

# Tutorial Sobre o Filtro de Partículas Aplicado em Localização de Robôs Móveis

Victor Romero, Anna Reali, Fabio Cozman, Leliane Nunes, Frank Dellaert.

Escola Politecnica da Universidad de São Paulo  
Departamento de Engenharia de Computação e Sistemas Digitais  
[victor.cano@poli.usp.br](mailto:victor.cano@poli.usp.br), [anna.reali@poli.usp.br](mailto:anna.reali@poli.usp.br), [fgcozman@usp.br](mailto:fgcozman@usp.br), [leliane@ime.usp.br](mailto:leliane@ime.usp.br),  
[frank@cc.gatech.edu](mailto:frank@cc.gatech.edu)

**Resume.** Este artigo descreve a aplicação do filtro de partículas na localização cooperativa de dois robôs móveis diferenciais e a derivação das equações cinemáticas que descrevem o sistema. Na primeira parte define-se o raciocínio bayesiano no qual se fundamentam os métodos de Monte Carlo, depois se apresenta cada fase do processo de estimação (predição e atualização) pro caso específico de um sensor laser para a coleta de observações. Este tutorial baseia-se no trabalho feito por Ioannis M. Rekleitis [2] e [3].

**Keywords:** Sistemas multi-robôs, Localização cooperativa, Filtro de partículas, Estimação Bayesiana.

## 1 Introdução:

Na localização, de robôs móveis se faz a estimação da pose do robô no tempo baseando-se em medidas de sensores ruidosos. Frente a esta problemática, tem se demonstrado [1] que é possível abordar o problema de localização numa forma muito mais eficiente se no lugar de um, utiliza-se vários robôs na realização das tarefas. Algumas vantagens são: é mais barato fazer um time de robôs simples que um só robô complexo; a capacidade de trocar informação entre robôs diminui a incerteza no processo de estimação.

O desafio por tanto consiste em encontrar estratégias para misturar eficientemente a informação coletada pelos sensores dos diferentes robôs. A pergunta que surge é: Como abordar a problemática da localização global numa forma cooperativa entre elementos autônomos distribuídos num mesmo espaço? Neste trabalho apresenta-se uma descrição geral do filtro de partículas aplicado na solução deste problema pro caso específico da estimação da pose de dois robôs móveis. Na abordagem proposta por [2], um dos robôs mantém estacionário em tanto que o outro se encontra em movimento. O robô estacionário atua como um alvo artificial a fim de que o robô em movimento possa medir sua pose respeito a ele. Estes papéis trocam-se ao longo da exploração. Na seguinte seção apresenta-se uma explicação breve sobre a estimação bayesiana na área de robótica móvel, no capítulo 3, descreve-se o filtro de partículas e suas etapas.

## 2 Estimação Bayesiana:

Esta abordagem permite a estimação do estado de um sistema na forma duma função de distribuição de probabilidade. Ela propõe um processo de estimação sequencial em duas fases, uma primeira na qual se gera uma predição da distribuição da pose (1) de cada robô no momento  $k$  dadas as ações até o momento  $k$  e as observações até o momento  $k-1$ . Dita distribuição calcula-se recursivamente a partir do modelo de movimento do robô e a distribuição a posteriori da pose calculada no momento prévio.

$$P(x_k | x_0, a^k, z^{k-1}) = \int P(x_k | a_k, x_{k-1}) P(x_{k-1} | x_0, a^{k-1}, z^{k-1}) d_{x_{k-1}} \quad (1)$$

Onde  $P(x_k | x_0, a^k, z^{k-1})$  é a predição,  $P(x_k | a_k, x_{k-1})$  é o modelo probabilístico de movimento de cada robô e  $P(x_{k-1} | x_0, a^{k-1}, z^{k-1})$  é a estimação a posteriori, calculada no momento  $k-1$ .

