

**UM ARCABOUÇO PARA PLANEJAMENTO E
CONTROLE DESCENTRALIZADO DE
ROBÔS HETEROGÊNEOS**

ARMANDO ALVES NETO

**UM ARCABOUÇO PARA PLANEJAMENTO E
CONTROLE DESCENTRALIZADO DE
ROBÔS HETEROGÊNEOS**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

ORIENTADOR: MARIO FERNANDO MONTENEGRO CAMPOS

Belo Horizonte

Agosto de 2012

© 2012, Armando Alves Neto.
Todos os direitos reservados.

Ficha catalográfica elaborada pela Biblioteca do ICEx – UFMG

Alves Neto, Armando.

A474a Um arcabouço para planejamento e controle descentralizado de robôs heterogêneos / Armando Alves Neto — Belo Horizonte, 2012.
xxvi, 105 f. : il. ; 29cm.

Tese (doutorado) — Universidade Federal de Minas Gerais.

Orientador: Mario Fernando Montenegro Campos.

1. Computação – Teses. 2. Robótica – Teses.
I. Orientador. II. Título.

519.6*82.9(043)



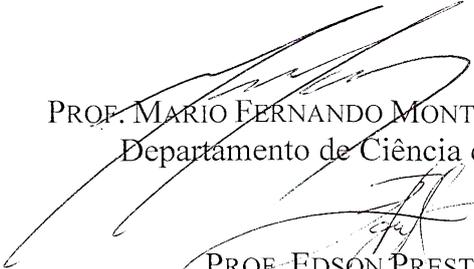
UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

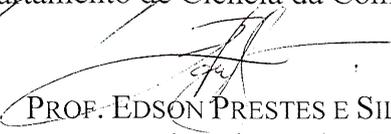
Um arcabouço para planejamento e controle descentralizado de robôs heterogêneos

ARMANDO ALVES NETO

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:



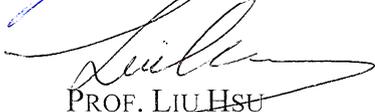
PROF. MARIO FERNANDO MONTENEGRO CAMPOS - Orientador
Departamento de Ciência da Computação - UFMG



PROF. EDSON PRESTES E SILVA JÚNIOR
Departamento de Informática Teórica - UFRGS



PROF. GUILHERME AUGUSTO SILVA PEREIRA
DEE - UFMG



PROF. LIU HSU
Departamento de Engenharia Elétrica – COPPE/UFRJ



PROF. LUIZ CHAIMOWICZ
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 29 de agosto de 2012.

*Para o meu pai, Celso, que nunca desistiu.
Nem mesmo no fim...*

Agradecimentos

Em primeiro lugar, gostaria de agradecer às pessoas mais importantes da minha vida: meu pai, Celso, minha mãe, Luci, e meu irmão Celsinho. Mas especialmente também quero agradecer à minha esposa, Grazielli, pela atenção e paciência todos esses anos, e a seu pai, Antônio, mãe, Madalena e irmãos Kelley, Anderson e Ivan (adotado) que se tornaram minha segunda família. Obrigado também ao Toby, ao Lôro e ao Bolinha.

Em segundo lugar, agradeço ao Professor Mario Campos pela orientação não apenas neste trabalho, mas também ao longo de boa parte de minha jornada acadêmica e pessoal. O Mario foi mais do que um mestre, foi uma referência em vários âmbitos de minha vida, não apenas a profissional. Obrigado também aos membros da banca avaliadora, Professor Edson Prestes e Silva Júnior do DIT/UFRGS, Professor Guilherme Augusto Silva Pereira do DEE/UFMG, Professor Liu Hsu da COPPE/UFRJ e Professor Luiz Chaimowicz do DCC/UFMG, que tanto contribuíram para o desenvolvimento e conclusão deste trabalho. O Chaimo também foi um grande amigo e uma referência profissional a ser seguida.

Agradeço também aos colegas de trabalho do VeRLab (Laboratório de Visão Computacional e Robótica)/DCC, cujas interações foram fundamentais para minha formação. Em especial, agradeço ao Douglas, companheiro de várias publicações, e ao Erickson, companheiro de várias lamentações. Aos demais alunos de doutorado, entre eles o Paulo Drews, amigão do peito, e o Vilar, que tanta falta fez ao laboratório. Aos alunos de mestrado, entre eles o Cantoni, com quem aprendi muito. Vários outros também abrilhantaram o laboratório, como o Python, o Gabriel, o Samuel, a Beth, o Vinícius e muitos mais. Alunos de graduação também foram vários: Dimas, Vitor, Bruno, Rodrigo, Alysson, Mathias, James. Todos deixaram suas marcas no VeRLab. Por último, mas não menos importante, nosso técnico de laboratório, Wolmar, que organizou o ambiente de trabalho e auxiliou em vários de nossos trabalhos, especialmente este aqui. Obrigado também ao CNPq, CAPES, FAPEMIG e demais órgãos financiadores de pesquisa que deram suporte a este e outros traba-

lhos desenvolvidos ao longo desses quatro anos.

Agradeço ainda às varias pessoas que ajudaram a tornar mais agradável o penoso caminho ao longo do último quadriênio: aos doutores, Dr. Watson, Dr. Jones, Dr. Brown, Dr. Manhattan, Dr. Hannibal, Dr. Moreau, Dr. Evil, Dr. Jonathan Crane, Dr. Zaius, Dr. Banner; e aos não-doutores, Gandalf – o cinzento, Bilbo e Frodo Baggins, Ablon – o anjo renegado, Rick Deckard, Rorschach, Dombledore, V, Winston Smith, Morpheus, Batman, Coringa, Homer Simpson, Tyler Durden, John Connor, Seu Madruga e muitos outros. Agradecimentos especialíssimos ainda a Pedro Oliveira, Aline Rodrigues, Jorge Prudente, Roger Lambert, Dr. Paulista, Mayra e Aratama pela constante companhia e aventuras vividas nos últimos anos. Obrigado aos que compartilharam e contribuíram para tais aventuras, Grazielli, Rogério Canela, Madalena, Erickson, Leandro Soriano e Douglas, e a meus amigos e eternos colegas de RPG, Elvis Sorriso Suíno, Pitico e Rogério Canela.

Por fim, gostaria de agradecer imensamente aos médicos, psicólogos, enfermeiros e demais funcionários do Centro de Terapia Renal e do Hospital Evangélico de Belo Horizonte pela dedicação e assistência que oferecem não apenas aos pacientes, mas também aos familiares, nos momentos mais difíceis da vida.

A todos, muitíssimo obrigado...

*“Um homem pode morrer, lutar, falhar, até mesmo ser esquecido,
mas suas ideias podem modificar o mundo mesmo tendo se passado 400 anos.”*

(V de Vingança)

Resumo

Esta tese aborda o problema de navegação e controle descentralizado de times de robôs cooperativos em ambientes com obstáculos. Propõem-se aqui uma metodologia para a coordenação de agentes heterogêneos sujeitos a perturbações e incertezas, compartilhando ambientes complexos. O método apresentado baseia-se em uma abordagem de planejamento de caminhos e trajetórias com exploração aleatória de espaços, cujas vantagens de eficiência e generalização possibilitam a aplicação em tempo real a problemas multirrobôs. Nesse caso, exploraram-se duas particularidades da solução: i) o compartilhamento de uma semente pseudoaleatória entre agentes do time e ii) o uso de abstrações (funções paramétricas) para o planejamento coordenado. No primeiro aspecto, os robôs utilizam funções determinísticas que simulam características de aleatoriedade para gerar planos idênticos aos dos outros robôs, com base em metodologias de planejamento em tempo real. Isso permite manter as vantagens dessas técnicas em um problema com solução determinística. No segundo aspecto, os robôs utilizam funções paramétricas que permitem o planejamento do grupo de forma descentralizada, sem a necessidade de que cada agente esteja ciente da existência de outros. Isso favorece a escalabilidade do método, tornando-o inteiramente descentralizado. A metodologia proposta é aplicada a dois tipos de problemas complexos: i) o controle de formação de um grupo de robôs aéreos e ii) a cooperação entre veículos aéreos e terrestres. Utilizaram-se como base de predição, modelos que incorporam tanto o comportamento cinemático quanto dinâmico dos robôs, além das principais restrições no espaço de estados. Os controladores empregados nesta abordagem são bastante simples, porém a metodologia não se restringe a eles, abrangendo lógicas de controle bem mais sofisticadas.

Palavras-chave: Planejamento Multirrobôs, Controle Descentralizado, Árvores Aleatórias de Exploração Rápida, Cooperação Aéreo-terrestre.

Abstract

This work addresses the problem of decentralized navigation and control of teams of cooperative robots in cluttered environments. We propose here a methodology for the coordination of heterogeneous agents subject to disturbances and uncertainties, typical of complex environments. Our method is based on an approach of path and trajectory planning with random exploration of spaces, whose advantages of efficiency and generalization allow its application in real-time multirobot problems. We explore two main features of the approach: i) the sharing of a pseudo-random seed between teammates and ii) the usage of abstractions (parametric functions) for coordinated planning. At first, the robots use deterministic functions, that simulate randomized characteristics, to generate plans that are the same as of other robots on the basis of the real-time methodologies. This allows maintaining the advantages of these techniques in a problem with deterministic solution. Secondly, the robots use parametric functions that make it possible the planning of the team in a distributed manner, without requiring that each agent be aware of the existence of others. That contributes for the scalability of the method, making it decentralized. The proposed methodology is applied to two kinds of complex problems: i) formation control of a group of aerial robots and ii) air-ground cooperation of vehicles. We use as basis of prediction, models that incorporate both the dynamic and the kinematic behavior of the robots, as well as the main constraints in the state space. The controllers employed in this approach are simple, but the method is general enough to incorporate more sophisticated control laws.

Keywords: Multirobot Planning, Decentralized Control, Rapidly-exploring Random Trees, Air-ground Cooperation.

Lista de Figuras

1.1	Problema de cooperação multirrobo	3
2.1	Expansão da Árvore Aleatória de Exploração Rápida [LaValle & Kuffner, 1999]	10
2.2	Expansão da árvore de exploração (RRT) em malha fechada [Kuwata et al., 2008]	12
2.3	Planejamento centralizado, baseado em RRTs, para múltiplos UAVs de asa fixa [Alves Neto et al., 2010]	16
3.1	Espaços de representação	22
3.2	Modelo de predição em malha aberta para o Algoritmo 3.1	24
3.3	Expansão da árvore \mathcal{T}_x da RRT de Malha Aberta	26
3.4	Modelo de predição do Algoritmo da RRT de Malha Fechada	27
3.5	Laço de expansão das árvores \mathcal{T}_r e \mathcal{T}_x na RRT de Malha Fechada	30
3.6	Laço de execução da RRT de Malha Fechada	32
3.7	Arquitetura de controle e comunicação apresentada em [Belta & Kumar, 2004]	33
4.1	Arquitetura de controle e comunicação proposta	43
4.2	Laço de expansão das árvores \mathcal{T}_α , \mathcal{T}_x^1 (robô circular) e \mathcal{T}_x^2 (robô quadrado) na MRPT	46
4.3	Visão detalhada da arquitetura de controle proposta	50
4.4	Laço de execução da MRPT	51
5.1	Frame de referências do veículo quadricóptero	56
5.2	Aplicação da metodologia proposta ao planejamento e controle em tempo real de um time composto de seis MicroUAVs (figura 1)	62
5.3	Aplicação da metodologia proposta ao planejamento e controle em tempo real de um time composto de seis MicroUAVs (figura 2)	64

5.4	Aplicação da metodologia proposta ao planejamento e controle em tempo real de um time composto de seis MicroUAVs (figura 3)	65
5.5	Avaliação do erro de predição para o time de UAVs	66
5.6	Comparação entre experimentos com e sem aplicação da heurística	67
5.7	Aplicação da metodologia proposta ao planejamento e controle em tempo real do time de UAVs usando curvas de Bézier (figura 1)	70
5.8	Aplicação da metodologia proposta ao planejamento e controle em tempo real do time de UAVs usando curvas de Bézier (figura 2)	71
5.9	Comparação entre experimentos com e sem aplicação da heurística no uso de curvas de Bézier	72
5.10	Plataforma MultiUAVs do VeRLab	73
5.11	Experimento real com o time de quadrirotors do VeRLab	75
6.1	Referenciais do robô de Ackerman	80
6.2	Aplicação da metodologia proposta ao problema de cooperação aéreo-terrestre (figura 1)	85
6.3	Aplicação da metodologia proposta ao problema de cooperação aéreo-terrestre (figura 2)	86
6.4	Aplicação da metodologia proposta ao problema de cooperação aéreo-terrestre (figura 3)	87
6.5	Simulação de falha em um dos agentes terrestres (figura 1)	88
6.6	Simulação de falha em um dos agentes terrestres (figura 2)	89
6.7	Plataforma de robôs heterogêneos do VeRLab	90
6.8	Teste real com o time de veículos heterogêneos do VeRLab (figura 1)	92
6.9	Teste real com o time de veículos heterogêneos do VeRLab (figura 2)	93

Lista de Algoritmos

3.1	RRT de Malha Aberta	25
3.2	RRT de Malha Fechada (Laço de Expansão)	29
3.3	RRT de Malha Fechada (Laço de Execução)	31
4.1	Laço de Expansão das Árvores Multirroboês	44
4.2	Laço de Execução Multirroboês	47

Lista de Abreviações

<i>Asctec</i>	<i>Ascending Technologies</i>
CL-RRT	<i>Closed Loop RRT</i> (RRT de Malha Fechada)
DMA-RRT	<i>Decentralized Multi-Agent RRT</i> (RRT Multiagente Descentralizada)
DMPC	<i>Distributed MPC</i> (Controle por Modelo Preditivo Distribuído)
DP	<i>Dynamic Programming</i> (Programação Dinâmica)
ERRT	<i>Execution-extended RRT</i> (RRT de Execução Estendida)
ESC	<i>Electronic Speed Control</i> (Controlador Eletrônico de Velocidade)
FINEP	Agência Financiadora de Estudos e Projetos
GPS	<i>Global Positioning System</i> (Sistema de Posicionamento Global)
IMU	<i>Inertial Measurement System</i> (Unidade de Medição Inercial)
ISA	<i>International Standard Atmosphere</i> (Atmosfera Padrão Internacional)
MILP	<i>Mixed Integer Linear Program</i> (Programação Linear Inteira Mista)
MPC	<i>Model Predictive Control</i> (Controle por Modelo Preditivo)
MRPT	<i>Multiple Rapidly-exploring Pseudorandom Trees</i> (Árvore Pseudoaleatória de Exploração Rápida Multiagente)
MUAV	<i>Micro Unmanned Aerial Vehicle</i> (Micro Veículo Aéreo Não tripulado)
OL-RRT	<i>Open Loop RRT</i> (RRT de Malha Aberta)
PCS	<i>Polymorphic Control System</i> (Sistemas de Controle Polimórfico)
PD	Proporcional–Derivativo
PID	Proporcional–Integral–Derivativo
PRM	<i>Probabilistic Road Map</i> (Mapa de Rotas Probabilístico)
RG-RRT	<i>Reachability-Guided RRT</i> (RRT Guiada por Alcançabilidade)
ROS	<i>Robot Operating System</i>
RRT	<i>Rapidly-exploring Random Tree</i> (Árvore Aleatória de Exploração Rápida)
UAV	<i>Unmanned Aerial Vehicle</i> (Veículo Aéreo Não Tripulado)
UFAM	Universidade Federal do Amazonas
UFMG	Universidade Federal de Minas Gerais

UGV	<i>Unmanned Ground Vehicle</i> (Veículo Terrestre Não tripulado)
VeRLab	Laboratório de Visão Computacional e Robótica
VTOL	<i>Vertical Take-off and Landing</i> (Pouso e Decolagem Vertical)

Lista de Símbolos

α	Vetor de estados de uma abstração
α_{alvo}	Estado final para o time de robôs
α^*	Melhor caminho corrente dentro da árvore \mathcal{T}_α
ΔT	Intervalo de tempo do planejamento em tempo real
δ	Comando de esterçamento (direção) do veículo de <i>Ackerman</i>
$\kappa(\cdot)$	Função de controle não linear para um único robô
$\rho(\cdot)$	Métrica de distância entre vértices para a RRT simples
$\varrho(\cdot)$	Heurística de correção baseada na estimativa dos estados do time
τ	Intervalo de tempo do modelo de predição
$\phi(\cdot)$	Função que mapeia estados dos robôs do time em \mathcal{A}
Ψ	Vetor de orientações de um corpo rígido em $\text{SO}(3)$
Ω	Vetor de velocidades angulares de um corpo rígido
ω	Vetor de velocidades rotacionais dos motores do quadrotor
\mathcal{A}	Espaço de representação de uma abstração
\mathbf{a}	Vetor que define os parâmetros de uma abstração
$\{\mathcal{B}\}$	Referencial preso ao corpo do robô
$\text{dir}(\cdot)$	Função que define um vetor-direção a partir de uma orientação
$F(\cdot)$	Modelo dinâmico não linear do time \mathcal{R}
$f(\cdot)$	Modelo dinâmico não linear de um único robô
\mathcal{G}	Espaço de representação de uma abstração em $\text{SE}(n)$
\mathbf{g}	Vetor de configuração de uma abstração em $\text{SE}(n)$
\mathbf{g}_o	Vetor gravidade
$g(\cdot)$	Restrições de movimento entre agentes
$h(\cdot)$	Distância entre as referências multirrobôs e um ponto em \mathcal{W}
$K(\cdot)$	Função de controle não linear para o time de robôs
\mathbb{N}	Conjunto dos números naturais
n	Número de robôs no time

\mathcal{O}	Conjunto de obstáculos do ambiente
\mathcal{O}_i	i -ésimo elemento do conjunto de obstáculos
\mathbf{p}	Posição no espaço de trabalho \mathcal{W}
$p_{\text{rand}}(\cdot)$	Função pseudoaleatória compartilhada entre os robôs
\mathcal{Q}	Espaço de configurações
\mathbf{q}	Vetor de configurações de um único robô
\mathcal{R}	Time de robôs heterogêneos
\mathbf{R}	Matriz de rotação em $\text{SO}(3)$ de $\{\mathcal{W}\}$ para $\{\mathcal{B}\}$
\mathbf{R}	Vetor de referências para o controle do time de robôs
\mathbb{R}	Conjunto dos números reais
\mathbb{R}^+	Conjunto dos números reais positivos
\mathbb{R}^n	Espaço Euclidiano de n dimensões
R_i	i -ésimo robô do time
\mathbf{r}	Vetor de referência para o controle
\mathcal{S}	Espaço de representação da forma de uma abstração
$\text{SE}(n)$	Grupo Euclidiano especial em n dimensões
$\text{SO}(n)$	Subespaço de rotações em n dimensões
\mathbf{s}	Vetor que define a forma geométrica de uma abstração
\mathcal{T}_α	Árvore de referências multirrobô
\mathcal{T}_r	Árvore de referências de um único robô
\mathcal{T}_x	Árvore de estados de um único robô
\mathbf{U}	Vetor de entrada do time de robôs
$\hat{\mathbf{U}}$	Vetor de entrada estimada para o time de robôs
\mathbf{u}	Vetor de entrada de um único robô
$\hat{\mathbf{u}}$	Vetor de entrada estimada para um único robô
\mathbf{v}	Velocidade translacional no espaço de trabalho \mathcal{W}
v	Comando de velocidade translacional do veículo de <i>Ackerman</i>
\mathcal{W}	Espaço de trabalho
$\mathcal{W}_{\text{livre}}$	Espaço de trabalho livre de obstáculos
$\{\mathcal{W}\}$	Referencial do espaço de trabalho
\mathbf{W}	Incertezas e perturbações que corrompem o time de robôs
\mathbf{w}	Incertezas e perturbações que corrompem o sistema
\mathcal{X}	Espaço de estados
\mathbf{X}	Vetor de estados do time de robôs
\mathbf{x}	Vetor de estados de um único robô
$\hat{\mathbf{x}}$	Vetor de estados estimados para um único robô

Sumário

Agradecimentos	ix
Resumo	xiii
Abstract	xv
Lista de Figuras	xvii
Lista de Algoritmos	xix
Lista de Abreviações	xxi
Lista de Símbolos	xxiii
1 Introdução	1
1.1 Motivação	2
1.2 Abordagem	4
1.3 Contribuições	4
1.4 Organização	5
2 Trabalhos Relacionados	7
2.1 Planejamento de Movimento	8
2.2 Planejamento e Controle de Multirrobo	12
2.3 Contextualização do Trabalho	17
3 Fundamentos	19
3.1 Espaços de Representação	19
3.1.1 Espaço de Trabalho	19
3.1.2 Espaço de Configurações	20
3.1.3 Espaço de Estados	21
3.2 Problema de Planejamento de Movimento	22
3.3 Árvore Aleatória de Exploração Rápida	23

3.3.1	RRT de Malha Aberta	24
3.3.2	RRT de Malha Fechada	27
3.4	Abstração	31
4	Metodologia	35
4.1	Definição do Problema	35
4.2	Suposições	38
4.3	Árvore Pseudoaleatória de Exploração Rápida Multiagente	39
4.4	MRPT Descentralizada em Tempo Real	41
4.4.1	Modelagem da Referência	41
4.4.2	Procedimento de Expansão das Árvores	43
4.4.3	Procedimento de Replanejamento em Tempo Real	46
5	Aplicação ao Controle de Formação	53
5.1	Introdução	53
5.2	Definição do Problema	55
5.2.1	Modelo Dinâmico do Veículo Aéreo	55
5.3	Experimentos Simulados	60
5.3.1	Elipses como Abstração	61
5.3.2	Curvas de Bézier como Abstração	68
5.4	Experimentos com Robôs Reais	73
5.5	Observações Finais	76
6	Aplicação à Cooperação Aéreo-terrestre	77
6.1	Introdução	77
6.2	Definição do Problema	79
6.2.1	Modelo Dinâmico do Veículo Terrestre	79
6.3	Experimentos Simulados	82
6.3.1	Sistema Híbrido como Abstração	82
6.3.2	Simulação de Falha	88
6.4	Experimentos com Robôs Reais	90
6.5	Observações Finais	94
7	Considerações Finais	95
7.1	Conclusões	95
7.2	Trabalhos Futuros	97
	Referências Bibliográficas	99

Capítulo 1

Introdução

– Primeira regra sobre o Clube da Luta... Você não fala sobre o Clube da Luta!

Tyler Durden, O Clube da Luta

Atualmente é notório o crescente número de sistemas autônomos disponíveis para aquisição de dados. Em função disso, os desafios e os benefícios em se utilizar múltiplos robôs na realização de tarefas cotidianas são cada vez mais evidentes. No momento, a coordenação e a cooperação de sistemas robóticos empregados em missões variadas e em ambientes complexos ainda representam um cenário restrito aos grandes centros de pesquisa da área. Porém, acredita-se que, em um futuro próximo, robôs se tornarão mais comuns em nosso dia-a-dia e a demanda por técnicas de controle descentralizadas e de baixo custo computacional serão ainda maiores. Hoje, é possível adquirir robôs terrestres para executar tarefas de limpeza doméstica ou veículos aéreos, equipados com câmeras, para monitoração a baixas altitudes. Isso leva à necessidade de se lidar com problemas de interação entre múltiplos sistemas com características heterogêneas em ambientes complexos. O termo “complexos” neste contexto corresponde a espaços com grande número de obstáculos estáticos e/ou dinâmicos.

Grupos de agentes heterogêneos, em particular, constituem um desafio ainda maior, pois apresentam comportamentos cinemáticos e dinâmicos distintos e interação em diferentes partes do espaço de trabalho. Assim, o problema principal tratado nesta tese pode ser definido da seguinte forma:

Dado um grupo \mathcal{R} composto de n robôs heterogêneos previamente alocados a tarefas conhecidas em um ambiente \mathcal{W} povoado por obstáculos e sujeito a incertezas, planejar e

executar trajetórias para cada um desses agentes em tempo real, de maneira coordenada e descentralizada.

1.1 Motivação

Planejamento é uma tarefa essencial quando se lida com a resolução de problemas complexos. Seja em Robótica, Inteligência Artificial ou Teoria de Controle, planejar ações para se alcançar um objetivo conhecido é um desafio que envolve implicitamente algum tipo de minimização de recursos. Porém, nem todas as questões requerem otimização. Muitas vezes, o simples fato de se encontrar uma solução para o problema já é algo difícil de alcançar. No caso específico dos sistemas robóticos, planejamento é de fundamental importância, pois permite que um agente seja capaz de realizar tarefas relativamente complexas por meio de sequências de ações mais simples, escolhidas por algoritmos de tomada de decisão.

As principais dificuldades de se planejar e executar tarefas em Robótica são basicamente inerentes à natureza das aplicações, sujeitas às características físicas do mundo real. Dentre essas dificuldades, é possível citar incertezas sensoriais, dinâmica dos sistemas e existência de múltiplos corpos (obstáculos, pessoas e robôs) povoando o ambiente. Atualmente grande parte das estratégias de planejamento existentes desconsidera uma ou mais dessas características para simplificar a resolução do problema. Em particular, planejamento de tarefas para mais de um robô constitui um problema ainda maior. Além de questões envolvendo o mundo real, tópicos como interação entre agentes de um mesmo time e coordenação para realização de missões emergem como dificuldades extras aos algoritmos de planejamento. Assim sendo, a principal motivação para se trabalhar nesse tópico é a necessidade por métodos de cooperação de grupos de robôs heterogêneos nos mais variados contextos e tarefas.

A pesquisa em Robótica Cooperativa tem recebido bastante investimento, especialmente na última década. Isso é consequência da variedade de sistemas robóticos de baixo custo que, a cada dia, têm se tornado mais acessíveis, e de sofisticadas tecnologias de sensoriamento e processamento. Atualmente o Laboratório de Visão Computacional e Robótica (VeRLab) da Universidade Federal de Minas Gerais (UFMG) conta com diversos agentes terrestres e plataformas aéreas, e tem atuado ativamente nas linhas de cooperação e coordenação de múltiplos robôs. Em um contexto onde os robôs são distintos entre si e atuam no mesmo meio, estratégias de controle descentralizadas, flexíveis e robustas, são essenciais para a realização de ta-



Figura 1.1. Interação entre múltiplos robôs heterogêneos compartilhando um mesmo ambiente.

refas. A Figura 1.1 ilustra, de maneira simples, a variedade de veículos que podem atuar simultaneamente em um ambiente.

Em casos específicos, como no problema de cooperação aéreo-terrestre, a diferença entre os veículos torna mais difícil a generalização de abordagens que permitam a realização de tarefas complexas. A própria dimensão do ambiente depende da plataforma sobre a qual o robô foi construído. Normalmente, em um mesmo espaço de trabalho, robôs aéreos podem ocupar ambientes tridimensionais (\mathbb{R}^3), ao passo que agentes terrestres se mantêm restritos a superfícies bidimensionais (\mathbb{R}^2). A geometria dos obstáculos também pode (ou não) ser relevante, em função do tipo do veículo. Degraus ou declives podem ser intransponíveis para certos robôs, mas facilmente superáveis por outros.

Exemplo da demanda por aplicações dessa natureza é o projeto denominado “Sistema de Mini-Vants para Monitoramento de Áreas com Minimização de Tempo”, desenvolvido pelas UFMG e Universidade Federal do Amazonas (UFAM), e fomentado pela FINEP¹. O projeto tem por finalidade a construção de grupos de Veículos Aéreos Não Tripulados (*Unmanned Aerial Vehicles*, ou UAVs) para tarefas colaborativas como inspeção, monitoramento e vigilância de grandes áreas de preservação ambiental. Esses são exemplos onde o alcance limitado dos dispositivos de comunicação impõe restrições sérias à coordenação e técnicas robustas de planejamento se tornam imprescindíveis.

¹Agência Financiadora de Estudos e Projetos do Ministério de Ciência e Tecnologia do Brasil.

1.2 Abordagem

Conforme mencionado, este trabalho aborda o problema da navegação de um grupo \mathcal{R} composto de n robôs heterogêneos sujeitos a incertezas de sensoriamento em um ambiente \mathcal{W} com obstáculos. Especificamente são consideradas tarefas com requisitos de planejamento e controle de trajetória para um time de veículos em espaços complexos. Esse time, com restrições dinâmicas de movimento, deve se mover pelo espaço e simultaneamente desviar dos obstáculos e evitar colisões com outros robôs. Isso é feito de forma descentralizada e com uma carga reduzida de comunicação direta entre os agentes.

Este trabalho apresenta um método de planejamento de trajetórias para grupos de veículos heterogêneos, baseado no Algoritmo da Árvore Aleatória de Exploração Rápida (*Rapidly-exploring Random Tree*, ou RRT) de Malha Fechada [Kuwata et al., 2008]. Essa técnica gera caminhos aleatórios para sistemas simples, em malha fechada e em tempo real. Nossa abordagem permite a cada agente R_i do time planejar sua própria trajetória, de maneira rápida e descentralizada. Sem conhecimento *a priori* do comportamento de outros robôs no ambiente, cada agente é capaz de evitar colisões com seus companheiros e de se coordenar em relação a eles.

Inicialmente é proposto o Árvore Pseudoaleatória de Exploração Rápida Multiagente (*Multiple Rapidly-exploring Pseudorandom Trees*, ou MRPT), um algoritmo determinístico que permite explorar todos os benefícios da RRT de Malha Fechada (*Closed Loop RRT*, ou CL-RRT) em sistemas multirrobôs cooperativos. Em seguida, é apresentado o uso de um modelo cinemático para a propagação da referência de controle do sistema, dependente do tipo de missão alocada, baseado na ideia de abstrações. Como será mostrado nos capítulos 5 e 6, curvas paramétricas simples são utilizadas como modelos de referência para realizar o planejamento do grupo de robôs. Por fim, é proposta uma extensão do Algoritmo CL-RRT para o caso multirrobô, com uma abordagem completamente descentralizada.

1.3 Contribuições

Este trabalho tem por objetivo apresentar um método de coordenação descentralizado para grupos de robôs heterogêneos. Assim sendo, as seguintes contribuições são oferecidas:

1. Um arcabouço de planejamento e controle cooperativo de múltiplos robôs heterogêneos. Comparada aos outros trabalhos da literatura, nossa abordagem

permite o planejamento descentralizado, e em tempo real, para grupos de robôs com características dinâmicas diversas. O algoritmo, baseado em malha fechada, reduz a susceptibilidade da técnica a perturbações e incertezas do sistema. Além disso, habilita o uso de veículos cujos modelos são complexos (não lineares), incertos e com muitos graus de liberdade. Nosso método se baseia na predição do comportamento do time por cada um de seus membros, de forma descentralizada. Isso reduz o volume de comunicação entre os agentes, tornando o sistema mais robusto a falhas ou perdas de conectividade temporárias (Capítulo 4).

2. Uma estratégia baseada no uso de abstrações (curvas paramétricas) para o controle de formação de múltiplos robôs. O método permite a manutenção de formações geométricas propagadas através do ambiente por meio do planejador descentralizado proposto aqui. Ao contrário dos trabalhos existentes, nossa abordagem possibilita o uso de inúmeras curvas paramétricas, inclusive as que não respeitam certas premissas do controle clássico baseado em abstrações (Capítulo 5).
3. Uma estratégia de cooperação aéreo-terrestre para missões de vigilância, monitoramento e escolta. É utilizada mais de uma abstração nesse caso, mostrando que a metodologia proposta possui a capacidade de gerenciar mais de um grupo de robôs simultaneamente. A aplicação de técnicas de controle simplificadas permite ainda a utilização de times compostos por robôs com características distintas, como robôs aéreos, terrestres ou mesmo aquáticos (Capítulo 6).

1.4 Organização

O restante do texto está organizado da seguinte forma:

- O Capítulo 2 apresenta a revisão de literatura contendo os trabalhos mais relevantes aos problemas de planejamento e controle cooperativo de múltiplos robôs.
- O Capítulo 3 introduz alguns dos conceitos fundamentais necessários ao entendimento desta tese. Traz ainda uma breve discussão das ferramentas utilizadas. São apresentadas outras referências ao longo do capítulo para prover maiores detalhes sobre os temas abordados.

- O Capítulo 4 apresenta e discute a metodologia principal do trabalho, a qual se denominou Árvore Pseudoaleatória de Exploração Rápida Multiagente (*Multiple Rapidly-exploring Pseudorandom Trees*, ou MRPT).
- O Capítulo 5 mostra a aplicação da metodologia proposta a um problema mais específico: o planejamento e controle de formação de um grupo de robôs. O grupo em questão é composto de robôs aéreos de asa-rotativa do tipo Pouso e Decolagem Vertical (*Vertical Take-off and Landing*, ou VTOL).
- O Capítulo 6 apresenta a instanciação do método a outro problema específico: a cooperação aéreo-terrestre em missões de vigilância e escolta. Nesse caso, nosso grupo é composto de robôs aéreos do tipo VTOL e de robôs terrestres baseados na plataforma de *Ackerman*.
- Finalmente no Capítulo 7 são realizadas discussões sobre o método apresentado e os possíveis trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Se todos os registros contassem o mesmo conto, então a mentira passaria a ser história e se tornaria verdade. “Quem controla o passado”, dizia o slogan do Partido, “controla o futuro: quem controla o presente, controla o passado”.

1984, George Orwell

O termo “algoritmo de planejamento” pode ser interpretado de diferentes formas, dependendo do tipo de contexto no qual está inserido [LaValle, 2006]. Em Robótica, significa métodos para gerar sequências de movimentos (caminhos ou trajetórias) capazes de guiar o robô através de ambientes com características geométricas complexas. Em Inteligência Artificial, corresponde geralmente a sistemas capazes de aplicar regras de decisão para calcular ações apropriadas ao agente em determinadas situações. Em Teoria de Controle, representa estratégias, em geral analíticas, para computar trajetórias realizáveis pelo sistema utilizando informação realimentada e visando otimização. Planejamento e controle são tipicamente usados como termos complementares em Robótica. Enquanto o primeiro é considerado um módulo de mais alto nível na arquitetura do sistema, o segundo é normalmente delegado a níveis mais baixos, sendo fundamental para as partes de hierarquia superior. Porém, tal divisão se torna menos evidente quando se consideram problemas com múltiplos robôs, uma vez que a coordenação do grupo pode acontecer em qualquer nível da arquitetura. Este capítulo apresenta uma revisão do estado da arte em planejamento e controle de sistemas multirrobôs, especialmente no que tange a métodos aleatórios e a estratégias descentralizadas.

2.1 Planejamento de Movimento

Atualmente a literatura na área de planejamento de movimento para veículos autônomos é bastante vasta. Em [LaValle, 2006], o autor apresenta diversas metodologias, que vão desde o planejamento para robôs pontuais em ambientes discretizados, até veículos com restrições cinemáticas e dinâmicas em espaços contínuos. No entanto, a maioria das técnicas é desenvolvida para um único robô interagindo com o meio. Os métodos existentes podem ser divididos em duas grandes categorias: *abordagens contínuas* e *abordagens discretas*. Essa divisão se baseia na forma como o ambiente de navegação é modelado dentro do problema.

Abordagens contínuas, também chamadas de Planejamento em Espaços de Estado Contínuos [LaValle, 2006], são os mais numerosos e podem ser divididos em uma série de subclasses. Um exemplo comum desse tipo são as técnicas baseadas em realimentação, como Campos Vetoriais (*Vector Fields*) [Arrowsmith & Place, 1992], ideais para planejamento em espaços mais simples, com um número pequeno de dimensões. Grande parte dos métodos contínuos entra em uma subcategoria de planejamento baseado em otimização, como o Controle por Modelo Preditivo (*Model Predictive Control*, ou MPC) e a Programação Linear Inteira Mista (*Mixed Integer Linear Program*, ou MILP). Tais abordagens tendem a gerar bons resultados, inclusive no caso de múltiplos robôs. Porém, são impraticáveis para problemas com grande quantidade de veículos e obstáculos em espaços com muitas dimensões. A MPC, em particular, é um tipo de técnica em que a lei de controle ótima em tempo finito é determinada por meio da solução de uma função de otimização [Keviczky et al., 2007]. Já a MILP [Schouwenaars et al., 2001] é uma técnica que incorpora o problema de colisão ao planejamento de maneira simples, permitindo a navegação ótima em espaços com obstáculos estáticos e dinâmicos. Em geral, abordagens baseadas em otimização se adaptam bem às características dinâmicas dos robôs. Porém, elas não escalam bem para o caso de ambientes complexos e grupos de agentes heterogêneos.

