



Πρόγραμμα Μεταπτυχιακών Σπουδών Στη Ρομποτική
Σχολή Μηχανικών Πληροφορικής
Διεθνές Πανεπιστήμιο Της Ελλάδος

Διπλωματική Εργασία

Τίτλος:

Μελέτη και υλοποίηση αλγόριθμου για τον εντοπισμό
ρομποτικού συστήματος με βάση την επέκταση του
φίλτρου Kalman (EKF)

Φοιτητής: Ασημίδης Βασίλειος Α.Μ. 6

Επιβλέπων καθηγητής: Ιωάννης Καλόμοιρος

Σέρρες 2019

Περιεχόμενα

Πρόλογος.....	4
Abstract	5
Περίληψη της εργασίας.....	6
1. Το πρόβλημα που καλούμαστε να λύσουμε, οι στόχοι της εργασίας και τα οφέλη που παίρνουμε από αυτή.	7
1.1. Εισαγωγή στο πρόβλημα του εντοπισμού.	7
1.2. Το πρόβλημα που καλούμαστε να λύσουμε και οι στόχοι της εργασίας.	8
2. Κατάσταση της Τέχνης (State of the Art).....	9
2.1. Πλοήγηση ρομπότ με βάση τον εντοπισμό, σε σύγκριση με προγραμματιστικές λύσεις. 14	
2.2. Κατηγοριοποίηση μηχανισμών εντοπισμού.	16
2.3. Βελτίωση απόδοσης ανιχνευτή απόστασης laser στον εντοπισμό μέσω laser.	18
2.4. Χρήση κάμερας στη διαδικασία εντοπισμού.....	19
2.5. Εντοπισμός μέσω WiFi και σύγκριση διαφόρων μεθόδων εντοπισμού.	22
3. Kalman Filter.....	23
3.1. Τι είναι το φίλτρο Kalman.....	23
3.2. Υλοποίηση φίλτρου Kalman	23
3.3. Ένα παράδειγμα γραμμικού φίλτρου Kalman.	28
3.4. Η επέκταση του φίλτρου Kalman, Extended Kalman Filter.	32
4. Το πρόβλημα και η μεθοδολογία που επιλέχθηκε για τη λύση του προβλήματος.	35
4.1. Διατύπωση του προβλήματος.....	35
4.2. Μεθοδολογία επίλυσης.....	35
5. Κύριο μέρος εργασίας.	40
5.1. Κατασκευή των τεσσάρων τοίχων (world map).....	41
5.2. Προσομοίωση laser beam.....	43
5.2.1. Εισαγωγή από τον χρήστη της πραγματικής θέσης του ρομπότ και εξαγωγή του rotation matrix.....	43
5.2.2. Κατασκευή του laser beam.....	44
5.2.3. Προσομοίωση μετρήσεων του Laser scanner.....	44
5.3. Εκτίμηση των r και a που καθορίζουν τους τοίχους σε πολικές συντεταγμένες, από τις μετρήσεις του laser scanner.	45
5.4. Κατασκευή ιστογράμματος γωνιών (και αποστάσεων) των τοίχων.....	47
5.5. Κατασκευή των Wall Clusters και υπολογισμός measurement Covariance και μέσων τιμών a,r των τοίχων.	49
5.6. Προσομοίωση οδομετρίας.	52
5.7. Εξαγωγή predicted τιμών.	53

5.8. Εφαρμογή Extended Kalman Filter.....	54
6. Αξιολόγηση των αποτελεσμάτων.....	60
7. Σύνοψη συμπερασμάτων και μελλοντική επέκταση της εργασίας.....	62
8. Βιβλιογραφία.....	64
9. Παραρτήματα	66
9.1. Παράρτημα 1: Επεξήγηση πινάκων συμμεταβλητότητας.....	66
9.2. Παράρτημα 2. Στοιχεία προγραμματισμού LabView 2011.....	68

Πρόλογος

Σε αυτή την εργασία έγινε μια πρώτη προσέγγιση στο πρόβλημα του εντοπισμού ενός αυτόνομου ρομπότ. Αρχικά ασχοληθήκαμε με το γενικότερο πρόβλημα του εντοπισμού και της χαρτογράφησης, γνωστό ως SLAM, όμως τελικά απλουστεύσαμε το πρόβλημα και ασχοληθήκαμε μόνο με τον εντοπισμό, καθώς τα χρονικά πλαίσια της ολοκλήρωσης της εργασίας ήταν περιορισμένα.

Αυτή δεν ήταν η μόνη απλούστευση που κάναμε στο αρχικό μας πρόβλημα. Ακόμη και το πρόβλημα του εντοπισμού είναι αρκετά πολύπλοκο στην γενική του μορφή. Έτσι περιοριστήκαμε σε εντοπισμό σε ένα δωμάτιο με το ρομπότ να κάνει ένα μόνο βήμα ξεκινώντας από γνωστή θέση και με γνωστό χάρτη.

Αν δει κανείς τη βιβλιογραφία θα διαπιστώσει ότι το πρόβλημα του εντοπισμού μελετάται από τα μέσα της δεκαετίας του 80 και έχει διερευνηθεί σε πολύ μεγάλο βαθμό. Το αποτέλεσμα που επιδιώκουμε με αυτή την εργασία είναι να δοθεί ένα απτό παράδειγμα, μια εφαρμογή ορισμένων τεχνικών εντοπισμού που εμφανίζονται στη βιβλιογραφία. Ιδιαίτερος μας ενδιέφερε να εφαρμόσουμε το φίλτρο Kalman και να δούμε πόσο καλά διορθώνει την αρχική εκτίμηση της θέσης που έχουμε από την οδομετρία. Αυτό το παράδειγμα χρησιμοποιεί έναν αισθητήρα (laser sensor) που μπορεί να θεωρηθεί κλασικός, μιας και στη βιβλιογραφία χρησιμοποιείται εδώ και πολλά χρόνια και δίνει αξιοπρεπείς λύσεις στο πρόβλημα.

Διαβάζοντας αυτό το παράδειγμα ελπίζουμε ότι νέοι σπουδαστές θα μπορέσουν να εισαχθούν γρήγορα στο πρόβλημα του εντοπισμού ενός αυτόνομου ρομπότ και να αποκτήσουν μια καλή εικόνα για τις παραμέτρους που υπεισέρχονται σε αυτό το πολύπλοκο πρόβλημα. Επίσης, μπορεί να αποτελέσει τη βάση για την ανάπτυξη πιο πολύπλοκων αλγόριθμων για την επίλυση του προβλήματος του εντοπισμού.

Θέλω να ευχαριστήσω τον επιβλέποντα Καθηγητή αυτής της εργασίας, τον κ. Ιωάννη Καλόμοιρο, ο οποίος με βοήθησε υποδεικνύοντας τον τρόπο προσέγγισης της εκτεταμένης βιβλιογραφίας που υπάρχει σχετικά με το πρόβλημα του εντοπισμού. Επίσης, με βοήθησε και σε κάθε επιμέρους βήμα αυτής της εργασίας. Χωρίς τη βοήθειά του αυτή η εργασία δεν θα μπορούσε να υλοποιηθεί.

Η εφαρμογή σχεδιάστηκε εξολοκλήρου με τη βοήθεια του λογισμικού LabView 2011, που επιτρέπει εύκολη οπτικοποίηση των αποτελεσμάτων.

Abstract

In this essay the stochastic problem of tracking a robot (self tracking) that moves in a plane is investigated.

A calculating model that has been created in LabView is used for the investigation. This model creates the following

1. A map of the indoor space with landmarks (the four walls)
2. A simulation of a laser scanner which takes measurements from the walls and is supposed to be fitted (physically) on the robot. In these measurements we include a simulated Gaussian error.
3. From the measurements we extract landmarks (r distance and a angle of each wall from the robot pose), witch we compare with the available map of the world.

Finally we apply the Kalman filter (specifically the Extended Kalman Filter) to reduce the starting uncertainty witch is inserted in the model from basic odometry calculations.

This project has mainly an educational purpose in that it creates a virtual environment that simulates the problem of robot self detection in an indoor environment. The project can also stand as a base for further research as different error models and mathematical stochastic filters can be applied and tampered with. Also different maps can be used on the model that is offered in this project and the model can be used as a base for many extensions of the basic self detection problem witch it solves.

The basic conclusion derived from this study is that the virtual environment created here is suitable for educational purposes as it offers a tool for quick supervision of a simplified version of the more general and difficult problem of robot self detection in indoor environments.

Περίληψη της εργασίας

Σε αυτή την εργασία διερευνούμε το στοχαστικό πρόβλημα του εντοπισμού ενός ρομπότ που κινείται στο επίπεδο.

Χρησιμοποιούμε ένα υπολογιστικό μοντέλο που το αναπτύσσουμε σε περιβάλλον LabView. Το μοντέλο αυτό δημιουργεί

- 1 έναν χάρτη εσωτερικού χώρου χρησιμοποιώντας landmarks (ορόσημα).
- 2 Ένα μοντέλο σαρωτή laser που λαμβάνει μετρήσεις από τους εικονικούς τοίχους. Στις μετρήσεις αυτές θεωρούμε ότι υπάρχει αβεβαιότητα η οποία εισάγεται με τη βοήθεια συναρτήσεων που έχουν κατανομή Gauss.
- 3 Από τις μετρήσεις εξάγουμε ορόσημα (π.χ δηλαδή απόσταση και γωνία των τοίχων από το pose του ρομπότ) και τα συγκρίνουμε με τα ορόσημα του χάρτη στο σύστημα αναφοράς του ρομπότ.

Τέλος εφαρμόζουμε το φίλτρο Kalman (ποιο συγκεκριμένα το Extended Kalman Filter) προκειμένου να μειώσουμε την αρχική αβεβαιότητα που εισάγει η οδομετρία και οι αβέβαιες μετρήσεις του σαρωτή laser.

Η εργασία αυτή έχει κυρίως εκπαιδευτικό στόχο και δημιουργεί ένα εικονικό περιβάλλον προσομοίωσης και επεξεργασίας του προβλήματος του εντοπισμού. Επίσης μπορεί να αποτελέσει βάση για έρευνα, καθώς στο μοντέλο αυτό μπορούν να γίνουν αναλύσεις με διαφορετικά ποσοστά σφάλματος, η διαφορετικά φίλτρα αβεβαιότητας ή και με διαφορετικά ορόσημα.

Ως συμπέρασμα αυτής της μελέτης εξάγεται ότι το εικονικό περιβάλλον που δημιουργήσαμε είναι κατάλληλο για τη διδασκαλία και απόδοση εποπτείας στο δύσκολο στοχαστικό πρόβλημα του εντοπισμού του ρομπότ.

1. Το πρόβλημα που καλούμαστε να λύσουμε και οι στόχοι της εργασίας.

1.1.Εισαγωγή στο πρόβλημα του εντοπισμού.

Η εργασία αυτή ασχολείται με το πρόβλημα του εντοπισμού ενός ρομπότ όπως αναφέρεται στο χώρο της ρομποτικής. Ο εντοπισμός αποτελεί συχνά μέρος του γενικότερου προβλήματος που ονομάζεται SLAM (Simultaneous Localization and Mapping). Έτσι πριν ξεκινήσει η εισαγωγή στο πρόβλημα του εντοπισμού είναι σωστό να αναφερθούν μερικά πράγματα για το γενικότερο πρόβλημα του εντοπισμού και χαρτογράφησης (SLAM). Σε αυτή την κατηγορία προβλημάτων αυτό που προσπαθούν να κάνουν οι ερευνητές είναι να κατασκευάσουν και να προγραμματίσουν ένα αυτόνομο ρομπότ ώστε αυτό να χαρτογραφεί το περιβάλλον του και ταυτόχρονα να εντοπίζει επακριβώς τον εαυτό του στον χάρτη που κατασκεύασε, καθώς στη γενικότερη περίπτωση αυτό κινείται σε ένα άγνωστο περιβάλλον.

Το γενικότερο πρόβλημα που ακούει στο όνομα SLAM εισήχθηκε για πρώτη φορά το 1986 από τους Smith και Cheeseman και έγινε δημοφιλές τη δεκαετία του 1990 κατά την οποία δημοσιεύθηκαν πολλά άρθρα που ασχολούταν με το συγκεκριμένο πρόβλημα. Στο πέρασμα των χρόνων εμφανίστηκαν καινούριες μέθοδοι που χρησιμοποιούσαν διαφορετικούς αισθητήρες για την επίλυση του προβλήματος δημιουργώντας καινούριους τρόπους απόδοσης των δεδομένων και καινούριους τύπους χαρτών.

Το πρόβλημα του εντοπισμού και της χαρτογράφησης (SLAM) θεωρείται ως ένα από τα κλειδιά για την επίτευξη αυτόνομων ρομπότ, και ως τέτοιο είναι ουσιαστικό για τα αυτοκινούμενα οχήματα που είναι επίσης ένα πρόβλημα που απασχολεί την βιομηχανία στις μέρες μας.

Αυτή η εργασία ασχολείται με ένα υποπρόβλημα του SLAM που είναι ο εντοπισμός ενός αυτόνομου ρομπότ. Το πρόβλημα του εντοπισμού αναφέρεται στη γνώση που μπορεί να έχει ένα ρομπότ για τη θέση του στο παγκόσμιο σύστημα αναφοράς. Είναι ένα πρόβλημα που θέτει το ρομπότ στον εαυτό του καθώς αναρωτιέται “που βρίσκομαι” ώστε να αποφασίσει τι θα κάνει στη συνέχεια.

Ο εντοπισμός μπορεί να ορισθεί ως η διαδικασία της εκτίμησης της πόζας (θέσης και προσανατολισμού) του ρομπότ σχετικά με ένα σημείο αναφοράς, που βασίζεται σε πληροφορίες από τους αισθητήρες του ρομπότ. Η απόδοση του συστήματος εντοπισμού που διαθέτει ένα ρομπότ αξιολογείται με βάση την ακρίβεια με την οποία καθορίζεται η πόζα του ρομπότ γνωρίζοντας την πραγματική πόζα του. Ο χάρτης του χώρου θεωρείται επίσης γνωστός.

Εδώ πρέπει να διαχωρίσουμε το πρόβλημα του εντοπισμού σε δύο κατηγορίες. Εντοπισμός σε εξωτερικό περιβάλλον, που ενδιαφέρει όταν έχουμε αυτοκινούμενα οχήματα και ρομπότ που λειτουργούν σε εξωτερικούς χώρους, και εντοπισμός σε εσωτερικούς χώρους, που ενδιαφέρει όταν έχουμε ρομπότ που κινούνται μόνο σε εσωτερικούς χώρους. Ανάλογα με το ποια κατηγορία εντοπισμού ενδιαφέρει χρησιμοποιούνται αισθητήρες με διαφορετικές προδιαγραφές. Σε εξωτερικό περιβάλλον για παράδειγμα χρησιμοποιείται GPS κάτι που δεν μπορεί να γίνει σε εσωτερικά περιβάλλοντα. Η απαιτούμενη ακρίβεια στην απόδοση του

συστήματος εντοπισμού επίσης διαφέρει ανάλογα με το αν δουλεύουμε σε εξωτερικό ή εσωτερικό περιβάλλον. Σε αυτή την εργασία μας ενδιαφέρει να προσομοιώσουμε ένα ρομπότ που κινείται και εντοπίζει τον εαυτό του σε εσωτερικό περιβάλλον.

1.2. Το πρόβλημα που καλούμαστε να λύσουμε και οι στόχοι της εργασίας.

Για την επίλυση του προβλήματος του εντοπισμού δημιουργούμε μια υπολογιστική εφαρμογή από την αρχή, η οποία κάνει τα παρακάτω:

1. Κάνει μια θεώρηση για την αβεβαιότητα της θέσης με βάση τα στοιχεία από την οδομετρία

Στη γενικότερη περίπτωση η αρχική θέση του ρομπότ δεν είναι γνωστή, αυτό το πρόβλημα ονομάζεται kidnapped robot problem και δεν προσπαθούμε να το λύσουμε σε αυτή την εργασία. Μας ενδιαφέρει εδώ να λύσουμε το πρόβλημα του εντοπισμού του ρομπότ όταν γνωρίζουμε την αρχική του θέση, από την οποία το ρομπότ κάνει ένα βήμα, στοιχεία για το οποίο μας δίνει η οδομετρία του.

2. Προσομοιώνει τον χάρτη του εσωτερικού χώρου στον οποίο θα κινηθεί το ρομπότ.

Στη παρούσα εργασία θεωρούμε έναν απλό χάρτη που αποτελείται από τέσσερις τοίχους και προσομοιώνει ένα δωμάτιο. Τα δεδομένα που καθορίζουν τους τοίχους του δωματίου είναι η απόσταση και η γωνία κάθε τοίχου από μια αρχή συντεταγμένων.

3. Προσομοιώνει τον laser scanner και λαμβάνει μετρήσεις μέσω της προσομοίωσης. Επίσης προσομοιώνει σφάλματα στις μετρήσεις του laser.

Πολλοί και διαφορετικοί αισθητήρες χρησιμοποιούνται για να βοηθήσουν το ρομπότ να εντοπίσει τον εαυτό του στη βιβλιογραφία. Οι κυριότεροι είναι : α) αισθητήρας laser β) κάμερα, γ) άλλοι τύποι αισθητήρων όπως μέτρηση έντασης σήματος wifi. Σε αυτή την εργασία ασχολούμαστε μόνο με αισθητήρα laser.

4. Εξάγει ευθείες των τοίχων από τις μετρήσεις (r, θ (απόσταση και γωνία στο σύστημα συντεταγμένων του ρομπότ)) του laser scanner και χρησιμοποιώντας ιστόγραμμα εξάγει από αυτές τις συντεταγμένες των οροσήμων (landmarks).
5. Συγκρίνει τα ορόσημα του γνωστού σε αυτό χάρτη με τα ορόσημα που εξήγαγε από τις μετρήσεις και ανανεώνει το διάλυμα της θέσης (pose) του ρομπότ με βάση το στοχαστικό φίλτρο EKF.

Εδώ πρέπει να αναφερθεί ότι υπάρχουν διάφορες παραλλαγές του στοχαστικού φίλτρου Kalman, εδώ επιλέχθηκε να χρησιμοποιηθεί η πιο κοινή παραλλαγή που είναι το EKF.

Στόχος μας σε αυτή την εργασία είναι το ρομπότ να κάνει ένα βήμα, και χρησιμοποιώντας τις πληροφορίες από την οδομετρία και τον laser scanner να βελτιώσει την εκτίμηση της πόζας του μέσω του φίλτρου EKF. Επίσης θέλουμε με αυτή την εργασία το πρόβλημα του εντοπισμού ενός ρομπότ να γίνει προσιτό στους σπουδαστές που ενδιαφέρονται, εισάγοντάς τους στον χώρο της ρομποτικής που ασχολείται με τέτοια προβλήματα.

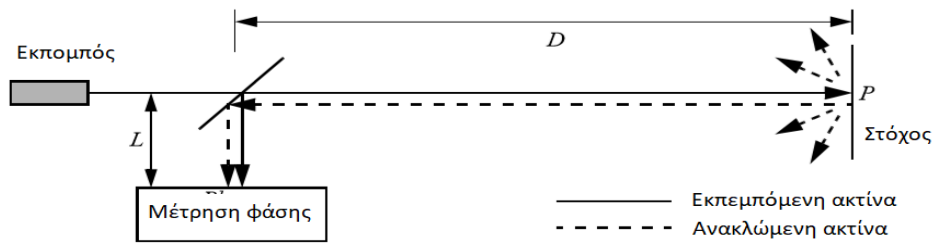
2. Κατάσταση της Τέχνης (State of the Art).

Το ρομπότ γενικά παίρνει πληροφορίες για το περιβάλλον του από αισθητήρες όπως laser και κάμερες, ενώ επιπλέον πληροφορία για τη θέση του μπορεί να του δώσει η οδομετρία καθώς αυτό κινείται στο χώρο. Στα αυτοκινούμενα ρομπότ χρησιμοποιείται μια μεγάλη ποικιλία από αισθητήρες. Μερικοί αισθητήρες χρησιμοποιούνται για τη μέτρηση απλών τιμών όπως η εσωτερική θερμοκρασία των ηλεκτρονικών του ρομπότ ή η ταχύτητα περιστροφής των κινητήρων του. Άλλοι ποιο εξεζητημένοι χρησιμοποιούνται για να πάρουμε πληροφορίες για το περιβάλλον του ρομπότ ή για την άμεση μέτρηση της παγκόσμιας θέσης του. Εδώ θα ασχοληθούμε λίγο με τους αισθητήρες που μας δίνουν πληροφορίες για το περιβάλλον του ρομπότ. Επειδή ένα αυτοκινούμενο ρομπότ κινείται συνεχώς, αντιμετωπίζει απρόβλεπτα περιβαλλοντικά χαρακτηριστικά, και έτσι οι αισθητήρες που μας δίνουν πληροφορίες για το περιβάλλον του είναι κρίσιμης σημασίας.

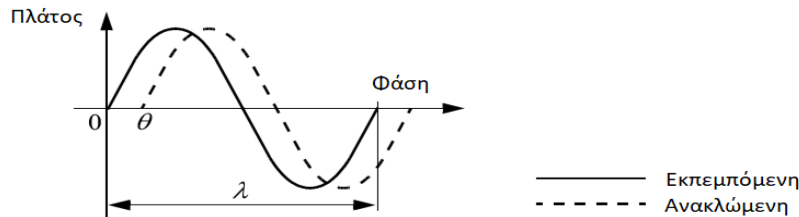
Ένας από τους σκοπούς αυτής της εργασίας είναι η προσομοίωση της λειτουργίας ενός laser sensor. Το ρομπότ το οποίο ξεκινά από μια γνωστή θέση και κάνει ένα βήμα, στα όρια του βήματος χάνεται η βεβαιότητα της θέσης και του προσανατολισμού του ρομπότ. Αυτή την αβεβαιότητα θέλουμε να ελαττώσουμε. Έπειτα παίρνει μια μέτρηση του γύρω του χώρου σκανάροντας σε 360° με ένα laser και τέλος χρησιμοποιώντας το φίλτρο Kalman διορθώνει την αρχική εκτίμηση της θέσης του που προέρχεται από την οδομετρία συγκρίνοντας τις μετρήσεις που πήρε με το laser με τα στοιχεία που έχει από τον χάρτη του χώρου. Στο πρόβλημά μας ο χάρτης του χώρου που κινείται το ρομπότ είναι γνωστός. Έτσι είναι σημαντικό να μελετήσουμε λίγο τους αντίστοιχους αισθητήρες που υπάρχουν στο εμπόριο σήμερα και θα μπορούσαν να χρησιμοποιηθούν σε μια πραγματική εφαρμογή.

Θα ασχοληθούμε με τον αισθητήρα τύπου laser που μετρά απόσταση (range finder). Αυτός είναι ένας αισθητήρας που εκπέμπει ηλεκτρομαγνητική ακτινοβολία (μονοχρωματική). Αναφέρεται ως αισθητήρας τύπου time of flight και είναι καλύτερος από τους αντίστοιχους αισθητήρες υπερήχων γιατί χρησιμοποιεί φως αντί για υπέρηχους. Αποτελείται από ένα εκπομπό που φωτίζει το στόχο με μονοχρωματική ακτινοβολία και ένα δέκτη που ανιχνεύει το στοιχείο του φωτός που ουσιαστικά είναι μια δέσμη με ίδιο άξονα με την εκπεμπόμενη. Συχνά αναφέρεται ως οπτικό ραντάρ ή LIDAR (light detection and ranging), αυτές οι συσκευές παράγουν μια εκτίμηση απόστασης βασιζόμενες στο χρόνο που χρειάζεται το φως για να φτάσει στο στόχο και να επιστρέψει. Σε ορισμένους τύπους LIDAR υπάρχει μια μηχανική κατασκευή που σαρώνει τον απαιτούμενο χώρο σε ένα επίπεδο ή και στις τρεις διαστάσεις.

Ένας τρόπος για να μετρήσει κανείς το time of flight σε μια δέσμη φωτός είναι να χρησιμοποιήσει ένα παλμικό laser και να μετρήσει τον χρόνο που κάνει η δέσμη να συναντήσει το εμπόδιο και να ανακλαστεί πίσω στον δέκτη απευθείας (αυτό σημαίνει εξάλλου time of flight). Για να γίνει αυτό χρειάζονται ηλεκτρονικά που μπορούν να μετρούν χρόνο της τάξης των picoseconds και έτσι αυτού του τύπου οι συσκευές είναι ακριβές. Ένας άλλος τρόπος είναι να μετρήσει τη συχνότητα παλμού μεταξύ ενός συνεχούς κύματος διαμορφωμένης συχνότητας και της εισερχόμενης αντανάκλασής του. Ένας τρίτος ποιο εύκολος τρόπος είναι να μετρήσει την μετατόπιση συχνότητας του ανακλώμενου φωτός. Ο τρόπος που γίνεται αυτό παρουσιάζεται στο παρακάτω σχήμα.



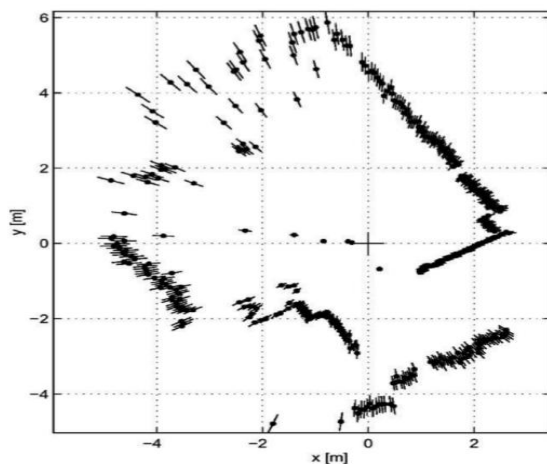
Αποστασιομέτρηση με laser εκμεταλλευόμενη το φαινόμενο της ολίσθησης φάσης



Εκτίμηση απόστασης μετρώντας την μετατόπιση φάσης μεταξύ της εκπεμπόμενης και της ανακλώμενης δέσμης

2-1 Μέτρηση απόστασης μέσω ολίσθησης φάσης με Laser scanner. [3]

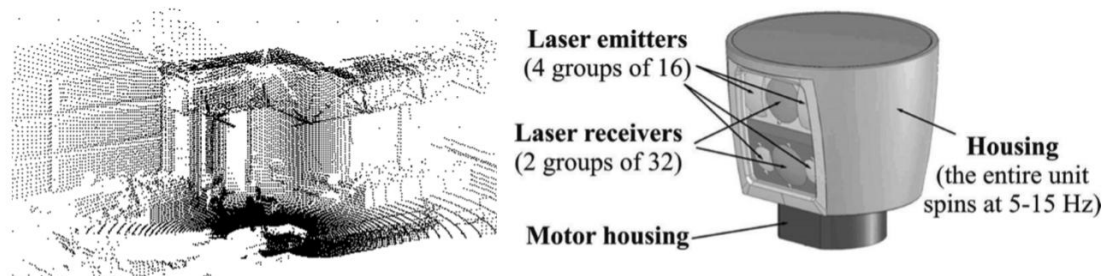
Η γωνιακή ακρίβεια των laser που μετράνε απόσταση είναι πολύ μεγαλύτερη από τους αισθητήρες υπερήχων. Ακρίβεια των 0.25 μοιρών είναι αναμενόμενη. Η ακρίβεια στην μέτρηση της απόστασης έχει τυπική τιμή $\pm 0.035\text{m}$ για αποστάσεις από 0.05m ως 20m και ανάλογα με τη συσκευή ως και 80m. Η ακρίβεια στην μέτρηση της απόστασης εξαρτάται από την ανακλαστικότητα της επιφάνειας στην οποία προσπίπτει το laser. Ανοιχτόχρωμοι τοίχοι σε μικρή απόσταση θα μετρηθούν με μεγάλη ακρίβεια ενώ ποιο σκούροι τοίχοι σε μεγάλη απόσταση θα μειώσουν την ακρίβεια της μέτρησης. Στην παρακάτω εικόνα φαίνεται η πληροφορία που μας δίνει ένας laser scanner που έχει σαρώσει ένα εσωτερικό χώρο (360°). Το πάχος στους τοίχους είναι το σφάλμα της μέτρησης του laser scanner.



2-2 Οι τοίχοι ενός χώρου όπως φαίνονται από τον laser scanner. Το πάχος των τοίχων δηλώνει το σφάλμα του ανιχνευτή laser. [3]

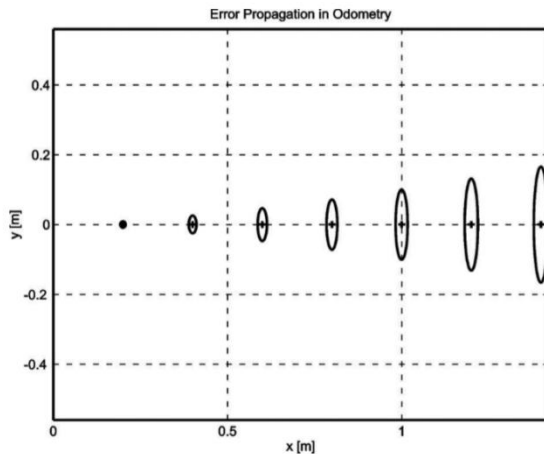
Εκτός των δύο διαστάσεων υπάρχουν και laser που σκανάρουν στις τρεις διαστάσεις. Μια τέτοια συσκευή, παίρνει πληροφορίες σε περισσότερα από ένα επίπεδο. Custom made τέτοιες συσκευές κατασκευάζονται περιστρέφοντας έναν 2D scanner με ένα βηματικό κινητήρα γύρο από άξονα παράλληλο με το επίπεδο που σκανάρει ο 2D scanner. Το μειονέκτημα είναι ότι ο

χρόνος για μια πλήρη 3D σάρωση μπορεί να φτάσει μερικά δευτερόλεπτα, κάτι που μας αναγκάζει να κινούμαστε με μικρές ταχύτητες. Μια σάρωση στις τρεις διαστάσεις φαίνεται στο παρακάτω σχήμα, μαζί με τον 3D laser scanner. Τέλος πρέπει να αναφέρουμε ότι υπάρχουν και άλλες μέθοδοι για τη μέτρηση της απόστασης όπως time of flight κάμερες, αλλά δεν θα αναφερθούν περισσότερα εδώ.



2-3 Τρισδιάστατος laser scanner

Για να πάρουμε μια ιδέα για το πόσο δύσκολο είναι το πρόβλημα του εντοπισμού ενός ρομπότ πρέπει να ασχοληθούμε λίγο με το θόρυβο που εισάγεται στις μετρήσεις μας αλλά και με το φαινόμενο της ψευδωνυμίας σε διάφορους αισθητήρες. Συχνά η πηγή των προβλημάτων θορύβου είναι ότι κάποια χαρακτηριστικά του χώρου δεν γίνονται αντιληπτά από το ρομπότ. Δείτε και σχήμα 2-2 για την περίπτωση θορύβου από ένα laser. Αν χρησιμοποιούσαμε αισθητήρες υπερήχων για να μετρήσουμε το δωμάτιο που φαίνεται στο σχήμα 2-2 θα είχαμε ακόμη μεγαλύτερο θόρυβο λόγω της φύσης αυτών των αισθητήρων. Σε αυτή την εργασία για παράδειγμα, που έχουμε ένα δωμάτιο με τέσσερις τοίχους, υπάρχουν περιπτώσεις όπου το laser scanner δεν μετρά τέσσερις τοίχους αλλά μόνο τρεις. Άλλες φορές, όπως μας έδειξε αυτή η εργασία, εμφανίζονται χαρακτηριστικά που δεν υπάρχουν πραγματικά. Αυτό είναι ακόμη πιο καταστροφικό αναφορικά με την ελπίδα μας να εντοπίσουμε καλύτερα το ρομπότ. Στο δωμάτιο που μελετάμε σε αυτή την εργασία έχουμε τέσσερις τοίχους, και παρόλο που χρησιμοποιούμε laser scanner (που έχει μικρά σχετικά σφάλματα στη μέτρηση) υπάρχουν περιπτώσεις όπου εμφανίζεται ένας τοίχος επιπλέον (τοίχος φάντασμα), πράγμα που δυσχεραίνει πολύ τα συμπεράσματά μας και δεν βοηθά στην τελική εκτίμηση της θέσης του ρομπότ. Εξάλλου για να σαρώσει το laser 360° πρέπει να περιστρέφεται με κάποιο κινητήρα. Έτσι η ακρίβεια στην περιστροφή του laser από τον κινητήρα εισάγει επιπλέον σφάλματα στις τελικές τιμές που θα πάρουμε. Αντίστοιχα σφάλματα εισάγονται στην οδομετρία από τους κινητήρες που κινούν το ρομπότ αλλά και από την πρόσφυση στο τερέν και από άλλους παράγοντες. Γενικώς οι κινητήρες του ρομπότ εισάγουν αβεβαιότητα για την μελλοντική του κατάσταση, όπως και άλλοι εξωτερικοί παράγοντες που δεν μπορούν να ληφθούν υπόψη (πχ γλιστερό πάτωμα). Έτσι η απλή πράξη της κίνησης του ρομπότ μεγαλώνει την αβεβαιότητα για το pose του (δείτε και το παρακάτω σχήμα 2-4). Αυτά τα σφάλματα που εισάγονται στην οδομετρία προσπαθούμε να διορθώσουμε με αυτή την εργασία και ο τρόπος για το πετύχουμε αυτό γενικώς είναι να πάρουμε πολλές μετρήσεις από διαφορετικές πηγές (πολλά landmarks) και να συμπτύξουμε τις πληροφορίες με τη βοήθεια των μαθηματικών και της στατιστικής.



