

---

## Algoritmos de navegação de robôs móveis com tração nas quatro rodas utilizando sistemas de localização absolutos GNSS e RTK

### Navigation algorithms for four-wheeled drive mobile robots using GNSS and RTK absolute localization systems

Pablo France Salarolli<sup>1</sup>, Leonardo Gonçalves Baptista<sup>1</sup>, Caio Lopes Oliveira<sup>1</sup>, Carlos Torturella Valadão<sup>1\*</sup>, Daniel Fernando Tello Gamarra<sup>2</sup>, Rogério Passos do Amaral Pereira<sup>1</sup>, Gustavo Maia de Almeida<sup>1</sup>, Marco Antonio de Souza Leite Cuadros<sup>1</sup>

---

#### RESUMO

A robótica móvel é uma área em grande crescimento na pesquisa científica e na indústria, devido a sua versatilidade e capacidade de substituir os humanos, especialmente em trabalhos perigosos, e por apresentar melhor desempenho em certas aplicações. Várias delas são em ambientes externos, como em indústrias, aplicações aeroespaciais, petroquímicas, dentre outras. Esse artigo mostra a implementação de um sistema de localização de um robô móvel em ambiente outdoor, utilizando sistema de navegação global por satélite e odometria. Pretende-se, assim, obter uma forma robusta de localização, permitindo uma navegação confiável em diferentes terrenos. Testes foram realizados com a navegação por satélite e a odometria. Percebeu-se que o robô pôde trabalhar bem em ambientes externos capturando corretamente a localização GPS quando comparado com o *Google Earth Pro*. Com relação a odometria, houve pequenos erros quando o robô se moveu em linha reta, o que aumentou ao se realizar rotações. Isso se deve à característica do funcionamento do robô diferencial 4WD, onde as curvas geram deslizamento das rodas. Além disso, pode-se demonstrar o bom funcionamento quando feita a fusão dos dados de GNSS, IMU e odometria.

**Palavras-chave:** Geolocalização; Robôs Móveis; GNSS; GPS; ROS.

---

#### ABSTRACT

Mobile robotics is an area that has been growing rapidly in scientific research and industrial applications due to its versatility and ability to replace humans, particularly in hazardous jobs, and to offer better performance in some applications. Several of these are in outdoor environments, such as in industries, aerospace, and petrochemical applications, among others. This article presents the implementation of a mobile robot localization system in an outdoor environment using global satellite navigation and odometry. The goal is to obtain a robust form of localization, allowing reliable navigation in different types of terrain. Tests were conducted using data from the satellite navigation system and odometry. It was observed that the robot could work well in outdoor environments, capturing the GPS location accurately when compared to *Google Earth Pro*. The odometry measurements presented small errors when the robot moved in a straight line, which increased when it made rotations. This happens due to 4WD differential model features, where curves generate slippage, resulting in a different angular speed indicated by robot's encoders. Nevertheless, combining GNSS, encoders and IMU the robot could move adequately in the test environment.

**Keywords:** Geopositioning; Mobile Robotics; GNSS; GPS; ROS.

---

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo.

\*E-mail: carlostvaladao@gmail.com

<sup>2</sup>Universidade Federal de Santa Maria

## INTRODUÇÃO

As tecnologias envolvidas na quarta revolução industrial, mais conhecida como Indústria 4.0, possibilitam a construção de fábricas inteligentes, com produtividade e eficiência elevadas, capazes de produzir produtos customizáveis diretamente de pedidos do consumidor (KAGERMANN; WAHLSTER; HELBIG, 2013). BRAÑA (2019) aponta que são 12 as principais tecnologias emergentes que transformarão o sistema produtivo globalmente, porém alguns autores consideram que 3 delas são as mais impactantes: Internet das Coisas (IoT), representando a alta conectividade entre dispositivos, otimizando o processo produtivo e agregando valor aos produtos acabados; manufatura aditiva, simplificando as etapas de pesquisa e desenvolvimento até as etapas de produção; e robótica avançada, reduzindo custos de produção e logística permitindo uma automação flexível e customizável (EUROFUND, 2018).

Dentro de robótica, um dos campos de mais rápida expansão da pesquisa científica é a robótica móvel, sendo que os robôs móveis podem substituir os humanos em muitos campos, principalmente por motivos de retirar humanos de trabalhos perigosos (VEGA et al., 2021), como é o caso de muitos robôs para desinfecção do COVID-19 (PERMINOV et al., 2021; RUAN; WU; XU, 2021). Os robôs também podem apresentar melhor performance que humanos em certas aplicações (PARK et al., 2014). As aplicações da robótica móvel são muitas, como por exemplo vigilância, exploração planetária, patrulhamento de segurança, operações de resgate de emergência, reconhecimento, aplicações petroquímicas, automação industrial, construção, entretenimento, guias de museus, serviços pessoais, intervenção em ambientes extremos, transporte, assistência médica, bem como muitas outras aplicações industriais e não industriais (RUBIO; VALERO; LLOPIS-ALBERT, 2019).

Em ambientes externos, robôs móveis também podem ser aplicados como veículos de patrulha e segurança, como é o caso da aplicação do trabalho (LU; JUANG, 2018). O robô foi projetado para encontrar e rastrear alvos pré-definidos, fazendo uma inspeção e desviando de obstáculos no caminho. Para localização, foi utilizado um sistema com GPS diferencial e odometria das rodas. Sensores ultrassons e infravermelhos foram utilizados para detectar obstáculos que podem surgir no caminho do robô ao longo da patrulha. Imagens de uma câmera embarcada são utilizadas para detectar e identificar pessoas no caminho. Ao fim da patrulha, o robô envia uma mensagem para um segurança

informando se a região patrulhada está em condições normais e, caso uma pessoa não identificada seja detectada, a localização desta será enviada para as devidas providências.

FOUNTAS et al. (2020) conduziram uma revisão sistemática sobre robôs para agricultura, sempre destacando a complexidade das operações de campo e a necessidade de levar em conta diversas questões para permitir a transição de tais atividades para a robótica, como o custo-benefício, requisitos específicos e adaptabilidade. Eles pesquisaram robôs com diversas funcionalidades para aplicações em diferentes ambientes agrícolas, identificando quais os sistemas robóticos já existentes e propondo alguns desafios e diretrizes para o desenvolvimento da robótica agrícola. Por outro lado, ZHANG et al. (2020) apresentaram um trabalho relacionado à inspeção de barragens de água, que utilizava um robô multissensorial, um servidor na nuvem e terminais para usuários, além de algoritmos de inteligência artificial para reconhecimento em tempo real de rachaduras e presença de pessoas durante a vigilância. O robô possui dados de sensores inerciais, GNSS, odometria e mapeamento por LiDAR fusionados por meio do filtro Kalman estendido (EKF).

Considerando a importância de aplicações em robótica, em especial às que decorrem em ambientes externos, conforme apresentado acima, o presente trabalho visa implementar um sistema de navegação autônoma, utilizando localização por odometria e sensores inerciais, junto a correções com GNSS RTK. Unindo a alta precisão dessa tecnologia de posicionamento por satélite junto à continuidade da odometria e dos sensores inerciais, pretende-se obter um sistema de localização robusto, permitindo uma navegação confiável. Para isso, uma plataforma robótica será utilizada e adequações necessárias serão realizadas para que o robô navegue em diferentes tipos de terreno.

## FUNDAMENTAÇÃO TEÓRICA

A robótica é uma tecnologia muito presente atualmente, sendo um dos pilares da quarta revolução industrial, a Indústria 4.0 (BRAÑA, 2019). A aplicação da robótica é vasta e pode ser utilizada na agricultura, silvicultura (VOUGIOUKAS, 2019), educação (CURTO; MORENO, 2016), indústrias, e medicina, dentre outros (RUNCIMAN; DARZI; MYLONAS, 2019).

Os robôs móveis são sistemas robóticos capazes de realizar tarefas em variados lugares e são constituídos por uma plataforma movida por elementos locomotores, como

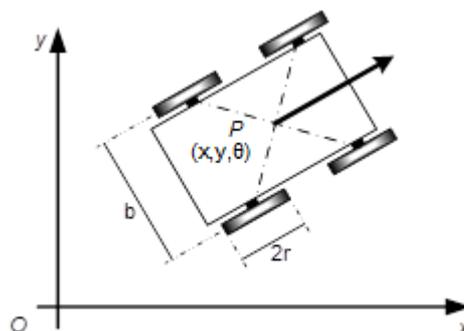
rodas, esteiras, pernas, hélices, dentre outros, dependendo do ambiente e da utilização do robô (GARCIA et al., 2007). Dentro da categoria dos robôs móveis, os que se locomovem por rodas, recebem um papel de destaque, visto que as rodas são um meio de locomoção mais simples, eficiente e barato se comparado a pernas ou esteiras, por exemplo (ALATISE; HANCKE, 2020; RUBIO; VALERO; LLOPIS-ALBERT, 2019). A desvantagem, contudo, é a dificuldade de locomoção em terrenos muito irregulares ou com muito obstáculos quando comparado com robôs com outras formas de locomoção (ALATISE; HANCKE, 2020).

