

## Ambiente de realidade mista para experimentos com múltiplos robôs de baixo custo

Héber Renato Fadel de Morais\* João Paulo Lima Silva de Almeida\*  
Lucia Valeria Ramos de Arruda\*\*

\* *Universidade Tecnológica Federal do Paraná / Instituto Federal do Paraná, (e-mail: {heber.morais, joao.almeida}@ifpr.edu.br).*

\*\* *Universidade Tecnológica Federal do Paraná, (e-mail: lvrruda@utfpr.edu.br)*

---

**Abstract:** In this paper, a mixed reality environment for navigation of multiple mobile robots is proposed. The developed environment allows the integration between virtual and real elements. To identify the real elements, ARTags markers are used, which can be obstacles, targets, robots or objectives, while the virtual ones (inserted into a virtual layer) are elements that can interact with the robots, such as virtual sensors, goals, obstacles, among others, depending on the experiment to be carried out. The Robot Operating System framework is used as a communication tool between the real and virtual layers. As a result, robots with hardware constraints acquire the skill to reach and intercept a moving goal. Three experiments are presented, highlighting the capacity of the proposed tool to provide the navigation of multiple robots in the presence of static and dynamic obstacles and targets. The obtained results validate the proposed mixed reality environment as a flexible experimental tool for the navigation of multiple robots with limited hardware.

### Resumo:

Neste artigo é proposto um ambiente de realidade mista para navegação de múltiplos robôs móveis. O ambiente desenvolvido possibilita a integração entre elementos virtuais e reais. Para identificar os elementos reais, são utilizados marcadores do tipo ARTags, os quais podem ser robôs, obstáculos, alvos ou objetivos, enquanto os virtuais (inseridos em uma camada virtual) são elementos que podem interagir com os robôs, tais como sensores virtuais, alvos, obstáculos, dentre outros, a depender do experimento a ser conduzido. É utilizado o *framework Robot Operating System* como intermediador de comunicação entre as camadas real e virtual. Neste ambiente, robôs com restrições *hardware* são capazes de alcançar e interceptar um alvo móvel. Três experimentos são apresentados para ilustrar a capacidade do ambiente em viabilizar a navegação de múltiplos robôs na presença de obstáculos e alvos, estáticos e dinâmicos. Os resultados obtidos validam o ambiente desenvolvido como uma ferramenta experimental flexível para a navegação de múltiplos robôs com *hardware* limitado.

*Keywords:* Robotics; Multi-Robot Systems; Navigation and Path Planning; Mixed Reality.

*Palavras-chaves:* Robótica; Sistemas Multi-Robôs; Navegação e Planejamento de Rotas; Realidade Mista.

---

### 1. INTRODUÇÃO

Ambientes de realidade virtual e realidade aumentada são facilmente encontrados em aplicativos para jogos e redes sociais. No entanto, várias áreas do conhecimento podem se beneficiar do uso de tais tecnologias imersivas. Entre elas cita-se a medicina, a indústria 4.0, indústria de defesa e segurança, cidades inteligentes e outras que se beneficiam da construção de sistemas ciber-físicos (CPS: do inglês cyber-physical systems) (Egger and Masood, 2020).

Ambientes de realidade aumentada que combinam medidas e dados oriundos de elementos reais e virtuais em uma mesma aplicação são denominados de ambientes de realidade mista (MR: mixed reality) e são uma área de pesquisa ativa em CPS. Algumas das vantagens do uso de realidade mista são flexibilidade espacial, baixo risco

para segurança de humanos e instalações, depuração de código computacional simplificada, entre outros (Honig et al., 2015).

Como componente importante de um CPS, os sistemas autônomos móveis, em especial os robôs móveis, podem se beneficiar do uso de realidade mista. Erat et al. (2018) desenvolveram uma interface MR para drones, que potencializa a visão humana. O sistema é uma simulação de visão de raio-x em que o usuário se concentra em explorar o espaço inacessível, enquanto o controle de voo está a cargo do piloto automático do drone. Uma interface para a teleoperação de robôs aéreos desenvolvida por Hedayati et al. (2018) permite novas formas de feedback intuitivo e visual para a teleoperação de robôs. O uso de MR resultou em melhoria significativa em medidas de desempenho e de velocidade de teleoperação, reduzindo erros. Na etapa

de projeto e desenvolvimento de robôs, a realidade mista possibilita a reprodução de habilidades requeridas para navegação de múltiplos robôs móveis, tais como emulação de sensores e de periféricos de comunicação. Dessa forma, é possível realizar diferentes experimentos antes da aquisição ou confecção do *hardware* dos robôs como, desenvolvido em (de Almeida et al., 2020), (Millard et al., 2018) e (Honig et al., 2015).

Neste contexto, o presente artigo propõe o desenvolvimento de um ambiente de realidade mista baseado em *Robot Operating System (ROS)*<sup>1</sup> (Quigley et al., 2009) para viabilizar experimentos com múltiplos robôs e a construção de robôs móveis de baixo custo. O objetivo dos robôs é o seguimento e interceptação de um alvo móvel enquanto navega em um ambiente semi desconhecido na presença de obstáculos fixos e móveis. Os robôs possuem controladores PID para movimentação e *fuzzy* para desvio de obstáculos, embarcados em seu microcontrolador.

É importante destacar que os robôs móveis desenvolvidos não possuem sensores reais, ficando a cargo da camada de realidade mista a emulação de sensores virtuais atribuídos individualmente aos robôs.