2-4 Αύξηση αβεβαιότητας της θέσης καθώς το ρομπότ κινείται

Πέρα από τα σφάλματα των αισθητήρων υπάρχει και το φαινόμενο της ψευδωνυμίας. Ας πάρουμε για παράδειγμα τις συνθήκες του προβλήματος που μελετάμε σε αυτή την εργασία. Έστω ότι το ρομπότ μας καταφέρνει να μετρήσει επακριβώς τη θέση των τοίχων στο δωμάτιο που βρίσκεται. Καθώς αντιπαραβάλλει αυτές τις μετρήσεις με το γνωστό χάρτη που έχει για τον ευρύτερο χώρο στον οποίο βρίσκεται μπορεί να υπάρχουν στον χάρτη δωμάτια με τις ακριβώς ίδιες διαστάσεις και προσανατολισμό. Σε αυτή την περίπτωση το ρομπότ είναι αδύνατο να ξέρει σε ποιο από αυτά τα δωμάτια βρίσκεται. Για να λυθεί αυτό το πρόβλημα πρέπει να πάρουμε μετρήσεις από διαφορετικούς αισθητήρες. Θα μπορούσαμε για παράδειγμα να χρησιμοποιήσουμε μια κάμερα για να εντοπίσουμε χαρακτηριστικά που είναι μοναδικά για κάθε δωμάτιο. Μια άλλη λύση είναι να εξοπλίσουμε τα δωμάτια με beacons που θα στέλνουν στο ρομπότ την πληροφορία σε ποιο δωμάτιο βρίσκεται. Στην παρούσα εργασία έχουμε μόνο ένα δωμάτιο στο χάρτη μας και έτσι αγνοούμε τελείως το πρόβλημα της ψευδωνυμίας.

Για να λυθεί το πρόβλημα του εντοπισμού και της χαρτογράφησης, που είναι αρκετά γενικό και πολύπλοκο, πρέπει να το χωρίσουμε σε μικρότερα πιο επιλύσιμα προβλήματα. Οι περισσότερες εργασίες που ασχολούνται με το πρόβλημα καταπιάνονται με το software και δεν ασχολούνται με το hardware. Κάποιες εργασίες προσπαθούν να λύσουν μόνο το πρόβλημα του εντοπισμού σε γνωστό χάρτη, ενώ κάποιες άλλες προσπαθούν να κάνουν και χαρτογράφηση. Σε όλες τις εργασίες παρατηρείται ότι εμφανίζονται αρκετά πολύπλοκα μαθηματικά προβλήματα τα οποία λύνονται με εκτεταμένους προγραμματιστικούς αλγόριθμους που απαιτούν μεγάλη υπολογιστική ισχύ. Δεν πρέπει να ξεχνάμε εδώ ότι ο τελικός αλγόριθμος που λύνει το πρόβλημα πρέπει να τρέχει κάθε φορά που το ρομπότ κάνει μια κίνηση, προχωρώντας μια μικρή απόσταση ή αλλάζοντας το pose του στη γενικότερη περίπτωση.

Πολλές εργασίες όπως προείπαμε ασχολούνται μόνο με το πρόβλημα του εντοπισμού του ρομπότ σε γνωστό χάρτη. Σε αυτή την εργασία απλουστεύουμε περισσότερο το πρόβλημα που έχουμε να επιλύσουμε. Προσπαθούμε να κάνουμε εντοπισμό σε γνωστό χάρτη με την επιπλέον πληροφορία ότι γνωρίζουμε την αρχική θέση του ρομπότ. Αυτό όπως θα φανεί παρακάτω μας βοηθά να απλουστεύσουμε το μαθηματικό πρόβλημα που καλούμαστε να επιλύσουμε. Έχοντας όμως κάνει αυτή την απλούστευση ο αλγόριθμός μας δεν μπορεί να δουλέψει στην περίπτωση που το ρομπότ μετακινηθεί από εξωτερικό παράγοντα και φτάσει σε άγνωστη θέση, η γενικότερα pose. (Το λεγόμενο kidnapped robot problem).

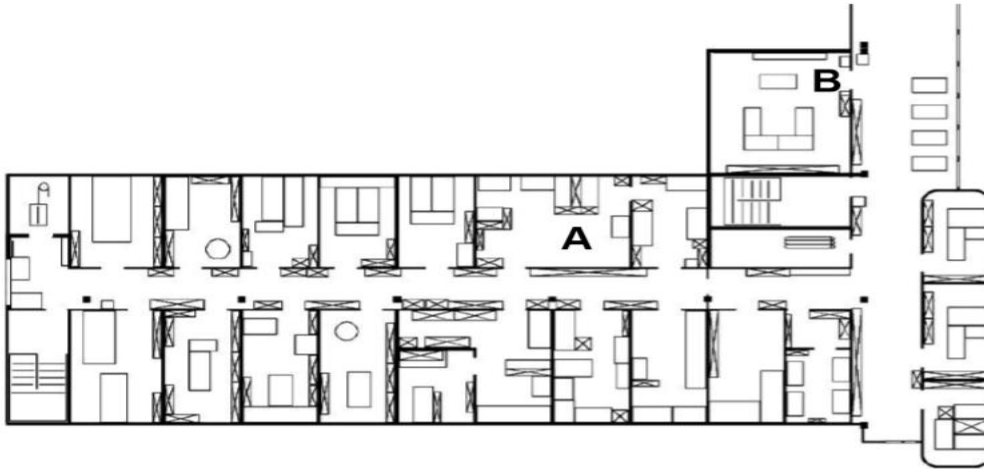
Ένας τρόπος για να λύσουμε το πρόβλημα που θέσαμε σε αυτή την εργασία είναι να χρησιμοποιήσουμε το φίλτρο Kalman. Το φίλτρο Kalman είναι ένας μαθηματικός αλγόριθμος που μας βοηθά να συμπτύξουμε την πληροφορία για τη θέση του ρομπότ που παίρνουμε από την οδομετρία, με πληροφορίες που παίρνουμε από laser scanners, με τους οποίους πρέπει να είναι εφοδιασμένο το ρομπότ, και μετράνε την απόστασή του από τους τοίχους του χώρου στον οποίο βρίσκεται (και γενικότερα από τα landmarks). Εφαρμόζοντας το φίλτρο Kalman κάθε landmark ανανεώνει και βελτιώνει την εκτίμηση που έχουμε για τη θέση που βρίσκεται το ρομπότ. Έτσι τελικά ανανεώνουμε τη θέση του ρομπότ και το εντοπίζουμε με την απαιτούμενη ακρίβεια μέσα στο χάρτη. (Περισσότερα για το φίλτρο Kalman θα αναφερθούν παρακάτω). Στην παρούσα εργασία χρησιμοποιούμε το Extended Kalman Filter.

Εδώ πρέπει να αναφέρουμε ότι όταν μιλάμε για landmarks εννοούμε περιοχές του χώρου που μπορούν να ταυτιστούν με χαρακτηριστικά του χάρτη. Για παράδειγμα τοίχοι σε ένα δωμάτιο, ή κάποιες κολώνες σε ένα φουαγιέ. Στην εργασία που εκπονήσαμε εδώ τοποθετήσαμε το ρομπότ σε ένα άδειο δωμάτιο με τέσσερις τοίχους, άρα τα μόνα landmarks που υπάρχουν είναι οι τέσσερις τοίχοι του χάρτη μας.

Τέλος πρέπει να αναφέρουμε ότι καθώς προχωράμε στην λύση του προβλήματος που κληθήκαμε να αντιμετωπίσουμε, βλέπουμε ότι εμφανίζονται κάποιες απαιτήσεις από το hardware στο οποίο τελικά θα τρέχει το software που προσφέρεται εδώ. Για παράδειγμα το laser που θα χρησιμοποιηθεί πρέπει να έχει ακρίβεια $\pm 0.0 - 0.05$ m στις μετρήσεις απόστασης (αυτός ο περιορισμός είναι αρκετά αυστηρός και laser με αυτές τις προδιαγραφές μπορούμε να τα βρούμε με αρκετά μεγάλη τιμή στην αγορά την ώρα που γράφεται αυτή η εργασία). Επιπλέον οι πληροφορίες από την οδομετρία πρέπει να έχουν Gaussian κατανομή τιμών με τυπική απόκλιση της τάξης του μέτρου (αυτός ο περιορισμός δεν είναι πολύ αυστηρός και μπορούμε εύκολα να τον πετύχουμε όταν κατασκευάζουμε ένα ρομπότ με ρόδες).

Το πρόβλημα που καλούμαστε να λύσουμε σε αυτή την εργασία είναι εξαιρετικά σημαντικό. Οποιοσδήποτε θέλει να κατασκευάσει ένα αυτόνομο ρομπότ πρέπει να λύσει το πρόβλημα του εντοπισμού του. Σκεφτείτε για παράδειγμα ένα ρομπότ που κάνει κάποια εργασία σε μια βιομηχανική αποθήκη. Θα πρέπει να ξέρει που ακριβώς βρίσκεται στην αποθήκη για να μπορεί να μετακινεί εμπορεύματα από ένα σημείο σε ένα άλλο χωρίς να εμποδίζει το υπόλοιπο προσωπικό και τα μηχανήματα. Ένα ρομπότ που βοηθά ένα άτομο με μειωμένη κινητικότητα πρέπει να εντοπίζει τον εαυτό του στο χώρο του ατόμου, ώστε να μπορεί να αλληλεπιδρά με ανθρώπινα όντα αλλά και να κινείται αυτόνομα στο χώρο ώστε να μην κάνει ζημιές. Εντοπίζοντας τον εαυτό του στο χώρο είναι σε θέση πλέον να ανοίγει ντουλάπια να καθαρίζει και ίσως να μαγειρεύει. Ακόμη και στην απλή περίπτωση που ένα αυτόνομο ρομπότ θέλει να ανοίξει μια πόρτα, αυτό πρέπει να ξέρει που να τοποθετήσει τον εαυτό του ώστε τα “χέρια” του να κάνουν τις κατάλληλες κινήσεις προς το πόμολο της πόρτας.

2.1.Πλοήγηση ρομπότ με βάση τον εντοπισμό, σε σύγκριση με προγραμματιστικές λύσεις.



2-5 Σύνηθες περιβάλλον περιήγησης ενός ρομπότ. [3]

Η παραπάνω εικόνα (3-1) δείχνει ένα σύνηθες εσωτερικό περιβάλλον στο οποίο μπορεί να κλιθεί να κινηθεί ένα ρομπότ. Ας υποθέσουμε ότι το ρομπότ πρέπει να παραδώσει μηνύματα μεταξύ δύο συγκεκριμένων δωματίων σε αυτό το περιβάλλον, τα δωμάτια A και B. Για τη δημιουργία ενός συστήματος περιήγησης είναι σαφές ότι το ρομπότ θα χρειαστεί αισθητήρες και κάποιο σύστημα ελέγχου κίνησης. Οι αισθητήρες είναι απαραίτητοι για την αποφυγή εμποδίων όπως άνθρωποι, και το σύστημα ελέγχου κίνησης χρειάζεται ώστε το ρομπότ να κινείται μόνο του.

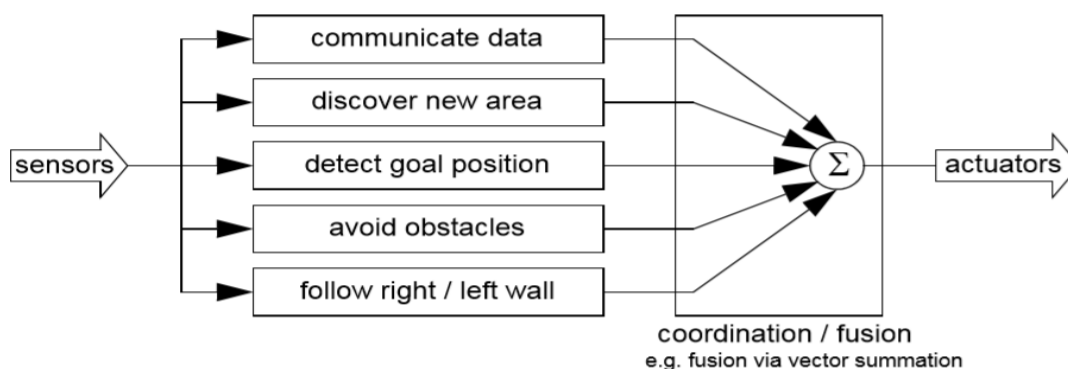
Ενώ είναι σαφές ότι το ρομπότ θα χρειαστεί κάποιους αισθητήρες και κάποιους κινητήρες, δεν είναι σαφές ότι χρειάζεται κάποιο σύστημα εντοπισμού στο χώρο. Ο εντοπισμός μπορεί να φαίνεται απαραίτητος για την επιτυχή πλοήγηση μεταξύ των δύο δωματίων. Έτσι κι αλλιώς μέσω του εντοπισμού στο χάρτη το ρομπότ μπορεί να ελπίζει ότι θα ανακτήσει τη θέση του και θα εντοπίσει πότε έφτασε στον στόχο του. Το ρομπότ πρέπει να έχει ένα τρόπο να εντοπίσει την θέση του τελικού στόχου του. Παρόλα αυτά ο εντοπισμός με χρήση ενός χάρτη δεν είναι η μόνη στρατηγική για τον εντοπισμό του στόχου του ρομπότ.

Μια εναλλακτική προτείνει ότι επειδή οι αισθητήρες και οι κινητήρες έχουν θόρυβο, θα πρέπει κανείς να αποφύγει να δημιουργήσει ένα γεωμετρικό χάρτη για τον εντοπισμό. Αντ' αυτού κάποιοι προτείνουν να σχεδιαστούν σεντ συμπεριφοράς που συνδυασμένα μεταξύ τους θα έχουν ως αποτέλεσμα την επιθυμητή κίνηση του ρομπότ. Κατά βάση αυτή η προσέγγιση αποφεύγει την κατηγορηματική λογική σχετικά με τον εντοπισμό και τη θέση και γενικώς αποφεύγει τον απόλυτο σχεδιασμό πορείας.

Αυτή η τεχνική βασίζεται στην αντίληψη ότι υπάρχει μια διαδικαστική λύση σε κάθε συγκεκριμένο πρόβλημα πλοήγησης του ρομπότ. Για παράδειγμα στην εικόνα 2-6 η διαδικαστική προσέγγιση στην πλοήγηση από το δωμάτιο A στο δωμάτιο B θα μπορούσε να ήταν η σχεδίαση μιας συμπεριφοράς κατά την οποία το ρομπότ θα ακολουθεί τον αριστερό τοίχο και στο δωμάτιο B θα υπάρχει ένα μοναδικό χαρακτηριστικό το οποίο το ρομπότ θα μπορεί να εντοπίσει. Όπως για παράδειγμα το χρώμα του χαλιού στο δωμάτιο. Έτσι το ρομπότ μπορεί να φτάσει στο δωμάτιο B ακολουθώντας τον αριστερό τοίχο και να καταλάβει

ότι έφτασε στον στόχο, οπότε θα σταματήσει τον αλγόριθμο, αντιλαμβανόμενο το χρώμα του χαλιού στο δωμάτιο B.

Η αρχιτεκτονική της λύσης σε ένα συγκεκριμένο περιβάλλον περιήγησης φαίνεται στο παρακάτω σχήμα (2-6). Το βασικό πλεονέκτημα αυτής της μεθόδου είναι ότι όταν είναι δυνατόν μπορεί να σχεδιαστεί εύκολα για ένα συγκεκριμένο περιβάλλον με ένα μικρό αριθμό στόχων που πρέπει να φτάσει το ρομπότ. Παρόλα αυτά η μέθοδος αυτή έχει κάποια μειονεκτήματα. Καταρχήν αυτή η μέθοδος δεν μπορεί άμεσα να εφαρμοστεί σε άλλα προβλήματα. Συχνά ο κώδικα περιήγησης του ρομπότ απευθύνεται σε ένα μόνο περιβάλλον και για να κινηθεί το ρομπότ σε διαφορετικό περιβάλλον ο κώδικας πρέπει να σχεδιαστεί από την αρχή.



2-6 Διαδικαστικός τρόπος περιήγησης του ρομπότ. [3]

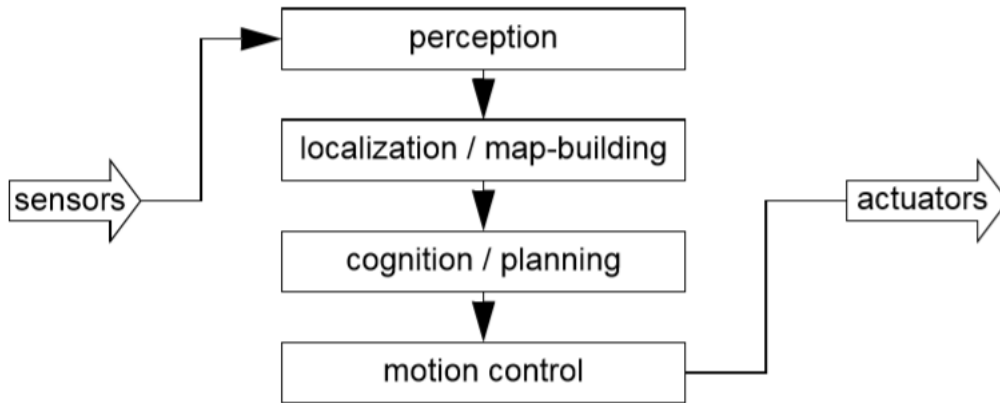
Επιπλέον οι διαδικασίες που θα σχεδιαστούν για την περιήγηση όπως η λογική να ακολουθήσει το ρομπότ τον τοίχο στο αριστερό του “χέρι” πρέπει να σχεδιαστούν προσεκτικά για να παραχθεί η επιθυμητή συμπεριφορά. Αυτή η διαδικασία είναι χρονοβόρα και εξαρτάται σε μεγάλο βαθμό από το συγκεκριμένο hardware του ρομπότ και από τα χαρακτηριστικά του περιβάλλοντος.

Ένα σύστημα που βασίζεται σε προγραμματισμό συμπεριφοράς μπορεί να έχει πολλές συμπεριφορές ενεργές σε μια στιγμή. Ακόμη και όταν συγκεκριμένες συμπεριφορές έχουν ρυθμιστεί κατάλληλα για να βελτιστοποιήσουν την επίδοση του ρομπότ, η σύμπτυξη συμπεριφορών και η μετάβαση από μια συμπεριφορά σε μια άλλη μπορεί να μηδενίσει την βελτιστοποίηση. Συχνά η προσθήκη καινούριων συμπεριφορών αναγκάζει τον σχεδιαστή να αλλάξει όλες τις υπάρχουσες συμπεριφορές για να σιγουρέψει την σταθερότητα του συστήματος.

Σε αντίθεση με τον προγραμματισμό συμπεριφορών, η προσέγγιση του εντοπισμού σε ένα χάρτη περιλαμβάνει ενότητες για τον εντοπισμό και ενότητες για την νόηση. Στην πλοήγηση μέσω χάρτη το ρομπότ προσπαθεί κατηγορηματικά να εντοπίσει τον εαυτό του συγκεντρώνοντας δεδομένα από τους αισθητήρες, και έπειτα ανανεώνοντας την εντύπωση που έχει σχετικά με τη θέση του με βάση τον χάρτη του περιβάλλοντος. Τα πλεονεκτήματα της πλοήγησης μέσω χάρτη είναι τα εξής:

- Η ξεκάθαρη αντίληψη, που βασίζεται στο χάρτη, για την θέση του ρομπότ κάνει την αντίληψη του συστήματος για τη θέση άμεσα διαθέσιμη στον ανθρώπινο χειριστή.

- Η ίδια η ύπαρξη του χάρτη αντιπροσωπεύει ένα μέσο για την επικοινωνία μεταξύ ρομπότ και του ανθρώπου που μπορεί να το χειρίζεται. Αν το ρομπότ μεταβεί σε καινούριο περιβάλλον ο άνθρωπος απλώς του δίνει ένα καινούριο χάρτη.
- Ο χάρτης, αν δημιουργείται από το ρομπότ, μπορεί να χρησιμοποιηθεί και από ανθρώπους.



2-7 Περιήγηση μέσω εντοπισμού και χάρτη. [3]

Η βασισμένη σε χάρτη προσέγγιση θα απαιτήσει περισσότερη προσπάθεια για τον προγραμματισμό του ρομπότ. Η ελπίδα είναι ότι η επιπλέον προσπάθεια σχεδίασης θα έχει ως αποτέλεσμα μια αρχιτεκτονική που μπορεί επιτυχώς να κάνει χαρτογράφηση και να πλοηγηθεί σε μια ποικιλία από περιβάλλοντα.

Το ρίσκο σε αυτή τη προσέγγιση είναι ότι κατασκευάζεται μια εσωτερική αναπαράσταση αντί για τον πραγματικό κόσμο και σε αυτή την αναπαράσταση βασίζεται το ρομπότ. Αν το μοντέλο που έχει φτιάξει το ρομπότ παρεκκλίνει από τη πραγματικότητα τότε η συμπεριφορά του ρομπότ μπορεί να μην είναι η επιθυμητή.

Σε αυτή την εργασία ασχολούμαστε με την προσέγγιση που είναι βασισμένη σε χάρτη, και ποιο συγκεκριμένα ασχολούμαστε με τον εντοπισμό του ρομπότ σε γνωστό χάρτη.

2.2.Κατηγοριοποίηση μηχανισμών εντοπισμού.

Ο εντοπισμός στον τομέα της ρομποτικής είναι αρκετά αχανής ώστε να χρειάζεται κατηγοριοποίηση. Αναφορικά με την κατηγοριοποίηση, υπάρχουν πολλές προσεγγίσεις από διαφορετικές οπτικές γωνίες. Θα αρχίσουμε την κατηγοριοποίηση των μηχανισμών εντοπισμού ως εξής

Παθητικός εντοπισμός: όπου υπάρχουν σήματα που παρατηρούνται και επεξεργάζονται από το σύστημα με σκοπό να συνάγουν την θέση των επιθυμητών χαρακτηριστικών. Ανάλογα με τις προδιαγραφές των σημάτων, χρειάζεται κάποια ευελιξία λόγω της παθητικότητας.

Ενεργός εντοπισμός: ο μηχανισμός εντοπισμού παράγει και χρησιμοποιεί τα δικά του σήματα για τον σκοπό του εντοπισμού.

Το ποιο είδος εντοπισμού θα προτιμηθεί έχει να κάνει με την εφαρμογή. Παρόλα αυτά μια απλή σύγκριση μας δείχνει ότι η δεύτερη προσέγγιση θα ήταν περισσότερο ανεξάρτητη από

το περιβάλλον και επομένως ποιο αξιόπιστη σε περισσότερες εφαρμογές. Αυτό φυσικά θα έχει αντιστάθμισμα κάποιες απαιτήσεις όπως υπολογιστική ισχύ, και περισσότερο hardware. Από μια άλλη οπτική γωνία που έχει να κάνει με το hardware που χρησιμοποιείται μπορούμε να παρουσιάσουμε τις παρακάτω τεχνικές εντοπισμού που μπορεί να είναι ενεργές ή παθητικές.

1. Άμεση διευθέτηση: χρησιμοποιεί κωδικοποιητές για να αντιληφθεί κινήσεις από στροφές βασιζόμενη στην ολοκλήρωση. Ανάλογα με την εφαρμογή υπάρχουν διαφορετικές λύσεις. Αυτή η κατηγορία είναι η πιο βασική αλλά και η πιο συνηθισμένη στην ρομποτική. Λόγο των χαρακτηριστικών της αυτή η μέθοδος έχει πολύ θόρυβο και δεν είναι αποδοτική. Από την άλλη λόγω της διασημότητας της έχει γίνει πολύ έρευνα σε αυτή τη μέθοδο και αυτό φαίνεται στην ποιότητα της μεθόδου.
2. INS μέθοδοι: αυτές οι μέθοδοι βασίζονται σε αισθητήρες αδράνειας, επιταχυνσιόμετρα και ανιχνευτές ηλεκτρομαγνητικών πεδίων και βαρύτητας. Επίσης βασίζονται στην ολοκλήρωση πάνω σε στοιχεία κίνησης, και έτσι μπορεί τελικά να συσσωρεύσουν θόρυβο ιδίως αν χρησιμοποιούνται αισθητήρες που μειώνουν την ολίσθηση. Αυτές οι μέθοδοι είναι πλέον αρκετά ολοκληρωμένες λόγω των ανεπτυγμένων μαθηματικών εργαλείων.
3. Όραση και οπτική οδομετρία: χρησιμοποιώντας μια κάμερα, ή στερεοσκοπική όραση αυτή η λύση μπορεί να χρησιμοποιηθεί για να μας δώσει περισσότερη πληροφορία από ένα απλό εντοπισμό. Αυτή η λύση μπορεί να απαιτεί αρκετά μεγαλύτερη υπολογιστική ισχύ ειδικά αν χρησιμοποιείται σε δυναμικό περιβάλλον, ή αν ασχολείται με μη στατικά χαρακτηριστικά. Παρόλα αυτά είναι πολύ δημοφιλής μέθοδος και υποστηρίζεται από τεχνικές επεξεργασίας σήματος γενετικούς αλγόριθμους και εξελικτικούς αλγόριθμους.
4. Αποστασιομέτρηση (ranging): Χρησιμοποιεί κάποιο μέσω μέτρησης απόστασης όπως laser, υπέρυθρη ακτινοβολία, ακουστικές συχνότητες ή ραδιοσήματα. Η αποστασιομέτρηση μπορεί να γίνει χρησιμοποιώντας διάφορες τεχνικές όπως καταγράφοντας το Time of Flight του σήματος, τη χρονική διαφορά της άφιξης ή το Round Trip Time of Flight του ακτινοβολούμενου σήματος, όπως και τη γωνία άφιξης. Αυτή η μέθοδος μπορεί να χρησιμοποιηθεί συνδυαζόμενη με παραδοσιακές τεχνικές για να βελτιώσει το αποτέλεσμα. Όπως γίνεται σε αυτή την εργασία όπου η οδομετρία συνδυάζεται με τις μετρήσεις απόστασης από laser.

Υπάρχουν επίσης μερικές όχι τόσο κοινές προσεγγίσεις στον εντοπισμό όπως οι παρακάτω

Αισθητήρες Doppler: Αυτοί οι αισθητήρες μπορούν να μετρήσουν ταχύτητα κινούμενων αντικειμένων. Ένα ημιτονοειδές σήμα εκπέμπεται από μια κινούμενη βάση και μετράται η ηχώ του σήματος από ένα δέκτη. Αυτοί οι αισθητήρες χρησιμοποιούν υπερήχους ή ραδιοφωνικά κύματα. Ανάλογα με το μήκος κύματος διαφοροποιείται η ακρίβεια. Όσο μικρότερο είναι το μήκος κύματος τόσο μεγαλύτερη είναι η ανάλυση του αισθητήρα. Αυτή η λύση βασίζεται στην ολοκλήρωση διανυσμάτων ταχύτητας με βάση το χρόνο.

Ιχνηλάτες ηλεκτρομαγνητικών κυμάτων: Μπορούν να εντοπίσουν τη θέση και τον προσανατολισμό ενός αντικειμένου με μεγάλη ακρίβεια. Είναι ακριβές μέθοδοι και οι αισθητήρες δεν έχουν μεγάλη εμβέλεια και είναι πολύ ευαίσθητοι σε μεταλλικά αντικείμενα. Αυτοί οι περιορισμοί κάνουν τέτοιες μεθόδους διαθέσιμες μόνο σε εφαρμογές όπως παιχνίδια υπολογιστών που υπολογίζουν τη στάση του σώματος ενός παίκτη.

Ιχνηλάτες φωτός: είναι πολύ ο ισχυροί και τυπικά μπορούν να πετύχουν μεγάλη ακρίβεια και ανάλυση. Παρόλα αυτά είναι ακριβοί και μηχανικά πολύπλοκοι. Ένα παράδειγμα είναι το σύστημα ιχνηλασίας κίνησης κεφαλιού.

Ο σωστός συνδυασμός οποιονδήποτε από τις παραπάνω τεχνικές εντοπισμού μπορεί να δώσει καλύτερα αποτελέσματα αλλά ταυτόχρονα κάνει την ρουτίνα εντοπισμού πιο πολύπλοκη και απαιτεί περισσότερη υπολογιστική ισχύ.

2.3.Βελτίωση απόδοσης ανιχνευτή απόστασης laser στον εντοπισμό μέσω laser.

Υπάρχουν αισθητήρες laser που παρέχουν πληροφορίες για την απόσταση αλλά και τη φωτεινότητα (όπου με τον όρο φωτεινότητα αναφερόμαστε στην δύναμη του σήματος) της ανακλώμενης ακτίνας καθώς και πληροφορία για την θερμοκρασία του αισθητήρα. Σε κάποιους αισθητήρες όπως ο AccuRange 4000 αυτές οι πληροφορίες έχουν μορφή αναλογικού σήματος.

Γενικά η ακρίβεια στη μέτρηση της γωνίας από αισθητήρες laser, που μετράνε και την γωνία εκτός από την απόσταση, εξαρτάται από το μέγεθος της κηλίδας της ακτίνας laser. Έτσι με τους αισθητήρες Laser η ανάλυση στην γωνία είναι πολύ μεγάλη, το σφάλμα στη μέτρηση της γωνίας μπορεί να θεωρηθεί μηδενικό.

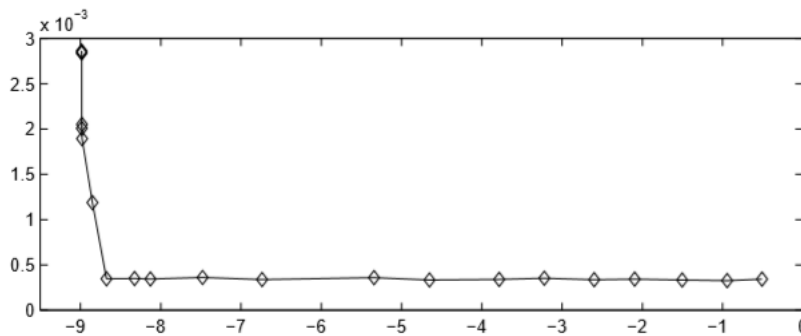
Όσο αναφορά στην ακρίβεια μέτρησης της απόστασης, υπάρχουν αρκετοί παράγοντες που επηρεάζουν την έκταση του θορύβου. Ένας είναι το πλάτος του ανακλώμενου σήματος, που μπορεί να μετρηθεί αν υπάρχει κατάλληλος αισθητήρας laser. Ένας άλλος είναι η ολίσθηση και οι διακυμάνσεις στο hardware του αισθητήρα. Ένας ακόμη είναι ο θόρυβος από τα ηλεκτρονικά που μετατρέπουν το αναλογικό σήμα σε ψηφιακό.

Έχει εξεταστεί από εργασίες (Adams M.D. Optical Range Analysis of Stable Target Pursuit in Mobile Robotics 1992) η αρχή της ολίσθησης φάσης της μέτρησης και μας έχει δώσει μια σχέση ανάμεσα στη μεταβλητότητα της εμβέλειας σε σχέση με το πλάτος του σήματος:

$$\sigma_p^2 = \frac{a}{V_r} + b$$

Όπου σ_p^2 είναι η μεταβλητότητα της εμβέλειας και V_r είναι το μετρούμενο πλάτος. Στην εργασία [1] βρέθηκε μια ανάστροφη, ελαφρός μη γραμμική σχέση μετά την αναγνώριση. Όπου για την αναγνώριση έγινε το παρακάτω πείραμα. Χρησιμοποιήθηκε ένας σταθερός στόχος στο 1m. Χρησιμοποιώντας ένα κομμάτι φιλμ Kodac μεταβλήθηκε η ένταση του επιστρέφοντος σήματος laser συστηματικά παίρνοντας 10000 μετρήσεις για κάθε διαφορετική απόχρωση του γκρι του φιλμ. Αυτό έγινε για 20 αποχρώσεις του γκρι.

