

Comparação entre o Filtro de Kalman e Filtro de Partículas Aplicados na Robótica Móvel

Adalberto Magalhães Malcher da Silva Neto*

*Departamento de Sistemas e Controle de Energia, Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, netomalcher@gmail.com

Resumo—A inteligência artificial dentro do escopo da ciência cognitiva é uma área distinta da robótica, porém ciência cognitiva e robótica são áreas complementares. O objetivo do trabalho é mostrar um exemplo da arquitetura de subsunção de Rodney Brooks através de um robô interagindo com o meio ambiente através de sensores para realizar sua localização. O Filtro de Kalman Estendido (FKE) e o Filtro de Partículas (FP) são os métodos utilizados na programação do robô e seus desempenhos e limitações foram comparadas.

Index Terms—Filtro de Kalman, Filtro de Partículas, Inteligência Artificial, Robótica.

1. INTRODUÇÃO

A partir da metade da década de 80 houve uma grande ruptura nas pesquisas de inteligência artificial causada por Rodney Brooks (MIT) por afirmar que os ambientes artificiais não são o suficiente para o desenvolvimento e resolução dos problemas encontrados na área de inteligência artificial. Rodney Brooks passa a utilizar a robótica como ambiente de desenvolvimento e a une com inteligência artificial através de sensores e atuadores trabalhando em conjunto para interagirem e aprenderem com o meio ambiente, ou seja, a premissa para a arquitetura de subsunção [1].

Para exemplificar o funcionamento da arquitetura de subsunção de Brooks, o presente trabalho utilizou um robô móvel em um ambiente para poder se localizar utilizando duas técnicas distintas, o Filtro de Kalman Estendido (FKE) [2][3][4] e o Filtro de Partículas (FP) [5].

Os sensores laser do robô fazem a leitura do ambiente para poder situa-lo baseado nas interseções das paredes (*landmarks*). Cada *landmark* é composto por coordenadas (x,y) e possuem características únicas denominadas de *features*, onde cada feature é composta por uma distância e um ângulo θ .

Para implementação do FKE os *landmarks* e *features* são incorporados ao filtro para determinar a pose do robô no ambiente. O FP utiliza os *landmarks* e *features* para referencia do robô real em relação à localização e orientação das partículas criadas.

A seção 2 do trabalho descreve a teoria sobre o Filtro de Kalman Estendido e como foi realizada a sua implementação. A seção 3 apresenta a teoria do Filtro de Partículas para a

aplicação no robô. A seção 4 mostra os resultados obtidos nas simulações e, finalmente, a seção 5 apresenta as comparações indicando as vantagens e desvantagens de cada filtro.

2. FILTRO DE KALMAN

O Filtro de Kalman foi desenvolvido por Rudolf Kalman com base em trabalhos anteriores de Thorvald Thiele e Peter Swerling e contribuições posteriores de Richard Bucy. A função do filtro é resolver problemas da estimação de estado em sistemas lineares ou linearizados (Filtro de Kalman Estendido). É a técnica de estimação de estado mais empregada, apesar de suas deficiências [6]. O modelo do Filtro de Kalman Discreto (FKD) é descrito de acordo com o as equações do sistema 1:

$$\begin{cases} s_t = A_t s_{t-1} + B_t u_t + \gamma_t \\ z_t = C_t s_t + \delta_t \end{cases} \quad (1)$$

onde $s \in R^n$ é o vetor de estados; $u \in R^l$ é o vetor das entradas de controle; $z \in R^m$ é o vetor de medições; a matriz $n \times n$, \mathbf{A} , é a matriz de transições de estados; \mathbf{B} , $n \times l$, é a matriz de coeficientes de entrada; a matriz \mathbf{C} , $m \times n$, é a matriz de observação; $\gamma \in R^n$ representa o vetor de ruídos do processo e $\delta \in R^m$ o vetor de erros de medição. Os índices t e $t-1$ representam os instantes de tempo atual e anterior respectivamente.

O funcionamento do filtro realiza a predição e atualização de acordo com as propriedades estatísticas do ruído. Existe um modelo interno do sistema que é utilizado para a atualização e uma realimentação realiza as medições de laser.

