Particle Filter-Based SLAM for Differential Drive Robots

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Βραχωρίτη Αλεξάνδρα (Α.Μ: 1092793), Γιάννη Νικόλαος (Α.Μ: 1092569)

12 Ιουνίου 2025

1 Εισαγωγή

Το Simultaneous Localization And Mapping (SLAM) πρόβλημα ορίζεται ως μια διαδικασία καθορισμού της θέσης του ρομπότ στον χώρο καθώς και της δημιουργίας ενός χάρτη του περιβάλλοντος στο οποίο κινείται. Ένας τρόπος επίλυσης του προβλήματος είναι το Extended Kalman Filter (EKF), που χρησιμοποιεί Gaussian κατανομή πολλαπλών μεταβλητών. Σε αντίθεση με το απλό Kalman Filter (KF), το EKF λειτουργεί και για μη γραμμικές συναρτήσεις των μοντέλων κίνησης και μέτρησης. Το μειονέκτημα αυτής της προσέγγισης είναι η αδυναμία μιας Gaussian κατανομής να περιγράψει τη δημιουργία πολλών διαφορετικών χαρτών, που οφείλεται στην εμφάνιση όμοιων ορόσημων σε διαφορετικά σημεία του περιβάλλοντος. Έχουν προταθεί τρόποι αντιμετώπισης του προβλήματος, ωστόσο δεν παρουσιάζουν ικανοποιητικά αποτελέσματα είτε λόγω κατάρρευσης του αλγορίθμου σε περίπτωση λάθους συσχέτισης δεδομένων, είτε λόγω μεγάλης πολυπλοκότητας, που δεν επιτρέπει την ολοκλήρωση του αλγορίθμου σε πραγματικό χρόνο. Ένας διαφορετικός αλγόριθμος επίλυσης του προβλήματος SLAM είναι ο FastSLAM, που συνδυάζει τα Particle Filters (PFs) για την εκτίμηση της θέσης του ρομπότ, με EKFs για την εκτίμηση της θέσης των ορόσημων (landmarks) στον χάρτη. Η δομή του προβλήματος SLAM επιτρέπει την εκτίμηση της θέσης κάθε ορόσημου στον χάρτη ανεξάρτητα από τα υπόλοιπα, αν είναι γνωστή η διαδρομή του ρομπότ, και αυτό εκμεταλλεύεται ο FastSLAM.

2 Πρόβλημα SLAM

Σε ένα πρόβλημα SLAM με τη χρήση έντροχου ρομπότ διαφορικής οδήγησης (Differential Drive Robot), η αλλαγή της κατάστασής του καθορίζεται από το μοντέλο κίνησης (motion model), ενώ η θέση στην οποία εντοπίζει ορόσημα καθορίζεται από το μοντέλο μέτρησης (measurement model).

$$p(s_t|u_t, s_{t-1}) = h(u_t, s_{t-1}) + \delta_t \tag{1}$$

$$p(z_t|s_t, \theta_{n_t}, n_t) = g(\theta_{n_t}, s_t) + \varepsilon_t \tag{2}$$

Η εξίσωση (1) δηλώνει το μοντέλο κίνησης, η εξίσωση (2) δηλώνει το μοντέλο μέτρησης και οι μεταβλητές που χρησιμοποιούνται κάθε χρονική στιγμή t είναι:

- s_t: η κατάσταση (state) του ρομπότ,
- z_t: η μέτρηση από τον αισθητήρα,
- n_t: η ταυτότητα (ID) του ορόσημου που παρατηρείται,
- θ_{n_t}: το ορόσημο που παρατηρείται,
- u_t: τα σήματα ελέγχου στις ρόδες του ρομπότ,
- δ_t: ο θόρυβος του μοντέλου κίνησης, που δίνεται από την κανονική κατανομή με μέση τιμή 0 και συνδιασπορά P_t,
- ε_t: ο θόρυβος του μοντέλου μέτρησης, που δίνεται από την κανονική κατανομή με μέση τιμή 0 και συνδιασπορά R_t.

