

## Trabalho #3 – COE841

Professor: Ramon Romankevicius Costa

Alunos: André Abido Figueiró

Tiago Bornia de Castro

### MODELO DE MOVIMENTO BASEADO EM VELOCIDADE

```
1: Algorithm motion_model_velocity( $x_t, u_t, x_{t-1}$ ):  
2:    $\mu = \frac{1}{2} \frac{(x - x') \cos \theta + (y - y') \sin \theta}{(y - y') \cos \theta - (x - x') \sin \theta}$   
3:    $x^* = \frac{x + x'}{2} + \mu(y - y')$   
4:    $y^* = \frac{y + y'}{2} + \mu(x' - x)$   
5:    $r^* = \sqrt{(x - x^*)^2 + (y - y^*)^2}$   
6:    $\Delta\theta = \text{atan2}(y' - y^*, x' - x^*) - \text{atan2}(y - y^*, x - x^*)$   
7:    $\hat{v} = \frac{\Delta\theta}{\Delta t} r^*$   
8:    $\hat{\omega} = \frac{\Delta\theta}{\Delta t}$   
9:    $\hat{\gamma} = \frac{\theta' - \theta}{\Delta t} - \hat{\omega}$   
10:  return  $\text{prob}(v - \hat{v}, \alpha_1 v^2 + \alpha_2 \omega^2) \cdot \text{prob}(\omega - \hat{\omega}, \alpha_3 v^2 + \alpha_4 \omega^2)$   
       $\cdot \text{prob}(\hat{\gamma}, \alpha_5 v^2 + \alpha_6 \omega^2)$ 
```

Figura 1 - Algoritmo do modelo baseado em movimento.

A Figura 1 apresenta o algoritmo do modelo de movimento baseado em velocidade. O modelo recebe como entrada a posição no momento anterior  $x_{t-1} = (x \ y \ \theta)$ , o controle  $u_t = (v \ \omega)$  e uma posição hipotética  $x_t = (x' \ y' \ \theta')$ . O resultado retornado pelo algoritmo é a probabilidade  $p(x_t | u_t, x_{t-1})$  da posição hipotética corresponder à posição do robô em  $x_t$  após a execução do controle  $u_t$  a partir do estado  $x_{t-1}$ .

Inicialmente o algoritmo calcula, a partir do modelo, os controles que corresponderiam ao movimento do robô, sem considerar ruído, obtendo os parâmetros  $\hat{v}$  e  $\hat{\omega}$ , (linhas 7 e 8). Para tal cálculo é necessário calcular primeiramente as coordenadas do eixo de rotação do movimento (linhas 2 e 3). Na linha 5 é feito o cálculo da distância percorrida e na linha 6 o cálculo variação do ângulo  $\theta$ . A orientação final recebe a perturbação de termo aleatório adicional  $\hat{\gamma}$ , linha 9.

A função  $\text{prob}(x, b^2)$  calcula a probabilidade de seu parâmetro  $x$  sob uma variável aleatória centrada em zero com variância  $b^2$ . Duas implementações possíveis são mostradas na Figura 2, para variáveis de erro com distribuição normal e distribuição triangular,

respectivamente. Os parâmetros  $\alpha_1$  e  $\alpha_2$  modelam o erro de translação, e os parâmetros  $\alpha_3$  e  $\alpha_4$  modelam o erro angular.

```

1:  Algorithm prob_normal_distribution( $a, b^2$ ):
2:      return  $\frac{1}{\sqrt{2\pi} b^2} \exp \left\{ -\frac{1}{2} \frac{a^2}{b^2} \right\}$ 
3:  Algorithm prob_triangular_distribution( $a, b^2$ ):
4:      return  $\max \left\{ 0, \frac{1}{\sqrt{6} b} - \frac{|a|}{6 b^2} \right\}$ 

```

Figura 2 - Algoritmos de distribuições de probabilidade.

A Figura 3 apresenta a implementação do algoritmo das distribuições de probabilidade (Figura2).

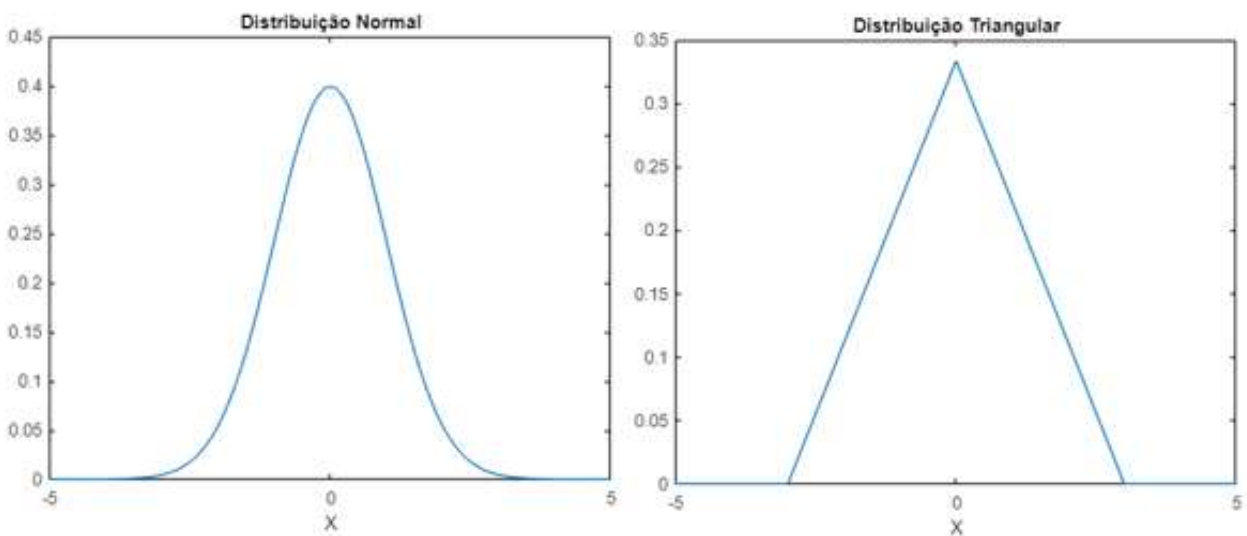


Figura 3 - Distribuições de probabilidade.

