

Auto-localização de uma cadeira de rodas motorizada utilizando fusão sensorial da odometria e do giroscópio

Self-location of a power wheelchair using sensory fusion of odometry and gyroscope

DOI:10.34117/bjdv8n9-091

Recebimento dos originais: 25/07/2022

Aceitação para publicação: 31/08/2022

Pablo France Salaroli

Graduação em Engenharia de Controle e Automação

Instituição: Instituto Federal do Espírito Santo (IFES) - Campus Serra

Endereço: Rodovia ES 010, Km 6,5, CEP: 29173-087, Manguinhos, Serra - ES, Brasil

E-mail: pablo_salaroli@hotmail.com

Vinicius da Rocha Motta

Mestrado em Engenharia de Controle e Automação

Instituição: Instituto Federal do Espírito Santo (IFES) - Campus Serra

Endereço: Rodovia ES 010, Km 6,5, CEP: 29173-087, Manguinhos, Serra - ES, Brasil

E-mail: vinicius.motta@ifes.edu.br

Rogério Passos do Amaral Pereira

Mestrado em Engenharia Elétrica

Instituição: Instituto Federal do Espírito Santo (IFES) - Campus Serra

Endereço: Rodovia ES 010, Km 6,5, CEP: 29173-087, Manguinhos, Serra - ES, Brasil

E-mail: rogeriop@ifes.edu.br

Gustavo Maia de Almeida

Doutorado em Engenharia Elétrica

Instituição: Instituto Federal do Espírito Santo (IFES) - Campus Serra

Endereço: Rodovia ES 010, Km 6,5, CEP: 29173-087, Manguinhos, Serra - ES, Brasil

E-mail: gmaia@ifes.edu.br

Marco Antonio de Souza Leite Cuadros

Doutorado em Engenharia Elétrica

Instituição: Instituto Federal do Espírito Santo (IFES) - Campus Serra

Endereço: Rodovia ES 010, Km 6,5, CEP: 29173-087, Manguinhos, Serra - ES, Brasil

E-mail: marcoantonio@ifes.edu.br

RESUMO

No quesito navegação de robôs móveis em ambientes internos, um dos maiores problemas encontrados é a obtenção correta da localização do robô. Dentre os diferentes métodos existentes, há poucos que são satisfatórios para assegurar uma navegação correta por acumular erros, como o caso da *dead reckoning* ou emprego de sensores inerciais, ou pela limitação e imprecisão quando se trata de ambientes internos como no caso do GPS. O presente trabalho traz a implementação da fusão dos dados da *dead reckoning* e giroscópio para estimar a orientação de um robô móvel, utilizando o Filtro de Kalman Estendido (EKF) em uma plataforma robótica

real, a fim de comparar o resultado da localização obtida com a fusão e com a *dead reckoning*. Para análise dos resultados, foi utilizado um sistema de localização por imagem, testado e validado, para fornecer a localização real do robô.

Palavras-chave: localização, robótica móvel, fusão de dados, filtro de kalman estendido, *dead reckoning*.

ABSTRACT

In all group of issues of robot navigation indoor environments, one of the main problems is to obtain the correct location of the mobiles robots. Among the several methods available, there are a few which are effective due to the accumulation of errors, such as the dead reckoning or the use of inertial sensors, or, still, for limitation and imprecision when it comes to internal environments such as GPS. The paper contains the implementation of the data fusion of the dead reckoning and gyroscope to estimate the orientation of a mobile robot, using the Extended Kalman Filter (EKF) in a real robot platform, in order to compare the result of the location obtained with fusion and with dead reckoning. For the analysis of the results, an image localization system, tested and validated, was used to provide the real location of the robot.

Keywords: localization, mobile robotics, data fusion, extended kalman filter, *dead reckoning*.

1 INTRODUÇÃO

Para navegação de robôs móveis, localização é um requisito básico (Chenavier and Crowley, 1992) assim como é um dos maiores problemas (Ganganath and Leung, 2012). Existem métodos que são soluções parciais para o problema da localização e estes podem ser grosseiramente classificados em dois grupos: medições relativas, que fornecem a localização do robô baseado em uma localização inicial, e.g. *dead reckoning* usando odometria por *encoders* e navegação inercial; e medições absolutas, que fornecem a localização global do robô, não necessitando de dados anteriores, e.g. GPS, bússola, navegação por *Landmarks*, etc. (Borenstein et al., 1997).