Já os métodos baseados em discretização do ambiente estão entre os primeiros a serem reportados na literatura. Eles recebem esse nome por empregarem ferramentas de busca fundamentadas em abordagens discretas para realizar o planejamento [Latombe, 1991]. Para isso, normalmente consideram a representação do mundo como uma grade uniforme e finita. Dentre as técnicas mais populares estão o algoritmo de *Dijkstra* [Dijkstra, 1959], o A^* [Hart et al., 1968] (e suas variantes, D^* [Stentz & Mellon, 1993] e R^* [Likhachev & Stentz, 2008]) e outras abordagens baseadas em Programação Dinâmica (*Dynamic Programming*, ou DP) [Flint et al., 2002]. Existem também os métodos combinatórios, como a Decomposição por Células e os

Mapas de Rotas (*Roadmaps*), capazes de determinar caminhos de forma exata, sem aproximações. Essas metodologias apresentam bons resultados e garantem convergência, porém, não se adaptam bem a grupos de veículos heterogêneos em função da diferença do comportamento dinâmicos entre os agentes, se tornando caros devido ao aumento das dimensões do espaço de busca.

Outro conjunto de estratégias de planejamento discreto é o constituído pelos chamados Algoritmos Baseados em Amostragem, também conhecidos por Mapas de Rotas Probabilísticos (*Probabilistic Road Maps*, ou PRMs) [LaValle, 2006]. Tratam-se de técnicas aleatórias, cuja principal vantagem é evitar a construção explícita do espaço de configurações para o problema, o que pode se tornar extremamente difícil no caso de ambientes complexos (obstáculos com formas não poligonais ou não poliédricas). Um levantamento detalhado das particularidades e dos métodos relativos a esse tópico é apresentado em [Lama, 2008].

No centro de tais estratégias está o *Módulo de Detecção de Colisão*, ferramenta que possibilita o desenvolvimento de planejadores independentes da geometria dos obstáculos no espaço de trabalho. Isso melhora o desempenho dos algoritmos de maneira significativa e permite a abordagem de problemas em ambientes complexos (com muitos obstáculos). Entretanto, perde-se em termos da otimalidade das soluções apresentadas. Além disso, não se pode garantir a completude de tais métodos. Em outras palavras, o procedimento só é probabilisticamente completo para um número relativamente grande de amostras aleatórias e iterações.

Dentre as diversas técnicas de planejamento de movimento baseadas em amostragem probabilística, destaca-se a *Árvore Aleatória de Exploração Rápida* (*Rapidly-exploring Random Tree*, ou RRT). Essa técnica é basicamente um algoritmo capaz de produzir árvores com caminhos de exploração que crescem rapidamente através do ambiente conhecido. A RRT de Malha Aberta (*Open Loop RRT*, ou OL-RRT) (ou simplesmente RRT) foi inicialmente introduzida em [LaValle & Kuffner, 1999]. A abordagem utiliza pontos aleatórios, sorteados no espaço de estados do robô, para gerar caminhos realizáveis de maneira incremental em um espaço povoado por obstáculos. Por realizável, se entende que as trajetórias respeitam todas as restrições de movimento do veículo e do ambiente.

A Figura 2.1 apresenta estágios intermediários da expansão de uma árvore para o caso de um robô holonômico através do espaço \mathbb{R}^2 , sem obstáculos. Partindo de um ponto inicial no centro do mapa, a árvore se espalha rapidamente, promovendo uma cobertura homogênea do local. Em princípio, não existe um objetivo que guie o espalhamento, mas a maior parte dos trabalhos estabelece um alvo para o crescimento direcionado da estrutura.

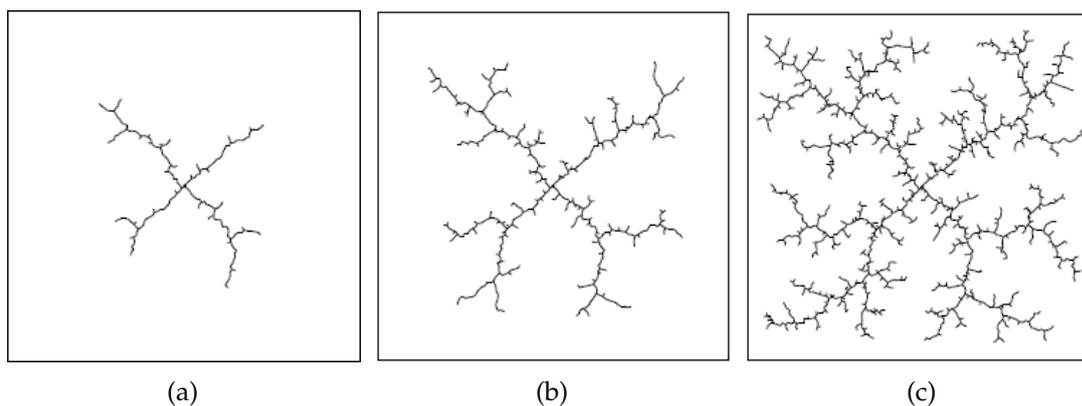


Figura 2.1. Expansão da Árvore Aleatória de Exploração Rápida (RRT) aplicada a um robô holonômico em um ambiente inicialmente sem obstáculos [LaValle & Kuffner, 1999].

RRTs têm sido exploradas na última década e vêm recebendo cada vez mais atenção em função das diversas vantagens que apresentam em relação a outras técnicas de planejamento. Algumas das características mais interessantes são a simplicidade de aplicação (mesmo em sistemas com alto grau de complexidade) e a rapidez na convergência dos resultados (dependendo do tipo de ambiente), além do fato de que o método é probabilisticamente completo. Por isso, a RRT constitui uma aproximação simples e eficiente, capaz de planejar o movimento ao longo de ambientes complexos (com grande número de obstáculos) de forma rápida. Ela ainda permite lidar com modelos de veículos lineares e não lineares, e pode ser aplicada a espaços de trabalho e de configurações (veja Capítulo 3 para mais detalhes).

Além disso, RRTs apresentam como vantagem para o nosso trabalho a capacidade de lidar com sistemas com grande número de graus de liberdade e tipos diferentes de restrições, caso dos grupos multirrobo heterogêneos. Em [Bruce & Veloso, 2002], por exemplo, os autores introduzem a RRT de Execução Estendida (*Execution-extended RRT*, ou ERRT), adaptação da OL-RRT onde um conjunto de heurísticas atua na otimização do planejamento de caminhos em tempo real.

O trabalho [Burns & Brock, 2007] apresenta um estudo que mostra que o uso de distribuições uniformes na etapa de amostragem para expansão da árvore aleatória reduz a eficiência dos métodos, à medida que as dimensões do problema crescem. Os autores propõem ainda um método que, em vez de utilizar distribuições aleatórias simples, aplica funções de utilidade para guiar o crescimento das árvores, no intuito de aumentar o desempenho dos algoritmos de planejamento. Em [Choi, 2007], o autor lida com o planejamento de movimento de manipuladores com grande número de graus de liberdade. Usando a RRT, é apresentada uma heurística

de otimização que utiliza mapas de rotas do espaço de trabalho para guiar a árvore de maneira mais eficiente. Enquanto o mapa é aplicado apenas ao planejamento da trajetória da ferramenta na extremidade do manipulador, a RRT se encarrega de determinar o movimento das demais juntas do robô, reduzindo a complexidade do problema.

Os autores de [Shkolnik et al., 2009] apresentam a RRT Guiada por Alcançabilidade (*Reachability-Guided RRT*, ou RG-RRT), variação da RRT que trata a otimização de planejamento para robôs terrestres diferenciais. A abordagem utiliza o diagrama de Voronoi para evitar que o robô atinja regiões muito próximas aos obstáculos ou às bordas do ambiente, características que reduzem a eficiência do método tradicional devido ao excesso de colisões.

Em contrapartida à RRT de Malha Aberta, os autores de [Kuwata et al., 2008, 2009] introduzem a RRT de Malha Fechada (*Closed Loop RRT*, ou CL-RRT). Como o nome diz, em vez de efetuar o planejamento para sistemas em malha aberta atuando diretamente nas entradas do veículo, na CL-RRT a árvore de caminhos gera sinais de referência para os controladores internos do robô. Esses controladores recebem o caminho, em princípio não realizável, e o transformam em planos exequíveis pelo agente autônomo (veja Capítulo 3 para detalhes). Isso traz como vantagens em relação ao método original a adição de duas características principais: i) redução do espaço de planejamento do problema, já que o espaço de referências de controle é, em geral, menor do que o de estados [Kuwata et al., 2008]; e ii) aumento da robustez a perturbações externas nos caminhos gerados quando da execução do plano, uma vez que controladores são utilizados na etapa de planejamento [Kuwata et al., 2009].

A Figura 2.2 apresenta um exemplo da expansão das árvores no sistema em malha fechada (adaptado de [Kuwata et al., 2008]). A árvore alaranjada representa o conjunto de sinais de referência empregados aos controladores do robô. Já a árvore em verde constitui a saída desses controladores, tendo a primeira árvore como referência. A ideia é que a expansão de ambas as estruturas conduzam o veículo através das partes do ambiente livres de obstáculos.

Além do procedimento de expansão da árvore, característico à RRT, o método possui ainda um laço de execução que aplica o melhor caminho corrente ao controlador do veículo. Isso torna o processo aplicável ao controle em tempo real de robôs reais e permite o replanejamento sob incertezas quanto aos obstáculos do ambiente. A técnica foi utilizada com sucesso como componente do sistema embarcado do veículo *Talos*, carro autônomo desenvolvido pelo time do *Massachusetts Institute of*

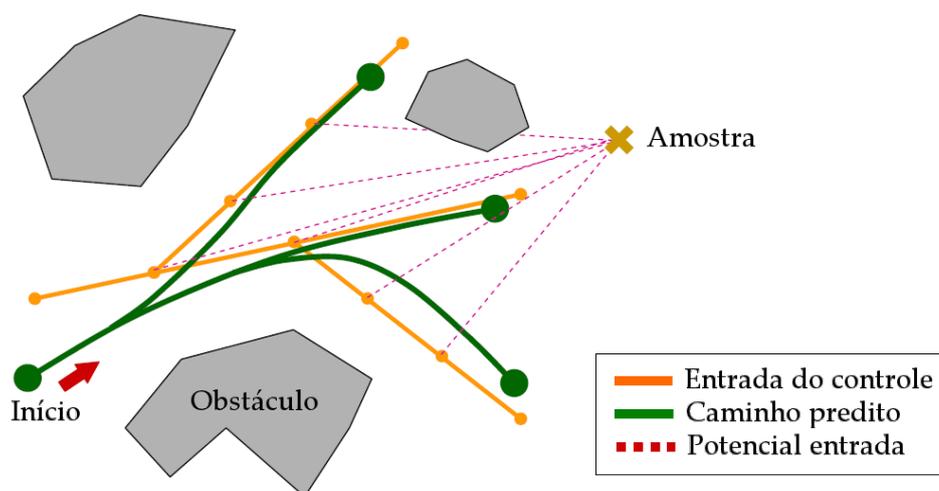


Figura 2.2. Expansão da árvore de exploração (RRT) em malha fechada aplicada a um veículo terrestre do tipo carro (modelo de *Ackerman*). A árvore alaranjada representa o conjunto de sinais de referência empregados aos controladores do robô, enquanto que a árvore em verde constitui a saída desses controladores tendo a primeira árvore como referência (adaptada de [Kuwata et al., 2008]).

Technology para o *DARPA Urban Challenge*¹ [Kuwata et al., 2009].

Em [Luders et al., 2010] os autores descrevem um estudo comparativo entre a OL-RRT e a CL-RRT no que se refere à propagação do erro nos estados do robô, submetido a perturbações externas. A conclusão principal é a de que essa última técnica é mais robusta a incertezas na predição das trajetórias do veículo, mantendo o erro sempre limitado entre o modelo (para o caso linear) e o comportamento real do robô. O modelo dinâmico bidimensional de um quadricóptero é utilizado como exemplo na análise. Já Kuwata et al. [2008] propõem ainda uma série de heurísticas para diminuição desse erro, levando o veículo a se comportar de maneira próxima ao modelo proposto. O presente trabalho estende o uso da CL-RRT para um sistema multirrobôs e propõe uma heurística que melhora a coordenação temporal entre os veículos do time, reduzindo erros na execução do plano e reduzindo a possibilidade de colisões entre os robôs.

2.2 Planejamento e Controle de Multirrobôs

O planejamento e controle de sistemas multirrobôs é um dos problemas mais fundamentais e mais estudados na área da Robótica Cooperativa. Trata-se de um desafio de maior complexidade quando comparado ao planejamento de movimento de um

¹Competição de veículos autônomos trafegando em ambientes urbanos (patrocinada pelo *Defense Advanced Research Projects Agency/USA*).

único robô, não apenas devido a possíveis colisões entre agentes, mas também, porque a ordem dimensional dos espaços tratados aumenta significativamente. Muitas das técnicas apresentadas anteriormente precisam ser adaptadas a esse novo contexto, e algumas delas não se adaptam bem, especialmente os métodos baseados em otimização.

No estudo dos sistemas multirrobo, existe uma série de desafios e subproblemas a serem considerados. Vários deles já foram abordados em projetos desenvolvidos no Laboratório de Visão Computacional e Robótica da UFMG. Apenas como exemplos, é possível citar a alocação de tarefas utilizando leilão [Shiroma & Campos, 2009], o tratamento de congestionamento em enxames de robôs [Marcolino & Chaimowicz, 2009b,a; Santos & Chaimowicz, 2011] e o transporte cooperativo de objetos [Pereira et al., 2004].

Em se tratando de sistemas distribuídos, como no caso de um time de agentes autônomos, descentralização se torna um ponto chave para a solução. Ela permite reduzir a complexidade do problema, dividindo-o em partes menores e permitindo a aplicação de estratégias mais simples do que as usadas de maneira centralizada. Infelizmente ela também traz outras dificuldades, como por exemplo, a limitação do caso de informação global.

Parker [2008] apresenta uma das taxonomias mais aceitas para a classificação das abordagens de planejamento multirrobo. Nele, a autora divide o problema em quatro partes: interações *coletivas*, *cooperativas*, *colaborativas* e *coordenativas*. Tal divisão é definida principalmente pela forma como os agentes interagem entre si.

Sistemas coletivos representam o tipo mais simples de interação entre robôs. Nesse caso, cada agente não tem qualquer conhecimento sobre os objetivos, ações ou mesmo existência dos demais componentes do time. Sistemas cooperativos são compostos por agentes que têm consciência da existência de outros e que compartilham objetivos e ações comuns em benefício de todo o time. Grupos de robôs cooperativos tipicamente realizam tarefas em conjunto, onde todos os membros contribuem para maximizar o ganho da equipe. Sistemas colaborativos, por sua vez, possuem robôs que apresentam objetivos individuais, mas que ao mesmo tempo estão cientes da presença de outros no espaço. Eles tentam, na medida do possível, ajudar o time. Finalmente sistemas coordenativos são grupos de robôs onde os membros sabem da presença de outros agentes no ambiente, porém suas missões individuais e suas ações não contribuem com o desempenho global do time de maneira direta. Embora nos Capítulos 5 e 6 sejam apresentadas aplicações da metodologia a grupos de robôs executando tarefas coordenativas e cooperativas respectivamente, é possível generalizar a abordagem para qualquer dos paradigmas discutidos.

Assim como no caso de um único robô, existem muitas técnicas para controle e planejamento de múltiplos veículos. A maioria é adaptada de metodologias mais simples, onde o problema se restringe a um único agente no ambiente. Mas, embora os métodos descritos na seção anterior apresentem vantagens quando considerados individualmente, é necessário algum tipo de coordenação entre os robôs que os executam para que o resultado coletivo seja satisfatoriamente alcançado [Keviczky et al., 2004]. Como exemplo, Dunbar & Murray [2002] apresentam o Controle por Modelo Preditivo Distribuído (*Distributed MPC*, ou DMPC), uma versão modificada da MPC para o caso multiagente. Assim como na técnica original, a solução da DMPC vem da resolução de um problema de otimização. A diferença nesse caso é que o problema é dividido em vários subproblemas, onde cada agente é levado a planejar seu movimento e comunicá-lo aos demais membros do time. Aplicações para casos de tempo real são discutidas em [Dunbar, 2002]. A principal desvantagem desses métodos é que apenas um agente pode modificar seu plano de cada vez. Isso leva a atrasos, caso haja necessidade de replanejamento por muitos membros do time simultaneamente.

Outra vertente do planejamento cooperativo vem dos algoritmos de consenso [Olfati-saber et al., 2007], estratégias de controle que permitem a coordenação de ações por meio da troca de mensagens entre os agentes do time. Em [Guerrero et al., 2010], os autores utilizam uma abordagem baseada em consenso e propõem um conjunto de controladores não lineares individuais para UAVs do tipo quadricóptero. Esses controladores são posteriormente utilizados no controle de formação de mais alto nível, baseado na abordagem de tipo seguimento de líder. A trajetória da formação é determinada em função do centro de massa do grupo.

Em [Ippolito et al., 2008], os autores apresentam uma estratégia de controle baseada em Sistemas de Controle Polimórfico (*Polymorphic Control System*, ou PCS). Essa abordagem permite a reconfiguração de sistemas de controle em coordenação de múltiplos agentes heterogêneos. Como prova de conceito, é apresentado o caso onde um UAV simula falha de sensoriamento no exato instante da execução de uma manobra de pouso automático. Identificada a falha, o arcabouço aloca um Veículo Terrestre Não tripulado (*Unmanned Ground Vehicle*, ou UGV), equipado com uma câmera, para auxiliar o veículo aéreo na tarefa, enviando informações do solo e fechando a malha. Essa reconfiguração aumenta a tolerância a falhas e torna a estratégia de controle mais eficiente, embora a abordagem seja totalmente centralizada.

Devido às principais vantagens, RRTs também foram aplicadas ao problema de coordenação e cooperação multirrobôs com relativo sucesso. Porém, a maior parte dos artigos existentes no reduzido corpo de literatura desse contexto apresenta es-

estratégias centralizadas e pouco escaláveis quanto ao número de agentes no time. Em [Kamio & Iba, 2005, 2006], por exemplo, a técnica aleatória é aplicada ao problema de transporte cooperativo de objetos. Robôs humanoides utilizam informações locais de suas próprias câmeras para planejar e replanejar suas ações no meio. Todo o arcabouço é centralizado em um computador que executa o planejamento e o distribui entre os agentes.

Os autores de [Vahrenkamp et al., 2010] também apresentam um algoritmo baseado na RRT para realização de transporte cooperativo entre múltiplos manipuladores móveis, cada um com cerca de vinte graus de liberdade. Também nesse caso, foi utilizado um método de verificação centralizado para garantir a não ocorrência de colisões entre robôs dentro do mesmo ambiente. O sucesso do procedimento mostra a capacidade da técnica de se lidar com espaços de configurações de alta dimensão.

Em nossos trabalhos [Macharet et al., 2009; Alves Neto et al., 2009, 2010], foram propostas técnicas centralizadas para o planejamento de trajetórias de UAVs de asa fixa de forma coordenada e em ambientes com obstáculos utilizando a RRT como base. Nesse contexto, todos os robôs apresentam modelos equivalentes e simplificados, baseados em restrições dinâmicas de curvatura. Em particular, foram utilizadas curvas paramétricas para minimizar o custo de geração da árvore de trajetórias no espaço \mathbb{R}^2 . Um procedimento de replanejamento gerencia a colisão entre os robôs aéreos, voando a altitudes constantes e equivalentes. O resultado do método aplicado a 4 UAVs de asa fixa é apresentado na Figura 2.3.

O método aqui proposto é completamente descentralizado e, diferentemente dos demais discutidos anteriormente, pode ser aplicado a grupos de robôs com características dinâmicas heterogêneas (detalhes no Capítulo 4).

Um das primeiras tentativas de descentralização em planejamento de multirobôs baseados em métodos aleatórios foi apresentada em [Frazzoli, 2003]. Nesse trabalho, o autor discute uma metodologia de planejamento para times multirobôs em ambientes com obstáculos. A estratégia apresentada é distribuída entre os agentes, mas o planejamento é sincronizado de modo que um agente planeje somente após conhecer o plano dos demais membros do time. Embora simples, a técnica se mostra bastante promissora, porém com custo computacional elevado.

Em um trabalho mais recente, Aoude et al. [2010] propõem o uso de predição baseada na simulação da CL-RRT para que um robô possa inferir a “intenção” de movimento dos demais agentes do time em determinado intervalo de tempo. Com isso, baseando-se apenas em informações dos estados dos vizinhos e conhecendo seus respectivos modelos, é possível determinar um conjunto de pontos alcançá-

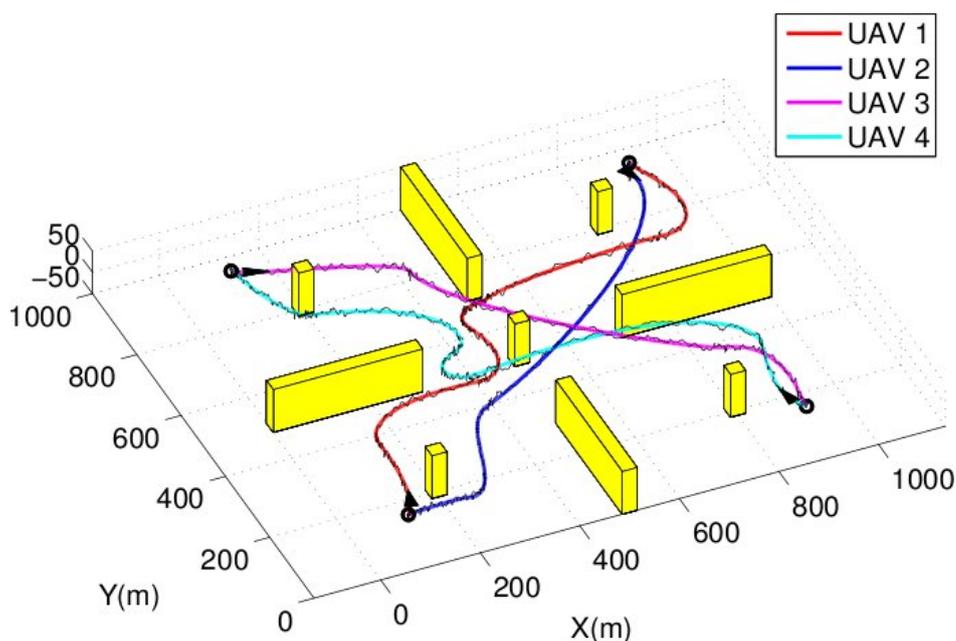


Figura 2.3. Planejamento de trajetórias centralizado, baseado em RRTs, para múltiplos UAVs de asa fixa [Alves Neto et al., 2010]. Cada aeronave parte de um ponto inicial até outro ponto final, utilizando curvas suaves para transporem os obstáculos.

veis por esses, os quais passam a ser evitados na etapa de planejamento. Assim, com a solução de um problema de otimização, o robô consegue evitar colisões com os demais de maneira descentralizada. Porém, tal abordagem lida apenas com informações locais, podendo levar a conflitos insolúveis uma vez que o método não permite a cooperação entre agentes.

Os autores de [Desaraju & How, 2011] mostram a RRT Multiagente Descentralizada (*Decentralized Multi-Agent RRT*, ou DMA-RRT), um conjunto de estratégias para o planejamento cooperativo de múltiplos robôs baseadas na CL-RRT. De forma semelhante a [Frazzoli, 2003], o método promove o sincronismo de planejamento entre agentes, mas nesse caso é utilizada uma técnica de leilão para determinar a ordem em que o replanejamento ocorre. Para evitar colisões os agentes são levados a reestruturar suas trajetórias, sendo permitido apenas ao vencedor do leilão fazê-lo. Um agente pode ainda impedir o movimento de outros em casos específicos (como corredores estreitos, por exemplo) melhorando assim a cooperação do time como um todo. Cada novo caminho deve ser informado aos demais agentes para que eles sejam capazes de atualizar suas restrições de colisão a cada novo instante de tempo.

Essas técnicas, apesar de descentralizadas, estipulam que cada agente do time saiba da existência dos demais e que conheçam pelo menos os modelos (ainda que

incertos) de todos os robôs. Apesar da falta de consenso sobre a questão, muitos pesquisadores não consideram isso como um sistema totalmente descentralizado, já que algum conhecimento *a priori* deve ser compartilhado entre o time. Além disso, esses métodos são fortemente dependentes de uma rede conectada (e perfeita) de comunicação e demandam grande tráfego de informação devido ao replanejamento e à coordenação (após cada mudança os robôs devem informar seus novos planos para os demais). Outro problema é que a modificação de planos individuais acarreta em eventuais invalidações de partes do planejamento global do time, o que causa desperdício de processamento e atraso no replanejamento. Neste trabalho é proposta uma técnica menos dependente da estrutura da rede e que permite o replanejamento totalmente descentralizado e simultâneo de vários agentes de forma descentralizada e sem perda das partes já planejadas.

2.3 Contextualização do Trabalho

A metodologia tratada neste trabalho para solução do problema de navegação utilizando múltiplos robôs (discutido em detalhes no Capítulo 4) pode ser descrita como uma abordagem em espaços de estados contínuos baseada em planejamento por amostragem aleatória. A técnica proposta é uma extensão da CL-RRT para o caso multirrobôs, mantendo a capacidade de planejamento e replanejamento de movimentos de maneira rápida, mesmo em ambientes complexos e dinâmicos.

Em suma, com base nas questões principais discutidas neste capítulo, é possível destacar as seguintes características para este trabalho:

1. trata-se de um método totalmente descentralizado;
2. permite a aplicação com replanejamento em tempo real;
3. é aplicável a grupos de veículos com características heterogêneas;
4. estende-se às considerações dinâmicas do robô e suas eventuais restrições no espaço de estados;
5. permite o planejamento em ambientes com obstáculos estáticos e/ou variantes no tempo;
6. robusto a incertezas nos parâmetros dos modelos dos robôs e nos sensores;
7. baixo tráfego de comunicação entre agentes;

8. replanejamento não interfere e não invalida planos de outros agentes;
9. apresenta comportamento determinístico (permite a predição por parte de outros agentes do time).

Tais itens são fundamentais para a definição do problema tratado nesta tese (a ser definido no Capítulo 4). A Tabela 2.1 contém uma listagem de alguns dos artigos considerados mais relevantes ao problema e suas respectivas limitações. Ela mostra a relação entre esses trabalhos e as características descritas anteriormente. A marca assinalada em cada célula indica que o trabalho apresenta a característica destacada. Na última linha, o trabalho proposto aqui é apresentado, o qual preenche todos os tópicos levantados. Essas características serão mais bem discutidas ao longo da apresentação da metodologia no Capítulo 4.

Tabela 2.1. Caracterização dos trabalhos e contribuições

Trabalhos/Contribuições	1	2	3	4	5	6	7	8	9
[Schouwenaars et al., 2001]					✓		✓		✓
[Flint et al., 2002]						✓	✓		✓
[Dunbar & Murray, 2002]	✓			✓		✓		✓	✓
[Dunbar, 2002]	✓	✓		✓		✓		✓	✓
[Frazzoli, 2003]	✓		✓	✓	✓	✓			
[Kamio & Iba, 2005, 2006]		✓			✓		✓		
[Olfati-saber et al., 2007]	✓			✓		✓			✓
[Ippolito et al., 2008]		✓	✓			✓	✓		
[Aoude et al., 2010]		✓	✓	✓	✓	✓			
[Alves Neto et al., 2010]					✓		✓	✓	
[Guerrero et al., 2010]	✓			✓		✓			✓
[Vahrenkamp et al., 2010]		✓			✓		✓	✓	
[Desaraju & How, 2011]		✓	✓	✓	✓	✓			
Nosso trabalho	✓	✓	✓	✓	✓	✓	✓	✓	✓

Apesar de alcançar todos os pontos levantados, existem outros tópicos cobertos pela literatura que o presente trabalho deixa de considerar. O principal deles é a questão da otimização dos planos gerados. Como o resultado da nossa metodologia é gerado com base em funções aleatórias e em heurísticas, não é possível garantir que o plano gerado será o melhor possível, segundo uma função de custo qualquer. Também não é possível garantir que seja gerado um resultado satisfatório em tempo finito e nem mesmo sua convergência para o objetivo da tarefa. Essas e outras questões são apresentadas como trabalho futuro ao final deste texto.

Capítulo 3

Fundamentos

Dr. Watson: “Mas o sistema solar!...”

Sherlock Holmes: “Que importância tem isso para mim?... Você diz que giramos em torno do Sol. Se girássemos em volta da Lua, isso não faria a menor diferença para o meu trabalho.”

Um Estudo em Vermelho, Sir Arthur Conan Doyle

Este capítulo apresenta uma breve revisão de alguns dos fundamentos necessários ao trabalho. Algumas notações e formalismos mais básicos do texto são introduzidos na sequência.

3.1 Espaços de Representação

Assim como em outras áreas, em Robótica a modelagem matemática de sistemas complexos utiliza *espaços de representação*. Essas representações servem para generalizar soluções para problemas característicos do controle de tais sistemas, modelando restrições de movimento, disposição espacial e comportamento dinâmico [Choset et al., 2005]. São três os espaços principais: espaços de trabalho, de configurações e de estados.

3.1.1 Espaço de Trabalho

O *espaço de trabalho* \mathcal{W} é definido como sendo a representação (normalmente Euclidiana) do meio físico onde o robô atua. Em Robótica, é essencialmente o espaço em que se estabelecem os referenciais do mundo. Robôs terrestres (planares ou pontuais) tipicamente atuam em um ambiente $\mathcal{W} \equiv \mathbb{R}^2$, ao passo que robôs aéreos utilizem

a representação $\mathcal{W} \equiv \mathbb{R}^3$. Já manipuladores robóticos tipicamente utilizam apenas parte desse espaço, representado por $\mathcal{W} \subset \mathbb{R}^2$ ou $\mathcal{W} \subset \mathbb{R}^3$.

No mundo real, é comum que nem todas as posições \mathbf{p} em \mathcal{W} sejam alcançáveis pelo robô em qualquer instante de tempo t . Também é comum que existam obstáculos espalhados pelo ambiente, representados por meio do conjunto \mathcal{O} . Assim, o *espaço de trabalho livre* é definido como o subconjunto de \mathcal{W} em que não há obstáculos. Formalmente:

$$\mathcal{W}_{\text{livre}} = \mathcal{W} \setminus \left(\bigcup_i \mathcal{O}_i \right),$$

onde \mathcal{O}_i é o i ésimo obstáculo em \mathcal{O} e \setminus é o operador de diferença entre conjuntos [Latombe, 1991].

Existem basicamente dois tipos de obstáculos, que conduzem à classificação de um ambiente em estático ou dinâmico. *Ambientes estáticos* possuem apenas de *obstáculos estáticos*, ou seja, obstáculos cuja configuração não muda ao longo do tempo. Nesse caso, as regiões ocupadas no espaço de trabalho são invariantes. Já os *ambientes dinâmicos* contêm, além de obstruções estáticas, *obstáculos dinâmicos* cuja configuração se altera com o tempo. Em arcabouços multirrobôs, onde os agentes compartilham o mesmo meio físico, um robô sempre será um obstáculo (dinâmico) para os demais. Ambientes dinâmicos ainda apresentam a característica de serem parcialmente conhecidos, uma vez que as posições de todos os robôs no espaço podem não ser conhecidas ao mesmo tempo. Este trabalho assume o cenário de múltiplos robôs em ambientes dinâmicos.

3.1.2 Espaço de Configurações

A segunda representação, também muito utilizada em Robótica, é o *espaço de configurações* \mathcal{Q} . A configuração \mathbf{q} de um objeto no espaço de trabalho \mathbb{R}^n é definida como sendo o conjunto mínimo de variáveis necessárias para se especificar completamente esse objeto no espaço. Em outras palavras, a configuração descreve cada ponto do objeto em relação ao referencial $\{\mathcal{W}\}$ do mundo. Já a cardinalidade desse conjunto é definida como o *número de graus de liberdade* do sistema.

As configurações também definem transformações de translação e rotação, representadas por um *Grupo de Lie*¹ $SE(n)$, tal que a movimentação espacial do robô se dá em espaços que são combinações desse grupo (chamados *Colchetes de Lie*), e cujas

¹Variedade diferenciável, chamada de *Conjunto de Lie*.

projeções recaiam sobre $\mathcal{W}_{\text{livre}}$. Agentes móveis planares (veículos terrestres em terrenos regulares ou aviões em atitude constante) pertencem ao grupo $SE(2)$. Já robôs em três dimensões (aeronaves ou plataformas subaquáticos) são representados em $SE(3)$.

Tendo como base o espaço de configurações, a maioria dos robôs reais está sujeita a limitações de deslocamento, as quais são normalmente separadas em dois tipos: de *igualdade* e de *desigualdade*. Restrições de igualdade estão ligadas a limitações no espaço de configurações do robô, muitas vezes restringindo sua movimentação. Já as restrições de desigualdade são fatores limitadores de determinadas variáveis (velocidades e acelerações) e são modeladas como saturações ou histereses de processo. Restrições de igualdade ainda podem ser classificadas em dois tipos mais comuns: as restrições *holonômicas* e as *não holonômicas*. Restrições holonômicas modificam as dimensões do espaço de configuração \mathcal{Q} , ao passo que as não holonômicas restringem a variação temporal das variáveis de configuração dos robôs.

3.1.3 Espaço de Estados

A terceira representação espacial, essa mais comum à área de Teoria de Controle, é o *espaço de estados* \mathcal{X} . Tal representação é utilizada para modelar o comportamento dinâmico de sistemas, de forma a incorporar forças e momentos que atuam sobre o robô. Nas situações em que apenas a cinemática é considerada, tem-se que $\mathcal{Q} \equiv \mathcal{X}$ (para robôs holonômicos). Porém, o mais comum é que $|\mathcal{X}| \gg |\mathcal{Q}|$ quando as acelerações do sistema são incorporadas ao modelo. Nesses casos, o espaço de estados de um robô é dado em função da composição simples de suas configurações e de suas variações temporais, $\mathbf{x} = [\mathbf{q} \ \dot{\mathbf{q}}]$, e das restrições holonômicas ou não holonômicas correspondentes.

Por se tratar de um espaço com número de dimensões tipicamente maior do que o de \mathcal{Q} , o espaço de estados permite representar o comportamento do robô real de maneira mais completa. Por outro lado, isso aumenta a complexidade das provas formais e de análise de estabilidade e de otimalidade de controle. Por essa razão, muitos dos trabalhos envolvendo principalmente planejamento de times de robôs (onde a dimensão do espaço depende do número de agentes) utilizam apenas o espaço de configurações, cuja representação é mais simples. A Figura 3.1 ilustra a relação entre os espaços de representação, formalmente descritos como $\mathcal{W} \subset \mathcal{Q} \subset \mathcal{X}$.

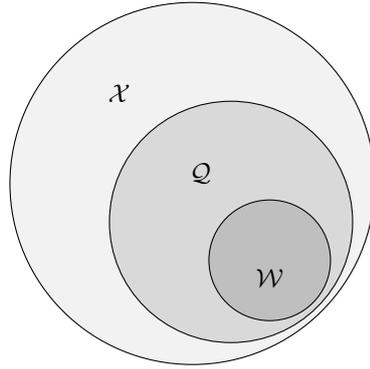


Figura 3.1. Relação entre os espaços de representação utilizados: \mathcal{W} é o espaço de trabalho do robô, \mathcal{Q} o espaço de configurações e \mathcal{X} o espaço de estados

3.2 Problema de Planejamento de Movimento

Nesta seção, é apresentada a formulação do problema de planejamento de movimento de um único robô que se desloca em um ambiente conhecido e povoado por obstáculos. Comece pela modelagem do veículo, descrita por meio da Equação 3.1:

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (3.1)$$

onde $\mathbf{x} \in \mathbb{R}^{n_x}$ e $\mathbf{u} \in \mathbb{R}^{n_u}$ correspondem respectivamente aos vetores de estados e de controle do robô, $\mathbf{w} \in \mathbb{R}^{n_w}$ é o vetor de perturbações e incertezas que corrompe o sistema e $\mathbf{x}_0 \in \mathbb{R}^{n_x}$ representa as condições iniciais do sistema em $t_0 = 0$, para t discreto. De maneira geral,

$$f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_x}$$

corresponde à função não linear que descreve a variação temporal dos estados do sistema.

