

Implementation and performance evaluation of UKF for Simultaneous Localization and Mapping

G. F. Perez Paina C. J. Paz M. Baudino A. Delfino E. A. Destéfani

Centro de Investigación en Informática para la Ingeniería
Universidad Tecnológica Nacional - Facultad Regional Córdoba

<http://cii.frc.utn.edu.ar>

Córdoba, Argentina



VII Jornadas Argentinas de Robótica
21, 22 y 23 de noviembre de 2012

- 1 Introducción y objetivos
- 2 Localización y mapeo simultáneos, SLAM
 - Filtro de Kalman para la estimación del SLAM
 - Modelo de movimiento y medición
- 3 Estimación del SLAM
 - Extended Kalman Filter (EKF)
 - Unscented Kalman Filter (UKF)
- 4 Implementación y resultados
 - Algoritmos
 - Resultados
- 5 Conclusiones y trabajos futuros

Introducción y objetivos

Localización y mapeo simultáneos (SLAM)

- El SLAM presenta una solución conjunta a dos de los problemas fundamentales a resolver para lograr robots autónomos: la localización y el mapeo.
- El problema del SLAM se resuelve generalmente por medio de filtrado bayesiano (KF, filtro de partículas).
- Una de las técnicas más utilizada el filtro de Kalman Extendido (EKF-SLAM).

Introducción y objetivos

Localización y mapeo simultáneos (SLAM)

- El SLAM presenta una solución conjunta a dos de los problemas fundamentales a resolver para lograr robots autónomos: la localización y el mapeo.
- El problema del SLAM se resuelve generalmente por medio de filtrado bayesiano (KF, filtro de partículas).
- Una de las técnicas más utilizada es el filtro de Kalman Extendido (EKF-SLAM).

Filtro de Kalman Unscented (UKF)

- Presenta una alternativa al EKF con menor error de linealización.
- Existen versiones comparables al EKF en cuanto a costo computacional.
- Más fácil de implementar que el EKF.

Introducción y objetivos

Localización y mapeo simultáneos (SLAM)

- El SLAM presenta una solución conjunta a dos de los problemas fundamentales a resolver para lograr robots autónomos: la localización y el mapeo.
- El problema del SLAM se resuelve generalmente por medio de filtrado bayesiano (KF, filtro de partículas).
- Una de las técnicas más utilizadas es el filtro de Kalman Extendido (EKF-SLAM).

Filtro de Kalman Unscented (UKF)

- Presenta una alternativa al EKF con menor error de linealización.
- Existen versiones comparables al EKF en cuanto a costo computacional.
- Más fácil de implementar que el EKF.

Evaluación comparativa del filtro UKF aplicado al SLAM con principal interés en la estimación tanto del estado del robot como del mapa.

Filtro de Kalman para la estimación del SLAM

Sistema dinámico en tiempo discreto

- Ecuación de proceso:

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

donde \mathbf{u}_k es la entrada de control y $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ el ruido de proceso.

- Ecuación de medición:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)$$

donde $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$.

Filtro de Kalman para la estimación del SLAM

Sistema dinámico en tiempo discreto

- Ecuación de proceso:

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

donde \mathbf{u}_k es la entrada de control y $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ el ruido de proceso.

- Ecuación de medición:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)$$

donde $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$.

Vector de estado en SLAM

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_M \end{bmatrix}, \quad \text{con} \quad \mathbf{x}_R = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad \mathbf{x}_M = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

\mathbf{x}_R es la pose (posición y orientación) del robot, y $\mathbf{x}_i, i = 1, \dots, N$ las posiciones de las características que componen el mapa.

Filtro de Kalman para la estimación del SLAM (cont.)

Dado el modelo de movimiento del robot y asumiendo en mapa estático, se tiene

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{R,k} \\ \mathbf{x}_{M,k} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_R(\mathbf{x}_{R,k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{x}_{M,k-1} \end{bmatrix}$$

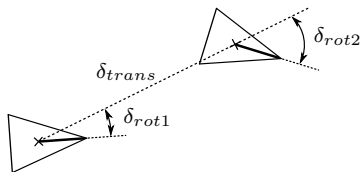
La estimación del filtro es

$$\begin{aligned} \hat{\mathbf{x}}_k &= \mathbb{E}[\mathbf{x}_k | \mathcal{Z}_{0:k}] \\ \mathbf{P}_k &= \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T | \mathcal{Z}_{0:k}] \\ &= \begin{bmatrix} \mathbf{P}_{RR,k} & \mathbf{P}_{RM,k} \\ \mathbf{P}_{RM,k}^T & \mathbf{P}_{MM,k} \end{bmatrix}, \quad \begin{cases} \mathbf{P}_{RR}: 3 \times 3 \\ \mathbf{P}_{MM}: 2N \times 2N \end{cases} \end{aligned}$$

donde $\mathcal{Z}_{0:k}$ es la secuencia de observaciones desde el instante inicial hasta k .

