

AValiação de técnicas de odometria aplicadas a um dispositivo robótico móvel

EDUARDO COTA^{*†}, FILIPE ROCHA^{*}, HÉCTOR AZPÚRUA^{*}, GUSTAVO FREITAS^{*}

^{*}*Instituto Tecnológico Vale (ITV)*
Ouro Preto, MG, Brasil

[†]*Escola de Minas - Universidade Federal de Ouro Preto*
Departamento de Eng. Controle e Automação e Técnicas Fundamentais
Ouro Preto, MG, Brasil

Emails: educota@gmail.com, filipe.rocha@itv.org, hector.azpurua@itv.org,
gustavo.medeiros.freitas@itv.org

Abstract— Mobile robots need to estimate their own location in order to reach operation success. The indoor localization becomes a challenge apart since there is no GPS signal. Thus, this work presents an evaluation of odometry techniques, using different approaches with encoders, RGB-D camera, lidar and inertial measurement unit (IMU) to estimate the displacement of a robotic device in a closed environment. We used Vicon cameras as motion capture system to define a ground truth and evaluate the obtained results. The tests were performed with the EspeleoRobô, a remote device for cave monitoring.

Keywords— Mobile robot, Odometry techniques, Kinematic modeling.

Resumo— Robôs móveis necessitam estimar a própria localização para que sua operação obtenha sucesso. A localização em ambientes fechados se torna um desafio à parte visto que não há sinal GPS. Nesse contexto, este trabalho aplica e avalia técnicas de odometria utilizando abordagens com diferentes sensores incluindo *encoders*, câmera RGB-D, sensor lidar e unidade de medição inercial (IMU), para estimar o deslocamento de um dispositivo robótico em ambiente fechado. Câmeras Vicon foram empregadas para capturar o movimentos definindo o *ground truth* para avaliar os resultados obtidos. Os testes foram realizados com o EspeleoRobô, um dispositivo desenvolvido para realizar o monitoramento remoto de cavidades.

Palavras-chave— Robô móvel, Técnicas de odometria, Modelagem cinemática.

1 INTRODUÇÃO

Robôs móveis são dispositivos teleoperados, semi ou completamente autônomos, desenvolvidos para atividades específicas em que a mobilidade é necessária. A localização é crucial para o desenvolvimento de autonomia de um robô móvel. Na teleoperação, um dos maiores desafios é proporcionar ao operador uma sensação de presença estando em um ambiente remoto. Desta forma, sensores são comumente acoplados aos dispositivos teleoperados e autônomos visando prover o maior número possível de informações sobre as condições do equipamento e da operação.

O EspeleoRobô é um projeto da Vale S.A. que visa reduzir a exposição de operadores a riscos presentes nas atividades de inspeção e monitoramento de cavidades naturais. Para cumprir esse objetivo, uma plataforma robótica móvel foi adquirida. Buscando reestruturar e estender as capacidades do robô, o Instituto Tecnológico Vale foi acionado. Inicialmente configurado como um robô de seis pernas, o dispositivo sofreu uma série de modificações mecânicas, de *hardware* e *software* (Cota et al., 2017), buscando adequação aos ambientes de operação. Dentre as adaptações, outros mecanismos de locomoção foram propostos. Rodas, rodas tipo estrela e esteiras, além de modos híbridos de locomoção combinando tais mecanismos, estão sendo utilizados no robô.

Após as melhorias, o dispositivo se mostrou uma solução versátil, não somente para inspeção e monitoramento de cavernas, bem como para atividades em ambientes de difícil acesso, espaços confinados e locais onde a entrada de humanos é perigosa. Nesse sentido, o robô tem sido útil em inspeções de galeria de barragens, minas desativadas, áreas de risco de desabamento, além de cavidades localizadas em áreas de mineração.

Uma das maiores dificuldades da operação do robô nesses ambientes é localizar o dispositivo no interior dos mesmos. Isso se deve basicamente à ausência de localização global (uso de GPS ou similares), à baixa confiabilidade de magnetômetros e sensores inerciais em ambientes com concentração de ferro, e ao escorregamentos das rodas. Além disso, a utilização de diferentes mecanismos de locomoção impede a definição de um único modelo cinemático para descrever o movimento do robô. Tendo em vista estas limitações, um sistema que estime ao longo do tempo o deslocamento do robô dentro dos locais de operação pode acarretar em melhorias na qualidade da teleoperação, bem como iniciar desenvolvimentos para autonomia e mapeamento de ambientes monitorados com nuvem de pontos coloridos 3D.

Este trabalho descreve parte da solução do problema de localização do EspeleoRobô, apresentando uma avaliação de técnicas de odometria desenvolvidas para ambientes estruturados, testa-

das em laboratório. Os desenvolvimentos propostos podem ser aplicados em diferentes arquiteturas de robôs móveis. São desenvolvidas quatro implementações utilizando *encoders*, câmera RGB-D, sensor laser e sensor laser combinado com IMU.

Testes realizados em laboratório compararam a performance das abordagens com um preciso *ground truth* de acurácia de 1mm, formado por um sistema de captura de movimento com oito câmeras Vicon¹. As implementações mostraram-se funcionais e podem ser utilizadas em atividades de inspeção e monitoramento de ambientes como galeria de barragem e outros ambientes restritos.

