



ANÁLISE DO ERRO ASSOCIADO DA FUSÃO DE SENSORES INERCIAIS E GPS EM AMBIENTE SIMULADO PARA VALORES DIFERENTES DE COVARIÂNCIA PARA MEDIÇÃO

Lucas Marins Batista ¹; Valéria Loureiro da Silva ²;

¹ SENAI CIMATEC; Salvador/Bahia; lucasmarbatista@gmail.com

² SENAI CIMATEC; Salvador/Bahia

Resumo: A aplicação de sensores cada vez mais precisos em sistema de localização autônoma é crescente em diversas áreas, podendo utilizar sensores inerciais como odometria, acelerômetros, giroscópios e GPS, porém cada uma tem uma incerteza associada na sua medição. Dessa forma, o objetivo desse artigo é utilizar o algoritmo de fusão de sensores, filtro de Kalman estendido, como um método para reduzir o erro na medição, configurando corretamente uma matriz de covariância do sistema. Como resultado, foi possível avaliar que quanto maiores os valores de covariância, mais rápido é possível obter uma resposta mais acurada caso alguns dos sensores perca sua importância, utilizando os dados obtidos e o modelo inercial apresentado.

Palavras-Chave: Localização; ROS; Filtro de Kalman; GPS; Odometria.

GPS AND INERTIAL SENSOR FUSION ERROR ANALYSIS FOR DIFFERENT VALUES OF COVARIANCE IN A SIMULATED ENVIRONMENT

Abstract: The application of accurate inertial sensors for autonomous localization is increasing in several areas, such as wheels odometry, accelerometers, gyros and GPS, each one with their uncertainty in measurement. Thus, the objective of this paper is to use the sensor fusion algorithm, extended Kalman filter, as a method to reduce the measurement error configuring a system covariance matrix. As a result, it was possible to evaluate that the higher the covariance values, the faster it is possible to obtain a more accurate response if the sensors become less important and a higher error, comparing the obtained data and the presented inertial model.

Keywords: Localization; ROS; Kalman Filter; GPS; Odometry.



1. INTRODUÇÃO

Navegação e localização de veículos autônomos são alguns dos maiores desafios na robótica, sendo bastante dependente da quantidade e qualidade dos sensores que podem ser utilizados para percepção e definição do movimento, como por exemplo, controle de rodas de um veículo móvel. O sistema de localização global (GPS) é, atualmente, o principal sensor de medição de posição absoluta utilizado para localização em ambientes, porém, nem sempre é possível usar ou adquirir um sensor com precisão suficiente para atingir determinado requisito. Dessa forma, o uso de sensores inerciais relativos, que identificam a posição do robô através da medição de outras grandezas como velocidade, orientação e aceleração, é necessário. [1]

Esse artigo tem o objetivo de fazer uma análise do funcionamento do pacote *robot_localization*, do *framework* de robótica ROS (*Robot Operating System*), com dados obtidos de sensores inerciais e GPS, e aplicação do filtro de Kalman Estendido com foco na configuração de suas matrizes de covariância e correlação [2]. Utilizando essa abordagem, os valores da matriz de covariância dos sensores e de processos do algoritmo serão variados com o objetivo de analisar a influência da configuração correta para o sistema localização.

Na simulação, o robô fará um percurso retilíneo, até o momento que o veículo será impedido de se movimentar por um objeto. As rodas manterão seu movimento, dando a falsa impressão do movimento, mesmo com o veículo parado gerando um erro crescente na odometria das rodas. Dessa forma, sensores adicionais de aceleração, orientação e de geolocalização fusionados, com a sua incerteza associada, precisam identificar esse erro de forma rápida e eficiente. [3]

1.1. Fusão de Sensores

Sensores desempenham um papel fundamental na determinação da localização automática de um determinado veículo, cada um com a suas incertezas e ruídos associados. Logo, é possível estimar a posição e orientação do robô através da odometria das rodas, com medidores que identificam o seu giro e velocidade, e sensores IMU (*Inertial Measurement Unit*) ou Unidade de Medição Inercial, com acelerômetros (aceleração) e giroscópios (orientação). Em sequência, utilizar algoritmos de fusão de sensores para reduzir o ruído com as informações obtidas para estimar um valor ótimo de localização [3].

