

Localización y Mapeo

Dr José Martínez Carranza
carranza@inaoep.mx

Dr Alejandro Gutiérrez Giles
alejandro.giles@inaoep.mx

Primavera 2023

Coordinación de Ciencias Computacionales, INAOE

¿Experiencias con la práctica 1?



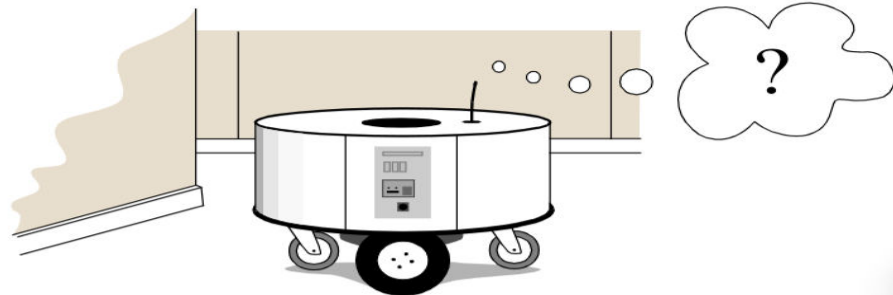
Estimación de la pose

Problemas:

- Para que un robot pueda navegar en un ambiente, debe conocer su ubicación en dicho ambiente (Localización).
- Para que un robot pueda interactuar con su ambiente, debe contar con una representación del ambiente (Mapeo).

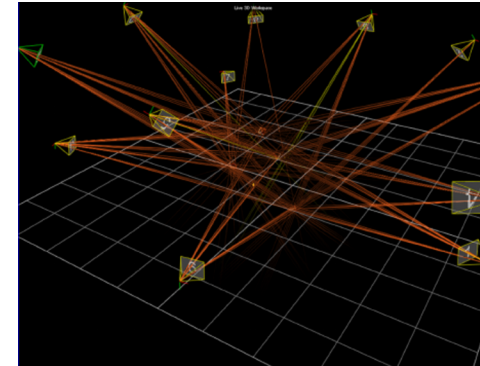
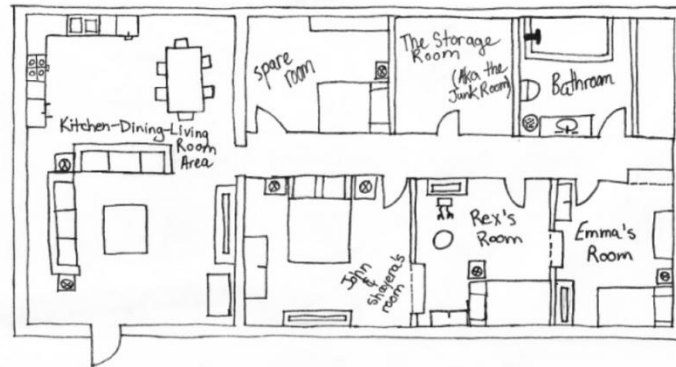
Localización

- La localización es la habilidad de un robot móvil para situarse en un ambiente .
- Saber “donde está” le sirve para decidir que hacer y construir una representación de lo que se encuentra a su alrededor (mapa).
- La localización puede realizarse de diversas maneras:
 - Utilizando mapas
 - A través de marcas en el ambiente
 - Estimando su posición a través de los datos obtenidos con los sensores a bordo (sonar, cámara, laser, odómetro, brújula, GPS, etc.).



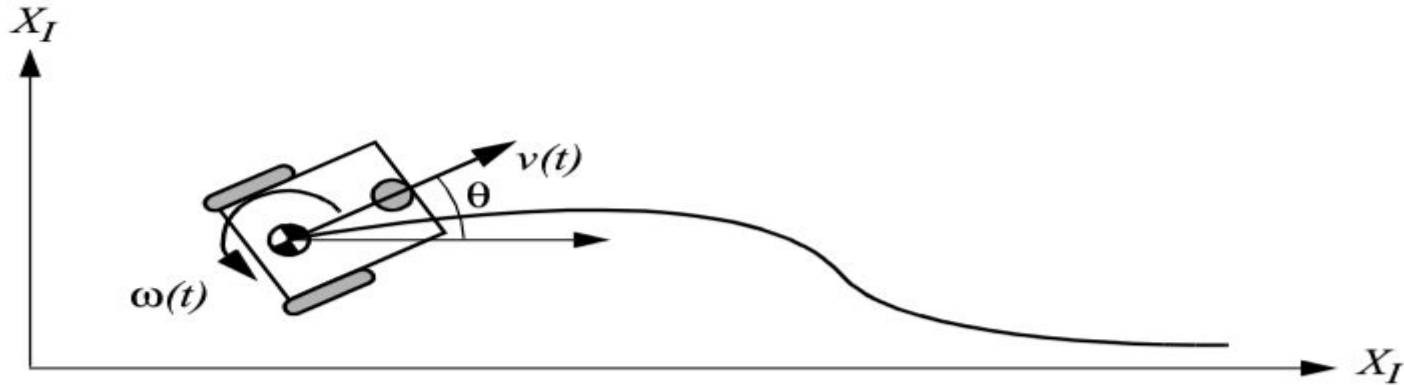
Ejemplo de sistemas de localización

- Global Positioning System (GPS)
- Ubicación urbana (código postal, dirección)
- Ubicación topológica
- Sistemas de captura de movimiento



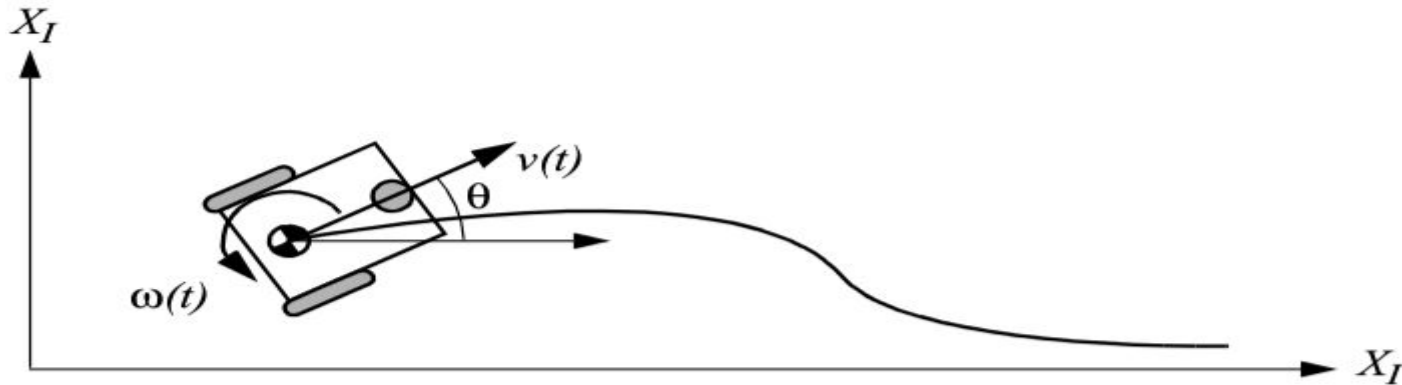
Localización

- La localización puede llevarse a cabo de dos maneras:
 - De manera continua (para cada movimiento del robot).

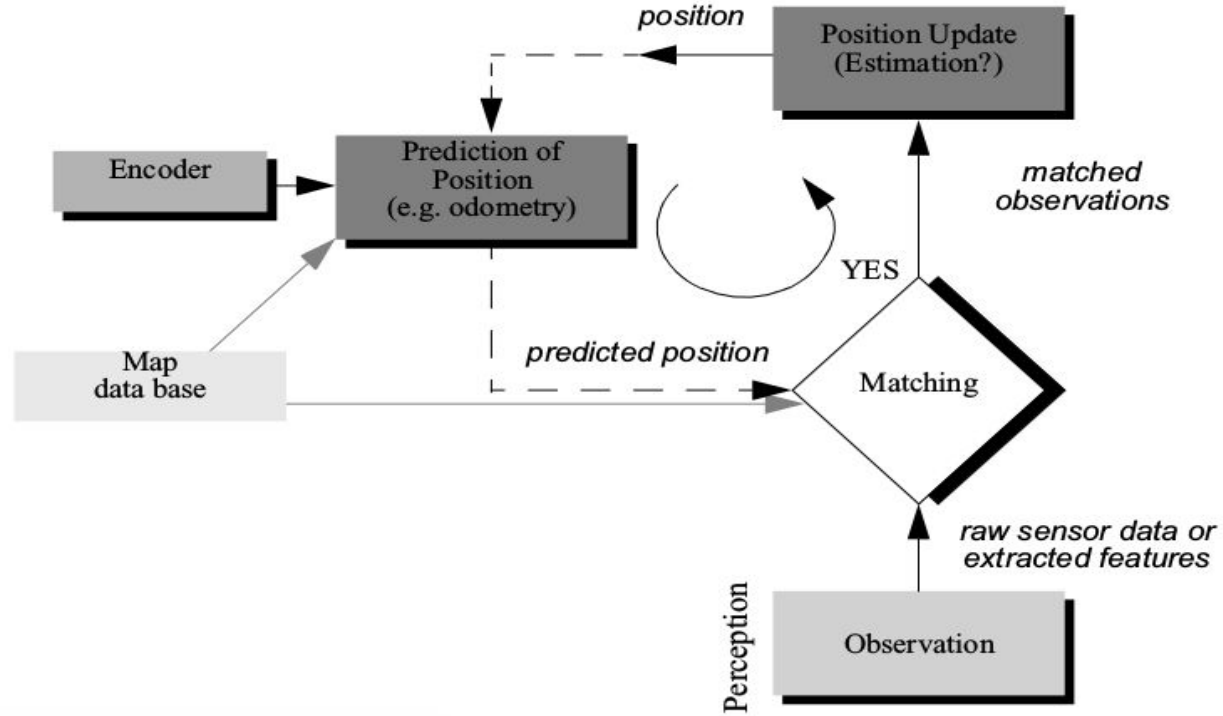


Localización

- La localización puede llevarse a cabo de dos maneras:
 - De manera continua (para cada movimiento del robot).
 - De manera puntual (sólo cuando se reconocen u observan marcas conocidas en el ambiente).

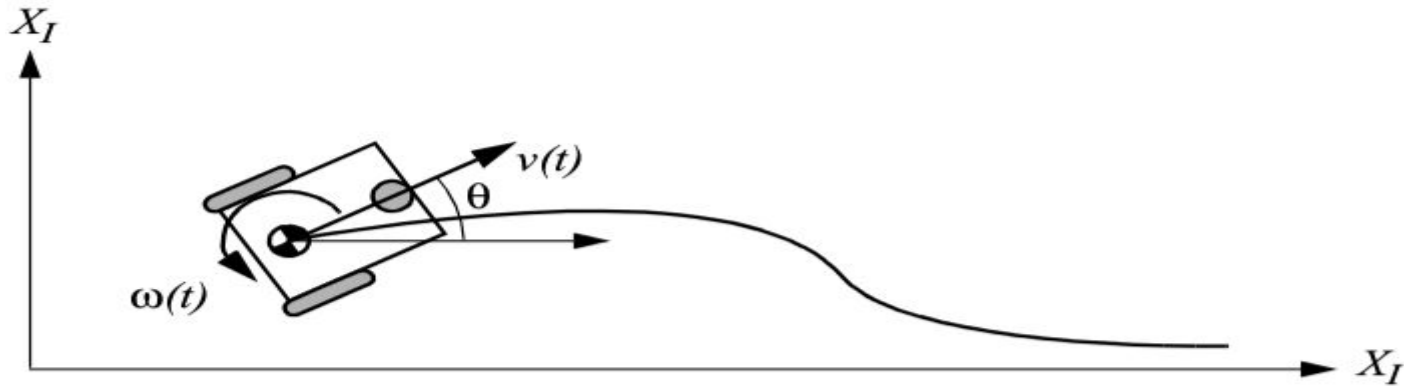


Esquema general de localización



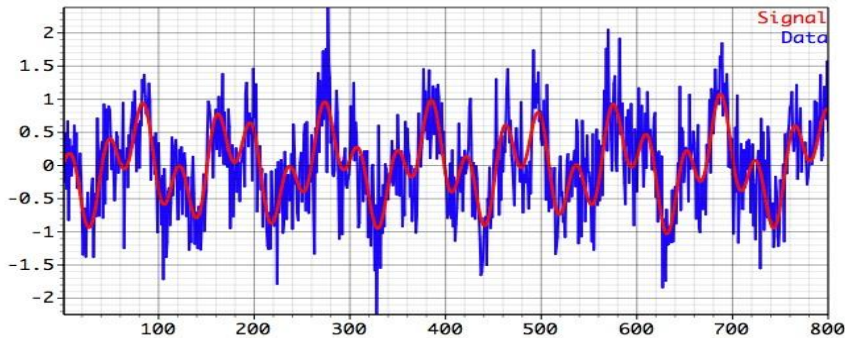
Localización

- Retos en el proceso de localización:
 - Ruido en los sensores.
 - Aliasing.
 - Ruido en los efectores.



Ruidos en sensores

- Sonares y laser (dependen del material y ángulo de incidencia).
- Sensores inerciales (acelerómetros, inclinómetros).
- Se recomienda tomar múltiples lecturas y hacen fusión temporal/multi-sensorial o filtrado.



Aliasing

- Mismas lecturas para diferentes estados.
- Ejemplo: las lecturas de un laser en dos cuartos diferentes pero de mismas dimensiones.



