

# LOCALIZAÇÃO POR KALMAN ESTENDIDO APLICADO A MAPAS BASEADOS EM MARCOS COM E SEM CORRESPONDÊNCIA CONHECIDA

DANIEL LEITE<sup>1</sup>, KARLA FIGUEIREDO<sup>2</sup>, MARLEY VELLASCO<sup>1</sup>.

1. *Laboratório de Inteligência Robótica Aplicadas -LIRA, Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro  
R. Marquês de São Vicente, 225 - Gávea, Rio de Janeiro  
E-mails: daniels.leite@hotmail.com, marley@ele.puc-rio.br*

2. *Laboratório de Inteligência Robótica Aplicadas -LIRA, CCMAT/UComp, Centro Universitário da Zona Oeste, UEZO  
Av. Manuel Caldeira de Alvarenga, 1203 - Campo Grande - Rio de Janeiro  
E-mails: karla.figueiredo@gmail.com*

**Abstract**— This paper presents a comparative assessment of location algorithms based on Extended Kalman Filter for maps represented by landmarks, with or without knowledge of the correspondence of these landmarks, comparing their performance in local and global localization issues. To evaluate these algorithms, the robot simulator Player / Stage will be used in order to evaluate a mobile robot with embedded color sensors and distance, which will detect artificial landmarks that compose the map, and then estimate the location of the robot. To simulate this uncertainty in the localization process, Gaussian noise will be inserted in the readings of the sensors and the inputs of the actuators. The artificial landmarks will have their signature based on color.

**Keywords**— Localization; Extended Kalman Filter; mobile robot; PlayerStage.

**Resumo**— Este trabalho apresenta uma avaliação comparativa de algoritmos de localização baseado em Filtro de Kalman Estendido para mapas representados por marcos, com e sem conhecimento da correspondência desses marcos, comparando seu desempenho em problemas de localização local e global. Para avaliação desses algoritmos, será utilizado o simulador de robôs Player/Stage, com o objetivo de avaliar um robô móvel embarcado com sensores de cor e distância, que irão detectar os marcos artificiais que compõe o mapa, para então, estimar a localização do robô. Para simular a incerteza presente no processo de localização, serão inseridos ruídos Gaussianos, nas leituras dos sensores e nas entradas dos atuadores. Os marcos artificiais terão sua assinatura baseado na cor.

**Palavras-chave**— Localização; Filtro de Kalman Estendido; Robótica móvel; PlayerStage.

## 1 Introdução

O problema de localização é um dos temas mais abordados no estudo da robótica móvel, e é de fundamental importância para a realização das tarefas que envolvam robôs móveis em ambientes reais. Esta tarefa consiste em estimar as coordenadas do robô em relação a um sistema de referência externa (Thrun, et al., 2005). Para tal, o robô necessita de um mapa que represente ou modele o ambiente, sendo este mapa previamente conhecido ou estimado, a partir de um processo iterativo de localização e mapeamento (SLAM) (Thrun, et al., 2005, pp. 309-317), além de sensores, com os quais serão extraídos os dados para identificar a posição do robô no mapa.

Determinar sua própria posição, dentro de um ambiente, é uma capacidade básica e fundamental para que tarefas de navegação, como o planejamento de uma trajetória, seja executada (Fox, 1999). O grande desafio na solução do problema de localização está no fato de que, tanto as informações sobre o ambiente, como os dados fornecidos pelos sensores, são normalmente limitados e imprecisos, e podem apresentar alto grau de incerteza, tornando a tarefa de localização tão ou mais complexa quanto a tarefa principal do robô.

A complexidade da tarefa de localização reside em diversos fatores como:

- *Imprecisão e ruído dos sensores;*
- *Conhecimento sobre a posição inicial;*
- *Problema de “sequestro”;*
- *Ambientes estáticos x dinâmicos;*

Assim, o problema da localização vem sendo amplamente estudado pelos pesquisadores da área o que contribuiu para a existência de algoritmos bastante eficientes. Esses algoritmos são, em geral, baseados em técnicas probabilísticas como Filtro de Kalman (Tuna, et al., 2012) e (Faisal, et al., 2014) e Localização de Monte Carlo (Dellaert, et al., 1999) e (Ruchti, et al., 2015), além de combinações e variações destes (Zhang, et al., 2006).