Para validar o ambiente de realidade mista proposto bem como os robôs desenvolvidos, foram realizados vários experimentos. Assim a capacidade de movimentação e desvio de obstáculos (estáticos e dinâmicos) dos robôs após a implementação do sensoriamento virtual é avaliada. Também analisa-se como o ambiente se comporta ao se adotar objetivos dinâmicos e novos elementos virtuais.

Este artigo está organizado da seguinte forma: as principais características da arquitetura desenvolvida são apresentadas na seção 2; as estratégias de controles na seção 3; os resultados são demonstrados na seção 4; e as considerações finais e trabalhos futuros na seção 5.

## 2. ARQUITETURA DE REALIDADE MISTA PARA SISTEMAS MULTIROBÔ

O funcionamento do ambiente proposto e os principais detalhes construtivos sobre o sistema multi-robô são explicados nesta seção. As descrições apresentadas servem como base para a idealização de experimentos a serem conduzidos no ambiente.

### 2.1 Ambiente de realidade mista

Uma visão geral do ambiente físico onde será inserido uma camada de realidade mista é apresentada na Figura 1. Este ambiente é composto por: (i) uma área plana real com fundo branco (Ambiente Real) de aproximadamente 1,8 m x 1,8 m; (ii) uma câmera USB, posicionada acima do espaço experimental; (iii) um computador central onde está instalado o *framework ROS*; e (iv) objetos identificados com marcadores ARTag, (Fiala, 2005), em sua parte superior, os quais podem ser estáticos (obstáculos) e ou dinâmicos (robôs móveis de pequeno porte).

A imagem capturada pela câmera é tratada pelo computador central usando o pacote ROS. Após a captura

<sup>1</sup> <https://wiki.ros.org/>

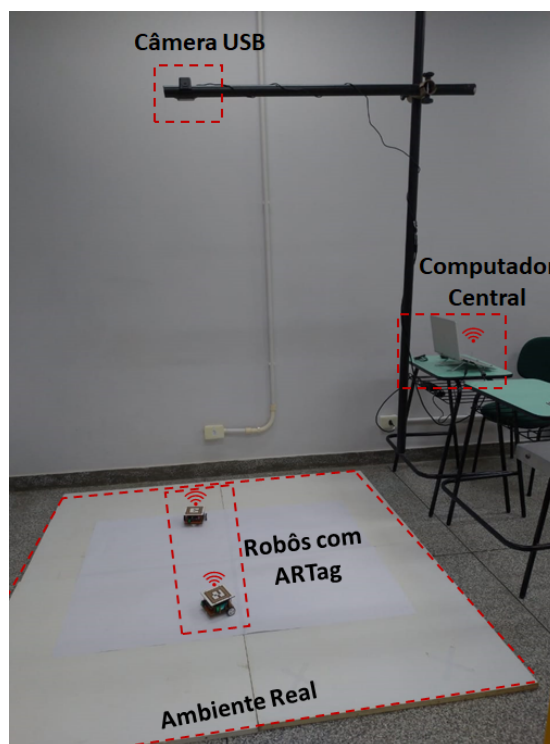


Figura 1. Visão geral do sistema de ambiente de realidade mista.

das imagens, é feito um processamento nos dados adquiridos dos marcadores ARTag identificados usando o pacote *ar\_track\_alvar*<sup>2</sup>. Para cada objeto real, identificado por sua ARTag, é associado um tópico ROS com a informação de sua posição atualizada ( $x, y$ ) no ambiente. As coordenadas assim obtidas são inseridas em uma lista com a posição de todos os objetos no ambiente. Esta lista contém tanto os objetos reais (com ARTags associadas) inseridos no piso como as coordenadas de elementos virtuais, aptos a interagir com os reais.

Portanto, a principal característica operacional da arquitetura proposta é a possibilidade de integração entre elementos reais e virtuais em um ambiente compartilhado. Além disso, a arquitetura é flexível em relação ao uso de objetos reais a serem considerados no ambiente, desde que tenham um marcador ARTag associado.

Os dados processados nesta etapa podem ser usados pelo computador central para a definição dos objetivos dos robôs, criação de obstáculos virtuais, planejamento de rotas a serem seguidas, e outras características, a depender do contexto do experimento a ser realizado.

### 2.2 Sistema Multi-Robô Desenvolvido

Com o objetivo de confirmar que a arquitetura de experimentos descrita é capaz de suportar experimentos com múltiplos robôs, neste trabalho, são utilizados dois robôs móveis, cada um deles associado a um ARTag individual, conforme mostrado na Figura 2. Esses robôs possuem uma arquitetura de *hardware* muito simples, o que os torna dependentes das habilidades virtuais a serem conferidas pela camada virtual do sistema para fins de navegação

<sup>2</sup> [http://wiki.ros.org/ar\\_track\\_alvar](http://wiki.ros.org/ar_track_alvar)

autônoma. Basicamente, os robôs não possuem sensores, e nenhum sistema de localização foi embutido em sua *hardware*.

A estrutura do chassi dos robôs foi feita em MDF cortados a laser, e alguns suportes foram produzidos com o auxílio de impressões 3D. O marcador virtual (identificador AR-Tag) é produzido no MDF superior, conforme é mostrado na Figura 2.

