσης

Για την εκτίμηση του επιπέδου της τάσης του συσσωρευτή σε κάθε κύκλο ελέγχου σε τμήμα του τυπωμένου κυκλώματος του ρυθμιστή τάσης των 12V κατασκευάστηκε ο διαιρέτης τάσης της εικόνας 4.7.



Εικόνα 4.7 Διαιρέτης τάσης για την μέτρηση της τάσης του συσσωρευτή (LiPo) του rover

Ο κεραμικός πυκνωτής 100nF λειτουργεί ως βραχυκύκλωμα για τις υψηλές συχνότητες και η διόδος Zener (BZX79C5V1) αποτελεί ασφάλεια τάσης με δυναμικό κατάρρευσης τα 5.1V κατά την ανάστροφη πόλωση της.

<sup>3</sup> Σταθμισμένος μέσος όρος ονομάζεται το αποτέλεσμα:

$\bar{x} = \frac{\sum_{i=1}^k (w_i x_i)}{\sum_{i=1}^k w_i}$  όπου  $w_i$  ο συντελεστής βαρύτητας του  $i$  στοιχείου  $x_i$

Η τάση εξόδου του διαιρέτη είναι ίση με

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{cc} \quad (1)$$

Οι τιμές των αντιστάσεων που επιλέχθηκαν είναι  $R_1 = 1M\Omega$  και  $R_2 = 100K\Omega$  συνεπώς

$$V_{out} = \frac{1}{11} V_{cc}$$

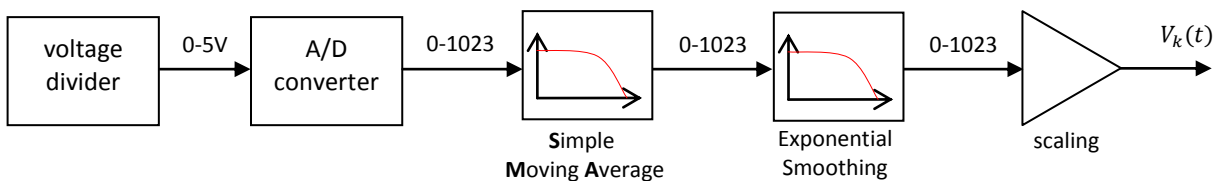
Η αναλογική έξοδος του διαιρέτη οδηγείται στην A0 είσοδο του 10bit ADC του Arduino Mega2560. Κριτήριο για την επιλογή των αντιστάσεων σε σχέση με την τάση του συσσωρευτή αποτελεί το ότι η μέγιστη τιμή της τάσης εξόδου του διαιρέτη δεν θα πρέπει να υπερβαίνει τα 5V ώστε να προστατευτεί το board από μια πιθανή καταστροφή. Ο συγκεκριμένος διαιρέτης είναι κατάλληλος για την μέτρηση dc τάσεων έως και 55V. Η τάση αναφοράς του ADC είναι 5V (ακροδέκτης AREF) οπότε η διακριτική ικανότητα θα είναι  $5V/1024 = 0.004883$ . Η ψηφιακή έξοδος του A/D μετατροπέα αποτελεί έναν ακέραιο αριθμό  $X$  από 0 έως και  $2^{10}-1$  ο οποίος εκφράζει το σχετικό μέγεθος της στάθμης του σήματος εισόδου του (έξοδο διαιρέτη τάσης) ως προς την τάση αναφοράς  $V_{ref}$ . Συγκεκριμένα θα ισχύει:

$$X = 2^{10} \frac{V_{out}}{V_{ref}} \Rightarrow V_{out} = \frac{X}{1024} V_{ref} \quad (2)$$

Από τις σχέσεις (1) και (2) προκύπτει ότι

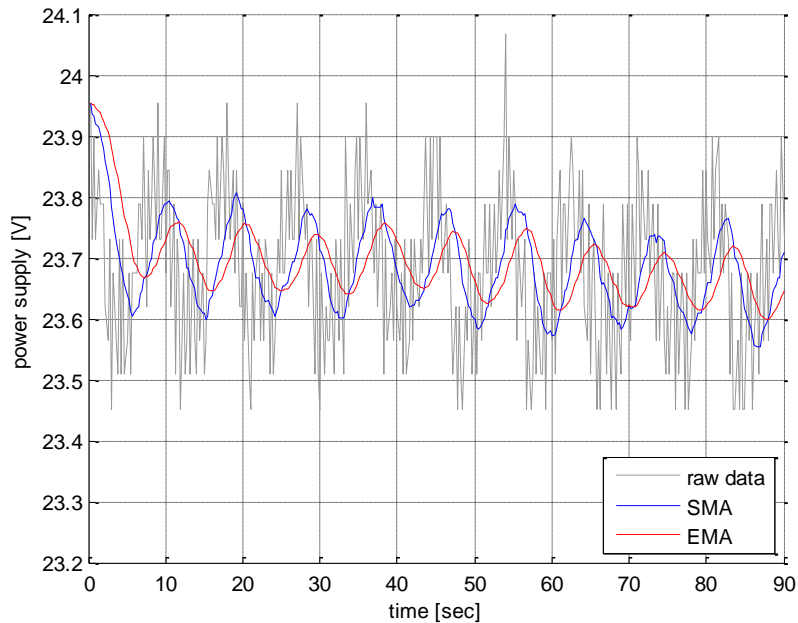
$$V_{cc} = \frac{X(R_1 + R_2)}{1024R_2} V_{ref}$$

Η διακριτοποιημένη τιμή της τάσης εξομαλύνεται για τον περιορισμό του θορύβου μέσω δυο φίλτρων κινούμενου μέσου όρου τα οποία υλοποιούνται ψηφιακά στον κώδικα της εφαρμογής. Η εξομαλυμένη τιμή διαβαθμίζεται σύμφωνα με την προηγούμενη σχέση. Η δειγματοληψία της τάσης εξόδου του διαιρέτη τάσης λαμβάνει χώρα κάθε 0.25 sec. Το απλό φίλτρο κινούμενου μέσου όρου είναι 15 δειγμάτων και το εκθετικά τροποποιούμενο έχει  $a = 0.125$ .



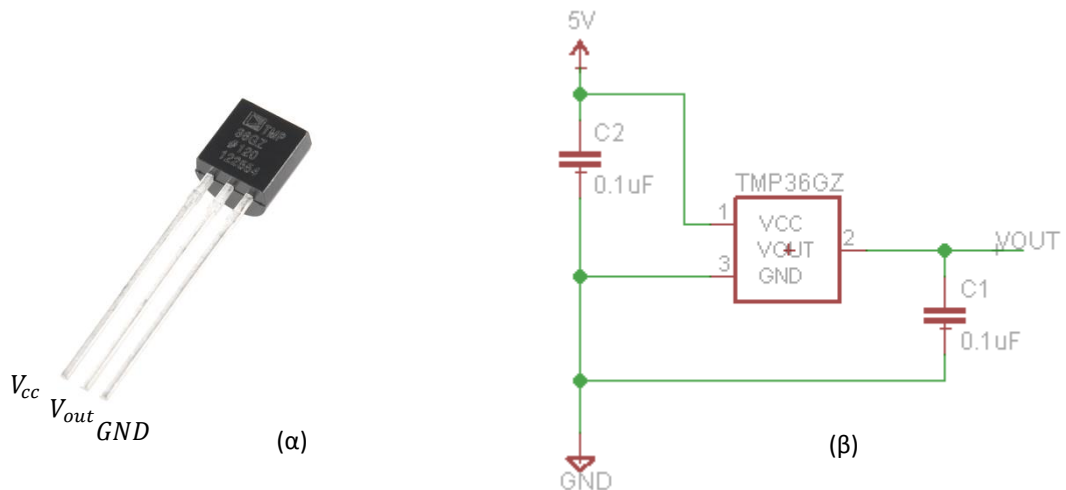
**Εικόνα 4.8** Διάγραμμα βαθμίδων για την διακριτοποίηση και εξομάλυνση της μετρούμενης τάσης του συσσωρευτή

Για τον έλεγχο της διακύμανσης της τάσης τροφοδοσίας τέθηκε περιοδικά επαναλαμβανόμενο τραπεζοειδές προφίλ ταχύτητας (κάθε 9 sec) ως σήμα αναφοράς με μέγιστη τιμή στο duty cycle 433/511 και στους 4 τροχούς σε έλεγχο κλειστού βρόγχου. Πηγή τροφοδοσίας αποτέλεσε η μπαταρία LiPo.



Εικόνα 4.9 Μετρούμενη τάση τροφοδοσίας κατά την εκτέλεση επαναλαμβανόμενου τραπεζοειδούς προφίλ ταχύτητας

### 4.2.3 Αισθητήρας θερμοκρασίας TMP36



Εικόνα 4.10 α) Αισθητήρας θερμοκρασίας TMP36GZ β) συνδεσμολογία

Το αισθητήριο θερμοκρασίας TMP36 παρουσιάζει ευαισθησία  $10\text{mV}/^{\circ}\text{C}$ , διαθέτει μια αναλογική έξοδο η οποία μεταβάλλεται γραμμικά με την αλλαγή της θερμοκρασίας και προκειμένου να δίνεται η δυνατότητα μέτρησης αρνητικών θερμοκρασιών λειτουργεί με μια αντιστάθμιση (offset) των  $500\text{mV}$ . Παρέχει ακρίβεια  $\pm 1^{\circ}\text{C}$  στην περιοχή των  $25^{\circ}\text{C}$  και  $\pm 2^{\circ}\text{C}$  στο εύρος λειτουργίας του από  $-40^{\circ}\text{C}$  έως  $+125^{\circ}\text{C}$ . Τροφοδοτείται με τάση από  $2.7\text{V}$  έως και  $5.5\text{V}$  μέγιστη. Στην θερμοκρασία των  $25^{\circ}\text{C}$  η τάση στον ακροδέκτη εξόδου είναι  $0.75\text{V}$ . Η τάση εξόδου του αισθητηρίου μετατρέπεται σε θερμοκρασία σε  $^{\circ}\text{C}$  σύμφωνα με την σχέση

$$\theta = \frac{V_{out} - 500\text{mV}}{10\text{mV}} \text{ } ^{\circ}\text{C} \rightarrow \theta = (V_{out} - 0.5) * 100 \text{ } ^{\circ}\text{C}$$

Η αναλογική έξοδος του αισθητηρίου οδηγείται στην είσοδο **A4** του 10bit A/D μετατροπέα του Arduino Mega2560. Αν ADC είναι η διακριτοποιημένη τιμή της τάσης εξόδου του αισθητηρίου και  $V_{ref} = 5V$  η τάση αναφοράς του A/D μετατροπέα θα είναι:

$$V_{out} = \frac{ADC}{1024} V_{ref}$$

και η θερμοκρασία:

$$\theta = \left( \frac{ADC}{1024} V_{ref} - 0.5 \right) * 100 \text{ } ^\circ C$$

Η διακριτοποιημένη τιμή της τάσης εξομαλύνεται μέσω χαμηλοπερατού φίλτρου κινούμενου μέσου όρου 20 δειγμάτων. Η δειγματοληψία πραγματοποιείται κάθε 1 sec.

Λόγω του ότι ο αισθητήρας καταναλώνει ρεύμα μικρότερο από 50mA ώστε να διατηρεί χαμηλά τα επίπεδα θερμότητας που αυτός παράγει (χαμηλότερα από 0.1 °C σε συνθήκες νηνεμίας) όταν βρίσκεται εκτεθειμένος σε ηλεκτρικά αφιλόξενο περιβάλλον είναι επιρρεπής σε ηλεκτρομαγνητικές παρεμβολές (**Radio Frequency Interference**). Η επίδραση των RFI στον αισθητήρα εκδηλώνεται ως μια μη φυσιολογική μετατόπιση του επιπέδου της dc τάσης εξόδου λόγω της ανόρθωσης των υψίσυχνων ρευμάτων. Για την ελαχιστοποίηση των επιδράσεων αυτών είναι απαραίτητη η σύνδεση κεραμικών πυκνωτών 0.1μF μεταξύ τάσης τροφοδοσίας-γείωσης και της αναλογικής εξόδου-γείωσης.

#### 4.2.3.1 Οδήγηση ανεμιστήρα

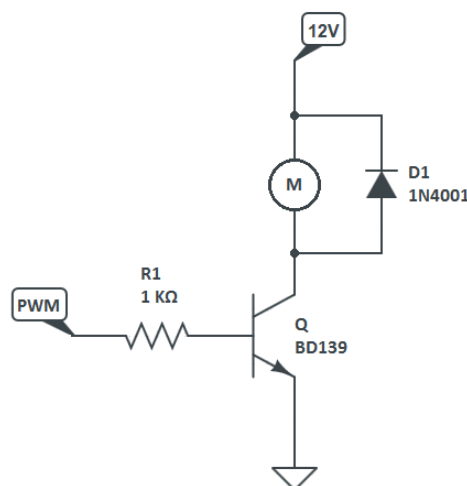
Για την απαγωγή της θερμότητας που παράγεται από τα ηλεκτρονικά κυκλώματα ισχύος χρησιμοποιήθηκε ανεμιστήρας της εικόνας 4.11 ο οποίος χρησιμοποιεί σύγχρονο ηλεκτροκινητήρα συνεχούς ρεύματος χωρίς ψήκτρες.



Model	FD1212-A3142E
Bearing	Sleeve
Weight [g]	120
Frame Dimension [mm]	120x120
Frame Thickness [mm]	25
Rated Voltage [V]	12
Rated Current [A]	0.31
Input Power [W]	3.72
Nominal Speed [rpm]	2100
Airflow [cfm]	73.7
Static Pressure [inch-H <sub>2</sub> O]	0.12
Noise Level [dBA]	38

**Εικόνα 4.11** Ο ανεμιστήρας FD1212-A3142E της εταιρείας ARX και τα τεχνικά του χαρακτηριστικά

Ο έλεγχος της παρεχόμενης ισχύος στον ανεμιστήρα από τον μικροελεγκτή ATmega2560 υλοποιείται μέσω κατάλληλα επιλεγμένου τρανζίστορ. Το διπολικό τρανζίστορ NPN (BD139) χρησιμοποιείται ως ημιαγωγικός διακόπτης και λειτουργεί είτε στην περιοχή αποκοπής είτε στην περιοχή κορεσμού με την αντίσταση στη βάση του να είναι 1KΩ. Η βάση του BJT οδηγείται με PWM κυματομορφή για τον έλεγχο στροφών του κινητήρα. Η είσοδος χρησιμοποιείται για την ομαλή σβέση του πηνίου του κινητήρα κατά την περίοδο off της PWM κυματομορφής. Μειονέκτημα αυτής της συνδεσμολογίας αποτελεί η κοινή γείωση ανάμεσα στο ψηφιακό σύστημα και τον κινητήρα και η έστω μικρή τάση κορεσμού συλλέκτη εκπομπού καθώς εκεί υπάρχουν απώλειες ισχύος.



Εικόνα 4.12 Κυκλωματικό διάγραμμα οδήγησης ανεμιστήρα με BJT

Η μέση στάθμη της τάσης του σήματος PWM είναι

$$V_{pwm} = \frac{V_H T_{on} + V_L T_{off}}{T_{pwm}}$$

Είναι  $V_L = 0V$  και  $V_H = 12V$

$$V_{pwm} = \frac{T_{on}}{T_{pwm}} 12 V$$

Χρησιμοποιείται η αυτοματοποιημένη παραγωγή σήματος PWM στον ακροδέκτη 6 (OC4A) του μικροελεγκτή από τον Timer4 μέσω της λειτουργίας Phase and Frequency Correct PWM mode 8. Λόγω του ότι η επιθυμητή τιμή της  $F_{pwm}$  δεν είναι άμεσα διαθέσιμη ο καθορισμός της μέγιστης τιμής μέτρησης TOP δίνεται μέσω του καταχωρητή ICR4(= 53). Η διακριτική ικανότητα του duty cycle είναι  $1/53*100 = 1.887\%$  Η συχνότητα του παραγόμενου PWM σήματος καθορίζεται από την επιλεγμένη τιμή  $N(=8)$  του prescaler και την συχνότητα ρολογιού του μικροελεγκτή  $F_{ck} = 16MHz$

$$F_{pwm} = \frac{F_{ck}}{2 * N * TOP} = 18.868 KHz$$

Η ελάχιστη τιμή του duty cycle για την περιστροφή της φτερωτής βρέθηκε πειραματικά ίση με 11/53. Ο έλεγχος που υλοποιείται για τη ρύθμιση της ταχύτητας περιστροφής του κινητήρα είναι ανοιχτού βρόγχου. Η ταχύτητα περιστροφής καθορίζεται από την μέτρηση της θερμοκρασίας στο εσωτερικό του κουτιού των ηλεκτρονικών ελέγχου μέσω του αισθητήρα TMP36GZ. Ο ανεμιστήρας ενεργοποιείται στην θερμοκρασία των 20° C και η μέγιστη ταχύτητα περιστροφής επιβάλλεται στους 40° C.

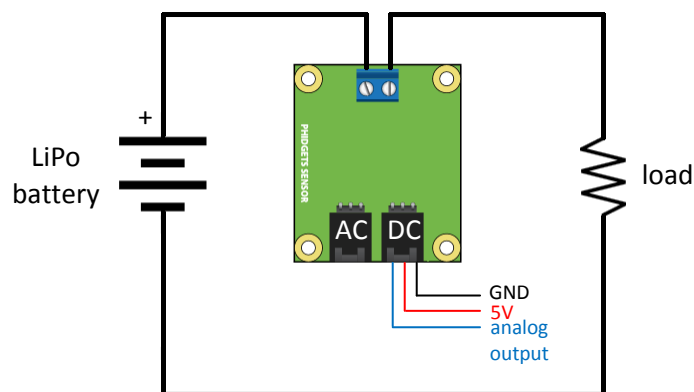
#### 4.2.4 Αισθητήριο ρεύματος

Στον ηλεκτρονικό εξοπλισμό περιλαμβάνεται και ένα αισθητήριο ρεύματος το οποίο χρησιμοποιείται για τη μέτρηση και την καταγραφή του συνολικού καταναλισκόμενου ρεύματος και κατ' επέκταση της αναλωμένης ισχύος από το όχημα κατά την διάρκεια των ελιγμών του.



Εικόνα 4.13 Το αισθητήριο ρεύματος της Phidgets με p/n 1119

Η διάταξη προέρχεται από την εταιρία Phidgets και αποτελεί μια υλοποίηση που βασίζεται στο ολοκληρωμένο ACS706 της εταιρείας Allegro η λειτουργία του οποίου στηρίζεται στο φαινόμενο Hall. Το εύρος μέτρησης της έντασης ρεύματος του αισθητηρίου στο συνεχές οριοθετείται από -15A έως +15A με την αντίστοιχη τάση εξόδου του να κυμαίνεται από 0 έως 5V. Στα 0A αντιστοιχεί τιμή τάσης εξόδου 2.5V. Η ευαισθησία του ACS706 είναι της τάξης των 100mV/A με το τυπικό σφάλμα της διάταξης να κυμαίνεται στο  $\pm 1.5\%$ . Το αισθητήριο τοποθετήθηκε σε σειρά στη γραμμή τροφοδοσίας όπως εικονίζεται στο κάτωθι διάγραμμα.

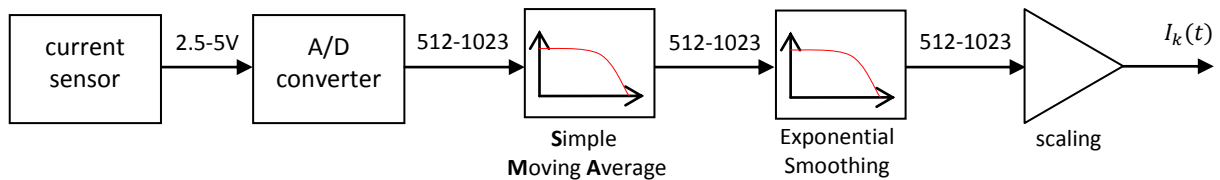


Εικόνα 4.14 Διαγραμματική αναπαράσταση του τρόπου σύνδεσης του αισθητηρίου ρεύματος

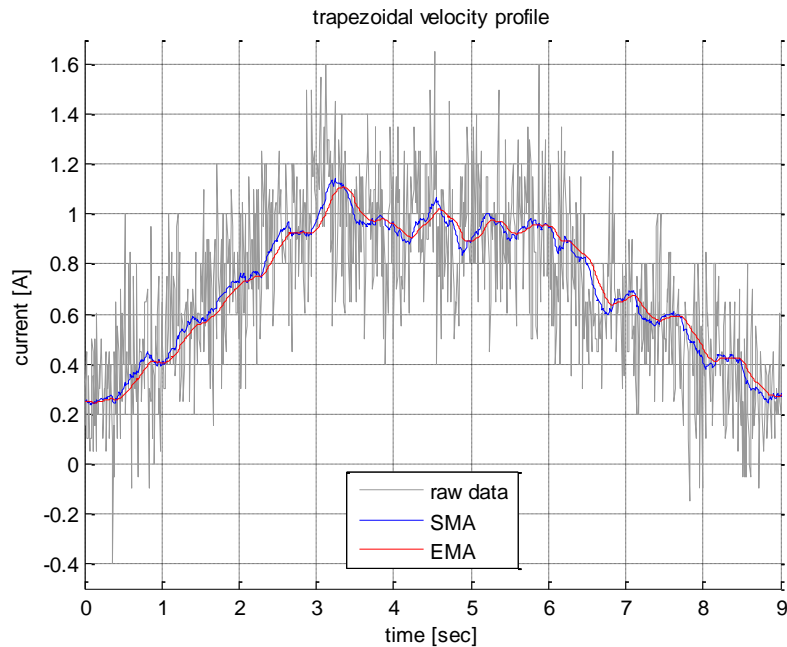
Η αναλογική έξοδος του αισθητηρίου οδηγείται στην είσοδο A3 του 10bit ADC του Arduino Mega2560. Το αναλογικό σήμα διαμερίζεται σε  $2^{10} = 1024$  διακριτές στάθμες. Για τον περιορισμό του θορύβου το σήμα από τον ADC εξομαλύνεται μέσω δύο φίλτρων κινούμενου μέσου όρου (moving average) τα οποία υλοποιούνται ψηφιακά στον κώδικα της εφαρμογής.

Η είσοδος του πρώτου φίλτρου δέχεται εύρος τιμών από 512 έως και 1023 από την διακριτοποίηση του αναλογικού σήματος της τάσης με εύρος 2.5V έως και 5V αντίστοιχα. Η έξοδος του πρώτου φίλτρου αποτελεί είσοδο για το δεύτερο. Η έξοδος του δεύτερου διαβαθμίζεται σύμφωνα με την αναγραφόμενη στο συνοδευτικό φυλλάδιο της διάταξης κάτωθι σχέση και παράγεται η ένταση του ρεύματος σε Amperes σε κάθε κύκλο ελέγχου (50Hz).

$$I = \frac{\text{smoothed sensor value}}{20} - 25$$



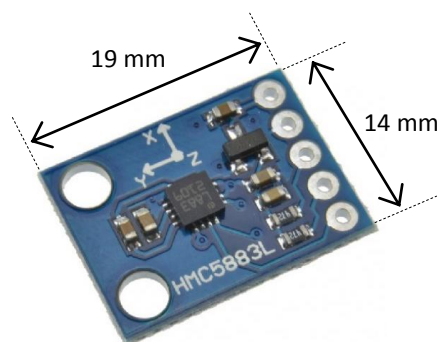
Εικόνα 4.15 Διάγραμμα βαθμίδων για την διακριτοποίηση και εξομάλυνσης της τάσης εξόδου του αισθητηρίου ρεύματος



Εικόνα 4.16 Καταγραφή του καταναλισκόμενου ρεύματος κατά την εκτέλεση τραπεζοειδούς προφίλ ταχύτητας με μέγιστη τιμή στο duty cycle 255/511.

## 4.2.5 Αισθητήρια για τον υπολογισμό του προσανατολισμού

### 4.2.5.1 Μαγνητόμετρο 3 αξόνων



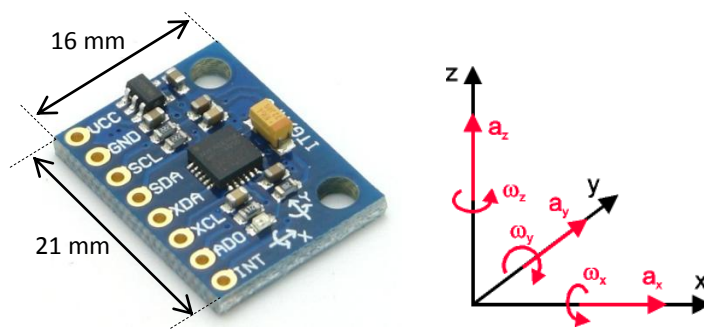
Εικόνα 4.17 Η ψηφιακή πυξίδα GY-273

Η ψηφιακή πυξίδα 3 αξόνων βασίζεται στο ολοκληρωμένο κύκλωμα HMC5883L της Honeywell. Τα βασικά χαρακτηριστικά του αισθητήρα είναι:

- επικοινωνία μέσω I<sup>2</sup>C
- 2.16 - 3.6V τάση τροφοδοσίας
- χαμηλή κατανάλωση ρεύματος (100μΑ)
- 12 bit ADC

- διακριτικότητα 2 mill-gauss
- εύρος  $\pm 8$  Gauss
- 160Hz ρυθμός δεδομένων εξόδου
- ακρίβεια  $1^\circ$  έως  $2^\circ$  για την γωνία εκτροπής (heading)
- λειτουργίες αντιστάθμισης και βαθμονόμησης

#### 4.2.5.2 Επιταχυνσιόμετρο & Γυροσκόπιο 3 αξόνων



Εικόνα 4.18 Η IMU GY - 521 MPU 6050

Η πλακέτα GY-521 βασίζεται στην MPU 6050 της InvenSense η οποία περιέχει ένα MEMS γυροσκόπιο 3 αξόνων και ένα MEMS επιταχυνσιόμετρο επίσης 3 αξόνων. Τροφοδοτείται με 3.3V και η επικοινωνία με το αισθητήριο επιτυγχάνεται μέσω του διαύλου I<sup>2</sup>C. Κάποια από τα βασικά τεχνικά χαρακτηριστικά του γυροσκοπίου είναι:

- διαθέσιμες επιλογές μέτρησης με εύρος:  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ ,  $\pm 2000$  °/s
- διαθέτει 16-bit ADC για κάθε κανάλι με ταυτόχρονη δειγματοληψία των 3 γυροσκοπίων κάθε άξονα
- σταθερότητα στην αλλαγή θερμοκρασίας μειώνοντας έτσι την ανάγκη για περαιτέρω βαθμονόμηση
- βελτιωμένη απόδοση στις χαμηλές συχνότητες θορύβου
- προγραμματιζόμενο χαμηλοπερατό φίλτρο
- ρεύμα κανονικής λειτουργίας: 3.6mA, σε αναμονή: 5μA
- δυνατότητα αυτοδιαγνωστικού έλεγχου

και του επιταχυνσιόμετρου:

- προγραμματιζόμενες διαθέσιμες επιλογές μέτρησης με εύρος  $\pm 2$ ,  $\pm 4$ ,  $\pm 8$ ,  $\pm 16g$
- διαθέτει 16-bit ADC για κάθε κανάλι με ταυτόχρονη δειγματοληψία των 3 επιταχυνσιομέτρων κάθε άξονα
- ρεύμα κανονικής λειτουργίας: 500μA
- ρυθμιζόμενη κατανάλωση ρεύματος σε σχέση με την συχνότητα δειγματοληψίας
- προγραμματιζόμενες διακοπές
- δυνατότητα αυτοδιαγνωστικού έλεγχου



## Κεφάλαιο

# 5

## 5. Έλεγχος κίνησης τροχών

Στο παρόν κεφάλαιο αρχικά

- παρουσιάζονται τα τεχνικά χαρακτηριστικά του χρησιμοποιούμενου σερβομηχανισμού για την κίνηση κάθε τροχού του οχήματος και εν συνεχεία χαράσσονται οι χαρακτηριστικές ροπής του
- υπολογίζεται η συνάρτηση μεταφοράς του κινητήριου συστήματος του οποίου φορτίο θεωρείται ο προσαρτημένος στον άξονα του τροχός με αυτόν να περιστρέφεται στον αέρα και να μην ακουμπά στο έδαφος
- περιγράφεται η αρχή λειτουργίας του ενσωματωμένου κωδικοποιητή θέσης και δίνεται ο αλγόριθμος αποκωδικοποίησης των σημάτων θέσης
- παρουσιάζεται η Η-γέφυρα για την οδήγηση του κινητήρα και ο τρόπος παραγωγής του σήματος PWM για την ρύθμιση της ταχύτητας περιστροφής του
- αναλύεται η μέθοδος εκτίμησης της γωνιακής ταχύτητας με το φίλτρο Savitzky-Golay

Ακολουθώντας με την βοήθεια της αναπτυξιακής πλακέτας Arduino Uno και για την πειραματική ταυτοποίηση του μοντέλου εκτελούνται μια σειρά από πειράματα για το σύστημα ανοικτού βρόγχου κατά τα οποία μεταβάλλεται η είσοδος (εφαρμοζόμενη τάση) και καταγράφεται η απόκριση του συστήματος (γωνιακή ταχύτητα) οπότε προσδιορίζεται για κάθε ένα σύστημα κινητήρα-τροχού:

- ένα μοντέλο το οποίο περιγράφει με ικανοποιητική προσέγγιση τα πειραματικά δεδομένα
- η μέγιστη ταχύτητα περιστροφής
- τα όρια της νεκρής ζώνης

Εν συνεχεία για την σύγκριση της εξόδου του εκτιμωμένου μοντέλου με την έξοδο του πραγματικού παρατίθενται σειρά γραφικών παραστάσεων.

Για τον έλεγχο του συστήματος κλειστού βρόγχου με PID αντιστάθμιση διακρίτοποιείται ο ελεγκτής και ρυθμίζονται τα κέρδη α) με την μέθοδο των Ziegler- Nichols και β) με την μέθοδο της άμεσης σύνθεσης.

Στις επόμενες ενότητες παρουσιάζονται:

- το τυπωμένο κύκλωμα ελέγχου της ταχύτητας περιστροφής των τροχών το οποίο περιλαμβάνει τέσσερις μικροελεγκτές ATmega328 έναν για κάθε κινητήρα με εγκατεστημένο σε αυτούς το αντίστοιχο firmware για τον έλεγχο του κλειστού βρόγχου κάθε συστήματος κινητήρα-τροχού
- ο δίαυλος I<sup>2</sup>C που χρησιμοποιείται για την μεταφορά δεδομένων μεταξύ των μικροελεγκτών ATmega328 και της αναπτυξιακής πλακέτας Arduino Mega 2560 και το ρεπερτόριο εντολών της βιβλιοθήκης Wire που υλοποιεί το συγκεκριμένο πρωτόκολλο

- οι χρησιμοποιούμενοι ακροδέκτες στην αναπτυξιακή πλακέτα Arduino Mega 2560
- τα λογικά διαγράμματα των εγκατεστημένων firmware στους μικροελεγκτές ATmega328 και ATmega2560
- η χρησιμοποιηθείσα εφαρμογή για την εκτύπωση γραφημάτων σε πραγματικό χρόνο

Τέλος παρατίθενται τα αποτελέσματα από την διεξαγωγή πειραμάτων του κινούμενου οχήματος για την αποτίμηση των στρατηγικών ελέγχου που μελετήθηκαν.

## 5.1 Ηλεκτρικός κινητήρας EMG49



Εικόνα 5.1 Ο dc ηλεκτροκινητήρας EMG49

Ο ηλεκτροκινητήρας EMG49 αποτελεί μια ολοκληρωμένη διάταξη ελέγχου κίνησης. Είναι κινητήρας συνεχούς ρεύματος μόνιμου μαγνήτη με ψήκτρες και ενσωματωμένο πλανητικό μειωτήρα στροφών με λόγο μείωσης 49:1. Στο πίσω μέρος του είναι προσαρτημένος κωδικοποιητής θέσης αυξητικού τύπου για την ανάδραση πληροφορίας θέσης. Παρατίθενται τα τεχνικά χαρακτηριστικά του κινητήριου συστήματος σύμφωνα με τα αναγραφόμενα δεδομένα στο datasheet.

Τεχνικά χαρακτηριστικά κινητήρα  
χωρίς μειωτήρα στροφών

Nominal voltage	24V			
Nominal torque	16 kg*cm	1.57 Nm	570 gf*cm	0.056 Nm
Nominal speed	122 rpm	12.78 rad/sec	5900 rpm	617.85 rad/sec
Nominal current	2100mA			
No load speed	143 rpm	14.97 rad/sec	7000 rpm	733.04 rad/sec
No load current	≤ 500mA			
Stall current	13A			
Rated output	34.7W			
Encoder counts per output shaft turn	980			
Reduction ratio	49			
Gearbox efficiency	60%			

*Nominal voltage* (ονομαστική τάση): Η συνιστώμενη μέγιστη τάση που μπορεί να εφαρμοστεί στο τύμπανο του κινητήρα. Η χαρακτηριστική ταχύτητας-ροπής χαράσσεται με αυτή την τάση. Ο κινητήρας μπορεί να τροφοδοτηθεί με λιγότερη ή περισσότερη τάση. Στην εφαρμογή υψηλότερης απαιτείται προσοχή ώστε να αποτραπεί η υπερθέρμανση των τυλιγμάτων του τυμπάνου λόγω των ωμικών αντιστάσεων.

*Nominal speed*: Η ταχύτητα του κινητήρα στην μέγιστη συνεχή ροπή λειτουργίας υπό ονομαστική τάση.

*Nominal torque*: Η μέγιστη συνεχής ροπή ασφαλούς λειτουργίας που μπορεί να παρέχει ο κινητήρας υπό ονομαστική τάση χωρίς να υπερθερμανθεί και απότομη βύθιση της ταχύτητας του.

*Nominal current*: Το ρεύμα που παράγει τη μέγιστη συνεχή ροπή.

*No load speed*  $n_0$  (rpm): Η ταχύτητα που αναπτύσσει ο κινητήρας τροφοδοτούμενος με την ονομαστική τάση παράγοντας ελάχιστη ροπή.

*No load current*  $i_0$ : Το ρεύμα που απαιτείται για να περιστραφεί ο κινητήρας χωρίς φορτίο και να παραχθεί η απαιτούμενη ροπή ώστε να υπερνικηθούν οι τριβές.

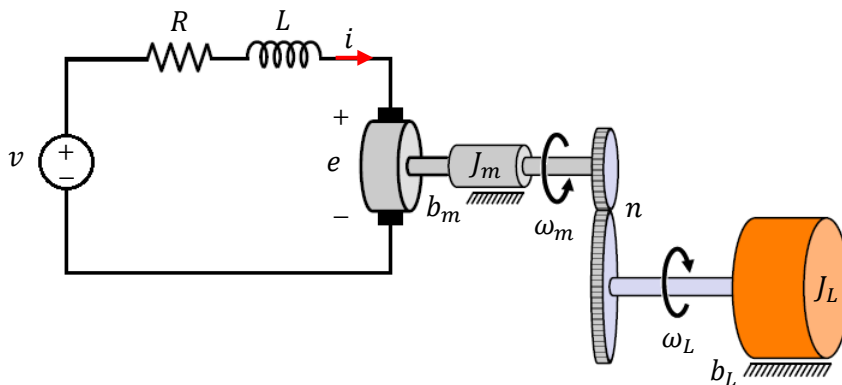
*Stall torque* (ροπή απώλειας ταχύτητας)  $\tau_s$ : Η μέγιστη δυνατή ροπή υπό την ονομαστική τάση που μπορεί να αποδώσει ο κινητήρας όταν υπερφορτιστεί ώστε να "φρενάρει".

*Stall current*: Η μέγιστη τιμή του ρεύματος όταν ο κινητήρας υπερφορτιστεί και φρενάρει.

*Torque (or motor) constant*  $K_t$  ή  $K_M$ : Η σταθερά αναλογίας της παραγόμενης ροπής ως προς το ρεύμα. Μονάδα μέτρησης στο SI [Nm/A].

*Speed constant*  $K_n$ : Η σταθερά αναλογίας της ταχύτητας περιστροφής ως προς την εφαρμοζόμενη τάση με μονάδα μέτρησης [rad/(V·sec)]. Είναι αριθμητικά ίση με το αντίστροφο της σταθεράς ροπής  $1/K_t$ .

## 5.2 Μοντελοποίηση κινητήριου συστήματος



Εικόνα 5.2 Ισοδύναμο κύκλωμα τυμπάνου κινητήρα dc και σύστημα μετάδοσης της κίνησης

Όταν το επαγωγικό τύμπανο του κινητήρα τροφοδοτηθεί με ρεύμα ξεκινά να στρέφεται μέσα στο μαγνητικό πεδίο αφού οι αναπτυσσόμενες στους αγωγούς δυνάμεις σχηματίζουν ζεύγος δυνάμεων, το οποίο ασκεί ροπή στρέψης. Καθώς αυτό στρέφεται γεννιέται στους αγωγούς του μια Αντieleκτρεγεργτική Δύναμη. Αυτή η ΑΗΕΔ είναι επαγωγικό δημιούργημα και σύμφωνα με τον κανόνα του Lenz αντιτίθεται προς την τάση της πηγής. Το ισοδύναμο κύκλωμα του επαγωγικού τυμπάνου ενός dc κινητήρα βασίζεται στο γεγονός ότι το τύλιγμα του τυμπάνου έχει ωμική αντίσταση  $R$ , αυτεπαγωγή  $L$  και επαγόμενη ΗΕΔ. Ο νόμος του Kirchhoff στο ηλεκτρικό υποσύστημα δίνει:

$$v = iR + L \frac{di}{dt} + e \quad (1)$$

όπου

$v$  η εφαρμοζόμενη τάση

$e$  η αντιηλεκτρεγερτική τάση

$i$  το ρεύμα στο τύμπανο του κινητήρα

Η επαγόμενη ΗΕΔ είναι ανάλογη της ταχύτητας του περιστροφής του άξονα του κινητήρα  $\omega_m$  και η σταθερά αναλογικότητας  $K_b$  είναι γνωστή ως σταθερά επαγόμενης ΗΕΔ (σε  $Vs/rad$ ).

$$e = K_b \omega_m \quad (2)$$

Η ροπή που αναπτύσσει ο κινητήρας είναι ανάλογη του ρεύματος του σπλισμού:

$$T_m = K_t i \quad (3)$$

Σύμφωνα με τον θεμελιώδη νόμο της στροφικής κίνησης το αλγεβρικό άθροισμα των ροπών που δρουν πάνω σε ένα στερεό σώμα το οποίο περιστρέφεται γύρω από σταθερό άξονα ισούται με το γινόμενο της ροπής αδράνειας  $J$  (υπολογισμένης ως προς τον άξονα περιστροφής) και της γωνιακής επιτάχυνσης του σώματος.

$$J\dot{\omega} = \sum T$$

Λόγω του μειωτήρα στροφών θα είναι  $\omega_L = \frac{\omega_m}{n}$  και για ιδανική μετάδοση ισχύος ισχύει:

$$\omega_m T_m = \omega_L T_L \Rightarrow T_L = n T_m$$

θα είναι

$$\begin{aligned} J_m \frac{d\omega_m}{dt} + \frac{1}{n} J_L \frac{d\omega_L}{dt} &= T_m - b_m \omega_m - \frac{1}{n} b_L \omega_L \\ \xrightarrow{\omega_L = \frac{\omega_m}{n}} J_m \frac{d\omega_m}{dt} + \frac{1}{n^2} J_L \frac{d\omega_m}{dt} &= T_m - b_m \omega_m - \frac{1}{n^2} b_L \omega_m \\ \Rightarrow T_m &= \left( J_m + \frac{1}{n^2} J_L \right) \frac{d\omega_m}{dt} + \left( b_m + \frac{1}{n^2} b_L \right) \omega_m \\ J &= J_m + \frac{1}{n^2} J_L \\ b &= b_m + \frac{1}{n^2} b_L \\ \xrightarrow{\quad} T_m &= J \frac{d\omega_m}{dt} + b \omega_m \quad (4) \end{aligned}$$

όπου

$J$  η ολική ροπή αδρανείας ανοιγμένη στον άξονα του κινητήρα

$J_L$  η ροπή αδρανείας του φορτίου

$J_m$  η ροπή αδράνειας του δρομέα του κινητήρα (σταθερά, μονάδα μέτρησης  $Kg \cdot m^2$ )

$\omega_L$  η γωνιακή ταχύτητα περιστροφής του φορτίου

$b_m$  ο συντελεστής τριβής του κινητήρα (σταθερά, μονάδα μέτρησης  $Nm/rad/sec$ )

$b_L$  ο συντελεστής τριβής του φορτίου

Μέσω του μετασχηματισμού Laplace οι εξισώσεις λειτουργίας του κινητήρα γίνονται:

$$(1) \stackrel{\mathcal{L}}{\Rightarrow} V(s) = RI(s) + LsI(s) + E(s) \Rightarrow I(s) = \frac{1}{R + sL} [(V(s) - E(s))]$$

$$(2) \stackrel{\mathcal{L}}{\Rightarrow} E(s) = K_b \Omega_m(s)$$

$$(3) \stackrel{\mathcal{L}}{\Rightarrow} T_m(s) = K_t I(s)$$

$$(4) \stackrel{\mathcal{L}}{\Rightarrow} T_m(s) = Js\Omega_m(s) + b\Omega_m(s) \Rightarrow \Omega_m(s) = \frac{1}{b + sj} T_m(s)$$

Μεταβολές στο φορτίο του κινητήρα μοντελοποιούνται συνήθως ως μια ροπή διαταραχής η οποία εισάγεται στην εξίσωση της περιστροφικής κίνησης.

$$J \frac{d\omega_m}{dt} = T_m - b\omega_m - T_d$$

Η συνάρτηση μεταφοράς του συστήματος είναι:

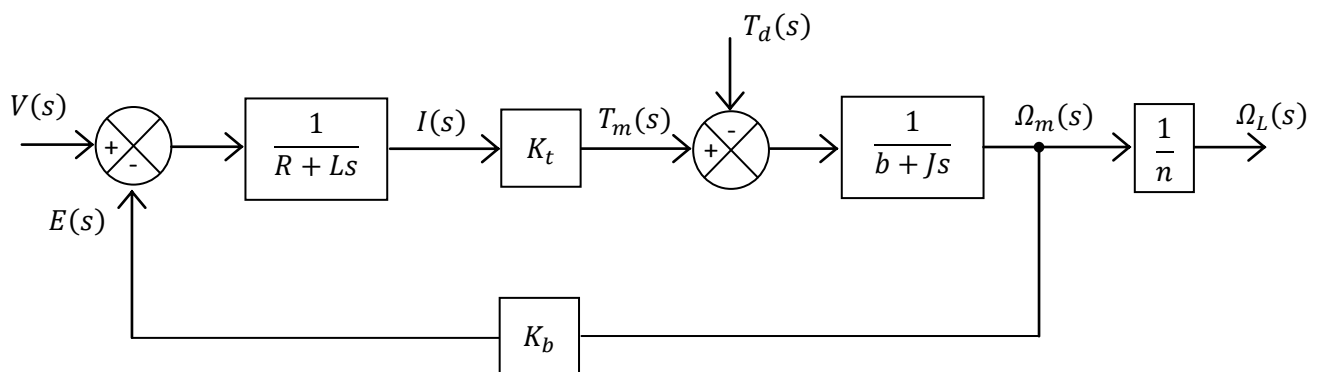
$$\frac{\Omega_m(s)}{V(s)} = \frac{\frac{K_t}{(R + Ls)(b + Js)}}{1 + \frac{K_b K_t}{(R + Ls)(b + Js)}} \Rightarrow \frac{\Omega_m(s)}{V(s)} = \frac{K_t}{(R + Ls)(b + Js) + K_b K_t} \Rightarrow$$

$$\frac{\Omega_m(s)}{V(s)} = \frac{K_t}{JLs^2 + (bL + JR)s + (bR + K_b K_t)} \Rightarrow$$

$$\frac{\Omega_m(s)}{V(s)} = \frac{\frac{K_t}{JL}}{s^2 + \left(\frac{b}{J} + \frac{R}{L}\right)s + \frac{(bR + K_b K_t)}{JL}}$$

$$\frac{\Omega_L(s)}{V(s)} = \frac{1}{n} \frac{\frac{K_t}{JL}}{s^2 + \left(\frac{b}{J} + \frac{R}{L}\right)s + \frac{(bR + K_b K_t)}{JL}}$$

Από τις προηγούμενες εξισώσεις προκύπτει και το διάγραμμα βαθμίδων



Εικόνα 5.3 Διάγραμμα βαθμίδων ηλεκτρικού κινητήρα μόνιμου μαγνήτη

### 5.3 Χάραξη χαρακτηριστικών ροπής

Η είσοδος στον κινητήρα είναι ηλεκτρική ενέργεια ενώ η έξοδος είναι μηχανική. Στη μόνιμη κατάσταση ισορροπίας το ρεύμα του τυμπάνου δεν μεταβάλλεται οπότε ο ρυθμός μεταβολής του ρεύματος τυμπάνου είναι μηδέν. Έτσι η εξίσωση (1) της τάσης απλοποιείται στην

$$v = iR + e \quad (5)$$

Για να πάρουμε το ισοζύγιο της ισχύος πολλαπλασιάζουμε την προηγούμενη εξίσωση με  $i$

$$vi = i^2R + ei$$

Ο όρος  $i^2R$  εκφράζει τις ωμικές απώλειες στο τύμπανο ενώ ο  $ei$  είναι η συνολική ισχύς εισόδου. Συνεπώς ο όρος  $ei$  εκφράζει την ωφέλιμη ισχύ που έχει μετατραπεί από η ηλεκτρική σε μηχανική. Η ισχύς αυτή εκφράζεται συναρτήσει της ηλεκτρομαγνητικής ροπής και της ταχύτητας ως

$$P = \omega_m T_m = ei \xrightarrow{(2),(3)} \omega_m K_t i = K_b \omega_m i \Rightarrow \boxed{K_b = K_t} \quad (6)$$

Συνεπώς η σταθερά αναλογίας  $K_b$  της ΑΗΕΔ και η σταθερά ροπής  $K_t$  είναι αριθμητικώς ίσες. Στην πράξη ο κινητήρας ενδέχεται να παρουσιάζει επιπρόσθετες απώλειες στη μετατροπή ενέργειας με αποτέλεσμα να ισχύει  $K_b > K_t$

Οπότε

$$(5) \xrightarrow{(2),(3)} v = \frac{T_m}{K_t} R + K_b \omega_m$$

$$\xrightarrow{(6)} \omega_m = \frac{v}{K_t} - \frac{R}{K_t^2} T_m \quad (7)$$

Για τη χάραξη της καμπύλης ταχύτητας ροπής η εφαρμοζόμενη τάση στο τύμπανο του κινητήρα είναι  $v = 24V$ . Η ταχύτητα (*no load speed*  $n_0$ ) που αναπτύσσει ο κινητήρας τροφοδοτούμενος με την τάση αυτή παράγοντας μηδενική ροπή δίνεται  $n_0 = 143 \text{ rpm}$  οπότε από την (7) προκύπτει η σταθερά  $K_b$  ΑΗΕΔ

$$K_b = \frac{24}{143 * 49 * \frac{2\pi}{60}} = 0.0327 \frac{Vsec}{rad}$$

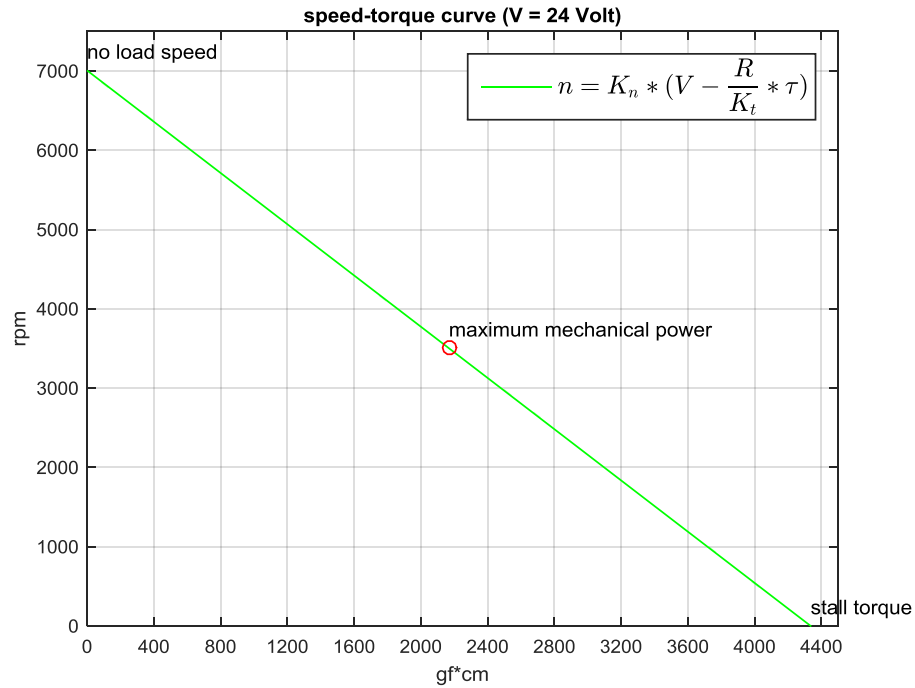
Η ωμική αντίσταση του τυμπάνου είναι ίση με το πηλίκο:

$$R = \frac{\text{rated voltage}}{\text{stall current}} = \frac{24V}{13A} = 1.8462 \Omega$$

Η μέγιστη δυνατή ροπή<sup>4</sup> stall torque (ροπή απώλειας ταχύτητας)  $\tau_s$  υπό την ονομαστική τάση που μπορεί να αποδώσει ο κινητήρας όταν υπερφορτιστεί ώστε να "φρενάρει" υπολογίζεται από την(7) ότι είναι ίση με

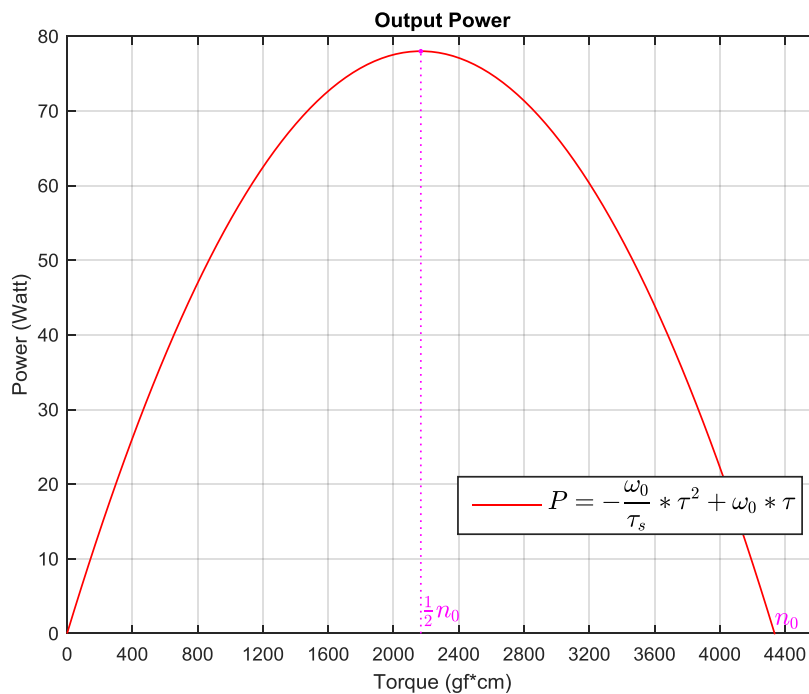
$$\tau_s = \frac{K_t v}{R} = 0.4252 \text{ Nm} = 4335.8 \text{ gf} * \text{cm}$$

<sup>4</sup>  $1 \text{ gf} = 1 \text{ gram} * 9.80665 \frac{\text{m}}{\text{sec}^2} = 0.001 \text{ Kg} * 9.80665 \frac{\text{m}}{\text{sec}^2} = 0.00980665 \text{ N} \Rightarrow 1 \text{ N} = 101.97162 \text{ gf}$   
 $1 \text{ Nm} = 101.97162 \text{ gf} * 100 \text{ cm} = 10197.16213 \text{ gf} * \text{cm}$

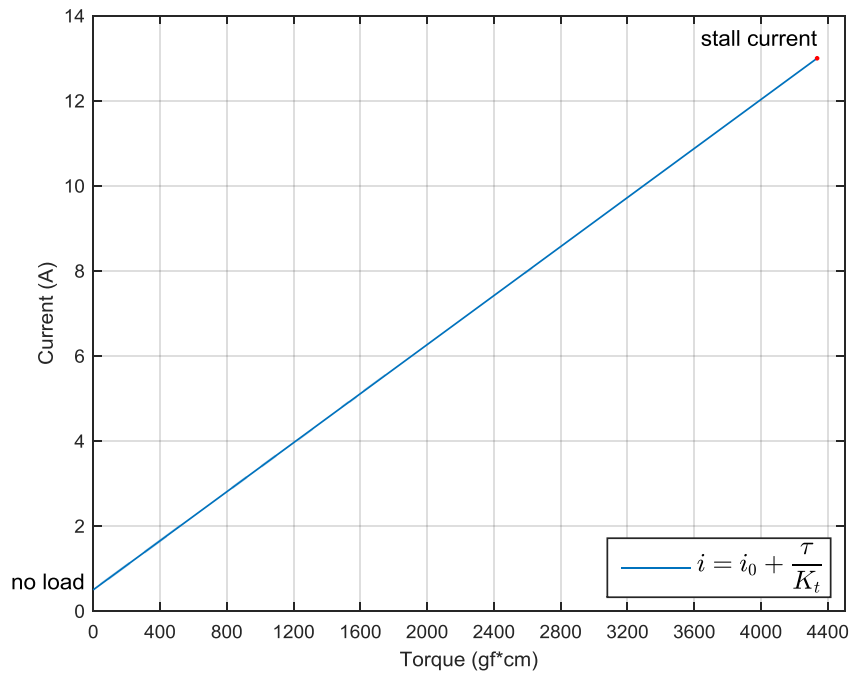


Εικόνα 5.4 Χαρακτηριστική ροπής ταχύτητας

Η ωφέλιμη ισχύς είναι  $P = T\omega \Rightarrow P = T \left( -\frac{\omega_0}{\tau_s} T + \omega_0 \right) \Rightarrow P = -\frac{\omega_0}{\tau_s} T^2 + \omega_0 T$   
 όπου  $\omega_0 = n_0 \frac{2\pi}{60} \text{ rad/sec}$



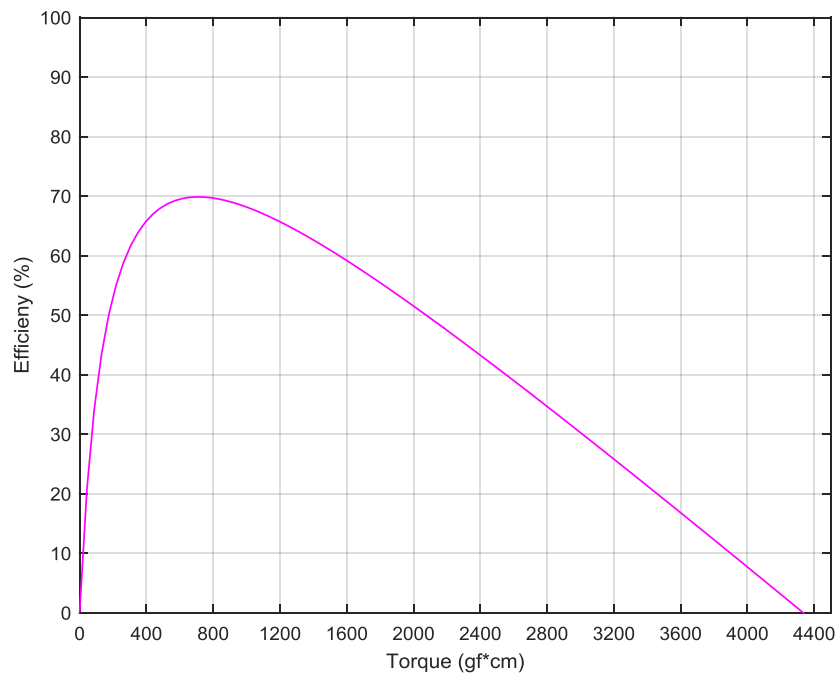
Εικόνα 5.5 Χαρακτηριστική ροπής-ωφέλιμης ισχύος



Εικόνα 5.6 Χαρακτηριστική ροπής - καταναλισκόμενου ρεύματος

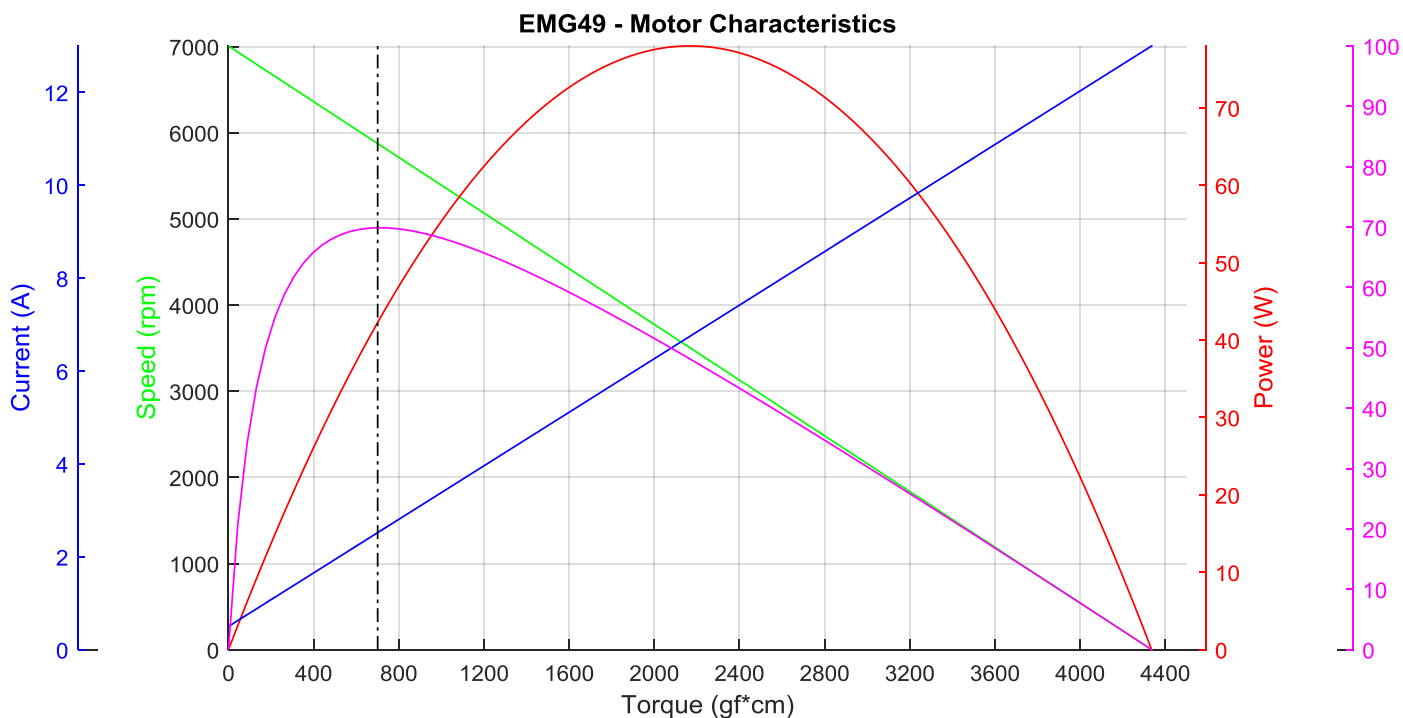
Ο βαθμός απόδοσης καλείται ο λόγος της ισχύος την οποία αποδίδει ο κινητήρας προς την ισχύ εισόδου εργαζόμενος με την ονομαστική του τάση.

$$\frac{P_{out}}{P_{in}} 100$$



Εικόνα 5.7 Χαρακτηριστική ροπής - βαθμού απόδοσης

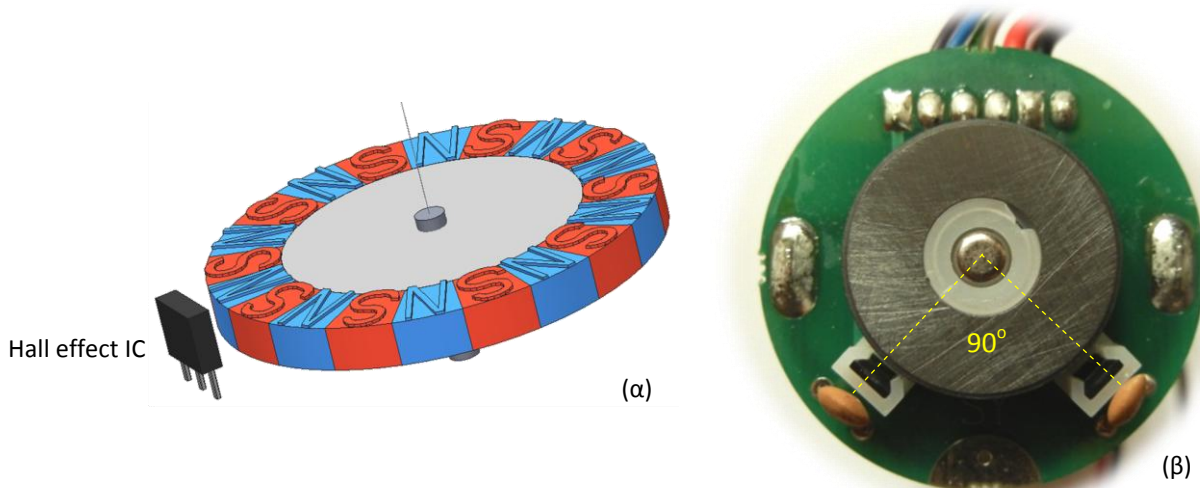




Εικόνα 5.8 Χαρακτηριστικές ροπής ηλεκτροκινητήρα EMG49

## 5.4 Περιστροφικός κωδικοποιητής αυξητικού τύπου

### 5.4.1 Εισαγωγικά στοιχεία

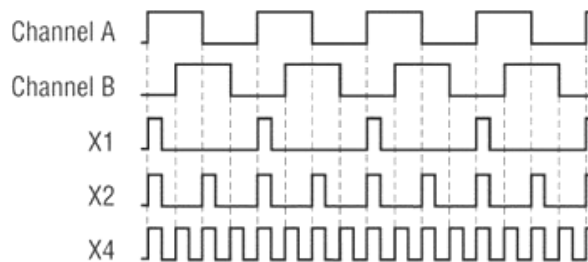


Εικόνα 5.9 α) Αρχή λειτουργίας μαγνητικού κωδικοποιητή θέσης β) Ο κωδικοποιητής θέσης του κινητήρα EMG49

Οι κωδικοποιητές αυξητικού τύπου ή σχετικής θέσης (incremental encoders) δεν μετρούν την απόλυτη θέση του στοιχείου που ελέγχουν αλλά μπορούν να καταγράψουν τις μεταβολές της. Για την μέτρηση της γωνιακής θέσης και ταχύτητας ενός ηλεκτροκινητήρα δύναται στο πίσω μέρος του να προσαρτηθεί απευθείας στον άξονα του κωδικοποιητής αυξητικού τύπου. Στους περιστροφικούς μαγνητικούς κωδικοποιητές πάνω στον άξονα του κινητήρα τοποθετείται κυλινδρικός δίσκος ο οποίος φέρει ομοιόμορφα καταναμημένα στην περιφέρεια του ετερόνυμες μαγνητικές περιοχές (βόρειους και νότιους πόλους). Σε αρκετά κοντινή απόσταση από την περιφέρεια του δίσκου και στο επίπεδο περιστροφής του

τοποθετείται αισθητήρας μαγνητικού πεδίου (Hall) ο οποίος ανιχνεύει αυτές τις εναλλαγές βόρειων και νότιων πόλων. Οι γραμμές του μαγνητικού πεδίου τέμνουν την επιφάνεια του αισθητήρα εγκάρσια ώστε να ισχύουν οι προϋποθέσεις για την εμφάνιση του φαινομένου Hall. Η έξοδος του αισθητήρα προκύπτει από την μέτρηση των ημιτονοειδών αλλαγών στην πυκνότητα της μαγνητικής ροής στην επιφάνεια του chip η οποία οδηγεί σε διακυμάνσεις την τάση Hall. Συσχετίζοντας ο αισθητήρας αυτές τις αυξομειώσεις στην τάση παράγει έναν τετραγωνικό παλμό η συχνότητα του οποίου είναι ανάλογη της ταχύτητα περιστροφής. Τοποθετώντας και έναν δεύτερο αισθητήρα περιμετρικά του δίσκου και στην ίδια ακτίνα αλλά σε γωνία  $90^\circ$  σε σχέση με τον πρώτο, παράγεται παλμοσειρά με διαφορά φάσης  $90^\circ$ . Από την προπορεία των παλμοσειρών μπορεί να διαπιστωθεί η φορά περιστροφής. Η μέτρηση της γωνίας στροφής είναι σχετική, μετριέται δηλαδή από κάποιο σημείο εκκίνησης ή σημείο αναφοράς και μετά. Η ανάλυση (resolution) των κωδικοποιητών αυξητικού τύπου καθορίζεται από τον αριθμό των παλμών που δημιουργούν ανά περιστροφή. Καλύτερη ανάλυση (resolution) επιτυγχάνεται είτε αυξάνοντας το πλήθος των μαγνητικών πόλων στο δίσκο είτε προσθέτοντας περισσότερους αισθητήρες. Ο κινητήρας *EMG49* διαθέτει στο πίσω μέρος του μια τέτοια μονάδα αυξητικού κωδικοποιητή που δίνει **980** απαριθμήσεις σε μια πλήρη περιστροφή.

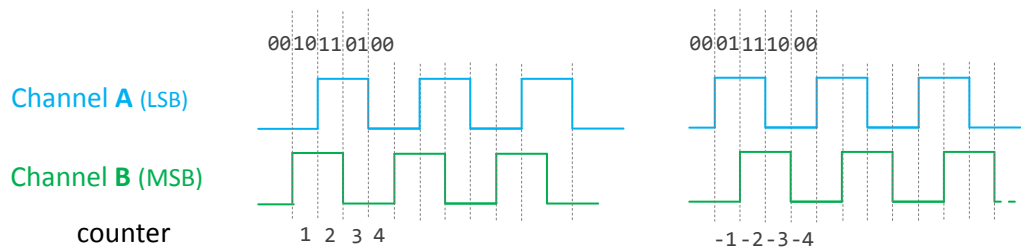
#### 5.4.2 Ανάγνωση κωδικοποιητή



Εικόνα 5.10 Περιπτώσεις αποκωδικοποίησης σημάτων κωδικοποιητή θέσης

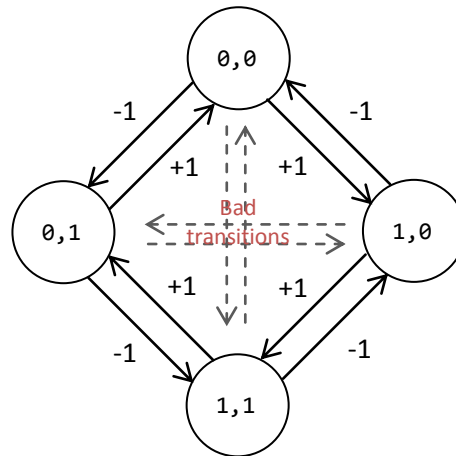
Καταμετρώντας τους παλμούς στα ψηφιακά σήματα εξόδου (κανάλια A και B, θα συμβολίζονται ως CHA και CHB αντίστοιχα) από το ζεύγος των ανιχνευτών του κωδικοποιητή είναι εφικτός ο υπολογισμός της ταχύτητας περιστροφής του κινητήρα. Η σχετική φάση των CHA και CHB παρέχει πληροφορία για τη φορά περιστροφής.

Για την απαρίθμηση των παλμών και την μέτρηση της ταχύτητας της περιστροφής του κινητήρα χρησιμοποιήθηκε ο μικροελεγκτής ATmega328P. Τα κανάλια A, B οδηγούνται στους δύο ακροδέκτες των εξωτερικών διακοπών INT0 και INT1 του μΕ αντίστοιχα. Η ανάγνωση του αυξητικού κωδικοποιητή του πραγματοποιήθηκε με αποκωδικοποίηση x4 οπότε η τιμή του απαριθμητή ενημερώνεται και στο ανερχόμενο και στο κατερχόμενο μέτωπο τόσο του CHA όσο και του CHB. Αν το σήμα του CHB προπορεύεται του CHA η κίνηση λαμβάνεται κατά τη θετική φορά και ο απαριθμητής αυξάνεται κατά 1. Αντίστοιχα εάν προπορεύεται το σήμα του CHA τότε έχουμε κίνηση κατά την ανάστροφη φορά και ο μετρητής απαρίθμησης μειώνεται κατά 1. Η μέθοδος αυτή μεγιστοποιεί τη διακριτική ικανότητα μέτρησης της μετατόπισης.



Εικόνα 5.11 Αποκωδικοποίηση x4

Το διάγραμμα καταστάσεων αποτυπώνει τις πιθανές μεταβάσεις οι οποίες μπορούν να προκύψουν κατά την περιστροφή. Οι μεταβάσεις αυτές χωρίζονται σε 3 κατηγορίες. Σε αυτές κατά τις οποίες δεν συμβαίνει καμία μετακίνηση. Η επόμενη ομάδα μεταβάσεων (bad transitions) δεν θα πρέπει να παρατηρούνται όμως αν η κίνηση είναι πολύ γρήγορη ή το σύστημα ανιχνεύει τις μεταβάσεις αργοπορημένα είναι δυνατόν αυτές να φαίνεται ότι συμβαίνουν (αύξηση ή μείωση του μετρητή απαρίθμησης κατά 2). Στην τελευταία ομάδα ανήκουν οι αναμενόμενες μεταβάσεις που πραγματοποιούνται φυσιολογικά κατά την διάρκεια της περιστροφής (αύξηση ή μείωση του μετρητή απαρίθμησης κατά 1).



Εικόνα 5.12 Διάγραμμα καταστάσεων για αποκωδικοποίηση x4

Πίνακας καταστάσεων

state	new		old		result
	channel B	channel A	channel B	channel A	
0	0	0	0	0	no movement
1	0	0	0	1	+1
2	0	0	1	0	-1
3	0	0	1	1	+2
4	0	1	0	0	-1
5	0	1	0	1	no movement
6	0	1	1	0	-2
7	0	1	1	1	+1
8	1	0	0	0	+1
9	1	0	0	1	-2
10	1	0	1	0	no movement
11	1	0	1	1	-1
12	1	1	0	0	+2
13	1	1	0	1	-1
14	1	1	1	0	+1
15	1	1	1	1	no movement

Η ρουτίνα εξυπηρέτησης κάθε εξωτερικής διακοπής καλεί των κώδικα της συνάρτησης που ακολουθεί. Η καθολική (global) μεταβλητή state απαρτίζεται από 4 δυαδικά ψηφία τα οποία αποτελούν σύνθεση των bit της τρέχουσας κατάστασης των CHB, CHA και της προηγούμενης τους κατάστασης.

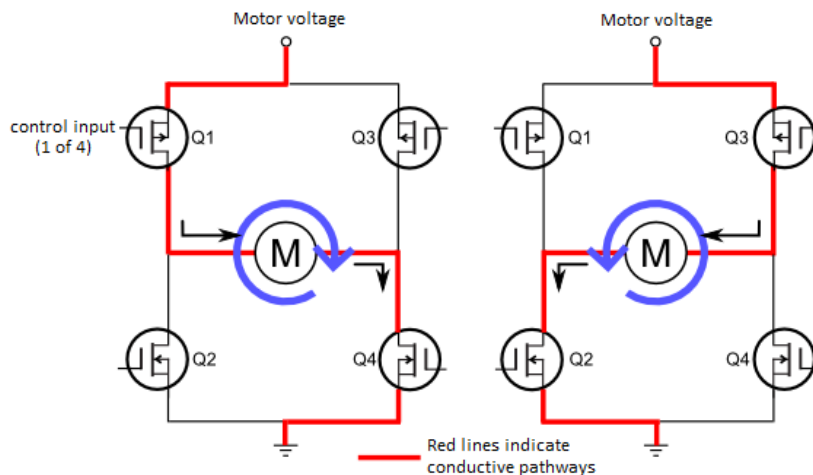
```
void update(void) {
    // get old status
    uint8_t s = state & 3; // bitwise AND with 0000 0011

    // read the value of the digital pin that corresponds to the channel A
    if (digitalRead(CHA)) s |= 4; // bitwise OR with 0000 0100

    // read the value of the digital pin that corresponds to the channel B
    if (digitalRead(CHB)) s |= 8; // bitwise OR with 0000 1000

    switch (s) {
        case 0: case 5: case 10: case 15: // no movement
            break;
        case 1: case 7: case 8: case 14: // increment by one
            count++;
            break;
        case 2: case 4: case 11: case 13: // decrement by one
            count--;
            break;
        case 3: case 12: // increment by two
            count += 2;
            break;
        case 6: case 9: // decrement by two
            count -= 2;
            break;
    }
    // do the new status old
    state = (s >> 2); // right shift
}
```

## 5.5 Οδήγηση του σερβοκινητήρα EMG49 μέσω H-γέφυρας



Εικόνα 5.13 Οδήγηση κινητήρα μέσω H-bridge

Για τον έλεγχο της ροής ισχύος προς τον κινητήρα χρησιμοποιείται H-γέφυρα. Τα τρανζίστορ ισχύος της γέφυρας ανοιγοκλείνουν ταχύτατα οδηγούμενα από σήματα διαμορφωμένα κατά πλάτος όποτε ο κινητήρας αντιλαμβάνεται τάση

$$V_m = \frac{t_{on}}{T_{pwm}} V$$

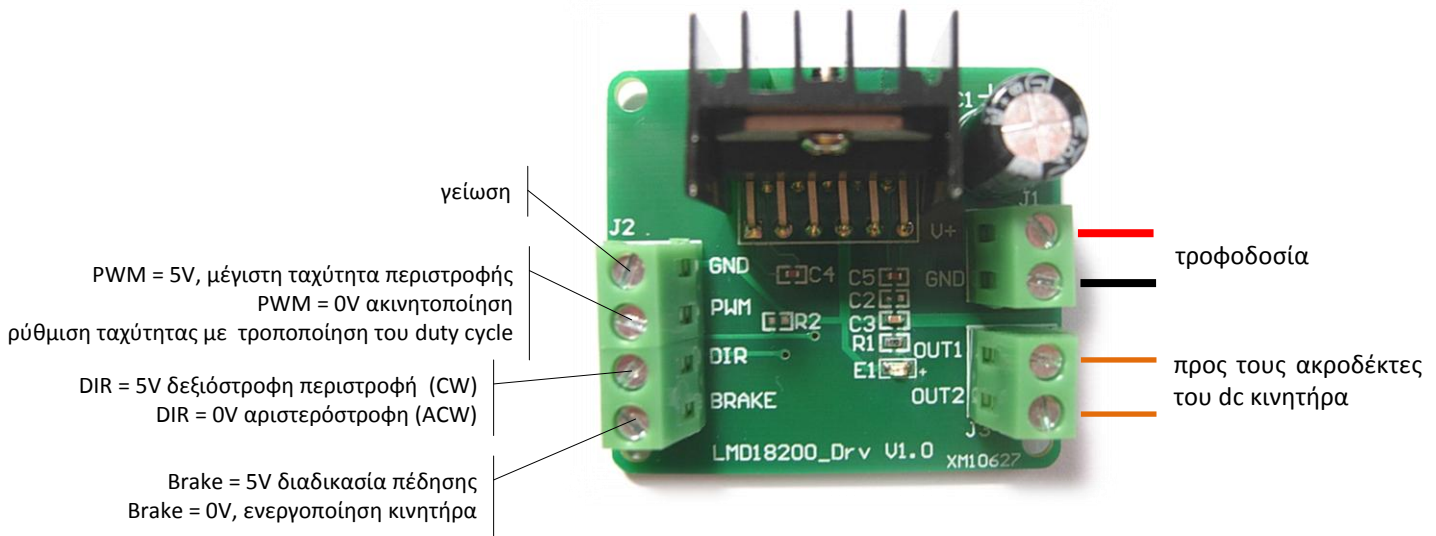
όπου

$t_{on}$  ο χρόνος που τα τρανζίστορ είναι ανοικτά  
 $T_{pwm}$  η περίοδος του σήματος ελέγχου  
 $V$  η τάση τροφοδοσίας της γέφυρας

Ελέγχοντας συνεπώς τον χρόνο  $t_{on}$  μπορεί να ελεγχθεί η τάση τυμπάνου του κινητήρα άρα και η ταχύτητα περιστροφής του. Η ταχύτητα περιστροφής συνεπώς ρυθμίζεται μέσω του duty cycle ενός PWM σήματος.

Η οδήγηση του κινητήρα EMG49 γίνεται με τη βοήθεια του ολοκληρωμένου LMD 18200T P+ το οποίο αποτελεί μια διάταξη H-γέφυρας για τον έλεγχο μέσω PWM σήματος της ταχύτητας και της φοράς περιστροφής του με μέγιστο συνεχές ρεύμα 3A και μέγιστη τάση τροφοδοσίας 55V. Η διαχείριση της γέφυρας γίνεται μέσω δύο ψηφιακών σημάτων (PWM, DIR) που παρέχονται από τον μικροελεγκτή ATmega328P-PU. Χρησιμοποιείται η αυτοματοποιημένη παραγωγή σήματος PWM στον ακροδέκτη 15 (OC1A) του μικροελεγκτή από τον Timer1 μέσω της λειτουργίας Phase Correct PWM mode 2 με 9-bit ρύθμιση του duty cycle. Η διακριτική ικανότητα του duty cycle είναι  $1/511*100 = 0.196\%$  Η συχνότητα του παραγόμενου PWM σήματος καθορίζεται από την επιλεγμένη τιμή  $N(=1)$  του prescaler και την συχνότητα ρολογιού του μικροελεγκτή  $F_{ck} = 16MHz$

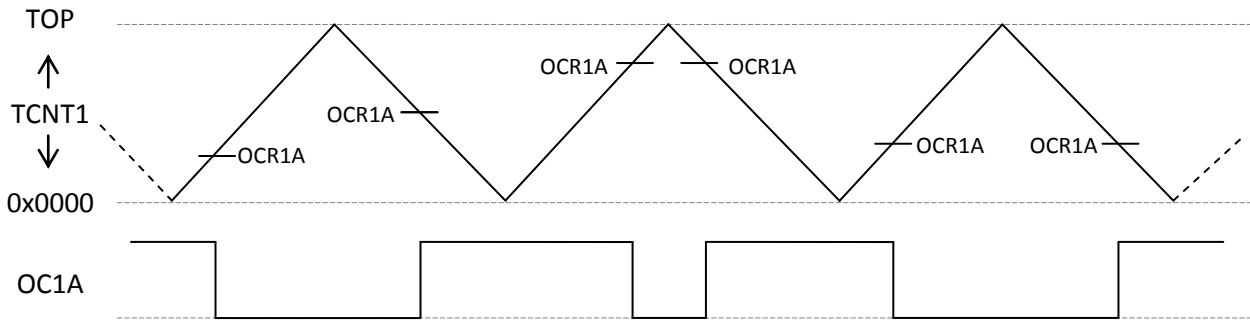
$$F_{pwm} = \frac{F_{ck}}{2 * N * TOP} = 15.656 KHz$$



Εικόνα 5.14 Η H-γέφυρα με βάση το ολοκληρωμένο κύκλωμα LPD 18200T P+

Η λειτουργία του Phase Correct PWM mode του Timer1 συνοπτικά είναι:

- Ο TCNT1 αρχικά αυξάνεται από το 0.
- Όταν υπάρξει ταύτιση με την τιμή του OCR1A η κατάσταση του OC1A εναλλάσσεται.
- Μόλις ο TCNT1 πάρει την μέγιστη τιμή TOP (=511) αρχίζει να μειώνεται κινούμενος προς το 0.
- Όταν κατεβαίνοντας υπάρξει ταύτιση με την τιμή του OC1A η κατάσταση του OC1A εναλλάσσεται.
- Μόλις ο TCNT1 πάρει την τιμή 0, αρχίζει πάλι να αυξάνεται κινούμενος προς το TOP και η παραπάνω διαδικασία ξεκινάει εκ νέου.



Εικόνα 5.15 Διάγραμμα χρονισμού του Phase Correct PWM mode

## 5.6 Εκτίμηση γωνιακής ταχύτητας περιστροφής

Για τον έλεγχο της ταχύτητας περιστροφής του συστήματος dc κινητήρα-φορτίου απαιτείται η εκτίμηση της γωνιακής ταχύτητας αφού με τη δεδομένη διάταξη δεν υπάρχει δυνατότητα απευθείας μέτρησης. Σε πρακτικό επίπεδο προσεγγίσεις με αυξημένες απαιτήσεις υπολογιστικής ισχύος δεν είναι δυνατό να υλοποιηθούν. Εναλλακτικά, ο υπολογισμός της ταχύτητας μπορεί να γίνει με κάποιο σχήμα αριθμητικής διαφόρισης για την εκτίμηση της παραγώγου του αντίστοιχου σήματος θέσης από τους χρησιμοποιούμενους κωδικοποιητές.

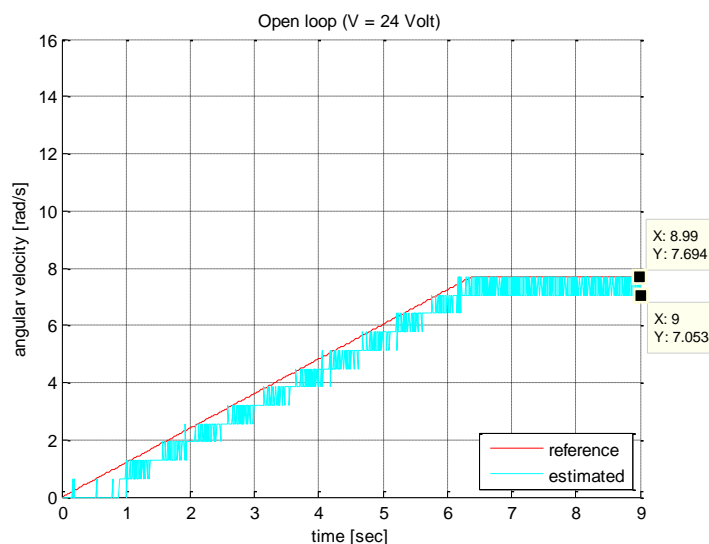
Αρχικά χρησιμοποιήθηκε η μέθοδος των προς τα πίσω διαφορών πρώτης τάξης. Η προσέγγιση αυτή αν και ιδιαίτερα απλή ενισχύει σημαντικά το θόρυβο κβάντισης που χαρακτηρίζει τις μετρήσεις των κωδικοποιητών θέσης. Η παρουσία αυτών των συνιστωσών θορύβου στο σήμα ανάδρασης της ταχύτητας υποβαθμίζει τον έλεγχο του συστήματος αφού υπεισέρχονται στον υπολογισμό του σφάλματος αλλά και του σήματος ελέγχου.

Στην παρούσα εφαρμογή πραγματοποιείται δειγματοληψία με περίοδο  $T_s = 0.01 \text{ sec}$  στο  $\theta(t)$  σήμα του κωδικοποιητή θέσης. Σε μια πλήρη περιστροφή του δρομέα ο απαριθμητής μέτρησης τροποποιείται κατά 980 μονάδες στην x4 αποκωδικοποίηση οπότε σε μια στοιχειώδη μετατόπιση  $\Delta\theta$  μέσα στον χρόνο δειγματοληψίας ο απαριθμητής μέτρησης αλλάζει κατά  $\Delta\theta/980$ .

$$\hat{\omega} = \frac{\Delta\theta}{980} \frac{rev}{T_s \text{ sec}} = \frac{2\pi\Delta\theta}{980T_s} \frac{rad}{sec} = \frac{60\Delta\theta}{980T_s} \text{ rpm}$$

όπου  $\Delta\theta = \theta_k - \theta_{k-1}$  με  $\theta_k$  η τιμή του απαριθμητή θέσης την χρονική στιγμή  $t_k = t_{k-1} + T_s$ . Το σφάλμα στην εκτίμηση της ταχύτητας  $\omega_k$  είναι ανεξάρτητο από την ταχύτητα περιστροφής και είναι  $-d/2 \leq e_k \leq d/2$  με το  $d$  να είναι

$$d = \frac{2\pi}{980T_s} = 0.6411 \frac{rad}{sec}$$



Εικόνα 5.16 Εκτίμηση γωνιακής ταχύτητας με διαφόριση του σήματος θέσης

Η δεύτερη μέθοδος που υλοποιήθηκε για την εκτίμηση της γωνιακής ταχύτητας βασίζεται στο φίλτρο πεπερασμένης κρουστικής απόκρισης (F.I.R.)<sup>5</sup> **Savitzky-Golay**. Ο αλγόριθμος εφαρμόζει διαδοχικά σε κάθε υποομάδα συγκεκριμένου πλήθους γειτονικών σημείων τη μέθοδο των ελαχίστων τετραγώνων προσεγγίζοντας τα με πολυώνυμο μικρού βαθμού. Όταν οι τιμές του μελετώμενου σήματος προκύπτουν από δειγματοληψία με σταθερό ρυθμό μπορεί να υπολογιστούν με αναλυτικές μεθόδους οι συντελεστές στατιστικού βάρους με τους οποίους θα πραγματοποιηθεί η συνέλιξη με τις τιμές του σήματος σε κάθε χρονικό παράθυρο για την εξομάλυνση του. Με κατάλληλες ομάδες ακεραίων συνέλιξης μπορούν να ληφθούν απευθείας οι παράγωγοι 1<sup>ης</sup>, 2<sup>ης</sup> κλπ τάξης ενός διακριτού σήματος ισάπεχοντων σημείων. Για ένα σύνολο σημείων  $(x_j, y_j)$  με το βήμα  $x_j - x_{j-1} = h$  σταθερό, η πρώτη παράγωγος είναι:

$$Y'_j = \frac{1}{Ch} \sum_{i=-m}^m y_{j+i}$$

όπου  $m = \frac{n+1}{2} - 1$  με το  $n \geq 3$  και περιττό αριθμό. Το  $n$  αντιπροσωπεύει το πλήθος των δειγμάτων που προσπελούνται (εύρος χρονικού παραθύρου) και  $C$  ο συντελεστής κανονικοποίησης.

Οι συντελεστές μπορούν να υπολογιστούν μέσω της συνάρτησης `sgolay` του Matlab.

```
N = 2; % βαθμός πολυωνύμου
F = 5; % μέγεθος χρονικού παραθύρου
[~,g] = sgolay(N,F); % υπολογισμός συντελεστών S-G
g(:,2)
```

<sup>5</sup> Τα ψηφιακά φίλτρα Πεπερασμένης Κρουστικής Απόκρισης ονομάζονται έτσι διότι η κρουστική τους απόκριση έχει μη μηδενικές τιμές που περιορίζονται χρονικά σε ένα «παραθύρο» έστω  $M$  δειγμάτων, ενώ πριν και μετά οι τιμές της είναι μηδενικές, άρα είναι πεπερασμένη χρονικά.

**Συντελεστές για τον υπολογισμό της 1ης παραγώγου**

Βαθμός πολωνύμου	1 <sup>ου</sup> , 2 <sup>ου</sup>				
Μέγεθος χρονικού παραθύρου <i>n</i>	3	5	7	9	...
-4				-4	
-3			-3	-3	
-2		-2	-2	-2	
-1	-1	-1	-1	-1	
0	0	0	0	0	
1	1	1	1	1	
2		2	2	2	
3			3	3	
4				4	
Συντελεστής κανονικοποίησης <i>C</i>	2	10	28	60	

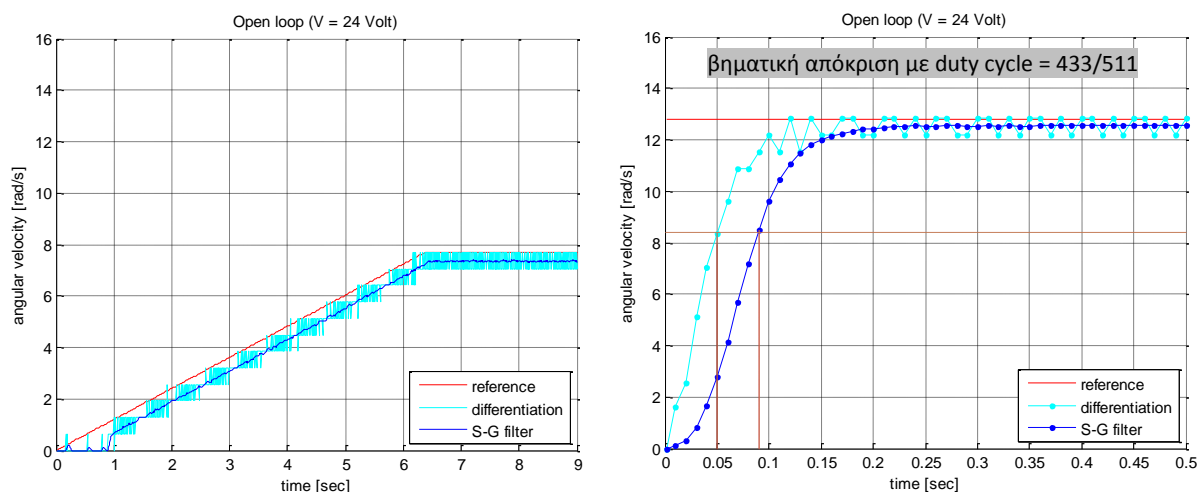
Για παράδειγμα η πρώτη παράγωγος με  $n = 5$  είναι

$$Y'_j = \frac{1}{10h} (-2 * y_{j-2} - 1 * y_{j-1} + 0 * y_j + 1 * y_{j+1} + 2 * y_{j+2})$$

Στο σύστημα dc κινητήρα-τροχού ανοικτού βρόχου η εκτίμηση της ταχύτητας από τις μετρήσεις θέσεις μπορεί να γίνει μέσω της ακόλουθης σχέσης:

$$\hat{\omega}_k = \frac{2\pi}{980} \frac{1}{CT_s} \sum_{i=0}^{n-1} \left( \frac{n-1}{2} - i \right) * \theta_{k-i}$$

Η ταχύτητα συνεπώς τη χρονική στιγμή  $k$  είναι ο γραμμικός συνδυασμός του τρέχοντος και των  $n - 1$  προηγούμενων δειγμάτων της θέσης. Στην συγκεκριμένη εφαρμογή χρησιμοποιήθηκε πολώνυμο 2<sup>ου</sup> βαθμού και χρονικό παράθυρο  $9 * 0.01 \text{sec} = 90 \text{ms}$ . ( $n = 9, C = 60$ ). Διαπιστώνεται από τα πειραματικά αποτελέσματα ότι η συγκεκριμένη μέθοδος για την εκτίμηση της ταχύτητας οδηγεί στην καταστολή του θορύβου κβάντισης επιτυγχάνοντας βελτιωμένη απόκριση εισάγοντας όμως χρονική καθυστέρηση περίπου 40ms.



**Εικόνα 5.17** Εκτίμηση γωνιακής ταχύτητας με διαφόριση και με χρήση του φίλτρου Savitzky–Golay



## 5.7 Πειραματική ταυτοποίηση συστήματος

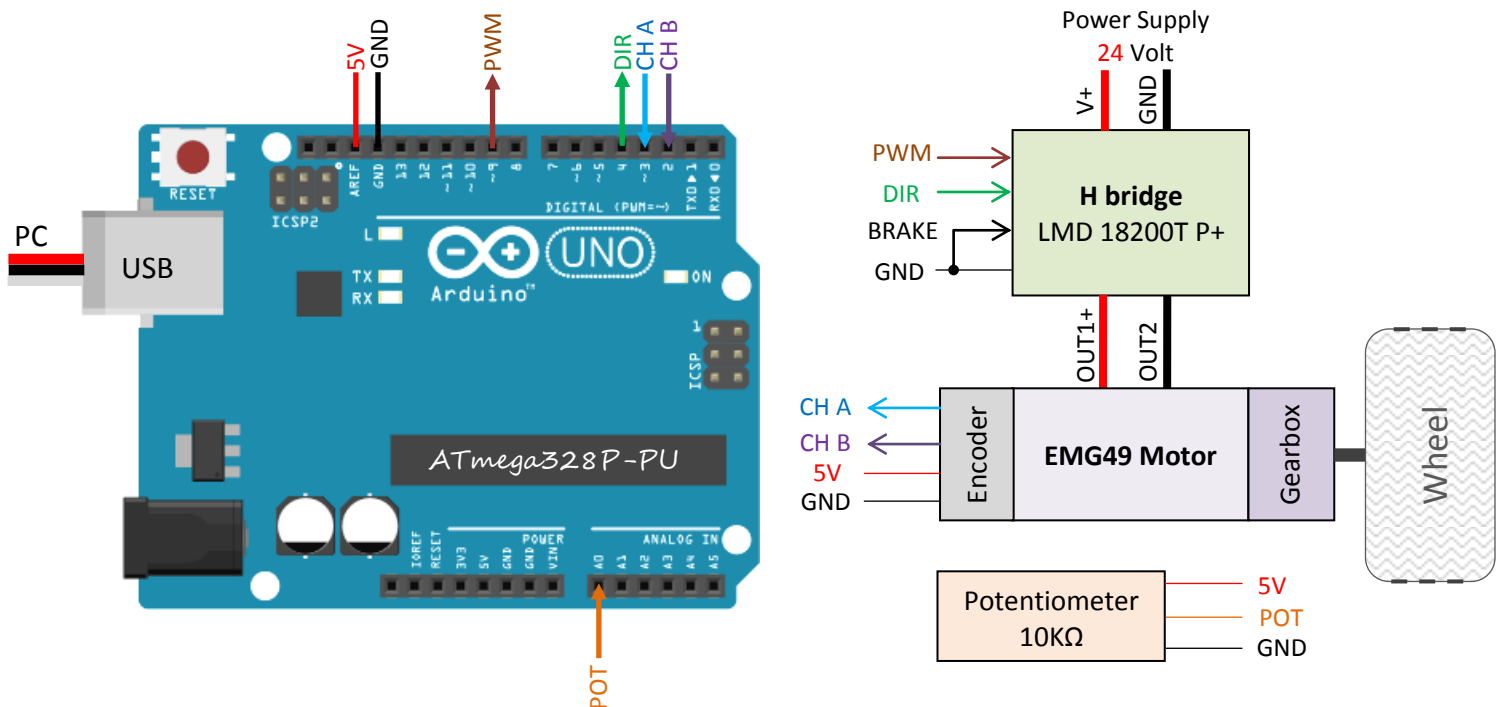
Για την υλοποίηση των δοκιμών σχεδιάστηκε και κατασκευάστηκε μια ξύλινη βάση τοποθέτησης του οχήματος ώστε να μπορούν να πραγματοποιηθούν οι απαραίτητες μετρήσεις. Για την διεξαγωγή πειραμάτων και την λήψη δεδομένων του συστήματος ανοικτού και κλειστού βρόχου πραγματοποιήθηκε το κύκλωμα το οποίο απεικονίζεται στην εικόνα 5.18. Ως φορτίο θεωρείται ο τροχός, ο οποίος δεν ακουμπά στο έδαφος, μαζί με τα παρελκόμενα για την σύνδεση του με τον κινητήρα άξονα.

Ο κύκλος ελέγχου χρονίζεται στα  $10\text{ ms}$  με τη χρήση χρονικού interrupt του Timer0 ώστε να λαμβάνονται 10 δείγματα στο χρονικό διάστημα που αντιστοιχεί στο χρόνο αποκατάστασης του συστήματος. Η βασική δομή σε ψευδοκώδικα του κυρίως βρόχου ελέγχου της εφαρμογής δίνεται παρακάτω:

```
while(1) {
    ...
    while(wait == 1);
}
Timer interrupt ISR: wait = 0;
```

Τα σήματα αναφοράς είναι κυρίως περιοδικά επαναλαμβανόμενα και δημιουργούνται στον κώδικα της εφαρμογής. Σε κάθε κύκλο ελέγχου για το σύστημα ανοικτού βρόχου αποστέλλονται μέσω της σειριακής θύρας το διάστημα των χρονικών στιγμών της μέτρησης και δεδομένα σχετικά με την τρέχουσα τιμή του σήματος αναφοράς (OCR1A), την τιμή του μετρητή απαρίθμησης του αποκωδικοποιητή θέσης και την εκτιμώμενη γωνιακή ταχύτητα του κινητήρα με βάση το φίλτρο Savitzky–Golay. Τα πειραματικά δεδομένα αποθηκεύτηκαν σε αρχεία τύπου \*.mat

Για την παραγωγή των γραφημάτων και την επεξεργασία των πειραματικών δεδομένων αναπτύχθηκαν στο *Matlab* κατάλληλα scripts.

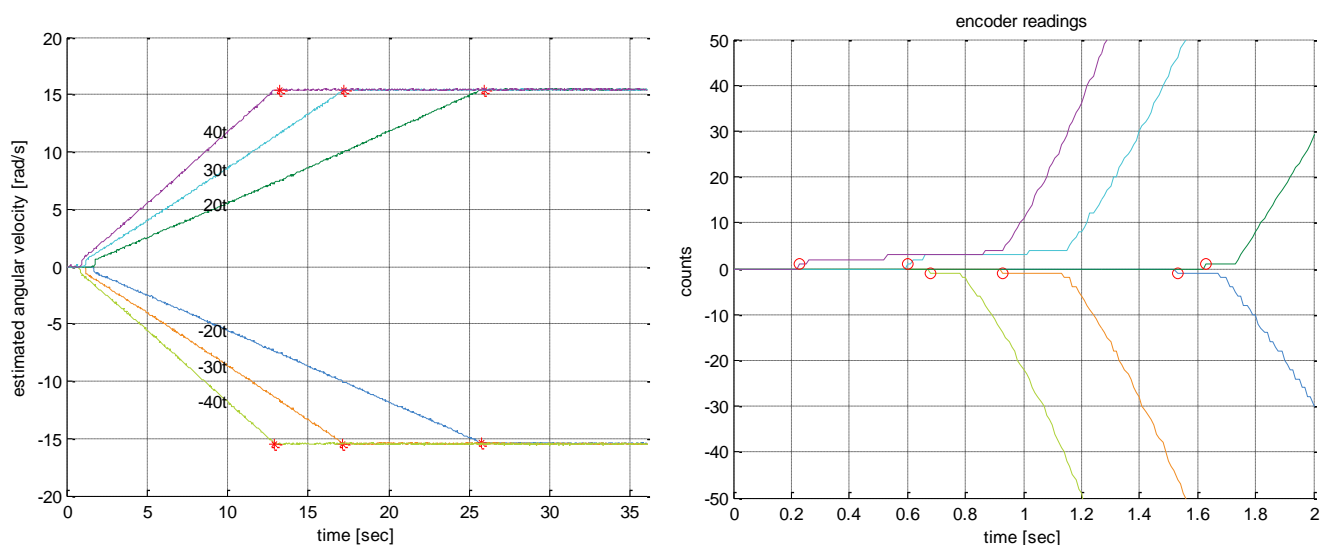


Εικόνα 5.18 Σχηματικό διάγραμμα της πειραματικής διάταξης ελέγχου και λήψης δεδομένων

Για τον προσδιορισμό των ορίων της νεκρής ζώνης<sup>6</sup> του συστήματος κινητήρα EMG49-τροχού και την εύρεση της μέγιστης γωνιακής ταχύτητας που μπορεί να αναπτυχθεί όταν τροφοδοτείται με την ονομαστική του τάση (24V) καταγράφεται η εκτιμώμενη γωνιακή ταχύτητα του καθώς το duty cycle (OCR1A/TOP) αυξάνεται γραμμικά με αργό ρυθμό μέχρι να σταθεροποιηθεί στην μέγιστη τιμή του.

