



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΤΟΜΕΑΣ Μ.Κ. & Α.Ε.  
ΕΡΓΑΣΤΗΡΙΟ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

Διπλωματική Εργασία

**Σχεδιασμός και Υλοποίηση Ηλεκτρικού/ Ηλεκτρονικού  
Υποσυστήματος, και Προγραμματισμός Πραγματικού  
Χρόνου Ρομπότ Διαστημικού Εξομοιωτή**

Κωνσταντίνος Ε. Μαχαιράς  
Επιβλέπων Καθηγητής : Ε. Γ. Παπαδόπουλος

Αθήνα 2013



Αφιερώνεται στην οικογένειά μου





# Περίληψη

Η παρούσα διπλωματική εργασία έχει θέμα το σχεδιασμό και την υλοποίηση του ηλεκτρικού/ηλεκτρονικού υποσυστήματος, καθώς και την ανάπτυξη του λογισμικού ενός ρομπότ επίπεδου διαστημικού εξομοιωτή. Ο εν λόγω εξομοιωτής βρίσκεται στο Εργαστήριο Αυτομάτου Ελέγχου του Τμήματος Μηχανολόγων Μηχανικών ΕΜΠ, και αποτελείται από ρομπότ, τα οποία αιωρούνται σε απόσταση 10 [μm] πάνω σε μια τράπεζα γρανίτη αμελητέας τραχύτητας. Σκοπός του εξομοιωτή είναι η πειραματική μελέτη της δυναμικής των ρομπότ σε συνθήκες που εξομοιώνουν την έλλειψη βαρύτητας στο επίπεδο, μέσω εξάλειψης της τριβής. Η πρωτοτυπία έγκειται στο ότι τα ρομπότ είναι μικρής μάζας, χαμηλού κόστους, πλήρως αυτόνομα και με υποσυστήματα ανάλογα με αυτά ενός πραγματικού διαστημικού ρομποτικού συστήματος. Η εργασία αυτή αφορά το νέο (δεύτερο) ρομπότ, αφού το πρώτο είναι λειτουργικό και έτοιμο για πειράματα. Κίνητρο για την ανάπτυξη ενός συστήματος σαν αυτό αποτελεί η συνεχώς αυξανόμενη σπουδαιότητα των ρομποτικών συστημάτων στο διάστημα σε περιπτώσεις όπως: εξερεύνηση του διαστήματος, κατασκευή, συντήρηση και επιθεώρηση συστημάτων, προσέγγιση και πρόσδεση σε άλλα σώματα που βρίσκονται σε τροχιά.

Τα ρομπότ του εξομοιωτή αποτελούνται από 3 υποσυστήματα: α) το μηχανολογικό: είναι κατασκευασμένα κυρίως από αλουμίνιο, διαθέτουν δύο βραχίονες δύο βαθμών ελευθερίας κι έναν σφόνδυλο αντίδρασης σαν εναλλακτικό τρόπο κίνησης, β) το πνευματικό: χρησιμοποιούν αέριο CO<sub>2</sub> για να αιωρούνται μέσω 3 αεροδράνων αλλά και για να κινούνται στο επίπεδο μέσω 3 ζευγών προωθητήριων (thrusters), γ) το ηλεκτρικό/ ηλεκτρονικό: προσδιορίζουν τη θέση τους μέσω δύο συστημάτων αισθητήρων (οπτικοί αισθητήρες στο σώμα τους και εξωτερική κάμερα) και μέσω αυτομάτου ελέγχου επενεργούν κατάλληλα στο περιβάλλον.

Χρησιμοποιώντας την τεχνική του σχεδιασμού βασισμένου στο μοντέλο (model-based design), ο σχεδιασμός και η ανάπτυξη του νέου ρομπότ έγινε σε συνολικό επίπεδο, αφού η ανάπτυξη αλγορίθμων και λογισμικού και η τελική εγκατάσταση στο υλικό (hardware) εξετάστηκαν σαν αλληλένδετα κομμάτια. Το πακέτο λογισμικού που χρησιμοποιήθηκε ήταν το xPC Target, της πρωτοπόρου στον τομέα εταιρίας, Mathworks. Στο περιβάλλον αυτό, τα απλά μοντέλα προσομοίωσης Simulink μετατρέπονται άμεσα σε προγράμματα C, τα οποία εκτελούνται σε πραγματικό χρόνο κατευθείαν στο τελικό υλικό (hardware) του ρομπότ. Πάνω σε αυτή τη βάση, έγινε δυνατή η αξιολόγηση και η εξέλιξη του σχεδιασμού σε όλη τη διάρκεια της ανάπτυξης. Έτσι, δόθηκε έμφαση στη καινοτομία, ενώ παρακάμφθηκαν πιθανά προβλήματα χαμηλού επιπέδου σε λογισμικό (software) και υλικό (hardware).

Σαν κεντρική υπολογιστική μονάδα επιλέχτηκε μια κάρτα PC-104, και συμπληρώθηκε με κάρτες ίδιου φορμάτ για παροχή ισχύος, και ενσωμάτωση ψηφιακών εισόδων/εξόδων. Για την τροφοδοσία του συστήματος χρησιμοποιήθηκαν μπαταρίες LiPo. Αναπτύχθηκε σύστημα εντοπισμού μέσω τριών οπτικών αισθητήρων προσαρτημένων στο σώμα του ρομπότ, αλλά και ένα δεύτερο σύστημα βασισμένο σε εξωτερική κάμερα που παρακολουθεί τον χώρο εργασίας. Δύο ασύρματες Ethernet γέφυρες χρησιμοποιήθηκαν για α) τηλεχειρισμό του ρομπότ και β) ασύρματη λήψη δεδομένων κίνησης από το σύστημα της εξωτερικής κάμερας σε πραγματικό χρόνο. Σχεδιάστηκαν και εκτυπώθηκαν οι απαραίτητες ηλεκτρονικές πλακέτες: μία για διανομή ισχύος στα διάφορα υποσυστήματα, μία για οδήγηση των προωθητήριων (thrusters), και μία για συλλογή δεδομένων από τους οπτικούς αισθητήρες. Τέλος, έγιναν αρκετά πειράματα κίνησης με το ρομπότ σε τελική μορφή, με στόχο την αξιολόγηση των αλγορίθμων επεξεργασίας που αναπτύχθηκαν, πάνω σε δεδομένα των δύο συστημάτων εντοπισμού.

# Abstract

The present diploma thesis deals with the design and implementation of the electric/ electronic subsystem, as well as the software development of a robot of a planar space simulator. This simulator has been placed in the Control Systems Lab of the Mechanical Engineering Department of NTUA, and it consists of robots hovering in a distance of 10 [ $\mu\text{m}$ ] over a granite table of negligible roughness. The purpose of the simulator is the experimental study of the robots' dynamics in conditions that simulate zero gravity in 2D, through the elimination of friction. The novelty lies in the fact that the robots are of low mass, low cost, completely autonomous and carry subsystems that resemble these of a real space robot. This work refers to the new (second) robot, as the first is functional and ready for experiments. The motivation for the development of such a system is the increasing importance of space robots in cases such as: space exploration, construction, maintenance and inspection of systems in space, approach and attachment to other bodies in orbit.

The robots of the simulator consist of 3 subsystems: a) mechanical: they are made of aluminum, and they bear two arms, with two joints each, and a reaction wheel as an alternative way of motion, b) pneumatic: they use  $\text{CO}_2$  gas in order to hover using 3 air-bearings, and for planar motion using 3 pairs of thrusters, c) electric/ electronic: they estimate their posture using two systems of sensors (optical sensors on their body and an external camera) and through automatic control act appropriately upon the environment.

Employing the technique of model-based design, the development of the new robot was carried out as a whole, since the design of algorithms, the software development and the final integration on the hardware were studied as interdependent parts. The software used was xPC Target, from Mathworks. In this environment, C code is quickly generated from simple Simulink models, and it is finally executed on the robot's hardware in hard real time. In this manner, validation and verification of the design were continuously performed throughout the development. As a result, emphasis was put on innovation, while at the same time potential low-level problems in hardware and software were surmounted.

In order to attain computation autonomy a PC-104 system is used, including a CPU board, a 48 Digital I/O card and a power supply board. Power autonomy is achieved using Li-Po batteries. The localization of the robot is achieved through two different systems of sensors: (a) relative estimation sensors, that is to say three optical sensors, for fast estimation and (b) an absolute estimation sensor, namely an off-board camera, providing more accurate estimation. Two wireless Ethernet bridges are used for a) the remote control of the robot, and b) to receive motion data from the camera system in real time. Four PCBs have been designed and printed for power routing, thrusters' control, collecting optical sensors' data and another one to be used as a control panel. Finally, several experiments were carried out with the robot in full format, in order to evaluate the processing algorithms that were developed, on data derived from the camera and the optical sensors systems.

# Ευχαριστίες

Θέλω να ευχαριστήσω πολύ τον Καθηγητή κ. Ε. Παπαδόπουλο για την εμπιστοσύνη του, τις πολύτιμες συμβουλές του αλλά και για τον ενθουσιασμό που μου μετέδωσε κατά την εκπόνηση αυτής της εργασίας. Με την βοήθεια και την στήριξη του απέκτησα σημαντική εμπειρία στο τομέα της μηχανολογίας και της ρομποτικής ειδικότερα. Θέλω να ευχαριστήσω επίσης τα μέλη του Εργαστηρίου Αυτομάτου Ελέγχου των Μηχανολόγων Μηχανικών του ΕΜΠ για την βοήθειά τους και για τις όμορφες και δημιουργικές στιγμές που περάσαμε μαζί. Ευχαριστώ ιδιαίτερα των Ιωσήφ Παρασκευά, χωρίς την βοήθεια του οποίου δεν θα είχαν γίνει όλα αυτά, καθώς επίσης τους Στέλιο Ανδρέου, Κώστα Βλάχο, Γιώργο Ρεκλείτη, Θάλεια Φλέσσα, και Αλέξανδρο Νικολακάκη. Τέλος, χρωστώ πολλά στην οικογένειά μου και στους φίλους μου για την αγάπη τους και την στήριξη τους όλα αυτά τα χρόνια.

# Περιεχόμενα

<b>ΠΕΡΙΛΗΨΗ</b> .....	1
<b>ABSTRACT</b> .....	2
<b>ΕΥΧΑΡΙΣΤΙΕΣ</b> .....	3
<b>ΠΕΡΙΕΧΟΜΕΝΑ</b> .....	4
<b>ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ</b> .....	7
<b>ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ</b> .....	11
<b>ΠΡΟΛΟΓΟΣ</b> .....	12
<b>ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ</b> .....	15
1.1 Ρομπότ στο διάστημα;.....	15
1.2 Διαστημικά σκουπίδια .....	16
1.3 Σύγχρονα διαστημικά ρομπότ.....	19
1.4 Τρόποι εξομίωσης διαστημικού περιβάλλοντος στη γη .....	21
1.5 Δομή Εργασίας .....	22
<b>ΚΕΦΑΛΑΙΟ 2 - ΠΕΡΙΓΡΑΦΗ ΕΠΙΠΕΔΟΥ ΔΙΑΣΤΗΜΙΚΟΥ ΕΞΟΜΟΙΩΤΗ ΤΟΥ ΕΡΓΑΣΤΗΡΙΟΥ - ΣΧΕΔΙΑΣΤΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ ΝΕΟΥ ΡΟΜΠΟΤ</b> .....	25
2.1 Εισαγωγή .....	25
2.2 Ο επίπεδος διαστημικός εξομοιωτής του εργαστηρίου .....	26
2.2.1 Τράπεζα γρανίτη .....	26
2.2.2 Ρομπότ .....	27
2.2.3 Σύστημα παρακολούθησης μέσω κάμερας .....	29
2.3 Σχεδιαστικές απαιτήσεις νέου ρομπότ σε ηλεκτρικό, ηλεκτρονικό και προγραμματιστικό επίπεδο .....	29
<b>ΚΕΦΑΛΑΙΟ 3 - ΑΡΧΙΚΟΣ ΗΛΕΚΤΡΙΚΟΣ/ ΗΛΕΚΤΡΟΝΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ</b> .....	31
3.1 Εισαγωγή .....	31
3.2 Υλικό (Hardware) .....	32
3.2.1 Υλοποίηση σε επίπεδο μικροελεγκτών.....	32
3.2.2 Χρήση Single Board Computer (SBC).....	33
3.3 Λογισμικό - Συστήματα Πραγματικού Χρόνου .....	34
3.3.1 Τρόποι υλοποίησης συστημάτων πραγματικού χρόνου .....	35
3.3.2 Χαρακτηριστικά λειτουργικού συστήματος πραγματικού χρόνου (RTOS).....	35
3.3.3 Πυρήνας RTOS.....	36
3.3.4 Διεργασίες (tasks) .....	36
3.3.5 Hard / Soft Real time.....	36
3.4 Σχεδιασμός βασισμένος στο μοντέλο (Model-based design) .....	37
3.5 xPC Target.....	38
3.5.1 Προγραμματισμός με λειτουργικά διαγράμματα (Block-Diagram-Based) .....	38
3.5.2 Functional Rapid Prototyping.....	39
3.5.3 Hardware in the Loop (HIL) .....	41
3.5.4 xPC Target .....	42
3.6 Αρχικός σχεδιασμός ηλεκτρικού υποσυστήματος .....	43
3.8 Επιλογή ηλεκτρικών/ ηλεκτρονικών στοιχείων.....	44
3.8.1 Κεντρική μονάδα επεξεργασίας CPU PC-104.....	44
3.8.2 Κάρτα εισόδων/ εξόδων (PC-104).....	44
3.8.3 Ασύρματη γέφυρα Ethernet (Wireless Ethernet bridge) .....	45

3.8.4 Μπαταρίες .....	45
3.8.5 Κάρτες παροχής ισχύος (Power Supply) .....	47
3.9 Το υλικό (hardware) σε λειτουργία.....	48
<b>ΚΕΦΑΛΑΙΟ 4 - ΕΥΡΕΣΗ ΘΕΣΗΣ ΚΑΙ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ ΑΠΟ ΣΥΣΤΗΜΑ ΚΑΜΕΡΑΣ.....</b>	<b>49</b>
4.1 Εισαγωγή .....	49
4.2 Χαρακτηριστικά συστήματος κάμερας.....	50
4.3 Γενική ιδέα προσδιορισμού θέσης και προσανατολισμού .....	51
4.4 Αλγόριθμος εύρεσης θέσης μεμονωμένων φωτεινών πηγών (LEDs) .....	51
4.5 Αλγόριθμοι εύρεσης θέσης/ προσανατολισμού των κινούμενων σωμάτων.....	53
4.5.1 Αρχική ιδέα και πρώτη υλοποίηση .....	53
4.5.2 Αλγόριθμος δύο επιπέδων φωτεινότητας και πειράματα .....	54
4.6 Ορισμός του Παγκόσμιου Συστήματος Συντεταγμένων .....	57
4.7 Ανάπτυξη τελικού προγράμματος για δύο ρομπότ.....	58
4.8 Αποστολή Δεδομένων στο υπολογιστικό Σύστημα του ρομπότ .....	59
4.8.1 Πρωτόκολλο TCP/IP .....	59
4.8.2 Πρωτόκολλο UDP/IP .....	60
4.8.3 Συμπεράσματα.....	61
<b>ΚΕΦΑΛΑΙΟ 5 - ΕΥΡΕΣΗ ΘΕΣΗΣ ΚΑΙ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ ΑΠΟ ΣΥΣΤΗΜΑ ΟΠΤΙΚΩΝ ΑΙΣΘΗΤΗΡΩΝ ....</b>	<b>63</b>
5.1 Εισαγωγή .....	63
5.2 Τύποι ποντικών Η/Υ.....	64
5.2.1 Οπτικά Ποντίκια .....	64
5.2.2 Ποντίκια laser .....	65
5.2.3 Καινούριες Τεχνολογίες .....	66
5.3 Οπτική Ροή (Optical flow).....	68
5.4 Βασικά τεχνικά χαρακτηριστικά οπτικών αισθητήρων .....	68
5.5 Αριθμός αισθητήρων στο ρομπότ .....	70
5.6 Σύνδεση αισθητήρων στο κεντρικό υπολογιστικό σύστημα.....	70
5.7 Περιγραφή συστήματος μικροελεγκτή - PS/2 ποντικών .....	71
5.8 Επιλογή μικροελεγκτή (microcontroller).....	73
5.9 Η πλατφόρμα Arduino .....	73
5.10 Σύνδεση μικροελεγκτή Arduino – PS/2 ποντικιού.....	75
5.11 Επιλογή οπτικού αισθητήρα.....	75
5.12 Αποστολή δεδομένων στο PC-104 .....	76
5.13 Σχεδιασμός και κατασκευή πρωτοτύπου Arduino Shield .....	77
5.14 Διάταξη αισθητήρων στο σώμα του ρομπότ.....	78
5.15 Αλγόριθμοι εντοπισμού .....	80
5.15.1 Βασικός κορμός αλγορίθμων .....	81
5.15.2 Αλγόριθμος Μέσου Όρου .....	84
5.15.3 Αλγόριθμος Ακριβέστερης Απόστασης .....	85
5.15.4 Αλγόριθμος ενδιάμεσης γωνίας .....	86
5.15.5 Αλγόριθμοι ανακατασκευής τρίτου αισθητήρα .....	87
5.16 Συνδυασμός με δεδομένα από το σύστημα της κάμερας .....	87
<b>ΚΕΦΑΛΑΙΟ 6 - ΣΧΕΔΙΑΣΜΟΣ ΛΟΓΙΣΜΙΚΟΥ (SOFTWARE DESIGN) .....</b>	<b>89</b>
6.1 Εισαγωγή .....	89
6.2 Εγκατάσταση πακέτου xPC Target και αρχικές ρυθμίσεις .....	90
6.2.1 Υπολογιστές Host και Target.....	90
6.2.2 Αρχικές Ρυθμίσεις xPC Target .....	90
6.2.3 Κτίσιμο Boot CD/USB .....	91

6.2.4 Τηλεχειρισμός – Επικοινωνία με το target PC .....	91
6.3 Λογισμικό Συστήματος Οπτικών Αισθητήρων.....	92
6.3.1 Προγραμματισμός μικροελεγκτή Arduino .....	92
6.3.2 Μοντέλο Simulink .....	93
6.4 Λογισμικό Συστήματος Κάμερας .....	94
6.4.1 Ασύρματη επικοινωνία με ρομπότ .....	94
6.4.2 Πρόγραμμα επεξεργασίας εικόνας.....	95
6.4.3 Μοντέλο Simulink .....	96
6.5 Εικονική Πραγματικότητα με VRML .....	97
6.5.1 Σχεδιασμός με V-Realm Builder .....	97
6.5.2 Έλεγχος εικονικού κόσμου μέσω Simulink .....	98
<b>ΚΕΦΑΛΑΙΟ 7 - ΗΛΕΚΤΡΟΝΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΕΚΤΥΠΩΣΗ ΠΛΑΚΕΤΩΝ .....</b>	<b>101</b>
7.1 Εισαγωγή .....	101
7.2 Πακέτα ηλεκτρονικής σχεδίασης NI Multisim και NI Ultiboard .....	102
7.3 Βασικές αρχές Ηλεκτρονικού Σχεδιασμού (Electronics Design).....	103
7.4 Συνολικό Ηλεκτρικό/ Ηλεκτρονικό Υποσύστημα του Ρομπότ.....	112
7.5 Πλακέτα Διανομής Ισχύος (Power Board).....	113
7.6 Πίνακας ελέγχου (Control panel) .....	115
7.7 Πλακέτα οδήγησης βαλβίδων των προωθητηρίων (Thrusters Board) .....	116
7.8 Πλακέτα οπτικών αισθητήρων (Optical Sensors Arduino Shield) .....	117
<b>ΚΕΦΑΛΑΙΟ 8 - ΡΟΜΠΟΤ ΣΕ ΤΕΛΙΚΗ ΜΟΡΦΗ - ΒΑΘΜΟΝΟΜΗΣΗ ΚΑΙ ΠΕΙΡΑΜΑΤΑ .....</b>	<b>121</b>
8.1 Το ρομπότ σε τελική μορφή .....	121
8.2 Βαθμονόμηση Συστήματος Οπτικών Αισθητήρων .....	122
8.2.1 Συντελεστής μετατροπής [counts] σε [mm] .....	123
8.2.2 Δεδομένα Πειραμάτων Βαθμονόμησης .....	125
8.3 Πειράματα προσδιορισμού θέσης/ προσανατολισμού από οπτικούς αισθητήρες .....	126
8.3.1 Πειράματα ευθύγραμμης τροχιάς .....	126
8.3.2 Πειράματα κυκλικής τροχιάς .....	127
8.3.3 Αλγόριθμοι με μεταβλητά βάρη σε πειράματα κυκλικής τροχιάς .....	130
8.3.4 Αλγόριθμοι με ανακατασκευή σε πειράματα κυκλικής τροχιάς .....	131
8.4 Πειράματα προσδιορισμού θέσης/ προσανατολισμού από κάμερα .....	134
8.4.1 Πειράματα ευθύγραμμης τροχιάς .....	134
8.4.2 Πειράματα κυκλικής τροχιάς .....	135
<b>ΚΕΦΑΛΑΙΟ 9 - ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ .....</b>	<b>139</b>
9.1 Συμπεράσματα .....	139
9.2 Μελλοντική Εργασία .....	140
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΑΛΛΕΣ ΠΗΓΕΣ.....</b>	<b>143</b>
<b>ΠΑΡΑΡΤΗΜΑ Α.....</b>	<b>146</b>
<b>ΠΑΡΑΡΤΗΜΑ Β.....</b>	<b>157</b>
<b>ΠΑΡΑΡΤΗΜΑ Γ .....</b>	<b>166</b>

# Κατάλογος σχημάτων

Σχήμα 1.1 Διαστημικά σκουπίδια σε τροχιά γύρω από τη γη.....	17
Σχήμα 1.2 Αριστερά: Χαρακτηριστικά μεγέθη διαστημικών σκουπιδιών. Δεξιά: Αποτέλεσμα σύγκρουσης σώματος μεγέθους 1 [cm] με ταχύτητα 6.8 [km/s].....	17
Σχήμα 1.3 Ο Theodore Solomons δίπλα σε μεταλλική σφαίρα που μόλις έπεσα από τον ουρανό σε μια φάρμα στο Worcester της Νότιας Αφρικής, τον Απρίλιο του 2000. ....	18
Σχήμα 1.4 Το πλήθος των διαστημικών σκουπιδιών από το 1960 μέχρι το 2210 (εκτιμώμενο). ....	19
Σχήμα 1.5 Αριστερά: Το πρόγραμμα TAMU Space Sweeper with Sling-Sat. Δεξιά: Το πρόγραμμα CleanSpace One. ....	19
Σχήμα 1.6 Αριστερά: Ο βραχίονας Canadarm 1 (SRMS) . Δεξιά: Ο αστροναύτης Stephen K. Robinson προσδεμένος στον βραχίονα Canadarm2. ....	20
Σχήμα 1.7 Παραβολική πτήση αεροσκάφους. ....	21
Σχήμα 1.8 Από τα αριστερά προς τα δεξιά α) Επίπεδος εξομοιωτής της Thales Alenia Space β) Επίπεδος εξομοιωτής AMPHIS γ) Το πρόγραμμα SPHERES του MIT. ....	22
Σχήμα 2.1 Τράπεζα γρανίτη του εργαστηρίου. ....	26
Σχήμα 2.2 Αριστερά: Πρώτο ρομπότ σε λειτουργία. Δεξιά: ζεύγος προωθητηρίων (thrusters).....	27
Σχήμα 2.3 Αριστερά: Σφόνδυλος αντίδρασης. Δεξιά: Πύργος καρτών PC-104.....	28
Σχήμα 2.4 Αριστερά: Το σασί του νέου ρομπότ όπως κατασκευάστηκε από τον Σ. Ανδρέου. Δεξιά: 3D σχέδιο του νέου ρομπότ. ....	30
Σχήμα 3.1 Διάφορα μοντέλα μικροελεγκτών. ....	32
Σχήμα 3.2 Παραδείγματα Single Board Computers (SBCs). ....	33
Σχήμα 3.3 Σχηματική αναπαράσταση προγραμματισμού σε πραγματικό χρόνο. ....	34
Σχήμα 3.4 Non-real time, soft real time και hard real time εφαρμογές. ....	34
Σχήμα 3.5 Συγκρότηση λογισμικού ενός ενσωματωμένου συστήματος. ....	36
Σχήμα 3.6 Σχηματική αναπαράσταση σύγκρισης συστημάτων 'hard' και 'soft' real time. ....	37
Σχήμα 3.7 Μέθοδος σχεδιασμού βασισμένου στο μοντέλο. ....	38
Σχήμα 3.8 Αρχική μοντελοποίηση του συστήματος σε επίπεδο προσομοίωσης. ....	39
Σχήμα 3.9 Παραγωγή κώδικα πραγματικού χρόνου από το μοντέλο φυσικού συστήματος (HIL).....	39
Σχήμα 3.10 Τυπική προσέγγιση και μέθοδος rapid prototyping.....	40
Σχήμα 3.11 Παραγωγή κώδικα πραγματικού χρόνου από το μοντέλο φυσικού συστήματος (HIL).....	41
Σχήμα 3.12 Διαδικασία ανάπτυξης μέσω xPC Target.....	42
Σχήμα 3.13 Επικοινωνία του Target PC με φυσικά συστήματα μέσω καρτών εισόδων/ εξόδων. ....	43
Σχήμα 3.14 Αρχικός ηλεκτρικός/ ηλεκτρονικός σχεδιασμός. ....	44
Σχήμα 3.15 Αριστερά: Η κεντρική μονάδα επεξεργασίας PCM-4153 της Advantech. Δεξιά: Κάρτα ψηφιακών εισόδων εξόδων PC104-DIO48 της Measurement Computing. ....	45
Σχήμα 3.16 Η ασύρματη γέφυρα Ethernet WNCE2001 της Netgear. ....	45
Σχήμα 3.17 Μπαταρία LiPo. ....	46
Σχήμα 3.18 Αριστερά: DC-DC converter υψηλής τάσης. Δεξιά: DC-DC converter χαμηλής τάσης.....	47
Σχήμα 3.19 Αρχική υλοποίηση του ηλεκτρικού σχεδίου. ....	48
Σχήμα 3.20 Αριστερά: πύργος PC-104 πάνω στο ρομπότ. Δεξιά: Ασύρματες γέφυρες Ethernet πάνω στο ρομπότ. ....	48
Σχήμα 4.1 Αριστερά: Η κάμερα mvBlueFOX-124G. Δεξιά: Φαίνεται ο τρόπος στήριξης της πάνω από τον χώρο εργασίας. ....	50
Σχήμα 4.2 Παραμορφωμένο στιγμιότυπο λόγω του ευρυγώνιου φακού της κάμερας. ....	53

Σχήμα 4.3 Αριστερά: Αστερισμός της Κασσιόπης. Δεξιά: Τοποθέτηση LEDs σε σχηματισμό όμοιου με τον αστερισμό.....	53
Σχήμα 4.4 Τα βήματα του αλγορίθμου εντοπισμού τριών σωμάτων σε εικόνες, όπως παράχθηκαν από τον κώδικα προσομοίωσης σε Matlab.....	55
Σχήμα 4.5 Η διάταξη των LEDs κατά τα πρώτα πειράματα.....	55
Σχήμα 4.6 Η οροφή του δεύτερου ρομπότ με τα LEDs όπως τοποθετήθηκαν τελικά.....	56
Σχήμα 4.7 Αριστερά: Όλα τα LEDs συνδεδεμένα. Δεξιά: Απλοποιημένη διάταξη- αποσύνδεση κάποιων LEDs.....	56
Σχήμα 4.8 Το Αρχικό παγκόσμιο Σύστημα συντεταγμένων και η γωνία του ρομπότ, όπως βρέθηκαν από πειράματα.....	57
Σχήμα 4.9 Αριστερά: Διανύσματα θέσης ως προς τα δύο συστήματα. Δεξιά: Διάνυσμα θέσης και γωνία στροφής ως προς το τελικό παγκόσμιο σύστημα.....	58
Σχήμα 4.10 Αριστερά: LEDs 1 <sup>ου</sup> σώματος. Δεξιά: LEDs 2 <sup>ου</sup> σώματος.....	58
Σχήμα 5.1 Τρόπος λειτουργίας των οπτικών ποντικιών.....	65
Σχήμα 5.2 Παρουσίαση των τεχνολογιών optical και laser.....	66
Σχήμα 5.3 Παρουσίαση της τεχνολογίας Blue Track.....	67
Σχήμα 5.4 Υπολογισμός κίνησης από δύο διαδοχικές εικόνες A,B με τη μέθοδο της οπτικής ροής.....	68
Σχήμα 5.5 Παράδειγμα υπολογισμού κίνησης με τη μέθοδο της οπτικής ροής.....	68
Σχήμα 5.6 Σωματόδετο σύστημα συντεταγμένων του αισθητήρα.....	70
Σχήμα 5.7 Αριστερά: Εύκολη σύνδεση οπτικού αισθητήρα με μικροελεγκτή. Δεξιά: Τα υποσυστήματα μιας πλακέτας οπτικού αισθητήρα.....	72
Σχήμα 5.8 Πακέτο δεδομένων Κίνησης.....	72
Σχήμα 5.9 Αριστερά: η πλακέτα Arduino. Δεξιά: Τεχνικά χαρακτηριστικά της πλακέτας.....	74
Σχήμα 5.10 Περιγραφή των pins της θύρας PS/2.....	75
Σχήμα 5.11 Microsoft Ready Mouse.....	76
Σχήμα 5.12 Αριστερά: Πείραμα λήψης δεδομένων από 3 οπτικούς αισθητήρες Δεξιά: δοκιμή κυκλώματος σειριακής επικοινωνίας RS-232.....	77
Σχήμα 5.13 Διαφορετικές όψεις του Arduino Shield.....	77
Σχήμα 5.14 Σύνδεση Shield με οπτικούς αισθητήρες, Arduino και σειριακή μετάδοση αποτελεσμάτων.....	78
Σχήμα 5.15 Αριστερά: πρώτη πειραματική διάταξη με οπτικούς αισθητήρες. Δεξιά: γράφημα των θέσεων των 3 οπτικών αισθητήρων σε πείραμα 4 sec.....	78
Σχήμα 5.16 Διανύσματα θέσης των 3 αισθητήρων ως προς το σωματόδετο σύστημα συντεταγμένων.....	79
Σχήμα 5.17 Κάτω και πάνω μέρος του ρομπότ. Τα ΣΣ των συστημάτων κάμερας και οπτικών αισθητήρων ταυτίζονται.....	79
Σχήμα 5.18 Αριστερά: Μηχανολογικό σχέδιο των ρυθμιζόμενων βάσεων των οπτικών αισθητήρων Δεξιά: φωτογραφία της πραγματικής διάταξης πάνω στο ρομπότ.....	80
Σχήμα 5.19 Τρία διαδοχικά στιγμιότυπα τυχαίας κίνησης και διανυσματική ανάλυση αυτής για έναν αισθητήρα.....	82
Σχήμα 5.20 Μετατοπίσεις (p1,p2,p3) των αισθητήρων ως προς το εξωτερικό ΣΣ O <sub>R</sub> και η εκτιμώμενη μετατόπιση και γωνία (pc, Θ) του σωματόδετου ΣΣ O <sub>L</sub> ως προς το ΣΣ O <sub>R</sub> .....	83
Σχήμα 5.21 Αριστερά: Οι γωνίες όπως επιστρέφονται από την συνάρτηση atan2(x,y). Δεξιά: Το πρόβλημα ασυνέχειας στο σημείο pi/-pi.....	84
Σχήμα 5.22 Λήψη δεδομένων από το σύστημα της κάμερας (κόκκινο στιγμιότυπο) και από το σύστημα των αισθητήρων (γκρι στιγμιότυπο).....	88
Σχήμα 6.1 Οι υπολογιστές host και target.....	90
Σχήμα 6.2 Γραφικό περιβάλλον xPC Target Explorer.....	91
Σχήμα 6.3 Normal και external mode.....	92



Σχήμα 6.4 Πρόγραμμα Arduino για συλλογή και αποστολή δεδομένων κίνησης. ....	93
Σχήμα 6.5 Μοντέλο Simulink για παραλαβή και επεξεργασία των δεδομένων των οπτικών αισθητήρων. ....	94
Σχήμα 6.6 Περιβάλλον του υπολογιστή της κάμερας με όλες τις συνιστώσες σε λειτουργία. ....	95
Σχήμα 6.7 Αριστερά: μοντέλο simulink. Δεξιά: ρυθμίσεις του μπλοκ Network Configuration. ....	96
Σχήμα 6.8 Αριστερά: ρυθμίσεις του μπλοκ UDP Receive. Δεξιά: ρυθμίσεις του μπλοκ Unpack. ....	96
Σχήμα 6.9 Μοντέλο του ρομπότ όπως φαίνεται στο περιβάλλον του V-Realm Builder. ....	97
Σχήμα 6.10 Μοντέλο Simulink για σύνδεση με εικονικό κόσμο. ....	98
Σχήμα 6.11 Εικονικός κόσμος κατά τη διάρκεια πειράματος. ....	99
Σχήμα 7.1 Τα περιβάλλοντα των σχεδιαστικών πακέτων Multisim και Ultiboard. ....	102
Σχήμα 7.2 Παραδείγματα καλής και κακής συνδεσμολογίας. ....	107
Σχήμα 7.3 Συνολικό ηλεκτρολογικό/ηλεκτρονικό υποσύστημα. ....	112
Σχήμα 7.4 Οι δύο όψεις της πλακέτας ισχύος σε 3D όπως παράχθηκαν από το NI Ultiboard. ....	114
Σχήμα 7.5 Οι δύο όψεις της πλακέτας ισχύος, αφού έχουν γίνει οι κολλήσεις των ηλεκτρονικών. ...	115
Σχήμα 7.6 Δύο όψεις του πίνακα ελέγχου του ρομπότ. ....	115
Σχήμα 7.7 Άλλες όψεις του πίνακα ελέγχου του ρομπότ. ....	116
Σχήμα 7.8 Αριστερά: Η πλακέτα οδήγησης των thrusters σε 3D όψη όπως παράχθηκε από το NI Ultiboard. Δεξιά: Η έτοιμη πλακέτα μετά τις κολλήσεις των ηλεκτρονικών. ....	117
Σχήμα 7.8 3D όψη του Arduino Shield όπως παράχθηκε από το NI Ultiboard. ....	118
Σχήμα 7.9 Το Optical Sensors Shield έτοιμο και συνδεδεμένο στην πλατφόρμα Arduino. ....	118
Σχήμα 7.10 Δοκιμή των κυκλωμάτων στο εργαστήριο πριν την αποστολή για εκτύπωση. ....	119
Σχήμα 7.11 Πλακέτες όπως παραλήφθηκαν μετά την εκτύπωση. ....	119
Σχήμα 7.12 Δοκιμή των πλακετών στο εργαστήριο πριν την τοποθέτηση στο ρομπότ. ....	120
Σχήμα 7.13 Οι πλακέτες τοποθετημένες στο ρομπότ. ....	120
Σχήμα 8.1 Ρομπότ στην τελική του μορφή, με όλα τα συστήματά του ενσωματωμένα κατά τη διάρκεια πειράματος με ροδάκια πάνω σε ξύλινη βάση. ....	122
Σχήμα 8.2 Ρύθμιση του ύψους του αισθητήρα στα 2 [mm] με χρήση κατάλληλου ελάσματος. ....	123
Σχήμα 8.3 Αριστερά: Πείραμα ευθύγραμμης κίνησης μεγάλης ακρίβειας. Δεξιά: Πείραμα ευθύγραμμης κίνησης με ακίνητο σώμα και κινούμενο επίπεδο. ....	124
Σχήμα 8.4 Πείραμα ευθύγραμμης κίνησης χρησιμοποιώντας μπάρες-οδηγούς. ....	125
Σχήμα 8.5 Τρία πειράματα ευθείας 197 [mm], μικρής ταχύτητας : 197 [mm] σε 3 [sec]. ....	126
Σχήμα 8.6 Τρία πειράματα ευθείας 197 [mm], μεγάλης ταχύτητας : 197 [mm] σε 1 [sec]. ....	127
Σχήμα 8.7 Πείραμα κυκλικής τροχιάς ακτίνας 160 [mm] σε ξύλινη βάση. ....	128
Σχήμα 8.8 Αριστερά: περίπου μισός κύκλος, με κόκκινο το κέντρο (αλγόριθμος μέσου όρου) και διακεκομμένη γραμμή η πραγματική τροχιά. Δεξιά: το ρομπότ στην αρχική του θέση μετά από έναν κύκλο (αλγόριθμος μέσου όρου). ....	128
Σχήμα 8.9 Αριστερά: τροχιά του κέντρου (αλγόριθμος μέσου όρου) από 3 πειράματα χαμηλής ταχύτητας, με διακεκομμένη η πραγματική τροχιά. Δεξιά: τροχιά του κέντρου (αλγόριθμος μέσου όρου) από 3 πειράματα υψηλής ταχύτητας, με διακεκομμένη η πραγματική τροχιά. ....	129
Σχήμα 8.10 Αριστερά: τροχιά του κέντρου (αλγόριθμος μέσου όρου) σε πείραμα κυκλικής τροχιάς δύο περιστροφών. Δεξιά: τροχιά του κέντρου (αλγόριθμος μέσου όρου) σε πείραμα κυκλικής τροχιάς τριών περιστροφών. ....	130
Σχήμα 8.11 Αριστερά: τροχιά του κέντρου (αλγόριθμος ακριβέστερης απόστασης) σε πείραμα κυκλικής κίνησης χαμηλής ταχύτητας με μεταβλητό βάρος w. Δεξιά: Μεγέθυνση του κρίσιμου σημείου όπου το σφάλμα του απλού αλγορίθμου αρχίζει να αυξάνεται. ....	131
Σχήμα 8.12 Αριστερά: τροχιά του κέντρου (αλγόριθμος ενδιάμεσης γωνίας) σε πείραμα κυκλικής κίνησης χαμηλής ταχύτητας με μεταβλητό βάρος w. Δεξιά: Μεγέθυνση του σημείου όπου το σφάλμα του απλού αλγορίθμου συνεχώς αυξάνεται. ....	131

Σχήμα 8.13 Τροχιά του κέντρου (αλγόριθμος ακριβέστερης απόστασης) σε πείραμα κυκλικής κίνησης υψηλής ταχύτητας με μεταβλητό βάρος $w$ χωρίς ανακατασκευή.....	132
Σχήμα 8.14 Τροχιά του κέντρου (αλγόριθμος ενδιάμεσης γωνίας) σε πείραμα κυκλικής κίνησης υψηλής ταχύτητας με μεταβλητό βάρος $w$ χωρίς ανακατασκευή.....	132
Σχήμα 8.16 Πειράματα ευθύγραμμης κίνησης 200 [mm] σε διάφορα σημεία κάτω από την κάμερα. .....	134
Σχήμα 8.17 Αποτελέσματα ευθύγραμμων κινήσεων σε διάφορες θέσεις πάνω στον χώρο εργασίας μετά από επεξεργασία του συστήματος της κάμερας .....	135
Σχήμα 8.18 Αποτελέσματα κυκλικών κινήσεων σε διάφορες θέσεις πάνω στον χώρο εργασίας μετά από επεξεργασία του συστήματος της κάμερας .....	136
Σχήμα 8.19 Μεταβολή της γωνίας του ρομπότ σε πειράματα κυκλικών κινήσεων σε διάφορες θέσεις πάνω στον χώρο εργασίας, μετά από επεξεργασία του συστήματος της κάμερας. Στον άξονα του χρόνου: 250 μετρήσεις αντιστοιχούν σε 1 δευτερόλεπτο .....	137

# Κατάλογος πινάκων

Πίνακας 4.1 Σύγκριση UDP/IP και TCP/IP.....	61
Πίνακας 7.1 Αντιστοιχία Ρεύματος [A] - Πάχος ίχνους [mil ή mm].....	104
Πίνακας 8.1 Αποτελέσματα πειραμάτων ευθύγραμμης κίνησης για διαδρομές 100 [mm], 200 [mm]. .....	125

# Πρόλογος

Έχει μεγάλο ενδιαφέρον αν αναλογιστούμε τη σύγχρονη διαστημική πραγματικότητα, και τους στόχους των διαστημικών αποστολών τα τελευταία χρόνια, σε επαγγελματικό αλλά και σε ερασιτεχνικό επίπεδο. Βλέπουμε ανθρώπους να ζουν μόνιμα στο διάστημα, κατοικίδια να μπαίνουν σε τροχιά, ακόμη και εταιρίες να γυρίζουν εκεί τα διαφημιστικά τους. Πειράματα μαθητών, εργαστήρια επιστημόνων, δίκτυα επικοινωνίας, τηλεσκοπία, κάμερες, όπλα, ακόμη και διαστημικά σκουπίδια γυρίζουν ασταμάτητα γύρω από τη γη. Υπερσύγχρονα ρομπότ εξερευνούν τους γειτονικούς πλανήτες και ήδη κλείνονται τα πρώτα εισιτήρια με προορισμό τον πλανήτη Άρη. Συγκινητική είναι επίσης η εναγώνια αναζήτηση του ανθρώπου για κοσμική παρέα. Μουσική και εικόνες από τη γη, μαζί με χαιρετισμούς προς εξωγήινους σε όλες τις γλώσσες, ταξιδεύουν εδώ και 30 χρόνια προς το άγνωστο διάστημα, μόνο και μόνο για να πουν σε κάποιους, ότι κάποιοι άλλοι ζούσαν πολύ παλιά, κάπου πολύ μακριά τους. Γη και διάστημα πλέον είναι ένα. Χάρη στην έμφυτη περιέργεια, και την ευφυΐα του ανθρώπου, το διάστημα έγινε επιτέλους το σπίτι του.

Παράλληλα όμως, με την τεχνολογική εξέλιξη που αφορά τα διαστημικά ταξίδια, μια άλλη τεχνολογική επανάσταση έλαβε χώρα τα τελευταία χρόνια στη γη. Ο λόγος για την άνθιση της ρομποτικής. Ευφυή ρομποτικά συστήματα αναλαμβάνουν όλο και περισσότερες εργασίες, που θεωρούνται επικίνδυνες, χρονοβόρες και πολλές φορές αδύνατες για τον άνθρωπο. Ρομποτικές σκούπες, ρομποτικά ψάρια ή ρομποτικά αυτοκίνητα ανήκουν πλέον στην καθημερινότητα μας. Μας διευκολύνουν, μας φροντίζουν, μας κάνουν παρέα αλλά και μας ψυχαγωγούν. Είναι πλέον αναπόσπαστο κομμάτι του πολιτισμού μας.

Δεν είναι πολλά χρόνια, που γίνεται λόγος για διαστημικά ρομπότ. Ρομποτικοί βραχίονες και ρόβερ εξερεύνησης ήδη υπάρχουν και λειτουργούν στο διάστημα, ενώ ο αριθμός τους είναι μικρός σχετικά με αυτά που σχεδιάζονται αυτή την στιγμή και θα ενεργοποιηθούν στο μέλλον. Ο λόγος του φαινομένου αυτού είναι απλός. Όπως περιγράφηκε πιο πάνω, στο διάστημα υπάρχουν άνθρωποι που σκέφτονται, επικοινωνούν και δημιουργούν ` υπάρχει πολιτισμός. Είναι άλλος ένας πολιτισμός σε τροχιά. Βιώνοντας τα τεράστια οφέλη της ρομποτικής στον γήινο πολιτισμό, μόνο παράλογο θα ήταν να ακυρώσουμε τη χρησιμότητα της στον νέο αυτόν διαστημικό πολιτισμό, έναν πολιτισμό μάλιστα χωρίς βαρύτητα.

Η πρώτη από τις τέσσερις φυσικές αλληλεπιδράσεις που βιώνει ένας άνθρωπος είναι η βαρύτητα. Η δυσκολία ενός μωρού να σταθεί στα πόδια του είναι η πρώτη γνωριμία του με αυτή. Αν και ασθενέστερη των υπολοίπων (ηλεκτρομαγνητισμός, ασθενής και ισχυρή πυρηνική αλληλεπίδραση), επηρεάζει ασύγκριτα την καθημερινότητα. Όλες οι ανθρώπινες κατασκευές, μαζί και τα σύγχρονα ρομποτικά συστήματα, σχεδιάστηκαν για να την αντιμετωπίσουν και να την εκμεταλλεύονται. Έτσι λοιπόν, όλη η τεχνογνωσία και εμπειρία των τελευταίων αιώνων θεμελιώθηκε πάνω στην δύναμη της βαρύτητας όπως αυτή γίνεται αντιληπτή πάνω στη γη (όλα τα σώματα πέφτουν με επιτάχυνση  $g=9.81\text{m/s}^2$ ).

Είναι λιγότερο από μισό αιώνα, που ο άνθρωπος άρχισε να σχεδιάζει και να κατασκευάζει μηχανές για λειτουργία σε συνθήκες έλλειψης βαρύτητας. Ο λόγος αυτού, η γνωριμία του με το διάστημα. Μια κοινή αντίληψη είναι ότι 'στο διάστημα δεν υπάρχει βαρύτητα, γι αυτό οι αστροναύτες πετούν μέσα στο δορυφόρο'. Η έκφραση αυτή μπορεί εύκολα να προκαλέσει παρερμηνείες. Βαρύτητα υπάρχει, και αυτός είναι μάλιστα ο λόγος που οι αστροναύτες, όπως και το φεγγάρι, μένουν σε τροχιά. Στην πραγματικότητα οι αστροναύτες 'πέφτουν' ελεύθερα μαζί με τον τεχνητό δορυφόρο που τους φιλοξενεί. Τα σώματα σε τροχιά δεν είναι παρά σώματα σε διαρκή ελεύθερη πτώση. Λόγω βέβαια της εφαιπτομενικής τους ταχύτητας ποτέ δεν θα φτάσουν στο έδαφος. Ένας άνθρωπος όμως σε ελεύθερη πτώση, έχει την αίσθηση της αώρησης, αισθάνεται αβαρής. Όταν πέφτει μαζί με το διαστημόπλοιο του προς τη γη, αισθάνεται ότι πετάει μέσα σε αυτό, βρίσκεται δηλαδή σε συνθήκες έλλειψης βαρύτητας.

Δεν είναι λίγα τα ατυχήματα που έχουν προκληθεί από ανθρώπινα χέρια σε εργασίες σε τροχιά, λόγω ακριβώς της πολύ διαφορετικής δυναμικής. Ο κίνδυνος μάλιστα που ελλοχεύει σε τέτοιες καταστάσεις, λόγω κυρίως των μεγάλων ταχυτήτων των σωμάτων σε τροχιά ( $10 \text{ [km/s]}$ ), καθιστά

επιτακτική τη χρήση ρομπότ για τέτοιου είδους εργασίες. Τις ίδιες όμως δυσκολίες θα αντιμετωπίσει κι ένα ρομπότ στις ίδιες συνθήκες. Όσο δύσκολο είναι για έναν άνθρωπο να συνηθίσει τη ζωή σε μηδενική βαρύτητα, τόσο, και πιο δύσκολο είναι να προγραμματίσει ένα ρομπότ να λειτουργήσει αυτόνομα σε τέτοιες συνθήκες. Η δυναμική των σωμάτων αλλάζει έντονα. Αρκεί να σκεφτεί κανείς πως ακόμη και η εμφάνιση του προσώπου των αστροναυτών αλλάζει κατά την παραμονή τους στο διάστημα.

Κατά την ανάπτυξη τέτοιων συστημάτων, η θεωρητική ανάλυση των μηχανισμών είναι εφικτή, η πειραματική όμως μελέτη πολύ δύσκολη. Όντας στη γη, οι τρόποι εξομοίωσης συνθηκών έλλειψης βαρύτητας δεν είναι πολλοί, η εφαρμογή τους δε, δύσκολη. Ένας τρόπος είναι οι παραβολικές πτήσεις με αεροσκάφη, ενώ ένας άλλος θάλαμοι σε ελεύθερη πτώση. Ο τρόπος εξομοίωσης όμως που χρησιμοποιείται κατά κόρον είναι οι επίπεδοι εξομοιωτές. Με αυτούς επιτυγχάνεται η εξομοίωση των συνθηκών έλλειψης βαρύτητας σε 2 διαστάσεις, μέσω της κίνησης χωρίς τριβή σωμάτων στο επίπεδο. Τέτοιοι εξομοιωτές χρησιμοποιούνται σε διάφορα εργαστήρια και αποτελούν έναν εξαιρετικό και ευέλικτο τρόπο πειραματικής επαλήθευσης θεωριών που αφορούν την ιδιότυπη αυτή δυναμική. Στο Εργαστήριο Αυτομάτου Ελέγχου των Μηχανολόγων Μηχανικών στο ΕΜΠ, έχει κατασκευαστεί και συνεχώς εξελίσσεται ένας τέτοιου τύπου επίπεδος διαστημικός εξομοιωτής που περιλαμβάνει ρομπότ με βραχίονες. Η ανάπτυξη του ηλεκτρικού/ ηλεκτρονικού υποσυστήματος και ο προγραμματισμός ενός τέτοιου ρομπότ είναι το θέμα της παρούσας διπλωματικής εργασίας.



# ΚΕΦΑΛΑΙΟ 1

## Εισαγωγή

### 1.1 Ρομπότ στο διάστημα;

Αναλογιζόμενοι τη σύγχρονη διαστημική πραγματικότητα, το πρώτο πράγμα που αντικρίζουμε είναι χιλιάδες τεχνητοί δορυφόροι σε τροχιά γύρω από τη γη. Οι στόχοι αυτών ποικίλουν · είναι πολύπλοκα συστήματα που αναλαμβάνουν από χρέη τηλεπικοινωνιακών σταθμών μέχρι χρέη μεγάλων αστρονομικών τηλεσκοπίων. Όπως όμως σε όλα τα πολύπλοκα τεχνολογικά συστήματα, οι βλάβες και τα τεχνικά προβλήματα δε λείπουν · αντίθετα, ανέρχονται σε χιλιάδες. Το ερώτημα που τίθεται εδώ, είναι αν η επισκευή, η συντήρηση και η επίβλεψη τέτοιων συστημάτων ενδείκνυται και μπορεί να γίνεται από ανθρώπους, όπως συνέβαινε μέχρι τώρα.

Γενικά, τα ρομπότ σχεδιάζονται και κατασκευάζονται για να εκτελούν εργασίες που ο άνθρωπος δεν μπορεί, ή δεν χρειάζεται να κάνει. Ένα χαρακτηριστικό λοιπόν παράδειγμα δύσκολων συνθηκών εντοπίζεται στον χώρο του διαστήματος. Έτσι λοιπόν, μία από τις σημαντικότερες εφαρμογές της σύγχρονης ρομποτικής είναι αυτή στον τομέα της εξερεύνησης και μελέτης του διαστήματος. Οι

μεγάλες ταχύτητες των σωμάτων σε τροχιά (της τάξης των 10 [km/s]), σε συνδυασμό με τις συνθήκες μικροβαρύτητας, συνθέτουν ένα πολύ επικίνδυνο για τον άνθρωπο περιβάλλον. Αυτό εξηγεί και τα αρκετά ατυχήματα που έχουν λάβει μέχρι σήμερα χώρα σε διαστημικές αποστολές και οφείλονται σε λανθασμένους ανθρώπινους χειρισμούς. Αυτό που διαφαίνεται στην εποχή μας είναι πως τα ρομπότ είναι αυτά που μπορούν να αντιμετωπίσουν τέτοια περίπλοκα και επικίνδυνα προβλήματα στο διάστημα, έτσι ώστε να περιοριστεί το ρίσκο και ο κίνδυνος του χειρισμού τους από ανθρώπους. Όπως και στη γη, οι εργασίες που μπορούν να αναλάβουν ρομποτικά συστήματα, είναι ρουτίνας (έλεγχος διαστημικού σταθμού) αλλά και αυξημένου κινδύνου (ταξίδι στα όρια του ηλιακού συστήματος).

Ένα άλλο αξιοπρόσεχτο σημείο στην σύγχρονη διαστημική πραγματικότητα είναι το μεγάλο κόστος των διαστημικών αποστολών. Πιο συχνές είναι οι μεγάλες, βαρείς και επομένως ακριβές αποστολές, και ο κύριος λόγος που συμβαίνει αυτό έγκειται στο γεγονός ότι πολύπλοκοι και ογκώδεις μηχανισμοί στέλνονται μονοκόμματοι σε τροχιά. Είναι γνωστό στην επιστημονική κοινότητα ότι η συναρμολόγηση πολύπλοκων συστημάτων στο διάστημα αντί για τη γη, θα μειώσει δραστικά το κόστος αλλά και το ρίσκο των αποστολών. Προϋπόθεση όμως γι' αυτό αποτελούν τα ευφυή ρομποτικά συστήματα που μόλις στις μέρες μας δοκιμάζονται για πρώτη φορά στις ιδιότυπες συνθήκες του διαστήματος.

Δεν γίνεται να μην αναφερθεί επίσης το πρόβλημα που δημιουργείται από τις αυξημένες ανάγκες των δορυφορικών συστημάτων σε καύσιμο. Η ζωή ενός τεχνητού δορυφόρου σήμερα εξαρτάται κυρίως από την ποσότητα καυσίμου που διαθέτει. Όταν αυτό τελειώσει, ο δορυφόρος παραμένει σε τροχιά σαν άχρηστο σώμα. Ένα σώμα που τρέχει με ιλιγγιώδη ταχύτητα και απειλεί κάθε τι λειτουργικό που βρίσκεται σε τροχιά. Αυτό δεν θα συνέβαινε, αν υπήρχε η κατάλληλη τεχνολογία για ανεφοδιασμό των δορυφορικών συστημάτων με καύσιμο. Οι πρώτες προσπάθειες για ανάπτυξη τέτοιας τεχνολογίας γίνονται μόλις στην εποχή μας, αλλά είναι σίγουρη η ευρεία εκμετάλλευση της στο άμεσο μέλλον. Επιτακτική κρίνεται βέβαια και σε αυτή την περίπτωση η χρήση ρομποτικών συστημάτων ακριβείας.

Ένα τελευταίο αλλά πολύ σημαντικό θέμα που χρήζει ειδικής αντιμετώπισης, είναι αυτό των διαστημικών σκουπιδιών. Ο λόγος για οποιοδήποτε άχρηστο σώμα βρίσκεται σε τροχιά. Ο κίνδυνος λόγω του αυξανόμενου αριθμού των σωμάτων αυτών είναι πιο μεγάλος από ποτέ στις μέρες μας. Όπως θα περιγραφεί παρακάτω, οι αστροναύτες, οι λειτουργικοί δορυφόροι αλλά ακόμη και οι άνθρωποι στη γη, απειλούνται καθημερινά από αυτά τα ιλιγγιωδώς κινούμενα σώματα. Το πεδίο των πιθανών λύσεων στο σύγχρονο αυτό πρόβλημα είναι ακόμα πολύ ευρύ, σίγουρο είναι όμως ότι ειδικά ρομποτικά συστήματα θα έχουν κυρίαρχο ρόλο στην αντιμετώπισή του.

Από τα παραπάνω συνάγεται ότι είναι αναγκαία πλέον η ανάπτυξη και χρήση ρομπότ ικανών να χειρίζονται άλλα παθητικά ή ενεργητικά σώματα στο διάστημα, διαδικασία γνωστή και ως 'On Orbit Servicing' (OOS). Εκτός βέβαια των προαναφερθέντων επιχειρημάτων, όπως αναφέρθηκε και στον πρόλογο, πρόκειται για άλλον έναν πολιτισμό σε τροχιά. Δεδομένου, ότι έχει αποδειχθεί η χρησιμότητα της ρομποτικής στον γήινο πολιτισμό, δεν υπάρχει κανένας λόγος αυτή να μην εφαρμοστεί και στον πολιτισμό που σιγά σιγά χτίζεται στο διάστημα. Το ερευνητικό ενδιαφέρον είναι μάλιστα τεράστιο, αν σκεφτεί κανείς ότι πολλές φορές οι τεχνολογίες που αναπτύσσονται για διαστημικούς σκοπούς, βρίσκουν ευρεία εφαρμογή στον γήινο πολιτισμό. Το σκουπάκι, οι σόλες των αθλητικών παπουτσιών, τα συνθετικά υλικά ακόμη και οι πάνες μωρού έχουν τις ρίζες τους στην διαστημική έρευνα.

## 1.2 Διαστημικά σκουπίδια

Παλαιότερα, κοινή ήταν η άποψη πως το διάστημα είναι απέραντο και απεριόριστο, το οποίο βέβαια σχετικά και γενικά ισχύει. Οι επιτόλαιες όμως πρακτικές των διαστημικών οργανισμών τα τελευταία χρόνια γέμισαν γρήγορα τον διαστημικό χώρο κοντά στη γη με χιλιάδες τεχνητούς δορυφόρους, χωρίς κανένα σχέδιο διαφυγής τους από εκεί μετά την ολοκλήρωση της αποστολής τους. Για περισσότερο από μισό αιώνα, άχρηστα σώματα, γνωστά και ως διαστημικά σκουπίδια, συσσωρεύονται στις τροχιές γύρω από τη γη (βλ. Σχήμα 1.1). Το πρώτο διαστημικό σκουπίδι μπορεί να θεωρηθεί το Sputnik (1957), το οποίο ολοκλήρωσε την αποστολή του τη στιγμή ακριβώς που μπήκε σε τροχιά. Τα διαστημικά σκουπίδια περιλαμβάνουν ανενεργούς δορυφόρους, δεξαμενές

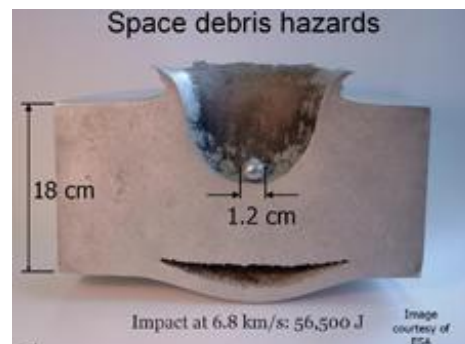


καυσίμου, ακόμη κι ένα χαμένο γάντι από το 1965. Σκουπίδια μικρότερου μεγέθους προκαλούνται επίσης λόγω του ότι η ισχυρή υπεριώδης ακτινοβολία προκαλεί αλλοίωση των υλικών, κι έτσι η μπογιά και άλλα υλικά επιφανειών αποβάλλονται σε μικρά κομματάκια. Τέλος, οι συγκρούσεις σωμάτων σε τροχιά αλλά και οι εκρήξεις (π.χ. δεξαμενών καυσίμου), αποτελούν ατυχήματα τα οποία εντείνουν το φαινόμενο και φανερώνουν τους μεγάλους κινδύνους που κρύβονται πίσω του.



**Σχήμα 1.1** Διαστημικά σκουπίδια σε τροχιά γύρω από τη γη.

Σήμερα, τα μικρότερα διαστημικά σκουπίδια ξεπερνούν σε αριθμό τους μετεωρίτες σε τροχιά. Οι διαστημικοί οργανισμοί παρακολουθούν συνεχώς τα αντικείμενα αυτά. Συγκεκριμένα παρακολουθούν περισσότερα από 22,000 αντικείμενα μεγαλύτερα των 10 [cm], δηλαδή αρκετά μεγάλα για να εντοπιστούν από ραντάρ. Υπολογίζονται ακόμη άλλα 500,000 μικρότερα αντικείμενα και περισσότερα από 10,000,000 μεγέθους μικρότερου του 1 [cm] (βλ. Σχήμα 1.2). Η συνολική μάζα των διαστημικών σκουπιδιών εκτιμάται περίπου στους 6,300 τόνους, ενώ η μέση ταχύτητα αυτών προκειμένου να μένουν σε τροχιά, είναι της τάξης των 10 [km/s]. Σε τέτοιες ταχύτητες, ακόμη και αντικείμενα μεγέθους 1 [mm] αποτελούν απειλή για ενεργούς δορυφόρους αλλά και αστροναύτες (βλ. Σχήμα 1.2). Ατυχήματα, όπως η σύγκρουση των δορυφόρων Iridium 33 και Kosmos 2251 (Φεβρουάριος 2009), όχι μόνο καταστρέφουν υπερσύγχρονους και ακριβούς δορυφόρους αλλά προκαλούν έντονα μη γραμμική αύξηση του πλήθους των σκουπιδιών λόγω των πολλαπλών προκληθέντων θραυσμάτων.



**Σχήμα 1.2** Αριστερά: Χαρακτηριστικά μεγέθη διαστημικών σκουπιδιών. Δεξιά: Αποτέλεσμα σύγκρουσης σώματος μεγέθους 1 [cm] με ταχύτητα 6.8 [km/s].

Τα σύγχρονα διαστημικά σκάφη κατασκευάζονται, έτσι ώστε να αντέχουν την σύγκρουση με πολύ μικρά αντικείμενα, τα μεγαλύτερα όμως παρακολουθούνται και απλά αποφεύγονται αν υπάρχει θέμα σύγκρουσης. Για παράδειγμα, τα τελευταία χρόνια ο διεθνής διαστημικός σταθμός αναγκάστηκε να κάνει περισσότερες από μία μανούβρες αποφυγής κάθε χρόνο.

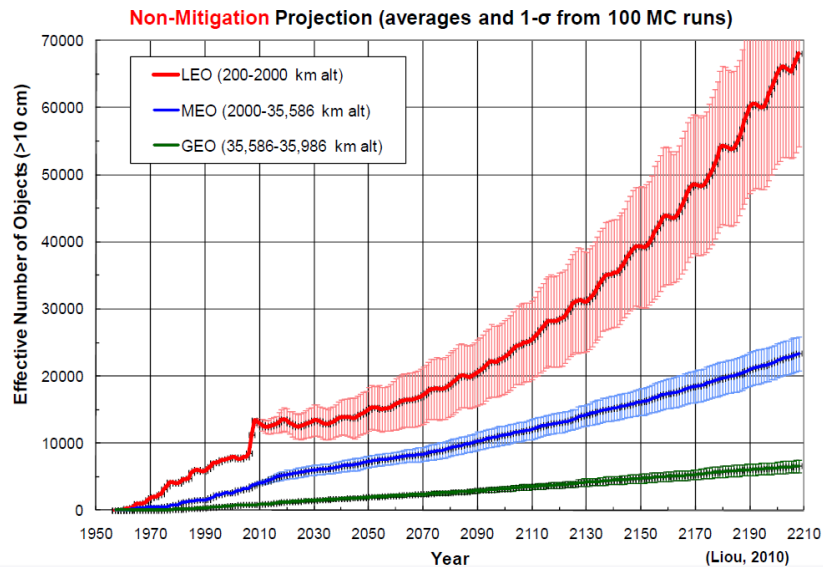
Ο κίνδυνος όμως δεν εξαντλείται στον χώρο του διαστήματος. Κίνδυνος υπάρχει και για τους ανθρώπους στη γη. Όπως οτιδήποτε άλλο σε τροχιά, έτσι και τα αντικείμενα αυτά χάνουν σιγά σιγά ενέργεια λόγω τριβών και κινούνται πολύ αργά προς τη γη. Υπολογίζεται ότι ένα από αυτά μπαίνει στην ατμόσφαιρα κάθε μέρα. Τα περισσότερα από αυτά καίγονται στην πορεία και δεν φτάνουν στο έδαφος. Δεν είναι όμως λίγα, τα παραδείγματα σωμάτων που επιζούν από τη τριβή με την ατμόσφαιρα και πέφτουν επικίνδυνα και ανεξέλεγκτα στη γη (βλ. Σχήμα 1.3).



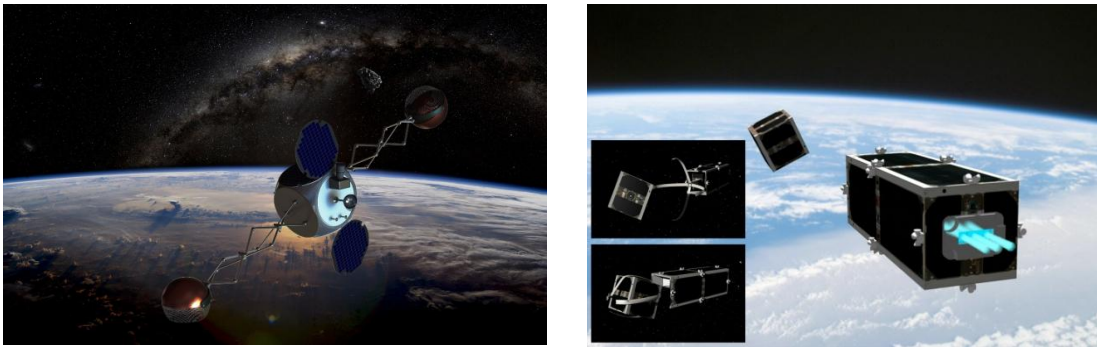
**Σχήμα 1.3** Ο Theodore Solomons δίπλα σε μεταλλική σφαίρα που μόλις έπεσα από τον ουρανό σε μια φάρμα στο Worcester της Νότιας Αφρικής, τον Απρίλιο του 2000.

Η ιδέα της ‘κρίσιμης πυκνότητας’ ήρθε από τον Kessler (1991) για να περιγράψει το όριο της αστάθειας του φαινομένου αυτού. Το γνωστό ως ‘Σύνδρομο Kessler’ προβλέπει ότι ο αριθμός των διαστημικών απορριμμάτων στη LEO (Low Earth Orbit ή Χαμηλή Γήινη Τροχιά) - αντικείμενα μεγέθους  $\geq 10$  cm - θα γνωρίσει γρήγορη μη-γραμμική αύξηση στα επόμενα 200 χρόνια, [22], (βλ. Σχήμα 1.4). Σημειώνεται δε, ότι ο αριθμός των απορριμμάτων σήμερα στη LEO έχει φτάσει το σημείο, όπου το περιβάλλον καθίσταται ασταθές και οι συγκρούσεις γίνονται η κύρια αιτία παραγωγής απορριμμάτων. Παρόλα αυτά, ενώ εκδηλώνεται μεγάλο ενδιαφέρον από πολλούς οργανισμούς διαστήματος ανά τον κόσμο όπως NASA, ASI, ESA, ISRO, JAXA, και UKSA, δεν υπάρχει ακόμα οργανωμένο σχέδιο αντιμετώπισης. Γεγονός είναι επίσης ότι τα διαστημικά σκουπίδια στη LEO θα αυξάνονται ακόμη και με πολύ καλή εφαρμογή των σύγχρονων τρόπων αντιμετώπισης. Για την διαφύλαξη του διαστήματος για τις μελλοντικές γενιές, απαιτούνται πλέον μέτρα πέρα από τα κλασικά, όπως η ενεργή αφαίρεση απορριμμάτων ή Active Debris Removal (ADR). Πρόκειται γενικότερα για μία από τις μεγαλύτερες τεχνολογικές προκλήσεις του 21<sup>ου</sup> αιώνα.

Πολλά ερευνητικά κέντρα εργάζονται σήμερα πάνω στο θέμα και προτείνουν μεθόδους αντιμετώπισης. Ένα προτεινόμενο σύστημα αφαίρεσης διαστημικών σκουπιδιών από το Texas A&M University, το TAMU Space Sweeper with Sling-Sat (βλ. Σχήμα 1.5), αιχμαλωτίζει ένα αντικείμενο και το απελευθερώνει σε χαμηλότερη τροχιά, εκμεταλλευόμενο την ορμή του για να συναντήσει κάποιο άλλο αντικείμενο χωρίς να ξοδευτεί πολύ καύσιμο. Στόχος είναι η αφαίρεση όσο περισσότερων σκουπιδιών με την ελάχιστη ποσότητα καυσίμου, [23]. Μια άλλη ιδέα παρουσιάστηκε τον Φεβρουάριο του 2012, όταν το ελβετικό διαστημικό κέντρο του EPFL, ανακοίνωσε το σχέδιο της αποστολής ενός δορυφόρου αφαίρεσης σκουπιδιών από την χαμηλή τροχιά LEO. Ο δορυφόρος των 11 εκατομμυρίων δολαρίων, με το όνομα CleanSpace One, έχει στόχο να αιχμαλωτίσει και να βγάλει εκτός τροχιάς ένα δορυφόρο- κύβο ακμής 10 [cm], τον Swisscube-1 που εκτοξεύτηκε το 2009. Το πρόγραμμα περιλαμβάνει συνάντηση με το στόχο, σύλληψη του κύβου μέσω ειδικών βραχιόνων, και καταστροφή κατά την είσοδό τους στην ατμόσφαιρα, [24].



**Σχήμα 1.4** Το πλήθος των διαστημικών σκουπιδιών από το 1960 μέχρι το 2210 (εκτιμώμενο).



**Σχήμα 1.5** Αριστερά: Το πρόγραμμα TAMU Space Sweeper with Sling-Sat. Δεξιά: Το πρόγραμμα CleanSpace One.

### 1.3 Σύγχρονα διαστημικά ρομπότ

Εκτός όμως από το επίπεδο των ιδεών, η ανθρωπότητα έχει περάσει εδώ και αρκετά χρόνια στο επίπεδο των πραγματικών δοκιμών αυτόνομων ρομποτικών συστημάτων στο χώρο του διαστήματος. Εδώ συχνά απαντώνται οι όροι 'space rendezvous' [25] και 'docking' (πρόσδεση) διαστημικών συστημάτων. Κάποιες χαρακτηριστικές αποστολές παρουσιάζονται παρακάτω:

#### 1. Demonstration for Autonomous Rendezvous Technology (DART) της NASA

Η αποστολή DART ήταν η πρώτη δοκιμή αυτόνομου rendezvous της NASA στο διάστημα. Στόχος ήταν μια πρώτη επίδειξη της απαραίτητης τεχνολογίας. Το αποτέλεσμα βέβαια δεν ήταν το αναμενόμενο, καθώς στην τελική φάση προσέκρουσε με τον δορυφόρο στόχο. [26]

#### 2. DARPA's Orbital Express

Το Orbital express είχε στόχο την επίδειξη των απαραίτητων τεχνολογιών για συνάντηση, σύλληψη και συντήρηση δορυφόρων σε τροχιά. Εκτελέστηκε επιτυχώς πρόσδεση και αποκόλληση κάτω από διάφορες συνθήκες καθώς και μεταφορά καυσίμου και αντικειμένων από το ένα όχημα στο άλλο, μέσω ρομποτικού χεριού. [27]

### 3. ETS-7 της Ιαπωνικής National Space Development Agency (NASDA)

Το πρόγραμμα ETS-7 περιλάμβανε έναν δορυφόρο κυνηγό κι έναν δορυφόρο στόχο, που επιτυχώς αποσυνδέθηκαν και επανασυνδέθηκαν. Εκτελέστηκαν πολλά πειράματα που ενίσχυσαν τις ελπίδες της Ιαπωνίας για μελλοντικές μη επανδρωμένες αποστολές. [28]

### 4. Automated Transfer Vehicle (ATV) της European Space Agency (ESA)

Τα μη επανδρωμένα οχήματα ATV έχουν στόχο τον εφοδιασμό του διεθνή διαστημικού σταθμού με προμήθειες. Τρία ATVs, τα 'Jules Verne', 'Johannes Kepler' και 'Edoardo Amaldi' προσδέθηκαν στον διαστημικό σταθμό μέχρι το 2008, ενώ προγραμματίζεται η αποστολή άλλων δύο μέχρι το 2015. [29]

### 5. XSS-10 του Air Force Research Laboratory (AFRL)

Ο μικρο-δορυφόρος XSS-10 αποσυνδέθηκε από ένα πύραυλο Delta 2 και επανασυνδέθηκε σε αυτόν, επαναλαμβάνοντας το αρκετές φορές, το 2003. [30]

### 6. XSS-11 του Air Force Research Laboratory (AFRL)

Η αποστολή XSS-11 (2005) είχε στόχο την εκτέλεση εκτεταμένων δοκιμών προσέγγισης (proximity). Είναι ένας μικρο-δορυφόρος, ικανός για αυτόνομη συνάντηση με πολλαπλά διαστημικά αντικείμενα. [31]

### 7. Prisma της Swedish Space Corporation (SSC)

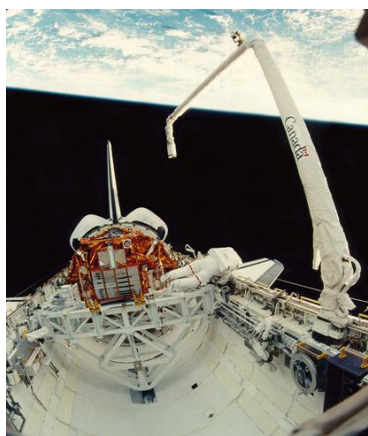
Τον Σεπτέμβριο του 2010 πραγματοποιήθηκε ο διαστημικός χορός των δορυφόρων Mango & Tango σε τροχιά [32]. Πρόκειται για το πρόγραμμα Prisma, το οποίο περιλάμβανε αυτόνομη πτήση σε σχηματισμό, προσέγγιση και συνάντηση (rendezvous) των δύο ρομποτικών συστημάτων.

### 8. Robonaut 2

Το Robonaut 2 είναι το πρώτο ανθρωποειδές ρομπότ το οποίο έστειλε ο άνθρωπος στο διάστημα. Είναι κατασκευασμένο με τέτοιο τρόπο, ώστε να μπορεί να εκτελεί δύσκολες, λεπτές και επικίνδυνες εργασίες αντί των αστροναυτών. Η συγκεκριμένη αποστολή είναι πειραματική και ένας από τους σκοπούς της είναι να μελετηθεί η λειτουργία ρομπότ σε συνθήκες έλλειψης βαρύτητας. [33]

### 9. Canadarm 1,2

Σημαντικά παραδείγματα ρομποτικών συστημάτων στο διάστημα είναι οι δύο βραχίονες Canadarm1 (SRMS) [34] και Canadarm2 [35] της CSA (Canadian Space Agency). Ο βραχίονας Canadarm 1 έχει επισκευάσει και επανατοποθετήσει σε επιθυμητή τροχιά δορυφόρους και έχει χρησιμοποιηθεί σε έντεκα διαστημικές αποστολές. Ο βραχίονας Canadarm 2 βρίσκεται στον Διεθνή Διαστημικό Σταθμό και χρησιμοποιείται για την κατασκευή και τη συντήρηση του σταθμού.



**Σχήμα 1.6** Αριστερά: Ο βραχίονας Canadarm 1 (SRMS). Δεξιά: Ο αστροναύτης Stephen K. Robinson προσδεμένος στον βραχίονα Canadarm2.



## 10. NASA's Robotic Refueling Mission (RRM)

Το πιο σύγχρονο και ελπιδοφόρο παράδειγμα είναι το πρόγραμμα RRM. Στόχος του πειράματος είναι η χρήση ρομπότ στην επισκευή και στον ανεφοδιασμό (καυσίμου) του Διεθνούς Διαστημικού Σταθμού. Τα πειράματα ξεκίνησαν τον Ιανουάριο του 2013 δίνοντας στην πορεία πολύ ενδιαφέροντα αποτελέσματα, [36]. Για τις εργασίες χρησιμοποιήθηκε το Dextre, γνωστό και ως Special Purpose Dexterous Manipulator (SPDM), που αποτελεί το άκρο του βραχίονα Canadarm 2.

### 1.4 Τρόποι εξομοίωσης διαστημικού περιβάλλοντος στη γη

Το διαστημικό περιβάλλον μπορεί να προσομοιωθεί στη γη με ποικίλους τρόπους: συστήματα φυσικής άνωσης (buoyancy tanks), συστήματα αντιστάθμισης βαρύτητας, παραβολικές πτήσεις κ.α. Η κάθε μία από τις παραπάνω μεθόδους παρουσιάζει διάφορα μειονεκτήματα, όπως αντίσταση στο νερό για τα συστήματα φυσικής άνωσης, μηχανικά ιδιόμορφα σημεία και ατέλειες στα συστήματα αντιστάθμισης και στην περίπτωση των παραβολικών πτήσεων, μεγάλο κόστος και λίγο διαθέσιμο χρόνο για πειράματα. Ένα σύστημα που χρησιμοποιείται επίσης συχνά για εξομοίωση των συνθηκών έλλειψης βαρύτητας σε δύο όμως διαστάσεις είναι ο επίπεδος εξομοιωτής αεροστατικών εδράνων. Πιο αναλυτικά:

#### Παραβολικές πτήσεις

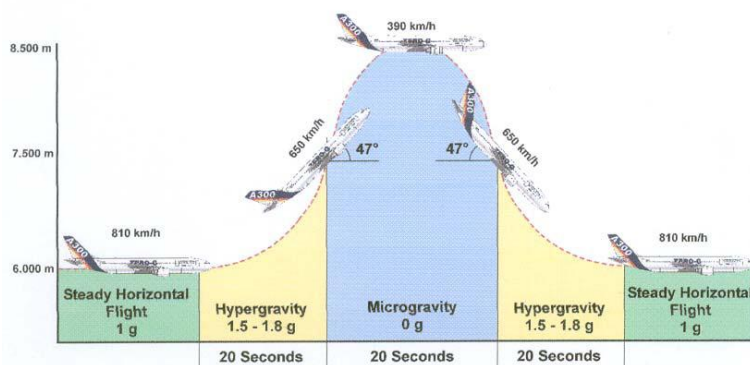
Η παραβολική πτήση συνίσταται στην ειδική πτήση παραβολικής τροχιάς ενός αεροσκάφους (βλ. Σχήμα 1.7). Για μικρό χρονικό διάστημα, εντός του αεροσκάφους επικρατούν συνθήκες μικροβαρύτητας. Αρχικά η παραβολική πτήση χρησιμοποιήθηκε για την εκπαίδευση αστροναυτών. Σήμερα χρησιμοποιείται και για τεχνολογικά πειράματα.

#### Υποβρύχια πειράματα

Όπως η παραβολική πτήση, έτσι και τα υποβρύχια πειράματα χρησιμοποιήθηκαν για πρώτη φορά στην εκπαίδευση αστροναυτών. Κύριο πλεονέκτημα αυτών των πειραμάτων, είναι η απεριόριστη χρονική διάρκειά τους, αλλά και σημαντικό μειονέκτημα ότι η αδράνεια του νερού επηρεάζει αρνητικά την πιστότητα αυτών. Τα ρομπότ πρέπει επίσης, να είναι ειδικής κατασκευής, ούτως ώστε να διασφαλίζεται η στεγανότητα των ηλεκτρονικών εξαρτημάτων τους.

#### Σύστημα ανάρτησης

Ένα άλλο σύστημα, τρισδιάστατης αυτή τη φορά, εξομοίωσης, στηρίζεται στην ανάρτηση του συστήματος εξομοίωσης από την οροφή. Το σύστημα ανάρτησης πρέπει, σε μόνιμη βάση, να εξασκεί δύναμη ίση με το βάρος κάθε συνιστώσας του εξομοιωτή, αλλά αντίθετης κατεύθυνσης, η οποία να διέρχεται από το κέντρο μάζας της συνιστώσας, ούτως ώστε να μην δημιουργούνται ροπές περιστροφής.



Σχήμα 1.7 Παραβολική πτήση αεροσκάφους.

#### Επίπεδοι εξομοιωτές αεροστατικών εδράνων

Το σύστημα που απαντάται πιο συχνά είναι ο επίπεδος εξομοιωτής με χρήση αεροστατικών εδράνων. Η λειτουργία του έγκειται στον εγκλωβισμό ενός λεπτού στρώματος ρευστού, μεταξύ των

εδράνων που φέρουν κινούμενα ρομποτικά συστήματα και μιας τράπεζας στήριξης, μειώνοντας σε πολύ μεγάλο βαθμό τις μηχανικές τριβές. Αρκετά ερευνητικά κέντρα ανά τον κόσμο έχουν κατασκευάσει τέτοια συστήματα, όπως το Πανεπιστήμιο του Stanford, το Τεχνολογικό Ινστιτούτο Μασαχουσέτης, το Τεχνολογικό Ινστιτούτο του Τόκιο, το Πανεπιστήμιο Carnegie-Mellon κ.α. [37] Ένας εξομοιωτής αυτού του τύπου εξετάζεται και στην παρούσα εργασία. Τρία παραδείγματα τέτοιων συστημάτων περιγράφονται παρακάτω.

### 1. Πρόγραμμα SPHERES του MIT

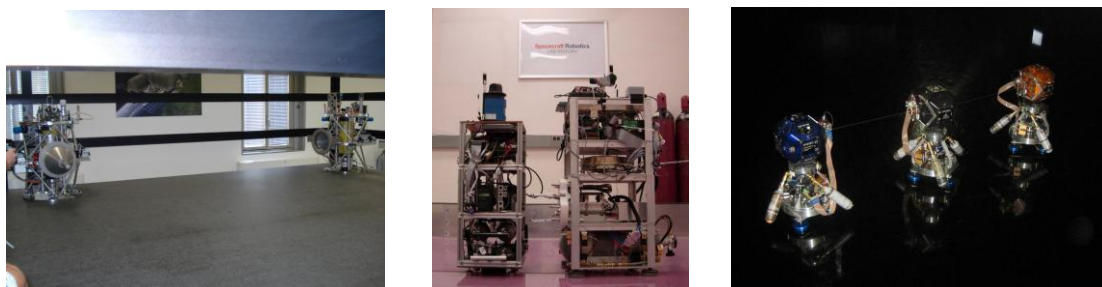
Το Εργαστήριο Διαστημικών Συστημάτων του MIT ανέπτυξε τον επίπεδο εξομοιωτή SPHERES (Synchronized Position Hold Engage and Reorient Experimental Satellites), έτσι ώστε να παρέχει στους ερευνητές μια πλατφόρμα πειραμάτων πάνω στον έλεγχο αυτόνομων συνεργαζόμενων διαστημικών συστημάτων (βλ. Σχήμα 1.8). Αποτελείται από τρία πλήρως αυτόνομα ρομπότ, που έχουν σχήμα σφαίρας. [38]

### 2. Επίπεδος διαστημικός εξομοιωτής AMPHIS

Η πλατφόρμα εξομοίωσης AMPIS (Autonomous Multi-Agent Physically Interacting Spacecraft) [13], [14] του Naval Postgraduate School στην Καλιφόρνια εξετάζει τα προβλήματα αλληλεπίδρασης πολλαπλών διαστημικών συστημάτων κατά την πτήση τους σε πολύ μικρές μεταξύ τους αποστάσεις. Αποτελείται από ρομπότ που φέρουν υποσυστήματα που είναι παρόμοια με αυτά των πραγματικών διαστημικών συστημάτων (βλ. Σχήμα 1.8).

### 3. Επίπεδος διαστημικός εξομοιωτής της Thales Alenia Space

Ο εξομοιωτής [39] αναπτύσσεται στο πλαίσιο του προγράμματος STEPS (Systems and Technologies for Space Exploration) και αποτελείται από ένα λείο δάπεδο και δύο ρομποτικά οχήματα: ενός κυνηγού ή chaser και ενός στόχου ή target (βλ. Σχήμα 1.8).



*Σχήμα 1.8 Από τα αριστερά προς τα δεξιά α) Επίπεδος εξομοιωτής της Thales Alenia Space β) Επίπεδος εξομοιωτής AMPHIS γ) Το πρόγραμμα SPHERES του MIT.*

## 1.5 Δομή Εργασίας

Στο **πρώτο κεφάλαιο**, γίνεται μια εισαγωγή στο θέμα των διαστημικών ρομπότ. Εξηγείται γιατί υπάρχουν, δίνονται παραδείγματα τέτοιων ρομπότ που λειτουργούν αυτή τη στιγμή σε τροχιά και αναφέρονται τα ερευνητικά κίνητρα που εμπεριέχει η ανάπτυξη τους. Αναφορά γίνεται επίσης στο φαινόμενο των ‘διαστημικών απορριμμάτων’, περιγράφονται οι κίνδυνοι γύρω από αυτό και εξηγείται η σχέση τους με τα ρομπότ σε τροχιά.

Στο **δεύτερο κεφάλαιο**, περιγράφεται ο επίπεδος διαστημικός εξομοιωτής του εργαστηρίου. Ιδιαίτερη σημασία δίνεται στην περιγραφή των συστημάτων του πρώτου ρομπότ του εξομοιωτή, αφού αυτό σχετίζεται άμεσα με την ανάπτυξη του δεύτερου ρομπότ. Τέλος, περιγράφονται οι σχεδιαστικές απαιτήσεις του τελευταίου σε ηλεκτρικό, ηλεκτρονικό και προγραμματιστικό επίπεδο.

Στο **τρίτο κεφάλαιο**, μελετώνται όλοι οι πιθανοί τρόποι ανάπτυξης ενός τέτοιου συστήματος σε επίπεδο υλικών και λογισμικού. Περιγράφεται η μέθοδος του σχεδιασμού βασισμένου στο μοντέλο (model-based design), η οποία χρησιμοποιείται όλο και περισσότερο στην ανάπτυξη σύγχρονων ρομποτικών συστημάτων. Στη συνέχεια, επιλέγονται τα εργαλεία που θα χρησιμοποιηθούν, με

κυρίαρχο το πακέτο λογισμικού xPC Target της Mathworks. Τέλος, γίνεται ένας πρώτος σχεδιασμός του συστήματος και επιλέγονται τα κατάλληλα ηλεκτρικά/ηλεκτρονικά εξαρτήματα (υπολογιστική μονάδα, κάρτα εισόδων/εξόδων, πλακέτες παροχής ισχύος κ.α.).

Στο **τέταρτο κεφάλαιο**, παρουσιάζεται το σύστημα εντοπισμού των ρομπότ μέσω μιας κάμερας, που τα παρακολουθεί στερεωμένη πάνω από την τράπεζα του γρανίτη. Περιγράφονται διάφορες μέθοδοι εντοπισμού και σχεδιάζονται οι αντίστοιχοι αλγόριθμοι επεξεργασίας εικόνας. Τέλος, σχεδιάζεται και υλοποιείται η ασύρματη επικοινωνία μεταξύ του υπολογιστή της κάμερας και του υπολογιστικών συστημάτων των ρομπότ.