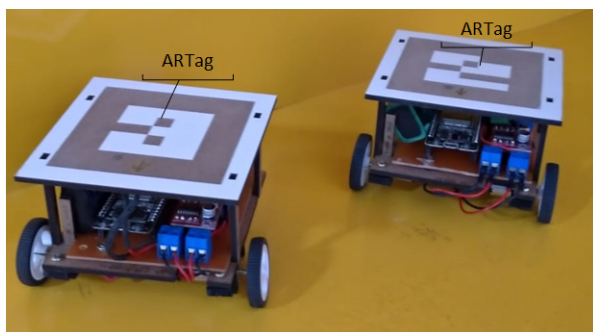


Figura 2. Robôs móveis desenvolvidos.

Os robôs possuem um microcontrolador ESP32 embarcado em sua estrutura, o qual permite conectividade sem fio (via Wi-Fi) entre o robô e o computador central, a fim de receberem de maneira individual as coordenadas de sua posição atual, as coordenadas do seu objetivo e uma lista de coordenadas de obstáculos (reais e virtuais) próximos. Tais informações são continuamente publicadas pelo computador central por meio de tópicos ROS específicos para cada uma das informações. Em uma aplicação de segurança, por exemplo, a câmera poderia ser utilizada como vigilância de um ambiente, e o computador central atuaria para comunicar aos robôs móveis sobre pontos de interesse no ambiente (áreas de exploração, alvos, obstáculos, etc).

Além de poderem acessar tópicos ROS, que contém informações publicadas pelo computador central, os robôs também podem enviar (publicar) informações em tópicos ROS individuais, com dados sobre sua navegação. Com isso, é estabelecida uma forma de comunicação entre computador central e robôs móveis, que é um dos requisitos essenciais de aplicações de sistemas multirrobô, por meio de tópicos ROS.

Para permitir a comunicação entre robôs e computador central é utilizado o pacote *rosserial\_server*<sup>3</sup>, o qual cria um soquete de rede, que provê a comunicação entre as duas pontas.

A fim de proporcionar uma flexibilidade na composição do sistema multirrobô, os algoritmos de controle de navegação (locomoção e tomadas de decisão) são integralmente embarcados no microcontrolador ESP32 e dependem apenas das informações de posições advindas do computador central. Os detalhes acerca da estratégia de navegação embarcada são apresentadas na próxima seção.

Os demais componentes que compõem a estrutura dos robôs são dados na Tabela 1. Tratam-se de componentes comuns em laboratórios e de fácil acesso, aquisição e utilização.

<sup>3</sup> [http://wiki.ros.org/rosserial\\_server](http://wiki.ros.org/rosserial_server)

Apesar dos robôs desenvolvidos terem o mesmo *hardware* e estrutura, o ambiente suporta a inclusão de outras configurações de robôs móveis, desde que atendam ao requisito de terem uma ARTag visível à câmera.

### 3. ESTRATÉGIA DE CONTROLE DE NAVEGAÇÃO

Conforme explicado anteriormente, os robôs móveis considerados neste trabalho não possuem sensores reais para a detecção de obstáculos e nem dispositivos próprios para a auto-localização. Portanto, para ser possível estabelecer uma estratégia de navegação, utilizam os recursos de realidade mista, proporcionados pelo ambiente experimental apresentado, para emular o funcionamento básico de sensores de proximidade e para estabelecer um sistema de localização.

#### 3.1 Sensoriamento Virtual

Dada as coordenadas de cada robô no ambiente, via imagem capturada pela câmera, foi traçado um semicírculo virtual a partir das coordenadas do robô (centro do marcador), conforme ilustrado na Figura 3, que representa a área de detecção de obstáculos de cada robô móvel. Tal área tem o objetivo de estabelecer um sistema de sensoriamento virtual para os robôs, como explorado em (Makhataeva and Varol, 2020; Reina et al., 2015), por meio das habilidades virtuais conferidas pelo ambiente de realidade mista. Com isso em mente, cada robô possui uma inteligência mínima para determinar se há um, ou mais, obstáculos em sua área de detecção e então executar manobras de desvio.

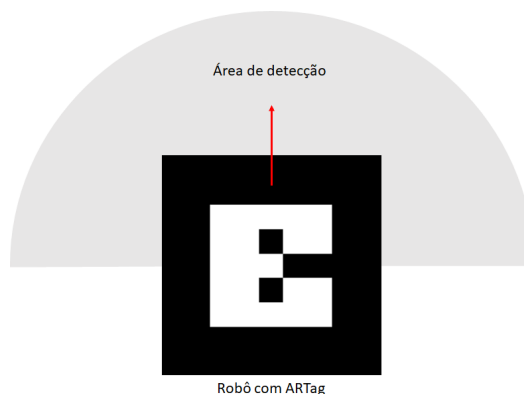


Figura 3. Funcionamento do sensor virtual dos robôs.

Do ponto de vista de execução do algoritmo embarcado em cada robô móvel, a cada iteração são acessados os tópicos ROS correspondentes às suas coordenadas em relação ao ambiente e também uma lista atualizada de coordenadas de objetos próximos, seja ela composta por qualquer outro marcador ARTag (incluindo outros robôs, como obstáculos

Tabela 1. Componentes de cada Robô.

Qtde	Componente
1	ESP32
2	Micro-motor DC 6V (N20)
2	Rodas pequenas compatíveis com o motor
1	Módulo Regulador de Tensão LM2596
1	Mini Ponte H L298N
1	Bateria 9v

dinâmicos), e/ou elementos virtuais inseridos pelo computador central. Caso alguma dessas coordenadas estejam no interior do semicírculo do robô (área de detecção), ela é considerada como um obstáculo ao robô.