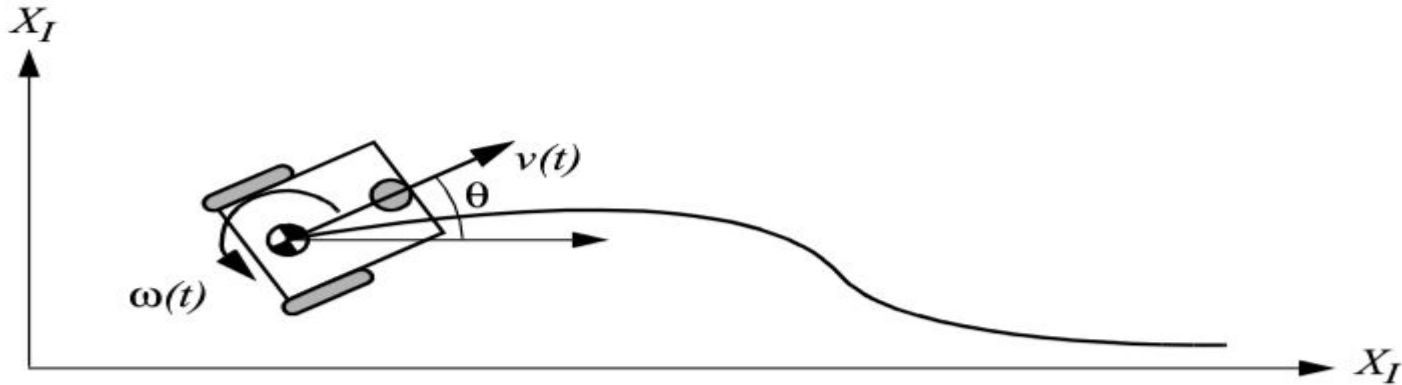
Ruido en los efectores

- Errores odométricos: poca resolución, no alineación de llantas, diámetro desigual, variaciones de contacto.
- Introducen errores en los estados futuros.
- El error de posición es acumulativo, por lo que hay que actualizar la posición continuamente.
- Algunos errores son sistemáticos – deterministas y otros aleatorios – no deterministas.

Dead reckoning

- Estimación de la posición utilizando odometría.

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$



Dead reckoning

- Estimación de la posición utilizando odometría.

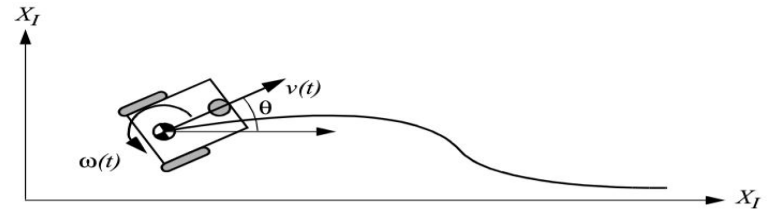
$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

$$\Delta x = \Delta s \cos(\theta + \Delta\theta / 2)$$

$$\Delta y = \Delta s \sin(\theta + \Delta\theta / 2)$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b}$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$



$(\Delta x; \Delta y; \Delta\theta)$ = path traveled in the last sampling interval;

$\Delta s_r; \Delta s_l$ = traveled distances for the right and left wheel respectively;

b = distance between the two wheels of differential-drive robot.

Dead reckoning

- Estimación de la posición utilizando odometría.

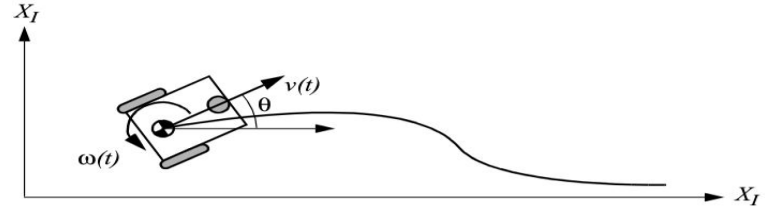
$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

$$\Delta x = \Delta s \cos(\theta + \Delta\theta / 2)$$

$$\Delta y = \Delta s \sin(\theta + \Delta\theta / 2)$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b}$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$



$(\Delta x; \Delta y; \Delta\theta)$ = path traveled in the last sampling interval;

$\Delta s_r; \Delta s_l$ = traveled distances for the right and left wheel respectively;

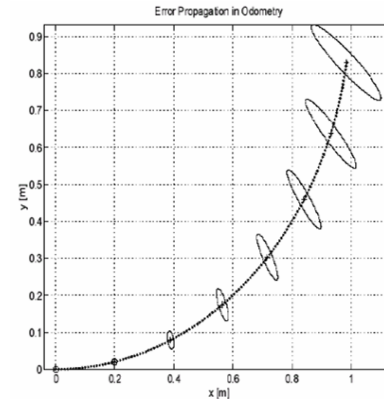
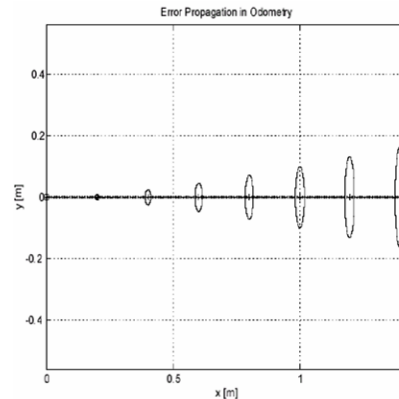
b = distance between the two wheels of differential-drive robot.

$$p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = p + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta / 2) \\ \Delta s \sin(\theta + \Delta\theta / 2) \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta / 2) \\ \Delta s \sin(\theta + \Delta\theta / 2) \\ \Delta\theta \end{bmatrix}$$

Dead reckoning (errores odométricos)

- Los errores de orientación son mayores que los de distancia.

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$



Localizarse o no localizarse

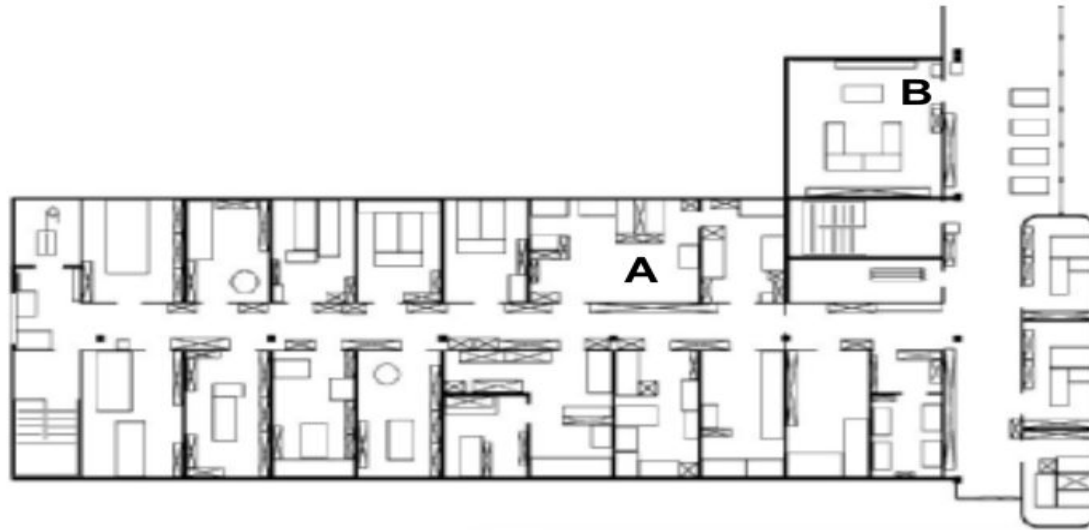
- La navegación no tiene que ser basada en localización necesariamente, esto es, no tiene por qué conocerse la posición/orientación del robot con respecto a un mapa.
- Localizarse con respecto a un mapa ayuda a conocer cuando un robot ha llegado a la meta.
- Sin embargo, conocer la ubicación explícita en un mapa no es la única solución.

Navegación basada en comportamiento

- Si los sensores y los actuadores son ruidosos de resolución limitada, entonces debe evitarse la creación de un mapa geométrico (métrico).
- **Alternativa:** diseñar un conjunto de comportamientos que en conjunto resulten en el movimiento deseado del robot.
 - Bajo este esquema se evita el razonamiento explícito acerca de la posición/localización del robot, así como de la planificación.

Navegación basada en comportamiento

- Debe existir una solución basada en un procedimiento más que en un cálculo geométrico.
 - **Ejemplo:** navegación siguiendo el muro izquierdo combinado con un reconocedor de la meta (la meta incluye algún rasgo distintivo único).

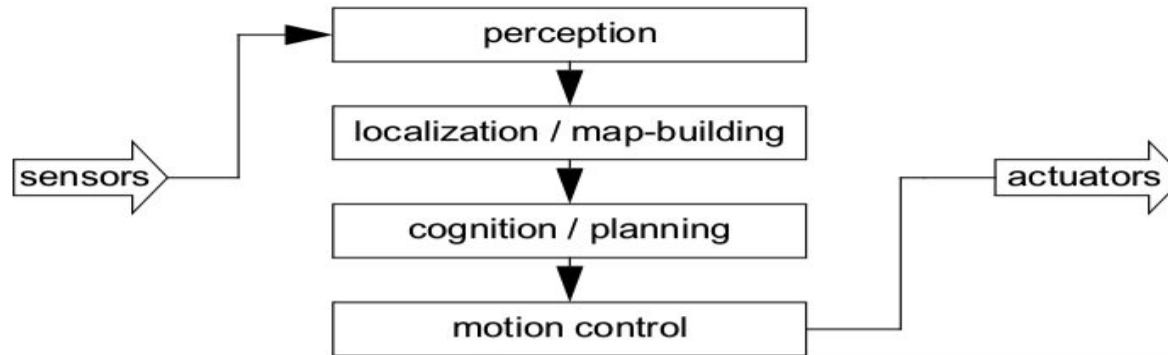
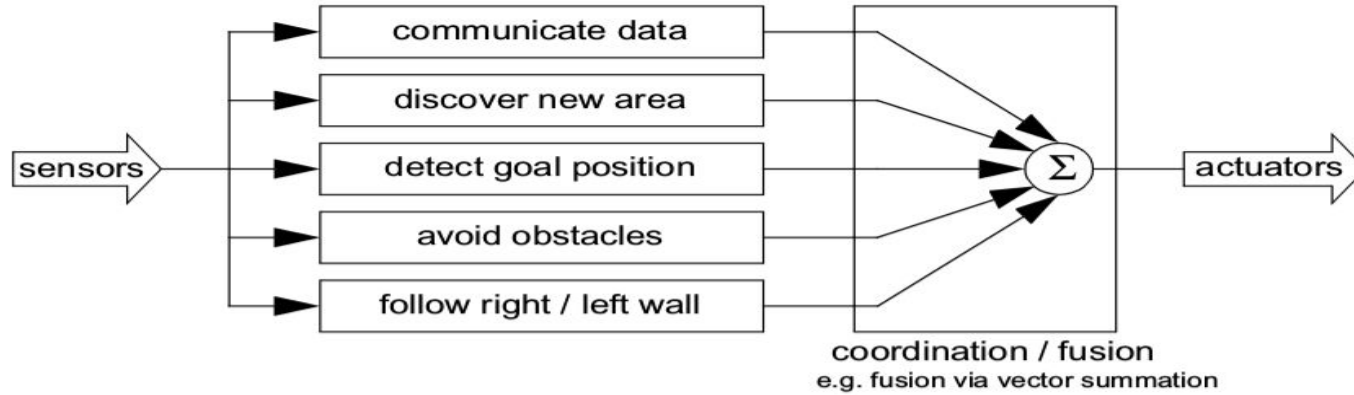


Navegación basada en comportamiento

- Debe existir una solución basada en un procedimiento más que en un cálculo geométrico.
 - **Ejemplo:** navegación siguiendo el muro izquierdo combinado con un reconocedor de la meta (la meta incluye algún rasgo distintivo único).



Localización vs Comportamiento

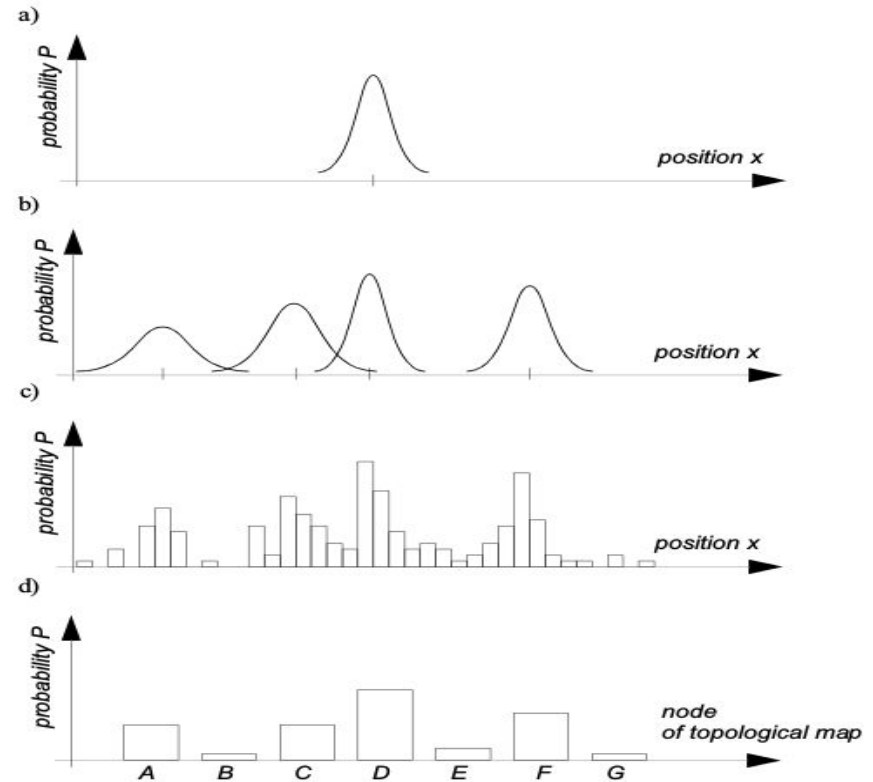


Representación de creencias

- Existen dos tipos de conceptos que deben ser representados para que el robot pueda llevar a cabo la navegación:
 - El modelo del ambiente a través de un mapa.
 - Su *creencia* acerca de su posición en el ambiente (mapa).
 - ¿Se cree en una sólo posición (hipótesis única)?
 - ¿Qué pasa si existen varias posibilidades (múltiples hipótesis)?

