

Usando a Plataforma LIMO para Aprender Robótica

Maurício Bittencourt Pimenta*, Fellipe Andrade Spagnol*, Luiz Miguel M. N. P. Tavares*

Wolmar Araujo Neto*, Mário Sarcinelli-Filho* and Daniel Khede Dourado Villa*

*Depto. de Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, ES, Brasil

E-mails: {mauricio.pimenta,fellipe.spagnol,luiz.m.tavares}@edu.ufes.br, {wolmar.araujo,mario.sarcinelli,daniel.villa}@ufes.br

Resumo—Este artigo apresenta o uso do robô terrestre LIMO para pesquisa e aplicações em robótica móvel. LIMO é um robô de código aberto, economicamente acessível, e de pequeno porte, ideal para espaços de trabalho limitados. Neste trabalho são apresentadas as principais características da plataforma robótica, bem como são demonstrados vários exemplos de uso da plataforma, em tarefas de rastreamento de trajetória, sistemas multirrobôs e navegação em ambientes complexos. O objetivo deste trabalho é mostrar o potencial do LIMO para facilitar o aprendizado ativo em robótica, por oferecer uma solução prática e acessível para estudantes e pesquisadores.

Palavras-chave — Robótica educacional, Plataforma robótica LIMO, Aprendizado ativo, Integração sensorial.

I. INTRODUÇÃO

O robô terrestre LIMO da fabricante AgileX se mostrou uma plataforma poderosa para ensino em robótica móvel. Este se destaca por possuir quatro modos de operação, a saber, diferencial, com esteiras, omnidirecional e por direção Ackermann (“car-like”), conforme pode ser visto na Figura 1. Além disso, o robô é equipado com sistemas a bordo sofisticados, como LiDAR, câmeras de profundidade, e computador de bordo de alta performance. Estas características tornam este robô interessante a laboratórios que buscam implementar técnicas de controle avançado e desenvolver algoritmos inteligentes para o controlar robôs de forma autônoma.

Neste contexto, as principais contribuições deste artigo são a explicação clara sobre como configurar os modos de operação do robô terrestre LIMO, apresentar características básicas de utilização do robô através da proposição de controladores cinemáticos, e apresentar exemplos de aplicações desta plataforma. Para isso o artigo prossegue com a Seção II, que detalha as principais características da plataforma LIMO, a Seção III, que discute estratégias de controle comuns para navegação autônoma em cada modo de operação do LIMO e a Seção IV, que apresenta os resultados obtidos a partir de

várias aplicações usando esta plataforma robótica. Finalmente, a Seção V fornece algumas considerações finais.

II. A PLATAFORMA ROBÓTICA AGILEX LIMO

LIMO, da AgileX Robotics, é um robô terrestre elétrico que se mostra uma plataforma de aprendizado adaptável a uma ampla gama de cenários. Ele pode ser configurado para quatro modos de operação, abrindo assim várias possibilidades de aplicação. Esta seção discute a estrutura do LIMO, seu hardware e seu software. Informações mais detalhadas podem ser encontradas no Manual do Usuário [1].

A. Hardware

O robô LIMO possui boa capacidade computacional a bordo e sensores adequados para estudos relacionados a navegação autônoma, SLAM, mapeamento, planejamento de trajetórias, desvio de obstáculos, etc. Sua bateria de 12V e 5200mAh fornece autonomia de 40 min de funcionamento ativo e até 2 horas em *stand-by*. A Tabela I mostra uma lista de alguns sensores/características desse robô.

A placa principal, uma Nvidia Jetson Nano, é um computador de pequeno porte poderoso capaz de suportar aplicações diversas assim como aplicações de IA de nível básico. Ela possui uma CPU Quad-Core ARM57 de 64 bits rodando a 1,43 GHz, 4 GB de memória RAM LPDDR4 e uma GPU Nvidia Maxwell de 128 núcleos, capaz de decodificação de vídeo em resolução 4K a 60 quadros por segundo. A placa possui um conector Ethernet Gigabit para acesso rápido à internet, saída HDMI, 4 portas USB 3.0, que permitem a conexão de vários dispositivos, e uma porta USB 2.0 Micro-B para alimentação.

Como mostra a Tabela I, o LIMO possui sensores úteis para percepção ambiental e mapeamento, como um LiDAR e uma câmera RGBD. O LiDAR, um EAI X2L, é um sensor laser de varredura bidimensional de 360° que opera com frequência



Fig. 1: Robôs LIMO disponíveis no Laboratório de Automação Inteligente e Robótica da UFES (LAB-AIR/UFES). Da esquerda para a direita: omnidirecional, com esteira e Ackermann/diferencial.

