

Trabalho #2 – COE841

Professor: Ramon Romankevicius Costa

Alunos: André Abido Figueiró

Tiago Bornia de Castro

FILTRO BAYESIANO DISCRETO

```
1: Discrete_Bayes_filter ( $\{p_{k,t-1}\}, u_t, z_t$ ) :  
2:   for all  $k$  do  
3:      $\bar{p}_{k,t} = \sum_i p(X_t = x_k | u_t, X_{t-1} = x_i) p_{i,t-1}$   
4:      $p_{k,t} = \eta p(z_t, X_t = x_k) \bar{p}_{k,t}$   
5:   endfor  
6:   return  $\{p_{k,t}\}$ 
```

Figura 1 - Algoritmo do Filtro Bayesiano Discreto.

Filtros Bayesianos discretos se aplicam a problemas com espaços de estados finitos, onde a variável aleatória X_t pode assumir muitos valores. O espaço é dividido em vários ladrilhos e cada ladrilho possui uma probabilidade de localização do robô.

A Figura 1 apresenta o algoritmo do Filtro Bayesiano Discreto. Este código é derivado do filtro Bayesiano geral, substituindo a integração por uma soma finita. A crença no tempo t é uma atribuição de uma probabilidade a cada estado x_k , denotado $p_{k,t}$. Assim, a entrada para o algoritmo é uma distribuição de probabilidade discreta $\{p_{k,t}\}$, junto com o u_t de controle mais recente e a medição z_t .

A etapa de previsão está representada na linha 3, onde se calcula a crença para o novo estado com base no controle. A etapa de correção é apresentada na linha 4, onde a previsão é então atualizada, de modo a incorporar a medição.

A simulação foi feita com base no exemplo 6.14 do livro [1]. A Figura 2 apresenta o cenário de simulação. O robô pode estar localizado em uma das 5 posições. Um sensor de cor é utilizado para fazer a leitura da cor do ladrilho em que o robô está localizado.

Foram feitas 3 medições (Branco, Preto, Preto). E em cada rodada o robô se deslocava uma célula. Probabilidade de controle:

Andar 1 ladrilho = 0.8 (neste caso o robô avança para o próximo ladrilho). Andar menos que um ladrilho = 0.1 (neste caso o robô não avança para o próximo ladrilho). Andar mais que 1 ladrilho = 0.1 (neste caso o robô avança dois ladrilhos).

Probabilidades de medição:

$$P(Z = \text{preto} | X = \text{preto}) = 0.6$$

$$P(Z = \text{branco} | X = \text{preto}) = 0.4$$

$$P(Z = \text{branco} | X = \text{branco}) = 0.8$$

$$P(Z = \text{preto} | X = \text{branco}) = 0.2$$

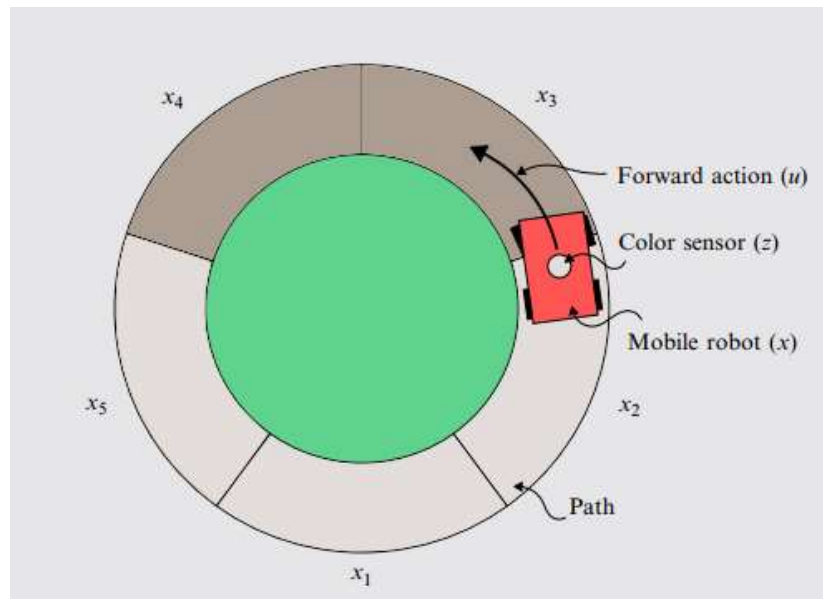


Figura 2 – Cenário de simulação.

Na primeira rodada a localização do robô era desconhecida, a probabilidade era uma distribuição discreta uniforme. Em seguida foi feita a primeira medição e a cor lida foi branca. A Figura 3 apresenta a previsão e a correção da primeira rodada.

