

Fabricio Bortolini de Sá

**Cooperação entre um Robô Aéreo e um Robô
Terrestre para Identificação de Rotas Livres em
Solo**

Brasil

2014

Fabricio Bortolini de Sá

Cooperação entre um Robô Aéreo e um Robô Terrestre para Identificação de Rotas Livres em Solo

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Espírito Santo como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica - Automação.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Engenharia Elétrica

Orientadora: Profa. Dra. Raquel Frizera Vassallo

Coorientador: Prof. Dr. Anselmo Frizera Neto

Brasil

2014

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)

S111c Sá, Fabricio Bortolini de, 1985-
Cooperação entre um robô aéreo e um robô terrestre para
identificação de rotas livres em solo / Fabricio Bortolini de Sá. –
2014.
84 f. : il.

Orientador: Raquel Frizera Vassallo.
Coorientador: Anselmo Frizera Neto.
Dissertação (Mestrado em Engenharia Elétrica) –
Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Robôs móveis. 2. Veículos autônomos. 3. Aeronave não
tripulada – Controle automático. 4. Processamento de imagens.
5. Veículo aéreo não tripulado (VANT). I. Vassallo, Raquel
Frizera. II. Frizera Neto, Anselmo. III. Universidade Federal do
Espírito Santo. Centro Tecnológico. IV. Título.

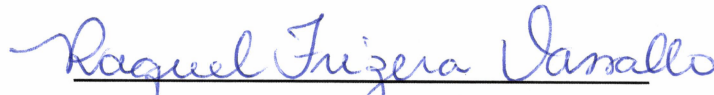
CDU: 621.3

Fabricio Bortolini de Sá

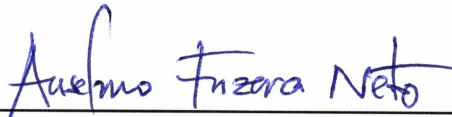
Cooperação entre um Robô Aéreo e um Robô Terrestre para Identificação de Rotas Livres em Solo

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Espírito Santo como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica - Automação.

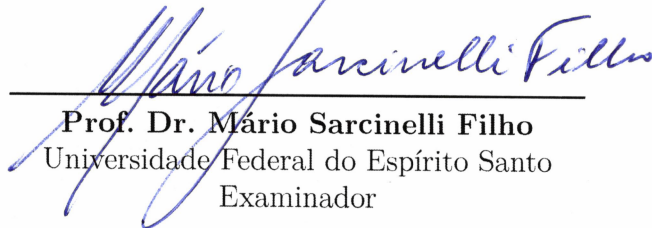
Trabalho aprovado. Brasil, 22 de dezembro de 2014:



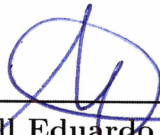
Profa. Dra. Raquel Frizera Vassallo
Universidade Federal do Espírito Santo
Orientadora



Prof. Dr. Anselmo Frizera Neto
Universidade Federal do Espírito Santo
Coorientador



Prof. Dr. Mário Sarcinelli Filho
Universidade Federal do Espírito Santo
Examinador



Prof. Dr. Maxwell Eduardo Monteiro
Instituto Federal do Espírito Santo
Examinador

Brasil
2014

A meus pais e a minha irmã.

AGRADECIMENTOS

Ao meu pai Paulo, pois além do bom exemplo foram os seus ensinamentos de mecânica que me ajudaram na construção do modelo utilizado nos experimentos.

À minha mãe Santina e minha irmã Juliana, pela educação, carinho e paciência. O apoio de vocês às minhas decisões foram fundamentais em todos os momentos.

À minha namorada Carolina, pelo amor e paciência. Foram muitos sábados e domingos de estudos e experimentos na Universidade, sempre recebendo seu apoio e por muitas vezes a sua companhia.

À minha tia Eliete, que sempre me incentivou seguir a carreira acadêmica e me tornar um Professor. Estou no caminho certo tia!

Ao meu coorientador Anselmo pelo apoio e pelos ensinamentos.

À minha orientadora Raquel, grande professora e exemplo que sigo desde o início da minha Graduação. Seus ensinamentos transcendem a Engenharia!

Ao meu amigo Flávio, por ter me acompanhado em muitos finais de semana na Universidade, me ensinando e ajudando nos experimentos.

Ao meu amigo Marino, que desde da iniciação científica sempre me ajudou nos trabalhos e estudos.

Aos meus amigos Fifi (Felippe) e Ricardinho (Ricardo) por me ajudarem na construção do quadrimotor utilizado neste trabalho, e por sempre participarem de todos os experimentos. Foram muitos testes, muitos desastres, e muito ânimo para sempre recomeçar! A nossa insistência está rendendo bons frutos!

A todos amigos do LAI: Mari (Mariana), Patrão (Vitor), Igor, Miltinho (Milton), Rafa (Rafael) e Robertinho (Roberto). As conversas, os debates, os cafés e os churrascos tornaram essa caminhada mais agradável.

Aos professores membros da comissão examinadora Dr. Mário Sarcinelli Filho e Dr. Maxwell Eduardo Monteiro, por aceitarem o convite e por todas as críticas e sugestões.

À FAPES pelo apoio financeiro através do projeto N° 54693209/2011 e ao CNPQ pela bolsa de Mestrado.

Muito obrigado!

*“I’m trying to free your mind, Neo.
But I can only show you the door.
You’re the one that has to walk through it.”*

Morpheus - The Matrix

RESUMO

Esta Dissertação de Mestrado aborda o problema de cooperação entre um robô aéreo e um terrestre em ambientes externos utilizando técnicas de controle não linear e de processamento de imagens. A ideia é implementar um sistema que permita ao robô terrestre navegar por um mapa previamente conhecido, sendo capaz de desviar-se de obstáculos e de recalcular a sua rota quando o robô aéreo, ou ele mesmo, identificar uma obstrução no caminho. Para isso, os robôs devem ser capazes de determinar por qual caminho navegarão, de detectar obstáculos e de trocar informações. A detecção de obstáculos e a identificação das *landmarks* pelo robô terrestre são realizadas utilizando, respectivamente, um sensor laser de varredura e processamento de imagens, enquanto o robô aéreo realiza essas mesmas atividades usando apenas o processamento de imagens. A vantagem que se obtém através da cooperação com um Veículo Aéreo Não Tripulado (VANT) é a otimização do tempo de navegação do robô terrestre ao se processar as imagens capturadas pelo robô aéreo. Esse processamento permitirá a detecção precoce de adversidades na rota do veículo terrestre e, também, das *landmarks* nesta mesma rota. Foram implementados controladores para navegação ponto a ponto e para desvio de obstáculos. Também se desenvolveu um método para realizar a correção da odometria do robô terrestre. Além disso, foi desenvolvido um simulador para o robô móvel e uma interface de comunicação entre os dois robôs. Por fim, foram realizados testes em ambientes externos a fim de verificar a eficácia do sistema desenvolvido. Através dos resultados alcançados pôde-se comprovar o funcionamento do sistema proposto. Os robôs foram capazes de navegar até o ponto destino e o robô terrestre foi capaz de desviar-se dos eventuais obstáculos, além de efetuar o recálculo do caminho quando uma obstrução foi encontrada na rota, tanto pelo robô aéreo quanto por ele próprio.

Palavras-chaves: Robôs Móveis, Controlador de Posição, Veículos aéreos não tripulados, Controle de aeronaves.

ABSTRACT

This Masters Thesis addresses the problem of cooperation between an aerial and a terrestrial robots in outdoor environments using nonlinear control and image processing techniques. The idea is to implement a system that allows the terrestrial robot to navigate in a map previously known, being able to avoid obstacles and to recalculate its route after the aerial robot, or the terrestrial robot, identifies an obstruction on the road. For this, the robots should be able to determine which path they will follow, to detect obstacles and to exchange information. The obstacle detection and the landmarks identification by the terrestrial robot are performed using, respectively, a laser sensor and image processing, while the aerial robot performs the same activities using only image processing. The advantage of performing a cooperation between a terrestrial robot and an Unmanned Aerial Vehicle (UAV) is the optimization of the terrestrial robot navigation time by processing the images captured by the aerial robot. This processing will allow an early detection of adversities on the route of the terrestrial vehicle and also the landmarks in this same route. Controllers for point to point navigation and for obstacle avoidance were implemented. A method to perform the correction of the terrestrial robot's odometry was also developed. In addition, a simulator for the mobile robot and also a communication interface between the two robots were developed. Finally, experimental tests were performed in outdoor environments in order to verify the effectiveness of the developed system. Through the results achieved with the tests, the effectiveness of the proposed system was comproved. The robots were able to navigate until reaching the destination point, the terrestrial robot was able to avoid the obstacles and also to recalculate its path when an obstruction on the route was found either by the aerial robot or by itself.

Key-words: Mobile Robots, Position Control, Unmanned Aerial Vehicles, Aircraft control.

LISTA DE ILUSTRAÇÕES

Figura 1 – Ilustração de robôs aéreo e terrestre navegando em cooperação.	20
Figura 2 – MobileSim.	22
Figura 3 – Interface	23
Figura 4 – O robô móvel <i>Pioneer 3-AT</i>	25
Figura 5 – Quadrimotor produzido pela 3D Robotics.	26
Figura 6 – Haste desenvolvida para o quadrimotor.	26
Figura 7 – Propulsor (a) e motor (b) utilizados na construção do Quadrimotor. . .	27
Figura 8 – ESC de 20A.	27
Figura 9 – Bateria principal (a) e regulador de tensão (b).	28
Figura 10 – Placa de distribuição de energia para as ESC's.	28
Figura 11 – <i>Link</i> de transmissão de odometria.	29
Figura 12 – Controle (a) e receptor (b).	29
Figura 13 – Conjunto transmissor/receptor (a) e bateria do transmissor (b).	29
Figura 14 – GPS (a) e ultrassom (b).	30
Figura 15 – Controlador de voo ArduPilot Mega 2.5.	30
Figura 16 – Foto do quadrimotor após as alterações.	31
Figura 17 – Plataforma de testes.	32
Figura 18 – Voo teste.	32
Figura 19 – O robô e um sistema de coordenadas fixo.	34
Figura 20 – Erro de posição em relação ao destino.	35
Figura 21 – Robô se aproximando de um obstáculo.	37
Figura 22 – Aproximação da <i>landmark</i>	38
Figura 23 – Dados capturados em uma varredura do sensor laser de varredura. . . .	39
Figura 24 – Vetor de distâncias do laser e vetor de diferenças, em milímetros. . . .	39
Figura 25 – Máquina de estados para detecção da <i>landmark</i>	40
Figura 26 – Correção da medição do sensor laser de varredura em relação ao centro do robô.	41
Figura 27 – Correção de posição utilizando as informações do sensor laser de varredura.	41

Figura 28 – Teste do algoritmo de correção de posição.	42
Figura 29 – Exemplo de mapa utilizado no trabalho.	43
Figura 30 – <i>Landmark</i> utilizada nos experimentos.	44
Figura 31 – Seções de um código de barras Code 128.	45
Figura 32 – Sub-regiões de um QR Code.	46
Figura 33 – Imagem capturada (a) e imagem segmentada (b).	47
Figura 34 – Interface.	49
Figura 35 – Visão geral do <i>software</i> desenvolvido.	49
Figura 36 – Fluxograma da função utilizada para carregar o mapa.	50
Figura 37 – Fluxograma da função utilizada para selecionar a origem e o destino dos robôs.	51
Figura 38 – Fluxograma da função utilizada para calcular as rotas dos robôs. . . .	52
Figura 39 – Fluxograma da função utilizada para recalculas rotas dos robôs. . . .	52
Figura 40 – Fluxograma da <i>thread</i> desenvolvida para a navegação do robô terrestre. .	54
Figura 41 – Robô terrestre se aproximando de um obstáculo intransponível.	55
Figura 42 – Fluxograma da <i>thread</i> desenvolvida para a câmera do robô terrestre. . .	57
Figura 43 – Fluxograma da <i>thread</i> desenvolvida para a navegação do robô aéreo. . .	58
Figura 44 – Fluxograma da <i>thread</i> desenvolvida para a câmera do robô aéreo. . . .	60
Figura 45 – Simulação do robô terrestre: identificação de obstrução (a), desvio de obstáculo (b) e destino alcançado (c).	61
Figura 46 – Mapa utilizado nos experimentos 1 e 2.	63
Figura 47 – Deslocamento das <i>landmarks</i> para os experimentos.	63
Figura 48 – Obstáculo utilizado no experimento 1.	64
Figura 49 – Caminho calculado no experimento 1.	64
Figura 50 – Caminho percorrido no experimento 1.	65
Figura 51 – Odometria do robô terrestre no experimento 1.	65
Figura 52 – Velocidades linear (a) e angular (b) do robô terrestre no experimento 1. .	66
Figura 53 – Caminho calculado no experimento 2.	67
Figura 54 – Obstáculo utilizado no experimento 2.	67
Figura 55 – Caminho percorrido no experimento 2.	68
Figura 56 – Odometria do robô terrestre no experimento 2.	68
Figura 57 – Velocidades linear (a) e angular (b) do robô terrestre no experimento 2. .	69
Figura 58 – Mapa utilizado nos experimentos 3 e 4.	70
Figura 59 – Obstáculo utilizado no experimento 3.	70
Figura 60 – Caminho calculado no experimento 3.	71
Figura 61 – Caminho recalculado para o quadrimotor no experimento 3.	72
Figura 62 – Caminho recalculado para o Pioneer no experimento 3.	72
Figura 63 – Caminho percorrido no experimento 3.	73
Figura 64 – Odometria de ambos robôs no experimento 3.	73

Figura 65 – Velocidades linear (a) e angular (b) do robô terrestre no experimento 3.	74
Figura 66 – Detecção do obstáculo pelo robô aéreo.	75
Figura 67 – Caminho recalculado para o Pioneer no experimento 4.	76
Figura 68 – Caminho percorrido no experimento 4.	76
Figura 69 – Odometria de ambos robôs no experimento 4.	77
Figura 70 – Velocidades linear (a) e angular (b) do robô terrestre no experimento 4.	78

LISTA DE ABREVIATURAS E SIGLAS

VANT	Veículo Aéreo Não Tripulado
VTNT	Veículo Terrestre Não Tripulado
GPS	<i>Global Positioning System</i>
DC	<i>Direct Current</i>
ESC	<i>Electronic Speed Controller</i>
LiPo	<i>Lithium-Polymer</i>
PVC	Policloreto de polivinila
ASCII	<i>American Standard Code for Information Interchange</i>
URL	<i>Uniform Resource Locator</i>
QR	<i>Quick Response</i>
EAN	<i>European Article Number</i>
UPC	<i>Universal Product Code</i>
OpenCV	<i>Open Source Computer Vision Library</i>
RGB	<i>Red, Green, Blue</i>
HSV	<i>Hue, Saturation, Value</i>
IDE	<i>Integrated Development Environment</i>
GUI	<i>Graphical User Interface</i>
ARIA	<i>Advanced Robotics Interface for Applications</i>

LISTA DE SÍMBOLOS

φ	Ângulo de orientação do robô
v	Velocidade linear do robô
ω	Velocidade angular do robô
ρ	Erro de posição do robô
θ	Orientação do robô em relação ao referencial inercial
α	Erro angular do robô
ϕ	Ângulo de desvio
β	Ângulo entre a orientação do robô e a parte do obstáculo que se encontra mais próxima deste
σ	Erro angular do robô, baseado nas medidas do laser
ξ	Erro de posição do robô, baseado nas medidas do laser
ρ_1	Distância do laser à <i>landmark</i>
ρ_2	Distância do centro do robô à <i>landmark</i>

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Cooperação entre Robôs Aéreos e Terrestres	18
1.2	Definição do Problema	20
1.3	Objetivos	20
1.3.1	Objetivo Geral	20
1.3.2	Objetivos Específicos	21
1.4	Metodologia	21
1.4.1	Parte Teórica	21
1.4.2	Parte de Simulação	22
1.4.3	Parte Experimental	22
1.5	Estrutura da Dissertação	23
2	ROBÔS MÓVEIS	24
2.1	Robô Terrestre	24
2.2	Robôs Aéreos	24
2.2.1	Quadrimotor utilizado	25
2.2.1.1	O <i>Frame</i>	26
2.2.1.2	Sensores e atuadores	26
3	MÉTODOS DESENVOLVIDOS	33
3.1	Controlador de Posição sem Orientação Final	33
3.1.1	O Modelo Cinemático do Robô	34
3.1.2	O Controlador de Posição	35
3.1.3	O Desvio de Obstáculos	36
3.1.4	Correção de Posição	37
3.2	Planejamento de Rota - O Algoritmo de Dijkstra	42
3.3	Processamento de Imagens	44
3.3.1	Detecção de <i>landmarks</i>	44
3.3.2	Detecção de obstáculos pelo Robô Aéreo	45
4	SOFTWARE DESENVOLVIDO	48
4.1	Carregar Mapa	50

4.2	Selecionar Origem e Destino	50
4.3	Calcular Caminho	51
4.4	Recalcular Caminho	52
4.5	Navegação Terrestre	53
4.6	Câmera Terrestre	56
4.7	Navegação Aérea	56
4.8	Câmera Aérea	58
4.9	Simulação	61
5	RESULTADOS	62
5.1	Experimento 1: Desvio de obstáculo	62
5.2	Experimento 2: Recálculo de rota	66
5.3	Experimento 3: Cooperação com o Quadrimotor no modo manual .	70
5.4	Experimento 4: Cooperação com o Quadrimotor no modo automático	74
6	CONSIDERAÇÕES FINAIS	79
6.1	Trabalhos Futuros	80
	Referências	81

CAPÍTULO 1

INTRODUÇÃO

Há mais de quinze anos o tema de cooperação de robôs para a realização de tarefas em conjunto tem sido abordado com grande interesse nas pesquisas de robótica. Os primeiros trabalhos envolviam poucos robôs e, à medida que os avanços tecnológicos e o barateamento de produtos eletrônicos permitiram, surgiram trabalhos envolvendo um maior número de agentes ou mesmo enxames de robôs (PARKER, 2000).

As principais vantagens de se trabalhar com grupos de robôs podem ser resumidas no aumento na capacidade sensorial, na diminuição do tempo para realização de tarefas e na robustez a falhas devido à redundância de agentes. Além disso, normalmente os robôs utilizados são robôs mais simples, pois há necessidade de plataformas menos complexas quando a tarefa é realizada em conjunto.

A maioria dos trabalhos utilizavam grupos de robôs terrestres para tarefas de mapeamento e exploração (BURGARD et al., 2005), vigilância e monitoramento (MARJOVI et al., 2009), transporte cooperativo de cargas (YAMASHITA et al., 2000; BAI; WEN, 2010) e deslocamento em formação (DESAI; OSTROWSKI; KUMAR, 2001; RAGHUWAIYA; SINGH; VANUALAILAI, 2011; RESENDE; CARELLI; SARCINELLI-FILHO, 2014).

Recentemente, diversos veículos aéreos autônomos e teleoperados têm sido desenvolvidos para aplicações em robótica. Existem aplicações nas quais a utilização de VANTs (Veículos Aéreos Não Tripulados) é vantajosa quando comparada com os VTNTs (Veículos Terrestres Não Tripulados). Os VTNTs possuem limitações para acessar determinados locais, principalmente devido à presença de obstáculos e às características do terreno. Dentre as atividades onde normalmente os VANTs apresentam vantagens quanto à sua aplicação podem-se citar mapeamento, detecção de incêndio em florestas, inspeção de tubulações de óleo e gás, rodovias, pontes e linhas de transmissão (EISENBEISS, 2004; DIOS; OLLERO, 2004; OLLERO; MERINO, 2004; RATHINAM et al., 2005).

Assim como aconteceu com os robôs terrestres, a fabricação e a comercialização de VANTs a baixo custo tem possibilitado a extensão dos trabalhos de robótica cooperativa para grupos de robôs aéreos. Devido às mesmas vantagens, como redução do tempo para realização de tarefas e redundância de agentes, já citadas para as atividades realizadas em cooperação, grupos de robôs aéreos começaram a ser utilizados em tarefas de mapeamento e exploração (SHIM et al., 2005; BRYSON; SUKKARIEH, 2007; ARTIEDA et al., 2009; HENG et al., 2011), controle de formação (TURPIN; MICHAEL; KUMAR, 2012), detecção e monitoramento de incêndios florestais (OLLERO et al., 2005) e até mesmo transporte de cargas (MELLINGER et al., 2010).

Devido às características e particularidades que os robôs terrestres e os robô aéreos possuem separadamente, a utilização de grupos híbridos, envolvendo os dois tipos de robôs para a execução de tarefas, permite explorar as vantagens que cada um apresenta. Uma breve discussão sobre as possibilidades de cooperação envolvendo esse grupos é feita na seção a seguir.

1.1 Cooperação entre Robôs Aéreos e Terrestres

A utilização de robôs aéreos em cooperação com robôs terrestres pode trazer uma série de vantagens, quando comparada às soluções que usam apenas um tipo deles. Uma dessas vantagens é que os robôs aéreos possuem um grande campo de visão, podendo capturar imagens com diferentes alcances visuais à medida que variam sua altura. Além disso, podem inspecionar mais rapidamente grandes áreas e visualizar regiões que normalmente não são alcançadas pelos robôs terrestres. Entretanto, não são capazes de localizar e realizar medidas com grande precisão, devido a limitações em suas imagens e altura. Por outro lado, os robôs terrestres podem se aproximar de alvos de interesse, obter informações mais detalhadas, além de interagir diretamente com objetos do ambiente.

Em (YU et al., 2001) robôs aéreos e terrestres são empregados para planejar trajetórias e rastrear alvos móveis cooperativamente. Já em (GARZÓN; DAVIDZAPATA; BARRIENTOS, 2013), um robô terrestre e um robô aéreo cooperam e compartilham dados sensoriais para explorar, mapear e navegar em um ambiente desconhecido. Outros exemplos de utilização de robôs aéreos e terrestres para mapeamento e exploração podem ser vistos em (HOW et al., 2009; KIM et al., 2010).

Existem ainda as aplicações de exploração e resgate como em (MICHAEL et al., 2012), em que pesquisadores usam um quadrimotor e um robô terrestre para explorar os três andares de um prédio após um desastre. A ideia é mostrar que, além de ser viável aplicar esse tipo de cooperação nas situações de desastre, quando se usam os dois robôs conjuntamente obtém-se vantagens que superam o uso de cada um separadamente nesse tipo de tarefa. Já em (MACARTHUR; MACARTHUR; CRANE, 2005), um robô aéreo é

utilizado em cooperação com um terrestre para detecção e desarme de minas terrestres.

Em (LONGO et al., 2007), um grupo de robôs é usado para ajudar na vigilância e nos momentos de crise em ambientes industriais e regiões vulcânicas, por exemplo, no caso de vazamento de substâncias tóxicas, explosões em indústrias petroquímicas e situações de riscos de explosões vulcânicas. Robôs aéreos são usados para uma primeira inspeção da área, para estimar a gravidade da situação e para coordenar o trabalho dos robôs terrestres. Estes, por sua vez, fazem o trabalho de aproximação, de medidas e de atuações no local atingido, além de carregarem robôs escaladores para serem liberados perto de prédios, para a inspeção de superfícies verticais.

Além de mapeamento, exploração e resgate, esse tipo de cooperação pode ser aplicado em atividades agrícolas e monitoramento de áreas florestais ou urbanas, usando robôs aéreos para orientar a movimentação e atuação de robôs terrestres.

Dentro desse contexto de cooperação entre robôs aéreos e terrestres, o objetivo deste trabalho consiste em aplicar técnicas de controle e visão computacional, baseadas nas informações de odometria, GPS e imagens capturadas por câmeras instaladas em um VANT e em um veículo terrestre, para navegação de robôs em um mapa previamente conhecido. A partir da realimentação visual obtida com a câmera a bordo de um robô aéreo, amplia-se a capacidade de navegação do robô terrestre, ao mesmo tempo que se mapeiam os obstáculos, unindo-se a informação da câmera com o sistema de posicionamento do robô em terra. A utilização da câmera acoplada ao VANT facilita a detecção de obstáculos presentes na rota de navegação do robô terrestre, o tipo do terreno a ser atravessado e a presença de obstruções ou não, sinalizando uma mudança de rota.

A Figura 1 ilustra um exemplo de navegação em cooperação entre um robô aéreo e um terrestre, assim como se pretende desenvolver nesse trabalho. A ideia é implementar um sistema que permita ao robô terrestre a navegação pelo mapa sendo capaz de desviar de obstáculos, e de recalcular a rota quando o robô aéreo, ou mesmo ele próprio, identificar uma obstrução no caminho.