Os sistemas 2 e 3 descrevem as fases de predição e atualização respectivamente para o FKD [6].

$$\begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_{t-1} \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases} \quad (2)$$

$$\begin{cases} K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1} \\ \mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \end{cases} \quad (3)$$

O filtro de Kalman representa o vetor de estados s_t no tempo t por sua média μ_t e covariância Σ_t . As matrizes \mathbf{R} , $n \times n$, e \mathbf{Q} , $l \times l$, são as matrizes de covariância dos ruídos de processo (γ) e medição (δ) respectivamente e a matriz \mathbf{K} , $n \times m$ significa o ganho do sistema [6].

Para a realização do trabalho foi utilizado o Filtro de Kalman Estendido que lineariza as funções de acordo com a estimação corrente utilizando as derivadas parciais do processo e das funções de medição para calcular as estimações, mesmo quando não forem lineares [6].

O modelo do sistema para o FKE é representado de acordo com o sistema 4:

$$\begin{cases} s_t = g(u_{t-1}, s_{t-1}) + \gamma_t \\ z_t = h(s_t) + \delta_t \end{cases} \quad (4)$$

sendo $g(u_{t-1}, s_{t-1})$ uma função não-linear que representa o modelo do sistema e $h(s_t)$ é uma função não-linear que representa o modelo das medições.

As equações dos sistemas 5 e 6 representam as etapas de predição e atualização respectivamente [6].

$$\begin{cases} \bar{\mu}_t = g(u_{t-1}, \mu_{t-1}) \\ \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t \end{cases} \quad (5)$$

$$\begin{cases} K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \\ \mu_t = \bar{\mu}_t + K_t (z_t - h_t(\bar{\mu}_t)) \\ \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \end{cases} \quad (6)$$

onde a matriz \mathbf{G} , $n \times n$, é o jacobiano que lineariza o modelo e \mathbf{H} , $l \times n$ é o jacobiano que lineariza o vetor de medições feitas com o laser do robô. As equações 7 e 8 definem essas matrizes [6].

$$G_t = \frac{\partial g(u_{t-1}, s_{t-1})}{\partial s_{t-1}} \quad (7)$$

$$H_t = \frac{\partial h(s_t)}{\partial s_t} \quad (8)$$

A subseção a seguir apresenta a implementação do filtro para o problema abordado.

2.1 FKE Aplicado na Robótica Móvel

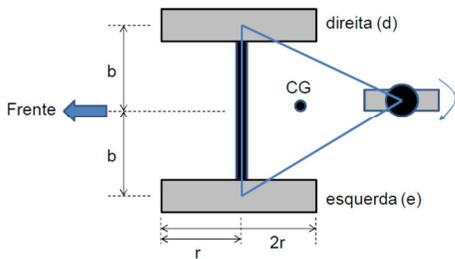


Figura 1. Representação do robô, onde CG é o centro de gravidade, (d) e (e) representam as rodas direitas e esquerdas do robô e $b+b = 2b$ significa o eixo das rodas [7].

A Figura 1 auxilia a compreensão do algoritmo do FKE que possui a seguinte sequência [9]:

1. Leitura da pose do robô.
2. Fase de predição.
3. Detecção de *features*.
4. Fase de atualização (cálculo do fator K_t).
5. Cálculo do valor de inovação.
6. Atualização da pose do robô.
7. Atualização da matriz covariância Σ_t da pose.
8. Robô dorme para repetir todas as etapas.

Para a primeira parte do algoritmo apenas são coletados os valores de x , y e θ da pose atual do robô. A fase de predição calcula a incerteza sobre a pose do robô (Σ_t)