TABELA I: Lista de sensores/características do robô LIMO.

Item	Descrição
Sensor laser	LIDAR EAI X2L
Câmera de profundidade	DaBai Astra ou RealSense D435
Computador de bordo	Placa NVIDIA Jetson Nano (4G)
Módulo de voz	Assistente de voz iFlytek/Assistente Google
Alto falantes	Esquerdo e direito, dois canais (2x2W)
Hub USB	Uma porta tipo C mais duas 2.0
Display frontal	Tela OLED branca de 1.54 polegadas, 128x64
Display traseiro	7 polegadas, tela de toque IPS 1024x600
Comunicação	Wi-Fi e Bluetooth 5.0
Bateria	12 V, 5200mAh

de varredura de 3 kHz e frequência de escaneamento de 6 Hz, com um alcance de até 8 metros em ambientes internos. A câmera de profundidade ORBBEC ASTRA+ utiliza tecnologia de imagem 3D por luz estruturada binocular. Ela contém uma câmera RGB e uma câmera infravermelha, e pode medir profundidades de 0,3 a 3 metros, com uma precisão de 6 mm na faixa de 1 m e um atraso de 30 a 45 ms. Sua resolução máxima de mapa de profundidade é de 640x400 pixels e sua resolução máxima de mapa de cores é de 1920x1080 pixels, ambos a 30 quadros por segundo.

A Unidade de Medição Inercial (IMU) HI226 de 6 graus de liberdade (6DoF) embarcada no LIMO combina um acelerômetro triaxial, um giroscópio triaxial e um microprocessador, permitindo adquirir dados de movimento em três eixos (X, Y e Z) com alta precisão ($\pm 2000^\circ/s$ para o giroscópio e até $\pm 8g$ para o acelerômetro), fornecendo excelente desempenho em diversas condições operacionais. Ela funciona à frequência máxima de saída, 100Hz, e utiliza algoritmos avançados de fusão de dados, podendo fornecer, simultaneamente, velocidades angulares e acelerações nos três eixos, quatérnions e ângulos de Euler, com pouca distorção de fase. Os erros (RMS) de inclinação e guinada são de $0,8^\circ$ estáticos e $2,5^\circ$ dinâmicos.

O chassi do LIMO é feito de chapas metálicas, pesa cerca de 4,8 kg e suporta até 4 kg de carga útil. Além disso, essa estrutura pode ser expandida para integrar uma unidade de processamento mais poderosa ou outros dispositivos de sensoriamento. Ou seja, o hardware nativo do LIMO é expansível, o que o torna ainda mais interessante para pesquisa/aplicações. O robô também conta com um módulo Bluetooth 5.0 embutido para conexão com um aplicativo móvel disponível para Android. O LIMO e a placa Jetson Nano estão diretamente conectados por meio de uma interface UART, e a placa pode controlar o chassi através dessa interface. O hub USB fornece 2 interfaces USB e 1 interface Tipo C, todas operando sob o protocolo USB2.0. A tela traseira é conectada ao hub USB através da interface USB2.0 e possui função de toque.

B. Software

A placa Nvidia Jetson Nano executa o Ubuntu 18.04, e utiliza o SDK Nvidia JetPack, que contém bibliotecas de aceleração para aprendizado profundo, visão computacional, gráficos e multimídia. O driver do LIMO, responsável pela comunicação entre o computador de bordo e o chassi, foi desenvolvido utilizando o sistema ROS (Robot Operating System), o que facilita a interação com outros sistemas ROS ao realizar tarefas de forma autônoma. Ele possui duas versões, a versão em C++ e a versão em Python, ambas podendo controlar o movimento do robô. A versão em Python está disponível no Python Package Index (PyPI) [2], e pode ser baixada usando o comando `pip`. A versão em C++ está disponível no Github. [3]

C. Pacotes ROS e Launchers

A Agilix fornece pacotes do LIMO contendo o *driver* C++, arquivos de configuração, *launchers* ROS e modelos

URDF em seu repositório oficial no GitHub [4] para ambas as versões do ROS [3], [5]. O pacote `limo_base` contém o *driver* em C++, que oferece a interface principal necessária para comunicar com o robô, obter informações de odometria e enviar comandos de velocidade. O *launcher* ROS para inicializar o driver é o `limo_base.launcher`.

