

JOÃO VÍTOR RODRIGUES DE VASCONCELOS

(RE)PLANEJAMENTO DE ROTAS EM TEMPO REAL PARA MISSÕES ESTRATÉGICAS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de Magister Scientiae.

Orientador: Alexandre Santos Brandão

**VIÇOSA - MINAS GERAIS
2021**

**Ficha catalográfica elaborada pela Biblioteca Central da Universidade
Federal de Viçosa - Campus Viçosa**

T

V331r
2021
Vasconcelos, João Vítor Rodrigues de, 1994-
(Re)planejamento de rota em tempo real para missões
estratégicas / João Vítor Rodrigues de Vasconcelos. – Viçosa,
MG, 2021.
65 f. : il. (algumas color.) ; 29 cm.

Orientador: Alexandre Santos Brandão.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f. 60-65.

1. Robôs móveis. 2. Tolerância a falha (Engenharia).
3. Automação. 4. Análise de caminho (Estatística). 5. Interação
homem-máquina. 6. Drone. I. Universidade Federal de Viçosa.
Departamento de Engenharia Elétrica. Programa de
Pós-Graduação em Ciência da Computação. II. Título.

CDD 22. ed. 629.8932

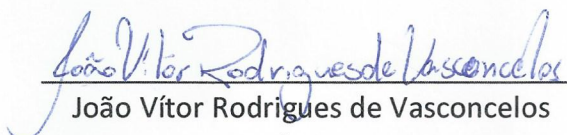
JOÃO VÍTOR RODRIGUES DE VASCONCELOS

(RE)PLANEJAMENTO DE ROTAS EM TEMPO REAL PARA MISSÕES ESTRATÉGICAS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 04 de março de 2021.

Assentimento:


João Vítor Rodrigues de Vasconcelos
Autor


Alexandre Santos Brandão
Orientador

*Este trabalho é dedicado à minha família,
tanto aquela formada através de laços sanguíneos e amor,
quanto a que foi sendo construída pelos mais estreitos laços de amizade.*

Agradecimentos

Após a confecção desta dissertação, não posso deixar de agradecer a todos que me ajudaram, tanto na realização quanto no emocional. Primeiramente gostaria de agradecer a Deus por ter abençoado todo meu percurso do mestrado.

Agradeço também aos meus pais, Júlio e Luciana, minha irmã, Juliana, pela paciência, acolhimento, apoio, presença, carinho e amor incondicional na minha vida, além de palavras que me deram forças para seguir em frente. Também, a toda minha família, principalmente, minha dindinha Luciene pelo incentivo, amor e carinho em momentos de difíceis dessa jornada.

Agradeço de coração o meu orientador Alexandre Brandão por aceitar conduzir o meu trabalho de pesquisa e ter desempenhado tal função com dedicação e amizade. Pela oportunidade de realizar pesquisas e trabalhos na área da Robótica.

Agradecimento ao Núcleo de Especialização em Robótica (NERO) pelo acolhimento, amizades feitas e disposição dos materiais para realização dos estudos, pesquisas e experimentos. Ao time Alexandre Caldeira, Daniel “Presidente”, Iure, Thiago Ridogério, Valentim, Vinícius Pacheco e Kevin, pelo apoio, momentos da zoeira, café estequiométrico e ar condicionado abaixo de zero.

Agradeço a todos meus amigos pelas horas passadas juntas dentro no laboratório, mercado, RU, Discord e jogatina. Momentos que foram difíceis e complicados, experimentos dando errado, robôs quebrando e batendo na parede, horas olhando para a tela de um computador para achar um erro de sinal. Pela presença nos momentos de glória e vitória, quando tudo dava certo no final, e recebia aprovação pelo trabalho.

Agradeço a minha namorada Isabella, pela companhia que me confortou em momentos difíceis, pelo amor que me deu forças na reta final, pelo carinho que sempre teve por mim e pela ajuda ao organizar minha vida e me direcionar para as coisas importantes, me ajudando a ter foco e dedicação em momentos de difícil concentração.

Agradeço ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, órgão do Ministério da Ciência, Tecnologia, Inovação e Comunicações, FAPES - Fundação de Amparo à Pesquisa e Inovação do Espírito Santo, e FAPEMIG - Fundação de Amparo à Pesquisa do Estado de Minas Gerais, pelo apoio dado a esta pesquisa. Agradeço também à CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pela bolsa que foi concedida, que me permitiu dedicar todo o tempo aos estudos de Mestrado.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

*“Me desculpe, Majestade. Eu sou um homem vulgar!
Mas eu garanto a você, minha música não é.”*

Wolfgang Amadeus Mozart

Resumo

VASCONCELOS, J. V. R. de, M.Sc., Universidade Federal de Viçosa, março de 2021. **(Re)planejamento de Rotas em Tempo Real para Missões Estratégicas**. Orientador: Alexandre Santos Brandão. Coorientador: Mário Sarcinelli Filho.

Esta dissertação visa o planejamento de rotas em tempo real para guiar formações heterogêneas do tipo estrutura virtual triangular, compostas por um veículo terrestre não tripulado (VTNT) e dois veículos aéreos não tripulados (VANTs). As formações navegam em ambientes semiestruturados e possuem estratégias tolerantes a falhas, capazes de lidar com situações de nível baixo de bateria, dificuldade de comunicação entre os robôs, problemas mecânicos com os robôs terrestres e detecção de rota impossível. Inicialmente, um planejador de caminhos baseado nos algoritmos *Lifelong Planning A** (LPA*) e Dijkstra determina a rota a ser seguida por um VTNT, em um ambiente sujeito a alterações devido à inclusão de obstáculos. Na sequência, a estrutura triangular é definida e as estratégias de tolerância a falhas são apresentadas para guiar uma formação heterogênea de um VTNT e dois VANTs (um real e um virtual). Em seguida, fazendo junção das abordagens, formações heterogêneas de robôs navegam e simulam possíveis falhas tanto no VANT como no VTNT. Por fim, as mesmas formações heterogêneas navegam em ambientes cujos mapas podem sofrer alterações e as rotas dos VTNTs devem ser replanejadas, quando necessário. Resultados de simulação e de experimentos reais validam as estratégias propostas e corroboram a análise teórica.

Palavras-chave: Robótica Móvel. Tolerante a Falhas. Formação. Planejamento de Caminho. Cooperação de robôs. VANT. VTNT.

Abstract

VASCONCELOS, J. V. R. de, M.Sc., Universidade Federal de Viçosa, March 2021. **(Re)Planning of Paths in Real Time for Strategic Missions**. Advisor: Alexandre Santos Brandão. Co-advisor: Mário Sarcinelli Filho.

This thesis aims to plan routes in real time to guide heterogeneous triangular formations, dealt with as virtual-structures type, composed by an unmanned ground vehicle (UGV) and two unmanned aerial vehicles (UAVs). The formations navigate in semi-structured environments and have fault-tolerant strategies, capable of dealing with situations such as low battery level, difficulty of communication between robots, mechanical problems with terrestrial robots and detection of an impossible route. Initially, a path planner based on the Lifelong Planning A* (LPA*) and Dijkstra algorithms determines the route to be followed by a UGV, in an environment subject to changes due to the inclusion of obstacles. Then, the triangular structure is defined and the strategies of fault tolerance are presented to guide a heterogeneous formation of a UGV and two UAVs (one real and one virtual). Then, associating both approaches, heterogeneous robot formations navigate and simulate possible failures in both UAV and UGV. Finally, the same heterogeneous formations navigate in environments whose maps may change, and the routes of the UGVs must be replanned, when necessary. Simulation and experimental results validate the proposed strategies and corroborate the theoretical analysis.

Keywords: Mobile Robots. Fault-Tolerant. Formation. Path Planning. Robot Cooperation. UAV. UGV.

Lista de ilustrações

Figura 1 – Robô móvel terrestre utilizado neste trabalho.	15
Figura 2 – VTNT diferencial e suas características.	16
Figura 3 – VANTs com quatro pás rotativas.	16
Figura 4 – VANT de quatro pás rotativas e suas características.	16
Figura 5 – Estratégias de controle de movimento.	17
Figura 6 – O fluxograma da estratégia de planejamento de rota para um robô móvel.	23
Figura 7 – Exemplo de mapa usado nos experimentos. Quando o custo do estado é infinito, indica que o nó ou estado não foi aberto.	24
Figura 8 – Comparação com <i>Dynamic Planning Navigation Algorithm</i> otimizado com algoritmo genético (DPNA-GA), controlador de Oliveira e Fernandes (2016).	26
Figura 9 – Comparação com o algoritmo <i>firefly</i> de Patle et al. (2017) e Códigos Binários de Matriz baseado em Algoritmo Genético (MGA), controlador de Patle et al. (2018).	27
Figura 10 – Comparação com Otimização de Colônia de Formigas com a Influência de Obstáculo Crítico (ACOIC) (HAN; PARK; SEO, 2016), e Algoritmo Genético Aprimorado (IGA) (ZHANG et al., 2016).	27
Figura 11 – Simulação para verificar o custo computacional do código.	29
Figura 12 – O cenário a ser descrito por uma grade de navegação.	30
Figura 13 – Experimento 1: Buscando o Objetivo. Instantâneos da rota percorrida, do estado inicial ao final, passando por alguns estágios intermediários.	33
Figura 14 – Experimento 2: Rota Impossível. Instantâneos da rota percorrida, do estado inicial ao final, passando por alguns estágios intermediários.	34
Figura 15 – Formação Triangular Heterogênea do tipo estrutura virtual.	36
Figura 16 – O esquema de controle multicamadas tolerante a falhas.	39
Figura 17 – Primeira estratégia de troca de posição de VANTs: triângulo isósceles.	41
Figura 18 – Segunda estratégia de troca de posição dos VANTs: mesma altitude.	41
Figura 19 – Estratégia tolerante a falhas para falhas no VTNT.	42
Figura 20 – Etapas do experimento de validação do trabalho tolerante a falhas proposto.	44
Figura 21 – Rastro dos robôs para cada estágio da missão de inspeção.	47
Figura 22 – Erros para cada parâmetro da formação #1 que funcionou durante a primeira etapa.	48
Figura 23 – Erros de cada parâmetro da formação #2 que funcionou ao longo das etapas iniciais.	49

Figura 24 – Erros para cada parâmetro da formação #3 funcional desde o estágio cinco até o final do experimento.	49
Figura 25 – Erros de posição: VANT #1 (esquerda) e #2 (direita). A região hachurada representa o VANT #2 pousado no VTNT #1.	50
Figura 26 – Ambiente virtual de simulação no software MATLAB®.	53
Figura 27 – Instantâneos da vista em perspectiva da simulação de validação do algoritmo obtido no Capítulo 2.	55
Figura 28 – Instantâneos da vista superior da simulação de validação do algoritmo obtido no Capítulo 2.	57
Figura 29 – Instantâneos da vista em perspectiva da simulação de validação da estratégia tolerante a falhas.	58
Figura 30 – Instantâneos da vista superior da simulação de validação da estratégia tolerante a falhas.	59

Lista de tabelas

Tabela 1 – Configuração de cada etapa durante a missão de inspeção.	46
---	----

Sumário

1	INTRODUÇÃO	13
1.1	Veículo Terrestre Não Tripulado (VTNT)	15
1.2	Veículos Aéreos Não Tripulados (VANTs)	15
1.3	Controladores de Navegação	17
1.4	Objetivos	17
1.5	Contribuições	18
1.6	Estrutura da Dissertação	18
2	PLANEJAMENTO DE ROTAS EM TEMPO REAL PARA MISSÕES ESTRATÉGICAS	20
2.1	Trabalhos Relacionados	21
2.2	A Estratégia do Planejamento de Caminho	22
2.3	Descrição do Ambiente de Navegação	24
2.4	Resultados e Discussão	26
2.4.1	Uma Breve Análise Comparativa	26
2.4.2	Análise de Escalabilidade	28
2.4.3	Experimento 1: Buscando o Objetivo	29
2.4.4	Experimento 2: Rota Impossível	31
2.5	Considerações Finais	31
3	FORMAÇÃO HETEROGÊNEA TOLERANTE A FALHAS	35
3.1	Caracterização da Formação	36
3.1.1	Transformações Direta e Inversa da Formação	36
3.1.2	Controle da Formação e Análise de Estabilidade	39
3.2	Estratégia Tolerante a Falhas	40
3.3	Descrição do Experimento	42
3.4	Resultados Experimentais e Discussão	45
3.4.1	Estratégia Tolerante a Falhas para Descarregamento de Bateria	45
3.4.1.1	Primeiro Tipo: Triângulo Isóceles	45
3.4.1.2	Segundo Tipo: Mesma Altitude	45
3.4.2	Missão de Inspeção	46
3.5	Considerações Finais	47
4	(RE)PLANEJAMENTO DE ROTAS COM ESTRATÉGIA TOLERANTE A FALHAS	51
4.1	A Estratégia	52

4.2	Simulação e Discussão	53
4.2.1	Simulação 1: Validação do LPA*	54
4.2.2	Simulação 2: Validação da Estratégia Tolerante a Falhas	54
4.3	Considerações Finais	56
	REFERÊNCIAS	60

1 Introdução

Nos últimos anos, a área de robótica vem chamando a atenção e recebendo destaque dos pesquisadores, por conta do seu potencial em inovação, pesquisa e prestação de serviços (RUBIO; VALERO; LLOPIS-ALBERT, 2019). Manipuladores robóticos recebem destaque pela sua alta velocidade, precisão e repetibilidade ao executar uma tarefa. Sua habilidade de realizar trabalhos está além da capacidade humana. Contudo, eles carecem de mobilidade, restringindo-se a um espaço de trabalho limitado (SIEGWART; NOURBAKHS; SCARAMUZZA, 2011). Por outro lado, os robôs móveis têm a mobilidade como sua principal característica. Através deles problemas que envolvem navegação começaram a ser solucionados (NAKHAEINIA et al., 2011). Esses robôs podem ser classificados em aquáticos, aéreos e terrestres. Nesta dissertação, o foco está nos veículos aéreos não tripulados (VANTs) e veículos terrestres não tripulados (VTNTs).

Robôs móveis autônomos podem auxiliar na solução de problemas existentes nas áreas de segurança, agricultura e inspeção. Dentre algumas missões, tais robôs podem atuar na cobertura de reservas naturais ou áreas que contêm riscos naturais, como vulcões e erosão; transporte de carga; mapeamento de ambientes de difícil acesso ou hostis; e manutenção de infraestruturas. Para exemplificar, Shimosasa et al. (1999) usaram um VTNT para patrulhar um ambiente, sendo capaz de detectar pessoas, fogo e vazamentos de gás e líquido. Han et al. (2020) fizeram uma cooperação entre um VTNT e um VANT para melhorar o sensoriamento remoto agrícola. Lee et al. (2018) desenvolveram um VTNT equipado com câmera, capaz de realizar missões de mapeamento e inspeção de ambientes. Villa, Brandao e Sarcinelli-Filho (2019) utilizaram dois VANTs para transportar e manipular uma carga, com foco no controle de sua posição. Caldeira et al. (2020) criaram uma estratégia de seguimento de caminho utilizando VANT para atravessar passagens estreitas, possibilitando o mapeamento de ambientes de difícil acesso. Zhang, Tian e Lu (2017) equiparam um VTNT com um manipulador robótico, tornando-o capaz de realizar manutenções em equipamentos de uma subestação. Além dos campos supracitados, os robôs móveis estão presentes também nas áreas militar, de inteligência, de telecomunicações, de mineração, de filmagem e de competições e lazer.

Realizar a navegação dos robôs móveis é um verdadeiro desafio, principalmente quando o ambiente contém obstáculos, estáticos ou dinâmicos. Para isso existem técnicas capazes de criar rotas para guiar o agente no ambiente de navegação. São algumas delas: Lógica *Fuzzy*, campos de potenciais artificiais, redes neurais artificiais, algoritmo genético e otimização via colônia de formigas. Singh e Thongam (2018) propuseram uma estratégia envolvendo Lógica *Fuzzy* que é capaz de guiar um VTNT em um ambiente que contém obstáculos estáticos. Nikranjbar, Haidari e Atai (2018) utilizaram a técnica de campos

potenciais para comandar um VTNT até o objetivo da missão, em um ambiente contendo obstáculos dinâmicos. Shamsfakhr e Sadeghibigham (2017) criaram uma rede neural que norteou um VTNT em um ambiente desconhecido com obstáculos estáticos e dinâmicos. Likaj et al. (2017) fizeram a união de duas técnicas: a Lógica *Fuzzy* para ser o instrutor do VTNT em um ambiente possuindo obstáculos estáticos e o algoritmo genético para encontrar o melhor intervalo de tempo para o sistema fazer consultas em sua lógica. Ajeil et al. (2020) guiaram um VTNT em um ambiente com obstáculos dinâmicos, utilizando a otimização via colônia de formigas. Ademais, existem os algoritmos de busca que são conhecidos na literatura, como o algoritmo de Dijkstra (DIJKSTRA, 1959) e o A* (HOLTE et al., 1994).

Em uma tarefa, onde há mais de um agente envolvido, tem-se comumente uma cooperação de robôs, cujo objetivo é otimizar, agilizar ou viabilizar a solução de uma operação, bem como ampliar as limitações individuais de cada ente em formação. Como exemplo, Pizetta, Brandão e Sarcinelli-Filho (2019) possibilitaram o aumento do peso máximo da carga realizando o seu transporte utilizando dois VANTs. Rabelo, Brandao e Sarcinelli-Filho (2020) empregaram a cooperação heterogênea envolvendo um VANT e um VTNT, onde o primeiro carregaria a mercadoria até o segundo, que aproximaria do local de destino de entrega, permitindo que o primeiro finalize o pedido. Entre vários outros exemplos de cooperação coordenada de robôs, é importante mencionar que as formações podem se dar segundo uma estratégia líder-seguidor, estrutura virtual, baseadas em comportamento e em consenso. Em (VILLA; BRANDAO; SARCINELLI-FILHO, 2020), em uma formação líder-seguidor, o VANT líder tem um alvo na sua traseira e o objetivo do VANT seguidor é manter uma posição definida em relação ao alvo, que foi detectado por meio de sua câmera frontal. Já em (ERNANDES-NETO; SARCINELLI-FILHO; BRANDÃO, 2019), é apresentada uma abordagem comportamental baseada em subsunção por espaço nulo, onde um conjunto de regras definem as prioridades de posição dos robôs e de forma da formação durante a navegação. Em (VILLA; MELO-JUNIOR; BRANDAO, 2017), uma estratégia baseada em consenso é implementada para um grupo de VTNTs, que devem alcançar ao mesmo tempo um ponto de encontro coletivo, manter a formação desejada e navegar conforme referência. Por fim, navegações baseadas em estruturas virtuais, as quais devem manter formas geométricas podem ser vistas em (BARBOSA et al., 2018; BACHETI; BRANDÃO; SARCINELLI-FILHO, 2020), para um seguimento de linha; em (BRANDÃO; SARCINELLI-FILHO, 2016; ERNANDES-NETO; BRANDÃO; SARCINELLI-FILHO, 2020), para uma formação triangular; e em (PACHECO et al., 2018; ERNANDES-NETO; PACHECO; BRANDÃO, 2020) para uma formação em quadrado.

Para complementar os conceitos apresentados até então, a sequência deste capítulo traz uma breve descrição acerca dos veículos terrestres e aéreos não tripulados, usados durante o trabalho, e as estratégias de navegação e controle utilizadas no cumprimento das missões estratégicas propostas.

1.1 Veículo Terrestre Não Tripulado (VTNT)

Os VTNTs utilizados na dissertação são robôs móveis de tração diferencial, denominado *Pioneer3DX* da *Adept MobileRobots*, mostrado na Figura 1. Suas rodas estão alinhadas sobre um mesmo eixo virtual, mas são acionadas individualmente. Seu modelo cinemático é dado por $\dot{\mathbf{x}} = \mathbf{A}_c \boldsymbol{\mu}$ e, na forma expandida onde, (x_c, y_c) é a posição do ponto de controle c , u é a velocidade linear do robô, sempre para sua frente, ω é a sua velocidade angular, ψ é a sua orientação, definida como o ângulo formado pelo vetor u em relação ao eixo- x e a é a distância entre o centro do eixo virtual que une as rodas e o ponto c , conforme mostrado na Figura 2.