Dentre os diversos métodos existentes, a odometria baseada na leitura de *encoders* das rodas é o mais amplamente utilizado para localização de robôs móveis, pois fornece boa precisão a curto prazo, possui baixo custo financeiro e permite altas taxas de amostragem (Borenstein et al., 1997). Porém, a localização odométrica acumula erros com o tempo de forma ilimitada, devido ao deslizamento das rodas, à rugosidade do chão e à amostragem discretizada de incrementos das rodas (Chong and Kleeman, 1997).

Outro método é a navegação inercial, utilizando integração das medidas de sensores inerciais, como giroscópio e acelerômetro que, segundo (Barshan and Durrant-Whyte, 1995), podem fornecer informações valiosas da posição do robô. As vantagens

desse método são altas taxas de amostragem e, ao contrário de *encoders*, a medição não é afetada devido deslizamento das rodas. Entretanto, erros devido integração de bias e ruído são acumulados com o tempo e também ocorrem leituras imprecisas devido ao mau alinhamento dos eixos dos sensores com a estrutura de navegação (Sukkarieh et al., 1999).

Para obter uma localização mais precisa, é possível utilizar a fusão sensorial de múltiplos sensores. A técnica mais amplamente utilizada para fusão de dados de sensores é o Filtro de Kalman Estendido (EKF) (Rigatos, 2012), que é uma das variantes do filtro de Kalman (KF) para sistemas não lineares.

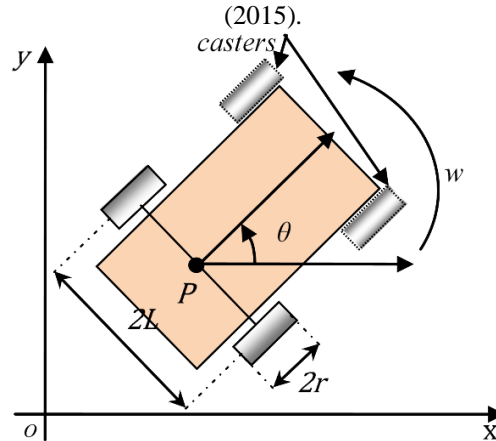
O objetivo deste trabalho é analisar a melhoria na localização obtida através da fusão dos dados de orientação do *dead reckoning* usando odometria por *encoders* e do giroscópio, utilizando o EKF, em relação à localização obtida puramente com odometria. Para isso, também é necessário obter a localização real do robô no ambiente de testes, de forma a ser referência para comparar os resultados da odometria e da fusão. A localização real foi obtida através de um sistema de processamento de imagem testado e validado.

O trabalho está organizado da seguinte forma: na seção 2 é descrito o modelo cinemático do robô móvel diferencial. Na seção 3 é descrita a plataforma robótica utilizada. Na seção 4 é descrito o método de localização por imagem. Na seção 5 é descrito o algoritmo de fusão de dados. Na seção 6 são apresentados os resultados. Na seção 7 conclui-se sobre os resultados.

2 MODELO CINEMÁTICO DO ROBÔ MÓVEL DIFERENCIAL

A configuração do robô móvel utilizado neste trabalho é o robô não holonômico de tração diferencial, como mostrado na Figura 1. Ele possui duas rodas motoras paralelas de tração independentes uma da outra e duas rodas livres (roda boba ou *caster*). As duas rodas motoras são idênticas, com raio r sendo que a distância entre elas é $2L$. A postura do robô é definida como o vetor $q = [x, y, \theta]^T$ cujos parâmetros (x, y) são coordenadas do ponto P que definem a localização do robô no eixo cartesiano e θ determina o ângulo de orientação do robô.

Figura 1. Configuração do robô móvel não-holonômico de tração diferencial. Adaptado de Tommasi et al. (2015).



A Equação (1) representa o modelo cinemático do robô móvel. O modelo é utilizado na odometria e também no algoritmo de fusão.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \quad (1)$$

onde v e w são as velocidades linear e angular do robô, respectivamente.