O restante do texto está estruturado da seguinte forma: uma discussão sobre trabalhos relacionados é apresentada na Seção 2, seguida pela Seção 3 na qual é descrita a metodologia proposta de avaliação de odometria. Os experimentos e resultados são apresentados na Seção 4 e finalmente na Seção 5 são mostradas as conclusões e trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

De acordo com (Dudek and Jenkin, 2010), o conceito básico de odometria consiste em desenvolver um modelo matemático que descreve como os movimentos selecionados de juntas induzem movimento no veículo como um todo, e então integrar esses movimentos específicos ao longo do tempo com o intuito de criar um modelo da posição e orientação atual do veículo.

Duas arquiteturas comumente utilizadas em veículos com rodas são: diferencial e *skid-steer*. O modelo diferencial é composto por duas rodas paralelas montadas em um eixo comum, controladas por motores separados. O modelo *skid-steer* geralmente utiliza rodas ou esteiras para se locomover. A direção de ambos os dispositivos é baseada no controle das velocidades relativas dos lados esquerdo e direito.

Para definir modelos cinemáticos que representem estas arquiteturas é necessário adotar simplificações que geram erros. Segundo (Pentzer et al., 2014), assumir que não há escorregamento em veículos *skid-steer* causa má estimativa de posição. Geralmente, para computar o movimento das rodas são utilizados *encoders*, sensores empregados para medir o giro destas. Quando a roda escorrega, o movimento é computado; porém o robô não se desloca. Dessa forma, esse erro será computado e propagado por todo o cálculo de deslocamento, somando-se a outros erros e causando imprecisão.

Devido à ausência do sinal GPS em ambientes fechados, à imprecisão da odometria utilizando a leitura incremental de *encoders* proveniente do escorregamento, e ao avanço na área da visão computacional, surge o conceito de odometria visual

citado por Maimone et al. (2007) e outros, além da utilização de odometria baseada em sensores laser, como proposto por Lu and Milios (1997). Nestas odometrias alternativas são utilizadas câmeras e sensores laser para aquisição de informações referenciais. As informações adquiridas, juntamente com cálculos matemáticos, permitem a determinação de distâncias entre o robô móvel e objetos ao seu redor, que passam a ser utilizados com referência durante a navegação. Em ambientes controlados, é possível ainda a utilização de *landmarks* artificiais (Chenavier and Crowley, 1992), que são referências adicionadas ao ambiente a fim de facilitar a resolução do problema de localização.

No desenvolvimento de Achtelik et al. (2009) é proposta a utilização em conjunto de um sensor laser *rangefinder* e câmera *stereo* para estimar posição e orientação de um helicóptero *quadrotor* em ambiente fechado não estruturado e com ausência de sinal GPS. Para validação dos resultados, o *ground truth* foi obtido através de um sistema de captura de movimento Vicon.

Surmann et al. (2003) propõem um sistema automático de exploração e digitalização de ambientes fechados utilizando um robô móvel, um sensor laser *rangefinder* e três módulos de *software*. O primeiro módulo utiliza um variante do algoritmo ICP (*Iterative Closest Point*) para registrar varreduras 3D em um sistema de coordenadas comum e relocalizar o robô. O segundo e terceiro módulos são responsáveis pelo desvio de obstáculos e navegação autônoma do robô.

A combinação de câmera e IMU para localização de dispositivo móveis é apresentada na implementação de Hesch et al. (2014). O sistema de odometria visual-inercial utiliza um filtro de Kalman estendido para fundir as medições das câmeras e IMU.

Endres et al. (2012) apresentam um sistema para a solução do problema de SLAM (*Simultaneous Localization and Mapping*) utilizando câmera RGB-D, como o Microsoft Kinect. A avaliação é feita observando acurácia, robustez e tempo de processamento para três diferentes descritores de características, incluindo o SURF (*Speeded Up Robust Features*). A abordagem dos autores pode ser dividida em quatro passos: extração de características visuais (do inglês *visual features extraction*) a partir de imagens coloridas; correspondência (do inglês *matching*) destas características com características de imagens anteriores; avaliação da profundidade da imagem na posição das características detectadas e estimativa de transformação relativa entre os *frames* usando RANSAC (*RANdom SAmple Consensus*).

Para os desenvolvimentos propostos, os autores utilizaram ROS - *Robot Operating System*, uma plataforma de prototipagem rápida de robôs (Quigley et al., 2009).

¹<https://www.vicon.com/>

3 METODOLOGIA

Esse capítulo descreve o dispositivo e sensores utilizados nos desenvolvimentos, os modelos cinemáticos adotados para testes com *encoders* e os métodos usados para o cálculo de odometria.

3.1 EspeleoRobô e instrumentação embarcada

Os desenvolvimentos foram realizados com o EspeleoRobô configurado com modo de locomoção de seis rodas. A locomoção é realizada por meio de seis motores Maxon MCD EPOS de 60W, que possuem *encoders* digitais magneto-resistivos, com os quais é possível medir a velocidade rotacional exercida em cada atuador.

