



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ Μ.Κ. & Α.Ε.

Εργαστήριο Αυτομάτου Ελέγχου

Μεταπτυχιακή Εργασία

**Σχεδιασμός και Κατασκευή Φορείου, με Κάμερα και Αλγόριθμο Επεξεργασίας
Εικόνας, για τον Έλεγχο Κίνησης Ρομποτικού Ψαριού.**

Σωτήρης Χρ. Ντούσκας

Επιβλέπων Καθηγητής: Ε. Γ. Παπαδόπουλος

ΑΘΗΝΑ 2011

Περίληψη

Σκοπός της εργασίας είναι ο έλεγχος και η καταγραφή των φυσικών μεγεθών της κίνησης ρομποτικού ψαριού. Το ρομποτικό ψάρι έχει κατασκευαστεί στο Εργαστήριο Αυτόματου Ελέγχου του τμήματος Μηχανολόγων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου. Για την εξυπηρέτηση του σκοπού αυτού, η εργασία, χωρίζεται σε δυο κυρίως μέρη, το κατασκευαστικό και το υπολογιστικό.

Το πρώτο μέρος αφορά στο σχεδιασμό και στην κατασκευή φορείου, στο οποίο προσδένεται το ρομποτικό ψάρι. Το φορείο μπορεί να εκτελεί οποιαδήποτε επίπεδη κίνηση. Με την χρησιμοποίηση αισθητήρα δύναμης, μπορούν να καταγράφονται οι δυνάμεις που ασκούνται στο ψάρι καθ'όλη τη διάρκεια της κίνησης του. Επιπλέον στο φορείο μπορεί να προσαρμόζεται κάμερα. Έτσι, όταν το ρομποτικό ψάρι κινείται αυτόνομα, το φορείο, μέσω αλγόριθμου επεξεργασίας εικόνας, θα μπορεί να ακολουθεί κάθε κίνηση του. Για το σχεδιασμό του φορείου, πρέπει να ληφθεί υπ'όψη, η υποδομή που υπάρχει στο εργαστήριο. Για παραδειγμα, το φορείο θα πρέπει να προσαρμόζεται στην δεξαμενή, και να κινείται χωρίς να υπάρχει κίνδυνος σύγκρουσης με άλλα αντικείμενα στο χώρο αυτό.

Το υπολογιστικό μέρος, αφορά στην μελέτη των φυσικών μεγεθών της κίνησης του ρομποτικού ψαριού. Τέτοια είναι η τροχιά, η ταχύτητα και η γωνία περιστροφής του σώματος. Ο υπολογισμός των μεγεθών αυτών, γίνεται με παρακολούθηση του ψαριού μέσω κάμερας. Με τη βοήθεια αλγόριθμου επεξεργασίας εικόνας, εντοπίζεται το ψάρι, και παρακολουθείται, και τα δεδομένα που καταγράφονται αποθηκεύονται για περαιτέρω επεξεργασία. Στη συνέχεια λαμβάνει χώρα η επεξεργασία των δεδομένων αυτών, και εξάγονται τα επιθυμητά αποτελέσματα. Τα αποτελέσματα μπορούν να εμφανιστούν υπό μορφή διαγραμμάτων πάνω στις εικόνες που στέλνει η κάμερα στον υπολογιστή, δίνοντας επιπλέον και μια φυσική αίσθηση και ερμηνεία τους. Αναφέρονται βασικές μέθοδοι ανίχνευσης αντικειμένων, και επιλέγεται η πιο πρόσφορη. Έπειτα παρουσιάζεται ο αλγόριθμος που δημιουργήθηκε για τη συγκεκριμένη εφαρμογή, με βάση την μέθοδο μετασχηματισμού στο πεδίο της συχνότητας (μετασχηματισμός Fourier).

Τέλος παρουσιάζονται τα αποτελέσματα που προέκυψαν από τη πειραματική διαδικασία, τα οποία αξιολογούνται και συγκρίνονται με τα αναμενόμενα. Από τη σύγκριση αυτή προκύπτει και η επιτυχία της λειτουργίας του αλγορίθμου.

Κλείνοντας, με βάση τα ανωτέρω, αναφέρονται τα σημεία στα οποία ο αλγόριθμος επιτυγχάνει και αυτά στα οποία υστερεί. Προτείνονται τρόποι βελτίωσης, του συνόλου της διάταξης, καθώς και μελλοντική εργασία που θα μπορούσε να πραγματοποιηθεί με βάση τα αποτελέσματα της παρούσας.

Abstract

The aim of this thesis is the control as well as the acquisition of the physical variables related to the motion of a robotic fish. The robotic fish has been constructed at the Control Systems Laboratory of the School of Mechanical Engineering of the National Technical University of Athens (NTUA). The thesis is separated into two parts. The first part presents the design and implementation of a moving x-y carriage for the robotic fish tank, while the second part focuses on the computational aspects of following the robotic fish.

In more details, the first part refers to the planning and the implementation of a carriage, in which the robotic fish can be adjusted to. The carriage can move in two-dimensions. With the use of a force sensor, the forces, acting on the fish during the movement, can be measured. In addition camera can be adjusted on the carriage. So when the robotic fish executes a trajectory, the carriage, through an image processing algorithm, will be able to follow its motion. For the design of the carriage, we should take in mind the infrastructure which already exists in the laboratory. For instance, the carriage must be able to be adjusted to the tank and move without the fear of crashing to any other object.

As far as the computational part is concerned, it is related to the study of the physical variables related to the motion that the robotic fish makes; such as the trajectory, the speed and the attitude of the body. These quantities are calculated by computer vision. The fish is located by using an image processing algorithm and all the results are recorded. These results are processed and the desired variables are obtained. The results can take the form of diagrams superimposed on the images that the camera sends to the computer, giving a natural sense. Basic methods of image processing are mentioned and the most efficient one is chosen. The algorithm created for the specific application, based on the transform in the frequency domain (Fourier Transform), is presented.

In the end, the experimental results, are presented and discussed. They are evaluated and compared with the expected ones. This comparison shows the accuracy of the implemented algorithm.

To sum up, according to the above, the advantages and disadvantages of the image processing algorithm are discussed. Ways of improving it are proposed. Future work that can follow the one presented here is outlined.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρ. Ευάγγελο Παπαδόπουλο, για τη στήριξη και την καθοδήγηση που μου παρείχε σε όλη τη διάρκεια της εκπόνησης της παρούσας εργασίας. Επιπλέον όμως θα ήθελα να τον ευχαριστήσω για την ευκαιρία που μου έδωσε να αποκτήσω νέες και πολύ σημαντικές γνώσεις, που μου παρέχουν μεγαλύτερη σιγουριά για τη μελλοντική μου σταδιοδρομία. Ακόμα θα ήθελα να ευχαριστήσω τον Υ.Δ. Μιχάλη Μακροδημήτρη για την εξαιρετική συνεργασία που είχαμε, και για τις πολύτιμες γνώσεις που απέκτησα μέσα από αυτή. Ένα μεγάλο ευχαριστώ πρέπει να πω στην οικογένεια μου, που με υποστηρίζει με κάθε τρόπο, παρέχοντας μου αφειδώς στήριξη όλα τα χρόνια των σπουδών μου. Τέλος θα ήθελα να ευχαριστήσω όλους όσους συνέβαλαν, με τον δικό τους ξεχωριστό τρόπο, εμφανώς ή αφανώς, στην ολοκλήρωση της παρούσας εργασίας.

Σωτήρης Χρ. Ντούσκας

Φθινόπωρο 2011

Περιεχόμενα

Περίληψη.....	2
Abstract.....	3
Περιεχόμενα.....	5
Κατάλογος Σχημάτων.....	7
Κατάλογος Πινάκων.....	9
1 Εισαγωγή.....	10
1.1 Σκοπός Εργασίας.....	10
1.2 Βιβλιογραφική Ανασκόπηση.....	11
1.3 Δομή Εργασίας.....	12
2 Εργαστηριακή Υποδομή.....	14
2.1 Εισαγωγή.....	14
2.2 Ρομποτικό Ψάρι.....	14
2.3 Δεξαμενή.....	18
2.4 Αποσκληρυντής Νερού.....	21
2.5 Φίλτρο Καθαρισμού.....	22
2.6 Συνολική Αξιολόγηση Υποδομής.....	23
3 Κατασκευή Φορείου.....	24
3.1 Εισαγωγή.....	24
3.2 Επιλογή Εξαρτημάτων.....	24
3.2.1 Επιλογή Φορείου.....	24
3.2.2 Επιλογή Μέσου Μετάδοσης Κίνησης.....	31
3.2.3 Επιλογή Κινητήρων.....	36
3.2.4 Επιλογή Τροφοδοτικού.....	41
3.3 Περιγραφή Ηλεκτρονικών Κυκλωμάτων.....	43
3.3.1 Εισαγωγή.....	43
3.3.2 Αναλυτική Περιγραφή.....	43
4 Παρακολούθηση Ρομποτικού Ψαριού Μέσω Κάμερας.....	48
4.1 Εισαγωγή.....	48
4.2 Βασικές Αρχές Ψηφιακών Εικόνων.....	48
4.3 Εικόνα και Εικονοροή στο Λογισμικό MATLAB.....	50
4.3.1 Επιλογή Λογισμικού.....	50
4.3.2 Αναπαράσταση Εικόνων στη MATLAB.....	51
4.4 Επιλογή Αλγόριθμου Εύρεσης και Παρακολούθησης Ρομποτικού Ψαριού.....	52
4.4.1 Εισαγωγή.....	52

4.4.2 Ταυτοποίηση Εικόνων.....	52
4.4.3 Μετασχηματισμός Hough.....	52
4.4.4 Αναγνώριση Αντικειμένων στο Πεδίο της Συχνότητας.....	63
4.4.5 Διαδικασία Επιλογής Αλγόριθμου Επεξεργασίας Εικόνας.....	68
4.5 Περιγραφή Αλγόριθμου Επεξεργασίας Εικόνας.....	69
4.5.1 Αναλυτική Περιγραφή Αλγόριθμου.....	73
5 Μετρήσεις και Πειραματικά Αποτελέσματα.....	97
5.1 Εισαγωγή.....	97
5.2 Μελέτη Κίνησης.....	97
6 Συμπεράσματα και Μελλοντική Εργασία.....	106
6.1 Εισαγωγή.....	106
6.2 Συμπεράσματα.....	106
6.3 Μελλοντική Εργασία.....	108
Βιβλιογραφία.....	110
Παράρτημα Α.....	115
Παράρτημα Β.....	121
Παράρτημα Γ.....	124

Κατάλογος Σχημάτων

2-1: Ρομποτικό Ψάρι με κάλυμμα.....	15
2-2: Το ρομποτικό ψάρι χωρίς το κάλυμμα του.....	15
2-3: Ρομποτικό ψάρι και κυριότερα μέρη του.....	16
2-4: Ρομποτικό ψάρι στη πιο πρόσφατη μορφή του.....	17
2-5: Ρομποτικό ψάρι στη πιο πρόσφατη μορφή του (τυχαία πλάγια όψη).....	17
2-6: Ρομποτικό ψάρι στη πιο πρόσφατη μορφή του (κάτοψη).....	18
2-7: Δεξαμενή (όψη από το εσωτερικό της) και φορείο.....	19
2-8: Δεξαμενή και λοιπές υποστηρικτικές κατασκευές.....	20
2-9: Δεξαμενή με μεταλλικό πλαίσιο και φορείο.....	21
2-10: Αποκληρυντής νερού Autotrol Osmonics 255/460 General Electric.....	22
2-11: Φίλτρο Υπεριώδους ακτινοβολίας.....	23
3-1: Τεμάχια (από Misumi) πριν την συναρμολόγηση.....	26
3-2: Τρισδιάστατο σχέδιο φορείου.....	26
3-3: Συναρμολογημένο φορείο πάνω στη δεξαμενή (πλάγια όψη α').....	27
3-4: Συναρμολογημένο φορείο πάνω στη δεξαμενή (πλάγια όψη β').....	28
3-5: Τρισδιάστατο σχέδιο φορείου πάνω στη δεξαμενή (κάτοψη).....	28
3-6: Τρισδιάστατο σχέδιο φορείου πάνω στη δεξαμενή (όψη από επάνω).....	28
3-7: Μέρος του συναρμολογημένου φορείου ανάμεσα σε άλλα εξαρτήματα.....	29
3-8: Ράβδος αλουμινίου για τη στήριξη κινητήρα και τροχαλίας στον x άξονα.....	29
3-9: Ράβδος αλουμινίου για τη στήριξη τροχαλίας στον x άξονα.....	30
3-10: Ράβδος αλουμινίου για τη στήριξη κινητήρα και τροχαλίας στον x άξονα.....	31
3-11: Ράβδος αλουμινίου για τη στήριξη τροχαλίας στον x άξονα.....	31
3-12: Αλυσίδα 24GCF της Berg.....	33
3-13: Τροχαλία 24B της Berg.....	34
3-14: Στοιχεία μετάδοσης κίνησης της διάταξης και κινητήρες.....	35
3-15: Συναρμολογημένο φορείο και αλυσίδα μετάδοσης κίνησης κατά τον y άξονα.....	35
3-16: Τροφοδοτικό SIEMENS SITOP 24VDC SMART 5A 6EP1-333-2BA01.....	43
3-17: Παλμοί Ρυθμιζόμενου Πλάτους PWM.....	45
3-18: Οι δυο βασικές καταστάσεις της γέφυρας-H.....	45
3-19: Σχηματικό διάγραμμα ηλεκτρονικών κυκλωμάτων ελέγχου.....	47
4-1: Δυναμική ψηφιακή εικόνα.....	49
4-2: Διακριτές αποχρώσεις του γκρι (μονοχρωματικές εικόνες αποχρώσεων του γκρι).....	49
4-3: Χρωματικό σύστημα RGB για απεικόνιση έγχρωμων εικόνων.....	50

4-4: Σύστημα Συντεταγμένων για Εικόνες του Λογισμικού MATLAB.....	51
4-5: (α) Σύνολο δυο συνευθειακών σημείων που ορίζουν μια ευθεία στον καρτεσιανό χώρο.....	54
4-5: (β) Αναπαράσταση της ευθείας στον παραμετρικό χώρο.....	54
4-6: Μετασχηματισμός Hough.....	55
4-7: Μετασχηματισμός Hough σε σύστημα πολικών συντεταγμένων.....	56
4-8: Μετασχηματισμός Hough για κύκλους.....	57
4-9: Έλλειψη με τους δύο άξονες της και τις παραμέτρους αξόνων.....	59
4-10: Γεωμετρικά μεγέθη γενικευμένου μετασχηματισμού Hough.....	61
4-11: Στοιχεία πίνακα ορισμού του στόχου.....	75
4-12: Εικόνα, περιοχή ενδιαφέροντος και εύρεση στόχου (peak).....	78
4-13: Έναρξη διαδικασίας απόκτησης δεδομένων (triggering).....	79
4-14: Σημείο Μέγιστης Ομοιότητας (peak) και Συστήματα Συντεταγμένων.....	84
4-15: Διαδικασία διατήρησης του στόχου κοντά στην επιθυμητή θέση.....	86
4-16: Πραγματικό διάγραμμα των δύο συνιστωσών της ταχύτητας.....	94
4-17: Διάγραμμα δυο ταχυτήτων μετά την χρήση της εντολής smooth.....	95
4-18: Διάγραμμα δυο ταχυτήτων μετά την γραμμική παρεμβολή στα μη μηδενικά στοιχεία.....	96
5-1: Εικόνα αρχικοποίησης πρώτου πειράματος και στόχοι αρχικοποίησης.....	98
5-2: Διάγραμμα τροχιάς ρομποτικού ψαριού στο επίπεδο.....	99
5-3: Τροχιά ρομποτικού ψαριού, πάνω σε πραγματική εικόνα από την κάμερα.....	100
5-4: Τροχιά και των δυο στόχων, πάνω σε πραγματική εικόνα από την κάμερα.....	101
5-5: Γωνία περιστροφής του ρομποτικού ψαριού κατά τη διάρκεια της κίνησης του.....	102
5-6: Διάγραμμα ταχύτητας ρομποτικού ψαριού στο επίπεδο.....	103
5-7: Διάγραμμα ταχύτητας ρομποτικού ψαριού, μετά την χρήση της εντολής smooth.....	104
5-8: Διανύσματα ταχύτητας σε διάφορα σημεία της τροχιάς που κινήθηκε το ρομποτικό ψάρι....	105

Κατάλογος Πινάκων

3-1: Πίνακας Επιλογής του Μέσου Μετάδοσης Κίνησης.....	32
3-2: Χαρακτηριστικά Στοιχεία Αλυσίδων 24GCF.....	33
3-3: Φυσικά μεγέθη για το φορείο κατά τον x άξονα.....	37
3-4: Φυσικά μεγέθη για τις τροχαλίες κατά τον x άξονα.....	37
3-5: Φυσικά μεγέθη για τον κινητήρα κατά τον x άξονα.....	38
3-6: Φυσικά μεγέθη για το φορείο κατά τον y άξονα.....	40
3-7: Φυσικά μεγέθη για τις τροχαλίες κατά τον y άξονα.....	40
3-8: Φυσικά μεγέθη για τον κινητήρα κατά τον y άξονα.....	41
3-9: Πίνακας Επιλογής Τροφοδοτικού.....	42
3-10: Χαρακτηριστικά Τροφοδοτικού.....	42
4-1: Πίνακας R - Γενικευμένος Μετασχηματισμός Hough.....	62
4-2: Πίνακας Επιλογής Μεθόδου Επεξεργασίας Εικόνας.....	69

1 Εισαγωγή

1.1 Σκοπός της εργασίας

Βασικό πρόβλημα στην μελέτη υποβρυχίων οχημάτων, είναι ο εντοπισμός της θέσης τους. Κατά καιρούς έχουν προταθεί διάφοροι τρόποι επίλυσης του συγκεκριμένου προβλήματος. Οι κυριότερες μέθοδοι επίλυσης είναι οι ακουστικές και οι οπτικές. Πολύ συχνά για τη διευκόλυνση της μελέτης, χρησιμοποιείται φορείο, κίνησης σε δυο τουλάχιστον άξονες, το οποίο προσαρμόζεται στη δεξαμενή μελέτης.

Έτσι ο σκοπός της συγκεκριμένης μεταπτυχιακής εργασίας είναι διπλός. Πρώτος στόχος είναι να κατασκευαστεί ένα φορείο, ικανό να εκτελεί κινήσεις σε δυο άξονες. Το φορείο αυτό θα προσαρτηθεί πάνω στην δεξαμενή που βρίσκεται στις εγκαταστάσεις του Εργαστηρίου Αυτόματου Ελέγχου, της Σχολής Μηχανολόγων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου. Στο εσωτερικό της δεξαμενής, μπορεί να κινείται ρομποτικό ψάρι, το οποίο έχει σχεδιασθεί και κατασκευαστεί εξ'ολοκλήρου στο Εργαστήριο. Το φορείο κατασκευάζεται έτσι, ώστε να προσδένεται πάνω του το ρομποτικό ψάρι. Στην συνέχεια, και με τη βοήθεια αισθητήρα δύναμης, μπορούν να μετρώνται οι δυνάμεις που εμφανίζονται στο ρομποτικό ψάρι κατά τη διάρκεια της κίνησης του.

Ο δεύτερος στόχος της εργασίας είναι ο εντοπισμός του ρομποτικού ψαριού και η καταγραφή των φυσικών μεγεθών της κίνησης του, με οπτική μέθοδο, δηλαδή μέσω κάμερας. Τέτοια μεγέθη είναι η τροχιά, η ταχύτητα και η γωνία περιστροφής. Αυτό επιτυγχάνεται και με κατάλληλο αλγόριθμο επεξεργασίας εικόνας, που δημιουργήθηκε για τη συγκεκριμένη εφαρμογή. Ο αλγόριθμος αυτός είναι σε θέση να εντοπίζει και να παρακολουθεί το ρομποτικό ψάρι, σε μια διαδικασία εικονοροής (video). Τα δεδομένα που προκύπτουν, αποθηκεύονται, και στη συνέχεια επεξεργάζονται κατάλληλα, ώστε να αποκτήσουμε τα επιθυμητά φυσικά μεγέθη.

1.2 Βιβλιογραφική Ανασκόπηση

Το βασικό πρόβλημα που πραγματεύεται η εργασία είναι ο εντοπισμός του υποβρύχιου οχήματος, δηλαδή στη προκειμένη περίπτωση, του ρομποτικού ψαριού. Η λύση του συγκεκριμένου προβλήματος είναι “κλειδί” στην εκτέλεση υποβρύχιων διαδικασιών, από μια απλή κίνηση μέχρι πιο σύνθετες ενέργειες.

Υπάρχουν δυο βασικές μέθοδοι εντοπισμού υποβρυχίων οχημάτων, η ακουστική και η οπτική. Στην ακουστική χρησιμοποιούνται κατά κύριο λόγο σόναρ [4, 26, 27, 28, 29], όπου το σώμα εντοπίζεται από το ηχητικό του αποτύπωμα. Δηλαδή μια διάταξη εκπέμπει ένα ηχητικό σήμα, και στη συνέχεια λαμβάνεται η ανάκλαση του ήχου πάνω στο σώμα. Συχνά χρησιμοποιούνται και πιο σύνθετες διατάξεις, όπως ακουστικές κάμερες [27, 28]. Οι ακουστικές κάμερες, στην ουσία λειτουργούν όπως τα σόναρ, δίνοντας όμως επιπλέον, τη δυνατότητα αναπαραγωγής του σχήματος του σώματος, πάντα με βάση την ανάκλαση του ήχου, πάνω σε αυτό. Και στις δυο περιπτώσεις η λειτουργία μπορεί να είναι είτε ενεργητική [4, 27, 28, 29] είτε παθητική [26]. Δηλαδή, μπορεί ο ήχος να εκπέμπεται, από ή προς, το υπό εντοπισμό σώμα. Η ακουστική μέθοδος, έχει το πλεονέκτημα, ότι μπορεί να λειτουργεί και σε μεγάλα βάθη. Όμως εμφανίζει προβλήματα θορύβου από κινήσεις μέσα στο νερό. Επιπλέον απαιτείται συγκεκριμένος προσανατολισμός της ακουστικής διάταξης, καθώς αν τοποθετηθεί πίσω από το ψάρι, δεν δίνει αξιόπιστες μετρήσεις, λόγω της συνεχούς κίνησης του ουραίου πτερυγίου. Ένα ακόμα μειονέκτημα της μεθόδου, είναι τα προβλήματα που παρουσιάζονται, όταν αυτή χρησιμοποιείται σε μικρούς χώρους, λόγω των ανακλάσεων του ήχου. Τέλος η μικρή εμπειρία σε θέματα ακουστικής, μας έκανε να το απορρίψουμε, σαν μέθοδο εντοπισμού.

Η δεύτερη μέθοδος είναι η οπτική [3, 29]. Σε αυτή χρησιμοποιείται κάμερα, η οποία παρακολουθεί το ρομποτικό ψάρι κατά την κίνηση του. Για να είναι η διαδικασία αυτοματοποιημένη, απαιτείται ένας αλγόριθμος επεξεργασίας εικόνας. Ο αλγόριθμος θα πρέπει να είναι σε θέση να εντοπίζει το όχημα σε κάθε εικόνα, και να το παρακολουθεί. Δηλαδή θα πρέπει να καταγράφει τη θέση του ψαριού ως προς ένα σύστημα αναφοράς, και να παρέχει στοιχεία για την θέση, την τροχιά του κ.α. Η μέθοδος αυτή υστερεί σε μεγάλα βάθη. Όμως είναι απλούστερη, απαιτεί πιο συμβατικό εξοπλισμό (π.χ. Web-camera), και παρέχει αξιόπιστα αποτελέσματα.

Τέλος υπάρχουν και άλλες μέθοδοι, που χρησιμοποιούνται σπανιότερα, όπως ο εντοπισμός μέσω ηλεκτρικού πεδίου [20]. Η μέθοδος αυτή χρησιμοποιείται συχνά στη φύση από πολλά ψάρια. Στην ουσία ένα ηλεκτρικό σήμα εκπέμπεται σχηματίζοντας ηλεκτρικό πεδίο, και επιτρέποντας με αυτό τον τρόπο να εντοπίζεται το υποβρύχιο όχημα. Η μέθοδος αυτή βρίσκεται σε πρώιμο ακόμα στάδιο, και παρά τα πρώτα ενθαρρυντικά αποτελέσματα χρήζει περαιτέρω βελτίωσης.

Συχνά, όπως και στην παρούσα εργασία, για την μελέτη της κίνησης του ρομποτικού ψαριού, χρησιμοποιείται φορείο, πάνω στο οποίο προσδένεται το όχημα. Στη βιβλιογραφία υπάρχουν πολλές εφαρμογές με κινούμενα φορεία. Σε αρκετές μελετάται μόνο η αλληλεπίδραση

ενός πτερυγίου με το νερό [1, 10, 11, 12]. Ακόμα υπάρχουν αρκετές εργασίες οι οποίες μελετούν την αλληλεπίδραση περισσότερων πτερυγίων [9, 11]. Σημαντική είναι για παράδειγμα η εργασία της ομάδας του βιολόγου G.Lauder [11], στην οποία μελετάται με μεγάλη ακρίβεια η αλληλεπίδραση δυο πτερυγίων, και εξετάζεται η ροή του υγρού γύρω από αυτά. Ωστόσο οι μελέτες αυτές υστερούν, αφού δεν εξετάζουν την αλληλεπίδραση του σώματος του ρομποτικού ψαριού με το ουραίο πτερύγιο.

Για το λόγο αυτό, έχουν γίνει πολλές εργασίες, όπου ολόκληρο το ρομποτικό ψάρι προσδένεται σε φορείο, και εξετάζεται η κίνηση του [2, 22, 23, 25, 30]. Μελετάται ακόμα και η περίπτωση ρομποτικού ψαριού με πλευρικά πτερύγια [8]. Χαρακτηριστική εργασία μελέτης ολόκληρου του σώματος, είναι αυτή της ομάδας των Barret, Triantafyllou. Στη συγκεκριμένη εργασία το ρομποτικό ψάρι, προδένεται πάνω σε φορείο, και με τη βοήθεια αισθητήρα δύναμης καταγράφονται δυνάμεις, υδροδυναμικά μεγέθη, και η ροή του ρευστού γύρω από το ψάρι κατά τη διάρκεια της κίνησης του. Η πιο ολοκληρωμένη εργασία κρίνεται αυτή της ομάδας του Tan [22]. Σε αυτή αφού μελετώνται οι παράμετροι της κίνησης ρομποτικού ψαριού, καλείται αυτό να εκτελέσει κυκλικές κινήσεις.

Στην εργασία γίνεται μια πολύ σύντομη αναφορά στη θεωρία που σχετίζεται με την επεξεργασία εικόνας [17, 39]. Στη συνέχεια, γίνεται μια σύντομη περιγραφή σε θεωρητικό επίπεδο, διαφόρων μεθόδων αναγνώρισης αντικειμένων, με βάση το βιβλίο των Nixon και Aguado [14]. Το συγκεκριμένο βιβλίο, κρίθηκε ως πολύ χρήσιμο και κατατοπιστικό στο θέμα της επεξεργασίας εικόνας, και προσφέρει το απαραίτητο θεωρητικό υπόβαθρο. Δεδομένου ότι ο αλγόριθμος, δημιουργήθηκε σε περιβάλλον MATLAB, εξετάζεται το θέμα της επεξεργασίας εικόνας, στο συγκεκριμένο περιβάλλον [6, 7]. Ακόμα πολλά παραδείγματα σχετικά, υπάρχουν στο Help της MATLAB, το οποίο είναι διαθέσιμο είτε στο ίδιο το περιβάλλον, είτε και στο διαδίκτυο [42, 43, 44, 46]. Τέλος, στο Help της MATLAB υπάρχουν ενδιαφέροντα παραδείγματα, που χρησιμοποιούν την ίδια μέθοδο αναγνώρισης αντικειμένων, με αυτήν που επιλέχθηκε στη συγκεκριμένη εργασία. Τα παραδείγματα αυτά, είναι πολύ κατατοπιστικά για την λειτουργία της μεθόδου και ενός αντίστοιχου αλγορίθμου.

1.3 Δομή της Εργασίας

Η εργασία, αποτελείται από έξι κεφάλαια. Ακολουθεί μια σύντομη περιγραφή του περιεχομένου κάθε κεφαλαίου.

Στο Δεύτερο Κεφάλαιο, αναφέρεται η υποδομή που υπήρχε στο εργαστήριο, πριν την έναρξη της εργασίας. Μέσα από την αναλυτική περιγραφή γίνονται εμφανείς οι ανάγκες που έπρεπε να καλύψει η κάθε κατασκευή.

Στο Τρίτο Κεφάλαιο, περιγράφεται ο σχεδιασμός και η κατασκευή του φορείου. Τα υποκεφάλαια πραγματεύονται την επιλογή των εξαρτημάτων του φορείου, του μέσου μετάδοσης κίνησης, των

κινητήρων και του τροφοδοτικού. Τέλος, περιγράφονται τα ηλεκτρονικά κυκλώματα που σχεδιάστηκαν για τον έλεγχο της κίνησης του φορείου, μέσω ηλεκτρονικού υπολογιστή.

Το Τέταρτο Κεφάλαιο, ασχολείται με την επεξεργασία εικόνων και εικονοροής. Γίνεται μια σύντομη αναφορά στην θεωρία των ψηφιακών εικόνων. Στη συνέχεια γίνεται επιλογή του λογισμικού επεξεργασίας εικόνων. Ακολουθεί η περιγραφή της αναπαράστασης εικόνων στο περιβάλλον MATLAB. Αναφέρονται συνοπτικά οι βασικότερες μέθοδοι επεξεργασίας εικόνας που αφορούν τον εντοπισμό αντικειμένων, και επιλέγεται η καταλληλότερη για την συγκεκριμένη εφαρμογή. Το κεφάλαιο κλείνει με την περιγραφή του αλγορίθμου που υλοποιεί την διαδικασία αυτή, με αναλυτικές αναφορές στον κώδικα που γράφτηκε.

Στο Πέμπτο Κεφάλαιο, παρουσιάζονται τα αποτελέσματα των πειραμάτων, που πραγματοποιήθηκαν. Γίνεται σύγκριση των επιθυμητών/αναμενόμενων αποτελεσμάτων, με αυτά που προέκυψαν από τα πειράματα. Έτσι επιβεβαιώνεται η επιτυχής λειτουργία όλης της διάταξης, και εντοπίζονται ενδεχόμενες αδυναμίες.

Στο Έκτο Κεφάλαιο, που είναι και το τελευταίο, γίνεται η αξιολόγηση στο σύνολο της εργασίας και αναφέρονται τα συμπεράσματα που προέκυψαν. Τέλος γίνονται και προτάσεις για μελλοντική εργασία, ώστε να βελτιωθεί η λειτουργία της διάταξης, και να ξεπεραστούν τα όποια προβλήματα και αδυναμίες έχουν διαπιστωθεί.

2 Εργαστηριακή Υποδομή

2.1 Εισαγωγή

Στο κεφάλαιο αυτό, περιγράφεται η υποδομή του Εργαστηρίου. Η υποδομή αυτή ήταν πολύ σημαντική και δαπανηρή. Για το λόγο αυτό, έπρεπε όλη η κατασκευή να σχεδιασθεί με τέτοιο τρόπο, ώστε να γίνει πλήρης εκμετάλλευσή της. Φυσικά, το σημαντικότερο στοιχείο της ήταν το ρομποτικό ψάρι. Επιπλέον, στα βασικά της στοιχεία, αποτελούνταν από τη δεξαμενή, το φίλτρο καθαρισμού με υπεριώδη ακτινοβολία και τον αποσκληρυντή νερού. Οποιοσδήποτε σχεδιασμός έπρεπε να γίνει με βάση αυτές τις κατασκευές, στις οποίες έπρεπε να προσαρμόζεται πλήρως. Στη συνέχεια θα αναφερθούμε αναλυτικά, σε κάθε ένα από τα στοιχεία αυτά, ώστε να γίνει πιο κατανοητός ο σχεδιασμός της κατασκευής και οι ανάγκες που έπρεπε να καλυφθούν.

2.2 Ρομποτικό Ψάρι

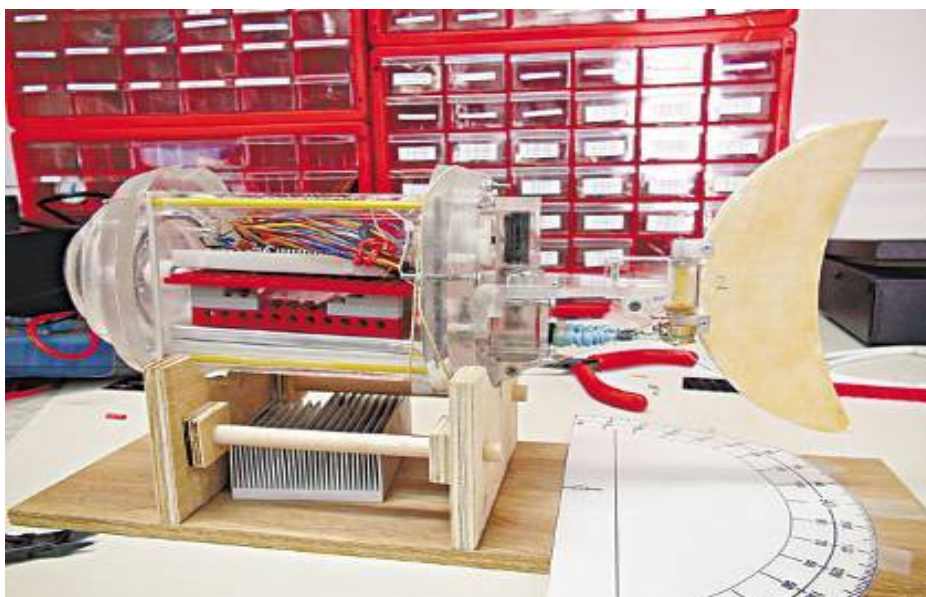
Η κατασκευή του ρομποτικού ψαριού, είχε ολοκληρωθεί σε σχέση με την εξωτερική του μορφή. Κάποιες μικρές παρεμβάσεις έλλειπαν, ώστε να γίνει πλήρως λειτουργικό. Ο σχεδιασμός του, είχε γίνει από τον Υ.Δ. Μιχάλη Μακροδημήτρη και η κατασκευή του έγινε με τη βοήθεια του κ.Λ.Μπαχά. Ο σκελετός του ρομποτικού ψαριού, είναι κατασκευασμένος από plexiglass. Στο πίσω μέρος είναι τοποθετημένος ο κινητήρας, ο οποίος με τη βοήθεια τροχαλίας, μεταδίδει την κίνηση στο ουραίο πτερύγιο. Ουραία πτερύγια κατασκευάστηκαν από ξύλο, από πεπιεσμένο χαρτί και από φύλλα αδιάβροχου χαρτονιού. Η τελευταία λύση προκρίθηκε, καθώς εξασφαλίζει ευκαμψία στην ουρά, κάνοντας έτσι πιο αποδοτικό τον μηχανισμό κίνησης. Επιπλέον κατασκευάστηκε και πλαστικό κάλυμμα του ψαριού, ώστε να δίνει μια πιο “αληθοφανή” και υδροδυναμική μορφή. Ωστόσο, δεν χρησιμοποιήθηκε, αφού ήταν πολύ συχνές οι αλλαγές και οι τροποποιήσεις που έπρεπε να γίνουν στο ψάρι. Η τοποθέτηση του καλύμματος θα έκανε πιο δύσκολες τις συχνές αυτές παρεμβάσεις. Έτσι η χρήση του, θα γίνει πιο συστηματικά μελλοντικά, όπου οι παρεμβάσεις στο ψάρι θα είναι πιο σπάνιες. Το συνολικό μήκος του ψαριού είναι 30cm.

Το Σχήμα 2-1 δείχνει το ψάρι με το κάλυμμα του. Όπως φαίνεται το κάλυμμα του ψαριού, είναι κίτρινου χρώματος και ελλειψοειδούς εκ περιστροφής σχήματος, ώστε να μοιάζει όσο το δυνατόν περισσότερο σε ένα αληθινό ψάρι.



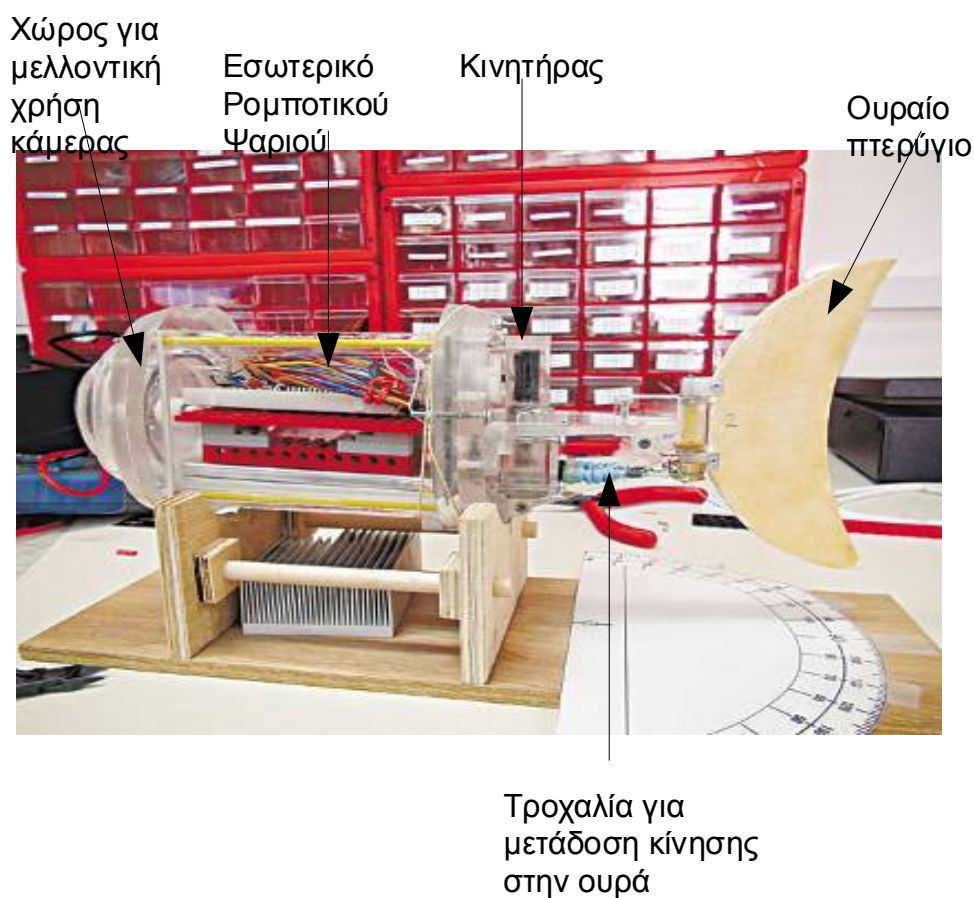
Σχήμα 2-1: Ρομποτικό Ψάρι με κάλυμμα.

Στο Σχήμα 2-2 φαίνεται το ρομποτικό ψάρι χωρίς το κάλυμμα του. Σε αυτή τη μορφή έγινε όλη η διαδικασία ρύθμισης και ελέγχου του ψαριού, καθώς και όλη η πειραματική διαδικασία.



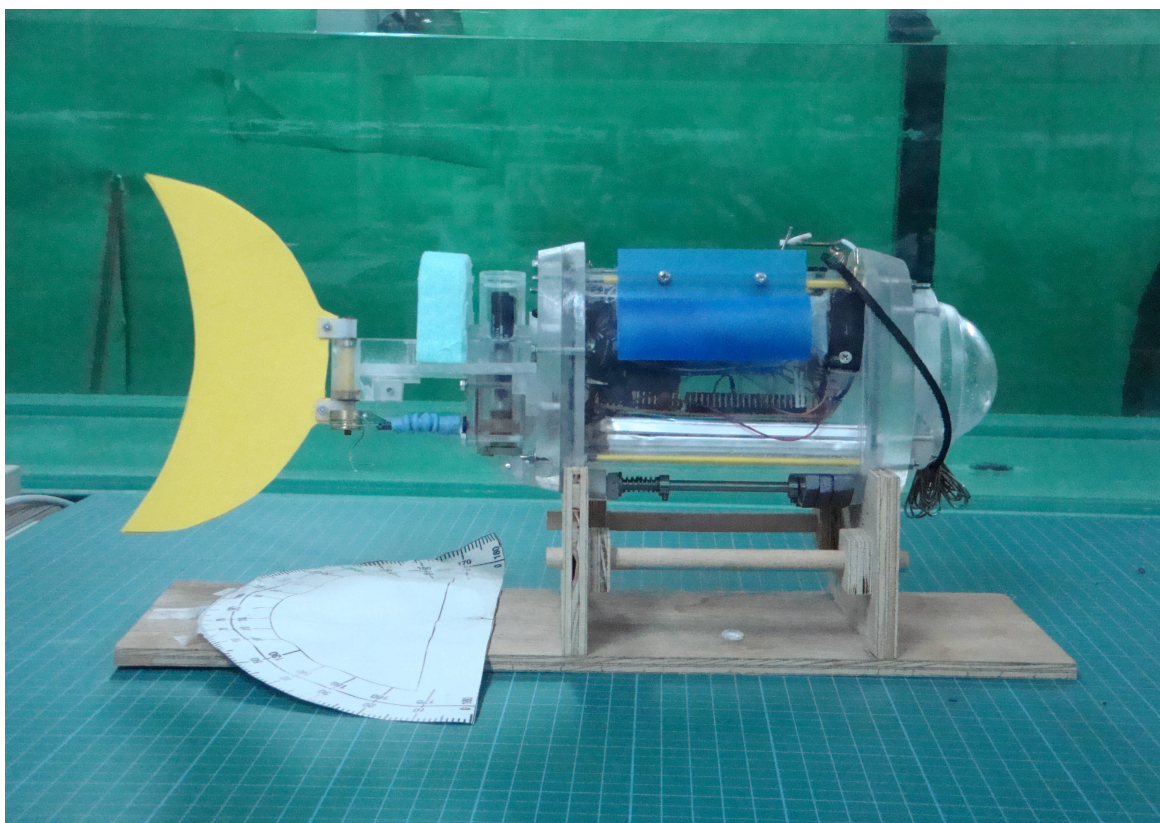
Σχήμα 2-2: Το ρομποτικό ψάρι χωρίς το κάλυμμα του.

Όπως φαίνεται στο Σχήμα 2-2, χωρίς το κάλυμμα, η πρόσβαση στο εσωτερικό του ψαριού, μπορεί να γίνει πολύ εύκολα και άμεσα. Στο Σχήμα 2-3 παρουσιάζονται τα σημαντικότερα κατασκευαστικά στοιχεία του ρομποτικού ψαριού.

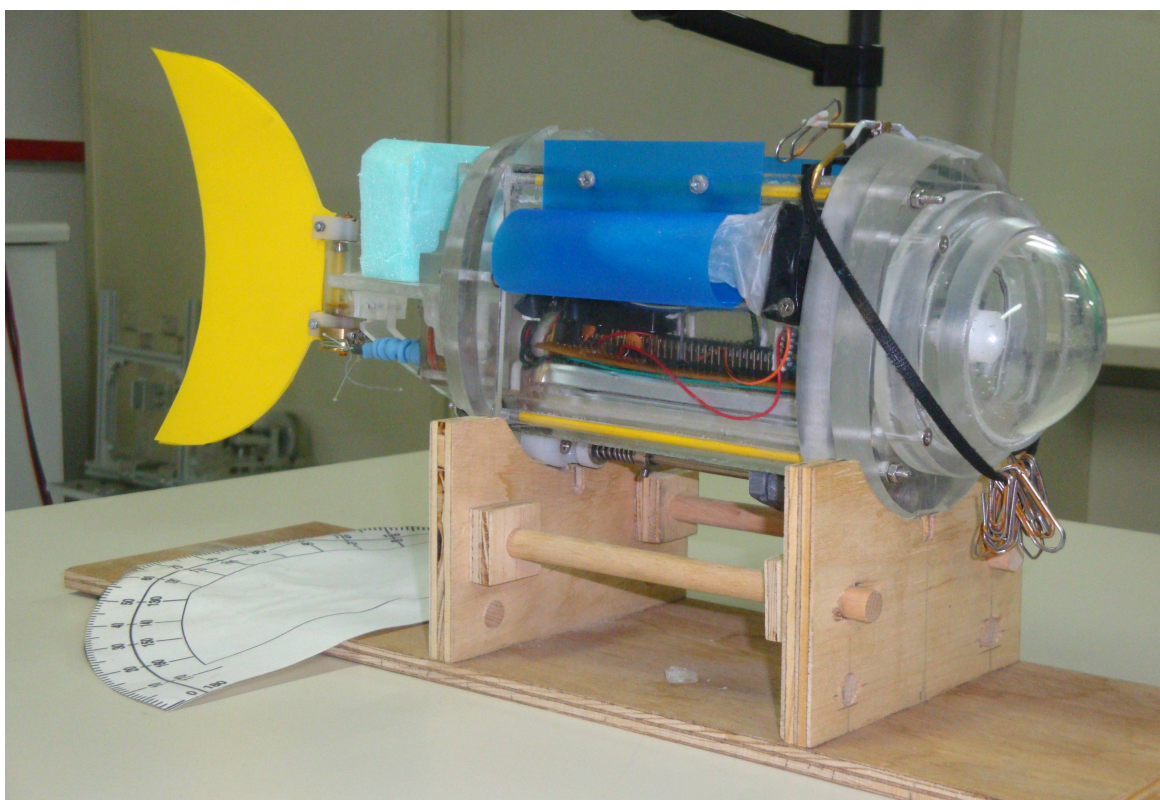


Σχήμα 2-3: Ρομποτικό ψάρι και κυριότερα μέρη του.

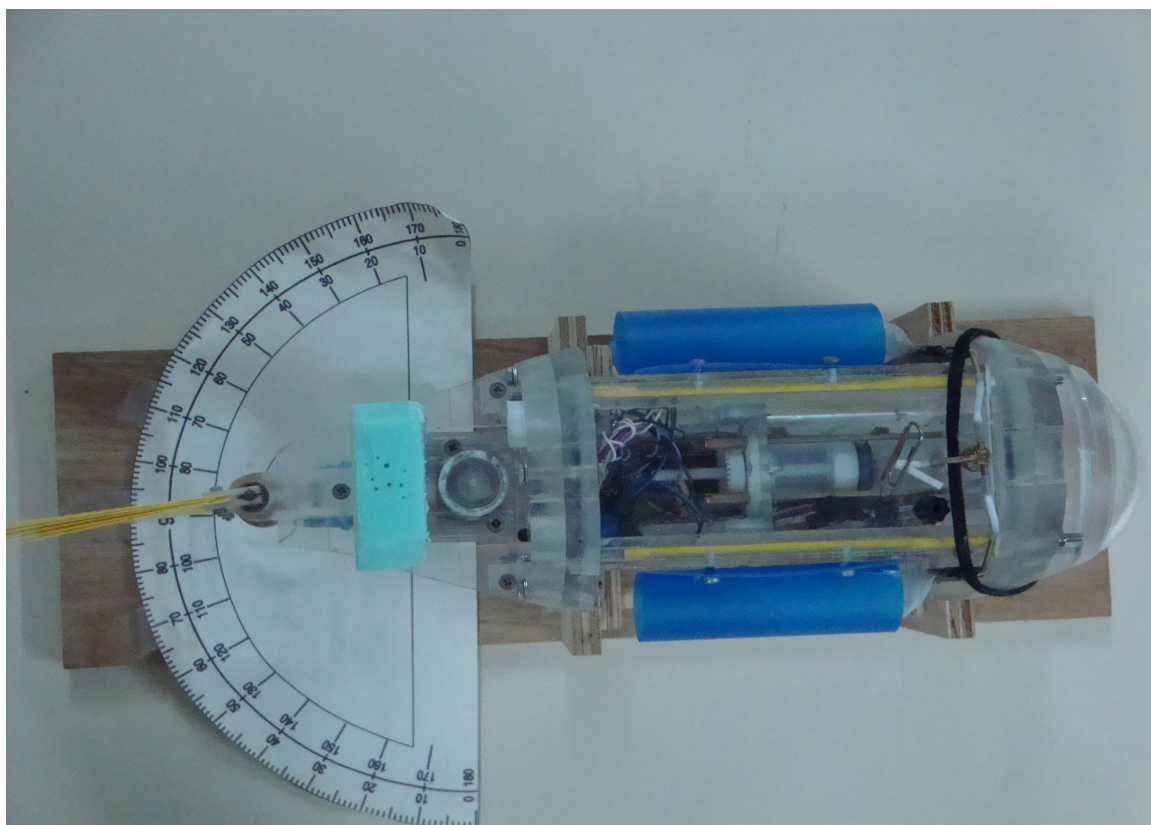
Στο κάτω μέρος του ψαριού, τοποθετήθηκε ερμάριο μεταβλητού βάρους, ώστε να επιτευχθεί και έλεγχος βάθους. Για το σκοπό αυτό, τοποθετήθηκε αρχικά στο πάνω μέρος του ψαριού, αεροφυλάκιο. Έτσι προσθέτοντας ή αφαιρώντας αέρα θα μπορούσαμε να ελέγχουμε το βάθος στην κίνηση του ψαριού. Επειδή όμως στο πάνω μέρος του ψαριού έπρεπε να τοποθετηθεί και στοιχείο πρόσδεσης στο υπό κατασκευή φορείο, ο αρχικός σχεδιασμός τροποποιήθηκε ελαφρώς. Έτσι, οι δεξαμενές αέρα έγιναν δυο και τοποθετήθηκαν στις δυο πλευρές του ψαριού. Επιπλέον, για να γίνει αυτόματη η ρύθμιση του βάθους, τοποθετήθηκε στο εσωτερικό του ψαριού, έμβολο και σερβομηχανισμός, ώστε να εκτοπίζεται ο αέρας από και προς το εσωτερικό του ψαριού. Ο έλεγχος του σερβομηχανισμού γίνεται με την βοήθεια μικροεπεξεργαστή. Με τον τρόπο αυτό επιτυγχάνεται ο έλεγχος βάθους του ρομποτικού ψαριού. Επιπλέον απελευθερώθηκε το πάνω μέρος του ψαριού. Έγινε έτσι εφικτή η πρόσδεση του, στο υπό σχεδιασμό φορείο. Όλες αυτές οι τροποποιήσεις φαίνονται στα Σχήματα 2-4 έως 2-6 που ακολουθούν.



Σχήμα 2-4: Ρομποτικό ψάρι στη πιο πρόσφατη μορφή του.



Σχήμα 2-5: Ρομποτικό ψάρι στη πιο πρόσφατη μορφή του (τυχαία πλάγια όψη).