Além disso, após a identificação de obstáculos em sua área de detecção, o algoritmo embarcado do robô calcula a distância entre sua posição atual e a posição dos obstáculos. O obstáculo mais próximo é selecionado pelo robô por representar o maior risco de colisão. Em seguida a distância é normalizada entre 0 (obstáculo próximo à coordenada central do robô) e 1 (obstáculo próximo à borda da área de detecção).

Com o desenvolvimento do sensor virtual, os robôs móveis considerados neste trabalho adquirem a capacidade de detectar obstáculos no interior de uma área de detecção pré-determinada e associar um índice de risco de colisão para cada detecção, para que se possa tomar a decisão de desvio conforme estratégia de controle, descrita na próxima seção.

### 3.2 Controle de Navegação

O principal objetivo da estratégia de controle de navegação autônoma considerada neste trabalho é, a partir da posição atual dos robôs móveis, guiá-los com segurança (sem colisão com obstáculos) até suas posições de destino. Esta estratégia considera o sensoriamento virtual descrito na subseção anterior.

Os robôs móveis construídos apresentam uma movimentação lenta, compatível com a capacidade da câmera em detectar os marcadores ARTag em movimento e do processamento dos pacotes ROS responsáveis por processar as imagens em obter as respectivas coordenadas detectadas. Dessa forma, não há uma variação significativa na velocidade linear dos robôs e, o controle de navegação implementado no *hardware* dos robôs consiste apenas em controlar a velocidade angular, correspondente à manobra que se pretende executar.

A estratégia de navegação considerada é compartilhada entre dois controladores: um controlador PID tradicional que aciona os motores para orientar o robô em relação à sua posição objetivo, enquanto um controlador *fuzzy* Mamdani, com defuzzificação por meio do centro de gravidade, realiza manobras que evitam colisões com obstáculos. Essa estratégia de controle pode ser vista pelo fluxograma da Figura 4.

Em relação ao controlador PID, é considerado o erro entre as orientações angulares atual do robô e a do objetivo. A ação de controle resultante (velocidade angular) visa alinhar o robô ao seu objetivo. Os ganhos do PID foram escolhidos com base em vários testes e observações.

O controlador *fuzzy* Mamdani considerado neste trabalho possui duas entradas e uma saída. A primeira entrada é a distância entre robô e obstáculo, normalizada entre 0 e 1, e suas funções de pertinência são apresentadas na Figura 5. A segunda entrada é o ângulo entre a orientação angular atual do robô e a direção do obstáculo detectado ( $\alpha$ ), cujas funções de pertinência são mostradas na Figura 6. Esta entrada varia de  $-\frac{\pi}{2}$  a  $\frac{\pi}{2}$ , pela representação adotada, um valor negativo representa um obstáculo à direita e um valor

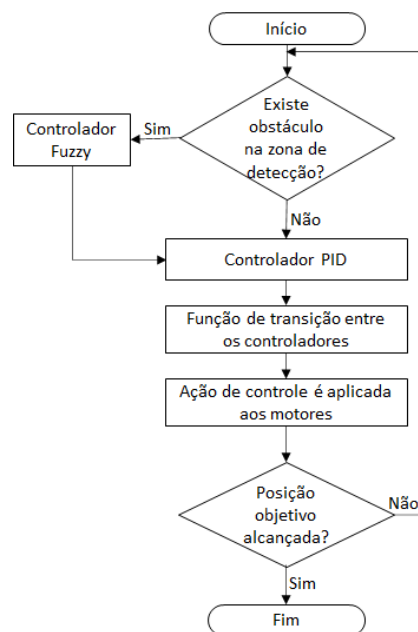


Figura 4. Fluxograma da estratégia de controle.

positivo um obstáculo à esquerda, já os valores próximos de 0 representam um situação em que o robô está posicionado bem à frente do obstáculo.

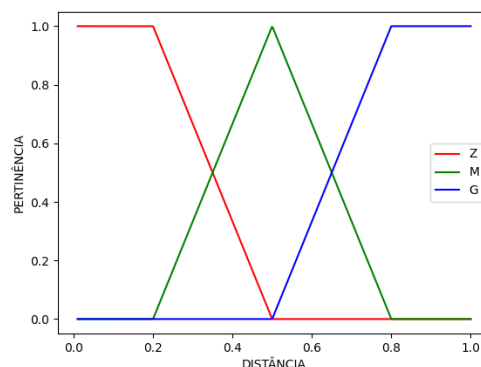


Figura 5. Variável de Entrada: Distância.

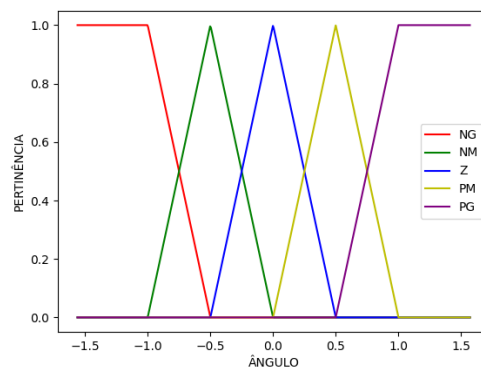


Figura 6. Variável de Entrada: Ângulo.

A saída do controlador *fuzzy* é a velocidade angular, normalizada entre -1 e 1, cujas funções de pertinência podem ser vistas na figura 7, na qual valores negativos e positivos representam manobras à direita e à esquerda, respectivamente.