O algoritmo clássico para fusão é o filtro de Kalman estendido que utiliza dados com incerteza associada de vários sensores para estimar o valor de posição como variável de saída, a partir da previsão, comparação e atualização das variáveis das equações de um sistema inercial conhecido. Sendo assim, valores de posição e velocidade de um sensor com ruído intrínseco, são comparados com um modelo e outros dados de aceleração, velocidade e posição vindos de um segundo ou terceiro sensor, obtendo uma informação mais precisa e acurada [4].



1.2. Filtro de Kalman

Para encontrar um valor estimado de posição e orientação a partir de determinadas variáveis inerciais, um modelo do sistema de equações pode ser definido a partir de vetores e matrizes que se relacionam, como demonstrado nas equações (1) e (2). Sendo, x_k o vetor de estados atual, x_{k-1} o vetor de estados anterior, A matriz de constantes para dimensionar estados, a partir do estado anterior x_{k-1} , u_k vetor de comandos de controle, B matriz de constantes para dimensionar estados, a partir do comando de controle u_k , w_k vetor de ruído de estados, z_k vetor com valores observados da medição, v_k vetor de ruído de medição e C matriz de constantes para dimensionar os valores medidos, a partir do estado atual x_k .

$$x_k = Ax_{k-1} + Bu_k + w_k \quad (1)$$

$$z_k = Cx_k + v_k \quad (2)$$

A partir desse modelo e suas covariâncias, é possível medir e prever valores estimados dos estados x_k iterativamente, seguindo o fluxograma da figura 1. A saída estimada depende de um ganho de Kalman G_k , que identifica qual sensor é mais confiável, baseado covariância de medição R de cada sensor representado na equação 4 e P_k , como uma matriz de covariância intrínseco do processo. R é a matriz essencial na definição dos pesos de importância de cada sensor para definição da matriz de ganho de Kalman G_k , equação 5, sendo o foco das simulações avaliadas [5].

$$z_k = \begin{bmatrix} \textit{distância} \\ \textit{velocidade} \\ \textit{aceleração} \\ \textit{orientação} \end{bmatrix} \quad (3)$$

$$R = \begin{bmatrix} \partial_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \partial_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \partial_z^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \partial_{velx}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \partial_{vely}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \partial_{velz}^2 \end{bmatrix} \quad (4)$$