O pacote `limo_bringup` oferece diversos *launchers* para inicializar os recursos disponíveis no robô. O *launcher* `limo_start.launch` inicializa o `limo_base` e o LiDAR, além de definir a árvore de transformadas entre a base do robô, a câmera, a IMU e o laser. Este pacote também contém outros *launchers* para execução de algoritmos de mapeamento, navegação, visualização, entre outros.

D. Aplicativo Android/iPhone para o LIMO

O aplicativo Nexus¹ é uma aplicação móvel projetada para gerenciar as configurações cinemáticas do LIMO e controlar seu movimento por meio de conexão Bluetooth. O aplicativo permite que os usuários alternem entre os diferentes modos de operação e calibrem o direcionamento das rodas quando o robô estiver na configuração Ackermann. Além disso, o aplicativo permite que os usuários definam as velocidades máximas e mínimas de lineares e angulares para cada configuração cinemática do LIMO.

E. Alternando os modos de operação

Uma característica importante do LIMO é que ele aceita quatro configurações cinemáticas diferentes, a saber, diferencial (com quatro rodas), com esteira, omnidirecional com rodas Mecanum e direção Ackermann (semelhante a carro). A plataforma LIMO possui um design inovador que permite a troca fácil entre seus modos de operação. Existem duas travas vermelhas na parte frontal do robô, as quais permitem selecionar o modo de operação. Para alternar para o modo Ackermann, basta levantar as travas de ambos os lados e girá-las 30 graus no sentido horário, tal que a linha mais longa em ambas as travas aponte para a frente do veículo, como mostrado no lado esquerdo da Figura 2. Quando na posição correta, as travas devem ficar presas, e as luzes do veículo ficarão verdes, permanecendo verdes enquanto ele estiver operando no modo Ackermann.

Para alternar para o modo diferencial de quatro rodas, levantam-se as duas travas e giram-se 30 graus no sentido horário, até que a linha mais curta nas travas aponte para a frente do corpo do veículo, como no lado direito da Figura 2. Nesse momento as travas estarão na posição de inserção. Então, deve-se ajustar o ângulo das rodas, para alinhá-las com

¹O link para baixar o aplicativo está disponível na documentação do LIMO.



Fig. 2: Posições das travas nos modos Ackermann (esquerda) e diferencial (direita).

a frente do veículo de forma que a trava possa ser inserida. Quando na posição correta, a luz do veículo deve ficar amarela, indicando que ele está operando no modo diferencial.

Para configurar o modo com esteira, o veículo deve estar no modo diferencial, e as esteiras devem ser acopladas às rodas. Recomenda-se que a esteira seja inserida primeiro na roda traseira, e as duas portas laterais do chassi devem ser levantadas, para evitar arranhões.

Para usar o modo omnidirecional de rodas Mecanum, primeiro removem-se as capas de cubo e os pneus, deixando apenas o motor do cubo (ver a Figura 3). Em seguida, deve-se garantir que os roletes de cada roda Mecanum estejam voltados para o centro do corpo. Então, instalam-se as rodas Mecanum com os parafusos M3.5 fornecidos no pacote, e ajusta-se esse modo de operação com o controle remoto/aplicativo.

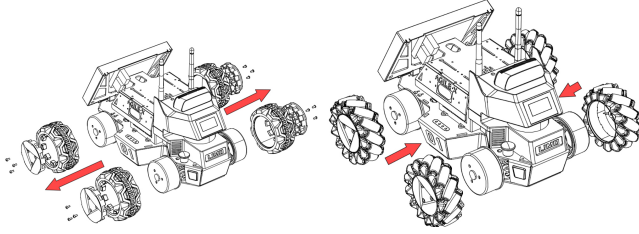


Fig. 3: Instalando as rodas Mecanum (certificar-se que cada roda seja instalada conforme acima.)

A Tabela II resume as cores das luzes do veículo para cada configuração.

TABELA II: Cor da luz e status das travas por configuração

Status das travas	Cor da luz	Configuração
Empurrada para baixo	Amarela	Diferencial ou com esteira
	Azul	Omnidirecional Mecanum
Puxada para cima	Verde	Ackermann

III. NAVEGAÇÃO AUTÔNOMA

Um controlador cinemático para cada modo de operação do LIMO é apresentado a seguir, aproveitando os controladores de baixo nível incorporados no robô, que convertem comandos de velocidade — como velocidade longitudinal ou velocidade angular — em comandos de rotação das rodas. Consequentemente, para cada modo de operação o sinal de controle de alto nível considerado é um vetor de velocidades. Embora uma compensação dinâmica possa ser empregada para aprimorar o desempenho obtido aplicando-se esses comandos de alto nível, este trabalho prioriza a simplicidade, e por isso se limita à modelagem cinemática para controle, evitando a complexidade da modelagem e da compensação dinâmicas. Isso pode ser feito sem perda significativa de desempenho quando operando com baixa aceleração, que é o caso aqui.