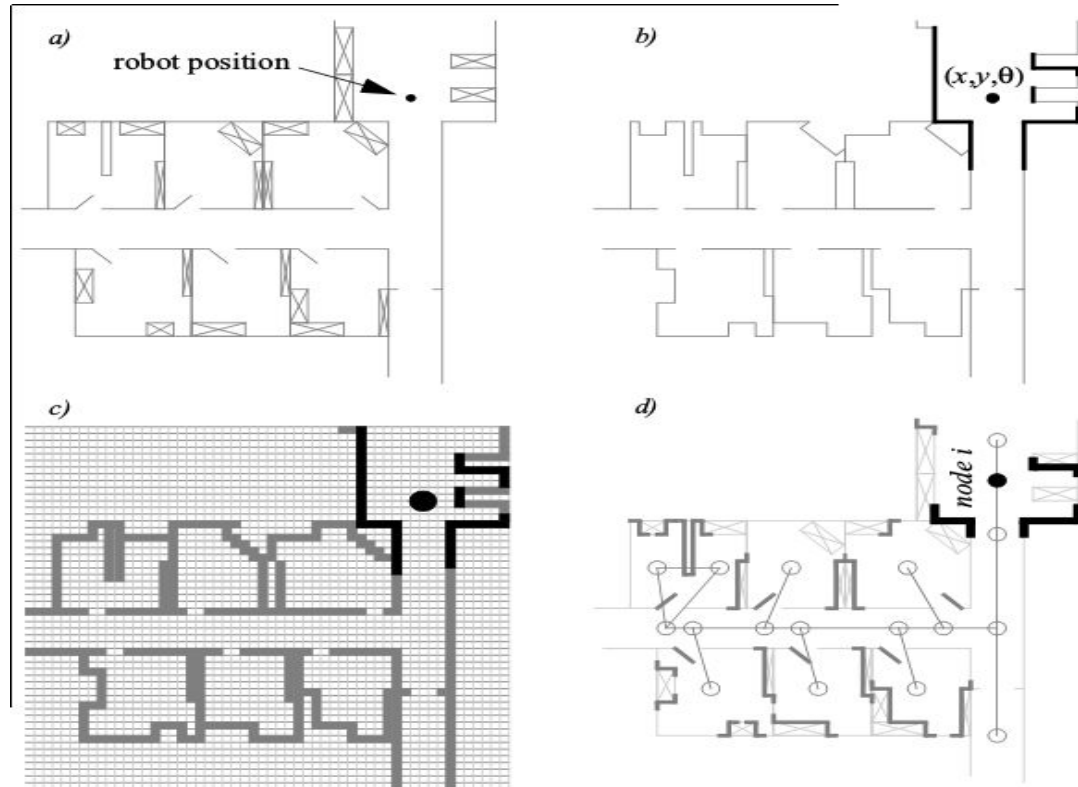
Representación de creencias

- Mapa continuo una hipótesis.
- Mapa continuo muchas hipótesis.
- Mapa discreto de rejillas con todas las posiciones.
- Mapa discreto de grafos con todos los nodos.



Representación de creencias

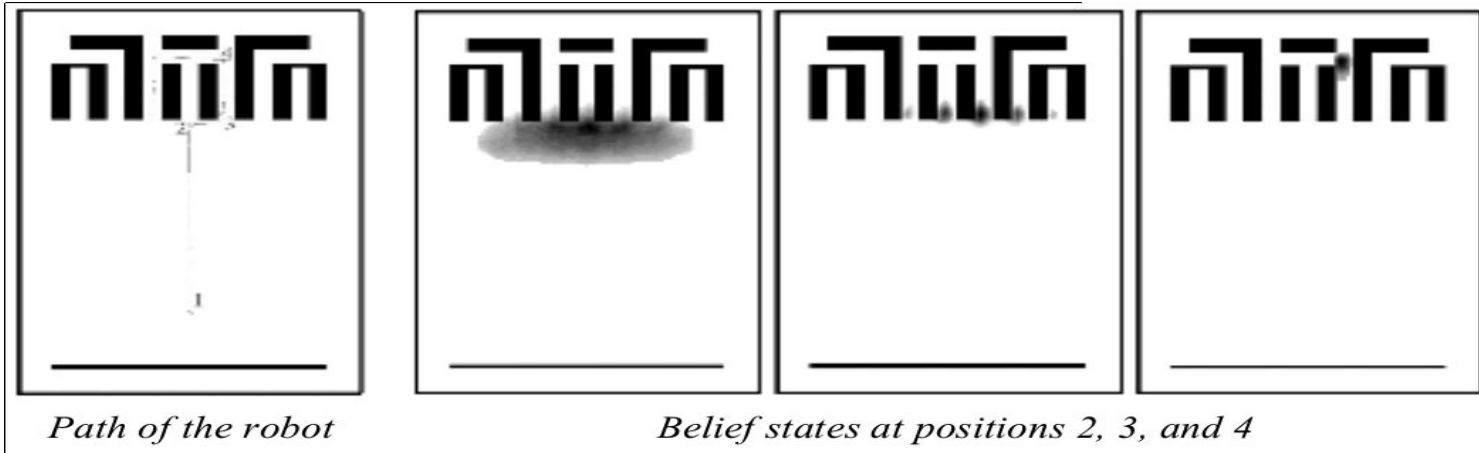
(Hipótesis única)



Representación de creencias

(Múltiples hipótesis)

- Se puede mantener un modelo de la incertidumbre con respecto a la posición del robot.
- **Problemas:**
 - De toma de decisión: ¿qué hipótesis es la mejor?
 - Asignación de probabilidad a cada hipótesis.
 - El incremento en el número de posibles hipótesis incrementa rápida el tiempo computacional (se vuelve intratable rápidamente en el espacio 3D).



Localización basada en mapas probabilísticos

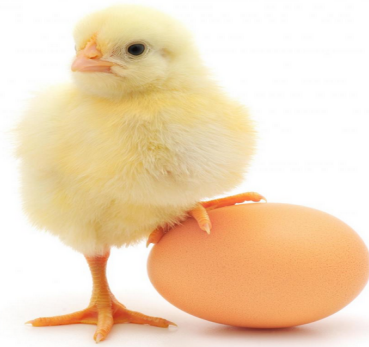
- Similar a la representación de creencias con múltiples hipótesis.
- Asociación de probabilidades a las posibles posiciones (configuraciones) del robot:
 - Localización de Markov.
 - Utiliza una distribución de probabilidad explícita.
 - Cada configuración se asume independiente de otra.
 - Localización basada en el Filtro Kalman (popular en SLAM).
 - Utiliza una distribución de probabilidad Gaussiana.
 - Las configuraciones no son independientes.
 - Desarrollado originalmente para sistemas lineales.
 - EKF, UKF, para sistemas no lineales.

Localización y Mapeo Simultaneos

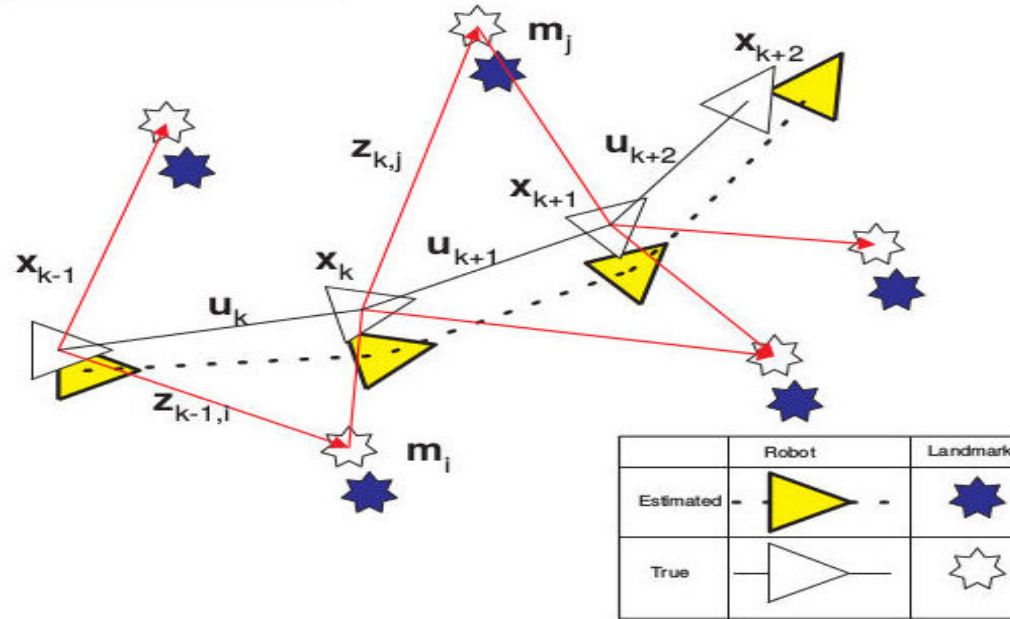
- SLAM por sus siglas en ingles: **S**imultaneous **L**ocalisation **A**nd **M**apping.
- ICRA-1986: Métodos probabilísticos comienzan a ser considerados para resolver el problema de localización y mapeo.
- ISRR-1995: Se inventa el término SLAM.
- ISRR-1999: Primer sesión acerca de SLAM.
- ICRA-2000: Taller sobre SLAM (15 investigadores).
- ICRA-2001: Taller sobre SLAM (150 investigadores).
- Escuela de verano sobre SLAM 2002, KTH Royal Institute of Technology in Stockholm: reunió a todos los investigadores claves en el área y 50 estudiantes de doctorado de varios lugares del mundo.
- Actualmente se organizan hasta 2 sesiones de SLAM en ICRA, IROS, RSS.

SLAM

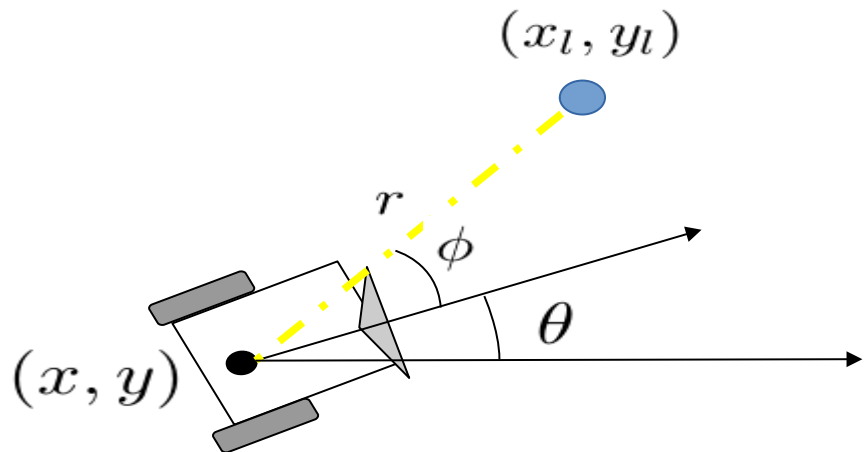
- Problema del huevo y la gallina



SLAM



SLAM para robot diferencial (basado en el EKF)

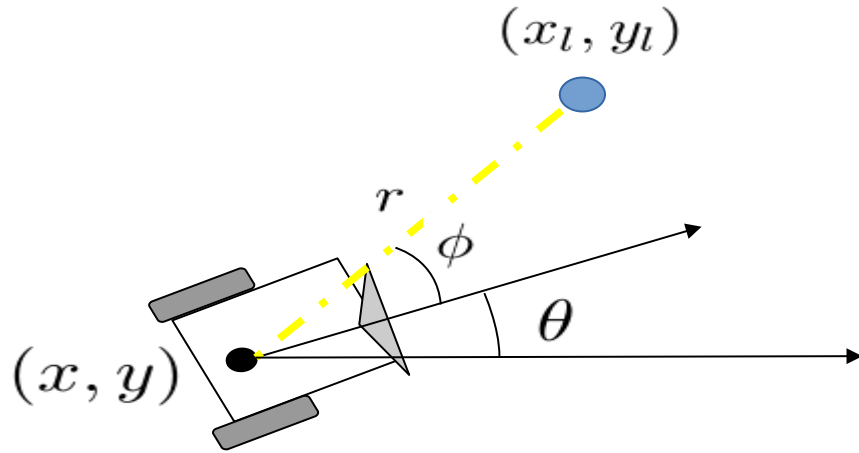


- Estado, posición del robot y componente del mapa.

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \\ x_l \\ y_l \end{bmatrix}$$

Sensor: mide ángulo (con respecto al eje del robot) y distancia del componente del mapa.