Quando nova informação sensorial (observações) encontra-se disponível, usa-se a regra de Bayes a fim de atualizar a distribuição do robô com as últimas observações (2).

$$P(x_k | x_0, a^k, z^k) = \frac{P(z_k | x_k) P(x_k | x_0, a^{k-1}, z^{k-1})}{P(z_k | x_0, a^{k-1}, z^{k-1})} \quad (2)$$

Onde  $P(z_k | x_k)$  é o modelo do sensor. Finalmente o denominador normalizador pode ser obtido por meio da equação (3).

$$P(z_k | x_0, a^{k-1}, z^{k-1}) = \int P(z_k | x_k) P(x_k | x_0, a^{k-1}, z^{k-1}) d_{x_k} \quad (3)$$

## 3 O Filtro de Partículas:

O filtro de partículas é uma implementação não paramétrica do filtro de Bayes. O seu objetivo principal é seguir uma variável de interesse em quanto evolui no tempo, tipicamente com uma *pdf* não-Gaussiana e potencialmente multimodal. A base do método é construir uma representação baseada em amostras de toda a *pdf*. Uma série de ações são feitas, cada uma modificando o estado da variável de interesse de acordo a um modelo. Além disso, em certos momentos uma observação que chega restringe o estado da variável de interesse, nesse momento. O filtro de partículas é recursivo por natureza e opera em duas fases: predição e atualização. Depois de cada ação, cada partícula é modificada de acordo com o modelo existente (etapa de predição), incluindo a adição de ruído aleatório a fim de simular o efeito do ruído sobre a variável de interesse. A variável de interesse (no nosso caso a pose do robô  $X^k = [x^k, y^k, \hat{\theta}^k]^T$ ) no tempo  $t=k$  é representada como um conjunto de  $M$  amostras (as “partículas”) ( $S_i^k = [X_j^k, w_j^k]: j=1 \dots M$ ), onde o índice  $j$  denota a partícula e não o robô, cada partícula está constituída de uma cópia da variável de interesse e um peso que define a contribuição desta partícula para a estimação global da variável. Uma

estimação da variável de interesse é obtida pela soma ponderada de todas as partículas.

### 3.1 Primeira Etapa: Predição.

A fim de prever a distribuição de probabilidade da pose do robô em movimento depois de se deslocar, nós precisamos ter um modelo do efeito do ruído na pose resultante. Qualquer movimento arbitrário  $[\Delta x, \Delta y]^T$  pode ser realizado como uma rotação seguida por uma translação. A pose inicial do robô é  $[x, y, \hat{\theta}]^T$ . Primeiro o robô vira  $\delta \hat{\theta} = \hat{\theta}_k - \hat{\theta}$ , onde  $\hat{\theta}_k = \arctan(\Delta y / \Delta x)$  para ficar na frente da sua posição objetivo, e depois se desloca para frente uma distancia  $\rho = \sqrt{\Delta x^2 + \Delta y^2}$ . Se a pose inicial é  $[x, y, \hat{\theta}]^T$ , a pose resultante  $[x', y', \hat{\theta}_k]^T$  é dada na equação 4. Consequentemente, o modelo de ruído é aplicado separadamente a cada um dos dois tipos de movimento, porque eles são assumidos independentes.

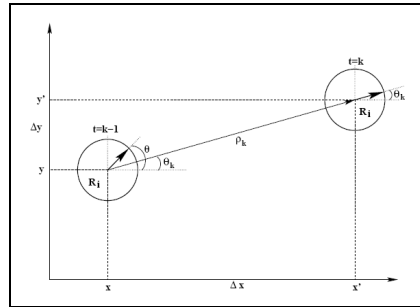


Figura 1. Movimento arbitrário  $[\Delta x, \Delta y]^T$  do robô Ri.

