



Autonomous Robotic Vehicles

Hellenic Mediterranean University

Lecture 12

Dr. Alina Eqtami

Μετά το σημερινό μάθημα θα μπορείτε:

- Να εξηγείτε τη διαφορά ανάμεσα σε localization και mapping.
- Να περιγράφετε το πρόβλημα του SLAM και γιατί είναι δύσκολο.
- Να κατανοείτε την έννοια της αβεβαιότητας και των **συσχετίσεων** μεταξύ ρομπότ και landmarks.
- Να εξηγείτε τα βασικά βήματα του EKF-SLAM:
 - state vector & covariance,
 - motion model και F_k ,
 - measurement model και H_k ,
 - prediction & update steps,
 - εισαγωγή νέων landmarks (state augmentation).
- Να εφαρμόζετε τα παραπάνω σε ένα απλό αριθμητικό παράδειγμα EKF-SLAM.

Ορισμός

Localization είναι η διαδικασία κατά την οποία το ρομπότ προσδιορίζει τη θέση και τον προσανατολισμό του μέσα σε έναν **γνωστό χάρτη**.

- Προϋποθέτει προϋπάρχον περιβάλλον.
- Το ρομπότ εκτιμά την τροχιά (path) που διανύει.
- Απαιτεί συγχώνευση αισθητήρων (sensor fusion).

Θυμηθείτε:

1) Prediction (Odometry/Dead Reckoning)

↓ Robot motion model

2) Sensor Measurements

↓ Corrections

3) Fusion (KF / MCL)

Ορισμός

Mapping είναι η διαδικασία δημιουργίας χάρτη του περιβάλλοντος, υποθέτοντας ότι η πραγματική τροχιά του ρομπότ είναι γνωστή.

- Κατασκευή γεωμετρικής αναπαράστασης χώρου.
- Ανίχνευση features (γωνίες, ευθείες, επίπεδα, αντικείμενα).
- Απαιτείται υψηλή ακρίβεια στην εκτίμηση της κίνησης του ρομπότ.

Mapping = Ανίχνευση features + Τοποθέτησή τους στον χάρτη με βάση την εκτιμ

Βασική Ιδέα

Το SLAM θέλει να κάνει **ταυτόχρονα**:

- 1 Localization (εκτίμηση θέσης)
- 2 Mapping (κατασκευή χάρτη)

μόνο με χρήση των **on-board αισθητήρων** του ρομπότ.

Χρησιμοποιούμενοι Αισθητήρες

- Οδομετρία → displacement και ταχύτητες.
- Laser scanners → ευθείες, γωνίες, επιφάνειες.
- Ultrasonic sensors → αποστάσεις σε κοντινά εμπόδια.
- Cameras → φωτογραφικά features (visual features).

Πηγές Δυσκολίας

Το SLAM είναι δύσκολο πρόβλημα επειδή υπάρχει θόρυβος:

- Στην εκτίμηση της κίνησης (path) λόγω ολισθήσεων, σφαλμάτων οδομετρίας.
- Στα **features** του περιβάλλοντος: γωνίες, γραμμές, επιφάνειες, αντικείμενα.
- Στην αστάθεια του περιβάλλοντος (ανοιχτές/κλειστές πόρτες, μεταβολή επίπλων).

Το ρομπότ πρέπει να ξεχωρίσει **πραγματικές αλλαγές** από **μετρητικό θόρυβο**.

Γιατί όχι χειροποίητοι χάρτες:

- Οι κλασικές μέθοδοι εντοπισμού απαιτούν προϋπάρχον χάρτη.
- Η χειροκίνητη κατασκευή χαρτών είναι δύσκολη, ακριβή και χρονοβόρα.
- Τα περιβάλλοντα αλλάζουν (π.χ. επίπλωση σπιτιού).
- Οι άνθρωποι αντιλαμβάνονται τον χώρο διαφορετικά → υποκειμενικοί χάρτες.

Βασική Ιδέα

Αφού το ρομπότ έχει αισθητήρες που ανιχνεύουν το περιβάλλον, οφείλει να είναι σε θέση να **δημιουργεί μόνο του** τον χάρτη.