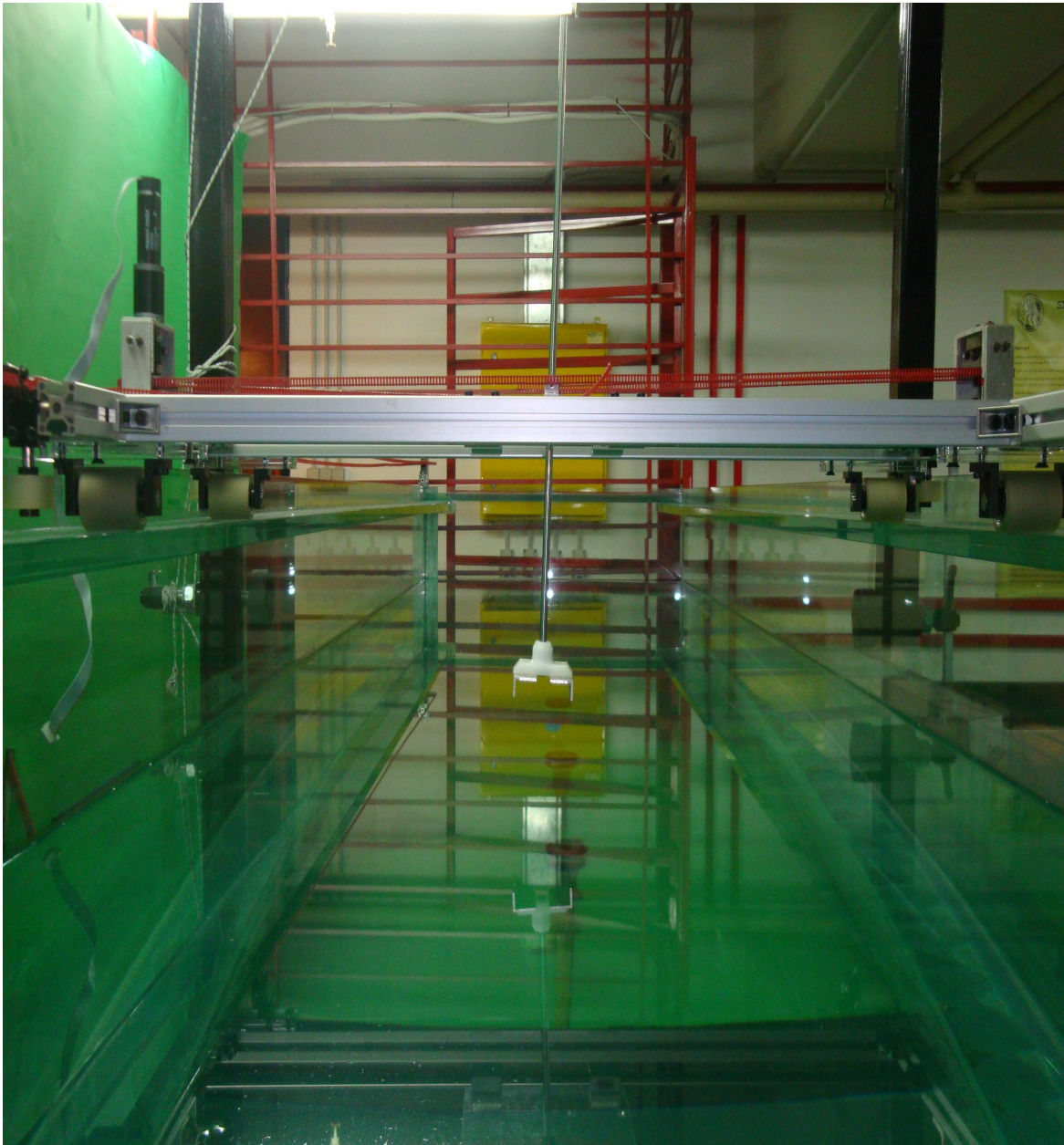
Σχήμα 2-6: Ρομποτικό ψάρι στη πιο πρόσφατη μορφή του (κάτοψη).

2.3 Δεξαμενή

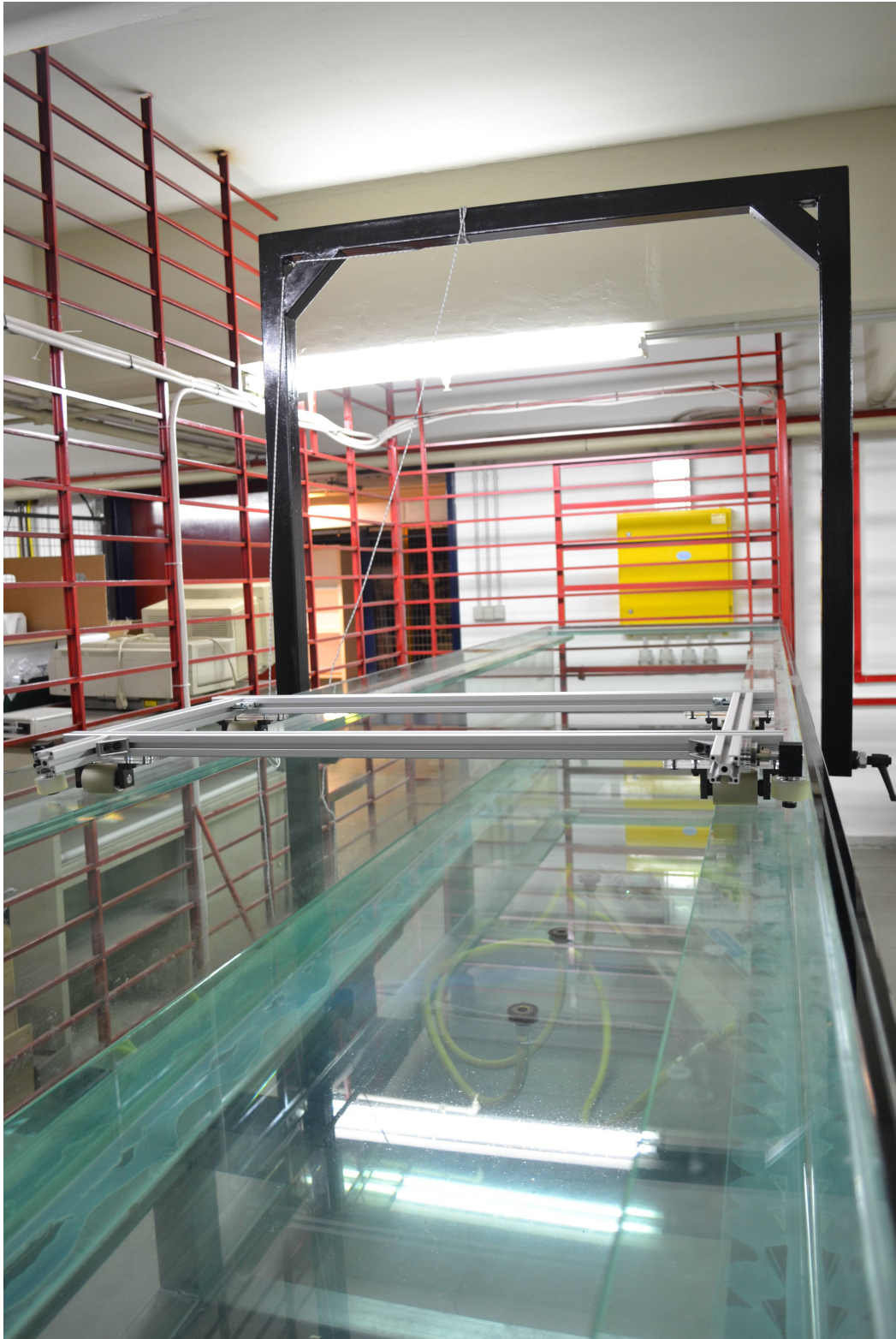
Ένα πολύ βασικό στοιχείο όλης της υποδομής, είναι η δεξαμενή. Είναι ο φυσικός χώρος κίνησης και μελέτης του ρομποτικού ψαριού. Η δεξαμενή που σχεδιάστηκε και κατασκευάστηκε για το σκοπό αυτό, βρίσκεται στο υπόγειο του κτιρίου Μ στη σχολή Μηχανολόγων Μηχανικών του Ε.Μ.Π., και συγκεκριμένα στο Εργαστήριο Αυτόματου Ελέγχου.

Οι διαστάσεις της δεξαμενής είναι $5 \times 1 \times 1$. Είναι κατασκευασμένη από ορθογώνια κομμάτια γυαλιού, πάχους 1cm, μήκους και πλάτους ανάλογων με την θέση του κάθε τμήματος. Τα κομμάτια αυτά συγκρατώνται μεταξύ τους με κόλλα σιλικόνης. Στο κάτω μέρος της, υπάρχει μεταλλικός σκελετός, πάνω στον οποίο τοποθετείται η δεξαμενή. Ένας περιορισμός που είχε τεθεί στο σχεδιασμό της δεξαμενής, ήταν να μην υπάρχουν μεταλλικά πλευρικά στηρίγματα. Όμως αυτό δημιουργούσε πρόβλημα στα πλευρικά τοιχώματα της δεξαμενής. Για το λόγο αυτό, τα γυάλινα πλευρικά τοιχώματα της δεξαμενής είναι διπλά. Δηλαδή αποτελούνται από δύο κομμάτια γυαλιού, πάχους 1cm το καθένα. Επιπλέον, τοποθετήθηκε μαύρο μεταλλικό πλαίσιο στο μέσο της δεξαμενής. Έτσι αποφεύγεται η εμφάνιση βαρελοειδούς μορφής της δεξαμενής σε βάθος χρόνου, λόγω της μεγάλης τάσης που ασκεί το νερό στα τοιχώματα. Ακόμα στο πάνω μέρος και εσωτερικά της δεξαμενής, υπάρχει κομμάτι γυαλιού, το οποίο σχηματίζει έναν διάδρομο περιμετρικά της. Σε αυτό το κομμάτι γυαλιού πατάει και το φορείο για να μπορεί να εκτελεί τις κινήσεις του.

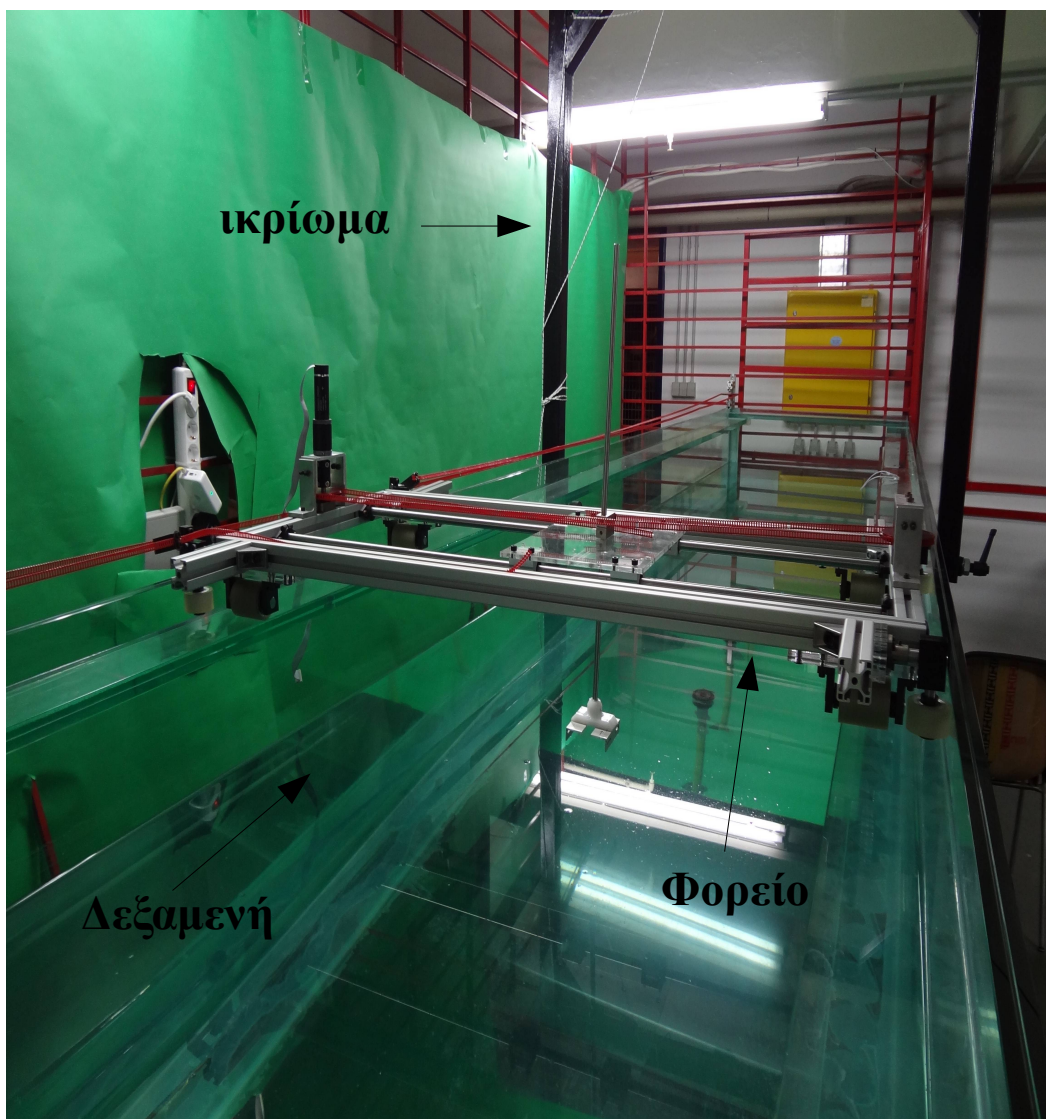
Στα Σχήματα 2-7 έως 2-9 που ακολουθούν, φαίνεται η δεξαμενή. Ακόμα διακρίνεται κάτω από το νερό, ο μεταλλικός σκελετός, ενώ πιο καθαρά διακρίνεται γύρω από αυτήν, και το μαύρο μεταλλικό πλαίσιο (ικρίωμα) στο μέσο της.



Σχήμα 2-7: Δεξαμενή (όψη από το εσωτερικό της) και φορείο.



Σχήμα 2-8: Δεξαμενή και λοιπές υποστηρικτικές κατασκευές.



Σχήμα 2-9: Δεξαμενή με μεταλλικό πλαίσιο και φορείο.

2.4 Αποσκληρυντής Νερού

Για το γέμισμα της δεξαμενής απαιτείται αρκετά μεγάλη ποσότητα νερού. Για παράδειγμα αν θελήσουμε να γεμίσουμε 80cm, τότε απαιτούνται 4000lt νερού. Η ποσότητα αυτή είναι αρκετά μεγάλη και δαπανηρή, για να προμηθευτείται από το δίκτυο της ΕΥΔΑΠ. Επιπλέον, το Ε.Μ.Πολυτεχνείο διαθέτει δικιά του γεώτρηση παροχής νερού. Όμως, το νερό από αυτήν την γεώτρηση εμφανίζει πολύ υψηλή περιεκτικότητα σε ασβέστιο. Αν χρησιμοποιούσαμε το νερό της γεώτρησης, στη δεξαμενή, χωρίς επεξεργασία, τότε πολύ σύντομα, δεν θα μπορούσαμε να δούμε από τα πλάγια το νερό. Κι αυτό, γιατί τα άλατα που θα κατακάθονταν στα τοιχώματα της δεξαμενής, θα σχημάτιζαν με τον καιρό, ένα στρώμα αρκετά μεγάλου πάχους, ώστε να είναι αδύνατη οποιαδήποτε οπτική επαφή με το εσωτερικό.

Για να αποφευχθεί αυτό το φαινόμενο, αποφασίστηκε να γίνει επεξεργασία του νερού της

γεώτρησης, με την βοήθεια αποσκληρυντή. Ο αποσκληρυντής κατακρατά τα ιόντα ασβεστίου, και έτσι το νερό αποσκληρύνεται. Το μοντέλο που επιλέχθηκε είναι το Autotrol Osmonics 255/460 της General Electric, και φαίνεται στο Σχήμα που ακολουθεί.



Σχήμα 2-10: Αποκληρυντής νερού Autotrol Osmonics 255/460 General Electric.

Η λειτουργία του κυκλώματος του νερού έχει ως εξής: το νερό της γεώτρησης, εισέρχεται από τη βρύση στο κύκλωμα. Περνάει μέσα από τον αποσκληρυντή, και απομακρύνονται τα άλατα. Στη συνέχεια καταλήγει καθαρό πλέον στη δεξαμενή.

2.5 Φίλτρο Καθαρισμού

Για να αποφευχθεί, η ανάπτυξη μικροοργανισμών στο νερό της δεξαμενής, χρησιμοποιήθηκε και φίλτρο καθαρισμού. Το νερό της δεξαμενής, εισέρχεται μέσα στο φίλτρο, το οποίο με τη βοήθεια υπεριώδους ακτινοβολίας, καθαρίζει το νερό σκοτώνοντας όλους τους μικροοργανισμούς που ενδεχομένως έχουν αναπτυχθεί μέσα σε αυτό. Στη συνέχεια το νερό, καθαρό πλέον με την βοήθεια αντλίας, επανεισέρχεται μέσα στη δεξαμενή. Στο Σχήμα 2-11 φαίνεται το φίλτρο υπεριώδους ακτινοβολίας που χρησιμοποιήθηκε.



Σχήμα 2- 11: Φίλτρο Υπεριώδους Ακτινοβολίας.

2.6 Συνολική αξιολόγηση υποδομής

Όπως γίνεται αντιληπτό, στο εργαστήριο υπήρχε μια πάρα πολύ σημαντική - από άποψη τεχνολογίας, κατασκευής και δαπάνης - υποδομή, για να εξυπηρετήσει τις ανάγκες της συγκεκριμένης έρευνας. Έτσι ήταν επιτακτική η ανάγκη, όλη αυτή η υποδομή να αξιοποιηθεί στο έπακρο.

Ενδεικτικά μπορεί να αναφερθεί ότι στο σχεδιασμό του φορείου, έπρεπε να συνυπολογιστεί η μορφή του “χείλους” περιμετρικά της δεξαμενής, αφού αυτό καθορίζει το πού και το πώς θα κινείται το φορείο. Η κίνηση του φορείου πρέπει να γίνεται με τέτοιο τρόπο, ώστε να εκμηδενίζεται η πιθανότητα πρόσκρουσης με οποιοδήποτε αντικείμενο στον περιβάλλοντα χώρο ή κάποια υποστηρικτική κατασκευή και πολύ περισσότερο με άνθρωπο που μπορεί να βρίσκεται κοντά σε αυτό. Ακόμα σημαντικό ρόλο στην επιλογή εξαρτημάτων έπαιξε η μορφή του ψαριού, αφού αυτή καθόρισε τον τρόπο πρόσδεσης του στο φορείο.

3 Κατασκευή Φορείου

3.1 Εισαγωγή

Το τρίτο κεφάλαιο της εργασίας, καλύπτει το σχεδιασμό και την κατασκευή του φορείου. Η κατασκευή αυτή έχει πολλαπλές χρήσεις. Μπορεί το υποβρύχιο ρομπότ να προσδένεται πάνω στο φορείο και να κινείται με την βοήθεια του. Με τον τρόπο αυτό μπορεί να γίνει μέτρηση των δυνάμεων που ασκούνται στο ψάρι κατά την κίνηση του. Επιπλέον όμως, το φορείο και το ρομπότ, μπορούν να κινούνται ανεξάρτητα. Έτσι, με την τοποθέτηση κάμερας στο φορείο, μπορεί να υλοποιηθεί οπτικός έλεγχος της κίνησης, του υποβρυχίου ρομπότ.

Με βάση αυτά, το φορείο θα πρέπει να εκτελεί κινήσεις σε δυο διαστάσεις (2-D), ενώ έχει ληφθεί πρόβλεψη, και για μελλοντική προσθήκη ενός επιπλέον άξονα, για έλεγχο βάθους. Η κίνηση κατά τον έναν άξονα γίνεται πάνω στα “χείλη” της δεξαμενής, στην οποία βρίσκεται το ψάρι. Στο δεύτερο άξονα, η κίνηση γίνεται πάνω στο ίδιο το φορείο.

Η κίνηση επιτυγχάνεται με την βοήθεια ηλεκτρικών σερβοκινητήρων συνεχούς ρεύματος (DC motor), ενώ ο έλεγχος της κίνησης, γίνεται είτε ενσύρματα, είτε και ασύρματα από ηλεκτρονικό υπολογιστή.

3.2 Σχεδιασμός Φορείου

Το φορείο επιλέχθηκε να σχεδιαστεί από μηδενική βάση. Η επιλογή αυτή έγινε ώστε το τελικό προϊόν να μπορεί να προσαρμοστεί, με ακρίβεια στη δεξαμενή. Επιπλέον, μπορούμε να γνωρίζουμε με λεπτομέρεια, όλες τις παραμέτρους κίνησης του φορείου. Τέλος, δίνεται η δυνατότητα της εναλλαξιμότητας των εξαρτημάτων. Δηλαδή, σε περίπτωση που κάποιο εξάρτημα χαλάσει, μπορούμε άμεσα να το εντοπίσουμε και να το αντικαταστήσουμε. Στη συνέχεια προχωρήσαμε στην απαραίτητη έρευνα αγοράς, ώστε να βρούμε την εταιρία που θα μας προμήθευε τα απαραίτητα εξαρτήματα, για να υλοποιηθεί ο αρχικός σχεδιασμός του φορείου.

Η γενική περιγραφή του σχεδιασμού του φορείου, είναι ένα ορθογώνιο πλαίσιο

αποτελούμενο από τέσσερα προφίλ αλουμινίου. Στο εσωτερικό του πλαισίου, τέσσερα βαγονάκια ανά ζεύγη, κινούνται πάνω σε δυο άξονες, παρέχοντας κίνηση κατά τον y άξονα. Τα τέσσερα βαγονάκια είναι τοποθετημένα έτσι ώστε, πάνω τους να προσδένεται ένα τεμάχιο plexiglass ανάλογων διαστάσεων. Πάνω σε αυτό το τεμάχιο, προσδένονται, διάφορα εξαρτήματα, όπως αισθητήρας δύναμης, πλακέτες, κινητήρας, ράβδος πρόσδεσης του ψαριού, κ.α.

Στη συνέχεια θα γίνει επεξήγηση της διαδικασίας επιλογής των διαφόρων εξαρτημάτων, που επιλέχθηκαν για τη διάταξη. Υπάρχει ένας μεγάλος αριθμός εξαρτημάτων που θα πρέπει να επιλεγούν, ώστε να καλυφθούν οι ανάγκες της συγκεκριμένης διάταξης. Οι κινητήρες που θα παρέχουν κίνηση, τα υλικά του φορείου, και των υπόλοιπων σταθερών τμημάτων της διάταξης, οι συνδέσεις αυτών, τα μέσα μετάδοσης κίνησης κ.α. Πέραν των κλασσικών υπολογισμών (αντοχής, ηλεκτρολογικών κ.α.) θα πρέπει να ληφθούν υπόψη και άλλοι παράγοντες στην επιλογή, όπως η ύπαρξη νερού, ο συγκεκριμένος χώρος της δεξαμενής, καθώς και το μεταλλικό πλαίσιο γύρω από από τη δεξαμενή κ.α.

3.2.1 Επιλογή φορείου

Το φορείο, αποτελεί το βασικό στοιχείο της διάταξης. Μέσω αυτού επιτυγχάνεται η κίνηση και στους δυο άξονες, και πάνω σε αυτό προσδένεται το υποβρύχιο ρομπότ, καθώς και άλλα εξαρτήματα, όπως αισθητήρας δύναμης, κινητήρας, ηλεκτρονικές πλακέτες κ.α.

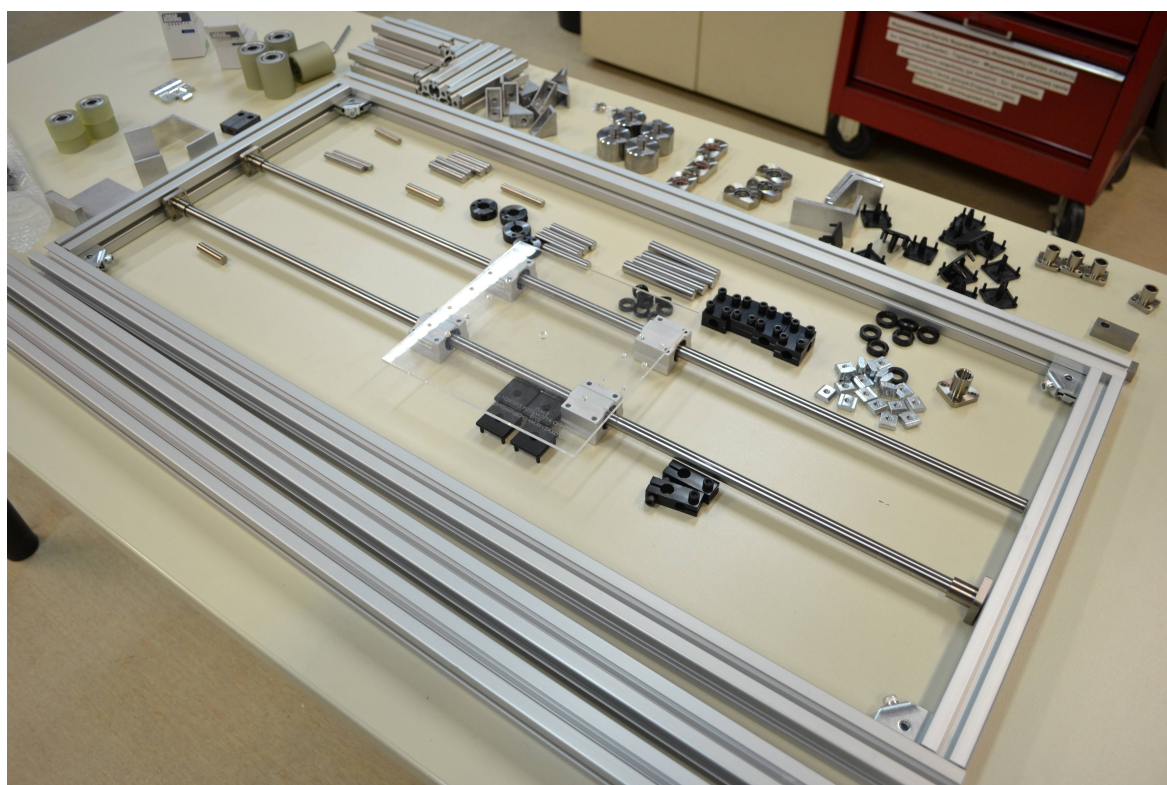
Όπως αναφέρθηκε έγινε η επιλογή το φορείο να σχεδιασθεί από μηδενική βάση. Οι λόγοι που οδήγησαν στην απόφαση αυτή αναφέρθηκαν προηγουμένως. Έτσι, έγινε έρευνα σε προϊόντα εταιριών, που κατασκευάζουν εξαρτήματα, για σχετικά μικρές μηχανολογικές κατασκευές. Τα προϊόντα των εταιριών αυτών χρησιμοποιούνται πολύ συχνά σε εφαρμογές ρομποτικής και αυτοματισμών, επομένως υπήρχε μια μικρή προηγούμενη εμπειρία.

Οι εταιρίες, στις οποίες κατέληξε η έρευνα, ήταν η Item Hellas και η Misumi Europe. Η πρώτη είναι αντιπρόσωπος στην Ελλάδα, της αντίστοιχης γερμανικής, ενώ η δεύτερη δεν έχει αντιπρόσωπο στην Ελλάδα, και οι παραγγελίες γίνονται απ'ευθείας στην εταιρία της Γερμανίας. Παρά το γεγονός αυτό, προτιμήθηκαν τα προϊόντα της Misumi, για μια σειρά από λόγους.

Αρχικά η Misumi παρέχει μια πολύ μεγαλύτερη γκάμα προϊόντων, τουλάχιστον για την συγκεκριμένη εφαρμογή, παρέχοντας μεγαλύτερη ευελιξία στο σχεδιασμό. Επιπλέον από οικονομικής άποψης, συνέφερε πολύ περισσότερο η Misumi. Και αυτό, συνυπολογίζοντας και τα μεταφορικά, από τη Γερμανία, παρά το ότι, για την Item δεν θα υπήρχαν τέτοια έξοδα. Τέλος, ύστερα από επαφή που επιδιώξαμε και με τις δυο εταιρίες, καταλήξαμε ότι η Misumi παρείχε πολύ πιο άμεση εξυπηρέτηση και υποστήριξη. Στα Σχήματα 3-1 και 3-2 παρουσιάζονται φωτογραφίες με όλα τα τεμάχια που αγοράστηκαν από την Misumi, και που αποτελούν σε μεγάλο μέρος την διάταξη.



Σχήμα 3-1: Τεμάχια (από Misumi) πριν την συναρμολόγηση.

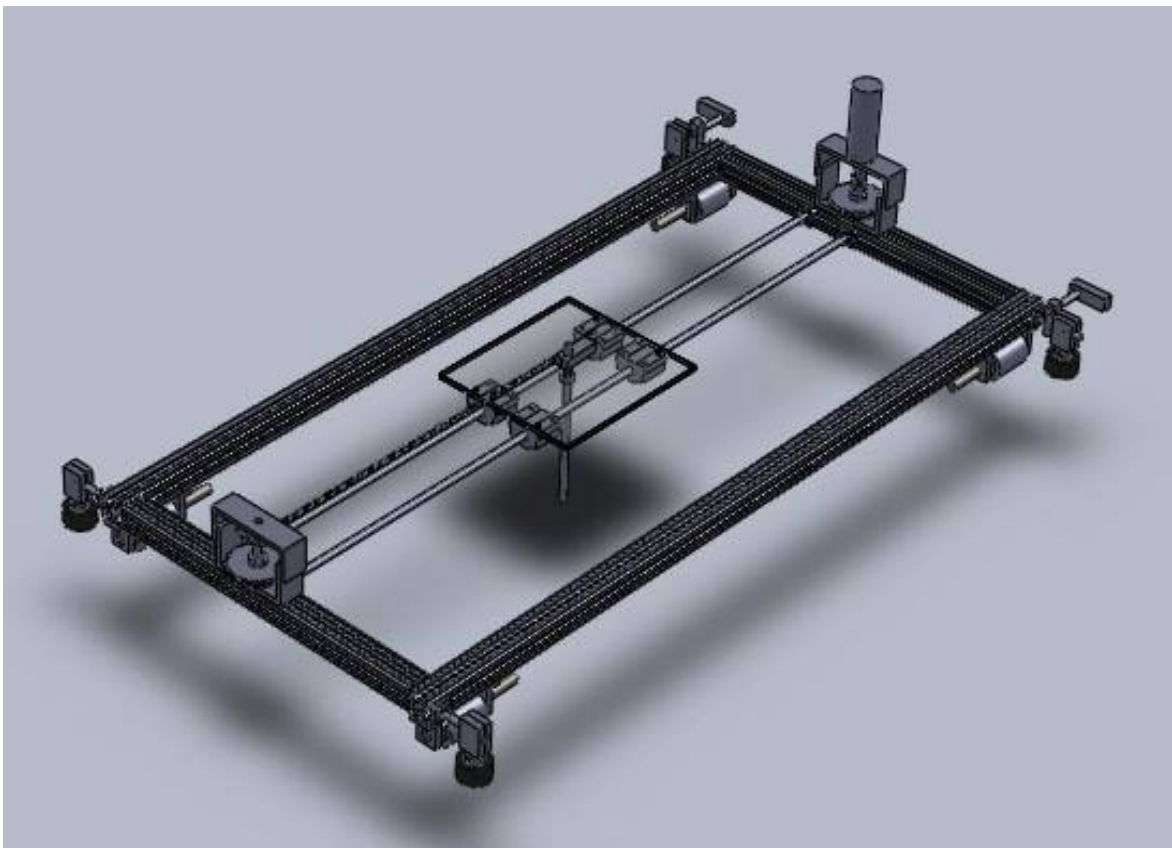


Σχήμα 3-2: Μέρος του συναρμολογημένου φορείου ανάμεσα σε άλλα εξαρτήματα.

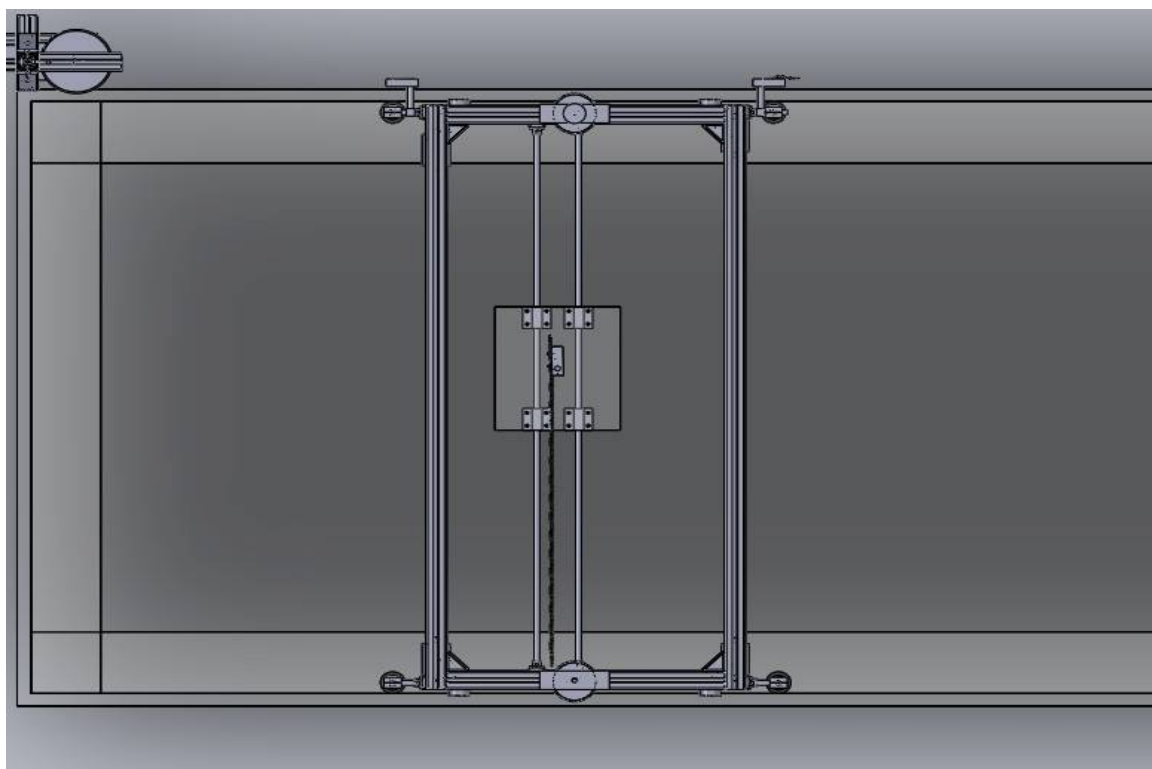
Η κίνηση του φορείου γίνεται πάνω σε τέσσερα ροδάκια πολυουρεθάνης, τα οποία κινούνται στα οριζόντια “χείλη” της δεξαμενής. Ακόμα, για να εξασφαλιστεί απόλυτα ευθύγραμμη

κίνηση, χρησιμοποιήθηκαν άλλα τέσσερα παρόμοια αλλά μικρότερα ροδάκια, τα οποία περιστρέφονται πάνω στα τοιχώματα της δεξαμενής που προεξέχουν πάνω από τα οριζόντια “χείλη” της. Όλα τα ροδάκια περιστρέφονται με τη βοήθεια ρουλεμάν. Έτσι επιτυγχάνεται απόλυτα ευθύγραμμη κίνηση, με εξαιρετικά χαμηλή τριβή.

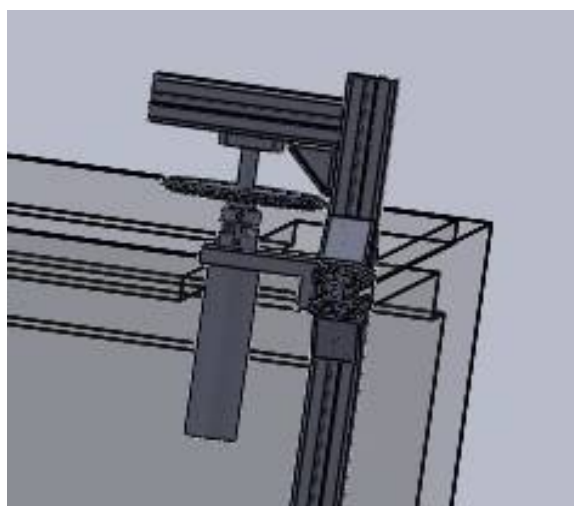
Στα Σχήματα 3-3 έως 3-9, όπου φαίνεται το συναρμολογημένο φορείο πάνω στη δεξαμενή, εύκολα μπορούν να διακριθούν τα βαγονάκια στα οποία γίνεται η κίνηση κατά τον y άξονα. Ακόμα διακρίνονται οι ρόδες πάνω στις οποίες στηρίζεται το φορείο και κινείται κατά τον x άξονα. Επιπλέον φαίνεται η ράβδος στην οποία μπορεί να προσδεθεί το ρομποτικό ψάρι κατά την κίνηση του. Εύκολα μπορεί να διακρίνει κανείς, τον κινητήρα του y άξονα, καθώς και την αλυσίδα, που μετατρέπει την περιστροφική κίνηση των τροχαλίων στην πρισματική του φορείου. Στα τελευταία θα αναφερθούμε εκτενέστερα στη συνέχεια. Τα Σχήματα 3-3 έως 3-6, αποτελούν μέρη των σχεδίων, στα οποία βασίστηκε η κατασκευή. Ακολουθούν τα Σχήματα 3-7 έως 3-11, που δείχνουν το φορείο μετά την ολοκλήρωση της κατασκευής, στις αντίστοιχες θέσεις, ώστε να είναι δυνατή η σύγκριση, του αρχικού σχεδιασμού με το τελικό αποτέλεσμα.



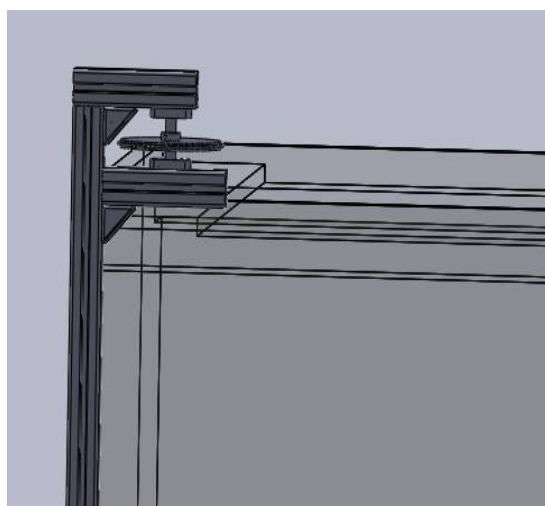
Σχήμα 3-3: Τρισδιάστατο σχέδιο φορείου.



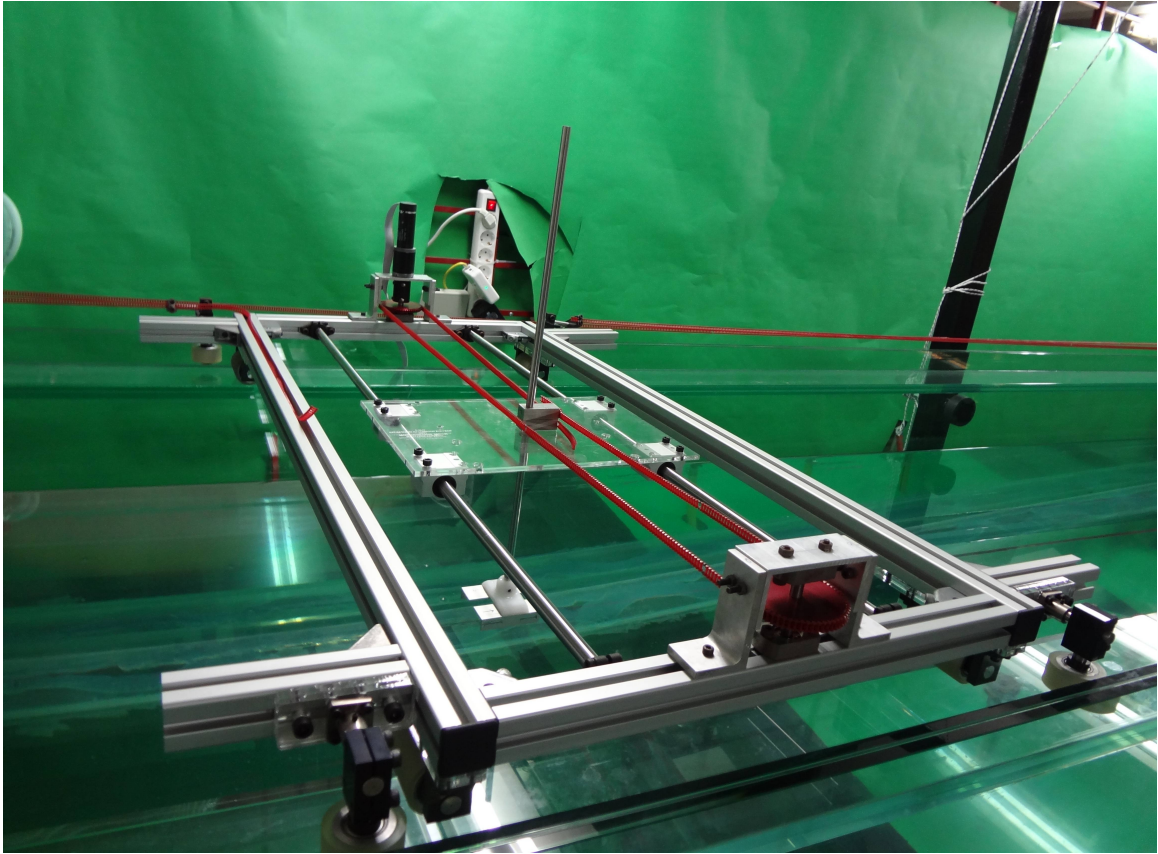
Σχήμα 3-4: Τρισδιάστατο σχέδιο φορείου πάνω στη δεξαμενή (κάτοψη).



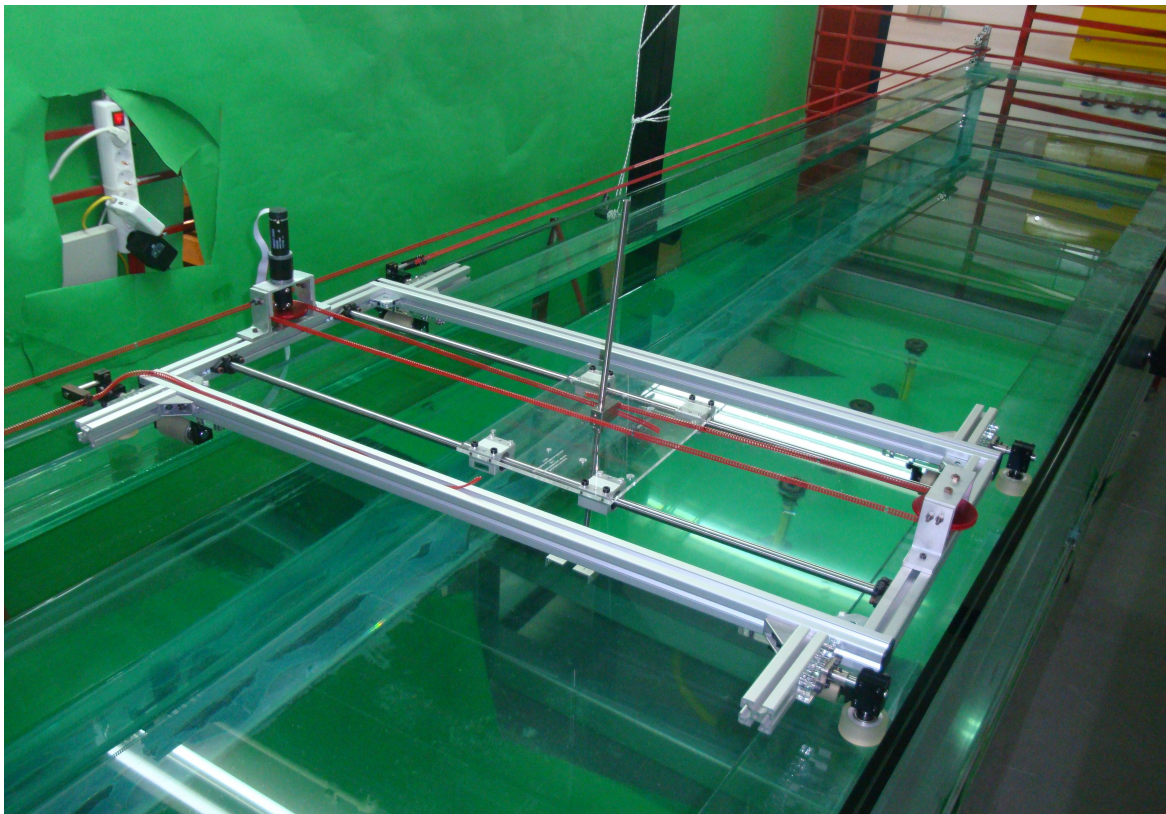
Σχήμα 3-5: Ράβδος αλουμινίου για τη στήριξη κινητήρα και τροχαλίας στον x άξονα.



Σχήμα 3-6: Ράβδος αλουμινίου για τη στήριξη τροχαλίας στον x άξονα.



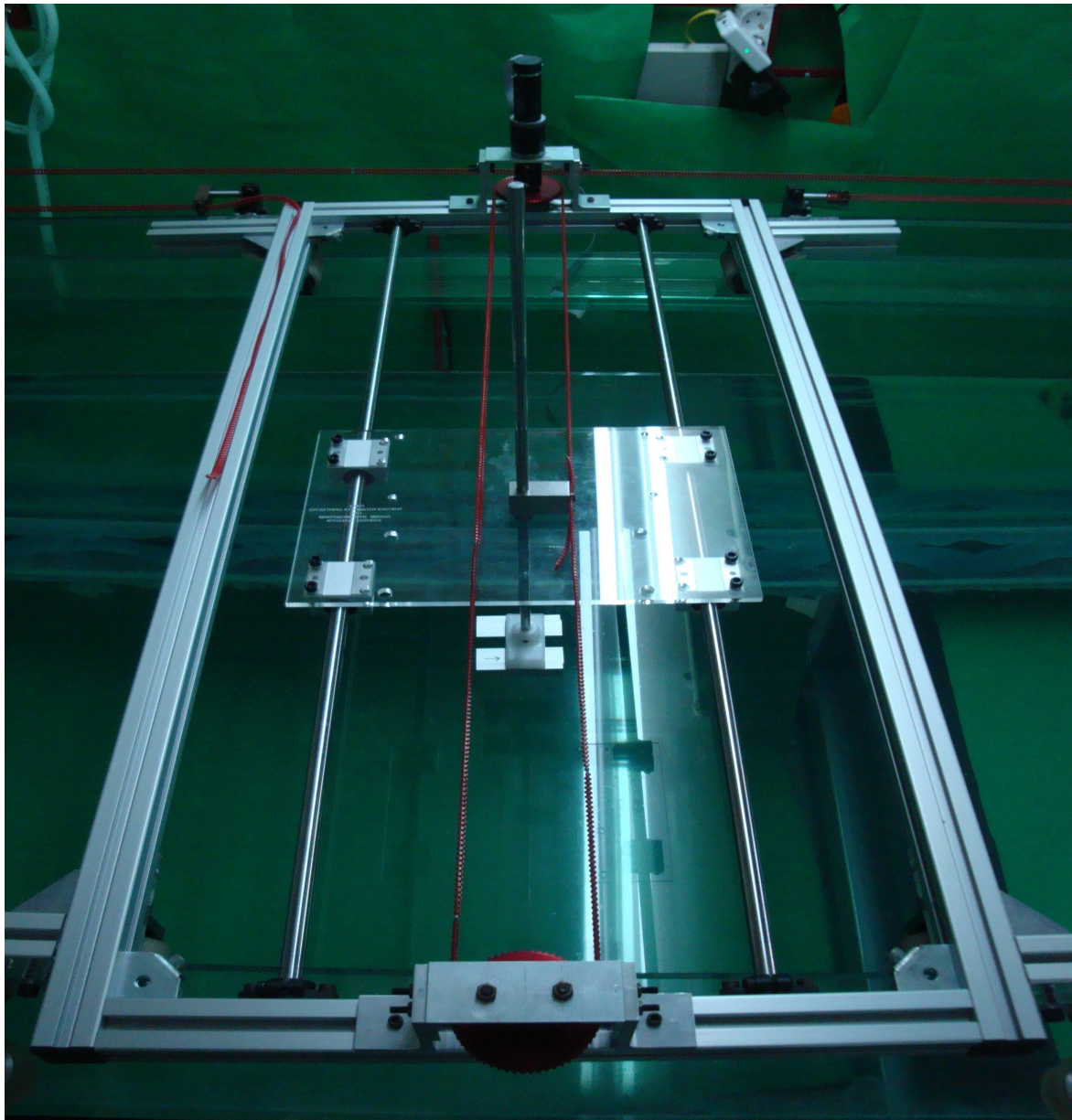
Σχήμα 3-7: Συναρμολογημένο φορείο πάνω στη δεξαμενή (πλάγια όψη α').



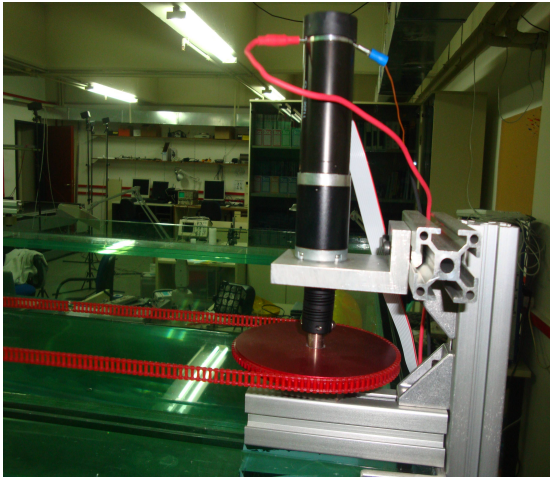
Σχήμα 3-8: Συναρμολογημένο φορείο πάνω στη δεξαμενή (πλάγια όψη β').

Με μεγαλύτερη λεπτομέρεια, στο Σχήμα 3-9, φαίνονται καθαρά τα τέσσερα βαγονάκια, στα οποία βιδώνεται το τεμάχιο του plexiglass. Τα βαγονάκια κινούνται πάνω στους δυο άξονες, επιτυγχάνοντας με τον τρόπο αυτό την κίνηση κατά τον y άξονα.

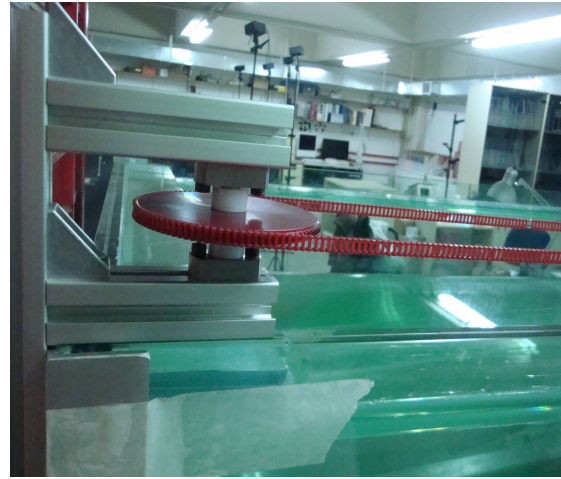
Στα Σχήματα 3-10 και 3-11 φαίνεται η πρόσδεση των τροχαλιών κατά τον x άξονα, μετά την ολοκλήρωση της συναρμολόγησης. Όπως φαίνεται και στα σχέδια (βλ. Σχ. 3-5 και 3-6), χρησιμοποιήθηκαν, δύο ράβδοι αλουμινίου στα δυο άκρα της δεξαμενής, οι οποίες στερεώθηκαν, πάνω σε αυτή. Στις ράβδους, στερεώθηκαν, με την βοήθεια μεταλλικών και αλουμινένιων τεμαχίων, όλα τα απαραίτητα εξαρτήματα για την κίνηση κατά τον x άξονα, δηλαδή ο κινητήρας και οι τροχαλίες για την μετάδοση της κίνησης.



Σχήμα 3-9: Τρισδιάστατο σχέδιο φορείου πάνω στη δεξαμενή (όψη από επάνω).



Σχήμα 3-10: Ράβδος αλουμινίου για τη στήριξη κινητήρα και τροχαλίας στον x άξονα.



Σχήμα 3-11: Ράβδος αλουμινίου για τη στήριξη τροχαλίας στον x άξονα.

Μετά την ολοκλήρωση της κατασκευής, ελάχιστες τροποποιήσεις είχαν γίνει σε σχέση με τα αρχικά σχέδια. Ένα ενδεικτικό παράδειγμα τέτοιας αλλαγής, ήταν το πώς θα κινείται η αλυσίδα γύρω από το μεταλλικό ικρίωμα στο μέσο της δεξαμενής αφού, εξαιτίας εσφαλμένης εκτίμησης αρχικά, οδήγησε σε επανασχεδιασμό την κατασκευή έστω σε μικρό βαθμό. Δηλαδή, στον αρχικό σχεδιασμό, το ένα στέλεχος της πλαστικής αλυσίδας του φορείου θα πέρναγε εσωτερικά του ικριώματος, και το άλλο εξωτερικά. Τελικά, όταν τοποθετήθηκε το φορείο και οι τροχαλίες στην θέση τους, αυτό δεν γίνονταν εφικτό. Έτσι, με κατάλληλες ρυθμίσεις στον εξοπλισμό του φορείου, και τα δυο στελέχη της αλυσίδας περνάνε εσωτερικά του μεταλλικού πλαισίου. Ένα ακόμα παράδειγμα φαίνεται στο Σχήμα 3-10, όπου παρουσιάζεται ο τρόπος πρόσδεσης του κινητήρα του x άξονα. Στο Σχήμα 3-5, φαίνεται ο αρχικός σχεδιασμός. Εξαιτίας της αλλαγής που έγινε στη διάταξη, λόγω της κακής εκτίμησης στην αλυσίδα που αναφέρθηκε, ο σχεδιασμός αυτός δεν μπορούσε να υλοποιηθεί. Τελικά η πρόσδεση του κινητήρα, έγινε με τον τρόπο που παρουσιάζεται στο Σχήμα 3-10. Όπως φαίνεται, η αλλαγή ήταν μικρής έκτασης, και δεν απαιτούσε κάποιο άλλο εξάρτημα, από αυτά που αρχικά είχαν σχεδιασθεί. Όμως ήταν αναγκαίο να γίνει με αυτόν τον τρόπο, ώστε να εξασφαλιστεί η σταθερή πρόσδεση του κινητήρα.