Χωρίς βλάβη της γενικότητας, η θεωρητική ανάλυση που γίνεται θεωρεί πως το ρομπότ βλέπει ένα ορόσημο τη φορά. Ωστόσο, η πραγματική υλοποίηση διαχειρίζεται πολλαπλά ορόσημα κάθε χρονική στιγμή. Η πιθανότητα posterior που υπολογίζουν οι περισσότεροι αλγόριθμοι SLAM είναι η

$$p(s_t, \Theta | z^t, u^t, n^t) \tag{3}$$

με τον εκθέτη t να δηλώνει το σύνολο των τιμών των μεταβλητών μέχρι τη χρονική στιγμή t και το Θ να δηλώνει τον χάρτη. Ο αλγόριθμος FastSLAM υποστηρίζει τον υπολογισμό της πιθανότητας posterior

$$p(s^t, \Theta | z^t, u^t, n^t) \tag{4}$$

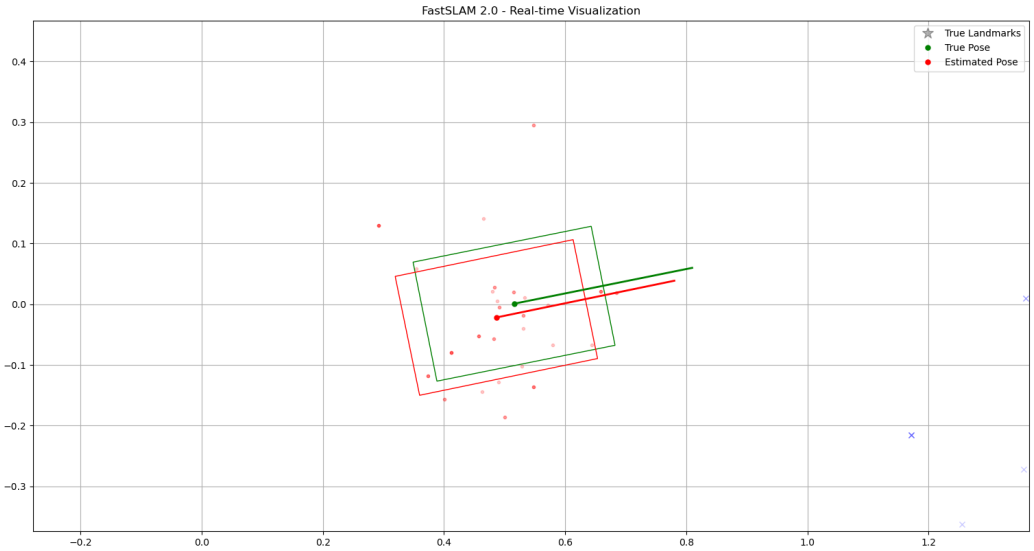
Όπως αναφέρθηκε, όταν είναι γνωστή η διαδρομή που ακολουθεί το ρομπότ, το πρόβλημα SLAM επιτρέπει την εκτίμηση της θέσης κάθε ορόσημου στον χάρτη ανεξάρτητα από τα υπόλοιπα και έτσι η (4) αναλύεται ως:

$$p(s^t, \Theta | z^t, u^t, n^t) = p(s^t | z^t, u^t, n^t) \prod_{n=1}^{N} p(\theta_n | s^t, n^t, z^t) \tag{5}$$

όπου N το πλήθος των ορόσημων. Ο πρώτος όρος $p(s^t | z^t, u^t, n^t)$ υλοποιείται με τη μέθοδο των PFs, ενώ κάθε όρος του γινομένου παραγόντων υπολογίζεται από τα EKFs.

3 Ανάλυση των Particle Filters

Για την υλοποίηση του SLAM με Particle Filters δημιουργούνται πολλαπλά εικονικά αντίγραφα του ρομπότ, τα οποία τοποθετούνται στην αρχική του θέση. Για την οπτικοποίηση της διαδικασίας, θεωρούνται $M = 50$ particles, τοποθετημένα κοντά στην αρχική θέση του ρομπότ.



Σχήμα 1. Στην εικόνα φαίνονται η πραγματική κατάσταση του ρομπότ (με πράσινο), η εκτιμώμενη κατάσταση του ρομπότ (με κόκκινο) και τα particles (με κόκκινες κουκίδες). Στην πραγματικότητα, όλα τα particles την αρχική χρονική στιγμή βρίσκονται στην ίδια θέση με αυτή του ρομπότ, όπως επίσης η κατάσταση του κόκκινου ρομπότ, ταυτίζεται με την κατάσταση του πράσινου ρομπότ.