Στο **πέμπτο κεφάλαιο**, γίνεται ο σχεδιασμός και η ανάπτυξη ενός δεύτερου συστήματος εντοπισμού του καινούριου ρομπότ μέσω οπτικών αισθητήρων που προσαρτώνται στο σώμα του. Στη συνέχεια, σχεδιάζονται διάφοροι αλγόριθμοι επεξεργασίας των δεδομένων των αισθητήρων με στόχο τον ακριβή προσδιορισμό θέσης και προσανατολισμού του ρομπότ.

Στο **έκτο κεφάλαιο**, γίνεται ο σχεδιασμός και η ανάπτυξη του λογισμικού του ρομπότ, στο πλαίσιο του xPC Target. Υλοποιούνται οι αλγόριθμοι που παρουσιάστηκαν στα προηγούμενα κεφάλαια, και τρέχουν στο υπολογιστικό σύστημα του ρομπότ σε πραγματικό χρόνο. Τέλος, παρουσιάζονται οι τρόποι επικοινωνίας και χειρισμού του ρομπότ μέσω εξωτερικού υπολογιστή, που τρέχει περιβάλλον εικονικής πραγματικότητας.

Στο **έβδομο κεφάλαιο**, περιγράφεται η διαδικασία του σχεδιασμού των απαραίτητων για τη λειτουργία του ρομπότ ηλεκτρονικών πλακετών. Τέλος, οι πλακέτες εκτυπώνονται και εγκαθίστανται στο ρομπότ.

Στο **όγδοο κεφάλαιο**, σχεδιάζονται διάφορα πειράματα κίνησης με ανάδραση από τα συστήματα εντοπισμού. Παρουσιάζονται τα αποτελέσματα των πειραμάτων και αξιολογούνται οι αλγόριθμοι που αναπτύχθηκαν στα προηγούμενα κεφάλαια.

Στο **ένατο κεφάλαιο**, παρουσιάζονται τα συμπεράσματα που προέκυψαν από την εκπόνηση της παρούσας εργασίας και προτείνονται κάποιες ενέργειες σαν μελλοντική εργασία.





# ΚΕΦΑΛΑΙΟ 2

## Περιγραφή επίπεδου διαστημικού εξομοιωτή του εργαστηρίου – Σχεδιαστικές απαιτήσεις νέου ρομπότ

### 2.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται μια περιγραφή του διαστημικού εξομοιωτή του Εργαστηρίου Αυτομάτου Ελέγχου του ΕΜΠ, και ειδικά του πρώτου ρομπότ που σχεδιάστηκε και κατασκευάστηκε πριν μερικά χρόνια από την ομάδα. Στη συνέχεια του κεφαλαίου, αναφέρονται και αναλύονται οι αρχικές σχεδιαστικές απαιτήσεις για το νέο (δεύτερο) ρομπότ, του οποίου το ηλεκτρικό/ ηλεκτρονικό υποσύστημα και ο προγραμματισμός, είναι και το αντικείμενο της παρούσας εργασίας.

Σημειώνεται ότι, η ανάπτυξη του συστήματος δεν άρχισε από μηδενική βάση, αφού υπήρχε διάχυτη στο εργαστήριο η εμπειρία από τις επιτυχίες και τα λάθη κατά τον σχεδιασμό του πρώτου ρομπότ. Εκτός αυτού, ο Σ. Ανδρέου [15] είχε πρόσφατα σχεδιάσει και κατασκευάσει τον μεταλλικό σκελετό και το πνευματικό υποσύστημα του ρομπότ, γεγονός που από τη μία επέβαλε περιορισμούς στον

ηλεκτρικό/ ηλεκτρονικό σχεδιασμό, από την άλλη όμως, παρείχε μία στέρεα βάση για την θεμελίωση και την υλοποίηση της διπλωματικής αυτής εργασίας.

Όπως αναφέρθηκε και προηγουμένως, απαραίτητη για τον σωστό σχεδιασμό του νέου ρομπότ, κρίθηκε η καλή γνώση των υπάρχοντων συστημάτων και η ανάλυση των πλεονεκτημάτων και των μειονεκτημάτων τους. Για τον λόγο αυτό, ακολουθεί μια περιγραφή του διαστημικού εξομοιωτή του εργαστηρίου στη μορφή που είχε αυτός τη στιγμή που ξεκίνησε η παρούσα εργασία. Αντίστοιχη παρουσίαση έχει γίνει πιο αναλυτικά στις εργασίες [1] και [2] .

## 2.2 Ο επίπεδος διαστημικός εξομοιωτής του εργαστηρίου

Ο διαστημικός εξομοιωτής του εργαστηρίου αποτελείται από ένα πλήρως αυτόνομο ρομπότ (το πρώτο), το οποίο αιωρείται σε ύψος 10 [μm] χρησιμοποιώντας αεροέδρανα πάνω σε μία τράπεζα γρανίτη εξαιρετικά μικρής τραχύτητας. Με αυτόν τον τρόπο, εξαλείφεται η τριβή και επιτυγχάνεται εξομοίωση συνθηκών έλλειψης βαρύτητας σε δύο διαστάσεις. Η επιλογή και ανάπτυξη του επίπεδου διαστημικού εξομοιωτή με αεροέδρανα έγινε με βάση τα πλεονεκτήματα τα οποία παρουσιάζει αυτή η μέθοδος:

- Ρεαλιστική εξομοίωση συνθηκών έλλειψης βαρύτητας σε δύο διαστάσεις
- Σχετικά εύκολος στην κατασκευή, στιβαρός και χαμηλού κόστους
- Σχετικά μικρός σε διαστάσεις και άμεσα προσβάσιμος
- Υποσυστήματα του ρομπότ αντίστοιχα με αυτά των πραγματικών διαστημικών συστημάτων
- Ενεργειακή και υπολογιστική αυτονομία καθώς και πλήρης αυτονομία κινήσεων του ρομπότ
- Δυνατότητα διεξαγωγής απεριόριστων και διαφορετικών πειραμάτων
- Χαμηλή κατανάλωση ενέργειας

Η πρωτοτυπία του συγκεκριμένου εξομοιωτή έγκειται στο ότι το ρομπότ είναι μικρής μάζας (περίπου 15 kg), πλήρως αυτόνομο και με υποσυστήματα ανάλογα ενός πραγματικού διαστημικού ρομποτικού συστήματος. Οι συνιστώσες του συστήματος παρουσιάζονται παρακάτω.

### 2.2.1 Τράπεζα γρανίτη

Μια τράπεζα από γρανίτη αποτελεί το επίπεδο, πάνω στο οποίο πραγματοποιείται η κίνηση των ρομπότ. Έχει διαστάσεις 2.2m x 1.8m x 0.3m και βάρος περίπου 3.5 tn. Η τραχύτητα της τράπεζας είναι μικρότερη των 5 [μm]. Η κλίση της είναι μικρότερη από 0.01° και ρυθμίζεται με μεταβολή των ποδιών στήριξης.



Σχήμα 2.1 Τράπεζα γρανίτη του εργαστηρίου.

### 2.2.2 Ρομπότ

Τελικός στόχος της ομάδας είναι η ύπαρξη δύο ρομπότ τα οποία θα αιωρούνται και θα (συν)εργάζονται πάνω στην τράπεζα καθώς και ενός σώματος-στόχου (target). Το πρώτο ρομπότ είναι σχεδόν ολοκληρωμένο και λειτουργικό, ενώ μέρος του σχεδιασμού και της κατασκευής του δεύτερου απασχολεί την παρούσα εργασία.

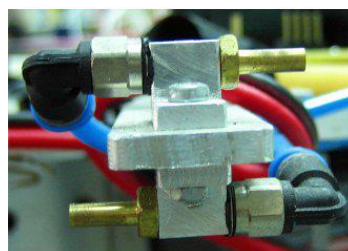
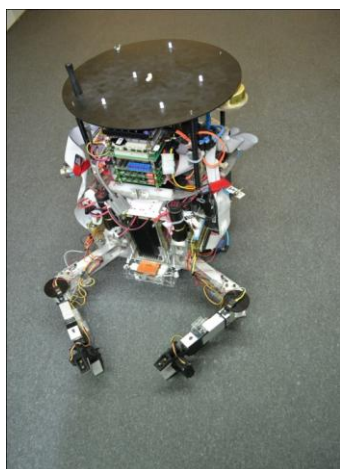
Το πρώτο ρομπότ (βλ. Σχήμα 2.2), το οποίο μπορεί να θεωρηθεί ολοκληρωμένο, αποτελείται από τέσσερα υποσυστήματα: το μηχανολογικό, το ρευστομηχανικό, και το ηλεκτρικό/ ηλεκτρονικό, και το λογισμικό του (software).

#### Μηχανολογικό υποσύστημα

Το ρομπότ είναι κατασκευασμένο από αλουμίνιο, το οποίο παρέχει στιβαρότητα ενώ είναι ταυτοχρόνως ελαφρύ. Η συνολική μάζα του ρομπότ είναι περίπου 15kg. Όλα τα υποσυστήματα του ρομπότ εδράζονται πάνω στον κυλινδρικού σχήματος σκελετό του. Το ρομπότ φέρει δύο βραχίονες, δύο βαθμών ελευθερίας για να αλληλεπιδρά με το περιβάλλον του. Διαθέτει επίσης, ένα σφόνδυλο αντίδρασης (reaction wheel), ο οποίος χρησιμοποιείται ως εναλλακτικός τρόπος για την περιστροφή του, με σκοπό την ελαχιστοποίηση της κατανάλωσης αερίου (βλ. Σχήμα 2.3). Ο σχεδιασμός του μηχανολογικού υποσυστήματος περιγράφεται αναλυτικά στην εργασία [5].

#### Ρευστομηχανικό υποσύστημα

Επάνω στο ρομπότ βρίσκεται μία φιάλη με CO<sub>2</sub>, μάζας 550 gr και πίεσης 60 bar (στους 200°C), η οποία παρέχει αέριο σε τρία αεροέδρανα (airbearings) διαμέτρου 25 [mm] από πορώδη γραφίτη, που ανυψώνουν το ρομπότ κατά 10 [μm] πάνω από την τράπεζα. Με αυτόν τον τρόπο το ρομπότ αιωρείται πάνω σε ένα λεπτό στρώμα αερίου και επιτυγχάνεται κίνηση χωρίς τριβή. Αυτός είναι και ο τρόπος εξομοίωσης μηδενικής βαρύτητας σε δύο διαστάσεις (στο επίπεδο). Το ρομπότ διαθέτει επίσης, τρία ζεύγη προωθητηρίων (thrusters), που τροφοδοτούνται από τη φιάλη με CO<sub>2</sub> (βλ. Σχήμα 2.2). Κάθε ζεύγος προωθητηρίων είναι δύο ακροφύσια τα οποία ανοίγουν και κλείνουν μέσω βαλβίδων. Τα δύο προωθητήρια είναι τοποθετημένα αντίθετα μεταξύ τους έτσι ώστε να αποδίδουν ένας ζεύγος αντίθετων ωθήσεων. Τα τρία ζεύγη προωθητηρίων είναι τοποθετημένα υπό γωνία 120° πάνω στον σκελετό του ρομπότ.



**Σχήμα 2.2** Αριστερά: Πρώτο ρομπότ σε λειτουργία. Δεξιά: ζεύγος προωθητηρίων (thrusters).

## Ηλεκτρικό/ Ηλεκτρονικό υποσύστημα

Η ενεργειακή αυτονομία του ρομπότ επιτυγχάνεται με την χρήση μπαταριών πολυμερούς λιθίου (LiPo). Το σύστημα διαχείρισης ισχύος χωρίζεται σε δύο απομονωμένα κυκλώματα: στο χαμηλής τάσης (8V), το οποίο τροφοδοτεί το υπολογιστικό σύστημα και στο υψηλής τάσης (33V), το οποίο τροφοδοτεί τους σερβοκινητήρες και το σύστημα οδήγησης των προωθητήρων (thrusters).

**Κεντρική μονάδα επεξεργασίας (CPU):** Η υπολογιστική αυτονομία του ρομπότ επιτυγχάνεται με έναν υπολογιστή PC-104 ο οποίος βρίσκεται πάνω στο ρομπότ. Η πλακέτα CPU είναι της εταιρίας RTD (Intel atom n450 1,66GHz 2GB RAM) και επιλέχθηκε με κριτήριο την υψηλή υπολογιστική της απόδοση, τη χαμηλή κατανάλωση ενέργειας, την αξιοπιστία, το χαμηλό της βάρος, τις μικρές διαστάσεις, το χαμηλό θόρυβο και την αντοχή της. Ο προγραμματισμός του PC-104 γίνεται σε γλώσσα C. Το PC-104 είναι σε μορφή πύργου και αποτελείται επίσης από τις παρακάτω μονάδες (ή κάρτες).

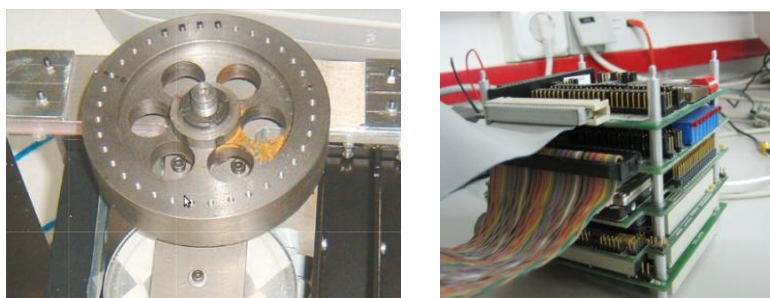
**Κάρτα Incremental Encoder:** Κάρτα ανάγνωσης αυξητικών παλμογεννητριών (encoder) από τους κινητήρες, μοντέλο DM6606HR. Κάθε κάρτα μπορεί να διαβάσει μέχρι τρεις encoders 16-bit εισόδους και διαθέτει 14 16-bit timers, 24 ψηφιακές I/O γραμμές, ένα 8MHz ρολόι και 4 interrupt circuits. Από τις 24 ψηφιακές I/O γραμμές, οι 6 μπορούν να προγραμματιστούν μόνο ως input και οι 2 μπορούν να προγραμματιστούν ως input ή output. Το PC-104 έχει δύο κάρτες Incremental Encoder, συνεπώς μπορεί να διαβάσει έξι encoder από τους αντίστοιχους κινητήρες και διαθέτει 48 ψηφιακές I/O γραμμές.

**Κάρτα Analog Output:** Κάρτα ψηφιοαναλογικού μετατροπέα (DAC), μοντέλο DM6604HR (8DAC). Η κάρτα DAC διαθέτει οκτώ 12-bit αναλογικές εξόδους γρήγορης απόκρισης και με εύρος τάσης  $\pm 5$ ,  $\pm 10$ , [0,5] ή [0,10]V. Η κάρτα έχει επίσης 24 ψηφιακές I/O γραμμές και τα software interrupts IRQ2-IRQ7, IRQ9-IRQ15.

**Κάρτα wireless:** Το PC-104 περιλαμβάνει επίσης μία κάρτα για ασύρματη σύνδεση (μοντέλο WLAN17202ER) συμβατή με το πρωτόκολλα 802.11a/b/g η οποία μπορεί να συνδέσει μεταξύ τους σε δίκτυο πολλαπλά μηχανήματα.

**Σκληρός δίσκος:** Τέλος, το PC-104 συμπληρώνεται από μία κάρτα η οποία είναι ο σκληρός δίσκος, χωρητικότητας 20GB (μοντέλο CMT3106ER). Ως εναλλακτική λύση αντί για τον σκληρό δίσκο, μπορεί να τοποθετηθεί μία κάρτα PCMCIA (MPL, IDE2CF) με μία μνήμη flash για να μειωθεί η κατανάλωση ενέργειας από το PC-104. Χρησιμοποιείται επίσης ένας σκληρός δίσκος SSD, μοντέλο SSD 60 GB CORSAIR FORCE SERIES F60.

Στο Σχήμα 2.3 στα δεξιά, φαίνεται το PC-104. Από κάτω προς τα πάνω διακρίνονται: η κάρτα για ασύρματη σύνδεση, η κάρτα κεντρικής μονάδας επεξεργασίας, η κάρτα με το σκληρό δίσκο, η κάρτα ψηφιοαναλογικού μετατροπέα και η μία από τις δύο κάρτες ανάγνωσης αυξητικών παλμογεννητριών. Αριστερά, φαίνεται ο σφόνδυλος αντίδρασης (reaction wheel). Όταν αυτός περιστρέφεται, το ρομπότ τείνει να περιστραφεί με αντίθετη φορά γύρω από το κέντρο βάρους του.



**Σχήμα 2.3** Αριστερά: Σφόνδυλος αντίδρασης. Δεξιά: Πύργος καρτών PC-104.

**Οπτικοί αισθητήρες:** Οι οπτικοί αισθητήρες είναι όμοιοι με αυτούς που χρησιμοποιούνται σε ένα καλό οπτικό ποντίκι για Η/Υ. Οι αισθητήρες χρησιμοποιούνται ως εναλλακτική μέθοδος για τον εντοπισμό της θέσης και του προσανατολισμού του ρομπότ. Η μέθοδος αυτή δεν προσθέτει μεγάλη μάζα και όγκο και είναι πολύ οικονομική με πολύ γρήγορη απόκριση. Ωστόσο, συγκεντρώνει μετά από κάποιο διάστημα οδομετρικό σφάλμα. Για την ελαχιστοποίηση του σφάλματος χρησιμοποιούνται τρεις οπτικοί αισθητήρες τοποθετημένοι στην βάση του ρομπότ. Οι αισθητήρες αυτοί πρέπει να βρίσκονται σε πολύ μικρή απόσταση από την επιφάνεια κίνησης.

**Άλλοι αισθητήρες:** Το ρομπότ διαθέτει διάφορους αισθητήρες για έλεγχο της εσωτερικής κατάστασής του: αισθητήρες τάσης των μπαταριών, πίεσης και ποσότητας CO<sub>2</sub>, αισθητήρες δύναμης στις αρπάγες, αισθητήρες Hall για την μέγιστη έκταση των βραχιόνων.

## Λογισμικό

Στο PC104 τρέχει τη διανομή Arch Linux μαζί με το Xenomai για την υλοποίηση των εφαρμογών πραγματικού χρόνου. Η διανομή Arch Linux επιλέχθηκε διότι είναι ιδιαίτερα ελαφριά και δεν επιβαρύνει το σύστημα με πλεονάζουσες διεργασίες. Το Xenomai είναι open source και έχει πολύ καλό documentation. Ένα πολύ σημαντικό χαρακτηριστικό του Xenomai είναι ότι επιτρέπει την ανάπτυξη αυστηρών εφαρμογών πραγματικού χρόνου σε kernel space και σε user space. Πιο αναλυτική περιγραφή υπάρχει στην εργασία [1].

### 2.2.3 Σύστημα παρακολούθησης μέσω κάμερας

Είναι το σύστημα εκείνο, το οποίο είναι επιφορτισμένο με την παρακολούθηση του εξομοιωτή, ούτως ώστε να μπορεί ανά πάσα χρονική στιγμή να τον πληροφορεί για την κατάστασή του. Ο κύριος ρόλος είναι ο προσδιορισμός των συντεταγμένων και του προσανατολισμού των σωμάτων που κινούνται πάνω στην τράπεζα του γρανίτη, ως προς κάποιο αδρανειακό σύστημα αναφοράς (γωνία τράπεζας).

Η κάμερα είναι τοποθετημένη επάνω από την τράπεζα έτσι ώστε να καλύπτει όλη την επιφάνεια εργασίας. Το σύστημα της κάμερας εντοπίζει τρία (κατ' ελάχιστο) LEDs, τα οποία είναι τοποθετημένα στην επάνω πλάκα του ρομπότ. Η εικόνα από την κάμερα αποστέλλεται σε εξωτερικό υπολογιστή και με τεχνικές επεξεργασίας εικόνας λαμβάνεται η θέση και ο προσανατολισμός του ρομπότ. Στην συνέχεια, τα δεδομένα της θέσης του ρομπότ αποστέλλονται μέσω TCP/IP στο PC-104. Για την διαδικασία αυτή έχουν αναπτυχθεί (α) ένα πρόγραμμα για την λήψη της εικόνας, (β) ένα πρόγραμμα για την επεξεργασία της εικόνας και (γ) ένα πρόγραμμα για την μεταφορά των δεδομένων. Τα προηγούμενα έχουν συνταχθεί σε γλώσσα C++ [3]

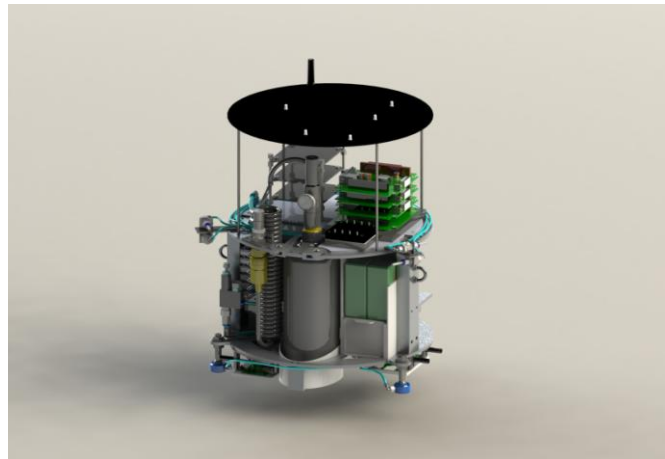
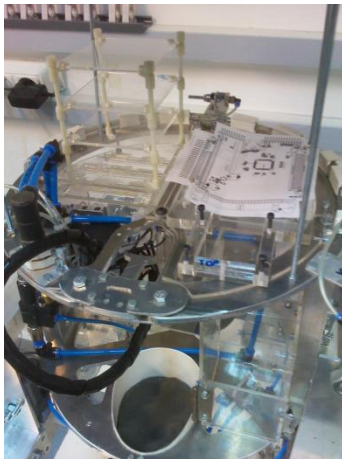
Η κάμερα είναι της εταιρείας Matrix Vision, μοντέλο mvBlueFOX. Έχει ανάλυση 1600x1200, απεικόνιση στην κλίμακα του γκρι με μέγεθος στοιχείου 8 bits, ρυθμό ανανέωσης 8 fps, αισθητήρα CCD (SONY, μοντέλο ICX274AL/AQ) και συνδέεται μέσω USB 1.1/2.0 (έως 480 Mbits/sec). Η κάμερα και ο οπτικός φακός επιλέχθηκαν έτσι ώστε από το ύψος στήριξης της διάταξης πάνω από την τράπεζα, να απεικονίζεται σε ένα καρέ ολόκληρη η περιοχή δράσης με ικανοποιητική ανάλυση.

## 2.3 Σχεδιαστικές απαιτήσεις νέου ρομπότ σε ηλεκτρικό, ηλεκτρονικό και προγραμματιστικό επίπεδο

Το νέο ρομπότ πρέπει να έχει την ίδια μορφή και σχεδόν τις ίδιες δυνατότητες με το πρώτο. Οι λειτουργίες που περιγράφηκαν παραπάνω πρέπει να ενσωματωθούν με παρόμοιο τρόπο στο ρομπότ που θα αναπτυχθεί σε αυτή την εργασία. Στο Σχήμα 2.4 φαίνεται το σχέδιο σε CAD και ο τελικός μεταλλικός σκελετός, πάνω στον οποίο θα αναπτυχθούν τα υπόλοιπα υποσυστήματα του νέου ρομπότ. Παρόλο όμως, που το τελικό αποτέλεσμα πρέπει να ομοιάζει στο πρώτο ρομπότ, δεν ορίζονται εξαρχής περιορισμοί στον τρόπο σχεδιασμού και στην διαδικασία ανάπτυξης. Το ρομπότ μελετάται και σχεδιάζεται από την αρχή.

Συγκεκριμένα οι αρχικές σχεδιαστικές απαιτήσεις σε ηλεκτρικό/ ηλεκτρονικό και προγραμματιστικό επίπεδο είναι οι εξής:

- Δυνατότητα εύρεσης θέσης με οπτικούς αισθητήρες
- Ασύρματη λήψη δεδομένων από σύστημα εντοπισμού μέσω κάμερας
- Έλεγχος βαλβίδων (thrusters) με παλμούς PWM και ενίσχυση στα 24V
- Έλεγχος κινητήρων των βραχιόνων και του σφονδύλου αντίδρασης (reaction wheel)
- Λήψη δεδομένων από encoders κινητήρων
- Προγραμματισμός υπολογιστικής μονάδας και ενσωμάτωση των παραπάνω σε ένα ενιαίο περιβάλλον
- Σχεδιασμός και κατασκευή απαραίτητων ηλεκτρονικών κυκλωμάτων (PCBs)



**Σχήμα 2.4** Αριστερά: Το σασί του νέου ρομπότ όπως κατασκευάστηκε από τον Σ. Ανδρέου. Δεξιά: 3D σχέδιο του νέου ρομπότ.

# ΚΕΦΑΛΑΙΟ 3

## Αρχικός ηλεκτρικός/ ηλεκτρονικός σχεδιασμός

### 3.1 Εισαγωγή

Στο κεφάλαιο αυτό, μελετώνται οι τρόποι με τους οποίους μπορούν να ικανοποιηθούν οι σχεδιαστικές απαιτήσεις που αναφέρθηκαν προηγουμένως. Για να γίνει αυτό, το ρομπότ μελετάται ως συνολικό σύστημα και αποφεύγεται η καθαρή διάκριση σε ηλεκτρικό, ηλεκτρονικό και προγραμματιστικό επίπεδο. Δεδομένου ότι, το κάθε επίπεδο εξαρτάται από τα υπόλοιπα κατά τη διάρκεια της ανάπτυξης του συνόλου, κρίνεται ότι ο αυτοτελής σχεδιασμός του καθενός ξεχωριστά και ανεξάρτητα μπορεί να οδηγήσει σε μεγάλους χρόνους ανάπτυξης. Ειδικά στην εργασία αυτή, κατά την οποία, το ρομπωτικό σύστημα μελετάται, σχεδιάζεται και υλοποιείται σε ηλεκτρικό, ηλεκτρονικό και προγραμματιστικό επίπεδο από έναν μόνο μηχανικό (τον συγγραφέα), είναι επιτακτικό, ο σχεδιασμός να γίνει όσο το δυνατόν συνολικά, λαμβάνοντας υπόψη τις απαιτούμενες ενέργειες σε όλα τα επίπεδα.



Αρχικά, μελετώνται οι δυνατότητες που υπάρχουν σε επίπεδο υλικού (hardware). Ύστερα, δίνοντας έμφαση στον προγραμματισμό πραγματικού χρόνου, μελετώνται οι δυνατότητες που υπάρχουν σήμερα σε επίπεδο λογισμικού και πώς αυτές σχετίζονται με το υλικό. Τέλος παρουσιάζεται ο πλέον σύγχρονος τρόπος ανάπτυξης πολύπλοκων συστημάτων αυτομάτου ελέγχου, γνωστός ως σχεδιασμός βασισμένος στο μοντέλο (model-based design).

### 3.2 Υλικό (Hardware)

Η κύρια μονάδα του υπολογιστικού συστήματος ενός ρομπότ είναι η μονάδα επεξεργασίας. Εκεί δηλαδή που θα υλοποιούνται οι αλγόριθμοι ελέγχου του ρομπότ και θα γίνεται η διαχείριση όλων των εργασιών που σχετίζονται με τα διάφορα υποσυστήματά του (αισθητήρες, επενεργητές, επικοινωνία). Στη σύγχρονη ρομποτική, λόγω και της μεγάλης εξέλιξης των υπολογιστικών συστημάτων, οι επιλογές σε αυτό το επίπεδο είναι αρκετές και εκτείνονται σε ένα μεγάλο εύρος δυνατοτήτων και κόστους. Σε επίπεδο λοιπόν υλικού (hardware), διακρίνονται αρχικά δύο τρόποι υλοποίησης:

#### 3.2.1 Υλοποίηση σε επίπεδο μικροελεγκτών

Αποτελεί μια μέθοδο ανάπτυξης σε χαμηλό επίπεδο. Αυτός ο τρόπος υλοποίησης περιλαμβάνει επιλογή και προγραμματισμό των κατάλληλων μικροελεγκτών στους οποίους θα τρέχουν όλοι οι αλγόριθμοι επεξεργασίας και αυτομάτου ελέγχου. Η συνολική ανάπτυξη περιλαμβάνει εκτύπωση κατάλληλων πλακετών για ενσωμάτωση ενός ή περισσότερων μικροελεγκτών μαζί με τα απαραίτητα περιφερειακά ηλεκτρονικά.

Ένας μικροελεγκτής είναι στην ουσία ένας υπολογιστής σε μια πιο συμπαγή μορφή. Έχει μία μονάδα επεξεργασίας (CPU) στην οποία εκτελούνται τα προγράμματα, μία μνήμη στην οποία αποθηκεύονται και ανανεώνονται κατά την εκτέλεση του προγράμματος οι διάφορες μεταβλητές καθώς και θύρες εισόδων – εξόδων (I/O ports). Η κύρια διαφορά τους από τους σύγχρονους υπολογιστές έγκειται στην περιορισμένη τους μνήμη (της τάξης μερικών Kbytes) ενώ δεν έχουν σκληρό δίσκο. Τέλος, λειτουργούν με χαμηλή ισχύ (της τάξης των 50 [mW]) σε σύγκριση με τα 50 [W] των ηλεκτρονικών υπολογιστών, και έχουν μικρότερη ταχύτητα επεξεργασίας δεδομένων.

Οι μικροελεγκτές είναι αναπόσπαστο κομμάτι όλων των σύγχρονων ηλεκτρονικών συσκευών. Ο ανταγωνισμός επομένως στη βιομηχανία μικροελεγκτών είναι πολύ ισχυρός κι έτσι, η αγορά βρίθκει από μοντέλα διαφορετικών δυνατοτήτων, αλλά σχεδόν πάντα χαμηλού κόστους (βλ. Σχήμα 3.1).



**Σχήμα 3.1** Διάφορα μοντέλα μικροελεγκτών.

Ο προγραμματισμός ενός μικροελεγκτή απαιτεί πρώτον, καλές γνώσεις ηλεκτρονικής και προγραμματισμού, και δεύτερον, τα απαραίτητα εργαλεία ανάπτυξης. Τα σημαντικά πλεονεκτήματα που παρουσιάζει η μέθοδος αυτή, είναι το πολύ χαμηλό κόστος, ο πλήρης έλεγχος του συστήματος σε χαμηλό επίπεδο, και οι απεριόριστες δυνατότητες κατά την ανάπτυξη. Πρέπει παρόλα αυτά να σημειωθεί ότι ο χρόνος ανάπτυξης ενός πολύπλοκου συστήματος μπορεί να είναι πολύ μεγάλος και



η διαδικασία αρκετά επίπονη (ανάλογα πάντα και με την εμπειρία). Ένα άλλο αρνητικό σημείο είναι η δυσκολία στην μελλοντική τροποποίηση του συστήματος (από τον κατασκευαστή και κυρίως από άλλους) και ο μικρός βαθμός επεκτασιμότητας του.

### 3.2.2 Χρήση Single Board Computer (SBC)

Ένας άλλος τρόπος υλοποίησης που επίσης χρησιμοποιείται κατά κόρον στη ρομποτική, είναι η χρήση πλακετών στις οποίες εμπεριέχεται ένα ολόκληρο υπολογιστικό σύστημα. Συνήθως περιλαμβάνουν ένα δυνατό επεξεργαστή (π.χ. ARM, Intel κ.α.), μνήμη, ψηφιακές και αναλογικές εισόδους/εξόδους και υποστηρίζουν διάφορους τρόπους επικοινωνίας (π.χ. μέσω δικτύου LAN, σειριακή επικοινωνία, USB κ.α.). Οι πλακέτες αυτές λειτουργούν τρέχοντας κάποιο λειτουργικό σύστημα (π.χ. Linux), ενώ συνδέονται κανονικά στα κλασικά περιφερειακά ενός Η/Υ (π.χ. οθόνη, ηχεία κ.α.)

Με την ραγδαία εξέλιξη της τεχνολογίας των ηλεκτρονικών, οι πλακέτες τέτοιου τύπου μικραίνουν συνεχώς σε μέγεθος και κοστίζουν όλο και λιγότερο. Ένα χαρακτηριστικό παράδειγμα αποτελεί η πλακέτα Raspberry Pi (βλ. Σχήμα 3.2) που αποτελεί έναν ολοκληρωμένο υπολογιστή σε μέγεθος πιστωτικής κάρτας, ενώ κοστίζει μόλις 50 Ευρώ. Ο ανταγωνισμός στην αγορά είναι γενικά πολύ μεγάλος, με αποτέλεσμα την μεγάλη ποικιλία τέτοιων πλακετών. Ένας βασικός διαχωρισμός που γίνεται, είναι με βάση το φορμάτ της κάθε πλακέτας. Αυτό στην ουσία δηλώνει τις διαστάσεις της και κάποια ακόμα χαρακτηριστικά της. Η αγορά βρίθκει από πλακέτες διαφορετικών φορμάτ, ιδιοτήτων και κόστους. Ένα πολύ γνωστό και αξιόπιστο φορμάτ είναι το PC-104, που παρουσιάζεται και παρακάτω.

Τα πλεονεκτήματα που συγκεντρώνει αυτός ο τρόπος ανάπτυξης ενός ρομποτικού συστήματος είναι πρώτον η ασφάλεια και η αξιοπιστία που παρέχει ένας υπολογιστής φορτωμένος με κάποιο λειτουργικό σύστημα, η επακόλουθη ευκολία στη χρήση, η μεγάλη επεξεργαστική ισχύς, ο υψηλός βαθμός επεκτασιμότητας του συστήματος και κυρίως ο μικρός στατιστικά συνολικός χρόνος ανάπτυξης εφαρμογών. Στα μειονεκτήματα συμπεριλαμβάνονται το μεγάλο συγκριτικά κόστος, οι μεγάλες απαιτήσεις σε ενέργεια και μερικές φορές το μεγάλο μέγεθος.



Σχήμα 3.2 Παραδείγματα Single Board Computers (SBCs).

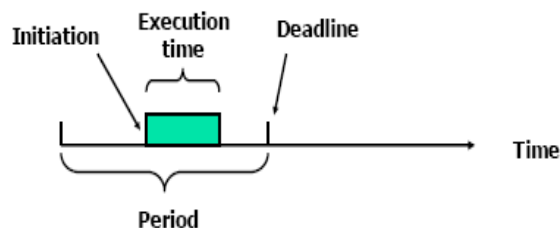
Μετά από ενδελεχή ανάλυση και σύγκριση των δύο τρόπων ανάπτυξης ενσωματωμένων συστημάτων, τελικά προτιμήθηκε η χρήση πλακέτας ολοκληρωμένου υπολογιστή SBC, ως πιο ευέλικτη, ασφαλής και γρήγορη στην ανάπτυξη λύση. Σημαντικό ρόλο στην επιλογή αυτή παίζει η δυνατότητα χρήσης λειτουργικού συστήματος και επομένως, η δυνατότητα ανάπτυξης λογισμικού από τη μέρα κιάλας της αγοράς της πλακέτας. Επίσης, δεν είναι αμελητέο ότι κυκλοφορούν σήμερα πάρα πολλά προγράμματα ανάπτυξης ενσωματωμένων συστημάτων τα οποία μπορούν να αξιοποιηθούν χρησιμοποιώντας μια τέτοια πλατφόρμα. Κρίνεται λοιπόν ότι αν η επιλογή της πλακέτας SBC γίνει σωστά, οι δυνατότητες ανάπτυξης είναι απεριόριστες αφού εκτός των άλλων δίνεται η δυνατότητα χρήσης πακέτων λογισμικού αιχμής, όπως θα φανεί και παρακάτω.

### 3.3 Λογισμικό - Συστήματα Πραγματικού Χρόνου

Σχετικά με το λογισμικό του ρομπότ, κύριος στόχος είναι η δυνατότητα προγραμματισμού του σε πραγματικό χρόνο. Γενικά, τα ρομποτικά συστήματα συνηθίζεται να είναι συστήματα πραγματικού χρόνου. Για τον λόγο αυτό, τα κύρια χαρακτηριστικά των συστημάτων αυτών περιγράφονται παρακάτω.

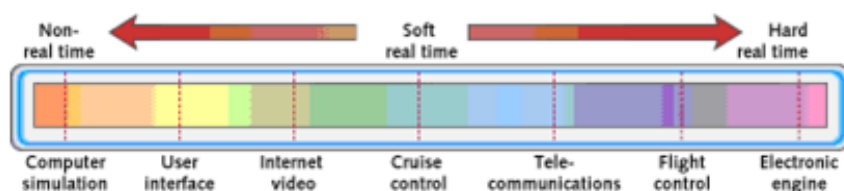
Στην πράξη, τα περισσότερα υπολογιστικά συστήματα που χειρίζονται εφαρμογές όπου η ασφάλεια παίζει αποφασιστική σημασία αλλά και γενικότερα τα ενσωματωμένα υπολογιστικά συστήματα (embedded computer systems), είναι συστήματα πραγματικού χρόνου. Ένα σύστημα πραγματικού χρόνου είναι ένα οποιοδήποτε σύστημα επεξεργασίας πληροφορίας το οποίο πρέπει να αποκριθεί σε εξωτερικά παραγόμενη είσοδο μέσα σε πεπερασμένο και προσδιορισμένο χρονικό διάστημα (βλ. Σχήμα 3.3). Είναι δηλαδή ένα σύστημα, ικανό να εκτελεί με αξιοπιστία προγράμματα με συγκεκριμένους χρονικούς περιορισμούς, κάτι που είναι σημαντικό σε εφαρμογές επιστημόνων και μηχανικών.

Μια συνηθισμένη λανθασμένη αντίληψη είναι πως τα συστήματα πραγματικού χρόνου έχουν ως κύριο χαρακτηριστικό τους γρήγορους υπολογισμούς. Κάτι τέτοιο δεν ισχύει. Το σημαντικό σε ένα τέτοιο σύστημα είναι να λειτουργεί με τέτοια ταχύτητα που να συμβαδίζει με τις χρονικές απαιτήσεις που θέτει ο εξωτερικός κόσμος. Φυσικά, η ταχύτητα εκτέλεσης αποτελεί κι αυτή σημαντικό παράγοντα.



**Σχήμα 3.3** Σχηματική αναπαράσταση προγραμματισμού σε πραγματικό χρόνο.

Τα συστήματα πραγματικού χρόνου χρησιμοποιούνται, κυρίως, σε εφαρμογές όπου απαιτούνται η επιτήρηση, ο έλεγχος και η απόκριση προς το εξωτερικό περιβάλλον, με αυξημένη αξιοπιστία και ασφάλεια. Εφόσον λοιπόν ελέγχουν το περιβάλλον στο οποίο λειτουργούν, πιθανή αστοχία στον έλεγχο μπορεί να οδηγήσει σε απώλειες ζωών, βλάβη στο περιβάλλον ή οικονομική ζημιά. Γενικότερα, το φάσμα των εφαρμογών πραγματικού χρόνου είναι ευρύ (βλ. Σχήμα 3.4). Μερικά παραδείγματα εφαρμογών είναι: τα αεροπορικά ηλεκτρονικά συστήματα (avionics), ο έλεγχος ενάριας κυκλοφορίας, τα πολυμέσα, οι τηλεπικοινωνίες και τα δίκτυα δεδομένων, η ιατρική (τηλεϊατρική, έλεγχος μονάδων εντατικής θεραπείας), και τα στρατιωτικά συστήματα.



**Σχήμα 3.4** Non-real time, soft real time και hard real time εφαρμογές.

Η επικοινωνία με το περιβάλλον γίνεται μέσω αισθητήρων, επενεργητών, και άλλων μονάδων εισόδου/εξόδου (I/O). Το υλικό που χρησιμοποιείται αποτελείται από επεξεργαστές, μονάδες μνήμης, μετατροπείς A/D και D/A, διαύλους επικοινωνίας και λοιπές περιφερειακές συσκευές. Πιο συγκεκριμένα, τα μέρη ενός συστήματος πραγματικού χρόνου είναι:

- Υλικό μέρος (CPUs, χρονιστές, περιφερειακές μονάδες I/O, επενεργητές, κανάλια επικοινωνίας, κλπ). Τρόποι υλοποίησης: ενσωματωμένοι επεξεργαστές ή μικροελεγκτές, συστήματα DSP ή συστήματα υλικού όπως FPGAs και ASICs.
- Ένα λειτουργικό σύστημα πραγματικού χρόνου (RTOS) ή απλά ένας πυρήνας πραγματικού χρόνου (Real-Time Kernel).
- Ένα σύνολο από 'διεργασίες' (tasks) πραγματικού χρόνου.

### 3.3.1 Τρόποι υλοποίησης συστημάτων πραγματικού χρόνου

**Πρώτη προσέγγιση:** προγραμματισμός του συστήματος σε χαμηλό επίπεδο, χρησιμοποιώντας γλώσσα Assembly και προγραμματίζοντας το σύστημα με χρήση χρονιστών, διακοπών (interrupts), επαναλαμβανόμενων βρόχων κλπ. Οι ρουτίνες σε assembly πρέπει να διαχειρίζονται τις θύρες I/O, τους μετατροπείς αναλογικού σήματος σε ψηφιακό και τα κανάλια επικοινωνίας, όπως οι UART, οι θύρες USB, οι δίαυλοι CAN (Controller Area Network) κ.λπ.

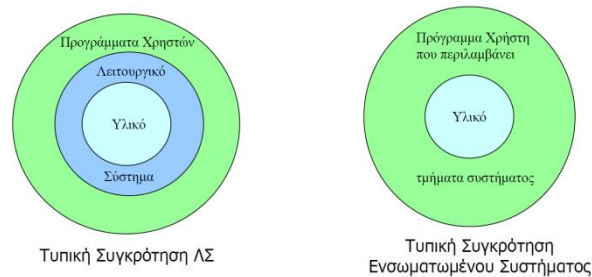
Μια **άλλη προσέγγιση**, πιο κατάλληλη για μεγάλα και πολύπλοκα συστήματα, είναι ο προγραμματισμός με χρήση παράλληλων (concurrent) γλωσσών πραγματικού χρόνου. Τέτοιες γλώσσες είναι η Ada 95, η Real-time Java, η RTL/2 και άλλες. Οι γλώσσες αυτές έχουν το χαρακτηριστικό ότι μπορούν να διαχειρίζονται παραπάνω από μία διεργασία (task), καθώς ένα πρόγραμμα μπορεί να αποτελείται από ένα σύνολο αυτόνομων σειριακών διεργασιών, που η λογική τους εκτέλεση γίνεται παράλληλα.

Μια **τρίτη προσέγγιση** είναι η χρήση μιας συμβατικής σειριακής γλώσσας προγραμματισμού (π.χ. PASCAL, C) και ενός λειτουργικού συστήματος που παρέχει υποστήριξη για διαδικασίες πραγματικού χρόνου. Τέτοια λειτουργικά συστήματα αναφέρονται ως λειτουργικά συστήματα πραγματικού χρόνου (RTOS) και παρέχουν ευκολίες για πολυδιεργασία (multitasking) και πολυνημάτωση (multithreading). Στην περίπτωση αυτή οι λειτουργίες του παραλληλισμού περνούν στο λειτουργικό σύστημα και όχι στην γλώσσα προγραμματισμού

### 3.3.2 Χαρακτηριστικά λειτουργικού συστήματος πραγματικού χρόνου (RTOS)

Τα συστήματα αυτομάτου ελέγχου ανήκουν, κατά κανόνα, στην κατηγορία εκείνη όπου η έγκαιρη απόκριση στις μεταβολές του περιβάλλοντος είναι καθοριστικής σημασίας. Ο έλεγχος αυτός απαιτεί μια πλατφόρμα που να εξασφαλίζει κάποιους καθορισμένους χρονικούς περιορισμούς. Τα περισσότερα λειτουργικά συστήματα (όπως Windows 95, 98, NT και XP) χρησιμοποιούν ορισμένους μηχανισμούς οι οποίοι τα καθιστούν ακατάλληλα για επεξεργασία πραγματικού χρόνου. Αυτά είναι συστήματα γενικού σκοπού και γι' αυτό είναι σχεδιασμένα να τρέχουν πολλές εφαρμογές ταυτόχρονα και να παρέχουν πλούσιο γραφικό περιβάλλον. Αντίθετα, τα συστήματα πραγματικού χρόνου σχεδιάζονται έτσι, ώστε να τρέχουν ένα μόνο πρόγραμμα με αυστηρές χρονικές απαιτήσεις. Για το λόγο αυτό χρησιμοποιείται ειδικό λογισμικό, το οποίο χαρακτηρίζεται ως λειτουργικό σύστημα πραγματικού χρόνου (RTOS) και διαφέρει σημαντικά από τα συμβατικά προγράμματα (βλ. Σχήμα 3.5).

Εξυπακούεται βέβαια ότι ένα λειτουργικό σύστημα πραγματικού χρόνου υποστηρίζει όλες τις βασικές λειτουργίες ενός συμβατικού λειτουργικού συστήματος, όπως η διαχείριση διακοπών (interrupt handling), η διαχείριση διεργασιών (task manager), η διαχείριση της μνήμης (εκχώρηση, απελευθέρωση και προστασία της μνήμης), η διαχείριση εισόδων/εξόδων μέσω οδηγών συσκευών (device drivers) και η επικοινωνία ανάμεσα σε διεργασίες.



Σχήμα 3.5 Συγκρότηση λογισμικού ενός ενσωματωμένου συστήματος.

### 3.3.3 Πυρήνας RTOS

Το κέντρο ενός λειτουργικού συστήματος πραγματικού χρόνου ονομάζεται πυρήνας (kernel). Στα λειτουργικά συστήματα πραγματικού χρόνου ο πυρήνας αποτελείται από τον χρονοπρογραμματιστή διεργασιών (scheduler), τον διαχειριστή διακοπών (interrupt handler), και τον διαχειριστή πόρων (resource manager), που διαχειρίζεται τη μνήμη και τον διαμοιρασμό της στις διάφορες διεργασίες.

### 3.3.4 Διεργασίες (tasks)

Οι διεργασίες είναι ρουτίνες λογισμικού, γραμμένες σε κάποια γλώσσα πραγματικού χρόνου ή είναι μηχανισμοί που δημιουργεί το λειτουργικό σύστημα, για να διαχειρίζεται την εκτέλεση τμημάτων κώδικα σε ένα πρόγραμμα. Οι διεργασίες μοιράζονται τους πόρους του συστήματος, επικοινωνούν και συγχρονίζονται μεταξύ τους και με το περιβάλλον.

Ένα λειτουργικό σύστημα πραγματικού χρόνου διαχειρίζεται τα διάφορα προγράμματα χωρίζοντάς τα σε νήματα διεργασιών, τις οποίες και ελέγχει κατάλληλα. Μια διεργασία είναι μια βασική μονάδα προγραμματιστικού έργου, μια οντότητα λογισμικού, που δημιουργείται από το λειτουργικό σύστημα προκειμένου να διαχειρίζεται την εκτέλεση ενός τμήματος κώδικα. Κάθε διεργασία είναι ανεξάρτητη από τις άλλες διεργασίες και μπορεί να συγχρονιστεί και να επικοινωνήσει μαζί τους.

Αφού δημιουργηθεί μια διεργασία, έχει μια διάρκεια ζωής, κατά την οποία μπορεί να βρεθεί σε διάφορες καταστάσεις, ανάλογα με το ποιες καταστάσεις διεργασιών υποστηρίζει το λειτουργικό σύστημα. Μπορεί, λοιπόν, να δημιουργείται, να βρίσκεται σε ανενεργή κατάσταση (sleeping) ή σε κατάσταση ετοιμότητας (ready), να εκτελείται και να τερματίζεται. Οι διεργασίες διαχωρίζονται στις παρακάτω κατηγορίες:

**Περιοδικές:** Πρόκειται για εργασίες οι οποίες συμβαίνουν σε τακτά χρονικά διαστήματα. Χαρακτηριστικά τους όπως: περίοδος, χρόνος υπολογισμού χειρότερης περίπτωσης (worst case execution time) είναι γνωστά εκ των προτέρων.

**Απεριοδικές:** Ενεργοποιούνται όταν συμβούν συγκεκριμένα γεγονότα. Τα χαρακτηριστικά τους (άφιξη, χρόνος υπολογισμού χειρότερης περίπτωσης, προθεσμία) γίνονται γνωστά μόνο με την άφιξή τους.

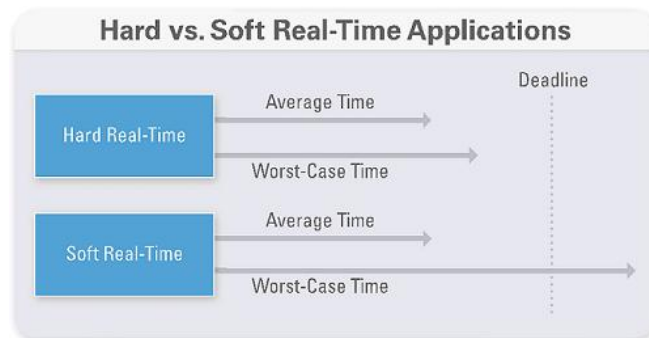
**Σποραδικές:** Για αυτές υπάρχει ένας ελάχιστος εγγυημένος χρόνος μεταξύ των αφίξεων. Είναι εφικτή η τήρηση αυστηρών προθεσμιών.

### 3.3.5 Hard / Soft Real time

Ένας βασικός διαχωρισμός στα υπολογιστικά συστήματα πραγματικού χρόνου είναι αυτός ανάμεσα σε 'hard real time' και 'soft real time' συστήματα (βλ. Σχήμα 3.6).

**Hard real-time:** είναι συστήματα στα οποία είναι απολύτως επιτακτικό, οι απαντήσεις να δίνονται μέσα στην απαιτούμενη προθεσμία π.χ. συστήματα ελέγχου πτήσεων.

**Soft real-time:** συστήματα όπου ενώ οι προθεσμίες είναι σημαντικές, θα συνεχίσουν να λειτουργούν ορθά αν περιστασιακά οι προθεσμίες δεν τηρηθούν π.χ. συστήματα πολυμέσων, συστήματα δρομολόγησης/μεταγωγής σε δίκτυα, κάποια συστήματα ελέγχου, παιχνίδια.



Σχήμα 3.6 Σχηματική αναπαράσταση σύγκρισης συστημάτων ‘hard’ και ‘soft’ real time.

### 3.4 Σχεδιασμός βασισμένος στο μοντέλο (Model-based design)

Γίνεται σιγά σιγά φανερό πως ο συμβατικός τρόπος ανάπτυξης παρουσιάζει δυσκολίες. Απαιτεί ειδικές γνώσεις προγραμματισμού πραγματικού χρόνου και όχι μόνο, γεγονός που κάνει δύσκολη την ανάπτυξη ενσωματωμένων συστημάτων από μικρό αριθμό μηχανικών και σε μικρό χρόνο. Μια διαφορετική λύση σε αυτά τα προβλήματα έρχεται να δώσει, ένας σχετικά νέος τρόπος ανάπτυξης συστημάτων, ο οποίος χρησιμοποιείται πλέον όλο και περισσότερο σε ευφυή συστήματα, γνωστός ως, σχεδιασμός βασισμένος στο μοντέλο (model-based design). [40]

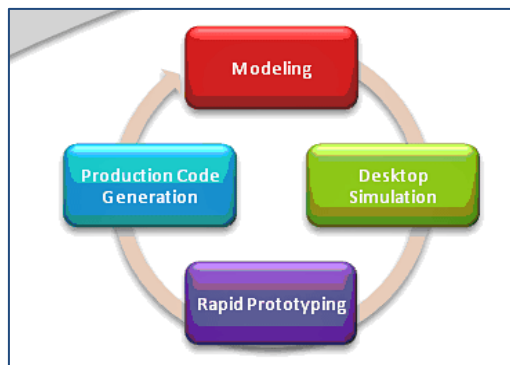
Αρχικά επισημαίνεται ότι, η παραδοσιακή διαδικασία που ακολουθείται για τη σχεδίαση και υλοποίηση συστημάτων πραγματικού χρόνου περιλαμβάνει τα ακόλουθα στάδια:

- Ανάπτυξη αλγορίθμου
- Ανάπτυξη λογισμικού
- Κατασκευή/Συναρμολόγηση του υλικού
- Προσομοίωση του συστήματος σε πραγματικό χρόνο με βάση τα παραπάνω

Τα στάδια αυτά επαναλαμβάνονται συνεχώς κατά την ανάπτυξη ενός συστήματος μέχρι να επιτευχθούν τα επιθυμητά αποτελέσματα. Αυτή η προσέγγιση απαιτεί πολύ χρόνο, επειδή η σχεδίαση του λογισμικού γίνεται ξεχωριστά από τη σχεδίαση των αλγορίθμων και τη δοκιμή στο τελικό υλικό (hardware). Επιπρόσθετα, τα παραπάνω στάδια απασχολούν ξεχωριστές ομάδες μηχανικών και επιστημόνων, γεγονός που απαιτεί τέλεια οργάνωση και αρκετό χρόνο για επικοινωνία. Η υλοποίηση δε σε επίπεδο συνολικού hardware γίνεται μόνο σε τελικό στάδιο και το ίδιο επομένως ισχύει και για τα πειράματα (δεδομένου ότι το σύστημα είναι συνήθως ογκώδες και πολύπλοκο π.χ. ανεμογεννήτρια). Τα λάθη που θα διαπιστωθούν στην πειραματική διαδικασία μπορεί να είναι θεμελιώδη και να επηρεάσουν σε μεγάλο βαθμό το γενικό σχέδιο φέρνοντας μεγάλη οικονομική ζημία και χρονική σπατάλη.

Ο σχεδιασμός βασισμένος στο μοντέλο (model-based design) είναι ένας σύγχρονος τρόπος ανάπτυξης ενσωματωμένων συστημάτων ο οποίος αντιμετωπίζει το σύστημα σαν σύνολο. Με αυτό το σκοπό, χρησιμοποιείται τεχνολογία αιχμής (λογισμικό) ώστε να παρέχεται μια πλατφόρμα ανάπτυξης, όπου οι αρχικές ιδέες υλοποιούνται, δοκιμάζονται και επαληθεύονται άμεσα χρησιμοποιώντας το πραγματικό υλικό (hardware). Η άμεση και εύκολη υλοποίηση μέσω αυτοματοποιημένων διαδικασιών παραγωγής κώδικα, προσδίδει στη μέθοδο το χαρακτηριστικό της δοκιμής και αξιολόγησης σχεδιαστικών ιδεών καθ’ όλη τη διαδικασία της ανάπτυξης. Όπως ακριβώς αναπαρίσταται στο Σχήμα 3.7, πρόκειται για έναν αέναο βρόχο με κόμβους, τη μοντελοποίηση σύμφωνα με τις αρχικές σχεδιαστικές απαιτήσεις και ιδέες, την προσομοίωση, την άμεση παραγωγή κώδικα πραγματικού χρόνου και την εκτέλεση χρησιμοποιώντας το τελικό υλικό (hardware). Αυτό σημαίνει ότι η σχεδίαση των αλγορίθμων συνδέεται άμεσα με την ανάπτυξη του λογισμικού και το

διαθέσιμο υλικό. Κύρια χαρακτηριστικά της μεθόδου είναι: η συνεχής υλοποίηση και αξιολόγηση ιδεών, η πειραματική αξιολόγηση σε πραγματικές συνθήκες καθ' όλη τη διάρκεια της ανάπτυξης και όχι μόνο στο τέλος, η έγκαιρη διάγνωση λαθών και ο μικρός χρόνος ανάπτυξης.



**Σχήμα 3.7** Μέθοδος σχεδιασμού βασισμένου στο μοντέλο.

Αναγνωρίζοντας τελικά τα ισχυρά πλεονεκτήματα της μεθόδου έναντι του παραδοσιακού τρόπου ανάπτυξης συστημάτων, επιλέχθηκε ο σχεδιασμός βασισμένος στο μοντέλο, σαν μέθοδος σχεδιασμού και ανάπτυξης του ρομπότ του διαστημικού εξομοιωτή. Όσον αφορά τα προγράμματα που χρησιμοποιούνται, κυριαρχούν αυτά των εταιριών MathWorks, dSPACE και National Instruments. Δεδομένου ότι το πακέτο που χρησιμοποιείται κατά κόρον στα σύγχρονα συστήματα αυτομάτου ελέγχου είναι το Simulink της Mathworks, επιλέχθηκε τελικά το πακέτο της Mathworks για model-based design, γνωστό ως xPC Target.

### 3.5 xPC Target

Το xPC Target [41] λοιπόν είναι η πρόταση της Mathworks σχετικά με τον σχεδιασμό βασισμένο στο μοντέλο. Το πακέτο αυτό παρέχει έναν πυρήνα αυστηρά πραγματικού χρόνου (hard real time) στον οποίο εκτελούνται (σε πραγματικό hardware) εφαρμογές που έχουν παραχθεί από απλά μοντέλα προσομοίωσης Simulink. Για να κατανοηθεί πλήρως η λειτουργία του, είναι πρώτα απαραίτητο να αναφερθούν και στην συνέχεια να περιγραφούν οι κύριες τεχνικές που ενσωματώνονται σε αυτό το χρήσιμο πακέτο λογισμικού.

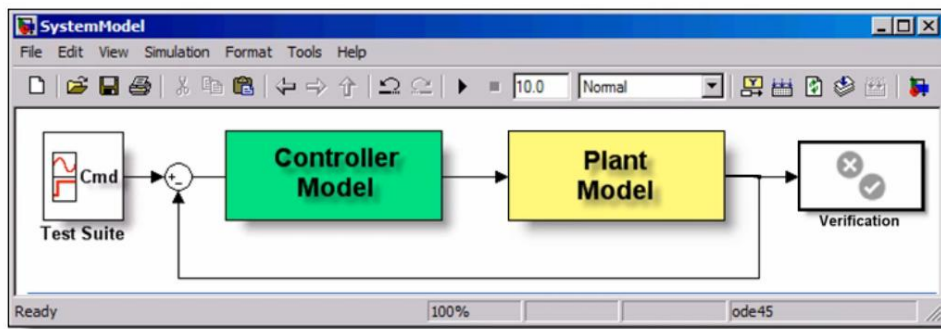
Κύριες τεχνικές που εμπεριέχονται στον σχεδιασμό βασισμένο στο μοντέλο με το xPC Target:

- Προγραμματισμός Block-Diagram-Based και αυτόματη παραγωγή κώδικα
- Functional Rapid Prototyping
- Hardware In the Loop (HIL)

#### 3.5.1 Προγραμματισμός με λειτουργικά διαγράμματα (Block-Diagram-Based)

Κυρίαρχο ρόλο στην μοντελοποίηση των συστημάτων ελέγχου έχουν τα λειτουργικά διαγράμματα (block diagrams). Από τότε που παρουσιάστηκαν από τον T. M. Stout αποτελούν ένα από τα δημοφιλέστερα εργαλεία στο χώρο του αυτομάτου ελέγχου. Τα block-διαγράμματα αποτελούν τον πιο δημοφιλή τρόπο για τον προσδιορισμό και την μοντελοποίηση ενός πολύπλοκου μαθητικού αλγορίθμου. Χάρη στην τεχνολογία αυτόματης παραγωγής κώδικα, ένα μοντέλο μπορεί εύκολα να μετατραπεί σε πρόγραμμα, απαλλάσσοντας το σχεδιαστή από το επίπονο έργο του προγραμματισμού. Στα πλαίσια του xPC Target η μοντελοποίηση γίνεται με τα πακέτα Simulink και Stateflow.



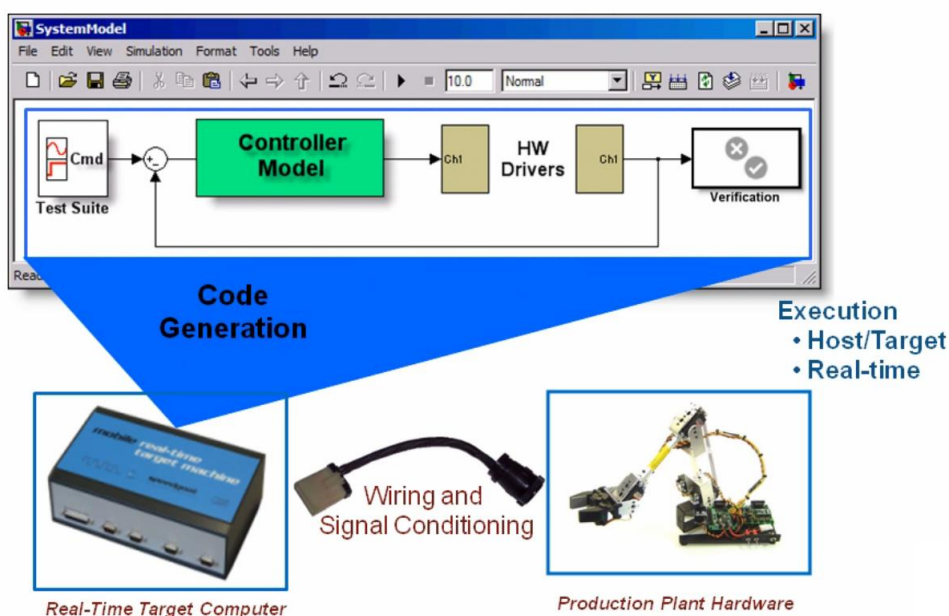


Σχήμα 3.8 Αρχική μοντελοποίηση του συστήματος σε επίπεδο προσομοίωσης.

Ο κύριος στόχος του σχεδιασμού βασισμένου στο μοντέλο είναι το πέρασμα από την προσομοίωση στη real time δοκιμή. Όπως ήδη αναφέρθηκε, τα εργαλεία μοντελοποίησης της Mathworks είναι τα πακέτα λογισμικού Simulink και Stateflow. Επομένως το ερώτημα που τίθεται είναι, πώς θα γίνει η μετάβαση από το επίπεδο προσομοίωσης των Simulink και Stateflow, στο επίπεδο εκτέλεσης των μοντέλων σαν εφαρμογές πραγματικού χρόνου και σε πραγματικό υλικό (hardware). Σε αυτή τη βάση, η Mathworks έχει να παρουσιάσει δύο μεθόδους για δοκιμή μοντέλων σε πραγματικό χρόνο.

### 3.5.2 Functional Rapid Prototyping

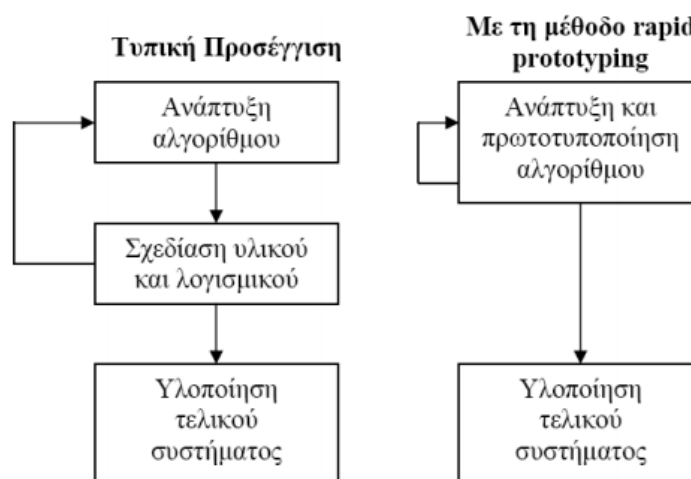
Στην κλασική προσομοίωση ενός μοντέλου στο Simulink (π.χ. μοντέλο Σχήματος 3.8), ο κώδικας που παράγεται αυτόματα για αυτό, τρέχει σε μη-πραγματικό χρόνο στον υπολογιστή όπου και σχεδιάστηκε. Η διαφορά εδώ είναι ότι, ειδικά για το μοντέλο του ελεγκτή παράγεται κώδικας πραγματικού χρόνου και μόνο αυτός εκτελείται σε έναν γρήγορο εξωτερικό υπολογιστή, ικανό συνήθως να εκτελεί πράξεις κινητής υποδιαστολής. Οι λειτουργίες εισόδου/ εξόδου γίνονται, συνήθως, με τη χρήση μιας κάρτας DAQ (data acquisition), με την οποία είναι εφοδιασμένος ο υπολογιστής, ενώ το σύστημα των αισθητήρων / επενεργητών αντικαθίσταται από τα πραγματικά, όπως φαίνεται στο Σχήμα 3.9. Οι παράμετροι του ελεγκτή μπορεί να μεταβάλλονται κατά τη λειτουργία (on-the-fly).



Σχήμα 3.9 Παραγωγή κώδικα πραγματικού χρόνου από το μοντέλο φυσικού συστήματος (HIL).

Όπως αναφέρθηκε και προηγουμένως, η κατασκευή του μοντέλου για το φυσικό μας σύστημα, αλλά και για τον ίδιο τον ελεγκτή, αποτελεί το πρώτο βήμα της διαδικασίας σχεδίασης. Στη συνέχεια, χρειάζεται, επιπλέον, να γίνει η υλοποίηση του συστήματος, δηλαδή η σύνδεση του φυσικού συστήματος με τον ελεγκτή. Δεν είναι λίγες οι φορές που το μεγαλύτερο μέρος της προσπάθειας ανάπτυξης ενός συστήματος ελέγχου αφιερώνεται στο κομμάτι αυτό. Ορισμένες από τις λεπτομέρειες της υλοποίησης, όπως η επιλογή του μικροεπεξεργαστή που θα χρησιμοποιηθεί (για παράδειγμα, αν χρησιμοποιεί αριθμητική σταθερής υποδιαστολής και με τί ακρίβεια), μπορεί να έχουν σημαντικές επιπτώσεις στο αρχικό μοντέλο του ελεγκτή, με αποτέλεσμα να πρέπει αυτό να τροποποιηθεί.

Η παραδοσιακή προσέγγιση στη σχεδίαση συστημάτων πραγματικού χρόνου περιλαμβάνει την συνεργασία διαφόρων ομάδων από προγραμματιστές και μηχανικούς. Για να προχωρήσει η διαδικασία χρειάζεται κάθε ομάδα να ολοκληρώσει το έργο της, ώστε η επόμενη να συνεχίσει με βάση αυτό. Κάτι τέτοιο οδηγεί σε μεγάλους χρόνους ανάπτυξης, ιδίως όταν παρουσιάζονται λάθη σε κάποιο στάδιο. Ο κύριος λόγος εμφάνισης λαθών είναι ότι οι σχεδιαστές του αλγορίθμου δεν δουλεύουν απευθείας πάνω στο υλικό. Με τη μέθοδο *rapid prototyping* συνδέονται οι φάσεις σχεδίασης του αλγορίθμου και υλοποίησης του λογισμικού και υλικού, αποτρέποντας έτσι τις καθυστερήσεις κατά την ανάπτυξη (βλ. Σχήμα 3.10). Οι μηχανικοί μπορούν, έτσι, άμεσα να δουν τα αποτελέσματα της σχεδίασης και να προβούν σε διορθώσεις προτού κατασκευαστεί το (ακριβό) τελικό σύστημα.



**Σχήμα 3.10** Τυπική προσέγγιση και μέθοδος *rapid prototyping*.

Οι τυπικές μέθοδοι σχεδίασης συστημάτων ελέγχου βασίζονται σε ένα μοντέλο του φυσικού συστήματος. Το μαθηματικό αυτό μοντέλο είναι συνήθως εξιδανικευμένο και δεν λαμβάνει υπόψη κάποιες άγνωστες παραμέτρους του συστήματος. Θα πρέπει να διεξαχθούν πειράματα προκειμένου να συγκεντρωθούν επιπλέον πληροφορίες για τη δυναμική συμπεριφορά του συστήματος, ώστε να γίνει κάποια εκτίμηση των άγνωστων παραμέτρων του. Η τεχνική *Rapid Prototyping* ενισχύει σημαντικά την παραπάνω μεθοδολογία. Αρχικά, το μοντέλο μπορεί να είναι ελλιπές ή να μην είναι διαθέσιμο. Σε πρώτη φάση, σχεδιάζεται ένα σύστημα ελέγχου, το οποίο θα σταθεροποιεί το σύστημα και θα επιτυγχάνει όσο γίνεται καλύτερη συμπεριφορά. Στη συνέχεια, μέσα από νέα πειράματα συλλέγονται επιπλέον δεδομένα για την απόκριση του συστήματος, τα οποία χρησιμοποιούνται για τη βελτίωση του αρχικού μοντέλου. Ταυτόχρονα, γίνονται αλλαγές και στο σύστημα ελέγχου με βάση το αναβαθμισμένο μοντέλο του φυσικού συστήματος. Με τον τρόπο αυτό, προσεγγίζεται όλο και περισσότερο η τελική μορφή του συστήματος ελέγχου.

Σκοπός της τεχνικής *rapid prototyping* είναι να επισημανθούν οι ατέλειες και τα λάθη χρησιμοποιώντας μια πειραματική διάταξη, προτού ξεκινήσει η παραγωγή του τελικού σχεδίου. Με άλλα λόγια, αποτελεί ένα γρήγορο και ασφαλή τρόπο επικύρωσης (*validation*) του αλγορίθμου ελέγχου με τη χρήση πραγματικού εξοπλισμού (αισθητήρες, επενεργητές, μετατροπείς A/D, D/A,



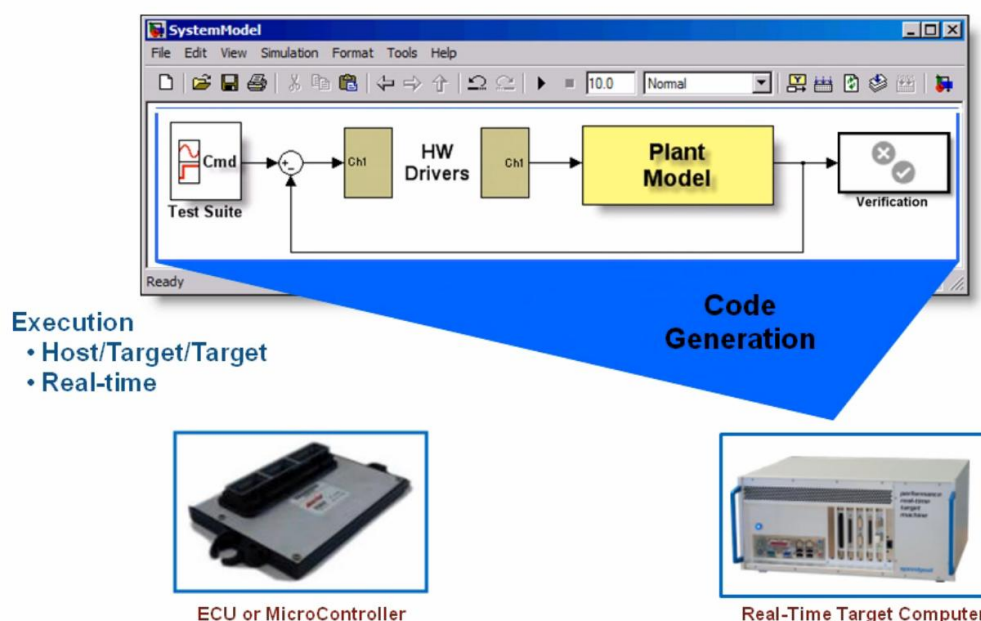
κλπ.). Αυτό αποτελεί μια πολύ συνηθισμένη και δημοφιλής πρακτική σχεδίασης. Παραδείγματος χάριν, στο χώρο του λογισμικού δοκιμάζεται και εξετάζεται πρώτα ένας δοκιμαστικός αλγόριθμος προτού προστεθούν εκτενή σχόλια και περαιτέρω λειτουργίες. Στην επιστημονική έρευνα και εκπαίδευση, όπου τα διάφορα συστήματα πειραματισμού σπάνια καταλήγουν στην παραγωγή, η τεχνική rapid prototyping βρίσκει μεγάλη ανταπόκριση.

Συνοπτικά, μερικά πλεονεκτήματα της τεχνικής είναι τα εξής: το σύστημα λειτουργεί με βάση την πραγματική δυναμική του συστήματος, πράγμα που δεν μπορεί να επιτευχθεί με την απλή προσομοίωση, ο αλγόριθμος ελέγχου δοκιμάζεται δυναμικά, γίνεται επαλήθευση του αλγορίθμου σε πραγματικές συνθήκες πριν το στάδιο του προγραμματισμού, γίνεται διάγνωση λαθών μέσω επανάληψης των πειραμάτων, και βελτιστοποιείται το αρχικό σχέδιο λόγω εύκολης και άμεσης ρύθμισης παραμέτρων.

### 3.5.3 Hardware in the Loop (HIL)

Προκειμένου να γίνει η προσομοίωση πιο ρεαλιστική, μια λύση είναι να αντικατασταθούν ορισμένα μόνο μέρη του συστήματος με πραγματικό υλικό, ενώ κάποια άλλα να προσομοιώνονται από λογισμικό. Αυτή η προσέγγιση ονομάζεται Hardware-in-the-loop (HIL) [43] και αποτελεί επέκταση της μεθόδου Rapid Prototyping. Στην προσομοίωση HIL το σύστημα διασπάται σε δύο ξεχωριστά διαγράμματα, ένα για το φυσικό σύστημα και ένα για τον ελεγκτή, τα οποία αλληλεπιδρούν μέσω ειδικών block επικοινωνίας (I/O blocks). Επειδή υπάρχει άμεση αλληλεπίδραση του λογισμικού με το υλικό, η έννοια του 'πραγματικού χρόνου' γίνεται πρωταρχικής σημασίας. Σε αυτή την περίπτωση, αντί να τρέχει ο αλγόριθμος του ελεγκτή σε πραγματικό χρόνο, το δυναμικό μοντέλο του συστήματος (plant model) μεταφράζεται σε κώδικα και τρέχει αυτό σε πραγματικό χρόνο (βλ. Σχήμα 3.11).

Η προσομοίωση HIL περιλαμβάνει την αλληλεπίδραση μεταξύ ενός μοντέλου προσομοίωσης για το φυσικό σύστημα και ενός επεξεργαστή, ο οποίος υλοποιεί σε hardware το κομμάτι του ελεγκτή. Η μέθοδος αναγνωρίζεται όλο και περισσότερο ως ένα πολύτιμο εργαλείο κατά τη διαδικασία ανάπτυξης συστημάτων. Αντί να χρησιμοποιείται κάποια ακριβή και δύσχρηστη πειραματική διάταξη (π.χ. αεροσκάφος, δορυφόρος κλπ), το φυσικό σύστημα μοντελοποιείται με τη βοήθεια ενός υπολογιστή, ο οποίος θα παρέχει σε πραγματικό χρόνο όλα τα απαραίτητα σήματα, όπως ακριβώς θα γινόταν στο φυσικό κόσμο. Η αντικατάσταση μέρους του μοντέλου με κατάλληλο hardware δίνει τη δυνατότητα πρόσβασης σε στοιχεία του υλικού που θα ήταν αδύνατο να χειριστεί κάποιος μέσα από την προσομοίωση.

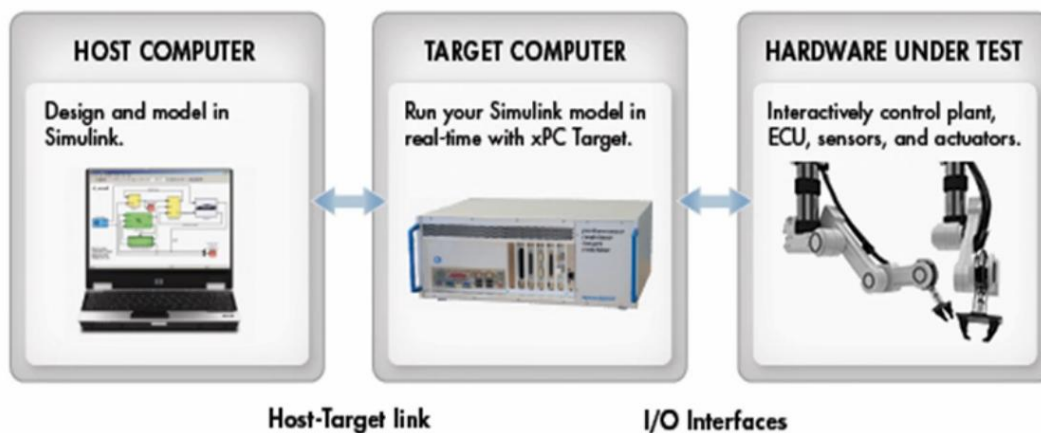


Σχήμα 3.11 Παραγωγή κώδικα πραγματικού χρόνου από το μοντέλο φυσικού συστήματος (HIL).

### 3.5.4 xPC Target

Αυτό που κάνει το λογισμικό xPC Target είναι να συνδυάζει τις τεχνικές που αναφέρθηκαν παραπάνω σε μια εύχρηστη πλατφόρμα, ιδανική για σχεδιασμό και ανάπτυξη συστημάτων αυτομάτου ελέγχου. Συγκεκριμένα, παρέχει ένα περιβάλλον αυστηρά πραγματικού χρόνου (Hard real time environment) για δοκιμή μοντέλων συστημάτων, για το οποίο απαιτείται ένας υπολογιστής (host PC) που τρέχει Matlab σε λειτουργικό Windows, και ένας δεύτερος υπολογιστής (target PC) που τρέχει τον πυρήνα πραγματικού χρόνου του xPC Target (βλ. Σχήμα 3.12).

Η διαδικασία είναι η εξής: το μοντέλο simulink μετατρέπεται σε κώδικα C πραγματικού χρόνου μέσω του Real Time Workshop (πακέτο της Mathworks), ο κώδικας C μετατρέπεται σε εφαρμογή πραγματικού χρόνου (.dlm) μέσω ενός C compiler, η εφαρμογή πραγματικού χρόνου αποστέλλεται στον δεύτερο υπολογιστή (target) μέσω σειριακής (RS-232) ή Ethernet επικοινωνίας, και τέλος αυτή εκτελείται στον πυρήνα πραγματικού χρόνου του xPC Target.



Σχήμα 3.12 Διαδικασία ανάπτυξης μέσω xPC Target.

Κατά την εκτέλεση είναι δυνατή η επικοινωνία μεταξύ της εφαρμογής πραγματικού χρόνου και του host υπολογιστή. Μέσω αυτής επιτρέπεται η ρύθμιση παραμέτρων κατά την εκτέλεση (on the fly), η απόκτηση δεδομένων από τις δοκιμές και η επεξεργασία των δεδομένων αυτών μέσω Matlab. Τέλος, υποστηρίζεται η σύνδεση με φιλικά στο χρήστη γραφικά περιβάλλοντα που μπορούν να ενσωματώνουν ακόμη και απεικόνιση παραμέτρων σε τρισδιάστατο εικονικό κόσμο (Virtual Reality).

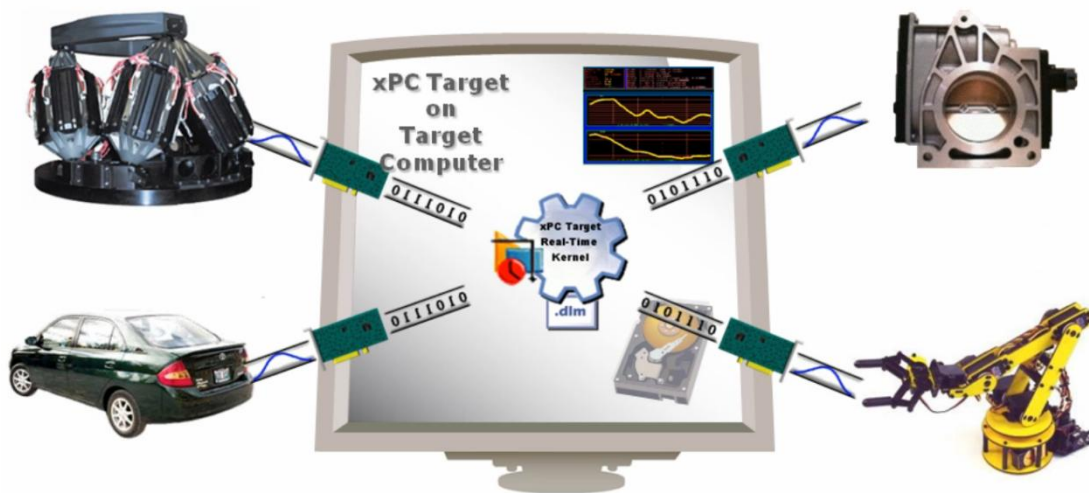
#### Πλεονεκτήματα Πακέτου:

- Εστίαση στους αλγορίθμους και όχι στην υλοποίηση σε χαμηλό επίπεδο
- Προγραμματισμός hard-real time
- Σημαντική μείωση του χρόνου ανάπτυξης
- Το σύστημα γίνεται μια πλατφόρμα απεριόριστων δοκιμών
- Δυνατότητα συνεχούς δοκιμής καινούριων ιδεών και εστίαση στην καινοτομία
- Επαλήθευση και βελτίωση του συστήματος από πολύ νωρίς στην διαδικασία ανάπτυξης
- Εξέλιξη του αρχικού σχεδιασμού κατά την ανάπτυξη
- Αποφυγή λαθών που θα κόστιζαν, αφού εντοπίζονται στα πολύ αρχικά στάδια
- Μείωση των πειραμάτων με το πραγματικό hardware (αεροπλάνο, υποβρύχιο) άρα και μείωση του συνολικού κόστους
- Μείωση κόστους συνολικής ανάπτυξης (στατιστικά περίπου κατά 50%)

#### Μειονεκτήματα :

- Η αυτόματη παραγωγή κώδικα δεν είναι πάντα βέλτιστη
- Εξάρτηση από συγκεκριμένο πακέτο (π.χ. xPC Target)

Σημειώνεται ότι η Mathworks παρέχει μπλοκ οδηγούς (driver blocks) για ένα σεβαστό αριθμό ηλεκτρονικών καρτών (CPU, I/O modules κ.α.), που ρόλο έχουν την αυτόματη παραγωγή κώδικα για επικοινωνία πραγματικού χρόνου με αυτές (βλ. Σχήμα 3.13). Για χρήση με το xPC Target υποστηρίζονται κάρτες διαφόρων λειτουργιών και φορμάτ, από αρκετούς κατασκευαστές ηλεκτρονικών ενσωματωμένων συστημάτων. Προβλήματα ανακύπτουν στην περίπτωση που το υλικό (hardware) προς χρήση, δεν περιλαμβάνεται στη λίστα αυτή [42]. Η μόνη λύση σε αυτό είναι ο προγραμματισμός εκ νέου του οδηγού της κάρτας στο πλαίσιο του xPC Target.



**Σχήμα 3.13** Επικοινωνία του Target PC με φυσικά συστήματα μέσω καρτών εισόδων/ εξόδων.

### 3.6 Αρχικός σχεδιασμός ηλεκτρικού υποσυστήματος

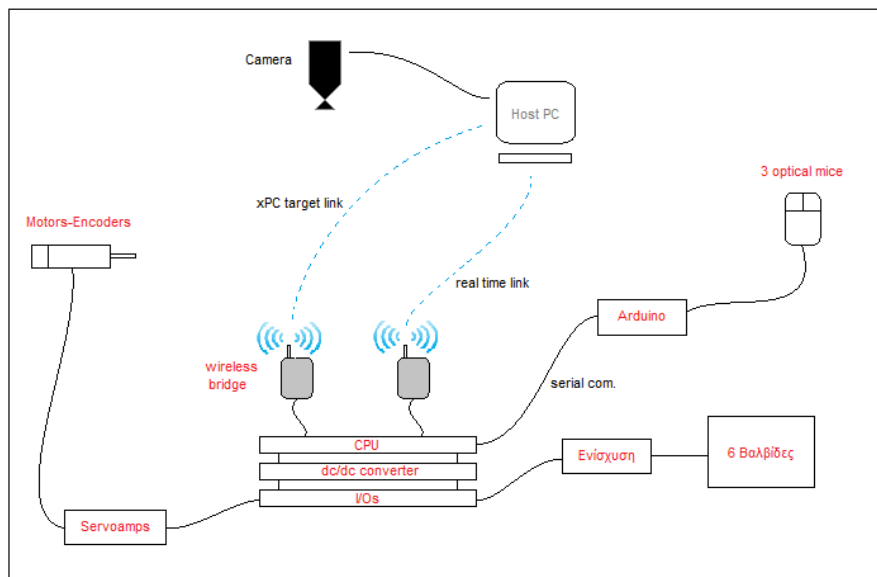
Στο σημείο αυτό περιγράφεται ο τρόπος που έγινε η επιλογή των ηλεκτρονικών πλακετών του νέου ρομπότ, με τρόπο, έτσι ώστε να είναι δυνατή η χρήση του πακέτου xPC Target. Αρχικά, επιδιώχθηκε η ομοιότητα με το υπολογιστικό σύστημα του πρώτου ρομπότ του εξομοιωτή, και επομένως οι πιθανές κάρτες περιορίστηκαν σε αυτές με φορμάτ PC-104 ή PC-104 plus. Εκτός των άλλων, είναι ένα φορμάτ που χρησιμοποιείται ευρέως στην ρομποτική και μπορεί εύκολα να προγραμματιστεί και χωρίς τη χρήση xPC Target.

Στη διαδικασία της επιλογής των καρτών του target υπολογιστή αντιμετωπίστηκαν κάποια προβλήματα. Αρχικά, δεν βρέθηκαν συμβατές κάρτες για ασύρματη επικοινωνία στις λίστες των υλικών (hardware). Επιπλέον, στην βιβλιογραφία και στο διαδίκτυο δεν βρέθηκε πουθενά να γίνεται λόγος για ασύρματη επικοινωνία σε περιβάλλον xPC Target. Σε αυτή τη βάση, η λύση που επινοήθηκε τελικά, μετά από έρευνα, ήταν η χρήση επικοινωνίας μέσω Ethernet (υποστηρίζεται από xPC Target) σε συνδυασμό με τη χρήση μίας ασύρματης Ethernet γέφυρας (wireless Ethernet bridge). Με τον τρόπο αυτό, το ρομπότ συνδέεται ασύρματα στο δίκτυο με έμμεσο τρόπο. Η σύνδεση αυτή καλύπτει μεν την ανάγκη για τον απαιτούμενο σύνδεσμο (host-target link) των υπολογιστών host και target μέσω Ethernet, δεν επαρκεί όμως και για τον δεύτερο δρόμο επικοινωνίας με το σύστημα της κάμερας. Απαιτείται έτσι η ενσωμάτωση κι ενός δεύτερου ασύρματου διαύλου επικοινωνίας, αφού στο xPC Target δεν μπορεί να επιτευχθεί επικοινωνία πραγματικού χρόνου μέσω του ήδη υπάρχοντος host - target συνδέσμου. Με γνώμονα το χαμηλό κόστος, αποφασίστηκε να μην αγοραστεί ξεχωριστή κάρτα για επικοινωνία μέσω Ethernet, μιας και οι περισσότερες κάρτες CPU έχουν ενσωματωμένα Ethernet chipsets.