Uma impressora 3D e o material ABS foram utilizados para imprimir uma torre e um suporte para compor, junto dos sensores, uma estrutura de instrumentação. Sobre a torre, fixada no centro do robô, foram posicionados um sensor laser Hokuyo UTM-30LX e um Kinect da Microsoft. Foi necessário elevar a altura desses sensores para que as rodas do robô não interferissem nas medições realizadas. O suporte foi fixado sobre o Kinect para posicionar esferas refletivas com as quais é possível rastrear o movimento do robô. Uma unidade de medição inercial (IMU) Xsens MTi-G-710 foi fixada, à direita da torre, com a mesma orientação dos sensores citados. O robô, a estrutura de instrumentação e o sistema de coordenadas do robô podem ser observados na Figura 1.

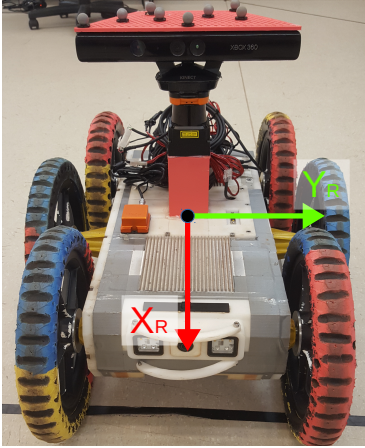


Figura 1: EspeleoRobô, sensores e suportes utilizados nos desenvolvimentos.

3.2 Técnicas de odometria

Nesta seção são apresentadas as técnicas de odometria implementadas.

3.2.1 Odometria com *encoders*

Nesta implementação foram utilizadas as informações de giro das rodas lidas pelos *encoders* como entradas em modelos cinemáticos para se calcular o deslocamento do robô.

Visando a simplificação dos cálculos e da implementação dos testes, uma primeira abordagem foi realizada considerando o dispositivo como tendo um sistema de direção diferencial. Após a primeira abordagem, foi estudada uma outra utilizando o modelo cinemático do *skid-steer*, com o qual o dispositivo utilizado nesta pesquisa pode ser melhor representado.

• Modelo diferencial

Observando a teoria apresentada por Siciliano et al. (2010) e utilizando a integração numérica de 1ª ordem pelo método de Euler é possível somar pequenas variações de posição (Δx , Δy) e orientação ($\Delta \theta$) em um curto intervalo de tempo Δt (Equações 1, 2 e 3), obtendo ao longo do tempo o deslocamento realizado pelo robô. As velocidades lineares nos flancos direito e esquerdo do robô (v_d e v_e) são calculadas a partir da leitura dos *encoders*.

$$\Delta x = \frac{(v_d + v_e)}{2} \cos(\theta) \Delta t \quad (1)$$

$$\Delta y = \frac{(v_d + v_e)}{2} \sin(\theta) \Delta t \quad (2)$$

$$\Delta \theta = \frac{(v_d - v_e)}{L} \Delta t \quad (3)$$

• Modelo *skid-steer*

Considerando os desenvolvimentos de Mandow et al. (2007), a cinemática de um robô *skid-steer* pode ser descrita por:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R(\theta) A \begin{bmatrix} v_e \\ v_d \end{bmatrix} \quad (4)$$

onde $R(\theta) \in$ grupo especial ortogonal $SO(3)$ é a matriz de rotação (Equação 5), $A \in \mathbb{R}^{3 \times 2}$ é a matriz de modelagem cinemática do dispositivo *skid-steer* (Equação 6) e v_e e v_d representam as velocidades lineares nos flancos esquerdo e direito, respectivamente.

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$A = \frac{\alpha}{2y_{CIR}} \begin{bmatrix} -y_{CIR} & -y_{CIR} \\ 0 & 0 \\ -1 & 1 \end{bmatrix} \quad (6)$$

As variáveis α e y_{CIR} são valores encontrados empiricamente para ajustar o modelo cinemático ao dispositivo real utilizado, reduzindo erros de simplificações como assumir que não há escorregamento nas rodas. O fator α é um fator de correção devido à mecânica do robô e ao tipo de terreno, e o parâmetro y_{CIR} é a posição do centro instantâneo de rotação com respeito ao eixo y_R .

Uma maneira prática de se encontrar a matriz A é apresentada por (Mandow et al., 2007). Experimentos foram desenvolvidos em laboratório para a obtenção dos valores utilizando as Equações 7 e 8:

$$\alpha = \frac{2d}{\int v_d(t)dt + \int v_e(t)dt} \quad (7)$$

$$y_{CIR} = \frac{\int v_d(t)dt - \int v_e(t)dt}{2\theta^*} \quad (8)$$

A determinação de α foi feita realizando um teste de locomoção em linha reta. Para essa parte do experimento foram coletadas a distância d total percorrida pelo dispositivo no percurso e as velocidades v_e e v_d desenvolvidas.

O parâmetro y_{CIR} foi encontrado por meio de um teste de rotação pura, no qual também foram coletadas as velocidades desenvolvidas e o ângulo rotacionado. Nesta parte do experimento foram aplicadas velocidades v_e e v_d de módulos iguais, mas de sinais inversos nas rodas do robô. Foram mensuradas as velocidades dos flancos ao longo do ensaio e o ângulo total (θ^*) rotacionado pelo dispositivo.