$$\begin{bmatrix} x' \\ y' \\ \hat{\theta}' \end{bmatrix} = \begin{bmatrix} x + \rho \cos(\hat{\theta}_k) \\ y + \rho \sin(\hat{\theta}_k) \\ \hat{\theta}_k \end{bmatrix} \quad (4)$$

#### 3.2.1 Rotação:

Quando o robô realiza uma rotação relativa  $\delta \hat{\theta}$  o ruído do erro odométrico é modelado como ruído gaussiano com média ( $M_{rot}$ ) experimentalmente estabelecida em [2] para robôs diferenciais, e sigma proporcional a  $\delta \hat{\theta}$ . Mais formalmente, se no tempo  $t=k$  o robô tem uma orientação  $\hat{\theta}$  então depois da rotação (tempo  $t=k+1$ ) a orientação do robô é dada pela equação 5. Portanto, para modelar a rotação de  $\delta \hat{\theta}$ , a orientação  $\hat{\theta}_j$  de cada partícula j é atualizada somando  $\delta \hat{\theta}$  mais um número aleatório obtido duma distribuição normal com média  $M_{rot}$  e desvio padrão  $\sigma_{rot} \delta \hat{\theta}$  ( $N(M_{rot}, \sigma_{rot} \delta \hat{\theta})$ ), onde  $\sigma_{rot}$  é em graus).

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \delta \hat{\theta} + N(M_{rot}, \sigma_{rot} \delta \hat{\theta}) \quad (5)$$

### 3.2.2 Translação:

O modelado da translação para frente é mais complicada. Há duas diferentes fontes de erro, a primeira relacionada com a distância realmente percorrida e a segunda relacionada com os câmbios na orientação durante a translação para frente. Durante a translação a orientação do robô cambia constantemente resultando num desvio da direção de translação desejada; tal efeito é chamado “drift” e o modelamos adicionando uma pequena quantidade de ruído à orientação do robô antes e depois de cada passo. Também, se a distancia planejada é  $\rho$ , a distancia real percorrida é dada por  $\rho$  mais algum ruído seguindo uma distribuição gaussiana. Resultados experimentais proporcionam o valor esperado e o desvio padrão para o “drift” e a translação pura. Já que é muito difícil modelar analiticamente o processo contínuo, uma simulação é usada para discretizar o movimento em  $K$  passos, onde  $K$  é escolhido o suficientemente baixo por eficiência computacional, mas o suficientemente alto a fim de descrever o efeito do ruído na translação para frente. Se  $[\sigma_{translação}, \sigma_{drift}]$  são valores experimentalmente obtidos por meio da distancia percorrida então a cada passo da simulação o desvio padrão usado é dado na equação 6.

$$\begin{aligned} \sigma_{trs} &= \sigma_{translação} \sqrt{K} \\ \sigma_{drft} &= \sigma_{drift} \sqrt{\frac{K}{2}} \end{aligned} \quad (6)$$

A figura dois apresenta uma ilustração gráfica do efeito dos dois parâmetros de ruído ( $\sigma_{trs}, \sigma_{drft}$ ) no modelo preditivo. Nos dois casos o robô faz um só movimento para a frente de 100cm (sub-plot superior da esquerda), 200cm (sub-plot superior da direita), 300 cm (sub-plot inferior da esquerda), e 400cm (sub-plot inferior da direita). Na figura 2a a incerteza na distancia percorrida é a incerteza dominante e por isto as partículas se espalham muito mais na direção do movimento. Em contraste, na figura 2b, onde o ruído do drift domina, as partículas estão espalhadas em um padrão circular. O Apêndice C de [3] contem um estudo experimental detalhado destes parâmetros usando a plataforma móvel Superscout da companhia Nomadic Technologies.