### ROBÔ MÓVEL DIFERENCIAL 4WD

O robô móvel diferencial 4WD (Four Wheel Drive) utilizado neste trabalho possui quatro rodas tracionadas de mesmo tamanho e, para realizar o acionamento, são utilizados dois motores elétricos, um para cada lado do robô. Um acoplamento mecânico faz com que as rodas de um mesmo lado girem juntas, possuindo assim a mesma velocidade.

Para calcular o modelo matemático do robô 4WD diferencial, utiliza-se a ideia de que o robô é um corpo rígido (SIEGWART; NOURBAKHS; SCARAMUZZA, 2011). Conforme mostrado na Figura 1, há dois sistemas de referência, que são o sistema absoluto (em relação à origem) e o sistema relativo (onde a referência está no próprio robô). A fim de calcular a posição do robô em relação ao sistema absoluto, consideramos a origem como referência e, desta maneira, temos as coordenadas  $(x, y)$ , além do ângulo de inclinação  $\theta$ . Além disso, há ainda duas constantes que são a distância entre os eixos e o raio, representados respectivamente pelas variáveis  $b$  e  $r$ .

Figura 1 - Configuração de um robô móvel não holonômico diferencial.



Fonte: Autoria própria.

A Equações (1) e (2) mostram os cálculos das velocidades linear  $v$  e angular  $\omega$ , com respeito a velocidade das rodas de cada lado, sendo  $V_{RD}$  e  $V_{RE}$  a velocidade das rodas do lado direito e esquerdo, respectivamente. Por fim, na Equação (3) é mostrada a equação diferencial vetorial da pose do robô (AL KHATIB; JARADAT; ABDEL-HAFEZ, 2020).

$$v = \frac{V_{RD} + V_{RE}}{2} \quad (1)$$

$$\omega = \frac{V_{RD} - V_{RE}}{b} \quad (2)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \omega \end{bmatrix} \quad (3)$$

## SISTEMAS DE LOCALIZAÇÃO DE ROBÔS MÓVEIS

A localização de robôs móveis é o processo pelo qual um robô móvel consegue estimar sua posição e orientação no ambiente que está inserido (SIEGWART; NOURBAKSH; SCARAMUZZA, 2011; THRUN et al., 2001). Muitos autores classificam a localização como um problema chave para permitir que um robô móvel tenha capacidades autônomas (ALATISE; HANCKE, 2020; GOEL; ROUMELIOTIS; SUKHATME, 1999; NEMEC et al., 2019; RUBIO; VALERO; LLOPIS-ALBERT, 2019), tanto em ambientes internos quanto externos (GAO et al., 2018).

As técnicas de localização de robôs móveis podem ser classificadas em relativas e absolutas. O primeiro grupo realiza a integral de informações produzidas por sensores, como encoders e sensores inerciais, atualizando continuamente a pose do robô a partir da posição inicial. Embora baratas e com altas taxas de amostragem, os erros crescem a cada iteração. Exemplos de técnicas relativas (*dead reckoning*) incluem odometria e navegação inercial. Já as técnicas absolutas (*reference guide*) obtêm a estimativa de localização diretamente do ambiente, utilizando referências externas, como *landmarks* artificiais, *active beacons* e sinais de satélite. Embora não permitam manter um rastreamento contínuo da localização do robô em pequenas distâncias, o erro não se incrementa conforme o robô

navega (ALATISE; HANCKE, 2020; BORENSTEIN et al., 1997; CHONG; KLEEMAN, 1997).

## SISTEMAS DE COORDENADAS LOCAL E GLOBAL

O problema da localização vai além de obter uma representação global de onde o robô está, mas também leva em conta a localização relativa a obstáculos dentro de um mapa, por exemplo (RUBIO; VALERO; LLOPIS-ALBERT, 2019; SIEGWART; NOURBAKHS; SCARAMUZZA, 2011). Em outras palavras, o sistema de localização deve atender basicamente a dois tipos de demanda: uma contendo uma descrição detalhada dos arredores do robô para controle e desvio de obstáculos; a outra em uma escala maior, fornecendo a localização do robô em relação a um referencial global, como um mapa. Ambas as demandas são importantes e possuem diferentes requisitos quanto ao tratamento de incertezas. É natural utilizar um frame (sistema de coordenadas) fixado no robô para a primeira demanda e um frame fixado no mapa para a segunda demanda. Apesar de suas vantagens, nem o frame fixo no robô nem o frame global são capazes de atender, sozinhos, com qualidade as duas demandas simultaneamente (MOORE et al., 2009).

Como uma solução, MOORE et al. (2009) propõem uma alternativa que é o uso de um terceiro frame além do global e do fixo no robô. Este terceiro sistema de coordenadas é local e permite combinar a localização suave do frame fixo no robô com a localização precisa a nível global do frame global. Para isso são necessárias duas fontes de dados de localização:

- Um algoritmo de localização que computa a posição do robô em relação ao *frame* global, como MCL, GPS, Markov, dentre outros. Este vai manter a estimativa da localização do robô precisa a nível global para longas distâncias percorridas.
- Um algoritmo de estimativa de movimento do robô, que compute a posição do robô em relação ao *frame* local. E.g.: técnicas de *dead reckoning* como odometria e navegação inercial. Este vai manter uma estimativa de localização do robô suave, contínua, com alta taxa de amostragem e com boa precisão para curtas distâncias percorridas.

A estimativa do *dead reckoning* em relação ao referencial local permite que o robô tenha uma navegação suave, desviando de obstáculos com alta precisão e confiabilidade local, porém os erros acumulados a cada iteração são mantidos. Enquanto isso, a estimativa do *reference guidance* em relação ao referencial global permite compensar os erros do *dead reckoning*, mantendo uma estimativa de localização consistente a nível global. Essa compensação é realizada através de uma transformação linear entre os frames global e local.

É importante salientar que nessa abordagem o sistema de localização completo, ou seja, *dead reckoning* como fonte de uma informação de localização local, *reference guidance* como fonte de uma informação de localização global, utilizando dois referenciais, um local e um global, relacionados através de uma transformação linear de compensação, gera uma estimativa precisa tanto localmente quanto globalmente, atendendo os requisitos das tarefas de navegação local (desvio de obstáculos, controle de posição) quanto global (planejamento de caminho global).

Estas relações, utilizando um “frame auxiliar” localmente contínuo, representam os deslocamentos do robô em relação à sua posição original com base apenas em atualizações relativas. Esta técnica é muito útil para resolver questões, conforme exemplifica FOOTE (2013) no caso de uma navegação simples de um robô quando as informações de localização são ruins. Considerando o robô avançando através de obstáculos e registrando-os no mapa (na referência global). Se a localização do robô saltar abruptamente, como pode acontecer quando se utiliza GNSS ou algoritmos probabilísticos como o *Monte Carlo Localization*, os obstáculos recentemente registrados ao lado do robô podem agora ser representados intersectando o robô, apesar do fato de o robô não tenha se movido significativamente desde a última observação.

Com isso, se os obstáculos forem registrados em relação ao frame local, não haverá influência das atualizações da localização global. No entanto, o frame local acumula erros ao longo do tempo, exigindo que os dados expirem antes que o erro seja relevante. Com esta técnica, o sistema de localização global pode publicar apenas correções da localização relativa entre os frames global e local, corrigindo apenas o frame local, e, por consequência, corrigindo a posição global do robô, sem que este perceba. Essa técnica foi fundamental para trabalhos como o de MARDER-EPPSTEIN et al. (2010) e o de MEEUSSEN et al. (2010), pois permitiu que o robô passasse por portas com tolerância menor que a precisão de sua localização.

## NAVEGAÇÃO AUTÔNOMA DE ROBÔS MÓVEIS

Navegação no campo da robótica móvel pode ser definida como uma metodologia que permite guiar um robô móvel para realizar uma missão em um ambiente com obstáculos de uma maneira segura (GARCIA et al., 2009). Um sistema de navegação robusto é pré-requisito necessário para realizar qualquer tarefa útil que uma aplicação de um robô móvel exige (MARDER-EPPSTEIN et al., 2010).

Muitos trabalhos podem ser encontrados na literatura sobre navegação de robôs móveis, e geralmente os sistemas de navegação possuem vários pontos em comum, como por exemplo sistema de localização, mapeamento, planejamento de caminho, desvio de obstáculos, controle de alto nível para alcançar um determinado objetivo e tratamento de erros (BURGARD et al., 1999; CONNER; WILLIS, 2017; MATHEW; HIREMATH, 2019).

Embora muitas estruturas e sistemas de navegação de robôs móveis tenham sido propostos, poucos rivalizaram com o impacto do *ROS Navigation Stack* (MARDER-EPPSTEIN et al., 2010) na crescente indústria de robótica móvel, apresentando mais de 580 citações no Google Scholar (maio de 2022). Dez anos depois, foi proposta uma arquitetura que é a evolução da *ROS Navigation Stack*, intitulada *Navigation2* (MACENSKI et al., 2020). Para aferir a robustez da arquitetura, experimentos de navegação de longa duração, totalizando aproximadamente 60 km de distância percorrida ao longo de 23 horas por dois robôs em um ambiente dinâmico, com alta circulação de pessoas dentro de uma universidade.