Ο υπολογισμός της μέγιστης γωνιακής ταχύτητας που μπορεί αναπτύξει κάθε τροχός υπολογίστηκε από τον μέσο όρο των γωνιακών ταχυτήτων μετά την σταθεροποίηση της τιμής του OCR1A. Τα όρια της νεκρής ζώνης προσδιορίζονται από τη μετατόπιση του τροχού καθώς η εφαρμοζόμενη τάση αυξάνεται με αργό ρυθμό. Η διαδικασία αυτή πραγματοποιήθηκε τόσο για θετικές όσο και για αρνητικές τιμές της τάσης. Η ελάχιστη τιμή του OCR1A η οποία απαιτείται για να τεθεί σε κίνηση κάθε τροχός για την δεξιόστροφη (CW) και την αριστερόστροφη (ACW) φορά περιστροφής καθώς και η μέγιστη ταχύτητα που μπορεί να αναπτύξει αναγράφονται στον κάτωθι πίνακα:

Τροχός	Μέγιστη γωνιακή ταχύτητα (rad/s)	Νεκρή ζώνη (duty cycle)	
		CW	ACW
Εμπρόσθιος αριστερός (FLW)	15.436	20/511	28/511
Εμπρόσθιος δεξιός (FRW)	15.256	18/511	16/511
Οπίσθιος αριστερός (RLW)	15.077	26/511	19/511
Οπίσθιος δεξιός (RRW)	15.17	21/511	15/511

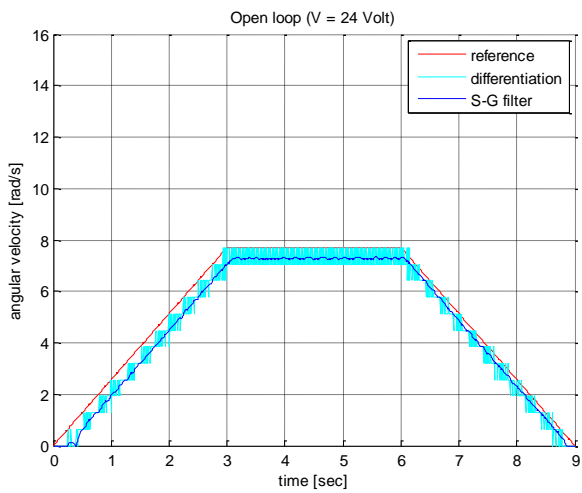
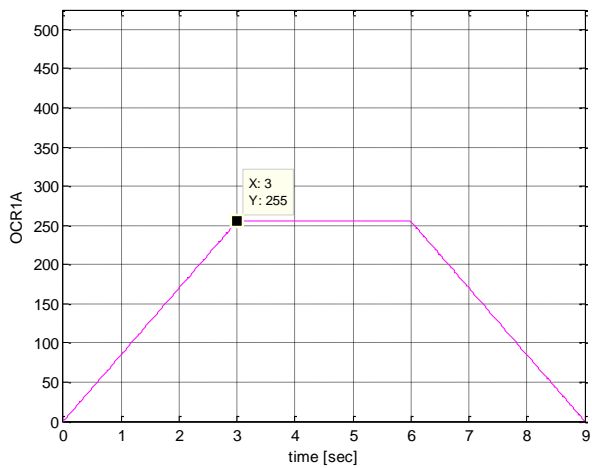
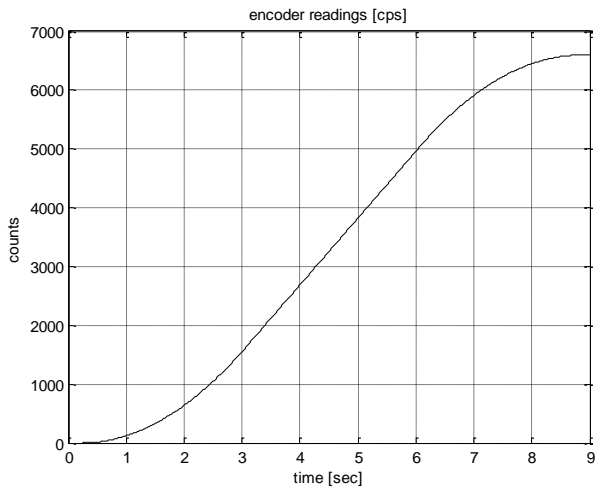


Εικόνα 5.19 Εύρεση μέγιστης ταχύτητας και νεκρής ζώνης εμπρόσθιου αριστερού τροχού

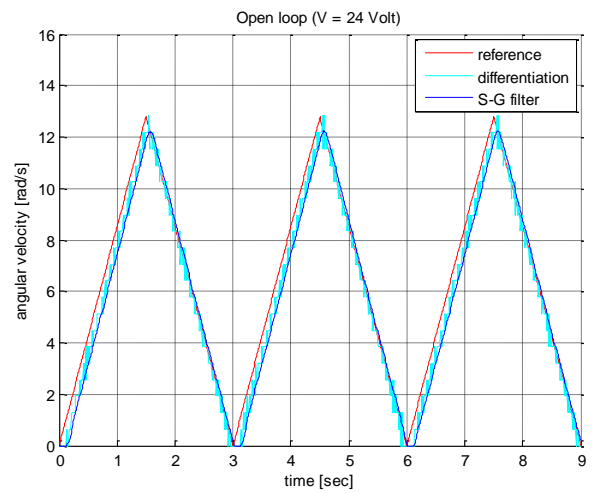
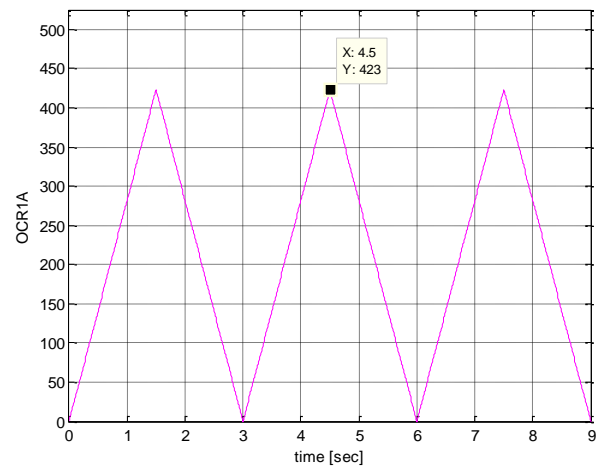
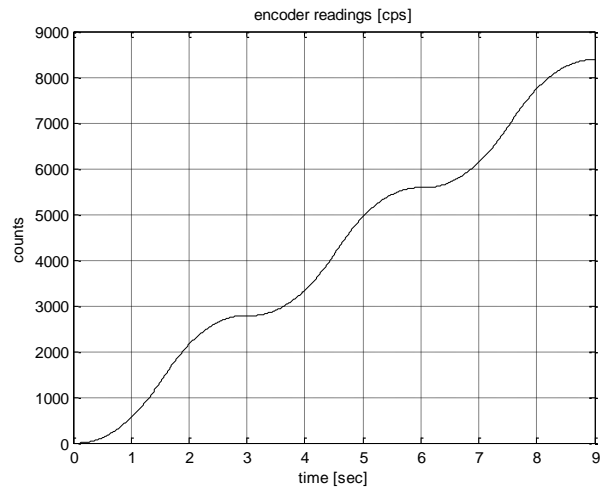
<sup>6</sup> Εκτός από τη γραμμική δυναμική τριβή η οποία συμπεριλαμβάνεται στο γραμμικό μοντέλο, στο σύστημα υπάρχουν και στατικές τριβές στα κινούμενα μέρη που το απαρτίζουν. Απαιτείται συνεπώς μια ελάχιστη ροπή προκειμένου να υπερνικηθεί η στατική τριβή. Πρακτικά είναι η περιοχή τιμών της εφαρμοζόμενης τάσης για την οποία δεν υπάρχει περιστροφή.

### 5.7.1 Πειραματικά αποτελέσματα ταχύτητας περιστροφής dc κινητήρα-τροχού ανοικτού βρόγχου

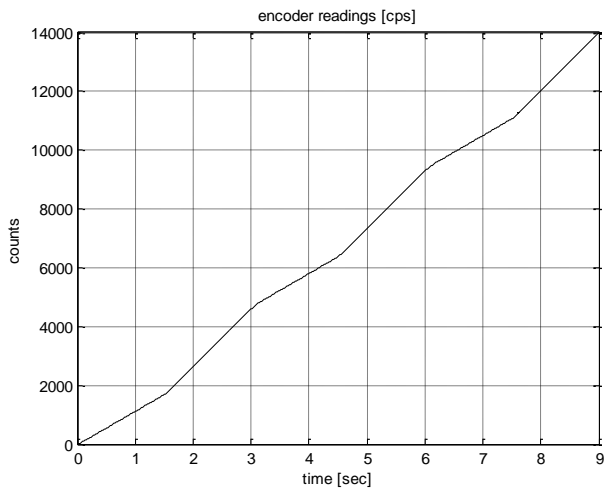
Τραπεζοειδές προφίλ κίνησης



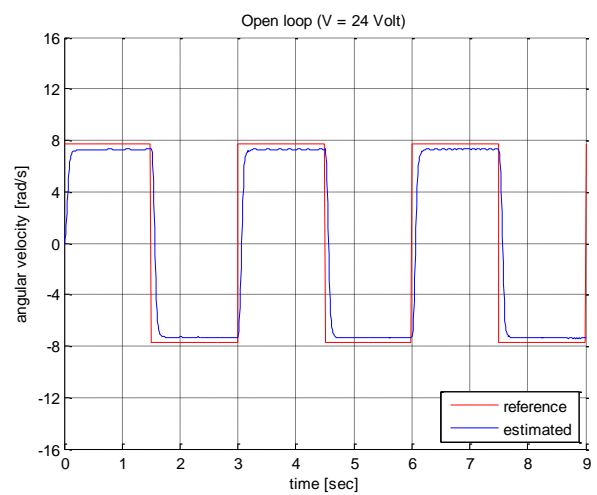
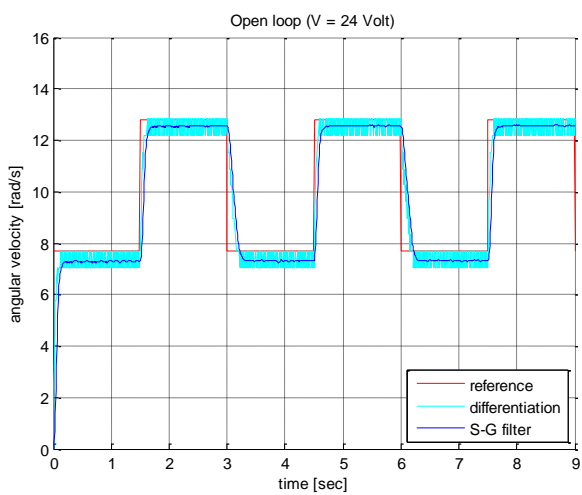
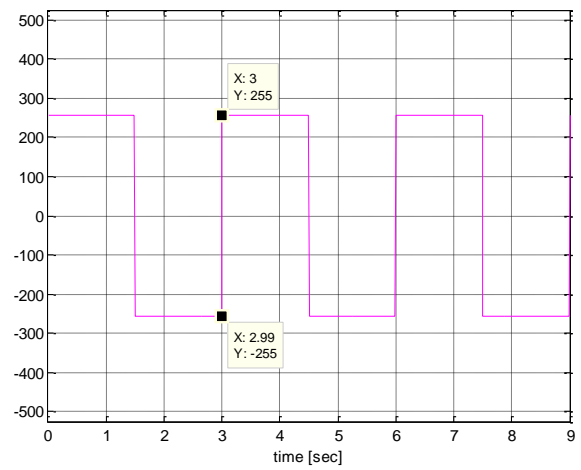
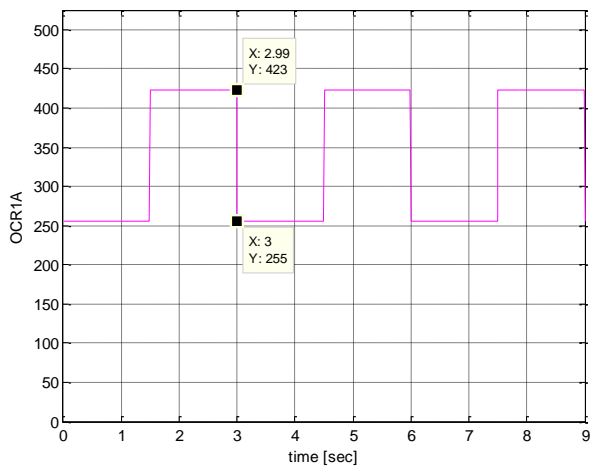
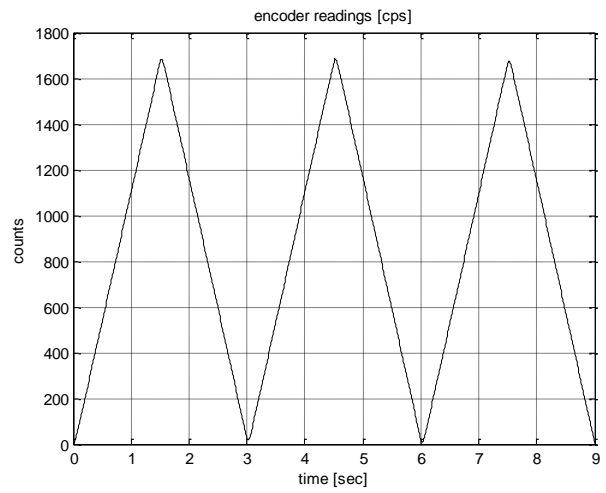
Τριγωνικό προφίλ κίνησης



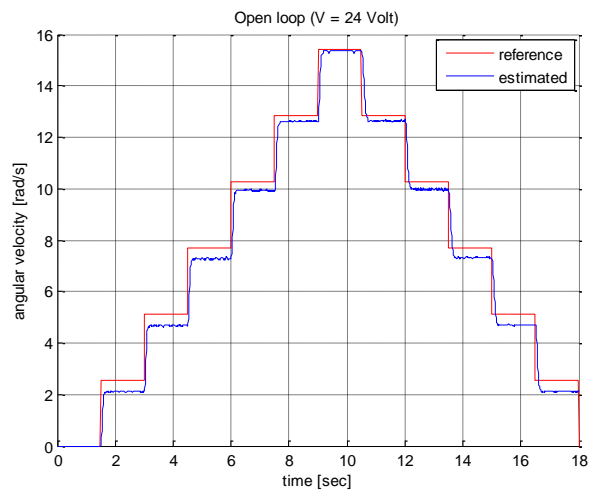
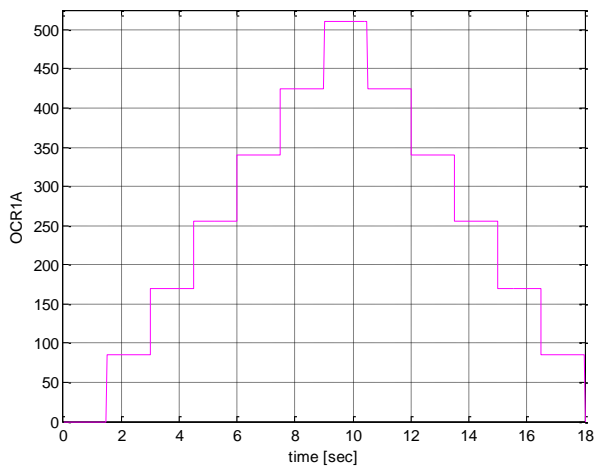
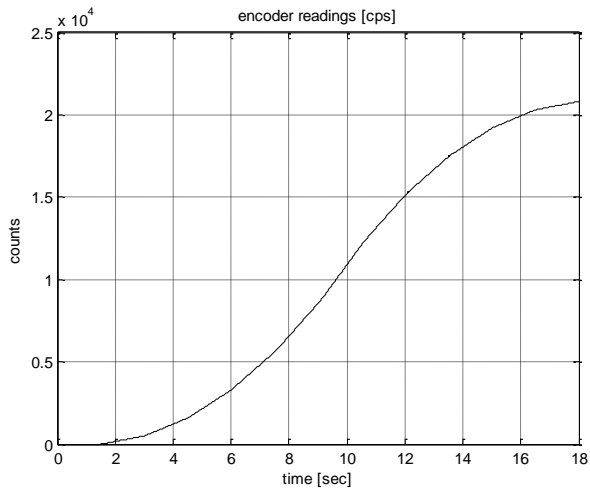
Τετραγωνικός παλμός



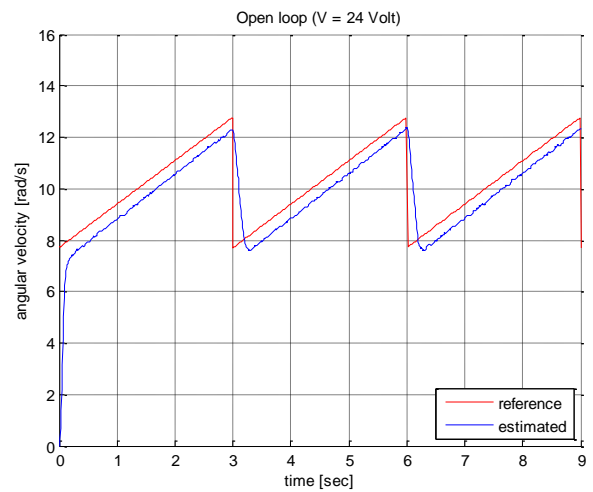
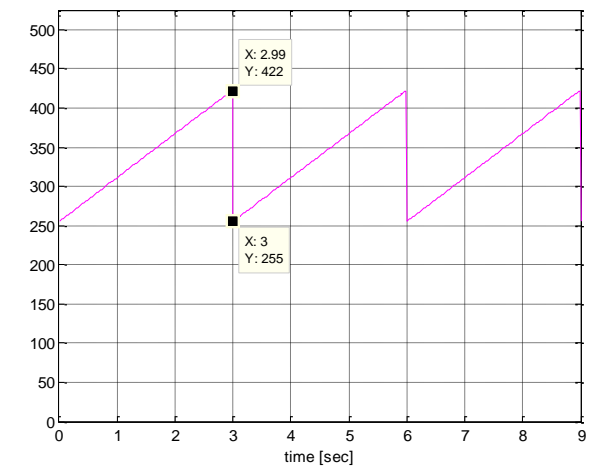
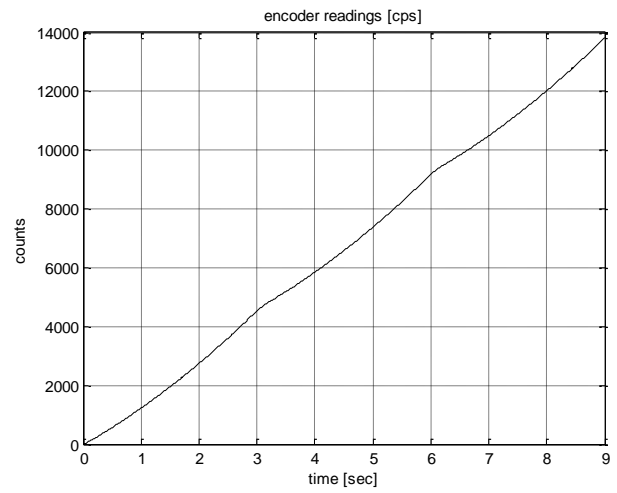
Τετραγωνικός παλμός με αλλαγή της φοράς περιστροφής (duty cycle = 255/511)



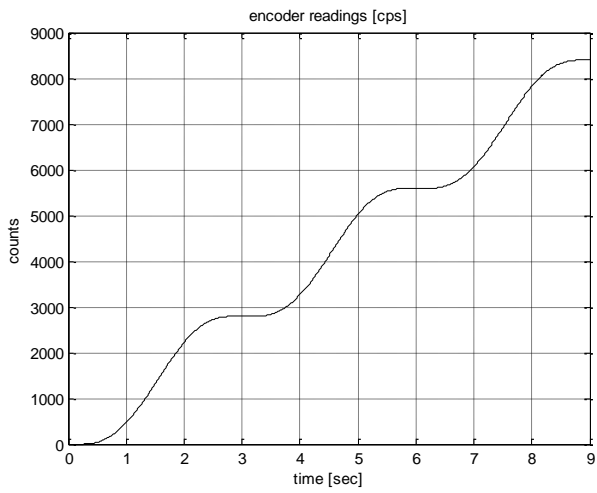
Κλιμακωτά τροποποιούμενο προφίλ κίνησης



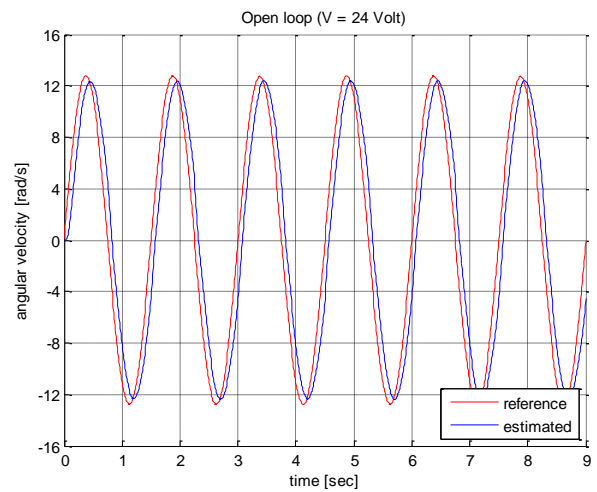
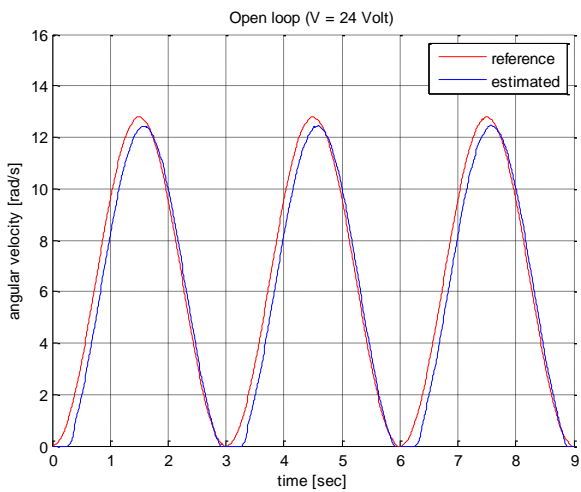
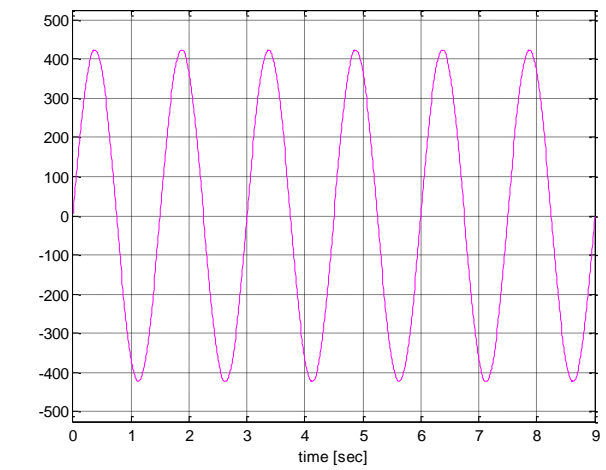
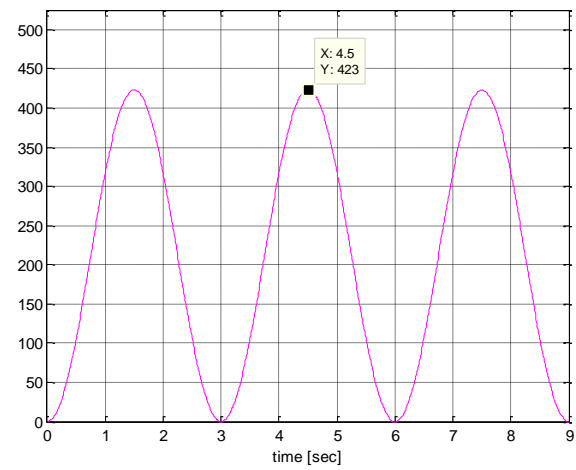
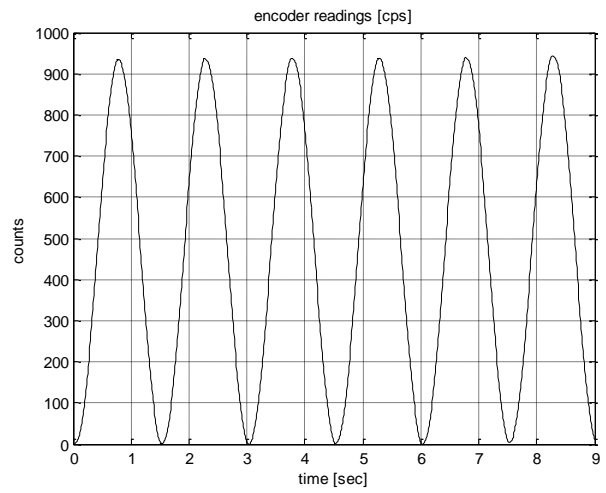
Πριονωτό προφίλ κίνησης



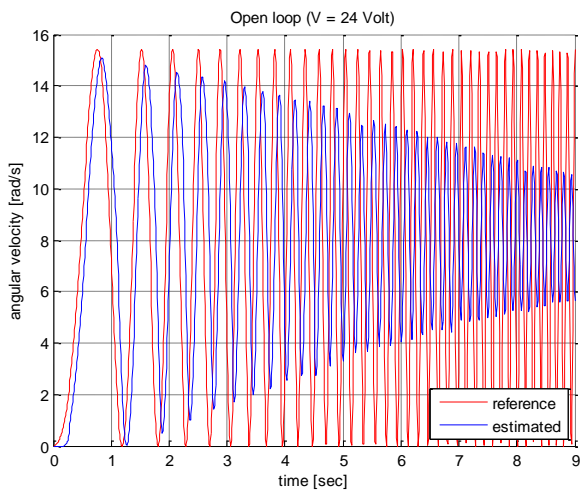
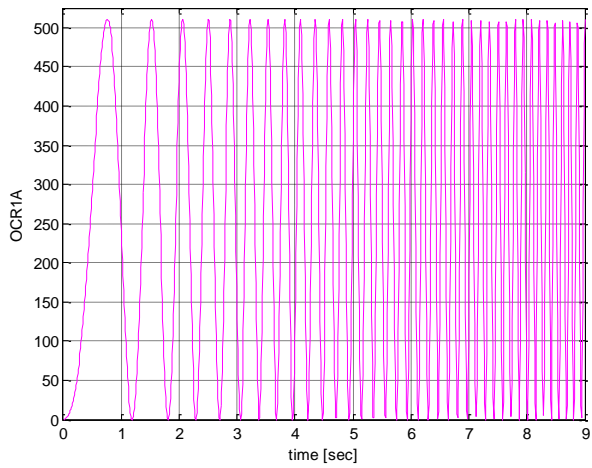
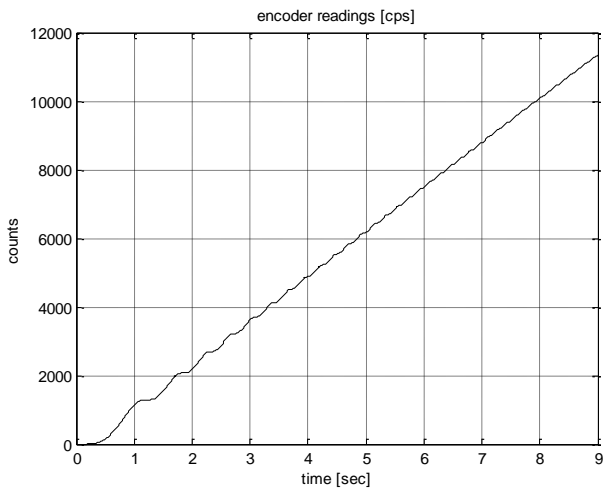
Ημιτονοειδές προφίλ κίνησης



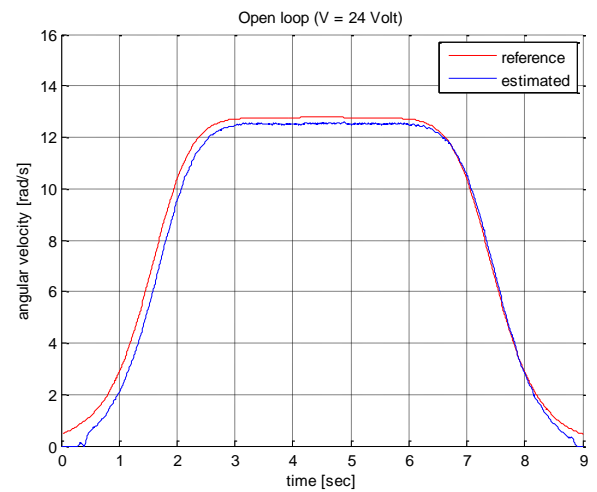
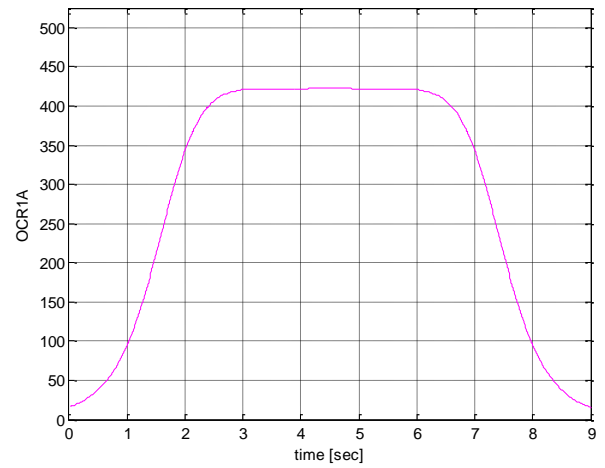
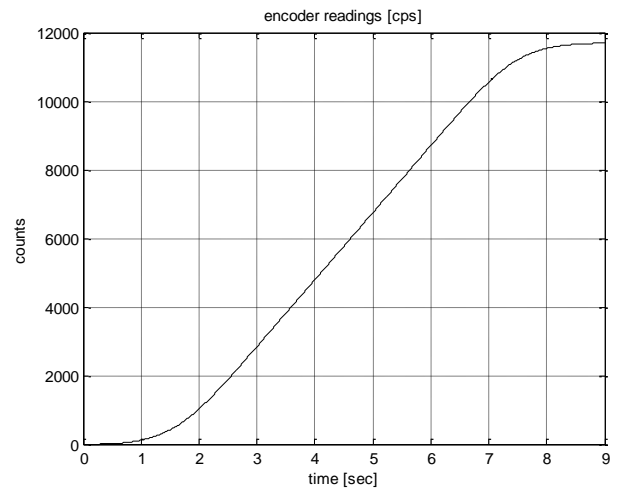
Ημιτονοειδές προφίλ κίνησης με αλλαγή της φοράς περιστροφής



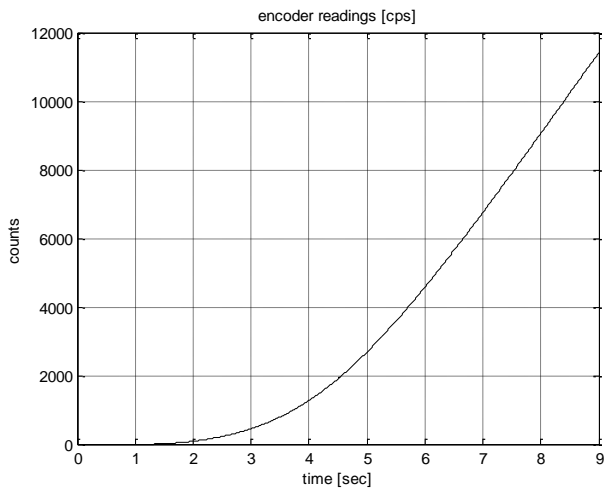
Ημιτονοειδές προφίλ κίνησης με γραμμική αύξηση της συχνότητας



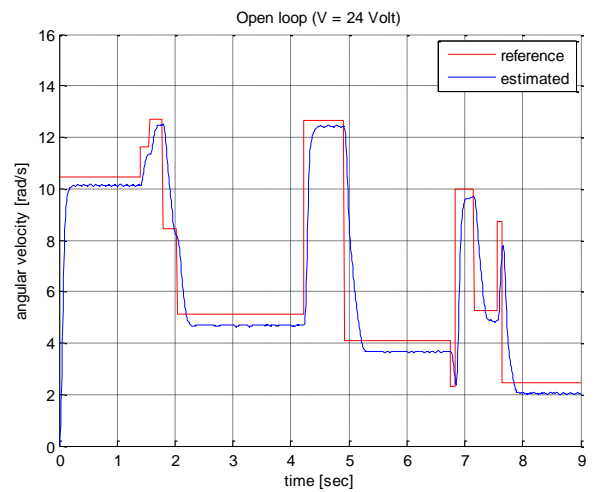
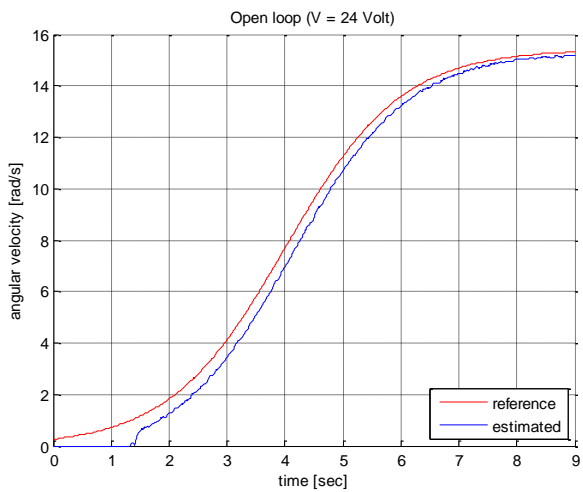
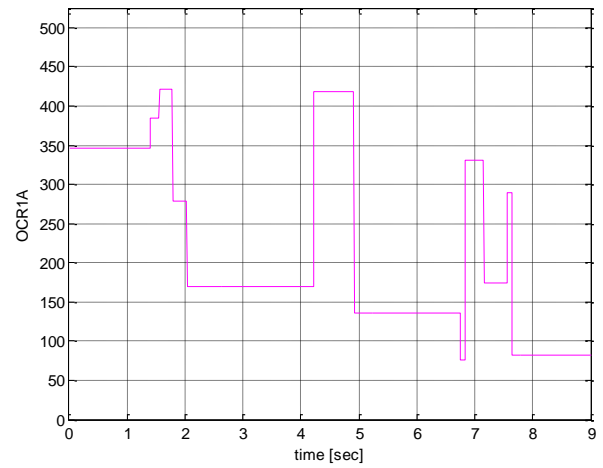
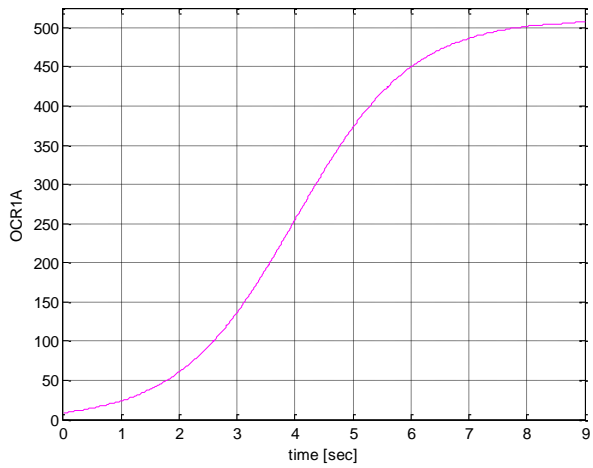
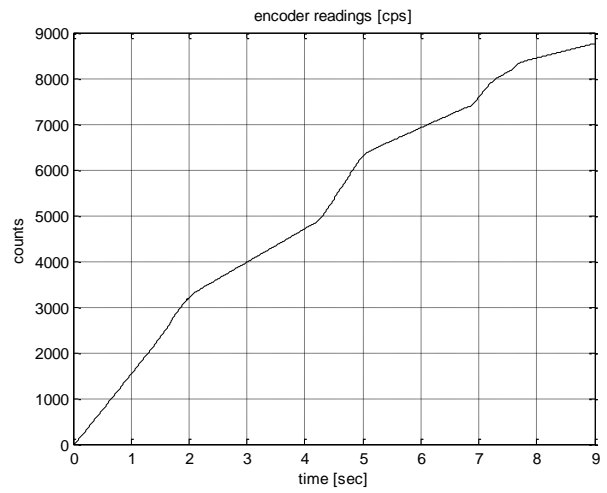
Bell shaped



Σιγμοειδές προφίλ κίνησης

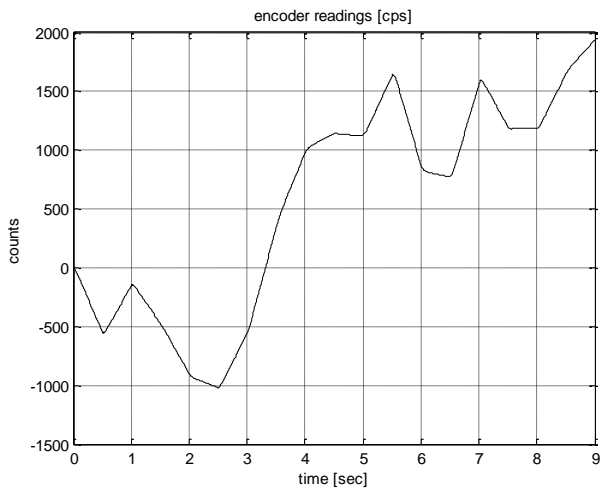


Τυχαία τροποποιούμενο με βηματικές αλλαγές

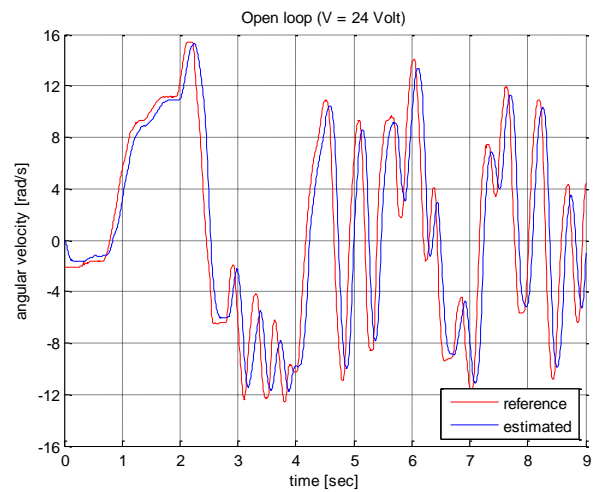
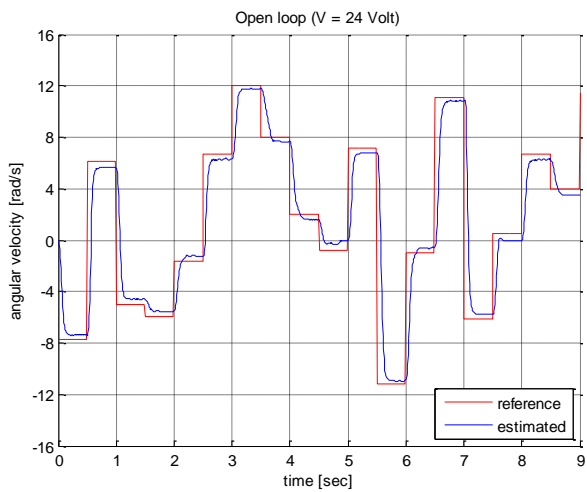
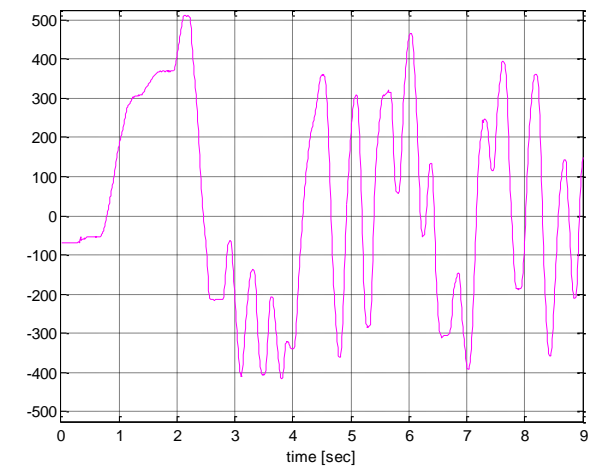
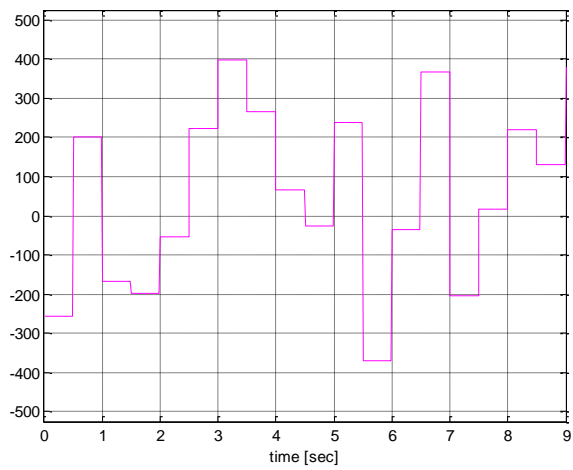
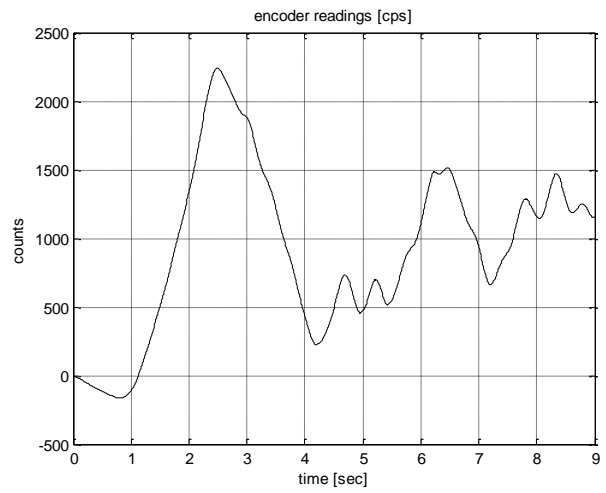




Τυχαία τροποποιούμενο κάθε 0.5 sec με βηματικές αλλαγές και αλλαγή της φοράς περιστροφής



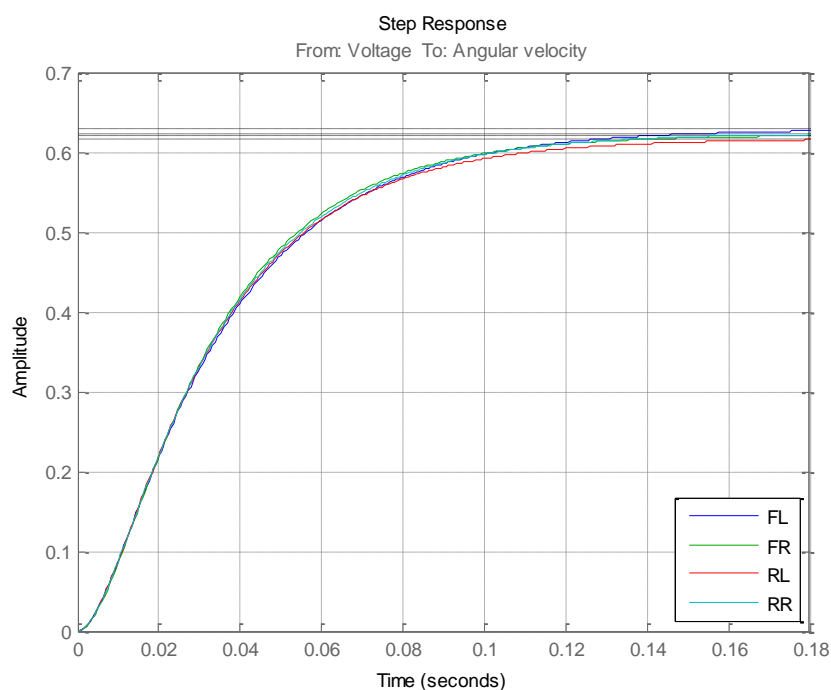
Εξωτερικό σήμα αναφοράς (ποτενσιόμετρο)



### 5.7.2 Συναρτήσεις μεταφοράς ανοικτού βρόχου

Το σύστημα θεωρήθηκε 2<sup>ης</sup> τάξης χωρίς μηδενικό και η συνάρτηση μεταφοράς του υπολογίστηκε με βάση τα πειραματικά δεδομένα (αρχεία \*.mat) και κατάλληλο script (sys\_ident.m) του Matlab που γράφτηκε για αυτό το σκοπό και παρατίθεται στο παράρτημα Ε της παρούσας. Η εκτίμηση της γωνιακής ταχύτητας στην προκειμένη περίπτωση για την εύρεση των συναρτήσεων μεταφοράς πραγματοποιήθηκε με απλή διαφόριση του διανύσματος θέσης.

Τροχός	Fit %	Συνάρτηση μεταφοράς $\frac{\Omega_L(s)}{V_m(s)}$	Στατικό κέρδος
Εμπρόσθιος αριστερός (FL)	87.86	$\frac{3105.8}{s^2 + 185s + 4938.8}$	0.6289
Εμπρόσθιος δεξιός (FR)	88.4	$\frac{2856.9}{s^2 + 163s + 4596.1}$	0.6216
Οπίσθιος αριστερός (RL)	88.67	$\frac{3024.2}{s^2 + 175.4s + 4903}$	0.6168
Οπίσθιος δεξιός (RR)	87.95	$\frac{2989.7}{s^2 + 173.3s + 4795.6}$	0.6234

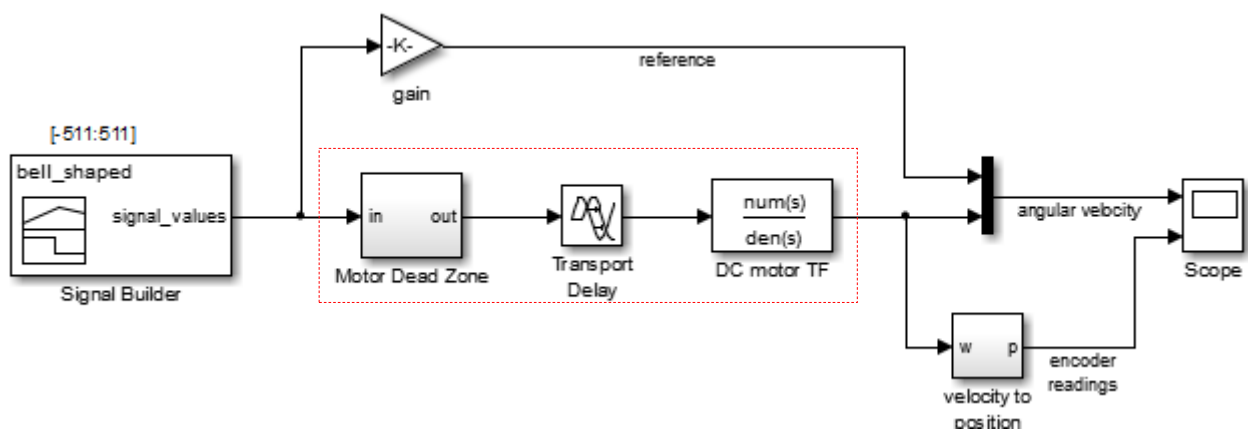


Εικόνα 5.20 Βηματικές αποκρίσεις των τεσσάρων πανομοιότυπων συστημάτων κινητήρα EMG49 - τροχού

### 5.7.3 Προσομοίωση συστήματος ανοικτού βρόγχου

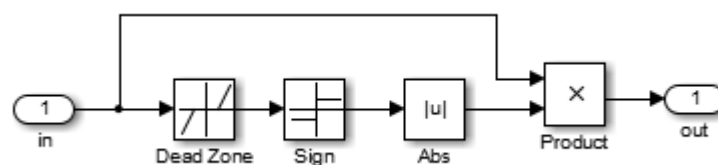
Το σύστημα dc κινητήρα-τροχού ανοικτού βρόγχου μοντελοποιήθηκε ως σύστημα δεύτερης τάξης με χρονική υστέρηση  $d$ . Αν και η χρονική καθυστέρηση που υπεισέρχεται στο μοντέλο από τον αλγόριθμο υπολογισμού της γωνιακής ταχύτητας είναι πολύ μικρή συμπεριλαμβάνεται σε αυτό καθώς η επίδραση της στον έλεγχο κλειστού βρόγχου είναι σημαντική αφού εισάγει επιπρόσθετη καθυστέρηση φάσης με συνέπεια την υποβάθμιση της μεταβατικής απόκρισης του συστήματος (υπερύψωση, εντονότερες ταλαντώσεις).

$$G(s) = \frac{\Omega_L(s)}{V(s)} = \frac{K e^{-ds}}{(\tau_1 s + 1)(\tau_2 s + 1)} = \frac{K}{\tau_1 \tau_2} e^{-ds} \frac{1}{\left(s + \frac{1}{\tau_1}\right)\left(s + \frac{1}{\tau_2}\right)}$$



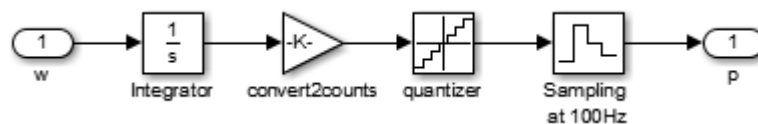
Εικόνα 5.21 Simulink μοντέλο του συστήματος ανοικτού βρόγχου

Για να βελτιωθεί περαιτέρω η ακρίβεια των προσομοιώσεων στο μοντέλο έχει συμπεριληφθεί και το υποσύστημα Motor Dead Zone στο οποίο έχει καταχωρηθεί το εύρος της νεκρής ζώνης οι τιμές του οποίου βρέθηκαν πειραματικά. Η εσωτερική του δομή είναι:



Εικόνα 5.22 Υποσύστημα Motor Dead Zone

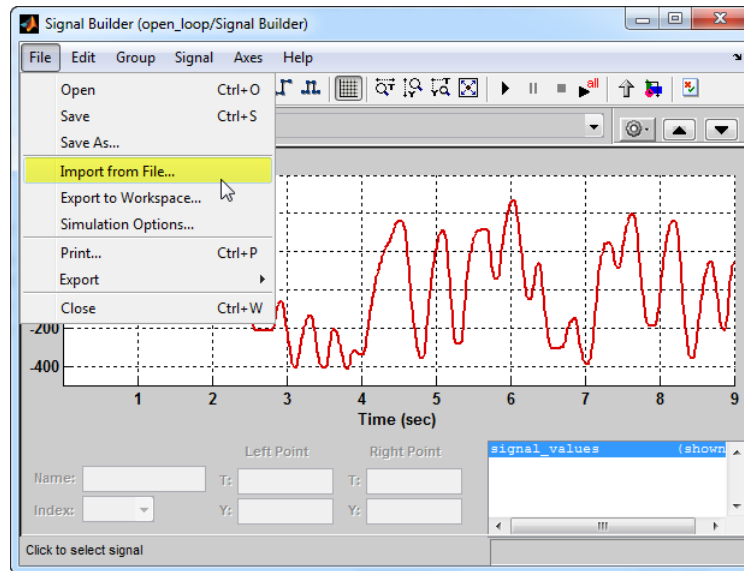
Παράγεται επίσης και για σύγκριση με την πραγματική ή τρέχουσα τιμή του μετρητή απαρίθμησης του κωδικοποιητή θέσης με το κάτωθι υποσύστημα:



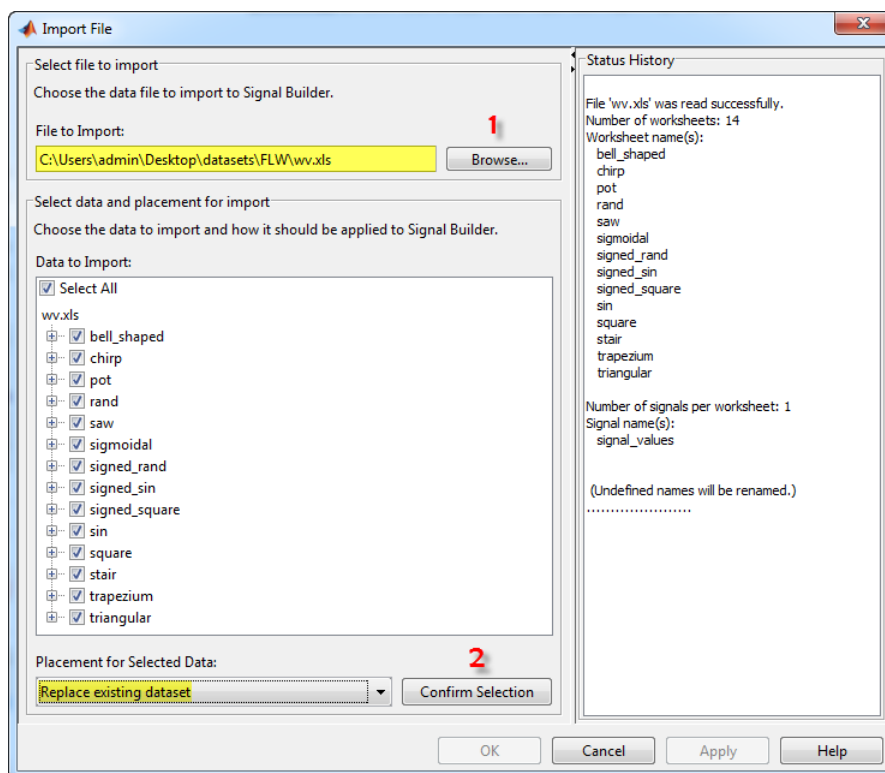
Εικόνα 5.23 Μετατροπή της γωνιακής ταχύτητας περιστροφής σε θέση

Το κέρδος για την μετατροπή είναι ίσο με  $980/(2\pi)$ .

Η παραγωγή των σημάτων αναφοράς γίνεται με τη βοήθεια του Signal Builder. Στη συγκεκριμένη βαθμίδα έχουν εισαχθεί μέσω αρχείου Excel οι τιμές των σημάτων αναφοράς που παρήχθησαν κατά τη διεξαγωγή των πειραμάτων. Το διατιθέμενο στο Παράρτημα E script xls4SB.m του Matlab προσπελαύνει τα αρχεία των πειραματικών δεδομένων \*.mat και δημιουργεί αυτοματοποιημένα το συγκεκριμένο αρχείο excel το οποίο περιέχει σε κάθε φύλλο εργασίας του το διάγραμμα των χρονικών στιγμών και τις αντίστοιχες τιμές του σήματος αναφοράς.



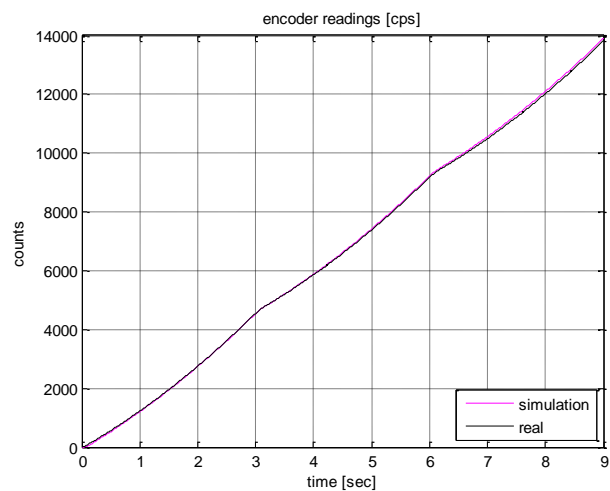
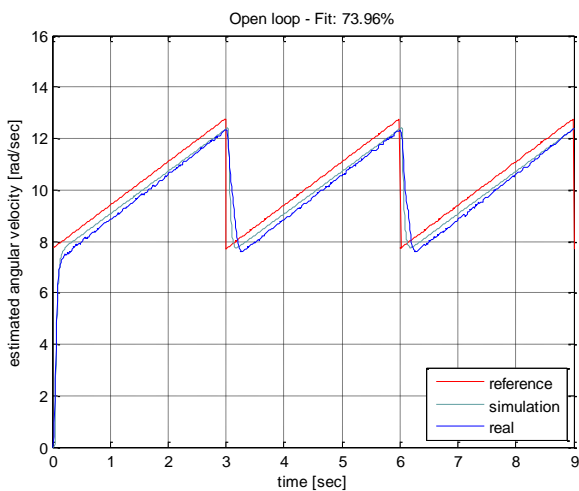
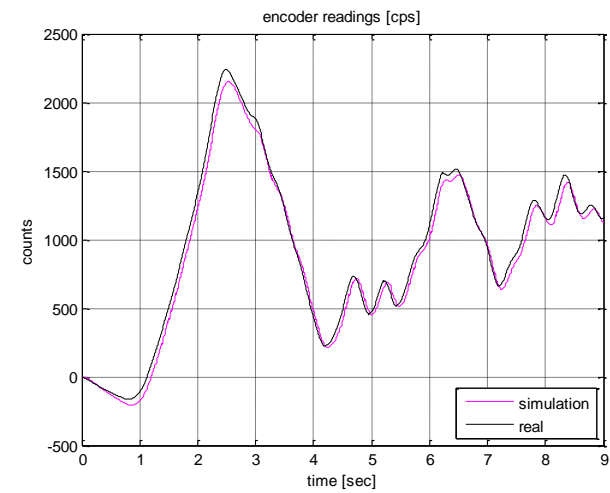
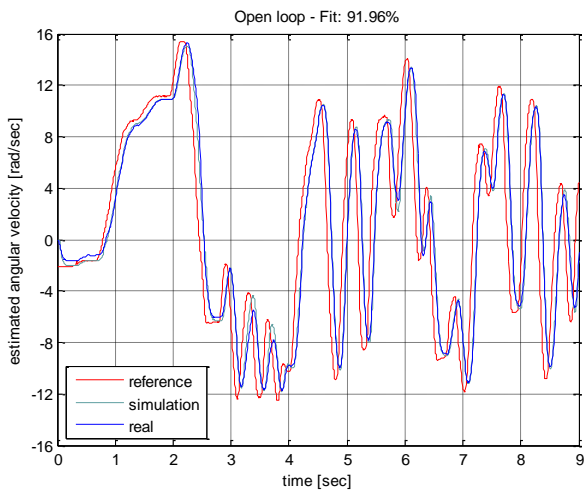
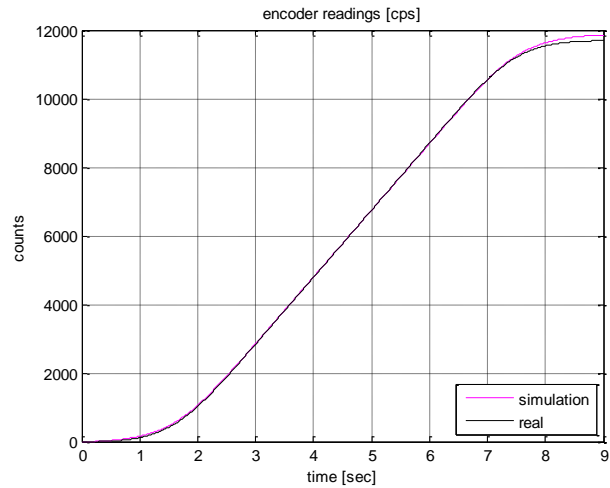
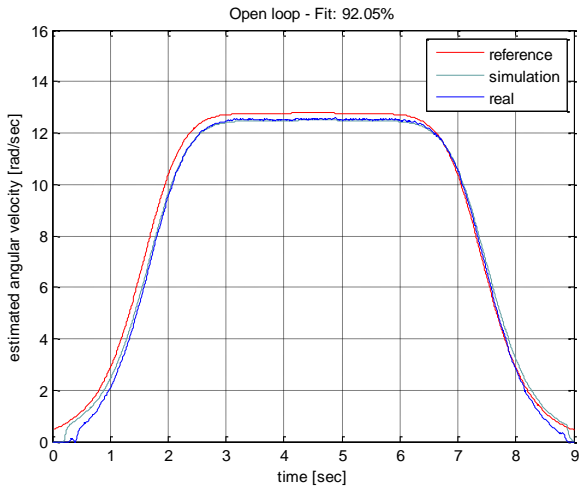
Εικόνα 5.24 Εισαγωγή αρχείου Microsoft Excel στον Signal Builder



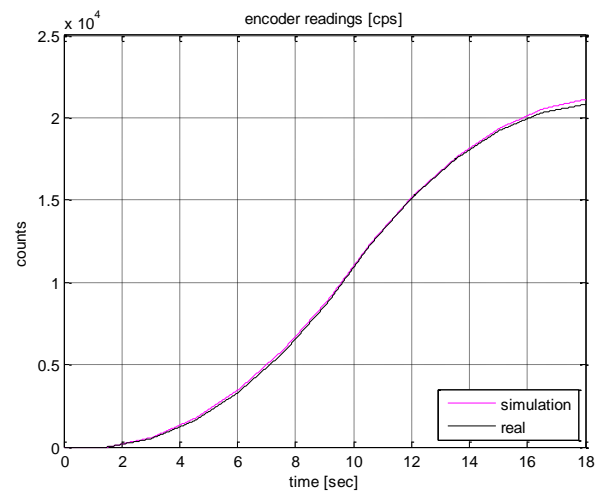
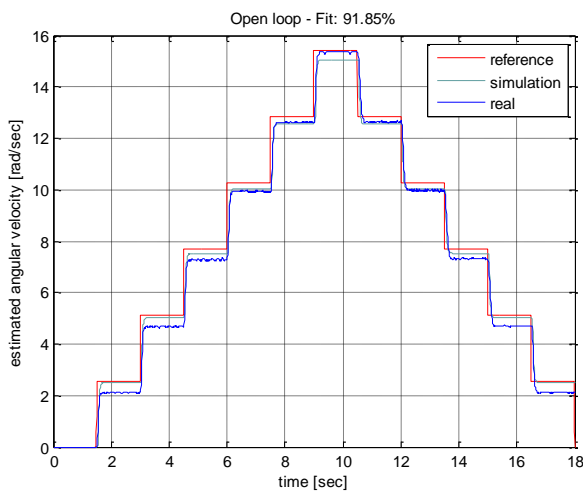
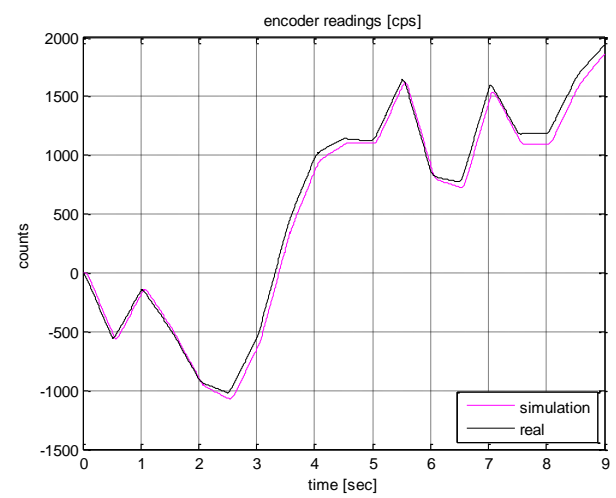
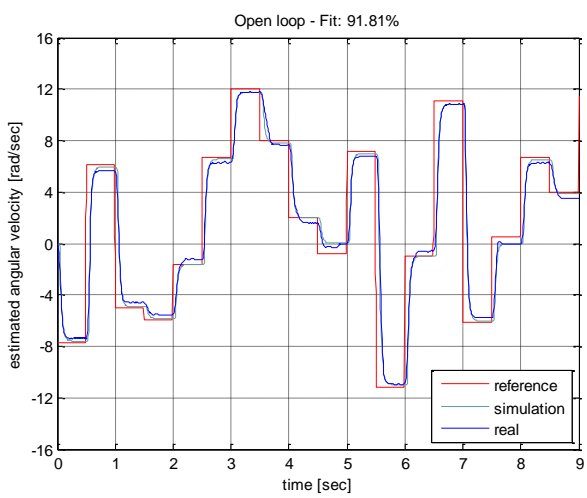
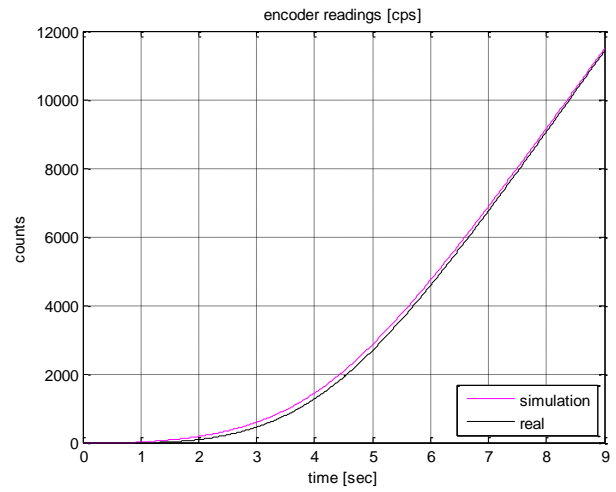
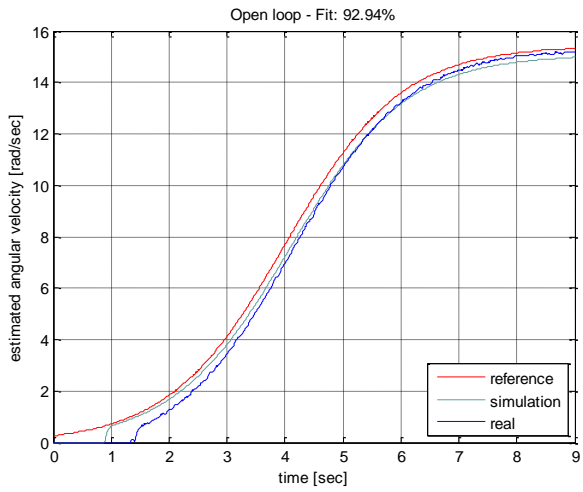
Εικόνα 5.25 1) Καθορισμός διαδρομής αρχείου Excel 2) Αντικατάσταση υπάρχοντος dataset

### 5.7.4 Αποτελέσματα προσομοίωσης

Για την σύγκριση της εξόδου του εκτιμώμενου γραμμικού μοντέλου με την έξοδο του πραγματικού παρήχθησαν τα κάτωθι διαγράμματα.



## Κεφάλαιο 5 - Έλεγχος κίνησης τροχών



## 5.8 Αναλογικός-ολοκληρωτικός-διαφορικός ελεγκτής

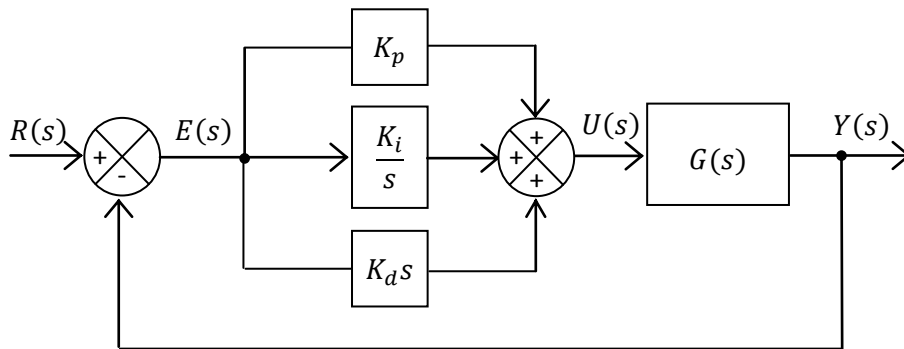
Ο PID ελεγκτής προέρχεται από τη σύνθεση τριών επιμέρους απλών ελεγκτών. Ο συγκεκριμένος ελεγκτής είναι κατάλληλος για έλεγχο συστημάτων με απότομες μεταβολές φορτίου συνδυάζει ταχεία δυναμική απόκριση και μηδενικό σφάλμα στη μόνιμη κατάσταση.

Στον PID έλεγχο ο αναλογικός όρος αντιδρά άμεσα στην ύπαρξη σφάλματος, η έξοδος του αναλογικού ελεγκτή είναι απευθείας ανάλογη του σφάλματος ελέγχου. Η σταθερά αναλογίας είναι το κέρδος του ελεγκτή. Ο αναλογικός όρος αδυνατεί όμως να εξαλείψει τελείως το σφάλμα μόνιμης κατάστασης.

Το μόνιμο σφάλμα του αναλογικού ελεγκτή εξαλείφεται με τον ολοκληρωτικό όρο. Επιπρόσθετα ο ολοκληρωτικός όρος αντιμετωπίζει με επιτυχία αποκλίσεις που προκαλούνται από διαταραχές του συστήματος. Η έξοδος του ολοκληρωτικού ελεγκτή μεταβάλλεται με ρυθμό που εξαρτάται από το μέγεθος του σφάλματος και από την τιμή της σταθεράς ολοκλήρωσης. Ο ολοκληρωτικός έλεγχος εισάγει στο σύστημα ένα επιπλέον πόλο στην αρχή των αξόνων του μιγαδικού επιπέδου. Η προσθήκη αυτή έχει ως αποτέλεσμα την αύξηση του τύπου του συστήματος κατά ένα το οποίο σημαίνει ότι για οποιαδήποτε βηματική διαταραχή το σφάλμα στη μόνιμη κατάσταση θα είναι μηδενικό. Ο μηδενισμός του σφάλματος γίνεται σε βάρος της ταχύτητας απόκρισης και της ευστάθειας του συστήματος.

Η έξοδος του διαφορικού ελεγκτή μεταβάλλεται ανάλογα με το χρονικό ρυθμό μεταβολής του σφάλματος ελέγχου. Ο διαφορικός όρος περιορίζει το σφάλμα κατά τη διάρκεια της μεταβατικής απόκρισης του συστήματος. Επιχειρεί την πρόληψη και καταστολή του σφάλματος πριν αυτό εκδηλωθεί επιβάλλοντας περιορισμό στην έξοδο του ελεγκτή σε απότομους ρυθμούς μεταβολής του σφάλματος.

### 5.8.1 Διακριτοποίηση και ψηφιακή υλοποίηση του PID ελεγκτή



Εικόνα 5.26 Διάγραμμα βαθμίδων PID ελεγκτή σε διάταξη μοναδιαίας αρνητικής ανάδρασης

Η σχέση εισόδου-εξόδου του PID ελεγκτή στο πεδίο του χρόνου είναι

$$v(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

ή

$$v(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right)$$

όπου

$K_p$  η σταθερά αναλογικού κέρδους  
 $K_i$  η σταθερά ολοκληρωτικού ελέγχου  
 $K_d$  το κέρδος του διαφορικού ελεγκτή

με  $T_i = K_p / K_i$  και  $T_d = K_d / K_p$  ο χρόνος ολοκλήρωσης και διαφορίσης αντίστοιχα.

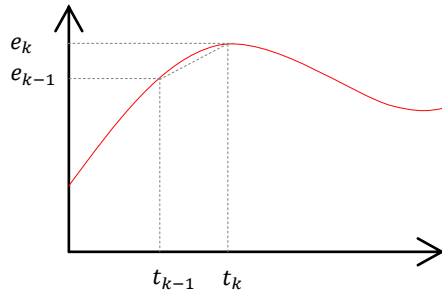
και μέσω του μετασχηματισμού Laplace γίνεται

$$U(s) = \left( K_p + \frac{K_i}{s} + K_d s \right) E(s)$$

Αν  $k$  ο δείκτης του τρέχοντος δείγματος ο αναλογικός όρος διακριτοποιείται ως εξής

$$P_k = K_p e_k$$

Ο ολοκληρωτικός όρος διακριτοποιείται με βάση τον κανόνα του τραπεζίου



Εικόνα 5.27 Διακριτοποίηση I-όρου

$$I_k = K_i \frac{T}{2} \sum_{n=1}^k (e_n + e_{n-1})$$

με  $T$  την περίοδο δειγματοληψίας ( $t_k = t_{k-1} + T$ )

Η διακριτοποίηση του διαφορικού όρου γίνεται με την μέθοδο των προς τα πίσω διαφορών.

$$D_k = K_d \frac{(e_k - e_{k-1})}{T}$$

Αθροίζοντας τους επιμέρους όρους προκύπτει

$$u_k = P_k + I_k + D_k = K_p e_k + K_i \frac{T}{2} \sum_{n=1}^k (e_n + e_{n-1}) + K_d \frac{(e_k - e_{k-1})}{T}$$

Για την έξοδο του ελεγκτή την χρονική στιγμή  $(k - 1)T$  προκύπτει:

$$u_{k-1} = K_p e_{k-1} + K_i \frac{T}{2} \sum_{n=1}^{k-1} (e_n + e_{n-1}) + K_d \frac{(e_{k-1} - e_{k-2})}{T}$$



Αφαιρώντας κατά μέλη τις σχέσεις για  $u_k$  και  $u_{k-1}$  θα είναι

$$u_k - u_{k-1} = K_p(e_k - e_{k-1}) + \frac{K_i T}{2}(e_k + e_{k-1}) + \frac{K_d}{T}(e_k - 2e_{k-1} + e_{k-2}) \Rightarrow$$

$$u_k - u_{k-1} = \left(K_p + \frac{K_i T}{2} + \frac{K_d}{2}\right)e_k + \left(\frac{K_i T}{2} - K_p - \frac{2K_d}{T}\right)e_{k-1} + \frac{K_d}{T}e_{k-2} \Rightarrow$$

$$u_k = u_{k-1} + A e_k + B e_{k-1} + C e_{k-2}$$

όπου

$$A = K_p + \frac{K_i T}{2} + \frac{K_d}{2}$$

$$B = \left(\frac{K_i T}{2} - K_p - \frac{2K_d}{T}\right)$$

$$C = \frac{K_d}{T}$$

### 5.8.2 Ρύθμιση ελεγκτή PID με τη μέθοδο των Ziegler- Nichols

Το πρόβλημα της σχεδίασης του ελεγκτή συνίσταται στην εύρεση (tuning) των κατάλληλων τιμών των παραμέτρων του ώστε το υπό έλεγχο σύστημα να λειτουργεί σύμφωνα με τις επιθυμητές προδιαγραφές. Για μια πρώτη εκτίμηση των τιμών τους εφαρμόστηκε η 2<sup>η</sup> μέθοδος των Ziegler-Nichols. Η μέθοδος βασίζεται στην παρατήρηση της βηματικής απόκριση της υπό έλεγχο διεργασίας. Με χρήση μόνο της αναλογικής βαθμίδας του ελεγκτή ( $K_i = 0, K_d = 0$ ) αυξάνεται σταδιακά ο αναλογικός όρος  $K_p$  έως την οριακή κρίσιμη τιμή  $K_{cr}$  για την οποία η έξοδος οδηγείται σε συντηρούμενες ταλαντώσεις (sustained oscillations) και μετράται η περίοδος  $T_{cr}$  των ταλαντώσεων. Με βάση την τιμή του κέρδους  $K_{cr}$  και την τιμή  $T_{cr}$  της περιόδου της ταλάντωσης οι προτεινόμενες από τους Ziegler-Nichols τιμές των παραμέτρων του ελεγκτή είναι:

$$K_p = \frac{3}{5} K_{cr} = 0.6 K_{cr}$$

$$K_i = \frac{2K_p}{T_{cr}}$$

$$K_d = \frac{K_p T_{cr}}{8} = 0.125 K_p T_{cr}$$

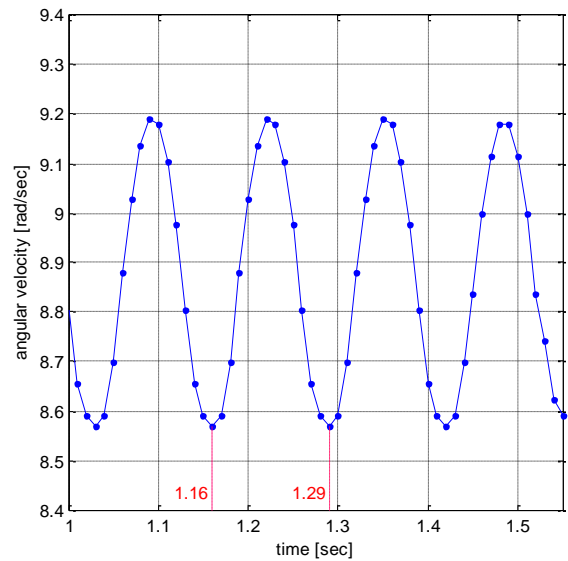
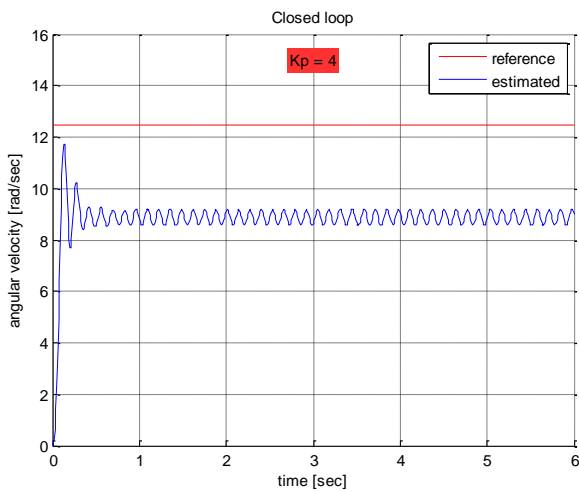
Για την εδώ εφαρμογή της μεθόδου καταγράφηκε η απόκριση του συστήματος με βηματική διέγερση  $\omega_d = 12.78 \text{ rad/s}$  (duty cycle=423/511) και μέγιστη τάση τροφοδοσίας 24V. Η κρίσιμη τιμή του αναλογικού κέρδους βρέθηκε  $K_{cr} = 4$  και η περίοδος των συντηρούμενων ταλαντώσεων υπολογίστηκε ίση με  $T_{cr} = 0.13 \text{ sec}$ . Συνεπώς οι τιμές των κερδών κάθε όρου του ελεγκτή θα είναι:

$$K_p = 2.4, K_i = 36.923, K_d = 0.039$$

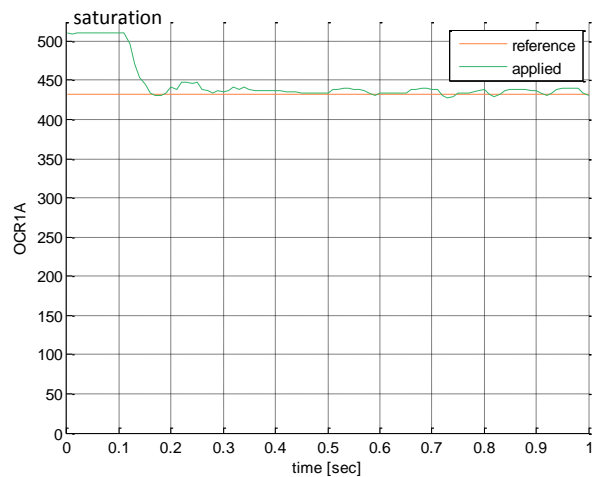
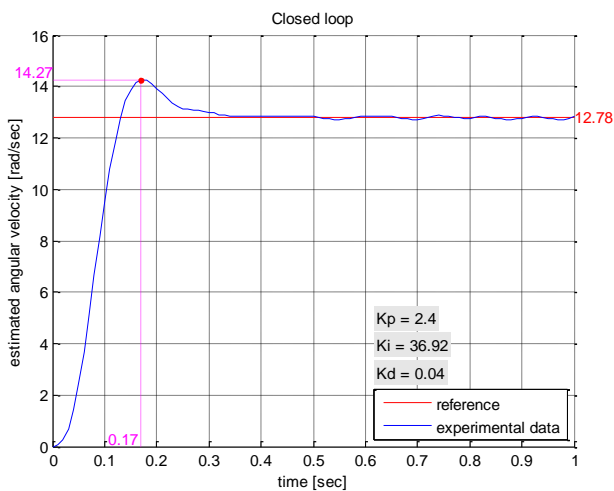
Η βηματική απόκριση του συστήματος κλειστού βρόχου με PID αντιστάθμιση παρουσιάζει ποσοστό υπερέψωσης 11.7%. Η χρήση του I-όρου στο νόμο ελέγχου υποβαθμίζει τη μεταβατική απόκριση του συστήματος αφού εμφανίζεται το φαινόμενο του ολοκληρωτικού κορεσμού (integral windup). Το σήμα ελέγχου οδηγείται στον κόρο λόγω της απότομης αλλαγής του σήματος αναφοράς με αποτέλεσμα η έξοδος του συστήματος να εκτρέπεται σημαντικά από την επιθυμητή τιμή. Για όσο χρόνο η έξοδος του ελεγκτή παραμένει κορεσμένη ο όρος ολοκλήρωσης συνεχίζει να αυξάνεται συσσωρεύοντας

σφάλμα και παρότι το σφάλμα μετέπειτα μειώνεται η έξοδος καθυστερεί να επανέλθει στη γραμμική περιοχή λειτουργίας. Αυτό έχει σαν αποτέλεσμα την υποβάθμιση της απόκρισης του συστήματος. Ο αλγόριθμος που εφαρμόστηκε στον μικροελεγκτή για την αντιμετώπιση του φαινομένου είναι:

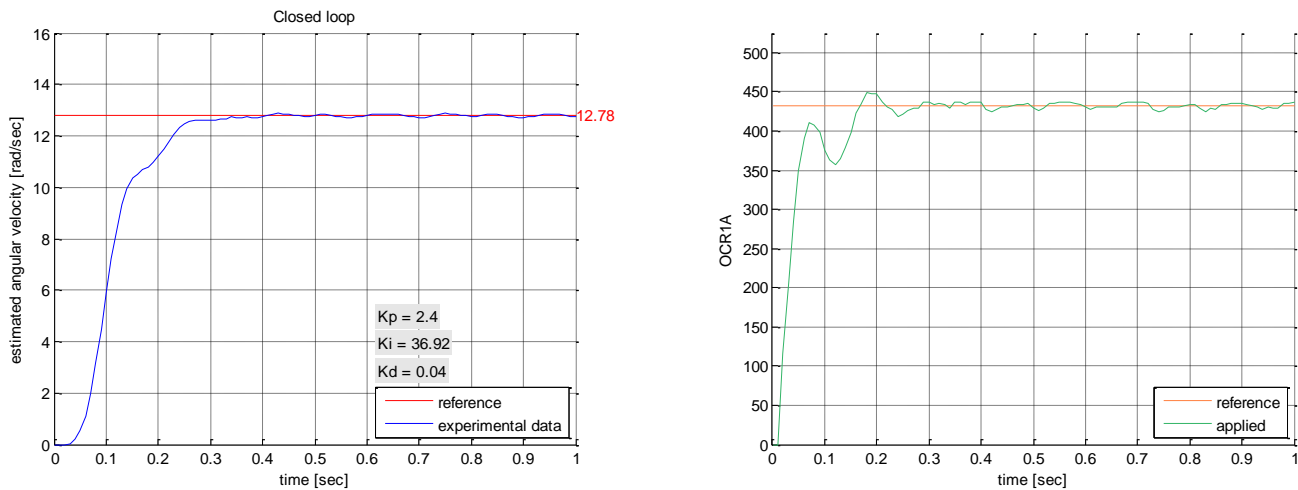
1. υπολογίζεται η τιμή του σήματος ελέγχου με βάση την υλοποίηση που δόθηκε προηγουμένως
2. αν η τιμή αυτή είναι εντός των επιτρεπόμενων ορίων γίνεται αποδεκτή
3. σε αντίθετη περίπτωση αν δηλαδή η τιμή του σήματος ελέγχου οδηγεί το σύστημα στον κόρο θέτεται η τρέχουσα τιμή του ίση με εκείνη που είχε προκύψει από τον προηγούμενο κύκλο ελέγχου.



Εικόνα 5.28 α) Συντηρούμενες ταλαντώσεις με αναλογικό κέρδος  $K_p = K_{cr} = 4$  β) Υπολογισμός περιόδου  $T_{cr}$  συντηρούμενων ταλαντώσεων

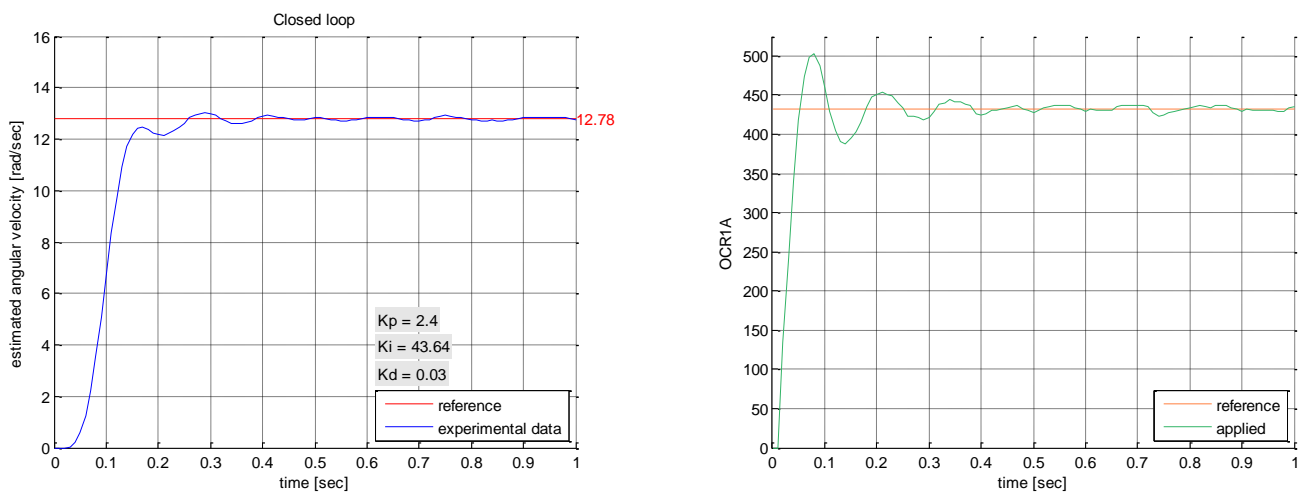


Εικόνα 5.29 Βηματική απόκριση κλειστού βρόχου συστήματος κινητήρα-τροχού με PID αντιστάθμιση

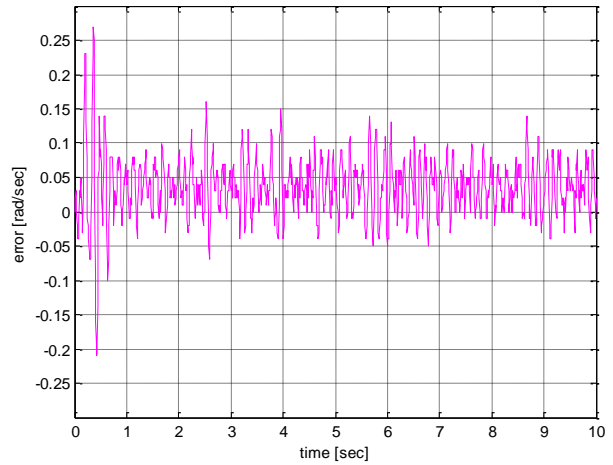
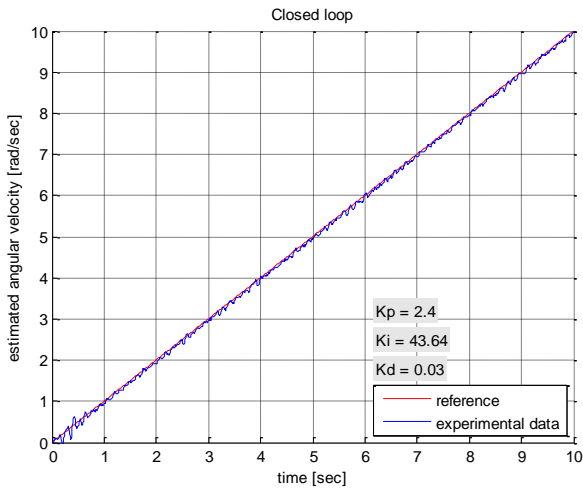


Εικόνα 5.30 Υλοποίηση του Integrator Anti-windup

Τα κέρδη του ελεγκτή επαναρυθμίστηκαν με βάση τη μέθοδο δοκιμής-σφάλματος και προσομοιώσεων στο Simulink. Αυξήθηκε το κέρδος του ολοκληρωτικού ελεγκτή ώστε να αντιμετωπίζονται οι αποκλίσεις που προκαλούνται από τυχόν διαταραχές (ανωφέρειες/κατωφέρειες, ανωμαλίες/εμπόδια στον αγρό). Ο συγκεκριμένος PID ελεγκτής ( $K_p = 2.4$ ,  $K_i = 43.64$ ,  $K_d = 0.03$ ) παρουσιάζει το μικρότερο σφάλμα στην παρακολούθηση του σήματος αναφοράς αλλά σε απότομες αλλαγές του εμφανίζει μη αποδεκτό ποσοστό υπερύψωσης κατά την μεταβατική απόκριση.

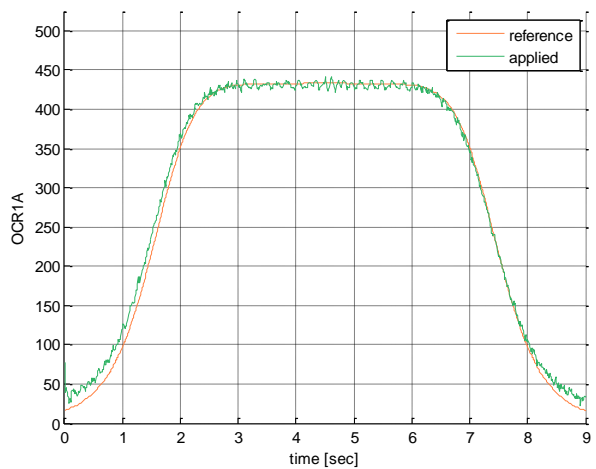
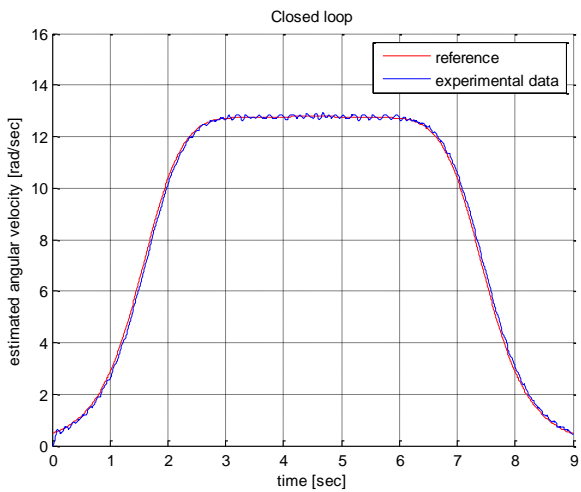
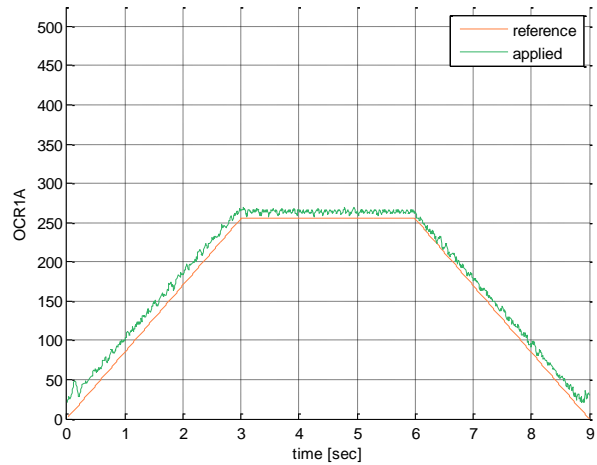
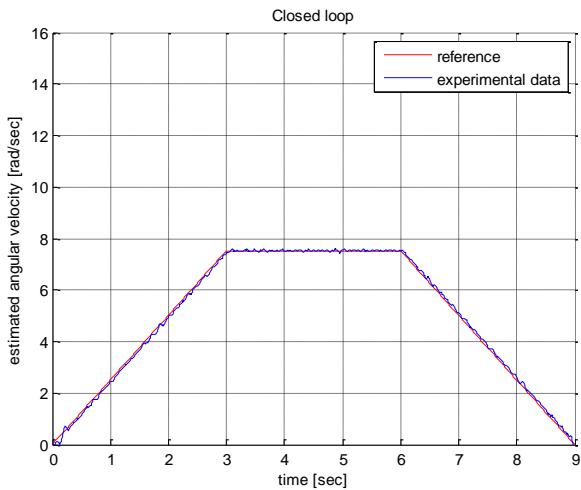


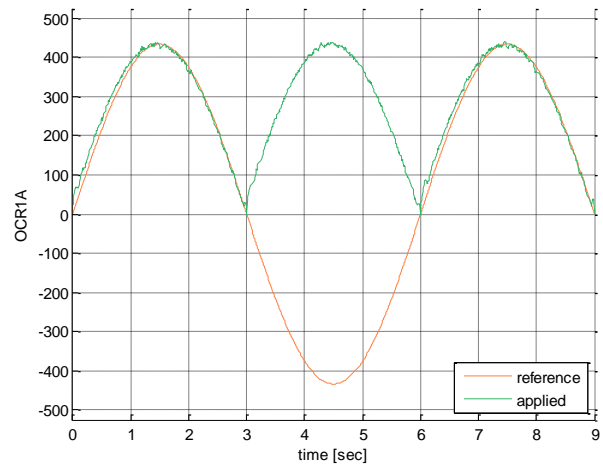
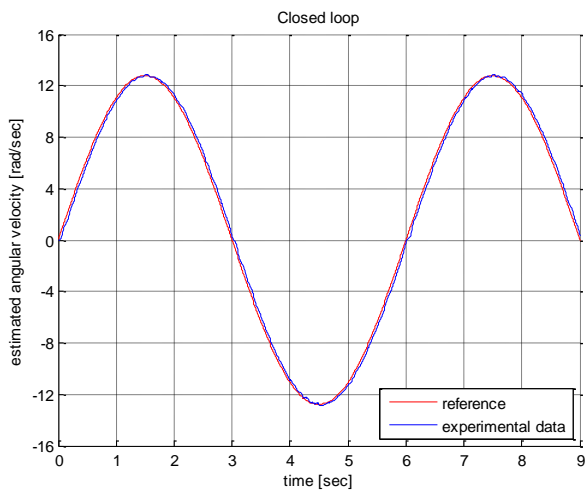
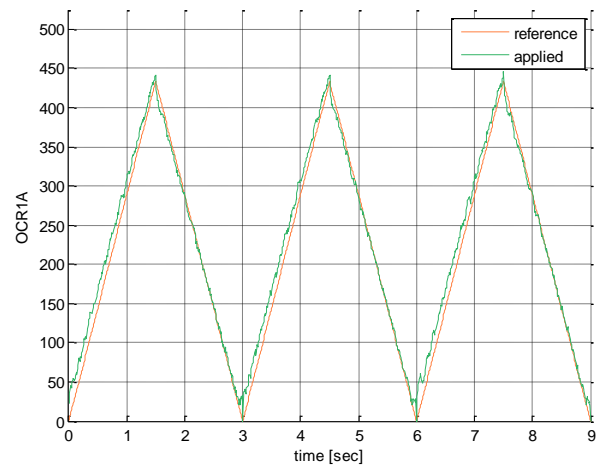
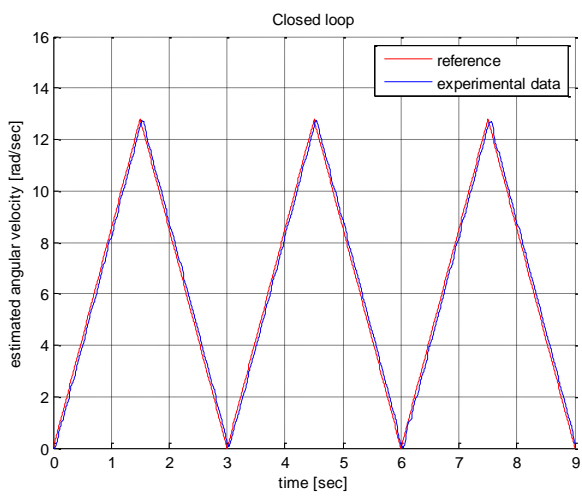
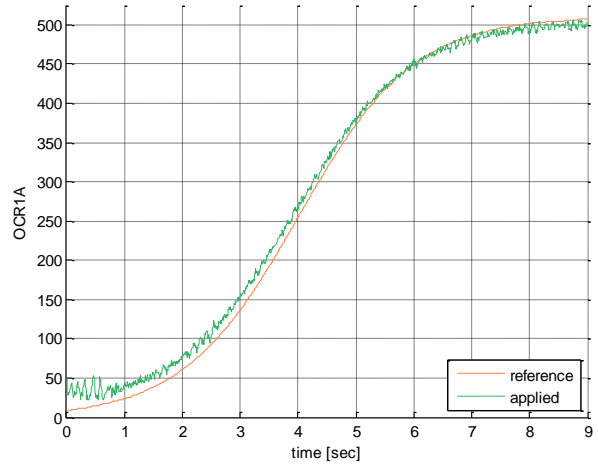
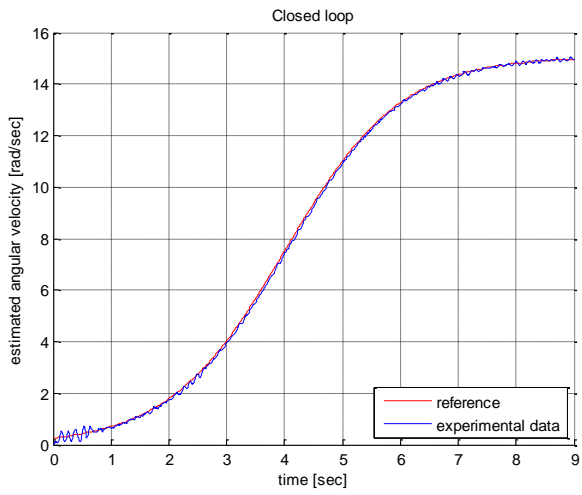
Εικόνα 5.31 Βηματική απόκριση κλειστού βρόχου συστήματος κινητήρα-τροχού με κέρδη PID ελεγκτή  $K_p = 2.4$ ,  $K_i = 43.64$ ,  $K_d = 0.03$  και συμπεριλαμβανόμενη την anti-windup τεχνική



**Εικόνα 5.32** Απόκριση αναρρίχησης του αντισταθμισμένου συστήματος κλειστού βρόγχου. Ένδειξη της ικανότητας παρακολούθησης του σήματος αναφοράς.

