

# Ρομποτική 2

Σμαράγδα Μπενέτου el18048 — Ανδρέας Βατίστας el18020

## Εξαμηνιαία εργασία 2B

*Εκτίμηση θέσης και προσανατολισμού κινούμενου ρομπότ σε γνωστό χάρτη (Mobile robot Localization)*



Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Ακαδημαϊκό Έτος 2021-2022

# Α.Θεωρητική Ανάλυση

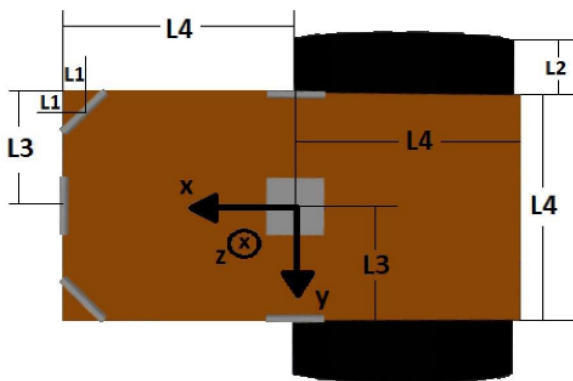
## Στόχος

Εκτίμηση θέσης και προσανατολισμού του αυτοκινούμενου ρομπότ σε γνωστό χάρτη με χρήση Επεκταμένου Φίλτρου Κάλμαν (EKF), κάνοντας χρήση του μοντέλου κίνησης διαφορικής οδήγησης με σύμμιξη αισθητηριακών πληροφοριών (sonars, IMU).

## Προδιαγραφές κινητού ρομπότ

Όπως αναπτύχθηκε και στην wall-following εργασία (2A):

Οι διαστάσεις του κινητού ρομπότ διαφορικής οδήγησης φαίνονται στην παρακάτω εικόνα και οι ρόδες του έχουν ακτίνα 0.1m. Επίσης διαθέτει 5 αισθητήρες σόναρ στις θέσεις που φαίνονται στην παρακάτω εικόνα και έναν IMU στο κέντρο.



Όπου  $L1=0.05\text{m}$ ,  $L2=0.018\text{m}$ ,  $L3=0.1\text{m}$  και  $L4=0.2\text{m}$

## Θόρυβος

Είναι γνωστό ότι οι 5 αισθητήρες υπερήχων σόναρ, οι οποίοι μετρούν απόσταση από εμπόδια, με μέγιστη δυνατότητα μέτρησης τα 2m, έχουν θόρυβο που ακολουθεί κανονική κατανομή **με τυπική απόκλιση 0.01m**.

Επίσης για την IMU (Inertial Measurement Unit) 9 βαθμών ελευθερίας, η οποία μετράει στροφικές ταχύτητες, γραμμικές επιταχύνσεις, καθώς και περιστροφή γύρω από κάθε άξονα, έχει θόρυβο μέτρησης που ακολουθεί

κανονική κατανομή με τυπική απόκλιση  $0.002\text{m/s}^2$  ,  $0.002\text{rad/s}$ , και  $0.002\text{rad}$ , αντιστοίχως.

## ΛΥΣΗ

Η εφαρμογή του Kalman φίλτρου γίνεται σε ρομποτικό όχημα που ικανοποιεί μη-ολόνομους περιορισμούς. Ως state vector ορίζουμε  $(x,y,\theta)$ .

Γνωρίζουμε με πλήρη με βεβαιότητα το αρχικό state  $(x,y,\theta)=(0,0,1.717)$ . [ $\theta=8\text{mod}\pi=1.717\text{rad}$ , όπως υπολογίστηκε και στο 2Α]

Βάσει των μετρήσεων οδομετρίας της IMU εκτελούμε το predict state του Kalman φίλτρου που υλοποιούμε και βάσει μετρήσεων από τα sonars εκτελούμε το update.