É considerado ainda que

$$\mathbf{u}(t) = \kappa(\mathbf{r}(t), \mathbf{x}(t)) \quad (3.2)$$

é a ação de controle do sistema em malha fechada, onde $\mathbf{r} \in \mathbb{R}^{n_r}$ é o sinal de referência, e que

$$\kappa : \mathbb{R}^{n_r} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$$

é a lei de controle não linear que promove a estabilização do processo. Tipicamente, nos casos em que o veículo apresenta dinâmica complexa (com muitos estados),

$$n_x \gg n_r.$$

Por fim, \mathcal{W} é considerado como sendo o espaço de trabalho inicialmente conhecido, e $\mathcal{W}_{\text{livre}} \subset \mathcal{W}$ as posições livres de obstáculos nesse espaço. Então, o problema de planejamento de movimentos para um único agente, em malha fechada, é definido da seguinte forma:

Problema 3.1 (Planejamento de movimentos para um único robô). *Seja o robô R representado pelo vetor de estados $\mathbf{x} \in \mathcal{X}$ com projeção em \mathcal{W} , e seja $\mathcal{W}_{\text{livre}}$ o espaço de trabalho livre de obstáculos. Gerar sinais de referência \mathbf{r} que levem o robô do estado inicial $\mathbf{x}_{\text{ini}} = \mathbf{x}_0$ em $t_0 = 0$ até o final \mathbf{x}_{alvo} em $t_f > t_0$ finito, tal que a projeção de $\mathbf{x}(t)$ em $\mathcal{W} \in \mathcal{W}_{\text{livre}} \forall t \in (t_0, t_f]$.*

3.3 Árvore Aleatória de Exploração Rápida

Dentre as diversas técnicas de planejamento de movimento baseadas em amostragem aleatória, destaca-se a Árvore Aleatória de Exploração Rápida (*Rapidly-exploring Random Tree*, ou RRT). Trata-se de um algoritmo capaz de produzir árvores de caminhos, realizáveis por um robô específico, que crescem rapidamente através do ambiente conhecido. Por realizável, assume-se que as trajetórias respeitam todas as restrições do veículo e do espaço de trabalho.

RRTs têm sido estudadas na última década e vêm recebendo cada vez mais atenção em função das diversas vantagens que apresentam em relação a outras técnicas. Elas constituem aproximações simples e eficientes, capazes de planejar movimentos ao longo de ambientes complexos (com muitos obstáculos) de forma rápida. Permitem ainda lidar com modelos de sistemas lineares ou não lineares com grande número de graus de liberdade, incertezas nos parâmetros e diferentes tipos de restrições.

A seguir são apresentadas duas variações de algoritmos para RRTs. O primeiro, chamado de RRT de Malha Aberta (*Open Loop RRT*, ou OL-RRT), representa a abordagem mais simples e lida apenas com ambientes estáticos e conhecidos, sendo pouco robusto a incertezas e perturbações. O segundo, denominado RRT de Malha Fechada (*Closed Loop RRT*, ou CL-RRT), é capaz de garantir a navegação do robô em meios dinâmicos e com incerteza. Esse algoritmo também é chamado de RRT de Tempo Real, pois admite o planejamento de trajetórias simultaneamente ao processo de execução de movimento do veículo. Ambos os métodos assumem a existência de um único sistema (robô). No Capítulo 4 serão apresentadas as contribuições desta

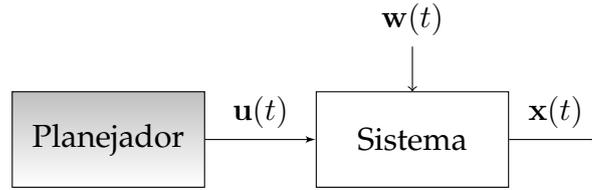


Figura 3.2. Modelo de predição em malha aberta para o Algoritmo 3.1: a trajetória planejada é aplicada diretamente à entrada do modelo.

pesquisa para possibilitar o uso dessas ferramentas em ambientes dinâmicos cooperativos com múltiplos robôs heterogêneos, sujeitos a restrições e a incertezas.

3.3.1 RRT de Malha Aberta

A RRT de Malha Aberta (ou simplesmente RRT) foi inicialmente apresentada em [LaValle & Kuffner, 1999]. Essa técnica é classificada como Mapa de Rotas Probabilístico (*Probabilistic Road Map*, ou PRM). Tais abordagens utilizam pontos aleatórios amostrados no espaço de estados x do robô para crescer árvores de trajetórias realizáveis em um ambiente de maneira incremental. Assim como acontece com outros métodos de planejamento contínuo baseados em amostragem, a RRT possui em seu cerne um módulo capaz de detectar colisões. Isso permite simplificar a representação dos espaços de trabalho, configurações e estados do problema.

Além do módulo de detecção de colisão, outra característica importante da OL-RRT é a existência de uma métrica, definida como ρ , utilizada para medir a distância entre dois estados do sistema. É com base nessa função,

$$\rho : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}^+,$$

que o algoritmo promove a expansão da árvore de exploração através do ambiente. Nos primeiros trabalhos na literatura envolvendo essa técnica, considerando apenas veículos holonômicos em espaços de duas ou três dimensões, a distância Euclidiana foi utilizada como métrica razoável, podendo ainda ser utilizada em outros casos mais complexos.

Para entender melhor o método, volte ao modelo da Equação 3.1, onde a evolução dos estados do sistema é explicada em função dos próprios x e da entrada u . Essa mesma representação pode ser expressa ainda de maneira gráfica pela Figura 3.2.

O modelo representa um processo em malha aberta, onde não há qualquer influência (realimentação) dos estados na ação de controle u . Isso torna o sistema

mais susceptível a perturbações w , por exemplo. Essa é ainda a maneira principal pela qual a RRT clássica aborda o problema. O objetivo é encontrar as entradas do processo, nesse caso do robô, de maneira a compor a árvore de exploração do ambiente, estrutura de dados descrita por \mathcal{T}_x . O conceito fica mais claro quando se analisa o Algoritmo 3.1, que mostra os passos básicos da OL-RRT.

O procedimento recebe como entrada dois parâmetros, $\mathbf{x}_{\text{ini}} = \mathbf{x}_0$ e \mathbf{x}_{alvo} . O primeiro representa o vetor inicial de estados do robô, enquanto o segundo corresponde ao vetor de estados alvo em t_f . Na linha 1, a árvore \mathcal{T}_x é inicializada com o vetor inicial de estados. A árvore geralmente é representada por um grafo sem ciclos, onde os vértices são estados \mathbf{x} e as arestas correspondem às trajetórias geradas pela ação de controle \mathbf{u} escolhidas. De forma mais intuitiva, ela pode ser modelada ainda como uma máquina de estados, onde para cada estado existe uma condição de transição (entrada de controle) que leva a outro.

Algoritmo 3.1 RRT de Malha Aberta

Recebe: estado inicial \mathbf{x}_{ini} e final \mathbf{x}_{alvo}

```

1:  $\mathcal{T}_x \leftarrow \mathbf{x}_{\text{ini}}$ 
2: enquanto  $\hat{\mathbf{x}}$  não alcançar  $\mathbf{x}_{\text{alvo}}$  faça
3:    $\mathbf{x}_{\text{rand}} \leftarrow$  escolhe amostra de  $\mathcal{X}$  compatível com  $\mathcal{W}_{\text{livre}}$ 
4:    $\mathbf{x}_{\text{prox}} \leftarrow$  encontra o vértice em  $\mathcal{T}_x$  que minimiza  $\rho(\mathbf{x}_{\text{prox}}, \mathbf{x}_{\text{rand}})$ 
5:    $\hat{\mathbf{x}} \leftarrow \mathbf{x}_{\text{prox}}$ 
6:   para  $\tau = t$  até  $(t+T)$  faça
7:      $\hat{\mathbf{u}}(\tau) \leftarrow$  seleciona entrada que minimiza  $\rho(\hat{\mathbf{x}}, \mathbf{x}_{\text{rand}})$  em  $\tau$ 
8:      $\hat{\mathbf{x}}(\tau+1) \leftarrow f(\hat{\mathbf{x}}(\tau), \hat{\mathbf{u}}(\tau))$ 
9:     se  $\hat{\mathbf{x}}(\tau) \notin \mathcal{W}_{\text{livre}}$  então
10:      interrompe laço
11:    fim se
12:  fim para
13:   $\mathcal{T}_x \leftarrow \mathcal{T}_x \cup \hat{\mathbf{x}} \text{ e } \hat{\mathbf{u}} \forall t \in (t, t+T]$ 
14: fim enquanto
15: retorna  $\mathcal{T}_x$ 

```

Uma representação gráfica de algumas execuções do algoritmo é mostrada na Figura 3.3. Nesse exemplo, \mathbf{x} apresenta duas dimensões (x e y) e o veículo considerado é modelado por um integrador simples, $\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{u}$, sem incertezas. O ambiente possui um conjunto \mathcal{O} com apenas um obstáculo.

O algoritmo é executado de maneira iterativa. Primeiramente, na linha 3, é escolhido um estado \mathbf{x}_{rand} em \mathcal{X} de forma aleatória e cuja projeção no espaço de trabalho pertença a $\mathcal{W}_{\text{livre}} \subset \mathcal{W}$. Esse estado indica o curso de propagação de $\hat{\mathbf{x}}$, o novo estado a ser incorporado à árvore. Tipicamente \mathbf{x}_{alvo} é escolhido como o estado \mathbf{x}_{rand}

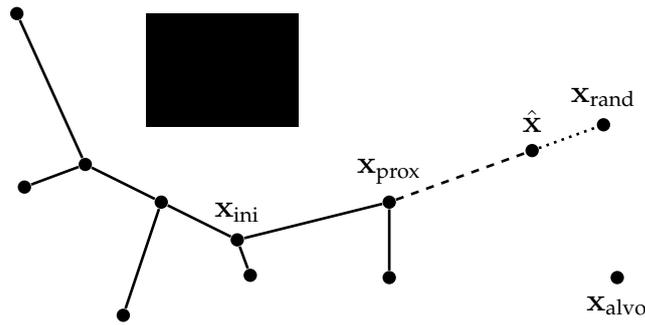


Figura 3.3. Expansão da árvore \mathcal{T}_x da RRT de Malha Aberta: o algoritmo inicia a partir de x_{ini} e itera até alcançar x_{alvo} , evitando todos os obstáculos contidos em \mathcal{O} . A cada estado aleatório x_{rand} escolhido, o estado em \mathcal{T}_x mais próximo x_{prox} é calculado. O modelo é propagado até \hat{x} , incorporado à árvore.

com alguma probabilidade, de modo que a expansão da árvore tenda em direção ao alvo. Em geral, isso faz com que o algoritmo convirja mais rapidamente para o resultado.

Posteriormente é escolhido um vértice em \mathcal{T}_x a partir do qual a árvore se expandirá na direção de x_{rand} (Figura 3.3). Esse vértice é chamado de x_{prox} e é selecionado por meio de minimização com base na métrica ρ , conforme apresentado na linha 4.

Na sequência, é realizada a predição em tempo finito T , a partir do instante corrente t , onde é determinada a trajetória entre um nó da árvore e o novo vértice. Para cada instante de tempo τ , é selecionada uma entrada \hat{u} que expandirá \mathcal{T}_x a partir de x_{prox} em direção a x_{rand} . Há diversas maneiras de se escolher essa ação de controle como, por exemplo, seleção aleatória por meio de cinemática (ou dinâmica) inversa. Porém, a forma mais utilizada é por meio de tabelas de pesquisa (*lookup tables*) [LaValle & Kuffner, 1999, 2001].

Finalmente o novo estado \hat{x} é calculado por intermédio de predição, com base na Equação 3.1, onde são desconsideradas as perturbações (linha 8). Esse é um dos motivos que tornam a OL-RRT sensível a incertezas e interferências externas. Os vetores \hat{x} e \hat{u} (esse último posteriormente enviado ao robô real), além da trajetória em cada instante de tempo τ , são incorporados à árvore a cada nova interação do algoritmo. Esse para quando \hat{x} atinge x_{alvo} (linha 2). Em função das incertezas que corrompem o sistema, \hat{x} provavelmente nunca será igual a x_{alvo} , de modo que o termo “atingir” deve ser entendido como sendo alcançar uma região em torno do objetivo.

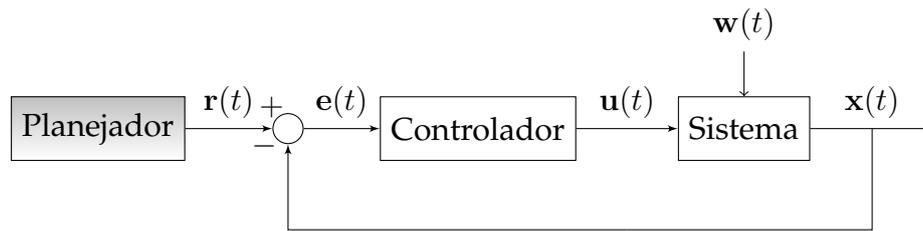


Figura 3.4. Modelo de previsão do Algoritmo da RRT de Malha Fechada: a trajetória planejada é aplicada como comandos de referência do sistema de controle, que se encarrega de gerar ações para o robô.

3.3.2 RRT de Malha Fechada

A OL-RRT, apresentada anteriormente, possui como base a suposição de que o ambiente é conhecido. Em outras palavras, dada qualquer posição x no espaço de estados do robô, existe uma *função* que determina se essa posição leva ou não a colisões com os obstáculos em \mathcal{O} , e tal função não muda ao longo do tempo. Esse tipo de ambiente de navegação é chamado de *ambiente estático*.

No entanto o mundo real é mais complexo. Além da dificuldade em se determinar o espaço de trabalho livre ($\mathcal{W}_{\text{livre}}$) do ambiente, esse pode variar substancialmente ao longo do tempo. Incluindo a possibilidade do surgimento de novos obstáculos e a mudança nas configurações do meio (portas e janelas antes abertas e agora fechadas), a própria existência de outros agentes (pessoas e robôs) faz com que o ambiente seja *dinâmico*.

Por esse motivo, a RRT de Malha Fechada (*Closed Loop RRT*, ou CL-RRT) foi introduzida em [Kuwata et al., 2009], como alternativa à RRT de Malha Aberta. Como o próprio nome diz, em vez de efetuar o planejamento escolhendo ações de controle u diretamente para o veículo, a CL-RRT delega essa função a um controlador. Esse recebe como entrada uma trajetória, possivelmente não realizável, e a transforma em comandos executáveis pelo robô. Uma análise da Figura 3.4 comparada à Figura 3.2 é útil para melhor compreensão da técnica.

A Figura 3.4 apresenta um típico modelo em malha fechada, onde r representa a referência de controle, aplicada à entrada do sistema. Além disso, existe uma realimentação dos estados do processo. De maneira genérica, a ação de controle agora passa a ser regida apenas em função da referência e da medição dos estados atuais, conforme descrito na Equação 3.2. Tipicamente nos casos em que o veículo apresenta dinâmica complexa, $n_x \gg n_r$. Em geral, isso facilita o processo de planejamento, pois são calculados apenas os sinais de referência para o robô, enquanto que a tarefa de tornar a trajetória realizável é deixada a cargo do controlador.

A CL-RRT basicamente é composta de duas partes, chamadas de Laço de Expansão e Laço de Execução. A primeira é responsável pelo planejamento em si e promove a exploração do ambiente de forma semelhante ao descrito no Algoritmo 3.1. Já na segunda, é escolhida uma parte do plano corrente para ser seguida no instante de tempo em que o robô executa a tarefa.

3.3.2.1 Laço de Expansão

Basicamente a etapa de expansão da CL-RRT, mostrada no Algoritmo 3.2, é semelhante ao Algoritmo 3.1, com algumas pequenas modificações. O objetivo aqui é o mesmo de antes: expandir a árvore de estados do robô através do ambiente de maneira aleatória. \mathcal{T}_x representa essa árvore (assim como na RRT de Malha Aberta), mas agora descreve as trajetórias do sistema em malha fechada. Além disso, uma nova estrutura é acrescentada: a árvore de sinais de referência, usados como entrada pelo controlador e simbolizada por \mathcal{T}_r . As entradas desse algoritmo são as árvores (as quais são retornadas como saída), o estado alvo \mathbf{x}_{alvo} e o tempo corrente t da interação em tempo real.

Como no caso anterior, a CL-RRT também possui um módulo de detecção de colisão e uma métrica ρ usada para medir a distância entre vetores no espaço. A diferença agora é que os estados do robô não são mais utilizados como entradas para a métrica, mas sim os sinais de referência apresentados ao controlador. Dessa forma, a função é redefinida como sendo:

$$\rho : \mathbb{R}^{n_r} \times \mathbb{R}^{n_r} \rightarrow \mathbb{R}.$$

Isso permite o uso de métricas mais simples para estabelecer a distância entre nós da árvore no espaço de estados. Como será apresentado adiante, CL-RRT realiza o planejamento no espaço de referência, uma simplificação do espaço de estados proporcionada pela lei de controle da Equação 3.2.

Inicialmente é selecionada a condição aleatória \mathbf{r}_{rand} para o crescimento de \mathcal{T}_r , conforme apresentado na linha 1 do Algoritmo 3.2. É importante que essa referência seja compatível com $\mathcal{W}_{\text{livre}}$, do contrário, o controlador do robô pode levá-lo a um estado de colisão.

Na sequência é escolhido o vértice \mathbf{r}_{prox} em \mathcal{T}_r mais próximo a \mathbf{r}_{rand} , segundo a métrica ρ . O novo nó $\hat{\mathbf{r}}$ é propagado em direção a \mathbf{r}_{rand} , também livre de colisões. Se considerarmos as duas árvores do método como conjuntos compostos por v vértices,

Algoritmo 3.2 RRT de Malha Fechada (Laço de Expansão)

Recebe: árvores \mathcal{T}_r e \mathcal{T}_x , alvo \mathbf{x}_{alvo} , tempo corrente t

- 1: $\mathbf{r}_{\text{rand}} \leftarrow$ escolhe amostra de referência compatível com $\mathcal{W}_{\text{livre}}$
- 2: $\mathbf{r}_{\text{prox}} \leftarrow$ encontra o vértice em \mathcal{T}_r que minimiza $\rho(\mathbf{r}_{\text{prox}}, \mathbf{r}_{\text{rand}})$
- 3: $\hat{\mathbf{r}} \leftarrow$ propaga $\mathbf{r}_{\text{prox}} \in \mathcal{W}_{\text{livre}}$ em direção a \mathbf{r}_{rand}
- 4: $\hat{\mathbf{x}} \leftarrow$ captura o vértice em \mathcal{T}_x correspondente a \mathbf{r}_{prox} em \mathcal{T}_r
- 5: **para** $\tau = t$ **até** $(t+T)$ **faça**
- 6: $\hat{\mathbf{u}}(\tau) \leftarrow \kappa(\hat{\mathbf{r}}, \hat{\mathbf{x}}(\tau))$
- 7: $\hat{\mathbf{x}}(\tau+1) \leftarrow f(\hat{\mathbf{x}}(\tau), \hat{\mathbf{u}}(\tau), 0)$
- 8: **se** $\hat{\mathbf{x}}(\tau+1)$ é realizável **então**
- 9: $\mathcal{T}_x \leftarrow \mathcal{T}_x \cup \hat{\mathbf{x}}(\tau+1)$
- 10: **fim se**
- 11: **se** $\hat{\mathbf{x}}(\tau+1)$ alcançou \mathbf{x}_{alvo} **então**
- 12: **interrompe laço**
- 13: **fim se**
- 14: **fim para**
- 15: $\mathcal{T}_r \leftarrow \mathcal{T}_r \cup \hat{\mathbf{r}}$
- 16: **retorna** \mathcal{T}_r e \mathcal{T}_x

relacionados entre si, então existe uma função bijetora

$$b : \mathcal{T}_r \rightarrow \mathcal{T}_x,$$

que associa valores de referência a estados em cada instante de tempo t . A partir de \mathbf{r}_{prox} , o vértice correspondente \mathbf{x}_{prox} em \mathcal{T}_x é capturado por meio dessa função bijetora. Na linha 4, \mathbf{x}_{prox} se torna condição inicial para a propagação de $\hat{\mathbf{x}}$. Esse estado é usado para propagar o modelo do robô através do ambiente com base na lei de controle da linha 6. Um novo nó $\hat{\mathbf{x}}$ é criado (linha 7) e incorporado à árvore.

Por fim, a Figura 3.5 mostra um exemplo de execução do laço de expansão do algoritmo da CL-RRT para o caso bidimensional simples. Nesse contexto, \mathcal{T}_r é a árvore apresentada em cinza, enquanto que \mathcal{T}_x está representada em preto.

3.3.2.2 Laço de Execução

A etapa de execução da CL-RRT tem por objetivo promover o seguimento de parte do plano desenvolvido até o momento, escolhendo o “melhor” caminho possível dentro da árvore \mathcal{T}_r que conduza o robô da posição corrente até \mathbf{x}_{alvo} . O Algoritmo 3.3 apresenta essa parte do método, o qual (como no caso da OL-RRT) recebe como entrada apenas a condição inicial \mathbf{x}_{ini} e a região alvo \mathbf{x}_{alvo} .

Nesse algoritmo, existe um laço de tempo real executado até que o estado real

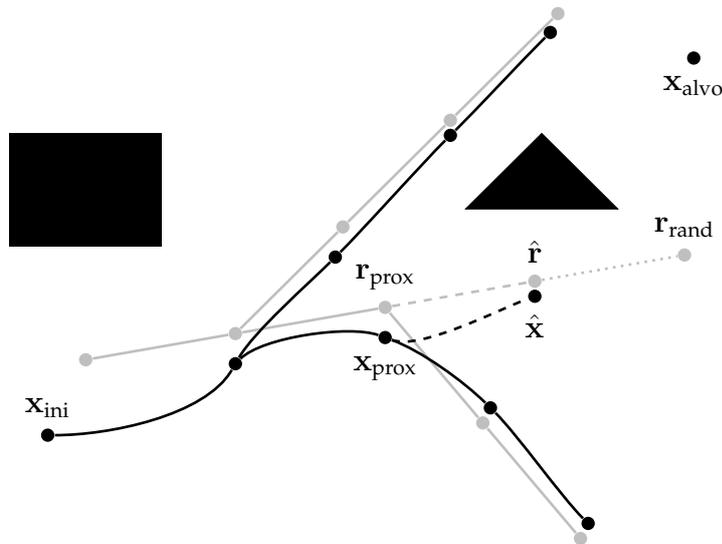


Figura 3.5. Laço de expansão das árvores \mathcal{T}_r (em cinza) e \mathcal{T}_x (em preto) na RRT de Malha Fechada: o algoritmo inicia a partir de x_{ini} e itera até alcançar x_{alvo} , evitando todos os obstáculos contidos em \mathcal{O} . Aqui, nós pretos representam vetores no espaço de estados, enquanto nós cinzas são vetores no espaço de referência. A cada estado de referência de controle r_{rand} escolhida, a \hat{r} mais próxima a r_{prox} é calculada e o modelo do robô é propagado pelo controlador até \hat{x} , que será incorporado à árvore \mathcal{T}_x .

do veículo $x(t)$ atinja x_{alvo} em t finito. O primeiro passo dentro desse laço é expandir as árvores de referência e de estados do robô durante certo período de tempo ΔT , por meio do algoritmo anterior. Uma vez que as árvores tenham sido expandidas, a próxima ação é obter as informações do robô para determinar seu estado corrente e atualizar (mapear) o espaço de trabalho livre, caso necessário. Essa última parte pode ser importante caso a técnica seja utilizada para a navegação em ambientes dinâmicos e parcial ou completamente desconhecidos. Um agente equipado com sensores laser, por exemplo, pode simultaneamente reconstruir ou identificar um ambiente à medida que planeja seu movimento.

Com essas novas informações, é calculado o melhor caminho r^* em \mathcal{T}_r a ser aplicado ao controlador, conforme visto em vermelho na Figura 3.6. O robô real segue a referência, realizando uma trajetória durante o intervalo ΔT , representada pela linha azul pontilhada, relativo à próxima iteração do algoritmo, onde as árvores serão novamente expandidas.

Caso não exista um melhor caminho a ser seguido (linha 8 do Algoritmo 3.3) o sistema toma uma ação segura. Exemplos podem ser parar o robô, caso seja possível, ou realizar manobras de segurança. Uma vez que r^* é determinado como não vazio, é preciso repropagar (calcular novamente) a trajetória do robô real a partir de $x(t)$ e

Algoritmo 3.3 RRT de Malha Fechada (Laço de Execução)

Recebe: estado inicial \mathbf{x}_{ini} e final \mathbf{x}_{alvo}

- 1: $t \leftarrow 0, \mathcal{T}_x \leftarrow \mathbf{x}_{\text{ini}}, \mathcal{T}_r \leftarrow \emptyset$
- 2: **enquanto** $\mathbf{x}(t)$ não alcançar \mathbf{x}_{alvo} **faça**
- 3: **enquanto** restar tempo em ΔT **faça**
- 4: expande \mathcal{T}_r e \mathcal{T}_x (Algoritmo 3.2)
- 5: **fim enquanto**
- 6: medir $\mathbf{x}(t)$ e atualizar $\mathcal{W}_{\text{livre}}(t)$
- 7: estima o melhor caminho corrente \mathbf{r}^* em \mathcal{T}_r
- 8: **se** $\mathbf{r}^* = \emptyset$ **então**
- 9: aplica ação de segurança e vai para linha 3
- 10: **fim se**
- 11: **se** [\mathbf{r}^* repropagado de $\mathbf{x}(t)$] $\notin \mathcal{W}_{\text{livre}}(t)$ **então**
- 12: remove partes não realizáveis de \mathbf{r}^*
- 13: **fim se**
- 14: aplica \mathbf{r}^* ao controlador do robô
- 15: $t \leftarrow t + \Delta T$
- 16: **fim enquanto**

remover as partes do caminho que o levem a colidir com os obstáculos. Finalmente o \mathbf{r}^* , livre de colisões, é aplicado ao controlador do sistema, até que as árvores sejam expandidas novamente no novo ciclo de planejamento.

3.4 Abstração

Uma abstração pode ser definida como o vetor \mathbf{a} de uma curva paramétrica, tal que exista uma função ϕ capaz de mapear n vetores no espaço de configurações \mathcal{Q} para essa curva. Em outras palavras:

$$\phi(\mathbf{q}^n) = \mathbf{a}, \quad \phi: \mathcal{Q}^n \rightarrow \mathcal{A}, \quad (3.3)$$

onde \mathcal{A} representa o espaço de abstração, cuja dimensionalidade independe do número de robôs do time. Além disso, $|\mathcal{A}| \ll |\mathcal{Q}^n|$.

Dessa forma, é possível encontrar leis de controle:

$$\mathbf{u}_i(t) = \kappa(\mathbf{a}(t), \mathbf{q}_i(t)) \quad (3.4)$$

para agentes do time modelados em \mathcal{Q} , tal que essa ação dependa apenas da propagação da abstração em \mathcal{W} e da medição da configuração do próprio robô.

O primeiro trabalho a usar esse tipo de modelagem na coordenação de grupos

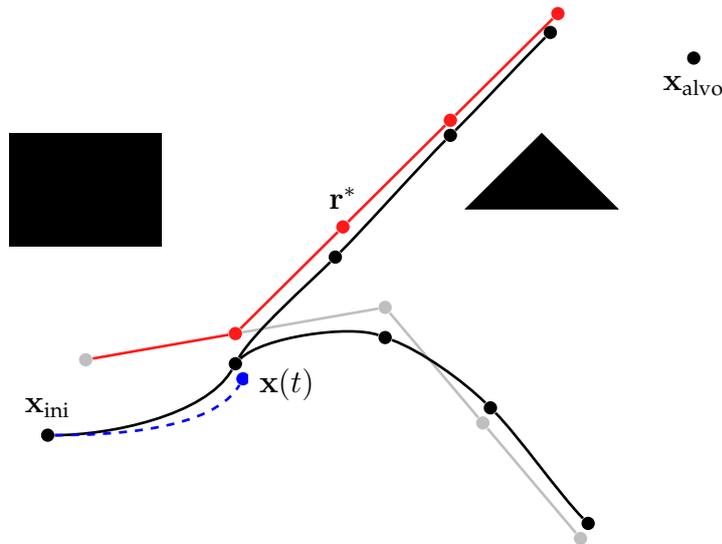


Figura 3.6. Laço de execução da RRT de Malha Fechada: a cada instante de tempo real, as referências do melhor caminho \mathbf{r}^* na árvore \mathcal{T}_r corrente são escolhidas (em vermelho), de modo a minimizar a distância entre o robô e \mathbf{x}_{alvo} . Então, o veículo real segue esse plano (pontilhado em azul) atualizado.

de robôs móveis foi [Belta & Kumar, 2004]. Utilizando curvas paramétricas simples (retângulos e elipses), os autores constroem leis de controle descentralizadas, dependentes apenas da pose de cada robô e de uma variedade abstrata, \mathcal{a} , calculada por um planejador centralizado. A ideia básica é apresentada na Figura 3.7.

Michael et al. [2006] expandem a abordagem para grupos de robôs holonômicos em espaços \mathbb{R}^3 , englobando robôs aéreos ao problema de planejamento. Já o trabalho [Michael & Kumar, 2009] trata o problema de colisão entre agentes dentro da abstração e apresenta controladores que garantidamente convergem para \mathbf{a}_r .

Tal abordagem é interessante, porque reduz o espaço de planejamento do grupo de robôs para outro menor (\mathcal{A}) e conseqüentemente a complexidade do problema. Além disso, diminui a quantidade de informação que trafega pela rede de comunicação. Cada robô, sendo capaz de medir sua própria configuração, deve conhecer apenas a abstração formada pelo grupo e a abstração desejada.

Como \mathcal{a} é matematicamente representada por uma curva paramétrica, a cada robô R_i é associado um parâmetro, tal que

$$\mathbf{a}_r(i) \rightarrow \mathbf{r}_i : \mathcal{A} \rightarrow \mathbb{R}^{n_r}.$$

Em outras palavras, dada a abstração de referência, cada robô pode calcular seu objetivo local no espaço de trabalho.

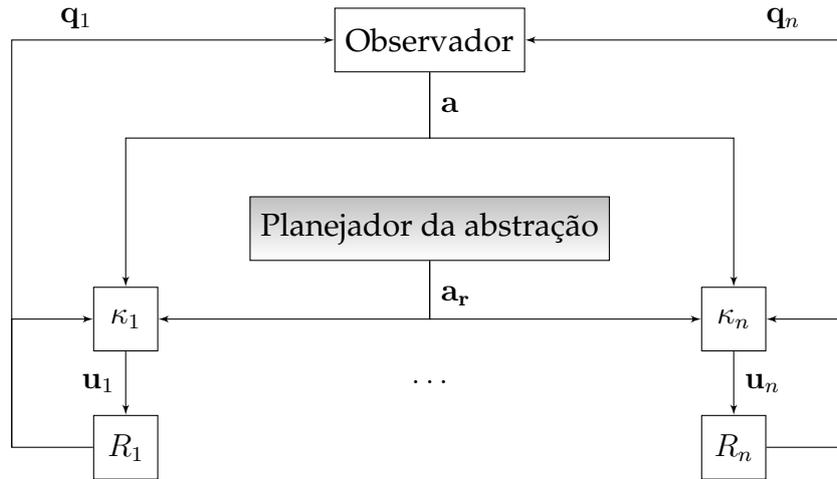


Figura 3.7. Arquitetura de controle e comunicação apresentada em [Belta & Kumar, 2004]: um planejador centralizado gera a trajetória \mathbf{a}_r da abstração e a envia aos controladores locais de cada robô R_i . Esses dependem apenas de suas próprias configurações e da forma da abstração real produzida pelo grupo de robôs, medida por um observador global.

Entretanto, essa estratégia depende de um observador global (uma câmera filmando os veículos, por exemplo) e de um planejador centralizado. Outra desvantagem é que o uso de abstrações se baseia em leis de controle de difícil generalização, restritas ao espaço de configurações e a robôs holonômicos representados em (no máximo) $SE(3)$. Belta & Kumar [2004] demonstram que a convergência de tais controladores está sujeita à existência da matriz Jacobiana $\delta\phi$, tal que:

$$\dot{\mathbf{q}}^n(t) = \delta\phi^T (\delta\phi\delta\phi^T)^{-1} \dot{\mathbf{a}}(t), \quad (3.5)$$

o que não acontece em todos os casos.

Além disso, a abstração escolhida deve ser decomposta em componentes de configuração e forma:

$$\mathbf{a} = \langle \mathbf{g}, \mathbf{s} \rangle, \quad \mathcal{A} = \mathcal{G} \times \mathcal{S}, \quad \phi = \langle \phi_{\mathbf{g}}, \phi_{\mathbf{s}} \rangle \quad (3.6)$$

onde \mathbf{g} é a pose em $\mathcal{G} \equiv SE(2)$ ou $SE(3)$ e \mathbf{s} é a forma da abstração, representada em um espaço \mathcal{S} . Isso implica na decomposição de ϕ em um mapeamento de pose ($\phi_{\mathbf{g}}$) e em um mapeamento geométrico ($\phi_{\mathbf{s}}$), de modo que ambos possuam pseudoinversa (Equação 3.5). Isso restringe o uso de abstrações a curvas com significado físico, para que essas sejam modeladas pelo planejador.

No próximo capítulo, é apresentado um método para o planejamento de

abstrações totalmente descentralizado e independente de um observador externo. Além disso, nossa estratégia permite o uso de qualquer curva paramétrica como abstração, sem a necessidade de que essa possua qualquer sentido físico. A metodologia é estendida ao espaço de estados, sem que seja necessário derivar lógicas de controle específicas para cada tipo de robô, uma vez que sejam conhecidos seus modelos matemáticos.

Capítulo 4

Metodologia

Frodo: “Queria que o anel nunca tivesse sido dado a mim e que nada disso tivesse acontecido.”

Gandalf: “Assim como eu e assim como todos que testemunham tempos como este. Mas não cabe a nós decidir isso. Tudo o que nos cabe é decidir o que fazer com o tempo que nos é dado.”

O Senhor dos Anéis, J. R. R. Tolkien

Este capítulo apresenta um arcabouço para o planejamento cooperativo de tarefas para times de robôs heterogêneos em ambientes com obstáculos. A metodologia é baseada em um conjunto de algoritmos pseudoaleatórios que permitem a exploração do ambiente em tempo real, enquanto levam o grupo de agentes à execução de tarefas de forma coordenada.

4.1 Definição do Problema

Considere um time \mathcal{R} , composto por n robôs heterogêneos que compartilham o mesmo ambiente \mathcal{W} . A Equação 4.1, uma generalização da Equação 3.1, representa o modelo dinâmico do comportamento desse time ao longo do tempo:

$$\mathbf{X}(t+1) = F(\mathbf{X}(t), \mathbf{U}(t), \mathbf{W}(t)), \quad \mathbf{X}(0) = \mathbf{X}_0, \quad (4.1)$$

onde

$$\mathbf{X}(t) = \left[\mathbf{x}_1^T(t) \quad \mathbf{x}_2^T(t) \quad \cdots \quad \mathbf{x}_n^T(t) \right]^T,$$

o vetor de estados que descreve o grupo, é a concatenação simples dos vetores de estados de cada robô R_i que compõem \mathcal{R} , sendo

$$\mathbf{X}_0 = \left[\mathbf{x}_1^T(0) \quad \mathbf{x}_2^T(0) \quad \cdots \quad \mathbf{x}_n^T(0) \right]^T$$

as condições iniciais do sistema.

Além disso, seja

$$\mathbf{U}(t) = \left[\mathbf{u}_1^T(t) \quad \mathbf{u}_2^T(t) \quad \cdots \quad \mathbf{u}_n^T(t) \right]^T,$$

o vetor de entradas do grupo, modelado como a concatenação das entradas de cada robô do time, e

$$\mathbf{W}(t) = \left[\mathbf{w}_1^T(t) \quad \mathbf{w}_2^T(t) \quad \cdots \quad \mathbf{w}_n^T(t) \right]^T,$$

o conjunto de perturbações que corrompem todo o sistema. Esse último é modelado como um vetor, pois se assume nesse caso a inexistência de correlação entre as perturbações que afetam os robôs do time.

De maneira equivalente,

$$F : \mathbb{R}^{\sum n_x^i} \times \mathbb{R}^{\sum n_u^i} \times \mathbb{R}^{\sum n_w^i} \rightarrow \mathbb{R}^{\sum n_x^i}$$

como sendo a função não linear que descreve a variação temporal de \mathcal{R} através de \mathcal{W} . Essa equação, porém, não pode ser definida pela concatenação das funções f de cada veículo R_i . Além das restrições de movimento do próprio robô, é necessário também considerar as limitações de navegação devido a colisões entre robôs do grupo. Tais restrições de colisão são modeladas normalmente por meio de desigualdades como as da Equação 4.2:

$$g(\mathbf{x}_i, \mathbf{x}_j) \leq 0, \quad \forall i, j \in 1 \dots n, \quad i \neq j \quad (4.2)$$

onde

$$g : \mathbb{R}^{n_x^i} \times \mathbb{R}^{n_x^j} \rightarrow \mathbb{R}$$

é a função que define as condições de colisão entre os agentes R_i e R_j do time. Tal função depende das características geométricas e dos níveis de incerteza no sensoriamento de cada veículo e, por isso, podem existir inúmeras funções diferentes em times de robôs heterogêneos.