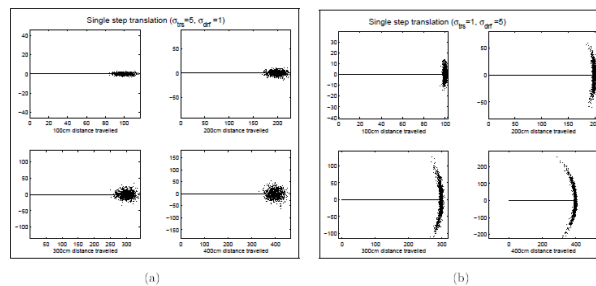


Figura 2. Efeito de  $\sigma_{trs}, \sigma_{drft}$  na translação para a frente: (a)  $\sigma_{trs} = \frac{5cm}{m}, \sigma_{drft} = 1^\circ/m$  (b)  $\sigma_{trs} = \frac{1cm}{m}, \sigma_{drft} = 5^\circ/m$  [2].

### 3.2.3 Resampling:

Um dos problemas que aparecem com o uso de filtros de partículas é a depleção da população após algumas iterações. A maioria das partículas tem se movimentado o suficientemente longe de tal forma que o seu peso torna-se demasiado pequeno como para contribuir à pdf do robô em movimento.

Liu et al. [4] refere-se a duas medidas diferentes que estimam o número de partículas com pesos perto a zero: um é o coeficiente de variação  $cv_t^2$  (ver equação 7) e a segunda é o tamanho efetivo de amostra ESS<sub>t</sub> (effective sample size) (ver equação 8).

$$cv_t^2 = \frac{\text{var}(w_t(i))}{E^2(w_t(i))} = \frac{1}{M} \sum_{i=1}^M (Mw_t(i) - 1)^2 \quad (7)$$

$$ESS_t = \frac{M}{1 + cv_t^2} \quad (8)$$

Quando o tamanho efetivo de amostra cai abaixo de certo umbral, usualmente uma porcentagem do número de partículas M, então a população de partículas é re-amostrada, eliminando (probabilisticamente) aquelas com pesos pequenos e duplicando aquelas com pesos maiores.

Diferentes métodos têm sido propostos para fazer *resampling*: três dos mais comuns são discutidos por [3]. Em cada caso a entrada é um vetor de pesos das partículas e a saída é um vetor de índices das partículas que vão ser propagadas. O requisito é que a pdf reconstruída pela população re-amostrada seja muito parecida à pdf antes do resampling. No método de *Escolha com substituição* (Select with replacement) cada partícula é escolhida para continuar com uma probabilidade igual a o seu peso. Em [2] usou-se a abordagem de Carpenter et al. [5].

A figura 3 apresenta dois exemplos de movimentos complexos e ilustra a performance da etapa de predição do filtro de partículas. Na figura 3a, o robô desloca-se para frente em três vezes, vira noventa graus, logo se desloca para frente três vezes mais, e depois vira 90 graus outra vez. Como pode ser visto, a incerteza cresce sem limite. A figura 3b apresenta a validação experimental do modelo de predição. Neste caso, o modelo preditivo foi guiado por um conjunto de comandos de movimento que foram usados em um experimento (para a completa descrição do experimento, olhar pro capítulo 8 de [2]). Em suma, o experimento consistiu de translações para frente, cada uma seguida de quatro rotações de noventa graus (a fim de sensar o ambiente em quatro direções diferentes). Os círculos ligados na figura 3b representam os valores não corrigidos dos odômetros. De fato, a trajetória real do robô foi mantida em uma linha reta, mas as estimações odométricas se desviaram devido ao ruído. O modelo preditivo foi construído usando os parâmetros estatísticos de ruído coletados experimentalmente (ver [3]). A nuvem de partículas predita pode ser vista ao redor dos valores registrados seguindo a trajetória com alta precisão.