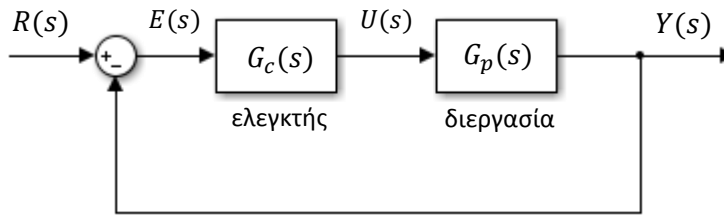
**Πειραματικά αποτελέσματα για τον έλεγχο ταχύτητας κλειστού βρόγχου με PID ελεγκτή**  
( $K_p=2.4, K_i=43.64, K_d=0.03$ )





### 5.8.3 Σχεδίαση ελεγκτή PID με την μέθοδο άμεσης σύνθεσης

Ως σημείο εκκίνησης για την ανάλυση θεωρούμε το κάτωθι διάγραμμα βαθμίδων



Εικόνα 5.33 Σύστημα αυτόματου ελέγχου κλειστού βρόγχου

Η συνάρτηση μεταφοράς του κλειστού βρόγχου είναι

$$\begin{aligned} Y(s) &= G_c(s)G_p(s)E(s) \Rightarrow Y(s) = G_c(s)G_p(s)(R(s) - Y(s)) \quad (1) \\ \Rightarrow Y(s) + G_c(s)G_p(s)Y(s) &= G_c(s)G_p(s)R(s) \Rightarrow \\ \frac{Y(s)}{R(s)} &= \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \end{aligned}$$

Λύνοντας την (1) ως προς  $G_c(s)$  προκύπτει μια έκφραση για τον αντισταθμιστή

$$G_c(s) = \frac{1}{G_p(s)} \frac{Y(s)}{R(s) - Y(s)} \Rightarrow G_c(s) = \frac{1}{G_p(s)} \cdot \frac{\frac{Y(s)}{R(s)}}{1 - \frac{Y(s)}{R(s)}}$$

Αντικαθιστώντας στην τελευταία σχέση την συνάρτηση μεταφοράς της πραγματικής διεργασίας  $G_p(s)$  με εκείνη του μοντέλου  $\tilde{G}_p(s)$ , το οποίο παρέχει μια προσέγγιση της συμπεριφοράς της πραγματικής διεργασίας και την άγνωστη συνάρτηση μεταφοράς του κλειστού βρόγχου με μια επιθυμητή συνάρτηση μεταφοράς κλειστού βρόγχου προκύπτει:

$$G_c = \frac{1}{\tilde{G}_p} \cdot \frac{\left(\frac{Y}{R}\right)_d}{1 - \left(\frac{Y}{R}\right)_d} \quad (2)$$

Αν  $(Y/R)_d = 1$  τότε η ρυθμιζόμενη μεταβλητή ακολουθεί τις μεταβολές της επιθυμητής τιμής χωρίς σφάλμα. Η ιδανική αυτή περίπτωση δεν μπορεί να επιτευχθεί με αντιστάθμιση με ανατροφοδότηση αφού ο παρανομαστής της συνάρτησης μεταφοράς του ελεγκτή μηδενίζεται.

Αν η συνάρτηση μεταφοράς της διεργασίας περιέχει μια γνωστή χρονική υστέρηση  $\theta$  μια λογική επιλογή για την επιθυμητή συνάρτηση μεταφοράς του κλειστού βρόγχου είναι ένα γραμμικό σύστημα πρώτης τάξης με χρονική καθυστέρηση  $\theta$ , η συνάρτηση μεταφοράς του οποίου θα είναι της μορφής:

$$\left(\frac{Y}{R}\right)_d = \frac{e^{-\theta s}}{\tau s + 1} \quad (3)$$

όπου  $\tau$  είναι η επιθυμητή σταθερά χρόνου του συστήματος κλειστού βρόχου. Ο όρος της χρονικής υστέρησης είναι απαραίτητος γιατί είναι αδύνατον η ρυθμιζόμενη μεταβλητή να ανταποκρίνεται στις αλλαγές της επιθυμητής τιμής σε χρόνο  $t = 0$  πριν  $t = \theta$ . Ο συνδυασμός των (2) και (3) δίνει:

$$G_c = \frac{1}{\tilde{G}_p} \frac{\frac{e^{-\theta s}}{\tau s + 1}}{1 - \frac{e^{-\theta s}}{\tau s + 1}} \Rightarrow G_c = \frac{1}{\tilde{G}_p} \frac{e^{-\theta s}}{\tau s + 1 - e^{-\theta s}}$$

Προσεγγιστικά είναι  $e^{-\theta s} \simeq 1 - \theta s$  (πρώτοι όροι της σειράς Taylor). Οπότε

$$G_c = \frac{1}{\tilde{G}_p} \frac{e^{-\theta s}}{\tau s + 1 - 1 + \theta s} \Rightarrow G_c = \frac{1}{\tilde{G}_p} \frac{e^{-\theta s}}{(\tau + \theta)s} \quad (4)$$

Η χρονική υστέρηση στον αριθμητή δεν είναι απαραίτητο να προσεγγιστεί γιατί απαλείφεται από την  $\tilde{G}_p$ . Η μελετώμενη διεργασία είναι δεύτερη τάξης με χρονική υστέρηση με συνάρτηση μεταφοράς:

$$\tilde{G}_p(s) = \frac{K e^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

Αντικαθιστώντας την στην (4) παράγεται ένα PID ελεγκτής

$$G_c(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{K e^{-\theta s}} \frac{e^{-\theta s}}{(\tau + \theta)s} \Rightarrow G_c(s) = \frac{\tau_1 \tau_2 s^2 + (\tau_1 + \tau_2)s + 1}{K(\tau + \theta)s} \Rightarrow$$

$$G_c(s) = \frac{\tau_1 + \tau_2}{K(\tau + \theta)} + \frac{1}{K(\tau + \theta)} \frac{1}{s} + \frac{\tau_1 \tau_2}{K(\tau + \theta)} s \Rightarrow G_c(s) = K_p + \frac{K_i}{s} + K_d s$$

όπου

$$K_p = C(\tau_1 + \tau_2), \quad K_i = C, \quad K_d = C\tau_1\tau_2$$

με

$$C = \frac{1}{K(\tau + \theta)}$$

Η σταθερά χρόνου  $\tau$  του συστήματος κλειστού βρόχου αποτελεί παράμετρο σχεδίασης και μπορεί να κάνει τον ελεγκτή πιο επιθετικό με μικρό  $\tau$  ή περισσότερο συντηρητικό με μεγάλο  $\tau$ .

### 5.8.3.1 Εφαρμογή της μεθόδου στο σύστημα του εμπρόσθιου αριστερού τροχού

Η συνάρτηση μεταφοράς του είναι

$$G(s) = \frac{511}{24} * e^{-0.04s} * \frac{145.9}{s^2 + 185s + 4939} \Rightarrow G(s) = \frac{3150.8 * e^{-0.04s}}{(s + 32.4) * (s + 152.7)} \Rightarrow$$

$$G(s) = \frac{3150.8e^{-0.04s}}{32.4 * 152.7 * \left(\frac{1}{32.4}s + 1\right) \left(\frac{1}{152.7}s + 1\right)} \Rightarrow$$

$$G(s) = 0.63 \frac{e^{-0.04s}}{\left(\frac{1}{32.4}s + 1\right) \left(\frac{1}{152.7}s + 1\right)}$$

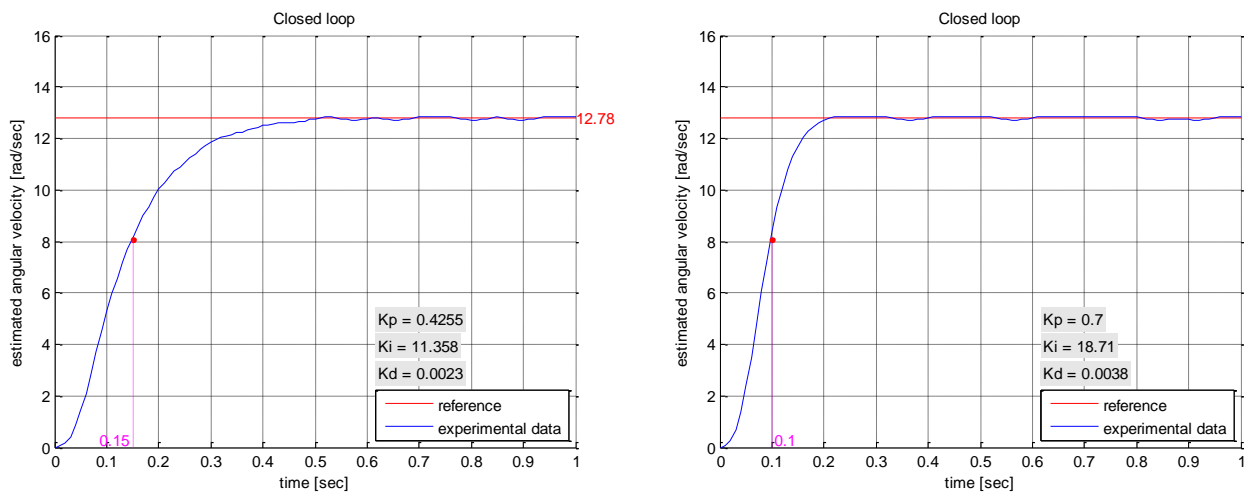
με  $K = 0.63$ ,  $\tau_1 = 1/32.4$  και  $\tau_2 = 1/152.7$

Αν η σταθερά χρόνου  $\tau$  τεθεί ίση με  $0.1 \text{ sec}$  τα κέρδη του PID ελεγκτή είναι:

$$K_p = 0.4255, K_i = 11.358, K_d = 0.0023$$

Αν τεθεί  $\theta = 0.035 \text{ sec}$  και  $\tau = 0.05 \text{ sec}$  τότε τα κέρδη του PID αντισταθμιστή είναι:

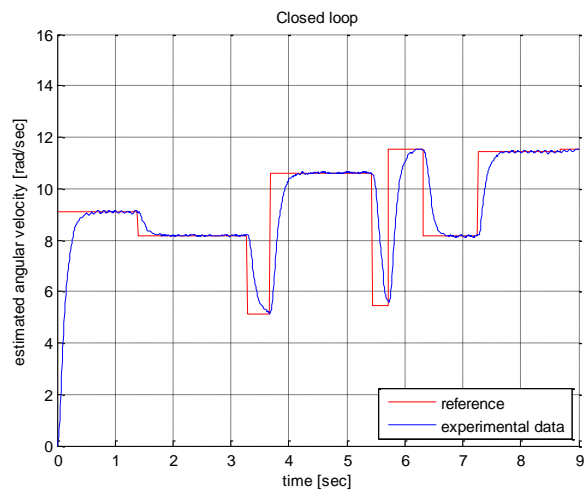
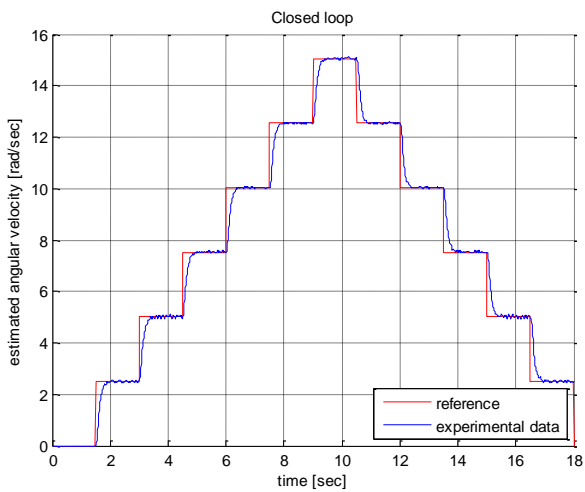
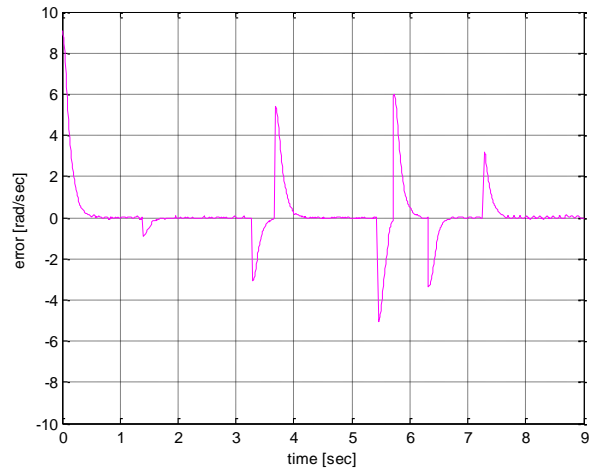
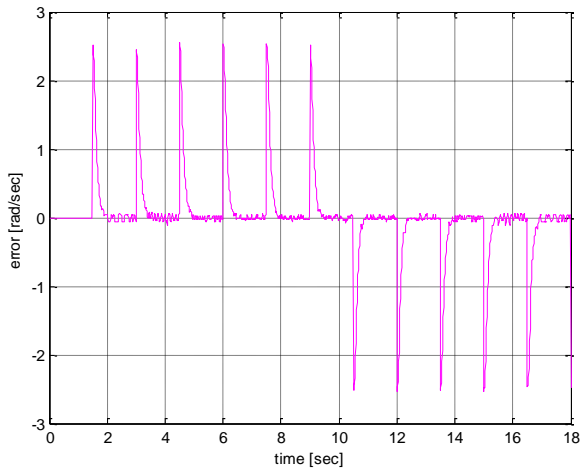
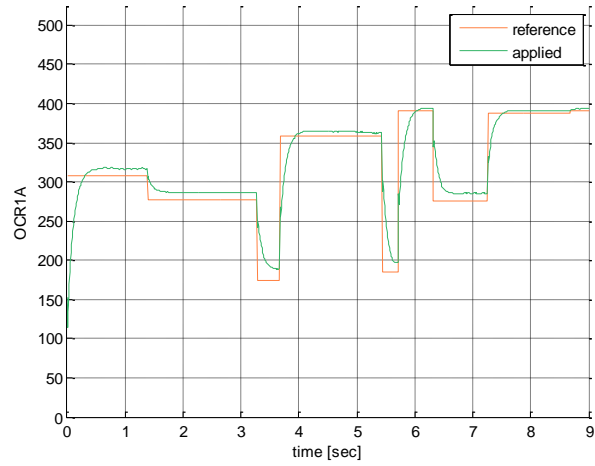
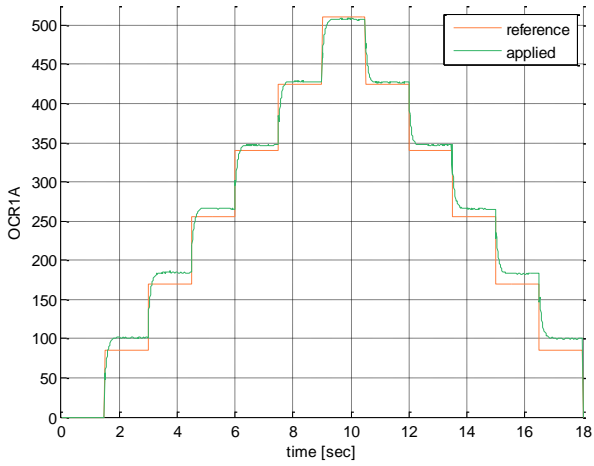
$$K_p = 0.7, K_i = 18.71, K_d = 0.0038$$



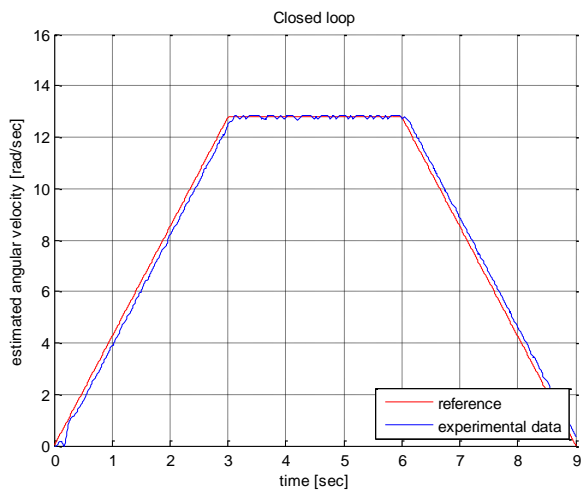
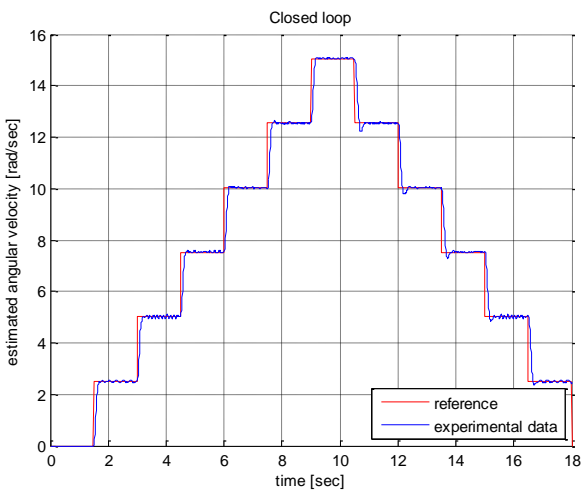
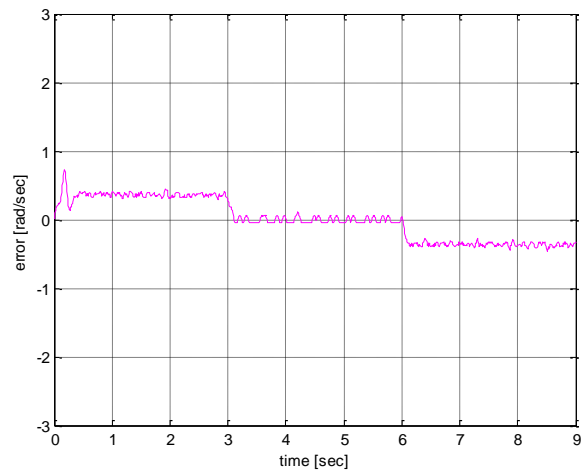
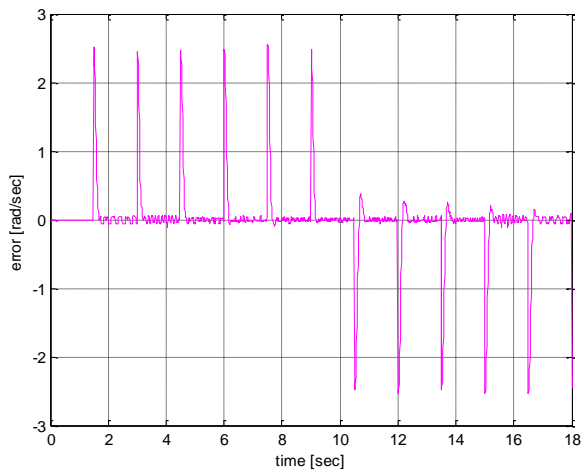
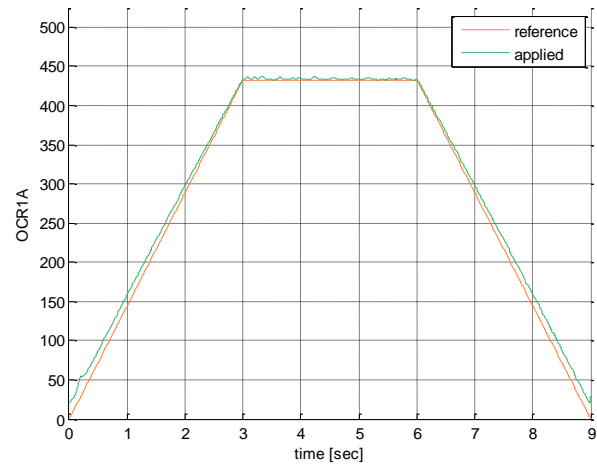
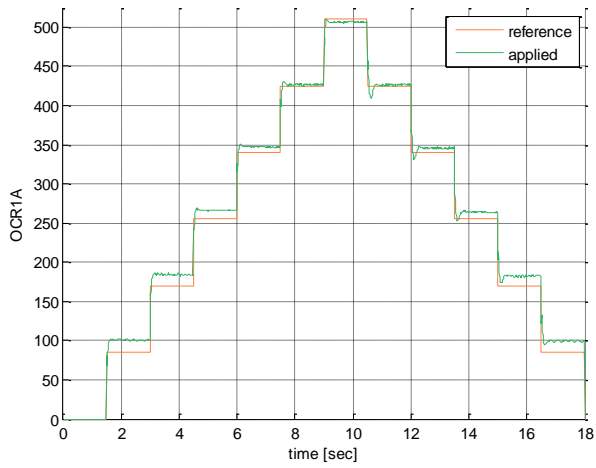
**Εικόνα 5.34** Βηματική απόκριση κλειστού βρόχου συστήματος κινητήρα-τροχού με τα κέρδη των PID ελεγκτών υπολογιζόμενα με βάση την μέθοδο άμεσης σύνθεσης



Πειραματικά αποτελέσματα για τον έλεγχο ταχύτητας κλειστού βρόχου με τα κέρδη του PID ελεγκτή υπολογιζόμενα με την μέθοδο άμεσης σύνθεσης ( $K_p=0.4255$ ,  $K_i=11.358$ ,  $K_d=0.0023$ )



Πειραματικά αποτελέσματα για τον έλεγχο ταχύτητας κλειστού βρόγχου με τα κέρδη του PID ελεγκτή υπολογιζόμενα με την μέθοδο άμεσης σύνθεσης ( $K_p=0.7$ ,  $K_i=18.71$ ,  $K_d=0.0038$ )

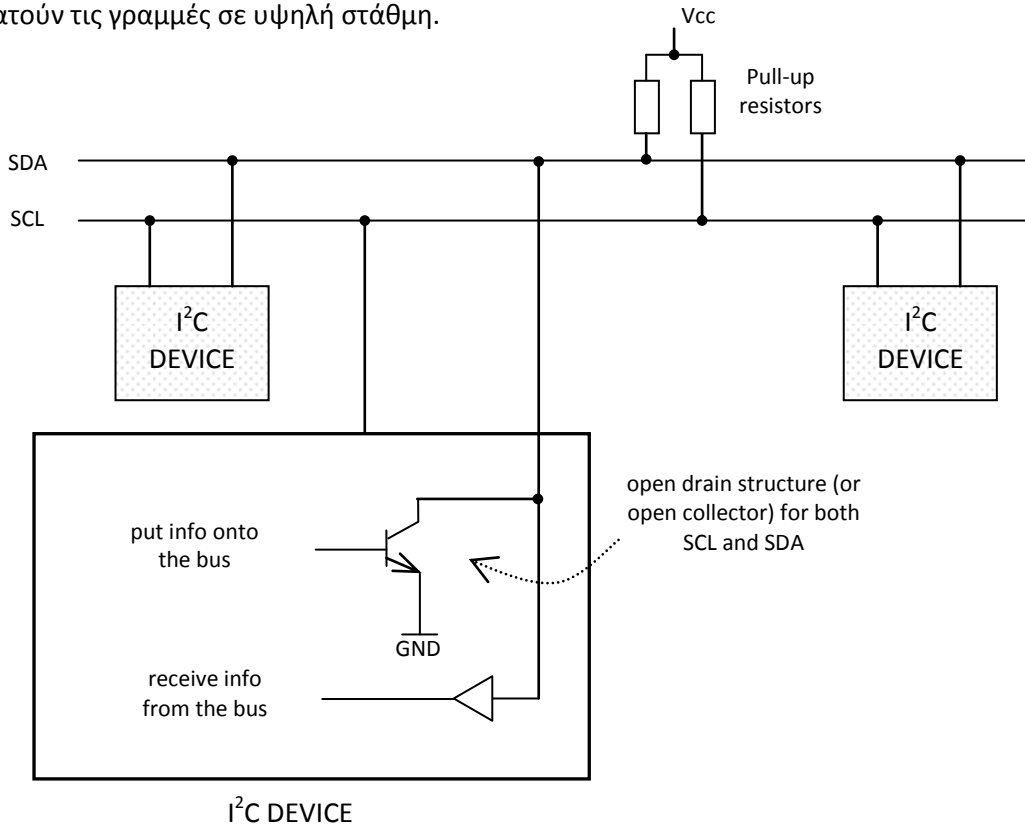


## 5.9 Ο διάυλος I<sup>2</sup>C

Ο διάυλος I<sup>2</sup>C (Inter-Integrated Circuit) είναι ένας σειριακός διάυλος και χρησιμοποιείται συνήθως για την διασύνδεση περιφερειακών συσκευών μικρής ταχύτητας σε μητρικές πλακέτες, ενσωματωμένα συστήματα και άλλες ηλεκτρονικές συσκευές. Ο διάυλος δεν χρησιμοποιείται μόνο για την επικοινωνία συσκευών που βρίσκονται πάνω σε ένα τυπωμένο κύκλωμα αλλά και για την επικοινωνία συσκευών που συνδέονται με καλώδια.

Χρησιμοποιεί μόνο δυο γραμμές μεταφοράς αμφίδρομης κατεύθυνσης, την γραμμή SCL (Serial **C**lock) που είναι η γραμμή ρολογιού και την γραμμή SDA (Serial **D**ata) που είναι η γραμμή δεδομένων. Οι γραμμές είναι τύπου ανοικτού συλλέκτη (open collector) ή ανοικτού απαγωγού (open drain).

Για τη διασύνδεση πολλαπλών εξόδων ψηφιακών κυκλωμάτων σε κοινή γραμμή λύση αποτελεί η χρήση ακροδεκτών ανοικτού συλλέκτη. Χρησιμοποιείται μια αντίσταση ανύψωσης δυναμικού (pull-up) προς την τάση τροφοδοσίας και η κοινή γραμμή θα βρίσκεται σε υψηλή στάθμη μόνον όταν όλες οι έξοδοι είναι σε υψηλή κατάσταση. Εάν έστω και μια έξοδος βρεθεί σε χαμηλή στάθμη όλη η γραμμή θα βρεθεί σε χαμηλή κατάσταση (wired-AND). Όταν λοιπόν ο διάυλος δεν χρησιμοποιείται οι αντιστάσεις pull-up κρατούν τις γραμμές σε υψηλή στάθμη.

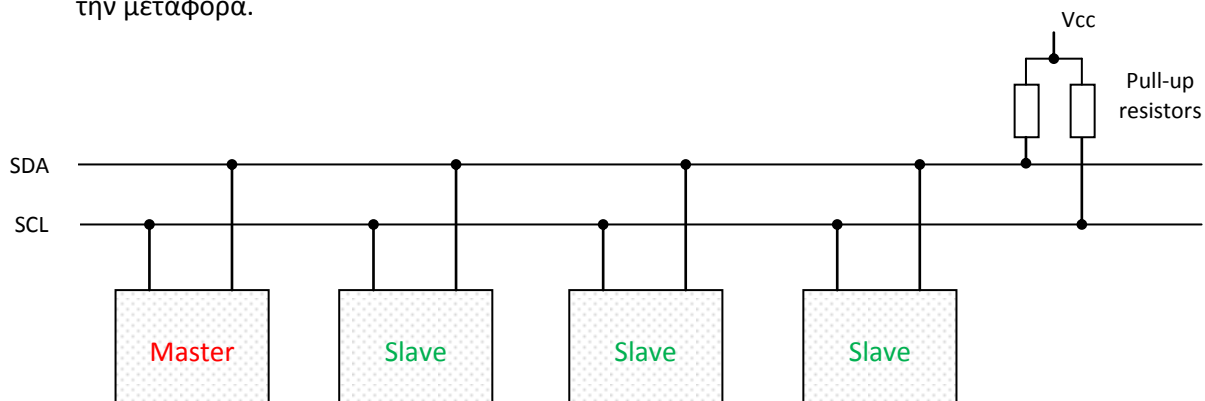


Εικόνα 5.35 Διάυλος I<sup>2</sup>C - χρήση ακροδεκτών ανοικτού συλλέκτη

Κάθε μια από τις γραμμές SCL και SDA συνδέονται μια αντίσταση στην γραμμή τροφοδοσίας. Συνηθισμένες τιμές αντιστάσεων είναι από 2KΩ έως 10KΩ. Η τιμή των αντιστάσεων δεν είναι κρίσιμη αλλά μαζί με την χωρητικότητα του διαύλου επηρεάζει την μέγιστη ταχύτητα λειτουργίας. Μεγάλες χωρητικότητες του διαύλου μπορούν να αντισταθμιστούν με μικρές αντιστάσεις. Οι αντιστάσεις αυτές δεν μπορεί να απουσιάζουν γιατί τότε οι γραμμές θα είναι μόνιμως σε κατάσταση λογικού 0 και ο διάυλος δεν θα δουλεύει.

Η standard συχνότητα ρολογιού για τον δίαυλο I<sup>2</sup>C είναι 100KHz ενώ μπορεί να λειτουργήσει και σε υψηλότερες (Fast mode: 400KHz, High speed mode: 3.4MHz). Τα κατώφλια του λογικού 0 και 1 τοποθετούνται στο 30% και το 70% της τάσης τροφοδοσίας αντίστοιχα. Συνεπώς μια τάση στην περιοχή 0-0.3Vcc θεωρείται λογικό 0 (low) ενώ μια τάση στην περιοχή 0.7Vcc έως Vcc θεωρείται λογικό 1 (high). Ο δίαυλος είναι ανεκτικός στον θόρυβο και δεν επηρεάζεται από απότομες μεταβολές της τάσης.

Οι συσκευές στον δίαυλο είναι είτε masters είτε slaves. Η master συσκευή είναι αυτή που ελέγχει και οδηγεί τη γραμμή ρολογιού. Οι slaves συσκευές είναι αυτές που ανταποκρίνονται στις συσκευές master. Μια συσκευή slave δεν μπορεί να ξεκινήσει μια μεταφορά πάνω στο δίαυλο, μόνο μια master μπορεί. Σε έναν δίαυλο μπορεί να είναι συνδεδεμένες πολλές συσκευές ως master και πολλές ως slaves. Και οι masters και οι slaves μπορούν να μεταφέρουν δεδομένα στον δίαυλο αλλά μόνο οι master συσκευές ελέγχουν την μεταφορά.

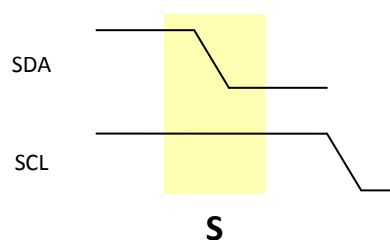


Εικόνα 5.36 Δίαυλος I<sup>2</sup>C - διασύνδεση συσκευών

Ο μέγιστος αριθμός κόμβων που μπορούν να συνδεθούν στον δίαυλο περιορίζεται από τον αριθμό των διαθέσιμων διευθύνσεων αλλά και από τη συνολική χωρητικότητα του διαύλου η οποία για το standard mode δεν πρέπει να υπερβαίνει τα 400pF, το οποίο περιορίζει και τις αποστάσεις επικοινωνίας. Στην πράξη μπορούμε να πετύχουμε μεγαλύτερα μήκη διαύλου μειώνοντας την ταχύτητα. Το πρωτόκολλο υποστηρίζει την ταυτόχρονη συνύπαρξη και άλλων master στον δίαυλο χωρίς να υπάρχουν συγκρούσεις δεδομένων ή απώλειες.

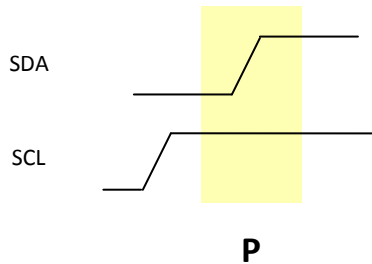
Σε όλες τις slave συσκευές που συνδέονται στον δίαυλο αποδίδεται ένας αριθμός ως διεύθυνση. Οι master συσκευές δεν είναι απαραίτητο να έχουν διεύθυνση εκτός αν υπάρχουν πολλές master στον δίαυλο (multi-master). Οι master συσκευές επικοινωνούν με τις slave χρησιμοποιώντας τη διεύθυνση της. Οι διευθύνσεις των συσκευών του διαύλου είναι είτε 7 bit (θεωρητικά έως 128 συσκευές στο δίαυλο), είτε 10 bit (1024 συσκευές).

Όταν μια master συσκευή επιθυμεί να επικοινωνήσει με μια slave ξεκινά στέλνοντας ένα start bit σηματοδοτώντας την εκκίνηση μεταφοράς δεδομένων. Σε multi-master περιβάλλον η κατάσταση αυτή σηματοδοτείται από εκείνον τον master που θα εκπέμψει πρώτος. Ο master θέτει την γραμμή SDA από υψηλή σε χαμηλή στάθμη ενόσω η γραμμή ρολογιού είναι σε κατάσταση λογικού 1. Ο δίαυλος θεωρείται ότι είναι απασχολημένος μετά από μια ακολουθία έναρξης (Start).



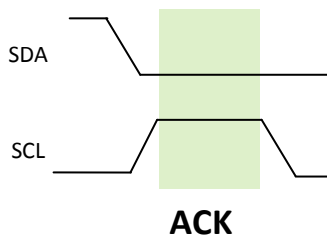
Εικόνα 5.37 Ακολουθία έναρξης

Η ακολουθία λήξης (stoP) σηματοδοτεί το τέλος μιας μεταφοράς δεδομένων σε μια slave συσκευή. Μαζί με την κατάσταση Start έχει την μεγαλύτερη προτεραιότητα και μπορεί να εκπεμφθεί σε οποιαδήποτε χρονική στιγμή απελευθερώνοντας τον δίαυλο ώστε να μπορούν άλλες συσκευές να τον χρησιμοποιήσουν. Ο master που θα εκπέμψει την ακολουθία λήξης θα πρέπει να θέσει την γραμμή SDA από χαμηλή σε υψηλή στάθμη ενόσω η γραμμή ρολογιού SCL θα είναι σε υψηλή. Όταν ολοκληρωθεί η stop κατάσταση οι γραμμές SDA και SCL θα είναι και οι δύο σε υψηλή στάθμη και ο δίαυλος μπαίνει σε αδρανή (idle) κατάσταση. Η repeated start (επανεναρξη) κατάσταση επιτρέπει στον master να δηλώσει μια ακολουθία έναρξης χωρίς να στείλει stop bit οπότε δεν χάνεται ο έλεγχος του διαύλου επιτρέποντας έτσι την αλλαγή συσκευής μετάδοσης των δεδομένων.



Εικόνα 5.38 Ακολουθία λήξης

Τα δεδομένα μεταφέρονται σε ακολουθίες των 8 bit. Για κάθε byte δεδομένων που μεταφέρονται η συσκευή που τα λαμβάνει στέλνει πίσω ένα low bit επιβεβαίωσης (**ACK**nowledgment) ότι το έλαβε και ότι είναι έτοιμη να λάβει το επόμενο byte. Έτσι για κάθε μεταφορά οκτώ δυαδικών ψηφίων απαιτούνται εννέα παλμοί ρολογιού. Αν το byte των δεδομένων είναι το τελευταίο της μετάδοσης ή ο δέκτης δεν μπορεί να λάβει περαιτέρω δεδομένα στέλνει ένα high bit μη επιβεβαίωσης (**Not ACK**nowledgment). Τα δεδομένα επιβεβαιώνονται όταν ο δέκτης σπρώχνει την γραμμή SDA στην χαμηλή στάθμη στην διάρκεια ενός παλμού του ρολογιού.



Εικόνα 5.39 Bit επιβεβαίωσης

Για να ξεκινήσει μια επικοινωνία το πρώτο πράγμα που θα κάνει η master συσκευή είναι να εκπέμψει την ακολουθία έναρξης. Αυτό ειδοποιεί όλες τις slave συσκευές στο δίαυλο ότι πρόκειται να ξεκινήσει μια αποστολή. Μετά η master θα εκπέμψει την 7 bit διεύθυνση της συσκευής. Η slave συσκευή με την διεύθυνση που στάλθηκε θα συνεχίσει ενώ όλες οι άλλες θα περάσουν σε αναμονή. Μαζί με την διεύθυνση της συσκευής αποστέλλεται και ένα επιπλέον δυαδικό ψηφίο ( $R/\bar{W}$ ) που χρησιμεύει για να πληροφορήσει την slave αν η master πρόκειται να γράψει ή να διαβάσει από αυτήν, για το αν θα αποτελεί δηλαδή ο master πομπό ή δέκτη. Εάν το bit είναι μηδέν η master συσκευή πρόκειται να γράψει. Εάν είναι 1 πρόκειται να διαβάσει από την slave. Τα 7 bit της διεύθυνσης τοποθετούνται στις περισσότερες σημαντικές θέσεις του byte ενώ το bit ανάγνωσης/εγγραφής στο λιγότερο σημαντικό ψηφίο (LSB).

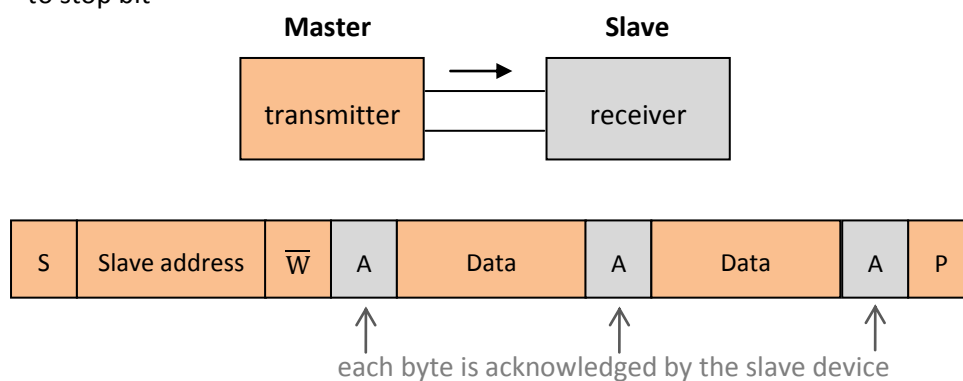
Όταν η master συσκευή διαβάζει από την slave, η slave τοποθετεί τα δεδομένα στον δίαυλο αλλά είναι η master που ελέγχει τους παλμούς του ρολογιού. Αν η slave δεν είναι

έτοιμη να στείλει δεδομένα το πρωτόκολλο επιτρέπει η slave να κρατάει το ρολόι σε κατάσταση λογικού 0 (επιμήκυνση ρολογιού).

### Παράδειγμα εγγραφής σε μια slave συσκευή

Η master συσκευή στέλνει:

1. το start bit
2. την διεύθυνση της slave συσκευής με το R/W' bit low δηλώνοντας έτσι ότι θέλει να στείλει δεδομένα
3. (προαιρετικά) την διεύθυνση του εσωτερικού καταχωρητή στον οποίο θέλει να γράψει
4. στέλνει το byte δεδομένων
5. (προαιρετικά) οποιονδήποτε επιπλέον αριθμό bytes
6. το stop bit

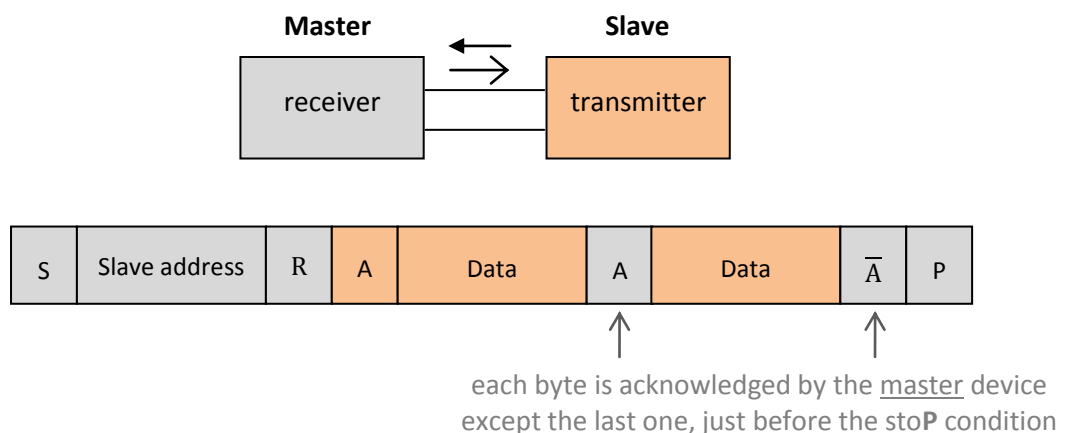


Εικόνα 5.40 Εγγραφή σε slave συσκευή

### Παράδειγμα ανάγνωσης από την slave συσκευή

Η master συσκευή:

1. στέλνει το start bit
2. στέλνει την διεύθυνση της slave συσκευής με το R/W' bit high
3. διαβάζει τα δεδομένα (ένα ή περισσότερα byte). Μετά από κάθε έγκυρη λήψη byte η master συσκευή στέλνει σήμα επιβεβαίωσης.
4. στέλνει το stop bit



Εικόνα 5.41 Ανάγνωση από slave συσκευή

### 5.9.1 Η βιβλιοθήκη Wire

Η βιβλιοθήκη Wire περιλαμβάνεται στην βασική έκδοση του Arduino IDE, υλοποιεί το πρωτόκολλο I<sup>2</sup>C και ενσωματώνει τις βασικές λειτουργίες επικοινωνίας με τις περιφερειακές συσκευές που το υποστηρίζουν. Διαθέτει το εξής ρεπερτόριο εντολών:

- `begin`
- `requestFrom`
- `beginTransmission`
- `endTransmission`
- `write`
- `available`
- `read`
- `onReceive`
- `onRequest`

Ακολουθεί μια σύντομη περιγραφή των συναρτήσεων αυτών.

#### `Wire.begin(address)`

Αρχικοποίηση πρωτοκόλλου και συμμετοχή της συσκευής στον διάυλο I<sup>2</sup>C ως master ή slave. Συνήθως καλείται μια φορά στο πρόγραμμα. Ως παράμετρος δίνεται η 7-bit διεύθυνση της slave συσκευής και αν δεν καθορίζεται τιμή η συσκευή συμμετέχει ως master.

#### `Wire.requestFrom(address, quantity, [stop])`

Χρησιμοποιείται όταν ο master αιτείται την αποστολή δεδομένων από μια slave συσκευή. Ως παράμετρος δίνεται η 7-bit διεύθυνση της slave συσκευής και το πλήθος των bytes που ζητούνται. Αν η παράμετρος stop είναι αληθής απελευθερώνεται ο διάυλος αποστέλλοντας ο master μια ακολουθία λήξης διαφορετικά στέλνεται ακολουθία επανέναρξης οπότε και δεν χάνεται ο έλεγχος του διαύλου αποτρέποντας έτσι και κάθε άλλη master συσκευή να τον ανακτήσει. Η συνάρτηση επιστρέφει τον αριθμό των bytes που στάλθηκαν από την slave συσκευή.

#### `Wire.beginTransmission(address)`

Έναρξη της μετάδοσης δεδομένων προς την slave συσκευή στην διεύθυνση που καθορίζεται.

#### `Wire.endTransmission([stop])`

Τερματίζεται η επικοινωνία με μια slave συσκευή. Ομοίως αν η παράμετρος stop είναι αληθής απελευθερώνεται ο διάυλος διαφορετικά η master συσκευή στέλνει ακολουθία επανέναρξης και διατηρεί τον έλεγχο του διαύλου. Αν δεν καθοριστεί η παράμετρος stop θεωρείται αληθής.

Επιστρέφεται ένα byte του οποίου η τιμή δηλώνει την επιτυχημένη ή όχι έκβαση της αποστολής των δεδομένων, συγκεκριμένα:

- 0: επιτυχημένη αποστολή
- 1: ο όγκος των δεδομένων είναι μεγάλος ώστε να χωρέσουν στον buffer μετάδοσης
- 2: λήψη σήματος μη επιβεβαίωσης (NACK) στην προσπάθεια επικοινωνίας με την slave συσκευή με την συγκεκριμένη διεύθυνση
- 3: λήψη σήματος μη επιβεβαίωσης (NACK) κατά την αποστολή των δεδομένων
- 4: άλλο σφάλμα

[Wire.write\(value\)](#)

[Wire.write\(string\)](#)

[Wire.write\(data, length\)](#)

Αποστολή των δεδομένων μιας slave συσκευής σε αίτηση της master ή εκπομπή δεδομένων από την master προς την slave. Ως παράμετρος δίνεται ένα byte ή μια συμβολοσειρά. Αν πρόκειται να σταλεί πίνακας δεδομένων πρέπει να καθορίζεται και το πλήθος των στοιχείων του. Επιστρέφεται το πλήθος των bytes που πραγματικά αποστάλθηκαν.

[Wire.available\(\)](#)

Επιστρέφει το πλήθος των διαθέσιμων bytes που είναι έτοιμα για ανάγνωση. Πρέπει να καλείται κατόπιν αίτησης της master για αποστολή δεδομένων από μια slave συσκευή ([requestFrom](#)) ή από μια slave με την κλήση της συνάρτησης [onReceive](#) κατά την λήψη δεδομένων.

[Wire.read\(\)](#)

Ανάγνωση ενός byte το οποίο αποστάλθηκε από μια slave συσκευή στην master μετά την κλήση της συνάρτησης [requestFrom](#) ή κατά την αποστολή από την master στην slave.

[Wire.onReceive\(handler\)](#)

Δήλωση του ονόματος της συνάρτησης η οποία θα χειρίζεται τα δεδομένα τα οποία λαμβάνει μια slave συσκευή από μια master. Η δηλωθείσα συνάρτηση δεν επιστρέφει κάποια τιμή και πρέπει να δέχεται ως παράμετρο το πλήθος των bytes που αναμένεται να ληφθούν από την slave.

[Wire.onRequest\(handler\)](#)

Δήλωση του ονόματος της συνάρτησης η οποία θα καλείται στην slave όταν η master αιτείται δεδομένα από αυτήν.

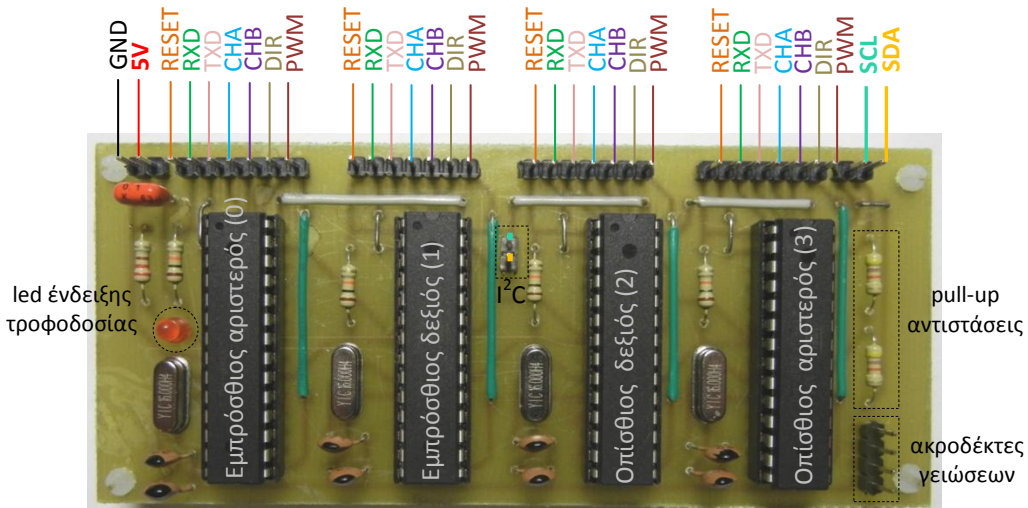
## 5.10 Τυπωμένο κύκλωμα ελέγχου της ταχύτητας περιστροφής των τροχών

Για τον έλεγχο της ταχύτητας περιστροφής κάθε τροχού κατασκευάστηκε το τυπωμένο κύκλωμα της εικόνας 5.42 οι διαστάσεις του οποίου είναι 110x50 mm περίπου. Σε κάθε κινητήρα αντιστοιχεί και ένας μικροελεγκτής ATmega328P-PU και το εγκατεστημένο σε αυτούς πρόγραμμα υλοποιεί τον έλεγχο κλειστού βρόγχου για την ρύθμιση της ταχύτητας περιστροφής του κάθε τροχού.

Ο μικροελεγκτής ATmega328P-PU επιλέχθηκε αφενός λόγω του ότι διαθέτει δυο ακροδέκτες εξωτερικών διακοπών (external interrupts) και δίνει την δυνατότητα να υλοποιηθεί η αποκωδικοποίηση x4 και αφετέρου διότι μπορεί να προγραμματιστεί εύκολα μέσω ενός board Arduino Uno. Κάθε μικροελεγκτής διαθέτει τον δικό του κρύσταλλο χρονισμού στα 16MHz.

Ο γειωμένος κεραμικός πυκνωτής 0.1μF στην είσοδο της τροφοδοσίας τοποθετήθηκε ώστε να εμποδίζεται η διέλευση ρευμάτων υψηλών συχνοτήτων αφού για αυτές αποτελεί βραχυκύκλωμα.



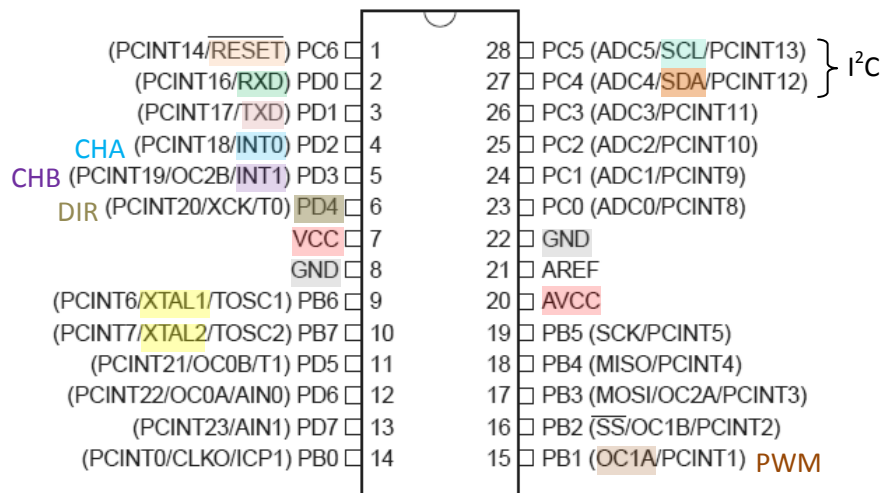


Εικόνα 5.42 Τυπωμένο κύκλωμα ελέγχου της ταχύτητας περιστροφής των τροχών

Οι μικροελεγκτές (slaves) επικοινωνούν με τον ATmega 2560 (σε λειτουργία master) μέσω του πρωτοκόλλου I<sup>2</sup>C. Για την ομαλή λειτουργία του διαύλου έχουν τοποθετηθεί στο board δυο αντιστάσεις 4.7KΩ ανύψωσης δυναμικού (pull-up) προς την τάση τροφοδοσίας (5V). Η συχνότητα λειτουργίας του διαύλου έχει καθοριστεί στα 100KHz.

Οι χρησιμοποιούμενοι ακροδέκτες σε κάθε ολοκληρωμένο κύκλωμα ATmega328P-PU καθώς και η λειτουργία τους εμφανίζονται στην εικόνα 5.43. Οι ακροδέκτες CHA και CHB για την ανάγνωση του κωδικοποιητή θέσης κάθε κινητήρα συνδέονται απευθείας στα ομώνυμα κανάλια και οι DIR και PWM στην αντίστοιχη Η-γέφυρα. Οι ακροδέκτες  $\overline{\text{RESET}}$  κάθε μικροελεγκτή συνδέονται με την ψηφιακή έξοδο 33 του board του ATmega2560 ώστε προγραμματιστικά να δίνεται η δυνατότητα επανεκκίνησης τους.

Κάθε μικροελεγκτής προγραμματίζεται ξεχωριστά με τη βοήθεια ενός board Arduino Uno όπως προαναφέρθηκε αφού εκ των προτέρων έχει αφαιρεθεί από την αντίστοιχη υποδοχή του το chip του μικροελεγκτή που αυτό φέρει. Συνδέονται οι ακροδέκτες 5V και GND με τους αντίστοιχους του Arduino Uno για την τροφοδοσία του και οι ακροδέκτες  $\overline{\text{RESET}}$ , RXD και TXD του ολοκληρωμένου που επιθυμούμε να προγραμματίσουμε με τους αντίστοιχους ακροδέκτες του Uno.



Εικόνα 5.43 Μικροελεγκτής ATmega328P-PU - χρησιμοποιούμενοι ακροδέκτες

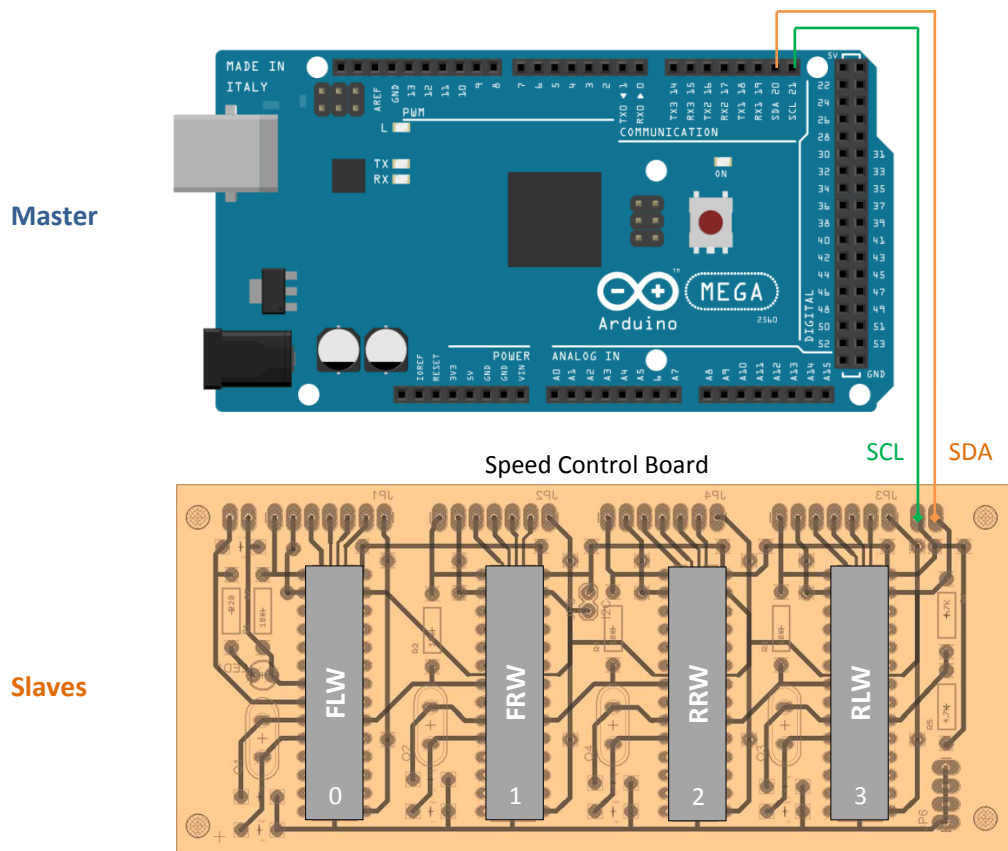
Ο master αποστέλλει κυκλικά σε κάθε slave το σήμα αναφοράς  $ref$  στο εύρος τιμών  $[-511,511]$  και την τρέχουσα μετρούμενη τάση τροφοδοσίας  $V_{cc}$  του συσσωρευτή. Οι διευθύνσεις που έχουν αποδοθεί στους slaves είναι 0,1,2 και 3 αντίστοιχα και εκείνοι με τις σειρά τους επιστρέφουν στον master όταν τους το ζητήσει την τρέχουσα εκτιμώμενη γωνιακή ταχύτητα περιστροφής των τροχών σε  $rad/s$  και την τιμή του μετρητή απαρίθμησης του κωδικοποιητή θέσης του κινητήρα. Το ληφθέν σήμα αναφοράς μετατρέπεται στην επιθυμητή τιμή της γωνιακής ταχύτητας περιστροφής  $\omega_d$  σύμφωνα με την σχέση

$$\omega_d = dcgain * V_{cc} * \frac{ref}{TOP}$$

με  $TOP = 511$  και  $dcgain = 0.6168$  το στατικό κέρδος του κινητήρα EMG49 που παρουσίασε κατά την διεξαγωγή των πειραματικών μετρήσεων στην μόνιμη κατάσταση τροφοδοτούμενος από 24V την χαμηλότερη γωνιακή ταχύτητα περιστροφής. Ο λόγος  $ref/TOP$  καθορίζει το duty cycle. Όταν  $ref = TOP$  τότε αυτός ισούται με την μονάδα και ο κινητήρας περιστρέφεται με την μέγιστη δυνατή γωνιακή ταχύτητα.

Η τιμή της εκτιμώμενης γωνιακής ταχύτητας (μεταβλητή τύπου float - 4 bytes) και η τιμή του μετρητή απαρίθμησης του κωδικοποιητή θέσης (ακέραια μεταβλητή 32 bit) μετατρέπονται σε bytes και αποστέλλονται μετέπειτα από τους slaves στον master. Για την μετατροπή τους δηλώνονται στην γλώσσα C++ ως ενώσεις (unions). Τα μέλη των ενώσεων αυτών μοιράζονται τον ίδιο χώρο μνήμης, τους εκχωρείται τιμή ανάλογη του τύπου της αλλά προσπελούνται ως bytes μέσω μονοδιάστατου πίνακα τεσσάρων στοιχείων.

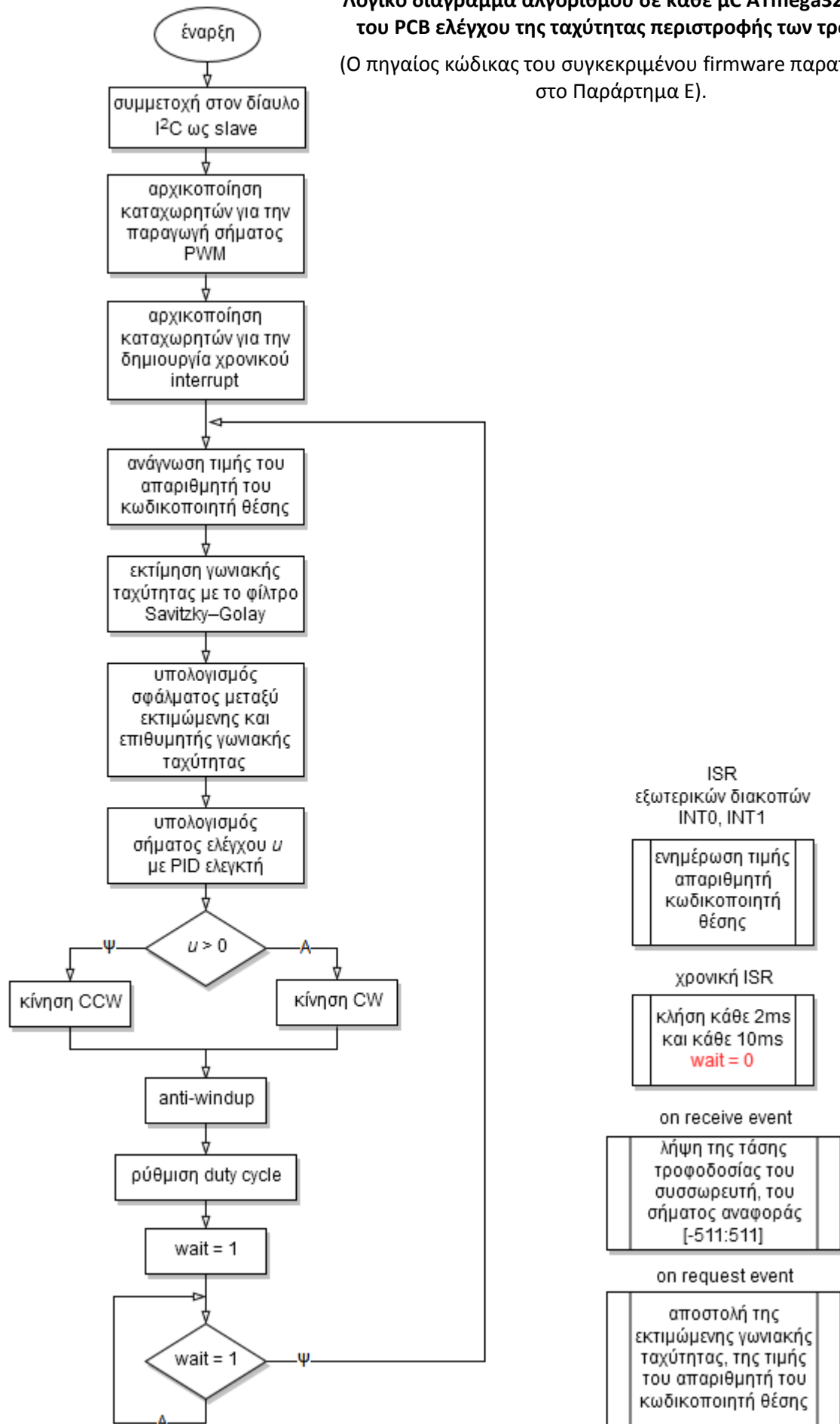
$V_{cc}, ref_{desired}: ref_l, ref_r \rightarrow$   
 $\leftarrow \omega_{actual}: \omega_0, \omega_1, \omega_2, \omega_3, enc\ cnts: c_0, c_1, c_2, c_3$



Εικόνα 5.44 Διασύνδεση του Arduino Mega 2560 με το PCB ελέγχου της ταχύτητας περιστροφής των τροχών

**Λογικό διάγραμμα αλγόριθμου σε κάθε  $\mu\text{C}$  ATmega328P-PU του PCB ελέγχου της ταχύτητας περιστροφής των τροχών**

(Ο πηγαίος κώδικας του συγκεκριμένου firmware παρατίθεται στο Παράρτημα Ε).



## 5.11 Η αναπτυξιακή πλακέτα Arduino Mega 2560

Η μητρική πλακέτα που προσφέρει την διασύνδεση όλων των επιμέρους διατάξεων και είναι επιφορτισμένη με τον συντονισμό και τον συγχρονισμό τους είναι η αναπτυξιακή πλατφόρμα Arduino Mega 2560 με ενσωματωμένο τον 8-bit μικροελεγκτή ATmega2560 αρχιτεκτονικής RISC της εταιρείας Atmel. Κριτήριο για την επιλογής της ήταν η δυνατότητα παραγωγής μεγάλου αριθμού σημάτων PWM (έως 15), η λήψη μέχρι 16 αναλογικών σημάτων, οι 4 μονάδες σειριακής διασύνδεσης USART και η υποστήριξη εξωτερικών διακοπών σε 6 ψηφιακούς ακροδέκτες. Τα τεχνικά χαρακτηριστικά του board συνοψίζονται στον κάτωθι πίνακα:

Τάση λειτουργίας	7-12V
Μικροελεγκτής	ATMega2560
Συχνότητα χρονισμού	16MHz
Flash memory	256KB (-8KB για τον bootloader)
SRAM	8KB
EEPROM	4KB
Ψηφιακοί ακροδέκτες I/O	54
Αναλογικές είσοδοι	16
Ακροδέκτες εξωτερικών διακοπών	6
Χρονιστές	4 των 16-bit και 2 των 8-bit
Υποστήριξη πρωτοκόλλων	SPI, I <sup>2</sup> C
Παροχή ρεύματος σε κάθε I/O ακροδέκτη	20mA
Παροχή ρεύματος για τον ακροδέκτη 3.3V	50mA
Μήκος	101.52mm
Πλάτος	53.3mm
Βάρος	37g

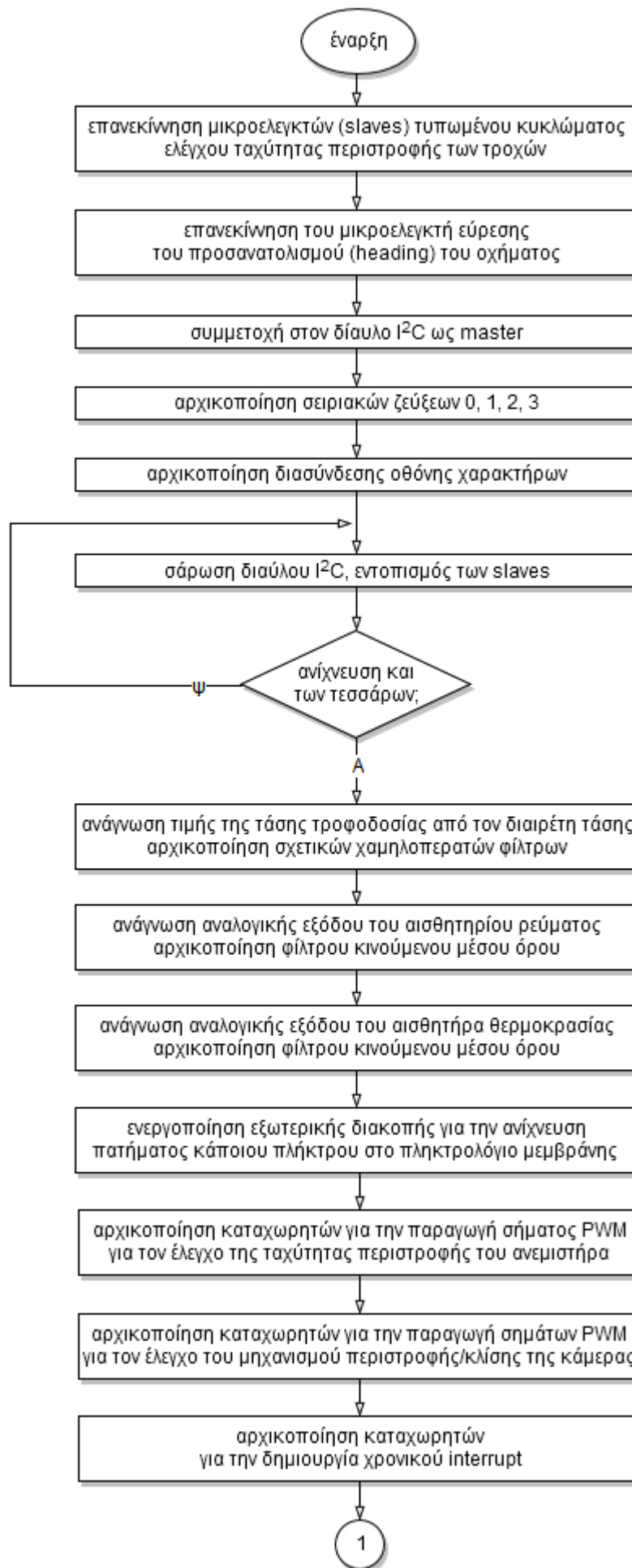
Προγραμματίζεται μέσω του περιβάλλοντος ανάπτυξης Arduino IDE με την γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο βιβλιοθηκών). Στην προκειμένη περίπτωση χρησιμοποιήθηκε ως επί το πλείστον το περιβάλλον ανάπτυξης Atmel Studio 6 με εγκατεστημένο το πρόσθετο (plugin) Visual Micro το οποίο προσφέρει περισσότερες δυνατότητες εκσφαλμάτωσης. Το πρόγραμμα σε περιβάλλον Arduino έχει την παρακάτω δομή:

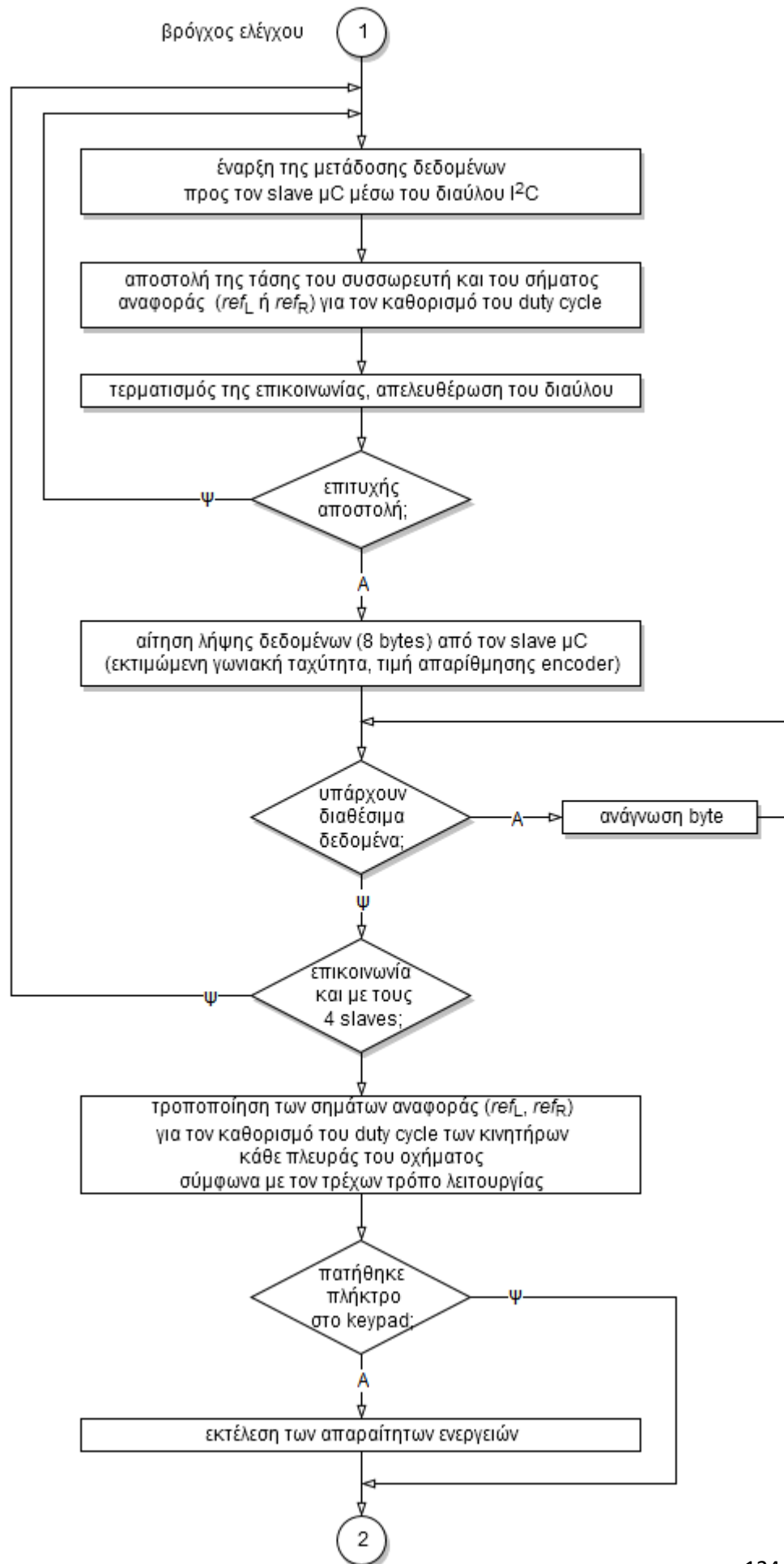
```
// δηλώσεις μεταβλητών
void setup() {
// αρχικοποίηση ρυθμίσεων
}

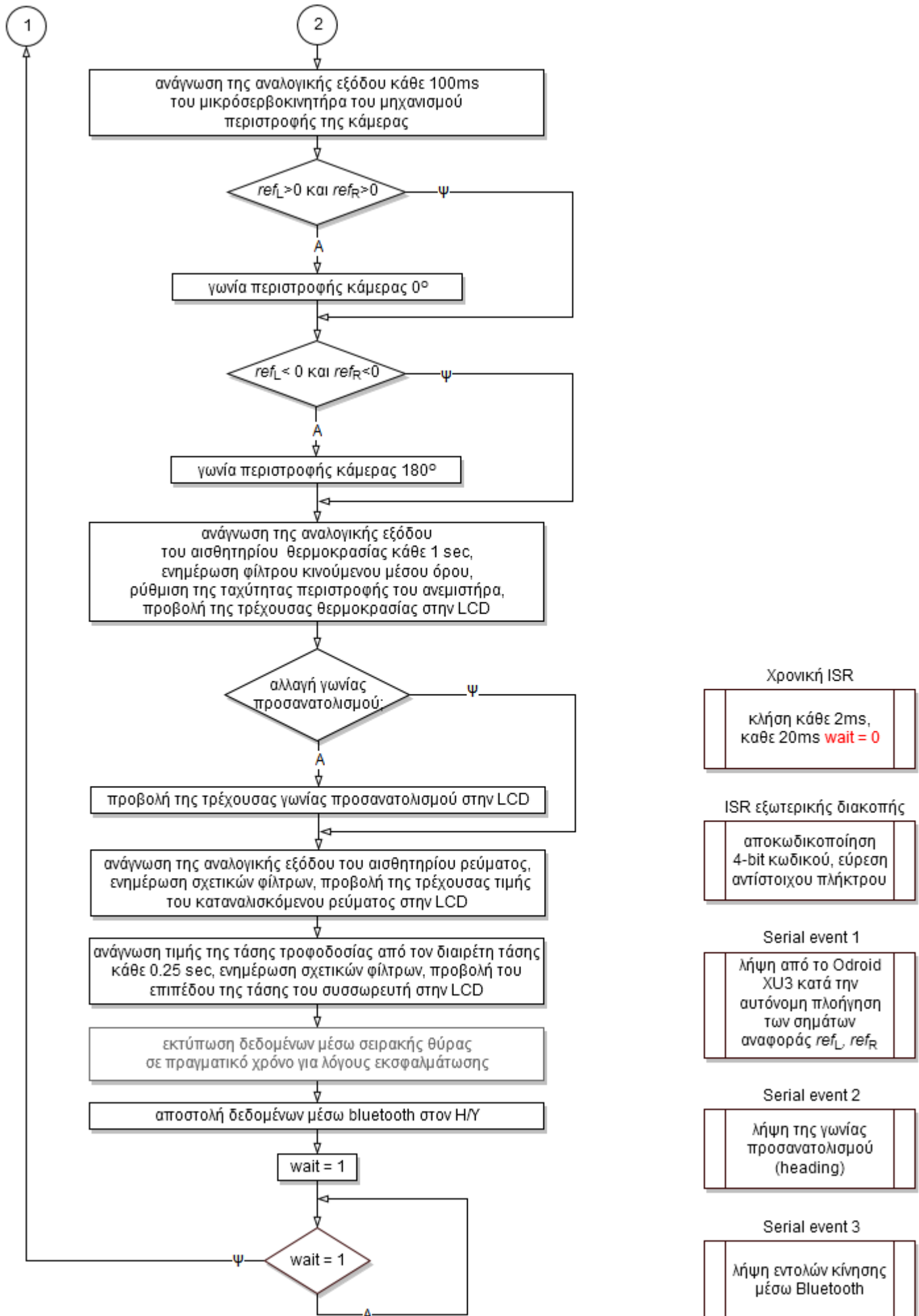
// εξωτερικός ατέρμων βρόγχος
void loop() {
...
while (wait==1); // εσωτερικός ατέρμων βρόγχος
}

Timer interrupt ISR: wait = 0;
```

Ειδικότερα για τον χρονισμό του βρόγχου ελέγχου χρησιμοποιείται ο 8-bit Timer 0 σε λειτουργία CTC (Clear Timer on Compare). Με την χρήση χρονικού interrupt ο εσωτερικός ατέρμων βρόγχος διακόπτεται κάθε 20ms οπότε εκτελείται η αντίστοιχη ρουτίνα εξυπηρέτησης της διακοπής (ISR). Ο πηγαίος κώδικας του εγκατεστημένου firmware στον μικροελεγκτή ATmega2560 παρατίθεται στο Παράρτημα Ε. Ακολουθεί το διάγραμμα ροής του συγκεκριμένου αλγορίθμου.







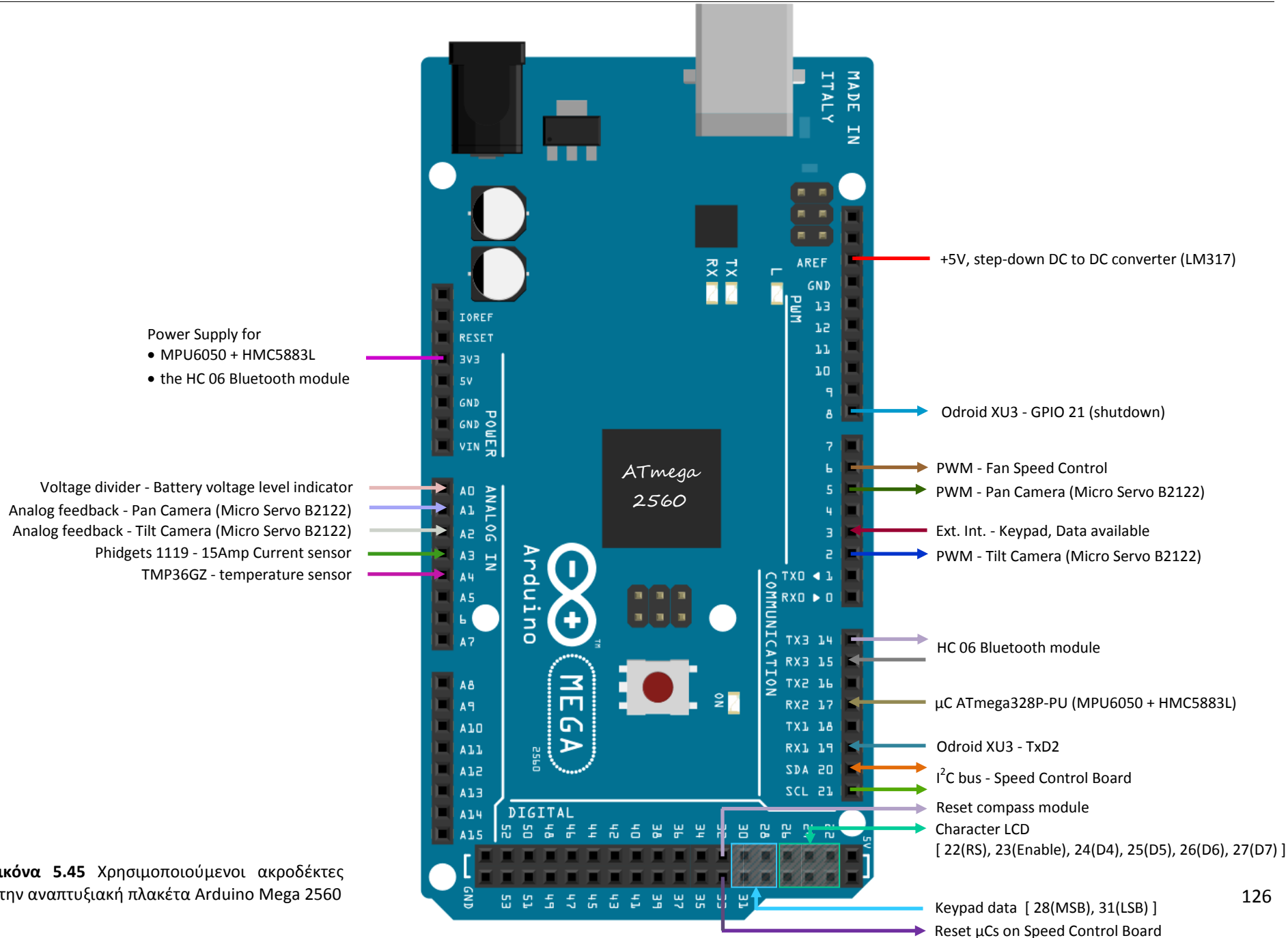
Χρονική ISR  
κλήση κάθε 2ms, κάθε 20ms **wait = 0**

ISR εξωτερικής διακοπής  
αποκωδικοποίηση 4-bit κωδικού, εύρεση αντίστοιχου πλήκτρου

Serial event 1  
λήψη από το Odroid XU3 κατά την αυτόνομη πλοήγηση των σημάτων αναφοράς  $ref_L$ ,  $ref_R$

Serial event 2  
λήψη της γωνίας προσανατολισμού (heading)

Serial event 3  
λήψη εντολών κίνησης μέσω Bluetooth

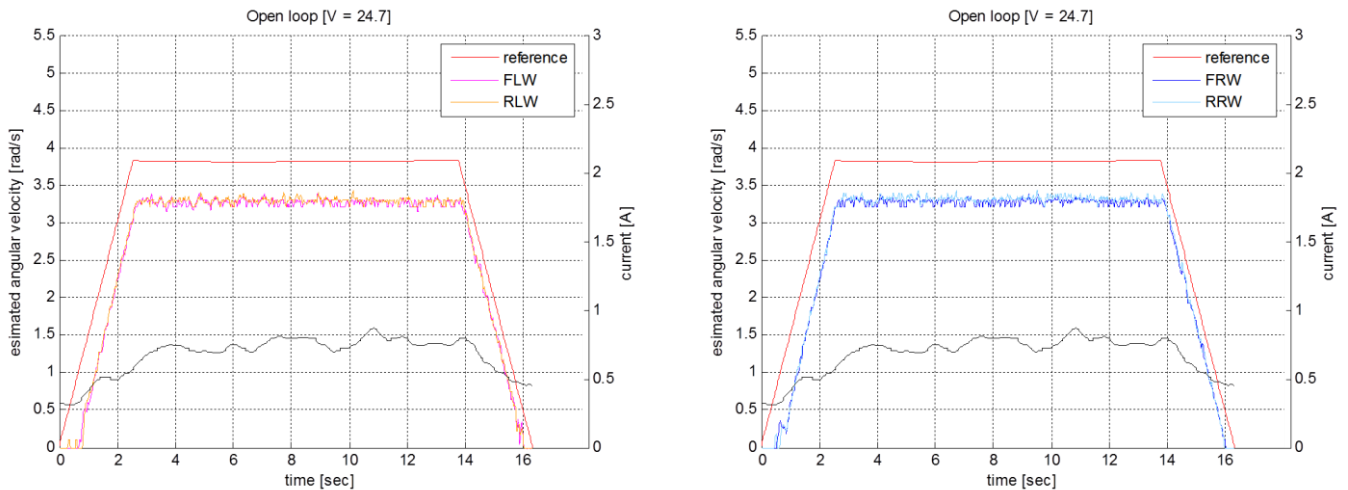


Εικόνα 5.45 Χρησιμοποιούμενοι ακροδέκτες στην αναπτυξιακή πλακέτα Arduino Mega 2560

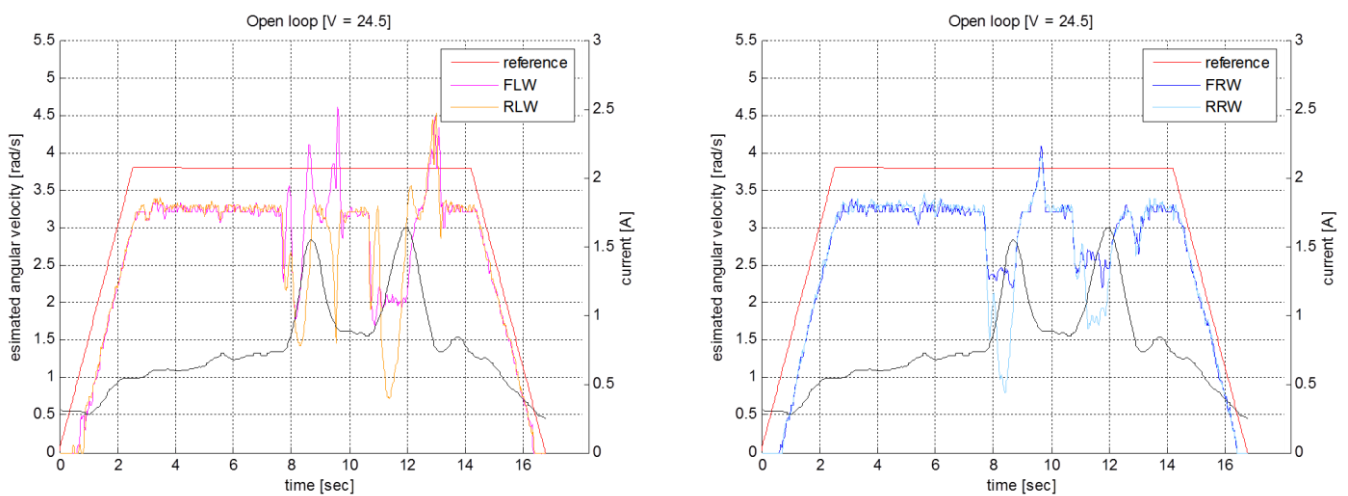


## 5.12 Πειραματικά αποτελέσματα

Παρατίθενται τα διαγράμματα από την διεξαγωγή πειραματικών μετρήσεων των ταχυτήτων περιστροφής των ηλεκτροκινητήρων. Το όχημα εκκινεί σε επίπεδη επιφάνεια σκυροδέματος τροφοδοτούμενο από τον συσσωρευτή. Σενάριο α) απλή διέλευση β) υπερπήδηση εμπόδιου 15 cm.



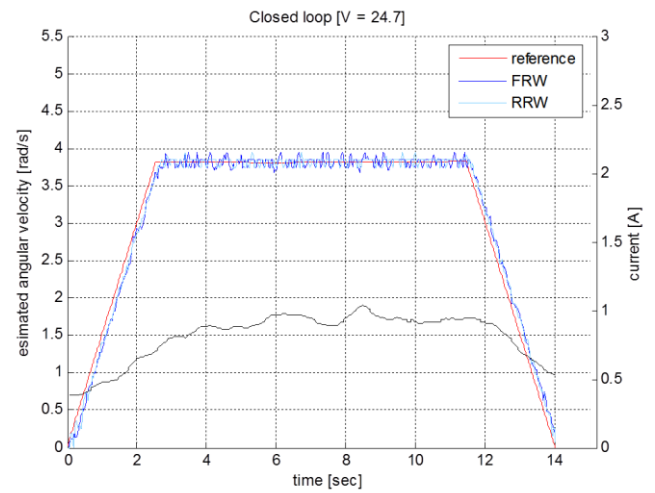
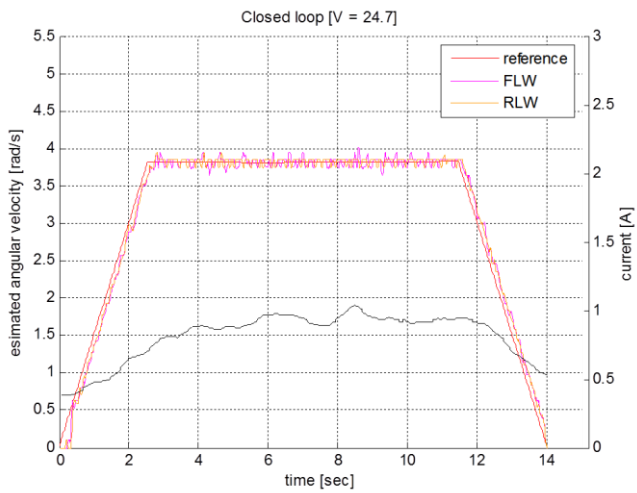
Εικόνα 5.46 Εκτιμώμενες ταχύτητες περιστροφής των κινητήρων σε ανοικτό βρόγχο



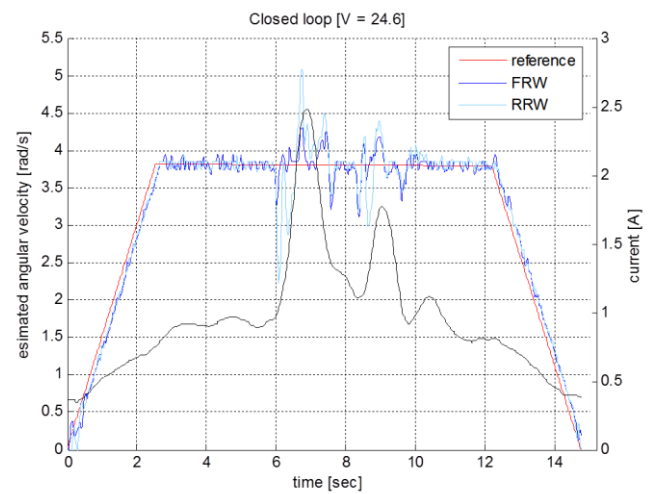
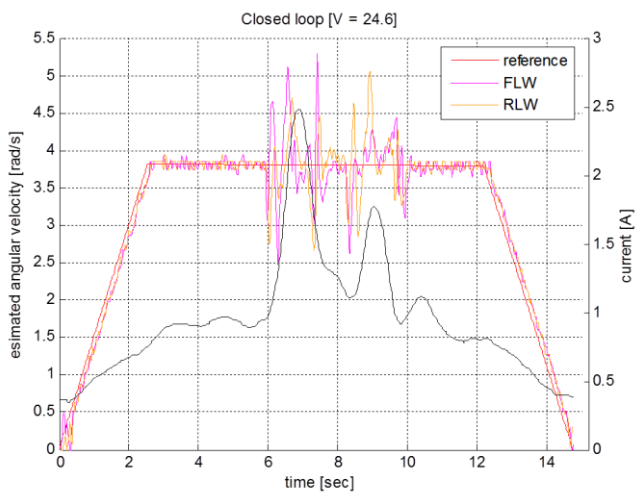
Εικόνα 5.47 Εκτιμώμενες ταχύτητες περιστροφής των κινητήρων σε ανοικτό βρόγχο. Το όχημα κατά την πορεία του και στην αριστερή του πλευρά συναντά και υπερπηδά εμπόδιο ύψους  $\approx 15$  cm

Τα κέρδη των ελεγκτών PID κάθε κινητήρα υπολογίστηκαν με τη μέθοδο της άμεσης σύνθεσης.

	FLW	FRW	RLW	RRW
$K_p$	0.70	0.67	0.68	0.68
$K_i$	18.71	18.93	19.07	18.87
$K_d$	0.0038	0.0041	0.0039	0.0039

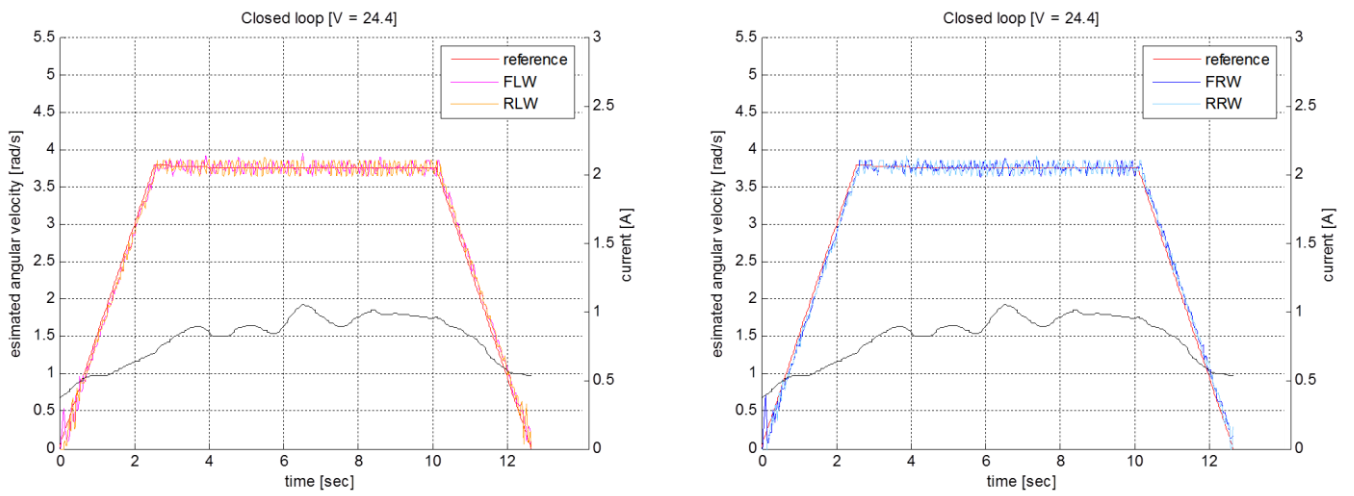


Εικόνα 5.48 Εκτιμώμενες ταχύτητες περιστροφής των κινητήρων σε κλειστό βρόγχο

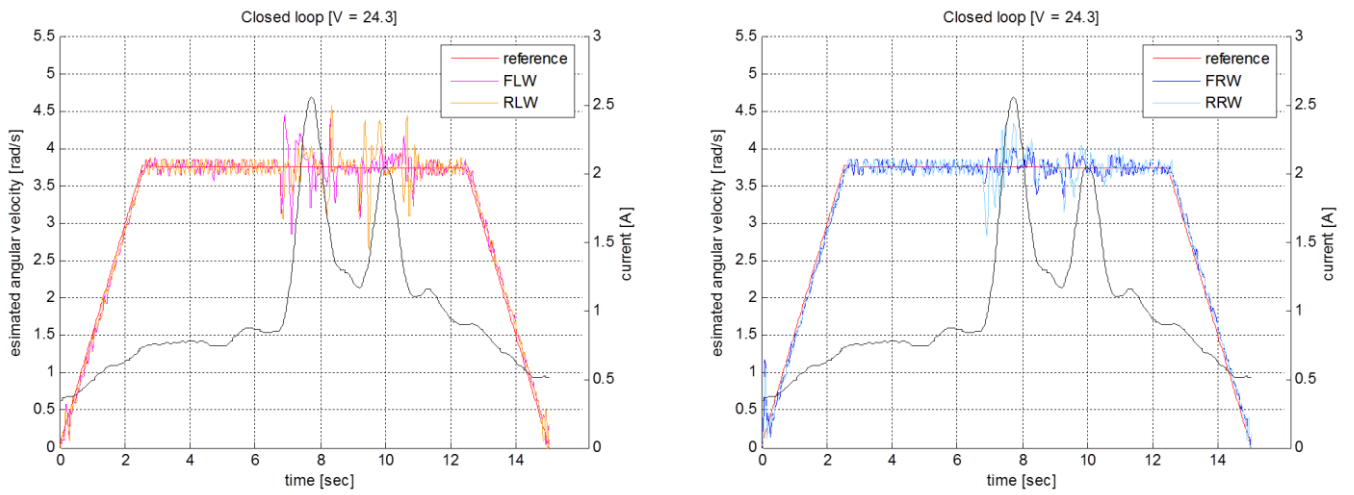


Εικόνα 5.49 Εκτιμώμενες ταχύτητες περιστροφής των κινητήρων σε κλειστό βρόγχο. Το όχημα κατά την πορεία του και στην αριστερή του πλευρά συναντά και υπερπηδά εμπόδιο ύψους  $\approx 15$  cm

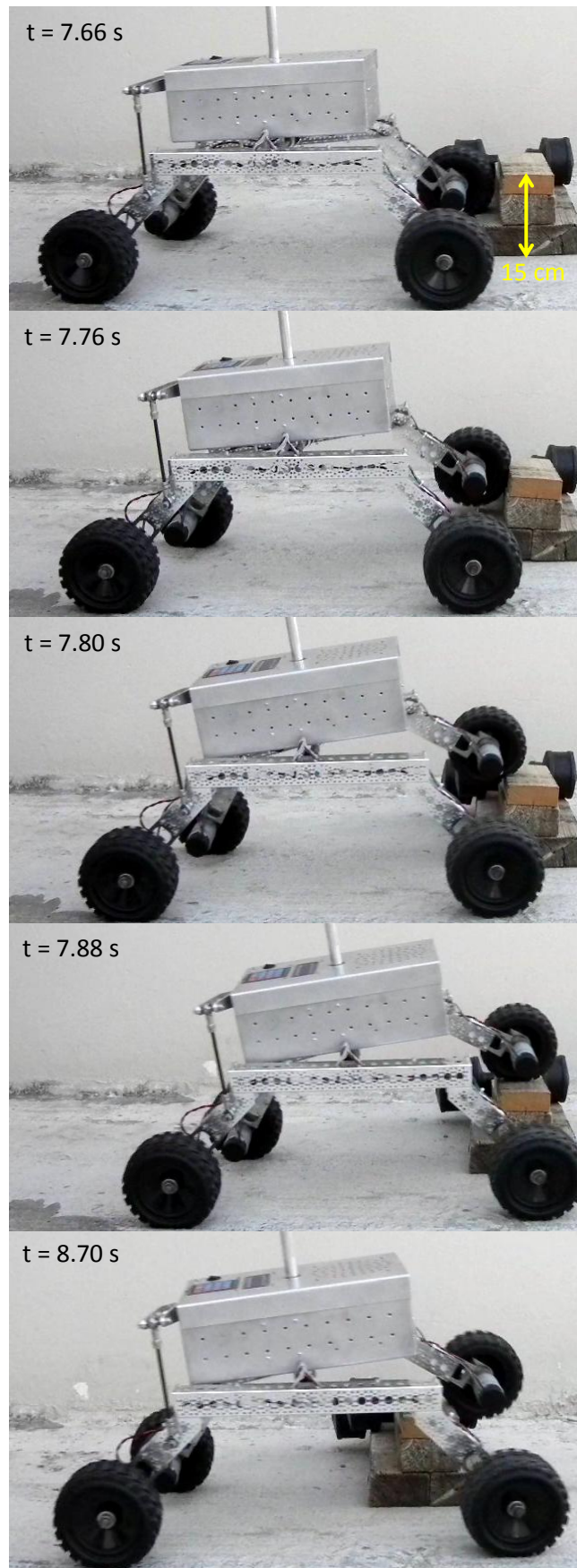
$$K_p = 2.4 \quad K_i = 43.64 \quad K_d = 0.033$$



Εικόνα 5.50 Εκτιμώμενες ταχύτητες περιστροφής των κινητήρων σε κλειστό βρόγχο



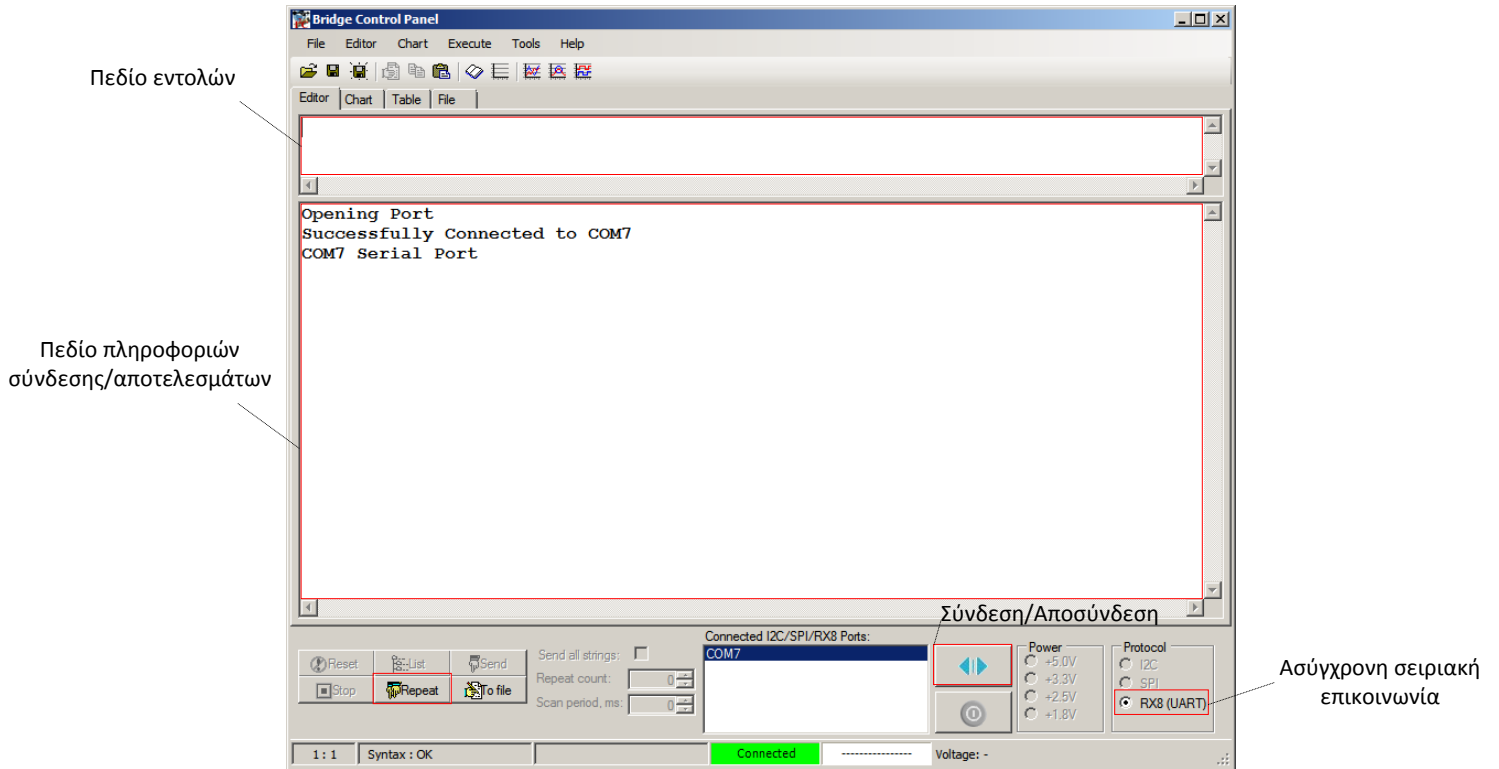
Εικόνα 5.51 Εκτιμώμενες ταχύτητες περιστροφής των κινητήρων σε κλειστό βρόγχο. Το όχημα κατά την πορεία του και στην αριστερή του πλευρά συναντά και υπερπηδά εμπόδιο ύψους  $\approx 15$  cm



Εικόνα 5.52 Στιγμιότυπα κίνησης του rover κατά την υπερπήδηση εμποδίου ύψους 15 cm

### 5.13 Εκτύπωση γραφημάτων σε πραγματικό χρόνο

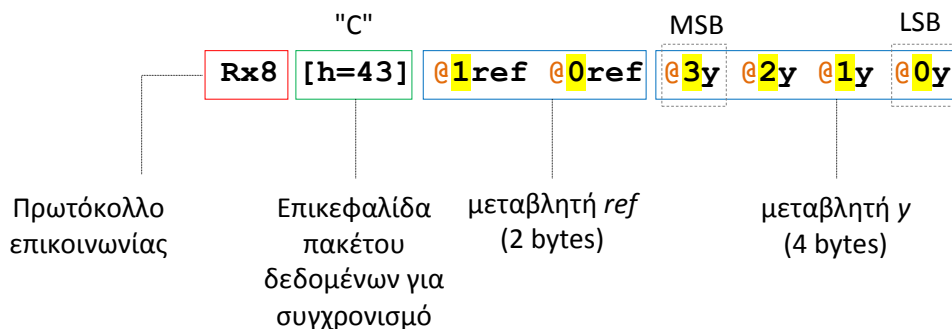
Για την αποτύπωση των δεδομένων που αποστέλλονται από την σειριακή θύρα ενός μικροελεγκτή σε γράφημα πραγματικού χρόνου (real time plotting) χρησιμοποιήθηκε η εφαρμογή Bridge Control Panel Software (Version 1.11.0.1926) της εταιρείας Cypress Semiconductor.