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix}, \quad (1.1)$$

1.2 Veículos Aéreos Não Tripulados (VANTs)

Os dois modelos de VANTs com quatro pás rotativas usados neste trabalho estão ilustrados na Figura 3, que são o *AR.Drone 2.0* e o *Bebop 2*, ambos desenvolvidos pela

Figura 1 – Robô móvel terrestre utilizado neste trabalho.

(a) *Pioneer3DX* da *MobileRobots*



(b) Plataforma de reconhecimento utilizada no sistema de rastreamento do laboratório NERO.



Figura 2 – VTNT diferencial e suas características.

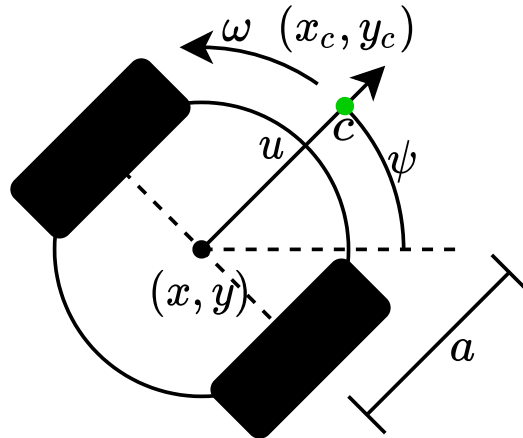


Figura 3 – VANTs com quatro pás rotativas.

(a) AR.Drone 2.0.



(b) Bebop 2.



Parrot Inc. Seu modelo cinemático é dado por $\dot{\mathbf{x}}^W = \mathbf{F}\mathbf{u}_{ref}^b$ e, na forma expandida, por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \\ u_\psi \end{bmatrix} \quad (1.2)$$

onde (x, y, z) é sua posição no mundo, (W do inglês, *world*), ψ é sua orientação em relação ao eixo- x^W e \mathbf{u}_{ref}^b é o vetor que contém as velocidades lineares, u_x, u_y, u_z e a

Figura 4 – VANT de quatro pás rotativas e suas características.

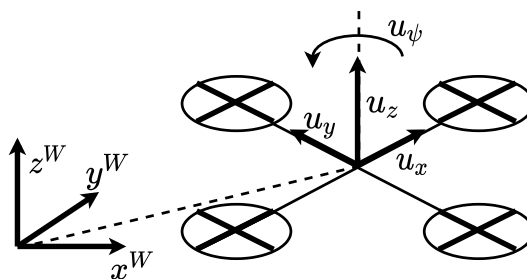
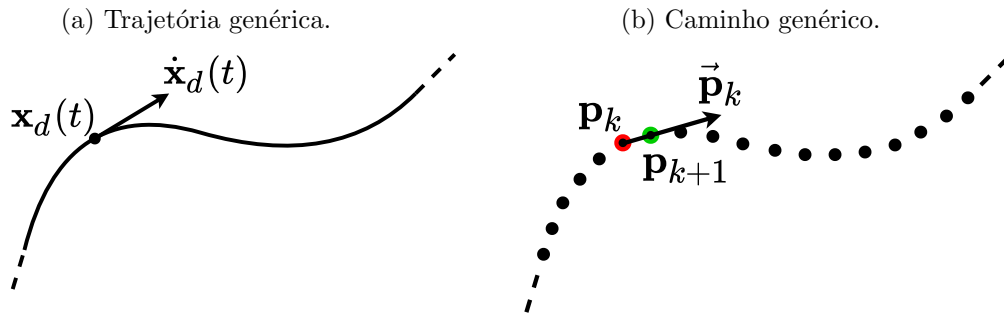


Figura 5 – Estratégias de controle de movimento.



velocidade angular u_ψ . A Figura 4 apresenta essas variáveis.

O uso do modelo cinemático é feito nas simulações para verificação de possíveis erros nas propostas feitas. Tal verificação é confiável devido a riqueza de dados de posicionamento e velocidade que é adquirido no experimento prático pelo *OpitTrack*.

1.3 Controladores de Navegação

Para realizar alguma movimentação com o robô, seja ele qualquer, é preciso utilizar uma estratégia de controle. Em suma, considerando um robô inicialmente estático, para que ele saia de sua posição atual $\mathbf{x} = [x \ y]^T$ e alcance uma posição desejada $\mathbf{x}_d = [x_d \ y_d]^T$, deverá haver uma redução do erro de posição, dado por $\tilde{\mathbf{x}} = \mathbf{x}_d - \mathbf{x}$. Para fazer com que $\mathbf{x} \rightarrow \mathbf{x}_d$, faz-se necessária a adoção de velocidades, a fim de levar $\tilde{\mathbf{x}} \rightarrow \mathbf{0}$, de forma convergente. A esta adoção de velocidades dá-se o nome de sinal de controle.

Caso o ponto desejado seja variante no tempo, conforme ilustrado na Figura 5a, tem-se uma trajetória. Por sua vez, caso uma rota seja composta por um conjunto de pontos e o robô não possua restrição temporal para alcançar cada ponto, tem-se um caminho, como ilustrado pela Figura 5b. Nela, \mathbf{p}_k é o ponto mais próximo do robô e $\vec{\mathbf{p}}_k$ é a velocidade de referência para que ele atinja o próximo ponto \mathbf{p}_{k+1} .

1.4 Objetivos

Esta dissertação visa o planejamento de rotas em tempo real para guiar formações heterogêneas do tipo estrutura virtual triangular, compostas por um veículo terrestre não tripulado (VTNT) e dois veículos aéreos não tripulados (VANTs). Além disso, essas formações são equipadas com estratégias tolerante a falhas. Para atingir esse propósito, são propostos os seguintes objetivos específicos:

1. Utilizar um algoritmo para guiar um VTNT em um ambiente semiestruturado, onde ocorre a inserção de obstáculos.

2. Desenvolver um método confiável para auxiliar a realização de missões de inspeção com base na formação heterogênea.
3. Propor uma estratégia tolerante a falhas envolvendo VANT e VTNT, tais como descarga da bateria dos VANTs, falha de comunicação entre VANT e VTNTs e falhas mecânicas no VTNT.
4. Desenvolver uma estratégia tolerante a falhas capaz de guiar formações heterogêneas em um ambiente semiestruturado, onde ocorre o acréscimo de obstáculos. Mantendo as falhas consideradas no item anterior, e com a adição de uma nova falha ao VTNT, quando o mesmo fica impossibilitado de alcançar o destino.

1.5 Contribuições

Esta dissertação apresenta contribuições traduzidas em produção científica publicadas ou em fase de publicação. O texto apresentado no Capítulo 2 é resultado de Vasconcelos, Brandão e Sarcinelli-Filho (2020) cujo objetivo era criar uma navegação em um ambiente semi-dinâmico utilizando a conjunção dos algoritmos LPA* e Dijkstra para encontrar um caminho viável e ótimo ligando o robô ao seu objetivo, sempre que um evento imprevisto ou inesperado ocorrer. O Capítulo 3 é resultado de Vasconcelos et al. (2020) cujo objetivo era solucionar dois tipos de falhas envolvendo VANT e VTNT. Foi tomado como falhas envolvendo os VANTs a descarga de bateria, e para os VTNTs aquelas relacionadas a falhas mecânicas ou de comunicação. Para finalizar, o Capítulo 4 é resultado do trabalho em fase de publicação cujo objetivo é guiar formações susceptíveis a falhas em ambiente semi-dinâmico, utilizando como navegação as contribuições do Capítulo 2, e como tratamento das falhas as estratégias criadas no Capítulo 3, adicionando a rota impossível como uma falha envolvendo o VTNT.

1.6 Estrutura da Dissertação

A dissertação foi estruturada para conter capítulos autocontidos, onde cada um terá sua introdução e conclusão de acordo com o seu objetivo.

- 1º Capítulo apresenta uma introdução geral e aborda brevemente o VTNT e os VANTs utilizados, bem como seus modelos cinemáticos.
- 2º Capítulo apresenta uma implementação do algoritmo de busca *Lifelong Planning A** (LPA*) para guiar um robô móvel em um ambiente que sofre alterações ao longo do tempo, detectando obstáculos e atualizando as informações de mapeamento simultaneamente. Primeiramente, simulações validam a estratégia, e posteriormente

experimentos comprovam esses resultados considerando uma aplicação real. A partir dos resultados, é possível comprovar que o algoritmo LPA* é capaz de guiar o robô móvel até seu objetivo em um ambiente semiestruturado. Obstáculos são incluídos simultaneamente no ambiente de navegação, para forçar o replanejamento da rota e a busca por uma nova solução ótima conectando a posição atual do robô e a posição desejada.

- 3° Capítulo focaliza os sistemas de inspeção envolvendo longas distâncias, que são comumente suscetíveis a falhas, as quais podem ser oriundas do ambiente, da comunicação entre os agentes, da estrutura eletromecânica do robô, entre outros. Visando a eficiência, estabilidade e segurança da inspeção envolvendo VTNTs e VANTs, foram propostas estratégias tolerantes a falhas usando uma estrutura virtual de forma triangular. Dois tipos específicos de falhas são tratados: descarga de bateria do VANT e falha de comunicação ou mecânica do VTNT. Uma estratégia tolerante a falhas é proposta para cada tipo e, para validação, foram feitos experimentos de cooperação com os robôs *Pioneer3DX*, *AR.Drone 2.0* e *Bebop 2*.
- 4° Capítulo apresenta a junção do algoritmo utilizado no Capítulo 2 com as estratégias tolerantes a falhas propostas pelo Capítulo 3, para realização de uma missão envolvendo inspeção em um ambiente semi-dinâmico. Além da falha mecânica ou de comunicação envolvendo o VTNT, uma nova falha será considerada, quando a formação estiver com o seu caminho totalmente obstruído. Simulações foram feitas para validar a proposta.

2 Planejamento de Rotas em Tempo Real para Missões Estratégicas

Nas últimas décadas, a robótica móvel tem sido usada com sucesso em muitas aplicações, como segurança, indústria e missões militares, por exemplo. Para realizar essas missões, um robô deve ser capaz de navegar e explorar o ambiente de forma autônoma e, frequentemente, o planejamento de caminho é um dos problemas a serem resolvidos. Este último envolve a busca de uma rota ótima ou sub-ótima da posição inicial até a desejada em seu espaço de trabalho (ZHANG; LIN; CHEN, 2018).

Em um ambiente dinâmico, como aqueles característicos de cenários de guerra ou zonas de desastre, as mudanças de rota ocorrem com frequência, devido às passagens ficarem obstruídas ou a perda de informações sobre alguma região. Nesse caso, o robô irá tomar a decisão de atualizar a rota para o seu destino, cada vez que ocorrer uma mudança no ambiente. Em outras palavras, ao se interromper momentaneamente ou definitivamente o caminho anterior/atual computado, o robô deve seguir uma nova rota, a qual será o resultado de um planejamento local realizado ao detectar um obstáculo ou perder informações sobre uma determinada zona por onde ele deveria passar.

Do ponto de vista militar, Ackerman (2014) menciona que nos últimos anos o Exército dos EUA incorporou robôs para reduzir o número de humanos em suas ações estratégicas. Como resultado, eles investiram amplamente em pesquisa e desenvolvimento de sistemas autônomos (SAPATY, 2015). Com um propósito semelhante, agentes inteligentes têm sido usados em locais de desastres naturais para minimizar as perdas humanas (BHONDVE; SATYANARAYAN; MUKHEDKAR, 2014; TUNA; GUNGOR; GULEZ, 2014; NARAYANAN; IBE, 2012). É importante notar que em ambos os cenários o ambiente pode mudar durante a execução da missão, sendo necessária a incorporação de uma estratégia de planejamento de caminhos capaz de atualizar a rota computada sempre que houver um evento imprevisto ou inesperado.

Existem vários algoritmos disponíveis na literatura para lidar com a otimização de rota, como o algoritmo de Dijkstra (DIJKSTRA, 1959), o algoritmo A* (HOLTE et al., 1994) e o algoritmo *Lifelong Planning A** (LPA*) (KOENIG; LIKHACHEV; FURCY, 2004). Todos eles podem resolver problemas de planejamento de caminhos e encontrar uma solução ótima. A principal diferença entre eles é o nível de complexidade do algoritmo. Todos esses três algoritmos lidam com o problema de planejamento de caminhos, encontrando uma rota ótima para o robô, com diferentes tempos de execução. Quando ocorre uma mudança no ambiente, como a adição/aparecimento de um obstáculo, se essa mudança

afeta o caminho encontrado anteriormente, os algoritmos de Dijkstra e A* resolvem o problema de planejamento do caminho novamente, enquanto o LPA* tira proveito da solução já obtida para este novo problema. Logo, esta última abordagem resolve o problema de forma mais rápida e eficaz.

Nesse contexto, este capítulo discute a aplicação do algoritmo LPA* em um problema real, cujo objetivo é guiar um robô móvel até um ponto-alvo em um ambiente que muda ao longo do tempo. Essas mudanças são caracterizadas pela adição de obstáculos que impedem a navegação ou causam incapacidade de seguir uma rota previamente calculada, emulando cenários de perigos ou desastres recentes.

2.1 Trabalhos Relacionados

Diversas estratégias têm sido propostas para guiar robôs autonomamente em ambientes dinâmicos. Algumas delas, relacionadas a este capítulo, são comentadas a seguir.

Um planejamento de movimento inteligente com desvio de obstáculos dinâmicos é proposto por Nasrinahar e Chuah (2018). Dois controladores de lógica *Sugeno Fuzzy* trabalham separadamente para lidar com tarefas de prevenção de colisões. Quatro controladores de comportamento são responsáveis por guiar a navegação. São eles: alcançar o alvo; alvo alcançado; controlador de velocidade e desvio de obstáculos. Este último é dividido em dois outros, para lidar separadamente com obstáculos estáticos e dinâmicos.

Oliveira e Fernandes (2016) contribuíram com uma nova estratégia de navegação denominada *Dynamic Planning Navigation Algorithm* otimizado com algoritmo genético (DPNA-GA), que garante que o robô alcance o objetivo em um ambiente desconhecido contendo obstáculos estáticos e dinâmicos. Em resumo, o robô segue um caminho composto por metas de localização e de destino.

Patle et al. (2017) usam o algoritmo *firefly* otimizado e o aplicam à navegação de robôs móveis. Para isso, foi feita uma zona em torno do agente que representa a intensidade da luz de um vaga-lume e, a partir disso, o algoritmo é capaz de detectar e evitar qualquer objeto dentro dessa zona segura em um ambiente estático. Patle et al. (2018) melhoram as decisões do robô a partir de um algoritmo genético, que e permite a navegação em ambientes dinâmicos.

Uma nova metodologia de planejamento de caminho baseada na Otimização por Colônia de Formigas é apresentada por Han, Park e Seo (2016) para ambientes com obstáculos estáticos. Os autores consideram críticos os obstáculos pertencentes ao segmento de reta que conecta o ponto de partida com o ponto de destino, e usam essa informação para melhorar o processo de otimização. Em seguida, eles introduzem a Otimização de Colônia de Formigas com a Influência de Obstáculo Crítico, ou simplesmente ACOIC.

Oliveira, Carvalho e Brandão (2019) lidam com o problema de planejamento de caminho para plataformas com sensor limitado e capacidades de processamento. Eles assumem que uma estratégia de mapeamento está disponível, porém o robô não possui informações anteriores sobre o ambiente. Conforme o robô navega em direção ao alvo, ele armazena informações que lhe permitem gerar rotas abaixo do ideal usando o algoritmo de busca A^* . Quanto mais informações o robô tiver sobre o ambiente, mais próxima da rota ótima estará a solução obtida. Considerando um ambiente dinâmico conhecido, o trabalho de Luan e Thinh (2020) usa um algoritmo de busca para encontrar, de maneira ótima, uma forma de chegar à posição desejada. Durante a execução da tarefa, o algoritmo é capaz de desviar e recomputar os caminhos, evitando assim colisões com obstáculos dinâmicos. Além disso, o algoritmo de navegação híbrido mostra-se mais eficiente que outros conhecidos na literatura e apresenta excelente desempenho em ambientes com muitos obstáculos dinâmicos.

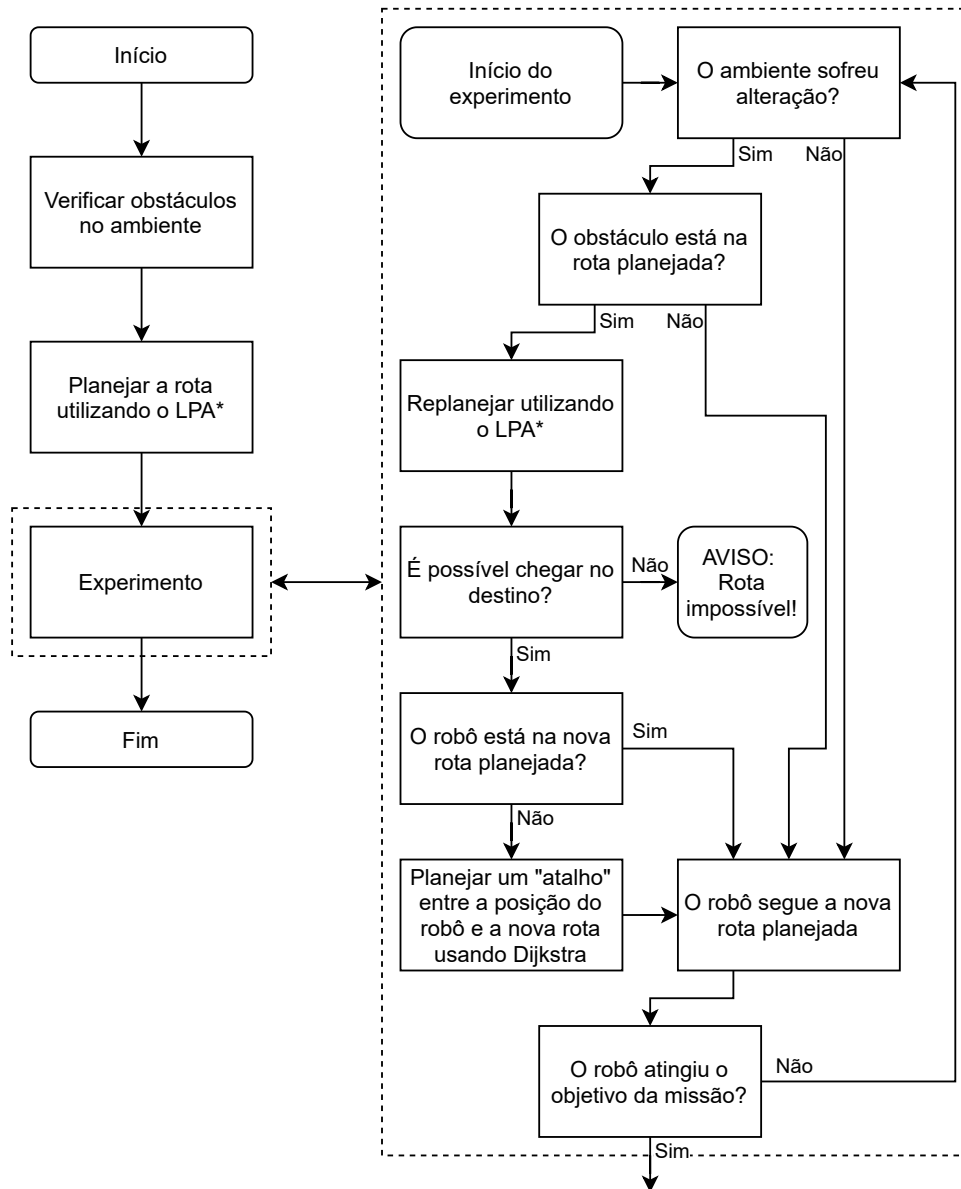
Por sua vez Zhang et al. (2016) utilizaram um algoritmo genético aprimorado capaz de encontrar um caminho ideal em um ambiente dinâmico. Este trabalho considera que todo o ambiente é conhecido pelo algoritmo e que o robô é uma partícula, ignorando suas dimensões físicas. O trabalho garante que um caminho ótimo seja encontrado, melhor que o A^* e o *Dynamic* A^* (D^*), que comumente encontram caminhos subótimos.