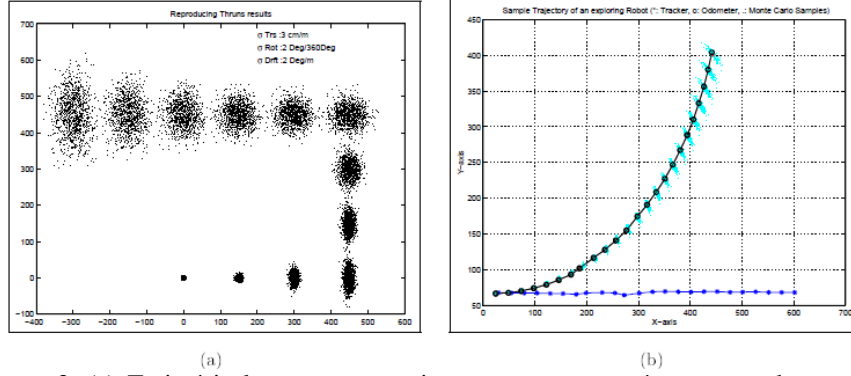


Figura 3. (a) Trajetória longa na que a incerteza crescente é representada por uma nuvem de partículas. (b) Serie de translações para frente e rotações de 360° realizadas por [2]. A linha que sobe representa os valores odométricos não corrigidos, os quais são seguidos muito bem pela nuvem de partículas; a linha de abaixo representa a trajetória real.

### 3.2.4 Segunda Etapa: Atualização.

Depois duma ação (o movimento de um dos robôs) um sensor é usado a fim de estimar a pose do robô em movimento. Os cálculos são dependentes da configuração do sensor utilizado. As seguintes duas subseções apresentam a atualização dos pesos das partículas do robô em movimento para a combinação de laser e alvo em dois casos diferentes. Primeiro deriva-se as equações de atualização quando o laser é montado no robô estacionário (subseção 3.2.4.1); segundo para quando o laser é montado sobre o robô em movimento e o alvo é montado no robô estacionário (subseção 3.2.4.2).

#### 3.2.4.1 Estimação da pose, Robô estacionário observando ao robô em movimento:

Se a posição do robô estacionário  $X_s = [x_s, y_s, \hat{\theta}_s]^T$  (com o laser) e a posição do robô em movimento  $X_m = [x_m, y_m, \hat{\theta}_m]^T$  (com o alvo) são conhecidas, então a medida do sensor  $z = [\rho, \hat{\theta}, \hat{\phi}]^T$  pode ser calculada pela equação 9:

$$z = \begin{bmatrix} \rho \\ \hat{\theta} \\ \hat{\phi} \end{bmatrix} = \begin{bmatrix} \sqrt{dx^2 + dy^2} \\ \arctan(dy/dx) - \hat{\theta}_s \\ \arctan(-dy/-dx) - \hat{\theta}_m \end{bmatrix} \quad (9)$$

Onde  $dx = x_m - x_s$  e  $dy = y_m - y_s$ .

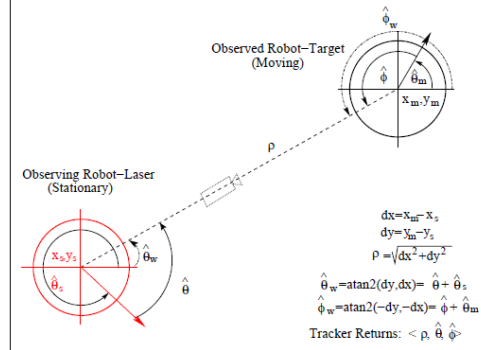


Figura 4. O robô estacionário com o laser observa o robô em movimento que leva o alvo [2].