Η τελική λοιπόν αναζήτηση της κάρτας CPU έγινε με κριτήρια: την υπολογιστική ισχύ, την χαμηλή ενεργειακή κατανάλωση, τη διπλή επικοινωνία μέσω Ethernet και την ενσωμάτωση επιπρόσθετων τρόπων επικοινωνίας όπως USB και Serial. Όπως φαίνεται και στον πίνακα στο ΠΑΡΑΡΤΗΜΑ Α,

αποδείχθηκε πολύ δύσκολη η εύρεση μιας κάρτας CPU PC-104 που να ενσωματώνει δύο συμβατά με το xPC Target Ethernet chipsets. Η τελική επιλογή παρουσιάζεται παρακάτω.

Σύμφωνα με ένα αρχικό λοιπόν σχεδιασμό, συμβατό με τις απαιτήσεις που τέθηκαν στο Κεφάλαιο 2, προκύπτει το σχέδιο του Σχήματος 3.14. Εκεί παρουσιάζεται το υπολογιστικό σύστημα του ρομπότ, συνδεδεμένο κατάλληλα με αισθητήρες και επενεργητές. Φαίνονται επίσης οι δύο δρόμοι επικοινωνίας, ένας για τον χειρισμό του ρομπότ κι ένας για την επικοινωνία πραγματικού χρόνου με το σύστημα της κάμερας. Η μορφή του υποσυστήματος των οπτικών αισθητήρων καθώς και τα υπόλοιπα τμήματα του σχήματος θα περιγραφούν αναλυτικά στα κεφάλαια που ακολουθούν.



Σχήμα 3.14 Αρχικός ηλεκτρικός/ ηλεκτρονικός σχεδιασμός.

### 3.8 Επιλογή ηλεκτρικών/ ηλεκτρονικών στοιχείων

#### 3.8.1 Κεντρική μονάδα επεξεργασίας CPU PC-104

Στον πίνακα του ΠΑΡΑΡΤΗΜΑΤΟΣ Α, παρουσιάζονται τα μοντέλα καρτών CPU PC-104 που ικανοποιούν τα κριτήρια που παρουσιάστηκαν παραπάνω. Ενώ η λίστα των καρτών που είναι συμβατές με xPC Target είναι αρκετά μεγάλη, οι πιθανές επιλογές σύμφωνα με τις σχεδιαστικές απαιτήσεις του παρόντος συστήματος είναι πολύ λίγες. Η κάρτα που επιλέχθηκε τελικά ήταν η PCM-4153 της Advantech, η οποία σε αρκετά χαμηλό κόστος, ικανοποιεί όλα τα κριτήρια που τέθηκαν προηγουμένως (βλ. Σχήμα ...). Στο ΠΑΡΑΡΤΗΜΑ Α, παρουσιάζονται αναλυτικά τα τεχνικά χαρακτηριστικά του μοντέλου.

#### 3.8.2 Κάρτα εισόδων/ εξόδων (PC-104)

Το υπολογιστικό σύστημα συμπληρώθηκε από μια ακόμη κάρτα PC-104 προκειμένου να παρέχει στο σύστημα τον απαραίτητο αριθμό ψηφιακών εισόδων/ εξόδων. Με βάση πάλι τις λίστες συμβατού υλικού (hardware) επιλέχθηκε η πολύ οικονομική κάρτα PC104-DIO48 της Measurement Computing (βλ. Σχήμα 3.15), η οποία παρέχει 48 ψηφιακές εισόδους/ εξόδους στο σύστημα. Στο ΠΑΡΑΡΤΗΜΑ Α, παρουσιάζονται αναλυτικά τα τεχνικά χαρακτηριστικά του μοντέλου.



**Σχήμα 3.15** Αριστερά: Η κεντρική μονάδα επεξεργασίας PCM-4153 της Advantech. Δεξιά: Κάρτα ψηφιακών εισόδων εξόδων PC104-DIO48 της Measurement Computing.

### 3.8.3 Ασύρματη γέφυρα Ethernet (Wireless Ethernet bridge)

Η ασύρματη επικοινωνία επιτεύχθηκε έτσι όπως περιγράφηκε παραπάνω, μέσω δύο ασύρματων γεφυρών Ethernet. Η μία χρησιμοποιείται για την χαμηλών απαιτήσεων επικοινωνία (host-target link) μεταξύ των υπολογιστών host και target, ενώ η δεύτερη για την real-time επικοινωνία του target υπολογιστή με τον υπολογιστή του συστήματος της κάμερας. Για τον σκοπό αυτό επιλέχθηκε το μοντέλο WNCE2001 της Netgear (βλ. Σχήμα 3.16), γιατί συγκεντρώνει πολύ ισχυρά χαρακτηριστικά σε χαμηλή τιμή. Στο ΠΑΡΑΡΤΗΜΑ Α, παρουσιάζονται αναλυτικά τα τεχνικά χαρακτηριστικά του μοντέλου.



**Σχήμα 3.16** Η ασύρματη γέφυρα Ethernet WNCE2001 της Netgear.

### 3.8.4 Μπαταρίες

Οι μπαταρίες που χρησιμοποιούνται ως επί το πλείστον σε αντίστοιχες εφαρμογές είναι οι μπαταρίες Ιόντων Λιθίου Πολυμερούς (LiPo). Τα στοιχεία Λιθίου - Πολυμερούς είναι σήμερα τα πλέον διαδεδομένα επαναφορτιζόμενα στοιχεία (βλ. Σχήμα 3.17). Προσφέρουν χαρακτηριστικά, όπως η υψηλή ειδική ενέργεια και η υψηλή πυκνότητα ενέργειας, σε ένα σχήμα μικρού πάχους. Πρόκειται για μπαταρίες πολύ ελαφριές, με τεράστια χωρητικότητα, απόδοση για το μέγεθος τους και γρήγορη εκφόρτιση με μεγάλες συγκριτικά εντάσεις. Τα στοιχεία LiPo έχουν ονομαστική τάση 3.7V. Η πρακτική περιοχή τάσης είναι από 4.2V όταν είναι φρεσκοφορτισμένα, έως 3V όταν εκφορτιστούν.

Είναι ευαίσθητες στην κακομεταχείριση, τοξικές και επικίνδυνες αν ανοιχτούν ή σκάσουν. Αποκλείεται ο κίνδυνος έκρηξης, αφού δεν μπορεί να αναπτυχθεί μεγάλη πίεση στο εσωτερικό του στοιχείου, όχι όμως και ο κίνδυνος ανάφλεξης. Κάθε μέρα βγαίνουν και νέα προϊόντα ελέγχου της κατάστασης τέτοιων μπαταριών.





**Σχήμα 3.17** Μπαταρία LiPo.

### Βασικές Έννοιες

- **Capacity:** Η χωρητικότητα της μπαταρίας δηλαδή πόσο ρεύμα μπορεί να δώσει σε 1 ώρα. Μετριέται σε μιλιαμπερόρια (mAh). Μία μπαταρία αποτελείται από στοιχεία της ίδιας χωρητικότητας.
- **Voltage:** Η τάση της μπαταρίας που μετριέται σε Volt (Βολτ) και εξαρτάται από τον αριθμό των στοιχείων (cells) που έχει.
- **Charge/ Discharge Rate (Ποσό φόρτισης/ αποφόρτισης):** Είναι στην πραγματικότητα το ρεύμα (σε Αμπέρ) που πρέπει να δίνεται ή να παίρνεται στην φόρτιση και στην εκφόρτιση αντίστοιχα.

Σε μια LiPo μπαταρία αναγράφονται δύο νούμερα (για παράδειγμα 4s2p). Το πρώτο νούμερο αφορά στα στοιχεία που έχει συνδεδεμένα στη σειρά και το δεύτερο πόσες σειρές έχει παράλληλα. ) Για να υπολογιστεί η συνολική τάση της μπαταρίας, ο αριθμός των στοιχείων σε σειρά πολλαπλασιάζεται με το 3.7 (τάση στοιχείου LiPo). Για να υπολογιστεί η συνολική χωρητικότητα πολλαπλασιάζεται η χωρητικότητα με το νούμερο στην δεύτερη στήλη.

### Φόρτιση

Τα στοιχεία LiPo απαιτούν ειδικές παραμέτρους φόρτισης και μόνο φορτιστές κατασκευασμένοι γι' αυτόν το σκοπό μπορούν να χρησιμοποιηθούν. Η μέγιστη τάση φόρτισης ανά στοιχείο (σε σειρά) είναι 4,235V. Στην πράξη η τάση φόρτισης ορίζεται στα 4.2V ή στα 4.15V για περισσότερη ασφάλεια. Αντίστοιχα, ο γενικός κανόνας ορίζει το μέγιστο ρεύμα φόρτισης στο 1C (όση η χωρητικότητα). Αν οι τιμές της τάσης ή της έντασης κατά την φόρτιση είναι μεγαλύτερες από τις επιτρεπόμενες, υπάρχει κίνδυνος ανάφλεξης ενός ή περισσότερων στοιχείων. Η φλόγα που βγαίνει από ένα στοιχείο LiPo θυμίζει βεγγαλικό και φθάνει τα 3-4 μέτρα. Μπορούν να φορτιστούν είτε με σύστημα σταθερού ρεύματος (CC), είτε με σύστημα σταθερού ρεύματος- σταθερής τάσης (CCCV). Σε κάθε περίπτωση, η μπαταρία φορτίζεται με σταθερό ρεύμα μέχρι να φτάσει το όριο των 4.2V . Από τη στιγμή εκείνη κι μετά, το ρεύμα μειώνεται συνεχώς έτσι ώστε να διατηρείται η τάση στα 4.2V. Όταν η τάση του στοιχείου φθάσει τα 4.2V, στο τέλος της φόρτισης η ένταση έχει πέσει στο 0.1-0.2C. Με αρχική ένταση το 1C η μπαταρία θα επανακτήσει το 90% της χωρητικότητάς της σε μία ώρα, ενώ το υπολειπόμενο 10% απαιτεί 45-50 ακόμη λεπτά. Σημειώνεται ότι, όταν φεύγουν από το εργοστάσιο κατασκευής τους, οι LiPo έχουν περίπου το 50% του φορτίου τους.

Οι συγκεκριμένες μπαταρίες θα πρέπει να έχουν στοιχεία το ίδιο φορτισμένα (με μία μικρή απόκλιση 0.1V) για να φορτιστούν σωστά και να μην είναι επικίνδυνες για υπερθέρμανση. Στο εμπόριο υπάρχουν μηχανήματα τα οποία λέγονται ισορροπιστές (balancers). Αυτά, αν υποστηρίζονται από την μπαταρία, μπορούν να αναλάβουν το έργο της ισορρόπησης της τάσης ανάμεσα στα στοιχεία της μπαταρίας. Μια όχι ισορροπημένη μπαταρία LiPo μπορεί να γίνει πολύ επικίνδυνη.

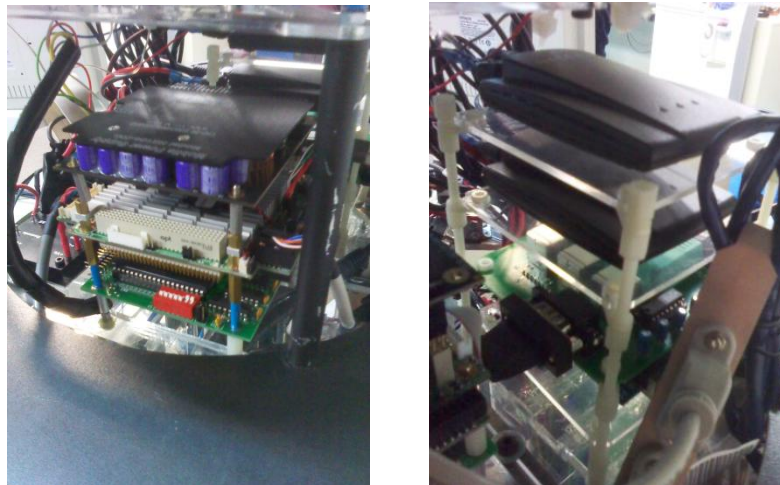


### 3.9 Το υλικό (hardware) σε λειτουργία

Στο Σχήμα 3.19 φαίνεται η σύνδεση των ηλεκτρικών εξαρτημάτων που αναφέρθηκαν παραπάνω σε μια πρώτη μορφή, πριν ακόμη εγκατασταθούν στο ρομπότ. Το βασικό κομμάτι του λογισμικού αναπτύχθηκε υπό αυτές τις συνθήκες, μέσω πολλών και αλληπάληλων πειραμάτων. Αφού αναπτύχθηκε και δοκιμάστηκε ο βασικός κορμός του λογισμικού, το σύστημα εγκαταστάθηκε τελικά στο ρομπότ σε προμελετημένες θέσεις, όπως φαίνεται στο Σχήμα 3.20.



*Σχήμα 3.19 Αρχική υλοποίηση του ηλεκτρικού σχεδίου.*



*Σχήμα 3.20 Αριστερά: πύργος PC-104 πάνω στο ρομπότ. Δεξιά: Ασύρματες γέφυρες Ethernet πάνω στο ρομπότ.*



# ΚΕΦΑΛΑΙΟ 4

## Εύρεση θέσης και προσανατολισμού από σύστημα κάμερας

### 4.1 Εισαγωγή

Για να θεωρηθεί το παρόν ρομποτικό σύστημα, ένα σύστημα αυτομάτου ελέγχου, είναι απαραίτητη η ικανότητα ακριβούς προσδιορισμού της κίνησής του στον χώρο εργασίας του. Τα προβλήματα που καλείται να εξετάσει το ρομπότ του διαστημικού εξομοιωτή είναι κυρίως προβλήματα δυναμικής στο επίπεδο. Βασική, όμως, προϋπόθεση για την ελεγχόμενη κίνηση του στο επίπεδο της τράπεζας του γρανίτη είναι η γνώση της θέσης και του προσανατολισμού του σε κάθε χρονική στιγμή με την επιθυμητή ακρίβεια.

Για το σκοπό αυτό, μέρος του διαστημικού εξομοιωτή είναι και μια ψηφιακή κάμερα, τοποθετημένη επάνω από τον χώρο δράσης των ρομπότ, με στόχο τη λήψη διαδοχικών στιγμιότυπων αυτού. Με επεξεργασία των στιγμιότυπων αυτών, είναι δυνατός ο προσδιορισμός της θέσης και του προσανατολισμού των κινούμενων σωμάτων του διαστημικού εξομοιωτή. Η μέθοδος αυτή ελέγχου ρομποτικών διατάξεων είναι γνωστή ως οπτική οδήγηση κλειστού βρόχου μέσω ανάδρασης θέσης

(visual servoing) και είναι αρκετά διαδεδομένη τα τελευταία χρόνια λόγω της αυξανόμενης ισχύος των υπολογιστικών συστημάτων και της κυκλοφορίας καμερών υψηλής απόδοσης και χαμηλού κόστους.

Η μελέτη του συστήματος της κάμερας, η επιλογή, η αγορά και η τοποθέτηση των εξαρτημάτων έγινε από τον Ι. Κοντολάτη και περιγράφεται αναλυτικά στο [3]. Εδώ θα αναφερθούν για λόγους πληρότητας τα βασικά χαρακτηριστικά του συστήματος και οι κύριοι λόγοι που αυτό έχει την παρούσα μορφή. Παρακάτω περιγράφεται συνοπτικά η διαδικασία εκλογής των στοιχείων του συστήματος, όπως έγινε από τον Ι. Κοντολάτη.

## 4.2 Χαρακτηριστικά συστήματος κάμερας

Αρχικά, η επιλογή του οπτικού αισθητήρα και του φακού της κάμερας έγιναν έτσι, ώστε τα στιγμιότυπα να καλύπτουν τον χώρο δράσης, η ανάλυση της εικόνας και η συχνότητα λήψης να είναι αποδεκτές και το κόστος να κυμαίνεται σε λογικά πλαίσια. Οπότε, οι φυσικές παράμετροι της πειραματικής διάταξης έθεσαν τους περιορισμούς στην επιλογή του συστήματος της κάμερας αλλά και στον τρόπο στήριξης του. Πιο αναλυτικά, τα φυσικά χαρακτηριστικά είναι τα εξής: Η τράπεζα, διαστάσεων 2200 x 1800 [mm] και βάρους 4 τόνων, είναι κατασκευασμένη από γρανίτη, με τραχύτητα επιφάνειας μικρότερη των 5 [μm] και αποτελεί την επιφάνεια κίνησης του ρομπότ. Η άνω επιφάνεια της τράπεζας απέχει από την οροφή του χώρου περίπου 1800 [mm]. Η θέση της τράπεζας βρίσκεται κάτω από δοκό στήριξης του κτηρίου πάχους 300[mm] και ύψους 250 [mm]. Η απόσταση της άνω επιφάνειας του γρανίτη από την επιφάνεια της δοκού είναι 1550 [mm] (βλ. Σχήμα 4.1).

Ο συνδυασμός του ύψους και της έκτασης της περιοχής δράσης οδήγησαν στην επιλογή ευρυγώνιου φακού. Ως ικανοποιητική ανάλυση καθορίστηκε εκείνη η οποία επιτρέπει τον καθορισμό της θέσης των αντικειμένων με ακρίβεια [mm], και είναι της τάξης των 4 εκατομμυρίων εικονοστοιχείων (4 MPixels). Ο ελάχιστος επιθυμητός ρυθμός δειγματοληψίας στιγμιότυπων καθορίστηκε στο 1 fps (frame per second). Κρίθηκε ότι, το χρώμα δεν είναι απαραίτητη πληροφορία για την συγκεκριμένη εφαρμογή.



**Σχήμα 4.1** Αριστερά: Η κάμερα mvBlueFOX-124G. Δεξιά: Φαίνεται ο τρόπος στήριξης της πάνω από τον χώρο εργασίας.

Με βάση τα προαναφερθέντα, και με μεγαλύτερη επιρροή να έχει το κριτήριο του κόστους, επιλέχθηκε η κάμερα mvBlueFOX-124G της εταιρείας MATRIX VISION με αισθητήρα τύπου CCD, απεικόνιση στη κλίμακα του γκρι με μέγεθος στοιχείου 8bits, ανάλυσης 1600x1200 εικονοστοιχείων, ρυθμού λήψης 8 fps, πρωτόκολλο επικοινωνίας USB 1.1/2.0 (έως 480Mbits/sec), συμβατότητα με λειτουργικά συστήματα Microsoft Windows και Linux, και κόστος αγοράς 1640 €. Πιο συγκεκριμένα, ο αισθητήρας CCD είναι της εταιρείας Sony με κωδικό ICX274AL/AQ, διαγώνιο 8.923 [mm], κατηγορίας 1/1.8", με μέγεθος εικονοστοιχείου 4.4 x 4.4 [μm]. Επίσης, η δυνατότητα έκθεσης είναι από 50 [μs] έως 10 [s] με βήμα 1 [μs]. Η κάμερα συνοδεύεται από βιβλιοθήκη εντολών για τη ρύθμιση και τον έλεγχο πληθώρας παραμέτρων, έτοιμα προγράμματα ανοικτού κώδικα για τη κτήση και επεξεργασία εικόνας, αλλά και πραγματοποίηση ρυθμίσεων. Επιπλέον, διαθέτει εξωτερικό μεταλλικό περίβλημα με οπές για τη τοποθέτηση της σε βάση, ενώ το βάρος της είναι 120gr. Τέλος, δέχεται οπτικούς φακούς έδρασης τύπων C, CS και S προαιρετικά.

Ο οπτικός φακός που επιλέχθηκε είναι ο H2Z0414C-MP της εταιρείας Computar, ο οποίος προορίζεται για χρήση σε κάμερες με αισθητήρα ανάλυσης εκατομμυρίων εικονοστοιχείων. Το κόστος του είναι 160 € και διαθέτει κυμαινόμενη εστιακή απόσταση  $f$  4-8 [mm] για να επιτευχθεί η όσο το δυνατόν μεγαλύτερη κάλυψη της επιφάνειας δράσης με τη λιγότερη δυνατή παραμόρφωση. Το βάρος του είναι 72 [gr], με διατομή 41.6 [mm] και μήκος 48.8 [mm], ενώ είναι έδρασης τύπου C. Ο φακός διαθέτει τρεις δακτυλίους για τη χειροκίνητη ρύθμιση της εστίασης (0,5 [m] ως άπειρο), της εστιακής απόστασης (4 – 8 [mm]) και του ανοίγματος του κλείστρου (F1.4 – F16C).

Η ανάλυση των 2MP παρέχει την ακρίβεια που απαιτείται, ενώ η ταχύτητα των 480Mbps/s που επιτυγχάνει το πρότυπο USB2.0 σε συνδυασμό με τις δυνατότητες του αισθητήρα CCD δίνουν ρυθμό ανανέωσης της εικόνας 8 fps, τιμή ικανοποιητική για την συγκεκριμένη εφαρμογή.

### 4.3 Γενική ιδέα προσδιορισμού θέσης και προσανατολισμού

Η ιδέα βασίζεται στην τοποθέτηση φωτεινών πηγών (φωτοдиодοι LEDs) σε κατάλληλους γεωμετρικούς σχηματισμούς στην οροφή των κινούμενων σωμάτων, με στόχο την ανίχνευση τους από την κάμερα και την εξαγωγή συμπερασμάτων για την θέση και τον προσανατολισμό τους. Πρέπει να σημειωθεί ότι, η σωστή ανίχνευση των φωτεινών πηγών προϋποθέτει την απουσία άλλων φωτεινών πηγών (σκοτάδι) την ώρα του πειράματος. Δεδομένου ότι η δυναμική των σωμάτων δεν πρέπει να επηρεάζεται από εξωτερικές δυνάμεις (π.χ. καλώδια), η ανατροφοδότηση της πληροφορίας αυτής (θέση και προσανατολισμός) στα υπολογιστικά συστήματα των ρομπότ πρέπει να πραγματοποιείται ασύρματα και σε πραγματικό χρόνο για να κλείνει ο βρόχος και να ελέγχεται η κίνηση τους. Η ασύρματη αυτή επικοινωνία υλοποιείται με τη χρήση ενός ενδιάμεσου εξωτερικού υπολογιστή, ο οποίος συνδέεται με την κάμερα μέσω USB, τρέχει κατάλληλους αλγόριθμους επεξεργασίας εικόνας και τελικά στέλνει ασύρματα τα χρήσιμα δεδομένα στα ρομπότ. Ο υπολογιστής αυτός, ο οποίος έχει χρέη κέντρου ελέγχου (control center) στον διαστημικό εξομοιωτή, τρέχει λειτουργικό σύστημα Windows 7.

Έχουν μελετηθεί και εφαρμοσθεί αρκετοί συνδυασμοί διατάξεων φωτοдиодων LEDs και αλγορίθμων αναγνώρισης τους, οι οποίοι περιγράφονται αναλυτικά παρακάτω. Σε όλες τις μεθόδους που δοκιμάστηκαν, οι φωτεινές πηγές, δεν ήταν άλλες από διόδους φωτοεκπομπής LEDs, και γι' αυτό στη συνέχεια θα αναφέρονται μόνο ως LEDs. Σε κάθε τέτοιο αλγόριθμο, είναι αναγκαία η ύπαρξη ενός αλγορίθμου κορμού, στον οποίο θα γίνεται η αναγνώριση των θέσεων σε [mm] όλων των μεμονωμένων φωτοдиодων LEDs, ως προς ένα παγκόσμιο σύστημα συντεταγμένων. Στη συνέχεια μόνο, λαμβάνοντας υπόψη τις πραγματικές θέσεις των LEDs πάνω στα σώματα, μπορούν να εφαρμοσθούν αλγόριθμοι προσδιορισμού θέσης/προσανατολισμού των σωμάτων αυτών. Αρχικά λοιπόν, θα αναλυθεί η απαραίτητη αυτή προκαταρκτική εργασία και θα περιγραφεί η δομή του αντίστοιχου προγράμματος, το οποίο έχει αναπτυχθεί από τους Ι. Κοντολάτη, Γ. Ρεκλείτη και Α. Καλγρεάδη [12].

### 4.4 Αλγόριθμος εύρεσης θέσης μεμονωμένων φωτεινών πηγών (LEDs)

Αρχικά, πρέπει να αναφερθεί ότι η εταιρεία Matrix Vision συνοδεύει την κάμερα mvBlueFOX με ένα σύνολο βιβλιοθηκών συναρτήσεων για την ανάπτυξη εφαρμογών με σκοπό τη πραγματοποίηση ρυθμίσεων στη λειτουργία της κάμερας αλλά και στη διαδικασία κτήσης της εικόνας. Η ανάπτυξη των εφαρμογών μπορεί να πραγματοποιηθεί σε γλώσσα προγραμματισμού C/C++. Στη εργασία [3] αναπτύχθηκε μια πρώτη μορφή του προγράμματος σε γλώσσα προγραμματισμού C++, η οποία εξελίχθηκε μεταγενέστερα από τους Γ. Ρεκλείτη και Α. Καλγρεάδη. Τα βήματα αυτού του αλγορίθμου κορμού, από την λήψη της εικόνας μέχρι και τον προσδιορισμό των θέσεων όλων των LEDs, είναι τα εξής:

#### A. Μέθοδος εμφάνισης γκρι αποχρώσεων (Halftoning)

Αρχικά, λαμβάνεται το τρέχον στιγμιότυπο από την κάμερα και αποθηκεύεται κατάλληλα. Το στιγμιότυπο σε αυτή την φάση έχει τη μορφή εικόνας σε κλίμακα του γκρι. Στο επόμενο βήμα, εφαρμόζεται ένας αλγόριθμος εμφάνισης του γκρι (Halftoning) [3], ο οποίος μετασχηματίζει μία

εικόνα κλίμακας του γκρι σε μία εικόνα που αποτελείται μόνο από 1 (μαύρο) και 0 (άσπρο). Με το μετασχηματισμό αυτό μειώνεται ο όγκος της πληροφορίας της εικόνας και εντοπίζονται καλύτερα τα ζητούμενα χαρακτηριστικά. Τα δεδομένα, λοιπόν, περνούν στην υπορουτίνα εφαρμογής του αλγόριθμου halftone, υπόκεινται σε επεξεργασία, και κατασκευάζεται το νέο μητρώο της εικόνας με τιμές 0 για τα μαύρα εικονοστοιχεία και μία μεγάλη τιμή για τα εικονοστοιχεία των φωτεινών περιοχών ανάλογα με τη τιμή ορίου που έχει τεθεί (βλ. επιβολή κατωφλίου παρακάτω).

## **B. Επιβολή κατωφλίου και αύξηση περιοχής**

Δύο θέματα, κοινά σε όλες τις μεθόδους κατάτμησης εικόνας, είναι η επιβολή κατωφλίου (threshold) και η αύξηση περιοχής. Η επιβολή κατωφλίου σε μία εικόνα θέτει τα εικονοστοιχεία της να έχουν τιμή 1 ή 0 και συμπληρώνει τον αλγόριθμο halftone που αναφέρθηκε παραπάνω. Η τιμή κατωφλίου μπορεί να ρυθμίζεται αλλάζοντας την τιμή της μεταβλητής threshold, και αποτελεί μια πολύ σημαντική παράμετρο του αλγορίθμου επεξεργασίας εικόνας. Μια τιμή που παρουσίασε καλά αποτελέσματα στα πειράματα είναι:  $threshold = 245$ ; και προτείνεται σαν αρχική τιμή σε μελλοντικά πειράματα.

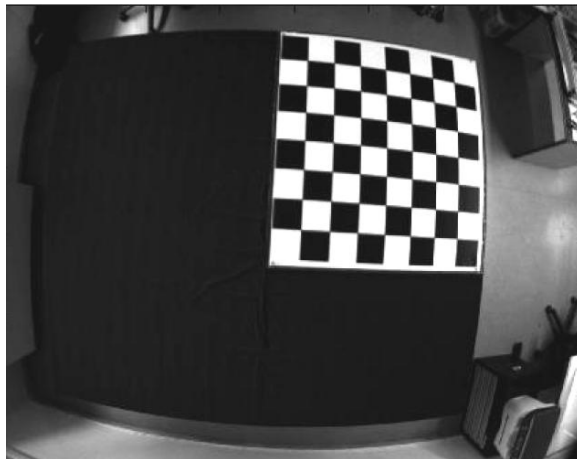
## **Γ. Προσδιορισμός θέσης μεμονωμένων LEDs**

Αμέσως μετά, βρίσκονται όλα τα 'νησιά' τιμής 1 στο μητρώο που πλέον αντιστοιχεί στην εικόνα. Κάθε τέτοια ομάδα γειτονικών τιμών 1 θεωρητικά αποτελεί ένα LED. Απαραίτητη βέβαια προϋπόθεση για να καταχωρηθεί ως LED, είναι η μικρή απόσταση των στοιχείων τιμής 1 μεταξύ τους και το επαρκώς μεγάλο συνολικό μέγεθος της ομάδας των στοιχείων. Η μέγιστη επιτρεπτή απόσταση στοιχείων 1 καθορίζεται στο πρόγραμμα από την παράμετρο/μεταβλητή  $pdis = 30$ ; , ενώ το ελάχιστο επιτρεπτό συνολικό μέγεθος της ομάδας (σε pixels) καθορίζεται από την παράμετρο/ μεταβλητή  $pthr = 30$ ; . Οι τελευταίες παράμετροι χρησιμοποιούνται έτσι, ώστε να αποφεύγεται η αναγνώριση αμελητέων φωτεινών περιοχών ως LEDs. Τέτοιες μικρές περιοχές μπορούν να είναι σκιάσεις, αντανάκλασεις του φακού ή άλλα τυχαία γεγονότα που συμβαίνουν κατά τη διάρκεια των πειραμάτων.

## **Δ. Αφαίρεση παραμόρφωσης του φακού**

Εφόσον βρεθούν οι θέσεις των τιμών 1 που αντιστοιχούν σε κάθε φωτεινή πηγή LED στο μητρώο, πρέπει να μεταφραστούν σε θέσεις εκφρασμένες σε [mm]. Αυτή θα ήταν μια απλή διαδικασία, αν δεν υπήρχε η παραμόρφωση του ευρυγώνιου φακού που στρεβλώνει τον φαινόμενο χώρο εργασίας (βλ. Σχήμα 4.2). Μέσω ενός αλγορίθμου εξάλειψης της παραμόρφωσης που αναπτύχθηκε από τον Α. Καλγρεάδη στο [12], μπορούν να υπολογιστούν οι πραγματικές θέσεις των LEDs που αντιστοιχούν στα φωτεινά εικονοστοιχεία (pixels), με ακρίβεια μερικών [mm]. Τέλος, με απλές πράξεις, υπολογίζονται οι θέσεις των κέντρων των LEDs. Οι συντεταγμένες, λοιπόν, όλων των LEDs που έχει ανιχνεύσει η κάμερα ως προς ένα παγκόσμιο σύστημα συντεταγμένων, είναι τελικά αποθηκευμένες σε κατάλληλες μεταβλητές, και η βάση του αλγορίθμου επεξεργασίας εικόνας είναι έτοιμη.

Σημειώνεται ότι το ρομπότ έχει κάποια φωτεινά στοιχεία, όπως π.χ. τα ενδεικτικά LEDs των πλακετών του ή οι αντανάκλασεις των μεταλλικών του μερών, που μπορούν να δημιουργήσουν προβλήματα στον προηγούμενο αλγόριθμο επεξεργασίας. Το πιο κλασικό πρόβλημα που εμφανίζεται, είναι η αναγνώριση των προαναφερθέντων φωτεινών περιοχών ως LEDs, ενώ αυτά δεν είναι. Οι ενέργειες που έγιναν για την αντιμετώπιση του προβλήματος αυτού, ήταν η δοκιμή διαφορετικών τιμών των παραμέτρων επεξεργασίας, και η κάλυψη του 'επικίνδυνης' φωτεινότητας μέρους του ρομπότ (πάνω μέρος) με κατάλληλο μαύρο κάλυμμα.



**Σχήμα 4.2** Παραμορφωμένο στιγμιότυπο λόγω του ευρυγώνιου φακού της κάμερας.

#### 4.5 Αλγόριθμοι εύρεσης θέσης/ προσανατολισμού των κινούμενων σωμάτων

Από το σημείο αυτό, στο οποίο είναι γνωστές οι θέσεις όλων των φωτεινών πηγών που ανιχνεύονται από την κάμερα στον χώρο εργασίας, μπορούν να εφαρμοσθούν πολλές ιδέες εύρεσης θέσης και προσανατολισμού των κινούμενων σε αυτόν σωμάτων. Οι τρόποι εντοπισμού που επινοήθηκαν και αναπτύχθηκαν, παρουσιάζονται παρακάτω.

##### 4.5.1 Αρχική ιδέα και πρώτη υλοποίηση

Η πρώτη ιδέα που επινοήθηκε και υλοποιήθηκε στο εργαστήριο πριν εκπονηθεί η παρούσα εργασία, ήταν η τοποθέτηση φωτεινών πηγών LEDs σε κατάλληλο γεωμετρικό σχηματισμό στην οροφή των σωμάτων. Έτσι και υλοποιήθηκε στο πρώτο ρομπότ, όπου τοποθετήθηκαν 5 φωτοδιόδοι LEDs, σε σχηματισμό που θύμιζε τον αστερισμό της Κασσιόπης (βλ Σχήμα 4.3). Ο αριθμός των φωτοδιόδων έχει σημασία, καθώς πρέπει να υπάρχουν τουλάχιστον τρεις, προκειμένου να είναι δυνατός ο προσδιορισμός της θέσης και του προσανατολισμού του σώματος που τα φέρει. Σε αυτήν την πρώτη υλοποίηση, χρησιμοποιήθηκαν περισσότερα LEDs για λόγους αξιοπιστίας και ακρίβειας. Σημαντικό είναι επίσης, να είναι τοποθετημένα έτσι, ώστε οι ανά μεταξύ τους αποστάσεις να είναι όλες μοναδικές, όταν παρατηρούνται κάθετα από πάνω.

Η υλοποίηση αυτή φαινόταν να παρουσιάζει ελαττώματα τα οποία δημιουργούσαν προβλήματα στα πειράματα. Ένα από αυτά, προκύπτει από τη δυσκολία τοποθέτησης 5 LEDs σε μοναδικά διαφορετικές μεταξύ τους αποστάσεις, σε βαθμό τέτοιο, ώστε να μη δημιουργείται πρόβλημα από την (παραμένουσα) παραμόρφωση του ευρυγώνιου φακού. Τα περισσότερα βέβαια προβλήματα αφορούσαν προγραμματιστικά λάθη και προβλήματα ρύθμισης των παραμέτρων της επεξεργασίας εικόνας.



**Σχήμα 4.3** Αριστερά: Αστερισμός της Κασσιόπης. Δεξιά: Τοποθέτηση LEDs σε σχηματισμό όμοιου με τον αστερισμό.

Με την πρώτη επανεξέταση του θέματος αυτού, υπό το πρίσμα της παρούσας διπλωματικής, αναπτύχθηκε ένας νέος αλγόριθμος ο οποίος υλοποιούσε απλούστερα την παραπάνω ιδέα με χρήση πινάκων. Αυτός παρουσιάζεται σε μορφή εξομοίωσης σε Matlab στο ΠΑΡΑΡΤΗΜΑ Γ και δίνεται σαν αρχείο (sim\_code\_pos\_detection) στο CD που συνοδεύει την εργασία αυτή. Τελικά, ενσωματώθηκε από τον Γ. Ρεκλείτη στον υπάρχοντα κώδικα C++ και έγιναν κάποια πρώτα πειράματα.

Γρήγορα φάνηκε ότι, ο αλγόριθμος αυτός δεν ήταν βέλτιστος για τις απαιτήσεις του εξομοιωτή. Επιθυμητός ήταν ένας γρήγορος, ακριβής και αξιόπιστος αλγόριθμος. Τα δεδομένα που μεταφέρονται είναι σημαντικά για τον έλεγχο των ρομπότ, και επομένως, η ταχύτητα επεξεργασίας και ο χρόνος αποστολής είναι μεγάλης σημασίας. Ο υπάρχον αλγόριθμος μπορεί να έδινε ικανοποιητικά αποτελέσματα σε πειράματα με ένα ρομπότ, αλλά η πολυπλοκότητα του (έλλειψη 'έξυπνων' στοιχείων) θα έδινε στο μέλλον μη αποδεκτά αποτελέσματα σε πειράματα εντοπισμού 3 σωμάτων. Τρία σώματα αντιστοιχούν σε 15 περίπου LEDs (με τη μέθοδο αυτή) και τελικά περισσότερες από 100 αποστάσεις πρέπει να υπολογιστούν και να αξιολογηθούν στον βρόχο ελέγχου. Αυτό οδήγησε στον σχεδιασμό νέων αλγορίθμων, που εφαρμόζουν την ιδέα των φωτεινών πηγών LEDs λίγο διαφορετικά.

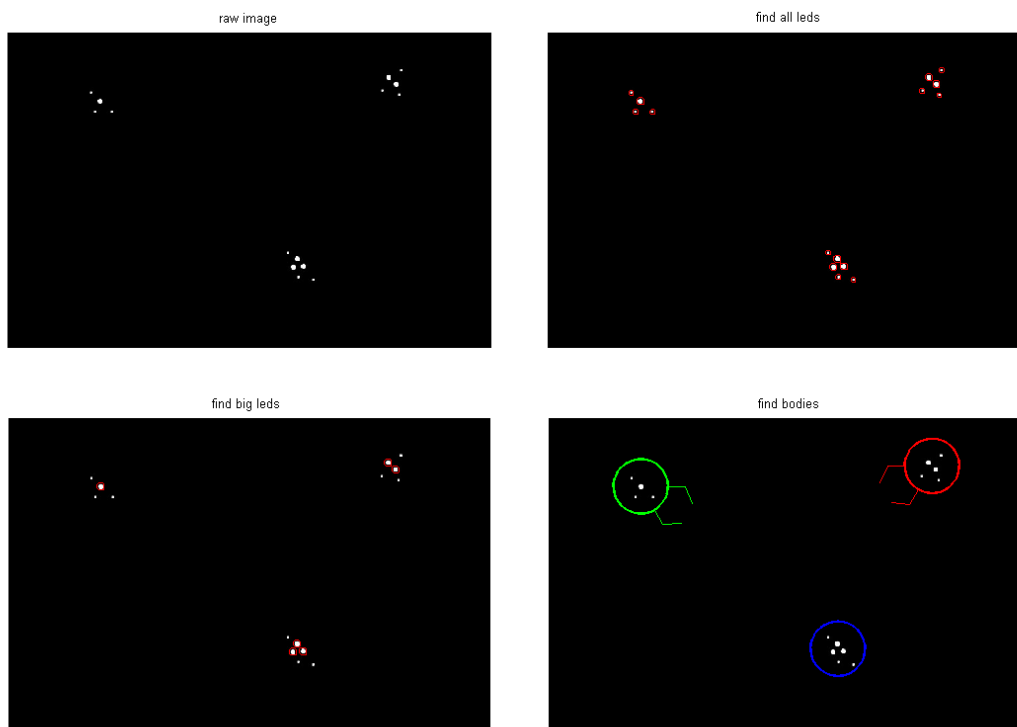
## 4.5.2 Αλγόριθμος δύο επιπέδων φωτεινότητας και πειράματα

### Αλγόριθμος εύρεσης τριών σωμάτων

Με απώτερο στόχο έναν αποτελεσματικό αλγόριθμο για τον εντοπισμό όχι ενός, αλλά τουλάχιστον τριών σωμάτων στο χώρο εργασίας, αποφασίστηκε η χρήση φωτεινών πηγών διαφορετικής φωτεινότητας. Αν για παράδειγμα, LEDs μεγαλύτερης φωτεινότητας τοποθετηθούν σε καίριες θέσεις στις οροφές των σωμάτων (π.χ. στα κέντρα τους), τότε είναι δυνατό να μειωθεί κατά πολύ το υπολογιστικό κόστος, η πολυπλοκότητα αλλά και το μέγεθος του αλγορίθμου επεξεργασίας εικόνας. Αυτό συμβαίνει γιατί, είναι εξαρχής γνωστό, ότι τα πιο φωτεινά LEDs βρίσκονται στα κέντρα των σωμάτων (ή και σε άλλες γνωστές θέσεις), και επομένως εξοικονομείται υπολογιστικός χρόνος, αφού πολύ λιγότερες αποστάσεις χρειάζεται πια να υπολογιστούν και να αξιολογηθούν.

Για να ελεγχθεί η ορθότητα του παραπάνω συλλογισμού, αναπτύχθηκε κώδικας προσομοίωσης με τη μορφή ενός Matlab Script (simulation\_multiple\_bodies\_detection στο CD). Δίνεται σαν είσοδος ένα υποθετικό στιγμιότυπο με τρία σώματα (από τα οποία διακρίνονται μόνο τα LEDs της οροφής τους), γίνεται η κατάλληλη επεξεργασία και σαν έξοδοι δίνονται οι θέσεις/ προσανατολισμοί αυτών. Ο τρόπος τοποθέτησης των LEDs που προτείνεται και υλοποιείται είναι ο εξής: Το πρώτο σώμα έχει 1 φωτεινό LED, το δεύτερο σώμα 2 φωτεινά LEDs, ενώ το τρίτο σώμα (στόχος) 3 φωτεινά LEDs (ή εναλλακτικά κανένα φωτεινό). Με αυτόν τον τρόπο τοποθέτησης επιτυγχάνεται με εύκολο τρόπο ο διαχωρισμός των σωμάτων ως εξής:

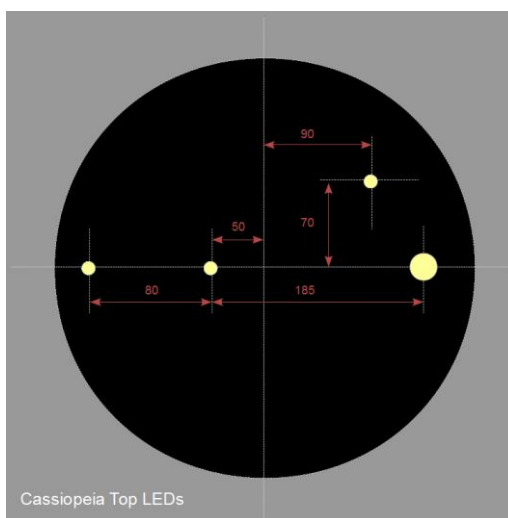
1. Σε πρώτο βήμα, έχουν εντοπιστεί όλα τα LEDs, με βάση τον αλγόριθμο κορμό, που αναπτύχθηκε προηγουμένως. Σαν επιπρόσθετο χαρακτηριστικό, όμως, αποθηκεύονται και τα μεγέθη (σε pixels) που φαίνονται να έχουν τα LEDs.
2. Στη συνέχεια, γίνεται διαχωρισμός ανάλογα με το μέγεθος τους. Έτσι, στο συγκεκριμένο παράδειγμα αναγνωρίζονται 6 φωτεινά και 9 λιγότερο φωτεινά LEDs.
3. Με βάση τις μεταξύ τους αποστάσεις τα φωτεινά LEDs χωρίζονται σε 3 ομάδες. Όσα βρίσκονται σε κοντινές αποστάσεις, τοποθετούνται στην ίδια ομάδα και αντιστοιχούν σε ένα σώμα.
4. Εδώ σημειώνεται ότι, τα φωτεινά LEDs έχουν τοποθετηθεί έτσι, ώστε το γεωμετρικό κέντρο των σχημάτων που σχηματίζουν τα γειτονικά, να ταυτίζεται με το κέντρο του αντίστοιχου σώματος. Με αυτόν τον τρόπο δεν χρειάζεται να αναγνωριστούν τα φωτεινά LEDs ενός σώματος, γιατί το μόνο που ενδιαφέρει είναι οι συντεταγμένες του βαρύκεντρου του σχήματος που αυτά σχηματίζουν. Με τον τρόπο αυτό μειώνεται επιπλέον η πολυπλοκότητα του αλγορίθμου επεξεργασίας.
5. Έχοντας εντοπίσει τα κέντρα των τριών σωμάτων, κατανέμονται στα σώματα και τα υπόλοιπα LEDs μικρότερης φωτεινότητας, με βάση την απόσταση τους από τα κέντρα αυτά.
6. Τελικά, για κάθε σώμα είναι γνωστά, το κέντρο του και οι θέσεις των μικρότερης φωτεινότητας LEDs. Συγκρίνοντας την τοπολογία αυτή με την γνωστή τοπολογία των LEDs στα σώματα, προκύπτουν οι θέσεις και οι γωνίες που έχουν τα σώματα στο τρέχον στιγμιότυπο (βλ. Σχήμα 4.4).



**Σχήμα 4.4** Τα βήματα του αλγορίθμου εντοπισμού τριών σωμάτων σε εικόνες, όπως παράχθηκαν από τον κώδικα προσομοίωσης σε Matlab.

**Πρώτα πειράματα**

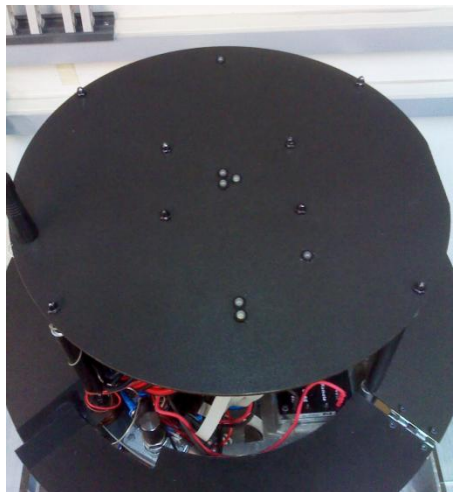
Με βάση αυτή την ιδέα και έχοντας πάρει θετικά αποτελέσματα από την προσομοίωση σε Matlab, έγιναν κάποια πρώτα πειράματα. Πρώτα, δοκιμάστηκε ο προσδιορισμός θέσης/ προσανατολισμού ενός σώματος, το οποίο έφερε ένα φωτεινό LED και άλλα τρία μικρότερης φωτεινότητας (βλ. Σχήμα 4.5). Πειράματα έγιναν από τον Α. Καλγρεάδη με τη βοήθεια ξύλινου ομοιώματος με ροδάκια αντί για αεροέδρανα (για εξοικονόμηση αερίου CO<sub>2</sub>). Τα αποτελέσματα, τα οποία παρουσιάζονται στο [12] ήταν καλά ως προς την ακρίβεια, και αποτέλεσαν την πρώτη πειραματική επιβεβαίωση, της ιδέας των δύο επιπέδων φωτεινότητας.



**Σχήμα 4.5** Η διάταξη των LEDs κατά τα πρώτα πειράματα.

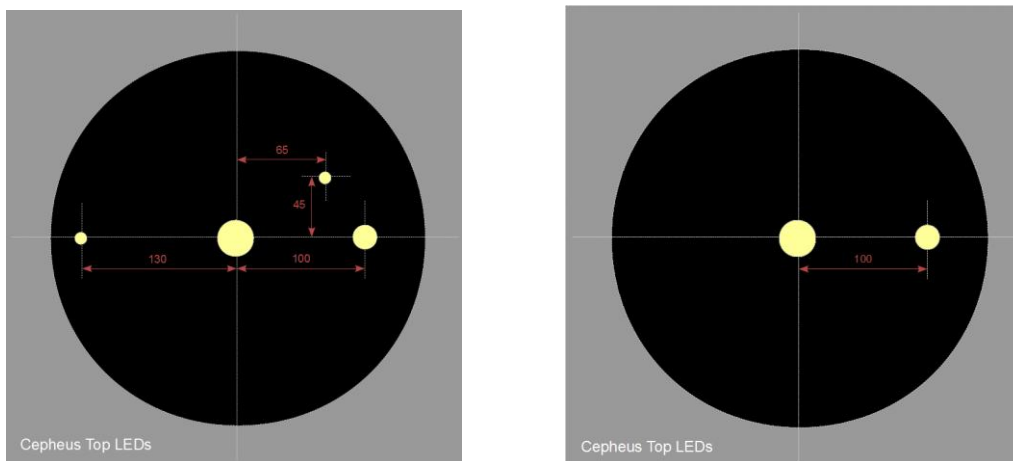
### Πειράματα με πραγματικό ρομπότ

Την ολοκλήρωση του ηλεκτρικού/ ηλεκτρονικού υποσυστήματος του δεύτερου ρομπότ, σηματοδότησε η κατασκευή της οροφής του, μαζί με την τοποθέτηση των LEDs αναγνώρισης πάνω σε αυτή. Σύμφωνα με την μέθοδο που αναφέρθηκε πριν, σχεδιάστηκε και τοποθετήθηκε, με την βοήθεια του I. Παρασκευά [4] [11], το σύστημα των LEDs, όπως φαίνεται στο Σχήμα 4.6. Ενώ θα μπορούσε να χρησιμοποιηθεί ένα LED αυξημένης φωτεινότητας, προτιμήθηκε να χρησιμοποιηθούν μικρά LEDs τοποθετημένα σε πολύ μικρές αποστάσεις, με το ίδιο τελικό αποτέλεσμα. Τα επίπεδα φωτεινότητας, προνοήθηκε να είναι τρία (τριπλό, διπλό, και μονά LEDs), σε αντίθεση με την προσομοίωση (δύο επίπεδα), για μεγαλύτερη ευελιξία. Ανάλογα με το πείραμα που διεξάγεται, υπάρχει η δυνατότητα αποσύνδεσης μερικών LEDs, για την επίτευξη διαφορετικών σχηματισμών.



**Σχήμα 4.6** Η οροφή του δεύτερου ρομπότ με τα LEDs όπως τοποθετήθηκαν τελικά.

Τα προγράμματα που αναπτύχθηκαν και τα πειράματα που διεξάχθηκαν στη συνέχεια, αφορούν στον πιο απλό γεωμετρικό σχηματισμό που γινόταν να δοκιμαστεί, και αυτός αποτελείται από το τριπλό LED στο κέντρο του ρομπότ και ένα μονό LED σε κάποια απόσταση από το πρώτο (βλ. Σχήμα 4.7). Αυτό υλοποιείται εύκολα αποσυνδέοντας τα κατάλληλα LEDs. Οι δύο αυτές φωτεινές πηγές είναι οι ελάχιστες απαραίτητες για τον προσδιορισμό θέσης και γωνίας του ρομπότ, και γι' αυτό επιλέχθηκε στα πρώτα πειράματα. Η χρήση και των υπολοίπων LEDs μπορεί με τον κατάλληλο αλγόριθμο, να οδηγήσει σε μεγαλύτερη ακρίβεια στην εκτίμηση θέσης/ προσανατολισμού, αλλά αυτό είναι κάτι που θα βασιστεί στα αποτελέσματα των πειραμάτων της απλής διάταξης και επομένως προτείνεται σαν μελλοντική εργασία.



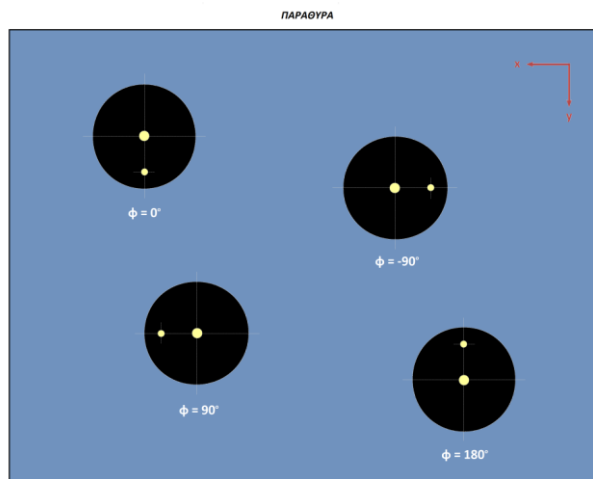
**Σχήμα 4.7** Αριστερά: Όλα τα LEDs συνδεδεμένα. Δεξιά: Απλοποιημένη διάταξη- αποσύνδεση κάποιων LEDs.



#### 4.6 Ορισμός του Παγκόσμιου Συστήματος Συντεταγμένων

Ο βασικός αλγόριθμος επεξεργασίας εικόνας που περιγράφηκε στην αρχή του κεφαλαίου, αναφέρεται σε ένα παγκόσμιο σύστημα συντεταγμένων, το οποίο εξαρτάται από τον τρόπο που η κάμερα 'βλέπει' τον χώρο εργασίας. Το πρόγραμμα C++ λοιπόν, δίνει σαν εξόδους συντεταγμένες ορισμένες ως προς ένα σύστημα που βρίσκεται κοντά στην πάνω δεξιά γωνία της τράπεζας του γρανίτη (κοιτώντας προς τα παράθυρα του εργαστηρίου). Δεδομένου ότι, το κέντρο ελέγχου (υπολογιστής) του εξομοιωτή είναι στο μπροστινό μέρος του τραπέζιου κοιτώντας προς τα παράθυρα, κρίθηκε σωστό να οριστεί το τελικό παγκόσμιο σύστημα συντεταγμένων στην κάτω αριστερά (κοιτώντας στην ίδια κατεύθυνση) γωνία της τράπεζας.

Συγκεκριμένα, οι συντεταγμένες xrobot, yrobot ως αποτέλεσμα του αλγορίθμου επεξεργασίας εικόνας δίνονται ως προς το σύστημα συντεταγμένων  $O_c$ , το οποίο φαίνεται στο Σχήμα 4.8. Φαίνεται επίσης ο τρόπος που υπολογίζεται η γωνία phiR από τον συγκεκριμένο αλγόριθμο. Οι γωνίες δίνονται στα διαστήματα  $[0,180]$  και  $[0,-180]$ . Ένας πιο χρηστικός τρόπος παρουσίασης των αποτελεσμάτων θα ήθελε ένα παγκόσμιο σύστημα συντεταγμένων  $O_G$  στην κάτω αριστερά γωνία της τράπεζας (κοιτώντας προς τα παράθυρα του εργαστηρίου), στο οποίο θα αναφέρονταν οι συντεταγμένες xrobot, yrobot, ενώ η γωνία phiR θα εκτεινόταν ιδανικά στο εύρος  $[0,360]$ . Για να επιτευχθεί αυτή η μετατροπή, οι παρακάτω μετασχηματισμοί στα αποτελέσματα του αλγορίθμου είναι αναγκαίοι.



**Σχήμα 4.8** Το Αρχικό παγκόσμιο Σύστημα συντεταγμένων και η γωνία του ρομπότ, όπως βρέθηκαν από πειράματα.

Παρατηρούμε στο Σχήμα 4.9 ότι τα συστήματα συντεταγμένων  $O_G$  και  $O_c$  έχουν αντίθετα τους άξονες z. Κάνοντας διάφορα πειράματα και λαμβάνοντας υπόψη, ότι οι διαστάσεις της τράπεζας του γρανίτη είναι  $1800 [mm] \times 2200 [mm]$ , διαπιστώθηκε ότι οι συντεταγμένες ως προς το παγκόσμιο σύστημα  $O_G$  ως εξής :

$$xRobot_G = (2200 - 480) - yRobot_c = 1720 - yRobot_c$$

$$yRobot_G = (1800 - 400) - xRobot_c = 1400 - xRobot_c$$

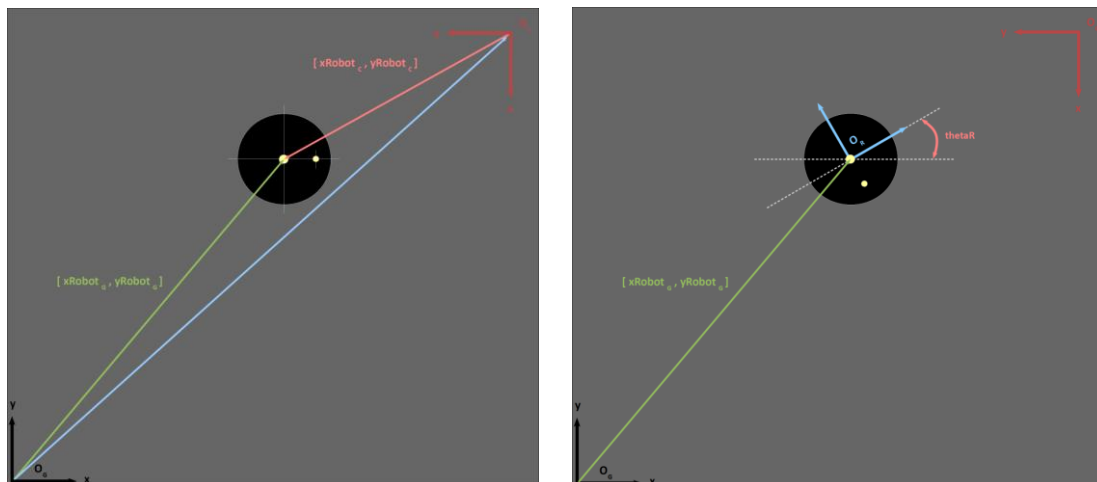
Όσον αφορά στη γωνία, ο κατάλληλος μετασχηματισμός για να επιτευχθεί ο,τι αναφέρθηκε παραπάνω, είναι ο εξής:

$$\text{Βήμα 1: } \mathbf{thetaR} = -\mathbf{phiR}$$

$$\text{Βήμα 2: Αν } \mathbf{thetaR} < 0, \text{ τότε } \mathbf{thetaR} = 360 + \mathbf{thetaR}$$

Ο τελικός κώδικας C++, που περιλαμβάνει όλα τα παραπάνω για τον εντοπισμό του δεύτερου ρομπότ, δίνεται στο CD που συνοδεύει την εργασία. Είναι εμπλουτισμένος με σχόλια, έτσι ώστε να

γίνεται εύκολα κατανοητός. Αναφορά σε αυτόν γίνεται επίσης στο Κεφάλαιο 6 που έχει να κάνει με τον προγραμματισμό του ρομπότ.

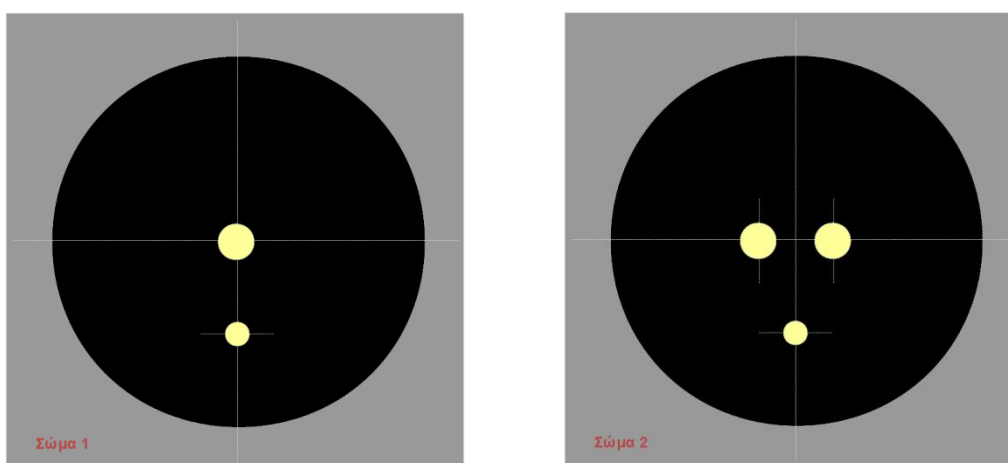


**Σχήμα 4.9** Αριστερά: Διανύσματα θέσης ως προς τα δύο συστήματα. Δεξιά: Διάνυσμα θέσης και γωνία στροφής ως προς το τελικό παγκόσμιο σύστημα.

#### 4.7 Ανάπτυξη τελικού προγράμματος για δύο ρομπότ

Μετά την ομαλή λειτουργία του αλγορίθμου για ένα ρομπότ, και εφόσον έγιναν αρκετά επιτυχή πειράματα (παρουσιάζονται στο Κεφάλαιο 8), συντάχθηκε πρόγραμμα C++, όμοιο με το υπάρχον, με το οποίο υλοποιείται για πρώτη φορά ο ταυτόχρονος εντοπισμός δύο κινούμενων ρομπότ στον χώρο εργασίας. Η ιδέα έχει ήδη περιγραφεί αναλυτικά παραπάνω, ενώ σε αυτό το στάδιο γίνεται μια πρώτη μεταφορά της σε γλώσσα C++.

Οι διατάξεις των LEDs στα δύο σώματα, είναι αυτές που παρουσιάζονται στο Σχήμα 4.10. Η οροφή του δεύτερου (νέου) ρομπότ διατηρεί την τοπολογία των LEDs που παρουσιάστηκε στην προηγούμενη ενότητα, ενώ στο άλλο σώμα τοποθετούνται με βάση την ίδια λογική στον απλούστερο σχηματισμό, σύμφωνα πάντα με την αρχική ιδέα.



**Σχήμα 4.10** Αριστερά: LEDs 1<sup>ου</sup> σώματος. Δεξιά: LEDs 2<sup>ου</sup> σώματος.

### Δομή Προγράμματος

1. Εύρεση των 5 LEDs σε τυχαία σειρά. Αυτό αντιστοιχεί στην εύρεση των συντεταγμένων των κέντρων τους και του μεγέθους τους σε pixels. Για κάθε LED αποθηκεύονται οι μεταβλητές: `xled`, `yled`, `ledpixels`.
2. Τα στοιχεία αυτά τοποθετούνται σε έναν δισδιάστατο πίνακα 5x3 (κάθε σειρά αντιστοιχεί σε ένα LED)
3. Ο πίνακας αυτός ταξινομείται, με την μέθοδο φυσαλίδας, ως προς το μέγεθος των LEDs (στήλη `ledpixels`), μέχρι τα στοιχεία των 3 φωτεινότερων LEDs να καταλάβουν τις 3 πρώτες σειρές.
4. Συγκρίνοντας τις απαραίτητες αποστάσεις (περιγράφεται σε προηγούμενη ενότητα), οι πληροφορίες χωρίζονται σε 2 δισδιάστατους πίνακες, οι οποίοι αντιστοιχούν στα LEDs των 2 σωμάτων.
5. Σε αυτό το βήμα γίνονται, όπως συνήθως, οι υπολογισμοί των θέσεων και των γωνιών των δυο σωμάτων.
6. Τα δεδομένα αποστέλλονται τελικά στα υπολογιστικά συστήματα των δύο ρομπότ.

Αυτή τη στιγμή, το πρόγραμμα αυτό είναι ακόμη σε φάση εύρεσης λαθών (*debugging*), αλλά η μορφή του δεν διαφέρει πολύ από την τελική, δεδομένου ότι οι προσομοιώσεις έχουν τρέξει με επιτυχία. Έχουν γίνει πειράματα, και ήδη διορθώνονται λάθη και ρυθμίζονται οι παράμετροι επεξεργασίας της εικόνας. Λόγω έλλειψης χρόνου, παραδίδεται ημιτελής στο CD της εργασίας.

## 4.8 Αποστολή Δεδομένων στο υπολογιστικό Σύστημα του ρομπότ

Στα συστήματα ευφυούς ελέγχου, όπου ζητούμενο είναι ο ελάχιστος χρόνος δειγματοληψίας, όταν πρόκειται για ανταλλαγή δεδομένων μέσω δικτύου, οι επιλογές επικοινωνίας είναι οι εξής: TCP/IP, UDP/IP, Raw Ethernet. Σχεδόν πάντα, προτιμάται η χρήση των πρωτοκόλλων UDP/IP ή Raw Ethernet λόγω της ταχύτητάς τους, ενώ το TCP/IP θεωρείται αργό, αν και πιο αξιόπιστο, και δεν χρησιμοποιείται σε πολλές εφαρμογές.

Εντούτοις, ο υπάρχων κώδικας, γραμμένος σε γλώσσα C++, χρησιμοποιεί το πρωτόκολλο TCP/IP για την ασύρματη αποστολή των δεδομένων θέσης και προσανατολισμού στο πρώτο (παλιό) ρομπότ. Αυτό έχει γίνει κυρίως για λόγους γρήγορης ανάπτυξης από την ομάδα, αφού μεγάλο μέρος του κώδικα επικοινωνίας TCP/IP ήταν ήδη έτοιμο όταν προγραμματιζόταν η επικοινωνία αυτή.

Στην εργασία αυτή, όλες οι επιλογές πρέπει, εκτός των άλλων, να είναι συμβατές με το xPC Target. Στο σημείο αυτό, τίθεται άλλος ένας περιορισμός, αφού η επικοινωνία TCP/IP δεν υποστηρίζεται για *real-time* ανταλλαγή δεδομένων. Οι δυσκολίες γύρω από αυτό το θέμα, περιγράφονται αναλυτικά στο κεφάλαιο του Προγραμματισμού του ρομπότ. Στο κεφάλαιο αυτό, αναλύονται τα κύρια χαρακτηριστικά των δύο πρωτοκόλλων επικοινωνίας, με σκοπό να φανούν πλεονεκτήματα και μειονεκτήματα αυτών, σύμφωνα πάντα με τις απαιτήσεις του συστήματος που εξετάζεται. Η μεταφορά δεδομένων με Raw Ethernet, αν και αρκετά γρήγορη, δεν εξετάζεται στην παρούσα εργασία λόγω της αυξημένης πολυπλοκότητάς της, σε σύγκριση με το πρωτόκολλο UDP/IP.

### 4.8.1 Πρωτόκολλο TCP/IP

Το TCP (Transmission Control Protocol) είναι ένα πρωτόκολλο βασισμένο στη σύνδεση, το οποίο προσφέρει διόρθωση σφαλμάτων και εγγυημένη παράδοση δεδομένων μέσω αυτού που είναι γνωστό ως έλεγχος ροής. Ο έλεγχος ροής καθορίζει τότε η ροή ενός ρεύματος δεδομένων πρέπει να σταματήσει, και τότε πακέτα δεδομένων που έχουν σταλεί, πρέπει να ξανασταλούν διασφαλίζοντας έτσι πλήρη και ακριβή παράδοση δεδομένων. Χρησιμοποιείται όπου η ακεραιότητα των δεδομένων είναι ύψιστης σημασίας, όπως στις συναλλαγές βάσης δεδομένων. Είναι, επίσης, το πρωτόκολλο που χρησιμοποιείται κατεξοχήν στον Internet. Το TCP δημιουργεί μια σειρά από μηνύματα (*segments*), τα οποία φαίνονται σαν μια συνεχής ροή δεδομένων. Αυτή η ροή είναι 2 κατευθύνσεων (αποστολέας-παραλήπτης) και ταυτόχρονα πολύ αξιόπιστη. Τα κυριότερα χαρακτηριστικά του TCP:

- **Είναι connection protocol**

Χρησιμοποιείται μόνο μεταξύ 2 υπολογιστών. Πριν ξεκινήσει η μεταφορά δεδομένων πρέπει να γίνει ένα τριπλό handshaking μεταξύ των 2 υπολογιστών (connection establishment) και το αντίστοιχο στον τερματισμό της αποστολής (connection termination).

- **Είναι αξιόπιστο**

Το TCP του παραλήπτη ενημερώνει συνεχώς το TCP του αποστολέα, για το πιο είναι το επόμενο πακέτο που περιμένει, σύμφωνα με τον αύξοντα αριθμό των πακέτων που έχει ήδη λάβει. Αν αντιληφθεί ότι κάποιο πακέτο χάθηκε στην πορεία, τότε το ξαναστέλνει (retransmission). Αν το πακέτο δεν μπορεί να έρθει μετά από πολλαπλά retransmissions, τότε η σύνδεση διακόπτεται (timeout).

- **Σωστή σειρά άφιξης**

Εγγυάται την σωστή σειρά άφιξης των δεδομένων στην εφαρμογή του παραλήπτη. Όταν τα δεδομένα ληφθούν από τον παραλήπτη με λάθος σειρά, τότε το TCP layer “κρατάει” αυτά τα δεδομένα μέχρι να έρθουν τα προηγούμενα τους. Αφού έρθουν τα διατάσσει στην σωστή σειρά και έπειτα τα παραδίδει στην εφαρμογή.

- **Απόρριψη διπλών δεδομένων**

Αποτρέπει την αποστολή διπλότυπων, δηλαδή δύο ακριβώς ίδιων δεδομένων.

- **Προσφέρει αυτοματοποιημένο έλεγχο ροής δεδομένων (flow control)**

Όταν ο buffer του παραλήπτη γεμίσει, τότε σταματάει προσωρινά το transmission ή ελαττώνει τον ρυθμό μετάδοσης του, μέχρι να αδειάσει ο buffer. Μια απαραίτητη λειτουργία σε ένα κόσμο όπου επικοινωνούν μηχανές διαφορετικών ταχυτήτων κάτω από διαφορετικά δίκτυα.

- **Προσφέρει αυτοματοποιημένο έλεγχο συμφόρησης (congestion control)**

Το TCP χρησιμοποιεί μια πληθώρα μηχανισμών για να επιτύχει την μέγιστη απόδοση μεταφοράς δεδομένων αποφεύγοντας την συμφόρηση δεδομένων στους routers του Internet, μια κατάσταση κατά την οποία πέφτει η απόδοση του δικτύου κατά μεγάλο βαθμό. Αυτοί οι μηχανισμοί ελέγχουν τον ρυθμό με τον οποίο τα δεδομένα μπαίνουν στο δίκτυο, κρατώντας αυτό το ρυθμό κάτω από ένα ασφαλές όριο.

- **Εγγυάται την ακεραιότητα του "μονοπατιού επικοινωνίας"**

Για παράδειγμα, η υποκλοπή δεδομένων από τρίτους είναι δύσκολη, μιας και ο κακόβουλος χρήστης θα πρέπει να ακούσει όλη την ροή δεδομένων, καθώς δεν υπάρχουν συγκεκριμένου μεγέθους πακέτα.

## 4.8.2 Πρωτόκολλο UDP/IP

Το Πρωτόκολλο Διαγραμμάτων Δεδομένων Χρήστη UDP (User Datagram Protocol είναι ένα πρωτόκολλο χωρίς συνδέσεις το οποίο σπάνια ασχολείται με τη μεταφορά σημαντικών δεδομένων επειδή δεν έχει έλεγχο ροής ή οποιαδήποτε άλλη μέθοδο για να διασφαλίσει την αξιόπιστη μεταφορά δεδομένων. Το UDP χρησιμοποιείται συνήθως σε αναπαραγωγή ήχου και βίντεο, όπου θεωρείτε πιο γρήγορο από το TCP λόγω της έλλειψης διόρθωσης σφαλμάτων και ελέγχου ροής, και όπου η απώλεια κάποιων πακέτων δεν είναι γενικά καταστροφική. Χρησιμοποιώντας το UDP τα προγράμματα μπορούν να στείλουν μικρά μηνύματα, γνωστά ως datagrams, το ένα στο άλλο. Τα κυριότερα χαρακτηριστικά του UDP είναι:

- **Είναι αναξιόπιστο (best effort delivery)**

Δε μπορεί να εγγυηθεί την ακεραιότητα ή τη σωστή σειρά άφιξης των δεδομένων, όπως το TCP. Τα πακέτα (datagrams) μπορούν να φτάσουν με διαφορετική σειρά, να εμφανίζονται διπλά ή να μην έρθουν και καθόλου χωρίς καμία ειδοποίηση.

- **Είναι γρήγορο**

Το παραπάνω χαρακτηριστικό του εξασφαλίζει μικρό delay και κάνει το UDP πιο γρήγορο από το TCP.

- **Πολλαπλή χρηστικότητα**

Μπορεί να χρησιμοποιηθεί τόσο σε Unicast όσο και σε Multicast δίκτυα, καθώς δεν είναι connection protocol.

- **Είναι ‘ελαφρύ’**

Λιγότερο απαιτητικό σε πόρους, σε σχέση με το TCP. Δεν δημιουργεί μεγάλο overhead στο δίκτυο, καθώς δεν ελέγχει αν όντως κάποιο πακέτο έφτασε ή όχι.

- **Έχει μικρότερο header**

Το UDP έχει 8 bytes header, σε σχέση με το TCP που έχει 20 bytes header. Αυτό σημαίνει μικρότερο έξτρα overhead στο δίκτυο.

### Εφαρμογές UDP

Το UDP υπάρχει ακριβώς γιατί υπάρχουν εφαρμογές, όπου δεν ενδιαφέρει τόσο η ακεραιότητα των δεδομένων, όσο τα δεδομένα να φτάσουν όσο δυνατόν γρηγορότερα στον παραλήπτη, έστω και με κάποια απώλεια. Μερικές εφαρμογές που χρησιμοποιούν το UDP είναι οι παρακάτω:

- Εφαρμογές οι οποίες μεταδίδουν real-time audio/video, όπως IPTV, VoIP. Εδώ ενδιαφέρει τα δεδομένα να φτάνουν την σωστή χρονική στιγμή. Οποιαδήποτε απώλεια τους επηρεάζει μόνο την ποιότητα του αναπαραγόμενου σήματος.
- Servers, οι οποίοι απαντάνε σε μικρά αιτήματα ενός τεράστιου αριθμού από clients, όπως στα online παιχνίδια. Οι Servers δεν απασχολούνται με το να ελέγχουν την κατάσταση του κάθε connection και των παραμέτρων του χρησιμοποιώντας UDP, και έτσι μπορούν να εξυπηρετήσουν ένα πολύ μεγαλύτερο αριθμό χρηστών σε αντίθεση με το αν χρησιμοποιούσαν TCP.
- Κάποιες επίσης πολύ σημαντικές εφαρμογές είναι τα: Domain Name System (DNS), Simple Network Management Protocol (SNMP), Dynamic Host Configuration Protocol (DHCP), Routing Information Protocol (RIP).

### 4.8.3 Συμπεράσματα

Μετά από όλα αυτά, είναι φανερό ότι ένα πρόγραμμα που χρησιμοποιεί το πρωτόκολλο UDP πρέπει να ασχοληθεί το ίδιο με τα προβλήματα επικοινωνίας που μπορεί να προκύψουν: την αξιόπιστη παράδοση, το packetization και την επανασυναρμολόγηση, τον έλεγχο ροής, την αποφυγή συμφόρησης, κλπ. Επίσης, δεδομένου ότι το UDP στερείται μηχανισμών αποφυγής και ελέγχου δικτυακής συμφόρησης, απαιτούνται network-based μηχανισμοί για να ελαχιστοποιηθούν τα πιθανά προβλήματα κατάρρευσης δικτύου λόγω ανεξέλεγκτα υψηλών ρυθμών αποστολής πακέτων UDP. Με άλλα λόγια, δεδομένου ότι οι αποστολές UDP δεν μπορούν να ανιχνεύσουν τη συμφόρηση, τα στοιχεία των δικτύων, όπως οι routers, πρέπει να χρησιμοποιούν τεχνικές packet queuing και απόρριψης πακέτων για να ελέγχουν την υπερβολική κίνηση πακέτων UDP στα δίκτυα.

UDP & TCP Comparison Table	UDP	TCP
<b>Packet header size</b>	8 Bytes	20 Bytes
<b>Transport layer packet entity</b>	Datagram	Segment
<b>Port numbering</b>	Yes	Yes
<b>Error detection</b>	Optional	Yes
<b>Reliability: Error recovery by automatic repeat request (ARQ)</b>	No	Yes
<b>Virtual circuits: Sequence numbering and reordering</b>	No	Yes
<b>Flow control</b>	No	Yes
<b>Congestion avoidance: Variable congestion window, slow start, time outs</b>	No	Yes
<b>Multiple streams</b>	No	No
<b>ECN support</b>	No	Yes

*Πίνακας 4.1 Σύγκριση UDP/IP και TCP/IP.*

Σύμφωνα με την παραπάνω ανάλυση των πρωτοκόλλων γίνεται πλέον κατανοητή η υπεροχή του UDP/IP έναντι του TCP/IP στα ενσωματωμένα συστήματα. Παρόλα αυτά, με γνώμονα τη γρήγορη ανάπτυξη του συστήματος και την εστίαση στους αλγορίθμους επεξεργασίας, ήταν αρχικά επιθυμητή η αυτούσια χρήση του υπάρχοντος αλγορίθμου με το πρωτόκολλο TCP/IP. Αυτό ήταν, όμως, αδύνατο στο πλαίσιο της ανάπτυξης του συστήματος με το πακέτο xPC Target, το οποίο δεν υποστηρίζει το πρωτόκολλο TCP/IP για τους ίδιους βέβαια λόγους με αυτούς που αναφέρθηκαν. Ήταν λοιπόν τελικά μονόδρομος, η αλλαγή του πρωτοκόλλου επικοινωνίας που χρησιμοποιούσε ο

υπάρχων κώδικας C++. Οι αλλαγές που έγιναν στο πρόγραμμα περιγράφονται αναλυτικά στο κεφάλαιο του Προγραμματισμού.

# ΚΕΦΑΛΑΙΟ 5

## Εύρεση θέσης και προσανατολισμού από σύστημα οπτικών αισθητήρων

### 5.1 Εισαγωγή

Παρόλο που το σύστημα αυτό έχει μελετηθεί, σχεδιαστεί και αναπτυχθεί ξανά στο παρελθόν στις [4][16][17], και που η απαιτούμενη ομοιότητά του με αυτό του πρώτου ρομπότ δεν αφήνει μεγάλα περιθώρια για επιλογή εναλλακτικών αισθητήρων, κρίνεται σκόπιμο να αναφερθούν συνοπτικά οι κύριοι λόγοι επιλογής τέτοιων αισθητήρων στα ρομπότ του διαστημικού εξομοιωτή.

Αρχικά, πρέπει να αναφερθεί ότι, παρόλο που ήδη στο προηγούμενο κεφάλαιο, αναπτύχθηκε ένα υποσύστημα εντοπισμού θέσης/ προσανατολισμού του ρομπότ, είναι χρήσιμο και αυξάνει την συνολική αξιοπιστία να ενσωματωθεί ένα επιπλέον υποσύστημα εντοπισμού, με χαρακτηριστικά συμπληρωματικά ως προς το πρώτο. Το σύστημα της κάμερας είναι μεν απόλυτο και ακριβές αλλά πρώτον, παρέχει δεδομένα σε μικρή συχνότητα (δύσκολα κλείνει βρόχος ελέγχου) και δεύτερον, είναι σύστημα εξωτερικό του ρομπότ και προκαλούνται καθυστερήσεις στη μεταφορά δεδομένων.

Υπάρχει λοιπόν ανάγκη για ένα γενικότερο σύστημα εύρεσης θέσης/ προσανατολισμού, το οποίο θα αποτελείται από δύο μετρητικά συστήματα: την εξωτερική κάμερα χαμηλής συχνότητας, και κάποιους αισθητήρες υψηλής συχνότητας δειγματοληψίας που θα εδράζονται στο σώμα του ρομπότ. Ο κατάλληλος συνδυασμός των δύο αυτών τρόπων εύρεσης θέσης, μπορεί να οδηγήσει αλγοριθμικά σε εξάλειψη μεμονωμένων σφαλμάτων και σε αξιόπιστα τελικά αποτελέσματα στη διαδικασία του ελέγχου του ρομπότ, όπως περιγράφεται αναλυτικά στην [12].

## Αισθητήρες

Υπάρχουν πάρα πολλά είδη αισθητήρων που χρησιμοποιούνται σε παρόμοιες εφαρμογές εντοπισμού θέσης (localization) στη ρομποτική. Αισθητήρες υπερήχων, αισθητήρες υπέρυθρης ακτινοβολίας, αισθητήρες laser, επιταχυνσιόμετρα (INS), αισθητήρες ραδιοκυμάτων κ.α. χρησιμοποιούνται εκτενώς για τέτοιους σκοπούς ανάλογα πάντα με τις απαιτήσεις της κάθε εφαρμογής. Όπως και στο πρώτο ρομπότ του διαστημικού εξομοιωτή, όπου έγινε εκτενής μελέτη πάνω στο θέμα, έτσι κι εδώ θα χρησιμοποιηθούν οπτικοί αισθητήρες για τον εντοπισμό του ρομπότ.

Οι οπτικοί αισθητήρες, γνωστοί κυρίως από την εφαρμογή τους στα ποντίκια υπολογιστών, χρησιμοποιούν συνήθως μια μικρή κάμερα για λήψη διαδοχικών εικόνων και εξαγωγή δεδομένων κίνησης μέσω της γρήγορης επεξεργασίας τους. Ο τρόπος λειτουργίας τους θα περιγραφεί αναλυτικά παρακάτω και θα παρουσιαστούν επίσης τα βασικά χαρακτηριστικά των διαφόρων ειδών οπτικών αισθητήρων πριν την τελική επιλογή. Σημειώνεται ότι, η χρήση οπτικών αισθητήρων για αναγνώριση κίνησης δεν είναι κάτι καινούριο. Χρησιμοποιούνται για την ανίχνευση μετατόπισης ή κατεύθυνσης και άλλες φορές για μέτρηση της διανυσθείσας απόστασης. Παρόλα αυτά, στην παγκόσμια βιβλιογραφία, είναι δύσκολο να βρει κανείς ένα πλήρες σύστημα εντοπισμού θέσης και προσανατολισμού στο επίπεδο.

Παρόλο που υπάρχουν οπτικοί αισθητήρες ειδικό για εφαρμογές σε ενσωματωμένα συστήματα (embedded systems), δεν παρουσιάζουν μεγάλη ποικιλία ενώ η αγορά τους μπορεί να γίνει μόνο σε εξειδικευμένα καταστήματα. Για τους λόγους αυτούς, και δεδομένου ότι οι τεχνολογίες αιχμής χρησιμοποιούνται κατευθείαν στα ποντίκια υπολογιστών, η μελέτη εδώ επικεντρώνεται στους διάφορους τύπους ποντικίων που κυκλοφορούν. Οι συσκευές αυτές μπορούν εύκολα να αγοραστούν από πλήθος καταστημάτων και να ενσωματωθούν άμεσα σε ένα ρομποτικό σύστημα.

## 5.2 Τύποι ποντικίων Η/Υ

Το πρώτο ποντίκι τέτοιου τύπου που εμφανίστηκε ήταν το μηχανικό, το οποίο δύσκολα απαντάται στις μέρες μας. Τα μηχανικά ποντίκια φέρουν στο κάτω μέρος τους μια σφαίρα η οποία, επαφτόμενη στο επίπεδο κίνησης μπορεί να κυλά ελεύθερα. Η κίνηση αυτή ανιχνεύεται μέσω ενός συστήματος LED – φωτοδιόδου και ύστερα μεταφράζεται σε μετατόπιση. Ένα τέτοιο σύστημα απαιτεί επαφή με το επίπεδο κίνησης, άρα δεν μπορεί να χρησιμοποιηθεί στο παρόν ρομποτικό σύστημα (απαιτείται μηδενική τριβή με το επίπεδο κίνησης) και δεν εξετάζεται περαιτέρω.

Τα είδη ποντικίων που χρησιμοποιούνται περισσότερο στις μέρες μας είναι τα ποντίκια laser αλλά και τα λίγο παλιότερα οπτικά ποντίκια.

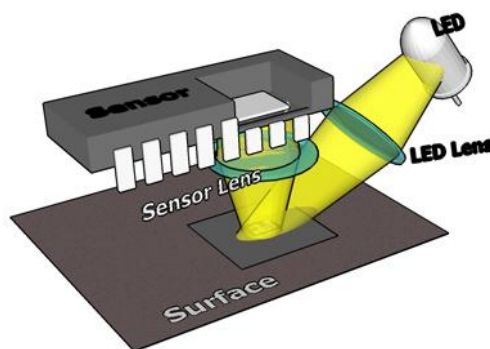
### 5.2.1 Οπτικά Ποντίκια

Τα ποντίκια αυτά χρησιμοποιούν έναν οπτοηλεκτρονικό αισθητήρα (ουσιαστικά, μια μικροσκοπική κάμερα χαμηλής ανάλυσης) για να τραβήξουν διαδοχικές εικόνες της επιφάνειας πάνω στην οποία λειτουργούν. Η εξέλιξη που γνώρισαν τα ολοκληρωμένα κυκλώματα οδήγησε στην ενσωμάτωση όλο και πιο ισχυρών μικροεπεξεργαστών επεξεργασίας εικόνας στα ποντίκια αυτά, δίνοντας έτσι την δυνατότητα λειτουργίας σε μεγάλο εύρος επιφανειών και καταργώντας την ανάγκη χρήσης ειδικών επιφανειών (mouse pads).



Ο πρώτη εμφάνιση τέτοιου ποντικιού έγινε το 1999 με το Microsoft Intellimouse, όπου χρησιμοποιήθηκε τεχνολογία ανεπτυγμένη από την Hewlett-Packard. Λειτουργούσε σχεδόν σε όλες τις επιφάνειες και αποτέλεσε μια πρώτη σημαντική βελτίωση σε σχέση με τα μηχανικά ποντίκια. Άλλοι κατασκευαστές γρήγορα ακολούθησαν την κίνηση της Microsoft χρησιμοποιώντας εξαρτήματα κατασκευασμένα από μια spin-off εταιρεία της HP, την Agilent Technologies, και σε μερικά χρόνια τα μηχανικά ποντίκια αντικαταστάθηκαν πλήρως από τα οπτικά.