Todos os trabalhos listados nesta seção aplicam algum algoritmo de otimização para realizar um planejamento de caminho. Alguns deles para ambientes estáticos e outros para ambientes dinâmicos. Assim, dão suporte teórico à estratégia proposta neste capítulo.

2.2 A Estratégia do Planejamento de Caminho

A Figura 6 ilustra o fluxograma da estratégia de planejamento de rota de um robô móvel, considerando a possibilidade de mudanças no ambiente, mediante o surgimento de obstáculos desconhecidos ou imprevistos. O Algoritmo 1 auxilia na interpretação do fluxograma.

Figura 6 – O fluxograma da estratégia de planejamento de rota para um robô móvel.



Inicialmente, é verificada a presença de obstáculos no ambiente. Em seguida, o algoritmo LPA* (KOENIG; LIKHACHEV; FURCY, 2004) fornece uma rota viável. Se houver alguma mudança no ambiente que torne impossível seguir a rota encontrada anteriormente, o Algoritmo 1 recalcula e redesenha a rota. Se uma solução viável não puder ser encontrada para atingir a meta, o robô para (ou retorna à posição inicial) e habilita um aviso de “Rota impossível”. Se, após o replanejamento, o robô não estiver na nova rota calculada, um “atalho” livre de colisão é estabelecido entre a posição atual e o ponto mais próximo da nova rota, usando o algoritmo de Dijkstra (DIJKSTRA, 1959). Para finalizar, a missão é cumprida quando o robô atinge o objetivo.

A principal contribuição deste capítulo é a forma como os algoritmos LPA* e Dijkstra colaboram, explicado no parágrafo anterior, para encontrar um caminho viável

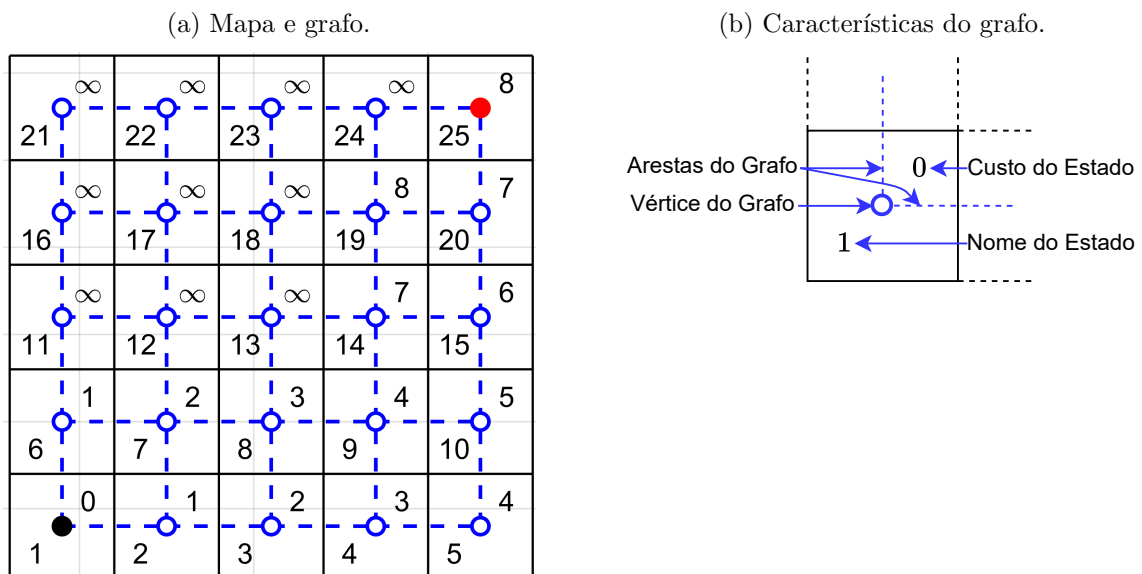
e ótimo ligando o robô ao seu objetivo, sempre que um evento imprevisto ou inesperado ocorrer.

2.3 Descrição do Ambiente de Navegação

Para realizar os experimentos, um mapa em formato de grade 5×5 medindo 3×3 m é usado, conforme mostrado na Figura 7a. Os pontos preto e vermelho são o estado inicial S_{init} e o objetivo S_g , respectivamente. A Figura 7b indica o que significa cada característica do mapa e grafo, que é representado em azul. Para implementar o algoritmo LPA*, as seguintes funções foram usadas :

- $T(n)$ é a função de transição que assume um estado n e retorna todos os estados sucessores de n .
- $g(n)$ retorna o custo entre o estado inicial S_{init} e o estado n .
- $C(n, c)$ retorna o custo da transição do estado n para o estado c .
- $h(n)$ é a função heurística que retorna uma estimativa do custo entre o estado n e o objetivo S_g .
- $f(n)$ retorna o resultado de $g(n) + h(n)$.

Figura 7 – Exemplo de mapa usado nos experimentos. Quando o custo do estado é infinito, indica que o nó ou estado não foi aberto.



Algorithm 1: *Lifelong Planning A** (LPA*)

```

Input:  $S_{init}$ ,  $S_g$ , FECHADO
Output: Rota de  $S_{init}$  até  $S_g$ 
if FECHADO está vazio then
  | Insere  $S_{init}$  em ABERTO e FECHADO
else
  | ABERTO recebe FECHADO e é ordenado por  $f(n)$ 
  while ABERTO não está vazio do
    |  $n$  recebe e retira o primeiro elemento de ABERTO
    if  $n = S_g$  then
      | Retorna a rota entre  $S_{init}$  e  $n$ 
      | break
    end
    |  $i$  recebe todos os sucessores de  $n$  a partir da função de transição  $T(n)$ 
    for  $c$  variando em cada estado de  $i$  do
      | if  $c$  não está contido em FECHADO then
        | Insere  $c$  em ABERTO e FECHADO
      | end
      | if  $c$  está em FECHADO e  $f(c) > g(n) + C(n, c) + h(c)$  then
        | Atualiza  $g(c)$  recebendo o valor de  $g(n) + C(n, c)$ 
        | Atualiza  $f(c)$  usando o novo valor de  $g(c)$ 
      | end
    end
    | Ordena ABERTO em ordem crescente por  $f(n)$ 
  end
end

```

Como dito anteriormente, o Algoritmo 1 explica computacionalmente a Figura 6.

Para o mapa proposto, adotou-se que um objeto ocupa o espaço da grade e ele é tratado como um obstáculo. No processo de definição do sucessor, o centro de cada célula é assumido como um nó. Neste capítulo, é adotado o sistema de vizinhança 4, ou seja, cada nó possui 4-vizinhos. Para calcular a função heurística, utilizou-se a métrica de distância euclidiana, descrita por,

$$h(n) = d(n, S_g) = \sqrt{(x_n - x_{S_g})^2 + (y_n - y_{S_g})^2}, \quad (2.1)$$

onde x_n e y_n representam a posição atual do robô, enquanto x_{S_g} e y_{S_g} são sua posição desejada.

2.4 Resultados e Discussão

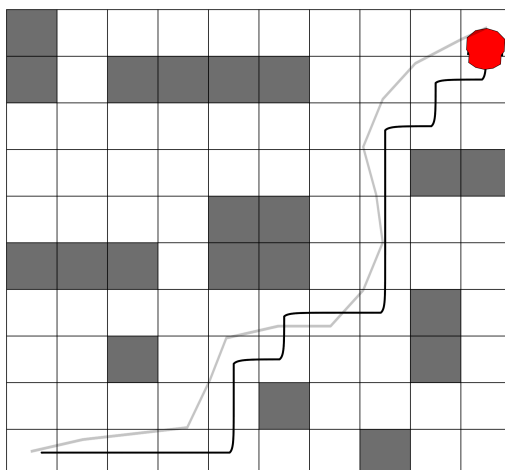
Esta seção apresenta a simulação e os resultados experimentais que validam a navegação proposta em um ambiente mutável. O objetivo é simular cenários de guerra ou desastre natural, onde a rota previamente definida deve ser alterada devido a obstruções inesperadas. Antes de validar o algoritmo proposto em cenários dinâmicos, ele foi comparado com outros que apresentam comportamento semelhante.

2.4.1 Uma Breve Análise Comparativa

Considerando que uma análise comparativa pode ser feita sobre a qualidade de uma estratégia de planejamento de caminho, esta subseção demonstra que a estratégia proposta funciona de forma semelhante (e muitas vezes melhor) do que as propostas por Oliveira e Fernandes (2016), Han, Park e Seo (2016), Patle et al. (2017), Patle et al. (2018). Assim, o objetivo das simulações a seguir é duplo: validar a estratégia proposta e compará-la com outras também em um ambiente estático.

As Figuras 8–10 mostram a comparação com cinco trabalhos, que tratam da navegação em ambientes estruturados ou dinâmicos. Antes de iniciar a comparação, vale a pena mencionar as seguintes considerações: este capítulo realiza (a) uma implementação do algoritmo LPA*, (b) o sistema Manhattan de navegação é usado para descrever a movimentação do robô, (c) todo o ambiente e qualquer mudança nele são considerados conhecidos (ou podem ser detectados e relatados ao planejador) e (d) todas as simulações são realizadas com características de ambiente semelhantes/idênticas (tamanho, discretização, posição dos objetos e assim por diante). Ao analisar as figuras, as rotas resultantes da estratégia proposta correspondem às linhas pretas, enquanto as rotas resultantes dos algoritmos em comparação são aquelas em cinza.

Figura 8 – Comparação com *Dynamic Planning Navigation Algorithm* otimizado com algoritmo genético (DPNA-GA), controlador de Oliveira e Fernandes (2016).



deliberativa, pois há um conhecimento prévio do mapa de navegação.

A Figura 9a mostra que o algoritmo *firefly* (PATLE et al., 2017) realiza uma otimização local, que conseqüentemente faz com que o robô execute mais movimentos do que o necessário para atingir o objetivo. Em contraste, o LPA* encontrou o caminho que requer menos movimentos para evitar o obstáculo, ou seja, o agente vai direto ao alvo, vencendo o obstáculo. A seguir, a Figura 9b ilustra que o algoritmo Códigos Binários de Matriz baseado em Algoritmo Genéticos (MGA) (PATLE et al., 2018), o qual é mais eficiente que o anterior, embora ainda realize uma manobra reativa. Em outras palavras, o agente só evita o obstáculo após detectá-lo.

A Figura 10a mostra a comparação com o algoritmo ACOIC (HAN; PARK; SEO, 2016). Observe-se que os caminhos percorridos são quase idênticos, porque ambos usam o sistema Manhattan de navegação e funcionam de forma deliberativa. Finalmente, um comportamento semelhante pode ser observado na Figura 10b, onde a diferença entre as rotas ocorre porque o algoritmo IGA (ZHANG et al., 2016) usa vizinhança-8 ao invés de 4, como é feito pelo LPA*.

2.4.2 Análise de Escalabilidade

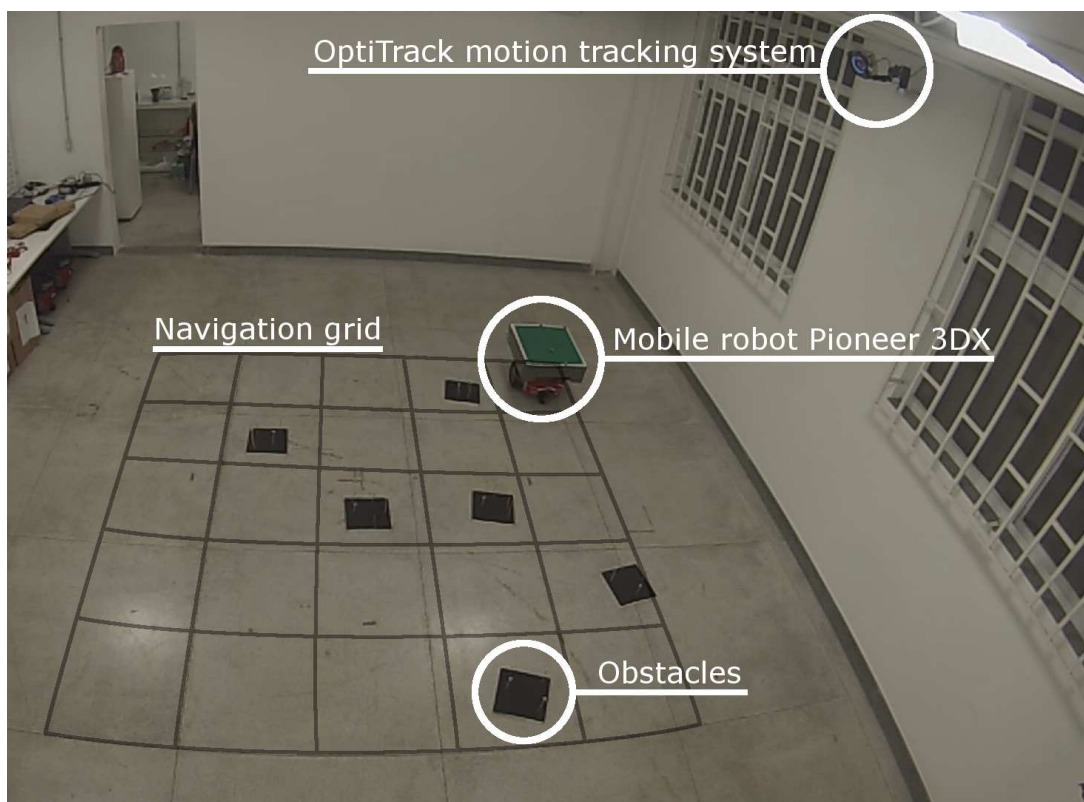
Para afirmar que o algoritmo proposto é escalável, esta seção realiza simulações usando um mapa 5×5 e um mapa 50×50 . Outros tamanhos de mapas também podem ser considerados, mas foi decidido explorar esses e, posteriormente, explicar como lidar com mapas maiores ou menores. No início, os dois mapas estão limpos. Todas as células são zonas de navegação livre. Em seguida, o robô começa a navegar, com o objetivo de cruzar e chegar ao lado oposto do mapa. Durante a navegação, o código de simulação pausa seis vezes e as zonas bloqueadas são colocadas aleatoriamente no ambiente. Após a última pausa, aproximadamente 24% dos mapas já estão rotulados como ocupados.

Sempre que o código é pausado, o robô para, o LPA* é executado e retorna uma nova rota ótima unindo a posição inicial à posição alvo. Depois, o algoritmo de Dijkstra é executado e encontra um “atalho” unindo o agente à rota calculada pelo algoritmo LPA*. Em seguida, o robô retoma sua navegação de acordo com a nova rota computada. A Figura 11 ilustra a situação da última parada e as soluções viáveis encontradas pela estratégia proposta em ambos os casos.

O computador usado para rodar as simulações tem uma placa-mãe *Gigabyte B450M*, um processador *AMD Ryzen 5 1600* 3,6 GHz, 16 GB de memória *RAM* de 3000 MHz e foi utilizado o software *MATLAB*[®]. Levando em consideração tal configuração de hardware e software, os tempos médios gastos para encontrar uma solução viável para os mapas 5×5 e 50×50 são 24,6 e 58,2 ms, respectivamente.

Como se pode presumir, quanto maior for o mapa, maior será o tempo necessário

Figura 12 – O cenário a ser descrito por uma grade de navegação.



a linha verde sólida indica a rota planejada pelo algoritmo e a linha vermelha tracejada indica a rota que o robô usou como referência.

O primeiro experimento, mostrado na Figura 13, é detalhado passo a passo, destacando os eventos que ocorrem a cada instante. Primeiramente, a Figura 13a ilustra o ambiente sem obstáculos e o caminho ótimo até a meta, usando o algoritmo LPA* para uma navegação Manhattan. Em seguida, três obstáculos são inseridos nas células 4, 10 e 13, conforme enfatizado pelo sinal X no mapa da Figura 13b. Conseqüentemente, o algoritmo procura outra solução viável. Vale destacar que a rota só muda se necessário. Observe que os obstáculos 4 e 10 requerem uma mudança de rota, ao contrário do obstáculo 13. Em outras palavras, o algoritmo LPA* replaneja a rota sempre que uma obstrução corrompe a rota computada anteriormente; o que não acontece quando o obstáculo 13 é colocado no ambiente. Mais uma vez, observe que o obstáculo 13 não obstrui o percurso, mesmo aquele criado após a colocação dos obstáculos 4 e 10. Agora, na Figura 13c, ocorre outro evento bloqueando a região 14, de forma que o robô não possa mais seguir seu caminho. Conforme já descrito, o algoritmo é executado e encontra outro caminho. Porém, o robô não está na nova rota e é necessário alcançá-la. Assim, um “atalho” é criado conectando a posição atual do robô ao seu ponto mais próximo no caminho. Destaca-se que nem sempre esse atalho é direto. Em vez disso, o algoritmo de Dijkstra garante um caminho livre de colisões conectando a posição atual do robô e o ponto mais próximo da nova rota. A Figura 13c ilustra esta situação de bloqueio, replanejamento e atalho. Finalmente, as regiões 17 e 24

recebem a etiqueta ocupada, conforme mostrado nas Figuras 13d e 13e, respectivamente. Note que o LPA* é ativado novamente e estabelece o caminho a ser seguido até que o robô alcance seu objetivo, conforme mostrado na Figura 13f.

Um vídeo do experimento está disponível em: <<https://youtu.be/j7qNcCzvB0c>>. Através do vídeo e da Figura 13, pode-se concluir que o sistema de rastreamento capturou com precisão todos os obstáculos e o algoritmo, ao perceber a alteração, replanejou rapidamente a rota. Assim, o robô atinge a meta.

2.4.4 Experimento 2: Rota Impossível

Neste experimento, foi implementado um caso para o qual obstáculos isolam o alvo e o robô, e aparece um alerta de Rota Impossível. A Figura 14 ilustra o experimento, assim como o vídeo disponível em <<https://youtu.be/tBI9kx9Itb8>>.

Analogamente ao primeiro experimento, este se inicia com um ambiente sem obstáculos, conforme ilustrado na Figura 14a. O caminho muda logo após definir as zonas 5 e 13 como bloqueadas. Atente à Figura 14b que o obstáculo 5 exige um redesenho do caminho, mas o obstáculo 13 não. Na Figura 14c, note que a obstrução ocorre em uma área onde o robô já passou, no caso, a zona 2. Como esperado, a rota não sofre nenhuma alteração e o robô segue em frente. No entanto, quando a zona 9 fica ocupada, conforme mostrado na Figura 14d, o LPA* procura uma nova rota viável. Contudo, o robô não está na nova rota e, portanto, a estratégia de atalho se encarrega de encontrar um caminho para conectá-los. Finalmente, a Figura 14e ilustra o robô seguindo o caminho, pouco antes de receber o alerta de Rota Impossível. Este evento acontece depois que as zonas 17 e 21 são rotuladas como não seguras para navegar, conforme mostrado na Figura 14f. Em outras palavras, o alerta indica que não há caminho conectando as zonas de partida e de destino. Então, o robô deve retornar à sua posição inicial ou parar. Neste capítulo, optou-se pela última abordagem, conforme destacado no vídeo do experimento.

2.5 Considerações Finais

Este capítulo usa uma combinação dos algoritmos *Lifelong Planning A** (LPA*) e Dijkstra para propor uma estratégia de planejamento de caminho para um robô móvel em um ambiente sujeito a alterações. O algoritmo LPA* replaneja o caminho quando um obstáculo aparece no caminho previamente calculado, enquanto o algoritmo Dijkstra cria um atalho entre a posição atual do robô e o novo caminho planejado para facilitar a transição do robô para o caminho atualizado.

Nas simulações, o algoritmo LPA* atingiu o objetivo proposto, guiando o robô móvel em um ambiente que muda ao longo do tempo. Da mesma forma, em experimentos reais o robô era, de fato, guiado pelo algoritmo implementado, que modificava interativamente

o caminho planejado conforme surgiam obstáculos imprevistos. Em suma, a proposta possibilita um replanejamento de caminhos e a busca de uma solução ótima que garanta o cumprimento da missão, sempre que houver uma rota viável disponível.