Se a informação conhecida é a pose do robô estacionário ( $x_s$ ) (com o laser) e a medida do sensor é  $z = [\rho, \hat{\theta}, \hat{\phi}]^T$  então a *estimativa* da pose do robô em movimento (alvo) ( $X_{m_{est}}(k+1)$ ) é dada na equação 10:

$$x_{m_{est}}(k+1) = \begin{bmatrix} x_{m_{est}} \\ y_{m_{est}} \\ \hat{\theta}_{m_{est}} \end{bmatrix} = \begin{bmatrix} x_s + \rho \cos(\hat{\theta}_s + \hat{\theta}) \\ y_s + \rho \sin(\hat{\theta}_s + \hat{\theta}) \\ \pi + \hat{\theta} + \hat{\theta}_s - \hat{\phi} \end{bmatrix} \quad (10)$$

As equações 9 e 10 são equivalentes. Conseqüentemente, as equações acima podem ser usadas a fim de calcular o peso de cada partícula do robô em movimento, assumindo um modelo de erro gaussiano para cada componente dos dados do sensor ( $\rho, \hat{\theta}, \hat{\phi}$ ), em dois formas diferentes. Primeiro, seja  $X_{mi}^k = [x_{mi}, y_{mi}, \hat{\theta}_{mi}]^T$  a  $i$ -ésima partícula no tempo  $k$ . Logo, se a pose do robô estacionário é conhecida  $X_s = [x_s, y_s, \hat{\theta}_s]^T$  a medida estimada  $z_i$  para a partícula  $i$  é dada na equação 11:

$$z_i = \begin{bmatrix} \rho_i \\ \hat{\theta}_i \\ \hat{\phi}_i \end{bmatrix} = \begin{bmatrix} \sqrt{dx_i^2 + dy_i^2} \\ \arctan(dy_i/dx_i) - \hat{\theta}_s \\ \arctan(-dy_i/-dx_i) - \hat{\theta}_{m_i} \end{bmatrix} \quad (11)$$

Onde  $dx_i = x_{m_i} - x_s$  e  $dy_i = y_{m_i} - y_s$ .

O peso da partícula  $i$  então é proporcional à probabilidade de  $X_{m_i}^{k+1}$  dados  $X_s$  e  $z_i$  (ver equação 12). Como pode se ver na equação 11 que o valor de  $\hat{\phi}_i$  é afetado pela

pose completa da partícula  $i$  (ambos posição e orientação). Por tanto o erro na posição  $(x_{m_i}, y_{m_i})$  é usado duas vezes. Na equação 12 as constantes  $\sigma_\rho, \sigma_{\hat{\theta}}, \sigma_{\hat{\phi}}$  são os presuntos desvios padrões do ruído da medida do sensor e elas significam a confiança com a qual se aceita cada medida.

$$P(x_{m_i}^{k+1} | x_s, z) = \frac{1}{\sqrt{2\pi}\sigma_\rho} e^{-\frac{(\rho-\rho_i)^2}{2\sigma_\rho^2}} \frac{1}{\sqrt{2\pi}\sigma_{\hat{\theta}}} e^{-\frac{(\hat{\theta}-\hat{\theta}_i)^2}{2\sigma_{\hat{\theta}}^2}} \frac{1}{\sqrt{2\pi}\sigma_{\hat{\phi}}} e^{-\frac{(\hat{\phi}-\hat{\phi}_i)^2}{2\sigma_{\hat{\phi}}^2}} \quad (12)$$

Resultados experimentais tem mostrado que a precisão da posição do robô é (quase) fixa (independente da distancia à qual o robô é visto). Desafortunadamente a medida do sensor está em coordenadas polares e por isto para um erro fixo no ângulo ( $\hat{\theta}$ ) em quanto maior é a distância ( $\rho$ ) maior é o erro. Na pratica, é necessário calcular  $\sigma_{\hat{\theta}}$  como uma função de  $\rho$  :

$$\sigma_{\hat{\theta}} = h(\rho, \sigma_{\hat{\theta}}) = \arcsin(\sigma_{\hat{\theta}} / \rho) \quad (13)$$

Uma função de peso alternativa é usar a diferença em coordenadas cartesianas e a estimação da orientação a fim de dar peso à partícula  $X_{m_i}^{k+1}$  dados  $X_s$  e  $z_i$  (ver equação 14).