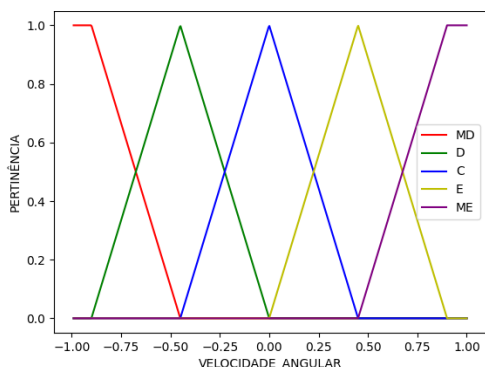


Figura 7. Variável de Saída: Velocidade Angular.

A base de regras do controlador *fuzzy* foi projetada para que o robô efetue a manobra de desvio do obstáculo detectado com segurança, por um caminho à direita ou à esquerda, de acordo com a distância e o seu ângulo em relação ao obstáculo. A superfície de resposta resultante é apresentada na Figura 8.

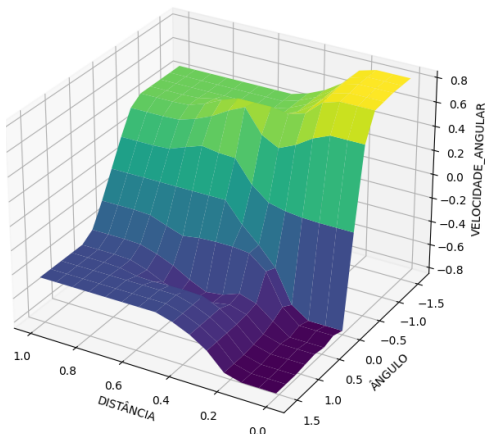


Figura 8. Superfície de resposta do controlador fuzzy.

Como pode ser visto no fluxograma da Figura 4, o controlador PID sempre busca alinhar o robô ao seu alvo, mesmo que ocorram manobras de desvios, já o controlador *fuzzy* apenas entra em atuação quando há obstáculo dentro da área de detecção do sensor virtual do robô, a fim de evitar colisões.

Com o intuito de suavizar a mudança entre PID e *fuzzy*, foi estabelecida uma função de transição entre os controladores, baseada em (de Almeida et al., 2019), como é mostrada na Eq. (1), em que:  $u$  é a ação de controle resultante (velocidade angular);  $u_{PID}$  é a ação de controle calculada pelo controlador PID;  $u_{fuzzy}$  é a ação de controle calculada pelo controlador *fuzzy*; e  $\tau$  é obtido da distância entre robô e seu obstáculo mais próximo, normalizado entre 0 e 1, em relação à área de detecção dos robôs (veja Figura 3). Dessa forma, quanto menor é o valor de  $\tau$ , mais acentuada é a atuação do controlador *fuzzy*. Conforme o obstáculo se distancia, devido ao desvio realizado, o controlador *fuzzy* deixa de ter uma atuação significativa, e a atuação do controlador PID começa a prevalecer. Quando não existem obstáculos a serem desviados ( $\tau = 1$ ) a ação de controle PID é considerada integralmente.

$$u = \tau \cdot u_{PID} + (1 - \tau) \cdot u_{fuzzy} \quad (1)$$

Evidencia-se que o algoritmo processa o desvio de apenas de um obstáculo por vez, caso múltiplos obstáculos estejam dentro da condição de desvio (distância e ângulo) apenas a informação do obstáculo mais próximo ao robô é considerada.

A partir dessas informações, o código embarcado no micro-controlador de cada robô também é responsável por converter a ação de controle resultante em sinais individuais aplicados aos motores da esquerda e direita, permitindo assim a sua movimentação.

## 4. RESULTADOS

Com o intuito de verificar se o sistema de realidade mista proposto, juntamente aos robôs móveis desenvolvidos, fornece as condições necessárias para experimentos com elementos reais e virtuais em um ambiente compartilhado, foram conduzidos três experimentos, os quais são explicados ao longo das subseções a seguir.

### 4.1 Experimento 01: Desvio de obstáculos estáticos e dinâmicos

O primeiro experimento busca mostrar a habilidade de navegação dos robôs em um ambiente onde existem obstáculos estáticos e dinâmicos, todos reais e identificados por ARTag individuais. Para isso os robôs R1 e R2 estão em suas posições iniciais e a posição final (objetivo) de R1 é a posição de R2, e vice-versa. Foram posicionados 2 obstáculos estáticos no ambiente OE1 e OE2, com o intuito de se criar um corredor por onde ambos os robôs devem passar.

O experimento foi gravado em um arquivo de vídeo (mesma imagem utilizada pelo computador central) e a sobreposição de alguns *frames* é mostrada na Figura 9, a fim de ilustrar a trajetória dos robôs.

A navegação tem início (imagem dos robôs mais opacas), onde os robôs começam a receber as coordenadas de sua posição, de seu objetivo e de obstáculos próximos. No início da navegação, ambos os robôs andam em linha reta, já que seus controladores PIDs atuam no intuito de alinhar o robô até o seu objetivo final. Por volta do centro do cenário, ambos os robôs alternam-se na detecção do obstáculo dinâmico (outro robô) e estáticos (marcadores ARTag fixos), já que o algoritmo embarcado processa o desvio de apenas de um obstáculo por vez, e a prioridade de desvio se dá pela menor distância entre robô e obstáculo. Os desvios, ora do obstáculo ora do outro robô, acontecem até que nenhum obstáculo esteja dentro da área de detecção que ativa o controlador *fuzzy*. Após os desvios ambos os robôs chegam às suas posições finais.