Nesse trabalho serão utilizadas técnicas probabilísticas baseadas no Filtro de Kalman Estendido (Thrun, et al., 2005, pp. 54-61) com o intuito analisar e comparar as limitações dessas técnicas para um problema de localização, local e global, para mapas baseados em marcos, com e sem correspondências conhecidas, mostrando o aumento da complexidade do problema de localização ao não se conhecer a correspondência entre os marcos e/ou estado inicial do robô móvel.

As avaliações foram realizadas no simulador 2D Stage (Gerkey & contributors, 2010) em conjunto com

o framework de robótica Player (Gerkey & contributors, 2010). Essa arquitetura foi escolhida, principalmente, por permitir uma fácil migração dos algoritmos de controle para uma plataforma real, objetivo final dos pesquisadores envolvidos nesse trabalho.

Assim, na seção 2 será feita uma breve descrição desses modelos e técnicas, na seção 3 as simulações realizadas e comparação dos resultados e, finalmente, a última seção, apresenta as conclusões e os próximos passos desse trabalho.

## 2 Metodologia

A seguir, serão apresentadas a estrutura do mapa utilizado para localização do robô, as características do ambiente de simulação e uma breve descrição sobre os algoritmos de localização implementados.

### 2.1 Estrutura do Mapa

Para que o robô possa se localizar em um dado ambiente, ele necessita de um mapa, ou seja, conjunto de objetos que representa esse ambiente. Esse mapa, em geral, é em forma de grade (Thrun, et al., 2005, pp. 284-292) ou baseado em marcos (Thrun, et al., 2005, pp. 176-179), sendo o último aplicado nesse trabalho.

Na representação de mapas baseados em marcos, alguns objetos do mapa são simbolizados e estes servem como referência para localização do robô móvel. Essa representação tem como vantagem facilitar a definição e inserção de novos marcos, além de reduzir a dimensão dos dados que serão tratados pelos algoritmos de localização (Thrun, et al., 2005). Contudo, possui uma menor precisão que o modelo por grade, pois apenas parte do ambiente passa a ser modelado.

A representação do mapa baseado em marcos foi a escolhida para esse trabalho devido ao menor custo computacional, quando comparada a representação por grade. Esse fator foi levado em consideração, pois, como se visa a futura migração para uma plataforma móvel real, a capacidade de processamento se tornará um fator importante para a execução do algoritmo em tempo real.

Os dados extraídos do sensor de identificação dos marcos,  $z_t$ , serão usados como argumento de uma função,  $f(z_t)$ , que irá transformar os valores lidos nas informações que serão tratadas pelos algoritmos de localização, reduzindo assim a dimensão dos dados de medição.

A função,  $f(z_t)$ , utilizada irá retornar um vetor  $3 \times 1$  correspondente a parâmetros relativos a cada marco identificado, cujos valores são:

- $r_t^i$ : a distância do robô ao objeto  $i$  observado no instante  $t$ ;
- $\phi_t^i$ : o ângulo entre o robô o objeto  $i$  observado no instante  $t$ ;
- $s_t^i$ : a assinatura, ou característica identificadora, do objeto  $i$  observado no instante  $t$ .

Em casa instante de tempo, mais de um marco pode ser identificado, portanto, a saída da função  $f(z_t)$  será uma lista de vetores (equação 1).

$$f(z_t) = \{f_t^1, f_t^2, \dots\} = \left\{ \begin{pmatrix} r_t^1 \\ \phi_t^1 \\ s_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ \phi_t^2 \\ s_t^2 \end{pmatrix}, \dots \right\} \quad (1)$$

Serão implementados algoritmos para os casos em que a correspondência dos marcos é conhecida, ou seja, o marco pode ser identificado unicamente no mapa, e para o caso em que essa correspondência é desconhecida.

Na configuração com correspondência conhecida, cada marco terá uma cor diferente que será usada como seu identificador, ou assinatura, já no caso para correspondência desconhecidas, todos os marcos terão a mesma cor, impossibilitando sua distinção dentre os demais objetos do mapa, apenas com a informação coletada pelos sensores de cor. Em ambos os casos serão usados os mesmos sensores e com os mesmos ruídos inseridos artificialmente, que serão melhor detalhados nas seções 2.2 e 3.