Η τεχνολογία που χρησιμοποιούν τα σύγχρονα οπτικά ποντίκια είναι γνωστή ως συσχέτιση ψηφιακής εικόνας (digital image correlation), η οποία πρώτα αναπτύχθηκε από την αμυντική βιομηχανία για εντοπισμό στρατιωτικών στόχων. Χρησιμοποιούνται αισθητήρες εικόνας για την απεικόνιση της υφής υλικών όπως ξύλο, ύφασμα, μέταλλο κ.α. Όταν οι επιφάνειες αυτές φωτιστούν υπό κατάλληλη γωνία από μία δίοδο εκπομπής φωτός LED, δημιουργούνται χαρακτηριστικές σκιάσεις (βλ. Σχήμα 5.1). Διαδοχικές εικόνες αυτών συγκρίνονται μεταξύ τους και οδηγούν στον υπολογισμό της διανυθείσας από το ποντίκι απόστασης, με την μέθοδο της οπτικής ροής (optical flow) [18]. Το χρώμα των διόδων φωτοεκπομπής ποικίλει. Αυτό που χρησιμοποιείται περισσότερο είναι το κόκκινο, γιατί πρώτον η αντίστοιχη δίοδος είναι πιο οικονομική και δεύτερον γιατί οι ανιχνευτές φωτός που χρησιμοποιούνται είναι πιο ευαίσθητοι στο κόκκινο φως. Ένα άλλο χρώμα που απαντάται μερικές φορές σε οπτικά ποντίκια είναι το μπλε.



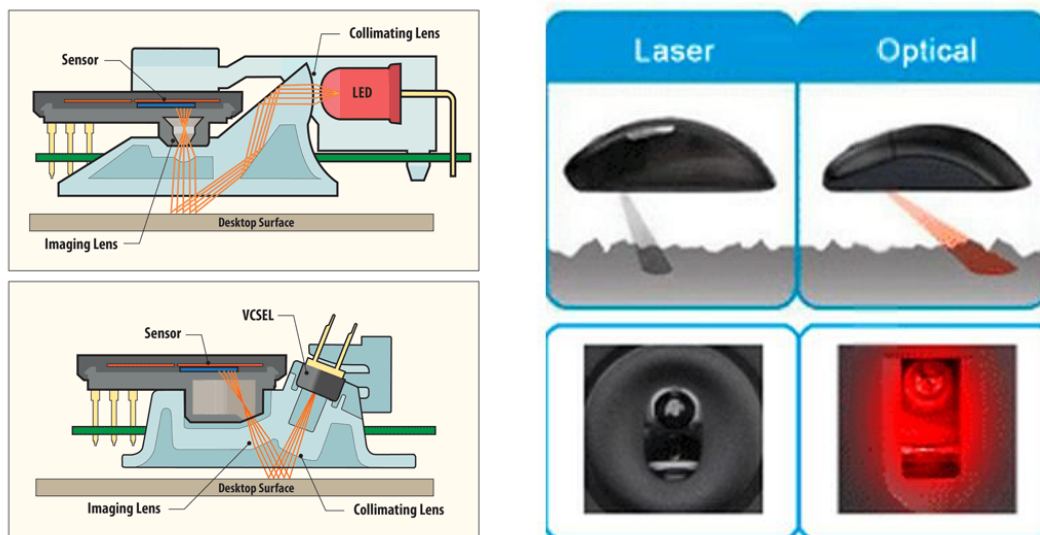
**Σχήμα 5.1** Τρόπος λειτουργίας των οπτικών ποντικιών.

Από έναν οπτικό αισθητήρα συλλέγονται 1000 ή περισσότερες διαδοχικές εικόνες το δευτερόλεπτο, καθεμία από τις οποίες αναλύεται σε ένα μητρώο μονόχρωμων pixels της τάξης του 18x18. Κάθε εικόνα είναι μετατοπισμένη ως προς την προηγούμενη ή επόμενη της κατά έναν αριθμό pixels, ανάλογα με την ταχύτητα κίνησης. Η συλλογή, η αποθήκευση και η επεξεργασία των εικόνων γίνονται από το ίδιο “ειδικής εφαρμογής ολοκληρωμένο κύκλωμα” (ASIC ή Application-Specific Integrated Circuit).

### 5.2.2 Ποντίκια laser

Τα ποντίκια laser χρησιμοποιούν μία δίοδο εκπομπής υπέρυθρου φωτός αντί της δίοδου φωτοεκπομπής LED, για να φωτίσουν την επιφάνεια κάτω από τον αισθητήρα (βλ. Σχήμα 5.2). Αν και άορατο στο ανθρώπινο μάτι, το υπέρυθρο φως που παράγεται συλλέγεται κανονικά από τον αισθητήρα CCD (charge-coupled device) ή CMOS (Complementary metal-oxide-semiconductor), ο οποίος είναι ευαίσθητος σε μεγαλύτερο εύρος ακτινοβολίας. Για πρώτη φορά το 1998, η Sun Microsystems παρουσίασε την τεχνολογία laser. Δεν ήταν όμως νωρίτερα από το 2004, όταν η Logitech σε συνεργασία με την Agilent Technologies, κυκλοφόρησε το MX 1000 laser ποντίκι.

Η τεχνολογία laser αύξησε την ανάλυση και την ακρίβεια των οπτικών αισθητήρων που τη χρησιμοποιούν. Τα ποντίκια laser επιτρέπουν υψηλότερα επίπεδα λεπτομέρειας στην ανίχνευση επιφανειών σε σχέση με τα οπτικά. Σαν αποτέλεσμα, το εύρος επιφανειών στις οποίες μπορούν να χρησιμοποιηθούν αυξάνεται, αλλά προβλήματα εξακολουθούν να υπάρχουν όσον αφορά τις διαφανείς και ανακλαστικές επιφάνειες.



Σχήμα 5.2 Παρουσίαση των τεχνολογιών optical και laser.

### 5.2.3 Καινούριες Τεχνολογίες

Η αυξανόμενη δημοτικότητα των laptop και η μεγάλη άνοδος στις πωλήσεις τους τα τελευταία χρόνια άλλαξαν το πλαίσιο χρήσης του τυπικού ποντικιού. Η ανάγκη συνοδείας των laptop από ποντίκι ικανό να λειτουργήσει πάνω σε οποιαδήποτε επιφάνεια, αποδείχθηκε μεγάλη. Έτσι, το 2008, η Microsoft παρουσίασε την πλέον προηγμένη τεχνολογία για ποντίκια στον κόσμο, γνωστή ως **Blue Track Technology** (βλ. Σχήμα 5.3). Η τεχνολογία αυτή συνδυάζει το πλάτος και τη δύναμη της οπτικής τεχνολογίας με την ακρίβεια του laser, και επιτρέπει τη χρήση σε οποιαδήποτε επιφάνεια εκτός από καθαρό γυαλί.

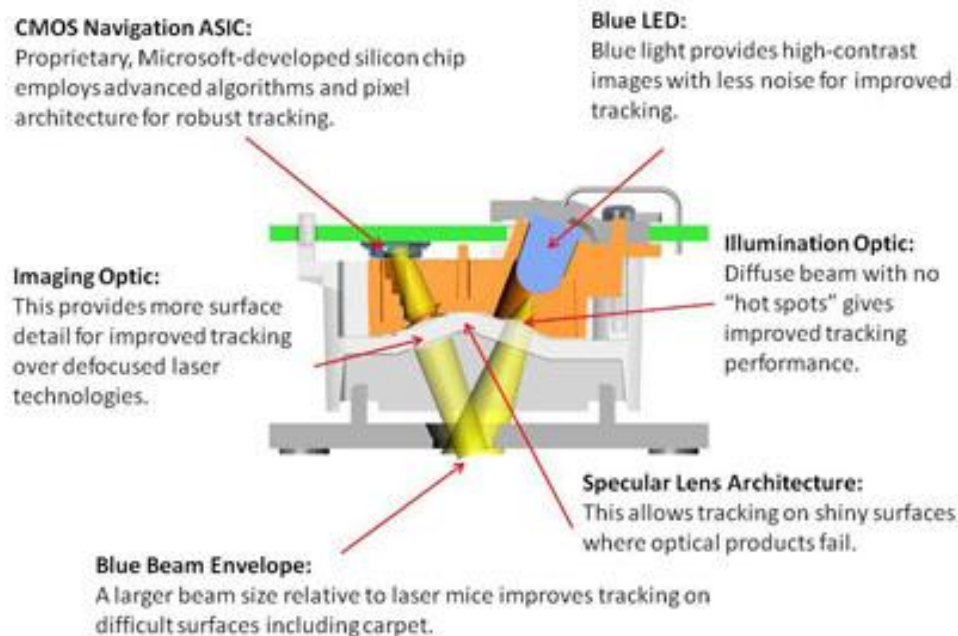
Η τεχνολογία BlueTrack της Microsoft δουλεύει σε περισσότερες επιφάνειες και από τα οπτικά και από τα laser ποντίκια. Η μεγάλη μπλε ακτίνα και η κατοπτρική οπτική τεχνολογία, σε συνδυασμό με έναν ειδικά σχεδιασμένο – αισθητήρα εικόνας από τη Microsoft, παράγουν μια εικόνα υψηλής αντίθεσης της επιφάνειας που επιτρέπει την εξαιρετική ακρίβεια στην ανίχνευση της κίνησης του ποντικιού. Η μπλε ακτίνα του ποντικιού BlueTrack είναι τέσσερις φορές μεγαλύτερη από τη μέση ακτίνα λέιζερ που χρησιμοποιείται στα σημερινά ποντίκια, επιτρέποντας τη σύλληψη μιας μεγαλύτερης εικόνας και προσφέροντας καλύτερη αντανάκλαση στην επιφάνεια συγκριτικά με την κόκκινη ακτίνα και το laser. Η τεχνολογία BlueTrack έκανε το ντεμπούτο της με το κανονικού μεγέθους Explorer Mouse και το μικρότερο Explorer Mini Mouse.

Αλλά και σ' εσωτερικούς χώρους, ακόμη και μετά την έλευση του BlueTrack της Microsoft, υπήρχαν επιφάνειες όπου ένα mouse υψηλών προδιαγραφών για Η/Υ γραφείου δεν μπορούσε να λειτουργήσει σωστά ή καθόλου, όπως οι γυάλινες. Σχετική έρευνα της Logitech απέδειξε ότι στο 40% των σύγχρονων οικιακών ή επαγγελματικών χώρων υπάρχουν γραφεία ή τραπέζια με τέτοιες επιφάνειες. Στόχος, λοιπόν, ήταν μία τεχνολογία σάρωσης που δεν θα αντιμετώπιζε κανένα πρόβλημα, πουθενά.

Η ελβετική εταιρεία έθεσε το πρόβλημα στη ρεαλιστικότερη βάση του: αν κατάφερνε να αναπτύξει μια τεχνολογία που θα σάρωνε σωστά σε γυάλινη επιφάνεια, θα ήταν σε θέση να κατασκευάσει ένα ποντίκι που λειτουργεί αληθινά παντού. Το γυαλί είναι ιδιότυπο υλικό, διάφανο (οπότε η ακτίνα laser του αισθητήρα δεν αντανακλάται, αλλά περνά από μέσα του) και απόλυτα λείο μετά την κατεργασία του για χρήση σε οικιακά ή επαγγελματικά έπιπλα. Αρχικά, φαινόταν αδύνατο να κατευθυνθεί φως πάνω σε γυάλινη επιφάνεια, κι αυτό να "γυρίσει" στη σωστή γωνία, ώστε να αναφέρει στον αισθητήρα ακριβή θέση για τη σάρωση. Η λύση της Logitech ήταν η ανάπτυξη μιας νέας τεχνολογίας, γνωστής ως **Darkfield**.

Η λύση βασίστηκε σε τεχνική που αξιοποιούν τα επαγγελματικά μικροσκόπια εργαστηρίων σε όλο τον κόσμο, με τρόπο που φυσικά προσαρμόστηκε στις ανάγκες των ποντικιών. Η τεχνική ονομάζεται "dark field" (σκοτεινό πεδίο) και λειτουργεί ως εξής: όταν το αντικείμενο προς μελέτη - το οποίο

συνήθως τοποθετείται κάτω από το μικροσκόπιο ανάμεσα σε δύο επιφάνειες από γυαλί - είναι ασυνήθιστα μικρό ή δεν ξεχωρίζει αρκετά από το γύρω υλικό του, το μικροσκόπιο δεν προσπαθεί να φωτίσει το ίδιο το αντικείμενο και από την αντανάκλαση να δώσει την οπτική αναπαράσταση που θα μελετηθεί. Αντιθέτως, φωτίζει υπό γωνία την επιφάνεια του γυαλιού και όχι κάθετα, με αποτέλεσμα ο παρατηρητής του μικροσκοπίου να βλέπει μία μαύρη επιφάνεια, πάνω στην οποία όμως ξεχωρίζει πεντακάθαρα το αντικείμενο προς μελέτη.



**Σχήμα 5.3** Παρουσίαση της τεχνολογίας Blue Track.

Ο μηχανισμός του ποντικιού Darkfield έχει σχεδιαστεί έτσι, ώστε να ανιχνεύει τις απίθανα μικρές, αόρατες με γυμνό μάτι ατέλειες που βρίσκονται ακόμη και στις γυάλινες επιφάνειες. Αυτές ακριβώς τις ατέλειες, στη συνέχεια, είτε πρόκειται για ανεπαίσθητες γρατσουνιές, είτε για δακτυλικά αποτυπώματα, είτε για απλούς κόκκους σκόνης, ο αισθητήρας τις χρησιμοποιεί ως σημεία αναφοράς ώστε να σαρώσει σωστά την επιφάνεια.

Δεδομένου ότι στον πραγματικό κόσμο, δεν υφίσταται επιφάνεια 100% απαλλαγμένη από οποιουδήποτε είδους σωματίδιο, η τεχνολογία αυτή βασίζεται στις ατέλειες, στις γρατσουνιές αλλά και στη σκόνη που ακόμη και μια γυάλινη επιφάνεια φέρει. Ο μηχανισμός της Logitech συνδυάζει δύο laser: το ένα είναι το καλύτερο συμβατικό, το οποίο επαρκεί για επιφάνειες από ξύλο, μέταλλο, πέτρα, ύφασμα, πλαστικό, οπουδήποτε υπάρχουν ούτως ή άλλως αρκετές λεπτομέρειες για να σαρωθούν. Το άλλο laser είναι το ειδικό του Darkfield, το οποίο μπορεί να ανιχνεύσει σωματίδια πάχους 5 [μm] (η ανθρώπινη τρίχα έχει πάχος 100 [μm]) και να τα αξιοποιήσει ως σημεία ορισμού όπως περιγράφηκε. Το δεύτερο laser ενεργοποιείται όταν ο αισθητήρας εντοπίσει γυαλί ή άλλη εξαιρετικά λεία επιφάνεια - σε άλλη περίπτωση, απλώς παραμένει σε αναμονή. Τα δύο πρώτα μοντέλα αυτής της τεχνολογίας ήταν τα Performance Mouse MX και Anywhere Mouse MX.

### Ποντίκια Glaser

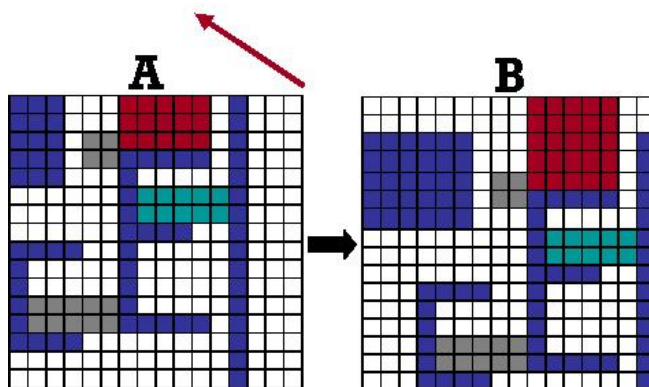
Μια άλλη τεχνολογία (Glaser) που αναπτύχθηκε από Κορεάτες, έδωσε επίσης τη δυνατότητα λειτουργίας σε γυάλινες και άλλες πολύ λείες επιφάνειες. Ο καινοτόμος διπλής εστίασης φακός σχεδιάστηκε έτσι, ώστε να λειτουργεί και σε διαφανείς επιφάνειες και γενικά στο 99% των επιφανειών. Επίσης, το φως ενός ποντικιού glaser δεν βλάπτει καθόλου το ανθρώπινο μάτι. Για πρώτη φορά, η A4Tech παρουσίασε το μοντέλο Glaser Mouse X6-70D, και ταυτόχρονα έσπασε το στερεότυπο που ήθελε τα ποντίκια laser να δυσλειτουργούν σε καθαρές επιφάνειες.

### 5.3 Οπτική Ροή (Optical flow)

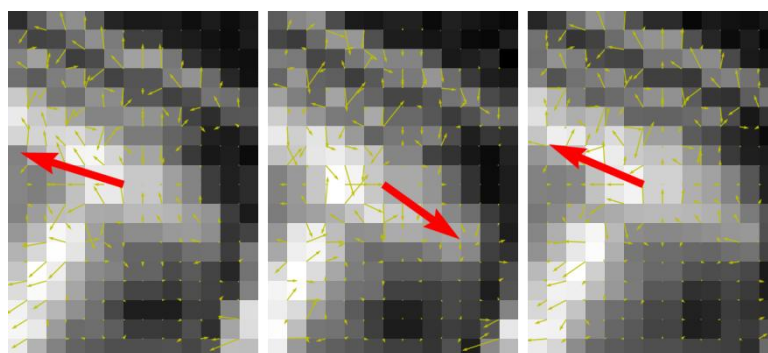
Μετά την σύντομη παρουσίαση των κυριότερων τεχνολογιών οπτικών αισθητήρων, κρίνεται αναγκαία η συνοπτική περιγραφή της μεθόδου της οπτικής ροής [18], με βάση την οποία λειτουργούν οι περισσότεροι. Η οπτική ροή αναφέρεται στο μοτίβο της κίνησης αντικειμένων, επιφανειών, και ακμών σε μια σκηνή, που οφείλεται στην σχετική κίνηση ενός παρατηρητή (αισθητήρα) ως προς τη σκηνή αυτή. Η ιδέα παρουσιάστηκε αρχικά από τον Αμερικανό φυσιολόγο James J. Gibson το 1940 για να περιγράψει την αντίληψη των ζώων για εικονικά ερεθίσματα.

Όταν η οπτική ροή εφαρμόζεται σε οπτικούς αισθητήρες, όπως είναι το ποντίκι υπολογιστή, στόχος είναι ο υπολογισμός της ακριβούς κίνησης του αισθητήρα πάνω στην επιφάνεια κίνησης. Για τον σκοπό αυτό, συλλέγονται διαδοχικές εικόνες κατά την διάρκεια της κίνησης. Η κάθε εικόνα συγκρίνεται με την προηγούμενη της και καταγράφεται η σχετική κίνηση των pixels της. Μέσα από τον υπολογισμό των διανυσμάτων κίνησης/ταχύτητας του συνόλου των pixels των 2 εικόνων, υπολογίζεται η συνολική εκτιμώμενη κίνηση του συστήματος (βλ. Σχήματα 5.4 και 5.5).

Οι αλγόριθμοι που χρησιμοποιούνται ποικίλουν. Συνοπτικά, οι κύριες μέθοδοι υπολογισμού της οπτικής ροής είναι οι εξής: phase correlation, μέθοδος ελαχίστων τετραγώνων, μέθοδοι βασισμένες σε παραγώγους και μέθοδοι διακριτής βελτιστοποίησης. Ας σημειωθεί ότι, το πεδίο εφαρμογών της τεχνικής αυτής είναι ευρύ και εκτείνεται από τη ρομποτική μέχρι και τη συμπίεση βίντεο.



Σχήμα 5.4 Υπολογισμός κίνησης από δύο διαδοχικές εικόνες A,B με τη μέθοδο της οπτικής ροής.



Σχήμα 5.5 Παράδειγμα υπολογισμού κίνησης με τη μέθοδο της οπτικής ροής.

### 5.4 Βασικά τεχνικά χαρακτηριστικά οπτικών αισθητήρων

- Πλήθος εικονοστοιχείων (pixels)

Ένας οπτικός αισθητήρας είναι στην ουσία μια μικρή ψηφιακή κάμερα, που λαμβάνει εικόνες στη κλίμακα του γκρι. Οι αισθητήρες που κυκλοφορούν διαθέτουν από 16x16 μέχρι και 30x30

εικονοστοιχεία (pixels). Περισσότερα εικονοστοιχεία αντιστοιχούν σε μεγαλύτερη ακρίβεια και μεγαλύτερη διακριτική ικανότητα, αλλά ταυτόχρονα απαιτούν μεγαλύτερη επεξεργαστική ισχύ.

- **Ανάλυση (Resolution)**

Η ανάλυση μετράται σε counts/inch (cpi) ή counts/mm (crmm) και είναι μέτρο της διακριτικής ικανότητας του αισθητήρα. Δείχνει ποια είναι η μικρότερη μετατόπιση που μπορεί να γίνει αντιληπτή. Η ανάλυση των σύγχρονων αισθητήρων κυμαίνεται από 800 μέχρι και 3200 cpi. Μια άλλη μονάδα μέτρησης που επίσης χρησιμοποιείται είναι το dpi (dots per inch).

- **Συχνότητα ανανέωσης (Imaging Rate)**

Μετράται σε fps (frames per second) ή Hz και αναφέρεται στη συχνότητα με την οποία η κάμερα παίρνει φωτογραφίες. Όσο μεγαλύτερη είναι, τόσο μεγαλύτερης ταχύτητας κινήσεις μπορούν να ανιχνευθούν αλλά και τόσο μεγαλύτερη επεξεργαστική ισχύς χρειάζεται. Τυπικές συχνότητες είναι 1500-7080 Hz.

- **Μέγιστη ταχύτητα (Tracking Speed)**

Μετράται σε inches/s ή mm/s και δηλώνει τη μέγιστη ταχύτητα κίνησης η οποία μπορεί να ανιχνευθεί. Εξαρτάται από όλα τα προηγούμενα χαρακτηριστικά και τυπικές τιμές είναι 10-50 inches/s. Μια άλλη μονάδα μέτρησης που απαντάται είναι το mickey per second, όπου ένα mickey αντιστοιχεί σε 1/200 της ίντσας.

Ένα άλλο χαρακτηριστικό ενός ποντικιού υπολογιστή, στο οποίο δεν έγινε αναφορά, είναι ο τρόπος σύνδεσης του με τον υπολογιστή, **ασύρματα ή ενσύρματα**. Τα σύγχρονα ποντίκια αξιοποιούν σε μεγάλο βαθμό την ασύρματη τεχνολογία, η οποία γνωρίζει μεγάλη ανάπτυξη στις μέρες μας. Αν και η ασύρματη σύνδεση, είναι εξίσου αξιόπιστη τώρα πια με την ενσύρματη, η επιλογή βασίζεται κυρίως στον τρόπο χρήσης της συσκευής. Σε κάποιες περιπτώσεις η ασύρματη επικοινωνία είναι αρκετά χρήσιμη, ενώ σε άλλες άχρηστη. Δεδομένου λοιπόν ότι, το κόστος ανεβαίνει και η ταχύτητα της απόκρισης του συστήματος μικραίνει λόγω του επιπρόσθετου chip (υπεύθυνο για την ασύρματη σύνδεση), η επιλογή ασύρματου ποντικιού χωρίς να υπάρχει σημαντικός λόγος μπορεί να είναι ζημιογόνα.

Στη περίπτωση που εξετάζεται εδώ, ο/οι οπτικοί αισθητήρες θα βρίσκονται πάνω στο ρομπότ και θα συνδέονται κατευθείαν στον υπολογιστή του. Οπότε, αρχικά, δεν υπάρχει κάποιος λόγος να χρησιμοποιηθεί ασύρματη σύνδεση και γι' αυτό οι σχετικές επιλογές δεν εξετάζονται στην παρούσα εργασία. Αν μελλοντικά κριθεί χρήσιμο οι οπτικοί αισθητήρες να συνδέονται ασύρματα (π.χ. για αποφυγή καλωδίων ή αποστολή δεδομένων σε εξωτερικό υπολογιστή), τότε η αντίστοιχη μελέτη θα έχει αξία.

## Είδη σφαλμάτων οπτικών αισθητήρων

Όπως ισχύει για κάθε μετρητική συσκευή, έτσι και ο οπτικός αισθητήρας επηρεάζεται από θόρυβο, που μπορεί να προέρχεται από πολλές πηγές. Το ιδιαίτερο χαρακτηριστικό όμως ενός οπτικού αισθητήρα είναι ότι το σφάλμα συσσωρεύεται. Το γεγονός αυτό είναι πολύ σημαντικό και πρέπει οπωσδήποτε να ληφθεί υπόψη κατά την ανάπτυξη του συστήματος των οπτικών αισθητήρων. Χρήσιμο είναι πρώτα να αναφερθούν τα είδη και οι αιτίες των σφαλμάτων αυτών. Ύστερα πρέπει να ληφθούν υπόψη κυρίως κατά τον σχεδιασμό αλγορίθμων επεξεργασίας και κατά την εκτέλεση πειραμάτων.

- Σφάλμα λόγω εξοικονόμησης ενέργειας. Ο αισθητήρας, όταν είναι ακίνητος, λειτουργεί σε μία κατάσταση ελαχιστοποίησης ενέργειας, και λαμβάνει φωτογραφίες σε αραιά διαστήματα. Μόλις διαπιστώσει κίνηση όμως, επανέρχεται στην κανονική του κατάσταση. Αυτό έχει σαν αποτέλεσμα ένα μικρό, χρονικό διάστημα, κατά το οποίο ο αισθητήρας δεν λαμβάνει δεδομένα, ή τα λαμβάνει με μεγάλο σφάλμα.



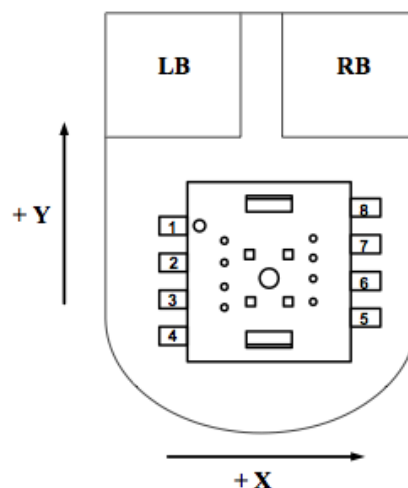
- Σφάλμα λόγω είδους επιφάνειας. Είναι γνωστό από την χρήση των οπτικών ποντικιών, ότι η απόδοση τους πέφτει σε επιφάνειες όπως το γυαλί. Γενικότερα, ανάλογα με την επιφάνεια κίνησης, πρέπει να αναμένεται διαφορετικός αριθμός παλμών για την ίδια μεταβολή θέσης.

- Σφάλμα λόγω απόστασης αισθητήρα από επιφάνεια. Η ενδεικνυόμενη απόσταση του κλείστρου από την επιφάνεια αναφέρεται συνήθως στα χαρακτηριστικά του αισθητήρα. Αναμένεται διαφορά στον αριθμό των παλμών για την ίδια κίνηση, ανάλογα την απόσταση του αισθητήρα από την επιφάνεια.

### 5.5 Αριθμός αισθητήρων στο ρομπότ

Πρέπει αρχικά να αναφερθεί ότι είναι αδύνατο με έναν μόνο οπτικό αισθητήρα να βρεθεί η θέση ενός σώματος που τον φέρει, ως προς ένα σταθερό εξωτερικό σύστημα συντεταγμένων. Οι μετατοπίσεις σε κάθε χρονικό βήμα δίνονται ως προς το κινούμενο σωματόδετο σύστημα συντεταγμένων του αισθητήρα (βλ. Σχήμα 5.6). Αυτό έχει σαν αποτέλεσμα την αδυναμία υπολογισμού πιθανής στροφής του αισθητήρα στο επίπεδο κίνησης, άρα και αδυναμία υπολογισμού θέσης του αισθητήρα ως προς ένα εξωτερικό σύστημα συντεταγμένων. Για την κατανόηση αυτού αρκεί να σκεφτούμε ότι, ο κέρσορας στην οθόνη του υπολογιστή μπορεί να ανεβαίνει προς τα πάνω, ανεξάρτητα με την κατεύθυνση κίνησης που επιβάλλει το χέρι μας στο ποντίκι, αρκεί αυτό να κινείται προς τα εμπρός σε σχέση με το σώμα του. Η κίνηση του κέρσορα εξαρτάται μόνο από τη κίνηση ως προς το σύστημα του αισθητήρα και όχι ως προς το παγκόσμιο σύστημα που ορίζεται π.χ. από το γραφείο μας.

Για τον υπολογισμό λοιπόν θέσης αλλά και προσανατολισμού (γωνίας), είναι απαραίτητη η χρήση τουλάχιστον δύο τέτοιων αισθητήρων εγκατεστημένων σε γνωστές θέσεις πάνω στη βάση του κινούμενου σώματος. Για λόγους εξάλειψης σφαλμάτων, τελικά κρίνεται καλή η χρήση τριών αισθητήρων. Σε αυτή τη περίπτωση τα δεδομένα είναι περισσότερα από τα ζητούμενα ( $x$ ,  $y$ ,  $\phi$ ), πράγμα που μπορεί να ενισχύσει την ακρίβεια στους υπολογισμούς, ανάλογα πάντα με τον αλγόριθμο επεξεργασίας των δεδομένων.



Σχήμα 5.6 Σωματόδετο σύστημα συντεταγμένων του αισθητήρα.

### 5.6 Σύνδεση αισθητήρων στο κεντρικό υπολογιστικό σύστημα

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, η ανάπτυξη του ρομποτικού συστήματος θα γίνει με τη μέθοδο σχεδιασμού βασισμένου στο μοντέλο (model-based design), και συγκεκριμένα με το πακέτο λογισμικού xPC Target της Mathworks. Το γεγονός αυτό, θέτει αυτόματα κάποιους περιορισμούς στη διαδικασία ανάπτυξης, οι οποίοι σχετίζονται με τη δομή και τις ιδιαιτερότητες του συγκεκριμένου πακέτου. Ένα χαρακτηριστικό του xPC Target που προσθέτει δυσκολία στην ανάπτυξη του συστήματος, είναι η μη συμβατότητα του με το πρωτόκολλο USB. Τη στιγμή που, η πλειοψηφία των ποντικιών υπολογιστή που κυκλοφορούν χρησιμοποιούν σύνδεση USB, το χαρακτηριστικό αυτό

αποτελεί πρόβλημα. Σε αντίθετη περίπτωση, το ιδανικότερο και απλούστερο σενάριο θα ήταν η απευθείας σύνδεση των αισθητήρων στη CPU μέσω USB, όπως γίνεται και στο πρώτο ρομπότ του διαστημικού εξομοιωτή του εργαστηρίου. Συγκεκριμένα, η Mathworks αναφέρει σχετικά με τις συνδέσεις USB:

*‘Ο λόγος που το USB δεν υποστηρίζεται, είναι ότι για να διεξαχθεί ένας επεξεργαστής επικοινωνία μέσω USB, χρησιμοποιεί System Management Interrupts (SMI), τα οποία μπλοκάρουν όλα τα άλλα interrupts όταν εκτελούνται. Οι περισσότεροι χειριστές SMI στα BIOS χρειάζονται παραπάνω από 100-200 [μsec] για να εκτελεστούν. Αυτό προκαλεί υπερφόρτωση για περιόδους δειγματοληψίας μικρότερες από κάποιες εκατοντάδες [μsec].’*

Η αρχική ιδέα ήταν η παράκαμψη αυτού του προβλήματος, με την ανάπτυξη ενός ανεξάρτητου αυτόνομου υποσυστήματος, υπεύθυνου για την σύνδεση, τη συλλογή και τελικά τη μεταφορά των δεδομένων των αισθητήρων στην κεντρική CPU. Οι πιθανές επιλογές στη φάση του σχεδιασμού του υποσυστήματος αυτού, ήταν η χρήση μικροελεγκτή ή η χρήση ενός SBC (single board computer), που θα αναλάμβανε τις παραπάνω λειτουργίες.

Παρόλο που στις μέρες μας, κυκλοφορούν αρκετά SBCs χαμηλού κόστους με θύρες USB και άλλες θύρες εισόδων εξόδων (I/O), θεωρήθηκε απλούστερη η υλοποίηση με μικροελεγκτή. Τα πλεονεκτήματα του τελευταίου είναι το πολύ χαμηλότερο κόστος, το μικρότερο μέγεθος, η μικρότερη απαιτούμενη ισχύς λειτουργίας, ο μικρότερος χρόνος ανάπτυξης και η δυνατότητα ανάπτυξης σε χαμηλότερο επίπεδο (δεν τρέχει κάποιο λειτουργικό σύστημα). Το μεγάλο μειονέκτημα όμως, είναι η δυσκολία χειρισμού συσκευών USB μέσω ενός μικροελεγκτή. Το πρωτόκολλο αυτό είναι αρκετά περίπλοκο, το ίδιο και ο αντίστοιχος προγραμματισμός του μικροελεγκτή, πόσο μάλλον στη περίπτωση επικοινωνίας με τρεις συσκευές USB (τρία ποντίκια).

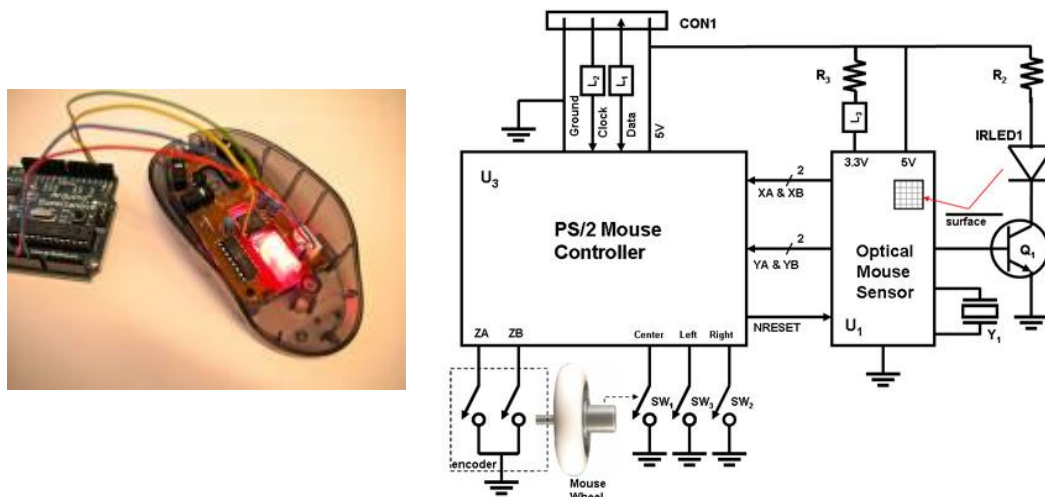
Το τελευταίο αυτό εμπόδιο ξεπεράστηκε, χρησιμοποιώντας το λίγο παλιότερο πρωτόκολλο επικοινωνίας των ποντικιών υπολογιστή, το γνωστό PS/2, δεδομένου ότι αρκετά ποντίκια που κυκλοφορούν στην αγορά είναι ακόμη συμβατά με αυτό. Αυτό επιλέχτηκε λόγω της απλότητας του σαν πρωτόκολλο, αλλά και της μεγάλης ευκολίας ενσωμάτωσής του σε ένα σύστημα με μικροελεγκτή.

## 5.7 Περιγραφή συστήματος μικροελεγκτή - PS/2 ποντικιών

Η σύνδεση PS/2 γίνεται με μια θύρα μορφής 6-pin Mini-DIN και χρησιμοποιείται για τη σύνδεση ποντικιών και πληκτρολογίων σε υπολογιστή. Το όνομα έρχεται από τη σειρά υπολογιστών Personal System/2 της IBM, που εμφανίστηκαν το 1987. Παρόλο που τα τελευταία χρόνια έχει αντικατασταθεί από το USB, είναι ένα πρωτόκολλο, που υποστηρίζεται ευρέως μέχρι και σήμερα. Παράδειγμα ότι, πολλά σύγχρονα ποντίκια λειτουργούν σε 2 modes (USB και PS/2) ανάλογα με τη θύρα στην οποία συνδέονται. Οι θύρες PS/2 ελέγχονται συνήθως από έναν μικροελεγκτή στη μητρική πλακέτα του υπολογιστή, με αποτέλεσμα το όλο σχέδιο και η κατασκευή να είναι πολύ απλά και φτηνά.

Το chip (μικροελεγκτής) που υπάρχει στην πλακέτα του αισθητήρα είναι αρκετά χρήσιμο παρόλο που οι χρήσιμες πληροφορίες μπορούν να αντληθούν και κατευθείαν από τον αισθητήρα, πράγμα το οποίο και δοκιμάστηκε επιτυχώς. Μέσω του chip αυτού συγκεντρώνονται όλες οι πληροφορίες κίνησης και κατάστασης πλήκτρων, και στέλνεται η επικοινωνία σύμφωνα με το πρωτόκολλο PS/2. Το όλο ηλεκτρονικό σύστημα λοιπόν, μιας συσκευής PS/2 ενδείκνυται για επικοινωνία με έναν κεντρικό μικροελεγκτή (βλ. Σχήμα 5.7) εφόσον συγκεντρώνει στοιχεία όπως: τάση TTL, χαμηλές απαιτήσεις ρεύματος και δέσμευση μόνο δύο pins για επικοινωνία.

Το PS/2 ποντίκι χρησιμοποιεί μία διπλής κατεύθυνσης σειριακή επικοινωνία για να μεταδώσει δεδομένα κίνησης και κατάστασης των πλήκτρων στον ελεγκτή του. Με τη σειρά του ο ελεγκτής μπορεί να στείλει διάφορες εντολές ρύθμισης της ανάλυσης, του ρυθμού αναφοράς αλλά και εντολές επανεκκίνησης ή κατάργησης. Σημειώνεται επίσης ότι, ένα μέσο PS/2 ποντίκι λειτουργεί στα 5V - 100mA.



**Σχήμα 5.7** Αριστερά: Εύκολη σύνδεση οπτικού αισθητήρα με μικροελεγκτή. Δεξιά: Τα υποσυστήματα μιας πλακέτας οπτικού αισθητήρα.

Το PS/2 πρωτόκολλο υποστηρίζει τις εξής εισόδους: x (δεξιά-αριστερά) κίνηση, y (πάνω-κάτω) κίνηση, αριστερό πλήκτρο, μεσαίο και δεξί πλήκτρο. Το ποντίκι περιοδικά διαβάζει αυτές τις εισόδους και ενημερώνει κάποιους μετρητές (counters) και κάποιες μεταβλητές (flags) ανάλογα με τις κινήσεις και την κατάσταση των πλήκτρων. Γενικά, μπορούν να υπάρχουν κι άλλα πλήκτρα και λειτουργίες. Έτσι, ένα απλό ποντίκι έχει 2 μετρητές (counters): έναν για τη κίνηση κατά x κι έναν για την κίνηση κατά y. Τα περιεχόμενά τους, μαζί με την κατάσταση των πλήκτρων, στέλνονται στον ελεγκτή σε μορφή ενός πακέτου δεδομένων κίνησης 3-byte.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Y overflow	X overflow	Y sign bit	X sign bit	Always 1	Middle Btn	Right Btn	Left Btn
Byte 2	X movement							
Byte 3	Y movement							

**Σχήμα 5.8** Πακέτο δεδομένων Κίνησης.

Το πακέτο δεδομένων κίνησης έχει την παραπάνω μορφή (βλ. Σχήμα 5.8). Οι τιμές για τη κίνηση είναι 9-bit συμπληρώματος ως προς 2 ακέραιοι, και αντιπροσωπεύουν τη μετατόπιση του αισθητήρα από τη θέση όπου στάλθηκε το προηγούμενο πακέτο, σε μονάδες που καθορίζονται από την ανάλυση (resolution) που έχει επιλεγεί. Το εύρος αυτών των τιμών εκτείνεται από -255 μέχρι +255. Σε περίπτωση που αυτό ξεπεραστεί, το αντίστοιχο bit υπερχείλισης ενεργοποιείται. Το status byte (Byte 1) περιέχει τα πρόσσημα των τιμών x,y και τις τιμές κατάστασης των πλήκτρων.

Υπάρχουν αρκετές βιβλιοθήκες προγραμματισμού μικροελεγκτή για επικοινωνία με οπτικό αισθητήρα μέσω του PS/2 πρωτοκόλλου, γραμμένες κυρίως σε C/C++. Η άμεση εύρεση και η εύκολη χρήση αυτών, ήταν ο κύριος λόγος που επιλέχθηκε το σύστημα PS/2 ποντίκι-μικροελεγκτή. Οι βιβλιοθήκες αυτές έχουν συνταχθεί με βάση αυτά που αναφέρθηκαν παραπάνω και χρησιμεύουν στη ρύθμιση παραμέτρων του αισθητήρα αλλά κυρίως στη λήψη δεδομένων κίνησης από αυτόν. Η βιβλιοθήκη που χρησιμοποιήθηκε στην παρούσα εφαρμογή είναι αυτή του Kris Chambers [19], η οποία δίνεται στο CD της εργασίας και επεξηγείται μέσα από κατάλληλα σχόλια επί του κώδικα.



## 5.8 Επιλογή μικροελεγκτή (microcontroller)

Ένας μικροελεγκτής είναι στην ουσία ένας υπολογιστής σε μια πιο συμπαγή μορφή. Έχει δηλαδή μία μονάδα επεξεργασίας (CPU) στην οποία εκτελούνται τα προγράμματα, μία μνήμη στην οποία αποθηκεύονται και ανανεώνονται κατά την εκτέλεση του προγράμματος οι διάφορες μεταβλητές καθώς και θύρες εισόδων – εξόδων (I/O ports). Η κύρια διαφορά τους από τους σύγχρονους υπολογιστές έγκειται στο γεγονός ότι οι μικροελεγκτές έχουν περιορισμένη μνήμη (της τάξης μερικών Kbytes, τα οποία αρκούν για τις περισσότερες εφαρμογές, ακόμα και τις πιο απαιτητικές) ενώ δεν έχουν σκληρό δίσκο. Τέλος, λειτουργούν με χαμηλή ισχύ (της τάξης των 50 [mW] σε σύγκριση με τα 50 [W] των ηλεκτρονικών υπολογιστών) και έχουν μικρότερη ταχύτητα επεξεργασίας δεδομένων.

Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές. Μερικοί από τους γνωστότερους κατασκευαστές είναι οι: ARM (δεν κατασκευάζει αλλά παραχωρεί δικαιώματα χρήσης του πυρήνα), Atmel, Epson, Hitachi, Microchip, NEC, Toshiba, Texas Instruments.

Ο προγραμματισμός ενός μικροελεγκτή απαιτεί πρώτον, γνώσεις ηλεκτρονικών και προγραμματισμού, και δεύτερον, τα απαραίτητα εργαλεία ανάπτυξης. Στις μέρες μας κυκλοφορούν στην αγορά αρκετές χαμηλού κόστους πλατφόρμες μικροελεγκτών, οι οποίες ‘κρύβουν’ τις δύσκολες λεπτομέρειες της αρχιτεκτονικής των μικροελεγκτών και επιτρέπουν τον άμεσο προγραμματισμό τους, προσφέροντας τα πάντα σε ένα ολοκληρωμένο πακέτο, έτοιμο για χρήση. Μερικά παραδείγματα είναι ο Basic Stamp της Parallax, ο BX-24 της Netmedia, το Handyboard του MIT, και βέβαια το διάσημο πλέον Arduino, το οποίο έκανε μια μικρή επανάσταση στον χώρο της ρομποτικής, δίνοντας την ευκαιρία σε πολλούς χρήστες (ερασιτέχνες και μη) να αναπτύξουν εύκολα τις δικές τους ιδέες. Για λόγους που θα φανούν στην πορεία, η πλατφόρμα Arduino ήταν αυτή που επιλέχθηκε τελικά για την ανάπτυξη του συστήματος των οπτικών αισθητήρων.

## 5.9 Η πλατφόρμα Arduino

Το Arduino [44] είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη C++ με κάποιες μετατροπές). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών εφαρμογών, αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Matlab κ.α.

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής. Οι πλακέτες Arduino προγραμματίζονται μέσω USB, χρησιμοποιώντας ένα τσίπ προσαρμογέα USB-to-serial όπως το FTDI FT232. Η πλακέτα του Arduino έχει εκτεθειμένες τις περισσότερες επαφές εισόδου/εξόδου για χρήση με άλλα κυκλώματα. Αυτές οι επαφές είναι διαθέσιμες στην κορυφή της πλακέτας μέσω θηλυκών συνδέσεων μεγέθους 0,1 ιντσών. Διάφορες plug-in πλακέτες ειδικών εφαρμογών γνωστές σαν ‘shields’ είναι, επίσης, διαθέσιμες στο εμπόριο.

Το περιβάλλον ανάπτυξης (IDE) του Arduino είναι βασισμένο στην Processing (η γλώσσα Java με φορτωμένες διάφορες βιβλιοθήκες ειδικές για γραφικά) και μπορεί να τρέξει σε πολλαπλές πλατφόρμες. Περιλαμβάνει επεξεργαστή κώδικα (επεξεργαστή κειμένου με διάφορα εύχρηστα εργαλεία) και μεταγλωττιστή, και έχει την ικανότητα να φορτώνει εύκολα το πρόγραμμα μέσω σειριακής θύρας από τον υπολογιστή στην πλακέτα.

## Πλεονεκτήματα της πλατφόρμας

Μερικά πλεονεκτήματα του Arduino έναντι άλλων πλακετών ανάπτυξης, που οδήγησαν και στην τελική επιλογή του για το υποσύστημα των οπτικών αισθητήρων, είναι:

- Χαμηλό κόστος. Οι πλακέτες του Arduino είναι εξαιρετικά φθηνές σε σχέση με άλλες πλατφόρμες μικροελεγκτών. Ειδικά δε μπορεί με τα σχηματικά που κυκλοφορούν στο Internet να κατασκευάσει κάποιος την φθηνότερη εκδοχή ενός Arduino. Ωστόσο ακόμα και αν προμηθευτεί την έτοιμη πλατφόρμα αυτή θα κοστίσει το μέγιστο 50 Ευρώ.
- Τρέχει σε διάφορα Λειτουργικά Συστήματα. Οι μηχανικοί λογισμικού, ανέπτυξαν το περιβάλλον προγραμματισμού του Arduino για Windows, Macintosh OSX και για λειτουργικά συστήματα Linux. Τα περισσότερα συστήματα ανάπτυξης μικροελεγκτών περιορίζονται στα Windows.
- Απλό, ξεκάθαρο προγραμματιστικό περιβάλλον. Το περιβάλλον προγραμματισμού ενός Arduino ενδείκνεται για αρχάριους, αλλά είναι ταυτόχρονα και ευέλικτο και για πιο προχωρημένους χρήστες.
- Ανοιχτού λογισμικού και λογισμικού που επεκτείνεται και παραμετροποιείται. Το software του Arduino διανέμεται με την μορφή εργαλείων ανοιχτού λογισμικού και είναι διαθέσιμο προς επέκταση για έμπειρους προγραμματιστές. Η γλώσσα προγραμματισμού του μπορεί να επεκταθεί διαμέσου των βιβλιοθηκών της C++.
- Ανοιχτού Υλικού το οποίο μπορεί να επεκταθεί. Το Arduino βασίζεται στους μικροελεγκτές της Atmel. Τα σχηματικά για τα αναπτυξιακά είναι κάτω από την άδεια της Creative Commons, επιτρέποντας σε σχεδιαστές να κατασκευάσουν το δικό τους αναπτυξιακό, εξελίσσοντας το ήδη υπάρχον, χωρίς να έχουν νομικά προβλήματα. Οι χρήστες μπορούν επίσης να επιδιώξουν την αντιγραφή και κατασκευή της πλακέτας ή μέρους αυτής για κάποια ειδική εφαρμογή.

## Το μοντέλο Arduino UNO

Κυκλοφορούν περίπου 10 πλατφόρμες Arduino, καθεμία από τις οποίες προσφέρει διαφορετικές δυνατότητες. Στην παρούσα εφαρμογή χρησιμοποιήθηκε το Arduino UNO, καθώς είναι σχετικά καινούριο, μικρό, οικονομικό (25 Ευρώ) και καλύπτει τις απαιτήσεις του εξεταζόμενου συστήματος.

Πιο αναλυτικά :

Το Arduino Uno είναι μία πλατφόρμα μικροελεγκτή βασισμένη στον ATmega328. Έχει 14 ψηφιακές εισόδους/εξόδους (από τις οποίες, 6 μπορούν να χρησιμοποιηθούν σαν έξοδοι PWM), 6 αναλογικές εισόδους, ένα κεραμικό ταλαντωτή 16 MHz, μια θύρα USB, ένα power jack, πινάκια ICSP, κι ένα κουμπί επανεκκίνησης. Τα τεχνικά χαρακτηριστικά του παρουσιάζονται στο Σχήμα 5.9.



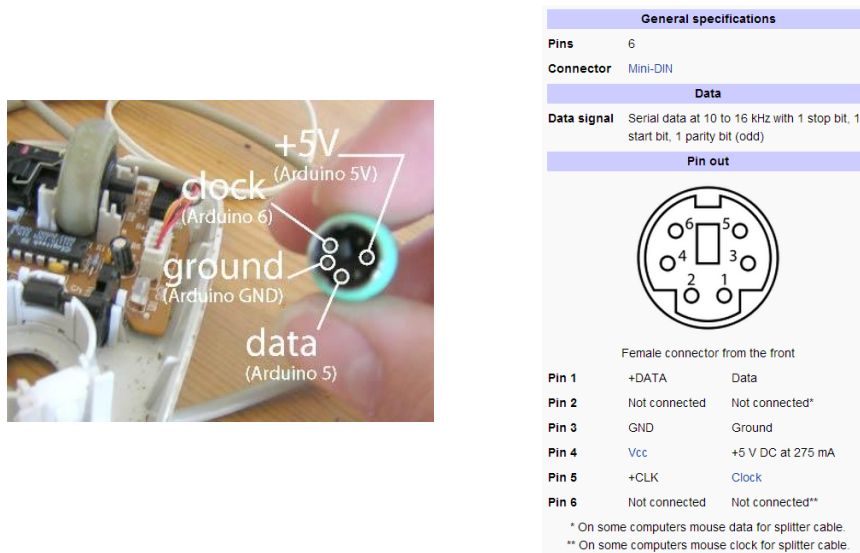
### Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3-3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

**Σχήμα 5.9** Αριστερά: η πλακέτα Arduino. Δεξιά: Τεχνικά χαρακτηριστικά της πλακέτας.

### 5.10 Σύνδεση μικροελεγκτή Arduino – PS/2 ποντικιού

Μία θύρα PS/2 αποτελείται από 6 pins, όπως φαίνεται και στο παρακάτω σχήμα. Ένα pin αντιστοιχεί στη γείωση, ένα στο power, 2 pins αφιερώνονται στο ρολόι (clock) και στα δεδομένα (data), ενώ τα άλλα 2 pins δεν χρησιμοποιούνται (βλ. Σχήμα 5.10).



Σχήμα 5.10 Περιγραφή των pins της θύρας PS/2.

Η σύνδεση ενός οπτικού αισθητήρα PS/2 στον μικροελεγκτή Arduino αναλύεται στις εξής 4 συνδέσεις. Το power pin συνδέεται στο +5 header pin του Arduino, το Ground pin στο αντίστοιχο Ground pin του Arduino και τέλος, το data pin και το clock σε δύο οποιαδήποτε digital pins. Με την απλή αυτή συνδεσμολογία και χρησιμοποιώντας κατάλληλα την βιβλιοθήκη PS/2 που αναφέρθηκε προηγουμένως, τα δεδομένα του οπτικού αισθητήρα μπορούν εύκολα να διαβαστούν μέσω της πλατφόρμας Arduino.

Αφού λοιπόν δοκιμάστηκε και επιβεβαιώθηκε η συλλογή δεδομένων κίνησης με τον τρόπο αυτόν, χρησιμοποιώντας 2-3 συσκευές ποντικιών υπολογιστή που ήταν διαθέσιμες, το επόμενο βήμα ήταν η κατάλληλη επιλογή οπτικού αισθητήρα που να ικανοποιεί τις απαιτήσεις του συστήματος.

### 5.11 Επιλογή οπτικού αισθητήρα

Σύμφωνα με την ανάλυση που έγινε στην αρχή του κεφαλαίου, οι τεχνολογίες Darkfield και Blue Track, υπερέρχουν στα τεχνικά χαρακτηριστικά, αλλά τα μοντέλα τους δεν είναι συμβατά με το πρωτόκολλο PS/2. Οι τεχνολογίες laser και Glaser, συγκεντρώνουν επίσης πολύ καλά χαρακτηριστικά και μάλιστα υπάρχουν μοντέλα τους στην αγορά τα οποία είναι συμβατά με σύνδεση PS/2. Παρόλα αυτά, παρατηρήθηκε η διαθεσιμότητα τους σε πολύ περιορισμένο αριθμό καταστημάτων, πράγμα που εμπόδιζε την άμεση αγορά τους.

Αυτό τελικά που μπορούσε να δοκιμαστεί άμεσα ήταν το Microsoft Ready mouse (ΠΑΡΑΡΤΗΜΑ Α), το οποίο ανήκει στη κατηγορία των οπτικών ποντικιών (βλ. Σχήμα 5.11). Όπως έχει ήδη αναφερθεί παραπάνω, είναι η κατηγορία με τη μικρότερη ακρίβεια και τα ασθενέστερα τεχνικά χαρακτηριστικά. Λόγω όμως, της δυνατότητας άμεσης χρήσης τους, χρησιμοποιήθηκε σε πειράματα. Τα χαρακτηριστικά της συσκευής ήταν πολύ ικανοποιητικά, ενώ μέσω των πειραμάτων διαπιστώθηκε η λειτουργικότητα πάνω στην τράπεζα γρανίτη. Γρήγορα αγοράστηκαν άλλα δύο τέτοια ποντίκια, και σε διάστημα λίγων ημερών μία πρώτη μορφή του συστήματος οπτικών αισθητήρων ήταν έτοιμη, σε πολύ χαμηλή τιμή (συνολικό κόστος περίπου στα 60 Ευρώ). Σημειώνεται ξανά ότι η υλοποίηση αυτή έγινε, με γνώμονα τον μικρό χρόνο και κόστος υλοποίησης και πρέπει σίγουρα να επανεξεταστεί στο

μέλλον. Παρόλα αυτά, όπως θα αποδειχθεί στην συνέχεια της εργασίας, τα πειραματικά αποτελέσματα ήταν πολύ καλά. Σε περίπτωση, που η ομάδα θελήσει στο μέλλον να επανασχεδιάσει το υποσύστημα αυτό, στο κεφάλαιο Μελλοντική Εργασία, προτείνονται εναλλακτικοί τρόποι υλοποίησης.



**Σχήμα 5.11** Microsoft Ready Mouse.

Σύμφωνα με το datasheet (βλ. ΠΑΡΑΡΤΗΜΑ Α) τα κύρια τεχνικά χαρακτηριστικά του μοντέλου Ready Mouse είναι τα ακόλουθα:

- Συχνότητα οπτικών αισθητήρων: 3000 frames per second
- Ανάλυση αισθητήρων: 800 ppi (31.5 points per mm)
- Μέγιστη ταχύτητα: 508 [mm] per second

## 5.12 Αποστολή δεδομένων στο PC-104

Ένας πολύ άμεσος και απλός τρόπος επικοινωνίας σε επίπεδο μικροελεγκτών είναι η σειριακή επικοινωνία [45]. Παρόλο που είναι σχετικά παλιός τρόπος επικοινωνίας, αλλά και αργός στην μεταφορά δεδομένων, ενδείκνυται για χρήση τόσο στο Arduino όσο και στο xPC Target. Τα παραδείγματα που υπάρχουν και για τα δύο πακέτα, είναι πολλά, και η επικοινωνία αυτή απλά και γρήγορα υλοποιήσιμη. Για μία ακόμη φορά, προτιμήθηκε ο γρήγορος και απλός τρόπος ανάπτυξης και όχι απαραίτητα ο βέλτιστος, με στόχο την γρήγορη πρώτη υλοποίηση ενός ολοκληρωμένου συστήματος. Η επικοινωνία λοιπόν που στήθηκε τελικά ανάμεσα στο Arduino και στην CPU του PC-104 ήταν η σειριακή μέσω μιας θύρας RS-232. Όπως περιγράφεται αναλυτικά σε επόμενο κεφάλαιο, χρησιμοποιήθηκε ένα ολοκληρωμένο κύκλωμα max232.

Αρχικά έγιναν αρκετές δοκιμές με περισσότερα του ενός οπτικά ποντίκια και με αποστολή των δεδομένων μέσω σειριακής επικοινωνίας. Τα κυκλώματα σχεδιάστηκαν και στήθηκαν σε breadboard, έτσι ώστε να αξιολογηθούν οι προηγούμενες ιδέες και να παρθούν αποφάσεις για την τελική υλοποίηση του συστήματος (βλ. Σχήμα 5.12). Τα αποτελέσματα από τους οπτικούς αισθητήρες αλλά και από την σειριακή επικοινωνία, αξιολογήθηκαν μέσω Matlab και τελικά, το συνολικό σύστημα κρίθηκε αξιόπιστο αφού ικανοποιούσε όλες τις σχεδιαστικές απαιτήσεις που αναφέρθηκαν σε προηγούμενο κεφάλαιο.

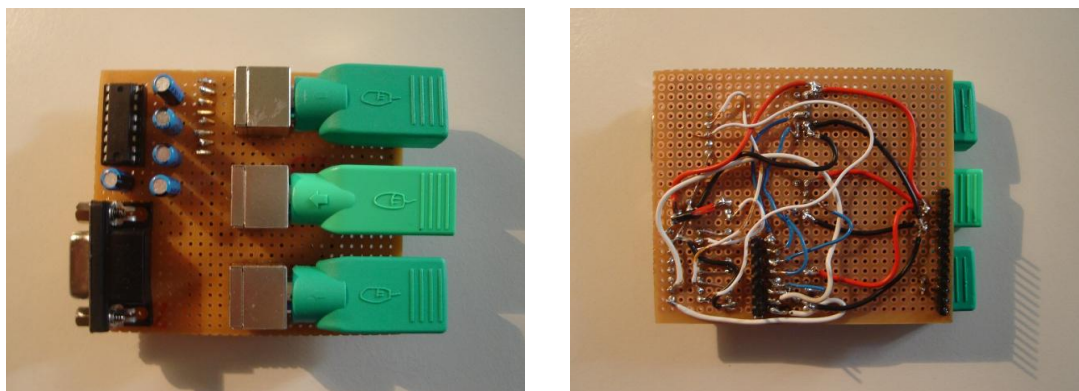


**Σχήμα 5.12** Αριστερά: Πείραμα λήψης δεδομένων από 3 οπτικούς αισθητήρες. Δεξιά: δοκιμή κυκλώματος σειριακής επικοινωνίας RS-232.

### 5.13 Σχεδιασμός και κατασκευή πρωτοτύπου Arduino Shield

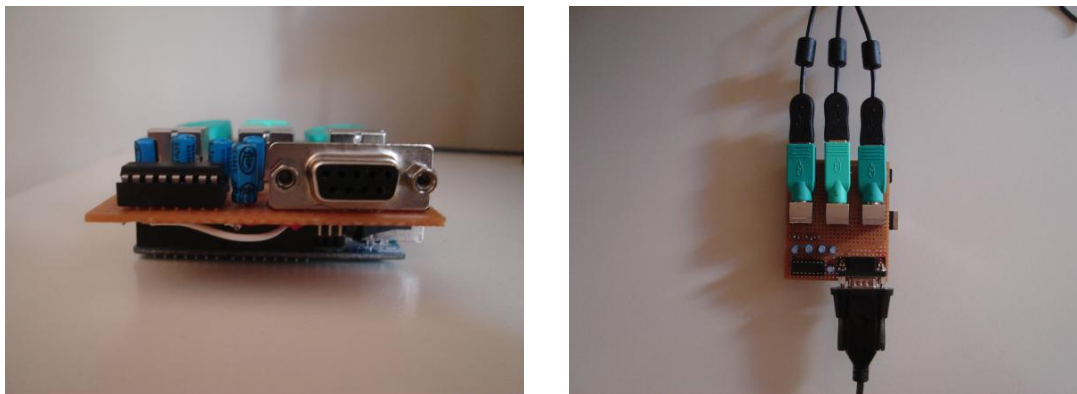
Σαν επόμενο βήμα, χρησιμοποιήθηκε διάτρητη πλακέτα, έτσι ώστε να κατασκευαστεί μια πρώτη μορφή ενός Arduino Shield ειδικού για συλλογή δεδομένων από 3 οπτικούς αισθητήρες και αποστολή δεδομένων σε ένα εξωτερικό υπολογιστικό σύστημα μέσω σειριακής επικοινωνίας RS-232. Όλα τα κυκλώματα που δοκιμάστηκαν στα παραπάνω πειράματα, ενσωματώθηκαν σε μια πλακέτα πλήρως λειτουργική, η οποία χρησιμοποιήθηκε στο εξής για πληθώρα πειραμάτων (βλ. Σχήματα 5.13 και 5.14).

Η σημαντικότητα αυτής της πλακέτας έγκειται στο γεγονός ότι είναι ένα πλήρως αυτόνομο υποσύστημα συλλογής και αποστολής δεδομένων κίνησης. Μπορεί λοιπόν, να χρησιμοποιηθεί σε άλλα ρομπότ του εργαστηρίου χωρίς να γίνει κάποια τροποποίηση. Επιπρόσθετα, σε περίπτωση που σε ένα ρομποτικό σύστημα δεν υπάρχει άλλη επεξεργαστική μονάδα εκτός από αυτή, για να αναλάβει την επεξεργασία των δεδομένων και τον εντοπισμό θέσης/ προσανατολισμού του σώματος, ο αλγόριθμος επεξεργασίας φορτώνεται και αυτός στο Arduino, μετατρέποντας το υποσύστημα αυτό σε μία αυτόνομη συσκευή εντοπισμού θέσης/προσανατολισμού στο επίπεδο. Ο κώδικας που πρέπει να φορτωθεί στο Arduino για τον σκοπό αυτό, η θεωρία του οποίου αναλύεται στη συνέχεια, δίνεται στο CD της εργασίας.



**Σχήμα 5.13** Διαφορετικές όψεις του Arduino Shield.

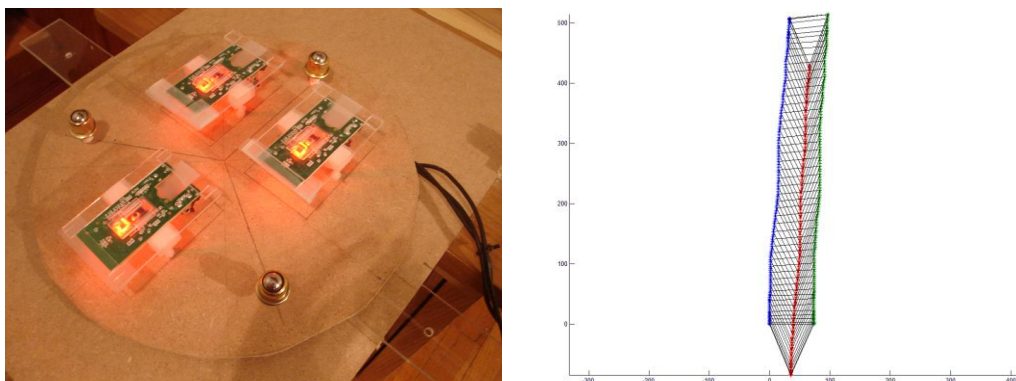




**Σχήμα 5.14** Σύνδεση Shield με οπτικούς αισθητήρες, Arduino και σειριακή μετάδοση αποτελεσμάτων.

Σε επόμενο κεφάλαιο, όπου περιγράφεται ο λεπτομερής σχεδιασμός και η τελική εκτύπωση του συνόλου των ηλεκτρονικών πλακετών του ρομπότ, αναλύεται και επισυνάπτεται μαζί με τα υπόλοιπα και το αναλυτικό ηλεκτρονικό σχέδιο του Arduino Shield.

### 5.14 Διάταξη αισθητήρων στο σώμα του ρομπότ

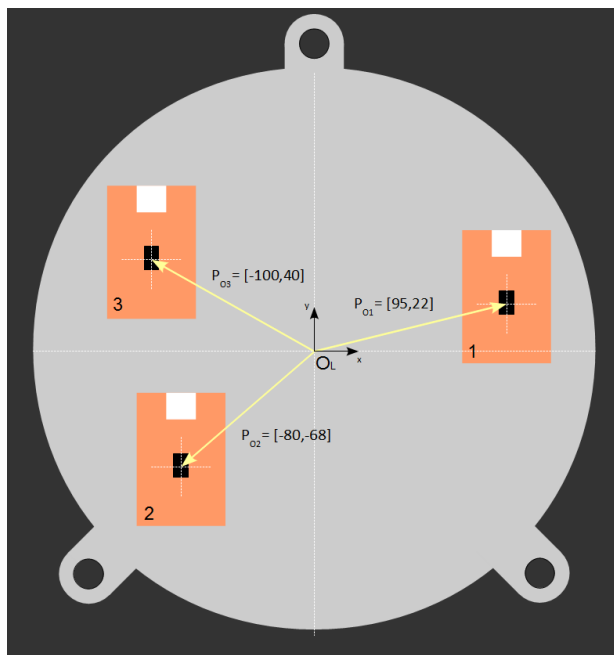


**Σχήμα 5.15** Αριστερά: πρώτη πειραματική διάταξη με οπτικούς αισθητήρες. Δεξιά: γράφημα των θέσεων των 3 οπτικών αισθητήρων σε πείραμα 4 sec.

Με στόχο τον προσδιορισμό της κίνησης ενός σώματος το οποίο φέρει αυτό το υποσύστημα εντοπισμού θέσης/προσανατολισμού, απαραίτητη είναι η τοποθέτηση των οπτικών αισθητήρων σε γνωστές (με ακρίβεια) αποστάσεις μεταξύ τους. Όσον αφορά στην διάταξη των οπτικών αισθητήρων, ενώ δεν υπάρχουν περιορισμοί, είναι προτιμότερο να αποφεύγονται σχήματα, όπως ευθεία ή ισόπλευρο τρίγωνο. Θεωρείται καλύτερη η διάταξη που παρουσιάζει λιγότερες συμμετρίες. Τα συστήματα συντεταγμένων τους, όμως, τοποθετούνται παράλληλα μεταξύ τους, διευκολύνοντας έτσι πολύ την διαδικασία του προγραμματισμού του αλγορίθμου εντοπισμού. Οι αποστάσεις των αισθητήρων από το σημείο του σώματος, στο οποίο θα ανάγονται οι μετατοπίσεις αυτού, είναι επίσης σημαντικές και πρέπει να είναι γνωστές με ακρίβεια κατά τον προγραμματισμό. Πιο συγκεκριμένα, πρέπει να οριστεί ένα σωματόδετο σύστημα συντεταγμένων που θα αποτελεί αναφορά για τις μετρήσεις που λαμβάνονται. Όπως φαίνεται στο Σχήμα 5.16, το σωματόδετο σύστημα ορίζεται έτσι, ώστε να έχει παράλληλους άξονες με αυτούς των συστημάτων των οπτικών αισθητήρων και αρχή των αξόνων στο κέντρο του κύκλου που περιγράφει το ρομπότ.

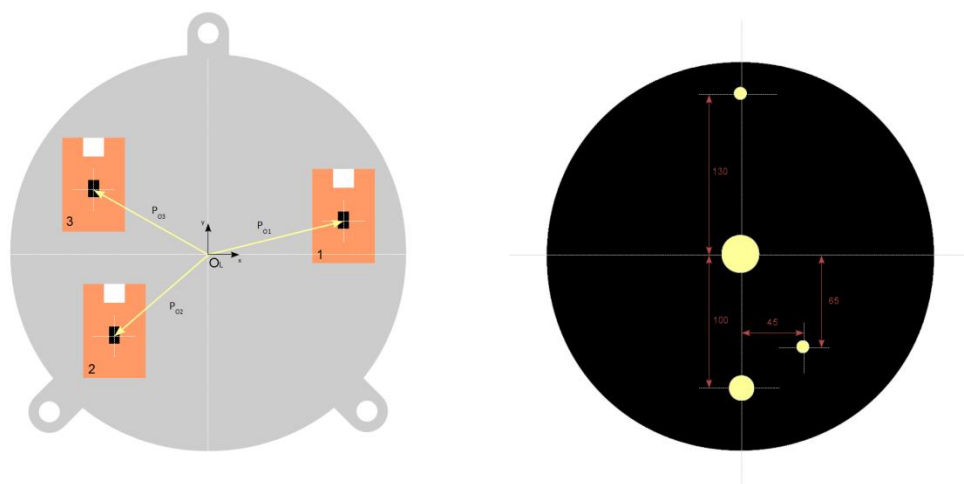
Κατά τα πρώτα πειράματα (βλ. Σχήμα 5.15), κατασκευάστηκε μία ξύλινη βάση, όμοια με το κάτω μέρος του πραγματικού ρομπότ, πάνω στην οποία τοποθετήθηκαν οι τρεις οπτικοί αισθητήρες. Αυτή ήταν η πειραματική διάταξη στην οποία εφαρμόστηκαν πολλοί από τους αλγόριθμους που περιγράφονται παρακάτω. Ενώ αρχικά (στην ξύλινη βάση), οι θέσεις των αισθητήρων

τοποθετήθηκαν σε σχήμα ισοσκελούς τριγώνου, η τελική τοποθέτηση στο πραγματικό ρομπότ, έγινε διαφορετικά (βλ. Σχήμα 5.16). Εδώ, επιβλήθηκαν γεωμετρικοί περιορισμοί από τα μηχανικά στοιχεία του ρομπότ, και αυτοί ήταν που οδήγησαν στην παρούσα διάταξη.



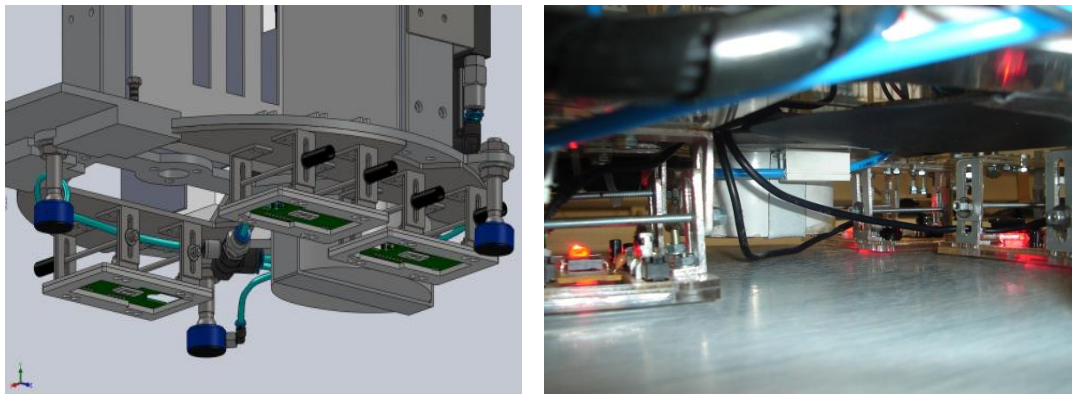
**Σχήμα 5.16** Διανύσματα θέσης των 3 αισθητήρων ως προς το σωματόδετο σύστημα συντεταγμένων.

Προσοχή δόθηκε επίσης, κατά την τοποθέτηση, προκειμένου το σύστημα συντεταγμένων των οπτικών αισθητήρων, να ταυτίζεται με το σωματόδετο σύστημα που αφορά την κάμερα, όπως αυτό ορίστηκε από την διάταξη των LEDs στο πάνω μέρος του ρομπότ (βλ. Σχήμα 5.17). Κάτι τέτοιο είναι σημαντικό, έτσι ώστε τα δεδομένα θέσης/προσανατολισμού των δύο συστημάτων εντοπισμού, να έχουν την ίδια αναφορά.



**Σχήμα 5.17** Κάτω και πάνω μέρος του ρομπότ. Τα ΣΣ των συστημάτων κάμερας και οπτικών αισθητήρων ταυτίζονται.

Για την σωστή και εύκολη τοποθέτηση των αισθητήρων στο κάτω μέρος του ρομπότ, κατασκευάστηκαν από τον Σ. Ανδρέου [15], ειδικές βάσεις. Η κάθε βάση φέρει ένα μηχανισμό ντίζας – περικοχλίων, με τον οποίο ρυθμίζεται το ύψος (απόσταση από το δάπεδο) του οπτικού αισθητήρα (βλ. Σχήμα 5.18). Αυτό θεωρήθηκε απαραίτητο έτσι, ώστε να υπάρχει η δυνατότητα ρύθμισης της σημαντικής αυτής παραμέτρου στην αρχή ή και κατά τη διάρκεια των πειραμάτων. Το σύστημα του μικροελεγκτή Arduino τοποθετείται στο πιο πάνω επίπεδο του ρομπότ, δίπλα στο κεντρικό υπολογιστικό σύστημα.



**Σχήμα 5.18** Αριστερά: Μηχανολογικό σχέδιο των ρυθμιζόμενων βάσεων των οπτικών αισθητήρων  
Δεξιά: φωτογραφία της πραγματικής διάταξης πάνω στο ρομπότ.

### 5.15 Αλγόριθμοι εντοπισμού

Η διαδικασία εντοπισμού μέσω του συστήματος των οπτικών αισθητήρων υλοποιείται στις χρονικές στιγμές ανάμεσα στις στιγμές που δεδομένα από την κάμερα είναι διαθέσιμα. Σε αυτή τη περίπτωση, υπάρχουν δεδομένα μόνο από το σύστημα αυτό. Σκοπός είναι η εύρεση συντεταγμένων κέντρου και γωνίας ρομπότ ως προς ένα εξωτερικό σύστημα συντεταγμένων (ΣΣ  $O_R$  στο σχήμα 5.19) που ορίστηκε την τελευταία φορά που υπήρχαν δεδομένα από την κάμερα.

Σημαντική δουλειά πάνω σε αυτό το θέμα έχει γίνει στις εργασίες [4] [16]. Στην παρούσα εργασία, το θέμα μελετάται λίγο διαφορετικά, με απώτερο στόχο την άμεση εφαρμογή και πειραματική αξιολόγηση των αλγορίθμων που θα αναπτυχθούν.

Στην παράγραφο αυτή, γίνεται η θεωρητική ανάλυση μερικών αλγορίθμων εντοπισμού που σχεδιάστηκαν για το σύστημα των αισθητήρων που παρουσιάστηκε προηγουμένως. Οι αλγόριθμοι αυτοί μπορούν να υλοποιηθούν σε δύο επίπεδα:

**α)** στο επίπεδο του μικροελεγκτή (προγραμματίζοντας δηλαδή κατάλληλα το Arduino). Σε αυτή την περίπτωση, οι πληροφορίες που δίνεται σαν έξοδος του συστήματος μέσω σειριακής είναι κατευθείαν οι συντεταγμένες του κέντρου, και η γωνία του κινούμενου σώματος. Η υλοποίηση αυτή (υπάρχει στο CD της εργασίας) έχει το πλεονέκτημα ότι κάνει τη συσκευή πλήρως αυτόνομη, και τα μειονεκτήματα ότι δύσκολα γίνονται αλλαγές στον αλγόριθμο κι ότι δεν υπάρχει εποπτεία σε χαμηλό επίπεδο την ώρα του πειράματος.

**β)** στο επίπεδο της κεντρικής υπολογιστικής μονάδας (CPU). Με αυτόν τον τρόπο, ο οποίος χρησιμοποιήθηκε περισσότερο στα πειράματα, το σύστημα του Arduino απλά μεταφέρει μέσω σειριακής στην κεντρική CPU, τα ακατέργαστα δεδομένα που παίρνει από τους οπτικούς αισθητήρες. Καμία επεξεργασία δεν γίνεται στο επίπεδο του μικροελεγκτή, ενώ ο αλγόριθμος επεξεργασίας υλοποιείται σαν μέρος του συνολικού προγράμματος που τρέχει το ρομπότ στο πλαίσιο του xPC Target. Η υλοποίηση σε αυτό το επίπεδο δίνει το πλεονέκτημα του πλήρους ελέγχου των δεδομένων κίνησης από τους οπτικούς αισθητήρες, κι αυτός είναι τελικά ο λόγος που προτιμάται.



Οι αλγόριθμοι επεξεργασίας δεν εξαρτώνται από το επίπεδο υλοποίησης (μικροελεγκτής ή CPU). Ακολουθεί λοιπόν, η μαθηματική ανάλυση της κίνησης ενός σώματος με 3 οπτικούς αισθητήρες και ύστερα περιγράφονται κάποιοι προτεινόμενοι αλγόριθμοι. Σημειώνεται, ότι η δυσκολία που παρουσιάζει αυτό το εγχείρημα έγκειται στο γεγονός ότι, σε τέτοιου είδους αισθητήρες η ακρίβεια είναι μεν μεγάλη, αλλά το σφάλμα συσσωρεύεται, αυξάνει με τον χρόνο και επομένως οι εκτιμήσεις γίνονται όλο και πιο αβέβαιες όσο το πείραμα εξελίσσεται.

### 5.15.1 Βασικός κορμός αλγορίθμων

Όλοι οι αλγόριθμοι που θα αναπτυχθούν στη συνέχεια, έχουν ένα κοινό βασικό κορμό. Σε αυτόν υλοποιείται η απαραίτητη προκαταρκτική επεξεργασία των δεδομένων κίνησης, και δίνεται η βάση για την θεμελίωση οποιουδήποτε αλγορίθμου επεξεργασίας. Στην παράγραφο αυτή θα περιγραφεί ο βασικός αυτός κορμός (βλ. Σχήμα 5.19). Σημειώνεται ξανά ότι, για λόγους απλοποίησης των υπολογισμών, οι αισθητήρες έχουν τοποθετηθεί έτσι, ώστε τα συστήματα συντεταγμένων τους να είναι παράλληλα μεταξύ τους και τελικά παράλληλα με το σωματόδετο σύστημα συντεταγμένων  $O_L$  του ρομπότ. Οπότε, η γωνία περιστροφής του ρομπότ θα είναι ίση με τη γωνία περιστροφής κάθε αισθητήρα. Τα βήματα που ακολουθούν υλοποιούνται για κάθε πακέτο δεδομένων που λαμβάνεται από το σύστημα των αισθητήρων.

1. Κάθε χρονική στιγμή λαμβάνονται 3 σετ μετατοπίσεων  $dx, dy$  από τους 3 αισθητήρες. Οι μετατοπίσεις αυτές, όπως τις δίνει ο κάθε αισθητήρας, είναι ορισμένες στο κινητό σύστημα συντεταγμένων του. Έτσι, για να αξιοποιηθούν κατάλληλα πρέπει πρώτα να εκφραστούν ως προς τα συστήματα των αισθητήρων την χρονική στιγμή μηδέν του πειράματος. Αναφορικά με το σχήμα..., πρέπει από τα  $dx, dy$  να υπολογιστούν τα  $Dx, Dy$ . Ο υπολογισμός αυτός απαιτεί τη γνώση της γωνίας  $\theta$  του ρομπότ ως προς το σύστημα  $O_R$  της στιγμής μηδέν, την γωνία στροφής δηλαδή του  $\Sigma O_L$  ως προς το  $\Sigma O_R$ . Η γωνία αυτή στον πρώτο βρόχο θεωρείται ίση με μηδέν, ενώ για κάθε επόμενο βρόχο έχει υπολογιστεί στο τέλος του προηγούμενου, όπως θα δειχθεί παρακάτω.

Για να βρεθούν λοιπόν τα  $Dx, Dy$  από τα  $dx, dy$  αρκεί να στραφεί το διάνυσμα  $\mathbf{p}^*$  κατά γωνία  $\theta$  έτσι ώστε οι συντεταγμένες του να εκφραστούν ως προς το εξωτερικό  $\Sigma O_R$ . Πολλαπλασιάζοντας με τον κατάλληλο πίνακα περιστροφής, προκύπτουν τα εξής (για έναν αισθητήρα):

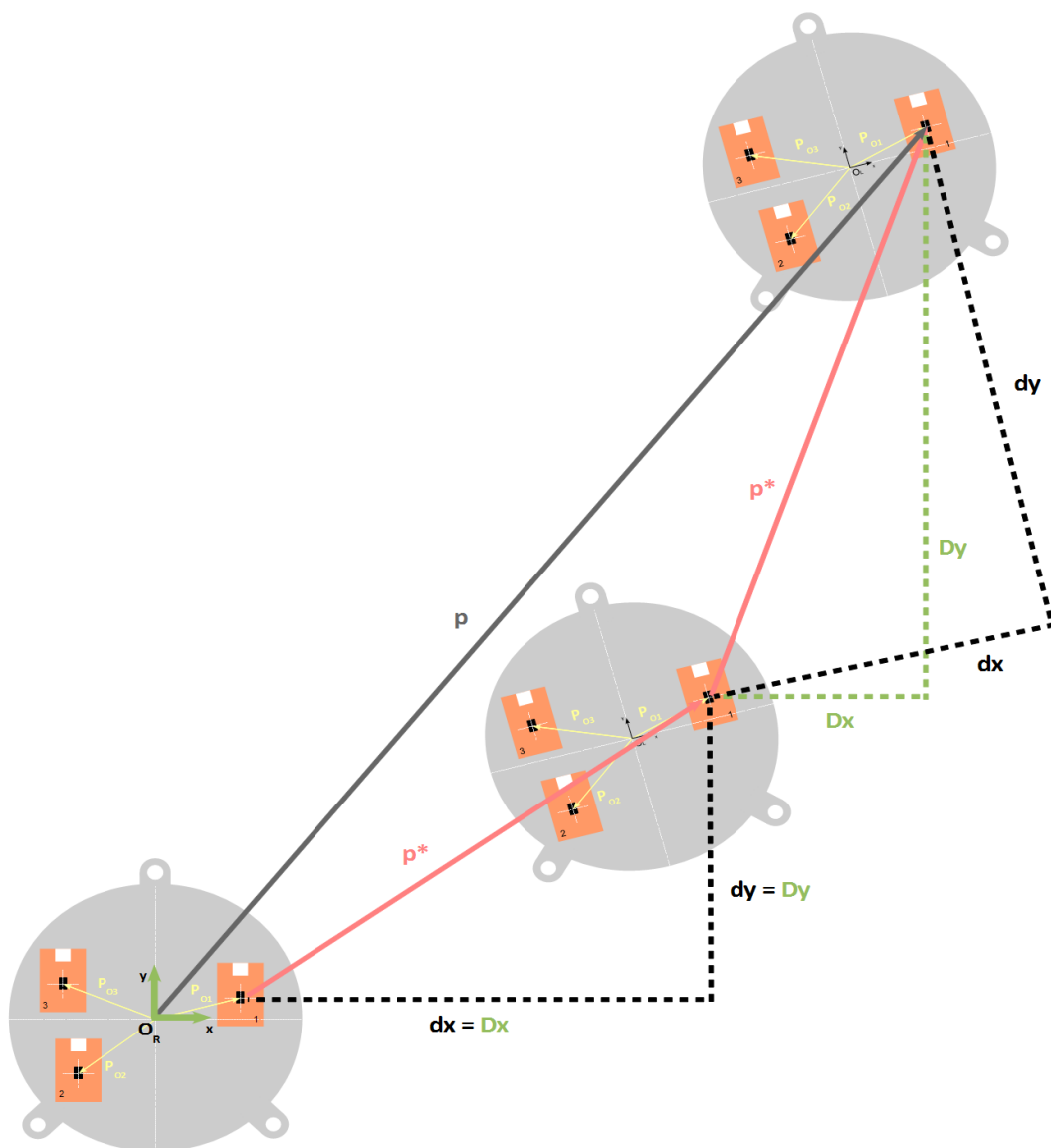
$$\begin{bmatrix} Dx \\ Dy \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}$$

Άρα,

$$\begin{aligned} Dx &= \cos\theta dx - \sin\theta dy \\ Dy &= \cos\theta dy + \sin\theta dx \end{aligned}$$

2. Δεδομένου ότι σε κάθε επανάληψη είναι γνωστή η προηγούμενη θέση του κάθε αισθητήρα έστω  $x_n, y_n$  ως προς το σύστημα του την χρονική στιγμή μηδέν, με πρόσθεση των νέων μετατοπίσεων  $Dx_n, Dy_n$  βρίσκεται η τρέχουσα θέση του ως προς το ίδιο σύστημα. Η θέση λοιπόν ενός αισθητήρα ως προς την αρχική του είναι:

$$\begin{aligned} x_{n, current} &= x_{n, old} + Dx_n & n &= 1,2,3 \\ y_{n, current} &= y_{n, old} + Dy_n & n &= 1,2,3 \end{aligned}$$



**Σχήμα 5.19** Τρία διαδοχικά στιγμιότυπα τυχαίας κίνησης και διανυσματική ανάλυση αυτής για έναν αισθητήρα.

3. Με σκοπό οι θέσεις των αισθητήρων να έχουν την ίδια αναφορά, εκφράζονται ως προς το σύστημα  $O_R$ . Αυτό είναι εφικτό, εφόσον είναι γνωστές οι θέσεις των 3 αισθητήρων ως προς το σωματόδετο σύστημα  $O_L$  του ρομπότ (θέσεις τοποθέτησης στο σασί) και περιέχονται στα διανύσματα θέσης  $\mathbf{p01}$ ,  $\mathbf{p02}$ ,  $\mathbf{p03}$ . Οι τρέχουσες λοιπόν θέσεις ως προς το ΣΣ  $O_R$  περιέχονται στα διανύσματα  $\mathbf{p1}$ ,  $\mathbf{p2}$ ,  $\mathbf{p3}$  (βλ. Σχήμα 5.20) και υπολογίζονται με πρόσθεση των κατάλληλων διανυσμάτων, ως εξής:

$$\mathbf{pn} = \begin{bmatrix} x^n \\ y^n \end{bmatrix} + \mathbf{p0n} \quad n = 1,2,3$$

4. Σε αυτό το βήμα είναι δυνατός ο υπολογισμός 3 γωνιών (στροφής του σώματος), μια για κάθε ζευγάρι αισθητήρων. Η μεταβολή της κάθε γωνίας από την γνωστή αρχική γωνία τοποθέτησης των δύο αντίστοιχων αισθητήρων, δίνει μια προσέγγιση της στροφής του ρομπότ ως προς το ΣΣ  $O_R$ . Τελικά, υπολογίζονται 3 γωνίες ως εξής:

$$\begin{aligned} \theta_{12} &= \text{atan2}((p1y - p2y), (p1x - p2x)) \\ \theta_{13} &= \text{atan2}((p1y - p3y), (p1x - p3x)) \\ \theta_{23} &= \text{atan2}((p2y - p3y), (p2x - p3x)) \end{aligned}$$

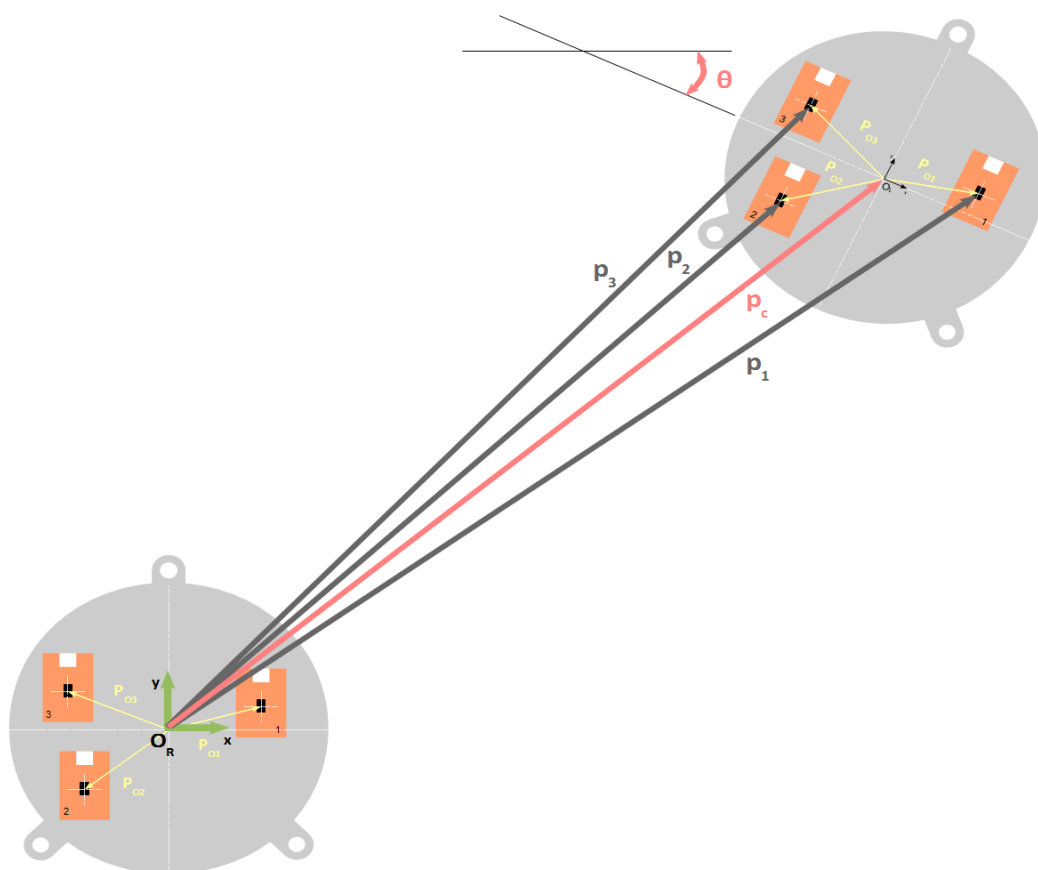
Όπου, 
$$\mathbf{p1} = \begin{bmatrix} p1x \\ p1y \end{bmatrix}, \mathbf{p2} = \begin{bmatrix} p2x \\ p2y \end{bmatrix}, \mathbf{p3} = \begin{bmatrix} p3x \\ p3y \end{bmatrix}$$

Τελικά, προκύπτουν οι 3 ακόλουθες εκτιμήσεις της γωνίας στροφής του σώματος ως προς το ΣΣ  $O_R$  (το εξωτερικό σύστημα που ορίζεται την χρονική στιγμή 0).

$$\theta\alpha = \theta_{12} - \theta_{12,0}$$

$$\theta\beta = \theta_{13} - \theta_{13,0}$$

$$\theta\gamma = \theta_{23} - \theta_{23,0}$$



**Σχήμα 5.20** Μετατοπίσεις ( $p1, p2, p3$ ) των αισθητήρων ως προς το εξωτερικό ΣΣ  $O_R$  και η εκτιμώμενη μετατόπιση και γωνία ( $p_c, \theta$ ) του σωματόδετου ΣΣ  $O_L$  ως προς το ΣΣ  $O_R$ .