As velocidades empregadas foram coletadas por meio da leitura dos *encoders*, enquanto a distância e os ângulos foram medidos com o sistema de captura de movimento com câmeras Vicon. Obtendo os valores de 0.9838 para α e 0.3045 para y_{CIR} , determina-se os valores da matriz de modelo cinemático A (Equação 6).

Considerando a matriz de rotação $R(\theta)$ (Equação 5), a matriz A calculada e as velocidades lineares das rodas esquerdas e direitas (v_e e v_d), é possível somar as pequenas variações em x , y e θ para cada instante Δt (Equação 9) aplicando novamente a integração numérica de 1ª ordem pelo método de Euler:

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = R(\theta)A \begin{bmatrix} v_e \\ v_d \end{bmatrix} \Delta t \quad (9)$$

A discretização é necessária para implementar o cálculo no programa de processamento de dados. O menor intervalo possível, definido considerando limitações dos sensores e *software* é de aproximadamente 0,15s. Tendo em vista que o robô se move a baixas velocidades tanto nos testes de laboratório, quanto em futuras operações de campo, esse intervalo foi considerado aceitável.

Na Figura 2 são mostradas as trajetórias calculadas utilizando os dois modelos durante um experimento. É possível observar que o modelo *skid-steer* comporta-se melhor quando comparado ao modelo diferencial para representar a trajetória do EspeleRobô por meio da leitura dos *encoders*. Dessa forma, apenas a abordagem com este modelo será comparada com as outras implementações de odometria.

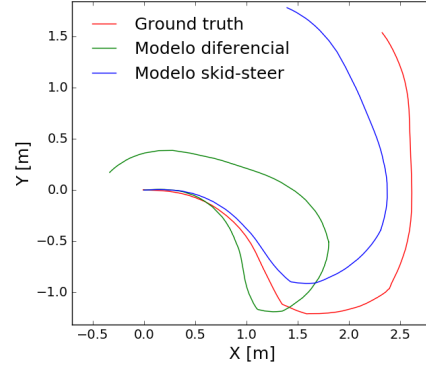


Figura 2: Comparação de precisão dos modelos diferencial e skid-steer.

3.3 Odometria visual com câmera RGB-D

Na avaliação da odometria visual, foi utilizado um sensor RGB-D (Kinect) de luz estruturada, o qual fornece dados de profundidade e imagens RGB simultaneamente, em conjunto com o algoritmo de SLAM proposto por (Labbé and Michaud, 2014).

O método usa técnicas de SLAM baseadas em grafos para realizar o fechamento de *loop*: a estrutura do mapa é dividida em nós e arcos, e em cada nó são armazenados as poses de odometria do mapa, e informações visuais como imagens, palavras visuais, etc. Os arcos armazenam as transformações entre um par de nós. Um *bag-of-words*, baseado em *features* extraídas com SURF nas imagens RGB, é computado para gerar um dicionário visual incremental.

Especificamente, a odometria usada neste trabalho é calculada usando a técnica F2M (*Frame-2-Map*). Essa técnica usa uma versão reduzida e temporal do mapa gerado, onde o último *frame* estimado do mapa é comparado com os *features* visuais e de profundidade de um novo *frame*. Finalmente é usado RANSAC para calcular a transformação/odometria entre as *features* correspondentes. Neste aplicação não é necessário o fechamento de *loop*.

3.4 Odometria com sensor laser

O dispositivo utilizado nesta abordagem é do tipo LiDAR, do inglês *Light Detection And Ranging*, que utiliza feixes de luz refletidos para medir distâncias e, a partir destas medições, obter informações sobre o ambiente ou objetos localizados nele. Com um ângulo de abrangência de 270°, resolução de 0.25° e alcance máximo de 30m, o sensor precisa de apenas 25ms para realizar uma varredura, retornando então 1080 medições de distâncias.

Foi avaliada a odometria com laser usando uma variante do algoritmo ICP (Iterative Closest Point) chamada PLICP (Censi, 2008). O ICP é um algoritmo que dados um conjunto de pontos $P = \{p_1, \dots, p_n\}$ e uma superfície S^{ref} estima as rotações e translações ($q = (t, \theta)$) que minimizam o erro quadrático entre esses dois conjuntos:

$$\min \sum \|p_i \oplus q_{k+1} - \Pi\{S^{ref}, p_i \oplus q_k\}\|^2 \quad (10)$$

onde $\oplus$ denota o operador de rotação e translação, e $\Pi\{S^{ref}, \cdot\}$ denota a projeção euclidiana em S^{ref} .

O intuito dessa técnica se baseia na teoria de que se as correspondências corretas são conhecidas entre dois conjuntos, então é possível calcular as transformações relativas entre eles, e assim conhecer o deslocamento (odometria). Censi (2008) desenvolveu uma variante do algoritmo que usa uma métrica *point-to-line* ao invés da *point-to-point*, a qual é demonstrada ser mais eficiente, precisando de menos iterações para convergir:

$$\min_{q_{k+1}} \sum (n_i^t [p_i \oplus q_{k+1} - \Pi\{S^{ref}, p_i \oplus q_k\}])^2 \quad (11)$$

O tempo do processo de registro de escaneamentos pode ser melhorado com a estimativa do movimento rotacional do robô entre os escaneamentos. Em uma segunda abordagem utilizando sensor laser, o algoritmo PLICP foi parametrizado para usar a componente de rotação em z (*yaw*) obtida através de um sensor IMU, onde a estimativa de rotação é calculada mediante a diferença absoluta entre o *yaw* do *frame* anterior e o atual.