### 2.2 Ambiente de simulação e sensores

Para criar o ambiente virtual no qual serão testados os algoritmos de localização, foi utilizado o simulador para robôs 2D Stage (Vaughan & contributors, 2011)

Para simulação do robô móvel foram utilizados os módulos:

- *blobfinder*: simula um sensor de cor realizado por uma câmera. Esse sensor percebe e identifica os marcos artificiais que foram inseridos no ambiente, sendo dele extraídas informações como: cor, tamanho e distância, angular e linear do objeto detectado.
- *model*: simula objetos com propriedades básicas como: tamanho, cor, visibilidade, reflexibilidade. Esse módulo foi utilizado para criar os marcos artificiais, que foram inseridos no ambiente para o modelo de mapas baseados em marcos. Cada marco corresponde a um objeto de 10cmx10cm de área de base e altura 1 metro, e onde a propriedade que identifica cada marco é sua cor.
- *position2d*: Esse módulo foi utilizado para criar no simulador um robô móvel com a base diferencial. A posição do robô, após o deslocamento, foi estimada, com base nas velocidades, linear e angular, aplicadas e sua posição antes de se deslocar, e comparada com a posição real, fornecida pelo simulador. Para aproximar o modelo simulado a um modelo real, foram inseridos ruídos gaussianos (média zero e desvio igual a um) nos valores das velocidades.
- *ranger*: simula um *array* de ultrassom ou infravermelho. Esse módulo permite especificar características como “*Field of View*” (FOV), resolução e alcance. Ele foi utilizado para dar

capacidade ao robô de evitar obstáculos, durante a navegação.

A partir dos módulos especificados, foi criado um ambiente similar a uma planta baixa de uma casa de 16m x 16m, com dezessete marcros artificiais inseridos em pontos conhecidos, conforme apresenta a figura 1. Esse ambiente será utilizado para a avaliação dos algoritmos com correspondência dos marcros conhecidos, sendo essa identificação, ou assinatura, baseada na cor.

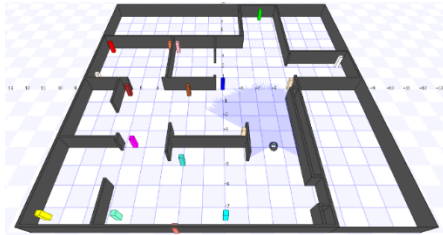


Figura 1. Perspectiva do ambiente para simulação com correspondência dos mapas conhecida

Para o caso do mapa cuja correspondência entre os marcros é desconhecida, foi utilizada a mesma planta baixa, porém com 20 marcros, e todos da mesma cor, conforme mostra a figura 2. O maior número de marcros foi necessário para tentar garantir que o robô estivesse observando pelo menos um marco a cada interação.

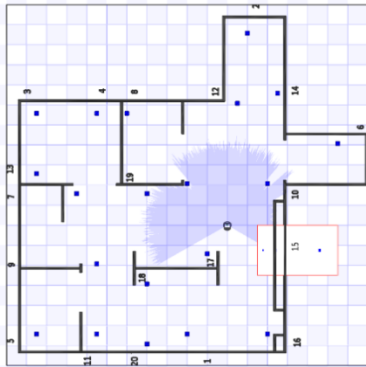


Figura 2. Visão superior ambiente para simulação com correspondência dos mapas desconhecida

### 2.3.1 Localização por filtro de Kalman Estendido com correspondência dos marcros conhecida (EKF-NC)

Conforme apresentado anteriormente, a representação dos mapas nesse trabalho é baseada na extração de características específicas do ambiente, ou seja, mapas baseados em marcros. Portanto, os dados fornecidos pelos sensores serão processados de modo a fornecer um vetor de características para cada objeto identificado, conforme apresentado na equação 1.

O algoritmo de Filtro de Kalman Estendido, para um problema de localização com mapas baseados em marcros de correspondência conhecida, foi aplicado conforme em (Thrun, et al., 2005, pp. 204-209), e pode ser observado na tabela 1.