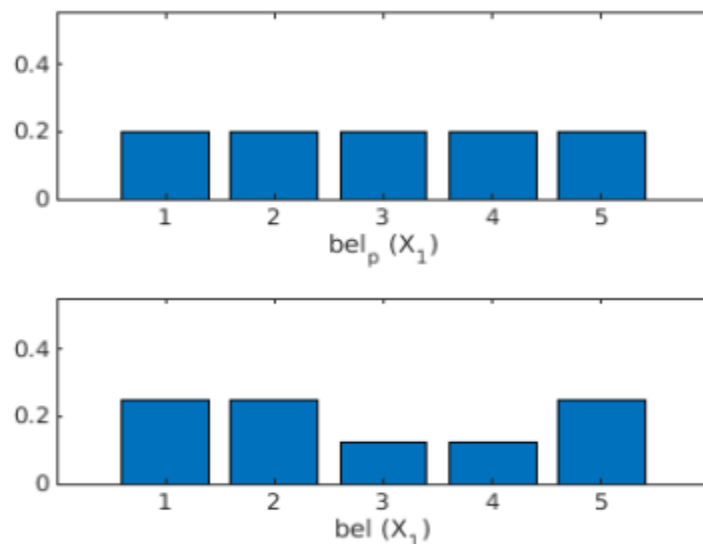


Figura 3 – Previsão e Correção – 1ª rodada.

A segunda rodada recebe como entrada a correção da primeira rodada. A segunda leitura do sensor de cor foi preta. A Figura 4 apresenta a previsão e a correção da segunda rodada.

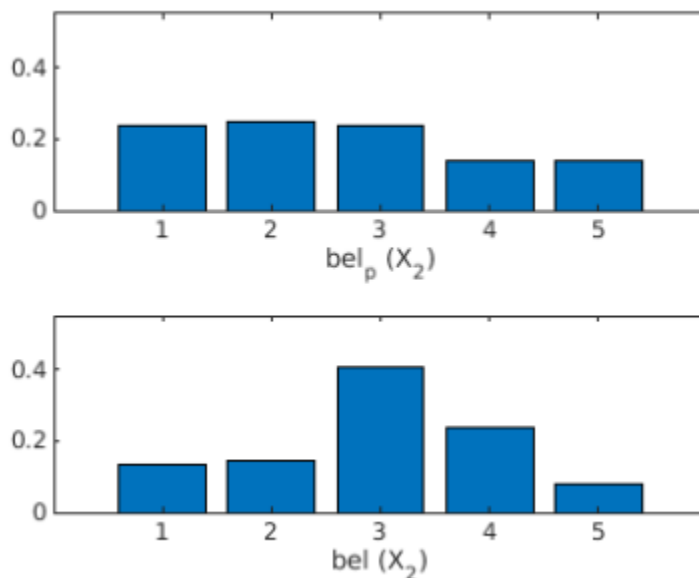


Figura 4 - Previsão e Correção – 2ª rodada.

Na terceira rodada a medida do sensor foi preta. A Figura 5 apresenta a previsão e a correção da segunda rodada.

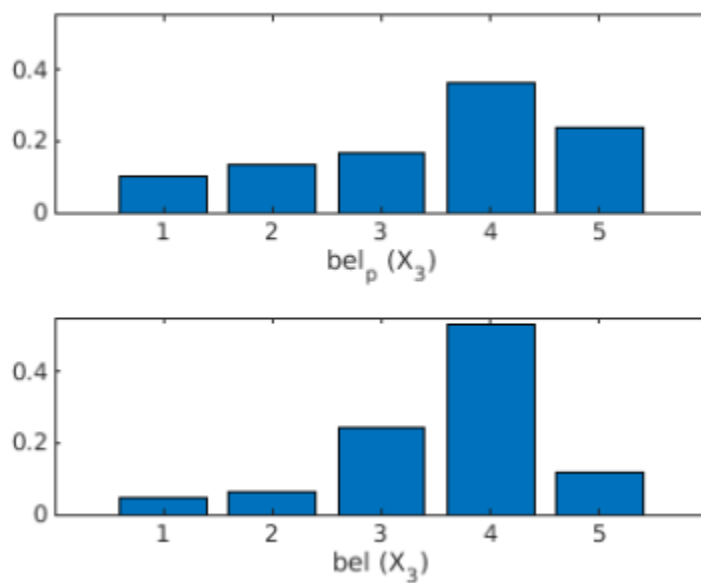


Figura 5 - Previsão e Correção – 3ª rodada.

FILTRO BAYESIANO BINÁRIO

```
1: Algorithm binary_Bayes_filter( $l_{t-1}, z_t$ ):  
2:    $l_t = l_{t-1} + \log \frac{p(x|z_t)}{1-p(x|z_t)} - \log \frac{p(x)}{1-p(x)}$   
3:   return  $l_t$ 
```

Figura 6 - Algoritmo do Filtro Bayesiano Binário.

O filtro bayesiano binário considera um evento probabilístico com estados que possam assumir apenas dois valores possíveis, sendo tal estado constante durante o funcionamento do algoritmo.

O algoritmo proposto busca obter uma estimativa (*belief*) para o estado, considerando valores de $-\infty$ a $+\infty$ para expressar o grau de certeza da estimativa. Seu algoritmo é apresentado na Figura 6.