3.2.2 Επιλογή Μέσου Μετάδοσης Κίνησης

Ένα άλλο τμήμα της διάταξης μας, που απαιτήσε, αρκετή μελέτη για να επιλεγεί, ήταν το μέσο μετάδοσης της κίνησης, κατά τους δυο άξονες. Οι παράγοντες που έπρεπε να ληφθούν υπόψη, ήταν το πολύ μεγάλο μήκος, ιδιαίτερα κατά τον άξονα x (με συνολικό μήκος μεγαλύτερο από 10m), και το γεγονός ότι θα ήταν εκτεθειμένο στην επαφή με τον άνθρωπο και με το νερό της δεξαμενής. Φυσικά, θα έπρεπε να τηρηθούν και οι υπόλοιποι “κλασσικοί” παράγοντες, δηλαδή της

αντοχής στις δυνάμεις που ασκούνται κατά την κίνηση του φορείου, κίνηση και προς τις δύο κατευθύνσεις και του μειωμένου θορύβου κατά την λειτουργία. Η λύση αναζητήθηκε στα εξής προϊόντα: βιομηχανικές αλυσίδες, ιμάντες χρονισμού, ιμάντες τύπου V, μονάδες ανακυκλοφορούμενων σφαιριδίων, σύστημα κορώνας-ατέρμονα κοχλία και πλαστικές αλυσίδες. Στον πίνακα που ακολουθεί φαίνεται η διαδικασία αξιολόγησης των παραπάνω λύσεων.

Πίνακας 3-1: Πίνακας επιλογής του μέσου μετάδοσης κίνησης.

	Βιομηχανικές αλυσίδες	Ιμάντες χρονισμού	Ιμάντες τύπου V	Μονάδα ανακυκλοφορούμενων σφαιριδίων	Κορώνα – ατέρμονας κοχλίας	Πλαστικές αλυσίδες
Μεγάλα μήκη	✓	X	X	?	?	✓
Εκτεθειμένα σε επαφή (νερό, άνθρωποι)	X	X	X	?	✓	✓
Αντοχή	✓	✓	?	✓	✓	✓
Κίνηση σε δύο κατευθύνσεις	✓	✓	✓	✓	?	✓
Αθόρυβη λειτουργία	X	✓	✓	X	X	✓

Με την ολοκλήρωση της διαδικασίας αυτής, επιλέχθηκαν ως μέσο μετάδοσης της κίνησης στους δυο άξονες, πλαστικές αλυσίδες. Οι πλαστικές αλυσίδες, μπορούν να χαρακτηριστούν σαν σύνθετα υλικά, αφού αποτελούνται από δυο συρμάτινα στελέχη επενδεδυμένα με πλαστικό. Με τον τρόπο αυτό επιτυγχάνεται πολύ υψηλή αντοχή, για τέτοιας κλίμακας εφαρμογές. Η λειτουργία τους δεν επηρεάζεται από το νερό, μπορεί να κινηθεί με την ίδια άνεση και στις δυο κατευθύνσεις, και φυσικά εξασφαλίζεται κατά το δυνατόν αθόρυβη λειτουργία.

Στη συνέχεια ακολούθησε έρευνα, ώστε να βρεθεί εταιρία που να εμπορεύεται πλαστικές αλυσίδες. Επειδή δεν βρέθηκε στην Ελλάδα τέτοια εταιρία, η έρευνα στράφηκε στο εξωτερικό. Μετά από αναζήτηση, καταλήξαμε στην W.M.Berg, εταιρία που κατασκευάζει εξαρτήματα ακριβείας, που χρησιμοποιούνται συχνά σε ρομποτικές εφαρμογές. Όπως αναφέρθηκε και

προηγούμενα, το μήκος της αλυσίδας του x άξονα, ξεπερνάει τα 10m, ενώ η αλυσίδα στον y άξονα πρέπει να ξεπερνάει τα 2m. Για να έχουμε την δυνατότητα να χρησιμοποιήσουμε, σε περίπτωση σφάλματος ή κακού υπολογισμού επιπλέον τμήμα της αλυσίδας, επιλέχθηκε, η προμήθεια τους να γίνει σε μικρότερα τμήματα, του ενός μέτρου. Τα κομμάτια αυτά με τη βοήθεια ειδικού εξοπλισμού που παρέχεται από την εταιρία, συνενώθηκαν, ώστε να φτάσουν στο επιθυμητό μήκος. Στις προϋποθέσεις που είχαν τεθεί για την διάταξη, η μέγιστη ταχύτητα του φορείου, θα πρέπει να φτάνει τα 60mm/s. Υπολογίζοντας τα φορτία που θα πρέπει να υπερνικήσει το φορείο κατά την κίνηση του, καταλήξαμε ότι θα πρέπει να του ασκείται δύναμη περίπου ίση με 53N. Επομένως θα πρέπει και η μέγιστη δύναμη που αντέχει η αλυσίδα να είναι μεγαλύτερη από 70N περίπου, λαμβάνοντας έναν συντελεστή ασφαλείας ίσο με 1.3. Ο συντελεστής αυτός μπορεί να φαίνεται πολύ υψηλός, ειδικά για μια τέτοια ελαφριά κατασκευή. Μας εξασφαλίζει όμως, έναντι των προσεγγίσεων που αναγκαστικά έγιναν σε διάφορα στάδια των υπολογισμών, όπως κατά τον υπολογισμό της τριβής, των υδροδυναμικών αντιστάσεων κ.α.

Με βάση όλους αυτούς τους παράγοντες, επιλέχθηκε για τον x άξονα, η αλυσίδα με κωδικό αριθμό 24GCF-380-E, και για τον y άξονα η αλυσίδα με κωδικό αριθμό 24GCF-95-E. Στην ουσία αυτές οι δυο αλυσίδες είναι ίδιες. Διαφοροποιούνται μόνο στο μήκος, που όπως αναφέρθηκε επιλέχθηκε διαφορετικό για τους δυο άξονες. Στη συνέχεια δίνονται συγκεντρωτικά τα χαρακτηριστικά της αλυσίδας, καθώς και φωτογραφία (Σχήμα 3-12).

Πίνακας 3-2: Χαρακτηριστικά Στοιχεία Αλυσίδων 24GCF.

Pitch	3.325mm
Μέγιστο φορτίο	440N
Φορτίο συνεχούς λειτουργίας	108N
Μέγιστη ταχύτητα	1.91m/s
Υλικό	Πολυουρεθάνη



Σχήμα 3-12: Αλυσίδα 24GCF της Berg.

Στη συνέχεια, επιλέχθηκαν και οι συνεργαζόμενες τροχαλίες. Αυτές θα πρέπει να συνεργάζονται με αυτόν τον τύπο αλυσίδας (ίδιο pitch κ.λ.π), και να έχουν διάμετρο που προέκυψε από τους υπολογισμούς για την απαιτούμενη ροπή. Έτσι, οι τροχαλίες που επιλέχθηκαν έχουν κωδικό 24B-110-96 και 24B-108-60, όπου οι δυο τελευταίοι αριθμοί είναι ο αριθμός των οδόντων. Έτσι οι τροχαλίες του x άξονα έχουν εξωτερική διάμετρο 103.6mm, ενώ οι τροχαλίες του y άξονα 65.54mm. Στο Σχήμα 3-13 φαίνεται η τροχαλία, που στην ουσία είναι ίδια για τους δύο άξονες, με μόνη διαφορά την διάμετρο.

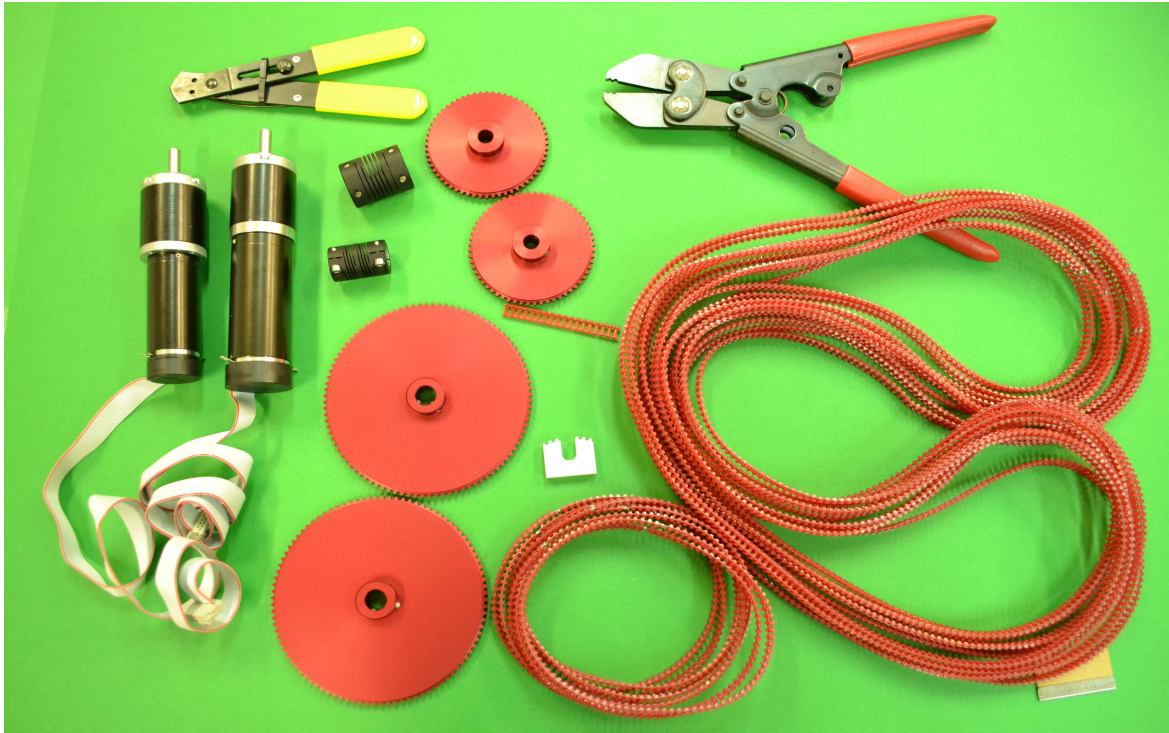


Σχήμα 3-13: Τροχαλία 24B της Berg.

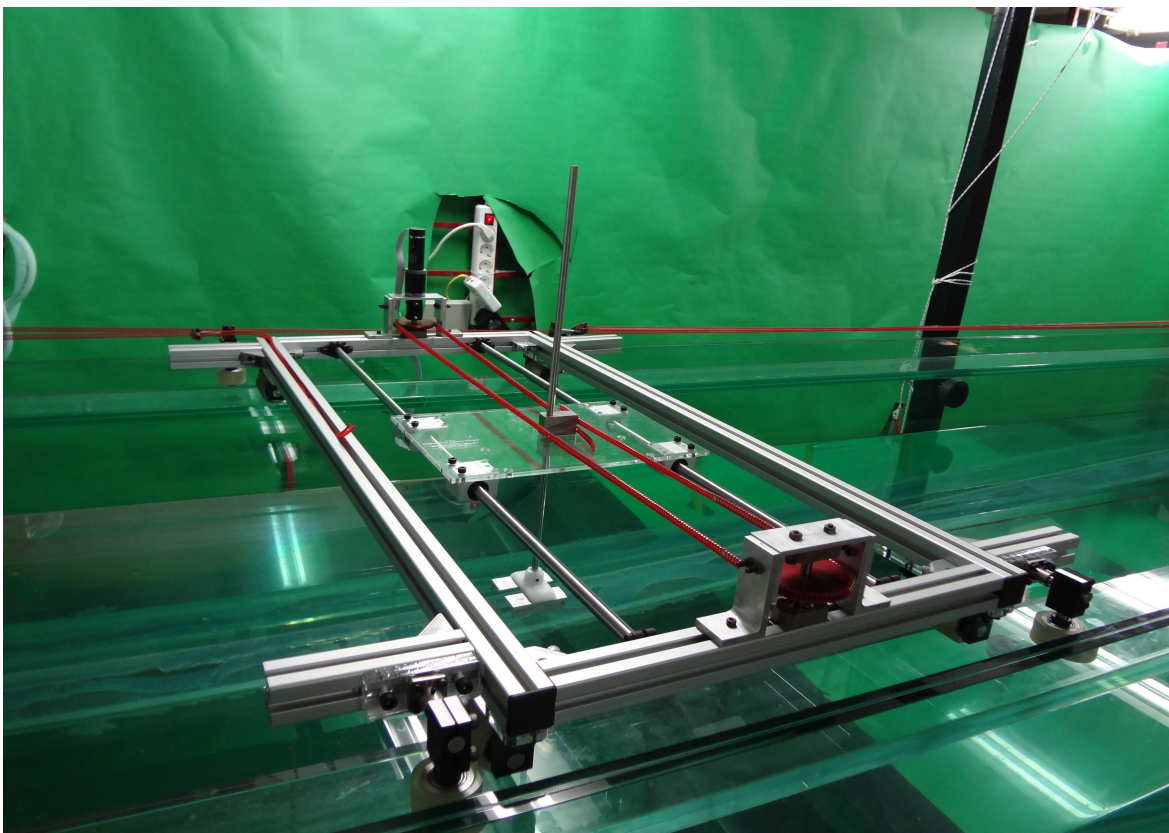
Όπως και στα αντίστοιχα σχήματα της διάταξης, προτιμήθηκε οι τροχαλίες να τοποθετηθούν, με τον άξονα περιστροφής τους, κάθετα προς το έδαφος. Αυτό έγινε, ώστε να αποφευχθεί μακροπρόθεσμα, η χαλάρωση του κάτω στελέχους της αλυσίδας, αν αυτή τοποθετούνταν με τον άξονα περιστροφής παράλληλα με το έδαφος.

Τέλος, από την ίδια εταιρία προμηθευτήκαμε και τα κόπλερ, που συνδέουν τους άξονες των τροχαλιών, με τους άξονες των κινητήρων. Αρχικά υπολογίσαμε τις ροπές στους άξονες αυτούς. Στη συνέχεια επιλέξαμε τις διαμέτρους των αξόνων, ώστε να αντέχουν στα φορτία αυτά. Έτσι με βάση τις υπολογισμένες διαμέτρους των αξόνων, προμηθευτήκαμε δυο κόπλερ για τους δυο άξονες, με κωδικούς CO41A-3M και COS78A-2M. Στο Σχήμα 3-14 που ακολουθεί φαίνονται όλα τα τεμάχια που προμηθευτήκαμε από την W.M.Berg. Αναλυτικά, αυτά είναι οι αλυσίδες για τον x και τον y άξονα, οι τροχαλίες των δυο αξόνων, τα δυο κόπλερ και εργαλεία για την συνένωση των επιμέρους κομματιών των αλυσίδων.

Στο Σχήμα 3-14 φαίνονται και οι κινητήρες για τους δυο άξονες, οι οποίοι θα περιγραφούν πιο αναλυτικά και στη συνέχεια.



Σχήμα 3-14: Στοιχεία μετάδοσης κίνησης της διάταξης και κινητήρες.



Σχήμα 3-15: Συναρμολογημένο φορείο και αλυσίδα μετάδοσης κίνησης κατά τον y άξονα.

3.2.3 Επιλογή Κινητήρων

Στη συνέχεια, θα γίνει η επιλογή των κινητήρων. Για να καταλήξουμε στον υπολογισμό του κάθε κινητήρα, θα ακολουθήσουμε την αντίστροφη διαδικασία. Υπολογίζουμε την ισχύ που θα απαιτηθεί για την κίνηση του φορείου, σε κάθε άξονα. Ο υπολογισμός γίνεται, με βάση τις προδιαγραφές ταχύτητας και επιτάχυνσης που έχουν τεθεί, και με βάση παραδοχές που έχουν γίνει κατά τον σχεδιασμό της διάταξης. Για παράδειγμα, την μέγιστη μάζα που θα μπορεί να έχει το φορείο, ή μια προσέγγιση των τριβών κατά την κίνηση του φορείου. Με βάση αυτά, υπολογίζουμε την ροπή στις τροχαλίες. Τέλος υπολογίζουμε την ισχύ και τη ροπή που θα πρέπει να παρέχει ο κίνητρες μέσω του κιβωτίου. Στην τελική ισχύ θα πρέπει να συνυπολογιστούν και οι απώλειες ισχύος στα διάφορα στάδια, δηλαδή αδράνειες, τριβές, απώλειες στο κιβώτιο ταχυτήτων. Ακόμα θα πρέπει να τηρούνται και οι υπόλοιποι παράγοντες που υπεισέρχονται στην επιλογή κινητήρα. Τέτοιοι είναι οι ονομαστικές στροφές, το μέγιστο ρεύμα, η μέγιστη και η ονομαστική ροπή κ.α.

Στη συνέχεια του κεφαλαίου, χρησιμοποιούμε την εξής ονοματολογία για τα διάφορα μεγέθη: κάποια μεγέθη που συμβολίζονται με κάποιο κεφαλαίο γράμμα αφορούν στον x άξονα. Αντίθετα τα αντίστοιχα μεγέθη που σχετίζονται με τον y άξονα, συμβολίζονται με τα αντίστοιχα μικρά γράμματα. Ακόμα οι αριθμητικές τιμές των μεγεθών που έχουν υπολογιστεί για την διάταξη μας, δίνονται στους αντίστοιχους πίνακες, και προέκυψαν από τις σχέσεις που παρατίθενται στη συνέχεια.

Αρχικά γίνεται η παραδοχή ότι η συνολική μάζα του φορείου δεν μπορεί να ξεπερνάει τα 25kg. Ακόμα στις προδιαγραφές της διάταξης έχει οριστεί η μέγιστη ταχύτητα του φορείου ίση με $v=0.6\text{m/s}$, και η μέγιστη επιτάχυνση $a=0.3\text{m/s}^2$. Ακόμα κάνουμε την παραδοχή, ότι η αντίσταση στη κίνηση του φορείου (λόγω τριβών, υδροδυναμικών αντιστάσεων κ.τ.λ.) δεν θα ξεπερνάει τα 30N.

Αυτό σημαίνει ότι για την κίνηση του φορείου, ώστε αυτό να κινηθεί με την μέγιστη επιτάχυνση απαιτείται ισχύς ίση με:

$$P_{\epsilon\pi.x} = (m_{\phi\omicron\rho.x} \cdot a_{MAX}) v_x \quad (3-1)$$

Ακόμα για να διατηρείται σταθερή η ταχύτητα του φορείου, θα πρέπει να του παρέχεται τόση ισχύς, ώστε να υπερνικούνται οι απώλειες λόγω τριβών. Επομένως είναι:

$$P_{\sigma\tau\alpha\theta.x} = F_{\sigma\tau\alpha\theta.x} \cdot v_x \quad (3-2)$$

Άρα η συνολική ισχύς που θα πρέπει να παρέχεται στο φορείο είναι:

$$P_{\phi\omicron\rho.x} = P_{\epsilon\pi.x} + P_{\sigma\tau\alpha\theta.x} \quad (3-3)$$

ενώ η συνολική δύναμη που θα πρέπει να ασκείται στο φορείο είναι ίση με:

$$F_{\phi\omicron\rho.x} = F_{\tau\rho\iota\beta.x} + F_{\epsilon\pi.x} \quad (3-4)$$

Στον πίνακα που ακολουθεί, εμφανίζονται συγκεντρωτικά τα φυσικά μεγέθη που υπολογίστηκαν για το φορείο κατά τον x άξονα:

Πίνακας 3-3: Φυσικά μεγέθη για το φορείο κατά τον x άξονα.

Φορείο		
Μέγιστη μάζα $[m_{φορ.x}]$	kg	25
Μέγιστη ταχύτητα $[v_{max,x}]$	m/s	0,6
Μέγιστη επιτάχυνση $[a_{MAX}]$	m/s ²	0,3
Μέγιστη δύναμη λόγω τριβών στο φορείο $[F_{τριβ.x}]$	N	30
Δύναμη στο φορείο κατά τη μέγιστη επιτάχυνση $[F_{επ.x}]$	N	7,5
Συνολική απαιτούμενη δύναμη στο φορείο $[F_{φορ.x}]$	N	37,5
Απαιτούμενη ισχύς στο φορείο κατά την μέγιστη επιτάχυνση $[P_{επ.x}]$	W	4,5
Απαιτούμενη ισχύς στο φορείο για σταθερή κίνηση με μέγιστη ταχύτητα $[P_{σταθ.x}]$	W	18
Συνολική απαιτούμενη ισχύς για το φορείο $[P_{σταθ.x}]$	W	18

Η ισχύς και η δύναμη, παρέχεται στο φορείο, με τη βοήθεια της πλαστικής αλυσίδας, από την τροχαλία. Η τροχαλία, όπως αναφέρθηκε, έχει ακτίνα $r_x \simeq 0.05\text{m}$. Η ροπή της τροχαλίας είναι:

$$T_{τροχ.x} = F_{φορ.x} \cdot r_x \quad (3-5)$$

και η ταχύτητα περιστροφής της τροχαλίας είναι

$$\omega_{τροχ.x} = \frac{v}{r_x} \quad (3-6)$$

Στον πίνακα που ακολουθεί, εμφανίζονται συγκεντρωτικά τα φυσικά μεγέθη που υπολογίστηκαν για τις τροχαλίες κατά τον x άξονα:

Πίνακας 3-4: Φυσικά μεγέθη για τις τροχαλίες κατά τον x άξονα.

Τροχαλία		
Ροπή τροχαλίας $[T_{τροχ.x}]$	Nm	1,88
Ταχύτητα περιστροφής τροχαλίας $[\omega_{τροχ.x}]$	rad/s	12
Ταχύτητα περιστροφής τροχαλίας $[n_{τροχ.x}]$	rpm	115
Ισχύς στην τροχαλία $[P_{τροχ.x}]$	W	22,56

Αν υποθέσουμε ότι έχουμε απώλειες ισχύος στον μειωτήρα του κινητήρα της τάξης του 30%, και για περισσότερη ασφάλεια, θεωρήσουμε ένα επιπλέον 20% απώλειες ισχύος αλλού. Τότε η ισχύς που θα πρέπει να παρέχει ο κινητήρας είναι:

$$P_{motor.x} = 1.5 \cdot P_{τροχ.x} \quad (3-7)$$

ενώ η ροπή στην έξοδο του κινητήρα θα πρέπει να είναι τουλάχιστον ίση με:

$$T_{OUT} = 1.3 \cdot T_{τροχ.x} \quad (3-8)$$

Στη συνέχεια επιλέγουμε μειωτήρα με λόγο μείωσης 35:1. Έτσι για να επιτύχουμε την μέγιστη ταχύτητα του φορείου (0.6m/sec), θα πρέπει οι στροφές του κινητήρα να είναι ίσες με

$$n_{motor.x} = n_{τροχ.x} \cdot i_x \quad (3-9)$$

ενώ η ροπή του κινητήρα (πριν το κιβώτιο) ίση με:

$$T_{motor.x} = \frac{T_{OUT}}{i_x} \quad (3-10)$$

Στον πίνακα που ακολουθεί, εμφανίζονται συγκεντρωτικά τα φυσικά μεγέθη που υπολογίστηκαν για τις τροχαλίες κατά τον x άξονα:

Πίνακας 3-5: Φυσικά μεγέθη για τον κινητήρα κατά τον x άξονα.

Κινητήρας			
Ταχύτητα περιστροφής κινητήρα	$[n_{motor.x}]$	rpm	4025
Ροπή κινητήρα (μετά το κιβώτιο)	$[T_{OUT}]$	Nm	2,44
Ροπή κινητήρα (πριν το κιβώτιο)	$[T_{motor.x}]$	mNm	69,7
Παρεχόμενη ισχύς από τον κινητήρα	$[P_{motor.x}]$	W	33,75

Ύστερα από αυτή την διαδικασία, επιλέχθηκε για τον άξονα x ο κινητήρας 310007 RE 60W, και κιβώτιο 358335 της εταιρίας Maxon Motors. Ο κινητήρας αυτός έχει σταθερά ροπής ίση με $K_t = 25.9 \text{ mNm} \cdot \text{A}^{-1}$ άρα το ρεύμα εκκίνησης θα είναι ίσο με:

$$i_{MAX} = \frac{T_{MOTOR}}{K_t} = \frac{69.2 \text{ mNm}}{25.9 \text{ mNm} \cdot \text{A}^{-1}} \Rightarrow i_{MAX} \simeq 2.6 \text{ A} \quad (3-11)$$

το οποίο είναι πολύ μικροτερο από το μέγιστο ρεύμα εκκίνησης που μπορεί να παρέχει ο συγκεκριμένος κινητήρας, που είναι ίσο με 39.3A. Αναφορικά με τον κινητήρα, όλες οι προϋποθέσεις για την επιλογή, τηρούνται απόλυτα, όπως οι ονομαστικές στροφές, η μέγιστη ροπή κ.α. Αναλυτικά τα στοιχεία του κινητήρα δίνονται στο παράρτημα της εργασίας.

Ακόμα θα πρέπει να ελέγξουμε αν οι παραδοχές που έγιναν και για τον μειωτήρα είναι αποδεκτές. Έτσι στα στοιχεία του μειωτήρα, δίνονται οι απώλειες ισχύος ίσες με 20%. Επομένως η παραδοχή που έγινε για απώλειες 30% είναι υπέρ της ασφάλειας. Τα υπόλοιπα στοιχεία του μειωτήρα καλύπτουν πλήρως τις προδιαγραφές της διάταξης, και δίνονται αναλυτικά στο παράρτημα της εργασίας.

Για τον έλεγχο θέσης και ταχύτητας του κινητήρα, έγινε και αγορά, ενός incremental encoder. Συγκεκριμένα αγοράστηκε ο encoder με κωδικό 228456 της Maxon, ο οποίος έχει 1000 counts ανά περιστροφή.

Με την ίδια διαδικασία γίνεται και η επιλογή του κινητήρα για τον y άξονα. Αρχικά κάνουμε

την παραδοχή ότι οι κινούμενες μάζες κατά τον y άξονα δεν μπορεί να είναι μεγαλύτερες από $m=7\text{kg}$. Ακόμα θέτουμε σαν προδιαγραφές για τον y άξονα μέγιστη ταχύτητα $v=0.3\text{m/s}$, και μέγιστη επιτάχυνση $a=0.3\text{m/s}^2$. Τέλος, υποθέτουμε ότι η μέγιστη αντίσταση κατά την κίνηση στον y άξονα, δεν μπορεί να ξεπερνάει τα 15N.

Έτσι, η ισχύς που θα πρέπει να παρέχεται στο φορείο κατά την επιταχυνόμενη κίνηση του (με την μέγιστη επιτάχυνση) είναι:

$$P_{\varepsilon\pi,y} = F_{\varepsilon\pi,y} \cdot v_y \quad (3-12)$$

ενώ κατά την κίνηση με σταθερή ταχύτητα η παρεχόμενη ισχύς θα πρέπει να είναι:

$$P_{\sigma\tau\alpha\theta,y} = F_{\tau\rho\iota\beta,y} \cdot v_y \quad (3-13)$$

Άρα, η συνολική ισχύς που θα πρέπει να παρέχεται στο φορείο θα είναι:

$$P_{\phi\omicron\rho,y} = P_{\varepsilon\pi,y} + P_{\sigma\tau\alpha\theta,y} \quad (3-14)$$

Η μέγιστη δύναμη που θα πρέπει να ασκείται στο φορείο είναι

$$F_{\phi\omicron\rho,y} = F_{\varepsilon\pi,y} + F_{\tau\rho\iota\beta,y} \quad (3-15)$$

Στον πίνακα που ακολουθεί, εμφανίζονται συγκεντρωτικά τα φυσικά μεγέθη που υπολογίστηκαν για το φορείο κατά τον y άξονα:

Πίνακας 3-6: Φυσικά μεγέθη για το φορείο κατά τον y άξονα.

Φορείο		
Μέγιστη μάζα $[m_{\phi\omicron\rho,y}]$	kg	7
Μέγιστη ταχύτητα $[v_{\max}]$	m/s	0,3
Μέγιστη επιτάχυνση $[a_{\max}]$	m/s ²	0,3
Μέγιστη δύναμη λόγω τριβών στο φορείο $[F_{\tau\rho\iota\beta,y}]$	N	15
Δύναμη στο φορείο κατά τη μέγιστη επιτάχυνση $[F_{\varepsilon\pi,y}]$	N	2,1
Συνολική απαιτούμενη δύναμη στο φορείο $[F_{\phi\omicron\rho,y}]$	N	17,1
Απαιτούμενη ισχύς στο φορείο κατά την μέγιστη επιτάχυνση $[P_{\varepsilon\pi,y}]$	W	0,63
Απαιτούμενη ισχύς στο φορείο για σταθερή κίνηση με μέγιστη ταχύτητα $[P_{\sigma\tau\alpha\theta,y}]$	W	4,5
Συνολική απαιτούμενη ισχύς για το φορείο $[P_{\phi\omicron\rho,y}]$	W	5,13

Οι τροχαλίες που έχουμε επιλέξει για τον y άξονα έχουν $r_y \simeq 0.03\text{m}$ και επειδή η μέγιστη γραμμική ταχύτητα σε αυτόν τον άξονα, έχει τεθεί ίση με $v=0.3\text{m/s}$, τότε οι μέγιστες στροφές της τροχαλίας είναι:

$$\omega_{\tau\rho\omicron\chi,y} = \frac{v_y}{r_y} \quad (3-16)$$

Ακόμα, η μέγιστη ροπή στην τροχαλία θα είναι:

$$T_{τροχ,y} = F_{φορ,y} \cdot r_y \quad (3-17)$$

Στον πίνακα που ακολουθεί, εμφανίζονται συγκεντρωτικά τα φυσικά μεγέθη που υπολογίστηκαν για τις τροχαλίες κατά τον y άξονα:

Πίνακας 3-7: Φυσικά μεγέθη για τις τροχαλίες κατά τον y άξονα.

Τροχαλία		
Ροπή τροχαλίας $[T_{τροχ,y}]$	Nm	0,51
Ταχύτητα περιστροφής τροχαλίας $[\omega_{τροχ,y}]$	rad/s	10
Ταχύτητα περιστροφής τροχαλίας $[n_{τροχ,y}]$	rpm	96
Ισχύς στην τροχαλία $[P_{τροχ,y}]$	Nm	5,13

Στη συνέχεια υποθέτουμε, απώλειες 40% στο κιβώτιο ταχυτήτων, και ένα 20% απώλειες σε άλλα σημεία. Άρα ο κινητήρας θα πρέπει να παρέχει ισχύ ίση με:

$$P_{motor,y} = 1.6 \cdot P_{τροχ,y} \quad (3-18)$$

Η ροπή που θα παρέχεται στην τροχαλία, θα πρέπει να είναι

$$T_{out} = 1.4 \cdot T_{τροχ,y} \quad (3-19)$$

Άρα αν επιλέξουμε το ίδιο κιβώτιο που επιλέξαμε και για τον x άξονα (με κωδικό 358335), που έχει σχέση μετάδοσης 35:1, η ροπή του κινητήρα θα πρέπει να είναι:

$$T_{motor,y} = \frac{T_{out}}{i_y} \quad (3-20)$$

ενώ οι στροφές του θα είναι:

$$n_{motor,y} = n_{T,y} \cdot i_y \quad (3-21)$$

Στον πίνακα 3-8, εμφανίζονται συγκεντρωτικά τα φυσικά μεγέθη που υπολογίστηκαν για τις τροχαλίες κατά τον x άξονα:

Πίνακας 3-8: Φυσικά μεγέθη για τον κινητήρα κατά τον y άξονα.

Κινητήρας		
Ταχύτητα περιστροφής κινητήρα $[n_{motor,y}]$	rpm	3360
Ροπή κινητήρα (μετά το κιβώτιο) $[T_{out}]$	Nm	0,72
Ροπή κινητήρα (πριν το κιβώτιο) $[T_{motor,y}]$	mNm	21
Παρεχόμενη ισχύς από τον κινητήρα $[P_{motor,y}]$	W	8,21

Επομένως επιλέγουμε τον κινητήρα RE 20W 118752 της Maxon. Ο κινητήρας αυτός έχει σταθερά

ροπής $K_t = 23.4 \text{ mNm} \cdot \text{A}^{-1}$. Δηλαδή το ρεύμα εκκίνησης θα είναι:

$$i_{\text{εκκιν.}} = \frac{T_{\text{motor}}}{K_t} = \frac{21 \text{ mNm}}{23.4 \text{ mNm}} = 0.90 \text{ A} \quad (3-22)$$

Όμως ο συγκεκριμένος κινητήρας μπορεί να δώσει μέγιστο ρεύμα εκκίνησης έως και 11A. Επομένως και όπως προκύπτει από τα υπόλοιπα στοιχεία του κινητήρα (δίνονται αναλυτικά στο παράρτημα), καλύπτει πλήρως τις προδιαγραφές που έχουν τεθεί για την κίνηση του φορείου στον y άξονα.

3.2.4 Επιλογή Τροφοδοτικού

Το τελευταίο βασικό στοιχείο για την λειτουργία της διάταξης, είναι το τροφοδοτικό. Για λόγους οικονομίας και απλότητας, επιλέχθηκε ένα κοινό τροφοδοτικό και για τους δυο κινητήρες. Αυτό είναι αρκετά εύκολο να επιτευχθεί, καθώς και οι δυο κινητήρες έχουν τάση τροφοδοσίας 24V. Επομένως, αρκεί το τροφοδοτικό να μπορεί να παρέχει αρκετό ρεύμα και για τους δυο κινητήρες ταυτόχρονα. Επιπλέον θα πρέπει να εξετάσουμε και άλλα στοιχεία του τροφοδοτικού, όπως την μέγιστη υπερακόντιση, την διακύμανση, την μέγιστη καθυστέρηση στην έξοδο, την μάζα και άλλα. Στον πίνακα που ακολουθεί δίνονται τα βασικά χαρακτηριστικά που θα πρέπει να έχει το τροφοδοτικό και για τους δυο κινητήρες, δηλαδή η τάση, το ρεύμα και η ισχύς.

Πίνακας 3-9: Πίνακας Επιλογής Τροφοδοτικού.

	Τάση (V)	Ρεύμα (A)	Ισχύς (W)
Κινητήρας x άξονα	24	3,5	80
Κινητήρας y άξονα	24	1,25	30
Και οι δυο κινητήρες ταυτόχρονα	24	4,75	110

Η αναζήτηση τροφοδοτικού που να καλύπτει αυτές τις προδιαγραφές, έγινε στη σειρά SITOP της SIEMENS. Τροφοδοτικά αυτής της σειράς, έχουν χρησιμοποιηθεί και παλαιότερα στο εργαστήριο, και θεωρούνται πολύ αξιόπιστα και λειτουργικά. Έτσι, από την συγκεκριμένη σειρά, επιλέχθηκε το τροφοδοτικό SIEMENS SITOP 24VDC SMART 5A 6EP1-333-2BA01. Το συγκεκριμένο τροφοδοτικό, καλύπτει την εφαρμογή, ως προς τα χαρακτηριστικά τάσης, ρεύματος και ισχύος. Ακόμα, στον πίνακα που ακολουθεί, δίνονται συνοπτικά, κάποια από τα κύρια χαρακτηριστικά που μας ενδιαφέρουν.

Πίνακας 3-10: Χαρακτηριστικά Τροφοδοτικού.

	Μέγιστη απόκλιση	Μέγιστη υπερακόντιση στην έξοδο	Καθυστέρηση στην έξοδο	Χρόνος αποκατάστασης	Αριθμός παράλληλων μονάδων σε λειτουργία	Βάρος
SIEMENS SITOP 24VDC SMART 5A 6EP1-333- 2BA01	3,00%	4,00%	0,1 s.	50 ms.	2	0,5 kg



Σχήμα 3-16: Τροφοδοτικό SIEMENS SITOP 24VDC SMART 5A 6EP1-333-2BA01.

3.3 Περιγραφή Ηλεκτρονικών Κυκλωμάτων Φορείου

3.3.1 Εισαγωγή

Σε αυτό το κεφάλαιο, θα δούμε πως υλοποιούνται ηλεκτρονικά οι εντολές που δίνονται από τον υπολογιστή, στους κινητήρες. Οι εντολές αυτές προκύπτουν σαν αποτελέσματα της επεξεργασίας που εκτελεί ο αλγόριθμος όπως αναφέρθηκε και πιο πάνω. Σκοπός είναι, η κάμερα του φορείου να βρίσκεται διαρκώς πάνω από τον στόχο, ώστε να τον παρακολουθεί. Κάποια από τα αποτελέσματα που προκύπτουν από την επεξεργασία είναι η απόσταση του στόχου από το κέντρο της εικόνας, που είναι και η επιθυμητή θέση για τον στόχο. Έτσι σχηματίζεται ένα διάνυσμα, το οποίο δίνεται σαν εντολή εισόδου σε έναν ελεγκτή, ο οποίος καθορίζει την κίνηση της κάμερας. Η κάμερα είναι προσδεσμένη σε ένα φορείο, ικανό να εκτελεί επίπεδες κινήσεις. Οι κινήσεις επιτυγχάνονται με

την βοήθεια των κινητήρων. Ο κάθε κινητήρας είναι υπεύθυνος για την κίνηση στον κάθε άξονα. Δηλαδή, ο ηλεκτρονικός εξοπλισμός (ελεγκτής), ελέγχει τον ηλεκτρολογικό εξοπλισμό (κινητήρες), ο οποίος με τη σειρά του κινεί τα μηχανικά τμήματα της διάταξης (φορείο-κάμερα). Τα δυο τελευταία κομμάτια της διάταξης έχουν περιγραφεί αναλυτικά, έτσι στη συνέχεια θα ασχοληθούμε με τον ελεγκτή.

3.3.2 Αναλυτική Περιγραφή Κυκλώματος Ελέγχου

Ο ελεγκτής αποτελείται από ένα σύνολο ηλεκτρονικών διατάξεων, επιφορτισμένων με διάφορες λειτουργίες. Το βασικό τμήμα του ηλεκτρονικού εξοπλισμού, είναι ο μικροελεγκτής (μE-microcontroller). Εκεί εισέρχονται οι εντολές, επεξεργάζονται, με βάση έναν αλγόριθμο, και στη συνέχεια παράγονται τα κατάλληλα εκείνα σήματα, που στέλνονται στους κινητήρες, ώστε οι τελευταίοι να εκτελέσουν τις επιθυμητές κινήσεις. Άλλα σημαντικά εξαρτήματα είναι η γέφυρα-H (H-Bridge), το κύκλωμα RS-232, ο σταθεροποιητής τάσης κ.α. Όλα αυτά θα τα εξετάσουμε αναλυτικά με βάση την πορεία του σήματος.

Αρχικά, θα ασχοληθούμε με την παροχή ισχύος στο κύκλωμα. Όλα τα ολοκληρωμένα κυκλώματα που περιλαμβάνονται στην ηλεκτρονική διάταξη, για να λειτουργήσουν πρέπει να τους παρέχεται τάση 5V. Τα περισσότερα όμως είναι αρκετά ευαίσθητα σε διακυμάνσεις τάσης. Για το λόγο αυτό θα πρέπει να χρησιμοποιήσουμε και έναν σταθεροποιητή τάσης. Ο σταθεροποιητής τάσης (voltage regulator), είναι μια ηλεκτρονική διάταξη που εξασφαλίζει σταθερή τάση στην έξοδο του, ανεξάρτητα από διακυμάνσεις στην είσοδο του. Όλες οι υπόλοιπες διατάξεις στο κύκλωμα, παίρνουν τάση από την έξοδο του σταθεροποιητή.

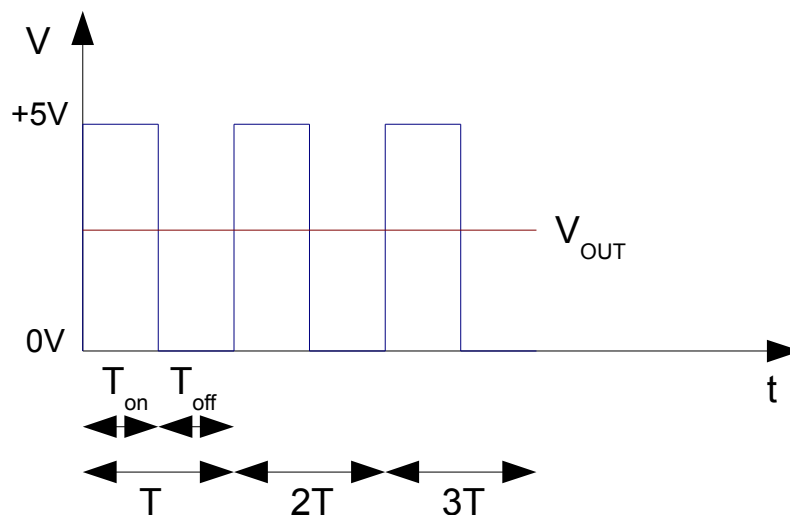
Οι εντολές που προκύπτουν από την επεξεργασία της εικονοροής, έρχονται στο κύκλωμα μέσω της σειριακής θύρας του υπολογιστή. Η πληροφορία αποτελείται από δυο αριθμούς. Οι αριθμοί αυτοί είναι η πραγματική απόσταση της τρέχουσας θέσης του στόχου, από την επιθυμητή, στους δυο άξονες (x και y), σε mm.

Η αναπαράσταση των δεδομένων στο κύκλωμα μας, είναι 0V για το 0, και +5V για το 1. Όμως η σειρακή θύρα του υπολογιστή δίνει -9V για το 0, και +9V για το 1. Για να μεταφράσουμε αυτήν την αναπαράσταση των δεδομένων, στους όρους του κυκλώματος μας, χρειαζόμαστε ένα ολοκληρωμένο κύκλωμα, το RS-232 chip. Το ολοκληρωμένο αυτό κάνει αυτή ακριβώς τη μετατροπή. Ύστερα από αυτό, τα δεδομένα μας είναι πλέον στην επιθυμητή μορφή, και μπορούμε πια να τα χρησιμοποιήσουμε σαν δεδομένα εισόδου στον ελεγκτή μας.

Ο μικροελεγκτής που χρησιμοποιήθηκε, είναι ο PIC18F4331 της Microchip. Τα κύρια χαρακτηριστικά του που αφορούν στην εφαρμογή μας είναι τα εξής. Μέγιστη ταχύτητα 40MHz, 8 ακροδέκτες PWM, για ταυτόχρονο έλεγχο μέχρι και δύο κινητήρων, αναλογοψηφιακό μετατροπέα ανάλυσης 10-bits, μονάδα ανάδρασης κίνησης με διεπαφή για οπτικό κωδικοποιητή, ακροδέκτες

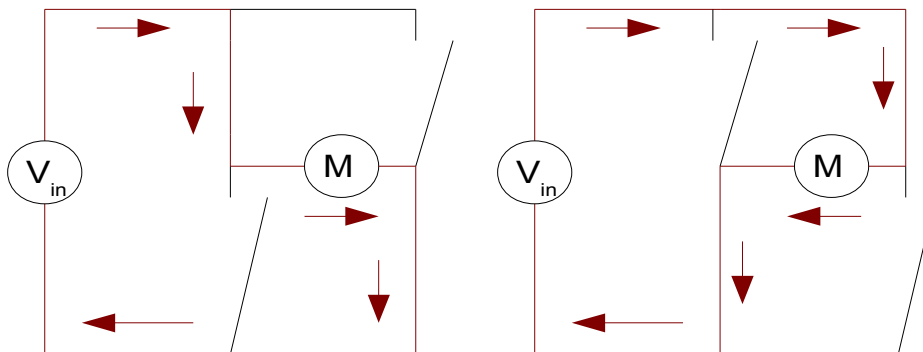
για επικοινωνίες (σειριακές, SPI/I2C), χρονιστές κ.α.

Το αποτέλεσμα της επεξεργασίας στον ελεγκτή, είναι η τάση έξοδου. Η τάση εμφανίζεται υπό τη μορφή παλμών ρυθμιζόμενου πλάτους – PWM (Pulse Width Modulation). Κάθε τέτοιος παλμός διαρκεί ένα χρονικό διάστημα. Ο κύκλος λειτουργίας, είναι το χρονικό διάστημα κατά το οποίο η έξοδος του ελεγκτή είναι σε κατάσταση 'on', ή 1, δηλαδή ισούται με 5V, ως προς τη συνολική διάρκεια κάθε παλμού. Στο παρακάτω σχήμα, φαίνεται πιο αναλυτικά η λειτουργία PWM. Έτσι στο Σχήμα 3-17 έχουμε, κάθε παλμό να διαρκεί χρονικό διάστημα T . Το διάστημα όπου ο παλμός είναι 5V, είναι T_{on} , ενώ το διάστημα κατά το οποίο ο παλμός είναι 0V, είναι T_{off} . Αν είναι $T_{on} = T_{off}$ τότε λέμε ότι ο κύκλος λειτουργίας είναι 50%. Επομένως, αν η τροφοδοσία του κινητήρα είναι 5V, τότε η τάση στα άκρα του θα είναι 2.5V. Δηλαδή ανάλογα με τον κύκλο λειτουργίας του PWM, ρυθμίζεται και η τάση του κινητήρα, ως κλάσμα της τάσης τροφοδοσίας. Αυτή είναι μια πολύ δημοφιλής μέθοδος ελέγχου κινητήρων.



Σχήμα 3-17: Παλμοί Ρυθμιζόμενου Πλάτους PWM.

Στη συνέχεια το σήμα, περνάει από ένα ολοκληρωμένο ηλεκτρονικό κύκλωμα, που ονομάζεται γέφυρα-H (H-Bridge). Η γέφυρα-H, επιτρέπει να εφαρμόζεται τάση, σε ένα φορτίο, σε οποιαδήποτε κατεύθυνση. Το πώς αυτό υλοποιείται φαίνεται στο Σχήμα 3-18, που ακολουθεί.



Σχήμα 3-18: Οι δυο βασικές καταστάσεις της γέφυρας-H.

Όπως φαίνεται η φορά περιστροφής του κινητήρα, εξαρτάται από την πολικότητα στα άκρα του. Επομένως, ανάλογα με το ποιο ζεύγος διακοπών είναι ανοιχτό, περιστρέφεται ανάλογα και ο κινητήρας.

Ο ελεγκτής κάνει αναλογικό-διαφορικό έλεγχο θέσης (PD-controller). Επομένως, παίρνει τα δεδομένα από την έξοδο του RS-232 chip, σαν σήματα εισόδου. Αν $u(t)$, τάση εξόδου και $e(t)$ η τάση εισόδου, τότε η εξίσωση που διέπει την λειτουργία ενός αναλογικού-διαφορικού ελεγκτή είναι η:

$$u(t) = K_c \cdot e(t) + K_c T_D \frac{de(t)}{dt} \quad (3-23)$$

ενώ η συνάρτηση μεταφοράς του ελεγκτή είναι

$$G_C(s) = K_C (1 + T_D s) \quad (3-24)$$

όπου K_C είναι το αναλογικό κέρδος, και T_D ο χρόνος διαφορίσης. Τα K_C και T_D μπορούν να ρυθμίζονται. Η χρήση του PD ελεγκτή, αυξάνει την ταχύτητα της απόκρισης των συστημάτων. Παράλληλα αυξάνει την ευστάθεια ενός συστήματος, γιατί ο διαφορικός όρος αυξάνει την απόσβεση του. Στην ουσία, ο αλγόριθμος του ελεγκτή υλοποιεί την παραπάνω εξίσωση. Τα K_C και T_D ρυθμίζονται με βάση την διάταξη για την οποία προορίζεται να λειτουργήσει ο ελεγκτής, σύμφωνα με τις αρχές της θεωρίας Αυτόματου Ελέγχου.

Στο Σχήμα 3-19, φαίνεται το σχέδιο της πλακέτας, όπου υλοποιήθηκε το ηλεκτρονικό κύκλωμα ελέγχου.

4 Παρακολούθηση Ρομποτικού Ψαριού Μέσω Κάμερας

4.1 Εισαγωγή

Στο δεύτερο μέρος της εργασίας, χρησιμοποιείται ψηφιακή κάμερα, για να παρακολουθείται διαρκώς το ρομποτικό ψάρι, και τα διάφορα στοιχεία της κίνησης του (π.χ. κίνηση ουράς κ.α.). Η κάμερα προσαρτάται πάνω στο φορείο, του οποίου ο σχεδιασμός και η κατασκευή, περιγράφηκε στο πρώτο μέρος της εργασίας, ή στο μαύρο μεταλλικό πλαίσιο, στο μέσο της δεξαμενής.

Έτσι, σε αυτό το τμήμα της εργασίας, θα περιγραφούν οι βασικές αρχές των ψηφιακών εικόνων αλλά και video (εικονοροής). Στη συνέχεια θα γίνει αναφορά, στις βασικές μεθόδους αναγνώρισης χαρακτηριστικών και σχημάτων, και η επιλογή της μεθόδου που τελικά επιλέχθηκε. Τέλος, θα παρουσιαστεί ο αλγόριθμος που χρησιμοποιήθηκε, για την παρακολούθηση του ρομποτικού ψαριού και τα αποτελέσματα που έδωσε.

4.2 Βασικές Αρχές Ψηφιακών Εικόνων

Ένας οπτικός αισθητήρας, λαμβάνει το σήμα του από τον πραγματικό κόσμο, δηλαδή λαμβάνει αναλογικό σήμα. Επομένως, πριν από οποιαδήποτε επεξεργασία, θα πρέπει το αναλογικό σήμα να μετατραπεί σε ψηφιακό. Έτσι μια πραγματική εικόνα, μεταφέρεται στον ψηφιακό κόσμο, με τη μορφή διακριτοποιημένου σήματος. Η αναπαράσταση μιας τέτοιας ψηφιακής εικόνας, μπορεί να είναι είτε δυαδική (binary image), είτε μονοχρωματική σε αποχρώσεις του γκρι (gray-scale image), είτε και έγχρωμη (color image).

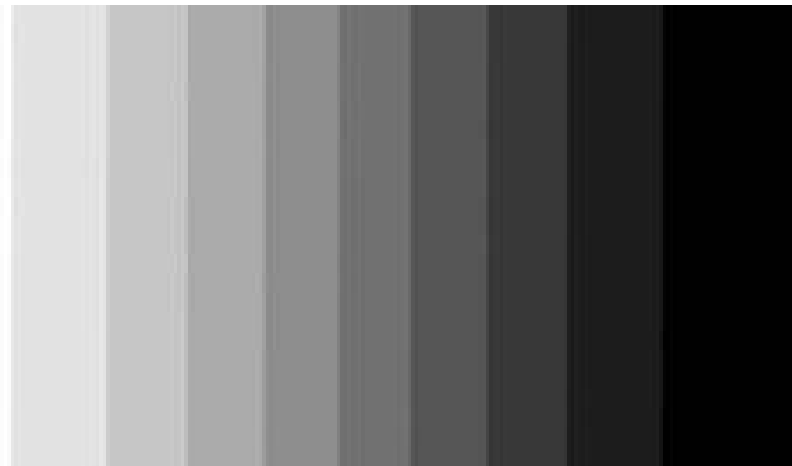
Η πιο απλή μορφή εικόνας είναι η δυαδική. Σε μια δυαδική εικόνα υπάρχουν μόνο δυο τιμές φωτεινότητας. Το μαύρο αντιστοιχεί στην τιμή 0, και το άσπρο στην τιμή 1. Προφανώς μια τέτοια εικόνα, καταλαμβάνει μικρότερη μνήμη και απαιτείται μικρότερο υπολογιστικό κόστος για την επεξεργασία της. Για το λόγο αυτό είναι πολύ χρήσιμη για την περιγραφή κάποιων γεωμετρικών ιδιοτήτων, όπως το εμβαδό και η περίμετρος. Εξαιτίας όμως της πολύ φτωχής ακρίβειας, είναι

τελείως ακατάλληλη για την αναπαράσταση πιο σύνθετων ιδιοτήτων. Μια τέτοια ψηφιακή δυαδική εικόνα φαίνεται στο Σχήμα 4-1.



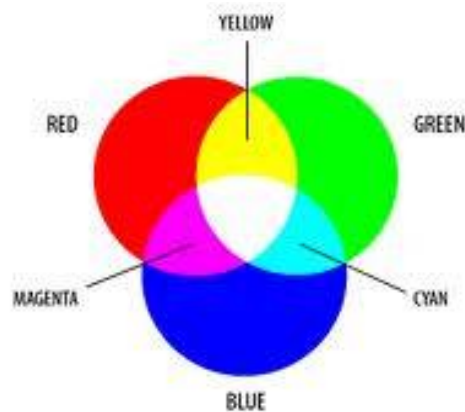
Σχήμα 4-1: Δυαδική ψηφιακή εικόνα.