Πιο συγκεκριμένα, για το predict κομμάτι:

Από τη δυναμική του δοσμένου συστήματος έχουμε

$$\hat{x}(t + dt|t) = \hat{x}(t) + [v_x(t)dt + \frac{1}{2}a_x dt^2] \cos\theta(t)$$

$$\hat{y}(t + dt|t) = \hat{y}(t) + [v_x(t)dt + \frac{1}{2}a_x dt^2] \sin\theta(t)$$

$$\hat{\theta}(t + dt|t) = \hat{\theta}(t) + \omega(t + dt)dt$$

Cw: Μήτρα συνδιακύμανσης μοντέλου θορύβου

$$C_w = E[w(t)w^T(t)] = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_\theta^2).$$

Αυθαίρετα θεωρήσαμε ασυσχέτιστο το θόρυβο μεταξύ  $x$ ,  $y$ ,  $\theta$  και προέκυψε διαγώνιος  $3 \times 3$  πίνακας.

Προφανώς με κατάλληλο tuning θα υπάρχει αρκετά καλύτερη predict εκτίμηση αλλά αρκούμαστε στην παρούσα φάση σε αυτή την υλοποίηση.

Για τις διασπορές των  $x$ ,  $y$ ,  $\theta$  αντίστοιχα έχουμε, από τις εξισώσεις της δυναμικής του συστήματος:

$$\sigma_x = \frac{1}{2} \sigma_{\alpha x} dt^2 \cos \theta,$$

$$\sigma_y = \frac{1}{2} \sigma_{\alpha x} dt^2 \sin \theta,$$

$$\sigma_\theta = \sigma_\omega dt$$

(Προφανώς θεωρώντας ότι η γνώση των αντίστοιχων μεγεθών για τωρινές/παρελθοντικές στιγμές έχουν πλήρη βεβαιότητα.)

Μήτρα A για γραμμικοποίηση του χώρου κατάστασης:

$$A = \begin{bmatrix} 1 & 0 & -[v_x(t)dt + \frac{1}{2}a_x dt^2] \sin \theta(t) \\ 0 & 1 & +[v_x(t)dt + \frac{1}{2}a_x dt^2] \cos \theta(t) \\ 0 & 0 & 1 \end{bmatrix}$$

P: Μήτρα συνδιακύμανσης του θορύβου βέλτιστης εκτίμησης κατάστασης την τρέχουσα χρονική στιγμή t

-για t=0, έχουμε με πλήρη βεβαιότητα τις τιμές του state vector και επομένως P(0)=3x3 μηδενικός πίνακας.

Γενικά θα ισχύει:  $P(k+1|k) = AP(k)A^T + C_w$  και

$$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}] \cdot \mathbf{P}(k+1|k)$$

Για το update:

Χωρίζουμε σε πέντε περιπτώσεις: sonar front + sonar front left, sonar front + sonar front right, sonar front , sonar front left + sonar left, sonar front right + sonar right

Εξετάζουμε τις περιπτώσεις αυτές με αυτήν την προτεραιότητα μέχρι να βρεθούν μετρήσεις για το  $x$ , για το  $y$  και για τον προσανατολισμό του κινητού.

Για sonar front+ sonar front left:

Ελέγχουμε τις 4 περιπτώσεις προσανατολισμού και τον αντίστοιχο τοίχο. (Οι τοίχοι αριθμούνται με 1 τον πάνω τοίχο και μέχρι το 4 αριστερόστροφα)