Os autores de (ZHANG et al., 2009) desenvolveram um controlador multi-sensorial para um quadrimotor. Foram utilizados uma pequena câmera e um sensor inercial para manter o robô aéreo sobre uma referência terrestre. Dessa forma, um uso interessante para esse controlador seria na implementação de um sistema onde um robô aéreo deveria seguir um robô terrestre em movimento, enviando para o mesmo uma imagem aérea com informações do terreno ao redor do robô.

Esse crescente desenvolvimento de robôs aéreos e terrestres e a necessidade de se desenvolver técnicas de controle para permitir que eles cooperem na realização de tarefas é uma das principais motivações deste trabalho de Mestrado.

Além disso, uma outra grande motivação para o desenvolvimento deste trabalho é

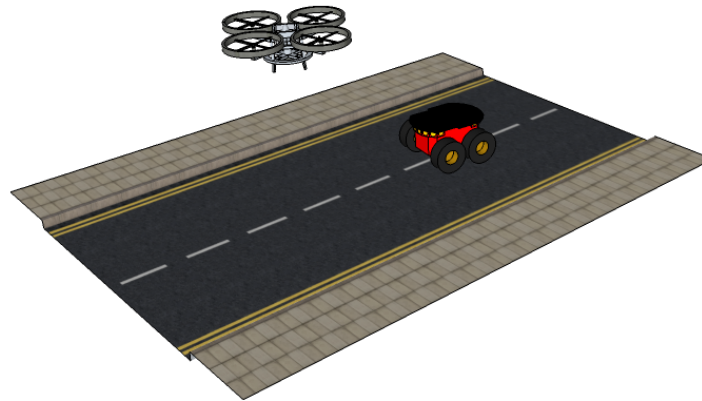


Figura 1 – Ilustração de robôs aéreo e terrestre navegando em cooperação.

a possibilidade de se utilizar robôs terrestres, aéreos ou a cooperação entre ambos para realização de tarefas de cunho social. Dentre essas tarefas podem-se citar o auxílio no atendimento de ambulâncias, auxílio ao Corpo de Bombeiros/Defesa Civil em situações de desastres naturais, entre outros.

1.2 Definição do Problema

O problema que se pretende resolver com este trabalho é a cooperação entre um robô terrestre e um robô aéreo de forma que eles troquem informações a respeito do ambiente onde estão inseridos e, assim, possam resolver a tarefa que lhes foi atribuída de uma maneira mais eficiente.

1.3 Objetivos

Os objetivos desta Dissertação de Mestrado podem ser divididos em objetivo geral e em objetivos específicos.

1.3.1 Objetivo Geral

Como objetivo geral, esta Dissertação de Mestrado visa implementar uma estratégia de cooperação entre um robô aéreo e um robô terrestre a fim de encontrar rotas livres para navegação. O robô aéreo deverá sobrevoar uma rota de navegação, identificar obstáculos que impossibilitem a navegação em solo, e enviar essas informações para o robô terrestre para que o mesmo recalcule a sua rota, quando necessário.

1.3.2 Objetivos Específicos

Para que o objetivo geral proposto seja alcançado, foram determinados os seguintes objetivos específicos:

- Habilitar um VANT para realizar um voo utilizando um GPS para seguir uma rota determinada.
- Desenvolver algoritmos de visão computacional, para ambos robôs, a fim de identificar marcas (*landmarks*) de localização no mapa de navegação.
- Desenvolver um controlador de posição com desvio de obstáculos para o robô terrestre.
- Realizar a correção de posição do robô terrestre utilizando informações de medição provenientes de um sensor laser de varredura.
- Implementar a cooperação entre o robô aéreo e o terrestre para navegação nas rotas calculadas.

1.4 Metodologia

A metodologia para se atingir cada um dos objetivos especificados pode ser resumida em três etapas (Teórica, Simulação e Experimental), indicando-se as atividades necessárias para completá-las.

1.4.1 Parte Teórica

Inicialmente foram realizados estudos sobre trabalhos envolvendo diversos quadrimotores. Em seguida, a cinemática e a dinâmica dos quadrimotores foram estudadas. Isso se fez necessário para um melhor entendimento do comportamento do quadrimotor durante uma sessão de voo.

Também foi levado em consideração nesta etapa o controlador necessário para comandar o robô terrestre. Tal controlador foi projetado utilizando métodos de análise de sistemas não-lineares, devido às características não lineares do modelo. Mais especificamente, foi utilizada a análise de Lyapunov para o projeto do controlador. Vale ressaltar que a experiência prévia de outros pesquisadores do laboratório, os quais têm trabalhado com helicópteros e robôs terrestres, foi de grande importância.

Uma vez feita a análise cinemática dos robôs e definido o controlador do veículo terrestre, foram estudados algoritmos para determinação de caminhos e detecção de rotas livres de obstáculos. O robô aéreo deverá ser capaz de navegar em um mapa previamente construído, sobrevoar os possíveis trajetos ou estradas, identificando (BRANDÃO, 2013;

BRANDÃO et al., 2014) as rotas livres para a passagem do robô terrestre (PEREIRA, 2006).

1.4.2 Parte de Simulação

Nesta parte do desenvolvimento do projeto foi utilizado o simulador MobileSIM, versão 0.7.3, da Adept MobileRobots, para realizar simulações envolvendo o robô móvel e seus sensores. O modelo do robô utilizado nas simulações foi um Pioneer 3DX. Uma tela deste simulador pode ser vista na Figura 2. O mapa foi gerado usando o programa Mapper3, versão 2.2.5, também da Adept MobileRobots.

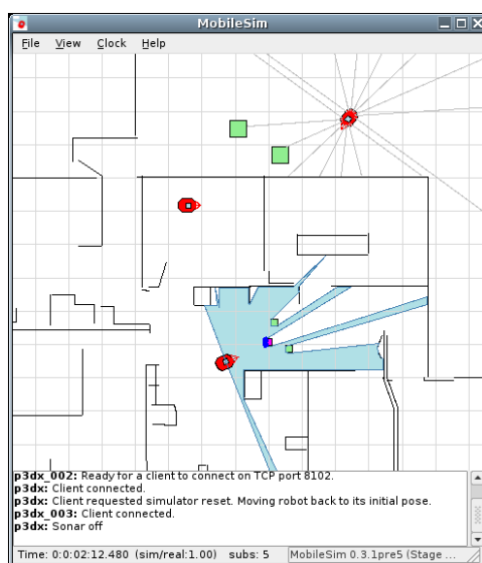


Figura 2 – MobileSim.

Também foi desenvolvida uma interface gráfica que permite que usuário carregue o mapa do mundo onde o robô se encontra, indique para o robô qual é o ponto onde ele se encontra no início da tarefa e selecione o ponto final do mapa onde se deseja que o robô chegue. Além disso, essa interface possui duas janelas onde as imagens capturadas pelas câmeras do robô e do quadrimotor são apresentadas, uma outra janela que é atualizada *online* e mostra a rota que o robô deve percorrer e a posição do robô terrestre e, por fim, uma região onde são escritas algumas mensagens para auxiliar o operador. A Figura 3 mostra uma tela da interface desenvolvida.

1.4.3 Parte Experimental

Após a realização do estudo teórico e de simulações, alguns experimentos foram realizados para comprovar a eficácia do método proposto. Os primeiros experimentos foram realizados apenas com o robô terrestre e, em seguida, com ambos robôs cooperando.

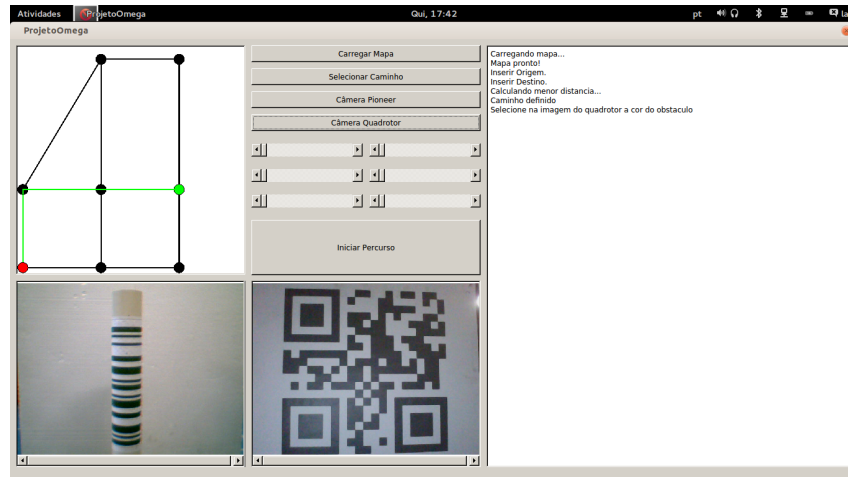


Figura 3 – Interface

Com esses experimentos percebeu-se que os métodos implementados são capazes de fazer com que tanto o robô terrestre quanto o robô aéreo realizem a tarefa dada a cada um deles. Em outras palavras, o robô terrestre foi capaz de navegar pelo espaço de trabalho e desviar dos eventuais obstáculos que surgiam em seu caminho, utilizando dados de um sensor laser de varredura, além de recalculá-la, caso necessário, e corrigir a sua posição todas as vezes que se aproximava de um ponto conhecido.

O robô aéreo teve como principal tarefa sobrevoar a região por onde o robô móvel terrestre deveria navegar e verificar se essa rota estava desobstruída. Caso ela não estivesse, ele deveria enviar essa informação para que a rota do robô terrestre fosse recalculada. O quadrimotor foi capaz de realizar essa tarefa.

1.5 Estrutura da Dissertação

Para tratar dos problemas de navegação e cooperação propostos neste capítulo, o restante desta Dissertação de Mestrado está organizada da seguinte forma:

- O Capítulo 2 apresenta os robôs utilizados nesse trabalho, detalhando os materiais e componentes utilizados na construção do quadrimotor.
- O Capítulo 3 traz o controlador que foi implementado para o robô terrestre, assim como sua prova de estabilidade, além dos métodos utilizados para realizar a correção de odometria do veículo terrestre, detecção das marcas por ambos robôs e cálculo de rotas.
- O Capítulo 4 apresenta o *software* desenvolvido para esta Dissertação.
- O Capítulo 5 apresenta os resultados e os comentários.
- O Capítulo 6 apresenta as considerações finais e as propostas para trabalhos futuros.

CAPÍTULO 2

ROBÔS MÓVEIS

Este capítulo apresenta os robôs móveis utilizados para realizar os experimentos de navegação e cooperação propostas. A Seção 2.1 discute as principais características do robô terrestre utilizado nos experimentos, destacando as suas principais configurações e dispositivos embarcados. Na Seção 2.2 são discutidas características dos VANT's utilizados em trabalhos atuais, e é apresentado o Quadrimotor desenvolvido neste trabalho.

2.1 Robô Terrestre

O robô terrestre utilizado na realização dos experimentos nesta Dissertação foi o *Pioneer 3-AT* (veja Figura 4), o qual é equipado com um sensor laser de varredura modelo SICK LMS200 e uma *WebCam* modelo VX3000 da Microsoft. Além disso, ele possui um *notebook* com processador Core i7-4500U, com quatro núcleos de 1,8GHz, 8GB de memória RAM e placa gráfica GeForce GT750M, utilizado para processar os dados provenientes dos sensores, tanto do robô aéreo quanto do terrestre, e enviar comandos de velocidade e rota para que os robôs naveguem pelo ambiente no qual estão inseridos.

O robô apresentado na Figura 4 possui um par diferencial para acionamento dos motores (cada motor aciona simultaneamente o par de rodas do seu respectivo lado) e um giroscópio embarcado. O sistema de processamento de baixo nível do robô utiliza o Filtro de Kalman para fundir as informações provenientes dos *encoders* das rodas com os ângulos do giroscópio, diminuindo assim o erro de odometria do veículo.

2.2 Robôs Aéreos

Entre os VANTs mais comumente utilizados encontram-se as aeronaves de asas fixas (aviões e planadores) e rotativas (helicópteros e veículos aéreos de quatro ou mais



Figura 4 – O robô móvel *Pioneer 3-AT*.

rotores). Quando comparados com os helicópteros, os veículos aéreos de quatro rotores (quadrimotores), são atualmente mais utilizados, por serem mais estáveis e apresentarem os mesmos benefícios obtidos pelos helicópteros. Devido à sua maior estabilidade, e principalmente a ausência de partes mecânicas móveis, o projeto dos controladores se torna menos complexo e mais versátil. Além disso, também apresentam a possibilidade de decolar, planar e aterrissar em ambientes de pequenas dimensões (TOURNIER et al., 2006), facilitando, dessa forma, experimentos em escala reduzida antes de se aplicar a estratégia em ambientes reais.

Levando-se em consideração as facilidades anteriormente apresentadas, neste projeto optou-se pelo uso de um quadrimotor para o qual foi desenvolvido o modelo apresentado na Subseção 2.2.1.

2.2.1 Quadrimotor utilizado

Para realização dos experimentos foi desenvolvido um quadrimotor baseado no modelo 3DR ArduCopter Quad-C produzido pela empresa 3D Robotics, que pode ser visto na Figura 5.

A construção do veículo aéreo pode ser dividida em dois passos, onde o primeiro deles consiste na confecção da sua estrutura física, o *frame*, e o segundo na adaptação dos sensores e atuadores.



Figura 5 – Quadrimotor produzido pela 3D Robotics.

2.2.1.1 O *Frame*

Existem diversos materiais que podem ser utilizados para a construção de aeromodelos, sejam eles de asas fixas ou rotativas, por exemplo a madeira balsa, o alumínio, o aço, o depron (um tipo de isopor), a fibra de carbono, a fibra de vidro, entre outros. Para a confecção da parte principal da estrutura deste quadrimotor foi escolhida a fibra de carbono, devido à sua excelente relação de resistência em função de seu peso. Foram utilizados tubos de fibra com 10mm de diâmetro externo e 8mm de diâmetro interno para compor os braços do veículo. A esses braços foram fixadas pequenas peças de náilon para permitir a fixação dos tubos na estrutura central, além de possibilitar a acomodação dos motores e do trem de pouso. Um dos braços feito de fibra de carbono com as peças de náilon fixadas pode ser visto na Figura 6.

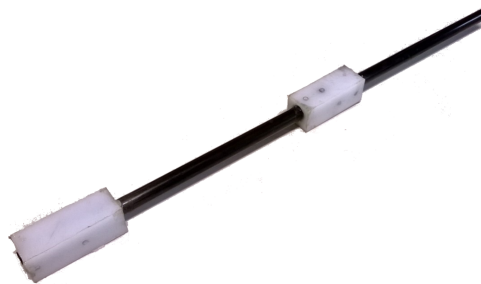


Figura 6 – Haste desenvolvida para o quadrimotor.

A haste desenvolvida possui um prolongamento de 20cm após a base de fixação do motor. O intuito desse prolongamento é de proteger a hélice do contato com o solo, o que comumente acontece durante o procedimento de aterrissagem, principalmente quando o voo é realizado em ambientes com ventos fortes. O massa total de cada haste é de 45g.

2.2.1.2 Sensores e atuadores

Os atuadores utilizados para acionar os propulsores 11x4,7 (11 polegadas de comprimento com um passo 4,7 polegadas) são motores DC Brushless de 880Kv (880 rpm

por volt). Esse tipo de motor tem ímãs permanentes fixados no rotor, o que dispensa a utilização de escovas, melhorando a sua eficiência energética e diminuindo o ruído operacional, quando comparados aos motores DC comuns. As Figuras 7 (a) e 7 (b) mostram, respectivamente, o propulsor e o motor utilizados no projeto.



Figura 7 – Propulsor (a) e motor (b) utilizados na construção do Quadrimotor.

Para acionar esse tipo de motor e controlar a sua velocidade de acordo com as necessidades do controlador de voo, foram utilizados ESCs (*Electronic Speed Controller*) de 20A. Esse dispositivo microcontrolado tem a função de criar um campo magnético girante (similar ao produzido em um motor trifásico de indução) e controlar a sua frequência de rotação, permitindo assim o controle de velocidade do motor. A Figura 8 apresenta o ESC utilizado.



Figura 8 – ESC de 20A.

Todos os circuitos são alimentados por uma bateria de tipo LiPo (*Lithium-Polymer*) de três células (ver Figura 9 (a)), exceto o link de vídeo, que é alimentado por uma bateria extra para evitar ruídos. Essa bateria tem uma tensão nominal de 11.1V, capacidade nominal de 5200mAh e capacidade de descarga de 30C (a bateria permite uma corrente de descarga 30 vezes maior que a sua corrente nominal), proporcionando ao modelo desenvolvido uma autonomia de voo de aproximadamente 15 minutos. Para regular a tensão da bateria foi utilizado o módulo de alimentação da Figura 9 (b). A finalidade desse módulo é fornecer uma tensão regulada de 5V para o controlador de voo, os sensores e os *links* de rádio e de odometria.



Figura 9 – Bateria principal (a) e regulador de tensão (b).

A Figura 10 mostra o circuito de distribuição de energia utilizado para alimentar os ESCs. O módulo realiza apenas a conexão física da bateria com os ESCs, não existindo nele nenhum regulador de tensão.

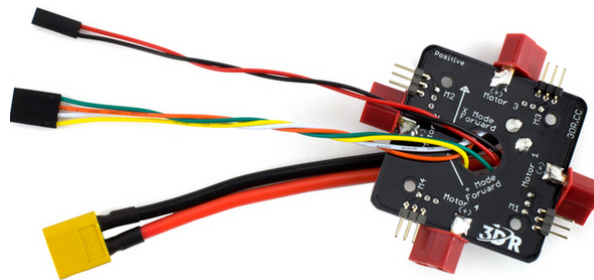


Figura 10 – Placa de distribuição de energia para as ESC's.

Foi utilizado um *link* de odometria de 915MHz para estabelecer uma conexão do quadrimotor com um computador. Essa conexão permite o acompanhamento da leitura dos sensores durante os voos, além de possibilitar a alteração de dados do piloto automático, como a gravação dos *WayPoints* que o veículo deve seguir. A Figura 11 mostra o conjunto utilizado, que consiste na antena que fica embarcada no quadrimotor mais a antena que fica conectada a uma porta usb do *notebook*.

O controle utilizado é o modelo 9XR da Turnigy, com 8 canais ppm, e um transmissor com alcance de 1km. O receptor instalado no quadrimotor é o modelo D4R-II da FrSky com um alcance de 1,5km. A Figura 12 (a) mostra o controle e a Figura 12 (b) o receptor instalado no veículo, utilizados para controlar o veículo quando no modo manual.

A câmera utilizada para capturar as imagens aéreas foi a GoPro HERO3+ Silver Edition, conforme Figura 13 (a). Essa câmera tem um peso de 136g, incluindo sua cápsula de proteção, e permite gravação de vídeos de até 1080p de resolução. Para transmitir as



Figura 11 – *Link* de transmissão de odometria.



(a)



(b)

Figura 12 – Controle (a) e receptor (b).

imagens capturadas no quadrimotor para o computador do robô terrestre foi utilizado um conjunto emissor/receptor da 3D Robotics, conforme a Figura 13 (b). O transmissor instalado no veículo foi alimentado com uma bateria separada para diminuir os ruídos causados pelos motores. A bateria utilizada (ver Figura 13 (c)) tem capacidade nominal de 900mAh e capacidade de descarga de 25C.



(a)

(b)

(c)

Figura 13 – Conjunto transmissor/receptor (a) e bateria do transmissor (b).

A Figura 14 (a) mostra o GPS utilizado, um módulo LEA-6H com antena de

cerâmica, taxa de atualização de até 5Hz e precisão de 2,5m. Para melhorar a medição de altura do quadrimotor, quando este está próximo ao solo, foi utilizado o sensor ultrassônico da Figura 14 (b), modelo MB1240 da série XL-MaxSonar-EZ, que mede distâncias entre 20cm e 765cm, com precisão de 1cm e taxa de atualização de até 10Hz.

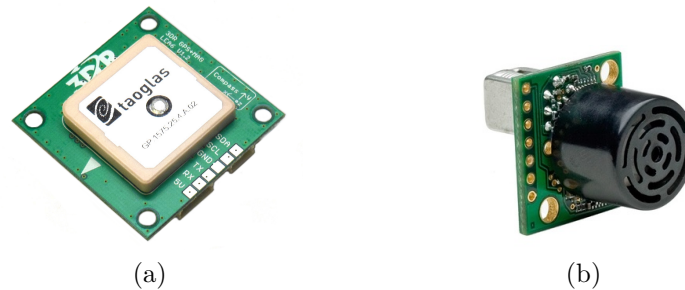


Figura 14 – GPS (a) e ultrassom (b).

Para fazer a integração de todos os sensores e realizar o controle de voo, tanto manual quanto automático, foi utilizado o controlador de voo ArduPilot Mega 2.5 (APM 2.5), mostrado na Figura 15. Esse dispositivo possui internamente bússola, giroscópio de 3 eixos, acelerômetro e magnetômetro, além de integrar às suas medições os outros sensores instalados anteriormente (Gps e Ultrassom). O APM 2.5 é *open source*, ou seja, é disponibilizado todo o código de programação, permitindo ao usuário que sejam feitas modificações no seu *firmware* (por exemplo, a implementação de outro controlador).

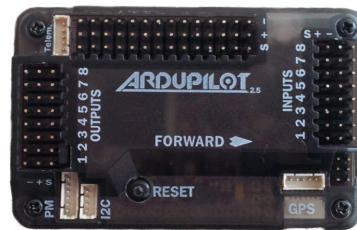


Figura 15 – Controlador de voo ArduPilot Mega 2.5.

A calibração dos sensores foi realizada utilizando o programa Mission Planer. As principais funcionalidades desse *software* são:

- Aquisição e visualização dos dados de voo.
- Planejamento de rotas utilizando pontos de GPS.
- Simulação de voos, utilizando dados armazenados.
- Calibração dos sensores.
- Calibração do controle remoto.

- Alteração dos ganhos do controlador de voo.
- Realização de testes dos sensores e atuadores.

A Figura 16 apresenta o quadrimotor desenvolvido, já com todos os sensores e atuadores embarcados. As esferas fixadas nas extremidades são utilizadas para identificar a orientação do veículo, as vermelhas indicam a frente e as verdes a traseira. O peso total do veículo é de 1890 gramas.



Figura 16 – Foto do quadrimotor após as alterações.

Antes de realizar o primeiro voo externo com o quadrimotor, foi utilizada a plataforma desenvolvida em (SANTANA, 2011) para verificar a estabilidade e os ganhos do seu controlador. A Figura 17 mostra o veículo fixado à plataforma.

Após a confirmação da estabilidade na plataforma, foi realizado um teste de voo com o quadrimotor em um ambiente externo. A Figura 18 mostra a odometria do veículo durante esse experimento. Nesse teste, o quadrimotor iniciou o voo no modo manual, controlado remotamente por um ser humano, e ao atingir a altitude de 1m foi enviado um comando para que o mesmo realizasse o voo autônomo, programado em seu piloto automático. Ao atingir o seu destino, um ponto localizado a aproximadamente 8m de sua posição inicial, foi acionado o controle remoto para realizar manualmente a sua aterrissagem.



Figura 17 – Plataforma de testes.

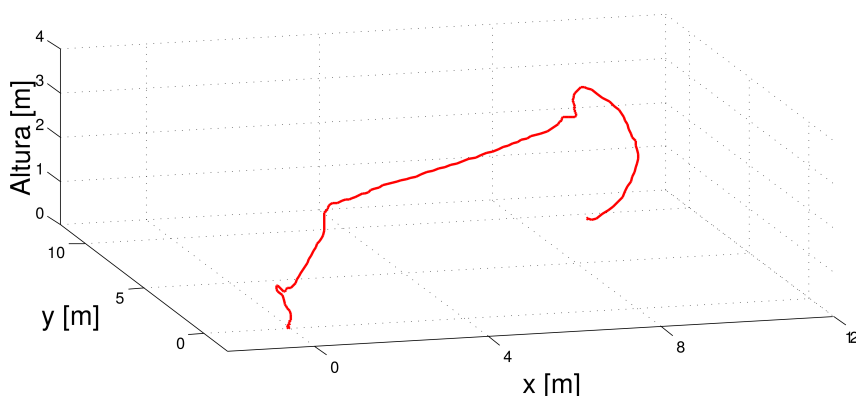


Figura 18 – Voo teste.