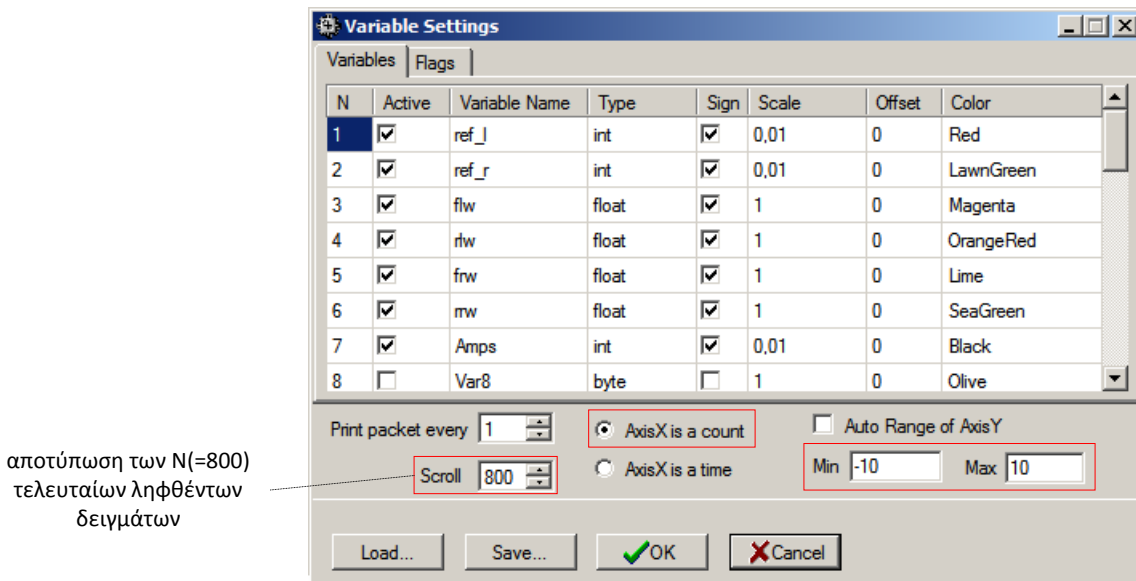


Εικόνα 5.53 Στιγμιότυπο οθόνης (screenshot) της εφαρμογής Bridge Control Panel

Μετά από μια επιτυχή σύνδεση με την επιθυμητή σειριακή θύρα δίνεται η δυνατότητα καθορισμού των ρυθμίσεων της επικοινωνίας (bit rate, data bits, parity type, stop bit) με την επιλογή Tools → Protocol Configuration → RX8(UART) μεταξύ του Η/Υ και του αναπτυξιακού board.

Η δήλωση των μεταβλητών που θα απεικονίζονται στο γράφημα πραγματοποιείται με την επιλογή Chart → Variable Settings του μενού της εφαρμογής. Στο αναδυόμενο παράθυρο διαλόγου καθορίζεται το όνομα της μεταβλητής, ο τύπος της (διαθέσιμες επιλογές: *byte*, *int*, *long int*, *float*), αν αυτή είναι προσημασμένη, ο συντελεστής κλίμακας κ.α. Στο πεδίο των εντολών δίνεται η κατάλληλη εντολή π.χ.





αποτύπωση των N(=800)  
τελευταίων ληφθέντων  
δειγμάτων

Εικόνα 5.54 Παράθυρο διαλόγου για τον καθορισμό των μεταβλητών

Μεταβλητές που λόγω του τύπου τους αποτελούνται από περισσότερα του ενός byte δηλώνεται κάθε ένα byte τους ως @Χόνομα\_μεταβλητής με X=0..3 καθορίζεται η θέση του στην δομή (πιο σημαντικό ή λιγότερο σημαντικό). Αν πρόκειται να αποσταλεί μεταβλητή του ενός byte τότε δηλώνεται ως @όνομα\_μεταβλητής. Σε γλώσσα προγραμματισμού Wiring σε περιβάλλον ανάπτυξης Arduino IDE ο μικροελεγκτής προγραμματίζεται για την αποστολή π.χ. ενός αριθμού 16 bit ως εξής:

```
Serial.print("C"); // χαρακτήρας συγχρονισμού
Serial.write(ref >> 8); // αποστολή σημαντικότερου byte
Serial.write(ref & 0xff); // αποστολή λιγότερου σημαντικού byte
```

Όταν πρόκειται π.χ. να αποσταλεί μεταβλητή τύπου float (4 bytes) δηλώνεται στην C/C++ ως ένωση (union). Τα στοιχεία της ένωσης μοιράζονται τον ίδιο χώρο μνήμης<sup>7</sup>, συνεπώς εκχωρείται η τιμή στην μεταβλητή y.fval ως float και προσπελαύνονται τα byte της ως πίνακας τεσσάρων στοιχείων με μια δομή επανάληψης.

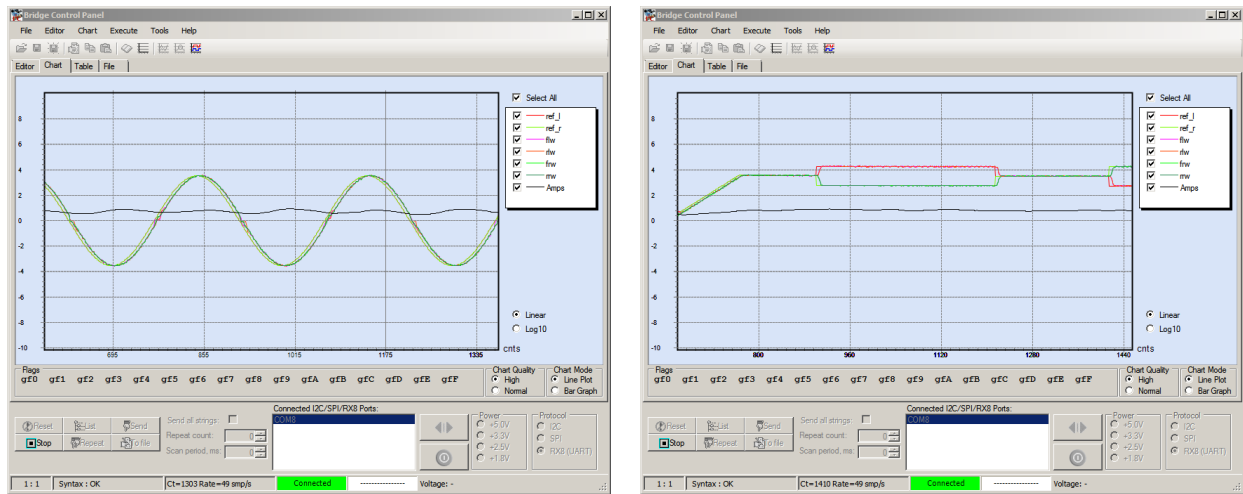
```
union f2b {
    byte b[4];
    float fval;
}y;
```

```
Serial.print("C");
for (int i=3;i>=0;i--)
    Serial.write(y.b[i]);
```

Μια διαφορετική προσέγγιση για την αποτύπωση float μεταβλητών σε γράφημα είναι η μετατροπή των τιμών τους, αν το επιτρέπει το εύρος τους και δεν μας ενδιαφέρει η καλύτερη δυνατή ακρίβεια, σε 16-bit ακέραιους αριθμούς πολλαπλασιάζοντας π.χ. με 100 και στρογγυλοποιώντας (round) τους στον πλησιέστερο ακέραιο. Κατά την λήψη τους οι τιμές διαιρούνται (scale) με τον ίδιο αριθμό (100) οπότε λαμβάνεται συγκεκριμένο μόνο πλήθος (2) από τα δεκαδικά τους ψηφία.

<sup>7</sup> η μνήμη είναι σχεδιασμένη να αποθηκεύει ομάδες 8 δυαδικών ψηφίων. Ο τρόπος με τον οποίο αποθηκεύεται μια σειρά bytes είναι little-endian: το λιγότερο σημαντικό byte αποθηκεύεται στην "μικρότερη" θέση μνήμης.

Η συνεχή συλλογή δεδομένων επιτυγχάνεται κάνοντας click στο πλήκτρο Repeat και η διακοπή της μεταφοράς τους με Stop. Στην καρτέλα Chart απεικονίζονται τα γραφήματα σε πραγματικό χρόνο.



- ref\_l
- ref\_r
- flw
- rlw
- frw
- mw
- Amps

- Γωνιακή ταχύτητα [rad/s]
- 1. σήματος αναφοράς αριστερών τροχών
- 2. σήματος αναφοράς δεξιών τροχών
- 3. εμπρόσθιου αριστερού τροχού
- 4. οπίσθιου αριστερού τροχού
- 5. εμπρόσθιου δεξιού τροχού
- 6. οπίσθιου δεξιού τροχού
- συνολικό καταναλισκόμενο ρεύμα [A]

Εικόνα 5.55 α) Στιγμιότυπα οθόνης κατά την διεξαγωγή πειραμάτων β) Υπόμνημα.

## Κεφάλαιο

## 6

## 6. Σύστημα μηχανικής όρασης - Υλικό & Λογισμικό

Στο πρώτο μέρος του κεφαλαίου αυτού παρουσιάζεται ο εγκατεστημένος εξοπλισμός για το σύστημα μηχανικής όρασης του οχήματος:

- ο μηχανισμός περιστροφής/κλίσης της κάμερας και οι ενσωματωμένοι σε αυτόν μικρό-σερβοκινητήρες
- ο μικροϋπολογιστής Odroid-XU3 στον οποίο εκτελείται το πρόγραμμα επεξεργασίας εικόνας κατά την αυτόνομη πλοήγηση της ρομποτικής πλατφόρμας σε καλλιέργεια σειρών

Ακολουθως δίνεται όλο το απαιτούμενο θεωρητικό υπόβαθρο και η επεξήγηση των εννοιών και των χρησιμοποιούμενων αλγορίθμων επεξεργασίας εικόνας που εφαρμόζονται στην προκειμένη περίπτωση. Παρουσιάζεται ο αλγόριθμος μηχανικής όρασης μέσω του οποίου εξάγονται τα απαραίτητα δεδομένα εισόδου για τον ασαφή ελεγκτή κίνησης. Κρίσιμο βήμα κατά την επεξεργασία αποτελεί η επιλογή του κατάλληλου χρωματικού δείκτη βλάστησης που χρησιμοποιείται κατά την τμηματοποίηση της ληφθείσας εικόνας.

### 6.1 Μηχανισμός περιστροφής/κλίσης της κάμερας



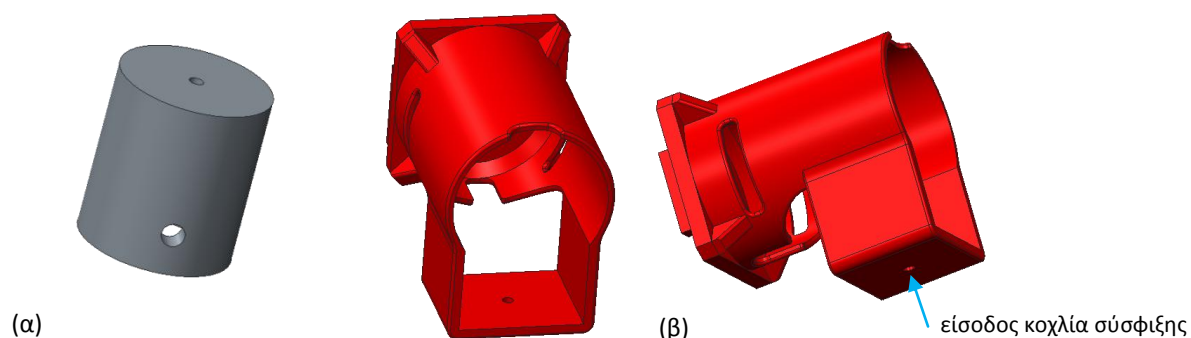
**Εικόνα 6.1** Τα πλαστικά μέρη της διάταξης και δυο διαφορετικές όψεις του συναρμολογημένου μηχανισμού

Στην συγκεκριμένη εφαρμογή λόγω του ότι το όχημα κατά την κίνηση μέσα στο «φυσικό διάδρομο» που σχηματίζουν οι συστοιχίες των φυτών της σειράς ενδέχεται να συναντήσει εμπόδιο το οποίο θα πρέπει να αποφύγει αλλάζοντας πορεία προς την αντίθετη κατεύθυνση σύμφωνα με του κανονισμούς του διαγωνισμού (ούτως ή άλλως δεν υπάρχει και διαθέσιμος χώρος για ελιγμό επιτόπιας περιστροφής) η κάμερα επιβάλλεται να περιστραφεί κατά 180° σύμφωνα με την κατεύθυνση του οχήματος. Επίσης σχεδιαστική απαίτηση αποτελεί η δυνατότητα ρύθμισης της κλίσης της κάμερας σε σχέση με το οριζόντιο επίπεδο για την εύρεση της κατάλληλης γωνίας. Για τους προαναφερθέντες λόγους και λόγω του χαμηλού του βάρους της (16g) επιλέχθηκε η διάταξη περιστροφής και



κλίσης (pan/tilt) της κάτωθι εικόνας. Οι διαστάσεις των υποδοχών της διάταξης στις οποίες σταθεροποιούνται οι δυο μικρό-σερβοκινητήρες (δεν εμπεριέχονται στην συσκευασία) είναι τυποποιημένες. Ο μηχανισμός τοποθετείται στο άκρο, κάθετου στο επίπεδο του οχήματος, αλουμινένιου σωλήνα εξωτερικής διαμέτρου 18 mm.

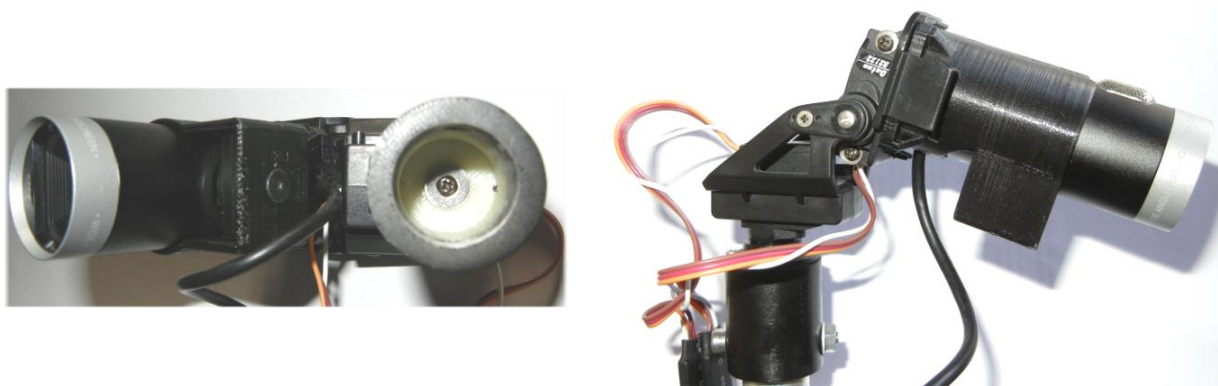
Για την προσαρμογή της web κάμερας στην διάταξη του μηχανισμού σχεδιάστηκε κατάλληλη βάση στήριξης με το λογισμικό PTC Creo Parametric 2.0 και στη συνέχεια κατασκευάστηκε από θερμοπλαστικό υλικό τύπου ABS με τη βοήθεια συστήματος τρισδιάστατης εκτύπωσης (3D printer). Η βάση της κάμερας κουμπώνει στις αντίστοιχες υποδοχές και μανταλώνει στα δυο πλαστικά κλείστρα που φέρει η πλατφόρμα κλίσης. Για την προσάρτηση της κάμερας στο κατασκευασμένο πλαίσιο στερέωσης της αφαιρέθηκε η εύκαμπτη βάση της.



**Εικόνα 6.2** α) 3D μοντέλο της βάσης στήριξης του μηχανισμού περιστροφής/κλίσης της κάμερας και β) της βάσης τοποθέτησης και στερέωσης της κάμερας

Η διατιθέμενη τετράγωνη βάση της διάταξης δεν χρησιμοποιήθηκε διότι δεν ταιρίαζε οπότε σχεδιάστηκε και κατασκευάστηκε μια εκ νέου στον 3D printer στην πάνω έδρα της οποίας βιδώθηκε το κατάλληλα ενισχυμένο με πλαστικό αποστάτη, hub του servo.

Ο μικρό-σερβοκινητήρας περιστροφής της κάμερας ενσωματώνεται (και κατ' επέκταση ολόκληρη η διάταξη) με την κοίλη κυλινδρική βάση μέσω κοχλία που διέρχεται από το κέντρο του πυθμένα της και τον άξονα του servo. Ο μηχανισμός συσφίγγεται με περικόχλιο M4 στο άκρο του σωλήνα μέσω κοχλία ο οποίος διέρχεται από ομόκεντρη διαμετρική οπή ίδιου διαμετρήματος της βάσης στήριξης και του σωλήνα τοποθέτησης.



**Εικόνα 6.3** Κάτω και πλάγια όψη του μηχανισμού περιστροφής/κλίσης της web κάμερας



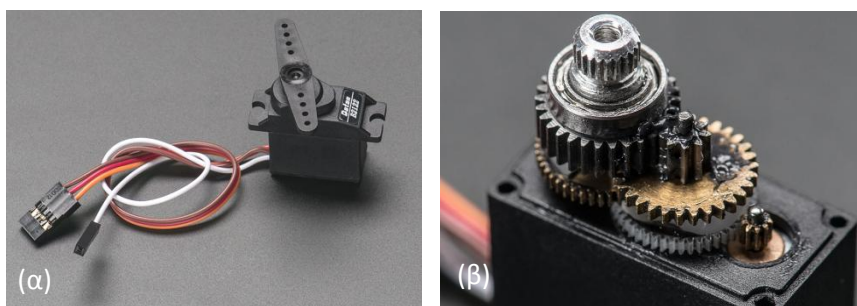
Εικόνα 6.4 Microsoft LifeCam Cinema 720p HD

Η web camera της εταιρείας Microsoft που χρησιμοποιείται έχει τα εξής τεχνικά χαρακτηριστικά:

Weight	98.3 g
Length	55.9 mm
Sensor	CMOS
Resolution	Motion Video: 1280x720 pixels
Imaging Rate	Up to 30 fps
Field of View	73° diagonal
Imaging Features	Auto focus, range from 6" to ∞

#### 6.1.1 Μικρό-σερβοκινητήρες Batan 2122 - Βαθμονόμηση

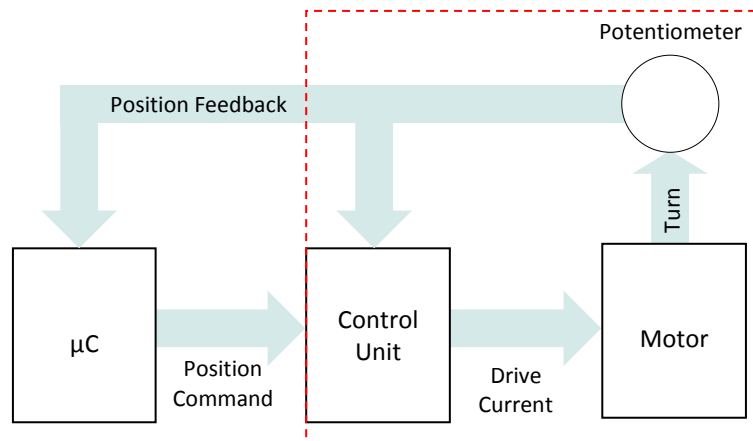
Για την περιστροφή του μηχανισμού και την ρύθμιση της γωνίας κλίσης της κάμερας χρησιμοποιούνται δυο μικρό-σερβοκινητήρες [Batan 2122](#). Διαθέτουν μεταλλικά γρανάζια στο μειωτήρα στροφών και εκτός των καλωδίων τροφοδοσίας (καφέ, κόκκινο) και του σήματος ελέγχου PWM (πορτοκαλί) προσφέρεται σύνδεση (άσπρο) με την μεσαία λήψη του ενσωματωμένου περιστροφικού ποτενσιόμετρου για έλεγχο της γωνιακής θέσης του άξονα μέσω αναλογικής εισόδου μικροελεγκτή. Η έξοδος αυτή χρησιμοποιείται για την περαιτέρω βελτίωση της ευστάθειας του συστήματος όταν ο άξονας δέχεται συνεχή υπέρμετρη ροπή ή ακόμη και για απλή καταγραφή της γωνιακής θέσης. Η λαμβανόμενη τάση από την μεσαία λήψη του ποτενσιόμετρου χρησιμοποιείται και εσωτερικά στο κύκλωμα ελέγχου της διάταξης για τον έλεγχο της θέσης του άξονα.



Εικόνα 6.5 α) Ο μικρό-σερβοκινητήρας Batan 2112 β) το σύστημα μετάδοσης κίνησης και ο μειωτήρας στροφών του

Τεχνικά χαρακτηριστικά

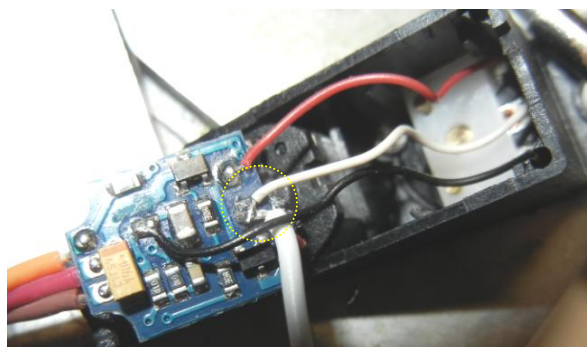
Διαστάσεις	22.6 x 11.4 x 22.2 mm	(4.8V)
Βάρος	15.8 g	
Ταχύτητα	0.12 sec/60°	
Ροπή	1.6Kg*cm = 0.157 Nm	



Εικόνα 6.6 Λήψη σήματος ανατροφοδότησης και από τον μικροελεγκτή για τον έλεγχο κλειστού βρόγχου μικρό-σερβοκινητήρα

Για τον ταυτόχρονο έλεγχο της γωνιακής θέσης των δυο μικρό-σερβοκινητήρων επιθυμείται η παραγωγή δυο PWM σημάτων με την ίδια συχνότητα αλλά διαφορετικό duty cycle. Χρησιμοποιείται η αυτοματοποιημένη παραγωγή σήματος PWM στον ακροδέκτη 5 (OC3A) του μικροελεγκτή ATmega2560 για τον σερβοκινητήρα περιστροφής (από 0-180°) και στον ακροδέκτη 2 (OC3B) για τον σερβοκινητήρα ο οποίος είναι επιφορτισμένος με την ρύθμιση της κλίσης από τον Timer3 μέσω της λειτουργίας Phase and Frequency Correct PWM mode 8. Λόγω του ότι η επιθυμητή τιμή της  $F_{pwm} = 50Hz$  δεν είναι άμεσα διαθέσιμη ο καθορισμός της μέγιστης τιμής μέτρησης TOP δίνεται μέσω του καταχωρητή ICR3(=20000). Η διακριτική ικανότητα του duty cycle είναι  $1/20000*100 = 0.005\%$  Η συχνότητα του παραγόμενου PWM σήματος καθορίζεται από την επιλεγμένη τιμή  $N(=8)$  του prescaler και την συχνότητα ρολογιού του μικροελεγκτή  $F_{ck} = 16MHz$

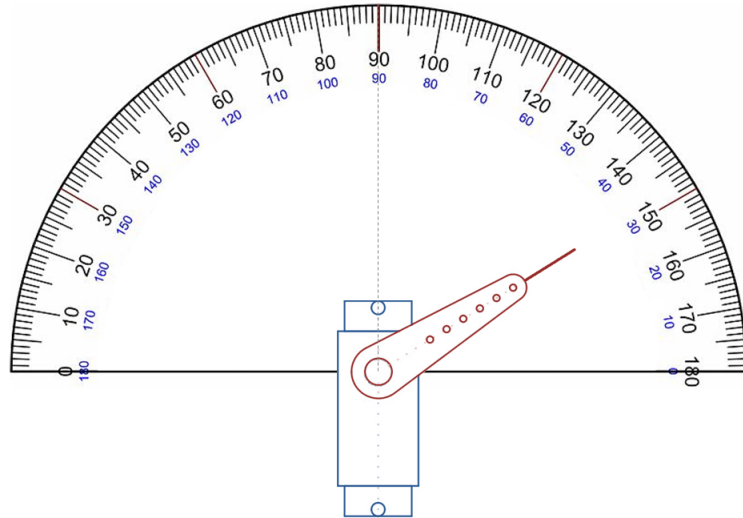
$$F_{pwm} = \frac{F_{ck}}{2 * N * TOP} = 50 Hz$$



Εικόνα 6.7 Το κύκλωμα ελέγχου στο εσωτερικό του micro servo Batan 2122.

Η διαδικασία βαθμονόμησης αποσκοπεί αφενός στον προσδιορισμό του απαιτούμενου σήματος ελέγχου προκειμένου ο άξονας του servo να οδηγείται με ακρίβεια σε συγκεκριμένη γωνία και αφετέρου στη συσχέτιση της εξόδου του ενσωματωμένου περιστροφικού ποτενσιόμετρου του servo με την γωνιακή θέση ώστε να είναι εφικτή η αποτίμηση της ακρίβειας με την οποία υλοποιείται η επιθυμητή θέση.

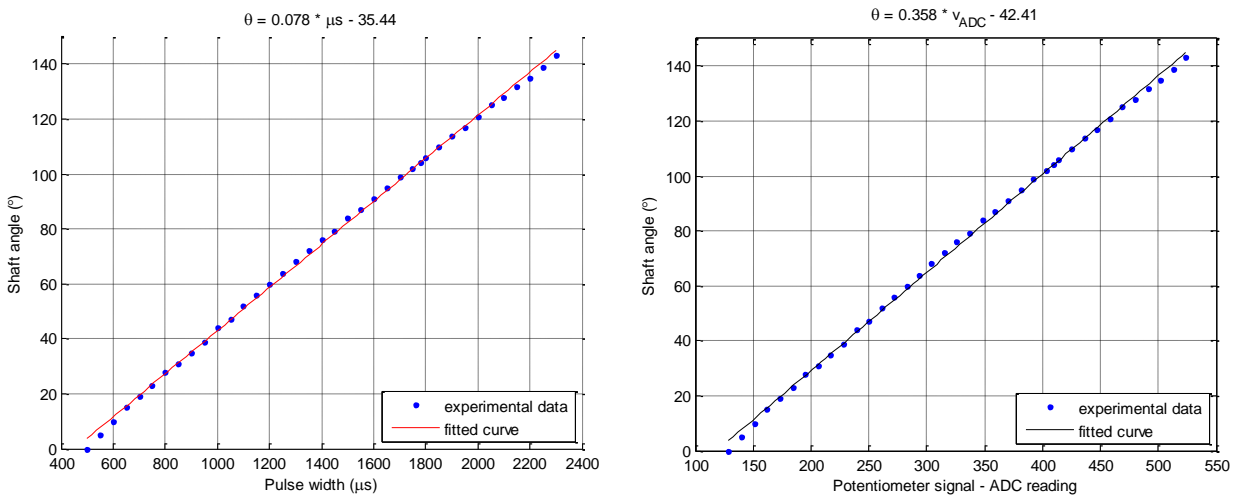
Για την διαδικασία κατασκευάστηκε η διάταξη η οποία απεικονίζεται στο παρακάτω σχήμα. Σε τμήμα χαρτονιού επικολλήθηκε βαθμονομημένο πολικό διάγραμμα και ο σερβοκινητήρας σταθεροποιήθηκε στο κέντρο του.



Εικόνα 6.8 Η διάταξη βαθμονόμησης των micro servo

Κατόπιν αναπτύχθηκε εφαρμογή στην αναπτυξιακή πλακέτα Arduino Mega2560 η οποία αποστέλλει μέσω της σειριακής θύρας την επιθυμητή τιμή του εύρους του PWM παλμού (ακέραιος αριθμός) και ο μικροελεγκτής επιστρέφει την τρέχουσα τιμή της τάσης της μεσαίας λήψης του περιστροφικού ποτενσιόμετρου θέσης σε μορφή αποτελέσματος μετατροπής 10bit-ADC (ακέραιος αριθμό με τιμές 0-1023). Η συγκεκριμένη ένδειξη της γωνιακής θέσης καταγράφηκε οπτικά με τη βοήθεια της διάταξης βαθμονόμησης.

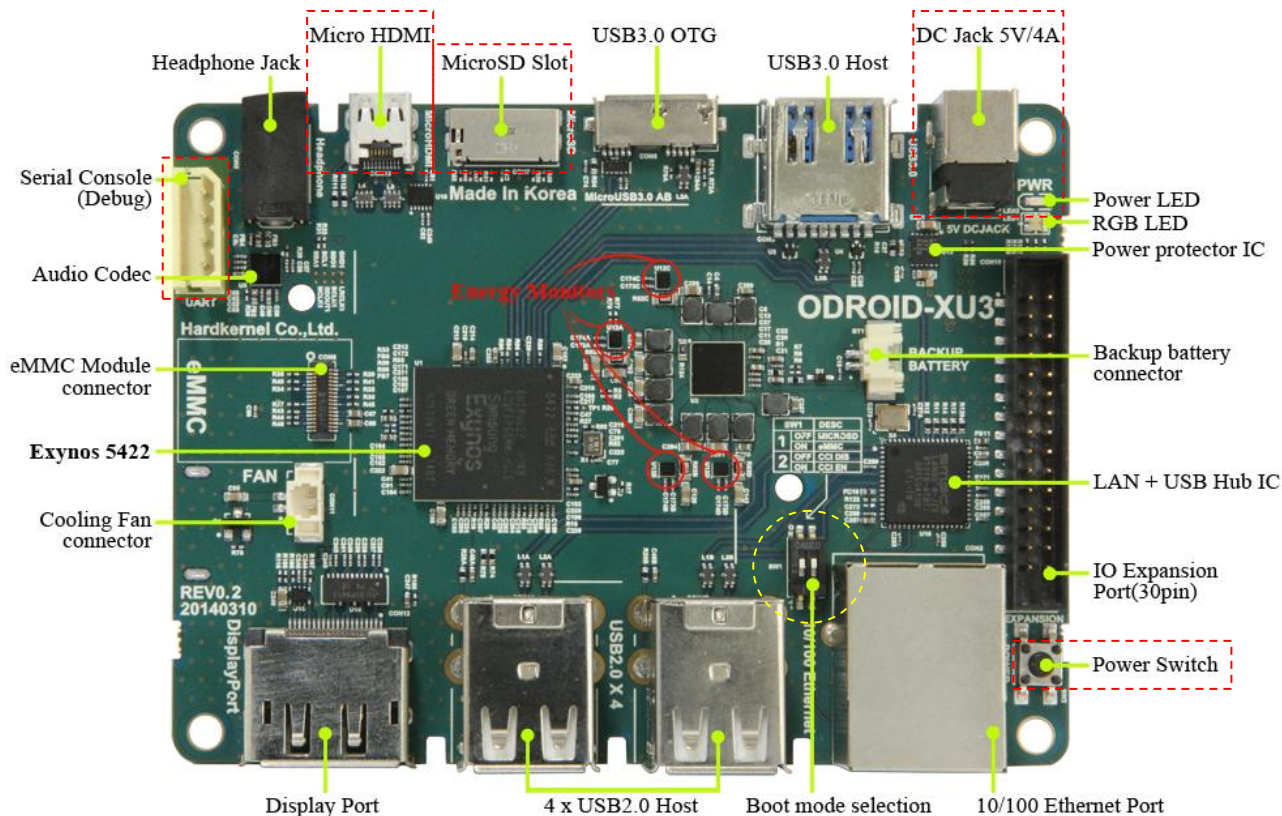
Τα δεδομένα που ελήφθησαν (εύρος παλμού PWM, μέτρηση από το ποτενσιόμετρο θέσης και η γωνιακή θέση) με την προηγούμενη διαδικασία επεξεργάστηκαν κατάλληλα στο Matlab μέσω της εντολής polyfit και προέκυψαν δυο πρωτοβάθμιες εξισώσεις. Η πρώτη περιγράφει τη θέση του άξονα συναρτήσει του εύρους παλμού του σήματος ελέγχου και η δεύτερη την θέση του άξονα συναρτήσει της μέτρησης του ποτενσιόμετρου.



Εικόνα 6.9 Πειραματικά δεδομένα βαθμονόμησης του micro servo Batan 2122 και γραμμική παρεμβολή

## 6.2 Ο μικροϋπολογιστής Odroid-XU3

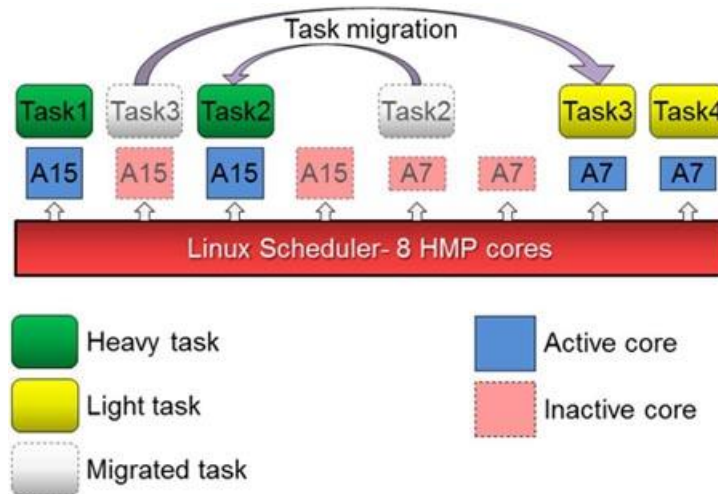
Το Odroid-XU3 της εταιρείας Hardkernel αποτελεί μικροϋπολογιστή (Single Board Computer) κατάλληλο για ενσωματωμένα συστήματα λόγω των μικρών διαστάσεων της πλακέτας, του ισχυρού επεξεργαστή που φέρει, της χαμηλής κατανάλωσης ενέργειας και του σχετικά χαμηλού κόστους αγοράς. Υποστηρίζει λειτουργικά συστήματα βασισμένα στο Linux και συγκεκριμένα εξειδικευμένες και τροποποιημένες εκδόσεις των γνωστότερων διανομών που προορίζονται για κοινούς υπολογιστές (π.χ. Ubuntu 14.04). Υποστηρίζεται από μια ιδιαίτερα ενεργή δικτυακή κοινότητα χρηστών.



Εικόνα 6.10 Odroid-XU3 Single Board Computer

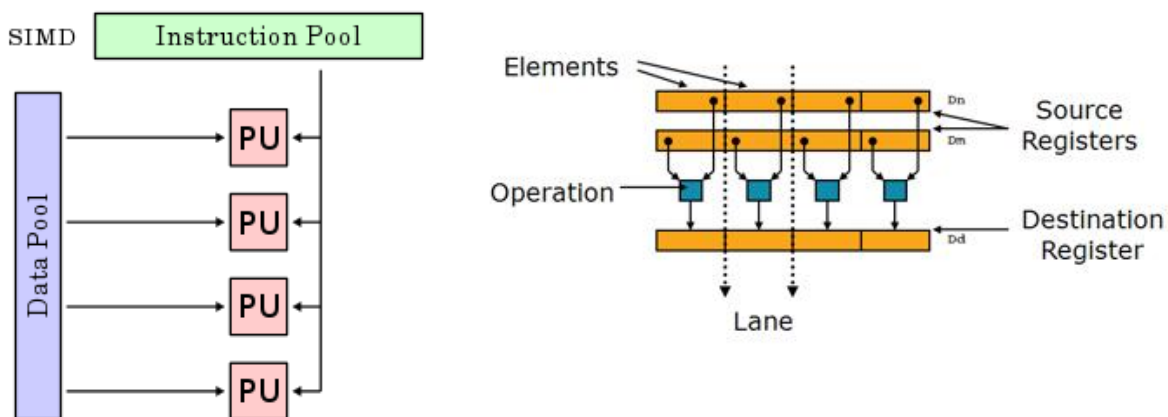
Ο octa-core επεξεργαστής *Exynos 5422* της Samsung του Odroid-XU3 υποστηρίζει την αρχιτεκτονική **big.LITTLE** η οποία επιτρέπει την ετερογενή πολυεπεξεργασία διεργασιών αξιοποιώντας βέλτιστα τους οκτώ πυρήνες. Οι 4 big πυρήνες (ARMv7<sup>8</sup> Cortex A15, 2GHz) υψηλών επιδόσεων αναλαμβάνουν την εκτέλεση των υπολογιστικά απαιτητικών διεργασιών και εκείνων με υψηλή προτεραιότητα με την ανάλογη αύξηση στην κατανάλωση ενέργειας ενώ οι 4 LITTLE πυρήνες (ARMv7 Cortex A7, 1.4GHz) εκτελούν τις διεργασίες με χαμηλό υπολογιστικό φόρτο αλλά με αντίστοιχα μειωμένη κατανάλωση. Με βάση τις ανάγκες και με στόχο την εξοικονόμηση ενέργειας αυτοματοποιημένα μεταφέρεται ο φόρτος στους κατάλληλους πυρήνες. Το Odroid-XU3 διαθέτει κυκλωματική διάταξη απαρτιζόμενη από τέσσερις αισθητήρες (ρεύματος/τάσης) για την μέτρηση της κατανάλωσης ενέργειας των 4+4 πυρήνων, της μονάδας επεξεργασίας γραφικών (GPU) και της δυναμικής μνήμης (DRAM) σε πραγματικό χρόνο.

<sup>8</sup> 32-bit RISC επεξεργαστές. Η συγκεκριμένη αρχιτεκτονική βασίζεται στο μειωμένο ρεπερτόριο εντολών (Reduced Instruction Set Computers) με συνέπεια τη χρήση σημαντικά μικρότερου αριθμού transistor σε σχέση με τους επεξεργαστές αρχιτεκτονικής CISC οι οποίοι χρησιμοποιούν ισχυρό σύνολο εντολών με αυξημένη πολυπλοκότητα. Στους RISC επεξεργαστές κάθε εντολή εκτελείται συνήθως σε έναν κύκλο μηχανής.



Εικόνα 6.11 Ετερογενής πολυεπεξεργασία διεργασιών στον επεξεργαστή Exynos 5422

Οι επεξεργαστές ARM της σειράς Cortex-A είναι εξοπλισμένοι με συνεπεξεργαστή με την επωνυμία NEON αρχιτεκτονικής SIMD<sup>9</sup> (Single Instruction, Multiple Data) κατά την οποία όλες οι μονάδες επεξεργασίας (PU) εκτελούν ταυτόχρονα την ίδια εντολή αλλά σε διαφορετικά ομαδοποιημένα σύνολα δεδομένων με αποτέλεσμα να παράγονται περισσότερα αποτελέσματα ανά κύκλο ρολογιού. Η συγκεκριμένη τεχνολογία ενδείκνυται για την αποδοτικότερη και γρηγορότερη εκτέλεση πολυμεσικών εφαρμογών. Ο συνεπεξεργαστής διαθέτει 32 καταχωρητές εύρους 64 δυαδικών ψηφίων (ή 16 εύρους 128 bits).



Εικόνα 6.12 Αρχιτεκτονική SIMD σε πολυπύρνο επεξεργαστή: μια μοναδική ροή εντολών, πολλαπλή ροή δεδομένων

Οι σύγχρονες εφαρμογές ιδιαίτερα οι πολυμεσικές χειρίζονται τεράστιο όγκο δεδομένων τα οποία είναι οργανωμένα σε ομάδες με μέγεθος (πλήθος δυαδικών ψηφίων) μικρότερο του μήκους της λέξης (4 bytes = 32 bit) της μονάδας επεξεργασίας. Για παράδειγμα στις εφαρμογές ήχου είναι σύνηθες το πλήθος κάθε πακέτου δεδομένων να είναι 16 bit και στις εφαρμογές γραφικών και video 8 bit. Όταν εκτελούνται λειτουργίες σε ένα 32 bit μικροεπεξεργαστή τμήματα των μονάδων υπολογισμού μένουν αχρησιμοποίητα αλλά συνεχίζουν να τροφοδοτούνται άρα και να καταναλώνουν ενέργεια. Για την καλύτερη αξιοποίηση των διαθέσιμων πόρων η τεχνολογία SIMD χρησιμοποιεί μια εντολή για την παράλληλη εκτέλεση ίδιων λειτουργιών σε πολλαπλές ομάδες δεδομένων του ίδιου τύπου και μεγέθους. Συνεπώς λόγω του υλικού δύναται π.χ. να προστεθούν δυο 32-bit τιμές από

<sup>9</sup> κατηγοριοποίηση κατά Flynn

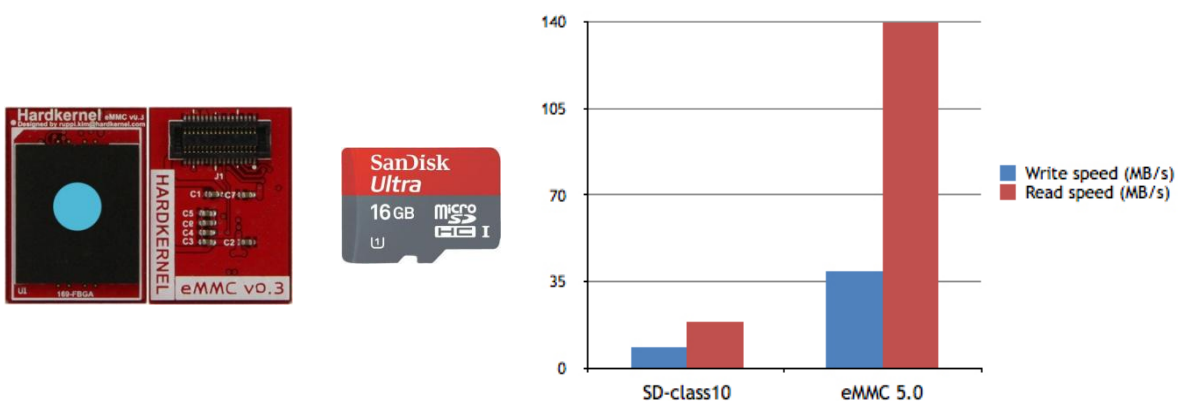
το να εκτελεστούν τέσσερις προσθήσεις των 8 bit. Η υλοποίηση της παράλληλης επεξεργασίας πραγματοποιείται στη μονάδα NEON από εξειδικευμένο ρεπερτόριο εντολών.

Για την μόνιμη αποθήκευση των δεδομένων και την εγκατάσταση του λειτουργικού συστήματος δίνονται δυο επιλογές: είτε με κάρτα μνήμης MicroSD είτε με άρθρωμα μνήμης eMMC<sup>10</sup> έκδοση 5.0. Η κάρτα μνήμης eMMC 5.0 είναι περίπου 7 φορές ταχύτερη από μια MicroSD - Class 10 κατά την ανάγνωση δεδομένων και 4 φορές περίπου κατά την εγγραφή. Η ενεργοποίηση του επιθυμητού μέσου αποθήκευσης επιτυγχάνεται μέσω της κατάλληλης ρύθμισης των δυο θέσεων του μικροδιακόπτη SW1.

SW1		1st Boot media
1	2	
ON	ON	eMMC
OFF	ON	MicroSD



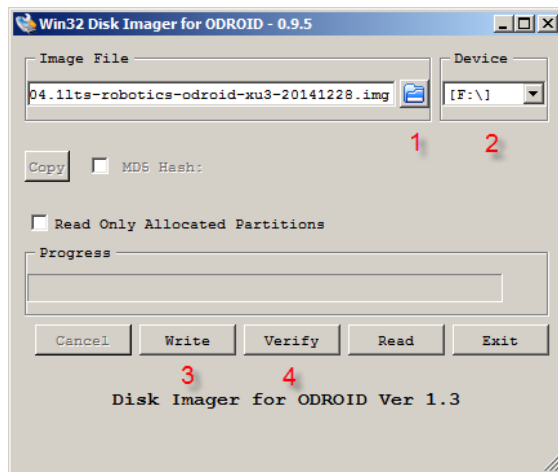
Εικόνα 6.13 Ρύθμιση των θέσεων του dip switch για την επιλογή αποθηκευτικού μέσου εκκίνησης



Εικόνα 6.14 Κάρτες μνήμης eMMC, MicroSD-Class 10 και συγκριτική αξιολόγηση

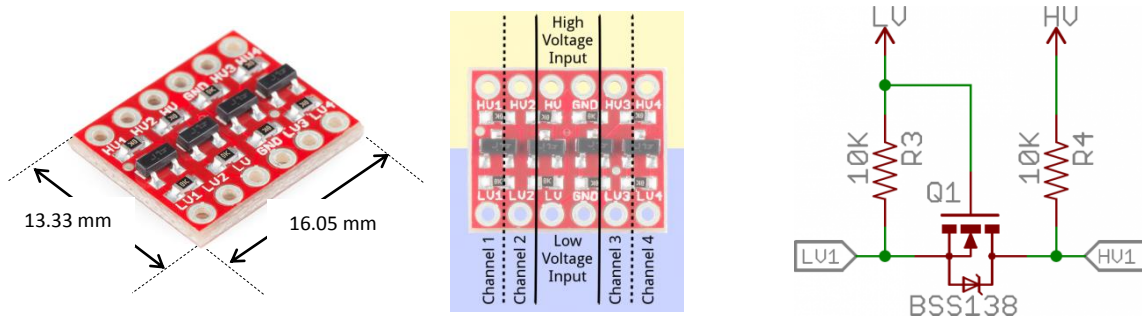
Για την εγκατάσταση του λειτουργικού συστήματος Ubuntu 14.04.01 χρησιμοποιήθηκε προρυθμισμένη εικόνα δίσκου (image disk) με προεγκατεστημένες τις βιβλιοθήκες OpenCV. Για αποθηκευτικό μέσο χρησιμοποιήθηκε κάρτα μνήμης MicroSD - Class 10 της SanDisk με χωρητικότητα 16GB. Η εικόνα δίσκου εγγράφηκε στην κάρτα μνήμης με τη βοήθεια USB card reader και της εφαρμογής Win32 Disk Imager for ODROID - 0.9.5. Μετά το πέρας της διαδικασίας η επικαιροποίηση των γραφθέντων αρχείων στην MicroSD πραγματοποιείται κάνοντας απλά κλικ στο κουμπί *Verify*. Η διαμέριση δίσκου (partition) που δημιουργήθηκε επεκτάθηκε ούτως ώστε να δεσμεύεται ολόκληρη η διαθέσιμη χωρητικότητα της κάρτας MicroSD με την εφαρμογή GParted.

<sup>10</sup> embedded Multi-Media Controller: κάρτα με μνήμη flash (μη πτητική) με ενσωματωμένο ελεγκτή για την ανάγνωση και εγγραφή των δεδομένων στο ίδιο ολοκληρωμένο κύκλωμα. Στο παρελθόν η διαχείριση της μνήμης γίνονταν μέσω αποκλειστικού ελεγκτή εκτός μνήμης ωστόσο η εξέλιξη της τεχνολογίας των ημιαγωγών οδήγησε σε μεγέθη μνημών με τεράστιες χωρητικότητες μη διαχειρίσιμες επαρκώς από εξωτερικό ελεγκτή. Συνεπώς το πρότυπο eMMC δημιουργήθηκε για τη βελτίωση των ποσοστών του ρυθμού μεταφοράς δεδομένων.



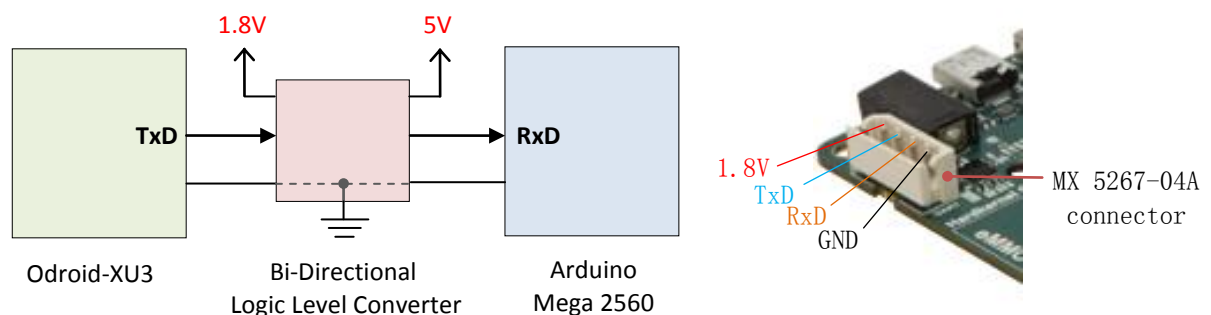
Εικόνα 6.15 Το GUI της εφαρμογής Win32 Disk Imager και ο USB card reader που χρησιμοποιήθηκε

Η διασύνδεση μεταξύ του Odroid-XU3 και του Arduino Mega 2560 επιτυγχάνεται ενσύρματα με ασύγχρονη σειριακή ζεύξη μέσω της υποδοχής MX 5267-04A του UART2 (/dev/ttySAC2) του μικροϋπολογιστή Odroid-XU3 και του USART1 του μικροελεγκτή ATmega2560. Τα σήματα που παράγει/λαμβάνει το ενσωματωμένο UART στους ακροδέκτες του Odroid-XU3 είναι επιπέδου λειτουργίας 1.8V και επομένως δεν μπορεί να επικοινωνήσει άμεσα με τα επιπέδου TTL +5V του ATmega2560. Η αμφίδρομη προσαρμογή των σημάτων επικοινωνίας επιτυγχάνεται από εξωτερικό κύκλωμα μετατροπής των λογικών επιπέδων βασικό δομικό στοιχείο του οποίου αποτελεί το N-channel MOS-FET BSS138.



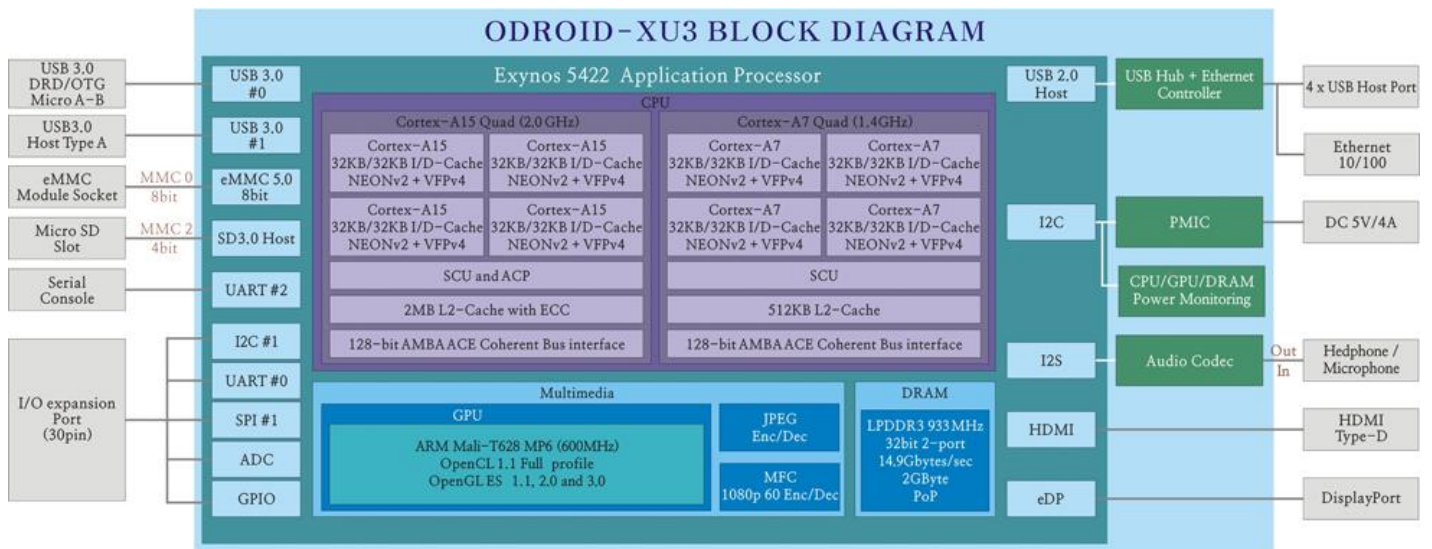
Εικόνα 6.16 Αμφίδρομος μετατροπέας λογικών επιπέδων 4 σημάτων με βάση το N-channel MOS-FET BSS138

Οι εξής επιλογές επικοινωνίας έχουν ρυθμιστεί για την μετάδοση: 8 data bits, 1 start bit, 1 stop bit, no parity bit. Η ταχύτητα αποστολής και λήψης των σειριακών δεδομένων καθορίστηκε στα 9600bps.



Εικόνα 6.17 Διασύνδεση του Odroid XU3 με τον Arduino Mega 2560 μέσω του μετατροπέα λογικών επιπέδων. Ονομασία ακροδεκτών στην υποδοχή MX 5267-04A του Odroid XU3 για την σειριακή επικοινωνία





Εικόνα 6.18 Δομή του μικροϋπολογιστή Odroid XU3 της εταιρείας Hardkernel

### Τεχνικά Χαρακτηριστικά Odroid-XU3

Επεξεργαστής	Samsung Exynos5422 ARM® Cortex™-A15 Quad 2.1GHz / Cortex™-A7 Quad 1.5GHz
Κύρια μνήμη	2Gbyte LPDDR3 RAM PoP stacked (933Mhz, 14.9GB/s memory bandwidth, 2x32bit bus)
3D επιταχυντής γραφικών	Mali™-T628 MP6 OpenGL ES 3.0 / 2.0 / 1.1 and OpenCL 1.1 Full profile
Διασύνδεση οθόνης	HDMI 1.4a - Standard Micro-HDMI connector (supports up to 1920 x 1080 resolution), DisplayPort 1.1 (supports up to 3840 x 2160 resolution)
Κωδικοποιητής ήχου	MAX98090 (a full-featured and high performance audio CODEC from Maxim)
Υποδοχές αποθηκευτικών μέσων	Micro-SD slot, eMMC 5.0 module connector
Θύρες	USB 3.0 Host x 1, USB 3.0 OTG x 1, USB 2.0 Host x 4
Τροφοδοσία	PWM for Cooler, Ethernet RJ-45, Headphone Jack, 30Pin : GPIO/IRQ/SPI/ADC
Διαστάσεις πλακέτας	5V, 4A
Βάρος	94 x 70 x 18 mm
	~72g

### 6.3 Θεωρητικό υπόβαθρο - Χρησιμοποιούμενοι αλγόριθμοι μηχανικής όρασης

#### 6.3.1 Κανονικοποίηση χρωματικών τιμών στον χώρο RGB

Το χρώμα ενός αντικειμένου σε μια εικόνα περιγραφόμενο σε κάποιο χώρο χρώματος εξαρτάται από τις συνθήκες φωτισμού του αντικειμένου, την τοποθέτηση του ως προς τις πηγές φωτός, την σκίαση του λόγω της θέσης του ως προς τα υπόλοιπα αντικείμενα της σκηνής. Σε σενάρια αναγνώρισης αντικειμένου με βάση το χρώμα είναι επιθυμητή η αφαίρεση της φωτεινότητας από μια χρωματική τιμή με στόχο μια περιγραφή του χρώματος αναλλοίωτη στις συνθήκες φωτισμού.

Αν  $(R, G, B)$  είναι οι συντεταγμένες ενός χρώματος στον χώρο  $RGB$  για την απαλοιφή της πληροφορίας της έντασης χρησιμοποιείται η κανονικοποίηση

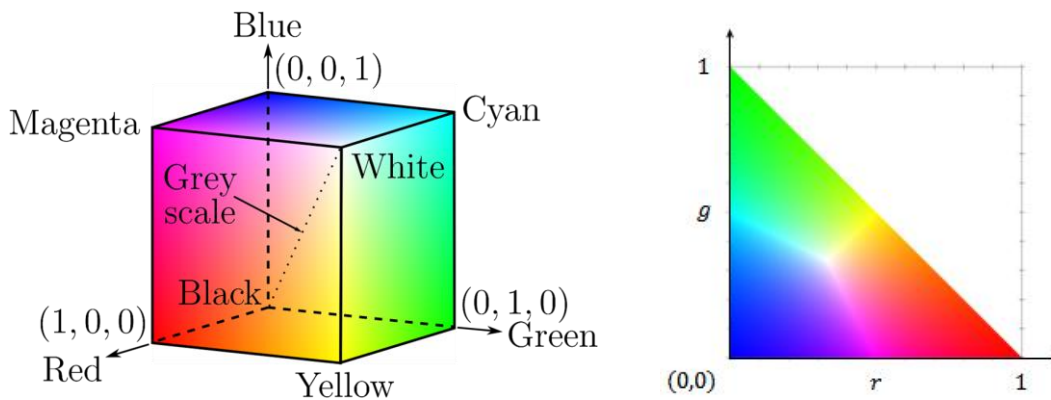
$$(r, g, b) = \left( \frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B} \right)$$

Για τις κανονικοποιημένες συντεταγμένες ισχύει πάντα

$$r + g + b = 1$$

Υποθέτοντας ένα γραμμικό μοντέλο για τον αισθητήρα λήψης, μια κλιμάκωση της έντασης του φωτισμού κατά παράγοντα  $s$  προκαλεί την ίδια ακριβώς κλιμάκωση στις μετρούμενες χρωματικές τιμές  $(sR, sG, sB)$ .

Θεωρώντας τις συνιστώσες ενός χρώματος σαν καρτεσιανές συνιστώσες σε έναν Ευκλείδειο τρισδιάστατο χώρο ορίζεται ένας τρισδιάστατος όγκος σε αυτό το γεωμετρικό χώρο. Κάθε συνιστώσα λαμβάνει μια τιμή στο διάστημα  $[0,1]$ . Στην αρχή των αξόνων στο σημείο  $(0,0,0)$  αντιστοιχεί το μαύρο χρώμα και το λευκό στην κορυφή  $(1,1,1)$ .



Εικόνα 6.19 Γεωμετρική απεικόνιση του χώρου RGB και της  $rg$  χρωματικότητας

Η προβολή των χρωματικών τιμών σε ένα χώρο δυο διαστάσεων ονομάζεται κανονικοποίηση χρώματος (color normalization) και αποσκοπεί στην αφαίρεση ενός βαθμού ελευθερίας από εκείνους που δημιουργούν το σύνολο των δυνατών χρωμάτων στο χρωματικό μοντέλο.

Τα διανύσματα στη νέα αναπαράσταση ανήκουν σε ένα επίπεδο. Ο δισδιάστατος χώρος που ορίζεται είναι γνωστός σαν  $rg$  χρωματικότητα ( $rg$  chromaticity). Το λευκό χρώμα στο διάγραμμα ορίζεται στο σημείο  $(1/3, 1/3)$ . Η κανονικοποίηση αυτή οδηγεί σε χρωματικές αναπαραστάσεις ανεξάρτητες της έντασης του φωτισμού.



Εικόνα 6.20 Κανονικοποίηση στο χώρο RGB και αναπαράσταση της rg χρωματικότητας

### 6.3.2 Τμηματοποίηση εικόνας - Μέθοδος κατωφλίωσης Otsu

Τμηματοποίηση (segmentation) εικόνας είναι η διαδικασία με την οποία διαχωρίζεται μια εικόνα σε κατάλληλες περιοχές ή αντικείμενα. Για την ανάπτυξη των τεχνικών διαχωρισμού χρησιμοποιούνται οι ιδιότητες της ασυνέχειας και της ομοιότητας. Η εύρεση κατωφλίων είναι μια δημοφιλής τεχνική για την τμηματοποίηση εικόνων που απεικονίζονται με αποχρώσεις του γκρι.

Η μέθοδος Otsu είναι ίσως η καλύτερη τεχνική εύρεσης κατωφλίου αν θέλουμε να διαχωρίσουμε μόνο δυο περιοχές φωτεινότητας (δυαδικό πρόβλημα). Η μέθοδος επιλέγει αυτοματοποιημένα το κατώφλι διαχωρίζοντας τα εικονοστοιχεία σε δυο ομάδες (παρασκήνιο, προσκήνιο) οι οποίες διαφέρουν στα επίπεδα του γκρι. Για κάθε υποψήφιο κατώφλι επιδιώκεται η μεγιστοποίηση της διακύμανσης μεταξύ των επιπέδων γκρι των δυο ομάδων.

Έστω ότι τα εικονοστοιχεία μιας δοθείσας εικόνας διαβαθμίζονται σε  $L$  επίπεδα του γκρι  $[1, 2, \dots, L]$ . Το πλήθος των εικονοστοιχείων της γκρι διαβάθμισης  $i$  συμβολίζεται με  $n_i$  και ο συνολικός αριθμός των εικονοστοιχείων της εικόνας είναι  $N = n_1 + n_2 + \dots + n_L$ . Η μέση τιμή των επιπέδων του γκρι της εικόνας θα είναι

$$\mu_T = \mu(L) = \sum_{i=1}^L i p_i$$

όπου  $p_i$  η πιθανότητα εμφάνισης της γκρι διαβάθμισης  $i$

$$p_i = \frac{n_i}{N}, \quad p_i \geq 0, \quad \sum_{i=1}^L p_i = 1$$

Αν χωριστεί η εικόνα σε δυο κλάσεις  $C_0$  και  $C_1$  χρησιμοποιώντας ένα κατώφλι στο επίπεδο  $k$  η κλάση  $C_0$  εμπεριέχει τα εικονοστοιχεία με διαβαθμίσεις του γκρι  $[1, 2, \dots, k]$  και η  $C_1$  εκείνα τα εικονοστοιχεία με επίπεδα  $[k + 1, \dots, L]$ . Οι πιθανότητες  $\omega_0, \omega_1$  ένα εικονοστοιχείο να ανήκει σε μια από τις δυο κλάσεις  $C_0, C_1$  είναι αντίστοιχα

$$\omega_0 = \sum_{i=1}^k p_i = \omega(k)$$

$$\omega_1 = \sum_{i=k+1}^L p_i = 1 - \omega(k)$$

και η δειγματική μέση τιμή (expected value) για κάθε κλάση

$$\mu_0 = \sum_{i=1}^k i \frac{p_i}{\omega_0} = \frac{\mu(k)}{\omega(k)}$$

$$\mu_1 = \sum_{i=k+1}^L i \frac{p_i}{\omega_1} = \frac{\mu_T - \mu(k)}{1 - \omega(k)}$$

με  $\omega_0 + \omega_1 = 1$  και  $\omega_0\mu_0 + \omega_1\mu_1 = \mu_T$

Οι διακυμάνσεις για κάθε μια κλάση δίνονται από τις σχέσεις

$$\sigma_0^2 = \sum_{i=1}^k (i - \mu_0)^2 \frac{p_i}{\omega_0}$$

$$\sigma_1^2 = \sum_{i=k+1}^L (i - \mu_1)^2 \frac{p_i}{\omega_1}$$

Για να αποτιμηθεί η χρησιμότητα του κατωφλίου καθορίζονται τα κάτωθι μέτρα διαχωριστικότητας των δυο κλάσεων και αναζητείται η μεγιστοποίηση τους

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}, \quad \kappa = \frac{\sigma_T^2}{\sigma_W^2}, \quad \eta = \frac{\sigma_B^2}{\sigma_T^2}$$

όπου

$$\sigma_W^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2$$

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 = \omega_0\omega_1(\mu_0 - \mu_1)^2$$

$$\sigma_T^2 = \sum_{i=1}^L (i - \mu_T)^2 p_i$$

ορίζονται αντίστοιχα η ενδο-διακύμανση  $\sigma_W^2$  (within-class variance) και η διακύμανση μεταξύ των κλάσεων  $\sigma_B^2$  (between-class variance) και η συνολική διακύμανση της εικόνας  $\sigma_T^2$ . Η μεγιστοποίηση ενός εκ των μέτρων διαχωριστικότητας προκύπτει για κατώφλι  $k$  για το οποίο ελαχιστοποιείται η  $\sigma_W^2$  και μεγιστοποιείται η  $\sigma_B^2$ . Η συνολική διακύμανση  $\sigma_T^2$  δεν εξαρτάται από την επιλογή του κατωφλιού. Επιλέγεται ως κριτήριο το μέτρο διαχωριστικότητας  $\eta(k)$  οπότε

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_T^2}$$

$$\sigma_B^2(k) = \frac{[\mu(k) - \mu_T\omega(k)]^2}{\omega(k)[1 - \omega(k)]}$$

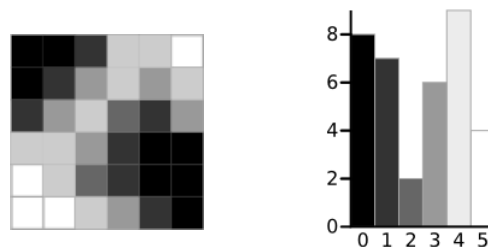
Για το βέλτιστο κατώφλι  $k^*$  στο οποίο μεγιστοποιείται η διακύμανση μεταξύ των επιπέδων γκρι των δυο κλάσεων θα είναι

$$\sigma_B^2(k^*) = \max_{1 \leq k \leq L} \sigma_B^2(k)$$

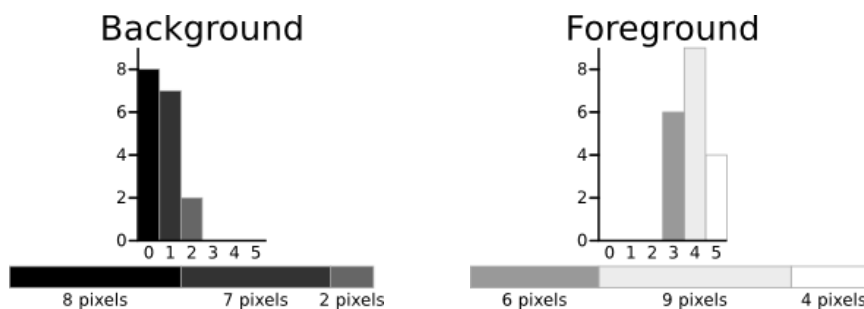
Ξεκινώντας από το αρχικό επίπεδο γκρι της εικόνας  $k = 1$  και καταλήγοντας στο επίπεδο  $L$  υπολογίζεται η  $\sigma_B^2$  και επιλέγεται το κατώφλι για το οποίο η  $\sigma_B^2$  γίνεται μέγιστη.

### 6.3.2.1 Παράδειγμα εφαρμογής της μεθόδου Otsu

Θεωρούμε μια εικόνα διαστάσεων 6x6 με καθένα από τα εικονοστοιχεία της να έχουν ένα από τα εξής 6 χρώματα: μαύρο, 4 διαβαθμίσεις του γκρι και άσπρο.



Εικόνα 6.21 Η προς εξέταση 6x6 εικόνα και το ιστόγραμμα της



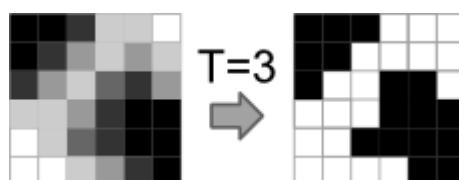
Εικόνα 6.22 Διαχωρισμός εικονοστοιχείων παρασκήνιου και προσκήνιου με κατώφλι T=3

<i>Background</i>	
<i>Weight</i>	$\omega_b = \frac{8 + 7 + 2}{36} = 0.4722$
<i>Mean</i>	$\mu_b = \frac{0 * 8 + 1 * 7 + 2 * 2}{8 + 7 + 2} = 0.6471$
<i>Variance</i>	$\sigma_b^2 = \frac{(0 - \mu_b)^2 * 8 + (1 - \mu_b)^2 * 7 + (2 - \mu_b)^2 * 2}{8 + 7 + 2} = 0.4637$
<i>Foreground</i>	
<i>Weight</i>	$\omega_f = \frac{6 + 9 + 4}{36} = 0.5278$
<i>Mean</i>	$\mu_f = \frac{3 * 6 + 4 * 9 + 5 * 4}{6 + 9 + 4} = 3.8947$
<i>Variance</i>	$\sigma_f^2 = \frac{(3 - \mu_f)^2 * 6 + (4 - \mu_f)^2 * 9 + (5 - \mu_f)^2 * 4}{6 + 9 + 4} = 0.5152$

Το επόμενο βήμα είναι ο υπολογισμός της διακύμανσης μεταξύ των επιπέδων γκρι των δυο κλάσεων

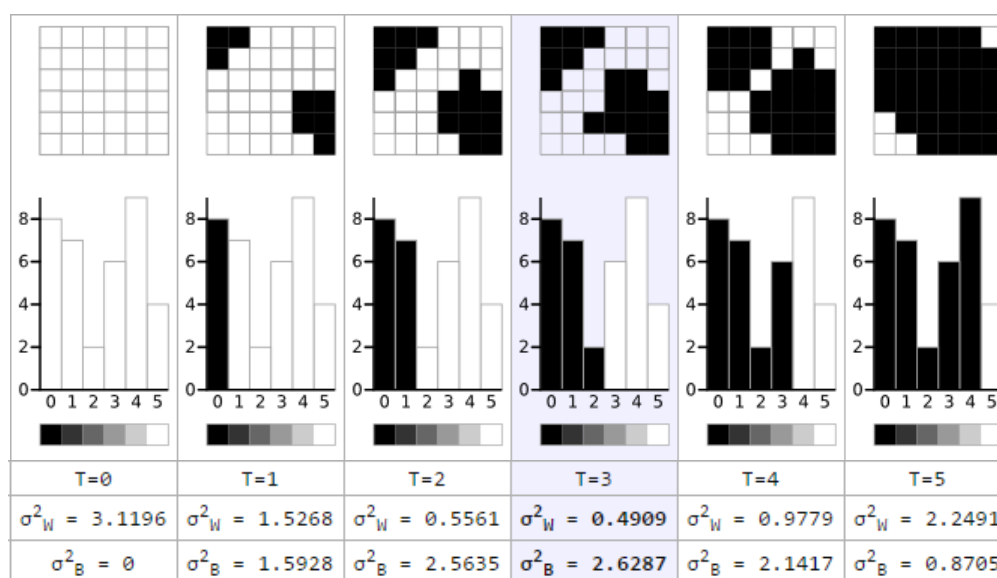
$$\sigma_B^2 = \omega_b \omega_f (\mu_b - \mu_f)^2 = 2.6287$$

Κάθε εικονοστοιχείο με διαβάθμιση του γκρι μικρότερη του 3 ανήκει στο φόντο και εκείνα με διαβάθμιση μεγαλύτερη ή ίση του 3 ανήκουν στο προσκήνιο.



Εικόνα 6.23 Κατάτμηση της 6x6 εικόνας με κατώφλι T=3

Για κάθε υποψήφιο κατώφλι το ιστόγραμμα της εικόνας χωρίζεται σε δυο περιοχές-κλάσεις. Από τις πιθανές τιμές κατωφλίωσης T=0 έως και T=5 επιλέγεται εκείνη με την μέγιστη διακύμανση  $\sigma_B^2$  μεταξύ των δυο κλάσεων.



Εικόνα 6.24 Προσδιορισμός κατωφλίου στην 6x6 εικόνα με την μέθοδο του Otsu

### 6.3.3 Χρωματικοί δείκτες βλάστησης στο ορατό φάσμα

Οι χρωματικοί δείκτες βλάστησης τονίζουν και αναδεικνύουν το πράσινο χρώμα των φυτών σε μια εικόνα ώστε να επιτευχθεί ο διαχωρισμός του εδάφους και των λοιπών στοιχείων (π.χ. πέτρες, αποξηραμένα φυτά) από την χλωρή φυτική μάζα. Η τμηματοποίηση αυτή αποτελεί ανοικτό πρόβλημα και ο αυξημένος βαθμός δυσκολίας οφείλεται σε διάφορους παράγοντες όπως για παράδειγμα στην γωνία φωτισμού, την γωνία λήψης σε σχέση με την επιφάνεια του εδάφους και την θέση του ήλιου, στις σκιές που δημιουργούνται από τα ίδια τα φυτά, το βαθμό ανάπτυξης των φυτών, τις διακυμάνσεις στο φυσικό φωτισμό που οφείλονται είτε στην εναλλαγή των καιρικών συνθηκών είτε στο πέρασμα της ώρας (π.χ σούρουπο).

Σε όλες τις ακόλουθες προσεγγίσεις απαιτείται η εφαρμογή κατωφλίου στην προκύπτουσα grayscale εικόνα για το τελικό στάδιο της κατάτμησης. Επικρατέστερη μέθοδος κατωφλίωσης αποτελεί η μέθοδος του Otsu (1979) αν και αυξάνει τον υπολογιστικό φόρτο λόγω του ότι η εύρεση του κατωφλίου επιτυγχάνεται μέσα από μια επαναληπτική διαδικασία. Με την μέθοδο αυτή η τιμή του κατωφλίου προκύπτει λίγο μεγαλύτερη από την μέση τιμή της έντασης των εικονοστοιχείων με αποτέλεσμα η ανιχνευθείσα βλάστηση στην εικόνα να εμφανίζεται ελαφρώς λεπτυσμένη.

### Excess Green vegetation index

Οι Woebbecke et al. (1995) διαπίστωσαν ότι ο δείκτης  $ExG = 2g - r - b$  αποδίδει μια εικόνα με διαβαθμίσεις του γκρι σχεδόν διαχωρισμένη με έμφαση (φωτεινές τονικά περιοχές) στα εικονοστοιχεία πράσινου χρώματος και με τα υπόλοιπα στοιχεία να κατηγοριοποιούνται στο φόντο. Για την εξαγωγή του δείκτη οι τιμές φωτεινότητας των εικονοστοιχείων κάθε καναλιού στο χρωματικό μοντέλο RGB ανάγονται στο διάστημα από 0 έως και 1:

$$R^* = \frac{R}{R_{max}}, \quad G^* = \frac{G}{G_{max}}, \quad B^* = \frac{B}{B_{max}}$$

για εικόνες με 8-bit βάθος χρώματος οι μέγιστες τιμές τονικότητας για κάθε βασικό χρώμα είναι  $R_{max} = G_{max} = B_{max} = 255$ . Οι χρωματικές συνιστώσες κανονικοποιούνται ώστε να μειωθούν οι επιδράσεις των συνθηκών φωτισμού

$$r = \frac{R^*}{R^* + G^* + B^*}, \quad g = \frac{G^*}{R^* + G^* + B^*}, \quad b = \frac{B^*}{R^* + G^* + B^*}$$

### Excess Green minus Excess Red

Για την αποτίμηση της ερυθρότητας οι Meyer et al. (1998) πρότειναν τον δείκτη Excess Red:

$$ExR = 1.4r - b$$

Ο δείκτης βλάστησης  $ExG - ExR$  που πρότειναν οι Meyer et al. (2008) δεν απαιτεί την εύρεση της τιμής κατωφλίου για την μετατροπή της εικόνας σε δυαδική. Τα εικονοστοιχεία που αντιστοιχούν στην χλωρή φυτική μάζα έχουν θετική τιμή και όλα τα υπόλοιπα τα οποία κατηγοριοποιούνται στο παρασκήνιο αρνητική ή μηδέν.

### Τροποποιημένος δείκτης ExG

Οι συντελεστές ( $c_r = -1, c_g = 2, c_b = -1$ ) στο δείκτη βλάστησης ExG τροποποιήθηκαν από τους Burgos-Artizzu et al. (2010) σε ( $c_r = -0.884, c_g = 1.262, c_b = -0.311$ ) με την βοήθεια γενετικού αλγόριθμου ώστε να βελτιστοποιηθεί η απόδοση του. Ως κατώφλι προτάθηκε ο μέσος όρος της φωτεινότητας των εικονοστοιχείων.

$$threshold = \frac{\sum_{x=1}^N \sum_{y=1}^M (c_r * R(x, y) + c_g * G(x, y) + c_b * B(x, y))}{M * N}$$

### Color index of vegetation extraction

Οι Kataoka et al. (2003) πρότειναν τον δείκτη:

$$CIVE = 0.441r - 0.811g + 0.385b + 18.78745$$

### Vegetative Index

Για να αντιμετωπιστεί η μεταβλητότητα του φυσικού φωτισμού κατά την αποτίμηση της βλάστησης οι Hague et al. (2006) συνέστησαν τον δείκτη

$$VEG = \frac{g}{r^a b^{1-a}}$$

με το  $a$  να δίνεται ίσο με 0.667

### Συγχώνευση των δεικτών ExG, ExGR, CIVE και VEG

Για την διεξαγωγή των πειραμάτων και την δημιουργία ενός δείκτη βλάστησης, ο οποίος θα συνδυάζει αποδοτικά ήδη διατυπωμένους σε μελέτες δείκτες που παρουσιάζουν υπό προϋποθέσεις καλές επιδόσεις, χρησιμοποιήθηκε κατά πλειοψηφία σύνολο εικόνων με πανοραμικές λήψεις σε προοπτική στις οποίες εκτός της βλάστησης και του καλλιεργημένου εδάφους καταγράφονταν και μέρος του ορίζοντα. Κατά την τμηματοποίηση καταβλήθηκε προσπάθεια ανίχνευσης τόσο της βλάστησης και του εδάφους αλλά και μέρους του ουρανού. Η εστίαση στον ορίζοντα αποσκοπεί στην αποκόμιση απαραίτητων πληροφοριών και από τον αισθητήρα λήψης εικόνων για την πλοήγηση των αυτόνομων ρομποτικών οχημάτων μέσα στην καλλιεργημένη έκταση.

Οι δείκτες βλάστησης οι οποίοι επιλέχθηκαν (ExG, ExGR, CIVE και VEG) παράγουν εικόνες σε αποχρώσεις του γκρι στις οποίες ενίοτε η βλάστηση απεικονίζεται υπερεκτεταμένη ή υπό ανιχνεύεται ανάλογα και με τις συνθήκες φωτισμού. Τα εικονοστοιχεία τα οποία αντιστοιχούν σε χλωρή φυτική μάζα αν και εμφανίζονται σε κάθε περίπτωση πανομοιότυπα και παρουσιάζουν γενικά υψηλές τιμές (ανοιχτόχρωμα) διαφοροποιούνται σε επίπεδα φωτεινότητας από δείκτη σε δείκτη. Σε μια προσπάθεια βελτιστοποίησης οι Guijarro et al. (2011) δημιούργησαν δείκτη ο οποίος αυτοματοποιημένα συγχωνεύει μέσω γραμμικού συνδυασμού τις εικόνες οι οποίες παράγονται από κάθε έναν από τους προαναφερθέντες δείκτες.

$$COM = \sum_{\Omega} w_{G_k} G_k = w_{G_{ExG}} * ExG + w_{G_{ExGR}} * ExGR + w_{G_{CIVE}} * CIVE + w_{G_{VEG}} * VEG$$

με  $\Omega = \{G_k | k \equiv ExG, ExGR, CIVE, VEG\}$

Οι grayscale εικόνες που παρουσιάζουν μεγαλύτερη αντίθεση (contrast) συνεισφέρουν με μεγαλύτερο ποσοστό συμμετοχής στην προκύπτουσα συνδυασμένη εικόνα του δείκτη COM. Κριτήριο για την αποτίμηση της αντίθεσης σε μια grayscale εικόνα αποτελεί ο υπολογισμός είτε της εντροπίας (μεταβλητότητας) είτε της ομοιομορφίας (uniformity) του ιστογράμματος (Gonzalez et al. 2004; Gonzalez and Woods, 2008). Η ομοιομορφία μεγιστοποιείται όταν όλα τα εικονοστοιχεία κατατάσσονται στο ίδιο επίπεδο τονικότητας ενώ σε αυτήν την περίπτωση η εντροπία ελαχιστοποιείται. Συνεπώς αναζητούνται εικόνες είτε με ελάχιστη ομοιομορφία είτε μέγιστη εντροπία. Προτιμάται η εύρεση της ομοιομορφίας για λόγους μειωμένου υπολογιστικού φόρτου.

$$U = \sum_{i=0}^{L-1} p^2(z_i)$$

όπου  $L$  τα επίπεδα διαβάθμισης του γκρι της εικόνας και  $p(z_i)$  η τιμή του ιστογράμματος στην κανονικοποιημένη διαβάθμιση γκρι επιπέδου  $z_i$  στο διάστημα  $[0,1]$ . Για τον υπολογισμό των κανονικοποιημένων στο  $[0,1]$  συντελεστών βαρύτητας  $w_{G_k}$  υπολογίζεται η ομοιομορφία  $U_{G_k}$  κάθε εικόνας για κάθε έναν δείκτη βλάστησης και κατόπιν αυτή ανάγεται στο  $[0,1]$

$$U_{G_k} = \frac{1 - U_{G_k}}{\sum_{G_h \in \Omega} U_{G_h}}, \quad w_{G_k} = \frac{U_{G_k}}{\sum_{G_h \in \Omega} U_{G_h}}$$

Για κάθε εικόνα η οποία συνδυάζεται προκύπτουν τέσσερις συντελεστές βαρύτητας μια για κάθε δείκτη. Η μεγιστοποίηση ενός συντελεστή βαρύτητας δηλώνει ελαχιστοποίηση της ομοιομορφίας της αντίστοιχης εικόνας. Από το μέσο όρο των τιμών αυτών σε κάθε δείκτη για το σύνολο των εικόνων προέκυψε στην συγκεκριμένη μελέτη ότι

$$w_{G_{ExG}} = 0.25, \quad w_{G_{ExGR}} = 0.3, \quad w_{G_{CIVE}} = 0.33, \quad w_{G_{VEG}} = 0.12$$



Για την κατωφλίωση επιλέχθηκε ο μέσος όρος φωτεινότητας των εικονοστοιχείων λόγω του ότι η τιμή κατωφλίου με την μέθοδο του Otsu προέκυπτε μεγαλύτερη με συνέπεια αρκετά πράσινα εικονοστοιχεία να μην κατηγοριοποιούνται ως βλάστηση.

$$m = \sum_{i=0}^{L-1} z_i p(z_i)$$

### Συγχώνευση των δεικτών ExG, CIVE και VEG

Σε μεταγενέστερη μελέτη των Guerrero et al. (2012) διαπιστώθηκε ότι στο σύνολο των εικόνων που χρησιμοποιήθηκαν ο δείκτης ExGR εσφαλμένα προσδιόριζε τις σκιές που δημιουργούνται από τα φυτά ως βλάστηση οπότε αποκλείστηκε από τον γραμμικό συνδυασμό και οι κανονικοποιημένοι στο διάστημα [0,1] πολλαπλασιαστές συμμετοχής των υπολοίπων δεικτών αναπροσαρμόστηκαν ανάλογα.

$$COM = 0.36 * ExG + 0.47 * CIVE + 0.17 * VEG$$

### Normalized Difference Vegetation Index (NDVI)

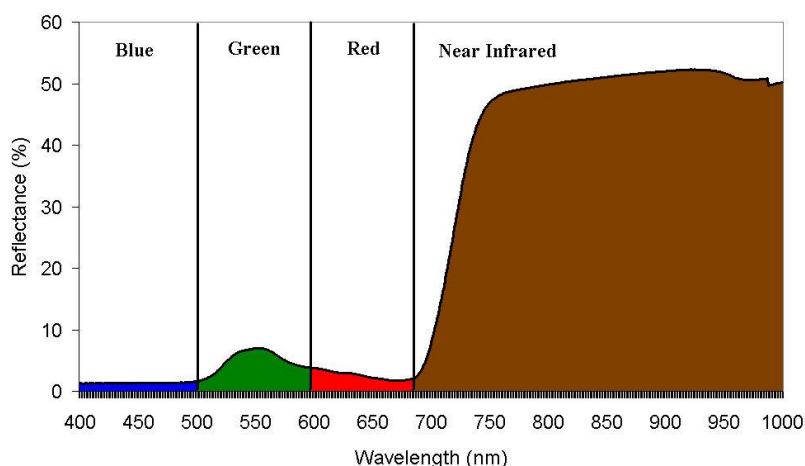
Ο δείκτης βλάστησης κανονικοποιημένης διαφοράς στο ορατό φάσμα χρησιμοποιεί μόνο το πράσινο και κόκκινο χρωματικό κανάλι

$$NDVI = \frac{G - R}{G + R}$$

Αποτελεί εξέλιξη του δείκτη βλάστησης για την δορυφορική τηλεπισκόπηση ο οποίος εκμεταλλεύεται τη διαφορά της ανάκλασης μεταξύ της ορατής και κοντινής υπέρυθρης περιοχής. Η χλωροφύλλη, απορροφά σε υψηλά ποσοστά το ορατό φως (από 0,4 έως 0,7 μm) στο ερυθρό για χρήση κατά τη φωτοσύνθεση. Από την άλλη πλευρά, η κυτταρική δομή των φύλλων ανακλά σε μεγάλο βαθμό το εγγύς υπέρυθρο φως (από 0,7 έως 1,1 μm). Εάν η ανακλώμενη ακτινοβολία στο εγγύς υπέρυθρο μήκος κύματος είναι πολύ περισσότερη από ότι στο ορατό μήκος φωτός, τότε η βλάστηση στο συγκεκριμένο εικονοστοιχείο είναι πιθανό να είναι πυκνή. Ο δείκτης NDVI για την δορυφορική τηλεπισκόπηση προκύπτει από τον λόγο της διαφοράς της ανακλαστικότητας στο εγγύς υπέρυθρο  $R_{NIR}$  και ερυθρό  $R_{red}$  προς το άθροισμα τους:

$$NDVI = \frac{R_{NIR} - R_{red}}{R_{NIR} + R_{red}}$$

Λαμβάνει τιμές από -1 (μηδενική φυτοκάλυψη) μέχρι +1 (πλούσια βλάστηση).



Εικόνα 6.25 Φασματική υπογραφή βλάστησης. Η βλάστηση έχει χαμηλή ανάκλαση στην ορατή περιοχή και υψηλή ανάκλαση στο κοντινό υπέρυθρο.



Original image



ExG + Otsu



ExGR + Positive threshold



Modified ExG + mean value



CIVE + Otsu



Combined (ExG, ExGR, CIVE, VEG) + mean value



VEG + mean value



Combined (ExG, CIVE, VEG) + Otsu



NDVI + Otsu

**Παράδειγμα εφαρμογής 1:** Τμηματοποίηση εικόνας με τη χρήση δεικτών βλάστησης



Original image



ExG + Otsu



ExGR + Positive threshold



Modified ExG + mean value



CIVE + Otsu



Combined (ExG, ExGR, CIVE, VEG) + mean value



VEG + mean value



Combined (ExG, CIVE, VEG) + Otsu



NDVI + Otsu

**Παράδειγμα εφαρμογής 2:** Τμηματοποίηση εικόνας με τη χρήση δεικτών βλάστησης



Original image



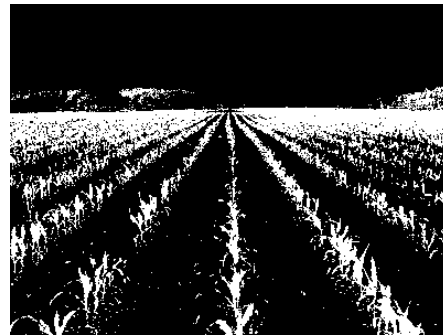
ExG + Otsu



ExGR + Positive threshold



Modified ExG + mean value



CIVE + Otsu



Combined (ExG, ExGR, CIVE, VEG) + mean value



VEG + mean value

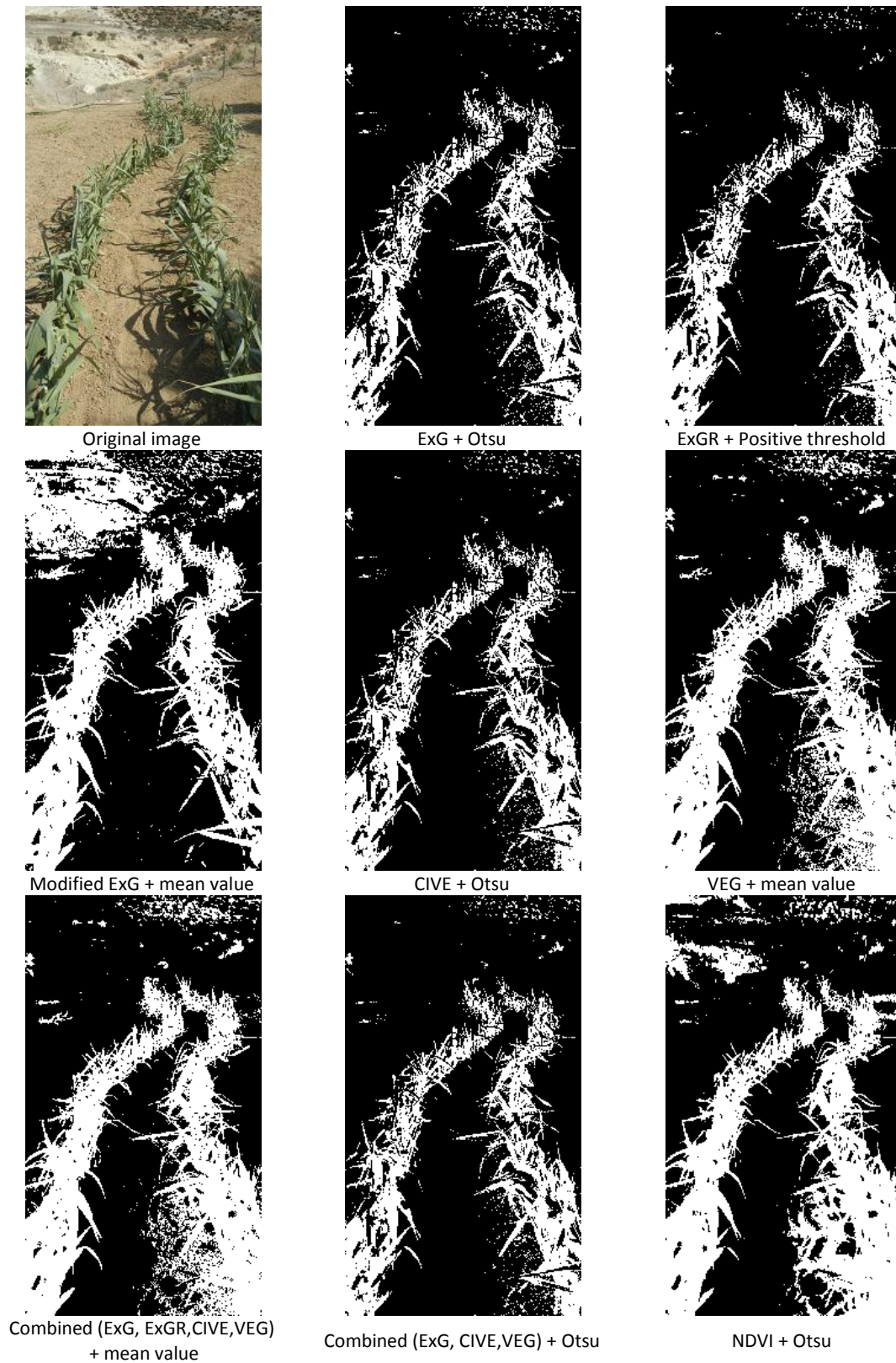


Combined (ExG, CIVE, VEG) + Otsu



NDVI + Otsu

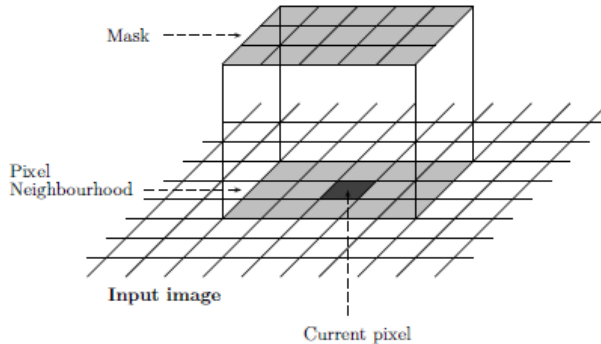
**Παράδειγμα εφαρμογής 3:** Τμηματοποίηση εικόνας με τη χρήση δεικτών βλάστησης



**Παράδειγμα εφαρμογής 4:** Τμηματοποίηση εικόνας με τη χρήση δεικτών βλάστησης

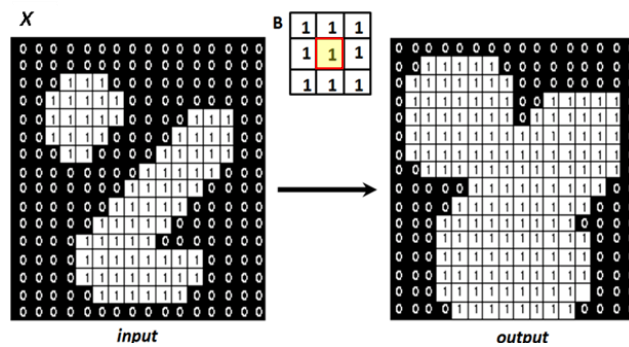
### 6.3.4 Μορφολογική επεξεργασία δυαδικών εικόνων

Οι πιο βασικοί τελεστές είναι η διαστολή (dilation) και η διάβρωση (erosion). Δέχονται δύο ορίσματα σαν είσοδο: μια εικόνα  $f(x, y)$ , και ένα δομικό στοιχείο (μάσκα ή πυρήνας)  $s(x, y)$ . Το δομικό στοιχείο είναι δυαδικός πίνακας μικρότερων διαστάσεων από την εικόνα οι μονάδες του οποίου σχηματίζουν οποιοδήποτε γεωμετρικό σχήμα με καθορισμένο κέντρο αναφοράς (anchor point). Τα ακριβή αποτελέσματα εξαρτώνται από το δομικό στοιχείο που θα επιλεγεί. Όσο πιο μεγάλο είναι τόσο πιο έντονη θα είναι η επίδραση του.



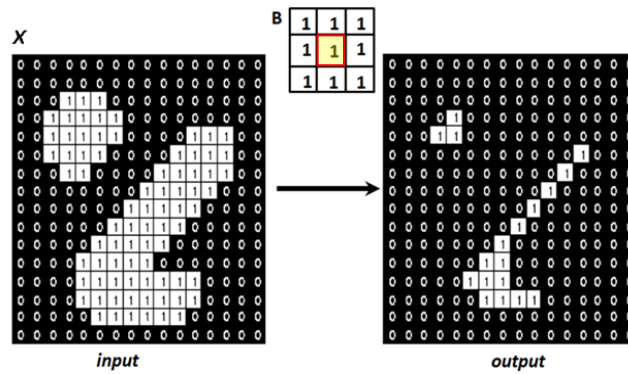
Εικόνα 6.26 Εφαρμογή μορφολογικού τελεστή σε δυαδική εικόνα

Η διαστολή  $f \oplus s$  είναι η συνέλιξη μιας εικόνας ή ενός τμήματος αυτής με κάποιο δομικό στοιχείο. Έστω και αν ένα εικονοστοιχείο με τιμή 1 του δομικού στοιχείου βρίσκεται μέσα στο τμήμα της εικόνας στο οποίο εφαρμόζεται τότε το εικονοστοιχείο της εικόνας το οποίο συμπίπτει με το anchor point γίνεται 1 διαφορετικά γίνεται 0. Μετά την εφαρμογή η βασική αλλαγή είναι η διεύρυνση (διαστολή) των συνόρων των περιοχών με τα λευκά εικονοστοιχεία.



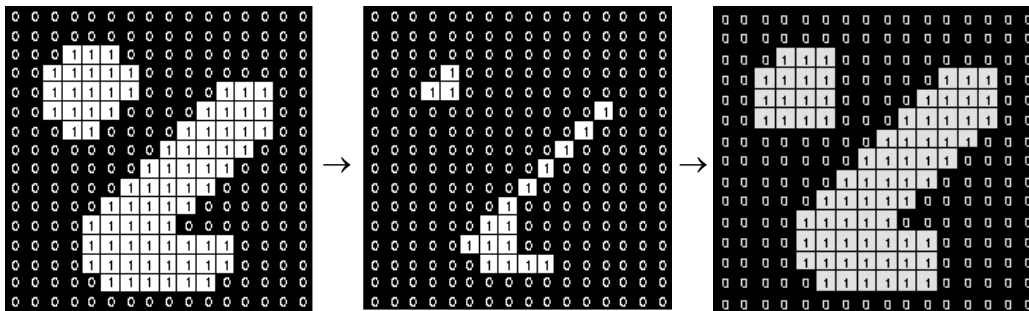
Εικόνα 6.27 Επίδραση της διαστολής (dilation) σε δυαδική εικόνα με δομικό στοιχείο τον 3x3 πίνακα B

Η βασική επίδραση του τελεστή της διάβρωσης (erosion)  $f \ominus s$  (ή συστολής) είναι να σμίκρυνση των αντικείμενων της εικόνας. Εάν ολόκληρο το δομικό στοιχείο (τα εικονοστοιχεία του με τιμή 1) εμπεριέχεται στο τμήμα της εικόνας στο οποίο εφαρμόζεται τότε το εικονοστοιχείο της εικόνας που συμπίπτει με το anchor point γίνεται 1 διαφορετικά γίνεται 0. Σε γενικές γραμμές η διαστολή τείνει να μεγαλώσει την περιοχή στην οποία εφαρμόζεται ενώ η συστολή να την σμικρύνει.