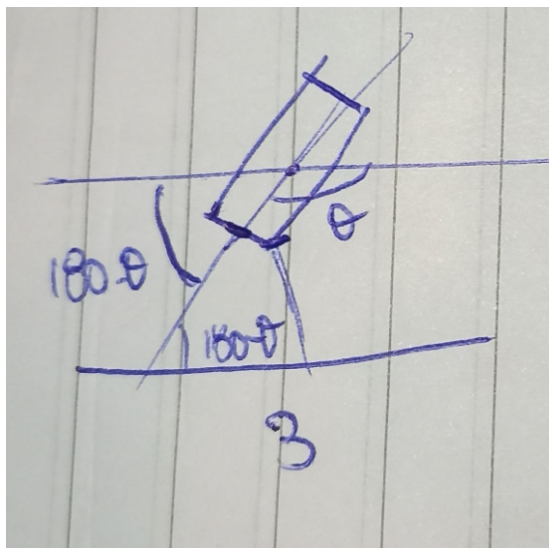
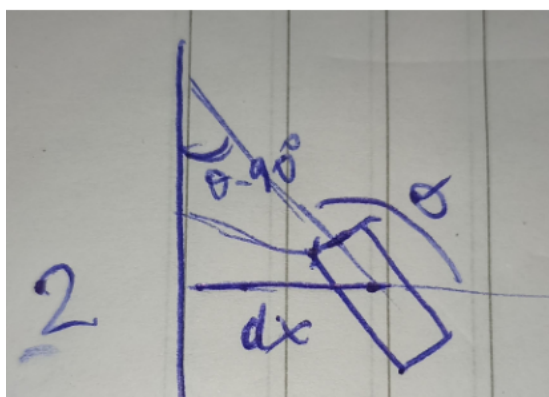
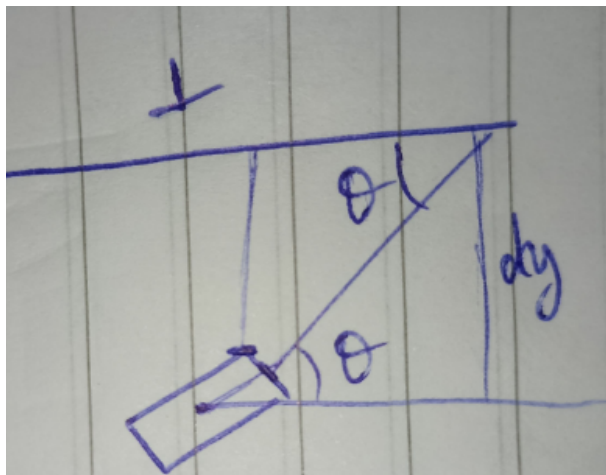
Όπως βλέπουμε στο παρακάτω σχήμα σε κάθε περίπτωση συσχετίζουμε την γωνία  $\theta$  την θεωρητική με την εσωτερική γωνία μεταξύ του τοίχου και του sonar front. Σκοπός μας να βρούμε το  $dy$  από τον τύπο :  $dy = \sin(a) * (\text{sonar\_front} + L4)$  όπου  $a$  η γωνία  $\theta$  που προκύπτει από τις μετρήσεις. Δηλαδή  $l3 = \sqrt{l1^2 + l2^2 - 2 * l1 * l2 * \cos(df)}$ ,  $a = \arcsin(\sin(df) * l1 / l3)$

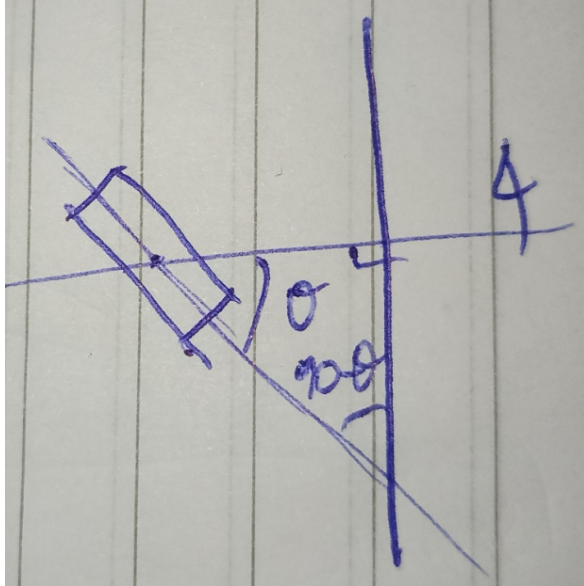
Όπου  $l1 = \text{sonar\_front\_left} + 0.106$ ,  $l2 = \text{sonar\_front} + 0.1414$  με 0.106 και 0.1414 οι αποστάσεις που υπολογίστηκαν στο μέρος Α για να σχηματιστεί τρίγωνο. Το  $l3$  είναι το κομμάτι του τοίχου που κλείνει το τρίγωνο.