Αφού εφαρμοστεί ένα σήμα ελέγχου στις ρόδες του ρομπότ, η κατάστασή του θα αλλάξει βάσει του μοντέλου κίνησης. Το ίδιο θα συμβεί και σε κάθε ένα από τα particles. Το μοντέλο κίνησης δεν προσεγγίζεται με γραμμικοποίηση της μη γραμμικής συνάρτησης που το περιγράφει, όπως στη μέθοδο των EKFs, αλλά είναι ακριβές. Συνεχίζοντας, για κάθε particle, αφού ληφθούν μετρήσεις από τον αισθητήρα LiDAR, συγκρίνονται με τις θεωρητικές τιμές των μετρήσεων των particles. Οι θεωρητικές τιμές των μετρήσεων εξάγονται από το αντίστροφο μοντέλο μέτρησης, που επιστρέφει την θέση των ορόσημων στον χάρτη, βάσει της απόστασης κάθε particle από αυτά. Κάθε particle έχει ένα βάρος w_i, το οποίο αναβαθμίζεται σύμφωνα με την παραπάνω διαδικασία εκτίμησης της θέσης του ρομπότ. Αφού καθοριστεί το w_i για κάθε particle, για το νέο σύνολο των particles επιλέγονται τυχαία M particles, με πιθανότητα ανάλογη του βάρους w_i. Η διαδικασία αυτή διαφέρει στο αρχικό βήμα ανάλογα με το αν θα χρησιμοποιηθεί ο αλγόριθμος FastSLAM 1.0 ή FastSLAM 2.0. Ο αλγόριθμος που υλοποιείται είναι ο FastSLAM 2.0 και αναλύεται παρακάτω.

3.1 Αλγόριθμος FastSLAM 2.0

1. Επέκταση του posterior της διαδρομής δειγματοληπτώντας νέες καταστάσεις
Η επόμενη κατάσταση κάθε particle δειγματοληπτείται από την προτεινόμενη κατανομή (proposal distribution). Αυτή υπολογίζεται χρησιμοποιώντας το μοντέλο κίνησης, τις μετρήσεις (z_t) και το ορόσημο (n_t) που θεωρείται ότι φαίνεται κάθε φορά:

$$s_t^{[m]} \sim p(s_t | s^{t-1,[m]}, u^t, z^t, n^t) \tag{6}$$

Το m δηλώνει το particle στο οποίο αναφέρεται ο αλγόριθμος. Η (6) αναλύεται βάσει του θεωρήματος του Bayes, της παραδοχής του Markov και εφαρμόζοντας γραμμικοποίηση της $g(\theta_{n_t}, s_t)$ του μοντέλου μέτρησης με την εφαρμογή του αναπτύγματος Taylor $1^{ου}$ βαθμού, και τελικά αναπαρίσταται από Gaussian κατανομή με μέση τιμή και συνδιασπορά:

$$\Sigma_{s_t}^{[m]} = [G_s^T Q_t^{[m]-1} G_s + P_t^{-1}]^{-1} \tag{7}$$

$$\mu_{s_t}^{[m]} = \Sigma_{s_t}^{[m]} G_s^T Q_t[m] - 1(z_t - \hat{z}_t{}^{[m]}) + \hat{s_t}{}^{[m]} \tag{8}$$

$$Q_t^{[m]} = R_t + G_\theta \Sigma_{n_t,t-1}^{[m]} G_\theta^T \tag{9}$$

με

$$G_\theta = \nabla_{\theta_{n_t}} g(\theta_{n_t}, s_t)|_{s_t = \hat{s_t}{}^{[m]}; \theta_{n_t} = \mu_{n_t,t-1}^{[m]}} \tag{10}$$

$$G_s = \nabla_{s_t} g(\theta_{n_t}, s_t)|_{s_t = \hat{s_t}{}^{[m]}; \theta_{n_t} = \mu_{n_t,t-1}^{[m]}} \tag{11}$$