Tabela 1. Algoritmo de localização com Filtro de Kalman Estendido e correspondência entre os marcros conhecida – EKF-NC

<b>Algoritmo de Localização EKF com Correspondência Conhecida</b> ( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t, m$ )	
1:	$\bar{\mu}_t = g(u_t, \mu_{t-1})$
2:	$\bar{\Sigma}_t = G_t * \Sigma_{t-1} * G_t^T + V_t M_t V_t^T$
3:	for all observed features $z_t^i = (r_t^i, \phi_t^i, s_t^i)^T$ do
4:	$j = c_t^i$
5:	$\bar{z}_t^i = h(\bar{\mu}_t, j, m)$
6:	$K_t^i = \bar{\Sigma}_t * H_t^{iT} * (H_t^i * \bar{\Sigma}_t * H_t^{iT} + Q_t)^{-1}$
7:	$\bar{\mu}_t = \bar{\mu}_t + K_t^i * (z_t^i - \bar{z}_t^i)$
8:	$\bar{\Sigma}_t = (I - K_t^i * H_t^i) * \bar{\Sigma}_t$
9:	endfor
10:	$\mu_t = \bar{\mu}_t$
11:	$\Sigma_t = \bar{\Sigma}_t$
12:	return $\mu_t, \Sigma_t$

Onde nas linhas 1 e 2 ocorre a etapa de predição da estimativa do estado pela estimativa de seus termos de primeiro,  $\bar{\mu}_t$ , e segundo momento,  $\bar{\Sigma}_t$ , e nas linhas de 3 à 9 ocorre a correção dessa predição.

Um fato importante a ser destacado, é que a correção é feita individualmente para cada objeto que foi observado no instante  $t$ , ou seja, a etapa de correção é realizada como se apenas um objeto fosse observado por vez, corrigindo os valores de predição da estimativa da média e covariância, para gerar novas predições. Isso é possível, dado que foi assumida a condição de independência entre os objetos do mapa.

Após a correção dada pela observação de cada objeto, a estimativa do valor médio e covariância do estado são atualizadas, linhas 10 e 11, com a última predição corrigida.

### 2.3.2 Localização por filtro de Kalman Estendido com correspondência dos marcros desconhecida- EKF-UC

O algoritmo de localização por EKF para mapas, baseados em marcros com correspondência desconhecida, se difere da aplicação anterior por não se conhecer a correspondência entre os marcros. Nesse caso, dado que um objeto foi observado pelos sensores, não se sabe exatamente a que objeto do mapa ele corresponde, tendo-se apenas uma distribuição de probabilidade para essa correspondência. A incapacidade de identificar o objeto unicamente irá aumentar o grau de dificuldade de localização do robô no mapa, pois a correção de sua localização estará comprometida com o nível de incerteza que o robô tem quanto a identidade do objeto que está sendo observado (Thrun, et al., 2005).

Para que o robô possa estimar a qual objeto do mapa pertence o objeto que está sendo observado, é calculada da distribuição de probabilidade do valor medido, dada a estimativa do estado atual, para cada objeto do mapa, escolhendo aquele que apresentar maior probabilidade dentre todos. A partir dessa adaptação, o algoritmo da tabela 1 pode ser reescrito conforme a tabela 2 (Thrun, et al., 2005, pp. 215-218).

Tabela 2. Algoritmo de localização com Filtro de Kalman Estendido e correspondência entre os marcos desconhecida – (EKF-UC)

Algoritmo de Localização EKF com Correspondência	EKF com Desconhecida
$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t, m)$ 1: $\bar{\mu}_t = g(u_t, \mu_{t-1})$ 2: $\bar{\Sigma}_t = G_t * \Sigma_{t-1} * G_t^T + V_t M_t V_t^T$ 3: for all observed features $z_t^i = (r_t^i, \phi_t^i, s_t^i)^T$ do 4: for all landmarks $k$ in the map $m$ do 5: $\bar{z}_t^k = h(\bar{\mu}_t, k, m)$ 6: $S_t^k = H_t^k * \bar{\Sigma}_t * H_t^{kT} + Q_t$ 7: endfor 8: $j(i) = \underset{k}{\operatorname{argmax}} \det(2\pi S_t^k)^{-\frac{1}{2}} * \exp\{-\frac{1}{2}(z_t - \bar{z}_t^k)^T [S_t^k]^{-1} * (z_t - \bar{z}_t^k)\}$ 9: $K_t^i = \bar{\Sigma}_t * [H_t^{j(i)}]^T * (H_t^{j(i)} * \bar{\Sigma}_t * [H_t^{j(i)}]^T + Q_t)^{-1}$ 10: $\bar{\mu}_t = \bar{\mu}_t + K_t^i * (z_t^i - \bar{z}_t^{j(i)})$ 11: $\bar{\Sigma}_t = (I - K_t^i * H_t^{j(i)}) * \bar{\Sigma}_t$ 12: endfor 13: $\mu_t = \bar{\mu}_t$ 14: $\Sigma_t = \bar{\Sigma}_t$ 15: return $\mu_t, \Sigma_t$	