Αφού βρούμε το  $dy$  υπολογίζουμε το  $y_{abs}$  που προκύπτει από τις μετρήσεις και είναι ίσο με  $y_{abs} = 2 - dy$

Τέλος, παίρνουμε την περίπτωση να μην βλέπουν τον ίδιο τοίχο τα δύο αυτά sonar. Σε αυτήν την περίπτωση το sonar front έχει εσωτερική γωνία την  $\pi - \theta$  και μετράει τώρα  $dx$  με τον ίδιο τρόπο. Ο έλεγχος για αυτήν την περίπτωση είναι εάν η γωνία  $a$  αποκλίνει κατά παραπάνω από 0.1 από την θεωρητική γωνία  $\theta$  όπου σε αυτήν την περίπτωση έχουμε ένα από τα δυο σόναρ εκτός του εξεταζόμενου τοίχου. Θέλουμε, ωστόσο, επιπλέον να ελέγξουμε εάν το μπροστά σόναρ που μας ενδιαφέρει είναι εκτός το εξεταζόμενου τοίχου. Αυτό γίνεται με την επιπλέον συνθήκη:

$(x > 0 \text{ and } y > 0 \text{ and } \text{angle} > 0) \text{ or } (x > 0 \text{ and } y < 0 \text{ and } \text{angle} > -\pi/2) \text{ or } (x < 0 \text{ and } y < 0 \text{ and } \text{angle} < 0) \text{ or } (x < 0 \text{ and } y > 0 \text{ and } \text{angle} > 0)$ . Ομοίως εκτελούμε για τα υπόλοιπα.





Ακολουθώντας τους παρακάτω τύπους έχουμε τα εξής.

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}_{k+1} \mathbf{r}(k+1) \quad \text{με} \quad \mathbf{r}(k+1) = (\mathbf{z}_{k+1} - h(\hat{\mathbf{x}}(k+1|k)))$$

και

$$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}] \cdot \mathbf{P}(k+1|k) \quad (\Pi.2-10)$$

$$\mathbf{r}(k+1) = \mathbf{z}_{k+1}$$

Οπου  $\mathbf{z} = h(\mathbf{x}, \mathbf{x}_{\text{sensor}}) = (x - x_{\text{sensor}}, y - y_{\text{sensor}}, \text{angle} - \text{angle}_{\text{sensor}})$

Στην πραγματικότητα συγκρίνονται οι απόλυτες τιμές των μεγεθών και πραγματοποιείται η διαφορά τους, δηλαδή  $|x| - |x_{\text{sensor}}|$  κοκ.

Θεωρούμε ως σημείο αναφοράς την θέση του κινητού επομένως η αναμενόμενη τιμή  $h(\hat{\mathbf{x}}(k+1|k)) = 0$  και  $\mathbf{z}_{k+1}$  είναι η διαφορά την εκτίμησης θέσης μας από την οδομετρία με την εκτίμηση της θέσης μας από τους αισθητήρες. Ανάλογα με αυτήν την διαφορά ανανεώνουμε την τιμή της εκτίμησης κατάστασης με βάση το κέρδος  $\mathbf{K}$ .

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}_{k+1} \mathbf{r}(k+1)$$

$$\text{Με } \mathbf{K}_{k+1} = \mathbf{P}(k+1|k) \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}(k+1|k) \mathbf{H}_{k+1}^T + \mathbf{C}_v)$$

$$\text{Όπου } \mathbf{H} = \mathbf{V} \mathbf{h} = [1 \ 1 \ 1] \cdot \mathbf{T}$$

Ως  $C_v$  ορίστηκε διαγώνιος πίνακας. Επειδή οι τιμές που προκύπτουν για τα  $x, y$  είναι γραμμικές ως προς τις μετρήσεις των sonars, προκύπτει:

$C_v = \text{diag}(0.01, 0.01, 0.01)$

Όσον αφορά την υλοποίηση των παραπάνω:

Δημιουργήσαμε κατάλληλο σκριπτάκι python (localization.py) το οποίο διαβάζει μέσω ros τις ενδείξεις από τους σένσορες (που παρέχονται από το simulation περιβάλλον). Εκτελεί τις πράξεις που προαναφέρθηκαν για να κάνει κάθε φορά εκτίμηση του τρέχοντος state.