Para o exemplo foi apresentado o caso de um teste de gravidez que pode apresentar falsos-positivos ou falsos-negativos. O teste é realizado várias vezes sem que haja mudança no estado (gravidez), sendo o resultado deste utilizado como observação para o filtro. A Tabela 1 apresenta as probabilidades associadas ao teste, ao passo que as Figuras 7 e 8 apresentam exemplos de resultados do filtro considerando casos com a mulher grávida e não grávida. Nesses casos, o resultado do teste é gerado aleatoriamente conforme a probabilidade apresentada na tabela correspondente.

Vale notar que, uma vez que as probabilidades de um teste ser um falso negativo são relativamente altas, resultados negativos não apresentam influência tão grande sobre as estimativas quanto os resultados positivos.

Tabela 1: Probabilidades associadas ao teste de gravidez.

X \ Z	Positivo	Negativo
Grávida	8%	8%
Não Grávida	2%	82%

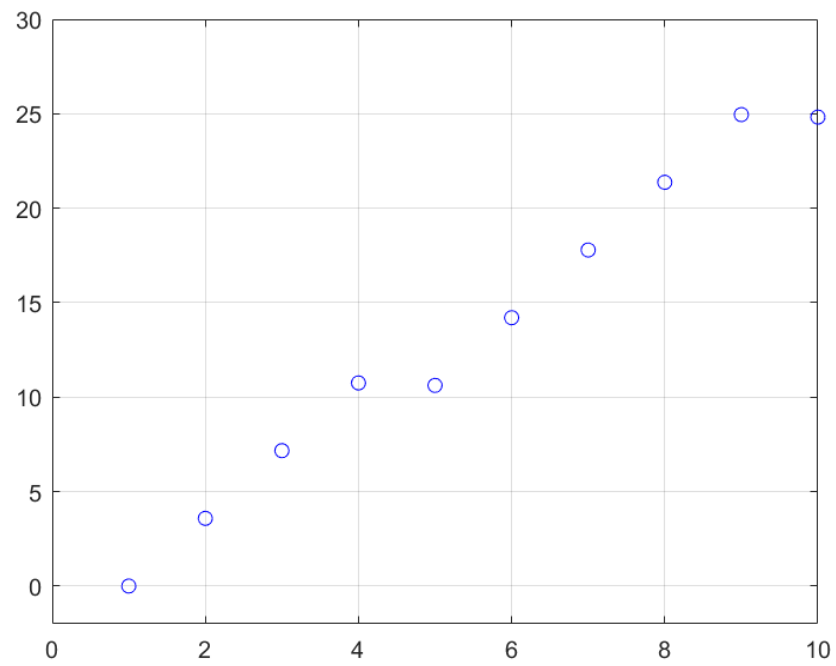


Figura 7 – Cenário com Mulher grávida.

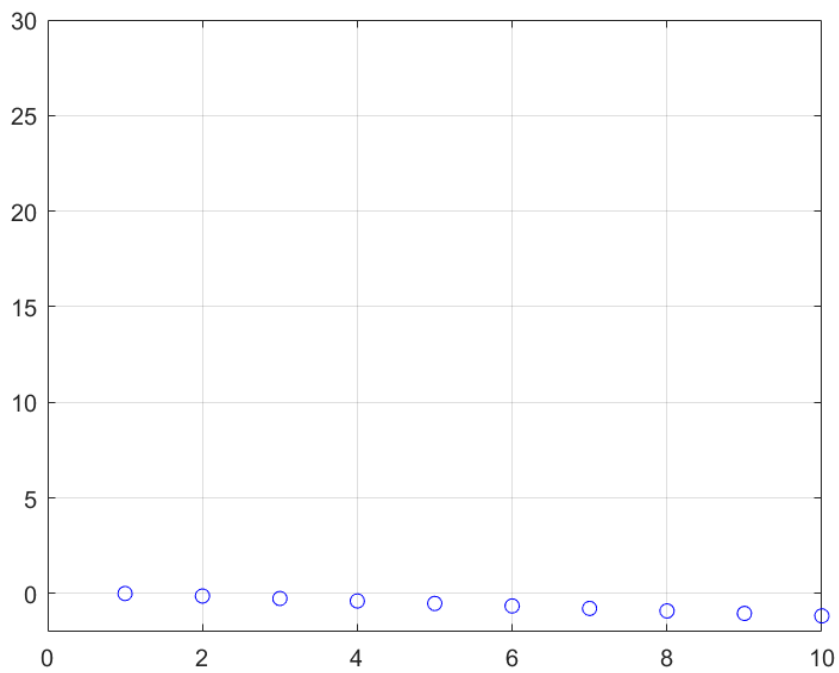


Figura 8 – Cenário com Mulher Não grávida.

FILTRO DE PARTÍCULAS

```
1: Particle_filter ( $\mathcal{X}_{t-1}, u_t, z_t$ ) :  
2:    $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$   
3:   for  $m = 1$  to  $M$  do  
4:     sample  $x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})$   
5:      $w_t^{[m]} = p(z_t | x_t^{[m]})$   
6:      $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$   
7:   endfor  
8:   for  $m = 1$  to  $M$  do  
9:     draw  $i$  with probability  $\propto w_t^{[i]}$   
10:    add  $x_t^{[i]}$  to  $\mathcal{X}_t$   
11:  endfor  
12:  return  $\mathcal{X}_t$ 
```

Figura 9 - Algoritmo do Filtro de Partículas.