CAPÍTULO 3

MÉTODOS DESENVOLVIDOS

Com o intuito de automatizar os robôs utilizados e possibilitar a cooperação entre os mesmos, alguns algoritmos e um controlador foram implementados neste trabalho:

- Um controlador de posição final com desvio de obstáculos para o robô terrestre.
- O algoritmo de Dijkstra para determinar o menor caminho entre a posição atual do robô e o destino.
- Um algoritmo que verifica se o obstáculo que está à frente do robô é intransponível ou não.
- Um algoritmo para a detecção de *landmarks* utilizando visão computacional.
- Um algoritmo para a detecção de obstáculos terrestres pelo quadrimotor utilizando visão computacional.
- Um algoritmo que utiliza as medidas obtidas por um sensor laser de varredura para corrigir a posição do robô dada por seus *encoders*.

Assim, neste capítulo serão apresentados os algoritmos e controlador desenvolvidos nesse trabalho, além de algumas simulações para validar as soluções encontradas.

3.1 Controlador de Posição sem Orientação Final

Será discutido nesta seção o modelo cinemático do robô terrestre utilizado e o controlador responsável por gerar as ações de controle necessárias para que o robô móvel alcance o seu objetivo final. Serão apresentados também os métodos implementados para permitir o desvio de obstáculos pelo robô terrestre, assim como a abordagem para a correção de posição.

3.1.1 O Modelo Cinemático do Robô

O modelo cinemático do robô fornece as equações matemáticas que representam o seu movimento. O robô móvel utilizado neste trabalho, o *Pioneer 3-AT* (P3-AT), é de tração diferencial, e seu movimento é similar ao de robôs monociclo.

Considere-se o robô em seu espaço de trabalho, conforme a Figura 19.

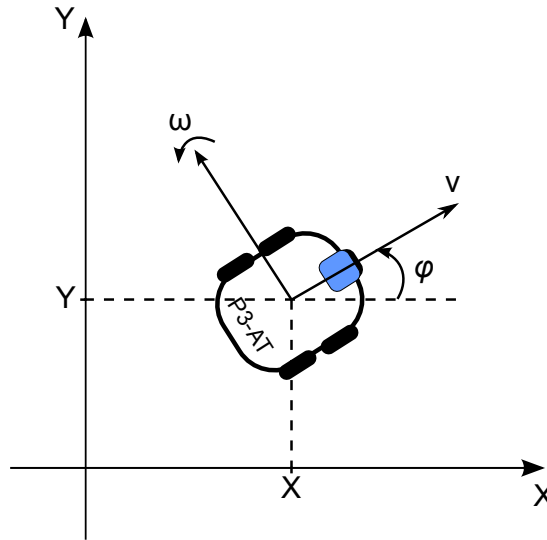


Figura 19 – O robô e um sistema de coordenadas fixo.

Sejam x e y as coordenadas do centro de gravidade do robô, φ a sua orientação, todos em relação ao referencial inercial, que corresponde ao ponto onde o robô foi inicialmente ligado, v a velocidade linear e ω a velocidade angular do robô. O modelo cinemático deste sistema é apresentado na Equação 3.1.

$$\begin{cases} \dot{x} = v \cos(\varphi) \\ \dot{y} = v \sin(\varphi) \\ \dot{\varphi} = \omega. \end{cases} \quad (3.1)$$

O modelo cinemático apresentado na Equação 3.1 está em coordenadas cartesianas. No entanto, para se desenvolver um controlador de posição convém representar o movimento do robô em coordenadas polares (SECCHI, 1998).

Considere o robô navegando com o objetivo de alcançar um destino. Sejam ρ e θ o erro de posição do veículo e a orientação do mesmo em relação ao referencial inercial situado no ponto destino, conforme pode ser visto na Figura 20.

A utilização do erro de posição ρ e da orientação do robô em relação ao referencial inercial θ permite que se escreva o modelo cinemático em coordenadas polares (Equação 3.2).

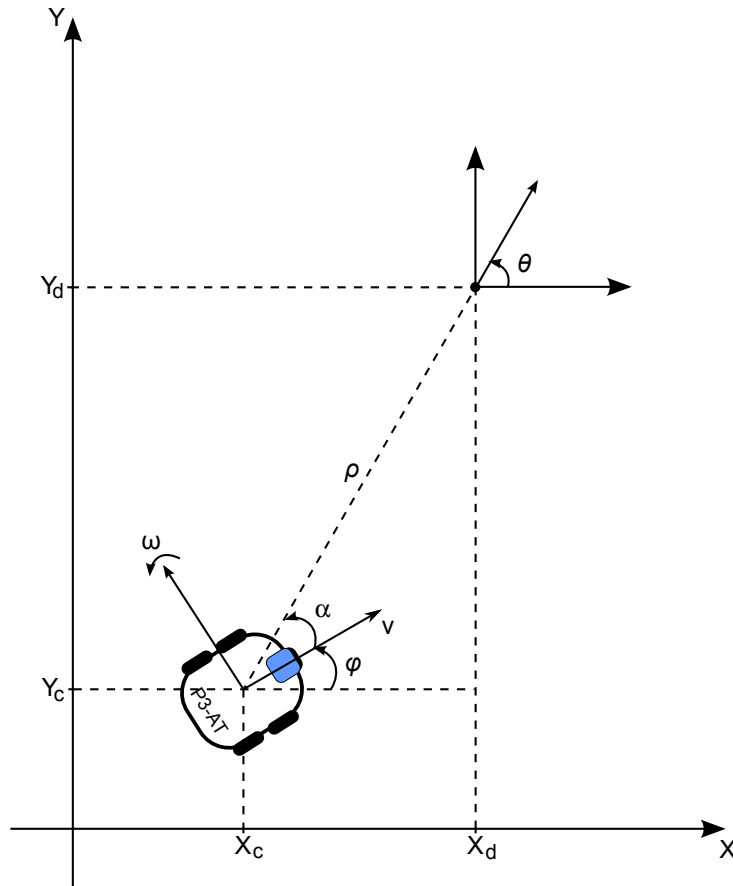


Figura 20 – Erro de posição em relação ao destino.

$$\begin{cases} \dot{\rho} = -v \cos(\theta - \varphi) \\ \dot{\theta} = v \frac{\text{sen}(\theta - \varphi)}{\rho} \\ \dot{\varphi} = \omega. \end{cases} \quad (3.2)$$

Pode-se observar na Figura 20 que $\theta = \alpha + \varphi$, ou seja $\alpha = \theta - \varphi$ é o ângulo entre o eixo de movimento do robô e o vetor do erro de posição ρ . Assim, a Equação 3.2 pode ser reescrita como

$$\begin{cases} \dot{\rho} = -v \cos(\alpha) \\ \dot{\theta} = v \frac{\text{sen}(\alpha)}{\rho} \\ \dot{\alpha} = -\omega + v \frac{\text{sen}(\alpha)}{\rho}. \end{cases} \quad (3.3)$$

3.1.2 O Controlador de Posição

Este controlador tem por objetivo permitir que o robô navegue de uma posição inicial até um ponto pré-definido.

O controlador recebe os dados referentes à odometria e à orientação do robô e estima os valores das velocidades linear v e angular ω que o robô deve executar para alcançar o ponto destino, fazendo com que os erros ρ e α tendam para zero.

Para que o objetivo de controle seja alcançado são propostas as seguintes ações de controle (SECCHI, 1998):

$$\begin{cases} v = K_v \tanh(\rho) \cos(\alpha) \\ \omega = K_\omega + K_v \frac{\tanh(\rho)}{\rho} \sin(\alpha) \cos(\alpha), \end{cases} \quad (3.4)$$

onde K_v e K_ω são constantes positivas e representam, respectivamente, o máximo valor da velocidade linear e um parâmetro de ajuste para o máximo valor da velocidade angular. Este controlador é capaz de fazer com que o robô atinja o ponto desejado, e tem sua prova de estabilidade apresentada em (SECCHI, 1998).

Neste trabalho será utilizado um mapa topológico construído por nós interligados por caminhos. O algoritmo de cálculo de rota informará ao robô quais são os nós por onde ele deve passar para atingir o objetivo final. Desse modo, o controlador proposto será utilizado entre nós consecutivos do caminho gerado, ou seja, serão passados ao controlador somente o nó atual onde se encontra o veículo e qual o próximo nó do seu caminho, até que o destino final seja alcançado.

No entanto, deseja-se que, além de alcançar o ponto destino, o robô seja capaz de detectar e desviar dos eventuais obstáculos que possam aparecer em seu caminho.

3.1.3 O Desvio de Obstáculos

Enquanto um robô navega por um ambiente ele pode, eventualmente, deparar-se com algum obstáculo. Com o objetivo de evitar que o robô colida com esses obstáculos, o robô móvel utiliza dados fornecidos pelo sensor laser de varredura acoplado a ele. Esses dados são processados e, caso seja confirmada a presença de um obstáculo, o destino do robô é alterado temporariamente até que o caminho seja desobstruído.

Para que o robô possa desviar dos obstáculos por ele detectados, o controlador gera um ângulo de desvio ϕ para que o alvo real seja rotacionado e, desta forma, possa ser criado um alvo virtual. Desse modo, enquanto o obstáculo não for ultrapassado, o robô navega em direção ao alvo virtual. Este ângulo é calculado em função da posição angular do obstáculo em relação ao robô. Assim, o ângulo de giro ϕ é determinado de tal forma que o alvo virtual é posicionado para que o robô possa desviar do objeto descrevendo uma trajetória tangente à sua borda.

Na Figura 21 é apresentada uma situação em que o robô encontra um obstáculo e o respectivo ângulo ϕ é gerado, para que haja um desvio tangencial (FERREIRA et al., 2008; PEREIRA, 2006).

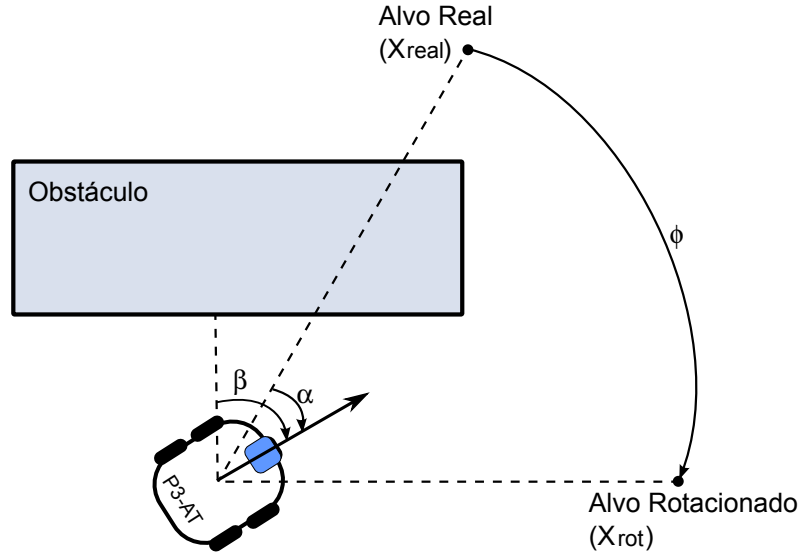


Figura 21 – Robô se aproximando de um obstáculo.

Pode-se observar na Figura 21 a existência de um ângulo β , que é o ângulo entre a orientação do robô e a parte do obstáculo que se encontra mais próxima deste. Este ângulo é obtido diretamente do sensor laser de varredura. De posse dos ângulos α e β , calcula-se o ângulo de rotação ϕ como

$$\phi = \text{sign}(\beta)(|\beta| - 90) - \alpha. \quad (3.5)$$

Após a determinação do ângulo de rotação ϕ , cria-se o alvo virtual, X_{rot} , da Figura 21. Este alvo consiste na rotação do alvo real que o robô deve atingir (X_{real} da Figura 21). As coordenadas do alvo virtual são obtidas pela transformação

$$X_{\text{rot}} = \begin{bmatrix} \cos(\phi) & \text{sen}(\phi) \\ -\text{sen}(\phi) & \cos(\phi) \end{bmatrix} X_{\text{real}}. \quad (3.6)$$

Toda vez que um obstáculo for encontrado, um alvo virtual é gerado e passa a ser o destino do robô enquanto o obstáculo não for transposto. Dessa maneira, o controlador para desvio de obstáculos é o mesmo controlador de Posição Final com o alvo modificado. Assim como o Controlador de Posição Final é estável, o controlador para desvio de obstáculos também o é.

3.1.4 Correção de Posição

A odometria de um robô móvel possui uma boa precisão para pequenas distâncias. Porém, como o cálculo de posição é baseado nessas informações e realizado de maneira incremental, o acúmulo de erro com o tempo de navegação (principalmente o erro de orientação) inviabiliza sua utilização para um longo caminho.

Para minimizar o erro angular obtido utilizando apenas a odometria, o robô P3-AT utiliza o Filtro de Kalman a fim de integrar essas medidas às informações angulares proveni-

entes de um giroscópio. Apesar disso, foi observado durante os primeiros experimentos que o erro acumulado durante a sua navegação impossibilitava o alcance do seu objetivo. Por exemplo, em um teste onde a distância total percorrida pelo robô foi de aproximadamente 20m e foram realizadas duas rotações de 90° , os erros angular e linear já ultrapassavam os valores de 20° e 2m, respectivamente.

Dessa maneira, para diminuir o erro acumulado foi utilizado o sensor laser embarcado para corrigir a posição do robô. Considerando que o mapa por onde se deve navegar é conhecido, ou seja, sabem-se todas as coordenadas das *landmarks*, sempre que o robô se aproxima de uma dessas marcas é possível utilizar as medidas relativas de distância do robô a esse ponto e corrigir a sua posição atual. A Figura 22 mostra uma situação onde o veículo se aproxima de uma *landmark*.

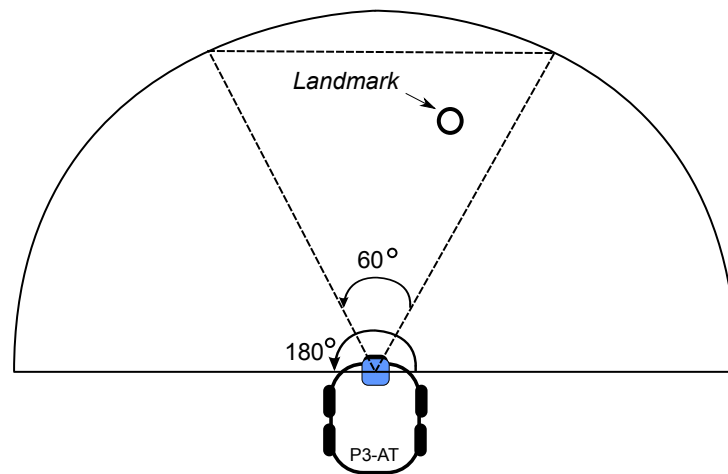


Figura 22 – Aproximação da *landmark*.

Na Figura 22 são destacados dois ângulos. O primeiro deles, o de 60° , representa o ângulo de abertura da câmera utilizada nesse trabalho, fixada acima do sensor laser de varredura e alinhada com o mesmo, enquanto o ângulo de 180° é a faixa de medição do sensor laser de varredura. Para diminuir a região de busca das *landmarks* nas leituras obtidas pelo sensor laser de varredura, é utilizada a informação da posição da marca na imagem. Quando identificada a *landmark*, o algoritmo de correção de posição inicia a busca da mesma no vetor de medição do sensor laser de varredura, considerando apenas 15° à esquerda e 15° à direita do ângulo da marca encontrada na imagem. Esses ângulos foram escolhidos considerando-se que a uma distância de 3m (normalmente essa é a distância que a câmera consegue identificar o código de barras de uma *landmark*) a região de procura da *landmark* é de aproximadamente 1,6m, dando uma boa margem de tolerância para medições imprecisas nas imagens, e eliminando outros objetos que possam atrapalhar a identificação da marca pelo sensor laser de varredura.

A Figura 23 mostra uma varredura do sensor laser de varredura. Nessa imagem é destacada, em vermelho, a *landmark* colocada próximo ao robô.

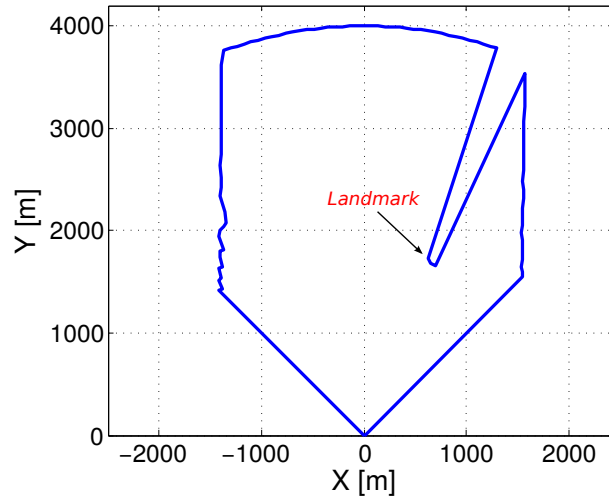


Figura 23 – Dados capturados em uma varredura do sensor laser de varredura.

É possível observar na Figura 23 que, quando a *landmark* é colocada em um ambiente sem muitos obstáculos, a sua identificação nas medições do sensor laser de varredura pode ser feita procurando-se no vetor de medidas uma transição brusca de valores, seguida por um pequeno intervalo de valores similares e uma nova transição brusca. Desta forma, para encontrar a marca na região do vetor de medidas do sensor laser de varredura referente à posição da imagem onde a mesma foi identificada, é calculado um vetor de diferenças, conforme a Figura 24. Esse vetor é construído pela subtração da medida $i+1$ do vetor do sensor laser de varredura pela medida i do mesmo.

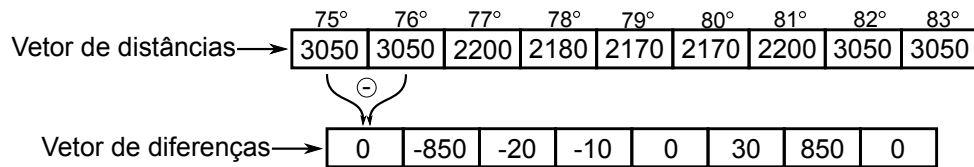


Figura 24 – Vetor de distâncias do laser e vetor de diferenças, em milímetros.

Dessa forma, utilizando este vetor de diferenças, foi projetada a máquina de estados da Figura 25 para identificar a *landmark* a partir das medidas do sensor laser de varredura. Foi utilizada uma Máquina de Estados Mealy, onde as variáveis T (T1 a T7) representam as saídas e entradas da máquina, conforme segue:

- T1: a máquina permanece no estado **A** enquanto o valor na posição i do vetor de diferenças for maior que -300mm . O contador i é incrementado.
- T2: a máquina de estados transita do estado **A** para o **B** quando o valor na posição i do vetor de diferenças for menor ou igual a -300mm . Esse estado representa que foi encontrada a primeira transição brusca de valores. O contador i é incrementado.

- T3: a máquina de estados retorna ao estado **A** caso seja detectado um valor menor que -30mm ou maior que 30mm. Foram escolhidos esses valores pois após detectar a primeira transição, espera-se que as próximas diferenças sejam pequenas (superfície do tubo de 100mm de diâmetro utilizado). O contador i é incrementado.
- T4: caso o valor na posição i seja um número entre -30mm e 30mm, a máquina muda para o estado **C**. O contador i é incrementado.
- T5: a máquina permanece no estado **C** enquanto o valor na posição i for um número entre -30mm e 30mm. Os contadores i e j são incrementados.
- T6: se o valor na posição i for um número entre -30mm e 30mm e o contador j um número maior que 8 a máquina retorna ao estado inicial **A**. O contador i é incrementado e o contador j é reiniciado.
- T7: caso o valor na posição i seja um número maior que 300mm e o contador j um número entre 1 e 8, a máquina retorna para o estado **A**. É gerada uma saída que indica que a *landmark* foi encontrada.

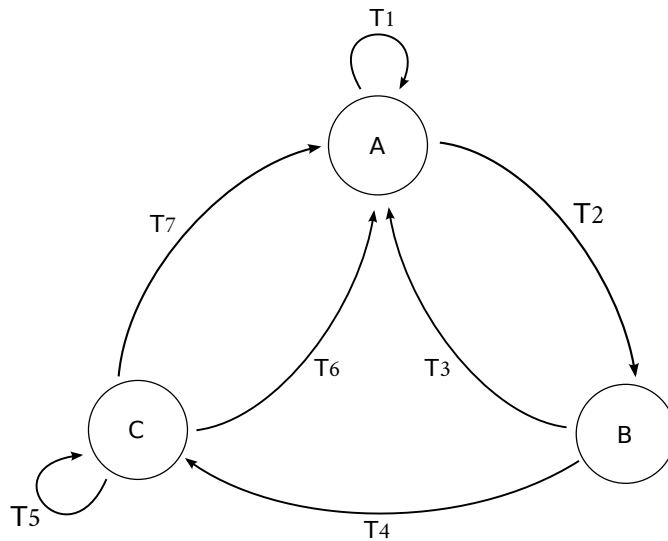


Figura 25 – Máquina de estados para detecção da *landmark*.

Após encontrar a *landmark*, a distância da mesma para o sensor laser de varredura é considerada como a menor distância encontrada no estado **C** (a menor distância entre o sensor e a marca). Porém, conforme pode ser observado na Figura 26 o sensor está posicionado a uma distância de 225mm do centro de rotação do robô.

Para corrigir esse deslocamento, deve-se relacionar a distância ρ_1 e o ângulo ψ de detecção da *landmark* conforme é apresentado na Equação 3.7.

$$\rho_2 = \sqrt{(\rho_1 \cos(\psi) + 250)^2 + (\rho_1 \sin(\psi))^2} \quad (3.7)$$

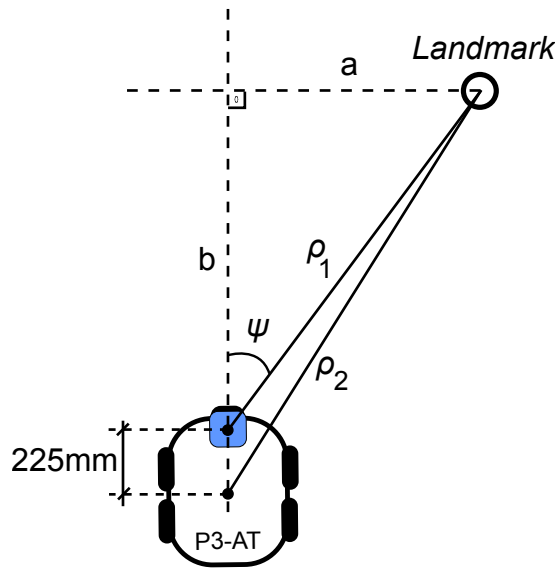


Figura 26 – Correção da medição do sensor laser de varredura em relação ao centro do robô.

A Figura 27 mostra uma situação onde o robô, situado na posição (X_O, Y_O) , de acordo com sua odometria, se aproxima de uma *landmark*. Nesse instante, utilizando somente a sua odometria, o seu erro angular calculado é α , a sua orientação é φ e o seu erro de posição é ρ .

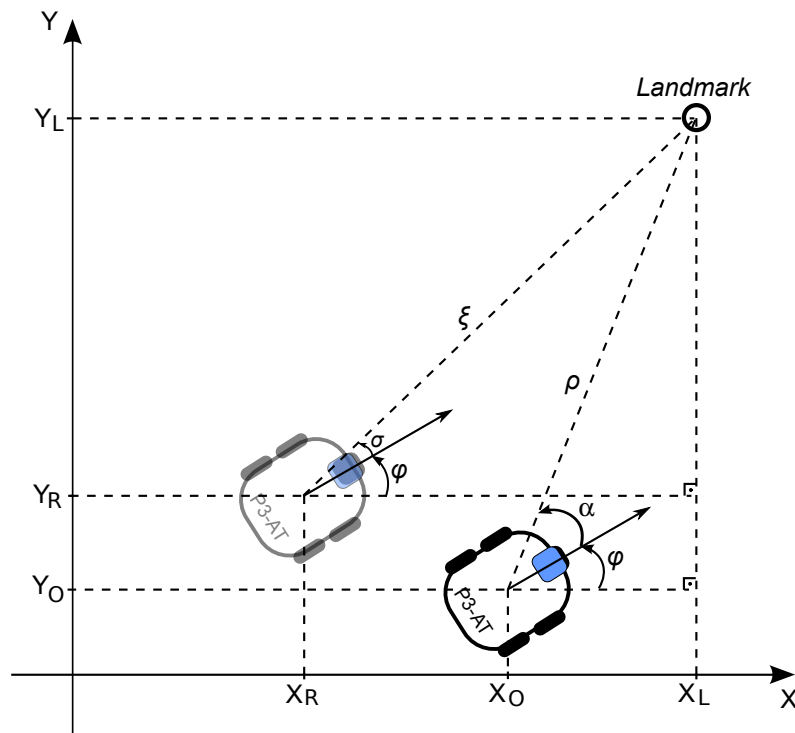


Figura 27 – Correção de posição utilizando as informações do sensor laser de varredura.