Considerando essa e outras características (como a estrutura da rede de comunicação, por exemplo) é possível estabelecer um paralelo com a Equação 3.2, a fim

de se propor uma lei de controle para o sistema multirrobo, tal que

$$\mathbf{U}(t) = K(\mathbf{R}(t), \mathbf{X}(t)), \quad (4.3)$$

onde a ação de controle do sistema é função dos estados de todos os robôs do grupo e do sinal de referência

$$\mathbf{R}(t) = \left[\mathbf{r}_1^T(t) \ \mathbf{r}_2^T(t) \ \cdots \ \mathbf{r}_n^T(t) \right]^T,$$

e

$$K : \mathbb{R}^{\sum n_r^i} \times \mathbb{R}^{\sum n_x^i} \rightarrow \mathbb{R}^{\sum n_u^i}$$

é a lei de controle não linear e descentralizada para grupos de robôs.

Assim, o problema de planejamento de movimentos para sistemas multirrobo pode ser formalizado da seguinte maneira:

Problema 4.1 (Problema de planejamento cooperativo descentralizado). *Considere um time \mathcal{R} de n robôs heterogêneos, envolvidos de forma cooperativa na realização de uma tarefa. Cada robô R_i é representado por um vetor \mathbf{x}_i no espaço de estados \mathcal{X} . Seja, ainda, \mathcal{W} o espaço de trabalho e $\mathcal{W}_{livre} \subset \mathcal{W}$ as regiões livres de obstáculos nesse espaço. Considere, ainda, a existência de incertezas no vetor \mathbf{x}_i de cada robô e a variação de \mathcal{W}_{livre} em função do tempo. Calcule sinais de referência \mathbf{R} que conduzam o time, de maneira coordenada e livre de colisões, do estado inicial $\mathbf{X}_{ini} = \mathbf{X}_0$ em $t_0 = 0$ até o estado final α_{alvo} em $t_f > t_0$ finito, tal que a projeção de $\mathbf{X}(t) \in \mathcal{W}_{livre}(t) \ \forall t \in (t_0, t_f]$.*

Implícita na definição desse problema está a questão sobre o quanto cada robô conhece a respeito dos demais componentes do time. Para muitos estudiosos da área, se cada agente souber da existência dos demais, conhecendo inclusive seus modelos dinâmicos, leis de controle e condições iniciais, isso pode descaracterizar a definição de um sistema descentralizado. Em outras palavras, se todos os robôs tiverem conhecimento do modelo da Equação 4.1, não se tem um problema de planejamento completamente descentralizado.

Na sequência, é apresentado um método de planejamento de movimento para múltiplos robôs baseado na propagação da referência \mathbf{R} através de um ambiente conhecido. A técnica é inspirada no Algoritmo CL-RRT, apresentado na Seção 3.3. Diferentemente do método descrito em [Desaraju & How, 2011], igualmente baseado na exploração por meio de árvores aleatórias, nossa abordagem é totalmente descentralizada, dispensando inclusive a informação da posição inicial dos agentes.

4.2 Suposições

Esta seção apresenta as suposições necessárias ao desenvolvimento da metodologia descrita na Seção 4.3.

Suposição 4.1. *Cada robô em \mathcal{R} conhece seu próprio comportamento dinâmico e condições iniciais (Equação 3.1), mas desconhece o modelo de outros agentes (Equação 4.1).*

Cada robô do time precisará conhecer seu modelo dinâmico, em malha fechada, para fazer uso da extensão do Algoritmo CL-RRT, de forma a planejar sua própria trajetória. Embora a existência de um modelo preciso melhore a qualidade da solução calculada [Kuwata et al., 2009], modelos mais simples (mesmo cinemáticos) ainda funcionam bem. Isso é garantido basicamente pela natureza em malha fechada desse método de planejamento.

Suposição 4.2. *As restrições de colisão entre agentes são simétricas entre todos os robôs do time.*

Essa restrição nos diz que, dado o vetor de estados de um robô R_i , se ele respeita todas as restrições de colisão em relação a um robô R_j , então R_j também as respeita em relação a R_i . Em outras palavras, $g(\mathbf{x}_i, \mathbf{x}_j) = g(\mathbf{x}_j, \mathbf{x}_i) \forall i, j \in 1..n, i \neq j$. Se g representar uma função de distância, por exemplo, os agentes devem manter uma distância mínima entre si.

Suposição 4.3. *A rede de comunicação formada pelos robôs do time é conectada. Além disso, a comunicação acontece sem perdas e não há atrasos significativos na transmissão da informação.*

A questão da conectividade e da qualidade da rede é algo fundamental para qualquer método aplicado ao problema de sistemas descentralizados. Essa é considerada uma suposição razoável em muitos trabalhos [Dunbar, 2002; Aoude et al., 2010; Vahrenkamp et al., 2010], especialmente no que tange à operação em ambientes internos, relativamente pequenos e sem muitos obstáculos. Já ambientes externos costumam ser maiores do que o raio de comunicação dos robôs, e a conectividade nem sempre pode ser garantida. Porém, mesmo nesse caso, sempre é possível garantir que haverá comunicação suficiente para resolver o problema de colisão entre agentes da Equação 4.2. A não satisfação dessa restrição implica na perda de sincronismo e falha do presente método. Outras características do método que envolva o tratamento do problema frente à existência de sub-redes de comunicação

são consideradas fora do escopo deste trabalho, e será considerado como passos futuros.

Suposição 4.4. *Todos os robôs do time têm acesso a um mesmo mapa do ambiente (no mesmo referencial), que possui dimensões finitas e pode ser composto por obstáculos estáticos ou dinâmicos.*

Essa é uma suposição fundamental no uso de qualquer algoritmo de planejamento baseado em Mapas de Rotas Probabilísticos [LaValle, 2006]. O mapa do ambiente é a componente principal do *Módulo de Detecção de Colisão* e o que torna tais abordagens tão vantajosas para o planejamento em tempo real. Nosso método assume que há consistência na informação que todos os robôs possuem desse módulo de colisão. No caso mais simples em que o ambiente é estático e conhecido, não ocorrem maiores problemas. A dificuldade está em garantir tal consistência quando os robôs não conhecem o ambiente (ou quando esse muda de maneira imprevisível). Nesse caso, é esperado que os robôs apresentem a capacidade de construir o mapa dinamicamente de maneira descentralizada.

4.3 Árvore Pseudoaleatória de Exploração Rápida Multiagente

Nesta seção, é introduzida a Árvore Pseudoaleatória de Exploração Rápida Multiagente (*Multiple Rapidly-exploring Pseudorandom Trees*, ou MRPT), uma adaptação da RRT para o caso multirrobô. Analisando o método originalmente proposto em [LaValle & Kuffner, 1999], é possível perceber que a única característica que torna a RRT um algoritmo aleatório é o sorteio de uma amostra do ambiente $\mathcal{W}_{\text{livre}}$, a qual é utilizada para a expansão da árvore de exploração, seja em malha aberta ou seja em malha fechada.

Atualmente existe uma grande variedade de hipóteses e heurísticas diferentes para guiar a escolha aleatória em algoritmos do tipo PRM. Porém, todas elas podem ser agrupadas em basicamente duas categorias: amostragem pseudoaleatória (*pseudo-random sampling*) e amostragem quasealeatória (*quasi-random sampling*). Ambas as abordagens representam funções determinísticas que simulam randomicidade. Mas enquanto números pseudoaleatórios constituem sequências “realmente” aleatórias, números quasealeatórios (também conhecido como sequências de baixa discrepância) são sequências de números totalmente determinísticas, escolhidas a partir de alguma distribuição de probabilidade conhecida [LaValle, 2006].

Devido a essa característica de baixa discrepância, sequências quasealeatórias superam os algoritmos pseudoaleatórios em desempenho no problema de planejamento de movimentos. Entretanto, em problemas com um grande número de dimensões, como o caso abordado neste trabalho, a diferença de desempenho desses dois tipos de abordagens é praticamente desprezível [LaValle et al., 2004].

Essa natureza aleatória das técnicas de PRMs é irrelevante para o planejamento de movimentos de agentes únicos, conforme usada na grande maioria dos trabalhos relacionados, mas pode acarretar sérios problemas se utilizadas em sistemas multirrobôs descentralizados e cooperativos. Quando vários robôs utilizam técnicas de planejamento aleatórias para navegar em um ambiente \mathcal{W} comum, é difícil utilizar qualquer tipo de conhecimento *a priori*, sobre o grupo e sobre o ambiente, para prever o comportamento do time e, por exemplo, evitar colisões entre agentes. Assim, é necessário o uso intensivo de comunicação entre os robôs, os quais devem propagar seus planos para os demais agentes toda vez que ocorrer alguma alteração [Frazzoli, 2003; Desaraju & How, 2011]. Isso reduz substancialmente a capacidade de replanejamento do time, além de provocar situações que levam a falhas devido à falta de comunicação temporária, já que esses métodos são fortemente dependentes de uma rede completamente conectada.

Agora, suponha a situação em que seja usada uma função determinística [Frazzoli, 2003; LaValle, 2006] (quasealeatória ou pseudoaleatória) para gerar amostras de posições do ambiente no procedimento clássico da RRT (Algoritmo 3.1) ou qualquer outro método do tipo PRM. Por razões de simplicidade e generalidade, essa função será chamada de *função pseudoaleatória* daqui para frente, a qual pode ser descrita como:

$$p_{\text{rand}} : \mathbb{N} \rightarrow \mathbb{R}^m, \quad (4.4)$$

sendo m , a dimensão de um vetor qualquer, escolhido arbitrariamente. Por exemplo, é possível utilizar a função pseudoaleatória para gerar uma amostra aleatória do espaço de referência de controle:

$$\mathbf{r} = p_{\text{rand}}(k), \quad \forall k = 1, 2, \dots, \quad p_{\text{rand}} : \mathbb{N} \rightarrow \mathbb{R}^{n_r}$$

onde k é uma sequência numérica natural. Dada essa sequência, a função gera pontos no espaço \mathbb{R}^{n_r} , em malha fechada, para um único robô.

Suponha agora que tal função seja compartilhada entre todos os agentes do time \mathcal{R} e que todos executem a mesma versão do algoritmo de planejamento. Baseado nas suposições descritas na seção anterior, é possível formular o seguinte coro-

lário:

Corolário 4.1. *Considerando as condições impostas pelas suposições anteriores e desconsiderando os possíveis erros devido a precisão numérica nos dispositivos computacionais de cada membro do time \mathcal{R} , é possível então assumir que cada robô seja capaz de prever o comportamento de todo o sistema multirrobô em malha fechada, descrito por meio do sinal de referência \mathbf{R} , de maneira descentralizada e com incerteza limitada.*

O Corolário 4.1 diz que, se uma sequência aleatória conhecida entre todos os agentes do time for utilizada, o uso de uma mesma técnica baseada em planejamento aleatório (como a RRT, por exemplo) gerará o mesmo plano de movimento para cada membro do time e, com isso, será possível a cada robô prever o comportamento dos demais, desde que o mesmo plano seja seguido por todos. Essa é a ideia base do método proposto em seguida, onde é utilizada uma versão modificada da RRT de tempo real, a CL-RRT, para o planejamento de movimento do time como um todo.

4.4 MRPT Descentralizada em Tempo Real

Conforme descrito de maneira mais geral no Problema 4.1, em tarefas de coordenação e cooperação de robôs, o principal desafio é gerar sinais de referência \mathbf{R} para o time, ao longo do tempo, de forma descentralizada e livre de colisões. Tais referências devem ainda conduzir o time a realizar missões que lhe foram atribuídas. Conforme será mostrado a seguir, o conjunto de algoritmos propostos neste trabalho é genérico o suficiente para ser aplicado a diversos tipos de tarefas cooperativas. Nos dois capítulos subsequentes, será apresentada respectivamente a utilização do método em duas missões diferentes: (a) o controle de formação de grupos de robôs aéreos e (b) a cooperação entre veículos aéreos e terrestres para monitoramento.

4.4.1 Modelagem da Referência

A primeira dificuldade imposta pelo Corolário 4.1 diz respeito à capacidade de qualquer robô do time em prever o comportamento dos demais em função de \mathbf{R} . Conforme visto na Seção 4.1, o sinal de referência para o time constitui um vetor que cresce proporcionalmente com o número n de agentes envolvidos. Isso implica na necessidade de que cada robô conheça, pelo menos, a quantidade de veículos envolvidos na missão, quebrando assim a característica de descentralização do método.

Para resolver essa dificuldade, é preciso encontrar uma representação diferente para \mathbf{R} , de modo que esse novo modelo seja independente do número de agentes do time. Uma forma simples de se alcançar esse objetivo é por meio do uso de *abstrações* [Belta & Kumar, 2004].

Conforme discutido na Seção 3.4, tal representação permite o controle descentralizado de grupos de robôs por meio de um planejador centralizado. Porém, a abordagem se restringe a grupos de robôs homogêneos, modelados exclusivamente como veículos holonômicos no espaço de configurações, e está sujeita a falhas do agente centralizador. Além disso, as abstrações se restringem a curvas paramétricas simples, com significado físico de forma e pose espacial, e a leis de controle específicas e pouco generalizáveis.

Tais restrições vão de encontro às especificações detalhadas no Problema 4.1 e, por isso, é preciso expandir a definição da modelagem de abstrações a fim de encontrar uma solução para esse problema. Com base na Equação 3.3, considere a nova função de mapeamento:

$$\phi(\mathbf{x}^n) = \phi(\mathbf{q}^n, \dot{\mathbf{q}}^n) = (\mathbf{a}, \dot{\mathbf{a}}) = \boldsymbol{\alpha}, \quad \phi : \mathcal{X}^n \rightarrow \mathcal{A}^2, \quad (4.5)$$

que agora transforma vetores no espaço de estados do time de robôs \mathcal{X}^n para o espaço da abstração e suas derivadas \mathcal{A}^2 . Nesse caso, $\boldsymbol{\alpha}$ representa o vetor de estados da abstração, composta pelo vetor original \mathbf{a} e por sua derivada temporal $\dot{\mathbf{a}}$. Como no caso anterior, $|\mathcal{A}^2| \ll |\mathcal{X}^n|$, o que torna o uso dessa nova abstração vantajoso para o nosso problema.

De maneira mais generalizada, é possível redefinir a Equação 3.4 da seguinte forma:

$$\mathbf{u}_i(t) = \kappa(\boldsymbol{\alpha}(t), \mathbf{x}_i(t)). \quad (4.6)$$

Novamente as ações de controle individuais do time dependem apenas da abstração e dos estados de cada robô. Isso permite instanciar o vetor de referências para o sistema multirrobô como sendo simplesmente

$$\mathbf{R}(t) = \boldsymbol{\alpha}(t).$$

Conforme será discutido mais adiante, a aplicação da CL-RRT como algoritmo de planejamento permite o uso de qualquer tipo de abstração, com ou sem significância física, desde que exista ϕ^{-1} tal que seja possível verificar se $\phi^{-1}(\boldsymbol{\alpha}) \in \mathcal{W}_{\text{livre}}$ (*Módulo de Colisão* da RRT). Com tais definições em mente, nossa arquitetura de controle e comunicação passa a ser representada pela Figura 4.1:

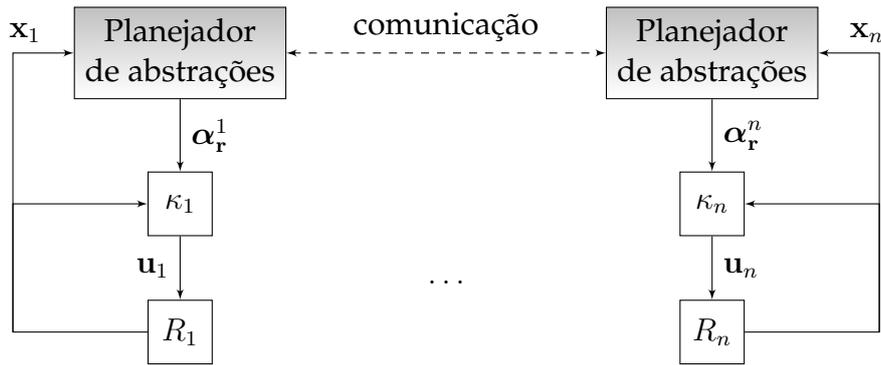


Figura 4.1. Arquitetura de controle e comunicação proposta: planejadores descentralizados, coordenados por meio da rede de comunicação, geram trajetórias α_r^i para os controladores locais de cada robô R_i do time. Esse, por sua vez, independem de um observador global, levando em consideração apenas os próprios estados do robô.

Em contraste à Figura 3.7, nossa estratégia não utiliza um observador externo, de modo que não há como medir o estado corrente da abstração formada pelo time. Assim, nosso planejador age em malha aberta do ponto de vista da abstração, o que torna mais difícil garantir a coordenação entre os agentes do time, dependendo da missão realizada. Porém, essa dificuldade é enfrentada por meio do uso de planejadores descentralizados, baseados na ideia da Função pseudoaleatória 4.4, que garantem uma exploração rápida do ambiente, de maneira coordenada. A rede de comunicação é ainda utilizada para manter o sincronismo de planejamento entre os agentes, uma vez que não existe a figura de um observador global do comportamento do time.

4.4.2 Procedimento de Expansão das Árvores

Em todos os algoritmos de exploração baseados em árvores, a parte mais fundamental é o modo de expansão dessas no ambiente. Essa é a etapa onde ocorre a parte principal do planejamento, conforme apresentando anteriormente (Algoritmo 3.1). A abordagem deste trabalho é baseada na técnica da RRT de Malha Fechada, sendo composta por duas estruturas de dados, a saber, a árvore de parâmetros da abstração \mathcal{T}_α (sinais de referência de controle multirrobo) e a árvore de estados \mathcal{T}_x^i do modelo correspondente ao robô R_i do time.

É importante não confundir essas duas estruturas com \mathcal{T}_r e \mathcal{T}_x , descritas no Capítulo 3. As primeiras são árvores de planejamento e controle descentralizado para o time, enquanto as últimas representam apenas o caso de um único robô. Tanto no Algoritmo RRT original quanto no método aqui proposto, ambas as árvores são

propagadas simultaneamente, de modo que sempre exista um único nó na árvore de estados correspondente a um único nó na árvore da abstração.

Conforme visto anteriormente, a CL-RRT é ainda composta por dois procedimentos básicos, os quais são igualmente estendidos para o caso multirrobô. O primeiro é o módulo de planejamento propriamente dito. Ele promove a expansão das duas árvores através do ambiente, eliminando os nós que levem o sistema a condições de colisão. Já o segundo é o módulo de replanejamento e de controle em tempo real. Ele determina o melhor caminho a ser seguido através das árvores, ainda em construção, e o aplica ao controlador do robô real, replanejando sempre que necessário, até que se atinja um plano realizável. Essa abordagem em malha fechada torna a navegação de cada robô mais robusta a perturbações w e, por isso, foi escolhida como base para o método deste trabalho.

Nesta seção, e na próxima, é apresentada uma nova versão desses dois módulos para o caso multirrobô. Ambos os algoritmos são executados de maneira determinística e descentralizada em cada robô do time, como mostrado na Figura 4.1. O Algoritmo 4.1, o qual representa uma adaptação do algoritmo de expansão da árvore da RRT [LaValle & Kuffner, 1999], é apresentado a seguir. O procedimento recebe como entradas as duas árvores a serem expandidas, \mathcal{T}_α e \mathcal{T}_x , além do vetor de parâmetros da abstração alvo α_{alvo} , do tempo corrente do sistema t e a sequência de sorteio do número pseudoaleatório k . Ele produz como saída as árvores expandidas.

Algoritmo 4.1 Laço de Expansão das Árvores Multirrobôs

Recebe: árvores \mathcal{T}_α e \mathcal{T}_x^i , alvo α_{alvo} , tempo corrente t , sequência numérica k

- 1: $\alpha_{\text{rand}} \leftarrow p_{\text{rand}}(k)$ tal que $\phi^{-1}(\alpha_{\text{rand}}) \in \mathcal{W}_{\text{livre}}$
 - 2: $\alpha_{\text{prox}} \leftarrow$ captura o vértice em \mathcal{T}_α mais próximo a α_{rand}
 - 3: $\alpha(t) \leftarrow \alpha_{\text{prox}}$
 - 4: $\hat{x}_i(t) \leftarrow$ captura o vértice em \mathcal{T}_x^i correspondente a α_{prox}
 - 5: **para** $\tau = t$ **até** $(t + T)$ **faça**
 - 6: $\alpha(\tau) \leftarrow$ propaga abstração em direção a α_{rand}
 - 7: $\hat{u}_i(\tau) \leftarrow \kappa(\alpha(\tau), \hat{x}_i(\tau))$
 - 8: $\hat{x}_i(\tau+1) \leftarrow f(\hat{x}_i(\tau), \hat{u}_i(\tau), \mathbf{0})$
 - 9: **se** $\hat{x}_i(\tau+1)$ é realizável e $\phi^{-1}(\alpha(\tau)) \in \mathcal{W}_{\text{livre}}$ **então**
 - 10: $\mathcal{T}_\alpha \leftarrow \mathcal{T}_\alpha \cup \alpha(\tau)$
 - 11: $\mathcal{T}_x^i \leftarrow \mathcal{T}_x^i \cup \hat{x}_i(\tau+1)$
 - 12: **fim se**
 - 13: **se** $\alpha(\tau)$ alcançou α_{rand} **ou** $\hat{x}_i(\tau+1)$ alcançou $\phi^{-1}(\alpha_{\text{alvo}})$ **então**
 - 14: **interrompe o laço**
 - 15: **fim se**
 - 16: **fim para**
- Retorna:** \mathcal{T}_α e \mathcal{T}_x^i
-

Inicialmente é escolhida uma amostra $\alpha_{\text{rand}} \in \mathcal{A}^2$ do espaço de abstração (linha 1), com projeção em $\mathcal{W}_{\text{livre}}$, a qual será usada para a propagação da referência de controle do sistema. Em seguida, é escolhido um vértice α_{prox} , pertencente a \mathcal{T}_α , que seja o mais próximo da amostra aleatória (linha 2). Tal proximidade é dada em termos da minimização de uma função de distância $h(\alpha_{\text{prox}}, \alpha_{\text{rand}})$, definida como

$$h : \mathcal{A}^2 \times \mathcal{A}^2 \rightarrow \mathbb{R}^+,$$

onde basicamente é considerada a distância da referência de controle em relação à amostra espacial α_{rand} .

Posteriormente é identificado o nó em \mathcal{T}_x^i correspondente a $\alpha_{\text{prox}} \in \mathcal{T}_\alpha$ (linha 4). Então, a partir desse ponto, a referência e os estados do sistema são propagados em direção a amostra α_{rand} durante um tempo finito $t+T$. A cada nova iteração do laço do algoritmo, a referência é propagada (linha 6), dependendo do tipo de missão. Os próximos capítulos apresentam casos onde nossa técnica de planejamento é aplicada aos problemas de controle de formação de robôs aéreos e de cooperação aéreo-terrestre. Nesse passo da abordagem, \hat{x} representa apenas uma estimativa dos estados do robô R_i , propagados segundo a Equação 3.1 (linha 8), desconsiderando-se as incertezas de medição e perturbações w .

A entrada de controle \hat{u} é calculada segundo a lei de controle definida pela Equação 4.6 (linha 7) de modo a minimizar a distância entre o robô e a referência α a cada interação do laço. Cada nova estimativa de estado $\hat{x}(\tau+1)$ livre de colisões gerada e sua correspondente referência $\alpha(\tau)$ são incorporadas às árvores \mathcal{T}_x^i e \mathcal{T}_α respectivamente. O laço é interrompido caso α_{rand} ou α_{alvo} sejam alcançados por $\alpha(\tau)$ e $\hat{x}(\tau+1)$ respectivamente. É preciso ressaltar ainda que o fato de a abstração não colidir com os obstáculos não garante que os robôs também não colidirão. Isso é garantido no algoritmo apresentado na sequência.

A Figura 4.2 apresenta o procedimento de expansão das árvores multirrobôs para o caso de um grupo \mathcal{R} composto de dois robôs, um representado por um quadrado e outro por um círculo. A curva paramétrica utilizada como abstração corresponde a um simples seguimento de reta de tamanho constante que liga o círculo ao quadrado. Ambos os veículos começam no estado inicial \mathbf{X}_{ini} e têm como objetivo o estado final α_{alvo} . Individualmente cada robô possui a mesma versão da árvore de abstração (em cinza), expandida por uso de uma função pseudoaleatória que determina a sequência de amostras α_{rand} . As árvores pretas, porém, são conhecidas apenas por seus respectivos robôs e representam a estrutura \mathcal{T}_x^i correspondente, composta pelos estados de cada robô R_i ao longo do tempo.

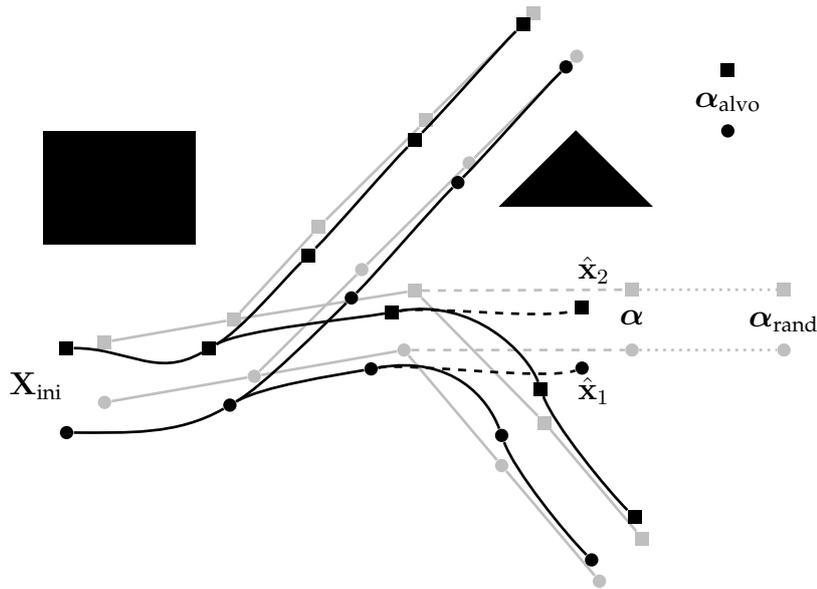


Figura 4.2. Laço de expansão das árvores \mathcal{T}_α (em cinza), \mathcal{T}_x^1 (em preto – robô circular) e \mathcal{T}_x^2 (em preto – robô quadrado) na MRPT: o algoritmo inicia a partir de \mathbf{X}_{ini} e itera até alcançar α_{alvo} , evitando todos os obstáculos contidos em \mathcal{O} . A cada referência de controle α_{rand} escolhida, a referência α mais próxima a α_{prox} é calculada e o modelo é propagado por meio do controlador até \hat{x} para cada robô R_i em \mathcal{R} .

4.4.3 Procedimento de Replanejamento em Tempo Real

Conforme mostrado anteriormente, a árvore de estados de cada robô R_i do time é obtida com base no modelo de previsão da Equação 3.1, desconsiderando-se a existência de perturbações e erros de medição sensorial. Porém, no caso real, os agentes estão sempre sujeitos a vários tipos de incertezas e, por esse motivo, as estimativas \hat{x} calculadas pelo Algoritmo 4.1 não garantem a ausência de colisão e nem o alcance de α_{alvo} . Devido principalmente a essa característica, a CL-RRT apresenta uma componente de execução em tempo real [Kuwata et al., 2008].

Basicamente o método utiliza o modelo dinâmico do veículo para propagar sua trajetória a partir da última medição realizada (com incertezas), através do melhor caminho encontrado na árvore até o instante corrente. Essa propagação avalia a factibilidade do caminho a partir da pose corrente do veículo até a posição que será alcançada no próximo instante de tempo. Para atribuir essa característica ao nosso método, é proposto o Algoritmo 4.2, apresentado a seguir. Trata-se de uma versão modificada do laço de execução da CL-RRT para o caso multirrobô. As modificações introduzidas tornam o procedimento original menos susceptível a problemas que possam ocorrer quando determinado agente R_i não segue corretamente o plano

estabelecido por ele, ou quando esse plano leva os companheiros de time a um estado de colisão.

O procedimento recebe como entrada apenas os estados inicial $\mathbf{x}_{\text{ini}}^i$ e final α_{alvo} do sistema multirrobôs, e itera até que o agente R_i (que executa o procedimento de forma descentralizada) atinja seu respectivo estado alvo $\mathbf{x}_{\text{alvo}}^i$ correspondente em α_{alvo} .

Algoritmo 4.2 Laço de Execução Multirrobôs

Recebe: estado inicial $\mathbf{x}_{\text{ini}}^i$, abstração alvo α_{alvo}

- 1: $t \leftarrow 0, k \leftarrow 1$
- 2: $\mathcal{T}_x^i \leftarrow \mathbf{x}_{\text{ini}}^i, \mathcal{T}_\alpha \leftarrow \mathbf{0}$
- 3: **enquanto** $\mathbf{x}_i(t)$ não alcançou $\phi^{-1}(\alpha_{\text{alvo}})$ **faça**
- 4: **enquanto** restar tempo em ΔT **faça**
- 5: expande \mathcal{T}_α e \mathcal{T}_x^i (Algoritmo 4.1)
- 6: $k \leftarrow k+1$
- 7: **fim enquanto**
- 8: medir $\mathbf{x}_i(t)$ e atualizar $\mathcal{W}_{\text{livre}}(t)$
- 9: $\tilde{k} \leftarrow$ passo pseudoaleatório k corrente
- 10: estima o melhor caminho corrente α^* em \mathcal{T}_α até \tilde{k}
- 11: **se** $\alpha^* = \emptyset$ **então**
- 12: aplica ação de segurança e vai para linha 4
- 13: **fim se**
- 14: $\alpha_r^i(t) \leftarrow \alpha^*(t) + \varrho(\hat{\mathbf{x}}_i(t) - \mathbf{x}_i(t))$
- 15: **se** $[\alpha_r^i(t) \text{ repropagado para } R_i] \notin \mathcal{W}_{\text{livre}}(t)$ **então**
- 16: remove as partes não realizáveis de $\alpha_r^i(t)$
- 17: **fim se**
- 18: aplica $\alpha_r^i(t)$ ao controlador do robô R_i
- 19: $t \leftarrow t + \Delta T$
- 20: **fim enquanto**

Após a inicialização das árvores, do tempo t e da sequência numérica k , o planejador do robô R_i utiliza o algoritmo anterior para promover a expansão das árvores, tal qual como ocorre no Algoritmo 3.3.

Em seguida é feita a medição do estado $\mathbf{x}_i(t)$ do robô no tempo corrente, além da atualização das informações do $\mathcal{W}_{\text{livre}}$ (linha 8). Novamente é preciso mapear o ambiente caso esse seja desconhecido ou sujeito a mudanças. Conforme discutido previamente, o método proposto aqui pode ser aplicado a ambientes que mudam dinamicamente. Porém é preciso garantir que todos os robôs do time possuam a mesma versão do mapa atualizado para que não ocorram incoerências entre seus planos individuais (vide Suposição 4.4). Por razões de simplicidade, foi escolhido representar apenas os estados do robô como entrada do plano (Figura 4.1).

Uma vez que as árvores tenham sido expandidas e que todos os robôs conheçam seus estados (e o ambiente), o próximo passo é determinar o melhor caminho corrente, assim como era feito no caso simples da CL-RRT. Porém, ao se realizar o planejamento descentralizado do time, surge um novo problema: garantir que todos os robôs, com suas diferentes capacidades de processamento, se encontrem no mesmo passo da sequência pseudoaleatória ao executarem o melhor caminho corrente. Esse ponto é fundamental para se manter a coordenação entre os planos dos agentes.

Para resolver esse problema, foi introduzida uma etapa de sincronização entre os robôs (linha 9). Ela estabelece o passo aleatório corrente, \tilde{k} , variável que governa a evolução do planejamento global do grupo. A ideia é que existam regras para garantir que nenhum robô se atrase em relação ao time, devido principalmente a limitações de processamento e a influência disso na expansão das árvores, ou que o atraso seja identificado e tratado pelo time. A maneira mais conservadora de se definir \tilde{k} é atribuir-lhe o valor do passo k do robô mais lento (aquele que mais demora em promover a expansão das árvores). Assim, mesmo que outros agentes se encontrem em níveis mais avançados do planejamento, a execução do plano se dará de maneira coordenada. A vantagem dessa abordagem é que os robôs precisam comunicar a seus companheiros apenas o valor do passo em que se encontram. Isso é feito com base na Suposição 4.3 através do canal de comunicação representado na Figura 4.1. Porém, caso algum robô se atrase muito em relação aos demais, ou não consiga realizar a missão por qualquer problema, o time pode tomar uma atitude, modificando a configuração dos veículos em relação ao plano original. Na Seção 6.3.2 é apresentado um experimento simulado com falha em um dos robôs do grupo e a reconfiguração do time em relação ao plano.

O passo pseudoaleatório corrente é então utilizado por todos os robôs na determinação do melhor caminho. Assim como no algoritmo original da CL-RRT, o melhor caminho α^* em \mathcal{T}_α a ser seguido é escolhido (linha 10). Tal escolha é normalmente realizada a partir da minimização de alguma heurística baseada em custos do caminho [Frazzoli et al., 2002]. Diversas heurísticas podem ser utilizadas para esse cálculo [Kuwata et al., 2009], dentre as quais a mais comum é a da distância Euclidiana até o alvo. Uma ação de segurança deve ser automaticamente aplicada, caso tal caminho não possa ser encontrado naquele instante. Os robôs podem parar de se movimentar (caso seja seguro) até que um novo caminho seja encontrado.

Embora a sincronização entre os agentes reduza os erros na execução da missão atribuída ao grupo, ela não garante inteiramente a ausência de colisão entre os robôs, causadas por eventuais atrasos na execução dos respectivos planos. Por ou-

tro lado, é razoável supor que, em um problema de coordenação e/ou cooperação de times de robôs, quanto menor for a diferença entre os estados reais $\mathbf{x}_i(t)$ de cada agente e suas estimativas $\hat{\mathbf{x}}_i(t)$ (linha 8 do Algoritmo 4.1), menores serão os riscos de colisão entre agentes do time. Em outras palavras, se cada robô R_i se comportar segundo o planejado ($\hat{\mathbf{x}}_i(t)$), maiores serão as chances de sucesso para a completude da missão.

Inspirados pelas proposições para diminuição do erro de predição apresentadas em [Kuwata et al., 2008] para o caso de um único veículo, é proposto aqui o uso da informação de predição dos estados do sistema, existentes localmente em cada um dos robôs, para melhorar a coordenação do time de robôs reais. Tal melhoria se dá exatamente por meio da diminuição do erro entre a predição (ou planejamento) dos estados de cada robô do time e da medição corrente desses estados, definido como $e_x^i(t) = \hat{\mathbf{x}}_i(t) - \mathbf{x}_i(t)$. Cada elemento de α^* corresponde a uma referência \mathbf{R} predita pelo Algoritmo 4.1. Esse seria o sinal de referência ideal caso e_x fosse sempre nulo. Entretanto, como isso não é sempre verdadeiro, é preciso modificar a referência dentro do planejador para diminuir esse erro.

Essa informação é utilizada para calcular o sinal de referência modificado α_r (linha 14), aplicado ao controlador do robô real. A função $\varrho(\cdot)$ promove a correção da referência em função do erro corrente. A Figura 4.3 apresenta em detalhes a malha de controle do robô, incluindo a etapa de correção do modelo. Essa abordagem tem a vantagem de não modificar a estrutura do controlador utilizado, o que poderia introduzir instabilidade ao sistema. Entretanto, variar bruscamente o sinal de referência pode ainda trazer dificuldades para que o controle siga a trajetória pré-determinada. Assim, o controlador deve ser projetado de modo a apresentar certa robustez em relação a tais efeitos. Nos Capítulos 5 e 6 será mostrado que o erro de predição diminui significativamente para os experimentos realizados.

Por fim, o modelo do veículo é repropagado a partir de seu estado atual (usando a nova referência modificada α_r) e as partes que levem o sistema a colisão são retiradas de α^* . No algoritmo original da CL-RRT, essa etapa é comumente chamada de Verificação Preguiçosa (*lazy check approach*) e serve para evitar a checagem de colisões na árvore inteira, o que pode se tornar dispendioso computacionalmente. A referência α_r , livre de colisões, é então aplicada ao controlador do robô R_i e o laço do algoritmo prossegue até que os robôs atinjam seus objetivos.