A ideia-chave do filtro de partículas é representar o bel (xt) posterior por um conjunto de amostras de estados aleatórios extraídas desse posterior. O algoritmo de filtro de partículas constrói a crença bel (xt) recursivamente a partir da crença bel (xt - 1) um passo de tempo antes. Como as crenças são representadas por conjuntos de partículas, isso significa que os filtros de partículas constroem o conjunto de partículas X_t recursivamente a partir do conjunto X_{t-1} .

A entrada desse algoritmo é o conjunto de partículas X_{t-1} , junto com o controle mais recente u_t e a medição mais recente z_t . O algoritmo então constrói primeiro um conjunto de partículas temporárias \bar{X} que representa a crença bel (xt). Ele faz isso processando sistematicamente cada partícula do conjunto de partículas de entrada X_{t-1} . Posteriormente, ele transforma essas partículas no conjunto X_t , que aproxima a distribuição posterior bel (xt).

A linha 4 gera uma predição para as partículas no tempo t com base no estado anterior e no controle u_t . A linha 5 calcula para cada partícula o fator de importância (peso de cada partícula). Fatores de importância são usados para incorporar a medição z_t no conjunto de partículas.

As linhas de 8 a 11 implementaram o que é conhecido como reamostragem ou amostragem de importância. O algoritmo extrai com substituição M partículas do conjunto temporário \bar{X}_t . A probabilidade de desenhar cada partícula é dada por seu peso de importância. A reamostragem transforma um conjunto de partículas M em outro conjunto de partículas do mesmo tamanho. A etapa de reamostragem é uma implementação

probabilística da ideia darwiniana de sobrevivência do mais apto: ela reorienta o conjunto de partículas para regiões no espaço de estados com alta probabilidade posterior. Ao fazer isso, ele concentra os recursos computacionais do algoritmo de filtro em regiões no espaço de estados onde eles são mais importantes.

A simulação foi feita com base no exemplo 6.24 do livro [1]. A Figura 10 mostra a localização de um robô utilizando filtro de partículas. Nesta figura as partículas são ilustradas em diferentes rodadas do filtro.