Após a detecção da marca pelo sensor laser de varredura a posição do robô é corrigida para um novo ponto calculado (X_R, Y_R) , conforme a Figura 27. Esse ponto é

encontrado considerando que os novos erros angular e de posição são exatamente o ângulo σ , e a distância ξ , medidos pelo sensor laser de varredura, e que a orientação do robô é a mesma calculada pela odometria. A Equação 3.8 apresenta o cálculo de correção realizado, onde o ponto (X_L, Y_L) é a posição da *landmark* no mapa.

$$\begin{cases} X_R = X_L - \xi \cos(\sigma + \varphi) \\ Y_R = Y_L - \xi \sin(\sigma + \varphi) \end{cases} \quad (3.8)$$

Para validar a correção proposta, foi realizado um teste em um corredor onde o robô inicia o seu percurso no ponto $(0, 0)$ e tem como destino o ponto $(9, 0)$. Ao alcançar esse ponto, o controlador recebe um novo destino, o ponto $(18, 0)$. A Figura 28 apresenta o caminho percorrido pelo robô durante o experimento. As marcações vermelhas destacam os instantes onde foram realizadas as correções de posição pelo algoritmo desenvolvido.

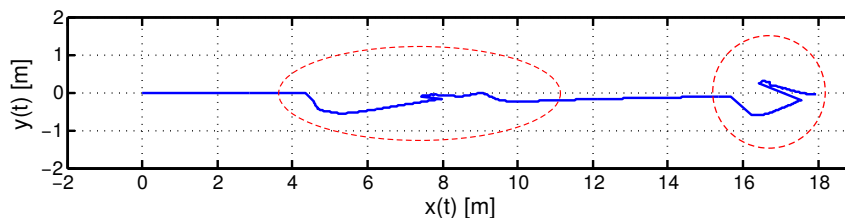


Figura 28 – Teste do algoritmo de correção de posição.

3.2 Planejamento de Rota - O Algoritmo de Dijkstra

Desde os anos 1980, os problemas de planejamento de movimento para robôs móveis tornaram-se um importante tópico de pesquisa que atraíram a atenção de muitos pesquisadores, os quais trabalharam extensivamente para obter métodos eficientes para resolver esses problemas (GARRIDO et al., 2006).

Existem diversos métodos para determinação de caminho em robótica móvel. Esse problema tem sido investigado e um grande número de abordagens heurísticas e não-heurísticas envolvendo movimento em duas ou três dimensões pode ser considerado (LUMELSKY; STEPANOV, 1986). Algumas das aplicações mais comuns utilizadas para planejamento de trajetória são *Roadmap* como Grafo de Visibilidade, Grafo de Voronoi (BHATTACHARYA; GAVRILOVA, 2007; MOHAMMADI; HAZAR, 2009), *Silhouette*; Decomposição em Células como Decomposição Exata, Decomposição Aproximada e Decomposição Hierárquica. Além dessas, também são utilizados alguns métodos mais modernos como Algoritmos Genéticos (GAO; FU; CHEN, 2005; SARIFF; BUNIYAMIN, 2010), Redes Neurais (BIN et al., 2004) e Otimização de Colônia de Formigas (BELL; MCMULLEN, 2004; BRAND et al., 2010; GIGRAS; GUPTA, 2012). Existem diversos algoritmos para buscas de rotas em um grafo. O algoritmo de Dijkstra (DIJKSTRA, 1959)

utiliza a técnica do relaxamento, que nada mais é que verificar se é possível melhorar o caminho obtido até o momento passando por um vértice diferente.

Neste trabalho parte-se do princípio que o mapa onde os robôs navegarão é previamente conhecido. A construção do mapa foi realizada de forma topológica utilizando-se somente nós, identificados por *landmarks*, conectados por caminhos, conforme a Figura 29. No mapa ilustrado, por exemplo, o nó número 1 apresenta ligação somente com os nós 2 e 4. Para cada um desses possíveis caminhos é atribuído um custo, que a princípio é diretamente proporcional à distância entre os nós em questão. Em relação aos outros pontos do mapa, o ponto 1 apresenta custo infinito. Posteriormente pretende-se agregar mais custos aos caminhos, por exemplo informações de trânsito e condições físicas do caminho entre os nós.

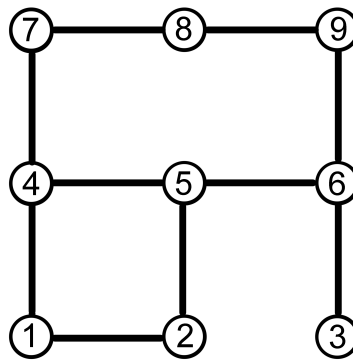


Figura 29 – Exemplo de mapa utilizado no trabalho.

Uma das intenções deste projeto é futuramente extrapolar a estratégia de cooperação e navegação para mapas de cidades e estradas obtidos via satélite, com a identificação de ruas, estradas e endereços de destino desejados. Assim, conhecendo-se o mapa, a posição inicial e o destino desejado, poder-se-á definir a rota de melhor relação custo-benefício e, durante a navegação, esse trajeto ser atualizado conforme a adversidades do caminho (trânsito intenso, manutenção nas vias, manifestações e acidentes).

Neste trabalho, para calcular o caminho de menor custo, é utilizado o Algoritmo de Dijkstra. O robô conhece a sua localização inicial e, ao receber o destino desejado, o algoritmo determina o caminho de menor custo para realização da navegação. Caso ocorra a detecção de obstáculos terrestres, tanto por parte do robô aéreo quanto por parte do robô terrestre, que impossibilitem a passagem pela rota que está sendo seguida, o programa encaminha o robô para o último nó pelo qual passou e atualiza o seu mapa, aumentando o peso do caminho obstruído. Com o robô posicionado nesse ponto é recalculada uma nova rota para atingir o objetivo inicial.

3.3 Processamento de Imagens

3.3.1 Detecção de *landmarks*

Como já mencionado, tanto o robô terrestre quanto o quadrimotor possuem câmeras embarcadas. As imagens capturadas por essas câmeras são utilizadas para detecção e identificação das *landmarks* (tanto no robô terrestre quanto no aéreo), e para detecção de obstáculos (robô aéreo).

As *landmarks* são compostas por dois códigos de barras, um linear, disposto na vertical e envolvendo um tubo de PVC de 100mm de diâmetro, e um bidimensional, disposto na horizontal em uma placa de madeira apoiada sobre o tubo. A estrutura montada de uma *landmarks* pode ser vista na Figura 30. Neste trabalho, o dado armazenado em cada um dos códigos de barras foi um numeral que indica um nó do mapa.



Figura 30 – *Landmark* utilizada nos experimentos.

O código de barras linear utilizado foi do tipo Code 128, que é capaz de codificar todos os 128 caracteres da tabela ASCII. O código é composto por seis seções apresentadas na Figura 31. São elas:

- Zona Calma (1)
- Caracter de Início (2)
- Dado (3)
- Caracter de Verificação (4)
- Caracter de Parada (2)
- Zona Calma (1)

O QR Code é um código de barras bidimensional que é facilmente detectado utilizando uma câmera. Este código é convertido em texto, podendo este ser uma URL,

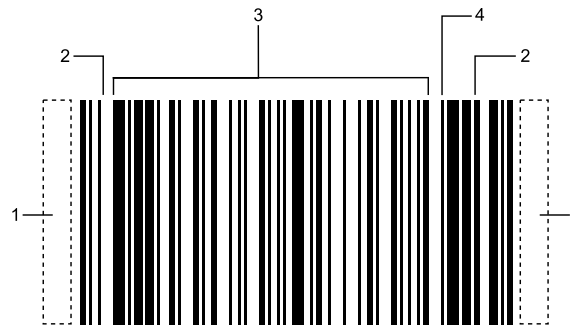


Figura 31 – Seções de um código de barras Code 128.

um número de telefone, uma localização georreferenciada ou qualquer outra informação na forma de texto. Inicialmente este código era utilizado para catalogar peças, mas hoje ele também é tipicamente encontrado em revistas e propagandas. Este código é composto por sete sub-regiões apresentadas na Figura 32, que são:

- Versão (1)
- Formato (2)
- Dados e chaves de verificação de erros (3)
- Padrões de referência (4)
- Padrão para posição (4.1)
- Padrão para alinhamento (4.2)
- Padrão para ajuste fino (4.3)

Sua capacidade de armazenamento depende do tipo de dado, podendo ser numérico, alfanumérico, binário ou kanji (símbolos utilizados na escrita Japonesa), além do tamanho, apresentando versões de 21x21, 25x25 e 177x177 módulos. O QR Code pode armazenar até 7089 caracteres na versão de 177x177 módulos e com tipo de dado numérico.

Para identificar os códigos de barras fixados nas landmarks foi utilizada a biblioteca de código aberto Zbar. Esta biblioteca permite a utilização de imagens provenientes de uma câmera, por exemplo, retornando o dado armazenado no código. Ela oferece suporte a diversas simbologias, como EAN-13/UPC-A, EAN-8, Code 128, Code 39 e QR Code, contemplando os dois tipos de códigos utilizados neste trabalho.

3.3.2 Detecção de obstáculos pelo Robô Aéreo

Uma das tarefas do robô aéreo é capturar imagens para determinar se uma determinada rota está livre ou obstruída por algum obstáculo. Enquanto navega pelo caminho

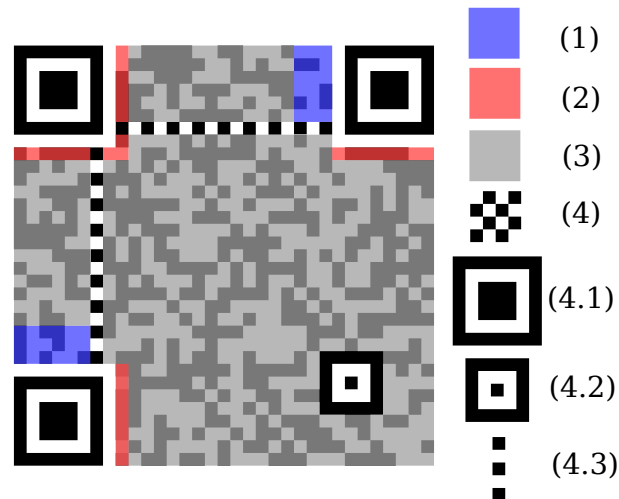


Figura 32 – Sub-regiões de um QR Code.

gerado pelo algoritmo de Dijkstra, o quadrimotor captura imagens através de sua câmera de bordo. Essas imagens são enviadas para um computador em terra e processadas, e extraem-se características que permitem realizar a detecção de obstáculos.

Nesta Dissertação de Mestrado foi utilizado um objeto colorido para indicar um obstáculo que o robô móvel não seria capaz de transpor. Por esse motivo, a técnica utilizada para detectar um obstáculo nas imagens fornecidas pela câmera do quadrimotor é a segmentação por cores.

Para realizar todo o processamento necessário sobre as imagens capturadas pela câmera do quadrimotor foi utilizada a biblioteca OpenCV (Open Source Computer Vision Library) (BRADSKI, 2000), que é uma biblioteca multiplataforma, totalmente livre para uso acadêmico e comercial, para o desenvolvimento de aplicativos na área de Visão Computacional.

A etapa de detecção do obstáculo se inicia com a captura da imagem pela câmera do quadrimotor e, em seguida, são realizadas operações de processamento de imagens para realizar a segmentação da cor escolhida no início da tarefa e, conseqüentemente, identificar o objeto. A segmentação se refere ao processo de dividir uma imagem em regiões, com o objetivo de simplificar a representação da imagem para facilitar sua análise.

As imagens capturadas pela câmera estão representadas no espaço de cores RGB (Red, Green, Blue) e são convertidas para o espaço de cores HSV (Hue, Saturation, Value). Essa conversão é feita porque no espaço de cores RGB uma cor é formada pela combinação do valor de cada componente R, G e B, enquanto no espaço de cores HSV a informação de cor está contida em apenas um canal, o canal-H. Isso torna o processo de segmentação de cores mais simples de se realizar.

Após a segmentação por cores ser realizada, é gerada uma imagem binarizada cujos pixels onde foi detectada a cor de interesse recebem o valor 1 (um) e todos os outros

recebem o valor 0 (zero). Com isso, verifica-se a área, em pixels, do maior objeto encontrado. Se essa área for maior que um limiar, considera-se que o obstáculo é intransponível pelo robô móvel e uma nova rota é calculada. A Figura 33 (a) mostra uma imagem capturada pela câmera contendo o obstáculo e a Figura 33 (b) mostra o resultado da segmentação por cores.



(a)



(b)

Figura 33 – Imagem capturada (a) e imagem segmentada (b).

CAPÍTULO 4

Software DESENVOLVIDO

Os métodos desenvolvidos no Capítulo 3 foram implementados utilizando a linguagem de programação C++ na IDE Qt Creator [®]. Essa IDE oferece desenvolvimento multiplataforma com diversas ferramentas para auxiliar no desenvolvimento de todo o ciclo do projeto. A seguir são apresentadas algumas ferramentas do Qt que estão integrados ao Qt Creator:

- Qt Designer: para projetar e construir interfaces gráficas de usuário (GUI) do Qt widgets.
- qmake: para a construção de aplicativos para diferentes plataformas de destino.
- Qt Assistant: para visualizar a documentação Qt.

O *software* desenvolvido para este trabalho possui uma interface gráfica para configurar e supervisionar o experimento de forma simplificada. A Figura 34 mostra uma tela da interface desenvolvida.

Ainda na Figura 34 é possível observar alguns detalhes da interface, como janelas para visualização do Mapa, das imagens capturadas pelas câmeras (robô terrestre e robô aéreo) e das mensagens geradas pelas funções implementadas. Na parte central estão situados os botões para interação como o usuário. Esses botões são utilizados para carregar o mapa, selecionar os pontos de origem e destino, calcular a rota para os robôs, iniciar a captura de imagens pelas câmeras, calibrar a cor do obstáculo, e dar início ao experimento. A Figura 35 mostra o fluxograma da estrutura geral do algoritmo desenvolvido nesta Dissertação de Mestrado.

Analisando a Figura 35 é possível observar de uma forma ampla o funcionamento do programa. Após as inicializações da biblioteca utilizada para interagir com o robô terrestre, a ARIA, assim como da biblioteca para comunicação com o sensor laser de

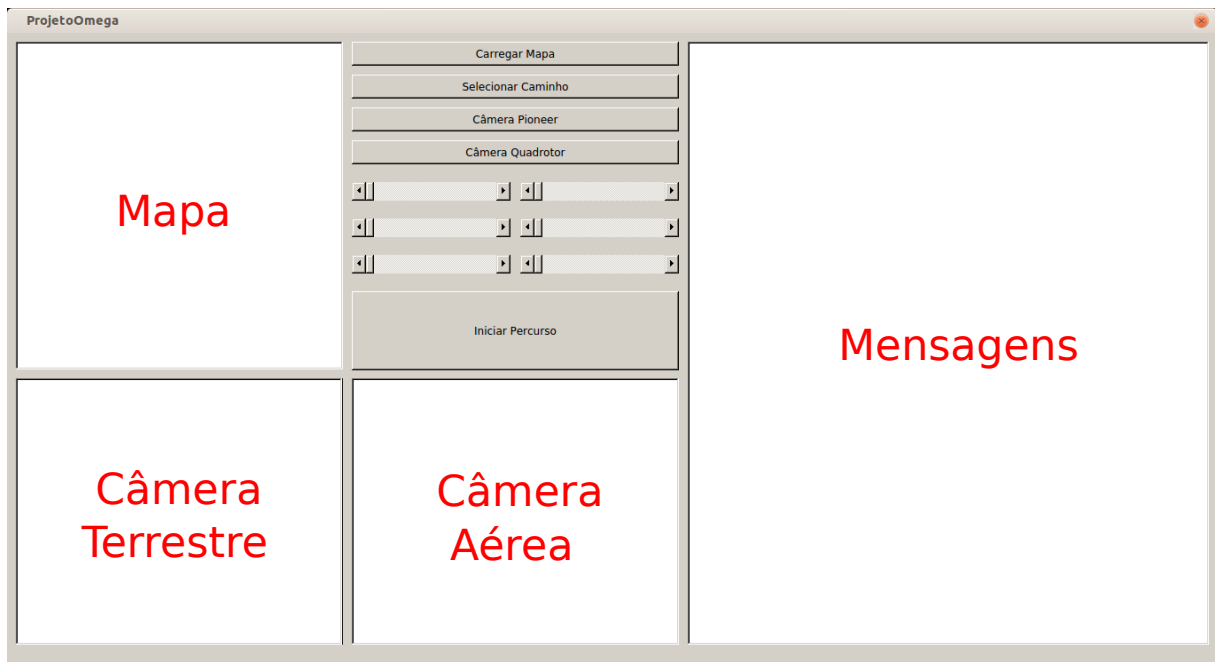


Figura 34 – Interface.

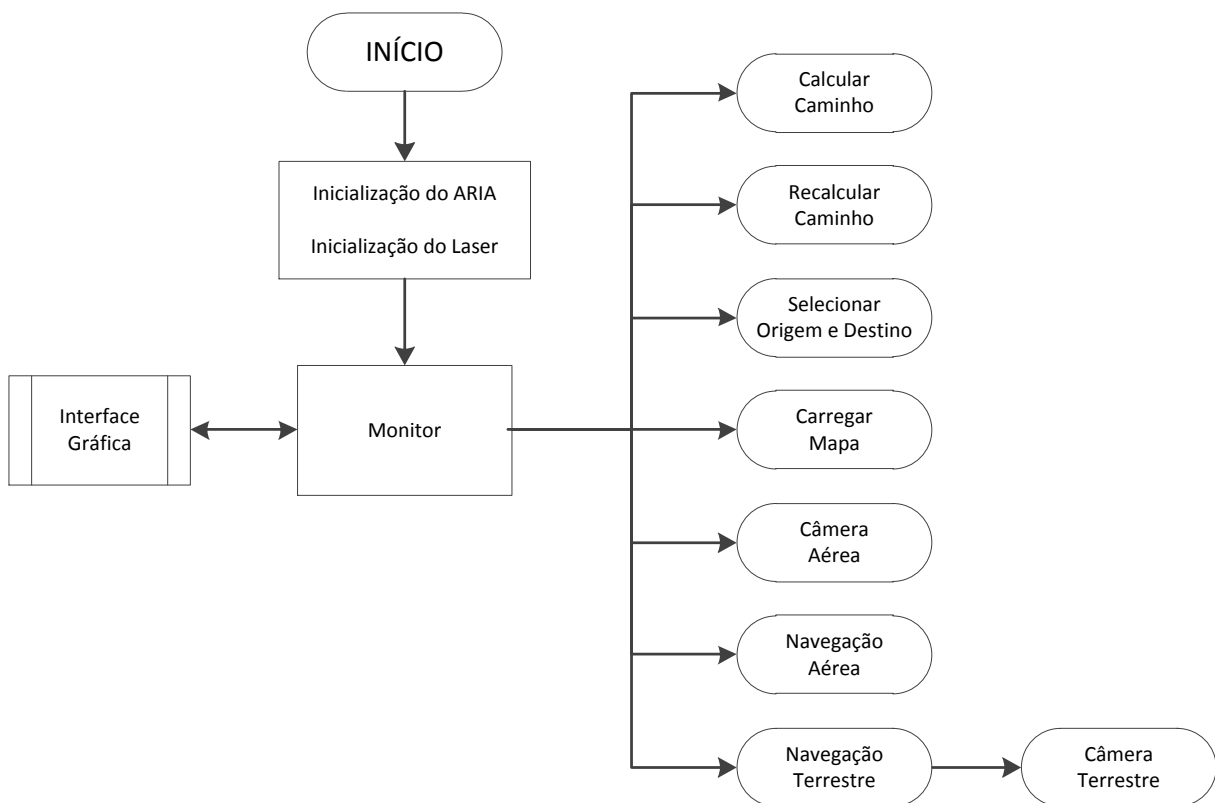


Figura 35 – Visão geral do *software* desenvolvido.

varredura, o programa é gerenciado por um Monitor. Esse monitor permite a integração da interface gráfica com as *threads* e funções criadas para executar os métodos propostos.

4.1 Carregar Mapa

A Figura 36 mostra o fluxograma da função criada para carregar um mapa quando o botão “Carregar Mapa” é pressionado na interface gráfica. Os blocos utilizados no fluxograma têm as seguintes funcionalidades:

- Ler Mapa Armazenado: carrega para o programa o mapa armazenado em um arquivo. Esse mapa é composto por uma matriz de zeros e uns, que determinam como os nós do mapa estão interligados, e pelas coordenadas cartesianas de cada um dos nós.
- Desenhar mapa: utilizar os dados gerados no bloco anterior para desenhar na interface gráfica o mapa armazenado.

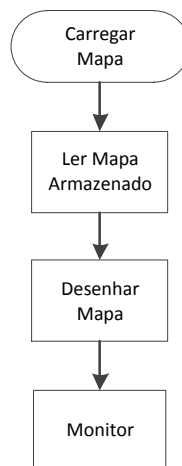


Figura 36 – Fluxograma da função utilizada para carregar o mapa.

4.2 Selecionar Origem e Destino

A Figura 37 mostra o fluxograma da função criada para selecionar a origem e o destino dos robôs quando o botão “Selecionar Caminho” é pressionado na interface gráfica.

Ao pressionar o botão, o programa solicita (são impressas mensagens de orientação na interface gráfica) que o usuário clique com o *mouse* sobre o nó de origem dos robôs, e, logo em seguida, sobre o nó de destino. Após realizar a seleção, a função “Calcular Caminho” é chamada. Os blocos utilizados no fluxograma têm as seguintes funcionalidades:

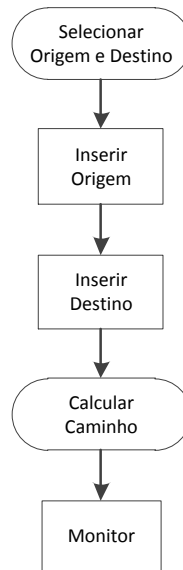


Figura 37 – Fluxograma da função utilizada para selecionar a origem e o destino dos robôs.

- Inserir Origem: imprimir mensagem na interface gráfica solicitando que o usuário clique com o *mouse* sobre o nó de origem dos robôs. As coordenadas do nó selecionado são armazenadas para o cálculo do caminho.
- Inserir Destino: imprimir mensagem na interface gráfica solicitando que o usuário clique com o *mouse* sobre o nó de destino dos robôs. As coordenadas do nó selecionado são armazenadas para o cálculo do caminho.

4.3 Calcular Caminho

Na Figura 38 é apresentado o fluxograma da função desenvolvida para calcular a rota por onde cada um dos robôs deve passar para atingir o objetivo.

As tarefas desenvolvidas por cada um dos blocos representados no fluxograma são descritas a seguir:

- Calcular Rota Terrestre: calcular a rota do robô terrestre com base nas informações de origem e destino informadas pelo usuário.
- Calcular Rota Aérea: calcular a rota do robô aéreo com base nas informações de origem e destino selecionadas pelo usuário. Apesar de inicialmente as duas rotas serem idênticas, posteriormente, quando ocorrer a detecção de um obstáculo pelo robô aéreo que é intransponível pelo robô terrestre, as rotas poderão ser diferentes em função da localização de cada um dos veículos nesse instante.
- Desenhar Rotas: desenhar as rotas calculadas anteriormente no mapa da interface gráfica.

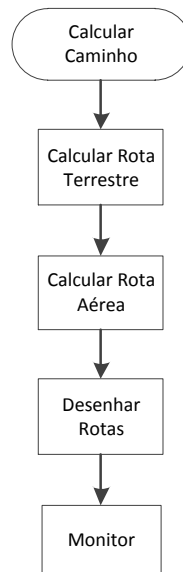


Figura 38 – Fluxograma da função utilizada para calcular as rotas dos robôs.

4.4 Recalcular Caminho

A Figura 39 apresenta o fluxograma da função desenvolvida para realizar o recálculo de rota quando um obstáculo terrestre intransponível é detectado.

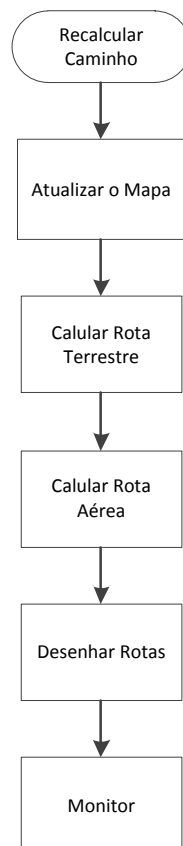


Figura 39 – Fluxograma da função utilizada para recalculas rotas dos robôs.

