



# FACULTAD DE INGENIERIA

Universidad de Buenos Aires

UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE INGENIERÍA

Año 2022 - 1<sup>er</sup> Cuatrimestre

LOCALIZACIÓN: FILTRO DE PARTÍCULAS Y EKF.  
ROBÓTICA MÓOVIL - UN ENFOQUE PROBABILÍSTICO

1er. Cuatrimestre

Curso: MAS

INTEGRANTES:

Fuentes Acuña, Brian Alex

- #101785

<bfuentes@fi.uba.ar>

# Índice

<b>1. Filtro de particulas</b>	<b>2</b>
<b>2. Filtro de Kalman Extendido</b>	<b>4</b>
<b>3. Conclusión</b>	<b>7</b>

# 1. Filtro de partículas

Un filtro de partículas representa eficientemente distribuciones no gaussianas(no paramétricas) y su principio básico radica en :

- Conjunto de estado hipótesis (Partículas = muestras).
- Supervivencia del mas apto.

El filtro de partículas es una implementación alternativa al filtro de bayes pero es del tipo *no-parametrico*. La idea de este filtro es representar la belief  $bel(x)$  por un conjunto de muestras esparcidas aleatoriamente y ponderadas con un peso de importancia.

Entonces cada partícula es un posible estado en el cual se puede encontrar el sistema. Cuanto más poblada se encuentra una subregión del espacio de estados por partículas, entonces, es más probable que el verdadero estado caiga dentro de dicha región.

Un filtro de partículas consiste principalmente de tres pasos:

- 1. Muestrear nuevas partículas usando el modelo de movimiento.
- 2. Calcular el peso de las nuevas partículas usando el modelo de medición.
- 3. Calcular el nuevo belief remuestreando las partículas de manera proporcional a sus pesos, con reemplazo.

El primer paso fue parte de la practica 2 por lo que ya está explicado.  
El segundo paso:

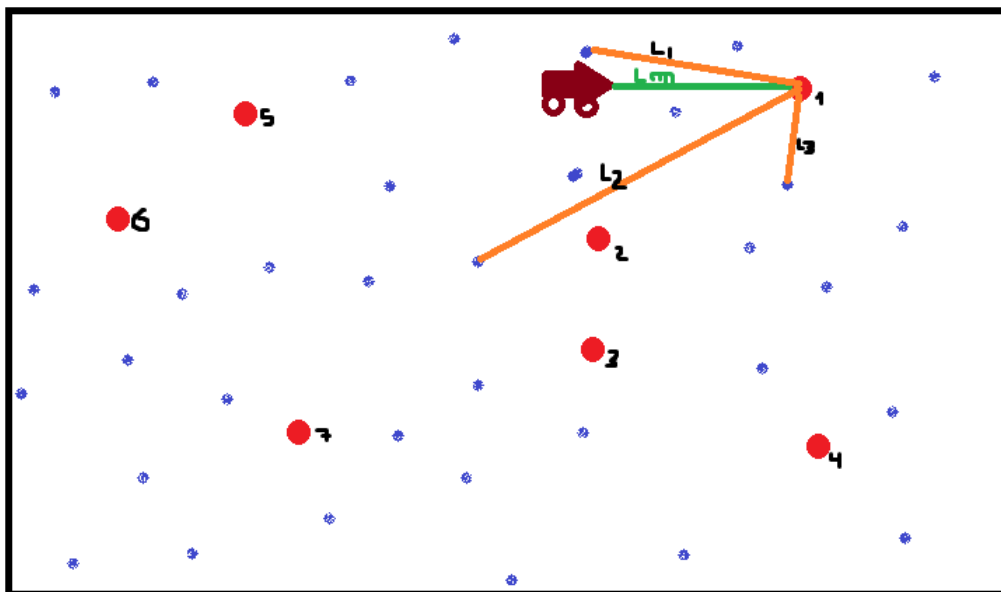


Figura 1: Robot móvil midiendo distancias al landmark

Para calcular el peso de las nuevas partículas (puntos azules) usando el modelo de medición se hace calculando a belief como el likehook, ósea  $P(z|x,l)$  el cual se asume gaussiana de media nula

y varianza dada, para calcular la probabilidad pedida.