A principal diferença desse algoritmo, para o apresentado na tabela 1, se encontra nas linhas de 4 à 8, onde são calculadas as distribuições de probabilidade do valor medido, dado o estado estimado, para cada objeto do mapa. Ao obter todas as distribuições, o objeto que possuir maior probabilidade é o escolhido para ser o utilizado no cálculo da correção, linha 8.

### 3 Experimentos

Nessa etapa serão realizados os testes com os dois algoritmos baseados no Filtro de Kalman Estendido.

Serão utilizadas duas versões de mapa, marcos com cores diferentes para que eles possam ser identificados unicamente e marcos com cores iguais.

Para que o robô execute a mesma trajetória durante a simulação de cada um dos métodos, foi inserido o mesmo conjunto de velocidades linear e angular na entrada dos motores. Esse vetor de controle possui velocidades lineares de -0,28m/s à 1,1m/s, com média real de 0,22m/s e média absoluta de 0,4597m/s, e velocidades angulares de -0.69575rad/s à 0.937281rad/s, com média real de -0.00053rad/s e média absoluta de 0,3139rad/s.

Para simular o ruído do sistema, para cada entrada de velocidade desejada, foi adicionado um ruído Gaussiano segundo as equações 2-5.

$$v_{aplicada} = v_{desejada} + N(0, \alpha_1) + \alpha_2 * (w_{desejada} + N(0, \alpha_4)) \quad (2)$$

$$w_{aplicada} = w_{desejada} + N(0, \alpha_4) + \alpha_3 * (v_{desejada} + N(0, \alpha_1)) \quad (3)$$

$$v_{computada} = v_{aplicada} + N(0, \alpha_1) + \alpha_2 * (w_{aplicada} + N(0, \alpha_4)) \quad (4)$$

$$w_{computada} = w_{aplicada} + N(0, \alpha_4) + \alpha_3 * (v_{aplicada} + N(0, \alpha_1)) \quad (5)$$

Onde,  $\alpha_1 = \alpha_3 = 0.15$  e  $\alpha_2 = \alpha_4 = 0.25$

Portando, para cada valor desejado de entrada das velocidades do motor foram geradas duas novas velocidades acrescidas de ruído, onde uma era aplicada no motor e a outra no algoritmo de localização. Um fato importante a ser observado é que o ruído aplicado não é proporcional a velocidade, portando, valores de velocidade com módulo pequeno, tende a ser mais influenciados pelo ruído inserido. Além dos ruídos do modelo de velocidade, foram inseridos ruídos Gaussianos no sensor de leitura dos marcos, onde o valor de distância ao marco,  $r_t^i$ , foi adicionado um ruído Gaussiano  $N(0, \sigma_r)$ , com  $\sigma_r = 0.1m$ , e o ângulo entre o robô e o marco,  $\phi_t^i$ , foi adicionado um ruído  $N(0, \phi_r)$ , com  $\phi_r = 5 \text{ graus}$ .

Para cada um dos métodos aplicados foram executadas duas configurações a partir do mesmo vetor de dados de controle. Sendo que ambas as configurações apresentam uma matriz de covariância elevada, ou seja, alta incerteza quando a posição inicial, contudo, na primeira configuração, o valor médio da posição inicial está correto e na segunda está errado. A posição inicial real do robô é  $(-4, -2.5, 0^0)^T$  e sua estimativa inicial errada é  $(2, 0, 50^0)^T$ .

A seguir são apresentados os resultados dos experimentos.