As tarefas desenvolvidas por cada um dos blocos representados no fluxograma são

descritas a seguir:

- Atualizar o Mapa: substituir no mapa armazenado no programa o caminho obstruído por um caminho impossível.
- Calcular Rota Terrestre: calcular a rota do robô terrestre, com base no novo mapa e na posição atual do robô.
- Calcular Rota Aérea: calcular a rota do robô aéreo, com base no novo mapa e na posição atual do robô.
- Desenhar Rotas: desenhar as rotas calculadas anteriormente no mapa da interface gráfica.

4.5 Navegação Terrestre

A Figura 40 apresenta o fluxograma da *thread* desenvolvida para realizar a navegação do robô terrestre. O algoritmo inicia com a atualização do ponto de origem do robô, que gira na direção do próximo nó para onde ele deve navegar. Após atingir a orientação desejada, o robô terrestre inicia a navegação para a próxima marca, até alcançar um erro de posição menor que 5cm. Durante esse percurso, pode ser observado que a cada laço de controle o robô verifica a presença de obstáculos na sua rota. Caso seja identificado um obstáculo o algoritmo gera um ponto virtual, conforme descrito no Capítulo 3, para executar o desvio. Se o obstáculo for intransponível, é enviado um comando para que o referido robô retorne até o último nó por onde passou, uma nova rota é calculada e o algoritmo é reiniciado.

Quando o erro linear é menor que 4m, a *thread* “Câmera Terrestre” é iniciada. Antes que o robô chegue até o nó desejado, espera-se que a *landmark* seja detectada pela câmera do robô, confirmando a sua localização. Caso o robô chegue ao nó pretendido sem a confirmação de que a *landmark* foi detectada pela câmera, ele realiza uma rotação de 360° em torno do seu próprio eixo, a fim de localizar a marca. Se ainda assim a *landmark* não for localizada, é enviado um comando para que o robô retorne até o último nó por onde passou; uma nova rota é calculada e o algoritmo é reiniciado.

As tarefas desenvolvidas por cada um dos blocos representados no fluxograma da Figura 40 são descritas a seguir:

- Girar na direção da *landmark*: utilizar o controlador desenvolvido, porém com velocidade linear zero, para que o robô gire na direção da *landmark* que deseja alcançar.

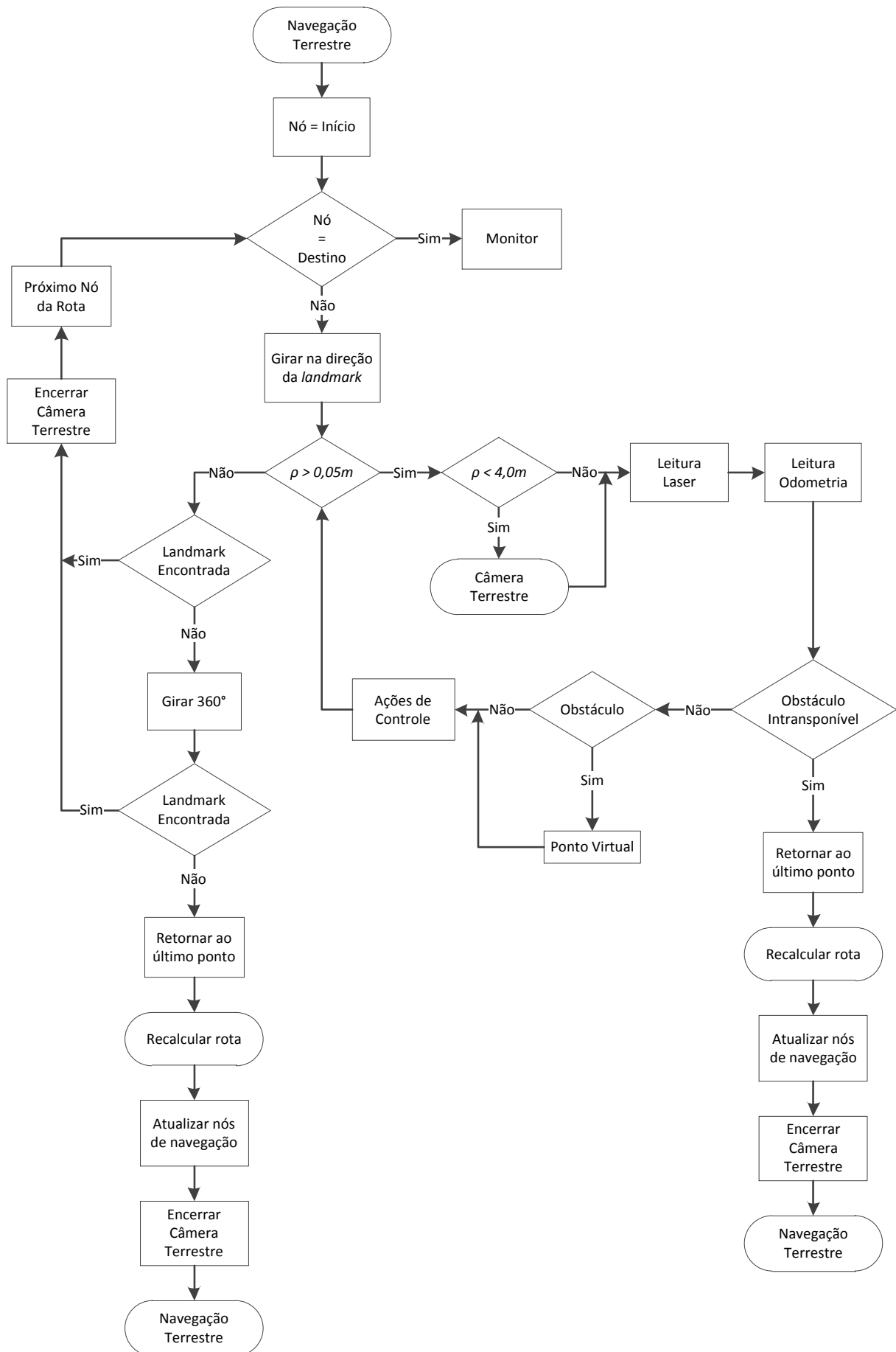


Figura 40 – Fluxograma da *thread* desenvolvida para a navegação do robô terrestre.

- Câmera Terrestre: iniciar a *thread* da câmera do robô terrestre. A *thread* é inicializada sempre que o robô chega a uma distância de 4m do nó.
- Leitura Laser: realizar uma varredura com o sensor laser de varredura e armazenar os valores encontrados em um vetor.
- Leitura Odometria: realizar leitura do *encoder* do robô.
- Ponto Virtual: criar temporariamente um ponto virtual para desviar o robô terrestre do obstáculo.
- Ações de controle: realizar as contas descritas para o controlador e enviar para o robô os valores de velocidades angular e linear calculadas.
- Retornar ao último ponto: realizar um giro de 180° e retornar com o robô para o último nó conhecido.
- Atualizar nós de navegação: atualizar o vetor com os nós que o robô terrestre deve seguir para chegar ao destino.
- Encerrar Câmera Terrestre: encerrar a *thread* da Câmera Terrestre.

Para realizar a detecção de um obstáculo intransponível pelo robô terrestre, são consideradas apenas as medidas do sensor laser de varredura correspondentes a 45° para direita e esquerda em relação à orientação do veículo, conforme pode ser visto na Figura 41. O obstáculo é considerado intransponível quando não existir, a uma distância de 1,2m do sensor laser de varredura, uma abertura com largura “d” maior que 120% da largura do robô. Tal valor foi escolhido empiricamente, sendo considerado suficiente para a passagem do robô com segurança.

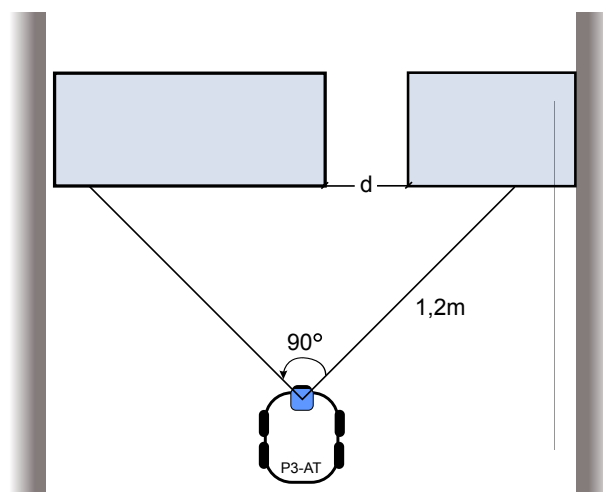


Figura 41 – Robô terrestre se aproximando de um obstáculo intransponível.

4.6 Câmera Terrestre

A Figura 42 apresenta o fluxograma da *thread* desenvolvida para identificar as *landmarks* da rota do robô, e corrigir a sua odometria. Nesse algoritmo, após as configurações iniciais, o programa inicia um *loop* infinito, e só é encerrado quando a *thread* “Navegação Terrestre” solicita o desligamento da câmera. Após o início do *loop* é iniciado o procedimento para busca da *landmark* do nó em questão. Quando a marca é reconhecida três vezes ($k=3$), é enviado um sinal para a *thread* “Navegação Terrestre” informando a detecção do nó nas imagens. Em seguida, inicia-se o processo de identificação da *landmark* no vetor de medidas do sensor laser de varredura, composto por 30 medidas (15° à esquerda até 15° à direita do ponto onde a marca foi identificada na imagem). Caso a *landmark* seja encontrada, as medidas armazenadas no vetor são utilizadas para corrigir as medidas da odometria do robô terrestre.

As tarefas desenvolvidas por cada um dos blocos representados no fluxograma da Figura 42 são descritas a seguir:

- Configuração Zbar: configurar a biblioteca Zbar para capturar um código de barras do tipo Code 128.
- Procurar Code 128: buscar na imagem capturada pelo robô terrestre um código de barras do tipo Code 128.
- Variável k: utilizada para contar quantas vezes o código de barras procurado foi encontrado.
- Variável i: utilizada para percorrer o vetor com medições do sensor laser de varredura.
- Máquina de Estados: executar a máquina de estados para detecção da *landmark* pelo sensor laser de varredura.
- Correção da Posição: realizar a correção de posição do robô e atualizar a sua odometria com os novos valores. Nesse instante também é comunicada à *thread* de navegação do robô que a marca foi encontrada.

4.7 Navegação Aérea

A Figura 43 apresenta o fluxograma da *thread* desenvolvida para monitorar a navegação do robô aéreo. Como a navegação do quadrimotor é pré-programada em seu piloto automático, ou conduzida por um ser humano, esse programa é utilizado para verificar por quais pontos o veículo aéreo passou. Dessa forma, quando um obstáculo é identificado pela *thread* “Navegação Aérea” o algoritmo solicita um recálculo de caminho, informando para essa função o caminho que está obstruído.

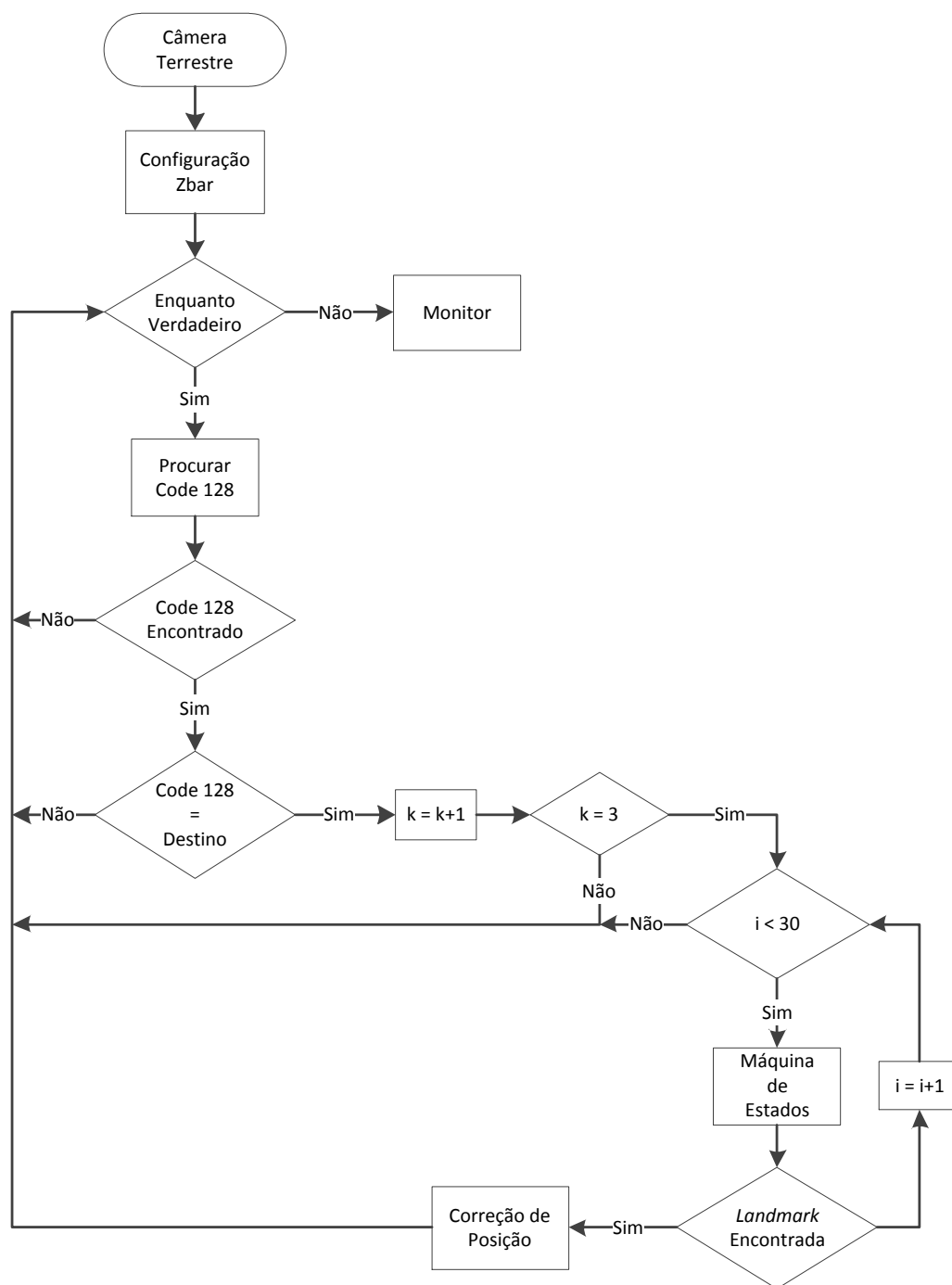


Figura 42 – Fluxograma da *thread* desenvolvida para a câmera do robô terrestre.

As tarefas desenvolvidas por cada um dos blocos representados no fluxograma são descritas a seguir:

- Sinal: ponto alcançado pelo quadrimotor. Esse sinal é emitido pela *thread* da câmera do robô aéreo, indicando que a *landmark* procurada foi encontrada.
- Sinal: obstáculo. Esse sinal é emitido pela *thread* da câmera do robô aéreo, indicando que foi encontrado um obstáculo terrestre.

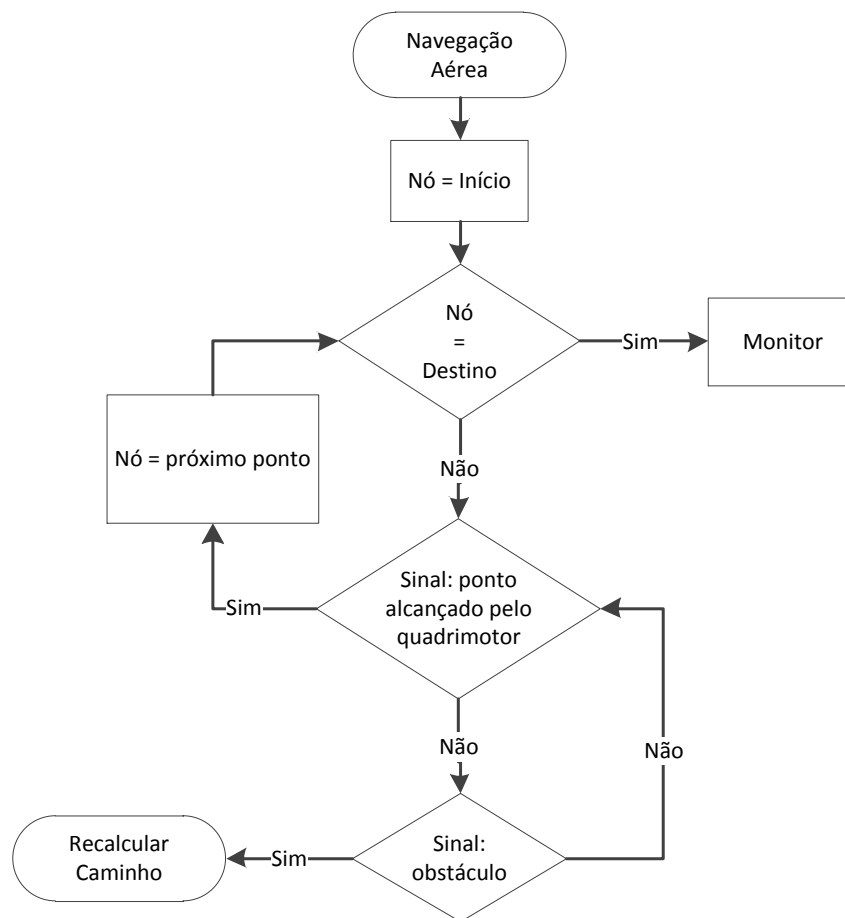


Figura 43 – Fluxograma da *thread* desenvolvida para a navegação do robô aéreo.

4.8 Câmera Aérea

A Figura 44 apresenta o fluxograma da *thread* desenvolvida para identificar obstáculos na rota calculada para o robô, e também detectar as *landmarks* do seu caminho. Nesse algoritmo, após as configurações iniciais, o programa inicia um *loop* infinito, e só é encerrado quando o robô chega no destino final. Após o início do *loop* são procurados obstáculos nas imagens capturadas, e em seguida é iniciado o procedimento para busca da *landmark* do nó em questão. Quando a marca é reconhecida três vezes ($k=3$), é enviado um sinal para a *thread* “Navegação Aérea” informando a detecção do nó nas imagens.

As tarefas desenvolvidas por cada um dos blocos representados no fluxograma da Figura 44 são descritas a seguir:

- Calibração da Cor do Obstáculo: calibrar a cor do obstáculo utilizado no experimento. O usuário deve posicionar a câmera sobre o obstáculo e utilizar a janela da câmera do robô aéreo na interface gráfica para clicar sobre o obstáculo. Por fim, para ajustar os valores da cor selecionada, o usuário pode utilizar as barras deslizantes da interface para ajustar os canais HSV do filtro utilizado nessa *thread*.
- Configuração Zbar: configurar a biblioteca Zbar para capturar um QR Code.
- Processamento de Imagem: realizar a segmentação da imagem capturada pelo quadrimotor para procurar por obstáculos na mesma.
- Sinal: obstáculo. Emitir sinal para *thread* Navegação Aérea indicando que um obstáculo terrestre foi encontrado.
- Procurar QR Code: buscar na imagem capturada pelo robô aéreo um QR Code.
- Sinal: ponto alcançado pelo quadrimotor. Emitir sinal para *thread* Navegação Aérea indicando que o nó foi encontrado.

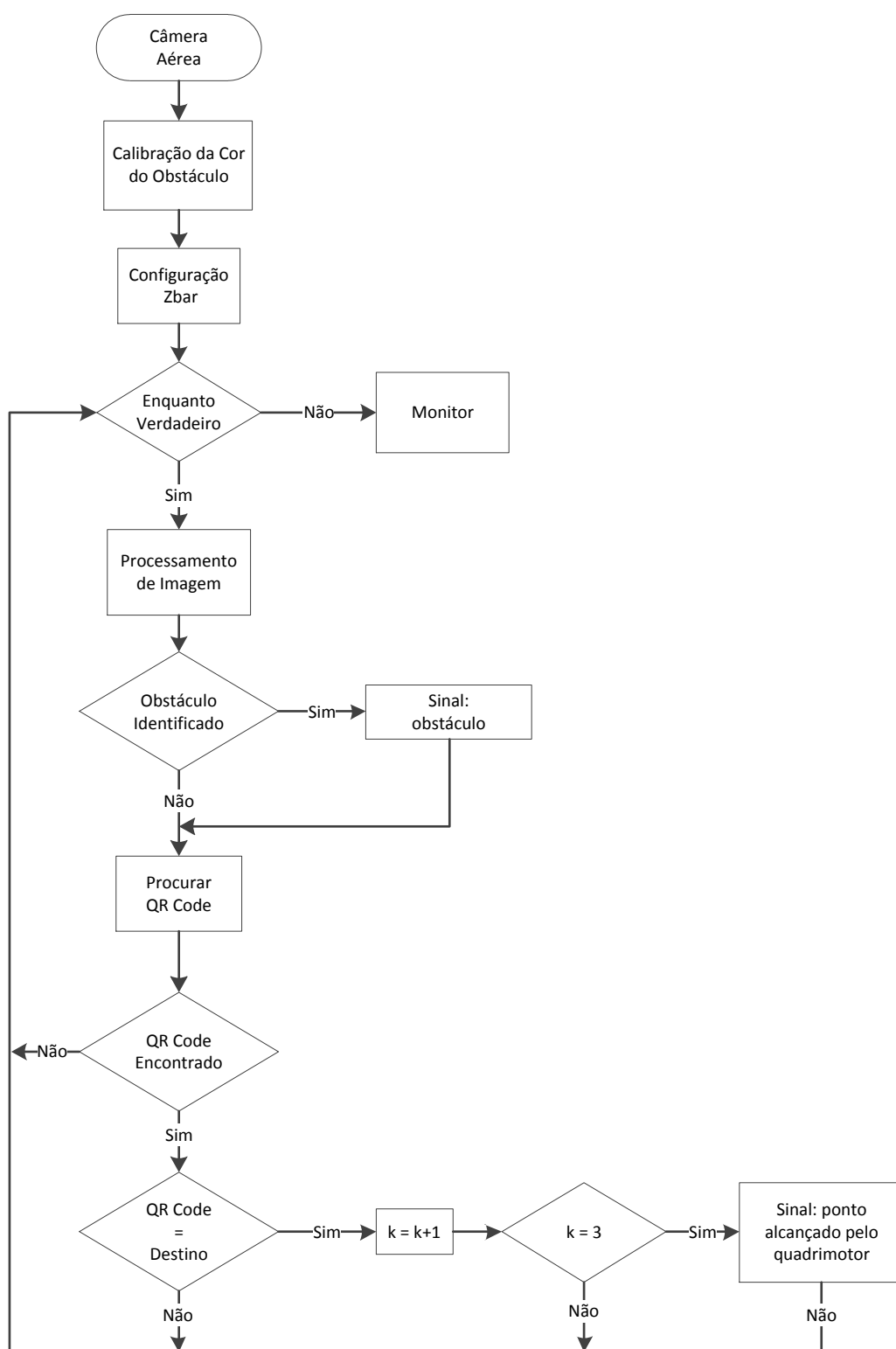


Figura 44 – Fluxograma da *thread* desenvolvida para a câmera do robô aéreo.

4.9 Simulação

Para testar os algoritmos desenvolvidos para o robô terrestre, foram realizadas simulações. A Figura 45 apresenta o resultado de um teste de navegação em um mapa. O círculo maior, de cor cinza, destaca o objetivo a ser alcançado pelo robô.