$$\begin{cases} \Sigma_t = (G_t \times \Sigma_{t-1} \times G_t^T) + (V_t \times \Sigma_{\Delta t} \times V_t^T) + R_t \\ G_t = \begin{bmatrix} 1 & 0 & -\Delta S_t \sin(\theta_{t-1} + \frac{\Delta\theta_t}{2}) \\ 0 & 1 & \Delta S_t \cos(\theta_{t-1} + \frac{\Delta\theta_t}{2}) \\ 0 & 0 & 1 \end{bmatrix} \\ V_t = \begin{bmatrix} a & c \\ \frac{b}{2b} & -\frac{1}{2b} \end{bmatrix} \\ a = \frac{1}{2} \cos(\theta_{t-1} + \frac{\Delta\theta_t}{2}) - \frac{\Delta S_t}{2b} \sin(\theta_{t-1} + \frac{\Delta\theta_t}{2}) \\ b = \frac{1}{2} \sin(\theta_{t-1} + \frac{\Delta\theta_t}{2}) + \frac{\Delta S_t}{2b} \cos(\theta_{t-1} + \frac{\Delta\theta_t}{2}) \\ c = \frac{1}{2} \cos(\theta_{t-1} + \frac{\Delta\theta_t}{2}) + \frac{\Delta S_t}{2b} \sin(\theta_{t-1} + \frac{\Delta\theta_t}{2}) \\ d = \frac{1}{2} \sin(\theta_{t-1} + \frac{\Delta\theta_t}{2}) - \frac{\Delta S_t}{2b} \cos(\theta_{t-1} + \frac{\Delta\theta_t}{2}) \end{cases} \quad (9)$$

onde:

- θ_{t-1} é o teta da pose atual do robô.
- $\Delta\theta_t$ é $(Vel. roda direita \times D_t - Vel. roda esquerda \times D_t) / eixo$.
- ΔS_t é $(Vel. roda direita \times D_t + Vel. roda esquerda \times D_t) / 2$.
- D_t é o tempo que o loop dorme para o algoritmo repetir.
- Σ_{t-1} é matriz de covariância da pose inicial, que será atualizada posteriormente.

$$\begin{cases} \Sigma_{\Delta t} = \begin{bmatrix} K_s |\Delta S_t| & 0 \\ 0 & K_\theta |\Delta\theta_t| \end{bmatrix} \\ K_\theta = 0.1 \\ K_s = 0.1 \end{cases} \quad (10)$$

É preciso considerar os erros de leituras que o laser possui, eles ficam armazenados na matriz R_t .

$$R_t = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \quad (11)$$

σ_x , σ_y e σ_θ representam os erros de leitura em relação às coordenadas x , y e θ . Os valores utilizados no trabalho foram 5, 5 e 1 respectivamente.

A 3ª etapa consiste na detecção de *features* onde o robô realiza uma leitura de laser do ambiente e identifica quais os *landmarks* detectados e consequentemente as suas características (*features*) para poder se localizar no mapa.

A atualização da pose do robô ocorre na 4ª etapa através do cálculo do fator K_t .

$$\left\{ \begin{array}{l} K_t = \Sigma_t \times H_t^T \times (H_t \times \Sigma_t \times H_t^T + Q_t)^{-1} \\ H_t = \begin{bmatrix} -\frac{L_x - x_t}{\sqrt{q}} & -\frac{L_y - y_t}{\sqrt{q}} & 0 \\ \frac{L_y - y_t}{q} & -\frac{L_x - x_t}{q} & -1 \end{bmatrix} \\ q = (L_x - x_t)^2 + (L_y - y_t)^2 \\ Q_t = \begin{bmatrix} \sigma_{ld}^2 & 0 \\ 0 & \sigma_{l\theta}^2 \end{bmatrix} \end{array} \right. \quad (12)$$

onde:

- x_t é o valor de x da pose do robô.
- y_t é o valor de y da pose do robô.
- H_t é a matriz do jacobiano da função h .
- Q_t é a matriz composta pelo desvio padrão do sensor em relação a distância e angulação, os valores utilizados foram 0,5 e 0,1.
- L_x e L_y representam as coordenadas de x e y dos *landmarks* detectados.

As matrizes H_t Q_t aumentam de tamanho em relação a sua diagonal principal de acordo com a quantidade de *features* detectadas [9].

A 5ª etapa consiste no cálculo da inovação feito através das seguintes equações [9]:

$$\left\{ \begin{array}{l} INOVA = [zsensor_t - zreal_t] \\ zsensor_t = \begin{bmatrix} zrange \\ zbearing \end{bmatrix} \\ zrange = \sqrt{((L_x - xR)^2 + (L_y - yR)^2)} \\ zbearing = \arctan2((L_y - yR), (L_x - xR)) - \theta R \\ zreal_t = \begin{bmatrix} lrange \\ lbearing \end{bmatrix} \\ lrange = \sqrt{((L_x - xR)^2 + (L_y - yR)^2)} \\ lbearing = \arctan2((L_y - yR), (L_x - xR)) - \theta R \end{array} \right. \quad (13)$$

onde:

- $zsensor_t$ está relacionado a informação fornecida pela função e detecção de *features*.
- $zreal_t$ está relacionada ao *landmark* verdadeiro fornecido pela função de detecção de *features*.
- $zrange$ e $zbearing$ contém a informação relativas a x e y dos *landmarks* que o laser acha que detectou.
- $lrange$ e $lbearing$ contém a informação relativas a x e y dos *landmarks* reais detectados pelo laser.
- xR , yR e θR correspondem ao x , y e θ da pose do robô.