A Figura 4.4 apresenta o laço de execução da MRPT para o caso multirrobô descrito na seção anterior. É possível observar que, após certa expansão das árvores \mathcal{T}_α (em cinza) e \mathcal{T}_x^i com $i = 1, 2$ (em preto), os robôs decidem de maneira descentralizada o melhor caminho corrente a ser seguido (destacado em vermelho). Parte

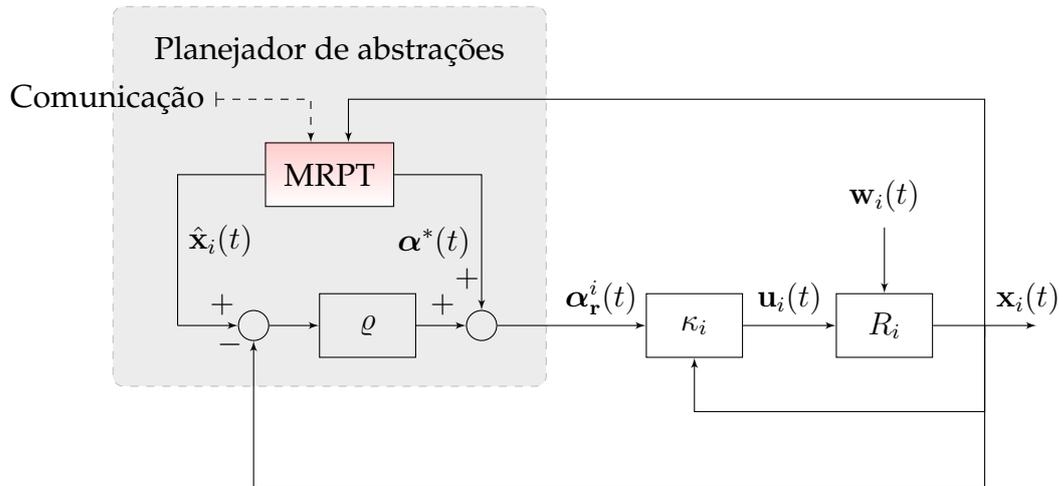


Figura 4.3. Visão detalhada da arquitetura de controle proposta: malha de controle com aplicação da heurística de correção (linha 14 do Algoritmo 4.2): o vetor de estados estimado $\hat{\mathbf{x}}_i$ é usado para reduzir o erro de previsão, enquanto a malha de controle do sistema segue a referência α_r .

desse caminho é aplicado aos controladores do veículo, que se movem da condição inicial \mathbf{X}_{ini} até posições mais próximas de α_{alvo} .

Na sequência são apresentados experimentos onde a metodologia é aplicada a diferentes tipos de missão. O Capítulo 5 mostra a aplicação da metodologia proposta a um problema mais específico: o planejamento e controle de formação de um grupo de robôs. O grupo em questão é composto de robôs aéreos de asa-rotativa do tipo VTOL. Já o Capítulo 6 apresenta a instanciação do método a outro problema específico: a cooperação aéreo-terrestre em missões de vigilância e escolta. Nesse caso, nosso grupo é composto de robôs aéreos do tipo VTOL e de robôs terrestres baseados na plataforma de *Ackerman*.

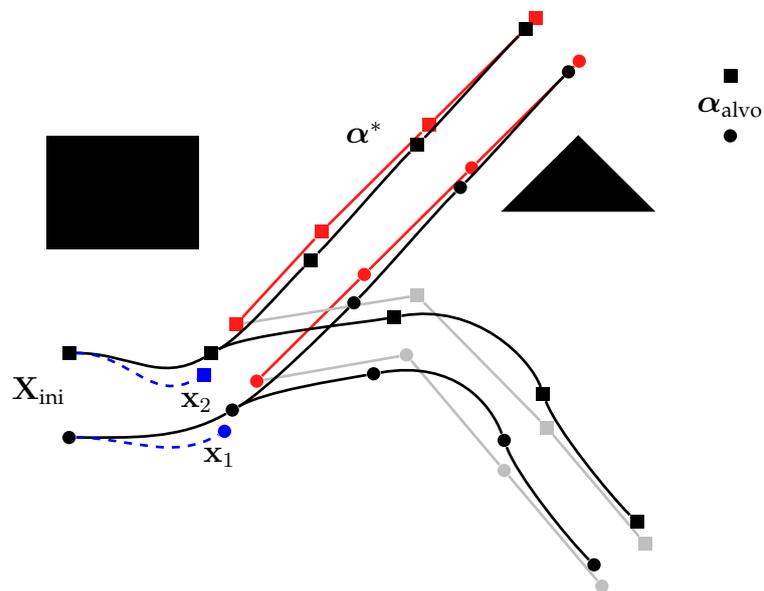


Figura 4.4. Laço de execução da MRPT: a cada instante de tempo real, as referências para o melhor caminho α^* na árvore \mathcal{T}_α corrente são escolhidas (em vermelho), de modo a minimizar a distância do sistema multirrobo real para α_{alvo} , e cada robô real do time segue seu melhor caminho correspondente (pontilhado em azul).

Capítulo 5

Aplicação ao Controle de Formação

– Cobras! Por que tinham que ser cobras?

Indiana Jones, Os Caçadores da Arca Perdida

Este capítulo avalia a aplicabilidade do método proposto no Capítulo 4 a um cenário desafiador: o controle de formação de um grupo de UAVs em ambiente externo. Nosso arcabouço permite ao time de robôs navegar através do ambiente enquanto seus membros se mantêm relativamente posicionados entre si. São considerados modelos dinâmicos aproximados para veículos (UAVs do tipo quadrirotor) e algoritmos de controle simples para estabilizar esses robôs ao longo de suas trajetórias. Foi utilizado também um modelo simplificado de colisão entre veículos, que leva em consideração perturbações por escoamento de ar provocado pelo conjunto de hélices. Por fim, são apresentados resultados para casos experimentais simulados (empregando diferentes tipos de abstrações) e reais.

5.1 Introdução

O problema do controle de formação é um dos mais fundamentais, e importantes, em cenários de sistemas cooperativos. Muitas vezes, tarefas executadas por um grupo de robôs devem ser realizadas enquanto esse mantém algum tipo de padrão geométrico, rígido ou intercambiável. Embora exista uma vasta literatura sobre o tema, algumas questões práticas e teóricas ainda demandam por melhores soluções, principalmente no que tange ao planejamento de movimento para formações e avaliações de desempenho dos métodos [Chen & Wang, 2005]. Esta parte do trabalho concentra esforços no problema de planejamento de movimento através de ambien-

tes complexos, em tempo real, para grupos de robôs sujeitos a restrições de formação geométrica.

Existem algumas categorias que englobam as técnicas de controle de formação para múltiplos agentes. Dentre as mais conhecidas estão as baseadas em *seguimento de líder*, em *comportamentos* e em *estrutura virtual* [Consolini et al., 2007]. A abordagem baseada em seguimento de líder [Desai et al., 2001], mais popular entre as três, utiliza um ou mais robôs líderes, que seguem uma trajetória pré-definida para coordenar outros robôs (seguidores) em formação. Embora simples e escalável, essa abordagem é pouco tolerante a falhas, por ser fortemente dependente dos líderes do grupo. Existem dois problemas principais nesse contexto: em caso de falha de um líder, seus seguidores se perdem em relação ao planejamento e, em caso de falha de um seguidor, o líder se mantém alheio e não altera seus planos. Uma abordagem cooperativa descrita em [Pereira et al., 2003] foi proposta para resolver esses problemas.

Na abordagem baseada em comportamentos (ou abordagem comportamental) [Balch & Arkin, 1997], o controle de cada robô do time depende de uma série de comportamentos específicos e pré-determinados, como por exemplo, manter a formação e desviar de obstáculos simultaneamente. Tal abordagem, porém, é de difícil avaliação formal, devido à alta complexidade, e nem sempre é possível garantir que alcance convergência para a formação desejada.

Finalmente, na abordagem baseada em estrutura virtual [Tan, 1997], uma estrutura geométrica e abstrata é utilizada para controlar os robôs como se esses fossem partículas da própria estrutura. Essa abordagem é normalmente aplicável a um grande número de agentes, entretanto requer grande largura de banda para comunicação entre os elementos do time. Além disso, esse método é pouco tolerante a falhas de rede. Outro problema dessa, e das demais técnicas discutidas, é a dificuldade de descentralização do planejamento, uma vez que sempre é necessário se conhecer o número de agentes envolvidos na tarefa. Em outras palavras, a dimensão da estrutura virtual cresce com o número de robôs do time.

Uma alternativa a todas essas técnicas é o uso de abstrações [Belta & Kumar, 2004]. Conforme discutido na Seção 3.4, essa metodologia apresenta uma série de vantagens, especialmente no que tange ao problema de controle de formação. O arcabouço proposto neste trabalho (descrito em detalhes no Capítulo 4) permite o controle de formação de um time de robôs com base na ideia de agrupamento por abstrações em ambientes com obstáculos.

5.2 Definição do Problema

Considerando o problema de controle de formação, uma possível abordagem é impor restrições geométricas a \mathbf{R} , de modo que a referência seguida pelo grupo no Problema 4.1 leve todos os robôs a se deslocarem individualmente segundo a formação desejada. Por razões previamente discutidas, foi utilizada uma abordagem de controle de formação baseada em abstrações. Assim, o problema específico tratado neste capítulo é definido da seguinte forma:

Problema 5.1 (Planejamento e controle de formação descentralizado). *Considerando o Problema 4.1, onde um grupo de robôs \mathcal{R} deve navegar através do espaço de trabalho de um estado inicial \mathbf{X}_{ini} para um estado final α_{alvo} , planejar uma trajetória para a formação \mathbf{R} de maneira descentralizada e livre de colisões entre os agentes do time.*

5.2.1 Modelo Dinâmico do Veículo Aéreo

Neste trabalho foi utilizado um modelo matemático para quadrirotores baseado no formalismo de *Newton-Euler*, em um espaço de trabalho $\mathcal{W} \equiv \mathbb{R}^3$. Esse modelo é adaptado do arcabouço apresentado em [Michael et al., 2010], onde os autores utilizam a mesmo tipo de UAV em seus experimentos. Foi considerado o vetor de estados $\mathbf{x} \in \mathbb{R}^{12}$, composto pelas posições e orientações do veículo no Grupo de *Lie* $SE(3)$, e suas respectivas derivadas, tal que

$$\mathbf{x} = \left[\mathbf{p} \quad \mathbf{v} \quad \Psi \quad \Omega \right]^T,$$

onde \mathbf{p} e \mathbf{v} são os vetores de posições e de velocidades lineares, respectivamente, do centro de massa do veículo em relação ao frame do mundo $\{\mathcal{W}\} \in \mathbb{R}^3$, Ψ é o vetor de orientações em $SO(3)$, também em $\{\mathcal{W}\}$ e Ω é o vetor de velocidades angulares em relação ao frame de referência do corpo do veículo $\{\mathcal{B}\}$.

Foi considerado ainda o vetor de entrada do sistema

$$\mathbf{u} = \omega,$$

onde $\omega \in \mathbb{R}^4$ representa o vetor de velocidades, em rpm¹, aplicada aos motores do quadrirotor. Esses, por sua vez, são aqui considerados como sistemas estáticos com saturações de operação, uma vez que a dinâmica de aceleração dos motores é mais rápida do que a variação dos próprios estados do veículo. A Figura 5.1 mostra uma representação das forças e dos momentos que atuam sobre o veículo, além dos

¹Rotações por minuto.

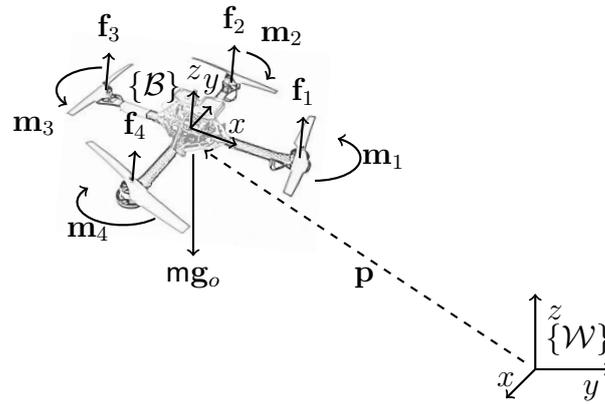


Figura 5.1. Frame de referências do veículo quadricóptero: forças e momentos atuantes na dinâmica do veículo.

sistemas de coordenadas adotados. Essa representação foi utilizada como base para a modelagem apresentada na sequência.

A variação temporal do movimento translacional do robô pode ser descrita conforme apresentado nas Equações 5.1 e 5.2 [Michael et al., 2010],

$$\dot{\mathbf{p}} = \mathbf{v}, \quad (5.1)$$

$$m\dot{\mathbf{v}} + \mathbf{C}\mathbf{v}|\mathbf{v}| + (\mathbf{v} \times m\boldsymbol{\Omega}) + m\mathbf{g}_o = \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 \mathbf{f}_i \end{bmatrix}, \quad (5.2)$$

onde $\mathbf{p} = [x, y, z]^T$ é dado em coordenadas inerciais em m (metros) e $\mathbf{v} = [\dot{x}, \dot{y}, \dot{z}]^T$ em m/s (metros por segundo).

Ainda nesse caso, m é a massa do quadricóptero, \mathbf{C} é a matriz de coeficientes de arrasto e \mathbf{R} a matriz de rotação em $\text{SO}(3)$ dos ângulos do veículo no espaço (discutida mais a frente) que leva de $\{\mathcal{B}\}$ para $\{\mathcal{W}\}$. Já \mathbf{C} é descrita segundo a Equação 5.3,

$$\mathbf{C} = \frac{1}{2}c\mathbf{A}\rho_o, \quad (5.3)$$

em que c é o coeficiente de arrasto médio, \mathbf{A} é a matriz diagonal representando as áreas das seções transversais em cada eixo do veículo e $\rho_o \approx 1,22 \text{ kg/m}^3$ é a densidade do ar de acordo com a Atmosfera Padrão Internacional (*International Standard Atmosphere*, ou ISA). Foi considerado também

$$\mathbf{g}_o = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T,$$

o vetor gravidade em relação ao $\{\mathcal{W}\}$, com $g \approx 9,78 \text{ m/s}^2$.

Já \mathbf{f} representa o vetor de forças geradas pela rotação dos motores (Figura 5.1), calculadas segundo a Equação 5.4,

$$\mathbf{f}_i = k_f \omega_i^2, \quad i = 1 \dots 4, \quad (5.4)$$

onde k_f é um valor escalar de transformação.

De forma semelhante, a variação temporal do movimento rotacional do quadrirotor é descrita por meio das Equações 5.5 e 5.6,

$$\dot{\Psi} = B\Omega, \quad (5.5)$$

$$I\dot{\Omega} + C\Omega|\Omega| + (\Omega \times I\Omega) = \begin{bmatrix} l(\mathbf{f}_2 - \mathbf{f}_4) \\ l(\mathbf{f}_3 - \mathbf{f}_1) \\ \sum_{i=1}^4 (-1)^{i+1} \mathbf{m}_i \end{bmatrix}, \quad (5.6)$$

onde $\Psi = [\phi, \theta, \psi]^T$, em rad (radianos), é o vetor de ângulos de *Euler* e $\Omega = [p, q, r]^T$, em rad/s, é a velocidade de rotação observada junto ao referencial do corpo do veículo $\{\mathcal{B}\}$. Ainda I representa a matriz inercial do robô, diagonal devido à simetria do veículo, e l é a distância entre a posição de cada motor e o centro de massa do sistema.

Foi utilizada aqui a representação em ângulos de *Euler ZYX*, onde ϕ , θ e ψ são os ângulos de rolamento (*roll*), arfagem (*pitch*) e guinada (*yaw*), respectivamente. Com isso, a matriz R de transformação de $\{\mathcal{B}\}$ para $\{\mathcal{W}\}$ é descrita como sendo

$$R = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}.$$

De forma semelhante, a matriz B pode ser calculada como

$$B = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix}.$$

A variação angular é controlada diretamente por meio das forças e dos mo-

mentos gerados pelos motores. O vetor de momentos angulares \mathbf{m} (Figura 5.1) é calculado por meio da Equação 5.7,

$$\mathbf{m}_i = k_m \omega_i^2, \quad i = 1 \dots 4, \quad (5.7)$$

onde k_m é outro ganho escalar do motor.

5.2.1.1 Leis de Controle

Para estabilizar e comandar os quadrirotores através do ambiente, foram utilizados neste trabalho controladores do tipo Proporcional–Derivativo (PD) e controladores não lineares. Esquemas de controle semelhantes aos utilizados para esse tipo de veículo podem ser vistos em [Gurdan et al., 2007; Guerrero et al., 2010; Achteik et al., 2011]. Todos os controladores foram obtidos com base na linearização do sistema em torno de um ponto de operação, correspondente à condição de voo pairado (veja [Michael et al., 2010] para mais detalhes). Em primeiro lugar, a entrada para a Equação de controle 4.3 é dada pelo vetor de referência $\mathbf{r} \in \mathbb{R}^3$, tal que

$$\mathbf{r} = [x_{\text{ref}}, y_{\text{ref}}, z_{\text{ref}}]^T,$$

seja a posição desejada em $\mathcal{W}_{\text{livre}}$.

Foi considerado ainda $\mathbf{p}_o = [x_o, y_o, z_o]^T$ e $\mathbf{v}_o = \Psi_o = \Omega_o = [0, 0, 0]^T$ como as condições de operação para esse caso. Nessa condição de voo pairado, as velocidades de rotação (em rpm) dos motores podem ser definidas da seguinte forma,

$$\omega = \begin{bmatrix} \Delta\omega_T - \Delta\omega_\theta + \Delta\omega_\psi \\ \Delta\omega_T + \Delta\omega_\phi - \Delta\omega_\psi \\ \Delta\omega_T + \Delta\omega_\theta + \Delta\omega_\psi \\ \Delta\omega_T - \Delta\omega_\phi - \Delta\omega_\psi \end{bmatrix}, \quad (5.8)$$

onde $\Delta\omega_T$ é a velocidade de tração que promove a sustentação do veículo, e $\Delta\omega_\Psi = [\Delta\omega_\phi, \Delta\omega_\theta, \Delta\omega_\psi]$ é o vetor de variações de rotação para a produção dos momentos angulares de rolamento, arfagem e guinada, respectivamente. Essas variações na rotação das hélices são determinadas pela lei de controle de atitude,

$$\Delta\omega_\Psi = \Lambda_\Psi (\Psi_{\text{ref}} - \Psi) - \Lambda_\Omega B\Omega, \quad (5.9)$$

onde Ψ_{ref} é o vetor de orientações desejadas (limitadas a pequenos ângulos devido ao ponto de linearização escolhido), Ψ e Ω são estimativas de atitude e de velocidade.

des angulares do robô. Aqui, Λ_{Ψ} e Λ_{Ω} são matrizes diagonais de ganhos definidas positivas.

Com relação ao controle de posição, e mais especificamente ao controle de altitude, a velocidade de rotação de tração $\Delta\omega_T$ é definida em função da altitude desejada z_{ref} , de acordo com a lei de controle não linear

$$\Delta\omega_T = \frac{\lambda_1 (z_{\text{ref}} - z) - \lambda_2 \dot{z} + \sqrt{\frac{mg}{4k_f}}}{\cos \phi \cos \theta}, \quad (5.10)$$

onde λ_1 e λ_2 são os ganhos proporcional e derivativo de controle, ambos positivos. As variáveis z e \dot{z} correspondem à medição de altitude e sua respectiva variação, ao passo que ϕ e θ compõem o vetor Ψ .

Após os calculados das leis de controle 5.9 e 5.10, cada uma das velocidades determinadas na Equação 5.8 está sujeito a saturações de operação do tipo

$$\omega_{\min} \leq \omega_i \leq \omega_{\max} \quad i = 1 \dots 4,$$

sendo ω_{\min} e ω_{\max} as velocidades de rotação mínima e máxima de operação dos motores do UAV.

Finalizando o controle de posição, os ângulos de referência do controle de atitude são utilizados para comandar o robô ao longo do plano da Terra. Por razões de simplificação, o ângulo de referência de arfagem ψ_{ref} foi arbitrado como zero, e os demais em função das posições desejadas x_{ref} e y_{ref} foram determinadas conforme a seguir:

$$\begin{bmatrix} \phi_{\text{ref}} \\ \theta_{\text{ref}} \end{bmatrix} = \lambda_3 \begin{bmatrix} x_{\text{ref}} - x \\ y_{\text{ref}} - y \end{bmatrix} - \lambda_4 \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (5.11)$$

onde x , y , \dot{x} e \dot{y} são medições de posições e de velocidades translacionais do veículo. Assim como no controle de altitude, λ_3 e λ_4 são ganhos positivos de estabilização. A Equação 5.11 representa o laço de controle mais externo ao controlador de atitude.

5.2.1.2 Restrições entre Agentes

Veículos do tipo helicóptero são exemplos especiais de plataformas aéreas. Uma vez que a sustentação venha diretamente do sistema de tração (diferentemente dos aviões que usam as asas para esse fim), tais plataformas se tornam mais suscetíveis a perturbações de escoamento de ar do que outros tipos de veículos, como aeronaves de asa fixa e dirigíveis. Assim, a área de perturbação sobre tais veículos é geralmente

maior do que a do veículo propriamente dita. Isso implica na necessidade de se ampliar o volume de colisão entre agentes do tipo helicóptero em um mesmo time, para evitar que o escoamento de ar das hélices de um agente R_i interfira na de outro agente R_j .

Experimentos apresentados em [Michael et al., 2010] mostram que a região de influência entre dois quadrirotores, idênticos aos utilizados nesse trabalho, pode ser aproximada por um volume cilíndrico com aproximadamente 0,5 metro de raio e 1,5 metros de altura, tomados a partir do centro geométrico do robô. Essas informações foram utilizadas para modelar a Desigualdade 4.2 do nosso sistema. Geralmente, $g(\mathbf{x}_i, \mathbf{x}_j)$ pode ser modelada por meio de qualquer função diferenciável e convexa, mas por simplicidade, foi escolhida a seguinte representação

$$g(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} (x_i - x_j)^2 + (y_i - y_j)^2 - r^2, & \text{se } |z_i - z_j| < h \\ \infty, & \text{do contrário} \end{cases},$$

onde r e h correspondem ao raio e à altura do cilindro no espaço, cujos valores numéricos são os mesmos citados anteriormente. Porém, esse modelo é válido apenas para pequenos valores de ϕ e θ (voo pairado) do quadrirotor e para veículos geometricamente equivalentes, como é o nosso caso. Ainda por razões de segurança, foi considerado um valor δ_x , tal que

$$g(\mathbf{x}_i, \mathbf{x}_j) \leq \delta_x$$

seja um limiar na validação da função de colisão, servindo como margem de segurança para incertezas de medição na localização dos robôs.

5.3 Experimentos Simulados

Para testar a metodologia proposta, primeiramente foi construído um arcabouço experimental simulado utilizando o programa *Matlab*. O objetivo principal foi avaliar a qualidade dos resultados produzidos e, sobretudo, reproduzir diversos testes para avaliar o uso de diferentes modelos de abstração.

Os experimentos consistiram em levar o grupo \mathcal{R} de seis quadrirotores, em formação, de um estado inicial \mathbf{X}_{ini} para um estado final (governado pela abstração α_{alvo}) em um ambiente com obstáculos. Para testar a robustez do método frente à existência de perturbações, foram acrescentadas à simulação forças laterais (paralelas ao eixo X) equivalentes a rajadas de vento constantes de aproximadamente 20 km/h. As dimensões do ambiente considerado são de 30 metros de largura por 50

de comprimento e 6 de altura, dentro do qual foram incorporados oito obstáculos cúbicos. Foi utilizada uma função do *Matlab* para gerar a sequência pseudoaleatória, com distribuição uniforme, igualmente compartilhada entre os robôs do time para escolher pontos $\mathbf{r}_{\text{rand}} \in \mathcal{W}_{\text{livre}}$.

As abstrações foram construídas segundo o modelo de duas curvas paramétricas: a elipse planar e a curva de Bézier. Na sequência são apresentados os dois casos.

5.3.1 Elipses como Abstração

A forma mais simples de abstração utilizada em [Belta & Kumar, 2004] é a representação elíptica. Os robôs do time são agrupados em componentes de configuração e forma, segundo a Equação 5.12

$$\mathbf{a}(\epsilon) = \langle \mathbf{g}, \mathbf{s} \rangle (\epsilon) = \mathbf{p}_c + \begin{bmatrix} \cos \psi \cos \epsilon & -\sin \psi \sin \epsilon \\ \sin \psi \cos \epsilon & \cos \psi \sin \epsilon \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}, \quad (5.12)$$

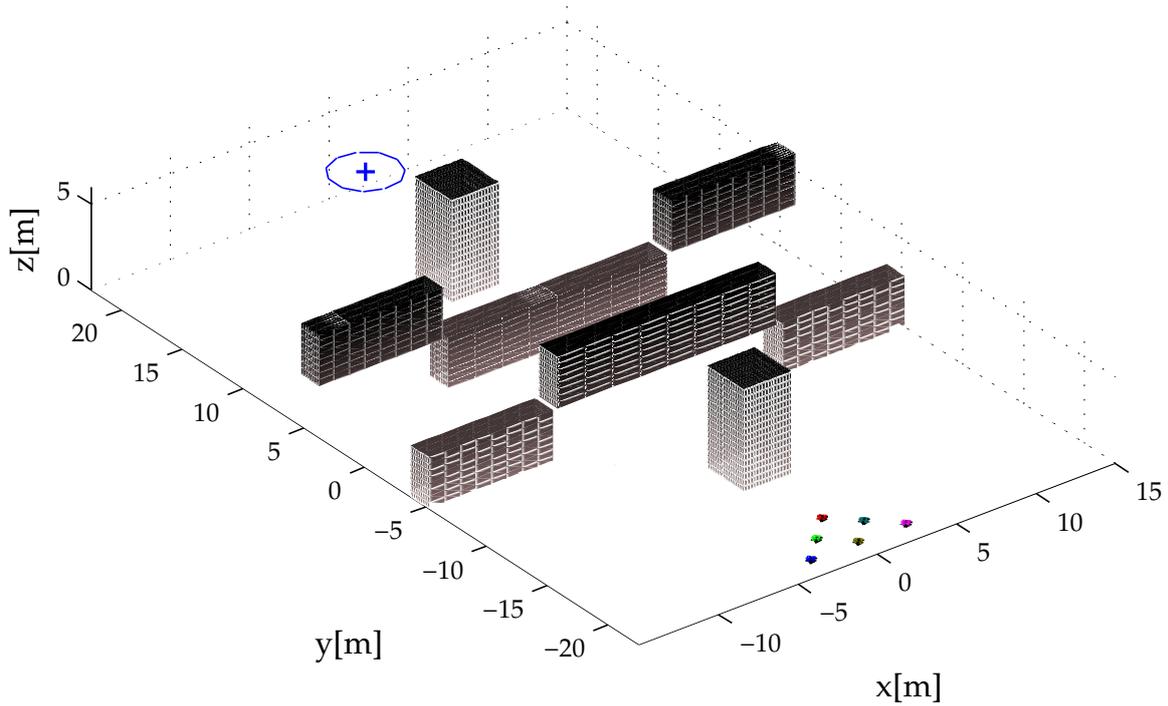
onde $\epsilon \in \mathbb{R}$ é o parâmetro da curva, $\mathbf{p}_c \in \mathbb{R}^3$ é o centroide da elipse, ψ é a orientação do eixo principal e $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2]^T$ são as dimensões dos eixos que determinam a forma da curva.

É possível ver que essa abstração se mantém sempre paralela ao Plano XY , ou seja, todos os seus pontos se encontram à mesma altitude. Essa, por sua vez, não é constante, variando em função da movimentação do centroide. A derivada temporal da abstração pode ser descrita por meio dessa movimentação

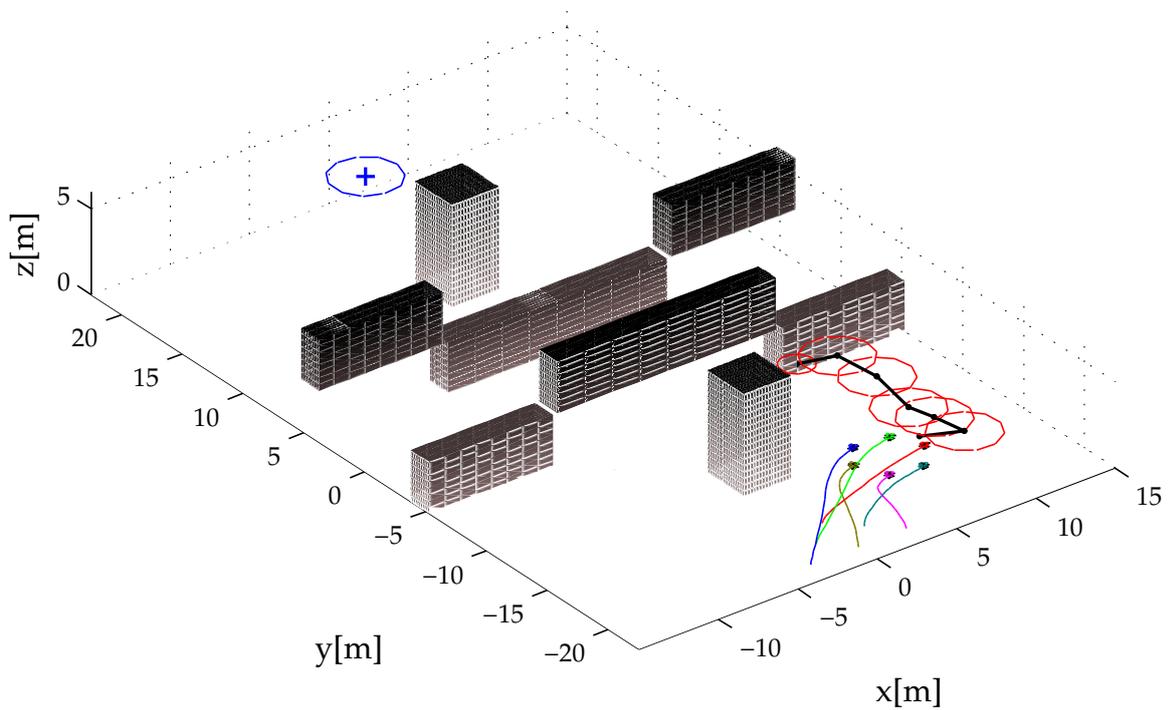
$$\dot{\mathbf{a}}(\epsilon) = \langle \dot{\mathbf{g}}, \dot{\mathbf{s}} \rangle (\epsilon) = \langle |\dot{\mathbf{p}}_c| \text{dir}(\beta), \mathbf{s}_{\text{rand}} \rangle, \quad (5.13)$$

onde β é o ângulo que aponta a direção de propagação da elipse (linha 6 do Algoritmo 4.1) e \mathbf{s}_{rand} é um vetor de forma, escolhido aleatoriamente de modo a evitar colisões com os obstáculos.

Uma vez modelada a abstração e conhecidos os modelos dinâmicos dos robôs do time, é possível partir para a simulação do arcabouço proposto. As Figuras 5.2, 5.3 e 5.4 apresentam o resultado de uma execução do método no cenário com seis veículos. A Figura 5.2(a) mostra os robôs em suas configurações iniciais (parados no solo), antes da execução do laço principal do Algoritmo 4.2. A gradação de sombra dos obstáculos corresponde à altura deles no espaço (quanto mais escuro, mais alto). A elipse tracejada em azul representa o α_{alvo} .



(a) Tempo de simulação = 00,0 s



(b) Tempo de simulação = 01,1 s

Figura 5.2. Aplicação da metodologia proposta ao planejamento e controle em tempo real de um time composto de seis MicroUAVs: (a) estado inicial e (b) propagação inicial da árvore e execução por parte dos veículos.

As árvores de referência \mathcal{T}_α e de estados \mathcal{T}_x^i de cada robô R_i começam a ser expandidas através do ambiente. Após certa expansão, o melhor caminho α^* é calculado (linha preta e contínua na Figura 5.2(b)). Foi utilizado como critério de escolha para α^* a métrica que minimiza a distância do mesmo até o objetivo. Isso pode eventualmente causar alterações abruptas na escolha da melhor referência a ser seguida em tempo real (conforme visto na Figura 5.3(a)). Esse problema é contornado pela aplicação da linha 16 do Algoritmo 4.2. O time então segue em formação o melhor caminho corrente.

O caminho planejado atinge o alvo em relativamente pouco tempo (Figura 5.3(b)) e leva o time ao seu objetivo de maneira coordenada e livre de colisões, conforme pode ser visto na Figura 5.4.

Nessa etapa, foram realizados ainda dois conjuntos de testes, com 1000 simulações cada, utilizando diferentes sementes pseudoaleatórias no planejamento. Foram utilizados ainda 20 ambientes distintos (com 5 a 15 obstáculos), cada um servindo a 50 experimentos. No primeiro conjunto, não foi utilizada a heurística de correção, proposta no Algoritmo 4.2 (linha 14), ao passo que no segundo, foi aplicada a heurística.

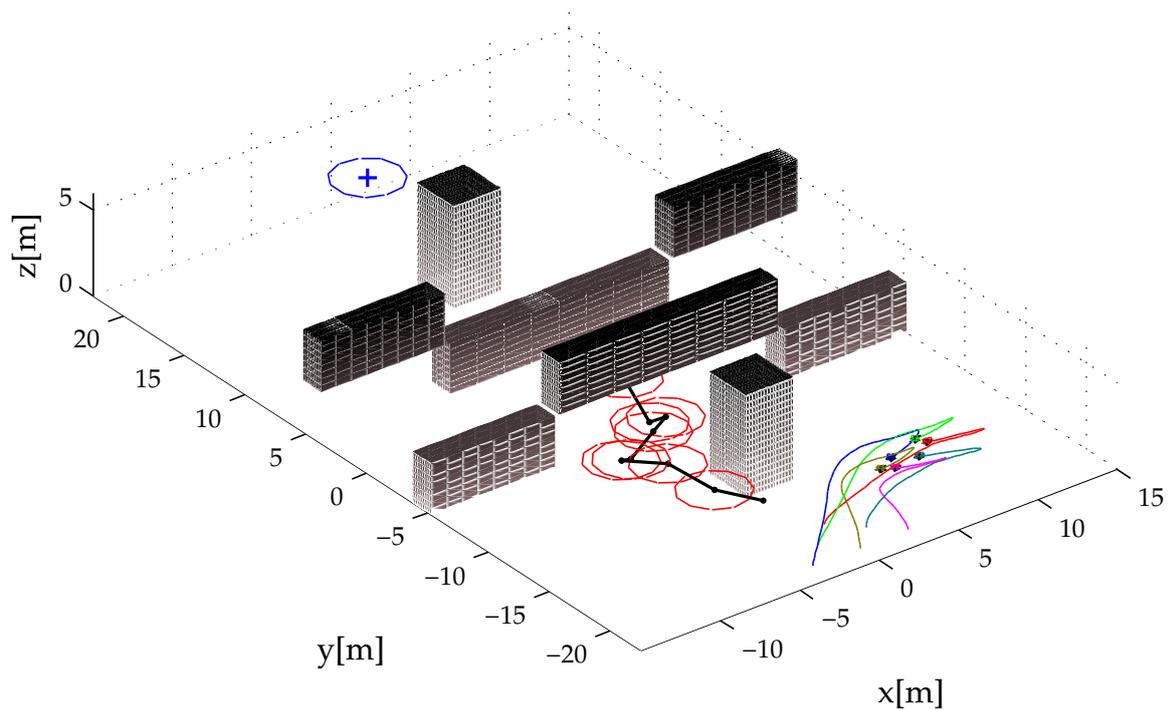
A função $\varrho(\cdot)$ foi definida como combinação linear das diferenças entre as posições e velocidades preditas e medidas por cada robô,

$$\alpha_r(t) = \alpha^*(t) + \phi \left(k_p (\hat{\mathbf{p}}_i - \mathbf{p}_i) + k_v (\hat{\mathbf{v}}_i - \mathbf{v}_i) \right) \quad (5.14)$$

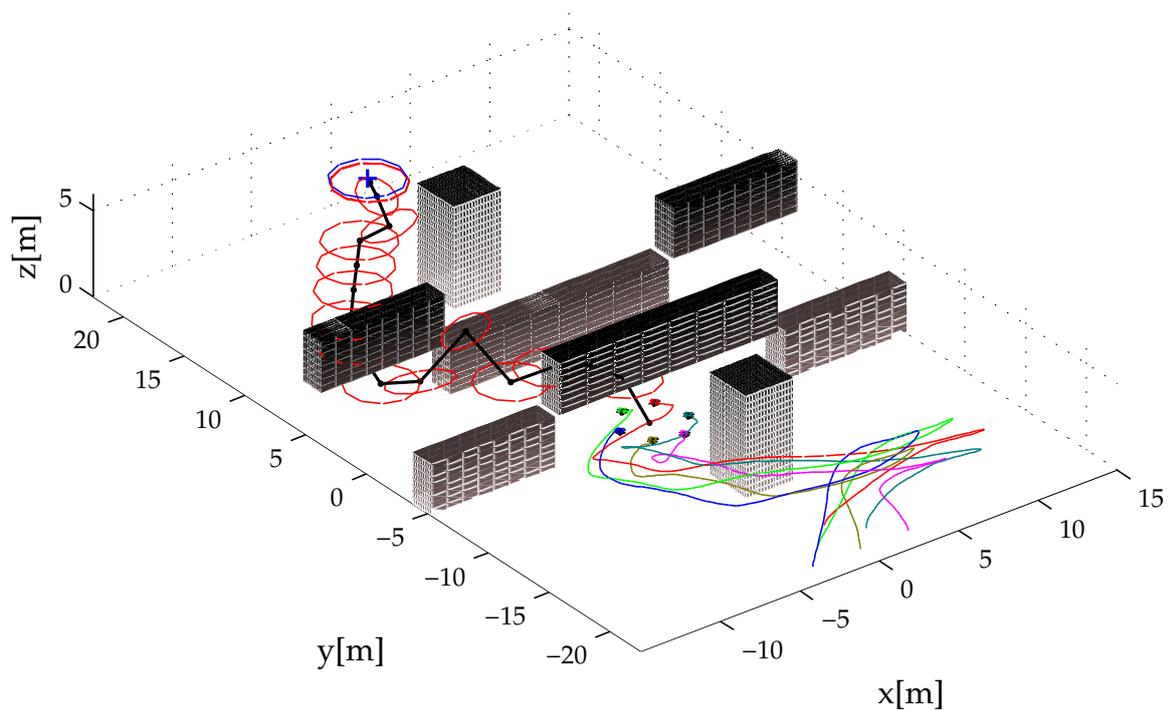
onde k_p e k_v são ganhos escalares positivos, cujos valores devem ser cuidadosamente escolhidos para manter a estabilidade do sistema em malha fechada. Finalmente, α_r representa a referência modificada, aplicada ao controlador do quadricóptero nas Equações 5.10 e 5.11, com base no parâmetro da abstração atribuído a cada veículo.