Uma vez que os robôs possuem a habilidade de publicarem informações via ROS, para o computador central, a respeito do registro de sua navegação, foi possível armazená-las e analisá-las a fim de verificar de forma detalhada como cada um dos robôs processou suas informações. As Figuras 10 e 11 apresentam o valor da velocidade angular aplicada aos robôs durante a execução do experimento. A linha tracejada em azul identifica os momentos em que houve

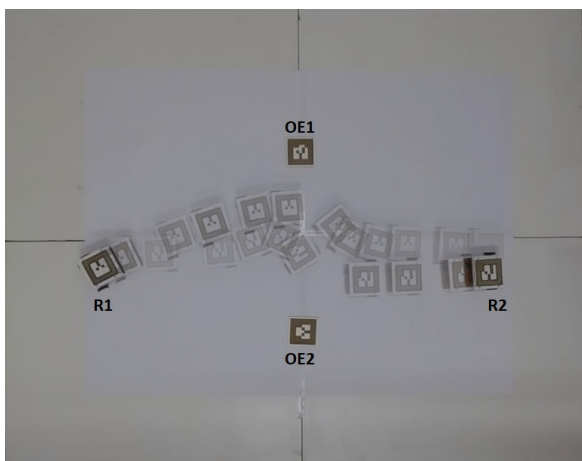


Figura 9. Trajetória dos robôs no experimento 01.

atuação do controlador *fuzzy*, independente do valor de  $\tau$  (0 - PID e 1 - *fuzzy*).

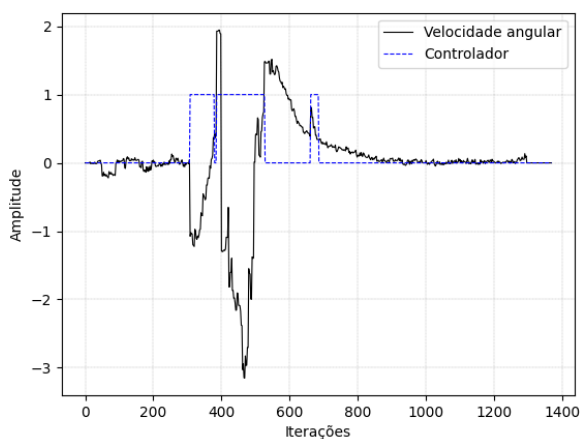


Figura 10. Dados de Navegação de R1: Ação de controle resultante.

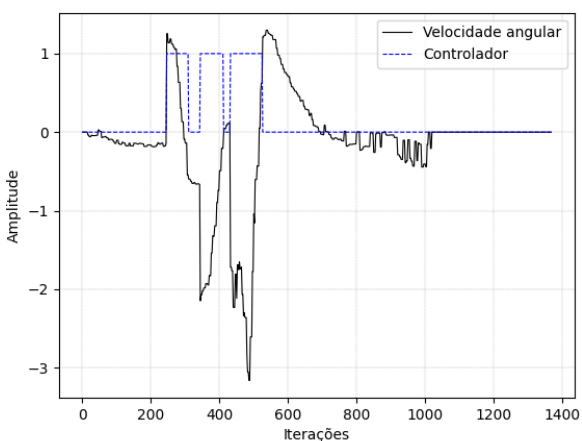


Figura 11. Dados de Navegação de R2: Ação de controle resultante.

Após a execução do experimento foi possível verificar a viabilidade da implementação do sensor virtual de desvio de obstáculos bem como se validou a ação de controle *fuzzy* considerada.

#### 4.2 Experimento 02: Alvos dinâmicos

Com o intuito de testar a capacidade do sistema proposto em se trabalhar com alvos dinâmicos, foi proposto um experimento onde um alvo virtual se move de maneira circular pelo ambiente. O algoritmo desenvolvido e gerenciado pelo computador central atribuiu este alvo à R1, já para R2 foi definido como alvo a posição atrasada de R1, a fim de evitar que os robôs se aproximem a uma medida de possível colisão, gerando uma zona de confiança de trajetória entre os robôs. Com o intuito de criar uma situação de líder-seguidor, para este experimento, o sensor virtual de desvio de obstáculos de R2 foi desativado.

A Figura 12 apresenta uma sequência de 4 *frames* da execução do experimento. No início do experimento ( $t = 0s$ ), o alvo virtual começou sua trajetória circular e anti-horária próximo da coordenada ( $x=0.87, y=0.47$ ), já R1 foi posicionado em ( $x=0.5, y=0.1$ ) e R2 em ( $x=0.3, y=0.1$ ). Ao iniciar o experimento, R1 se conduziu ao alvo dinâmico virtual, enquanto R2 seguiu a trajetória de R1 (alvo real). Os demais instantes de tempos apresentados na Figura 12 apresentam a movimentação dos robôs dentro do cenário.

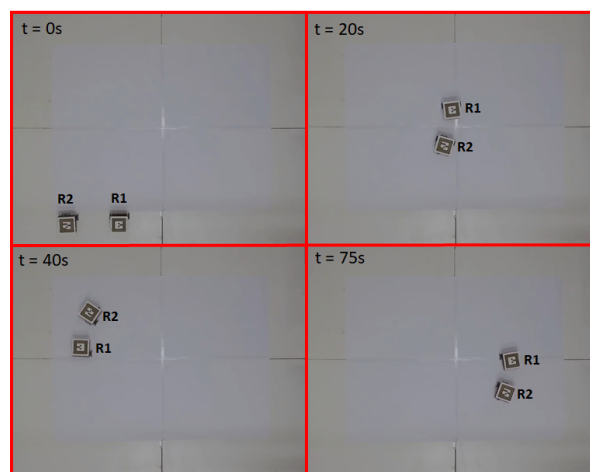


Figura 12. Sequência de *frames* do experimento 02.