#### 3.1. Resultado das simulações

Nas figuras abaixo serão exibidos os erros e resultados para a localização local e global do robô, que irá executar o mesmo trajeto, para os diferentes algoritmos apresentados na seção 2 e em seus respectivos mapas. Nas figuras 3, 5, 7 e 9, a linha vermelha é a estimativa do algoritmo de localização e a linha preta é a trajetória real do robô. Essas linhas foram geradas a partir da ligação dos pontos que representam a posição real e a estimativa a cada instante de tempo da simulação, com o intuito de facilitar a visualização dos dados da localização e a trajetória do robô. Nos gráficos das figuras 4, 6, 8 e 10, são apresentados os valores dos erros de posição  $(x, y)$  e orientação,  $\theta$ , do robô obtidos a partir diferença absoluta entre o valor real e o estimado pelos algoritmos de localização, a cada passo da simulação. Os valores de posição são apresentados em metro e o de orientação em radiano.

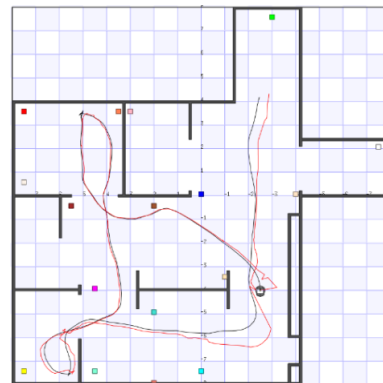


Figura 3. Trajeto EKF NC com posição inicial conhecida

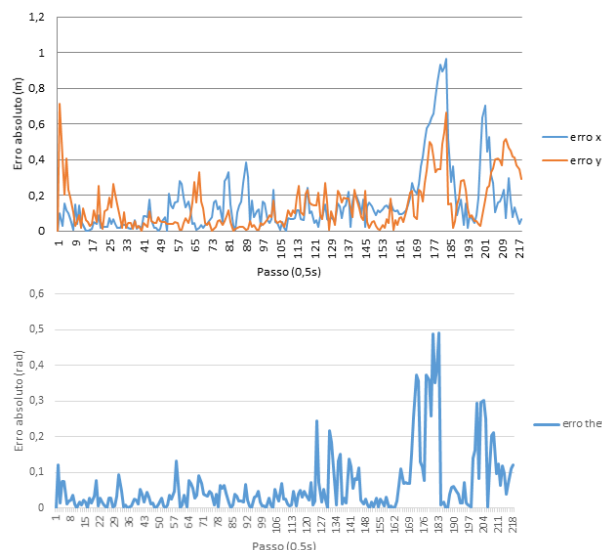


Figura 4. Erro absoluto de posição e orientação EKF NC com posição inicial conhecida

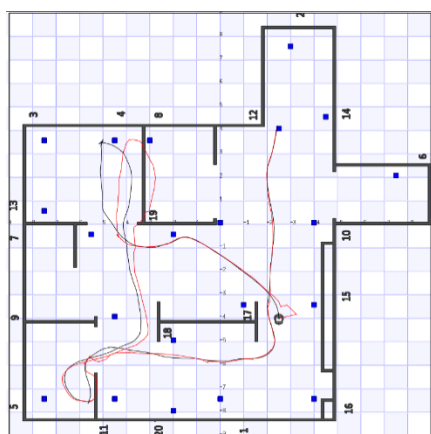


Figura 5. EKF UC com posição inicial conhecida

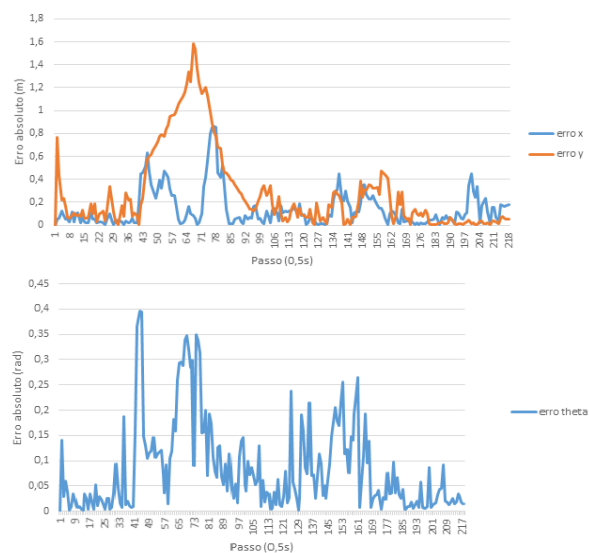


Figura 6. Erro absoluto de posição e orientação EKF UC com posição inicial conhecida