Έχοντας υπολογίσει τις εκτιμώμενες θέσεις των 3 αισθητήρων καθώς και τις 3 εκτιμήσεις της γωνίας στροφής ως προς το εξωτερικό ΣΣ  $O_R$  (σωματόδετο του ρομπότ τη χρονική στιγμή μηδέν), οι αλγόριθμοι που μπορούν να εφαρμοσθούν για την εύρεση θέσης και γωνίας του ρομπότ είναι πολλοί και διαφέρουν ως προς την αποδοτικότητα, την ακρίβεια και την πολυπλοκότητά τους. Παρακάτω, θα αναφερθεί ο πλέον απλός αλγόριθμος και ύστερα θα προταθούν άλλοι τέσσερις αλγόριθμοι μεγαλύτερης πολυπλοκότητας και σε κάποιες περιπτώσεις μεγαλύτερης ακρίβειας.

### 5.15.2 Αλγόριθμος Μέσου Όρου

1. Από τις 3 εκτιμήσεις ( $\theta_\alpha$ ,  $\theta_\beta$ ,  $\theta_\gamma$ ) για την γωνία του σωματόδετου  $\Sigma\Sigma O_L$  ως προς το  $\Sigma\Sigma O_R$ , μία καλύτερη εκτίμηση μπορεί να προκύψει μέσω του υπολογισμού του μέσου όρου αυτών, ως εξής:

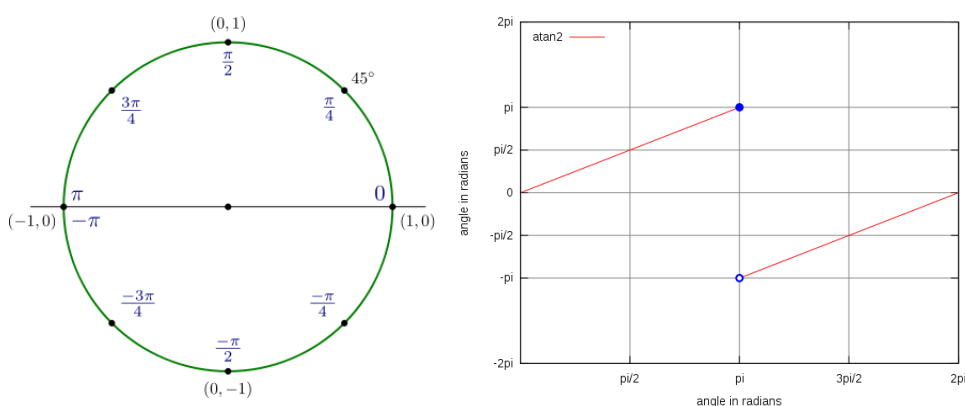
$$\theta = \frac{\theta_\alpha + \theta_\beta + \theta_\gamma}{3}$$

#### Πρόβλημα ασυνέχειας της atan2

Ένα πρόβλημα που προκύπτει σε αυτό το σημείο, αφορά στην συνάρτηση atan2 η οποία χρησιμοποιείται για τον υπολογισμό των τριών γωνιών. Αυτή η συνάρτηση υπολογίζει την αντίστροφη εφαπτομένη των δύο μεταβλητών  $x$  και  $y$ . Αυτό είναι όμοιο με τον υπολογισμό της αντίστροφης εφαπτομένης του  $y/x$ , εκτός του ότι τα πρόσημα και των δύο παραμέτρων χρησιμοποιούνται για τον υπολογισμό του τεταρτημορίου στο οποίο ανήκει το αποτέλεσμα. Η συνάρτηση επιστρέφει το αποτέλεσμα, το οποίο βρίσκεται μεταξύ του  $-\pi$  και του  $\pi$  (συμπεριλαμβανομένων των άκρων), σε rad. Όπως φαίνεται και στο Σχήμα 5.21, υπάρχει μια ασυνέχεια στο πεδίο τιμών της στο σημείο  $\pi/-\pi$ .

Το πρόβλημα δημιουργείται τη στιγμή που, καθώς η γωνία του σώματος αυξάνει, για παράδειγμα προς το  $\pi$ , μια εκτιμώμενη γωνία (από δύο αισθητήρες) ξεπεράσει πρώτη την τιμή  $\pi$ . Η αλλαγή της τιμής της γωνίας σε αυτή την περίπτωση είναι πολύ μεγάλη λόγω της ασυνέχειας της atan2 στο σημείο αυτό. Όταν τη στιγμή αυτή, γίνει ο υπολογισμός του μέσου όρου, όπως περιγράφεται παραπάνω, η γωνία που θα προκύψει σαν αποτέλεσμα θα είναι λανθασμένη. Αυτό βέβαια δεν θα αποτελούσε πρόβλημα αν όλες οι γωνίες είχαν περάσει το σημείο αυτό.

Αυτό εξηγείται με ένα απλό παράδειγμα: Αν γινόταν λόγος για υπολογισμό μέσου όρου 2 γωνιών όπως αυτές προκύπτουν από υπολογισμούς μέσω της atan2, και η μία είχε τιμή  $\pi$ , ενώ η άλλη  $-\pi$ , ο μέσος όρος τους θα ήταν γωνία 0, αντί για  $\pi$  ή  $-\pi$ . Δεδομένου ότι αντίστοιχο λάθος συμβαίνει και στην περίπτωση του μέσου όρου των τριών γωνιών στο σημείο  $\pi$ , πρέπει να αναπτυχθεί κατάλληλος αλγόριθμος που θα το αντιμετωπίζει.



**Σχήμα 5.21** Αριστερά: Οι γωνίες όπως επιστρέφονται από την συνάρτηση atan2( $x,y$ ). Δεξιά: Το πρόβλημα ασυνέχειας στο σημείο  $\pi/-\pi$ .

Ο αλγόριθμος που επινοήθηκε αποτελείται από 2 απλά βήματα, με τα οποία όταν οι εκτιμώμενες γωνίες βρεθούν κοντά στη τιμή  $\pi$ , οι τιμές τους μετασχηματίζονται στο εύρος  $[0, 2\cdot\pi)$ . Έτσι αναιρείται η ασυνέχεια στο σημείο  $\pi$  και δεν υπάρχει πρόβλημα στον υπολογισμό του μέσου όρου. Πιο αναλυτικά: Αν οι εκτιμώμενες γωνίες ( $\theta_\alpha$ ,  $\theta_\beta$ ,  $\theta_\gamma$ )  $\alpha$  παρουσιάσουν διαφορετικό πρόσημο και  $\beta$ ) είναι κοντά στο  $\pi$  κατά απόλυτη τιμή (για να μην εφαρμοστεί και στη περίπτωση που είναι κοντά στο

0, όπου η συνάρτηση δεν παρουσιάζει ασυνέχεια), τότε οι γωνίες  $\theta_\alpha, \theta_\beta, \theta_\gamma$  μετασχηματίζονται στο  $[0, 2\cdot\pi)$  ως εξής:

$$\begin{aligned}\theta_{\alpha,\beta,\gamma} &= \theta_{\alpha,\beta,\gamma} + 2\cdot\pi \\ \theta_{\alpha,\beta,\gamma} &= \text{rem}(\theta_{\alpha,\beta,\gamma}, 2\cdot\pi)\end{aligned}$$

Σημειώνεται ότι η συνάρτηση  $\text{rem}()$  στο περιβάλλον Matlab είναι η γνωστή συνάρτηση υπολοίπου  $\text{mod}$ .

Αν αυτός ο μετασχηματισμός γινόταν ανεξάρτητα της γωνίας του σώματος (δηλαδή και για γωνίες κοντά στο 0), τότε το πρόβλημα της ασυνέχειας θα μεταφερόταν απλά στο σημείο 0, και ο αλγόριθμος αυτός δεν θα είχε αξία.

2. Μετά τον υπολογισμό του μέσου όρου  $\theta$  των γωνιών, μπορούν να υπολογιστούν 3 προσεγγίσεις των συντεταγμένων του κέντρου του ρομπότ, μία για κάθε αισθητήρα. Για να γίνει αυτό, πρέπει να υπολογιστούν τα τρέχοντα διανύσματα θέσης των αισθητήρων  $\mathbf{p01}, \mathbf{p02}, \mathbf{p03}$  ως προς το εξωτερικό ΣΣ  $O_R$ , μετά δηλαδή από την περιστροφή τους κατά γωνία  $\theta$ . Πολλαπλασιάζοντας τα 3 διανύσματα με τον κατάλληλο πίνακα περιστροφής παίρνουμε 3 νέα διανύσματα, ως εξής:

$$\mathbf{p0n}' = \mathbf{R} \mathbf{p0n} \quad n = 1,2,3$$

Όπου,

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Σύμφωνα με το Σχήμα 5.20, οι συντεταγμένες του κέντρου προκύπτουν, αν από το διάνυσμα θέσης κάθε αισθητήρα  $\mathbf{pn}$  αφαιρεθεί το αντίστοιχο  $\mathbf{p0n}'$  που μόλις υπολογίστηκε. Επομένως, οι 3 προσεγγίσεις των συντεταγμένων του κέντρου του ρομπότ είναι:

$$\mathbf{pcenter}, n = \mathbf{pn} - \mathbf{p0n}' \quad n = 1,2,3$$

$$\mathbf{pcenter}, 1 = \begin{bmatrix} pc1x \\ pc1y \end{bmatrix}, \quad \mathbf{pcenter}, 2 = \begin{bmatrix} pc2x \\ pc2y \end{bmatrix}, \quad \mathbf{pcenter}, 3 = \begin{bmatrix} pc3x \\ pc3y \end{bmatrix}$$

3. Υπολογίζοντας τον μέσο όρο των συνιστωσών των 3 αυτών διανυσμάτων, προκύπτει η τελική εκτίμηση των συντεταγμένων του κέντρου του ρομπότ.

$$x_{Robot} = \frac{pc1x + pc2x + pc3x}{3}, \quad y_{Robot} = \frac{pc1y + pc2y + pc3y}{3}$$

Ο αλγόριθμος που μόλις παρουσιάστηκε, χρησιμοποιήθηκε εκτενώς στα πειράματα που έγιναν στο εργαστήριο και αποδείχθηκε λειτουργικός με αρκετά ακριβή αποτελέσματα. Εντούτοις, σε περίπτωση που ένας από τους αισθητήρες παρουσιάσει κάποια στιγμή μεγάλο σφάλμα, ο αλγόριθμος αυτός παρουσιάζει σημαντικά λάθη από εκείνη τη στιγμή και μετά. Έτσι, παρουσιάζεται η ανάγκη για ανάπτυξη πιο 'έξυπνων' αλγορίθμων, ικανών να εξαλείφουν πιθανά μεγάλα σφάλματα των αισθητήρων. Δύο τέτοιοι αλγόριθμοι παρουσιάζονται παρακάτω. Η τεχνική που χρησιμοποιείται και στους δύο, είναι η ανίχνευση του αισθητήρα που παρουσίασε το μεγάλο σφάλμα και η εύρεση των απαραίτητων στοιχείων μέσω των άλλων δύο ακριβέστερων αισθητήρων.

### 5.15.3 Αλγόριθμος Ακριβέστερης Απόστασης

Βασίζεται στη σύγκριση των αποστάσεων που φαίνονται να έχουν μεταξύ τους οι αισθητήρες στις εκτιμώμενες θέσεις τους, με τις πραγματικές τους αποστάσεις. Από τις 3 αυτές αποστάσεις επιλέγεται η ακριβέστερη, η οποία οδηγεί σε 2 από τους 3 αισθητήρες. Οι 2 αυτοί θεωρούνται οι πιο αξιόπιστοι για την τρέχουσα χρονική στιγμή, και γι' αυτό τους δίνεται αυξημένη βαρύτητα κατά τον υπολογισμό της γωνίας και της θέσης του ρομπότ. Πιο αναλυτικά τα βήματα είναι τα εξής:

1. Υπολογισμός των 3 μεταξύ των αισθητήρων αποστάσεων με βάση τις εκτιμώμενες θέσεις τους. Αυτό επιτυγχάνεται με αφαίρεση των διανυσμάτων  $\mathbf{p}_n$  ανά ζεύγη, και υπολογίζοντας το μέτρο των 3 διανυσμάτων που προκύπτουν.
2. Υπολογισμός των διαφορών των αποστάσεων αυτών από τις αντίστοιχες γνωστές πραγματικές αποστάσεις.
3. Η απόσταση με το μικρότερο σφάλμα οδηγεί στους 2 πιο αξιόπιστους αισθητήρες για την τρέχουσα επανάληψη. Αντί λοιπόν του απλού μέσου όρου, υπολογίζεται ο σταθμισμένος μέσος όρος των 3 γωνιών, δίνοντας ένα συντελεστή βαρύτητας  $w_{big}$  στην αντίστοιχη γωνία ως εξής:

$$\theta = \frac{w_{big} \cdot \theta_a + w_{small} \cdot \theta_b + w_{small} \cdot \theta_c}{3}$$

Όπου,  $w_{big} \in [1,3]$  και  $w_{small} = (3-w_{big})/2$

Εδώ σημειώνεται ότι, εμφανίζεται πάλι το πρόβλημα της ασυνέχειας της atan2 για γωνίες κοντά στο  $\pi$ . Επομένως, η ίδια λύση που περιγράφηκε στον αλγόριθμο του μέσου όρου, πρέπει να εφαρμοσθεί και εδώ για τον σωστό υπολογισμό του  $\theta$ .

Τα βήματα από εδώ και κάτω είναι ίδια με αυτά του αλγορίθμου μέσου όρου που παρουσιάστηκε παραπάνω. Η διαφορετικότητα λοιπόν αυτού του αλγορίθμου συνίσταται στον υπολογισμό της γωνίας δίνοντας βάρος στους 2 αισθητήρες που κρίθηκαν αξιόπιστοι στην τρέχουσα επανάληψη. Η αξιοπιστία των αισθητήρων κρίνεται με βάση το ποσοστό που έχει διατηρηθεί η γεωμετρία τους. Η αποτελεσματικότητα και η ακρίβεια αυτού του αλγορίθμου φαίνεται στα πειράματα που παρουσιάζονται σε επόμενο κεφάλαιο.

#### 5.15.4 Αλγόριθμος ενδιάμεσης γωνίας

Βασίζεται στην εύρεση της ενδιάμεσης γωνίας από τις 3 εκτιμώμενες γωνίες ( $\theta_a$ ,  $\theta_b$ ,  $\theta_c$ ), αν τις διατάξουμε σε αύξουσα ή φθίνουσα σειρά. Στην γωνία αυτή είναι που δίνεται βάρος κατά τον υπολογισμό του μέσου όρου των γωνιών. Με αυτό τον τρόπο, επιχειρείται η εξάλειψη ακραίων τιμών, οι οποίες μπορεί να οφείλονται σε μεγάλα στιγμιαία μετρητικά σφάλματα. Τα βήματα είναι τα εξής:

1. Όταν υπολογιστούν οι 3 εκτιμήσεις  $\theta_a$ ,  $\theta_b$ ,  $\theta_c$  της γωνίας του σώματος, αυτές τοποθετούνται κατά αύξουσα ή φθίνουσα σειρά. Η ενδιάμεση τιμή από αυτές θεωρείται η πιο αξιόπιστη κάθε φορά και σε αυτή δίνεται βάρος στον υπολογισμό της γωνίας του σώματος.
2. Αντί του απλού μέσου όρου, υπολογίζεται ο σταθμισμένος μέσος όρος των 3 γωνιών, δίνοντας ένα συντελεστή βαρύτητας  $w_{big}$  στην αντίστοιχη γωνία ως εξής:

$$\theta = \frac{w_{big} \cdot \theta_a + w_{small} \cdot \theta_b + w_{small} \cdot \theta_c}{3}$$

Όπου,  $w_{big} \in [1,3]$  και  $w_{small} = (3-w_{big})/2$

Σημειώνεται ότι, εμφανίζεται πάλι το πρόβλημα της ασυνέχειας της atan2 για γωνίες κοντά στο  $\pi$ . Επομένως, η ίδια λύση που περιγράφηκε στον αλγόριθμο του μέσου όρου, πρέπει να εφαρμοσθεί και εδώ για τον σωστό υπολογισμό του  $\theta$ .

Τα βήματα από εδώ και κάτω είναι ίδια με αυτά του αλγορίθμου μέσου όρου που παρουσιάστηκε παραπάνω. Η διαφορετικότητα λοιπόν αυτού του αλγορίθμου συνίσταται στον υπολογισμό της γωνίας δίνοντας βάρος στην μεσαία κατά μέτρο γωνία στην τρέχουσα επανάληψη, με στόχο την

απόσβεση ακραίων τιμών. Η αποτελεσματικότητα και η ακρίβεια αυτού του αλγορίθμου φαίνεται στα πειράματα που παρουσιάζονται σε επόμενο κεφάλαιο.

### 5.15.5 Αλγόριθμοι ανακατασκευής τρίτου αισθητήρα

Σαν βάση αυτών των αλγορίθμων, χρησιμοποιούνται οι δύο τελευταίοι αλγόριθμοι, ακριβέστερης απόστασης και ενδιάμεσης γωνίας. Μέσω αυτών των αλγορίθμων, σε κάθε επανάληψη, 2 αισθητήρες χαρακτηρίζονται ως αξιόπιστοι, ενώ ο τρίτος ως λιγότερο αξιόπιστος. Η πληροφορία αυτή εντούτοις, χρησιμοποιείται μόνο στον υπολογισμό της γωνίας του σώματος, και επομένως έχει σχετικά μικρή επιρροή στο τελικό αποτέλεσμα.

Στην παράγραφο αυτή, προτείνονται δύο παραλλαγές των αλγορίθμων αυτών, στις οποίες η πληροφορία των αξιόπιστων αισθητήρων χρησιμοποιείται επιπρόσθετα για την ανακατασκευή του λιγότερο αξιόπιστου αισθητήρα. Το κριτήριο αξιοπιστίας των αισθητήρων είναι διαφορετικό για τους αλγορίθμους ακριβέστερης απόστασης και ενδιάμεσης γωνίας, αλλά αυτό δεν αλλάζει τα επόμενα βήματα που προτείνονται εδώ. Τα ακόλουθα εφαρμόζονται με παρόμοιο τρόπο στους δύο αυτούς αλγορίθμους.

Τα βήματα είναι τα ίδια μέχρι τον υπολογισμό της γωνίας  $\theta$  με χρήση του σταθμισμένου μέσου των  $\theta_a$ ,  $\theta_b$ ,  $\theta_g$  και για τους δύο αλγορίθμους. Ύστερα, με βάση τη γωνία  $\theta$  και τις θέσεις των 2 αισθητήρων υπολογίζεται εύκολα η θέση του τρίτου αισθητήρα. Το τελευταίο γίνεται έτσι, ώστε να παραλειφθεί η εσφαλμένη μέτρηση και να μην επηρεάσει τις επόμενες επαναλήψεις (εκτός από την επιρροή στον υπολογισμό της γωνίας). Η διαδικασία αυτή γίνεται με κατάλληλη περιστροφή και πρόσθεση διανυσμάτων (περιγράφεται με κατάλληλα σχόλια επί του κώδικα που υπάρχει στο CD της εργασίας). Τέλος, βρίσκονται 3 εκτιμήσεις των συντεταγμένων του ρομπότ, μία για κάθε αισθητήρα, και σαν τελικό αποτέλεσμα υπολογίζεται ο μέσος όρος των εκτιμήσεων αυτών.

Ο αλγόριθμος αυτός αναμένεται να δώσει καλύτερα αποτελέσματα από τους προηγούμενους σε κάποιες ακραίες συνθήκες. Η εφαρμογή του αλγορίθμου, η παρουσίαση των αποτελεσμάτων και η αξιολόγηση τους γίνονται σε επόμενο κεφάλαιο, και παρουσιάζουν μεγάλο ενδιαφέρον.

### 5.16 Συνδυασμός με δεδομένα από το σύστημα της κάμερας

Στην παράγραφο αυτή παρουσιάζεται, για λόγους πληρότητας, η βάση του αλγορίθμου συνδυασμού των δεδομένων κίνησης που λαμβάνει το ρομπότ από τα δύο συστήματα εκτίμησης θέσης. Ο αλγόριθμος αυτός δεν υλοποιήθηκε στα πλαίσια της εργασίας αυτής. Το θέμα αυτό (Sensor Fusion) αποτελεί από μόνο του ένα μεγάλο κεφάλαιο, και χρήζει ενδελεχούς μελέτης. Αντίστοιχη μελέτη πάνω στην συγκεκριμένη εφαρμογή γίνεται στο [12].

Μερικά σημαντικά σημεία που πρέπει να ληφθούν υπόψη είναι τα εξής. Πρώτον, το σύστημα εντοπισμού της κάμερας παρέχει δεδομένα στο ρομπότ με αρκετά μικρότερη συχνότητα από ότι το σύστημα των οπτικών αισθητήρων. Δεύτερον, λόγω της ασύρματης επικοινωνίας και του υπολογιστικού βάρους της επεξεργασίας εικόνας, τα δεδομένα από την κάμερα φθάνουν στο ρομπότ με κάποια καθυστέρηση. Αυτό σημαίνει ότι η θέση και ο προσανατολισμός που λαμβάνει το ρομπότ κάποια χρονική στιγμή από το σύστημα αυτό, αντιστοιχούν στη θέση και στον προσανατολισμό που είχε αυτό μια προηγούμενη χρονική στιγμή. Η χρονική διαφορά των δύο αυτών στιγμών είναι ίση με τον χρόνο επεξεργασίας της εικόνας συν τον χρόνο αποστολής της πληροφορίας.

Οι τρόποι αντιμετώπισης του προβλήματος αυτού ποικίλουν. Στο σημείο αυτό, θεωρείται δεδομένο ότι τη στιγμή της λήψης των δεδομένων από το σύστημα της κάμερας, γίνεται κατάλληλη αναγωγή αυτών στο παρόν, οπότε και μπορούν να αξιοποιηθούν. Με βάση αυτή λοιπόν την παραδοχή, ο συνδυασμός των δεδομένων των δύο συστημάτων εύρεσης θέσης, μπορεί να αναλυθεί στα παρακάτω βήματα. Η ονοματολογία των μεταβλητών και των συστημάτων συντεταγμένων αναφέρεται στο Σχήμα 5.22.

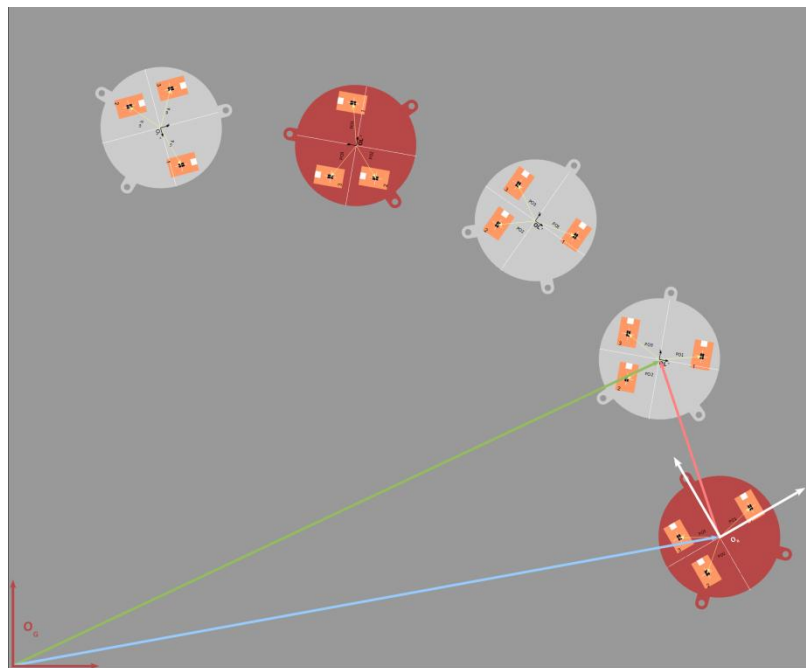
1. Τη χρονική στιγμή  $t_1$  από το σύστημα της κάμερας λαμβάνονται  $xRobot$ ,  $yRobot$ ,  $\theta_{R}$  ως προς το παγκόσμιο ΣΣ  $O_G$ . Αυτά ορίζουν το ΣΣ  $O_R$ , ως προς το οποίο θα υπολογιστούν οι συντεταγμένες

του κέντρου και η γωνία του ρομπότ από τους οπτικούς αισθητήρες τις επόμενες χρονικές στιγμές, μέχρι να ληφθούν νέα δεδομένα από τη κάμερα.

2. Τη χρονική στιγμή  $t_2$  από το σύστημα των οπτικών αισθητήρων λαμβάνονται το διάνυσμα θέσης του κέντρου  $p_c$  και η γωνία  $\theta$  του ρομπότ ως προς το ΣΣ  $O_R$  που ορίστηκε τη στιγμή  $t_1$ . Τα  $p_c$  και  $\theta$  μεταφέρονται μέσω του ΣΣ  $O_R$  στο παγκόσμιο ΣΣ  $O_G$  και έτσι προκύπτουν τα νέα  $x_{Robot}$ ,  $y_{Robot}$ ,  $\theta_{thetaR}$ .

3. Στη συνέχεια, πριν νέα δεδομένα από τη κάμερα γίνουν διαθέσιμα, θέση και γωνία βρίσκονται όπως στο βήμα 2, κάθε φορά που λαμβάνονται δεδομένα από τους οπτικούς αισθητήρες. Υπολογίζονται δηλαδή το διάνυσμα θέσης του κέντρου  $p_c$  και η γωνία  $\theta$  του ρομπότ ως προς το ΣΣ  $O_R$  που ορίστηκε τη στιγμή  $t_1$ . Τα  $p_c$  και  $\theta$  μεταφέρονται μέσω του ΣΣ  $O_R$  στο παγκόσμιο ΣΣ  $O_G$  και έτσι προκύπτουν τα νέα  $x_{Robot}$ ,  $y_{Robot}$ ,  $\theta_{thetaR}$ .

4. Όταν ληφθούν νέα δεδομένα από την κάμερα, η παραπάνω διαδικασία επαναλαμβάνεται αρχίζοντας πάλι από το πρώτο βήμα.



**Σχήμα 5.22** Λήψη δεδομένων από το σύστημα της κάμερας (κόκκινο στιγμιότυπο) και από το σύστημα των αισθητήρων (γκρι στιγμιότυπο).



# ΚΕΦΑΛΑΙΟ 6

## Σχεδιασμός λογισμικού (Software Design)

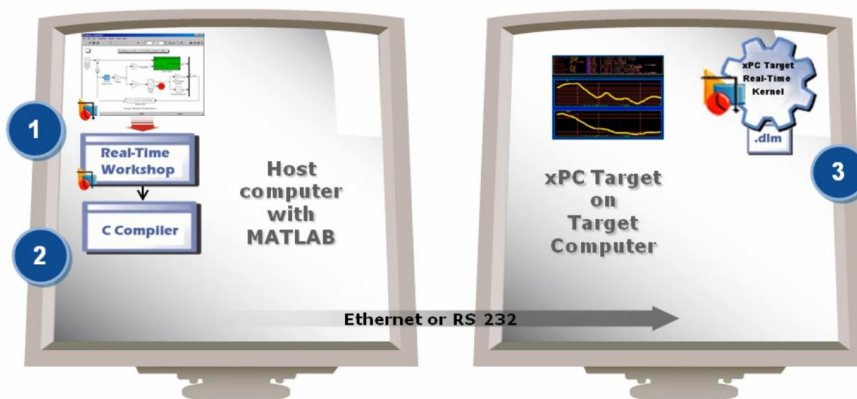
### 6.1 Εισαγωγή

Στο κεφάλαιο αυτό, περιγράφεται η διαδικασία προγραμματισμού του ρομποτικού συστήματος. Εδώ φαίνονται οι ευκολίες που παρέχει το πακέτο xPC Target, στον προγραμματισμό. Όπως έχει ήδη αναφερθεί, σε αντίθεση με τον παραδοσιακό τρόπο προγραμματισμού ενσωματωμένων συστημάτων, ο σχεδιασμός βασισμένος στο μοντέλο (model-based design) συνδέει άρρηκτα τον σχεδιασμό των αλγορίθμων με την ανάπτυξη των προγραμμάτων και την εκτέλεση τους στο τελικό υπολογιστικό σύστημα του ρομπότ. Μέσω του σχεδιασμού απλών μοντέλων Simulink (που χωρίς το xPC Target θα ήταν απλά άλλη μία εξομοίωση), μπορεί να δοκιμαστεί οποιαδήποτε αλγοριθμική ιδέα κατευθείαν στο τελικό ρομπότ, με την μορφή προγράμματος C αυστηρά πραγματικού χρόνου (hard real-time). Ακολουθεί μια περιγραφή του λογισμικού που αναπτύχθηκε και γενικά της διαδικασίας ανάπτυξης του ρομποτικού συστήματος στο πλαίσιο του xPC Target.

## 6.2 Εγκατάσταση πακέτου xPC Target και αρχικές ρυθμίσεις

### 6.2.1 Υπολογιστές Host και Target

Όπως έχει αναφερθεί ξανά, η χρήση του πακέτου xPC Target προϋποθέτει την ύπαρξη δύο υπολογιστών (βλ. Σχήμα 6.1), ενός host που τρέχει Matlab, και ενός target υπολογιστή που αποτελεί το υπολογιστικό σύστημα της τελικής εφαρμογής. Έχει περιγραφεί προηγουμένως η διαδικασία επιλογής του υπολογιστή target (κάρτα CPU PC-104), ενώ σε αυτό το σημείο είναι απαραίτητη η επιλογή και η κατάλληλη ρύθμιση ενός υπολογιστή host, στον οποίο θα γίνει όλη η ουσιαστική εργασία.



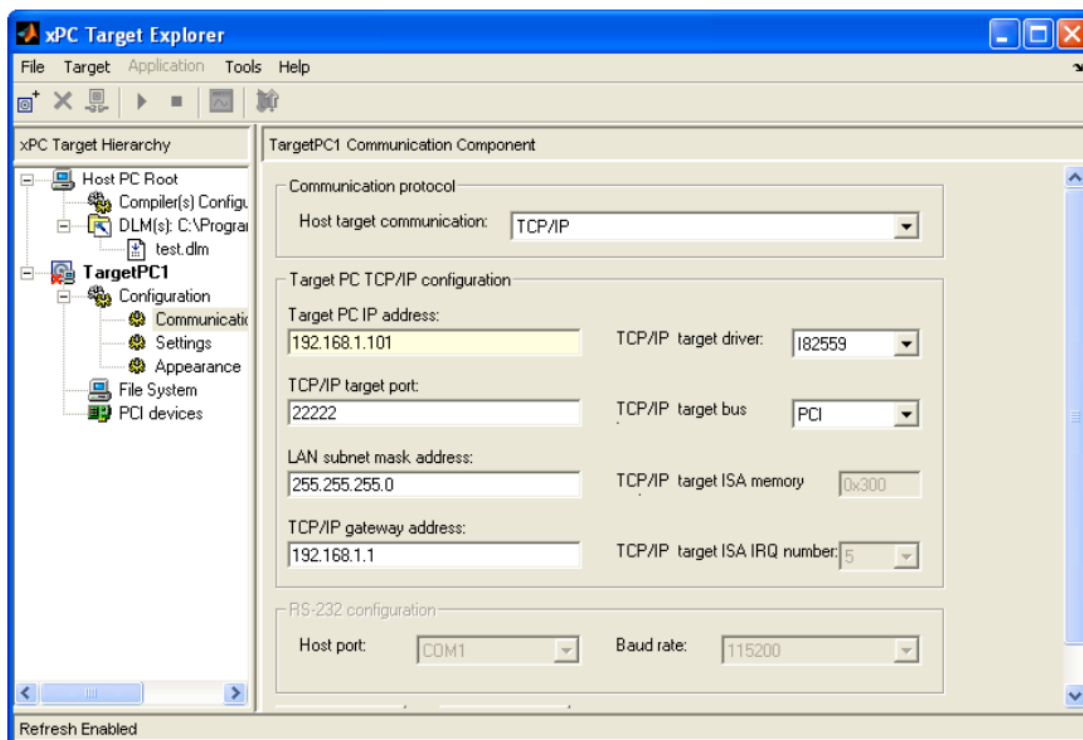
Σχήμα 6.1 Οι υπολογιστές host και target.

Ο διαθέσιμος υπολογιστής γι' αυτόν τον σκοπό στο εργαστήριο ήταν ένα laptop MacBook Pro. Λόγω όμως προβλημάτων συμβατότητας των Mac OS με το πακέτο xPC Target, χρησιμοποιήθηκε η εικονική μηχανή VirtualBox για την ανάπτυξη του λογισμικού. Στο εικονικό αυτό περιβάλλον, εγκαταστάθηκε το λειτουργικό σύστημα Windows XP 32Bit, και όλα τα απαραίτητα προγράμματα σύμφωνα με την επίσημη σελίδα του xPC Target (π.χ. Real-Time Workshop, Microsoft Visual C++ Express 2010 κ.α.) [20]. Σημειώνεται, ότι τα προγράμματα αυτά [21] πρέπει να εγκατασταθούν με τη σειρά που προτείνεται από την Mathworks, αλλιώς δημιουργούνται προβλήματα κατά τη λειτουργία του xPC Target.

### 6.2.2 Αρχικές Ρυθμίσεις xPC Target

Αναλυτικές οδηγίες για εγκατάσταση και σωστή ρύθμιση του περιβάλλοντος xPC Target υπάρχουν στα [46] [47]. Τα βήματα είναι αρκετά, αλλά η διαδικασία απλή, και για τον λόγο αυτό εδώ θα αναφερθούν μόνο μερικά βασικά σημεία αυτής.

Οι ρυθμίσεις μπορούν να γίνουν πολύ εύκολα μέσω ενός ειδικού γραφικού περιβάλλοντος (GUI), το 'xPC Target Explorer' (βλ. Σχήμα 6.2). Σε αυτό αρχικά επιλέγεται ο τρόπος επικοινωνίας host και target υπολογιστή (serial ή network). Εδώ κρίθηκε πιο ευέλικτη και γρήγορη η επικοινωνία μέσω Ethernet, η οποία και επιλέχθηκε. Μεταξύ άλλων στοιχείων σχετικά με την επικοινωνία μέσω Ethernet, εισάγεται απαραίτητα η διεύθυνση IP (IP address) του Target PC και το μοντέλο του Ethernet chipset που χρησιμοποιεί ο υπολογιστής target (για να φορτωθούν οι κατάλληλοι οδηγίες). Η διεύθυνση IP που δόθηκε στον target υπολογιστή του ρομπότ είναι η: 192.168.1.90. Το επόμενο βήμα είναι η δημιουργία μιας συσκευής εκκίνησης (Boot device) για το Target PC.



Σχήμα 6.2 Γραφικό περιβάλλον xPC Target Explorer.

### 6.2.3 Κτίσιμο Boot CD/USB

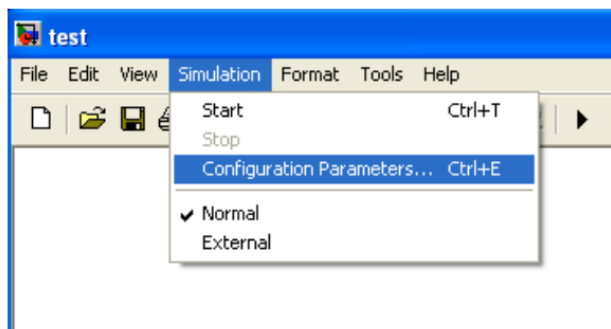
Υπάρχουν αρκετές δυνατότητες σχετικά με τον τρόπο εκκίνησης του target υπολογιστή (ρομπότ). Στο πλαίσιο αυτής της εργασίας δοκιμάστηκαν δύο τρόποι: με Boot CD, και με USB flash drive. Έγιναν επίσης προσπάθειες να υλοποιηθεί το ιδανικό σενάριο εκκίνησης μέσω δικτύου (LAN Boot), οι οποίες όμως δεν απέδωσαν καρπούς. Κατά τις πρώτες δοκιμές, η εκκίνηση γινόταν μέσω CD, η διαδικασία κτίσης του οποίου περιγράφεται στο [46]. Λόγω του όγκου του CD Drive που απαιτεί αυτός ο τρόπος εκκίνησης, ο τρόπος που επιλέχτηκε τελικά, ήταν αυτός μέσω USB. Η διαδικασία για την δημιουργία του USB εκκίνησης είναι πολύ απλή και περιγράφεται επίσης στο [46].

### 6.2.4 Τηλεχειρισμός – Επικοινωνία με το target PC

Υπάρχουν αρκετοί τρόποι για την επικοινωνία του host και του target υπολογιστή (host target link). Δεν πρόκειται για κάποια real time επικοινωνία ή ανταλλαγή δεδομένων. Είναι μια επικοινωνία αφιερωμένη στην εποπτεία και τον χειρισμό του συστήματος. Μέσω αυτής επιτρέπονται οι εξής λειτουργίες: εκκίνηση και τερματισμός της εφαρμογής, ρύθμιση παραμέτρων του συστήματος ενόσω τρέχει το πρόγραμμα (on-the-fly), φόρτωση και εκτέλεση διαφορετικών εφαρμογών σε χρόνο λίγων λεπτών, παρουσίαση δεδομένων σε μορφή διαγραμμάτων στο host και στο target PC την στιγμή που τρέχει το πρόγραμμα, αποθήκευση δεδομένων στο host ή στο target PC για μετέπειτα επεξεργασία, απεικόνιση αποτελεσμάτων μέσω 3D κόσμου εικονικής πραγματικότητας (VRML), και πολλές ακόμη.

Οι τρόποι με τους οποίους πραγματοποιούνται τα παραπάνω είναι οι εξής: μέσω εντολών στην γραμμή εντολών Matlab, μέσω αυτοσχέδιου GUI (Graphical User Interface) ανεπτυγμένο σε Matlab, μέσω ειδικού GUI για το xPC Target που ονομάζεται 'xPC Target Explorer', και τέλος μέσω του ίδιου του μοντέλου Simulink σε 'external mode'. Οι τρόποι που χρησιμοποιήθηκαν και αξιολογήθηκαν κατά την ανάπτυξη του συστήματος, ήταν αρχικά το GUI 'xPC Target Explorer', και τελικά η χρήση των μοντέλων Simulink σε 'external mode' (βλ. Σχήμα 6.3). Το τελευταίο σημαίνει ότι, το μοντέλο επικοινωνεί απευθείας με το target PC και μπορεί να λειτουργήσει σαν γραφικό περιβάλλον. Το 'xPC Target Explorer' αποδείχτηκε πολύ χρηστικό, ειδικά κατά το στάδιο εξοικείωσης με το πακέτο, αλλά σύντομα προτιμήθηκε η μετατροπή των μοντέλων Simulink σε γραφικά περιβάλλοντα ελέγχου

('external mode'). Τα μοντέλα που παρουσιάζονται παρακάτω χρησιμοποιούν τον τελευταίο τρόπο επικοινωνίας.



Σχήμα 6.3 Normal και external mode.

### 6.3 Λογισμικό Συστήματος Οπτικών Αισθητήρων

Αρχικά παρουσιάζεται η διαδικασία προγραμματισμού του μικροελεγκτή Arduino και ύστερα ο τρόπος επικοινωνίας του συστήματος μικροελεγκτής-οπτικοί αισθητήρες με την κεντρική CPU του ρομπότ.

#### 6.3.1 Προγραμματισμός μικροελεγκτή Arduino

Το πιο μεγάλο πλεονέκτημα της πλατφόρμας Arduino είναι η ευκολία και η αμεσότητα στον προγραμματισμό του μικροελεγκτή. Παρέχεται ένα ειδικό και πολύ φιλικό περιβάλλον (IDE), που απλοποιεί τα περίπλοκα διαδικαστικά σημεία της ανάπτυξης ενός συστήματος μικροελεγκτή, και επιτρέπει την εστίαση στον προγραμματισμό. Η γλώσσα προγραμματισμού που χρησιμοποιείται μοιάζει σε μεγάλο βαθμό στη γλώσσα C++.

Για την συγκεκριμένη εφαρμογή (συλλογή δεδομένων από τρεις οπτικούς αισθητήρες), το πρόβλημα αντιμετωπίστηκε με δύο τρόπους. Στο πρώτο σενάριο, ο μικροελεγκτής συλλέγει τα δεδομένα από τους οπτικούς αισθητήρες, τα επεξεργάζεται και τελικά στέλνει στην κεντρική CPU το τελικό αποτέλεσμα (θέση/ προσανατολισμός του ρομπότ) μέσω σειριακής επικοινωνίας. Έτσι, το υποσύστημα μικροελεγκτής-οπτικοί αισθητήρες γίνεται μια αυτόνομη συσκευή εντοπισμού θέσης/ προσανατολισμού. Στο δεύτερο σενάριο, ο μικροελεγκτής συλλέγει τα δεδομένα και τα στέλνει ακατέργαστα στην κεντρική CPU μέσω σειριακής επικοινωνίας για επεξεργασία. Σε αυτή την περίπτωση, ο μικροελεγκτής απλώς συλλέγει και μεταφέρει τα δεδομένα κίνησης τα οποία χειρίζεται κατάλληλα το κεντρικό υπολογιστικό σύστημα του ρομπότ. Τα αντίστοιχα προγράμματα και για τα δύο σενάρια υλοποιήθηκαν και δοκιμάστηκαν με επιτυχία (υπάρχουν στο CD της εργασίας). Λόγω ευελιξίας και καλύτερου χειρισμού των δεδομένων, πιο πολύ χρησιμοποιήθηκε ο δεύτερος τρόπος, όπου ο μικροελεγκτής Arduino απλά συλλέγει και στέλνει ακατέργαστα τα δεδομένα. Αυτός είναι που παρουσιάζεται παρακάτω (βλ. Σχήμα 6.4). Χρησιμοποιείται η βιβλιοθήκη PS/2 όπως περιγράφηκε σε προηγούμενο κεφάλαιο.

```

thecode3_dx_dy | Arduino 0022
File Edit Sketch Tools Help
thecode3_dx_dy $
#include <PS2Mouse.h>
#define MOUSE_DATA 3
#define MOUSE_CLOCK 2

PS2Mouse mouse(MOUSE_CLOCK, MOUSE_DATA, STREAM);
PS2Mouse mouse2(4, 5, STREAM);
PS2Mouse mouse3(6, 7, STREAM);

void setup()
{
  Serial.begin(115200);
  mouse.initialize();
  mouse2.initialize();
  mouse3.initialize();
}

void loop()
{
  int data[3];
  mouse.report(data);
  int data2[3];
  mouse2.report(data2);
  int data3[3];
  mouse3.report(data3);

  Serial.print("dxa1");
  Serial.println(data[1]);
  Serial.print("dya1");
  Serial.println(data[2]);
  Serial.print("dxa2");
  Serial.println(data2[1]);
  Serial.print("dya2");
  Serial.println(data2[2]);
  Serial.print("dxa3");
  Serial.println(data3[1]);
  Serial.print("dya3");
  Serial.println(data3[2]);
}

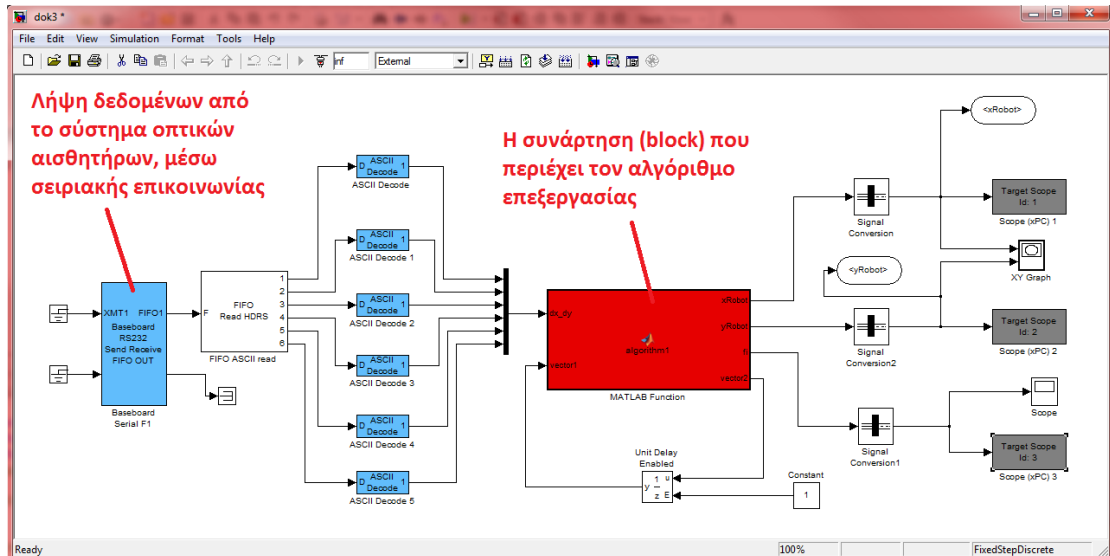
```

Σχήμα 6.4 Πρόγραμμα Arduino για συλλογή και αποστολή δεδομένων κίνησης.

### 6.3.2 Μοντέλο Simulink

Στην ενότητα αυτή παρουσιάζεται το μοντέλο Simulink, το οποίο αφού μετατραπεί αυτόματα σε πρόγραμμα C πραγματικού χρόνου, είναι υπεύθυνο για την παραλαβή των δεδομένων από το σύστημα του μικροελεγκτή μέσω σειριακής επικοινωνίας, και την μετέπειτα επεξεργασία τους. Με βάση τα παραδείγματα που παρέχει η Mathworks για σειριακή επικοινωνία, η επιλογή των κατάλληλων μπλοκς (blocks) και η κατασκευή του μοντέλου, έγιναν εύκολα και γρήγορα. Οι λειτουργίες των επιμέρους στοιχείων του μοντέλου γίνονται κατανοητές μέσω του Σχήματος 6.5 και μέσω της περιήγησης στο πραγματικό μοντέλο που συμπεριλαμβάνεται στο CD της εργασίας. Το αλγοριθμικό κομμάτι βρίσκεται στο κόκκινο μπλοκ 'Matlab function'. Αυτό είναι ουσιαστικά μια συνάρτηση Matlab στην οποία υλοποιείται ένας κάθε φορά αλγόριθμος επεξεργασίας, από αυτούς που παρουσιάστηκαν στο Κεφάλαιο 5. Στο CD της εργασίας υπάρχουν, τόσα τέτοια μοντέλα Simulink, όσοι και οι αλγόριθμοι του Κεφαλαίου 5. Σημειώνεται επίσης ότι τα μοντέλα αυτά έχουν αναπτυχθεί σε 'external mode', οπότε μέσω των μπλοκς 'XY Graph' και 'Scopes', είναι δυνατή η ζωντανή επίβλεψη του πειράματος στον υπολογιστή host μέσω κατάλληλων γραφημάτων. Τα 'target scopes' φαίνονται στον υπολογιστή target (ρομπότ) αν σε αυτόν συνδεθεί οθόνη υπολογιστή.

Το επόμενο βήμα είναι το τρέξιμο του προγράμματος στον υπολογιστή target (ρομπότ) σε πραγματικό χρόνο και γίνεται ως εξής: *Tools>External Mode Control Panel>Connect>Start Real-Time Code>Stop Real-Time Code*. Η απλή αυτή διαδικασία είναι σύντομη σε διάρκεια και ισχύει και για τα μοντέλα που ακολουθούν.



Σχήμα 6.5 Μοντέλο Simulink για παραλαβή και επεξεργασία των δεδομένων των οπτικών αισθητήρων.

## 6.4 Λογισμικό Συστήματος Κάμερας

### 6.4.1 Ασύρματη επικοινωνία με ρομπότ

Έχει ήδη αναφερθεί στο Κεφάλαιο 4 ότι το πρόγραμμα αυτό προϋπήρχε της εργασίας και ότι οι αλγοριθμικές αλλαγές έγιναν πάνω στον προϋπάρχοντα κορμό αυτού. Έγινε επίσης παρουσίαση των πρωτοκόλλων TCP/IP και UDP/IP δίνοντας έμφαση στα στοιχεία που αφορούν τη συγκεκριμένη εφαρμογή. Το συμπέρασμα που βγήκε ήταν η υπεροχή του πρωτοκόλλου UDP. Το πρόβλημα εδώ είναι ότι το πρόγραμμα C++ αποστολής των δεδομένων έχει ήδη προγραμματιστεί σε TCP/IP.

Αρχικά, επιχειρήθηκε να γίνει κατάλληλη μετατροπή στο πρόγραμμα. Το σχέδιο όμως αυτό αναβλήθηκε για δύο λόγους. Πρώτον, παρόλο που το UDP/IP αποτελεί πιο εύκολο και απλό πρωτόκολλο επικοινωνίας, αντιμετωπίστηκαν δυσκολίες στην υλοποίηση του αντίστοιχου αλγορίθμου, κυρίως λόγω έλλειψης εμπειρίας σε Socket programming. Εκτός αυτού, αν άλλαζε το πρωτόκολλο επικοινωνίας από τη μεριά του συστήματος της κάμερας, αντίστοιχη αλλαγή θα έπρεπε να γίνει και στην μεριά των υπολογιστών των δύο ρομπότ. Αυτό ήταν επιθυμητό για το δεύτερο ρομπότ, το οποίο είναι σε φάση εξέλιξης, αλλά όσον αφορά στο πρώτο ρομπότ, η μετατροπή TCP/IP σε UDP/IP έπρεπε να γίνει από τους υπεύθυνους γι' αυτό ερευνητές μηχανικούς της ομάδας.

Η λύση που βρέθηκε στα παραπάνω προβλήματα, με γνώμονα πάντα την γρήγορη ανάπτυξη του συστήματος, ήταν η εξής: Το υπεύθυνο για την αποστολή κομμάτι του κώδικα C++ παραμένει αμετάβλητο στην δομή του και σε ότι αφορά το πρωτόκολλο TCP/IP. Αυτό που αλλάζει στην περίπτωση μόνο του δεύτερου ρομπότ είναι ότι, τα δεδομένα δεν αποστέλλονται κατευθείαν στο υπολογιστικό σύστημα του ρομπότ, όπως π.χ. στο πρώτο. Αλλάζοντας την IP αποστολής σε localhost IP (121.11.123), τα δεδομένα στέλνονται πρώτα σε ένα μοντέλο simulink, το οποίο μέσω κατάλληλων blocks μετατρέπει το πρωτόκολλο σε UDP/IP. Πιο συγκεκριμένα, το μοντέλο αυτό παίρνει σαν είσοδο την πληροφορία από τον κώδικα C++ χρησιμοποιώντας TCP/IP, και την στέλνει στο υπολογιστικό σύστημα του δεύτερου ρομπότ χρησιμοποιώντας UDP/IP.

Πλεονεκτήματα αυτής της ιδέας, είναι η γρήγορη υλοποίηση της επικοινωνίας, οι ελάχιστες αλλαγές που έγιναν στον κοινό για τα δύο ρομπότ κώδικα C++, η καλή, αξιόπιστη και real-time επικοινωνία Simulink-xPC Target μέσω έτοιμων ειδικών blocks και η γενικότερη ευχρηστία που παρέχει το περιβάλλον Matlab (εύκολα γραφήματα, απεικόνιση μέσω 3d μοντέλων, γρήγορη και εύκολη αποθήκευση δεδομένων, ευκολία στην αλλαγή παραμέτρων και στον γενικότερο χειρισμό των συσκευών).

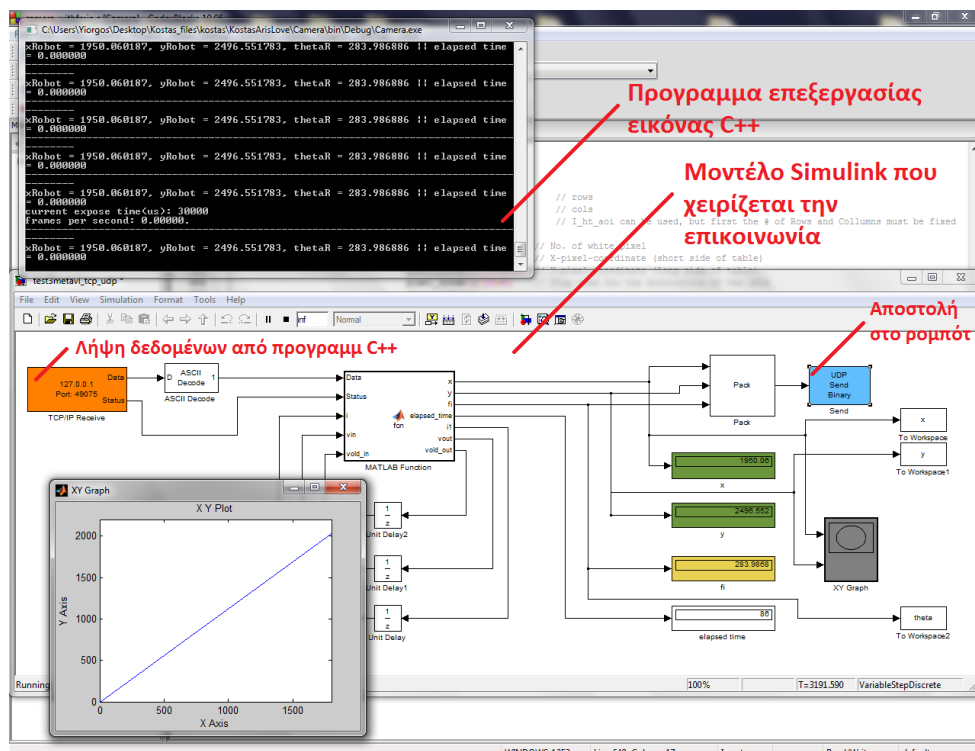
Μειονεκτήματα της ιδέας, είναι η επιπλέον χρήση του απαιτητικού υπολογιστικά προγράμματος Matlab-Simulink, η επιπρόσθετη καθυστέρηση της πληροφορίας με τη χρήση τριών κόμβων (πρόγραμμα C++, Simulink Model, xPC Target), η διαφορετικότητα των τρόπων επικοινωνίας των δύο ρομπότ και τέλος μερικές προγραμματιστικές δυσκολίες που εμφανίστηκαν σε επίπεδο Simulink.

### Συμπεράσματα

Ήταν ένα καλός τρόπος για να ξεπεραστούν γρήγορα τα προβλήματα που προέκυψαν λόγω ασυμβατότητας του xPC Target με το πρωτόκολλο TCP/IP, αλλά δεν αποτελεί σε καμία περίπτωση τον πιο γρήγορο και ιδανικό τρόπο επικοινωνίας για την συγκεκριμένη εφαρμογή. Όπως προτείνεται και στην ενότητα Μελλοντική Εργασία, θα ήταν καλό στο μέλλον να αλλάξει το πρωτόκολλο αποστολής του προγράμματος C++, και να γίνουν και οι απαιτούμενες τροποποιήσεις στο πρώτο ρομπότ, για ένα βέλτιστο δίκτυο επικοινωνίας. Τέλος, το μοντέλο Simulink μπορεί να χρησιμοποιείται σαν μέσο εποπτείας του συστήματος αναγνώρισης, μιας και ο χρήστης μπορεί να ελέγχει τις εξόδους του προγράμματος επεξεργασίας εικόνας C++ μέσω των εργαλείων που παρέχει η Mathworks.

### 6.4.2 Πρόγραμμα επεξεργασίας εικόνας

Το πρόγραμμα επεξεργασίας εικόνας έχει ήδη περιγραφεί στο Κεφάλαιο 4. Είναι αυτό που τρέχει στον εξωτερικό υπολογιστή που είναι συνδεδεμένος με την κάμερα. Στο θέμα της αποστολής των δεδομένων στο ρομπότ, δεν χρειάζεται να γίνει καμία δομική αλλαγή. Η διαδικασία λειτουργίας του προγράμματος είναι η εξής: πρώτα τρέχει το πρόγραμμα C++ και αναμένει απάντηση από κάπου πριν αρχίσει να στέλνει δεδομένα. Τα δεδομένα στέλνονται σε αυτό το πρόγραμμα που κατάφερε να συνδεθεί με το πρόγραμμα επεξεργασίας, οπότε δεν χρειάζεται στο κώδικα κάποια IP αποστολής (καθορίζεται αυτόματα από τον παραλήπτη). Ο παραλήπτης σε αυτή την περίπτωση είναι όχι ο υπολογιστής του ρομπότ, αλλά το ενδιαμέσο μοντέλο Simulink, όπως εξηγήθηκε και παραπάνω. Το περιβάλλον του υπολογιστή της κάμερας κατά τη λειτουργία του συστήματος φαίνεται στο Σχήμα 6.6 με όλες τις συνιστώσες που αναφέρθηκαν.

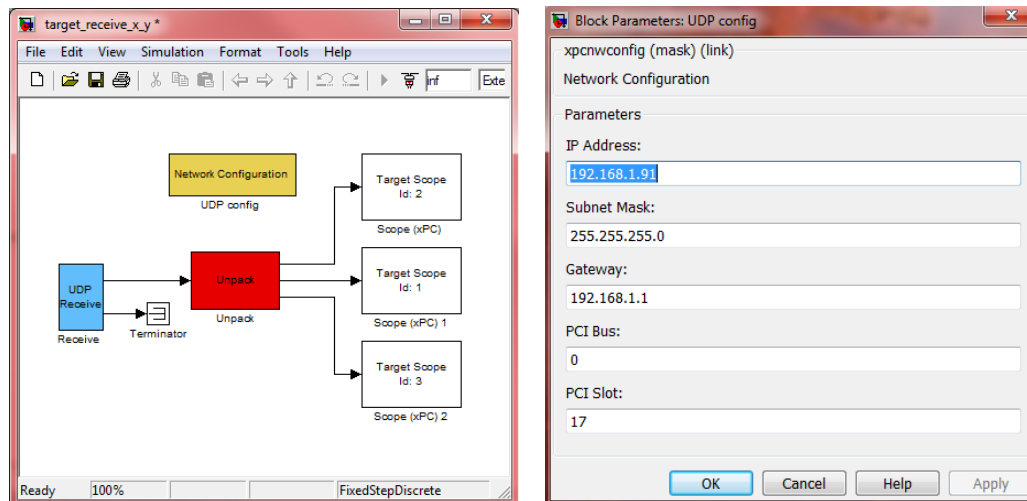


Σχήμα 6.6 Περιβάλλον του υπολογιστή της κάμερας με όλες τις συνιστώσες σε λειτουργία.



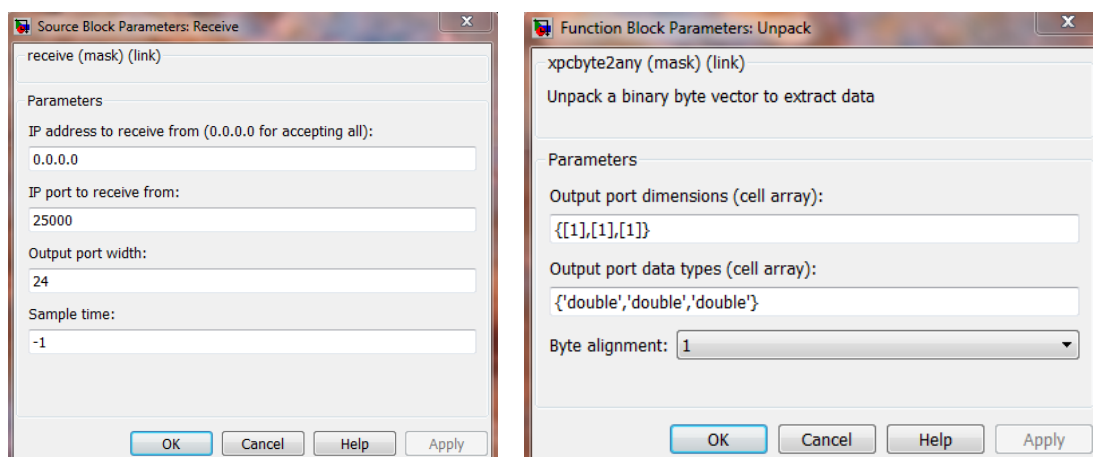
### 6.4.3 Μοντέλο Simulink

Εδώ παρουσιάζεται το μοντέλο Simulink που μετατρέπεται σε κώδικα C και τελικά τρέχει στο υπολογιστικό σύστημα του ρομπότ (target PC) σε πραγματικό χρόνο. Η λειτουργία του μοντέλου περιορίζεται στην επικοινωνία με το προηγούμενο 'ενδιάμεσο' μοντέλο Simulink μέσω του πρωτοκόλλου UDP/IP σε πραγματικό χρόνο. Τα μπλοκ που χρησιμοποιούνται στο μοντέλο μαζί με τις κατάλληλες ρυθμίσεις τους, παρουσιάζονται στα Σχήματα 6.7 και 6.8.



Σχήμα 6.7 Αριστερά: μοντέλο simulink. Δεξιά: ρυθμίσεις του μπλοκ Network Configuration.

Με το μπλοκ 'Network Configuration' γίνεται η αρχικοποίηση της επικοινωνίας μέσω UDP. Οι πληροφορίες λαμβάνονται στην τοπική διεύθυνση (local IP) 192.168.1.91, και PCI slot: 17. Αυτά σχετίζονται με την δεύτερη ασύρματη Ethernet γέφυρα (wireless Ethernet bridge), που χρησιμοποιείται ειδικά για να μπορεί αυτή η επικοινωνία να γίνεται σε πραγματικό χρόνο.



Σχήμα 6.8 Αριστερά: ρυθμίσεις του μπλοκ UDP Receive. Δεξιά: ρυθμίσεις του μπλοκ Unpack.



## 6.5 Εικονική Πραγματικότητα με VRML

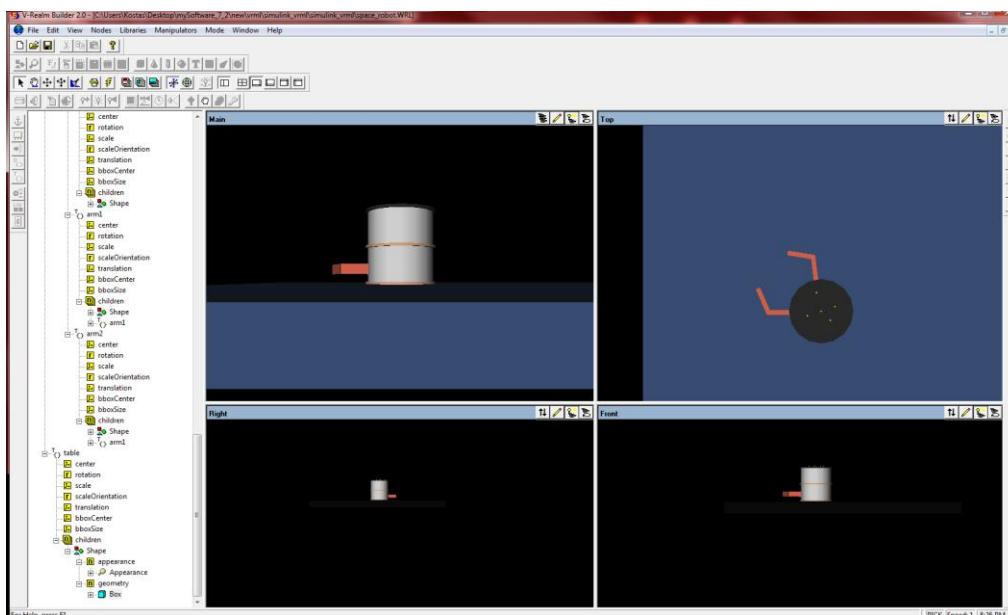
Στην ενότητα αυτή περιγράφεται ο τρόπος ανάπτυξης εικονικών κόσμων με τη βοήθεια της γλώσσας VRML (Virtual Reality Modeling Language). Είναι ένα πρότυπο δημιουργίας τρισδιάστατων γραφικών για τον παγκόσμιο ιστό. Αποτελεί την πρώτη ουσιαστική προσπάθεια για τη δημιουργία τρισδιάστατης γλώσσας για χρήση κυρίως στο διαδίκτυο. Παρουσιάστηκε το 1995 και βασίστηκε πάνω σε ένα υποσύνολο του 3D μοντέλου της Silicon Graphics. Οι εφαρμογές της γλώσσας είναι πολλές: διαφημίσεις στο διαδίκτυο, ψυχαγωγία, εκπαίδευση, αρχιτεκτονική, εικονογραφίες, εικονική πραγματικότητα κ.α. Παρουσιάζει ευκολία διασύνδεσης με λογισμικά προσομοιώσεις και έλεγχου σε πραγματικό χρόνο (π.χ. Simulink, xPC Target).

Είναι μια γλώσσα με την οποία μπορούν εύκολα να δημιουργηθούν απλά τρισδιάστατα σχήματα (κύβος, σφαίρα, κώνος), να καλυφθούν με χρωματιστές υφές και να ενσωματωθούν σε πιο σύνθετα αντικείμενα. Απλή είναι η μεταφορά, η περιστροφή, η μεγέθυνση/ σμίκρυνση κλπ. Τα αρχεία έχουν κατάληξη '.wrl'. Ένα αρχείο VRML αποτελείται από ένα δένδρο κόμβων. Τέτοιοι κόμβοι μπορεί να είναι κόμβοι σχήματος ή κόμβοι ιδιότητας. Ένας κόμβος (node) αποτελείται από ένα τύπο κόμβου και μια λίστα από πεδία (fields) και ιδιότητες (properties). Μπορούν να συνταχθούν με έναν οποιοδήποτε κειμενογράφο ή μπορούν να εξαχθούν από εφαρμογή τρισδιάστατης μοντελοποίησης (3D Studio Max, V-Realm Builder κ.α.).

### 6.5.1 Σχεδιασμός με V-Realm Builder

Είναι ένα πρόγραμμα μοντελοποίησης εικονικών κόσμων [48]. Το V-Realm Builder και η VRML, δεν δημιουργήθηκαν για να αντικαταστήσουν τα σύγχρονα εργαλεία μοντελοποίησης, που μπορούν να δημιουργήσουν ρεαλιστικά αντικείμενα με εκατομμύρια πολύγωνα. Το πλεονέκτημα αυτών είναι η ελαχιστοποίηση του μεγέθους σύνθετων αρχείων. Ενδείκνυται η επικοινωνία με Matlab, ενώ το V-Realm Builder εμπεριέχεται ήδη στο πακέτο του. Πρέπει να εγκατασταθεί πληκτρολογώντας στο command window της Matlab: `>>vrinstall('-install','editor')`. Τρέχοντας τελικά το αρχείο 'vrbuild2.exe' ανοίγει το περιβάλλον στο οποίο μπορεί να κατασκευαστεί ένας 3D κόσμος.

Το V-Realm Builder έχει φιλικό για τον χρήστη παραθυρικό περιβάλλον, για σχεδιασμό και έλεγχο ενός 3D κόσμου. Δεν είναι απαραίτητη η συγγραφή κώδικα, αφού όλα μπορούν να γίνουν με χρήση ποντικιού. Ο τρόπος μοντελοποίησης περιγράφεται με λεπτομέρεια στα [46] [48] [51], και δεν θα παρουσιαστεί στην παρούσα εργασία. Όπως φαίνεται στο Σχήμα 6.9 δημιουργήθηκε ένας εικονικός κόσμος, όμοιος με αυτόν του διαστημικού εξομοιωτή του εργαστηρίου.

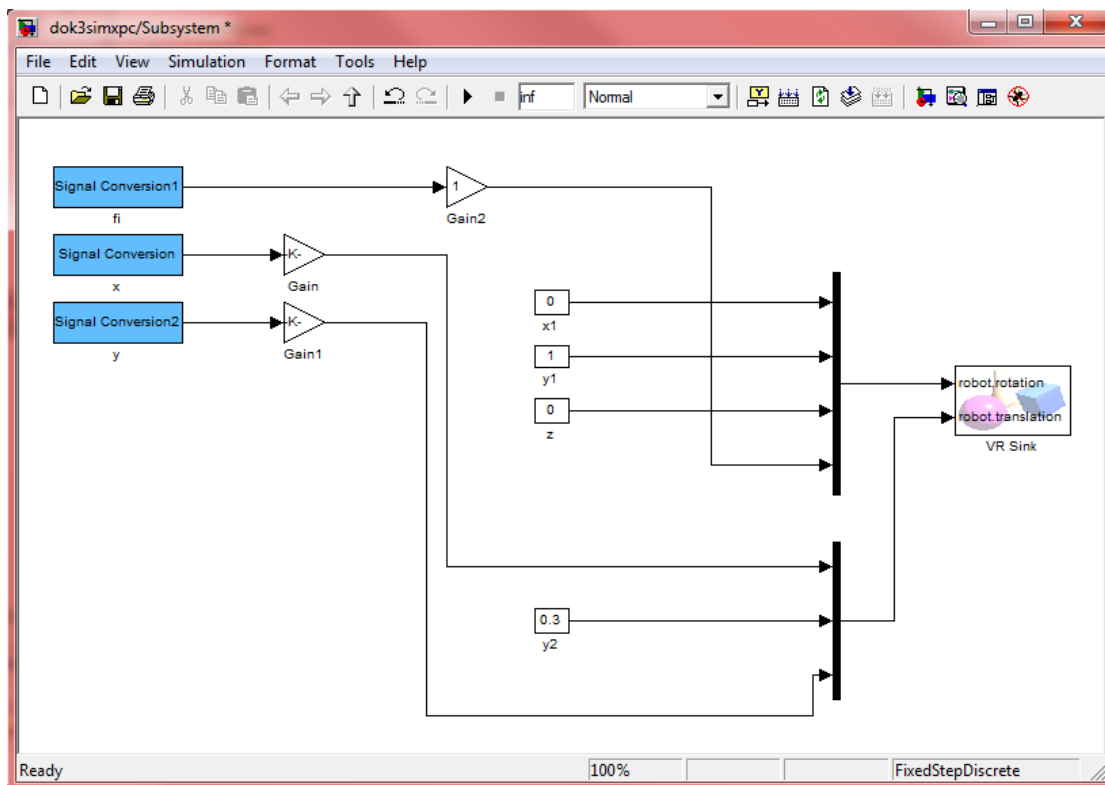


Σχήμα 6.9 Μοντέλο του ρομπότ όπως φαίνεται στο περιβάλλον του V-Realm Builder.

### 6.5.2 Έλεγχος εικονικού κόσμου μέσω Simulink

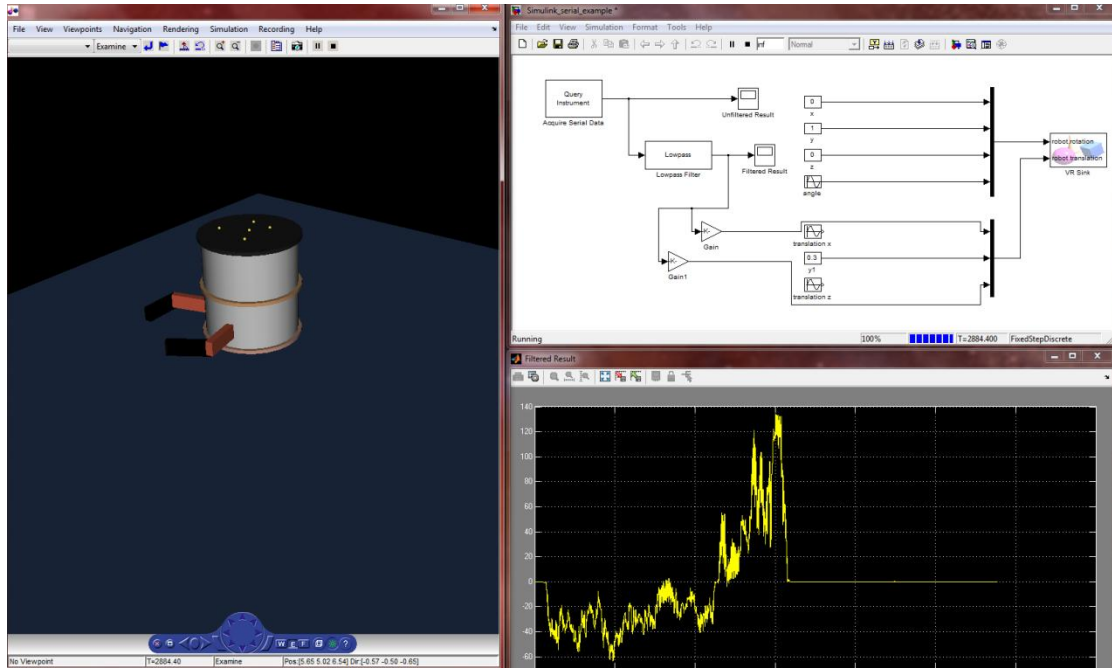
Ο εικονικός κόσμος που δημιουργήθηκε μπορεί τώρα να ελεγχθεί μέσω ενός μοντέλου Simulink. Η βιβλιοθήκη 3D Animation [49] παρέχει τα κατάλληλα blocks για σύνδεση των σημάτων του μοντέλου προσομοίωσης με τις παραμέτρους του εικονικού κόσμου. Με το κουτί 'VR Sink' είναι δυνατή η εγγραφή δεδομένων από το simulink στο κόσμο. Οι παράμετροι ελέγχου επιλέγονται μέσα από το περιβάλλον του Simulink. Οι ιδιότητες των κόμβων παρατίθενται ιεραρχικά σε μορφή δέντρου. Όταν εισαχθεί στο μπλοκ 'VR Sink' ένας εικονικός κόσμος, σαρώνεται αυτόματα για διαθέσιμους VRML κόμβους, που μπορούν να οριστούν ως εισόδοι, και να οδηγηθούν σε αυτές τα σήματα που παράγονται από το λογισμικό Simulink. Μετά την επιλογή των κόμβων, το διάγραμμα VR Sink ενημερώνεται με εισόδους και εξόδους, που αντιστοιχούν στους επιλεγμένους αυτούς κόμβους του εικονικού κόσμου (βλ. Σχήμα 6.10). Ο χρήστης μπορεί να παρακολουθεί την προσομοίωση στον 3D κόσμο μέσω του λογισμικού αναπαράστασης VRML Viewer.

Στην παρούσα εφαρμογή, χρησιμοποιήθηκε μια δυνατότητα που παρέχει το Simulink για αυτόματη σύνδεση του μοντέλου και του εικονικού κόσμου μέσω συγκεκριμένων παραμέτρων/ σημάτων. Έγινε ως εξής: Αφού δημιουργήθηκε το μοντέλο simulink, τα κατάλληλα σήματα μαρκάρονται (tagging/marking signals), τους δίνονται δηλαδή συγκεκριμένο όνομα και ιδιότητες. Ύστερα, χρησιμοποιώντας την συνάρτηση 'xpcsliface' δημιουργούνται τα κατάλληλα 'From xPC Target' μπλοκς, τα οποία περιέχουν τα σήματα που δηλώθηκαν. Ο απλός αυτός τρόπος περιγράφεται αναλυτικά στο [50]. Το μοντέλο Simulink που παράγεται αυτόματα από αυτή τη διαδικασία παρουσιάζεται στο Σχήμα 6.10.



**Σχήμα 6.10** Μοντέλο Simulink για σύνδεση με εικονικό κόσμο.

Τρέχοντας λοιπόν αυτό το μοντέλο ταυτόχρονα με τα προηγούμενα μοντέλα του xPC Target προσφέρεται μια καλύτερη και πιο άμεση αναπαράσταση των πειραματικών αποτελεσμάτων (βλ. Σχήμα 6.11). Μάλιστα, η απόκριση του εικονικού κόσμου στα πειραματικά αποτελέσματα είναι γρήγορη και έτσι δεν παρατηρείται χρονική καθυστέρηση σε σχέση με τις πραγματικές κινήσεις του ρομπότ πάνω στην τράπεζα του γρανίτη.



Σχήμα 6.11 Εικονικός κόσμος κατά τη διάρκεια πειράματος.



# ΚΕΦΑΛΑΙΟ 7

## Ηλεκτρονικός σχεδιασμός και εκτύπωση πλακετών

### 7.1 Εισαγωγή

Είναι φανερό ότι για να λειτουργήσει το ρομποτικό σύστημα, είναι απαραίτητες κάποιες ηλεκτρονικές πλακέτες οι οποίες θα συνδέουν κατάλληλα τα διάφορα υποσυστήματα που έχουν ήδη μελετηθεί. Ο σχεδιασμός των πλακετών αυτών έγινε με στόχο την ευκολία των συνδέσεων από τον χρήστη, την εύκολη εποπτεία του συστήματος ανά πάσα στιγμή, τον εύκολο έλεγχο των σημαντικών σημείων του κυκλώματος, την ελαχιστοποίηση του χώρου και τέλος την επεκτασιμότητα του κυκλώματος.

Ο σχεδιασμός και η εκτύπωση των κυκλωμάτων έγινε με βάση τις βασικές αρχές του ηλεκτρικού/ηλεκτρονικού σχεδιασμού, στις οποίες θα γίνει αναφορά παρακάτω. Χρησιμοποιήθηκε το πακέτο λογισμικού της National Instruments, Multisim [10] για τον αρχικό σχεδιασμό των κυκλωμάτων, και το Ultiboard για την τελική τοποθέτηση των ηλεκτρονικών στις πλακέτες και τελικά την εξαγωγή των απαραίτητων αρχείων για την εκτύπωση. Η εκτύπωση έγινε σε εταιρεία εκτύπωσης ηλεκτρονικών

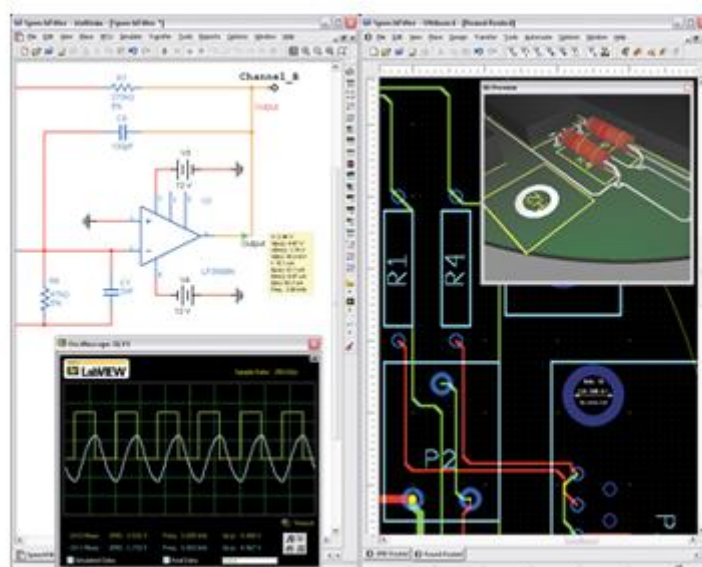
πλακετών. Η επιλογή των συγκεκριμένων σχεδιαστικών πακέτων έγινε λόγω της προτίμησης τους στον ηλεκτρονικό σχεδιασμό του πρώτου ρομπότ του διαστημικού εξομοιωτή, αλλά και λόγω της ευχρηστίας και της αξιοπιστίας τους. Άλλα αξιόπιστα προγράμματα ηλεκτρονικού σχεδιασμού είναι τα Eagle, Cadense και Fritzing. Παρακάτω γίνεται μια σύντομη περιγραφή των Multisim, Ultiboard.

## 7.2 Πακέτα ηλεκτρονικής σχεδίασης NI Multisim και NI Ultiboard

Σήμερα, η σχεδίαση ηλεκτρονικών κυκλωμάτων έχει περάσει σε ένα νέο επίπεδο και χρησιμοποιεί ηλεκτρονικούς υπολογιστές και προγράμματα σχεδίασης τα οποία μειώνουν τον χρόνο σχεδίασης και παρέχουν δυνατότητες που δεν μπορούν να αποκτηθούν χρησιμοποιώντας απλά μέσα σχεδίασης.

Στην εξέλιξη των προγραμμάτων σχεδίασης ηλεκτρονικών κυκλωμάτων, έχουν εισαχθεί και νέα στοιχεία που αυξάνουν την ποιότητα των κυκλωμάτων και μειώνουν το κόστος παραγωγής τους, χρησιμοποιώντας μεθόδους προσομοίωσης και ανάλυσης αυτών. Το λογισμικό Multisim της National Instruments είναι ένα πρόγραμμα σχεδιασμού, προσομοίωσης και ανάλυσης ηλεκτρονικών κυκλωμάτων. Με το Multisim είναι δυνατός ο σχεδιασμός κυκλωμάτων, χρησιμοποιώντας μία μεγάλη βάση δεδομένων από εξαρτήματα του εμπορίου. Είναι μία εφαρμογή που έχει σχεδιαστεί για Windows και χαρακτηρίζεται από όλα τα στοιχεία που αποτελούν μία αντίστοιχη εφαρμογή (παράθυρα, πλήκτρα, μπάρες εργαλείων, κουτιά συμπλήρωσης κειμένου κ.τ.λ.). Το πρόγραμμα αποτελείται από ένα κεντρικό παράθυρο, μέσα στο οποίο γίνεται η σχεδίαση των κυκλωμάτων.

Το Multisim παρέχει έναν εύκολο τρόπο μετατροπής των κυκλωμάτων που σχεδιάζονται στο χώρο εργασίας σε μορφή τυπωμένου κυκλώματος (βλ Σχήμα 7.1). Η διαδικασία αυτή γίνεται με τη χρήση του λογισμικού Ultiboard που είναι ένα πρόγραμμα σχεδιασμού τυπωμένων κυκλωμάτων. Το Ultiboard μπορεί να λειτουργήσει και ως αυτόνομο πρόγραμμα σχεδιασμού τυπωμένων αλλά και σε συνδυασμό με το Multisim. Τα εξαρτήματα που τοποθετούνται στο Multisim έχουν δύο τύπους. Μπορεί να είναι εικονικά εξαρτήματα (Virtual) τα οποία χρησιμοποιούνται μόνο για σκοπούς προσομοίωσης, ή πραγματικά εξαρτήματα που χρησιμοποιούνται για σκοπούς προσομοίωσης και για εξαγωγή σε μορφή τυπωμένου κυκλώματος. Η διαφορά ενός πραγματικού εξαρτήματος από ένα εικονικό, είναι ότι το πραγματικό εξάρτημα έχει ένα επιπλέον στοιχείο που ονομάζεται Footprint και είναι η μορφή του εξαρτήματος στο τυπωμένο κύκλωμα. Όταν τοποθετείται λοιπόν ένα εξάρτημα μέσα στο χώρο εργασίας θα πρέπει να επιλεγθεί το επιθυμητό footprint μέσα από την αντίστοιχη λίστα του εξερευνητή εξαρτημάτων.