Modelo de movimiento y medición

Modelo de movimiento de odometría



Dadas las lecturas odométricas

$$\mathbf{x}_{k-1}^{odom} = [x_{k-1}^{odom} \quad y_{k-1}^{odom} \quad \theta_{k-1}^{odom}]^T$$

$$\mathbf{x}_k^{odom} = [x_k^{odom} \quad y_k^{odom} \quad \theta_k^{odom}]^T$$

Se tiene

$$\delta_{rot1} = \text{atan2}(y_k^{odom} - y_{k-1}^{odom}, x_k^{odom} - x_{k-1}^{odom}) - \theta_{k-1}^{odom}$$

$$\delta_{trans} = \sqrt{(x_{k-1}^{odom} - x_k^{odom})^2 + (y_{k-1}^{odom} - y_k^{odom})^2}$$

$$\delta_{rot2} = \theta_k^{odom} - \theta_{k-1}^{odom} - \delta_{rot1}$$

La acción de control es

$$\mathbf{u} = [\hat{\delta}_{rot1} \quad \hat{\delta}_{trans} \quad \hat{\delta}_{rot2}]^T$$

$$\mathbf{P}_u = \text{diag}(\sigma_{rot1}^2, \sigma_{trans}^2, \sigma_{rot2}^2)$$

$$\sigma_{rot1} = \alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|$$

$$\sigma_{trans} = \alpha_3 |\delta_{trans}| + \alpha_4 (|\delta_{rot1}| + |\delta_{rot2}|)$$

$$\sigma_{rot2} = \alpha_1 |\delta_{rot2}| + \alpha_2 |\delta_{trans}|$$

Modelo de movimiento y medición (cont.)

El modelo de movimiento queda

$$\mathbf{x}_{R,k} = \mathbf{f}_R(\mathbf{x}_{R,k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$$

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \delta_{trans} \cos(\theta_{k-1} + \delta_{rot1}) \\ \delta_{trans} \sin(\theta_{k-1} + \delta_{rot1}) \\ \delta_{rot1} + \delta_{rot2} \end{bmatrix}$$

Modelo de observación

$$\mathbf{z}_k^i = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k), \quad i = 1, \dots, N_z$$

con $\mathbf{z}_k^i = [z_{x,k}^i \quad z_{y,k}^i]^T$ la i -ésima característica en el espacio de mediciones, y N_z el número de observaciones.

Extended Kalman Filter (EKF)

Predicción

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

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

Corrección

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-))$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

Las matrices \mathbf{F}_k y \mathbf{H}_k son los jacobianos de las ec. de movimiento y medición.

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{F}_{R,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{Q}_k = \begin{bmatrix} \mathbf{F}_{u,k} \mathbf{P}_u \mathbf{F}_{u,k}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_{R,k}^1 & \mathbf{H}_{M,k}^1 & \cdots & \mathbf{0} \\ \mathbf{H}_{R,k}^2 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{R,k}^N & \mathbf{0} & \cdots & \mathbf{H}_{M,k}^N \end{bmatrix}, \quad \mathbf{R}_k = \mathbf{J}_{pc} \mathbf{P}_l \mathbf{J}_{pc}^T$$

donde

$$\mathbf{F}_{R,k} = \frac{\partial \mathbf{f}_R(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_{R,k}}, \quad \mathbf{F}_{u,k} = \frac{\partial \mathbf{f}_R(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{u}_k}, \quad \mathbf{H}_{R,k}^i = \frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_{R,k}}, \quad \mathbf{H}_{M,k}^i = \frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_{M,k}^i}$$

UKF - Transformada Unscented (UT)

Método numérico para aproximar la distribución conjunta

$$\mathbf{X} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P}), \quad \mathbf{Y} = \mathbf{g}(\mathbf{X})$$

Del algoritmo

- λ : parámetro de escala.
- α : constante que fija la dispersión de los puntos sigma (e.g., $1 \leq \alpha \leq 1 \times 10^{-4}$).
- κ : es un parámetro de escala secundario (0 o $3 - n$).
- β : es un parámetro opcional (para dist. gaussiana $\beta = 2$).
- $\left(\sqrt{(n + \lambda)\mathbf{P}}\right)_i$: es la columna i -ésima de la raíz cuadrada de la matriz.