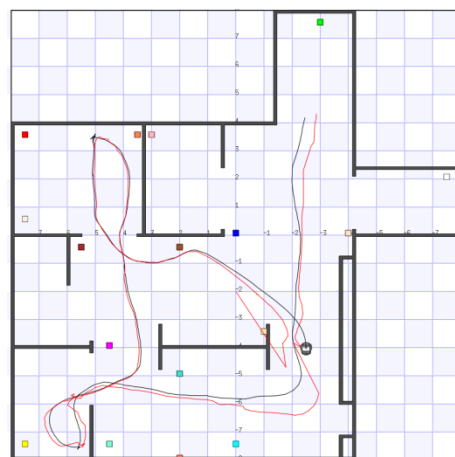


Figura 7. EKC NC com posição inicial desconhecida

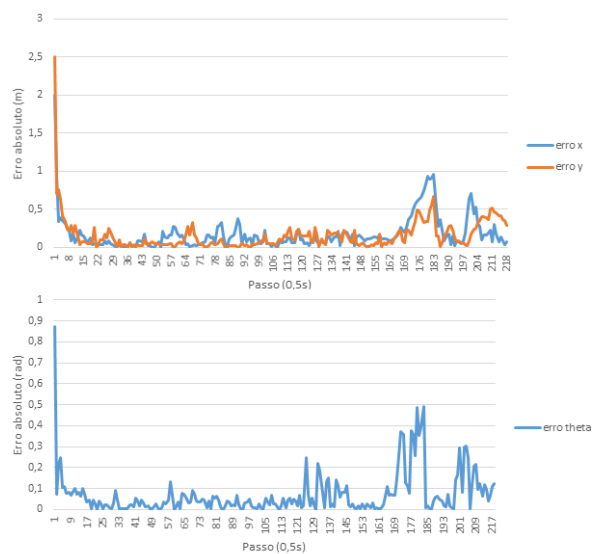


Figura 8. Erro absoluto de posição e orientação EKC NC com posição inicial desconhecida

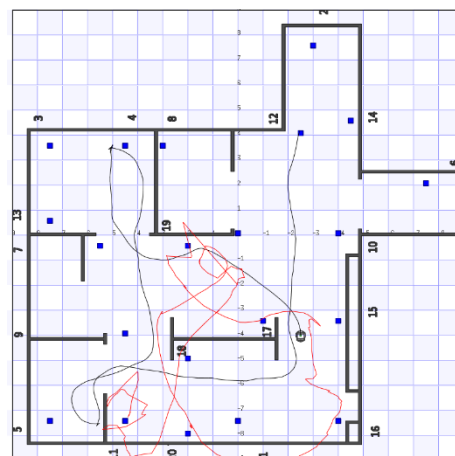


Figura 9. EKF UC com posição inicial desconhecida

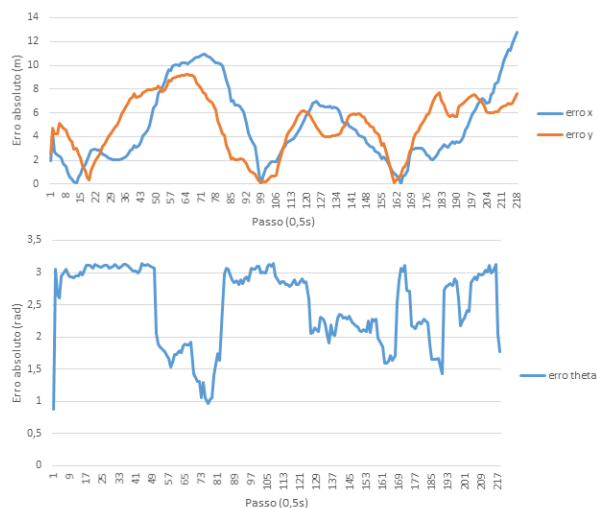


Figura 10. Erro absoluto de posição e orientação EKF UC com posição inicial desconhecida

### 3.3 Análise dos resultados

Ao observar os resultados dos métodos com localização local, ou seja, estimativa da posição inicial correta, ambos apresentaram excelentes resultados, sendo a região de maior erro do EKF NC, entre os passos 169 e 183, justificadas pela não observação de marcos nessa região. Já para o método EKF UC, o maior erro de localização ocorre entre os passos 43 e 85, esse erro pode ser justificado pela simetria do posicionamento dos marcos no mapa e por não ter sido aplicada uma rotina externa para exclusão de marcos cujos sensores não poderiam captar simultaneamente no momento de identificação pela máxima probabilidade.