A Figura 13 apresenta a evolução da movimentação do alvo virtual e dos robôs durante a execução do experimento, a linha tracejada em preto representa a trajetória do alvo virtual, já as linhas vermelha e azul representam as trajetórias de R1 e R2, respectivamente.

#### 4.3 Experimento 03: Interceptação de um alvo virtual

O último experimento proposto visou a criação de um cenário onde um intruso é identificado pelo computador central, o qual processa informações sobre sua trajetória e aciona um robô que deve interceptá-lo. Apenas um robô móvel foi considerado para a condução deste experimento.

Para descentralizar as informações a respeito do intruso, foi criado um algoritmo apenas para atualizar as coordenadas do mesmo, o qual, para este experimento, foi configurada uma trajetória linear, iniciando-se na posição ( $x=0.1, y=0.7$ ) e terminando em ( $x=0.9, y=0.7$ ), onde o algoritmo atualiza a posição do intruso em 0.0005 (pontos do ambiente) a cada 50 milissegundos.

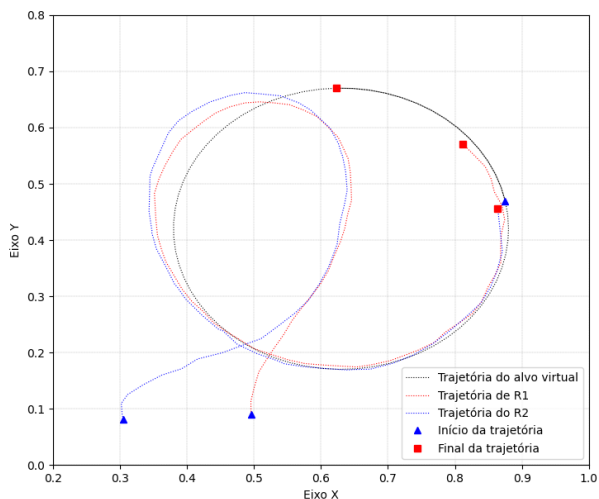


Figura 13. Trajetórias do alvo virtual, de R1 e de R2, no experimento 02.

No computador central, um outro algoritmo, que tem o objetivo de definir qual é o alvo do robô, tem conhecimento apenas da coordenada atual do intruso e do robô. A partir deste ponto, foi realizada uma estimação simples da trajetória do objeto intruso, com base em aproximação de equação da reta, abordado em (Zhu et al., 2013), e assim definir o ponto de interceptação do robô.

O experimento teve início com o robô (R1) posicionado aproximadamente em  $(x=0,98, y=0,16)$ , e o intruso virtual (i) na posição  $(x=0,1, y=0,7)$ . A Figura 14 apresenta uma sequência de 4 *frames* da execução do experimento. A marcação em azul foi inserida manualmente na imagem apenas para ilustrar uma posição aproximada do alvo virtual (ponto de interceptação estimado a cada iteração do código) em cada *frame*.

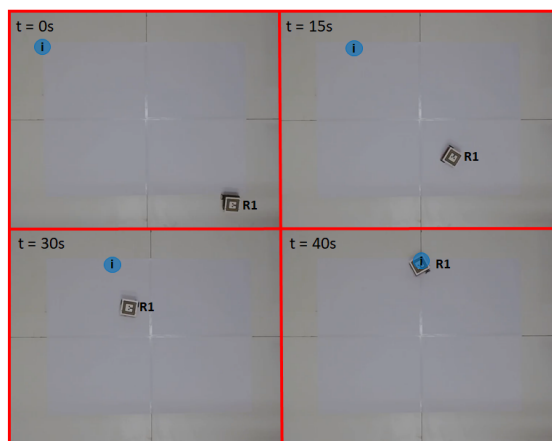


Figura 14. Sequência de frames do experimento 03.

Após a coleta dos registros de navegação foi possível plotar a trajetória percorrida pelo robô e intruso dentro da área de experimentos, as quais são apresentadas na Figura 15. Para este experimento considerou-se a velocidade do intruso inferior à velocidade do robô, conforme apontado em (Zhu et al., 2013).

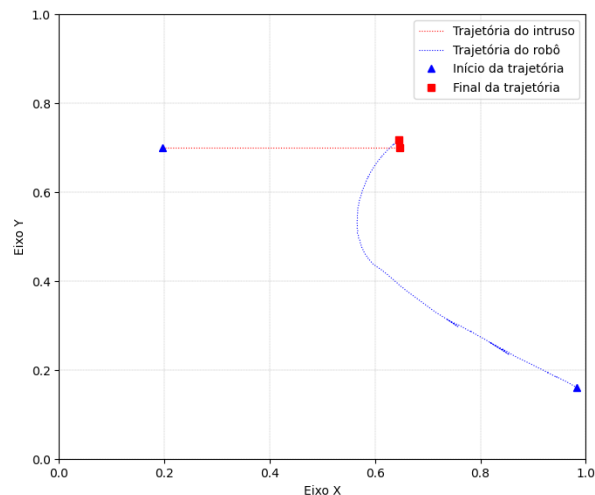


Figura 15. Trajetórias do intruso e R1.