SLAM para robot diferencial (basado en el EKF)

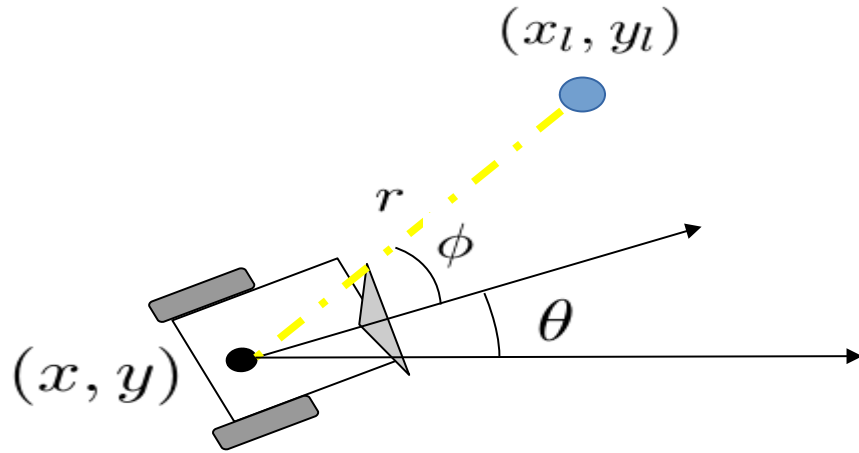


Sensor: mide ángulo (con respecto al eje del robot) y distancia del componente del mapa.

● Modelo dinámico

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{y}(k+1) \\ \hat{\theta}(k+1) \\ \hat{x}_l(k+1) \\ \hat{y}_l(k+1) \end{bmatrix} = f(\hat{x}_k, \hat{y}_k, \hat{\theta}_k, \hat{x}_l(k), \hat{y}_l(k)) = \begin{bmatrix} \hat{x}_k + \frac{\Delta s_r + \Delta s_l}{2} \cos(\hat{\theta}_k + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \hat{y}_k + \frac{\Delta s_r + \Delta s_l}{2} \sin(\hat{\theta}_k + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \hat{\theta}_k + \frac{\Delta s_r - \Delta s_l}{2b} \\ \hat{x}_l(k) \\ \hat{y}_l(k) \end{bmatrix}$$

SLAM para robot diferencial (basado en el EKF)



Sensor: mide ángulo (con respecto al eje del robot) y distancia del componente del mapa.

● Modelo de observación/medición

$$\begin{bmatrix} \hat{r} \\ \hat{\phi} \end{bmatrix} = h(\hat{x}_k, \hat{y}_k, \hat{\theta}_k, \hat{x}_{l(k)}, \hat{y}_{l(k)}) = \begin{bmatrix} \sqrt{(\hat{x}_{l(k+1)} - \hat{x}_{(k+1)})^2 + (\hat{y}_{l(k+1)} - \hat{y}_{(k+1)})^2} \\ \tan^{-1} \left(\frac{\hat{y}_{l(k+1)} - \hat{y}_{(k+1)}}{\hat{x}_{l(k+1)} - \hat{x}_{(k+1)}} \right) - \hat{\theta}_{(k+1)} \end{bmatrix}$$

SLAM para robot diferencial (basado en el EKF)

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{y}(k+1) \\ \hat{\theta}(k+1) \\ \hat{x}_l(k+1) \\ \hat{y}_l(k+1) \end{bmatrix} = f(\hat{x}_k, \hat{y}_k, \hat{\theta}_k, \hat{x}_l(k), \hat{y}_l(k)) = \begin{bmatrix} \hat{x}_k + \frac{\Delta s_r + \Delta s_l}{2} \cos(\hat{\theta}_k + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \hat{y}_k + \frac{\Delta s_r + \Delta s_l}{2} \sin(\hat{\theta}_k + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \hat{\theta}_k + \frac{\Delta s_r - \Delta s_l}{2b} \\ \hat{x}_l(k) \\ \hat{y}_l(k) \end{bmatrix}$$

$$\begin{bmatrix} \hat{r} \\ \hat{\phi} \end{bmatrix} = h(\hat{x}_k, \hat{y}_k, \hat{\theta}_k, \hat{x}_l(k), \hat{y}_l(k)) = \begin{bmatrix} \sqrt{(\hat{x}_l(k+1) - \hat{x}(k+1))^2 + (\hat{y}_l(k+1) - \hat{y}(k+1))^2} \\ \tan^{-1} \left(\frac{\hat{y}_l(k+1) - \hat{y}(k+1)}{\hat{x}_l(k+1) - \hat{x}(k+1)} \right) - \hat{\theta}(k+1) \end{bmatrix}$$

Filtro de Kalman Extendido (EKF)

Predicción

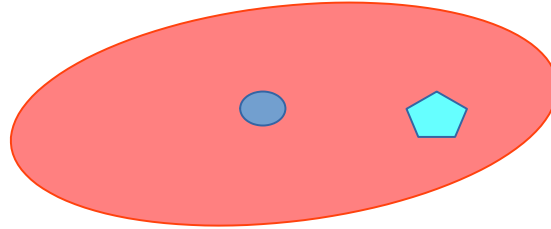
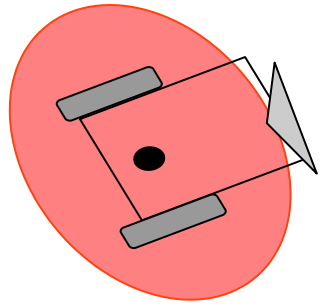
$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0) \\ \mathbf{P}_{k|k-1} &= \nabla \mathbf{f}_{\mathbf{x}} \mathbf{P}_{k-1} \nabla \mathbf{f}_{\mathbf{x}}^{\top} + \nabla \mathbf{f}_{\mathbf{w}} \mathbf{Q} \nabla \mathbf{f}_{\mathbf{w}}^{\top} \\ \hat{\mathbf{z}}_k &= \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}, 0) \\ \mathbf{S}_k &= \nabla \mathbf{h}_{\mathbf{x}} \mathbf{P}_{k-1} \nabla \mathbf{h}_{\mathbf{x}}^{\top} + \nabla \mathbf{h}_{\mathbf{v}} \mathbf{R} \nabla \mathbf{h}_{\mathbf{h}}^{\top} \end{aligned}$$

Corrección

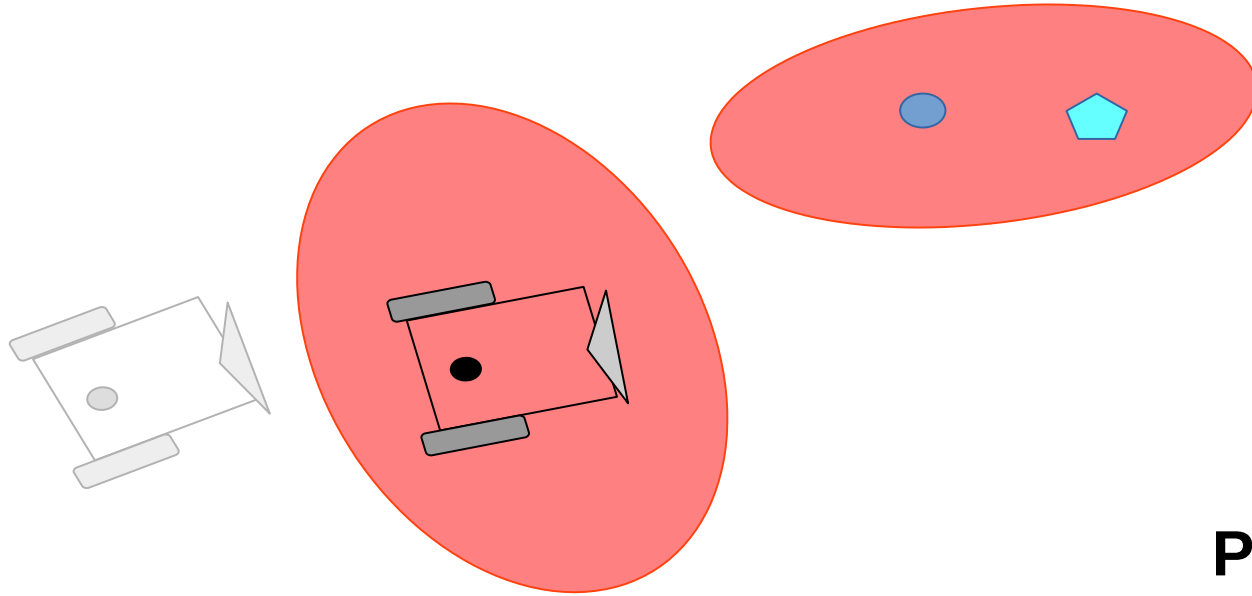
$$\begin{aligned} \mathbf{y}_k &= \mathbf{z}_k - \hat{\mathbf{z}}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \nabla \mathbf{h}_{\mathbf{x}}^{\top} \mathbf{S}^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{y}_k \\ \mathbf{P}_k &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \nabla \mathbf{h}_{\mathbf{x}} \mathbf{P}_{k|k-1} \end{aligned}$$

$$\nabla \mathbf{f}_{\mathbf{x}} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}}, \quad \nabla \mathbf{f}_{\mathbf{w}} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}}, \quad \nabla \mathbf{h}_{\mathbf{x}} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}, \quad \nabla \mathbf{h}_{\mathbf{v}} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{v}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}$$

SLAM para robot diferencial (basado en el EKF)

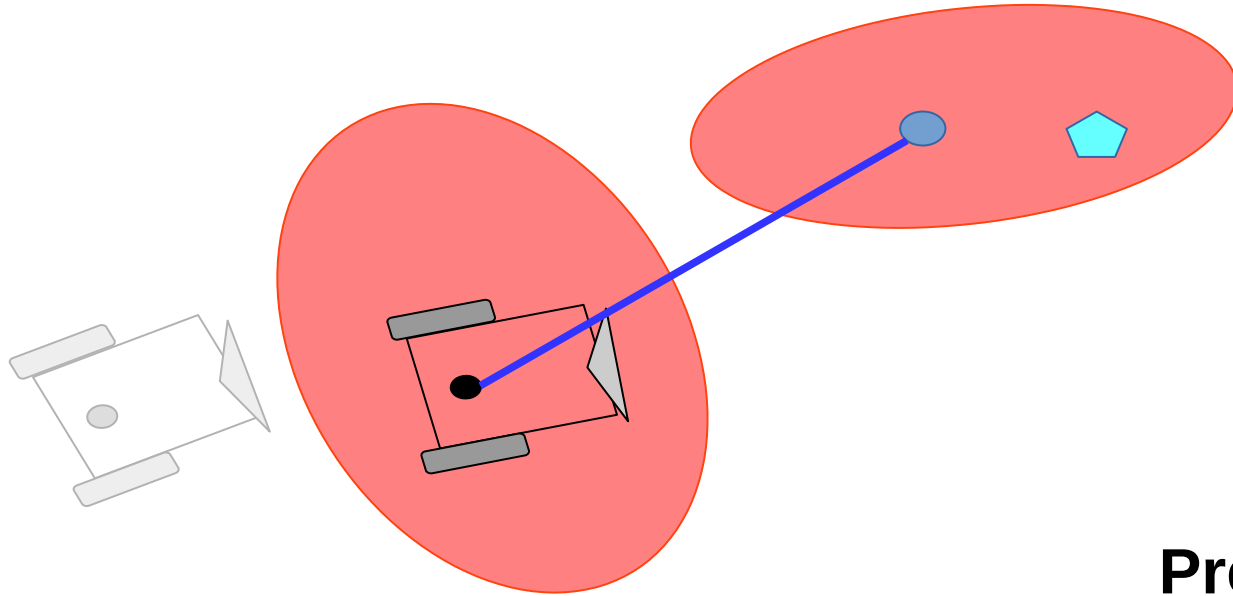


SLAM para robot diferencial (basado en el EKF)



Predicción
Se aplica el modelo dinámico

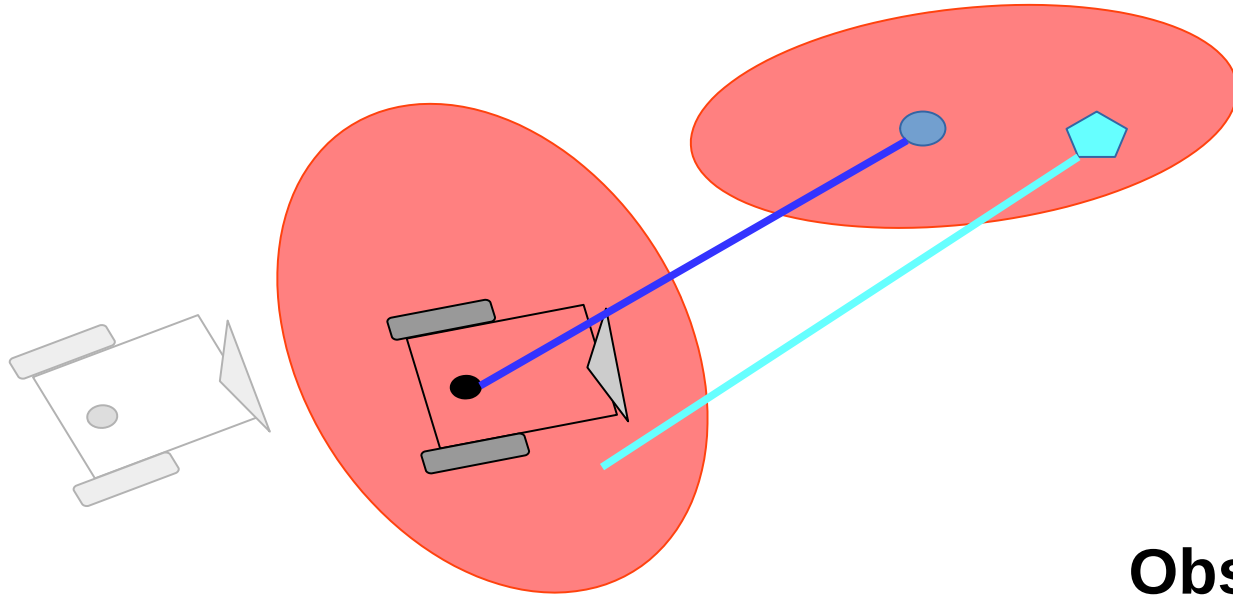
SLAM para robot diferencial (basado en el EKF)