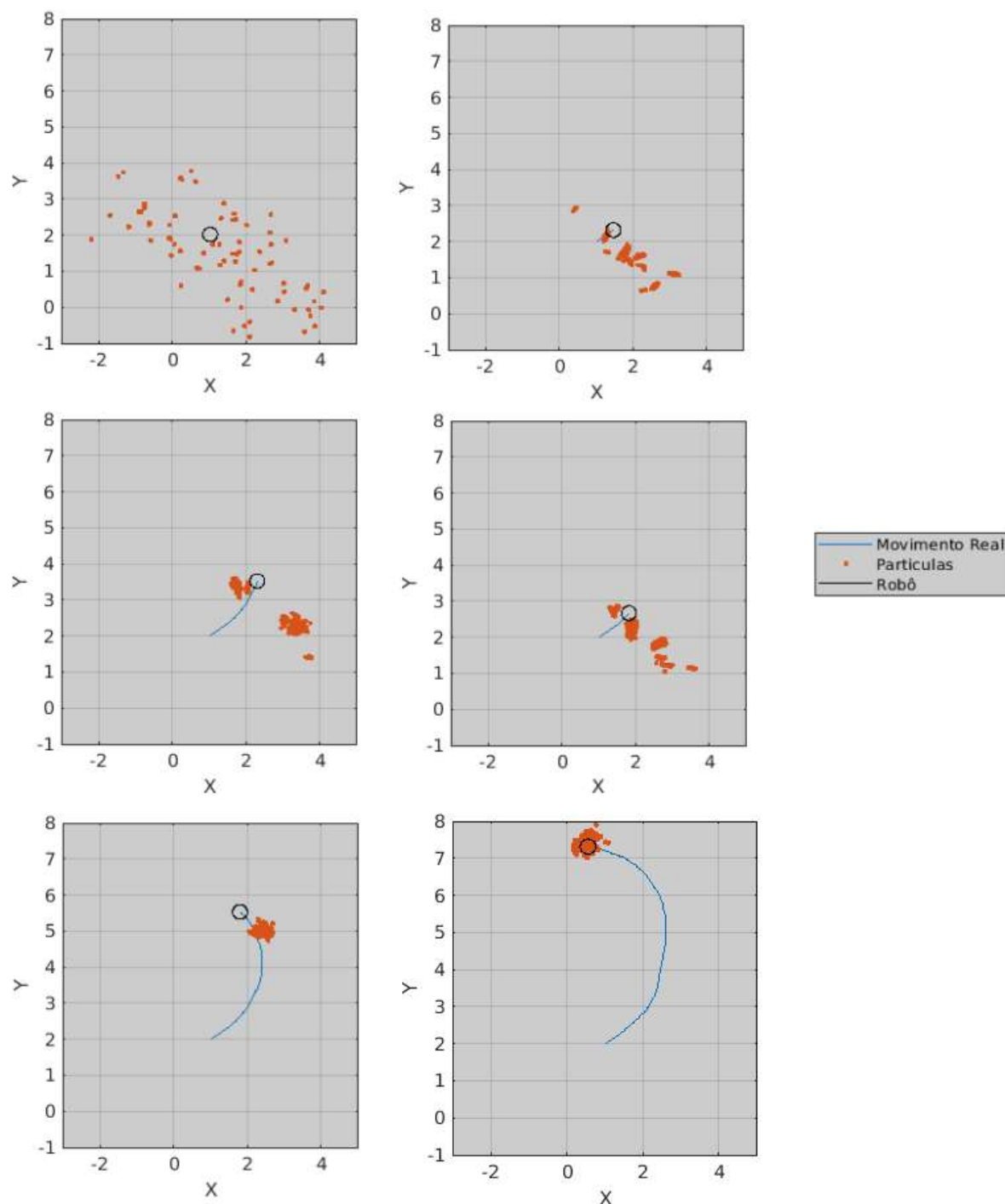


Figura 10 - Simulação do Filtro de Partículas.

AMOSTRADOR DE BAIXA VARIÂNCIA

Um passo relevante no algoritmo do filtro de partículas é a reamostragem. Com a reamostragem, as partículas em uma iteração são selecionadas para a iteração seguinte, em geral de acordo com o peso designado para cada uma destas.

O algoritmo de Low Variance Sampler busca portanto selecionar aleatoriamente um conjunto de partículas que represente a população proporcionalmente ao peso designado às partículas. A Figura 11 apresenta o algoritmo.

A seleção das partículas supõe que os pesos destas é normalizado, sendo realizado o sorteio com distribuição uniforme de um número r entre 0 e $1/M$, sendo M o número de partículas.

As partículas são então selecionadas conforme a Figura 12 ilustra a seleção das partículas, sendo w^n os pesos correspondentes a cada partícula n .

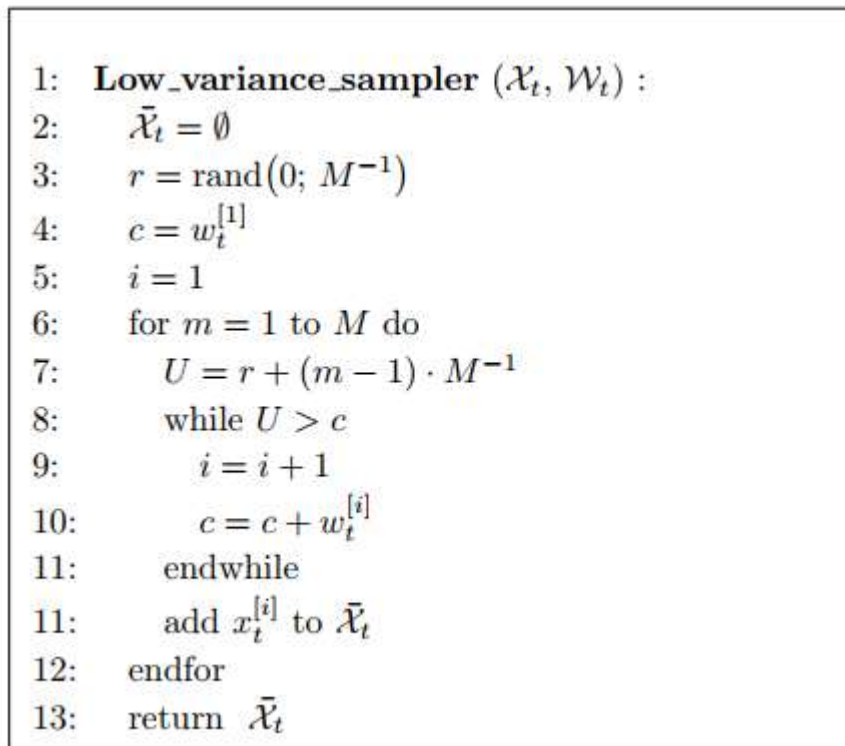


Figura 11 - Algoritmo do Amostrador de Baixa Variância.

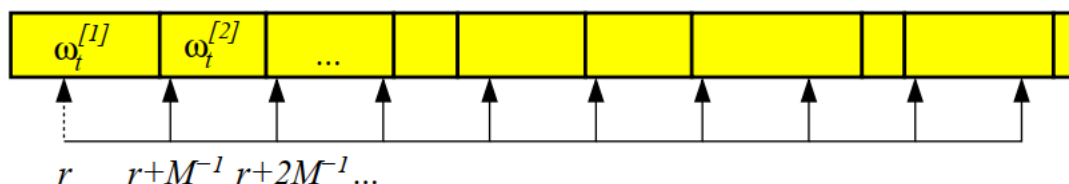


Figura 12 – Ilustração da seleção das partículas.