## PERCEPÇÃO

A percepção do robô pode ser definida como uma forma de representar os dados dos sensores, a fim de realizar a modelagem do ambiente ao redor do robô. Esta representação pode ser realizada na forma de mapas de custo. Como as tecnologias de sensores evoluíram nos últimos anos, o uso de uma abordagem de mapas de custo em camadas consegue melhor aproveitar estas novas tecnologias, montando uma hierarquia de camadas para diferentes tipos de sensores, resoluções e taxas de aquisição. Um mapa de custos em camadas permite que um único mapa de custos seja atualizado de forma coerente por várias fontes de dados e algoritmos (LU; HERSHBERGER; SMART, 2014). Cada camada pode estender ou modificar o mapa de custos que herda e, em seguida,

encaminhar as novas informações para planejadores e controladores. (MACENSKI et al., 2020)

Dessa forma, é possível rastrear diferentes informações acerca do ambiente de forma organizada. No trabalho de MACENSKI et al. (2020), por exemplo, foram implementadas 3 camadas: *Static Layer*: usa o mapa estático, carregado do disco ou fornecido por um algoritmo de SLAM; *Inflation Layer*: infla obstáculos letais no mapa de custos com um decaimento exponencial, considerando a área ocupada pelo robô; *Spatio-Temporal Voxel Layer*: ou STVL, mantém uma grade de *voxel* volumétrica esparsa 3D temporal que decai ao longo do tempo por meio de modelos de sensores lasers e das câmeras RGBD (MACENSKI; TSAI; FEINBERG, 2020).

## PLANEJADORES E CONTROLADORES DE CAMINHO

O planejador é o algoritmo responsável por encontrar o menor caminho global entre o robô e o objetivo. Enquanto o controlador usa informações locais (dos mapas de custo) para calcular o melhor caminho local e os sinais de controle para mover o robô (MACENSKI et al., 2020). O “melhor” caminho neste caso diz respeito a desviar de obstáculos, permanecendo dentro do caminho global em direção ao objetivo.

Para navegar suavemente em ambientes altamente dinâmicos, é possível combinar o controlador local TEB (*Timed-Elastic Bands*) (RÖSMANN; HOFFMANN; BERTRAM, 2015) com o STVL para percepção 3D dinâmica. O controlador TEB é capaz de receber detecções e rastreamentos de objetos e leva estas informações em conta para calcular comandos de velocidade. Também é adequado para uso em robôs estilo diferencial, omnidirecional e *ackermann*. Há também o controlador DWB, uma implementação DWA (*Dynamic Window Approach*) (FOX; BURGARD; THRUN, 1997), que apresenta desempenho inferior em ambientes altamente dinâmicos, se comparado com o TEB (MACENSKI et al., 2020).

## RECUPERAÇÃO

Durante a navegação, falhas podem ocorrer, como o robô não conseguir prosseguir no caminho planejado. Para mitigar essas falhas, podem ser utilizados comportamentos de recuperação. Por convenção, esses comportamentos são ordenados

de forma crescente desde ações conservadoras até mais agressivas. Alguns desses comportamentos de recuperação são:

- Limpar mapas de custo: Limpa as camadas do mapa de custo em caso de falha do sistema de percepção;
- Rotacionar: Executa uma rotação sobre o próprio eixo para buscar um caminho livre quando o robô se encontra preso;
- Esperar: no caso de obstáculos temporais, como uma pessoa andando, o robô aguarda para prosseguir. Há também casos em que se aguarda mais informações dos sensores.

## ORQUESTRAÇÃO

Para organizar o comportamento de cada um dos módulos envolvidos na navegação, uma estrutura do tipo *behavior tree* (BT) foi utilizada. Seu modelo federado é estruturado de tal forma que cada nó da árvore de comportamento invoca um servidor para computar cada uma das tarefas. Cada servidor implementa uma interface de plug-in padrão para permitir que novos algoritmos ou técnicas sejam facilmente criados e selecionados em tempo de execução (MACENSKI et al., 2020).

Um comportamento geralmente é composto por uma sequência de subcomportamentos que são independentes. Os subcomportamentos podem ser projetados recursivamente, adicionando mais detalhes. Os BTs são executados de uma maneira, que permitem que o comportamento seja realizado de forma reativa. Por exemplo, suponha uma BT para pegar uma bola e posicioná-la em outro lugar, possuindo 3 subcomportamentos: “Encontrar bola”, “Pegar bola” e “Posicionar bola”. Caso durante a finalização do subcomportamento “Posicionar bola”, a bola escorregue, o robô abortará o subcomportamento atual e executará novamente o subcomportamento “Pegar bola” ou “Encontrar bola”, de acordo com a situação atual (COLLEDANCHISE; ÖGREN, 2018).

## SISTEMA GLOBAL DE NAVEGAÇÃO POR SATÉLITE

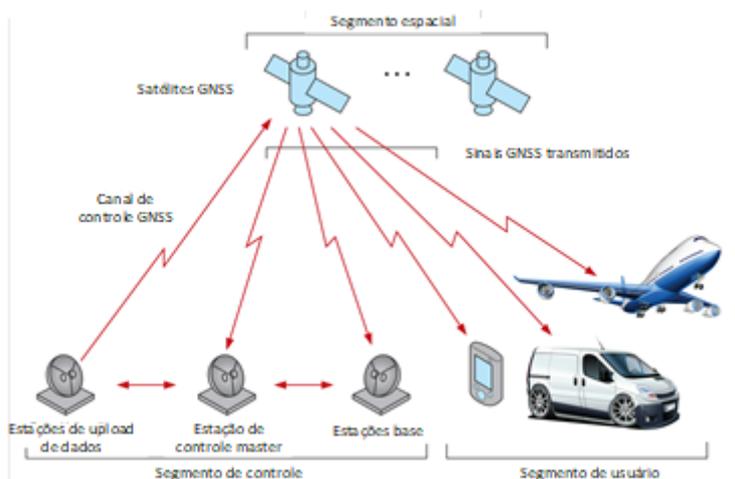
O termo GNSS, sigla para *Global Navigation Satellite System*, ou sistema global de navegação por satélite, é um termo genérico para descrever um sistema composto por uma constelação de satélites com cobertura global que fornece posicionamento sobre a

superfície terrestre, sendo o GPS o exemplo mais comumente conhecido (DE SOUSA PISSARDINI et al., 2017; IBGE, 2018; NOVATEL, 2015).

## ARQUITETURA

A arquitetura dos sistemas GNSS é composta por três segmentos: segmento espacial, segmento de controle e segmento de usuário, conforme ilustra a Figura 2.

Figura 2 - Segmentos GNSS.



Fonte: Adaptado de (NOVATEL, 2015).

O segmento espacial, formado pela constelação de satélites responsáveis por fazer a transmissão de sinais de alta precisão para fornecer informações de localização e tempo; o segmento de controle, o qual é responsável pelo monitoramento e controle da constelação de satélites através de estações de controle terrestres; e, por fim, o segmento de usuário, o qual é composto por dispositivos que recebem os sinais vindo do GNSS e calculam a posição tridimensional por meio da técnica de trilateração, considerando o offset de tempo entre o relógio do receptor e dos satélites (KAPLAN; HEGARTY, 2005; NOVATEL, 2015).

## TÉCNICAS PARA MELHORAR A PERFORMANCE DOS SISTEMAS GNSS

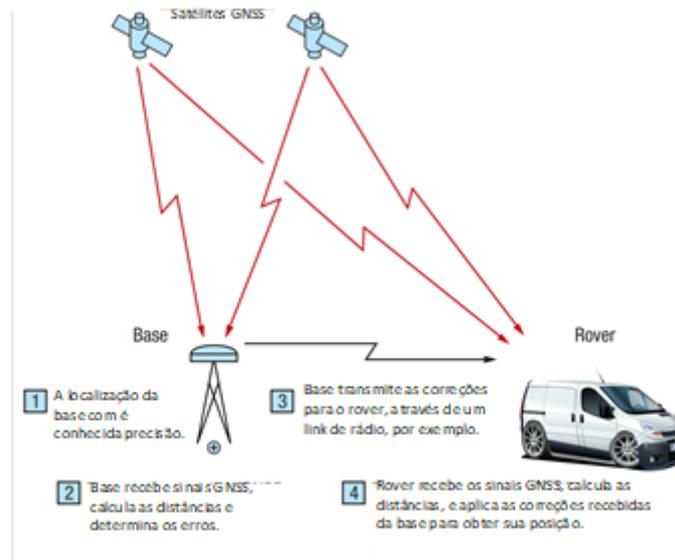
Como as medições de posicionamento utilizando sistemas GNSS possuem erros, algumas técnicas foram desenvolvidas para mitigar esses erros, tais como média de várias

medidas (método menos eficiente), modelagem do erro e predição dos valores de correção e técnicas de correções diferenciais (NOVATEL, 2015).