Predicción

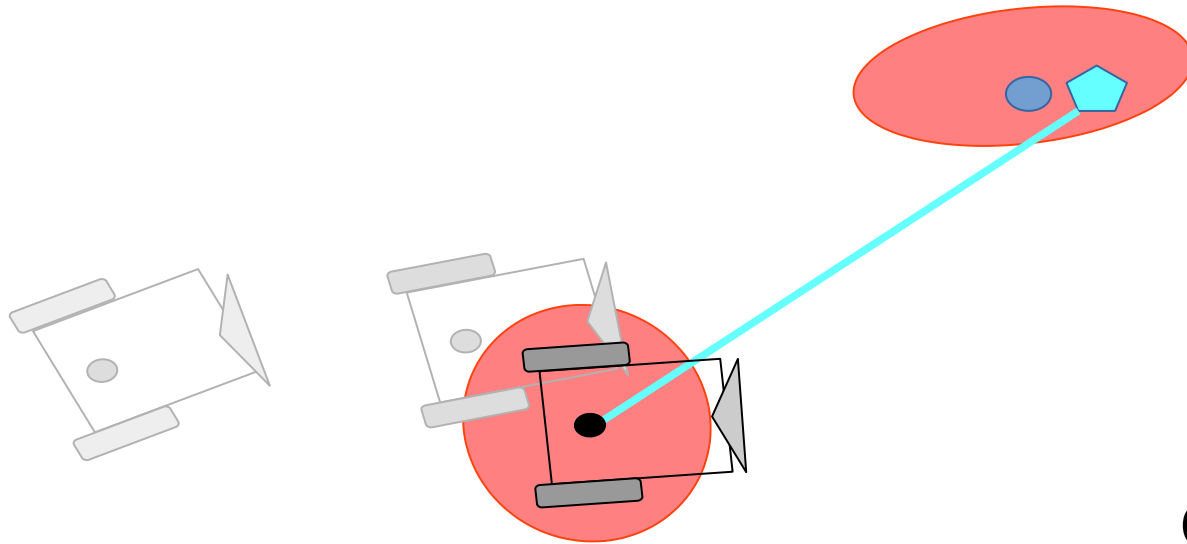
Se aplica el modelo de medición

SLAM para robot diferencial (basado en el EKF)

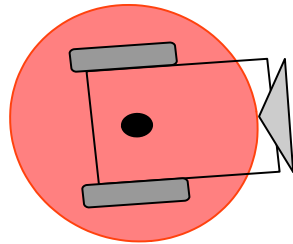
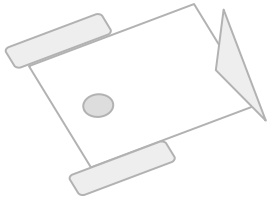
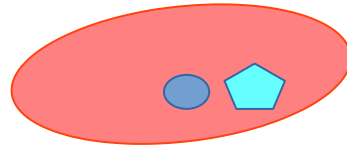


Observación
Se obtienen mediciones
de los sensores

SLAM para robot diferencial (basado en el EKF)



SLAM para robot diferencial (basado en el EKF)

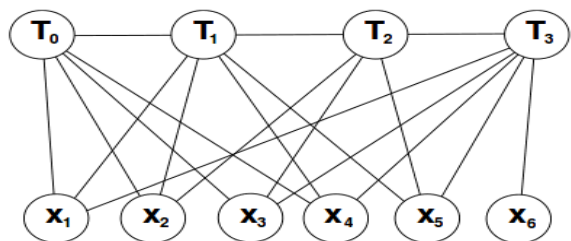


Se empieza otra vez

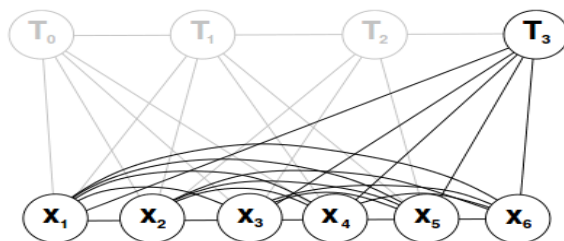
Componentes en SLAM

- Modelo dinámico.
- Modelo de medición.
- Asociación de datos.
- Mantenimiento del mapa.
 - Creación de nuevos componentes.
 - Eliminación de nuevos componentes.
- Cierre de bucle.

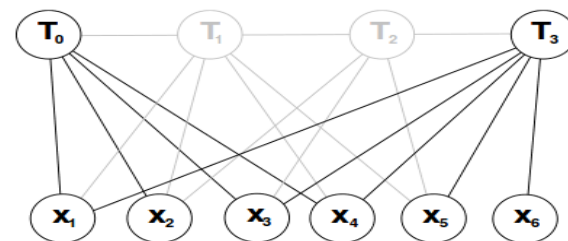
SLAM



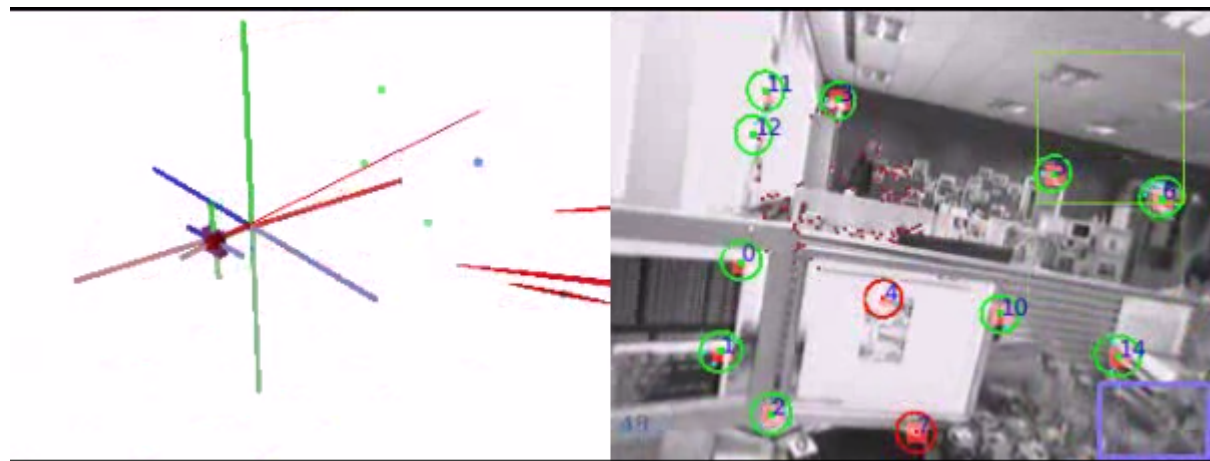
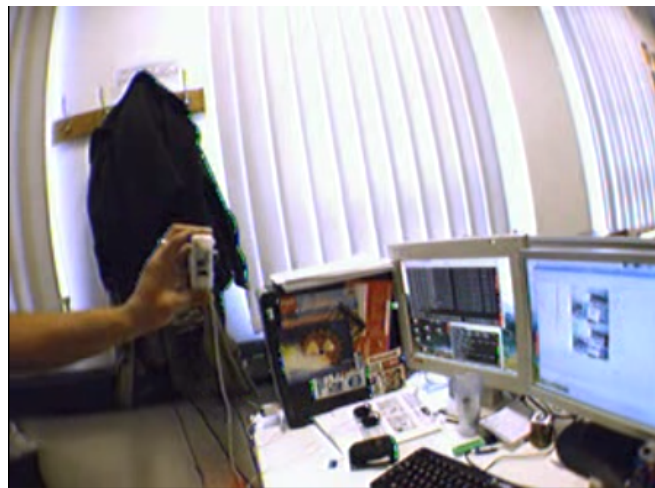
(a) Markov Random Field

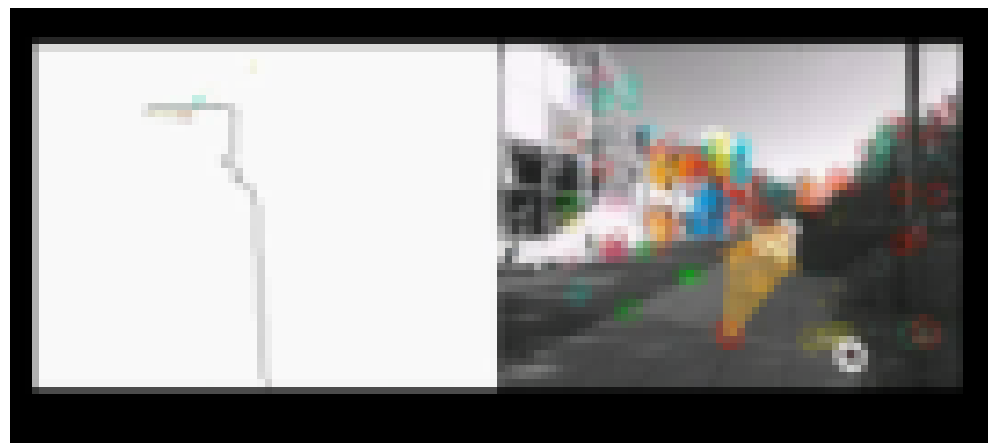
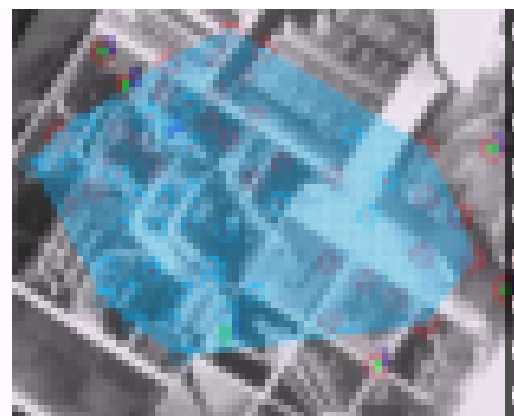
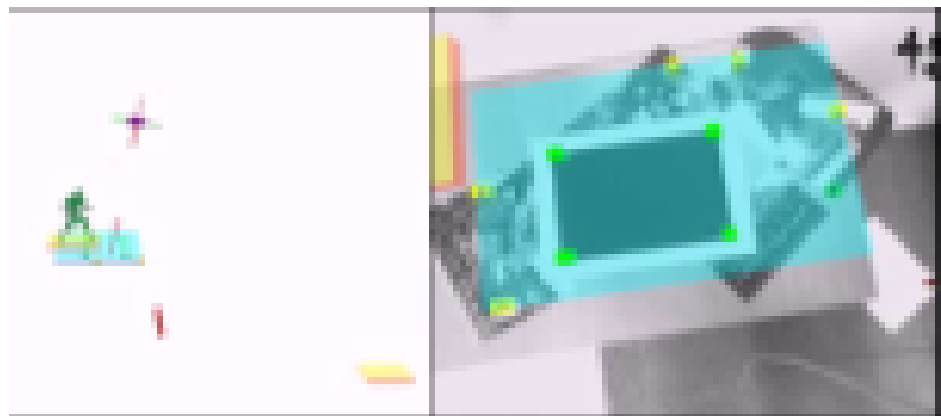


(b) Filter



(c) Keyframe BA





SLAM Monocular (Visual SLAM)

- MonoSLAM: Primer sistema que utilizó el marco teórico de SLAM en el dominio de visión computacional puro.
- Localización y Mapeo Simultaneo con una cámara con movimiento libre (6D), sin controlador, sin marcas en el ambiente.
- Desarrollado y publicado en 2003 por **Andrew Davison** (actualmente Profesor en Imperial College) (Davison, 2003). A la fecha este paper cuenta con 1369 citas.
- Su trabajo en general cuenta con 9693 citas (6603 desde 2010). Fuente: google scholar.
- En 2010 publico un paper titulado “¿Por qué el filtro?”, con el que argumenta que el filtro (EKF) no es la mejor opción (Davison,2010).