A. Modo diferencial

Robôs de tração diferencial são caracterizados por duas rodas motrizes que podem desenvolver rotações diferentes. Também há casos, como o LIMO, de duas rodas de um mesmo

lado acionadas por um motor, obtendo-se a configuração diferencial com quatro rodas. Quando as rodas em lados opostos giram na mesma direção o robô avança com uma velocidade linear (v), e quando elas giram em direções opostas se obtém uma velocidade angular (ω). Tais velocidades podem ser obtidas em termos das rotações das rodas esquerda (ω_L) e direita (ω_R) por

$$v = \frac{(\omega_L + \omega_R)r_w}{2}, \quad \omega = \frac{(\omega_R - \omega_L)r_w}{b}, \quad (1)$$

em que r_w é o raio da roda e b é a distância entre as rodas de um lado e as do outro. Para controlar as rotações das rodas da esquerda e da direita, sinais PWM são enviados para os controladores eletrônicos de velocidade (ESCs) montados em cada um dos dois motores. No entanto, o robô LIMO no modo diferencial é comandado por sinais de controle de alto nível, a saber $u = [v \ \omega]^T$. Portanto, os sinais de controle PWM e a dinâmica do veículo são geridos pelo controlador de baixo nível disponível a bordo. Isso permite o controle das velocidades linear e angular do veículo usando sinais de controle de alto nível, com uma perda de desempenho mínima para movimentos de baixa aceleração.

Tal controlador cinemático se baseia no modelo cinemático de um robô terrestre diferencial, a saber, $\dot{x}_c = H u$ (ver a Figura 4) [6], ou seja,

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} = \begin{bmatrix} c_\psi & -a s_\psi \\ s_\psi & a c_\psi \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2)$$

em que $c(\cdot)$ e $s(\cdot)$ representam \cos e \sin , respectivamente. O controlador cinemático é obtido diretamente pela cinemática inversa [7], tal que $u = H^{-1} \dot{x}_{c,ref}$, ou seja,

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} c_\psi & s_\psi \\ -\frac{1}{a} s_\psi & \frac{1}{a} c_\psi \end{bmatrix} \begin{bmatrix} \dot{x}_{c,ref} \\ \dot{y}_{c,ref} \end{bmatrix}. \quad (3)$$

Observe-se que a mudança de variáveis x para x_c permite que H seja invertível. A velocidade de referência $\dot{x}_{c,ref}$ é uma lei de controle de *feedforward* mais *feedback* proporcional, tal que

$$\dot{x}_{c,ref} = \dot{x}_{c,des} + \kappa(x_{c,des} - x_c), \quad (4)$$

em que κ é uma matriz diagonal definida positiva.

Como observação, no pacote ROS `limo_base` fornecido pelo fabricante os comandos de velocidade linear v e angular

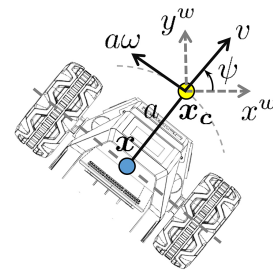


Fig. 4: Cinemática 2D de um robô diferencial. A posição do robô, x , é deslocada por um valor a . Isso resulta em um novo vetor de posição x_c , mudança essa que simplifica a implementação do controlador cinemático.

ω são publicados no tópic `cmd_vel`, devendo ser atribuídas a `linear.X` e `angular.Z`, respectivamente.

B. Modo com esteira

Quando opera na configuração com esteira o LIMO é controlado da mesma maneira que no modo diferencial. Porém, o modo com esteira oferece maior estabilidade e tração, sendo mais adequado para terrenos irregulares ou superfícies com atrito variável. Não é necessário alterar as rodas para preparar o LIMO para o modo com esteira, nem configurar o ROS. Basta sobrepor as esteiras às rodas, como ilustra a Figura 1, e enviar os mesmos sinais de controle obtidos a partir de (3), atribuindo-se as velocidades linear e angular a `linear.X` e `angular.Z`, respectivamente.