AMOSTRADOR DE BAIXA VARIÂNCIA - SIMULAÇÃO

Para ilustrar o filtro de partículas e sua variante com amostrador de baixa variância foi realizada uma simulação em Matlab/Simulink, representada na Figura 13.

A simulação representa um robô cujas entradas são a velocidade linear e angular, sendo considerados ruídos (gaussianos) de processo correspondentes. As saídas do sistema correspondem às distâncias entre o robô e pontos de referência, não sendo realizadas medidas dos ângulos correspondentes, sendo acrescentados às saídas ruídos (gaussianos) de medição.

Tanto o modelo do robô como os filtros de partículas convencional e com Low Variance Sampling são implementados por meio de s-functions.

As Figuras 14 a 17 apresentam a comparação entre os resultados, sendo representadas em preto as partículas, em vermelho a posição real e em azul a posição estimada pela média ponderada das posições das partículas.

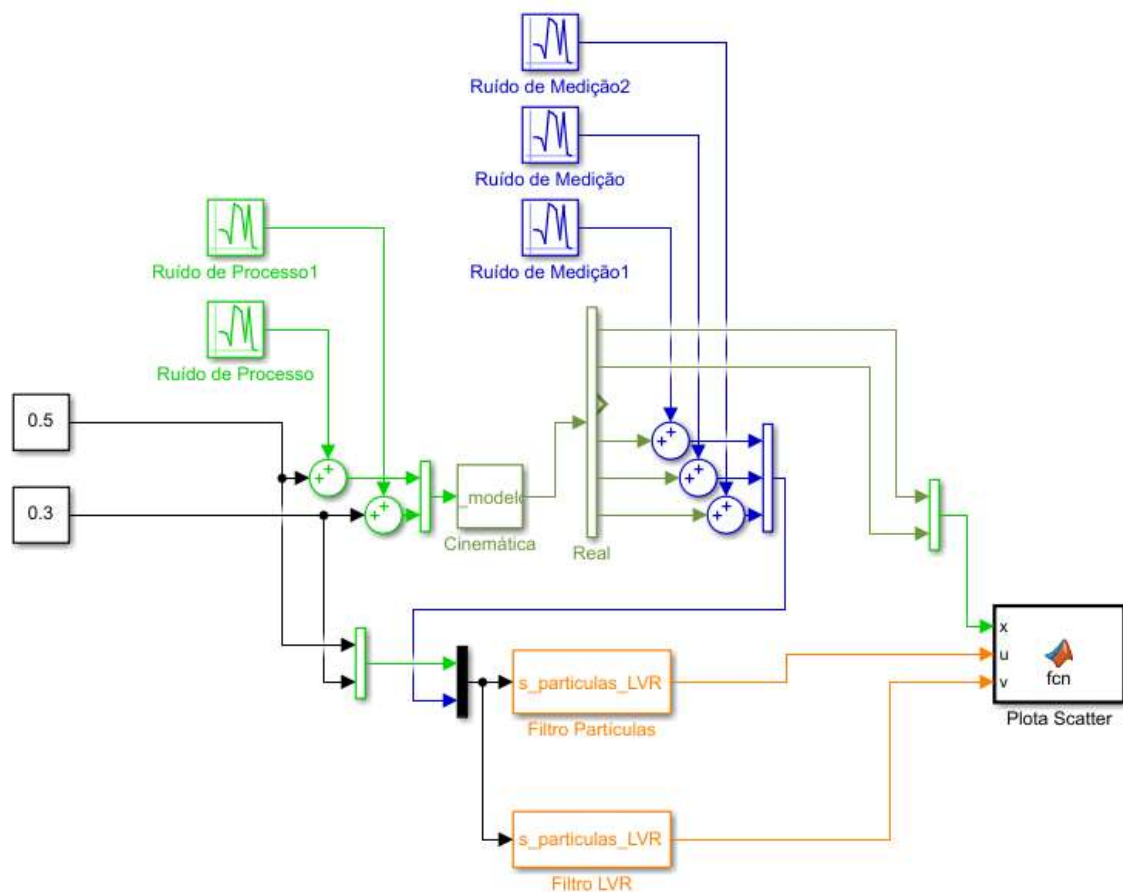


Figura 13 - Simulink do Amostrador de Baixa Variância.