SLAM Monocular

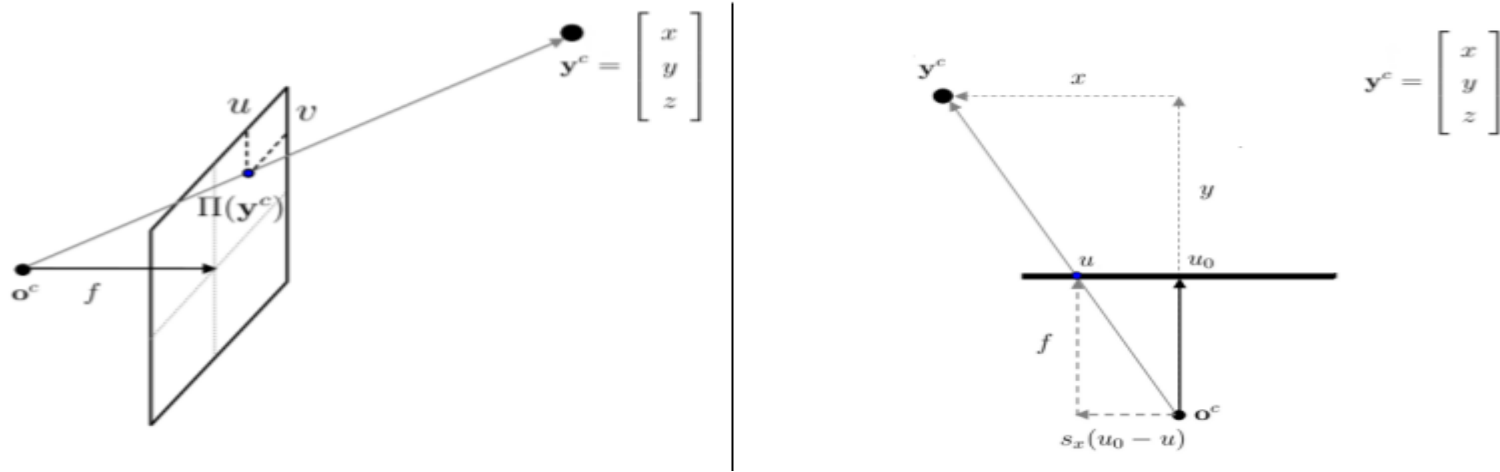
- Modelo dinámico de la cámara (velocidad constante):

$$\mathbf{x}_{c_{k+1}} = \begin{bmatrix} \mathbf{r}_{k+1}^w \\ \mathbf{q}_{k+1}^w \\ \mathbf{v}_{k+1}^w \\ \omega_{k+1}^c \end{bmatrix} = \mathbf{f}_c(\mathbf{x}_{c_k}, \mathbf{n}) = \begin{bmatrix} \mathbf{r}_k^w + (\mathbf{v}_k^w + \nu)\Delta t \\ \mathbf{q}_k^w \otimes \mathbf{q}((\omega_k^c + \Omega)\Delta t) \\ \mathbf{v}_k^w + \nu \\ \omega_k^c + \Omega \end{bmatrix}$$

$$\mathbf{n} = \begin{bmatrix} \nu \\ \Omega \end{bmatrix} = \begin{bmatrix} \eta\Delta t \\ \alpha\Delta t \end{bmatrix}$$

SLAM Monocular

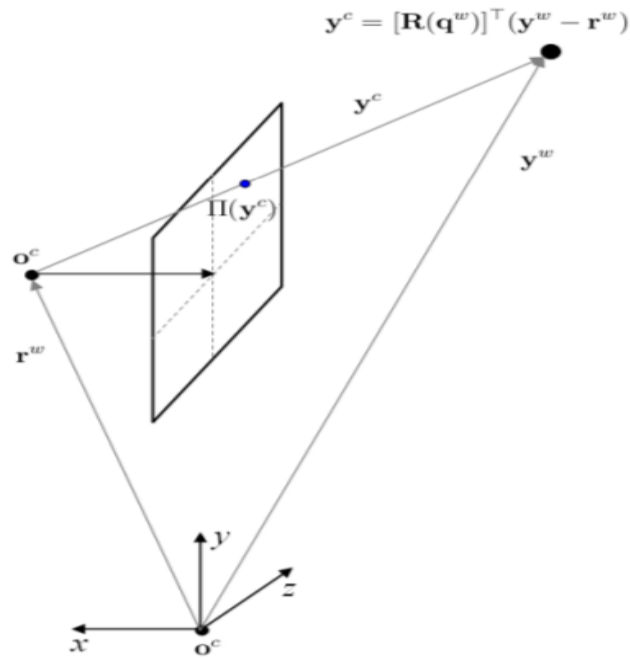
● Modelo de medición (pin-hole):



$$\begin{bmatrix} u \\ v \end{bmatrix} = \Pi(\mathbf{y}^c) = \begin{bmatrix} u_0 - \frac{fx}{s_x z} \\ v_0 - \frac{fy}{s_y z} \end{bmatrix}$$

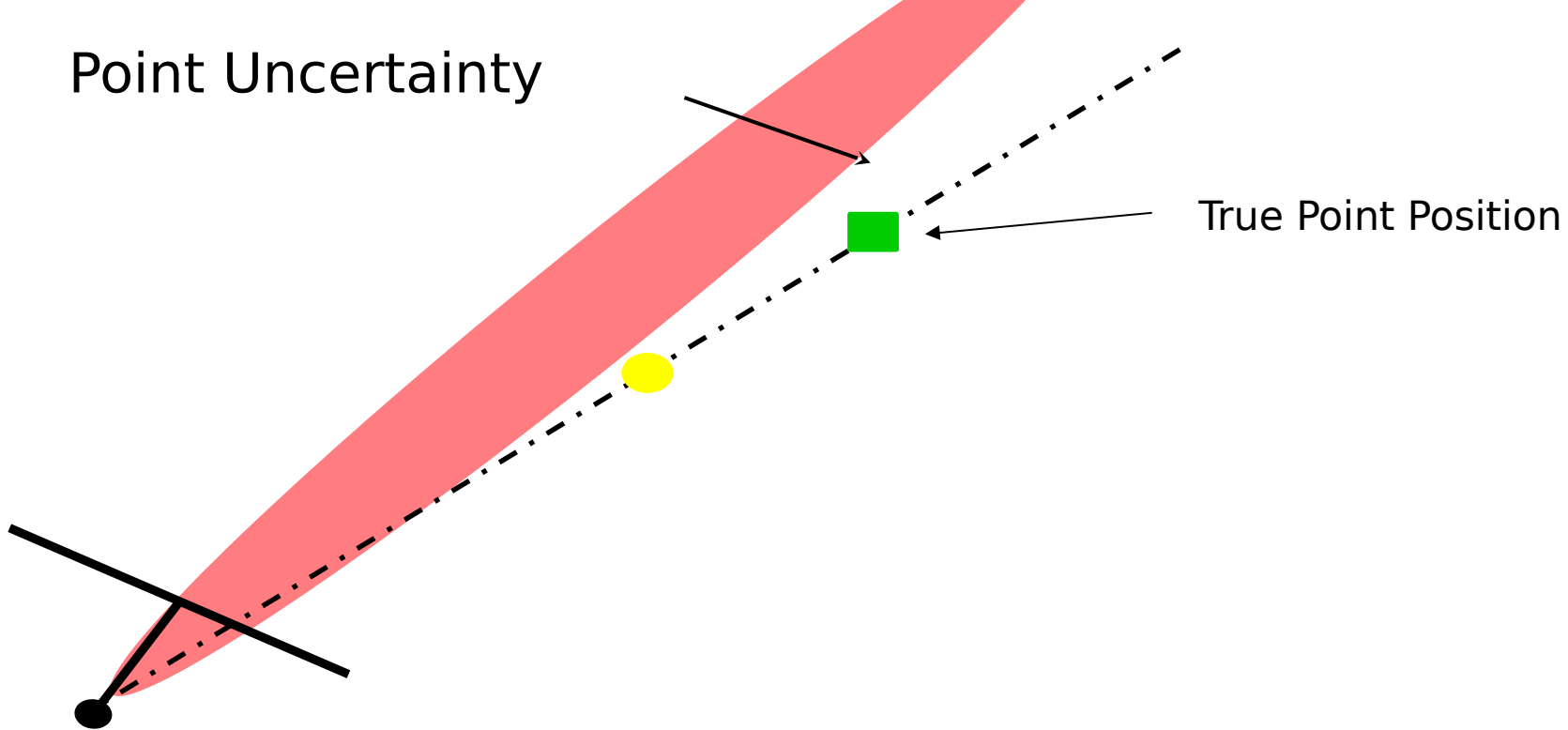
SLAM Monocular

- Modelo de medición (pin-hole):



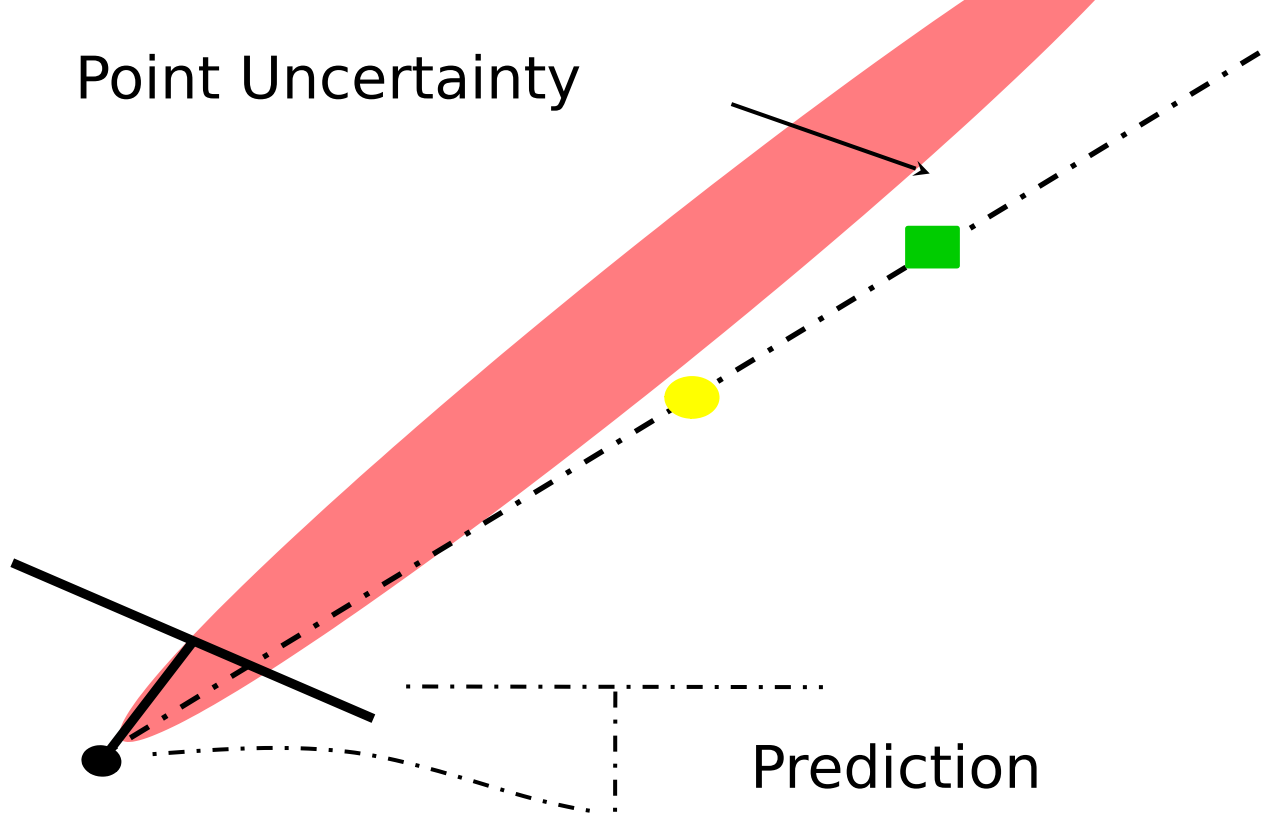
$$\begin{bmatrix} u \\ v \end{bmatrix} = \Pi(y^c) = \begin{bmatrix} u_0 - \frac{fx}{s_x z} \\ v_0 - \frac{fy}{s_y z} \end{bmatrix}$$