- Αυτόνομη εξερεύνηση.
- Αναγνώριση θέσεων (place recognition).
- Συνεχής ενημέρωση και βελτίωση του χάρτη.
- Θεμέλιο της πραγματικής ρομποτικής αυτονομίας.

Τελικός Στόχος

Ξεκινώντας από άγνωστη αρχική θέση, το ρομπότ πρέπει:

- 1 Να εξερευνήσει το περιβάλλον.
- 2 Να αναγνωρίσει χαρακτηριστικά.
- 3 Να χτίσει τον χάρτη.
- 4 Να εντοπίζει συνεχώς τη θέση του πάνω στον χάρτη.

Αυτή είναι η ουσία του **Simultaneous Localization and Mapping (SLAM)**.

- Αρχικά: μηδενική αβεβαιότητα στη θέση.
- Παρατήρηση πρώτου feature → χαρτογράφηση με αβεβαιότητα (λόγω αισθητήρα).
- Κίνηση ρομπότ → αύξηση αβεβαιότητας θέσης (odometry error).
- Νέες παρατηρήσεις → αβεβαιότητα = αισθητήρες + αβεβαιότητα θέσης.
- Ο χάρτης γίνεται **συσχετισμένος** με την εκτίμηση θέσης.

Τι συμβαίνει;

- Αν ο χάρτης είναι αβέβαιος → η θέση του ρομπότ γίνεται αβέβαιη.
- Αν η θέση του ρομπότ είναι αβέβαιη → τα νέα features χαρτογραφούνται με μεγαλύτερη αβεβαιότητα.

Συμπέρασμα

Η αβεβαιότητα του ρομπότ και του χάρτη είναι **αλληλένδετες**.

Τι είναι το Loop Closure;

- Το ρομπότ αναγνωρίζει ένα feature που έχει ξαναδεί.
- Η αβεβαιότητα θέσης **μειώνεται σημαντικά**.
- Ο χάρτης 'διορθώνεται' και βελτιώνεται παγκόσμια.

Γιατί είναι κρίσιμο:

Το loop closure σταθεροποιεί τον χάρτη και μειώνει το drift.

The Chicken-and-Egg Problem

Περιγραφή

- Για localization πρέπει να γνωρίζουμε πού βρίσκονται τα features.
- Για το mapping πρέπει να ξέρουμε πού βρίσκεται το ρομπότ.

SLAM = Τα λύνει και τα δύο ταυτόχρονα

Αυτός ο αλληλοεξαρτώμενος χαρακτήρας είναι η καρδιά του SLAM.

Βήμα 1: Αρχική θέση και πρώτη παρατήρηση

- Το ρομπότ ξεκινά με μηδενική αβεβαιότητα στη θέση του.
- Παρατηρεί ένα χαρακτηριστικό (feature) στο περιβάλλον.
- Το feature χαρτογραφείται με αβεβαιότητα από το sensor model.

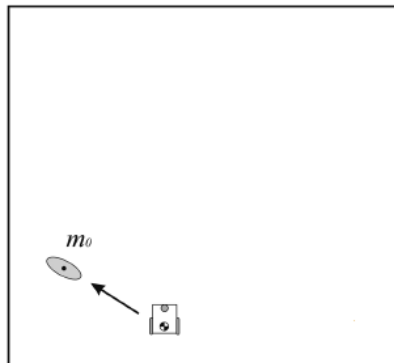


Image source: Siegwart et al., Introduction to
Autonomous Mobile Robots

Βήμα 2: Κίνηση ρομπότ

- Το ρομπότ κινείται.
- Η αβεβαιότητα στη θέση του αυξάνεται λόγω σφαλμάτων οδομετρίας.
- Το drift συσσωρεύεται.