Μια μονοχρωματική εικόνα αποχρώσεων του γκρι, μπορούμε να την περιγράψουμε σαν ένα σύνολο επιμέρους στοιχείων, που συχνά αναφέρονται ως εικονοστοιχεία ή pixels (από την φράση picture element). Τα στοιχεία αυτά, μπορούμε να θεωρήσουμε ότι σχηματίζουν ένα διδιάστατο πίνακα $I(i,j)$, όπου τα i ($i=1,...,N$), j ($j=1,...,M$), συμβολίζουν την γραμμή και την στήλη στην οποία βρίσκεται το κάθε εικονοστοιχείο, ενώ η τιμή $I(i,j)$, είναι ανάλογη της φωτεινότητας της εικόνας, στο εικονοστοιχείο (i,j) . Η τιμή της φωτεινότητας, μπορεί να πάρει μόνο τιμές, που προκύπτουν ως δυνάμεις του 2. Δηλαδή, $0 \leq I(i,j) \leq G-1$ όπου $G=2^m$. Πολύ συχνά είναι $m=8$, επομένως έχουμε 256 αποχρώσεις του γκρι. Κατά κανόνα βέβαια εκτός από το G , και τα i,j είναι δυνάμεις του 2, και αυτό για να διευκολύνεται η ψηφιακή επεξεργασία της εικόνας. Στο Σχήμα 4-2 φαίνονται οι διακριτές αποχρώσεις του γκρι, όπως χρησιμοποιούνται για την αναπαράσταση ψηφιακών εικόνων.



Σχήμα 4-2: Διακριτές αποχρώσεις του γκρι (μονοχρωματικές εικόνες αποχρώσεων του γκρι).

Μια πολύ πιο ρεαλιστική απεικόνιση του πραγματικού κόσμου, αποτελούν οι έγχρωμες εικόνες. Μια έγχρωμη εικόνα, μπορεί να θεωρηθεί ότι αποτελείται από τρεις gray-level εικόνες. Το πιο συνηθισμένο σύστημα απεικόνισης έγχρωμων εικόνων είναι το RGB (Red-Green-Blue). Δηλαδή κάθε εικόνα, προκύπτει από το άθροισμα των επιμέρους συνιστωσών, των διακριτών διαβαθμίσεων της εικόνας σε κόκκινο, πράσινο και μπλε επίπεδο, όπως φαίνεται και στο Σχήμα 4-3.



Σχήμα 4-3: Χρωματικό σύστημα RGB για απεικόνιση έγχρωμων εικόνων.

Με βάση τα προηγούμενα, βλέπουμε ότι μια εικόνα διαστάσεων $M \times N$ και πλήθους αποχρώσεων $G = 2^m$, απαιτεί $b = N \times M \times m$ bits, για να αποθηκευτεί. Το m , ονομάζεται βάθος bit (bit depth), ή βάθος χρώματος (color depth). Μεγαλύτερο m , σημαίνει περισσότερες διαθέσιμες αποχρώσεις, δηλαδή πιο πιστή αναπαράσταση της πραγματικής εικόνας. Έτσι μια έγχρωμη εικόνα, απαιτεί τριπλάσιο αριθμό bits, από μια αντιστοιχη gray-scale.

4.3 Εικόνα και Εικονοροή στο Περιβάλλον MATLAB

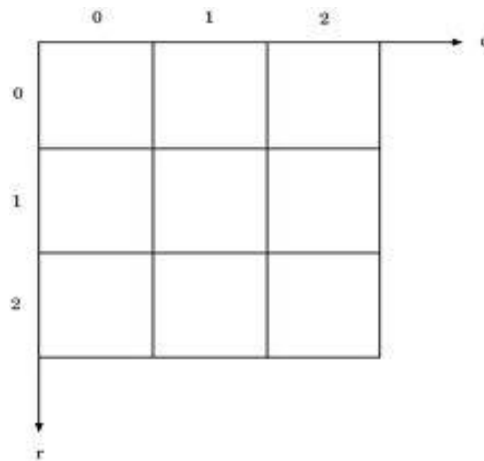
4.3.1 Επιλογή Λογισμικού

Για τις ανάγκες της εργασίας, χρησιμοποιήθηκε το λογισμικό MATLAB. Ο μεγάλος βαθμός εξοικείωσης με το συγκεκριμένο λογισμικό, και οι μεγάλες δυνατότητες που παρέχει, οδήγησε στην επιλογή του, έναντι άλλων αντίστοιχων. Επιπλέον, αρχικά είχε τεθεί το ζήτημα της χρησιμοποίησης λειτουργικού συστήματος πραγματικού χρόνου. Όμως, εξετάζοντας τις υπολογιστικές ανάγκες γι'αυτήν την εφαρμογή, κάτι τέτοιο δεν κρίθηκε απαραίτητο. Επιπλέον, στη MATLAB, υπάρχουν τρεις εργαλειοθήκες (toolboxes), που είναι προσαρμοσμένες στην απόκτηση εικόνας και video, και στην επεξεργασία τους. Αυτά είναι, το Image Acquisition toolbox, το Image Processing toolbox και το Image and Video Processing Blockset. Οι εργαλειοθήκες αυτές παρέχουν εργαλεία και συναρτήσεις, απόλυτα προσαρμοσμένα σε τέτοιου είδους εφαρμογές. Με τα εργαλεία αυτά μπορεί σχετικά εύκολα να προγραμματιστεί μια διαδικασία επεξεργασίας εικόνας. Επιπλέον υπάρχει και η δυνατότητα προσομοίωσης όλης της

διάταξης, δηλαδή του φορείου των κινητήρων και των ελεγκτών τους. Έτσι μπορούμε να έχουμε μια συνολική προσομοίωση του συνόλου της διάταξης, ώστε να μπορούμε να την μελετήσουμε πριν καν κατασκευαστεί.

4.3.2 Αναπαράσταση Εικόνων στη MATLAB

Όπως όλα τα δεδομένα στη MATLAB, έτσι και οι εικόνες αναπαριστώνται σαν πίνακες, διδιάστατοι στην περίπτωση δυαδικών εικόνων, ή εικόνων σε αποχρώσεις του γκρι. Κάθε στοιχείο, ενός τέτοιου πίνακα αντιστοιχεί σε ένα pixel της εικόνας. Για παράδειγμα, μια εικόνα διαστάσεων 480×640 η MATLAB, την “βλέπει” σαν έναν πίνακα διαστάσεων 480×640 . Μια έγχρωμη εικόνα ίδιων διαστάσεων σε σύστημα RGB, γίνεται αντιληπτή σαν ένας πίνακας διαστάσεων $480 \times 640 \times 3$, με βάση αυτά που έχουν αναφερθεί προηγουμένως. Στο σχήμα 4-4 που ακολουθεί φαίνεται ο τρόπος με τον οποίο, αναπαριστώνται οι εικόνες.



Σχήμα 4-4: Σύστημα Συντεταγμένων για Εικόνες του Λογισμικού MATLAB.

Όπως φαίνεται από το σχήμα ο αριθμός των γραμμών αυξάνει προς τα κάτω ενώ ο αριθμός των στηλών αυξάνει προς τα δεξιά. Προφανώς τα pixels κάθε εικόνας, είναι διακριτά, δηλαδή οι συντεταγμένες τους, μπορούν να έχουν διακριτές ακέραιες τιμές. Ο συμβολισμός κάθε pixel γίνεται με χρήση δυο στοιχείων (r,c). Ο πρώτος αριθμός δείχνει την γραμμή (row), και ο δεύτερος την στήλη (column). Ένα σημαντικό στοιχείο αυτού του συμβολισμού, σε σχέση με αυτά που αναφέρονται συνήθως στη θεωρία, είναι ότι η αρίθμηση στη MATLAB ξεκινάει από το (1,1) για το πάνω-αριστερά εικονοστοιχείο, και όχι από το (0,0).

4.4 Επιλογή Αλγόριθμου Εύρεσης και Παρακολούθησης Ρομποτικού Ψαριού

4.4.1 Εισαγωγή

Υπάρχουν διάφορες μέθοδοι, για την εύρεση στόχων σε μια εικόνα (object detection). Στη συνέχεια θα εξετάσουμε τις βασικότερες από αυτές, θα δούμε τα πλεονεκτήματα και τα μειονεκτήματα καθεμιάς, καθώς και τους λόγους που μας οδήγησαν σε αυτή που επιλέχθηκε.

Η διαδικασία εύρεσης μιας εικόνας-στόχου, μέσα σε μια μεγαλύτερη εικόνα, προϋποθέτει μια εικόνα του στόχου που θα χρησιμοποιηθεί σαν πρότυπο, και φυσικά την ευρύτερη εικόνα μέσα στην οποία ψάχνουμε για τον στόχο.

4.4.2 Ταυτοποίηση Εικόνων

Η πιο απλή μέθοδος, είναι η σύγκριση των δυο εικόνων, που μπορεί να επιτευχθεί για παράδειγμα μέσω αφαίρεσης. Έτσι στο σημείο όπου οι δυο εικόνες ταυτίζονται, αναμένεται το αποτέλεσμα της αφαίρεσης να είναι μηδέν. Αυτό όμως, πολύ σπάνια είναι δυνατόν να συμβαίνει. Μικρές διαφορές στο φωτισμό, μικρές αλλαγές στον προσανατολισμό και άλλοι απρόβλεπτοι παράγοντες, σπάνια επιτρέπουν την απόλυτη ταύτιση των δυο εικόνων. Αυτό το πρόβλημα μπορεί να ξεπεραστεί, θέτοντας ένα κατώφλι (thresholding). Δηλαδή, αν θεωρήσουμε ότι μικρές διαφορές στις δυο εικόνες είναι επιτρεπτές, δίνοντας ως τελικό αποτέλεσμα την ταύτιση τους. Η μέθοδος αυτή είναι πολύ απλή και πολύ εύκολο να προγραμματισθεί. Όμως παρουσιάζει και σοβαρά μειονεκτήματα, που εξηγούν και τον λόγο που πολύ σπάνια χρησιμοποιείται σε τέτοιες εφαρμογές. Για να βρεθεί η εικόνα-στόχος, πρέπει ένα τέτοιο πρόγραμμα να δοκιμάσει στην υπό εξέταση εικόνα, σε κάθε pixel αν υπάρχει ταύτιση με την εικόνα-στόχο. Επιπλέον δημιουργείται πρόβλημα στα εξωτερικά εικονοστοιχεία της εικόνας, καθώς δεν μπορεί να γίνει αφαίρεση. Το πρόβλημα αυτό ξεπερνιέται προσθέτοντας εικονοστοιχεία (padding), όμως δημιουργεί επιπλέον υπολογισμούς μεγαλώνοντας έτσι το χρόνο επεξεργασίας. Ακόμα χειρότερα αποτελέσματα προκύπτουν στην περίπτωση που ο στόχος έχει τυχαίο προσανατολισμό. Σε αυτή την περίπτωση, θα πρέπει να εξεταστεί κάθε pixel σε όλη την επιφάνεια της εικόνας, και μάλιστα με κάθε δυνατό προσανατολισμό. Για τους λόγους αυτούς η μέθοδος αυτή αποφεύγεται για την επεξεργασία εικόνων, ενώ για την περίπτωση εικονοροής όπου οι εικόνες έρχονται με πολύ γρήγορους ρυθμούς, είναι πρακτικά αδύνατη η εφαρμογή της.

4.4.3 Μετασχηματισμός Hough

Μια άλλη πολύ συχνά χρησιμοποιούμενη μέθοδος είναι ο μετασχηματισμός Hough. Ο μετασχηματισμός Hough (Hough Transform - HT) είναι μια τεχνική για να εντοπίζονται γεωμετρικά σχήματα σε μια εικόνα και προτάθηκε από τον Paul Hough το 1962. Αρχικά

χρησιμοποιήθηκε για την εύρεση ευθύγραμμων τμημάτων. Στη συνέχεια, εξελίχθηκε και βελτιώθηκε, ώστε να βρίσκει κύκλους, ελλείψεις και τελικά οποιασδήποτε μορφής σχήμα. Θα ακολουθήσει μια σύντομη περιγραφή των διαφορών μετασχηματισμών Hough, ώστε ο αναγνώστης να αντιληφθεί τον τρόπο λειτουργίας της μεθόδου. Για μεγαλύτερη εμβάθυνση, παρέχεται βιβλιογραφία [14], όπου η μέθοδος αναλύεται διεξοδικά.

4.4.3.1 Μετασχηματισμός Hough για Γραμμές

Κάθε σημείο στο επίπεδο μπορεί να χαρακτηριστεί από ένα ζεύγος συντεταγμένων. Στον HT όμως, κάθε σημείο χαρακτηρίζεται από ένα σύνολο ευθειών, που διέρχονται από αυτό. Έτσι, εξετάζουμε ένα σύνολο συνευθειακών σημείων, και τις ευθείες, που διέρχονται από καθένα από αυτά. Όπως είναι γνωστό από την Ευκλείδεια Γεωμετρία, θα υπάρχει μόνο μια ευθεία, που διέρχεται από όλα αυτά τα σημεία, και αντίστροφα. Δηλαδή, από όλες τις ευθείες που διέρχονται από κάθε σημείο, μόνο μια διέρχεται και από τα υπόλοιπα συνευθειακά σημεία ταυτόχρονα. Χρησιμοποιώντας καρτεσιανή παραμετροποίηση, ένα σύνολο συνευθειακών σημείων (x,y) μπορεί να συσχετιστεί με την παρακάτω σχέση:

$$y=mx+c \quad (4-1)$$

όπου m, η κλίση της ευθείας και c, το σημείο τομής με τον άξονα y'-y. Η παραπάνω εξίσωση μπορεί να γραφτεί και ως εξής:

$$Ay+Bx+1=0 \quad (4-2)$$

όπου $A = -1/c$, και $B = m/c$. Με τον τρόπο αυτό, ορίζεται μια γραμμή στο επίπεδο, από ένα ζεύγος τιμών (A,B). Έτσι, μπορούμε να αντιστοιχίσουμε μια γραμμή ορισμένη στον καρτεσιανό χώρο (x,y), σε έναν παραμετρικό χώρο με παραμέτρους (A,B).

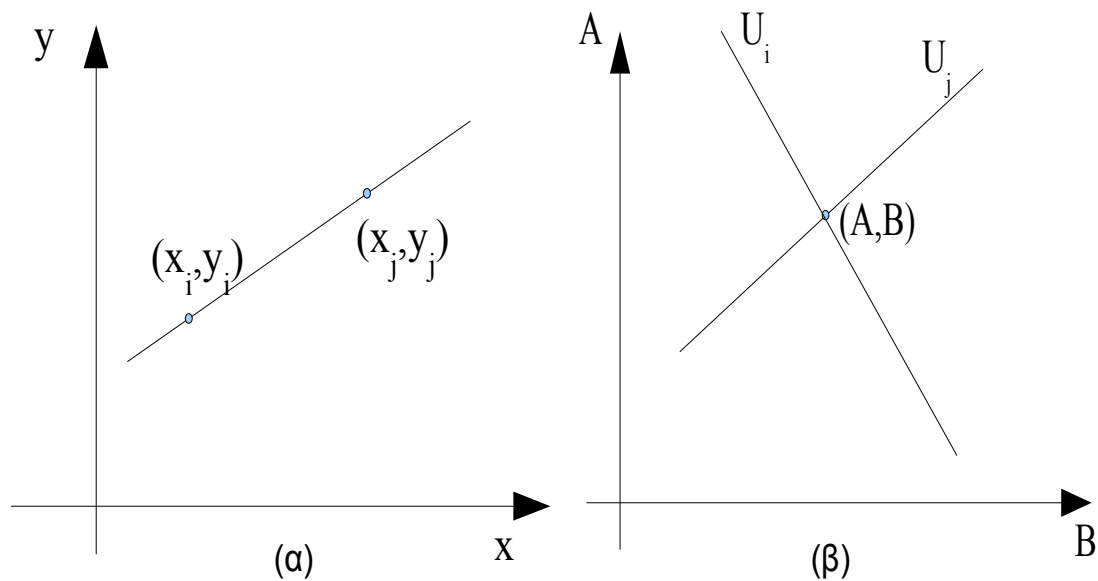
Τελικά μπορούμε να πούμε ότι ένα σύνολο συνευθειακών σημείων $\{(x_i, y_i)\}$, ορίζει την γραμμή (A,B), και έτσι ικανοποιείται η εξίσωση

$$A y_i + B x_i + 1 = 0 \quad (4-3)$$

Η παραπάνω εξίσωση μπορεί να θεωρηθεί ως ένα σύστημα εξισώσεων, και πολύ εύκολα μπορεί να μεταφερθεί πάλι στον καρτεσιανό χώρο, με βάση την σχέση

$$c = -x_i m + y_i \quad (3-4)$$

Στο σχήμα 4-5(α), φαίνονται δυο σημεία στο καρτεσιανό επίπεδο. Από καθένα από αυτά διέρχονται άπειρες ευθείες. Μόνο μια όμως για το κάθε σημείο, θα διέρχεται και από το άλλο. Τις δυο ευθείες αυτές τις ονομάζουμε U_i και U_j . Όπως είναι γνωστό από την Ευκλείδεια Γεωμετρία, οι δυο ευθείες θα ταυτίζονται. Όταν αναπαρασταθούν οι δυο αυτές ευθείες στον παραμετρικό χώρο, παρατηρούμε στο σχήμα 4-5(β) ότι τέμνονται στο σημείο (A,B). Επομένως το ζεύγος (A,B) στον παραμετρικό χώρο αρκεί για να ορίσει μια ευθεία στον καρτεσιανό χώρο.



Σχήμα 4-5: (α) Σύνολο δυο συνευθειακών σημείων που ορίζουν μια ευθεία στον καρτεσιανό χώρο.

(β) Αναπαράσταση της ευθείας στον παραμετρικό χώρο.

Με τον τρόπο αυτό, κάθε σημείο του επιπέδου $x-y$, αντιστοιχίζεται στο παραμετρικό επίπεδο $a-b$. Όμως η συγκεκριμένη αναπαράσταση, παρουσιάζει ένα σημαντικό πρόβλημα. Οι ευθείες που είναι παράλληλες στον άξονα $y'-y$ (του επιπέδου $x-y$), θεωρητικά έχουν $m = \infty$, αφού είναι κάθετες στον άξονα $x'-x$. Το πρόβλημα αυτό ξεπεράστηκε, κάνοντας χρήση των πολικών παραμέτρων της ευθείας. Έτσι η προηγούμενη αναπαράσταση αντικαταστάθηκε από την

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta \quad (4-5)$$

και το παραμετρικό επίπεδο (a,b) , από το $(\theta-\rho)$.

Η αντιστοίχιση με την προηγούμενη αναπαράσταση ευθείας, γίνεται με βάση το γεγονός ότι το γινόμενο των κλίσεων των δυο ευθειών που είναι κάθετες μεταξύ τους είναι -1 , και είναι:

$$c = \frac{\rho}{\sin(\theta)} \quad (4-6)$$

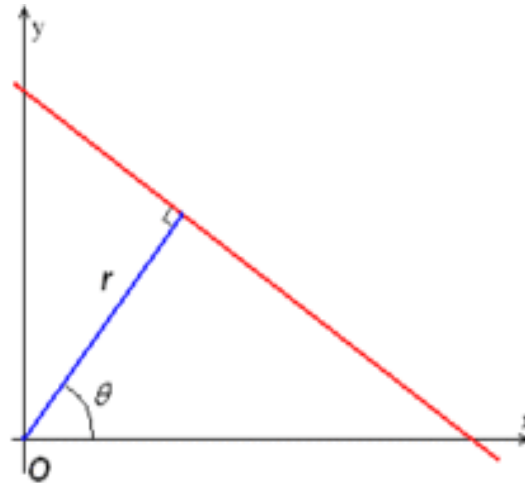
και,

$$m = -\frac{1}{\tan(\theta)} \quad (4-7)$$

Για να γίνει πιο κατανοητή η λειτουργία του μετασχηματισμού Hough, θα αναφερθούμε στο Σχήμα 4-6 που ακολουθεί.

Κάθε ευθύγραμμο τμήμα του επιπέδου, χαρακτηρίζεται μονοσήμαντα, από ένα ζεύγος αριθμών (θ, ρ) , όπου ρ είναι η κάθετη από την αρχή του συστήματος συντεταγμένων προς την ευθεία, και θ η γωνία που σχηματίζει η ρ με τον άξονα $x'-x$. Δηλαδή κάθε ευθεία στο επίπεδο $x-y$, αντιστοιχίζεται σε ένα μοναδικό σημείο στον παραμετρικό χώρο $(\theta-\rho)$. Όλα τα σημεία (x_p, y_p) που ανήκουν σε αυτήν την ευθεία ικανοποιούν τη σχέση:

$$\rho = x_p \cdot \cos\theta + y_p \cdot \sin\theta$$

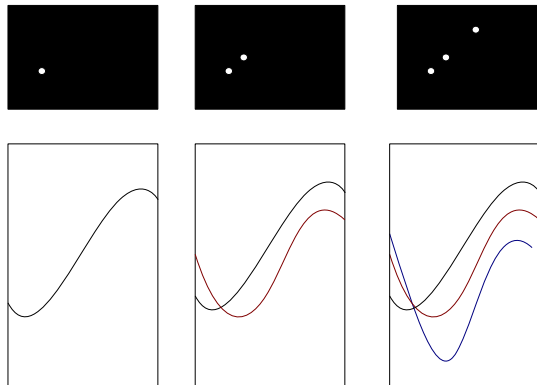


Σχήμα 4-6: Μετασχηματισμός Hough.

Όπως βλέπουμε στο Σχήμα 4-7, ένα σημείο στον Καρτεσιανό χώρο, αντιστοιχεί στον παραμετρικό χώρο σε μια καμπύλη. Επομένως ένα σύνολο συνευθειακών σημείων, αντιστοιχεί σε ένα σύνολο καμπυλών του παραμετρικού χώρου, με κοινό σημείο τομής. Οι συντεταγμένες του σημείου τομής των καμπυλών (θ, ρ) , αντιστοιχούν στα θ και ρ , της εξίσωσης:

$$\rho = x_p \cdot \cos\theta + y_p \cdot \sin\theta \quad (4-8)$$

Επομένως με βάση αυτά που αναφέρθηκαν πιο πάνω, είναι πολύ εύκολη η αναπαράσταση μιας ευθείας στον παραμετρικό χώρο $(\theta-\rho)$, και η μετάβαση πίσω στον Καρτεσιανό χώρο.



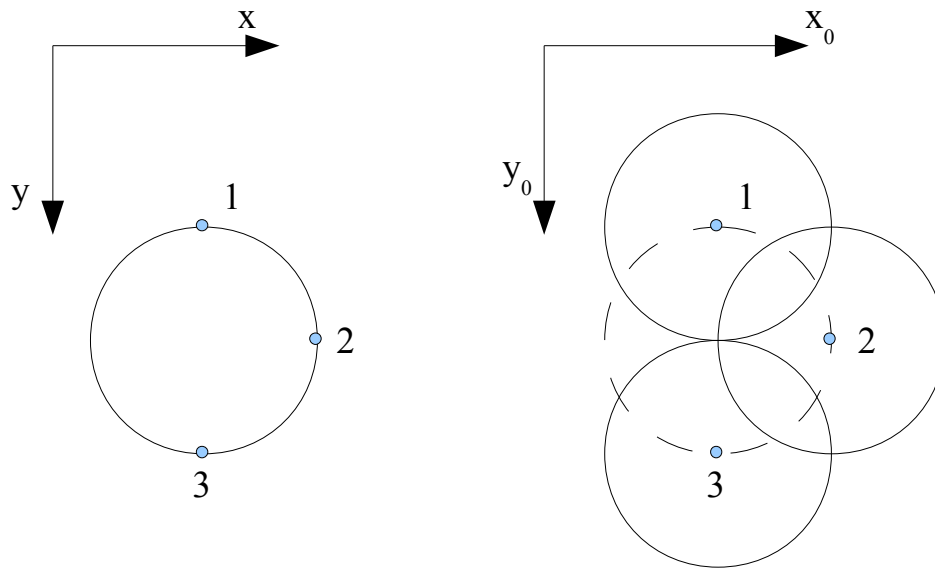
Σχήμα 4-7: Μετασχηματισμός Hough σε σύστημα πολικών συντεταγμένων.

4.4.3.2 Μετασχηματισμός Hough για Κύκλους

Ο μετασχηματισμός Hough, όπως αναφέρθηκε και προηγούμενα, μπορεί να επεκταθεί, ώστε να αναγνωρίζει και κύκλους. Η εξίσωση ενός κύκλου δίνεται από την σχέση:

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \quad (4-9)$$

Η εξίσωση αυτή ορίζει ένα γεωμετρικό τόπο σημείων (x, y) , γύρω από ένα κέντρο (x_0, y_0) και με ακτίνα r . Για την υλοποίηση του μετασχηματισμού Hough για κύκλους, χρησιμοποιείται μια διαφορετική θεώρηση της παραπάνω εξίσωσης. Δηλαδή, αντιμετωπίζεται σαν ένα σύνολο κέντρων (x_0, y_0) , γύρω από κάθε σημείο του κύκλου (x, y) , σε ακτίνα r . Στο σημείο που βρίσκεται το κέντρο του προς αναζήτηση κύκλου, θα συντρέχουν όλοι οι κύκλοι γύρω από τα σημεία (x, y) . Η διαδικασία αυτή φαίνεται στο Σχήμα 4-8, που ακολουθεί.



Σχήμα 4-8: Μετασχηματισμός Hough για κύκλους.

Όπως είναι προφανές τα παραπάνω ισχύουν για μια συγκεκριμένη τιμή της ακτίνας r . Αν αυτή δεν είναι γνωστή εκ των προτέρων, τότε η διαδικασία αυτή θα πρέπει να επαναληφθεί για το σύνολο των τιμών στις οποίες αναμένεται να βρίσκεται η τιμή της ακτίνας. Αν και το σύνολο τιμών της ακτίνας διακριτοποιείται για την διαδικασία της αναζήτησης, γίνεται αντιληπτό ότι καθώς αυξάνει η τάξη της καμπύλης που αναζητούμε, αυξάνει και η υπολογιστική ισχύς που απαιτείται.

Η εξίσωση του κύκλου μπορεί να γραφτεί σε παραμετρική μορφή ως εξής:

$$\begin{aligned} x &= x_0 + r \cdot \cos(\theta) \\ y &= y_0 + r \cdot \sin(\theta) \end{aligned} \quad (4-10)$$

Το πλεονέκτημα της παραμετρικής γραφής, είναι ότι μπορούμε να λύσουμε ως προς τις παραμέτρους. Έτσι η απεικόνιση του μετασχηματισμού Hough, γίνεται πολύ απλά ως εξής:

$$\begin{aligned} x_0 &= x - r \cdot \cos(\theta) \\ y_0 &= y - r \cdot \sin(\theta) \end{aligned} \quad (4-11)$$

Ο μετασχηματισμός Hough για κύκλους όπως και για ευθείες, παρουσιάζει πολλά πλεονεκτήματα. Μπορεί να διαχειρίζεται πολύ καλά εικόνες με μεγάλο θόρυβο, ανοιχτά κυκλικά περιγράμματα κ.α. Όμως, απαιτεί μεγάλη υπολογιστική ισχύ, ειδικά στην περίπτωση που η ακτίνα δεν είναι γνωστή, και πρέπει να ψάξει σε όλο το εύρος των πιθανών τιμών. Σε αυτό το ενδεχόμενο, ο αλγόριθμος έχει 3 άγνωστες παραμέτρους, τις οποίες πρέπει να αναζητήσει. Τις συντεταγμένες (x_0, y_0) του κέντρου του κύκλου σε όλο το εύρος της εικονας, και τις πιθανές τιμές της ακτίνας.

Αυτό έχει σαν αποτέλεσμα να μην χρησιμοποιείται, σε εφαρμογές εικονοροής, και γενικά σε εφαρμογές που απαιτείται πολύ γρήγορη επεξεργασία της εικόνας.

4.4.3.3 Μετασχηματισμός Hough για Ελλείψεις

Στην συγκεκριμένη εφαρμογή που σχεδιάζουμε, ο στόχος είναι, να αναγνωρίζουμε και να παρακολουθούμε ένα ρομποτικό ψάρι. Το σχήμα των ψαριών, μπορεί να χαρακτηριστεί περισσότερο ελλειπτικό, τουλάχιστον κατά το μεγαλύτερο μέρος του. Κατά συνέπεια θα ήταν άτοπο να χρησιμοποιούσαμε αλγόριθμο για την εύρεση κυκλικών σχημάτων. Ωστόσο, ο μετασχηματισμός Hough, έχει επεκταθεί και στην εύρεση ελλείψεων. Η διαδικασία παρουσιάζει πολλές ομοιότητες με τους κύκλους, αφού μια έλλειψη αναπαρίσταται από δυο εξισώσεις:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\rho) & \sin(\rho) \\ -\sin(\rho) & \cos(\rho) \end{bmatrix} \cdot \begin{bmatrix} S_x \\ S_y \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (4-12)$$

Στη συνέχεια, θεωρούμε ότι:

$$a_0 = t_x, \quad a_x = S_x \cdot \cos(\rho), \quad b_x = S_y \cdot \sin(\rho)$$

$$b_0 = t_y, \quad a_y = -S_x \cdot \sin(\rho), \quad b_y = S_y \cdot \cos(\rho)$$

Με βάση αυτές τις σχέσεις μπορούμε να γράψουμε την εξίσωση της έλλειψης με πολικές παραμέτρους στην εξής μορφή:

$$\begin{aligned} x &= a_0 + a_x \cos(\theta) + b_x \sin(\theta) \\ y &= b_0 + a_y \cos(\theta) + b_y \sin(\theta) \end{aligned} \quad (4-13)$$

Το σύστημα αυτών των δύο εξισώσεων δίνει την πολική αναπαράσταση της έλλειψης. Όπως βλέπουμε η έλλειψη χαρακτηρίζεται από ένα σύνολο έξι παραμέτρων ($a_0, b_0, a_x, b_x, a_y, b_y$). Το σύνολο των έξι αγνώστων παραμέτρων γίνεται τελικά πέντε αν λάβουμε υπόψη ότι οι άξονες της έλλειψης είναι κάθετοι μεταξύ τους, και επομένως ισχύει ότι:

$$a_x \cdot b_x + a_y \cdot b_y = 0 \quad (4-14)$$

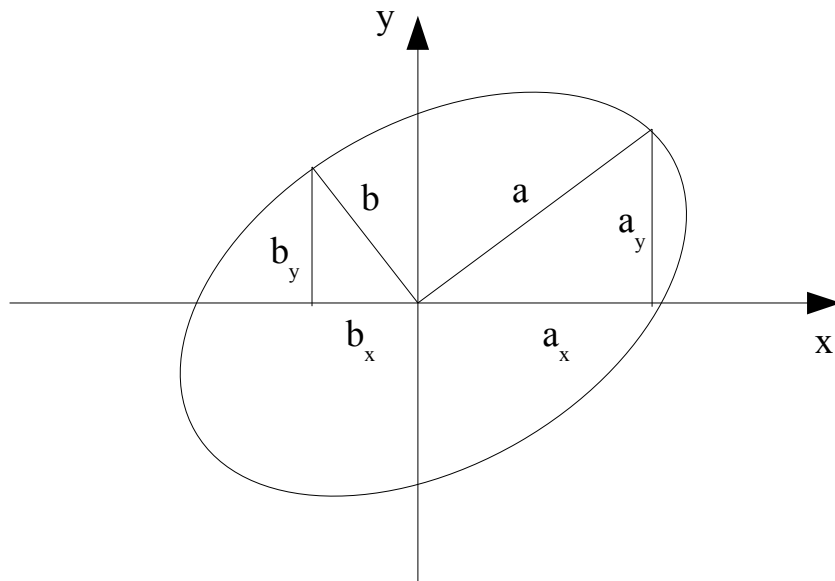
Αυτό τελικά σημαίνει ότι, μια έλλειψη, για να οριστεί αρκεί να δοθούν 5 παράμετροι. Οι δύο συντεταγμένες του κέντρου της (a_0, b_0), και τρεις από τις τέσσερις παραμέτρους των αξόνων της (a_x, b_x, a_y, b_y), κάτι που και διαισθητικά φαίνεται λογικό. Οι τρεις τελευταίες παράμετροι, καθορίζουν το μέγεθος της έλλειψης στους δύο άξονες, και τον προσανατολισμό της. Αυτό γίνεται πιο αντιληπτό στις σχέσεις που ακολουθούν. Ο προσανατολισμός της έλλειψης δίνεται από τη σχέση:

$$\tan(\rho) = \frac{a_y}{a_x} \quad (4-15)$$

ενώ το μέγεθος της πάνω στους δυο άξονες, δίνεται από τις σχέσεις:

$$a = \sqrt{a_x^2 + a_y^2}, \quad b = \sqrt{b_x^2 + b_y^2} \quad (4-16)$$

όπου (a, b), είναι οι άξονες της έλλειψης, όπως φαίνεται στο Σχήμα 4-9.



Σχήμα 4-9: Έλλειψη με τους δύο άξονες της και τις παραμέτρους αξόνων.

Γενικά θεωρείται ότι, ο μετασχηματισμός Hough για ελλείψεις, μπορεί να παρουσιάσει πάρα πολύ καλά αποτελέσματα [14]. Όπως και στους αντίστοιχους μετασχηματισμούς Hough για γραμμές και για κύκλους, είναι εξαιρετικά ανθεκτικός στον θόρυβο καθώς και σε μη συνεχή σχήματα. Όμως παρουσιάζει και κάποια σοβαρά μειονεκτήματα. Όπως αναφέρθηκε και πιο πάνω, καθώς αυξάνει η πολυπλοκότητα των καμπυλών, τόσο αυξάνει και η απαιτούμενη υπολογιστική ισχύς που απαιτείται.

Στην περίπτωση των ελλείψεων, υπάρχουν πέντε ανεξάρτητες μεταβλητές, οι οποίες πρέπει να εντοπιστούν, ώστε να οριστεί πλήρως η έλλειψη. Αν δεν υπάρχει καμμία γνωστή παράμετρος, το υπολογιστικό κόστος είναι προφανές ότι αυξάνει πολύ. Το γεγονός αυτό, κάνει τη συγκεκριμένη μέθοδο ασύμφορη, σε περιπτώσεις όπου απαιτείται πολύ γρήγορη επεξεργασία εικόνων (π.χ. Εικονοροή). Επειδή όμως ο συγκεκριμένος αλγόριθμος λειτουργεί πολύ καλά στην αναγνώριση των συγκεκριμένων γεωμετρικών σχημάτων, έχουν αναπτυχθεί μέθοδοι ώστε να μειώνεται το υπολογιστικό κόστος (π.χ. Parameter space reduction κ.α.).

4.4.3.4 Γενικευμένος Μετασχηματισμός Hough

Όπως αναφέρθηκε και στην εισαγωγή του κεφαλαίου, ο μετασχηματισμός Hough, μπορεί με τις κατάλληλες τροποποιήσεις, να αναγνωρίζει σχήματα πολύ πιο σύνθετα από γραμμές, κύκλους και ελλείψεις. Το σκεπτικό είναι ότι, οποιοδήποτε τυχαίο σχήμα, μπορεί να αναλυθεί σε επιμέρους βασικά γεωμετρικά σχήματα, όπως ευθείες, τμήματα κύκλων και ελλείψεις. Η μέθοδος αυτή,

ονομάζεται γενικευμένος μετασχηματισμός Hough, και μπορεί να εντοπίζει τυχαία σχήματα, με άγνωστη θέση, μέγεθος και προσανατολισμό. Θα μπορούσε κανείς να πει ότι μοιάζει με την μέθοδο ταυτοποίησης προτύπων που έχει αναφερθεί πιο πάνω. Όμως χρησιμοποιώντας κατάλληλες τεχνικές, καταφέρνει να μειώσει σημαντικά το υπολογιστικό κόστος που απαιτείται για μια τέτοια διαδικασία.

Γενικά, η καμπύλη που περιγράφει ένα τυχαίο σχήμα μπορεί να περιγραφεί στην γενική περίπτωση από τη σχέση:

$$v(\theta) = x(\theta) \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y(\theta) \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (4-17)$$

Για παράδειγμα στην περίπτωση του κύκλου είναι:

$$\begin{aligned} x(\theta) &= r \cdot \cos(\theta) \\ y(\theta) &= r \cdot \sin(\theta) \end{aligned} \quad (4-18)$$

Για την εύρεση του σχήματος σε μια εικόνα, λαμβάνοντας υπόψη την μετατόπιση, την περιστροφή και την κλίμακα, θεωρούμε την παρακάτω σχέση:

$$\omega(\theta, b, \lambda, \rho) = b + \lambda \cdot \mathbf{R}(\rho) \cdot v(\theta) \Rightarrow b = \omega - \lambda \cdot \mathbf{R}(\rho) \cdot v(\theta) \quad (4-19)$$

Επιπλέον απαιτούμε έναν ακόμα περιορισμό. Ο περιορισμός αυτός, θα μας επιτρέψει να συνταιριάσουμε τα σημεία της εικόνας, με τα σημεία της εικόνας-πρότυπου. Ο περιορισμός αυτός βασίζεται στην κατεύθυνση της κλίσης. Αυτό επιλέγεται, από το γεγονός, ότι δεν γνωρίζουμε το ακριβές σχήμα αλλά διάφορα σημεία του σχήματος. Έτσι επιλέγεται από τον αλγόριθμο η συγκεκριμένη ιδιότητα, για να μας παρέχει μια ένδειξη ταύτισης. Έτσι:

$$\varphi'(\theta) = \frac{y'(\theta)}{x'(\theta)} \quad (4-20)$$

και

$$\hat{\varphi}'(\theta) = \tan^{-1}(\varphi'(\theta)) \quad (4-21)$$

Για να υπάρξει ανίχνευση του στόχου, ανεξάρτητα από τον προσανατολισμό, θα πρέπει η κατεύθυνση του διανύσματος κλίσης σε κάθε σημείο της εικόνας να ταιριάζει με αυτήν του αντίστοιχου σημείου, στην (ενδεχομένως στραμμένη) εικόνα-πρότυπο. Δηλαδή, $\varphi'_i = \hat{\varphi}'(\theta) - \rho$, όπου $\hat{\varphi}'_i$ είναι η γωνία στο αντίστοιχο σημείο ω_i . Όπως είναι κατανοητό, η κατεύθυνση του διανύσματος κλίσης, είναι ανεξάρτητη από την κλίμακα της εικόνας, ενώ μεταβάλλεται με τον ίδιο ρυθμό, όπως η περιστροφή της εικόνας.

Στη συνέχεια, χρησιμοποιούμε τις πολικές συντεταγμένες, για την συνέχιση της ανάλυσης, και γράφουμε την παραπάνω σχέση σε μια μορφή που συνδυάζει την περιστροφή και την κλίμακα:

$b = \omega(\theta) - \gamma(\lambda, \rho)$, όπου $\gamma^T(\lambda, \rho) = [\gamma_x(\lambda, \rho) \gamma_y(\lambda, \rho)]$, και η συνδυασμένη περιστροφή και κλίμακα, δίνεται από τις σχέσεις:

$$\begin{aligned}\gamma_x(\lambda, \rho) &= \lambda \cdot (x(\theta) \cdot \cos(\rho) - y(\theta) \cdot \sin(\rho)) \\ \gamma_y(\lambda, \rho) &= \lambda \cdot (x(\theta) \cdot \sin(\rho) + y(\theta) \cdot \cos(\rho))\end{aligned}\quad (4-22)$$

Με τον τρόπο αυτό ορίζεται ένα διάνυσμα $\gamma(\lambda, \rho)$, η εφαπτομένη γωνία και το πλάτος του οποίου δίνονται από τις σχέσεις:

$$\tan(\alpha) = \frac{\gamma_y(\lambda, \rho)}{\gamma_x(\lambda, \rho)} \quad (4-23)$$

και

$$r = \sqrt{\gamma_x^2(\lambda, \rho) + \gamma_y^2(\lambda, \rho)} \quad (4-24)$$

Επομένως στη συνέχεια, θα πρέπει να υπολογίσουμε τις τιμές για τα α και r . Μετά από αλγεβρικές πράξεις καταλήγουμε στα εξής:

$$\alpha = \varphi(\theta) + \rho \quad (4-25)$$

και

$$r = \lambda \cdot \Gamma(\theta) \quad (4-26)$$

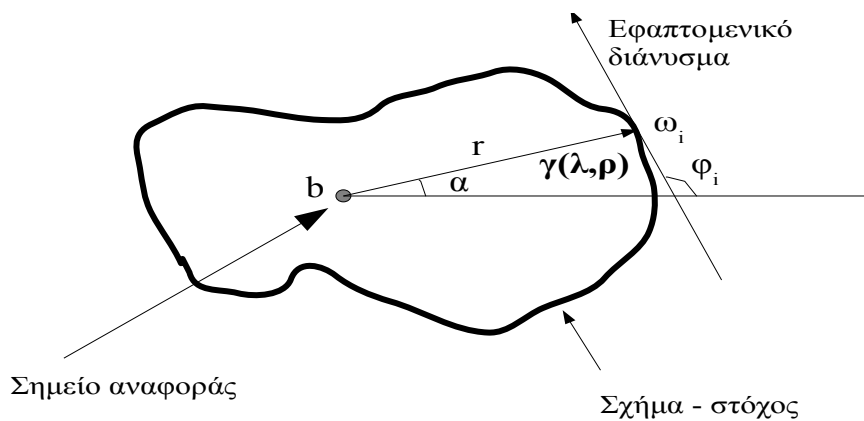
όπου:

$$\varphi(\theta) = \tan^{-1} \left(\frac{y(\theta)}{x(\theta)} \right) \quad (4-27)$$

και

$$\Gamma(\theta) = \sqrt{x(\theta)^2 + y(\theta)^2} \quad (4-28)$$

Τα μεγέθη αυτά φαίνονται στο Σχήμα 4-10, που ακολουθεί:



Σχήμα 4-10: Γεωμετρικά μεγέθη γενικευμένου μετασχηματισμού Hough.

Ύστερα από τα ανωτέρω, αρκεί να βρούμε το $\nu(\theta)$. Γνωρίζοντας το $\hat{\phi}'_i$, αρκεί να βρούμε εκείνο το σημείο $\nu(\theta)$, του οποίου η κατεύθυνση του διανύσματος κλίσης είναι $\hat{\phi}'_i + \rho = 0$. Στη συνέχεια πρέπει να χρησιμοποιήσουμε το $\nu(\theta)$, για να βρούμε το διάνυσμα μετατόπισης $\gamma(\lambda, \rho)$. Ο αλγόριθμος του γενικευμένου μετασχηματισμού Hough, υπολογίζει αρχικά την λύση αυτού του προβλήματος, και τοποθετεί τα αποτελέσματα σε έναν πίνακα, που ονομάζεται πίνακας R (R-table). Ο πίνακας R αποθηκεύει το διάνυσμα $\gamma(\lambda, \rho)$, για κάθε τιμή του $\hat{\phi}'_i$, για $\rho = 1$ και $\lambda = 1$. Η αναπαράσταση μπορεί να γίνει σε πολικές συντεταγμένες, όπου το διάνυσμα παριστάνεται σαν ζεύγος μέτρου και γωνίας. Ακόμα μπορεί να γίνει και σε Καρτεσιανές συντεταγμένες, όπου το διάνυσμα παριστάνεται σαν ζεύγος συντεταγμένων.

Η διαδικασία σχηματισμού του πίνακα R, περιγράφεται στην συνέχεια. Το εύρος τιμών για την γωνία $\hat{\phi}'_i$ είναι από $-\pi/2$, έως $\pi/2$ rad. Το εύρος αυτό χωρίζεται σε N ίσα τμήματα, τα οποία αποτελούν τις σειρές των δεδομένων στον πίνακα R. Η κατεύθυνση της ακμής σε κάθε σημείο του περιγράμματος, καθορίζει την κατάλληλη γραμμή στον πίνακα. Το μήκος r , και η κατεύθυνση α , από το σημείο αναφοράς, εισέρχονται στον πίνακα σαν στοιχεία στήλης, στην συγκεκριμένη γραμμή, για κάθε σημείο περιγράμματος του σχήματος. Με αυτόν τον τρόπο οι N γραμμές του πίνακα R, περιέχουν στοιχεία με πληροφορίες σχετικές με το περίγραμμα του σχήματος. Το μέγεθος κάθε σειράς του πίνακα, καθορίζεται από τον αριθμό των στοιχείων που έχουν κλίση ίση με αυτή που περιγράφεται στην συγκεκριμένη σειρά. Για την καλύτερη κατανόηση του σχηματισμού του πίνακα R, δίνεται ένας ενδεικτικός που θα μπορούσε να αφορά το προηγούμενο σχήμα.

Πίνακας 4-1: Πίνακας R - Γενικευμένος Μετασχηματισμός Hough.

$\hat{\phi}'_i$	$\gamma(r, \alpha)$
0	$(r_0, \alpha_0), (r_1, \alpha_1), (r_2, \alpha_2)$
$\Delta\phi$	$\vdots$
$2\Delta\phi$	$\vdots$
....	

Με τον τρόπο αυτό σαρώνονται όλα τα σημεία της εικόνας, ώστε να βρεθούν εκείνα στα οποία το διάνυσμα κλίσης είναι παρόμοιο με το ζητούμενο. Η ισότητα των διανυσμάτων κλίσης καθορίζεται από μια τιμή κατωφλίου (threshold). Στη συνέχεια, γίνεται έρευνα ώστε να βρεθεί αν τα σημεία αυτά παρουσιάζουν τις ίδιες ιδιότητες, με αυτές που περιγράφονται στον πίνακα R.

Ο αλγόριθμος παρουσιάζει εξαιρετικά αποτελέσματα στην αναγνώριση αντικειμένων οποιουδήποτε τυχαίου σχήματος. Όπως όλοι οι αντίστοιχοι μετασχηματισμοί για τα υπόλοιπα σχήματα που αναφέρθηκαν πιο πάνω, παρουσιάζει εξαιρετική αντοχή σε θόρυβο. Ακόμα, μπορεί

να αναγνωρίζει αντικείμενα που δεν είναι ορατά στο συνολό τους, είτε λόγω θορύβου, είτε στην περίπτωση που καλύπτονται σε ένα μέρος τους από άλλα αντικείμενα. Τέλος, μπορεί σε μεγάλο βαθμό να αναγνώριζε τα αντικείμενα ακόμα και ύστερα από περιστροφή, ή αλλαγή της κλίμακας τους (μεγέθυνση ή σμίκρυνση).

Όμως υπάρχουν και κάποια σημαντικά μειονεκτήματα. Ένα σημαντικό πρόβλημα, είναι ότι λόγω της διακριτοποίησης που κάνει στα σχήματα, δεν μπορεί να τα αναπαραγάγει με ακρίβεια. Λόγω αυτού, ακόμα μεγαλύτερο πρόβλημα υπάρχει στις περιπτώσεις όπου τα σχήματα που αναζητά, έχουν περιστραφεί, ή έχουν αλλάξει την κλίμακα τους. Ένα επιπλέον σημαντικό πρόβλημα, που γίνεται εμφανές από την ανάλυση που προηγήθηκε, είναι ο μεγάλος αριθμός πράξεων που απαιτούνται. Κατά συνέπεια απαιτείται μεγάλη υπολογιστική ισχύς. Ένας ακόμα παράγοντας αναξιοπιστίας του αλγορίθμου, είναι η μεγάλη εξάρτηση που έχει από το διάνυσμα κλίσης. Όλα αυτά κατατείνουν στην μη χρησιμοποίηση του συγκεκριμένου αλγορίθμου για αναγνώριση αντικειμένων τυχαίων σχημάτων. Πολλές προτάσεις για βελτίωση του, έχουν γίνει, όμως σε καμία περίπτωση δεν υπερνικούνται τα βασικά προβλήματα που παρουσιάζει ο γενικευμένος μετασχηματισμός Hough.

4.4.4 Αναγνώριση Αντικειμένων στο Πεδίο της Συχνότητας

Μια πολύ σημαντική μέθοδος αναγνώρισης αντικειμένων, είναι η μεταφορά των σημάτων στο πεδίο της συχνότητας. Η υλοποίηση αυτής της μεθόδου γίνεται με την βοήθεια του μετασχηματισμού Fourier. Στη συνέχεια θα αναφερθούμε, συνοπτικά στον μετασχηματισμό Fourier, στον διακριτό και στον διδιάστατο μετασχηματισμό Fourier.

Η σχέση που ορίζει τον μετασχηματισμό Fourier Fp , ενός σήματος p , είναι:

$$Fp(\omega) = \int_{-\infty}^{+\infty} p(t) \cdot e^{-j\omega t} dt \quad (4-29)$$

όπου $Fp(\omega)$ είναι ο μετασχηματισμός Fourier, ω η γωνιακή συχνότητα ($\omega = 2\pi f$) σε rad/s, $p(t)$ είναι συνεχές σήμα, και

$$e^{-j\omega t} = \cos(\omega t) - j \cdot \sin(\omega t) \quad (4-30)$$

η εξίσωση που δίνει τις συνιστώσες της συχνότητας του σήματος $p(t)$. Το αποτέλεσμα του μετασχηματισμού Fourier, είναι ένας σύνθετος αριθμός. Έτσι μπορεί να αναπαρασταθεί με την βοήθεια του πλάτους και της γωνίας, ως εξής:

$$\int_{-\infty}^{+\infty} p(t) \cdot e^{-j\omega t} dt = \Re[Fp(\omega)] + j \cdot \Im[Fp(\omega)] \quad (4-31)$$

όπου $\Re[Fp(\omega)]$ και $\Im[Fp(\omega)]$ το πραγματικό και το φανταστικό μέρος του μετασχηματισμού, αντίστοιχα. Κατά συνέπεια το πλάτος του μετασχηματισμού είναι

$$\left| \int_{-\infty}^{+\infty} p(t) \cdot e^{-j\omega t} dt \right| = \sqrt{\Re[Fp(\omega)]^2 + \Im[Fp(\omega)]^2} \quad (4-32)$$

και η γωνία (φάση):

$$\angle \int_{-\infty}^{+\infty} p(t) \cdot e^{-j\omega t} dt = \tan^{-1} \frac{\Im[Fp(\omega)]}{\Re[Fp(\omega)]} \quad (4-33)$$

Το πλάτος του μετασχηματισμού, υποδηλώνει την "ποσότητα" από κάθε συνιστώσα της συχνότητας που περιέχεται στο σήμα, ενώ η φάση δείχνει το πότε αυτή συμβαίνει. Το μετασχηματισμένο, στο πεδίο της συχνότητας, σήμα μπορεί να επανέλθει στο πεδίο του χρόνου, με τον αντίστροφο μετασχηματισμό Fourier, ο οποίος δίνεται στην επόμενη σχέση:

$$p(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} Fp(\omega) \cdot e^{-j\omega t} d\omega \quad (4-34)$$

Στην όλη διαδικασία μεταφοράς του σήματος από το πεδίο του χρόνου προς το πεδίο της συχνότητας και αντίστροφα, σημαντικό ρόλο παίζει το κριτήριο δειγματοληψίας του Nyquist. Σύμφωνα με αυτό, για να είναι αξιόπιστη η ανακατασκευή ενός σήματος από τα δείγματα του, θα πρέπει η συχνότητα δειγματοληψίας να είναι τουλάχιστον διπλάσια από τη μέγιστη συχνότητα που εμφανίζεται στο προς δειγματοληψία σήμα. Σε αντίθετη περίπτωση, υπάρχει μεγάλη απώλεια πληροφορίας κατά τον μετασχηματισμό Fourier. Επομένως τα αποτελέσματα δεν μπορούν να θεωρηθούν αξιόπιστα.