Na 6ª etapa a pose do robô é atualizada e a 7ª etapa atualiza a matriz de covariância da pose do robô de acordo com 14.

$$\left\{ \begin{array}{l} poseRobo = X_t + K_t \times INOVA \\ \Sigma_t = (I - K_t H_t) \Sigma_t \\ I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \right. \quad (14)$$

A 8ª e última etapa apenas faz o robô dormir por 2 segundos para que o algoritmo seja repetido. Para fazer a comparação entre desempenho, eficiência e custo computacional com o FKE, o algoritmo do Filtro de Partículas foi implementado.

3. FILTRO DE PARTÍCULAS

O Filtro de Partículas (FP) pertence à classe dos métodos estocásticos designados como “Monte Carlo”. São métodos que utilizam a amostragem para estabelecer o valor de uma grandeza que não é disponível através de uma expressão matemática para expressá-la [8].

O FP não utiliza linearizações nem requer que as incertezas sejam modeladas por distribuições normais. Por exemplo, seja f uma função complexa e seja necessário calcular a sua integral no intervalo $[a, b]$ [8].

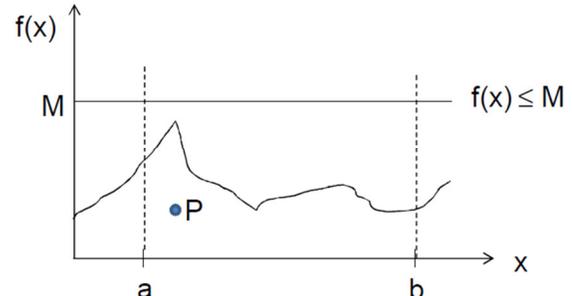


Figura 2. São gerados K pontos $P = (x, w)$ tal que x possui distribuição uniforme entre $[a, b]$ e w possui distribuição uniforme entre $[0, M]$ (as amostras). [7]

A partir da Figura 2 verificamos quantos pontos satisfazem a condição $f(x) \geq w$ (L pontos) [8].

$$\int_a^b f(x) dx \approx \frac{L}{K} (b - a) M \quad (15)$$

Em um FP as partículas operam em um ambiente simulado de maneira a obter medidas no âmbito contínuo, onde cada partícula é uma hipótese, nesse caso, a pose do robô. O conjunto de muitas partículas forma um sistema multimodal

de modo a preencher o espaço de uma maneira uniforme grande parte do espaço de busca [8].

Quanto maior a quantidade de partículas, maior a probabilidade de encontrar a solução para o problema proposto (pose correta do robô). Em robótica o FP é comumente utilizado para a localização global de um robô $[x, y, \theta]$ [8].

A aplicação do filtro de partículas para a localização do robô no mapa foi realizada de acordo com a descrição da seção seguinte.

3.1 FP Aplicado na Robótica Móvel

Como o robô não sabe onde está localizado, supõe-se que se encontra em qualquer lugar do mapa, o que gera uma distribuição uniforme onde todas as posições do mapa possuem a mesma probabilidade de serem a posição verdadeira.

Nessa abordagem o mapa não é discretizado, é contínuo, logo o FP espalha de maneira uniforme n partículas ao longo do mapa. Cada partícula é uma simulação de um robô munido dos mesmos sensores o que robô real, e sua distribuição será uniforme em x , y e θ (a orientação de cada partícula também varia) [8].