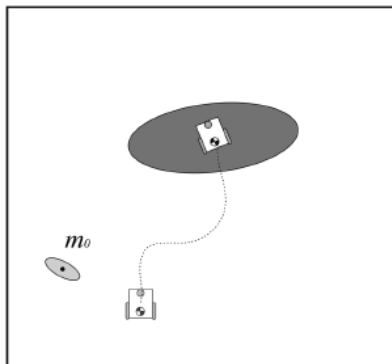


Image source: Siegwart et al., Introduction to
Autonomous Mobile Robots

Βήμα 3: Νέα features με μεγαλύτερη αβεβαιότητα

- Το ρομπότ, απο τη νέα θέση, παρατηρεί δύο νέα features.
- Η αβεβαιότητα προέρχεται από:
 - σφάλμα αισθητήρα
 - αβεβαιότητα θέσης του ρομπότ
- Ο χάρτης γίνεται συσχετισμένος με το localization.

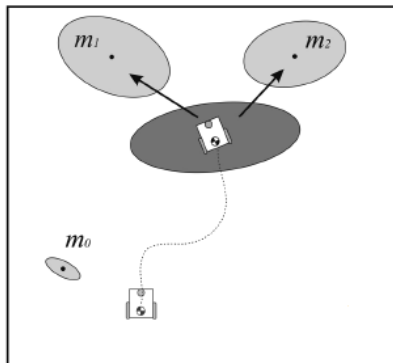


Image source: Siegwart et al., Introduction to
Autonomous Mobile Robots

Βήμα 4: Συσχέτιση χάρτη-θέσης

- Η αβεβαιότητα της θέσης του ρομπότ περνά στα features.
- Τα αβέβαια features κάνουν την εκτίμηση θέσης ακόμη πιο αβέβαιη.
- Χάρτης και θέση γίνονται αλληλεξαρτώμενα.

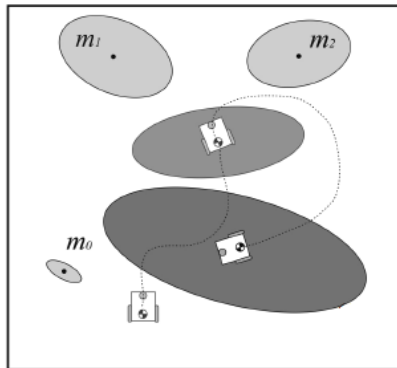


Image source: Siegwart et al., Introduction to
Autonomous Mobile Robots

Βήμα 5: Loop Closure

- Το ρομπότ επαναπαρατηρεί ένα γνωστό feature.
- Η αβεβαιότητα θέσης μειώνεται απότομα.
- Ο χάρτης βελτιώνεται παγκόσμια.
- Μειώνεται η αβεβαιότητα όλων των προηγούμενων θέσεων και features.

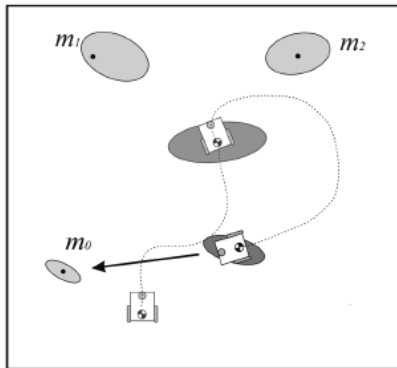


Image source: Siegwart et al., Introduction to
Autonomous Mobile Robots

Κύρια Ιδέα

- Ο χάρτης που δημιουργεί το ρομπότ **σχετίζεται** με το σφάλμα στην εκτίμηση της θέσης του.
- Το ρομπότ ενημερώνει τη θέση του μέσω **observations** από features που δεν είναι πλήρως γνωστά.
- Άρα, η εκτίμηση θέσης του ρομπότ **συσχετίζεται** με την εκτίμηση θέσης των features.
- Για να μειωθεί η αβεβαιότητα, το ρομπότ πρέπει να παρατηρεί features με όσο το δυνατόν πιο **γνωστή** θέση.
- Τέτοια features είναι εκείνα που έχει ξαναδεί → **loop closure detection**.
- Όταν ανιχνευθεί loop closure:
 - μειώνεται η αβεβαιότητα της θέσης του ρομπότ,
 - βελτιώνεται ο χάρτης,
 - μειώνονται οι αβεβαιότητες όλων των σχετικών features και προηγούμενων poses.