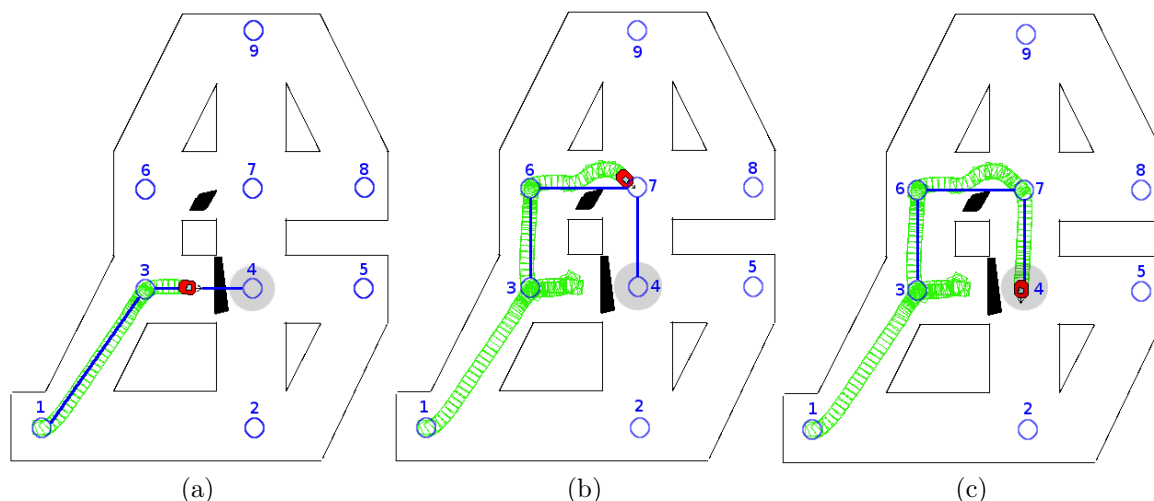


Figura 45 – Simulação do robô terrestre: identificação de obstrução (a), desvio de obstáculo (b) e destino alcançado (c).

Como pode ser visto na Figura 45 (a), a rota de menor custo é formada pelos caminhos 1-3 e 3-4. Ainda na mesma figura, é destacado o momento no qual o robô detecta uma obstrução na rota no trecho 3-4. Após a identificação do bloqueio, o robô retorna ao nó 3 e recalcula uma nova rota de menor custo, desconsiderando o caminho 3-4.

Conforme apresentado na Figura 45 (b), a nova rota é formada pelos caminhos 3-6, 6-7 e 7-4. Pode-se observar também que o veículo realizou um desvio de obstáculo entre os pontos 6-7. Completando a simulação, a Figura 45 (c) mostra o robô terrestre atingindo o ponto final.

A função do quadrimotor neste trabalho é detectar obstáculos intransponíveis no caminho que será realizado pelo robô terrestre, para que este possa recalculá-la sua rota antes mesmo de encontrar o obstáculo. Dessa forma, independente da forma que o obstáculo intransponível é identificado (se pelo sensor laser de varredura do robô terrestre ou pela câmera do veículo aéreo) a simulação anterior prova que o robô terrestre é capaz de recalculá-la sua rota. Dessa maneira não foi necessário realizar simulação com o quadrimotor.

CAPÍTULO 5

RESULTADOS

Com o objetivo de validar a estratégia de cooperação entre um robô aéreo e um robô terrestre, proposta nesta Dissertação de Mestrado, foram realizados experimentos utilizando estes robôs em ambientes externos. Neste capítulo são apresentados 4 resultados experimentais. O Experimento 1 mostra o robô terrestre em navegação solo (sem o auxílio do veículo aéreo), onde o mesmo alcança o objetivo selecionado pelo usuário desviando de um obstáculo. No Experimento 2, o robô terrestre, também em navegação solo, testa o algoritmo de recálculo de rota devido a uma obstrução no caminho.

Os Experimentos 3 e 4 foram realizados para comprovar a funcionalidade do algoritmo de cooperação entre o robô aéreo e o terrestre. No terceiro teste é realizado um percurso com o robô aéreo no modo manual. Ao se detectar um obstáculo terrestre através da imagem aérea as rotas de ambos robôs são recalculadas. No último teste, o quadrimotor navega no modo autônomo, utilizando uma rota pré-programada em seu piloto automático, necessitando do usuário somente para a decolagem e aterrizagem. Quando o obstáculo é identificado pelo robô aéreo, o mesmo continua o percurso inicial, alterando somente o caminho para o robô em terra, o Pioneer 3-AT.

5.1 Experimento 1: Desvio de obstáculo

Neste experimento o robô terrestre navega sem o auxílio do veículo aéreo, e tem como principal objetivo alcançar um ponto de destino efetuando desvio de obstáculos. Para a realização deste experimento foi construído o mapa da Figura 46 em uma quadra poliesportiva.

Para todos os experimentos as *landmarks* foram deslocadas de uma distância de 1 metro na coordenada x e -1 metro na coordenada y, em relação aos nós por onde o robô deve passar. A Figura 47 mostra o robô posicionado no nó cujas coordenadas cartesianas

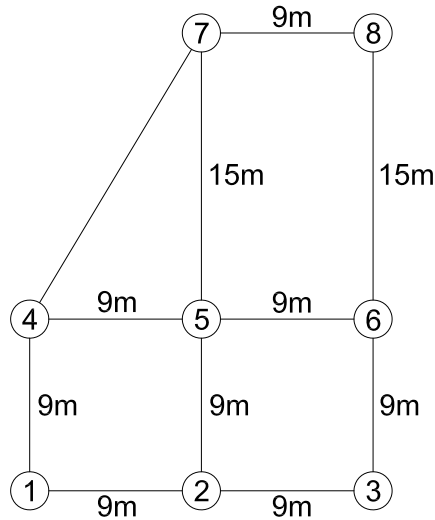


Figura 46 – Mapa utilizado nos experimentos 1 e 2.

são $(0, 0)$ m, e a *landmark* deslocada para a posição $(1, -1)$ m. Esse deslocamento foi realizado para retirar a *landmark* do caminho por onde o robô deve passar, e permitir que o mesmo possa detectá-la e efetuar a sua correção de odometria quando necessária.

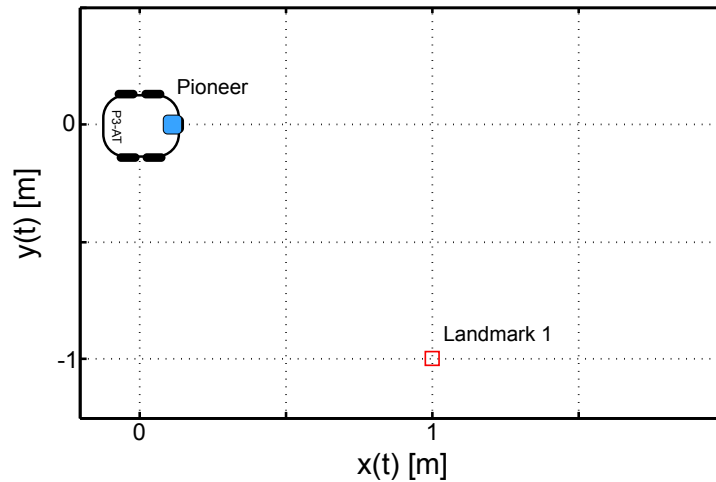


Figura 47 – Deslocamento das *landmarks* para os experimentos.

Neste experimento foi utilizado um obstáculo com altura suficiente para ser detectado pelo sensor laser de varredura do robô móvel. Esse obstáculo foi colocado entre as marcas 5 e 6, exatamente no caminho por onde o robô deveria passar para atingir o destino final, conforme a Figura 48.

Com o robô posicionado na *landmark* 1 foi solicitado pelo usuário que ele navegasse até o ponto 6 como destino final. Na Figura 49 é apresentada a interface gráfica após o programa ter calculado o caminho, na cor verde, que o veículo deve seguir para atingir a marca desejada.

A Figura 50 apresenta a interface do programa após a realização do percurso. Nesta



Figura 48 – Obstáculo utilizado no experimento 1.

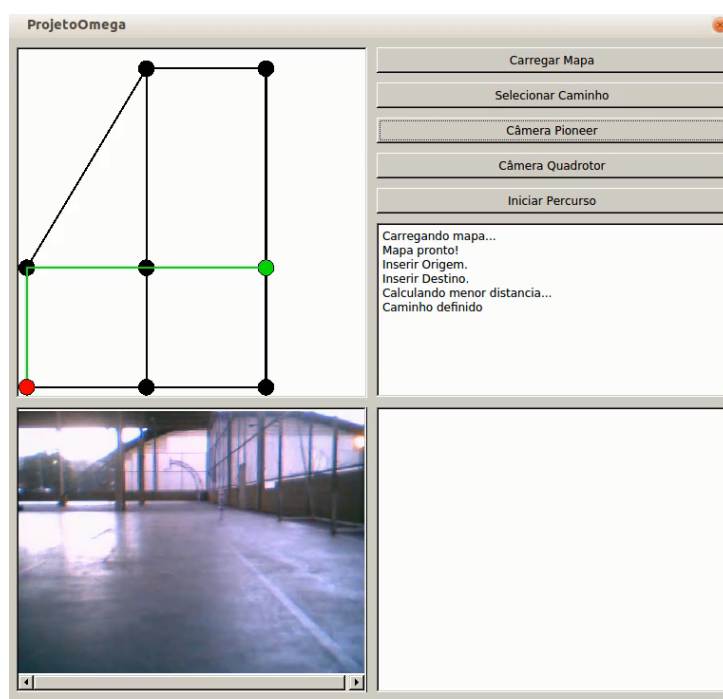


Figura 49 – Caminho calculado no experimento 1.

imagem é possível observar o caminho percorrido pelo robô, em vermelho, e as mensagens de detecção das *landmarks* encontradas.

A odometria do robô pode ser vista na Figura 51. Em verde estão destacados os momentos onde as medidas do sensor laser de varredura foram utilizadas para corrigir o erro de posição do veículo. O tracejado vermelho indica o ponto onde o robô identificou o obstáculo e realizou o desvio.

As Figuras 52 (a) e 52 (b) apresentam, respectivamente, as velocidades linear e angular desenvolvidas pelo robô móvel durante o percurso. Próximo aos instantes de tempo 50s, 100s e 150s, no gráfico de velocidade linear (Figura 52 (a)), é possível observar uma

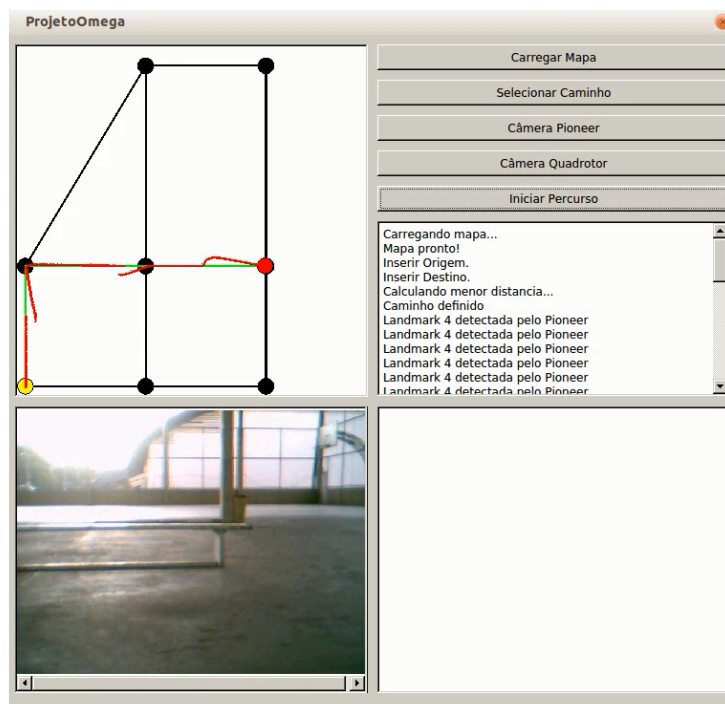


Figura 50 – Caminho percorrido no experimento 1.

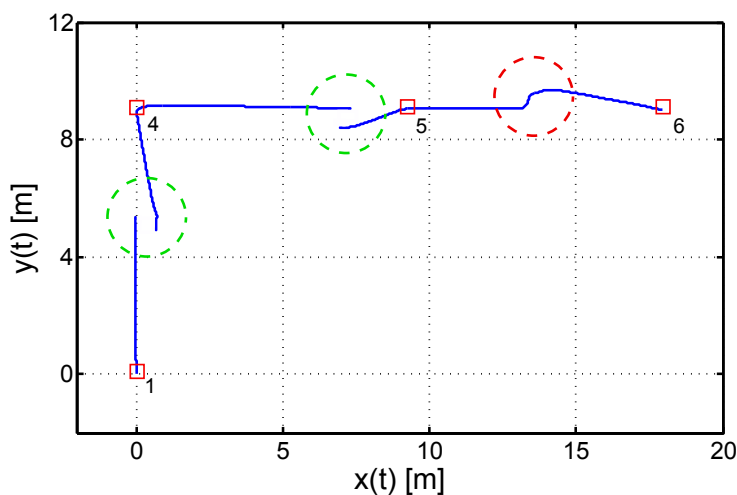


Figura 51 – Odometria do robô terrestre no experimento 1.

desaceleração do robô, que são os momentos onde o veículo se aproxima das *landmarks*. Essa desaceleração ocorre pois o robô terrestre utiliza um controlador de posição final, e dessa forma, para o controlador cada ponto da rota de navegação é um ponto final que o robô deve atingir, com velocidades linear e angular iguais a zero. No gráfico de velocidade angular (Figura 52 (b)) podem-se destacar os pontos próximos aos tempos de 30s e 80s, os quais se referem à correção da posição dada pela odometria do robô devido às medidas obtidas pelo sensor laser de varredura. Ainda na Figura 52 (b), os pontos próximos aos instantes de tempo 50s e 100s correspondem, respectivamente, ao giro de 90° na marca 4 e à correção na marca 5 para alinhar o robô em direção aos próximos destinos. Tanto na Figura 52 (a) quanto na Figura 52 (b) podem-se observar as variações das velocidades

linear e angular no instante de tempo 120s, em que o robô realiza a operação de desvio de obstáculo.

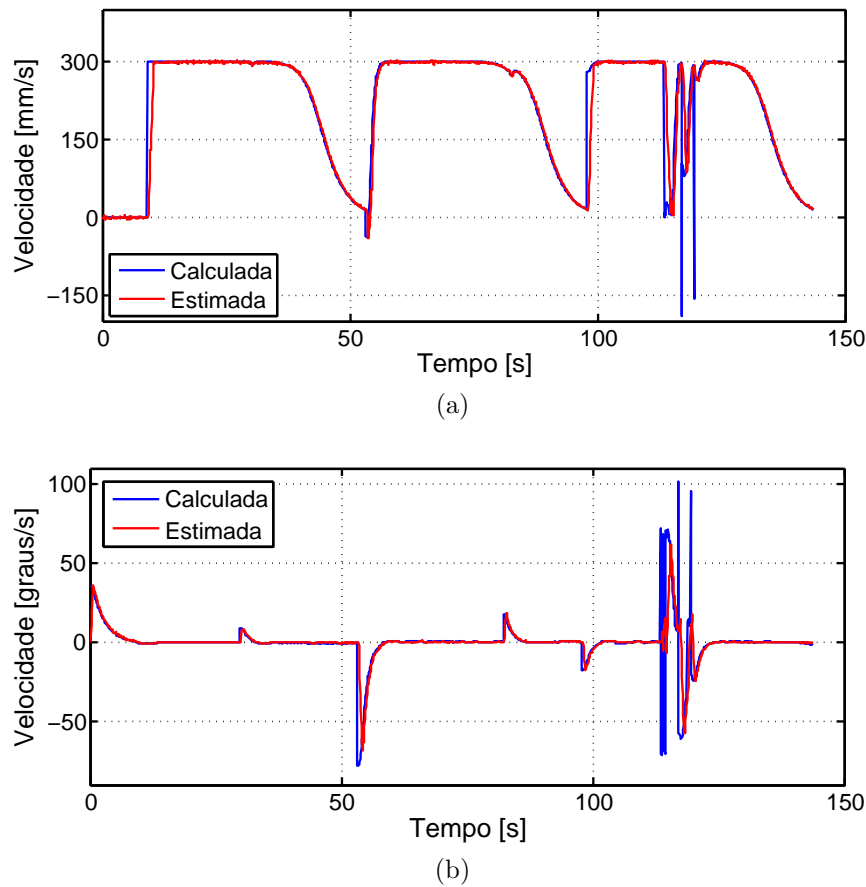


Figura 52 – Velocidades linear (a) e angular (b) do robô terrestre no experimento 1.

Observa-se, pelos resultados deste experimento, que o robô móvel foi capaz de calcular uma rota e percorrê-la, identificando as marcas conhecidas, realizando a correção da sua posição e desviando de eventuais obstáculos.

5.2 Experimento 2: Recálculo de rota

Neste experimento o robô terrestre navega, assim como no Experimento 1, sem o auxílio do veículo aéreo, e tem como principal objetivo alcançar um ponto de destino efetuando um recálculo de rota, devido a uma obstrução encontrada no seu caminho. Para a realização deste experimento foi utilizada a mesma topologia apresentada na Figura 46 da Seção 5.1, em uma quadra poliesportiva. Com o robô posicionado na marca 2 foi indicado pelo usuário o ponto 7 como destino final. Na Figura 53 pode ser vista a interface gráfica após o programa ter calculado o caminho, na cor verde, que o veículo deve seguir para atingir a marca desejada.

Para simular um obstáculo intransponível para o robô terrestre foi introduzido

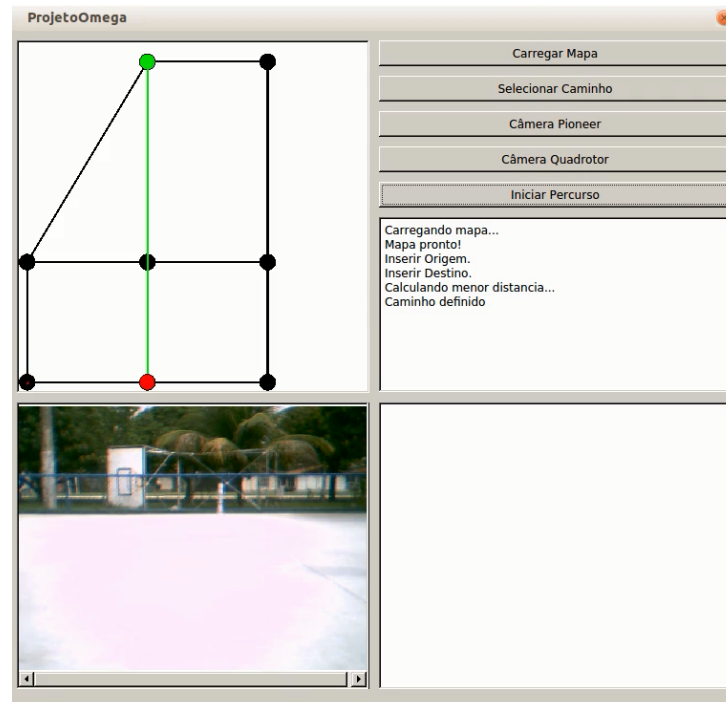


Figura 53 – Caminho calculado no experimento 2.

entre as marcas 5 e 7 um conjunto composto por uma caixa e duas placas metálicas, conforme a Figura 54.



Figura 54 – Obstáculo utilizado no experimento 2.

A Figura 55 apresenta a interface do programa após a realização do percurso. Nessa imagem é possível observar o caminho percorrido pelo robô, em vermelho, e as mensagens de detecção das *landmarks* encontradas e do momento onde a rota foi recalculada.

A odometria do robô pode ser vista na Figura 56. Em verde estão destacados os momentos onde as medidas do sensor laser de varredura foram utilizadas para corrigir o

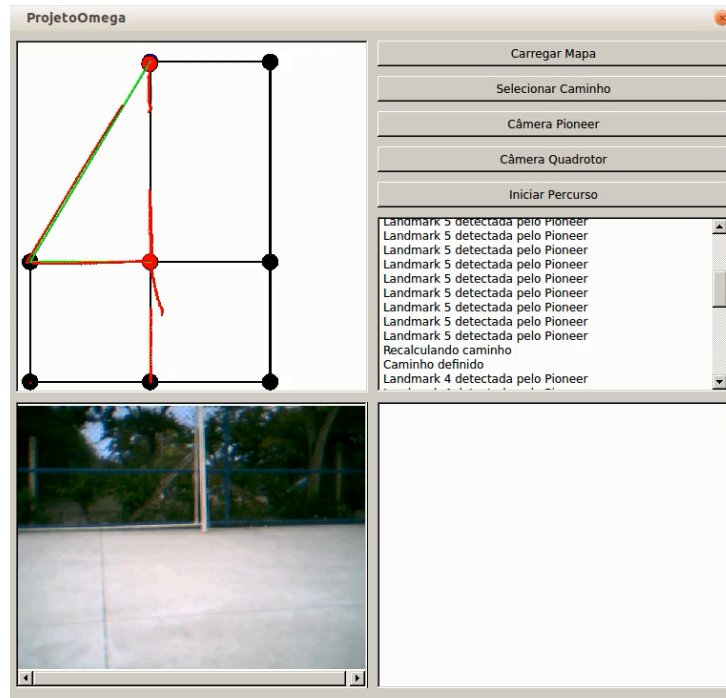


Figura 55 – Caminho percorrido no experimento 2.

erro de posição do veículo. O tracejado vermelho indica o ponto onde o robô identificou o obstáculo intransponível e realizou o recálculo de rota.

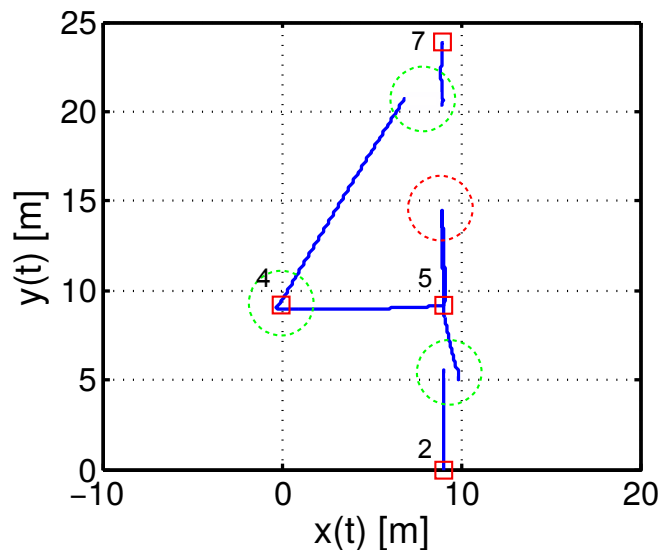


Figura 56 – Odometria do robô terrestre no experimento 2.

As Figuras 57 (a) e 57 (b) apresentam, respectivamente, as velocidades linear e angular desenvolvidas pelo robô móvel durante o percurso. Próximo aos instantes de tempo 50s, 130s, 150s e 220s, no gráfico de velocidade linear (Figura 57 (a)), é possível observar uma desaceleração do robô, que são os momentos onde o veículo se aproxima das *landmarks*. No gráfico de velocidade angular (Figura 57 (b)) podem-se destacar os pontos próximos aos tempos de 30s e 250s, os quais se referem à correção da posição dada pela

odometria do robô devido às medidas realizadas pelo laser. Ainda na Figura 57 (b), os pontos próximos aos instantes de tempo 125s e 185s correspondem, respectivamente, ao giro de 90° na marca 5 e ao giro de aproximadamente 110° na marca 4, para alinhar o robô em direção aos próximos destinos. Continuando na mesma figura, próximo ao instante de tempo 175s, o veículo alcançou o ponto sem identificar a *landmark* nas imagens capturadas. Nesse caso, o robô desenvolveu uma velocidade angular, até encontrar a marca, e em seguida uma velocidade linear para corrigir a posição, conforme pode ser observado no mesmo instante de tempo na Figura 57 (a). Tanto na Figura 57 (a) quanto na Figura 57 (b) pode-se observar as variações das velocidades, linear e angular, no instante de tempo 70s, que correspondem ao instante em que o robô faz a detecção da obstrução no caminho e realiza um giro de 180° para seguir em direção ao último ponto conhecido.

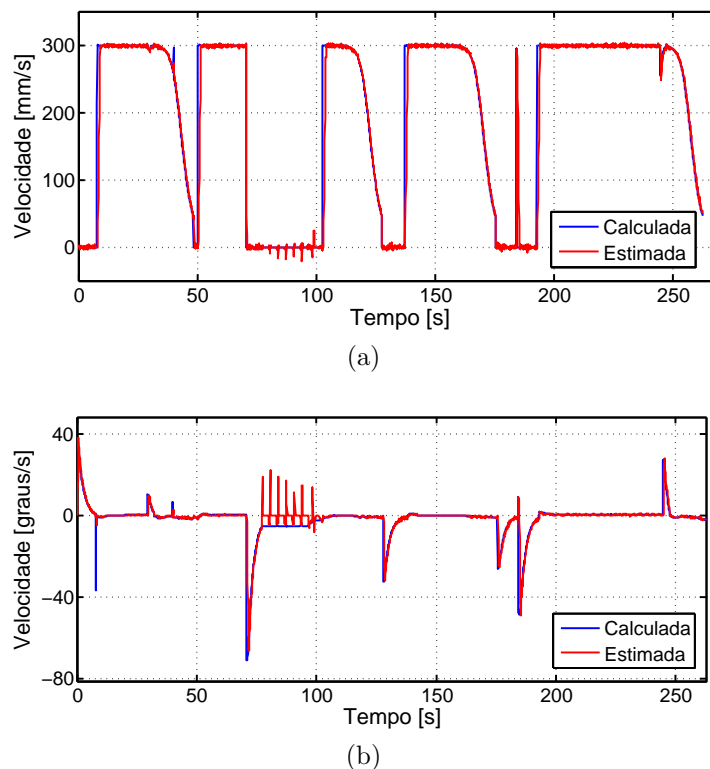


Figura 57 – Velocidades linear (a) e angular (b) do robô terrestre no experimento 2.