3 PLATAFORMA ROBÓTICA

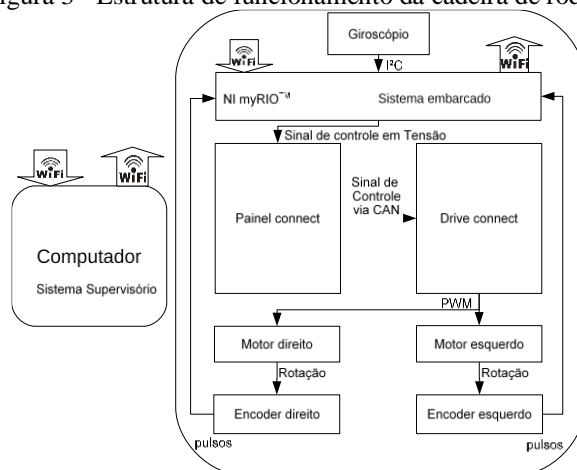
O robô utilizado é uma modificação de uma cadeira de rodas motorizada, através da instalação de dois *encoders* de 5000 pulsos por volta e de um sistema embarcado NI myRIO®, como pode ser visto na Figura 2. Como os *encoders* foram instalados na correia de tração das rodas, houve uma alteração na relação de pulsos do *encoder* por volta das rodas do robô, ultrapassando a faixa dos 50 mil pulsos por volta. O algoritmo de leitura dos pulsos dos *encoders* está implementado em um FPGA, de funcionamento independente do processador principal, contidos no NI myRIO®. Este FPGA possui entradas para *encoders* que podem realizar leituras de até 100 kHz, o que garante a leitura dos pulsos para as velocidades de utilização da cadeira de rodas. As rodas motoras da cadeira possuem um raio $r = 0,245 \text{ m}$, distantes entre si de $L = 0,6 \text{ m}$.

Figura 2. Plataforma robótica utilizada.



Também pode ser observado na Figura 2 que foram realizadas alterações na estrutura. Tais alterações são temporárias, a fim de facilitar a instalação de um padrão de reconhecimento, utilizado no sistema de localização por imagem apresentado na Seção 4. A Figura 3 mostra a estrutura de funcionamento da cadeira controlada através do NI myRIO®.

Figura 3 - Estrutura de funcionamento da cadeira de rodas.



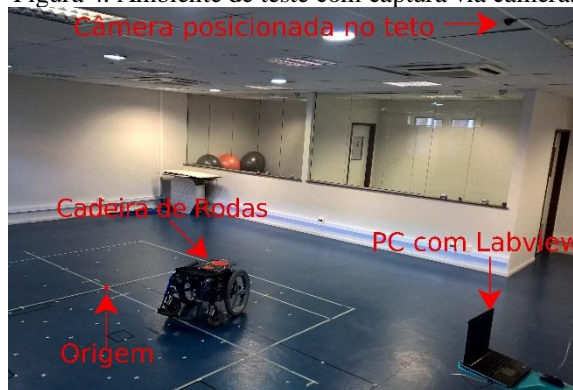
Conforme pode ser observado na Figura 3, o controle de velocidade dos motores se dá através de sinal analógico de tensão e a leitura do giroscópio é realizada através do protocolo I²C.

Todo o processamento é realizado de forma embarcada. O software executado no NI myRIO® foi desenvolvido no LabVIEW® e atualmente conta com algoritmos de geração de trajetória, controle de trajetória e localização. Para este trabalho, o controlador de trajetória utilizado foi o *backstepping*, controlador baseado nos trabalhos de (Fierro and Lewis, 1998) e (Ye, 2008). O controlador *backstepping* foi o escolhido por ser recomendado nos trabalhos de (Fierro and Lewis, 1998), (Jiang and Nijmeijer, 1997) e (Kukao et al., 2000) e por obter o melhor desempenho segundo Tommasi et. al. (2015).

4 LOCALIZAÇÃO POR IMAGENS

Para obter a postura real do robô foi utilizado um sistema de localização por imagem com o uso de uma câmera e um padrão de reconhecimento, que tem dois círculos de diferentes diâmetros e uma cor específica que os destaca no ambiente de testes, conforme o trabalho de Pimentel (2015). O ambiente de teste pode ser visto na Figura 4. A área válida para a captura com imagens tem dimensões 2,5 m no eixo x e 3,5 m no eixo y .

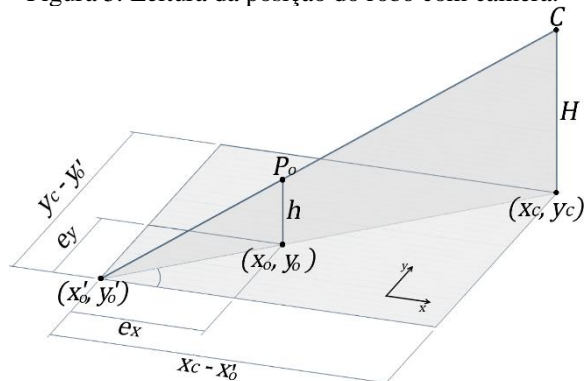
Figura 4. Ambiente de teste com captura via câmera.