**Σχήμα 7.1** Τα περιβάλλοντα των σχεδιαστικών πακέτων Multisim και Ultiboard.

### 7.3 Βασικές αρχές Ηλεκτρονικού Σχεδιασμού (Electronics Design)

Για κάποιους σχεδιαστές, το σχέδιο PCB (Printed Circuit Board ή πλακέτα τυπωμένου κυκλώματος) είναι φυσική και εύκολη επέκταση της σχεδιαστικής διαδικασίας. Πολλές εταιρίες όμως, έχουν δικά τους τμήματα αφοσιωμένα στη σχεδίαση PCB. Αυτό δεν προκαλεί έκπληξη, αν σκεφτεί κανείς ότι χρειάζεται ιδιαίτερες γνώσεις και ταλέντο για την τοποθέτηση εκατοντάδων υλικών και χιλιάδων καλωδιώσεων σε ένα σχέδιο το οποίο υπακούει σε μεγάλο αριθμό φυσικών και ηλεκτρικών απαιτήσεων. Σε πολλά σχέδια (για παράδειγμα ψηφιακά με σήματα υψηλής συχνότητας, αναλογικά χαμηλού επιπέδου και RF) το layout του PCB μπορεί να φτιάξει ή να χαλάσει την λειτουργία και την ηλεκτρική απόδοση του σχεδίου. Τα ίχνη (αγωγοί) του PCB έχουν αντίσταση και χωρητικότητα, όπως και το κύκλωμα. Υπάρχουν πολλοί βασικοί κανόνες και πρακτικές [7], αλλά πέρα από αυτό, ο σχεδιασμός PCB είναι μια πολύ δημιουργική και ιδιαίτερη εργασία που βασίζεται σε μεγάλο βαθμό στον σχεδιαστή. Πολλοί σχεδιαστές PCB βλέπουν τα layout των PCB σαν έργα τέχνης, με τη λογική ότι 'αν δείχνει καλό, θα δουλεύει και καλά'.

Στις προ του υπολογιστή μέρες, τα PCB σχεδιάζονταν με το χέρι, χρησιμοποιώντας αυτοκόλλητες ταινίες και pads σε καθαρό διάφανο φιλμ. Πολλές ώρες ξοδεύονταν πάνω από ένα κουτί φθορισμού φωτός, κόβοντας, τοποθετώντας και συνδέοντας τα εξαρτήματα και τους αγωγούς με το χέρι. Αυτές οι μέρες έχουν φύγει, από τη στιγμή που ο σχεδιασμός PCB μέσω υπολογιστή έχει αντικαταστήσει εντελώς τις παλιές μεθόδους για επαγγελματίες και μη. Τα βασισμένα σε υπολογιστές προγράμματα CAD επιτρέπουν την ευελιξία στο σχεδιασμό και την εύκολη διόρθωση των πλακετών. Αυτό που κάποτε έπαιρνε ώρες, τώρα μπορεί να γίνει σε λίγα λεπτά.

#### Μονάδες μέτρησης

Το πρώτο πράγμα που πρέπει να ξεκαθαριστεί για το σχέδιο PCB, είναι ποιες μονάδες μέτρησης χρησιμοποιούνται και οι κοινές ορολογίες τους, οι οποίες μπορεί να προκαλέσουν σύγχυση. Η πλειοψηφία των ηλεκτρονικών υλικών έχει κατασκευαστεί με μονάδες μέτρησης τις ίντσες. Το μιλιμετρικό σύστημα καθιστά την απεικόνιση (layout) τη πλακέτας δυσκολότερη και πιο ακατάστατη. Το στοιχείο που κυρίως χρησιμοποιείται για την κατασκευή και σχεδίαση των PCB είναι το thou. Το thou είναι το 1/1000 μιας ίντσας, χρησιμοποιείται παγκοσμίως και αναγνωρίζεται από όλους τους σχεδιαστές και τους κατασκευαστές PCB. 1 mil είναι το ίδιο με 1 thou, οπότε 1 mil = 1 thou = 1/1000 της ίντσας. Επίσης η 1 ίντσα είναι συνήθως γνωστή ως 1 pitch. Έτσι, είναι κοινό να ακουστεί η φράση '0,1 pitch της ίντσας', ή απλούστερα '0,1 pitch'.

Κατά γενικό κανόνα, χρησιμοποιούνται τα thous για τις συνδέσεις, τα pads, τα διαστήματα και τα πλέγματα, που είναι οι κύριες απαιτήσεις του σχεδίου. Χρήση χιλιοστόμετρων (mm) γίνεται συνήθως για τις μηχανικές και κατασκευαστικές απαιτήσεις, όπως τα μεγέθη οπών και οι διαστάσεις της πλακέτας.

#### Σχεδιασμός ιχνών συνδέσεων (tracks)

Δεν υπάρχει κανένα συνιστώμενο πρότυπο για το μέγεθος του ίχνους της σύνδεσης. Το ποιο μέγεθος θα χρησιμοποιηθεί θα εξαρτηθεί από τις ηλεκτρικές απαιτήσεις του σχεδίου, το διάστημα και τη καθαρότητα της συνδεσμολογίας. Κάθε σχέδιο έχει ένα διαφορετικό σύνολο ηλεκτρικών απαιτήσεων που μπορεί να ποικίλουν μεταξύ των συνδεσμολογιών στην πλακέτα. Όλα, εκτός από τα απλά σχέδια, απαιτούν ένα μίγμα μεγεθών ίχνους συνδέσεων. Γενικός κανόνας είναι ότι, όσο μεγαλύτερο είναι το πλάτος ίχνους, τόσο το καλύτερο. Τα μεγαλύτερα ίχνη έχουν χαμηλότερη συνεχή αντίσταση, χαμηλότερη αυτεπαγωγή, είναι ευκολότερα και φτηνότερα να χαραχθούν από τον κατασκευαστή και ευκολότερα για επιθεώρηση.

Είναι καλή η χρήση μεγάλου ορίου ίχνους/διαστήματος, εκτός κι αν οι παράμετροι του σχεδίου απαιτούν κάτι μικρότερο. Σαν αρχή, καλό είναι να χρησιμοποιείται 25 thou για τα ίχνη των σημάτων, 50 thou για τροφοδοσία και γειώσεις, και 10-15 thou για πέρασμα μεταξύ των pads των υλικών και των ολοκληρωμένων κυκλωμάτων. Η ορθή πρακτική σχεδίου είναι να κρατηθούν τα ίχνη όσο το δυνατόν μεγαλύτερα, και μόνο σε περίπτωση ανάγκης να γίνουν μικρότερα έτσι ώστε να συμφωνούν

με τις απαιτήσεις. Η αλλαγή του ίχνους από μεγάλο σε μικρό και έπειτα πάλι σε μεγάλο, είναι γνωστή ως 'necking'. Αυτό απαιτείται συχνά όταν πρέπει το ίχνος να περάσει μεταξύ του ολοκληρωμένου κυκλώματος ή pads υλικών. Έτσι, επιτρέπεται να υπάρχουν μεγάλα ίχνη χαμηλής σύνθετης αντίστασης, αλλά και ευελιξία στη συνδεσμολογία μεταξύ των δύσκολων σημείων.

Στην πράξη, το πλάτος ίχνους υπαγορεύεται από την τρέχουσα ροή του, και τη μέγιστη θερμοκρασία που θα ανεχτεί η πλακέτα. Το κάθε ίχνος έχει ένα ορισμένο ποσό αντίστασης και θα διασκορπίσει τη θερμότητα ακριβώς όπως ένας αντιστάτης. Όσο μεγαλύτερο το ίχνος τόσο χαμηλότερη η αντίσταση. Επίσης το πάχος του χαλκού στο PCB, καθώς και οποιοδήποτε επίστρωμα ύλης συγκολλησεως παίζει σημαντικό ρόλο. Το πάχος του χαλκού στο PCB μετριέται ονομαστικά σε ουγγιές ανά τετραγωνικό πόδι, με το 1oz χαλκό να είναι η πιο κοινή μονάδα μέτρησης. Τα παχύτερα στρώματα χαλκού είναι χρήσιμα για το υψηλό ρεύμα και τα σχέδια υψηλής σταθερότητας.

Οι υπολογισμοί για την εύρεση του απαραίτητου πλάτους του ίχνους που βασίζονται στο ρεύμα και τη μέγιστη άνοδο θερμοκρασίας είναι σύνθετοι. Μπορεί επίσης να είναι αρκετά ανακριβείς, δεδομένου ότι τα πρότυπα ορίζονται από ένα σύνολο μη γραμμικών γραφικών παραστάσεων που βασίζονται σε συλλογή στοιχείων σχεδόν μισό αιώνα πριν. Εμπειρικά, μια άνοδος θερμοκρασίας 10 βαθμών °C στο ίχνος είναι ένα ασφαλές όριο για να σχεδιάσει κανείς. Παρακάτω επισυνάπτεται ένας πίνακας πλατών ίχνους για μια άνοδο θερμοκρασίας 10 βαθμών °C, ανάλογα με το ρεύμα που περνά από το ίχνος (Πίνακας 7.1).

IPC Recommended Track Width For 1 oz cooper PCB and 10 °C Temperature Rise

Current/A	Track Width(mil)	Track Width(mm)
1	10	0.25
2	30	0.76
3	50	1.27
4	80	2.03
5	110	2.79
6	150	3.81
7	180	4.57
8	220	5.59
9	260	6.60
10	300	7.62

**Πίνακας 7.1** Αντιστοιχία Ρεύματος [A] - Πάχος ίχνους [mil ή mm].

## Pads

Τα μεγέθη των pads, οι μορφές και οι διαστάσεις θα εξαρτηθούν όχι μόνο από το υλικό που χρησιμοποιείται, αλλά και από τη διαδικασία κατασκευής της πλακέτας. Υπάρχει ένα σύνολο προτύπων και θεωριών πίσω από τα μεγέθη και τις μορφές τους. Υπάρχει μια σημαντική παράμετρος γνωστή ως αναλογία pads/οπές. Αυτή είναι η αναλογία του μεγέθους των pads προς το μέγεθος των οπών. Κάθε κατασκευαστής θα έχει την δική του ελάχιστη προδιαγραφή ως προς την αναλογία αυτή. Η απλή εμπειρική μέθοδος λέει ότι το pad θα πρέπει να είναι τουλάχιστον 1.8 φορές μεγαλύτερο από τη διάμετρο της οπής, ή τουλάχιστον 0.5 mm. Υπάρχουν μερικές κοινές πρακτικές που χρησιμοποιούνται για τα pads των υλικών. Τα pads για επιμολυβδωμένα υλικά όπως οι αντιστάτες, οι πυκνωτές και οι δίοδοι πρέπει να είναι στρογγυλά, με περίπου 70 του διάμετρο συνήθως.

Στα υλικά Dual In Line(DIL) όπως τα ολοκληρωμένα κυκλώματα ταιριάζουν καλύτερα τα ωοειδή διαμορφωμένα pads (60 του ύψος και 90-100 του εύρος συνήθως). Η ακίδα 1 ενός chip θα πρέπει πάντα να έχει διαφορετική μορφή pad, συνήθως ορθογώνια, και ίδιες διαστάσεις με τις άλλες ακίδες.



## Vias

Τα Vias συνδέουν τα ίχνη από την μια πλευρά της πλακέτας στην άλλη, μέσω μιας οπής στην πλακέτα. Τα vias γίνονται με ηλεκτρικά καλυμμένες με μέταλλο οπές, που αποκαλούνται επιμεταλλωμένες ενδιάμεσες οπές through hole (PTH). Οι επιμεταλλωμένες ενδιάμεσες οπές (through holes) επιτρέπουν την ηλεκτρική σύνδεση μεταξύ των διαφορετικών στρωμάτων στην πλακέτα. Οι τρύπες στα vias είναι συνήθως λίγο μικρότερες από τα pads των υλικών, με τυπικό μέγεθος 0.5-0.7mm. Η χρήση του via για τη σύνδεση δύο επιπέδων καλείται συνήθως 'ράψιμο' (stitching), δεδομένου ότι ενώνονται ηλεκτρικά τα δύο επίπεδα μαζί, όπως θα περνούσε μια βελόνα με κλωστή πέρα δώθε μεταξύ των υλικών.

## Αποστάσεις (clearances)

Οι ηλεκτρικές αποστάσεις είναι μια σημαντική προϋπόθεση για τις πλακέτες. Μια πάρα πολύ μικρή απόσταση μεταξύ των ιχνών και των pads μπορεί να οδηγήσει σε προβλήματα κατά τη διάρκεια της διαδικασίας κατασκευής. Αυτά τα προβλήματα μπορεί να είναι πολύ δύσκολο να βρεθούν αφού κατασκευασθεί η πλακέτα. Τουλάχιστον 15 thou είναι ένα καλό όριο απόστασης για τα βασικά σχέδια through hole, με 10 ή 8 thou να χρησιμοποιούνται για πυκνότερες επιφάνειες τοποθέτησης στην πλακέτα.

## Τοποθέτηση υλικών

Οι μέθοδοι τοποθέτησης υλικών είναι όσοι και οι σχεδιαστές. Αν το ίδιο κύκλωμα δοθεί σε 100 διαφορετικούς σχεδιαστές, θα προκύψουν 100 διαφορετικά PCBs. Δεν υπάρχει κάποιος απόλυτα σωστός τρόπος να τοποθετηθούν τα υλικά. Υπάρχουν όμως αρκετοί βασικοί κανόνες που βοηθούν και διευκολύνουν τη συνδεσμολογία, δίνουν καλύτερη ηλεκτρική απόδοση, και απλοποιούν τα μεγάλα και σύνθετα σχέδια. Παρακάτω δίνεται μια ιδέα των βασικών βημάτων που απαιτούνται για την απεικόνιση (layout) μιας πλήρους πλακέτας.

Ο καλύτερος τρόπος να αρχίσει το layout, είναι να τοποθετηθούν πρώτα όλα τα υλικά στην οθόνη του Η/Υ. Ο απλούστερος τρόπος να γίνει αυτό είναι να γίνει εισαγωγή του σχηματικού σχεδίου στο πρόγραμμα PCB και να επιλεχτούν αυτόματα όλα τα υλικά. Κάποια μέρη του σχεδίου μπορούν να χωριστούν σε δομικές μονάδες, ενώ τα ηλεκτρικά ευαίσθητα μέρη του σχεδίου καλό είναι να χωρίζονται σε μεγαλύτερα μπλοκ. Ένα γενικό παράδειγμα που πρέπει να αποφεύγεται είναι, τα ανακατεμένα ψηφιακά και αναλογικά κυκλώματα. Τα ψηφιακά και τα αναλογικά δεν είναι καλό να αναμιγνύονται, και θα πρέπει να είναι φυσικά και ηλεκτρικά διαχωρισμένα. Το ίδιο ισχύει και για τα κυκλώματα υψηλής συχνότητας και υψηλού ρεύματος με τα ευαίσθητα κυκλώματα χαμηλής συχνότητας και χαμηλού ρεύματος.

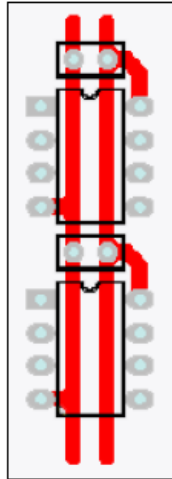
Κατά γενικό κανόνα, τα υλικά πρέπει να είναι ευθυγραμμισμένα. Πρέπει τα ολοκληρωμένα κυκλώματα να τοποθετούνται στην ίδια κατεύθυνση, οι αντιστάσεις σε τακτοποιημένες στήλες, οι πολωμένοι πυκνωτές τοποθετημένοι με τον ίδιο τρόπο, και οι connectors στην άκρη της πλακέτας. Δεν πρέπει όμως να γίνονται τα παραπάνω, αν είναι να οδηγήσουν σε ένα ηλεκτρικά φτωχό layout, ή μια υπερβολικά μεγάλη πλακέτα. Οι ηλεκτρικές προϋποθέσεις πρέπει πάντα να προηγούνται από τα ωραία παραταγμένα υλικά. Η συμμετρία βέβαια, είναι πραγματικά απαραίτητη σε ένα σχέδιο PCB.

Μετά την τοποθέτηση των υλικών, ακολουθεί η συνδεσμολογία για όλες τις ξεχωριστές δομικές μονάδες. Στο τέλος, είναι συνήθως απλό να μετακινηθούν και να τακτοποιηθούν οι δομικές μονάδες στο σχέδιο. Ο έλεγχος κανόνων σχεδίου (DRC) είναι ένα ουσιαστικό βήμα ώστε να εξασφαλίζεται ότι η πλακέτα είναι σωστή πριν από την κατασκευή. Ένας έλεγχος DRC ελέγχει βασικά τη σωστή συνδεσιμότητα των ιχνών καθώς και τα σωστά πλάτη και διαστήματα.

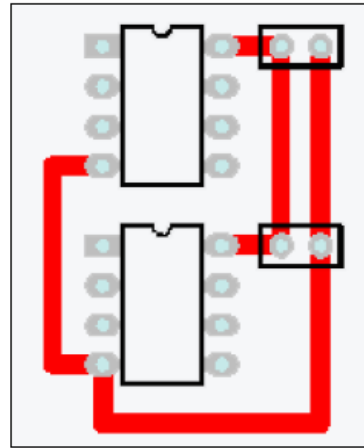
## Βασική συνδεσμολογία (routing)

Αναφέρονται μερικοί βασικοί κανόνες συνδεσμολογίας (βλ. Σχήμα 7.2). Η συνδεσμολογία (routing) είναι επίσης γνωστή ως 'tracking' και είναι η διαδικασία τοποθέτησης ιχνών για να συνδεθούν τα υλικά στην πλακέτα. Μια ηλεκτρική σύνδεση μεταξύ δύο ή περισσότερων pads είναι γνωστή ως net.

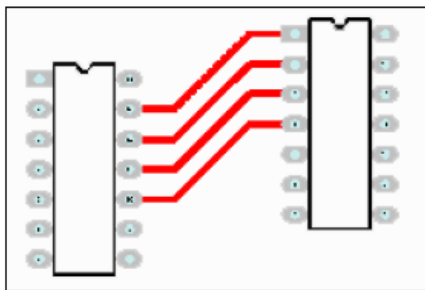
- Τα nets πρέπει να είναι όσο το δυνατόν πιο μικρά. Όσο μεγαλύτερο το συνολικό μήκος του ίχνους, τόσο μεγαλύτερη θα είναι η αντίσταση, η αγωγιμότητα και η αυτεπαγωγή, που μπορούν να είναι ανεπιθύμητοι παράγοντες.
- Τα ίχνη πρέπει να έχουν μόνο γωνίες 45° μοιρών. Αποφεύγεται η χρήση των ορθών γωνιών, και ποτέ δε χρησιμοποιείται γωνία μεγαλύτερη από 90° μοίρες.
- Πάντα περνά ένα μόνο ίχνος μεταξύ pads των 100 thou. Μόνο σε μεγάλα και πολύ πυκνά σχέδια περνούν δύο ίχνη μεταξύ των pads.
- Για υψηλά ρεύματα σε πολυστρωματικές πλακέτες, χρησιμοποιούνται πολλαπλά νίαι κατά τη μετάβαση μεταξύ των στρωμάτων. Αυτό μειώνει τη σύνθετη αντίσταση του ίχνους και βελτιώνει την αξιοπιστία. Είναι ένας γενικός κανόνας ώστε να μειωθεί η σύνθετη αντίσταση του ίχνους.
- Τα ίχνη της τροφοδοσίας σχεδιάζονται όσο το δυνατόν μεγαλύτερα.
- Τα ίχνη τροφοδοσίας και γείωσης κρατούνται σε στενή εγγύτητα το ένα στο άλλο εάν είναι δυνατόν, και δεν στέλνονται σε αντίθετες κατευθύνσεις στην πλακέτα. Αυτό χαμηλώνει την αυτεπαγωγή των βρόχων της τροφοδοσίας του συστήματος και επιτρέπει την αποτελεσματική παράκαμψη.
- Δεν αφήνεται κανένα αποσυνδεδεμένο γέμισμα χαλκού, (αποκαλούμενο 'νεκρός χαλκός'). Γειώνονται ή αφαιρούνται.
- Δεν τοποθετούνται νίαι κάτω από τα υλικά. Μόλις συγκολληθεί το υλικό δεν θα υπάρχει πρόσβαση στην ένωση.
- Όσο περισσότερη ύλη συγκολλήσεως υπάρχει, τόσο λιγότερο αξιόπιστη γίνεται η πλακέτα, ενώ παίρνει και πολύ περισσότερο καιρό για να κατασκευαστεί.



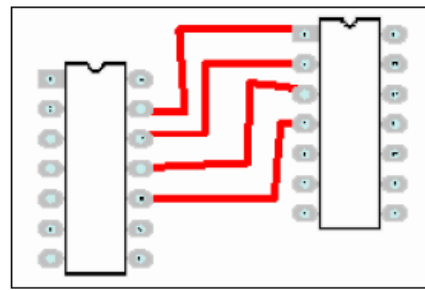
α. καλή συνδεσμολογία των επαφών τροφοδοσίας του ολοκληρωμένου



β. κακή συνδεσμολογία των επαφών τροφοδοσίας του ολοκληρωμένου



α. καλή συνδεσμολογία



β. κακή συνδεσμολογία

Σχήμα 7.2 Παραδείγματα καλής και κακής συνδεσμολογίας.

### Σχέδιο μονής όψευς

Το σχέδιο μονής όψευς μπορεί να μειώσει πολύ το κόστος της πλακέτας. Εάν μπορεί να χωρέσει το σχηματικό σε ένα σχέδιο μονής όψευς, είναι προτιμότερο. Τα σχέδια αυτά χρησιμοποιούνται αρκετά επειδή είναι φτηνά στην κατασκευή. Η τοποθέτηση υλικών είναι ακόμα κρίσιμότερη σε ένα σχέδιο μονής όψευς. Τα υλικά πρέπει να τοποθετούνται έτσι ώστε να δίνουν την κοντινότερη και αποδοτικότερη δυνατή συνδεσμολογία.

### Σχέδιο διπλής όψευς

Το σχέδιο διπλής όψευς δίνει έναν πρόσθετο βαθμό ελευθερίας για το σχεδιασμό της πλακέτας. Πράγματα που ήταν σχεδόν αδύνατα σε έναν σχέδιο μονής όψευς, γίνονται σχετικά εύκολα όταν προστεθεί ένα ακόμη στρώμα (layer). Η τοποθέτηση των υλικών παίζει και εδώ μεγάλο ρόλο, και δεν αποτελεί καλή λύση η χρήση πολλών vias. Πρέπει να δίνεται έμφαση στη χρήση τεχνικών καλής τοποθέτησης υλικών και επαρκούς συνδεσμολογίας δομικών μπλοκ. Το σχέδιο διπλού επιπέδου μπορεί επίσης να δώσει την δυνατότητα να χρησιμοποιηθεί η καλή τεχνική του επιπέδου γείωσης, που απαιτείται για τα σχέδια υψηλής συχνότητας και όχι μόνο (περιγράφεται παρακάτω).

### Πολυστρωματικό σχέδιο (multi layer design)

Ένα πολυστρωματικό PCB είναι ακριβότερο και πιο δύσκολο να κατασκευαστεί απ' ό,τι μια πλακέτα μονής ή διπλής όψης, αλλά δίνει πρόσθετη πυκνότητα στην συνδεσμολογία ιχνών τροφοδοσίας και

σήματος. Έχοντας τα σήματά στο εσωτερικό της πλακέτας, μπορούν να τοποθετηθούν τα υλικά πιο κοντά το ένα στο άλλο και να προκύψει έτσι ένα πιο συμπαγές σχέδιο.

Η απόφαση για τη μετάβαση από σχέδιο δύο όψεων σε πολυστρωματικό μπορεί να είναι μια μεγάλη απόφαση, και είναι σίγουρο ότι ένα πολυστρωματικό σχέδιο εγγυάται μεγαλύτερη πολυπλοκότητα. Οι πολυστρωματικές πλακέτες περιέχουν ζυγό αριθμό στρωμάτων. Τα 6 και τα 8 στρώματα είναι τα πιο κοινά. Τεχνικά μπορούν να περιέχουν μονό αριθμό στρωμάτων, όπως πλακέτα 3 στρωμάτων, αλλά αυτό πραγματικά δεν διαφέρει σε κόστος από μια πλακέτα 4 στρωμάτων.

Σε μια πολυστρωματική πλακέτα, αφιερώνεται ένα στρώμα για όλες τις τροφοδοσίες και ένα άλλο για τις γειώσεις. Εάν χρειαστεί, θα μπορούσαν να μπουν και μερικά ίχνη σημάτων στο επίπεδο με τις τροφοδοσίες. Τα στρώματα τροφοδοσίας είναι σχεδόν πάντα στη μέση της πλακέτας, με τις γειώσεις πιο κοντά στο κορυφαίο στρώμα.

Στο πολυστρωματικό σχέδιο υπάρχει η επιλογή διαφορετικών τύπων vias ώστε να βελτιωθεί η πυκνότητα της συνδεσμολογίας. Υπάρχουν τρεις τύποι vias – τα τυπικά, τα τυφλά και τα θαμμένα (standard, blind, and buried). Τα τυπικά vias περνούν από ολόκληρη την πλακέτα και μπορεί να συνδέσουν οποιοδήποτε στρώμα ή σημείο. Αυτό μπορεί να αποτελεί σπατάλη χώρου στα στρώματα που δεν συνδέονται. Τα 'τυφλά' vias πηγαίνουν από την εξωτερική επιφάνεια σε ένα από τα εσωτερικά στρώματα μόνο. Τα 'θαμμένα' vias συνδέουν μόνο δύο ή περισσότερα εσωτερικά στρώματα, χωρίς η οπή να είναι ορατή από την πλακέτα. Τα τυφλά και τα θαμμένα vias κοστίζουν περισσότερο στην κατασκευή από τα τυπικά vias, αλλά είναι πολύ χρήσιμα, και σχεδόν υποχρεωτικά στα σχέδια πολύ υψηλής πυκνότητας. Απαντώνται σε πολύ συμπακνωμένα ηλεκτρονικά εξαρτήματα όπως ισχυρούς μικροεπεξεργαστές.

### **Στρώμα Silkscreen (Silkscreen Layer)**

Υπάρχουν μερικά άλλα σημαντικά στρώματα στο PCB εκτός από τα στρώματα ίχνους χαλκού. Το layer 'silkscreen' είναι επίσης γνωστό ως 'επίστρωση υλικών' ή 'στρώμα υλικού'. Είναι το στρώμα στην κορυφή της πλακέτας (και κατώτατο στρώμα εάν είναι απαραίτητο) όπου εμφανίζονται τα περιγράμματα των υλικών, τα χαρακτηριστικά τους (C1, R1 κ.λπ.), και ελεύθερο κείμενο. Αυτό προστίθεται στην πλακέτα χρησιμοποιώντας μια silkscreening διαδικασία. Το λευκό είναι το πιο συνηθισμένο χρώμα. Εξασφαλίζεται ότι όλα τα πολωμένα υλικά είναι χαρακτηρισμένα, και ότι η ακίδα 1 στα Dual In Line (DIL) κυκλώματα είναι προσδιορισμένη.

### **Μάσκα ύλης συγκολλήσεως**

Μια μάσκα ύλης συγκολλήσεως είναι ένα λεπτό πολυμερές επίστρωμα στην πλακέτα που περιβάλλει τα pads για να αποτρέψει την ύλη συγκολλήσεως να κάνει γεφύρωμα μεταξύ των ακίδων. Αυτό είναι ουσιαστικό για συσκευές που τοποθετούνται στην επιφάνεια. Η μάσκα ύλης συγκολλήσεως χαρακτηριστικά καλύπτει τα πάντα εκτός από τα pads και τα vias. Το πρόγραμμα PCB θα αφαιρέσει αυτόματα τη μάσκα ύλης συγκολλήσεως από τα pads και vias.

### **Μηχανικό στρώμα (Mechanical layer)**

Το μηχανικό στρώμα χρησιμοποιείται για να παρέχει ένα περίγραμμα στην πλακέτα. Δεν είναι μέρος του πραγματικού σχεδίου PCB, αλλά είναι πολύ χρήσιμο στον κατασκευαστή PCB για την κατασκευή της πλακέτας. Δεν υπάρχει κανένας κανόνας για αυτό το στρώμα. Χρησιμοποιείται κατά το δοκούν.

### **Netlists**

Ένα netlist είναι ουσιαστικά ένας κατάλογος συνδέσεων που αντιστοιχούν στη σχηματική αναπαράστασή. Περιέχει επίσης τον κατάλογο από τα υλικά, τα περιγράμματα των υλικών, τα ίχνη των υλικών και άλλες πληροφορίες σχετικά με τη σχηματική αναπαράστασή. Το αρχείο netlist μπορεί

να παραχθεί από το πακέτο του σχηματικού. Το πακέτο σχεδίασης PCB μπορεί να εισαγάγει αυτό το αρχείο netlist και να το αξιοποιήσει. Μπορεί αυτόματα να φορτώσει όλα τα απαιτούμενα υλικά επάνω στην πλακέτα. Μπορεί επίσης να ορίσει ένα όνομα σε κάθε μια από τις ακίδες των υλικών. Με τα nets ορισμένα στα υλικά PCB, είναι τώρα δυνατό να χρησιμοποιηθεί η Αυτόματη Συνδεσμολογία (Auto Route), ο Κανόνας Ελέγχου Σχεδίου DRC, και να απεικονιστεί η συνδεσιμότητα των υλικών. Αυτή είναι και η θεμελιώδης έννοια πίσω από τα σύγχρονα πακέτα CAD που αφορούν σχηματικές αναπαραστάσεις και PCB.

### Έλεγχος κανόνων σχεδίασης (DRC)

Ο έλεγχος κανόνων σχεδίασης (DRC) επιτρέπει τον αυτόματο έλεγχο του σχεδίου PCB ως προς τη συνδεσιμότητα, τα διαστήματα και άλλα λάθη κατασκευής. Με τα μεγάλα και σύνθετα PCB που σχεδιάζονται σήμερα, δεν είναι πρακτικός ο έλεγχος με το χέρι. Το DRC είναι ένα απόλυτα αναγκαίο βήμα στο σχεδιασμό PCB. Παραδείγματα που μπορούν να ελεγχθούν με το DRC είναι:

- Η συνδεσιμότητα κυκλώματος. Ελέγχει εάν κάθε διαδρομή στο layout ταιριάζει με τη συνδεσιμότητα της σχηματικής αναπαράστασης.
- Τα ηλεκτρικά διαστήματα. Ελέγχονται τα διαστήματα μεταξύ των ιχνών, των pads, και των υλικών.
- Οι κατασκευαστικές ανοχές όπως τα κατώτατα/άνωτα μεγέθη τρυπών, τα πλάτη ίχνους, τα πλάτη via, τα μεγέθη δακτυλίων, τα άκυρα μεγέθη και τα βραχυκυκλώματα.

Ένας πλήρης DRC έλεγχος εκτελείται συνήθως αφότου τελειώσει το PCB. Μερικά πακέτα εντούτοις έχουν τη δυνατότητα ενός DRC πραγματικού χρόνου που ελέγχει όσο δημιουργείται η πλακέτα.

### Επίπεδα τροφοδοσίας (power planes)

Είναι ορθή πρακτική να χρησιμοποιούνται τα επίπεδα τροφοδοσίας για να διανεμηθεί η τροφοδοσία σε ολόκληρη την πλακέτα. Η χρήση των επιπέδων τροφοδοσίας μπορεί να μειώσει δραστικά την αυτεπαγωγή και τη σύνθετη αντίσταση καλωδίωσης τροφοδοσίας στα υλικά. Αυτό μπορεί να είναι ζωτικής σημασίας για ένα υψηλής ταχύτητας ψηφιακό σχέδιο. Είναι λοιπόν καλή πρακτική να χρησιμοποιούνται τα επίπεδα τροφοδοσίας όποτε είναι δυνατόν. Μπορούν ακόμη να χρησιμοποιηθούν και στις πλακέτες διπλής όψης.

Ένα επίπεδο τροφοδοσίας είναι βασικά ένα στερεό στρώμα χαλκού της πλακέτας που αφιερώνεται στην τροφοδοσία, στην γείωση, ή και στις δύο. Τα επίπεδα τροφοδοσίας τοποθετούνται στα μέσα στρώματα της πλακέτας, συνήθως στα στρώματα που είναι πιο κοντά στις εξωτερικές επιφάνειες. Για μια πλακέτα 4 στρωμάτων με σύνθετες απαιτήσεις τροφοδοσίας, είναι συνηθισμένο να αφιερώνεται ένα στρώμα στη τροφοδοσία και ένα άλλο στρώμα στους διάφορους θετικούς και αρνητικούς αγωγούς τροφοδοσίας. Η τροχιά (rail) γείωσης είναι συνήθως η γραμμή αναφοράς σημάτων, έτσι ένα επίπεδο γείωσης είναι η πρώτη σκέψη πριν το επίπεδο τροφοδοσίας.

Σε πολλά πακέτα PCB τα ειδικά στρώματα επιπέδων τροφοδοσίας σχεδιάζονται και απεικονίζονται αντίστροφα από τα κανονικά στρώματα σχεδίασης. Σε ένα κανονικό στρώμα σχεδίασης, η πλακέτα υποτίθεται ότι είναι κενή ενώ σε ένα επίπεδο τροφοδοσίας, η πλακέτα υποτίθεται ότι είναι καλυμμένη με χαλκό. Το layout των ιχνών σε ένα επίπεδο τροφοδοσίας αφαιρεί το χαλκό.

Ένα απλό επίπεδο τροφοδοσίας δεν θα έχει καθόλου ίχνη (ή αφαιρούμενα κομμάτια χαλκού) επάνω του, αλλά απλά θα είναι ένα στερεό στρώμα χαλκού. Εντούτοις, είναι κοινή πρακτική στις πιο σύνθετες πλακέτες ο διαχωρισμός του επιπέδου τροφοδοσίας να επιτυγχάνεται με την απεικόνιση ιχνών. Έτσι διαχωρίζεται μια αναλογική από μια ψηφιακή γείωση και μειώνεται το ποσό ψηφιακού θορύβου της γείωσης που συνδέεται με πιο ευαίσθητα αναλογικά στοιχεία κυκλώματος. Ένα τυπικό διαχωρισμένο επίπεδο τροφοδοσίας θα περιελάμβανε ένα ίχνος τοποθετημένο δίπλα στον συνδετήρα τροφοδοσίας εισόδου (βύσμα τροφοδοσίας) ή στους κεντρικούς πυκνωτές φίλτρων καθώς και στην απέναντι άκρη της πλακέτας.

## Καλή γείωση

Η γείωση είναι θεμελιώδης στη λειτουργία πολλών κυκλωμάτων. Οι καλές ή μη τεχνικές γείωσης μπορούν να φτιάξουν ή να χαλάσουν το σχέδιό. Υπάρχουν αρκετές τεχνικές γείωσης που είναι πάντα ορθές και ενσωματώνονται σε οποιοδήποτε σχέδιο:

- Όσο περισσότερος χαλκός υπάρχει στο διάδρομο γείωσης, τόσο χαμηλότερη η σύνθετη αντίσταση. Αυτό είναι ιδιαίτερα επιθυμητό για πολλούς ηλεκτρικούς λόγους.
- Πάντα αφιερώνεται ένα από τα επίπεδα των πολυστρωματικών πλακετών στη γείωση. Το επίπεδο αυτό πρέπει να βρίσκεται κοντά στο κορυφαίο στρώμα.
- Φτιάχνονται χωριστοί διάδρομοι γείωσης για τα κρίσιμα μέρη του κυκλώματός. Αυτό είναι γνωστό ως γείωση 'αστέρι', επειδή η γείωση ακολουθεί όλη τη διαδρομή από ένα κεντρικό σημείο και πέρα, μοιάζοντας με ένα αστέρι. Οι χωριστές γραμμές γείωσης, δεν αφήνουν το ρεύμα και το θόρυβο από ένα υλικό να επηρεάσουν τα υπόλοιπα υλικά.
- Αν χρησιμοποιηθεί ένα επίπεδο γείωσης, εφαρμόζονται οι τεχνικές διαχωρισμού επιπέδου για να δοθεί αποτελεσματική γείωση 'αστεριού'.
- 'Σύρραψη των σημείων, όπου είναι απαραίτητο κατ' ευθείαν στο επίπεδο γείωσης, χωρίς να χρησιμοποιηθεί επιπλέον μήκος ίχνους από όσο χρειάζεται.
- Χρήση πολλαπλών vias για να μειωθεί η σύνθετη αντίσταση ιχνών στη γείωση.

## Καλή παράκαμψη (Good Bypassing)

Τα ενεργά υλικά και τα σημεία στο κύκλωμα που τραβούν αρκετό εναλλασσόμενο ρεύμα πρέπει πάντα να είναι υπό 'παράκαμψη'. Αυτό πρόκειται να εξομαλύνει τη μπάρα ρεύματος που πηγαίνει σε μια συσκευή. Μια τυπική τιμή πυκνωτών παράκαμψης είναι τα 100nF, αν και άλλες τιμές, όπως 1uF, 10nF και 1nF, χρησιμοποιούνται συχνά για να παρακαμφθούν οι διαφορετικές συχνότητες. Μπορούν ακόμη να τοποθετηθούν παράλληλα δύο ή τρεις πυκνωτές διαφορετικών τιμών. Δεν είναι ασυνήθιστο σε ένα μεγάλο σχέδιο να υπάρξουν εκατοντάδες πυκνωτές παράκαμψης. Κατά γενικό κανόνα, πρέπει να χρησιμοποιείται τουλάχιστον ένας πυκνωτής παράκαμψης ανά ολοκληρωμένο κύκλωμα ή άλλο υλικό-διακόπτη, αν είναι δυνατόν. Τυπικές τιμές πυκνωτών παράκαμψης είναι τα 100nF για γενική χρήση, και 1uF ή 10uF για χαμηλές συχνότητες.

## Τεχνικές σχεδίου υψηλής συχνότητας

Σε σχέδια υψηλής συχνότητας πρέπει οπωσδήποτε να εξετάζονται τα αποτελέσματα της παρασιτικής αυτεπαγωγής, της χωρητικότητας και της σύνθετης αντίστασης του σχεδίου PCB. Εάν το σήμα είναι πάρα πολύ γρήγορο, και η διαδρομή αρκετά μεγάλη, τότε αυτή μπορεί να αποκτήσει τις ιδιότητες μιας γραμμής μετάδοσης. Σε τέτοιες καταστάσεις εάν δεν χρησιμοποιηθούν οι κατάλληλες τεχνικές για γραμμές μετάδοσης, μπορεί να δημιουργηθούν αντανakλάσεις και άλλα προβλήματα ακεραιότητας σημάτων.

Στις τυπικές πλακέτες χαλκού, ένα σήμα ταξιδεύει, κατά προσέγγιση, 6 ίντσες ανά nsec. Μια εμπειρική μέθοδος δηλώνει ότι πρέπει να προσεχθεί ιδιαίτερα όταν το μήκος ίχνους προσεγγίσει το μισό του μεγέθους του αριθμού αυτού. Στην πραγματικότητα όμως μπορεί το μήκος να είναι πολύ μικρότερο από αυτό. Σημειώνεται ότι τα ψηφιακά τετραγωνικά σήματα έχουν ένα πολλαπλό αρμονικό περιεχόμενο (πολλές αρμονικές που συνδυαζόμενες σχηματίζουν τους τετραγωνικούς παλμούς), έτσι ένα τετραγωνικό κύμα των 100MHz μπορεί πραγματικά να έχει τμήματα σημάτων που να εκτείνονται στην περιοχή των GHz. Στα κυκλώματα υψηλών συχνοτήτων, το επίπεδο γείωσης είναι θεμελιώδες στη διατήρηση της ακεραιότητας των σημάτων και στη μείωση των εκπομπών ηλεκτρομαγνητικής ακτινοβολίας EMI (θωράκιση). Επιτρέπει τη δημιουργία ιχνών ελεγχόμενης σύνθετης αντίστασης, τα οποία ταιριάζουν με την ηλεκτρική πηγή και το φορτίο. Επιτρέπει επίσης να κρατούνται τα σήματα συνδεδεμένα 'σφιχτά' με την πορεία επιστροφής τους (γείωση).

Μερικές χρήσιμες πληροφορίες και εμπειρικές τεχνικές για τα σχέδια υψηλής συχνότητας είναι:

- Τα ίχνη σημάτων υψηλής συχνότητάς κρατούνται όσο το δυνατόν πιο κοντά σε μήκος.

- Αποφεύγεται το πέρασμα των κρίσιμων ιχνών σημάτων υψηλής συχνότητας πάνω από οποιοδήποτε σημείο διακοπής στο επίπεδο γείωσης. Αυτό προκαλεί ασυνέχεια στην επιστροφή των σημάτων, και μπορεί να οδηγήσει σε προβλήματα EMI. Αποφεύγονται επίσης οι διακοπές στο επίπεδο γείωσης, όπου βέβαια είναι δυνατόν. Μια ασυνέχεια λειτουργεί διαφορετικά σε ένα διαχωρισμένο επίπεδο, όταν τα ίχνη σημάτων υψηλής συχνότητας είναι συνεχή.
- Τοποθετείται ένας πυκνωτής ανά ακίδα (pin) τροφοδοσίας.
- Αν είναι δυνατό, συνδέεται η ακίδα τροφοδοσίας του ολοκληρωμένου κυκλώματος στον πυκνωτή παράκαμψης πρώτα, και έπειτα στο επίπεδο τροφοδοσίας. Αυτό μειώνει το θόρυβο μετατροπής στο επίπεδο τροφοδοσίας.
- Πρέπει να είναι γνωστό ότι τα vias θα προκαλέσουν ασυνέχειες στη χαρακτηριστική σύνθετη αντίσταση μιας γραμμής μετάδοσης.
- Για να ελαχιστοποιηθεί η διασταύρωση μεταξύ δύο ιχνών επάνω σε ένα επίπεδο γείωσης, ελαχιστοποιείται η απόσταση μεταξύ επιπέδου γείωσης και ίχνους, και μεγιστοποιείται η απόσταση μεταξύ των ιχνών.
- Τα vias μικρών διαμέτρων έχουν χαμηλότερη παρασιτική αυτεπαγωγή, και γι' αυτό προτιμούνται όταν η συχνότητα γίνεται υψηλή.
- Η τάση εισόδου δεν συνδέεται απευθείας στα επίπεδα τροφοδοσίας, αλλά μέσω βασικών φίλτρων πυκνωτών.

### Φόρτωμα διπλής όψης (double sided loading)

Το φόρτωμα με υλικά και των δύο πλευρών ενός PCB μπορεί να έχει πολλά οφέλη. Πράγματι, γίνεται όλο και πιο δημοφιλές και είναι απαραίτητη επιλογή κατά την απεικόνιση ( layout ) μιας πλακέτας. Υπάρχουν δύο κύριοι παράγοντες που οδηγούν στο γέμισμα και των δύο όψεων. Ο πρώτος είναι το μέγεθος της πλακέτας. Δηλαδή εάν απαιτείται ένα ορισμένο μέγεθος πλακετών, και τα υλικά δεν χωράνε σε μια πλευρά, τότε χρησιμοποιείται και η άλλη όψη της. Ο δεύτερος λόγος είναι ότι πρέπει να τηρηθούν ορισμένες ηλεκτρικές προδιαγραφές. Συχνά, με τα υλικά τοποθετημένα πυκνά πάνω σε μια πλακέτα δεν υπάρχει χώρος για τους απαραίτητους πυκνωτές παράκαμψης που πρέπει να τοποθετηθούν αρκετά κοντά στην εκάστοτε συσκευή (ολοκληρωμένο) ώστε να είναι αποτελεσματικοί.

### Αυτόματη συνδεσμολογία (Auto Routing)

Η αυτόματη συνδεσμολογία είναι η διαδικασία όπου το λογισμικό PCB συνδέει αυτόματα τα ίχνη. Θα προσπαθήσει να συνδέσει ολόκληρη την πλακέτα αν του επιτραπεί. Αυτό το κάνουν τα περισσότερα πακέτα PCB ενώ η τεχνολογία και η θεωρία πίσω από τις τεχνικές αυτόματης συνδεσμολογίας μπορεί να είναι ιδιαίτερα περίπλοκες (τεχνητή νοημοσύνη, τεχνολογία νευρωνικών δικτύων). Ανεξάρτητα όμως από το πόσο 'έξυπνος' είναι ένας αυτόματος router, δεν μπορεί να αντικαταστήσει έναν καλό σχεδιαστή PCB.

### Αυτόματη τοποθέτηση (auto placement)

Τα εργαλεία αυτόματης τοποθέτησης είναι διαθέσιμα σε πολλά πακέτα PCB. Η λειτουργία αυτή δεν οδηγεί σχεδόν ποτέ στο βέλτιστο αποτέλεσμα για την πλακέτα. Αυτά τα εργαλεία εντούτοις, έχουν μια χρήσιμη λειτουργία, αφού δείχνουν, με εύκολο τρόπο, πως μπορούν να απλωθούν τα υλικά σε ολόκληρη την πλακέτα.

### Συγκόλληση (Soldering)

Η σπουδαιότητα της συγκόλλησης πρέπει να λαμβάνεται υπόψη κατά την απεικόνιση ( layout ) της πλακέτας. Υπάρχουν τρεις βασικές τεχνικές συγκόλλησης – χειρός, κύματος και επανακύλισης. Η συγκόλληση επανακύλισης είναι η πιο πρόσφατη τεχνική, και είναι κατάλληλη για όλα τα υλικά που τοποθετούνται στην επιφάνεια. Η συγκόλληση κυμάτων έχει το πλεονέκτημα του χαμηλού κόστους, αλλά το μειονέκτημα της επιβολής ορίων τοποθέτησης στα υλικά. Η συγκόλληση επανακύλισης είναι

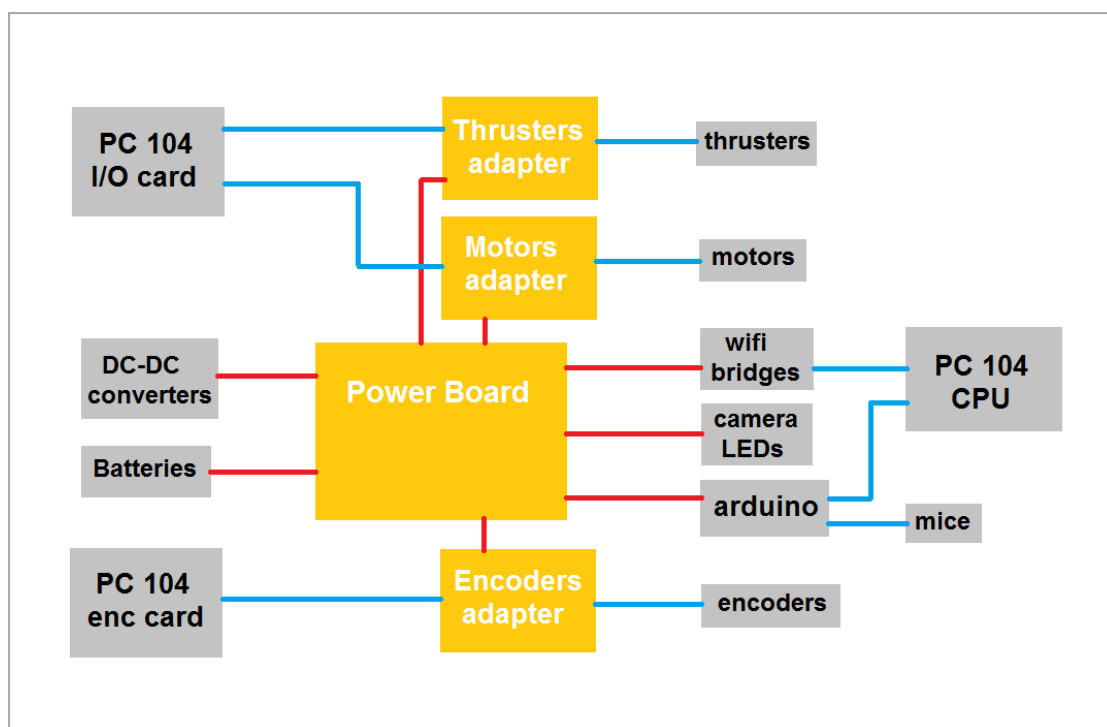
πιο σύνθετη και ακριβή, αλλά επιτρέπει την πολύ πυκνή τοποθέτηση των υλικών στην επιφάνεια. Η συγκόλληση χειρός είναι η παραδοσιακή μέθοδος που χρησιμοποιείται συνήθως για τα πρωτότυπα και τα μικρά πακέτα παραγωγής.

### Υποβολή του σχεδίου για παραγωγή

Το πρώτο πράγμα που πρέπει να ξέρουμε είναι το format με το οποίο θα αποσταλεί το αρχείο PCB. Η αποστολή του αρχικού αρχείου PCB θα εξασφαλίσει πως, ότι βλέπουμε στην οθόνη είναι αυτό που θα πάρουμε όταν παραδοθεί η πλακέτα. Αν αυτό δεν είναι εφικτό, τα αρχεία Gerber είναι το παραδοσιακό και αναγνωρισμένο από τη βιομηχανία format αρχείων, που όλοι οι σημαντικοί κατασκευαστές θα δεχτούν. Πολλοί σχεδιαστές PCB επιμένουν ακόμα να παράγουν και να παρέχουν τα αρχεία Gerber οι ίδιοι, προκειμένου να υπάρξει συνολικός έλεγχος της διαδικασίας κατασκευής. Σε όλες εκτός από μερικές ακραίες περιπτώσεις, η παραγωγή των αρχείων Gerber δεν είναι απαραίτητη και ανήκει στο παρελθόν. Καλό είναι να αποφεύγουμε τη χρήση των αρχείων αυτών αν είναι δυνατό, γιατί μπορεί να γίνουν εστία προβλημάτων, εκτός αν ξέρουμε ακριβώς πώς να τα παράγουμε σωστά. Γενικά, ο κατασκευαστής θα ζητήσει πολλές πληροφορίες πριν την εκτύπωση, ενώ τα αρχεία θα πρέπει να έχουν συγκεκριμένη μορφή, μετά από συνεννόηση μαζί του.

### 7.4 Συνολικό Ηλεκτρικό/ Ηλεκτρονικό Υποσύστημα του Ρομπότ

Από το Σχήμα 7.3 φαίνεται ποιες είναι οι απαραίτητες πλακέτες για την συνολική λειτουργία του συστήματος (κίτρινα μπλοκ). Από αυτές, αρχικά απαραίτητες είναι μόνο η πλακέτα διανομής ισχύος (Power Board) και η πλακέτα για την οδήγηση των βαλβίδων των προωθητήριων (thrusters). Το υποσύστημα των κινητήρων δεν υλοποιήθηκε στην παρούσα εργασία, οπότε και οι αντίστοιχες πλακέτες (motors και encoders adapters) είναι προτιμότερο να σχεδιαστούν συνολικά στο μέλλον. Επιπλέον, σχεδιάστηκε εκ νέου και εκτυπώθηκε το Arduino Shield για λόγους ασφάλειας και στιβαρότητας της κατασκευής (αλλιώς θα έμενε στη μορφή διάτρητης πλακέτας).



Σχήμα 7.3 Συνολικό ηλεκτρολογικό/ηλεκτρονικό υποσύστημα.



Γενικά η λειτουργία του ηλεκτρικού/ ηλεκτρονικού υποσυστήματος είναι η εξής: η ενέργεια εξασφαλίζεται μέσω μπαταριών και διαμοιράζεται στα υποσυστήματα χαμηλής και υψηλής τάσης. Η ακριβής κυκλοφορία της ισχύος και ο αντίστοιχος σχεδιασμός των πλακετών περιγράφεται στη συνέχεια.

## 7.5 Πλακέτα Διανομής Ισχύος (Power Board)

Αυτή είναι η κεντρική πλακέτα του ρομπότ, όπου συνδέονται σαν είσοδοι οι μπαταρίες (χαμηλής και υψηλής τάσης) και μέσω της οποίας η ισχύς διανέμεται κατάλληλα σε όλα τα υποσυστήματα. Βασικό χαρακτηριστικό της είναι η σύνδεση της με έναν αυτοσχέδιο πίνακα ελέγχου (Control Panel), από τον οποίο καθορίζεται η τροφοδοσία του κάθε υποσυστήματος μέσω ξεχωριστών διακοπών και ενδεικτικών LEDs. Ο πίνακας ελέγχου παρουσιάζεται πιο κάτω.

### Βασικές Λειτουργίες

Παρακάτω γίνεται ένας διαχωρισμός των εισόδων και των εξόδων της πλακέτας διανομής ισχύος. Ο ρόλος της πλακέτας είναι ο κατάλληλος χειρισμός των εισόδων έτσι, ώστε να προκύψουν οι κατάλληλες έξοδοι στις οποίες θα συνδεθούν κατάλληλα όλα τα υποσυστήματα του ρομπότ.

Είσοδοι:

- Μπαταρία LiPO χαμηλής τάσης (~8V)
- Μπαταρίες LiPO υψηλής τάσης (2x 16V) και σύνδεση τους σε σειρά για να προκύψουν 33V
- Ρυθμισμένη χαμηλή τάση 5V (μετά τον DC-DC converter χαμηλής τάσης)
- Ρυθμισμένη χαμηλή τάση 12V (μετά τον DC-DC converter χαμηλής τάσης)
- Ρυθμισμένη υψηλή τάση 24V (μετά τον DC-DC converter υψηλής τάσης)
- Σήμα RF από το Panic Button
- Σήμα από κεντρικό υπολογιστή

Έξοδοι:

- Χαμηλή τάση από μπαταρία (~8V) προς DC-DC converter
- Υψηλή τάση (~33V) προς DC-DC converter υψηλής τάσης
- Τροφοδοσία υψηλής τάσης 24V της πλακέτας των thrusters
- Τροφοδοσία υψηλής τάσης 24V της πλακέτας κινητήρων (βραχίονες και σφόνδυλος αντίδρασης)
- Τροφοδοσία χαμηλής τάσης 5V των διόδων φωτοεκπομπής LEDs της οροφής
- Τροφοδοσία χαμηλής τάσης 12V του συστήματος των οπτικών αισθητήρων
- Τροφοδοσία χαμηλής τάσης 5V των 2 Wireless Ethernet Bridges

Έξοδοι γενικής χρήσης: χαμηλής τάσης 5V, υψηλής τάσης 5V, χαμηλής τάσης 12V, υψηλής τάσης 24V, γείωση κυκλώματος χαμηλής τάσης, γείωση κυκλώματος υψηλής τάσης.

### Λειτουργία Πλακέτας

Η πλακέτα αποτελείται από 2 κυκλώματα, το κύκλωμα χαμηλής τάσης, και το κύκλωμα υψηλής τάσης. Το χαμηλής τάσης είναι υπεύθυνο για την τροφοδοσία των υπολογιστικών συστημάτων (απαιτούνται μικρά ρεύματα), ενώ το υψηλής τάσης για την τροφοδοσία των επενεργητών του συστήματος (απαιτούνται μεγάλα ρεύματα). Αυτά για λόγους ασφαλείας είναι πλήρως διαχωρισμένα μεταξύ τους, δεν υπάρχει δηλαδή κοινή γείωση (common ground). Ενώ λοιπόν δεν υπάρχει κανένα κοινό σημείο των 2 κυκλωμάτων, δίνεται η επιλογή στο χρήστη να μπορεί με κατάλληλο jumper (βραχυκύκλωμα δύο ακίδων) να ενώσει τις γειώσεις χαμηλής και υψηλής τάσης σε μία κοινή (common ground). Η δυνατότητα αυτή δίνεται, γιατί συνηθίζεται σε τέτοια συστήματα να υπάρχει κοινή αναφορά στις 2 τάσεις.

Σημαντικό χαρακτηριστικό είναι ο τρόπος που ελέγχεται η ροή ισχύος στο κύκλωμα της υψηλής τάσης. Η υψηλή τάση των 24V διανέμεται στα υποσυστήματα υψηλής τάσης μόνο αν: το αντίστοιχο ρελέ και ο αντίστοιχος μηχανικός διακόπτης στον πίνακα ελέγχου είναι κλειστά. Το ρελέ κλείνει το

κύκλωμα μόνο αν πάρει τα κατάλληλα σήματα από τη CPU του ρομπότ και από το χειριστήριο του panic button. Το χειριστήριο panic button έχει ήδη κατασκευαστεί για το πρώτο ρομπότ, και έτσι το κύκλωμα ενσωματώθηκε απαράλλακτο σε αυτή τη πλακέτα. Όταν το panic button πατηθεί αυτόματα τα κυκλώματα υψηλής τάσης των δύο ρομπότ ανοίγουν. Η σύνδεση δηλαδή με τις μπαταρίες υψηλής ισχύος θα γίνει μόνο αν δεν έχει πατηθεί το panic button, αν δεν υπάρχει κάποιο πρόβλημα από τη CPU του ρομπότ και αν ο αντίστοιχος μηχανικός διακόπτης στον πίνακα ελέγχου είναι κλειστός.

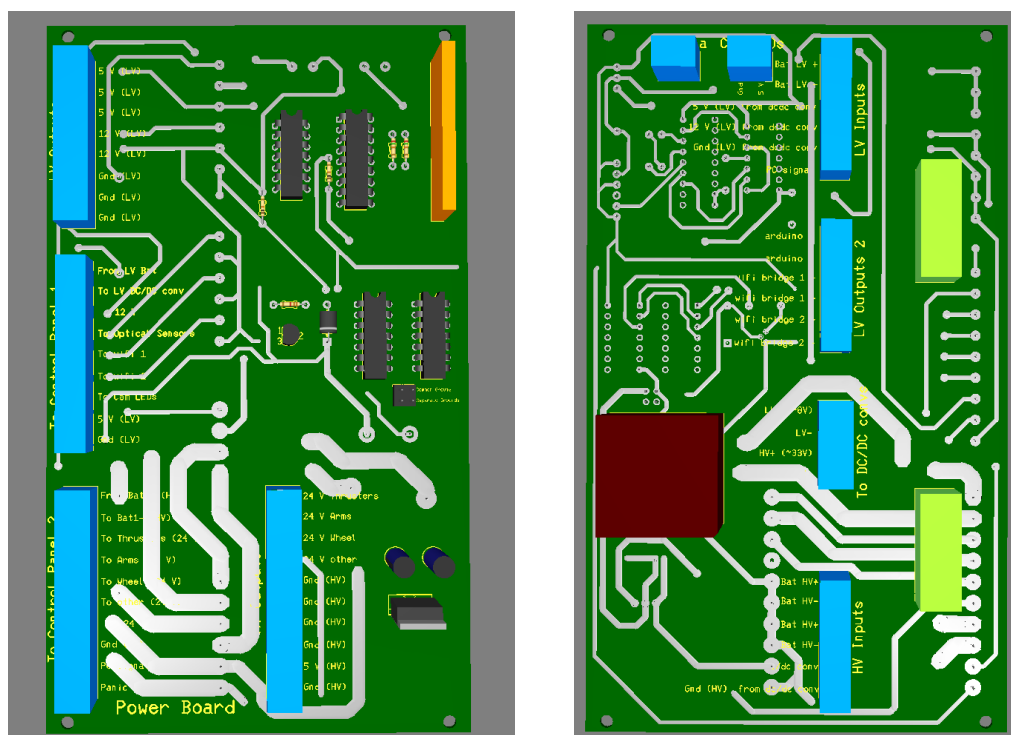
#### Ιδιαίτερα ηλεκτρονικά και ολοκληρωμένα κυκλώματα (ICs) που χρησιμοποιήθηκαν

- Ρελέ (Relay) – διακόπτης του κυκλώματος υψηλής τάσης
- Rf (Radio Frequency) receiver και decoder – λαμβάνει σήμα ασύρματα από το χειριστήριο του panic button. Το κύκλωμα είναι ίδιο με αυτό του πρώτου ρομπότ.
- Ασφάλειες (fuses) – προστατεύουν τη πλακέτα από μεγάλα ρεύματα
- Τρανζίστορ και δίοδος – αυξάνουν την ένταση των σημάτων που ενεργοποιούν το ρελέ
- Λογικές πύλες NOT και AND – χειρίζονται τα σήματα της CPU και του panic button
- Ρυθμιστής τάσης (Voltage regulator) – ρυθμίζει την υψηλή τάση στα 5V για χρήση στο ρελέ
- Πυκνωτές και αντιστάσεις – χρησιμοποιούνται συμπληρωματικά στα παραπάνω κυκλώματα

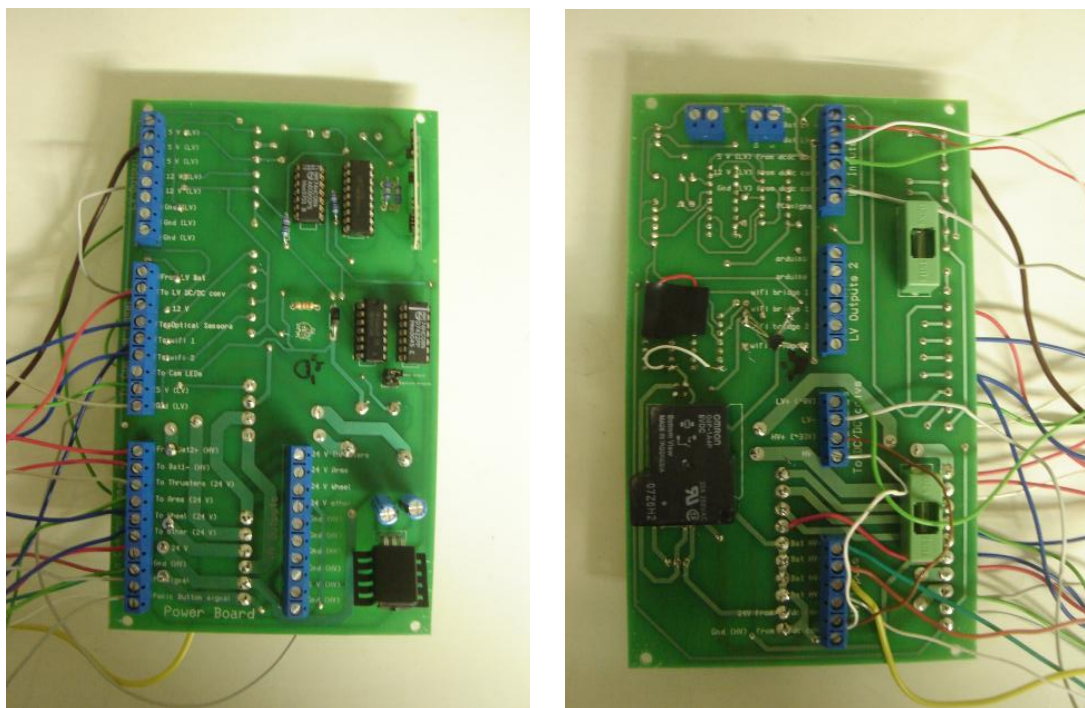
Τα ακριβή τεχνικά χαρακτηριστικά τους περιγράφονται στα datasheets τους, τα οποία εμπεριέχονται στο CD της εργασίας.

#### Ιδιαίτερα κατασκευαστικά χαρακτηριστικά

Έγινε προσπάθεια τήρησης των κανόνων και τεχνικών καλού ηλεκτρονικού σχεδιασμού που παρουσιάστηκαν παραπάνω. Αυτό ήταν επιτακτικό για το λόγο του ότι σε αυτή την πλακέτα εμπεριέχονται τάσεις της τάξης των 30 V και μεγάλα ρεύματα αλλά και σήματα RF. Δόθηκε λοιπόν μεγάλη προσοχή στην διατήρηση της ακεραιότητας των υψίσουχων αυτών σημάτων, μέσω των αντίστοιχων τεχνικών σχεδιασμού. Για τις εισόδους και τις εξόδους χρησιμοποιήθηκαν κλέμες. Σε όλα τα σημεία σύνδεσης, αναγράφονται οι απαραίτητες πληροφορίες στο επίπεδο silkscreen (άσπρο χρώμα). Τέλος, λόγω των πολλών εξαρτημάτων χρησιμοποιήθηκαν και οι 2 όψεις της πλακέτας (βλ. Σχήματα 7.4 και 7.5).



Σχήμα 7.4 Οι δύο όψεις της πλακέτας ισχύος σε 3D όπως παράχθηκαν από το NI Ultiboard.

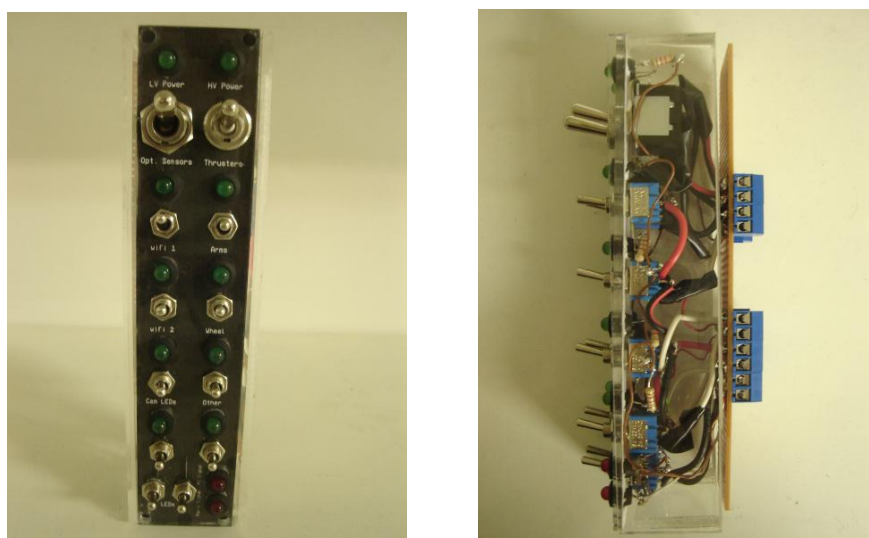


Σχήμα 7.5 Οι δύο όψεις της πλακέτας ισχύος, αφού έχουν γίνει οι κολλήσεις των ηλεκτρονικών.

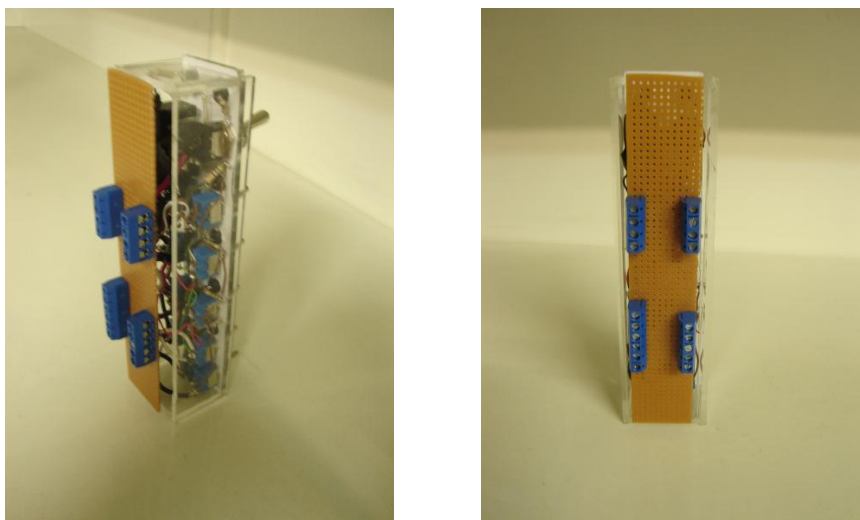
## 7.6 Πίνακας ελέγχου (Control panel)

Ο πίνακας αυτός αντιστοιχεί στο μπλοκ Switches του σχηματικού της κάρτας διανομής ισχύος (ΠΑΡΑΡΤΗΜΑ Β). Είναι ένας πίνακας με διακόπτες και ενδεικτικά LEDs, μέσω του οποίου ελέγχεται η ροή ισχύος σε όλα τα υποσυστήματα του ρομπότ. Τα πράσινα LEDs δείχνουν τα υποσυστήματα που βρίσκονται κάθε στιγμή σε λειτουργία, ενώ τα δύο κόκκινα LEDs ανάβουν αν σταλθεί σήμα λάθους από το panic button ή από τη CPU. Τέλος, υπάρχει η δυνατότητα αποσύνδεσης των LEDs, μέσω δύο διακοπών. Το νόημα αυτού είναι η αποφυγή προβλημάτων φωτεινότητας κατά τη διάρκεια πειραμάτων με το σύστημα της κάμερας (η κάμερα προσπαθεί να εντοπίσει όλα τα φωτεινά σημεία).

Κατασκευάστηκε από Plexiglas και μια διάτρητη πλακέτα (βλ. Σχήματα 7.6, 7.7). Χρησιμοποιήθηκαν κατάλληλοι διακόπτες και αντιστάσεις ανάλογα με την τάση και το ρεύμα του κάθε κυκλώματος. Οι κλέμες τοποθετήθηκαν με τη σειρά που πρέπει να συνδεθούν και στην πλακέτα διανομής ισχύος. Κατασκευάστηκε με μόνο στόχο να προσδώσει ευκολία στη χρήση και στην εποπτεία του ρομποτικού συστήματος.



Σχήμα 7.6 Δύο όψεις του πίνακα ελέγχου του ρομπότ.



**Σχήμα 7.7** Άλλες όψεις του πίνακα ελέγχου του ρομπότ.

### 7.7 Πλακέτα οδήγησης βαλβίδων των προωθητηρίων (Thrusters Board)

Η πλακέτα αυτή έχει τον ρόλο της οδήγησης των βαλβίδων που χρησιμοποιούν τα προωθητήρια. Είναι σχεδιασμένη έτσι ώστε να δέχεται σαν εισόδους τα κατάλληλα σήματα 5V και να τα μετατρέπει σε παλμούς τάσης 24V. Αυτοί είναι οι απαιτούμενοι παλμοί για την λειτουργία των βαλβίδων. Τα 24V αντιστοιχούν σε ανοιχτή βαλβίδα, ενώ τα 0V σε κλειστή. Εναλλάσσοντας τις τιμές αυτές σε μεγάλες συχνότητες, γίνεται δυνατή η μεταβολή της ροής του αερίου CO<sub>2</sub> που θα οδηγηθεί στα thrusters και θα δώσει κίνηση στο ρομπότ.

#### Βασικές Λειτουργίες

Είσοδοι:

- Υψηλή τάση 24V από την πλακέτα διανομής ισχύος
- Υψηλή τάση 5V από την πλακέτα διανομής ισχύος
- Γείωση υψηλής τάσης από την πλακέτα διανομής ισχύος
- Γείωση χαμηλής τάσης από την πλακέτα διανομής ισχύος
- 8 σήματα PWM από την κάρτα εισόδων/ εξόδων του υπολογιστικού συστήματος του ρομπότ.

Έξοδοι:

- 8 παλμοί 24V προς τις βαλβίδες των προωθητηρίων

#### Λειτουργία

Ο διαχωρισμός των κυκλωμάτων υψηλής και χαμηλής τάσης διατηρείται και σε αυτή την πλακέτα. Αυτό επιτυγχάνεται μέσω ειδικών ολοκληρωμένων κυκλωμάτων που λέγονται optocouplers. Αυτά ενσωματώνουν μέσα τους ένα σύστημα LEDs-φωτοευαίσθητων διόδων και έτσι γίνεται δυνατή η επικοινωνία στις 2 πλευρές ενός κυκλώματος χωρίς να υπάρχει στην ουσία επαφή μεταξύ τους. Πιο συγκεκριμένα, οι παλμοί χαμηλής τάσης 5V PWM μετατρέπονται μέσω των optocouplers σε παλμούς 5V υψηλής τάσης, οι οποίοι ενισχύονται με χρήση ολοκληρωμένων κυκλωμάτων Η-γεφυρών (H-Bridges) στα 24V και τελικά συνδέονται στις βαλβίδες.

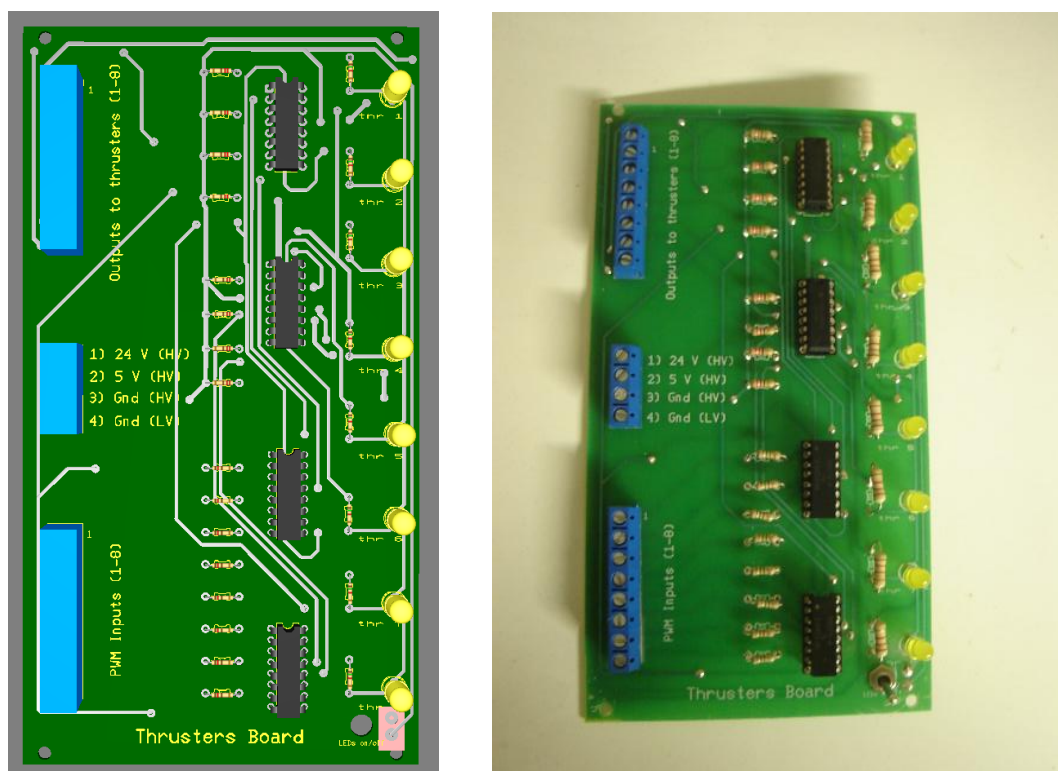
#### Ιδιαίτερα ηλεκτρονικά και ολοκληρωμένα κυκλώματα (ICs) που χρησιμοποιήθηκαν

- Optocouplers – διαχωρίζουν το κύκλωμα σε χαμηλής και υψηλής τάσης
- H-bridge – ενισχύει τους παλμούς από 5V σε 24V

- LEDs – ενδεικτικά, δείχνουν ποια βαλβίδα λειτουργεί και με τι ρυθμό. Απενεργοποιούνται με κατάλληλο διακόπτη για αποφυγή προβλημάτων φωτεινότητας κατά την διάρκεια πειραμάτων με το σύστημα της κάμερας.

### Ιδιαίτερα κατασκευαστικά χαρακτηριστικά

Αποτελεί ένα δύσκολο κύκλωμα καθώς απαιτεί πολλές συνδέσεις σε μικρό χώρο. Για να αντιμετωπιστεί αυτό χρησιμοποιήθηκαν αρκετά νίσι. Για τις εισόδους και τις εξόδους χρησιμοποιήθηκαν ξανά κλέμες και σε όλες τις θέσεις τυπώθηκαν σε επίπεδο silkscreen οι κατάλληλες πληροφορίες σύνδεσης (βλ. Σχήμα 7.8).



**Σχήμα 7.8** Αριστερά: Η πλακέτα οδήγησης των thrusters σε 3D όψη όπως παράχθηκε από το NI Ultiboard. Δεξιά: Η έτοιμη πλακέτα μετά τις κολλήσεις των ηλεκτρονικών.

### 7.8 Πλακέτα οπτικών αισθητήρων (Optical Sensors Arduino Shield)

Ο ρόλος της πλακέτας αυτής (βλ. Σχήματα 7.8 και 7.9) έχει περιγραφεί αρχικά στο κεφάλαιο των οπτικών αισθητήρων. Ο σχεδιασμός και η εκτύπωση που γίνεται εδώ αποτελεί μια δεύτερη υλοποίηση, που έρχεται να αντικαταστήσει αυτή της διάτρητης πλακέτας, που περιγράφηκε προηγουμένως.

#### Βασικές Λειτουργίες

Είσοδοι:

- 3 οπτικοί αισθητήρες PS/2
- 12V χαμηλής τάσης

Έξοδοι:

- RS-232 σειριακή επικοινωνία με CPU

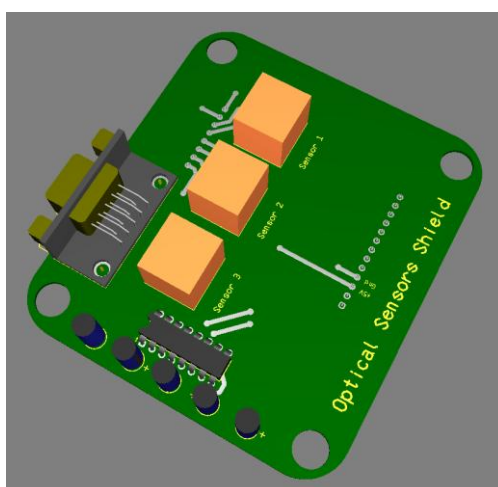


### Ιδιαίτερα ηλεκτρονικά και ολοκληρωμένα κυκλώματα (ICs) που χρησιμοποιήθηκαν

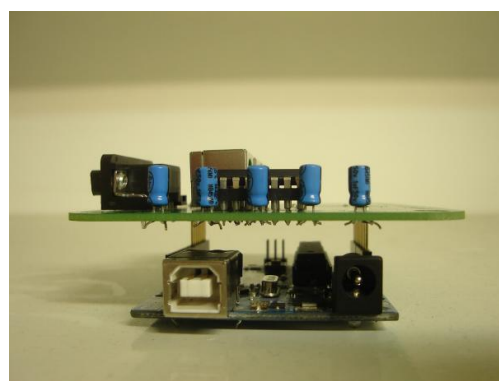
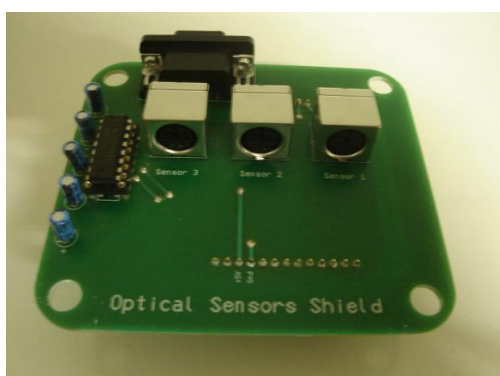
- max232 – ολοκληρωμένο (IC) που χειρίζεται τη σειριακή επικοινωνία
- Θύρες PS/2 – για τη σύνδεση των οπτικών αισθητήρων
- Θύρα RS-232 – για την τελική αποστολή των δεδομένων στη CPU
- Πυκνωτές – απαιτούνται για την ομαλή λειτουργία του κυκλώματος

### Ιδιαίτερα κατασκευαστικά χαρακτηριστικά

Η ιδιαιτερότητα σε αυτή την πλακέτα είναι η χρήση επιπέδου γείωσης. Η μία πλευρά της είναι καλυμμένη από χαλκό και συνδέεται με τη γείωση του κυκλώματος. Αυτό αποτελεί μια καλή τεχνική για την εξάλειψη θορύβου, όπως αναλύθηκε στις αρχές του σωστού σχεδιασμού.

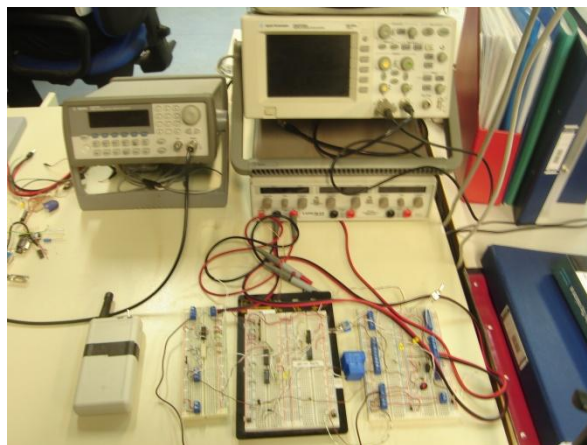


**Σχήμα 7.8** 3D όψη του Arduino Shield όπως παράχθηκε από το NI Ultiboard.



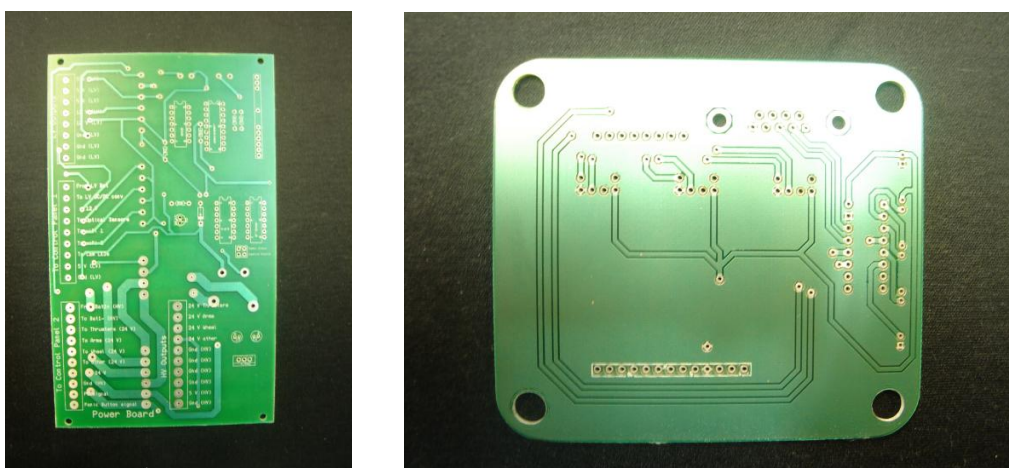
**Σχήμα 7.9** Το Optical Sensors Shield έτοιμο και συνδεδεμένο στην πλατφόρμα Arduino.

Στο Σχήμα 7.10 φαίνεται ο τρόπος που τα κυκλώματα των πλακετών δοκιμάστηκαν στο εργαστήριο πριν ακόμα σταλθούν για εκτύπωση. Η ορθή λειτουργία τους διαπιστώθηκε χρησιμοποιώντας τον κατάλληλο εξοπλισμό του εργαστηρίου. Κατά τη διάρκεια των πειραμάτων, η ισχύς δινόταν στα κυκλώματα μέσω τροφοδοτικού, τα σήματα μέσω παλμογεννήτριας και τέλος γίνονταν μετρήσεις μέσω παλμογράφου και πολυμέτρου. Η διαδικασία αυτή είναι πάντα απαραίτητη στη διαδικασία του ηλεκτρονικού σχεδιασμού, γι' αυτό και κρίθηκε σκόπιμο να θιχτεί σε αυτό το σημείο.



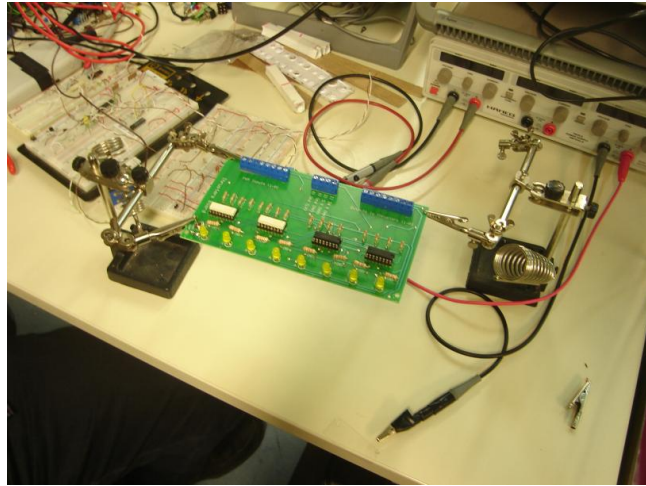
**Σχήμα 7.10** Δοκιμή των κυκλωμάτων στο εργαστήριο πριν την αποστολή για εκτύπωση.

Οι πλακέτες μετά την εκτύπωση έχουν την όψη που φαίνεται στο Σχήμα 7.11. Το βήμα που θα ολοκληρώσει τυπικά τη διαδικασία ανάπτυξης των ηλεκτρονικών πλακετών είναι η σωστή τοποθέτηση και κόλληση των ηλεκτρονικών. Αυτή είναι μια διαδικασία που απαιτεί καθαρό μυαλό κατά την τοποθέτηση και λίγη εμπειρία κατά την κόλληση.



**Σχήμα 7.11** Πλακέτες όπως παραλήφθηκαν μετά την εκτύπωση.

Ένα άλλο στάδιο στην ανάπτυξη του συστήματος των πλακετών που είναι επίσης απαραίτητο, είναι η δοκιμή των πλακετών μετά την εκτύπωση και πριν ακόμη τοποθετηθούν στο ρομπότ και συνδεθούν στο πραγματικό hardware. Δεν είναι καθόλου σπάνια η εύρεση λαθών ακόμη και σε αυτό το επίπεδο. Η δοκιμή γίνεται πάλι με τα ίδια όργανα του εργαστηρίου όπως φαίνεται στο Σχήμα 7.12.