Observa-se, pelos resultados deste experimento, que o robô móvel foi capaz de calcular uma rota, e posteriormente redefini-la devido a um obstáculo intransponível identificado no caminho. Ambas rotas foram percorridas pelo robô que, durante o percurso, identificou as marcas conhecidas e realizou a correção da sua posição quando possível.

5.3 Experimento 3: Cooperação com o Quadrimotor no modo manual

Neste experimento o robô terrestre e o robô aéreo navegam em cooperação, e o principal objetivo é alcançar o ponto de destino com ambos robôs, efetuando um recálculo de rota devido a um obstáculo identificado pelo veículo aéreo. Durante todo percurso o quadrimotor é controlado por um ser humano, que o guia seguindo as instruções enviadas pelo programa embarcado no Pioneer. Para a realização deste experimento foi construído o mapa da Figura 58 em uma quadra poliesportiva. Com o robô posicionado na marca 1 foi indicado pelo usuário o ponto 7 como destino final.

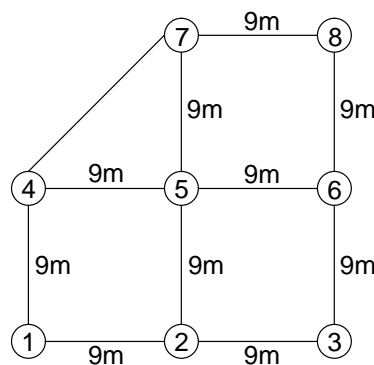


Figura 58 – Mapa utilizado nos experimentos 3 e 4.

Neste experimento foi utilizado um obstáculo, na cor azul, posicionado entre as *landmarks* 4 e 7, conforme a Figura 59.



Figura 59 – Obstáculo utilizado no experimento 3.

Na Figura 60 pode ser vista a interface gráfica após o programa ter calculado o caminho, na cor verde, que ambos veículos devem seguir para atingirem a marca desejada. Na mesma figura, na imagem da câmera da direita (pertencente ao quadrimotor), é observado o procedimento de calibração da cor que o robô aéreo utilizará para detectar o obstáculo terrestre.

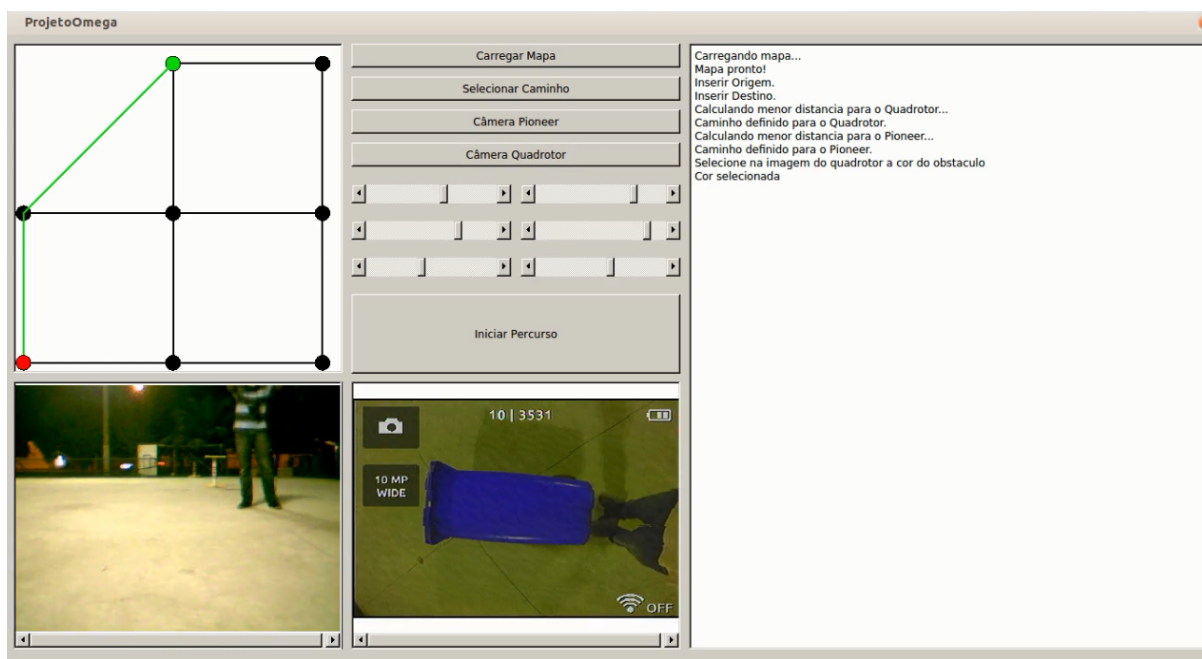


Figura 60 – Caminho calculado no experimento 3.

O experimento iniciou-se quando o robô aéreo estava sobre a *landmark* 1, para que ambos saíssem ao mesmo tempo em direção ao objetivo. Durante o percurso, enquanto o robô terrestre ainda estava navegando da marca 1 para a 4, o quadrimotor identificou o obstáculo azul entre as marcas 4 e 7, conforme a Figura 61. Na mesma figura ainda é possível observar a nova rota calculada, na cor azul, para o veículo aéreo. Essa nova rota é informada ao piloto, que abandona o caminho anterior para seguir o novo percurso. O Pioneer, nesse momento ainda não recalcula o seu caminho, continuando o percurso até a próxima *landmark*.

Após alcançar a próxima marca na sua rota, a marca 4, o robô terrestre recalcula a sua rota, conforme a Figura 62.

A Figura 63 apresenta a interface do programa após a realização do percurso. Nesta imagem é possível observar o caminho percorrido pelo robô, em vermelho, assim como as mensagens de detecção das *landmarks* encontradas, da detecção do obstáculo, e do recálculo de rota tanto para o veículo aéreo quanto para o veículo terrestre.

A odometria de ambos robôs pode ser vista na Figura 64. Em verde estão destacados os momentos onde as medidas do sensor laser de varredura foram utilizadas para corrigir o erro de posição do veículo terrestre. O obstáculo azul desenhado mostra o ponto onde o robô aéreo identificou um obstáculo intransponível para o robô terrestre (conforme definido em 3.3.2), realizando assim o recálculo de rota.

As Figuras 65 (a) e 65 (b) apresentam, respectivamente, as velocidades linear e angular desenvolvidas pelo robô móvel terrestre durante o percurso. Próximo aos instantes de tempo 50s, 125s e 190s, no gráfico de velocidade linear (Figura 65 (a)), é possível

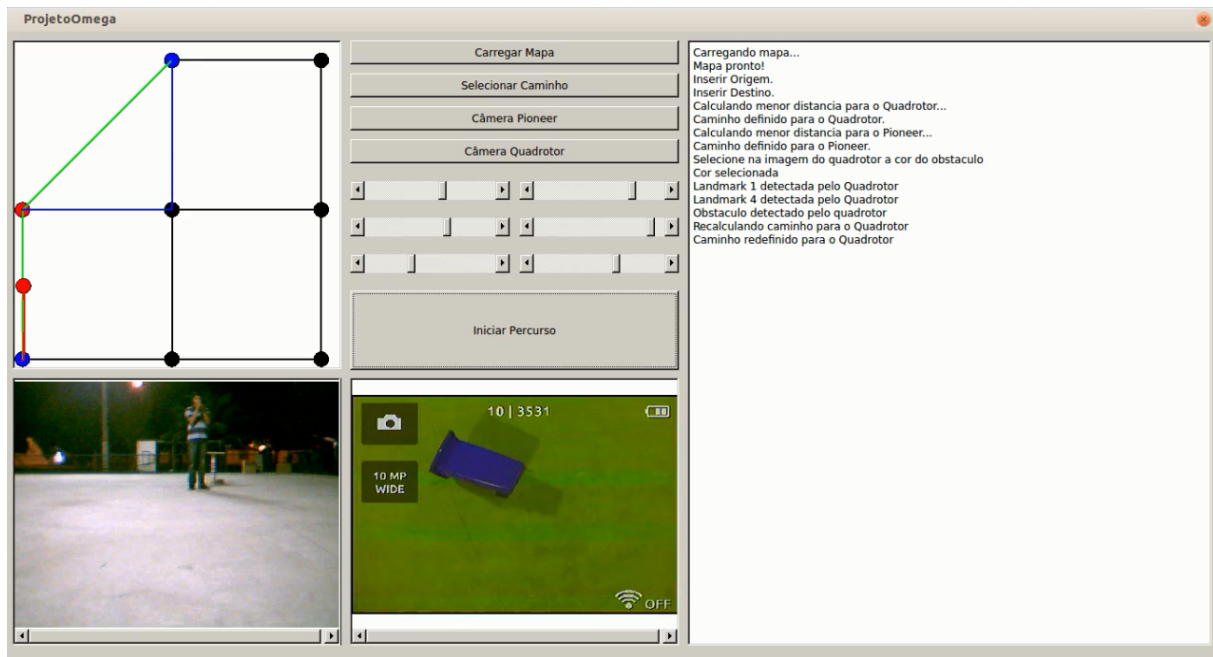


Figura 61 – Caminho recalculado para o quadrimotor no experimento 3.

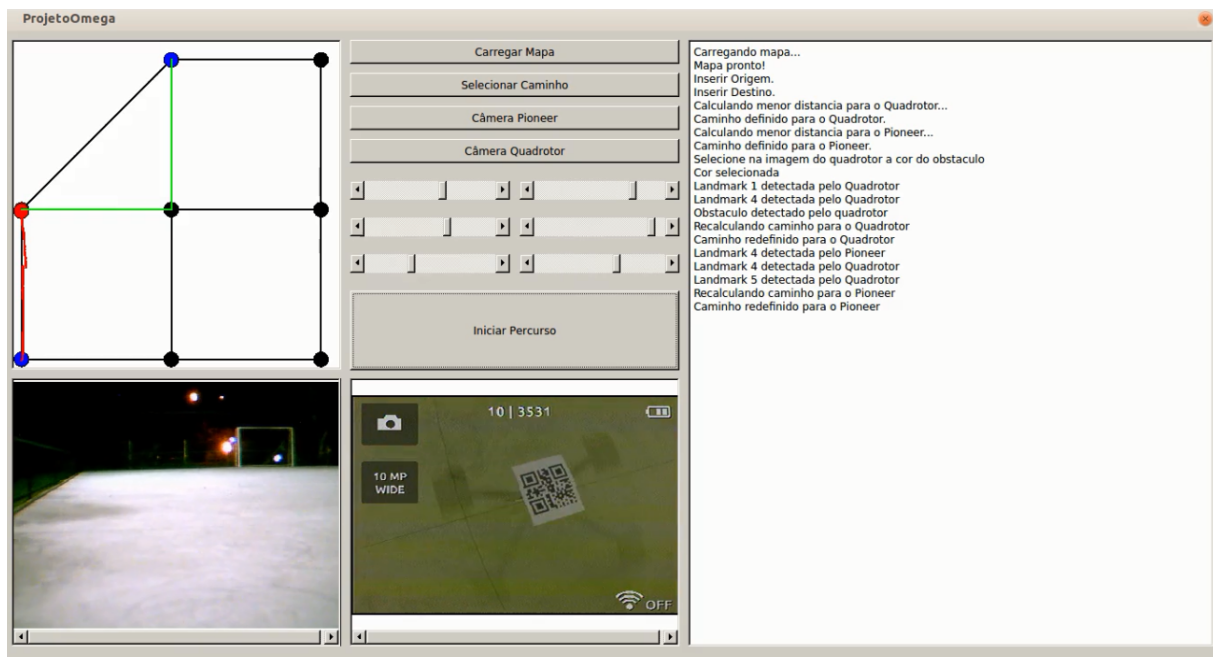


Figura 62 – Caminho recalculado para o Pioneer no experimento 3.

observar uma desaceleração do robô, que são os momentos onde o veículo se aproxima das *landmarks*. Ainda no mesmo gráfico, observa-se uma parada repentina do robô, próximo ao instante de tempo 35s, representando o momento em que o obstáculo é detectado pelo robô aéreo. O comando de parada sempre é realizado, pois em situações onde o obstáculo detectado pelo robô aéreo encontra-se entre o veículo terrestre e a próxima *landmark* o algoritmo solicita o retorno do robô para a marca anterior para que uma nova rota seja determinada. Neste experimento, como a obstrução encontrada não estava entre os nós em

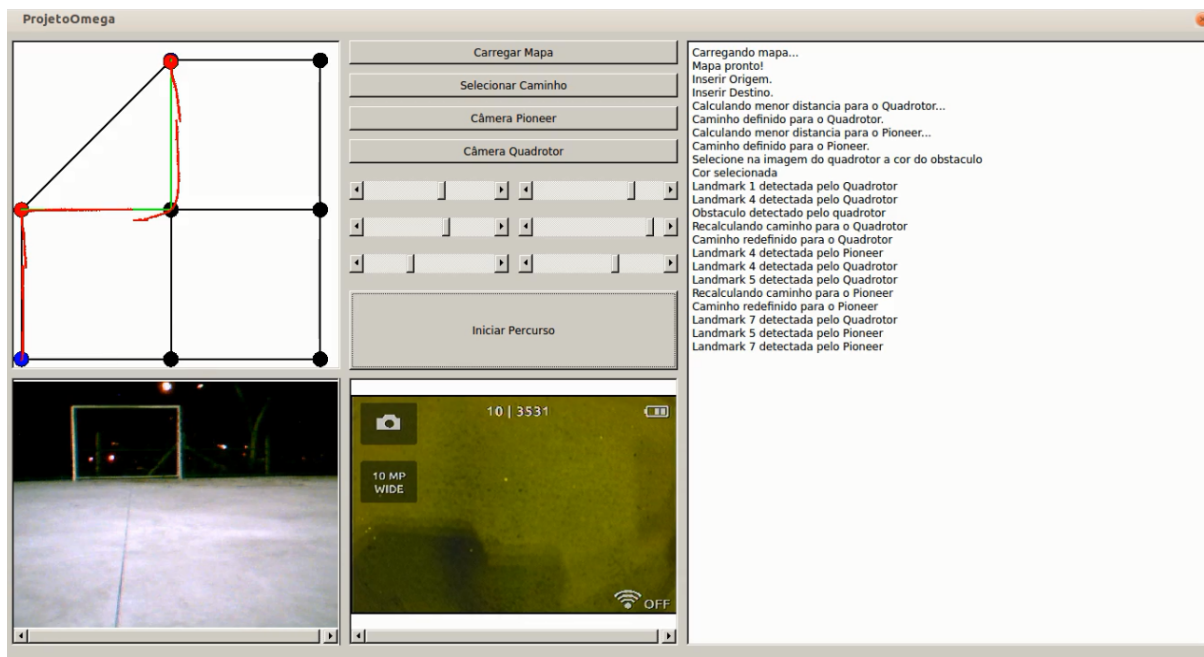


Figura 63 – Caminho percorrido no experimento 3.

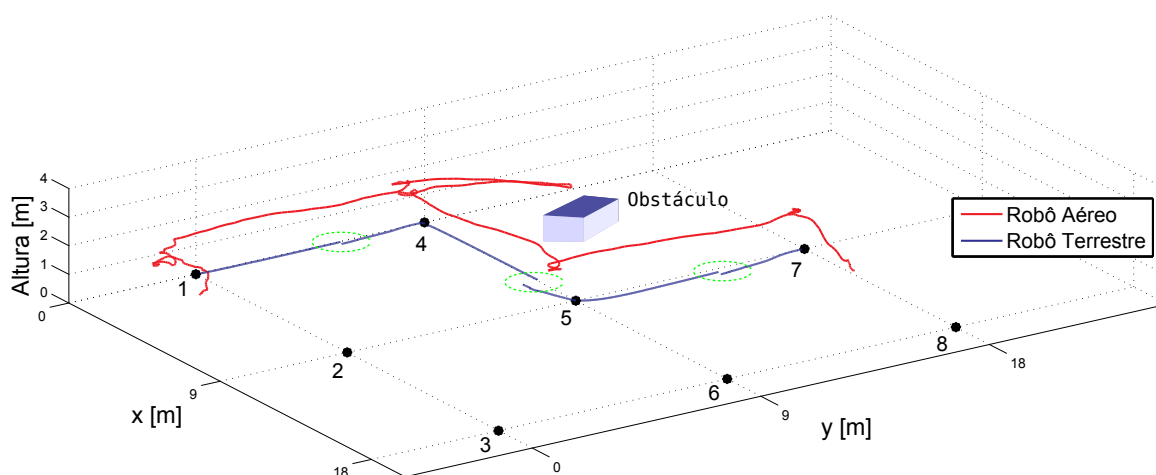


Figura 64 – Odometria de ambos robôs no experimento 3.

que o robô terrestre navegava, o mesmo continuou o percurso até a próxima marca. No gráfico de velocidade angular (Figura 65 (b)) podem-se destacar os pontos próximos aos instantes de tempo de 55s, 120s e 190s, os quais se referem à correção da posição dada pela odometria do robô devido às medidas obtidas pelo sensor laser de varredura. Ainda na Figura 65 (b), os pontos próximos aos instantes de tempo 70s e 140s correspondem, respectivamente, aos giros de 90° nas marcas 4 e 5 para alinhar o robô em direção aos próximos destinos.

Observa-se, pelos resultados deste experimento, que o algoritmo desenvolvido calculou uma rota tanto para o robô móvel terrestre quanto para o aéreo, e posteriormente as redefiniu devido a um obstáculo terrestre intransponível identificado pelo quadrimotor.

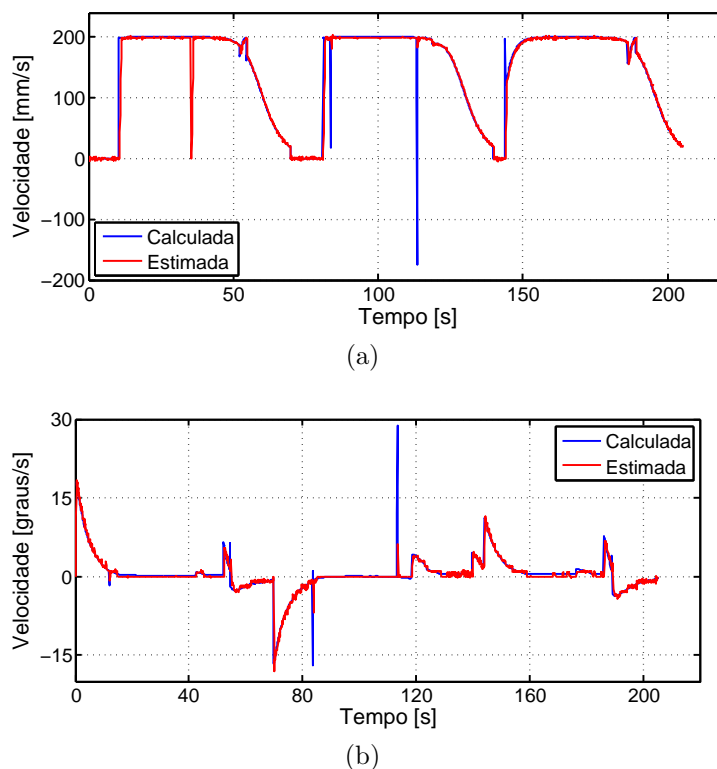


Figura 65 – Velocidades linear (a) e angular (b) do robô terrestre no experimento 3.

Ambas rotas foram percorridas pelos robôs que, durante o percurso, identificaram as marcas conhecidas e, além disso, o Pioneer realizou a correção da sua posição sempre que possível.

5.4 Experimento 4: Cooperação com o Quadrimotor no modo automático

Neste experimento o robô terrestre e o robô aéreo navegam em cooperação, e o principal objetivo é alcançar o ponto de destino com ambos robôs, efetuando o recálculo da rota do Pioneer devido a um obstáculo identificado pelo veículo aéreo. Durante todo percurso o quadrimotor está no modo automático, sendo controlado por um ser humano apenas para levantar voo e para aterrizar. Para a realização deste experimento foi construído o mesmo mapa da Seção 5.3 em uma quadra poliesportiva.

Tanto o obstáculo quanto a posição inicial dos robôs e a rotas calculadas são as mesmas das Figuras 59 e 60, respectivamente, da Seção 5.3.

Antes do início do experimento, foram capturadas as coordenadas geográficas, utilizando o GPS do veículo aéreo, referentes aos pontos correspondentes a cada marca do mapa. A rota que o robô aéreo segue foi inserida manualmente em seu piloto automático, após ter sido calculada e informada pelo algoritmo embarcado no computador do robô

terrestre. Dessa forma, nesse experimento, mesmo identificando o obstáculo, o robô aéreo segue a sua rota inicial planejada até o destino final, pois ainda não foi possível implementar uma alteração dinâmica no controle automático do quadrimotor.

O experimento iniciou-se somente quando o robô aéreo estava sobre a *landmark* 1, para que ambos saíssem ao mesmo tempo em direção ao objetivo. Durante o percurso, enquanto o robô terrestre ainda estava navegando da marca 1 para a 4, o quadrimotor identificou o obstáculo azul entre as marcas 4 e 7, conforme a Figura 66. Na mesma figura ainda é possível observar que não foi recalculada uma nova rota para o robô aéreo (a mensagem publicada informando esse recálculo será ignorada), visto que não era possível alterar a rota pré-programada em seu piloto automático.

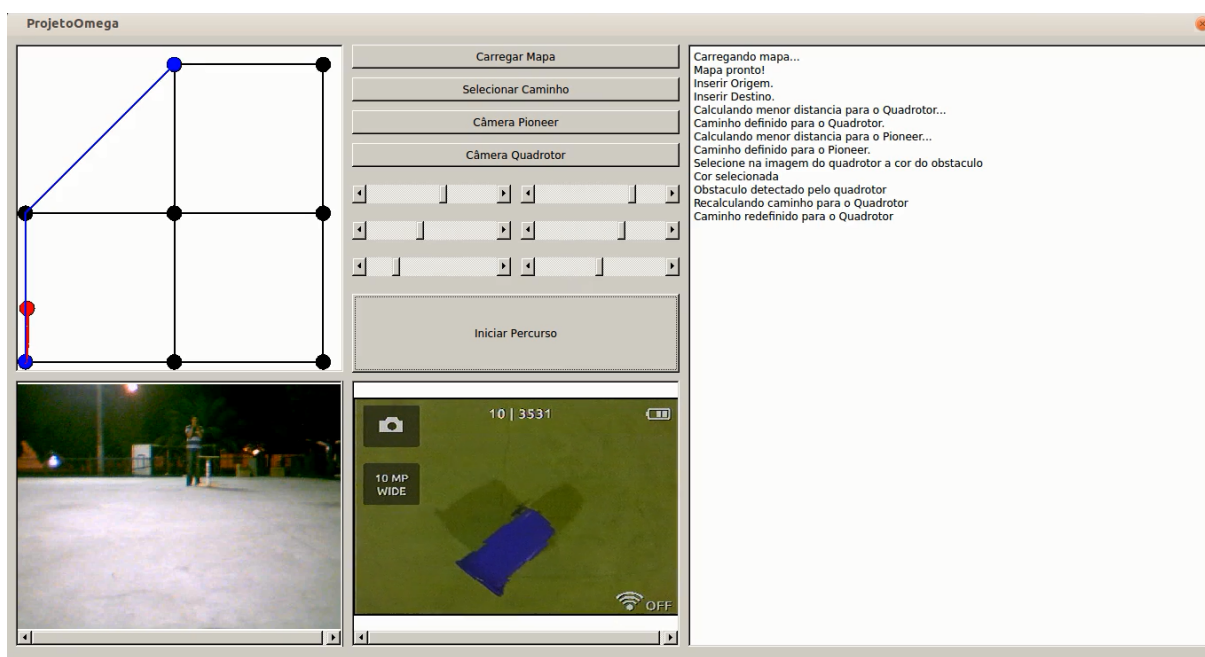


Figura 66 – Detecção do obstáculo pelo robô aéreo.