O padrão de reconhecimento foi posicionado sobre o robô, com o círculo menor localizado à frente. Para relacionar os pixels com as dimensões reais, é necessário inicialmente realizar a calibração, isto é feito tomando 54 pontos demarcados no chão da área de teste com coordenadas conhecidas. Com isso, o programa calcula a posição do centroide dos círculos pequeno e grande, com coordenadas (x'_p, y'_p) e (x'_g, y'_g) , respectivamente.

As coordenadas obtidas com o processamento descrito acima não descrevem a real posição dos círculos devido um efeito de projeção, como pode ser visto na Figura 5.

Figura 5. Leitura da posição do robô com câmera.



Observando a Figura 5 e abstraindo para a medição da posição de um objeto qualquer localizado em P_o com uma altura h nas coordenadas (x_o, y_o) , onde (x_c, y_c) são as coordenadas da câmera localizada em C com uma altura H , (x'_o, y'_o) são as coordenadas do objeto lidas através do processamento da imagem, e_x e e_y são os erros da medição nas coordenadas x e y , respectivamente. As variáveis da Figura 5 tem como referência a origem definida na calibração.

Utilizando a geometria da Figura 5, foram obtidas as Equações (2) e (3) de forma a corrigir o erro na medição e obter a coordenada real do objeto (x_o, y_o) .

$$\begin{cases} x_o = x'_o - e_x \\ y_o = y'_o - e_y \end{cases} \quad (2)$$

$$\begin{cases} e_x = \frac{h}{H} \sqrt{(x_c - x'_o)^2 + (y_c - y'_o)^2} \cos \left[\tan^{-1} \left(\frac{y_c - y'_o}{x_c - x'_o} \right) \right] \\ e_y = \frac{h}{H} \sqrt{(x_c - x'_o)^2 + (y_c - y'_o)^2} \sin \left[\tan^{-1} \left(\frac{y_c - y'_o}{x_c - x'_o} \right) \right] \end{cases} \quad (3)$$

As Equações (2) e (3) foram aplicadas para (x'_p, y'_p) e (x'_g, y'_g) , obtendo (x_p, y_p) e (x_g, y_g) , que são a posição real do centroide dos círculos pequeno e grande, respectivamente. As coordenadas do centroide do robô (x_r, y_r) juntamente com sua orientação θ_r formam a postura do robô, que pode ser calculada conforme a Equação (4).

$$\begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} (x_p + x_g)/2 \\ (y_p + y_g)/2 \\ \text{atan2}((y_p - y_g), (x_p - x_g)) \end{bmatrix} \quad (4)$$

Para verificar a validade do método utilizado, o robô foi posicionado manualmente em 28 pontos distintos com orientações diferentes e os erros absolutos máximos e médios da medição por imagem estão presentes na Tabela 1. D é a distância entre o ponto corrigido e o ponto real.

Tabela 1. Erros na medição da pose por imagem.

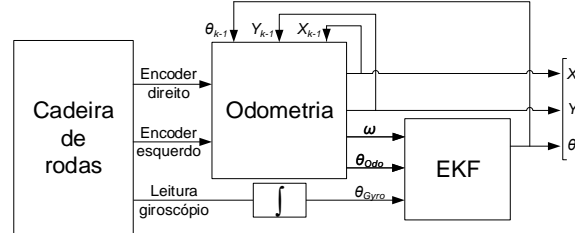
Variável	Erro médio absoluto	Erro máximo absoluto
x	0,4 (10^{-2} m)	1,2 (10^{-2} m)
y	0,5 (10^{-2} m)	1,6 (10^{-2} m)
θ	0,01 (rad)	0,04 (rad)
D	0,7 (10^{-2} m)	1,9 (10^{-2} m)

Pode ser observado na Tabela 1 que os erros nas variáveis x , y , θ e D são muito pequenos frente às dimensões do robô e às dimensões da área de testes, podendo assim ser desprezados. Esta observação permite assumir que a postura lida com o processamento de imagens é a postura real do robô.