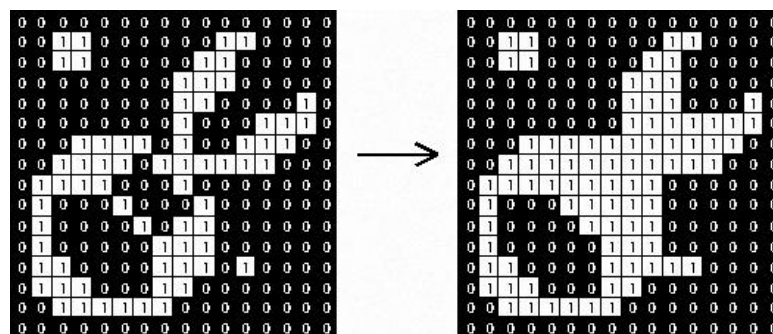
Εικόνα 6.28 Επίδραση της διάβρωσης (erosion) σε δυαδική εικόνα με δομικό στοιχείο τον 3x3 πίνακα B

Η εφαρμογή του τελεστή της διάβρωσης σε μια εικόνα ακολουθούμενη από μια διαστολή αναφέρεται ως μορφολογική πράξη του ανοίγματος (**opening**). Υλοποιείται με διάβρωση της  $f$  από το  $s$  ακολουθούμενη από μια διαστολή του αποτελέσματος με το ίδιο δομικό στοιχείο  $f \circ s = (f \ominus s) \oplus s$ . Η μορφολογική πράξη του ανοίγματος εξομαλύνει το περίγραμμα ενός αντικειμένου και εξαλείφει τις προεξοχές.



Εικόνα 6.29 Λειτουργία opening

Η εφαρμογή του τελεστή της διαστολής σε μια δυαδική εικόνα ακολουθούμενη από μια συστολή του αποτελέσματος με το ίδιο δομικό στοιχείο αναφέρεται ως λειτουργία του κλεισίματος (**closing**)  $f \cdot s = (f \oplus s) \ominus s$ . Η μορφολογική πράξη του κλεισίματος συμπληρώνει τις σπές με διαστάσεις μικρότερες του δομικού στοιχείου, συγχωνεύει στενά θραύσματα διατηρώντας το αρχικό μέγεθος του αντικειμένου.



Εικόνα 6.30 Λειτουργία closing

### 6.3.5 Υπολογισμός κέντρου μάζας (centroid)

Για μια εικόνα σε αποχρώσεις του γκρι με εικονοστοιχεία φωτεινότητας  $I(x, y)$  οι γεωμετρικές ροπές  $m_{ji}$  υπολογίζονται ως εξής

$$m_{ji} = \sum_x \sum_y I(x, y) y^j x^i$$

Οι συντεταγμένες του κέντρου μάζας θα είναι

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

όπου

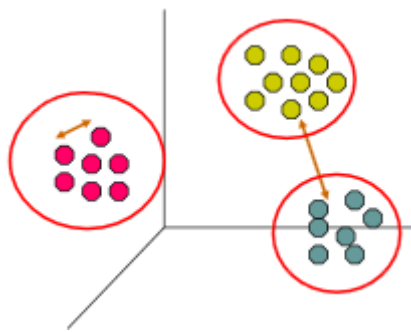
$$m_{10} = \sum_x \sum_y I(x, y) x$$

$$m_{01} = \sum_x \sum_y I(x, y) y$$

$$m_{00} = \sum_x \sum_y I(x, y)$$

### 6.3.6 Διαχωριστική συσταδοποίηση σημείων - Αλγόριθμος k-means

Μια συστάδα (ομάδα) είναι ένα σύνολο από σημεία κάθε ένα από τα οποία βρίσκεται κοντύτερα στα υπόλοιπα σημεία της συστάδας που ανήκει από ότι με οποιοδήποτε άλλο σημείο που δεν ανήκει στη συστάδα. Συσταδοποίηση είναι ο διαμερισμός των στοιχείων ενός συνόλου σε μη επικαλυπτόμενα υποσύνολα ώστε κάθε στοιχείο να ανήκει σε ακριβώς ένα υποσύνολο. Οι αποστάσεις μέσα στη συστάδα ελαχιστοποιούνται και ανάμεσα στις συστάδες μεγιστοποιούνται. Κάθε συστάδα συσχετίζεται με ένα κεντρικό σημείο (centroid). Το κεντρικό σημείο είναι συνήθως το μέσο (mean) των σημείων της συστάδας το οποίο μπορεί να μην είναι ένα από τα δεδομένα εισόδου.



Εικόνα 6.31 K-means clustering

Η διαδικασία ομαδοποίησης ενός συνόλου δεδομένων με βάση τον διαχωριστικό αλγόριθμο k-means είναι απλή αρκεί να είναι εκ των προτέρων καθορισμένος ο αριθμός  $k$  των ομάδων (clusters) που θα προκύψουν. Ο αριθμός των ομάδων αποτελεί είσοδο στον αλγόριθμο. Δοθέντος ενός συνόλου  $n$  σημείων  $X = \{x_1, x_2, \dots, x_n\}$  ζητούνται  $k$  σύνολα-ομάδες  $C_1, C_2, \dots, C_k$  με  $k \ll n$  έτσι ώστε  $C_i \neq \emptyset \forall i = 1, 2, \dots, k$  και οι  $k$  ομάδες αποτελούν διαμέριση του συνόλου  $X$ . Κάθε σημείο ανήκει σε μια και μόνο ομάδα.



Η αρχική θέση των centroids επηρεάζει το αποτέλεσμα που θα δώσει ο αλγόριθμος. Για αυτό συνίσταται η εκτέλεση του πολλές φορές ώστε να μειωθεί η επίδραση αυτή. Συχνά θεωρείται καλύτερη η επιλογή εκείνων των centroids που απέχουν μεταξύ τους όσο περισσότερο γίνεται.

Το επόμενο βήμα είναι η επιλογή κάθε στοιχείου από το σύνολο των δεδομένων και η συσχέτιση του με το κοντινότερο σε αυτό centroid. Η εγγύτητα των σημείων υπολογίζεται με βάση κάποια απόσταση. Επειδή η απόσταση υπολογίζεται συχνά ο υπολογισμός της πρέπει να είναι σχετικά απλός. Εφόσον τα clusters είναι σαφώς διαχωρισμένα μεταξύ τους μπορεί να χρησιμοποιηθεί σαν μέτρο απόστασης η Ευκλείδεια απόσταση. Σε κάθε βήμα ελέγχεται αν το στοιχείο  $x$  ανήκει στο cluster  $i$ , αν η απόσταση του από το centroid του  $i$  cluster είναι μικρότερη σε σχέση με όλες τις άλλες αποστάσεις από τα centroid των υπολοίπων clusters. Όταν αυτό πραγματοποιηθεί για όλα τα σημεία του συνόλου δεδομένων έχει προκύψει μια πρώτη ομαδοποίηση.

Στη συνέχεια επανα - υπολογίζονται  $k$  νέα centroids τα οποία αποτελούν το κέντρο βάρους για κάθε ένα cluster που πρόεκυψε από το προηγούμενο βήμα. Αφού οριστούν τα νέα  $k$  centroids ακολουθεί η ίδια διαδικασία ανάθεσης κάθε ενός στοιχείου του συνόλου των δεδομένων στο κοντινότερο σε αυτό νέο centroid. Οι συστάδες που παράγονται διαφέρουν από το ένα τρέξιμο του αλγορίθμου στο άλλο.

Όταν σε κάποια επανάληψη δεν σημειωθούν μετακινήσεις σημείων και τα cluster παραμένουν αμετάβλητα τότε τερματίζει η εκτέλεση του αλγορίθμου (ή η απόσταση μεταξύ των νέων centroids από τα προηγούμενα να είναι μικρή). Ο αλγόριθμος στοχεύει να ελαχιστοποιήσει μια αντικειμενική συνάρτηση. Η πιο διαδεδομένη στρατηγική ομαδοποίησης είναι το άθροισμα των τετραγώνων των σφαλμάτων (Sum of Squared Error)

$$SSE = \sum_{i=1}^k \sum_{x \in C_i} \|x - m_i\|^2$$

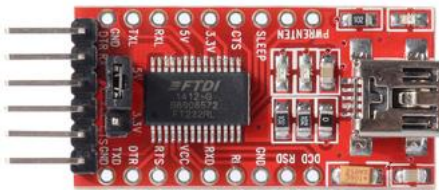
όπου  $x$  είναι ένα σημείο στη συστάδα  $C_i$  και  $m_i$  το centroid της συστάδας

#### Αλγόριθμος

- 1: Επιλογή  $k$  σημείων ως τα αρχικά centroids
- 2: Επανάλαβε
- 3:       Ανάθεσε κάθε σημείο στη συστάδα με το κοντινότερο centroid
- 4:       Επανα - υπολόγισε νέα centroids για κάθε cluster
- 5: Μέχρι να ικανοποιηθεί το κριτήριο σύγκλισης

## 6.4 Αλγόριθμος μηχανικής όρασης

Η ανάπτυξη της εφαρμογής για την εξαγωγή σε πραγματικό χρόνο των απαραίτητων δεδομένων κίνησης μέσω της κάμερας για την αυτόνομη πλοήγηση της ρομποτικής πλατφόρμας πραγματοποιήθηκε με την βοήθεια του ολοκληρωμένου προγραμματιστικού περιβάλλοντος ανάπτυξης (IDE) Visual Studio 2013 της Microsoft με εγκατεστημένο σε αυτό την πρόσθετη εφαρμογή [Image Watch](#) για την προεπισκόπηση των αποτελεσμάτων επεξεργασίας στις εικόνες κατά την διάρκεια της εκσφαλμάτωσης. Χρησιμοποιήθηκε η αντικειμενοστραφής γλώσσα προγραμματισμού C++ και οι βιβλιοθήκες της OpenCV<sup>11</sup> στην έκδοση 2.4.9. Στα πρώτα στάδια συγγραφής του πηγαίου κώδικα είσοδο στον αλγόριθμο αποτέλεσε μαγνητοσκοπημένο video της διαδρομής που επρόκειτο να διασχίσει το όχημα. Τα δεδομένα αποστέλλονταν σειριακά από τον Η/Υ στην αναπτυξιακή πλακέτα Arduino Mega 2560 του οχήματος μέσω ενός μετατροπέα USB σε TTL (FTDI). Η μεταγλώττιση του πηγαίου κώδικα και η δημιουργία του εκτελέσιμου αρχείου της εφαρμογής στο Λ.Σ. Ubuntu 14.04 το οποίο είναι εγκαταστημένο στον μικροϋπολογιστή Odroid-XU3 πραγματοποιήθηκε μέσω του Eclipse IDE. Η εκκίνηση του συγκεκριμένου μεταγλωττιστή απαιτεί δικαιώματα διαχειριστή και επιτυγχάνεται μέσα από ένα τερματικό με την εντολή `sudo eclipse`. Οι απαιτούμενες ρυθμίσεις για την μεταγλώττιση των projects του MS Visual Studio 2013 αλλά και του Eclipse IDE που χρησιμοποιούν τις βιβλιοθήκες της OpenCV διατίθενται στο παράρτημα Δ της παρούσας εργασίας.



Εικόνα 6.32 FT232RL FTDI USB to Serial Adapter

### Επεξήγηση αλγορίθμου

Αρχικά λαμβάνεται σε πραγματικό χρόνο στιγμιότυπο (frame) από την web camera του οχήματος σε ανάλυση 640x480 εικονοστοιχείων και μειώνεται (resize) η ανάλυση του κατά 30%.



Εικόνα 6.33 Ληφθέν στιγμιότυπο προς επεξεργασία

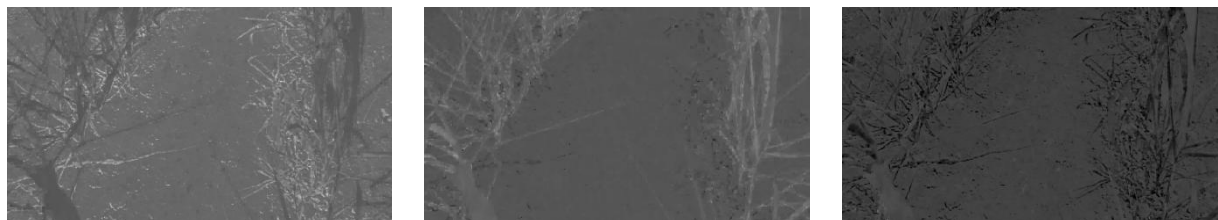
<sup>11</sup> Η **OpenCV** (Open Source Computer Vision) είναι ένα σύνολο βελτιστοποιημένων βιβλιοθηκών ανοικτού κώδικα, γραμμένες σε γλώσσα προγραμματισμού C/C++, οι οποίες οι αποτελούνται από συναρτήσεις, δομές δεδομένων και κλάσεις που επικεντρώνονται σε εφαρμογές συστημάτων τεχνητής όρασης πραγματικού χρόνου και ψηφιακής επεξεργασίας εικόνας. Είναι λογισμικό το οποίο τρέχει σε διαφορετικές πλατφόρμες υλικού ή λειτουργικά συστήματα (Windows, Linux, Mac OS, iOS και Android) αρκεί να μεταγλωττιστεί σε αυτά. Επιτρέπει την ενσωμάτωση και διασύνδεση της με Python, Java και Matlab και επωφελείται υπολογιστικά την ύπαρξη πολλών πυρήνων. Διανέμεται με άδεια χρήσης BSD (Berkeley Software Distribution) συνεπώς προσφέρεται δωρεάν τόσο για ακαδημαϊκή όσο και για επαγγελματική χρήση. Δίνεται η δυνατότητα δηλαδή να ενσωματωθεί σε εμπορικό προϊόν χωρίς να δημιουργηθούν περίπλοκα νομικά ζητήματα.

Διαχωρίζονται τα χρωματικά κανάλια Red, Green και Blue.

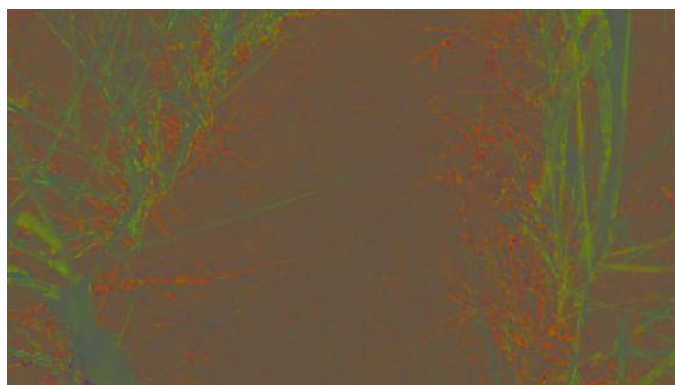


Εικόνα 6.34 Χρωματικά κανάλια Red, Green και Blue αντίστοιχα

και κανονικοποιούνται οι χρωματικές τιμές του στιγμιότυπου

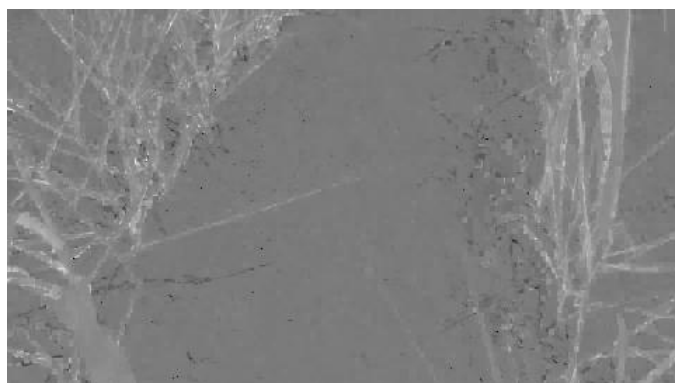


Εικόνα 6.35 Μετατροπή από το μοντέλο απεικόνισης RGB σε rg χρωματικότητα (r,g και b αντίστοιχα)



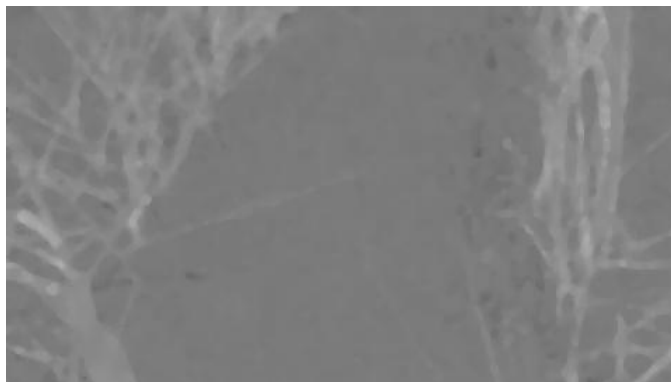
Εικόνα 6.36 Αναπαράσταση της rg χρωματικότητας του τρέχοντος στιγμιότυπου

Εφαρμόζεται ο κατάλληλος δείκτης βλάστησης (CIVE) για την τμηματοποίηση της εικόνας με βάση το χρώμα και κανονικοποιούνται οι φωτεινότητες των εικονοστοιχείων στο εύρος 0 έως 255 (βάθος χρώματος 8bit).



Εικόνα 6.37 Προκύπτουσα εικόνα μετά την εφαρμογή του χρωματικού δείκτη CIVE

Εφαρμόζεται φίλτρο διάμεσης τιμής (median)<sup>12</sup> με μάσκα 7x7 εικονοστοιχείων για την εξομάλυνση της εικόνας και την καταστολή του θορύβου.



Εικόνα 6.38 Εξομαλυσμένη εικόνα με το φίλτρο διάμεσης τιμής

Η εξομαλυσμένη εικόνα κατωφλιώνεται με την μέθοδο του Otsu



Εικόνα 6.39 Προκύπτουσα δυαδική εικόνα μετά την κατωφλίωση

Ακολουθεί η μορφολογική επεξεργασία της δυαδικής εικόνας και συγκεκριμένα η εφαρμογή της πράξης του κλεισίματος (*closing*) ακολουθούμενη από μια λειτουργία ανοίγματος (*opening*) του αποτελέσματος με το ίδιο δομικό στοιχείο (μάσκα 7x7 με ελλειψοειδής μορφή).



Εικόνα 6.40 Μορφολογική επεξεργασία α)λειτουργία closing β) λειτουργία opening

<sup>12</sup> Οι τιμές κάθε γειτονιάς εικονοστοιχείων ταξινομούνται κατά αύξουσα σειρά και επιλέγεται το μεσαίο στοιχείο για την εκπροσώπηση όλης της περιοχής.

Κατόπιν βρίσκονται τα εξωτερικά περιγράμματα των λευκών περιοχών της δυαδικής εικόνας και ταξινομούνται οι περιοχές με κριτήριο το εμβαδόν τους. Οι περιοχές εκείνες με εμβαδόν επιφάνειας μικρότερο από το 10% της μεγαλύτερης απορρίπτονται.



**Εικόνα 6.41** Αποτύπωση στο τρέχων στιγμιότυπο των περιγραμμάτων των περιοχών που τελικά ελήφθησαν υπόψιν  
Βρίσκονται τα γεωμετρικά κέντρα (centroids) κάθε περιοχής.



**Εικόνα 6.42** Απεικόνιση των κέντρων μάζας (πορτοκαλί κουκίδες) των λευκών περιοχών στην αρχική εικόνα

Αν υπάρχουν μόνο δυο περιοχές άσπρων εικονοστοιχείων, η περιοχή της οποίας η τετμημένη<sup>13</sup> του γεωμετρικού κέντρου της είναι μικρότερη βρίσκεται στην αριστερή πλευρά της εικόνας και η δεύτερη περιοχή δεξιά. Αν υπάρχουν περισσότερες από δυο περιοχές για την ομαδοποίηση και των διαχωρισμό των λευκών περιοχών σε αυτές που ανήκουν στο αριστερό τμήμα της εικόνας από εκείνες που υπάρχουν στο δεξιό τμήμα εφαρμόζεται ο αλγόριθμος k-means με δεδομένα εισόδου τα γεωμετρικά κέντρα κάθε περιοχής. Οι περιοχές κατηγοριοποιούνται και ετικετοποιούνται σύμφωνα με την ομάδα στην οποία ανήκουν και ταυτόχρονα υπολογίζεται το μέσο των σημείων κάθε συστάδας.

<sup>13</sup> Θεωρείται καρτεσιανό σύστημα συντεταγμένων με αρχή την πάνω αριστερή γωνία και άξονα τετμημένων τον οριζόντιο άξονα και άξονα τεταγμένων τον κατακόρυφο.



**Εικόνα 6.43** Εύρεση των κεντρικών σημείων (κόκκινες κουκίδες) κάθε ομάδας γεωμετρικών κέντρων με βάση τον αλγόριθμο k-means

Όμοια αν η τετμημένη του μέσου της ομάδας των γεωμετρικών κέντρων των λευκών περιοχών είναι μικρότερη σε σχέση με την τετμημένη του μέσου της δεύτερης ομάδας τότε η ομάδα αυτή κατηγοριοποιείται στο αριστερό τμήμα της εικόνας και οι περιοχές της άλλης ομάδας στο δεξί τμήμα.



**Εικόνα 6.44** Διαχωρισμός και κατηγοριοποίηση των περιοχών των λευκών εικονοστοιχείων.

Το σύνολο των σημείων του εξωτερικού περιγράμματος κάθε περιοχής λευκών εικονοστοιχείων κάθε μιας ομάδας συγχωνεύεται σε πίνακα και διοχετεύεται ως όρισμα σε συνάρτηση (fitLine), η οποία βασίζεται στην μέθοδο των ελαχίστων τετραγώνων, για την εύρεση των βοηθητικών ευθειών αριστερά και δεξιά του φυσικού "διαδρόμου" που σχηματίζουν τα φυτά. Επιστρέφεται ένα διάνυσμα 4 στοιχείων:

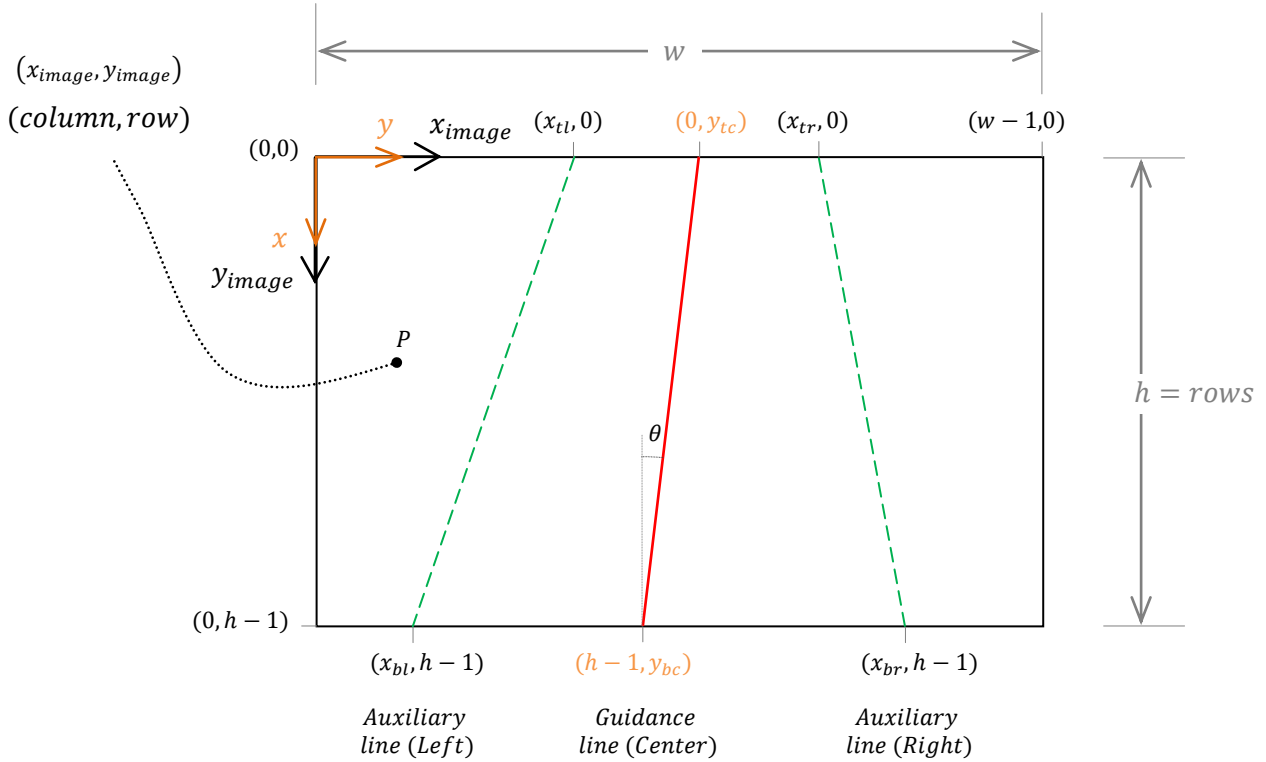
$$(p_0, p_1, p_2, p_3) = (vx, vy, x_0, y_0)$$

όπου  $(vx, vy)$  ένα κανονικοποιημένο διάνυσμα συγγραμμικό με την ζητούμενη ευθεία και  $(x_0, y_0)$  ένα σημείο της ευθείας. Οπότε ο συντελεστής διεύθυνσης  $\lambda$  της ευθείας είναι

$$\lambda = \frac{vy}{vx}$$

και ο σταθερός όρος  $c$

$$y_0 = \frac{vy}{vx} x_0 + c \rightarrow c = y_0 - \frac{vy}{vx} x_0 \rightarrow c = p_3 - \frac{p_1}{p_0} p_2$$



**Εικόνα 6.45.** Καθορισμός των συστημάτων συντεταγμένων στο επίπεδο της εικόνας και χάραξη της κατευθυντήριας και των δυο βοηθητικών ευθειών

Με βάση την εξίσωση κάθε βοηθητικής ευθείας υπολογίζεται η τετμημένη του πάνω και κάτω σημείου σε κάθε πλευρά της εικόνας (μακριά και κοντά αντίστοιχα από την ρομποτική πλατφόρμα). Π.χ. στην αριστερή πλευρά του οχήματος πάνω σημείο θεωρείται το σημείο με τετμημένη  $x_{tl}$  που είναι το σημείο τομής της αριστερής βοηθητικής ευθείας με τον οριζόντιο άξονα και το κάτω σημείο είναι εκείνο με τετμημένη  $x_{bl}$  και τεταγμένη  $y_{bl} = h - 1$  όπου  $h$  το πλήθος των γραμμών της δυαδικής εικόνας.

$$y_{tl} = \lambda x_{tl} + c \xrightarrow{y_{tl}=0} x_{tl} = -\frac{c}{\lambda} \rightarrow x_{tl} = -\frac{p_3 - \frac{p_1}{p_0} p_2}{\frac{p_1}{p_0}} \rightarrow x_{tl} = p_2 - \frac{p_0}{p_1} p_3$$

$$y_{bl} = \lambda x_{bl} + c \rightarrow x_{bl} = \frac{y_{bl} - c}{\lambda} \rightarrow x_{bl} = \frac{p_0}{p_1} \left( y_{bl} - p_3 + \frac{p_1}{p_0} p_2 \right) \rightarrow$$

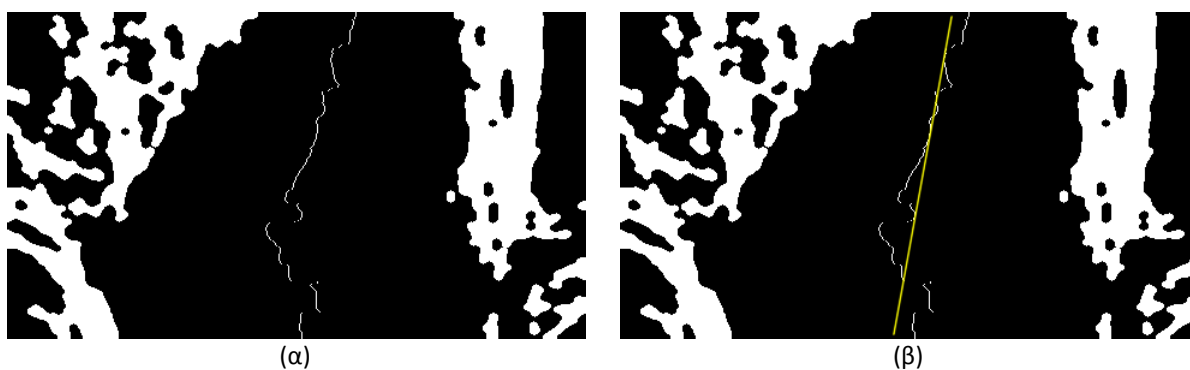
$$x_{bl} = \frac{p_0}{p_1} (h - 1 - p_3) + p_2$$

Αν η τετμημένη του πάνω σημείου (ή του κάτω) προκύψει εκτός των ορίων της εικόνας (πρέπει  $0 \leq x < w$ ) βρίσκονται δυο σημεία, ένα με την μικρότερη και ένα με την μεγαλύτερη τετμημένη. Από αυτά το σημείο εκείνο με την μικρότερη τεταγμένη θεωρείται ως άνω και το σημείο με την μεγαλύτερη ως κάτω.



**Εικόνα 6.46** Εμφάνιση των βοηθητικών ευθειών στην αριστερή και δεξιά πλευρά της εικόνας και των σημείων τομής των ευθειών με τα όρια της εικόνας (άνω άσπρα και κάτω μαύρα τετραγώνια)

Για την εύρεση της κατευθυντήριας γραμμής του σχήματος η οποία προσεγγίζει το μέσο του διαδρόμου σαρώνεται ανά γραμμή η δυαδική εικόνα από κάτω (τελευταία γραμμή του πίνακα) προς τα επάνω και από τη μέση της εικόνας προς την αριστερή και δεξιά πλευρά. Αφετηρία αποτελεί το εικονοστοιχείο  $([w/2], h - 1)$  της τελευταίας γραμμής και της μεσαίας στήλης και αναζητούνται σ' αυτήν την γραμμή τόσο δεξιά όσο και αριστερά τα κοντινότερα προς την μέση άσπρα εικονοστοιχεία. Ο μέσος όρος των τεταγμένων αυτών των δυο άσπρων εικονοστοιχείων που ευρέθησαν αποτελεί ένα σημείο της μέσης του διαδρόμου. Η σάρωση συνεχίζεται στην αμέσως επόμενη γραμμή με αφετηρία το εικονοστοιχείο με τεταγμένη ίση με εκείνη του εικονοστοιχείου που βρέθηκε ως μέσο στην προηγούμενη γραμμή. Αναζητούνται ομοίως και σ' αυτήν την γραμμή τα κοντινότερα εικονοστοιχεία άσπρου χρώματος και υπολογίζεται ο μέσος όρος των τεταγμένων τους. Το εικονοστοιχείο με τεταγμένη το μέσο όρο των τεταγμένων των άσπρων εικονοστοιχείων που ευρέθησαν στην γραμμή που προσπελαύνεται και τετμημένη τον αριθμό αυτής της τρέχουσας γραμμής αποτελεί το επόμενο σημείο το οποίο ανήκει στην μέση του διαδρόμου. Η διαδικασία συνεχίζεται ως και την πρώτη γραμμή του πίνακα με αποτέλεσμα να προκύπτει ένα σύνολο σημείων τα οποία προσεγγίζουν την μέση του διαδρόμου. Για την εύρεση της ευθείας η οποία παρεμβάλλεται μεταξύ των σημείων αυτών χρησιμοποιείται η μέθοδος των ελαχίστων τετραγώνων. Στην προκειμένη περίπτωση ως καρτεσιανό σύστημα συντεταγμένων θεωρείται εκείνο με αρχή την πάνω αριστερή γωνία και με άξονα τετμημένων τον κατακόρυφο άξονα και άξονα τεταγμένων τον οριζόντιο.



**Εικόνα 6.47** α) το σύνολο των σημείων που προσεγγίζουν τη μέση του διαδρόμου των φυτών και β) η παρεμβαλλόμενη ευθεία

Για την τεταγμένη του πάνω άκρου της ευθείας το οποίο τέμνει τον οριζόντιο άξονα θα είναι

$$y_{tc} = \lambda x_{tc} + c \xrightarrow{x_{tc}=0} y_{tc} = c$$

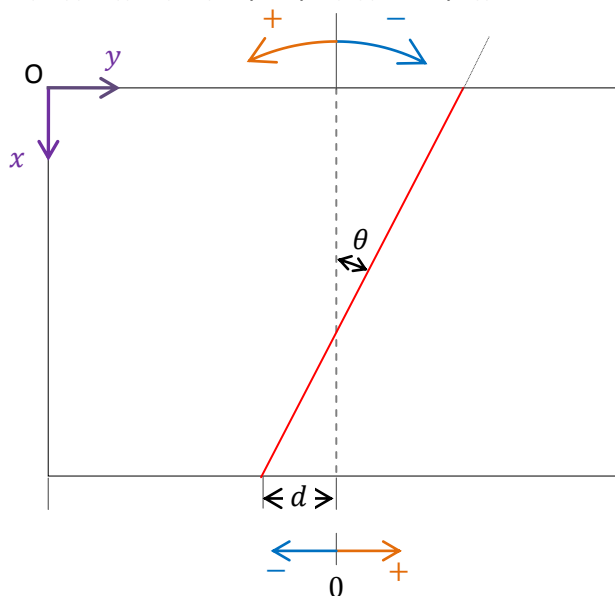


και για την τεταγμένη του κάτω άκρου ισχύει

$$y_{bc} = \lambda x_{bc} + c \xrightarrow{x_{bc}=h-1} y_{bc} = \frac{vy}{vx}(h-1) + c \rightarrow y_{bc} = \frac{p_1}{p_0}(h-1) + c$$

Αν η τεταγμένη του πάνω άκρου της ευθείας (ή του κάτω) βρεθεί μικρότερη του 0 ή μεγαλύτερη ή ίση του πλήθους των στηλών ως πάνω άκρο θεωρείται εκείνο με την μικρότερη τετμημένη και ως κάτω το εικονοστοιχείο με την μεγαλύτερη.

Σε κάθε στιγμιότυπο υπολογίζεται η κλίση  $\theta$  της κατευθυντήριας ευθείας σε μοίρες και η απόσταση  $d$  του κάτω άκρου της ευθείας αυτής από το μέσο της εικόνας εκφρασμένη σε ποσοστό επί τοις εκατό. Η γωνία  $\theta$  και η απόσταση  $d$  είναι προσημασμένα μεγέθη όπως αυτά εικονίζονται στο παρακάτω σχήμα και αποτελούν τις εισόδους σε ελεγκτή ασαφούς λογικής για τον υπολογισμό της διαφοράς η οποία προσθαφαιρείται στο εύρος του duty cycle για την ρύθμιση της ταχύτητας περιστροφής των τροχών κάθε πλευράς του οχήματος.



Εικόνα 6.48 Καθορισμός των δεδομένων εισόδου του ασαφή ελεγκτή αυτόνομης πλοήγησης

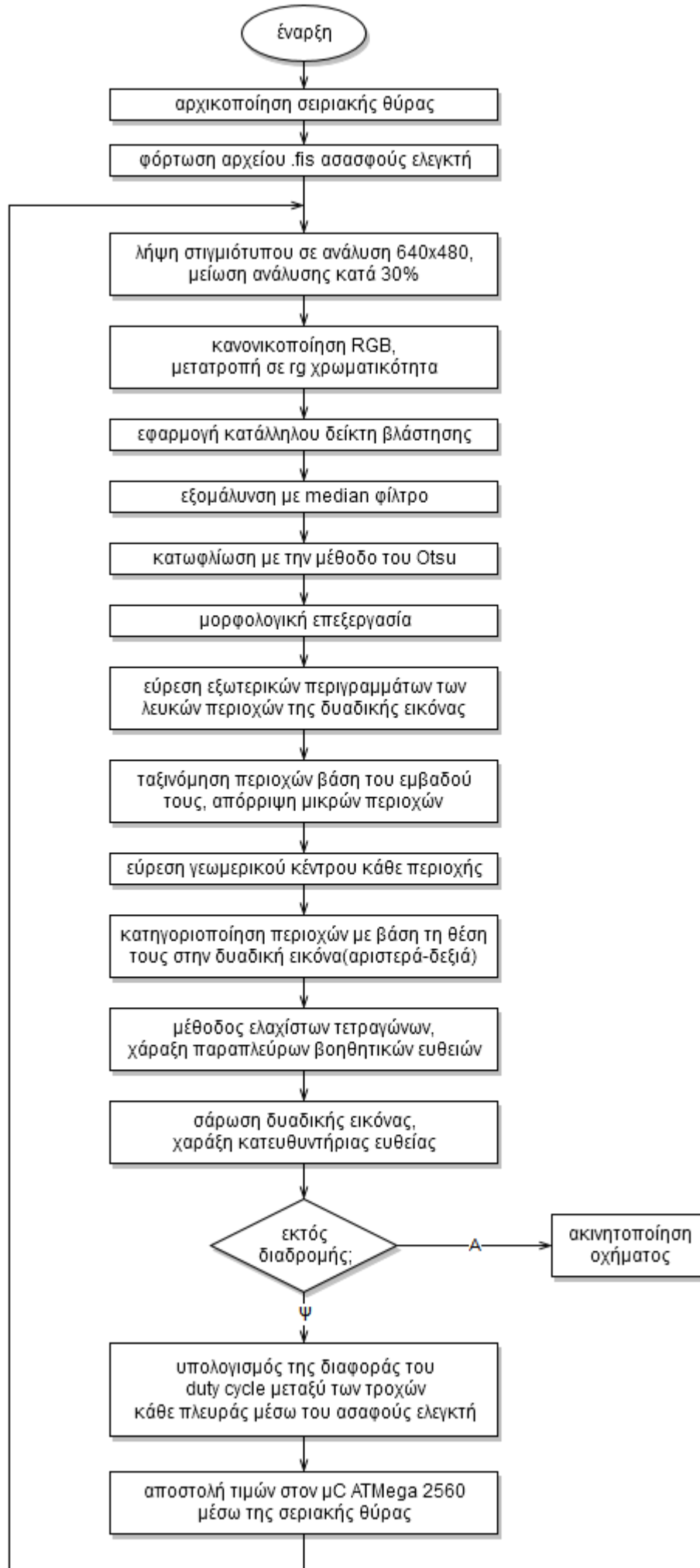


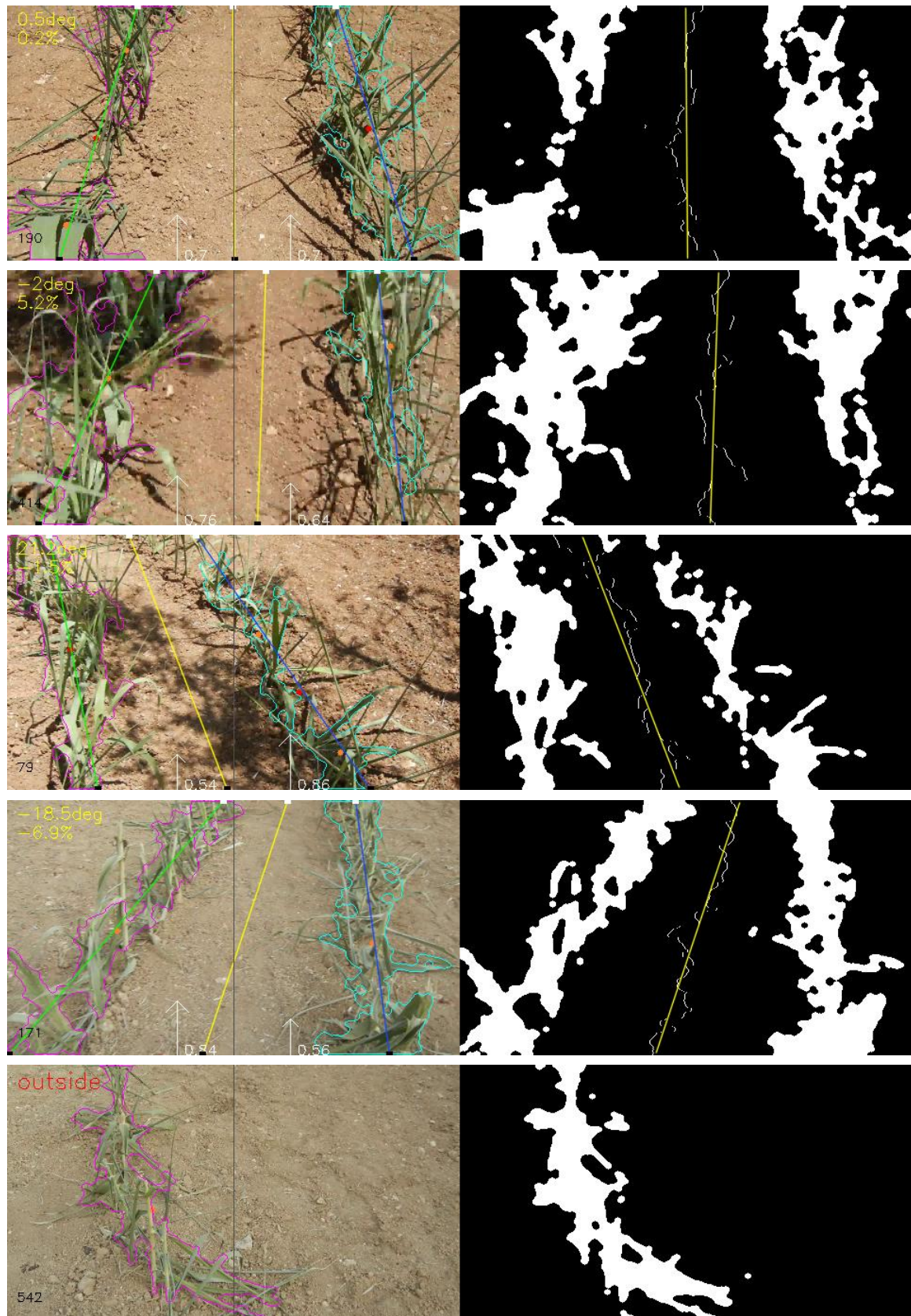
Εικόνα 6.49 Αποτύπωση στο ληφθέν στιγμιότυπο της κατευθυντήριας ευθείας στο μέσο του μονοπατιού και των πληροφοριών που σχετίζονται με αυτήν: κλίση σε μοίρες και απόσταση του κάτω άκρου της από την μέση %

Για την παραμονή της ρομποτικής πλατφόρμας μέσα στο σχηματιζόμενο μονοπάτι των φυτών θα πρέπει να ισχύει κάθε χρονική στιγμή

$$x_{tl} \leq y_{tc} \leq x_{tr} \text{ και } x_{bl} \leq y_{bc} \leq x_{br}$$

Η κωδικοποίηση του συγκεκριμένου αλγόριθμου διατίθεται στο Παράρτημα Ε. Ακολουθεί το λογικό διάγραμμα του.





Εικόνα 6.50 Εικόνες από μαγνητοσκοπημένο βίντεο από την εκτέλεση του αλγόριθμου

## 6.5 Αυτόματη εκτέλεση εφαρμογών κατά την έναρξη του Λ.Σ. Ubuntu

Μέσα από το γραφικό περιβάλλον του Λ.Σ. Ubuntu για να ανοιχτεί ένα terminal πληκτρολογείται `Ctrl+Alt+T`. Πληκτρολογώντας `pwd` και πατώντας `enter` θα εμφανιστεί ο τρέχων κατάλογος `/home/odroid`. Κατόπιν στην γραμμή εντολών δίνονται οι εντολές ώστε να δημιουργηθεί ή να προσπελαστεί αρχείο με επέκταση `.desktop`

```
cd .config/autostart    μετακίνηση στον κατάλογο autostart
sudo nano               εκτέλεση εφαρμογής κειμενογράφου nano,
                        ή
                        η εντολή sudo είναι ένας τρόπος προσωρινής
sudo nano filename     απόκτησης δικαιωμάτων διαχειριστή
```

Μια λίστα όλων των αρχείων και των καταλόγων που υπάρχουν μέσα στον τρέχοντα κατάλογο δίνεται με την εντολή `ls -l`. Το αρχείο με επέκταση `.desktop` ακολουθεί την κάτωθι δομή

```
[Desktop Entry]
Type = Application
Name = όνομα εφαρμογής
Exec = εντολή εκκίνησης της εφαρμογής από το κέλυφος (shell)
Terminal = αληθής ή ψευδής, καθορίζεται αν η εφαρμογή απαιτεί την εκτέλεση
           της σε τερματικό
```

Παράδειγμα αποτελεί το εκτελέσιμο αρχείο της εφαρμογής της μηχανικής όρασης για την αυτόνομη πλοήγηση της ρομποτικής πλατφόρμας. Κατά την εκτέλεση του εκτελέσιμου αρχείου `opencv_prj` προστίθενται στην μεταβλητή κελύφους `LD_LIBRARY_PATH` η θέση των αρχείων βιβλιοθήκης του ασαφούς ελεγκτή. (Με την αυτόματη εκτέλεση της η συγκεκριμένη εφαρμογή κατά την εκκίνηση του Λ.Σ. δεσμεύει την web camera συνεπώς αν χρειαστεί να τροποποιηθεί ο πηγαίος κώδικας της, να αποσφαλμάτωση και να μεταγλωττιστεί εκ νέου μέσω του Eclipse θα πρέπει στο αρχείο `opencv_prj.desktop` το πεδίο `Exec` να είναι κενό ή να αλλαχθεί το μονοπάτι της εφαρμογής σε εσφαλμένη θέση ώστε κατά την επανεκκίνηση του το Odroid XU3 να μην εκτελεί αυτοματοποιημένα την εφαρμογή `opencv_prj`).

Αρχείο: `opencv_prj.desktop`

```
[Desktop Entry]
Type = Application
Name = opencv_prj
Exec = sudo LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
/home/odroid/MyProjects/fuzzylite/fuzzylite.lib/debug/bin
/home/odroid/MyProjects/opencv_prj/Debug/opencv_prj
Terminal = true
```

Με την βοήθεια της εντολής `pgrep process_name` επιστρέφεται το ID της διεργασίας αν αυτή εκτελείται και με την εντολή `sudo kill process_ID` τερματίζεται η εκτέλεση της.

Για τον ομαλό τερματισμό του Λ.Σ. Ubuntu και κλείσιμο (shutdown) του Odroid XU3 εκκινείται αυτοματοποιημένα και η εφαρμογή ανίχνευσης πατήματος του πλήκτρου 0 του πληκτρολόγιου μεμβράνης

Αρχείο: shutdown.desktop

[Desktop Entry]

Type = Application

Name = gpio

Exec = sudo /home/odroid/MyProjects/gpio/Debug/gpio

Terminal = true

Κατά τη χρήση της sudo στο τερματικό ζητείται να πληκτρολογηθεί ο κωδικός πρόσβασης. Για την αποφυγή επανειλημμένης εισόδου του κωδικού και για την αυτοματοποίηση της διαδικασίας τροποποιήθηκε το αρχείο `sudoers`. Το συγκεκριμένο αρχείο περιλαμβάνει τους κανόνες που πρέπει να ακολουθούν οι χρήστες όταν χρησιμοποιούν την εντολή `sudo`.

Στη γραμμή εντολών δόθηκαν οι κάτωθι εντολές:

```
su root
```

```
export VISUAL=nano; visudo
```

σύνδεση ως διαχειριστής

επιλογή κειμενογράφου nano;

άνοιγμα αρχείου `/etc/sudoers`

και στο τέλος του αρχείου `sudoers` πληκτρολογήθηκε `odroid ALL=(ALL) NOPASSWD:ALL`  
Με `Ctrl + X` κλείνει ο κειμενογράφος και οι τρέχουσες αλλαγές αποθηκεύονται πληκτρολογώντας `Y` και `enter`.

## Κεφάλαιο

# 7

## 7. Ασαφής ελεγκτής αυτόνομης πλοήγησης

Στο παρόν κεφάλαιο αναλύονται οι λόγοι και η αναγκαιότητα εφαρμογής της θεωρίας των ασαφών συνόλων και της ασαφούς λογικής στον έλεγχο συστημάτων. Περιγράφονται τα δομικά μέρη ενός ασαφούς ελεγκτή και αναφέρονται τα επιμέρους βήματα υλοποίησης του. Για την αυτόνομη πλοήγηση της ρομποτικής πλατφόρμας σε καλλιέργεια σειρών σχεδιάζεται ασαφής ελεγκτής τύπου Mamdani με δυο εισόδους και μια έξοδο. Τα δεδομένα εισόδου προκύπτουν από τον αλγόριθμο επεξεργασίας εικόνας και η τιμή που παράγεται ως έξοδος χρησιμοποιείται για την ρύθμιση του duty cycle κάθε ζεύγους κινητήρων του οχήματος.

### 7.1 Ασαφής έλεγχος - εισαγωγικά στοιχεία

Η εφαρμογή τεχνικών της συμβατικής θεωρίας ελέγχου προϋποθέτει την ύπαρξη ενός πλήρους αναλυτικού μοντέλου του ελεγχόμενου συστήματος. Κάτι τέτοιο είναι συχνά αδύνατο λόγω της πολυπλοκότητας των διεργασιών ή και της αδυναμίας μετρήσεων, ή όταν δεν έχουμε ένα σταθερά διαμορφωμένο περιβάλλον ή έστω ένα περιβάλλον που μεταβάλλεται με προβλέψιμο τρόπο.

Ο Lofti A. Zadeh που θεωρείται ο 'πατέρας' της ασαφούς λογικής δήλωνε ότι η προσπάθεια μας να κατασκευάσουμε μηχανές με νοημοσύνη δεν μπορεί να προχωρήσει αν δεν βρεθεί ένας τρόπος ώστε οι μηχανές να σκέφτονται με παρόμοιους μηχανισμούς όπως ένας άνθρωπος.

Η θεωρία του μη συμβατικού ελέγχου αντί να προσπαθεί να μοντελοποιήσει το ελεγχόμενο σύστημα ψάχνει να βρει ένα σύνολο λεκτικών προτάσεων που περιγράφουν τις αντιδράσεις ενός ανθρώπου χειριστή του συστήματος. Ο στόχος ενός ευφυούς ελεγκτή είναι να λειτουργεί με τους ίδιους κανόνες όπως ένας άνθρωπος ελεγκτής μιας διεργασίας. Το πλεονέκτημα των ανθρώπων σαν ελεγκτές μιας διεργασίας είναι ότι μπορούν να αντιδράσουν άμεσα σε απρόβλεπτες καταστάσεις. Ένας καλά σχεδιασμένος ευφυής ελεγκτής πρέπει να μπορεί να μιμηθεί τον καλύτερο άνθρωπο χειριστή της συγκεκριμένης διεργασίας. Έτσι ένα πρώτο πρόβλημα που πρέπει να απαντηθεί από τον σχεδιαστή ενός ευφυούς ελεγκτή είναι η καταγραφή των κανόνων με βάση τους οποίους λειτουργεί ο άνθρωπος χειριστής της διεργασίας ώστε το σύστημα να είναι εύρωστο να έχει δηλαδή την ικανότητα να παραμένει λειτουργικό κάτω από μη αναμενόμενες συνθήκες.

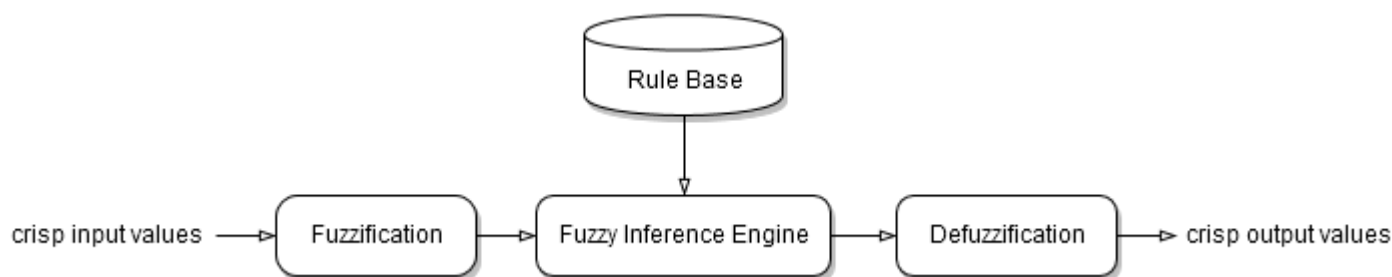
Μέσω της αναπαράστασης των λεκτικών όρων από τα ασαφή σύνολα η ασαφής λογική αποτελεί την γέφυρα επικοινωνίας ανάμεσα στον άνθρωπο και την μηχανή. Η ασαφής λογική δημιουργήθηκε ώστε οι υπολογιστικές μηχανές να μπορούν να χειριστούν λεκτικούς όρους. Ο έλεγχος συστημάτων των οποίων η σχέση διέγερσης-απόκρισης χαρακτηρίζεται από ισχυρές μη γραμμικότητες είναι εφικτός στο πλαίσιο της ασαφούς λογικής.

Ο Zadeh πρότεινε ένα διευρυμένο τρόπο αναπαράστασης όπου μια τιμή ανήκει ταυτόχρονα σε πολλά υποσύνολα, στο κάθε ένα με ένα βαθμό συμμετοχής. Τα ασαφή σύνολα (fuzzy sets) χρησιμοποιούνται για να αναπαραστήσουν της μεταβλητές εισόδου και εξόδου με λεκτικούς όρους. Η περιγραφή μιας αυστηρά αριθμητικής τιμής με λεκτικούς όρους ονομάζεται ασαφοποίηση (fuzzification). Ο βαθμός συμμετοχής μιας τιμής σ' ένα ασαφές σύνολο αποτελεί το βαθμό βεβαιότητας (degree of certainty) ότι η πρόταση είναι αληθής. Το πλάτος ενός ασαφούς συνόλου είναι το εύρος του πεδίου ορισμού του και αποτελεί ένα μέτρο ασάφειας (fuzziness) του όρου που περιγράφει.

Ο ασαφής κανόνας είναι ένας μηχανισμός αναπαράστασης της γνώσης ο οποίος ταιριάζει στον ανθρώπινο τρόπο σκέψης. Αποτελείται από δυο μέρη α) το τμήμα υπόθεσης και β) το τμήμα απόφασης π.χ. *if x is A then y is B* με A και B ασαφή σύνολα και x μια τιμή μιας μεταβλητής εισόδου η οποία ασαφοποιείται αποκτάει δηλαδή ένα βαθμό συμμετοχής στο ασαφές σύνολο A, η έξοδος y έξοδος του συστήματος εκφράζει την απόφαση του κανόνα και παρέχεται από τον μηχανισμό του συμπεράσματος (inference) σε ασαφή μορφή. Το ασαφές συμπέρασμα απο-ασαφοποιείται και προκύπτει μια crisp τιμή που μπορεί να χειριστεί η υπολογιστική μηχανή. Κανόνας του οποίου η έξοδος είναι ασαφές σύνολο ονομάζεται κανόνας τύπου *mamdani*.

Τα βασικά δομικά στοιχεία ενός ασαφούς ελεγκτή (fuzzy controller) είναι:

- Ο ασαφοποιητής (fuzzifier) ο οποίος μετατρέπει τις πραγματικές τιμές των εισόδων σε ασαφή σύνολα.
- Η βάση κανόνων (rule base) στην οποία είναι αποθηκευμένοι οι κανόνες (if-then rules) για τον έλεγχο του συστήματος.
- Ο μηχανισμός συμπερασμού (inference engine) ο οποίος με τη χρήση της βάσης γνώσης εξάγει τα ασαφή σύνολα των συμπερασμάτων.
- Ο αποασαφοποιητής (defuzzifier) ο οποίος μετατρέπει τα συμπεράσματα που εξάγει ο μηχανισμός συμπερασμού σε πραγματικούς αριθμούς.



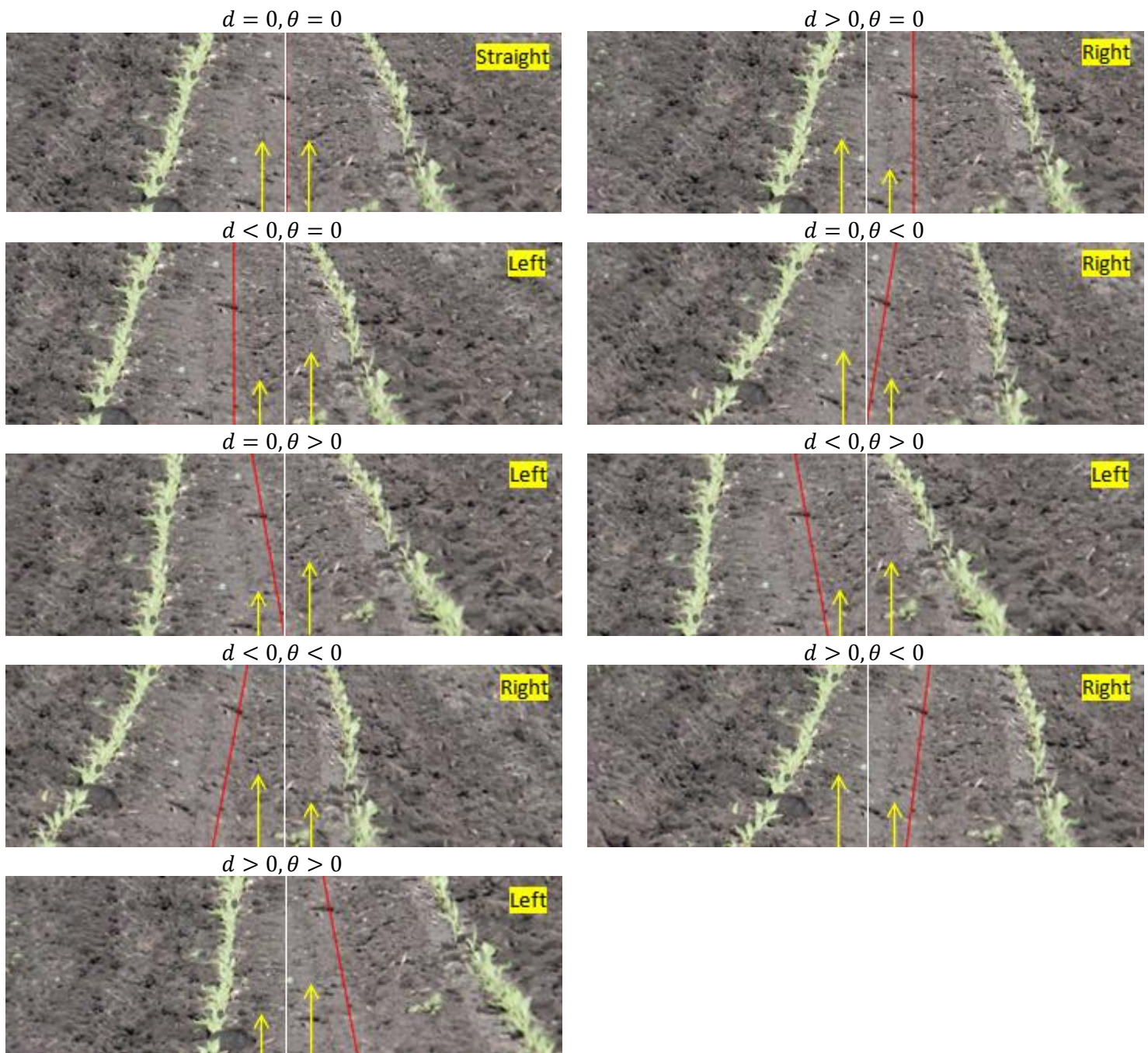
Εικόνα 7.1 Τα δομικά στοιχεία ασαφούς ελεγκτή

## 7.2 Σχεδίαση ασαφούς ελεγκτή

Για την σχεδίαση του ενός ασαφούς ελεγκτή θα πρέπει να ακολουθηθούν τα κάτωθι βήματα:

1. Λεκτικός διαμερισμός (κατανομή) των εισόδων, εξόδων
2. Διατύπωση των κανόνων
3. Καθορισμός του τύπου της ασαφούς συνεπαγωγής
4. Καθορισμός του τύπου της συνάθροισης των κανόνων (fuzzy aggregation)
5. Επιλογή του τύπου της απο-ασαφοποίησης

Στην προκειμένη περίπτωση είσοδοι του ασαφούς ελεγκτή αποτελούν η γωνία κλίσης  $\theta$  και η απόσταση  $d$  του κατώτατου σημείου της κατευθυντήριας ευθείας σε σχέση με την μέση της εικόνας που καταγράφει σε πραγματικό χρόνο η κάμερα. Η ευθεία αυτή αντιπροσωπεύει την μέση του φυσικού διαδρόμου που σχηματίζουν τα φυτά. Όλες οι πιθανές περιπτώσεις κατά τις οποίες το όχημα μπορεί να αποκλίνει από την επιθυμητή διαδρομή απεικονίζονται στις κάτωθι εικόνες. Η κόκκινη γραμμή αναπαριστά την μέση του διαδρόμου και η άσπρη την κατεύθυνση του οχήματος. Για τον καθορισμό των κανόνων του ελεγκτή άρα και για την πραγματοποίηση των κατάλληλων διορθωτικών ελιγμών θα πρέπει να δίνονται κατάλληλες εντολές κίνησης ούτως ώστε η άσπρη ευθεία να προσεγγίζει και να ταυτίζεται με την κατευθυντήρια.

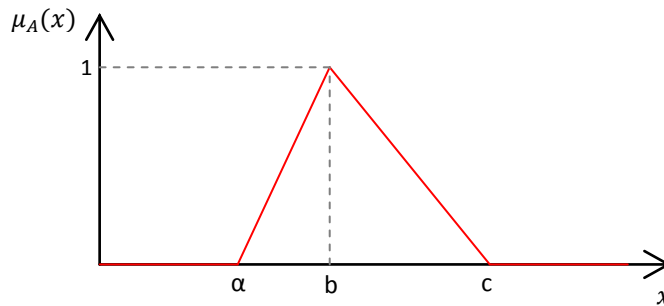


Εικόνα 7.2 Πιθανές περιπτώσεις κατά τις οποίες το όχημα αποκλίνει από την επιθυμητή διαδρομή



Και οι 2 μεταβλητές εισόδου διαμερίζονται με 7 λεκτικούς όρους: *Negative Big*, *Negative Medium*, *Negative Small*, *Zero*, *Positive Small*, *Positive Medium*, *Positive Big*. Η έξοδος  $u$  διαμερίζεται και αυτή σε 7: *Left Big*, *Left Medium*, *Left Small*, *Zero*, *Right Small*, *Right Medium*, *Right Big*. Τα ασαφή σύνολα που αντιπροσωπεύουν αναπαριστούνται για τις εισόδους και την έξοδο με τριγωνικές συναρτήσεις συμμετοχής. Η τριγωνική συνάρτηση συμμετοχής χαρακτηρίζεται από τρεις παραμέτρους  $\{\alpha, b, c\}$

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



Εικόνα 7.3 Τριγωνική συνάρτηση συμμετοχής

Επιλέγοντας τα αντίστοιχα ασαφή σύνολα από κάθε μεταβλητή οι κανόνες είναι της μορφής:

*if (d is Positive Medium) and (theta is Positive Big) then (u is Left Medium)*

Το *and* και το *then* (καθορισμός του τύπου της ασαφούς συνεπαγωγής) υλοποιούνται με τον τελεστή  $\min$ <sup>14</sup> ο οποίος λαμβάνει το μικρότερο από τους βαθμούς συμμετοχής. Παράγεται ο βαθμός εκπλήρωσης (degree of fulfillment) του κανόνα και εκφράζεται με την αντίστοιχη τομή- $\alpha$  ( $\alpha$ -cut)<sup>15</sup> του ασαφούς συνόλου που αντιπροσωπεύει την βαρύτητα που έχει το αποτέλεσμα του κανόνα. Τα αντίστοιχα  $\alpha$ -cuts που δημιουργούνται στις εξόδους κάθε κανόνα συναθροίζονται με την εφαρμογή του τελεστή *max* ώστε να προκύψει ένα ασαφές σύνολο το οποίο αποσαφοποιείται με εφαρμογή της μεθόδου κέντρου βάρους (centroid) ώστε να παραχθεί μια αυστηρά αριθμητική τιμή.

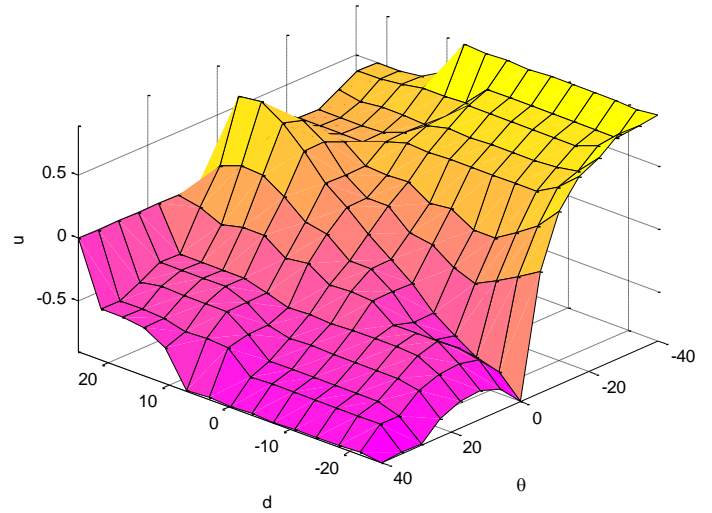
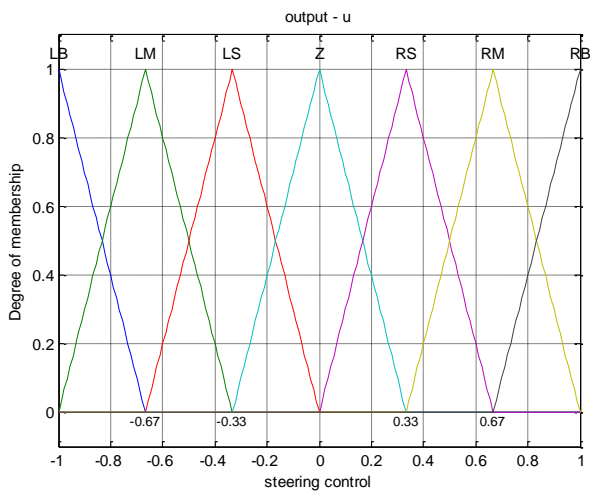
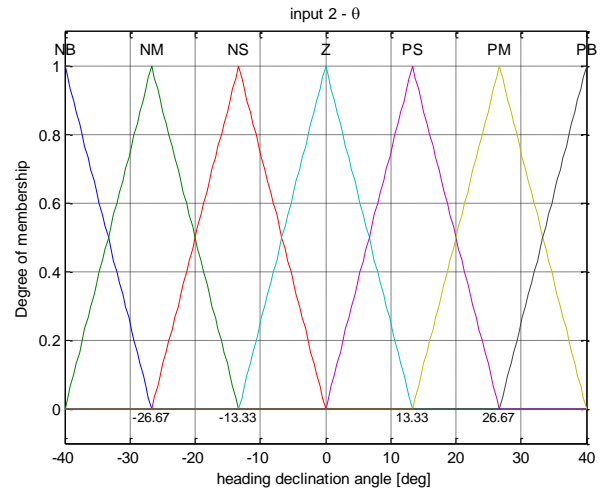
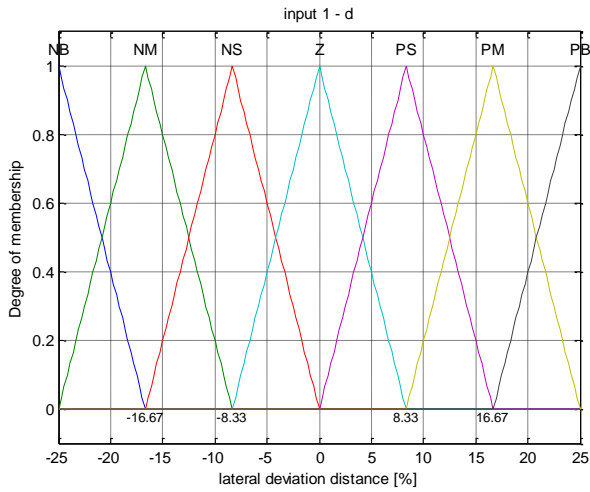
$$x^* = \frac{\sum \mu(x)x}{\sum \mu(x)}$$

<sup>14</sup> Αν θεωρήσουμε δυο ασαφή σύνολα A και B ορισμένα πάνω στο ίδιο κλασικό σύνολο X. Τότε η τομή  $A \cap B$  αυτών είναι ένα ασαφές σύνολο με συνάρτηση συμμετοχής  $\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x)$  όπου  $\wedge$  ο τελεστής ελαχίστου  $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x), \forall x \in X\}$ . Αντίστοιχα η ένωση  $A \cup B$  είναι ένα ασαφές σύνολο του X με συνάρτηση συμμετοχής  $\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x)$  όπου ο τελεστής  $\vee$  μεγίστου  $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x), \forall x \in X\}$ . Η τομή δυο συνόλων αντιστοιχεί στο λεκτικό ΚΑΙ (AND) ενώ η ένωση στο λεκτικό Ή (OR).

<sup>15</sup> Έστω ένα ασαφές σύνολο A με συνάρτηση συμμετοχής  $\mu_A(x)$ . Η τομή- $\alpha$  του ασαφούς συνόλου A είναι ένα νέο ασαφές σύνολο με συνάρτηση συμμετοχής  $\mu_\alpha(x) = \begin{cases} \mu_A(x) & 0 \leq \mu_A(x) \leq \alpha \\ \alpha & \alpha \leq \mu_A(x) \leq 1 \end{cases}$

Όταν η προσημασμένη έξοδος  $u$  του ελεγκτή είναι

- αρνητική ( $u < 0$ ) το όχημα στρίβει αριστερά,
- θετική ( $u > 0$ ) το όχημα στρίβει δεξιά
- μηδέν τότε το όχημα δεν αλλάζει κατεύθυνση και κινείται ευθεία.



**Εικόνα 7.4** Διαμέριση και συναρτήσεις συμμετοχής των δυο εισόδων και της εξόδου του ασαφούς ελεγκτή και 3D γραφική απεικόνιση της εξόδου  $u$  του ελεγκτή συνάρτηση των εισόδων του ( $d, \theta$ )

Στον πίνακα που ακολουθεί αναγράφονται οι 49 κανόνες από τους οποίους αποτελείται η βάση κανόνων του ασαφούς ελεγκτή για την αυτόνομη πλοήγηση του ρομποτικού οχήματος.

		LB	LM	LS	Z	RS	RM	RB
$u$								
		<i>heading declination angle (<math>\theta</math>)</i>						
		NB	NM	NS	Z	PS	PM	PB
<i>lateral deviation distance (<math>d</math>)</i>	NB	RB	RB	RM	LB	LM	LB	LB
	NM	RB	RM	RS	LM	LS	LM	LB
	NS	RB	RM	RS	LS	LS	LM	LB
	Z	RB	RM	RS	Z	LS	LM	LB
	PS	RB	RS	RS	RS	LS	LS	LB
	PM	RM	RS	RS	RM	LS	LS	LM
	PB	RS	RS	RS	RB	LS	LS	LS

Π.χ. για τιμές εισόδου  $d=1\%$  και  $\theta=7.5^\circ$  ενεργοποιούνται οι κάτωθι κανόνες:

- if ( $d$  is Z) and ( $\theta$  is Z) then ( $u$  is Z)*
- if ( $d$  is Z) and ( $\theta$  is PS) then ( $u$  is LS)*
- if ( $d$  is PS) and ( $\theta$  is Z) then ( $u$  is RS)*
- if ( $d$  is PS) and ( $\theta$  is PS) then ( $u$  is LS)*

Στα διαγράμματα της εικόνας 7.5 παρουσιάζεται πως αποασαφοποιούνται οι τιμές, ενεργοποιείται κάθε κανόνας και δημιουργούνται τα αντίστοιχα  $\alpha$ -cuts στις εξόδους κάθε κανόνα. Τα τέσσερα ασαφή συμπεράσματα συναθροίζονται σε ένα ασαφές συμπέρασμα με την εφαρμογή του τελεστή *max* και στη συνέχεια το ασαφές αυτό σύνολο αποασαφοποιείται με την μέθοδο centroid και προκύπτει η τελική πραγματική τιμή που είναι το αποτέλεσμα των τεσσάρων κανόνων.

Συγκεκριμένα αν  $A_i, B_i$  είναι ασαφή σύνολα τα οποία αντιστοιχούν στις εισόδους (απόσταση και γωνία αντίστοιχα) και  $C_i$  το ασαφές σύνολο που αντιστοιχεί στην έξοδο του  $i$  κανόνα και  $C'_i$   $\alpha$ -τομή του συνόλου  $C_i$ , ο βαθμός εκπλήρωσης των κανόνων είναι

$$a_1 = \min \{ \mu_{A_1}(1), \mu_{B_1}(7.5) \} = 0.44 \quad a_2 = \min \{ \mu_{A_2}(1), \mu_{B_2}(7.5) \} = 0.56$$

$$a_3 = \min \{ \mu_{A_3}(1), \mu_{B_3}(7.5) \} = 0.12 \quad a_4 = \min \{ \mu_{A_4}(1), \mu_{B_4}(7.5) \} = 0.12$$

επιλέχθηκε ο τελεστής *min* λόγω του λογικού τελεστή ΚΑΙ μεταξύ των κανόνων. Η έξοδος του κάθε κανόνα παράγεται ως εξής (τελεστής συνεπαγωγής)

$$C'_1 = \min \{ a_1, \mu_{C_1}(u) \} \quad C'_2 = \min \{ a_2, \mu_{C_2}(u) \}$$

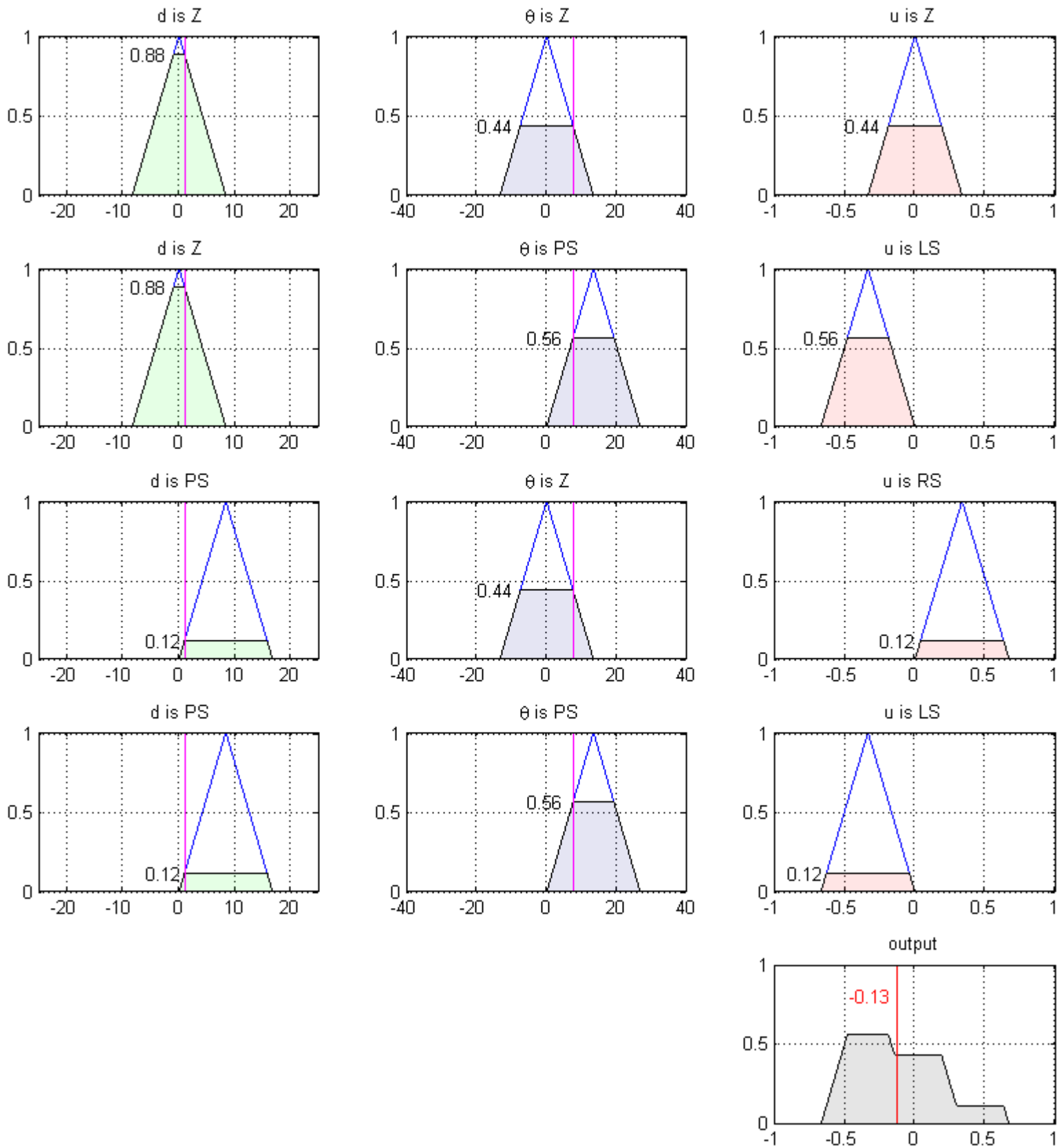
$$C'_3 = \min \{ a_3, \mu_{C_3}(u) \} \quad C'_4 = \min \{ a_4, \mu_{C_4}(u) \}$$

Το αποτέλεσμα των τεσσάρων κανόνων θα είναι

$$C = C'_1 \cup C'_2 \cup C'_3 \cup C'_4$$

και υπολογίζεται χρησιμοποιώντας τον τελεστή *max*

$$\mu_C(u) = \max \{ \mu_{C'_1}(u), \mu_{C'_2}(u), \mu_{C'_3}(u), \mu_{C'_4}(u) \}$$



Εικόνα 7.5 Ενεργοποίηση των αντίστοιχων κανόνων για τιμές εισόδου  $d=1\%$  και  $\theta=7.5^\circ$ , συνάθροιση των συμπερασμάτων τους και αποασαφοποίηση

Για την παραγωγή των σημάτων αναφοράς και την ρύθμιση του duty cycle του ζεύγους των ηλεκτροκινητήρων κάθε πλευράς η έξοδος  $u$  του ασαφούς ελεγκτή με εύρος  $[-1,1]$  διαβαθμίζεται στο  $[0,1]$

$$\omega_l = 0.7 + 0.3 * u$$

$$\omega_r = 0.7 - 0.3 * u$$

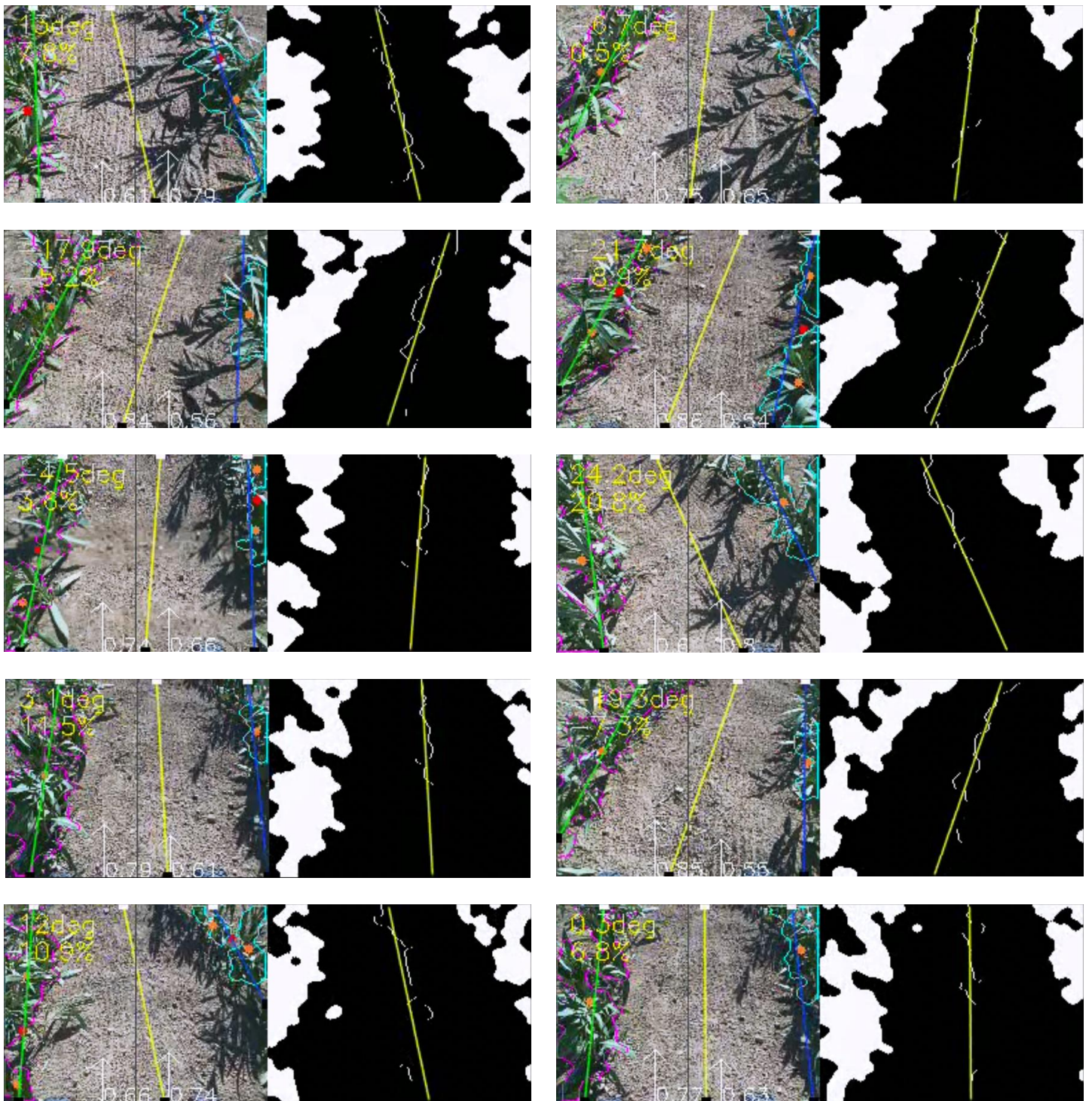
όπου  $\omega_l$  και  $\omega_r$  τα σήματα αναφοράς για το αριστερό και δεξί ζεύγος κινητήρων αντίστοιχα. Οι τιμές των  $\omega_l$  και  $\omega_r$  αποστέλλονται σειριακά από τον μικροϋπολογιστή Odroid-XU3 στην αναπτυξιακή πλακέτα Arduino Mega 2560 και εκεί πολλαπλασιάζονται με την μέγιστη επιθυμητή γωνιακή ταχύτητα των κινητήρων (σε rad/s). Η μέγιστη αυτή ταχύτητα καθορίζεται όταν το όχημα είναι σε στάση μέσω του ηλεκτρολογίου μεμβράνης.

Ο ασαφής ελεγκτής υλοποιήθηκε αρχικά με την βοήθεια του GUI του FIS editor του Matlab και κατόπιν μέσω ανάλογου script. Το αρχείο \*.fis που παράχθηκε ενσωματώνεται στον κώδικα της εφαρμογής της μηχανικής όρασης μέσω της open-source βιβλιοθήκης fuzzylite v.5 (<http://www.fuzzylite.com/>). Τα απαραίτητα αρχεία dll της συγκεκριμένης βιβλιοθήκης δημιουργήθηκαν μεταγλωττίζοντας τον πηγαίο τους κώδικα στο Λ.Σ. Ubuntu 14.04. Αναλυτικές οδηγίες εγκατάστασης και χρήσης της βιβλιοθήκης μέσω της γλώσσας προγραμματισμού C++ υπάρχουν διαθέσιμες στον δικτυακό τόπο της βιβλιοθήκης.

Για την αποτίμηση του ασαφούς ελεγκτή και κατ' επέκταση της δυνατότητας του οχήματος να ακολουθεί την σχηματιζόμενη διαδρομή από τις συστοιχίες των φυτών διαμορφώθηκε σε αγρό κατάλληλο μονοπάτι (εικόνα 7.6) μήκους περίπου 12 μέτρων στα πρότυπα του διαγωνισμού "Robot Field Event". Στην αφετηρία και για την ομαλή εκκίνηση του οχήματος η ταχύτητα των κινητήρων κάθε πλευράς αυξάνει γραμμικά έως ότου ανέλθει στην διαμορφωμένη τιμή εξόδου του ασαφούς ελεγκτή. Το όχημα ακινητοποιείται αυτοματοποιημένα όταν τεθεί εκτός πορείας ή στο τέλος του μονοπατιού. Η γωνία κλίσης της κάμερας ρυθμίστηκε στις  $45^\circ$ .



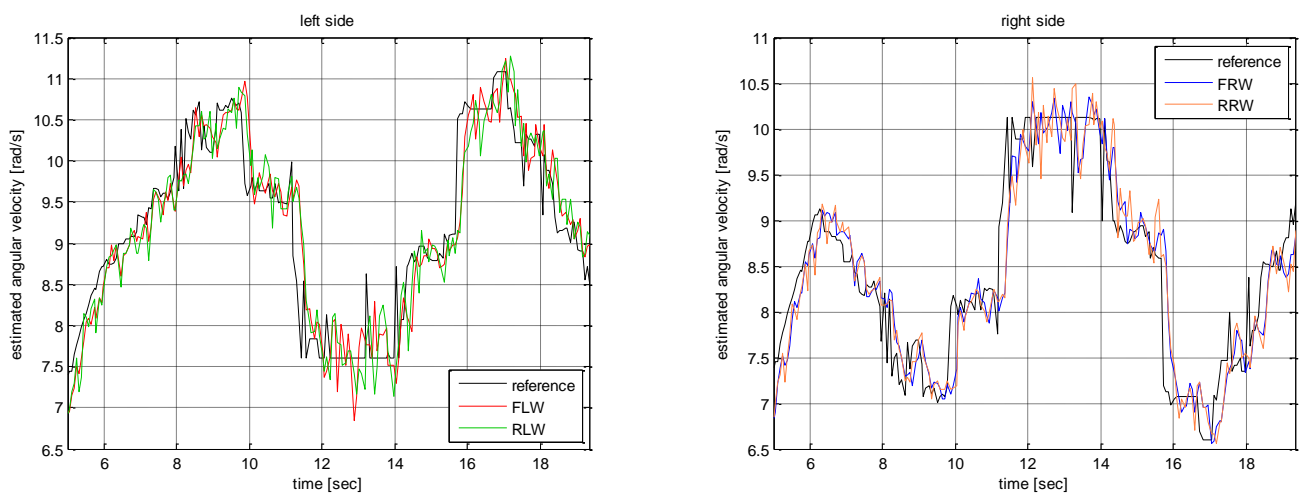
Εικόνα 7.6 Χώρος διεξαγωγής πειραμάτων



Εικόνα 7.7 Εικόνες από καταγεγραμμένο βίντεο μέσω της κάμερας του οχήματος κατά την διάρκεια διάσχισης της διαδρομής της εικόνας 7.6



Εικόνα 7.8 Στιγμιότυπα κίνησης από την πορεία του ρομποτικού οχήματος



Εικόνα 7.9 Εκτιμώμενες γωνιακές ταχύτητες των ηλεκτροκινητήρων κάθε πλευράς του οχήματος κατά την διάσχιση της διαδρομής.

## Κεφάλαιο

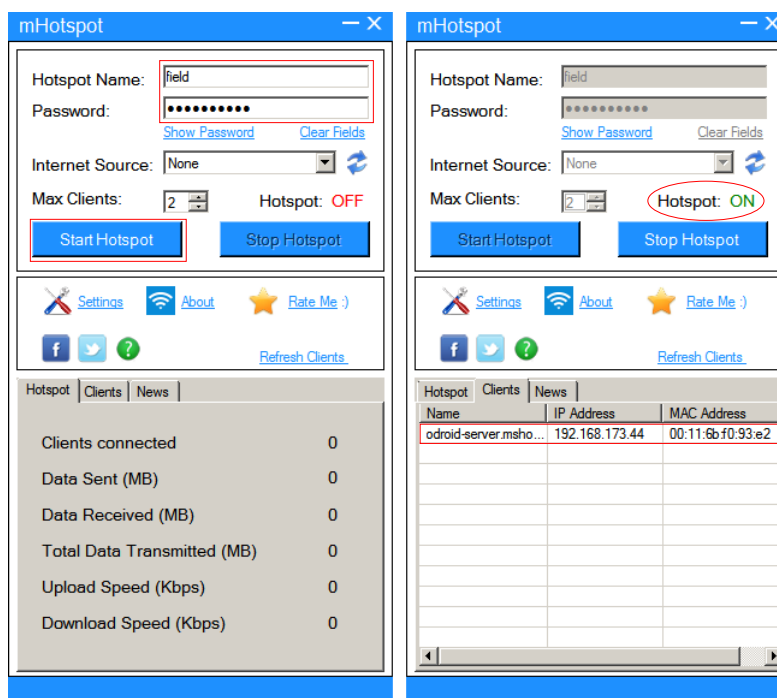
## 8

## 8. Ασύρματη επικοινωνία

Στο κεφάλαιο αυτό δίνονται οι κατάλληλες οδηγίες χρήσης εξειδικευμένου λογισμικού για την επίτευξη ασύρματης σύνδεσης (Wi-Fi) μεταξύ απομακρυσμένου Η/Υ και του μικροϋπολογιστή Odroid-XU3 του ρομποτικού οχήματος ώστε να καταστεί εφικτή η εξ αποστάσεως διαχείριση του. Ακολούθως αναφέρεται ο τρόπος σύνδεσης του χρησιμοποιούμενου Bluetooth module με την αναπτυξιακή πλακέτα Arduino Mega 2560 καθώς και οι εντολές παραμετροποίησης του. Εν συνεχεία περιγράφονται οι απαραίτητες ενέργειες για την αποστολή εντολών κίνησης στο όχημα μέσω της σύνδεσης Bluetooth κάνοντας χρήση είτε κινητού τηλεφώνου και συγκεκριμένης δωρεάν εφαρμογής είτε μέσω της αναπτυχθείσας για αυτό το σκοπό διεπαφής χρήστη στο Matlab.

### 8.1 Διαχείριση του Odroid-XU3 εξ' αποστάσεως μέσω Wi-Fi

Για την επίτευξη ασύρματης ζεύξης χωρίς σύνδεση στο διαδίκτυο και για την απομακρυσμένη διαχείριση του Odroid XU3 από Η/Υ με Λ.Σ. Microsoft Windows (7,8,10) και ασύρματο προσαρμογέα δικτύου χρησιμοποιήθηκε η εφαρμογή mHotspot ([www.mhotspot.com](http://www.mhotspot.com)) με τη βοήθεια της οποίας ο Η/Υ μετατρέπεται σε Wi-Fi Hotspot (virtual router).

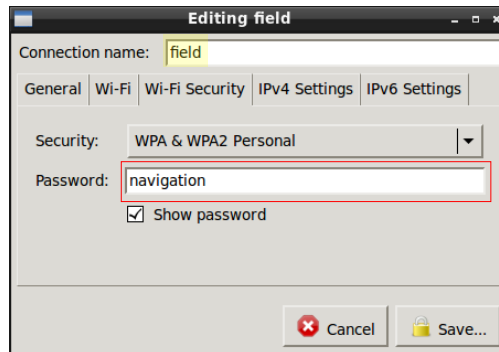


Εικόνα 8.1 Στιγμιότυπα οθόνες της εφαρμογής mHotspot



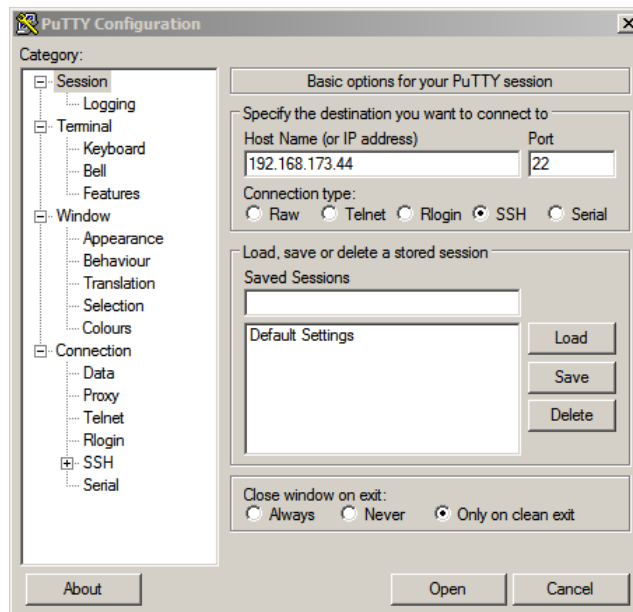
Αρχικά δίνεται ένα όνομα (π.χ. field) της επιλογής μας για το δίκτυο του Wi-Fi Hotspot και ένας κωδικός ασφαλείας τουλάχιστον 8 χαρακτήρων (π.χ. navigation), κατόπιν στο πεδίο Internet Source επιλέγουμε none και καθορίζουμε το μέγιστο επιτρεπόμενο αριθμό συσκευών που μπορούν να συνδεθούν στο δίκτυο κατόπιν πατάμε το πλήκτρο StartHotspot. Με επιτυχή σύνδεση στην καρτέλα Clients εμφανίζεται η αποδοθείσα IP διεύθυνση και η MAC address του Odroid XU3.

Η αυτοματοποιημένη σύνδεση στο δημιουργηθέν ασύρματο δίκτυο επιτυγχάνεται με μια σειρά κατάλληλων ρυθμίσεων (μόνο την πρώτη φορά) μέσα από το παραθυρικό περιβάλλον (GUI) του Λ.Σ. Ubuntu 14.04 του Odroid XU3. Στα δεξιά της γραμμής εργασιών στην περιοχή των ειδοποιήσεων κάνοντας δεξί κλικ στο εικονίδιο του δικτύου επιλέγεται διαδοχικά από το μενού Edit Connections, στο αναδυόμενο παράθυρο το επιθυμητό όνομα Wi-Fi δικτύου και Edit. Στην καρτέλα General τσεκάρετε η ρύθμιση της αυτόματης σύνδεσης όταν το συγκεκριμένο δίκτυο είναι διαθέσιμο (automatically connect to this network when it is available) και στην Wi-Fi Security δίνεται ο προκαθορισμένος στην εφαρμογή mHotspot κωδικός ασφαλείας.



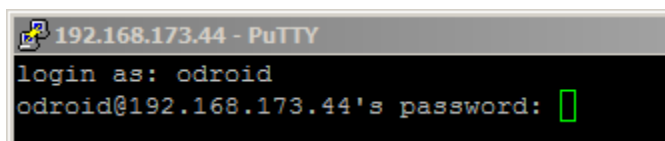
Εικόνα 8.2 Ρυθμίσεις σύνδεσης στο Λ.Σ Ubuntu

Με δεδομένη την IP διεύθυνση του Odroid XU3 η απομακρυσμένη διαχείριση του μπορεί να επιτευχθεί σε περιβάλλον MS Windows μέσω της εφαρμογής PuTTY.exe, κατά την εκκίνηση της οποίας στο παράθυρο διαλόγου της στο πεδίο Host Name (or IP address) καταχωρείται η IP διεύθυνση του Odroid XU3, επιλέγεται ως πρωτόκολλο σύνδεσης το Secure SHell και τέλος Open.



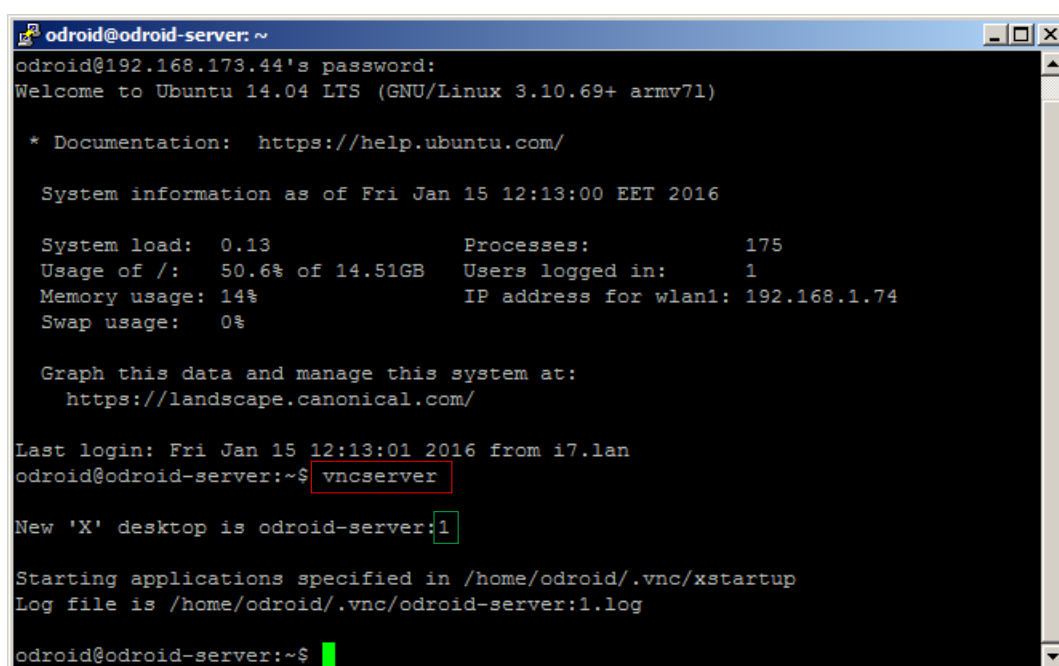
Εικόνα 8.3 Παράθυρο ρυθμίσεων της εφαρμογής PuTTY

Αν επιτευχθεί η επικοινωνία το είδος διεπαφής χρήστη (user interface) είναι με γραμμή εντολών (terminal) και ζητείται όνομα χρήστη και κωδικός ασφαλείας για την πρόσβαση (και τα δυο έχουν οριστεί odroid).



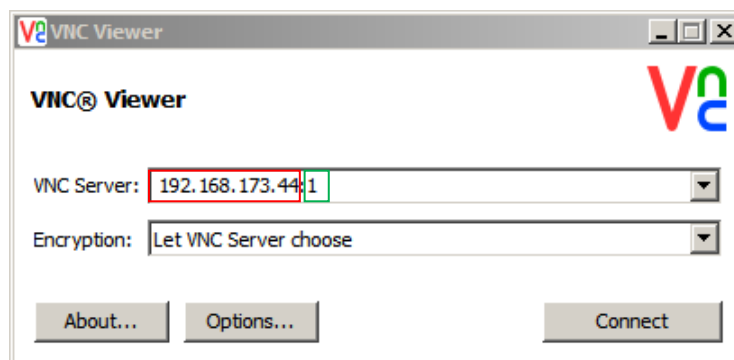
Εικόνα 8.4 Γραμμή εντολών κατά την σύνδεση μέσω της εφαρμογής PuTTY

Για σύνδεση απομακρυσμένης επιφάνειας εργασίας και διαχείριση μέσω του γραφικού περιβάλλοντος στην γραμμή εντολών δίνεται η εντολή `vncserver` ώστε να εκτελεστεί η εφαρμογή εξυπηρετητή του VNC. (Έχει προεγκατασταθεί στο Λ.Σ. Ubuntu η εφαρμογή Tightvnc Server δίνοντας στο κέλυφος την εντολή `sudo apt-get install tightvncserver`).