Finalmente, a estratégia proposta é escalável e pode ser adaptada para missões em ambientes maiores. Em suma, o mapa seria subdividido em setores contendo grades menores. Em um nível superior, o algoritmo calcularia um caminho que passa por setores até chegar ao seu destino. Em um nível baixo, dentro de cada setor, outra rota levaria o robô para o próximo setor. No entanto, essa estratégia não é abordada neste capítulo e fica como uma sugestão para sua continuidade.

Figura 13 – Experimento 1: Buscando o Objetivo. Instantâneos da rota percorrida, do estado inicial ao final, passando por alguns estágios intermediários.

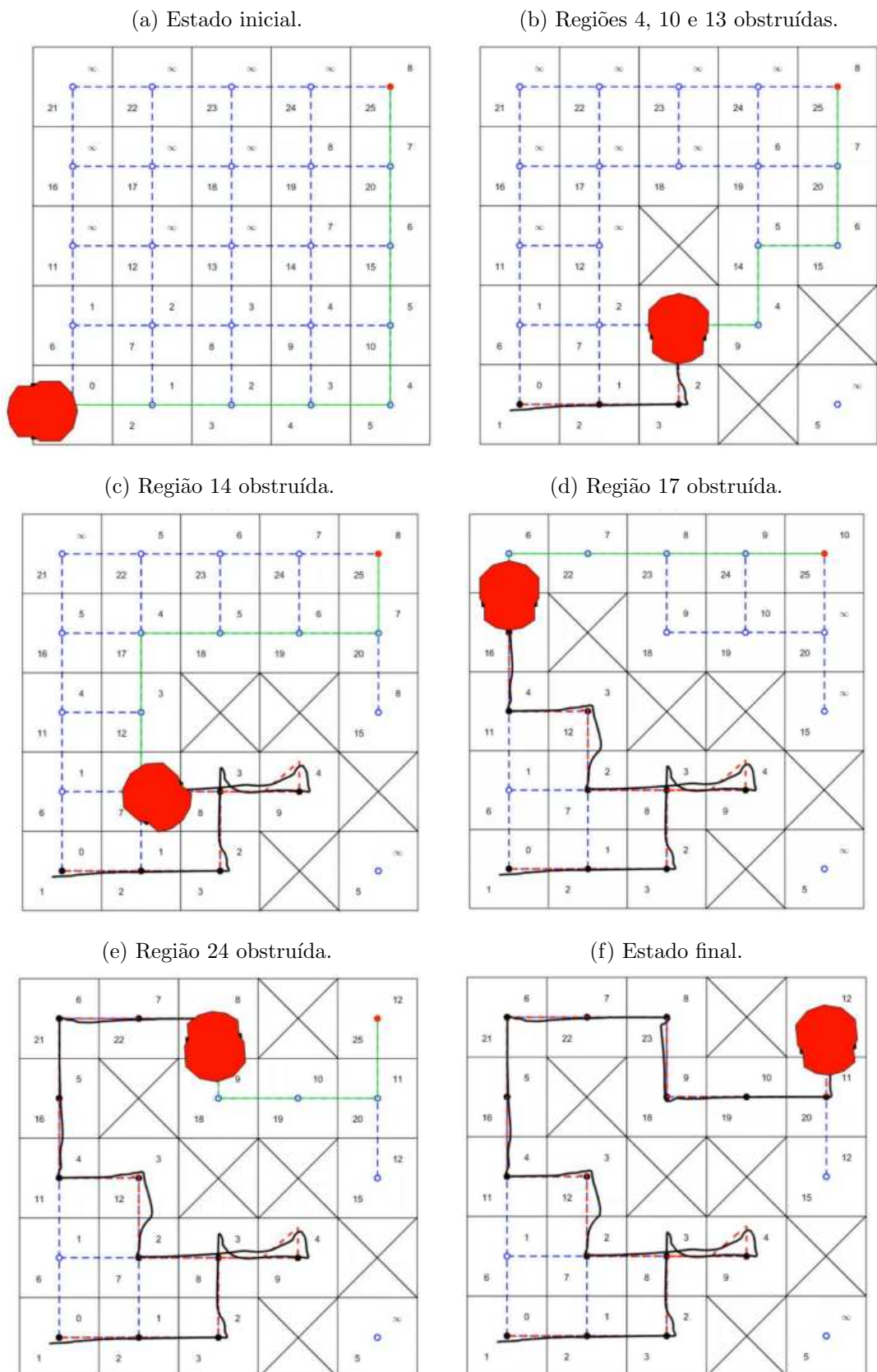
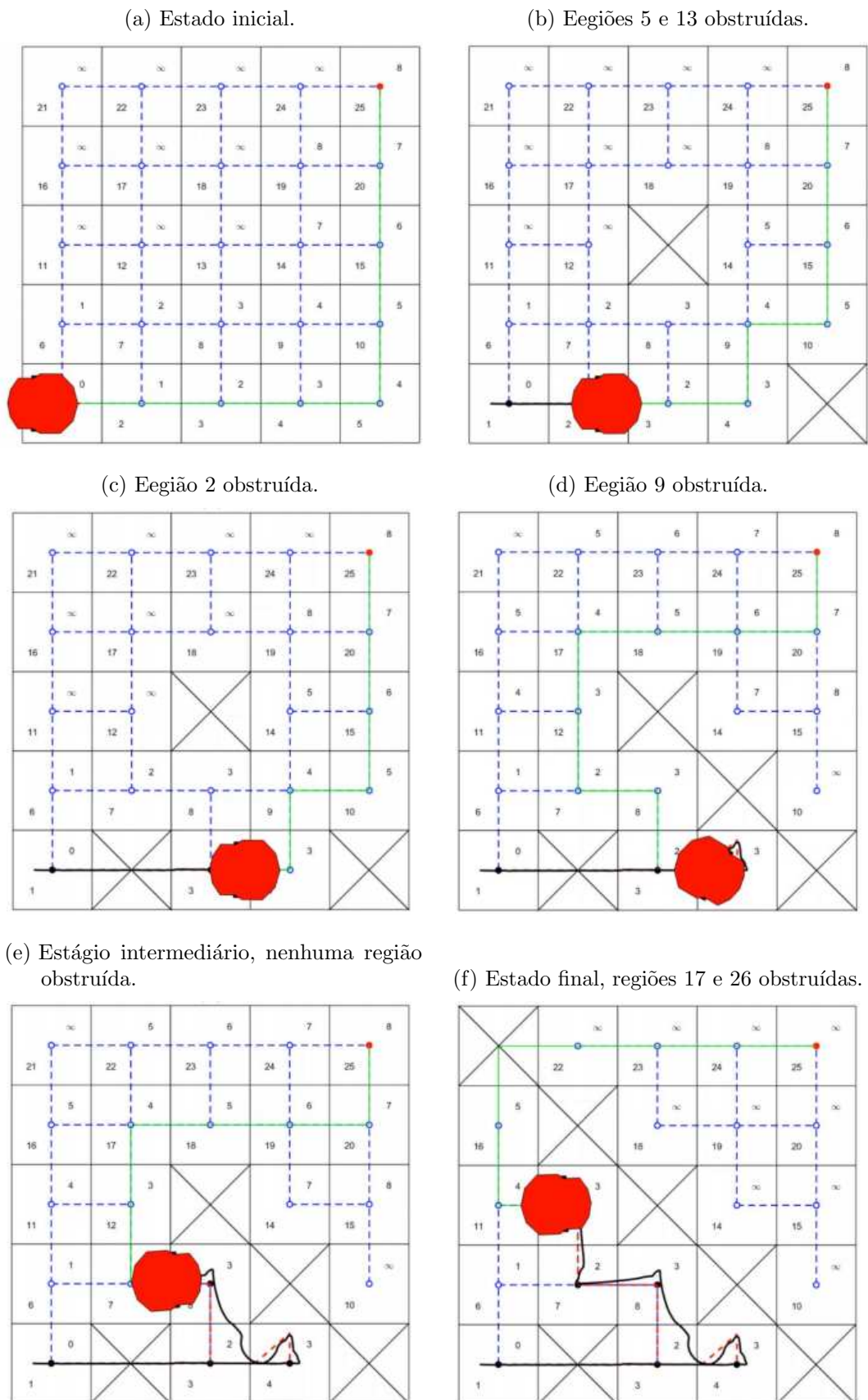


Figura 14 – Experimento 2: Rota Impossível. Instantâneos da rota percorrida, do estado inicial ao final, passando por alguns estágios intermediários.



3 Formação Heterogênea Tolerante a Falhas

A ampla gama de pesquisa e desenvolvimento no campo da cooperação de robôs aéreos e terrestres se concentra principalmente em aproveitar a mobilidade e o campo de visão dos VANTs, bem como a capacidade para transporte de carga e desempenho da bateria dos VTNTs (ÇAŞKA; GAYRETLI, 2014). Várias aplicações investem em visão computacional aplicada em VANTs para busca de alvos (TANNER, 2007), detecção e rastreamento (SHEN et al., 2017), vigilância ambiental e interna (PHAN; LIU, 2008; SASKA; KRAJNIK; PFEUCIL, 2012), localização e mapeamento para várias aplicações *indoor* e *outdoor*, em pequena e grande escala (CANTELLI et al., 2013; POTENA et al., 2019; KIM et al., 2019; MAINI; SUJIT, 2015). Além disso, a entrega autônoma de pacote (GUÉRIN et al., 2015; ARBANAS et al., 2016), bem como outros serviços industriais e comerciais, é beneficiada pela cooperação envolvendo VTNT-VANT.

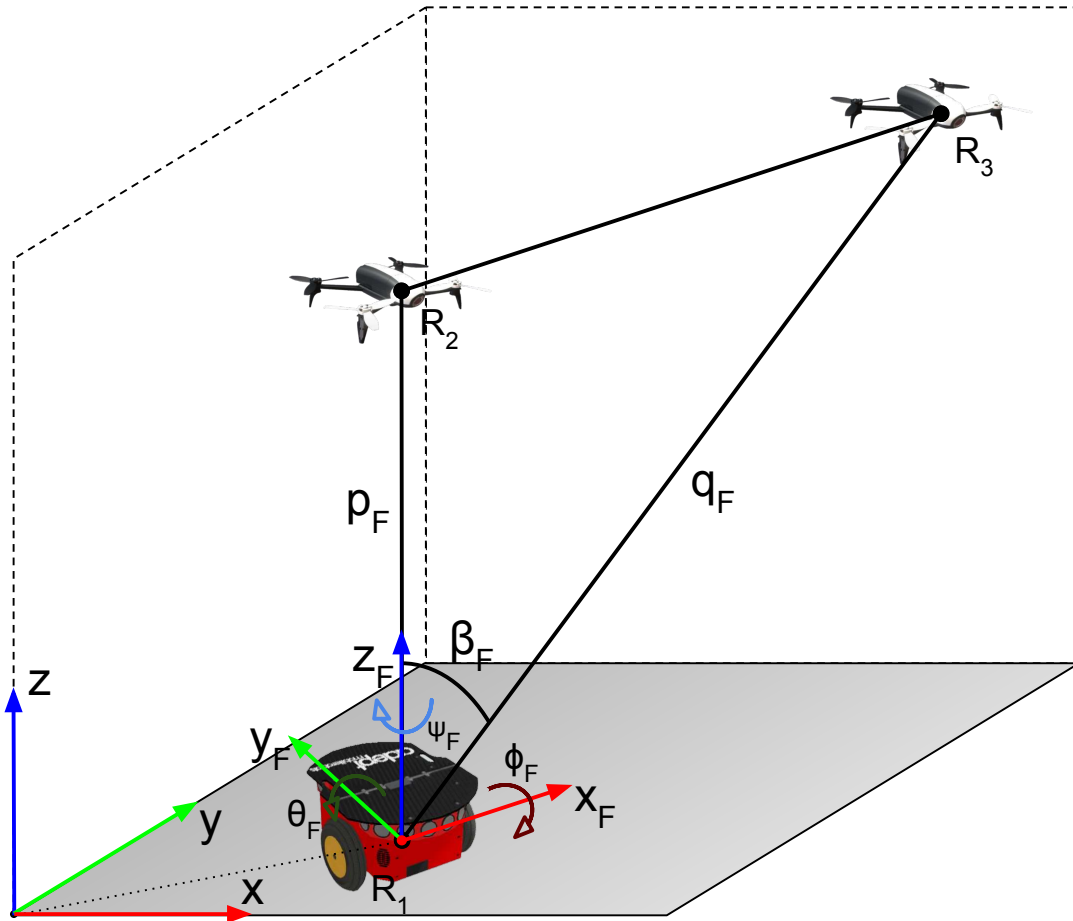
No entanto, apesar da aplicabilidade e popularidade dessas colaborações, existem relativamente poucas análises de estratégias tolerantes a falhas envolvendo formação de robôs, especialmente na colaboração de robôs terrestres e aéreos. É sensato pensar que há um número considerável de possibilidades e fontes de falhas e erros. Sendo assim, este capítulo traz uma abordagem para o problema de identificação e tratamento de exceções e falhas na cooperação VTNT-VANT, especificamente usando um esquema de controle de múltiplas camadas para formações de estrutura virtual (triangular).

Para desenvolver um método confiável para controlar e planejar missões de inspeção com base em uma formação heterogênea, é proposta uma estratégia tolerante a falhas envolvendo VANT e VTNT. Nesse caso, tomou-se como falhas envolvendo os VANTs a descarga da bateria, e para os VTNTs aquelas relacionadas à mecânica ou comunicação. A contribuição deste capítulo é apresentar uma solução para esses dois tipos de falhas. No primeiro caso, é pressuposto que o VTNT pode recarregar a bateria de um VANT pousado nele. A partir daí, se houver um VANT sobre um VTNT e um segundo VANT se aproximar do robô terrestre, alertando bateria fraca, é feita uma troca de posição entre os VANTs. No segundo caso, se um VTNT estiver com defeito e houver um VANT funcional que o acompanha, a aeronave irá romper sua formação atual e será incorporada a outras formações que estejam executando outras missões.

3.1 Caracterização da Formação

Esta seção apresenta a estrutura virtual descrita por uma forma triangular capaz de controlar quaisquer três robôs móveis, em particular estruturas virtuais envolvendo dois VANTs e um VTNT. As variáveis de estado que representam a estrutura virtual são mostradas na Figura 15. A posição da formação é definida por $\xi_F = [x_F \ y_F \ z_F]^T$ e sua orientação por $\eta_f = [\theta_F \ \phi_F \ \psi_F]^T$. A forma da estrutura triangular é definida por $\sigma_F = [p_F \ q_F \ \beta_F]^T$, onde p_F e q_F são os comprimentos dos lados $\overline{R_1R_2}$ e $\overline{R_1R_3}$ do triângulo formado considerando os robôs como vértices, e β_F é o ângulo $\widehat{R_2R_1R_3}$. Toda a estrutura virtual é representada por $\mathbf{q} = [\xi_F^T \ \eta_f^T \ \sigma_F^T]^T$. Por definição, a posição da estrutura virtual ξ_F é a própria posição do robô R_1 .

Figura 15 – Formação Triangular Heterogênea do tipo estrutura virtual.



3.1.1 Transformações Direta e Inversa da Formação

Um novo espaço referencial surge ao se criar uma estrutura virtual, e é necessário encontrar a relação entre as posições dos robôs com a estrutura virtual em si, conhecida como transformação direta. A posição da estrutura, conforme afirmado acima, é equivalente

à posição do robô R_1 , dada por

$$\boldsymbol{\xi}_F = [x_1 \quad y_1 \quad z_1]^T. \quad (3.1)$$

Para a forma da estrutura virtual, tem-se

$$\boldsymbol{\sigma}_F = \begin{bmatrix} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \\ \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2} \\ \arccos\left(\frac{p_F^2 + q_F^2 - r_F^2}{2p_F q_F}\right) \end{bmatrix}, \quad (3.2)$$

onde

$$r_F = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2}. \quad (3.3)$$

Para obter a orientação da estrutura, Seja N o vetor normal ao plano que contém a estrutura virtual, dado por $N = (N_x, N_y, N_z) = \overrightarrow{R_1 R_2} \times \overrightarrow{R_1 R_3}$. Daí se obtém $\phi_F = \arctan\left(\frac{\|N_{x,y}\|}{N_z}\right) - \frac{\pi}{2}$ e $\psi_F = \frac{\pi}{2} + \arctan\left(\frac{N_y}{N_x}\right)$. Sejam, também $\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_{\phi_F} & -S_{\phi_F} \\ 0 & S_{\phi_F} & C_{\phi_F} \end{bmatrix}$

a matriz de rotação em torno do eixo- x e $\mathbf{R}_z = \begin{bmatrix} C_{\psi_F} & -S_{\psi_F} & 0 \\ S_{\psi_F} & C_{\psi_F} & 0 \\ 0 & 0 & 1 \end{bmatrix}$ a matriz de rotação em torno do eixo- z . Daí se obtém que $\theta_F = \arctan\left(\frac{G_x}{G_z}\right)$, onde $G = (\mathbf{R}_x \mathbf{R}_z)^{-1} \overrightarrow{R_1 R_2}$.

Assim, a orientação

$$\boldsymbol{\eta}_F = \begin{bmatrix} \arctan\left(\frac{G_x}{G_z}\right) \\ \arctan\left(\frac{\|N_{x,y}\|}{N_z}\right) - \frac{\pi}{2} \\ \frac{\pi}{2} + \arctan\left(\frac{N_y}{N_x}\right) \end{bmatrix} \quad (3.4)$$

é obtida.

Conseqüentemente, é possível representar a transformação direta da estrutura virtual, dada por $\mathbf{q} = f(\mathbf{x})$, onde $\mathbf{x} = [\boldsymbol{\xi}_1^T \quad \boldsymbol{\xi}_2^T \quad \boldsymbol{\xi}_3^T]^T$ é o vetor referente às posições dos três robôs pertencentes à estrutura virtual. Note que para descrever a formação não se faz necessária a informação da postura dos robôs.

O próximo passo é relacionar a estrutura à posição dos robôs, chamada de transformação inversa. Assumindo que $\mathcal{R} = \mathbf{R}_z \mathbf{R}_x \mathbf{R}_y$ onde, $\mathbf{R}_y = \begin{bmatrix} C_{\theta_F} & 0 & -S_{\theta_F} \\ 0 & 1 & 0 \\ S_{\theta_F} & 0 & C_{\theta_F} \end{bmatrix}$ é a matriz

de rotação em torno do eixo- y , $\mathbf{P}_F = [0 \ 0 \ p_F]^T$ e $\mathbf{Q}_F = [q_F S_{\beta_F} \ 0 \ q_F C_{\beta_F}]^T$, se obtém que,

$$\mathbf{x} = f^{-1}(\mathbf{q}) = \begin{bmatrix} \boldsymbol{\xi}_F \\ \mathcal{R}\mathbf{P}_F + \boldsymbol{\xi}_F \\ \mathcal{R}\mathbf{Q}_F + \boldsymbol{\xi}_F \end{bmatrix}. \quad (3.5)$$

A transformação é usada para representar a posição dos robôs em uma estrutura virtual e vice-versa. Porém, ao se trabalhar com velocidades, faz-se necessário usar a matriz Jacobiana para que se obtenha a relação cinemática no espaço dos robôs e no espaço da formação. Com isso obtém-se $\dot{\mathbf{q}} = \mathbf{J}(\mathbf{x})\dot{\mathbf{x}}$ e $\dot{\mathbf{x}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{q}}$, onde,

$$\mathbf{J}(\mathbf{x}) = \frac{\partial \mathbf{q}_{n \times 1}}{\partial \mathbf{x}_{m \times 1}} \text{ e } \mathbf{J}^{-1}(\mathbf{q}) = \frac{\partial \mathbf{x}_{m \times 1}}{\partial \mathbf{q}_{n \times 1}}, \quad (3.6)$$

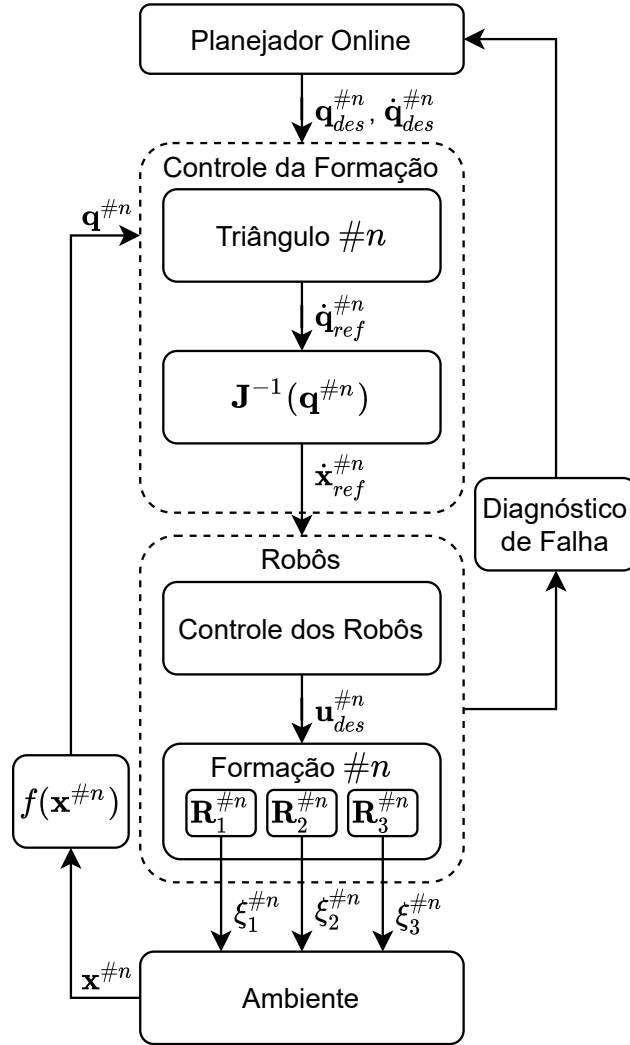
para $m, n = 1, 2, \dots, 9$.