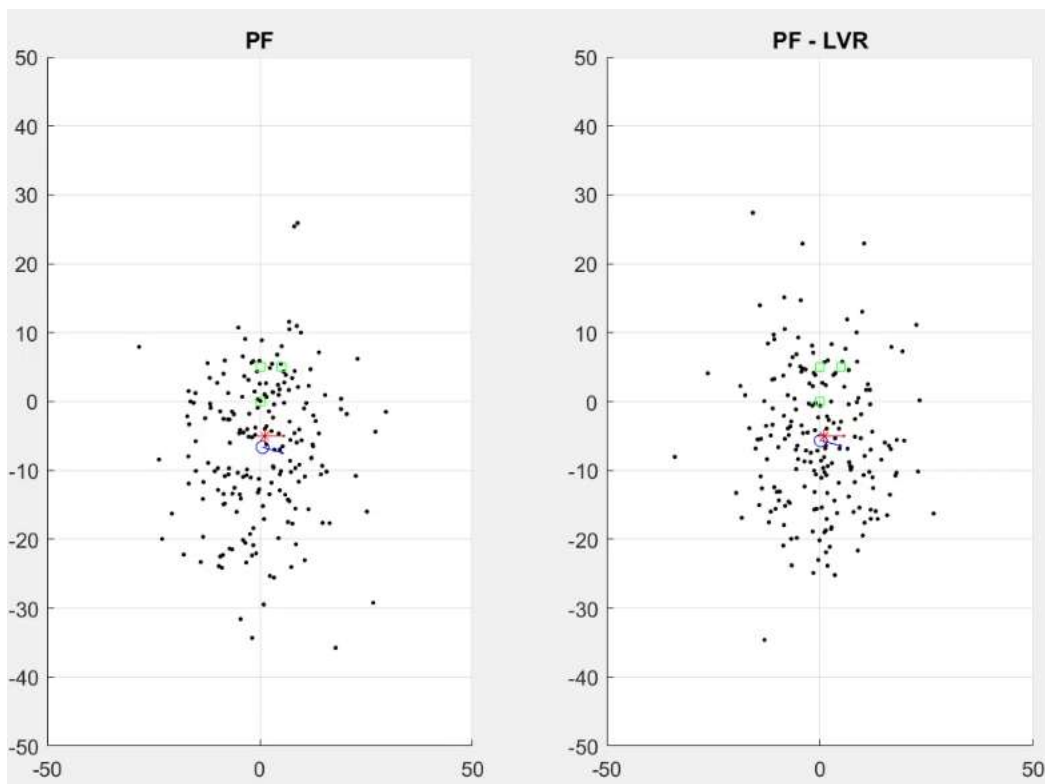


Figura 14 - Representação dos Filtros de Partículas.

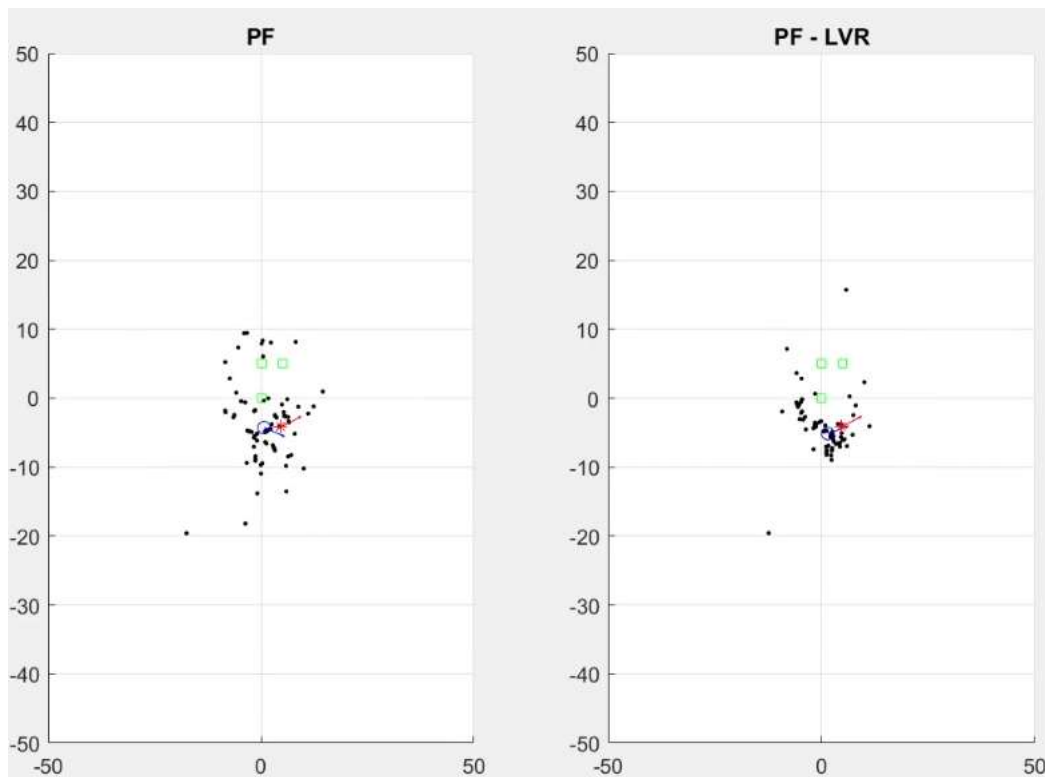


Figura 15 - Representação dos Filtros de Partículas.