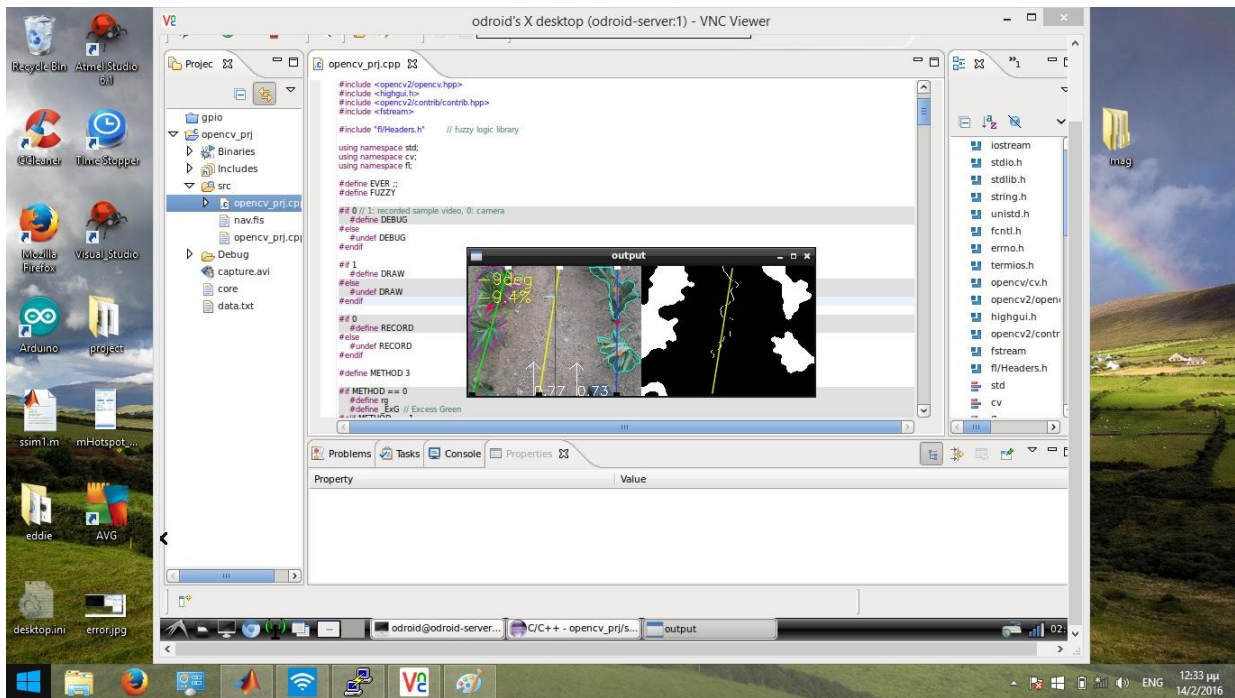


Εικόνα 8.5 Εκτέλεση της εφαρμογής vncserver στο Odroid XU3 μέσω της γραμμής εντολών

Σε περιβάλλον MS Windows εκτελείται η εφαρμογή πελάτη (VNC Viewer) στο παράθυρο διαλόγου της οποίας στο πεδίο VNC Server δίνεται η IP διεύθυνση και ο αριθμός θύρας χωριζόμενα με άνω και κάτω τελεία. Επιλέγεται σύνδεση και δίνεται ο κωδικός ασφαλείας (odroid).

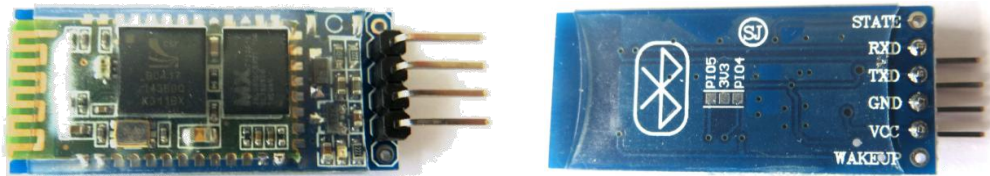


Εικόνα 8.6 Παράθυρο διαλόγου της εφαρμογής VNC Viewer



Εικόνα 8.7 Απομακρυσμένη διαχείριση του Odroid-XU3 μέσω Virtual Network Computing (VNC)

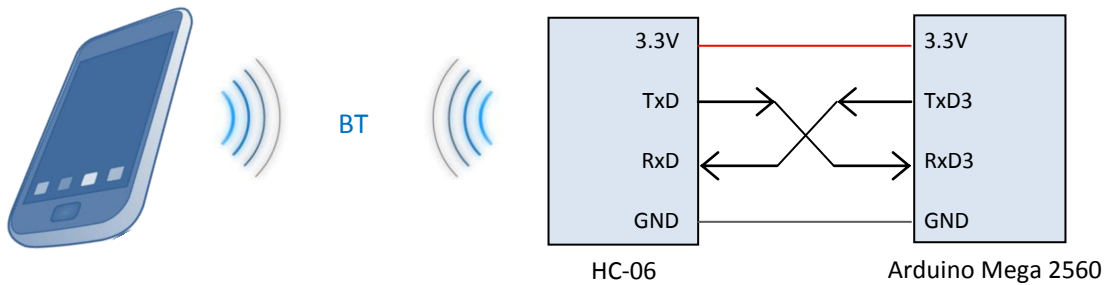
## 8.2 Διαχείριση και έλεγχος οχήματος μέσω Bluetooth



Εικόνα 8.8 Άνω και κάτω όψη του Bluetooth module HC-06

Για την ασύρματη επικοινωνία και τον έλεγχο της ρομποτικής πλατφόρμας μέσω κινητού τηλεφώνου ή Η/Υ εξοπλισμένου με πομποδέκτη Bluetooth το όχημα είναι εξοπλισμένο με το module HC-06 (τύπου slave) η λειτουργία του οποίου βασίζεται στο Bluetooth Serial Port Profile. Η αναρτηρήρια συσκευή επιδιώκει και εγκαθιστά την ζεύξη και η άλλη συσκευή περιμένει και αποδέχεται την σύνδεση παρέχοντας τον κωδικό ασφάλειας που απαιτείται.

Η τάση τροφοδοσίας της συσκευής είναι 3.3V και παρέχεται απευθείας από το αναπτυξιακό board του Arduino Mega 2560. Οι ακροδέκτες TxD (Transmit Data) και RxD (Receive Data) αποτελούν την σειριακή έξοδο και είσοδο αντίστοιχα. Το TxD από το HC-06 στέλνει δεδομένα στο RxD του ATmega 2560. Αντίστροφα το TxD του 2560 στέλνει δεδομένα στο RxD στο module.



Εικόνα 8.9 Σύνδεση του Bluetooth module HC-06 με το board του Arduino Mega 2560

Ο τρόπος λειτουργίας της σειριακής επικοινωνίας ρυθμίστηκε με:

- ρυθμός μετάδοσης 9600 bps
- μήκος χαρακτήρα 8 bits
- 1 stop bit
- ισοτιμία απενεργοποιημένη

Για τον απομακρυσμένο έλεγχο του ρομποτικού οχήματος αποστέλλεται κατάλληλος ASCII χαρακτήρας μέσω του κινητού τηλεφώνου, είτε μέσω του Η/Υ, ο οποίος αντιπροσωπεύει τον επιθυμητό τρόπο κίνησης (ακινητοποίηση, κίνηση εμπρός, πίσω, στροφή δεξιά, στροφή αριστερά, περιστροφή αριστερά, περιστροφή δεξιά ή αυτόνομη λειτουργία).

Μέσω AT εντολών οι οποίες αποστέλλονται στο HC-06 με το Serial Monitor του Arduino IDE δίνεται η δυνατότητα να τροποποιηθούν μεταξύ άλλων:

α) ο ρυθμός μετάδοσης δεδομένων (baud rate)                      AT+BAUDX όπου X = 1 .. 9

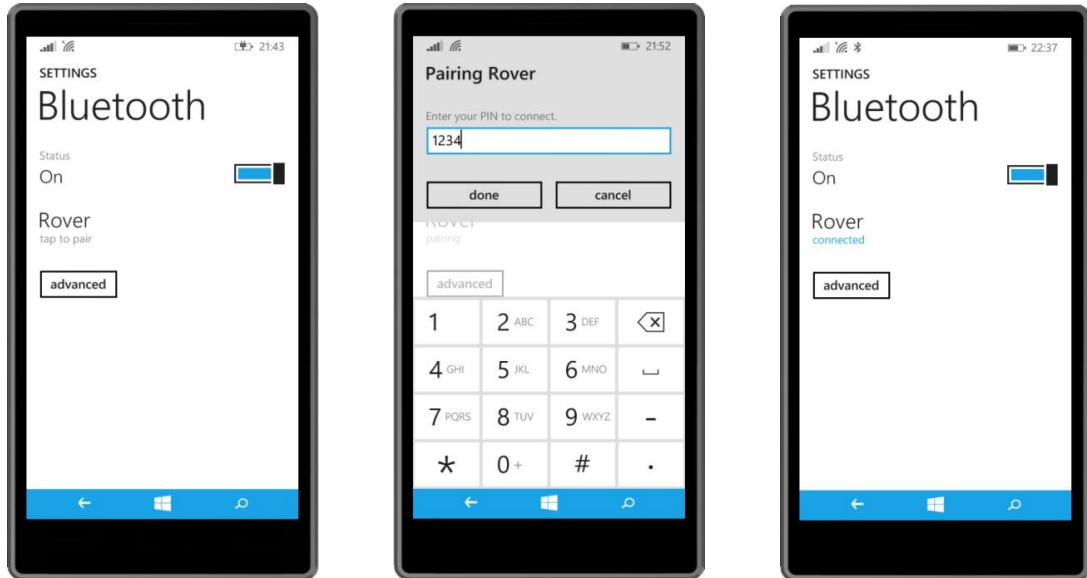
X	bps	
4	9600	default
5	19200	
6	38400	
7	57600	
8	115200	

β) το όνομα της συσκευής                      AT+NAMEdesired\_device\_name

γ) ο κωδικό πρόσβασης                      AT+PINXXXX

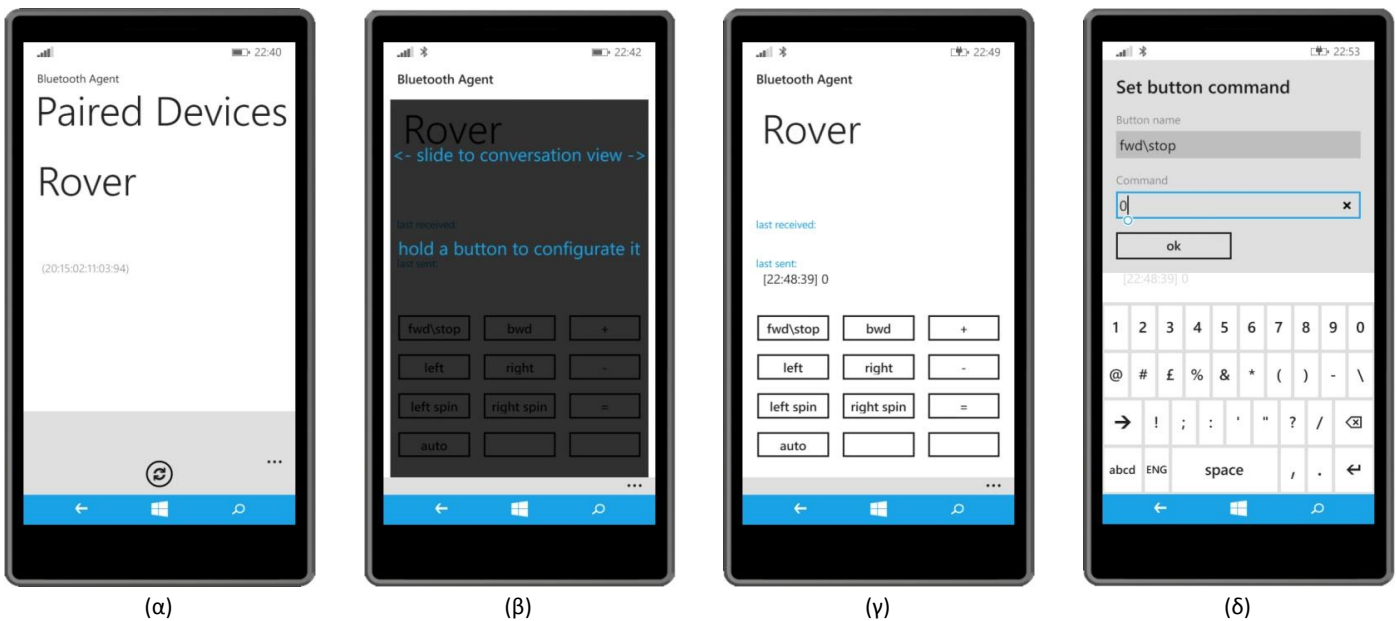
Χρησιμοποιήθηκε κινητό τηλέφωνο με λειτουργικό σύστημα windows phone 8.1 και η εφαρμογή [Bluetooth Agent 1.2.1.1](#) η οποία δίνει την δυνατότητα ασύρματης επικοινωνίας με το module HC-06 κάνοντας χρήση του προφίλ SPP.

Για την επίτευξη της ζεύξης μεταξύ του κινητού τηλεφώνου και του module HC-06 επιλέγεται από την λίστα εφαρμογών του κινητού και διαδοχικά Settings → Bluetooth → Turn Status to On. Αυτοματοποιημένα πραγματοποιείται διαδικασία σάρωσης-ανίχνευσης και εύρεσης των Bluetooth συσκευών που βρίσκονται εντός εμβέλειας. Αποτέλεσμα της αναζήτησης αποτελεί μια λίστα με τα συμβολικά ονόματα των ανιχνευθέντων συσκευών. Στο module HC-06 έχει αποδοθεί το όνομα *Rover* το οποίο και επιλέγεται ώστε να επιτευχθεί η ζεύξη. Κατά την πρώτη προσπάθεια σύνδεσης και για λόγους ασφάλειας ζητείται η εισαγωγή κωδικού (PIN: 1234). Για όσο χρονικό διάστημα η σύνδεση παραμένει ενεργή η φωτοδίοδος ένδειξης της κατάστασης του HC-06 διατηρείται σταθερά αναμμένη ενώ όσο το HC-06 είναι σε κατάσταση αναμονής ζεύξης αναβοσβήνει περιοδικά.



Εικόνα 8.10 Ενεργοποίηση Bluetooth, εισαγωγή PIN και επίτευξη σύνδεσης του κινητού τηλεφώνου με το HC-06.

Με την εκκίνηση της εφαρμογής Bluetooth Agent εμφανίζεται η αποδοθείσα ονομασία 'Rover' για το module HC-06 και η φυσική διεύθυνση (MAC address) της συσκευής (Εικόνα 8.11-α). Αυτή επιλέγεται και κατόπιν (Εικόνα 8.11-β) είτε εισερχόμαστε σε προβολή 'συνομιλίας' (conversation view) αποστέλλοντας κάθε φορά που επιθυμούμε αλλαγή του τρόπου κίνησης του οχήματος τον αντίστοιχο ASCII χαρακτήρα πληκτρολογώντας τον, είτε εισερχόμαστε στην προβολή του keypad (Εικόνα 8.11-γ) και κάθε αλλαγή στην κίνηση επιτυγχάνεται με την επιλογή του αντίστοιχου πλήκτρου οπότε και ο ASCII χαρακτήρας αποστέλλεται στο HC-06 αυτοματοποιημένα. Σε κάθε ένα από τα δώδεκα πλήκτρα του keypad με παρατεταμένη επαφή δίνεται η δυνατότητα αλλαγής α) του αναγραφόμενου σε αυτό ονόματος και β) του ASCII κωδικού που αποστέλλεται (Εικόνα 8.11-δ).



(α)

(β)

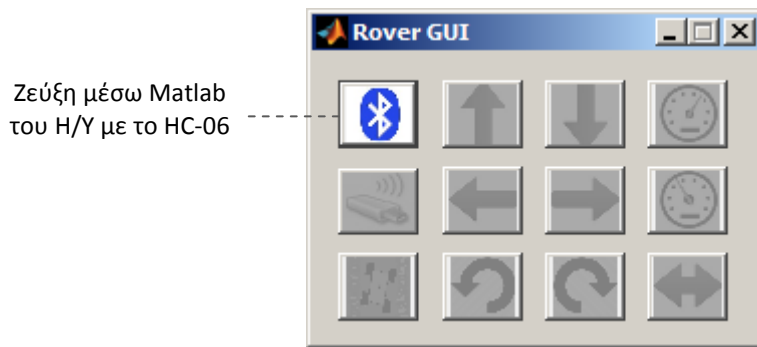
(γ)

(δ)

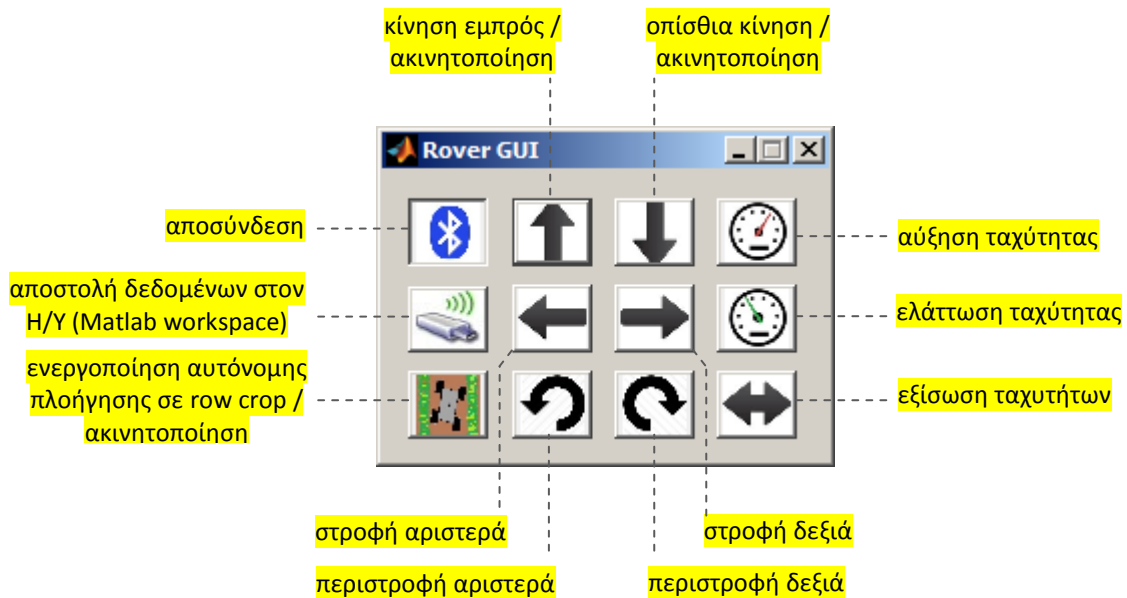
Εικόνα 8.11 Screenshots της εφαρμογής Bluetooth Agent 1.2.1.1

Για την ασύρματη επικοινωνία του Η/Υ μέσω Bluetooth με την ρομποτική πλατφόρμα δημιουργήθηκε διεπαφή χρήστη στο γραφικό περιβάλλον του λογισμικού Matlab (GUI). Η σύνδεση/αποσύνδεση του Η/Υ με το module HC-06 επιτυγχάνεται κάνοντας κλικ στο κουμπί το οποίο φέρει το εικονίδιο του λογότυπου του Bluetooth. Τα κουμπιά ελέγχου επιτελούν τις ίδιες ακριβώς λειτουργίες με εκείνα της εφαρμογής Bluetooth Agent και αποστέλλεται ο ίδιος ASCII χαρακτήρας. Με την ενεργοποίηση της αυτόνομης πλοήγησης δίνεται η δυνατότητα συλλογής δεδομένων. Κάνοντας κλικ και στο κουμπί της αποστολής δεδομένων επιστρέφεται σε πραγματικό χρόνο σε κάθε κύκλο ελέγχου στο workspace του Matlab διάγραμμα 10 στοιχείων αποτελούμενο από την:

Χρονική στιγμή μέτρησης [sec]	Γωνιακή ταχύτητα [rad/s]						Τάση συσσωρευτή [V]	Ένταση καταναλισκόμενου ρεύματος [A]	Heading [degrees]
	Σήμα αναφοράς		Εκτιμώμενη						
	Τροχών αριστεράς πλευράς	Τροχών δεξιάς πλευράς	Εμπρόσθιος αριστερός τροχός	Οπίσθιος αριστερός τροχός	Εμπρόσθιος δεξιός τροχός	Οπίσθιος δεξιός τροχός			



Εικόνα 8.12 Screenshot κατά την εκκίνηση της διεπαφής πριν τη ζεύξη



Εικόνα 8.13 Επεξήγηση των κουμπιών ελέγχου της εφαρμογής

<b>Σύμβολο</b>	<b>Αποστελλόμενος ASCII χαρακτήρας</b>	<b>Περιγραφή λειτουργίας</b>
↑	0	Κίνηση προς τα εμπρός με γραμμική αύξηση της ταχύτητας των τροχών και σταθεροποίησης τους στο 25% της τιμής του duty cycle / ακινητοποίηση οχήματος
↔	1	Ελάττωση της ταχύτητας των τροχών εκείνης της πλευράς του οχήματος με τη μεγαλύτερη τιμή και εξίσωση τους με την μικρότερη.
+	2	Βαθμιαία αύξηση της ταχύτητας
-	3	Βαθμιαία ελάττωση της ταχύτητας
←	4	Στροφή αριστερά με βαθμιαία αύξηση της ταχύτητας των τροχών της δεξιάς πλευράς
→	5	Στροφή δεξιά με βαθμιαία αύξηση της ταχύτητας των τροχών της αριστεράς πλευράς
↶	6	Αριστερή περιστροφή γύρω από τον κατακόρυφο άξονα συμμετρίας με ίσες κατά απόλυτη τιμή ταχύτητες αλλά με αντίθετη φορά
↷	7	Δεξιά περιστροφή γύρω από τον κατακόρυφο άξονα συμμετρίας με ίσες κατά απόλυτη τιμή ταχύτητες αλλά με αντίθετη φορά
↓	9	Οπίσθια κίνηση με περιστροφή της web κάμερας κατά 180°
	a	Αυτόνομη πλοήγηση σε row crop

## Κεφάλαιο

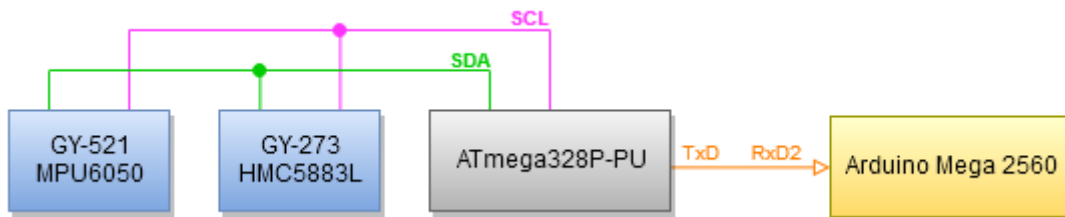
## 9

## 9. Προσανατολισμός οχήματος

Στο κεφάλαιο αυτό παρουσιάζεται η κατασκευασθείσα διάταξη για την εκτίμηση της γωνίας προσανατολισμού (heading) του οχήματος ως προς την αρχική του θέση. Η γωνία heading υπολογίζεται μέσω των δεδομένων μέτρησης του μαγνητόμετρου και συναρτήσει των γωνιών στροφής roll και pitch (tilt compensation) λαμβάνοντας υπόψη το σφάλμα πόλωσης του αισθητηρίου μόνο από στατικές μαγνητικές πηγές. Για την εκτίμηση των γωνιών roll και pitch εφαρμόζεται διακριτό φίλτρο Kalman για την συγχώνευση των δεδομένων του επιταχυνσιόμετρου και του γυροσκοπίου.

### 9.1 Υλοποιηθείσα διάταξη

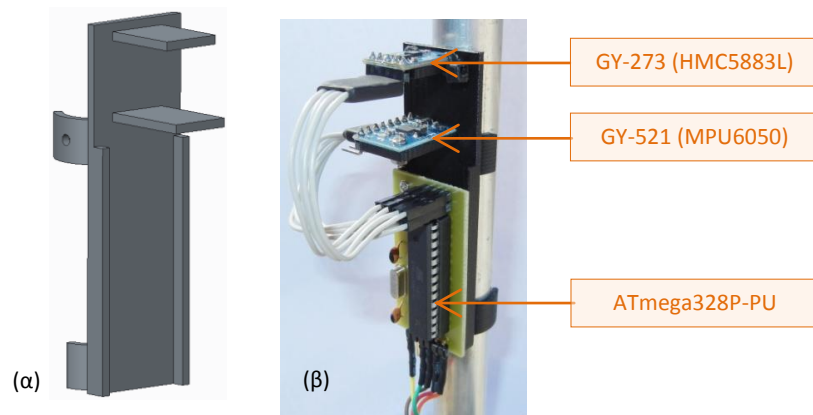
Ο υπολογισμός του προσανατολισμού (heading) του οχήματος ως προς την αρχική του θέση πραγματοποιείται με την ανάγνωση από έναν μικροελεγκτή ATmega328P-PU των τιμών των αισθητηρίων του επιταχυνσιόμετρου και γυροσκοπίου του τυπωμένου κυκλώματος GY-521 (MPU6050) και του μαγνητόμετρου (GY-273, HMC5883L). Τα πρωτογενή δεδομένα από τα αισθητήρια συλλέγονται από τον μικροελεγκτή, μέσω του τοπικού διαύλου I<sup>2</sup>C, συγχωνεύονται-συνδυάζονται κατάλληλα και αποστέλλεται η γωνία yaw μέσω του UART στην αναπτυξιακή πλακέτα του Arduino Mega 2560.



Εικόνα 9.1 Συνδεσμολογία διάταξης

Λόγω του ότι το μαγνητόμετρο είναι πολύ ευαίσθητο και επηρεάζεται από μαγνητικά πεδία του περιβάλλοντος χώρου για την απομάκρυνση του από τον λοιπό ηλεκτρονικό εξοπλισμό του οχήματος σχεδιάστηκε μέσω της εφαρμογής PTC Creo Parametric 2.0 μια βάση στήριξης τόσο των αισθητηρίων αυτών όσο και του τυπωμένου κυκλώματος του μικροελεγκτή και κατασκευάστηκε από θερμοπλαστικό υλικό τύπου ABS με τη βοήθεια συστήματος τρισδιάστατης εκτύπωσης. Η βάση αυτή είναι κατάλληλα σχεδιασμένη ούτως ώστε συγκρατείται στον στύλο τοποθέτησης της κάμερας.



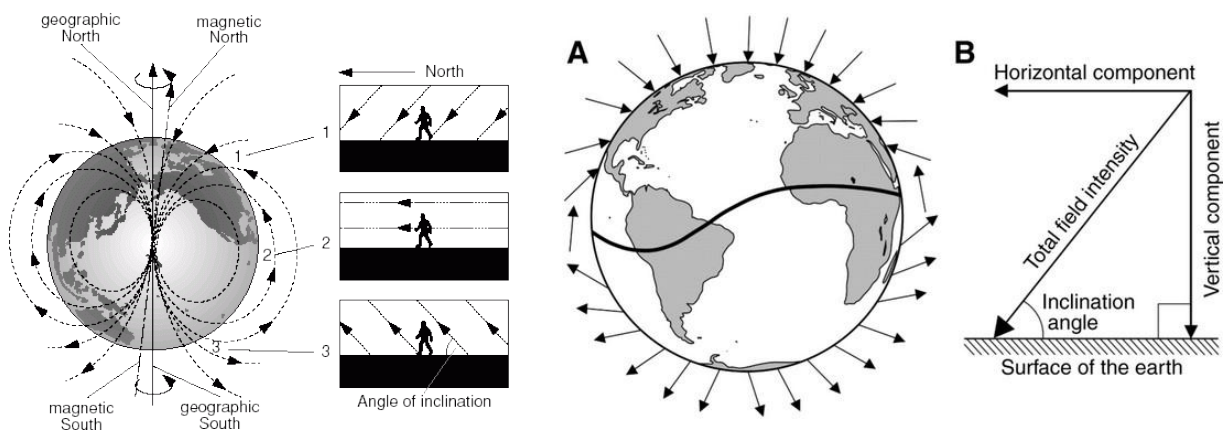


Εικόνα 9.2 α) 3D CAD μοντέλο της βάσης στήριξης β) φωτογραφία της διάταξης

Η τροφοδοσία των αισθητηρίων είναι στα 3.3V και παρέχεται απευθείας από την πλακέτα του Arduino Mega 2560 χάρη στο ενσωματωμένο ρυθμιστή τάσης που αυτή διαθέτει και του τυπωμένου κυκλώματος του μικροελεγκτή ATmega328 στα 5V μέσω ενός εκ των κυκλωμάτων μετατροπής από DC σε DC.

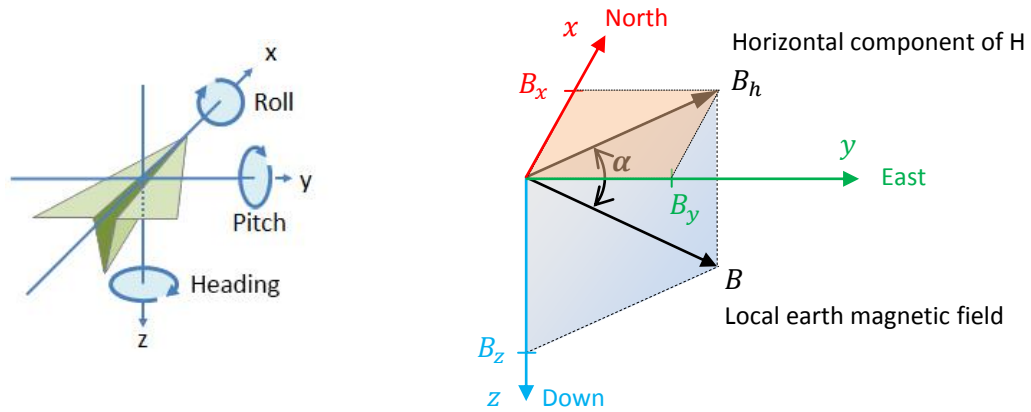
## 9.2 Μαγνητόμετρο

Τα μαγνητόμετρα βασίζονται στη μέτρηση του μαγνητικού πεδίου της Γης. Η Γη μπορεί να θεωρηθεί ως ένα μαγνητικό δίπολο με τις θέσεις των πόλων του να βρίσκονται κοντά στο γεωγραφικό βόρειο και νότιο πόλο. Το διάνυσμα της έντασης του μαγνητικού πεδίου είναι παράλληλο με την επιφάνεια της Γης κοντά στον ισημερινό. Στο βόρειο ημισφαίριο το διάνυσμα έχει φορά προς την επιφάνεια της Γης και στο νότιο ημισφαίριο από την επιφάνεια της Γης και προς τα έξω.



Εικόνα 9.3 Δυναμικές γραμμές του γήινου μαγνητικού πεδίου και μαγνητική έγκλιση

Η γωνία  $a$  μεταξύ του διανύσματος της έντασης του γήινου μαγνητικού πεδίου και του οριζόντιου επιπέδου XY ονομάζεται μαγνητική έγκλιση (dip ή inclination angle). Η γωνία έγκλισης κυμαίνεται από  $0^\circ$  στον ισημερινό και μέχρι  $+90^\circ$  στο βόρειο ημισφαίριο ή μέχρι  $-90^\circ$  στο νότιο ημισφαίριο. Η γωνία μεταξύ του μαγνητικού και γεωγραφικού βορά ορίζεται ως απόκλιση (declination angle) και κυμαίνεται στο εύρος των  $\pm 20^\circ$ .



Εικόνα 9.4 Πλαίσιο συντεταγμένων και γωνίες στροφής roll, pitch, heading

Το επίπεδο  $XY$  είναι παράλληλο με την επιφάνεια της Γης και το διάνυσμα της έντασης του βαρυτικού πεδίου της Γης κάθετο σ' αυτό. Η ένταση του μαγνητικού πεδίου  $B$  έχει μια αμετάβλητη οριζόντια συνιστώσα  $B_h$  στο επίπεδο  $XY$  με φορά προς το μαγνητικό βορά της Γης. Η γωνία heading ορίζεται ως η γωνία στροφής μεταξύ του άξονα  $x$  και του μαγνητικού βορά στο οριζόντιο επίπεδο  $XY$ . Η γωνία pitch είναι η γωνία που σχηματίζει ο άξονας  $x$  με το οριζόντιο επίπεδο  $XY$  και ως γωνία roll ορίζεται η γωνία στροφής μεταξύ του άξονα  $y$  και του οριζόντιου επιπέδου  $XY$ . Η φορά περιστροφής για όλες τις γωνίες στροφής θεωρείται αυτή των δεικτών του ρολογιού (clockwise). Όταν το μαγνητόμετρο είναι σε οριζόντια επίπεδη θέση σε σχέση με την επιφάνεια της Γης οι γωνίες roll και pitch είναι 0 και η γωνία heading είναι

$$\text{heading} = \arctan \frac{B_y}{B_x}$$

οι τιμές των  $B_x$ ,  $B_y$  αποτελούν δεδομένα μέτρησης του μαγνητόμετρου στους αντίστοιχους άξονες. Αν το μαγνητόμετρο είναι υπό κλίση οι μετρούμενες τιμές των  $B_x$ ,  $B_y$  αλλάζουν και η προκύπτουσα τιμή της γωνίας heading δεν είναι ακριβής. Για την αποφυγή αυτού του σφάλματος οι γωνίες  $\varphi$ ,  $\theta$  (roll, pitch) υπολογίζονται από την συγχώνευση των πρωτογενών δεδομένων του επιταχυνσιόμετρου και του γυροσκοπίου 3 αξόνων. Στην προκειμένη περίπτωση δεν μας ενδιαφέρει η πλοήγηση σε σχέση με το μαγνητικό βορά αλλά η σχετική γωνία yaw ως προς την αρχική θέση.

Αν υπάρχουν υλικά που μαγνητίζονται στο χώρο λειτουργίας του μαγνητόμετρου επηρεάζονται οι μετρήσεις του ή και προκαλείται μόνιμο σφάλμα πόλωσης στο αισθητήριο. Οι μετρήσεις του αισθητήρα διαστρεβλώνονται από πεδία τα οποία κατηγοριοποιούνται σε στατικές πηγές (hard iron field distortion) και δυναμικές πηγές μαγνητικής δραστηριότητας (soft iron).

Στην πρώτη κατηγορία παράγεται προσθετική διαστρέβλωση στο πεδίο με αποτέλεσμα οι τιμές του μαγνητόμετρου να προκύπτουν αυξημένες κατά μια σταθερή ποσότητα. Πηγές τέτοιων σφαλμάτων μπορεί να είναι κινητήρες και τμήματα της μεταλλικής πλατφόρμας. Η διαταραχή αυτή αντισταθμίζεται μέσω μιας απλής διαδικασίας και το σφάλμα πόλωσης αφαιρείται από κάθε μέτρηση. Στην δεύτερη κατηγορία των μεταβαλλόμενων μαγνητικών πεδίων επίδρασης η αντισταθμηση είναι μια πολύπλοκη διαδικασία που ξεφεύγει από το σκοπό της παρούσας αναφοράς.

Αν ο άξονας  $x$  του μαγνητόμετρου είναι παράλληλος με την συνιστώσα  $B_x$  η ένταση του γήινου μαγνητικού δίνεται από το διάνυσμα

$$\begin{bmatrix} B_x \\ 0 \\ B_z \end{bmatrix} = \begin{bmatrix} B \cos \alpha \\ 0 \\ B \sin \alpha \end{bmatrix} = B \begin{bmatrix} \cos \alpha \\ 0 \\ \sin \alpha \end{bmatrix}$$

ως προς το αδρανειακό πλαίσιο συντεταγμένων η έξοδος του μαγνητόμετρου είναι

$$M = R_x(\varphi)R_y(\theta)R_z(\psi)B \begin{bmatrix} \cos \alpha \\ 0 \\ \sin \alpha \end{bmatrix} + V$$

όπου  $V = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$  το σφάλμα πόλωσης από τις στατικές πηγές μαγνητικής δραστηριότητας

$$R_x(\varphi)R_y(\theta)R_z(\psi) \begin{bmatrix} B \cos \alpha \\ 0 \\ B \sin \alpha \end{bmatrix} = M - V \rightarrow R_z(\psi) \begin{bmatrix} B \cos \alpha \\ 0 \\ B \sin \alpha \end{bmatrix} = R_y(-\theta)R_x(-\varphi)(M - V) \rightarrow$$

$$\begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B \cos \alpha \\ 0 \\ B \sin \alpha \end{bmatrix} = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\varphi & -s_\varphi \\ 0 & s_\varphi & c_\varphi \end{bmatrix} \begin{bmatrix} M_x - V_x \\ M_y - V_y \\ M_z - V_z \end{bmatrix} \rightarrow$$

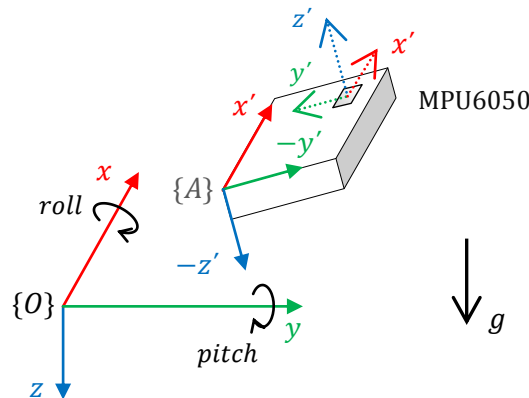
$$\begin{bmatrix} B c_\alpha c_\psi \\ -B c_\alpha s_\psi \\ B s_\alpha \end{bmatrix} = \begin{bmatrix} c_\theta & s_\theta s_\varphi & c_\theta s_\varphi \\ 0 & c_\varphi & -s_\varphi \\ -s_\theta & s_\theta c_\varphi & c_\theta c_\varphi \end{bmatrix} \begin{bmatrix} M_x - V_x \\ M_y - V_y \\ M_z - V_z \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} B c_\alpha c_\psi \\ -B c_\alpha s_\psi \\ B s_\alpha \end{bmatrix} = \begin{bmatrix} (M_x - V_x)c_\theta + (M_y - V_y)s_\theta s_\varphi + (M_z - V_z)c_\theta s_\varphi \\ (M_y - V_y)c_\varphi - (M_z - V_z)s_\varphi \\ -(M_x - V_x)s_\theta + (M_y - V_y)s_\theta c_\varphi + (M_z - V_z)c_\theta c_\varphi \end{bmatrix}$$

$$\frac{-B c_\alpha s_\psi}{B c_\alpha c_\psi} = \frac{(M_y - V_y)c_\varphi - (M_z - V_z)s_\varphi}{(M_x - V_x)c_\theta + (M_y - V_y)s_\theta s_\varphi + (M_z - V_z)c_\theta s_\varphi} \rightarrow$$

$$\tan \psi = \frac{-(M_y - V_y)c_\varphi + (M_z - V_z)s_\varphi}{(M_x - V_x)c_\theta + (M_y - V_y)s_\theta s_\varphi + (M_z - V_z)c_\theta s_\varphi}$$

### 9.3 Εύρεση γωνιών roll και pitch από το επιταχυνσιόμετρο



Εικόνα 9.5 Πλαίσιο συντεταγμένων αισθητήρα ως προς το αδρανειακό σύστημα αναφοράς

Αν ο άξονας Z του αισθητηρίου είναι ευθυγραμμισμένος με το βαρυτικό πεδίο της γης αυτό ανιχνεύεται μόνο στον συγκεκριμένο άξονα και δίνει ένδειξη ίση με  $1g$ . Ως προς το αδρανειακό πλαίσιο συντεταγμένων  $\{O\}$  η έξοδος του επιταχυνσιόμετρου είναι

$$A = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

όπου  $R$  ο πίνακας στροφής. Οι τρεις βασικές γωνίες ( $\varphi = roll, \theta = pitch, \psi = yaw$ ) περιγράφουν πλήρως τον προσανατολισμό γύρω από τους σταθερούς άξονες του ακίνητου συστήματος  $\{O\}$ . Η σύνθετη στροφή βρίσκεται με τον κατά σειρά πολλαπλασιασμό από αριστερά με τους πίνακες των αντίστοιχων βασικών στροφών

$$A = R_x(\varphi)R_y(\theta)R_z(\psi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow$$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\varphi & s_\varphi \\ 0 & -s_\varphi & c_\varphi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow$$

$$A = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ c_\varphi s_\psi - s_\varphi s_\theta c_\psi & c_\varphi c_\psi + s_\varphi s_\theta s_\psi & s_\varphi c_\theta \\ s_\varphi s_\psi + c_\varphi s_\theta c_\psi & s_\varphi c_\psi - c_\varphi s_\theta s_\psi & c_\varphi c_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow$$

$$A = \begin{bmatrix} -s_\theta \\ s_\varphi c_\theta \\ c_\varphi c_\theta \end{bmatrix}$$

διαιρούμε με το μέτρο του ώστε να προκύψει το μοναδιαίο διάνυσμα

$$\frac{A}{\|A\|} = \frac{1}{\sqrt{A_x^2 + A_y^2 + A_z^2}} \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} -s_\theta \\ s_\varphi c_\theta \\ c_\varphi c_\theta \end{bmatrix}$$

από την τελευταία σχέση για την γωνία  $\varphi$  βρίσκουμε

$$\boxed{\tan\varphi = \frac{A_y}{A_z}}$$

και για την  $\theta$  θα είναι

$$\tan\theta = -\frac{A_x}{A_y} \sin\varphi \rightarrow \tan\theta = -\frac{A_x}{A_z} c_\varphi$$

ισχύει ότι

$$s_\varphi^2 + c_\varphi^2 = 1 \rightarrow \tan^2 \varphi + 1 = \frac{1}{c_\varphi^2} \rightarrow c_\varphi^2 = \frac{1}{1 + \tan^2 \varphi} \rightarrow c_\varphi^2 = \frac{A_z^2}{A_y^2 + A_z^2} \rightarrow c_\varphi = \frac{A_z}{\sqrt{A_y^2 + A_z^2}}$$

άρα

$$\boxed{\tan\theta = -\frac{A_x}{\sqrt{A_y^2 + A_z^2}}}$$

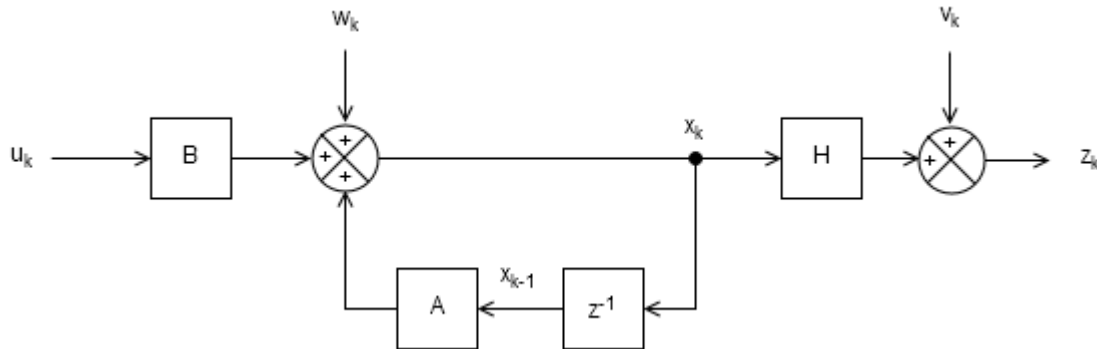
## 9.4 Διακριτό φίλτρο Kalman

Το φίλτρο Kalman κάνει δυνατή την πρόβλεψη της δυναμικής κατάστασης ενός συστήματος λαμβάνοντας υπόψη την επίδραση των προηγούμενων παρατηρήσεων. Στοχεύει στην εκτίμηση της στιγμιαίας κατάστασης (state) ενός γραμμικού δυναμικού συστήματος που διαταράσσεται από λευκό θόρυβο και το οποίο περιγράφεται από μια εξίσωση διαφορών της μορφής

$$x_k = Ax_{k-1} + Bu_k + w_k$$

και μέτρηση η οποία δίνεται από την εξίσωση

$$z_k = Hx_k + v_k$$



Εικόνα 9.6 Block διάγραμμα διακριτού συστήματος

Ο  $n \times n$  πίνακας μετάβασης  $A$  συσχετίζει την προηγούμενη κατάσταση κατά την διακριτή στιγμή  $k - 1$  με την τρέχουσα κατάσταση την στιγμή  $k$ . Ο  $n \times 1$  πίνακας  $B$  συνδέει την είσοδο ελέγχου  $u$  με το  $n \times 1$  διάνυσμα κατάστασης του συστήματος  $x_k$ .  $z_k$  είναι μια μέτρηση ( $m \times 1$ ) τη χρονική στιγμή  $t_k$  και  $H$  είναι ένας πίνακας ( $m \times n$ ) μέτρησης ή παρατήρησης που συνδέει το  $x_k$  με το  $z_k$ .

Η τυχαία μεταβλητή  $w_k$  μοντελοποιεί λευκό γκαουσιανό θόρυβο μηδενικής μέσης τιμής και με συσχετιστικότητα που δίνεται από τον πίνακα ( $n \times n$ )  $Q$ .

$$w_k \sim N(0, Q_k)$$

Ο θόρυβος του συστήματος αναπαριστά τις τυχαίες επιδράσεις τις οποίες αυτό υφίσταται και την ατελή μοντελοποίηση της δυναμικής κατάστασης του. Η μεταβλητή  $v_k \sim N(0, R_k)$  αναπαριστά τον θόρυβο στις μετρήσεις των αισθητηρίων με  $R_k$  ο πίνακας συσχετιστικότητας (covariance matrix). Ο θόρυβος που εισάγεται στο φίλτρο και ο θόρυβος του συστήματος πρέπει να είναι γνωστός. Μπορεί να μοντελοποιηθεί με γκαουσιανή κανονική κατανομή. Η συνάρτηση που περιγράφει αυτή την κατανομή είναι

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Το φίλτρο Kalman γενικότερα:

- παρέχει μια βέλτιστη εκτίμηση των μεταβλητών κατά τέτοιο τρόπο ώστε το εκάστοτε σφάλμα να ελαχιστοποιείται στατιστικά
- χρησιμοποιεί τις μετρήσεις για να διορθώσει την πρόβλεψη
- παρέχει εκτιμήσεις με σφάλμα μικρότερο του σφάλματος μέτρησης
- δεν απαιτεί περιοδικές ενημερώσεις μετρήσεων
- έχει την δυνατότητα να διαχειρίζεται δεδομένα από ανόμοιους αισθητήρες, οι οποίοι δεν είναι καν συγχρονισμένοι (πολυαισθητηριακή συγχώνευση πληροφοριών).

### 9.4.1 Εξισώσεις αναπροσαρμογών

Οι εξισώσεις που απαρτίζουν το φίλτρο διακρίνονται σε:

- εξισώσεις χρονικών αναπροσαρμογών (πρόβλεψη)
- εξισώσεις αναπροσαρμογών εξ αιτίας των μετρήσεων (διόρθωση)

Υπολογίζονται για κάθε επόμενη χρονική στιγμή η κατάσταση και συμμεταβλητότητα  $P$  του σφάλματος. Οι εξισώσεις του φίλτρου Kalman επαναλαμβάνονται κυκλικά και κάθε φορά που υπάρχει μια διαθέσιμη μέτρηση και αφού ολοκληρωθεί η πρόβλεψη της κατάστασης για αυτόν τον κύκλο από το μοντέλο του συστήματος εκτελείται το φίλτρο αυτό καθαυτό για την βέλτιστη εκτίμηση της τρέχουσας κατάστασης.

Η εκτίμηση της κατάστασης πριν την ενσωμάτωση οποιονδήποτε νέων μετρήσεων σημειώνεται με  $\hat{x}^-$  όπου το σύμβολο  $\hat{\phantom{x}}$  δηλώνει μια εκτίμηση και το μείον στον εκθέτη την εκτίμηση πριν την ενσωμάτωση των μετρήσεων.

#### Πρόβλεψη (time update)

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k && \text{πρόβλεψη της κατάστασης βάση του μοντέλου του συστήματος} \\ P_k^- &= AP_{k-1}A^T + Q_k && \text{η αβεβαιότητα αυξάνει λόγω της πρόβλεψης} \end{aligned}$$

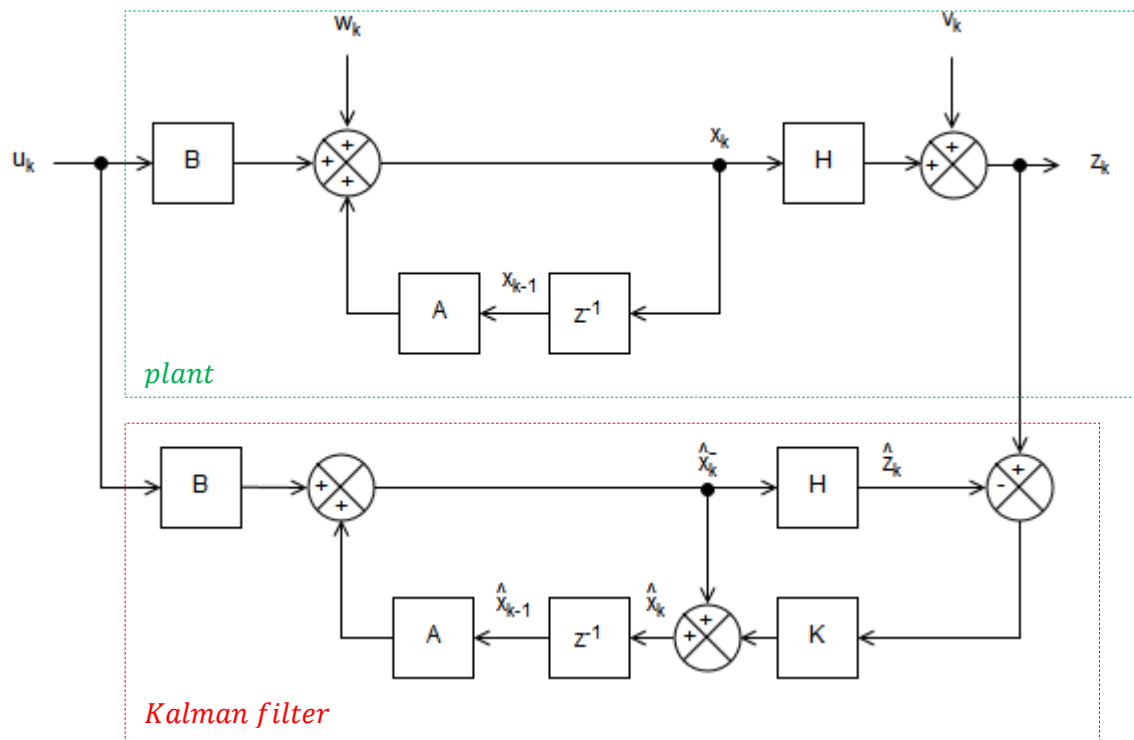
#### Διόρθωση (measurement update)

$$\begin{aligned} \hat{z}_k &= H\hat{x}_k^- && \text{εκτιμώμενη μέτρηση} \\ y &= z_k - \hat{z}_k && \text{διαφορά μεταξύ της πραγματικής και της εκτιμώμενης μέτρησης} \\ S &= HP_k^-H^T + R_k && \text{συνδιακύμανση μέτρησης} \\ K &= P_k^-H^TS^{-1} && \text{το βάρος το οποίο αποδίδεται στην μέτρηση} \\ \hat{x}_k &= \hat{x}_k^- + Ky && \text{νέα εκτίμηση της κατάστασης με ενσωματωμένη την μέτρηση} \\ P_k &= (I - KH)P_k^- && \text{η αβεβαιότητα μειώνεται λόγω της διόρθωσης} \end{aligned}$$

Οι δυο πρώτες εξισώσεις πραγματοποιούνται σε υψηλή συχνότητα και οι υπόλοιπες όταν υπάρχουν διαθέσιμες μετρήσεις. Ο πίνακας  $K$  υπολογίζεται ώστε να ελαχιστοποιεί τη συμμεταβλητότητα (covariance) της a posteriori εκτίμησης σφάλματος  $P_k$ . Μετά το τέλος κάθε κύκλου λειτουργίας προωθείται η τελευταία a posteriori εκτίμηση στις εξισώσεις πρόβλεψης ώστε να προκύψει μια νέα a priori εκτίμηση. Εν συνεχεία καταγράφεται μια μέτρηση και υπολογίζεται μια a posteriori εκτίμηση της κατάστασης με βάση τη νέα μέτρηση. Το κέρδος του φίλτρου ανανεώνεται με κάθε νέα μέτρηση.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$$



Εικόνα 9.7 Kalman Estimator

## 9.5 Εφαρμογή του διακριτού φίλτρου Kalman για τον συγκερασμό των δεδομένων του επιταχυνσιόμετρου και του γυροσκοπίου

Οι τιμές εξόδου του γυροσκοπίου βραχυπρόθεσμα κρίνονται ικανοποιητικές ενώ του επιταχυνσιόμετρου μακροπρόθεσμα. Για την εκτίμηση των γωνιών στροφής *roll* και *pitch* εφαρμόζεται το διακριτό φίλτρο Kalman για την συγχώνευση των πρωτογενών δεδομένων του επιταχυνσιόμετρου και το γυροσκοπίου της IMU GY-521. Με την συνδυασμένη χρήση των εξόδων που παρέχουν τα αισθητήρια αυξάνεται η ευρωστία των μετρήσεων. Οι μετρήσεις από το επιταχυνσιόμετρο και το γυροσκόπιο χρησιμοποιούνται για να διορθώσουν και να σταθεροποιήσουν τον προσανατολισμό ως προς μαγνητικό πεδίο της Γης αποφεύγοντας έτσι την συσσώρευση λαθών.

Είσοδος ελέγχου  $u_k$  θεωρείται η μέτρηση του γυροσκοπίου τη χρονική στιγμή  $k$  δηλαδή ο ρυθμός περιστροφής  $\dot{\theta}_k$  γύρω από κάποιον άξονα ( $x$  ή  $\psi$ ) και μετράται σε μοίρες ανά δευτερόλεπτο ( $^\circ/s$ ). Λόγω της συσσώρευσης σφάλματος μέτρησης υπάρχει διολίσθηση πόλωσης (bias drift)  $\hat{\theta}_b$  στη μέτρηση του αισθητηρίου. Για να υπολογιστεί η πραγματική τιμή του ρυθμού περιστροφής αφαιρείται η διολίσθηση από την τρέχουσα μέτρηση του γυροσκοπίου.

$$\dot{\theta}_k - (\dot{\theta}_b)_{k-1} = \frac{\theta_k - \theta_{k-1}}{\Delta t} \rightarrow \theta_k = \theta_{k-1} - \Delta t (\dot{\theta}_b)_{k-1} + \Delta t \dot{\theta}_k$$

ως διάνυσμα κατάστασης  $x_k$  θεωρείται ο πίνακας

$$x_k = \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_k$$

η εξίσωση κατάστασης είναι

$$x_k = Ax_{k-1} + Bu_k + w_k$$

$$\begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_k = A \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k-1} + B\dot{\theta}_k + w_k \rightarrow \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_k = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_{k-1} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \dot{\theta}_k + w_k$$

ο πίνακας μετάβασης  $A$  που συνδέει την προηγούμενη κατάσταση κατά την διακριτή στιγμή  $k - 1$  με την τρέχουσα κατάσταση την στιγμή  $k$  είναι

$$A = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix}$$

και ο πίνακας  $B$  που συσχετίζει την είσοδο ελέγχου  $u_k$  με το διάνυσμα κατάστασης του συστήματος

$$B = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix}$$

με  $w_k \sim N(0, Q_k)$  όπου  $Q_k$  ο  $2 \times 2$  πίνακας συνδιακύμανσης<sup>16</sup>

$$Q_k = \begin{bmatrix} Q_\theta & 0 \\ 0 & Q_{\dot{\theta}_b} \end{bmatrix} \Delta t$$

οι μεταβλητές  $\theta$  (μέτρηση επιταχυνσιόμετρου) και  $\dot{\theta}_b$  είναι *ασυσχέτιστες (ανεξάρτητες)* οπότε  $cov(\theta, \dot{\theta}_b) = 0$ . Οι διακυμάνσεις  $Q_\theta$ ,  $Q_{\dot{\theta}_b}$  των μεταβλητών  $\theta$  και  $\dot{\theta}_b$  αντίστοιχα πολλαπλασιάζονται με  $\Delta t$  πράγμα το οποίο υποδηλώνει την αύξηση του θορύβου του συστήματος όσο αυξάνει ο χρόνος από την τελευταία ενημέρωση του διανύσματος κατάστασης. Οι τιμές των  $Q_\theta$ ,  $Q_{\dot{\theta}_b}$  πρέπει να είναι γνωστές και όσο μεγαλύτερες οριστούν τόσοι περισσότεροι θόρυβος υπεισέρχεται στην εκτίμηση του διανύσματος κατάστασης. Αν η εκτιμώμενη γωνία αποκλίνει πρέπει να αυξηθεί η τιμή του  $Q_{\dot{\theta}_b}$ . Αν τείνει να καθυστερεί δίνεται μεγάλη εμπιστοσύνη στην εκτίμηση της γωνίας και η τιμή της  $Q_\theta$  θα πρέπει να ελαττωθεί ώστε να παρουσιάζεται καλύτερη απόκριση.

Για την μέτρηση  $z_k$  ισχύει ότι  $z_k = Hx_k + v_k$  και λόγω του ότι μετράται η τιμή της γωνίας  $\theta$  από το επιταχυνσιόμετρο είναι  $H = [1 \ 0]$ . Ο θόρυβος  $v_k$  ο οποίος επιδρά στη μέτρηση είναι ίσος με τη διακύμανση της μεταβλητής αυτής. Αν η τιμή της διακύμανσης του θορύβου στη μέτρηση είναι μεγάλη το φίλτρο θα παρουσιάζει αργή απόκριση λόγω του ότι δίνεται στις νέες μετρήσεις μικρότερη εμπιστοσύνη, διαφορετικά αν είναι πολύ μικρή τότε η ποσότητα του θορύβου που υπεισέρχεται στη μέτρηση θα είναι μεγάλη δεδομένου ότι δίνεται περισσότερο αξιοπιστία στις πρόσφατες μετρήσεις του επιταχυνσιόμετρου.

Στην συγκεκριμένη εφαρμογή οι τιμές των διακυμάνσεων των μεταβλητών  $\theta$  και  $\dot{\theta}_b$  και της μέτρησης της γωνίας τέθηκαν ίσες με

$$Q_\theta = 0.001, Q_{\dot{\theta}_b} = 0.003 \text{ και } v = 0.03$$

<sup>16</sup> $Q = \begin{bmatrix} var(x) & cov(x, y) \\ cov(x, y) & var(y) \end{bmatrix}$  με  $var(x) = \frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N-1}$  και  $cov(x, y) = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$



Πρόβλεψη

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \rightarrow \hat{x}_k^- = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \hat{x}_{k-1} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \dot{\theta}_k \rightarrow \hat{\theta}_k^- = \hat{\theta}_{k-1} + \Delta t \left( \dot{\theta}_k - \left( \hat{\theta}_b \right)_{k-1} \right)$$

προκύπτει ότι μια a priori εκτίμηση της γωνίας  $\hat{\theta}_k^-$  είναι ίση με την προηγούμενη εκτίμηση της  $\hat{\theta}_{k-1}$  συν την τον ρυθμό μεταβολής της γωνίας απαλλαγμένο από την διολίσθηση επί το χρόνο  $\Delta t$ . Προγραμματιστικά σε γλώσσα C αυτό επιτυγχάνεται με τις ακόλουθες εντολές

```
rate = newRate - bias;
angle += dt * rate;
```

ο 2x2 πίνακας  $P$  εκτιμάται (a priori) βάση της προηγούμενης του εκτίμησης

$$P_k^- = AP_{k-1}A^T + Q_k \rightarrow$$

$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_k^- = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k-1} \begin{bmatrix} 1 & 0 \\ -\Delta t & 1 \end{bmatrix} + \begin{bmatrix} Q_\theta & 0 \\ 0 & Q_{\hat{\theta}_b} \end{bmatrix} \Delta t =$$

$$\begin{bmatrix} (P_{00})_{k-1} + \Delta t(\Delta t(P_{11})_{k-1} - (P_{01})_{k-1} - (P_{10})_{k-1} + Q_\theta) & (P_{01})_{k-1} - \Delta t(P_{11})_{k-1} \\ (P_{10})_{k-1} - \Delta t(P_{11})_{k-1} & (P_{11})_{k-1} + Q_{\hat{\theta}_b} \Delta t \end{bmatrix}$$

οι προηγούμενες εξισώσεις υλοποιούνται σε γλώσσα C ως εξής:

```
P[0][0] += dt * (dt*P[1][1] - P[0][1] - P[1][0] + Q_angle);
P[0][1] -= dt * P[1][1];
P[1][0] -= dt * P[1][1];
P[1][1] += Q_bias * dt;
```

Διόρθωση

η διαφορά μεταξύ της τρέχουσας μέτρησης της γωνίας από το επιταχυνσιόμετρο και της εκτιμώμενης μέτρησης της είναι

$$y = z_k - H\hat{x}_k^- = \theta_k - [1 \ 0] \begin{bmatrix} \hat{\theta} \\ \hat{\theta}_b \end{bmatrix}_k^- \rightarrow y = \theta_k - \hat{\theta}_k^-$$

```
y = newAngle - angle;
```

ο πίνακας συνδιακύμανσης

$$S = HP_k^-H^T + R_k = [1 \ 0] \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_k^- \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R_k = (P_{00})_k^- + R$$

```
S = P[0][0] + R;
```

το κέρδος του φίλτρου είναι ένας 2x1 πίνακας

$$K = P_k^-H^T S^{-1} \rightarrow \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_k^- \begin{bmatrix} 1 \\ 0 \end{bmatrix} S^{-1} = \frac{\begin{bmatrix} P_{00} \\ P_{11} \end{bmatrix}_k^-}{S}$$

```
K[0] = P[0][0] / S;
K[1] = P[1][0] / S;
```

η νέα εκτίμηση του διανύσματος κατάστασης με ενσωματωμένη την μέτρηση

$$\hat{x}_k = \hat{x}_k^- + Ky \rightarrow \begin{bmatrix} \hat{\theta} \\ \hat{\theta}_b \end{bmatrix}_k = \begin{bmatrix} \hat{\theta} \\ \hat{\theta}_b \end{bmatrix}_k^- + \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} y$$

```
angle += K[0]*y;
bias += K[1]*y;
```

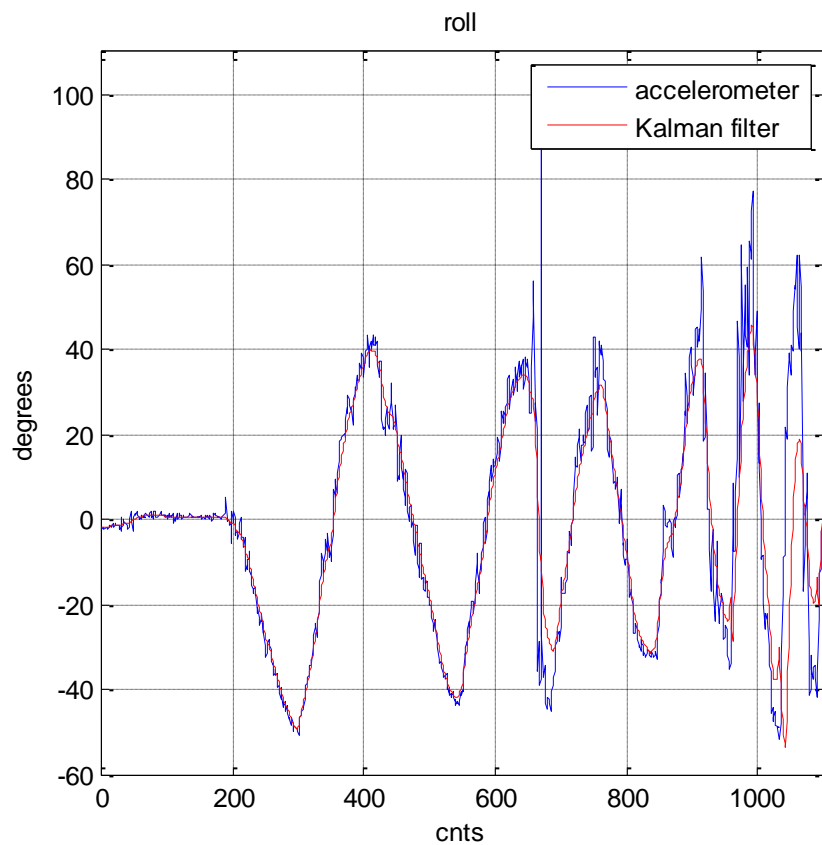
επαναυπολογισμός του πίνακα  $P$  λόγω της διόρθωσης

$$P_k = (I - KH)P_k^- \rightarrow$$

$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_k = \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_k^- \rightarrow$$

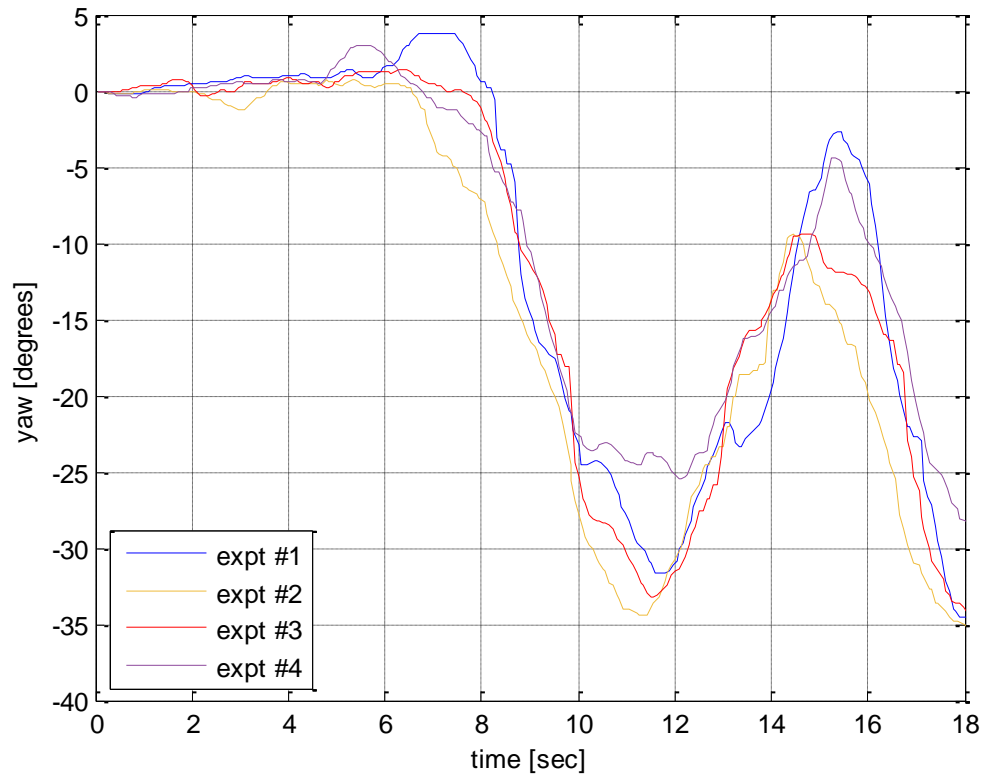
$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_k = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_k^- - \begin{bmatrix} K_0 P_{00} & K_0 P_{01} \\ K_1 P_{10} & K_1 P_{11} \end{bmatrix}_k^-$$

```
P[0][0] -= K[0]*P[0][0];
P[0][1] -= K[0]*P[0][1];
P[1][0] -= K[1]*P[0][0];
P[1][1] -= K[1]*P[0][1];
```



Εικόνα 9.8 Πειραματικά αποτελέσματα προσδιορισμού της γωνίας roll

Για την αξιολόγηση της μεθόδου κατά την διεξαγωγή τεσσάρων διαδοχικών πειραμάτων και συγκεκριμένα της διάσχισης του μονοπατιού της εικόνας 7.6 καταγράφηκε κάθε φορά η μεταβολή της γωνίας yaw μέσω του Bluetooth module HC-06 και της διεπαφής χρήστη που κατασκευάστηκε στο γραφικό περιβάλλον του λογισμικού Matlab. Η διαδρομή όπως αναπαριστά και το κάτωθι διάγραμμα περιλαμβάνει διαδοχικά: κίνηση ευθεία, δεξιά στροφή, αριστερή στροφή και τέλος δεξιά στροφή.



Εικόνα 9.9 Μεταβολή της γωνίας yaw του οχήματος κατά την διάσχιση μονοπατιού της εικόνας 7.6

## Κεφάλαιο

## 10

## 10. Συμπεράσματα - Προτάσεις για μελλοντική εξέλιξη

Η παρούσα εργασία αποτελεί το πρώτο βήμα για μια πιθανή συμμετοχή του Τ.Ε.Ι. Κρήτης στον διεθνή διαγωνισμό "Robot Field Event". Αντικείμενο μελέτης αποτέλεσε η εξ αρχής ανάπτυξη και η κατασκευή μιας ρομποτικής πλατφόρμας η οποία θα έχει την δυνατότητα να κινείται με άνεση σε αγρό και συγκεκριμένα να πλοηγείται αυτόνομα μέσω οπτικής πληροφορίας σε οργανωμένη καλλιέργεια σειρών (row crop). Ο σχεδιασμός και η κατασκευή του ρομποτικού οχήματος απαίτησε το συνδυασμό πολλών επιστημονικών τομέων: αυτόν της μηχανολογίας, της ηλεκτρονικής-ηλεκτρολογίας και της πληροφορικής.

Μέσα από την διεξαγωγή πειραματικών δοκιμασιών σε ανώμαλο έδαφος διαπιστώθηκε η ικανότητα του οχήματος να κινείται και να ανταπεξέρχεται με σχετική ευκολία στους εφικτούς ελιγμούς που δύναται να υλοποιήσει ένα τετράτροχο όχημα με διαφορική κίνηση παρουσιάζοντας παράλληλα ενεργειακή αυτονομία που υπερκαλύπτει τις απαιτήσεις του διαγωνισμού. Κλειδί για την επιτυχή κίνηση του οχήματος αποτέλεσε το σύστημα παθητικής ανάρτησης σε συνδυασμό με τον συγχρονισμένο έλεγχο των τεσσάρων κινητήριων τροχών μέσω PID ελεγκτών και της χρήσης του διαύλου I<sup>2</sup>C.

Η διάσχιση του μονοπατιού που σχηματίζουν τα φυτά χάρις στον ασαφή ελεγκτή πραγματοποιείται με ανεκτή ακρίβεια, ικανοποιητική ταχύτητα και παρουσιάζει επαναληψιμότητα κάτω από διάφορες συνθήκες φωτισμού και αέρα με μοναδικό αισθητήριο άντλησης δεδομένων για τον χώρο κίνησης του οχήματος την web camera. Ο μικροϋπολογιστής Odroid-XU3 και οι βιβλιοθήκες της OpenCV αποδείχτηκαν από πλευράς ταχύτητας επεξεργασίας αντάξιες των προσδοκιών για την συγκεκριμένη εφαρμογή.

Επετεύχθη επίσης η ασύρματη επικοινωνία και η απομακρυσμένη διαχείριση του Odroid-XU3 από Η/Υ χωρίς την ανάγκη ύπαρξης σύνδεσης στο διαδίκτυο καθώς και η αποστολή δεδομένων στο workspace του Matlab κατά την διάρκεια της κίνησης του οχήματος μέσω του Bluetooth module HC-06.

Η εγκυρότητα της μεθόδου για την εύρεση της γωνίας προσανατολισμού του οχήματος (heading) που περιγράφηκε στο Κεφάλαιο 10 για τις ανάγκες της παρούσας εφαρμογής κρίνεται επαρκής.

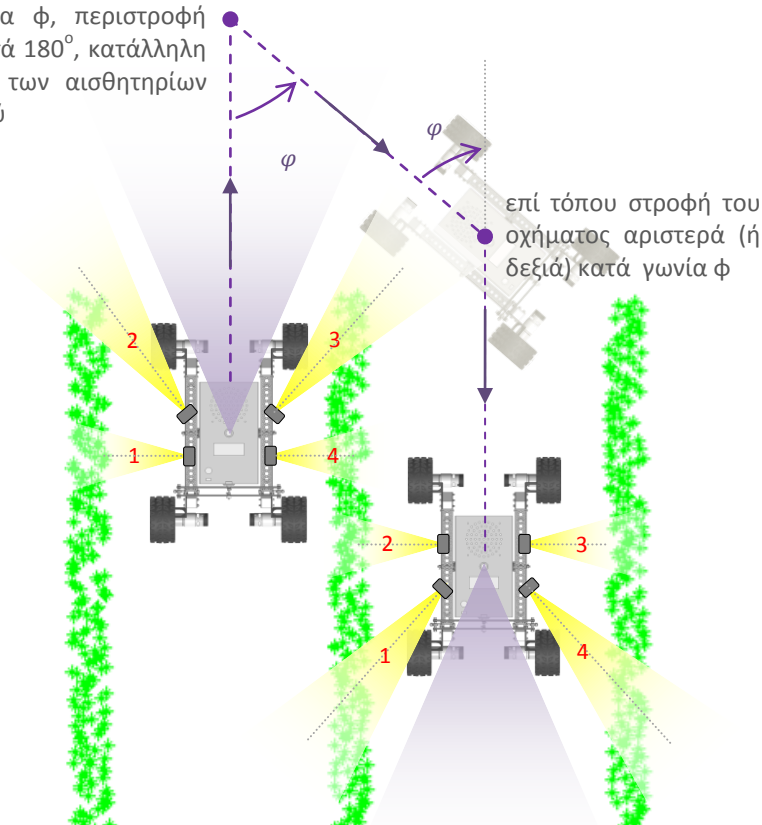
Για περαιτέρω αύξηση της αξιοπιστίας και βελτίωση της συμπεριφοράς του οχήματος ώστε αυτό να μην παρεκκλίνει της πορείας του, διατηρώντας παράλληλα το κόστος υλοποίησης χαμηλά, προτείνεται αυτό να εφοδιαστεί περιμετρικά με επαρκή αριθμό αισθητήρων ηχοβολισμού (sonar) τα οποία θα είναι τοποθετημένα σε περιστρεφόμενες βάσεις (μέσω σερβομηχανισμών) για την λήψη μετρήσεων στις επιθυμητές κατευθύνσεις. Τα αισθητήρια υπερήχων θα πρέπει να είναι συμβατά με το πρωτόκολλο I<sup>2</sup>C ώστε να ενταχθούν και αυτά στον δίαυλο επικοινωνίας και η δέσμη του σήματος μέτρησης τους να παρουσιάζει μικρό γωνιακό εύρος (κατευθυντικότητα). Για την περιστροφή τους ο μικρο-σερβοκινητήρας Batan2122 που χρησιμοποιήθηκε και στην

παρούσα εργασία για τον μηχανισμό περιστροφής και κλίσης της κάμερας πληροί τις προϋποθέσεις. Τα αισθητήρια αυτά προτείνεται να τοποθετηθούν στους πλευρικούς βραχίονες του σασί (rocker arms) κοντά στον κύριο άξονα περιστροφής με τους σερβοκινητήρες εντός της δοκού. Επιλέγοντας τα sonar να μην έχουν σταθερό προσανατολισμό διατηρείται η δυνατότητα στο όχημα, με ταυτόχρονη περιστροφή της κάμερας κατά  $180^\circ$ , να πραγματοποιεί τους ελάχιστους δυνατούς ελιγμούς κατά την είσοδο του στην επόμενη σειρά καλλιέργειας. Επίσης κατά την διεξαγωγή της 2<sup>ης</sup> δραστηριότητας του διαγωνισμού και κατά την αναζήτηση του επιθυμητού διαδρόμου στον οποίο το όχημα θα εισέλθει σύμφωνα με το σχέδιο πορείας το οποίο έχει δοθεί στην αφετηρία, οι πλευρικοί αισθητήρες υπερήχων αν ευθυγραμμιστούν με το διαμήκη άξονα του οχήματος αυξάνουν την διακριτική ικανότητα και δίνουν την δυνατότητα να ανιχνεύεται η θέση των φυτών με μεγαλύτερη ακρίβεια. Στην ίδια επίσης δραστηριότητα ενδέχεται να απουσιάζουν τα φυτά για μια συγκεκριμένη απόσταση από την μια ή την άλλη πλευρά του διαδρόμου οπότε η αυτόνομη πλοήγηση του οχήματος θα πραγματοποιείται σε σημαντικό βαθμό με τα δεδομένα που λαμβάνονται από τους αισθητήρες υπερήχων.



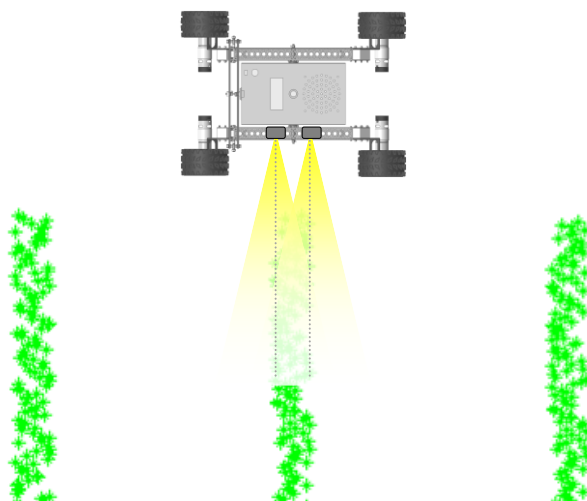
Εικόνα 10.1 Sonar τοποθετημένο σε ειδική περιστρεφόμενη βάση

επί τόπου στροφή του οχήματος δεξιά (ή αριστερά) κατά γωνία  $\phi$ , περιστροφή κάμερας κατά  $180^\circ$ , κατάλληλη περιστροφή των αισθητηρίων ηχοβολισμού



επί τόπου στροφή του οχήματος αριστερά (ή δεξιά) κατά γωνία  $\phi$

Εικόνα 10.2 Απαιτούμενες ενέργειες και ελιγμοί κατά την είσοδο του οχήματος στην επόμενη σειρά.



**Εικόνα 10.3** Διατάξη πλευρικών αισθητήρων ηχοβολισμού κατά την αναζήτηση της επιθυμητής σειράς

Στην παρούσα εφαρμογή ο μικροϋπολογιστής Odroid-XU3 αποστέλλει μόνο δεδομένα σειριακά στην αναπτυξιακή πλακέτα Arduino Mega 2560. Η ένταξη του Odroid-XU3 στον δίαυλο I<sup>2</sup>C μέσω των αντίστοιχων ακροδεκτών της υποδοχής GPIO θα λύσει το πρόβλημα της λήψης δεδομένων και θα αυξήσει σημαντικά την μεταξύ τους ταχύτητα επικοινωνίας. Τα δεδομένα που αποστέλλονται κατά την διάρκεια κίνησης του οχήματος μέσω του Bluetooth module μπορούν να στέλνονται έτσι απευθείας στο Odroid-XU3 για περαιτέρω επεξεργασία και εκσφαλμάτωση του πηγαίου κώδικα. Πληροφορίες για την υλοποίηση της εφαρμογής δίνονται στην ιστοσελίδα:

<http://odroid.com/dokuwiki/doku.php?id=en:odroid-xu3>

Για την εξοικονόμηση χώρου στο κουτί των ηλεκτρονικών ελέγχου και την τακτοποίηση των καλωδιώσεων συνίσταται η ενοποίηση των τυπωμένων κυκλωμάτων της διανομής της ηλεκτρικής ενέργειας και των τεσσάρων κυκλωμάτων ρύθμισης της τάσης: εκείνου των 12V, των 5V (2 συνολικά: ένα γενικής χρήσης και ένα για την τροφοδοσία των 2 μικρο-σερβοκινητήρων Bata2122) και εκείνου των 7V για την τροφοδοσία του Arduino Mega 2560.

Σημαντικό πρόβλημα αποτελεί επίσης η ύπαρξη και η συντήρηση ενός χώρου διεξαγωγής πειραμάτων. Για τον λόγο αυτό προτείνεται η δημιουργία ενός εικονικού περιβάλλοντος μέσω προγραμμάτων εξομοίωσης (π.χ. V-Rep της Coppelia Robotics σε συνεργασία με Matlab/Simulink) ώστε να πραγματοποιούνται σε πρώτο στάδιο οι δοκιμές των ελεγκτών κίνησης του οχήματος. Πιθανών χρήσιμοι σύνδεσμοι για το εν λόγω εγχείρημα:

<http://www.car.upm-csic.es/searfs/>

<http://jpdelaCroix.com/software/simiam.html>

[http://petercorke.com/Robotics\\_Toolbox.html](http://petercorke.com/Robotics_Toolbox.html)

# Βιβλιογραφία

## Ελληνόγλωσση

### Άρθρα

- [1] Καρυδάς Χ., Συλλαίος Ν., (10/2000). "Γεωργία Ακριβείας: Περιγραφή της μεθόδου - Υφιστάμενη κατάσταση και προοπτικές", 2<sup>ο</sup> συνέδριο Πληροφοριακά συστήματα στον Αγροτικό τομέα της Ελληνικής Εταιρείας Επιχειρησιακών Ερευνών, Πρακτικά σελ. 134-146
- [2] Τρέσσος Κ.Δ., Ανδρέου Ι.Α., Blackmore S. , Φουντάς Σ., Γέμτος Θ.Α. (18-20/10/2007). "Σχεδιασμός και υλοποίηση ρομποτικής πλατφόρμας χαμηλού κόστους για την υποστήριξη γεωργικών πρακτικών", Εταιρεία Γεωργικών Μηχανικών Ελλάδος - Πρακτικά 5<sup>ου</sup> Εθνικού συνεδρίου γεωργικής μηχανικής σελ. 898-905

### Σημειώσεις και διδακτικά συγγράμματα

- [1] Συστήματα αυτομάτου ελέγχου II, Δρ. Σφακιωτάκης Μιχαήλ, 01/2011, ΤΕΙ Κρήτης - Τμήμα Ηλεκτρολογίας
- [2] Μικροελεγκτές - Διαλέξεις θεωρίας, Δρ. Σφακιωτάκης Μιχαήλ, 01/2011, ΤΕΙ Κρήτης
- [3] Αισθητήρια Ρομποτικής, Δρ. Σφακιωτάκης Μιχαήλ, ΤΕΙ Κρήτης - Τμήμα Μηχανολογίας
- [4] Ασαφή Συστήματα - Θεωρία, και Εργαστηριακές Ασκήσεις, Παπαδάκης Στυλιανός, Αδαμίδης Παναγιώτης, 05/2004 ΤΕΙ Θεσσαλονίκης - Τμήμα Πληροφορικής
- [5] Ευφυής έλεγχος, Θεωρία και εφαρμογές, Διδακτικές σημειώσεις τμήματος Πληροφορικής και Επικοινωνιών, Δρ. Βολογιαννίδης Σταύρος, ΤΕΙ Σερρών - Τομέας Αρχιτεκτονικής Υπολογιστικών και Βιομηχανικών Εφαρμογών

## Ξενόγλωσση

### Άρθρα

- [1] Petrella R., Tursini M., Peretti L. & Zigliotto M. (2007). "Speed measurement algorithms for low-resolution incremental encoder equipped drives: a comparative analysis". International Aegean conference on electrical machines and power electronics, pp. 780-787. doi:10.1109/acemp.2007.4510607
- [2] Ali Zogheib, (2008). "Autonomous navigation tool for teal & virtual field robots", Mid Sweden University, Sundsvall, Sweden, 1<sup>st</sup> International Conference on Machine Control & Guidance
- [3] T. Carlone, (2013). "Investigating suspension lift-off of skid-steer rovers with passive differential suspension". Masters of Science in Robotics, Carnegie Mellon University, Pittsburgh, Pennsylvania
- [4] Foglia M. and Reina G. (2008), "Locomotion performance evaluation of an all-terrain rover". International Journal of Mechanics and Control, Vol. 09 pp. 13-25, No. 02
- [5] Reina G. and Foglia M. (2013), "On the mobility of all-terrain rovers". Industrial Robot, Vol. 40 Iss: 2, pp. 121 - 131. doi:10.1108/01439911311297720
- [6] J. Amato, J. Anderson and T. Carlone, (04/2011). "Oryx 2.0: A planetary exploration mobility platform". Project report for the Degree of Bachelor of Science, Worcester Polytechnic Institute
- [7] Yuanjie Wang, Fuzeng Yang, 2, Yu Zhou, Guanting Pan, Jinyi He, Yubin Lan, (9/2013). "A riding fuzzy control system for a mountain agricultural robot". College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling, Shaanxi, P.R.China, International journal on smart sensing and intelligent systems, Vol. 6, No. 40

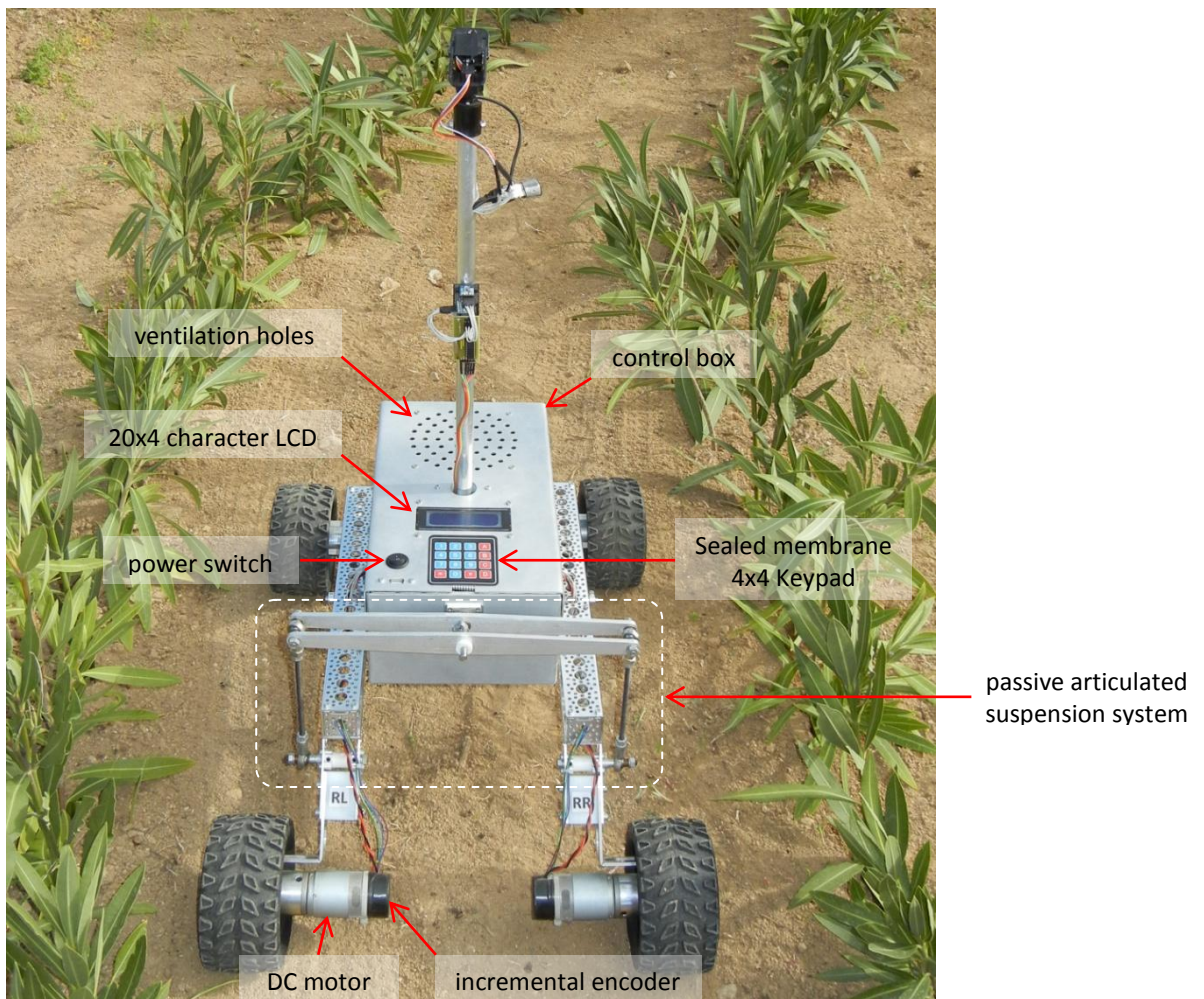
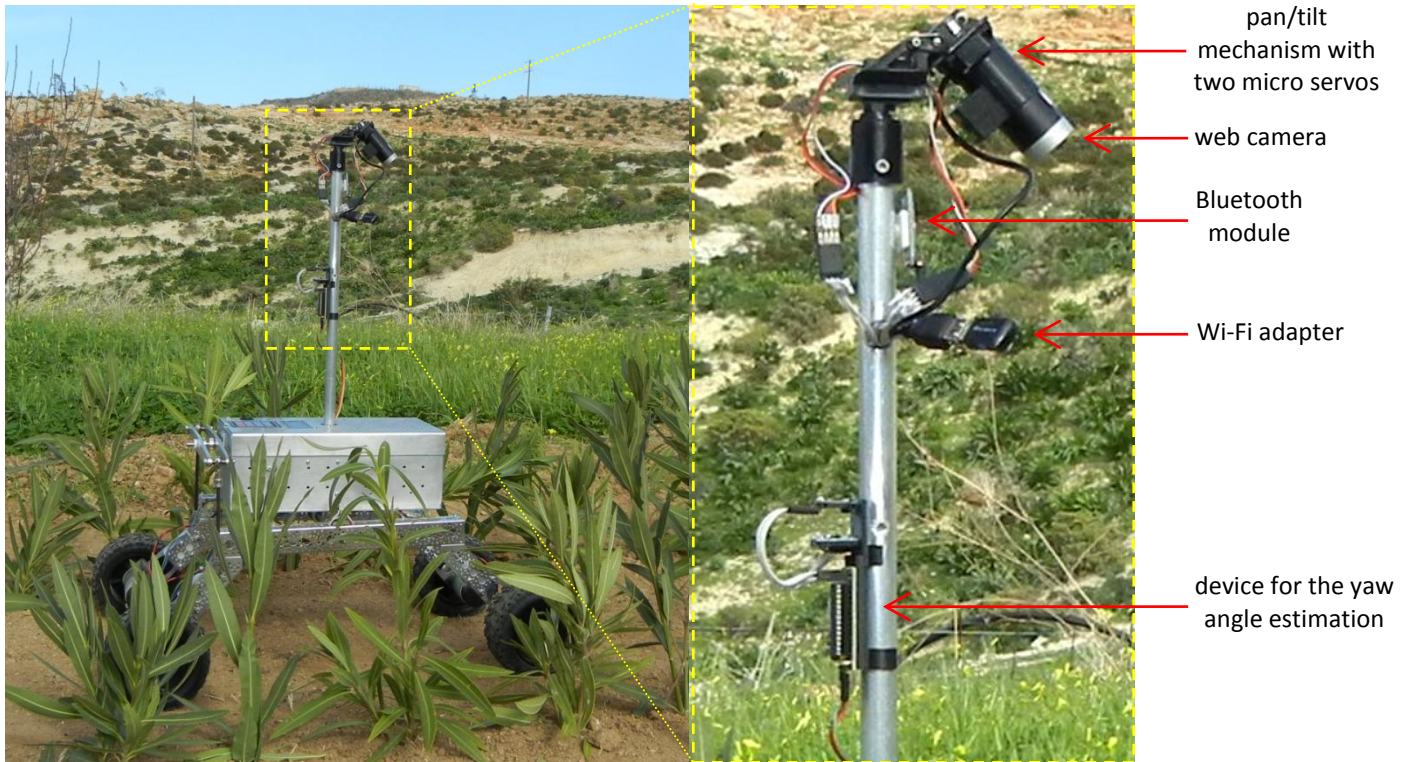
- [8] D.M. Woebbecke, G. E. Meyer, K. Von Bargen & D.A. Mortensen, (1995). "Color indices for weed identification under various soil, residue and lighting conditions". Transactions of the ASAE 38(1), 259–269. doi:10.13031/2013.27838
- [9] Meyer G.E., Hindman T.W., Lakshmi K.,(1999) "Machine vision detection parameters for plant species identification". Precision Agriculture and Biological Quality, Proceedings of SPIE. vol. 3543, Bellingham,WA, pp. 327–335. doi:10.1117/12.336896
- [10] Meyer G.E., Neto J.C. (2008). "Verification of color vegetation indices for automated crop imaging applications". Computers and Electronics in Agriculture 63(2), 282-293. doi:10.1016/j.compag.2008.03.009
- [11] Burgos-Artizzu X.P., Ribeiro A., Guijarro M. & Pajares G., (2011). "Real-time image processing for crop/weed discrimination in maize fields". Computers and Electronics in Agriculture 75(2), 337-346. doi:10.1016/j.compag.2010.12.011
- [12] Kataoka T., Kaneko T., Okamoto H. & Hata S., (2003). "Crop growth estimation system using machine vision". Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003). doi:10.1109/aim.2003.1225492
- [13] Hague T., Tillet N.D., Wheeler H., (2006). "Automated crop and weed monitoring in widely spaced cereals". Precision Agriculture, 7(1), 21-32. doi:10.1007/s11119-005-6787-1.
- [14] Guijarro M., Pajares G., Riomoros I., Herrera P., Burgos-Artizzu X. & Ribeiro A., (2011). "Automatic segmentation of relevant textures in agricultural images". Computers and Electronics in Agriculture 75(1), 75-83. doi:10.1016/j.compag.2010.09.013
- [15] Guerrero J., Pajares G., Montalvo M., Romeo J., Guijarro M., (2012). "Support Vector Machines for crop/weeds identification in maize fields". Expert Systems with Applications Vol. 39, Issue 12, pp. 11149–11155. doi:10.1016/j.eswa.2012.03.040
- [16] Woebbecke D.M., Meyer G. E., Von Bargen K. and Mortensen D.A. (1993). "Plant species identification, size, and enumeration using machine vision techniques on near-binary images", Proc. SPIE 1836, Optics in Agriculture and Forestry, 208. doi: 10.1117/12.144030
- [17] Keun Ha Choi, Sang Kwon Han, Sang Hoon Han, Kwang-Ho Park, Kyung-Soo Kim, Soohyun Kim, (2015). "Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields". Computers and Electronics in Agriculture, Vol. 113, pp. 266-274. doi:10.1016/j.compag.2015.02.014
- [18] Faragher R. (09/2012). "Understanding the Basis of the Kalman Filter via a Simple and Intuitive Derivation", IEEE Signal processing magazine 29(5) pp. 128-132. doi:10.1109/msp.2012.2203621
- [19] AJ Neal, (01/201). "Tips for selecting DC Motors for your mobile Robot"
- [20] Mark Pedley (03/2013). "Tilt sensing using a three-axis accelerometer", Freescale Semiconductor, Application note, Document Number: AN3461, Rev. 6
- [21] Talat Ozyagcilar, (11/2015). "Implementing a tilt-compensated eCompass using accelerometer and magnetometer sensors", Freescale Semiconductor, Application note, Document Number: AN4248, Rev. 4.0
- [22] Vadym Grygorenko, "Sensing - Magnetic compass with tilt compensation", AN2272
- [23] "Using LSM303DLH for a tilt compensated electronic compass". 2010, STMicroelectronics, Application note: AN3192
- [24] Jean-Marc Irazabal, Steve Blozis, (24/03/2003). "I<sup>2</sup>C Manual", Philips Semiconductors, Application note: AN10216-01
- [25] Onkar Nitin Raut, (2011). "Shape recognition for plane closed curves using error model of an elliptical fit and Fourier descriptors", Dissertation for the degree of Master of Science, University of North Carolina, Charlotte



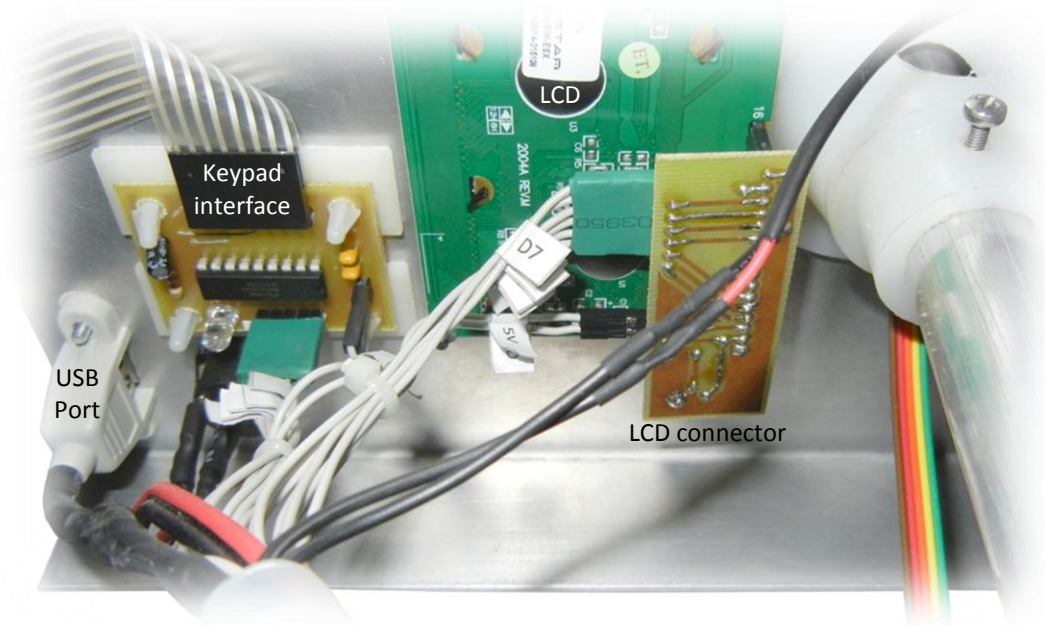
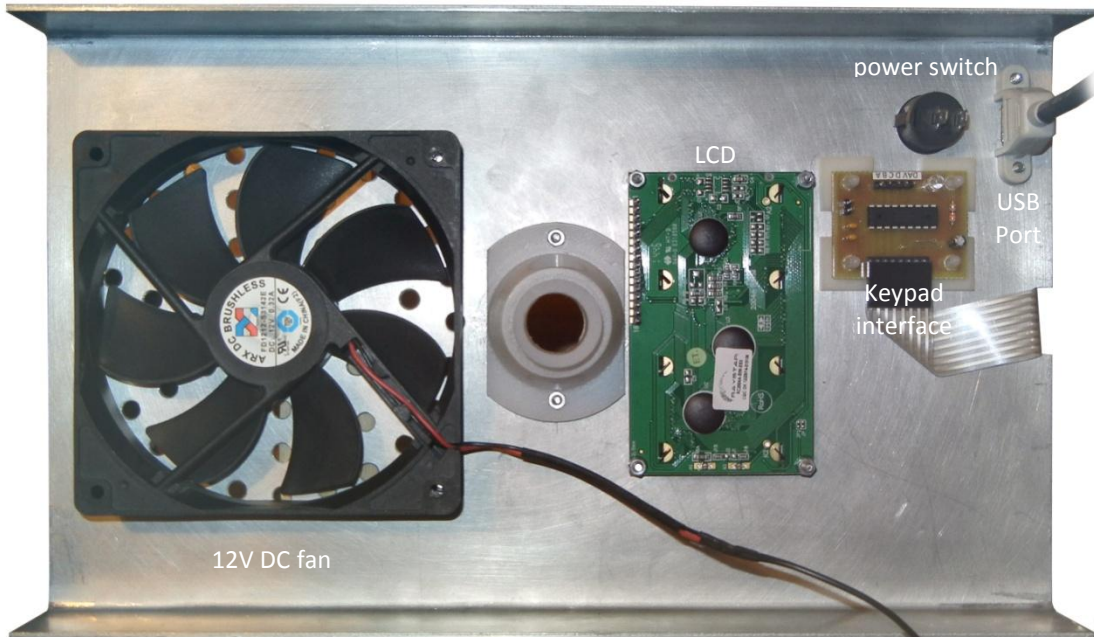
## Ιστοσελίδες