Με αυτό το πείραμα παρατηρήθηκε έντονη αύξηση του θορύβου για μετρήσεις κάτω από ένα συγκεκριμένο επίπεδο έντασης του επιστρεφόμενου σήματος laser. Αυτό απλουστεύει το μοντέλο για τη μεταβλητότητα της εμβέλειας σε μια σταθερή τιμή ανεξάρτητη της απόστασης και της έντασης του σήματος. Τα αποτελέσματα του πειράματος φαίνονται στο παρακάτω σχήμα 3-4.



2-8 Τυπική απόκλιση της εμβέλειας (y άξονας) σε σχέση με το μετρούμενο πλάτος του σήματος (x άξονας).[1]

Το βασικό αποτέλεσμα που πρέπει να υιοθετήσουμε από αυτή την εργασία είναι ότι μπορούμε να απορρίψουμε μετρήσεις του αισθητήρα laser σε σχέση με τις μετρήσεις της φωτεινότητας της δέσμης. Κάτω από μια συγκεκριμένη φωτεινότητα οι μετρήσεις δεν είναι αξιόπιστες. Αυτό είναι πολύ σημαντικό στις εφαρμογές όπου μετράμε απόσταση με laser στις οποίες υπάρχουν επιφάνειες που μεταβάλλουν τη φωτεινότητα του σήματος και όχι μόνο.

2.4. Χρήση κάμερας στη διαδικασία εντοπισμού.

Μιας και ασχολούμαστε με την κατάσταση της τέχνης, πρέπει να αναφέρουμε ότι για την διαδικασία του εντοπισμού χρησιμοποιούνται και κάμερες όπως και άλλου τύπου ανιχνευτές απόστασης πχ υπέρηχοι. Ο εντοπισμός μέσω κάμερας είναι ένας τομέας που εξελίσσεται πάρα πολύ τα τελευταία χρόνια. Στον εντοπισμό μέσω κάμερας μπορεί να χρησιμοποιούνται τόσο κάμερα όσο και laser, ή δύο κάμερες, ή στερεοσκοπική όραση. Οι εφαρμογές με κάμερες μπορούν να δώσουν καλύτερα αποτελέσματα από το laser όμως απαιτούν πολύ μεγαλύτερη υπολογιστική ισχύ και έχουν πολύ πιο πολύπλοκους αλγόριθμους από τις εφαρμογές με laser. Επίσης απαιτείται λεπτομερέστερος χάρτης. Ας τα δούμε όμως όλα αυτά μέσα από ένα παράδειγμα. Θα μελετήσουμε μια εφαρμογή που συνδυάζει κάμερα και laser και βρίσκεται στην αναφορά [1].

Σε αυτή την εφαρμογή εξάγονται χαρακτηριστικά γραμμών από τους τοίχους μέσω του αισθητήρα laser. Χρησιμοποιώντας πολικές συντεταγμένες εξάγονται γραμμές από το περιβάλλον σύμφωνα με τον τύπο

$$\rho \cos(\varphi - \alpha) - r = 0$$

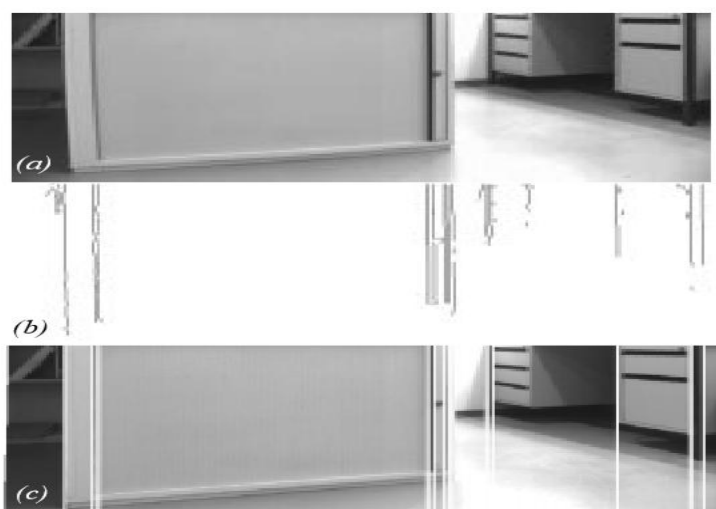
Όπου ρ και φ είναι τα δεδομένα από το laser και α και r είναι οι παράμετροι του μοντέλου. Με κάποιο μαθηματικό μοντέλο που δεν θα αναφερθεί εδώ οι τοίχοι μπορούν να γίνουν ευδιάκριτοι ως ευθείες ακόμη και αν υπάρχουν εμπόδια μπροστά τους όπως πόρτες ή άλλες ασυνέχειες. Οι τοίχοι αυτοί έπειτα συγκρίνονται με τους τοίχους στον χάρτη του ρομπότ.

Αυτό που μας ενδιαφέρει περισσότερο σε αυτή την εφαρμογή είναι η χρήση της CCD κάμερας. Το σύστημα όρασης αποτελείται από μια κάμερα Plunix TM-9701 αποχρώσεων του γκρι. Έχει ανάλυση 640x480 και η επεξεργασία της εικόνας γίνεται από τον κεντρικό επεξεργαστή του ρομπότ. Ένα σύστημα όρασης που έχει ως σκοπό να εξάγει ακριβή γεωμετρικά χαρακτηριστικά του χώρου απαιτεί εκτενή ρύθμιση. Για αυτό το λόγο πρέπει να

καθοριστούν παράμετροι της κάμερας όπως η θέση της και ο προσανατολισμός της. Το κέντρο της εικόνας ο παράγοντας κλίμακας, το εστιακό βάθος και παράμετροι παραμόρφωσης. Για να υπολογιστούν αυτά τα εσωτερικά και εξωτερικά χαρακτηριστικά της κάμερας πρέπει να γίνει ρύθμιση (καλιμπράρισμα) της κάμερας σε ένα γνωστό περιβάλλον.

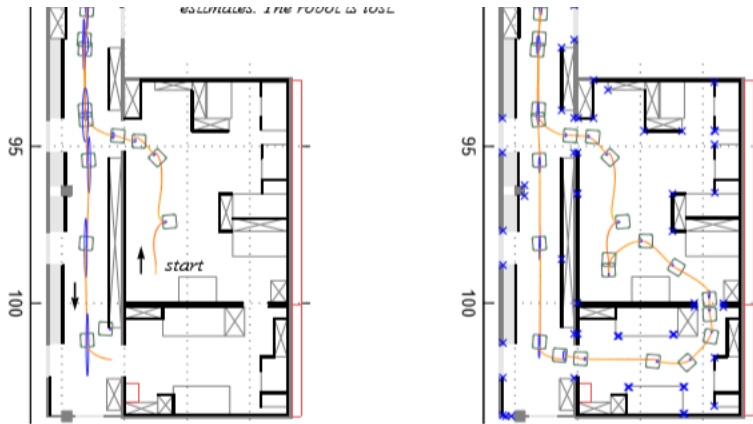
Επειδή η κάμερα έχει σταθερή οριζόντια θέση οι κατακόρυφες δομές της εικόνας (πχ κατακόρυφες ακμές όπως αυτές σε μια πόρτα ή μια ντουλάπα) είναι αμετάβλητες. Επιπλέον δεν απαιτούν πολύπλοκες τεχνικές επεξεργασίας εικόνας για την εξαγωγή τους που είναι σημαντικό σε εφαρμογές real time. Σε αυτή την εργασία χρησιμοποιήθηκε μόνο το κάτω μισό της εικόνας που παίρνει η κάμερα για πρακτικούς λόγους, κυρίως για την αποφυγή μεγάλων αλλαγών στο φωτισμό όπως από παράθυρα και λάμπες.

Χρησιμοποιώντας μια διαδικασία εξαγωγής των χαρακτηριστικών που ενδιαφέρουν από την εικόνα, το ταίριασμα των κατακόρυφων γραμμών καταλήγει να είναι ένα μονοδιάστατο πρόβλημα. Σε κάθε βήμα της διαδικασίας η αβεβαιότητα των παραμέτρων ρύθμισης καθώς και μια σταθερή αβεβαιότητα για τη θέση x διαδίδονται στους υπολογισμούς. Το τελικό αποτέλεσμα είναι μια σειρά από κατακόρυφες γραμμές που περιγράφεται από δύο αριθμούς, τη γωνία ϕ και το τετράγωνο της τυπικής απόκλισης σ_ϕ^2 . Η εξαγωγή των ακμών φαίνεται στην παρακάτω εικόνα.



2-9 Εξαγωγή κατακόρυφων ακμών α) η αρχική μισή εικόνα, β) τα pixel που ανήκουν στις ακμές μετά από κατάλληλη επεξεργασία, γ) οι προκύπτουσες κατακόρυφες ακμές.[1]

Η πληροφορία που παίρνουμε από τις ακμές της εικόνας συνδυάζεται με αυτή από το laser με χρήση του EKF. Τα τελικά αποτελέσματα συγκρίνονται με τον χάρτη ο οποίος δεν έχει μονο πληροφορία για τους τοίχους αλλά περιέχει και την επιπλέον πληροφορία για το που βρίσκονται οι ακμές που κοιτά η κάμερα στον χώρο. Όπως καταλαβαίνουμε άμεσα για να κατασκευαστεί αυτός ο χάρτης χρειάζεται μια επιπλέον διαδικασία ώστε να περιέχει τις παραπάνω πληροφορίες. Στο παρακάτω σχήμα φαίνεται ο απλός χάρτης του χώρου αλλά και ο χάρτης με την επιπλέον πληροφορία για τις ακμές.



2-10 Αριστερά ο απλός χάρτης που χρησιμοποιεί μια εφαρμογή με αισθητήρα laser. Δεξιά ο περισσότερο πολύπλοκος χάρτης που χρησιμοποιεί η εφαρμογή με την κάμερα. Η επιπλέον πληροφορία των ακμών του χώρου φαίνεται με τα μπλέ x στον αριστερό χάρτη. [1]

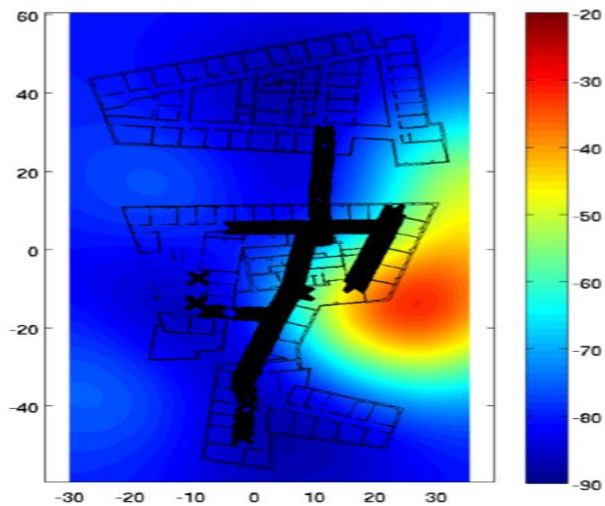
Από αυτό αντιλαμβανόμαστε ότι για να χρησιμοποιήσουμε κάμερα με τον παραπάνω τρόπο χρειάζεται αρκετά περισσότερη υπολογιστική προσπάθεια από την πλευρά του ανθρώπου, όχι μόνο για να εξαχθούν τα χαρακτηριστικά από τις εικόνες αλλά και για να κατασκευαστεί ο χάρτης του χώρου. Επιπλέον επειδή η επεξεργασία εικόνας είναι αρκετά χρονοβόρα, για να είναι αυτή η εφαρμογή real time χρειάζεται αρκετά μεγάλη υπολογιστική ισχύ. Ένα απλό raspberry PI δεν να μπορούσε να ανταποκριθεί καλά σε μια real time εφαρμογή με χρήση όλων των αισθητήρων που αναφέρθηκαν στην εργασία [1] και έτσι για μια τέτοια εφαρμογή θα πρέπει να κατασκευαστεί αρκετά μεγάλο σε διαστάσεις ρομπότ. Για την εργασία αυτή χρησιμοποιήθηκε το ρομπότ Πυγμαλίων, ο επεξεργαστής του είναι στα 100 MHz. Το μέγεθος του ρομπότ της εργασίας [1] φαίνεται στο παρακάτω σχήμα.



2-11 Το ρομπότ Πυγμαλίων, που χρησιμοποιήθηκε στην εργασία [1] το 1999.

2.5. Εντοπισμός μέσω WiFi.

Ο εντοπισμός μέσω WiFi που γίνεται χρησιμοποιώντας γραφικές κατανομές του σήματος WiFi σε χάρτη, μας δίνει έναν ακριβή τρόπο εντοπισμού του ρομπότ στο χάρτη. Σε αυτή την προσέγγιση η μέση τιμή και η τυπική απόκλιση του σήματος WiFi RSSI προσεγγίζονται χρησιμοποιώντας γραμμική παρεμβολή και αποτυπώνονται σε ένα γράφημα του χώρου. Αυτό μας οδηγεί σε μια εξίσωση που υπολογίζει την πιθανότητα παρατήρησης της ισχύος του σήματος WiFi RSSI σε κάθε σημείο του χώρου που λύνεται εύκολα. Η επίλυση της εξίσωσης για κάθε πιθανότητα παρατήρησης είναι ανεξάρτητη των δειγμάτων που χρησιμοποιούνται για την εκπαίδευση του συστήματος. Το μειονέκτημα είναι ότι για να κατασκευαστεί η γραφική της κατανομής του σήματος WiFi RSSI πάνω στο χάρτη του χώρου πρέπει το ρομπότ να τοποθετηθεί με ακρίβεια σε πολλές θέσεις στο χάρτη και να πάρει δείγματα για την εκπαίδευση του συστήματος. Παρόλα αυτά αυτός ο τρόπος εντοπισμού μπορεί να απλοποιηθεί χρησιμοποιώντας γκαουσιανές διαδικασίες κάτι που δεν απαιτεί να παρθούν δείγματα εκπαίδευσης από συγκεκριμένες θέσεις του χάρτη.

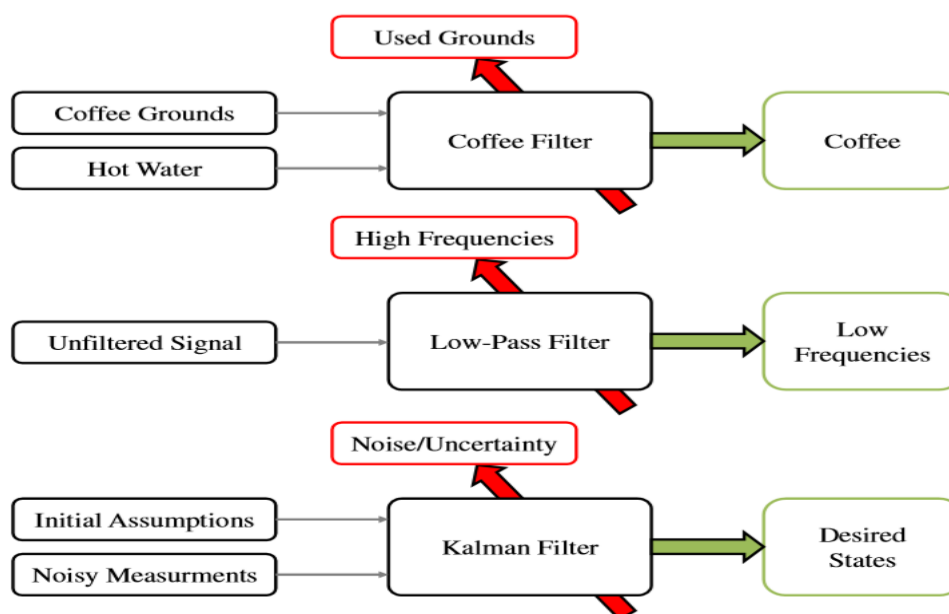


2-12 Gaussian Process-learnt WiFi γράφημα για ένα μοναδικό σημείο WiFi access point. Οι μέσες τιμές RSSI έχουν χρωματικό κώδικα από -90 dbm έως και -20dbm. Οι θέσεις του ρομπότ φαίνονται στο χάρτη με χ

3. Kalman Filter

3.1. Τι είναι το φίλτρο Kalman

Ας ξεκινήσουμε από τον όρο. Kalman είναι ο βασικός σχεδιαστής του φίλτρου που ακούει στο όνομα Rudolf Kalman. Επομένως αυτό είναι απλώς το όνομα που δίνεται σε φίλτρα ενός συγκεκριμένου τύπου. Το φίλτρο Kalman συχνά αναφέρεται και ως γραμμική τετραγωνική εκτίμηση. Ας δούμε τώρα γιατί λέγεται φίλτρο. Όλα τα φίλτρα έχουν ένα κοινό σκοπό: να αφήσουν κάτι να περάσει ενώ κάτι άλλο δεν περνά. Για παράδειγμα ένα φίλτρο καφέ αφήνει το υγρό να περάσει ενώ οι στερεοί κόκκοι του καφέ δεν περνάνε. Άλλο παράδειγμα είναι ένα χαμηλοπερατό φίλτρο, όπου οι χαμηλές συχνότητες περνάνε αλλά οι υψηλές όχι. Το φίλτρο Kalman λειτουργεί κι αυτό ως φίλτρο, αλλά η λειτουργία του είναι λίγο πιο πολύπλοκη και δυσνόητη. Το φίλτρο Kalman δέχεται στην είσοδο πληροφορία που ξέρουμε ότι έχει κάποιο λάθος, αβεβαιότητα ή θόρυβο. Ο στόχος του φίλτρου είναι να λάβει αυτή την πληροφορία, να ξεχωρίσει τα χρήσιμα κομμάτια που έχουν κάποια αξία και να μειώσει την αβεβαιότητα ή το θόρυβο.



3-1 Διάφοροι τύποι φίλτρων. [2]

3.2. Υλοποίηση φίλτρου Kalman

3.2.1. Καθορισμός Προβλήματος.

Το φίλτρο Kalman έχει σχεδιαστεί για να δουλεύει σε γραμμικά προβλήματα.

$$\mathbf{x}_k = \mathbf{F}_{k-1} \mathbf{x}_{k-1} + \mathbf{G}_{k-1} \mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (1)$$

$$\mathbf{y}_k = \mathbf{H}_{k-1} \mathbf{x}_k + \mathbf{v}_k \quad (2)$$

Όπου οι παραπάνω μεταβλητές ορίζονται στον παρακάτω πίνακα:

Variable	Description	Dimension
\mathbf{x}	State Vector	$n_x \times 1$
\mathbf{y}	Output Vector	$n_y \times 1$
\mathbf{u}	Input Vector	$n_u \times 1$
\mathbf{w}	Process Noise Vector	$n_x \times 1$
\mathbf{v}	Measurement Noise Vector	$n_y \times 1$
\mathbf{F}	System Matrix – State	$n_x \times n_x$
\mathbf{G}	System Matrix – Input	$n_x \times n_u$
\mathbf{H}	Observation Matrix	$n_y \times n_x$

3-2 Μεταβλητές των εξισώσεων Kalman [2].

Σημειώστε ότι ο αριθμός των καταστάσεων, οι έξοδοι (outputs) και οι εισοδοί (inputs) είναι ανεξάρτητοι, έτσι οι πίνακες \mathbf{G} και \mathbf{H} μπορεί να μην είναι τετραγωνικοί, αλλά ο πίνακας \mathbf{F} είναι πάντα τετραγωνικός. Η δυναμική του συστήματος περιγράφεται από την εξίσωση (1) ενώ η έξοδος του συστήματος περιγράφεται από την (2). Οι εξισώσεις (1) και (2) είναι εξισώσεις διανυσμάτων. (Τα διανύσματα είναι με bold γράμματα).

Το διάνυσμα κατάστασης \mathbf{x} είναι οι τιμές που θα εκτιμηθούν από το φίλτρο. Χρησιμοποιώντας αναλογίες φίλτρου, τα στοιχεία αυτού του διανύσματος είναι τα πράγματα που θέλουμε να περάσουν από το φίλτρο. Μερικές φορές μπορεί να συμπεριλαμβάνονται στο διάνυσμα αυτό περισσότερα από τα πράγματα που είναι απαραίτητα ώστε να καθοριστεί τι πραγματικά χρειάζεται. Για παράδειγμα αν θέλουμε να καθορίσουμε την θέση ενός αντικειμένου χρησιμοποιώντας πληροφορία σχετική με την επιτάχυνση του, θα χρειαστεί πιθανώς να εκτιμηθεί και η ταχύτητα του αντικειμένου. Αυτή είναι μια σημαντική διαφορά μεταξύ των διανυσμάτων κατάστασης και εξόδου του συστήματος. Ενώ σε άλλες περιπτώσεις αυτό που ζητάμε είναι η έξοδος, για προβλήματα εκτίμησης κατάστασης χρησιμοποιώντας το φίλτρο Kalman, η κατάσταση είναι το επιθυμητό αποτέλεσμα.

Το διάνυσμα εξόδου \mathbf{y} δεν είναι αυτό που επιχειρείται να εξαχθεί από το φίλτρο, αλλά είναι αυτό που μπορεί να μετρηθεί. Χρειάζεται να είναι ικανός κανείς να εκφράζει τις μετρήσεις σε όρους καταστάσεων του συστήματος ώστε να μπορούν να συγκριθούν οι καταστάσεις με τις μετρήσεις. Σε μια συγκεκριμένη σύνθεση δεν χρειάζεται να εμφανίζονται όλες οι μετρήσεις στο διάνυσμα εξόδου. Μερικές φορές κάποιες μετρήσεις είναι απαραίτητο να χρησιμοποιηθούν στο διάνυσμα καταστάσεων (εξίσωση 1). Το διάνυσμα εξόδου (2) πρέπει να απαρτίζεται από τιμές που μπορεί να καθοριστούν μαθηματικά από τις καταστάσεις καθώς και μέσω ενός ανεξάρτητου συστήματος μέτρησης.

Το διάνυσμα εισόδου \mathbf{u} έχει ποιο δύσκολο ορισμό. Αυτό το διάνυσμα περιέχει πληροφορίες που είναι απαραίτητες στον καθορισμό της δυναμικής του συστήματος. Οι τιμές του \mathbf{u} μπορεί να είναι μετρήσεις από αισθητήρες, παρόλα αυτά σε αυτή τη περίπτωση η αβεβαιότητα σε αυτές τις εισόδους πρέπει να ληφθεί υπόψη. Γενικά όταν καθορίζονται οι εξισώσεις του συστήματος (1) και (2) μετά τον καθορισμό των απαραίτητων καταστάσεων, όλοι οι άλλοι όροι που εμφανίζονται στο φίλτρο που δεν χρειάζεται να εκτιμηθούν ως καταστάσεις μπορούν να θεωρηθούν εισοδοί. Για παράδειγμα, για να βρεθεί η δυναμική μιας κατάστασης ταχύτητας, μπορεί να συμπεριληφθεί μια μέτρηση επιτάχυνσης ως είσοδος.

Οι όροι w και v που αντιστοιχούν στα διανύσματα θορύβου της διαδικασίας και της μέτρησης για το σύστημα είναι ενδιαφέροντες στο ότι τυπικά δεν εμφανίζονται άμεσα στις εξισώσεις ενδιαφέροντος. Αντ'αυτού αυτοί οι όροι χρησιμοποιούνται στην μοντελοποίηση της αβεβαιότητας ή του θορύβου στις εξισώσεις. Αυτό μπορεί να γίνει με διάφορους τρόπους. Ο πρώτος είναι η μοντελοποίηση του σφάλματος που είναι η αβεβαιότητα στην ίδια την εξίσωση. Πόσο μπορούμε να εμπιστευτούμε την εξίσωση; Μερικές εξισώσεις, όταν παράγονται από αρχές της φυσικής για παράδειγμα, έχουν αμελητέο σφάλμα. Παρόλα αυτά υπάρχουν εξισώσεις σε ορισμένες περιπτώσεις που είναι εριστικές και δεν μπορούν να χρησιμοποιηθούν με απόλυτη εμπιστοσύνη για την ορθότητά τους. Σε αυτές τις περιπτώσεις πρέπει να λάβουμε υπόψη μια ορισμένη ποσότητα σφάλματος και αυτό το σφάλμα έρχεται στη μορφή των w και v . Μια άλλη περίπτωση σφάλματος είναι το σφάλμα των αισθητήρων. Οι όροι w και v μπορούν να χρησιμοποιηθούν για να συμπεριλάβουμε στην εξίσωση τέτοιου τύπου σφάλματα. Στην πραγματικότητα αυτός είναι ο πιο κοινός τρόπος χρήσης του v στις εξισώσεις εξόδου. Πρέπει να προσέξουμε ότι το σφάλμα δεν είναι στην ίδια την εξίσωση, αλλά μετρά το πόσο καλή είναι η εξίσωση σχετικά με τη μέτρηση. Οι όροι w και v δεν περιέχονται στις εξισώσεις (1) και (2) μιας και υποθέτουμε ότι μετράνε τυχαία σφάλματα με μέση τιμή μηδέν, αλλά χρησιμοποιούνται για να εξαχθεί πληροφορία για τους πίνακες σύμβαλλητότητας Q και R του θορύβου της διαδικασίας και του θορύβου της μέτρησης. Στο παράρτημα αναφέρονται περισσότερες πληροφορίες για τον πίνακα σύμβαλλητότητας.

Τι γίνεται όμως με τους πίνακες του συστήματος F, G και H ; αυτοί οι πίνακες εξαρτώνται από το πρόβλημα που πρέπει να λυθεί και χρησιμοποιούνται ώστε να αντιπροσωπεύουν τις εξισώσεις ως γραμμικό σύστημα καταστάσεων και εισόδων. Ο F περιέχει τους συντελεστές των όρων της κατάστασης στη δυναμική της κατάστασης (1), ενώ ο H κάνει το ίδιο για τις εξισώσεις εξόδου (2). Ο πίνακας G περιέχει συντελεστές των όρων εισόδου στην δυναμική της κατάστασης (1). Αυτοί οι πίνακες μπορεί γενικά να μεταβάλλονται με το χρόνο αλλά δεν μπορούν να αλλάξουν σε σχέση με τις καταστάσεις ή τις εισόδους. Για πολλά προβλήματα αυτοί οι πίνακες είναι σταθεροί.

3.2.2. Ο αλγόριθμος του φίλτρου.

Το φίλτρο Kalman χρησιμοποιεί μια πρόβλεψη ακολουθούμενη από μια διόρθωση ώστε να καθορίσει τις καταστάσεις του φίλτρου. Αυτό ονομάζεται πρόβλεψη-επικαιροποίηση. Η βασική ιδέα είναι ότι χρησιμοποιώντας πληροφορία σχετική με τη δυναμική μιας κατάστασης το φίλτρο θα προβλέψει ποια θα είναι η επόμενη κατάσταση του συστήματος. Ένα απλό παράδειγμα θα αναφερθεί εδώ. Αν γνωρίζεις κανείς που ήταν (προηγούμενη κατάσταση) και πόσο γρήγορα κινείτο (δυναμική της κατάστασης) μπορεί να μαντέψει που είναι τώρα (τρέχουσα κατάσταση). Αυτό μπορεί να γίνει με αριθμητική ολοκλήρωση όπως η μέθοδος Euler ή η Runge Kutta. Το κομμάτι της διόρθωσης ή της επικαιροποίησης συμπεριλαμβάνει τη σύγκριση μιας μέτρησης με αυτό που προβλέπει το φίλτρο ότι θα έπρεπε να πάρουμε από τη μέτρηση βασισμένο στις καταστάσεις όπως έχουν προβλεφθεί. Τώρα θα συζητήσουμε για τις εξισώσεις του φίλτρου Kalman. Ξεκινώντας από μια αρχική κατάσταση x με πίνακα σύμβαλλητότητας P_0 εφαρμόζεται το σύστημα πρόβλεψης – διόρθωσης επαναλαμβανόμενα σε κάθε χρονικό βήμα χρησιμοποιώντας βρόγχο. Πρώτα προβλέπεται το διάνυσμα κατάστασης από την εξίσωση της δυναμικής της κατάστασης

$$\mathbf{x}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \quad (3)$$

όπου $\mathbf{x}_{k|k-1}$ είναι το προβλεφθέν διάνυσμα κατάστασης, \mathbf{x}_{k-1} είναι το διάνυσμα κατάστασης της προηγούμενης εκτίμησης, \mathbf{u} είναι το διάνυσμα των εισόδων και \mathbf{F} και \mathbf{G} είναι οι πίνακες που καθορίζουν τη δυναμική του συστήματος. Να σημειωθεί ότι ο δείκτης $k|k-1$ διαβάζεται ως k δεδομένου του $k-1$ και συμβολίζει την κατάσταση στο διακριτό χρόνο k δεδομένης της προηγούμενης κατάστασης στο χρόνο $k-1$. Έπειτα πρέπει να προβλεφτεί ο πίνακας συμμεταβλητότητας της κατάστασης

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \quad (4)$$

Όπου $\mathbf{P}_{k|k-1}$ παριστάνει τον προβλεφθέν πίνακα συμμεταβλητότητας της κατάστασης, \mathbf{P}_{k-1} είναι η προηγούμενη εκτίμηση για τον πίνακα και το \mathbf{Q} είναι ο πίνακας συμμεταβλητότητας του θορύβου. Παίρνοντας αυτές τις τιμές μπορεί πλέον να υπολογιστεί το κέρδος του Kalman.

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (5)$$

Όπου \mathbf{H} είναι ένας πίνακας απαραίτητος για τον καθορισμό της εξίσωσης εξόδου και \mathbf{R} είναι η συμμεταβλητότητα του θορύβου της μέτρησης. Το διάνυσμα κατάστασης ανανεώνεται κλιμακώνοντας την «καινοτομία», που είναι η διαφορά μεταξύ της μέτρησης της εξόδου και της προβλεπόμενης εξόδου, με το κέρδος Kalman όπως φαίνεται παρακάτω:

$$\mathbf{x}_k = \mathbf{x}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_{k|k-1}) \quad (6)$$

Κατά τον ίδιο τρόπο ανανεώνεται και ο πίνακας συμμεταβλητότητας του σφάλματος της κατάστασης

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}. \quad (7)$$

Όπου \mathbf{I} είναι ο μοναδιαίος πίνακας.

3.2.3. Σχόλια για το συμβολισμό.

Μια σημαντική διαφορά στο συμβολισμό που εμφανίζεται μεταξύ διάφορων πηγών είναι η χρήση του δείκτη k . Υπάρχουν τρεις διαφορετικές μορφές συμβολισμού που μπορούν να βρεθούν στη βιβλιογραφία σχετικά με τη χρήση του δείκτη k κάθε μια έχει ως αποτέλεσμα την διαφορετική γραφή των εξισώσεων 1 και 2. Που διαφέρουν στους δείκτες τους. Αυτοί οι συμβολισμοί συνοψίζονται στον παρακάτω πίνακα:

Time Format	State Dynamics (1)	Output Equations (2)
k and $k-1$ [1]	$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$	$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$
$k+1$ and k with output k [2,3]	$\mathbf{x}_{k+1} = \mathbf{F}_k\mathbf{x}_k + \mathbf{G}_k\mathbf{u}_k + \mathbf{w}_k$	$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$
$k+1$ and k with output $k+1$ [8]	$\mathbf{x}_{k+1} = \mathbf{F}_k\mathbf{x}_k + \mathbf{G}_k\mathbf{u}_k + \mathbf{w}_k$	$\mathbf{y}_{k+1} = \mathbf{H}_{k+1}\mathbf{x}_{k+1} + \mathbf{v}_{k+1}$

3-3 Διαφορές στον συμβολισμό από διαφορετικές πηγές για τις εξισώσεις (1) και (2) [2].

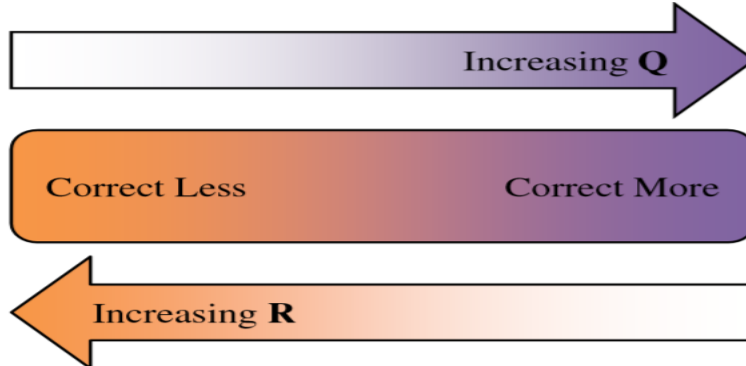
3.2.4. Φίλτρα Kalman συνεχούς και διακριτού χρόνου.