A Figura 4 apresenta as implementações do algoritmo do modelo de movimento baseado em velocidade (Figura 1). Na Figura 4.1 foram utilizados valores moderados dos  $\alpha$ . Na Figura 4.2, a simulação utilizou  $\alpha_1$  e  $\alpha_2$  grandes (modelando um grande erro de translação) e  $\alpha_3$  e  $\alpha_4$  pequenos (modelando um pequeno erro angular). Na Figura 4.3, a simulação utilizou  $\alpha_1$  e  $\alpha_2$  pequenos (modelando um pequeno erro de translação) e  $\alpha_3$  e  $\alpha_4$  grandes (modelando um grande erro angular).

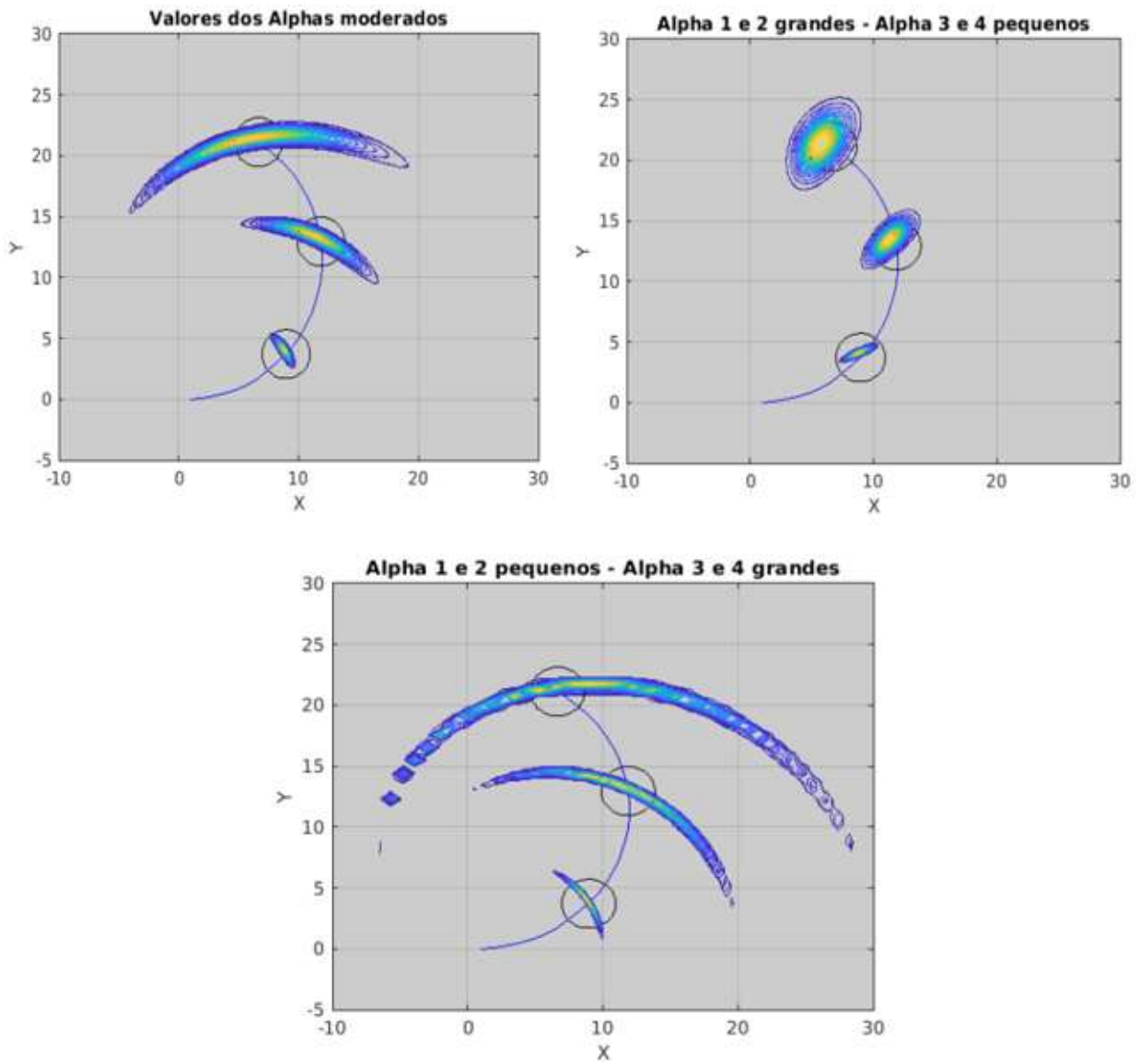


Figura 4 - Modelo baseado em movimento, adaptado de (EECS, 2018).

#### MODELO DE MOVIMENTO BASEADO EM VELOCIDADE - SAMPLE

1: **Algorithm sample\_motion\_model\_velocity( $u_t, x_{t-1}$ ):**

```

2:    $\hat{v} = v + \text{sample}(\alpha_1 v^2 + \alpha_2 \omega^2)$ 
3:    $\hat{\omega} = \omega + \text{sample}(\alpha_3 v^2 + \alpha_4 \omega^2)$ 
4:    $\hat{\gamma} = \text{sample}(\alpha_5 v^2 + \alpha_6 \omega^2)$ 
5:    $x' = x - \frac{\hat{v}}{\hat{\omega}} \sin \theta + \frac{\hat{v}}{\hat{\omega}} \sin(\theta + \hat{\omega} \Delta t)$ 
6:    $y' = y + \frac{\hat{v}}{\hat{\omega}} \cos \theta - \frac{\hat{v}}{\hat{\omega}} \cos(\theta + \hat{\omega} \Delta t)$ 
7:    $\theta' = \theta + \hat{\omega} \Delta t + \hat{\gamma} \Delta t$ 
8:   return  $x_t = (x', y', \theta')^T$ 

```

Figura 5 - Algoritmo do modelo sample baseado em movimento.