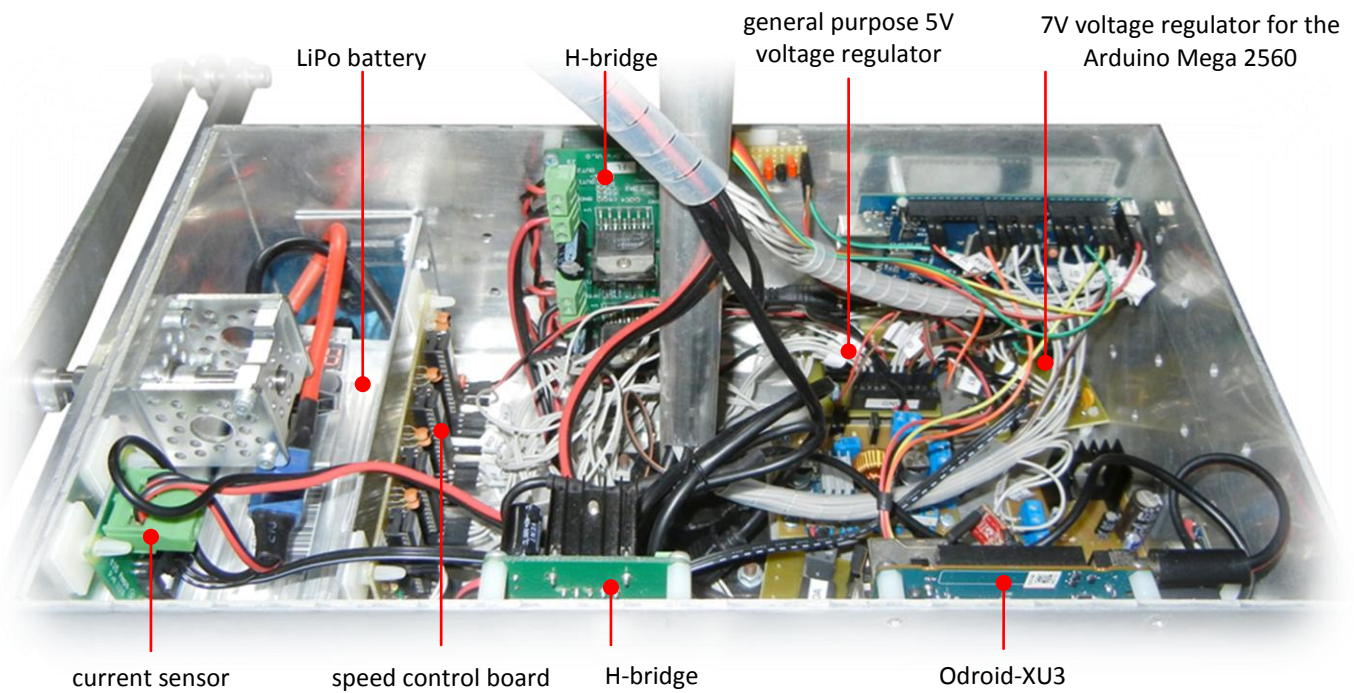
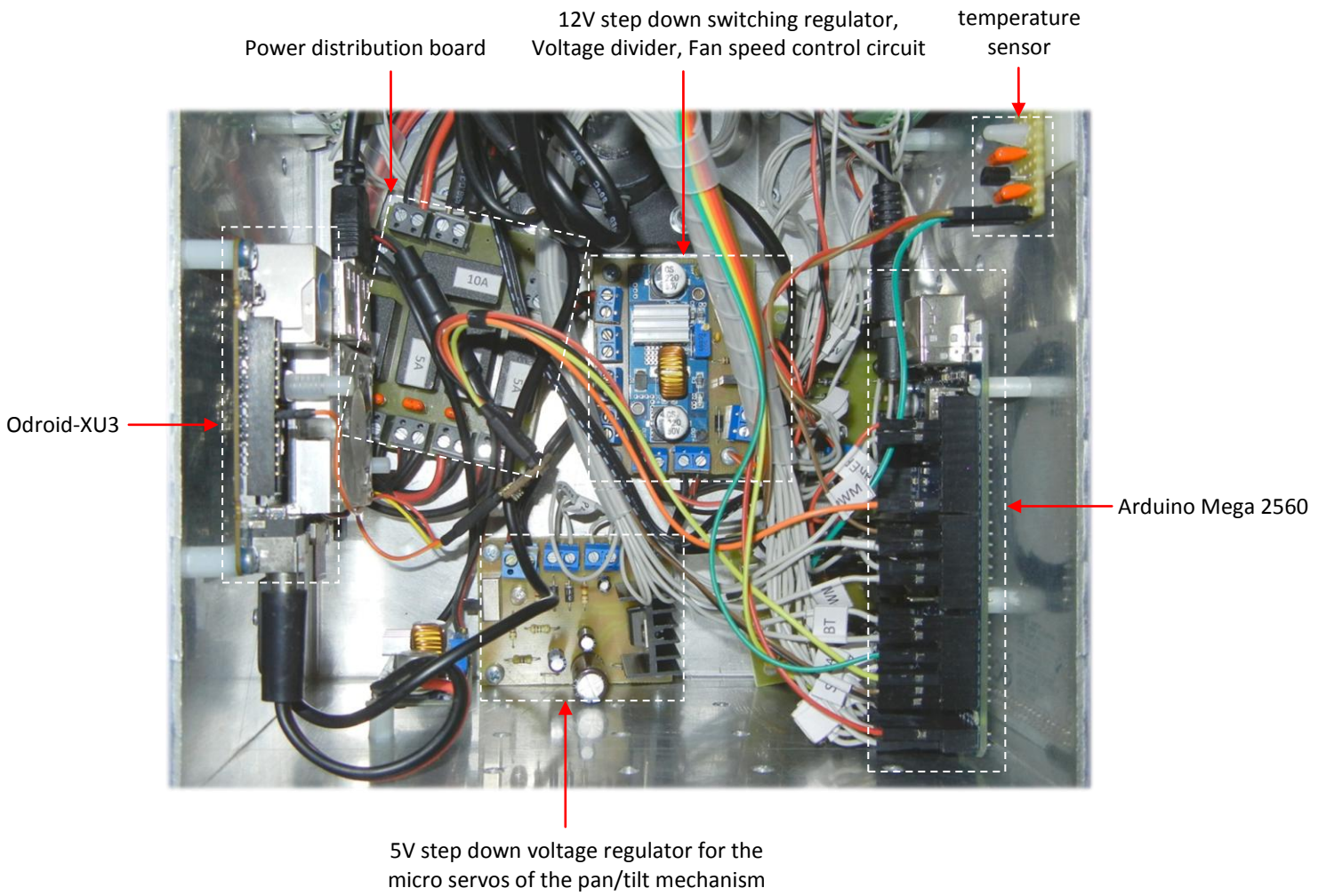
- [1] *Drive Motor Sizing Tutorial* by Coleman Benson  
<http://www.robotshop.com/blog/en/drive-motor-sizing-tutorial-3661>
- [2] *MM74C922N-based encoded matrix keypad*  
<http://embedded-lab.com/blog/mm74c922n-based-encoded-matrix-keypad/>
- [3] *The surface mobility platform*  
[http://www.gearseds.com/surface\\_mobility\\_platform.html](http://www.gearseds.com/surface_mobility_platform.html)
- [4] *A practical approach to Kalman filter and how to implement it*  
<http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it/comment-page-1/>
- [5] *Savitzky–Golay filter*  
[https://en.wikipedia.org/wiki/Savitzky%E2%80%93Golay\\_filter](https://en.wikipedia.org/wiki/Savitzky%E2%80%93Golay_filter)
- [6] *Otsu Thresholding*  
<http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>
- [7] *Accelerometer & Gyro Tutorial* by Gadget Gangste  
<http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/step1/The-Accelerometer/>
- [8] *Install OpenCV 2.4.10 and use it in MSVC 2013*  
<https://drthitirat.wordpress.com/2014/12/26/install-opencv-2-4-10-and-use-it-in-msvc-2013-and-qt-5-4-0-of-windows-8-1-64-bit-os-x64-based-processor/>
- [9] *How to build applications with OpenCV inside the Microsoft Visual Studio*  
[http://docs.opencv.org/2.4/doc/tutorials/introduction/windows\\_visual\\_studio\\_Openv/windows\\_visual\\_studio\\_Opencv.html](http://docs.opencv.org/2.4/doc/tutorials/introduction/windows_visual_studio_Openv/windows_visual_studio_Opencv.html)

## Παράρτημα Α - Φωτογραφίες οχήματος



Bottom view of the cover



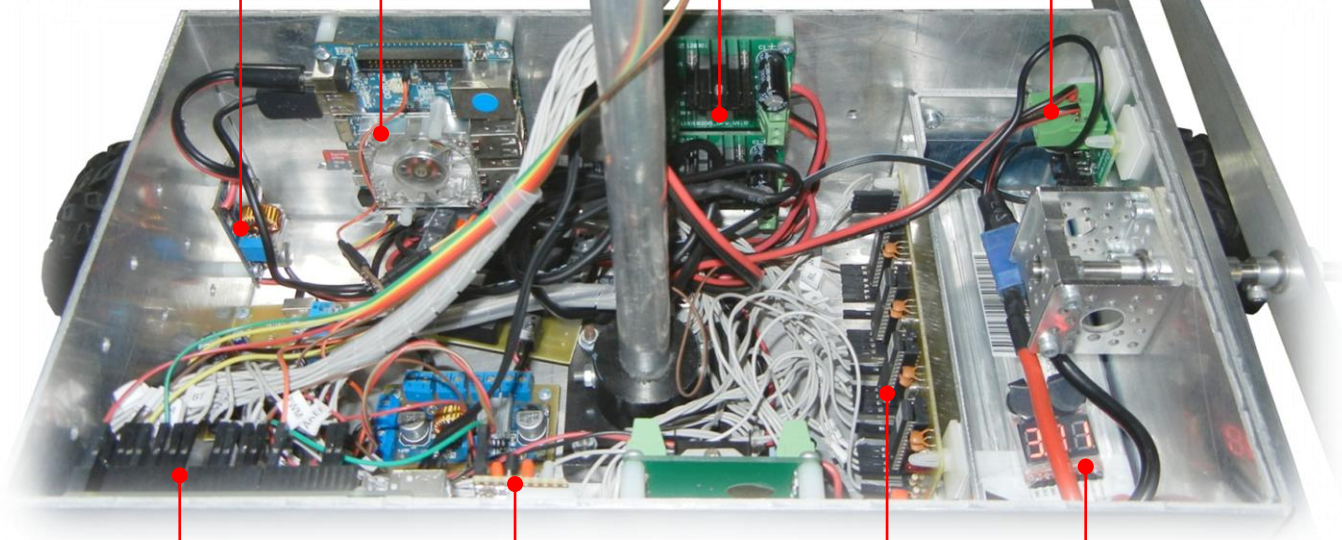


5V step down switching regulator for the Odroid-XU3

Odroid-XU3

H-bridge

current sensor



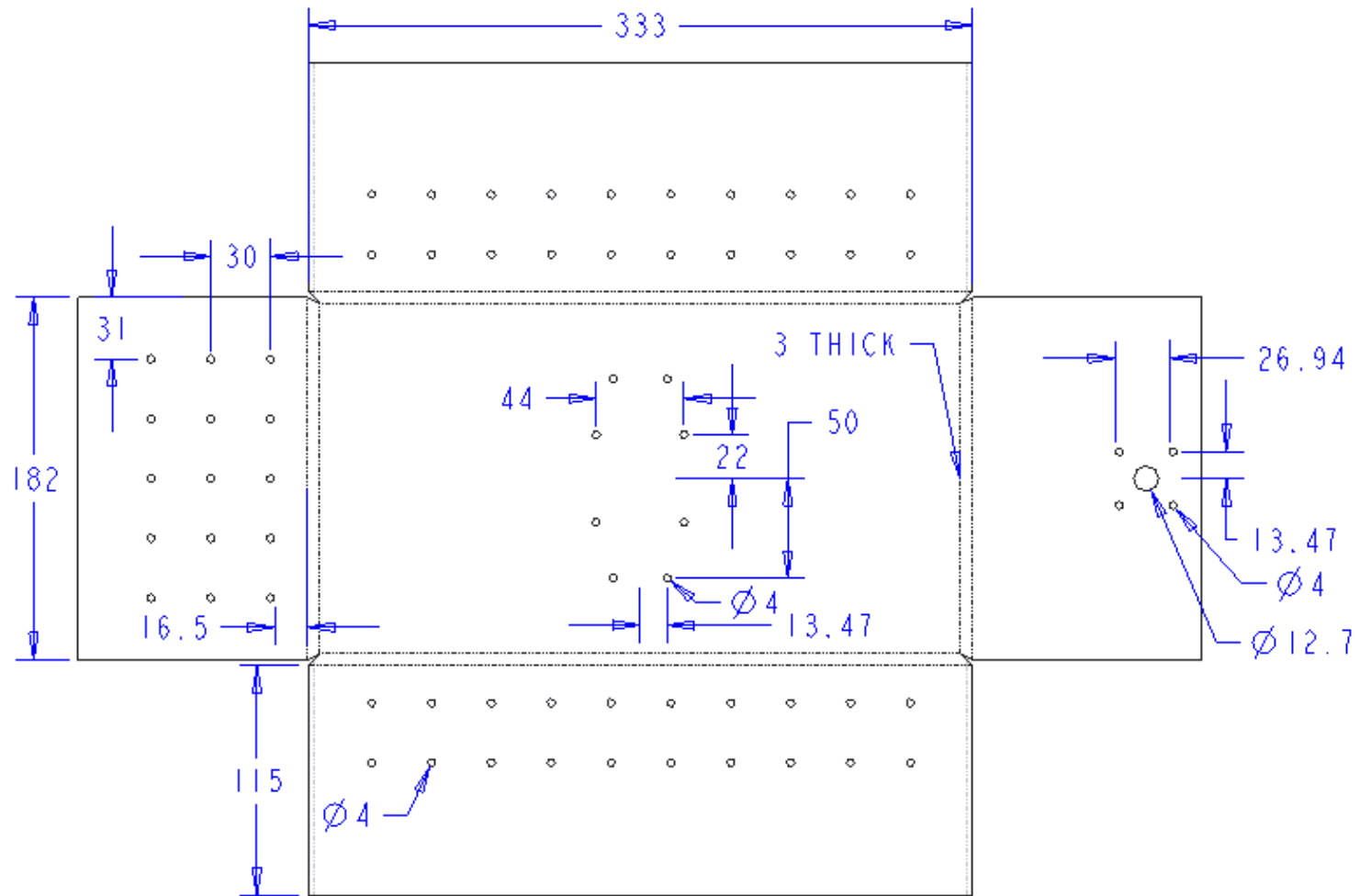
Arduino Mega 2560

temperature sensor

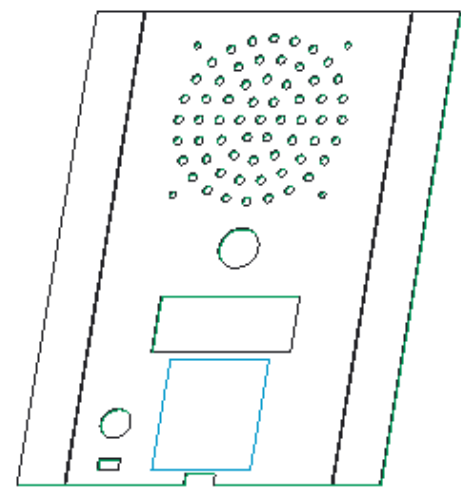
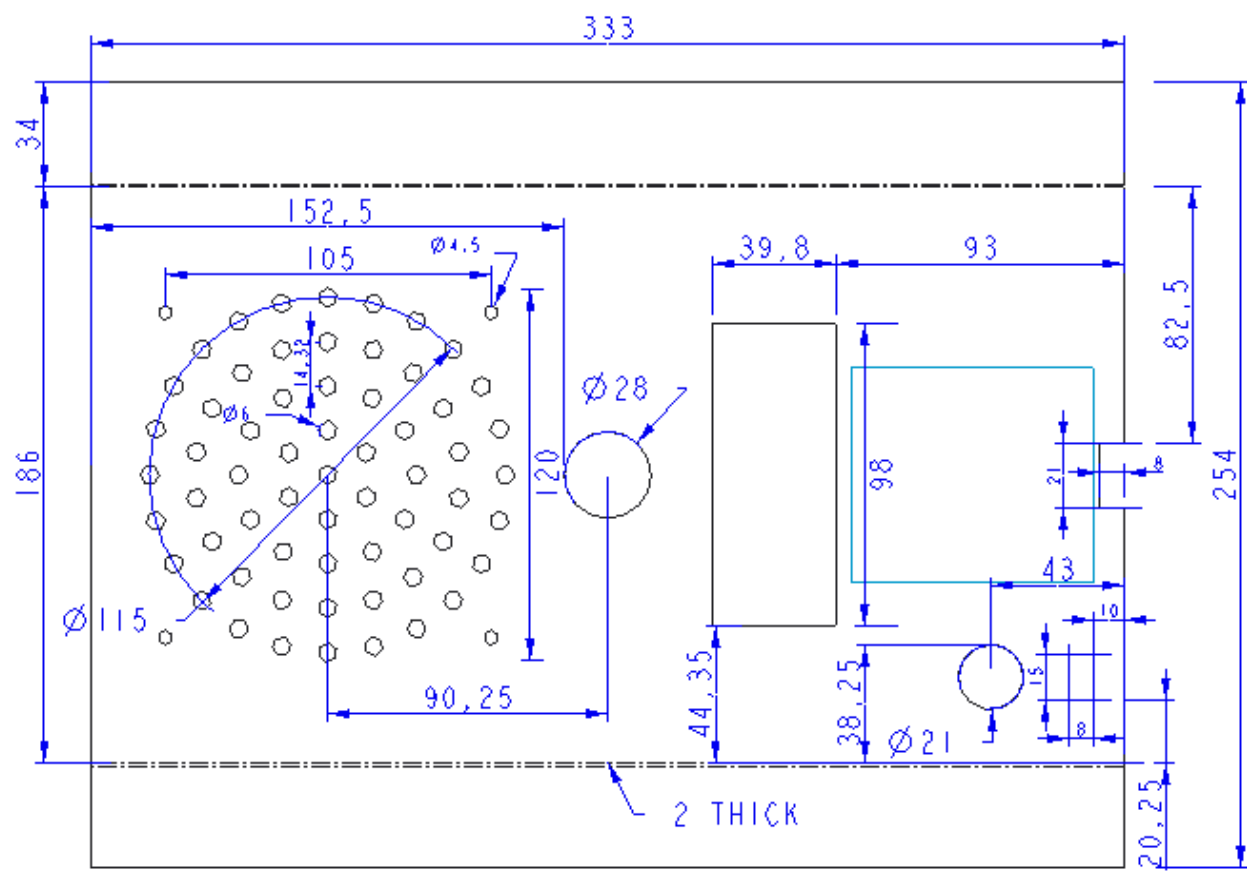
speed control board

LiPo battery tester, low voltage buzzer alarm

## Παράρτημα Β - Μηχανολογικά Σχέδια

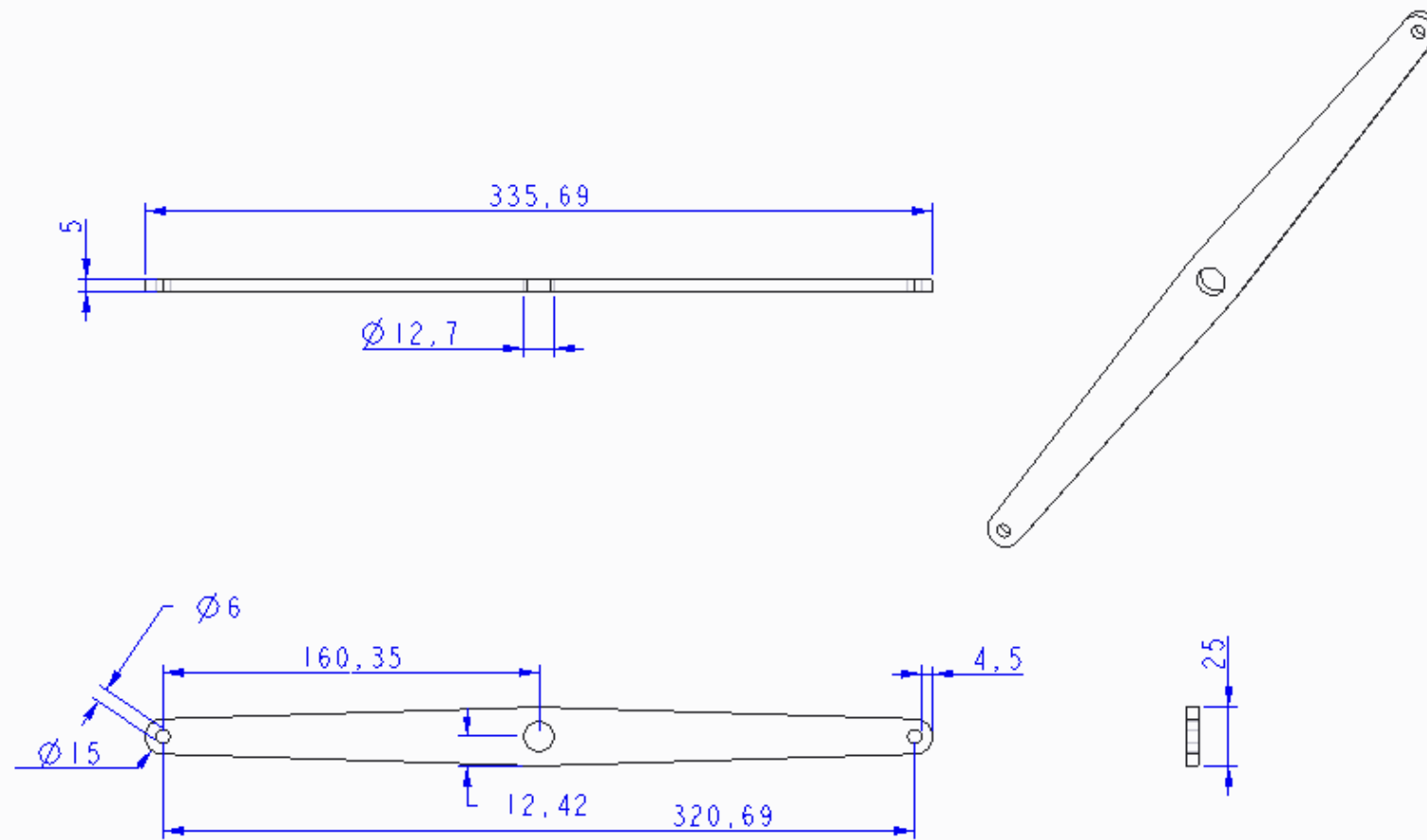


Scale 0.35	MSc thesis	Rover for the Robot Field Event
	Model Name	Control Box
Quantity 1	Designer	Kalykakis Emmanouil
	Maker	Makrakis Antonios



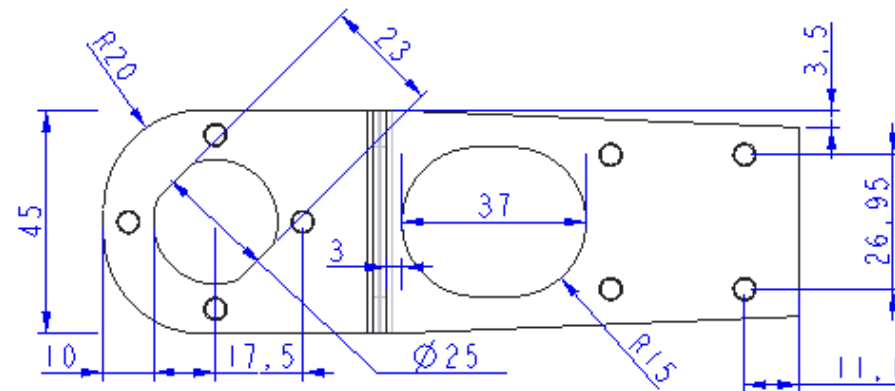
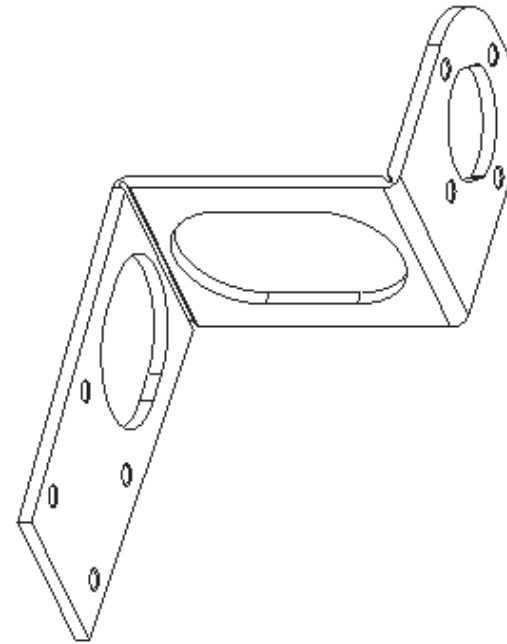
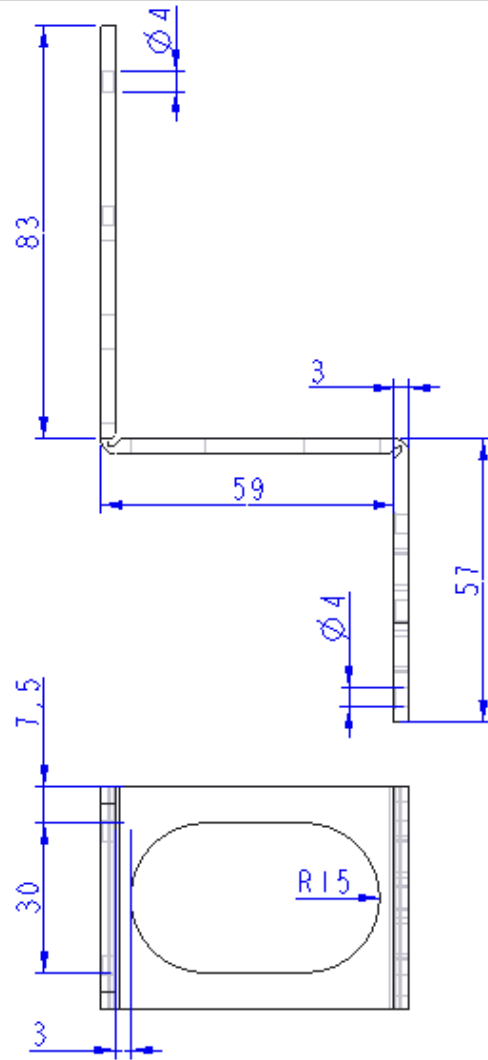
SCALE 0,250

Scale 0.5	MSc thesis	Rover for the Field Robot Event
	Model name	Lid of the Control Box
Quantity 1	Designer	Kalykakis Emmanouil
	Maker	Mountakis Nikolaos

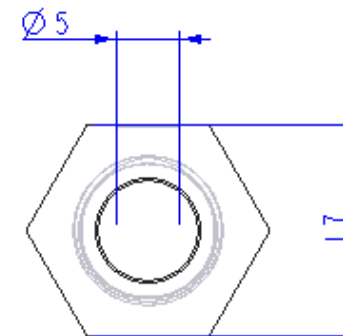
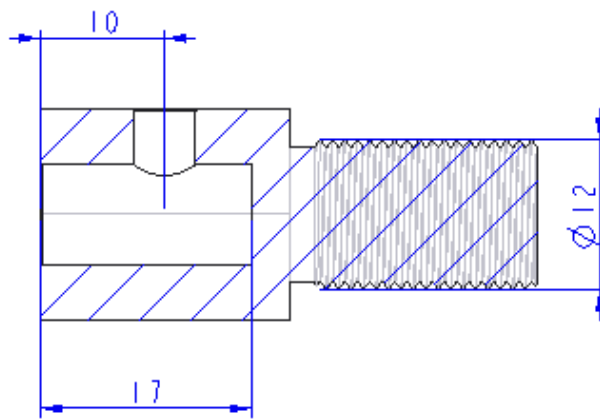
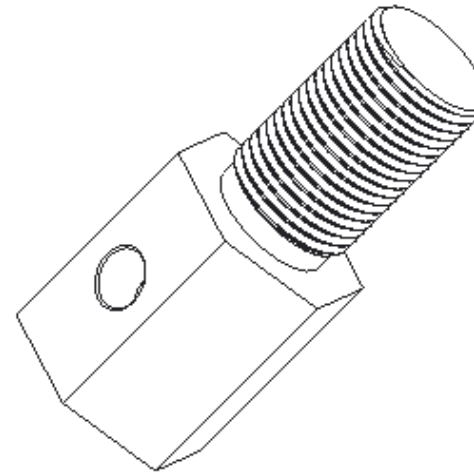
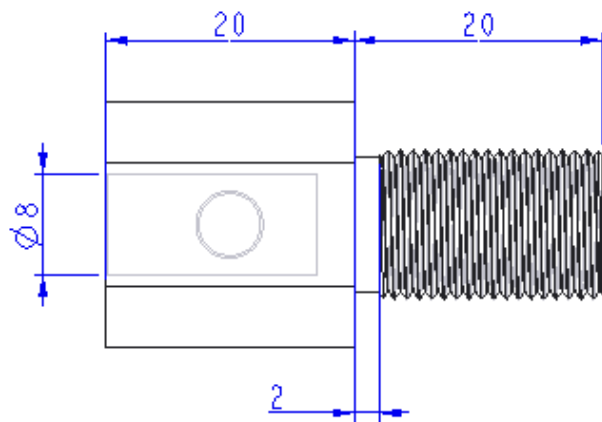


Scale 0.4	MSc thesis	Rover for the Robot Field Event
	Model name	Differential bar
Quantity 2	Designer	Kalykakis Emmanouil
	Maker	Mountakis Nikolaos

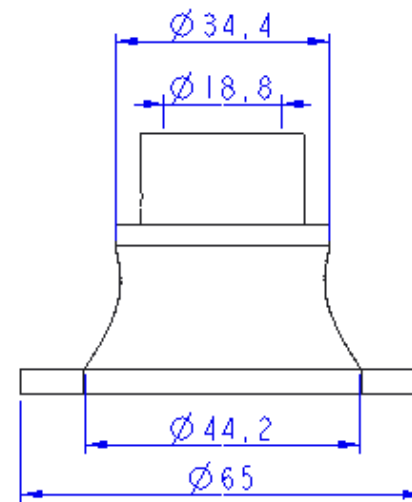
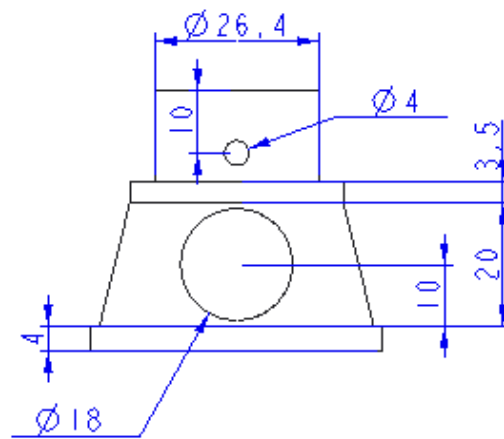
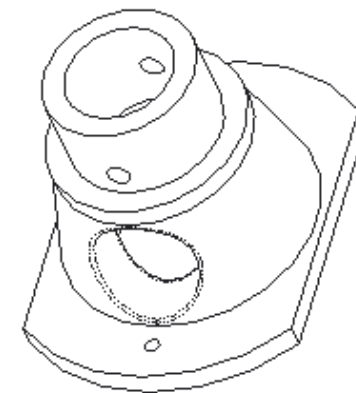
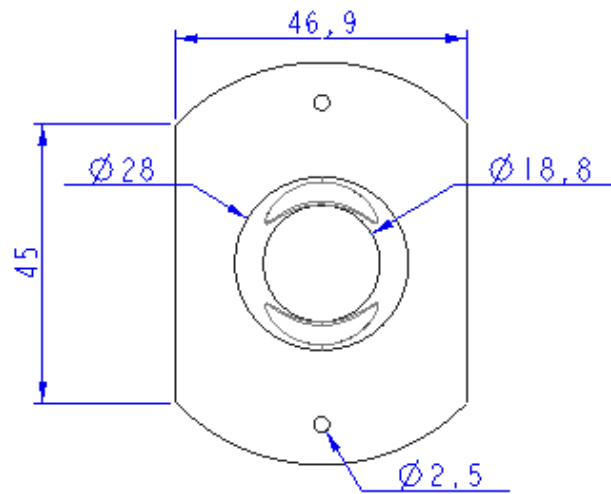




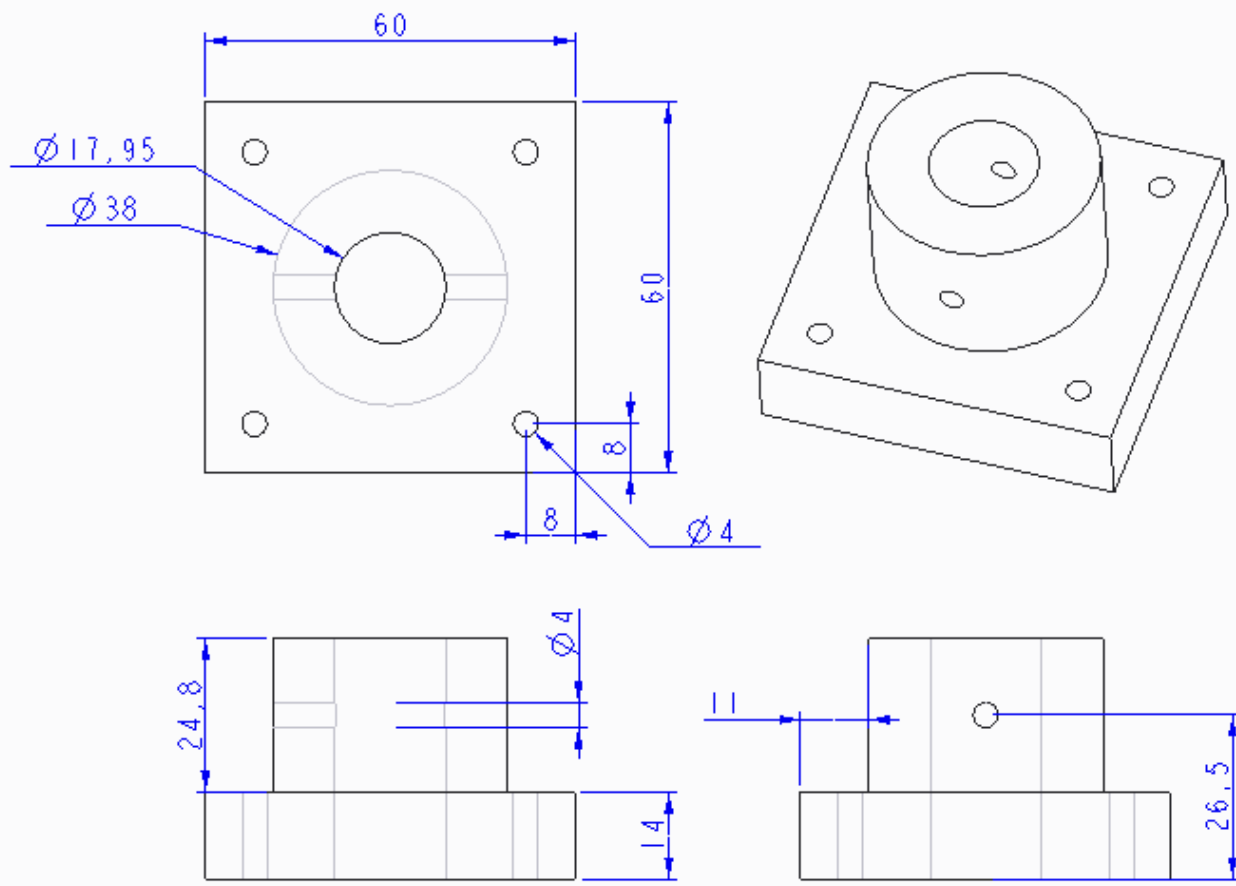
Scale	MSc thesis	Rover for the Field Robot Event
0.8	Model Name	Motor mount
Quantity	Designer	Kalykakis Emmanouil
	4	Maker



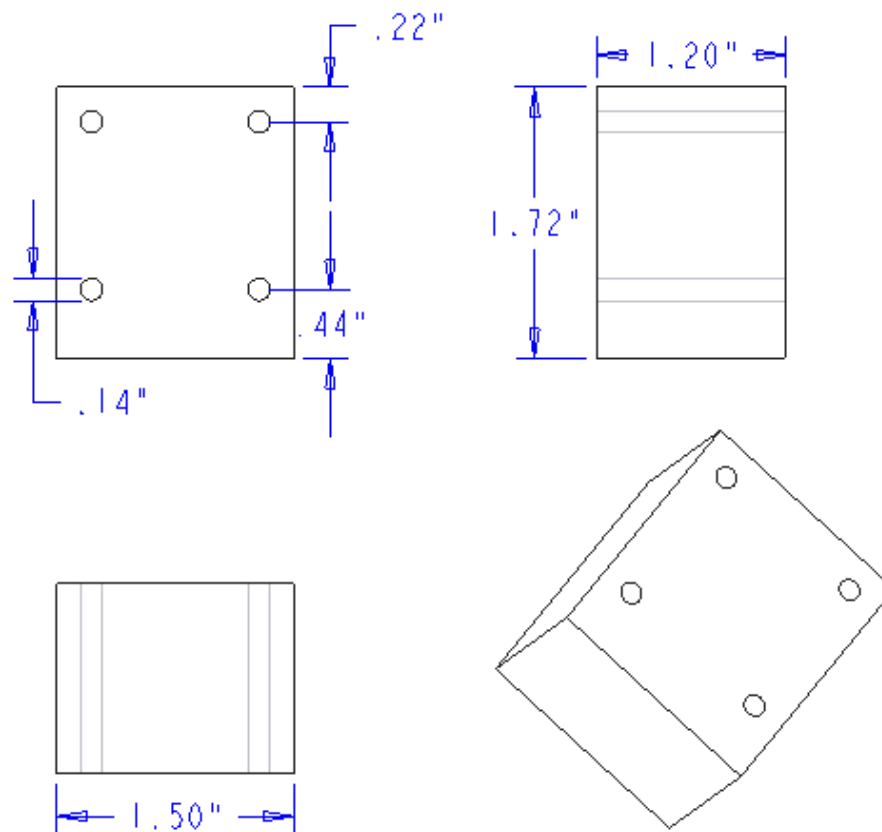
Scale 2:1	MSc thesis	Rover for the Robot Field Event
	Model name	Drive rim adapter
Quantity 4	Designer	Kalykakis Emmanouil
	Maker	Mountakis Nikolaos



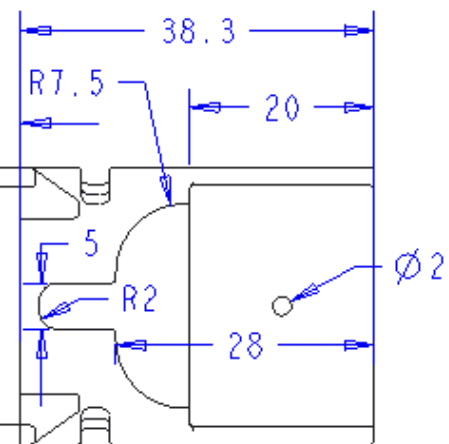
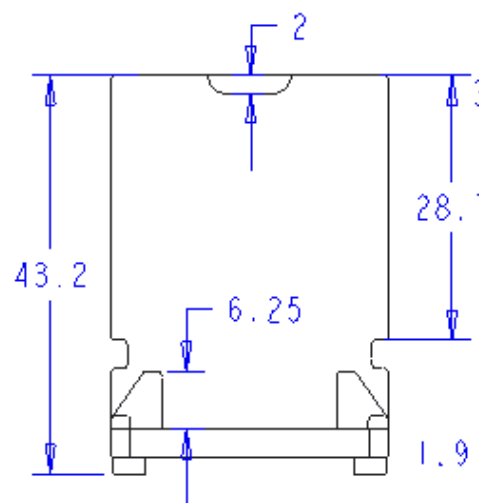
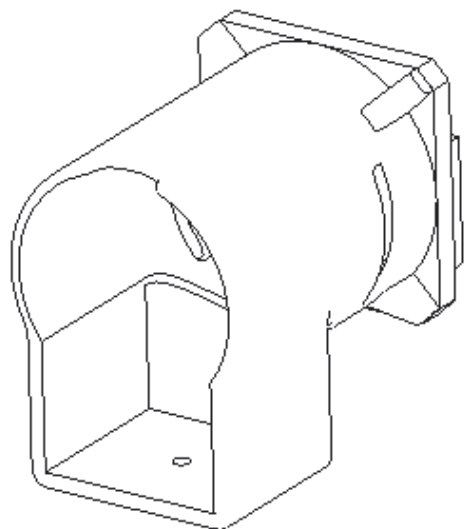
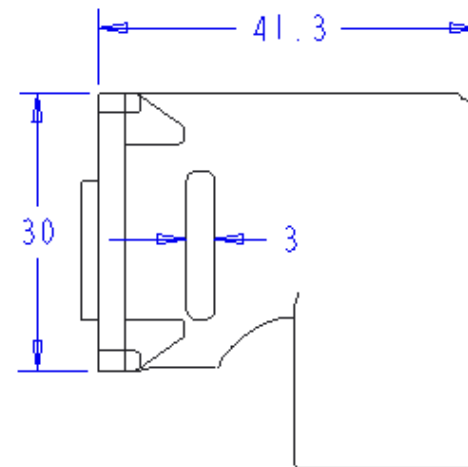
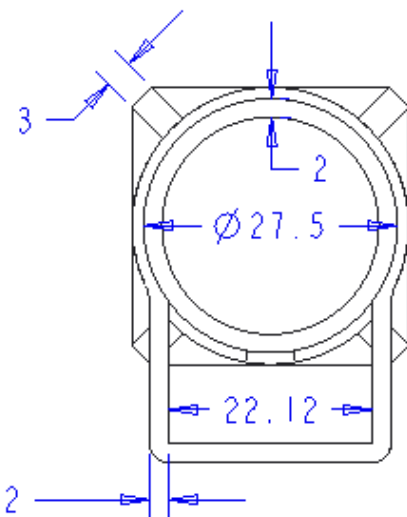
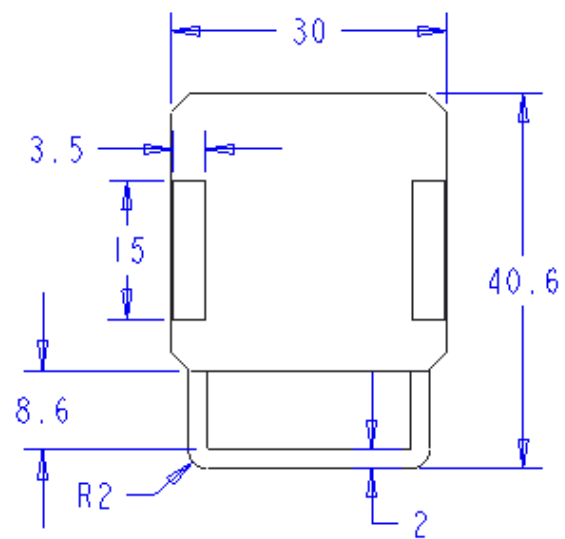
Scale 1:1	MSc thesis	Rover for the Robot Field Event
	Model name	Support base for the lid
Quantity 1	Designer	Kalykakis Emmanouil
	Maker	Froudarakis Konstantinos



Scale 1:1	MSc thesis	Rover for the Robot Field Event
	Model name	Pipe base
Quantity 1	Designer	Kalykakis Emmanouil
	Maker	Kontes Georgios

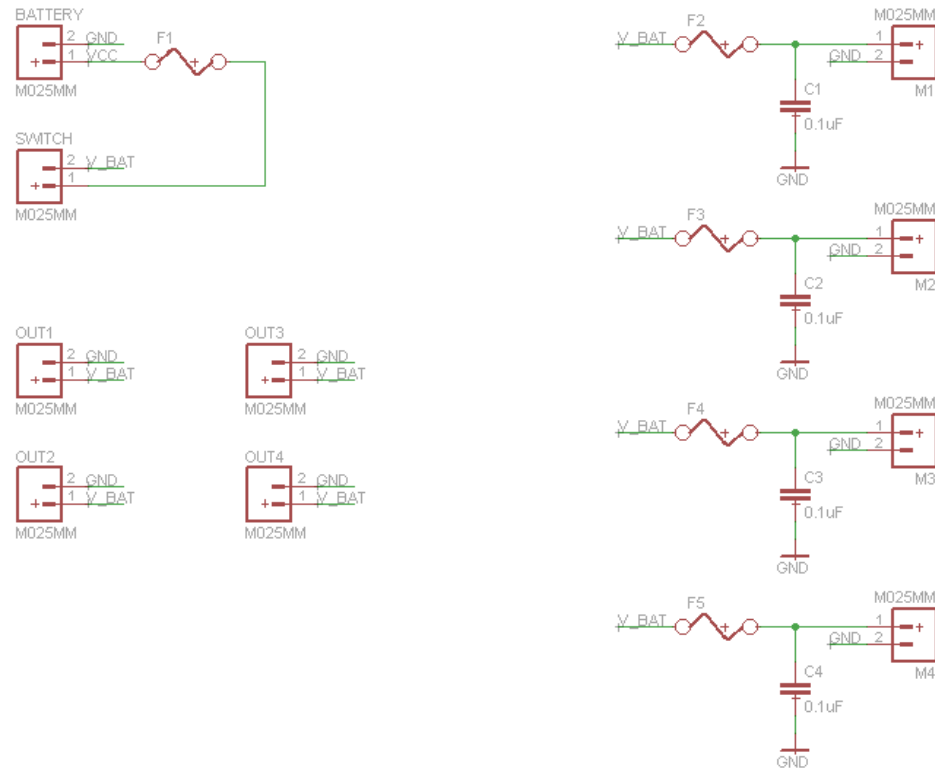


Scale 1:1	MSc thesis	Rover for the Robot Field Event
	Model name	Standoff
Quantity 4	Designer	Kalykakis Emmanouil
	Maker	Mountakis Nikolaos



Scale 1.5	MSc thesis	Rover for the Robot Field Event
	Model name	Camera mount
Quantity 1	Designer	Kalykakis Emmanouil
	Constructed by 3D printer	

## Παράρτημα Γ - Τυπωμένα κυκλώματα



**Rover for the robot field event**

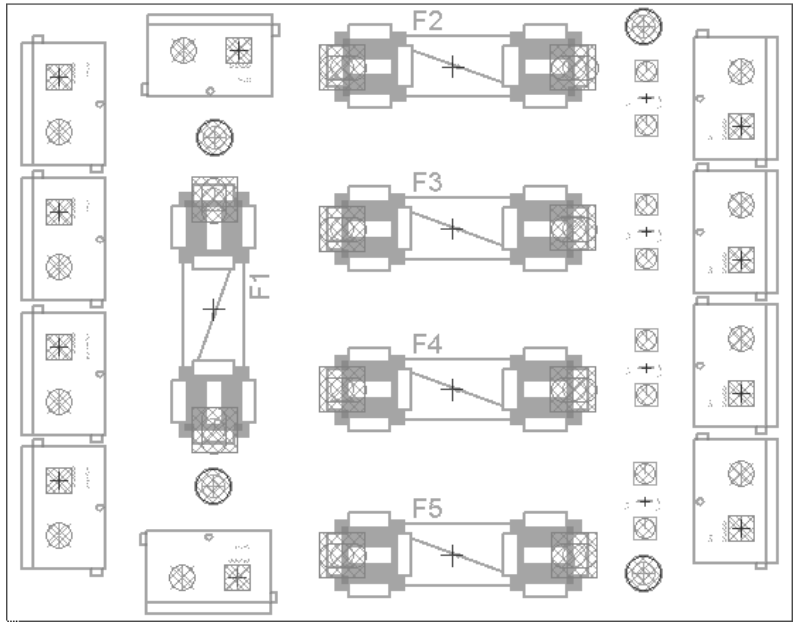
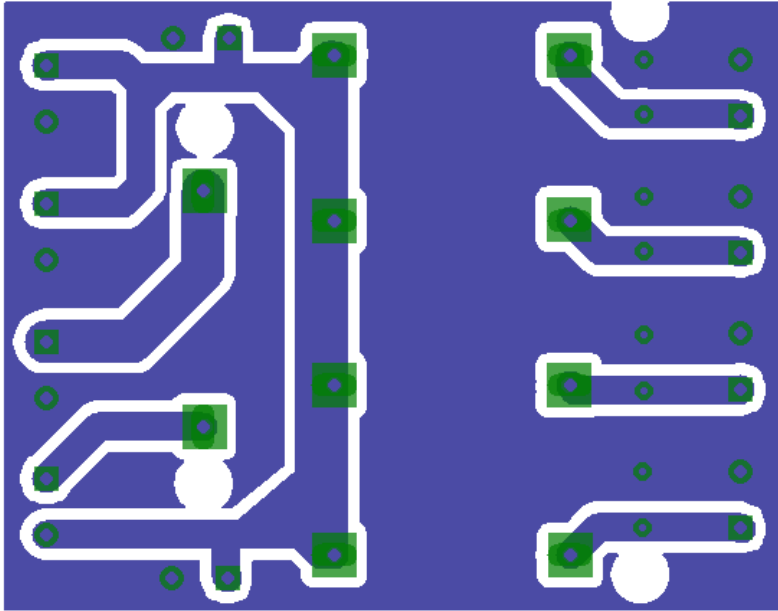
TITLE: Power Distribution Board

Document Number:

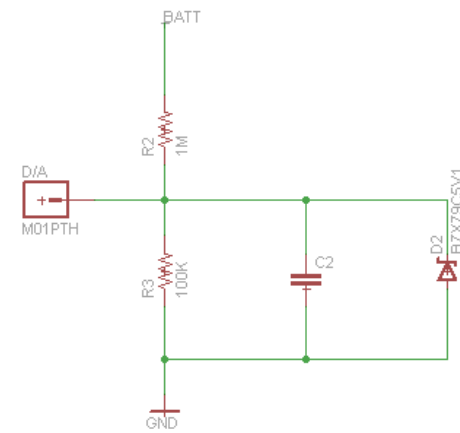
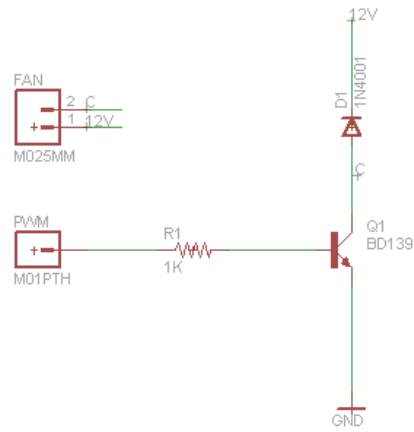
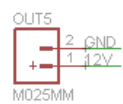
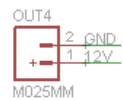
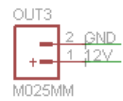
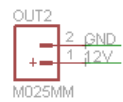
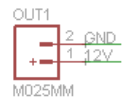
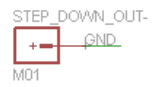
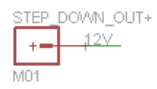
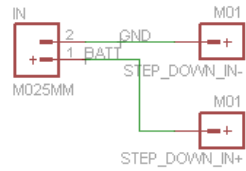
REV:

Date: 20/11/2014 9:29:36 μμ

Sheet: 1/1







**Rover for the robot field event**

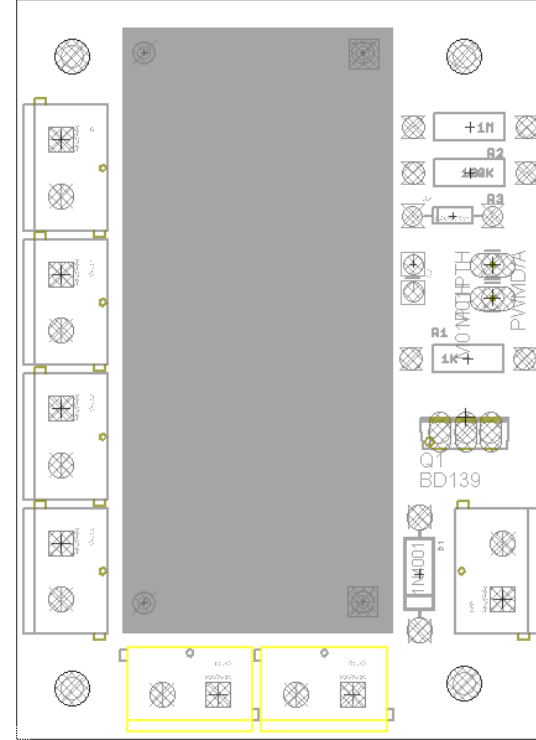
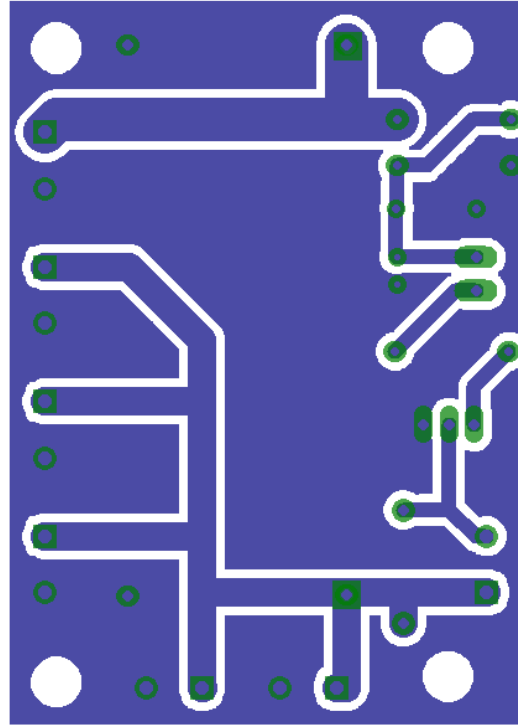
TITLE: Step down voltage regulator (12V)

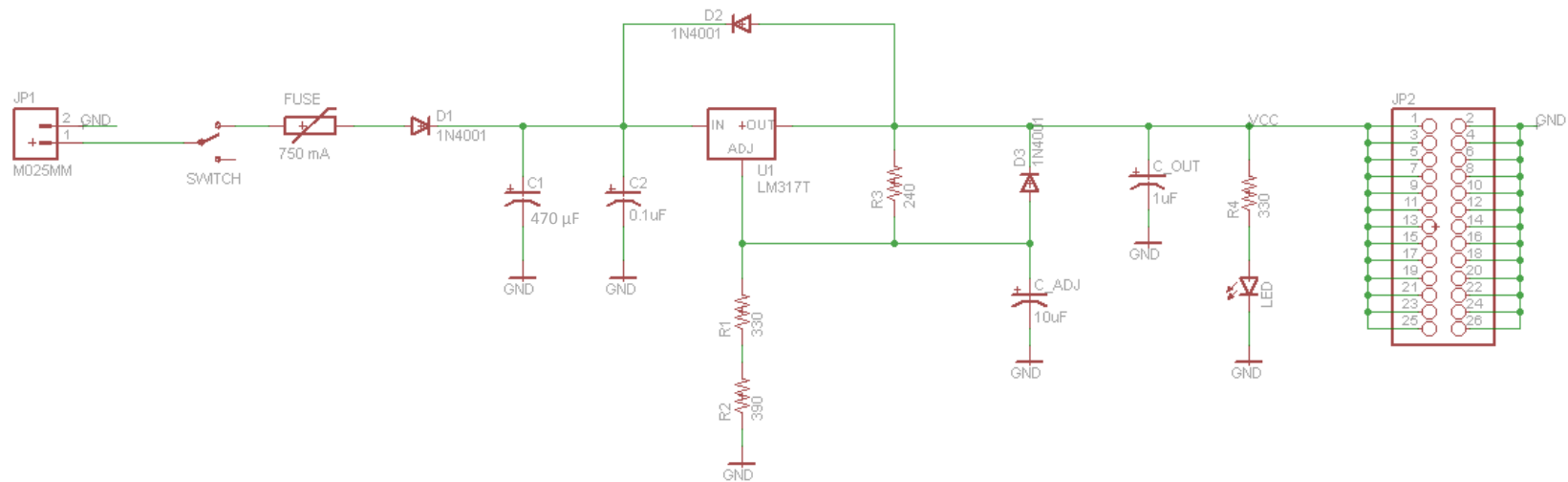
Document Number:

REV:

Date: 21/4/2015 10:24:41 µµ

Sheet: 1/1





### Rover for the robot field event

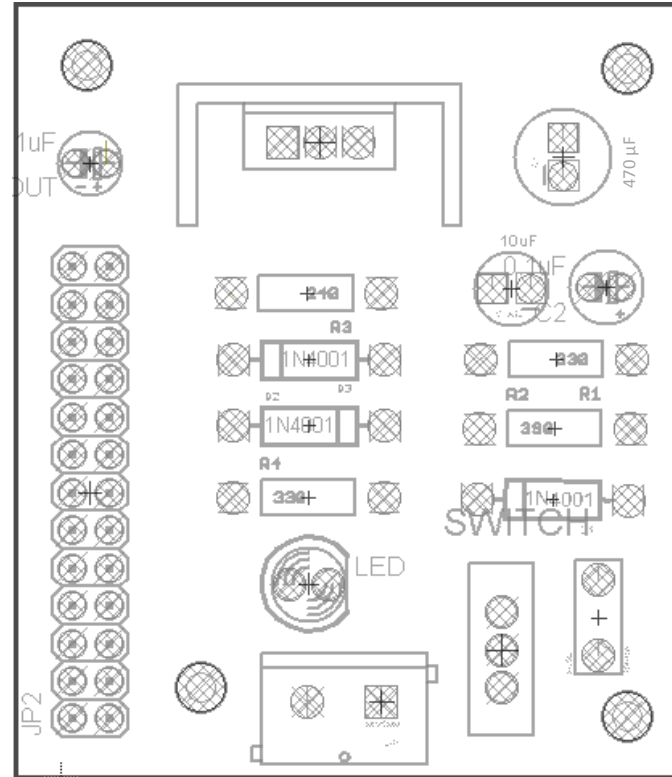
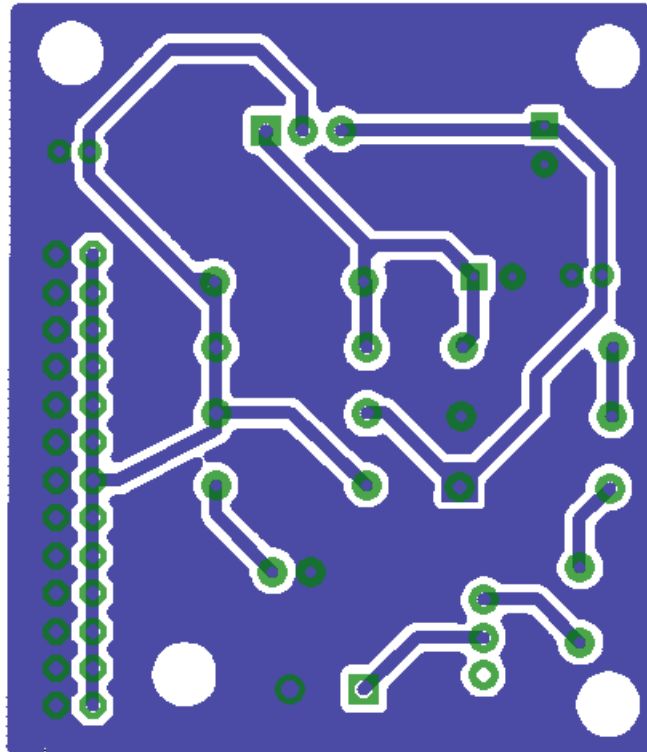
TITLE: Step down voltage regulator (5V)

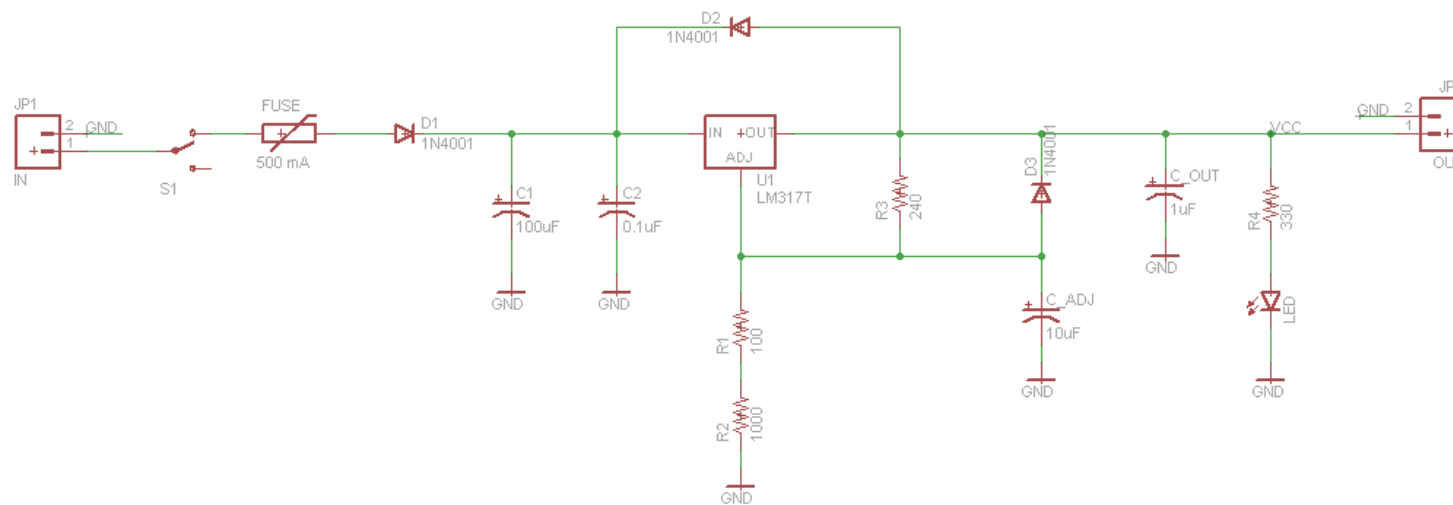
Document Number:

REV:

Date: 16/11/2014 11:38:01 πμ

Sheet: 1/1





### Rover for the robot field event

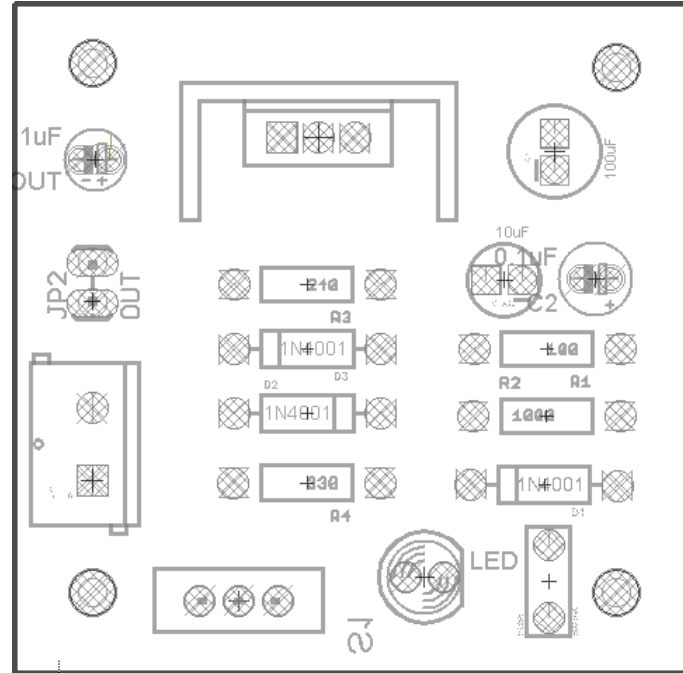
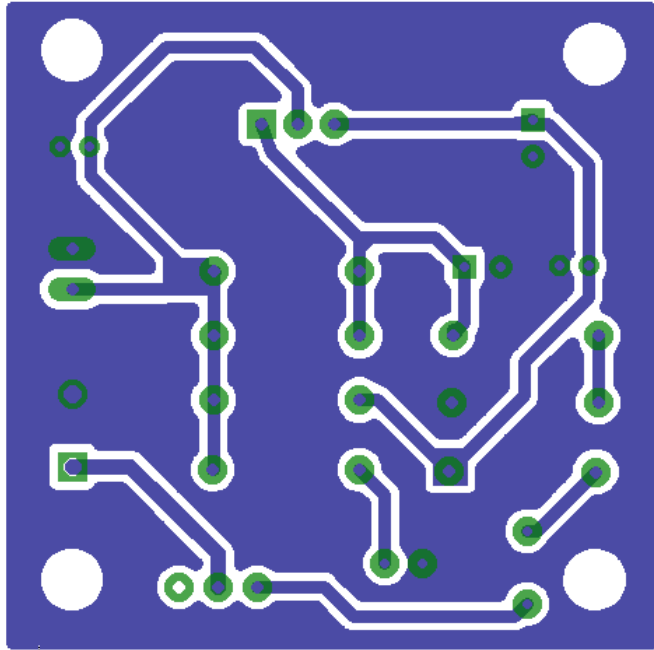
TITLE: Step down voltage regulator (7V)

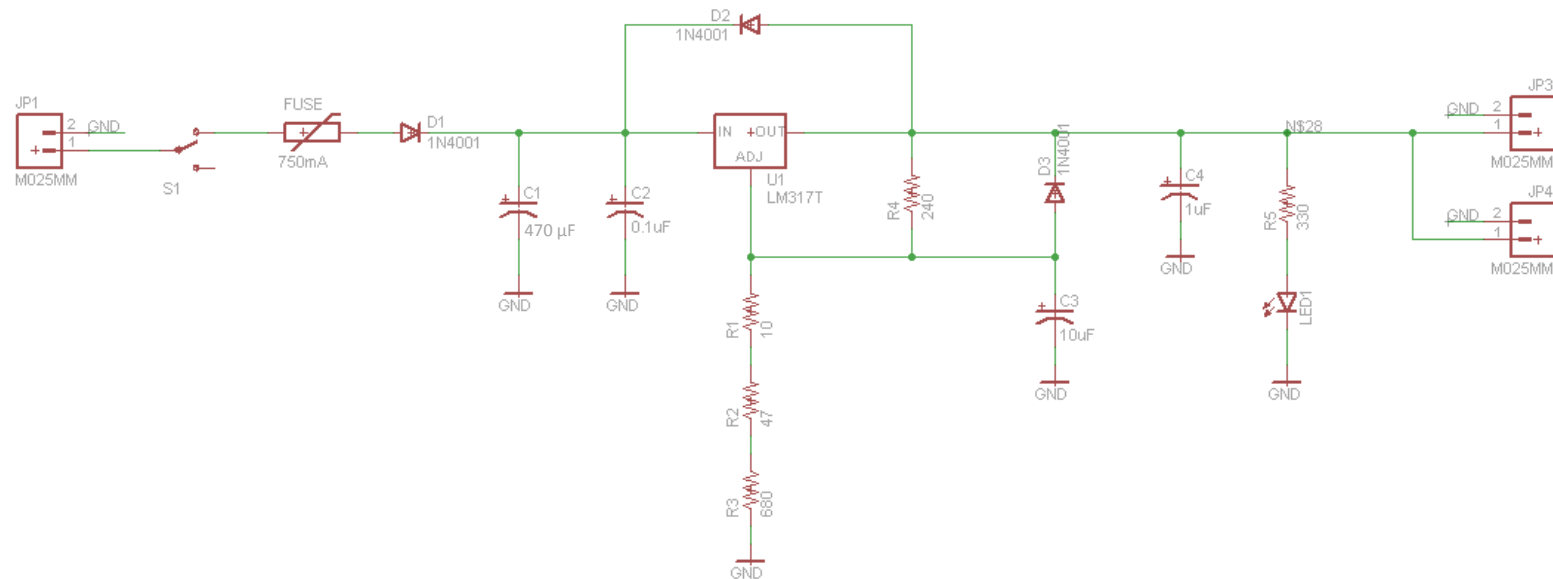
Document Number:

REV:

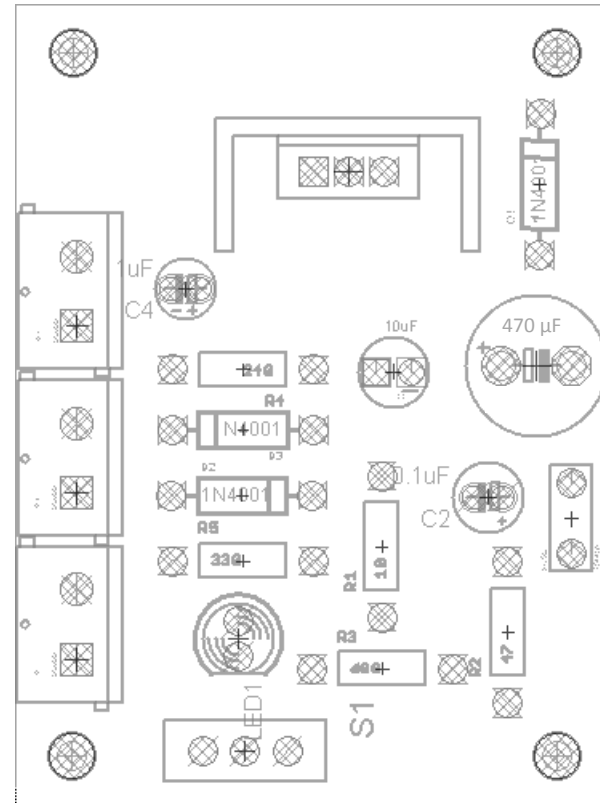
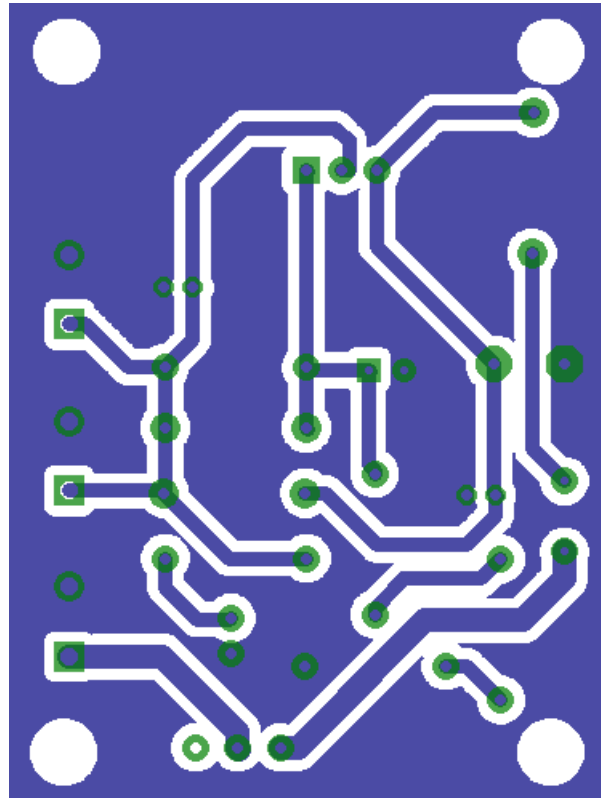
Date: 20/11/2014 12:01:55 πμ

Sheet: 1/1

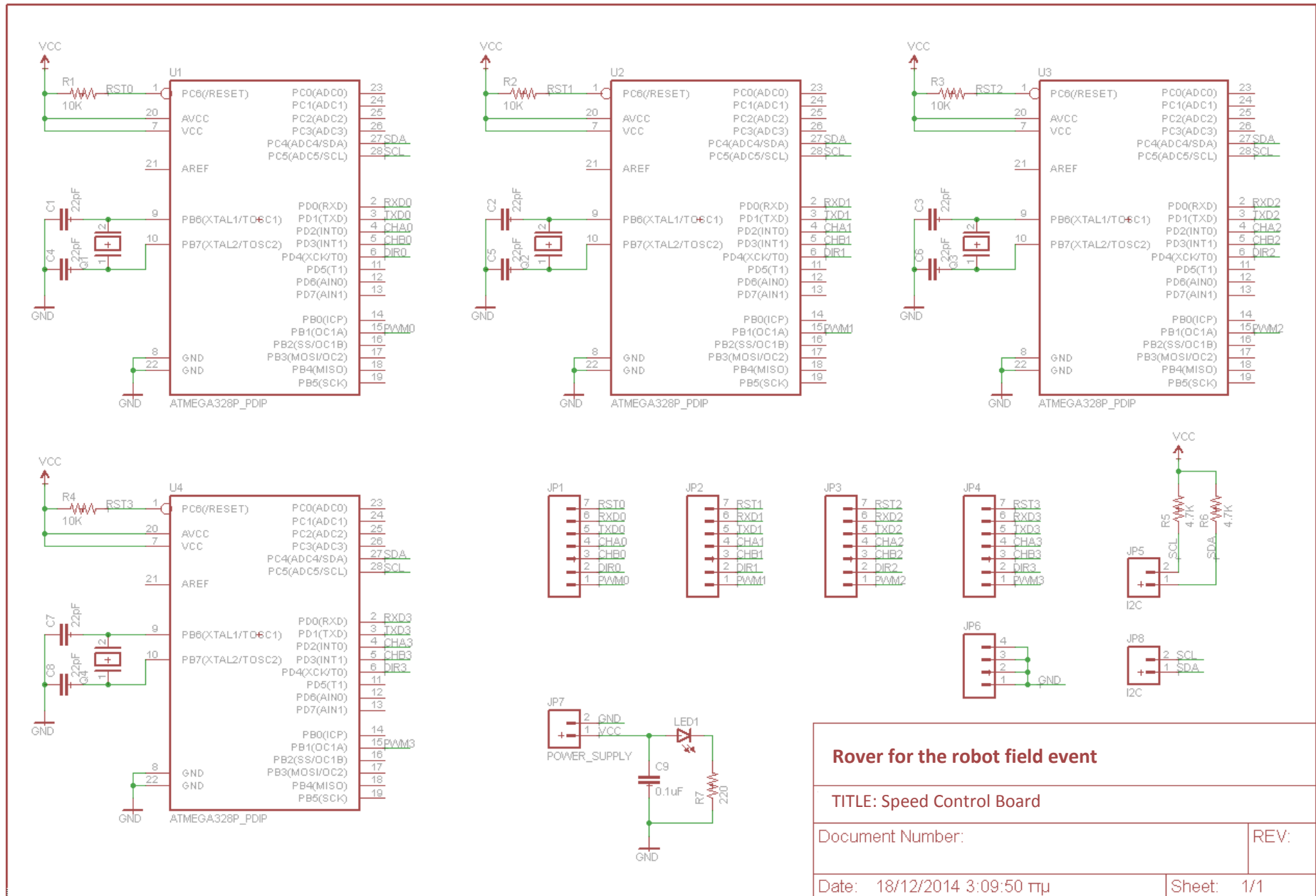




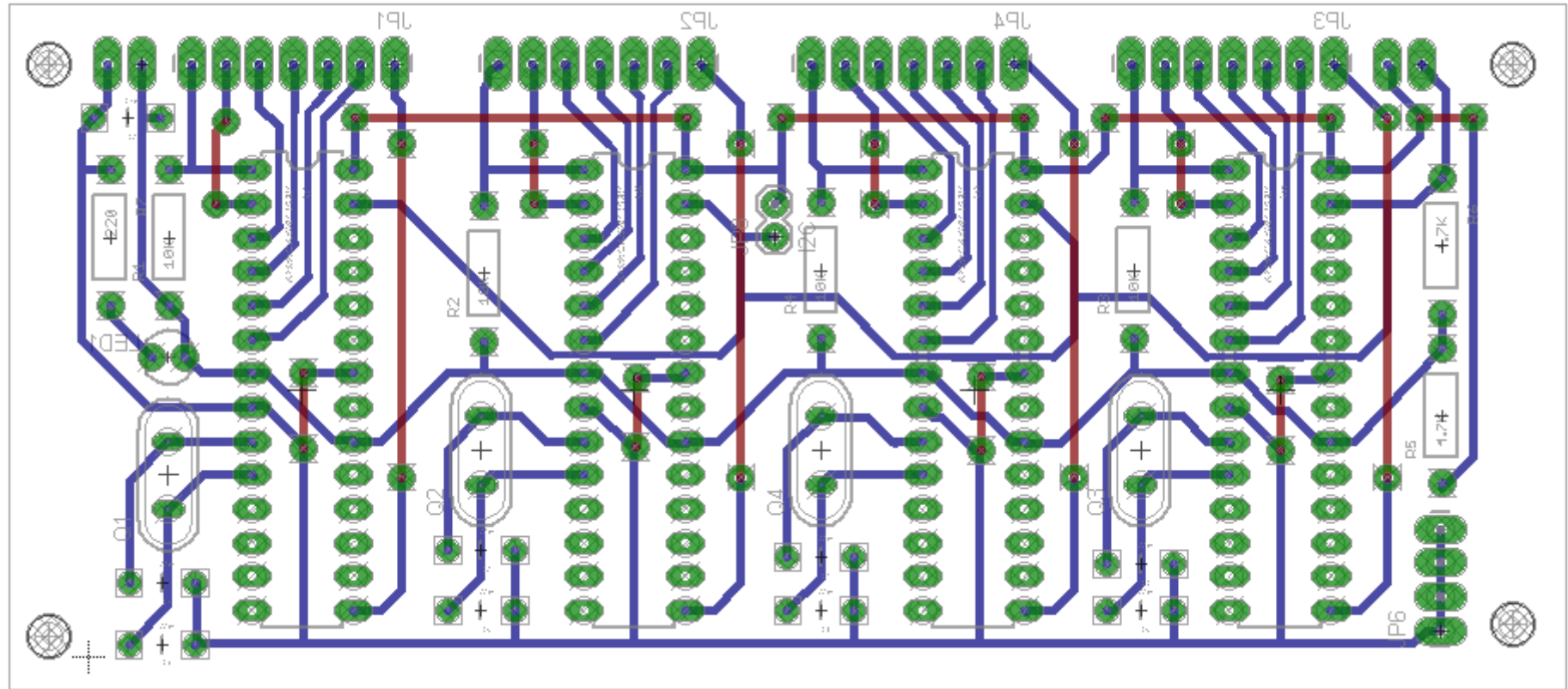
<b>Rover for the robot field event</b>	
TITLE: Step down voltage regulator (5V)	
Document Number:	REV:
Date: 3/5/2015 3:13:38 µµ	Sheet: 1/1

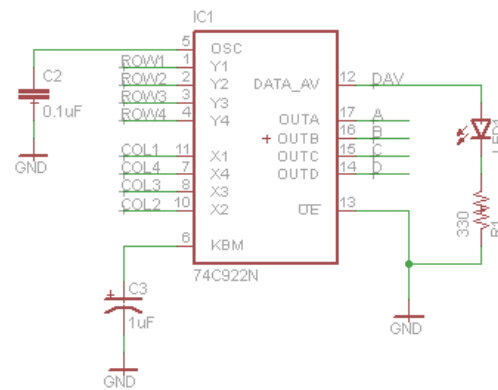
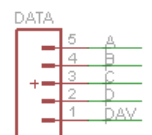
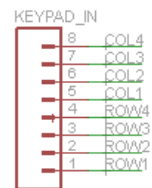
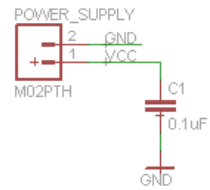




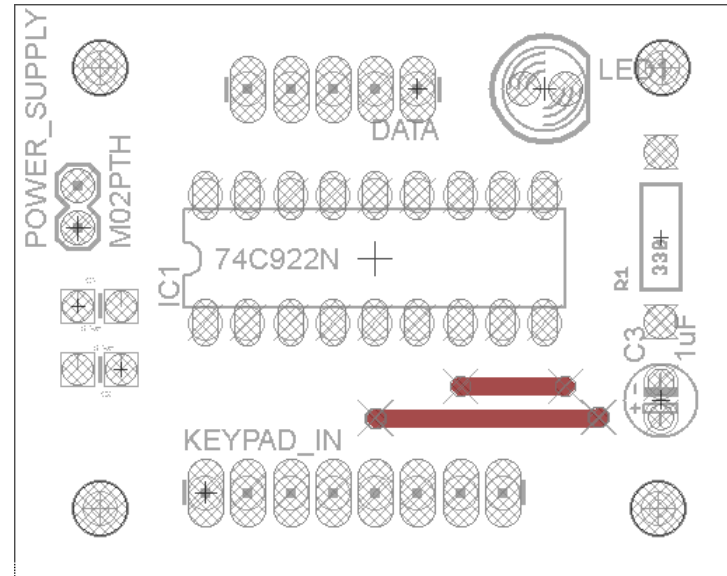
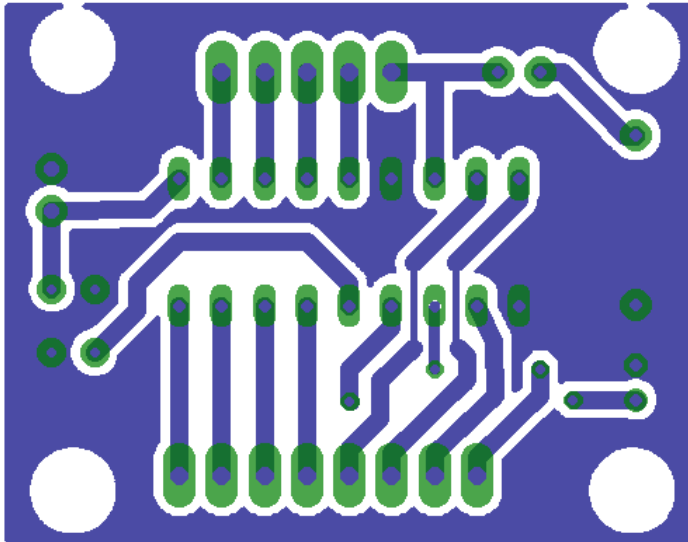


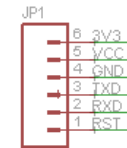
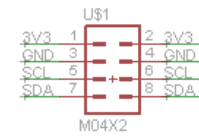
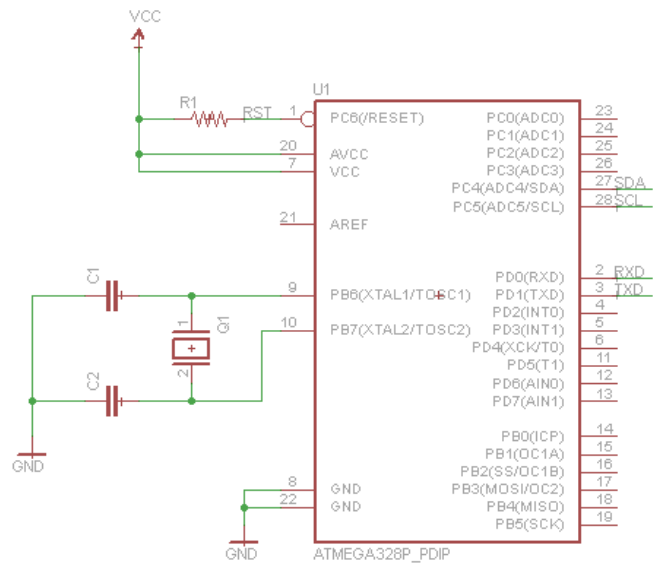
<b>Rover for the robot field event</b>	
TITLE: Speed Control Board	
Document Number:	REV:
Date: 18/12/2014 3:09:50 πμ	Sheet: 1/1





<b>Rover for the robot field event</b>	
TITLE: Keypad interface	
Document Number:	REV:
Date: 20/11/2014 11:05:58 μm	Sheet: 1/1





## Rover for the robot field event

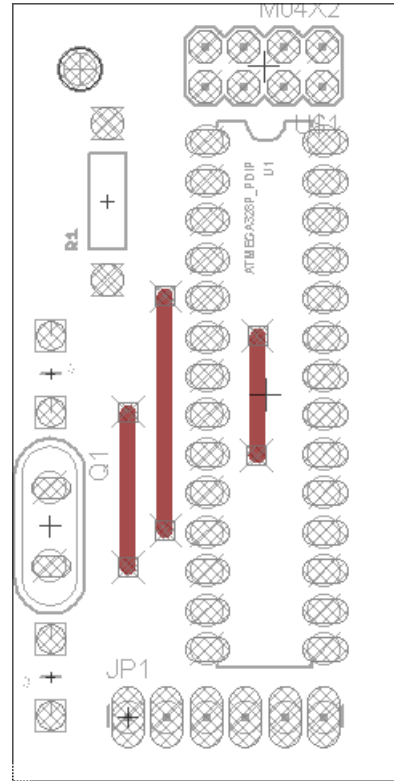
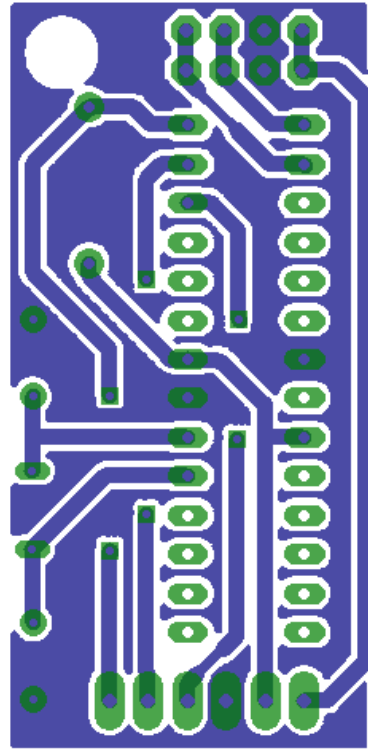
TITLE: Yaw estimation

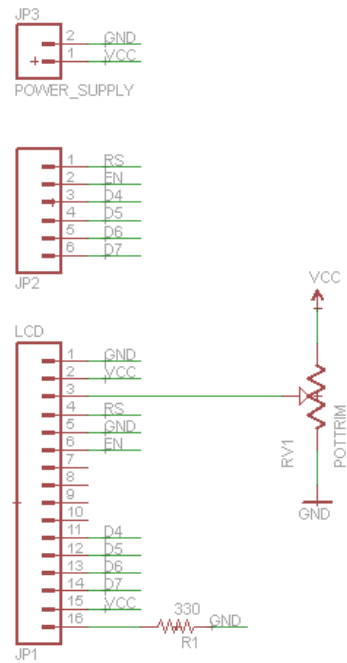
Document Number:

REV:

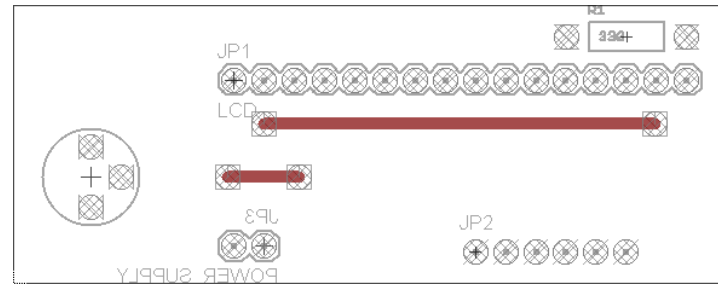
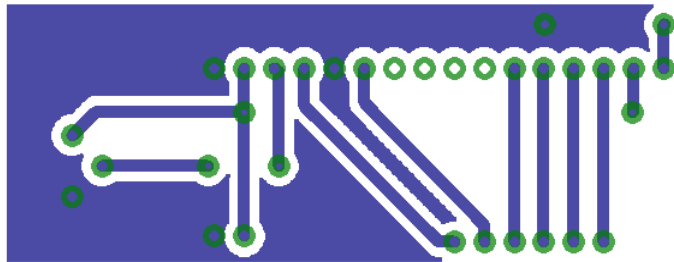
Date: 21/12/2015 2:06:22 πμ

Sheet: 1/1





<b>Rover for the robot field event</b>	
TITLE: LCD connector	
Document Number:	REV:
Date: 5/9/2015 10:14:31 μμ	Sheet: 1/1





## Παράρτημα Δ - Ρυθμίσεις μεταγλώττιστων

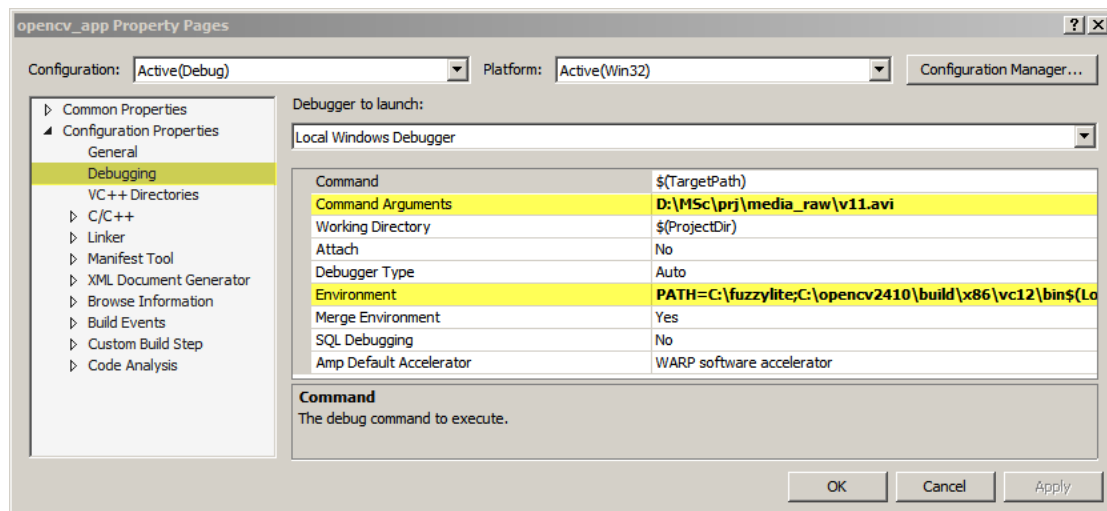
Για την μεταγλώττιση ενός project και την παραγωγή του εκτελέσιμου αρχείου (exe) που χρησιμοποιεί τις βιβλιοθήκες OpenCV και fuzzyLite τόσο στο ολοκληρωμένο περιβάλλον ανάπτυξης Microsoft Visual Studio 2013 όσο και στο Eclipse IDE θα πρέπει να γίνουν οι κατάλληλες ρυθμίσεις στο τρέχων έργο (project).

### MS Visual Studio 2013

Από το μενού επιλέγουμε Project → (application name) Properties → Configuration Properties → Debugging και στα κάτωθι πεδία συμπληρώνουμε ανάλογα:

Command Arguments: το πλήρες μονοπάτι του αρχείου εισόδου (μαγνητοσκοπημένο βίντεο avi) απαραίτητο για την offline εκτέλεση του αλγορίθμου (όρισμα στην συνάρτηση main).

Environment: προσθήκη στην μεταβλητή περιβάλλοντος PATH των μονοπατιών των αρχείων dll, PATH = C:\fuzzyLite;C:\opencv2410\build\x86\vc12\bin



Παράθυρο διαλόγου των ρυθμίσεων των ιδιοτήτων project του MS Visual Studio 2013

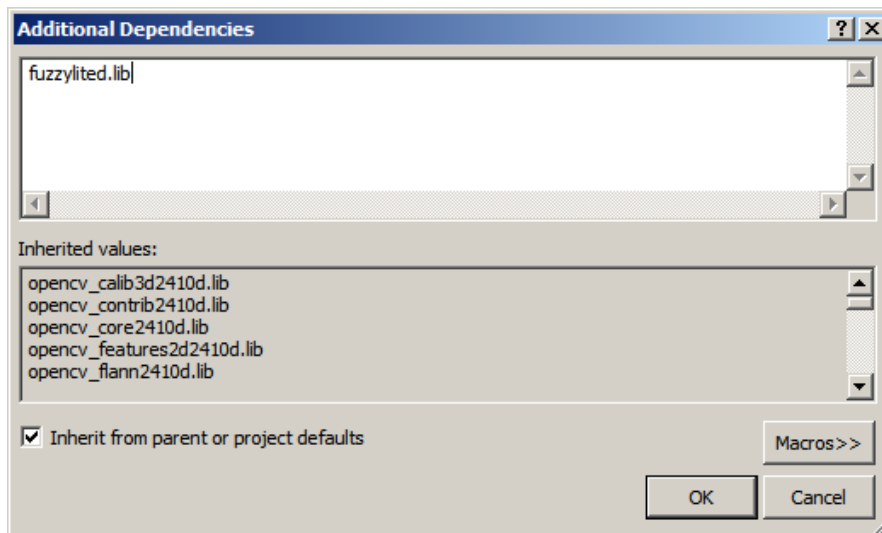
... → Configuration Properties → VC++ Directories ενδεικτικά συμπληρώνουμε  
Executable Directories: C:\opencv2410\build\x86\vc12\bin  
Library Directories: C:\opencv2410\build\x86\vc12\lib

... → Configuration Properties → C/C++ → General  
Additional Include Directories: C:\fuzzyLite; C:\opencv2410\build\include

... → Configuration Properties → Linker → General  
Additional Library Directories: C:\fuzzyLite; C:\opencv2410\build\x86\vc12\lib

Δήλωση των ονομάτων των αρχείων των χρησιμοποιούμενων βιβλιοθηκών:

... → Configuration Properties → Linker → Input → Additional Dependencies

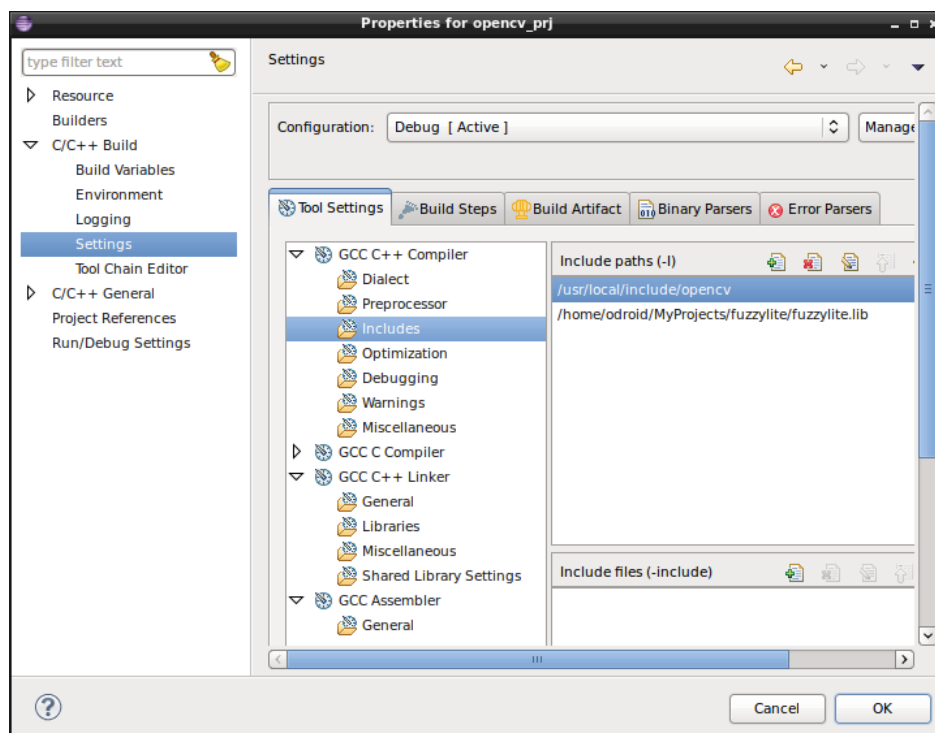


Παράθυρο διαλόγου καταχώρησης αρχείων των χρησιμοποιούμενων βιβλιοθηκών του MS Visual Studio 2013

Περισσότερες πληροφορίες παρέχονται στους αντίστοιχους συνδέσμους που αναφέρονται στις βιβλιογραφικές πηγές.

## Eclipse IDE

Από το μενού επιλογών Projects → Properties → C/C++ Build → Settings και στην καρτέλα Tool Settings → GCC C++ Compiler → Includes προσθέτουμε  
 /usr/local/include/opencv  
 /home/odroid/MyProjects/fuzzylite/fuzzylite.lib



Παράθυρο διαλόγων των ρυθμίσεων των ιδιοτήτων project του Eclipse IDE

Στην ίδια καρτέλα επιλέγουμε GCC C++ Linker → Libraries, στο πεδίο Libraries προσθέτουμε τα ονόματα των αρχείων των χρησιμοποιούμενων βιβλιοθηκών π.χ. fuzzylited, opencv\_core κ.α. και στο πεδίο Library search path την πλήρη διαδρομή του μονοπατιού που αυτές βρίσκονται π.χ.  
`/home/odroid/MyProjects/fuzzylite/fuzzylite.lib.debug.bin, /usr/local/lib`

Στο παράθυρο διαλόγου Properties επιλέγουμε από την λίστα Run/Debug Settings, έπειτα το όνομα για το ενεργό project και κατόπιν Edit. Στην καρτέλα Arguments καταχωρούνται τα ορίσματα που διοχετεύονται στην συνάρτηση main π.χ. το πλήρες μονοπάτι ενός μαγνητοσκοπημένου βίντεο .