4 RESULTADOS EXPERIMENTAIS

Nesta seção são apresentados o sistema de validação e estrutura de testes utilizados, além dos experimentos realizados e resultados atingidos.

4.1 Sistema de validação e estrutura de testes

Neste trabalho foi utilizado como *ground truth* um conjunto de oito câmeras Vicon, duas modelo Vero e seis modelo Bonita, formando o sistema de captura de movimento. As câmeras identificam distâncias por meio de reflexos luminosos e funcionam integradas a um *software* próprio, chamado Tracker (versão 3.3). A disposição das câmeras no Laboratório de Robótica do Instituto Tecnológico Vale pode ser observada na Figura 3. Para identificar o EspeleoRobô foram coladas marcas refletivas como mostrado na Figura 4a. Na figura 4b é possível observar o objeto EspeleoRobô criado no *software* proprietário, além dos eixos x_R y_R do sistema de coordenadas do robô.

Na Figura 5 é possível observar o esquema de ligação dos dispositivos para realização de testes.

As câmeras Vicon são interligadas por um *switch* que por sua vez está conectado à estação Vicon, computador com sistema operacional Windows 7. As câmeras capturam os movimentos descritos pelo conjunto de esferas refletivas pré-configurado no *software* Tracker (Figura 4b). Os dados de leitura das câmeras são processados na estação Vicon, e disponibilizados através

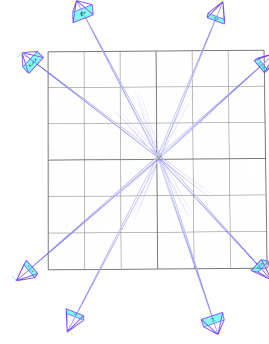


Figura 3: Disposição das 8 câmeras no Laboratório de Robótica do ITV.

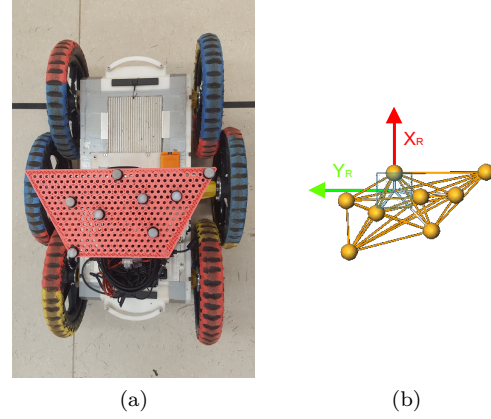


Figura 4: Posicionamento das esferas refletivas e objeto criado no *software* Tracker para captura de movimento.

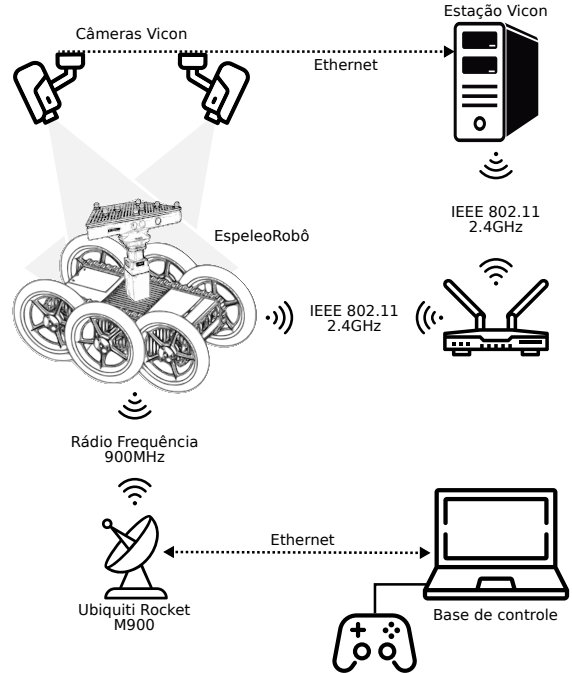


Figura 5: Esquema de ligação de equipamentos para testes.

de uma API (*Application Programming Interface*) para uma rede *wireless*.

A base de controle controlada por um operador envia os comandos ao robô através um rádio Ubiquiti modelo Rocket M900, ligado a uma antena omnidirecional Yagi AMY-9M16. O mini-computador embarcado no EspeleoRobô recebe e processa os comandos da base de controle, acarretando no movimento do robô.

O mesmo computador acessa na rede as informações processadas das câmeras e recebe diretamente as informações dos *encoders*, Hokuyo, Kinect e unidade inercial via portas USB. Em um mesmo nó do ROS, todas essas informações são processadas gerando arquivos csv e gráficos para comparação das implementações.

4.2 Experimentos

Foram conduzidos experimentos num ambiente controlado (laboratório) visando avaliar as abordagens de odometria propostas.