Σε αυτή την εργασία ασχολούμαστε με το φίλτρο Kalman διακριτού χρόνου. Υπάρχει και ένα φίλτρο συνεχούς χρόνου το οποίο συνήθως αναφέρεται ως φίλτρο Kalman-Bucy. Επειδή τα

περισσότερα φίλτρα Kalman υλοποιούνται σε ψηφιακούς υπολογιστές, η υλοποίηση του φίλτρου συνεχούς χρόνου δεν είναι πρακτική σε πολλές καταστάσεις. Αυτό δεν σημαίνει ότι μοντέλα συνεχούς χρόνου δεν μπορούν να προσεγγιστούν με το φίλτρο Kalman διακριτού χρόνου.

3.2.5. Οι επιδράσεις των υποθέσεων σχετικά με τους πίνακες συμμεταβλητότητας του θορύβου.

Η επιλογή των πινάκων συμμεταβλητότητας Q, R και P0 μπορεί να έχει σημαντική επιρροή στις επιδόσεις των προβλέψεων του φίλτρου Kalman. Η επιλογή του P0 ζευγαρώνεται με την αρχική κατάσταση του συστήματος και επιδρά στην αρχική σύγκλιση του φίλτρου. Σε πολλές περιπτώσεις η επίδραση του P0 δεν είναι σημαντική και συχνά αρχικοποιείται ως ένας μοναδιαίος πίνακας για απλότητα. Η επιρροή των πινάκων Q και R είναι πολύ πιο σημαντική και οι πίνακες αυτοί επηρεάζουν την συνολική επίδοση του φίλτρου. Ένας βασικός τρόπος για να δει κανείς τους πίνακες Q και R είναι ως παράγοντες βάρους μεταξύ των εξισώσεων της πρόβλεψης (κατάσταση συστήματος) και της μέτρησης (εξόδου). Αυτή η αναλογία φαίνεται στην εξίσωση για το κέρδος Kalman (εξίσωση 5). Μεγαλύτερα Q είναι συνώνυμα με μεγαλύτερη αβεβαιότητα στις εξισώσεις κατάστασης, που είναι συνώνυμο με το να έχουμε λιγότερη πίστη στα αποτελέσματα αυτών των εξισώσεων. Που σημαίνει ότι το φίλτρο πρέπει να τις διορθώσει περισσότερο με την ανανέωση της μέτρησης. Ομοίως μεγαλύτερο R είναι συνώνυμο με μεγαλύτερη αβεβαιότητα στην μέτρηση, που σημαίνει ότι δεν εμπιστευόμαστε πολύ τη μέτρηση που έχει ως αποτέλεσμα ότι το φίλτρο πρέπει να διορθώσει λιγότερο χρησιμοποιώντας την ανανέωση της μέτρησης. Ένα διάγραμμα που παριστάνει όσα αναφέρθηκαν εδώ φαίνεται στο παρακάτω σχήμα.



3-4 Η επίδραση των υποθέσεων για τους πίνακες συμμεταβλητότητας του θορύβου [2].

3.2.6. Το φίλτρο Kalman έχει γραμμικό μετασχηματισμό.

Ο σκοπός του πίνακα H είναι ουσιαστικά να μετατρέψει τις καταστάσεις σε εξόδους. Για το γραμμικό φίλτρο Kalman αυτό συμβαίνει λαμβάνοντας υπόψη κάποιο γραμμικό συνδυασμό των καταστάσεων, και έτσι ο H είναι γραμμικός μετασχηματισμός. Σε ορισμένες απλές περιπτώσεις ο πίνακας H χρησιμοποιείται για να επιλέξει συγκεκριμένες καταστάσεις που μετρούνται όταν κάποιες άλλες δεν μετρούνται. Για παράδειγμα αν η πρώτη η δεύτερη και η πέμπτη κατάσταση ενός 5 διαστάσεων διανύσματος κατάστασης είναι μετρήσιμες ο πίνακας H θα γραφόταν ως

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ωστε τελικά παίρνουμε

$$Y_k = H_k x_k$$

$$y_k = \begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix}$$

Ο πίνακας H μπορεί να χρησιμοποιηθεί επίσης για κλιμάκωση. Για παράδειγμα αν η ακτίνα ενός κύκλου είναι η κατάσταση, αλλά η διάμετρος είναι η ποσότητα που μετράται τότε η εξίσωση εξόδου μπορεί να παρασταθεί ως

$$Y_k = H_k x_k \text{ όπου}$$

$$Y_k = d_k, H_k = 2 \text{ και } x_k = r_k$$

Άλλη χρήση του H θα μπορούσε να ήταν ο συνδυασμός καταστάσεων, για παράδειγμα αν τα μήκη τριών πλευρών ενός τριγώνου είναι οι καταστάσεις, τότε η περίμετρος μπορεί να μετρηθεί και η εξίσωση εξόδου θα είναι

$$Y_k = x_1 + x_2 + x_3, H_k = [1 \ 1 \ 1], x_k = [x_1 \ x_2 \ x_3]^T$$

3.3 Ένα παράδειγμα γραμμικού φίλτρου Kalman.

Για να γίνει κατανοητή καλύτερα η χρήση του φίλτρου Kalman θα αναφερθεί ένα παράδειγμα. Το παράδειγμα ασχολείται με ένα αντικείμενο σε ελεύθερη πτώση, υποθέτοντας ότι δεν υπάρχει αντίσταση από τον αέρα. Ο σκοπός του φίλτρου είναι να καθορίσει τη θέση του αντικειμένου βασισμένο σε αβέβαιη πληροφορία για την αρχική θέση του και μετρήσεις της θέσης που παρέχονται από αισθητήρα laser (μέτρησης απόστασης). Χρησιμοποιώντας κινηματική σωματιδίων περιμένουμε ότι η επιτάχυνση του αντικειμένου θα είναι ίση με την επιτάχυνση της βαρύτητας. Ορίζεται το ύψος του αντικειμένου από το έδαφος σε μέτρα με την μεταβλητή h .

$$\ddot{h}(t) = -g \quad (18)$$

Όπου g είναι η επιτάχυνση λόγω βαρύτητας. Ολοκληρώνοντας αυτή τη σχέση για μικρό χρονικό διάστημα Δt έχουμε

$$\ddot{h}(t) = \frac{\dot{h}(t) - \dot{h}(t - \Delta t)}{\Delta t} = -g \quad (19)$$

Που είναι μια ανάποδη εξίσωση διαφορών, κάτι που είναι χρήσιμο για εφαρμογές του φίλτρου Kalman λόγω της αναδρομικής δομής του φίλτρου. Κάθε βήμα του φίλτρου αναφέρεται στο προηγούμενο βήμα. Απλοποιώντας την εξίσωση (19) έχουμε

$$\dot{h}(t) = \dot{h}(t - \Delta t) - g\Delta t \quad (20)$$

Ολοκληρώνοντας ξανά την εξίσωση (20) παίρνουμε μια κινηματική εξίσωση που χρησιμοποιείται συχνά και σχετίζει διαδοχικές θέσεις του σωματιδίου σε σχέση με το χρόνο για σταθερή επιτάχυνση.

$$h(t) = h(t - \Delta t) + \dot{h}(t - \Delta t)\Delta t - \frac{1}{2}g(\Delta t)^2 \quad (21)$$

Είναι χρήσιμο να λογιστούμε την παραπάνω εξίσωση όχι ως συνεχούς χρόνου αλλά με όρους διακριτού χρόνου με δείκτη k , όπου το k ορίζεται ως $t=k\Delta t$. Επιπλέον οι διακριτές τιμές χρόνου θα γράφονται παρακάτω ως δείκτες.

$$h(t) = h(k\Delta t) = h_k$$

$$h(t-\Delta t) = h(k\Delta t - \Delta t) = h(\Delta t(k-1)) = h_{k-1} \quad (22)$$

Εδώ θα ξαναγράψουμε τις κινηματικές εξισώσεις του προβλήματος με όρους διακριτού χρόνου

$$\dot{h}_k = \dot{h}_{k-1} - g\Delta t$$

$$h_k = h_{k-1} + \dot{h}_{k-1}\Delta t - \frac{1}{2}g\Delta t^2 \quad (23)$$

Από αυτές τις κινηματικές εξισώσεις βλέπουμε ότι έχουμε δύο εξισώσεις που περιγράφουν την κίνηση του αντικειμένου. Μια εξίσωση για τη θέση και μια για την ταχύτητα. Αφού μας ενδιαφέρει η εκτίμηση της θέσης, καταλαβαίνουμε ότι η θέση πρέπει να συμπεριληφθεί ως κατάσταση του συστήματος στο φίλτρο. Όμως βλέπουμε ότι η ταχύτητα συμπεριλαμβάνεται στην εξίσωση της θέσης επομένως πρέπει να πάρουμε και αυτή την πληροφορία. Ένας τρόπος για να είμαστε σίγουροι ότι έχουμε πληροφορία για την ταχύτητα είναι να συμπεριλάβουμε την ταχύτητα στην κατάσταση του συστήματος. Έτσι καθορίζουμε το διάνυσμα κατάστασης του φίλτρου ως

$$x_k = \begin{bmatrix} h_k \\ \dot{h}_k \end{bmatrix} \quad (24)$$

Αυτό μας οδηγεί στην παρακάτω έκφραση

$$x_k = \begin{bmatrix} h_{k-1} + \dot{h}_{k-1}\Delta t - \frac{1}{2}g\Delta t^2 \\ \dot{h}_{k-1} - g\Delta t \end{bmatrix} \quad (25)$$

Έχοντας αυτή τη μορφή μπορούμε να ξαναγράψουμε τις εξισώσεις με όρους του διανύσματος καταστάσεων πλέον ως εξής

$$x_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} -\frac{1}{2}(\Delta t)^2 \\ -\Delta t \end{bmatrix} g \quad (26)$$

Τώρα έχουμε το πρόβλημα στη κατάλληλη μορφή για να εφαρμόσουμε το φίλτρο Kalman.

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} \quad (27)$$

όπου οι πίνακες F και G είναι όπως παρακάτω

$$F_{k-1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, G_{k-1} = \begin{bmatrix} -\frac{1}{2}(\Delta t)^2 \\ -\Delta t \end{bmatrix} \quad (28)$$

Και το διάνυσμα εισόδου ορίζεται ως

$$\mathbf{u}_{k-1} = \mathbf{g} \quad (29)$$

ο λόγος που επιλέγουμε το \mathbf{g} ως είσοδο είναι επειδή αυτή η πληροφορία είναι απαραίτητη για να καθοριστεί η δυναμική της κατάστασης, αλλά το \mathbf{g} δεν ορίζεται ως κατάσταση στο φίλτρο. Αυτή η επιπλέον πληροφορία είναι είσοδος στις εξισώσεις σε κάθε χρονικό βήμα. Πρέπει να σημειωθεί ότι στο συγκεκριμένο πρόβλημα οι τιμές F, G και \mathbf{u} δεν μεταβάλλονται σε σχέση με το χρόνο (το k). Επιπλέον δεν υπάρχει ο όρος \mathbf{w} . Για αυτές τις συγκεκριμένες εξισώσεις υποθέτουμε ότι δεν υπάρχει σφάλμα στις ίδιες τις εξισώσεις. Αυτό είναι μια υπόθεση διότι στην πραγματικότητα μπορεί να υπάρχουν διαταραχές στην πορεία του σωματιδίου από τον αέρα ή άλλες πηγές. Όμως υποθέτοντας ότι αυτές οι διαταραχές είναι αμελητέες δεν είμαστε υποχρεωμένοι να μοντελοποιήσουμε θόρυβο στη διαδικασία λύσης αυτού του προβλήματος. Έτσι ο πίνακας συμμεταβλητότητας του θορύβου \mathbf{Q} τίθεται μηδέν.

Στο επόμενο βήμα θα ασχοληθούμε με το μέρος του συστήματος που είναι η μέτρηση. Ας υποθέσουμε το σενάριο κατά το οποίο η θέση του αντικειμένου μετράται με laser με σφάλμα που έχει τυπική απόκλιση 2m . Επειδή η θέση είναι αυτό που μετρείται, πρέπει να ορίσουμε μια εξίσωση εξόδου που δίνει τη θέση σε συνάρτηση με τις καταστάσεις του φίλτρου. Υπάρχει κάποια αβεβαιότητα στη μέτρηση που σημειώνεται στις εξισώσεις από το διάνυσμα θορύβου της μέτρησης \mathbf{v}_k .

$$y_k = h_k + v_k \quad (30)$$

Η θέση μπορεί να περιγραφεί με όρους του διανύσματος κατάστασης

$$y_k = [1 \ 0] \mathbf{x}_k + v_k \quad (31)$$

Τώρα έχουμε τις εξισώσεις εξόδου του συστήματος ορισμένες στην κατάλληλη μορφή όπως στην εξίσωση (2), όπου ο πίνακας H δίνεται από τη σχέση

$$H_k = [1 \ 0] \quad (32)$$

Το σύστημα της μέτρησης έχει τυπική απόκλιση σφάλματος 2m , που μια μεταβλητότητα 4m^2 . Εξαιτίας αυτού και του γεγονότος ότι στο διάνυσμα εξόδου υπάρχει μόνο ένας όρος, προκύπτει ένας πίνακας συμμεταβλητότητας σφάλματος που είναι ένας αριθμός $R=4\text{m}^2$.

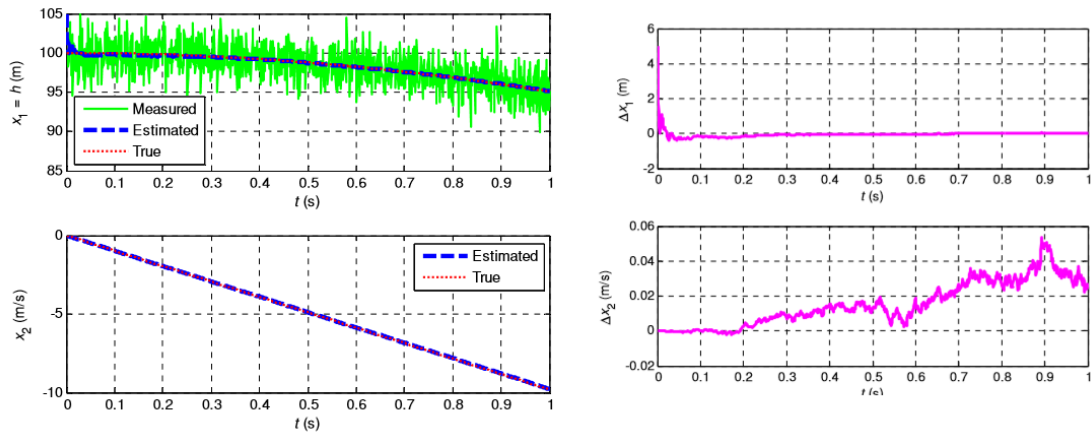
Επιπλέον του θορύβου της μέτρησης πρέπει να ληφθεί υπόψη και η αβεβαιότητα της αρχικής θέσης. Η αρχική θέση τίθεται περίπου 105m πριν αρχίσει το σώμα να πέφτει ενώ η πραγματική αρχική θέση είναι 105m . Η αρχική θέση δεν έχει εκτιμηθεί καλά και έτσι πρέπει να έχει σχετικά μεγάλη αρχική συμμεταβλητότητα. Έτσι υποθέτουμε ένα σφάλμα 10m^2 για την αρχική θέση. Για την αρχική ταχύτητα υποθέτουμε ότι το σώμα ξεκινά από στάση. Αυτή η υπόθεση είναι σχετικά σωστή και έτσι θέτουμε μικρότερη τιμή αβεβαιότητας $0.01\text{m}^2/\text{s}^2$. Όλα αυτά συνοψίζονται στον παρακάτω πίνακα.

Term	Definition
State Vector	$\mathbf{x}_k = \begin{bmatrix} h_k \\ \dot{h}_k \end{bmatrix}$
Output Vector	$\mathbf{y}_k = h_k + \mathbf{v}_k$
Input Vector	$\mathbf{u}_{k-1} = g$
System State Matrix	$\mathbf{F}_{k-1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$
Input Matrix	$\mathbf{G}_{k-1} = \begin{bmatrix} -\frac{1}{2}(\Delta t)^2 \\ -\Delta t \end{bmatrix}$
Observation Matrix	$\mathbf{H}_k = [1 \ 0]$
Process Noise Covariance Matrix	$\mathbf{Q}_{k-1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
Measurement Noise Covariance Matrix	$\mathbf{R}_k = 4$
True Initial State Vector	$\mathbf{x}_0 = \begin{bmatrix} 100 \\ 0 \end{bmatrix}$
Assumed Initial State Vector	$\hat{\mathbf{x}}_0 = \begin{bmatrix} 105 \\ 0 \end{bmatrix}$
Assumed Initial State Error Covariance Matrix	$\mathbf{P}_0 = \begin{bmatrix} 10 & 0 \\ 0 & 0.01 \end{bmatrix}$
Time Increment	$\Delta t = 0.001$

3-5 Συνοψισμός όρων για το παράδειγμα του φίλτρου Kalman [2].

Ολοκληρώνοντας το παράδειγμα αυτό θέλουμε να προσομοιώσουμε τιμές για τις αληθινές καταστάσεις του συστήματος καθώς και για τις μετρήσεις που περιέχουν θόρυβο. Σε πραγματικά προβλήματα αυτό το βήμα φυσικά δεν γίνεται μιας και οι αληθινές καταστάσεις δεν είναι γνωστές και οι μετρήσεις παρέχονται από το σύστημα μετρήσεων.

Το παραπάνω παράδειγμα εκτελέστηκε για 100 βήματα και οι υπολογισμένες καθώς και οι αληθινές καταστάσεις φαίνονται στο παρακάτω σχήμα 4-6 αριστερά. Τα λάθη των εκτιμήσεων φαίνονται στο σχήμα 3-6 δεξιά.



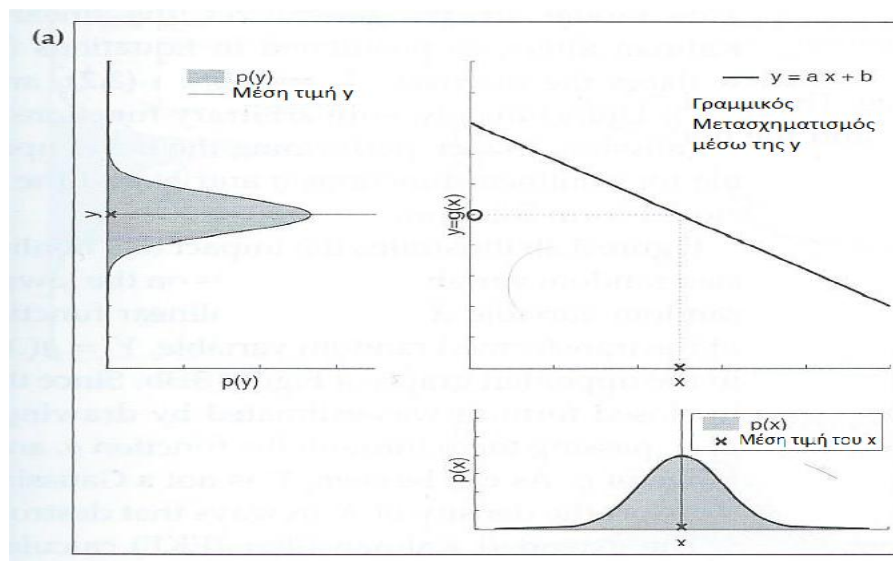
3-6 Παράδειγμα φίλτρου Kalman, αληθινές και υπολογισμένες καταστάσεις αριστερά, λάθη εκτιμήσεων δεξιά. [2]

3.4.Η επέκταση του φίλτρου Kalman, Extended Kalman Filter.

Το Extended Kalman Filter είναι η επέκταση του φίλτρου Kalman σε μη γραμμικά συστήματα.

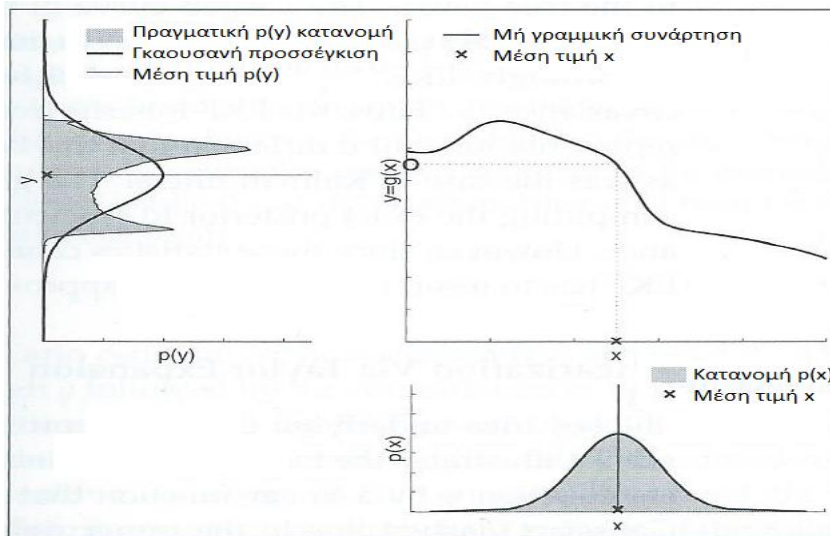
Οι υποθέσεις ότι οι παρατηρήσεις είναι γραμμικές εξισώσεις της κατάστασης και ότι η επόμενη κατάσταση είναι γραμμική εξίσωση της προηγούμενης κατάστασης είναι κρίσιμες για την ορθότητα του φίλτρου Kalman. Για την εξαγωγή του αλγόριθμου Kalman έπαιξε σημαντικό ρόλο η παρατήρηση ότι οποιοσδήποτε γραμμικός μετασχηματισμός μιας τυχαίας μεταβλητής με γκαουσιανή κατανομή έχει ως αποτέλεσμα μια νέα τυχαία μεταβλητή που έχει επίσης γκαουσιανή κατανομή. Η αποδοτικότητα του φίλτρου Kalman αποδίδεται στο γεγονός ότι οι παράμετροι της προκύπτουσας γκαουσιανής μπορούν να υπολογιστούν σε κλειστή μορφή.

Παρακάτω φαίνεται ένας γραμμικός μετασχηματισμός μιας τυχαίας μεταβλητής.



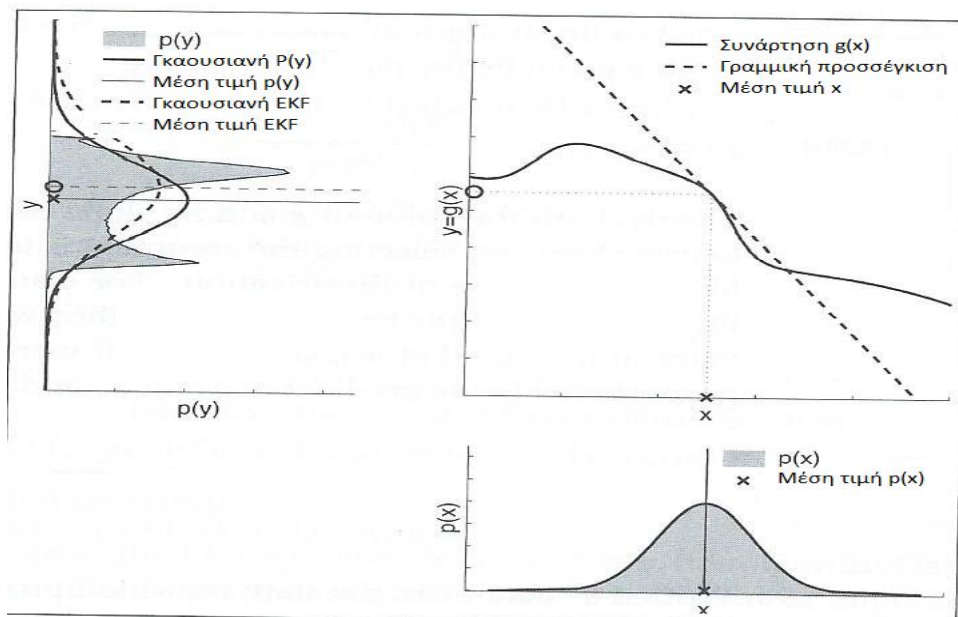
3-7 Γραμμικός μετασχηματισμός τυχαίας μεταβλητής. Παρατηρούμε ότι η γκαουσιανή κατανομή μετασχηματίζεται σε γκαουσιανή κατανομή. [16]

Αν ο μετασχηματισμός δεν είναι γραμμικός έχουμε μια κατάσταση όπως φαίνεται στο παρακάτω σχήμα.



3-8 Μη γραμμικός μετασχηματισμός τυχαίας μεταβλητής με γκαουσιανή κατανομή. [16]

Χρησιμοποιώντας το EKF είναι σαν να παίρνουμε μια γραμμική προσέγγιση της μη γραμμικής συνάρτησης μετασχηματισμού, όπως φαίνεται στο παρακάτω σχήμα.



3-9 Γραμμικοποίηση μέσω EKF. Αντί να περάσουμε τη τυχαία μεταβλητή x μέσω της μη γραμμικής συνάρτησης g χρησιμοποιούμε τη γραμμική προσέγγιση της συνάρτησης. Η αντίστοιχη μεταβλητή y έχει τώρα γκαουσιανή κατανομή (φαίνεται με τη διακεκομμένη γραμμή). [16]

Η διαφορά με το απλό φίλτρο Kalman είναι ότι οι εξισώσεις κατάστασης και/ή εξόδου μπορούν να περιέχουν μη γραμμικές συναρτήσεις. Έτσι αντί για συστήματα με εξισώσεις όπως οι (1) και (2) το Extended Kalman Filter έχει πιο γενικές μη γραμμικές εξισώσεις όπως οι παρακάτω

$$\mathbf{X}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \quad (33)$$

$$\mathbf{Y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (34)$$

Όπου το \mathbf{f} είναι ένα μη γραμμική διανυσματική εξίσωση μεταβατικής κατάστασης του συστήματος και το \mathbf{h} είναι μια συνάρτηση εξόδου ή παρατήρησης που έχει επίσης μορφή διανύσματος.

Για να χρησιμοποιηθούν οι μη γραμμικές εξισώσεις, υπολογίζονται οι ιακοβιανοί πίνακες ώστε να γραμμικοποιηθούν οι εξισώσεις σε κάθε χρονικό βήμα, χρησιμοποιώντας και το προηγούμενο βήμα (προηγούμενη κατάσταση του συστήματος) όπως παρακάτω

$$\mathbf{F}_k = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_{k-1}}$$

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}_{k|k-1}}$$

Χρησιμοποιώντας τις παραπάνω ιακοβιανές, οι εξισώσεις της επέκτασης του φίλτρου Kalman φαίνονται παρακάτω

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1})$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k [\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})]$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

3-10 Εξισώσεις του Extended Kalman Filter [2].

4. Το πρόβλημα και η μεθοδολογία που επιλέχθηκε για τη λύση του προβλήματος.

Το πρόβλημα το οποίο μελετάται στο πλαίσιο της παρούσας εργασίας είναι ο βέλτιστος εντοπισμός ενός οχήματος που κινείται στο επίπεδο και το οποίο μετρά την κίνησή του με κατάλληλους αισθητήρες οδομετρίας. Επίσης διαθέτει σύστημα καταγραφής αντικειμένων στον χώρο και ποιο συγκεκριμένα έναν αισθητήρα μέτρησης απόστασης με δέσμη laser.

Στη μελέτη του προβλήματος αυτού υπεισέρχονται τα σφάλματα των μετρήσεων τόσο του αισθητήρα laser όσο και των αισθητήρων οδομετρίας. Για τον λόγο αυτό και προκειμένου να έχουμε την βέλτιστη εκτίμηση της πόζας (θέσης και προσανατολισμού) χρησιμοποιήθηκαν στοχαστικά φίλτρα εκτίμησης όπως είναι το φίλτρο Kalman.

Το παραπάνω πρόβλημα, αν και έχει μελετηθεί επαρκώς τα τελευταία είκοσι χρόνια παρουσιάζει δυσκολία στην παρουσίαση των βημάτων επίλυσης με αποτέλεσμα να δυσχεραίνεται η διδασκαλία και η κατανόησή του. Στην εργασία αυτή χρησιμοποιήθηκαν υπολογιστικά εργαλεία προκειμένου να περιγράψουμε εποπτικά τα βασικά βήματα μέτρησης αποστάσεων στο χώρο και εντοπισμού του ρομπότ. Έτσι το ιδιαίτερο πρόβλημα με το οποίο ασχοληθήκαμε διατυπώνεται ως εξής.

4.1. Διατύπωση του προβλήματος.

Σχεδίαση και υλοποίηση υπολογιστικής εφαρμογής για την εποπτική εκπαίδευση – διδακτική προσομοίωση των παρακάτω βημάτων:

1. Ενός σαρωτή laser για τη λήψη μετρήσεων απόστασης γωνίας με βάση έναν εικονικό χάρτη.
2. Την εφαρμογή ενός αλγορίθμου για την εξαγωγή των μετρούμενων οροσήμων (landmarks) με βάση τις μετρήσεις του σαρωτή laser.
3. Την ενσωμάτωση της πληροφορίας της οδομετρίας με τη μορφή μιας αρχικής αβέβαιης εκτίμησης της θέσης και του προσανατολισμού του ρομπότ.
4. Την εφαρμογή ενός στοχαστικού φίλτρου, στην περίπτωσή μας της επέκτασης του φίλτρου Kalman για την βελτιστοποίηση της εκτίμησης της πόζας ενσωματώνοντας την πληροφορία από τα μετρούμενα ορόσημα

4.2. Μεθοδολογία επίλυσης

Για τη διερεύνηση των προβλημάτων που αναφέρθηκαν στην προηγούμενη παράγραφο χρησιμοποιήθηκε η εξής μεθοδολογία.

A. Προσομοίωση. Το πρόβλημα του εντοπισμού σε γνωστό χάρτη προσεγγίστηκε μέσω μιας υπολογιστικής προσομοίωσης που περιλαμβάνει την επίλυση των επιμέρους σημείων. Έτσι προσομοιάστηκε τόσο η εκτίμηση της θέσης από την οδομετρία με τη χρήση γκαουσιανών κατανομών όσο και οι μετρήσεις του laser σαρωτή. Σε αυτή την προσομοίωση λήφθηκαν υπόψη και σφάλματα που υπεισέρχονται στις μετρήσεις.

B. Ενσωμάτωση ευρέως διαδεδομένων αλγορίθμων όπως το EKF καθώς και διαδεδομένων τεχνικών μέτρησης με τη βοήθεια υπάρχουσας βιβλιογραφίας.

Για την προσομοίωση χρησιμοποιήσαμε το περιβάλλον προγραμματισμού LabView το οποίο ενσωματώνει οπτικά στοιχεία προγραμματισμού και πολλές έτοιμες μαθηματικές συναρτήσεις που είναι σχετικές με το πρόβλημα που αντιμετωπίζουμε.

Ο κώδικας του LabView είναι ευανάγνωστος και μπορεί εύκολα να χρησιμοποιηθεί ως εκπαιδευτικό υλικό όπως θα φανεί και παρακάτω. Χρησιμοποιήθηκαν έτοιμες μαθηματικές συναρτήσεις που μας παρέχει το περιβάλλον του LabView οι οποίες βοήθησαν στην εξαγωγή συμπερασμάτων από τον αλγόριθμο. Τέτοιες συναρτήσεις είναι η συνάρτηση που μας φτιάχνει ιστογράμματα, η συνάρτηση που μας δίνει το πίνακα συμμεταβλητότητας και συναρτήσεις που κάνουν πράξεις μεταξύ πινάκων όπως πολλαπλασιασμός πινάκων, εύρεση του ανάστροφου και του αντίστροφου πίνακα. Επίσης μεγάλη βοήθεια ήταν οι συναρτήσεις που μας παράγουν τις γραφικές παραστάσεις και μας επιτρέπουν να οπτικοποιήσουμε τη λύση που κατασκευάσαμε. Περισσότερα για το LabView θα αναφερθούν παρακάτω, στο κύριο μέρος της εργασίας καθώς και σε παράρτημα.

Σε σχέση με τη βιβλιογραφία και τους έτοιμους αλγόριθμους σημαντική βοήθεια λάβαμε από το βιβλίο Autonomous Mobile Robots και τις άλλες πηγές που αναφέρονται στο κεφάλαιο βιβλιογραφία.

Η προτεινόμενη εργασία δημιουργεί μια υπολογιστική υποδομή τόσο για την εποπτική επίδειξη της λειτουργίας του αισθητήρα laser όσο και για την διερεύνηση και αξιολόγηση διαφορετικών αλγορίθμων που σχετίζονται με τη συνολική διαδικασία (πχ εξαγωγή ορόσημων από τις μετρήσεις, χρήση διαφορετικών σχεσιακών φίλτρων κλπ).