A Figura 5 apresenta o algoritmo sample para o modelo baseado em velocidade. Ele recebe como entrada o controle  $u_t$  e o estado  $x_{t-1}$ , e retorna amostras aleatórias  $x_t$  de acordo com a distribuição  $p(x_t | u_t, x_{t-1})$ .

Nas linhas 2 a 4, as amostras incorporam os parâmetros de controle com ruídos. Na linha 4, a orientação final recebe a perturbação de termo aleatório adicional  $\hat{\gamma}$ .

As linhas 5 a 7 geram as novas posições da amostra. Assim, o procedimento de amostragem implementa um modelo de movimento de robô físico simples que incorpora ruído de controle em sua previsão.

Na Figura 6 os algoritmos geram amostras aleatórias de distribuição centrada em zero com variância  $b^2$ . Esses algoritmos podem ser utilizados para amostragem da modelo sample baseado em velocidade. A Figura 7 ilustra os dois algoritmos da Figura 6 para 6 milhões de amostras.

```

1:  Algorithm sample_normal_distribution( $b^2$ ):
2:       $\text{return } \frac{1}{2} \sum_{i=1}^{12} \text{rand}(-b, b)$ 

3:  Algorithm sample_triangular_distribution( $b^2$ ):
4:       $\text{return } \frac{\sqrt{6}}{2} [\text{rand}(-b, b) + \text{rand}(-b, b)]$ 

```

Figura 6 - Algoritmos sample de distribuições de probabilidade.

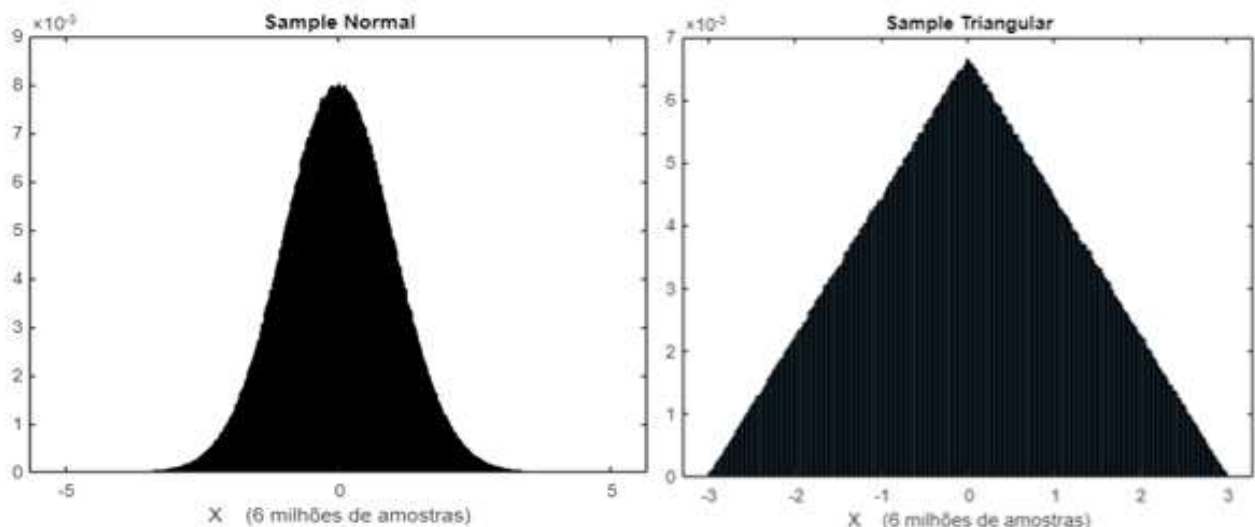


Figura 7 – Amostragem da distribuição normal e triangular.

A Figura 8 apresenta as implementações do algoritmo do modelo sample de movimento baseado em velocidade (Figura 5). Na Figura 8.1 foram utilizados valores moderados dos  $\alpha$ . Na Figura 8.2, a simulação utilizou  $\alpha_1$  e  $\alpha_2$  grandes (modelando um grande erro de translação) e  $\alpha_3$  e  $\alpha_4$  pequenos (modelando um pequeno erro angular). Na Figura 8.3, a simulação utilizou  $\alpha_1$  e  $\alpha_2$  pequenos (modelando um pequeno erro de translação) e  $\alpha_3$  e  $\alpha_4$  grandes (modelando um grande erro angular).