Περιγραφή Βημάτων

Σε αυτή την ενότητα θα εξηγήσουμε:

- 1 Το **διάνυσμα κατάστασης** (state vector) του SLAM.
- 2 Τον **πίνακα συσχέτισης αβεβαιότητας** P .
- 3 Το **motion model** και την Jacobian του (F_k).
- 4 Το **observation model** και την Jacobian του (H_k).
- 5 Το EKF **prediction step**.
- 6 Το EKF **correction (update) step**.
- 7 Την εισαγωγή **νέων landmarks** (state augmentation).
- 8 Πώς όλα αυτά συνδέονται στη συνολική διαδικασία EKF-SLAM.

State Vector

$$\mathbf{x}_R = [x \quad y \quad \theta]^T$$

Κάθε landmark:

$$\mathbf{m}_i = [m_{ix}, \quad m_{iy}]^T$$

Πλήρες state:

$$\mathbf{x} = [\mathbf{x}_R \quad \mathbf{m}_1 \quad \dots \quad \mathbf{m}_N]^T$$

Covariance

$$\mathbf{P} = \text{cov}(\mathbf{x}) \quad \text{με διαστάσεις} \quad \mathbf{P} \in \mathbb{R}^{(3+2N) \times (3+2N)}$$

όπου N είναι το πλήθος των landmarks.

Ο πίνακας συνεμφάνισης P περιέχει:

- την αβεβαιότητα του ρομπότ,
- την αβεβαιότητα των landmarks,
- τις cross-covariances μεταξύ ρομπότ–landmarks και μεταξύ διαφορετικών landmarks.

Γενική Δομή του P

$$P = \begin{bmatrix} P_{RR} & P_{Rm_1} & \dots & P_{Rm_N} \\ P_{m_1R} & P_{m_1m_1} & \dots & P_{m_1m_N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m_NR} & P_{m_Nm_1} & \dots & P_{m_Nm_N} \end{bmatrix}$$

- $P_{RR} \in \mathbb{R}^{3 \times 3}$: αβεβαιότητα ρομπότ,
- $P_{m_i m_i} \in \mathbb{R}^{2 \times 2}$: αβεβαιότητα landmark,
- P_{Rm_i} : συσχέτιση ρομπότ–landmark,
- $P_{m_i m_j}$: συσχέτιση μεταξύ landmarks.

Αρχική χρονική στιγμή

Συνήθως ξεκινάμε με διαγώνιο covariance:

$$P_0 = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_{m_1,x}^2, \sigma_{m_1,y}^2, \dots)$$

Αυτό σημαίνει:

$$P_{Rm_i} = 0, \quad P_{m_i,m_j} = 0, \quad i \neq j$$

δηλαδή **καμία αρχική συσχέτιση**.

Πώς δημιουργούνται cross-covariances;

Καθώς το ρομπότ:

- κινείται,
- κάνει μετρήσεις από το ίδιο sensor,
- συνδυάζει πληροφορία μέσω του EKF,

Τα δεδομένα “δένουν” και οι cross-covariances γίνονται **μη μηδενικές**.

Unicycle Model

$$x_k = x_{k-1} + v_k \Delta t \cos(\theta_{k-1})$$

$$y_k = y_{k-1} + v_k \Delta t \sin(\theta_{k-1})$$

$$\theta_k = \theta_{k-1} + \omega_k \Delta t$$

$$x_{R,k} = f(x_{R,k-1}, u_k) + w_k$$

Σημείωση

Τα landmarks είναι στατικά:

$$m_{i,k} = m_{i,k-1}$$

Jacobian ως προς το ρομπότ

$$F_x = \begin{bmatrix} 1 & 0 & -v_k \Delta t \sin(\theta_{k-1}) \\ 0 & 1 & v_k \Delta t \cos(\theta_{k-1}) \\ 0 & 0 & 1 \end{bmatrix}$$

Πλήρης Jacobian

$$F_k = \begin{bmatrix} F_x & 0 \\ 0 & I_{2N} \end{bmatrix}$$

Μόνο η pose αλλάζει. Ο χάρτης μένει σταθερός.