Ao lidar com tarefas envolvendo dois robôs (um VTNT e um VANT), uma formação em linha é suficiente, além de ser simples. Porém, a formação triangular tem potencial para trazer outras propriedades, conforme afirmado por Mas, Petrovic e Kitts (2008). Além disso, para uma formação de dois robôs, pode-se considerar a existência de um robô virtual para compor a formação triangular, conforme proposto a seguir.

Para satisfazer o controle de múltiplas formações, aplicaram-se os conceitos de trabalhos anteriores (BRANDAO et al., 2014; BRANDÃO et al., 2015). A Figura 16 descreve a rotina de controle usada por n formações ou n estruturas virtuais. Essa rotina é dividida em duas camadas, denominadas Formações e Robôs. Primeiro, a camada de formação controla o grupo de robôs que se dispõem no formato triangular. Ele recebe a posição e velocidade desejada do planejador *online*, bem como a posição atual do robô do ambiente. Em seguida, o controlador calcula as velocidades dos robôs, para reduzir o erro de formação existente. Na sequência, a camada dos robôs é responsável por guiar todos os robôs de acordo com o sinal de comando recebido da camada de formação. Nesse ponto, cada robô usa seu próprio controlador cinemático e calcula seu próprio sinal de controle. Ao receber tal sinal de controle, os robôs executam o movimento no ambiente de trabalho, que alimenta o sistema com a posição atualizada, a qual irá se refletir na formação a que o robô pertence.

O sinalizador de falha é atualizado se um robô detectar bateria fraca, falha mecânica ou de comunicação. Tais informações retroalimentam o planejador *online*, por meio do diagnóstico de falhas, que recalcula a posição desejada dos robôs em função deste novo cenário.

Figura 16 – O esquema de controle multicamadas tolerante a falhas.



3.1.2 Controle da Formação e Análise de Estabilidade

Primeiramente, é importante dizer que o planejador de missão está programado na camada *online* do controle multicamadas, onde \mathbf{q}_{des} é a posição desejada e $\dot{\mathbf{q}}_{des}$ é a primeira derivada temporal de \mathbf{q}_{des} . No início da etapa de controle, o controlador da formação recebe a posição inicial \mathbf{q}_{init} . Depois disso, o laço é realimentado com as características atuais da formação \mathbf{q} . Com essas variáveis, o erro existente entre a formação atual e a desejada $\tilde{\mathbf{q}} = \mathbf{q}_{des} - \mathbf{q}$ é calculado. Por sua vez, a referência para o controlador da formação é dada por

$$\dot{\mathbf{q}}_{ref} = \dot{\mathbf{q}}_{des} + \mathbf{K}_1 \tanh(\mathbf{K}_2 \tilde{\mathbf{q}}), \quad (3.7)$$

onde $\dot{\mathbf{q}}_{ref}$ é a velocidade de referência para formação minimizar o erro $\tilde{\mathbf{q}}$. A razão para usar a função $\tanh(\cdot)$ é limitar suavemente o sinal de controle para grandes erros de formação $\tilde{\mathbf{q}}$. Vale ressaltar que quando \mathbf{q}_{des} não tem variação temporal, ou seja, o resultado da derivada temporal é zero, com $\dot{\mathbf{q}}_{des} = \mathbf{0}$, a missão de controle é de posicionamento (ou regulação). As matrizes diagonais definidas positivas \mathbf{K}_1 e \mathbf{K}_2 representam os ganhos do controlador,

de modo que $\{\mathbf{K}_1, \mathbf{K}_2\} \in \mathbb{R}^{9 \times 9}$

Considerando um seguimento perfeito de velocidade da formação, tem-se que $\dot{\mathbf{q}} = \dot{\mathbf{q}}_{ref}$ e, conseqüentemente, a Equação (3.7) será reescrita como

$$\dot{\tilde{\mathbf{q}}} + \mathbf{K}_1 \tanh(\mathbf{K}_2 \tilde{\mathbf{q}}) = \mathbf{0}. \quad (3.8)$$

Para analisar a estabilidade do controle de formação, considera-se a função candidata de Lyapunov globalmente limitada da classe C^2 e definida positivamente dada por

$$V = \frac{1}{2} \tilde{\mathbf{q}}^T \tilde{\mathbf{q}}, \quad (3.9)$$

Tomando sua primeira derivada temporal e substituindo (3.8), tem-se

$$\dot{V} = \tilde{\mathbf{q}}^T \dot{\tilde{\mathbf{q}}} = -\tilde{\mathbf{q}}^T \mathbf{K}_1 \tanh(\mathbf{K}_2 \tilde{\mathbf{q}}) < 0, \quad (3.10)$$

pois as matrizes \mathbf{K}_1 e \mathbf{K}_2 são definidas positivas e $\tanh(\cdot)$ é uma função ímpar. Logo, é possível concluir que $\tilde{\mathbf{q}} \rightarrow 0$, para $t \rightarrow \infty$.

3.2 Estratégia Tolerante a Falhas

Em missões de fiscalização em longas distâncias podem ocorrer falhas no sistema, como descarga de baterias, perda de informações, falha mecânica, entre outras. O controle de formação pode lidar e superar essas situações aplicando estratégias de detecção, análise e supressão de falhas.

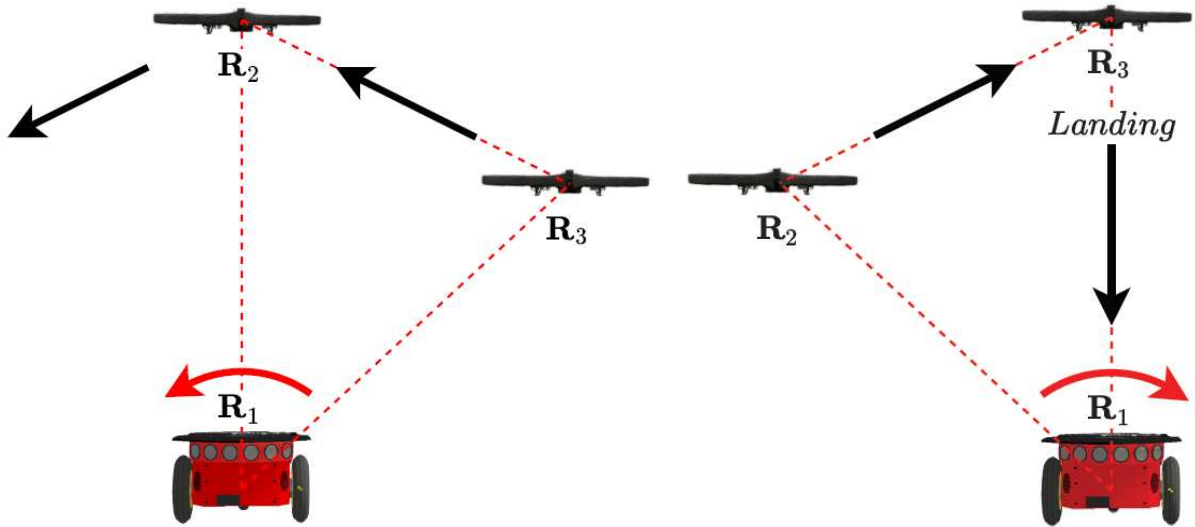
Comparativamente, o ciclo da bateria de um VANT é menor do que o de um VTNT. Portanto, existem preocupações com a descarga de baterias para VANTs. Por sua vez, um VTNT é capaz de recarregar um VANT. Sendo assim, para mitigar a situação de descarga de bateria do VANT, ele pode pousar sobre o VTNT, para iniciar a fase de recarga. Caso um VTNT já estiver transportando um VANT, o VANT carregado decola e dá lugar àquele em voo. Nesse caso, os VANTs executaram uma estratégia de troca, usando a formação triangular.

Para o caso em que ambas as aeronaves estejam em voo, as Figuras 17 e 18 mostram como os VANTs R_2 e R_3 são trocados, usando a formação triangular. Neste trabalho, foram propostas duas maneiras para a execução da troca. Para explicar cada uma delas, faz-se necessário ilustrar o cenário imediatamente antes do alerta de bateria fraca. Em resumo, R_2 está pairando sobre R_1 , quando recebe o alerta de falha de R_3 . O processo de troca é iniciado com o afastamento de R_2 para possibilitar o pouso de R_3 sobre R_1 .

A Figura 17 ilustra a primeira estratégia de troca. Assumindo que os VANTs estão voando, a Figura 17a ilustra os movimentos que os VANTs devem realizar para evitar a colisão. Em outras palavras, R_3 toma o lugar de R_2 , que se afasta. Matematicamente, a

Figura 17 – Primeira estratégia de troca de posição de VANTS: triângulo isósceles.

- (a) Movimentos iniciais, evitar a colisão e posicionar R_3 em cima de R_1 . (b) Movimento final, pouso de R_3 e retorno de R_2 .



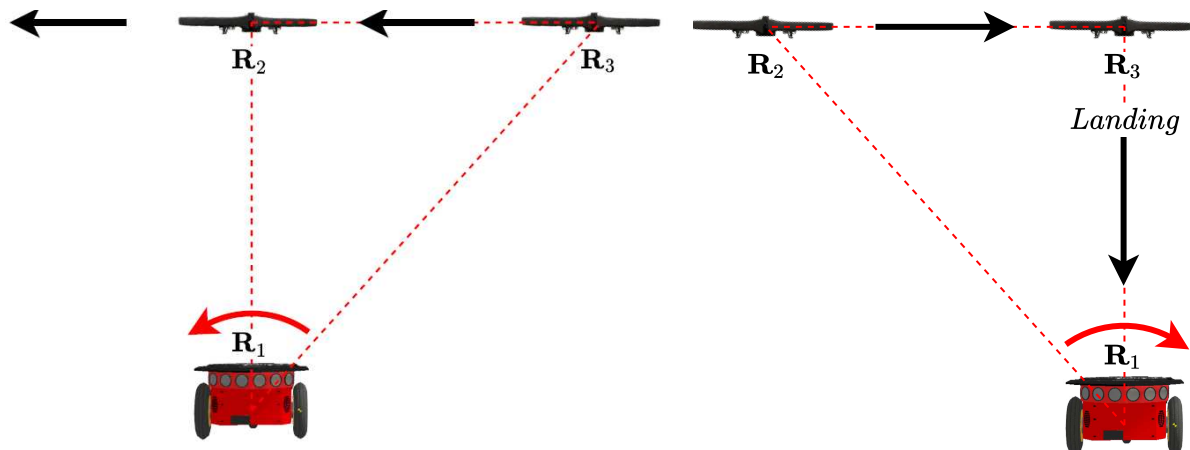
formação triangular rotaciona θ_F , representado pela seta vermelha, e os VANTS alcançam a posição mostrada na Figura 17b. Neste momento, R_3 se prepara para pousar no VTNT, enquanto R_2 retorna à posição inicial.

A Figura 18 ilustra a segunda estratégia. Nesse caso, os VANTS estão na mesma altitude e as etapas de troca são análogas à primeira estratégia.

Considerando agora uma falha mecânica ou de comunicação, elas podem ocorrer com qualquer robô da formação. Se ocorrer uma falha de VANT, um VANT virtual o substitui, até que o VANT com falha se recupere. Entretanto, em caso de dano permanente, o VANT virtual ou outro VANT real toma o lugar do VANT com falha para dar prosseguimento ao

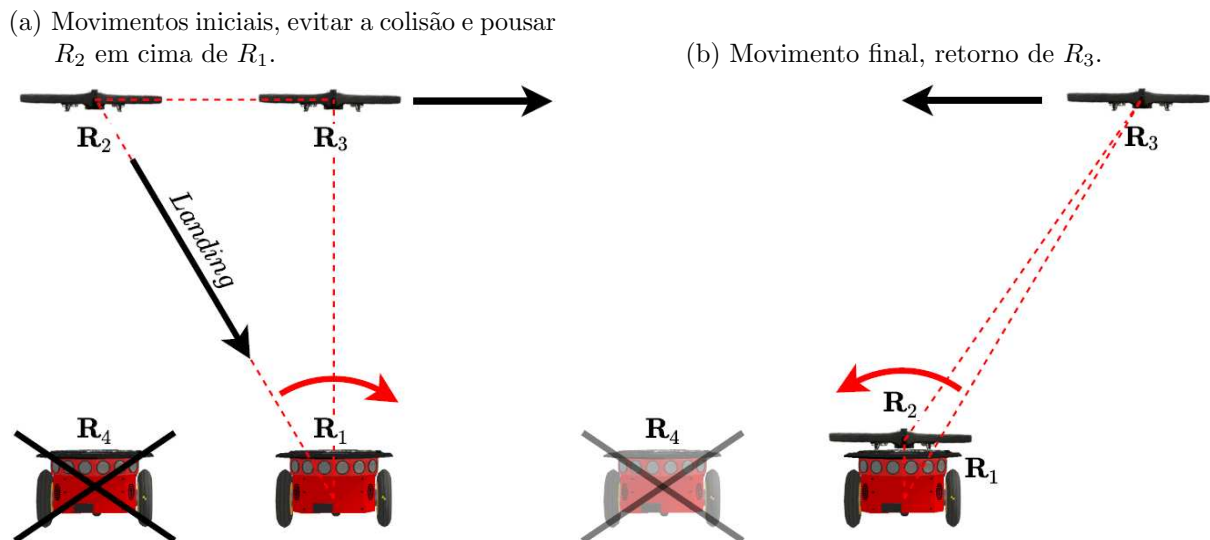
Figura 18 – Segunda estratégia de troca de posição dos VANTS: mesma altitude.

- (a) Movimentos iniciais, evitar a colisão e posicionar R_3 em cima de R_1 . (b) Movimento final, pouso de R_3 e retorno de R_2 .



cumprimento da missão. Caso a falha ocorra no VTNT, o VANT remanescente procura a formação funcional mais próxima, solicita permissão para compô-la e, finalmente, pousa no VTNT. A Figura 19 ilustra a estratégia tolerante a falhas para falhas em um VTNT.

Figura 19 – Estratégia tolerante a falhas para falhas no VTNT.



3.3 Descrição do Experimento

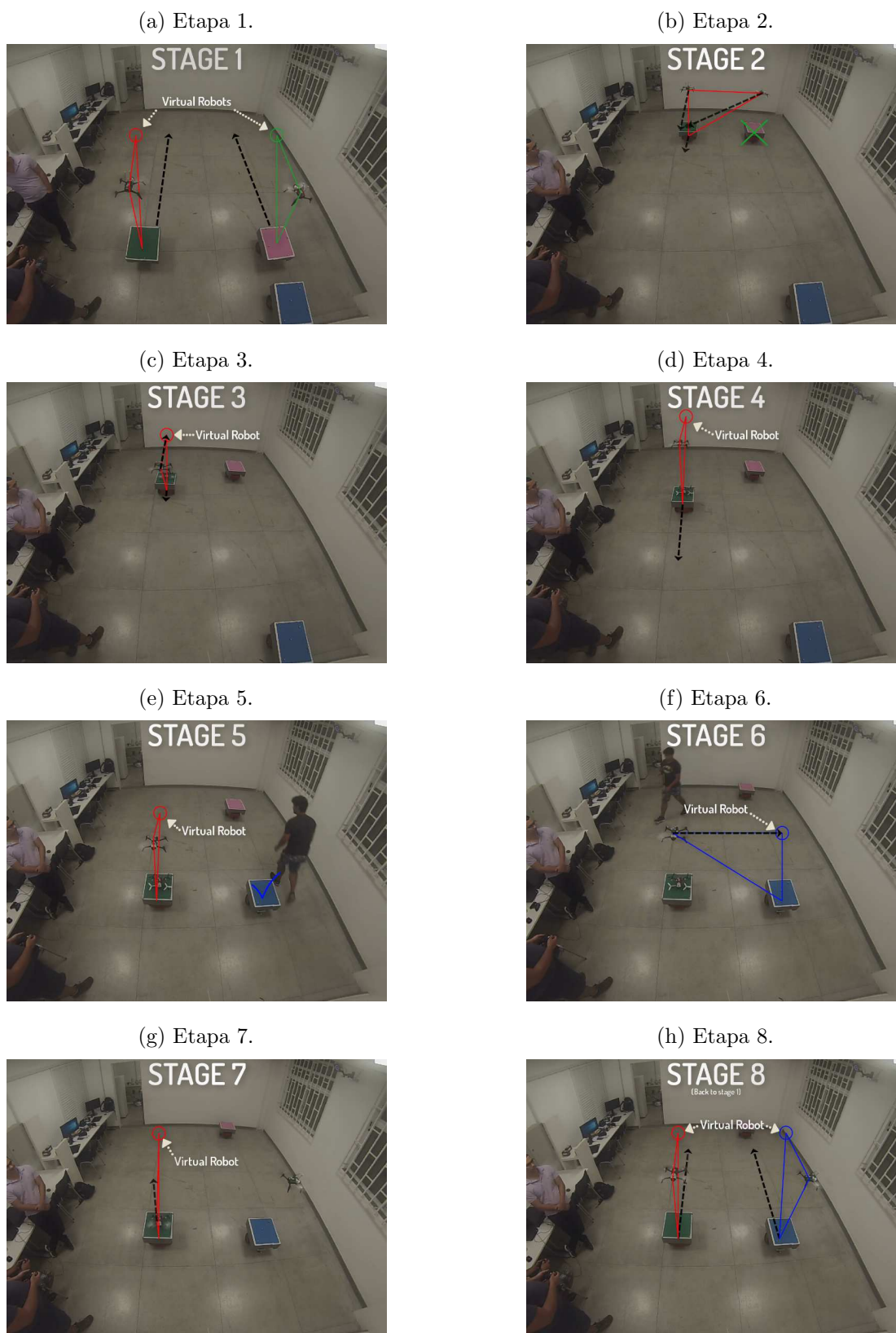
Esta seção explica como são realizados os experimentos, bem como a missão a ser cumprida, a quantidade de robôs e o controlador que orienta a formação. Todos os experimentos foram realizados no laboratório do NERO (Núcleo de Especialização em Robótica), da UFV. O ambiente utilizado para a validação possui um sistema *OptiTrack* composto por oito câmeras. A plataforma de comunicação e controle foi previamente validada utilizando a plataforma AuRoRA (PIZETTA; BRANDAO; SARCINELLI-FILHO, 2016) e comunicação ROS. Os testes foram executados com três VTNTs *Pioneer3DX* da *Adept MobileRobots* e quatro VANTs da *Parrot Inc* (dois *AR.Drone 2.0* e dois são *Bebop 2*).