Όμως οι εικόνες, όπως και πολλά άλλα δεδομένα, είναι διακριτά και όχι συνεχή σήματα. Επομένως χρειάζεται μια μορφή του μετασχηματισμού Fourier, που να μπορεί να χειρίζεται τέτοια δεδομένα. Αυτή η μορφή του μετασχηματισμού Fourier, ονομάζεται διακριτός μετασχηματισμός Fourier. Έτσι, θεωρούμε ένα σύνολο N διακριτών σημείων p_x . Ακόμα υποθέτουμε ότι η δειγματοληψία έχει γίνει σύμφωνα με το θεώρημα Nyquist. Τότε ο διακριτός μετασχηματισμός Fourier (DFT-Discrete Fourier Transform), δίνεται από την σχέση:

$$Fp_u = \frac{1}{\sqrt{N}} \cdot \sum_{x=0}^{N-1} p_x \cdot e^{-j\left(\frac{2\pi}{N}\right)xu} \quad (4-35)$$

Αυτό είναι το διακριτό ανάλογο του μετασχηματισμού Fourier για συνεχή σήματα. Όπως φαίνεται το συνεχές σήμα αντικαθιστάται από ένα σύνολο δειγμάτων, οι συνεχείς συχνότητες από δείγμα, ενώ το ολοκλήρωμα αντικαθιστάται από ένα άθροισμα. Για να επανέλθει το σήμα πάλι στο πεδίο του χρόνου, γίνεται χρήση του αντίστροφου διακριτού μετασχηματισμού Fourier, ο οποίος δίνεται στη συνέχεια:

$$p_x = \sum_{u=0}^{N-1} Fp_u \cdot e^{j\frac{2\pi}{N}ux} \quad (4-36)$$

Στην περίπτωση της εικόνας, το σήμα που πρέπει να επεξεργαστεί είναι διδιάστατο. Κατά

συνέπεια, θέλουμε ένα διδιάστατο διακριτό μετασχηματισμό Fourier. Επομένως έχουμε δύο διαστάσεις συχνοτήτων u και v , μια οριζόντια και μια κάθετα. Ο διδιάστατος διακριτός μετασχηματισμός Fourier, υπολογίζει τα δεδομένα των συχνοτήτων $Fp_{u,v}$ από μια εικόνα $N \times N$ εικονοστοιχείων, με βάση την σχέση:

$$Fp_{u,v} = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} P_{x,y} \cdot e^{-j(\frac{2\pi}{N})(ux+vy)} \quad (4-37)$$

Ο αντίστροφος διδιάστατος διακριτός μετασχηματισμός Fourier, που επαναφέρει το σήμα στο πεδίο του χρόνου δίνεται από την παρακάτω σχέση:

$$P_{x,y} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} Fp_{u,v} \cdot e^{j(\frac{2\pi}{N})(ux+vy)} \quad (4-38)$$

Στη συνέχεια, θα αποφύγουμε να αναφερθούμε σε έκταση στον μετασχηματισμό Fourier, και στις ιδιότητες του, καθώς κάτι τέτοιο ξεφεύγει από τους στόχους της παρούσας εργασίας. Θα αναφερθούμε, όμως στις ιδιότητες του εκείνες, που τον καθιστούν ένα εξαιρετικό εργαλείο, στην επεξεργασία εικόνων.

Η πράξη του πολλαπλασιασμού μεταξύ δυο σημάτων στο πεδίο του χρόνου, αντιστοιχεί στο πεδίο της συχνότητας, στην πράξη της συνέλιξης (convolution). Κατά συνέπεια μπορούμε να πούμε ότι υπάρχει ένας δυϊσμός αυτών των δυο πράξεων στα δύο πεδία. Η συνέλιξη, ορίζεται από την εξής πράξη:

$$p_1(t) * p_2(t) = \int_{-\infty}^{+\infty} p_1(\tau) \cdot p_2(t-\tau) d\tau \quad (4-39)$$

Η σχέση αυτή αποτελεί την βάση της θεωρίας συστημάτων, όπου η έξοδος ενός συστήματος είναι η συνέλιξη μιας διέγερσης, έστω p_1 , και της απόκρισης του συστήματος p_2 .

Επιπλέον, η πράξη της συσχέτισης (correlation), ορίζεται ως εξής:

$$p_1(t) \odot p_2(t) = \int_{-\infty}^{+\infty} p_1(\tau) \cdot p_2(t+\tau) d\tau \quad (4-40)$$

Η πράξη της συσχέτισης, στην ουσία δίνει ένα μέτρο της ομοιότητας των δυο σημάτων, p_1 και p_2 . Όταν είναι $p_1(\omega) = p_2(\omega)$, τότε στην ουσία συσχετίζουμε ένα σήμα με τον εαυτό του, διαδικασία η οποία ονομάζεται αυτοσυσχέτιση (autocorrelation).

Για την αναγνώριση αντικειμένων, θα χρησιμοποιήσουμε τις ιδιότητες του μετασχηματισμού Fourier, που αναφέρθηκαν. Δηλαδή, τον δυϊσμό της συνέλιξης και του πολλαπλασιασμού. Επιπλέον, όπως θα δούμε στη συνέχεια υπάρχει άμεση σχέση της πράξης της συνέλιξης και της συσχέτισης. Ο πολλαπλασιασμός στο πεδίο του χρόνου, όπως έχουμε αναφέρει, ισοδυναμεί με συνέλιξη στο πεδίο της συχνότητας. Το γεγονός αυτό, επιτρέπει να ξεπεραστεί το βασικό πρόβλημα των αντίστοιχων αλγόριθμων αναζήτησης αντικειμένων. Το πρόβλημα αυτό

είναι η αυξημένη υπολογιστική ισχύς. Έτσι, αν μεταφερθούν τα σήματα στο πεδίο της συχνότητας, η επίλυση του προβλήματος, ανάγεται στην εκτέλεση απλούστερων πράξεων. Μειώνεται έτσι σημαντικά το υπολογιστικό φορτίο. Δηλαδή, για να βρούμε ένα σχήμα, αρκεί να υπολογίσουμε την συσχέτιση, σαν πολλαπλασιασμό στο πεδίο της συχνότητας. Όμως έχουμε αναφέρει ότι το αντίστοιχο του πολλαπλασιασμού, είναι η συνέλιξη και όχι η συσχέτιση. Επομένως θα πρέπει να εκφράσουμε την συσχέτιση (correlation), σε όρους συνέλιξης (convolution). Στη συνέχεια δίνεται, το πως μπορεί αυτό να γίνει. Αρχικά ξαναγράφουμε τον ορισμό της συσχέτισης ως εξής:

$$I \odot T = \sum_{x, y \in W} I_{x', y'} \cdot T_{x'-i, y'-j} \quad (4-41)$$

όπου $x' = x + i$ και $y' = y + j$. Η συνέλιξη ορίζεται ως:

$$I * T = \sum_{x, y \in W} I_{x', y'} \cdot T_{i-x', j-y'} \quad (4-42)$$

Επομένως μπορούμε να γράψουμε

$$I \odot T = I * T' = \sum_{x, y \in W} I_{x', y'} \cdot T'_{i-x', j-y'} \quad (4-43)$$

όπου: $T' = T_{-x, -y}$

Δηλαδή, η συσχέτιση είναι ισοδύναμη με την συνέλιξη, όταν το σήμα-πρότυπο αλλάζει σύμφωνα με την τελευταία σχέση. Η εξίσωση αυτή, αντιστρέφει το σύστημα συντεταγμένων, και ταυτόχρονα, αντιμεταθέτει τον οριζόντιο και τον κάθετο άξονα.

Έτσι, για τον εντοπισμό ενός αντικειμένου, η διαδικασία, συνοψίζεται στα επόμενα βήματα:

- Μετασχηματίζουμε την εικόνα του αντικειμένου-πρότυπο (το αντικείμενο που αναζητούμε - template), και την εικόνα στην οποία θα ψάξουμε, στο πεδίο της συχνότητας, με την βοήθεια του κατάλληλου μετασχηματισμού Fourier (διδιάστατος, διακριτός, ταχύς μετασχηματισμός Fourier).
- Εκτελούμε την πράξη της συσχέτισης, στο πεδίο της συχνότητας, δηλαδή τον πολλαπλασιασμό των μετασχηματισμένων σημάτων, σημείο προς σημείο.
- Μεταφέρουμε το αποτέλεσμα που προκύπτει, στο πεδίο του χρόνου, κάνοντας χρήση του αντίστροφου μετασχηματισμού. Το αποτέλεσμα που θα προκύψει, δίνει τα σημεία, όπου βρέθηκε ταύτιση, μεταξύ των δυο εικόνων.

Όπως φαίνεται από τα παραπάνω, η όλη διαδικασία εντοπισμού, απλοποιείται πολύ, με την συγκεκριμένη μεθοδολογία. Μπορούμε να πούμε ότι, συνοψίζεται στην μεταφορά στο πεδίο συχνότητας, σε έναν πολλαπλασιασμό και επαναφορά στο πεδίο του χρόνου. Με τις δεδομένες ικανότητες των σημερινών υπολογιστών, ένας τέτοιος αλγόριθμος, μπορεί να εκτελεστεί, τάχιστα και χωρίς προβλήματα, ακόμα και σε έναν φορητό υπολογιστή.

Στη συνέχεια θα δούμε ένα παράδειγμα υπολογισμού συσχέτισης, σε ένα σημείο ενός πίνακα. Έστω ότι έχουμε τον εξής πίνακα A:

$$A = \begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{bmatrix}$$

και τον πίνακα h:

$$h = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

Το αποτέλεσμα της συσχέτισης των δυο πινάκων, θα είναι ένας νέος πίνακας, διαστάσεων ίσων με τις διαστάσεις του πίνακα A.

Στη συνέχεια θα υπολογίσουμε ενδεικτικά το στοιχείο (2,4) του πίνακα εξόδου. Όλα τα στοιχεία του πίνακα εξόδου προκύπτουν με τον ίδιο τρόπο. Αρχικά μετακινούμε το κεντρικό στοιχείο του πίνακα συσχέτισης h, έτσι ώστε να βρίσκεται πάνω στο στοιχείο (2,4) του πίνακα A. Έπειτα πολλαπλασιάζουμε κάθε στοιχείο του πίνακα h, με τα αντίστοιχα στοιχεία του A, πάνω στα οποία βρίσκονται. Τέλος, αθροίζουμε τα ανεξάρτητα γινόμενα που προέκυψαν στο προηγούμενο βήμα, όπως φαίνεται και στον πίνακα που ακολουθεί:

$$\begin{bmatrix} 17 & 24 & 1^8 & 8^1 & 15^6 \\ 23 & 5 & 7^3 & 14^5 & 16^7 \\ 4 & 6 & 13^4 & 20^9 & 22^2 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{bmatrix}$$

Επομένως το στοιχείο (2,4) του πίνακα εξόδου, που θα προκύψει σαν αποτέλεσμα της συσχέτισης των πινάκων A και h, είναι:

$$1 \cdot 8 + 8 \cdot 1 + 15 \cdot 6 + 7 \cdot 3 + 14 \cdot 5 + 16 \cdot 7 + 13 \cdot 4 + 20 \cdot 9 + 22 \cdot 2 = 585$$

Το αποτέλεσμα του πολλαπλασιασμού στα μετασχηματισμένα σήματα, είναι ο βαθμός ομοιότητας, που έχει το πρότυπο και η εικόνα σε κάθε σημείο. Δηλαδή, όσο πιο όμοιοι είναι οι δυο πίνακες σε ένα σημείο τους, τόσο μεγαλύτερο θα είναι το αποτέλεσμα της συσχέτισης τους, στο συγκεκριμένο στοιχείο. Κατά συνέπεια, μπορεί να εντοπιστούν διάφορα σημεία σε μια εικόνα, που να μοιάζουν με το πρότυπο, άλλα περισσότερο και άλλα λιγότερο. Με την χρήση μια τιμής κατωφλίου (threshold), μπορούμε να απομονώσουμε μόνο αυτά που μοιάζουν με το πρότυπο από ένα βαθμό και πάνω. Αν θέλουμε να βρούμε το σημείο με την μεγαλύτερη ομοιότητα, αρκεί να βρούμε σε ποιο σημείο ο πολλαπλασιασμός δίνει το μέγιστο αποτέλεσμα. Δηλαδή, ο αλγόριθμος, εκτός από την ταχύτητα, είναι σε θέση να παρέχει πολύ εύκολα, πολύ διαχειρίσιμα αποτελέσματα. Αυτά μπορούν, είτε να επεξεργασθούν περαιτέρω, είτε να οπτικοποιηθούν πολύ εύκολα, και να έχουν μια άμεση φυσική ερμηνεία. Επιπλέον, ο αλγόριθμος παρουσιάζει πολύ καλά αποτελέσματα

σε εικόνες με θόρυβο, καθώς και σε περιπτώσεις όπου δεν υπάρχει ομοιόμορφος φωτισμός.

Αν και η μεθοδολογία μεταφοράς στο πεδίο της συχνότητας, είναι μια τεχνική που χρησιμοποιείται ευρέως, στην επεξεργασία εικόνας και στις εφαρμογές όρασης μηχανής (computer vision), παρουσιάζει και κάποια σημαντικά μειονεκτήματα. Ένα σοβαρό μειονέκτημα είναι και σε αυτή την περίπτωση, τα σφάλματα στην ακρίβεια που προκύπτουν από την διακριτοποίηση της εικόνας. Αυτό γίνεται ακόμα μεγαλύτερο αν η εικόνα βρίσκεται υπό κλίμακα. Στην περίπτωση αυτή ένας τέτοιος αλγόριθμος δεν παρουσιάζει την επιθυμητή συμπεριφορά. Όμως στην εφαρμογή που θέλουμε να υλοποιήσουμε αυτό δεν αποτελεί πρόβλημα, καθώς θεωρούμε ότι ο στόχος δεν αλλάζει σημαντικά την αποσταση του από την κάμερα. Ακόμα όμως και αν άλλαζε την αποσταση του, υπάρχουν διάφοροι τρόποι ώστε προγραμματιστικά να ξεπεραστεί αυτό το πρόβλημα, η περιγραφή των οποίων ξεφεύγει από τους στόχους της παρούσας εργασίας. Ένα ακόμα πρόβλημα, που προκύπτει από την λειτουργία του ίδιου του αλγορίθμου, είναι ότι δεν μπορεί να αναγνωρίσει αντικείμενα που έχουν περιστραφεί. Αυτό συμβαίνει γιατί ένα σχήμα που έχει περιστραφεί, δίνει τελείως διαφορετικό σήμα στο πεδίο της συχνότητας. Επειδή θέλουμε στην συγκεκριμένη εφαρμογή ο στόχος να περιστρέφεται χωρίς προβλήματα, θα περιγραφεί στη συνέχεια ένας τρόπος αντιμετώπισης του προβλήματος αυτού.

4.4.5 Διαδικασία Επιλογής Αλγορίθμου Επεξεργασίας Εικόνας

Προηγούμενα αναφέρθηκαν οι σημαντικότεροι αλγόριθμοι για την αναγνώριση αντικειμένων. Παρουσιάστηκαν τα σημαντικότερα στοιχεία τους, τα πλεονεκτήματα και τα μειονεκτήματα. Στη συνέχεια, και έχοντας αυτά υπόψη, θα προχωρήσουμε στην διαδικασία επιλογής της κατάλληλης μεθόδου για την εφαρμογή μας. Στον πίνακα 4-2 που ακολουθεί φαίνονται συνοπτικά τα σημαντικότερα στοιχεία κάθε μεθόδου που θα μας οδηγήσουν στην τελική επιλογή.

Πίνακας 4-2: Πίνακας Επιλογής Μεθόδου Επεξεργασίας Εικόνας.

Μεθοδολογία	Καταλληλότητα για το σχήμα-στόχο της εφαρμογής	Ικανότητα σε εικόνες με ασταθή φωτισμό	Καταλληλότητα για εφαρμογές εικονοροής	Απαιτούμενη Υπολογιστική Ισχύς	Συνολική Αξιολόγηση για την συγκεκριμένη εφαρμογή
Ταυτοποίηση Εικόνων	Κατάλληλη	Μέτρια	Πολύ Κακή	Πολύ Μεγάλη	X
Μετασχηματισμός Hough για γραμμές	Κακή	Μέτρια	Πολύ Καλή	Μικρή	X
Μετασχηματισμός Hough για κύκλους	Κακή	Μέτρια	Μέτρια	Αυξημένη	X
Μετασχηματισμός Hough για ελλείψεις	Αρκετά Καλή	Μέτρια	Πολύ Κακή	Πολύ Μεγάλη	X
Γενικευμένος Μετασχηματισμός Hough	Άριστη	Μέτρια	Πολύ Κακή	Μεγάλη	X
Μεταφορά στο Πεδίο της Συχνότητας	Άριστη	Πολύ καλή	Πολύ Καλή	Μικρή	✓

Όπως φαίνεται στον παραπάνω πίνακα, τελικά επιλέχθηκε η μεταφορά στο πεδίο της συχνότητας, ως μέθοδος επεξεργασίας εικόνας για την εφαρμογή μας. Τα πλεονεκτήματα της μεθόδου, αναφέρθηκαν προηγουμένως. Επιπλέον, έγινε προσπάθεια κατά την δημιουργία του κώδικα, να γραφτεί με τέτοιο τρόπο, ώστε να ξεπεραστούν και κάποιες από τις εγγενείς αδυναμίες της μεθόδου. Όλα τα χαρακτηριστικά του αλγορίθμου, αναφέρονται στο επόμενο κεφάλαιο, όπου περιγράφεται αναλυτικά η λειτουργία του και όλα του τα χαρακτηριστικά.

4.5 Περιγραφή Αλγορίθμου Επεξεργασίας Εικόνων

Στο παρόν κεφάλαιο, περιγράφεται αναλυτικά ο αλγόριθμος. Ο κώδικας είναι γραμμένος σε περιβάλλον MATLAB. Η βασική αρχή λειτουργίας στηρίζεται στην μεταφορά των σημάτων στο πεδίο της συχνότητας, με την βοήθεια του μετασχηματισμού Fourier. Τα μετασχηματισμένα σήματα επεξεργάζονται με βάση την πράξη της συσχέτισης (correlation), όπως αναφέρθηκε πιο πάνω. Τα αποτελέσματα της επεξεργασίας επαναφέρονται στο πεδίο του χρόνου με την βοήθεια του αντίστροφου μετασχηματισμού Fourier. Πριν από την επεξεργασία ορίζονται δυο εικόνες - στόχοι. Στη συνέχεια ξεκινάει η διαδικασία της εικονοροής. Κάθε εικόνα που έρχεται από την κάμερα, συγκρίνεται με τις δυο εικόνες πρότυπα, ώστε να εντοπιστεί ο καθένας από τους δυο

στόχους. Εδώ πρέπει να αναφερθεί ότι μόνο ο πρώτος στόχος ενδιαφέρει πραγματικά, ενώ ο δεύτερος είναι καθαρά βοηθητικός για την μέτρηση της γωνίας περιστροφής του ρομποτικού ψαριού, ως προς το σύστημα αναφοράς της εικόνας (όπως αυτό ορίζεται στη MATLAB – βλ. Κεφ. 4.3.2).

Τα αποτελέσματα, που προκύπτουν κατά την λειτουργία του αλγορίθμου, αποθηκεύονται στη μνήμη. Αρχικά, γίνεται οπτικοποίηση τους, στην ίδια τη διαδικασία της εικονοροής. Δηλαδή, κατά την διάρκεια εκτέλεσης του αλγορίθμου, εμφανίζεται η εικόνα που στέλνεται από την κάμερα στον υπολογιστή, υπό την μορφή εικονοροής (video).

Οι εικόνες είναι σχεδόν συνεχόμενες, ώστε το ανθρώπινο μάτι να μην αντιλαμβάνεται, παρά μόνο μικρές ασυνέχειες, στην εικονοροή. Ενδεχόμενες ασυνέχειες που μπορεί να παρατηρηθούν, οφείλονται στο χρονικό διάστημα που απαιτείται από τον αλγόριθμο, για να εκτελέσει τις απαιτούμενες πράξεις. Ο χρόνος εκτέλεσης αυτών των πράξεων, μπορεί να είναι μεγαλύτερος από τον χρόνο, στον οποίο παρέχει νέες εικόνες η κάμερα στον υπολογιστή. Έτσι κάποιες εικόνες (frames) ίσως να χάνονται. Αν χάνονται πολλές εικόνες, το ανθρώπινο μάτι, αντιλαμβάνεται την ασυνέχεια που δημιουργείται. Στην συνέχεια θα αναφερθεί ένας τρόπος αντιμετώπισης αυτού του προβλήματος.

Επιπλέον όμως, πάνω σε αυτό το θέμα, υπάρχει και ένα ακόμα πρόβλημα, που είναι πιο δύσκολο να ξεπεραστεί. Αυτό είναι, η διαδικασία αναπαραγωγής των εικόνων που στέλνει η κάμερα, στην οθόνη του υπολογιστή. Η λειτουργία αυτή, απαιτεί μεγάλη υπολογιστική ισχύ, και σημαντικούς πόρους από τον υπολογιστή (CPU intensive task). Το θετικό όμως είναι, ότι αν και τα αποτελέσματα μπορεί να μην εμφανίζονται απόλυτα ομαλά στην οθόνη, οι υπόλοιπες λειτουργίες του αλγορίθμου συνεχίζονται κανονικά, χωρίς καθυστερήσεις. Δηλαδή, έστω ότι θέλουμε ο αλγόριθμος να επεξεργαστεί 200 εικόνες (frames). Τότε υπάρχει το ενδεχόμενο να αναπαραχθούν στην οθόνη μας, λιγότερες εικόνες (π.χ. 170 frames). Στην πραγματικότητα όμως, ο αλγόριθμος έχει επεξεργαστεί και τις 200 εικόνες. Απλά δεν προλαβαίνει να τις εμφανίσει όλες, με ρυθμό τέτοιο ώστε να φαίνεται στο μάτι, μια συνεχόμενη διαδικασία εικονοροής. Και με αυτό το πρόβλημα όμως, έγινε μια αξιόλογη προσπάθεια για να ξεπεραστεί, η οποία θα αναφερθεί στη συνέχεια.

Ακόμα, πάνω στις εικόνες, που εμφανίζονται στην οθόνη, σημειώνονται επιπλέον στοιχεία, ώστε να οπτικοποιούνται καλύτερα τα αποτελέσματα. Για παράδειγμα, εμφανίζεται, με πράσινο πλαίσιο, το περίγραμμα του πρώτου και βασικού στόχου που έχουμε επιλέξει. Ακόμα, το μπλε αστέρι συμβολίζει το κέντρο του στόχου. Δηλαδή είναι το κέντρο του παραλληλογράμμου. Ακόμα, το κίτρινο παραλληλόγραμμο, περιέχει το δεύτερο και βοηθητικό στόχο. Και εδώ υπάρχει ένα μπλε αστέρι στο κέντρο του στόχου. Τέλος, με το κόκκινο πλαίσιο εμφανίζεται η περιοχή αναζήτησης, μέσα στην οποία γίνεται η αναζήτηση για τους δυο στόχους. Για την περιοχή αυτή θα γίνει πιο αναλυτική αναφορά στη συνέχεια.

Μετά την ολοκλήρωση της λειτουργίας του αλγορίθμου, παρέχεται και μια σειρά

αριθμητικών αποτελεσμάτων. Αυτά μπορούμε να τα μελετήσουμε αναλυτικά μετά το πέρας της εκτέλεσης του αλγορίθμου. Όλα τα αριθμητικά δεδομένα στην MATLAB, εμφανίζονται υπό την μορφή πινάκων. Έτσι, είναι πολύ εύκολο να εξετάσουμε τα αριθμητικά αποτελέσματα, σχηματίζοντας τις αντίστοιχες γραφικές παραστάσεις.

Το σύνολο των αριθμητικών αποτελεσμάτων που παρέχει ο αλγόριθμος, περιγράφονται στην συνέχεια. Μας παρέχει, αρχικά, τις συντεταγμένες του στόχου. Οι συντεταγμένες αυτές, στην ουσία είναι το σημείο εκείνο της εικόνας, όπου βρέθηκε η μέγιστη ομοιότητα, μεταξύ της εικόνας - προτύπου και της τρέχουσας εικόνας.

Επιπλέον, γνωρίζουμε εκ των προτέρων, το μέγεθος της εικόνας. Κατά συνέπεια μπορούμε να γνωρίζουμε ανά πάσα στιγμή το κέντρο της εικόνας. Έτσι μπορούμε να εκτελέσουμε μια πολύ σημαντική διαδικασία, στην περίπτωση που η κάμερα, είναι προσδεσμένη στο φορείο. Την παρακολούθηση του ψαριού από το φορείο.

Τα δύο μέγεθι που περιγράφονται πιο πάνω, μας παρέχουν την δυνατότητα, για τη συγκεκριμένη λειτουργία. Γνωρίζοντας την τρέχουσα θέση του στόχου, και την επιθυμητή, δηλαδή το κέντρο της εικόνας, γνωρίζουμε και το αντίστοιχο διάνυσμα μετατόπισης, από την γνωστή σχέση:

$$\Delta \vec{x} = \vec{x}_{\text{τελικό}} - \vec{x}_{\text{αρχικό}} \quad (4-44)$$

όπου $\vec{x}_{\text{αρχικό}}$, η τρέχουσα θέση του στόχου και $\vec{x}_{\text{τελικό}}$ η επιθυμητή θέση, δηλαδή το μέσο της εικόνας. Το διάνυσμα αυτό, προφανώς έχει δύο συνιστώσες. Η μια συνιστώσα, είναι η μετατόπιση κατά τον x άξονα, και η άλλη η μετατόπιση κατά τον y άξονα. Τους δυο αυτούς αριθμούς μπορούμε να τους παρέχουμε ως σήματα εισόδου (εντολές), στα συστήματα ελέγχου των κινητήρων των δυο αξόνων, αντίστοιχα. Με τον τρόπο αυτό, οι κινητήρες θα διατηρούν το φορείο, και κατά συνέπεια την κάμερα που είναι προσδεσμένη σε αυτό, πάνω, ή σχεδόν πάνω από το ρομποτικό ψάρι – στόχο. Μάλιστα, ο στόχος θα εμφανίζεται κοντά στο μέσο της εικόνας. Αναλυτικά, την υλοποίηση αυτής της λειτουργίας, θα την δούμε σε επόμενο σημείο της εργασίας.

Τα παραπάνω αριθμητικά αποτελέσματα, αφορούν θέσεις ή αποστάσεις σημείων. Η μονάδα μέτρησης όλων αυτών είναι προφανώς το εικονοστοιχείο (pixel). Για να βρούμε την πραγματική απόσταση των σημείων εκείνων που φαίνονται στην εικόνα, θα πρέπει να γνωρίζουμε το μέγεθος κάθε εικονοστοιχείου. Το στοιχείο αυτό παρέχεται από τον κατασκευαστή της κάμερας. Στη συνέχεια, θα πρέπει να βρούμε την απόσταση (σε mm, cm) που αντιστοιχεί σε ένα εικονοστοιχείο. Δηλαδή αν γνωρίζουμε πόσα εικονοστοιχεία απέχουν στην εικόνα μας δυο σημεία, να γνωρίζουμε πόσα χιλιοστά απέχουν στην πραγματικότητα.

Για να το επιτύχουμε, ακολουθήσαμε την διαδικασία που περιγράφεται στην συνέχεια. Εξετάσαμε με την κάμερα ένα αντικείμενο γνωστού μήκος (έναν χάρακα, μήκους 40cm), τοποθετημένο στο βάθος που θα κινείται το ρομποτικό ψάρι. Στην συνέχεια είδαμε πόσα εικονοστοιχεία ήταν στην εικόνα μας το μήκος του χάρακα. Με τον τρόπο αυτό, έγινε γνωστή η

αντιστοίχιση εικονοστοιχείου σε εκατοστά. Το αποτέλεσμα που προέκυψε είναι ότι το μήκος (ή το πλάτος) ενός εικονοστοιχείου, αντιστοιχεί σε μήκος 1.1196 mm.

Εδώ πρέπει να σημειωθεί ότι η μέθοδος αυτή δεν παρέχει πάντα απόλυτη ακρίβεια. Αυτό συμβαίνει γιατί η εικόνα που παρέχεται από την κάμερα είναι ελαφρώς καμπυλωμένη στα άκρα της. Το γεγονός αυτό οφείλεται στο βαρελοειδές σχήμα του φακού. Υποθέτουμε όμως, ότι η καμπύλωση λαμβάνει χώρα μόνο στα άκρα της εικόνας. Η λειτουργία του αλγόριθμου που σχεδιάσαμε, είναι τέτοια που δεν επιτρέπει στον στόχο να ξεφεύγει πολύ από το μέσο της εικόνας. Έτσι μπορούμε να πούμε ότι η αντιστοίχιση που βρήκαμε (εικονοστοιχείο σε χιλιοστά), είναι ακριβής για την εφαρμογή που εξετάζουμε και για την μεθοδολογία με την οποία το επιτυγχάνουμε.

Φυσικά, μπορούμε να μελετήσουμε μόνο τη θέση στην οποία βρίσκεται ο στόχος σε κάθε εικόνα, με βάση το σύστημα συντεταγμένων της MATLAB. Από αυτά τα δεδομένα, μπορούμε να εξάγουμε συμπεράσματα, για μια σειρά μεγεθών, όπως τροχιά, ταχύτητα κ.α.

Όπως έχει αναφερθεί, στη διαδικασία επεξεργασίας εικόνας, ο αλγόριθμος αναγνωρίζει δυο στόχους. Αν και οι δυο στόχοι βρίσκονται πάνω στο ίδιο σώμα – στη συγκεκριμένη εφαρμογή, στο ρομποτικό ψάρι – τότε μπορούμε εύκολα να βρούμε την γωνία, με την οποία έχει περιστραφεί το σώμα. Από την επεξεργασία της εικόνας προκύπτουν οι συντεταγμένες των δυο εικόνων στόχων. Μέσω της σχέσης:

$$\tan\theta = \frac{y_2 - y_1}{x_2 - x_1} \quad (4-45)$$

βρίσκουμε την εφαπτομένη της γωνίας περιστροφής. Επομένως μπορούμε να γνωρίζουμε ανά πάσα στιγμή, πόσο έχει περιστραφεί το σώμα, ως προς το σύστημα συντεταγμένων της εικόνας.

Όσον αφορά, την αδυναμία του αλγόριθμου να επεξεργάζεται επιτυχώς εικόνες υπό κλίμακα, έγινε η παραδοχή, ότι οι αυξομειώσεις βάθους κατά την κίνηση του ρομποτικού ψαριού είναι τόσο μικρές ώστε ο αλγόριθμος να μπορεί να τις διαχειριστεί. Ένα ακόμα σημαντικό πρόβλημα στην λειτουργία του αλγόριθμου, είναι η αδυναμία του να διαχειριστεί εικόνες που έχουν περιστραφεί. Το πρόβλημα αυτό είναι πολύ σημαντικό στην συγκεκριμένη εφαρμογή, καθώς ο στόχος ήταν να παρακολουθείται το ρομποτικό ψάρι, καθ'όλη τη διάρκεια της κίνησης του, ανεξάρτητα από αλλαγές κατεύθυνσης. Για να ξεπεραστεί το πρόβλημα αυτό, έγινε μια προσθήκη στον κώδικα. Ο αλγόριθμος κατά την λειτουργία του, χρειάζεται προφανώς την εικόνα-πρότυπο, την οποία αναζητά σε κάθε εικόνα που του παρέχεται από την διαδικασία της εικονοροής. Αν η εικόνα-πρότυπο είναι σταθερή, ο αλγόριθμος θα ψάχνει για το συγκεκριμένο σχήμα με το συγκεκριμένο προσανατολισμό. Όταν το ρομποτικό ψάρι περιστραφεί σε μια μεγάλη σχετική γωνία, θα αδυνατεί να το βρίσκει. Επομένως, θα πρέπει η λειτουργία εισαγωγής στον αλγόριθμο, της εικόνας-πρότυπο να είναι μια δυναμική διαδικασία. Δηλαδή να ανανεώνεται συνεχώς, ώστε ο αλγόριθμος να έχει διαρκώς την τρέχουσα εικόνα που έχει ο στόχος, στην εικόνα που

επεξεργάζεται. Στην όλη διαδικασία της εικονοροής, ο αλγόριθμος επεξεργάζεται κάθε μια εικόνα (frame) που του παρέχεται από την κάμερα. Η κάμερα παρέχει ένα μεγάλο αριθμό εικόνων ανά δευτερόλεπτο (30 frames per second-fps). Για το λόγο αυτό, μπορούμε να θεωρήσουμε ότι μεταξύ δυο διαδοχικών εικόνων, το ψάρι θα έχει περιστραφεί τόσο λίγο, ώστε ο αλγόριθμος να μπορεί ακόμα να το αναγνωρίζει. Έτσι μετά την ολοκλήρωση της επεξεργασίας κάθε εικόνας (frame), μπορούμε να ορίσουμε ως νέο στόχο για την επόμενη εικόνα, την περιοχή γύρω από το σημείο που βρέθηκε ο στόχος στην τρέχουσα. Με τον τρόπο αυτό, επιτυγχάνεται η διαδικασία ανανέωσης του στόχου σε κάθε εικόνα (frame). Η ανανέωση αυτή γίνεται με τρόπο ομαλό. Αυτό συμβαίνει γιατί ο κάθε νέος στόχος διαφέρει από τον προηγούμενο, κατά μια πολύ μικρή γωνία, την οποία ο αλγόριθμος μπορεί να διαχειριστεί. Έτσι, μπορεί να παρακολουθεί το ρομποτικό ψάρι και σε μεγάλες γωνίες, αφού στην ουσία τις χωρίζει σε πολύ μικρότερες που είναι διαχειρίσιμες γι'αυτόν.

4.5.1 Αναλυτική Περιγραφή Αλγορίθμου

Στο κεφάλαιο αυτό, θα παρουσιάσουμε αναλυτικά τον κώδικα, γραμμή-γραμμή. Θα αναφέρουμε τις εντολές της MATLAB που χρησιμοποιήσαμε κατά τη συγγραφή του κώδικα και θα περιγράψουμε αναλυτικά τη λειτουργία τους. Εκτός από τις γενικές εντολές της γλώσσας, χρησιμοποιούμε και πιο εξειδικευμένες εντολές. Αυτές αφορούν στην απόκτηση και την επεξεργασία εικόνας, και παρέχονται από τις αντίστοιχες εργαλειοθήκες (toolboxes). Οι εργαλειοθήκες της MATLAB, που χρησιμοποιήθηκαν στην συγκεκριμένη εφαρμογή είναι το Image Acquisition Toolbox, και το Image Processing Toolbox. Το πρώτο παρέχει εξειδικευμένες εντολές, για την απόκτηση εικόνας, από την αντίστοιχη συσκευή, δηλαδή την κάμερα. Το Image Processing Toolbox, προσφέρει ένα μεγάλο αριθμό εντολών, κατάλληλων για επεξεργασία εικόνας. Στις γραμμές που ακολουθούν, όσα τμήματα του κειμένου περικλείονται σε πράσινο πλαίσιο, σημαίνει ότι είναι αυτούσια κομμάτια του κώδικα από το αντίστοιχο m-file της MATLAB.

Αρχικά, θα πρέπει να δηλώσουμε το αντικείμενο που θα παρέχει την εικόνα στον αλγόριθμο. Αυτό επιτυγχάνεται με την εντολή:

```
vid = videoinput('macvideo',2);
```

Έτσι έχουμε μια είσοδο στον αλγόριθμο, που ονομάσαμε **vid**, η οποία στην ουσία είναι ροή εικόνων. Το πόσες, ποιές και πώς ονομάζονται οι κάμερες που είναι συνδεδεμένες στον υπολογιστή, μπορούμε να το ελέγξουμε με χρήση της εντολής **imaqhwinfo**. Η έξοδος της εντολής είναι ένας κατάλογος, όλων των συσκευών απόκτησης εικόνας που είναι συνδεδεμένες εκείνη την στιγμή στον υπολογιστή. Επομένως, μπορούμε να δηλώσουμε την συσκευή που θέλουμε, βάζοντας την σαν όρισμα εισόδου στην αντίστοιχη εντολή.

Η εντολή μας δίνει και την δυνατότητα να αλλάξουμε κάποιες από τις ρυθμίσεις της κάμερας. Για παράδειγμα τον ρυθμό απόκτησης εικόνων, το μέγεθος ή το format. Επειδή οι ρυθμίσεις της κάμερας καλύπτουν τις ανάγκες της εφαρμογής, θα διατηρήσουμε προς το παρόν τις

ρυθμίσεις του κατασκευαστή. Αν χρειαστεί να τις αλλάξουμε, μπορούμε να το κάνουμε και σε μεταγενέστερο στάδιο, όπως θα δούμε στη συνέχεια.

Έπειτα, και για να διατηρείται η διαδραστικότητα με τον αλγόριθμο, ο χρήστης καλείται να επιλέξει με το ποντίκι του υπολογιστή, τις εικόνες - πρότυπα (βασικό και βοηθητικό στόχο). Αυτές τις δύο εικόνες, θα αναζητήσουμε στην διαδικασία της εικονοροής. Για να επιτευχθεί αυτό, πρέπει πριν, να έχουμε μια γενικότερη εικόνα του τι βλέπει η κάμερα. Έτσι ο αλγόριθμος, καλεί την κάμερα, να του παράσχει μια εικόνα. Η εικόνα αυτή θα ονομάζεται στη συνέχεια **εικόνα αρχικοποίησης**. Στην εικόνα αρχικοποίησης ο χρήστης μπορεί να επιλέξει τον κύριο και τον βοηθητικό στόχο, καθώς και την περιοχή ενδιαφέροντος. Η εικόνα αυτή παρέχεται με την εντολή:

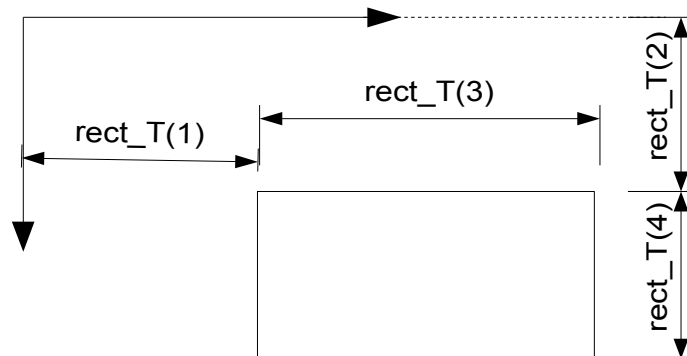
```
T = getsnapshot(vid);
```

Με τον τρόπο αυτό, η κάμερα παρέχει στον υπολογιστή, την **εικόνα αρχικοποίησης** που ονομάζεται **T**, μέσα στην οποία πρέπει να βρίσκονται και οι δυο στόχοι. Η εικόνα αυτή, πρέπει να είναι σε φορμάτ RGB, στο οποίο έχουμε αναφερθεί προηγούμενα. Αν είναι σε κάποιο άλλο φορμάτ, θα πρέπει να το μετατρέψουμε σε αυτό. Για παράδειγμα, ο υπολογιστής στον οποίο γράφτηκε ο κώδικας, παρέχει τις εικόνες σε διαφορετικό φορμάτ (YCbCr). Για να τις μετατρέψουμε σε RGB, κάνουμε χρήση της εντολής,

```
T = ycbcr2rgb(T);  
[sub_Tinit, rect_T] = imcrop(T);
```

Τώρα μπορεί ο χρήστης να επιλέξει την εικόνα πρότυπο, με την βοήθεια της τελευταίας εντολής.

Η εντολή **imcrop**, δίνει την δυνατότητα στον χρήστη να απομονώσει και να διατηρήσει μόνο ένα μέρος της εικόνας. Η νέα εικόνα **sub_Tinit** προκύπτει, αν τοποθετήσουμε το ορθογώνιο **rect_T**, πάνω στην εικόνα **T**. Το ορθογώνιο **rect_T** αντιστοιχίζεται σε έναν πίνακα διαστάσεων (1x4) . Το πρώτο και το δεύτερο στοιχείο αντιστοιχούν στις x και y συντεταγμένες, του πάνω αριστερά άκρου του ορθογωνίου. Τα δύο επόμενα στοιχεία αντιστοιχούν στο μέγεθος του ορθογωνίου κατά τον x και τον y άξονα, αντίστοιχα. Πάντα το σύστημα συντεταγμένων είναι αυτό που χρησιμοποιεί η MATLAB. Αναλυτικά το πώς ορίζεται το ορθογώνιο **rect_T**, φαίνεται στο Σχήμα 4-11:



Σχήμα 4-11: Στοιχεία πίνακα ορισμού του στόχου.

Η νέα μικρότερη πλέον εικόνα, που αποτελεί την εικόνα-πρότυπο ονομάζεται **sub_Tinit** και θα πρέπει να περιέχει μόνο τον βασικό στόχο, που θέλουμε να αναζητήσουμε στην διαδικασία της εικονοροής. Για καλύτερα αποτελέσματα, θα πρέπει η εικόνα-πρότυπο να είναι όσο το δυνατόν μικρή, και να περιέχει μόνο την απαραίτητη πληροφορία. Δηλαδή θα πρέπει να περιέχει μόνο τον στόχο, ή τμήμα αυτού. Δεν είναι βάσιμο το σκεπτικό ότι όσο πιο μεγάλη είναι η εικόνα-πρότυπο, τόσο πιο καλά θα "βλέπει" ο υπολογιστής τον στόχο. Ο αλγόριθμος είναι δομημένος κατά τέτοιο τρόπο ώστε να μπορεί να ξεχωρίζει ακόμα και παρόμοιους στόχους, και να βρίσκει τον επιθυμητό με πολύ μεγάλη ακρίβεια. Αντίθετα, όπως είναι κατανοητό, όσο λιγότερα τα δεδομένα τόσο ταχύτερη είναι η επεξεργασία. Επιπλέον, αν επιλέξουμε πολύ μεγάλη την εικόνα πρότυπο, υπάρχει το ενδεχόμενο να μην μπορεί καν ο υπολογιστής να αναπαράξει τις επεξεργασμένες εικόνες υπό μορφή εικονοροής. Δηλαδή μπορεί να επιτύχουμε ακριβώς αντίθετα αποτελέσματα από τα επιθυμητά. Κατά συνέπεια για την επιλογή της εικόνας πρότυπο, επιλέγουμε μόνο την απαραίτητη πληροφορία. Ο κώδικας συνεχίζεται με την ονομασία στοιχείων του πίνακα **rect_T**, τα οποία θα μας χρειαστούν στην συνέχεια. Ακόμα μετατρέπουμε τα στοιχεία του **rect_T**, σε διπλής ακρίβειας:

```
xlength = rect_T(3);
ylength = rect_T(4);
rect_T = double(rect_T);
sub_T = sub_Tinit;
```

Με την τελευταία εντολή καθορίζουμε, για την έναρξη της επεξεργασίας, ως εικόνα-πρότυπο **sub_T**, το τμήμα της εικόνας, που έχει καθοριστεί από το χρήστη **sub_Tinit**. Αυτό ισχύει μόνο για την πρώτη εικόνα, αφού η εικόνα-πρότυπο αλλάζει διαρκώς, αυτόματα όπως θα δούμε στη συνέχεια.

Με τον ίδιο τρόπο, καθορίζεται από τον χρήστη και η δεύτερη εικόνα-πρότυπο **sub_T2**. Η εικόνα αυτή, όπως έχει αναφερθεί είναι στην ουσία ο βοηθητικός στόχος. Και αυτός ο στόχος βρίσκεται πάνω στην πρώτη εικόνα **T**.

```
[sub_Tinit2, rect_T2] = imcrop(T);  
xlength2 = rect_T2(3);  
ylength2 = rect_T2(4);  
rect_T2 = double(rect_T2);  
sub_T2 = sub_Tinit2;
```

Ένας ακόμα πολύ σημαντικός τρόπος να μειώσουμε τον χρόνο επεξεργασίας κάθε εικόνας, είναι να επικεντρώσουμε την αναζήτηση σε μια περιοχή ενδιαφέροντος. Όπως είναι προφανές, αν ο αλγόριθμος αναζητά τον στόχο σε όλη την εικόνα απαιτείται ένας μεγάλος αριθμός πράξεων. Δηλαδή πρέπει να μετατρέψει, όλα τα εικονοστοιχεία της εικόνας στο πεδίο της συχνότητας, να εκτελέσει τους απαιτούμενους υπολογισμούς για όλα αυτά, και να τα επαναφέρει στο πεδίο του χρόνου. Για να μειώσουμε αυτήν την διαδικασία, ορίζουμε μια μικρότερη περιοχή ενδιαφέροντος (Region Of Interest – **ROI**). Δηλαδή, κάνουμε την παραδοχή ότι ο στόχος μπορεί να βρίσκεται μόνο στην συγκεκριμένη περιοχή, επομένως δεν εκτελούνται πράξεις για οτιδήποτε βρίσκεται έξω από αυτήν. Με τον τρόπο αυτό, γλυτώνουμε έναν πολύ μεγάλο αριθμό πράξεων, που στην ουσία μας είναι άχρηστες. Επιπλέον, η παραδοχή αυτή είναι απόλυτα ρεαλιστική. Το ρομποτικό ψάρι, που είναι ο στόχος μας, έχει μια μέγιστη ταχύτητα. Έτσι, σε κάθε νέα εικόνα, μπορεί να βρίσκεται μόνο σε έναν συγκεκριμένο χώρο γύρω από το σημείο που έχει εντοπιστεί στην προηγούμενη. Ο χώρος αυτός, καθορίζεται από την μέγιστη ταχύτητα του, και από τον χρόνο που μεσολαβεί από εικόνα σε εικόνα. Επομένως είναι απόλυτα ρεαλιστικό και σημαντικό, να περιορίσουμε την αναζήτηση του στόχου, σε ένα μόνο μέρος της εικόνας. Αυτό επιτυγχάνεται με χρήση των εντολών:

```
[roi, rect_roi] = imcrop(T);  
imshow(roi)  
roi_xstart = rect_roi(1);  
roi_xend = rect_roi(1) + rect_roi(3);  
roi_ystart = rect_roi(2);  
roi_yend = rect_roi(2) + rect_roi(4);
```

Όπως φαίνεται η διαδικασία επιλογής είναι πανομοιότυπη με την διαδικασία επιλογής του στόχου. Δηλαδή, δηλώνουμε την περιοχή αναζήτησης **roi**, με βάση ένα ορθογώνιο που το επιλέγουμε με το ποντίκι του υπολογιστή, πάνω στην εικόνα και το ονομάζουμε **rect_roi**. Το ορθογώνιο δηλώνεται μόνο στην πρώτη εικόνα, και στη συνέχεια ανανεώνεται αυτόματα για τις υπόλοιπες εικόνες. Επιπλέον, κάποια σημαντικά μεγέθη, τα οποία θα τα χρησιμοποιήσουμε, τα ονοματίζουμε, για να είναι πιο εύχρηστα.

Ένα τελευταίο σημαντικό πρόβλημα που έπρεπε να λυθεί σχετικά με την επιλογή της περιοχής ενδιαφέροντος, ήταν η διαδικασία αναπαράστασης των αποτελεσμάτων, πάνω στην διαδικασία της εικονοροής. Όπως αναφέρθηκε και προηγούμενα η διαδικασία αυτή έχει σημαντικό

υπολογιστικό κόστος. Η πιο συνηθισμένη μορφή με την οποία εμφανίζονταν αυτό το πρόβλημα, ήταν αν επιλέγονταν μεγάλη **περιοχή ενδιαφέροντος** (region of interest-ROI), ή μεγάλες εικόνες πρότυπα. Το μέγεθος των εικόνων-προτύπων δεν αποτελούσε πρόβλημα, καθώς ο αλγόριθμος είναι σε θέση να αναγνωρίζει και μικρές εικόνες. Όμως, στην περίπτωση που η κάμερα είναι σταθερή πάνω από τη δεξαμενή και παρακολουθεί το ψάρι που κινείται από κάτω της, το πρόβλημα οξύνεται. Στην περίπτωση αυτή, ως περιοχή ενδιαφέροντος, θα πρέπει να οριστεί όλη, ή μεγάλο μέρος της εικόνας. Έτσι η διαδικασία αναπαράστασης των δεδομένων καταρρέει. Πάντως ακόμα και έτσι ο αλγόριθμος λειτουργεί, δηλαδή επεξεργάζεται τις εικόνες και εξάγει αποτελέσματα. Απλά δεν τα εμφανίζει κατά τη διαδικασία της εικονοροής, αλλά μετά το πέρας της επεξεργασίας. Όμως το πρόβλημα παραμένει, αφού δεν γνωρίζουμε πότε ξεκινά η λειτουργία του αλγόριθμου, ώστε αυτή να συγχρονιστεί με την κίνηση του ρομποτικού ψαριού. Ακόμα δεν μπορούμε να γνωρίζουμε αν ο στόχος παραμένει μέσα στην περιοχή ενδιαφέροντος. Αυτό οδηγεί σε εσφαλμένη αξιολόγηση των αποτελεσμάτων που προκύπτουν από την διαδικασία.

Η λύση που βρέθηκε για το συγκεκριμένο πρόβλημα, είναι η **μεταβλητή περιοχή ενδιαφέροντος**. Με τη διαδικασία αυτή, η περιοχή ενδιαφέροντος δεν είναι στατική πάνω στην ακολουθία των εικόνων, δηλαδή δεν ορίζεται μια φορά στην αρχή του αλγορίθμου και παραμένει αυτή μέχρι το τέλος της επεξεργασίας. Αντίθετα, η θέση της ανανεώνεται μετά την επεξεργασία κάθε εικόνας. Η νέα θέση της καθορίζεται από την νέα θέση στην οποία βρέθηκε ο στόχος. Δηλαδή, εξετάζεται η μετατόπιση του στόχου σε σχέση με την προηγούμενη θέση του – έστω $\Delta \vec{x}$. Στη νέα αναζήτηση, η νέα θέση της περιοχής ενδιαφέροντος, προκύπτει από την προηγούμενη της, μετατοπισμένη κατά $\Delta \vec{x}$.

Με τον τρόπο αυτό, η περιοχή ενδιαφέροντος βρίσκεται πάντα πάνω από το στόχο. Μπορεί να τον ακολουθεί σε όλο το εύρος της εικόνας, χωρίς να διαταράσσεται η διαδικασία αναπαράστασης των αποτελεσμάτων, πάνω στη διαδικασία της εικονοροής. Ένα ακόμα πλεονέκτημα που προκύπτει από την λειτουργία αυτή, είναι η σημαντική εξοικονόμηση υπολογιστικού κόστους. Αυτό συμβαίνει γιατί, πλέον δεν χρειάζεται μεγάλη περιοχή ενδιαφέροντος, ώστε να καλύπτεται όσο το δυνατόν μεγαλύτερο εύρος της κίνησης του ψαριού. Αντίθετα και μια πολύ μικρότερη περιοχή ενδιαφέροντος αρκεί για να παρακολουθείται ο στόχος σε όλο μάλιστα το εύρος της κίνησης του μέσα σε μια εικόνα. Το πώς υλοποιείται αυτό προγραμματιστικά θα το δούμε στην συνέχεια του κώδικα.

Τέλος, θα πρέπει να βρίσκουμε το μέσο της οθόνης. Στην περίπτωση όπου η κάμερα είναι προσδεσμένη στο φορείο, η γνώση του είναι πολύ σημαντική, αφού αυτό είναι το σημείο, κοντά στο οποίο θέλουμε να βρίσκεται ο στόχος μας.

```
[r, c, d] = size(T);
gpos = [c/2, r/2];
i = 1;
movex = zeros(201,1);
```

```

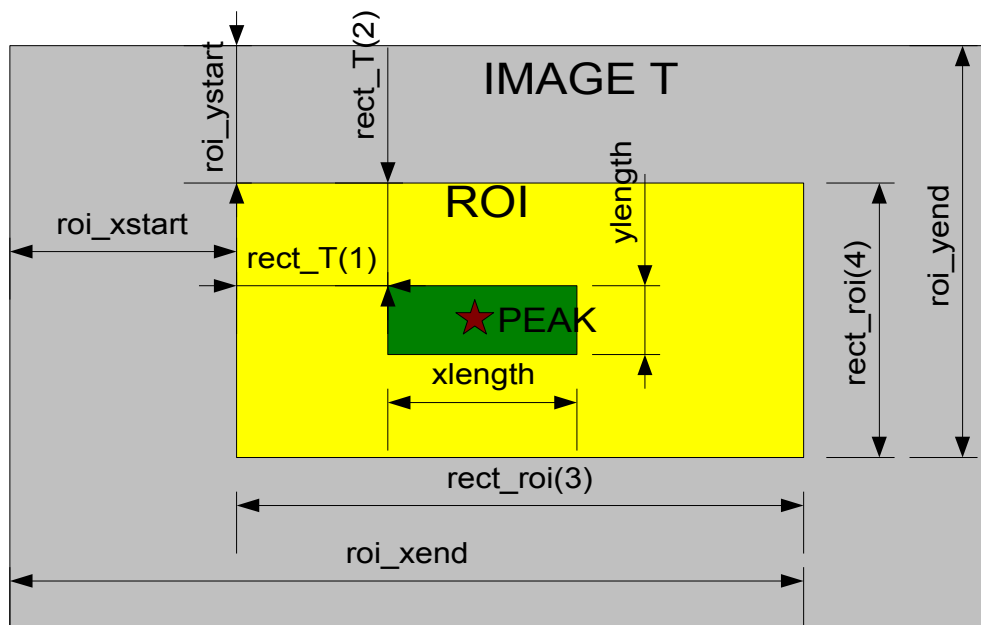
movey = zeros(201,1);
xreal = zeros(200,1);
yreal = zeros(200,1);
xreal2 = zeros(200,1);
yreal2 = zeros(200,1);
timetable = zeros(200,1);

```

Στις τελευταίες γραμμές, δηλώνουμε κάποιες μεταβλητές τις οποίες θα χρησιμοποιήσουμε στη συνέχεια, οπότε και θα περιγραφούν πιο αναλυτικά.