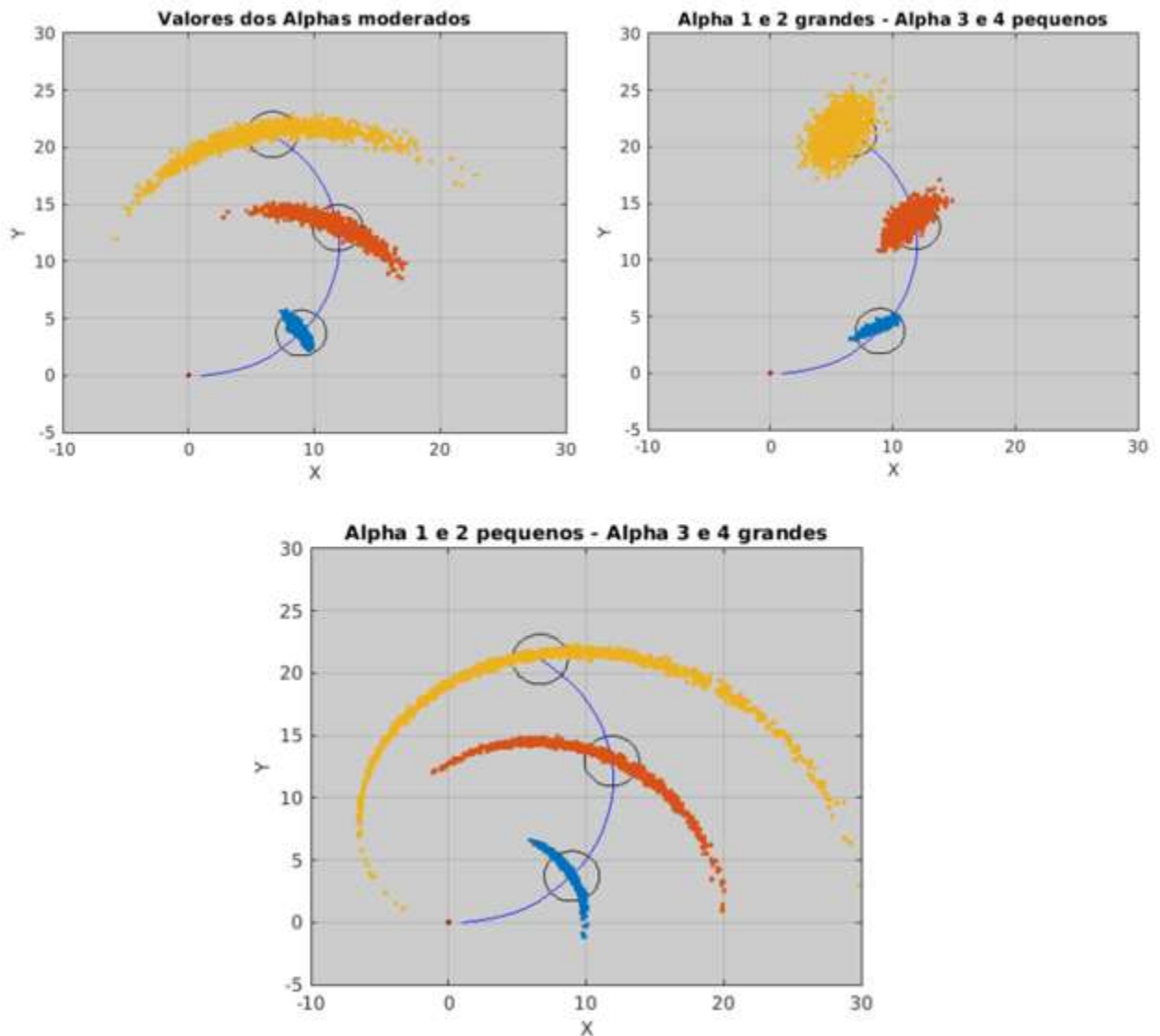


Figura 8 - Modelo sample baseado em movimento, adaptado de (EECS, 2018).

```

1:   Algorithm motion_model_odometry( $x_t, u_t, x_{t-1}$ ):
2:        $\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$ 
3:        $\delta_{trans} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2}$ 
4:        $\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$ 
5:        $\hat{\delta}_{rot1} = \text{atan2}(y' - y, x' - x) - \theta$ 
6:        $\hat{\delta}_{trans} = \sqrt{(x - x')^2 + (y - y')^2}$ 
7:        $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$ 
8:        $p_1 = \text{prob}(\delta_{rot1} - \hat{\delta}_{rot1}, \alpha_1 \hat{\delta}_{rot1}^2 + \alpha_2 \hat{\delta}_{trans}^2)$ 
9:        $p_2 = \text{prob}(\delta_{trans} - \hat{\delta}_{trans}, \alpha_3 \hat{\delta}_{trans}^2 + \alpha_4 \hat{\delta}_{rot1}^2 + \alpha_4 \hat{\delta}_{rot2}^2)$ 
10:       $p_3 = \text{prob}(\delta_{rot2} - \hat{\delta}_{rot2}, \alpha_1 \hat{\delta}_{rot2}^2 + \alpha_2 \hat{\delta}_{trans}^2)$ 
11:      return  $p_1 \cdot p_2 \cdot p_3$ 

```

Figura 9 –Algoritmo para modelo baseado em odometria.

Enquanto o modelo baseado em velocidade utiliza as informações de velocidade linear e angular, o modelo por odometria se utiliza do deslocamento e rotações do robô. Tipicamente tal abordagem faz uso da integração dos encoders das rodas ao longo do tempo, obtendo resultados em geral superiores ao modelo baseado em velocidade.

Para efeito desse algoritmo, as medidas de odometria são consideradas como se fossem entradas (controle) para o robô, sendo o movimento entre dois momentos separado em uma primeira rotação, seguida de uma translação e uma segunda rotação.

A Figura 9 apresenta o algoritmo do modelo de movimento baseado em odometria. O modelo recebe como entrada a posição no momento anterior  $x_{t-1} = (x \ y \ \theta)$ , o controle  $u_t = (\bar{x}_{t-1}, \bar{x}_t)$  que corresponde às medidas da odometria antes e depois do movimento e uma posição hipotética  $x_t = (x' \ y' \ \theta')$ . O resultado retornado pelo algoritmo é a probabilidade  $p(x_t | u_t, x_{t-1})$  da posição hipotética corresponder à posição do robô em  $x_t$  após a execução do controle  $u_t$  a partir do estado  $x_{t-1}$ .

As Figuras 10 a 13 apresentam as simulações para o algoritmo com diferentes valores de  $\alpha = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ . Para a Figura 10 foram utilizados valores baixos, com  $\alpha = [1 \ 1 \ 4 \ 4] \cdot 10^{-4}$ , ao passo que na Figura 11 incrementou-se o valor de  $\alpha_1$  para  $10^{-3}$ , resultando em maior espalhamento da função de probabilidade correspondente à pose do robô. Na Figura 12 incrementou-se o valor de  $\alpha_2$  para  $10^{-3}$ , resultando em maior espalhamento da função de probabilidade correspondente à pose do robô, bem como em alongamento na função de probabilidade correspondente à posição, na direção do movimento.