### DGNSS (DIFFERENTIAL GNSS)

Uma técnica comum é o uso do GNSS diferencial (DGNSS). Esta técnica consiste em um sistema fixo, com uma posição conhecida e calibrada, chamada de estação base, (ou simplesmente base). A base coleta dados de GNSS e compara com sua posição conhecida, gerando estimativa dos erros, que são enviados para outro equipamento, chamado de *rover*, que corrige sua posição medida utilizando os erros obtidos pela base. Nesse tipo de técnica, é necessário o uso de um canal de comunicação entre base e *rover*. Como os satélites estão em altitudes na ordem de milhares de quilômetros, o funcionamento desse sistema é apropriado dentro de um limite de distância entre base e *rover* de dezenas de quilômetros pois os erros entre os satélites e a base, e entre os satélites e o *rover* são semelhantes. O DGNSS utiliza medição do tempo de transmissão por código (KAPLAN; HEGARTY, 2005; NOVATEL, 2015). A Figura 3 ilustra o funcionamento do DGNSS.

Figura 3 – *Overview* do DGNSS.

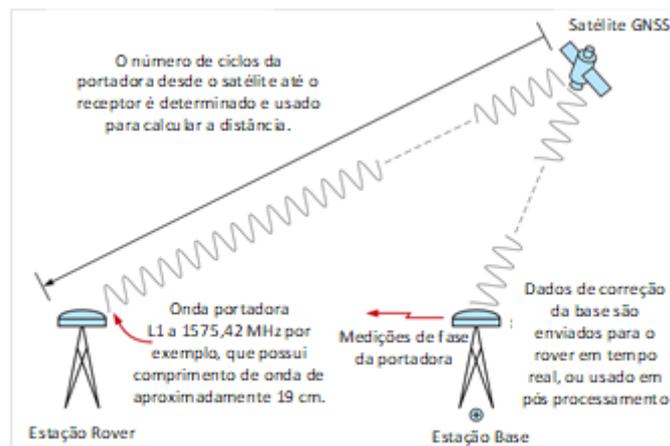


Fonte: Adaptado de (NOVATEL, 2015)

## RTK (REAL-TIME KINEMATIC)

A técnica de RTK, assim como o DGNSS, utiliza medições diferenciais entre base e *rover* para gerar correções nas medidas de distância dos satélites até os receptores. Porém, de forma simplificada, conceitualmente o que diferencia o RTK do DGNSS é a forma de medição do tempo de propagação do sinal do satélite, enquanto o DGNSS utiliza medição por código, o RTK utiliza medição por fase da portadora. Como já comentado, a principal dificuldade desse método é a resolução da ambiguidade para determinar o número de comprimento de ondas entre o satélite e o receptor. Apesar de ser um processo complexo, os receptores GNSS de alta precisão resolvem este problema quase que instantaneamente. A Figura 4 ilustra o funcionamento do RTK.

Figura 4 – Overview do RTK.



Fonte: Adaptado de (NOVATEL, 2015)

## METODOLOGIA

### PLATAFORMA ROBÓTICA

A estrutura mecânica do robô possui uma correia sincronizadora, a qual conecta as rodas de cada lado do robô, fazendo com que o comportamento do mesmo seja como um robô esteira, apesar deste possuir apenas rodas. Com isso, ele fica mais adaptado para aplicações outdoor que exigem mais robustez, além de poder ser utilizado para aplicações indoor.

O robô possui 60 cm de comprimento, 56 cm de largura e 46 cm de altura, com um peso total aproximado de 55 kg e sua base consiste em um chassi fabricado com perfis

de aço, onde foram fixados inicialmente os elementos de transmissão e os moto-redutores. A Figura 5 mostra o robô utilizado.

Figura 5 - Robô utilizado para aplicações indoor e outdoor.

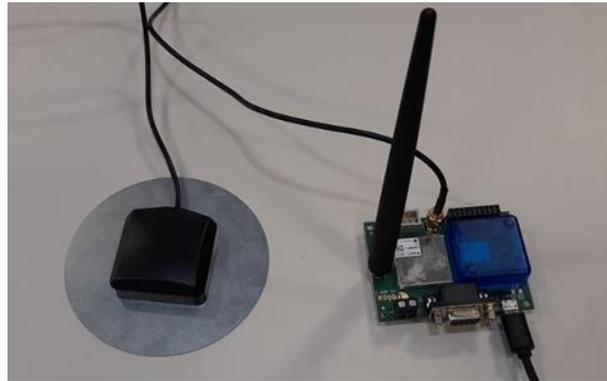


Fonte: A autoria própria.

Na instrumentação, o robô possui sensores como *encoders* (de 2000 pulsos por volta) para determinar a velocidade de cada par de rodas, além de outros sensores para visualizar o ambiente como o LiDAR modelo Sick TiM551, que possui um ângulo de leitura de 270° e alcance de até 10 metros. Para o processamento do controle de baixo nível, ou seja, para ler encoders e executar as malhas de controle das velocidades das rodas foi utilizado um sistema embarcado NI myRIO 1900. Além disso, para permitir o robô se conectar com o computador que processa o controle de alto nível e utilizar o sistema ROS, foi colocado um switch, que garante a conexão entre os dispositivos.

Como receptor GNSS, foi utilizado o kit de desenvolvimento RTK da *u-blox* C94-M8P-2, que contém duas placas de aplicação (ambas com o chip NEO-M8P-2), duas antenas UHF de 915 MHz, 2 antenas GNSS, 2 placas de terra e 2 cabos USB. Com esse kit, é possível operar em diversos modos, como um simples receptor convencional, até como base e *rover* em DGNSS ou DGNSS RTK.

Figura 6 – Metade do kit de aplicação RTK u-blox C94-M8P-2.



Fonte: Autoria própria.

## CONTROLE AUTOMÁTICO DO ACIONAMENTO DO ROBÔ

O acionamento do robô é realizado por meio de um controle da velocidade linear das rodas do robô, utilizando controladores PI (proporcional - integral) para cada um dos lados do robô, ou seja, há o controle independente das velocidades das rodas direita e esquerda, utilizando como realimentação as velocidades obtidas através da odometria das rodas.

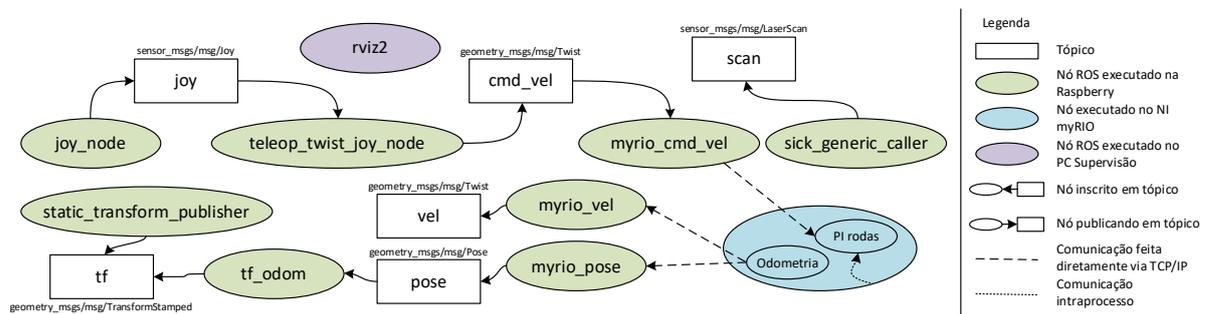
## VISÃO GERAL DO SOFTWARE

O software do robô conta com as seguintes funcionalidades de odometria calculada a partir dos encoders das rodas, controle de velocidade das rodas e de velocidade linear e angular do robô, controle remoto do acionamento via joystick, leitura do LiDAR, informação de odometria e posição do LiDAR na árvore de transformadas e supervisão via RViz. A versão utilizada do ROS 2 foi a Foxy, de modo que no minicomputador foi utilizado a versão contendo apenas os pacotes básicos, relativos às bibliotecas de comunicação, mensagens e ferramentas de linha de comando. Já no computador de supervisão, foi instalada a versão completa, contendo todos os softwares com interface gráfica para visualização.

O fluxo de informação ocorre da seguinte forma: Um comando é gerado pelo operador do robô via joystick, o programa *joy\_node* lê este comando, e o programa *teleop\_twist\_joy\_node* converte este comando do joystick em set points de velocidade linear e angular. Estes setpoints são enviados para o NI-myRIO através do programa *myrio\_cmd\_vel*. Dentro do NI-myRIO, é executada a aquisição de dados dos encoders,

calculada a odometria e executadas as malhas de controle das velocidades das rodas. A odometria é retornada ao ROS através dos executáveis *myrio\_pose* e *myrio\_vel*. O nó *tf\_odom*, insere as informações de odometria na árvore de transformadas, e a transformada que relaciona a posição do LiDAR com o centro do robô é publicada na árvore de transformadas através de um *static\_transform\_publisher*. Cada um destes programas será melhor detalhado abaixo.