Τα μεγέθη αυτά, είναι ένας μετρητής **i**, και πίνακες στους οποίους αποθηκεύονται, η απόλυτη και η σχετική θέση (ως προς το κέντρο της εικόνας) ενός εκ των δύο στόχων, αντίστοιχα (και για τον άλλο στόχο ισχύουν ακριβώς τα ίδια).

Στο Σχήμα 4-12 που ακολουθεί φαίνονται πιο αναλυτικά όλα αυτά τα μεγέθη που περιγράφηκαν .

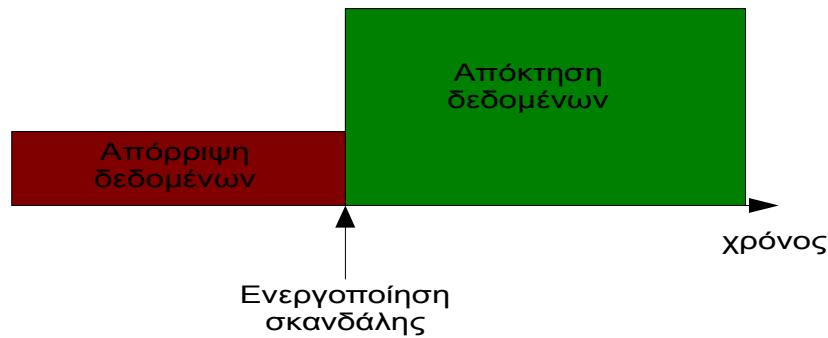


Σχήμα 4-12: Εικόνα, περιοχή ενδιαφέροντος και εύρεση στόχου (peak).

Στο Σχήμα 4-12, για λόγους ευκρίνειας, έχει παραληφθεί ο δεύτερος στόχος, και αναπαρίσταται μόνο ο πρώτος και βασικός. Προφανώς με βάση τα ανωτέρω, τα ίδια ισχύουν και για τον δεύτερο (βοηθητικό) στόχο.

Μια ακόμα ρύθμιση που πρέπει να κάνουμε πριν την έναρξη της επεξεργασίας των εικόνων, είναι η αρχικοποίηση της κάμερας και της διαδικασίας της εικονοροής.

Ένα βασικό στοιχείο, στην απόκτηση δεδομένων από εξωτερικές πηγές, είναι η έννοια της σκανδάλης (trigger). Στην ουσία, όταν ενεργοποιείται η σκανδάλη, ξεκινά η απόκτηση των δεδομένων από τον υπολογιστή, όπως φαίνεται και στο Σχήμα 4-13.



Σχήμα 4-13: Έναρξη διαδικασίας απόκτησης δεδομένων (triggering).

Δηλαδή, η κάμερα μπορεί να στέλνει συνέχεια εικόνα, όμως ο υπολογιστής δεν την λαμβάνει αν δεν έχει ενεργοποιηθεί η σκανδάλη. Όταν ενεργοποιηθεί η σκανδάλη, ο υπολογιστής λαμβάνει πλέον τις εικόνες από την κάμερα. Τα δεδομένα αυτά μπορεί να τα επεξεργάζεται, ή να τα αποθηκεύει, στη μνήμη ή σε κάποιο αρχείο στον δίσκο. Η προεπιλογή γι' αυτήν την δυνατότητα, είναι η αποθήκευση στη μνήμη. Επειδή σε πρώτη φάση, δεν θα προχωρήσουμε σε περαιτέρω επεξεργασία των εικόνων, θα διατηρήσουμε την επιλογή της αποθήκευσης στη μνήμη. Ορίζουμε, επίσης ότι η ενεργοποίηση της σκανδάλης θα επαναλαμβάνεται αυτόματα, για όση ώρα διαρκεί η λήψη της εικονοροής και η επεξεργασία των εικόνων. Ακόμα, ορίζουμε ότι μεταξύ δυο διαδοχικών ενεργοποιήσεων της σκανδάλης, ο υπολογιστής θα λαμβάνει 300 εικόνες (frames) από την κάμερα. Από αυτές, θα κρατάει και θα αποθηκεύει μια μόνο εικόνα, από κάθε 5 που στέλνει η κάμερα. Αυτός είναι ένας ρυθμός, που αφ'ενός επιτρέπει μια συνέχεια στη διαδικασία της εικονοροής, και αφ'ετέρου, διατηρεί τον όγκο των δεδομένων προς επεξεργασία σε διαχειρίσιμα επίπεδα.

```
set(vid, 'TriggerRepeat', Inf);
set(vid, 'FramesPerTrigger', 300);
vid.FrameGrabInterval = 5;
```

Πριν αρχίσει, η διαδικασία της εικονοροής, ο αλγόριθμος βρίσκει στην αρχική εικόνα **T**, τους δυο στόχους (βασικό και βοηθητικό) που έχουμε ορίσει. Η εύρεση των στόχων, γίνεται με βάση την πράξη της συσχέτισης, όπως έχουμε αναφέρει. Ο λόγος που γίνεται η εύρεση των στόχων και στην αρχική εικόνα **T**, είναι γιατί τα αποτελέσματα που θα προκύψουν, θα χρησιμοποιηθούν στην συνέχεια του αλγόριθμου. Συγκεκριμένα γίνεται χρήση τους στην διαδικασία της μεταβλητής περιοχής ενδιαφέροντος. Όπως αναφέρθηκε προηγούμενα, η περιοχή ενδιαφέροντος, καθορίζεται σε κάθε εικόνα, από το σημείο που βρέθηκε ο στόχος στην προηγούμενη. Όμως για την πρώτη εικόνα, που θα στείλει η κάμερα, δεν υπάρχει προηγούμενη. Για το λόγο αυτό βρίσκουμε τους δυο στόχους και στην **εικόνα αρχικοποίησης**. Η θέση που

βρίσκεται ο βασικός στόχος στην εικόνα αρχικοποίησης, καθορίζει την θέση της περιοχής ενδιαφέροντος, για την πρώτη εικόνα που στέλνει η κάμερα, κατά την διάρκεια εκτέλεσης του αλγορίθμου. Στις γραμμές που ακολουθούν, καθορίζεται η περιοχή ενδιαφέροντος, με βάση την θέση του στόχου στην προηγούμενη εικόνα, όπως έχει περιγραφεί. Οι γραμμές στον κώδικα που εκτελούν τις παραπάνω πράξεις ακολουθούν στη συνέχεια:

```

c_im1 =
normxcorr2(sub_T(:, :, 1), T(roi_ystart:roi_yend, roi_xstart:roi_xend, 1));
%Correlation in the frequency domain
[max_c_im1, imax_im1] = max(c_im1(:));
[ypeak_im1, xpeak_im1] = ind2sub(size(c_im1), imax_im1);

c_im2 =
normxcorr2(sub_T2(:, :, 1), T(roi_ystart:roi_yend, roi_xstart:roi_xend, 1));
%Correlation in the frequency domain
[max_c_im2, imax_im2] = max(c_im2(:));
[ypeak_im2, xpeak_im2] = ind2sub(size(c_im2), imax_im2);

```

Όπως φαίνεται από τις γραμμές του κώδικα, εκτελούμε την πράξη της κανονικοποιημένης συσχέτισης σε δυο διαστάσεις (**normxcorr2**). Η πράξη αυτή εκτελείται μεταξύ των δυο εικόνων στόχων (**sub_T** και **sub_T2**) και μόνο στην περιοχή ενδιαφέροντος που έχει οριστεί στην **εικόνα αρχικοποίησης**.

Τέλος, ενεργοποιούμε την κάμερα μέσω του υπολογιστή. Με τον τρόπο αυτό, ανοίγει στην ουσία, ο διάυλος επικοινωνίας, μεταξύ της κάμερας και του αλγόριθμου, ώστε η πρώτη να παρέχει δεδομένα στον δεύτερο.

```

start(vid);

```

Στο σημείο αυτό, έχει ολοκληρωθεί η διαδικασία της αρχικοποίησης, και ακολουθεί η διαδικασία της επεξεργασίας των εικόνων που παρέχονται από την κάμερα, με τη μορφή εικονοροής. Προκειμένου η επεξεργασία να ολοκληρωθεί αυτόματα, θεωρούμε έναν αριθμό εικόνων που θα λαμβάνεται από τον αλγόριθμο, για επεξεργασία. Όταν συμπληρωθεί ο αριθμός αυτός, η διαδικασία επεξεργασίας, σταματά αυτόματα. Για να επιτευχθεί αυτό, βάζουμε όλο τον κώδικα της επεξεργασίας, μέσα σε ένα βρόχο **while**. Επομένως η επεξεργασία των εικόνων, εκτελείται μέχρι να ικανοποιηθεί η συνθήκη τερματισμού του βρόχου.

```

while(vid.FramesAcquired<=200)           % Stop after 200 frames

```

Όπως αναφέρθηκε και προηγουμένως, οι εικόνες αποθηκεύονται στη μνήμη του υπολογιστή. Επομένως χρειαζόμαστε πρόσβαση στην μνήμη. Αυτό επιτυγχάνεται με την εντολή **getdata**. Η εντολή **getdata**, μπορεί να έχει την εξής σύνταξη:

```

[data,time] = getdata(...)

```

Σε αυτήν την περίπτωση, παρέχονται δυο έξοδοι στα αποτελέσματα. Η πρώτη είναι τα δεδομένα που αποθηκεύτηκαν στη μνήμη. Επιπλέον όμως, δίνεται και ο σχετικός χρόνος απόκτησης κάθε “πακέτου” δεδομένων στη μνήμη, ως προς την έναρξη απόκτησης τους. Αυτό

είναι πολύ σημαντικό, γιατί μπορούμε να γνωρίζουμε με πολύ μεγάλη ακρίβεια, κάθε πότε αποθηκεύτηκε το κάθε πακέτο δεδομένων, δηλαδή η κάθε εικόνα, στη μνήμη.

Επειδή κάθε φορά, ο αλγόριθμος επεξεργάζεται μόνο την πιο πρόσφατη εικόνα, χρειαζόμαστε μόνο την τελευταία εικόνα που αποθηκεύτηκε στη μνήμη. Στον υπολογιστή που έτρεξε ο κώδικας, οι εικόνες αποθηκεύονται με βάση το χρωματικό μοντέλο **YCbCr**. Επομένως θα πρέπει να μετατρέψουμε την εικόνα που ανακαλούμε από τη μνήμη, στο χρωματικό μοντέλο **RGB**. Είδαμε και σε προηγούμενο σημείο ότι αυτό επιτυγχάνεται με την εντολή **ycbcr2rgb**. Αν ο υπολογιστής αποθηκεύει τις εικόνες στο χρωματικό μοντέλο **RGB**, τότε φυσικά η εντολή αυτή παραλείπεται. Η εικόνα που ανακλήθηκε από τη μνήμη, στο κατάλληλο πλέον χρωματικό μοντέλο, αποθηκεύεται όπως όλα τα δεδομένα, με μορφή πίνακα, που ονομάζεται **data**. Οι εντολές που εκτελούν τις παραπάνω λειτουργίες δίνονται στη συνέχεια:

```
[data time] = getdata(vid,1);           % Grab last frame in the
buffer
data = ycbcr2rgb(data);
```

Στη συνέχεια, λαμβάνει χώρα η σημαντικότερη ίσως, πράξη στο σύνολο του κώδικα. Αυτή είναι η πράξη της συσχέτισης (ή ετεροσυσχέτιση – correlation or cross-correlation). Όπως έχει αναφερθεί, το αποτέλεσμα της συσχέτισης είναι, ο βαθμός ομοιότητας δυο πινάκων στα διάφορα σημεία τους. Πρώτο βήμα για την εξέλιξη της πράξης είναι η μετατροπή των σημάτων στο πεδίο της συχνότητας με βάση τον μετασχηματισμό Fourier. Η μετατροπή γίνεται αυτόματα κατά την εκτέλεση της εντολής. Πιο αναλυτικά, ένα μέρος της εντολής είναι η **xcorr2**. Η εντολή αυτή υπολογίζει την ετεροσυσχέτιση δυο πινάκων έστω A και B, σε δύο διαστάσεις. Το αποτέλεσμα που προκύπτει δεν είναι σε κλίμακα, δηλαδή δεν είναι κανονικοποιημένο. Είναι όμως κατάλληλη για διδιάστατους πίνακες, όπως είναι στη συγκεκριμένη εφαρμογή οι εικόνες. Εδώ πρέπει να αναφερθεί ότι οι εικόνες που λαμβάνονται είναι σε μορφή RGB, δηλαδή απεικονίζονται σαν τρισδιάστατοι πίνακες (x,y,3). Οι τρεις διαστάσεις αφορούν κάθε μια χρωματική συνιστώσα του μοντέλου RGB. Όμως ο αλγόριθμος για την επεξεργασία κρατάει μόνο την πρώτη. Επομένως οι δυο εικόνες στις οποίες εκτελείται η συσχέτιση είναι πλέον διδιάστατες. Το αποτέλεσμα που προκύπτει εμφανίζει την μέγιστη τιμή του, στο σημείο όπου οι δυο εικόνες εμφανίζουν τη μέγιστη ομοιότητα. Αναλυτικά για την πράξη της συσχέτισης, έχουμε αναφερθεί σε προηγούμενο κεφάλαιο. Η εντολή συμπληρώνεται με την κανονικοποίηση του αποτελέσματος, στο διάστημα [-1,1].

```
c =
normxcorr2(sub_T(:, :, 1), data(roi_ystart:roi_yend, roi_xstart:roi_xend, 1))
;
% Correlation in the frequency domain
c2 =
normxcorr2(sub_T2(:, :, 1), data(roi_ystart:roi_yend, roi_xstart:roi_xend, 1)
);
```

Η γενική σύνταξη της εντολής είναι:

c = normxcorr2(template,A)

όπου **template** είναι η εικόνα-πρότυπο, την οποία και αναζητούμε. Α είναι η ευρύτερη εικόνα, στην οποία αναζητούμε την εικόνα-πρότυπο. Για να έχει νόημα η πράξη, θα πρέπει ο πίνακας Α να είναι μεγαλύτερος από τον πίνακα **template**. Όπως βλέπουμε στη σύνταξη του κώδικα, ο πίνακας **template** που αναζητούμε είναι ο **sub_T(:, :, 1)** και ο **sub_T2(:, :, 1)**, δηλαδή αυτοί που αρχικά είχαμε ορίσει ως τον κύριο και τον βοηθητικό στόχο. Επιπλέον, επιλέγουμε μόνο την πρώτη από τις τρεις συνιστώσες, όπως αναφέρθηκε, μιας και η εικόνα έχει χρωματικό μοντέλο το **RGB**. Εδώ έγινε η παραδοχή ότι, όποια από τις τρεις συνιστώσες επιλέξουμε, θα έχει κατά μέσο όρο την ίδια επίδραση στο αποτέλεσμα. Ο αντίστοιχος πίνακας Α, είναι η τελευταία εικόνα που αποθηκεύτηκε στη μνήμη, με το κατάλληλο χρωματικό μοντέλο, δηλαδή η μεταβλητή **data**, όπως αυτή αναφέρθηκε προηγουμένως. Και στην μεταβλητή **data**, επιλέγεται μόνο η πρώτη από τις τρεις χρωματικές συνιστώσες με βάση το χρωματικό μοντέλο **RGB**. Όμως, όπως είχαμε αναφέρει, δεν θα γίνει η πράξη σε όλο το εύρος της εικόνας **data**, αλλά μόνο στην περιοχή ενδιαφέροντος – **ROI**. Η περιοχή **ROI**, για την πρώτη εικόνα έχει επιλεγεί από τον χρήστη στο στάδιο αρχικοποίησης, και στη συνέχεια ανανεώνεται αυτόματα όπως έχει αναφερθεί. Δηλαδή, ο χρήστης ανοίγει ένα "παράθυρο" στην εικόνα **data**. Ο αλγόριθμος δεν ασχολείται με οτιδήποτε υπάρχει έξω από αυτό, αλλά κάνει πράξεις μόνο για τα σημεία που περικλείονται μέσα σε αυτό το παράθυρο. Το παράθυρο μετακινείται πάνω στην εικόνα **data**, με βάση την κίνηση του στόχου.

Τώρα ο αλγόριθμος έχει συσχετίσει τους δυο στόχους στην περιοχή ενδιαφέροντος. Έχει σχηματιστεί ένας διδιάστατος πίνακας **c**, και ένας διδιάστατος πίνακας **c2**, οι οποίοι έχουν διαστάσεις ίσες με τις διαστάσεις της περιοχής ενδιαφέροντος **ROI**. Τα στοιχεία καθενός από τους πίνακες **c** και **c2**, είναι ο βαθμός ομοιότητας των δυο εικόνων-προτύπων, σε κάθε σημείο της περιοχής **ROI**, αντίστοιχα.

Για την περαιτέρω επεξήγηση του αλγορίθμου, θα αναφερθούμε μόνο στον βασικό στόχο για λόγους απλότητας. Προφανώς ισχύουν τα ίδια και για τον δεύτερο και βοηθητικό στόχο.

Το αποτέλεσμα της ομοιότητας υπάρχει πάντα, ακόμα και όταν οι εικόνες διαφέρουν εντελώς μεταξύ τους. Έτσι για μηδενικό βαθμό ομοιότητας σε ένα σημείο, το αντίστοιχο στοιχείο του πίνακα **c**, θα είναι -1. Για το σημείο που οι δυο εικόνες, ενδεχομένως, ταυτίζονται πλήρως το αντίστοιχο στοιχείο του πίνακα **c**, θα είναι ίσο με 1. Επομένως, θα πρέπει να βρούμε το σημείο που εμφανίζεται ο μέγιστος βαθμός ομοιότητας, δηλαδή το μεγαλύτερο στοιχείο του **c**. Θα πρέπει να σημειωθεί ότι, δεν είναι πρακτικό να ψάξουμε για σημείο με απόλυτη ομοιότητα. Έστω και ανεπαίσθητες διαφορές (π.χ. στον φωτισμό), δίνουν ως αποτέλεσμα στον **c**, στοιχεία μικρότερα από 1, δηλαδή όχι πλήρη ταύτιση. Άρα, αρκεί να βρούμε το στοιχείο εκείνο στο οποίο εμφανίζεται η μεγαλύτερη ομοιότητα. Το σημείο αυτό είναι εκείνο, όπου έχει εντοπιστεί ο στόχος μας. Αυτό επιτυγχάνεται με την εντολή,

```
[max_c, imax] = max(c(:));
```

Η συγκεκριμένη εντολή, εντοπίζει την μέγιστη τιμή του πίνακα **c**, καθώς και σε ποιά θέση εντοπίστηκε. Με **max_c** ονομάζεται η μέγιστη τιμή του **c**, ενώ **imax** ονομάζεται η θέση του σημείου, δηλαδή οι συντεταγμένες του. Έπομένως θα πρέπει να εντοπίσουμε σε ποιά θέση βρέθηκε η μέγιστη ομοιότητα. Δηλαδή σε ποιά γραμμή και σε ποιά στήλη βρίσκεται η μέγιστη τιμή του πίνακα **c**. Η εντολή που μας παρέχει τις συντεταγμένες του συγκεκριμένου σημείου είναι η **ind2sub**. Η ακριβής σύνταξη έχει την μορφή

```
[x, y] = ind2sub(siz,IND);
```

Η έξοδος της εντολής, είναι δύο πίνακες **x** και **y**, οι οποίοι περιέχουν τις γραμμές και τις στήλες αντίστοιχα, των στοιχείων ενός πίνακα μεγέθους **siz**, που ικανοποιούν την συνθήκη **IND**. Για να γίνει πλήρως κατανοητή η λειτουργία της εντολής **ind2sub**, θα εξετάσουμε τον παρακάτω πίνακα **A**:

$$A = \begin{bmatrix} 1 & 2 & 8 \\ 5 & 3 & 6 \\ 9 & 4 & 7 \end{bmatrix}$$

Έστω ότι θέλουμε να βρούμε που βρίσκεται εκείνο το στοιχείο του πίνακα **A**, το οποίο είναι ίσο με 4. Τότε η σύνταξη της εντολής θα είναι:

```
[x, y] = ind2sub(size(A),find(A(:,:)==4))
```

και η έξοδος θα είναι,

x = 3 και **y = 2**. Οι **x** και **y** είναι δυο μονοδιάστατοι πίνακες. Ο πρώτος αντιστοιχεί στην γραμμή στην οποία βρίσκεται το στοιχείο, και ο δεύτερος στην αντίστοιχη στήλη. Προφανώς, αν αναζητούσαμε τα στοιχεία εκείνα του **A**, τα οποία είναι για παράδειγμα μεγαλύτερα ή ίσα από το 6, οι πίνακες **x** και **y** θα περιείχαν 4 στοιχεία, όσα δηλαδή ικανοποιούν αυτήν την συνθήκη στον πίνακα **A**.

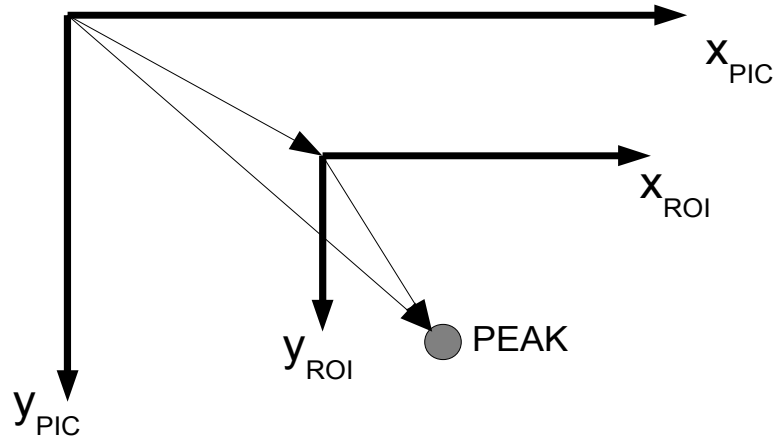
Η σύνταξη της εντολής που χρησιμοποιήσαμε στον κώδικα είναι,

```
[ypeak, xpeak] = ind2sub(size(c),imax);
```

Εντύπωση ίσως να προκαλεί το γεγονός ότι πρώτα δηλώνεται η στήλη και στη συνέχεια η σειρά. Αυτό οφείλεται στον τρόπο με τον οποίο αναπαριστώνται τα δεδομένα εικόνων στην MATLAB, όπως έχει αναφερθεί και προηγούμενα.

Κάτι που θα πρέπει να διαρκώς να υπενθυμίζεται, είναι το γεγονός πως, μέχρι τώρα βρισκόμαστε μέσα στην περιοχή ROI. Δηλαδή οι συντεταγμένες (**xpeak,ypeak**) δίνονται ως προς το σύστημα αναφοράς που ορίζεται από την περιοχή ενδιαφέροντος. Η αρχή αυτού του συστήματος συντεταγμένων, ως προς το σύστημα συντεταγμένων της εικόνας, είναι το πάνω αριστερά άκρο της περιοχής ROI. Υπενθυμίζεται εδώ ότι την περιοχή αυτή την έχει ορίσει ο χρήστης στην διαδικασία αρχικοποίησης του κώδικα. Για να εντοπίσουμε τα αποτελέσματα στο σύνολο της εικόνας, θα πρέπει να βρούμε και την σχετική θέση της περιοχής ROI, ως προς το σύστημα συντεταγμένων της ευρύτερης εικόνας. Το σημείο μέγιστης ομοιότητας (**peak**), είναι

εκφρασμένο στο σύστημα αναφοράς της περιοχής ROI. Αν θέλουμε να εκφράσουμε τις συντεταγμένες του σημείου **peak**, ως προς το σύστημα αναφοράς της εικόνας, θα πρέπει να λάβουμε υπ'όψη την σχετική θέση των δυο συστημάτων αναφοράς.



Σχήμα 4-14: Σημείο Μέγιστης Ομοιότητας (**peak**) και Συστήματα Συντεταγμένων.

Στην εικόνα φαίνεται, το σύστημα συντεταγμένων της εικόνας ($x_{\text{PEAK}}, y_{\text{PEAK}}$), και το σύστημα συντεταγμένων της περιοχής **ROI** ($x_{\text{ROI}}, y_{\text{ROI}}$). Επομένως οι συντεταγμένες του σημείου **peak**, ως προς το σύστημα συντεταγμένων της εικόνας ($x_{\text{PIC}}, y_{\text{PIC}}$) θα είναι:

$$(x, y) = (x_{\text{peak}} + \text{rect_roi}(1), y_{\text{peak}} + \text{rect_roi}(2)).$$

Όπως αναφέρθηκε ($x_{\text{peak}}, y_{\text{peak}}$) είναι οι συντεταγμένες του σημείου **peak**, ως προς το σύστημα συντεταγμένων που ορίζει η περιοχή **ROI**. Ακόμα **rect_roi(1)** και **rect_roi(2)**, είναι οι μετατοπίσεις του συστήματος συντεταγμένων της περιοχής ROI, ως προς το σύστημα συντεταγμένων της ευρύτερης εικόνας. Οι μεταβλητές **rect_roi(1)** και **rect_roi(2)** πρέπει να υπενθυμιστεί ότι έχουν ονομαστεί **roi_xstart** και **roi_ystart**, και με αυτά τα ονόματα θα αναφέρονται στο εξής. Όλες αυτές οι μετατροπές περιγράφονται στις παρακάτω εξισώσεις του κώδικα:

```
xgreen = round(xpeak+roi_xstart-xlength);
ygreen = round(ypeak+roi_ystart-ylength);
% Calculate the current position of the target.
cpos = [xgreen+(xlength/2), ygreen+(ylength/2)];
timetable(i) = time;
```

Στην τελευταία γραμμή συμπληρώνεται ο πίνακας **timetable**, με τις τιμές της μεταβλητής **time**. Το ποιά θέση του πίνακα **timetable**, συμπληρώνεται κάθε φορά, καθορίζεται από την τιμή

του μετρητή i.

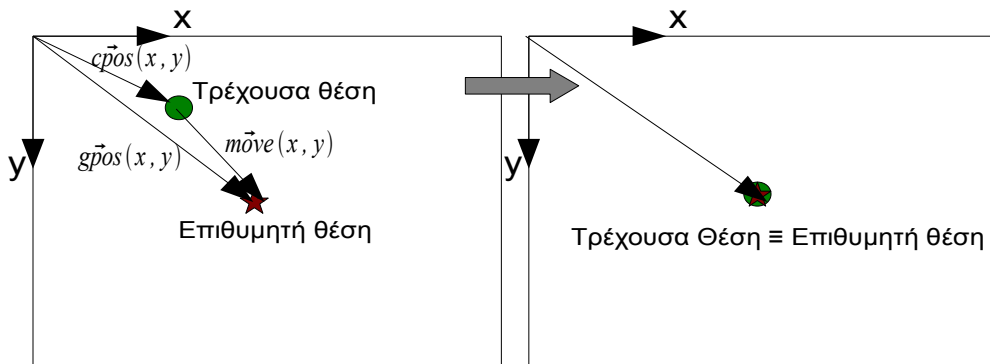
Στις παραπάνω εξισώσεις, με **cp** συμβολίζονται οι τρέχουσες συντεταγμένες του σημείου **peak**, ως προς το σύστημα αναφοράς της εικόνας. Στην περίπτωση πρόσδεσης της κάμερας στο φορείο, ο σκοπός είναι όταν ο αλγορίθμος εντοπίζει το ρομποτικό ψάρι να το παρακολουθεί. Δηλαδή, να διατηρεί την κάμερα, σχεδόν πάνω από τον στόχο. Επειδή το ρομπότ κινείται στο επίπεδο, θα πρέπει το φορείο να μπορεί να εκτελεί κινήσεις και στους δυο άξονες. Η κίνηση επιτυγχάνεται με την βοήθεια ενός κινητήρα στον κάθε άξονα. Η λειτουργία κάθε κινητήρα καθορίζεται από έναν ελεγκτή. Το σήμα εισόδου στον ελεγκτή, θα πρέπει να είναι η επιθυμητή μετατόπιση κατά τον αντίστοιχο άξονα. Η επιθυμητή μετατόπιση, είναι η απόσταση από την τρέχουσα σχετική θέση της κάμερας ως προς τον στόχο, ως την θέση κατά την οποία, η κάμερα βρίσκεται ακριβώς πάνω από τον στόχο. Σε αυτήν την τελευταία θέση, ο στόχος θα εντοπίζεται ακριβώς στο μέσο της εικόνας. Στην εικόνα που ακολουθεί, φαίνεται η διαδικασία διατήρησης του στόχου κοντά στην επιθυμητή θέση. Η διαδικασία αυτή ολοκληρώνεται και επαναλαμβάνεται διαρκώς, σε ολη τη διάρκεια λειτουργίας του αλγορίθμου.

Στο Σχήμα 4-15, με $c\vec{p}os(x, y)$ δηλώνεται η τρέχουσα θέση του στόχου στην εικόνα, ως προς το σύστημα συντεταγμένων της εικόνας. Με $g\vec{p}os(x, y)$, δηλώνεται η επιθυμητή θέση του στόχου, δηλαδή το μέσο της εικόνας. Το διάνυσμα $m\vec{o}ve(x, y)$, είναι η μετατόπιση που πρέπει να εκτελεστεί ώστε η τρέχουσα θέση και η επιθυμητή να συμπίσουν. Τότε ο στόχος θα βρίσκεται ακριβώς κάτω από την κάμερα, και θα εμφανίζεται στο μέσο της εικόνας. Το διάνυσμα $m\vec{o}ve(x, y)$, βρίσκεται από τη γνωστή σχέση της Αναλυτικής Γεωμετρίας

$$\vec{x} = x_{\text{τελικό}} - x_{\text{αρχικό}} \quad (4-46)$$

και για την συγκεκριμένη περίπτωση,

$$m\vec{o}ve(x, y) = g\vec{p}os(x, y) - c\vec{p}os(x, y) \quad (4-47)$$



Σχήμα 4-15: Διαδικασία διατήρησης του στόχου κοντά στην επιθυμητή θέση.

Οι γραμμές στον κώδικα που εκτελούν τις παραπάνω λειτουργίες είναι:

```
% Calculate the desired move in both axes.
movex(i) = gpos(1) - cpos(1);
movey(i) = gpos(2) - cpos(2);
move = [movex, movey];
```

Όπως φαίνεται, οι πράξεις γίνονται ξεχωριστά για τον κάθε άξονα. Στη συνέχεια, οι δυο συνιστώσες **movex** και **movey**, σχηματίζουν το διδιάστατο διάνυσμα **move**. Οι μονάδες των movex και movey, είναι τα **pixels**. Επομένως για να αντιστοιχίσουμε τα νούμερα αυτά με τις πραγματικές αποστάσεις, θα πρέπει να βρούμε την αντιστοίχιση του ενός pixel σε mm, ή σε cm. Έτσι, αφού βρούμε τις πραγματικές αποστάσεις, μπορούμε τα movex και movey, να τα στέλνουμε ως σήματα εισόδου στους κινητήρες των αντίστοιχων αξόνων. Με τον τρόπο αυτό, η κάμερα διατηρεί τον στόχο, σε μόνιμη βάση, πολύ κοντά στο μέσο της εικόνας της.

Ακόμα, για να μελετήσουμε την πορεία του στόχου, δηλαδή την τροχιά που διέγραψε, εξετάζουμε τη θέση του. Η θέση του δίνεται σα διάνυσμα στο επίπεδο, με σύστημα αναφοράς, το σύστημα συντεταγμένων της εικόνας.

```
xreal(i) = cpos(1);
yreal(i) = cpos(2);
real = [xreal, yreal];
```

Όλα τα προαναφερθέντα, αφορούν τον πρώτο και βασικό στόχο. Τα ίδια πρέπει να εφαρμοστούν και για τον δεύτερο-βοηθητικό στόχο. Ιδιαίτερη επεξήγηση δεν χρειάζεται, καθώς η διαδικασία είναι ίδια και για τους δυο στόχους. Οι γραμμές του κώδικα που αφορούν τον βοηθητικό στόχο δίνονται στη συνέχεια:

```

[max_c2, imax2] = max(c2(:));
[ypeak2, xpeak2] = ind2sub(size(c2),imax2);
% sub_T = data((ypeak-ylength:ypeak),(xpeak-xlength:xpeak),:);
corr_offset2 = [(xpeak2-size(sub_T2,2)), (ypeak2-size(sub_T2,1))];
xgreen2 = round(xpeak2+roi_xstart-xlength2);
ygreen2 = round(ypeak2+roi_ystart-ylength2);
% Calculate the current position of the target.
cpos2 = [xgreen2+(xlength2/2), ygreen2+(ylength2/2)];
% Calculate the desired move in both axes.
xreal2(i) = (cpos2(1))*1.1196;
yreal2(i) = (cpos2(2))*1.1196;
real2 = [xreal2, yreal2];

```

Όπως φαίνεται, η διαδικασία είναι ίδια με αυτήν που περιγράφηκε και για τον πρώτο στόχο. Το επόμενο κομμάτι του κώδικα, είναι επιφορτισμένο με την αναπαράσταση των αποτελεσμάτων, παράλληλα με την διαδικασία της εικονοροής. Έτσι, με τις επόμενες εντολές, αναπαράγονται οι εικόνες που στέλνει η κάμερα, με την μορφή εικονοροής (video). Πάνω σε αυτές τις εικόνες, εμφανίζεται το σημείο μέγιστης ομοιότητας για τους δυο στόχους, με δυο μπλε αστέρια - ένα για τον κάθε στόχο. Επιπλέον επισημαίνονται οι περιοχές γύρω από το σημείο μέγιστης ομοιότητας, με ένα πράσινο ορθογώνιο για τον βασικό στόχο και με ένα κίτρινο ορθογώνιο για τον βοηθητικό. Το κάθε ορθογώνιο έχει διαστάσεις, ίδιες με τις διαστάσεις της αντίστοιχης εικόνας-στόχου. Στην ουσία, η περιοχή μέσα στο κάθε ορθογώνιο, είναι η περιοχή της εικόνας, η οποία εμφανίζει την μεγαλύτερη ομοιότητα με την αντίστοιχη εικόνα-στόχο. Ακόμα, εμφανίζεται και επιθυμητή θέση για τον στόχο, δηλαδή το μέσο της εικόνας, με έναν κόκκινο κύκλο. Τέλος, δηλώνεται με κόκκινο ορθογώνιο, η περιοχή ενδιαφέροντος, μέσα στην οποία γίνεται η αναζήτηση των στόχων, όπως έχει αναφερθεί. Με τον τρόπο αυτό, οπτικοποιούνται καλύτερα τα αποτελέσματα που εξάγονται από τον αλγόριθμο, και γίνονται πιο άμεσα και κατανοητά.

```

imshow(data, 'InitialMagnification', 'fit')
%imagesc(data)
%drawnow
hold on
rectangle('Position', [xgreen,ygreen,xlength,ylength], ...
    'EdgeColor','g');
rectangle('Position', [xgreen2,ygreen2,xlength2,ylength2], ...
    'EdgeColor','y');
rectangle('Position', [roi_xstart,roi_ystart,roi_width,roi_height],
...
    'EdgeColor','r')
plot(cpos(1),cpo(2), 'b*')
plot(cpos2(1),cpo2(2), 'b+')
hold off
i = i+1;

```

Ακόμα, μόλις ολοκληρωθεί η επεξεργασία κάθε εικόνας, αυξάνεται η τιμή του μετρητή **i**. Ο

μετρητής *i* δείχνει σε ποιά θέση θα αποθηκεύονται τα αποτελέσματα που προκύπτουν από την επεξεργασία κάθε εικόνας. Γενικά οποιοδήποτε αριθμητικό αποτέλεσμα στη MATLAB, αποθηκεύεται υπό μορφή πίνακα, όπως έχει αναφερθεί. Τα αποτελέσματα, όπως αυτά που αφορούν στη θέση του στόχου, στη σχετική του θέση ως προς το μέσο της εικόνας, αποθηκεύονται σε πίνακες. Οι πίνακες αυτοί είναι μονοδιάστατοι, δηλαδή έχουν μόνο μια στήλη. Ο αριθμός των γραμμών τους, εξαρτάται από τον αριθμό των δεδομένων που θέλουμε να αποθηκεύσουμε. Για παράδειγμα, αν θέλουμε να επεξεργαστούμε 200 εικόνες (frames), θα εξαχθούν 200 αποτελέσματα για την απόλυτη θέση του στόχου, ένα για κάθε εικόνα. Ο μετρητής *i*, φροντίζει ώστε αυτά να αποθηκευτούν στις αντίστοιχες θέσεις στους πίνακες για τους οποίους προορίζονται. Επομένως, όταν τελειώσει η επεξεργασία μιας εικόνας, τα δεδομένα αποθηκεύονται στην αντίστοιχη γραμμή των πινάκων. Έπειτα ο μετρητής αυξάνει κατά ένα. Έτσι τα δεδομένα που θα προκύψουν από την επεξεργασία της επόμενης εικόνας, θα αποθηκευτούν στην επόμενη γραμμή. Η διαδικασία αυτή συνεχίζεται μέχρι να ολοκληρωθεί η λειτουργία του αλγορίθμου. Διαφυλάσσεται έτσι η σωστή αποθήκευση των δεδομένων.

Το επόμενο τμήμα του κώδικα, περιέχει μια πολύ σημαντική λειτουργία του, δηλαδή την ανανέωση της εικόνας-προτύπου. Όπως αναφέρθηκε, η εισαγωγή της εικόνας-πρότυπο, είναι μια δυναμική διαδικασία. Δηλαδή η εικόνα-πρότυπο κάθε στόχου, αλλάζει διαρκώς, αλλά με ένα τρόπο εσωτερικό που δεν γίνεται αντιληπτός από τον χρήστη. Αυτή η διαρκής μεταβολή γίνεται, ώστε ο αλγόριθμος να έχει πάντα την καλύτερη δυνατή εικόνα του στόχου, και να βελτιστοποιείται έτσι, κατά κάποιο τρόπο, η διαδικασία αναζήτησης. Στη συνέχεια περιγράφεται αναλυτικά η διαδικασία ανανέωσης της εικόνας-πρότυπο. Όπως είδαμε, ένα από τα αποτελέσματα που εξάγει ο αλγόριθμος για τον βασικό στόχο, είναι μεταξύ άλλων και το πράσινο ορθογώνιο, πάνω στην αλληλουχία των εικόνων. Το ορθογώνιο αυτό είναι η περιοχή γύρω από το σημείο μέγιστης ομοιότητας. Έτσι, μετά την ολοκλήρωση της επεξεργασίας κάθε εικόνας, η περιοχή που περικλείεται από αυτό το ορθογώνιο, δηλώνεται ως η νέα εικόνα-πρότυπο, για την επεξεργασία της επόμενης εικόνας. Με τον τρόπο αυτό, ο αλγόριθμος διατηρεί πάντα μια ανανεωμένη εικόνα του στόχου. Δηλαδή, μέσω των μικρών αλλαγών που εμφανίζονται από εικόνα σε εικόνα, ο αλγόριθμος είναι σε θέση να διαχειρίζεται μεγαλύτερες αλλαγές στην εικόνα πρότυπο σε πιο μακροπρόθεσμη βάση. Οι αλλαγές αυτές στην ουσία αντιμετωπίζονται ως άθροισμα μικρότερων και πιο διαχειρίσιμων.

Η λειτουργία ανανέωσης της εικόνας-προτύπου όμως, εμφανίζει ένα σημαντικό πρόβλημα. Αν για κάποιο λόγο χαθεί ή δεν αναγνωρίσει τον στόχο στιγμιαία, στη συνέχεια δεν μπορεί να επανέλθει στη κανονική λειτουργία. Υπάρχουν πολλοί λόγοι, για τους οποίους ο αλγόριθμος μπορεί στιγμιαία να χάσει το στόχο. Για παράδειγμα, μπορεί ο στόχος να μετακινηθεί έξω από την περιοχή ενδιαφέροντος. Ακόμα, στιγμιαία εμπόδια μπροστά από την κάμερα, ή και στιγμιαίες αλλά μεγάλες αλλαγές στο φωτισμό. Στην περίπτωση αυτή, ο αλγόριθμος δεν θα βρίσκει την δοσμένη εικόνα-πρότυπο. Παρόλα αυτά όμως, θα συνεχίσει να ψάχνει την περιοχή εκείνη της εικόνας που

θα εμφανίζει την μέγιστη ομοιότητα, με την δοσμένη. Η νέα αυτή εικόνα, θα είναι εκείνη με την μεγαλύτερη ομοιότητα στο πεδίο της συχνότητας με τη δοσμένη. Όμως στην ουσία, δεν θα είναι η εικόνα που ψάχνουμε. Επιπλέον η νέα αυτή εικόνα, καταχωρείται ως εικόνα-πρότυπο για την επεξεργασία της επόμενης εικόνας. Έτσι, ακόμα και αν επανεμφανιστεί, η επιθυμητή εικόνα-πρότυπο, που εμείς έχουμε θέσει, ο αλγόριθμος θα την αγνοήσει, αφού πλέον έχει καταχωρημένη μια νέα εικόνα-πρότυπο. Η νέα αυτή εικόνα-πρότυπο δεν έχει καμία σχέση με αυτή που εμείς έχουμε ορίσει, και έτσι ο αλγόριθμος ψάχνει στη συνέχεια για κάτι διαφορετικό από αυτό που θα έπρεπε. Η περίπτωση αυτή είναι μεν ακραία, αφού πολύ δύσκολα μπορεί να συμβεί, ωστόσο δεν μπορεί να αποκλειστεί. Ειδικά στην περίπτωση της δεξαμενής, όπου ακριβώς από πάνω είναι τοποθετημένοι λαμπτήρες, οι αλλαγές στον φωτισμό είναι σημαντικές, λόγω των αντανάκλασεων στο νερό. Για να λυθεί αυτό το πρόβλημα, προγραμματίσαμε τον αλγόριθμο, να κάνει έναν έλεγχο στο τέλος της επεξεργασίας κάθε εικόνας. Έτσι, μόλις ολοκληρωθεί η επεξεργασία της εικόνας, ο αλγόριθμος ελέγχει τον βαθμό ομοιότητας, **max_c**, στο σημείο μέγιστης ομοιότητας που έχει βρει. Αν αυτός είναι μικρότερος από μια προκαθορισμένη τιμή, τότε ως εικόνα-πρότυπο για την επόμενη επεξεργασία, τίθεται ξανά η αρχική εικόνα πρότυπο (sub_Tinit). Η αρχική εικόνα-πρότυπο είναι αυτή που ορίστηκε από τον χρήστη στην διαδικασία αρχικοποίησης. Η προκαθορισμένη τιμή έχει τεθεί ίση με 0.87. Όπως έχει αναφερθεί η μέγιστη τιμή ομοιότητας είναι ίση με 1, όπου έχουμε απόλυτη ταύτιση. Έτσι θέτοντας ως τιμή κατωφλίου τον αριθμό 0.87, γίνονται επιτρεπτές μικρές αλλαγές του στόχου που έχει βρεθεί σε σχέση με την εικόνα-πρότυπο. Δηλαδή, δίνεται η δυνατότητα να ικανοποιούνται ταυτόχρονα δυο αντικρουόμενες μεταξύ τους απαιτήσεις. Από τη μία, πρέπει να επιτρέπονται μικρές διαφορές στην εικόνα που προκύπτει στα αποτελέσματα, σε σχέση με την αρχική. Από την άλλη, η επιτρεπτή διαφορά στην ομοιότητα πρέπει να μην είναι πολύ μεγάλη. Σε μια τέτοια περίπτωση, αν χάνονταν ο στόχος και στη συνέχεια επανέρχονταν στην εικόνα, ο αλγόριθμος θα συνέχιζε να τον αγνοεί. Πρέπει να υπενθυμίσουμε ότι ο βαθμός ομοιότητας **max_c**, βρίσκεται στο διάστημα [-1,1]. Έτσι, μπορεί μια άσχετη εικόνα, με την εικόνα-πρότυπο να παρουσιάζει βαθμό ομοιότητας της τάξης του 0.6. Αν έχουμε ορίσει τιμή κατωφλίου, για παράδειγμα, ίση με 0.5, τότε σε περίπτωση που χαθεί ο στόχος, ο αλγόριθμος θα συνεχίσει να τον αγνοεί, όταν επανέλθει στην εικόνα. Αυτό θα συμβαίνει γιατί, η διαφορά στον βαθμό ομοιότητας, είναι πάνω από την τιμή κατωφλίου. Επομένως, για το πρόγραμμα, υπάρχει κανονικά, εικόνα-πρότυπο. Άρα θα πρέπει να καλύπτονται και οι δυο αυτές απαιτήσεις, όσο το δυνατόν σε μεγαλύτερο βαθμό. Με τον τρόπο αυτό εξασφαλίζεται η ομαλή λειτουργία του προγράμματος, ακόμα και σε μια τέτοια ακραία περίπτωση.

Να σημειωθεί εδώ, ότι η τιμή 0.87, προέκυψε εμπειρικά, μέσω δοκιμών και αξιολόγησης της συμπεριφοράς του προγράμματος. Σίγουρα είναι ένα θέμα που επεδέχεται περαιτέρω διερεύνηση και βελτίωση. Έτσι ο αλγόριθμος όταν τελειώσει την επεξεργασία κάθε εικόνας, εκτελεί την διαδικασία ανανέωσης της εικόνας-πρότυπου. Συγκρίνει τον βαθμό μέγιστης ομοιότητας, μεταξύ της εικόνας του στόχου, που προέκυψε ως αποτέλεσμα της επεξεργασίας, και

της εικόνας-πρότυπο. Αν ο βαθμός μέγιστης ομοιότητας είναι μεγαλύτερος ή ίσος από 0.87, τότε ως νέα εικόνα-πρότυπο, τίθεται η περιοχή γύρω από το σημείο μέγιστης ομοιότητας που βρέθηκε στην εικόνα που μόλις επεξεργάστηκε. Η περιοχή αυτή, για τον βασικό στόχο, ορίζεται από το πράσινο ορθογώνιο, ενώ για το βοηθητικό, από το κίτρινο ορθογώνιο, όπως περιγράφηκε ανωτέρω. Αν ο βαθμός ομοιότητας είναι μικρότερος από 0.87, τότε θεωρείται ότι ο αλγόριθμος έχει χάσει το στόχο. Ως εικόνα-πρότυπο για την επόμενη αναζήτηση, τίθεται η αρχική εικόνα-πρότυπο (sub_Tinit), όπως αυτή έχει οριστεί από τον χρήστη στην φάση αρχικοποίησης. Οι αντίστοιχες γραμμές του κώδικα, που εκτελούν τις παραπάνω λειτουργίες είναι:

```
if max_c<0.87
    sub_T = sub_Tinit;
else
    sub_T = data(ygreen:ygreen+ylength,xgreen:xgreen+xlength,1);
end
if max_c2<0.87
    sub_T2 = sub_Tinit2;
else
    sub_T2 =
data(ygreen2:ygreen2+ylength2,xgreen2:xgreen2+xlength2,1);
end
```

Με την παραπάνω μεθοδολογία, δεν αντιμετωπίζεται το πρόβλημα σε όλη του την έκταση. Ακόμα και έτσι, υπάρχουν περιπτώσεις, κατά τις οποίες μπορεί να χαθεί στιγμιαία ο στόχος, και ο αλγόριθμος να μην μπορέσει να επανέλθει στην κανονική λειτουργία του. Μια τέτοια περίπτωση είναι να υπάρχει περιοχή στην εικόνα, η οποία να εμφανίζει βαθμό ομοιότητας με την εικόνα-πρότυπο μεγαλύτερο από 0.87. Η περίπτωση όμως αυτή είναι πάρα πολύ σπάνιο να συμβεί. Ακόμα κι αν συμβεί όμως, είναι πολύ εύκολο να αντιμετωπιστεί, αυξάνοντας την τιμή κατωφλίου στον κώδικα. Έτσι στην συντριπτική πλειοψηφία των δοκιμών και των πειραμάτων του προγράμματος, ο αλγόριθμος παρουσιάζει πολύ καλά αποτελέσματα.

Πριν ολοκληρωθεί ο αλγόριθμος, επιτελείται η διαδικασία ανανέωσης της περιοχής ενδιαφέροντος. Η διαδικασία αυτή περιγράφηκε αναλυτικά προηγούμενα. Επομένως εδώ θα αναφερθεί απλά, πώς αυτό υλοποιείται προγραμματιστικά στον κώδικα. Αρχικά, τα σημεία όπου βρέθηκε η μέγιστη ταύτιση με τους δυο στόχους, τίθενται ως βάση για την επόμενη αναζήτηση στην επόμενη εικόνα. Έτσι ταυτίζονται τα ονόματα των μεταβλητών που συμβολίζουν τα δυο αυτά σημεία, με τις μεταβλητές αυτές που καθορίζουν την περιοχή ενδιαφέροντος στην επόμενη αναζήτηση. Αυτό επιτυγχάνεται στις επόμενες γραμμές:

```
xpeak_im = xpeak;
ypeak_im = ypeak;
xpeak_im2 = xpeak2;
ypeak_im2 = ypeak2;
```

Στη συνέχεια, οι μεταβλητές **xpeak_im**, και **ypeak_im**, αποτελούν τις μεταβλητές εκείνες,

που μαζί με άλλες, καθορίζουν την θέση της περιοχής ενδιαφέροντος για την επόμενη αναζήτηση. Οι άλλες μεταβλητές είναι οι διαστάσεις της περιοχής ενδιαφέροντος, όπως αυτές έχουν οριστεί από τον χρήστη στην διαδικασία αρχικοποίησης. Ακόμα λαμβάνεται υπόψη και η σχετική θέση της περιοχής ενδιαφέροντος, ως προς το σύστημα συντεταγμένων της εικόνας. Αναλυτικά οι εξισώσεις, που εκτελούν αυτήν τη διαδικασία φαίνονται στις γραμμές που ακολουθούν:

```
roi_xstart = xpeak_im+roi_xstart+xlength/2 - roi_width/2;  
roi_ystart = ypeak_im+roi_ystart+ylength/2 - roi_height/2;  
roi_xend = roi_xstart + roi_width;  
roi_yend = roi_ystart + roi_height;
```

Στη συνέχεια, εκτελείται ένας μικρός βρόχος ελέγχου. Ο βρόχος αυτός, βοηθάει στην περίπτωση που ο στόχος έχει βγει εκτός της περιοχής ενδιαφέροντος. Αν συμβεί κάτι τέτοιο, τότε επανακαθορίζει την θέση της **ROI**, τοποθετώντας την στην αρχική της θέση. Ως αρχική θέση της **ROI** θεωρούμε την θέση που έχει ορίσει ο χρήστης στην διαδικασία αρχικοποίησης.

```
if max_c < 0.82 || max_c2 < 0.82  
    roi_xstart = rect_roi(1);  
    roi_ystart = rect_roi(2);  
    roi_xend = roi_xstart + roi_width;  
    roi_yend = roi_ystart + roi_height;  
end  
end
```

Δηλαδή στο τέλος της επεξεργασίας κάθε εικόνας, έχουν εκτελεστεί δύο έλεγχοι. Ο πρώτος εξετάζει την ομοιότητα των εικόνων προτύπων, με αυτές που έχουν προκύψει από την επεξεργασία. Αν ο βαθμός ομοιότητας (κάποιων εκ των δύο ή και των δύο) είναι μικρότερος από μια καθορισμένη τιμή, τότε ως νέος στόχος ορίζεται εκείνος που έχει οριστεί στην διαδικασία αρχικοποίησης. Ακόμα αν κάποιος από τους στόχους βγει έξω από την περιοχή ενδιαφέροντος, τότε ως νέα περιοχή ενδιαφέροντος, ορίζεται αυτή που έχει δοθεί από τον χρήστη στην διαδικασία αρχικοποίησης. Οι δυο αυτοί έλεγχοι, δεν εξασφαλίζουν πάντα ότι η διαδικασία θα εξελιχθεί ομαλά, ειδικά σε περιπτώσεις όπου οι στόχοι κινούνται ή περιστρέφονται πάρα πολύ γρήγορα. Για την περίπτωση όμως του ρομποτικού ψαριού, το οποίο κινείται ομαλά και σε χαμηλές σχετικά ταχύτητες, ο αλγόριθμος λειτουργεί πολύ καλά, για όλο το εύρος της κίνησης του.