A missão utilizada para validar a proposta é a de inspeção, em que duas formações triangulares realizam um deslocamento longitudinal no eixo- y . A Figura 20 apresenta todas as etapas da missão durante o experimento, a quais são detalhadas a seguir:

- **Etapa 1**, Figura 20a: Trajetória sendo representada pela seta preta tracejada, representação gráfica das duas formações triangulares;
- **Etapa 2**, Figura 20b: O VTNT da formação triangular verde sofre uma falha e o VANT substitui o robô virtual da formação vermelha. Em seguida, realiza uma manobra de pouso em cima do VTNT, enquanto o outro VANT se afasta para evitar colisões;

- **Etapa 3**, Figura 20c: O VANT que se distanciou para evitar colisões recupera sua posição anterior;
- **Etapa 4**, Figura 20d: O VTNT continua sua trajetória de retorno à posição inicial, definida na Etapa 1;
- **Etapa 5**, Figura 20e: O VTNT com falha é substituído por um novo VTNT e uma nova formação triangular é criada;
- **Etapa 6**, Figura 20f: O VANT voador muda para o novo VTNT e integra a nova formação;
- **Etapa 7**, Figura 20g: O VANT que pousou, decola e substitui o último VANT;
- **Etapa 8**, Figura 20h: Uma nova inspeção é iniciada com os mesmos objetivos da Etapa 1.

Figura 20 – Etapas do experimento de validação do trabalho tolerante a falhas proposto.



3.4 Resultados Experimentais e Discussão

Esta seção apresenta experimentos reais para validar a proposta, primeiro considerando a estratégia tolerante a falhas para baterias descarregadas (Subseção 3.4.1) e depois para falhas mecânicas ou de comunicação (Subseção 3.4.2).

3.4.1 Estratégia Tolerante a Falhas para Descarregamento de Bateria

O experimento aqui executado está dividido em duas partes, que se referem às estratégias ilustradas nas Figuras 17 e 18.

3.4.1.1 Primeiro Tipo: Triângulo Isóceles

O cenário projetado para emular uma situação de descarga de bateria é descrito por um VANT a bordo do VTNT e outro VANT trabalhando cooperativamente com este VTNT. Decidiu-se não mover o VTNT para facilitar a visualização da estratégia de mudança.

Quando o VANT voador seta sua *flag* de bateria fraca, ele envia à formação um pedido para pousar e recarregar. Nesse momento, a estratégia de troca se inicia. O VANT voador vai para a direita, permitindo que o VANT pousado dê um *take-off*. Uma vez voando, ele vai para a esquerda e, de forma coordenada, o outro VANT também se move para a esquerda, pousando sobre o VTNT. Por fim, o VANT que está voando vai para a direita e para sobre o VTNT.

Para comparar o tempo gasto necessário durante a estratégia de troca, simulou-se um cenário, cuja postura da estrutura virtual é $\sigma_F = [1,4\ m\ 1,4\ m\ 60^\circ]^T$, e outro, com $\sigma_F = [1,2\ m\ 1,2\ m\ 60^\circ]^T$. Eles gastam aproximadamente $T_{Exc} = 37,5\ s$ e $T_{Exc} = 8\ s$, respectivamente, e as manobras podem ser vistas em <https://youtu.be/kmFnZeICU3M>.

Vale ressaltar que quanto mais rápido a troca ocorrer, maior será o tempo de consumo da bateria. Portanto, isso deve ser levado em consideração para missões longas que requerem muitas recargas de bateria.

3.4.1.2 Segundo Tipo: Mesma Altitude

A configuração experimental é semelhante à mencionada anteriormente. No entanto, a troca ocorre com os VANTs na mesma altitude. Eles se movem lateralmente para dar lugar um ao outro. Neste caso, a postura da estrutura virtual é $\sigma_F = [1,2\ m\ 1,697\ m\ 45^\circ]^T$ e o tempo gasto para trocar posições é $T_{Exc} = 30\ s$. O experimento está disponível em <https://youtu.be/kmFnZeICU3M?t=120>.

Embora intuitiva, a estratégia de mesma altitude não tem um desempenho melhor do que a anterior, em termos de tempo gasto. Além disso, apresenta uma desvantagem de

segurança, uma vez que os VANTs podem colidir durante o deslocamento lateral, caso ocorra um *overshoot* do sinal de controle. Em contraste, na estratégia do triângulo isósceles, os VANTs executam um deslocamento de arco, onde o risco de colisão é menor.

3.4.2 Missão de Inspeção

Nesta seção, uma missão de inspeção foi proposta para ilustrar um cenário de tolerância a falhas. Todo o experimento, inclui breves explicações de cada estágio mostrado na Figura 20 e está disponível em <<https://youtu.be/9MLioHdTWZY>>. Os estágios do experimento são representados na Figura 21 em que as cores vermelho, verde e azul, representam as formações triangulares #1, #2 e #3, respectivamente. A Tabela 1 mostra as informações de cada estágio, bem como as formações ativas, os VANTs ativos, o tempo gasto para concluir uma troca e o tempo total dos sub-experimentos.

As Figuras 22, 23 e 24 mostram os dados de cada formação triangular, #1, #2 e #3, respectivamente, apresentando seus erros.

A falha ocorre no VTNT da formação #2. O VANT desta formação pede para aterrissar no VTNT da formação #1. Depois de um tempo, o VTNT danificado é substituído por outro, criando a formação #3. Observe que a formação #1 permanece ativa durante todo o experimento, enquanto que a formação #2 se encerra em 30 s e a formação #3 se torna ativa a partir de 75 s aproximadamente. Da Equação (3.1), vale lembrar que a posição de formação também é a posição do VTNT que lhe pertence.

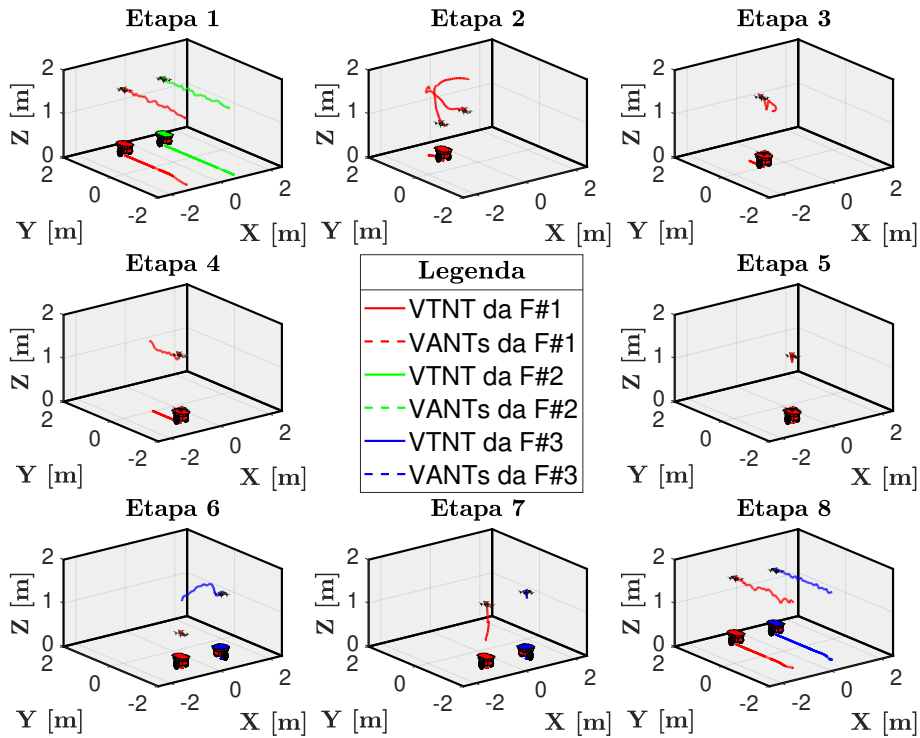
A Figura 25 mostra os erros de rastreamento para cada VANT. Durante o experimento, o VANT #1 está ativo o tempo todo, enquanto o VANT #2 precisa pousar por um tempo (região hachurada no gráfico) e, quando possível, retoma voo para ocupar o espaço deixado pelo VANT #1 na formação #1 ao mudar para nova formação #3.

O experimento validou a estratégia proposta tolerante a falhas, abordando problemas mecânicos ou de comunicação. Analisando os gráficos de formação, os erros em todos os

Tabela 1 – Configuração de cada etapa durante a missão de inspeção.

Etapas	Formações Ativas	VANTs Ativos	Tempo das Etapas (s)	Tempo do Experimento (s)
1	#1 e #2	#1 e #2	30	30
2	#1	#1 e #2	7.5	37.5
3	#1	#1	7.5	45
4	#1	#1	15	60
5	#1	#1	15	75
6	#1 e #3	#1	15	90
7	#1 e #3	#1 e #2	7.5	97.5
8	#1 e #3	#1 e #2	30	127.5

Figura 21 – Rastro dos robôs para cada estágio da missão de inspeção.



casos giram em torno de 0, 20 m, demonstrando uma resposta rápida mesmo na transição de etapas. Isso mostra a capacidade de resposta da abordagem.

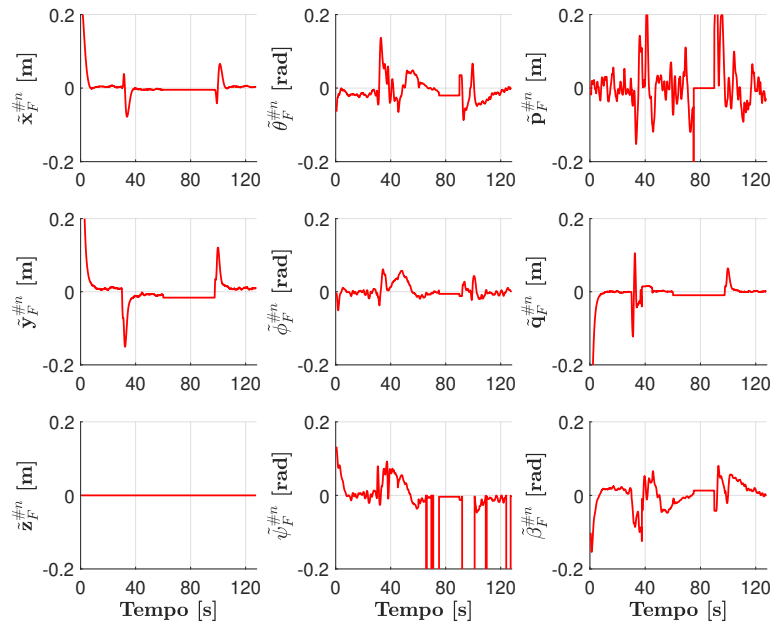
Em relação à Figura 25, observe que os erros oscilam em torno de 0, 20 m em valor absoluto, mostrando que os VANTs respondem rapidamente à mudança de etapas. No estágio 2, por exemplo, pode-se observar que, com a chegada do VANT #2, a formação mantém uma “zona de segurança” entre os VANTs, resultando em um deslocamento do VANT #1 para permitir a aterrissagem do VANT #2. Esta manobra evita a colisão do VANT e atenua a turbulência causada pelas hélices de um VANT para outro, o que torna o VANT #2 instável e dificulta seu pouso.

3.5 Considerações Finais

Este capítulo propôs e validou experimentalmente uma estratégia tolerante a falhas para formações triangulares. Em linhas gerais, foram explorados dois tipos de falhas, as quais envolvem a descarga da bateria de um VANT e a falha mecânica ou de comunicação de um VTNT.

Para a estratégia tolerante a falhas envolvendo o VANT, obteve excelentes resultados usando um VANT de baixo custo para o primeiro tipo de estratégia, o triângulo isósceles. Para o segundo tipo, de mesma altitude, o resultado obtido não foi o esperado, pois, apesar

Figura 22 – Erros para cada parâmetro da formação #1 que funcionou durante a primeira etapa.



de ser o mais intuitivo, há um maior risco de colisão entre os VANTs durante a transição.

Durante a missão de inspeção utilizada para validar a proposta, notou-se uma maior oscilação no parâmetro \mathbf{p}_F , que representa a distância entre o VANT e o VTNT, quando comparado com os demais parâmetros. Concluiu-se que ajustes de ganho e aprimoramento na comunicação podem resolver essa oscilação, ficando então como sugestão para trabalhos futuros.

Por fim, concluiu-se que a formação triangular funciona corretamente na presença de um agente virtual, possibilitando que um agente real entre numa formação previamente definida, em caso de falhas, e toda a formação continue trabalhando sem comprometer o cumprimento da missão. Tal vantagem abre espaço para aplicações com agentes heterogêneos e homogêneos.

Figura 23 – Erros de cada parâmetro da formação #2 que funcionou ao longo das etapas iniciais.

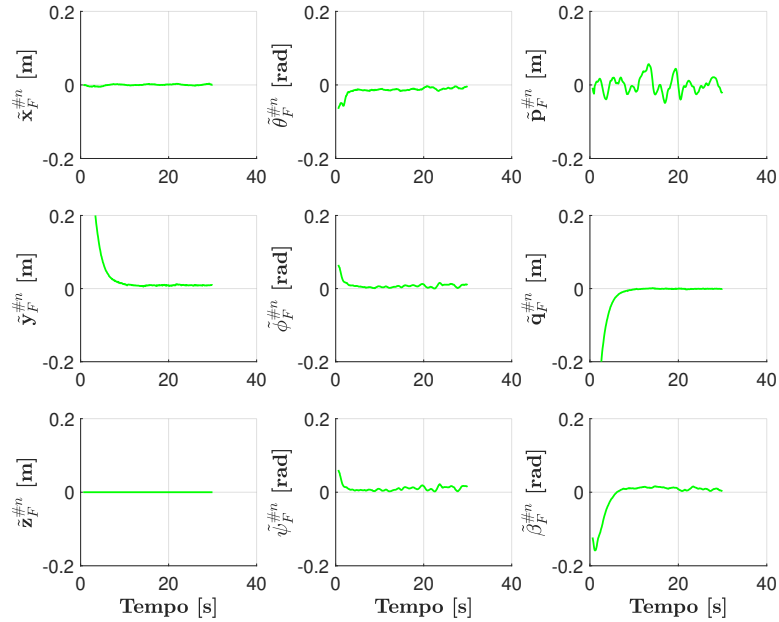


Figura 24 – Erros para cada parâmetro da formação #3 funcional desde o estágio cinco até o final do experimento.

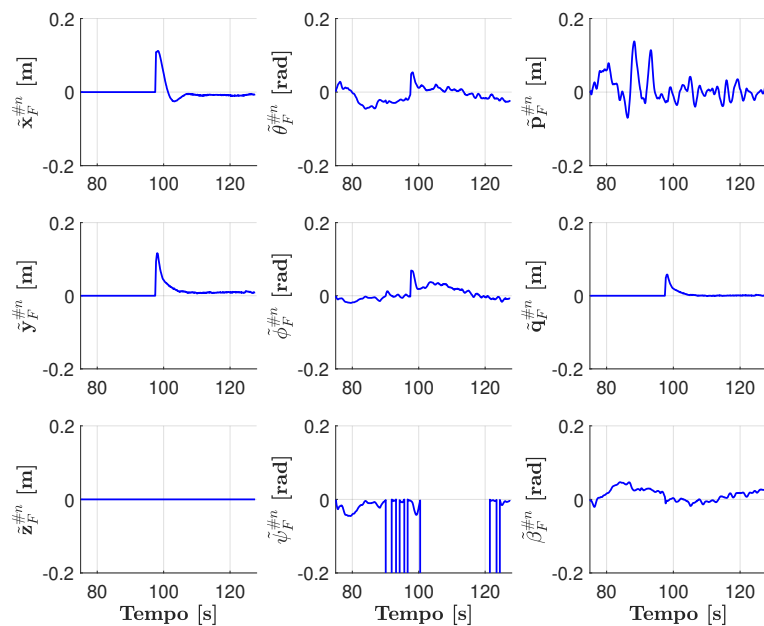
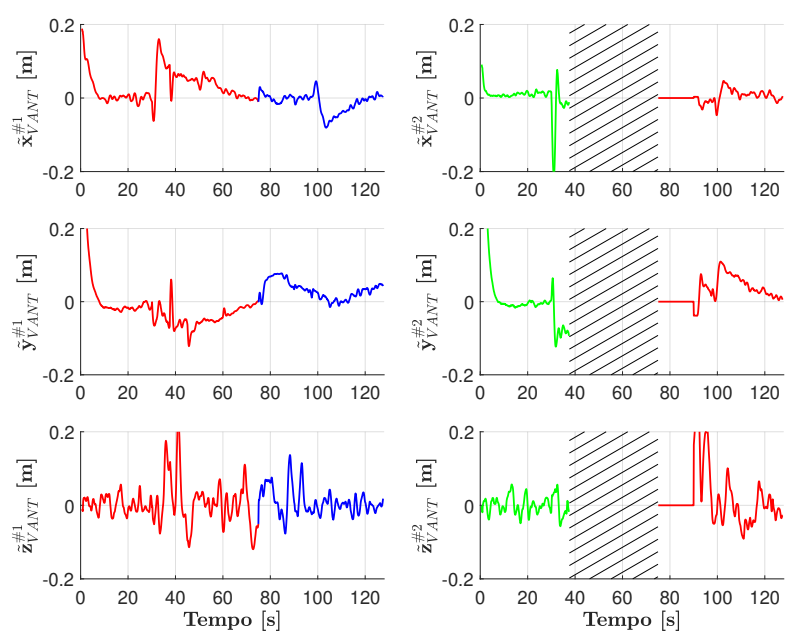


Figura 25 – Erros de posição: VANT #1 (esquerda) e #2 (direita). A região hachurada representa o VANT #2 pousado no VTNT #1.



4 (Re)planejamento de rotas com estratégia tolerante a falhas

A área de cooperação de robôs vem chamando a atenção de pesquisadores, pois a solução de problemas tem tido melhor desempenho, visto que, ao se trabalhar com um único robô, o gasto de energia ou bateria, o tempo de execução do processo, a distribuição de sensores, as suas limitações físicas de limite de carga máxima são ressaltadas (ARKIN, 1998). Em contrapartida, ao trabalhar com mais de um robô em cooperação, tais limitações são quebradas. Por exemplo, ao se comparar um grupo de robôs com um único agente com as mesmas configurações, a tarefa de limpeza de um galpão ou mapeamento de um ambiente é mais rapidamente, a carga útil de um pelotão é maior, e assim por diante. Entretanto, ao mesmo tempo que as soluções tem suas limitações ultrapassadas, a complexidade do sistema aumenta proporcional ao número de robôs cooperando. Em outras palavras, o novo desafio e uma das principais dificuldades de um controle multirrobô é a coordenação do movimento dos agentes simultaneamente, onde há maior chance de ocorrer algum tipo de falha (HERNANDEZ et al., 2009).

A fim de deixar os sistemas mais seguros, independente do número de robôs utilizados, existem estratégias tolerantes a falhas, que são capazes de diagnosticar o tipo de falha e aplicar o método para realização do reparo da mesma. Tais métodos podem ser usados para corrigir falhas nos atuadores de VANTs Yu, Li e Zhang (2017), Liu, Jiang e Zhang (2020), Nguyen e Hong (2019), nos sensores de odometria de um VTNT Abci et al. (2020) ou no desbalanceamento das pás de um VANT de pás rotativas Iannace, Ciaburro e Trematerra (2019). Stavrou et al. (2016) propuseram uma estratégia capaz de diagnosticar e corrigir falhas de má performance, falha de comunicação e a falha mecânica de cada roda de um VTNT. No contexto de formação, Shi et al. (2018) propuseram uma estratégia para minimizar o erro global de um pelotão quando algum agente se distancia de sua posição desejada. No caso, a formação ajusta sua postura desejada, a fim minimizar os erros de formação e seguimento de referência.

A movimentação coordenada é outro problema a ser resolvido na cooperação de robôs, seja ela em um ambiente aberto Ghommam et al. (2010), que requerem desvio de obstáculos Leica et al. (2018), Lee e Chwa (2018) ou que priorizem ou possuam hierarquias de missões Moreira, Brandao e Sarcinelli-Filho (2019), Leica et al. (2019).

Para tratar do (re)planejamento de rotas em uma formação com tolerância a falhas, o presente capítulo traz a estratégia proposta na Seção 4.1, apresenta e discute os resultados de simulação na Seção 4.2 e, por fim, elucida as considerações finais na Seção 4.3.