5 FUSÃO DE DADOS

Das três variáveis da postura do robô, apenas θ foi fusionada, conforme o diagrama da Figura 6. A orientação foi escolhida por ser uma variável crítica, pois é usada diretamente nos cálculos das variáveis x e y , como visto na Equação (1).

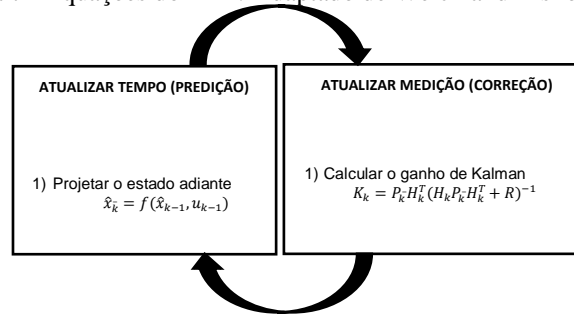
Figura 6. Diagrama da fusão de dados para θ .



Como o modelo que representa o comportamento robô móvel é não linear, como apresentado na Equação (1), devemos utilizar uma ferramenta adequada. Embora a fusão seja realizada apenas na variável θ , única da postura que tem comportamento linear (Equação (1)), decidiu-se utilizar o filtro de Kalman estendido para facilitar a implementação de trabalhos futuros.

O EKF é o filtro de Kalman linearizado sobre a média e covariância atuais para poder estimar estados de processos e/ou relação entre estados e saída não-lineares (Welch and Bishop, 1995). O EKF consiste nas equações da Figura 7:

Figura 7 - Equações do EKF. Adaptado de Welch and Bishop (1995)



Boa parte das variáveis das equações da Figura 7 são iguais às do filtro de Kalman discreto: x é o vetor de estados, u é o vetor de entrada, z é o vetor de medições, P é a matriz de covariância dos erros da estimativa, K é a matriz de ganhos de Kalman, Q e R são as matrizes de covariância dos ruídos de processo e medição, respectivamente, elas são utilizadas como parâmetros de sintonia do EKF. O sobrescrito ‘-’ sinaliza um valor a priori, já o sobrescrito ‘^’ indica que aquela variável é estimada. As funções f e h bem como as matrizes F e H serão apresentadas adiante. Para este trabalho, x e z foram definidos conforme as Equações (5) e (6).

$$\hat{x}_k = \hat{\theta}_k = \hat{\theta}_{Fusão_k} \quad (5)$$

$$z_k = [\theta_{odometria_k} \quad \theta_{Giroscópio_k}]^T \quad (6)$$

As funções f (Equação (7)) e h (Equação (8)) representam o modelo discretizado do estado e das medições, respectivamente. As matrizes F e H são as jacobianas que linearizam o modelo e as medições, respectivamente, conforme as Equações (9) e (10).

$$f(\hat{\theta}_{k-1}, w_k) = \hat{\theta}_k^- = \hat{\theta}_{k-1} + w_k T \quad (7)$$

$$h(\hat{x}_k^-) = [\hat{x}_k^- \quad \hat{x}_k^-]^T = [\hat{\theta}_k^- \quad \hat{\theta}_k^-]^T \quad (8)$$

$$F_k = J_f(x_{k-1}) = \frac{\partial f(\theta_{k-1}, w_k)}{\partial \theta_{k-1}} = 1 \quad (9)$$

$$H_k = J_h(x_{k-1}) = \frac{\partial h(\theta_k)}{\partial \theta_k} = [1 \quad 1]^T \quad (10)$$

6 RESULTADOS

Para os testes realizados, o EKF foi sintonizado conforme as Equações (11) e (12), onde σ_P , σ_O , σ_G são os desvios padrão do processo, da odometria e do giroscópio, respectivamente. Estes valores foram encontrados por tentativa e erro.

$$Q = [\sigma_{\hat{p}}^2] = [0,17^2] \quad (11)$$

$$R = \begin{bmatrix} \sigma_o^2 & 0 \\ 0 & \sigma_g^2 \end{bmatrix} = \begin{bmatrix} 0,55^2 & 0 \\ 0 & 0,55^2 \end{bmatrix} \quad (12)$$

Os testes realizados utilizaram duas trajetórias: o círculo da Equação (13) (velocidades linear e angular constantes), e a lemniscata da Equação (14) (velocidades linear e angular variáveis). Para ambas trajetórias, a postura inicial do robô foi $q = [0 \ 0 \ 0]^T$.