C. Modo omnidirecional

Para ativar o modo omnidirecional as rodas padrão devem ser substituídas pelas rodas Mecanum (ver a Figura 3). Além disso, são necessárias duas outras ações para operar neste modo: (i) habilitar o “modo Mecanum” usando o aplicativo Nexus, conforme descrito na Subseção II-D (esse modo só pode ser ativado via aplicativo Nexus, não sendo possível habilitá-lo diretamente pelo ROS), e (ii) modificar o arquivo `limo_base.launch` ajustando o argumento `use_mecanum` para `true` enquanto estiver no modo omnidirecional e para `false` nos outros modos.

No modo omnidirecional os movimentos longitudinais e laterais são independentes da orientação do veículo. Para distinguir os estados neste modo, usamos o subscrito ‘o’ para se referir aos estados, medidos em relação ao centro de massa do robô omnidirecional. Ao contrário do modo diferencial, o deslocamento de posição a não é utilizado no modo omnidirecional, mas o vetor de estados é ampliado, para incluir a orientação ψ , tornando-se $\mathbf{x}_o = [x \ y \ \psi]^T$.

Neste modo de operação, a entrada de controle consiste nas velocidades lineares longitudinal e lateral, respectivamente \dot{x}^b e \dot{y}^b , e na velocidade angular ω . Portanto, o modelo cinemático para o modo omnidirecional, obtido a partir da Figura 5, é $\dot{\mathbf{x}}_o = \mathbf{H}_o \mathbf{u}_o$ [6], ou

$$\mathbf{x}_o = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}^b \\ \dot{y}^b \\ \omega \end{bmatrix}. \quad (5)$$

O controlador cinemático é obtido usando-se a cinemática inversa, tal que $\mathbf{u}_o = \mathbf{H}_o^{-1} \dot{\mathbf{x}}_{o,ref}$ [7], ou

$$\begin{bmatrix} \dot{x}^b \\ \dot{y}^b \\ \omega \end{bmatrix} = \begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_{ref} \\ \dot{y}_{ref} \\ \dot{\psi}_{ref} \end{bmatrix}, \quad (6)$$

em que $\dot{\mathbf{x}}_{o,ref}$ pode ser como em (4).

D. Configuração Ackermann

Para modelar a configuração Ackermann, em vez de considerar o modelo cinemático completo de um robô Ackermann com quatro rodas, normalmente se adota uma abordagem simplificada, considerando um modelo de duas rodas, onde

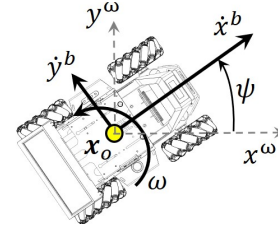


Fig. 5: Cinemática do robô LIMO omnidirecional.

ambas as rodas estão localizadas ao longo do eixo central do robô. Essa aproximação é chamada modelo cinemático de bicicleta [6]. Sabe-se que as rodas esquerda e direita de um veículo com direção Ackermann devem girar a diferentes velocidades durante uma curva, o que exige ângulos de direção diferentes para cada roda. A estrutura mecânica da configuração Ackermann aborda essa diferença entre os ângulos de direção, de forma que o modelo cinemático de bicicleta, ilustrado na Figura 6, fornece uma aproximação factível para esse veículo.

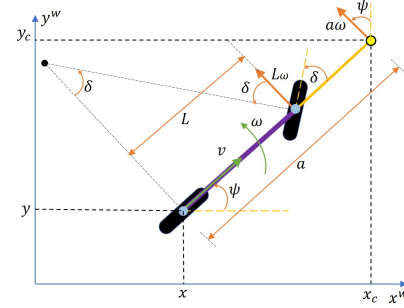


Fig. 6: Cinemática de um robô tipo carro simplificado como modelo cinemático de bicicleta. O vetor de posição do eixo traseiro, \mathbf{x} , é deslocado por uma distância a , resultando em um novo vetor de posição \mathbf{x}_c , como no caso diferencial.