Figura 7 - Diagrama dos módulos de software atuais, incluindo forma de comunicação entre eles.



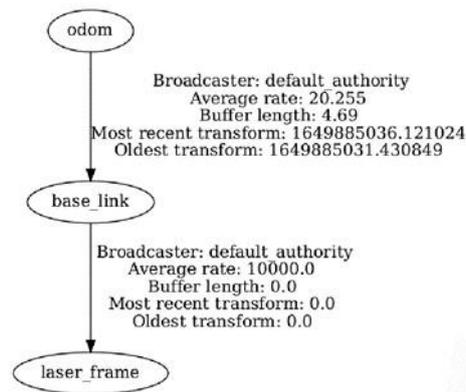
Fonte: Autoria própria.

Para a configuração do LiDAR, foi necessário configurar o IP do dispositivo para um valor fixo, dentro da sub rede utilizada. Uma vez feito isso, o pacote *sick\_scan2*, desenvolvido e disponibilizado pelo próprio fabricante, possui um programa chamado *sick\_generic\_caller*, que quando executado com os parâmetros definidos, é capaz de se conectar ao sensor e disponibilizar os dados na forma de mensagens do tipo *sensor\_msgs/msg/LaserScan*. O arquivo de parâmetro criado contém informações do modelo do sensor, endereço IP e porta para comunicação com o sensor, frame do sistema de coordenados do laser, ângulos máximo e mínimo e alcance do sensor.

## CONFIGURAÇÃO DAS TRANSFORMADAS

A árvore de transformadas foi configurada com 2 transformadas: a primeira é a inserção dos dados da odometria na árvore de transformadas, entre o frame de referência local e o frame do robô, através da transformada *odom* → *base\_link*; e a segunda é para registrar a posição do LiDAR em relação ao robô, através da transformada *base\_link* → *laser\_frame*.

Figura 8 - Árvore de transformadas do sistema atual.



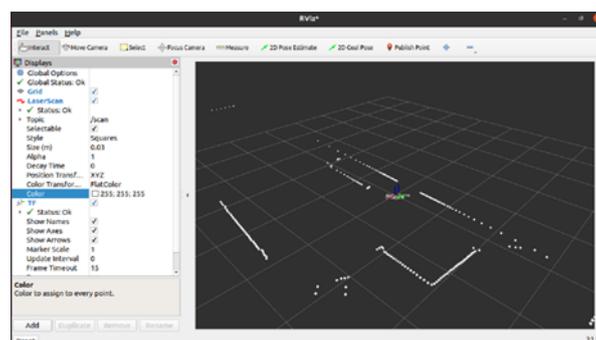
Fonte: Autoria própria.

O sistema embarcado no robô não possui suporte ao ROS. Devido a isso, foi desenvolvida uma ponte entre os dois sistemas utilizando uma conexão via socket TCP/IP. Ao todo, foram utilizadas 4 conexões, sendo duas delas transmitindo a velocidade e a pose do sistema embarcado para o ROS, uma transmitindo os setpoints de velocidade do ROS para o sistema embarcado e uma para solicitar ao sistema embarcado zerar a odometria.

## SUPERVISÃO NO RVIZ

O software RViz foi utilizado para visualizar os dados relacionados ao robô no ROS, como a nuvem de pontos medida pelo LiDAR, as transformadas publicadas pela odometria etc. A Figura 9 mostra a tela do RViz, com destaque aos pontos medidos pelo LiDAR na cor branca, enquanto a Figura 10 mostra em destaque a representação dos frames da árvore da transformada a tela de visualização do RViz.

Figura 9 – Tela do RViz atualmente.



Fonte: Autoria própria.

Figura 10 – Tela do RViz com destaque para os frames da árvore de transformadas.



Fonte: Autoria própria.

## CONFIGURAÇÃO DO GNSS RTK

Toda configuração dos receptores do kit de desenvolvimento GNSS utilizado foi executada no software disponibilizado pelo fabricante do kit, chamado *u-center*. Os receptores foram configurados para trabalhar no modo de multi-constelação, tratando sinais do GPS e do GLONASS. Esta é a configuração padrão do dispositivo. É importante conhecer a configuração utilizada, pois dependendo do sistema utilizado, é necessário alterar o tipo de mensagem a ser enviada da base para o *rover*.

## CONFIGURAÇÃO DA BASE

Após posicionar a antena que vai ligada ao receptor base, a configuração do receptor se resume em três passos. Em primeiro lugar, deve-se configurar a posição fixa, que pode ser realizada solicitando que o receptor faça o levantamento da sua localização, ou informar manualmente as coordenadas da base. Neste trabalho, foi utilizado o método de executar o levantamento automático, que consiste em configurar o levantamento com duas condições que devem ser satisfeitas para encerrar o levantamento: duração e precisão. Foi configurado para que o sistema aguarde no mínimo 600 segundos e que a precisão esteja abaixo de 1,5 metros.

O segundo passo é configurar para que o sistema transmita pelo rádio mensagens do protocolo RCTM (*Radio Technical Commission for Maritime*), que é o protocolo padrão para envio de mensagens de correção de órbita em sistemas diferenciais.

O último passo é configurar quais mensagens RCTM serão transmitidas. No caso do sistema operando com GPS e GLONASS, é necessário enviar as mensagens do tipo:

1005 (coordenadas da estação base), 1077 (observações GPS), 1087 (observações GLONASS) e 1230 (BIASes de fase de código do GLONASS).

## CONFIGURAÇÃO DO ROVER

A configuração do receptor *rover* consiste em definir que pelo rádio ele deve receber mensagens do tipo RCTM. Uma vez que as mensagens começarem a chegar, e as condições de sinal do *rover* forem boas, o dispositivo automaticamente seguirá a seguinte sequência alternando os modos de operação: 3D > 3D/DGNSS > 3D/DGNSS/FLOAT > 3D/DGNSS/FIXED.

- 3D é o modo de operação padrão, onde o receptor obtém sozinho sua posição.
- 3D/DGNSS é o modo diferencial, onde o receptor recebe dados de correção da base, mas ainda não está no modo RTK.
- 3D/DGNSS/FLOAT é o modo de operação RTK flutuante, pois o receptor ainda não resolveu todas as ambiguidades e apenas consegue estimá-las. Nesse modo a acurácia fica normalmente abaixo de 30 centímetros.
- 3D/DGNSS/FIXED é o modo de operação RTK fixo, pois o receptor conseguiu resolver todas as ambiguidades. Nesse modo a acurácia é fixada em 1 cm.

## TESTES E RESULTADOS

### TESTES COM SISTEMA GNSS RTK SEM A NAVEGAÇÃO DO ROBÔ

#### TESTES DE CURTAS DISTÂNCIAS

Foram coletados 6 pontos medidos pelo *rover*, utilizando uma barra de alumínio com 0,5 m de comprimento para certificar um distanciamento padronizado entre as antenas da base e do *rover* em cada ponto medido, conforme pode ser visto na Figura 11.

Figura 11 – Antena do receptor base (à esquerda) e rover (à direita), separados por uma barra com 0,5 m de comprimento.



Fonte: Autoria própria.

Para auxiliar nos testes, o software u-center foi configurado para exibir o valor da posição relativa entre o rover e a base. Dos seis pontos coletados, quatro foram coletados posicionando a antena do rover até que a posição relativa estivesse aproximadamente 0,5m a norte, 0,5 m a leste, -0,5m a norte e -0,5m a leste, e em seguida, a barra de alumínio era utilizada para confirmar que o posicionamento estava correto. Os outros dois pontos foram posicionados em direções arbitrárias, um na extremidade da barra (a 0,5m da base) e o outro aproximadamente no meio da barra (a 0,25m da base). A Figura 12 mostra as coordenadas dos pontos obtidos dentro do software *Google Earth Pro*, ressaltando a imagem com zoom da direita, mostrando como a distribuição das coordenadas está coerente se comparados aos valores especificados acima.

Figura 12 – Pontos coletados do rover próximos à base. À esquerda visualização mais ampla, e à direita visualização aproximada. Sentido norte apontando para cima.



Fonte: Autoria própria.

A Tabela 1 apresenta os valores obtidos para cada ponto medido, onde a coluna Distância representa a distância euclidiana calculada da base para o *rover* utilizando a posição relativa.

**Tabela 1** – Resultados obtidos com o posicionamento do receptor *rover* em modo RTK próximo à base.

Ponto	Posição relativa Norte [m]	Posição relativa Leste [m]	Distância [m]
0,5 m a Norte	0,503	-0,001	0,503
0,5 m a Leste	-0,004	0,527	0,503
-0,5 m a Leste	0,005	-0,498	0,498
-0,5 m a Norte	-0,506	0,022	0,507
0,5 m aleatório	0.395	0.311	0,503
0,25 m aleatório	0.240	0.130	0,273

Fonte:

## TESTES DE LONGAS DISTÂNCIAS

Para estes testes, foram coletados quatro pontos: três deles no estacionamento e o outro no meio do campo aberto. As coordenadas podem ser vistas na Figura 13.