Os testes foram realizados com o robô se locomovendo numa trajetória de circuito fechado, com formato semelhante ao número oito ou símbolo de infinito (∞). Essa configuração é comumente utilizada em avaliações de odometria. A execução da trajetória foi comandada por um operador por meio da base de controle do EspeleoRobô. O dispositivo pode chegar a aproximadamente 1m/s e foi operado em dois modos de velocidade: (i) com restrição de 50% da velocidade máxima e (ii) sem restrição de velocidade.

Para iniciar cada teste o robô foi posicionado no ponto zero (posição: $x = 0, y = 0, z = 0$; orientação: $roll = 0, pitch = 0$ e $yaw = 0$) com respeito ao sistema de coordenadas do sistema de captura de movimento com câmeras Vicon (*ground truth*). Dessa forma, no início de cada teste, os sistemas de coordenadas do robô e do sistema de captura são coincidentes. Após o posicionamento, todos os algoritmos foram acionados (também começando do ponto zero) para que, durante os movimentos, a posição e orientação fossem computados.

Com acesso direto aos sensores e ao *ground truth*, um nó do ROS implementado no mini-computador embarcado na plataforma robótica realiza o processamento dos dados das diferentes técnicas de odometria e gera um *log* com os dados processados.

A partir da locomoção do robô é possível extrair o deslocamento em x, y , e a rotação em torno do eixo z ao longo do tempo. Dessa forma, tem-se a posição e orientação do robô no plano de forma temporal as quais integradas formam a trajetória descrita pelo dispositivo.

Na abordagem com *encoders* os cálculos consideraram o modelo cinemático *skid-steer*. Para a implementação de odometria visual foi utilizado o algoritmo proposto em (Labbé and Michaud,

2014), *rtabmap_ros*² o qual foi disponibilizado pelos autores. Nos experimentos com laser, foi usada uma implementação do algoritmo *PLICP* (Censi, 2008) chamada *laser_scan_matcher*³, também disponibilizado pelo autor. Na abordagem em que é utilizado o sensor inercial em conjunto com o laser, as leituras do sensor contribuem para o cálculo da rotação em torno do eixo z .

Nas Figuras 6 e 7 é possível observar as trajetórias descritas pelo dispositivo e nas Figuras 8 e 9 são mostrados os erros entre as melhores abordagens (Hokuyo e Hokuyo+IMU). Os resultados obtidos com velocidade máxima limitada à 50% e sem limitação se encontram sumarizados na Tabela 1. Para melhor compreensão da magnitude dos erros, é necessário considerar que os testes foram realizados em uma área de 6x6m.

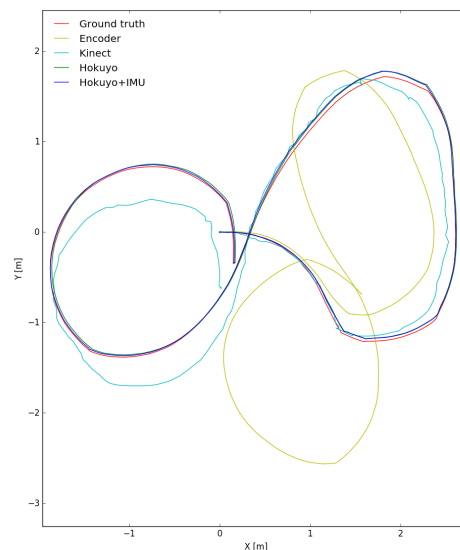


Figura 6: Trajetória computada ao acionar o robô com velocidade limitada.

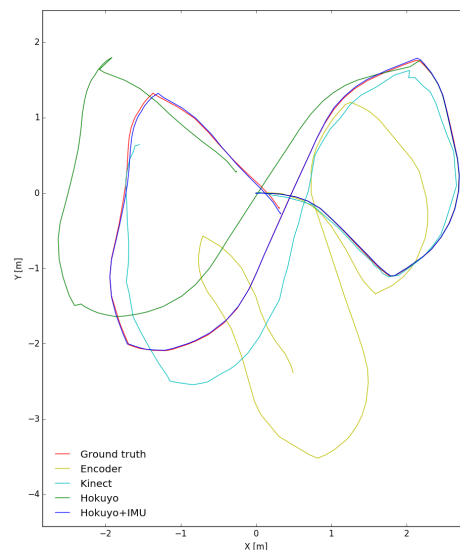


Figura 7: Trajetória computada ao acionar o robô com velocidade sem restrições.

²http://wiki.ros.org/rtabmap_ros

³http://wiki.ros.org/laser_scan_matcher

Erro	x (cm)				y (cm)				θ (°)			
	Enc.	Kinect	Hok.	H.+IMU	Enc.	Kinect	Hok.	H.+IMU	Enc.	Kinect	Hok.	H.+IMU
Máx.	253,9	185,6	75,1	8,8	224,2	84,7	60,3	10,5	395,9	353,4	349,2	7,7
Máx.*	277,3	27,3	3,9	3,4	189,3	53,6	7,5	8,5	403,5	355,8	3,1	2,8
Méd.	64,6	42,8	30,8	1,5	95,2	30,3	22,6	2,2	121,0	33,5	11,6	2,2
Méd.*	82,9	11,3	1,5	1,1	53,8	14,1	2,8	2,5	126,6	7,1	1,2	1,1
σ	75,7	62,4	29,5	1,6	86,2	31,5	22,1	2,4	94,4	63,2	50,0	1,4
σ^*	84,4	7,8	1,0	0,8	53,8	15,5	2,2	2,3	105,9	37,4	0,7	0,6
IAE	9170	6077	4376	220,3	13517	4296	3211	309,6	17185,6	4757	1652	313,1
IAE*	21720	2953,1	404,0	278,0	14107	3689	732,6	660,2	33170	1868,1	312,2	276,9

*com limite de 50% da velocidade máxima

Tabela 1: Tabela de erros de odometria com velocidade média e velocidade máxima.