*Σχήμα 7.12* Δοκιμή των πλακετών στο εργαστήριο πριν την τοποθέτηση στο ρομπότ.

Στην εικόνα του Σχήματος 7.13 οι πλακέτες έχουν τελικά τοποθετηθεί στο ρομπότ. Εδώ φαίνεται ότι οι διαστάσεις των πλακετών δεν επιλέχθηκαν τυχαία, αλλά προκείμενου να καλύψουν τις ειδικές για το σκοπό αυτό θέσεις στο σασί. Όπως φαίνεται έχουν τη δυνατότητα να μπαίνουν συρταρωτά και να σταθεροποιούνται με τις κατάλληλες σπές εγκατάστασης που ανοίχτηκαν κατά την εκτύπωσή τους.



*Σχήμα 7.13* Οι πλακέτες τοποθετημένες στο ρομπότ.



# ΚΕΦΑΛΑΙΟ 8

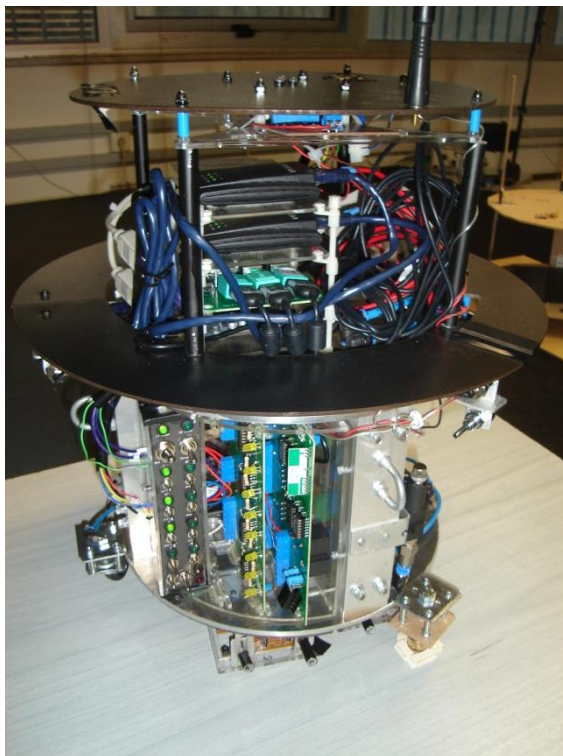
## Ρομπότ σε τελική μορφή - Βαθμονόμηση και πειράματα

### 8.1 Το ρομπότ σε τελική μορφή

Στο κεφάλαιο αυτό, το ρομπότ παρουσιάζεται στην τελική του μορφή. Ενσωματώνονται δηλαδή όλα τα υποσυστήματα που αναπτύχθηκαν σε αυτή την εργασία, και τίθενται σε λειτουργία. Στήνονται διάφορα πειράματα κατά τα οποία δοκιμάζεται ο χειρισμός του ρομπότ μέσω του xPC Target και αξιολογούνται οι αλγόριθμοι εντοπισμού θέσης που αναπτύχθηκαν στα προηγούμενα κεφάλαια. Στα πειράματα αυτά, το ρομπότ κινείται πάνω σε ειδικά ροδάκια που έχουν πάρει τη θέση των αεροδράνων (βλ. Σχήμα 8.1). Αυτό γίνεται καταρχήν για την εξοικονόμηση του αερίου CO<sub>2</sub> και δεύτερον λόγω του ασύγκριτα ευκολότερου χειρισμού του όταν αυτό πατάει στο έδαφος έναντι του χειρισμού όταν αυτό αιωρείται. Δίνεται λοιπόν η δυνατότητα για απεριόριστο αριθμό πειραμάτων και σε διαφορετικές επιφάνειες (όχι μόνο στον γρανίτη). Τα πειράματα βέβαια αυτά δεν αντικαθιστούν επάξια τα πειράματα στην τράπεζα του γρανίτη. Άλλωστε είναι ένα σύστημα που έχει σχεδιαστεί σύμφωνα με αυτές τις συγκεκριμένες απαιτήσεις. Το κέρδος όμως, είναι τα πολλά και σημαντικά συμπεράσματα ως προς τις γενικές αρχές των αλγορίθμων και της κίνησης του ρομπότ. Σημαντική είναι επίσης η δυνατότητα του γρήγορου εντοπισμού των διαφόρων λαθών χωρίς να ξοδευτεί καθόλου αέριο. Σημειώνεται τέλος ότι τα κανονικά πειράματα αιώρησης πάνω στην

τράπεζα του γρανίτη, συνιστούν μια περίπλοκη και χρονοβόρα διαδικασία, η οποία μάλιστα απαιτεί την ενασχόληση τουλάχιστον τριών ατόμων της ομάδας. Τα πειράματα που περιγράφονται παρακάτω έγιναν από ένα μόνο άτομο (τον συγγραφέα) και σε μικρό χρονικό διάστημα.

Τελικά δεν έμεινε χρόνος για πειράματα με αεροέδρανα. Τα συμπεράσματα όμως που προέκυψαν είναι αρκετά χρήσιμα και βάζουν τα θεμέλια για μια σειρά μελλοντικών πειραμάτων στις κανονικές συνθήκες του επίπεδου διαστημικού εξομοιωτή.



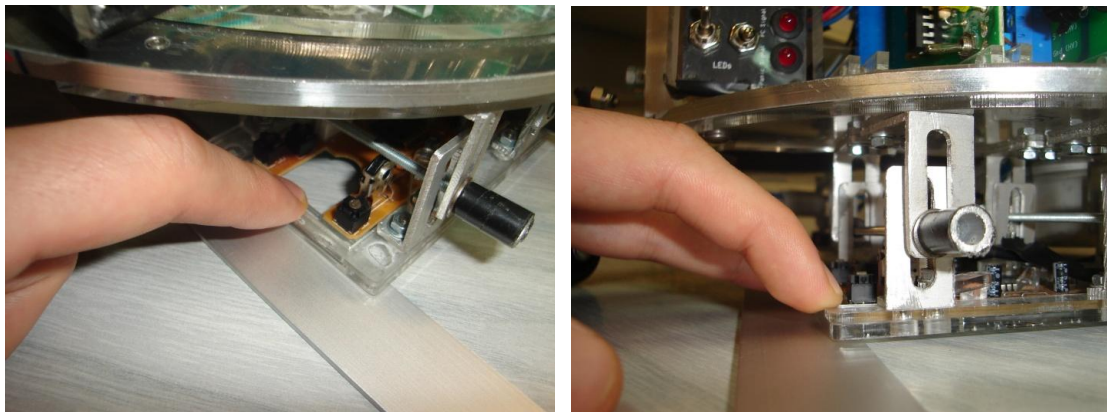
**Σχήμα 8.1** Ρομπότ στην τελική του μορφή, με όλα τα συστήματά του ενσωματωμένα κατά τη διάρκεια πειράματος με ροδάκια πάνω σε ξύλινη βάση.

## 8.2 Βαθμονόμηση Συστήματος Οπτικών Αισθητήρων

Μετά την ενσωμάτωση του συστήματος των οπτικών αισθητήρων στο ρομπότ και πριν την έναρξη των πειραμάτων είναι απαραίτητη μια διαδικασία βαθμονόμησης. Εκεί που πρέπει να δοθεί ιδιαίτερη προσοχή, είναι στη ρύθμιση της απόστασης των αισθητήρων από το επίπεδο κίνησης.

Η απόσταση του αισθητήρα από το επίπεδο είναι μια σημαντική παράμετρος. Παρατηρήθηκε ότι στην συσκευή (ποντίκι), όπως ήταν στην αρχική της μορφή, το ύψος αυτό ήταν 2[mm]. Σε πειράματα που έγιναν, παρατηρήθηκε ότι τα counts του αισθητήρα εξαρτώνται από την απόσταση αυτή. Πιο, συγκεκριμένα, ο αισθητήρας πράγματι είναι αποδοτικός και ακριβής σε απόσταση κοντά στα 2[mm] από το επίπεδο. Τοποθετώντας τους αισθητήρες σε μικρότερες ή μεγαλύτερες αποστάσεις μειώνεται η ακρίβεια και η ορθότητα των μετρήσεων, ενώ για μεγάλες αποκλίσεις καμία κίνηση δεν μπορεί να ανιχνευθεί.

Γίνεται λοιπόν κατανοητή η σημασία των αποστάσεων των αισθητήρων από το επίπεδο, και η ανάγκη αυτές να είναι βέλτιστες και αμετάβλητες κατά τη διάρκεια των πειραμάτων. Εδώ πρέπει να αναφερθεί ότι ο μηχανισμός της βάσης κάθε αισθητήρα επιτρέπει την εύκολη μεταβολή της απόστασης από το επίπεδο και ύστερα την σταθεροποίηση σε μία συγκεκριμένη απόσταση με χρήση ντίζας και περικοχλίων.



**Σχήμα 8.2** Ρύθμιση του ύψους του αισθητήρα στα 2 [mm] με χρήση κατάλληλου ελάσματος.

Ένας απλός τρόπος ρύθμισης των αποστάσεων των αισθητήρων από το επίπεδο είναι η χρήση ελασμάτων κατάλληλου πάχους, για τη ρύθμιση του ύψους των βάσεων. Έτσι, χρησιμοποιώντας μεταλλικά ελάσματα πάχους 2[mm], όπως φαίνεται στις φωτογραφίες, είναι δυνατή η ακριβής ρύθμιση αυτής της σημαντικής για το πείραμα παραμέτρου (βλ Σχήμα 8.2). Ενδεικνύεται η διαδικασία αυτή να γίνεται πριν από κάθε πείραμα, για πιο έγκυρα αποτελέσματα.

### 8.2.1 Συντελεστής μετατροπής [counts] σε [mm]

Όπως έχει ήδη αναφερθεί, η έξοδος κάθε οπτικού αισθητήρα είναι δύο αριθμοί μετρήσεων (counts), ανάλογα με την απόσταση κατά  $x$  και κατά  $y$  που διανύθηκε σε κάθε χρονικό βήμα. Προκειμένου να γίνει υπολογισμός της θέσης του ρομπότ σε [mm], είναι απαραίτητη η γνώση του συντελεστή μετατροπής των μετρήσεων αυτών σε αποστάσεις εκφρασμένες σε [mm]. Για να γίνει αυτό, επινοήθηκε το εξής πείραμα.

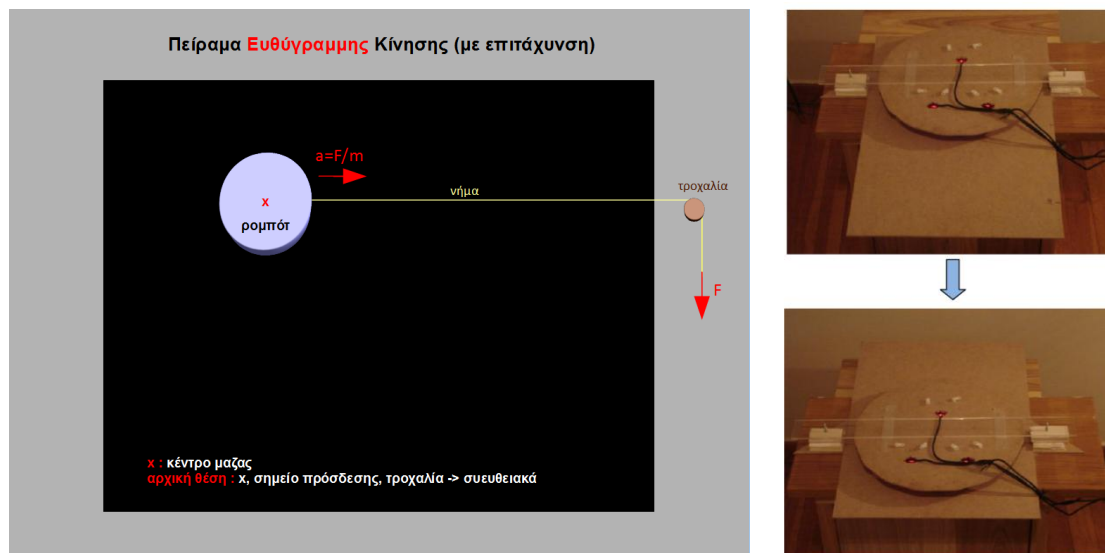
Όταν στην κίνηση ενός αισθητήρα δεν υπάρχει στροφή, τότε μπορεί να υπολογιστεί με ακρίβεια η θέση του στο επίπεδο, πράγμα αδύνατο στην μη ευθύγραμμη κίνηση. Με βάση το παραπάνω γεγονός, έγιναν διάφορα πειράματα κίνησης σε απόλυτη ευθεία (στα μέτρα του δυνατού) στη διεύθυνση ενός κάθε φορά από τους άξονες  $x, y$  των συστημάτων συντεταγμένων των αισθητήρων. Σε ένα τέτοιο πείραμα, η έξοδος ενός αισθητήρα αναμένεται μηδενική για τον άξονα που είναι κάθετος στην ευθύγραμμη κίνηση. Σαν αποτέλεσμα, η απόσταση που διανύθηκε περιέχεται μόνο στην μετατόπιση που αντιστοιχεί στον παράλληλο στην κίνηση άξονα. Αθροίζοντας όλες τις μικρές μετατοπίσεις που ελήφθησαν από τον αισθητήρα κατά τον άξονα αυτόν, προκύπτει το μήκος εκφρασμένο σε [counts] της ευθύγραμμης τροχιάς του πειράματος. Τελικά, ο υπολογισμός του συντελεστή μετατροπής γίνεται με απλή διαίρεση της πραγματικής μετατόπισης σε [mm] με την μετατόπιση σε [counts]. Λόγω των μηδενικών σχετικών γωνιών των συστημάτων συντεταγμένων των τριών αισθητήρων, δεν χρειάστηκε να γίνουν πειράματα για κάθε έναν ξεχωριστά. Με βάση τα παραπάνω, αναπτύχθηκε ειδικό πρόγραμμα για βαθμονόμηση των αισθητήρων (υπάρχει στο CD της εργασίας).

Το πρόβλημα σε αυτή τη διαδικασία ήταν η δυσκολία εκτέλεσης πειράματος ακριβούς ευθύγραμμης τροχιάς. Εφαρμόστηκαν διάφορες ιδέες για πειράματα ευθύγραμμης κίνησης (βλ. Σχήμα 8.3). Λόγω όμως του ότι, και με τέλεια σχεδιασμένο πείραμα, η απόλυτη ακρίβεια είναι αδύνατο να επιτευχθεί (οι αισθητήρες ανιχνεύουν την παραμικρή κίνηση σε επίπεδο μικρότερο του χιλιοστού), αποφασίστηκε η εκτέλεση πολλών ίδιων πειραμάτων και η στατιστική τους ανάλυση για την εξαγωγή αποτελεσμάτων.

Ένα πείραμα ευθύγραμμης κίνησης που επινοήθηκε αρχικά, ήθελε το ρομπότ να είναι τελείως ακίνητο ενώ, το επίπεδο από κάτω του να κινείται ευθύγραμμα ως προς αυτό (Σχήμα 8.3). Ήταν μία καλή ιδέα που έδωσε κάποια πρώτα πολύ χρήσιμα συμπεράσματα. Παραλλαγή της μπορεί σίγουρα να εφαρμοστεί βελτιωμένη στο μέλλον σε πειράματα με καλή αναμενόμενη ακρίβεια.

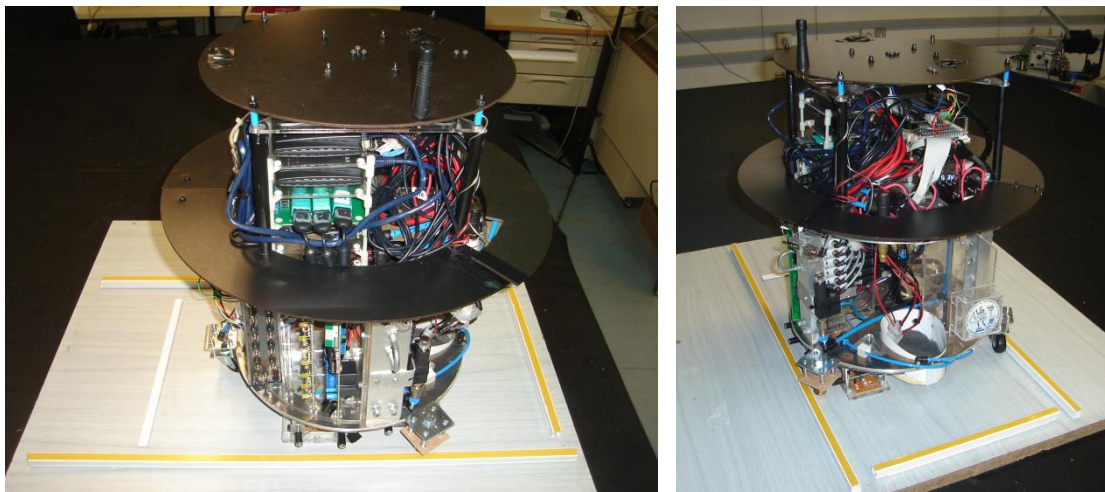
Μία άλλη ιδέα περιγράφεται πάλι στο Σχήμα 8.3 και εξηγείται ως εξής: Όταν σε ένα στερεό σώμα η συνισταμένη των δυνάμεων που του ασκούνται είναι σταθερή και ανήκει στην ευθεία που περνά από το κέντρο βάρους του, τότε λόγω μηδενικών ροπών, το σώμα εκτελεί ευθύγραμμη ομαλά επιταχυνόμενη κίνηση χωρίς περιστροφή, με επιτάχυνση  $a=F/m$ . Σε αυτή τη βάση, προσδένοντας ένα νήμα σε ένα σημείο της περιφέρειας του ρομπότ, περνώντας το νήμα από μια εξωτερική τροχαλία (με κατεύθυνση του νήματος όπως περιγράφηκε) και ασκώντας ελεγχόμενη δύναμη στο νήμα, η κίνηση του ρομπότ δεν μπορεί παρά να είναι ευθύγραμμη. Η ιδέα αυτή είναι ιδανική για υλοποίηση με χρήση αεροεδράνων υπό κανονικές συνθήκες λειτουργίας του εξομοιωτή (χωρίς ρόδες). Η φαινομενικά απόλυτη ακρίβεια, βέβαια θα μειωθεί λόγω μικρών παραγόντων που δεν μπορούν να προβλεφθούν σε θεωρητική βάση αλλά τελικά, η ευκολία στην κατασκευή και στην εκτέλεση καθώς και η τελική ακρίβεια της τροχιάς θεωρείται ότι υπερέρχουν των στοιχείων όλων των άλλων πειραμάτων που επινοήθηκαν.

Το πείραμα αυτό, λόγω πίεσης χρόνου δεν εκτελέστηκε, αλλά προτείνεται σαν μελλοντική εργασία. Τέλος, εναλλακτικές εφαρμογές του νήματος μπορούν να οδηγήσουν και σε θεωρητικά τέλεια κυκλική κίνηση, που όπως δείχνεται παρακάτω, ενδείκνυται για την αξιολόγηση αλγορίθμων επεξεργασίας.



**Σχήμα 8.3** Αριστερά: Πείραμα ευθύγραμμης κίνησης μεγάλης ακρίβειας. Δεξιά: Πείραμα ευθύγραμμης κίνησης με ακίνητο σώμα και κινούμενο επίπεδο.

Η πειραματική διάταξη που τελικά χρησιμοποιήθηκε, ως απλή και εύκολη στην υλοποίηση, παρουσιάζεται στο Σχήμα 8.4. Η ευθύγραμμη κίνηση επιτυγχάνεται με δύο μπάρες-οδηγούς που τοποθετούνται στο επίπεδο κίνησης. Άλλες δύο μπάρες τοποθετούνται κάθετα έτσι, ώστε να σηματοδοτούν την αρχή και το τέλος μιας διαδρομής συγκεκριμένου μήκους. Η κίνηση γίνεται χειροκίνητα και το ρομπότ κινείται πάνω στα ροδάκια που προαναφέρθηκαν. Έγιναν πειράματα με διάφορες ταχύτητες και μήκη διαδρομών. Τα συμπεράσματα που βγήκαν σχετικά με τον συντελεστή μετατροπής των [counts] σε [mm] παρουσιάζονται παρακάτω.



Σχήμα 8.4 Πείραμα ευθύγραμμης κίνησης χρησιμοποιώντας μπάρες-οδηγούς.

### 8.2.2 Δεδομένα Πειραμάτων Βαθμονόμησης

Έγιναν συνολικά 8 πειράματα με την πειραματική διάταξη του Σχήματος 8.4 για ευθύγραμμη κίνηση μήκους 100 [mm] και 200 [mm] και τα αποτελέσματα παρουσιάζονται στους παρακάτω πίνακες. Παρατηρείται ότι είναι δύσκολο να μην υπάρχουν μετατοπίσεις στη κάθετη διεύθυνση ως προς την κίνηση. Οι τιμές βέβαια στις στήλες των μετατοπίσεων κατά  $y$  είναι αμελητέες σε σχέση με τις τιμές των στηλών  $x$ . Ένα αξιόπιστο συμπέρασμα, προϋποθέτει την εκτέλεση αρκετών πειραμάτων και την στατιστική ανάλυση των αποτελεσμάτων τους (βλ. Πίνακα 8.1).

100 [mm] along the y axis						
# of exper.	xMouse1	yMouse1	xMouse2	yMouse2	xMouse3	yMouse3
1	5	4011	17	3874	-91	4134
2	-30	3996	90	3956	-53	3980
3	11	3891	112	4006	-10	4043
4	66	3798	29	3987	-19	4095
Average	13	<b>3924</b>	62	<b>3955,75</b>	-43,25	<b>4063</b>

200 [mm] along the y axis						
# of exper.	xMouse1	yMouse1	xMouse2	yMouse2	xMouse3	yMouse3
1	40	7981	83	7897	-8	8019
2	-3	8134	7	7952	-47	8183
3	19	7899	11	8001	16	7953
4	112	8070	29	8094	-23	8085
Average	42	<b>8021</b>	32,5	<b>7986</b>	-15,5	<b>8060</b>

Πίνακας 8.1 Αποτελέσματα πειραμάτων ευθύγραμμης κίνησης για διαδρομές 100 [mm] , 200 [mm].

Από τα πρώτα πειράματα προκύπτει ότι για 100 [mm] έχουμε κατά μέσο όρο περίπου 4000 [counts], ενώ για 200 [mm] παίρνουμε από τους αισθητήρες περίπου 8000 [counts]. Δεδομένου του σφάλματος που προκύπτει από την τοποθέτηση των αισθητήρων, και αυτού που προκύπτει από την ακρίβεια της διαδρομής, η τιμή του συντελεστή μετατροπής [counts] σε [mm] είναι:

$$counts2mm = \frac{100}{4000} = \frac{200}{8000} = 0.025$$



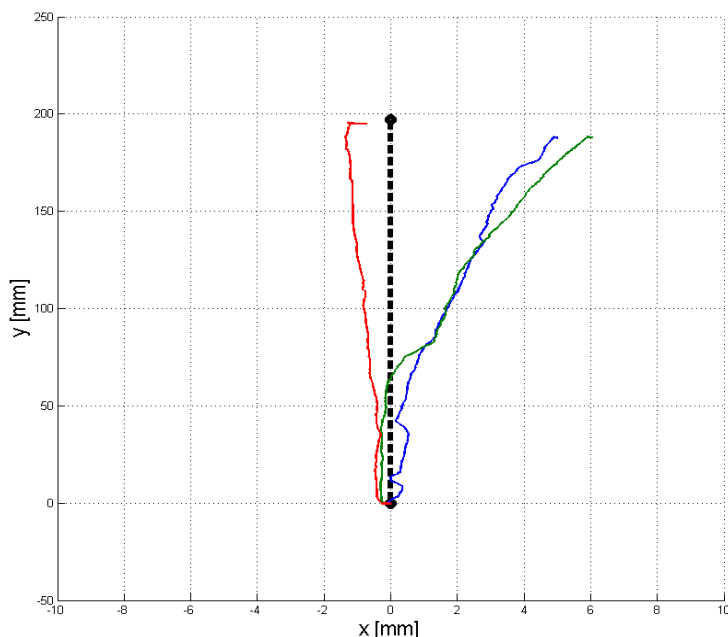
Η τιμή αυτή λοιπόν χρησιμοποιείται κατάλληλα στα προγράμματα που την απαιτούν, και μόνο έτσι μπορούν να αξιολογηθούν τα πειράματα που ακολουθούν.

### 8.3 Πειράματα προσδιορισμού θέσης/ προσανατολισμού από οπτικούς αισθητήρες

#### 8.3.1 Πειράματα ευθύγραμμης τροχιάς

Έγιναν πειράματα ευθύγραμμης κίνησης 200 [mm], δύο ταχυτήτων (3 [sec] και 1 [sec] τα 200 [mm]). Στην πραγματικότητα η πειραματική διαδρομή ήταν 197 [mm], γι αυτό και τα παρακάτω διαγράμματα δεν φθάνουν ποτέ την τιμή 200. Χρησιμοποιήθηκε παρόμοια διάταξη με αυτή του Σχήματος 8.4.

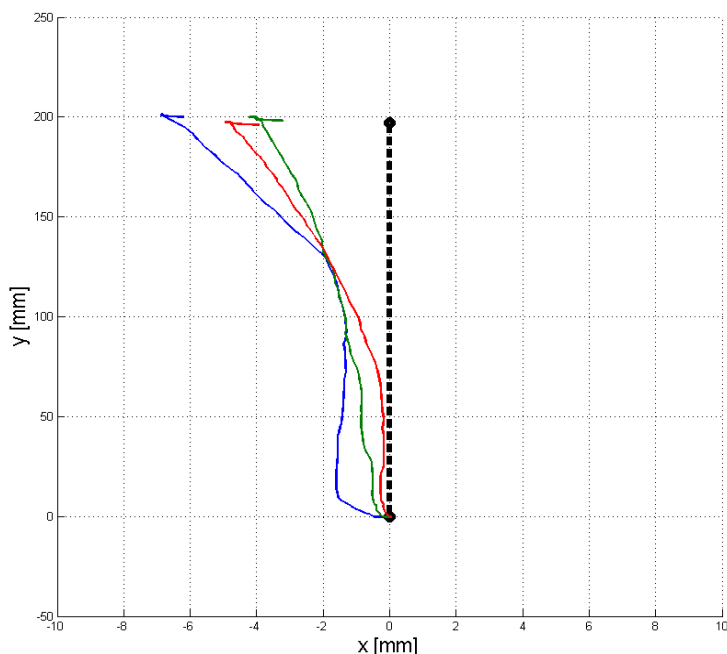
Στο Σχήμα 8.5 σχεδιάζεται η εκτιμώμενη πορεία που προκύπτει με εφαρμογή μόνο του αλγορίθμου μέσου όρου (ο απλούστερος αλγόριθμος). Οι άλλοι αλγόριθμοι δεν χρησιμοποιήθηκαν στο πρώτο αυτό πείραμα γιατί, όπως εξηγείται παρακάτω, η μεγάλη ανακρίβεια του, κάνει αδύνατη τη σύγκριση αλγορίθμων επεξεργασίας. Η πραγματική διαδρομή που έγινε από το ρομπότ δεν είναι γνωστή με αποδεκτή ακρίβεια (ακρίβεια εκατοστού [cm]), οπότε δεν μπορεί να βρεθεί και ο αλγόριθμος που την εκτίμησε πιο πιστά.



**Σχήμα 8.5** Τρία πειράματα ευθείας 197 [mm], μικρής ταχύτητας : 197 [mm] σε 3 [sec].

Όπως αναφέρθηκε και πριν, παρόλο που τα αποτελέσματα του αλγορίθμου μέσου όρου (Σχήμα 8.5, 8.6) φαίνονται λογικά (μετατόπιση μικρότερη του εκατοστού [cm] κατά  $y$ ), δύσκολα μπορούν να βγουν συμπεράσματα για την ακρίβεια και την διαφορετικότητα του σε σχέση με τους υπόλοιπους αλγόριθμους. Πρώτον, η κίνηση δεν περιλαμβάνει καμία περιστροφή (θεωρητικά), που είναι ίσως η πιο βασική παράμετρος της κίνησης και δεύτερον, η κίνηση που πραγματικά εκτελείται από το ρομπότ δεν είναι γνωστή με ακρίβεια. Σαν συνέπεια, τα πειραματικά αποτελέσματα των αλγορίθμων δεν μπορούν να συγκριθούν με κάποια πραγματικά δεδομένα κίνησης. Δεν υπάρχει δηλαδή τρόπος να συγκριθεί η εκτιμώμενη κίνηση με αυτή που πραγματικά εκτελέστηκε. Από τα παρακάτω διαγράμματα λοιπόν μπορούν να εξαχθούν μόνο 'χοντρικά' συμπεράσματα για τους αλγόριθμους εντοπισμού. Τελικά, παρουσιάζεται λοιπόν η ανάγκη εκτέλεσης διαφορετικού είδους πειραμάτων

στα οποία η κίνηση θα είναι καμπυλόγραμμη με περιστροφές και η οποία θα είναι ταυτόχρονα γνωστή με αρκετά καλή ακρίβεια, με στόχο την τελική σύγκριση των αποτελεσμάτων.



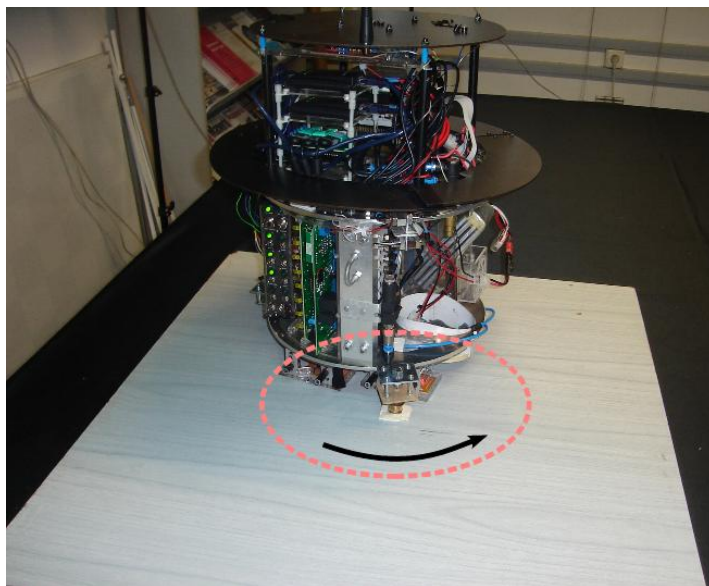
**Σχήμα 8.6** Τρία πειράματα ευθείας 197 [mm], μεγάλης ταχύτητας : 197 [mm] σε 1 [sec].

### 8.3.2 Πειράματα κυκλικής τροχιάς

Η ανάγκη για ένα πείραμα καμπυλόγραμμης τροχιάς της οποίας το σχήμα θα είναι γνωστό με ακρίβεια λίγων χιλιοστών, οδήγησε στην εξής σκέψη. Αν στη παρούσα μορφή του ρομπότ (με ροδάκια αντί για αεροέδρανα), κρατηθεί με κάποιο τρόπο σταθερό το ένα από τα τρία ροδάκια, τότε ασκώντας δύναμη στο ρομπότ, αυτό κινείται αναγκαστικά σε κυκλική τροχιά (βλ. Σχήμα 8.7).

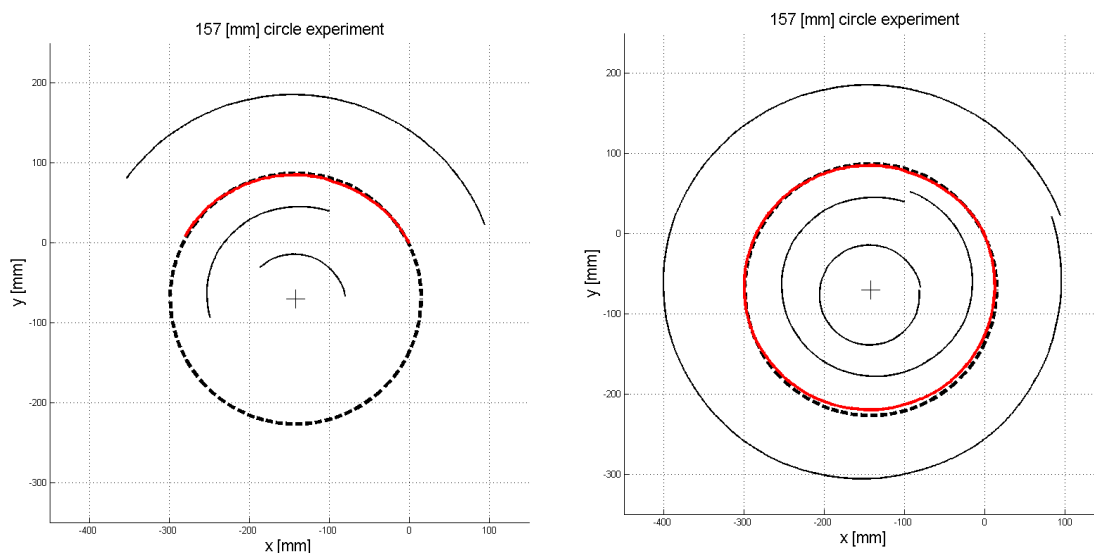
Χρησιμοποιώντας ξανά μια ξύλινη βάση, και περιορίζοντας κατάλληλα το ένα ροδάκι, η πειραματική διάταξη που περιγράφηκε παραπάνω είναι έτοιμη. Στα πειράματα που ακλουθούν, η εκτιμώμενη κίνηση του κέντρου του ρομπότ, θα συγκρίνεται κάθε φορά με την πραγματική. Ως πραγματική τροχιά θεωρείται ένας κύκλος με ακτίνα την απόσταση του κέντρου του ρομπότ από το ροδάκι που περιορίστηκε. Η απόσταση αυτή μετρήθηκε στα 160 [mm].

Έγιναν πειράματα κυκλικής κίνησης δύο ταχυτήτων (2.5 [sec] και 5 [sec] τον κύκλο) μίας περιστροφής καθώς και πειράματα κίνησης 2 και 3 περιστροφών. Οι αλγόριθμοι που χρησιμοποιήθηκαν και αξιολογούνται σε αυτό το κεφάλαιο, είναι αυτοί που αναπτύχθηκαν στο Κεφάλαιο 5, και στο εξής θα αναφέρονται μόνο με τα ονόματά τους όπως ορίστηκαν εκεί.



**Σχήμα 8.7** Πείραμα κυκλικής τροχιάς ακτίνας 160 [mm] σε ξύλινη βάση.

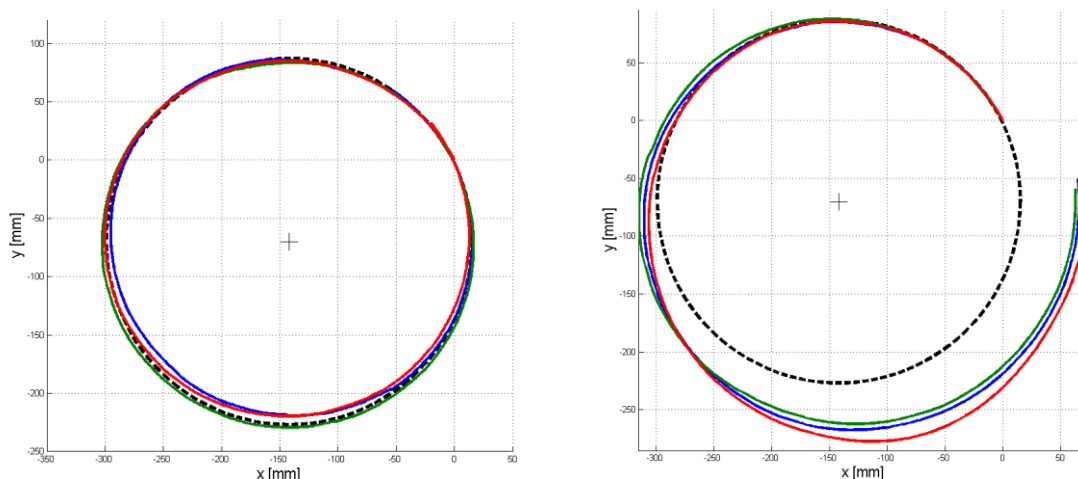
Στο Σχήμα 8.8 φαίνεται πώς εφαρμόζεται ο αλγόριθμος μέσου όρου σε ένα από τα πρώτα πειράματα μικρής ταχύτητας που έγιναν. Φαίνονται μεν τα ατομικά σφάλματα των τριών οπτικών αισθητήρων, αλλά φαίνεται και ότι η τροχιά του κέντρου έχει πολύ μικρό σφάλμα. Αυτό είναι το ζητούμενο και συμβαίνει λόγω του υπολογισμού του μέσου όρου των μετρήσεων. Συμπερασματικά, από αυτό το πείραμα φαίνεται ότι για μικρές ταχύτητες (1 περιστροφή ανά 5 [sec]) έχουμε ακρίβεια χιλιοστού στην εκτίμηση της κίνησης, τουλάχιστον για τα πρώτα δευτερόλεπτα της κίνησης.



**Σχήμα 8.8** Αριστερά: περίπου μισός κύκλος, με κόκκινο το κέντρο (αλγόριθμος μέσου όρου) και διακεκομμένη γραμμή η πραγματική τροχιά. Δεξιά: το ρομπότ στην αρχική του θέση μετά από έναν κύκλο (αλγόριθμος μέσου όρου).

Τονίζεται ότι, χαρακτηριστικό του συστήματος των οπτικών αισθητήρων είναι η συσσώρευση του σφάλματος, οπότε είναι και λογικό να παρουσιάζονται μεγάλα σφάλματα για μεγάλα χρονικά διαστήματα κίνησης. Η προσοχή σε αυτό αλλά και στα πειράματα που ακολουθούν, δίνεται στα πρώτα δευτερόλεπτα της κίνησης. Ενδιαφέρον μεγάλο βέβαια παρουσιάζει και το συνολικό τελικό σφάλμα θέσης και προσανατολισμού που παρουσιάζει ο κάθε αλγόριθμος για το ίδιο πείραμα.





**Σχήμα 8.9** Αριστερά: τροχιά του κέντρου (αλγόριθμος μέσου όρου) από 3 πειράματα χαμηλής ταχύτητας, με διακεκομμένη η πραγματική τροχιά. Δεξιά: τροχιά του κέντρου (αλγόριθμος μέσου όρου) από 3 πειράματα υψηλής ταχύτητας, με διακεκομμένη η πραγματική τροχιά.

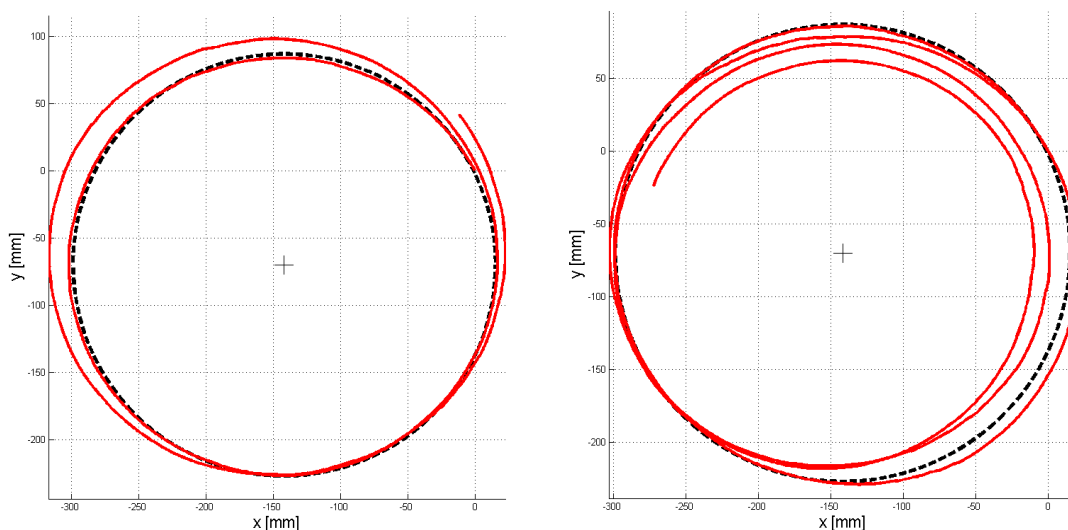
Στο αριστερό γράφημα του Σχήματος 8.9 φαίνονται τα αποτελέσματα τριών ίδιων πειραμάτων χαμηλής πάλι ταχύτητας (επαναλήψεις του ίδιου αρχικού πειράματος), όπως προκύπτουν από τον απλό αλγόριθμο του μέσου όρου. Φαίνεται οριστικά ότι τα σφάλματα σε τέτοιες ταχύτητες είναι της τάξης του ενός χιλιοστού ακόμη και με τον απλό αλγόριθμο επεξεργασίας. Το δε τελικό σφάλμα είναι και αυτό της τάξης του ενός χιλιοστού.

Τα προβλήματα αρχίζουν για παράδειγμα, όταν εκτελεστούν πειράματα μεγαλύτερης ταχύτητας. (1 περιστροφή ανά 2.5 [sec]). Όπως φαίνεται στο δεξί γράφημα του Σχήματος 8.9, ο αλγόριθμος του μέσου όρου εκτιμά σωστά την τροχιά μόνο για τις πρώτες χρονικές στιγμές της κίνησης, ενώ γρήγορα το σφάλμα παίρνει μεγάλες τιμές. Η τελική απόκλιση από την πραγματική θέση είναι κοντά στα 50 [mm].

Παρατηρείται ότι, ο αλγόριθμος αυτός λόγω της απλότητάς του, παρουσιάζει αδυναμία όταν κάποιος αισθητήρας εμφανίσει σημαντικό σφάλμα. Δεν έχει κάποιον καλό τρόπο εκμετάλλευσης της έξτρα πληροφορίας που διαθέτει. Σημειώνεται ξανά ότι, οι πληροφορίες του τρίτου αισθητήρα δεν είναι απαραίτητες για τον υπολογισμό θέσης/γωνίας, και ρόλο έχουν μόνο την βελτίωση της συνολικής εκτίμησης του συστήματος.

Εκτός από τα σφάλματα που είναι πιθανό να εμφανίσουν οι οπτικοί αισθητήρες λόγω μεγάλης ταχύτητας, σημαντικά σφάλματα εμφανίζονται επίσης για μεγάλα χρονικά διαστήματα κίνησης. Στο Σχήμα 8.10 παρουσιάζονται τα αποτελέσματα του απλού πάλι αλγόριθμου σε πειράματα δύο και τριών περιστροφών με μικρή ταχύτητα. Παρατηρούμε ένα σημαντικό τελικό σφάλμα της τάξης των 30 [mm], το οποίο εμφανίζεται λόγω συσσώρευσης σφάλματος επί μεγάλο χρονικό διάστημα.

Σε αυτό το σημείο, έχουν φανεί τα μεγάλα σφάλματα που παρουσιάζει ο αλγόριθμος του μέσου όρου σε κάποιες περιπτώσεις. Με στόχο την αναζήτηση τη βέλτιστη εκμετάλλευση της επιπρόσθετης πληροφορίας του τρίτου αισθητήρα, στο Κεφάλαιο 5, σχεδιάστηκαν αλγόριθμοι μεγαλύτερης πολυπλοκότητας που θεωρητικά βελτιώνουν την ακρίβεια της εκτίμησης. Οι αλγόριθμοι αυτοί εφαρμόζονται εδώ στα ίδια πειράματα στα οποία ο απλός αλγόριθμος παρουσιάζει κακά αποτελέσματα. Παρατηρούνται συγκεκριμένα η συμπεριφορά τους στα σημεία στα οποία το συνολικό σφάλμα αρχίζει να αυξάνει επικίνδυνα, αλλά και η απόκλιση από την πραγματική τελική θέση/γωνία.



**Σχήμα 8.10** Αριστερά: τροχιά του κέντρου (αλγόριθμος μέσου όρου) σε πείραμα κυκλικής τροχιάς δύο περιστροφών. Δεξιά: τροχιά του κέντρου (αλγόριθμος μέσου όρου) σε πείραμα κυκλικής τροχιάς τριών περιστροφών.

### 8.3.3 Αλγόριθμοι με μεταβλητά βάρη σε πειράματα κυκλικής τροχιάς

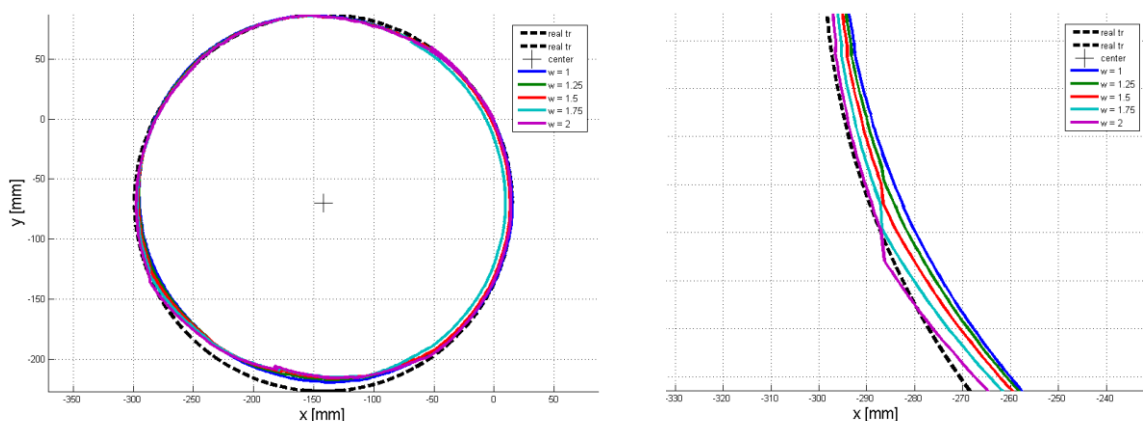
Στα ίδια λοιπόν πειράματα εφαρμόζονται εκ των υστέρων οι αλγόριθμοι ακριβέστερης απόστασης και ενδιάμεσης γωνίας, που έχουν εξηγηθεί στο Κεφάλαιο 5. Στην ενότητα αυτή, επιχειρείται συγκεκριμένα η διερεύνηση της επιρροής των βαρών, που χρησιμοποιούν αυτοί οι αλγόριθμοι, στο τελικό αποτέλεσμα, καθώς και η εύρεση του βέλτιστου ανά περίπτωση βάρους. Υπενθυμίζεται ότι, το βάρος αυτό αντιστοιχεί στην αυξημένη επιρροή των δεδομένων των δύο πιο αξιόπιστων αισθητήρων στο αποτέλεσμα.

Για αυτό το σκοπό, οι αλγόριθμοι εφαρμόζονται σε πειράματα που ο απλός αλγόριθμος του μέσου όρου δεν δίνει ικανοποιητικά αποτελέσματα και εξετάζεται η πιθανή βελτίωσή τους.

#### Αλγόριθμος ακριβέστερης απόστασης

Αρχικά ο αλγόριθμος εφαρμόζεται σε πείραμα μικρής ταχύτητας μιας περιστροφής. Σε αυτό ο απλός αλγόριθμος έδωσε πολύ καλά αποτελέσματα, αλλά σε κάποια χρονική στιγμή αναπόφευκτα παρουσίασε αποκόλληση της εκτιμώμενης από την πραγματική τροχιά. Εδώ εξετάζεται το κατά πόσο ο αλγόριθμος ακριβέστερης απόστασης μπορεί να παρουσιάσει μικρότερο σφάλμα στο σημείο της αποκόλλησης αυτής και να δώσει καλή ακρίβεια για περισσότερο χρόνο.

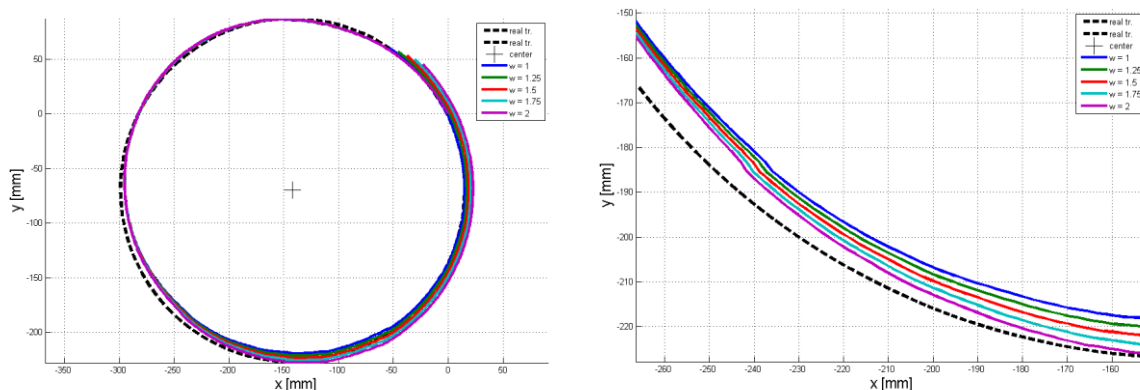
Στο Σχήμα 8.11 φαίνεται ότι πράγματι πετυχαίνει τη βελτίωση της ακρίβειας της εκτίμησης στο σημείο αυτό. Σημειώνεται ότι η μπλε γραμμή ( $w=1$ ) είναι το αποτέλεσμα του απλού αλγορίθμου μέσου όρου. Ένα βάρος κοντά στο  $w=2$  (μωβ γραμμή) εξαλείφει σε μεγάλο βαθμό το σφάλμα στο σημείο όπου αρχίζει να αυξάνει απότομα με τον πρώτο αλγόριθμο. Συμπερασματικά, σε σύγκριση με τον απλό αλγόριθμο, αυτός κρατά την εκτιμώμενη τροχιά κοντά στην πραγματική για περισσότερο χρόνο.



**Σχήμα 8.11** Αριστερά: τροχιά του κέντρου (αλγόριθμος ακριβέστερης απόστασης) σε πείραμα κυκλικής κίνησης χαμηλής ταχύτητας με μεταβλητό βάρος  $w$ . Δεξιά: Μεγέθυνση του κρίσιμου σημείου όπου το σφάλμα του απλού αλγορίθμου αρχίζει να αυξάνεται.

### Αλγόριθμος ενδιάμεσης γωνίας

Ομοίως, στα δεδομένα του ίδιου πειράματος, εφαρμόζεται ο αλγόριθμος ενδιάμεσης γωνίας. Στο Σχήμα 8.12 φαίνεται ότι ένα βάρος κοντά στο 2 εξαλείφει σε μεγάλο βαθμό (μωβ γραμμή) το σφάλμα σε σημείο όμως αρκετά μετά το σημείο όπου ο απλός αλγόριθμος ( $w=1$ ) παρουσιάζει πρώτη φορά αύξηση σφάλματος. Σε αυτή την περίπτωση δηλαδή, ο συγκεκριμένος αλγόριθμος ‘αργεί’ να αντιληφθεί τα μεγάλα σφάλματα, σε αντίθεση με τον αλγόριθμο ακριβέστερης απόστασης, ο οποίος, όπως αναφέρθηκε προηγουμένως, δρα άμεσα στο ίδιο πείραμα.

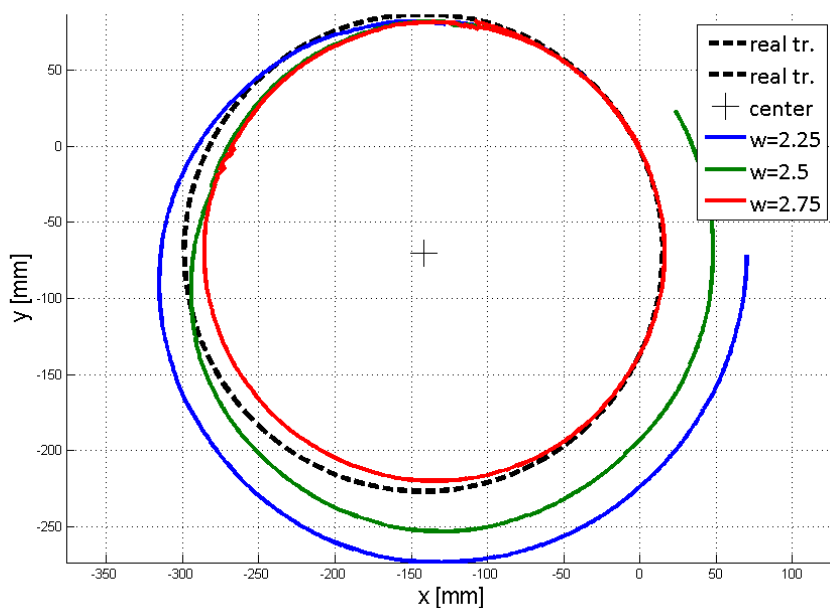


**Σχήμα 8.12** Αριστερά: τροχιά του κέντρου (αλγόριθμος ενδιάμεσης γωνίας) σε πείραμα κυκλικής κίνησης χαμηλής ταχύτητας με μεταβλητό βάρος  $w$ . Δεξιά: Μεγέθυνση του σημείου όπου το σφάλμα του απλού αλγορίθμου συνεχώς αυξάνεται.

### 8.3.4 Αλγόριθμοι με ανακατασκευή σε πειράματα κυκλικής τροχιάς

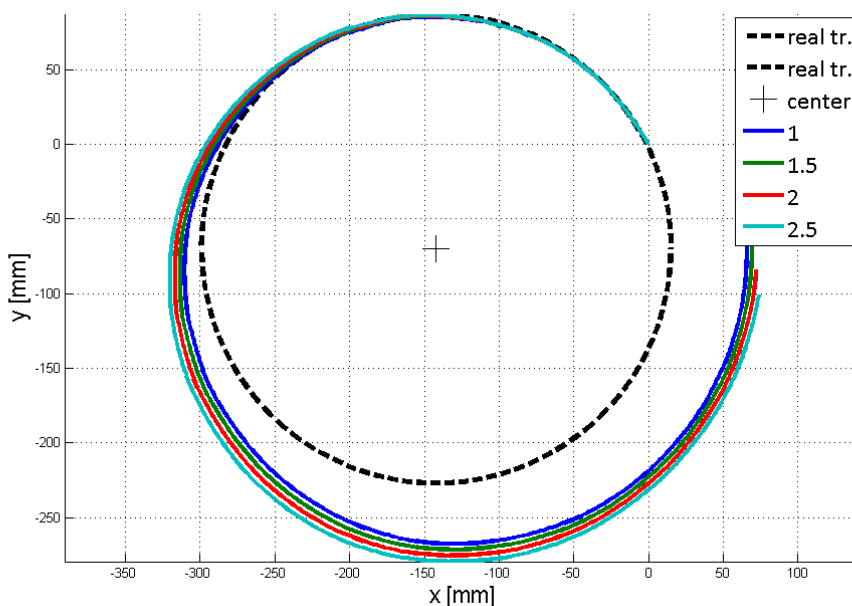
Όταν έχουμε για κάποιο λόγο πολύ μεγάλα σφάλματα (πχ. κάποιος αισθητήρας δεν λειτουργήσει, λάθος λόγω μεγάλης επιτάχυνσης/ ταχύτητας κ.α.), οι παραπάνω ‘έξυπνοι’ αλγόριθμοι δεν μπορούν να δώσουν πολύ καλύτερα αποτελέσματα από τον αλγόριθμο μέσου όρου. Αυτό συμβαίνει γιατί, στην ουσία, είναι κι αυτοί αλγόριθμοι μέσου όρου (με διαφορετικό βέβαια βάρος ανά όρο) και επομένως, κάθε φορά όλοι οι αισθητήρες επιδρούν στο αποτέλεσμα. Στη περίπτωση όπου η μέτρηση ενός είναι τελείως λανθασμένη, η συνολική τροχιά επηρεάζεται σε μεγάλο βαθμό και αμετάκλητα. Με λίγα λόγια, στους προηγούμενους αλγορίθμους δεν υπάρχει η δυνατότητα (εκτός

αν μιλάμε για βάρος αλγορίθμου κοντά στο  $w=3$ , όπου οι δύο αναξιόπιστοι αισθητήρες έχουν μηδενικό ρόλο), διόρθωσης μεγάλων ατομικών σφαλμάτων των αισθητήρων.



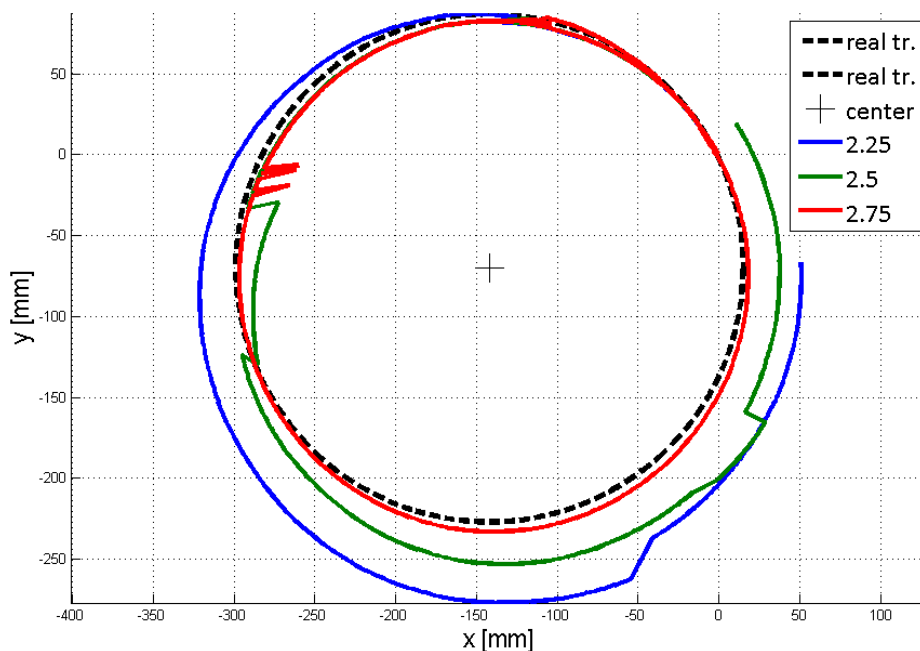
**Σχήμα 8.13** Τροχιά του κέντρου (αλγόριθμος ακριβέστερης απόστασης) σε πείραμα κυκλικής κίνησης υψηλής ταχύτητας με μεταβλητό βάρος  $w$  χωρίς ανακατασκευή.

Στο Σχήμα 8.13, φαίνεται ότι για μεγάλη βαρύτητα στους αξιόπιστους αισθητήρες ( $w=2.75$ ), ο αλγόριθμος ακριβέστερης απόστασης παρουσιάζει κάποια βελτίωση στην εξέλιξη του σφάλματος (κόκκινη γραμμή), αλλά σε καμία περίπτωση δεν είναι ικανοποιητική, αφού μεγάλο σφάλμα εμφανίζεται από πολύ νωρίς.



**Σχήμα 8.14** Τροχιά του κέντρου (αλγόριθμος ενδιάμεσης γωνίας) σε πείραμα κυκλικής κίνησης υψηλής ταχύτητας με μεταβλητό βάρος  $w$  χωρίς ανακατασκευή.

Στο Σχήμα 8.14 φαίνεται ένα παράδειγμα στο οποίο παραδόξως, ο αλγόριθμος ενδιάμεσης γωνίας, παρουσιάζει αύξηση του σφάλματος καθώς αυξάνει η βαρύτητα των δεδομένων των πιο αξιόπιστων αισθητήρων. Επίσης, δίνει χειρότερα αποτελέσματα όσο αυξάνεται το βάρος (ή εμπιστοσύνη) των αισθητήρων που δίνουν ενδιάμεσα σε μέγεθος αποτελέσματα. Αυτό συμβαίνει γιατί σε ακραίες συνθήκες όπως αυτή, η ενδιάμεση γωνία μπορεί να είναι πιο εσφαλμένη από κάποια ακραία. Τώρα για παράδειγμα, ο πιο αξιόπιστος αισθητήρας είναι ο πιο κοντινός στο κέντρο της κυκλικής τροχιάς γιατί κινείται με μικρότερη ταχύτητα, αλλά με αυτόν τον αλγόριθμο μπορεί να θεωρηθεί αναξιόπιστος. Εδώ λοιπόν, φαίνεται καθαρά ότι η χρήση εξυπνότερων θεωρητικά αλγορίθμων δεν είναι πάντα μονόδρομος.



Σχήμα 8.15 Τροχιά του κέντρου (αλγόριθμος ακριβέστερης απόστασης) σε πείραμα κυκλικής κίνησης υψηλής ταχύτητας με μεταβλητό βάρος  $w$  και ανακατασκευή.

Στο σημείο αυτό, όπου έχουν φανεί τα ελαττώματα ακόμη και των θεωρητικά εξυπνότερων του απλού αλγορίθμων, εφαρμόζονται στα ίδια πειραματικά δεδομένα οι αλγόριθμοι ανακατασκευής για να διερευνηθεί η ακρίβειά και των δικών τους εκτιμήσεων.

Στο Σχήμα 8.15 παρουσιάζονται τα αποτελέσματα του αλγορίθμου ακριβέστερης απόστασης με ανακατασκευή, επί των πειραματικών δεδομένων κυκλικής κίνησης με μεγάλη ταχύτητα. Όπως φαίνεται, είναι ο πρώτος αλγόριθμος που παρουσιάζει σχεδόν ολική εξάλειψη σφάλματος (κόκκινη γραμμή, για  $w=2.75$  δηλαδή για μεγάλη βαρύτητα στους πιο αξιόπιστους αισθητήρες). Εντούτοις, παρατηρείται ότι εμφανίζονται ανωμαλίες στην τροχιά (λογικό λόγω της ανακατασκευής) που είναι πιθανό να δημιουργήσουν προβλήματα στον υπολογισμό ταχύτητας ή επιτάχυνσης.

Ο αλγόριθμος ενδιάμεσης γωνίας με ανακατασκευή στο κυκλικό πείραμα υψηλής ταχύτητας δεν έδωσε καλύτερα αποτελέσματα από τον απλό αλγόριθμο και δεν παρουσιάζεται εδώ. Ίσως αξίζει να δοκιμαστεί σε πείραμα διαφορετικής τροχιάς.

Διεξάχθηκαν πολλά τέτοια πειράματα και έδειξαν ότι σε πολλές περιπτώσεις όπως αυτή, με έναν τέτοιο αλγόριθμο τα μεγάλα σφάλματα όντως εξαλείφονται. Φαίνεται λοιπόν, ότι σε κινήσεις με γρήγορες και απότομες μεταβολές συμφέρει η χρήση του αλγορίθμου ακριβέστερης απόστασης με ανακατασκευή και πολύ μεγάλο βάρος ( $w_B = [2.5-3]$ ), έτσι ώστε να δίνεται εμπιστοσύνη μόνο στους πιο αξιόπιστους αισθητήρες, αφού αναμένεται ο τρίτος να έχει αρκετά μεγάλο σφάλμα. Ίσως βέβαια αυτό να μην λειτουργήσει σε όλες τις περιπτώσεις, πχ όταν θεωρηθεί αξιόπιστος ένας αισθητήρας ενώ δεν είναι.

Δεδομένα από αρκετά πειράματα είναι αποθηκευμένα σε κατάλληλα αρχεία .txt και παραδίδονται στο CD αυτής της εργασίας. Σε αυτά τα αρχεία, υπάρχουν τα ακατέργαστα δεδομένα των πειραμάτων, και επομένως μπορούν να χρησιμοποιηθούν για αξιολόγηση αλγορίθμων που θα αναπτυχθούν στο μέλλον.

#### 8.4 Πειράματα προσδιορισμού θέσης/ προσανατολισμού από κάμερα

Με στόχο την αξιολόγηση του συστήματος της κάμερας και ειδικά του αλγορίθμου επεξεργασίας έτσι όπως τροποποιήθηκε σε προηγούμενο κεφάλαιο, χρησιμοποιείται η ίδια πειραματική διάταξη που χρησιμοποιήθηκε στα πειράματα των οπτικών αισθητήρων. Αρχικά γίνονται πειράματα ευθύγραμμης τροχιάς και ύστερα κυκλικής και παρουσιάζονται τα αποτελέσματα μετά την επεξεργασία.

##### 8.4.1 Πειράματα ευθύγραμμης τροχιάς

Αρχικά, χρησιμοποιείται η ξύλινη βάση με τις ευθύγραμμες μπάρες για εκτέλεση πειραμάτων ευθύγραμμης κίνησης 200 [mm]. Η πειραματική διάταξη μεταφέρεται σε διάφορα σημεία σε όλη την έκταση της τράπεζας του γρανίτη (βλ. Σχήμα 8.16), έτσι ώστε να γίνουν μετρήσεις σε αρκετά σημεία του χώρου εργασίας. Έγιναν συνολικά 5 πειράματα. Η βάση τοποθετήθηκε στις 4 γωνίες και στο κέντρο της τράπεζας.



**Σχήμα 8.16** Πειράματα ευθύγραμμης κίνησης 200 [mm] σε διάφορα σημεία κάτω από την κάμερα.

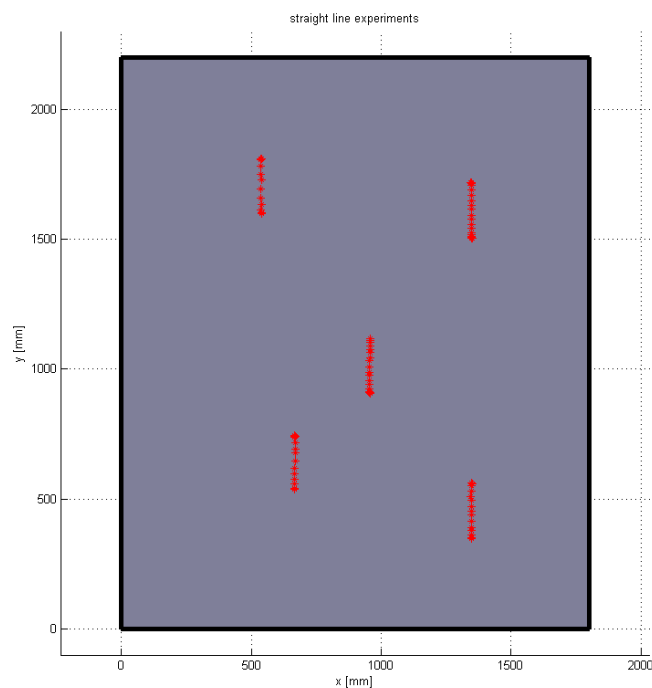
Τα αποτελέσματα των πειραμάτων αποθηκεύονται κατευθείαν μέσω του μοντέλου Simulink που παρουσιάστηκε προηγουμένως και τελικά παρουσιάζονται συνολικά σε ένα γράφημα (Σχήμα 8.17). Καταρχήν φαίνεται ότι ο αλγόριθμος αφαίρεσης της παραμόρφωσης του φακού, [12] είναι αποτελεσματικός σε ικανοποιητικό βαθμό, αφού οι τροχιές που ανιχνεύονται προσεγγίζουν τις πραγματικές ευθείες σε επίπεδο μερικών χιλιοστών.

Στη συνέχεια για να βγει ακριβές συμπέρασμα για την αποτελεσματικότητα και την ακρίβεια του αλγορίθμου επεξεργασίας, συγκρίνεται το μήκος των τροχιών που υπολογίστηκαν, με το πραγματικό. Υπολογίζοντας τα μήκη των γραμμών στα πέντε σημεία της τράπεζας του γρανίτη, προκύπτουν τα εξής:

- Πείραμα κάτω αριστερά: μήκος τροχιάς  $s = 206.68$  [mm]
- Πείραμα κάτω δεξιά: μήκος τροχιάς  $s = 214.17$  [mm]
- Πείραμα πάνω αριστερά: μήκος τροχιάς  $s = 211.30$  [mm]
- Πείραμα πάνω δεξιά: μήκος τροχιάς  $s = 212.90$  [mm]
- Πείραμα στο κέντρο: μήκος τροχιάς  $s = 211.150$  [mm]

Παρατηρείται ότι είναι περίπου κατά 10 [mm] μεγαλύτερες οι αποστάσεις από αυτές που διανύθηκαν στην πραγματικότητα. Αυτό είναι λογικό και οφείλεται στο επιπρόσθετο ύψος των 20 [mm] που προσδίδει η ξύλινη βάση που χρησιμοποιείται για το συγκεκριμένο πείραμα. Αυτό σημαίνει ότι υπό κανονικές συνθήκες λειτουργίας το ρομπότ θα βρίσκεται 20[mm] χαμηλότερα, και τότε το σύστημα της κάμερας θα παρουσιάζει ακόμη πιο ακριβή αποτελέσματα. Σημαντικό σε αυτή τη φάση είναι ότι οι τροχιές αναγνωρίζονται πράγματι σαν ευθείες από την κάμερα.

Στο μέλλον, μπορούν να γίνουν πειράματα με την ίδια διάταξη αλλά και προς άλλες κατευθύνσεις (όχι μόνο κάθετα) για εξαγωγή επιπλέον χρήσιμων συμπερασμάτων. Τελικά, τα πειράματα που είναι επιτακτικό να εκτελεστούν μελλοντικά, είναι αυτά με το ρομπότ να κινείται στο προδιαγεγραμμένο ύψος.



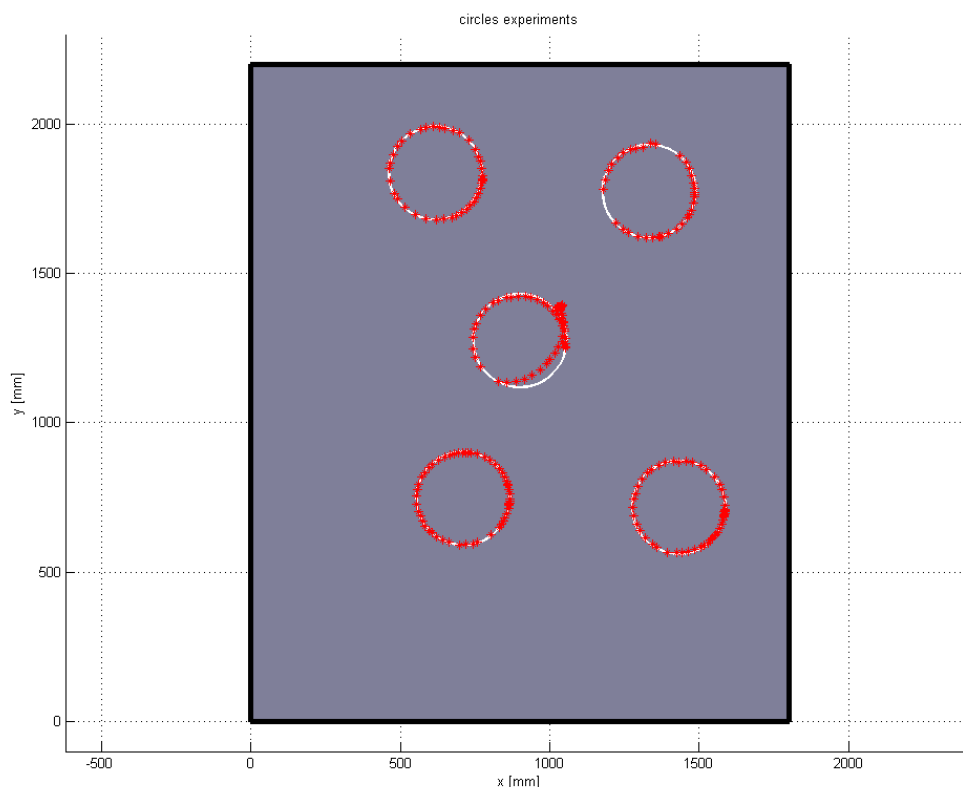
**Σχήμα 8.17** Αποτελέσματα ευθύγραμμων κινήσεων σε διάφορες θέσεις πάνω στον χώρο εργασίας μετά από επεξεργασία του συστήματος της κάμερας

#### 8.4.2 Πειράματα κυκλικής τροχιάς

Η επόμενη σειρά πειραμάτων, αφορά κινήσεις κυκλικής τροχιάς. Η πειραματική διάταξη που χρησιμοποιήθηκε και στα πειράματα των οπτικών αισθητήρων, τοποθετείται εδώ σε 5 σημεία πάνω στην τράπεζα του γρανίτη έτσι, ώστε να καλυφθεί το σύνολο του χώρου εργασίας. Γίνονται λοιπόν 5 πειράματα (4 γωνίες και κέντρο τράπεζας, βλ. Σχήμα 8.18), με ταχύτητα η οποία φαίνεται για κάθε πείραμα στο Σχήμα 8.19.

Οι κύκλοι που διαγράφονται σύμφωνα με τον αλγόριθμο εντοπισμού του συστήματος της κάμερας έχουν ακτίνα 157-160 [mm]. Όπως αναφέρθηκε και στα πειράματα της ευθύγραμμης κίνησης, πρέπει να ληφθεί υπόψη και το συστηματικό σφάλμα που εισάγεται στο πείραμα λόγω της ξύλινης βάσης που χρησιμοποιείται (το πείραμα γίνεται σε ύψος μεγαλύτερο κατά 20[mm]). Επομένως, τα αποτελέσματα είναι ικανοποιητικά, αφού η ακτίνα της πραγματικής κυκλικής τροχιάς έχει μετρηθεί στα 160 [mm]. Το πείραμα βέβαια δεν διεξάγεται με ακρίβεια [mm], γεγονός που δικαιολογεί και τις διακυμάνσεις των αποτελεσμάτων. Το σημαντικό εδώ είναι ότι, η μορφή των αποτελεσμάτων είναι με μεγάλη ακρίβεια (χιλιοστού) κυκλική, γεγονός που δείχνει την ορθότητα και την ακρίβεια του προγράμματος επεξεργασίας.



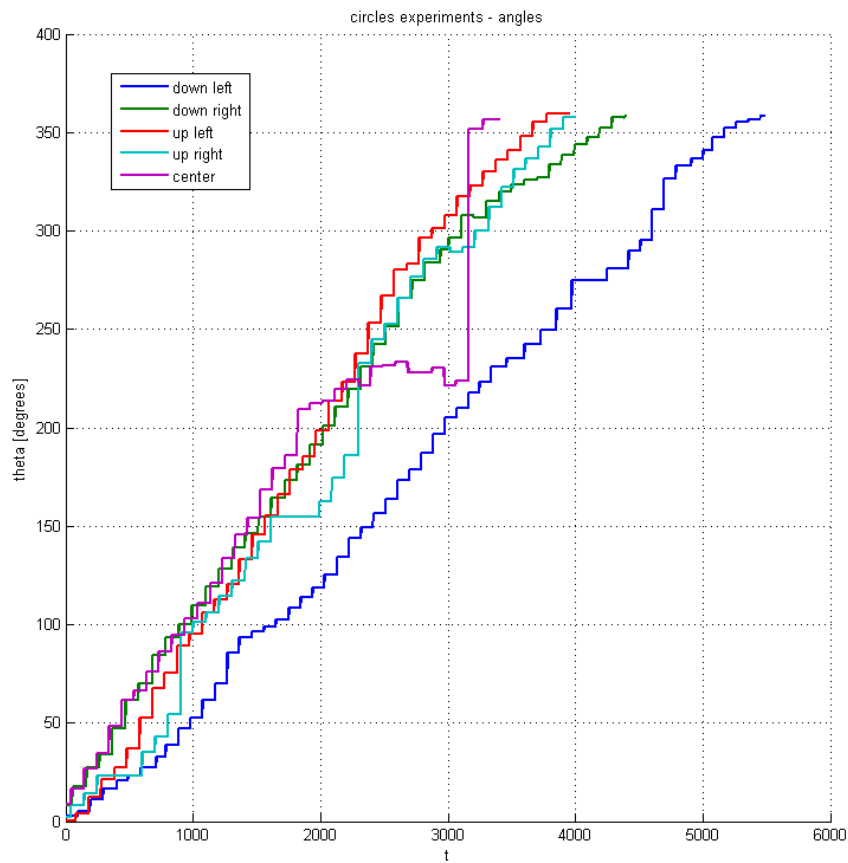


**Σχήμα 8.18** Αποτελέσματα κυκλικών κινήσεων σε διάφορες θέσεις πάνω στον χώρο εργασίας μετά από επεξεργασία του συστήματος της κάμερας.

Στο πείραμα που έγινε στο κέντρο της τράπεζας, τα αποτελέσματα αποκλίνουν αρκετά από την κίνηση που όντως έγινε. Παρατηρείται ότι σε κάποια σημεία της κυκλικής τροχιάς, το πρόγραμμα επεξεργασίας δεν ανιχνεύει σωστά την κίνηση που εκτελείται. Όπως διαπιστώθηκε, με δοκιμές εκ των υστέρων, τα σημεία που βρίσκονται ακριβώς κάτω από την κάμερα, παρουσιάζουν κάποιες ιδιαιτερότητες, που δημιουργούν προβλήματα στον αλγόριθμο επεξεργασίας. Αυτά οφείλονται κυρίως στον ευρυγώνιο φακό, ο οποίος ‘βλέπει’ τα αντικείμενα που βρίσκονται ακριβώς από κάτω του μεγεθυμένα σε σχέση αυτά που υπάρχουν στον υπόλοιπο χώρο. Αυτό για τον αλγόριθμο σημαίνει ότι ένα μικρής φωτεινότητας LED μπορεί υπό αυτές τις συνθήκες να θεωρηθεί μεγάλης φωτεινότητας και αντίστροφα. Ένα άλλο πρόβλημα που παρουσιάζεται στα σημεία κάτω από την κάμερα, είναι η εμφάνιση κάποιων LED ‘φαντασμάτων’, τα οποία δεν υπάρχουν αλλά δημιουργούνται λόγω διαφόρων ανακλάσεων. Αυτό σημαίνει για τον αλγόριθμο, ότι υπάρχουν περισσότερα από τα αναμενόμενα LEDs στον χώρο εργασίας, και άρα αυτός καλείται να αντιμετωπίσει ένα τελείως διαφορετικό πρόβλημα.

Τα προβλήματα αυτά μπορούν να αντιμετωπιστούν αλγοριθμικά. Είναι χρήσιμο, στο μέλλον να γίνουν αντίστοιχα πειράματα, και μελέτη των αποτελεσμάτων επεξεργασίας σε ένα μεγάλο εύρος των σημαντικών παραμέτρων επεξεργασίας. Οι κύριες παράμετροι, δεν είναι άλλες από: το threshold (τιμή κατωφλίου), και τα αναμενόμενα απ’ τον αλγόριθμο μεγέθη των LEDs. Μία τεχνική που μπορεί να βοηθήσει στη λύση των προβλημάτων, είναι η τροποποίηση του αλγορίθμου έτσι, ώστε να μεταβάλλει τις παραμέτρους επεξεργασίας ανάλογα με τη θέση του σώματος πάνω στην τράπεζα. Για παράδειγμα, όσο πιο κοντά στο κέντρο ανιχνεύεται κίνηση, τόσο μεγαλύτερη τιμή threshold θα εφαρμόζεται.





**Σχήμα 8.19** Μεταβολή της γωνίας του ρομπότ σε πειράματα κυκλικών κινήσεων σε διάφορες θέσεις πάνω στον χώρο εργασίας, μετά από επεξεργασία του συστήματος της κάμερας. Στον άξονα του χρόνου: 250 μετρήσεις αντιστοιχούν σε 1 δευτερόλεπτο.



# Κεφάλαιο 9

## Συμπεράσματα και μελλοντική εργασία

### 9.1 Συμπεράσματα

Εργασία αυτή ήταν εκτενής και πολυεπίπεδη. Πρωταρχικός στόχος ήταν ο σχεδιασμός και η ανάπτυξη του ηλεκτρικού/ηλεκτρονικού συστήματος του δεύτερου ρομπότ και ο προγραμματισμός του σε ένα λογικό χρονικό πλαίσιο, πράγμα δύσκολο αν λάβουμε υπόψη την πολυπλοκότητα του συστήματος και τις υψηλές προδιαγραφές που έπρεπε να ικανοποιηθούν.

Βασικό χαρακτηριστικό της ανάπτυξης ήταν ο σχεδιασμός βασισμένος στο μοντέλο (model-based design), ο πλέον σύγχρονος τρόπος σχεδιασμού και ανάπτυξης συστημάτων αυτομάτου ελέγχου. Συγκεκριμένα επιλέχτηκε το αρκετά διαδεδομένο πακέτο xPC Target της Mathworks. Διαπιστώθηκαν όλα σχεδόν τα πλεονεκτήματα που υπόσχεται το πακέτο, σε σχέση με τον παραδοσιακό τρόπο ανάπτυξης, αλλά παρατηρήθηκαν και κάποιες δυσκολίες. Ενώ παρείχε μια πλατφόρμα πολύ φιλική στη χρήση, που ενσωμάτωνε όλα τα επίπεδα ανάπτυξης (σχεδιασμός – προγραμματισμός –

υλοποίηση), αντιμετωπίστηκαν μερικά προβλήματα σε κάποια συγκεκριμένα σημεία. Α) Ασύρματη επικοινωνία, Β) Πρόβλημα επικοινωνίας USB, Γ) Δυσκολία στην εύρεση συμβατών hardware.

Τελικά κρίνεται ότι τα πλεονεκτήματα ήταν περισσότερα και σημαντικότερα από τα μειονεκτήματα, κυρίως γιατί δόθηκε η ευκαιρία ανάπτυξης μεγάλου μέρους του συστήματος σε μικρό χρονικό διάστημα. Δοκιμάστηκαν αρκετές καινοτόμες λύσεις και ιδέες. Πολλές από αυτές δεν ήταν βέλτιστες και χρήζουν επανεξέτασης, πράγμα απλό λόγω του μεγάλου βαθμού επεκτασιμότητας που χαρακτηρίζει το πακέτο. Σημαντικό είναι ότι, με χρήση των γνωστών μόνο μοντέλων Simulink, ήταν δυνατός ο προγραμματισμός και ο έλεγχος του ρομπότ σε αυστηρά πραγματικό χρόνο (hard real time). Δόθηκε δηλαδή η δυνατότητα της άμεσης μετάβασης από την προσομοίωση στην υλοποίηση, και η παράκαμψη χαμηλού επιπέδου προγραμματιστικών προβλημάτων. Δεν μπορεί επίσης να μην αναφερθεί η δυνατότητα χρήσης και εύκολης ενσωμάτωσης γνωστών εξωτερικών πακέτων λογισμικού στην εφαρμογή, όπως Solidworks, VRML, Labview κ.α., πράγμα που διευκόλυνε και επιτάχυνε την διαδικασία ανάπτυξης.

Πρέπει τέλος να αναφερθεί ότι, σε καμία περίπτωση, η μέθοδος του σχεδιασμού βασισμένου στο μοντέλο δεν αντικαθιστά την χαμηλού επιπέδου και μη αυτοματοποιημένη ανάπτυξη των σύγχρονων συστημάτων. Χρησιμοποιείται κυρίως στο αρχικό στάδιο του σχεδιασμού και της ανάπτυξης ενσωματωμένων συστημάτων και έτσι μπορεί να μειωθεί δραστικά ο συνολικός χρόνος ανάπτυξης. Στην εκπαίδευση και στην έρευνα ιδιαίτερα, όπου δεν απαιτούνται τελικά προϊόντα, αποτελεί ιδανικό τρόπο ανάπτυξης συστημάτων.

Συνοπτικά, τα γενικά συμπεράσματα που προέκυψαν από την εκπόνηση της εργασίας είναι τα εξής:

- Ο σχεδιασμός του ηλεκτρικού / ηλεκτρονικού υποσυστήματος ενός ρομπότ είναι πολυεπίπεδος και απαιτεί γνώσεις από αρκετά επιστημονικά πεδία.
- Η ενημέρωση σχετικά με την σύγχρονη τεχνολογία, οι καλές βασικές γνώσεις, η εμπειρία και η φαντασία είναι τα εφόδια σε ένα τέτοιο εγχείρημα.
- Η τεχνική model-based design έκανε ευκολότερο το γενικό πρόβλημα και δυσκολότερα κάποια ειδικά προβλήματα, και σίγουρα άξιζε την δοκιμή.
- Το xPC Target, που αποτελεί τον κύριο εκπρόσωπο της μεθόδου, είναι γενικά ένα πολύ χρήσιμο πακέτο ειδικά κατά την διαδικασία του σχεδιασμού σύγχρονων συστημάτων αυτομάτου ελέγχου.

## 9.2 Μελλοντική Εργασία

Στην εργασία αυτή τοποθετήθηκαν τα θεμέλια για την ανάπτυξη του καινούριου ρομπότ του επιπέδου διαστημικού εξομοιωτή. Σχεδιάστηκαν τα κύρια συστήματα, αγοράστηκαν τα βασικά υλικά, αναπτύχθηκε η βάση του λογισμικού και έγιναν αρκετά πειράματα.

Λόγω όμως των υψηλών προδιαγραφών, και των υψηλών στόχων που έχουν τεθεί, υπάρχουν αρκετά πράγματα που πρέπει να γίνουν σαν συμπλήρωση και ολοκλήρωση αυτών που άρχισαν εδώ. Επιπλέον, ο χώρος είναι ανοιχτός για καινούριες ιδέες, τόσο στην περαιτέρω ανάπτυξη του ρομπότ, όσο και στην θεωρητική και πειραματική μελέτη που σχετίζεται με αυτό.

Πιο συγκεκριμένα, σαν συνέχεια της εργασίας αυτής και γενικά της ανάπτυξης του ρομποτικού συστήματος αυτού, προτείνονται τα παρακάτω:

- Χρήση σταθερού desktop υπολογιστή ως host PC (για το xPC Target). Η μέχρι τώρα ανάπτυξη έγινε σε ένα MacBook Pro μέσω Virtual Machine στο οποίο έτρεχε Windows Xp.
- Μεταφορά των μοντέλων / προγραμμάτων που αναπτύχθηκαν στη νέα έκδοση του Matlab (Matlab 8).
- Ενσωμάτωση των επιμέρους μοντέλων Simulink σε ένα ενιαίο.
- Ανάπτυξη γραφικού περιβάλλοντος ελέγχου (GUI) μέσω Matlab, με ενσωματωμένο VRML Viewer για χειρισμό του ρομπότ. Το πακέτο Matlab ενδείκνυται αλλά μπορεί να χρησιμοποιηθεί και το ειδικό για τέτοιες εφαρμογές Labview.
- Χειρισμός ρομπότ με joystick. Ενδείκνυται στο xPC Target.