Na Figura 13 incrementou-se o valor de  $\alpha_3$  para  $2 \times 10^{-3}$ , resultando em maior espalhamento da função de probabilidade correspondente à posição, na direção transversal ao movimento.

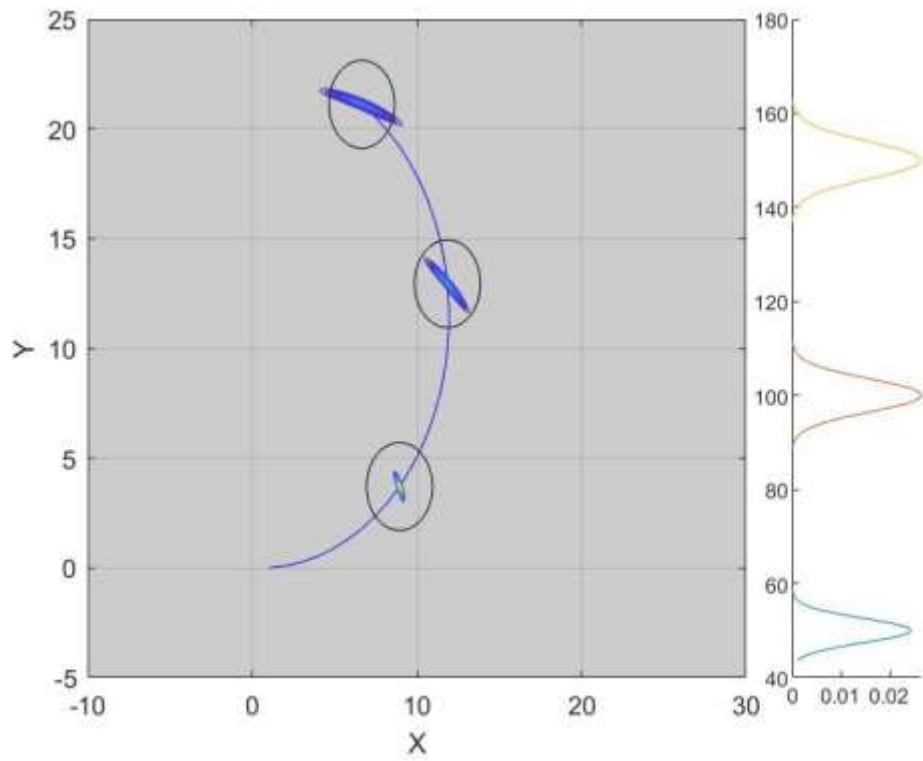


Figura 10

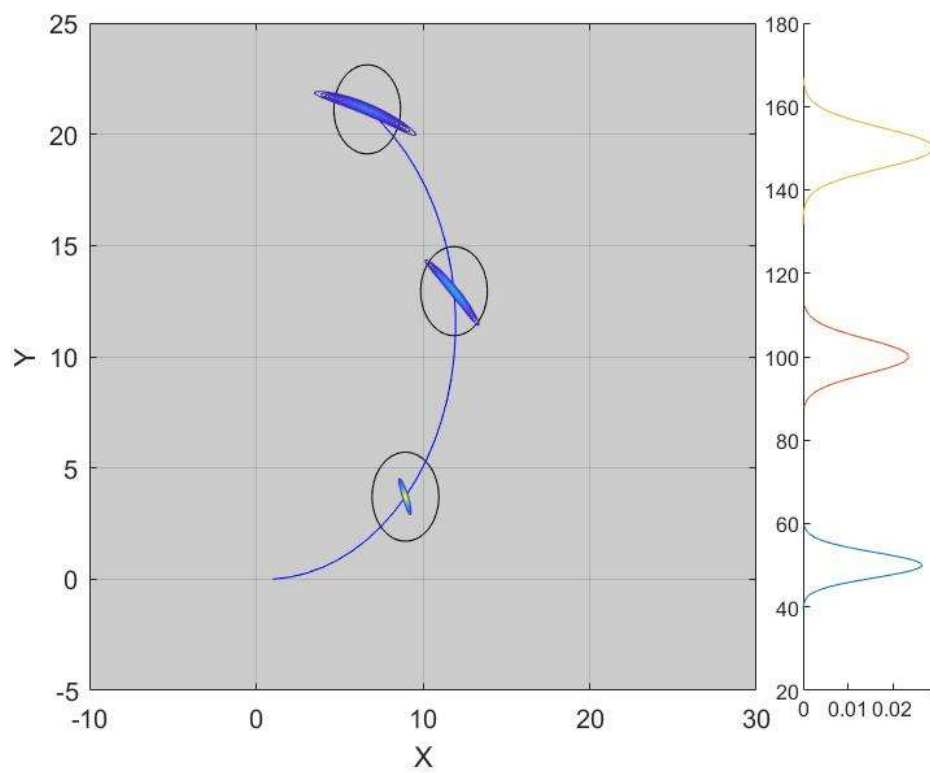


Figura 11

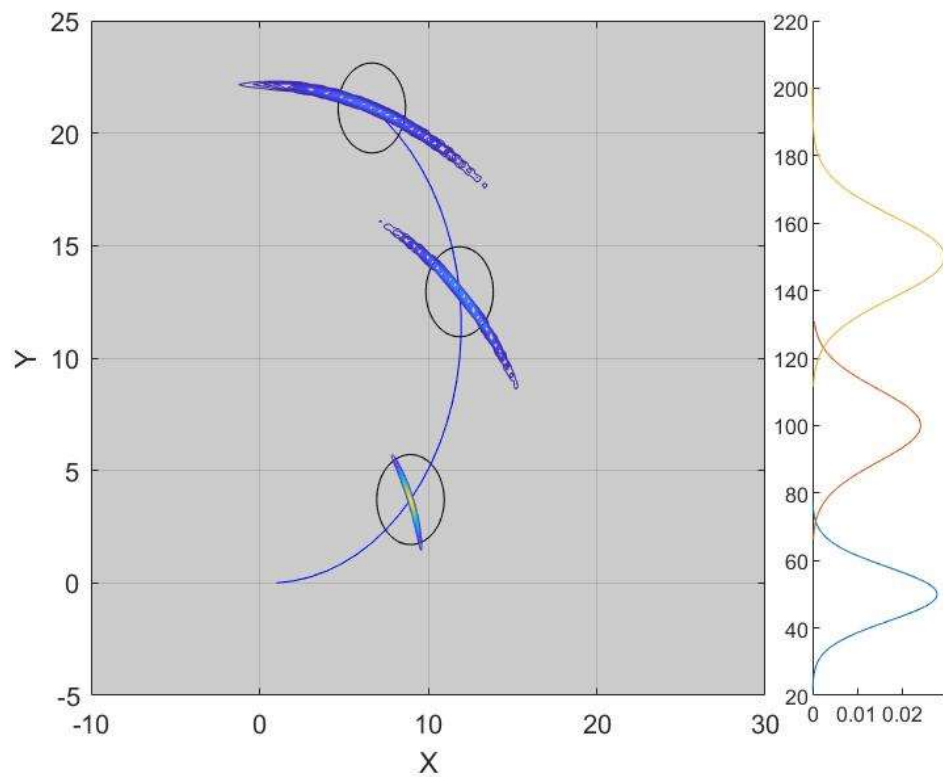


Figura 12

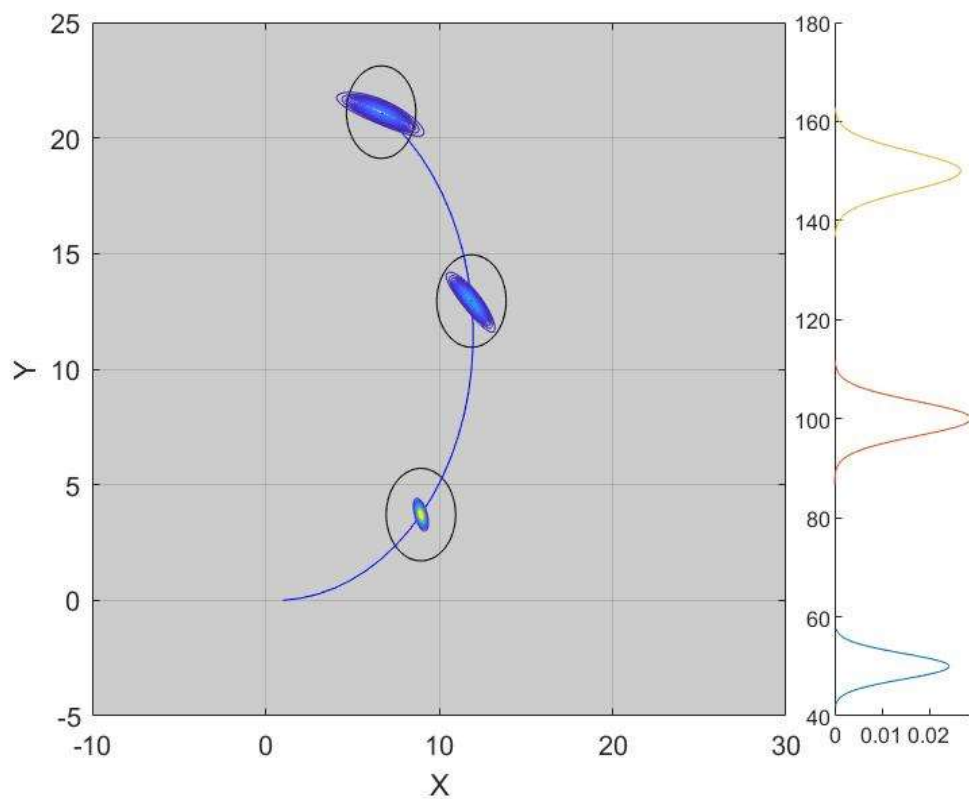


Figura 13



```

1:  Algorithm sample_motion_model_odometry( $u_t, x_{t-1}$ ):
2:       $\delta_{\text{rot1}} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$ 
3:       $\delta_{\text{trans}} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2}$ 
4:       $\delta_{\text{rot2}} = \bar{\theta}' - \bar{\theta} - \delta_{\text{rot1}}$ 

5:       $\hat{\delta}_{\text{rot1}} = \delta_{\text{rot1}} - \text{sample}(\alpha_1 \delta_{\text{rot1}}^2 + \alpha_2 \delta_{\text{trans}}^2)$ 