$$P(x_{m_i}^{k+1} | x_s, z) = \frac{1}{\sqrt{2\pi}\sigma_\rho} e^{-\frac{(dx-dx_i)^2}{2\sigma_\rho^2}} \frac{1}{\sqrt{2\pi}\sigma_\rho} e^{-\frac{(dy-dy_i)^2}{2\sigma_\rho^2}} \frac{1}{\sqrt{2\pi}\sigma_{\hat{\phi}}} e^{-\frac{(\hat{\phi}_m-\hat{\phi}_{m_i})^2}{2\sigma_{\hat{\phi}}^2}} \quad (14)$$

A segunda abordagem é usar a equação 10 e dar peso a cada partícula dependendo de que tão longe está da pose estimada do robô em movimento (ver equação 15). Onde se  $X_{m_{est}}(k+1) = [x_{m_{est}}, y_{m_{est}}, \hat{\theta}_{m_{est}}]$  é a estimação da pose e  $X_{m_i}^k = [x_{m_i}, y_{m_i}, \hat{\theta}_{m_i}]^T$  é a  $i$ -ésima partícula então  $d_i = \sqrt{(x_{m_{est}} - x_{m_i})^2 + (y_{m_{est}} - y_{m_i})^2}$ . A desvantagem de esta abordagem é que  $\sigma_d, \sigma_{\hat{\theta}}$  não representam o modelo do ruído do sensor.

$$P(x_{m_i}^{k+1} | x_s, z) = \frac{1}{\sqrt{2\pi}\sigma_d} e^{-\frac{(d_i)^2}{2\sigma_d^2}} \frac{1}{\sqrt{2\pi}\sigma_{\hat{\theta}}} e^{-\frac{(\hat{\theta}_{m_{est}}-\hat{\theta}_{m_i})^2}{2\sigma_{\hat{\theta}}^2}} \quad (15)$$

Durante a estimação do peso é usada a pose do robô estacionário. Já que a pose real não é conhecida, podem ser usadas diferentes estimativas  $\bar{X}_s$ . As seguintes opções têm sido consideradas:

- A melhor partícula (Aquela com peso máximo):  $\bar{X}_s = X_s^{\max}$
- Média ponderada:  $\bar{X}_s = \sum_{j=1}^M X_j^s w_j$
- Usar cada partícula do robô estacionário ( $O(n^2)$ ):

$$P(x_{m_i}^{k+1} | x_s, z) = \sum_{j=1}^n P(x_{m_i}^{k+1} | x_s^j, z)$$

- Média Robusta: Escolher só as partículas que estão a menos de  $\mathcal{E}$  da partícula com máximo peso. A vantagem deste método é que ele seleciona o modo da distribuição e reduz o erro de discretização (que acontece quando é usada uma partícula só).

$$\bar{x}_s = \sum_{j=1}^k x_s^j w_j : |x_s^j - x_s^{\max}| \leq \mathcal{E}$$

### 3.2.4.2 Estimação da pose, robô em movimento observando o robô estacionário:

Esta vez o robô estacionário tem o alvo. Se as poses dos dois robôs ( $X_s = [x_s, y_s, \hat{\theta}_s]^T$  e  $X_m = [x_m, y_m, \hat{\theta}_m]^T$ ) são conhecidas então a medida do sensor ( $z = [\rho, \hat{\theta}, \hat{\phi}]^T$ ) pode ser calculada pela equação 16.

$$\begin{bmatrix} \rho \\ \hat{\theta} \\ \hat{\phi} \end{bmatrix} = \begin{bmatrix} \sqrt{dx^2 + dy^2} \\ \arctan(dy/dx) - \hat{\theta}_m \\ \arctan(-dy/-dx) - \hat{\theta}_s \end{bmatrix} \quad (16)$$

Onde  $dx = x_s - x_m$  e  $dy = y_s - y_m$ .