$$\hat{z_t}{}^{[m]} = g(\mu_{n_t,t-1}^{[m]}, \hat{s_t}{}^{[m]}) \tag{12}$$

$$\hat{s_t}{}^{[m]} = h(s_{t-1}^{[m]}, u_t) \tag{13}$$

Με βάση τα παραπάνω, προσδιορίζεται η ταυτότητα του ορόσημου που είναι πιο πιθανό να έχει παρατηρηθεί και συμβολίζεται με $\hat{n}_t$. Η επιλογή γίνεται βάσει της πιθανότητας:

$$p_{n_t} = |2\pi Q_{t,n_t}^{[m]}|^{-\frac{1}{2}} exp\{-\frac{1}{2}(z_t - g(\mu_{n_t,t-1}^{[m]}, s_{n_t,t}^{[m]}))^T Q_{t,n_t}^{[m]}{}^{-1} (z_t - g(\mu_{n_t,t-1}^{[m]}, s_{n_t,t}^{[m]}))\} \tag{14}$$

και του maximum likelihood estimator. Σε περίπτωση που κανένα από τα p_{n_t} δεν ξεπερνά το p_0 (κάτω όριο), τότε έχει εντοπιστεί νέο ορόσημο. Εδώ καθίσταται εμφανές το πλεονέκτημα του αλγορίθμου FastSLAM 2.0, αφού η συσχέτιση των δεδομένων γίνεται ανά particle. Σε περίπτωση λάθους συσχέτισης δεδομένων, ο αλγόριθμος δεν

καταρρέει, διότι το λάθος δημιουργείται τοπικά σε κάθε particle. Κατά τη διαδικασία της επαναδειγματοληψίας τα particle με λάθος συσχετίσεις δεδομένων θα έχουν πολύ μικρή πιθανότητα να επιλεγούν ξανά.

2. Αναβάθμιση της κατάστασης των ορόσημων που παρατηρήθηκαν - Υπολογισμός των βαρών των particles

Για κάθε ορόσημο του περιβάλλοντος διατηρείται η μεταβλητή τ που καθορίζει πόσο πολύ πιστεύεται ότι ένα ορόσημο πράγματι υπάρχει.

Γενικά, υπάρχουν τρεις περιπτώσεις ελέγχου. Μελετάμε αν το ορόσημο έχει παρατηρηθεί ξανά, αν παρατηρείται για πρώτη φορά, ή αν δεν παρατηρήθηκε, ενώ θα έπρεπε να έχει παρατηρηθεί, λόγω προηγούμενου εντοπισμού του. Έτσι, αποφασίζουμε για την αναβάθμιση των εκτιμώμενων θέσεων των ορόσημων.

a) Ορόσημο που παρατηρείται ξανά:
Αναβαθμίζεται η εκτίμηση της κατάστασής του χρησιμοποιώντας EKF. Γι' αυτό, υπολογίζονται νέα μέση τιμή και συνδιασπορά ως εξής:

$$\mu_{\hat{n}_t,t}^{[m]} = \mu_{\hat{n}_t,t-1}^{[m]} + K_t^{[m]}(z_t - \hat{z}_{t,\hat{n}_t}^{[m]}) \tag{15}$$

$$\Sigma_{\hat{n}_t,t}^{[m]} = (I - K_t^{[m]} G_{\theta,\hat{n}_t}))\Sigma_{\hat{n}_t,t-1}^{[m]} \tag{16}$$

όπου

$$K_t^{[m]} = \Sigma_{\hat{n}_t,t-1}^{[m]} G_{\theta,\hat{n}_t}^T {Q_{t,\hat{n}_t}^{[m]}}^{-1} \tag{17}$$

Το βάρος του particle m τη χρονική στιγμή t, για την $i-$οστή παρατήρηση είναι:

$$w_t^{[m]} = \frac{target\ \ distribution}{proposal\ \ distribution} = |2\pi L_t^{[t]}|^{-\frac{1}{2}} exp\{-\frac{1}{2}(z_t - \hat{z}_{t,\hat{n}_t})^T {L_t^{[t]}}^{-1}(z_t - \hat{z}_{t,\hat{n}_t})\} \tag{18}$$

με τη συνδιασπορά να είναι:

$$L_t^{[t]} = G_{s,\hat{n}_t} P_t G_{s,\hat{n}_t}^T + G_{\theta,\hat{n}_t} \Sigma_{n_t,t-1}^{[m]} G_{\theta,\hat{n}_t}^T + R_t \tag{19}$$