Note on N

Το N είναι ο αριθμός των landmarks στον χάρτη. Κάθε landmark έχει δύο συντεταγμένες (m_{ix}, m_{iy}) , άρα ο χάρτης έχει συνολικά $2N$ μεταβλητές. Γι' αυτό στο prediction step χρησιμοποιούμε τον πίνακα I_{2N} , καθώς τα landmarks δεν μεταβάλλονται από το μοντέλο κίνησης του ρομπότ.

State Prediction

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

$$\hat{m}_{i,k|k-1} = \hat{m}_{i,k-1|k-1}$$

Οι εκτιμήσεις στο $k|k-1$ είναι η **πρόβλεψη πριν τη μέτρηση**. Το ρομπότ κινείται, ενώ τα landmarks παραμένουν σταθερά.

Covariance Prediction

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

Ο πίνακας F_k γραμμικοποιεί το μοντέλο κίνησης και η Q_k είναι η αβεβαιότητα του ελέγχου.

Note

Ο Q_k εκφράζει θόρυβο/σφάλμα στην κίνηση του ρομπότ (π.χ. σφάλμα στα v, ω , ολίσθηση, μη ιδανικοί κινητήρες).

Range-Bearing Measurement Model

$$r = \sqrt{(m_x - x)^2 + (m_y - y)^2}$$

$$\phi = \text{atan2}(m_y - y, m_x - x) - \theta$$

όπου:

- (x, y, θ) : pose ρομπότ,
- (m_x, m_y) : θέση του landmark i ,
- r : απόσταση ρομπότ–(landmark range),
- ϕ : σχετική γωνία ως προς τη διεύθυνση θ (bearing).

Measurement Vector

$$z_k^i = \begin{bmatrix} r \\ \phi \end{bmatrix}$$

Αυτό είναι το διάνυσμα μέτρησης του αισθητήρα (π.χ. lidar, radar, camera) για το landmark i στον χρόνο k .

Measurement Model

$$z_k^i = h(x_R, m_i) + v_k$$

Το $h(\cdot)$ δίνει την **ιδανική** (χωρίς θόρυβο) μέτρηση, ενώ $v_k \sim \mathcal{N}(0, R_k)$ είναι ο **θόρυβος της μέτρησης** (σφάλμα του αισθητήρα).

Ορισμοί

Για ρομπότ στη θέση (x, y, θ) και landmark στη θέση (m_x, m_y) :

$$\Delta x = m_x - x, \quad \Delta y = m_y - y$$

$$q = \Delta x^2 + \Delta y^2, \quad r = \sqrt{q}$$

- Το r είναι η **απόσταση** ρομπότ–landmark.
- Η γωνία $\phi = \text{atan2}(\Delta y, \Delta x) - \theta$ είναι το bearing στο local frame του ρομπότ.
- Ο EKF χρειάζεται τις παραγώγους του (r, ϕ) ως προς (x, y, θ, m_x, m_y) για την γραμμικοποίηση.

Jacobian ως προς pose ρομπότ (x, y, θ)

$$H_{xR} = \frac{\partial(r, \phi)}{\partial(x, y, \theta)} = \begin{bmatrix} -\Delta x/r & -\Delta y/r & 0 \\ \Delta y/q & -\Delta x/q & -1 \end{bmatrix}$$

Πρώτη γραμμή: παράγωγοι του r . Δεύτερη γραμμή: παράγωγοι του ϕ .

Jacobian ως προς landmark (m_x, m_y)

$$H_{m_i} = \frac{\partial(r, \phi)}{\partial(m_x, m_y)} = \begin{bmatrix} \Delta x/r & \Delta y/r \\ -\Delta y/q & \Delta x/q \end{bmatrix}$$

Ο πίνακας είναι ίδιος με του ρομπότ αλλά με τα κατάλληλα πρόσημα, καθώς τώρα οι παράγωγοι λαμβάνονται ως προς το landmark.

Innovation

$$y_k = z_k - h(\hat{x}_{k|k-1})$$

Η **innovation** (ή **measurement residual**) είναι η διαφορά μεταξύ της πραγματικής μέτρησης και της **αναμενόμενης** μέτρησης με βάση την πρόβλεψη.