Figura 13 – Pontos coletados para avaliação do posicionamento via GNSS RTK em distâncias maiores.



Fonte: Autoria própria.

A Tabela 2 apresenta os valores obtidos para cada ponto medido, onde a coluna Distância representa a distância euclidiana calculada da base para o *rover* utilizando a

posição relativa, e a coluna Distância Google Earth é a distância entre base e *rover* obtida através de ferramentas de medição de distância dentro do software *Google Earth*.

Tabela 2 – Resultados obtidos com o posicionamento do receptor *rover* em modo RTK em pontos distantes da base.

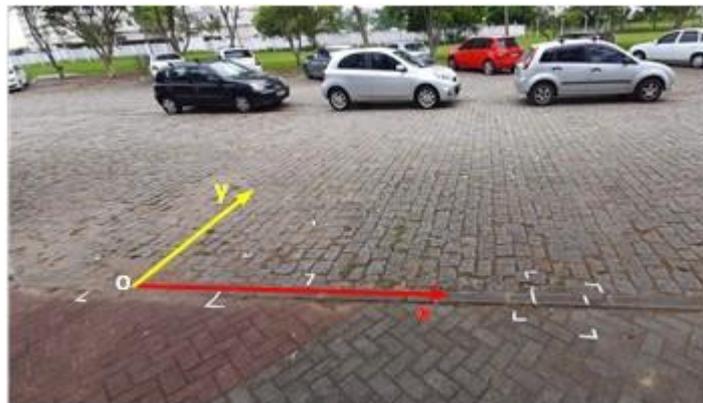
Ponto	Posição relativa Norte [m]	Posição relativa Leste [m]	Distância [m]	Distância Google Earth [m]
Bueiro 1	13,527	9,001	16,248	16,22
Bueiro 2	-17,426	32,843	37,179	37,14
Bueiro 3	-98,007	93,034	135,132	135,12
Campo aberto	-5,278	79,157	79,333	79,32

Fonte: Autoria própria

## ODOMETRIA OUTDOOR E GNSS DIFERENCIAL

Foram realizados testes em um estacionamento pavimentado com pedras, onde o robô foi teleoperado e posicionado em 10 diferentes pontos, partindo de um ponto definido como a origem do mapa, que está ilustrada na Figura 14.

Figura 14 – Local dos testes de localização outdoor, com destaque para a origem e orientação do plano cartesiano utilizado.

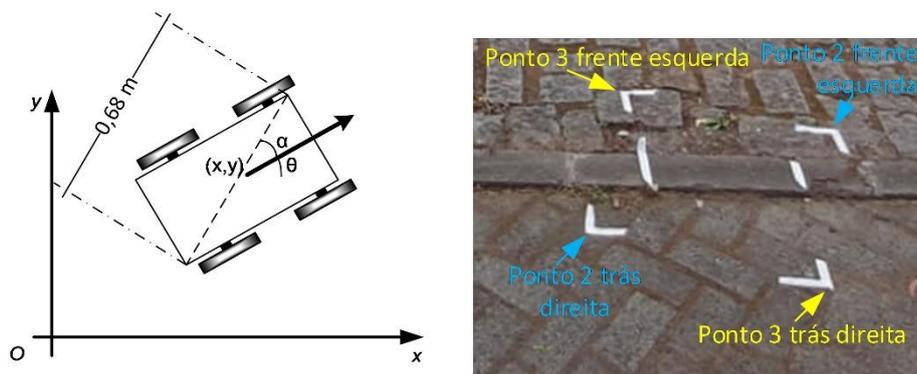


Fonte: Autoria própria.

Para obter um resultado de performance de localização, foi necessário gerar um *ground truth*, e para isso, cada um dos pontos coletados foi medido com auxílio de trenas convencionais e a laser, além de perfis de madeira para facilitar a medição em pontos mais distantes. Como estas medidas chegavam a estar na casa de mais de uma dezena de metros, sentiu-se a necessidade de efetuar uma validação das medições. Para isto, em cada

um dos 10 posicionamentos do robô, foram marcados dois pontos: o vértice traseiro direito e o vértice frontal esquerdo do robô, conforme ilustra a Figura 15. Dessa forma, é possível calcular três grandezas: posição e orientação do robô e a distância entre estes pontos, que permite efetuar a validação das medidas de posição e orientação, verificando se esta distância é similar ao valor real, que é de aproximadamente 0,68 m.

Figura 15 – Método para obtenção da posição e orientação real do robô, com validação pelas dimensões do robô. À direita destaque para como os pontos foram marcados no ambiente de testes.



Fonte: Autoria própria.

Foram coletados dados de odometria para cada ponto e da posição com base no sinal de DGNS apenas, pois a configuração do modo RTK foi realizada apenas dentro do software u-center, e não dentro do ROS 2. Para converter os dados obtidos do DGNS de latitude e longitude para o plano cartesiano utilizado, primeiramente foram obtidas as diferenças entre os valores de latitude e longitude de cada ponto em relação às coordenadas da origem, obtendo diferenciais angulares nas direções leste e norte.

Devido os testes estarem sendo realizados próximo ao nível do mar e por serem pontos próximos uns aos outros se comparado à circunferência da Terra, estas diferenças angulares foram convertidas em distâncias utilizando o raio da Terra, obtendo suas coordenadas em um plano cartesiano, com origem no ponto de origem definido nos testes, porém com eixos x e y alinhados com as direções leste e norte do planeta, respectivamente. Foi então executada uma rotação de cada um desses pontos com o ângulo obtido através da Figura 16. Dessa forma, foram obtidas as coordenadas de cada ponto coletados pelo DGNS dentro do plano cartesiano utilizado nos testes.

Figura 16 – Local dos testes de localização outdoor, com destaque para a origem e orientação do plano cartesiano utilizado definido.



Fonte: Autoria própria.

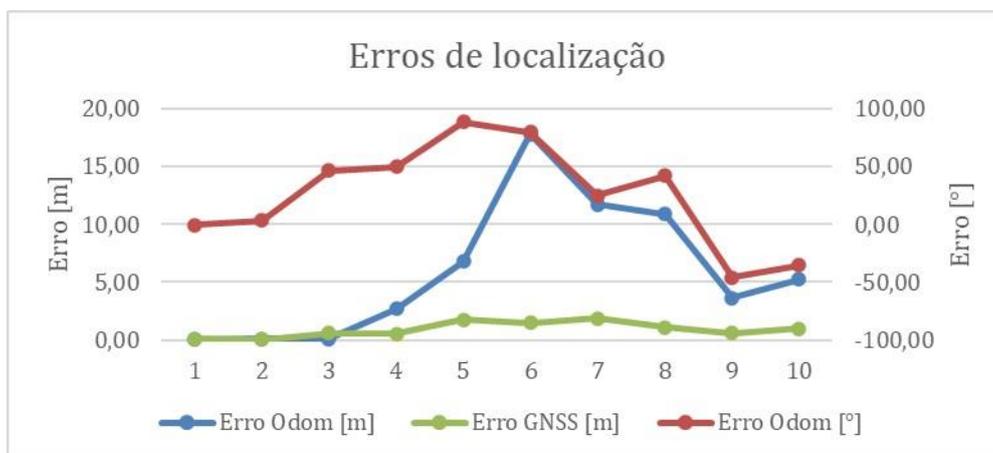
Os resultados obtidos podem ser vistos na Tabela 3, sendo que  $X_{real}$ ,  $Y_{real}$  e  $\theta_{real}$  representam a localização do *ground truth* utilizado.  $X_{Odom}$ ,  $Y_{Odom}$  e  $\theta_{Odom}$ , representam a localização obtida com a odometria  $D_{valid}$ .  $D_{valid}$  representa a distância entre os vértices do robô para validação da medição.  $X_{GNSS}$  e  $Y_{GNSS}$ , representam a posição obtida com o GNSS diferencial.  $Erro_{Odom}$  [m] representa o erro medido na forma da distância euclidiana entre a posição obtida na odometria e o *ground truth*.  $Erro_{GNSS}$  representa o erro medido na forma da distância euclidiana entre a posição obtida com o DGNSS e o *ground truth*.  $Erro_{Odom}$  [°] representa o erro da orientação obtida com a odometria em relação ao *ground truth*.

Tabela 3 – Resultados obtidos para avaliação da localização por odometria e por GNSS diferencial.