Τέλος ένα βασικό πρόβλημα προς διερεύνηση στον εντοπισμό κινούμενων ρομπότ είναι τα όρια των σφαλμάτων που είναι επιτρεπτά, ώστε το στοχαστικό φίλτρο να συγκλίνει και να παράγει την βέλτιστη εκτίμηση. Τέτοια σφάλματα οφείλονται στους αισθητήρες οδομετρίας καθώς και στις μετρήσεις απόστασης-γωνίας που επιστρέφει ο αισθητήρας laser scanner. Επίσης μπορεί να οφείλονται στην ακρίβεια με την οποία σχεδιάστηκε και καταχωρήθηκε ο χάρτης του χώρου.

Η συγκεκριμένη διερεύνηση μπορεί να διευκολυνθεί μέσω του προτεινόμενου εργαλείου προσομοίωσης καθώς η τυπική απόκλιση των διαφόρων μετρήσεων δύναται να μεταβάλλεται εύκολα με υπολογιστικό τρόπο.

Όπως αναφέραμε και παραπάνω το πρόβλημα που έχουμε να λύσουμε σε αυτή την εργασία είναι ο εντοπισμός ενός αυτόνομου ρομπότ μέσα σε γνωστό χάρτη. Ουσιαστικά θέλουμε να κατασκευάσουμε έναν αλγόριθμο ο οποίος θα μας επιτρέψει να συμπτύξουμε πληροφορία από την οδομετρία με πληροφορία από ένα αισθητήρα laser που μετρά την απόσταση του ρομπότ από τους τοίχους του δωματίου. Η σύμπτυξη της πληροφορίας γίνεται με την επέκταση του φίλτρου Kalman, γιατί όπως θα δούμε οι εξισώσεις που περιγράφουν το πρόβλημα δεν είναι γραμμικές επομένως το απλό φίλτρο Kalman δεν επαρκεί.

Σε αυτή την εργασία μας ενδιαφέρει κυρίως να δούμε πως το φίλτρο EKF μας βοηθά να αποκτήσουμε καλύτερη αντίληψη της θέσης του ρομπότ. Καλούμαστε να απαντήσουμε σε ερωτήματα όπως, πόσο εύκολη είναι η υλοποίηση του φίλτρου Kalman σαν αλγόριθμο; Με πόσο μεγάλη ακρίβεια ανανεώνεται η θέση του ρομπότ; Πόσο μεγάλο θα μπορούσε να είναι

το σφάλμα των αισθητήρων προτού αρχίσει να επηρεάζει πρακτικά τον εντοπισμό του ρομπότ; Μπορεί αυτή η εφαρμογή να τρέχει σε πραγματικό χρόνο;

Για να απαντήσουμε στα παραπάνω ερωτήματα χρησιμοποιήσαμε την εξής μεθοδολογία. Καταρχήν χρησιμοποιήσαμε μόνο αισθητήρα laser ο οποίος περιστρέφεται κατά 360 μοίρες και μας δίνει πληροφορία απόστασης των αντικειμένων που ανακλούν τη δέσμη από το ρομπότ, καθώς και την γωνία στην οποία εντοπίζονται τα αντικείμενα. Επίσης χρησιμοποιούμε την πληροφορία σχετικά με την πόζα του ρομπότ από την οδομετρία. Το ρομπότ είναι τοποθετημένο σε ένα δωμάτιο με τέσσερις τοίχους και διαστάσεις όσες ενός κοινού δωματίου. Οι πληροφορίες του laser scanner προσομοιώνονται από το λογισμικό. Οι πληροφορίες από την οδομετρία επίσης προσομοιώνονται από το λογισμικό. Όπου χρειάζεται, για να είμαστε συμβατοί με πραγματικές εφαρμογές, εισάγεται σφάλμα στις μετρήσεις. Η σύμπτυξη της πληροφορίας γίνεται με το EKF.

Το λογισμικό που επιλέξαμε να χρησιμοποιήσουμε είναι το LabView 2011 που είναι μια γλώσσα προγραμματισμού με γραφικά αντικείμενα. Το επιλέξαμε για πολλούς λόγους. Καταρχήν ο προγραμματισμός στο LabView είναι αρκετά γρήγορος και αποδοτικός. Υπάρχουν πολλές έτοιμες βιβλιοθήκες σχετικές με την εργασία που θέλουμε να κάνουμε. Επίσης η οπτικοποίηση των αποτελεσμάτων μας σε διαγράμματα και γραφικές παραστάσεις είναι πολύ άμεση και εύκολη. Η μετάβαση από τον κώδικα του LabView σε κάποια άλλη γλώσσα προγραμματισμού όπως η python είναι αρκετά απλή. Ο κώδικας του LabView μπορεί να χρησιμοποιηθεί ως λογικό διάγραμμα για να γράψει κανείς κώδικα σε μια άλλη γλώσσα προγραμματισμού συμβατή με το hardware που θα φέρει το ρομπότ που χρησιμοποιεί. Μελετώντας τον κώδικα αυτής της εργασίας μπορεί κανείς να αποφασίσει τι συναρτήσεις πρέπει να υλοποιηθούν για τη λύση του προβλήματος και να κατασκευάσει αντίστοιχες βιβλιοθήκες. Χρησιμοποιώντας τον κώδικα αυτής της εργασίας (vi) μπορεί να τρέξει τον κώδικά του βήμα προς βήμα και να αποφανθεί αν συμφωνούν τα αποτελέσματα. Μια μικρή εισαγωγή στο LabView 2011 υπάρχει στα παραρτήματα αυτής της εργασίας. Αυτή η εισαγωγή έχει σκοπό να βοηθήσει τον αναγνώστη να καταλάβει γρήγορα τον κώδικα αυτής της εργασίας και να κάνει αλλαγές όπου κρίνει απαραίτητο για να προσαρμοστεί ο κώδικας στο δικό του ρομπότ.

Αποφασίσαμε σε αυτή την εργασία να χρησιμοποιήσουμε δεδομένα από δύο τύπους αισθητήρων. Από την οδομετρία και από ένα αισθητήρα laser. Σε ρομπότ με ρόδες, οι πληροφορίες της οδομετρίας είναι άμεσα διαθέσιμες, και ανάλογα με την διευθέτηση των τροχών του ρομπότ η συλλογή πληροφορίας για την πόζα του είναι σχετικά εύκολη. Αυτός είναι ο λόγος που χρησιμοποιήσαμε την πληροφορία από την οδομετρία. Σχετικά με τον αισθητήρα laser, από αυτά που αναφέραμε στο state of the art, καταλαβαίνουμε ότι η συλλογή πληροφορίας από τέτοιους αισθητήρες είναι πολύ ευκολότερη σε σχέση με άλλους όπως κάμερες, κυρίως στο υπολογιστικό κομμάτι. Αισθητήρες lidar είναι εύκολο να προμηθευτεί κανείς από την αγορά την ώρα που γράφεται αυτή η εργασία. Είναι λίγο ακριβοί αλλά έχουν αρκετή ακρίβεια για την εφαρμογή που θέλουμε. Επιπλέον η πληροφορία από τους δύο αυτούς αισθητήρες, (laser και οδομετρίας) είναι διαθέσιμη γρήγορα και οι υπολογισμοί που χρειάζονται για την εφαρμογή του φίλτρου Kalman όπως θα δούμε σε αυτή την εργασία μπορούν να γίνουν με αρκετά μεγάλη ταχύτητα, ώστε να επιτρέπεται στο ρομπότ να κινείται με σχετικά καλό ρυθμό στο περιβάλλον και να εντοπίζει τον εαυτό του σε αυτό.

Ο τύπος του χάρτη που επιλέξαμε να χρησιμοποιήσουμε είναι ένας πίνακας με 2500 στοιχεία x και y. Αυτά τα στοιχεία προκύπτουν από τέσσερις τυχαίες γραμμές που τέμνονται μεταξύ

τους σχηματίζοντας ένα δωμάτιο με τέσσερις τοίχους. Το πρόγραμμά μας παίρνει τις εξισώσεις των τοίχων και από αυτές μας δίνει ένα πίνακα με συντεταγμένες από σημεία των τοίχων. Τα σημεία δεν εκτείνονται πέρα από τα σημεία τομής των ευθειών που παριστάνουν τους τοίχους. Φυσικά θα μπορούσε κανείς να χρησιμοποιήσει ένα οποιοδήποτε χάρτη από εφαρμογές CAD. Αυτό που πρέπει να γίνει για να χρησιμοποιηθούν τέτοιοι χάρτες είναι ότι η κάθε γραμμή του χάρτη πρέπει να μετατραπεί σε ένα διάνυσμα σε πολικές συντεταγμένες που θα έχει ένα r και ένα θ που είναι η απόσταση του τοίχου από μια καθορισμένη αρχή και η γωνία που σχηματίζει ο τοίχος με μια καθορισμένη αρχική κατεύθυνση, ώστε κάθε τοίχος να είναι ευδιάκριτος. Ο πίνακας που κατασκευάσαμε παραμένει στη μνήμη του προγράμματος και χρησιμοποιείται για να προσομοιώσουμε τις ενδείξεις του laser scanner.

Το laser scanner προσομοιώνεται ως εξής. Με βήμα πέντε μοιρών (που μπορεί φυσικά να αλλάξει) κατασκευάζεται μια ευθεία αποτελούμενη από αρκετά σημεία (1000 σε αυτή την εφαρμογή, κάτι που μπορεί φυσικά να αλλάξει κατά βούληση) με μικρή απόσταση μεταξύ τους (0.005m σε αυτή την εφαρμογή). Έπειτα για κάθε σημείο της ευθείας υπολογίζεται η ευκλείδεια απόσταση από κάθε σημείο του τοίχου, και όταν αυτή η απόσταση είναι μικρότερη από μια συγκεκριμένη (0.03m) θεωρούμε ότι έχουμε μέτρηση από τον αισθητήρα. Όπου όταν λέμε μέτρηση, εννοούμε τα r και θ των ευθειών του laser σε πολικές συντεταγμένες.

Έχοντας τώρα τα r και θ από τις δέσμες laser μετατρέπουμε τις μετρήσεις μας σε αποστάσεις και γωνίες των τοίχων από την θέση του ρομπότ. Εδώ χρειάζεται να γίνουν κάποιες μετατροπές από το παγκόσμιο σύστημα συντεταγμένων που βρίσκεται ο χάρτης μας στο σύστημα συντεταγμένων του ρομπότ. Παρατηρούμε επίσης ότι στον κώδικά μας μεγάλο μέρος είναι αφιερωμένο στην προσομοίωση των μετρήσεων των αισθητήρων laser. Σε μια πραγματική εφαρμογή αυτή η προσομοίωση φυσικά δεν χρειάζεται μιας και οι μετρήσεις έρχονται από τον πραγματικό αισθητήρα. Αυτό θα μειώσει τον υπολογιστικό χρόνο του προγράμματος.

Για να προσομοιώσουμε την οδομετρία χρησιμοποιήσαμε τη εξής μεθοδολογία. Ο χρήστης εισάγει την πραγματική θέση του ρομπότ μέσα στο δωμάτιο. Έπειτα πάνω σε αυτή τη θέση προσθέτουμε ένα γκαουσιανό θόρυβο με συναρτήσεις του LabView. Επιλέχθηκε τυπική απόκλιση στις γκαουσιανές 1 μέτρο για τη θέση x και y στην οποία βρίσκεται το ρομπότ και 5 μοίρες τυπική απόκλιση για την γωνία στην οποία είναι στραμμένο το ρομπότ. Φυσικά αυτές οι τιμές μπορούν να αλλάξουν από τον χρήστη του προγράμματος. Εδώ πρέπει να προσέξουμε ώστε το ρομπότ να τοποθετηθεί μέσα στο δωμάτιο και όχι έξω από αυτό. Αν τοποθετηθεί έξω το πρόγραμμα δεν θα δώσει αξιόπιστα αποτελέσματα.

Από τα ρ και ϕ που είναι οι γωνίες του laser beam και οι αποστάσεις των εμποδίων που εμφανίζονται από το ρομπότ, εξάγουμε τις γωνίες και τις αποστάσεις των τοίχων στο frame του ρομπότ. Έπειτα αφού βάλουμε όλες τις παραπάνω μετρήσεις σε ένα πίνακα γωνιών και αποστάσεων παράγουμε δύο ιστογράμματα όπου οι κορυφές του κάθε ιστογράμματος είναι εκεί που εντοπίζονται οι τοίχοι από το ρομπότ. Το ιστόγραμμα παράγεται με βοήθεια έτοιμης συνάρτησης από το LabView.

Για κάθε peak στο ιστόγραμμα των γωνιών παίρνουμε όλες τις γωνίες που απέχουν ± 10 μοίρες από το peak και φτιάχνουμε ένα πίνακα για κάθε τοίχο με τις γωνίες και τις αντίστοιχες αποστάσεις. Οι τιμές των αποστάσεων εξάγονται από τις μετρήσεις των

αποστάσεων αφού πάρουμε τη μέση τιμή. Έτσι έχουμε μια μέση τιμή απόστασης για κάθε μια από τις τέσσερις τιμές των γωνιών.

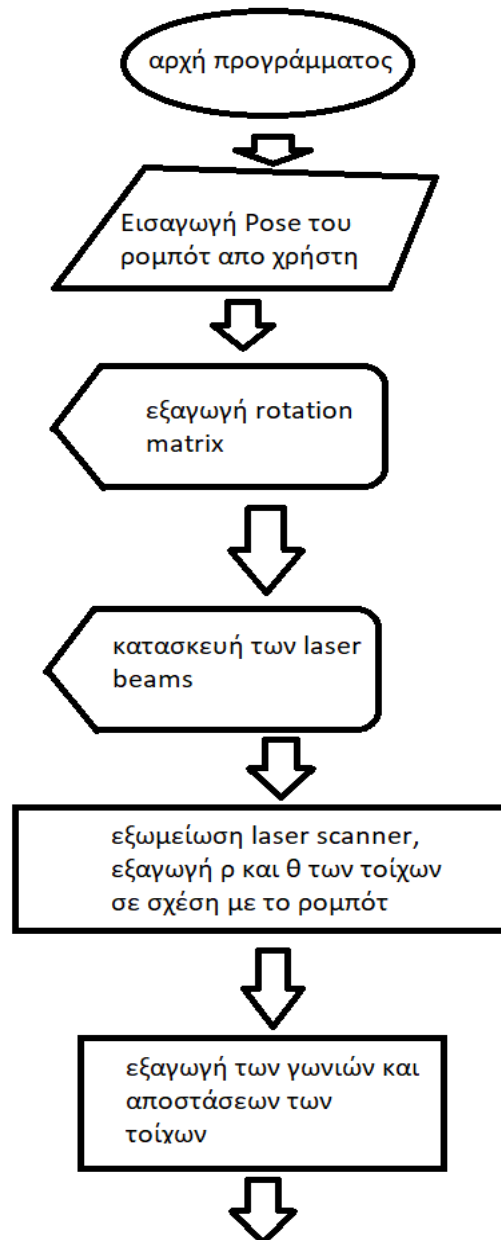
Από τον τύπο $w-\theta$ όπου w είναι η γωνία του τοίχου στο world frame και θ είναι η γωνία στην οποία κοιτάζει το ρομπότ από την οδομετρία παράγω τις predicted τιμές των γωνιών στις οποίες συναντούμε τους τοίχους. Από αντίστοιχο τύπο που θα αναφερθεί στο κύριο μέρος της εργασίας εξάγω και τις predicted αποστάσεις των τοίχων από το ρομπότ.

Έτσι έχω τις μετρούμενες τιμές γωνιών και αποστάσεων των τοίχων που τις παίρνω από το ιστόγραμμα. Έχω επίσης τις predicted τιμές που τις παίρνω από τους τύπους που αναφέρθηκαν στην προηγούμενη παράγραφο. Τέλος έχω τις τιμές γωνιών και αποστάσεων των τοίχων που παίρνω από τον χάρτη.

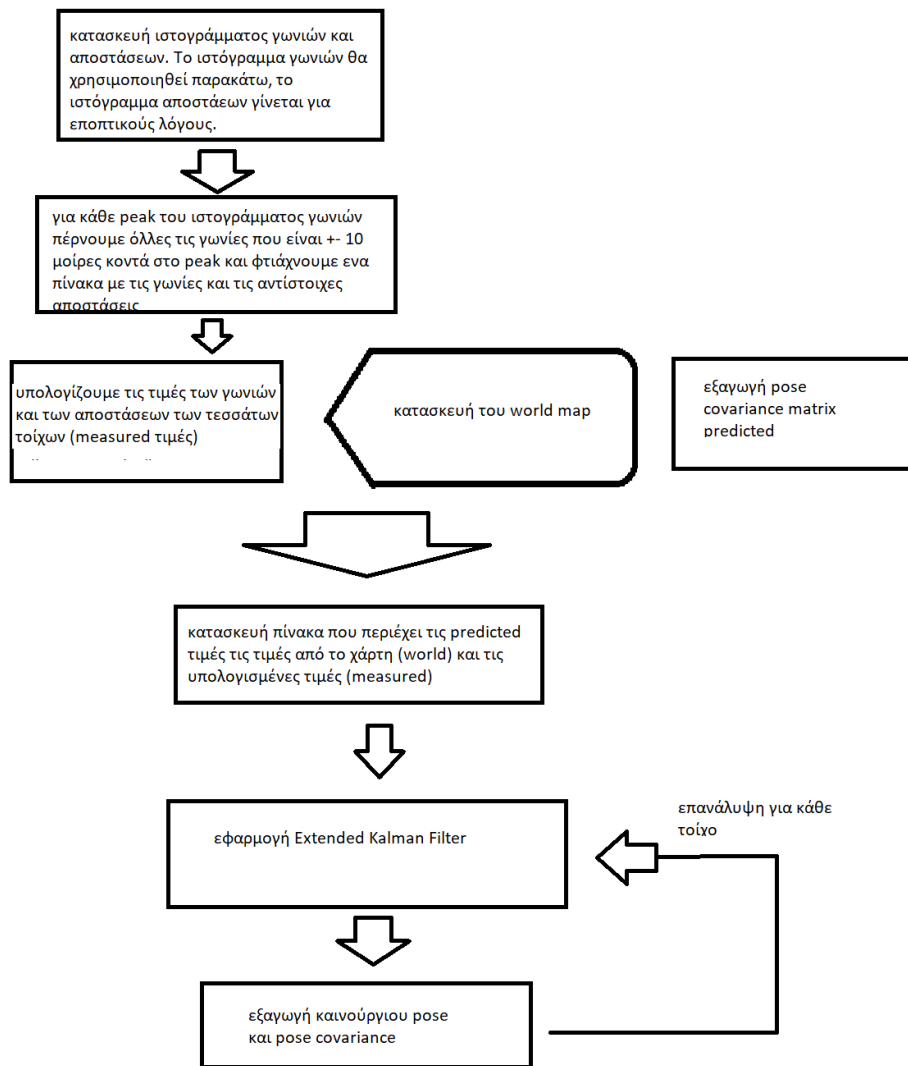
Χρησιμοποιώντας τις παραπάνω τιμές καθώς και τον pose covariance matrix από τις predicted τιμές, τον οποίο μας δίνει το LabView με έτοιμη συνάρτηση, εφαρμόζουμε το φίλτρο EKF. Αφού πάρουμε τα αποτελέσματα του EKF για τον πρώτο τοίχο, κατασκευάζουμε πάλι τις πληροφορίες που χρειαζόμαστε και ανανεώνουμε την εκτιμώμενη θέση του ρομπότ, εφαρμόζουμε το EKF διαδοχικά για τους υπόλοιπους τοίχους, ανανεώνοντας κάθε φορά τη θέση του ρομπότ. Όπως θα δούμε κάθε εφαρμογή του EKF μας δίνει μια εκτίμηση για τη θέση καλύτερη από την προηγούμενη και στο τέλος παίρνουμε την θέση του ρομπότ με αρκετά καλή ακρίβεια.

5. Κύριο μέρος εργασίας.

Σε αυτή την ενότητα θα παρουσιάσουμε τη λύση του προβλήματος, όπως το θέσαμε στην ενότητα 5, θα παρατεθεί ο κώδικας του LabView με επεξήγηση για κάθε βήμα του αλγόριθμου. Πριν γίνει αυτό όμως θα παραθέσουμε έναν λογικό αλγόριθμο που περιγράφει τη λύση όπως την προσεγγίσαμε εδώ.



5-1 Λογικός αλγόριθμος εργασίας, συνεχίζεται στην επόμενη σελίδα.



5-2 Λογικός αλγόριθμος εργασίας, δεύτερο μέρος.

5.1.Κατασκευή των τεσσάρων τοίχων (world map)

Για την κατασκευή των τεσσάρων τοίχων που αποτελούν το δωμάτιο εργαζόμαστε ως εξής. Χρησιμοποιούμε τον τύπο

$$y = ax + \beta \quad (1)$$

Από αυτό τον τύπο με κατάλληλα a και β κατασκευάζουμε τέσσερα ευθύγραμμα τμήματα που σταματάνε εκεί που τέμνονται μεταξύ τους. Για την πρώτη ευθεία παίρνουμε 1000 σημεία με βήμα στο x 0.004m. το βήμα στον x άξονα είναι το ίδιο και για τα τέσσερα ευθύγραμμα τμήματα. Έτσι παίρνουμε τέσσερεις πίνακες με ζεύγη x και y ο καθένας. Αυτοί οι πίνακες θα χρησιμοποιηθούν από το υπόλοιπο πρόγραμμα για την εύρεση των σημείων όπου ο laser scanner προσπίπτει πάνω στον τοίχο. Ο πρώτος τοίχος έχει 1000 σημεία, ο δεύτερος 250 ο τρίτος 1000 και ο τέταρτος 250.

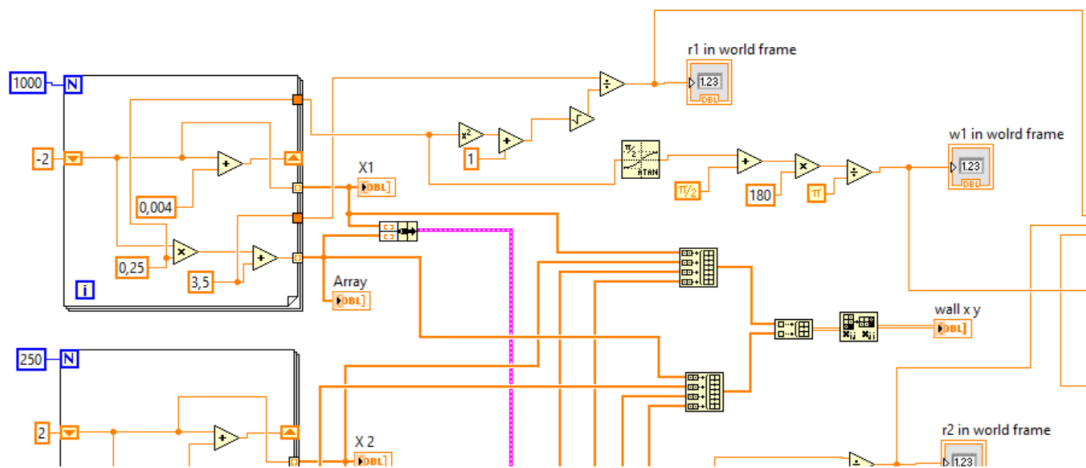
Επίσης χρειαζόμαστε τους τέσσερις τοίχους σε μορφή πολικών συντεταγμένων, δηλαδή να προσδιορίζονται ως ένα ζεύγος απόστασης r και γωνίας w στο σύστημα συντεταγμένων του χάρτη, το οποίο θα αναφέρουμε ως παγκόσμιο σύστημα συντεταγμένων (world frame). Για να πάρουμε τους τοίχους στη μορφή πολικών συντεταγμένων χρησιμοποιούμε τους παρακάτω τύπους

$$r = \frac{\beta}{\sqrt{\alpha^2 + 1}} \quad (2)$$

Όπου r είναι η απόσταση του τοίχου από την αρχή των αξόνων, και για τη γωνία

$$w = \left(\text{atan}(\alpha) + \frac{\pi}{2} \right) \frac{180}{\pi} \quad (3)$$

Και w είναι η γωνία που σχηματίζει η κάθετη από την αρχή των αξόνων μέχρι τον τοίχο από την αρχή μέτρησης των γωνιών στις πολικές συντεταγμένες. Όπου τα α και β ορίζονται όπως στην εξίσωση (1). Η μοβ γραμμή στον παρακάτω κώδικα είναι η ενσωμάτωση των x, y των τοίχων ώστε να μπορέσουμε να τα απεικονίσουμε σε γράφημα. Το τετράγωνο στα αριστερά του κώδικα LabView είναι ένα for loop, όπου το loop επαναλαμβάνεται 1000 φορές και η αρχική τιμή για το x είναι το -2 . Η κλίμακα είναι σε μέτρα, επομένως όλες οι αριθμητικές ποσότητες είναι σε μέτρα. Επιπλέον του υπολογισμού της γωνίας w , γίνεται και μετατροπή της σε μοίρες. Παρακάτω παρατίθεται ο κώδικας στο LabView.

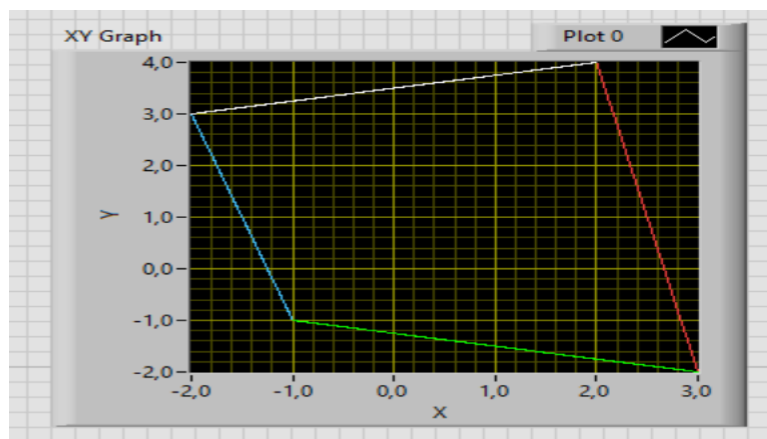


5-3 Κατασκευή ενός τοίχου με κώδικα στο LabView. Εξάγονται τα x, y του τοίχου και επίσης εξάγονται τα r, w του τοίχου σε πολικές συντεταγμένες. Διαφορετικοί τοίχοι έχουν διαφορετικό μήκος, και επομένως έχουν διαφορετικό αριθμό σημείων x, y . (1000 και 250) .

Η εξαγωγή των τοίχων γίνεται σε ξεχωριστό νι που αποτελεί sub-νι της κύριας εργασίας στο LabView. Σε αυτό το sub-νι γίνεται και απεικόνιση του map που κατασκευάσαμε. Αυτή φαίνεται στο παρακάτω σχήμα.

Εδώ πρέπει να αναφερθεί ότι δεν δίνουμε μεγάλη βαρύτητα στην επεξήγηση των προγραμματιστικών τεχνικών και συναρτήσεων που παρέχει το LabView και χρησιμοποιήθηκαν εδώ για την υλοποίηση του κώδικα. Επίσης ο ίδιος ο κώδικας δεν επεξηγείται αναλυτικά, παρά μόνο η βασική λογική που εφαρμόζεται. Αυτό γίνεται για να μην επιβαρυνθεί ο αναγνώστης με προγραμματιστικές λεπτομέρειες και να μην ξεφύγουμε από τον αρχικό στόχο που είναι η παρουσίαση του αλγορίθμου που λύνει το πρόβλημα που αντιμετωπίζουμε σε αυτή την εργασία. Εισαγωγικά στοιχεία και βοήθεια σχετικά με τις

συναρτήσεις που χρησιμοποιούνται μπορεί να πληροφορηθεί ο αναγνώστης από το αντίστοιχο παράρτημα στο τέλος της εργασίας.



5-4 Το δωμάτιο στο οποίο θα πλοηγηθεί το ρομπότ μας. Η απεικόνιση έγινε με κατάλληλες συναρτήσεις στο LabView 2011

5.2. Προσομοίωση laser beam

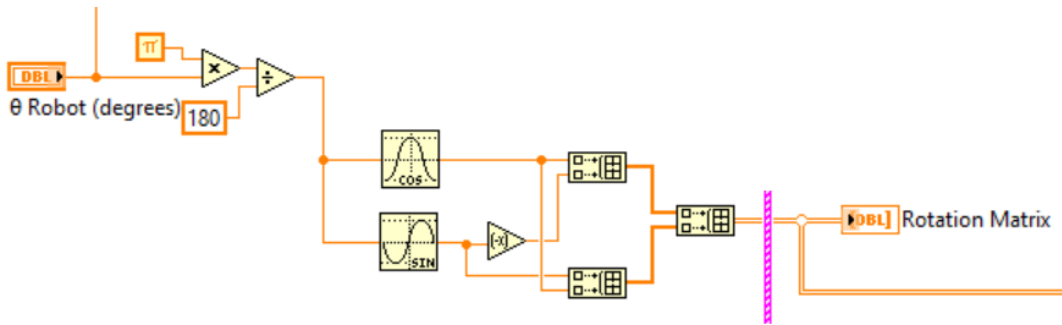
5.2.1. Εισαγωγή από τον χρήστη της πραγματικής θέσης του ρομπότ και εξαγωγή του rotation matrix.

Για να προσομοιώσουμε τη συμπεριφορά του αισθητήρα laser χρειάζεται να πάρουμε από το χρήστη τη πραγματική θέση του ρομπότ μέσα στο χάρτη και να εξάγουμε του rotation matrix. Το rotation matrix χρησιμοποιείται για να μεταφέρουμε τα laser beams στο robot frame (σύστημα συντεταγμένων του ρομπότ). Εδώ πρέπει να αναφερθεί ότι όταν λέμε πραγματική θέση του ρομπότ εννοούμε τη θέση στην οποία το ρομπότ βρίσκεται πραγματικά και όχι την θέση που νομίζει ότι βρίσκεται βασιζόμενο στις πληροφορίες από την οδομετρία. Χρειαζόμαστε αυτή τη πραγματική θέση διότι ο laser scanner μετρά τις αποστάσεις των εμποδίων από την πραγματική θέση του ρομπότ και όχι από αυτή στην οποία το ρομπότ νομίζει ότι βρίσκεται.

Σε αυτό το σημείο του κώδικα ο χρήστης εισάγει την πραγματική θέση και τον προσανατολισμό (pose) του ρομπότ μέσα στον χάρτη. Εδώ χρειάζεται προσοχή ώστε το ρομπότ να τοποθετηθεί μέσα στο δωμάτιο που απαρτίζεται από τους τέσσερις τοίχους και όχι έξω από αυτό. Αν το ρομπότ τοποθετηθεί έξω από το δωμάτιο, τα αποτελέσματα του προγράμματος δεν είναι αξιόπιστα. Επίσης εξάγεται το rotation matrix από την πόζα του ρομπότ, το οποίο θα μας χρειαστεί παρακάτω. Ο τύπος που μας δίνει το rotation matrix δίνεται παρακάτω.

$$rotation\ matrix = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (4)$$

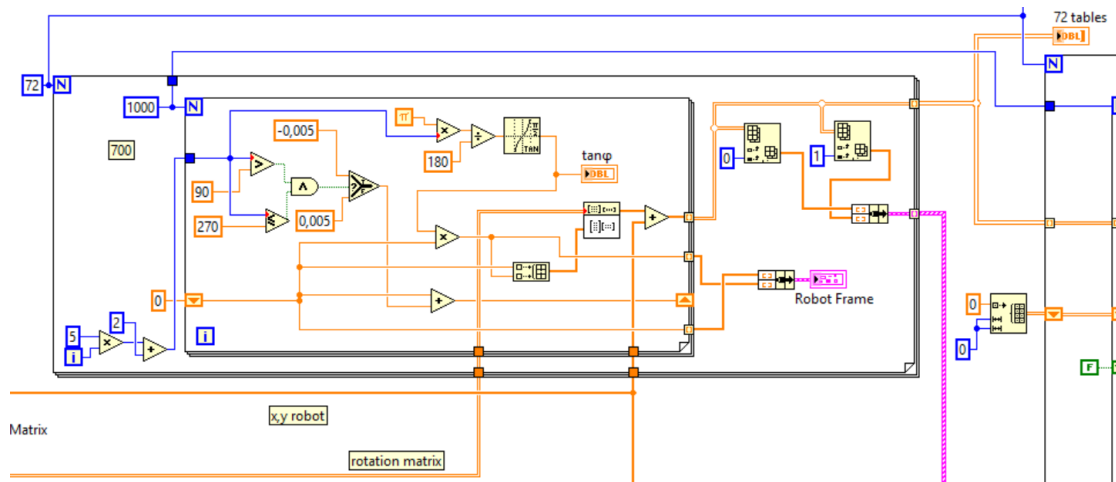
Παρακάτω παρατίθεται ο κώδικα LabView που κάνει υλοποιεί τα παραπάνω.