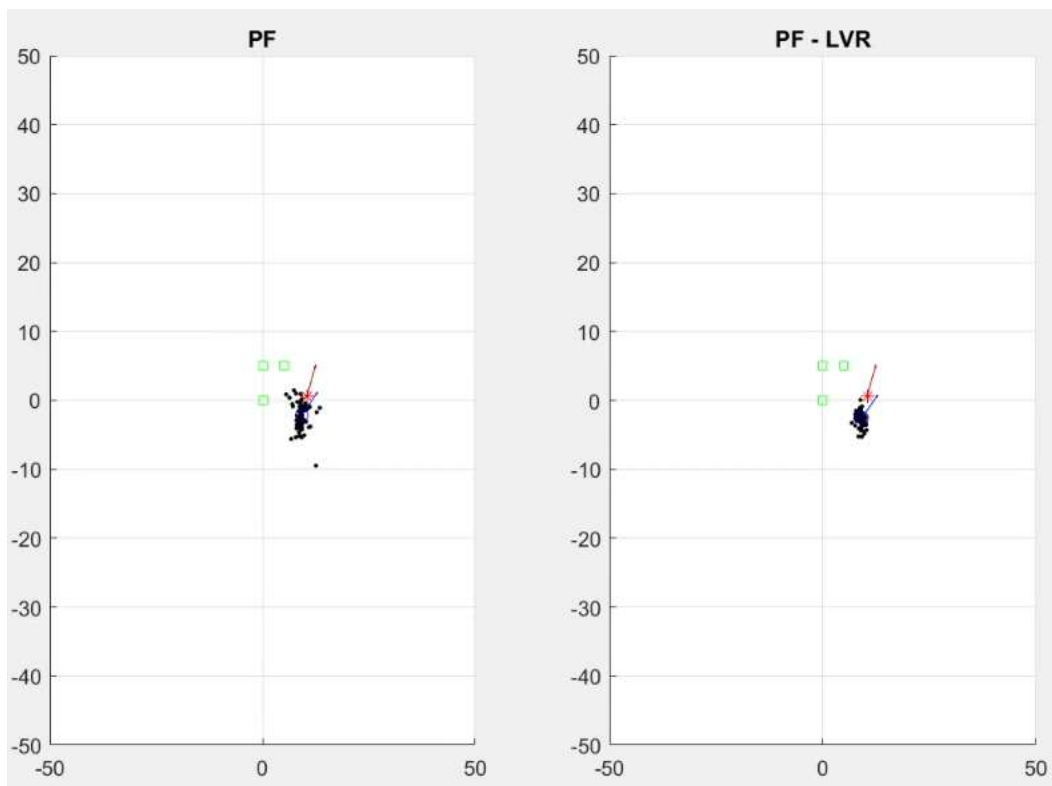


Figura 16 - Representação dos Filtros de Partículas.

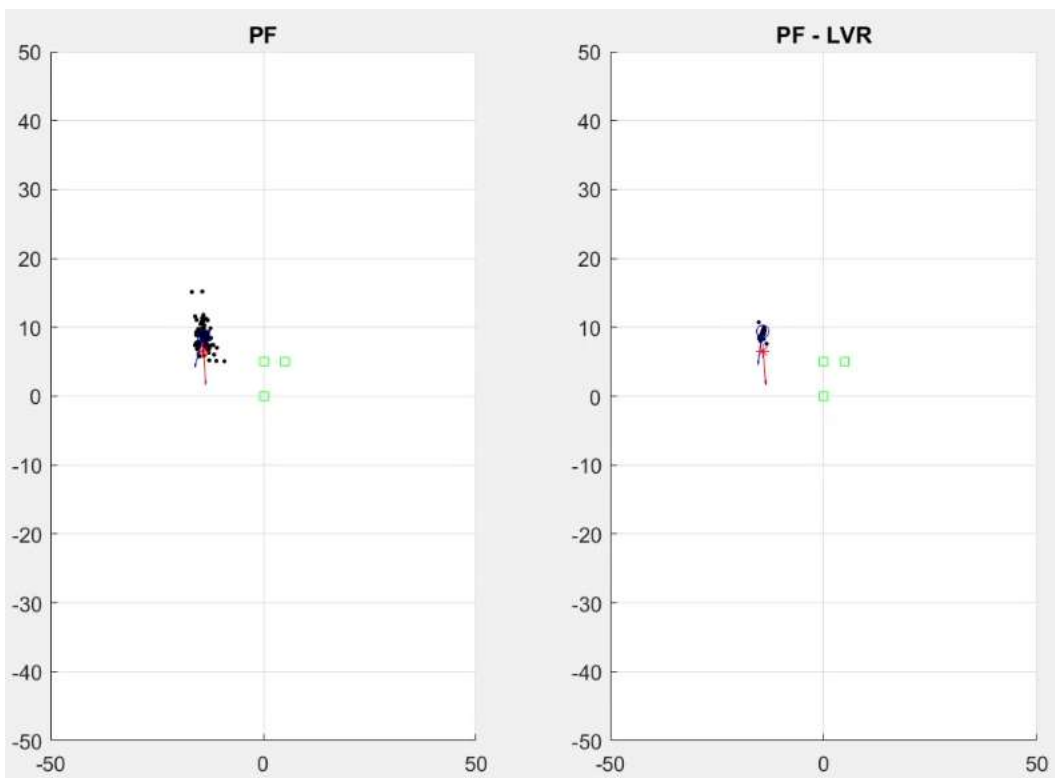


Figura 17 - Representação dos Filtros de Partículas.