## 5. CONCLUSÃO

Um ambiente de realidade mista envolvendo navegação autônoma de múltiplos robôs de baixo custo é proposto neste trabalho. Este ambiente é construído em um computador central onde está instalado o *framework ROS*, a partir de imagens do ambiente real obtidas com uma câmera USB.

Para validar o ambiente proposto foram realizados experimentos com robôs, projetados e fabricados com dispositivos comuns e de baixo custo. Os robôs não possuem sensores reais, os quais foram simulados com o auxílio do ambiente de realidade mista proposto.

Além disso, controladores do tipo PID e *fuzzy* foram embarcados diretamente no microcontrolador de cada robô, diminuindo a carga computacional do computador central. O controlador PID é responsável pela movimentação dos robôs e o controlador *fuzzy* implementa manobras de desvio de obstáculos. Estes controladores possibilitam a navegação autônoma e independente dos robôs. As tarefas dadas aos robôs são relativas à navegação no ambiente onde eles devem atingir uma posição alvo (fixa ou móvel) e evitar colisões com obstáculos estáticos ou dinâmicos, reais ou virtuais.

O computador central disponibiliza para os robôs, por meio de publicações ROS, as coordenadas de sua localização, objetivos e alvos próximos, para que com segurança (sem colisão com obstáculo) eles conseguissem chegar a suas respectivas posições alvos, além de permitir que objetivos dinâmicos sejam definidos para os robôs.

Os resultados obtidos mostram que o ambiente desenvolvido é capaz de fornecer um módulo de sensoriamento virtual aos robôs móveis, emulando sensores de proximidade e de localização.

Assim, é possível concluir que o sistema de ambiente de realidade mista proposto neste trabalho juntamente com os robôs desenvolvidos são uma opção viável para testar as principais funcionalidades de sistemas com múltiplos robôs móveis, onde foi considerado um misto de recursos virtuais e reais. Este ambiente permite notadamente oti-

mizar tempo e recursos de pesquisas em experimentos e desenvolvimentos na área de robótica.

Para trabalhos futuros, pretende-se aumentar a quantidade de robôs inseridos no ambiente, gerando aplicações para robótica de enxame. E também espera-se o desenvolvimento de novas estratégias de navegação e colaboração, incluindo a comunicação direta entre robôs durante a execução das tarefas.

## REFERÊNCIAS

- de Almeida, J.P.L.S., Nakashima, R.T., Neves-Jr, F., de Oliveira, A.S., and de Arruda, L.V.R. (2020). Autonomous navigation of multiple robots with sensing and communication constraints based on mixed reality. *Journal of Control, Automation and Electrical Systems*, 31(5), 1165–1176. doi:10.1007/s40313-020-00629-1. URL <https://doi.org/10.1007/s40313-020-00629-1>.
- de Almeida, J.P.L.S., Nakashima, R.T., Neves-Jr, F., and de Arruda, L.V.R. (2019). Bio-inspired on-line path planner for cooperative exploration of unknown environment by a multi-robot system. *Robotics and Autonomous Systems*, 112, 32 – 48. doi:10.1016/j.robot.2018.11.005.
- Egger, J. and Masood, T. (2020). Augmented reality in support of intelligent manufacturing – a systematic literature review. *Computers and Industrial Engineering*, 140, 1–22. doi:10.1016/j.cie.2019.106195.
- Erat, O., Isop, W.A., Kalkofen, D., and Schmalstieg, D. (2018). Drone-augmented human vision: Exocentric control for drones exploring hidden areas. *IEEE Transactions on Visualization and Computer Graphics*, 24(4), 1437–1446. doi:10.1109/TVCG.2018.2794058.
- Fiala, M. (2005). Artag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, 590–596. doi:10.1109/CVPR.2005.74.
- Hedayati, H., Walker, M.W., and Szafir, D.J. (2018). Improving collocated robot teleoperation with augmented reality. *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 78–86. doi:10.1145/3171221.3171251.
- Honig, W., Milanes, C., Scaria, L., Phan, T., Bolas, M., and Ayanian, N. (2015). Mixed reality for robotics. *IEEE International Conference on Intelligent Robots and Systems*, 2015-December, 5382–5387. doi:10.1109/IROS.2015.7354138.
- Makhataeva, Z. and Varol, H.A. (2020). Augmented reality for robotics: A review. *Robotics*, 9(2). doi:10.3390/robotics9020021. URL <https://www.mdpi.com/2218-6581/9/2/21>.
- Millard, A.G., Redpath, R., Jewers, A., Arndt, C., Joyce, R., Hilder, J.A., McDaid, L.J., and Halliday, D.M. (2018). ARDebug: An augmented reality tool for analysing and debugging swarm robotic systems. *Frontiers Robotics AI*, 5(JUN), 1–6. doi:10.3389/frobt.2018.00087.
- Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A.Y. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.
- Reina, A., Salvaro, M., Francesca, G., Garattoni, L., Pinciroli, C., Dorigo, M., and Birattari, M. (2015). Augmented reality for robots: Virtual sensing technology applied to a swarm of e-pucks. In *2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 1–6. doi:10.1109/AHS.2015.7231154.
- Zhu, Q., Hu, J., and Henschen, L. (2013). A new moving target interception algorithm for mobile robots based on sub-goal forecasting and an improved scout ant algorithm. *Applied Soft Computing*, 13(1), 539–549. doi:10.1016/j.asoc.2012.08.013. URL <https://doi.org/10.1016/j.asoc.2012.08.013>.