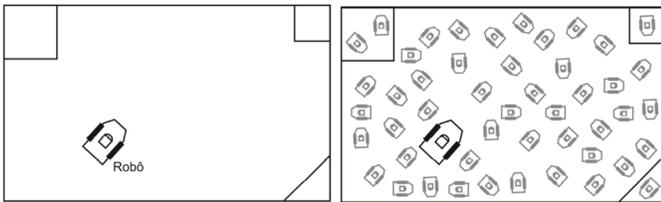


Figura 3. Criação das partículas no mapa de maneira uniforme ao longo do mapa com x , y e θ aleatórios.

Primeiramente foram simuladas 3000 partículas ao longo do mapa. Em seguida o robô e as partículas realizam uma leitura de laser, onde a leitura do robô utiliza o seu laser e a leitura de laser das partículas é simulada utilizando equações matemáticas [8][10].

As medidas do robô real são comparadas as medidas dos sensores das partículas e guardam-se as partículas cujas medidas são similares às do robô real, as demais são descartadas. Em seguida são geradas novas 300 partículas utilizando as melhores (as que estão mais próximas do robô) como referência [8][10].

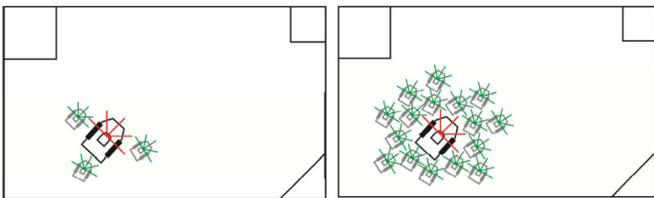


Figura 4. As partículas localizadas mais próximas do robô são selecionadas e as demais são descartadas.

Movimenta-se o robô e todas as partículas pelo ambiente de maneira igual e repetem-se os passos anteriores de obtenção e comparação de medidas e reprodução das melhores partículas. O ambiente foi implementado como circular, ou seja, durante

a movimentação caso a partícula sai por um lado, ela surge do outro.

Os passos descritos são repetidos recursivamente fazendo as partículas tenderem a ficar na mesma orientação e a média das poses de todas elas fornecerá uma boa aproximação da pose real do robô real.

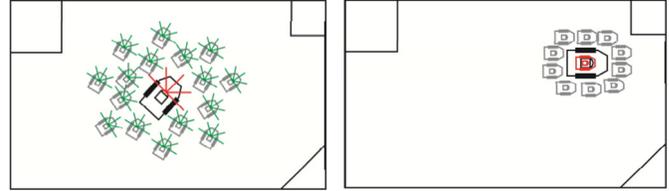
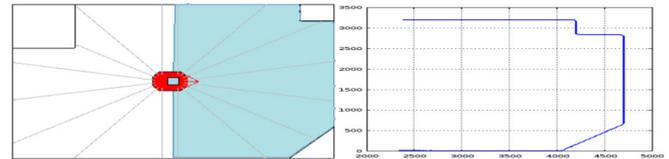


Figura 5. As melhores partículas são movimentadas junto com o robô. Após diversas iterações tendem a se posicionar na mesma orientação do robô.

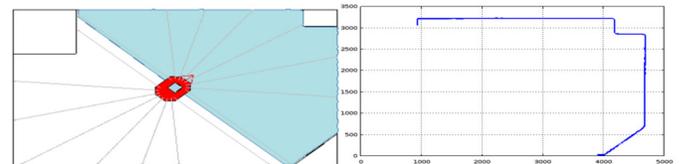
Após o término do algoritmo do filtro de partículas o robô real realiza uma ultima leitura de laser para indicar a posição real e compara-la com a posição encontrada pelas partículas. A seção 4 apresenta os resultados obtidos através do filtro de partículas e o filtro de Kalman.

4. RESULTADOS DAS SIMULAÇÕES

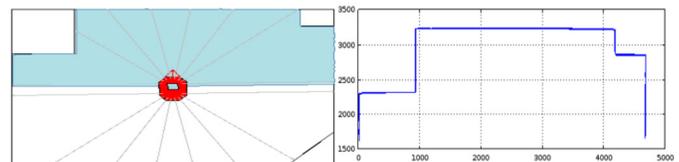
Para verificar o funcionamento do filtro de Kalman o robô parado realizou leituras variando seu ângulo θ em 0° , 45° , 90° , 135° , 180° , 225° , 270° e 315° .