A partir da Figura 6 obtém-se a equação

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} = \begin{bmatrix} c_\psi & -as_\psi \\ s_\psi & ac_\psi \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (7)$$

a partir da qual o mesmo controlador caracterizado nas equações (3) e (4) é obtido. No entanto, o sinal de controle de alto nível para o modo de direção Ackermann é $\mathbf{u}_a = [v \ \delta]^T$, em que v é a velocidade linear, já obtida nas equações (3) e (4), e δ é o ângulo de direção. Para obter δ , considera-se a restrição não-holônoma aplicada à roda dianteira, como ilustrado na Figura 6, ou seja,

$$L\omega \cos(\delta) - v \cos\left(\frac{\pi}{2} - \delta\right) = L\omega \cos(\delta) - v \sin(\delta) = 0, \quad (8)$$

resultando em $\delta = \arctan\left(\frac{L\omega}{v}\right)$, em que $L = 0.2 \text{ m}$ é a distância entre as rodas traseiras e dianteiras. Com isso se obtém o sinal de controle \mathbf{u}_a para o modo Ackermann do LIMO. O raio de giro mínimo, no modo Ackermann, é 0.37 m , e corresponde ao ângulo de direção máximo de 0.49 rad ou 28° .

A Tabela III resume os sinais de controle e seus respectivos valores limite, para cada configuração do robô LIMO.

TABELA III: Entradas de controle e seus respectivos limites

Configuração	Entradas de controle	Limites
Diferencial/Esteira	$\mathbf{u} = [v \ \omega]^T$	$\mathbf{u} = [\pm 1 \frac{m}{s} \ \pm 3.14 \frac{rad}{s}]^T$
Omnidirecional	$\mathbf{u}_o = [\dot{x}^b \ \dot{y}^b \ \omega]^T$	$\mathbf{u}_o = [\pm 1 \frac{m}{s} \ \pm 1 \frac{m}{s} \ \pm 3.14 \frac{rad}{s}]^T$
Ackermann	$\mathbf{u}_a = [v \ \delta]^T$	$\mathbf{u}_a = [\pm 1 \frac{m}{s} \ \pm 0.49 rad]^T$

IV. APLICAÇÕES

A. A arena de testes

Os experimentos foram realizados no LAB-AIR, UFES, que possui um sistema de captura de movimento *OptiTrack* para rastrear a posição e a orientação dos robôs. Um PC Linux rodando ROS *melodic* estabelece a comunicação, via Wi-Fi, entre o computador de controle e os robôs, com os sinais de controle enviados a uma taxa de 30 Hz.

B. Rastreamento de trajetória

Na primeira aplicação o robô LIMO, em cada configuração, cumpre uma tarefa de rastreamento de trajetória. Os controladores apresentados na Seção III foram usados para rastrear uma trajetória em forma de lemniscata, caracterizada por $\mathbf{x}_d = [r_x \sin \frac{2\pi t}{T} \ r_y \sin \frac{4\pi t}{T}]^T$, com $r_x = 2.0 \text{ m}$, $r_y = 1.5 \text{ m}$, e $T = 25 \text{ s}$. Adotando a matriz de ganho $\mathbf{\kappa} = \text{diag}(1, 1)$, e considerando que o robô começa a se mover a partir da posição inicial $\mathbf{x} = [0 \ 0]^T$, com uma orientação de 0 rad , os resultados obtidos estão sintetizados na Figura 7, que mostra a vista superior dos percursos desejado e de cada configuração do LIMO, além das normas dos erros de rastreamento.

Observa-se um desempenho de rastreamento comparável entre os modos de operação na Figura 7, com exceção do modo omnidirecional. Durante o experimento, foi percebido deslizamento excessivo das rodas Mecanum, o que explica o desempenho aquém para este modo. Devido a complexidade dinâmica deste modo de direção, um controlador puramente cinemático apresenta desempenho inferior comparado aos demais modos presentes no LIMO.

Vídeos demonstrando o experimento podem ser encontrados em <https://youtu.be/1LwW6xIPjxM> e <https://youtu.be/ryAswwQzGgI>.

C. Tarefas de formação

Uma formação composta por um robô LIMO diferencial e um quadrimotor Parrot Bebop 2 é considerada. Para controlar o movimento coordenado da formação, sendo a posição do robô LIMO o ponto de interesse para controle (PoI) da formação, adotou-se o paradigma de estrutura virtual [8], [9]. A estrutura virtual, cujo movimento deve ser controlado, é a linha reta virtual que une os dois robôs, conforme ilustrado na Figura 8. Para evitar obstáculos durante a navegação, foi adotada uma estratégia de desvio de obstáculos baseada em campos potenciais artificiais, conforme descrito em [10], [11].

Um vídeo demonstrando este experimento pode ser encontrado em <https://youtu.be/Z0fej38eTAQ>.