Προφανώς το random\_walker.cpp είναι υπεύθυνο για το control κομμάτι της κίνησης και επιλέχθηκε να μην αλλάξει.

Σε όλη την παραπάνω διαδικασία υποθέτουμε ότι predicts & updates εκτελούνται με το ίδιο rate. Αυτό φαίνεται να ισχύει από το simulation περιβάλλον. Προφανώς δεν είναι δεσμευτικό και θα μπορούσε να προσαρμοστεί κατάλληλα ώστε να αξιοποιούνται πλήρως όλες οι διαδοχικές μετρήσεις που λαμβάνονται.

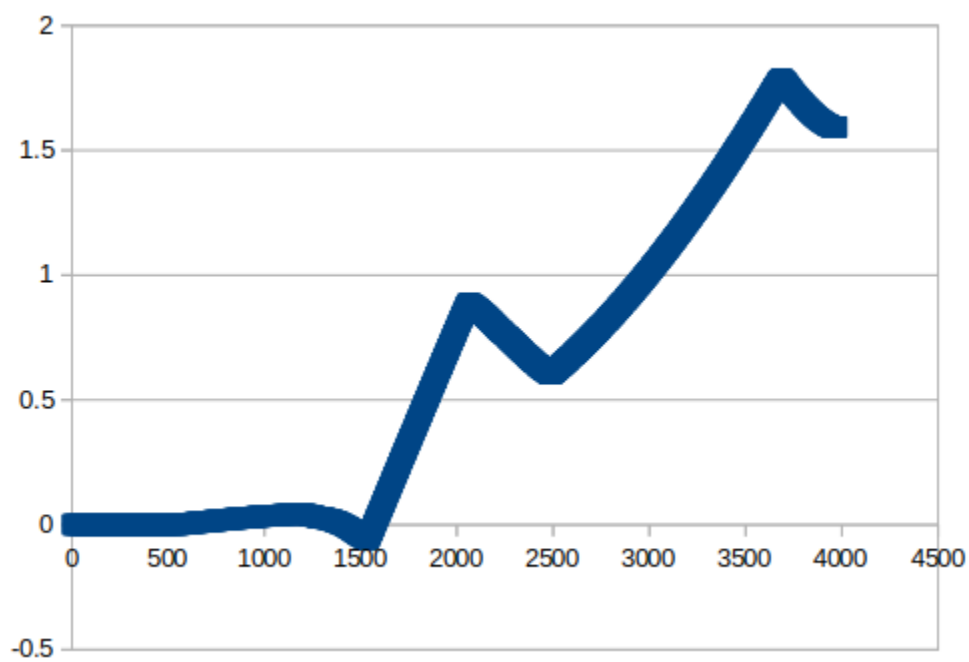
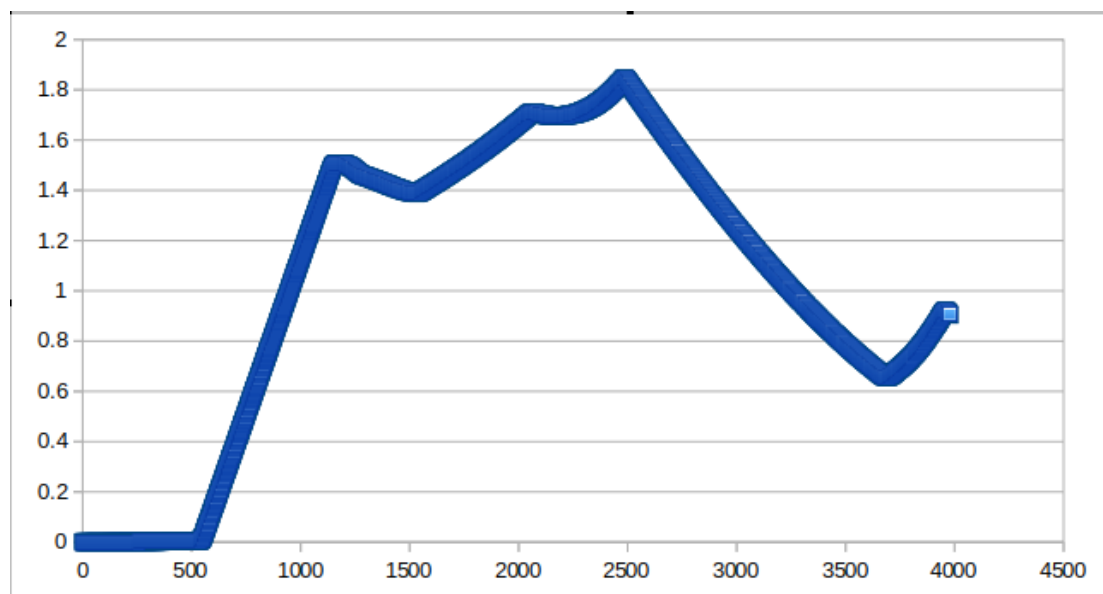
## B. Προσομοίωση - Διαγράμματα