$$\begin{cases} X_r = 1,1\text{sen}(0,0118t) \text{ [m]} \\ Y_r = 1,1\text{cos}(0,0118t) - 0,75 \text{ [m]} \end{cases} \quad (13)$$

$$\begin{cases} X_r = -1,4\text{cos}(0,005t) \text{ [m]} \\ Y_r = 1,1\text{sen}(0,01t) - 0,75 \text{ [m]} \end{cases} \quad (14)$$

Foram calculados os erros da orientação (θ) da fusão, da odometria e do giroscópio bem como os erros absolutos da posição da fusão e da odometria, todos em relação à postura real do robô. Os resultados são mostrados em forma de gráficos de erros absolutos e tabelas de erros médio absoluto e máximo absoluto.

Na trajetória circular foram completadas 5 voltas, percorrendo uma distância total de 35,4 metros em um tempo de 141,6 segundos. A Figura 8 mostra o módulo dos erros da orientação obtida com a fusão, com o giroscópio e com a odometria no teste. A Tabela 2 contém os valores médio e máximo dos erros apresentados na Figura 8. A Figura 9 mostra o módulo dos erros na posição da fusão e da odometria. Os valores médio e máximo dos erros apresentados na Figura 9 estão contidos na Tabela 3.

Figura 8. Erros absolutos da orientação na a trajetória circular.

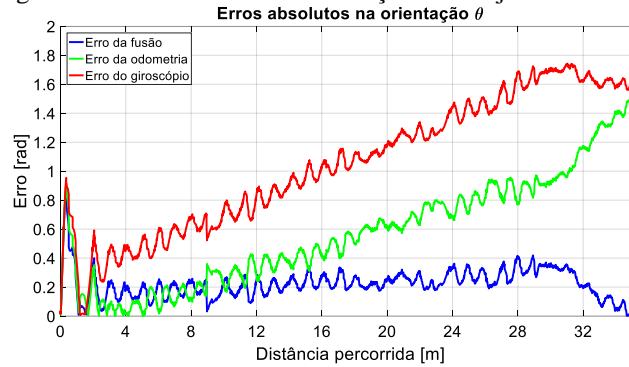


Tabela 2. Erros absolutos da orientação na trajetória circular.

Variável	Erro médio absoluto (rad)	Erro máximo absoluto (rad)
$\theta_{Fusão}$	0,22	0,87
$\theta_{Giroscópio}$	1,03	1,81
$\theta_{Odometria}$	0,56	1,49

Figura 9. Erros absolutos da posição na a trajetória circular.

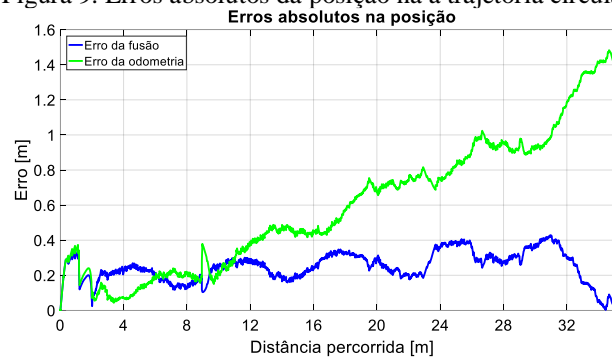


Tabela 3. Erros absolutos da posição na trajetória circular.

Variável	Erro médio absoluto (m)	Erro máximo absoluto (m)
$D_{Fusão}$	0,24	0,43
$D_{Odometria}$	0,58	1,48

Na trajetória lemniscata foram completadas 3 voltas, percorrendo uma distância total de 36,6 metros em um tempo de 204,7 segundos. A Figura 10 mostra o módulo dos erros da orientação obtida com a fusão, com o giroscópio e com a odometria no teste. A Tabela 4 contém os valores médio e máximo dos erros apresentados na Figura 10. A Figura 11 mostra o módulo dos erros na posição da fusão e da odometria. Os valores médio e máximo dos erros apresentados na Figura 11 estão contidos na Tabela 5.

Figura 10. Erros absolutos da orientação na trajetória lemniscata.

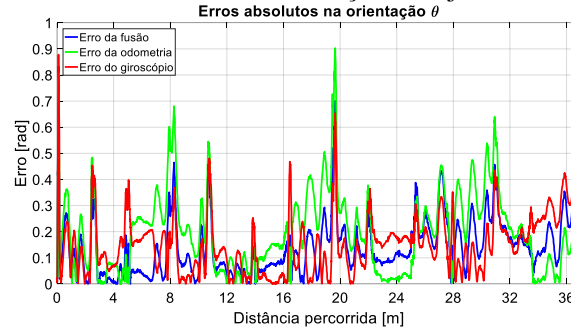


Tabela 4. Erros absolutos da orientação na trajetória lemniscata.