Innovation Covariance

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$

Η S_k εκφράζει την **αβεβαιότητα της innovation**. Συνδυάζει:

- την αβεβαιότητα της πρόβλεψης ($P_{k|k-1}$),
- την αβεβαιότητα του αισθητήρα (R_k).

Ορισμός

$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$

Το **Kalman Gain** καθορίζει πόσο βάρος δίνουμε:

στην πρόβλεψη vs. στη μέτρηση.

Διαισθητικά

- Αν η μέτρηση είναι ακριβής (R_k μικρό) $\rightarrow S_k^{-1}$ μεγάλο $\rightarrow K_k$ μεγάλο \rightarrow “εμπιστευόμαστε” τη μέτρηση.
- Αν η πρόβλεψη είναι ακριβής ($P_{k|k-1}$ μικρό) $\rightarrow K_k$ μικρό \rightarrow κάνουμε μικρή διόρθωση.
- Αν υπάρχει μεγάλη αβεβαιότητα \rightarrow το K_k αυξάνεται \rightarrow η μέτρηση τραβάει το state προς τη σωστή κατεύθυνση.

State Update

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k$$

Η μέτρηση διορθώνει την πρόβλεψη. Η νέα εκτίμηση προκύπτει προσθέτοντας το innovation σταθμισμένο από το Kalman Gain.

Covariance Update

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

Η αβεβαιότητα μειώνεται μετά την ενσωμάτωση της μέτρησης. Το $(I - K_k H_k)$ “συσφίγγει” την χαρτογράφηση σύμφωνα με την πληροφορία της μέτρησης.

Note

Η ενημέρωση εφαρμόζεται μόνο στα στοιχεία του state που επηρεάζονται από το αντίστοιχο H_k (στο ρομπότ & observed landmarks).

Από τη μέτρηση στο global frame

Έστω νέα μέτρηση:

$$z_k^i = [r, \phi]^T$$

και εκτίμηση θέσης ρομπότ:

$$\hat{x}_{R,k|k} = \begin{bmatrix} \hat{x}_{k|k} \\ \hat{y}_{k|k} \\ \hat{\theta}_{k|k} \end{bmatrix}.$$

Η θέση του νέου landmark στο **global frame** υπολογίζεται ως:

$$\hat{m}_{i,k|k} = \begin{bmatrix} \hat{x}_{k|k} + r \cos(\phi + \hat{\theta}_{k|k}) \\ \hat{y}_{k|k} + r \sin(\phi + \hat{\theta}_{k|k}) \end{bmatrix}.$$

Δηλαδή μετασχηματίζουμε το range-bearing (τοπικό σύστημα) σε παγκόσμιες συντεταγμένες.

State Augmentation

Προσθέτουμε το νέο landmark στο state:

$$\hat{\mathbf{x}}_{k|k}^{new} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_{i,k|k} \end{bmatrix}.$$

Covariance Augmentation

Επεκτείνουμε το covariance:

$$\mathbf{P}_{k|k}^{new} = \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{P}_{xR,m_i} \\ \mathbf{P}_{m_i,xR} & \mathbf{P}_{m_i,m_i} \end{bmatrix}$$

όπου:

$$\mathbf{P}_{m_i,m_i} = \mathbf{J}_m \mathbf{R}_k \mathbf{J}_m^T, \quad \mathbf{P}_{xR,m_i} = \mathbf{J}_{xm} \mathbf{R}_k \mathbf{J}_m^T.$$

Τα \mathbf{J}_{xm} και \mathbf{J}_m είναι οι Jacobians του μετασχηματισμού από (r, ϕ) σε (m_x, m_y) .

Κύκλος ΕΚΦ-ΣΛΑΜ

- 1 **Prediction**: ενημέρωση pose και covariance.
- 2 **Observation**: μετρήσεις από features.
- 3 **Update**: διόρθωση state και covariance.
- 4 **Augmentation**: εισαγωγή νέων landmarks.

Σχόλιο

Η **συσχέτιση** μεταξύ ρομπότ και landmarks κωδικοποιείται στα cross-covariances.