A Figura 5.5 apresenta o valor médio (para todos os robôs), ao longo do tempo, dos erros absolutos de predição no experimento das Figuras 5.2, 5.3 e 5.4. A linha tracejada representa o erro sem a etapa de correção. Nesse caso, a referência do melhor caminho, α^* , é aplicada diretamente ao controlador de cada robô. A linha contínua mostra o mesmo erro quando utilizada a etapa de correção. É possível perceber uma diminuição significativa no valor do erro médio, o que significa que cada robô navegou mais próximo da trajetória desejada. Como consequência disso, o time reduziu também o tempo de execução da missão.

A Figura 5.6 mostra uma representação estatística da comparação entre os conjuntos de experimentos realizados. O Histograma 5.6(a) apresenta a distribuição da porcentagem de redução no erro de predição em relação ao uso da heurística de

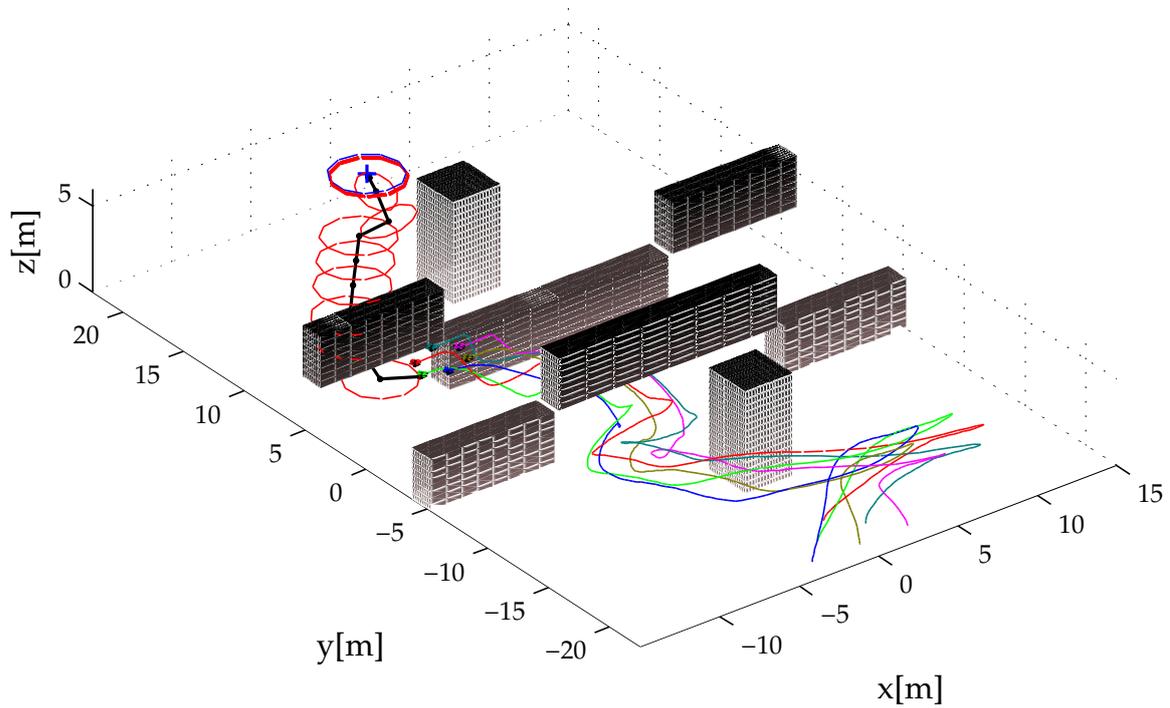


(a) Tempo de simulação = 08,7 s

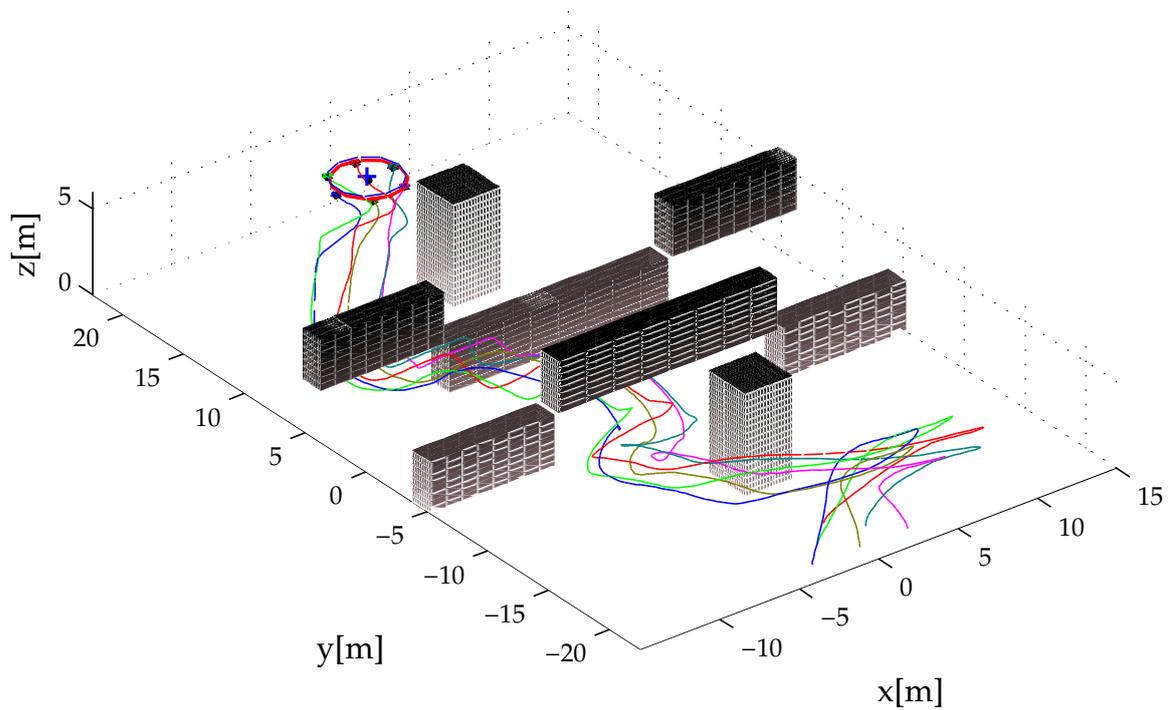


(b) Tempo de simulação = 24,6 s

Figura 5.3. Aplicação da metodologia proposta ao planejamento e controle em tempo real de um time composto de seis MicroUAVs: (a) mudança abrupta do melhor caminho e (b) árvore de referências atinge o alvo.



(a) Tempo de simulação = 51,7 s



(b) Tempo de simulação = 60,1 s

Figura 5.4. Aplicação da metodologia proposta ao planejamento e controle em tempo real de um time composto de seis MicroUAVs: (a) aproximação e (b) chegada ao alvo.

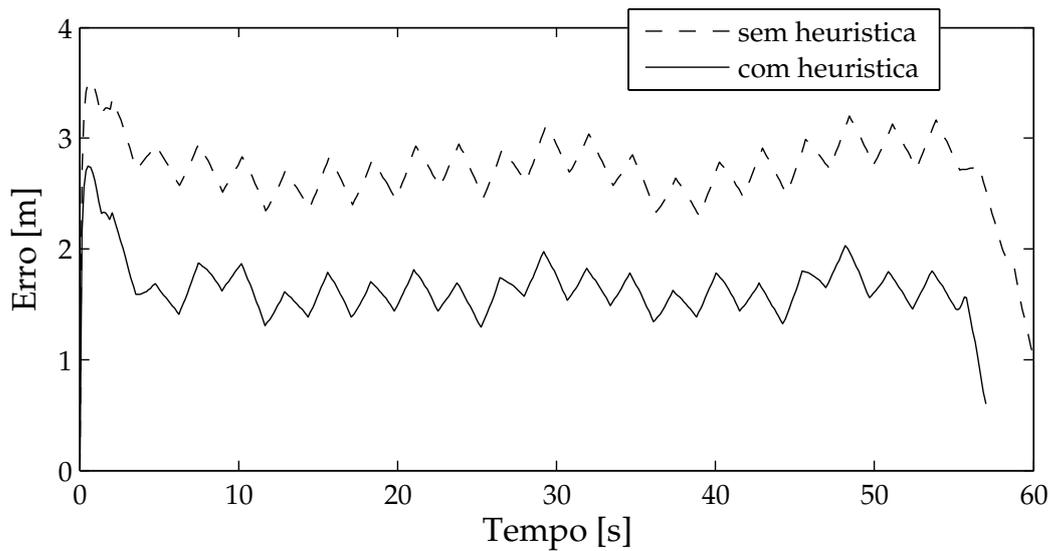
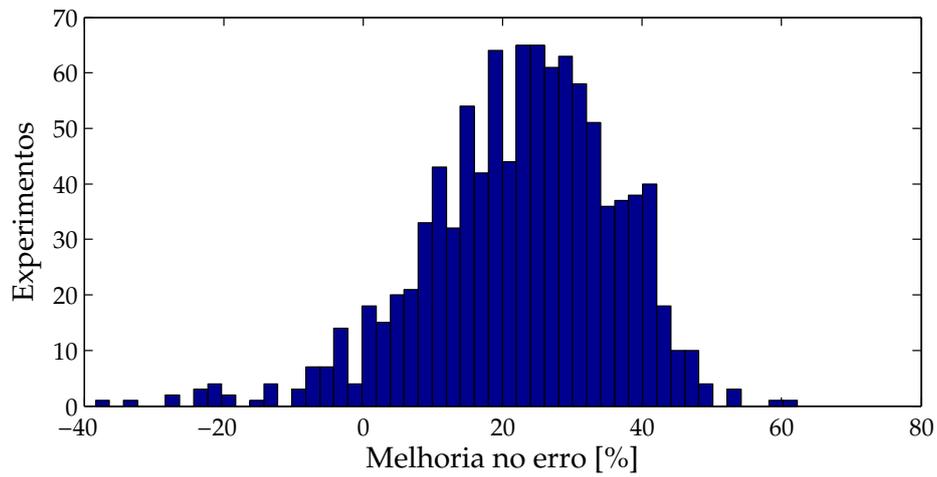


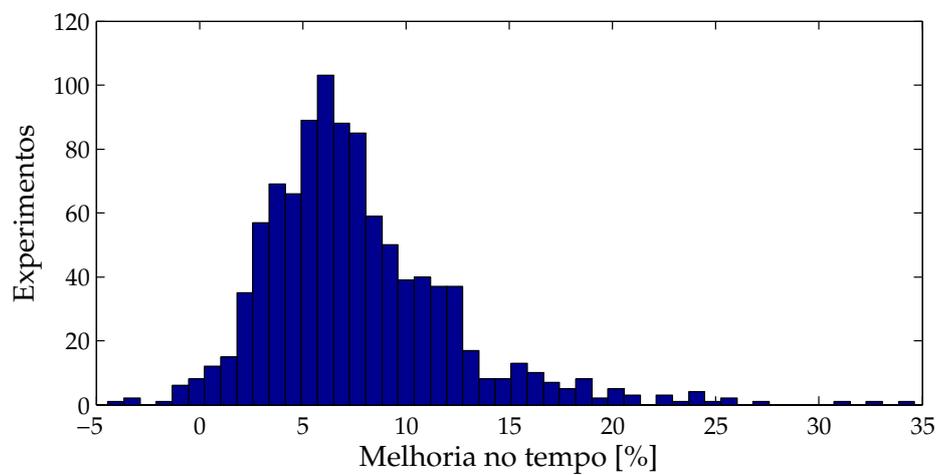
Figura 5.5. Erro médio de previsão dos veículos do time, ao longo de suas trajetórias, com (linha contínua) e sem (linha tracejada) aplicação da heurística de correção do Algoritmo 4.2 (linha 14).

correção. Na média, houve uma diminuição de cerca de 23% no erro ao longo da trajetória, indicando que o time se comportou melhor segundo o planejado. Essa melhoria ainda se deu em 95% das comparações realizadas.

A heurística promoveu ainda uma redução média no tempo de execução da missão de aproximadamente 8%, conforme apresentado na Figura 5.6(b). A redução ocorreu ainda em cerca de 98% das comparações.



(a)



(b)

Figura 5.6. Comparação entre experimentos com e sem aplicação da heurística: (a) histograma da redução no erro de previsão médio para o time de 6 UAVs; (b) histograma da redução no tempo da missão para o time.

5.3.2 Curvas de Bézier como Abstração

A utilização de algoritmos do tipo PRM permite o planejamento de movimento dos mais variados modelos em ambientes com grande número de obstáculos. Devido a essa característica, nosso método permite o uso de diferentes curvas como abstrações para o controle de grupos de robôs, desde que a forma analítica das mesmas seja conhecida. O principal desafio para o uso da metodologia proposta aqui é encontrar uma forma de propagar a abstração através do ambiente, definindo assim um modelo para $\dot{\mathbf{a}}$. É possível inclusive se utilizar de funções cujos parâmetros não apresentem representações físicas diretas, como é o caso das *Curvas de Bézier*.

Curvas de Bézier apresentam como principal vantagem, a possibilidade de se definir estruturas virtuais complexas para o controle de formação, utilizando para isso poucos parâmetros. Em outras palavras, é possível desenhar qualquer forma geométrica a ser executada pela formação.

Sendo assim, considere a seguinte função paramétrica

$$\mathbf{a}(\epsilon) = \langle \mathbf{g}, \mathbf{s} \rangle (\epsilon) = \sum_{i=0}^b \mathbf{p}_i B_i^b(\epsilon), \quad (5.15)$$

onde b é a ordem polinomial da curva, \mathbf{p}_i são pontos de controle e

$$B_i^b(\epsilon) = \binom{b}{i} (1 - \epsilon)^{b-i} \epsilon^i, \quad 0 \leq \epsilon \leq 1 \quad (5.16)$$

é o chamado Polinômio de *Bernstein*. Utilizamos $b = 5$ em nossos experimentos, de modo que a curva planejada tivesse pelo menos um ponto de inflexão.

É possível perceber que a forma padrão dessas curvas não apresenta explicitamente as características de configuração e forma, como no caso da elipse. Para contornar parte desse problema, é possível reescrever a Equação 5.15 da seguinte maneira,

$$\mathbf{a}(\epsilon) = \mathbf{p}_c + \sum_{i=0}^b \Delta \mathbf{p}_i B_i^b(\epsilon), \quad (5.17)$$

onde $\mathbf{p}_i = \mathbf{p}_c + \Delta \mathbf{p}_i$, com \mathbf{p}_c representando o centroide da curva e $\Delta \mathbf{p}_i$ correspondendo a pequenas variações em torno de cada um dos $b+1$ pontos de controle. Com base na propagação do centroide dessa curva de Bézier, é possível definir a variação da abstração no espaço de maneira bastante semelhante à Equação 5.13,

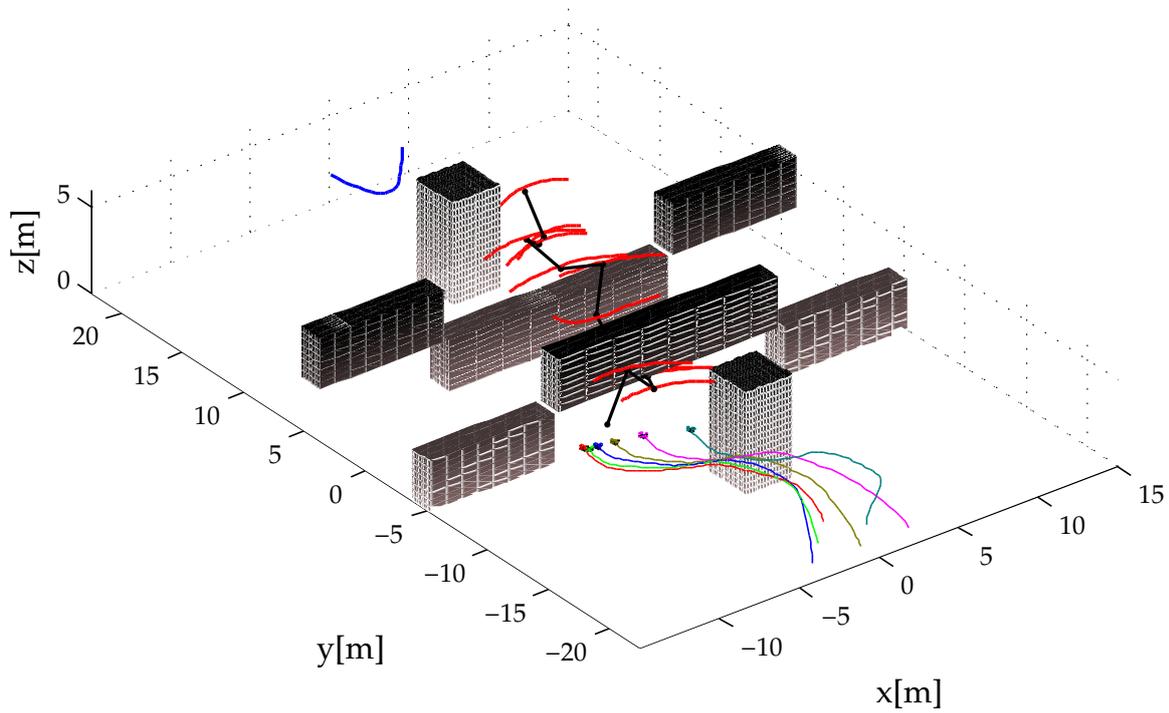
$$\dot{\mathbf{a}}(\epsilon) = \langle \dot{\mathbf{g}}, \dot{\mathbf{s}} \rangle (\epsilon) = \langle |\dot{\mathbf{p}}_c| \text{dir}(\beta), \Delta \mathbf{p}_i^{\text{rand}} \forall i = 0 \dots b \rangle, \quad (5.18)$$

onde novamente a variação do centroide é escolhida como uma constante e β é a direção que leva do vértice α_{prox} para α_{rand} . As variações dos pontos de controle $\Delta \mathbf{p}_i$ são escolhidas aleatoriamente, segundo a função pseudoaleatória conhecida previamente.

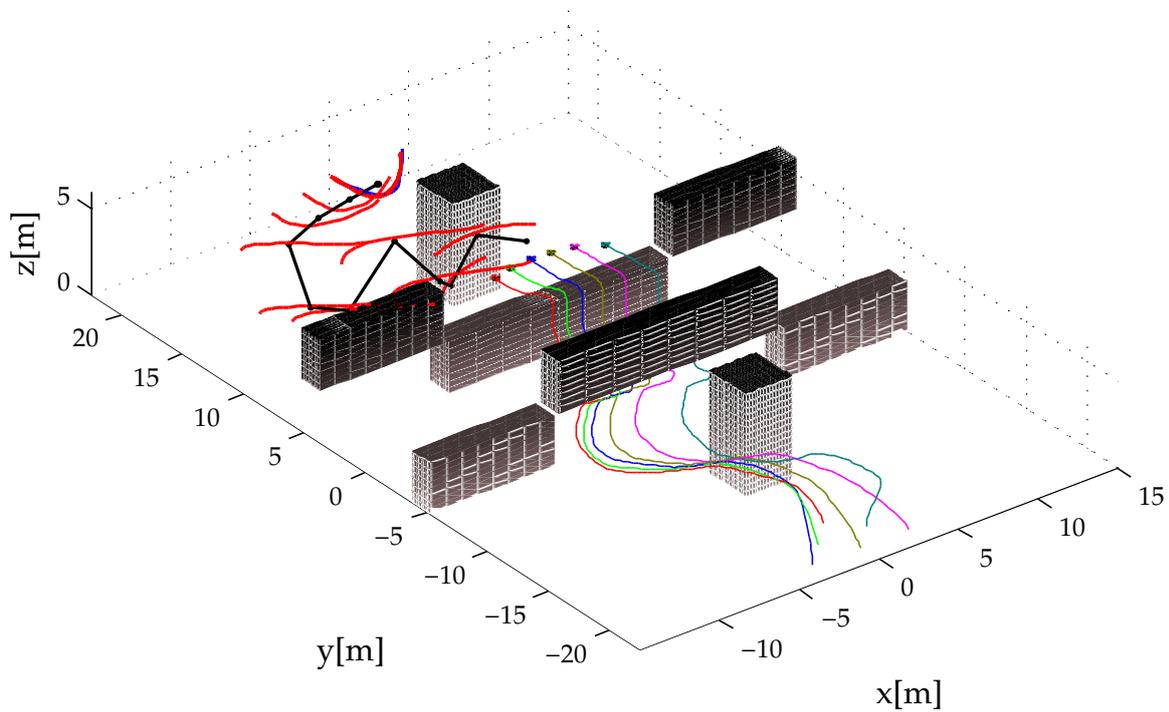
As Figuras 5.7 e 5.8 apresentam etapas da aplicação do método proposto ao controle do time de seis UAVs, utilizando curvas de Bézier como abstração. Novamente os robôs cruzam o ambiente, em formação, evitando colisões com obstáculos e com outros agentes.

Foram repetidos os testes para avaliação do erro de predição nesse caso, utilizando a mesma função de correção anterior (Equação 5.14). A Figura 5.9 mostra uma representação estatística da comparação entre 500 simulações realizadas. O Histograma 5.9(a) apresenta a distribuição da porcentagem de redução no erro de predição em relação ao uso da heurística de correção. Na média, houve uma diminuição de cerca de 14% no erro ao longo da trajetória. A melhoria ocorreu em 83% das comparações realizadas.

A heurística promoveu novamente uma redução média no tempo de execução da missão de aproximadamente 5%, conforme apresentado na Figura 5.9(b). A redução ocorreu ainda em cerca de 85% das comparações.

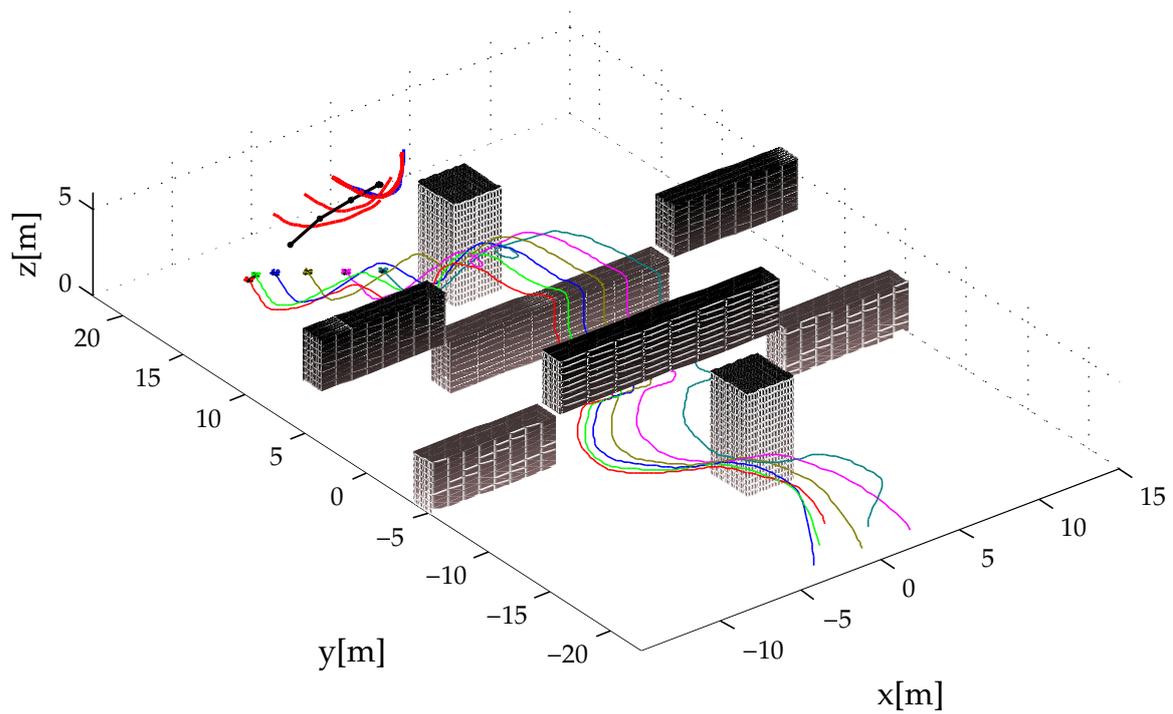


(a) Tempo de simulação = 11,2 s

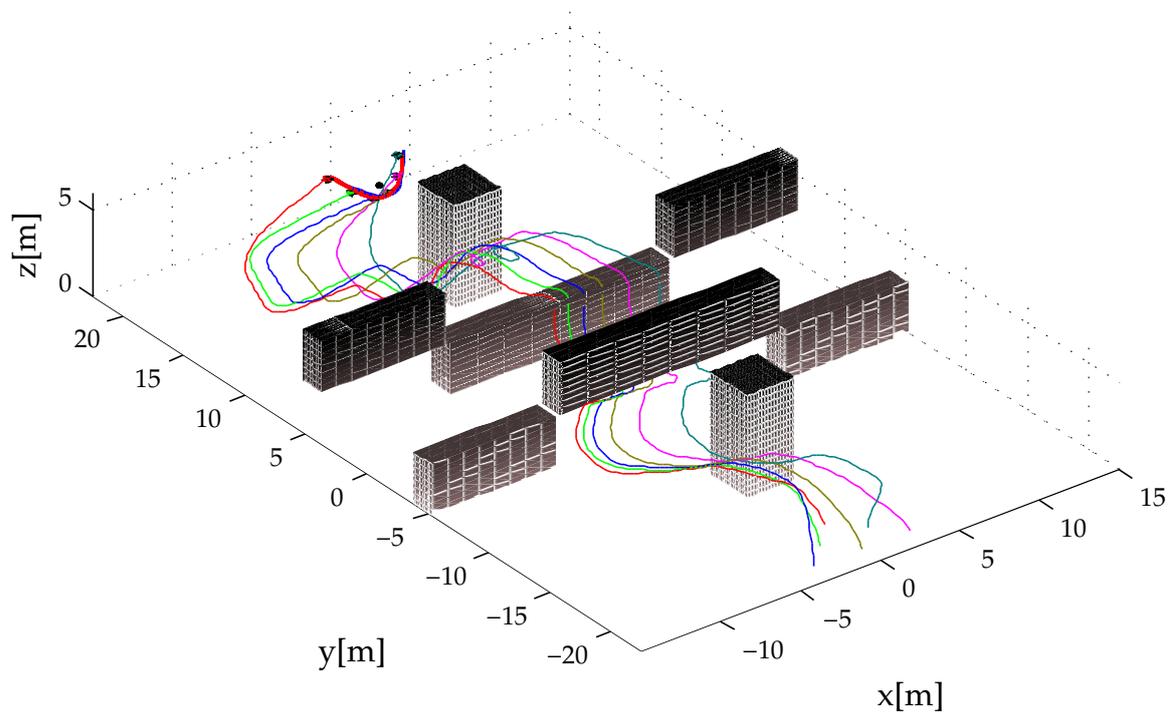


(b) Tempo de simulação = 24.9 s

Figura 5.7. Aplicação da metodologia proposta ao planejamento e controle em tempo real de um time composto de seis MicroUAVs (curvas de Bézier): (a) propagação inicial da árvore e execução por parte dos veículos e (b) árvore de referências atinge o alvo.

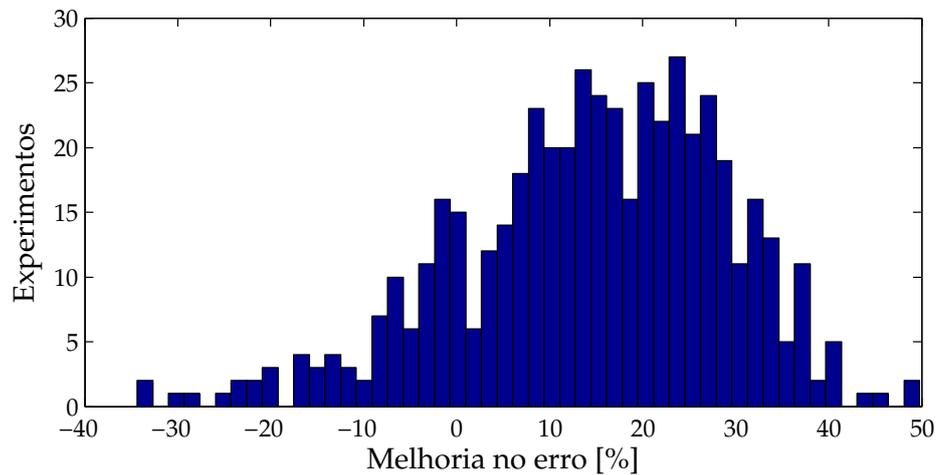


(a) Tempo de simulação = 47,0 s

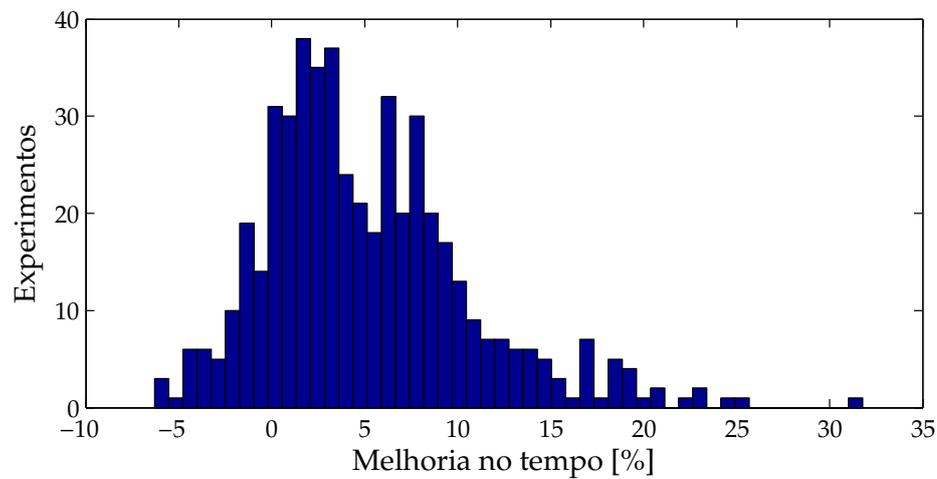


(b) Tempo de simulação = 55,4 s

Figura 5.8. Aplicação da metodologia proposta ao planejamento e controle em tempo real de um time composto de seis MicroUAVs (curvas de Bézier): (a) aproximação e (b) chegada ao alvo.



(a)



(b)

Figura 5.9. Comparação entre experimentos com e sem aplicação da heurística no uso de curvas de Bézier: (a) histograma da redução no erro de predição médio para o time de 6 UAVs; (b) histograma da redução no tempo da missão para o time.



Figura 5.10. Plataforma MultiUAVs do VeRLab: foram utilizados três quadrirotores *Hummingbird/Asctec* nos experimentos reais.

5.4 Experimentos com Robôs Reais

Nesta seção são apresentados experimentos utilizando robôs reais em um ambiente externo sujeito a perturbações. A plataforma de voo empregada nos testes foi o quadrirotor *Hummingbird Autopilot*, vendido pela *Ascending Technologies (Asctec)* [AscTec Hummingbird AutoPilot, 2011]. Os três veículos, apresentados na Figura 5.10, fazem parte do arcabouço experimental disponível no VeRLab e são equipados com uma série de sensores que auxiliam na navegação autônoma ou remotamente operada.

Uma bússola de três eixos é utilizada na estabilização látero direcional do robô em relação ao eixo magnético da Terra. Um barômetro provê medições da altitude, e de sua variação, para o controle de posição vertical. Uma Unidade de Medição Inercial (*Inertial Measurement System*, ou IMU), composta de três giroscópios e três acelerômetros dispostos de forma ortogonal, fornecem medições de velocidade angular e aceleração translacional a taxa de aproximadamente 330 Hz. Esses dados são utilizados no controle de atitude do robô, executando a frequência de ≈ 1 kHz. Além disso, um receptor Sistema de Posicionamento Global (*Global Positioning System*, ou GPS) fornece informação para o controle de posição em ambientes externos.

Cada quadricóptero possui ainda um canal sem fio, do tipo *XBee*, para propósitos de comunicação e duas CPUs embarcadas, das quais uma é inteiramente programável.

Foi utilizado ainda uma extensão do modelo matemático de quadricópteros proposto em [Michael et al., 2010] para compor a base experimental discutida a seguir. Nesse trabalho, os autores introduzem o arcabouço para múltiplos Micro-UAVs do *Grasp/Upenn Laboratory* e apresentam uma série de estratégias de controle e coordenação de múltiplos quadricópteros. Elas vão desde a modelagem dinâmica do veículo em $SE(3)$ até a coordenação anticolisão e com considerações de interação aerodinâmica entre agentes. O trabalho provê boa modelagem e parametrização para os UAVs especificamente usados em nossos testes.

Na sequência são apresentados resultados da nossa metodologia aplicada a times de robôs reais, compostos de três quadricópteros (mostrados na Figura 5.10). Foram utilizadas as mesmas configurações para o modelo de abstração da Seção 5.3.1 (elipse) e para a função de correção descritas anteriormente. Foi utilizado ainda controladores de atitude e de posição, instanciados nos processadores de bordo dos veículos, para guiá-los através do ambiente. A comunicação e a telemetria dos robôs foram fornecidas pelo módulo *Asctec Autopilot* da plataforma *Robot Operating System* (ROS) [ROS Asctec Autopilot Driver, 2011] e por outro *software* desenvolvido na linguagem *Python* exclusivamente para este trabalho.

Devido a limitações computacionais dos sistemas embarcados nos quadricópteros, foi utilizado um computador portátil para calcular três instâncias separadas da metodologia, emulando sistemas descentralizados. Os planos foram novamente calculados utilizando o *Matlab*, e a realimentação de posição dos agentes e as referências modificadas α_r foram enviadas do e para os quadricópteros, respectivamente, via canal de comunicação *Zigbee*².

O ambiente de testes foi um espaço arborizado próximo ao prédio da Reitoria da UFMG, com aproximadamente 50,0 metros de comprimento por outros 50,0 de largura, tomado por árvores das mais variadas formas e tamanhos, conforme mostrado na Figura 5.11. A Figura 5.11(a) apresenta determinado passo da expansão da árvore \mathcal{T}_α através do ambiente. As elipses em vermelho indicam a propagação da abstração pelo melhor caminho encontrado até o alvo, em azul. Já a Figura 5.11(b) apresenta as trajetórias dos UAVs através do espaço de trabalho, seguindo o plano pré-estabelecido.

²Dispositivo eletrônico de comunicação sem fio de baixa potência (padrão IEEE 802.15.4).