Variável	Erro médio absoluto (rad)	Erro máximo absoluto (rad)
$\theta_{Fusão}$	0,15	0,78
$\theta_{Giroscópio}$	0,17	0,88
$\theta_{Odometria}$	0,21	0,90

Figura 11. Erros absolutos da posição na trajetória lemniscata.



Tabela 5. Erros absolutos da posição na trajetória lemniscata.

Variável	Erro médio absoluto (m)	Erro máximo absoluto (m)
$D_{Fusão}$	0,20	0,50
$D_{Odometria}$	0,85	1,88

A Figura 12 mostra a localização na primeira volta da trajetória lemniscata obtida com a fusão, odometria e a localização real. No caso da trajetória circular, para uma melhor visualização dos erros, foi necessário apresentar a localização em cinco pontos distintos ao longo de todo o teste, pois o erro é acumulado sobre a trajetória, como pode ser visto na Figura 13.

Figura 12 - Localizações na primeira volta da trajetória lemniscata.

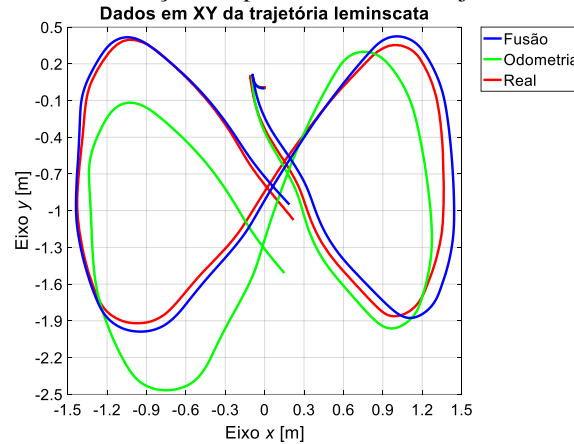
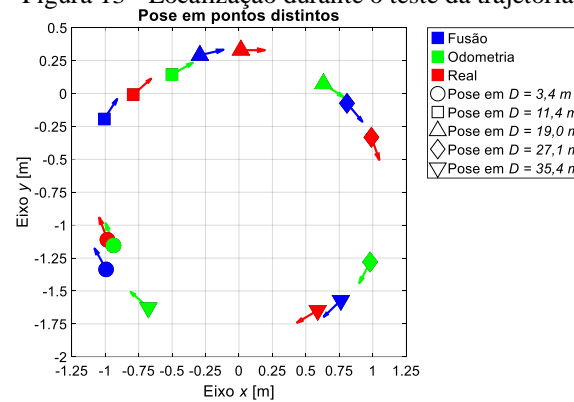


Figura 13 - Localização durante o teste da trajetória circular.



7 CONCLUSÃO E TRABALHOS FUTUROS

Observando os resultados apresentados nos gráficos e tabelas da seção anterior, pode-se verificar que a orientação obtida através da fusão sensorial possui um erro significativamente menor se comparado ao erro da odometria e do giroscópio. Além disso, foi verificado que a diminuição do erro na orientação do robô influenciou diretamente em sua localização, diminuindo o erro de forma muito significativa nas duas trajetórias testadas. É possível observar que há um grande acúmulo de erros utilizando a odometria e um acúmulo pequeno no caso da fusão. Sendo assim, verifica-se que a utilização da fusão sensorial entre a odometria e giroscópio para a orientação resulta em uma localização mais precisa que a obtida apenas com a odometria.

Uma possível melhoria seria adicionar à fusão realizada dados de outro sensor (e.g. bússola). Outra melhoria relevante seria a adição de um sistema de localização absoluto, para corrigir a localização com certa frequência (e.g. *landmarks*) ou utilizar mapas juntamente com filtro de partículas.

Nos trabalhos de Ganganath e Leung (2012), Houshangi e Azizi (2006) e de Lategahn et al. (2010), os erros da localização obtida com as técnicas de fusão são notoriamente menores quando comparados com a odometria isolada. Além disso, nos trabalhos de Houshangi e Azizi (2006) e Lategahn et al. (2010), que utilizaram o giroscópio, observa-se que o resultado obtido fusionando apenas a odometria com o giroscópio representa uma melhoria significativa em relação à odometria.