Se a pose do robô estacionário ( $X_s$ ) (levando o alvo) e a observação ( $z = [\rho, \hat{\theta}, \hat{\phi}]^T$ ) são conhecidas então a estimativa da pose do robô em movimento (levando o laser) ( $X_m$ ) é dada na equação 17.

$$x_{m_{est}}(k+1) = \begin{bmatrix} x_{m_{est}} \\ y_{m_{est}} \\ \hat{\theta}_{m_{est}} \end{bmatrix} = \begin{bmatrix} x_s + \rho \cos(\hat{\phi} + \hat{\theta}_s) \\ y_s + \rho \sin(\hat{\phi} + \hat{\theta}_s) \\ \pi + \hat{\phi} + \hat{\theta}_s - \hat{\theta} \end{bmatrix} \quad (17)$$

Aplicando a mesma metodologia que na subseção anterior, as funções para a atualização dos pesos são idênticas com as das equações 12, 14, 15.

## Conclusão:

A figura 5 apresenta uma ilustração do processo acima descrito durante duas iterações. A primeira coluna apresenta a fase de *predição* e a segunda coluna a fase de *atualização*. O robô em movimento inicia na posição  $[0,0]$ , e o robô estacionário é localizado em  $[0,100]$ . Na figura 8a o robô em movimento desloca-se até  $[100\text{cm}, 100\text{cm}]$  e as partículas formam uma nuvem de aproximadamente 20cm de raio. A figura 8b apresenta a fase de atualização baseada na medida do sensor (a cor mais escura representa maiores pesos). No segundo passo o robô desloca-se até  $[100\text{cm}, -100\text{cm}]$  e a figura 8c apresenta a nuvem de partículas. É bom notar que as partículas com maiores pesos (cinza mais escuro) têm se espalhado. Finalmente, a figura 8d apresenta a segunda fase de atualização onde outra vez as partículas mais próximas à pose medida têm maiores pesos.

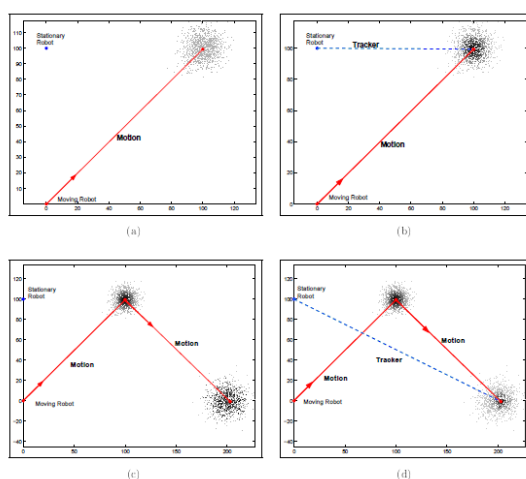


Figura 5. (a) Predição do primeiro passo. (b) Atualização usando a observação. (c) Predição do segundo passo. (d) Atualização usando a observação.

## References

1. Cooperative Positioning with Multiple Robots - Ryo Kurazume and Shigemi Nagata - FUJITSU LABORATORIES LTD. - Tokyo Institute of Technology – 1994.
2. Ioannis M. Rekleitis. Cooperative Localization and Multi-Robot Exploration. PhD thesis, School of computer Science, McGill University, Montreal, Quebec, Canada (2003).
3. Ioannis M. Rekleitis. A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Center for Intelligent Machines, McGill University, Montreal, Quebec, Canada (2004).
4. Jun S. Liu, Rong Chen, and Tanya Logvinenko.: A theoretical framework for sequential importance sampling and resampling. In A. Doucet, N. de Freitas, and N. J. Gordon, editors, Sequential Monte Carlo in Practice. Springer-Verlag (2001).
5. J. Carpenter, P. Clifford, and P. Fearnhead. An improved particle filter for non-linear problems. IEE proceedings – Radar, Sonar and Navigation (1999).