Στην περίπτωση αυτή, η μεταβλητή τ του ορόσημου αυξάνεται κατά ρ^+, που είναι ένας σταθερός αριθμός.

b) Ορόσημο που παρατηρείται πρώτη φορά:
Με τη χρήση μόνο του μοντέλου κίνησης, του σήματος ελέγχου και της προηγούμενης κατάστασης του ρομπότ προσδιορίζεται η επόμενη κατάσταση του ρομπότ και η αρχικοποίηση του EKF για το ορόσημο γίνεται βάση της μέσης τιμής και της συνδιασποράς:

$$\mu_{n,t}^{[m]} = g^{-1}(z_t, s_{n,t}^{[m]}) \tag{20}$$

$$\Sigma_{n,t}^{[m]} = (G_{\theta,n} R_t^{-1} G_{\theta,n}^T)^{-1} \tag{21}$$

Το βάρος του particle για την $i-$οστή παρατήρηση δίνεται από το p_0. Στην περίπτωση αυτή, η μεταβλητή τ του ορόσημου αρχικοποιείται στην τιμή ρ^+.

c) Ορόσημο που δεν παρατηρήθηκε, ενώ θα έπρεπε να έχει παρατηρηθεί, λόγω προηγούμενου εντοπισμού του:
Εδώ εννοείται ότι το ορόσημο είναι εντός του εύρους εντοπισμού του LiDAR, αλλά για κάποιο λόγο δεν παρατηρήθηκε. Στην περίπτωση αυτή, η μεταβλητή τ του ορόσημου μειώνεται κατά ρ^-, που είναι ένας σταθερός αριθμός. Αν η τιμή του τ γίνει αρνητική, τότε το ορόσημο διαγράφεται οριστικά από την λίστα του particle.

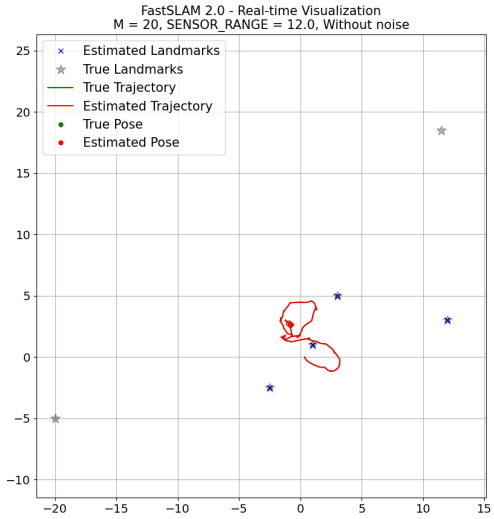
Αφού ολοκληρωθούν οι παραπάνω διαδικασίες για κάθε μέτρηση που λαμβάνεται τη χρονική στιγμή t για κάθε ένα particle, αυτό προστίθεται στο νέο σύνολο από particles με την κατάσταση του ρομπότ να ισούται με τον μέσο όρο των προβλεπόμενων καταστάσεων που έχουν δημιουργηθεί σε κάθε παρατήρηση και βάρος ίσο με τον μέσο όρο των βαρών κάθε παρατήρησης.

3. Επαναδειγματοληψία
Για το νέο σύνολο των particles επιλέγονται τυχαία M particles, με πιθανότητα ανάλογη του βάρους τους w_i, τα οποία χρησιμοποιούνται στο επόμενο βήμα της προσομοίωσης.

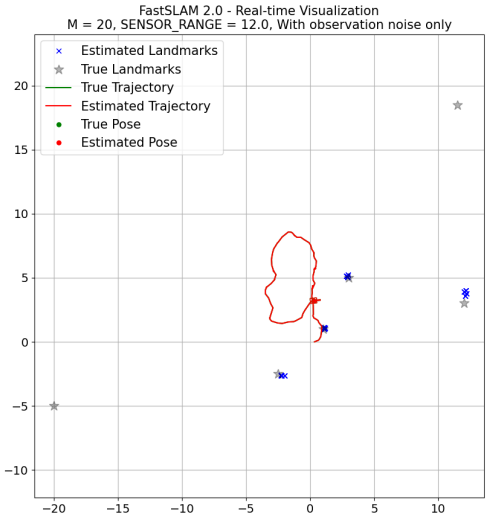
4 Πειραματικά αποτελέσματα προσομοίωσης με Python

Κάνοντας προσομοιώσεις για διαφορετικές τιμές θορύβου στο μοντέλο κίνησης και παρατήρησης, του αριθμού των particles και του εύρους του LiDAR, προκύπτουν τα ακόλουθα:

1. Προσομοιώσεις με μεταβλητό θόρυβο



Σχήμα 2.1. Προσομοίωση με 20 particles, εύρος LiDAR 12 και καθόλου θόρυβο.