6:       $\hat{\delta}_{\text{trans}} = \delta_{\text{trans}} - \text{sample}(\alpha_3 \delta_{\text{trans}}^2 + \alpha_4 \delta_{\text{rot1}}^2 + \alpha_4 \delta_{\text{rot2}}^2)$ 
7:       $\hat{\delta}_{\text{rot2}} = \delta_{\text{rot2}} - \text{sample}(\alpha_1 \delta_{\text{rot2}}^2 + \alpha_2 \delta_{\text{trans}}^2)$ 

8:       $x' = x + \hat{\delta}_{\text{trans}} \cos(\theta + \hat{\delta}_{\text{rot1}})$ 
9:       $y' = y + \hat{\delta}_{\text{trans}} \sin(\theta + \hat{\delta}_{\text{rot1}})$ 
10:      $\theta' = \theta + \hat{\delta}_{\text{rot1}} + \hat{\delta}_{\text{rot2}}$ 

11:     return  $x_t = (x', y', \theta')^T$ 

```

Figura 15

A Figura 15 apresenta o algoritmo de sample para o modelo baseado em odometria. O modelo em questão recebe como entradas o estado  $x_{t-1}$  e  $u_t = (\bar{x}_{t-1}, \bar{x}_t)$  que corresponde às medidas da odometria antes e depois do movimento. O algoritmo calcula amostras aleatórias  $x_t$  de acordo com a distribuição  $p(x_t | u_t, x_{t-1})$ .