- Εκτέλεση πειραμάτων ευθείας και κύκλου με αεροέδρανα και όχι με ροδάκια χρησιμοποιώντας νήμα, όπως προτάθηκε. Προκύπτουν τροχιές τέλειων σχημάτων, που θα βοηθήσουν στη βαθμονόμηση αλλά και αλλού.
- Εκτέλεση πειραμάτων με το ακριβές σύστημα καμερών phase space, για βαθμονόμηση.
- Δοκιμή της πλακέτας I/O με PWM στις βαλβίδες, για κίνηση με προωθητήρια (thrusters).
- Δοκιμές με άλλες περιόδους δειγματοληψίας και άλλους επιλύτες στο xPC Target.
- Επανεξέταση του συστήματος των οπτικών αισθητήρων. Χρήση οικονομικού SBC (single board computer) , για παράδειγμα του 'raspberry pi', για διάβασμα αισθητήρων μέσω USB και τελική αποστολή στον κεντρικό υπολογιστή.
- Προσπάθεια για χρήση οπτικών αισθητήρων τεχνολογιών αιχμής Blue Track και Darkfield, σε περίπτωση που το υπάρχον σύστημα παρουσιάσει προβλήματα ακρίβειας.
- Αλλαγή τρόπου αποστολής των δεδομένων της κάμερας, από TCP/IP σε UDP/IP, στον κώδικα επεξεργασίας εικόνας C++, και παράλειψη μοντέλου simulink που τώρα κάνει την μετατροπή.
- Ανάπτυξη αλγορίθμου επεξεργασίας στον οποίο θα μεταβάλλονται οι παράμετροι επεξεργασίας εικόνας ανάλογα με τη θέση του ρομπότ στο τραπέζι (π.χ. μικρότερο threshold στο κέντρο).
- Εκλογή και αγορά ενισχυτές (servoamplifiers) για τους κινητήρες του ρομπότ.
- Σχεδιασμός και κατασκευή των συστημάτων βραχιόνων και σφονδύλου αντίδρασης.
- Σχεδιασμός υποσυστήματος που θα διαβάζει τους encoders των κινητήρων και θα επικοινωνεί κατάλληλα με την κεντρική CPU του ρομπότ.
- Να εξεταστεί η κάρτα 'MPL PATI PC-104'. Υποστηρίζεται από το xPC Target και ίσως είναι ικανή για σύνδεση πολλών encoders. Αξιολόγηση αλγορίθμων επεξεργασίας χρησιμοποιώντας τα ακατέργαστα δεδομένα των πειραμάτων του συστήματος οπτικών αισθητήρων (αρχεία txt στο CD) με διαφορετικούς /καινούριους /παλιούς αλγορίθμους επεξεργασίας για εξαγωγή συμπερασμάτων για αυτούς
- Αξιοποίηση των ακατέργαστων δεδομένων (υπάρχουν στο CD της εργασίας) των πειραμάτων ευθείας και κύκλου που έγιναν, για αξιολόγηση παλιών και καινούριων αλγορίθμων επεξεργασίας στα συστήματα κάμερας και οπτικών αισθητήρων.



# Βιβλιογραφία και άλλες πηγές

- [1] Φλέσσα, Θ., 'Μοντελοποίηση, Έλεγχος και Προγραμματισμός Ρομπότ Επίπεδου Διαστημικού Εξομοιωτή', Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, 2009.
- [2] Καλιακάτσος, Ι., 'Σχεδιασμός Συστήματος Προώθησης Πεπιεσμένου CO<sub>2</sub> και Έλεγχος Κίνησης Εξομοιωτή Διαστημικού Ρομπότ', Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, 2006.
- [3] Κοντολάτης, Ι., 'Ανάδραση Θέσης για την Οπτική Οδήγηση Εξομοιωτή Διαστημικού Ρομπότ', Μεταπτυχιακή Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, 2008.
- [4] Παρασκευάς, Ι., 'Ανάπτυξη Συστήματος Οπτικών Αισθητήρων για Εντοπισμό Θέσης Ρομπότ σε Επίπεδη Κίνηση', Μεταπτυχιακή Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, 2008.
- [5] Τσιφάκης, Α., 'Σχεδιασμός Συστήματος Προώθησης Πεπιεσμένου CO<sub>2</sub> και Έλεγχος Κίνησης Εξομοιωτή Διαστημικού Ρομπότ', Μεταπτυχιακή Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, 2006.
- [6] Ψαρρός, Δ., 'Ανάλυση, Σχεδίαση και Αξιολόγηση Σφονδύλου Αντίδρασης για Επίπεδο Εξομοιωτή Διαστημικού Ρομπότ', Μεταπτυχιακή Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, 2006.
- [7] 'Ηλεκτρονική Κατασκευή – PCB Design Tutorial', Τ.Ε.Ι. ΠΕΙΡΑΙΑ Τμήμα Η/Υ Συστημάτων, Επιμέλεια: Κ. ΤΑΤΣΗ ,Καθηγητής : Κ. ΒΑΤΑΒΑΛΗΣ.
- [8] Craig, J. J., 'Introduction to Robotics, Mechanics and Control', Addison-Wesley Publishing Company, 2nd edition 1989.
- [9] Παπαδόπουλος, Ε. και Κυριακόπουλος, Κ., 'Συστήματα Ευφυούς Ελέγχου & Ρομποτική', Εκδόσεις ΕΜΠ, Αθήνα 2004.
- [10] 'MultiSIM για Μηχανικούς-Εγχειρίδιο Αναλογικών και Ψηφιακών Κυκλωμάτων, Περιβάλλον Προσομοίωσης και Μετρήσεων με Διασύνδεση LabVIEW', Εκδόσεις Τζιόλα, ISBN: 978-960-418-164-3.
- [11] Παρασκευάς, Ι., 'Σύλληψη στόχων από Διαστημικό Ρομποτικό Σύστημα με χρήση του Κρουστικού Κέντρου', Εθνικό Μετσόβιο Πολυτεχνείο, Διπλωματική Εργασία, 2005.
- [12] Καλγρεάδης, Α., 'Ανάπτυξη Αλγορίθμων Επεξεργασίας Εικόνας και Επεκτεταμένου Φίλτρου Kalman (EKF) για Εντοπισμό Θέσης Ρομπότ Διαστημικού Εξομοιωτή', Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, 2013.
- [13] 'Guidance and Navigation Software Architecture Design for the Autonomous Multi-Agent Physically Interacting Spacecraft (AMPHIS) Test Bed', By Blake D. Eikenberry December 2006, NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA.
- [14] 'Control System of a Three DOF Spacecraft Simulator by Vectorable Thrusters and Control Moment Gyros', By William D. Price December 2006, NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA.
- [15] Ανδρέου, Σ., 'Δυναμική Δορυφόρου - Στρεφόμενου Προσαρτήματος με Μηχανισμό χωρίς Αντιδράσεις και Μελέτη Οφέλους Εξοικονόμησης Προωθητικού Καυσίμου', Μεταπτυχιακή Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, 2012.
- [16] Τεγοπούλου, Α., 'Προσδιορισμός Κίνησης Στερεού Σώματος από Ανακριβείς Μετρήσεις Θέσης Σημείων του', Μεταπτυχιακή Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, 2011.



- [17] Καλίτσης, Ν., 'Σύστημα Οπτικών Αισθητήρων για την Αναγνώριση της Μετατόπισης και της Περιστροφής Σώματος – Υλοποίηση με χρήση διαύλου USB', Μεταπτυχιακή Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, 2011.
- [18] [http://en.wikipedia.org/wiki/Optical\\_flow](http://en.wikipedia.org/wiki/Optical_flow)
- [19] <http://sohilp.blogspot.gr/2011/03/mouse-based-project.html>
- [20] [http://www.mathworks.com/support/product/sys\\_requirements.html?product=XP](http://www.mathworks.com/support/product/sys_requirements.html?product=XP)
- [21] <http://www.mathworks.com/support/compilers/R2012b/win64.html>
- [22] [http://en.wikipedia.org/wiki/Kessler\\_syndrome](http://en.wikipedia.org/wiki/Kessler_syndrome)
- [23] <http://jonathanmissel.com/index.html>
- [24] <http://www.spacesafetymagazine.com/2012/02/15/cleanspace-satellite-clean-space-debris/>
- [25] [http://en.wikipedia.org/wiki/Space\\_rendezvous](http://en.wikipedia.org/wiki/Space_rendezvous)
- [26] [http://www.nasa.gov/missions/science/dart\\_into\\_space.html](http://www.nasa.gov/missions/science/dart_into_space.html)
- [27] Weismuller, T., Leinz, M., GN&C Technology Demonstrated by the Orbital Express Autonomous Rendezvous and Capture Sensor System. Preceedings from the 29th Annual AAS Guidance and Control Conference, Breckenridge, CO, Feb. 4-8, 2006.
- [28] <http://www.spacedaily.com/spacenet/text/ets7-b.html>
- [29] [http://www.esa.int/Our\\_Activities/Human\\_Spaceflight/ATV](http://www.esa.int/Our_Activities/Human_Spaceflight/ATV)
- [30] [http://www.space.com/missionlaunches/xss10\\_update\\_030130.html](http://www.space.com/missionlaunches/xss10_update_030130.html)
- [31] Mitchell, I. T., Gorton, T. B., Taskov, K., Drews, M. E., Luckey, D., Osborne, M. L., Page, L. A., Norris, H. L. III, Shepperd, S. W., GN&C Development of the XSS-11 Micro Satellite for Autonomous Rendezvous and Proximity Operations. Preceedings from the 29th Annual AAS Guidance and Control Conference, Breckenridge, CO, Feb. 4-8, 2006.
- [32] <http://www.ohb-sweden.se/prisma>
- [33] <http://robonaut.jsc.nasa.gov/default.asp>
- [34] <http://en.wikipedia.org/wiki/Canadarm>
- [35] [http://en.wikipedia.org/wiki/Mobile\\_Servicing\\_System](http://en.wikipedia.org/wiki/Mobile_Servicing_System)
- [36] <http://www.nasaspaceflight.com/2013/01/dextre-rrm-satellite-refuelling-demo/>
- [37] Schwartz, J., Peck M., Hall C., "Historical Review of Air-Bearing Spacecraft Simulators", Journal of Guidance, Control, and Dynamics, vol. 26, no. 4, pp. 513-522, 2003.
- [38] <http://ssl.mit.edu/spheres/index.html>
- [39] Giorgio Guglieri, Franco Maroglio, Pasquale Pellegrino and Liliana Torre, Design and development of guidance navigation and control ALGORITHMS for spacecraft rendezvous and docking experimentATION, Acta Astronautica, <http://dx.doi.org/10.1016/j.actaastro.2013.02.010>.
- [40] <http://www.ni.com/white-paper/4074/en>
- [41] <http://www.mathworks.com/products/xpctarget/>

- [42] <https://www.mathworks.com/products/xpctarget/supported/hardware-drivers.html>
- [43] 'Προσομοίωση HIL (Hardware In the Loop) με τον Μικροελεγκτή INFINEON C167CS. έλεγχος Συστημάτων με τη Μέθοδο Βέλτιστου Πλάτους', ΑΡΓΥΡΙΟΣ Κ. ΤΖΑΚΑΣ ΜΑΡΤΙΟΣ 2006.
- [44] <http://www.arduino.cc/>
- [45] 'Serial Communication between Arduino and MATLAB', By-Aman Mangal, IIT Bombay, June 6, 2012.
- [46] Help του πακέτου λογισμικού Matlab 2012a
- [47] [http://www.mathworks.co.uk/help/releases/R13sp2/pdf\\_doc/xpc/xpc\\_target\\_gs.pdf](http://www.mathworks.co.uk/help/releases/R13sp2/pdf_doc/xpc/xpc_target_gs.pdf)
- [48] V-Realm Builder –User’s Guide and Reference
- [49] Simulink 3D Animation User’s Guide R2012a
- [50] [http://www.kxcad.net/cae\\_MATLAB/toolbox/xpc/ug/f4-19341.html](http://www.kxcad.net/cae_MATLAB/toolbox/xpc/ug/f4-19341.html)
- [51] 'Ανάπτυξη Εκπαιδευτικού Λογισμικού Για Προσομοίωση και Έλεγχο Πραγματικού Ρομποτικού Βραχίονα Τεσσάρων Βαθμών Ελευθερίας, Ανώτατο Εκπαιδευτικό Ίδρυμα Κρήτης, Τμήμα ηλεκτρονικής, Δημήτριο Θ. Τσόντου, Χανιά, 2009.
- [52] Papadopoulos, E., Paraskevas, I., Flessa, T., Nanos, K., and Kontolatis, I., "The NTUA Space Robot Simulator: Design & Results," 10th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA 2008), 11 - 13 November 2008, ESA/ESTEC, Noordwijk, The Netherlands.

# ΠΑΡΑΡΤΗΜΑ Α

## Κάρτες CPU PC-104 plus

Κάρτα	Network	ethernet chipsets	Επεξεργαστής	Εταιρεία	serial ports	usb ports	Κόστος
PCM-4153	2 x Fast Ethernet	Intel 82551ER	AMD LX800 500 MHz	advantech	4	4	301 E
CPU-1484	1xGigabit+1xFast	82541 + 82562	Intel® Pentium® M 1.4GHz	Eurotech	2	4	1102 E
CPU-1474	1xGigabit+1xFast	82541 + 82562	Intel® Celeron® M 1GHz	Eurotech	2	4	856 E
CPU-1454	1xFast + 1xGigabit	82541PI + 82562ET	Intel® Celeron® 400MHz	Eurotech			733 E
Cool LiteRunner-LX800	2 x Fast Ethernet	82551IT	AMD Geode LX800 500 MHz	Lippert Adlink	4	3	450 E
CPC304	2 x Fast Ethernet	2 x 82551IT	AMD Geode LX800 500 MHz	fastwel	4	2	611 E
CME136686LX500HR-128	2 x Fast Ethernet	Intel 82551QM	AMD Geode LX800 500 MHz	RTD	4	2	\$1545

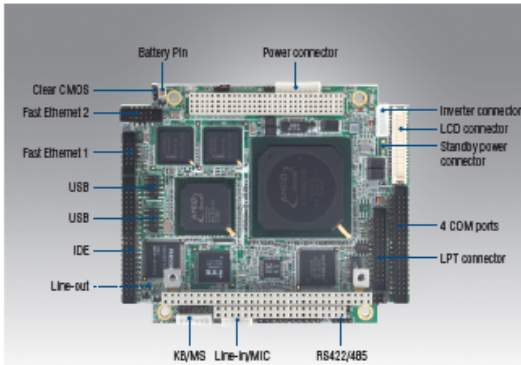
Provide a dedicated Ethernet card on your target PC. A dedicated Ethernet card is to be used only for model-based EtherCAT communications and not for communication between the host PC and target PC. This requirement means that your target PC must have at least two Ethernet cards, one for host PC to target PC communications, and one for model-based EtherCAT communication. The xPC Target software model-based EtherCAT communication blocks support the following Intel (Vendor ID 0x8086) chipsets:

Device ID	Chip Description
0x1229	Intel PRO/100 M Desktop Adapter: 82557, 82558, 82559, 82550, 82551
0x1209	Fast Ethernet Controller: 82551xER, 82551IT
0x1059	Fast Ethernet PCI Controller: 82551QM
0x2449	Integrated 10Base-T/100Base-T TX Ethernet Controller: 8x559ER
0x103A	LAN Controller: 82562ET, 82562EZ, 82563ET, 82801DB
0x1039	LAN Controller: 82562ET, 82562EZ, 82562VE, 82562VM
0x1050	Pro/100 VE Network Connection: 8201EB, 8201ER
0x27DC	Intel PRO/100 VE Desktop Adapter: 82562V

# Datasheet PC-104 CPU

## PCM-4153

AMD LX800 PC/104-Plus SBC, Onboard Memory/Flash, VGA, TTL, Extended Temp. -40 ~ 85° C



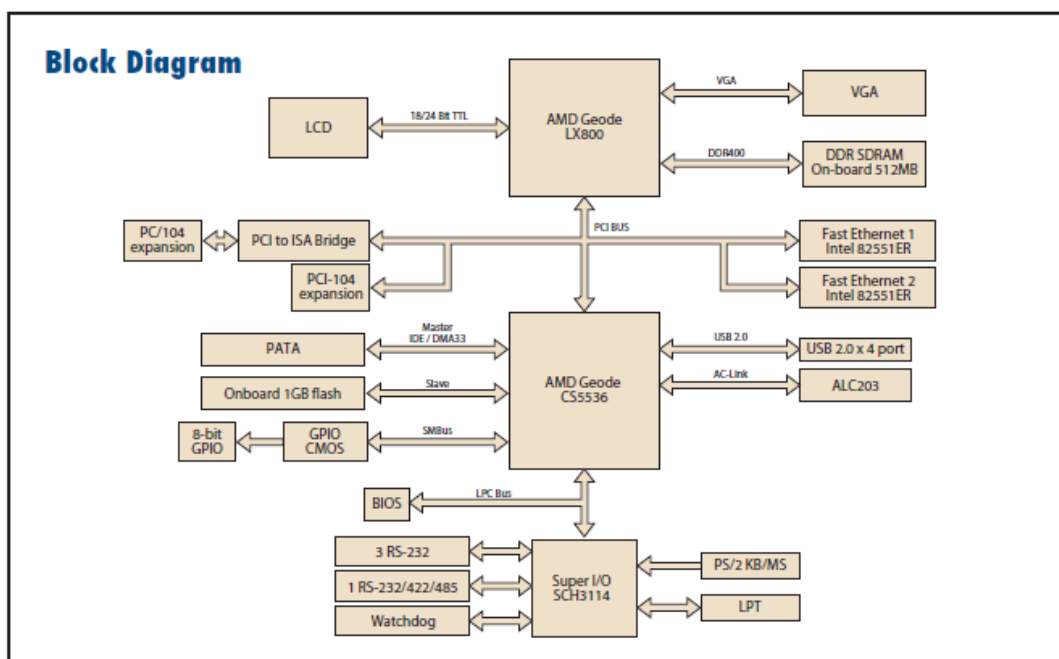
### Features

- AMD low power LX800 500 MHz processor
- Supports extended temperature range -40 ~ 85° C
- PC/104-plus expansion
- Onboard 1 GB Flash and onboard DDR 333MHz 512 MB memory
- Supports SUSIAccess and Embedded Software APIs



### Specifications

Processor System	CPU	AMD Geode™ LX800, 500 MHz
	Frequency	500 MHz
	L2 Cache	128 KB
	System Chipset	AMD Geode LX800, with AMD CS5536
	BIOS	Award 4-Mbit
Memory	Technology	DDR 333 MHz
	Max. Capacity	512 MB
	Onboard memory	512 MB
Display	Chipset	AMD Geode LX800
	VRAM	Optimized Shared Memory Architecture up to 64 MB system memory
	VGA	Supports up to 1920 x 1440 x 32 bpp at 85 Hz Supports up to 1600 x 1200 x 32 bpp at 100 Hz
	TTL LCD	Supports up to 1600 x 1200 x 32 bpp at 60 Hz for 24-bit single channel TFT
	Dual Display	VGA+TTL
Ethernet	Speed	10/100 Mbps
	Controller	Fast Ethernet1: Intel 82551 ER Fast Ethernet2: Intel 82551 ER
	Connector	Box Header
Audio	Chipset	Realtek ALC203
Watchdog Timer		Output System Reset Programmable counter from 1 ~ 255 minutes/ seconds
Storage	PATA	1 Channel
	Onboard Flash	1 GB (up to 4 GB)
	Floppy	1, 82077AA compatible (transfer from LPT, available by custom BIOS)
Internal I/O	USB	4 x USB 2.0
	Serial	3 RS-232 from COM1/3/4, 1 RS-232/422/485 from COM2 (ESD protection for RS-232: Air gap ±15kV, Contact ±8kV)
	Parallel (LPT)	1, IEEE 1284, EPP, and ECP compatible (FDD mode supported)
	SMBus	1
	Keyboard/Mouse	1
	GPIO	8-bit general purpose input/output
Expansion	PC/104-Plus Slot	1
Power	Power Type	AT
	Power Supply Voltage	5V ± 5% only to boot up (12 V is optional for LCD inverter and add on card)
	Power Consumption (Typical)	1.35 A @ +5 V, 0.1 A @ +12 V (7.95 Watt)
	Power Consumption (Max, test in HCT)	1.51 A @ +5 V, 0.1 A @ +12 V (8.75 Watt)
Environment	Power Management	ACPI/APM1.2
	Operational	0 ~ 60° C (32 ~ 140° F) (Operational humidity: 40° C @ 85% RH non-condensing)
Physical Characteristics	Non-Operational	-40° C ~ 85° C and 60° C @ 95% RH non-condensing
	Dimensions (L x W)	96 x 115 mm (3.8" x 4.5")
	Weight	0.574 kg (1.26 lb) (with heat-sink)
	Height	Top Side: 11.45 mm; Bottom Side: 10.6 mm



### Ordering Information

Part No.	CPU	Onboard Memory	VGA	TTL	Fast Ethernet	USB 2.0	RS-232	RS-232/422/485	Onboard Flash	Audio	Expansion	Thermal Solution	Operating Temp.
PCM-4153F-LOA2E	AMD LX800	512 MB	Yes	18/24 bit	2	4	3	1	1 GB	Yes	PC/104+	Passive	0 – 60° C
PCM-4153FZ-LOA2E	AMD LX800	512 MB	Yes	18/24 bit	2	4	3	1	1 GB	Yes	PC/104+	Passive	-20 – 80° C
PCM-4153FZ2-LOA2E	AMD LX800	512 MB	Yes	18/24 bit	2	4	3	1	1 GB	Yes	PC/104+	Passive	-40 – 85° C

Note: Passive = fanless; Active = with fan

### Packing List

Part No.	Description	Quantity
	PCM-4153 SBC	
	Startup Manual	
	Utility CD	
170000898	VGA cable D-SUB 15P(F)/12P-1.25 mm 15 cm	1
170000918	Audio cable 10 cm	1
170003491	AT power cable 1 x 8P-2.0/B4P-5.08 x 2 15 cm	1
170006202	Cable 6P-6P-6P PS/2 KB & Mouse 20 cm	1
170026250	LPT port cable 25P to 26P 2.0 mm 25 cm	1
1701100202	LAN flat cable IDC10P 2.0 mm/RJ-45 20 cm	1
1701400181	COM 4 ports flat cable 18 cm IDC40P 2.0 mm	1
1701440350	IDE cable 44P/44P/44P 35 cm	1
1703040157	RS-422/485 W/D-SUB COM 4P 15 cm	1
1703060053	PS2 cable 6P (MINI-DIN)-6P (Wafer 2.0 mm) 6 cm	1
1703100121	USB 2-port cable 10P 12 cm IDC 2.0 mm	1
9660104000	PC/104 screw and copper post package	1
1960005764	Heatsink for PCM-4153 (80.44 x 77.97 x 7.82 mm)	1

### Optional Accessories

Part No.	Description
1653130421	PCI-104 connector 120-pin (Long pin)
165313222B	PC/104 connector 64-pin (Long pin)
165312022B	PC/104 connector 40-pin (Long pin)
1700001531	LPT to FDD cable

### Embedded OS/API

Embedded OS/API	Part No.	Description
WinCE	2070000729	Image GX3 CE 5.0 Pro Plus Eng
	2070001612	CE 6.0 Pro GX3 4 Com V1.0 ENG
	2070001576	XPE FP2007 GX3 (LX800) V3.0 ENG
Win XPE	2070003216	XPE FP2007 GX3 Group V3.1 ENG (717.22 MB)
	2070003557	XPE FP2007 GX3 Group V3.0 CHS (641.41 MB)
QNX		V6.3.2/ 6.4.1
Software API	205E000019	SUSI 3.0 SW API for ESBC B: 20091116 XP

Online Download [www.advantech.com/products](http://www.advantech.com/products)


ADVANTECH



# Datasheet High Voltage DC-DC Converter

DC-DC

**75-350 Watts**

xppower.com 

**QSB Series**



- Wide Input Range
- Industry Standard Packages
- -40 °C to +100 °C Operating Temperature
- High Power Density
- Base Plate Cooled
- Remote On/Off & Remote Sense
- 3 Year Warranty

## Specification

### Input

- Input Voltage Range • See tables
- Input Current • See tables
- Input Reverse Voltage Protection • None
- Input Filter • Pi network
- Undervoltage Lockout • QSB75-150: 24 Vin - turn on 8.8 V, turn off 8.0 V for 9-36 V, QSB200: 24 Vin - turn on 9.6 V, turn off 8.8 V, QSB75-200: 48 Vin - turn on 17.0 V, turn off 16.0 V, QSB350: 24 Vin - turn on 17.0 V, turn off 16.0 V, QSB150: 48 Vin - turn on 35.0 V, turn off 33.0 V
- Input Surge • 24 Vin: 50 VDC for 100 ms  
48 Vin: 100 VDC for 100 ms

### Output

- Output Voltage Trim • ±10%, see application notes
- Initial Set Accuracy • ±1.5% max
- Line Regulation • ±0.2% max measured from high line to low line
- Load Regulation • ±0.2% max measured from 0-100% load
- Transient Response • 5% max deviation, recovery to within 1% in 500  $\mu$ s, 25% step load change
- Ripple & Noise • 3.3 & 5 V models: 100 mV pk-pk  
12 & 15 V models: 150 mV pk-pk  
24 V & 28 V models: 280 mV max pk-pk  
20 MHz bandwidth (see note 1)
- Oversvoltage Protection • 115-140%
- Short Circuit Protection • Continuous
- Thermal Shutdown • Case temperature >105 °C
- Temperature Coefficient • ±0.03%/°C
- Current Limit • 110-140% nominal output for 75 & 150 W  
110-160% nominal output for 100 W  
110-150% nominal output for 200 W  
105-140% nominal output for 350 W
- Remote On/Off • See notes 2 & 3
- Remote Sense • Compensates up to 10% of Vout nominal, total of output trim and remote sense

### General

- Efficiency • See tables
- Isolation Voltage • 1500 VDC Input to Output  
1500 VDC Input to Case  
1500 VDC Output to Case
- Isolation Resistance •  $10^7 \Omega$
- Isolation Capacitance • QSB75-200: 100 pF typical,  
QSB350: 1000 pF typical
- Switching Frequency • QSB75: 300 kHz typical  
QSB100, 150 & 200: 250 kHz typical  
QSB350 (3V3 & 5 V): 300 kHz typical  
QSB350 (12, 24 & 28 V): 330 kHz typical
- Power Density • QSB75: 45.4 W/in<sup>3</sup>, QSB100: 60.5 W/in<sup>3</sup>,  
QSB150: 57.2 W/in<sup>3</sup>, QSB200: 70.3 W/in<sup>3</sup>,  
QSB350: 123.0 W/in<sup>3</sup>
- MTBF • QSB75 & 100: 730 kHrs  
QSB150 & 200: 1 MHrs  
QSB350: 658 kHrs  
typical to MIL-HDBK-217F at 25 °C, GB

### Environmental

- Operating Base Plate Temperature • -40 °C to +100 °C, see derating curves
- Storage Temperature • -55 °C to +105 °C
- Operating Humidity • Up to 90% non-condensing
- Cooling • Base plate cooled, see derating curves

### EMC & Safety

- Emissions • EN55022, level A conducted with external components, see test reports for further details
- ESD Immunity • EN61000-4-2, level 2, Perf Criteria B
- Radiated Immunity • EN61000-4-3, 3 V/m, Perf Criteria A
- EFT/Burst • EN61000-4-4, level 1, Perf Criteria A
- Surge • EN61000-4-5, level 1, Perf Criteria A
- Conducted Immunity • EN61000-4-6, 3 V rms  
Perf Criteria A
- Safety Approvals • QSB150 & QSB200: UL60950-1



T H E X P E R T S I N P O W E R



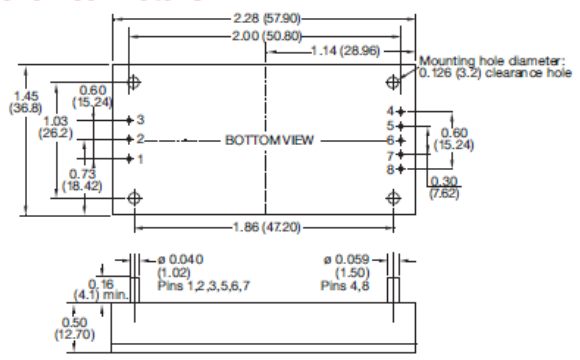
**Models and Ratings**

**QSB75/100 XP**

Input Voltage	Output Voltage	Output Current	Input Current <sup>1)</sup>		Efficiency	Model Number <sup>1)</sup>
			No Load	Full Load		
9-36 VDC (24 V nominal)	3.3 V	12.00 A	50 mA	2.04 A	81.0%	QSB7524S3V3
	5.0 V	12.00 A	50 mA	2.98 A	84.0%	QSB7524S05
	12.0 V	6.25 A	50 mA	3.64 A	86.0%	QSB7524S12
	15.0 V	5.00 A	50 mA	3.64 A	86.0%	QSB7524S15
	24.0 V	3.12 A	50 mA	3.63 A	86.0%	QSB7524S24
18-75 VDC (48 V nominal)	3.3 V	12.00 A	30 mA	1.10 A	82.0%	QSB7548S3V3
	5.0 V	12.00 A	30 mA	1.47 A	85.0%	QSB7548S05
	12.0 V	6.25 A	30 mA	1.82 A	86.0%	QSB7548S12
	15.0 V	5.00 A	30 mA	1.80 A	87.0%	QSB7548S15
	24.0 V	3.12 A	30 mA	1.80 A	87.0%	QSB7548S24
9-36 VDC (24 V nominal)	3.3 V	30.00 A	120 mA	4.80 A	86.0%	QSB10024S3V3
	5.0 V	20.00 A	120 mA	4.82 A	86.5%	QSB10024S05
	12.0 V	8.30 A	80 mA	4.80 A	86.5%	QSB10024S12
	15.0 V	6.70 A	80 mA	4.84 A	86.5%	QSB10024S15
18-75 VDC (48 V nominal)	3.3 V	30.00 A	60 mA	2.37 A	88.0%	QSB10048S3V3
	5.0 V	20.00 A	60 mA	2.37 A	88.0%	QSB10048S05
	12.0 V	8.30 A	30 mA	2.36 A	88.0%	QSB10048S12
	15.0 V	6.70 A	30 mA	2.38 A	88.0%	QSB10048S15
	24.0 V	4.17 A	30 mA	2.37 A	88.0%	QSB10048S24

- Notes**
1. Ripple & noise is measured with a 10 μF tantalum capacitor and 0.1 μF ceramic capacitor across output.
  2. Logic compatibility: Ref to -ve input. Module On = open circuit. Module Off = <0.8 VDC.
  3. Add suffix 'N' to the model number to receive the unit with negative logic Remote On/Off.
  4. Input current specified at 24 V for 9-36 VDC and 48 V for 18-75 VDC models.

**Mechanical Details**

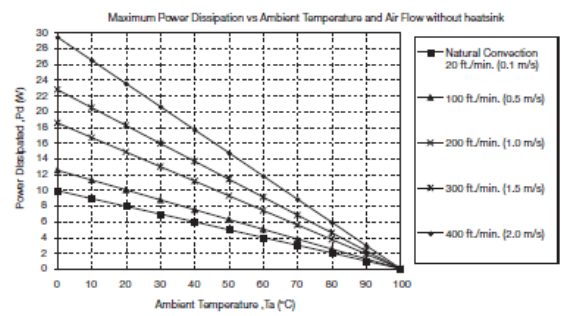


PIN CONNECTIONS	
Pin	Function
1	+Vin
2	Remote On/Off
3	-Vin
4	-Vout
5	-Sense
6	Trim
7	+Sense
8	+Vout

- Notes**
1. Dimensions are in inches (mm)
  2. Tolerances: XXX = ±0.02 (XX = ±0.5), XXXX = ±0.01 (XX = ±0.25)
  3. Weight: 0.15 lbs (66 g) approx

**Thermal Resistance Information**

**Derating Curve**



Air Flow Rate	Typical R <sub>ca</sub>
Natural Convection 20 ft./min (0.1 ms)	10.1 °C/W
100 ft./min (0.5 ms)	8.0 °C/W
200 ft./min (1.0 ms)	5.4 °C/W
300 ft./min (1.5 ms)	4.4 °C/W
400 ft./min (2.0 ms)	3.4 °C/W

R<sub>ca</sub> = Thermal resistance case to ambient



**Models & Ratings**

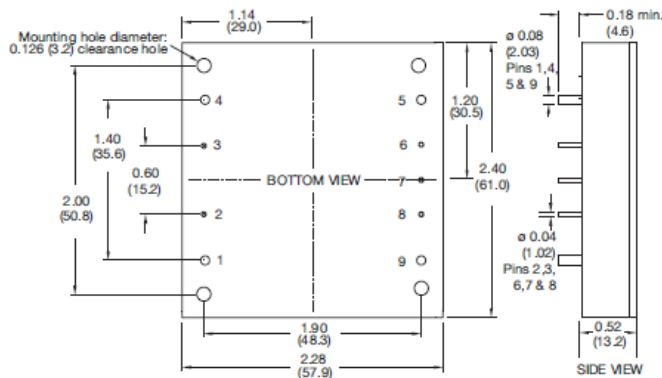
**QSB150-350 XP**

Input Voltage	Output Voltage	Output Current	Input Current <sup>(1)</sup>		Efficiency	Model Number <sup>(4)</sup>
			No Load	Full Load		
9-36 VDC (24 V nominal)	3.3 V	30.00 A	200 mA	4.75 A	87.0%	QSB15024S3V3
	5.0 V	30.00 A	200 mA	7.19 A	87.0%	QSB15024S05
	12.0 V	12.50 A	100 mA	7.10 A	88.0%	QSB15024S12
	15.0 V	10.00 A	100 mA	7.19 A	87.0%	QSB15024S15
	24.0 V	6.50 A	100 mA	7.34 A	88.0%	QSB15024S24
18-75 VDC (48 V nominal)	3.3 V	30.00 A	100 mA	2.34 A	88.0%	QSB15048S3V3
	5.0 V	30.00 A	100 mA	3.47 A	90.0%	QSB15048S05
	12.0 V	12.50 A	50 mA	3.44 A	91.0%	QSB15048S12
	15.0 V	10.00 A	50 mA	3.47 A	90.0%	QSB15048S15
	24.0 V	6.50 A	50 mA	3.61 A	90.0%	QSB15048S24
10-36 VDC (24 V nominal)	3.3 V	50.00 A	150 mA	7.90 A	87.0%	QSB20024S3V3
	5.0 V	40.00 A	150 mA	9.58 A	87.0%	QSB20024S05
	12.0 V	16.70 A	100 mA	9.71 A	86.0%	QSB20024S12
	15.0 V	13.30 A	100 mA	9.67 A	86.0%	QSB20024S15
	24.0 V	8.30 A	100 mA	9.54 A	87.0%	QSB20024S24
18-75 VDC (48 V nominal)	3.3 V	40.00 A	80 mA	3.13 A	88.0%	QSB20048S3V3
	5.0 V	40.00 A	80 mA	4.69 A	89.0%	QSB20048S05
	12.0 V	16.70 A	60 mA	4.74 A	88.0%	QSB20048S12
	15.0 V	13.30 A	60 mA	4.72 A	88.0%	QSB20048S15
	24.0 V	8.30 A	60 mA	7.72 A	88.0%	QSB20048S24
18-36 VDC (24 V nominal)	3.3 V	70.0 A	130 mA	10.94 A	88.0%	QSB35024S3V3
	5.0 V	70.0 A	250 mA	16.29 A	89.5%	QSB35024S05
	12.0 V	29.2 A	220 mA	15.96 A	91.5%	QSB35024S12
	24.0 V	14.6 A	40 mA	16.22 A	90.0%	QSB35024S24
	28.0 V	12.5 A	50 mA	16.03 A	91.0%	QSB35024S28
36-75 VDC (48 V nominal)	3.3 V	70.0 A	80 mA	5.41 A	89.0%	QSB35048S3V3
	5.0 V	70.0 A	120 mA	8.01 A	91.0%	QSB35048S05
	12.0 V	29.2 A	100 mA	7.89 A	92.5%	QSB35048S12
	24.0 V	14.6 A	40 mA	7.98 A	91.5%	QSB35048S24
	28.0 V	12.5 A	30 mA	7.88 S	92.5%	QSB35048S28

**Notes**

- Ripple & noise is measured with a 10  $\mu$ F tantalum capacitor and 0.1  $\mu$ F ceramic capacitor across output.
- Logic compatibility: Ref to -ve input. Module On = open circuit. Module Off = <math>\lt; 0.8 \text{ VDC}</math>.
- Add suffix 'N' to the model number to receive the unit with negative logic Remote On/Off.
- Input current specified at 24 V for 10-36 VDC models and 48 V for 18-75 VDC models.

**Mechanical Details**



Pin	PIN CONNECTIONS Function
1	+Vin
2	Remote On/Off
3	Case
4	-Vin
5	-Vout
6	-Sense
7	Trim
8	+Sense
9	+Vout

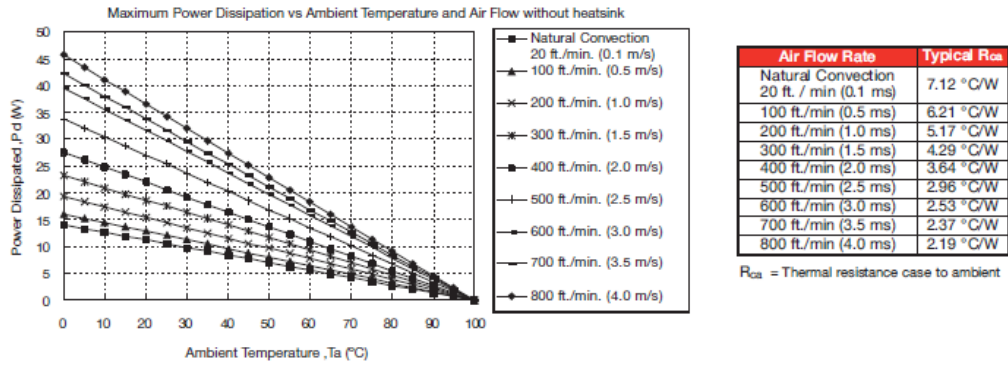
**Notes**

- Dimensions are in inches (mm)
- Tolerances: X.XX =  $\pm 0.02$  (XX =  $\pm 0.5$ )  
X.XXX =  $\pm 0.01$  (XXX =  $\pm 0.25$ )
- Weight: 0.25 lbs (114 g) approx



THE X P E R T S I N P O W E R

Derating Curve



Application Notes

Output Voltage Trim - QSB75 & QSB150 Series

Voltage trim up

Connect trim resistor  $R_{trim}$  between Trim pin and -Sense pin.

$$R_{trim\ up} = \left( \frac{R1 \left( Vr - Vf \left( \frac{R2}{R2 + R3} \right) \right)}{V_{des} - V_{nom}} \right) \cdot \frac{R2 \times R3}{R2 + R3} \text{ (k}\Omega\text{)}$$

Voltage trim down

Connect trim resistor  $R_{trim}$  between Trim pin and +Sense pin

$$R_{trim\ down} = \frac{R1 \times (V_{des} - Vr)}{V_{nom} - V_{des}} - R2 \text{ (k}\Omega\text{)}$$

Where:  $R_{trim}$  up/down is the external resistor in k $\Omega$ .  $V_{nom}$  is the nominal output voltage.  $V_{des}$  is the desired output voltage.  $R1$ ,  $R2$ ,  $R3$  and  $Vr$  are internal to the unit and are defined in the table below.

Output Voltage (V)	R1 (k $\Omega$ )	R2 (k $\Omega$ )	R3 (k $\Omega$ )	Vr (V)	Vf (V)
3.3 V	3.00	12.0	4.3	1.24	0.46
5.0 V	2.32	3.3	0.0	2.5	0.0
12.0 V	9.10	51.0	5.1	2.5	0.46
15.0 V	12.0	56.0	8.25	2.5	0.46
24.0 V	20.0	100.0	7.5	2.5	0.46

Output Voltage Trim - QSB100, QSB200 & QSB350 Series

Voltage trim down - Connect trim resistor  $R_{trim}$  between Trim pin and -Sense pin

$$R_{trim\ down} = \left( \frac{511}{\Delta\%} - 10.22 \right) \text{ k}\Omega$$

Where:

$$\Delta\% = \left( \frac{V_{nom} - V_{des}}{V_{nom}} \right) \times 100$$

Voltage trim up - Voltage trim up, Connect trim resistor  $R_{trim}$  between Trim pin and +Sense pin.

$$R_{trim\ up} = \left( \frac{5.11 \times V_{nom} (100 + \Delta\%)}{1.225 \times \Delta\%} - \frac{511}{\Delta\%} - 10.22 \right) \text{ k}\Omega$$

Where:

$$\Delta\% = \left( \frac{V_{des} - V_{nom}}{V_{nom}} \right) \times 100$$



# Datasheet Wireless Ethernet Bridge

**NETGEAR**  
Connect with Innovation™

## Universal WiFi Internet Adapter WNCE2001

Data Sheet

**N300**  
2.4 GHz



### Connect any device with a network port to your existing home wireless network

- Wirelessly connects networked home theater devices such as Xbox 360®, PS3™, TiVo®, VUDU, Blu-ray player, and Internet TVs to your home network.
- Universal—works with any network enabled device. No need to buy a Wi-Fi adapter that only works with one type of device
- Powered by any USB port, avoids an additional power outlet
- Avoid long stringing cables. Freedom to place your home entertainment center anywhere and still connect to the Internet
- Compact design provides space-saving placement on your HDTV<sup>1</sup>
- Push 'N' Connect secures connection to the existing network at a push of a button, true plug-and-play

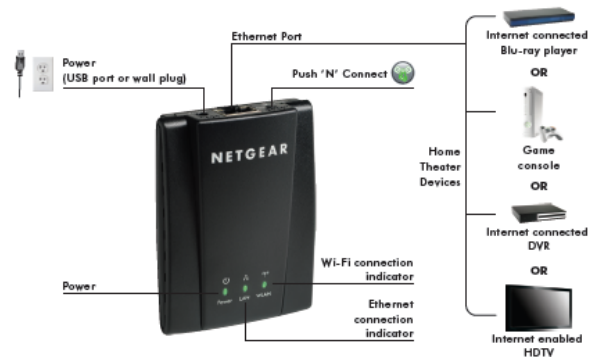


<sup>1</sup>For best performance place adapter in front or to side of television.

### The NETGEAR Difference

#### WNCE2001

- ✓ Universal compatibility
- ✓ CD-less, wireless installation
- ✓ Push 'N' Connect
- ✓ Powered by USB port



1-888-NETGEAR (638-4327)  
Email: info@NETGEAR.com



**System Requirements**

- Existing wireless home network with Internet connection
- 2.4 GHz 802.11 b/g/n AP/gateway/router
- TV, DVR, game console or home A/V device with Ethernet port
- Optional — USB port on A/V device to power
- Microsoft® Internet Explorer® 5.0, Firefox® 2.0 or Safari 1.4 or higher for configuration

**Features**

- Connects to any existing 2.4 GHz Wi-Fi access point, gateway or router
- Push 'N' Connect using Wi-Fi Protected Setup® for easy setup
- Dynamic LEDs display real-time network connection status
- Wi-Fi Protected Access® (WPA2-PSK, WPA-PSK) and WEP
- One (1) Ethernet port to connect to home A/V devices

**Standards**

- IEEE 802.11n version 2.0 at 2.4 GHz

**Physical Specifications**

- Dimensions: 81.3 x 62 x 17.9mm (3.20 x 2.44 x .70 in)
- Weight: 60 g (2.12 oz)

**Package Contents**

- Universal WiFi Internet Adapter (WNCE2001)
- Ethernet cable
- USB power cable
- Power adapter, localized to country of sale

**Warranty**

- Localized to country of sale

Learn more at [www.netgear.com/connectWiFi](http://www.netgear.com/connectWiFi)

**NETGEAR®**

350 E. Plumeria Drive  
San Jose, CA 95134-1911  
1-888-NETGEAR (638-4327)  
E-mail: [info@NETGEAR.com](mailto:info@NETGEAR.com)  
[www.NETGEAR.com](http://www.NETGEAR.com)

© 2010 NETGEAR, Inc. NETGEAR, the NETGEAR Logo, and Connect with Innovation are trademarks and/or registered trademarks of NETGEAR, Inc. and/or subsidiaries in the United States and/or other countries. Mac and the Mac logo are trademarks of Apple Inc., registered in the U.S. and other countries. Other brand names mentioned herein are for identification purposes only and may be trademarks of their respective holder(s). Information is subject to change without notice. All rights reserved.

This product is packaged with a limited warranty, the acceptance of which is a condition of sale.

\*Basic technical support provided for 90 days from date of purchase.

<sup>1</sup>Maximum wireless signal rate derived from IEEE standard 802.11 specifications. Actual data throughput will vary. Network conditions and environmental factors, including volume of network traffic, building materials and construction, and network overhead, lower actual data throughput rate. NETGEAR makes no express or implied representations or warranties about this product's compatibility with any future standards.

<sup>2</sup>N\*, "Wireless-N" and "802.11n" refer to compliance with IEEE 802.11n specification version 2.0 only.

D:WNCE2001-2

# Optical sensor Datasheet

# Microsoft® Ready Mouse

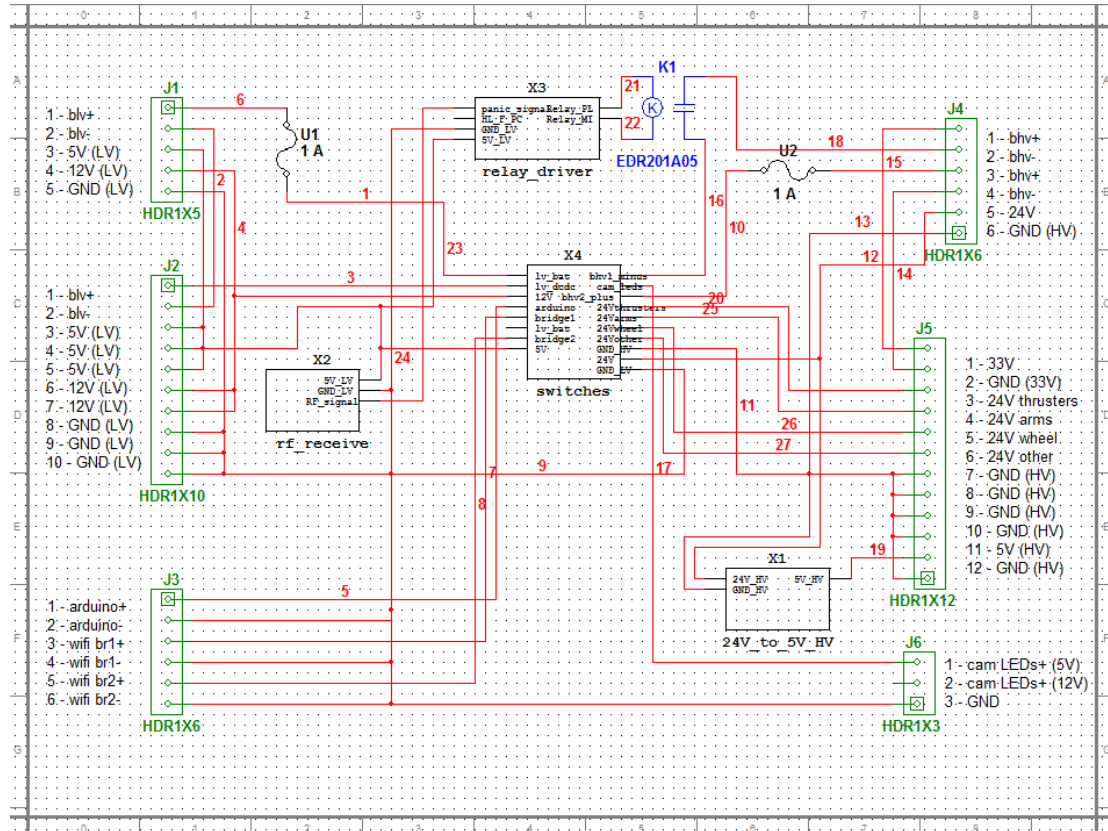


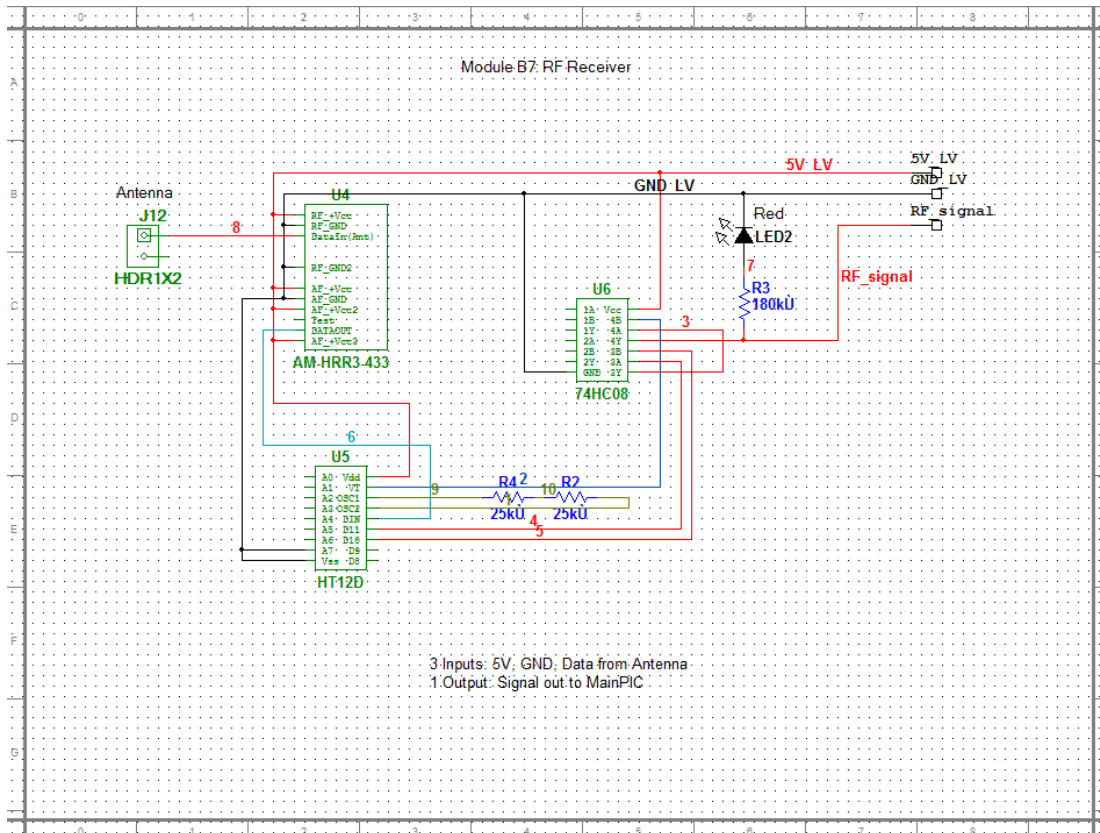
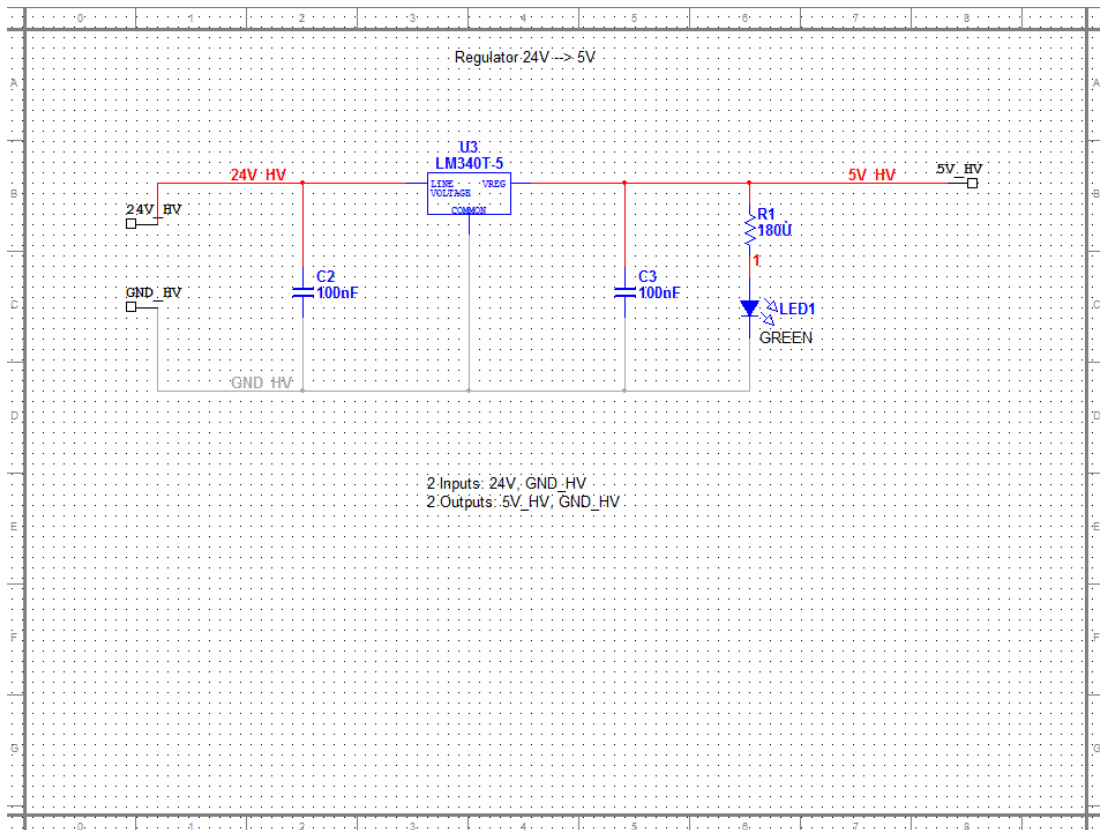
Version Information	
Product Name	Microsoft® Ready Mouse
Product Version	Microsoft Ready Mouse
Mouse Version	Microsoft Ready Mouse
Product Dimensions	
Mouse Length	4.47 inches (1134 millimeters)
Mouse Width	2.28 inches (57.9 millimeters)
Mouse Depth/Height	1.54 inches (39.2 millimeters)
Mouse Weight	3.27 ounces (92.8 grams)
Cable Length	72 +6/-0 inches (1830 +150/-0 millimeters)
Compatibility and Localization	
Interface	USB Compatible
Operating Systems	<ul style="list-style-type: none"> <li>• Microsoft Windows™ 7, Windows Vista™, and Windows XP (excluding Windows XP 64-bit)</li> <li>• Mac OS X v10.1-10.5.7</li> </ul>
Top-line System Requirements	<p><b>For a Windows-based PC:</b> Requires a PC that meets the requirements for and has installed one of these operating systems:  <ul style="list-style-type: none"> <li>• Microsoft Windows 7, Windows Vista, or Windows XP (excluding Windows XP 64-bit)</li> <li>• USB port or PS/2 port</li> <li>• No software required</li> </ul> </p> <p><b>For a Macintosh:</b>  <ul style="list-style-type: none"> <li>• Mac OS X v10.1-10.5.7</li> <li>• USB port</li> <li>• No software required</li> </ul> </p>
Compatibility Logos	<ul style="list-style-type: none"> <li>• Compatible with Microsoft Windows 7</li> <li>• Certified for Microsoft Windows Vista</li> <li>• Mac logo</li> <li>• Universal Mac logo</li> <li>• Certified USB logo</li> </ul>
Tracking Technology	
Mouse Tracking System	Optical tracking system
Imaging Rate	Up to 3000 frames per second
X-Y Resolution	800 points per inch (31.5 points per millimeter)
Tracking Speed	Up to 20 inches (508 millimeters) per second
Product Feature Performance	
Mouse Button Features	3 buttons including scroll wheel button
Right & Left Button Life	3,000,000 actuations
Wheel Button Life	250,000 actuations
Mouse Scrolling Features	Standard vertical scroll wheel
Wheel Vertical Scrolling Life	<ul style="list-style-type: none"> <li>• 100,000 revolutions (away from user) at no more than 60 rotations per minute</li> <li>• 300,000 revolutions (toward user) at no more than 60 rotations per minute</li> </ul>
Storage Temperature & Humidity	-40 °F (-40 °C) to 140 °F (60 °C) at <5% to 65% relative humidity (non-condensing)
Operating Temperature & Humidity	14 °F (-10 °C) to 104 °F (40 °C) at <5% to 80% relative humidity (non-condensing)
Certification Information	
Country of Manufacture	People's Republic of China (PRC)
ISO 9001 Qualified Manufacturer	Yes
ISO 14001 Qualified Manufacturer	Yes
Restriction on Hazardous Substances	This device complies with all applicable worldwide regulations and restrictions including, but not limited to: EU directive 2002/95/EC on the Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment and EU Registration Evaluation and Authorization of Chemicals (REACH) regulation regarding Substances of Very High Concern.
FCC ID	This device complies with part 15 of the FCC Rules and Industry Canada ICES-003. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation. Tested to comply with FCC standards. For home and office use. Model number: 1113, Microsoft Ready Mouse.
Agency and Regulatory Marks	<ul style="list-style-type: none"> <li>• ACMA Declaration of Conformity (Australia and New Zealand)</li> <li>• ICES-003 report on file (Canada)</li> <li>• EIP Pollution Control Mark, EPUP (China)</li> <li>• CE Declaration of Conformity, Safety and EMC (European Union)</li> <li>• WEEE (European Union)</li> <li>• VCCI Certificate (Japan)</li> <li>• MIC Certificate (Korea)</li> <li>• GOST Certificate (Russia)</li> <li>• CITC Letter (Kingdom of Saudi Arabia)</li> <li>• BSMI Certificate (Taiwan)</li> <li>• FCC Declaration of Conformity (USA)</li> <li>• UL and cUL Notice of Approval (USA and Canada)</li> <li>• CB Scheme Certificate (International)</li> </ul>
Windows Hardware Quality Labs (WHQL)	ID: PENDING Microsoft Windows 7

# ΠΑΡΑΡΤΗΜΑ Β

## Σχηματικά ηλεκτρονικών πλακετών (NI Multisim)

### Σχηματικό πλακέτας διανομής ισχύος

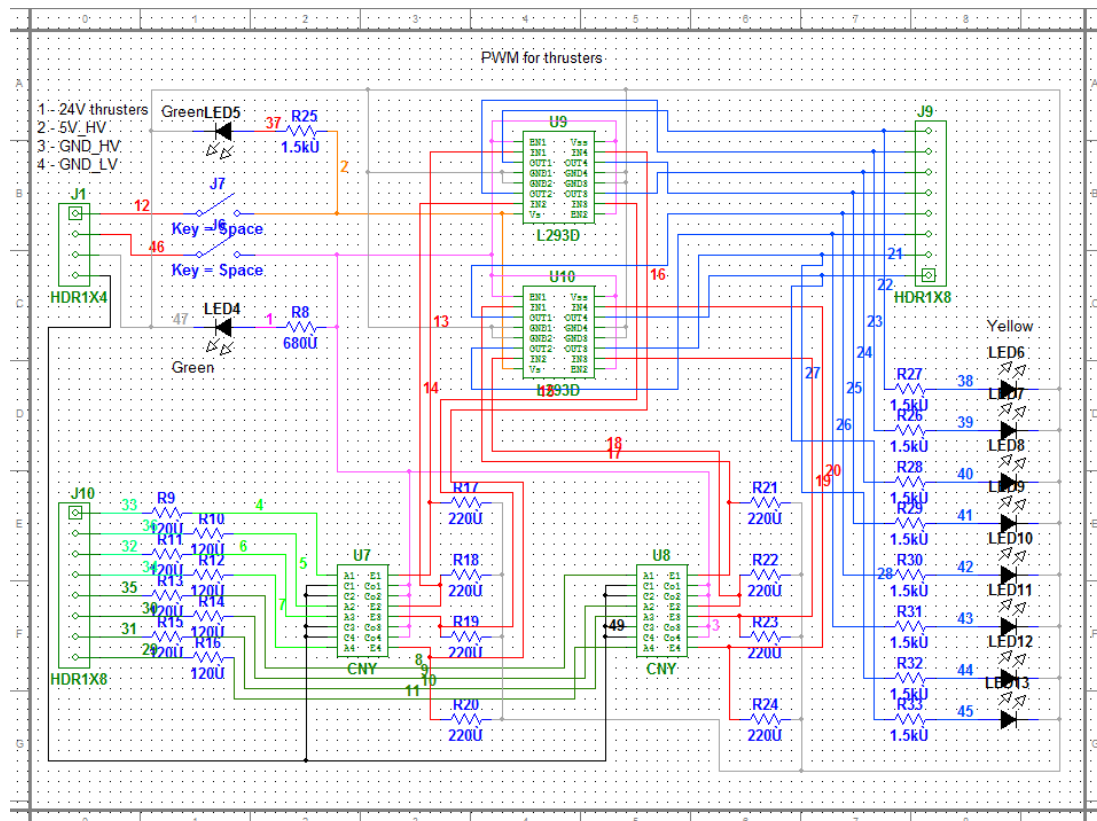




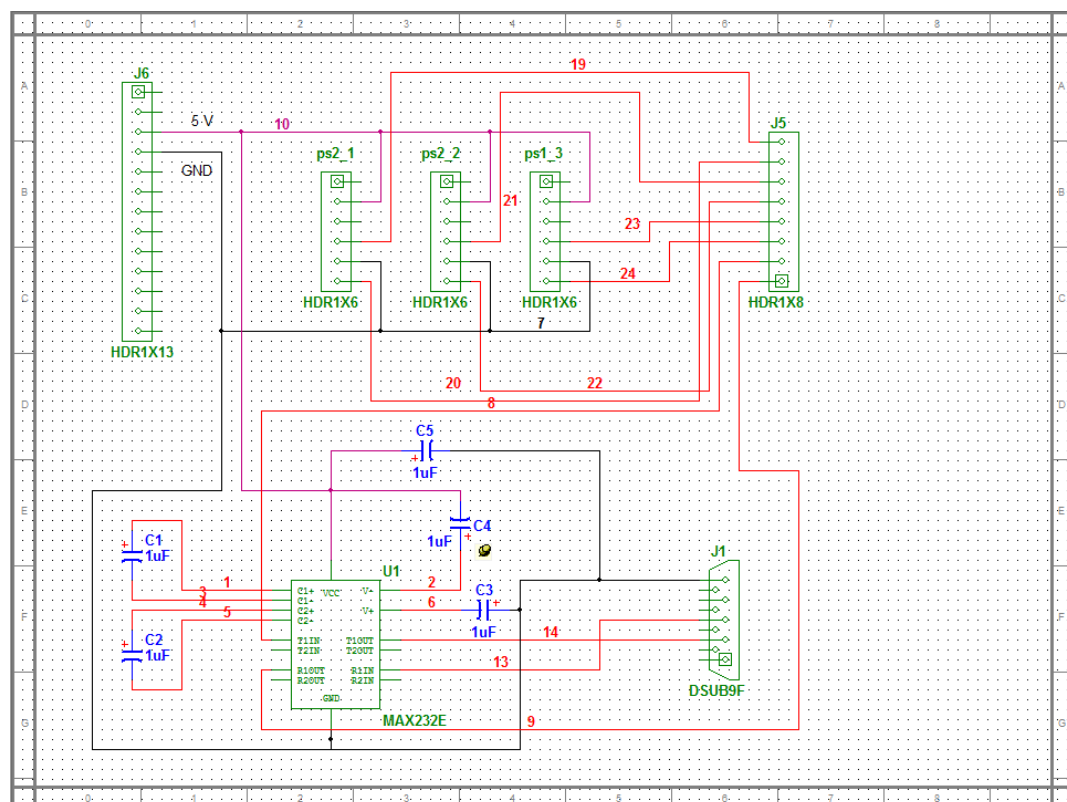




## Σχηματικό πλακέτας οδήγησης των thrusters

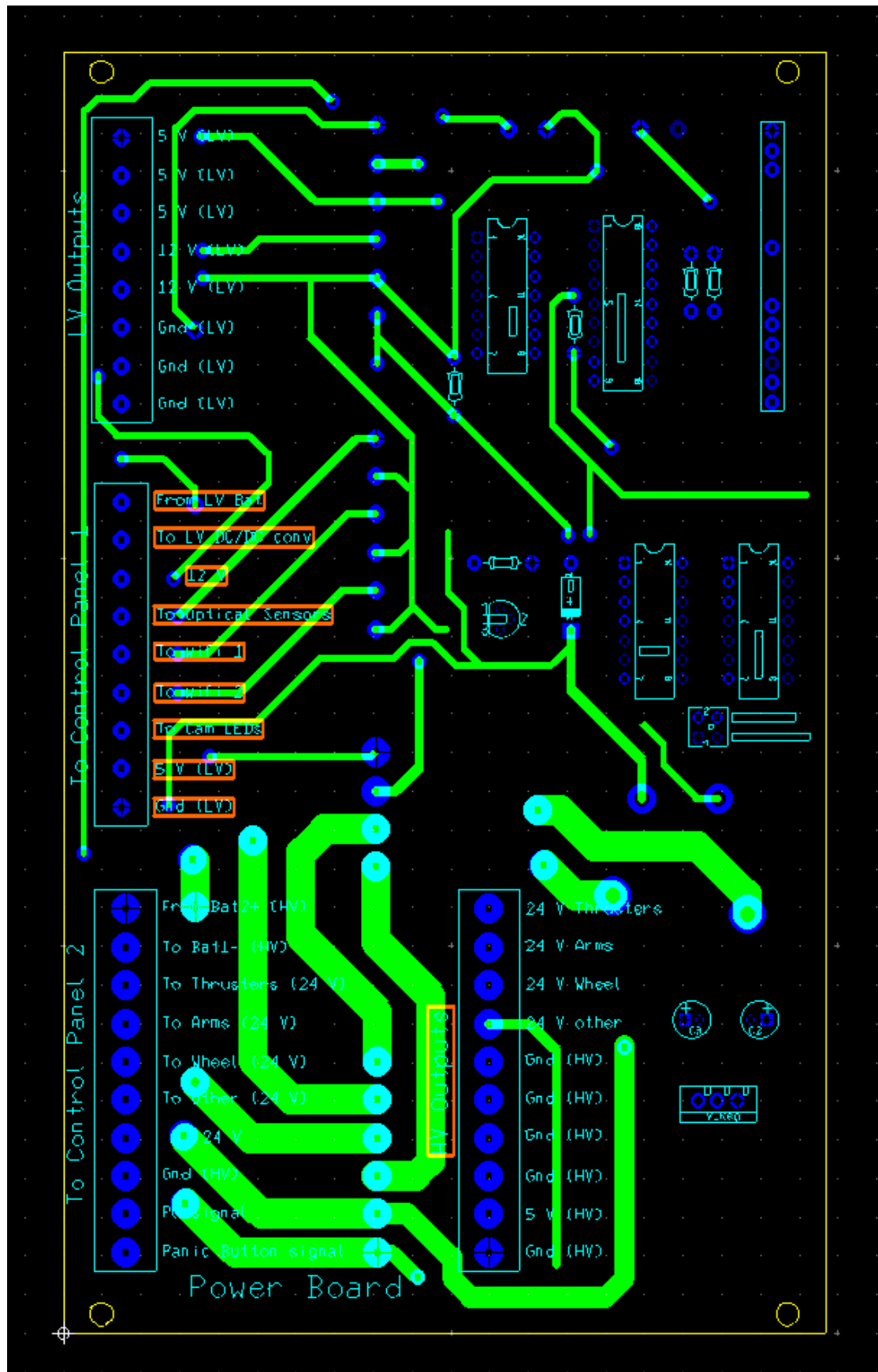


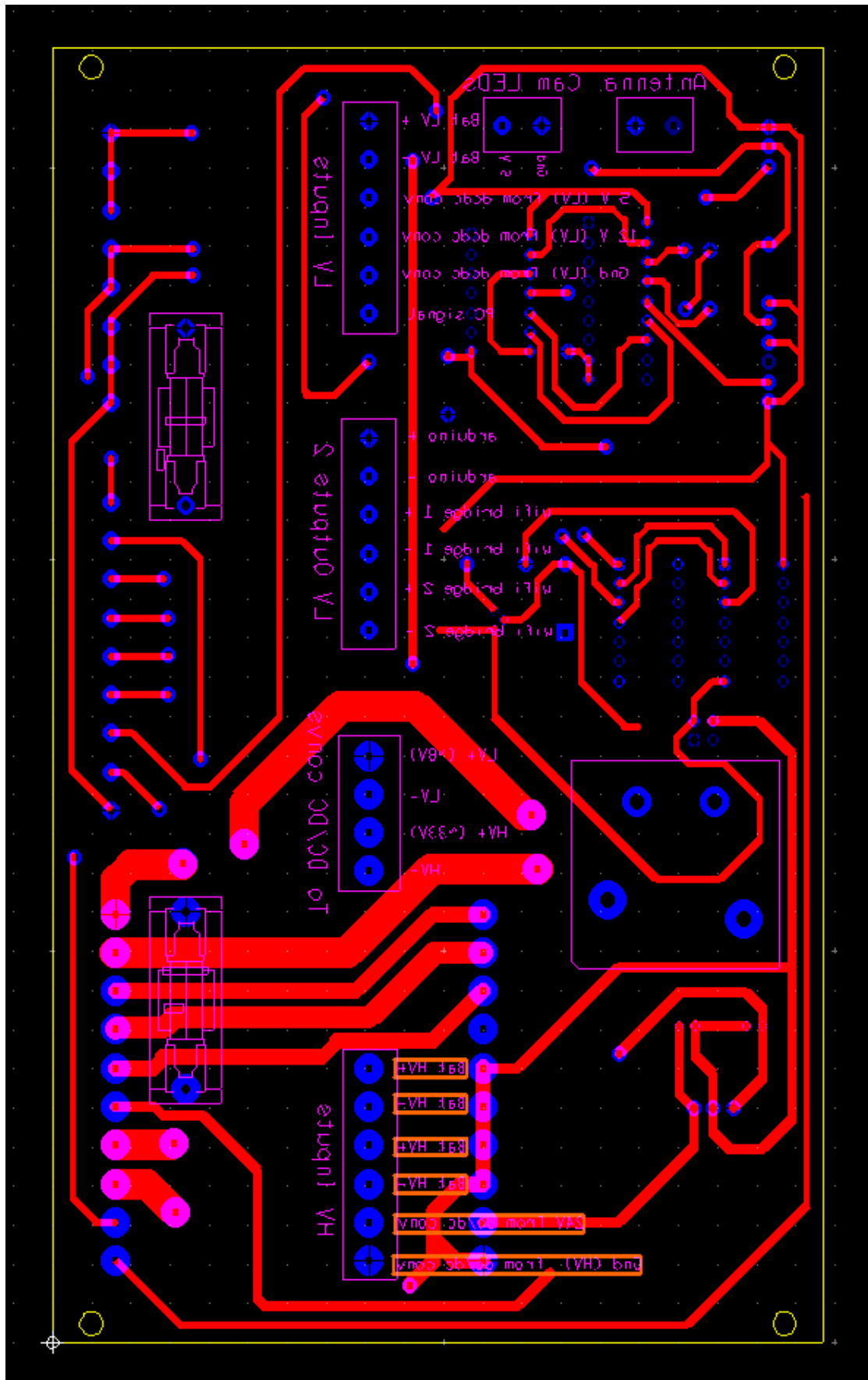
## Σχηματικό Arduino Shield



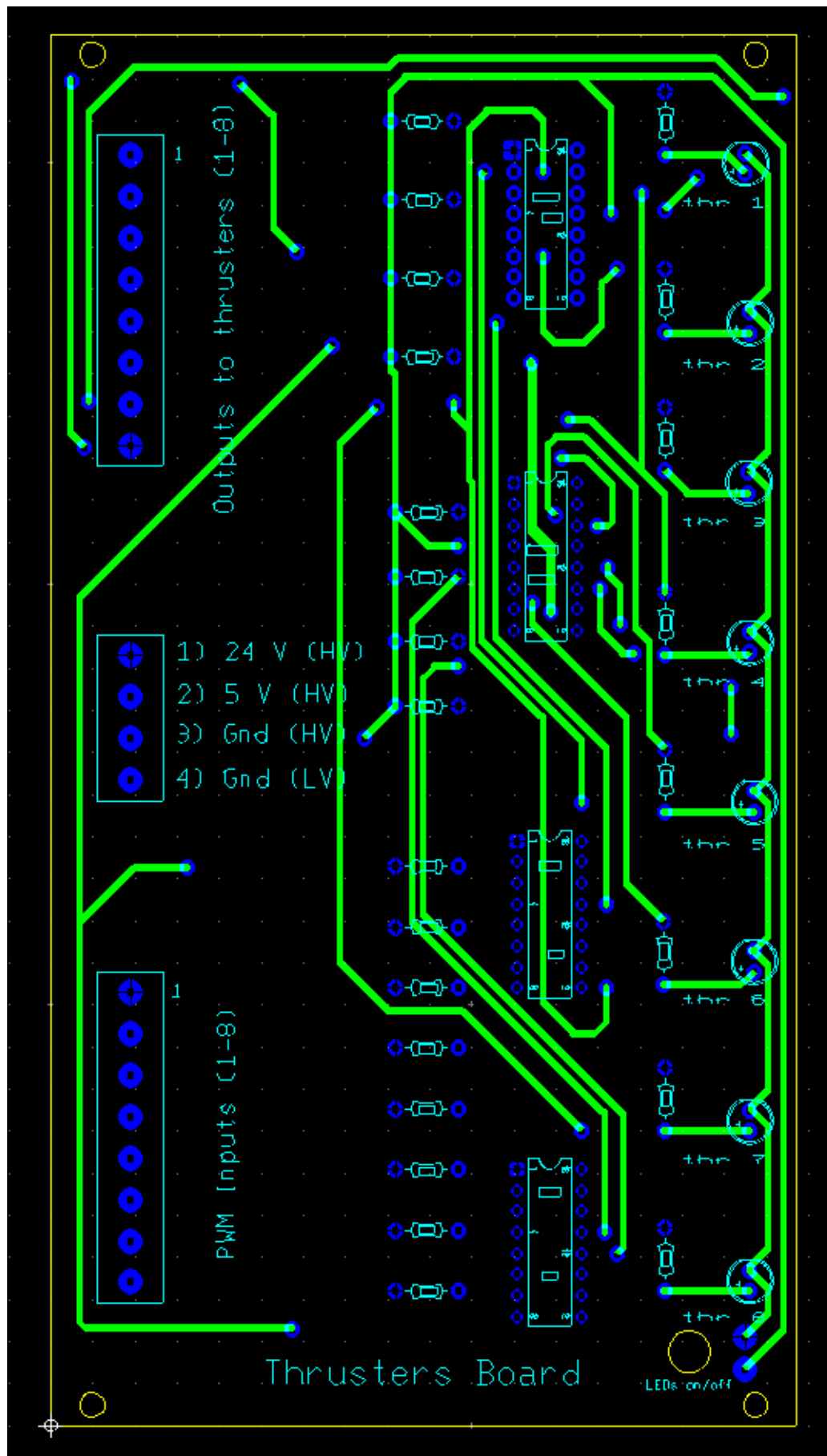
## Layout πλακετών - PCB Designs (NI Ultiboard)

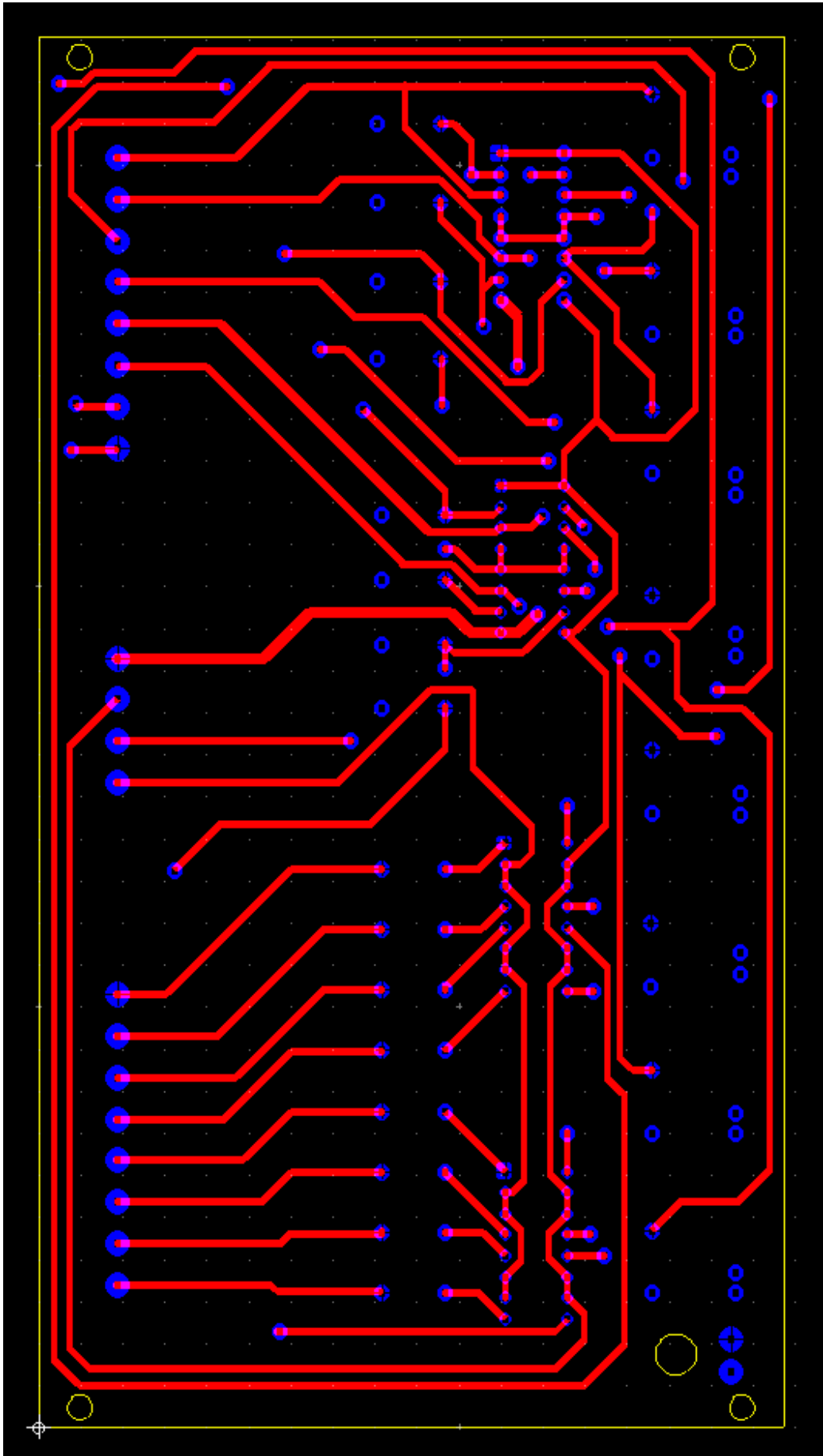
### Πλακέτα διανομής Ισχύος (Power board)



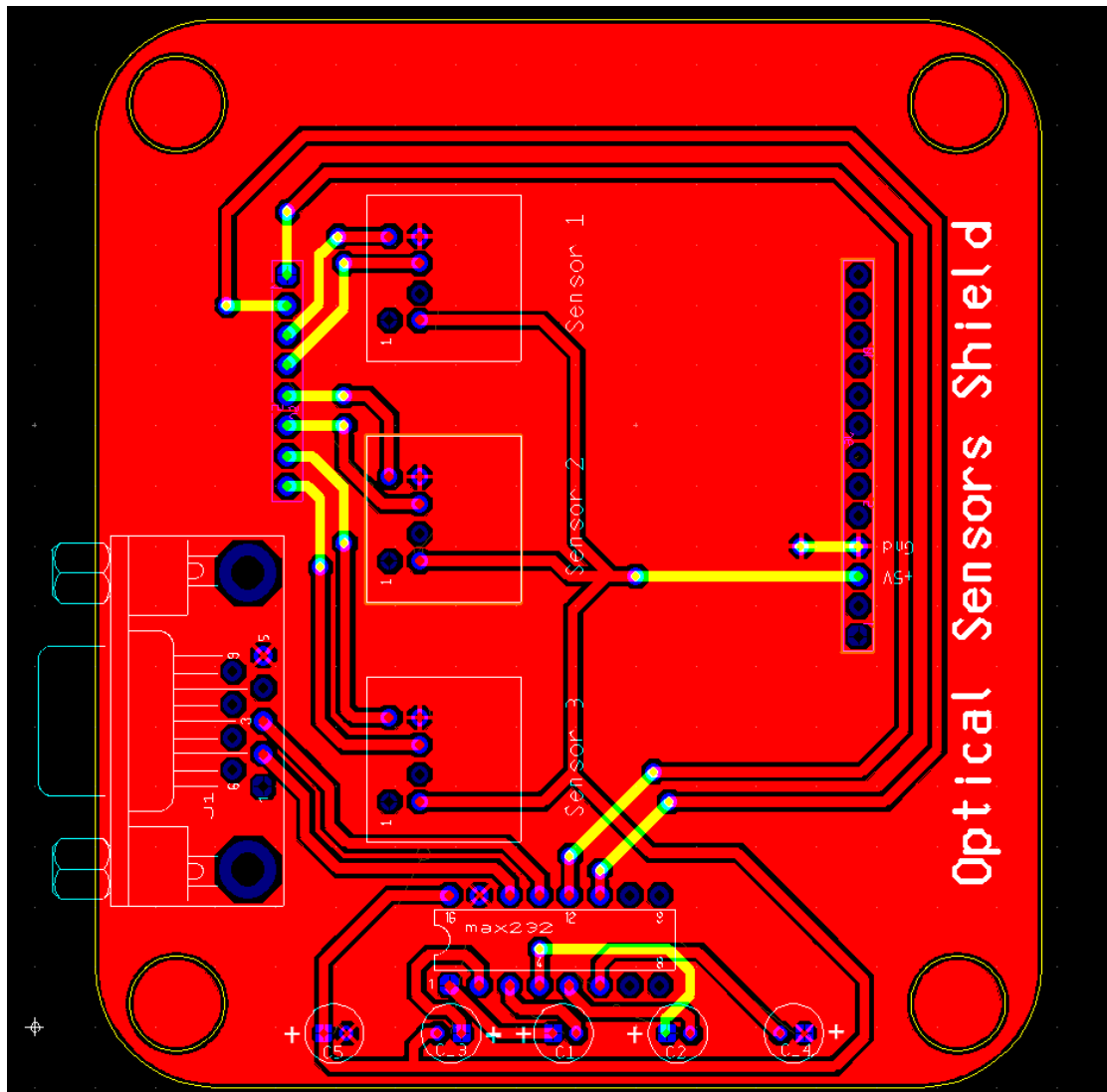


## Πλακέτα οδήγησης προωθητηρίων (Thrusters board)





## Πλακέτα Οπτικών Αισθητήρων (Optical sensors Shield)



# ΠΑΡΑΡΤΗΜΑ Γ

## Πρόγραμμα matlab για εύρεση θέσης και προσανατολισμού ενός σώματος με φωτεινές πηγές σε σχηματισμό

```
clc; clear all;
tic;

% seira led --> 1 2 3 4 5
% o pinakas 'distances_real' perixei tis pragmatikes apostaseis tw n led (to kathena apo ta upoloipa)
% de xreiazetai na einai sti morfi summetrikou
% evala tuxaies times

distances_real = [0 3 5 2 8
                  3 0 9 1 2
                  5 9 0 4 6
                  2 1 4 0 7
                  8 2 6 7 0];

% estw oti metrountai se auti ti seira se sxesi me ta parapanw --> 2 1 5 4 3
% vriskoume tis apostaseis tw n led metaksu tous (=10 apostaseis) kai
% gemizoume to pinaka 'distances_measured'
% evala ki ena sfalma 0.4 stis metroumenes apostaseis
% de xreiazetai na einai summetrikos

distances_measured = [0 3.4 2.4 1.4 9.4
                      3.4 0 8.4 2.4 5.4
                      2.4 8.4 0 7.4 6.4
                      1.4 2.4 7.4 0 4.4
                      9.4 5.4 6.4 4.4 0];

% algorithmos pou psaxnei na vrei poio led einai poio

seira = [0 0 0 0 0];
i=1;k=1;

while i<6 % trexei mexri na gemisei to pinaka 'seira'
    c=0; l=1; j=1; % to 'c' deixnei posa koina stoixeia exei mia seira tou 'distances_real' me mia tou 'distances_measured'
    while (l<6) && (c<4) % trexei mexri na vrei 4 koines apostaseis metaksi 2 seirwn tw n 2 pinakwn h mexri na teleiwsei h seira
        if abs(distances_real(k,l)-distances_measured(i,j))<0.5 % an 2 stoixeia diaferoun kata 0.5 theorountai idia
            c=c+1;
            j=j+1; % paei na psaksei gia to epomeno stoixeio ths seiras tou 'distances_measured'
            l=0; % psaxnei apo thn arxi th seira tou 'distances_real'
        end
        l=l+1; %proxwra sto epomeno stoixeio ths seiras tou 'distances_real'
    end
    if c==4 % an sto prohgomeno loop vrethikan 4 koines apostaseis, tote vrethike kai mia antistoixisi
        seira(i)=k; % to i einai to k
        i = i+1; % paei na vrei se poia seira tou 'distances_real' antistoixei h epomenh seira tou 'distances_measured'
        k = 1;
    else
        k=k+1; % an de vrei sugrinei me thn epomenh seira tou 'distances_real'
    end
end

seira % tupwnei poia einai telika ta led pou metrisame se sxesi me ta arxika
toc % apaitoumenos xronos upologismou (o algorithmos mporei na veltistopoiithe)
```