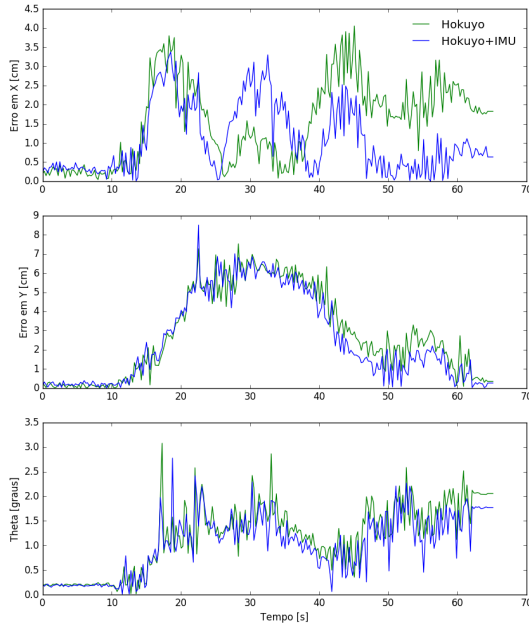


Figura 8: Erros computados ao acionar o robô com velocidade limitada - Hokuyo vs. Hokuyo+IMU.

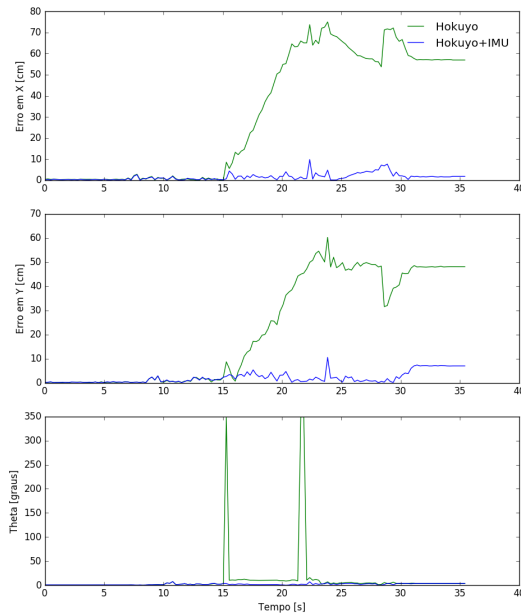


Figura 9: Erros computados ao acionar o robô com velocidade sem restrições - Hokuyo vs. Hokuyo+IMU.

A odometria calculada a partir da leitura do movimento rotacional realizado pelas rodas apresenta o pior resultado em ambas situações testadas (velocidade limitada e não-limitada). É possível observar claramente quando, em curvas mais fechadas, há o escorregamento das rodas. Os *encoders* computam o movimento, mas o robô não o realiza. Isso resulta na computação de curvas mais fechadas do que a realidade. O erro se acumula, e o ponto de chegada apresentado pela abordagem com *encoder* é bem distante no ponto real. Erros chegam a atingir 277,3cm na posição em x e mais de uma volta completa (403,5° e 395,9°) na orientação θ .

A odometria implementada com câmera RGB-D e *rtabmap_ros* apresenta resultados com erros de até 27,3cm no eixo x , 53,6cm no eixo y , quando o robô descreve movimentos com velocidade de até 50% da sua capacidade máxima. Entretanto, quando não há restrições de velocidade, os cálculos perdem a referência, o método falha e a computação da trajetória é interrompida antes do término dos movimentos, apresentando grande erro.

A odometria com sensor laser do tipo LiDAR e *laser_scan_matcher* apresentou os melhores resultados. Para movimentos com velocidades limitadas o erro se manteve em até 3,9cm em x , 7,5cm em y e 3,1° na orientação. A adição da IMU trouxe melhoria em praticamente todas as métricas, como pode ser observado pela integral do erro absoluto, principalmente no erro em x .

Para movimentos sem restrições de velocidade, a adição da IMU trouxe melhoria relevante, principalmente quando há movimento bruscos em curvas ou rotações em torno do próprio eixo. Os erros em x e y apresentados por esta abordagem se mantiveram abaixo de 10,5cm, enquanto sem a utilização de IMU o erro chegou a 75,1cm em x e 60,3cm em y . Em relação aos pontos de chegada, a abordagem sem IMU tem erro 31 vezes maior em x e quase 7 vezes maior em y , se comparada com a abordagem que utiliza IMU. A diferença também apresenta grande magnitude quando comparados os erros médios, desvio padrão e integral do erro absoluto. Na Figura 9 também é possível observar a melhoria referida.