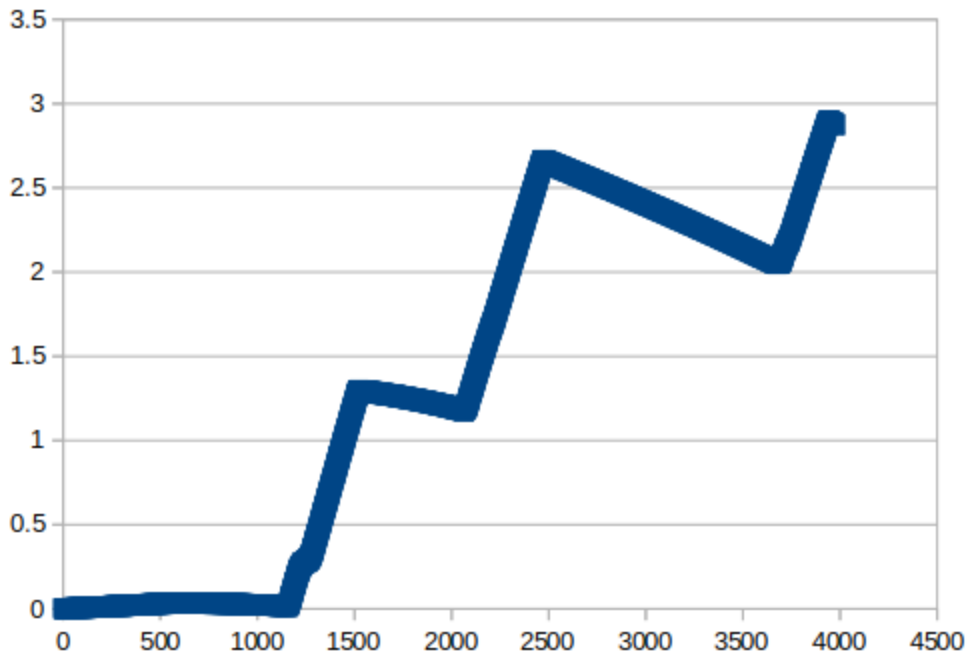
Για την προσομοίωση εφαρμόστηκε  $\theta(0)=0$  για λόγους απλότητας. Ελλείψει χρόνου δεν προσαρμόστηκαν μετά τα αποτελέσματα ξανα για την αρχική γωνία 1.717 που ζητείται, αλλά προφανώς είναι ανάλογα.

Με τη βοήθεια εξωτερικού λογισμικού και για ένα τυχαίο walk τυπώνουμε τα  $x, y, \theta$ . Άξονας  $x$  είναι των δειγμάτων (100Hz).

Με τη σειρά  $x\_position, y\_position, \theta\_angle$ :







Παραμείναμε στην διαισθητική αξιολόγηση, αν και θα είχε πολύ ενδιαφέρον να γινόταν πλήρης ανάλυση με εκτύπωση και άλλων παραμέτρων για να δούμε και ποσοτικά το πόσο σωστά πραγματοποιείται το φιλτράρισμα.