Στο σημείο αυτό, ολοκληρώνεται η διαδικασία επεξεργασίας των εικόνων και η εξαγωγή αποτελεσμάτων. Ο κώδικας κλείνει, σβήνει και καθαρίζει τις μεταβλητές που αντιστοιχούν στην μονάδα εισαγωγής εικόνων δηλαδή στη κάμερα. Οι εντολές που ακολουθούν, σταματούν την λειτουργία της κάμερας, την διαγράφουν και καθαρίζουν την μεταβλητή με την οποία αναφέρεται, αντίστοιχα.

```
stop(vid)  
delete(vid)  
clear vid
```

Οι εντολές που ακολουθούν, αφορούν στην επεξεργασία των αποτελεσμάτων, με βάση τα δεδομένα που αποκτήθηκαν μέχρι τώρα. Με βάση τα δεδομένα αυτά μπορούμε να εξετάσουμε διάφορα στοιχεία της κίνησης του ψαριού, όπως για παράδειγμα την θέση που βρέθηκε κάθε στιγμή, την ταχύτητα του, και την γωνία περιστροφής του σώματος του, ως προς το σύστημα συντεταγμένων της εικόνας.

Οι μεταβλητές **xreal** και **yreal**, αποθηκεύουν την θέση του ψαριού, κατά x και κατά y άξονα αντίστοιχα. Δηλαδή σε ποιά θέση βρίσκεται ο στόχος σε σχέση με την αρχή του συστήματος συντεταγμένων της εικόνας (σε mm). Επομένως με τη βοήθεια αυτών των μεταβλητών, μπορούμε να εξετάσουμε την τροχιά στην οποία κινήθηκε το ψάρι. Μπορούμε επίσης να κάνουμε και τα αντίστοιχα διαγράμματα, που δείχνουν την κίνηση του ρομποτικού ψαριού στους δυο άξονες, ή στο επίπεδο με τις επόμενες εντολές:

```
plot(xreal)
plot(yreal)
plot(xreal,yreal)
```

Ακόμα μπορούμε να εξετάσουμε την γωνία περιστροφής του σώματος κατά την διάρκεια της κίνησης του, ως προς το σύστημα συντεταγμένων της εικόνας. Όπως αναφέρθηκε και προηγουμένως αυτό επιτυγχάνεται κάνοντας χρήση του τύπου:

$$\tan\theta = \frac{y_{\text{τελ.}} - y_{\text{αρχ.}}}{x_{\text{τελ.}} - x_{\text{αρχ.}}} \quad (4-48)$$

από τον οποίο προκύπτει η εφαπτομένη της γωνίας. Στη συνέχεια με τη βοήθεια της αντίστροφης εφαπτομένης βρίσκεται η γωνία σε ακτίνια. Τέλος, μετατρέπουμε τα ακτίνια σε μοίρες και βρίσκουμε το ζητούμενο. Επιπλέον μπορούμε να δούμε υπό μορφή διαγράμματος τα αποτελέσματα για την γωνία περιστροφής, καθ'όλη τη διάρκεια της κίνησης. Όλα αυτά πραγματοποιούνται με τη σειρά στις επόμενες γραμμές του κώδικα:

```
tantheta = (yreal2-yreal)./(xreal2-xreal);
theta = atan(tantheta);
theta = theta*180/pi;
plot(theta)
```

Ένα ακόμα στοιχείο που μπορεί να βρεθεί με βάση τα αποτελέσματα που έχουν προκύψει είναι η ταχύτητα του σώματος. Για το λόγο αυτό ορίστηκε ο πίνακας **timetable**, στον οποίο αποθηκεύονται ο χρόνος κατά τον οποίο λήφθηκε κάθε εικόνα από την κάμερα, και αποθηκεύτηκε στη μνήμη. Το κάθε στοιχείο του πίνακα, είναι ο χρόνος κατά τον οποίο αποθηκεύτηκε κάθε εικόνα στη μνήμη, σε σχέση με την έναρξη της αποθήκευσης. Επομένως για να βρούμε το χρονικό διάστημα που μεσολαβεί για την λήψη δυο διαδοχικών εικόνων, αρκεί να αφαιρέσουμε από την χρονική στιγμή, την προηγούμενη της. Το αποτέλεσμα αποθηκεύεται σε έναν πίνακα που τον ονομάζουμε **dt**. Το χρονικό διάστημα αυτό δεν είναι σταθερό, αφού εξαρτάται από διαφορετικούς παράγοντες. Οι διαφορές μεταξύ των διαστημάτων είναι της τάξης του 1-10 msec. Προσεγγιστικά

μπορούμε να θεωρήσουμε ότι ο χρόνος που μεσολαβεί από εικόνα σε εικόνα είναι ίσος με το μέσο όρο των στοιχείων του πίνακα **dt**. Για την συνέχεια των υπολογισμών, θα θεωρούμε ότι μεταξύ δυο διαδοχικών εικόνων, μεσολαβεί χρόνος **dt**, ίσος με τον μέσο όρο των στοιχείων του πίνακα **dt**, και όχι τα πραγματικά στοιχεία. Αυτό υλοποιείται με τις παρακάτω γραμμές κώδικα:

```
dt = zeros(199,1);
for i = 2:200
    dt(i) = timetable(i) - timetable(i-1);
end
dt = mean(dt);
```

Στο σημείο αυτό γνωρίζουμε τη θέση που βρίσκεται ο στόχος σε κάθε εικόνα, και τον χρόνο που μεσολαβεί από εικόνα σε εικόνα. Επομένως μπορούμε να γνωρίζουμε την ταχύτητα που έχει ο στόχος από εικόνα σε εικόνα, δηλαδή, στην ουσία, κατά τη διάρκεια της τροχιάς του. Η ταχύτητα βρίσκεται από τη γνωστή σχέση της νευτώνειας μηχανικής,

$$v = \frac{x_{\text{τελ.}} - x_{\text{αρχ.}}}{t_{\text{τελ.}} - t_{\text{αρχ.}}} = \frac{x_{\text{τελ.}} - x_{\text{αρχ.}}}{dt} \quad (4-49)$$

Μέσω αυτής της σχέσης, μπορεί να υπολογιστεί η ταχύτητα για κάθε άξονα κίνησης στο επίπεδο. Έτσι στη συνέχεια σχηματίζουμε το διάνυσμα ταχύτητας που αποτελείται από τις δυο συνιστώσες της, δηλαδή την ταχύτητα σε κάθε άξονα. Ακολουθούν οι γραμμές του κώδικα, που υλοποιούν αυτούς τους υπολογισμούς.

```
v_x = zeros(200,1);
v_y = zeros(200,1);

for k = 2:200
    v_x(k) = (xreal(k) - xreal(k-1)) / dt;
    v_y(k) = (yreal(k) - yreal(k-1)) / dt;
end

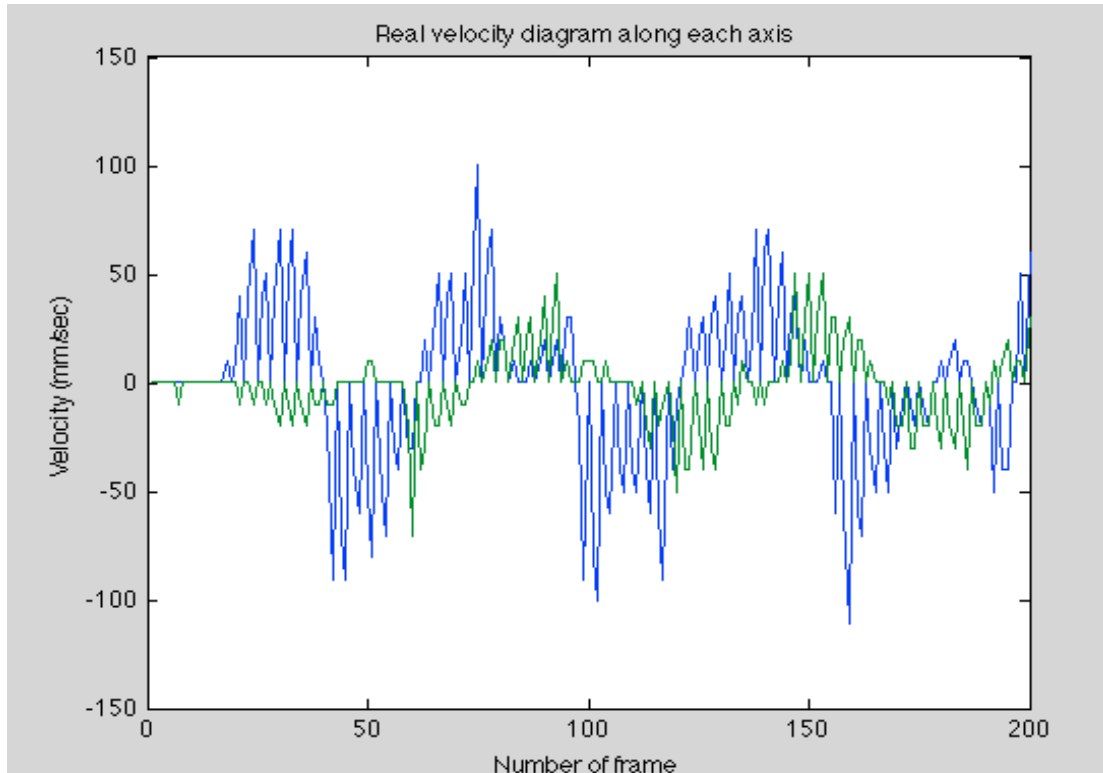
v = [v_x, v_y];
```

Στον πίνακα **v**, αποθηκεύονται οι τιμές της ταχύτητας του σώματος, με τις συνιστώσες της. Τα αποτελέσματα αυτά, μπορούμε να τα οπτικοποιήσουμε υπό μορφή διαγραμμάτων. Η αντίστοιχη εντολή είναι η:

```
plot(timetable, v_x, timetable, v_y)
```

Τα αποτελέσματα αυτής της εντολής, παρουσιάζουν κάποιες ιδιομορφίες. Συχνά ο στόχος από εικόνα σε εικόνα εντοπίζεται στο ίδιο εικονοστοιχείο. Αυτό οφείλεται κατά ένα μέρος, στο γεγονός ότι χρησιμοποιείται ο μέσος όρος των χρονικών διαστημάτων. Έτσι μπορεί σε δυο (ή και περισσότερες) διαδοχικές εικόνες ο στόχος να εμφανίζεται στο ίδιο σημείο χωρίς να έχει μετατοπιστεί. Στην πραγματικότητα μπορεί να έχει μετατοπιστεί, αλλά τόσο λίγο ώστε να καταγράφεται στο ίδιο εικονοστοιχείο. Δηλαδή δύο (ή περισσότερες) διαδοχικά, συντεταγμένες του να είναι ίδιες, σε τουλάχιστον έναν άξονα κίνησης. Αυτό έχει σαν αποτέλεσμα στα αντίστοιχα

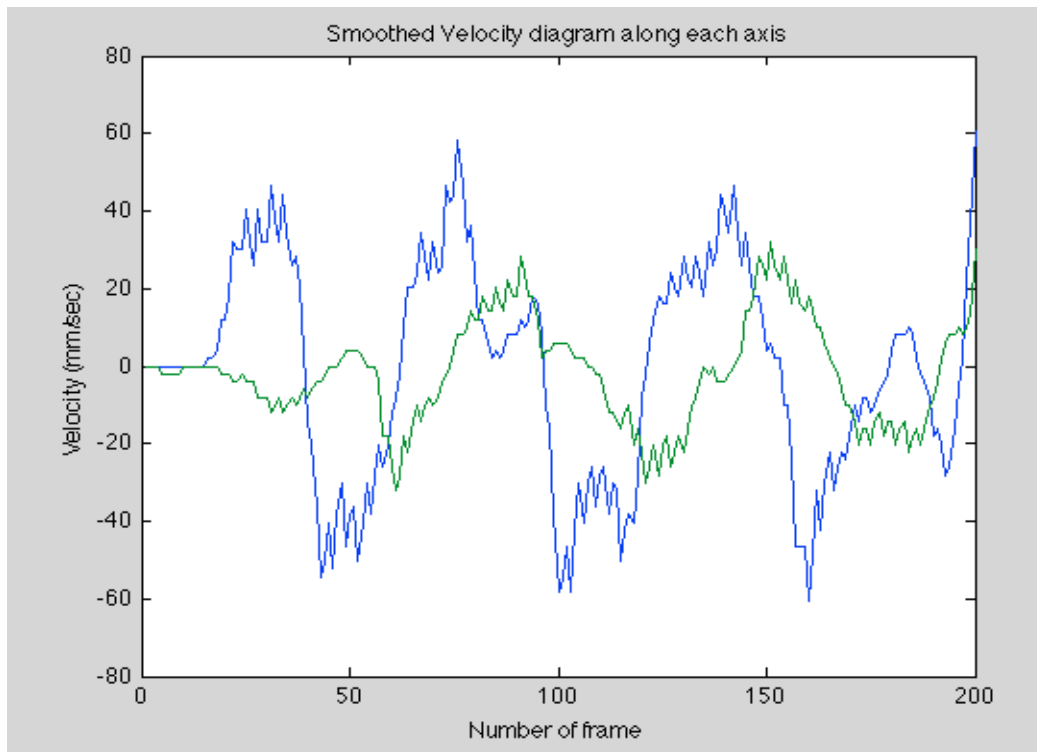
σημεία του διαγράμματος η ταχύτητα να είναι ίση με μηδέν, και η τελική μορφή του διαγράμματος, να εμφανίζει έντονες αυξομειώσεις, στην ταχύτητα. Δηλαδή, σε σημεία όπου η καμπύλη έχει μη μηδενικές τιμές, να πέφτει απότομα στο μηδέν για ένα ή περισσότερα σημεία, και στη συνέχεια να επανέρχεται απότομα στα προηγούμενα επίπεδα της ταχύτητας. Ένα τέτοιο διάγραμμα που προέκυψε από μετρήσεις φαίνεται στο Σχήμα 4-16.



Σχήμα 4-16: Πραγματικό διάγραμμα των δύο συνιστωσών της ταχύτητας.

Ένας τρόπος να μειώσουμε το φαινόμενο αυτό είναι να κρατήσουμε τα μη μηδενικά στοιχεία των πινάκων v_x και v_y . Όμως αυτή η μέθοδος δεν ενδείκνυται, αφού χάνεται σημαντική ποσότητα πληροφορίας, λόγω των πολλών μηδενικών στοιχείων. Με τον τρόπο αυτό από τις 200 εικόνες που έχει λάβει ο υπολογιστής κατά την διάρκεια της επεξεργασίας, καταλήγουμε με 100 ή και λιγότερες.

Έτσι, χρησιμοποιήσαμε μια στατιστική μέθοδο, που παρέχεται στην MATLAB, μέσω της εντολής **smooth**. Η εντολή αυτή, στην ουσία στρογγυλεύει τις αιχμές του διαγράμματος. Η λείανση γίνεται, χρησιμοποιώντας τον μέσο όρο των πέντε σημείων εκατέρωθεν του υπό εξέταση σημείου. Όμως με αυτό τον τρόπο υπάρχει πρόβλημα ακρίβειας. Για να γίνει αυτό πιο αντιληπτό δίνεται στο Σχήμα 4-17 το διάγραμμα που προέκυψε με την εντολή **smooth**, από το αντίστοιχο διάγραμμα του Σχήματος 4-16:



Σχήμα 4-17: Διάγραμμα δυο ταχυτήτων μετά την χρήση της εντολής smooth.

Όπως βλέπουμε στα σχήματα τα δυο διαγράμματα φαίνονται να μοιάζουν σε μεγάλο βαθμό. Ωστόσο, παρουσιάζουν πολύ σημαντικές διαφορές. Για παράδειγμα η μέγιστη τιμή στο Σχήμα 4-16, είναι κοντά στα 200 mm/sec, ενώ στο Σχήμα 4-17, μόλις ξεπερνάει τα 100 mm/sec.

Έτσι μια άλλη λύση για την ομαλοποίηση του διαγράμματος, είναι η γραμμική παρεμβολή μεταξύ των δυο μη μηδενικών στοιχείων, εκατέρωθεν ενός μηδενικού. Αυτό έγινε εφικτό με βάση τις εξής γραμμές κώδικα:

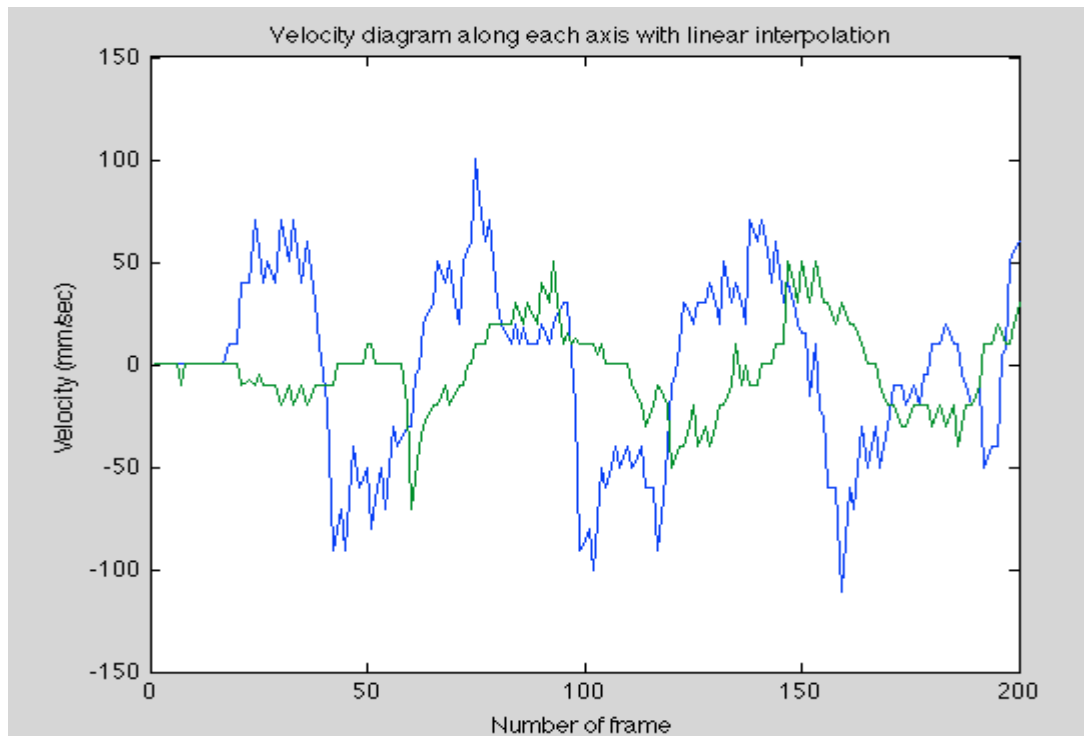
```

for k=2:199
if v_x(k-1) ~= 0 && v_x(k+1)~= 0 && v_x(k) == 0
v_x(k) = (v_x(k-1) + v_x(k+1))/2
else v_x(k) = v_x(k);
end
end

for k=2:199
if v_y(k-1) ~= 0 && v_y(k+1)~= 0 && v_y(k) == 0
v_y(k) = (v_y(k-1) + v_y(k+1))/2
else v_y(k) = v_y(k);
end
end

```

Τα αποτελέσματα που προέκυψαν από την γραμμική παρεμβολή στα μη μηδενικά στοιχεία του διαγράμματος, φαίνονται στο Σχήμα 4-18, που ακολουθεί:



Σχήμα 4-18: Διάγραμμα δυο ταχυτήτων μετά την γραμμική παρεμβολή στα μη μηδενικά στοιχεία.

Όπως φαίνεται στο σχήμα 4-18, το φαινόμενο των απότομων μεταβάσεων από και προς το μηδέν έχει εξαλειφθεί πλήρως, χωρίς σημαντική μείωση της ακρίβειας των αποτελεσμάτων.

5 Μετρήσεις – Πειραματικά Αποτελέσματα

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναφερθούν τα αποτελέσματα που προέκυψαν από την διαδικασία των μετρήσεων. Θα εξεταστεί μια περίπτωση κίνησης. Η κίνηση αυτή είναι ενδεικτική καθώς το ψάρι καλείται να εκτελέσει μια τροχιά τύπου “S”. Έτσι ο αλγόριθμος πρέπει να παρακολουθήσει το ρομποτικό ψάρι σε μια σύνθετη κίνηση, η οποία αναλύεται σε τρεις απλούστερες. Μια ελαφριά αριστερή στροφή, κίνηση σε ευθεία, και λίγο πριν την ολοκλήρωση μια ελαφριά δεξιά στροφή.

5.2 Μελέτη Κίνησης

Όπως αναφέρθηκε, η κίνηση του ψαριού που μελετάται, είναι σχετικά σύνθετη. Εκτελεί μια ελαφριά στροφή αριστερά και στη συνέχεια κινείται ευθύγραμμα και ακριβώς λίγο πριν το τέλος της εκτέλεσης του αλγορίθμου, εκτελεί μια ελαφριά δεξιά στροφή. Η εικόνα αρχικοποίησης στην εκτέλεση του αλγορίθμου δίνεται στη συνέχεια (Σχ. 5-1). Σε αυτή φαίνεται το ρομποτικό ψάρι πάνω δεξιά στην εικόνα, όπως αυτό φαίνεται από την κάμερα. Εδώ θα πρέπει να γίνει υπενθύμιση ότι η κάμερα είναι σταθερά στερεωμένη πάνω από τη δεξαμενή.

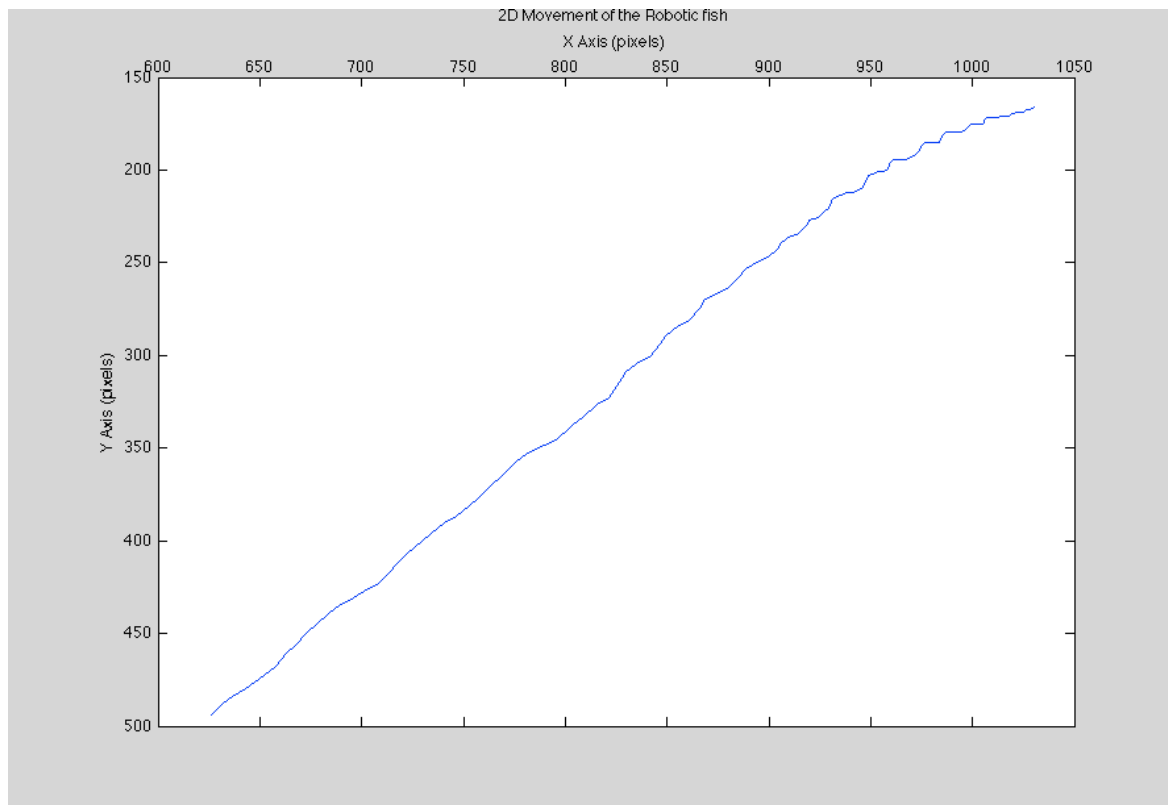
Στην εικόνα αρχικοποίησης, επιλέξαμε τις δυο εικόνες – πρότυπα. Ως πρώτο και βασικό στόχο, επιλέξαμε τμήμα του εμβόλου που βρίσκεται στο εσωτερικό του ψαριού, για τον έλεγχο του βάθους. Ως δεύτερος – βοηθητικός στόχος, επιλέχθηκε το κυλινδρικό τεμάχιο στο πίσω μέρος του ψαριού, μέσα στο οποίο βρίσκεται τοποθετημένος ο κινητήρας. Στο Σχήμα 5-1, εκτός από την εικόνα αρχικοποίησης, δίνονται και οι δυο στόχοι που επιλέχθηκαν. Οι στόχοι φαίνονται υπομεγένθυνη για να είναι πιο ευδιάκριτοι. Με τα αντίστοιχα βέλη μπορούμε να δούμε τα σημεία στο ρομποτικό ψάρι από τα οποία επιλέχθηκαν οι δυο εικόνες – πρότυπα.



Σχήμα 5-1: Εικόνα αρχικοποίησης πειράματος και στόχοι αρχικοποίησης.

Το ψάρι, όπως αναφέρθηκε, αφού εκτελέσει ελαφριά αριστερή στροφή κινείται ευθύγραμμα. Λίγο πριν την ολοκλήρωση της κίνησης του παίρνει εντολή να εκτελέσει μια ελαφριά δεξιά στροφή. Αρχικά κινεί την ουρά του με συχνότητα 1Hz. Δηλαδή από το ένα άκρο της κίνησης της ουράς μέχρι το άλλο, μεσολαβεί ένα δευτερόλεπτο. Κάπου στο μέσο της κίνησης του, η ταχύτητα αυξάνει στα 2 Hz. Δηλαδή ο χρόνος που μεσολαβεί για να εκτελέσει η ουρά μια πλήρη κίνηση από το ένα άκρο και πάλι πίσω, μεσολαβεί 1 δευτερόλεπτο. Λογικά η αλλαγή αυτή θα πρέπει να γίνεται αντιληπτή στο διάγραμμα της ταχύτητας. Ακόμα θα εξετάσουμε την τροχιά του ρομποτικού ψαριού, για τα δυο σημεία που επιλέξαμε. Τέλος θα δούμε και την γωνία περιστροφής του ρομποτικού ψαριού ως προς το σύστημα συντεταγμένων της εικόνας.

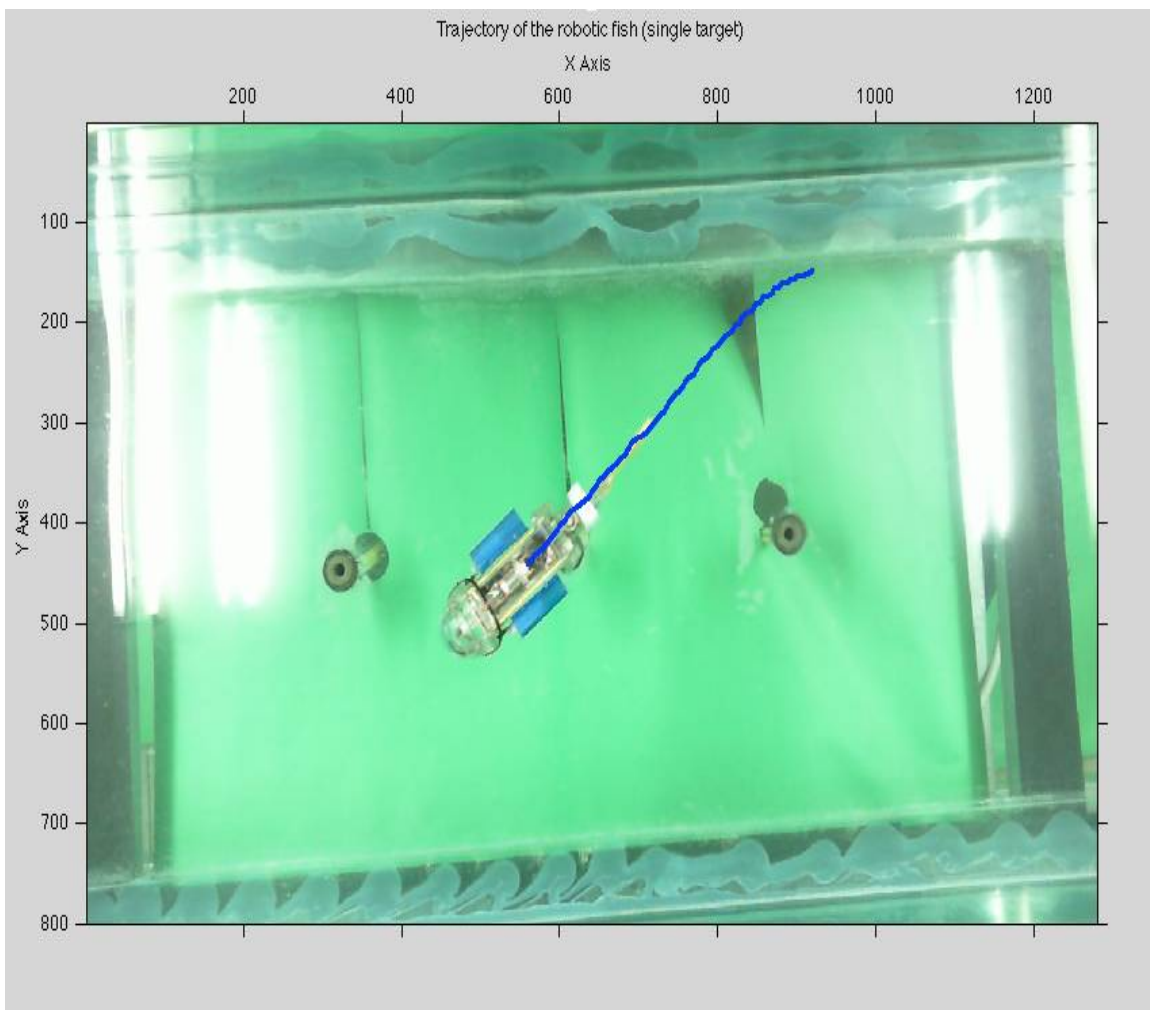
Αρχικά, θα εξετάσουμε την τροχιά που ακολούθησε το ψάρι, με τη βοήθεια του πρώτου – βασικού στόχου. Τα αποτελέσματα της κίνησης στο επίπεδο φαίνονται στο Σχήμα 5-2.



Σχήμα 5-2: Διάγραμμα τροχιάς ρομποτικού ψαριού στο επίπεδο.

Για να γίνουν πιο εύληπτα τα αποτελέσματα, θα παρουσιαστεί η τροχιά που ακολούθησε το ψάρι, πάνω στην εικόνα που προέκυψε από την κάμερα. Από την αλληλουχία των εικόνων, θα χρησιμοποιήσουμε την τελευταία, ώστε να έχουμε καλύτερη άποψη για το ιστορικό της κίνησης.

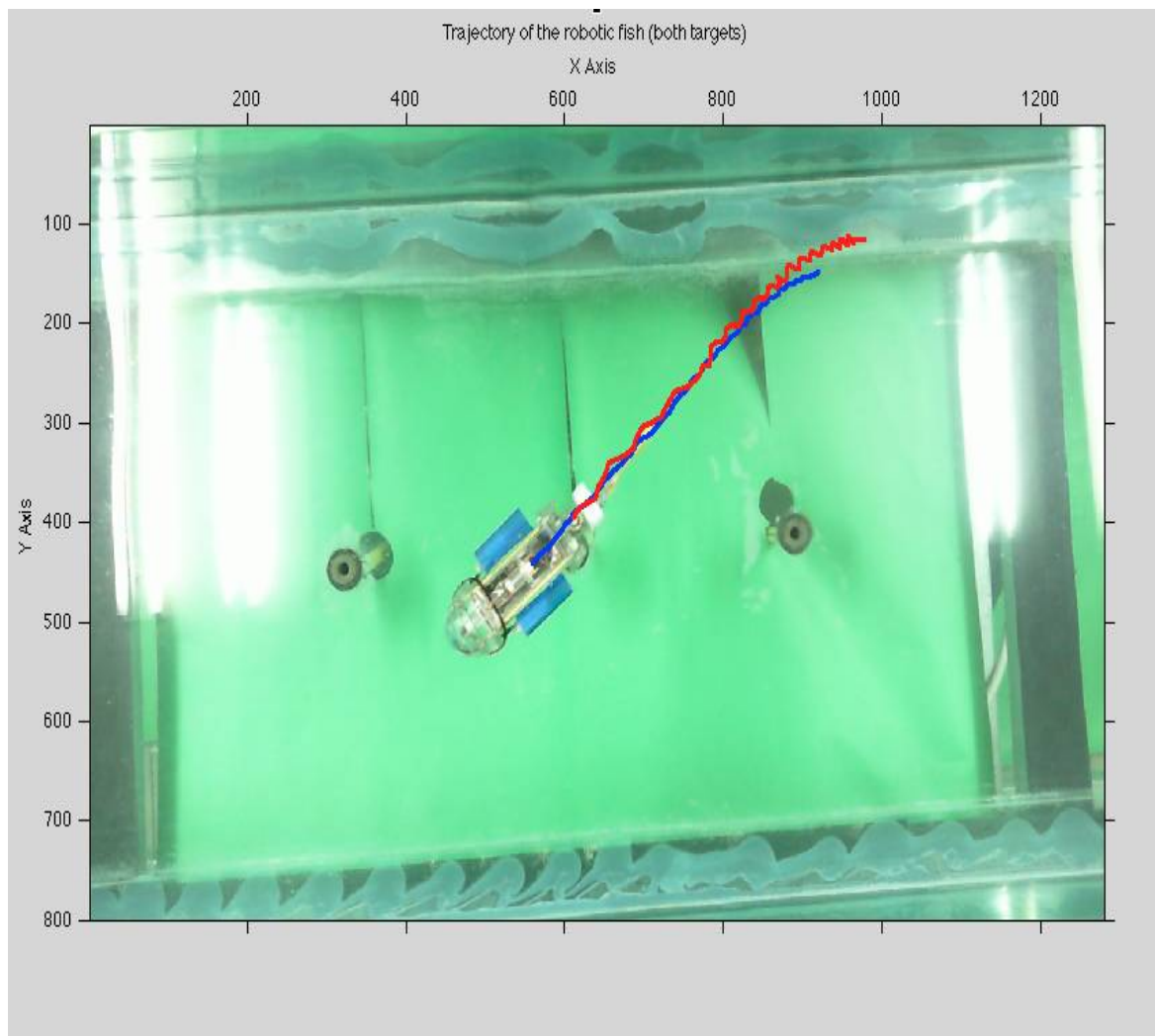
Στο Σχήμα 5-3, που ακολουθεί, φαίνεται η τελευταία εικόνα που ελήφθηκε από την κάμερα, και πάνω σε αυτή φαίνεται η τροχιά που ακολούθησε το ψάρι. Η τροχιά γίνεται ορατή με μπλε γραμμή. Ξεκινάει από το σημείο έναρξης της κίνησης του ψαριού (βλ. Εικόνα αρχικοποίησης Σχ. 5-1), και τελειώνει στο σημείο που βρίσκεται ο βασικός στόχος, στην τελευταία αποθηκευμένη εικόνα. Το σύστημα συντεταγμένων είναι αυτό που χρησιμοποιεί η MATLAB. Όπως φαίνεται πράγματι η κίνηση που έχει καταγράψει είναι μια ελαφριά αριστερή στροφή του ψαριού, και στη συνέχεια ευθύγραμμη κίνηση. Η ελαφριά δεξιά στροφή δεν εμφανίζεται στο διάγραμμα. Επομένως μπορούμε να υποθέσουμε ότι η εντολή δόθηκε λίγο πριν ή σχεδόν ταυτόχρονα με την ολοκλήρωση της καταγραφής του ψαριού από την κάμερα. Παρόλα αυτά, θα εξετάσουμε στη συνέχεια και τα υπόλοιπα δεδομένα που προέκυψαν, για να δούμε αν η τελευταία περιστροφή καταγράφηκε, με κάποιον άλλο τρόπο. Οι μονάδες του συστήματος συντεταγμένων, είναι τα pixels και όχι τα mm, για να συμπίπτουν η θέση του ψαριού, όπως προκύπτει από την κάμερα, με την τροχιά. Αν θελήσουμε να εξετάσουμε τα πραγματικά αποτελέσματα (σε mm), αρκεί να τα πολλαπλασιάσουμε επί 1.1196 (βλ. Κεφ 4-5).



Σχήμα 5-3: Τροχιά ρομποτικού ψαριού, πάνω σε πραγματική εικόνα από την κάμερα.

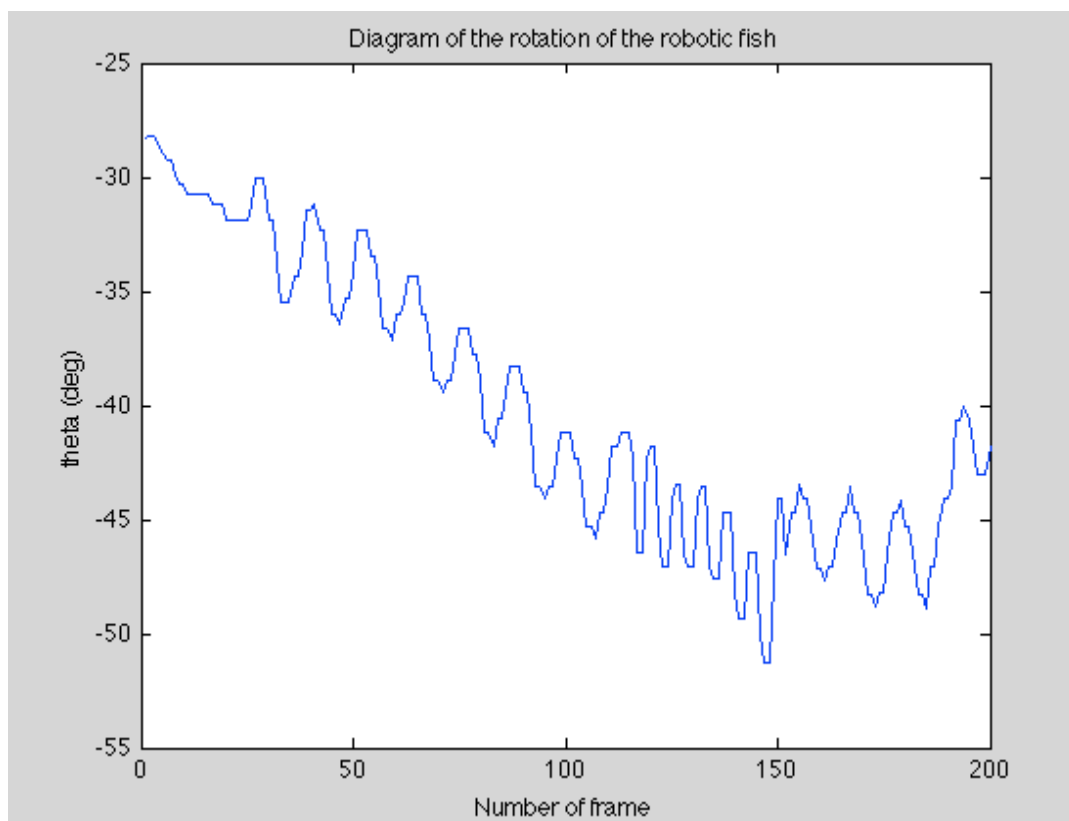
Για την πληρέστερη παρουσίαση των αποτελεσμάτων, θα δούμε τα αποτελέσματα και για τον δεύτερο – βοηθητικό στόχο. Υπενθυμίζεται ότι ο δεύτερος στόχος είναι το κυλινδρικό τεμάχιο στο πίσω μέρος του ψαριού (βλ. Εικόνα αρχικοποίησης, Σχ. 5-1). Με βάση τον προσανατολισμό που έχει το ψάρι στην εικόνα αρχικοποίησης και την κίνηση που εκτελεί, θα πρέπει να βρίσκεται διαρκώς, λίγο πιο πάνω και πιο δεξιά σε σχέση με τα αντίστοιχα σημεία του βασικού στόχου.

Στο Σχήμα 5-4 εμφανίζεται πάνω στην τελευταία εικόνα στη μνήμη, η τροχιά που ακολούθησαν και τα δυο σημεία στόχοι, κατά την διάρκεια της κίνησης. Όπως φαίνεται τα δυο σημεία εκτελούν παρεμφερείς τροχιές. Με μπλε γραμμή εμφανίζεται η τροχιά που διέγραψε ο κύριος στόχος, και με κόκκινη η τροχιά που διέγραψε ο βοηθητικός.



Σχήμα 5-4: Τροχιά και των δυο στόχων, πάνω σε πραγματική εικόνα από την κάμερα.

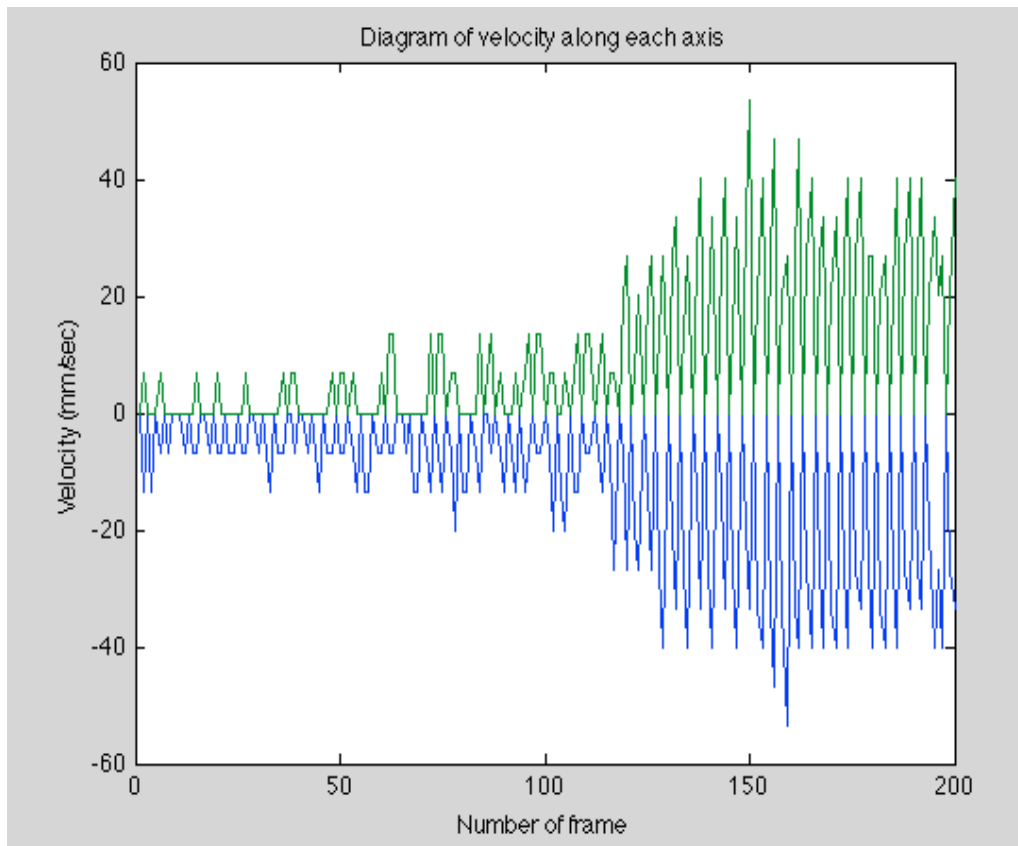
Μετά την ολοκλήρωση της μελέτης της τροχιάς του ρομποτικού ψαριού, εξετάζουμε την γωνία που σχηματίζει το σώμα ως προς το σύστημα συντεταγμένων της εικόνας. Ο τρόπος που υλοποιείται η εύρεση της γωνίας, αναφέρεται αναλυτικά στο κεφ. 4.5.1 Τα δύο σημεία που χρησιμοποιούμε είναι προφανώς ο βασικός και ο βοηθητικός στόχος που έχουμε δηλώσει. Το αποτέλεσμα της μέτρησης φαίνεται στο σχήμα 5-5 που ακολουθεί.



Σχήμα 5-5: Γωνία προσανατολισμού του ρομποτικού ψαριού κατά τη διάρκεια της κίνησης του.

Οι διακυμάνσεις που παρατηρούνται στο σχήμα οφείλονται στις αναταράξεις που δημιουργούνται κατά την κίνηση του ψαριού, λόγω της ταλάντωσης της ουράς. Με μια αναφορά στο Σχήμα 5-4, όπου διακρίνονται οι τροχιές και των δυο σημείων στόχων, γίνεται αντιληπτή η αντιστοιχία των αποτελεσμάτων στα δυο διαγράμματα, τουλάχιστον για τις 180 περίπου πρώτες εικόνες. Η γωνία φτάνει μέχρι και τις -51° . Στην συνέχεια λαμβάνει χώρα μια μεταβολή της γωνίας, η οποία επανέρχεται σε τιμές μέχρι και -40° . Έτσι προκύπτει ότι το σώμα, λίγο πριν τελειώσει η καταγραφή του, εκτελεί μια περιστροφή ωρολογιακά, όπως βλέπουμε το ψάρι από την κάμερα. Στην ουσία, αυτή η περιστροφή είναι η δεξιά στροφή, που δώσαμε εντολή να εκτελέσει. Στα διαγράμματα θέσης, δεν μπορέσαμε να διακρίνουμε αυτήν την κίνηση. Όμως στο διάγραμμα της γωνίας περιστροφής, αυτό γίνεται ορατό. Προφανώς, η εντολή δόθηκε σχεδόν λίγο πριν τη λήξη της καταγραφής. Έτσι το ρομποτικό ψάρι ξεκίνησε να περιστρέφεται (κατά 10° ωρολογιακά), αλλά αυτό δεν μπόρεσε να γίνει ορατό με όρους διαγράμματος τροχιάς. Αυτό είναι ένα ενδεικτικό στοιχείο της λεπτομερούς μελέτης που μπορεί να επιτευχθεί με τη χρήση του αλγορίθμου.

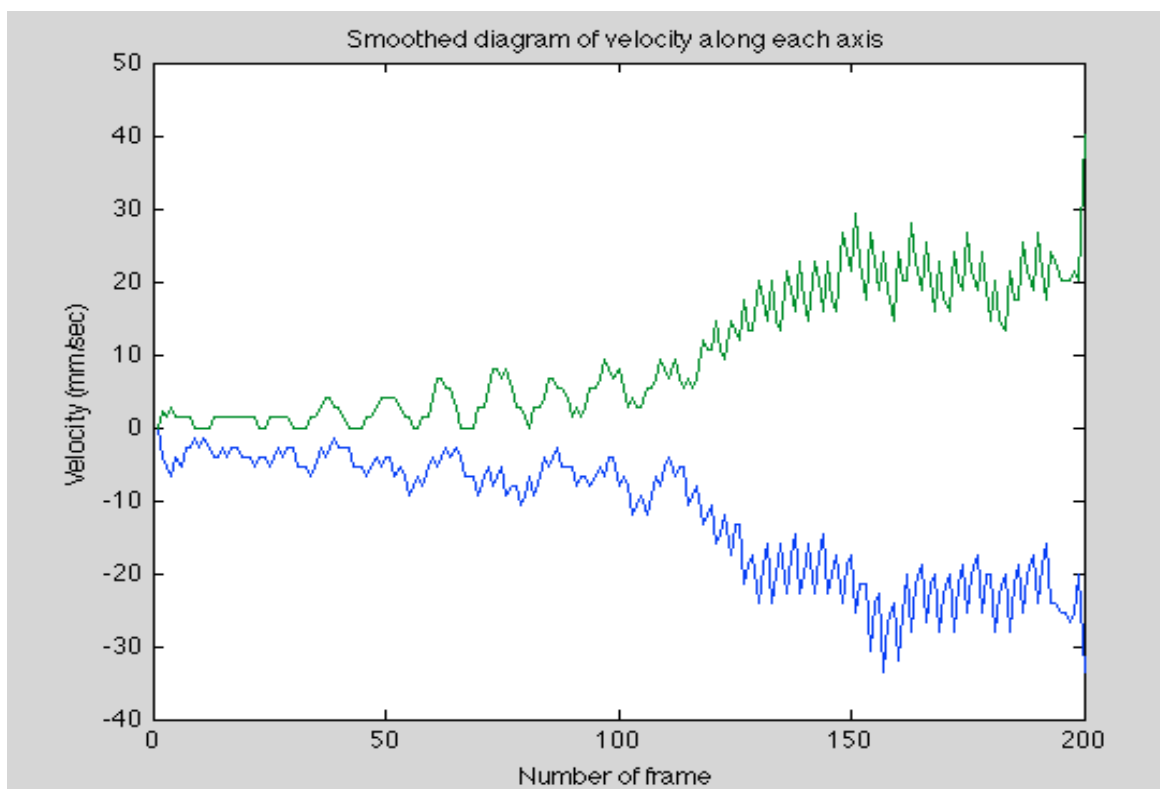
Αφού ολοκληρώθηκε και η μελέτη της γωνίας περιστροφής, μένει να εξετάσουμε την ταχύτητα της κίνησης του ρομποτικού ψαριού. Όπως αναφέρθηκε, σε κάποιο σημείο μετά το μέσο της κίνησης, η ταχύτητα της ουράς αυξήθηκε από 1 στα 2 Hz. Η αλλαγή αυτή θα πρέπει να είναι ορατή στο διάγραμμα της ταχύτητας. Στο σχήμα που ακολουθεί δίνεται η ταχύτητα του ψαριού όπως αυτή μετρήθηκε από την επεξεργασία που εκτέλεσε ο αλγόριθμος.



Σχήμα 5-6: Διάγραμμα ταχύτητας ρομποτικού ψαριού στο επίπεδο.

Επειδή η ταχύτητα έχει δυο συνιστώσες, στο διάγραμμα εμφανίζεται κάθε συνιστώσα ξεχωριστά. Με το μπλε χρώμα εμφανίζεται η συνιστώσα κατά τον x άξονα, ενώ με το πράσινο χρώμα εμφανίζεται η κατά y συνιστώσα της ταχύτητας. Οι μονάδες είναι σε mm/sec. Από το διάγραμμα μπορούμε να εξάγουμε διάφορα σημαντικά συμπεράσματα. Αρχικά, παρατηρούμε ότι η συνιστώσα κατά τον άξονα x, έχει αρνητικές τιμές, ενώ η συνιστώσα κατά y έχει θετικές τιμές. Αυτό είναι και το αναμενόμενο αν θυμηθούμε το σύστημα συντεταγμένων της MATLAB. Ακόμα, είναι εμφανής η αλλαγή στην ταχύτητα που εμφανίζεται μετά την 110η (περίπου) εικόνα. Ακόμα, στο τέλος της κίνησης παρατηρούμε μια μικρή μείωση της συνιστώσα της ταχύτητας ως προς τον y άξονα. Αυτό είναι ενδεικτικό της δεξιάς στροφής που καλείται να εκτελέσει το ψάρι, όπως είδαμε και στο αντίστοιχο διάγραμμα της γωνίας περιστροφής. Πάντα, για τη σωστή ερμηνεία των αποτελεσμάτων, θα πρέπει να έχουμε υπόψη το σύστημα συντεταγμένων της εικόνας.