Point Uncertainty



True Point Position

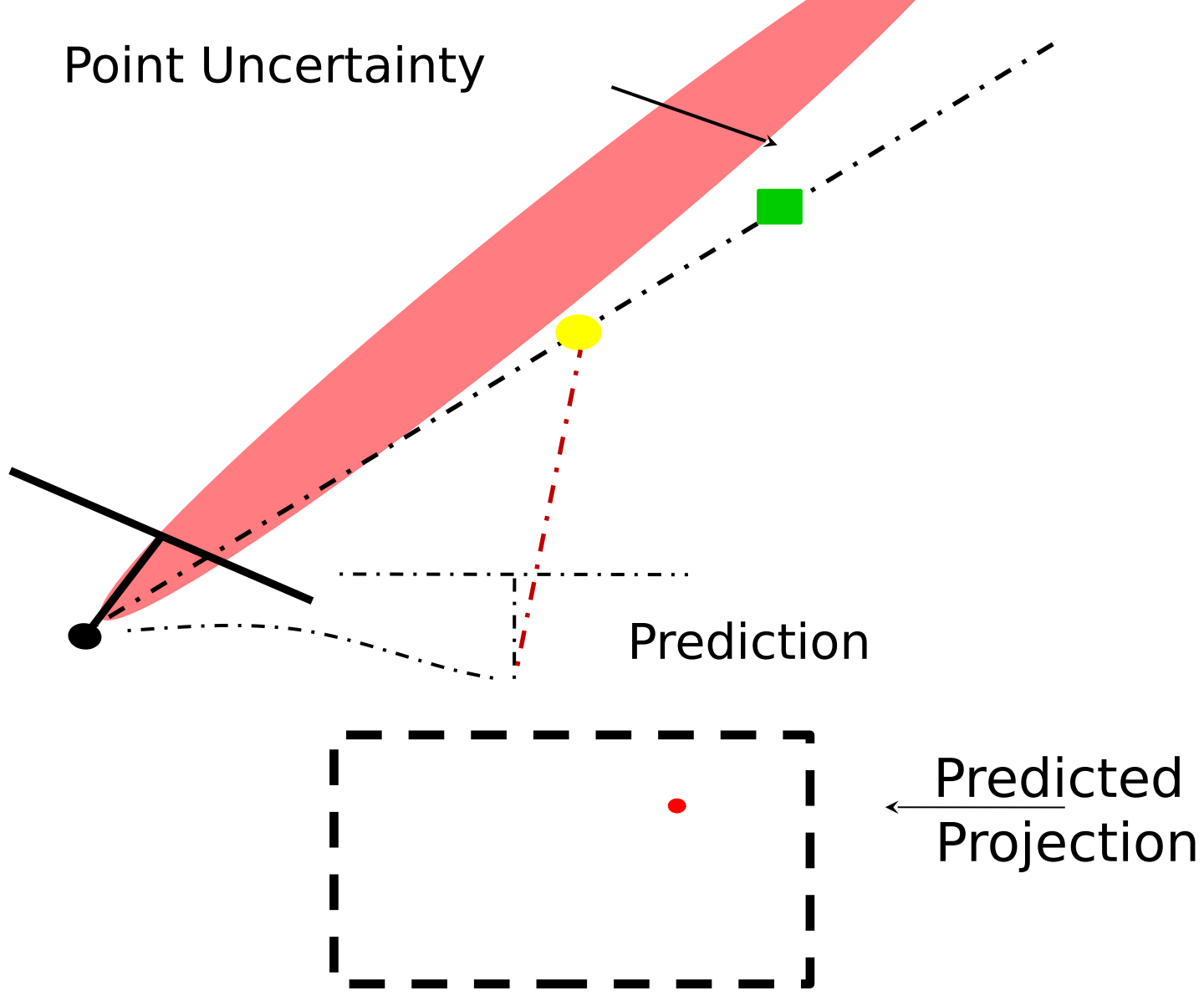
Point Uncertainty



Prediction



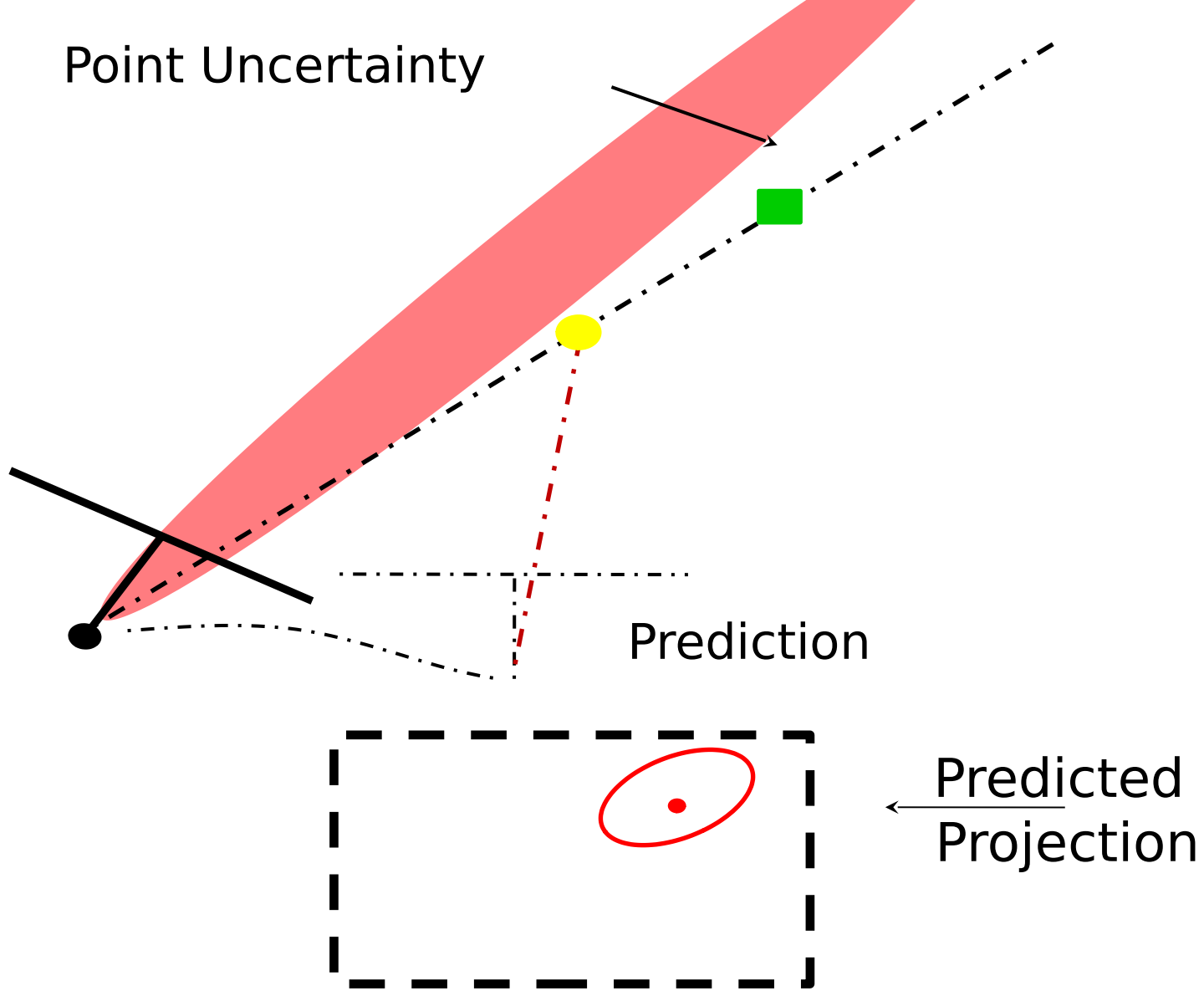
Point Uncertainty



Prediction

Predicted
Projection

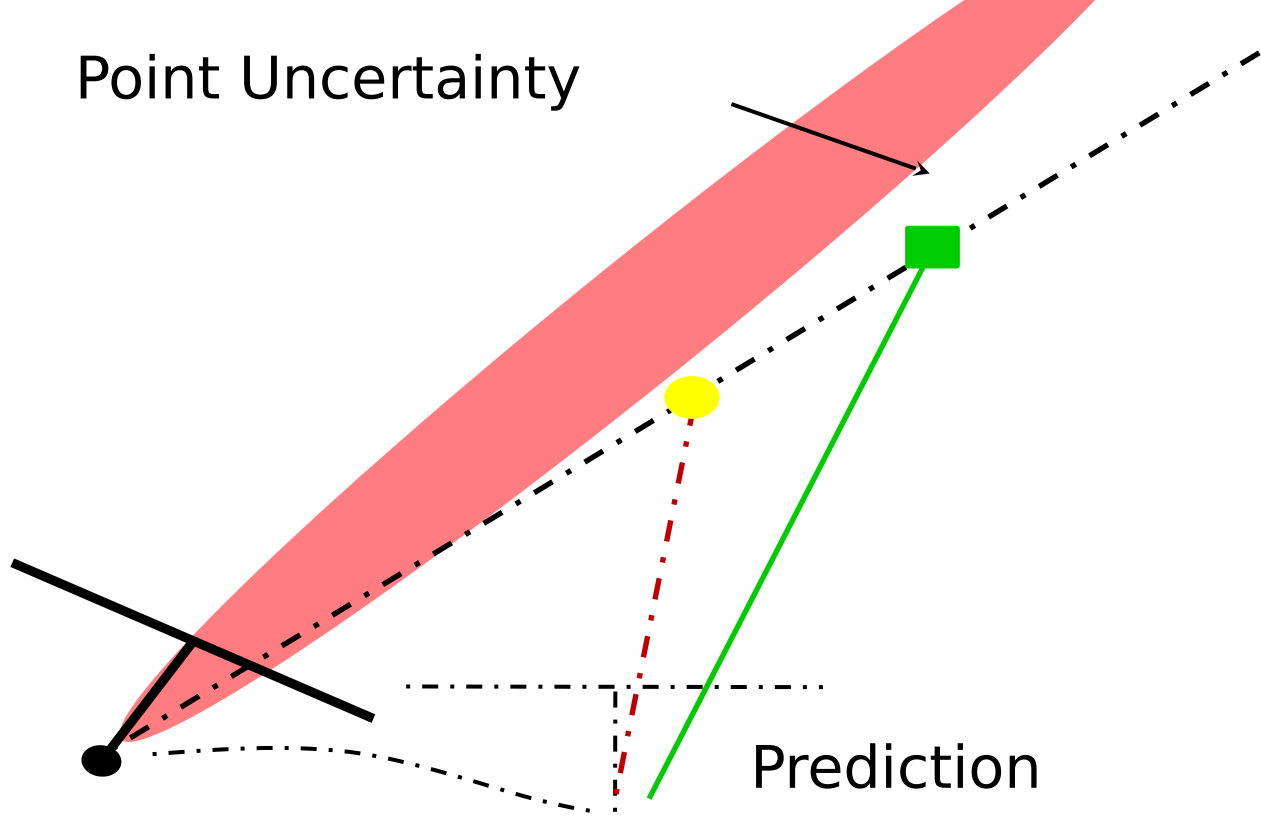
Point Uncertainty



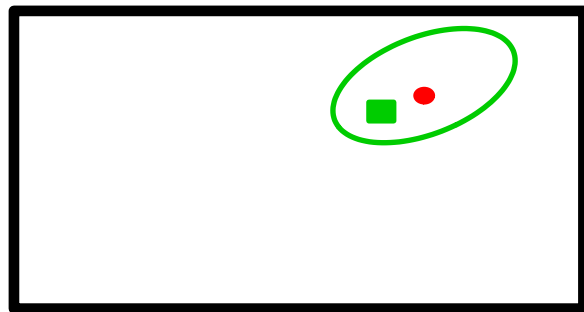
Prediction

Predicted
Projection

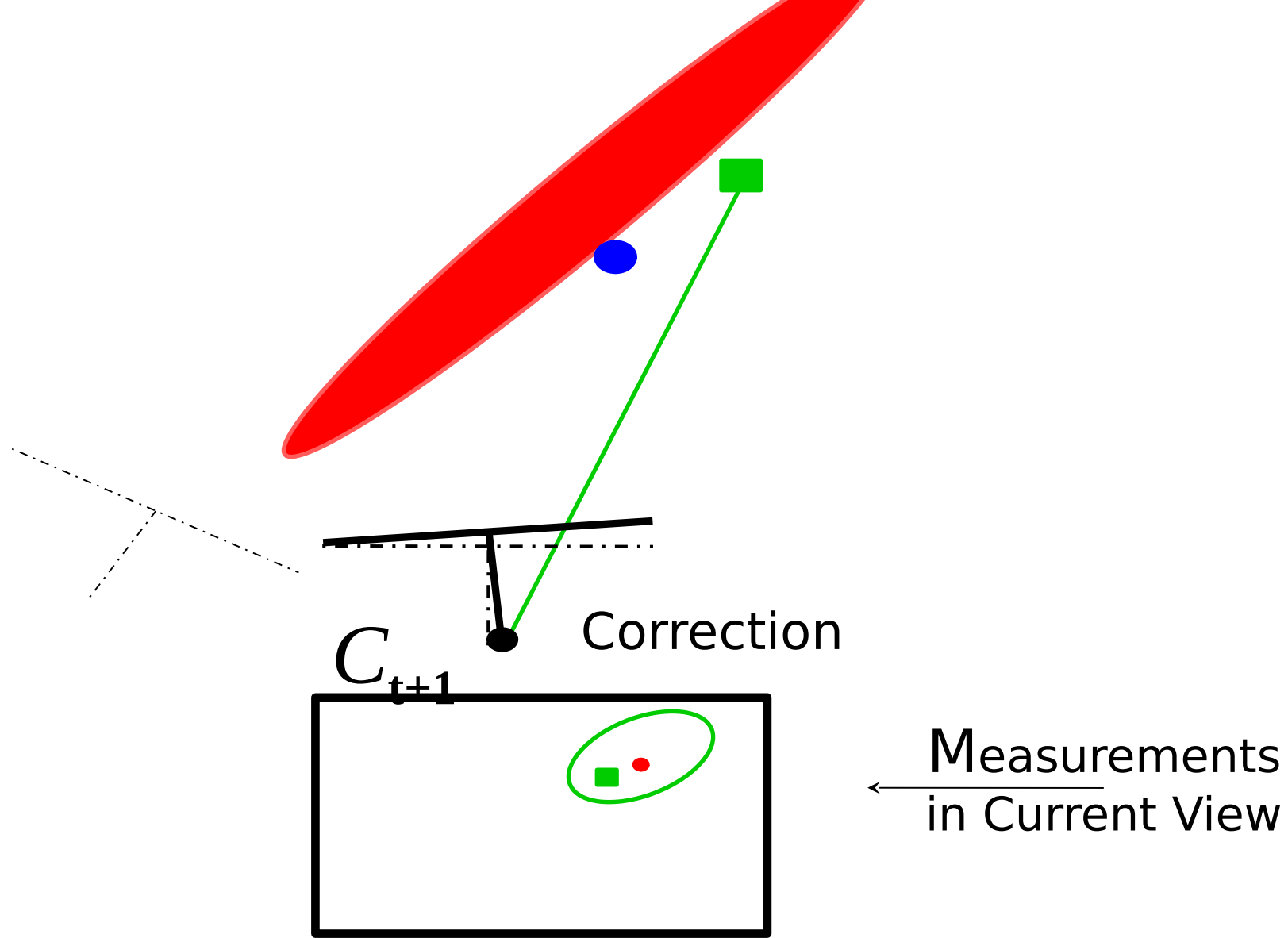
Point Uncertainty

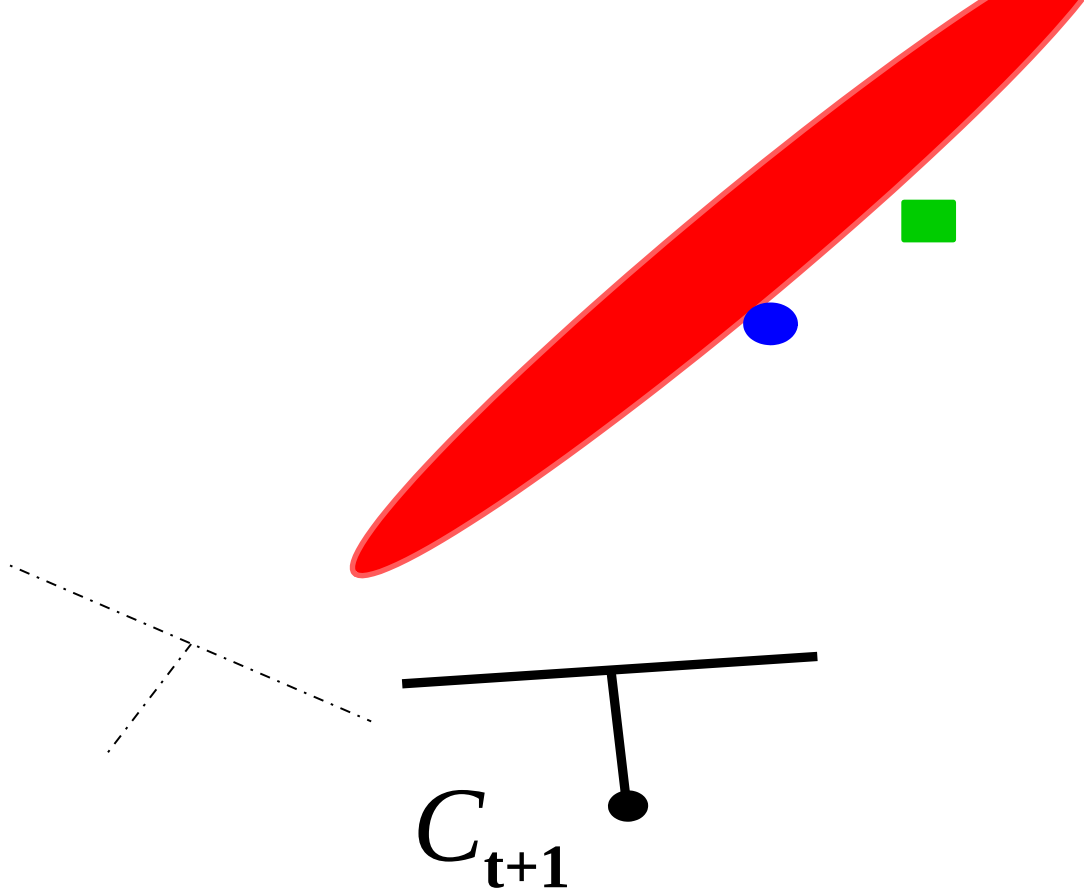


Prediction



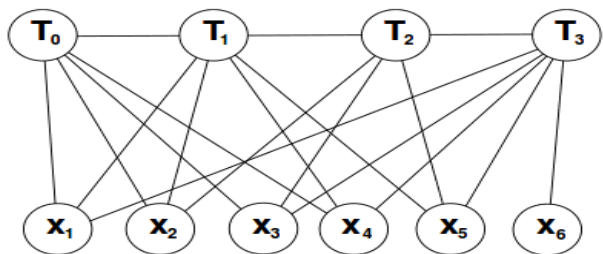
← Measurements
in Current View



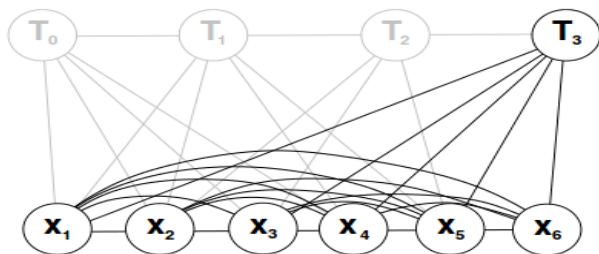


Visual SLAM

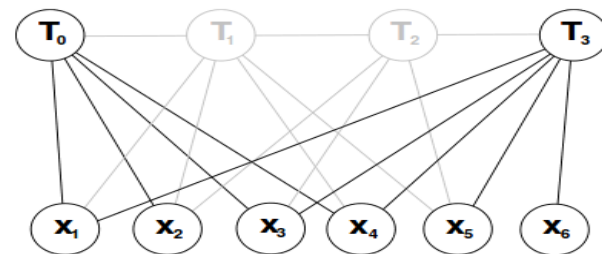
$$\chi^2(\mathbf{y}) = \sum_{\mathbf{z}_{i,j} \in \mathcal{Z}_{0:i}} (\mathbf{z}_{i,j} - \hat{\mathbf{z}}(\mathbf{T}_i, \mathbf{x}_j))^2$$



(a) Markov Random Field



(b) Filter

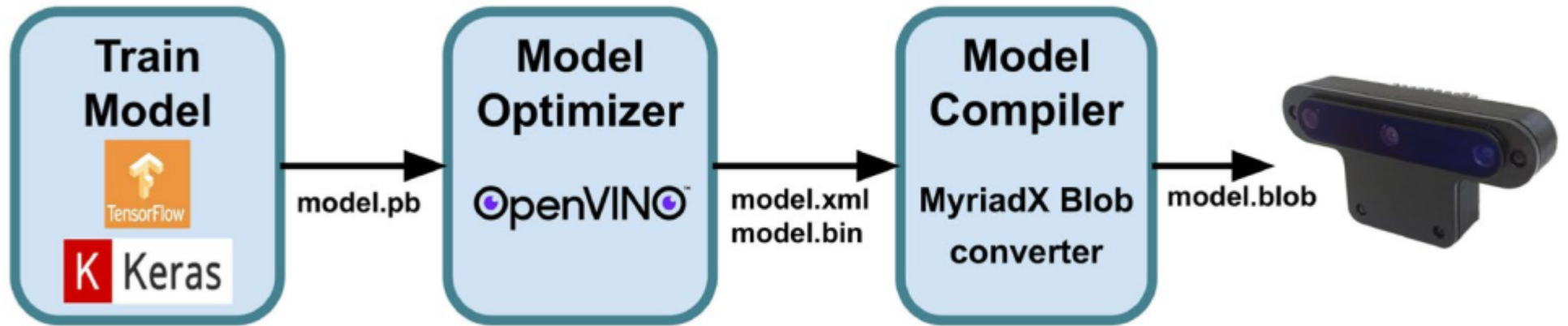


(c) Keyframe BA

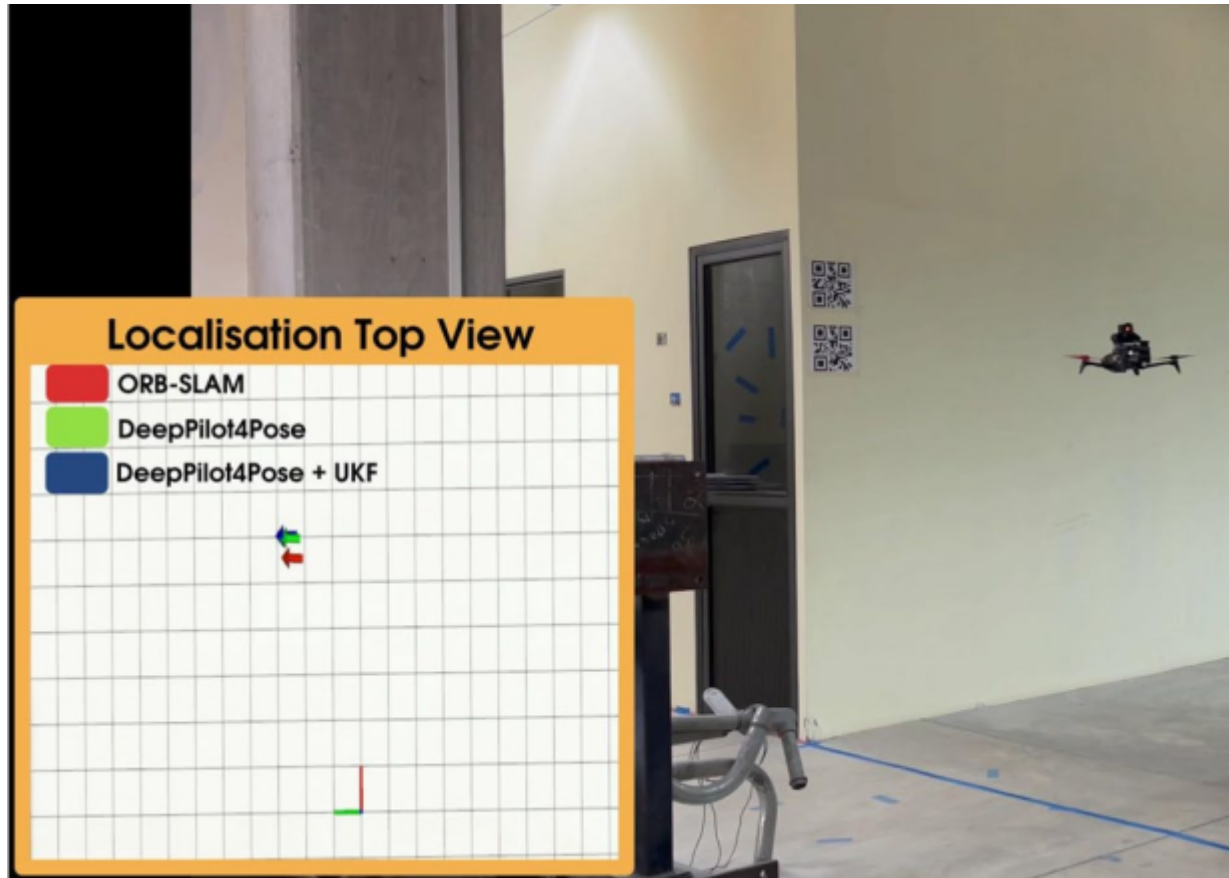
Localización Neuronal



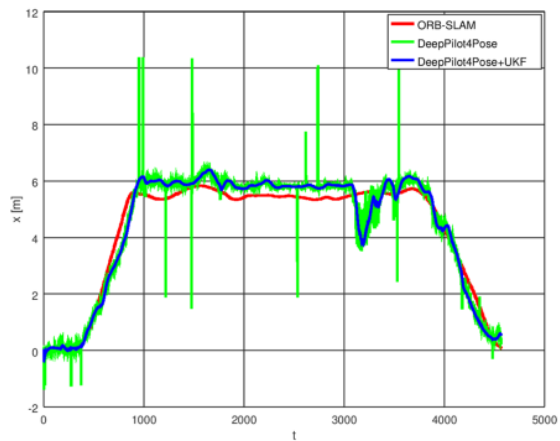
Localización Neuronal



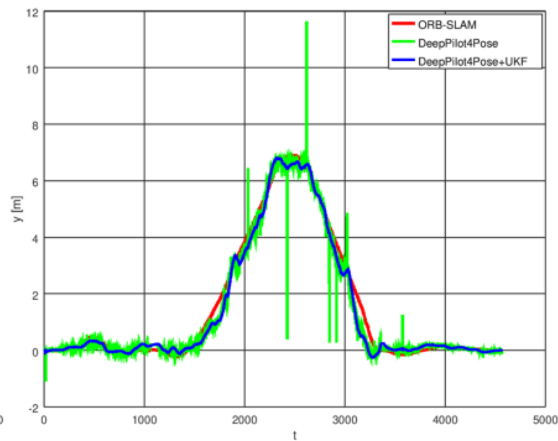
Localización Neuronal



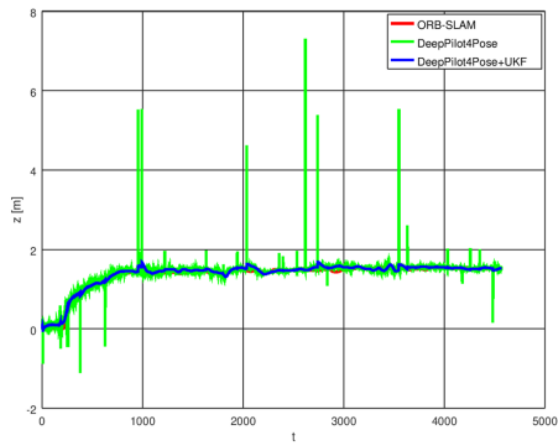
Localización Neuronal



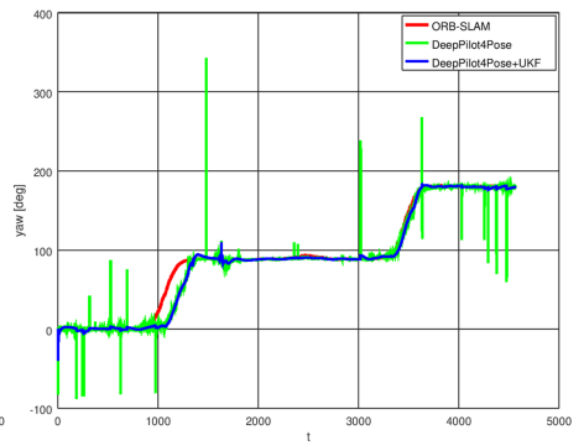
(a)



(b)



(c)



(d)

Referencias

1. Durrant-Whyte, H. and Bailey, T. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. IEEE ROBOTICS AND AUTOMATION MAGAZINE, 2006.