Στην καρτέλα Environment για την βιβλιοθήκη fuzzylite και για την μεταγλώττιση εντός του IDE απαιτείται η εκχώρηση στην μεταβλητή περιβάλλοντος LD\_LIBRARY\_PATH του πλήρους μονοπατιού στο οποίο αυτή βρίσκεται π.χ.  
`/home/odroid/MyProjects/fuzzylite/fuzzylite.lib/debug/bin`

## Παράρτημα Ε - Πηγαίος Κώδικας

Παρατίθενται μέρος του πηγαίου κώδικα, όλα τα απαραίτητα αρχεία βρίσκονται στον υπερσύνδεσμο [Download Source Code](#)

Αρχείο sys\_ident.m

Λειτουργίες:

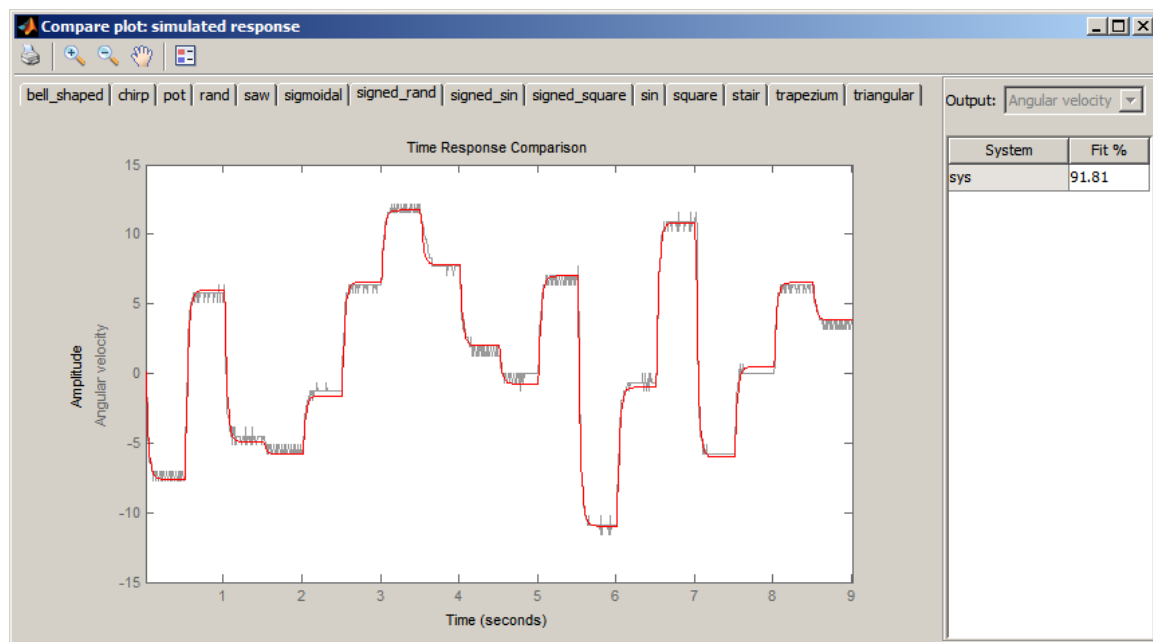
1. Προσπέλαση αρχείων \*.mat των πειραματικών δεδομένων (κάθε ένα από αυτά περιέχει: το διάγραμμα των χρονικών στιγμών της μέτρησης, την τρέχουσα τιμή του σήματος αναφοράς (OCR1A), την τιμή του μετρητή απαρίθμησης του αποκωδικοποιητή θέσης και την εκτιμώμενη γωνιακή ταχύτητα του κινητήρα με βάση το φίλτρο Savitzky-Golay)
2. δημιουργία αντίστοιχων αντικειμένων iddata και συγχώνευση τους,
3. εύρεση της εκτιμώμενης συνάρτησης μεταφοράς του γραμμικού μοντέλου,
4. υπολογισμός του στατικού κέρδους και των κερδών του PID ελεγκτή με βάση την μέθοδο άμεσης σύνθεσης.

```
% create iddata objects
% samples collected at the a sampling rate of 0.01 second
clear all; close all; clc;
folder_name = uigetdir;
if (folder_name ~= 0)
    cd(folder_name)
    currentFolder = pwd;
    info = what(currentFolder);
    mat_files = info.mat;
    if (~isempty(mat_files))
        dt = 0.002;
        for i = 1:numel(mat_files);
            fname = cell2mat(mat_files(i));
            load(fname);
            [~,name,~]=fileparts(fname);
            varname = genvarname(name);
            vel = [0;2*pi()*diff(encPos)./(980*diff(time)*dt)]; % rad/sec
            V = r*24/511;
            % estimated angular velocity with differentiation
            temp = iddata(vel,r,0.01);
            temp = iddata(vel,V,0.01);
            % estimated angular velocity with Savitzky-Golay Filter
            temp = iddata(fvel,r,0.01);
            temp = iddata(fvel,V,0.01);
            temp.ExperimentName = name;
            temp.InputName = 'Voltage';
            temp.InputUnit = 'Volt';
            temp.Name = name;
            temp.OutputName = 'Angular velocity';
            temp.OutputUnit = 'rad/s';
            eval([varname '= temp;']);
        end
        clear temp;
        vars = whos;
        fh = str2func('merge');
        iddata_names = char.empty(1,0);
        for i = 1:numel(vars)
            if strcmp(vars(i,1).class,'iddata')
                iddata_names = [iddata_names,vars(i,1).name, ','];
            end
        end
        iddata_names = iddata_names(1:numel(iddata_names)-1);
        evalin('caller',[ 'z = merge(' ,iddata_names,') ']); % creating dataset
```

```

np = 2;
nz = 0;
sys = tfest(z,np,nz) % linear model
compare(z,sys,'r');
DC_gain = dcgain(sys) % (rad/sec)/V
% Direct Synthesis Method
p = -roots(sys.den); % poles
tau1 = 1/p(1);
tau2 = 1/p(2);
K = (sys.num*511/24)/(p(1)*p(2));
d = 0.035; % delay
tau = 0.05;
C = 1/(K*(tau+d));
format long;
P = C*(tau1+tau2)
I = C
D = C*tau1*tau2
clearvars -except sys DC_gain z P I D
else
disp('There are no .mat files in the selected folder!');
end
else
disp('User selected Cancel');
end
break;
[~,fit,~] = compare(z,sys); % no plot is generated
fit = cell2mat(fit');
r = sum(fit)/length(fit)

```



Σύγκριση απόκρισης του εκτιμώμενου γραμμικού μοντέλου με αυτή του πραγματικού

## Αρχείο xls4SB.m

### Λειτουργίες:

1. Προσπέλαση αρχείων \*.mat των πειραματικών δεδομένων,
2. Αυτοματοποιημένη δημιουργία αρχείου excel με πολλαπλά φύλλα εργασίας. Κάθε φύλλο περιέχει το διάγραμμα των χρονικών στιγμών και την αντίστοιχη τιμή του σήματος αναφοράς (OCR1A).

Χρήση: παραγωγή σημάτων αναφοράς με τη βοήθεια του Signal Builder για την σύγκριση του μοντέλου Simulink με το πραγματικό.

```
clear all; clc;
disp('Choose a folder that contains *.mat files ..');
folder_name = uigetdir;
if (folder_name ~= 0)
    cd(folder_name)
    currentFolder = pwd;
    info = what(currentFolder);
    mat_files = info.mat;
    num = numel(mat_files);
    if (num ~= 0)
        objExcel = actxserver('Excel.Application');
        set(objExcel, 'Visible', 1);
        % insert a new workbook
        Workbooks = objExcel.Workbooks;
        Workbook = invoke(Workbooks, 'Add',1); % add one sheet
        for i=2:num
            invoke(objExcel.Sheets, 'Add');
        end
        Sheets = objExcel.ActiveWorkBook.Sheets;
        for i = 1:num
            fname = cell2mat(mat_files(i));
            load(fname);
            [~,name,~]=fileparts(fname);
            t = time*2*10^-3;
            t = cat(1, 'time', num2cell(t));
            r = cat(1, 'signal_values', num2cell(r));
            data = [t,r];
            range = ['A1:B', num2str(numel(t))];
            invoke(get(Sheets, 'Item', i), 'Activate');
            Sheets.Item(i).Name = name;
            Activesheet = objExcel.Activesheet;
            ActivesheetRange = get(Activesheet, 'Range', range);
            set(ActivesheetRange, 'Value', data);
        end
        path = [currentFolder '\wv.xls'];
        objExcel.DisplayAlerts = false; % overwrite if it exists
        invoke(Workbook, 'SaveAs', char(path));
        Workbook.Saved = 1;
        invoke(Workbook, 'Close');
        invoke(objExcel, 'Quit');
        delete(objExcel);
        disp(['The file ', path, ' was created successfully']);
    else
        disp('There are no .mat files in the selected folder!');
    end
else
    disp('User selected Cancel')
end
clear all;
```

Firmware της αναπτυξιακής πλακέτας Arduino Uno για την πραγματοποίηση των πειραματικών μετρήσεων και της καταγραφής των δεδομένων που αποσκοπούν στην μοντελοποίηση κάθε κινητήριου τροχού.

```
#include "Encoder.h"

#if 1 // 1: open loop, 0: closed loop control
#define OPEN_LOOP
#else
#undef OPEN_LOOP
#endif

#if 1
#define PRINT2SERIAL
#else
#undef PRINT2SERIAL // Bridge Control Panel Software
#endif

// FLW
//#define dcgain 0.628607531451523
//maxVel = 15.435532843214652; // rad/s
// FRW
#define dcgain 0.621527773238278
float maxVel = 15.255545597116900;
// RRW
//#define dcgain 0.623318030327511
//maxVel = 15.170036141631114;
// RLW
//#define dcgain 0.616722919482069
//maxVel = 15.077451681639259;

#define DZ_POS 20
#define DZ_NEG 28

#define Vcc 24.0 // Volt
#define tt 1*4500 // total time, 1*4500 = 9 sec

// H bridge
#define dirPin 4
#define pwmPin 9

// Hall sensor
#define encPinA 3
#define encPinB 2

#define freq_factor 5 // sample time 5*2ms = 0.01s, f = 100 Hz
float Ts = 0.01; // sample time
#define dt 0.002

#define TOP 511.0 // 0x01FF, 9 bit
// for the waveform creation
#define v1 255 // min value for the OCR1A
#define v2 433 // max

volatile uint32_t t = 0, previousT; // elapsed time = t*dt
volatile uint8_t timer0_compa_count = 0;
volatile boolean wait;
volatile boolean toggleFlag = 1;

Encoder we(encPinA,encPinB);
int32_t encPos;

enum MODE {STEP,RAMP,STAIR,TRIANGULAR,SQUARE,SIGNED_SQUARE,SAW,SIN,SIGNED_SIN,CHIRP,TRAPEZIUM,
BELL_SHAPED,SIGMOIDAL,RAND,SIGNED_RAND,POT};

MODE current_mode = TRAPEZIUM;
```

```

int16_t ref = 0;

float sp, ocr1a, diff, err, err1, err2, u=0, u1, y;

// Ziegler-Nichols method
//float Ku = 4.0, Tu = 0.13;
//float Kp = 0.6*Ku, Ki = 2*Kp/Tu, Kd = Kp*Tu/8.0; // classic PID
//float Kp = 0.2*Ku, Ki = 2*Kp/Tu, Kd = Kp*Tu/3.0; // no overshoot

float Kp = 2.4, Ki = 43.636, Kd = 0.033;

// Direct synthesis method
//float Kp = 0.700810662369615, Ki = 18.708059029390377, Kd = 0.003787989533032;

uint8_t idx = 0;
// Savitzky-Golay filter
// order of polynomial fit: 2
const uint8_t n = 9; // window size = 3:2:19, must be odd
float* c = (float*)malloc(sizeof(float)*n);
uint16_t N[] = {2,10,28,60,110,182,280,408,570}; // normalization
// queue
int32_t* q = (int32_t*)malloc(sizeof(int32_t)*n);

union u_tag {
    byte b[4];
    float val;
} fval;

void setup() {
    Serial.begin(115200);

    analogReference(EXTERNAL);
    randomSeed(analogRead(A5));

    // Savitzky-Golay differentiation filter
    // order of polynomial fit: 2
    // Calculate S-G coefficients
    for(uint8_t i=0;i<n;i++) {
        *(c+i) = (-(n-1)/2.0 + i)/N[int((n-3)/2.0)];
        // Serial.println(c[i],15);
        *(q+i) = 0;
    }
    pinMode(dirPin,OUTPUT);
    digitalWrite(dirPin,HIGH); // 1:CW , 0:CCW
    pinMode(pwmPin,OUTPUT);

    TCNT1 = 0;
    TCCR1A = 0;
    TCCR1B = 0;
    TCCR1B = 0;
    // duty cycle = OCR1A/TOP
    OCR1A = 0;
    // Mode 2, Fpwm = 15.656 KHz
    // PWM, Phase Correct 9-bit, TOP = 0x1FF
    bitSet(TCCR1A,WGM11);
    // Clear OC1A on compare match when up-counting
    // Set OC1A on compare match when down-counting
    bitSet(TCCR1A,COM1A1);
    // No prescaling
    bitSet(TCCR1B,CS10);

    TCNT0 = 0;
    TCCR0A = 0;
    TCCR0B = 0;
    bitSet(TCCR0A,WGM01); // CTC mode
    bitSet(TCCR0B,CS02); // N = 256
    OCR0A = 125-1;
    bitSet(TIMSK0,OCIE0A);
}

```



```

void loop() {
    previousT = t;
    if (t > tt) {
        // stop
        cli();
        OCR1A = 0;
    } else {
        switch(current_mode) {
            case STEP: {
                ref = v2;
                break;
            }
            case RAMP: {
                ref=(ref<v2)? 40.0*t*dt:v2;
                break;
            }
            case STAIR: {
                static boolean flag = 0;
                static int sign = 1;
                if(flag == toggleFlag) {
                    ref += (sign)*85;
                    flag ^= 1;
                    if ((ref == TOP-1)|| (ref == 0)) sign *= -1;
                }
                break;
            }
            case TRIANGULAR: {
                //triangular wave
                if(toggleFlag) {
                    ref = round(v1 + (v2-v1)/1.5*dt*(t%750));
                } else {
                    ref = round(v2 - (v2-v1)/1.5*dt*(t%750));
                }
                break;
            }
            case SQUARE: {
                // square waveform
                ref = (toggleFlag)?v1:v2;          // toggle every 1.5 sec
                break;
            }
            case SIGNED_SQUARE: {
                ref = (toggleFlag)?v1:-v1;
                break;
            }
            case SAW: {
                ref = round(v1 + (v2-v1)/3.0*dt*(t%1500));
                break;
            }
            case SIN: {
                // sine wave
                ref = round(v2/2.0+(v2/2.0)*sin(2*PI/3.0*t*dt-PI/2));
                break;
            }
            case SIGNED_SIN: {
                ref = round(v2*sin(2*PI/6.0*t*dt));
                break;
            }
        }
    }
}

```

```

case CHIRP : {
// T1 > T2 -> f1 < f2
//  $\omega = 2\pi f = d\phi/dt$ ,  $f = f1 + (f2-f1)/tt*t$ 
// freqSlope =  $2\pi*(f2-f1)/targetTime$ 
// instantaneousFreq = initialFreq + freqSlope*t
// instantaneousPhase = integral(instantaneousFreq) =
// initialFreq*t + 0.5*freqSlope*t^2
// f1 = 1/3.0 (initial frequency),
// f2 = 1/(1/8.0) (frequency at target time)
ref = round(v2/2.0*(1+sin(PI*(2*1/3.0*t*dt + (1/(1/8.0)-
1/3.0)/(tt*dt)*pow(t*dt,2))-PI/2.0)));

break;
}
case TRAPEZIUM: {
// count is increased every 1.5 sec
static unsigned int count = 0;
static boolean flag = toggleFlag;
if (flag == toggleFlag) {
count++;
flag^=1;
}
if (count<= 2)
ref = round(v2/3.0*dt*(t%1500));
else if (count <= 4)
ref = v2;
else if (count <= 6)
ref = round(-v2/3.0*((t%4500)*dt-9.0));
else
count = 1;
break;
}
case BELL_SHAPED: {
const float a=3.0, b=4.0, c=4.5;
float y = abs(t*dt-c)/a;
float val = v2*(1/(1+pow(y,(2*b))));
ref = (int)val;
break;
}
case SIGMOIDAL: {
const float a=1.0, c=4.0;
float val = 1/(1+exp(-a*(t*dt-c)));
ref = (int)(v2*val);
break;
}
case RAND: {
static boolean flag = 1;
if ((t % random(100,250) == 0) || flag) { //100:0.2s-250:0.5s
flag = 0;
ref = random(0,v2);
}
break;
}
case SIGNED_RAND: {
static boolean flag = 1;
if ((t % 250 == 0) || flag) { // 250: 0.5s
flag = 0;
ref = random(-v2,v2);
}
break;
}
case POT : {
ref = map(analogRead(A0),0,1023,-TOP,TOP);
break;
}
}
}

```

```

ref = constrain(ref,-TOP,TOP);

encPos = we.read();

diff = 0;

// right shift
for(idx=0;idx<n-1;idx++)
    *(q+idx) = *(q+idx+1);

q[idx] = encPos;

for (idx=0;idx<n;idx++)
    diff += c[idx]*q[idx];

y = 2*PI*diff/(980*Ts); // estimated angular velocity [rad/sec]

if(isnan(y)) y = 0;

#ifdef OPEN_LOOP
    if (ref >=0) {
        digitalWrite(dirPin,HIGH);
    } else {
        digitalWrite(dirPin,LOW);
    }
    OCR1A = abs(ref);
#else // closed loop control
    maxVel = dcgain*Vcc;

    sp = maxVel*ref/TOP;

    err2 = isnan(err1)?0:err1; // before previous
    err1 = isnan(err)?0:err; // previous error

    err = sp - y; // current error = setpoint - actual

    u1 = isnan(u)?0:u;

// control signal, classic PID
u=u1+err*(Kp+Ki*Ts/2.0+Kd/Ts)+err1*(Ki*Ts/2.0-Kp-2*Kd/Ts)+err2*(Kd/Ts);

    if (u >= 0) {
        digitalWrite(dirPin,HIGH);
        ocr1a = DZ_POS + u/Vcc*(TOP-DZ_POS);
    } else {
        digitalWrite(dirPin,LOW);
        ocr1a = DZ_NEG - u/Vcc*(TOP-DZ_NEG);
    }

    if (ocr1a>TOP) {
        u = u1;
    } else {
        OCR1A=(ref!=0)?ocr1a:0;
    }
#endif

```

```

#ifdef PRINT2SERIAL
    #ifdef OPEN_LOOP
        Serial.print(t); Serial.print("\t"); // time vector
        Serial.print(ref); Serial.print("\t");
        Serial.print(encPos); Serial.print("\t");
        Serial.println(y,4); // response
    #else // closed loop ctrl
        Serial.print(t); Serial.print("\t");
        Serial.print(ref); Serial.print("\t");
        Serial.print(OCR1A); Serial.print("\t");
        Serial.print(encPos); Serial.print("\t");
        Serial.print(err); Serial.print("\t");
        Serial.print(u,4); Serial.print("\t");
        Serial.print(sp,4); Serial.print("\t"); // desired angular velocity
        Serial.println(y,4);
    #endif
#else //4 Bridge Control Panel Software:Rx8 [h=43] @1ref @0ref @3res @2res @1res @0res
    Serial.print("C");
    // setpoint
    // writes binary data (as a byte) to the serial port
    Serial.write(ref>>8); // hi8 : the most significant 8 bits of 16 bit integer
    Serial.write(ref&0xff); // lo8 : the least significant
    // response
    fval.val = y;
    for (int i=3;i>=0;i--)
        Serial.write(fval.b[i]);
#endif
}

wait = true;
while(wait);

Ts = (t-previousT)*dt;
}

ISR(TIMER0_COMPA_vect) {
    t++; // every 2 ms
    if(++timer0_compa_count == freq_factor) {
        timer0_compa_count = 0;
        wait = false;
    }
    if (t % 750 == 0) toggleFlag ^= 1; // every 1.5 sec
}

```

Εγκατεστημένο firmware σε κάθε μικροελεγκτή ATmega328 του τυπωμένου κυκλώματος του ελέγχου της ταχύτητας περιστροφής των τροχών.

```
// i2c slave

#include "Encoder.h"
#include <Wire.h>

// ATmega328P-PU address
#define slave_address 0 // 0:3

// Hall sensor
#define encPinA 2 // swap the pin numbers to the left side!
#define encPinB 3

#if 0 // 1: open loop, 0: closed loop control
#define OPEN_LOOP
#else
#undef OPEN_LOOP
#endif

#if slave_address == 0 // FLW
#define DZ_POS 20
#define DZ_NEG 28
#define Kp 0.700810662369615
#define Ki 18.708059029390377
#define Kd 0.003787989533032
#elif slave_address == 1 // FRW
#define DZ_POS 18
#define DZ_NEG 16
#define Kp 0.671228158713290
#define Ki 18.927124003490206
#define Kd 0.004118042791569
#elif slave_address == 2 // RRW
#define DZ_POS 21
#define DZ_NEG 15
#define Kp 0.681854683580905
#define Ki 18.871128673401923
#define Kd 0.003935071035020
#else slave_address == 3 // RLW
#define DZ_POS 26
#define DZ_NEG 19
#define Kp 0.682141510543612
#define Ki 19.073158644532846
#define Kd 0.003876824219768
#endif

#define dcgain 0.616722919482069 // steady state gain

#define TOP 511.0 // 9 bit
uint16_t Vin = 0;

// samples collected at the a sampling rate of 0.01 second
#define freq_factor 5 // sample time 5*2ms = 0.01s, f = 100 Hz

// H bridge
#define dirPin 4
#define pwmPin 9

#define dt 0.002

volatile uint32_t t = 0, previousT; // elapsed time = t*dt
volatile uint8_t timer0_compa_count = 0;
volatile boolean wait;
```

```

Encoder we(encPinA,encPinB);
int32_t encPos;

float Ts = 0.01; // sample time
float Vcc;

// Savitzky-Golay filter
// order of polynomial fit : 2
const uint8_t window_size = 9; // 3:2:19, must be odd
float* c = (float*)malloc(sizeof(float)*window_size);
uint16_t N[] = {2,10,28,60,110,182,280,408,570}; // normalization
// queue
int32_t* q = (int32_t*)malloc(sizeof(int32_t)*window_size);

union f2b {
    byte b[4];
    float val;
} res;

union i2b {
    byte b[4];
    int32_t val;
} pos;

byte b2t[8]; // bytes to transmit

uint8_t idx;
volatile int16_t ref = 0;
float sp, y, diff, err, err1, err2, u, u1, ocr1a;

// float Kp = 2.4, Ki = 43.64, Kd = 0.033; // classic PID

void setup()
{
    Wire.begin(slave_address);

    pinMode(pwmPin,OUTPUT);
    digitalWrite(pwmPin,LOW);

    pinMode(dirPin,OUTPUT);
    digitalWrite(dirPin,HIGH); // 1:CW , 0:CCW

    TCNT1 = 0;

    TCCR1A = 0;
    TCCR1B = 0;
    TCCR1B = 0;

    // duty cycle
    OCR1A = 0;

    // Mode 2, Fpwm = 15.656 KHz
    // PWM, Phase Correct 9-bit, TOP = 0x1FF
    bitSet(TCCR1A,WGM11);
    // Clear OC1A on compare match when up-counting.
    // Set OC1A on compare match when down-counting
    bitSet(TCCR1A,COM1A1);

    // No prescaling
    bitSet(TCCR1B,CS10);

    Serial.begin(115200);
    Serial.print("Address:");
    Serial.print(slave_address);

    // register a function to be called when receives a transmission from the master
    Wire.onReceive(receiveEvent);
    // register a function to be called when a master requests data from this slave device
    Wire.onRequest(requestEvent);
}

```

```

// Calculate S-G coefficients
for(uint8_t i=0;i<window_size;i++) {
    *(c+i) = (-(window_size-1)/2.0 + i)/N[int((window_size-3)/2.0)];
    //Serial.println(c[i],15);
    *(q+i) = 0;
}

TCNT0 = 0;
TCCR0A = 0;
TCCR0B = 0;

bitSet(TCCR0A,WGM01); // CTC mode
bitSet(TCCR0B,CS02); // N = 256
OCR0A = 125-1;

bitSet(TIMSK0,OCIE0A);
}

void loop()
{
    previousT = t;

    encPos = we.read();
    pos.val = encPos;

    diff = 0;

    // right shift
    for(idx=0;idx<window_size-1;idx++)
        *(q+idx) = *(q+idx+1);

    q[idx] = encPos;

    for (idx=0;idx<window_size;idx++)
        diff += c[idx]*q[idx];

    y = (isnan(y))?0:2*PI*diff/(980*Ts); // estimated angular velocity [rad/sec]
    res.val = y;

    // create data packet that will be sent to the master
    // the first four bytes are the value of the counter (encoder measurement)
    // and the next four are the estimated angular velocity
    for (idx=0;idx<8;idx++)
        b2t[idx]=(idx<4)?pos.b[idx]:res.b[idx-4];

#ifdef OPEN_LOOP

    sp = dcgain*Vcc*ref/TOP; // desired velocity

    err2 = isnan(err1)?0:err1; // before last
    err1 = (isnan(err1))?0:err; // previous error

    err = sp - y; // current error = setpoint - actual

    u1 = (isnan(u))?0:u;
    // control signal
    u = u1 + err*(Kp+Ki*Ts/2.0+Kd/Ts)+err1*(Ki*Ts/2.0-Kp-2*Kd/Ts)+err2*(Kd/Ts); // PID

    if (u >= 0) {
        digitalWrite(dirPin,HIGH);
        ocr1a = DZ_POS + u/Vcc*(TOP-DZ_POS);
    } else {
        digitalWrite(dirPin,LOW);
        ocr1a = DZ_NEG - u/Vcc*(TOP-DZ_NEG);
    }
}

```

```

// anti-windup
if (ocr1a>TOP) {
    u = u1;
} else {
    OCR1A=(ref!=0)?ocr1a:0;
}
#else
if (ref >=0) {
    digitalWrite(dirPin,HIGH);
} else {
    digitalWrite(dirPin,LOW);
}
OCR1A = abs(ref);
#endif

/*
#if 0
Serial.print(t); Serial.print("\t");
Serial.print(ref); Serial.print("\t");
Serial.print(OCR1A); Serial.print("\t");
Serial.print(encPos); Serial.print("\t");
Serial.print(u,4); Serial.print("\t");
Serial.print(sp,4); Serial.print("\t");
Serial.println(y,4);
#else
//4 Bridge Control Panel Software : Rx8 [h=43] @1ref @0ref @3res @2res @1res @0res
Serial.print("C");
// writes binary data (as a byte) to the serial port
Serial.write(ref>>8); // hi8 : the most significant 8 bits of 16 bit integer
Serial.write(ref&0xff); // lo8 : the least significant
// response
for (int i=3;i>=0;i--)
    Serial.write(res.b[i]);
#endif
*/

wait = true;
while(wait);

Ts = (t-previousT)*dt;
}

ISR(TIMER0_COMPA_vect) {
    t++; // every 2 ms
    if(++timer0_compa_count == freq_factor) {
        timer0_compa_count = 0;
        wait = false;
    }
}

// function that executes whenever data is received from master
void receiveEvent(int numBytes) {
    while(Wire.available()) {
        if (numBytes == 4) {
            Vin = Wire.read()<<8|Wire.read(); // power supply
            Vcc = Vin/100.0; // convert integer to float
            ref = Wire.read()<<8|Wire.read(); // setpoint
        }
    }
}

void requestEvent() {
    Wire.write(b2t,8); // encoder ticks, angular velocity
}

```



```

#include "Stats.h"

Stats::Stats(int n) {
    sz = n;
    a = (float*) malloc(sz*sizeof(float));
    v = (float*) malloc(sz*sizeof(float));
    if (a == NULL || v == NULL) sz = 0;
    clear();
}

Stats::~Stats() {
    if (a!=NULL) free(a);
    if (v!=NULL) free(v);
}

void Stats::clear() {
    cnt = 0;
    idx = 0;
    s = 0;
    for (int i=0;i<sz;i++)
        *(a+i)=0.0;
}

void Stats::addValue(float f) {
    if (a == NULL) return;
    s -= a[idx];
    a[idx] = f;
    s += a[idx];
    idx++;
    if (idx == sz) idx = 0;
    if (cnt < sz) cnt++;
}

float Stats::getAvg() {
    if (cnt == 0) return NAN;
    return s/cnt;
}

// n should be between 1 and sz
void Stats::fill(float f, int n) {
    clear();
    for (int i=0; i<n; i++)
        addValue(f);
}

int float_comp(const void* a, const void* b) {
    if(*(const float*)a < *(const float*)b) return -1;
    return *(const float*)a > *(const float*)b;
}

float Stats::getMedian(void) {
    for (uint8_t i=0; i<sz; i++) *(v+i) = *(a+i); // copy
    qsort(v,sz,sizeof(float),float_comp); // quick sort

    if(sz % 2 == 0) {
        // if there is an even number of elements, return mean of the two elements in the middle
        return((v[sz/2] + v[sz/2-1])/2.0);
    } else {
        // else return the element in the middle
        return v[sz/2];
    }
}

```

```

#include "Stats.h"
// modification: internal pull-up resistors on the SCL and the SDA pins have been disabled!
#include <Wire.h>
#include <LiquidCrystal.h>

#define DEMO

#define PRINT_TXT_LCD
#define SERIAL_PRINT
#define REAL_TIME_PLOT
#define BT_DATA

#define freqPin 52 // check if the freq is 50Hz

#define VccPin A0 // voltage divider
#define panServoFdbkPin A1
#define tiltServoFdbkPin A2
#define csPin A3 // current sensor
#define shutdown_odroid 8
#define tempPin A4 // TMP36 - Analog temperature sensor
#define rstCP 32 // reset Compass
#define rstPin 33 // reset chips on the Speed Ctrl Board

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(22, 23, 24, 25, 26, 27); // RS, Enable, D4, D5, D6, D7

byte bt_icon[8] = {0x04,0x16,0x0D,0x06,0x06,0x0D,0x16,0x04}; // Bluetooth icon

#define dt 0.002 // 2 ms
#define freq_factor 10 // sample time: freq_factor*dt = 0.01 sec, f = 50Hz
float Ts = 0.02; // sample time

#define dcgain 0.616722919482069
// maximum counter value
#define TOP 511.0 // 0x01FF, 9 bit

// voltage divider
#define R1 998200.0 // 1MΩ
#define R2 97340.0 // 100KΩ
#define Vref 5.076 // AREF pin

float Vcc, previousVcc = 0, Vf, Vf1; // power supply
uint16_t V = 0;
volatile boolean updateVcc = false;

float If, If1, Amps; // current
float temperature;
volatile boolean updateTemp = false;
uint16_t I;

#define fanPWMPin 6 // 12V computer fan

#define tiltServoPWMPin 2 // Batan B2122 micro servo
#define panServoPWMPin 5
#define PAN_SERVO_MIN_PULSE_WIDTH 420 // in ms
#define PAN_SERVO_MAX_PULSE_WIDTH 2350
#define PAN_SERVO_MIN_POT_VAL 78
#define PAN_SERVO_MAX_POT_VAL 475

#define TILT_SERVO_MIN_PULSE_WIDTH 468 // in ms
#define TILT_SERVO_MAX_PULSE_WIDTH 2302
#define TILT_SERVO_MIN_POT_VAL 122
#define TILT_SERVO_MAX_POT_VAL 524

uint16_t panServoFdbkVal, tiltServoFdbkVal, desiredPotVal;

```

```

// degress = c1*pulse_width + c0
float c1 = 0.078420491128247, c0 = -35.441313543460318;
float tilt_angle = 35.0;

volatile boolean keyAvailable = false;
char key = NULL;
const uint8_t keypadPins[4] = { 28, 29, 30, 31 }; // D (MSB), C, B, A (LSB)
// keymap defines the character returned when the corresponding key is pressed
const char keymap[] = { '1', '2', '3', 'A',
                        '4', '5', '6', 'B',
                        '7', '8', '9', 'C',
                        '*', '0', '#', 'D' };

volatile uint32_t t = 0, time = 0; // t variable increments by one every 2 ms

uint32_t previousT; // previous value of the t variable
volatile uint8_t timer0_compa_cnt = 0;
volatile boolean wait, toggleFlag = 1, _100ms=false;

int slave_address; // 0:3 for ATmega328 - Speed control board
byte error;

union i2b {
    byte b[4];
    int32_t val;
} encPos[4];

union f2b {
    byte b[4];
    float val;
} y[4]; // estimated angular velocity for each wheel

int16_t avel;

int16_t r = 0, ref, init_ref, rl, rr, ref_l, ref_r, ofs = 10;
boolean fwd = true, lt = false, rt = false, inc = false, dec = false;

float v0 = 128.0;
#define v1 255
#define v2 433 // rated

int16_t V1, Vr;

enum MODE {START, STRAIGHT, INCR, DCR, LEFT_TURN, RIGHT_TURN, LEFT_SPIN, RIGHT_SPIN, STOP,
BWD, IDLE, AUTO='a', BT_DT, TRAPEZIUM, SIGNED_SIN};
MODE current_mode = STOP;

uint8_t state, previous_state;

// moving average
Stats L0_Vcc(15), L0_current(35), temp(20), Hdg(11);

volatile boolean sendBtData = false;

char* bt_data = new char[64];

volatile float out1, out2;

volatile boolean serial1event = false, serial2event = false;
boolean startup = false;

float headingOfst = 0, thisHdg = 0;
volatile float yaw;
volatile int16_t strgAng = 0;
boolean stRot = false;

```

```

void setup() {
  pinMode(rstPin,OUTPUT);
  digitalWrite(rstPin,LOW);

  pinMode(shutdown_odroid,OUTPUT);
  digitalWrite(shutdown_odroid,LOW);

  pinMode(rstCP,OUTPUT);
  digitalWrite(rstCP,LOW);

  Wire.begin(); // address is not specified, join to the I2C bus as master

  // set the data rate in bps for serial data transmission
  Serial.begin(115200); // for debugging
  Serial1.begin(9600); // ODRROID XU3
  Serial2.begin(9600); // heading
  Serial3.begin(9600); // Bluetooth module HC-06
/*
  Serial3.println("AT+NAMERover");
  delay(1000);
  Serial3.println("AT+BAUD4");
  delay(1000);
*/
  // set up the LCD's number of columns and rows
  lcd.begin(20, 4);

  lcd.createChar(0, bt_icon);
  lcd.clear();

  // 5V applied to the AREF pin and is used as the reference
  analogReference(EXTERNAL);

  pinMode(freqPin,OUTPUT);

  digitalWrite(rstPin,HIGH);
  digitalWrite(rstCP,HIGH);
  delay(3500); // wait to stabilize

  Serial.println(F("scanning I2C bus ..."));

  uint8_t nDevices = 0;
  while(1) {
    for (slave_address=0;slave_address<4;slave_address++){
      Wire.beginTransmission(slave_address);
      error = Wire.endTransmission();
      if (error==0) {
/*
          Serial.print("I2C device found at address 0x");
          if (slave_address<16)
              Serial.print("0");
          Serial.println(slave_address,HEX);
*/
          nDevices++;
        }
      }
    if (nDevices==4) {
      Serial.println("4 I2C devices detected!");
      break;
    }
    nDevices = 0;
  }

  uint16_t adc_data = analogRead(VccPin);
  L0_Vcc.fill(adc_data,15);
  Vf = adc_data;
  Vcc = Vf*Vref*(R1+R2)/(1024.0*R2);
}

```

```

adc_data = analogRead(csPin);
L0_current.fill(adc_data,35);
If = adc_data;

adc_data = (analogRead(tempPin)*Vref/1024.0-0.47)*100;;
temp.fill(adc_data,20);

for (uint8_t i=0;i<4;i++) {
    pinMode(keypadPins[i],INPUT);
}

// data available output used as an interrupt signal for the µC when there is a keystroke
attachInterrupt(1,keypadISR,FALLING);

// control fan speed with PWM
pinMode(fanPWMPin,OUTPUT);
TCNT4 = 0;
TCCR4A = 0;
TCCR4B = 0;
// Fpwm = 18.868KHz
// Fpwm = F_CPU/(2*N*TOP)
ICR4 = 53; // TOP
// mode 8: PWM, Phase and Frequency Correct
bitSet(TCCR4B,WGM43);
// Prescaler: 8
bitSet(TCCR4B,CS41);
// Clear OC4A on compare match when up-counting
// Set OC4A on compare match when down counting
bitSet(TCCR4A,COM4A1);
OCR4A = 0;

// pan/tilt camera system
// micro servo control
pinMode(tiltServoPWMPin,OUTPUT);
pinMode(panServoPWMPin,OUTPUT);
TCNT3 = 0;
TCCR3A = 0;
TCCR3B = 0;
// Fpwm = 50Hz
ICR3 = 20000;
bitSet(TCCR3B,WGM33); // mode 8: PWM, Phase and Frequency Correct
bitSet(TCCR3B,CS31); // Prescaler: 8
// Clear OC3A on compare match when up-counting.
// Set OC3A on compare match when down counting
bitSet(TCCR3A,COM3A1);
bitSet(TCCR3A,COM3B1);
OCR3A = PAN_SERVO_MAX_PULSE_WIDTH;
OCR3B = 0;

desiredPotVal = setTiltServoPWM(tilt_angle);
// Wait until it reaches the target
// tolerance: 5, max feedback measurement error
while(abs(analogRead(tiltServoFdbkPin) - desiredPotVal) > 5){}

// loop control
TCNT0 = 0;
TCCR0A = 0;
TCCR0B = 0;
bitSet(TCCR0A,WGM01); // CTC mode
bitSet(TCCR0B,CS02); // clk/256
OCR0A = 125-1;
bitSet(TIMSK0,OCIE0A);
} // setup

```

```

void loop() {
    digitalWrite(freqPin,1);
    previousT = t;

    slave_address = 0;

    do {
        Wire.beginTransmission(slave_address); // talk to slave

        switch (slave_address) {
            case 0: // flw
            case 3: // rlw
                r = ref_l;
                break;
            case 1: // frw
            case 2: // rrw
                r = ref_r;
                break;
        }

        r = constrain(r,-TOP,TOP);
        V = round(Vcc*100.0);

        // queues bytes for transmission from the master to slave device
        Wire.write((byte)(V>>8)); // power supply
        Wire.write((byte)(V&0xff));
        Wire.write((byte)(r>>8)); // reference signal
        Wire.write((byte)(r&0xff));
        error = Wire.endTransmission(true); // leave i2c bus

        if(error!=0) {
            onerror(error);
        } else {

            // estimated angular velocity is send by slave
            Wire.requestFrom(slave_address,8); // 4:float
            while(Wire.available()) {

                for(uint8_t i=0;i<8;i++) {
                    if (i<4)
                        encPos[slave_address].b[i] = Wire.read();

                    else
                        y[slave_address].b[i-4] = Wire.read();

                }

            }
            slave_address++;
        }
    } while(slave_address<4 && error==0);
}

```

```

if (error==0) { // 0: success
    switch (current_mode) {
        case START: {
            r = 80.0*t*dt;

            if (r > v0) r = v0;
            if (!fwd) r*=-1;

            ref_l = r;
            ref_r = r;

            break;
        }
        case STOP: {
            // firstly make velocities in each side equal if they are not
            if (ref_l!=0 || ref_r !=0) {
                if (abs(ref_l)!=abs(ref_r))
                    doVelEq(ref_l,ref_r);
            }
            else {
                if(ref_l>0 && ref_l>0) {
                    ref_l-=2; ref_r-=2;
                } else if(ref_l<0 && ref_r<0) {
                    ref_l+=2; ref_r+=2;
                } else if (ref_l>0 && ref_r<0) {
                    ref_l-=2; ref_r+=2;
                } else if (ref_l<0 && ref_r>0) {
                    ref_l+=2; ref_r-=2;
                }
            }
            } else {
                ref_l=0; ref_r=0;
            }
            }
            if (abs(ref_l)<6||abs(ref_r)<6) {
                ref_l = 0;
                ref_r = 0;
            }
            break;
        }
        case STRAIGHT: {
            if (state == 1 || state == 2) {
                doVelEq(ref_l,ref_r);
                if (abs(ref_l) == abs(ref_r))
                    current_mode = IDLE;
            } else if (state == 3) {
                if (previous_state == 1) {
                    if(ref_l<=ref_r) ref_l+=2;
                } else if (previous_state == 2) {
                    if (ref_l<=ref_r) ref_r-=2;
                }
            } else if (state == 4) {
                if (previous_state == 1) {
                    if (ref_l>=ref_r) ref_r+=2;
                } else if (previous_state == 2) {
                    if (ref_l>=ref_r) ref_l-=2;
                }
            }
            }
            break;
        }
    }
}

```

```

case LEFT_TURN: {
    static uint8_t step = 0;
    if (lt) {
        if (ref_l>0 && ref_r>0) {
            ref_r+=1;
        }
        else if (ref_l<0 && ref_r<0) {
            ref_r-=1;
        }
        if (++step>ofs) {
            lt = false;
            step = 0;
        }
    }
    break;
}

case RIGHT_TURN: {
    static uint8_t step = 0;
    if (rt) {
        if (ref_l>0 && ref_r>0) {
            ref_l+=1;
        }
        else if (ref_l<0 && ref_r<0) {
            ref_l-=1;
        }
        if (++step>ofs) {
            rt = false;
            step = 0;
        }
    }
    break;
}

case INCR: {
    static uint8_t step = 0;
    if (inc) {
        if (ref_l<0 && ref_r>0) { // left spin
            ref_l-=1;
            ref_r+=1;
        } else if (ref_l>0 && ref_r<0) { // right spin
            ref_l+=1;
            ref_r-=1;
        } else if (ref_l>0 && ref_r>0) { // forward
            ref_l+=1;
            ref_r+=1;
        } else if (ref_l<0 && ref_r<0) { // backwards
            ref_l-=1;
            ref_r-=1;
        }
        if (++step > ofs) {
            inc = false;
            step = 0;
        }
    }
    break;
}

```



```

case DCR: {
    static uint8_t step = 0;
    if (dec) {
        if (ref_l<0 && ref_r>0) { // left spin
            ref_l+=1;
            ref_r-=1;
        } else if (ref_l>0 && ref_r<0) { // right spin
            ref_l-=1;
            ref_r+=1;
        } else if (ref_l>0 && ref_r>0) { // forward
            ref_l-=1;
            ref_r-=1;
        } else if (ref_l<0 && ref_r<0) { // backwards
            ref_r+=1;
            ref_l+=1;
        }
        if(++step > ofs) {
            dec = false;
            step = 0;
        }
    }
    break;
}

case LEFT_SPIN: {
    static uint8_t eq = 0;

    if (eq==0) {
        if (abs(ref_l)!=abs(ref_r)) {
            doVelEq(ref_l,ref_r);
        } else {
            eq = 1;
            init_ref = abs(ref_l);
        }
    } else {
        if (ref_l>-init_ref) ref_l-=2;
        if (ref_r<init_ref) ref_r+=2;
        if ((ref_l<=-init_ref) && (ref_r>=init_ref)) {
            eq = 0;
            current_mode = IDLE;
        }
    }

    break;
}

case RIGHT_SPIN: {
    static uint8_t _eq = 0;

    if (_eq==0) {
        if (abs(ref_l)!=abs(ref_r)) {
            doVelEq(ref_l,ref_r);
        } else {
            _eq = 1;
            init_ref = abs(ref_l);
        }
    } else {
        if (ref_l<init_ref) ref_l+=2;
        if (ref_r>-init_ref) ref_r-=2;
        if ((ref_l>=init_ref) && (ref_r<=-init_ref)) {
            _eq = 0;
            current_mode = IDLE;
        }
    }

    break;
}

```

```

case AUTO: {
    static boolean lhs, rhs;
    if (startup) {
        lhs = false;
        rhs = false;
    }
    static uint8_t halt = 0;
    if (serial1event) {
        if (out1 == 0 && out2 == 0) {
            halt++;
            if (halt >= 50) {
                halt = 0;
                current_mode = STOP;
            }
        } else {
            Serial.print(out1,4); Serial.print("\t");
            Serial.println(out2,4);
            halt = 0;
            Vl = round(v0*out1);
            Vr = round(v0*out2);

            r = 50*t*dt;

            if (startup) {
                if (r>Vl)
                    lhs = true;
                if (r>Vr)
                    rhs = true;
                if (lhs && rhs)
                    startup = false;
            }

            ref_l = (lhs) ? Vl:r;
            ref_r = (rhs) ? Vr:r;
        }
        serial1event = false;
    }
    break;
}

// modes for demonstration

#ifndef DEMO
case TRAPEZIUM: {

    // count is increased every 1.5 sec
    static unsigned int count = 0;
    static boolean flag = toggleFlag;
    if (flag == toggleFlag) {
        count++;
        flag^=1;
    }
    if (count<= 2)
        ref = round(v0/3.0*dt*(t%1500));
    else if (count <= 4)
        ref = v0;
    else if (count <= 6)
        ref = round(-v0/3.0*((t%4500)*dt-9.0));
    else
        count = 1;

    ref_l = ref;
    ref_r = ref;

    break;
}
}

```

```

        case SIGNED_SIN: {
            ref = round(v0*sin(2*PI/6.0*t*dt));
            ref_l = ref; ref_r = ref;

            break;
        }

    #endif // DEMO

} // mode

ref_l = constrain(ref_l,-TOP,TOP);
ref_r = constrain(ref_r,-TOP,TOP);

} // error = 0, success

if(strgAng != 0) {
    uint8_t _state = getState(ref_r,ref_r);
    if (_state != 3 && _state != 4)
        previous_state = _state;

    thisHdg = abs(yaw)+abs(strgAng);
    if (strgAng>0)
        current_mode = LEFT_SPIN;
    else
        current_mode = RIGHT_SPIN;
    strgAng = 0;
    stRot = true;
}

if(abs(yaw) >= thisHdg && stRot) {
    state = getState(ref_r,ref_r);
    current_mode = STRAIGHT;
    stRot = false;
}

if (keyAvailable) {
    key = getKey();
    switch(key) {
        case 'A': { // start, stop
            if (ref_l!=0 || ref_r!=0) {
                current_mode = STOP;
            } else {
                current_mode = START;
                t = 0;
                fwd = true;
            }
            break;
        }

        case '1': {
            if (ref_l == 0 && ref_r == 0) {
                t = 0;
                current_mode = TRAPEZIUM; // demo
            }
            break;
        }

        case 'B': {
            if (ref_l == 0 && ref_r == 0) {
                setSpeed();
            }
            break;
        }
    }
}

```

```

        case 'C': {
            if (ref_l == 0 && ref_r == 0) {
                t = 0;
                startup = true;
                current_mode = AUTO;
            }
            break;
        }
        case 'D': {
            if (ref_l == 0 && ref_r == 0) {
                setTiltAngle();
            }
            break;
        }
        case '*': {
            if (ref_l == 0 && ref_r == 0) {
                digitalWrite(rstCP,LOW);
                digitalWrite(rstPin,LOW);
                uint32_t timeCurrent = t;
                while((t-timeCurrent)<=15); // delay
                digitalWrite(rstCP,HIGH);
                digitalWrite(rstPin,HIGH);
                headingOfst = 0;
            }
            break;
        }
        case '0': {
            if (ref_l == 0 && ref_r == 0) {
                digitalWrite(shutdown_odroid,HIGH);
                uint32_t timeCurrent = t;
                while((t-timeCurrent)<=50);
                digitalWrite(shutdown_odroid,LOW);
            }
            break;
        }
    } // switch
} // keyAvailable

if (_100ms) {
    panServoFdbkVal = analogRead(panServoFdbkPin);
    _100ms = false;
}

if (ref_l>0 && ref_r>0) {
    fwd = true;
    if (panServoFdbkVal <= PAN_SERVO_MAX_POT_VAL)
        OCR3A = PAN_SERVO_MAX_PULSE_WIDTH;
} else if (ref_l<0 && ref_r<0) {
    fwd = false; // backwards
    if (panServoFdbkVal >= PAN_SERVO_MIN_POT_VAL)
        OCR3A = PAN_SERVO_MIN_PULSE_WIDTH;
}

if(updateTemp) {
    temperature = (analogRead(tempPin)*Vref/1024.0-0.47)*100;
    // Serial.println(temperature);
    temp.addValue(temperature);
    temperature = temp.getAvg();
}

```

```

        // update display with new value of the temperature and change fan speed
        if (temperature >=20)
            OCR4A = map(temperature,20,40,11,53); // tempMin = 20, tempMax = 40°C
        else
            OCR4A = 0;

        #ifdef PRINT_TXT_LCD
            lcd.setCursor(8,0);
            lcd.print(temperature,1); lcd.write(178); // degree sign
            lcd.print("C"); lcd.write(32); lcd.write(32);
        #endif

        updateTemp = false;
    }

    //
    if(serial2event) {
        Serial.println(yaw);
        serial2event = false;

        #ifdef PRINT_TXT_LCD
            lcd.setCursor(8,1);
            lcd.print(yaw,1);
            lcd.write(178); lcd.write(32); lcd.write(32);
        #endif
    }

    L0_current.addValue(analogRead(csPin));
    If1 = If; // store last value
    If = If1 +1/11.0*(L0_current.getAvg()-If1);
    Amps = If/20.0-25;

    if (updateVcc) {
        L0_Vcc.addValue(analogRead(VccPin)); // sma
        // exponential moving average
        Vf1 = Vf; // previous value
        Vf = Vf1 + 0.125*(L0_Vcc.getAvg()-Vf1); // alpha = 1/8
        Vcc = Vf*Vref*(R1+R2)/(1024.0*R2);
        updateVcc = false;
    }

    #ifdef PRINT_TXT_LCD
        lcd.setCursor(0,1);
        lcd.print(Amps,1); lcd.print("A"); lcd.write(32);

        if (previousVcc != Vcc) { // update display with new value of the Vcc
            lcd.setCursor(0,0); lcd.print(Vcc,1); lcd.print("V");
            if (floor(Vcc/10.0) == 0) // Vcc < 10V
                lcd.write(32); // space character
            previousVcc = Vcc;
        }
    #endif

    #ifdef SERIAL_PRINT
        #ifndef REAL_TIME_PLOT
            Serial.print(t); Serial.print("\t");

            Serial.print(Vcc,2); Serial.print("\t");
            Serial.print(Amps,2); Serial.print("\t");

            Serial.print(ref_l); Serial.print("\t");
            Serial.print(ref_r); Serial.print("\t");
            // response
            Serial.print(y[0].val,2); Serial.print("\t"); // lhs
            Serial.print(y[3].val,2); Serial.print("\t");
            Serial.print(y[1].val,2); Serial.print("\t"); // rhs
            Serial.println(y[2].val,2);
        #endif
    #endif

```

```

else // real time plotting with Bridge Control Panel Software
// command: Rx8 [h=43] @1ref_l @0ref_l @1ref_r @0ref_r
                        @3flw @2flw @1flw @0flw
                        @3frw @2frw @1frw @0frw
                        @3rrw @2rrw @1rrw @0rrw
                        @3rlw @2rlw @1rlw @0rlw
    Serial.print("C");
    // writes binary data (as a byte) to the serial port
    r1 = (dcgain*Vcc*ref_l/TOP)*100.0;
    // hi8 : the most significant 8 bits of 16 bit integer
    Serial.write(r1>>8);
    Serial.write(r1&0xff); // lo8 : the least significant

    rr = (dcgain*Vcc*ref_r/TOP)*100.0;
    Serial.write(rr>>8);
    Serial.write(rr&0xff);
    for (slave_address=0;slave_address<4;slave_address++) { // response
        for (int i=3;i>=0;i--)
            Serial.write(y[slave_address].b[i]);
    }
    I = round(Amps*100.0);
    Serial.write(I>>8);
    Serial.write(I&0xff);
#endif
#endif

#ifdef BT_DATA
    if (sendBtData) { // perform bluetooth data transfer

        r1 = (int16_t)round((dcgain*Vcc*ref_l/TOP)*100.0);
        rr = (int16_t)round((dcgain*Vcc*ref_r/TOP)*100.0);

        // 0: flw, 3: rlw ,1: frw, 2: rrw
        // vector elements: time, refL, refR, FLW, RLW, FRW, RRW, Vcc, Amps, yaw
        sprintf(bt_data,"%ld %d %d %d %d %d %d %d %d %d",

time,r1,rr,(int16_t)round(y[0].val*100.0),(int16_t)round(y[3].val*100.0),
(int16_t)round(y[1].val*100.0),(int16_t)round(y[2].val*100.0),
(int16_t)round(Vcc*100.0),(int16_t)round(Amps*100.0),
(int16_t)round(yaw*100.0));
        Serial3.println(bt_data);
    }
#endif

    wait = true;
    while(wait);

    Ts = (t-previousT)*dt;
    // Serial.println(Ts);

    digitalWrite(freqPin,0);
} // loop

ISR(TIMER0_COMPA_vect) { // loop control ISR
    t++; // every 2 ms
    if(++timer0_compa_cnt == freq_factor) { // 50 Hz
        timer0_compa_cnt = 0;
        wait = false;
        time = t;
    }

    if(t % 50 == 0) { // get feedback from the micro servo, 10Hz
        _100ms = true;
    }

    if(t % 125 == 0) { // read battery level every 0.25 sec, 4Hz
        updateVcc = true;
    }
}

```

```

    if(t % 500 == 0) { // every sec
        updateTemp = true;
    }

#ifdef DEMO
    if (t % 750 == 0) toggleFlag ^= 1; // every 1.5 sec
#endif
}

void serialEvent1() { // Odroid-XU3
    // 2*int16_t + space + '\r', returning values from the Odroid XU3
    byte* buffer = (byte*) malloc(6);
    while(Serial1.available()) {
        for (uint8_t idx=0;idx<6;idx++) {
            *(buffer+idx) = Serial1.read();
        }
        if (buffer[0] == 0x20 && buffer[5] == '\r') {
            uint16_t val = buffer[2]<<8|buffer[1];
            // fuzzy output 1, multiplier of the velocity of the left wheels
            out1 = val/10000.0;
            val = buffer[4]<<8|buffer[3];
            // fuzzy output 2, multiplier of the velocity of the right wheels
            out2 = val/10000.0;
            if (out1<=1 && out2<=1)
                serial1event = true;
        }
    }
    free(buffer);
}

void serialEvent2() { // yaw angle estimation from MPU6050 + HMC8553L
    byte* buffer = (byte*) malloc(3); // int16_t + '\n'
    while(Serial2.available()) {
        for (uint8_t idx=0;idx<3;idx++) {
            *(buffer+idx) = Serial2.read();
        }
        if (buffer[2] == '\n') {

            int16_t angle = (byte)buffer[1]<<8|(byte)buffer[0];
            yaw = angle/100.0;
            if (headingOfst == 0) {
                headingOfst = yaw;
                Serial.println(headingOfst);
            }

            yaw -= headingOfst;

            if (yaw < -180) yaw += 360;
            else if (yaw > 180) yaw-=360;

            yaw *= -1;

            Hdg.addValue(yaw);

            yaw = Hdg.getMedian();

            serial2event = true;
        }
    }
    free(buffer);
}
}

```

```

void serialEvent3() { // Bluetooth HC-06

    if(Serial3.available()>0) {
        byte incomingByte = Serial3.read();

        if (incomingByte == '+' || incomingByte == '-') {
            strgAng = 0;
            int8_t sign = 1;
            if (incomingByte == '-') sign = -1;
            while(Serial3.available()) {
                strgAng = strgAng*10 + (Serial3.read()-'0');
            }
            Serial.println(strgAng*sign);
            return;
        }

        if (incomingByte>='0' && incomingByte <='9')
            incomingByte -= '0';

        if (ref_l !=0 && ref_r !=0 && current_mode == AUTO && incomingByte != 0)
            return;

        switch(incomingByte) {
            case 0:

                if (ref_l!=0 || ref_r!=0) {
                    current_mode = STOP;
                } else {
                    current_mode = START;
                    t = 0;
                    fwd = true;
                }

                break;

            case INCR:
                current_mode = INCR;
                inc = true;
                break;

            case DCR:
                current_mode = DCR;
                dec = true;
                break;

            case LEFT_TURN:
                current_mode = LEFT_TURN;
                lt = true;
                break;

            case RIGHT_TURN:
                current_mode = RIGHT_TURN;
                rt = true;
                break;

            case STRAIGHT:
                state = getState(ref_r,ref_r);
                current_mode = STRAIGHT;
                break;

            case LEFT_SPIN:
                current_mode = LEFT_SPIN;
                break;

            case RIGHT_SPIN:
                current_mode = RIGHT_SPIN;
                break;
        }
    }
}

```



```

        case BWD:
            if (current_mode == STOP) {
                t = 0;
                fwd = false;
                current_mode = START;
            }
            break;

        case AUTO:
            if (ref_l == 0 && ref_r == 0) {
                t = 0;
                startup = true;
                current_mode = AUTO;
            }
            break;

        case BT_DT: // send data to Matlab workspace

            sendBtData = !sendBtData;
            sendBtData ? Serial.println(F("Bluetooth data transfer is on")):
            Serial.println(F("Bluetooth data transfer is off"));
            break;

            } // switch
    } // available

    if (current_mode == LEFT_SPIN || current_mode == RIGHT_SPIN) {
        uint8_t _state = getState(ref_r,ref_r);
        if (_state != 3 && _state != 4)
            previous_state = _state;
    }
}

void keypadISR() {
    keyAvailable = true;
}

// a keystroke is detected
char getKey() {
    keyAvailable = false;
    uint8_t num = 0; // nibble
    for (uint8_t i=0;i<3;i++) {
        num |= digitalRead(keypadPins[i]);
        num <<= 1;
    }
    num |= digitalRead(keypadPins[3]);

    // encoded nibble are mapped to the key names through a lookup table
    return keymap[num];
}

uint16_t setTiltServoPWM(uint8_t deg) {
    uint16_t pw = (uint16_t)(1/c1*deg - c0/c1); // pulse width in µs
    OCR3B = constrain(pw,TILT_SERVO_MIN_PULSE_WIDTH,TILT_SERVO_MAX_PULSE_WIDTH);
    // Calculate the target feedback value for the final position
    uint16_t target = map(OCR3B, TILT_SERVO_MIN_PULSE_WIDTH , TILT_SERVO_MAX_PULSE_WIDTH,
TILT_SERVO_MIN_POT_VAL,TILT_SERVO_MAX_POT_VAL);
    return target;
}

```

```

// This function establishes the feedback values for 2 positions of the servo
// With this information, we can interpolate feedback values for intermediate positions
void microServoCalibration(uint16_t minPW, uint16_t maxPW) {
    // Move to the minimum position and record the feedback value
    OCR3A = minPW; // or OCR3B
    delay(1500);
    panServoFdbkVal = analogRead(panServoFdbkPin);
    Serial.println(panServoFdbkVal);
    // Move to the maximum position and record the feedback value
    OCR3A = maxPW;
    delay(1500);
    panServoFdbkVal = analogRead(panServoFdbkPin);
    Serial.println(panServoFdbkVal);
}

void doVelEq(int16_t& _r1, int16_t& _rr) {
    if (abs(_r1) == abs(_rr)) return;

    if (_r1>=0 && _rr>=0) { // forward
        (_r1>_rr)?_r1-=1:_rr-=2;
    }
    else if (_r1<=0 && _rr<=0) { // backwards
        (_r1<_rr)?_r1+=1:_rr+=2;
    }
    else if (_r1>=0 && _rr<=0) { // right spin
        (_r1>_-rr)?_r1-=1:_rr+=2;
    }
    else if (_r1<=0 && _rr>=0) { // left spin
        (-_r1>_rr)?_r1+=1:_rr-=2;
    }
}

uint8_t getState(int16_t _r1,int16_t _rr) {
    uint8_t rval = 0;
    if (ref_l>0 && ref_r>0)
        rval = 1;
    else if (ref_l<0 && ref_r<0)
        rval = 2;
    else if (ref_l<0 && ref_r>0)
        rval = 3;
    else if (ref_l>0 && ref_r<0)
        rval = 4;
    return rval;
}

void setSpeed() {
    lcd.clear();
    String txt = "CAV [0-15rad/s]:";
    float omega = dataEntry(txt,0);
    uint16_t dc = round(omega*TOP/(dcgain*Vcc));
    if (dc >=0 && dc <= TOP)
        v0 = dc;
    lcd.clear();
}

void setTiltAngle() {
    lcd.clear();
    String txt = "Camera Tilt:";
    tilt_angle = dataEntry(txt,0);
    // if tilt_angle = 0 then the camera will be on horizontal position
    tilt_angle = 80 - tilt_angle;
    if (tilt_angle >= 35 && tilt_angle <= 80) {
        desiredPotVal = setTiltServoPWM(tilt_angle);
        while(abs(analogRead(tiltServoFdbkPin) - desiredPotVal) > 5){}
    }
    lcd.clear();
}
}

```

```

// number's entry via the keypad
// txt: the string that is printed on the LCD
// row: the row on which the cursor is positioned
float dataEntry(String txt, uint8_t row) {
    int i;
    boolean ok = false;
    byte num[5];
    uint8_t p = 0, dot = 0;
    float f;
    lcd.setCursor(0,row);
    lcd.print(txt);
    uint8_t k = txt.length();
    for (i=0;i<sizeof(num);i++) *(num+i)=0;
    while(1) {
        if(keyAvailable) {
            key = getKey();
            Serial.println(key);
            if (key>='0' && key <='9') {
                if (p<sizeof(num)) {
                    lcd.print(key);
                    key -= '0';
                    num[p] = key;
                    p++;
                }
            } else if (key == '.' && p > 0 && p <sizeof(num - 1) && dot == 0) {
                lcd.print(".");
                dot = p;
            } else if(key=='#') { // enter
                if (dot == 0) {
                    f = 0;
                    float e = 1;
                    for(i=p-1;i>=0;i--) {
                        f += num[i]*e;
                        e *= 10.0;
                    }
                } else {
                    f = 0;
                    float e = 1;
                    for(i=dot-1;i>=0;i--) {
                        f+=num[i]*e;
                        e *= 10.0;
                    }
                    e = 1;
                    for(i=dot;i<p;i++) {
                        e *= 1/10.0;
                        f+=num[i]*e;
                    }
                }
                ok = true;
            } else if (key == 'C') { // clear
                lcd.setCursor(k,row);
                for (uint8_t c = k; c<=19 ; c++)
                    lcd.write(32);
                lcd.setCursor(k,row);
                for (i=0;i<sizeof(num);i++)
                    *(num+i)=0;
                dot = 0;
                p = 0;
            }
        } // if(keyAvailable)
        if (ok) {
            Serial.println(f,4);
            break;
        }
    } // while
    previousVcc = 0;
    Serial.println(f);
    return f;
}

```

```

void onerror(uint8_t n) { // I2C error handling
    switch(n) {
        case 1:
            Serial.println(F("data too long to fit in transmit buffer"));
            break;
        case 2:
            Serial.println(F("received NACK on transmit of address"));
            break;
        case 3:
            Serial.println(F("received NACK on transmit of data "));
            break;
        case 4:
            Serial.println(F("other error"));
            break;
    }
}

```

Firmware εγκατεστημένο στον μικροελεγκτή ATmega328 ο οποίος είναι επιφορτισμένος με τον υπολογισμό του προσανατολισμού (yaw angle estimation)

```

#include <Wire.h>

#define SMPLRT_DIV 0x19
#define PWR_MGMT_1 0x6B
#define WHO_AM_I 0x75
#define ACCEL_XOUT_H 0x3B

#define CONFIG_REG_A 0x00
#define MODE_REG 0x02
#define MAG_XOUT_H 0x03

#define MAG0MAX 351
#define MAG0MIN -350
#define MAG1MAX 216
#define MAG1MIN -597
#define MAG2MAX 333
#define MAG2MIN -357

const uint8_t IMUAddr = 0x68;
const uint8_t MagAddr = 0x1E; // address of magnetometer

const uint16_t I2C_TIMEOUT = 1000;

// IMU raw data
float aX, aY, aZ;
float gX, gY, gZ;
int16_t temp;
float mX, mY, mZ;
// roll and pitch are calculated using the accelerometer
// while yaw is calculated using the magnetometer
float roll, pitch, yaw;

float phi, theta, psi; // calculated angle using a Kalman filter

float prevHdg = 0;

```

```

uint32_t t;
uint8_t i2cData[14]; // buffer for I2C data

float magOfst[3] = { (MAG0MAX + MAG0MIN) / 2, (MAG1MAX + MAG1MIN) / 2, (MAG2MAX + MAG2MIN) / 2
};
float magGain[3];

union i2b {
    byte b[2];
    int16_t val;
} heading;

// I2C
uint8_t i2cWrite(uint8_t address, uint8_t registerAddress, uint8_t data, bool sendStop) {
    return i2cWrite(address, registerAddress, &data, 1, sendStop); // returns 0 on success
}

uint8_t i2cWrite(uint8_t address, uint8_t registerAddress, uint8_t *data, uint8_t length, bool
sendStop) {
    Wire.beginTransmission(address);
    Wire.write(registerAddress);
    Wire.write(data, length);
    uint8_t rcode = Wire.endTransmission(sendStop); // returns 0 on success
    if (rcode) {
        Serial.print(F("i2cWrite failed: "));
        Serial.println(rcode);
    }
    return rcode;
}

uint8_t i2cRead(uint8_t address, uint8_t registerAddress, uint8_t *data, uint8_t nbytes) {
    uint32_t ms;
    Wire.beginTransmission(address);
    Wire.write(registerAddress);
    uint8_t rcode = Wire.endTransmission(false); // don't release the bus
    if (rcode) {
        Serial.print(F("i2cRead failed: "));
        Serial.println(rcode);
        return rcode;
    }
    // Send a repeated start and then release the bus after reading
    Wire.requestFrom(address, nbytes, (uint8_t)true);
    for (uint8_t i = 0; i < nbytes; i++) {
        if (Wire.available())
            data[i] = Wire.read();
        else {
            ms = micros();
            while (((micros() - ms) < I2C_TIMEOUT) && !Wire.available());
            if (Wire.available())
                data[i] = Wire.read();
            else {
                Serial.println(F("i2cRead timeout"));
                return 5;
            }
        }
    }
    return 0; // success
}

```

```

// b represents the gyro bias/drift, it corrects the measured rotation rate

// state vector x = [0;b]

// A = [1 -Δt;
//      0  1]

// control input: the gyroscope measurement in °/s (rate)
// B = [Δt;0]

// process noise is Gaussian distributed with a zero mean and with covariance Q

// the measurement is just the measurement from the accelerometer
// H = [1,0]

// noise of the measurement have to be Gaussian distributed as well
// with zero mean and R as the covariance
// is equal to the variance of the measurement,
// since the covariance of the same variable is equal to the variance

// if you set the measurement noise variance too high the filter will respond slowly
// as it is trusting new measurements less,
// but if it is too small we trust accelerometer measurement too much

class Kalman {
public:
    // constructor
    Kalman() {

        Q_angle = 0.001;
        Q_bias = 0.003;
        R = 0.03;

        angle = 0; // reset the angle
        bias = 0; // reset bias

        // since we assume that the bias is 0 and we know the starting angle (use setAngle),
        // the error covariance matrix is set like so
        P[0][0] = 0; P[0][1] = 0; P[1][0] = 0; P[1][1] = 0;
    };

    // angle in degrees, rate in degrees per second
    float getAngle(float newAngle, float newRate, float dt) {
        // prediction
        rate = newRate - bias;
        angle += rate*dt;

        // state uncertainty estimate increases due to prediction
        P[0][0] += (P[1][1]*dt-P[0][1]-P[1][0]+Q_angle)*dt;
        P[0][1] -= P[1][1]*dt;
        P[1][0] -= P[1][1]*dt;
        P[1][1] += Q_bias*dt;

        // correction
        S = P[0][0] + R;
        K[0] = P[0][0] / S;
        K[1] = P[1][0] / S;

        y = newAngle - angle;
        // new estimate incorporating the measurement
        angle += K[0]*y;
        bias += K[1]*y;
        // calculate estimation error covariance - update the error covariance
        // state uncertainty estimate decreases due to correction
        P[0][0] -= K[0]*P[0][0];
        P[0][1] -= K[0]*P[0][1];
        P[1][0] -= K[1]*P[0][0];
        P[1][1] -= K[1]*P[0][1];
        return angle;
    };

    // Used to set angle, this should be set as the starting angle
    void setAngle(float newAngle) { angle = newAngle; };
};

```

```

private:
    // process noise variance
    float Q_angle; // process noise variance for the accelerometer
    float Q_bias; // process noise variance for the gyro bias
    float R; // variance of the measurement noise

    float angle; // The angle calculated by the Kalman filter - part of the 2x1 state vector
    // The gyro bias calculated by the Kalman filter - part of the 2x1 state vector
    float bias;
    // unbiased rate calculated from the rate and the calculated bias
    // you have to call getAngle to update the rate
    float rate;
    // error covariance matrix (a measure of the estimated accuracy of the state estimate)
    // This is a 2x2 matrix
    float P[2][2];
    float K[2]; // Kalman gain, the weight to give to the measurement - This is a 2x1 vector
    float y; // difference between actual and expected measurement
    float S; // estimate error
};

Kalman AngXEst, AngYEst, AngZEst; // create the Kalman instances

void updateIMU() {
    while (i2cRead(IMUAddr, ACCEL_XOUT_H, i2cData, 14)); // get accelerometer and
    gyroscope values
    aX = ((i2cData[0] << 8) | i2cData[1]);
    aY = -((i2cData[2] << 8) | i2cData[3]);
    aZ = ((i2cData[4] << 8) | i2cData[5]);
    temp = (i2cData[6] << 8) | i2cData[7];
    gX = -(i2cData[8] << 8) | i2cData[9];
    gY = (i2cData[10] << 8) | i2cData[11];
    gZ = -(i2cData[12] << 8) | i2cData[13];
}

void updateMag() {
    while (i2cRead(MagAddr, MAG_XOUT_H, i2cData, 6)); // get magnetometer values
    mX = ((i2cData[0] << 8) | i2cData[1]);
    mZ = ((i2cData[2] << 8) | i2cData[3]);
    mY = ((i2cData[4] << 8) | i2cData[5]);
}

void updatePitchRoll() {
    // atan2 outputs the value of -π to π (radians)
    roll = atan2(aY, aZ) * RAD_TO_DEG;
    pitch = atan(-aX / sqrt(aY * aY + aZ * aZ)) * RAD_TO_DEG;
}

void updateYaw() {
    // invert axis - this it done here, as it should be done after the calibration
    mX *= -1;
    mZ *= -1;

    mX *= magGain[0];
    mY *= magGain[1];
    mZ *= magGain[2];
    mX -= magOfst[0];
    mY -= magOfst[1];
    mZ -= magOfst[2];

    float rollAngle = phi * DEG_TO_RAD;
    float pitchAngle = theta * DEG_TO_RAD;

    float Bfy = mZ * sin(rollAngle) - mY * cos(rollAngle);
    float Bfx = mX * cos(pitchAngle) + mY * sin(pitchAngle) * sin(rollAngle) + mZ *
    sin(pitchAngle) * cos(rollAngle);
    yaw = atan2(-Bfy, Bfx) * RAD_TO_DEG;
    yaw *= -1;
}

void calibrateMag() {
    i2cWrite(MagAddr, CONFIG_REG_A, 0b00010001, true);
    delay(100); // wait for sensor to get ready
    updateMag(); // read positive bias values
    int16_t magPosOfst[3] = { mX, mY, mZ };
    i2cWrite(MagAddr, CONFIG_REG_A, 0b00010010, true);
}

```

```

delay(100); // wait for sensor to get ready
updateMag(); // read negative bias values
int16_t magNegOfst[3] = { mX, mY, mZ };

i2cWrite(MagAddr, CONFIG_REG_A, 0b00010000, true); // back to normal
magGain[0] = -2500 / float(magNegOfst[0] - magPosOfst[0]);
magGain[1] = -2500 / float(magNegOfst[1] - magPosOfst[1]);
magGain[2] = -2500 / float(magNegOfst[2] - magPosOfst[2]);

#if 0
    Serial.print("Mag cal: ");
    Serial.print(magNegOfst[0] - magPosOfst[0]);
    Serial.print(",");
    Serial.print(magNegOfst[1] - magPosOfst[1]);
    Serial.print(",");
    Serial.println(magNegOfst[2] - magPosOfst[2]);

    Serial.print("Gain: ");
    Serial.print(magGain[0]);
    Serial.print(",");
    Serial.print(magGain[1]);
    Serial.print(",");
    Serial.println(magGain[2]);
#endif
}

void setup() {
    delay(100); // Wait for sensors to get ready
    Serial.begin(9600);
    Wire.begin();
    TWBR = ((F_CPU / 400000L) - 16) / 2; // Set I2C frequency to 400kHz

    i2cData[0] = 7; // gyro output rate = 8KHz, sample rate = 8KHz/(1+7) = 1KHz
    // disable Frame Synchronization, set Digital Low Pass Filter Accel: 260Hz Gyro: 256Hz
    i2cData[1] = 0;
    i2cData[2] = 0; // set gyro full scale range ±250 °/s
    i2cData[3] = 0; // set accel full scale range ±2g
    // write to all four registers at once
    while (i2cWrite(IMUAddr, SMPLRT_DIV, i2cData, 4, false));
    // PLL with X axis gyroscope reference, disable sleep mode
    while (i2cWrite(IMUAddr, PWR_MGMT_1, 0x01, true));
    while (i2cRead(IMUAddr, WHO_AM_I, i2cData, 1));
    if (i2cData[0] != IMUAddr) {
        Serial.print(F("error reading sensor GY-521"));
        while (true);
    }
    // configure device for continuous- measurement mode
    while (i2cWrite(MagAddr, MODE_REG, 0, true));
    calibrateMag();
    delay(100); // wait for sensors to stabilize
    // set Kalman and gyro starting angle
    updateIMU();
    updateMag();

    updatePitchRoll();
    updateYaw();
    AngXEst.setAngle(roll);
    AngYEst.setAngle(pitch);
    AngZEst.setAngle(yaw);
    t = micros(); // initialize the timer
}

```



```

void loop() {
  // update all the IMU values
  updateIMU();
  updateMag();
  float dt = (float)(micros() - t) / 1000000;
  t = micros();
  // roll and pitch estimation
  updatePitchRoll();
  float gXrate = gX / 131.0; // Convert to °/s
  float gYrate = gY / 131.0;
  float gZrate = gZ / 131.0;
  // this fixes the transition problem when the acc. angle jumps between -180° and 180°
  if ((roll < -90 && phi > 90) || (roll > 90 && phi < -90)) {
    AngXEst.setAngle(roll);
    phi = roll;
  } else
    // calculate the angle using a Kalman filter
    phi = AngXEst.getAngle(roll, gXrate, dt);
  // invert rate, so it fits the restricted accelerometer reading
  if (abs(phi) > 90) gYrate = -gYrate;
  theta = AngYEst.getAngle(pitch, gYrate, dt);
  // yaw estimation
  updateYaw();
  prevHdg = psi;
  // this fixes the transition problem when the yaw angle jumps between -180° and 180°
  if ((yaw < -90 && psi > 90) || (yaw > 90 && psi < -90)) {
    AngZEst.setAngle(yaw);
    psi = yaw;
  } else
    // Calculate the angle using a Kalman filter
    psi = AngZEst.getAngle(yaw, gZrate, dt);

  if (prevHdg != psi) {
    heading.val = round(psi*100.0);
    Serial.write(heading.b[0]);
    Serial.write(heading.b[1]);
    Serial.write('\n');
  }
  #if 0 // print data
  // Serial.print(roll); Serial.print("\t");
  // Serial.print(phi); Serial.print("\t");
  // Serial.print(pitch); Serial.print("\t");
  // Serial.print(theta); Serial.print("\t");
  // Serial.print(yaw); Serial.print("\t");
  // Serial.print(psi); // Serial.print("\t");
  #endif
  #if 0 // Set to 1 to print the IMU data
  Serial.print(aX / 16384.0); Serial.print("\t"); // converted into g's
  Serial.print(aY / 16384.0); Serial.print("\t");
  Serial.print(aZ / 16384.0); Serial.print("\t");
  Serial.print(gXrate); Serial.print("\t"); // converted into °/s
  Serial.print(gYrate); Serial.print("\t");
  Serial.print(gZrate); Serial.print("\t");
  Serial.print(mX); Serial.print("\t"); // after gain and offset compensation
  Serial.print(mY); Serial.print("\t");
  Serial.print(mZ); Serial.print("\t");
  #endif
  #if 0 // Set to 1 to print the temperature
  float temperature = (float)temp / 340.0 + 36.53;
  Serial.print(temperature);
  #endif
  // Serial.println();
  delay(20);
}

```

```
#include <iostream>
#include <stdio.h>           // standard input / output functions
#include <stdlib.h>
#include <string.h>         // string function definitions
#include <unistd.h>         // UNIX standard function definitions
#include <fcntl.h>          // File control definitions
#include <errno.h>          // Error number definitions
#include <termios.h>        // POSIX terminal control definitions
#include <opencv/cv.h>
#include <opencv2/opencv.hpp>
#include <highgui.h>
#include <opencv2/contrib/contrib.hpp>
#include <fstream>
#include "fl/Headers.h" // fuzzy logic library

using namespace std;
using namespace cv;
using namespace fl;

#define EVER ;;
#define FUZZY

#if 0 // 1: recorded sample video, 0: camera
    #define DEBUG
#else
    #undef DEBUG
#endif

#if 1
    #define DRAW
#else
    #undef DRAW
#endif

#if 0
    #define RECORD
#else
    #undef RECORD
#endif

#define METHOD 3
```

```

#if METHOD == 0
    #define rg
    #define _ExG // Excess Green
#elif METHOD == 1
    #define rg
    #define _ExGR // Excess Green minus Excess Red
#elif METHOD == 2
    #define _MExG // Modified ExG
#elif METHOD == 3
    #define rg
    #define _CIVE // Color Index of Vegetation Extraction
#elif METHOD == 4
    #define rg
    #define _VEG // Vegetative
#elif METHOD == 5
    #define rg
    #define _ExG
    #define _ExGR
    #define _CIVE
    #define _VEG
    #define _COM // Combined (ExG, ExGR, CIVE, VEG)
#elif METHOD == 6
    #define rg
    #define _ExG
    #define _CIVE
    #define _VEG
    #define _MCOM // Combined (ExG, CIVE, VEG)
#elif METHOD == 7
    #define _NDVI // Normalized Difference Vegetation Index
#endif

void drawArrow(cv::Mat& img, cv::Point pStart, cv::Point pEnd, int len, int alpha, cv::Scalar& color, int thickness = 1, int lineType = 8);

bool cmpContourAreas(vector<Point> contour1, vector<Point> contour2) {
    return (fabs(contourArea(Mat(contour1))) > fabs(contourArea(Mat(contour2))));
}

```

```

int main(int argc, char** argv) {
    // serial configuration
    int hCom = open("/dev/ttySAC2", O_RDWR | O_NONBLOCK | O_NDELAY);

    if (hCom < 0) {
        cout << "Error " << errno << " opening " << "/dev/ttySAC2" << ": " << strerror (errno) << endl;
    }

    struct termios tty;
    memset (&tty, 0, sizeof tty);

    if (tcgetattr (hCom, &tty) != 0) {
        cout << "Error " << errno << " from tcgetattr: " << strerror(errno) << endl;
    }

    // set Baud Rate
    cfsetospeed (&tty, B9600);
    cfsetispeed (&tty, B9600);
    tty.c_cflag &= ~PARENB;           // make 8n1
    tty.c_cflag &= ~CSTOPB;
    tty.c_cflag &= ~CSIZE;
    tty.c_cflag |= CS8;
    tty.c_cflag &= ~CRTSCTS;         // no flow control
    tty.c_cc[VMIN] = 1;              // read doesn't block
    tty.c_cc[VTIME] = 5;             // 0.5 seconds read timeout
    tty.c_cflag |= CREAD | CLOCAL;  // turn on READ & ignore ctrl lines
    // make raw
    cfmakeraw(&tty);
    // flush port, then applies attributes
    tcflush( hCom, TCIFLUSH );
    if ( tcsetattr ( hCom, TCSANOW, &tty ) != 0) {
        cout << "Error " << errno << " from tcsetattr" << endl;
    }
    // serial

#ifdef DEBUG
    VideoCapture inputVideo;

    // Run -> Run Configurations -> opencv_app -> Arguments
    if (!inputVideo.open(argv[1])) {
        cout << "Error can't find the file." << endl;
        return -1;
    }
}

```

```

// acquire input size
const unsigned int height = static_cast<int>(inputVideo.get(CV_CAP_PROP_FRAME_HEIGHT));
const unsigned int width = static_cast<int>(inputVideo.get(CV_CAP_PROP_FRAME_WIDTH));
#else

VideoCapture inputVideo(0); // open the default camera

if (!inputVideo.isOpened())
    return -1;

const unsigned int width = inputVideo.get(CV_CAP_PROP_FRAME_WIDTH);
const unsigned int height = inputVideo.get(CV_CAP_PROP_FRAME_HEIGHT);
// print camera frame size
// cout << "Camera properties\n";
// cout << "width = " << width << endl << "height = " << height << endl;

#endif

const unsigned int scale_factor =30;
const int nRows = height * scale_factor / 100;
const int nCols = width * scale_factor / 100;

#ifdef RECORD
ofstream fID("data.txt");
VideoWriter outputVideo;

#ifdef DEBUG
const int codec = static_cast<int>(inputVideo.get(CV_CAP_PROP_FOURCC)); // Get codec type
outputVideo.open("output.avi", codec, inputVideo.get(CV_CAP_PROP_FPS), Size(2 * nCols, nRows), true);
#else
outputVideo.open("capture.avi",CV_FOURCC('D','I','V','X') , 30, Size(2 * nCols, nRows), true);
#endif

if (!outputVideo.isOpened()) {
    cout << "Could not open the output video for write." << endl;
    return -1;
}

#endif

int idx;
union i2b {
    char b[2];
    unsigned short int n;
} val;

```

```

const unsigned short int length = 2 * sizeof(unsigned short int);
uint8_t data[length+2];

double t, thresh;
Scalar level; // average intensity of image
Mat frame, img, R, G, B, r, g, b, Sum, h_concat, Iblur, vPL, vPR;
Mat ExG, ExGR, MExG, CIVE, COM, MCOM, NDVI, BW;
Mat VEG = Mat::zeros(nRows, nCols, CV_64FC1);
vector<Mat> channels;
vector<vector<Point> > contours;
vector<Vec4i> hierarchy;
vector<Point> pts;
// starting and end points
vector<Point> P; // 0:(top,left) 1:(bottom, left) 2:(top,right) 3:(bottom,right)

Vec4f line_param;
Point pt, pt1, pt2;
int _min, _max;
float c, diff, wl, wr;
double theta = 0, dist;
ostringstream num;

const unsigned int morph_size = 3;
Mat se = getStructuringElement(MORPH_ELLIPSE, Size(2 * morph_size + 1, 2 * morph_size + 1), Point(morph_size, morph_size));

int row, left, right, middle;
uchar *p;
bool flag_lhs, flag_rhs, outside = false;

Engine* ctrl = fl::FisImporter().fromFile("/home/odroid/MyProjects/opencv_prj/src/nav.fis");
std::string status;
if (!(ctrl->isReady(&status)))
    throw fl::Exception("Engine not ready. "
        "The following errors were encountered:\n" + status, FL_AT);

vector<float> distRng, angleRng;
distRng.push_back(static_cast<float>(ctrl->getInputVariable("d")->getMinimum()));
distRng.push_back(static_cast<float>(ctrl->getInputVariable("d")->getMaximum()));
angleRng.push_back(static_cast<float>(ctrl->getInputVariable("a")->getMinimum()));
angleRng.push_back(static_cast<float>(ctrl->getInputVariable("a")->getMaximum()));

```

```

for (EVER) { // loop
    t = (double)getTickCount();

    inputVideo >> frame;
    if (!frame.data)
        break; // ran out of film

#ifdef DEBUG
    unsigned int nf = static_cast<unsigned int>(inputVideo.get(CV_CAP_PROP_POS_FRAMES));
#endif

    resize(frame, frame, cvSize(nCols, nRows));

    frame.copyTo(img);
    split(img, channels);

    B = channels[0]; // Blue
    G = channels[1]; // Green
    R = channels[2]; // Red

    R.convertTo(R, CV_32F);
    G.convertTo(G, CV_32F);
    B.convertTo(B, CV_32F);

#ifdef rg
    // rg chromaticity space
    R.copyTo(r);
    G.copyTo(g);
    B.copyTo(b);
    Sum = r + g + b;
    r /= Sum;
    g /= Sum;
    b /= Sum;
    /*
    Mat bgr = Mat(img.rows, img.cols, CV_32FC3);
    Mat in[] = { b, g, r };
    int from_to[] = { 0, 0, 1, 1, 2, 2 };
    mixChannels(in, 3, &bgr, 1, from_to, 3);
    imshow("rg chromacity", bgr);
    */
#endif
}

```

```

#ifdef _ExG // Excess Green vegetation index
    ExG = 2 * g - r - b;
#ifdef _COM
#ifdef _MCOM
        normalize(ExG, ExG, 0, 255, NORM_MINMAX);
        ExG.convertTo(ExG, CV_8UC1);
        medianBlur(ExG, Iblur, 7); // smoothing image
        thresh = threshold(Iblur, BW, 0, 255, CV_THRESH_BINARY | CV_THRESH_OTSU);
#endif
#endif
#endif

#ifdef _ExGR // Excess Green minus Excess Red
// ExR = 1.4*r - g; // Excess Red
ExGR = 3*g - 2.4*r - b;

#ifdef _COM
    BW = ExGR > 0; // Positive threshold
#endif
#endif

#ifdef _MExG // Modified ExG
MExG = -0.884*R + 1.262*G - 0.311*B;
normalize(MExG, MExG, 0, 255.0, NORM_MINMAX);
MExG.convertTo(MExG, CV_8UC1);

    medianBlur(MExG, Iblur, 7); // smoothing image
// level = mean(MExG);
// BW_MExG = MExG > level.val[0];
    thresh = threshold(Iblur, BW, 0, 255, CV_THRESH_BINARY | CV_THRESH_OTSU);
#endif

#ifdef _CIVE // Color Index of Vegetation Extraction
CIVE = 0.441*r - 0.811*g + 0.385*b + 18.78745;
normalize(CIVE, CIVE, 0, 255, NORM_MINMAX);
CIVE.convertTo(CIVE, CV_8UC1);
CIVE = 255 - CIVE;

#ifdef _COM
#ifdef _MCOM
        medianBlur(CIVE, Iblur, 7); // smoothing image
        thresh = threshold(Iblur, BW, 0, 255, CV_THRESH_BINARY | CV_THRESH_OTSU);
#endif
#endif
#endif

```



```

#ifdef _VEG    // Vegetative index
    r.convertTo(r, CV_64FC1);
    g.convertTo(g, CV_64FC1);
    b.convertTo(b, CV_64FC1);

    Mat m1, m2;
    pow(r, 0.667, m1); // alpha = 0.667
    pow(b, 0.333, m2); // 1 - alpha
    multiply(m1, m2, m1);

    divide(g, m1, VEG);
    VEG.convertTo(VEG, CV_32FC1);

    for (int i = 0; i < img.rows; i++) {
        for (int j = 0; j < img.cols; j++) {
            if (isinf(VEG.at<float>(i, j))) VEG.at<float>(i, j) = 0;
        }
    }

#ifdef _COM
#ifdef _MCOM
    normalize(VEG, VEG, 0, 255, NORM_MINMAX);
    VEG.convertTo(VEG, CV_8UC1);

    medianBlur(VEG, Iblur, 7); // smoothing image
    level = mean(Iblur); // mean intensity
    BW = Iblur > level.val[0];

#endif
#endif
#endif

#ifdef _COM    // Combined (ExG, ExGR, CIVE, VEG)
    CIVE.convertTo(CIVE, CV_32FC1);

    COM = 0.25*ExG + 0.3*ExGR + 0.33*CIVE + 0.12*VEG;

    normalize(COM, COM, 0, 255, NORM_MINMAX);
    COM.convertTo(COM, CV_8UC1);

    medianBlur(COM, Iblur, 7); // smoothing image

    level = mean(Iblur); // mean intensity
    BW = Iblur > level.val[0];

#endif

```

```

#ifdef _MCOM
    CIVE.convertTo(CIVE, CV_32FC1);

    MCOM = 0.36*ExG + 0.47*CIVE + 0.17*VEG;

    normalize(MCOM, MCOM, 0, 255, NORM_MINMAX);
    MCOM.convertTo(MCOM, CV_8UC1);
    medianBlur(MCOM, Iblur, 7); // smoothing image

    thresh = threshold(Iblur, BW, 0, 255, CV_THRESH_BINARY | CV_THRESH_OTSU);
#endif

#ifdef _NDVI
    // Normalized Difference Vegetation Index
    NDVI = (G - R) / (G + R);

    normalize(NDVI, NDVI, 0, 255, NORM_MINMAX);
    NDVI.convertTo(NDVI, CV_8UC1);

    thresh = threshold(NDVI, BW, 0, 255, CV_THRESH_BINARY | CV_THRESH_OTSU);
#endif

    // improving segmentation
    morphologyEx(BW, BW, MORPH_CLOSE, se);
    morphologyEx(BW, BW, MORPH_OPEN, se);

    findContours(Mat(BW).clone(), contours, hierarchy, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE, Point(0, 0));
    // sort contours
    sort(contours.begin(), contours.end(), cmpContourAreas);

    unsigned int n = 0;
    for (;n < contours.size(); n++) {
        if (contourArea(contours[n]) < 0.1*contourArea(contours[0])) // 10%
            break;
    }
    // drawContours(img, contours, n, CV_RGB(255, 255, 0), 1, 8, hierarchy, 0, Point());

    if (n == 0)
        continue;

    vector<vector<Point> > ctrL, ctrR;
    vector<Moments> mu(n);
    vector<Point2f> mc(n);

```

```

for (unsigned int i = 0; i < n; i++) {
    mu[i] = moments(contours[i], true);
    mc[i] = Point2f(static_cast<float>(mu[i].m10 / mu[i].m00), static_cast<float>(mu[i].m01 / mu[i].m00));
    circle(img, mc[i], 3, CV_RGB(255, 127, 39), CV_FILLED, 8, 0);
}

if (n == 1)
    ctrL.push_back(contours[0]);
else if (n == 2) {
    if (mc[0].x > mc[1].x)
        swap(contours[0], contours[1]);
    ctrL.push_back(contours[0]);
    ctrR.push_back(contours[1]);
} else if (n > 2) {
    const int K = 2; // number of clusters to split the set by

    Mat labels, centers = Mat(mc)(Range(0, 2), Range::all());
    TermCriteria criteria ( TermCriteria::COUNT + TermCriteria::EPS, 10, 1 ); // algorithm termination criteria
    kmeans(Mat(mc).reshape(1,mc.size()), K, labels, criteria, 5, KMEANS_PP_CENTERS, centers);

    // one row per each cluster center
    for (int i = 0; i < K; i++)
        circle(img, Point(centers.row(i)), 3, CV_RGB(255, 0, 0), CV_FILLED, 8, 0);

    if (centers.at<float>(0, 0) > centers.at<float>(1, 0))
        labels = abs(labels-1);

    for (int j = 0; j < labels.rows; j++)
        (labels.at<float>(j, 0) == 0) ? ctrL.push_back(contours[j]):ctrR.push_back(contours[j]);
}

vPL = Mat(ctrL[0]);
for (unsigned int i = 0; i < ctrL.size(); i++) {
    if (i > 0) vconcat(vPL, Mat(ctrL[i]), vPL);
    drawContours(img, ctrL, i, CV_RGB(255, 0, 255), 1, 8, hierarchy, 0, Point());
}

if (n > 1) {
    vPR = Mat(ctrR[0]);
    for (unsigned int i = 0; i < ctrR.size(); i++) {
        if (i > 0) vconcat(vPR, Mat(ctrR[i]), vPR);
        drawContours(img, ctrR, i, CV_RGB(0, 255, 255), 1, 8, hierarchy, 0, Point());
    }
}

```

```

P.clear();
for (int i = 0; i < 2; i++) {
    fitLine((i==0)?vPL:vPR, line_param, CV_DIST_L2, 0, 0.01, 0.01);

    // top, (left / right)
    pt1.y = 0;
    pt1.x = static_cast<int>((-line_param[0] / line_param[1] * line_param[3] + line_param[2]));
    P.push_back(pt1);

    // bottom, (left / right)
    pt2.y = nRows - 1;
    pt2.x = static_cast<int>(line_param[0] / line_param[1] * (pt2.y - line_param[3]) + line_param[2]);
    P.push_back(pt2);

#ifdef DRAW
    if (!outside) {
        // auxiliary line
        line(img, pt1, pt2, (i == 0) ? CV_RGB(0, 255, 0) : CV_RGB(17, 71, 255), 1, CV_AA, 0); // plants' area
    }
#endif

    if (i == 1)    i = 2;
    _min = nCols, _max = -1;
    if (P.at(i).x < 0 || P.at(i).x > nCols || P.at(i+1).x < 0 || P.at(i+1).x > nCols) {
        for (int y = 0; y < nRows; y++){
            pt.x = static_cast<int>(line_param[0] / line_param[1] * (y - line_param[3]) + line_param[2]);
            if (pt.x > 0 && pt.x < nCols) {
                if (_min > pt.x) {
                    _min = pt.x;
                    P.at(i).x = pt.x;
                    P.at(i).y = y;
                }
            }
            if (pt.x > 0 && pt.x < nCols) {
                if (_max < pt.x) {
                    _max = pt.x;
                    P.at(i+1).x = pt.x;
                    P.at(i+1).y = y;
                }
            }
        }
    }
}
if (P.at(i).y > P.at(i+1).y)
    swap(P.at(i), P.at(i+1));

```

```

#ifdef DRAW
    if (!outside) {
        rectangle(img, Point(P.at(i).x-3, P.at(i).y-3), Point(P.at(i).x+3, P.at(i).y+3), CV_RGB(255,255,255), CV_FILLED, 8, 0);
        rectangle(img, Point(P.at(i+1).x-3,P.at(i+1).y-3), Point(P.at(i+1).x+3,P.at(i+1).y+3),CV_RGB(0,0,0),CV_FILLED, 8, 0);
    }
#endif

    if (n == 1) break;
}
contours.clear();
ctrL.clear(); ctrR.clear();
middle = nCols / 2;
// scan image
for (row = nRows - 1; row >= 0; row--) {
    // acquire a pointer to the start of each row
    p = BW.ptr<uchar>(row);
    left = middle;
    right = middle;
    flag_lhs = false;
    flag_rhs = false;
    for (EVER) {
        if (!flag_lhs) {
            if (p[left] == 255)
                flag_lhs = true;
            else
                if (left > 0)
                    left--;
        }
        if (!flag_rhs) {
            if (p[right] == 255)
                flag_rhs = true;
            else
                if (right < nCols)
                    right++;
        }
        if (flag_lhs && flag_rhs) {
            middle = (left + right) / 2;
            if (!outside) p[middle] = 255;
            pts.push_back(Point(row, middle));
            break;
        }
        else if (left == 0 || right == nCols - 1)
            break;
    }
} // scanning

```

```

if (pts.empty())
    continue;
// least squares method
fitLine(pts, line_param, CV_DIST_L2, 0, 0.01, 0.01);
pts.clear();
// output line parameters is a vector of 4 elements (vx, vy, x0 , y0)
// where (vx,vy) is a normalized vector collinear to the line
// and (x0,y0) is a point on the line
//  $y_0 = v_y/v_x * x_0 + c \rightarrow c = y_0 - x_0 * v_y/v_x$ 
c = line_param[3] - line_param[2] * line_param[1] / line_param[0];
pt1.y = 0; // Point(x,y) represent (column,row)
pt1.x = static_cast<int>(c);
pt2.y = nRows - 1;
pt2.x = static_cast<int>(line_param[1] / line_param[0] * pt2.y + c);
cvtColor(BW, BW, CV_GRAY2RGB);

#ifdef DRAW
if (!outside) {
    // guidance line, raw data
    line(img, pt1, pt2, CV_RGB(255, 255, 0), 1, CV_AA, 0);
    line(BW, pt1, pt2, CV_RGB(255, 255, 0), 1, CV_AA, 0); // bw region
}
#endif

_min=nCols, _max=-1;
if (pt1.x < 0 || pt1.x > nCols || pt2.x < 0 || pt2.x > nCols) {
    for (int y = 0; y < nRows; y++) {
        pt.x = static_cast<int>(line_param[1] / line_param[0] * y + c);
        if (pt.x > 0 && pt.x < nCols) {
            if (_min > pt.x) {
                _min = pt.x;
                pt1.x = pt.x;
                pt1.y = y;
            }
        }
        if (pt.x > 0 && pt.x < nCols) {
            if (_max < pt.x) {
                _max = pt.x;
                pt2.x = pt.x;
                pt2.y = y;
            }
        }
    }
}
}

```

```

    if (pt1.y > pt2.y)
        swap(pt1, pt2);

    theta = atan2(line_param[1], line_param[0]) * 180 / CV_PI; // slope of the line
    dist = (pt2.x - nCols / 2.0) * 100 / (float)nCols;
//    cout << dist << "\t" << theta << endl;

    if (P.size() == 2)
        outside = true;
    else if (!(pt1.x > P.at(0).x && pt1.x < P.at(2).x && pt2.x > P.at(1).x && pt2.x < P.at(3).x)) {
        outside = true;
    } else {
        outside = false;
#ifdef FUZZY
        if (dist >= distRng.at(0) && dist <= distRng.at(1) && theta >= angleRng.at(0) && theta <= angleRng.at(1)) {
            ctrl->getInputVariable("d")->setInputValue(dist);
            ctrl->getInputVariable("a")->setInputValue(theta);
            ctrl->process();
            diff = static_cast<float>(ctrl->getOutputVariable("u")->getOutputValue());
            wl = 0.7 + 0.3*diff;
            wr = 0.7 - 0.3*diff;
#ifdef DRAW
            Scalar arrowColor(255, 255, 255);
            drawArrow(img, Point(nCols/2-nCols/8, nRows-1), Point(nCols/2-nCols/8, (nRows-1- nRows*wl/3)), 10, 45, arrowColor );
            num.str("");
            num << round(wl * 100) / 100.0;
            putText(img, string(num.str()), cvPoint(nCols/2-nCols/8, nRows-1), FONT_ITALIC, 0.5, CV_RGB(255, 255, 255), 1, 8);

            drawArrow(img, Point(nCols/2+nCols/8, nRows-1), Point(nCols/2 + nCols/8, (nRows-1-nRows*wr/3)), 10, 45, arrowColor );
            num.str("");
            num << round(wr * 100) / 100.0;
            putText(img, string(num.str()), cvPoint(nCols/2+nCols/8, nRows-1), FONT_ITALIC, 0.5, CV_RGB(255, 255, 255), 1, 8);
#endif
        }
#endif
    }

    if (outside) {
        putText(img, "outside", cvPoint(10, 25), FONT_ITALIC, 0.8, CV_RGB(255, 0, 0), 1, 8);
        wl = 0; wr = 0;
    }
}

```

```

        if (!isnan(wr) && !isnan(wl)) {
//            cout << wl << "\t" << wr << endl;
            val.n = round(wl*10000.0);
            data[0] = 0x20; // start of data
            for(idx=1;idx <=2; idx++) {
                data[idx] = val.b[idx-1];
            }
            val.n = round(wr*10000.0);
            for(idx=3;idx <=4; idx++) {
                data[idx] = val.b[idx-3];
            }
            data[5] = '\r'; // end of data
            for (idx = 0; idx<=5; idx++)
                write(hCom,data+idx,1);
        }
#ifdef RECORD
        if (fID.is_open())
            fID << dist << "\t" << theta << endl;
        outputVideo << h_concat;
#endif
#ifdef DRAW
        if (!outside) {
            num.str("");
            num << round(theta * 10) / 10.0;
            putText(img, string(num.str()).append("deg"), cvPoint(10, 20), FONT_ITALIC, 0.6, CV_RGB(255, 255, 0), 1, 8);

            num.str("");
            num << round(dist * 10) / 10.0;
            putText(img, string(num.str()).append("%"), cvPoint(10, 40), FONT_ITALIC, 0.6, CV_RGB(255, 255, 0), 1, 8);
        }
}

```



```

#ifdef DEBUG
    num.str("");
    num << nf;
    putText(img, string(num.str()), cvPoint(10, nRows-20), FONT_ITALIC, 0.45, CV_RGB(0, 0, 0), 1, 8);
#endif

    if (!outside) {
        // pt1: top, center
        // pt2: bottom, center
        rectangle(img, Point(pt1.x - 3, pt1.y - 3), Point(pt1.x + 3, pt1.y + 3), CV_RGB(255, 255, 255), CV_FILLED, 8, 0);
        rectangle(img, Point(pt2.x - 3, pt2.y - 3), Point(pt2.x + 3, pt2.y + 3), CV_RGB(0, 0, 0), CV_FILLED, 8, 0);
    }
    line(img, Point(nCols / 2, 0), Point(nCols / 2, nRows), CV_RGB(80, 80, 80), 1, 8, 0); // vertical line
//    hconcat(frame, img, h_concat);
    hconcat(img, BW, h_concat);
    imshow("output", h_concat);
//    imshow("output", img);

    if (waitKey(1) >= 0) // +1ms delay
        break;
#endif

    t = ((double)getTickCount() - t) / getTickFrequency();
//    cout << "Time passed in seconds:" << t << endl;
} // loop

#ifdef RECORD
    fID.close();
    outputVideo.release();
#endif

    close(hCom);
    return 0;
}

void drawArrow(Mat & img, Point pStart, Point pEnd, int len, int alpha, cv::Scalar& color, int thickness, int lineType) {
    Point arrow;
    double angle = atan2((double)(pStart.y - pEnd.y), (double)(pStart.x - pEnd.x));
    line(img, pStart, pEnd, color, thickness, lineType);
    arrow.x = pEnd.x + len * cos(angle + CV_PI * alpha / 180);
    arrow.y = pEnd.y + len * sin(angle + CV_PI * alpha / 180);
    line(img, pEnd, arrow, color, thickness, lineType);
    arrow.x = pEnd.x + len * cos(angle - CV_PI * alpha / 180);
    arrow.y = pEnd.y + len * sin(angle - CV_PI * alpha / 180);
    line(img, pEnd, arrow, color, thickness, lineType);
}

```