5-5 Εξαγωγή rotation matrix από το pose του ρομπότ το οποίο εισάγει ο χρήστης.

5.2.2. Κατασκευή του laser beam

Το επόμενο βήμα στον αλγόριθμο είναι η κατασκευή του Laser beam. Για να γίνει αυτό χρησιμοποιείται ένα for loop. Με βήμα 5 μοιρών παίρνουμε 72 δέσμες Laser ($72 \cdot 5 = 360$) με την αρχική δέσμη να ξεκινά από τις 7 μοίρες ώστε να καλύψουμε 360 μοίρες, όπως θα έκανε ένας σαρωτής Laser τοποθετημένος στο ρομπότ μας. Για κάθε τέτοια δέσμη παίρνουμε 1000 σημεία της με αρχή την αρχική pose του ρομπότ. Τα σημεία αυτά απέχουν 5 χιλιοστά το ένα από το άλλο κατά τον άξονα x. Έτσι εξάγουμε 72 πίνακες σημείων που αντιπροσωπεύουν τις ακτίνες του laser και θα χρησιμοποιηθούν παρακάτω στην προσομοίωση του laser scanner. Ο κώδικας στο labView φαίνεται παρακάτω.



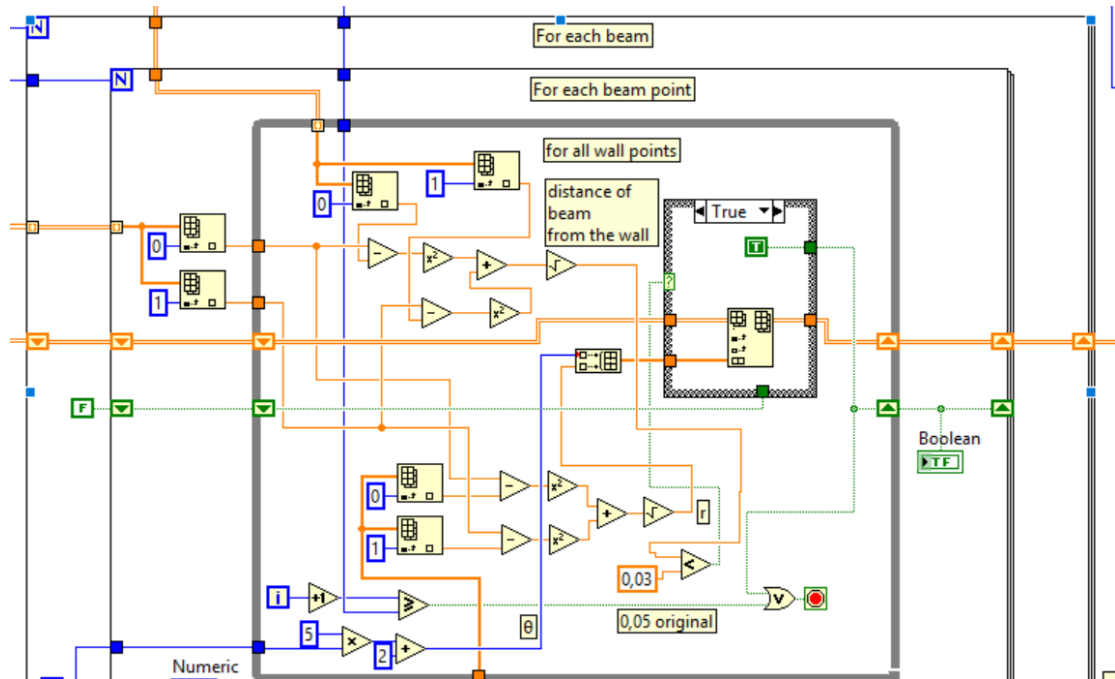
5-6 Εξαγωγή 72 laser beams ως 72 πίνακες 1000 σημείων, με αντιστοιχία στο pose του ρομπότ.

Η έξοδος αυτού του κώδικα είναι 72 πίνακες των 1000 σημείων και αντιπροσωπεύει τις 72 ακτίνες του laser καθώς αυτό σαρώνει το χώρο με βήμα 5 μοίρες.

5.2.3. Προσομοίωση μετρήσεων του Laser scanner.

Χρησιμοποιώντας ενσωματωμένα for και while loop προσομοιώνουμε τον εξής λογικό αλγόριθμο. Για κάθε ακτίνα laser, που παίρνουμε από τη μέθοδο που περιγράφηκε στην παραπάνω παράγραφο, για κάθε σημείο της ακτίνας, για όλα τα σημεία του τοίχου, υπολογίζεται η ευκλείδεια απόσταση του σημείου της ακτίνας από το σημείο του τοίχου. Αν αυτή η απόσταση είναι μικρότερη των 3 εκατοστών κρατάμε σε ένα πίνακα που είναι η

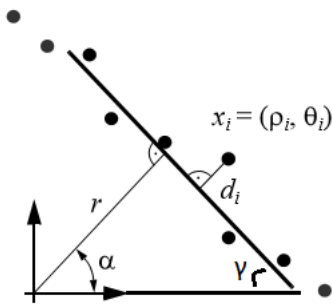
τελική έξοδος του αλγόριθμου την απόσταση του σημείου από το ρομπότ καθώς και την γωνία της δέσμης Laser στην οποία ανήκει. Έτσι τελικά παίρνουμε έναν πίνακα με γωνίες και αποστάσεις που είναι ουσιαστικά οι μετρήσεις του αισθητήρα Laser εκεί που βρήκε εμπόδιο η ακτίνα laser. Ο κώδικας σε LabView παρατίθεται παρακάτω.



5-7 Προσομοίωση μέτρησης του laser scanner. Μια σάρωση 360 μοίρες.

5.3. Εκτίμηση των r και α που καθορίζουν τους τοίχους σε πολικές συντεταγμένες, από τις μετρήσεις του laser scanner.

Αυτό που θέλουμε να πετύχουμε σε αυτό το τμήμα του αλγόριθμου είναι το εξής. Από τις μετρήσεις γωνίας και απόστασης του Laser scanner θέλουμε να εξάγουμε τους τέσσερις τοίχους, σε μορφή πολικών συντεταγμένων, όπου περιγράφονται με τα r , α . Όπου r είναι η απόσταση του τοίχου από το ρομπότ και α η γωνία του τοίχου από τον προσανατολισμό του ρομπότ. Αυτό φαίνεται και στο παρακάτω σχήμα. Οι τελείες στο παρακάτω σχήμα είναι τα δεδομένα που έχουμε από τον πίνακα που ήταν η έξοδος στο προηγούμενο βήμα του κώδικα. Δηλαδή οι μετρήσεις του laser scanner.



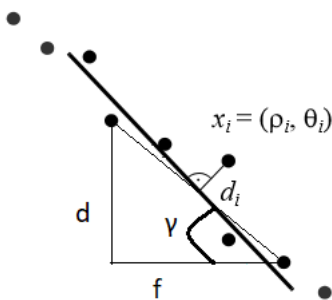
5-8 Εξαγωγή των r, α των τοίχων.

Όπου ρ_i και θ_i είναι οι μετρήσεις του Laser scanner και r, α είναι οι πολικές συντεταγμένες του τοίχου στο σύστημα του ρομπότ. Τέλος γ είναι η γωνία που σχηματίζει ο τοίχος με τον άξονα x .

Για να εξάγουμε τα r, α εργαζόμαστε ως εξής. Παίρνουμε δύο σημεία που μας δίνει ο Laser scanner στα οποία η δέσμη συνάντησε τον τοίχο. Παρατηρήστε εδώ ότι αμέσως εισάγεται ένα σφάλμα στις μετρήσεις του Laser scanner, και έτσι τα σημεία που μας δίνει δεν είναι ακριβώς πάνω στην ευθεία του τοίχου. Η γωνία γ που σχηματίζει ο τοίχος με τον άξονα x έχει ως εφαπτομένη την

$$\tan(\gamma) = \frac{d}{f} = -\frac{\rho_i \sin(\theta_i) - \rho_{i+k} \sin(\theta_{i+k})}{\rho_i \cos(\theta_i) - \rho_{i+k} \cos(\theta_{i+k})} \quad (5)$$

Όπου τα σύμβολα εξηγούνται στο παρακάτω σχήμα.



5-9 Εξαγωγή της εφαπτομένης της γωνίας γ .

Παρατηρείται ότι όσο πιο μακριά είναι τα δύο σημεία i και $i+k$ τόσο η ευθεία που σχηματίζουν προσεγγίζει την ευθεία του τοίχου.

Γνωρίζοντας τώρα τη γωνία γ μπορούμε να εξάγουμε την γωνία α από τον τύπο

$$\alpha = 90 - \gamma \quad (6)$$

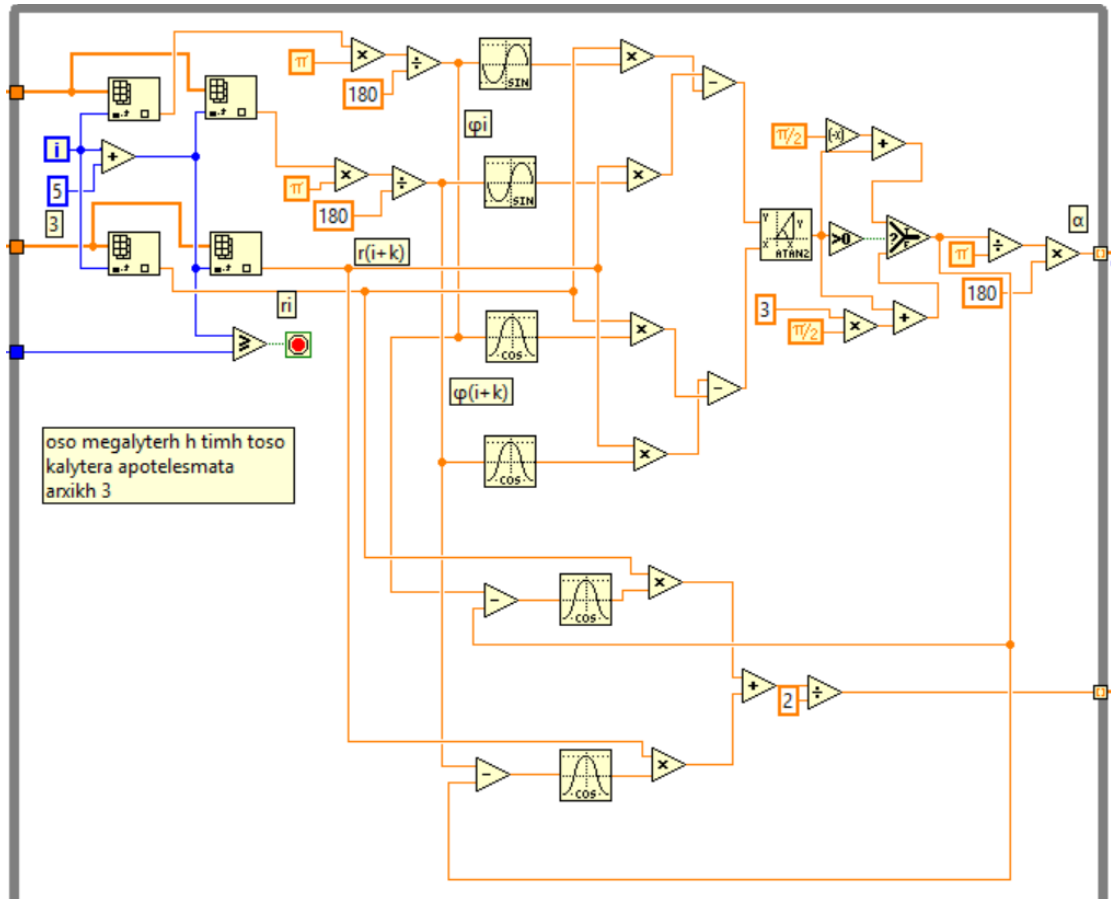
Αυτό εξάγεται εύκολα από το σχήμα 6.8. Εδώ βλέπουμε ότι εισάγεται ένα ακόμη σφάλμα στους υπολογισμούς μας, μιας και στο τρίγωνο του σχήματος 6.8 η μια πλευρά του τριγώνου

είναι ο τοίχος ενώ η αντίστοιχη πλευρά του τριγώνου του σχήματος 6.9 δεν είναι ακριβώς ο τοίχος.

Για την εξαγωγή του r χρησιμοποιούμε τον παρακάτω τύπο:

$$r = \frac{\rho_i \cos(\theta_i - a) + \rho_{i+k} \cos(\theta_{i+k} - a)}{2} \quad (7)$$

Όπου a η γωνία που σχηματίζει η κάθετος του τοίχου με τον άξονα x όπως φαίνεται στο σχήμα 6-8 και δίνεται από τον τύπο (6). Ο κώδικας LabView που υλοποιεί τα παραπάνω φαίνεται στο σχήμα.



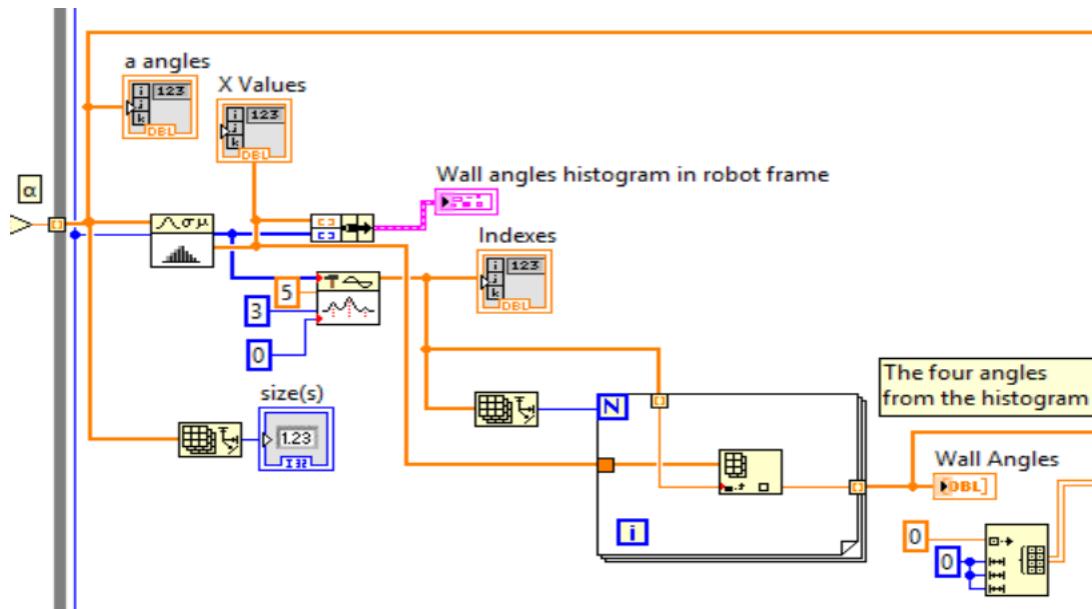
5-10 Εξαγωγή των r και a των τοίχων. Χρησιμοποιείται while loop, η πάνω έξοδος είναι το a και η κάτω το r (στα δεξιά).

Τα εξαγόμενα r και τα αντίστοιχα a τοποθετούνται σε δύο πίνακες μιας διάστασης, όπου στο στοιχείο μηδέν του πίνακα r είναι η απόσταση και στο ίδιο στοιχείο του πίνακα a είναι αποθηκευμένη η αντίστοιχη γωνία.

5.4. Κατασκευή ιστογράμματος γωνιών (και αποστάσεων) των τοίχων.

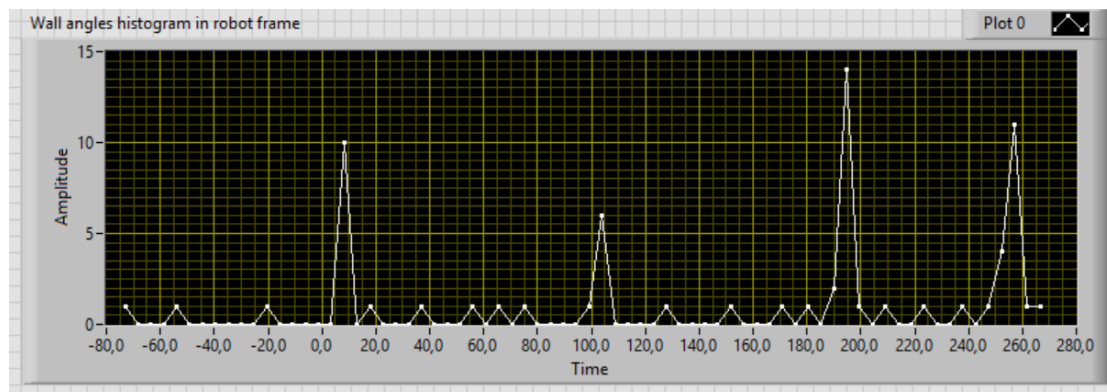
Σε αυτό το βήμα του αλγόριθμου μας ενδιαφέρει να εξάγουμε ένα ιστόγραμμα των γωνιών των τεσσάρων τοίχων, όπως υπολογίστηκαν στην παράγραφο 6.3. Χρησιμοποιώντας τον

πίνακα των γωνιών α που είναι η έξοδος του προηγούμενου κώδικα, στέλνουμε τον πίνακα αυτό ως είσοδο στη συνάρτηση Histogram του LabView. Αυτή η συνάρτηση μας δίνει ως έξοδο το $h(x)$ που είναι οι κορυφές του ιστογράμματος των γωνιών. Μας δίνει επίσης ως έξοδο το x που είναι οι γωνίες στις οποίες έχουμε μετρήσεις από το laser scanner. Χρησιμοποιώντας τις εξόδους αυτής της συνάρτησης κατασκευάζουμε ένα ιστόγραμμα. Σε αυτό περιμένουμε να δούμε τέσσερις κορυφές που πρέπει να είναι οι γωνίες α των τεσσάρων τοίχων του δωματίου. Ο κώδικας σε LabView για αυτό το βήμα φαίνεται παρακάτω.



5-11 Κατασκευή ιστογράμματος των γωνιών α . Στον ίδιο κώδικα γίνεται και φιλτράρισμα των τεσσάρων γωνιών. Η έξοδος του φιλτραρίσματος είναι ένας πίνακας με τέσσερις γωνίες (Wall Angles).

Ένα παράδειγμα του ιστογράμματος που κατασκευάζεται με τον παραπάνω κώδικα φαίνεται στο παρακάτω σχήμα. Εδώ το ρομπότ τοποθετήθηκε από τον χρήστη στο (0,0) στον χώρο.

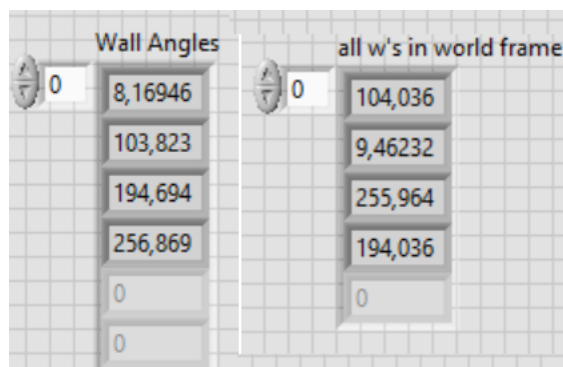


5-12 Το ιστόγραμμα των γωνιών α των τεσσάρων τοίχων. Ο άξονας x είναι η γωνίες και ο άξονας y είναι ο αριθμός των παρατηρήσεων της συγκεκριμένης γωνίας από τον Laser scanner. Όσο περισσότερες φορές βλέπουμε μια γωνία τόσο πιο πιθανό είναι να υπάρχει τοίχος στην συγκεκριμένη γωνία.

Παρατηρήστε ότι τέσσερις από τις γωνίες στις οποίες έχουμε μέτρηση από τον laser scanner έχουν μετρηθεί πολλές φορές (πάνω από πέντε φορές). Σε κάθε μια από αυτές τις γωνίες

περιμένουμε να έχουμε τοίχο. Οι υπόλοιπες γωνίες που εμφανίζονται στο ιστόγραμμα είναι σφάλματα του laser scanner.

Έχοντας εξάγει το παραπάνω ιστόγραμμα με τη βοήθεια της συνάρτησης Histogram, θέλουμε να πάρουμε τώρα ένα πίνακα με τέσσερις γωνίες που θα είναι οι γωνίες α των τοίχων. Αυτό το καταφέρνουμε με τη συνάρτηση peak detector του LabView. Ο κώδικας φαίνεται στο σχήμα 6.11 παραπάνω. Η τελική έξοδος είναι ένας πίνακας μιας διάστασης που περιέχει τέσσερις τιμές και ονομάζεται Wall Angles. Για το προηγούμενο παράδειγμα όπου το ρομπότ έχει τοποθετηθεί στο (0,0) ο πίνακας Wall Angles φαίνεται στο παρακάτω σχήμα.

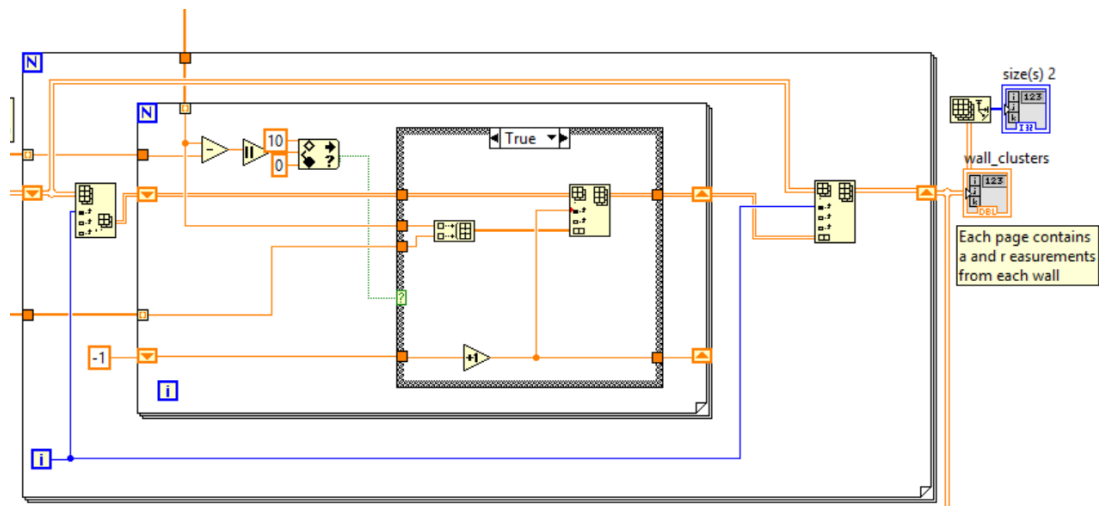


5-13 Ο πίνακας όπου έχουμε εξάγει τις τέσσερις γωνίες α των τοίχων από τις μετρήσεις του laser scanner (αριστερά). Στα δεξιά φαίνονται οι πραγματικές γωνίες α των τοίχων. Παρατηρήστε ότι οι γωνίες που υπολογίσαμε είναι αρκετά κοντά στις πραγματικές. Επίσης οι δύο πίνακες δεν έχουν αντιστοιχηθεί, δηλαδή το στοιχείο μηδέν του πρώτου πίνακα δεν αναφέρεται στον ίδιο τοίχο με το στοιχείο μηδέν του δεύτερου πίνακα.

5.5. Κατασκευή των Wall Clusters και υπολογισμός measurement Covariance και μέσω των τιμών α, r των τοίχων.

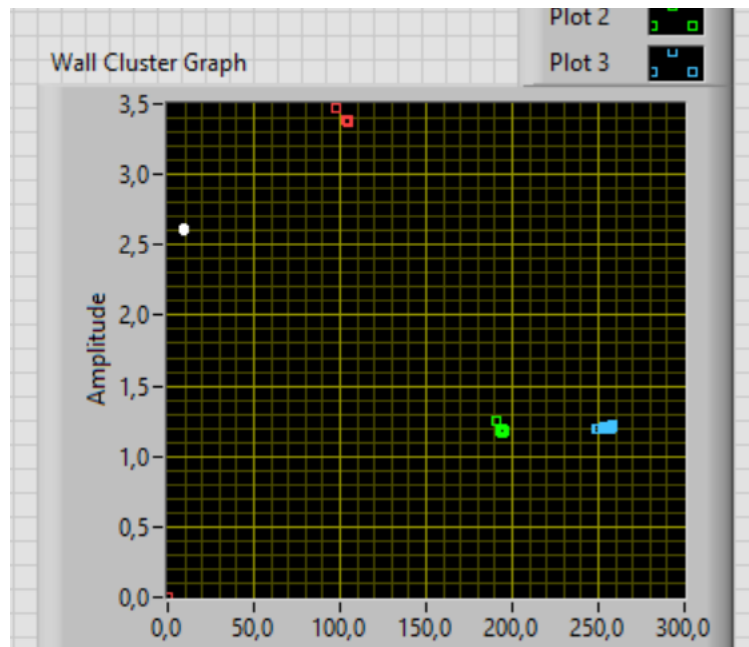
Το επόμενο βήμα στον αλγόριθμο που κατασκευάζουμε είναι η εξαγωγή των wall clusters. Τα wall clusters δεν είναι τίποτε άλλο από τις τιμές των α και r του κάθε τοίχου ομαδοποιημένες σε ένα γράφημα. Μας χρειάζεται για να παρατηρήσουμε τη διασπορά των σημείων του κάθε τοίχου και να δούμε αν υπάρχουν λογικά σφάλματα στην εκτίμηση των χαρακτηριστικών των τοίχων. Επίσης ο πίνακας που περιέχει τα wall clusters θα μας χρειαστεί παρακάτω στον αλγόριθμο.

Ο αλγόριθμος για την κατασκευή των wall cluster δουλεύει ως εξής. Για κάθε κορυφή του ιστογράμματος των γωνιών που έχουμε, παίρνουμε όλες τις γωνίες που είναι ± 10 μοίρες κοντά στο peak και φτιάχνουμε ένα πίνακα με τις γωνίες και τις αντίστοιχες αποστάσεις. (Δουλεύουμε στο robot frame). Ο κώδικας σε LabView φαίνεται παρακάτω.



5-14 Εξαγωγή των wall clusters. Ο πίνακας στην έξοδο έχει τέσσερις υπό πίνακες με a και r έναν για κάθε τοίχο.

Τον παραπάνω πίνακα μπορούμε να τον οπτικοποιήσουμε σε ένα γράφημα. Ένα τέτοιο γράφημα φαίνεται στο παρακάτω σχήμα. Εδώ πρέπει να αναφέρουμε ότι στο γράφημα φαίνονται εκτός των τοίχων και κάποια σημεία στο $(0,0)$. Αυτά τα σημεία εισάγονται λόγω λογικού λάθους στον κώδικα LabView. Τα σημεία αυτά δεν επηρεάζουν τους περαιτέρω υπολογισμούς μας και ο χρήστης πρέπει να τα παραβλέψει. Στο παράδειγμα που ακολουθεί το ρομπότ βρίσκεται στο $(0,0)$.

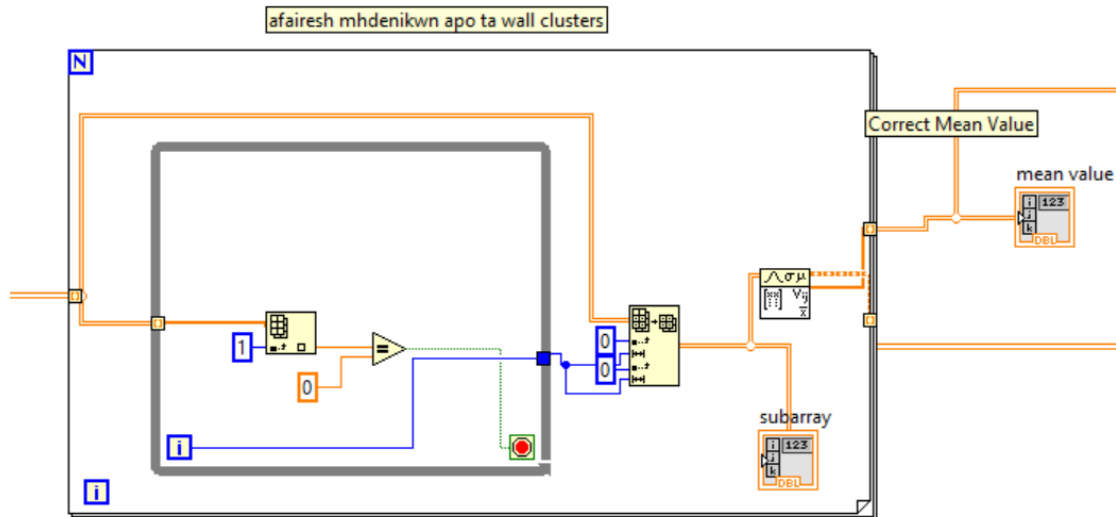


5-15 Τα τέσσερα wall cluster. Ο άξονας x είναι σε μίρες και ο άξονας y σε μέτρα. Οι μετρήσεις που φαίνονται στο $(0,0)$ πρέπει να αγνοηθούν.

Πέρα από την οπτικοποίηση των αποτελεσμάτων μας, ο πίνακας με τα wall clusters θα μας χρειαστεί παρακάτω. Όπως καταλαβαίνουμε από τα παραπάνω ο πίνακας wall clusters περιέχει τις τιμές των χαρακτηριστικών των τοίχων όπως μετρήθηκαν από τον laser scanner.

Αυτές τις τιμές πρέπει να τις επεξεργαστούμε λίγο πριν τις χρησιμοποιήσουμε παρακάτω στον αλγόριθμο.

Αυτό που θέλουμε να εξάγουμε από τα wall clusters είναι δύο πράγματα. Το ένα είναι μια μέση τιμή για την απόσταση και τη γωνία για κάθε τοίχο, και το δεύτερο είναι η συμμεταβλητότητα των μετρήσεων. Αυτές οι μέσες τιμές και η συμμεταβλητότητα θα χρειαστούν στην υλοποίηση του φίλτρου Kalman. Η συμμεταβλητότητα είναι ένας πίνακας 2x2 για κάθε τοίχο. Ο κώδικας που υπολογίζει τις μέσες τιμές και τη συμμεταβλητότητα στο LabView φαίνεται παρακάτω.



5-16 Υπολογισμός μέσης τιμής και συμμεταβλητότητας για τα a, r του κάθε τοίχου. Επίσης αφαιρούνται τα μηδενικά που εισάγει αυτόματα το LabView στον πίνακα των wall cluster.

Ο παραπάνω κώδικας έχει δύο εξόδους. Η μία είναι ένας πίνακας με τις μέσες τιμές των γωνιών και των αποστάσεων των τοίχων. Η δεύτερη είναι ένας πίνακας συμμεταβλητότητας για κάθε τοίχο, που θα αναφέρεται παρακάτω ως συμμεταβλητότητα της μέτρησης.

Στο παράδειγμα όπου το ρομπότ είναι στο (0,0) ο πίνακας με τις μέσες τιμές φαίνεται παρακάτω.

mean value from function		
0	0,16381	2,60946
0	1,80059	3,39258
	3,38335	1,19143
	4,44278	1,20451
	0	0

5-17 Μέσες τιμές γωνιών (πρώτη στήλη) και αποστάσεων (δεύτερη στήλη) για τον πρώτο τοίχο όταν το ρομπότ είναι στο (0,0).

Επιπλέον ο πίνακας με τη συµµεταβλητότητα για τον πρώτο τοίχο µε το ρομπότ στο (0,0) φαίνεται στο παρακάτω σχήµα.

0	3,8624E-5	-1,48823E-5	0
0	-1,48823E-5	7,33075E-5	0
0	0	0	0
0	0	0	0

5-18 Πίνακας συµµεταβλητότητας για τον πρώτο τοίχο µε το ρομπότ στο (0,0).