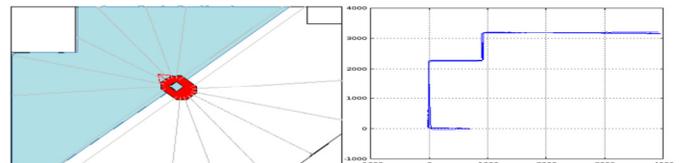
(a) Leitura em 0°



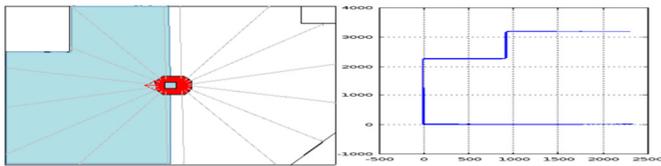
(b) Leitura em 45°



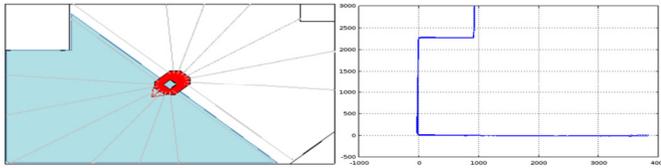
(c) Leitura em 90°



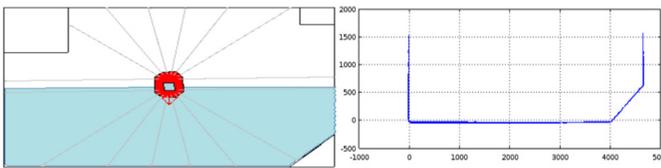
(d) Leitura em 135°



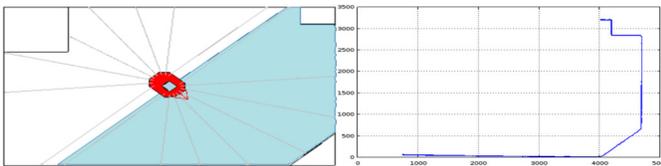
(e) Leitura em 180°



(f) Leitura em 225°



(g) Leitura em 270°



(h) Leitura em 315°

Figura 6. O Filtro de Kalman converge as paredes detectadas através da leitura de laser para uma linha apenas, significa que a pose do robô está sendo atualizada corretamente e que sua localização é precisa.

Para verificar o funcionamento do FK durante a movimentação, o robô realiza um passeio aleatório pelo mapa e mostra através do gráfico as paredes detectadas de acordo com a sua posição.

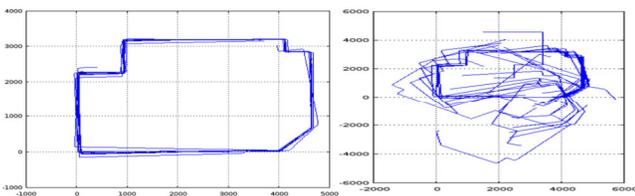


Figura 7. As paredes detectadas convertem corretamente para a posição real de acordo com o mapa. Quando o mapa é desenhado totalmente rotacionado em relação a sua posição original, significa que o robô está perdido no ambiente.

O Filtro de Partículas apresentou os seguintes resultados:

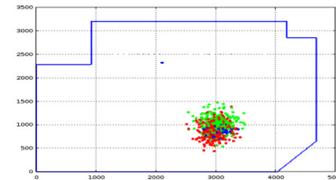
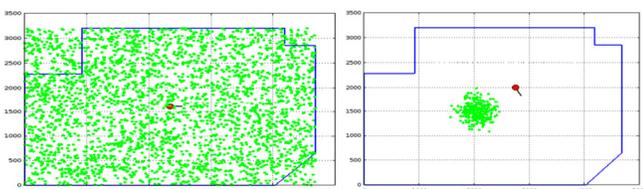


Figura 8. As partículas verdes primeiramente são igual a 3000, depois da sua reamostragem passam a ser 300. As partículas em azul representam as melhores e as vermelhas são as novas partículas geradas a partir de uma média obtida entre as partículas azuis.