5 CONCLUSÕES

Nesse trabalho foi apresentada uma metodologia de avaliação de técnicas de odometria para robôs móveis. As técnicas mostradas foram implementadas no EspeleoRobô, um robô terrestre que possui múltiplos mecanismos e modos de locomoção. A validação dos resultados foi realizada utilizando um sistema de captura de movimento com câmeras Vicon.

É possível concluir que a abordagem de odometria com leitura da velocidade dos eixos do EspeleoRobô através de *encoders* traz imprecisões e acúmulo de erros com o tempo. A abordagem com o sensor Kinect não trouxe resultados praticáveis, visto que em movimentos rápidos a referência é perdida e os cálculos interrompidos. Isso se deve provavelmente ao alto processamento demandado pelo algoritmo, causando o não término dos cálculos em determinado ponto. A implementação utilizando somente sensor laser mostrou ser imprecisa em alta velocidade (movimentos bruscos e rotações em torno do próprio eixo).

A implementação que une sensor laser e IMU, por sua vez, apresentou resultados positivos consistentes em todas as situações testadas. Não houve em nenhum teste erros significativos na trajetória ou no ponto de chegada calculados.

A abordagem usando Hokuyo+IMU mostrou-se promissora para ser utilizada com o EspeleoRobô ou em plataformas móveis similares, principalmente em atividades de inspeção e monitoramento de ambientes confinados, de difícil acesso, como galerias de barragem. O desenvolvimento de abordagem utilizando sensores laser se faz ainda mais relevante quando aplicado a robôs que utilizam diferentes mecanismos de locomoção, como o EspeleoRobô, não sendo necessária a definição de um modelo cinemático para se calcular o deslocamento do dispositivo.

Em trabalhos futuros será considerada a combinação de Kinect com IMU, além do processamento *offline* das imagens geradas, visando verificar se realmente faltou processamento durante os cálculos de odometria e investigar a real causa da interrupção dos mesmos. Também serão avaliados os erros em relação à distância total percorrida, cada componente da pose e o perfil de velocidade durante os experimentos. Por fim, será avaliado o impacto do relevo do terreno com o intuito de propiciar um cálculo preciso de odometria para ambientes não planares e não controlados como cavernas.

Referências

Achtelik, M., Bachrach, A., He, R., Prentice, S. and Roy, N. (2009). Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments, SPIE.

- Censi, A. (2008). An icp variant using a point-to-line metric, *Robotics and Automation*, 2008. ICRA 2008. IEEE International Conference on, IEEE, pp. 19–25.
- Chenavier, F. and Crowley, J. L. (1992). Position estimation for a mobile robot using vision and odometry, *Robotics and Automation*, 1992. Proceedings., 1992 IEEE International Conference on, IEEE, pp. 2588–2593.
- Cota, E., Torre, M., Rocha, F., Garcia, G., Junior, A., Ramos, V., Queiroz, V., Zanini, V., Brito, G., Marques, A., Pinto, E., Nogueira, L., Freitas, G., Miola, W., Dos Reis, M., Araújo, R. and Brandi, I. (2017). Dispositivo de monitoramento remoto de cavidades - EspeleoRobô, *XIII Simpósio Brasileiro de Automação Inteligente*.
- Dudek, G. and Jenkin, M. (2010). *Computational principles of mobile robotics*, Cambridge university press.
- Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D. and Burgard, W. (2012). An evaluation of the rgb-d slam system, *Robotics and Automation (ICRA)*, 2012 IEEE International Conference on, IEEE, pp. 1691–1696.
- Hesch, J. A., Kottas, D. G., Bowman, S. L. and Roumeliotis, S. I. (2014). Camera-imu-based localization: Observability analysis and consistency improvement, *The International Journal of Robotics Research* **33**(1): 182–201.
- Labbé, M. and Michaud, F. (2014). Online global loop closure detection for large-scale multi-session graph-based slam, *Intelligent Robots and Systems (IROS 2014)*, 2014 IEEE/RSJ International Conference on, IEEE, pp. 2661–2666.
- Lu, F. and Milios, E. (1997). Robot pose estimation in unknown environments by matching 2d range scans, *Journal of Intelligent and Robotic systems* **18**(3): 249–275.
- Maimone, M., Cheng, Y. and Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers, *Journal of Field Robotics* **24**(3): 169–186.
- Mandow, A., Martinez, J. L., Morales, J., Blanco, J. L., Garcia-Cerezo, A. and Gonzalez, J. (2007). Experimental kinematics for wheeled skid-steer mobile robots, *Intelligent Robots and Systems*, 2007. IROS 2007. IEEE/RSJ International Conference on, IEEE, pp. 1222–1227.
- Pentzer, J., Brennan, S. and Reichard, K. (2014). Model-based prediction of skid-steer robot kinematics using online estimation of track instantaneous centers of rotation, *Journal of Field Robotics* **31**(3): 455–476.
- Quigley, M., Faust, J., Foote, T. and Leibs, J. (2009). Ros: an open-source robot operating system.
- Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. (2010). *Robotics: modelling, planning and control*, Springer Science & Business Media.
- Surmann, H., Nüchter, A. and Hertzberg, J. (2003). An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments, *Robotics and Autonomous Systems* **45**(3-4): 181–198.