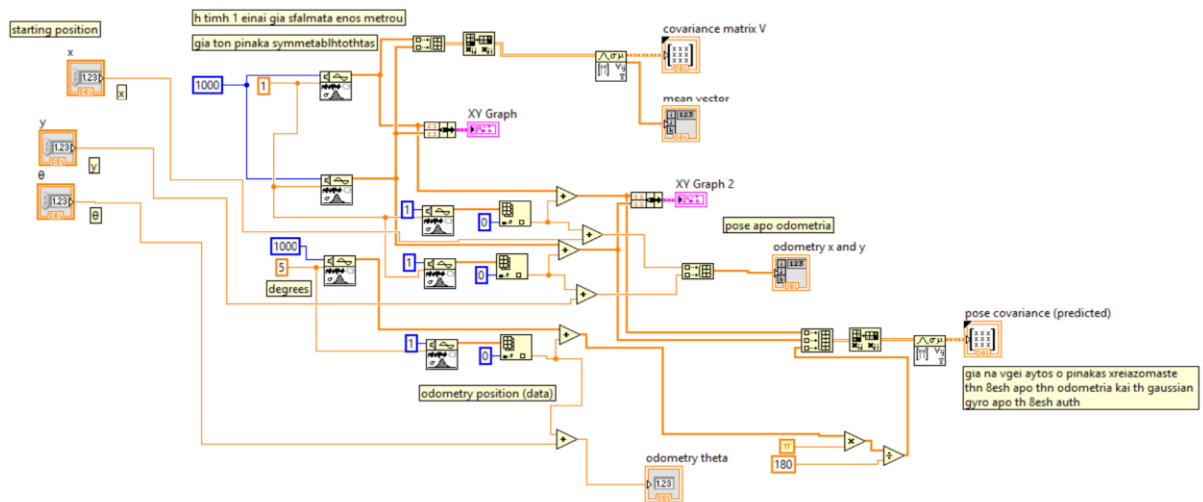
Παρατηρείται ότι οι τιµές του πίνακα αυτού είναι πολύ κοντά στο 0. Οπτικά αυτό φαίνεται από το διάγραµµα των wall clusters όπου ο πρώτος τοίχος εμφανίζεται µε άσπρο χρώµα. Εδώ παράγονται τέσσερεις πίνακες 2x2 ένας για κάθε τοίχο. Το LabView υπολογίζει τον πίνακα αυτό µε τη βοήθεια της συνάρτησης Covariance Matrix.

5.6. Προσοµοίωση οδοµετρίας.

Για να συνεχίσουµε τον αλγόριθµο πρέπει να εισάγουµε σε αυτόν τις µετρήσεις της οδοµετρίας. Σύµφωνα µε την οδοµετρία το ρομπότ βρίσκεται σε µια εκτιµώµενη θέση. Αυτή η θέση δεν είναι η πραγµατική αλλά περιέχει κάποιο σφάλµα που εισάγεται από την οδοµετρία. Για να προσοµοιώσουµε αυτό το σφάλµα παίρνουµε την πραγµατική θέση του ρομπότ και εισάγουµε σε αυτή ένα τυχαίο σφάλµα που έχει γκαουσιανή κατανοµή. Διαλέχθηκε γκαουσιανή µε τυπική απόκλιση στα (x,y) της θέσης του ρομπότ το ένα µέτρο. Στην εκτίµηση του προσανατολισµού του ρομπότ, δηλαδή της γωνίας που σχηµατίζει η πρόσοψη του ρομπότ µε τον άξονα x επιλέχθηκε να εισαχθεί σφάλµα µε γκαουσιανή κατανοµή και τυπική απόκλιση 5 µοίρες.

Επιπλέον έχοντας το pose από την οδοµετρία και την γκαουσιανή κατανοµή γύρω από τη θέση αυτή, υπολογίζουµε το covariance matrix (πίνακα συµµεταβλητότητας) του pose.

Οι παραπάνω υπολογισµοί υλοποιούνται στο LabView µε τον παρακάτω κώδικα. Η έξοδος του κώδικα, που χρησιµοποιείται στον αλγόριθµο παρακάτω είναι τα x,y,θ που δηλώνουν τη θέση του ρομπότ όπως προκύπτει από την οδοµετρία, και ο πίνακας συµµεταβλητότητας της θέσης αυτής. Το LabView µας παρέχει έτοιµη συνάρτηση υπολογισµού του πίνακα της συµµεταβλητότητας.



5-19 Αλγόριθμος προσομοίωσης των τιμών της οδομετρίας. Εισάγεται σφάλμα με Gaussian κατανομή.

5.7.Εξαγωγή predicted τιμών.

Αυτό που μας ενδιαφέρει εδώ είναι να εξάγουμε τις predicted τιμές των θέσεων των τοίχων με βάση τη θέση του ρομπότ όπως εξάγεται από την οδομετρία. Όταν λέμε predicted εννοούμε το που θα έπρεπε να βρίσκονται οι τοίχοι σύμφωνα με τη θέση του ρομπότ από την οδομετρία στο σύστημα συντεταγμένων του ρομπότ (robot frame). Με άλλα λόγια predicted τιμές είναι εκεί που περιμένει το ρομπότ να βρει τα r και α των τοίχων σύμφωνα με τη θέση που το πληροφορεί η οδομετρία ότι βρίσκεται.

Για να εξάγουμε αυτές τις τιμές χρησιμοποιούμε τις παρακάτω εξισώσεις.

Για την εξαγωγή των predicted τιμών των γωνιών των τοίχων (α) χρησιμοποιούμε τον παρακάτω τύπο.

$$\hat{\alpha} = w - \theta \quad (8)$$

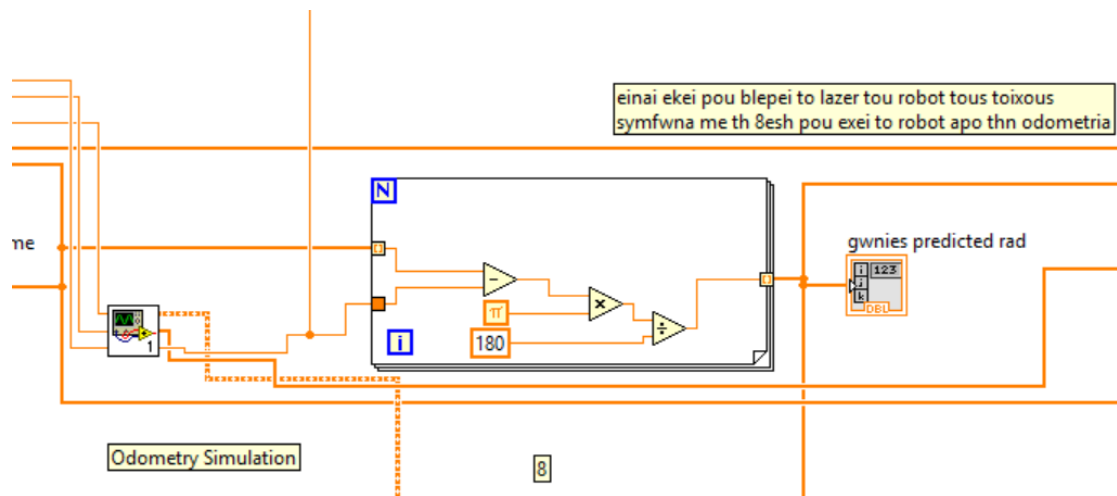
Όπου α με καπελάκι είναι η predicted τιμές των γωνιών των τοίχων (α). διαφορετικά είναι οι τιμές στις οποίες περιμένει το ρομπότ να δει τους τοίχους σύμφωνα με την θέση του στον κόσμο όπως την καταγράφει η οδομετρία. w είναι οι τιμές των γωνιών των τοίχων όπως τις έχουμε από τον χάρτη (world frame) και θ είναι η γωνία που σχηματίζει η πρόσοψη του ρομπότ με τον άξονα x του world frame.

Για την εξαγωγή των predicted τιμών των αποστάσεων των τοίχων από το ρομπότ χρησιμοποιούμε τον παρακάτω τύπο.

$$\hat{r} = r - (x \cos(w) + y \sin(w)) \quad (9)$$

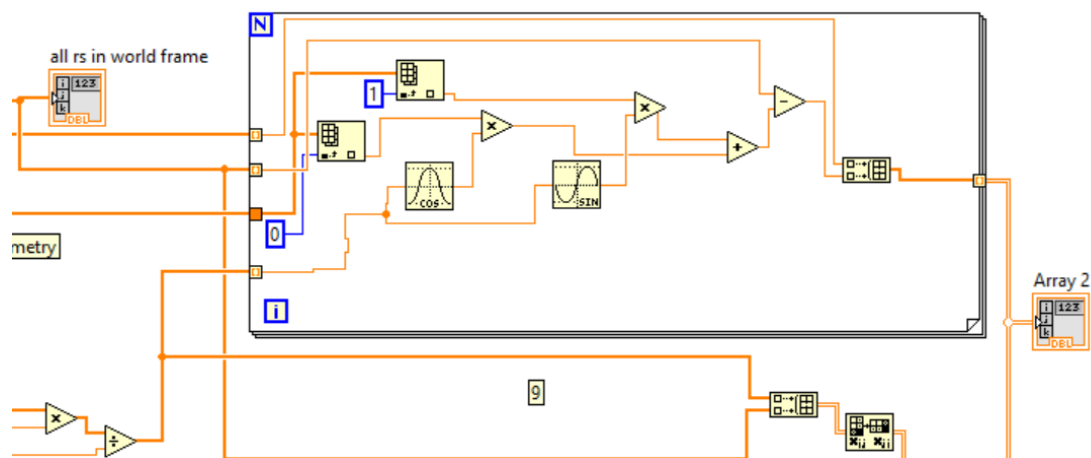
Όπου r με καπελάκι είναι η predicted τιμή των αποστάσεων των τοίχων από το ρομπότ, r είναι η απόσταση του κάθε αντίστοιχου τοίχου από την αρχή των αξόνων του παγκόσμιου συστήματος συντεταγμένων (world frame) και w είναι η γωνία των τοίχων όπως εξάγεται από το world frame.

Ο κώδικας που υλοποιεί τον παραπάνω αλγόριθμο υπολογισμού των predicted τιμών των γωνιών α των τοίχων στο LabView παρατίθεται παρακάτω.



5-20 Εύρεση predicted τιμών γωνιών (α), χρησιμοποιείται ο τύπος 8.

Ο κώδικας που υλοποιεί τον παραπάνω αλγόριθμο υπολογισμού των predicted τιμών των αποστάσεων r των τοίχων στο LabView παρατίθεται παρακάτω.

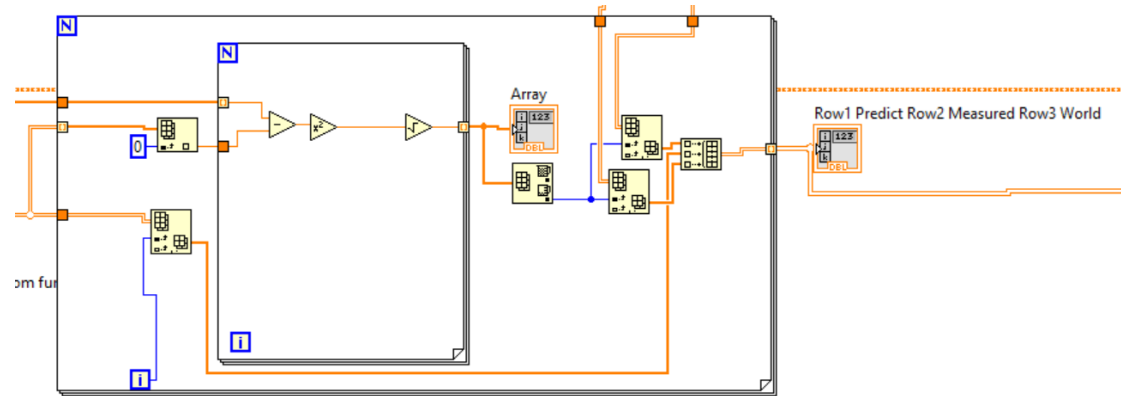


5-21 Εύρεση predicted τιμών αποστάσεων (r) των τοίχων, σύμφωνα με τον τύπο (9).

5.8.Εφαρμογή Extended Kalman Filter

Το φίλτρο Kalman χρειάζεται κάποιες ποσότητες ως εισόδους για να λειτουργήσει. Αυτές είναι τα ζευγάρια τιμών (α, r) των τοίχων όπως τις γνωρίζουμε από τον χάρτη. Επίσης χρειαζόμαστε τα (α, r) των τοίχων όπως προβλέπονται από την οδομετρία (predicted). Τέλος πρέπει να εισάγουμε στο φίλτρο τα (α, r) των τοίχων όπως υπολογίστηκαν από το σαρωτή laser. Όλα τα παραπάνω δεδομένα που χρειάζονται στο φίλτρο Kalman υπολογίστηκαν στις προηγούμενες παραγράφους. Έτσι αυτό που μένει τώρα για να ξεκινήσει η εφαρμογή του

φίλτρου Kalman είναι να συσσωρεύσουμε τις παραπάνω πληροφορίες σε ένα πίνακα. Αυτό γίνεται στον παρακάτω κώδικα LabView.



5-22 Συσσώρευση των τιμών (α,r) των τοίχων από την οδομετρία, από τις μετρήσεις laser και από τον χάρτη σε ένα πίνακα.

Η έξοδος του παραπάνω κώδικα είναι ένας πίνακας που στην πρώτη γραμμή έχει τις τιμές (α,r) όπως υπολογίστηκαν με βάση τη θέση του ρομπότ από την οδομετρία (predicted τιμές) στο robot frame. Στη δεύτερη γραμμή είναι τα (α,r) όπως μετρήθηκαν από τον αισθητήρα laser στο robot frame. Στη τρίτη γραμμή έχουμε τα (α,r) όπως μας τα δίνει ο χάρτης του χώρου στο world frame. Έτσι έχουμε κατασκευάσει ένα πίνακα με τρεις γραμμές και δύο στήλες για κάθε ένα τοίχο του χάρτη.

Έχοντας αυτές τις πληροφορίες ομαδοποιημένες μπορούμε να αρχίσουμε την εφαρμογή του φίλτρου EKF.

Ο στόχος του φίλτρου EKF είναι ο υπολογισμός δύο ποσοτήτων. Της θέσης του ρομπότ και της συμμεταβλητότητας της θέσης. Η συμμεταβλητότητα της θέσης με απλά λόγια μας δείχνει με πόση σιγουριά ξέρουμε την θέση του ρομπότ. Ο στόχος του EKF είναι ο υπολογισμός των δύο αυτών ποσοτήτων όσο καλύτερα γίνεται με βάση τη παρατήρηση από τους αισθητήρες. Έτσι για κάθε landmark που έχουμε εντοπίσει στο χάρτη θα διορθώσουμε τη θέση του ρομπότ αρκεί το landmark να έχει παρατηρηθεί στην πραγματική του θέση. Έτσι landmarks που παρατηρούνται στην πραγματική τους θέση μας διορθώνουν την εκτίμηση για τη θέση του ρομπότ ενώ αυτά που παρατηρούνται σε λάθος θέση χαλάνε την εκτίμηση της θέσης του ρομπότ. Οι τύποι που δίνουν τη θέση και τη συμμεταβλητότητά της φαίνονται παρακάτω.

$$x_t = \hat{x}_t + K_t v_t \quad (10)$$

$$P_t = \hat{P}_t - K_t \Sigma_{IN} K_t^T \quad (11)$$

Όπου στον τύπο (10) x_t είναι η νέα εκτίμηση της θέσης του ρομπότ. x_t με καπελάκι είναι η πρόβλεψη της θέσης του ρομπότ από την οδομετρία. K_t είναι μια ποσότητα που ονομάζεται Kalman Gain και v_t είναι το λεγόμενο innovation.

Στον τύπο (11) P_t είναι η συμμεταβλητότητα της θέσης. P_t με καπελάκι είναι η συμμεταβλητότητα της θέσης όπως προβλέφθηκε από την οδομετρία, K_t είναι το Kalman Gain και Σ_{IN} είναι η συμμεταβλητότητα του innovation.

Το Kalman Gain είναι ένας παράγοντας που υπολογίζεται από την παρακάτω σχέση.

$$K_t = P_t H_t (\Sigma_{IN})^{-1} \quad (12)$$

Όπου το H_t είναι μια ιακoβιανή που χρησιμοποιείται για τη γραμμικοποίηση των εξισώσεων που εισέρχονται στο EKF. Ας θυμηθούμε εδώ την εξίσωση που μας δίνει τη πρόβλεψη της θέσης του ρομπότ από την οδομετρία (predicted τιμές) που είναι οι τύποι (8) και (9).

$$\begin{bmatrix} \hat{a} \\ \hat{r} \end{bmatrix} = \begin{bmatrix} w - \theta \\ r - (\hat{x}_t \cos(w) + \hat{y}_t \sin(w)) \end{bmatrix}$$

Σε αυτή τη σχέση παίρνουμε την ιακoβιανή από τον παρακάτω τύπο.

$$H = \begin{bmatrix} \frac{\partial a}{\partial x} & \frac{\partial a}{\partial y} & \frac{\partial a}{\partial \theta} \\ \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial \theta} \end{bmatrix}$$

Το αποτέλεσμα των παραγωγίσεων μας δίνει τον παρακάτω τύπο για την ιακoβιανή.

$$H = \begin{bmatrix} 0 & 0 & -1 \\ -\cos(w) & -\sin(w) & 0 \end{bmatrix} \quad (13)$$

Εδώ πρέπει να αναφέρουμε ότι η σχέση που μας δίνει την ιακoβιανή δεν είναι γραμμική. Όμως μπορούμε ποιο εύκολα να προσεγγίσουμε το $\cos(w)$ και το $\sin(w)$ με μια γραμμή σε μια μικρή περιοχή παρά το $x\cos(w)+y\sin(w)$. Έτσι με την ιακoβιανή πετυχαίνουμε μια γραμμικοποίηση με λιγότερο σφάλμα. Αυτή η ιακoβιανή υπολογίζεται μια φορά στην αρχή του αλγορίθμου για όλα τα landmarks και χρησιμοποιούμε το αντίστοιχο H_t ανάλογα με το ποιο landmark δουλεύουμε.

Το Σ_{IN} είναι η συμμεταβλητότητα του innovation και δίνεται από τον παρακάτω τύπο.

$$\Sigma_{IN} = H \hat{P}_t H^T + R_t \quad (14)$$

Όπου R_t είναι η συμμεταβλητότητα της μέτρησης, την οποία υπολογίσαμε στην παράγραφο 5.5. Το R_t είναι επίσης μια ποσότητα που υπολογίζεται εξαρχής και για όλα τα landmarks και χρησιμοποιούμε το αντίστοιχο R_t ανάλογα με το ποιο Landmark δουλεύουμε.

Τέλος για να υπολογίσουμε το innovation που συμβολίζεται με v στη σχέση (10) αφαιρούμε από τη μετρούμενη τιμή (measured) για τα (a,r) των τοίχων, την predicted τιμή από την οδομετρία. Χρησιμοποιούμε τον παρακάτω τύπο.

$$v = \begin{bmatrix} a \\ r \end{bmatrix} - \begin{bmatrix} w - \theta \\ r - (\hat{x}_t \cos(w) + \hat{y}_t \sin(w)) \end{bmatrix} \quad (15)$$

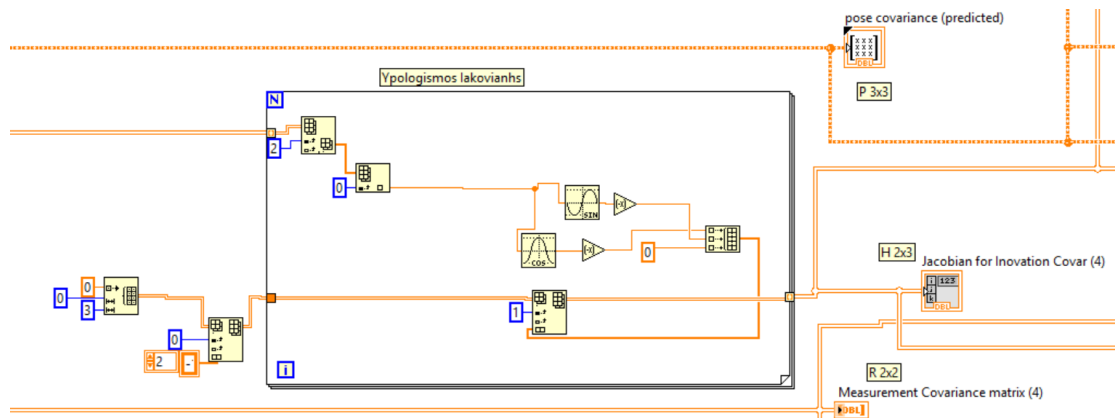
Σύμφωνα με τις παραπάνω σχέσεις υπολογίζουμε τα x,P για τον πρώτο τοίχο, αυτό που κάνουμε στην ουσία είναι ότι χρησιμοποιώντας το πρώτο landmark ανανεώνουμε τη θέση του ρομπότ διορθώνοντας την εκτίμηση που έχουμε από την οδομετρία.

Στη συνέχεια χρησιμοποιώντας τα καινούρια x,P της θέσης επαναλαμβάνουμε τον αλγόριθμο για το επόμενο landmark υπολογίζοντας ξανά όλες τις παραπάνω ποσότητες και ανανεώνουμε για δεύτερη φορά την εκτίμηση της θέσης του ρομπότ σύμφωνα με το δεύτερο landmark. Συνεχίζουμε τη διαδικασία για όλα τα Landmarks. Σε αυτό το πρόβλημα έχουμε

τέσσερα landmarks όμως κάτι τέτοιο προφανώς δεν ισχύει σε κάθε εφαρμογή του αλγόριθμου. Το ρομπότ στη γενική περίπτωση μπορεί να βρίσκεται σε χώρο με περισσότερα ή λιγότερα landmarks. Εδώ πρέπει να τονίσουμε ότι κάθε landmark που μετρήθηκε σωστά (που η θέση του προσδιορίστηκε με ακρίβεια) θα βελτιώσει την εκτίμηση της θέσης του ρομπότ, ενώ κάθε landmark που μετρήθηκε λάθος θα προσθέσει σφάλμα στην εκτίμηση της θέσης του ρομπότ. Εδώ πρέπει να αναφέρουμε ότι όπως παρατηρήθηκε σε αυτή την εργασία κάποιες φορές ένα landmark μπορεί να παραλειφθεί κατά τη διάρκεια της μέτρησης ενώ άλλες φορές μπορεί να εμφανιστεί κάποιο landmark σε μια θέση που δεν θα έπρεπε να παρατηρηθεί τίποτα. Παρόλα αυτά, όπως μας έδειξε αυτή η εργασία, ο αλγόριθμος συγκλίνει σχεδόν πάντα και στο τέλος παίρνουμε μια εκτίμηση για τη θέση του ρομπότ πολύ καλύτερη από αυτή που μας παρέχει η οδομετρία.

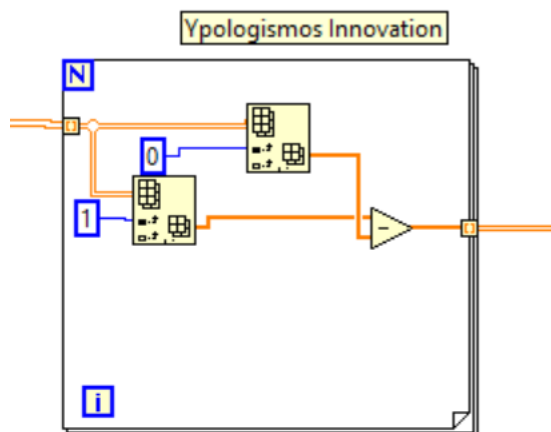
Ας δούμε τώρα μερικά σημεία του αλγόριθμου EKF που αναλύσαμε παραπάνω όπως υλοποιήθηκαν με το LabView.

Παρακάτω δίνεται ο κώδικας που υπολογίζει την ιακοβιανή, δηλαδή τον πίνακα H .



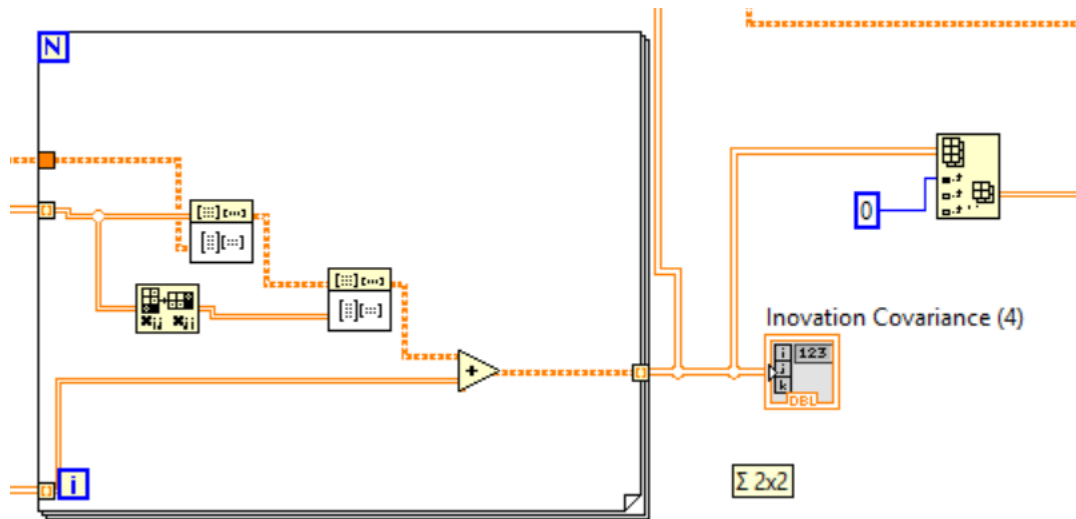
5-23 Υπολογισμός Ιακοβιανής (H είναι πίνακας 2×3).

Παρακάτω δίνεται ο κώδικας που υπολογίζει το innovation (v). Το innovation πρέπει να υπολογιστεί μια φορά για το κάθε landmark. Το innovation είναι ένα διάνυσμα 2×1 .



5-24 Κώδικας LabView για τον υπολογισμό του innovation.

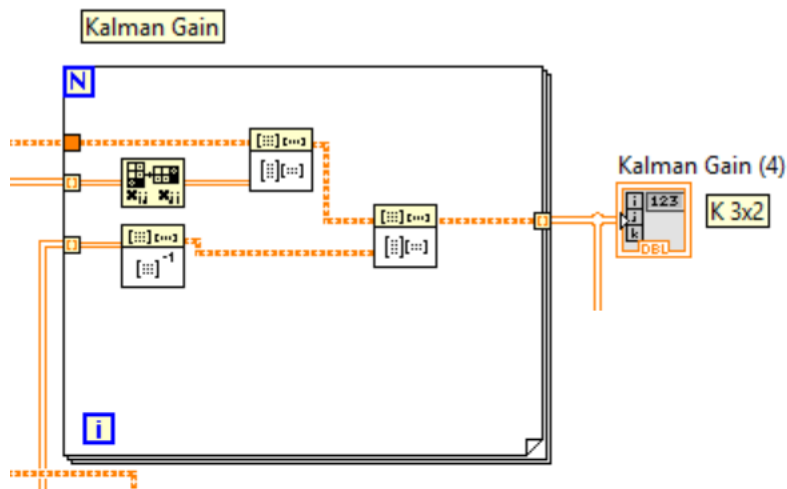
Ο παρακάτω κώδικας υπολογίζει το innovation covariance (συμμεταβλητότητα του innovation). Είναι ένας πίνακας 2x2 και τον συμβολίζουμε με Σ .



5-25 Υπολογισμός Innovation covariance (Σ) 2x2.

Το innovation covariance υπολογίζεται μια φορά για κάθε landmark.

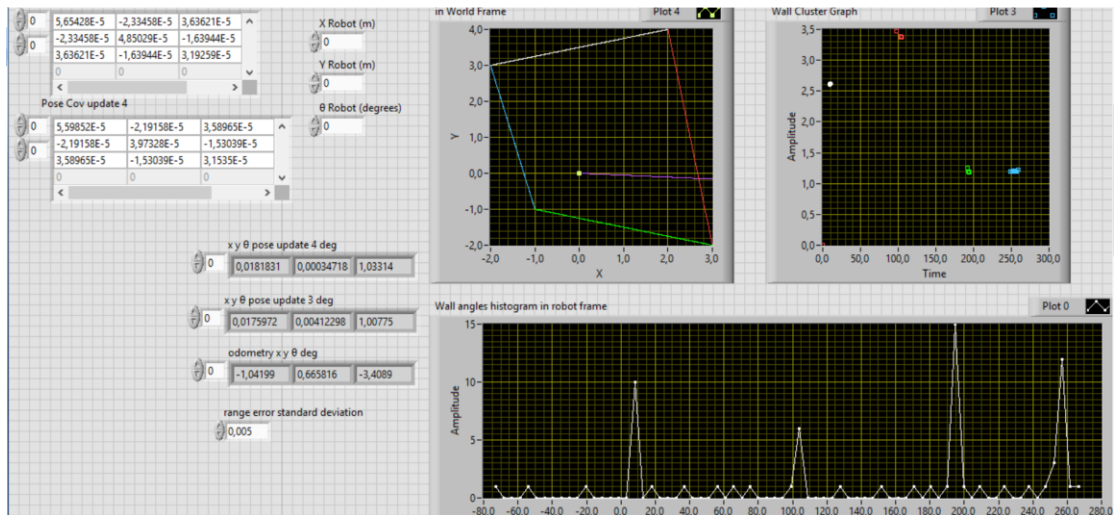
Ο παρακάτω κώδικας υπολογίζει το Kalman Gain (K που είναι πίνακας 3x2).



5-26 Υπολογισμός Kalman Gain. (K 3x2). Το Kalman Gain υπολογίζεται εκ νέου για κάθε landmark.

Παρατηρήστε στα παραπάνω παραδείγματα κώδικα LabView ότι μας παρέχονται πολλές έτοιμες συναρτήσεις για την υλοποίηση του αλγόριθμου. Κάποιες από αυτές είναι ο υπολογισμός πολλαπλασιασμού πινάκων αντιστροφή πινάκων κλπ.

Στο παράδειγμα με το ρομπότ στο (0,0) παίρνουμε τα παρακάτω τελικά αποτελέσματα στην εκτίμηση της θέσης του ρομπότ. Επίσης παρουσιάζονται το δωμάτιο στο οποίο βρίσκεται το ρομπότ και το ιστόγραμμα των landmarks που στο παράδειγμά μας είναι τέσσερα. Μαζί με το διάγραμμα των wall cluster. Τα δεδομένα έχουν ομαδοποιηθεί σε ένα front panel του LabView.



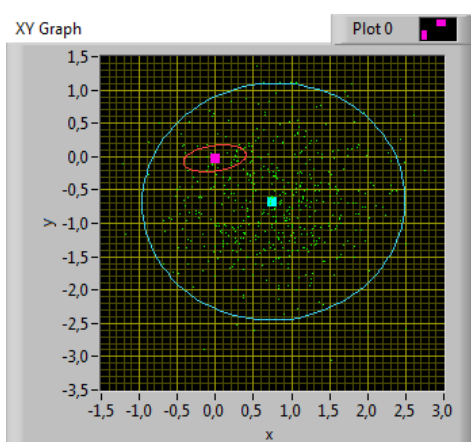
5-27 Αριστερά η πραγματική θέση του ρομπότ και η εκτίμηση της θέσης με την εφαρμογή του EKF (pose update 3 και 4) επίσης παρατίθενται οι πίνακες συμμεταβλητότητας της εκτίμησης που μας δείχνουν την αβεβαιότητα των updates. Δεξιά το δωμάτιο που είναι τοποθετημένο το ρομπότ. Τα wall clusters και το ιστόγραμμα όπου φαίνεται πως εντοπίζονται από τον κώδικά μας τα 4 landmarks.

Παρατηρούμε ότι η εκτίμηση της θέσης είναι πολύ καλή καθώς και η εκτίμηση της γωνίας. Το σφάλμα στην εκτίμηση της θέσης είναι στην περιοχή των εκατοστών ενώ στην εκτίμηση της γωνίας έχουμε σφάλμα 1 μοίρα. Τα αποτελέσματα είναι πολύ καλύτερα από την εκτίμηση της θέσης μέσω οδομετρίας. Οι πίνακες συμμεταβλητότητας της θέσης είναι σχεδόν μηδενικοί κάτι που μας πληροφορεί ότι γνωρίζουμε τη θέση με πολύ μικρή (μηδενική) αβεβαιότητα.

Ολόκληρος ο κώδικας υλοποίησης του αλγόριθμου της εργασίας μπορεί να βρεθεί στο CD που συνοδεύει αυτή την εργασία.

6. Αξιολόγηση των αποτελεσμάτων

1. Ο στόχος μας για βελτίωση της εκτίμησης της θέσης του ρομπότ από την αρχική εκτίμηση μέσω οδομετρίας χρησιμοποιώντας το στοχαστικό φίλτρο Kalman επιτεύχθηκε. Η καινούρια εκτιμημένη θέση είναι πολύ κοντά στην αρχική και τη γνωρίζουμε με πολύ λιγότερη αβεβαιότητα από την αρχική εκτίμηση της οδομετρίας. Αυτό μπορεί να συνοψιστεί στο παρακάτω σχήμα.