4.1 A Estratégia

Esta seção irá apresentar a estratégia completa formada pela junção das partes discutidas até aqui nesta dissertação. Para fins de organização, a estratégia envolve a) o algoritmo de planejamento de caminho, b) o controle de formação e c) as estratégias tolerante a falhas.

Relembrando o Capítulo 2, o algoritmo utilizado para guiar as formações será o LPA* com a adaptação que faz o uso do algoritmo de Dijkstra para encontrar um “atalho” até a nova rota, caso o VTNT, ou formação, não esteja presente na mesma.

A formação utilizada para a estratégia será a formação do tipo estrutura virtual triangular proposta no Capítulo 3. Cada formação terá dois VANTs, sendo um real e um virtual, e um VTNT, que é o líder da formação. Para a simulação serão utilizadas duas dessa formação, que, se necessário, irão cooperar entre si.

O controlador utilizado para guiar as formações pela rota planejada será o de caminho, onde a rota será dividida em n pontos. Para fazer o caminho guiar as formações, será usado o controle de formação apresentado na Subseção 3.1.2, com $\dot{\mathbf{q}}_{ref} = \dot{\mathbf{q}}_d + \mathbf{K}_1 \tanh(\mathbf{K}_2 \tilde{\mathbf{q}})$ e

$$\begin{cases} \dot{\mathbf{q}}_d = \dot{\mathbf{p}}_{ref} = v_{max} \vec{\mathbf{p}}_k \\ \tilde{\mathbf{q}} = \mathbf{p}_{k+1} - \mathbf{q} \end{cases} \quad (4.1)$$

onde, v_{max} é a velocidade máxima para a referência gerada pelo caminho, $\vec{\mathbf{p}}_k$ é o vetor unitário entre ponto \mathbf{p}_k e \mathbf{p}_{k+1} e $\dot{\mathbf{p}}_{ref}$ é a velocidade de referência gerada pelo caminho.

Para as estratégias tolerante a falhas serão utilizadas as propostas pelo Capítulo 3. A bateria fraca é considerado falha para os VANTs. Daí, um alerta é emitido e o VANT irá pousar no VTNT de sua formação. Quando o local de pouso estiver ocupado por outro VANT, os dois irão realizar a estratégia de troca, para que o VANT com bateria fraca possa ocupar o lugar do VANT que está pousado sobre o VTNT. A falha de comunicação ou mecânica são consideradas para os VTNTs. Caso ocorra, o VANT associado a ele irá deixar sua formação atual e irá buscar aquela mais próxima para auxiliá-la.

Vale mencionar que o ambiente de navegação dessa vez é semi-dinâmico, ou seja, os obstáculos irão aparecer e podem atrapalhar a movimentação das formações. Para isto, será considerado uma falha de VTNT, quando uma formação tiver sua área de missão totalmente obstruída por conta do acréscimo de obstáculos. Como o VTNT continua funcional, foi decidido que o mesmo irá voltar para seu ponto de origem e o VANT irá buscar a formação mais próxima ajudando-a na missão em curso.

4.2 Simulação e Discussão

Esta seção aborda as simulações, os resultados obtidos e a discussão para a navegação de formações heterogêneas, levando em consideração possíveis falhas nos agentes.

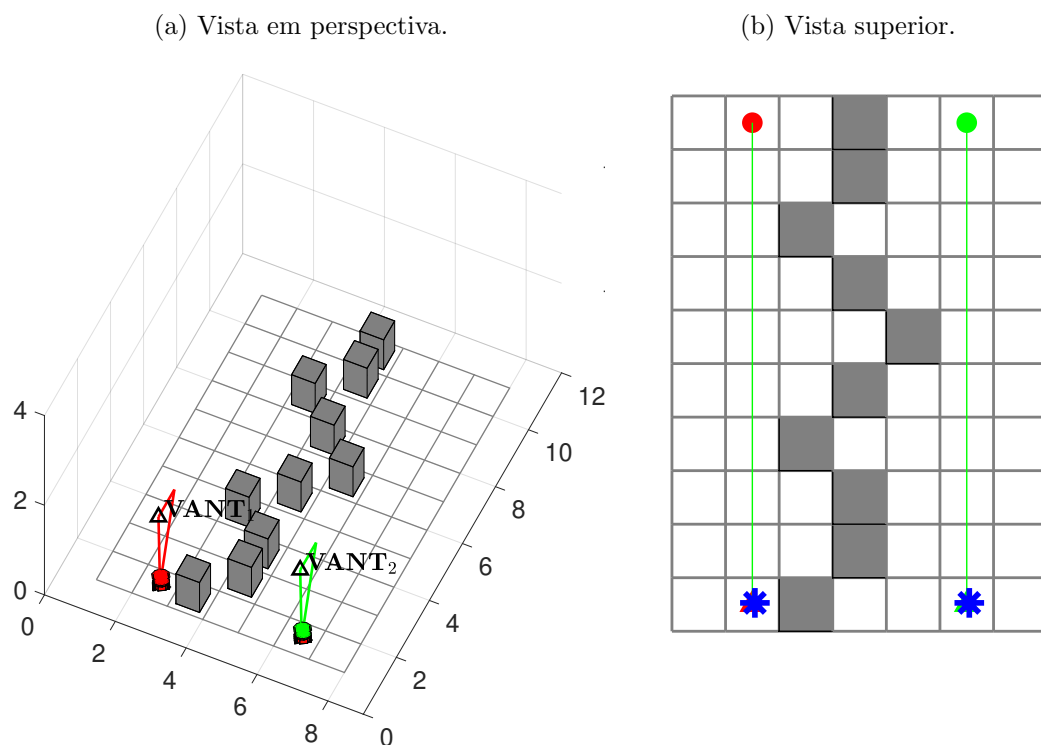
O software MATLAB[®] foi novamente utilizado para a realização das simulações. O ambiente criado é ilustrado pela Figura 26, contém uma grade 7×10 de navegação, cuja vista superior é mostrada na Figura 26b. Esta última facilita a visualização dos obstáculos, os quais são representados pelos cubos de cor cinza.

Existem obstáculos iniciais para dividir o ambiente de cada formação, como visto na Figura 26a. As formações são diferenciadas pelas cores vermelho e verde para as formações #1 e #2, respectivamente. Os VTNTs são representados pelo modelo do Pioneer3DX e na vista superior por um * azul, enquanto os VANTs são representados por um \triangle seguido do texto “VANT_{*n*}”, onde *n* indica sua formação. Vale ressaltar que a vista superior é destinada à visualização do deslocamento do VTNT.

Sobre os traçados da vista em perspectiva, cada formação tem o seu rastro marcado por sua cor de referência. Na vista superior, o rastro dos dois VTNTs são pretos e a rota planejada dos dois VTNTs são verdes. As marcações na vista superior são referentes ao ponto de partida e o ponto de chegada da missão, que são representados, respectivamente, por um triângulo e um círculo.

A vista superior é interativa e responde ao *click* do mouse. A cada ação do usuário,

Figura 26 – Ambiente virtual de simulação no software MATLAB[®].



um obstáculo é inserido na grade selecionada. Ao colocar um obstáculo na vista superior, o mesmo irá aparecer na forma de um cubo na vista em perspectiva. Como é um ambiente virtual não foi tratado o caso de colisão do VANT com os obstáculos na vista em perspectiva, pois o modelo foi definido apenas para estética da simulação. Vale dizer que a altura do obstáculo pode ser alterada via código, deixando o ambiente virtual mais próximo da realidade, contudo isso fica como um tema em aberto para trabalhos futuros.

4.2.1 Simulação 1: Validação do LPA*

Nessa simulação é feita a validação do algoritmo de busca LPA*, usando Dijkstra para criar “atalhos”, quando necessário. A formação é guiada utilizando o controle de caminho a partir da rota planejada. Ao longo da simulação foram adicionados obstáculos, como visto nas Figuras 27 e 28, as quais representam, respectivamente, os instantes da simulação, na vista em perspectiva e na vista superior do ambiente. Um vídeo desta simulação está disponível em <https://youtu.be/la_DiU6kgOQ>.

Nota-se, a partir da simulação, que o algoritmo respondeu bem a aparição dos obstáculos, escolhendo uma nova rota sempre que algum desses obstáculos obstruíssem a rota atual. Como são duas rotas distintas, o algoritmo mostrou ser capaz de realizar o cálculo para duas rotas distintas de forma ótima. No contexto do controle de formação, pode-se afirmar que o VANT acompanhou seu respectivo VTNT durante toda a missão, conforme esperado.

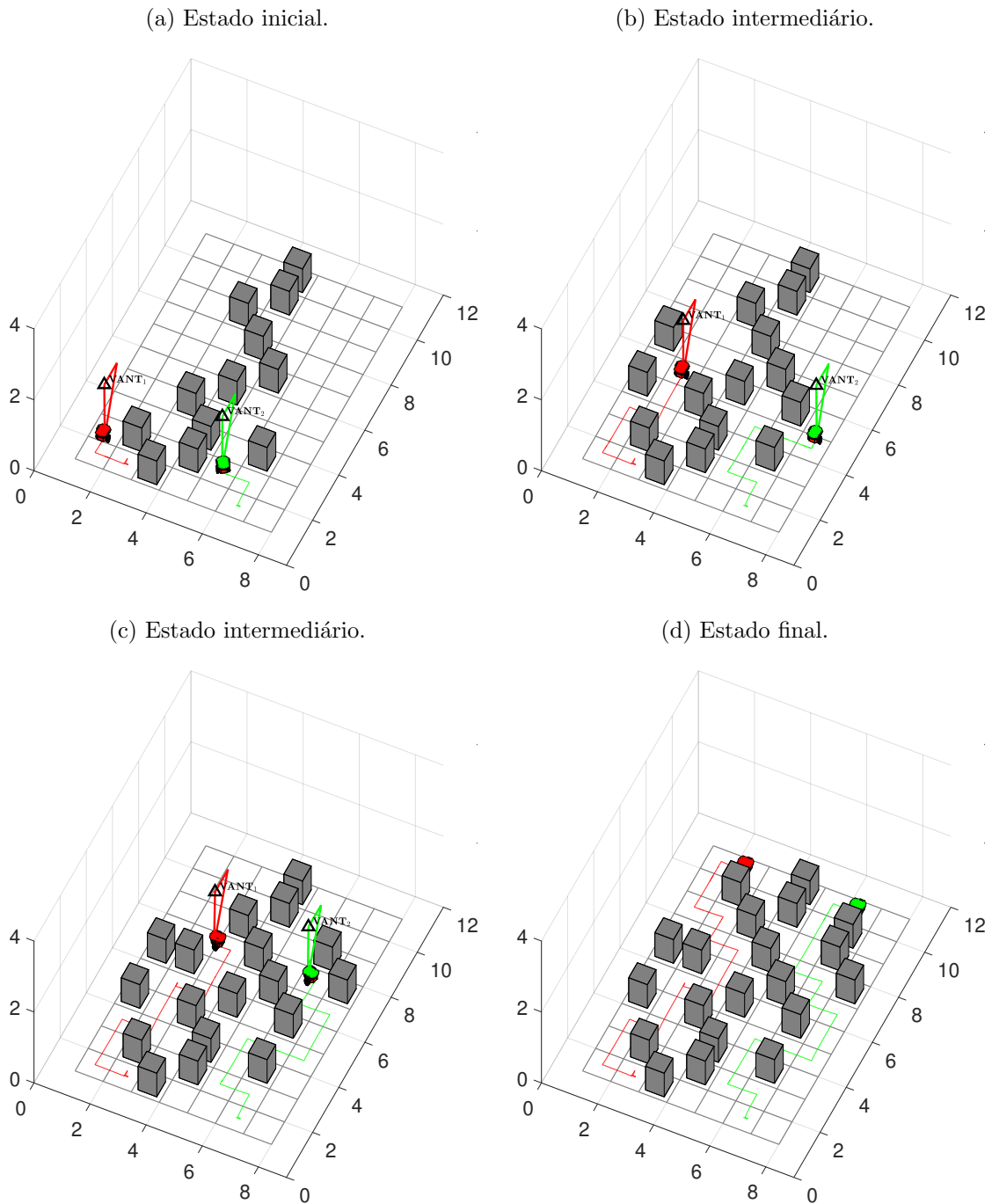
4.2.2 Simulação 2: Validação da Estratégia Tolerante a Falhas

A segunda simulação valida a estratégia tolerante a falha para rota impossível, a qual acontece quando a formação #2 não consegue determinar um caminho factível até o destino. Daí, na ocorrência deste evento, a formação é desfeita, o VTNT retorna para a posição de partida e o VANT procura a formação mais próxima para auxiliá-la no restante da missão. O vídeo descritivo desta simulação está disponível em <https://youtu.be/la_DiU6kgOQ?t=60>.

Ao longo da simulação foram adicionados obstáculos e a rota da formação #2 foi totalmente obstruída, conforme ilustrados nas vistas em perspectiva e de topo das Figuras 29 e 30, respectivamente.

Inicialmente, cada formação é estabelecida por um VTNT e um VANT, como pode ser visto na Figura 29a. Instantes após o início da navegação, a formação #2 tem sua rota bloqueada e a estratégia de troca acionada. Conforme planejamento da estratégia, o VANT^{#2} requisita a permissão para participação na formação #1 e, após confirmação, substitui o VANT virtual. A Figura 29b mostra o VANT preparando para pousar no VTNT^{#1}, enquanto que o VTNT^{#2} volta para a posição de origem. Na sequência, a Figura

Figura 27 – Instantâneos da vista em perspectiva da simulação de validação do algoritmo obtido no Capítulo 2.



29c mostra que o VANT^{#2} realizou o pouso e foi substituído por um VANT virtual, para que a formação #1 continuasse sua missão. Caso o VANT^{#1} que está voando precise de uma recarga bateria, o alerta será acionado, e ele irá trocar sua posição com o VANT^{#2} que está pousado, utilizando a estratégia proposta no Capítulo 3. A Figura 29d mostra o fim da missão, que é quando a formação #1 atinge sua posição de destino. Vale destacar que a formação #2 não completa sua missão, entretanto garante-se a segurança dos equipamentos.

Para finalizar, a Figura 30 ilustra a vista de topo com a navegação do VTNT de cada formação. Como já foi dito, a formação #2 foi desfeita e seu VTNT retorna para a posição de partida, enquanto a formação #1 segue a busca pela posição de destino. Vale frisar que o algoritmo de buscas funcionou perfeitamente guiando as formações, quando possível, e provendo alternativas seguras aos agentes, caso contrário.

4.3 Considerações Finais

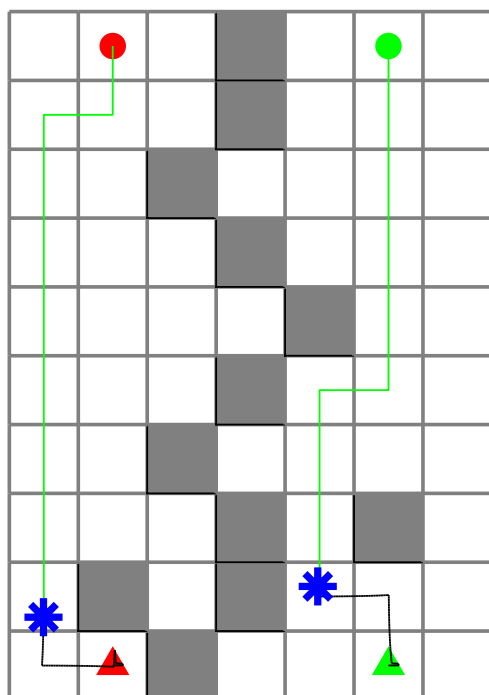
Este capítulo apresentou uma missão estratégica, para uma formação composta por um VTNT e dois VANTs, sendo um real e um virtual. Estratégias tolerante a falhas garantiram a eficiência, estabilidade e segurança da missão, conforme mostrado durante a etapa de validação numérica.

A situação de rota impossível apresentada nesta fase do trabalho, enfatizou a estratégia de distribuição de recursos, através da incorporação do VANT funcional à formação mais próxima, e a segurança dos agentes, através do retorno do VTNT à sua origem. Fica como sugestão para continuidade deste trabalho, a situação de replanejamento para o caso de remoção de um obstáculo. Esta situação irá possibilitar a retomada da navegação do VTNT e, conseqüentemente, a reconstrução da formação #2 com um dos VANTs da formação #1.

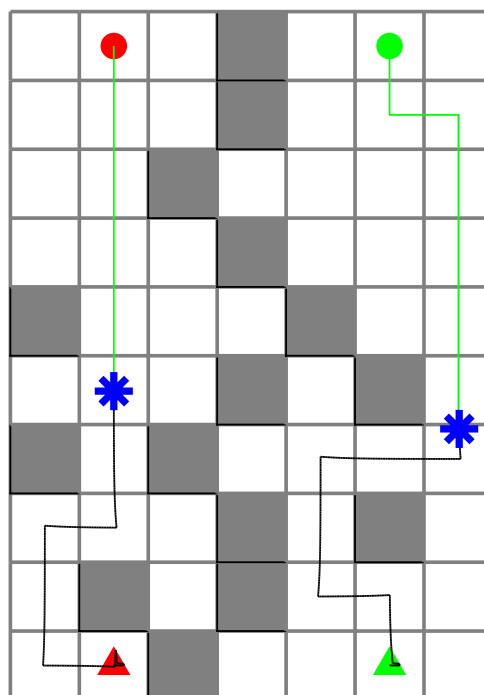
Por fim, assumindo a limitação do campo de visão dos VTNTs e condições diversas que venham comprometer o voo de longa duração dos VANTs, ainda como sugestão de trabalhos futuros, pode-se pensar na utilização vários VANTs em formação com um único VTNT. Neste caso, a visão indireta do VTNT é mantida pela troca de VANTs, seguindo as estratégias apresentadas nesta dissertação.

Figura 28 – Instantâneos da vista superior da simulação de validação do algoritmo obtido no Capítulo 2.

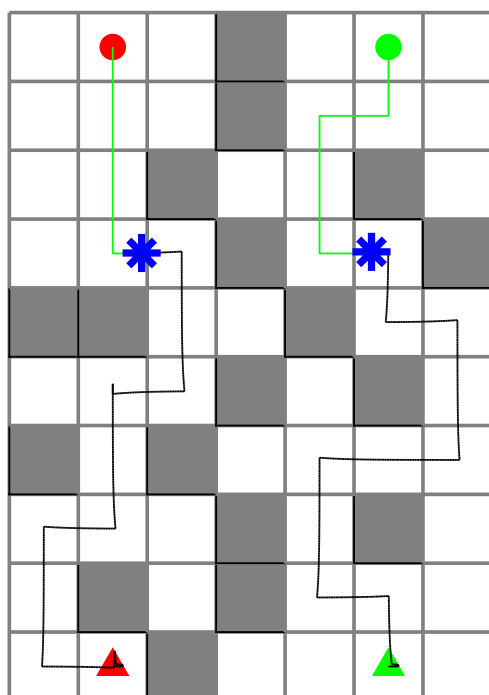
(a) Estado inicial.



(b) Estado intermediário.



(c) Estado intermediário.



(d) Estado final.