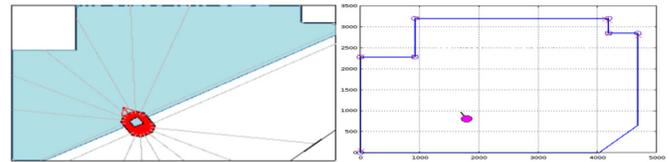


Figura 9. O critério de parada é atingido quando ao fazer uma leitura de laser das partículas e compara-las com a leitura do robô real, verifica-se que pelo menos dois *landmarks* iguais foram detectados em ambas leituras. Após atingir o critério de parada o robô real e as partículas fazem uma leitura de laser para mostrar os *landmarks* detectados de acordo com a posição em que se encontram.

5. CONCLUSÃO

O experimento realizado permite concluir que ambos os filtros possuem suas vantagens e desvantagens quando utilizados para fazer a localização em robótica móvel.

O Filtro de Kalman converge mesmo com um estado inicial muito impreciso, fornece uma estimativa da incerteza do estado em cada estágio, permite a incorporação de toda informação (sensores) disponível para uma melhor precisão e sua implementação é simples utilizando álgebra de matrizes.

Porém a otimalidade é garantida apenas para sistemas lineares com ruído gaussiano, pode sofrer divergência por instabilidade numérica ou por modelo incorreto, requer calibração e não resolve o problema do robô sequestrado.

O Filtro de Partículas permite representar densidades arbitrárias, pode ser utilizado em problemas sem ruídos gaussianos, possui foco adaptativo em regiões prováveis no espaço de estados e trata o problema do sequestro do robô.

Suas desvantagens são o alto custo computacional, dificuldade de determinar o número ótimo de partículas e o número de partículas cresce conforme o aumento da dimensão do modelo.

6. REFERÊNCIAS

- [1] GUDWIN, R. R. Notas de aulas da disciplina de Sistemas de Cognição Artificial – Aula 09: Robótica. Campinas, 2014. Disponível em <http://www.dca.fee.unicamp.br/~gudwin/ftp/ia889/Aula09.pdf> Acesso em 15 de outubro de 2014.
- [2] SWERLIN, P., “First Order Error Propagation in a Staggered Smoothing Procedure for Satellite Observation,” *Journal of the Astronautical Sciences*, Vol. 6, No. 3, Outono 1959.
- [3] KALMAN, R. E., “A New Approach to Line Filtering Prediction Problems,” *Transactions of the ASME, Journal of Basic Engineering*, Series 83D, Março 1960, pp. 35-45.

- [4] KALMAN, R. E., and R. S., Bucy., “New Results in Linear Filtering and Prediction Theory,” *Transactions of the ASME, Journal of Basic Engineering*, Series 83D, Março 1961, pp. 95-108.
- [5] HAMMERSLEY, J. M. and Morton, “Poor Man’s Monte Carlo”, *Journal of the Royal Statistical Society*, Series B (*Statistical Methodology*), 1954, pp. 23-38.
- [6] SANTANA, A, M. e MEDEIROS, A, A, D, “Localização de um Robô Móvel Baseado em Odometria e Marcos Naturais Utilizando Filtro de Kalman Extendido”. Disponível em <http://www.dca.ufrn.br/~adelardo/artigos/SBAI07b.pdf>. Acesso em 10 de dezembro de 2014.
- [7] CARDOZO, E. PINHEIRO, P. OLIVI. L. Notas de aulas da disciplina de Métodos Estocásticos em Robótica Móvel. Campinas, 2014. Disponível em <http://143.106.148.168:9080/Cursos/IA368W/parte1.pdf> Acesso em 17 de março de 2014.
- [8] CARDOZO, E. PINHEIRO, P. OLIVI. L. Notas de aulas da disciplina de Métodos Estocásticos em Robótica Móvel. Campinas, 2014. Disponível em <http://143.106.148.168:9080/Cursos/IA368W/parte2.pdf> Acesso em 30 de março de 2014.
- [9] CARDOZO, E. PINHEIRO, P. Notas de aulas da disciplina de Métodos Estocásticos em Robótica Móvel. Campinas, 2014. Disponível em <http://143.106.148.168:9080/Cursos/IA368W/KalmanFilter-TA.pdf> Acesso em 17 de abril de 2014.
- [10] CARDOZO, E. PINHEIRO, P. OLIVI. Notas de aulas da disciplina de Métodos Estocásticos em Robótica Móvel. Campinas, 2014. Disponível em <http://143.106.148.168:9080/Cursos/IA368W/PartFilter-TA.pdf> Acesso em 2 de junho de 2014.