$$G_k = P_k C^T (C P_k C^T + R)^{-1} \quad (5)$$

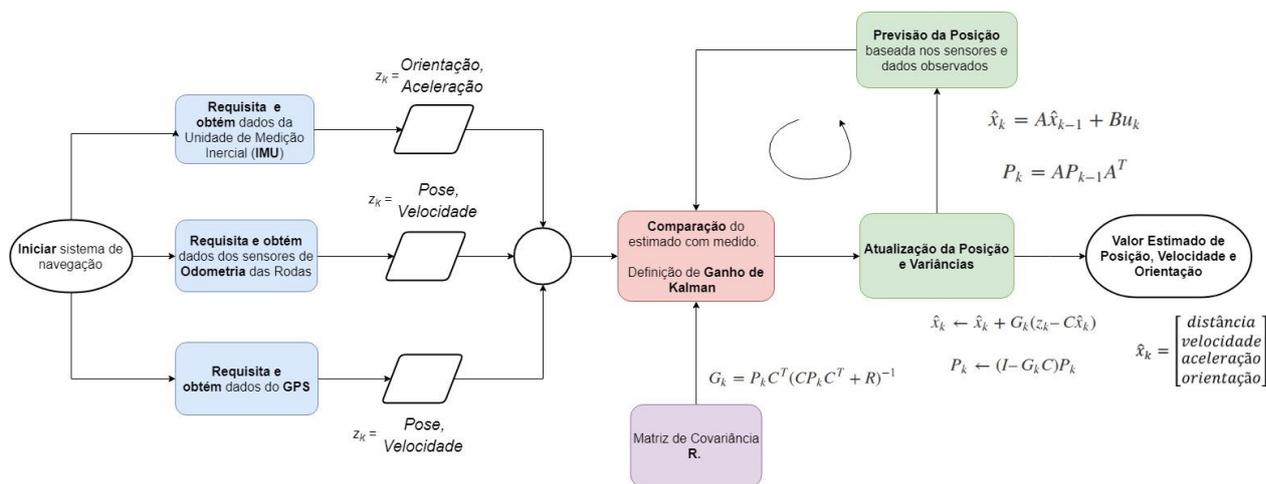


1.3. Diagramas

Todo o sistema de localização pode ser representado por quatro etapas, obtenção de dados, comparação, atualização e previsão através do modelo das variáveis de posição, orientação, velocidade, aceleração e uma matriz de covariâncias de importâncias. Na nossa representação, o pacote *robot_localization*, utiliza os dados inerciais obtidos de odometria das rodas, geolocalização de um GPS, como posição nos eixos X e Y e velocidade linear, e uma unidade de medição inercial (IMU), como aceleração e orientação simuladas. Dessa forma, a informação será fundida pelo sistema de navegação, utilizando como base o sistema modelado e a matriz de covariância de importância R, que representa o ruído médio de saída de todos os sensores, com o objetivo de encontrar valores de localização estimados que contenha uma maior precisão e acurácia, reduzindo o erro [6].

O diagrama a seguir representa o fluxo de dados e algumas relações que o sistema de navegação deve ter com a comunicação dos sensores antes de ser enviado para o algoritmo e filtro de Kalman, como pode ser verificado na Figura 1. O sistema de navegação precisa requisitar os dados a três sensores e enviar essas informações até uma rotina do filtro de Kalman. Os dados serão obtidos pelo sistema de navegação, esses são comparados com os resultados de um modelo inicial linearizado, baseado na física do movimento e os ruídos associados, determinado por matrizes de covariâncias. Após isso, novos valores serão obtidos através de uma função de atualização, novamente previstos, e utilizados para nova comparação e estimação.

Figura 1. Fluxograma do processo de obtenção de dados do sistema de navegação e rotina para obtenção de dados do Filtro de Kalman. Fonte: Própria





2. METODOLOGIA

O objetivo desse artigo é analisar a influência da configuração de matrizes de covariância dos sensores, através do pacote para localização e fusão de sensores do framework de robótica ROS (*Robot Operating system*), utilizando como ambiente simulação 3D, *Gazebo*, com dados de sensores disponibilizados de forma otimizada. Para realizar essa análise, testes simulados foram feitos, utilizando os sensores inerciais como IMU, com acelerômetros e giroscópios de 3 eixos, sinal de GPS (*Global Positioning System*) e odometria das rodas do robô diferencial Husky, veículo robótico terrestre especializado em realizar missões em ambientes abertos e íngremes [7].

O veículo robótico diferencial *Husky* simulado no *Gazebo*, realizará um movimento em linha reta durante 20m, saindo do centro do mapa até as bordas, onde deve encontrar uma barreira pré-colocada, como representado na Figura 2. Dessa forma, os dados dos três serão obtidos e fusionados, obtendo um valor de erro de localização no eixo X (posição obtida menos *ground truth*) e apresentados em gráficos das Figuras 3, 4 e 5.

Figura 2. Ambiente de simulação com o veículo robótico terrestre simulado no momento do impacto. Fonte: Simulação *Gazebo*



Em todos os experimentos, há um escorregamento na odometria das rodas, causando um erro sistemático crescente, proveniente da estrutura do movimento das rodas e da modelagem apresentada. Esse erro pode ser menor caso uma modelagem correta for aplicada. Porém, para o erro ser visível de forma gráfica em um movimento a uma pequena distância, foi necessária uma ordem de grandeza maior.

3. RESULTADOS E DISCUSSÃO

Nos seguintes experimentos, a linha vermelha representada pelo resultado obtido pelo filtro de Kalman estendido segue os valores do GPS com uma pequena variação devido a fusão de dados de odometria e da IMU. No entanto, em determinado momento, o choque ocorre com a parede e o erro da odometria cresce exponencialmente e a partir desse momento deve ser desconsiderado.