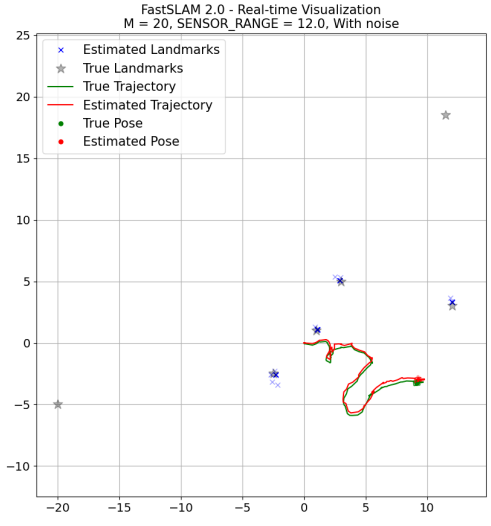


Σχήμα 2.2. Προσομοίωση με 20 particles, εύρος LiDAR 12 και θόρυβο μόνο στις μετρήσεις.

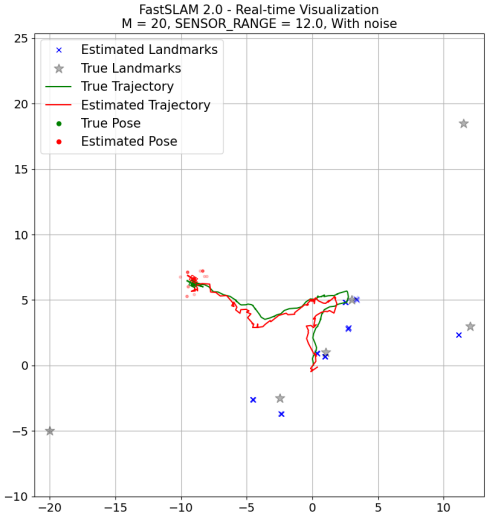
Παρατηρήσεις

Στο Σχήμα 2.1, που δεν εμφανίζεται καθόλου θόρυβος η εκτίμηση της θέσης των ορόσημων είναι ακριβής, το ίδιο και η εκτίμηση της θέσης του ρομπότ. Προσθέτοντας θόρυβο στις μετρήσεις του LiDAR η εκτίμηση της θέσης των ορόσημων δεν είναι πια ακριβής, αλλά προσεγγίζει την πραγματικότητα. Προχωρώντας, στο Σχήμα 2.4, σε αντίθεση με το Σχήμα 2.3, η προσθήκη αρκετά μεγάλου θορύβου προκαλεί αστοχίες τόσο στην εκτίμηση της θέσης των ορόσημων, όσο και στην εκτίμηση της θέσης του ρομπότ.

Σημείωση: Στις επόμενες προσομοιώσεις οι τιμές του θορύβου θα διατηρηθούν σε ένα μέσο επίπεδο, δηλαδή ούτε πολύ μεγάλες, ούτε πολύ μικρές, όπως έγινε στην προσομοίωση του Σχήματος 2.3, ώστε να εξάγουμε συμπεράσματα και για τις υπόλοιπες παραμέτρους.

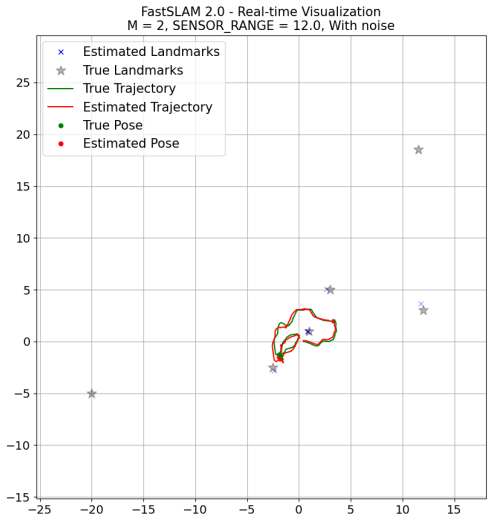


Σχήμα 2.3. Προσομοίωση με 20 particles, εύρος LiDAR 12 και κανονικό θόρυβο.