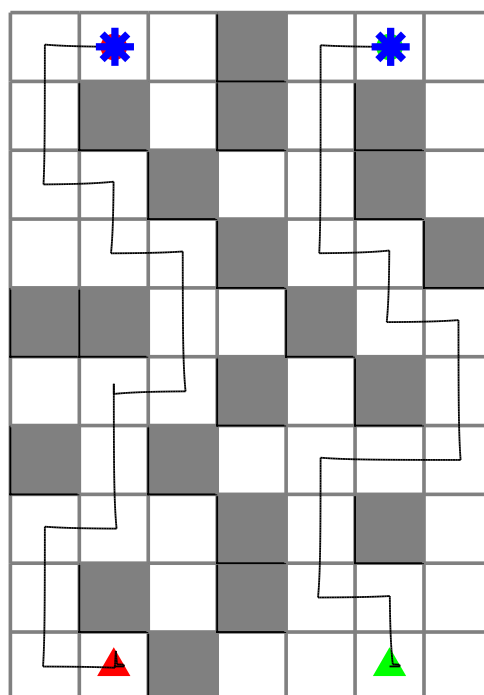
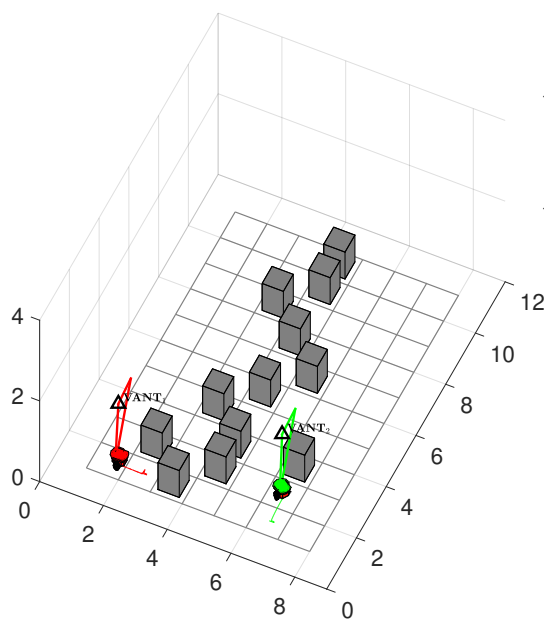
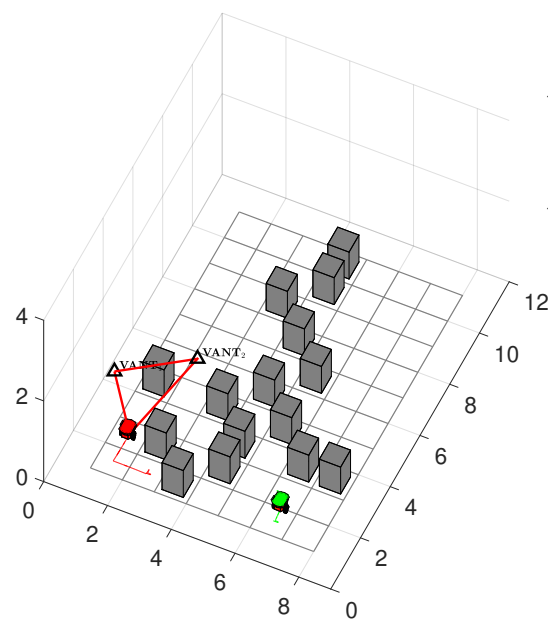


Figura 29 – Instantâneos da vista em perspectiva da simulação de validação da estratégia tolerante a falhas.

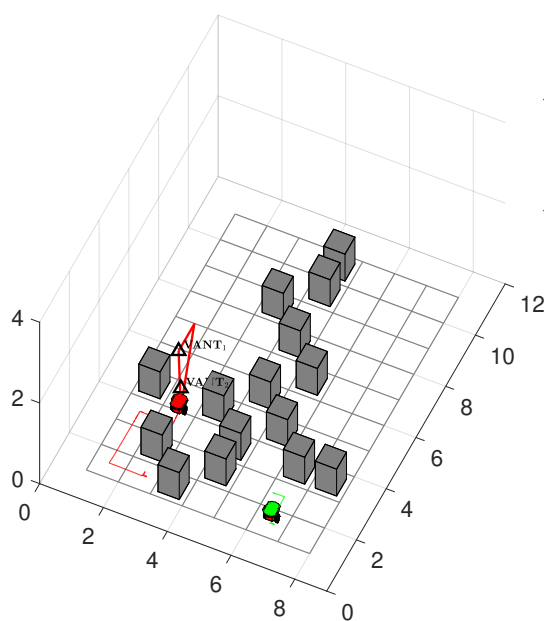
(a) Estado inicial.



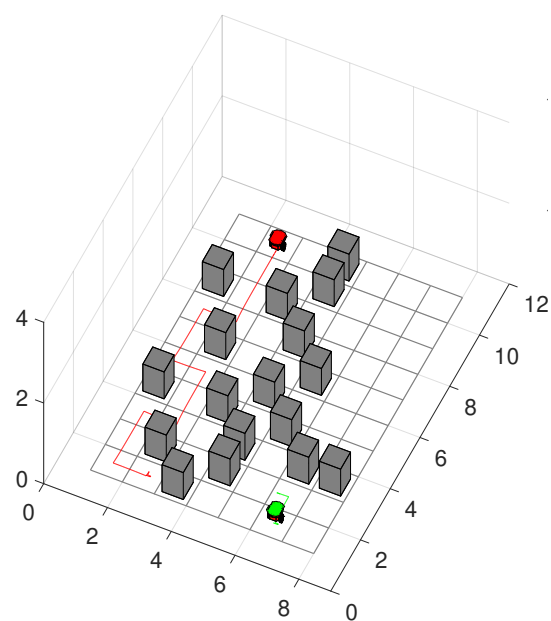
(b) Estado intermediário.



(c) Estado intermediário.



(d) Estado final.



Referências

- ABCI, B. et al. An informational approach for sensor and actuator fault diagnosis for autonomous mobile robots. *Journal of Intelligent and Robotic Systems: Theory and Applications*, v. 99, n. 2, p. 387–406, 2020. ISSN 15730409.
- ACKERMAN, E. *US Army considers replacing thousands of soldiers with robots*. 2014. IEEE Spectrum.
- AJEIL, F. H. et al. Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments. *Sensors (Switzerland)*, v. 20, n. 7, 2020. ISSN 14248220.
- ARBANAS, B. et al. Aerial-ground robotic system for autonomous delivery tasks. In: IEEE. *2016 IEEE international conference on robotics and automation (ICRA)*. [S.l.], 2016. p. 5463–5468.
- ARKIN, R. C. *Behaviour-Based Robotics*. [S.l.: s.n.], 1998.
- BACHETI, V. P.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Seguimento de Caminho Usando Espaço Nulo para uma Formação VANT-VTNT. In: *Anais do Congresso Brasileiro de Automática 2020*. [S.l.]: sbabra, 2020.
- BARBOSA, J. P. et al. Line formation using multi-layer control scheme. *Proceedings - 15th Latin American Robotics Symposium, 6th Brazilian Robotics Symposium and 9th Workshop on Robotics in Education, LARS/SBR/WRE 2018*, p. 45–50, 2018.
- BHONDVE, T. B.; SATYANARAYAN, R.; MUKHEDKAR, M. Mobile rescue robot for human body detection in rescue operation of disaster. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, v. 3, n. 6, p. 9876–9882, 2014.
- BRANDAO, A. S. et al. A multi-layer control scheme for a centralized uav formation. In: IEEE. *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.], 2014. p. 1181–1187.
- BRANDÃO, A. S. et al. The multilayer control scheme: A strategy to guide n -robots formations with obstacle avoidance. *Journal of Control, Automation and Electrical Systems*, Springer, v. 26, n. 3, p. 201–214, 2015.
- BRANDÃO, A. S.; SARCINELLI-FILHO, M. On the Guidance of Multiple UAV using a Centralized Formation Control Scheme and Delaunay Triangulation. *Journal of Intelligent and Robotic Systems: Theory and Applications*, v. 84, n. 1-4, p. 397–413, 2016. ISSN 15730409.
- CALDEIRA, A. G. et al. UAV Path-Following Strategy for Crossing Narrow Passages. *2020 International Conference on Unmanned Aircraft Systems, ICUAS 2020*, p. 26–31, 2020.

- CANTELLI, L. et al. Uav/ugv cooperation for surveying operations in humanitarian demining. In: IEEE. *2013 IEEE international symposium on safety, security, and rescue robotics (SSRR)*. [S.l.], 2013. p. 1–6.
- ÇAŞKA, S.; GAYRETLI, A. A survey of uav/ugv collaborative systems. *CIE44&IMSS*, v. 14, p. 453–463, 2014.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische mathematik*, Springer, v. 1, n. 1, p. 269–271, 1959.
- ERNANDES-NETO, V.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Manipulação de Carga por uma Formação Triangular de Quadrimotores. In: *Anais do Congresso Brasileiro de Automática 2020*. [S.l.]: sbabra, 2020.
- ERNANDES-NETO, V.; PACHECO, G. V.; BRANDÃO, A. S. On the Weighting of Control Signals in a Multi-robot System: A Formation-Based Analysis. *Journal of Control, Automation and Electrical Systems*, v. 31, n. 5, p. 1121–1131, 2020. ISSN 21953899.
- ERNANDES-NETO, V.; SARCINELLI-FILHO, M.; BRANDÃO, A. S. Trajectory-tracking of a heterogeneous formation using null space-based control. *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*, p. 187–195, 2019.
- GHOMMAM, J. et al. Formation path following control of unicycle-type mobile robots. *Robotics and Autonomous Systems*, v. 58, n. 5, p. 727–736, 2010. ISSN 09218890.
- GUÉRIN, F. et al. Uav-ugv cooperation for objects transportation in an industrial area. In: IEEE. *2015 IEEE International Conference on Industrial Technology (ICIT)*. [S.l.], 2015. p. 547–552.
- HAN, J.; PARK, H.; SEO, Y. Path planning for a mobile robot using ant colony optimization and the influence of critical obstacle. In: *Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management Detroit, Michigan, USA*. [S.l.: s.n.], 2016.
- HAN, X. et al. Autonomous mobile ground control point improves accuracy of agricultural remote sensing through collaboration with UAV. *Inventions*, v. 5, n. 1, 2020. ISSN 24115134.
- HERNANDEZ, J. A. et al. Formation control of cooperative robots with limited sensing using a virtual robot as reference. In: *2009 6th Latin American Robotics Symposium (LARS 2009)*. IEEE, 2009. p. 1–7. ISBN 978-1-4244-6256-8. Disponível em: <<http://ieeexplore.ieee.org/document/5418317/>>.
- HOLTE, R. C. et al. Searching with abstractions: A unifying framework and new high-performance algorithm. In: CITESEER. *Proceedings of the biennial conference-Canadian society for computational studies of intelligence*. [S.l.], 1994. p. 263–270.
- IANNACE, G.; CIABURRO, G.; TREMATERRA, A. Fault diagnosis for UAV blades using artificial neural network. *Robotics*, v. 8, n. 3, 2019. ISSN 22186581.
- KIM, P. et al. Uav-ugv cooperative 3d environmental mapping. In: *ASCE International Conference on Computing in Civil Engineering 2019 American Society of Civil Engineers*. [S.l.: s.n.], 2019.

- KOENIG, S.; LIKHACHEV, M.; FURCY, D. Lifelong planning a*. *Artificial Intelligence*, Elsevier, v. 155, n. 1-2, p. 93–146, 2004.
- LEE, G.; CHWA, D. Decentralized behavior-based formation control of multiple robots considering obstacle avoidance. *Intelligent Service Robotics*, v. 11, n. 1, p. 127–138, 2018. ISSN 18612784.
- LEE, J. D. et al. Development of mobile robot with vision inspection system and three-axis robot. *2018 3rd International Conference on Control and Robotics Engineering, ICCRE 2018*, p. 6–10, 2018.
- LEICA, P. et al. Dynamic obstacle avoidance based on time-variation of a potential field for robots formation. *2017 IEEE 2nd Ecuador Technical Chapters Meeting, ETCM 2017*, v. 2017-January, n. 978, p. 1–6, 2018.
- LEICA, P. et al. Null-Space-Based Controller for Heterogeneous Robot Formation in Congested Environments. *Proceedings - 2019 International Conference on Information Systems and Computer Science, INCISCOS 2019*, p. 238–243, 2019.
- LIKAJ, R. et al. Path finding for a mobile robot using fuzzy and genetic algorithms. *International Journal of Mechanical Engineering and Technology*, v. 8, n. 8, p. 659–669, 2017. ISSN 09766359.
- LIU, C.; JIANG, B.; ZHANG, K. Adaptive Fault-Tolerant H-Infinity Output Feedback Control for Lead-Wing Close Formation Flight. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 50, n. 8, p. 2804–2814, 2020. ISSN 21682232.
- LUAN, P. G.; THINH, N. T. Real-time hybrid navigation system-based path planning and obstacle avoidance for mobile robots. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 10, n. 10, p. 3355, 2020.
- MAINI, P.; SUJIT, P. On cooperation between a fuel constrained uav and a refueling ugv for large scale mapping applications. In: IEEE. *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.], 2015. p. 1370–1377.
- MAS, I.; PETROVIC, O.; KITTS, C. Cluster space specification and control of a 3-robot mobile system. In: IEEE. *2008 IEEE International Conference on Robotics and Automation*. [S.l.], 2008. p. 3763–3768.
- MOREIRA, M. S. M.; BRANDAO, A. S.; SARCINELLI-FILHO, M. Null space based formation control for a UAV landing on a UGV. *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*, p. 1389–1397, 2019.
- NAKHAEINIA, D. et al. A review of control architectures for autonomous navigation of mobile robots. *International Journal of Physical Sciences*, v. 6, n. 2, p. 169–174, 2011. ISSN 19921950.
- NARAYANAN, R. G. L.; IBE, O. C. A joint network for disaster recovery and search and rescue operations. *Computer Networks*, Elsevier, v. 56, n. 14, p. 3347–3373, 2012.
- NASRINAHAR, A.; CHUAH, J. H. Intelligent motion planning of a mobile robot with dynamic obstacle avoidance. *Journal on Vehicle Routing Algorithms*, Springer, v. 1, n. 2-4, p. 89–104, 2018.

- NGUYEN, N. P.; HONG, S. K. Fault diagnosis and fault-tolerant control scheme for quadcopter UAVs with a total loss of actuator. *Energies*, v. 12, n. 6, 2019. ISSN 19961073.
- NIKLANJBAR, A.; HAIDARI, M.; ATAI, A. A. Adaptive Sliding Mode Tracking Control of Mobile Robot in Dynamic Environment Using Artificial Potential Fields. *Journal of Computer & Robotics*, v. 11, n. 1, p. 1–14, 2018.
- OLIVEIRA, Á. V. de; FERNANDES, M. A. Dynamic planning navigation strategy for mobile terrestrial robots. *Robotica*, Cambridge University Press, v. 34, n. 3, p. 568–583, 2016.
- OLIVEIRA, G. C. R. de; CARVALHO, K. B. de; BRANDÃO, A. S. A hybrid path-planning strategy for mobile robots with limited sensor capabilities. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 19, n. 5, p. 1049, 2019.
- PACHECO, G. V. et al. Control signal weighting for robot formation. *Proceedings of the IEEE International Conference on Industrial Technology*, v. 2018-February, p. 111–116, 2018.
- PATLE, B. et al. Matrix-binary codes based genetic algorithm for path planning of mobile robot. *Computers & Electrical Engineering*, Elsevier, v. 67, p. 708–728, 2018.
- PATLE, B. et al. On firefly algorithm: optimization and application in mobile robot navigation. *World Journal of Engineering*, Emerald Publishing Limited, 2017.
- PHAN, C.; LIU, H. H. A cooperative uav/ugv platform for wildfire detection and fighting. In: IEEE. *2008 Asia Simulation Conference-7th International Conference on System Simulation and Scientific Computing*. [S.l.], 2008. p. 494–498.
- PIZETTA, I. H. B.; BRANDAO, A. S.; SARCINELLI-FILHO, M. A hardware-in-the-loop platform for rotary-wing unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, Springer, v. 84, n. 1-4, p. 725–743, 2016.
- PIZETTA, I. H. B.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Avoiding obstacles in cooperative load transportation. *ISA Transactions*, v. 91, p. 253–261, 2019. ISSN 00190578.
- POTENA, C. et al. Agricolmap: Aerial-ground collaborative 3d mapping for precision farming. *IEEE Robotics and Automation Letters*, IEEE, v. 4, n. 2, p. 1085–1092, 2019.
- RABELO, M. F. S.; BRANDAO, A. S.; SARCINELLI-FILHO, M. Landing a UAV on Static or Moving Platforms Using a Formation Controller. *IEEE Systems Journal*, p. 1–9, 2020. ISSN 1932-8184.
- RUBIO, F.; VALERO, F.; LLOPIS-ALBERT, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, v. 16, n. 2, p. 1–22, 2019. ISSN 17298814.
- SAPATY, P. Military robotics: latest trends and spatial grasp solutions. *International Journal of Advanced Research in Artificial Intelligence*, Citeseer, v. 4, n. 4, p. 9–18, 2015.
- SASKA, M.; KRAJNIK, T.; PFEUCIL, L. Cooperative μ uav-ugv autonomous indoor surveillance. In: IEEE. *International Multi-Conference on Systems, Signals & Devices*. [S.l.], 2012. p. 1–6.

- SHAMSAKHR, F.; SADEGHIBIGHAM, B. A neural network approach to navigation of a mobile robot and obstacle avoidance in dynamic and unknown environments. *Turkish Journal of Electrical Engineering and Computer Sciences*, v. 25, n. 3, p. 1629–1642, 2017. ISSN 13036203.
- SHEN, C. et al. Collaborative air-ground target searching in complex environments. In: IEEE. *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. [S.l.], 2017. p. 230–237.
- SHI, J. et al. Fault tolerant multivehicle formation control framework with applications in multiquadrotor systems. *Science China Information Sciences*, v. 61, n. 12, p. 3–5, 2018. ISSN 18691919.
- SHIMOSASA, Y. et al. Security service system using autonomous mobile robot. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, v. 4, n. 2, p. 825–829, 1999. ISSN 08843627.
- SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D. *Introduction to Autonomous Mobile Robots, Second Edition*. [S.l.: s.n.], 2011. ISSN 0263-5747. ISBN 026219502X.
- SINGH, N. H.; THONGAM, K. Mobile Robot Navigation Using Fuzzy Logic in Static Environments. *Procedia Computer Science*, v. 125, p. 11–17, 2018. ISSN 18770509.
- STAVROU, D. et al. Fault detection for service mobile robots using model-based method. *Autonomous Robots*, Springer US, v. 40, n. 2, p. 383–394, 2016. ISSN 15737527.
- TANNER, H. G. Switched uav-ugv cooperation scheme for target detection. In: IEEE. *Proceedings 2007 IEEE International Conference on Robotics and Automation*. [S.l.], 2007. p. 3457–3462.
- TUNA, G.; GUNGOR, V. C.; GULEZ, K. An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters. *Ad Hoc Networks*, Elsevier, v. 13, p. 54–68, 2014.
- VASCONCELOS, J. V. R.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Real-time path planning for strategic missions. *Applied Sciences (Switzerland)*, v. 10, n. 21, p. 1–13, 2020. ISSN 20763417.
- VASCONCELOS, J. V. R. et al. Agent Fault-Tolerant Strategy in a Heterogeneous Triangular Formation. *2020 International Conference on Unmanned Aircraft Systems, ICUAS 2020*, p. 1040–1047, 2020.
- VILLA, D. K. D.; BRANDAO, A. S.; SARCINELLI-FILHO, M. Path-following and attitude control of a payload using multiple quadrotors. *2019 19th International Conference on Advanced Robotics, ICAR 2019*, p. 535–540, 2019.
- VILLA, D. K. D.; BRANDAO, A. S.; SARCINELLI-FILHO, M. Navegação em ambientes externos com quadrimotores em formação utilizando controle por modo deslizante adaptativo. In: *Anais do Congresso Brasileiro de Automática 2020*. [S.l.]: sbabra, 2020.
- VILLA, D. K. D.; MELO-JUNIOR, F. M. S. R. de; BRANDAO, A. S. Seguimento de Trajetória, por Consenso, de uma Formação Descentralizada de Robôs. In: *XIII Simpósio Brasileiro de Automação Inteligente, SBAI 2017*. [S.l.: s.n.], 2017.

- YU, X.; LI, P.; ZHANG, Y. Fault-tolerant control design against actuator faults with application to UAV formation flight. *Chinese Control Conference, CCC*, p. 7167–7171, 2017. ISSN 21612927.
- ZHANG, H.-y.; LIN, W.-m.; CHEN, A.-x. Path planning for the mobile robot: A review. *Symmetry*, Multidisciplinary Digital Publishing Institute, v. 10, n. 10, p. 450, 2018.
- ZHANG, X. et al. Dynamic path planning algorithm for a mobile robot based on visible space and an improved genetic algorithm. *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 13, n. 3, p. 91, 2016.
- ZHANG, Y.; TIAN, Y.; LU, S. Design of a live maintenance mobile robot system for power substation equipment. *Proceedings - 2017 Chinese Automation Congress, CAC 2017*, v. 2017-January, p. 882–887, 2017.