2. Bailey, T. and Durrant-Whyte, H. "Simultaneous localization and mapping (SLAM): Part II." IEEE Robotics & Automation Magazine 13.3 (2006): 108-117.
3. Welch, G., & Bishop, G. An introduction to the kalman filter. Proceedings of the Siggraph Course, Los Angeles. 2001.
4. S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics. MIT Press, 2005.
5. Davison, A. Real-Time Simultaneous Localisation and Mapping with a Single Camera. International Conference in Computer Visión, 2003.
6. Javier Civera, Óscar G. Grasa, Andrew J. Davison, J. M. M. Montiel
7. Civera, J. 1-Point RANSAC for EKF-Based Structure from Motion. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2009), 2009.
8. Neira, J. Tardos, J.D. Data association in stochastic mapping using the joint compatibility test, Robotics and Automation, IEEE Transactions, 2001.
9. Civera, J. Davison, A. and Montiel, J. M. M. Inverse Depth Parametrization for Monocular SLAM. IEEE Transactions on Robotics, 2008.
10. Strasdat, H. Montiel, J.M.M. and Davison, A. Why Filter? International Conference on Robotics and Automation, 2010.
11. Meilland, M., & Comport, A. I. On unifying key-frame and voxel-based dense visual SLAM at large scales. In Intelligent Robots and Systems (IROS), 2013.
12. Rojas-Perez, L. O., & Martinez-Carranza, J. (2023). DeepPilot4Pose: a fast pose localisation for MAV indoor flight using the OAK-D camera. Journal of Real-Time Image Processing, 20(1), 8.