- 1: **Input:** $\bar{\mathbf{x}}, \mathbf{P}_{\mathbf{x}}$ ▷ \mathbf{X} mean and covariance matrix
- 2: $\mathbf{g}(\cdot)$ ▷ non-linear function
- 3: $\{\alpha, \beta, \kappa\}$ ▷ parameters
- 4: **Output:** $\bar{\mathbf{y}}, \mathbf{P}_{\mathbf{y}}$ ▷ \mathbf{Y} mean and covariance matrix
- 5: $\mathbf{P}_{\mathbf{xy}}$ ▷ cross-covariance matrix
- 6: $n \leftarrow$ length of $\bar{\mathbf{x}}$
- 7: $\lambda \leftarrow \sigma^2(n + \kappa) - n$

I. Compute weights

- 8: $W_0^{(m)} \leftarrow \lambda / (n + \lambda)$
- 9: $W_0^{(c)} \leftarrow \lambda / (n + \lambda) + (1 - \alpha^2 + \beta)$
- 10: $W_i^{(m)} \leftarrow W_i^{(c)} = 1 / (2(n + \lambda)), \quad i = 1, \dots, 2n$

II. Form a set of $2n + 1$ sigma points (SP)

- 11: $\mathcal{X}_0 \leftarrow \bar{\mathbf{x}}$
- 12: $\mathcal{X}_i \leftarrow \bar{\mathbf{x}} + \left(\sqrt{(n + \lambda)\mathbf{P}}\right)_i, \quad i = 1, \dots, n$
- 13: $\mathcal{X}_i \leftarrow \bar{\mathbf{x}} - \left(\sqrt{(n + \lambda)\mathbf{P}}\right)_{i-n}, \quad i = n + 1, \dots, 2n$

III. Propagate SP through $\mathbf{g}(\cdot)$

- 14: $\mathcal{Y}_i \leftarrow \mathbf{g}(\mathcal{X}_i), \quad i = 0, \dots, 2n$

IV. Estimate mean and covariance matrix

- 15: $\bar{\mathbf{y}} \leftarrow \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_i^{(i)}$
- 16: $\mathbf{P}_{\mathbf{y}} \leftarrow \sum_{i=0}^{2n} W_i^{(c)} (\mathcal{Y}_i^{(i)} - \bar{\mathbf{y}})(\mathcal{Y}_i^{(i)} - \bar{\mathbf{y}})^T$
- 17: $\mathbf{P}_{\mathbf{xy}} \leftarrow \sum_{i=0}^{2n} W_i^{(c)} (\mathcal{X}_i^{(i)} - \bar{\mathbf{x}})(\mathcal{Y}_i^{(i)} - \bar{\mathbf{y}})^T$

Unscented Kalman Filter (UKF)

Utilizando la UT con $f(\cdot)$:

Dada

$$\mathbf{X}_{k-1} \sim \{\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}\}$$

se obtiene

$$\mathbf{X}_k^- \sim \{\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-\}$$

Predicción

Se utiliza la UT para obtener $\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-$ a partir de $\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}$ y $f(\cdot)$

Utilizando la UT con $h(\cdot)$:

Dada

$$\mathbf{X}_k^- \sim \{\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-\}$$

se obtiene

$$\mathbf{Z}_k^- \sim \{\hat{\mathbf{z}}_k^-, \mathbf{R}_k^-\}, \quad \mathbf{R}_k^- = \mathbf{P}_{\mathbf{z}\mathbf{z},k}$$

además de $\mathbf{P}_{\mathbf{x}\mathbf{z},k}$.

Corrección

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}\mathbf{z},k} \mathbf{S}_k^{-1}$$

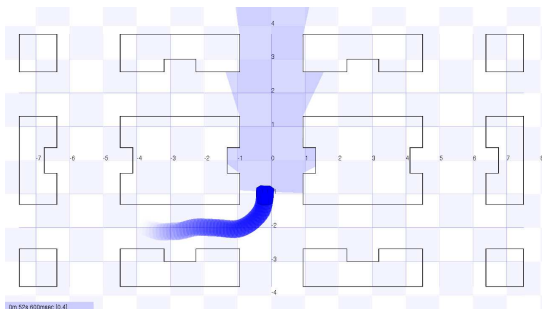
$$\mathbf{x}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k^-)$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T$$

con $\mathbf{S}_k = \mathbf{R}_k + \mathbf{R}_k^-$

Implementación

- Se utilizó el entorno de desarrollo de robótica Player/Stage.
- Aplicado a un robot móvil de tracción diferencial con un escáner láser.
- Se evaluaron diferentes escenarios de entornos interiores.