Entonces usando la función  $\text{normpdf}(x, \mu, \sigma)$  se calculan los pesos para cada partícula (posibles estados) pero para esto hay que terminar quien es  $x$ . Este será la diferencia entre la distancia medida y la esperada, por ejemplo:  $L_m - L_1$ , y asumiendo que las probabilidades son independientes se puede calcular dicha probabilidad.

Ahora hay que tener en cuenta que en nuestro caso el modelo de medición no solo mide distancia al landmark sino también el landmark que está midiendo (de alguna forma) pero además puede medir más de un landmark a la vez por lo que para cada set de mediciones voy a tener que obtener un peso para cada partícula, usando la independencia simplemente multiplico los pesos de un set de mediciones.

Para el tercer paso se hace uso del muestreo estocástico universal (SUS).

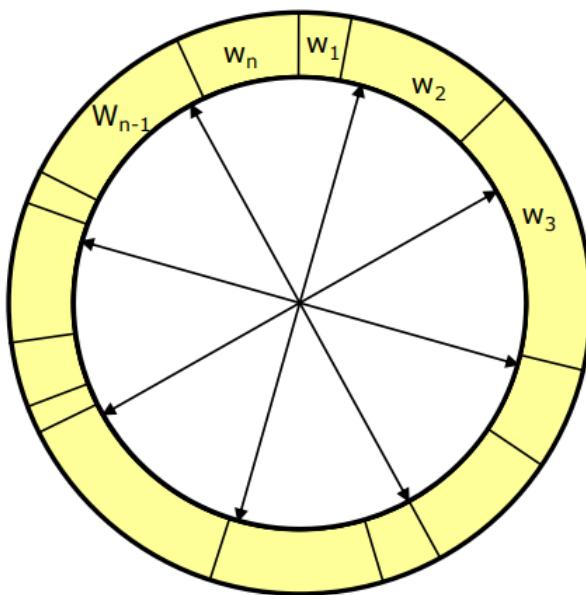
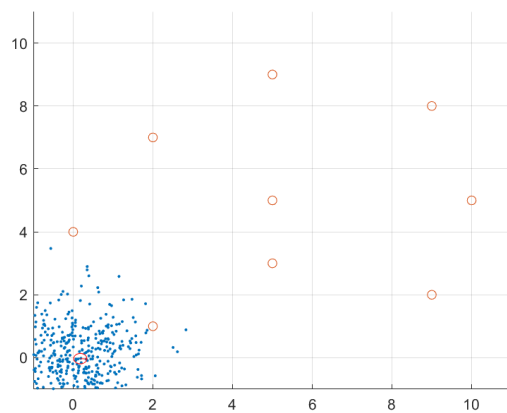
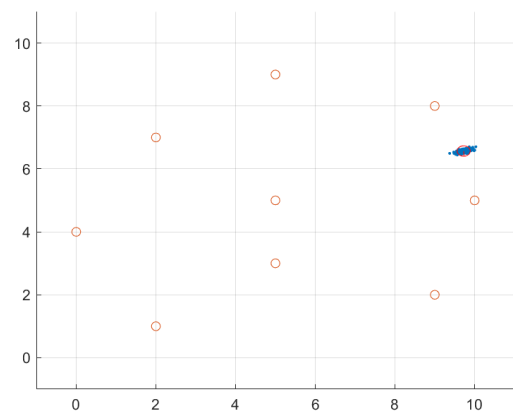


Figura 2: Muestreo estocástico universal

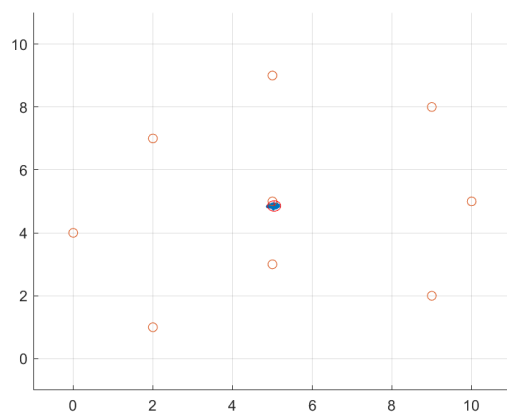
El cual consiste distribuir aleatoriamente los pesos e ir seleccionando cada  $1/n$ , de esta forma se reduce la complejidad ( $O(n)$ ) del remuestreo y se puede ver que es más probable caer en muestras que tienen mayor peso, incluso dos veces, que, en muestras con poco peso, incluso se las puede saltar, así "filtrándolas".