Para o caso de localização global, ou seja, estimativa da posição inicial errada, o método de Kalman para correspondências conhecidas, manteve o excelente resultado sendo capaz de se localizar com facilidade. Já o método para correspondências desconhecidas, não conseguiu se localizar no trajeto especificado, esse resultado se justifica mais uma vez pela simetria entre as posições dos marcos dos mapas e por não ter sido elaborada uma rotina externa que eliminasse correspondências cujos sensores não poderiam captar na etapa de seleção do marco por máxima probabilidade. Outro fator, que influenciou a não localização do robô, é o fato do método com correspondência desconhecida necessitar de muito mais passos para convergir, pois, idealmente, deve-se observar todos os marcos e, se possível, mais de uma vez.

## 4 Conclusão

Com a realização desse trabalho foi possível observar e compreender melhor o problema de localização de robôs, assim como as limitações e diferenças dos algoritmos aplicados nesse estudo de caso.

Como proposto, foi possível observar a maior complexidade do problema de localização para mapas com a correspondência entre os marcos desconhecida,

principalmente para o caso de localização global. Onde, como esperado, são necessárias muito mais interações para se obter resultados próximos ao caso de correspondência conhecida, já que a incerteza presente nessa configuração é muito maior.

Para trabalhos futuros, pretende-se implementar os algoritmos apresentados nesse trabalho em uma plataforma real, cujo poder de processamento é mais limitado.

## Referências Bibliográficas

1. Dellaert, F. et al., 1999. Monte Carlo localization for mobile robots. *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, Volume 2, pp. 1322 - 1328.
2. Faisal, M. et al., 2014. Robot localization using extended kalman filter with infrared sensor. *Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on*, pp. 356 - 360.
3. Fox, D. B. W. a. T. S., 1999. *Markov localization for reliable robot*. s.l.:s.n.
4. Gerkey, B. & contributors, 2010. *The Player Robot Device Interface*. [Online] Available at: <http://playerstage.sourceforge.net/doc/Player-3.0.2/player/> [Acesso em 01 12 2014].
5. Prof. Dr. Wolfram Burgard, P. D. M. B. D. G. D. T. D. L. S., 2015. *Autonome Intelligente Systeme Albert-Ludwigs-Universität Freiburg*. [Online] Available at: <http://ais.informatik.uni-freiburg.de/teaching/ss14/robotics/> [Acesso em 01 12 2014].
6. Ruchti, P., Steder, B., Ruhnke, M. & Burgard, W., 2015. Localization on OpenStreetMap Data using a 3D Laser Scanner. *In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*.
7. Sanda, B., Electr. & Comput. Eng. Dept., W. M. U. K. M. U., Abdel-Qader, I. & Akanmu, A., 2014. Kalman filters for reducing error in RFID real-time localization systems. *Electro/Information Technology (EIT), 2014 IEEE International Conference on*, pp. 324 - 329.
8. Thrun, S., Burgard, W. & Fox, D., 2005. *Probabilistic Robotics*. Cambridge: The MIT Press.
9. Tuna, G., Bilgisayar Programcılığı Bolumu, T. U. E. T., Gungor, V. & Gulez, K., 2012. GPS aided Extended Kalman Filter based localization for unmanned vehicles. *Signal Processing and Communications Applications Conference (SIU), 2012 20th*, pp. 1-4.
10. Vaughan, R. & contributors, 2011. *The Stage Robot Simulator*. [Online] Available at: <http://rtv.github.io/Stage/> [Acesso em 01 12 2014].
11. Zhang, X. et al., 2006. Vision-based Monte Carlo - Kalman Localization in a Known Dynamic Environment. *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*, pp. 1 - 7.