Após alcançar a próxima marca na sua rota, a marca 4, o robô terrestre recalcula a sua rota, conforme a Figura 67.

A Figura 68 apresenta a interface do programa após a realização do percurso. Nesta imagem é possível observar o caminho percorrido pelo robô, em vermelho, assim como as mensagens de detecção das *landmarks* encontradas, da detecção do obstáculo, e do recálculo de rota para o veículo terrestre.

A odometria de ambos robôs pode ser vista na Figura 69. Em laranja estão destacados os momentos onde as medidas do laser foram utilizadas para corrigir o erro de posição do veículo terrestre. O tracejado na cor preta destaca os pontos do GPS utilizados para traçar a rota do robô aéreo. O obstáculo azul desenhado mostra o ponto onde o robô aéreo identificou um obstáculo intransponível para o robô terrestre, realizando assim o recálculo de rota. No seu último ponto, WP 7 (*Way Point* 7), o quadrimotor iniciou o

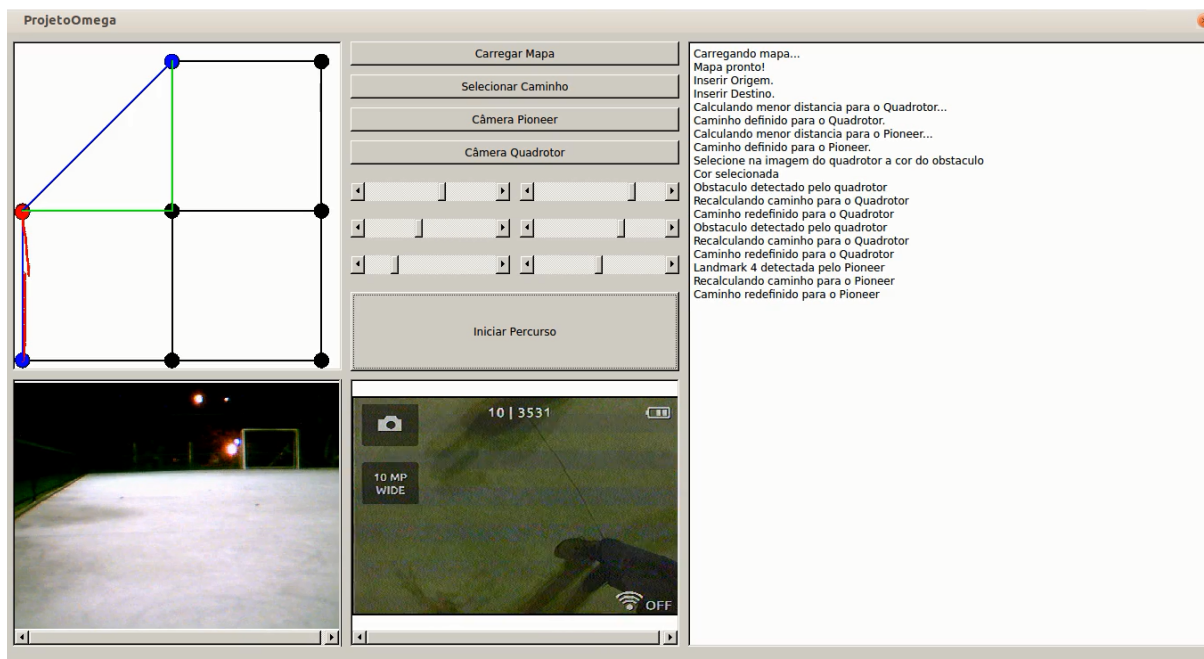


Figura 67 – Caminho recalculado para o Pioneer no experimento 4.

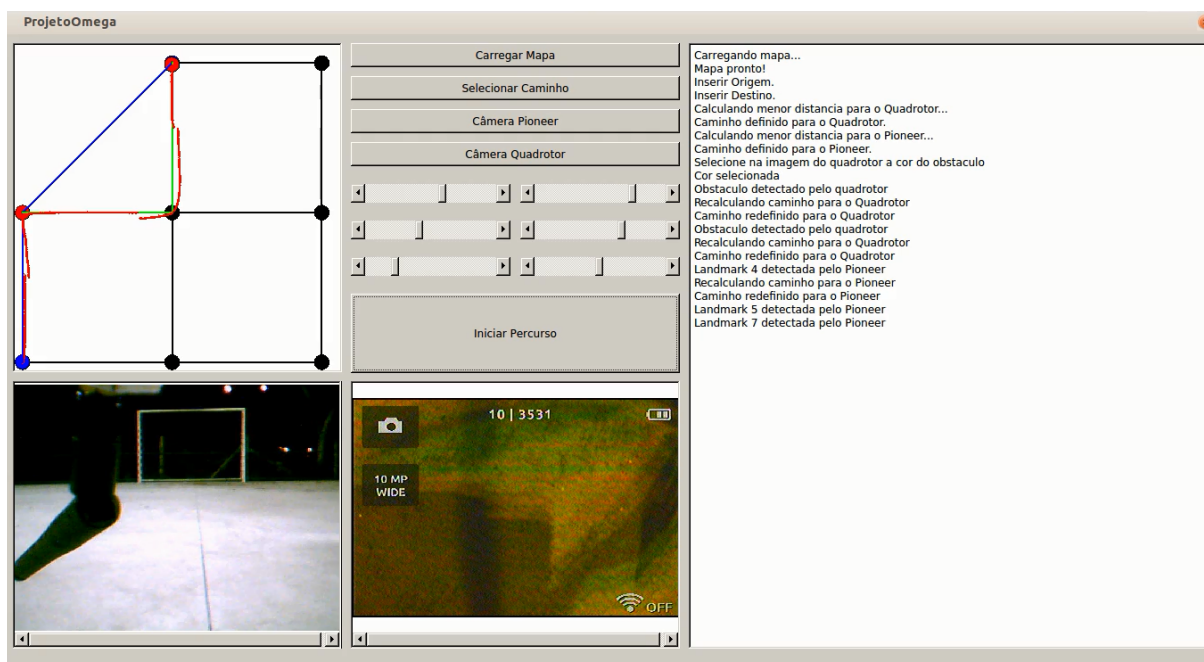


Figura 68 – Caminho percorrido no experimento 4.

procedimento de voo em círculo, até o piloto assumir o controle e iniciar o procedimento de aterrissagem.

As Figuras 70 (a) e 70 (b) apresentam, respectivamente, as velocidades linear e angular desenvolvidas pelo robô móvel terrestre durante o percurso. Próximo aos instantes de tempo 50s, 125s e 190s, no gráfico de velocidade linear (Figura 70 (a)), é possível observar uma desaceleração do robô, que são os momentos onde o veículo se aproxima das *landmarks*. Ainda no mesmo gráfico, observa-se uma parada repentina do robô, próximo

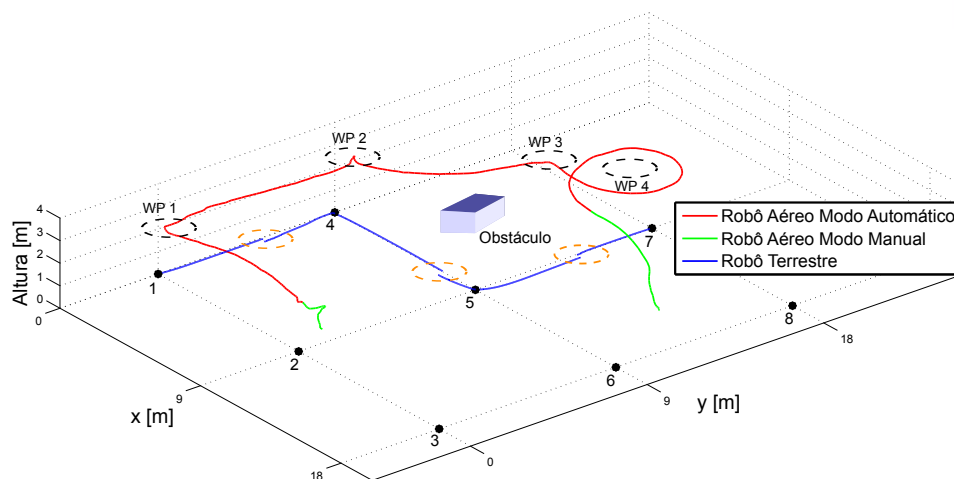
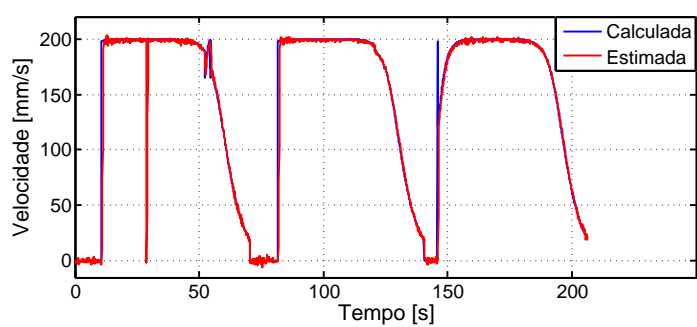


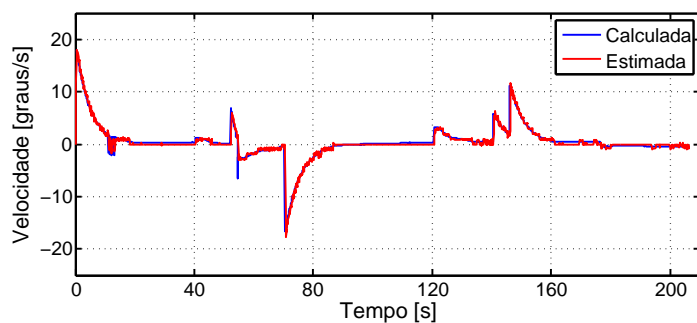
Figura 69 – Odometria de ambos robôs no experimento 4.

ao instante de tempo 30s, representando o momento em que o obstáculo é detectado pelo robô aéreo. Nesse caso, como a obstrução encontrada não estava entre os nós em que o robô terrestre navegava, o mesmo continuou o percurso até a próxima marca. No gráfico de velocidade angular (Figura 70 (b)) podem-se destacar os pontos próximos aos instantes de tempo de 50s, 120s e 180s, os quais se referem à correção da posição dada pela odometria do robô devido às medidas obtidas pelo sensor laser de varredura. Ainda na Figura 65 (b), os pontos próximos aos instantes de tempo 70s e 140s correspondem, respectivamente, aos giros de 90° nas marca 4 e 5 para alinhar o robô em direção aos próximos destinos.

Observa-se, pelos resultados deste experimento, que o algoritmo desenvolvido calculou uma rota tanto para o robô móvel terrestre quanto para o aéreo e, posteriormente, redefiniu a rota do Pioneer devido a um obstáculo terrestre intransponível identificado pelo quadrimotor. Ambas rotas foram percorridas pelos robôs que, durante o percurso, identificaram as marcas conhecidas e, além disso, o Pioneer realizou a correção da sua posição sempre que possível.



(a)



(b)

Figura 70 – Velocidades linear (a) e angular (b) do robô terrestre no experimento 4.

CAPÍTULO 6

CONSIDERAÇÕES FINAIS

Nessa Dissertação de Mestrado foi abordado o tema de cooperado entre um robô aéreo e um robô terrestre. Para realizar essa cooperação, foram propostos métodos para calcular e recalculas rotas de navegação, identificar e desviar de obstáculos (por imagens e medidas de um sensor laser de varredura), identificar *landmarks* conhecidas, corrigir posição e controlar um robô terrestre, além da construção de um quadrimotor.

Em relação aos algoritmos de controle, cálculo e recálculo de rotas, foi possível observar, com os resultados das simulações realizadas, o funcionamento adequado do programa desenvolvido. Foi possível simular para diferentes mapas e tipos de obstáculos que o robô era capaz de, partindo de um ponto inicial conhecido, alcançar o destino desejado.

Os algoritmos de identificação de *landmarks* e obstáculos foram testados experimentalmente antes da sua utilização na aplicação de cooperação. Os resultados apresentados mostram que foi possível realizar a detecção dessas marcas, e utilizar as informações capturadas pelo sensor laser de varredura para melhorar o posicionamento do robô.

Foram realizados voos experimentais com o quadrimotor desenvolvido, tanto no modo manual quanto no modo autônomo, com resultados satisfatórios. Foi possível verificar o correto funcionamento do seu controlador de voo, e capturar todos os dados de odometria da sua navegação (pontos de gps, valores dos acelerômetros, giroscópios, sensor ultrassônico, entre outros).

Os resultados obtidos nos experimentos realizados em um ambiente externo comprovam a cooperação proposta. Um robô terrestre e um robô aéreo foram capazes de realizar uma navegação por um mapa conhecido, com o robô terrestre desviando de obstáculos e tendo a sua rota recalculada quando um obstáculo intransponível era previamente identificado pelo veículo aéreo.

6.1 Trabalhos Futuros

Para aumentar a confiabilidade e automatização da cooperação proposta, ainda são necessárias a implementação de algumas técnicas.

No trabalho apresentado as *landmarks* seguidas pelo robô aéreo são geradas pelo algoritmo de Dijkstra programado no computador do Pioneer. Porém, essa rota deve ser gravada manualmente no *firmware* do controlador de voo. Como existe uma comunicação sem fio entre o controlador de voo e o computador do Pioneer (*link* de odometria de voo) o próximo passo é estudar o protocolo de comunicação utilizado nessa comunicação, o *mavlink*, para possibilitar a gravação automática (e regravação, quando a rota for alterada) no quadrimotor. O recálculo da rota para o quadrimotor permite que este sempre percorra o mesmo caminho que o robô terrestre, podendo assim encontrar obstáculos em novos trechos e manter a cooperação com o outro robô até o final do percurso.

Outra melhoria proposta diz respeito à odometria do robô. Como foi observado durante os experimentos, para testes em ambientes externos onde o robô deve navegar por caminhos extensos, realizando por diversas vezes mudança de orientação, os erros de odometria acumulados impossibilitam uma localização adequada do veículo no ambiente. Apesar de se utilizarem as marcas conhecidas para diminuir esses erros, existem casos (quando o robô não consegue localizar a *landmark*, por exemplo) em que a correção não é suficiente. Para diminuir esses problemas, propõe-se utilizar um GPS acoplado ao robô terrestre, e a implementação de um filtro de Kalman para integrar as informações de todos os sensores, melhorando a localização do veículo no ambiente externo.

Uma possibilidade de aplicação do modelo construído nesta Dissertação é na captura de imagens em ambientes rurais, a fim de realizar mapeamento, identificação de plantações ilegais, ataque de pragas a plantações, focos de incêndio, ou mesmo auxílio à defesa civil no caso de desastres naturais, alertando equipes em terra, compostas por seres humanos ou robôs terrestres.

REFERÊNCIAS

- ARTIEDA, J. et al. Visual 3-d slam from uavs. *J. Intell. Robotics Syst.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 55, n. 4-5, p. 299–321, ago. 2009. ISSN 0921-0296.
- BAI, H.; WEN, J. T. Cooperative load transport: A formation-control perspective. *IEEE Transactions on Robotics*, v. 26, n. 4, p. 742–750, 2010.
- BELL, J. E.; MCMULLEN, P. R. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, v. 18, n. 1, p. 41 – 48, 2004.
- BHATTACHARYA, P.; GAVRILOVA, M. Voronoi diagram in optimal path planning. In: *Voronoi Diagrams in Science and Engineering, 2007. ISVD '07. 4th International Symposium on*. [S.l.: s.n.], 2007. p. 38–47.
- BIN, N. et al. Recurrent neural network for robot path planning. In: LIEW, K.-M. et al. (Ed.). *PDCAT*. [S.l.]: Springer, 2004. (Lecture Notes in Computer Science, v. 3320), p. 188–191. ISBN 3-540-24013-6.
- BRADSKI, G. *Dr. Dobb's Journal of Software Tools*, 2000.
- BRAND, M. et al. Ant colony optimization algorithm for robot path planning. In: *Computer Design and Applications (ICDDA), 2010 International Conference on*. [S.l.: s.n.], 2010. v. 3, p. V3–436–V3–440.
- BRANDÃO, A. S. *Projeto de Controladores Não Linear para Voo Autônomo de Veículos Aéreos de Pás Rotativas*. Tese (Doutorado) — Universidade Federal do Espírito Santo, UFES, Brasil., 2013.
- BRANDÃO, A. S. et al. PVTOL maneuvers guided by a high-level nonlinear controller applied to a rotorcraft machine. *European Journal of Control*, v. 20, n. 4, p. 172 – 179, 2014. ISSN 0947-3580. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0947358014000326>>.
- BRYSON, M.; SUKKARIEH, S. Cooperative localisation and mapping for multiple uavs in unknown environments. In: *Aerospace Conference, 2007 IEEE*. [S.l.: s.n.], 2007. p. 1–12.
- BURGARD, W. et al. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, v. 21, n. 3, p. 376–378, 2005.

- DESAI, J. P.; OSTROWSKI, J. P.; KUMAR, V. Modeling and control of formations of nonholonomic mobile robots. *IEEE T. Robotics and Automation*, v. 17, n. 6, p. 905–908, 2001.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. *NUMERISCHE MATHEMATIK*, v. 1, n. 1, p. 269–271, 1959.
- DIOS, J. M.-D.; OLLERO, A. An illumination-robust robot infrared vision system for robotics outdoor applications. In: *Automation Congress, 2004. Proceedings. World*. [S.l.: s.n.], 2004. v. 15, p. 413–418.
- EISENBEISS, H. A mini unmanned aerial vehicle (uav): system overview and image acquisition. In: *Image Acquisition International Workshop on Processing and Visualization using High-Resolution Imagery*. [S.l.: s.n.], 2004.
- FERREIRA, A. et al. An approach to avoid obstacles in mobile robot navigation: the tangential escape. *SBA: Controle & Automação*, scielo, v. 19, p. 395 – 405, 12 2008. ISSN 0103-1759.
- GAO, X.-G.; FU, X.-W.; CHEN, D.-Q. A genetic-algorithm-based approach to uav path planning problem. In: *Proceedings of the 5th WSEAS International Conference on Simulation, Modelling and Optimization*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2005. (SMO'05), p. 523–527. ISBN 960-8457-32-7.
- GARRIDO, S. et al. Path planning for mobile robot navigation using voronoi diagram and fast marching. In: *IROS*. [S.l.: s.n.], 2006. p. 2376–2381.
- GARZÓN, M.; DAVIDZAPATA, J. ao V.; BARRIENTOS, A. An aerial-ground robotic system for navigation and obstacle mapping in large outdoor areas. *Sensors*, v. 13, n. 1, p. 1247–1267, 2013. ISSN 1424-8220.
- GIGRAS, Y.; GUPTA, K. Artificial intelligence in robot path planning. *International Journal of Soft Computing and Engineering*, v. 2, p. 471–474, 2012.
- HENG, L. et al. Real-time photo-realistic 3d mapping for micro aerial vehicles. In: *IROS*. [S.l.]: IEEE, 2011. p. 4012–4019. ISBN 978-1-61284-454-1.
- HOW, J. P. et al. Increasing autonomy of uavs. *Robotics and Automation Magazine, IEEE*, v. 16, n. 2, p. 43–51, June 2009.
- KIM, B. et al. Multiple relative pose graphs for robust cooperative mapping. In: *ICRA*. [S.l.]: IEEE, 2010. p. 3185–3192.
- LONGO, D. et al. A mixed terrestrial aerial robotic platform for volcanic and industrial surveillance. In: *2007 IEEE International Workshop on Safety, Security*. [S.l.: s.n.], 2007.
- LUMELSKY, V. J.; STEPANOV, A. Dynamic path planning for a mobile automaton with limited information on the environment. *Automatic Control, IEEE Transactions on*, v. 31, n. 11, p. 1058–1063, Nov 1986. ISSN 0018-9286.

- MACARTHUR, E. Z.; MACARTHUR, D.; CRANE, C. Use of cooperative unmanned air and ground vehicles for detection and disposal of mines. In: B. Gopalakrishnan (Ed.). *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. [S.l.: s.n.], 2005. (Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, v. 5999), p. 94–101.
- MARJOVI, A. et al. Multi-robot exploration and fire searching. In: *IROS*. [S.l.]: IEEE, 2009. p. 1929–1934.
- MELLINGER, D. et al. Cooperative grasping and transport using multiple quadrotors. In: MARTINOLI, A. et al. (Ed.). *DARS*. [S.l.]: Springer, 2010. (Springer Tracts in Advanced Robotics, v. 83), p. 545–558. ISBN 978-3-642-32722-3.
- MICHAEL, N. et al. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *J. Field Robotics*, v. 29, n. 5, p. 832–841, 2012.
- MOHAMMADI, S.; HAZAR, N. A voronoi-based reactive approach for mobile robot navigation. In: SARBAZI-AZAD, H. et al. (Ed.). *Advances in Computer Science and Engineering*. [S.l.]: Springer Berlin Heidelberg, 2009, (Communications in Computer and Information Science, v. 6). p. 901–904. ISBN 978-3-540-89984-6.
- OLLERO, A. et al. Multiple eyes in the skies: architecture and perception issues in the comets unmanned air vehicles project. *IEEE Robot. Automat. Mag.*, v. 12, n. 2, p. 46–57, 2005.
- OLLERO, A.; MERINO, L. Control and perception techniques for aerial robotics. *Annual Reviews in Control*, Elsevier, v. 28, n. 2, p. 167–178, 2004.
- PARKER, L. E. Current state of the art in distributed robot systems. In: PARKER, L. E.; BEKEY, G.; BARHEN, J. (Ed.). *Distributed Autonomous Robotic Systems 4*. [S.l.]: Springer, 2000. v. 2606/2003, p. 3–12.
- PEREIRA, F. G. *Navegação e Desvio de Obstáculos usando um Robô Móvel Dotado de Sensor de Varredura Laser*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo - UFES, 2006.
- RAGHUWAIYA, K. S.; SINGH, S.; VANUALAILAI, J. Formation control of mobile robots. *World Academy of Science, Engineering and Technology*, v. 5, n. 12, p. 1689 – 1694, 2011. ISSN 1307-6892.
- RATHINAM, S. et al. Vision based following of locally linear structures using an unmanned aerial vehicle. In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*. [S.l.: s.n.], 2005. p. 6085 – 6090.
- RESENDE, C. Z.; CARELLI, R.; SARCINELLI-FILHO, M. Coordinated path-following for multi-robot systems using the cluster space framework approach. In: *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*. [S.l.: s.n.], 2014. p. 332–337.
- SANTANA, L. V. *Controle de Altitude e Ângulo de Guinada de um Mini Helicóptero*. Dissertação (Dissertação de Mestrado) — Universidade Federal do Espírito Santo, Março 2011.

- SARIFF, N. B.; BUNIYAMIN, N. Genetic algorithm versus ant colony optimization algorithm - comparison of performances in robot path planning application. In: FILIPE, J.; ANDRADE-CETTO, J.; FERRIER, J.-L. (Ed.). *ICINCO (1)*. [S.l.]: INSTICC Press, 2010. p. 125–132.
- SECCHI, H. A. *Control de Vehículos Auto guiados con Realimentación Sensorial*. Dissertação (Mestrado) — Facultad de Ingeniería de la Universidad Nacional de San Juan, San Juan - Argentina, 1998.
- SHIM, D. H. et al. Autonomous exploration in unknown urban environments for unmanned aerial vehicles. In: *in Proc. AIAA GN&C Conference*. [S.l.: s.n.], 2005.
- TOURNIER, G. P. et al. Estimation and control of a quadrotor vehicle using monocular vision and moiré patterns. In: *In AIAA Guidance, Navigation and Control Conference*. [S.l.]: AIAA, 2006. p. 2006–6711.
- TURPIN, M.; MICHAEL, N.; KUMAR, V. Trajectory design and control for aggressive formation flight with quadrotors. *Auton. Robots*, v. 33, n. 1-2, p. 143–156, 2012.
- YAMASHITA, A. et al. Motion planning for cooperative transportation of a large object by multiple mobile robots in a 3d environment. In: *ICRA*. [S.l.]: IEEE, 2000. p. 3144–3151. ISBN 0-7803-5889-9.
- YU, H. et al. Probabilistic path planning for cooperative target tracking using aerial and ground vehicles. In: *American Control Conference (ACC)*. San Francisco, CA: [s.n.], 2001. p. 4673 – 4678.
- ZHANG, T. et al. Autonomous hovering of a vision/imu guided quadrotor. In: *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*. [S.l.: s.n.], 2009. p. 2870 –2875.