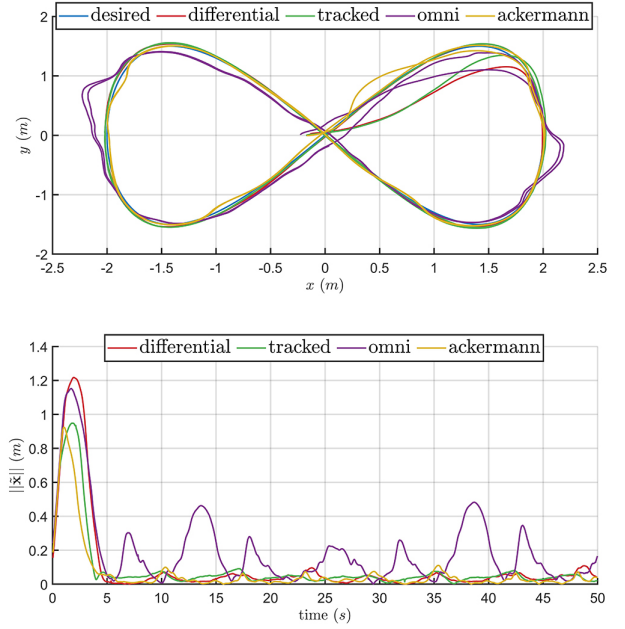


Fig. 7: Desempenho de rastreamento considerando as quatro configurações do LIMO.

D. Mapeamento e Localização

O LIMO oferece os recursos computacionais e sensores necessários para realizar tarefas de mapeamento e localização, tais como SLAM. Aliado a pacotes já existentes do ecossistema ROS, é possível executar estes algoritmos de forma autônoma, com todo processamento feito a bordo do veículo. Um exemplo de aplicação do LIMO em tarefas de mapeamento pode ser acessado em <https://youtu.be/KLNGWxkwNZ8>. Uma captura do mapa gerado no experimento pode ser vista na Figura 9.

E. Navegação em ambientes congestionados

Neste exemplo se demonstra como a plataforma LIMO pode navegar em ambientes congestionados. Equipada com sensores LiDAR e RGB-D, a plataforma é capaz de criar mapas seguros para navegação e reagir a obstáculos estáticos ou móveis.

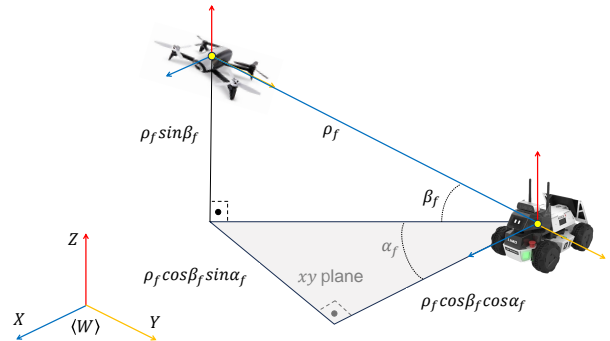


Fig. 8: Estrutura virtual, em que α_f , β_f , ρ_f são variáveis de formação.

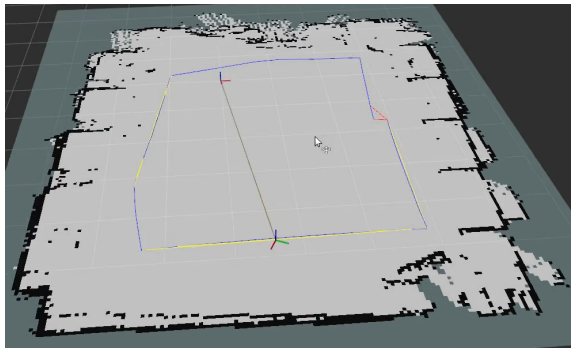


Fig. 9: Mapa gerado no experimento de mapeamento



Fig. 10: Captura do experimento demonstrando navegação em cenários congestionados.

No entanto, uma limitação do LiDAR montado no LIMO é a presença de pontos cegos na parte traseira do veículo, o que restringe a navegação segura apenas ao movimento para frente. Isso se deve ao fato do LiDAR estar montado dentro da estrutura do LIMO. Para superar essa limitação, um sensor LiDAR adicional foi instalado no topo da plataforma, eliminando os pontos cegos.

Para garantir uma navegação segura um algoritmo baseado em Mapa de Ocupação utiliza os dados do LiDAR para criar um mapa do ambiente. Em sequência, algoritmos de árvore de busca aleatória rápida (RRT) foram executados para planejar o movimento do robô entre as posições inicial e final. O mapa de ocupação, RRT e algoritmos de controle são todos executados a bordo do veículo, demonstrando a capacidade do LIMO de realizar missões de forma totalmente autônoma. Uma captura dos resultados deste teste está mostrado na Figura 10, e um vídeo do experimento completo pode ser encontrado em <https://youtu.be/fpvKulwoZTM>.