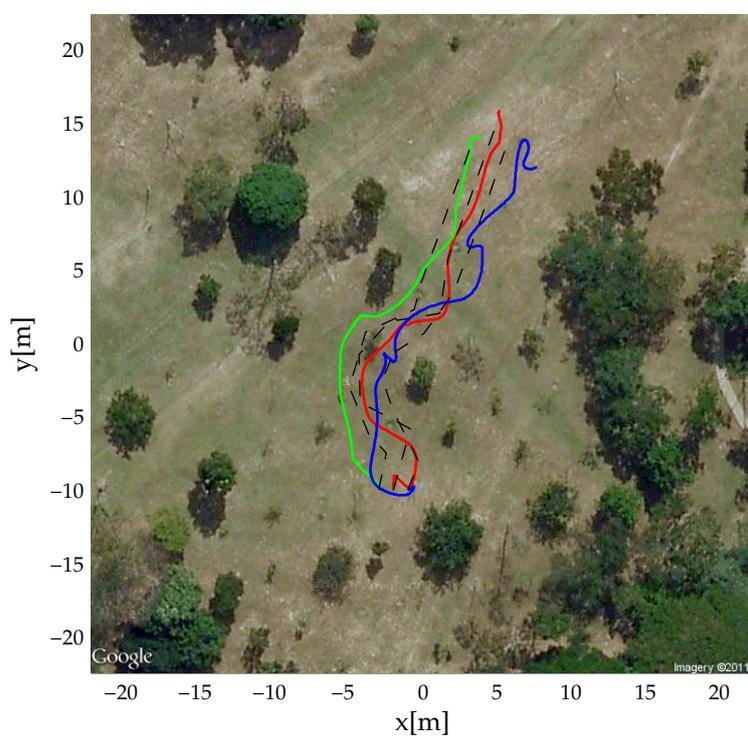
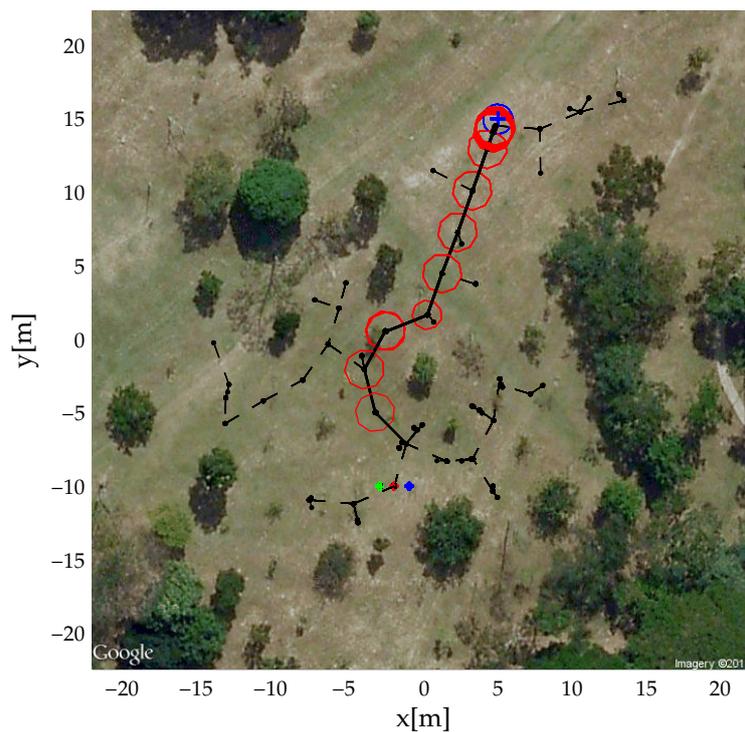


Figura 5.11. Experimento real com o time de quadrirotores do VeRLab (figuras impressas a partir de imagens do *Google Maps*): (a) expansão da árvore de referências pelo ambiente e (b) trajetórias finais dos robôs.

5.5 Observações Finais

Este capítulo apresentou a aplicação da Árvore Pseudoaleatória de Exploração Rápida Multiagente ao problema de controle de formação de um grupo de robôs em ambientes com obstáculos. Foi mostrado que é possível utilizar uma estratégia simples, baseada em comportamento aleatório, para cumprir a missão.

O uso de estratégias de controle delegadas às camadas de mais baixo nível do sistema facilita o planejamento em malha fechada. Porém, é preciso garantir que os robôs sejam capazes de seguir o plano para que a tarefa seja cumprida com sucesso. Os experimentos demonstraram que o uso da heurística de correção auxilia nesse aspecto, reduzindo erros e desvios no comportamento do time. As análises apresentadas mostraram melhorias significativas, tanto na qualidade da execução da missão quanto no tempo de realização dessa.

Por fim, foi mostrado que é possível utilizar diferentes tipos de curvas paramétricas no planejamento de formação. Tanto funções com características de configuração e forma bem definidas (como as elipses) quanto as não tão bem definidas (curvas de Bézier) produziram bons resultados em termos da realização da tarefa. Embora deformáveis, as elipses utilizadas ainda apresentam características de forma semirígida, como é o caso de outras estruturas virtuais utilizadas em controle de formação. As curvas de Bézier, porém, são mais flexíveis e fáceis de se ajustar às restrições geométricas do ambiente.

Capítulo 6

Aplicação à Cooperação Aéreo-terrestre

– Cedo ou tarde, você irá aprender, assim como eu aprendi, que existe uma diferença entre conhecer o caminho e trilhar o caminho.

Morpheus, The Matrix

Este capítulo apresenta a aplicação do método a outro tipo de problema encontrado em Robótica Cooperativa: a cooperação entre veículos aéreos e terrestres em missões de exploração. Esse é um problema bastante complexo, uma vez que os agentes do time apresentam diferenças significativas entre si, não apenas do ponto de vista dinâmico, mas também de sensoriamento. Geralmente, enquanto robôs aéreos possuem uma visão mais privilegiada (global) do ambiente provida por sensores exteroceptivos, robôs terrestres têm melhores condições de atuação (local) no meio. Foram utilizadas características em uma missão que envolve escolta de um grupo de UGVs por parte de um UAV, em cenários com obstáculos. Ao final, são apresentados resultados para casos experimentais simulados e reais.

6.1 Introdução

No problema da cooperação aéreo-terrestre, tanto o espaço de atuação quanto a dinâmica e capacidade de sensoriamento dos agentes do time varia muito, de modo que a dificuldade em se estabelecer métodos para a execução de tarefas com esse time heterogêneo aumenta consideravelmente. Um dos primeiros trabalhos a lidar com a complexidade do problema de coordenação aéreo-terrestre foi [Elfes et al., 1999]. Os autores apresentam um projeto (em fase preliminar) para aplicações de cooperação entre múltiplos veículos, para fins de pesquisa e monitoramento ambiental. Já em [Vaughan et al., 2000], os autores tem como foco a localização de Micro

Veículos Aéreos Não tripulados (MUAVs) por meio da cooperação com UGVs. MUAVs geralmente apresentam baixa capacidade de carga, o que implica em redução no sensoriamento do sistema. Nesse trabalho, veículos terrestres equipados com receptores de GPS informam sua posição aos veículos aéreos, que por sua vez são capazes de se localizar imagiando (por meio de uma câmera simples) esses UGVs.

Em [Chaimowicz & Kumar, 2004] os autores propõem um paradigma no qual um grupo de veículos aéreos monitora e auxilia grupos de robôs terrestres na tarefa de navegação em ambientes com obstáculos. Esses UAVs possuem sensores (câmeras) capazes de monitorar a posição dos UGVs dentro de um enxame. Experimentos reais são apresentados em [Chaimowicz et al., 2004], onde é utilizado um robô aéreo (dirigível) equipado com câmera, entre outros sensores, para a localização de robôs terrestres. Em [Chaimowicz et al., 2005] os autores apresentam uma plataforma de testes para a aplicação de times de múltiplos robôs em ambientes urbanos. O trabalho se concentra especificamente na aplicação a missões como manutenção da conectividade entre veículos, *situational awareness* e sensoriamento cooperativo aéreo-terrestre (esse último é mais bem detalhado em [Grocholsky et al., 2004]).

Em [Tanner, 2007; Tanner & Christodoulakis, 2007] são propostas estratégias de coordenação de grupos de veículos heterogêneos baseada em controle descentralizado. Inicialmente é apresentado um controlador para manter a formação e evitar colisões em um grupo de UGVs, se movendo para uma missão pré-estabelecida. O grupo fornece uma estimativa da posição de seu centroide, via comunicação direta, para um grupo de UAVs, que usa essa informação como referência para seu próprio posicionamento. Outro controlador mantém os veículos aéreos em trajetória circular ao redor do centroide, evitando colisões e promovendo vigilância em torno do grupo de robôs terrestres.

No trabalho [Michael et al., 2007] os autores também consideram o paradigma de controle de múltiplos UGVs utilizando UAVs. Aqui, um único veículo aéreo é considerado para a realização dessa tarefa e a estratégia proposta permite que o UAV controle os UGVs por meio de uma *abstração*, sem que para isso haja a necessidade de se conhecer a configuração de cada robô terrestre. Essa abstração modela o time de robôs terrestres como uma elipse, cujo formato e a posição do centro são controlados pelo UAV em uma abordagem descentralizada.

Um arcabouço para missões de localização e combate a incêndios baseada em cooperação aéreo-terrestre é apresentada em [Phan & Liu, 2008]. Nesse trabalho, um UAV do tipo dirigível atua como um agente centralizador na coordenação de grupos de veículos aéreos e terrestre. Após localizar um incêndio e calcular seu modelo de propagação, o dirigível escolhe a melhor forma de combater o problema utilizando

o time.

6.2 Definição do Problema

Da mesma forma que no capítulo anterior, é possível redefinir o Problema 4.1 de modo a permitir a aplicação da técnica proposta no Capítulo 4. A missão considerada aqui é a mais comum entre os artigos apresentados anteriormente, escolta aérea de veículos terrestres.

Problema 6.1 (Planejamento em escolta descentralizada). *Considerando o Problema 4.1, onde um grupo de robôs \mathcal{R} deve navegar através do espaço de trabalho \mathcal{W} de um estado inicial \mathbf{X}_{ini} para um estado final α_{alvo} , planejar trajetórias onde grupos de robôs aéreos promovam a escolta de grupos de robôs terrestres em ambiente com obstáculos, de maneira descentralizada e livre de colisões.*

6.2.1 Modelo Dinâmico do Veículo Terrestre

Em contrapartida ao modelo dinâmico de veículos aéreos, apresentado na Seção 5.2.1, foi considerado aqui o modelo de um veículo terrestre. Novamente foi utilizado um modelo matemático baseado no formalismo de *Newton-Euler* em um espaço de trabalho $\mathcal{W} \equiv \mathbb{R}^3$. Esse modelo é adaptado do arcabouço apresentado em [Cruz et al., 2007], onde os autores constroem um UGV a partir da mesma plataforma apresentada aqui, um veículo do tipo *Ackerman* [Weinstein & Moore, 2010].

Considere o vetor de estados $\mathbf{x} \in \mathbb{R}^{12}$, composto pelas posições e orientações do veículo no Grupo de Lie $SE(3)$, e suas respectivas derivadas, tal que

$$\mathbf{x} = \left[\mathbf{p} \quad \mathbf{v} \quad \Psi \quad \Omega \right]^T,$$

onde \mathbf{p} e Ψ são posições e orientações em $\{\mathcal{W}\} \in \mathbb{R}^3$ e $SO(3)$ respectivamente, e \mathbf{v} e Ω são derivadas desses vetores em relação à $\{\mathcal{B}\}$.

Considere ainda o vetor de entradas do sistema,

$$\mathbf{u} = [v, \delta]^T,$$

onde v representa o comando de velocidade translacional, em m/s, e δ o esterçamento (direção) do veículo *Ackerman*. A Figura 6.1 apresenta os referenciais e forças que atuam sobre o robô.

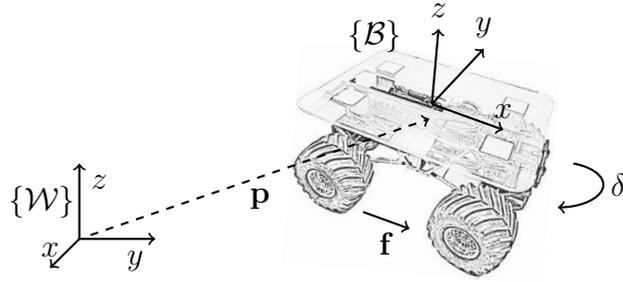


Figura 6.1. Referenciais do robô de Ackerman: forças e momentos atuantes na dinâmica do veículo.

A variação temporal do movimento translacional do robô terrestre pode ser descrita conforme apresentado nas Equações 6.1 e 6.2,

$$\dot{\mathbf{p}} = R\mathbf{v}, \quad (6.1)$$

$$m\dot{\mathbf{v}} + \eta\mathbf{v}|\mathbf{v}| + (\mathbf{v} \times m\boldsymbol{\Omega}) = \begin{bmatrix} \mathbf{f} \\ 0 \\ 0 \end{bmatrix}, \quad (6.2)$$

onde $\mathbf{p} = [x, y, z]$ é dado em metros e $\mathbf{v} = [v, 0, 0]$ em metros por segundo. Além disso, R é a matriz de rotação, m é a massa do veículo e η é o seu coeficiente de atrito cinético. Aqui, \mathbf{f} representa o vetor força de tração produzida pelos motores do carro (Figura 6.1), calculada segundo a Equação 6.3,

$$\mathbf{f} = k_c v, \quad (6.3)$$

onde k_c corresponde a um ganho proporcional positivo.

De forma semelhante, a variação temporal do movimento rotacional do robô é descrita por meio das Equações 6.4 e 6.5,

$$\dot{\boldsymbol{\Psi}} = \boldsymbol{\Omega}, \quad (6.4)$$

$$I\dot{\boldsymbol{\Omega}} + \eta\boldsymbol{\Omega}|\boldsymbol{\Omega}| + (\boldsymbol{\Omega} \times I\boldsymbol{\Omega}) = \begin{bmatrix} 0 \\ 0 \\ \frac{\mathbf{f}}{L} \tan \delta \end{bmatrix}, \quad (6.5)$$

onde $\boldsymbol{\Psi} = [\phi, \theta, \psi]^T$ é novamente dado em radianos e $\boldsymbol{\Omega} = [p, q, r]^T$ em radianos por segundo. Ainda, L é o comprimento do robô ao longo de seu eixo X .

6.2.1.1 Leis de Controle

Para controlar individualmente os UGVs ao longo do espaço de trabalho, foi utilizado um conjunto de controladores Proporcional–Integral–Derivativo (PID) não lineares, conforme descrito na sequência. Essa estratégia foi adaptada do modelo de controlador não holonômico apresentado em [Siegwart & Nourbakhsh, 2004]. Abordagens semelhantes podem ser vistas em [Cruz et al., 2007].

Assim como no capítulo anterior, foi estabelecido inicialmente o vetor de referência para os controladores do robô,

$$\mathbf{r} = [x_{\text{ref}}, y_{\text{ref}}]^T,$$

como sendo uma posição no espaço de trabalho livre $\mathcal{W}_{\text{livre}}$.

Tal qual acontece no caso do robô aéreo, o UGV possui duas malhas de controle, uma de orientação (mais interna) e outra de posição (mais externa). A malha de controle de orientação visa alinhar o robô segundo uma direção pré-estabelecida, determinada em função da configuração atual do robô e da referência almejada. A Equação 6.6 apresenta o erro de orientação do robô em relação a \mathbf{r} ,

$$\gamma = \arctan \left(\mathbf{r} - [x, y]^T \right) - \psi, \quad (6.6)$$

sendo que $\gamma \in [-\pi \dots \pi]$.

Conforme apresentado nas Equações 6.4 e 6.5, a orientação do veículo de Ackerman pode ser controlada diretamente por meio do esterçamento da direção δ . Assim, foi utilizada a seguinte lei de controle,

$$\delta = \left(\lambda_8 \gamma + \lambda_9 \int \gamma + \lambda_{10} \dot{\phi} \right), \text{ sendo } -\pi < \delta < \pi, \quad (6.7)$$

onde λ_8, λ_9 e λ_{10} são ganhos positivos de controle.

Já a malha de controle de posição é dada pela equação

$$v = \left(\lambda_5 \xi + \lambda_6 \int \xi + \lambda_7 v \right), \text{ sendo } v \geq 0, \quad (6.8)$$

onde $\xi = d \left(\mathbf{r} - [x, y]^T \right)$ é a distância euclidiana da posição do robô até o alvo.

6.2.1.2 Restrições entre Agentes

Para estabelecer funções de colisão entre os robôs em um time de veículos aéreos e terrestres, será acrescentada uma nova consideração à Suposição 4.2. Aqui, se

parte do pressuposto de que veículos terrestres podem colidir apenas com outros UGVs. O mesmo é válido para os robôs aéreos, que só podem se chocar com outros UAVs. Dessa forma, é possível manter as restrições estabelecidas no Capítulo 5 para os quadrirotos e propor a seguinte função de colisão para os veículos do tipo *Ackerman*,

$$g(\mathbf{x}_i, \mathbf{x}_j) = (x_i - x_j)^2 + (y_i - y_j)^2 - r^2$$

onde agora r corresponde ao raio do menor círculo que compreende todo o corpo de UGV.

Novamente, por razões de segurança, foi considerado um valor negativo δ_x , tal que

$$g(\mathbf{x}_i, \mathbf{x}_j) \leq \delta_x$$

seja um limiar na validação da função de colisão, servindo como margem de segurança para incertezas de medição na localização dos robôs.

6.3 Experimentos Simulados

Nessa etapa, o arcabouço de simulação construído em *Matlab* foi estendido para a execução de experimentos incluindo veículos aéreos e terrestres. O principal objetivo aqui foi avaliar a aplicabilidade da metodologia proposta a grupos de veículos heterogêneos, em missões cujos objetivos de cada agente do time possam mudar dinamicamente. Para isso, foi proposto um modelo de abstração cuja variação temporal depende do estado do sistema. Em outras palavras, foi utilizado um sistema híbrido para determinar a missão do time.

Os experimentos consistiram em utilizar um quadrirotor (cujo modelo dinâmico foi discutido na Seção 5.2.1) para interceptar e escoltar um grupo de robôs *Ackerman* de um ponto a outro do ambiente tomado por obstáculos.

6.3.1 Sistema Híbrido como Abstração

Para ampliar a generalização do método (e também por razões de simplicidade) foi escolhido modelar o comportamento do grupo de robôs por meio não de uma, mas de duas funções paramétricas compondo a mesma abstração. Considere o seguinte modelo de referência,

$$\alpha(\epsilon) = \langle \alpha_{\text{uav}}, \alpha_{\text{ugv}} \rangle,$$

onde α_{uav} descreve o comportamento do UAV e α_{ugv} dos UGVs do time. A missão definida no Problema 6.1 permite tão separação, dada as diferenças entre as plataformas, embora se acredite que seja possível utilizar uma única curva (como a de Bézier) para o caso.

Além da divisão do time em dois grupos, é possível separar também a própria missão em duas etapas: (i) a procura do UAV pelos UGVs e (ii) a escolta propriamente dita até o alvo. Em cada uma dessas etapas, as abstrações consideradas podem apresentar comportamentos diferentes, a fim de cumprir determinadas partes da missão. Assim, obtem-se um sistema híbrido bastante simples, composto por dois estados: o estado de busca e o estado de escolta.

O time inicia a missão no estado de busca, onde o robô aéreo parte a fim de encontrar o time de UGVs em determinado ponto do ambiente. A transição de estado ocorre quando o UAV atinge esse objetivo. A partir daí, tem início a escolta do grupo para a posição alvo. Nesse caso, α_{uav} é definido como sendo

$$\mathbf{a}_{\text{uav}}(\epsilon) = \mathbf{p}_{\text{uav}}, \quad (6.9)$$

e

$$\dot{\mathbf{a}}_{\text{uav}}(\epsilon) = \begin{cases} |\dot{\mathbf{p}}_{\text{uav}}| \text{dir}(\beta), & \text{no estado de busca} \\ \text{dir}(\mathbf{p}_{\text{ugv}}), & \text{no estado de escolta} \end{cases}, \quad (6.10)$$

onde $\mathbf{p}_{\text{uav}} \in \mathbb{R}^3$ é a posição do UAV no espaço, $\mathbf{p}_{\text{ugv}} \in \mathbb{R}^2$ é o centroide do grupo de UGVs e β é o mesmo ângulo apresentado no capítulo anterior. A Equação 6.10 diz que, no estado de busca, a posição do veículo aéreo é propagada aleatoriamente (segundo β) em direção ao grupo de robôs. Já no estado de escolta, ele segue o centroide da formação dos veículos terrestres, promovendo a cobertura aérea.

Já α_{ugv} pode ser definida segundo a forma elíptica,

$$\mathbf{a}_{\text{ugv}}(\epsilon) = \mathbf{p}_{\text{ugv}} + \begin{bmatrix} \cos \psi \cos \epsilon & -\sin \psi \sin \epsilon \\ \sin \psi \cos \epsilon & \cos \psi \sin \epsilon \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}, \quad (6.11)$$

$$\dot{\mathbf{a}}_{\text{ugv}}(\epsilon) = \begin{cases} 0, & \text{no estado de busca} \\ \langle |\dot{\mathbf{p}}_{\text{ugv}}| \text{dir}(\beta), \mathbf{s}_{\text{rand}} \rangle, & \text{no estado de escolta} \end{cases}, \quad (6.12)$$

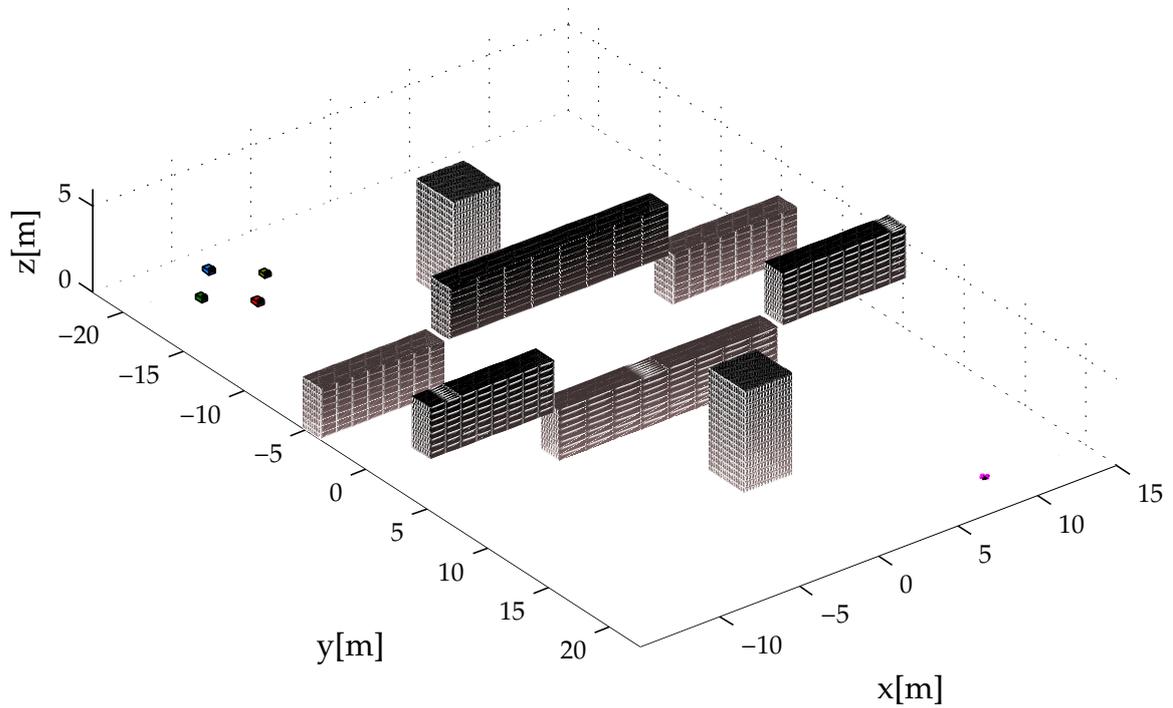
onde $\epsilon \in \mathbb{R}$ é o parâmetro da curva, ψ é a orientação do eixo principal e $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2]^T$

são as dimensões positivas dos eixos que determinam a forma da elipse. Novamente β é o ângulo que aponta a direção de propagação da elipse e s_{rand} é um vetor de forma, escolhido aleatoriamente de modo a evitar colisões com os obstáculos.

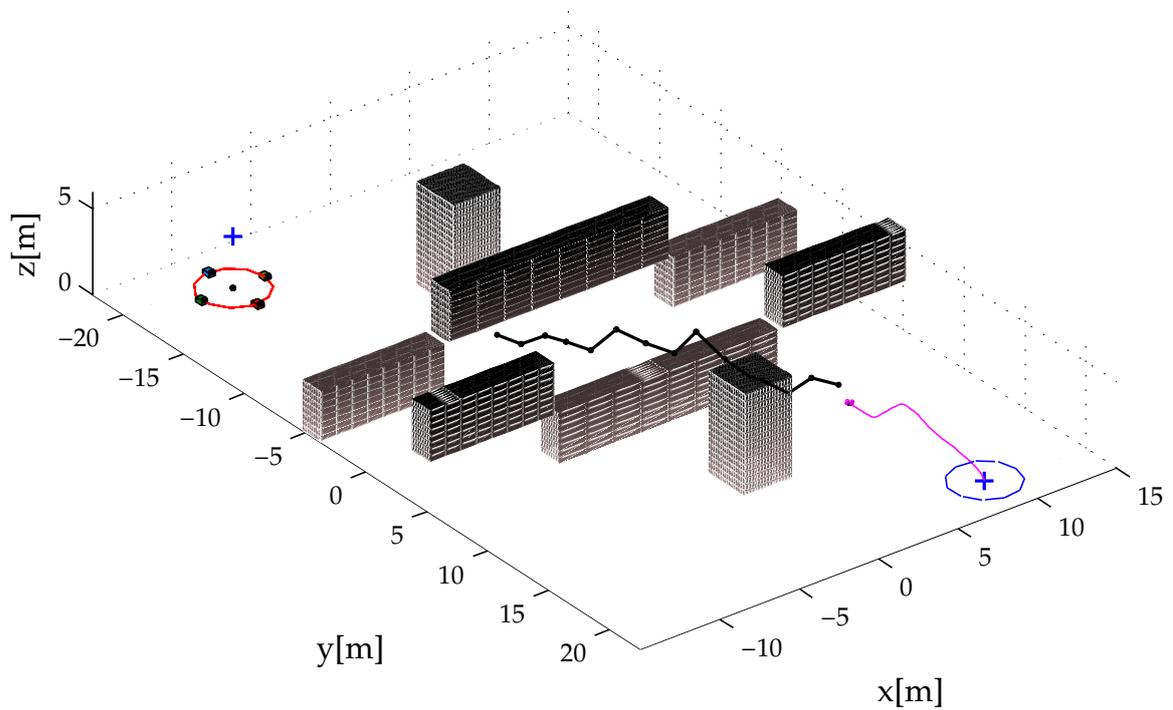
As Figuras 6.2, 6.3 e 6.4 apresentam etapas de uma das simulações envolvendo um UAV do tipo quadricóptero e quatro UGVs do tipo *Ackerman* com esse modelo de abstração. Na Figura 6.2(a), é possível ver a configuração inicial do time, com os robôs terrestres à esquerda e o aéreo à direita, parado no solo. Já a Figura 6.2(b) mostra o início da etapa de busca do UAV, cujo objetivo é a posição conhecida do grupo de UGVs.

A árvore de referência atinge o objetivo da etapa de busca, conforme visto na Figura 6.3(a). Quando o quadricóptero encontra os outros robôs, tem início a segunda etapa da missão. O veículo aéreo começa a escolta de seus companheiros de volta ao ponto inicial. A nova árvore de referências do time é apresentada na Figura 6.3(b).

A MRPT atinge o alvo na etapa de escolta, segundo apresentado na Figura 6.4(a), após modificar parte do plano da figura anterior. Por fim, o time alcança o alvo na Figura 6.4(b) e a missão termina com sucesso.

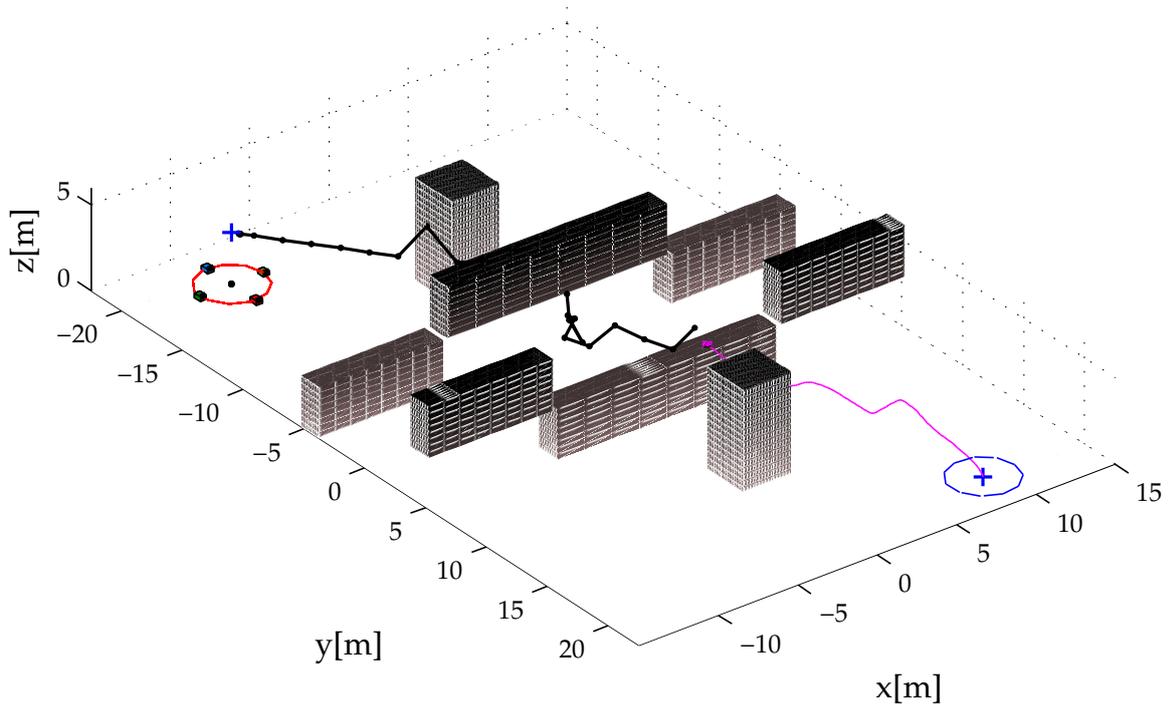


(a) Tempo de simulação = 00,0 s

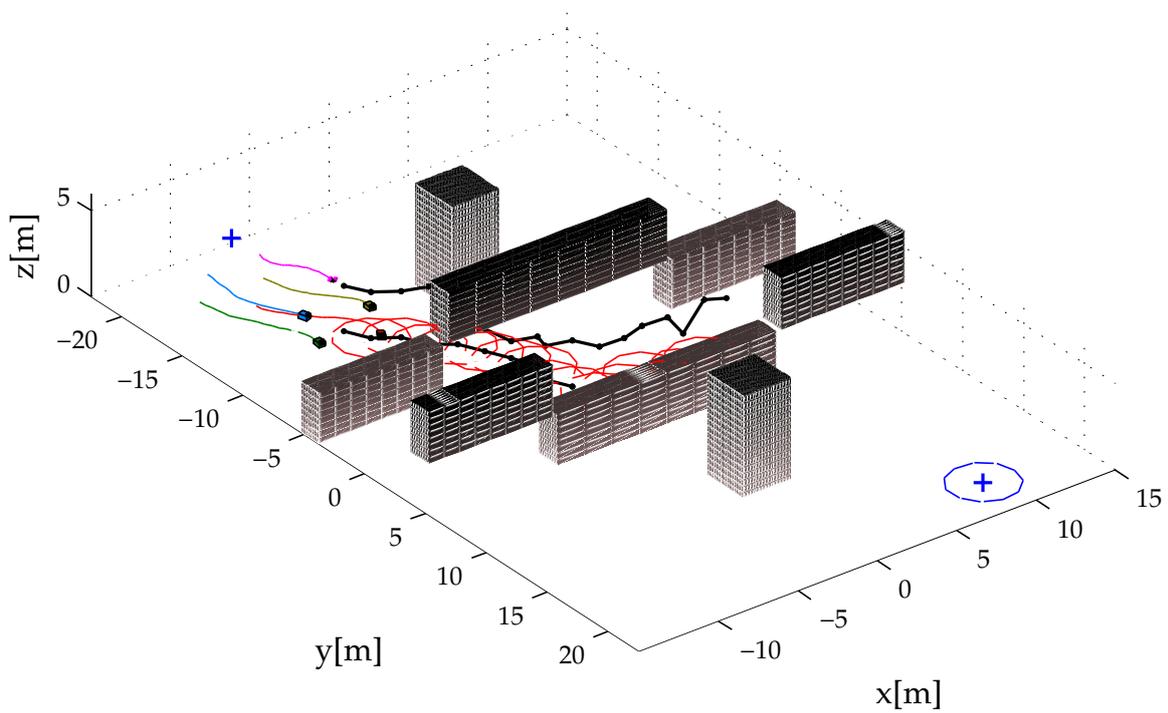


(b) Tempo de simulação = 7,4 s

Figura 6.2. Aplicação da metodologia proposta ao problema de cooperação aéreo-terrestre: (a) configuração inicial do time e (b) início da etapa de busca.

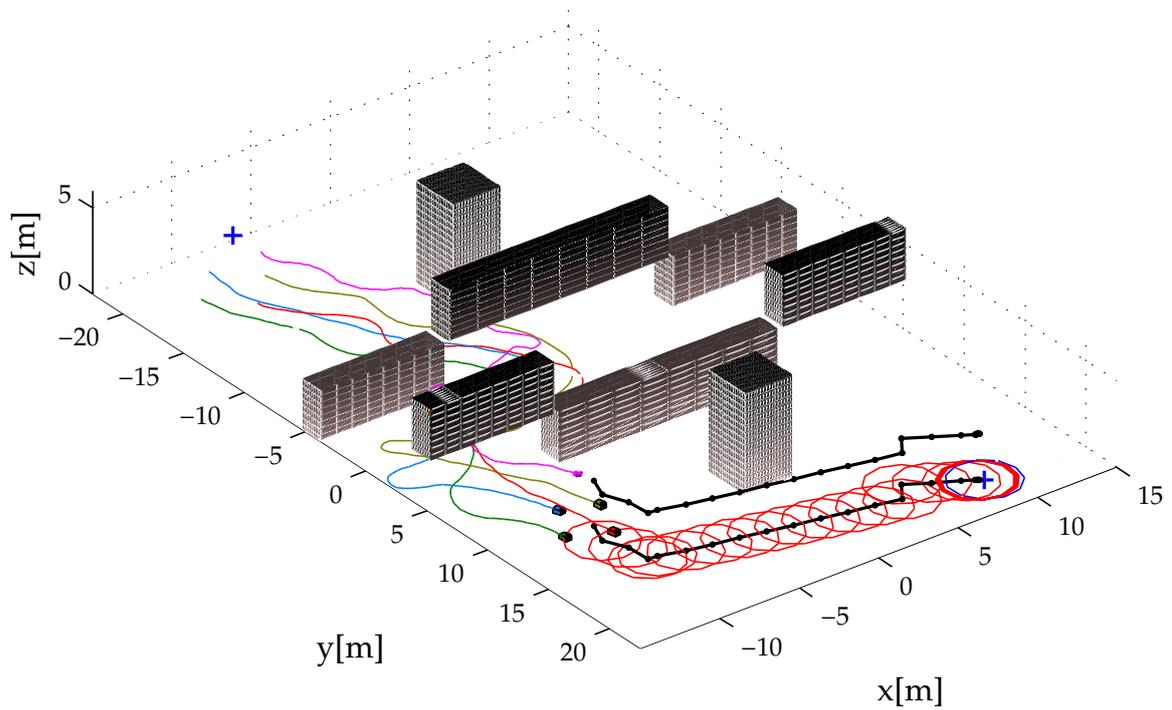


(a) Tempo de simulação = 16,5 s

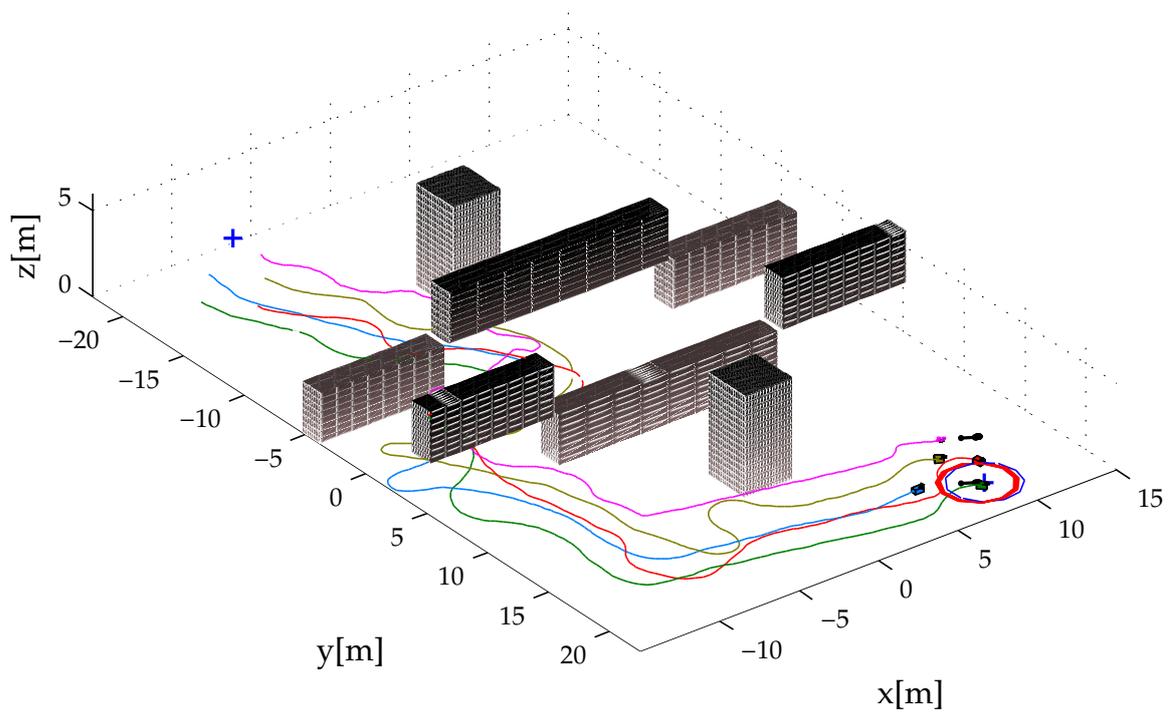


(b) Tempo de simulação = 53,8 s

Figura 6.3. Aplicação da metodologia proposta ao problema de cooperação aéreo-terrestre: (a) árvore atinge o primeiro objetivo e (b) início da etapa de escolha.