Sensor láser

$$\mathcal{L} = \{(\rho_i, \phi_i), i = 1, \dots, N_l\}$$

donde (ρ_i, ϕ_i) es el i -ésimo punto medido en coordenadas polares. Los parámetros son $\rho_{max} = 8m$, $\sigma_\rho = 10\% \rho_{max}$ y $\sigma_\phi = 10\% \Delta_\phi$.

Odometría

Parámetros del modelo de movimiento:

$$\alpha_1 = 1e^{-2}, \quad \alpha_2 = 8e^{-3}$$

$$\alpha_3 = 1e^{-2}, \quad \alpha_4 = 5e^{-4}$$

Algoritmos

EKF-SLAM

- 1: **Input:** $\mathbf{x}_0, \mathbf{P}_0$ ▷ prior information
 2: **Output:** $\mathbf{x}_k, \mathbf{P}_k$ ▷ SLAM estimate

I. Filter prediction (time update)

- 3: $[\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-] \leftarrow \text{ekf_prediction}(\hat{\mathbf{x}}_{k-1}^-, \mathbf{P}_{k-1}^-)$

II. Data association

- 4: $[\hat{\mathbf{z}}_k^-, \mathbf{H}_k] \leftarrow \text{measurement_prediction}(\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-)$
 5: $[\mathbf{z}_k, \mathbf{R}_k] \leftarrow \text{new_measurement}()$
 6: $\mathcal{H} \leftarrow \text{data_association}(\mathbf{z}_k, \mathbf{R}_k, \hat{\mathbf{z}}_k^-, \mathbf{H}_k, \mathbf{P}_k^-)$

III. Update (time correction)

- 7: $[\hat{\mathbf{x}}_k, \mathbf{P}_k] \leftarrow \text{ekf_update}(\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-, \mathbf{z}_k, \mathbf{R}_k, \hat{\mathbf{z}}_k^-, \mathbf{H}_k, \mathcal{H})$

IV. Map management

- 8: $[\hat{\mathbf{x}}_k, \mathbf{P}_k] \leftarrow \text{add_feature_to_map}(\hat{\mathbf{x}}_k, \mathbf{P}_k, \mathbf{z}_k, \mathbf{R}_k, \mathcal{H})$

Asociación de datos.

- Medida de compatibilidad: distancia de Mahalanobis.
- Criterio de selección: Vecino más cercano (NN).

UKF-SLAM

- 1: **Input:** $\mathbf{x}_0, \mathbf{P}_0$ ▷ prior information
 2: **Output:** $\mathbf{x}_k, \mathbf{P}_k$ ▷ SLAM estimate

I. Filter prediction (time update)

- 3: $\hat{\mathbf{x}}_{k-1}^a \leftarrow [\hat{\mathbf{x}}_{k-1}^T \mathbf{w}_k^T]^T$
 4: $\mathbf{P}_{k-1}^a \leftarrow \text{diag}(\mathbf{P}_{k-1}, \mathbf{P}_u)$
 5: $[\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-] \leftarrow \text{ukf_prediction}(\hat{\mathbf{x}}_{k-1}^a, \mathbf{P}_{k-1}^a)$

II. Data association

- 6: $\hat{\mathbf{x}}_k^{-a} \leftarrow [\hat{\mathbf{x}}_k^{-T} \mathbf{v}_k^T]^T$
 7: $\mathbf{P}_k^{-a} \leftarrow \text{diag}(\mathbf{P}_k^-, \mathbf{P}_l)$
 8: $[\hat{\mathbf{z}}_k^-, \mathbf{R}_k^-, \mathbf{P}_{\mathbf{z}\mathbf{z},k}] \leftarrow \text{measurement_prediction}(\hat{\mathbf{x}}_k^{-a}, \mathbf{P}_k^{-a})$
 9: $[\mathbf{z}_k, \mathbf{R}_k] \leftarrow \text{new_measurement}()$
 10: $\mathcal{H} \leftarrow \text{data_association}(\mathbf{z}_k, \mathbf{R}_k, \hat{\mathbf{z}}_k^-, \mathbf{R}_k^-)$