Ponto	X Real [m]	Y Real [m]	$\theta$ Real [°]	X Odom [m]	Y Odom [m]	$\theta$ Odom [°]	D valid. [m]	X GNSS [m]	Y GNSS [m]	Erro Odom [m]	Erro GNSS [m]	Erro Odom [°]
1	1,05	0,02	0,8	1,10	0,01	0,2	0,69	1,01	0,13	0,05	0,11	-0,57
2	3,16	-0,01	-1,7	3,28	-0,02	1,3	0,68	3,14	0,04	0,13	0,05	3,09
3	3,17	-0,01	90,6	3,28	-0,02	137,3	0,69	3,29	-0,62	0,11	0,63	46,66
4	3,21	3,28	95,2	0,73	2,25	145,1	0,69	2,65	3,26	2,68	0,56	49,84
5	-1,32	4,27	171,0	-0,62	-2,51	-100,7	0,67	-2,60	5,44	6,81	1,74	88,26
6	-9,02	9,27	135,6	-5,47	-8,16	-144,9	0,69	-10,46	9,57	17,78	1,47	79,46
7	3,15	13,00	-5,5	3,13	1,29	19,5	0,68	1,40	12,32	11,70	1,87	25,05
8	9,31	6,68	-153,2	8,89	-4,21	-111,0	0,67	9,75	7,66	10,89	1,08	42,29
9	0,78	0,74	-150,9	-1,53	-2,05	163,4	0,68	1,12	1,22	3,62	0,59	-45,70
10	-3,63	-6,62	-125,1	-8,86	-6,18	-160,8	0,69	-2,94	-5,92	5,26	0,97	-35,64

Fonte: Autoria própria.

Figura 17 – Resultados dos erros de posição da odometria e do GNSS diferencial e da orientação da odometria.



Fonte: Autoria própria.

Ao avaliar os resultados da odometria primeiramente, que podem ser vistos na Tabela 3, verifica-se que no ponto 2, após o robô percorrer 3m em linha reta, o erro de distância ficou próximo dos 10cm e o erro angular na casa dos 3°, o que pode ser considerado normal. Porém, do ponto 2 para o ponto 3, foi efetuada apenas uma rotação de 90° no robô sobre o seu eixo, e verificou-se que o erro angular saltou para 46,6°, que é um valor alto se pensar em um robô diferencial apenas, mas que se apresenta coerente pois o comportamento do robô utilizado assemelha-se a uma esteira, em que toda curva realizada implica em deslizamento das rodas.

O erro do DGNSS ficou abaixo dos 2m em todos os pontos, ficando na média bem próximo de 1m de erro, que se comparado à odometria já traz uma melhoria. Além disso, apesar do GNSS RTK não estar implementado para uso no robô, para avaliar a localização utilizando o GNSS RTK, foram realizados dois testes utilizando o software u-center, um para verificar a precisão em distâncias na casa dos centímetros e outro na casa de dezenas de metros. A base foi posicionada em um ponto fixo que permitisse uma visada minimamente livre para cada posicionamento do *rover*.

## APLICAÇÃO DO ALGORITMO DE LOCALIZAÇÃO GNSS RTK NO ROBÔ

Os algoritmos de localização via satélite foram configurados e executados no robô para localização global juntamente com a fusão com a IMU e odometria para localização

local e desvio de obstáculos. O sistema totalmente configurado tem aproximadamente 39 nós que se comunicam através de 64 tópicos.

Para avaliar a estrutura montada, inicialmente o robô foi posicionado em um estacionamento. Em seguida, foi solicitado ao robô para ir a um certo ponto. Depois de atingir o primeiro objetivo, foi solicitado ao robô para ir para outro ponto, no entanto, no meio do caminho um obstáculo aparece para testar a capacidade de evitar obstáculos dinamicamente. A Figura 18 mostra o ambiente de supervisão e teste, enquanto Figura 19 mostra o momento em que o robô detecta e evita o obstáculo. O resultado completo pode ser assistido em um vídeo do GAIN (2022), cujo link é (<https://youtu.be/AU1z58yum58>).

Figura 18 – Teste de navegação com o RViz e a imagem do robô sincronizados.



Fonte: Autoria própria.

Figura 19 – Teste de navegação, mostrando o RViz e a imagem do robô sincronizadas no momento de evitar o obstáculo.



Fonte: Autoria própria.

## CONCLUSÕES

O robô atende as especificações necessárias para trabalhar em ambiente outdoor, conforme pode ser visto no vídeo indicado na seção anterior. As especificações da

plataforma robótica foram resultadas de adaptações realizadas para aumentar a capacidade do robô operar em diferentes condições de terreno.

O controle das variáveis de velocidades linear e angular está sendo atualmente realizado de forma indireta. Entretanto, como observado na análise dos resultados dos erros da odometria apresentados, verifica-se que esta funciona dentro do esperado em linha reta, mas pode inserir um erro de até 50% em rotações ( $46,6^\circ$  de erro em uma rotação real de  $90^\circ$ ). Devido a característica do funcionamento do robô diferencial 4WD ser similar a uma esteira, onde toda curva implica em deslizamento das rodas, verifica-se que a velocidade angular indicada pela odometria das rodas é inferior a velocidade angular real do robô. Sendo assim, reforça-se a necessidade de se implementar a fusão sensorial entre odometria e sensores inerciais para obter uma velocidade angular, e consequentemente uma orientação mais acurada.

Como pode ser visto nos resultados, o ROS 2 foi usado para desenvolver um aplicativo robótico de navegação autônoma ao ar livre. Assim como demonstrado no vídeo do teste realizado, a implementação foi bem-sucedida, sendo que a estratégia de desenvolver aplicações de forma distribuída e isolada contribuiu para que as aplicações fossem desacopladas e organizadas. A utilização do ROS teve participação fundamental para que grande parte do código não precisasse ser reescrita novamente, mas reaproveitado por meio de pacotes disponibilizados pela própria comunidade do ROS.

Como trabalhos futuros, a fusão com o GNSS poderá ser melhor gerenciada, de acordo com a covariância do sinal, o que permitirá que o sistema de localização fique livre de falhas de correções RTK, tornando a navegação ainda mais confiável e suave. Além disso, ainda é possível melhorar o aplicativo para deixá-lo pronto para uso, ou seja, configurar o software para ser executado imediatamente após o robô ser ligado, permitindo que o usuário o utilize com o joystick ou em um aplicativo autônomo. Desta forma, diversas aplicações específicas como agricultura, vigilância, inspeção industrial etc. pode ser desenvolvidas.

## REFERÊNCIAS

AL KHATIB, Ehab I.; JARADAT, Mohammad Abdel Kareem; ABDEL-HAFEZ, Mamoun F. Low-cost reduced navigation system for mobile robot in indoor/outdoor environments. **IEEE Access**, [S. l.], v. 8, p. 25014–25026, 2020.

ALATISE, Mary B.; HANCKE, Gerhard P. A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. **IEEE Access**, [S. l.], v. 8, p. 39830–39846, 2020. DOI:

10.1109/ACCESS.2020.2975643.

BORENSTEIN, J.; EVERETT, H. R.; FENG, L.; WEHE, D. **Mobile Robot Positioning: Sensors and Techniques**. *Journal of Robotic Systems*, 1997. Disponível em: [https://deepblue.lib.umich.edu/bitstream/handle/2027.42/34938/2\\_ftp.pdf](https://deepblue.lib.umich.edu/bitstream/handle/2027.42/34938/2_ftp.pdf).

BRAÑA, Francisco-Javier. A fourth industrial revolution? Digital transformation, labor and work organization: a view from Spain. *Journal of Industrial and Business Economics*, [S. l.], v. 46, n. 3, p. 415–430, 2019.

BURGARD, Wolfram; CREMERS, Armin B.; FOX, Dieter; HÄHNEL, Dirk; LAKEMEYER, Gerhard; SCHULZ, Dirk; STEINER, Walter; THRUN, Sebastian. Experiences with an interactive museum tour-guide robot. *Artificial intelligence*, [S. l.], v. 114, n. 1–2, p. 3–55, 1999.

CHONG, Kok Seng; KLEEMAN, L. Accurate odometry and error modelling for a mobile robot. *In: PROCEEDINGS OF INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION 1997, Anais [...]*. : IEEE, 1997. p. 2783–2788. DOI: 10.1109/ROBOT.1997.606708. Disponível em: <http://ieeexplore.ieee.org/document/606708/>.

COLLEDANCHISE, Michele; ÖGREN, Petter. **Behavior Trees in Robotics and AI**. [s.l.] : CRC Press, 2018. DOI: 10.1201/9780429489105. Disponível em: <https://www.taylorfrancis.com/books/9780429950902>.

CONNER, David C.; WILLIS, Justin. Flexible Navigation: Finite state machine-based integrated navigation and control for ROS enabled robots. *In: SOUTHEASTCON 2017 2017, Anais [...]*. [s.l.: s.n.] p. 1–8.

CURTO, Belén; MORENO, Vidal. Robotics in education. *Journal of Intelligent \& Robotic Systems*, [S. l.], v. 81, n. 1, p. 3, 2016.

DE SOUSA PISSARDINI, Rodrigo; DE OLIVEIRA, Rafael Henrique; VAZ, Jhonnes Alberto; DE ALMEIDA FILHO, Flávio Guilherme Vaz; DA FONSECA JUNIOR, Edvaldo Simões. O problema do posicionamento para transporte terrestre no ambiente urbano. *Revista Brasileira de Geomática*, [S. l.], v. 5, n. 3, p. 380–403, 2017.

EUROFUND. Game changing technologies: Exploring the impact on production processes and work. [S. l.], 2018.