No primeiro experimento da Figura 3, ao utilizar valor de variância do modelo, σ_x^2 , igual a 0,2, os resultados atualizados, após o choque, da fusão de sensores se mantém próximo do obtidos do GPS, identificando rapidamente a medição errônea e reduzindo o ganho de Kalman representativo da odometria das rodas na linha verde.

No experimento seguinte, representado pela Figura 4, ao utilizar valor de variância do modelo, σ_x^2 , igual a 0,02, os resultados atualizados, após alguns segundos do choque, da fusão de sensores se mantém próximo do obtidos do GPS. No entanto, a identificação da medição errônea e redução do ganho de Kalman representativo da odometria são feitos de maneira mais lenta, causando um pico de erro no momento do choque.

Ao realizar terceiro experimento, representado pela Figura 5, o valor de variância do modelo, σ_x^2 , igual a 0,0002 (valor bem menor comparado aos anteriores), com resultados atualizados, após alguns segundos do choque, da fusão de sensores divergem totalmente do GPS. Essa reação provém da lenta convergência do ganho de Kalman, ao existir uma mudança brusca das medições.

Figura 3. Gráfico de representação dos erros para Odometria das Rodas (verde), GPS (azul) e EKF (vermelho) no eixo X para valor de Covariância da Matriz R, σ_x^2 igual a 0,2.