Um detalhe importante é que as três primeiras aplicações acima foram implementadas por estudantes de graduação em Engenharia Elétrica, as duas primeiras por um estudante do segundo ano, e a terceira por um estudante do último ano, ambos após três aulas sobre modelos cinemáticos para robôs móveis e operação da arena de testes. Já a quarta aplicação

foi desenvolvida por um estudante de Mestrado em Engenharia Elétrica, como parte de sua pesquisa.

V. NOTAS CONCLUSIVAS

Este artigo apresenta o robô a rodas AgileX LIMO como uma plataforma versátil para pesquisa em robótica. Se discutem aspectos de hardware, software e interfaces de comunicação com o ROS, fornecendo uma visão abrangente do sistema. Além disso, é delineado um framework de controle para cada um dos quatro modos operacionais, demonstrando sua aplicação em várias tarefas, como rastreamento de trajetória, controle de formação e navegação em cenários congestionados.

Os experimentos descritos neste manuscrito utilizaram os pacotes ROS contidos no repositório `limo_ros` fornecido pela AgileX. Portanto, com o framework aqui fornecido um pesquisador ou estudante pode facilmente replicar e começar a usar o AgileX LIMO para aplicações robóticas. Para mais informações, são fornecidas métricas de desempenho e demonstrações em vídeo, mostrando o potencial da plataforma para alavancar a educação e pesquisa em robótica.

AGRADECIMENTOS

Os autores agradecem à FAPES - Fundação de Amparo à Pesquisa e Inovação do Espírito Santo, e ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, por financiarem este trabalho. O Dr. Araujo-Neto também expressa sua gratidão à FAPES por sua bolsa de pós-doutorado, que lhe permitiu dedicar-se integralmente à pesquisa.

REFERÊNCIAS

- [1] AgileX Robotics. (2021) LIMO user manual. Accessed: 2024-08-05. [Online]. Available: [https://github.com/agilexrobotics/limo-doc/blob/master/Limo%20user%20manual\(EN\).md](https://github.com/agilexrobotics/limo-doc/blob/master/Limo%20user%20manual(EN).md)
- [2] —. (2021) LIMO python driver. Accessed: 2024-08-10. [Online]. Available: <https://pypi.org/project/pylimo/>
- [3] —. (2021) LIMO ros 1 c++ packages. Accessed: 2024-08-10. [Online]. Available: https://github.com/agilexrobotics/limo_ros
- [4] —. (2020) AgileX github repository. Accessed: 2024-08-10. [Online]. Available: <https://github.com/agilexrobotics>
- [5] —. (2021) LIMO ros 2 c++ packages. Accessed: 2024-08-10. [Online]. Available: https://github.com/agilexrobotics/limo_ros
- [6] M. Sarcinelli-Filho and R. Carelli, "Kinematic models," in *Control of Ground and Aerial Robots*. Cham, Switzerland: Springer International Publishing, 2023, ch. 2, pp. 5–22.
- [7] —, "Motion control," in *Control of Ground and Aerial Robots*. Cham, Switzerland: Springer International Publishing, 2023, ch. 4, pp. 47–72.
- [8] —, "Control of multi-robot systems," in *Control of Ground and Aerial Robots*. Cham, Switzerland: Springer International Publishing, 2023, ch. 5, pp. 99–144.
- [9] M. Sarcinelli-Filho, "Controle de formações baseado em estruturas virtuais," in *Controle de Sistemas Multirrobôs*. São Paulo, Brasil: Blucher, 2023, ch. 4, pp. 69–94.
- [10] F. A. Spagnol, N. M. dos Santos Rosa, L. F. Nunes, W. A. Neto, R. de Angelis Cordeiro, M. Sarcinelli-Filho, and D. K. D. Villa, "Comparação de técnicas de cooperação de robôs utilizando espaço nulo e estrutura virtual," in *to appear in Proceedings of the XXV Congresso Brasileiro de Automática*, Rio de Janeiro, Brazil, 2024.
- [11] M. C. P. Santos, C. D. Rosales, M. Sarcinelli-Filho, and R. Carelli, "A novel null-space-based uav trajectory tracking controller with collision avoidance," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2543–2553, 2017.