Nas linhas de 2 a 4 são calculados os movimentos angulares e linear correspondentes às entradas de odometria, ao passo que as linhas de 5 a 7 geram os movimentos correspondentes às amostras, estas distribuídas de acordo com a função de probabilidade correspondente.

Finalmente, as linhas 8 a 10 calculam a posição e pose do robô para cada amostra gerada.

As Figuras 16 a 19 apresentam as simulações para o algoritmo com diferentes valores de  $\alpha = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ . Para a Figura 16 foram utilizados valores baixos, com  $\alpha = [1 \ 1 \ 4 \ 4] \cdot 10^{-4}$ , ao passo que na Figura 17 incrementou-se o valor de  $\alpha_1$  para  $10^{-3}$ , resultando em maior espalhamento da função de probabilidade correspondente à pose do robô. Na Figura 18 incrementou-se o valor de  $\alpha_2$  para  $10^{-3}$ , resultando em maior espalhamento da função de probabilidade correspondente à pose do robô, bem como em alongamento na função de probabilidade correspondente à posição, na direção do movimento.

Na Figura 19 incrementou-se o valor de  $\alpha_3$  para  $2 \cdot 10^{-3}$ , resultando em maior espalhamento da função de probabilidade correspondente à posição, na direção transversal ao movimento.