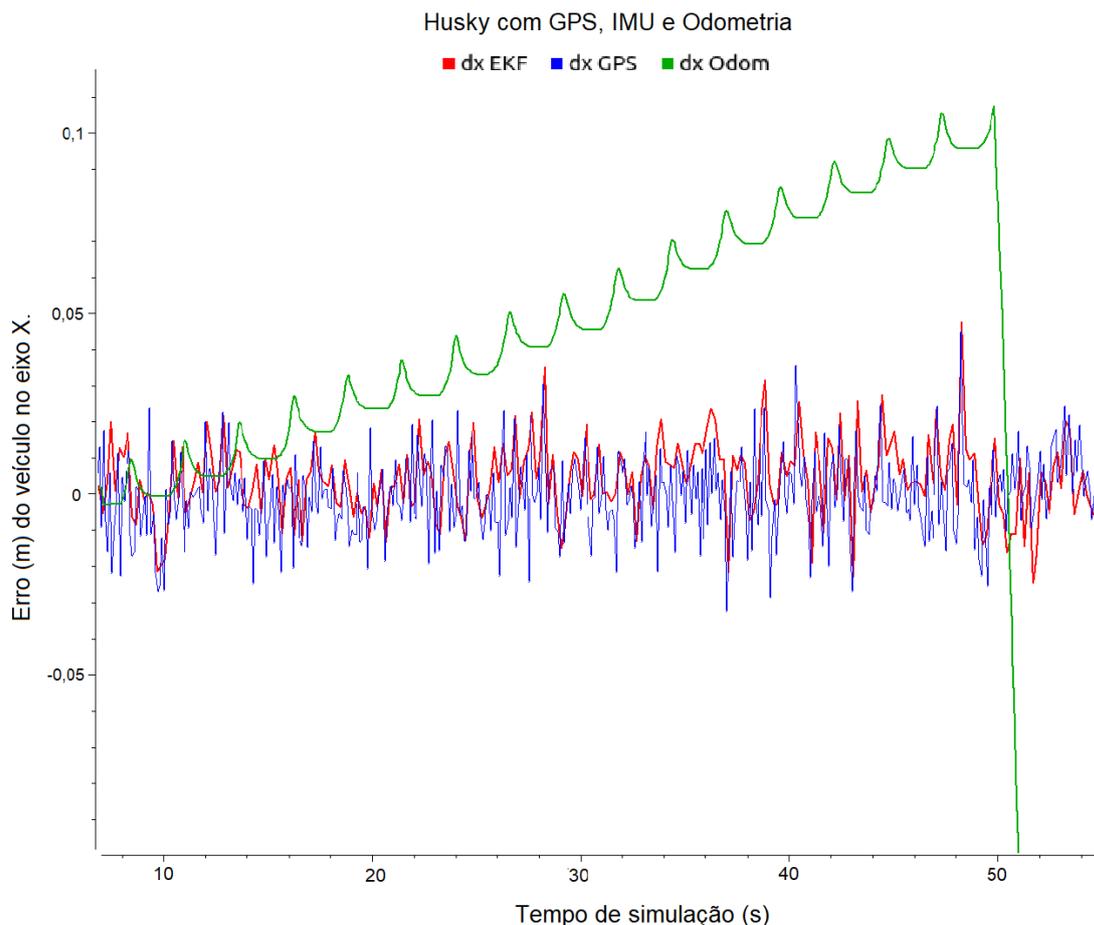




Figura 4. Gráfico de representação dos erros para Odometria das Rodas (verde), GPS (azul e EKF (vermelho) no eixo X para valor de Covariância da Matriz R , σ_{x-}^2 igual a 0,02.

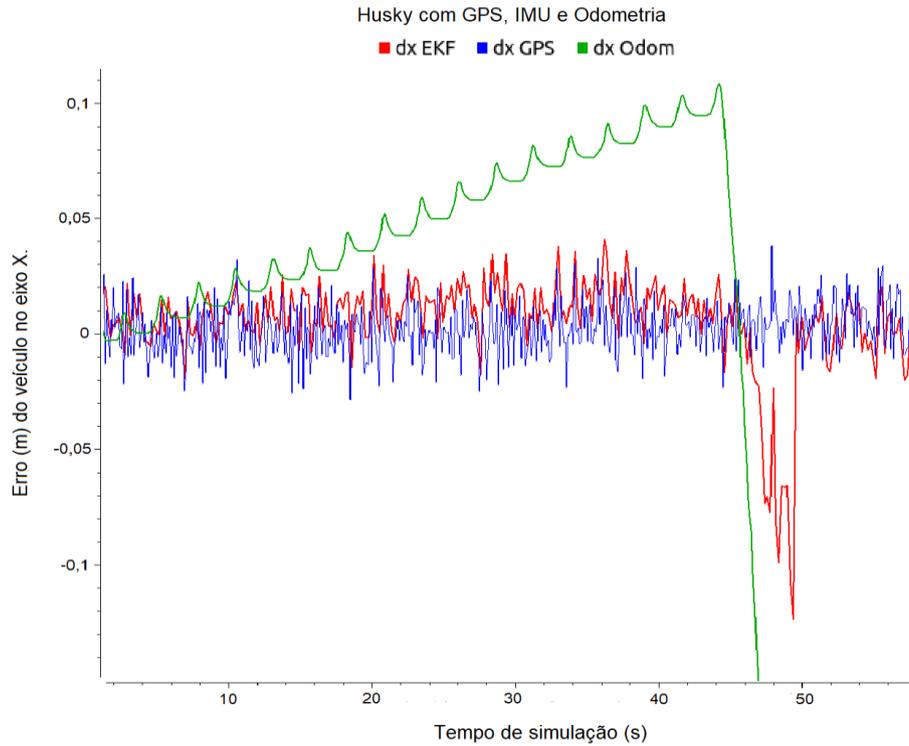
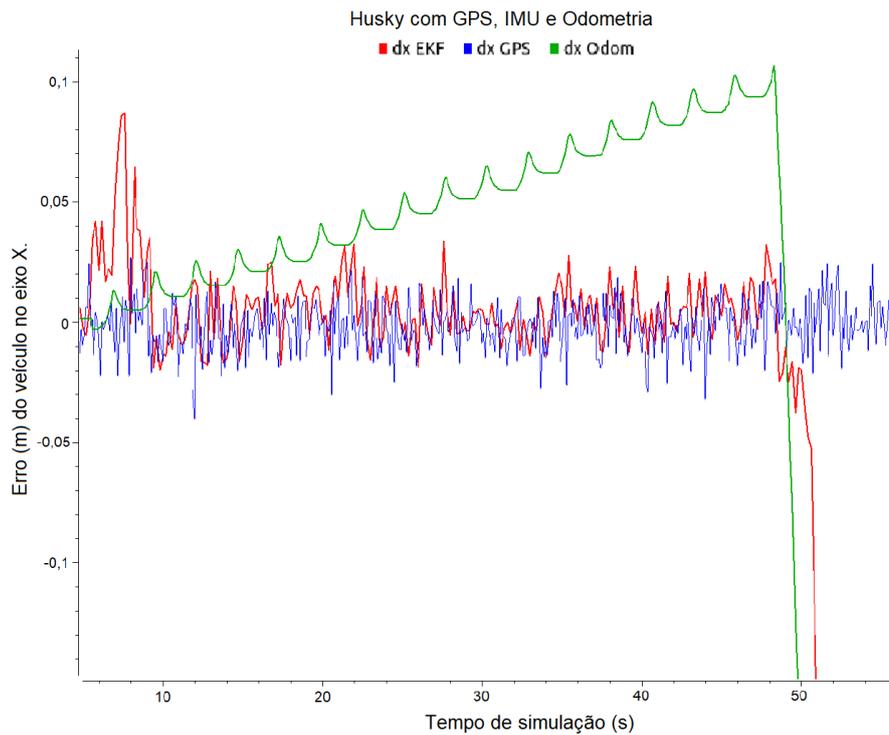