6-1 Η αρχική εκτίμηση θέσης φαίνεται με μπλέ χρώμα και η τελική με ροζ. Η αβεβαιότητα στην εκτίμηση της θέσης παριστάνεται με μπλέ (κύκλος) και κόκκινο (έλλειψη). Η διόρθωση έγινε με εφαρμογή του ΕΚΦ για τέσσερα ορόσημα (τους τέσσερεις τοίχους). Η πραγματική θέση είναι η (0, 0). [19]

2. Το λογισμικό προσομοίωσης που δημιουργήσαμε αποδεικνύεται άριστο εκπαιδευτικό εργαλείο για τη μελέτη στοχαστικών προβλημάτων εντοπισμού του ρομπότ, καθώς αποτελείται από επιμέρους υποπρογράμματα το καθένα από τα οποία εικονοποιεί ή επιλύει κάποιο μέρος του προβλήματος. Έτσι υποβοηθείται η κατανόηση όλων των βημάτων του εντοπισμού, από τη δημιουργία του χάρτη και τη λήψη μετρήσεων μέχρι και την εφαρμογή του φίλτρου Kalman.
3. Το εργαλείο προσομοίωσης και γραφικής απόδοσης του προβλήματος του εντοπισμού που δημιουργήσαμε θα μπορούσε να χρησιμοποιηθεί στην τάξη για την καλύτερη κατανόηση του προβλήματος του εντοπισμού και των εργαλείων που χρησιμοποιούνται, προγραμματιστικών και μαθηματικών, για την επίλυση αυτού του προβλήματος.
4. Το περιβάλλον LabView αποδεικνύεται άριστο εργαλείο εποπτείας και υλοποίησης όλων των βημάτων του προβλήματος με το οποίο καταπιάνεται αυτή η εργασία. Καταρχήν για την κατανόηση του κώδικα που παράγεται με το LabView δεν απαιτείται γνώση κάποιας γλώσσας προγραμματισμού. Το ίδιο το LabView είναι μια γλώσσα προγραμματισμού με οπτικά στοιχεία που μπορεί να διαβαστεί με λίγη προσπάθεια ως λογικό διάγραμμα. Επιπλέον τα μαθηματικά εργαλεία που χρησιμοποιούνται μπορούν να γίνουν εύκολα κατανοητά και άμεσα διαχειρίσιμα. Επίσης μας προσφέρει έναν εύκολο τρόπο οπτικοποίησης των αποτελεσμάτων μας σε διαγράμματα και πίνακες.
5. Το περιβάλλον που αναπτύχθηκε για την επίλυση του προβλήματος είναι ένα αποτελεσματικό περιβάλλον για ανάλυση του τύπου what if. Όπου αλλαγές στον

τύπο των σφαλμάτων και το ποσοστό της αβεβαιότητας επηρεάζουν τη λειτουργία του αλγόριθμου.

6. Το εργαλείο που κατασκευάστηκε είναι επεκτάσιμο και μπορεί να χρησιμοποιηθεί για τη προσομοίωση του εντοπισμού με διάφορους χάρτες, με διάφορα ορόσημα και με διαφορετικά μαθηματικά φίλτρα από το Kalman που χρησιμοποιήθηκε εδώ.

7. Σύνοψη συμπερασμάτων και μελλοντική επέκταση της εργασίας.

Σε αυτή την εργασία ασχοληθήκαμε με το πρόβλημα του εντοπισμού ενός ρομπότ σε γνωστό χάρτη. Παρόλα αυτά δεν λύσαμε το γενικότερο πρόβλημα του εντοπισμού, αλλά το απλοποιήσαμε ώστε να μπορέσουμε να το λύσουμε εύκολα. Σε αυτή την εργασία υποθέσαμε ότι το ρομπότ βρίσκεται σε ένα μόνο δωμάτιο και τα μόνα landmark που έχει είναι οι τέσσερις τοίχοι του δωματίου. Επίσης υποθέσαμε ότι κάνει ένα μόνο βήμα και δεν προσομοιώσαμε συνεχή κίνηση του ρομπότ. Αυτό το απλοποιημένο πρόβλημα εντοπισμού λύθηκε με την απαιτούμενη ακρίβεια και ικανοποιητικά αποτελέσματα.

Προσομοιώσαμε ικανοποιητικά τα δεδομένα που παίρνει το ρομπότ από το περιβάλλον του χρησιμοποιώντας οδομετρία και έναν αισθητήρα laser μέτρησης αποστάσεων. Η μόνη υπόθεση που κάναμε για την οδομετρία και το laser είναι ότι τα σφάλματα που εισάγονται ακολουθούν γκαουσιανή κατανομή. Η προσομοίωση των σφαλμάτων που εισάγονται έγινε σε πολύ καλό βαθμό.

Τέλος εφαρμόσαμε το φίλτρο EKF για να διορθώσουμε την εκτίμηση της θέσης από την οδομετρία χρησιμοποιώντας τα δεδομένα του laser. Έτσι καταφέραμε να διορθώσουμε σε πολύ καλό βαθμό την εκτίμηση της θέσης του ρομπότ χρησιμοποιώντας τα landmarks.

Χρησιμοποιώντας το περιβάλλον LabView καταφέραμε να κατασκευάσουμε ένα πρόγραμμα προσομοίωσης φιλικό προς το χρήστη και ανοικτό σε περαιτέρω επεκτάσεις. Αυτό το πρόγραμμα μπορεί εύκολα να χρησιμοποιηθεί για διδακτικούς σκοπούς μιας και απλουστεύει και οπτικοποιεί το πρόβλημα του εντοπισμού ενός ρομπότ έστω και στην απλή μορφή που μελετήθηκε σε αυτή την εργασία.

Φυσικά αυτή η εργασία επιδέχεται πολλές επεκτάσεις και μπορεί να χρησιμοποιηθεί ως αρχή για την επίλυση ενός συνθετότερου προβλήματος εντοπισμού του ρομπότ καθώς και του γενικότερου προβλήματος εντοπισμού και χαρτογράφησης που αναφέρεται στην βιβλιογραφία ως SLAM.

Στην γενικότερη μορφή του προβλήματος του εντοπισμού το ρομπότ θα εκτελούσε μια σχετικά ομαλή κίνηση στον χώρο του καθώς θα εντόπιζε τον εαυτό του μέσα σε αυτό σε κάθε του βήμα. Επίσης η αρχική θέση του ρομπότ θα μπορούσε να είναι άγνωστη (kidnapped robot problem). Ο χώρος θα μπορούσε να απαρτίζεται από περισσότερα του ενός δωμάτια και η μορφή του χάρτη που χρησιμοποιείται για να περιγράψει το χώρο θα μπορούσε να είναι διαφορετική. Επίσης θα μπορούσαν να χρησιμοποιηθούν άλλες μέθοδοι βελτίωσης της εκτίμησης της θέσης (στοχαστικά φίλτρα) από το Extended Kalman Filter που χρησιμοποιήθηκε εδώ. Από πλευράς hardware θα μπορούσαν να χρησιμοποιηθούν επιπλέον αισθητήρες για τον εντοπισμό των landmark όπως κάμερες.

Άλλη μια επέκταση της εργασίας θα ήταν η λύση του προβλήματος “κλεισίματος του βρόγχου” (closing the loop) όπως αναφέρεται στον βιβλιογραφία όπου το ρομπότ προσπαθεί να περιηγηθεί στον χώρο και να καταλήξει στο αρχικό σημείο από το οποίο ξεκίνησε. Επιπλέον η εργασία αυτή επιδέχεται και άλλες επεκτάσεις, όπως η χρήση έτοιμων μαθηματικών εργαλείων μέσω του matlab, η εισαγωγή των οποίων στο LabView μπορεί να γίνει πολύ εύκολα. Τέλος ως επέκταση της παρούσας εργασίας θα μπορούσε κανείς να μεταφέρει τον κώδικα που παρατίθεται σε αυτή την εργασία σε κάποια άλλη γλώσσα προγραμματισμού όπως η python ώστε το πρόγραμμα να τρέχει σε πραγματικό χρόνο σε

κάποιο hardware όπως raspberry PI το οποίο θα είναι τοποθετημένο πάνω σε ένα πραγματικό ρομπότ.

8. Βιβλιογραφία

- [1] K. O. Arras, N. Tomatis, “Improving Robustness and Precision in Mobile Robot Localization by Using Laser Range Finding and Monocular Vision”, in the Proceedings of the Third European Workshop on Advanced Mobile Robots (Eurobot’99), Zurich, Switzerland, 6-8 Sept. 1999.
- [2] M. B. Rhuby, R.A. Salguero and K. Holappa, A Kalman Filtering tutorial for undergraduate Students, International Journal of Computer Science and Engineering Survey, vol. 8 (1), 2017, pp. 1-18.
- [3] R. Siegwart, I.R. Nourbakhsh, and D. Scaramuzza, Introduction to Autonomous Mobile Robots second edition (Book), The MIT Press, Cambridge, Massachusetts, 2011 Massachusetts Institute of Technology, Original Edition 2004.
- [4] J. Biswas and M. Veloso, Multi-Sensor Mobile Robot Localization For Diverse Environments (2014), RoboCup 2013: Robot World Cup XVII. 1333, Retrieved from https://scholarworks.umass.edu/cs_faculty_pubs/1333
- [5] HUGH DURRANT-WHYTE AND TIM BAILEY, Tutorial, Simultaneous Localization and Mapping: Part I & Part II.
- [6] Søren Riisgaard and Morten Rufus Blas, Slam For Dummies (Book).
- [7] Eduardo Nebot Australian Centre for Field Robotics The University of Sydney NSW 2006 Australia, Simultaneous Localization and Mapping 2002 Summer School, July 31, 2002 Version 0.9
- [8] Guido Zunino, Simultaneous Localization and Mapping for Navigation in Realistic Environments, Licentiate Thesis Royal Institute of Technology Numerical Analysis and Computer Science, February 2002.
- [9] Heber Sobreira A. Paulo Moreira Paulo Costa José Lima , (2016),"Robust mobile robot localization based on a security laser: an industry case study", Industrial Robot: An International Journal, Vol. 43 Iss 6 pp.
- [10] A. de la Eseglera, L. Moreno, J. Ma Armingol, M. A. Saliebs, Real Time Mobile Robot Localization by Using a Laser and a Geometric Map, Universidad Carlos III. Escuela Politecnica Superior, Departamento de Ingenieria Leganes, Madrid, Spain
- [11] Hamed Bastani and Hamid Mirmohammad-Sadeghi (2010). Ranging Fusion for Accurating State of the Art Robot Localization, Robot Localization and Map Building, Hanafiah Yussof (Ed.), ISBN: 978-953-7619-83-1, InTech, Available from: <http://www.intechopen.com/books/robot-localization-and-map-building/ranging-fusionfor-accurating-state-of-the-art-robot-localization>.
- [12] Kai Lingemanna,*, Andreas Nüchtera, Joachim Hertzberga, Hartmut Surmannb, High-speed laser localization for mobile robots a University of Osnabrück, Institute of Computer Science, Knowledge-Based Systems Research Group, Albrechtstr. 28, D-49069 Osnabrück,

Germany b Fraunhofer Institute for Autonomous Intelligent Systems (AIS), Schloss Birlinghoven, D-53754 Sankt Augustin, Germany, Available online 7 April 2005

[13] Biswas, Joydeep and Veloso, Manuela M., "Multi-Sensor Mobile Robot Localization For Diverse Environments" (2014).RoboCup 2013: Robot World Cup XVII. 1333.

[14] David Sandberg, Krister Wolff, and Mattias Wahde, A Robot Localization Method Based on Laser Scan Matching, Department of Applied Mechanics Chalmers University of Technology, 412 96 Goteborg, Sweden.

[15] Takeshi Takahashi, 2D localization of outdoor mobile robots using 3D laser range data, Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania 15213, May 2007.

[16] S. Thrun W. Burgard D. Fox, Probabilistic Robotics (BOOK), The MIT Press Cambridge Massachusetts London England 2006.

[17] Amani Ben Afia, Lina Deambrogio, Daniel Salós, Anne-Christine Escher, Christophe Macabiau, et al.. Review and classification of vision-based localisation techniques in unknown environments. IET Radar, Sonar. Navigation, 2014, 8 (9), pp. 1059-1072. 10.1049/iet-rsn.2013.0389 hal-00996022

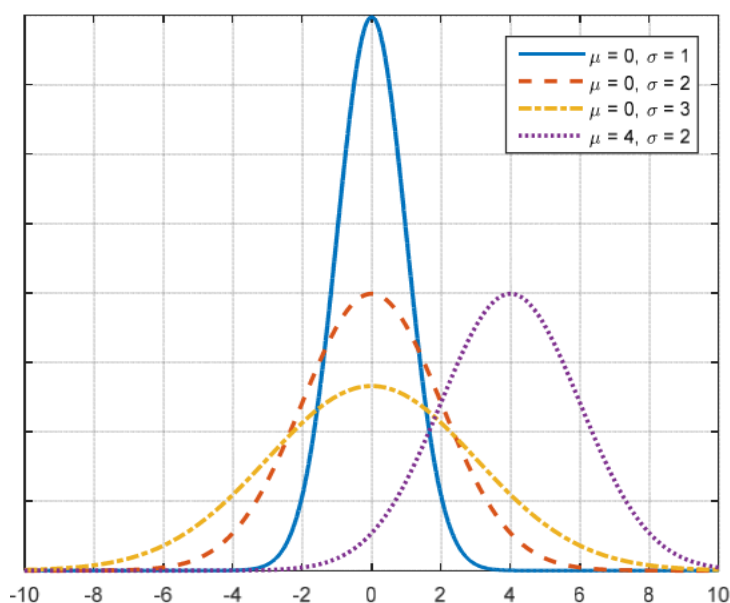
[18] Guillaume Bresson, Zayed Alsayed, LiYu, Sébastien Glaser. Simultaneous Localization And Mapping: A Survey of Current Trends in Autonomous Driving. IEEE Transactions on Intelligent Vehicles, IEEE, 2017, XX, pp.1.-10.1109/TIV.2017.2749181 hal-01615897

[19] I. Καλόμοιρος «Εισαγωγή στο πρόβλημα του εντοπισμού», Σημειώσεις από το μάθημα Αυτόνομα Ρομποτικά Συστήματα που διδάσκεται στο μεταπτυχιακό ρομποτικής.

9. Παραρτήματα

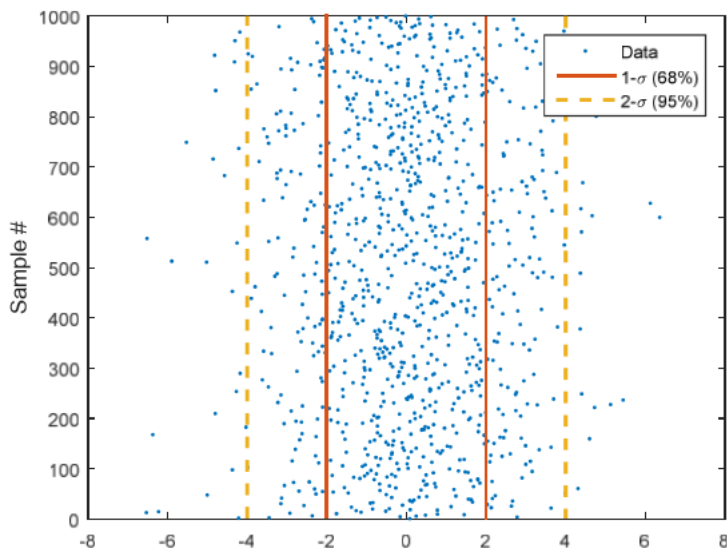
9.1.Παράρτημα 1: Επεξήγηση πινάκων συµµεταβλητότητας

Για να µπορέσουµε να µιλήσουµε για τη συµµεταβλητότητα, πρέπει να καταλάβουµε τι είναι η µεταβλητότητα, που µπορεί να οριστεί ως το τετράγωνο της τυπικής απόκλισης. Τυπικός συµβολισµός για τη µεταβλητότητα είναι σ^2 όπου µε το σ αναφερόµαστε στην τυπική απόκλιση. Σε αυτό το παράδειγµα θα προσπαθήσουµε να µην αναφέρουµε πολλές έννοιες στατιστικής, παρόλα αυτά κάποιες έννοιες είναι απαραίτητες. Ας υποθέσουµε ότι έχουµε µια µεταβλητή x που έχει γκαουσιανή κατανοµή (να σηµειωθεί ότι αυτή η υπόθεση χρησιµοποιείται στο φίλτρο Kalman). Η γκαουσιανή είναι µια καµπύλη σε µορφή καµπάνας και µπορεί να παραµετροποιηθεί µε δύο τιµές: τη µέση τιµή μ και την τυπική απόκλιση σ (ή τη µεταβλητότητα σ^2). Ο συµβολισµός µε όρους στατιστικής θα ήταν $x \sim N(\mu, \sigma^2)$, όπου µε N συµβολίζουµε την κανονική κατανοµή. Αν πάρουµε τυχαία ένα σηµείο από αυτή την κατανοµή υπάρχει µεγαλύτερη πιθανότητα ότι η τιµή που θα πάρουµε θα είναι κοντά στη µέση τιµή, και µικρότερη πιθανότητα ότι θα είναι αρκετά διαφορετική από τη µέση τιµή. Αυτές οι πιθανότητες καθορίζονται από την κατανοµή σε µορφή καµπάνας. Όσο µεγαλύτερη είναι η τυπική απόκλιση τόσο µεγαλύτερη θα είναι η διασπορά των τιµών για την κατανοµή. Μερικά παραδείγµατα τέτοιων κατανοµών φαίνονται παρακάτω. Να σηµειωθεί εδώ ότι το επίσηµο όνοµα της κατανοµής είναι Probability Density Function (PDF).



9-1 Διάφορες Gaussian κατανομές με αντίστοιχες μέσες τιμές και τυπικές αποκλίσεις. [2]

Τώρα θα παρθούν δεδομένα από µια κανονική κατανοµή µε µέση τιµή µηδέν $m=0$ και τυπική απόκλιση $\sigma=2$. Παίρνοντας 1000 δείγµατα από αυτή την κατανοµή µας δίνει το αποτέλεσµα στο σχήµα 10-2. Να σηµειωθεί ότι στο σχήµα έχουν σχεδιαστεί γραµµές για να δείξουν αποκλείσεις $\pm 1\sigma$ και $\pm 2\sigma$ που καταδείχνουν 68% και 95% διαστήµατα εµπιστοσύνης αντίστοιχα. Περιµένουµε ότι 68% των δεδοµένων πρέπει να βρίσκονται µεταξύ των γραµµών µε τυπική απόκλιση $\pm 1\sigma$ και 95% µεταξύ των γραµµών µε τυπική απόκλιση $\pm 2\sigma$.

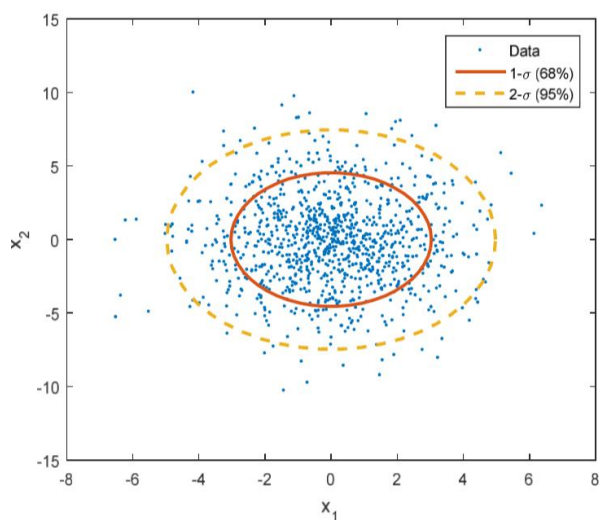


9-2 Δειγματοληψία 1000 δειγμάτων από κανονική κατανομή με $\mu=0$ και $\sigma=2$. [2]

Τώρα πρέπει να επεκτείνουμε τη συζήτησή μας σε περισσότερες διαστάσεις. Πώς μπορούμε να περιγράψουμε τυχαίες διακυμάνσεις σε διανύσματα αντί για αριθμούς; Θα περιοριστούμε στις δύο διαστάσεις για απλότητα, αλλά η ιδέα είναι ίδια και για περισσότερες διαστάσεις. Όταν αναφερόμαστε σε πολλές διαστάσεις αντί για τη χρήση της μεταβλητότητας για την περιγραφή της κατανομής χρειάζεται να υπολογίσουμε το πίνακα συμμεταβλητότητας. Ας υποθέσουμε ένα διάνυσμα με 2 διαστάσεις χ_1 και χ_2 . Στην απλή περίπτωση που η κατανομή των δύο συνιστωσών είναι ανεξάρτητη, δηλαδή η διακύμανση του χ_1 δεν εξαρτάται από αυτή του χ_2 και ανάποδα, ο πίνακας συμμεταβλητότητας θα είναι διαγώνιος. Ας πάρουμε ένα παράδειγμα με μέσες τιμές μηδέν $\mu=0$ και με πίνακα συμμεταβλητότητας όπως παρακάτω

$$P = \begin{bmatrix} \sigma_{\chi_1}^2 & 0 \\ 0 & \sigma_{\chi_2}^2 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 9 \end{bmatrix}$$

Εάνά παίρνονται 1000 δείγματα και το αποτέλεσμα φαίνεται στο παρακάτω σχήμα.

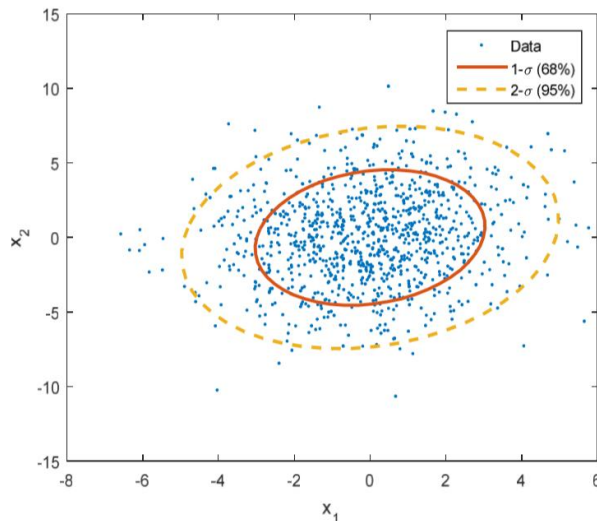


9-3 Κατανομή 1000 δειγμάτων με τα χ_1 και χ_2 ανεξάρτητα. [2]

Να σημειωθεί ότι τα σύνορα με $\pm 1\sigma$ και $\pm 2\sigma$ είναι πλέον ελλείψεις αντί για γραμμές. Σε περισσότερες διαστάσεις είναι πιθανό ότι το σφάλμα στη μία συνιστώσα εξαρτάται από την άλλη συνιστώσα. Σε αυτή τη περίπτωση υπάρχει μια σχέση μεταξύ των δύο συνιστωσών. Παρακολουθείστε το παρακάτω παράδειγμα

$$P = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2}^2 \\ \sigma_{x_1 x_2}^2 & \sigma_{x_2}^2 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 1 & 9 \end{bmatrix}$$

Η τελική κατανομή φαίνεται στο παρακάτω σχήμα.



9-4 Κατανομή 1000 δειγμάτων με τα x_1 και x_2 να έχουν κάποια εξάρτηση μεταξύ τους. [2]

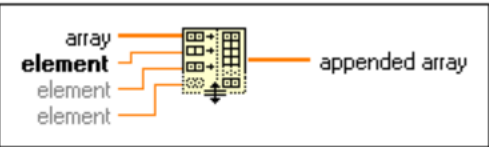
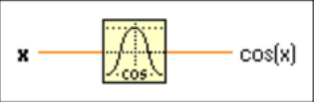
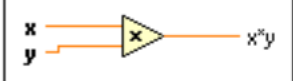
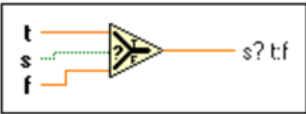

Σημειώστε τις διαφορές μεταξύ των σχημάτων 10-3 και 10-4. Στο 10-4 φαίνεται η στροφή της κατανομής που περιγράφει την ζεύξη των δύο μεταβλητών. Στο πλαίσιο του φίλτρου Kalman το σημαντικό στους πίνακες συμμεταβλητότητας είναι πώς σχετίζονται τα σφάλματα, αν δηλαδή το σφάλμα στην πρώτη μέτρηση εξαρτάται από το σφάλμα στη δεύτερη και ούτο καθ' εξής.

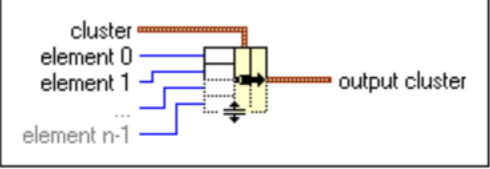
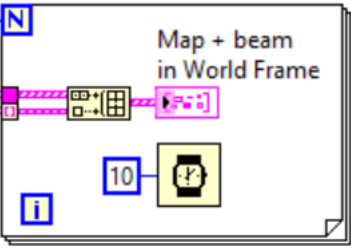
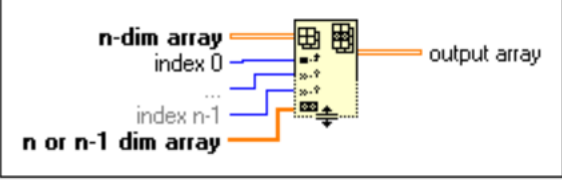

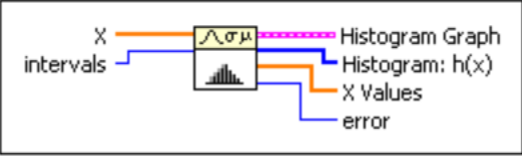
9.2. Παράρτημα 2. Στοιχεία προγραμματισμού LabView 2011.

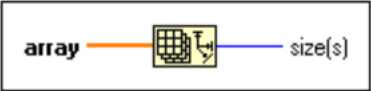
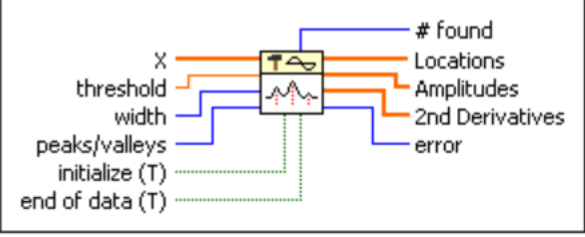
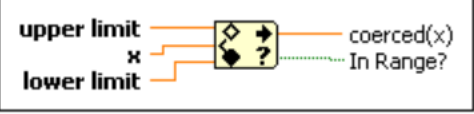
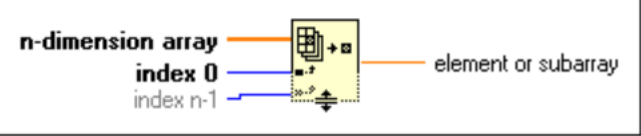

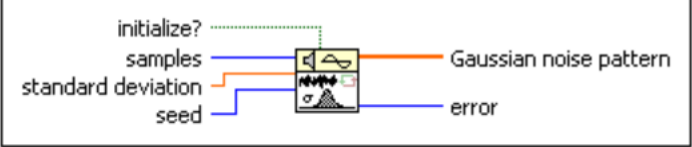

Το LabView 2011 είναι μια γλώσσα προγραμματισμού με οπτικά στοιχεία. Επιλέχθηκε να χρησιμοποιηθεί για την υλοποίηση του αλγόριθμου που κατασκευάστηκε σε αυτή την εργασία διότι παρέχει ένα άμεσο και εύκολο τρόπο αναπαράστασης διαγραμμάτων και πινάκων. Εδώ θα επεξηγηθούν ορισμένα στοιχεία προγραμματισμού που χρησιμοποιήθηκαν σε αυτή την εργασία.

Στον παρακάτω πίνακα συνοψίζονται και επεξηγούνται τα κυριότερα από τα οπτικά στοιχεία προγραμματισμού που χρησιμοποιήθηκαν σε αυτή την εργασία και αναφέρεται η παλέτα στην οποία τα βρίσκουμε στο LabView 2011.

Στοιχείο	Επεξήγηση	Παλέτα στην
----------	-----------	-------------

		οποία βρίσκεται
	<p>Συνάρτηση κατασκευής πίνακα (array). Φτιάχνει ένα πίνακα με τα στοιχεία που δίνονται ως είσοδοι. Μπορούν να δοθούν και υποπίνακες ως είσοδοι για την κατασκευή ενός μεγαλύτερου πίνακα.</p>	<p>Programming → Array → Build Array</p>
	<p>Συνάρτηση συνημίτονου. Το x πρέπει να είναι ακτίνια. Στην ίδια παλέτα προσφέρονται όλες οι κοινές τριγωνομετρικές συναρτήσεις.</p>	<p>Mathematics → Elementary → Trigonometric.</p>
	<p>Συνάρτηση πολλαπλασιασμού. Πολλαπλασιάζει τα δύο στοιχεία εισόδου. Στην ίδια παλέτα προσφέρονται οι κοινές πράξεις των μαθηματικών (διαίρεση, αφαίρεση κλπ)</p>	<p>Mathematics → Numeric</p>
	<p>Συνάρτηση Select. Αν η συνθήκη είναι αληθής εκτελείται η είσοδος true ενώ αν είναι ψευδής εκτελείται η false. Στην ίδια παλέτα υπάρχουν συναρτήσεις σύγκριση όπως ίσο, άνισο, μεγαλύτερο κλπ.</p>	<p>Programming → Comparison</p>
	<p>Συνάρτηση πολλαπλασιασμού πινάκων. AxB. Στην ίδια παλέτα βρίσκουμε</p>	<p>Mathematics → Linear Algebra</p>

	<p>συναρτήσεις διαχείρισης πινάκων όπως τάξη πίνακα, ανάστροφος πίνακα αντίστροφος πίνακα κλπ.</p>	
	<p>Συνάρτηση δημιουργίας συστάδας (Bundle). Μας βοηθά να ομαδοποιήσουμε δεδομένα που θέλουμε για παράδειγμα να παρασταθούν σε ένα διάγραμμα.</p>	<p>Programming → Cluster Class and Variant</p>
	<p>Συνάρτηση For Loop. Υλοποιεί ένα for loop. Στην ίδια παλέτα βρίσκουμε το while loop και άλλες παρόμοιες προγραμματιστικές δομές.</p>	<p>Programming → Structures</p>
	<p>Συνάρτηση insert into array. Εισάγει σε ένα προϋπάρχον πίνακα ένα στοιχείο ή ένα υποπίνακα. Η είσοδος γίνεται στη θέση που καθορίζουμε.</p>	<p>Programming → Array</p>
	<p>Συνάρτηση covariance matrix. Υπολογίζει τη μέση τιμή και τη συμμεταβλητότητα ενός πίνακα.</p>	<p>Mathematics → Prob & Stat</p>
	<p>Συνάρτηση κατασκευής ιστογράμματος. (Histogram) Φτιάχνει το ιστόγραμμα της εισόδου X.</p>	<p>Mathematics → Prob & Stat</p>

	<p>Συνάρτηση Array Size. Μας δίνει το μέγεθος (μεγέθη) του πίνακα εισόδου.</p>	<p>Programming → Array</p>
	<p>Συνάρτηση Peak Detector. Παίρνει ως είσοδο την έξοδο της συνάρτησης Histogramm και μας δίνει τις θέσεις των κορυφών.</p>	<p>Signal Processing → Sig Operation</p>
	<p>Συνάρτηση in range. Δίνουμε ως είσοδο ένα εύρος τιμών και μας επιστρέφει αν η τιμή που μας ενδιαφέρει είναι ή όχι μέσα στο εύρος.</p>	<p>Programming → Comparison</p>
	<p>Συνάρτηση Index Array. Δίνουμε ως είσοδο ένα πίνακα και μια θέση και μας επιστρέφει το στοιχείο ή τον υποπίνακα στη θέση αυτή.</p>	<p>Programming → Array</p>
	<p>Συνάρτηση Transpose Array. Δίνουμε ένα πίνακα και μας επιστρέφει τον ανάστροφο.</p>	<p>Programming → Array</p>
	<p>Συνάρτηση Gaussian White Noise. Μας δίνει μια γκαουσιανή τυχαία κατανομή τιμών. Μπορούμε να καθορίσουμε τη τυπική απόκλιση της κατανομής και τον αριθμό των δειγμάτων που θέλουμε.</p>	<p>Signal Processing → Sig Generation.</p>
<p>XY Graph</p> 	<p>Στο Front Panel. Είδος γραφήματος χ,γ.</p>	<p>Δεξί κλικ στο Front Panel → Graph Indicators</p>