(a) Inicio de la trayectoria



(b) Mitad de trayectoria



(c) Fin de trayectoria

Figura 3: Movimiento del robot con PF

Luego de la implementación se ha simulado y se observan 3 imágenes la primera es el inicio, en donde se ve que en vez de poner partículas en todo el mapa se han iniciado aleatoriamente alrededor de donde se sabía que estaba el robot inicialmente, así acelerando la convergencia de los estados, esto es muy eficiente dado que si se conoce la ubicación inicial del robot se puede ahorrar tiempo de cómputo.

Luego el robot irá moviéndose de acuerdo al modelo de movimiento, los estados con 500 partículas convergen todos cercanos a la posición del robot en cada instante, aunque es interesante ver que en el caso b) las partículas se van dispersando más, ósea la varianza aumenta, eso es porque en esos instante hay pocas mediciones entonces la varianza debe crecer, luego habrá más mediciones y las partículas prosiguen con menor varianza hasta así llegar al estado final caso c).

## 2. Filtro de Kalman Extendido

El filtro de Kalman asume problemas lineales y distribuciones gaussianas, un modelo paramétrico. Pero en la práctica los problemas no tienen por qué ser lineales, de hecho no lo son, entonces aparece

el filtro de Kalman extendido que asume que la probabilidad del próximo estado y las probabilidades de las observaciones están gobernadas por funciones no lineales  $g$  y  $h$  respectivamente.

$$x_t = g(u_t, x_{t-1}) + \epsilon_t$$

$$z_t = h(x_t) + \delta_t$$

Con  $\epsilon_t$  y  $\delta_t$  ruidos del proceso y medicion respectivamente, independiente.

La idea es hacer una **linealización local** de dichas funciones y aproximar, por supuesto esta linealización va a ser buena o mala dependiendo la varianza del sistema.

Entonces el algoritmo a usar será el siguiente:

## 1. **Extended\_Kalman\_filter**( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ ):

### 2. Predicción:

$$3. \quad \bar{\mu}_t = g(u_t, \mu_{t-1}) \quad \longleftarrow \quad \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$$

$$4. \quad \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + Q_t \quad \longleftarrow \quad \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$$

### 5. Corrección:

$$6. \quad K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1} \quad \longleftarrow \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}$$

$$7. \quad \mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t)) \quad \longleftarrow \quad \mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$$

$$8. \quad \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \quad \longleftarrow \quad \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

### 9. **Return** $\mu_t, \Sigma_t$

$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \quad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$

Figura 4: Algoritmo del EKF

En donde se ve que también aparece un paso de corrección, al igual que en el caso del filtro de partículas, el análisis es el mismo solo que acá se trabaja directamente con las medias y la varianza, entonces directamente la posición estará dada por la media.

el modelo de movimiento está dado por la odometria:

$$g(\mu_t, u_{t-1}) = \begin{pmatrix} \mu_x + \delta_{trans} \cdot \cos(\theta + \delta_{rot1}) \\ \mu_y + \delta_{trans} \cdot \sin(\theta + \delta_{rot1}) \\ \mu_\theta + \delta_{rot1} + \delta_{rot2}; \end{pmatrix} \quad (1)$$

y para la aproximacion lineal se usa el jacobiano, quedando:

$$G = \begin{pmatrix} 1 & 0 & -\delta_{trans} \cdot \sin(\delta_{rot_1} + \mu_\theta) \\ 0 & 1 & \delta_{trans} \cdot \cos(\delta_{rot_1} + \mu_\theta) \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

Para la correccion hay que tener en cuenta que solo se mide el rango por lo que la funcion de medicion sin ruido es:

$$h(m, x) = (\sqrt{(m_x - x)^2 + (m_y - y)^2}) \quad (3)$$

Luego la prediccion de la medicion será:

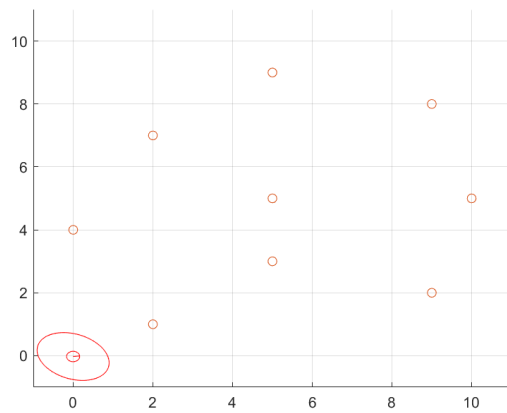
$$\hat{z}_t(m, \mu) = (\sqrt{(m_x - \mu_x)^2 + (m_y - \mu_y)^2}) \quad (4)$$

y su jacobiano con respecto a la pose:

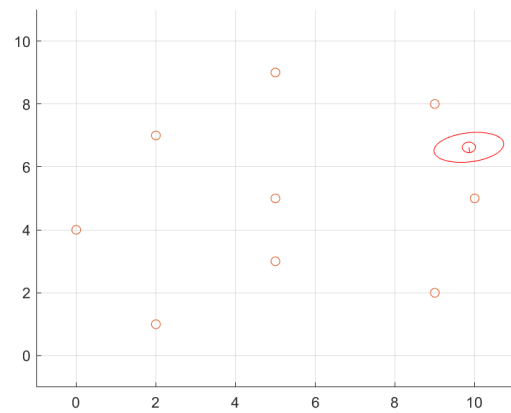
$$H(m, \mu) = \begin{pmatrix} \frac{(\mu_x - m_x)}{\sqrt{(m_x - \mu_x)^2 + (m_y - \mu_y)^2}} & \frac{(\mu_y - m_y)}{\sqrt{(m_x - \mu_x)^2 + (m_y - \mu_y)^2}} & 0 \end{pmatrix} \quad (5)$$

Luego hay que tener en cuenta que para cada medición se obtiene una matriz H, puesto que puede haber n mediciones de landmark a tiempo t, la matriz H será de nx3.

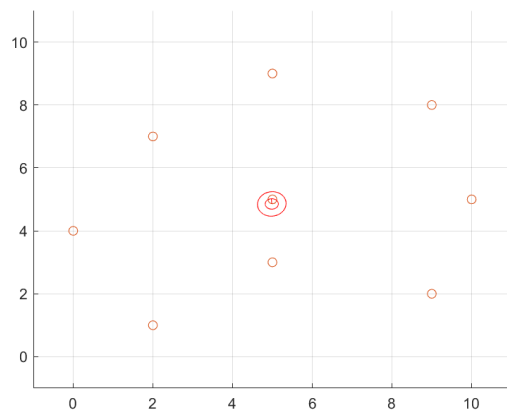
Simulando el modelo se obtiene lo siguiente:



(a) Inicio de la trayectoria



(b) Mitad de trayectoria



(c) Fin de trayectoria

Figura 5: Movimiento del robot con EKF

Notar que en este método lo que en realidad se ve son gaussianas, en donde la media es la posición del robot y se ira graficando la varianza a cada paso de iteración, empezando en el instante inicial con gran varianza y luego esta ira reduciendo con el paso de corrección, es así como nuevamente en el caso b) se puede ver como se tiene gran varianza, nuevamente es porque en esos instantes hay pocas mediciones por lo que la corrección es poca. luego el robot sigue su movimiento hasta terminar en el caso c) donde ya tiene poca varianza dado que en esa posición tiene muchas buenas mediciones de los landmarks.

### 3. Conclusión

Si bien no se ve reflejado en los gráficos, en general el EKF funciona muy bien para no-linealidades moderadas, pero suele requerir menos poder de cómputo que el filtro de partículas justamente porque se lo linealiza, esto lo hace bastante eficaz para ser usado siempre que no nos alejemos demasiado de las hipótesis iniciales.

El filtro de partícula si bien a mayor cantidad de partículas y tamaño del mapa, puede requerir mucho



tiempo de cómputo, este tiene la enorme ventaja de que trabaja con distribuciones no gaussianas no tiene que linealizar nada, funciona siempre que tenga las partículas necesarias. y esta es una de sus desventajas.

El PF depende mucho de las partículas que se le asigne puesto que a mayor estados hipótesis más posibilidades se tiene de que entre ellos esté la verdadera POSE en caso contrario simplemente no hay "buenas partículas" candidatas y el sistema no converge.