Figura 5. Gráfico de representação dos erros para Odometria das Rodas (verde), GPS (azul) e EKF (vermelho) no eixo X para valor de Covariância da Matriz R , σ_{x-}^2 igual a 0,0002.





4. CONCLUSÃO

Como conclusão desse trabalho, foi possível analisar os algoritmos de filtro de Kalman disponibilizados pelo ROS, além de observar a importância de utilizar outros sensores e uma matriz de covariância configurada corretamente, como uma forma de trazer melhores resultados, evitando erros nos dados que precisam ser cada vez mais precisos e acurados na sua aplicação. Para trabalhos futuros, uma simulação com movimentos variados, com outros sensores e uma nova matriz de covariância, permitem análise complementar com maior número de dados e situações.

Agradecimentos

Agradecimentos à EMBRAPA pelo financiamento da bolsa de projeto, aos Eng. Rafael Barreto, Mateus Amarantes e Ivan de Oliveira pelo tempo disponibilizado e grandes ensinamentos no mundo da robótica.

5. REFERÊNCIAS

- ¹ SIEGWART, R.; NOURBAKHSI, I. "Robotic Autonomy Summer Course", In: Mobile Robot Localization. **Carnegie Mellon University**, 2002. Disponível em: <<http://www.cs.cmu.edu/~rasc/Download/AMRobots5.pdf>>. Acesso em: 10 Agosto 2019.
- ² OPEN SOURCE ROBOTICS, F. About ROS and Open Source Robotics. **Open Source Robotics**, 2018. Disponível em: <<http://www.ros.org/about-ros>>. Acesso em: 15 Julho 2019.
- ³ LI, Q.; LI, R.; JI, K. **Kalman filter and its application**. Proc. - 8th Int. Conf. Intell. Networks Intell. Syst.. [S.l.]: ICINIS 2015. 2016. p. 74-77.
- ⁴ LEE, D.; SON, S.; YANK, K. **Sensor Fusion Localization System for Outdoor Mobile Robot**. 2009 ICCAS-SICE. [S.l.]: [s.n.]. 2009. p. 1384-1387.
- ⁵ LEVY, S. The Extended Kalman Filter: An Interactive Tutorial. **SIMON D. LEVY - Computer Science Department, Washington and Lee University**, 2018. Disponível em: <<https://simondlevy.academic.wlu.edu/kalman-tutorial/the-extended-kalman-filter-an-interactive-tutorial-for-non-experts-part-10/>>. Acesso em: 02 Agosto 2019.
- ⁶ ZDEŠAR, A. et al. **Wheeled Mobile Robotics**. 1ª. ed. Oxford: Butterworth-Heinemann, 2017.
- ⁷ CLEAR PATH ROBOTICS. HUSKY - UNMANNED GROUND VEHICLE. **Clear Path Robotics - Husky**, 2018. Disponível em: <<https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/>>. Acesso em: 05 Agosto 2019.