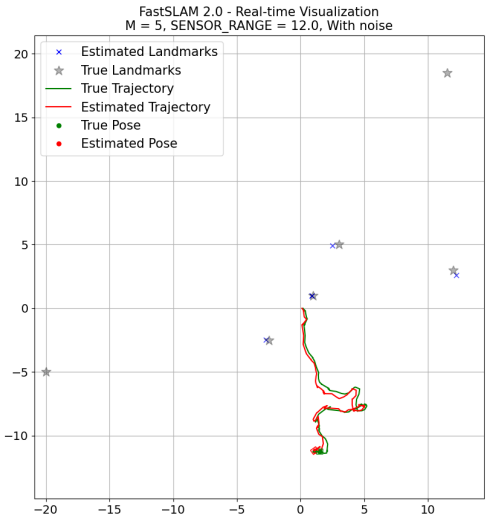


Σχήμα 2.4. Προσομοίωση με 20 particles, εύρος LiDAR 12 και πολύ θόρυβο.

2. Προσομοιώσεις με διαφορετικό αριθμό particles



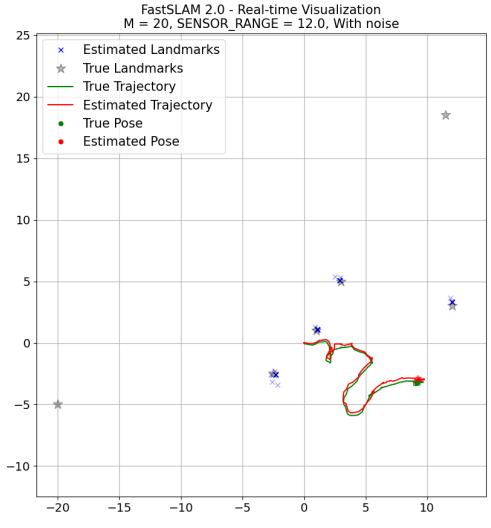
Σχήμα 3.1. Προσομοίωση με 2 particles, εύρος LiDAR 12 και κανονικό θόρυβο.



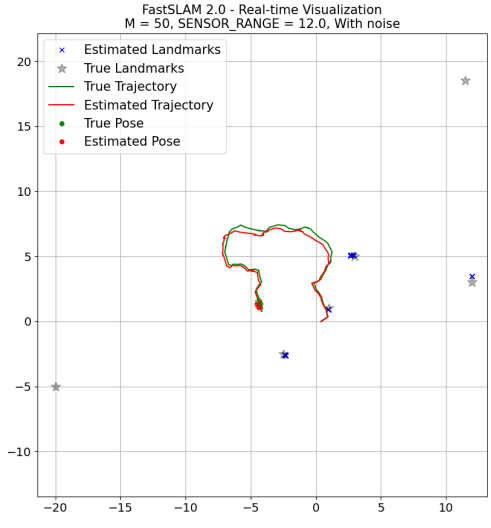
Σχήμα 3.2. Προσομοίωση με 5 particles, εύρος LiDAR 12 και κανονικό θόρυβο.

Παρατηρήσεις

Φαίνεται να μην υπάρχει μεγάλη διαφορά μεταξύ των Σχημάτων 3.1, 3.2, 3.3 και 3.4. Ωστόσο, όσο ο αριθμός των particles αυξάνεται τόσο πιο εύκολο θα είναι να αφαιρεθούν particles με λανθασμένες συσχετίσεις δεδομένων κατά την επαναδειγματοληψία.