(a) Tempo de simulação = 87,0 s



(b) Tempo de simulação = 110,8 s

Figura 6.4. Aplicação da metodologia proposta ao problema de cooperação aéreo-terrestre: (a) árvore atinge o segundo e último objetivo e (b) o time chega ao alvo.

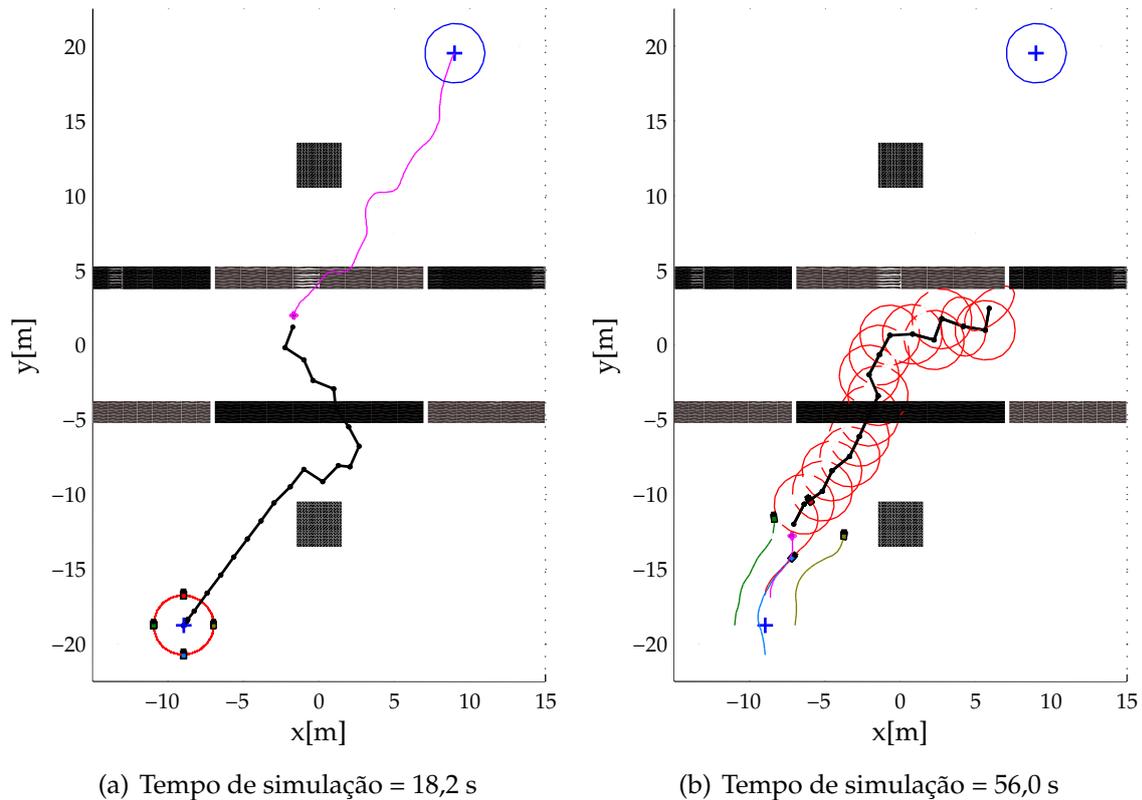


Figura 6.5. Simulação de falha em um dos agentes terrestres: (a) UAV vai de encontro aos UGVs e (b) robô amarelo falha após certo tempo.

6.3.2 Simulação de Falha

Para avaliar a tolerância a falhas do arcabouço proposto, foram repetidos os testes anteriores com a simulação de falha em um dos agentes do subgrupo de UGVs. As Figuras 6.5 e 6.6 mostram os diferentes estágios do experimento. Como na seção anterior, o quadricóptero efetua o planejamento, indo de encontro ao grupo de robôs terrestres, conforme apresentado na Figura 6.5(a). Novamente o sistema muda do estado de busca para o estado de escolta (Figura 6.5(b)), e os UGVs começam a se mover, cobertos pelo UAV.

Poucos segundos após o início da movimentação, é introduzida uma falha em um dos UGVs, de modo que esse começa a se atrasar em relação ao plano. Ele para totalmente, mas ainda se mantém conectado aos demais agentes do time. A informação de falha é transmitida ao time, que reconfigura a posição dos agentes ainda ativos em relação à abstração. Aqui, essa reconfiguração é feita recalculando-se o parâmetro ϵ correspondente de cada agente em relação às abstrações. O novo comportamento pode ser observado na Figura 6.6(a), onde os robôs se mantêm equi-

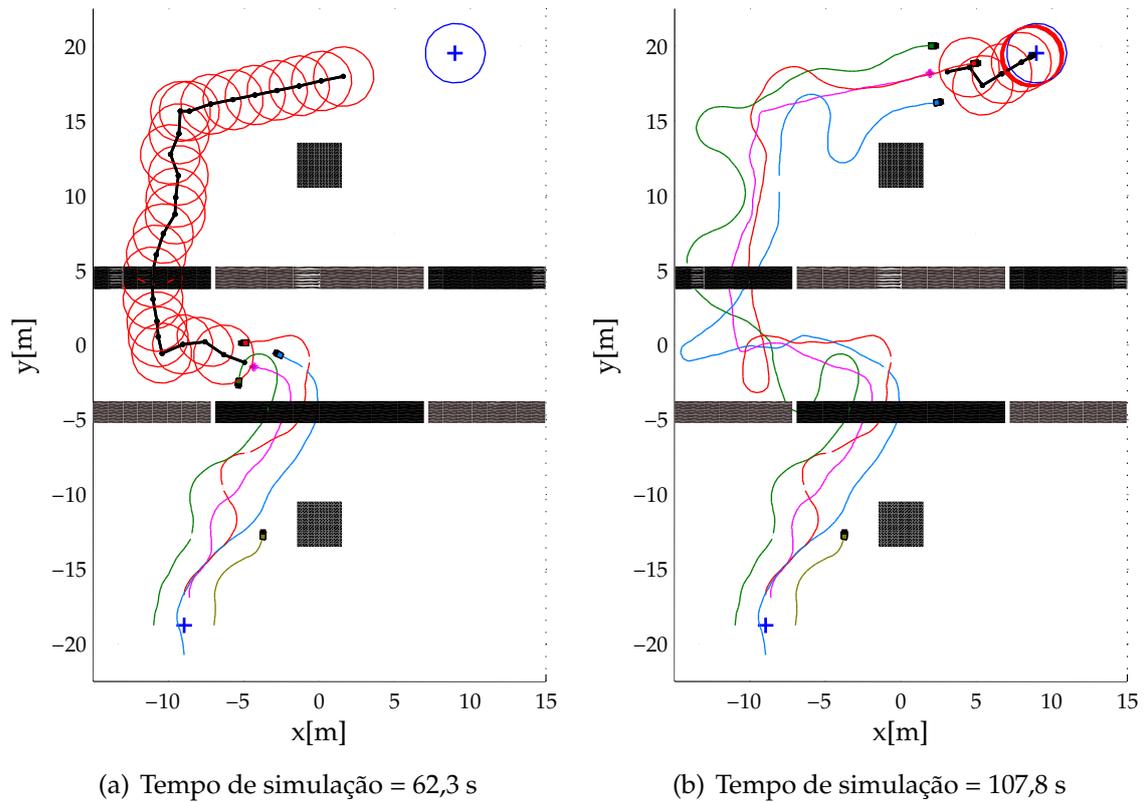


Figura 6.6. Simulação de falha em um dos agentes terrestres: (a) time identifica a falha, se reconfigura e segue a missão, e (b) time atinge o alvo com apenas três UGVs.

distantes uns dos outros após a falha. O time atinge enfim o alvo na Figura 6.6(b).



Figura 6.7. Plataforma de robôs heterogêneos do VeRLab: foram utilizados nos experimentos reais um quadrirotor *Hummingbird/Asctec* e dois robôs terrestres do tipo *Ackerman* construídos no VeRLab sobre a plataforma *Tamiya TXT-1 Monster Truck/Tamiya*.

6.4 Experimentos com Robôs Reais

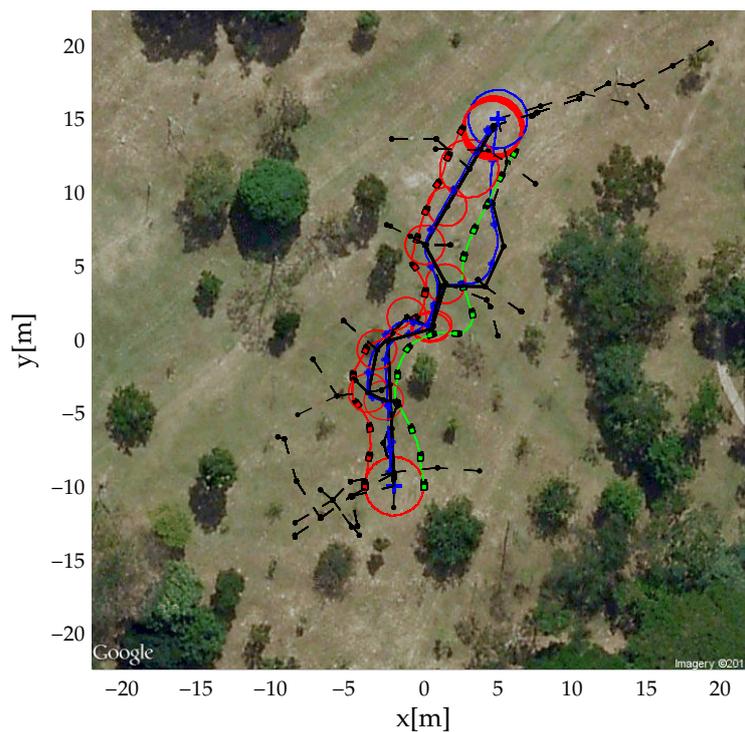
Esta seção apresenta experimentos de cooperação aéreo-terrestre envolvendo robôs reais em ambientes com obstáculos. Foi utilizado aqui um quadrirotor da mesma plataforma descrita na Seção 5.4, o *Hummingbird Autopilot*. Além disso, foram empregados dois robôs terrestres do tipo *Ackerman*, construídos a partir da plataforma *Tamiya TXT-1 Monster Truck* comercializada pela empresa *Tamiya Inc*. O veículo foi modificado para funcionar com um Controlador Eletrônico de Velocidade (*Electronic Speed Control*, ou ESC) com atuação linear, fornecido pela mesma empresa. A Figura 6.7 mostra o time de robôs durante a execução de um experimento.

Os veículos terrestres foram desenvolvidos no VeRLab e são equipados com uma série de sensores que permitem a navegação autônoma em ambientes externos. Uma IMU composta de três giroscópios, três magnetômetros e três acelerômetros, todos dispostos de forma ortogonal, fornecem medições de velocidade angular, orientação magnética e aceleração translacional respectivamente a taxas de aproximadamente 75 Hz. Os dispositivos compõem a *Robovero*, placa dedicada ao controle de robôs vendida pela empresa *Gumstix*. Um receptor GPS fornece informações de

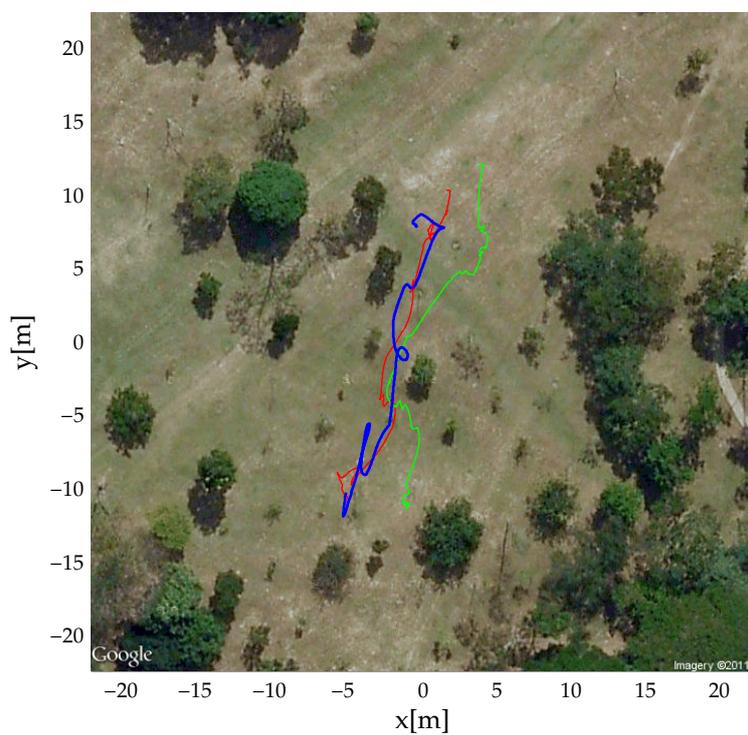
posição e velocidade a taxas de 5 Hz e precisão na localização de cerca de 5 metros. Cada veículo possui ainda um canal sem fio, do tipo *XBee*, que permite estabelecer a rede de comunicação entre os times de UAVs e UGVs. Por fim, a autonomia do sistema foi programada em um *Netbook* da *ASUS*, com sistema operacional *Ubuntu 11.10*, e os módulos de controle e comunicação foram implementados em linguagem *Python*.

As Figuras 6.8 e 6.9 apresentam a expansão das árvores de planejamento e as trajetórias dos agentes do time em uma realização do experimento real. O ambiente de testes foi o mesmo apresentado na Seção 5.4. Na Figura 6.8(a), as elipses azul e vermelha representam o alvo e o ponto de partida do grupo de UGVs respectivamente. Nessa etapa, é possível ver a árvore de planejamento do quadricóptero, executando a tarefa de busca. Já a trajetória do UAV pode ser vista, em azul, na Figura 6.8(b).

A Figura 6.9(a) mostra o planejamento do time na etapa de escolta. As elipses em vermelho representam a propagação da abstração do time de UGVs através do ambiente. As trajetórias finais do time podem ser vistas na Figura 6.9(b). As curvas em vermelho e em verde representam o movimento dos dois robôs terrestres, ao passo que a curva em azul mostra a trajetória do quadricóptero.



(a)



(b)

Figura 6.9. Experimento real com o time de robôs heterogêneos do VeRLab (figuras impressas a partir de imagens do *Google Maps*): (a) expansão da árvore de referências dos robôs e (b) trajetórias finais no estado de escolta.

6.5 Observações Finais

Este capítulo apresentou a aplicação da Árvore Pseudoaleatória de Exploração Rápida Multiagente ao problema de cooperação aéreo-terrestre em ambientes com obstáculos. Foi mostrado que é possível utilizar a metodologia proposta para controlar, de maneira descentralizada, grupos de veículos heterogêneos.

O problema da cooperação entre agentes aéreos e terrestres, em particular, representa um grande desafio, devido à enorme diferença entre as plataformas envolvidas. Os experimentos realizados mostraram que é possível expandir a ideia do uso de abstrações para controlar não apenas grupos, mas também subgrupos de robôs dentro de um mesmo time. É possível ainda modelar o comportamento desses grupos em diferentes etapas da missão por meio de sistemas híbridos. Tais sistemas descrevem o comportamento cinemático das abstrações que controlam o time de formas variadas, dependendo do ponto em que o sistema se encontra na máquina de estados do arcabouço.

Os experimentos realizados consideraram ainda condições de falha no time de robôs. Foi mostrado que é possível manter o mesmo plano pré-calculado antes da falha, apenas modificando a posição dos agentes em relação a esse.

Capítulo 7

Considerações Finais

– Estradas? Para onde vamos não precisamos de estradas.

Doutor Emmet Brown, De volta para o futuro

7.1 Conclusões

Foi proposto neste trabalho um método de planejamento e controle descentralizado para grupos de robôs heterogêneos, sujeitos a incertezas, em ambientes com obstáculos. Nossa técnica, a *Árvore Pseudoaleatória de Exploração Rápida Multiagente* (*Multiple Rapidly-exploring Pseudorandom Trees*, ou MRPT), permite que cada agente do time seja capaz de navegar pelo espaço, evitando colisões com os obstáculos e com outros veículos, e seguindo o plano compartilhado entre o grupo. O método proposto, baseado em uma abordagem de planejamento de caminhos com exploração aleatória, apresenta as vantagens de eficiência, permitindo o uso em tempo real, e de generalização, possibilitando a aplicação a múltiplos robôs heterogêneos.

Ao contrário dos trabalhos mais importantes levantados na literatura, nossa técnica incorpora todas as especificações citadas ao final do Capítulo 2, a saber: planejamento descentralizado e de tempo real, abrangência de diferentes modelos dinâmicos para robôs heterogêneos, navegação em ambientes com obstáculos sujeito a incertezas e perturbações, e baixa taxa de dados na rede de comunicação. Além disso, um ponto importante da nossa abordagem é que o replanejamento de cada agente não invalida e nem leva a conflitos com o plano dos demais, uma vez que cada robô é capaz de prever o comportamento do time por meio do uso de abstrações.

A avaliação da metodologia foi realizada com base em dois casos distintos, derivados do Problema 4.1. O Capítulo 5 explorou o controle de formação (tarefa comum na área da Robótica Cooperativa) e a navegação em ambientes com obstáculos para um time de Micro Veículos Aéreos Não tripulados. Ainda que o grupo apresentasse comportamento homogêneo (todos os robôs eram idênticos), esse problema em particular é bastante desafiador, tanto do ponto de vista teórico quanto experimental. Foi mostrado que é possível utilizar diferentes curvas paramétricas como abstrações, nos casos em que se faz o planejamento com técnicas aleatórias, baseadas em Modelos de Colisão. Tais curvas não precisam apresentar características geométricas bem definidas, premissa assumida nos trabalhos encontrados na literatura até então. Foi mostrado também que a técnica foi bem sucedida frente a perturbações (vento lateral) e que o uso da realimentação dos estados \hat{x} no planejamento promoveu melhoria na coordenação entre os agentes e no tempo de realização da missão.

Já o Capítulo 6 explorou o método frente ao problema de cooperação em grupos de robôs heterogêneos, compostos por veículos aéreos e terrestres. Foi mostrado nesse caso que a modelagem dos sinais de referência R pode ser feita de modo a conformar missões com diferentes etapas, garantindo assim a premissa de que é possível realizar tarefas complexas por meio de pequenos passos mais simples. Além disso, o planejador permite incorporar vários subgrupos dentro de \mathcal{R} , reduzindo a complexidade do controle e permitindo a coordenação de múltiplos times de robôs no mesmo ambiente. Foi mostrado ainda que o método pode se adaptar a possíveis falhas em agentes do time, garantindo ainda a completude da missão.

Por ser baseado em variações do algoritmo RRT, uma especialização de métodos do tipo PRM (embora não restrito a tais métodos), a MRPT herda uma série de desvantagens em relação a outras técnicas. Primeiramente não há garantias de que seja obtido um resultado final satisfatório em tempo finito. O plano gerado é completamente aleatório e imprevisível, apesar do uso de funções pseudoaleatórias na base da abordagem. Como consequência, também não há garantias quanto à otimalidade, ou mesmo qualidade, dos resultados apresentados. As trajetórias realizadas pelos robôs do time dificilmente serão as melhores possíveis, embora se possa utilizar de heurísticas para melhorar o plano nesse sentido. Finalmente é difícil implantar qualquer tipo de análise ou avaliação formal ao método. Estudos sobre a complexidade e custo computacional de algoritmos RRTs raramente são encontrados na literatura. Além de se tratarem de procedimentos aleatórios, sua generalização com relação aos modelos dinâmicos e às incertezas embutidas nos sistemas torna tais análises complexas. Assim, do ponto de vista formal, é necessária uma

série de suposições para provar o funcionamento dos algoritmos em certas circunstâncias.

Etapas iniciais do trabalho, seguindo a mesma linha de planejamento aleatório baseado em RRTs para um ou múltiplos robôs, foram publicados em revistas e conferências internacionais, conforme listado a seguir:

1. Armando Alves Neto, Douglas G. Macharet, Mario F. M. Campos. Feasible Path Planning for Fixed-wing UAVs Using Seventh Order Bézier Curves. *Journal of the Brazilian Computer Society*, 1–11, 2012.
2. Armando Alves Neto, Douglas G. Macharet, Mario F. M. Campos. On the Generation of Trajectories for Multiple UAVs in Environments with Obstacles. *Journal of Intelligent and Robotic Systems*, 57(4):123–141, 2010.
3. Armando Alves Neto, Douglas G. Macharet, Mario F. M. Campos. Feasible RRT-based Path Planning Using Seventh Order Bézier Curves. *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010.
4. Armando Alves Neto, Douglas G. Macharet, Mario F. M. Campos. On the Generation of Trajectories for Multiple UAVs in Environments with Obstacles – *Selected papers from the 2nd International Symposium on UAVs*, Reno, U.S.A. June 8-10, 2009. K.P.; Beard, R.; Oh, P.; Ollero, A.; Piegl, L.; Shim, H. Valavanis (ed.), Springer, New York, USA, 2010.
5. Armando Alves Neto, Douglas G. Macharet, Mario F. M. Campos. On the Generation of Feasible Paths for Aerial Robots in Environments with Obstacles. *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, USA, 2009.

7.2 Trabalhos Futuros

A abordagem proposta neste texto abre possibilidades para uma série de melhorias e extensões futuras. Uma possível solução para minimizar o custo computacional da técnica, quando aplicada a grupos com grande número de veículos, pode ser a utilização de uma hierarquia de prioridade entre os robôs, de forma que cada agente precise computar apenas o plano de seus respectivos vizinhos. Tal hierarquia reduziria a dependência da estrutura de rede, aumentando a robustez da cooperação.

Ainda sobre trabalhos futuros, supõe-se que a capacidade de generalização do método permita a aplicação a missões mais abrangentes, além das discutidas nos capítulos anteriores. Acredita-se que o cálculo de \mathbf{R} possa ser facilmente generalizado para tarefas em times sem restrições de formação, mas cuja navegação seja fundamental para o processo de coordenação.

É possível ainda avaliar a técnica proposta frente a outros pontos não abordados neste texto, como a navegação em ambientes desconhecidos, onde os robôs precisam construir de maneira cooperativa o mapa no qual farão o planejamento. Outro foco importante de exploração é a estrutura de rede e a interferência de possíveis falhas de comunicação na coordenação do grupo.

Referências Bibliográficas

- Achtelik, M.; Achtelik, M.; Weiss, S. & Siegwart, R. (2011). Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. Em *IEEE International Conference on Robotics and Automation*, pp. 3056–3063.
- Alves Neto, A.; Guimarães Macharet, D. & Campos, M. F. M. (2009). On the Generation of Trajectories for Multiple UAVs in Environments with Obstacles. Em *2nd International Symposium on Unmanned Aerial Vehicles (UAV)*, Nevada, USA.
- Alves Neto, A.; Guimarães Macharet, D. & Campos, M. F. M. (2010). On the Generation of Trajectories for Multiple UAVs in Environments with Obstacles. *Journal of Intelligent and Robotic Systems*, 57(4):123–141.
- Aoude, G.; Luders, B.; Levine, D. & How, J. (2010). Threat-aware path planning in uncertain urban environments. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6058–6063. ISSN 2153-0858.
- Arrowsmith, D. & Place, C. (1992). *Dynamical Systems: Differential Equations, Maps, and Chaotic Behaviour*. Chapman and Hall Mathematics Series. Taylor & Francis. ISBN 9780412390807.
- AscTec Hummingbird AutoPilot (2011). <http://www.asctec.de/asctec-hummingbird-autopilot-5/>.
- Balch, T. & Arkin, R. C. (1997). Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 14:926–939.
- Belta, C. & Kumar, V. (2004). Abstraction and control for groups of robots. *IEEE Transactions on Robotics*, 20(5):865–875. ISSN 1552-3098.
- Bruce, J. & Veloso, M. (2002). Real-time randomized path planning for robot navigation. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pp. 2383–2388.

- Burns, B. & Brock, O. (2007). Single-query motion planning with utility-guided random trees. Em *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3307–3312. ISSN 1050-4729.
- Chaimowicz, L.; Cowley, A.; Gomez-Ibanez, D.; Grocholsky, B.; Hsieh, M.; Hsu, H.; Keller, J.; Kumar, V.; Swaminathan, R. & Taylor, C. (2005). Deploying air-ground multi-robot teams in urban environments. Em Parker, L.; Schneider, F. & Schultz, A., editores, *Multi-Robot Systems. From Swarms to Intelligent Automata*, volume III, pp. 223–234. Springer Netherlands.
- Chaimowicz, L.; Grocholsky, B.; Keller, J. F.; Kumar, V. & Taylor, C. J. (2004). Experiments in multirobot air-ground coordination. Em *International Conference on Robotics and Automation*, pp. 4053--4058.
- Chaimowicz, L. & Kumar, V. (2004). Aerial Shepherds: Coordination among UAVs and Swarms of Robots. Em *7th International Symposium on Distributed Autonomous Robotic Systems*, pp. 231--240.
- Chen, Y. Q. & Wang, Z. (2005). Formation control: a review and a new consideration. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3181–3186.
- Choi, J. (2007). Workspace-guided rapidly-exploring random tree method for a robot arm. Final report for a course of University Illinois.
- Choset, H.; Lynch, K.; Hutchinson, S.; Kantor, G.; Burgard, W.; Kavraki, L. & Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms and Implementation*. MIT Press.
- Consolini, L.; Morbidi, F.; Prattichizzo, D. & Tosques, M. (2007). A geometric characterization of leader-follower formation control. Em *IEEE International Conference on Robotics and Automation*, pp. 2397–2402. ISSN 1050-4729.
- Cruz, D.; McClintock, J.; Perteet, B.; Orqueda, O.; Cao, Y. & Fierro, R. (2007). Decentralized cooperative control - a multivehicle platform for research in networked embedded systems. *IEEE Control Systems*, 27(3):58--78. ISSN 1066-033X.
- Desai, J.; Ostrowski, J. & Kumar, V. (2001). Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908. ISSN 1042-296X.

- Desaraju, V. R. & How, J. P. (2011). Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees. Em *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 385--403.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269--271. ISSN 0029-599X.
- Dunbar, W. B. (2002). Model predictive control: Extension to coordinated multi-vehicle formations and real-time implementation. CDS Technical Memo CIT-CDS 01-01.
- Dunbar, W. B. & Murray, R. M. (2002). Model predictive control of coordinated multi-vehicle formations. Em *IEEE Conference on Decision and Control*, pp. 4631--4636.
- Elfes, A.; Bergerman, M.; Carvalho, J.; Paiva, E.; Ramos, J. & Bueno, S. (1999). Air-ground robotic ensembles for cooperative applications: concepts and preliminary results. Em *International Conference on Field and Service Robotics*, pp. 75--80, Pittsburgh, PA, USA.
- Flint, M.; Polycarpou, M. & Fernandez-Gaucherand, E. (2002). Cooperative path-planning for autonomous vehicles using dynamic programming. Em *15th IFAC World Congress*, pp. 1694--1700.
- Frazzoli, E. (2003). Quasi-random algorithms for real-time spacecraft motion planning and coordination. *Acta Astronautica*, 53(4-10):485--495. ISSN 0094-5765.
- Frazzoli, E.; Dahleh, M. A. & Feron, E. (2002). Real-time motion planning for agile autonomous vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 25:116--129.
- Grocholsky, B.; Bayraktar, S.; Kumar, V. & Pappas, G. (2004). UAV and UGV Collaboration for Active Ground Feature Search and Localization. Em *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, pp. 223--234.
- Guerrero, J.; Fantoni, I.; Salazar, S. & Lozano, R. (2010). Flight formation of multiple mini rotorcraft via coordination control. Em *IEEE International Conference on Robotics and Automation*, pp. 634--639.
- Gurdan, D.; Stumpf, J.; Achtelik, M.; Doth, K.-M.; Hirzinger, G. & Rus, D. (2007). Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz. Em *IEEE International Conference on Robotics and Automation*, pp. 361--366.

- Hart, P.; Nilsson, N. & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107. ISSN 0536-1567.
- Ippolito, C.; Joo, S.; Al-Ali, K. & Yeh, Y. H. (2008). Polymorphic Control Reconfiguration in an Autonomous UAV with UGV Collaboration. Em *IEEE Aerospace Conference*, pp. 1–14. ISSN 1095-323X.
- Kamio, S. & Iba, H. (2005). Random sampling algorithm for multi-agent cooperation planning. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1265–1270.
- Kamio, S. & Iba, H. (2006). Cooperative Object Transport with Humanoid Robots using RRT Path Planning and Re-Planning. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2608–2613.
- Keviczky, T.; Borrelli, F. & Balas, G. J. (2004). A study on decentralized receding horizon control for decoupled systems. Em *American Control Conference (ACC)*, pp. 4921–4926, Boston, MA.
- Keviczky, T.; Borrelli, F. & Balas, G. J. (2007). *Distributed Predictive Control: Synthesis, Stability and Feasibility*, pp. 79–108. John Wiley & Sons, Ltd.
- Kuwata, Y.; Teo, J.; Fiore, G.; Karaman, S.; Frazzoli, E. & How, J. P. (2009). Realtime motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems*, 17:1105–1118.
- Kuwata, Y.; Teo, J.; Karaman, S.; Fiore, G.; Frazzoli, E. & How, J. P. (2008). Motion planning in complex environments using closed-loop prediction. Em *AIAA Guidance, Navigation, and Control Conf. and Exhibit*.
- Lama, M. A. O. (2008). Sampling-based motion planning: a survey. *Computación y Sistemas*, 12(1):5–24.
- Latombe, J.-C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA. ISBN 079239206X.
- LaValle, S. & Kuffner, J.J., J. (1999). Randomized kinodynamic planning. Em *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pp. 473–479.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, Urbana, Illinois, U.S.A. <http://planning.cs.uiuc.edu/>.

- LaValle, S. M.; Branicky, M. S. & Lindemann, S. R. (2004). On the relationship between classical grid search and probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7-8):673–692.
- LaValle, S. M. & Kuffner, J. J. (2001). Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400.
- Likhachev, M. & Stentz, A. (2008). R* search. Em *23rd national conference on Artificial intelligence*, pp. 344–350. AAAI Press.
- Luders, B. D.; Karaman, S.; Frazzoli, E. & How, J. P. (2010). Bounds on tracking error using closed-loop rapidly-exploring random trees. Em *American Control Conference (ACC)*, volume 1, pp. 5406–5412.
- Macharet, D. G.; Alves Neto, A. & Campos, M. F. M. (2009). On the generation of feasible paths for aerial robots in environments with obstacles. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, USA.
- Marcolino, L. S. & Chaimowicz, L. (2009a). Traffic control for a swarm of robots: Avoiding group conflicts. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, USA.
- Marcolino, L. S. & Chaimowicz, L. (2009b). Traffic control for a swarm of robots: Avoiding target congestion. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, USA.
- Michael, N.; Belta, C. & Kumar, V. (2006). Controlling three dimensional swarms of robots. Em *IEEE International Conference on Robotics and Automation*, pp. 964–969. ISSN 1050-4729.
- Michael, N.; Fink, J. & Kumar, V. (2007). Controlling a team of ground robots via an aerial robot. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 965–970.
- Michael, N. & Kumar, V. (2009). Planning and control of ensembles of robots with non-holonomic constraints. *The International Journal of Robotics Research*, 28(8):962–975.
- Michael, N.; Mellinger, D.; Lindsey, Q. & Kumar, V. (2010). The GRASP Multiple Micro-UAV Testbed. *IEEE Robotics Automation Magazine*, 17(3):56–65. ISSN 1070-9932.

- Olfati-saber, R.; Fax, J. A. & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. Em *Proceedings of the IEEE*, volume 95, pp. 215 – 233.
- Parker, L. E. (2008). Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems. *Journal of Physical Agents*, 2(1):5–14.
- Pereira, G.; Das, A.; Kumar, V. & Campos, M. (2003). Formation control with configuration space constraints. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pp. 2755–2760.
- Pereira, G. A. S.; Kumar, V. & Campos, M. F. M. (2004). Decentralized algorithms for multi-robot manipulation via caging. *International Journal of Robotics Research*, 23(7).
- Phan, C. & Liu, H. (2008). A cooperative UAV/UGV platform for wildfire detection and fighting. Em *7th International Conference on System Simulation and Scientific Computing*, pp. 494–498.
- ROS Asctec Autopilot Driver (2011). http://www.ros.org/wiki/asctec_autopilot.
- Santos, V. G. & Chaimowicz, L. (2011). Hierarchical congestion control for robotic swarms. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4372--4377.
- Schouwenaars, T.; DeMoor, B.; Feron, E. & How, J. (2001). Mixed integer programming for multi-vehicle path planning. Em *European Control Conference*, pp. 2603–2608.
- Shiroma, P. M. & Campos, M. F. M. (2009). Comutar: A framework for multi-robot coordination and task allocation. Em *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4817--4824, St. Louis, USA.
- Shkolnik, A.; Walter, M. & Tedrake, R. (2009). Reachability-guided sampling for planning under differential constraints. Em *IEEE International Conference on Robotics and Automation (ICRA)*., pp. 2859–2865. ISSN 1050-4729.
- Siegwart, R. & Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*. Bradford Company, Scituate, MA, USA. ISBN 026219502X.
- Stentz, A. & Mellon, I. C. (1993). Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics and Automation*, 10:89–100.

- Tan, K.-H. (1997). Virtual structures for high-precision cooperative mobile robot control. *Autonomous Robots*, 4:387–403.
- Tanner, H. G. (2007). Switched uav-ugv cooperation scheme for target detection. Em *International Conference on Robotics and Automation*, pp. 3457--3462.
- Tanner, H. G. & Christodoulakis, D. K. (2007). Decentralized cooperative control of heterogeneous vehicle groups. *Robot. Auton. Syst.*, 55:811–823. ISSN 0921-8890.
- Vahrenkamp, N.; Kuhn, E.; Asfour, T. & Dillmann, R. (2010). Planning Multi-Robot Grasping Motions. Em *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pp. 593--600, Nashville, USA.
- Vaughan, R. T.; Sukhatme, G. S.; Mesa-martinez, F. J. & Montgomery, J. F. (2000). Fly spy: Lightweight localization and target tracking for cooperating air and ground robots. Em *Int. Symp. Distributed Autonomous Robot Systems*, pp. 315--324.
- Weinstein, A. J. & Moore, K. L. (2010). Pose estimation of ackerman steering vehicles for outdoors autonomous navigation. Em *IEEE International Conference on Industrial Automation*, pp. 579--584, Valparaiso, Chile.