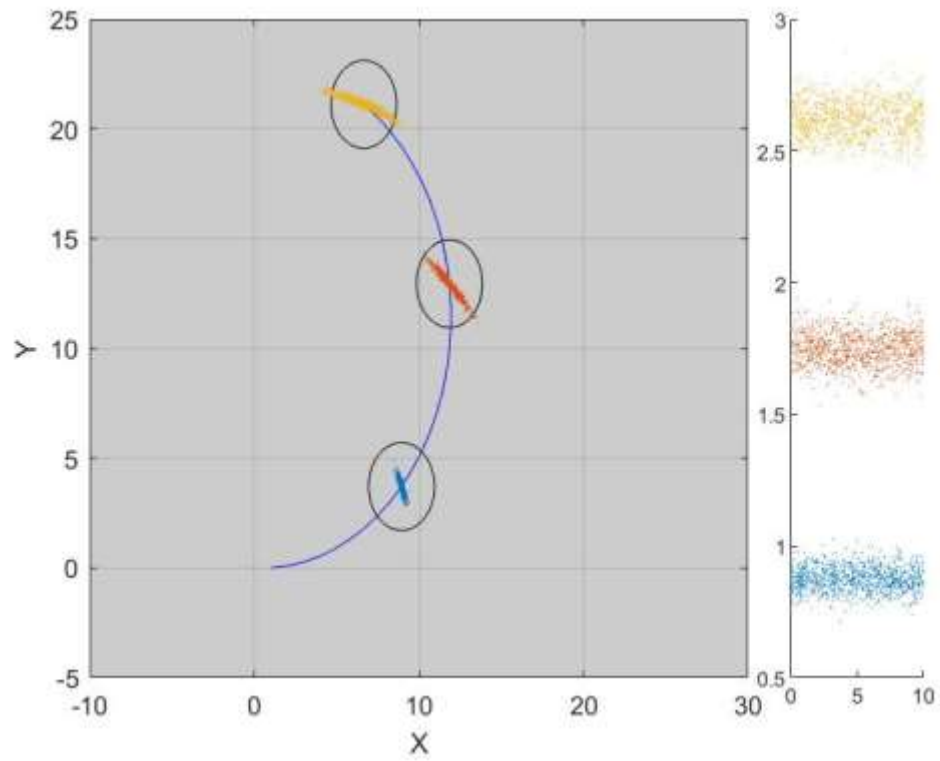


Figura 16

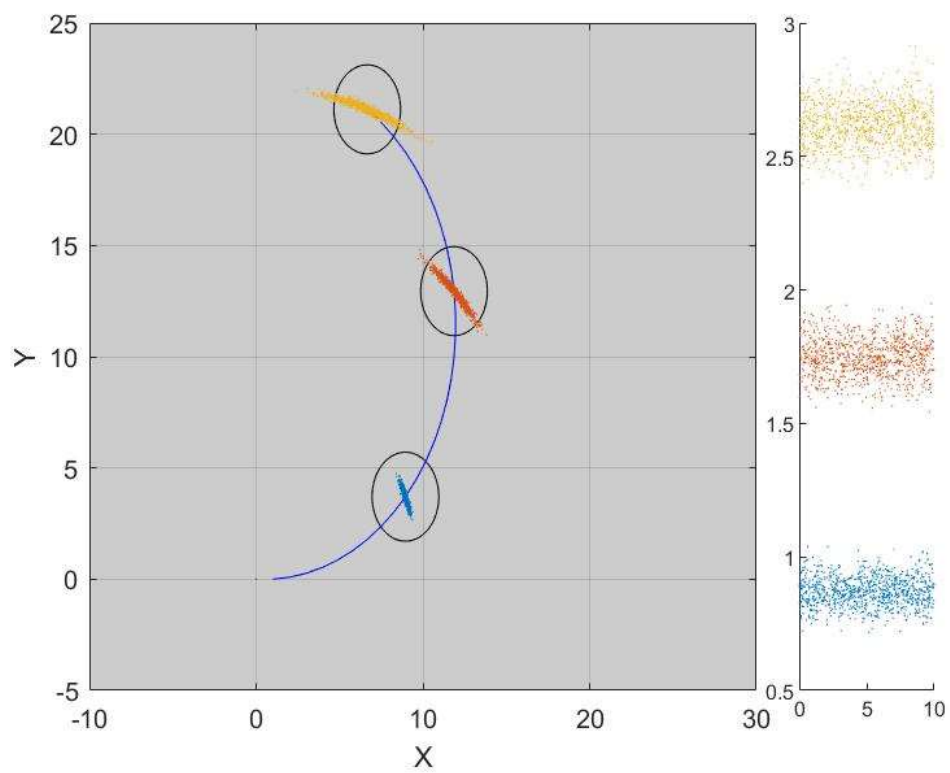


Figura 17

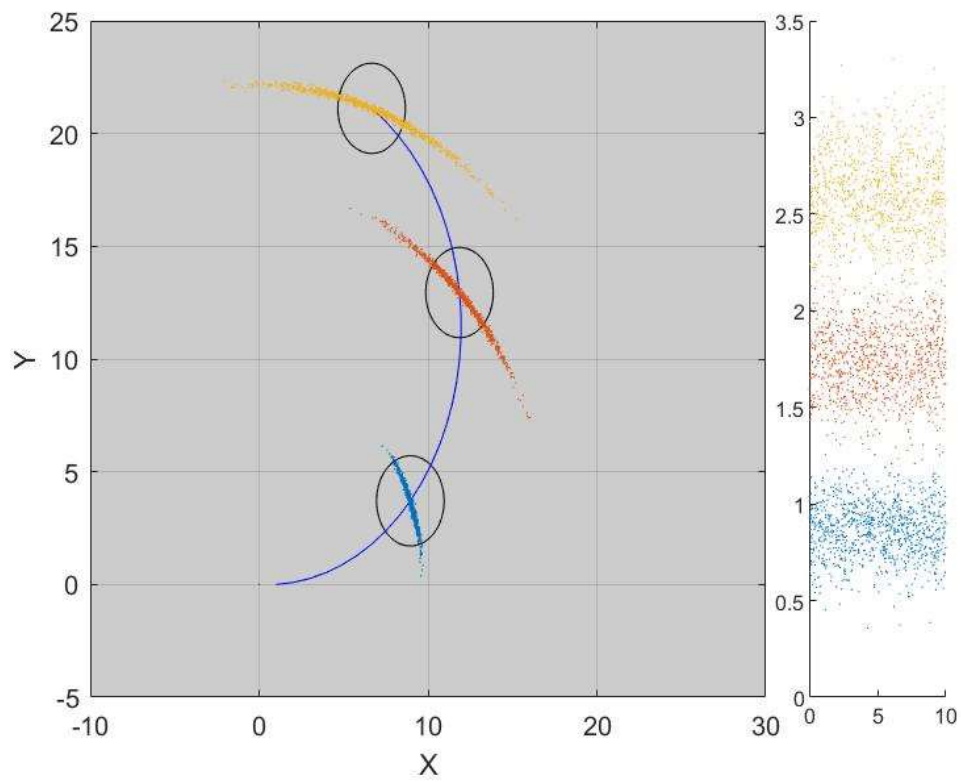


Figura 18

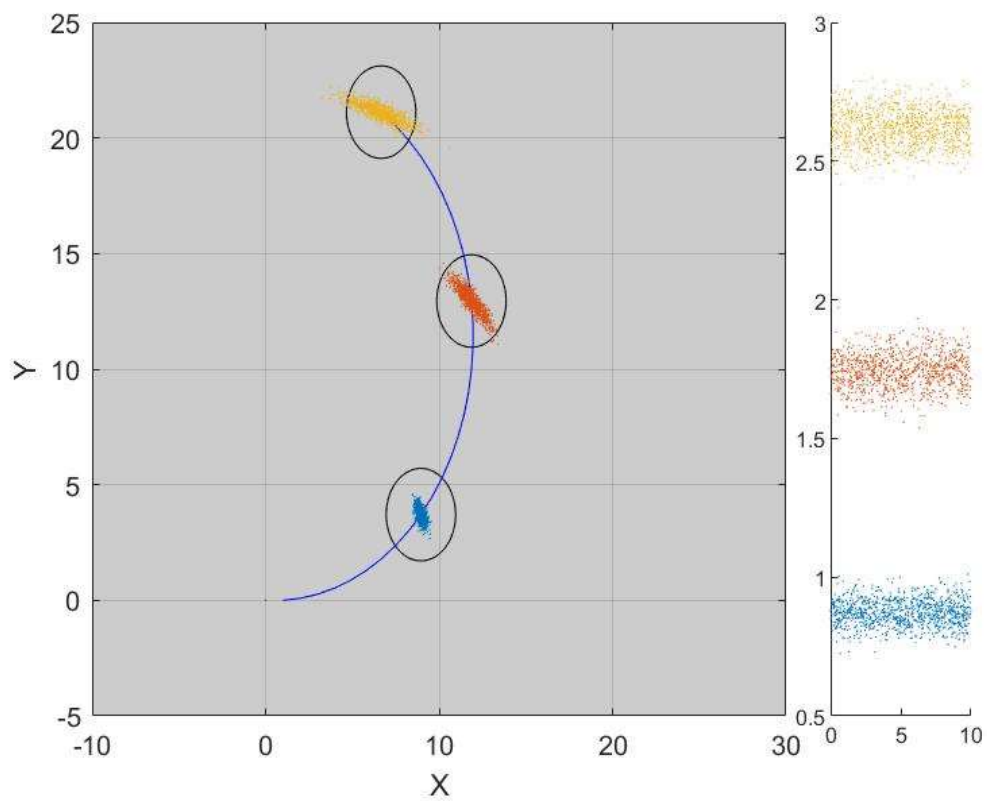


Figura 19

## **REFERÊNCIA**

EECS. (2018). *[https://www.eecs.yorku.ca/course\\_archive/2017-18/W/4421/labs/lab4/lab4.html](https://www.eecs.yorku.ca/course_archive/2017-18/W/4421/labs/lab4/lab4.html)*.