FOOTE, Tully. tf: The transform library. *In: 2013 IEEE CONFERENCE ON TECHNOLOGIES FOR PRACTICAL ROBOT APPLICATIONS (TEPRA) 2013, Anais [...]*. [s.l.: s.n.] p. 1–6.

FOUNTAS, Spyros; MYLONAS, Nikos; MALOUNAS, Ioannis; RODIAS, Efthymios; HELLMANN SANTOS, Christoph; PEKKERJET, Erik. Agricultural robotics for field operations. *Sensors*, [S. l.], v. 20, n. 9, p. 2672, 2020.

FOX, Dieter; BURGARD, Wolfram; THRUN, Sebastian. The dynamic window approach to collision avoidance. *IEEE Robotics \& Automation Magazine*, [S. l.], v. 4, n. 1, p. 23–33, 1997.

GAIN. **ROS 2 in the development of an autonomous robot application**. 2022. Disponível em: <https://www.youtube.com/watch?v=AUIz58yum58>. Acesso em: 1 jan. 2023.

GAO, Xinyu; LI, Jinhai; FAN, Lifeng; ZHOU, Qiao; YIN, Kaimin; WANG, Jianxu; SONG, Chao; HUANG, Lan; WANG, Zhongyi. Review of wheeled mobile robots' navigation problems and application prospects in agriculture. *Ieee Access*, [S. l.], v. 6, p. 49248–49268, 2018.

GARCIA, Elena; JIMENEZ, Maria Antonia; DE SANTOS, Pablo Gonzalez; ARMADA,

Manuel. The evolution of robotics research. **IEEE Robotics & Automation Magazine**, [S. l.], v. 14, n. 1, p. 90–103, 2007.

GARCIA, M. A. Porta; MONTIEL, Oscar; CASTILLO, Oscar; SEPULVEDA, Roberto; MELIN, Patricia. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. **Applied Soft Computing**, [S. l.], v. 9, n. 3, p. 1102–1110, 2009.

GOEL, Puneet; ROUMELIOTIS, Stergios I.; SUKHATME, Gaurav S. Robust localization using relative and absolute position estimates. *In*: PROCEEDINGS 1999 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS. HUMAN AND ENVIRONMENT FRIENDLY ROBOTS WITH HIGH INTELLIGENCE AND EMOTIONAL QUOTIENTS (CAT. NO. 99CH36289) 1999, **Anais [...]**. [s.l: s.n.] p. 1134–1140.

IBGE. **GNSS**. 2018. Disponível em: <https://atlascolar.ibge.gov.br/conceitos-gerais/o-que-e-cartografia/sistema-global-de-navegac-a-o-por-sate-litess.html>. Acesso em: 20 fev. 2022.

KAGERMANN, Henning; WAHLSTER, Wolfgang; HELBIG, Johannes. Securing the future of German manufacturing industry. **Recommendations for implementing the strategic initiative INDUSTRIE**, [S. l.], v. 4, n. 199, p. 14, 2013.

KAPLAN, Elliott; HEGARTY, Christopher. **Understanding GPS**. 2. ed. Norwood, MA: Artech House, 2005.

LU, David V; HERSHBERGER, Dave; SMART, William D. Layered costmaps for context-sensitive navigation. *In*: 2014 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS 2014, **Anais [...]**. [s.l: s.n.] p. 709–715.

LU, Yu-Hsiang; JUANG, Jih-Gau. Application of differential global positioning system and path planning to robot outdoor patrol. **Sensors and Materials**, [S. l.], v. 30, n. 8, p. 1643–1654, 2018.

MACENSKI, Steve; MARTIN, Francisco; WHITE, Ruffin; CLAVERO, Jonatan Ginés. The marathon 2: A navigation system. *In*: 2020 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS) 2020, **Anais [...]**. [s.l: s.n.] p. 2718–2725.

MACENSKI, Steve; TSAI, David; FEINBERG, Max. Spatio-temporal voxel layer: A view on robot perception for the dynamic world. **International Journal of Advanced Robotic Systems**, [S. l.], v. 17, n. 2, p. 1729881420910530, 2020.

MARDER-EPPSTEIN, Eitan; BERGER, Eric; FOOTE, Tully; GERKEY, Brian; KONOLIGE, Kurt. The office marathon: Robust navigation in an indoor office environment. *In*: 2010 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION 2010, **Anais [...]**. [s.l: s.n.] p. 300–307.

MATHEW, Robins; HIREMATH, Somashekhar S. Reinforcement learning based approach for mobile robot navigation. *In*: 2019 INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND KNOWLEDGE ECONOMY (ICCIKE) 2019, **Anais [...]**. [s.l: s.n.] p. 523–526.

MEEUSSEN, Wim et al. Autonomous door opening and plugging in with a personal robot. *In*: 2010 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION 2010, **Anais [...]**. [s.l: s.n.] p. 729–736.

MOORE, David C.; HUANG, Albert S.; WALTER, Matthew; OLSON, Edwin; FLETCHER, Luke; LEONARD, John; TELLER, Seth. Simultaneous local and global state estimation for robotic navigation. *In*: 2009 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND

AUTOMATION 2009, **Anais** [...]. [s.l: s.n.] p. 3794–3799.

NEMEC, Dušan; ŠIMÁK, Vojtech; JANOTA, Aleš; HRUBOŠ, Marián; BUBEN\`IKOVÁ, Em\`ilia. Precise localization of the mobile wheeled robot using sensor fusion of odometry, visual artificial landmarks and inertial sensors. **Robotics and Autonomous Systems**, [S. l.], v. 112, p. 168–177, 2019.

NOVATEL. **An Introduction to GNSS**. 2. ed. [s.l.] : NovAtel Inc, 2015.

PARK, Dong Il; PARK, Chanhun; DO, Hyunmin; CHOI, Taeyong; KYUNG, Jinho. Development of dual arm robot platform for automatic assembly. *In: 2014 14TH INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION AND SYSTEMS (ICCAS 2014) 2014*, **Anais** [...]. [s.l: s.n.] p. 319–321.

PERMINOV, Stepan; MIKHAILOVSKIY, Nikita; SEDUNIN, Alexander; OKUNEVICH, Iaroslav; KALINOV, Ivan; KURENKOV, Mikhail; TSETSERUKOU, Dzmitry. Ultrabot: Autonomous mobile robot for indoor uv-c disinfection. *In: 2021 IEEE 17TH INTERNATIONAL CONFERENCE ON AUTOMATION SCIENCE AND ENGINEERING (CASE) 2021*, **Anais** [...]. [s.l: s.n.] p. 2147–2152.

RÖSMANN, Christoph; HOFFMANN, Frank; BERTRAM, Torsten. Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control. *In: 2015 EUROPEAN CONTROL CONFERENCE (ECC) 2015*, **Anais** [...]. [s.l: s.n.] p. 3352–3357.

RUAN, Kaicheng; WU, Zehao; XU, Qingsong. Smart cleaner: A new autonomous indoor disinfection robot for combating the covid-19 pandemic. **Robotics**, [S. l.], v. 10, n. 3, p. 87, 2021.

RUBIO, Francisco; VALERO, Francisco; LLOPIS-ALBERT, Carlos. A review of mobile robots: Concepts, methods, theoretical framework, and applications. **International Journal of Advanced Robotic Systems**, [S. l.], v. 16, n. 2, p. 1729881419839596, 2019.

RUNCIMAN, Mark; DARZI, Ara; MYLONAS, George P. Soft robotics in minimally invasive surgery. **Soft robotics**, [S. l.], v. 6, n. 4, p. 423–443, 2019.

SIEGWART, Roland; NOURBAKSH, Illah Reza; SCARAMUZZA, Davide. **Introduction to autonomous mobile robots**. [s.l.] : MIT press, 2011.

THRUN, Sebastian; FOX, Dieter; BURGARD, Wolfram; DELLAERT, Frank. Robust Monte Carlo localization for mobile robots. **Artificial Intelligence**, [S. l.], v. 128, n. 1–2, p. 99–141, 2001. DOI: 10.1016/S0004-3702(01)00069-8. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0004370201000698>.

VEGA, Jesús Elias Miranda; CHAIDEZ, Anastacio González; GARC\`IA, Cuauhtémoc Mariscal; LÓPEZ, Moisés Rivas; FUENTES, Wendy Flores; SERGIYENKO, Oleg. Recognition system by using machine vision tools and machine learning techniques for mobile robots. *In: Examining Optoelectronics in Machine Vision and Applications in Industry 4.0*. [s.l.] : IGI Global, 2021. p. 258–287.

VOUGIOUKAS, S. Annual review of control, robotics, and autonomous systems. **Agricultural robotics**, [S. l.], v. 2, n. 1, p. 365–392, 2019.

ZHANG, Chao; ZHAN, Quanzhong; WANG, Qi; WU, Haichao; HE, Ting; AN, Yi. Autonomous dam surveillance robot system based on multi-sensor fusion. **Sensors**, [S. l.], v. 20, n. 4, p. 1097, 2020.