Comparando os resultados disponíveis na literatura com os resultados obtidos neste trabalho, pode-se concluir que é possível obter um melhor sistema de localização utilizando apenas um giroscópio fusionado com a odometria e algumas modificações no software.

AGRADECIMENTOS

À Cooperação CAPES/FAPES - Programa de Desenvolvimento da Pós-Graduação - PDPG, através do projeto “TIC+TAC: Tecnologia da Informação e Comunicação + Tecnologia de Automação e Controle, As Tecnologias Inteligentes, Prioritárias,” pelo apoio financeiro da pesquisa, por meio do Edital FA-PES/CNPq N° 23/2018 – PRONEM (Termo de Outorga 133/2021 e Processo N° 2021-CFT5C).

REFERÊNCIAS

- Barshan, B. and Durrant-Whyte, H. (1995). Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 3, pp. 328-342.
- Borenstein, J.; Everett, H. R.; Feng, L. and Wehe, D. (1997). Mobile robot positioning: Sensors and techniques, *Journal of Robotic Systems*, Vol. 14, No. 4, pp. 231-249.
- Chenavier, F. and Crowley, J. L. (1992). Position estimation for a mobile robot using vision and odometry. *Proceedings - IEEE International Conference on Robotics and Automation*, 1992, Vol. 3, pp. 2588-2593.
- Chong, K. and Kleeman, L. (1997). Accurate odometry and error modelling for a mobile robot. *Proceedings - IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 2783-2788.
- Fierro, R. and Lewis, F.L. (1998). Control of a nonholonomic mobile robot using neural networks, *IEEE Transactions on Neural Networks*, Vol. 9, No. 4, pp. 589-600.
- Ganganath, N. and Leung, H. (2012). Mobile robot localization using odometry and kinect sensor. *2012 IEEE International Conference on Emerging Signal Processing Applications (ESPA)*, pp. 91-94.
- Houshangi, N.; Azizi, F. (2006) Mobile robot position determination using data integration of odometry and gyroscope. *IEEE 2006 World Automation Congress*, pp. 1-8.
- Jiang, Z. P. and Nijmeijer, H. (1997). Tracking control of mobile robots: a case study in backstepping. *Automatica*, Vol. 33, No. 7, pp. 1393-1399.
- Kukao, T.; Nakagawa, H. and Adachi, N. (2000). Adaptive tracking control of a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 5, pp. 609-615.
- Lategahn, J., Kuenemund, F., and Roehrig, C. (2010). Mobile robot localization using wlan, odometry and gyroscope data. *International Journal of Computing*, Vol. 9, No. 1, pp. 22-30.
- Pimentel, L. S. S.; Cuadros, M. A. S.; Amaral, R. P. and Gamarra, D. F. T. (2015). Development of a mobile robotics platform for navigation tasks using image processing. In *Computer Science and Applications: Proceedings of the 2014 Asia-Pacific Conference on Computer Science and Applications (CSAC 2014)*, pp. 457-463.
- Rigatos, G. (2012). Nonlinear Kalman Filters and Particle Filters for integrated navigation of unmanned aerial vehicles. *Robotics and Autonomous Systems*, Vol. 60, No. 7, pp. 978-995.
- Sukkarieh, S.; Nebot, E. and Durrant-Whyte, H. (1999). A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 3, pp. 572-578.
- Tommasi, E. E. V; Faria, H. G.; Cuadros, M. A. S.; Almeida, M. A.; Resende, C. Z. and Gamarra, D. F. T. (2015). Estudo comparativo de controladores de seguimento de trajetória para robôs de tração diferencial: *fuzzy*, ganhos fixos e *backstepping*. In: *Simpósio Brasileiro de Automação Inteligente*.

Welch, G. and Bishop, G. (1995). An introduction to the Kalman filter. Department of Computer Science, University of North Carolina. [online] Available at: <http://byron.soe.ucsc.edu/projects/SeaSlug/Documents/Publications/Kalman%20Filtering/An%20Introduction%20to%20the%20Kalman%20Fiter2.pdf> [Accessed 12 Jan. 2017].

Ye J. (2008). Tracking control for nonholonomic mobile robots: integrating the analog neural network into the backstepping technique, *Neurocomputing*, Vol. 71, No. 16, pp. 3373-3378.