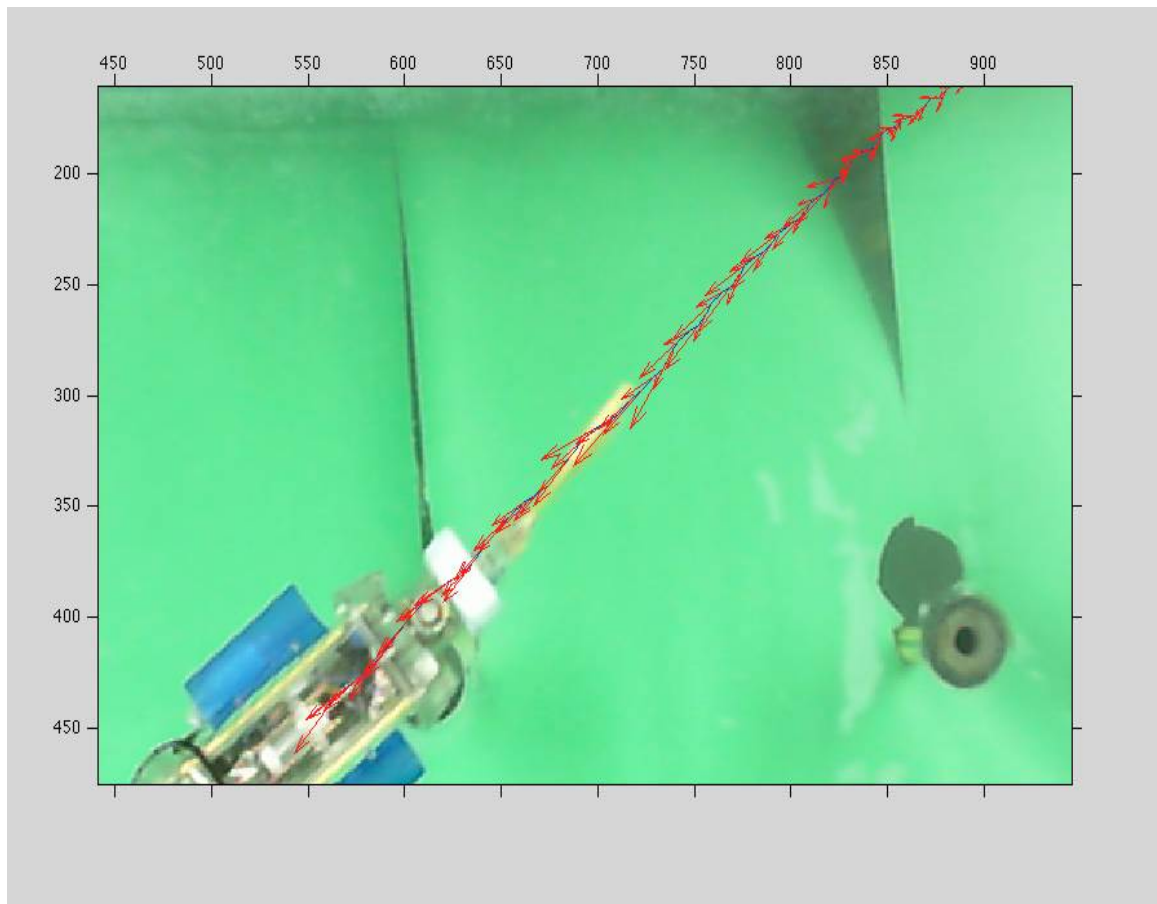
Για την καλύτερη μελέτη της ταχύτητας, θα χρησιμοποιήσουμε την εντολή **smooth** του MATLAB. Η εντολή αυτή λειαίνει την καμπύλη. Η λείανση γίνεται αντικαθιστώντας κάθε σημείο, με το μέσο όρο των 5 σημείων γύρω από αυτό. Έτσι θα έχουμε και μια καλύτερη εικόνα για τον μέσο όρο της ταχύτητας. Το διάγραμμα που προέκυψε φαίνεται στο Σχήμα 5-7.



Σχήμα 5-7: Διάγραμμα ταχύτητας ρομποτικού ψαριού στο επίπεδο, μετά την χρήση της εντολής smooth.

Όπως φαίνεται στο Σχήμα 5-7, το σώμα ξεκινάει από την ακινησία, και φτάνει περίπου τα 10 mm/s, στον y και στον x άξονα. Στη συνέχεια η ταχύτητα αυξάνει πάλι περίπου στην 110η εικόνα μέχρι που φτάνει τα 20 mm/s. Επόμενως και στην μελέτη της ταχύτητας τα αποτελέσματα είναι τα αναμενόμενα και ερμηνεύσιμα με βάση την κίνηση του ρομποτικού ψαριού. Κατά συνέπεια, ο αλγόριθμος μπορεί να παρέχει ολοκληρωμένα αποτελέσματα, για την μελέτη κίνησης του σώματος.

Για την καλύτερη εικόνα της ταχύτητας στις διάφορες φάσης της κίνησης του ρομποτικού ψαριού, δίνεται η εικόνα του Σχήματος 5-8. Η εικόνα είναι λεπτομέρεια του Σχήματος 5-3, με την επιπλέον διαφορά ότι στα διάφορα σημεία της τροχιάς, εμφανίζεται και το αντίστοιχο διάνυσμα ταχύτητας που έχει μετρηθεί στη συγκεκριμένη θέση.



Σχήμα 5-8: Διανύσματα ταχύτητας στα σημεία της τροχιάς που κινήθηκε το ρομποτικό ψάρι.

6 Συμπεράσματα και Μελλοντική Εργασία

6.1 Εισαγωγή

Στο τελευταίο κεφάλαιο της εργασίας, θα αναφέρουμε τα συμπεράσματα που προέκυψαν ύστερα από την ολοκλήρωση του κατασκευαστικού μέρους, και της μελέτης της κίνησης του ρομποτικού ψαριού. Παρουσιάζεται η αξιολόγηση της λειτουργίας του αλγορίθμου, καθώς και τα όποια προβλήματα παρατηρήθηκαν κατά την λειτουργία του.

6.2 Συμπεράσματα

Το κατασκευαστικό μέρος της εργασίας αφορούσε στο σχεδιασμό και στην κατασκευή του φορείου. Ο σκοπός της κατασκευής επιτεύχθηκε αφού πλέον υπάρχει ένα απόλυτα λειτουργικό φορείο. Είναι πλήρως προσαρμοσμένο στην προϋπάρχουσα υποδομή του εργαστηρίου, που ήταν κατά κύριο λόγο η δεξαμενή, και φυσικά σε όλες τις υποστηρικτικές σε αυτήν, κατασκευές. Κινείται με πολύ μικρή τριβή στα “χείλη” της δεξαμενής και είναι ικανό να σαρώνει το επίπεδο κίνησης του, αφού μπορεί να κινείται σε δυο διαστάσεις. Έχει την απαραίτητη θέση για την πρόσδεση του ψαριού, γεγονός που διευκολύνει την μελλοντική εκτέλεση πειραμάτων με τη βοήθεια του. Η κατασκευή ακολούθησε σε μεγάλο βαθμό το σχεδιασμό, αφού μικρές τροποποιήσεις χρειάστηκαν για την υλοποίηση των σχεδίων. Τέλος τα τεμάχια επιλέχθηκαν και με κριτήριο τις όσο το δυνατόν λιγότερες ιδιοκατασκευές που θα έπρεπε να γίνουν, κάτι που σε μεγάλο βαθμό επιτεύχθηκε. Η κίνηση γίνεται με την βοήθεια δυο κινητήρων συνεχούς ρεύματος, που επιτρέπουν τον έλεγχο της θέσης και της ταχύτητας του και στους δυο άξονες.

Στο δεύτερο μέρος της εργασίας αναπτύξαμε αλγόριθμο επεξεργασίας εικόνας, και παρουσιάστηκαν τα αποτελέσματα που προέκυψαν από αυτόν. Στο κεφάλαιο 5 δίνονται αποτελέσματα από τις μετρήσεις που έγιναν με τη χρησιμοποίηση του.

Ο αλγόριθμος είχε να εκτελέσει μια λειτουργία εξαιρετικά περίπλοκη και με υψηλό υπολογιστικό κόστος. Οι επιφυλάξεις ήταν έντονες πριν την υλοποίηση του για το αν θα είχε τα επιθυμητά αποτελέσματα. Ωστόσο, τα αποτελέσματα που προέκυψαν ήταν ιδιαίτερα

ενθαρρυντικά. Μάλιστα αποκτήσαμε διαγράμματα με τα φυσικά μεγέθη της κίνησης του ρομποτικού ψαριού, που πολύ συχνά παρουσίασαν ακρίβεια μεγαλύτερη του αναμενόμενου. Για παράδειγμα μπορεί να θυμηθεί κανείς, τα διαγράμματα τα οποία παρουσίαζαν διαταραχές, οι οποίες οφείλονταν στην ταλάντωση της ουράς. Ένα ακόμα δείγμα της αυξημένης ακρίβειας των αποτελεσμάτων, εμφανίστηκε στο διάγραμμα της γωνίας περιστροφής, και επιβεβαιώθηκε στο διάγραμμα της ταχύτητας. Στο συγκεκριμένο διάγραμμα έγινε ορατή η ελαφριά δεξιά στροφή που κλήθηκε να εκτελέσει το ψάρι. Η κίνηση αυτή ήταν τόσο ελαφριά που δεν είχε γίνει ορατή στα διαγράμματα της τροχιάς. Ακόμα, θα πρέπει να αναφερθούμε στην διαδραστικότητα του αλγορίθμου, που δίνει την δυνατότητα στον χρήστη να καθορίζει τις εικόνες – πρότυπα. Αυτό τον καθιστά κατάλληλο και για άλλες παρόμοιες εφαρμογές παρακολούθησης στόχων. Τέλος, στην περίπτωση που ο αλγόριθμος παρακολουθεί έναν μόνο στόχο τα αποτελέσματα είναι σαφώς καλύτερα και σε μεγαλύτερες ταχύτητες, αφού ο όγκος των υπολογισμών είναι μικρότερος.

Υπάρχουν φυσικά και κάποιες αδυναμίες στον αλγόριθμο. Μία από αυτές είναι ο μεγάλος αριθμός πράξεων που καλείται να εκτελέσει σε κάθε κύκλο λειτουργίας. Έτσι αδυνατεί να παρακολουθήσει τον στόχο σε μεγάλες ταχύτητες. Όπως αναφέρθηκε είναι σαφώς πιο κατάλληλος για την παρακολούθηση ενός στόχου παρά δυο. Ένα ακόμα σημαντικό πρόβλημα είναι τα αδύναμα αποτελέσματα που παρουσιάζει όταν ο στόχος εκτελεί μεγάλες περιστροφές. Έγινε μια προσπάθεια να αντιμετωπιστεί το πρόβλημα αυτό, όμως δεν αντιμετωπίστηκε πλήρως. Επιβαρυντικός παράγοντας σε αυτό το πρόβλημα είναι η αναζήτηση δύο στόχων αντί ενός. Ένας εξαιρετικά σημαντικός παράγοντας αστοχίας του αλγορίθμου, είναι ο φωτισμός στο χώρο της δεξαμενής. Σε πολλά σημεία, όπως φαίνεται και στις φωτογραφίες, το φως αντανακλάται στο νερό. Αυτό έχει σαν αποτέλεσμα όταν ο στόχος βρίσκεται στις συγκεκριμένες περιοχές, η κάμερα να “τυφλώνεται”. Δηλαδή βλέπει την έντονη αντανάκλαση, με αποτέλεσμα να μην διακρίνεται οτιδήποτε βρίσκεται κάτω από αυτήν. Γενικά ο αλγόριθμος θεωρείται ότι έχει πολύ καλή αντοχή στις αλλαγές του φωτισμού. Στη συγκεκριμένη περίπτωση όμως, οι αλλαγές είναι τόσο έντονες, που συχνά οδηγούν σε αστοχία τον αλγόριθμο.

Σε γενικές γραμμές όμως, και με βάση τα αποτελέσματα που ελήφθησαν, μπορούμε να πούμε ότι ο αλγόριθμος εκτέλεσε με επιτυχία την λειτουργία για την οποία προορίστηκε. Μάλιστα τα αποτελέσματα εμφάνισαν αξιοσημείωτη ακρίβεια, παρέχοντας όλα εκείνα τα μεγέθη που είναι απαραίτητα για τη μελέτη της κίνησης του ρομποτικού ψαριού.

6.3 Μελλοντική Εργασία

Οι δυνατότητες που είναι διαθέσιμες, μετά την ολοκλήρωση όλης της διαδικασίας που περιγράφηκε, είναι αναμφισβήτητα πολλές. Αφ'ενός, αφορούν στη χρησιμοποίηση του φορείου που κατασκευάστηκε, και αφ'ετέρου την βελτίωση της λειτουργίας του αλγορίθμου.

Το φορείο όπως αναφέρθηκε είναι πλέον πλήρως λειτουργικό, και κατά συνέπεια έτοιμο για να χρησιμοποιηθεί σε πειράματα. Έτσι προτείνεται η πραγματοποίηση των πειραμάτων που περιγράφηκαν προηγούμενα. Το πιο σημαντικό πείραμα που πρέπει να γίνει, είναι η καταγραφή των δυνάμεων που ασκούνται στο ψάρι κατά τη διάρκεια της κίνησης του. Επιπλέον το φορείο μπορεί, με πρόσδεση κάμερας και τον αλγόριθμο επεξεργασίας εικόνας που περιγράφηκε, να παρακολουθεί το ψάρι όταν αυτό κινείται ελεύθερα στο νερό. Με ενδεχόμενη χρησιμοποίηση και δεύτερης κάμερας, μπορούν να καταγράφονται και άλλα μεγέθη, όπως η συχνότητα κίνησης της ουράς. Επομένως με τη χρησιμοποίηση του φορείου, μπορούν πλέον να γίνουν κάποια πολύ σημαντικά πειράματα, που θα βοηθήσουν στην μελέτη της κίνησης του ρομποτικού ψαριού μέσα στο νερό.

Και η απόδοση του αλγορίθμου επιδέχεται πολλών βελτιώσεων. Σε πρώτη φάση θα πρέπει να αντιμετωπιστούν οι αδυναμίες της παρούσας διάταξης. Η σημαντικότερη από αυτές αναφέρθηκε και είναι ο ανομοιογενής φωτισμός. Λύση σε αυτό, μπορούν να αποτελέσουν ειδικά πλαστικά κάλυπτρα για τις λάμπες. Τα κάλυπτρα αυτά είναι σχεδιασμένα έτσι ώστε να διαχέουν το φως. Με αυτόν τον τρόπο αποφεύγονται οι έντονες αντανakλάσεις πάνω στο νερό της δεξαμενής που τυφλώνουν την κάμερα. Μια ακόμα λύση, πιο πρόχειρη, είναι η αφαίρεση των λαμπτήρων πάνω από τη δεξαμενή. Ένας τρόπος περαιτέρω βελτίωσης, είναι το χρησιμοποιούμενο υπολογιστικό σύστημα. Στη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε ένας φορητός υπολογιστής (MacBook Pro). Θα μπορούσε να βελτιώσει την λειτουργία του, η χρησιμοποίηση ενός σταθερού υπολογιστή, αυξημένων υπολογιστικών δυνατοτήτων. Επίσης για την καλύτερη υλοποίηση της εφαρμογής, και την εξασφάλιση ενός αυτόνομου συστήματος, προτείνεται έντονα, η υλοποίηση του αλγορίθμου σε ένα FPGA. Το FPGA, είναι αυτόνομο υπολογιστικό σύστημα, ιδανικό για την υλοποίηση ενσωματωμένων συστημάτων αυξημένων υπολογιστικών απαιτήσεων. Υπάρχουν εξ' άλλου FPGA, ειδικά σχεδιασμένα για εφαρμογές επεξεργασίας εικόνων και video, παρουσιάζοντας εξαιρετικά αποτελέσματα αναφορικά με την ταχύτητα επεξεργασίας. Θεωρείται ότι η χρησιμοποίησή του, θα μεγιστοποιήσει την απόδοση του αλγορίθμου, εμφανίζοντας ακόμα πιο ακριβή αποτελέσματα, που θα δίνουν μεγαλύτερες δυνατότητες επεξεργασίας. Ενθαρρύνεται ακόμα, η περαιτέρω χρησιμοποίηση των αποτελεσμάτων των μετρήσεων αυτών που έγιναν και αυτών που μπορούν πλέον να γίνουν (π.χ. Μέτρηση δυνάμεων). Με τον τρόπο αυτό μπορεί να γίνει πλήρως αυτόνομη η λειτουργία του ρομποτικού ψαριού.

Τέλος, θα ήταν εξαιρετικά χρήσιμη η μεγαλύτερη διαδραστικότητα του αλγορίθμου. Ως ένα

βαθμό επιτυγχάνεται, όμως σίγουρα υπάρχει περιθώριο βελτίωσης. Έτσι θα ήταν χρήσιμο, ο χρήστης να ξεκινάει και να σταματά την λειτουργία του αλγορίθμου, με την χρήση κουμπιών, σε μια οθόνη διεπαφής με τον υπολογιστή. Έτσι ο αλγόριθμος δεν θα σταματά την λειτουργία του, μόλις ολοκληρώσει την επεξεργασία ενός προκαθορισμένου αριθμού εικόνων, αλλά όποτε το επιθυμεί ο χρήστης.

Όλα τα παραπάνω είναι προτάσεις για μελλοντική εργασία, που προκύπτουν άμεσα από την ολοκλήρωση της διάταξης. Σίγουρα θα υπάρχουν και άλλα προβλήματα στα οποία θα μπορεί να δοθεί λύση ή/και βελτίωση, σε σχέση με τα μέχρι τώρα αποτελέσματα. Σε κάθε περίπτωση όμως, η όλη διάταξη αποτελεί ένα αξιόπιστο εργαλείο για περαιτέρω έρευνα και εργασία.

Βιβλιογραφία

- [1] Anderson, J.M., Streitlen, K., Barret, D.S., Triantafyllou, M.S., “Oscillating foils of high propulsive efficiency”, *J. Fluid Mech.*, Cambridge University Press, 1998, vol. 360, pp. 41–72.
- [2] Barrett D., S., Triantafyllou M., S., Yue, D., K., P., Grosenbaugh, M., A., Wolfgang, M., J., “Drag reduction in fish-like locomotion”, *J. Fluid Mech.*, Cambridge University Press, 1999, vol. 392, pp. 183–212.
- [3] Carreras, M., Ridao, P., Garcia, R., Nicosevici, T., “Vision-based Localization of an Underwater Robot in a Structured Enviroment”, *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*.
- [4] Diamant, R., Lampe, L., *Underwater Localization with Time-Synchronization and Propagation Speed Uncertainties*, 8th Workshop on Positioning Navigation and Communication, 2011.
- [5] Ferecatu, M., Shahbi, H., “Multi-View object matching and tracking using canonical correlation analysis”, *16th IEEE Interntional Conference on Image Processing (ICIP)*, 2009.
- [6] Gonzalez, R., Woods, R., *Digital Image Processing*, Gatesmark Publishing, 3rd Edition, 2007.
- [7] Gonzalez, R., Woods, R., Eddins, S., *Digital Image Processing Using Matlab*, 2rd Edition, Gatesmark Publishing, 2009.
- [8] Kato, N., “Control Performance in the Horizontal Plane of a Fish Robot with Mechanical Pectoral Fins”, *IEEE Journal of Oceanic Engineering*, Vol. 25, No. 1, January 2000.
- [9] Kelly, S., D., Xiong, H., “Controlled Hydrodynamic Interactions in Schooling Aquatic Locomotion”, *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005.
- [10] Kelly, S. D., Murray, R.M., “Modeling Efficient Pisciform swimming for control”,

International Journal of Robust and Nonlinear Control, 2000.

- [11] Lauder, G., Anderson, E., Tangorra, J., Madden, P., “Fish biorobotics: kinematics and hydrodynamics of self-propulsion, *The Journal of Experimental Biology*, 2007.
- [12] Licht, S., Polidoro, V., Flores, M., Hover, F. S., Triantafyllou, M.S., “Design and Projected Performance of a Flapping Foil AUV”, *IEEE Journal of Oceanic Engineering*, Vol. 29, No. 3, July 2004.
- [13] Morris, T., *Computer Vision and Image Processing*, Palgrave MacMillan, New York, c2004.
- [14] Nixon, M., and Aguado, A., *Feature Extraction and Image Processing*, Academic Press, Elsevier, 2nd Edition 2008.
- [15] Parker, J.R., *Algorithms for Image Processing and Computer Vision*, Wiley Computer Publishing, 1996.
- [16] Perez-Careta, E., Torres-Cisneros, M., Debeir, O., Sanchez-Montragon, I., Avina-Cervantes, J.G., Ibarra-Manzano, O.G., May-Arriola, D., Perez-Pentoja, E., Negrete-Romero, G., “Cell tracking by normalized cross – correlation with image processing”, *IEEE/LEOS Digest of the Summer Topical Meetings*, 2008.
- [17] Petrou, M., Petrou, C., *Image Processing : The Fundamentals*, Wiley Computer Publishing, 2nd Edition, 2010.
- [18] Sarvaiya, J.N., Patnaik, S., Bombaywala, S., “Image Registration by Template Matching Using Normalized Cross – Correlation”. *International Conference on Advances in Computing, Control & Telecommunication Technologies*, ACT'09, 2009.
- [19] Shih, F., *Image Processing and Pattern Recognition*, Wiley – IEEE Press, 2010.
- [20] Solberg, J., R., Lynch, K., M., MacIver, M., A., “Robotic Electrolocation: Active Underwater Target Localization with Electric Fields”, *IEEE International Conference on Robotics and Automation*, 2007.
- [21] Tamaki, A., “Detection the object figure in the line drawing by using cross-correlation”, *IEEE International Conference Systems, Man and Cybernetics, Conference Proceedings*, IEEE SMC'99, 1999 .
- [22] Tan, X., Carpenter, M., Thon, T., Alequin-Ramos, F., “Analytical Modeling and

- Experimental Studies of Robotic Fish Turning”, *2010 IEEE International Conference on Robotics and Automation*, 2010.
- [23] Valdivia y Alvarado, P., Youcef-Toumi, K., “Design of Machines With Compliant Bodies for Biomimetic Locomotion in Liquid Environments”, *Journal of Dynamic Systems, Measurement, and Control*, Vol. 128, No.13, March 2006.
- [24] Wu, J., Yue, H., Cui, Z., Chen, J., “Moving Object Tracking Method Based on Adaptive On-line Clustering and Prediction based Cross-Correlation”, *10th IEEE International Conference on Computer and Information Technology (CIT)*, 2010.
- [25] Xiao, Q., Kong, F., Tao, J., “Research on Point-to-point of Biomimetic Robot-fish Based on Fuzzy Control”, *International Conference on Electric Information and Control Engineering (ICEICE)*, 2011.
- [26] Yang, Z., Xu, W., Xiao, Z., Pan, X., “Passive Localization of an Autonomous Underwater Vehicle with Periodic Sonar Signaling”, *IEEE – Oceans*, 2010.
- [27] Yu, S.C., Kim, T.W., Asada, A., Weatherwax, S., Collinsand, B., Yuh, J., “Development of High-Resolution Acoustic Camera based Real-Time Object Recognition System by using Autonomous Underwater Vehicles, *IEEE – Oceans*, 2006.
- [28] Yu, S.,C., Kim, T.,W., Marani, G., Choi, S., K., “Real-Time 3D Sonar Image Recognition for Underwater Vehicles”, *Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, 2007.
- [29] Zhang, P., Milios, E., Gu, J., “Underwater Robot Localization using Artificial Visual Landmarks”, *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, 2004.
- [30] Zhou, C., Low, K., H., “Study and Implementation of Station-holding Performance on a Fish Robot in Adverse Unsteady Flow, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2010.
- [31] Dorf, R. and Bishop, R., *Σύγχρονα Συστήματα Αυτόματου Ελέγχου*, Εκδόσεις Τζιόλα, Θεσσαλονίκη 2008.
- [32] Καραγιάννης, Γ. και Τζιτζιράχου, Κ., *Εισαγωγή στα Σήματα και Συστήματα*, Εκδόσεις

- Παπασωτηρίου, Αθήνα 2003.
- [33] Κρικέλης, Ν., *Εισαγωγή στον Αυτόματο Έλεγχο – Θεωρία και Εφαρμογές*, Εκδόσεις Συμμετρία, 3η Έκδοση, Αθήνα 2000.
 - [34] Κόλλιας, Σ., *Επεξεργασία, Ανάλυση και Τεχνολογία Εικόνων-Βιντεο*, Εκδόσεις Ε.Μ.Π., Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Ε.Μ.Π., Αθήνα 2005.
 - [35] Μακρής, Α., *Πρόχειρες Σημειώσεις στις Μεταφορικές και Ανυψωτικές Μηχανές*, Εκδόσεις Ε.Μ.Π., Τμήμα Μηχανολόγων Μηχανικών Ε.Μ.Π., Αθήνα 2004.
 - [36] Παπαδανιήλ, Ε., Σφαντζικόπουλος, Μ., *Μηχανολογικό Σχέδιο*, Εκδόσεις Ε.Μ.Π., Τμήμα Μηχανολόγων Μηχανικών Ε.Μ.Π., Αθήνα 1999.
 - [37] Παπαδόπουλος, Ευ., *Ηλεκτρομηχανολογικά Συστήματα Μετατροπής Ενέργειας*, Εκδόσεις Ε.Μ.Π., Τμήμα Μηχανολόγων Μηχανικών Ε.Μ.Π., Αθήνα 2001.
 - [38] Παπαδόπουλος, Ευ., Κυριακόπουλος, Κ., *Σημειώσεις Ρομποτικής*, Εκδόσεις Ε.Μ.Π., Τμήμα Μηχανολόγων Μηχανικών Ε.Μ.Π., Αθήνα.
 - [39] Παπανάνος, Ν., *Ψηφιακή Επεξεργασία και Ανάλυση Εικόνας*, Εκδόσεις Γκιούρδας Αθήνα 2005.
 - [40] Σφαντζικόπουλος, Μ., *Σχεδιασμός Μηχανολογικών Κατασκευών I & II*, Εκδόσεις Ε.Μ.Π., Τμήμα Μηχανολόγων Μηχανικών Ε.Μ.Π., Αθήνα 2003.
 - [41] Φούντας, Γρ., *Αντοχή Υλικών – Τόμος I*, Εκδόσεις Φούντας – Fountas Engineering Books, Αθήνα 2002.
 - [42] <http://www.mathworks.com/help/toolbox/images/f0-3373.html>, “Matlab Documentation → Image Processing Toolbox”, Matlab 2010-b.
 - [43] <http://www.mathworks.com/help/toolbox/imaq/f9-51679.html>, “Matlab Documentation → Image Acquisition Toolbox”, Matlab 2010-b.
 - [44] <http://www.mathworks.com/help/toolbox/vision/gs/bsrf2z9-1.html>, “Matlab Documentation → Computer Vision Toolbox”, Matlab 2011-b.
 - [45] <http://www.mathworks.com/help/toolbox/vision/ug/f11920.html>, “Matlab Documentation → Computer Vision Toolbox → User's guide → Motion Analysis and Tracking → Track an Object Using Correlation”, Matlab 2011-b.

- [46] <http://www.mathworks.com/help/toolbox/vision/gs/bq45z82.html>, “Matlab Documentation → Computer Vision Toolbox → Getting Started → Strategies for Real-Time Video Processing in Simulink”, Matlab 2011-b.
- [47] <http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ipexnormxcorr2.html> , “Matlab Documentation → Image Processing Toolbox → Example and Webinars → Registering an Image Using Normalized Cross-Correlation”, Matlab 2011-b.
- [48] <http://www.misumi-europe.com/start.aspx?lang=EN&ref=> .
- [49] http://www.misumi-europe.com/index.aspx?lang=EN&fn=sub1_3.aspx&mu=05&css=L2&ref=, pdf catalogues, Standard, purchased and custom made components for assembly automation.
- [50] <http://precisionparts.wmberg.com>

Παράρτημα Α

Κώδικας αλγόριθμου επεξεργασίας εικόνας.

% Create a video object. Use the default format.

```
vid = videoinput('macvideo',1);
```

% Initialize the target. Use the mouse to declare to target. Double-click when

% finished.

```
T = getsnapshot(vid);
```

```
T = ycbcr2rgb(T);
```

```
[sub_Tinit, rect_T] = imcrop(T);
```

```
imshow(sub_Tinit)
```

```
xlength = rect_T(3);
```

```
ylength = rect_T(4);
```

```
rect_T = double(rect_T);
```

```
close(gcf)
```

```
sub_T = sub_Tinit;
```

```
[sub_Tinit2, rect_T2] = imcrop(T);
```

```
imshow(sub_Tinit2)
```

```
xlength2 = rect_T2(3);
```

```
ylength2 = rect_T2(4);
```

```
rect_T2 = double(rect_T2);
```

```
close(gcf)
```

```
sub_T2 = sub_Tinit2;
```

```
[roi, rect_roi] = imcrop(T);
```

```
imshow(roi)
```

```
roi_xstart = rect_roi(1);
```

```
roi_xend = rect_roi(1) + rect_roi(3);
```

```
roi_ystart = rect_roi(2);
```

```
roi_yend = rect_roi(2) + rect_roi(4);
```

```
close(gcf)
```

```
roi_width = rect_roi(3);
roi_height = rect_roi(4);
```

```
% Calculate the center of the image.
```

```
[r, c, d] = size(T);
gpos = [c/2, r/2];
i = 1;
movex = zeros(201,1);
movey = zeros(201,1);
```

```
xreal = zeros(200,1);
yreal = zeros(200,1);
```

```
xreal2 = zeros(200,1);
yreal2 = zeros(200,1);
```

```
timetable = zeros(200,1);
```

```
% Set the video input object properties for the application.
```

```
% Here we use both SET method and dot notation method.
```

```
set(vid,'TriggerRepeat', Inf);
set(vid, 'FramesPerTrigger',300);
vid.FrameGrabInterval = 5;
```

```
% Set value of a video source object property.
```

```
vid_src = getselectedsource(vid);
```

```
c_im1 = normxcorr2(sub_T(:, :, 1), T(roi_ystart:roi_yend, roi_xstart:roi_xend, 1)); % Correlation
in the frequency domain
```

```
[max_c_im1, imax_im1] = max(c_im1(:));
[ypeak_im1, xpeak_im1] = ind2sub(size(c_im1), imax_im1);
```

```
c_im2 = normxcorr2(sub_T2(:, :, 1), T(roi_ystart:roi_yend, roi_xstart:roi_xend, 1)); % Correlation
in the frequency domain
```

```

[max_c_im2, imax_im2] = max(c_im2(:));
[ypeak_im2, xpeak_im2] = ind2sub(size(c_im2),imax_im2);

start(vid);

% Here comes the calculations!
while(vid.FramesAcquired<=200) % Stop after 200 frames
    [data time] = getdata(vid,1); % Grab last frame in the buffer
    data = ycbcr2rgb(data);

    c = normxcorr2(sub_T(:,1),data(roi_ystart:roi_yend,roi_xstart:roi_xend,1)); % Correlation
    in the frequency domain
    c2 = normxcorr2(sub_T2(:,1),data(roi_ystart:roi_yend,roi_xstart:roi_xend,1));

    [max_c, imax] = max(c(:));
    [ypeak, xpeak] = ind2sub(size(c),imax);

    corr_offset = [(xpeak-size(sub_T,2)), (ypeak-size(sub_T,1))];

    xgreen = round(xpeak+roi_xstart-xlength);
    ygreen = round(ypeak+roi_ystart-ylength);

    % Calculate the current position of the target.
    cpos = [xgreen+(xlength/2), ygreen+(ylength/2)];

    timetable(i) = time;
    % Calculate the desired move in both axes.
    movex(i) = gpos(1) - cpos(1);
    movey(i) = gpos(2) - cpos(2);
    move = [movex, movey];

    xreal(i) = (cpos(1))*1.1196;
    yreal(i) = (cpos(2))*1.1196;
    real = [xreal, yreal];

    [max_c2, imax2] = max(c2(:));
    [ypeak2, xpeak2] = ind2sub(size(c2),imax2);

```

```

% sub_T = data((ypeak-ylength:ypeak),(xpeak-xlength:xpeak),:);
corr_offset2 = [(xpeak2-size(sub_T2,2)), (ypeak2-size(sub_T2,1))];

xgreen2 = round(xpeak2+roi_xstart-xlength2);
ygreen2 = round(ypeak2+roi_ystart-ylength2);

% Calculate the current position of the target.
cpos2 = [xgreen2+(xlength2/2), ygreen2+(ylength2/2)];
% Calculate the desired move in both axes.

xreal2(i) = (cpos2(1))*1.1196;
yreal2(i) = (cpos2(2))*1.1196;
real2 = [xreal2, yreal2];

imshow(data,'InitialMagnification','fit')
%imagesc(data)
%drawnow
hold on
rectangle('Position', [xgreen,ygreen,xlength,ylength], ...
    'EdgeColor','g');
rectangle('Position', [xgreen2,ygreen2,xlength2,ylength2], ...
    'EdgeColor','y');
rectangle('Position', [roi_xstart,roi_ystart,roi_width,roi_height], ...
    'EdgeColor','r')
plot(cpos(1),cpo(2),'b*')
plot(cpos2(1),cpo2(2),'b+')
hold off

i = i+1;
if max_c<0.87
    sub_T = sub_Tinit;
else
    sub_T = data(ygreen:ygreen+ylength,xgreen:xgreen+xlength,1);
end

```

```

if max_c2<0.87
    sub_T2 = sub_Tinit2;
else
    sub_T2 = data(ygreen2:ygreen2+ylength2,xgreen2:xgreen2+xlength2,1);
end

xpeak_im = xpeak;
ypeak_im = ypeak;
xpeak_im2 = xpeak2;
ypeak_im2 = ypeak2;

roi_xstart = xpeak_im + roi_xstart + xlength/2 - roi_width/2;
roi_ystart = ypeak_im + roi_ystart - ylength/2 - roi_height/2;
roi_xend = roi_xstart + roi_width;
roi_yend = roi_ystart + roi_height;

if max_c < 0.82 || max_c2 < 0.82
    roi_xstart = rect_roi(1);
    roi_ystart = rect_roi(2);
    roi_xend = roi_xstart + roi_width;
    roi_yend = roi_ystart + roi_height;
end
end

stop(vid)
% Always Clean Up in the end:(
% delete(vid)
% clear vid
% clear
close(gcf)

tantheta = (yreal2-yreal)./(xreal2-xreal);
theta = atan(tantheta);
theta = theta*180/pi;
plot(theta)

dt = zeros(200,1);

```

```

for i = 2:200
    dt(i) = timetable(i) - timetable(i-1);
end
dt = mean(dt);

v_x = zeros(200,1);
v_y = zeros(200,1);

for k = 2:200
    v_x(k) = (xreal(k) - xreal(k-1)) / dt;
    v_y(k) = (yreal(k) - yreal(k-1)) / dt;
end
v = [v_x,v_y];

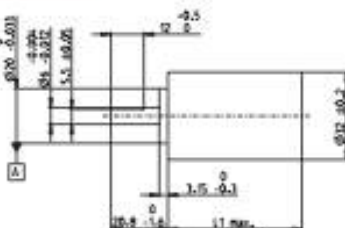
```

Παράρτημα Β

Κινητήρας στον άξονα x:

Planetary Gearhead GP 32 BZ Ø32 mm, 0.75 - 4.5 Nm

Low backlash



Technical Data

Planetary Gearhead	straight teeth
Output shaft	stainless steel, hardened
Bearing at output	ball bearing
Radial play, 5 mm from flange	max. 0.1 mm
Axial play	max. 0.7 mm
Max. permissible axial load	120 N
Max. permissible force for press fit	120 N
Sense of rotation, drive to output	=
Recommended input speed	< 4000 min ⁻¹
Recommended temperature range	-40 ... +100°C
Number of stages	1 2 3
Max. radial load, 12 mm from flange	70 N 140 N 210 N

M 1:2

- Stock program
- Standard program
- Special program (on request)

Order Number

354875	354942	355031	357048	358335	358385	358512	358513	358515	358616
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Gearhead Data	354875	354942	355031	357048	358335	358385	358512	358513	358515	358616
1 Reduction	3.7 : 1	5.2 : 1	19 : 1	27 : 1	35 : 1	71 : 1	100 : 1	130 : 1	181 : 1	236 : 1
2 Reduction absolute	3.7 : 1	5.2 : 1	19 : 1	27 : 1	35 : 1	71 : 1	100 : 1	130 : 1	181 : 1	236 : 1
3 Max. motor shaft diameter	mm	5.5	3	3	3	3	3	3	3	3
4 Number of stages	1	1	2	2	2	3	3	3	3	3
5 Max. continuous torque	Nm	0.75	0.75	2.25	2.25	2.25	4.5	4.5	4.5	4.5
Max. continuous torque with the pinion	Nm	0.5	0.5	1.1	1.1	1.1	1.7	1.7	1.7	1.7
6 Intermittently permissible torque at gear output	Nm	1.1	1.1	3.2	3.2	3.2	6.2	6.2	6.2	6.2
7 Max. efficiency	%	90	90	80	80	80	70	70	70	70
8 Weight	g	160	160	190	190	190	240	240	240	240
9 Average backlash no load	"	0.15	0.15	0.35	0.35	0.35	0.5	0.5	0.5	0.5
10 Mass inertia	gmm ²	1.25	1.25	0.75	0.75	0.75	0.7	0.7	0.7	0.7
11 Gearhead length L1*	mm	33.5	33.5	43.6	43.6	43.6	53.1	53.1	53.1	53.1

* For EC 32 L1 8 + 84 mm, for RE 2012 8 + 1.2 mm



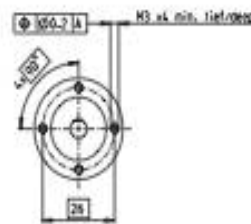
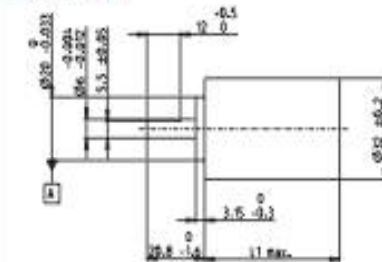
maxon Modular System			
Motor	Page	Sensor / Brake	Page
RE 25	77/79	MR	202
RE 25	77/79	Enc 22	276
RE 25	77/79	HED_5540	318
RE 25	77/79	DCT 22	318
RE 25, 20 W	78	MR	202
RE 25, 20 W	78	HED_5540	267/270
RE 25, 20 W	78	DCT22	276
RE 25, 20 W	78	AB 28	318
RE 25, 20 W	78	HED_5540 / AB 28	267/318
RE 25, 20 W	79	AB 28	318
RE 25, 20 W	79	HED_5540 / AB 28	
RE 30, 60 W	80	MR	
RE 30, 60 W	80		
RE 30, 90 W	81	MR	
RE 30, 90 W	81	HED_5540	
RE 30, 90 W	81	DCT 22	
RE 30, 90 W	81	AB 28	
RE 30, 90 W	81	HEDS 5540 / AB 28	
A-max 26	101-108		
A-max 26	101-107 ME-c 13		
A-max 26	102-108 MR		
A-max 26	102-108 Enc 22		
A-max 26	102-108 HED_5540		
A-max 32	110/111		
A-max 32	110/112		
A-max 32	110/112 MR		
A-max 32	110/112 HED_5540		
EC 32, 80 W	153		
EC 32, 80 W	153	HED_5540	267/269
EC 32, 80 W	153	Res 26	277
MCD EPOS, 60 W	313		
MCD EPOS P 60 W 313			

Κινητήρας στον άξονα γ:

Planetary Gearhead GP 32 BZ Ø32 mm, 0.75 - 4.5 Nm

Low backlash

maxon gear



Technical Data

Planetary Gearhead	straight teeth
Output shaft	stainless steel, hardened
Bearing at output	ball bearing
Radial play, 5 mm from flange	max. 0.1 mm
Axial play	max. 0.7 mm
Max. permissible axial load	120 N
Max. permissible force for press fit	120 N
Sense of rotation, drive to output	=
Recommended input speed	< 4000 min ⁻¹
Recommended temperature range	-40 ... +100°C
Number of stages	1 2 3
Max. radial load, 12 mm from flange	70 N 140 N 210 N

M 1:2

- Stock program
- Standard program
- Special program (on request)

Order Number

356975	351942	356331	357908	356335	356305	356512	356513	356515	356516
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Gearhead Data										
1 Reduction	3.7 : 1	5.2 : 1	19 : 1	27 : 1	35 : 1	71 : 1	100 : 1	139 : 1	181 : 1	236 : 1
2 Reduction absolute	1/100	5/100	18/100	27/100	35/100	71/100	100/100	139/100	181/100	236/100
3 Max. motor shaft diameter	mm	5.5	3	3	3	3	3	3	3	3
4 Number of stages		1	2	2	2	3	3	3	3	3
5 Max. continuous torque	Nm	0.75	0.75	2.25	2.25	2.25	4.5	4.5	4.5	4.5
6 Max. continuous torque with no pulsing	Nm	0.5	0.5	1.1	1.1	1.1	1.7	1.7	1.7	1.7
7 Intermittently permissible torque at gear output	Nm	1.1	1.1	3.2	3.2	3.2	6.2	6.2	6.2	6.2
8 Max. efficiency	%	90	90	80	80	80	70	70	70	70
9 Weight	g	150	150	190	190	190	240	240	240	240
10 Average backlash no load	"	0.15	0.15	0.35	0.35	0.35	0.5	0.5	0.5	0.5
11 Mass inertia	gcm ²	1.25	1.25	0.75	0.75	0.75	0.7	0.7	0.7	0.7
12 Gearhead length L1*	mm	33.5	33.5	43.6	43.6	43.6	53.1	53.1	53.1	53.1

* M EC 30 L1 8 - 64 195 70 PE 25 L1 8 - 100 19

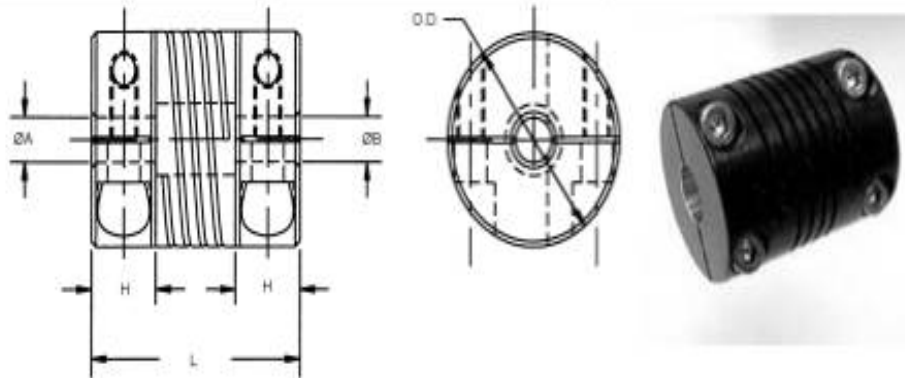


maxon Modular System										
+ Motor	Page	+ Sensor / Brake	Page	Overall length [mm] = Motor length + gearhead length + (sensor / brake) + assembly parts						
RE 25	77/79			88.1	88.1	98.2	98.2	107.7	107.7	107.7
RE 25	77/79	MR		99.1	99.1	109.2	109.2	118.7	118.7	118.7
RE 25	77/79	Enc 22		102.2	102.2	112.3	112.3	121.8	121.8	121.8
RE 25	77/79	HED_5540		108.9	108.9	119.0	119.0	128.5	128.5	128.5
RE 25	77/79	DCT 22		110.4	110.4	120.5	120.5	130.0	130.0	130.0
RE 25, 20 W	78			76.6	76.6	86.7	86.7	96.2	96.2	96.2
RE 25, 20 W	78	MR	262	87.6	87.6	97.7	97.7	107.2	107.2	107.2
RE 25, 20 W	78	HED_5540	267/270	97.4	97.4	107.5	107.5	117.0	117.0	117.0
RE 25, 20 W	78	DCT22	276	98.9	98.9	109.0	109.0	118.5	118.5	118.5
RE 25, 20 W	78	AB 26	318	110.7	110.7	120.8	120.8	130.3	130.3	130.3
RE 25, 20 W	78	HED_5540 / AB 26	267/318	127.9	127.9	138.0	138.0	147.5	147.5	147.5
RE 25, 20 W	78	AB 26	318	122.2	122.2	132.3	132.3	141.8	141.8	141.8
RE 25, 20 W	79	HED_5540 / AB 26		130.3	130.3	140.4	140.4	150.0	150.0	150.0
RE 30, 60 W	80			102.6	102.6	112.7	112.7	122.2	122.2	122.2
RE 30, 60 W	80	MR		114.0	114.0	124.1	124.1	133.6	133.6	133.6
RE 35, 90 W	81			104.6	104.6	114.7	114.7	124.2	124.2	124.2
RE 35, 90 W	81	MR		116.0	116.0	126.1	126.1	135.6	135.6	135.6
RE 35, 90 W	81	HED_5540		125.3	125.3	135.4	135.4	144.9	144.9	144.9
RE 35, 90 W	81	DCT 22		122.7	122.7	132.8	132.8	142.3	142.3	142.3
RE 35, 90 W	81	AB 26		140.7	140.7	150.8	150.8	160.3	160.3	160.3
RE 35, 90 W	81	HEDS 5540 / AB 26		157.8	157.8	167.9	167.9	177.4	177.4	177.4
A-max 25	101-108			78.3	78.3	88.4	88.4	97.9	97.9	97.9
A-max 25	101-107	MEnc 13		85.4	85.4	95.5	95.5	105.0	105.0	105.0
A-max 25	102-108	MR		87.1	87.1	97.2	97.2	106.7	106.7	106.7
A-max 25	102-108	Enc 22		92.7	92.7	102.8	102.8	112.3	112.3	112.3
A-max 25	102-108	HED_5540		97.1	97.1	107.2	107.2	116.7	116.7	116.7
A-max 32	109-111			95.5	95.5	105.6	105.6	115.1	115.1	115.1
A-max 32	110-112			95.1	95.1	105.2	105.2	114.7	114.7	114.7
A-max 32	110-112	MR		106.3	106.3	116.4	116.4	125.9	125.9	125.9
A-max 32	110-112	HED_5540		115.8	115.8	125.9	125.9	135.5	135.5	135.5
EC 32, 80 W	153			100.2	100.2	110.3	110.3	119.8	119.8	119.8
EC 32, 80 W	153	HED_5540	267/269	118.6	118.6	128.7	128.7	138.2	138.2	138.2
EC 32, 80 W	153	Res 26	277	120.3	120.3	130.4	130.4	139.9	139.9	139.9
MOD EPOS, 80 W	313			153.6	153.6	163.7	163.7	173.2	173.2	173.2
MOD EPOS P, 80 W	313			153.6	153.6	163.7	163.7	173.2	173.2	173.2

Κόπλες κινητήρων - αξόνων:

SINGLE BEAM FLEXIBLE COUPLING

BORE	STYLE	MATERIAL
3MM TO 25MM	CLAMP	ALUMINUM ANODIZED



STOCK NO.	ØA +0.05	ØB +0.05	ØO.D.	L	H	MAX. WORKING TORQUE (N·m)	ANGULAR OFFSET	PARALLEL OFFSET
COS72A-1M COS72A-2M COS72A-3M COS72A-4M	3.00 4.00 4.00 5.00	3.00 4.00 4.00 5.00	12.7	19.0	5.3	45	5°	0.25
COS74A-1M COS74A-2M COS74A-3M COS74A-4M COS74A-5M	3.00 3.00 4.00 4.00 5.00	3.00 3.00 4.00 4.00 5.00	16.0	20.3	6.1	68	5°	0.25
COS76A-1M COS76A-2M COS76A-3M COS76A-4M COS76A-5M	4.00 4.00 5.00 5.00 6.00	4.00 4.00 5.00 5.00 6.00	19.1	22.9	7.1	110	5°	0.25
COS78A-1M COS78A-2M COS78A-3M COS78A-4M COS78A-5M	6.00 6.00 8.00 8.00 10.00	6.00 6.00 8.00 8.00 10.00	25.5	31.8	8.4	170	5°	0.25
COS80A-1M COS80A-2M COS80A-3M	10.00 10.00 12.00	10.00 12.00 12.00	31.8	44.5	11.2	360	5°	0.25
COS82A-1M COS82A-2M COS82A-3M	12.00 16.00 19.00	12.00 16.00 19.00	38.0	67.0	16.0	508	5°	0.25
COS84A-1M COS84A-2M COS84A-3M	12.00 16.00 19.00	12.00 16.00 19.00	44.5	76.2	20.0	678	5°	0.25
COS86A-1M COS86A-2M COS86A-3M	19.00 22.00 25.00	19.00 22.00 25.00	57.0	130.3	32.0	1130	5°	0.25

Central relief diameter may be smaller than bore in some cases.

G

Παράρτημα Γ

Κατάλογος Υλικών εταιρίας Misumi.

	Name	Order Number	Price Per Unit	Quantity	Total Price
1	Aluminum Frames 30/60 Square	HFS6-3030-947	7,86	2	15,72
2	Aluminum Frames 30/60 Square	HFS6-3030-400	3,32	2	6,64
3	Post-Assembly Easy Brackets	HABL6	3,7	4	14,8
4	Frame Caps	HFCB6-3030-B	2,3	4	9,2
5	Shaft Supports – L Shaped (Machined Products)	SHKLB12-25	16,7	8	126,92
6	Set Collars – Set Screw – Standard Type	SCCN12-6	1,17	8	9,36
7	Rollers – Bearing Type	RORUA40-50	40,7	4	162,8
8	Precision Linear Shafts – Straight Type	SSFJ12-80	4,47	4	16,99
9	Strut Clamps	ALTTS10	23,9	4	90,82
10	Shaft Supports – Bracket Shaped – Tapped Mounting Hole type	STHST10M	14,96	4	59,84
11	Set Collars – Set Screw – Standard Compact type	SCCJ10-5	1,5	8	12
12	Rollers – Urethane Thickness Selectable Type	ROERUA40-10-20-T5	27,1	4	108,4
13	Precision Linear Shafts – Straight Type	SSFJ10-55	3,6	4	13,68
14	Precision Linear Shafts – Straight Type	SSFJ10-52	3,6	4	13,68
15	Strut Clamps	KLSTN10-8	14,7	2	27,93
16	Shaft Supports – Bottom Mount (Machined Products)	SHSC8-30	10,6	2	21,2
17	Precision Linear Shafts – Straight	SSFJ8-60	3,6	2	6,84

	Type				
18	Precision Linear Shafts – Straight Type	SSFJ12-887	31,9	2	60,61
19	Shaft Supports – Bracket Type	STHS12	13,95	4	55,8
20	Linear Bushing Housing Unit with Dowel Holes	LHBBN12	17,5	4	70
21	Precision Linear Shafts – Straight Type	SSFJ8-500	11,38	1	11,38
22	Set Collars – Set Screw – Standard Type	PSCCN8-6	2,34	2	4,68
23	Shaft Supports – Bottom Mount (Machined Products)	SHSB8-30	20,2	2	40,4
24	L-Shaped Aluminum Angles	LRA6-A65-B36-L30	8,5	4	34
25	C-Shaped Section Steels	CASA4-8040-30	13,5	2	27
26	Bearings with Housings - Retaining Ring Type	BGCA698DD	16,8	3	50,4
27	Rotary Shaft – Straight Type	PSFR8-58	5,2	1	5,2
28	Rotary Shaft – Straight Type	PSFR8-43	4,6	1	4,6
29	Aluminum Frames 30/60 Square	HFS6-3030-1000	8,3	2	16,6
30	Aluminum Frames 30/60 Square	HFS6-3030-120	3,7	8	29,6
31	Frame Caps	HFC6-3030-B	1,15	15	17,25
32	Bearings with Housings - Retaining Ring Type	BGCA6800D	17,5	3	52,5
33	L-Shaped Aluminum Angles	LRA10-A75-B30-L60	19,1	1	19,1
34	Rotary Shaft – Straight Type	PSFR10-50	4,6	1	4,6
35	Rotary Shaft – Straight Type	PSFR10-40	4,6	1	4,6
36	Brackets with Tabs	HBLFSN6	0,68	10	6,8
37	Pre-Assembly Fitting Nuts for Alu. Frames	HNTT6-6	0,34	20	6,8
38	Set Collars – Side Mount – Slit Type	SCJS12	5,8	4	23,2

39	Precision Linear Shafts- One End Stepped Type	SFAC40-25-F10-P12	10	4	38
40	Pre-Assembly Fitting Nuts for Alu. Frames	HNTT6-4	0,34	10	3,4
				Total=	1303,34

Κατάλογος Υλικών εταιρίας W.M.Berg.

Berg's Name	Order Number	Price per Unit	Quantity
<u>Plastic Chain</u>			
Gear Chain Drive	24GCF-380-E	26,95	9
Gear Chain Drive	24GCF-95-E	9,79	7
<u>Sprockets</u>			
Sprocket Gears	24B110-96	48,06	2
Sprocket Gears	24B108-60	27,49	2
<u>Couplings</u>			
Six Beam Flexible Couplng	CO41A-3M	?	1
Single Beam Flexible Couplng	COS78A-2M	?	1

Τεχνικά Χαρακτηριστικά Φίλτρου Υπεριώδους Ακτινοβολίας.

ΛΑΜΠΑ UV ΤΥΠΟΣ 600E	
Ικανότητα	600 lt/h
Πίεση (max)	8 bar
Διατομή	1/2"
Θερμοκρασία (max)	40 °C
Λειτουργία	220 - 240 V
Διάρκεια	9000 h