III. Update (time correction)

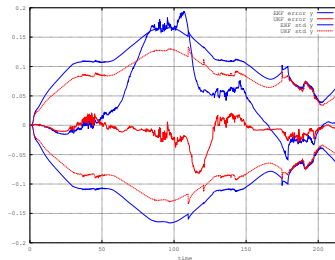
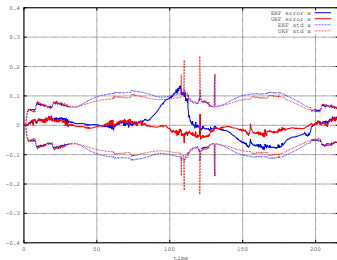
- 11: $[\hat{\mathbf{x}}_k, \mathbf{P}_k] \leftarrow \text{ukf_update}(\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-, \mathbf{z}_k, \mathbf{R}_k, \hat{\mathbf{z}}_k^-, \mathbf{R}_k^-, \mathbf{P}_{\mathbf{z}\mathbf{z},k}, \mathcal{H})$

IV. Map management

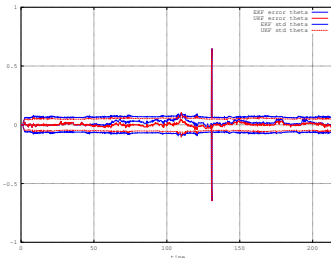
- 12: $[\hat{\mathbf{x}}_k, \mathbf{P}_k] \leftarrow \text{add_feature_to_map}(\hat{\mathbf{x}}_k, \mathbf{P}_k, \mathbf{z}_k, \mathbf{R}_k, \mathcal{H})$

Estimación de la pose del robot

Estimación en posición (x, y)

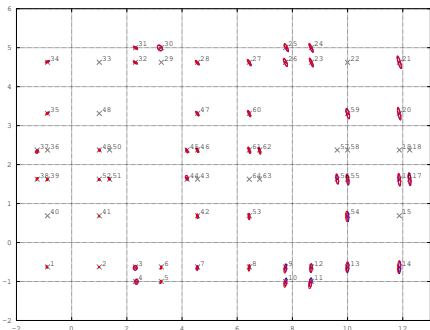


Estimación en orientación θ

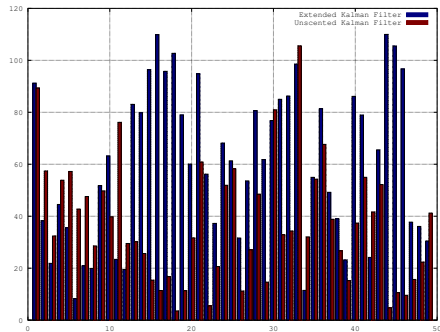


Estimación del mapa

Mapa real y estimado

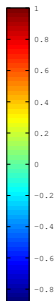
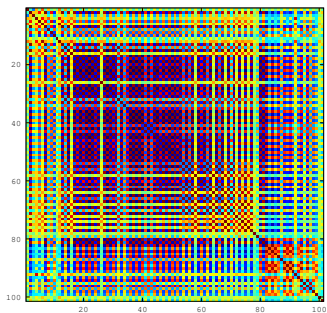


Distancia de Mahalanobis

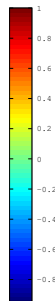
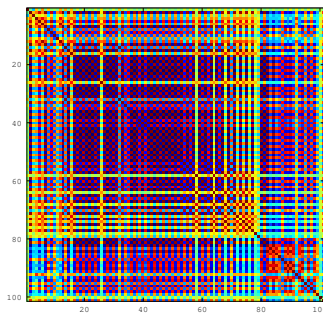


Matriz de correlación

EKF-SLAM



UKF-SLAM



$$-1 \leq c_{i,j} \leq 1$$

Conclusiones y trabajos futuros

- Se realizó la implementación de un filtro UKF para la estimación del SLAM con mapa de características.
- Se llevó a cabo una evaluación comparativa con la implementación típica con EKF, enfocado principalmente en la estimación tanto de la pose del robot como las posiciones de las características del mapa.
- Los resultados muestra una mejora no muy significativa en la estimación de la pose del robot aunque notable en el error de la estimación del mapa.
- El UKF presenta una mayor correlación en le estimación final del SLAM.
- La principal ventaja del UKF es su mayor facilidad en la implementación respecto al EKF.
- Evaluar el UKF en escenarios de mayor dimensiones, además de la aplicación en SLAM visual con modelos altamente no lineales.

Implementation and performance evaluation of UKF for Simultaneous Localization and Mapping

Gracias por su atención.

Gonzalo F. Perez Paina
gperez@scdt.frc.utn.edu.ar