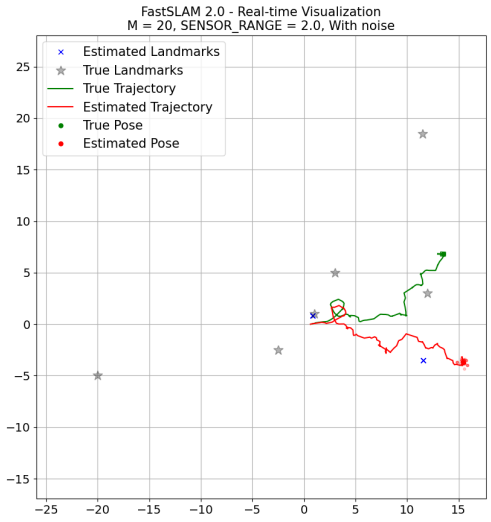


Σχήμα 3.3. Προσομοίωση με 20 particles, εύρος LiDAR 12 και κανονικό θόρυβο.

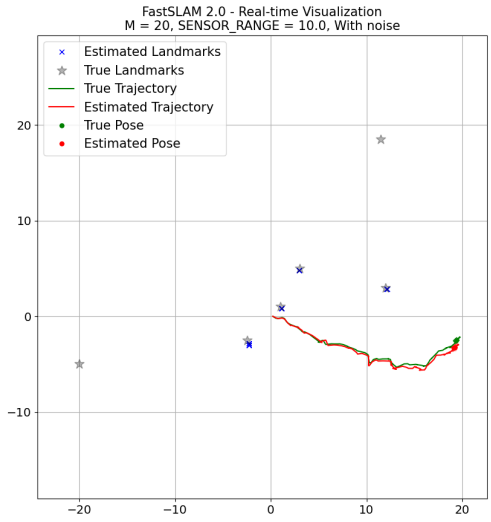


Σχήμα 3.4. Προσομοίωση με 50 particles, εύρος LiDAR 12 και κανονικό θόρυβο.

3. Προσομοιώσεις με διαφορετικό εύρος του LiDAR



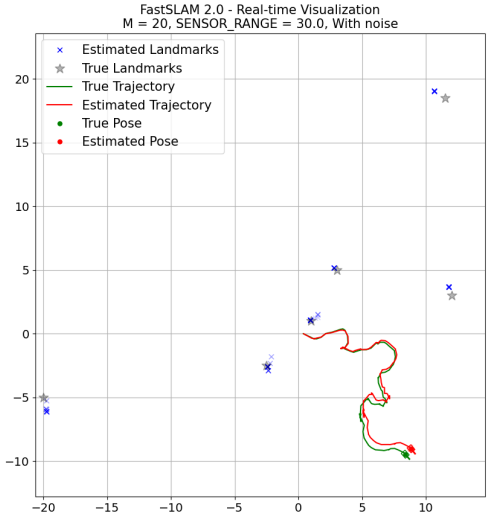
Σχήμα 4.1. Προσομοίωση με 20 particles, εύρος LiDAR 2 και κανονικό θόρυβο.



Σχήμα 4.2. Προσομοίωση με 20 particles, εύρος LiDAR 10 και κανονικό θόρυβο.

Παρατηρήσεις

Για να μπορέσει ο αλγόριθμος να λειτουργήσει σωστά και να κάνει ορθή εκτίμηση της θέσης του ρομπότ και του χάρτη στην περίπτωση άγνωστων αντιστοιχήσεων με ορόσημα, απαιτείται ο χάρτης που χαρτογραφεί να έχει μια πληθώρα ορόσημων. Αυτό επιβεβαιώνεται και από τα σχήματα 4.1, 4.2 και 4.3. Στο Σχήμα 4.1 το εύρος του LiDAR είναι πολύ μικρό και αυτό μεταφράζεται σε ένα περιβάλλον με ορόσημα πολύ μακριά το ένα από το άλλο, με αποτέλεσμα ο αλγόριθμος να μην λειτουργεί ικανοποιητικά.



Σχήμα 4.3. Προσομοίωση με 20 particles, εύρος LiDAR 30 και κανονικό θόρυβο.

5 References

[1] Sebastian Thrun, Michael Montemerlo, Daphne Koller, Ben Wegbreit, Juan Nieto, and Eduardo Nebot. FastSLAM: An Efficient Solution to the Simultaneous Localization And Mapping Problem with Unknown Data Association.

[2] Clear and Concise Particle Filter Tutorial with Python Implementation - Part 1: Problem Formulation - Part 2: Derivation of Particle Filter Algorithm From Scratch - Part 3: Python Implementation of Particle Filter Algorithm.