



Universidade Federal da Bahia
Escola Politécnica
Programa de Pós-Graduação em Engenharia
Elétrica



TIAGO TRINDADE RIBEIRO

CONTROLADORES NMPC DESCENTRALIZADOS
PARA O CONTROLE DE FORMAÇÃO DE ROBÔS
MÓVEIS

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

Orientador: Prof. Dr. André Gustavo Scolari Conceição
Coorientador: Prof. Dr. Antonio Marcus Nogueira Lima

Salvador
Maio - 2017



Universidade Federal da Bahia
Escola Politécnica
Curso de Graduação em Engenharia Elétrica



CONTROLADORES NMPC DESCENTRALIZADOS PARA O CONTROLE DE FORMAÇÃO DE ROBÔS MÓVEIS

Autor: Tiago Trindade Ribeiro
Orientador: Prof. Dr. André Gustavo Scolari Conceição - UFBA
Coorientador: Prof. Dr. Antonio Marcus Nogueira Lima

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Bahia como parte dos requisitos necessários para obtenção do grau de Doutor em Engenharia Elétrica.

Área de Concentração: Processamento de Informação e Energia
Linha de Pesquisa: Controle e Automação

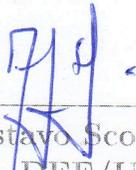
Salvador
2017

Controladores NMPC Descentralizados para o Controle de Formação de Robôs Móveis

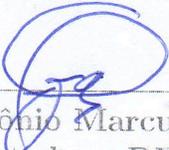
TESE DE DOUTORADO

Autor: Tiago Trindade Ribeiro
Orientador: Prof. Dr. André Gustavo Scolari Conceição - UFBA
Coorientador: Prof. Dr. Antônio Marcus Nogueira Lima

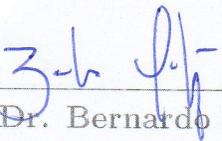
Tese de Doutorado aprovada em 12 de maio de 2017 pela banca examinadora composta pelos seguintes membros:



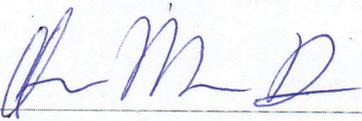
Prof. Dr. André Gustavo Scolari Conceição
Orientador - DEE/UFBA



Prof. Dr. Antônio Marcus Nogueira Lima
Coorientador - DEE/UFBA



Prof. Dr. Bernardo Ordoñez
Examinador interno - DEE/UFBA



Prof. Dr. Anfranserai Morais Dias
Examinador externo - DETEC/UEFS



Prof. Dr. Carlos Eduardo Trabuco Dórea
Examinador externo - DCA/UFRN

R484 Ribeiro, Tiago Trindade
Controladores NMPC Descentralizados para o Controle de
Formação de Robôs Móveis / Tiago Trindade Ribeiro. – Salvador,
2017.

219f. : il. color.

Orientador: Prof. Doutor André Gustavo Scolari Conceição
Coorientador: Prof. Doutor Antonio Marcus Nogueira Lima
Tese (Doutorado - Programa de Pós-Graduação em Engenharia
Elétrica) – Universidade Federal da Bahia, Escola
Politécnica, 2017.

1. Robótica. 2. Controle de formação. 3. Controle preditivo.
4. Sistemas multi-robôs. I. Conceição, André Gustavo Scolari.
II. Nogueira Lima, Antonio Marcus. III. Título.

CDD: 629.892

Controladores NMPC Descentralizados para o Controle de Formação de Robôs Móveis

TESE DE DOUTORADO

Autor: Tiago Trindade Ribeiro
Orientador: Prof. Dr. André Gustavo Scolari Conceição - UFBA
Coorientador: Prof. Dr. Antônio Marcus Nogueira Lima

Tese de Doutorado aprovada em 12 de maio de 2017 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. André Gustavo Scolari Conceição
Orientador - DEE/UFBA

Prof. Dr. Antônio Marcus Nogueira Lima
Coorientador - DEE/UFCG

Prof. Dr. Bernardo Ordonez
Examinador interno - DEE/UFBA

Prof. Dr. Anfranserai Moraes Dias
Examinador externo - DETEC/UEFS

Prof. Dr. Carlos Eduardo Trabuco Dórea
Examinador externo - DCA/UFRN

À minha filha Maria Fernanda Wanderley Ribeiro

Agradecimentos

A Deus, em primeiro lugar, por sempre me dar força e coragem para conquistar meus objetivos, pela manutenção da fé inabalável que sempre me sustenta na superação dos obstáculos.

À minha esposa, Fernanda Wanderley Ribeiro, pelo amor, paciência e companheirismo.

À minha filha, Maria Fernanda, por ser luz do meu caminho e fonte de contínua inspiração. O maior presente que Deus poderia me dar.

Aos meus pais pelo suporte incondicional na realização de todos os meus projetos.

Ao meu orientador Professor André Scolari pelos sólidos conhecimentos francamente repassados de maneira irrestrita para o desenvolvimento deste trabalho, além dos conselhos fundamentais para a minha carreira acadêmica.

Ao meu coorientador Antônio Marcus pelos conhecimentos que nortearam a expansão dos horizontes no âmbito desta tese tanto no desenvolvimentismo quanto na apresentação dos resultados.

Ao professor Tito Santos pelas colaborações precisas e pertinentes em quase todas as etapas do desenvolvimento dos controladores preditivos deste trabalho.

Por fim agradeço à FAPESB pelo suporte financeiro, aos funcionários da UFBA, em especial do PPGEE, e todos os colegas, professores e alunos que de alguma forma se fizeram presentes nesta caminhada.

*“Com a sabedoria edifica-se a casa,
e com a inteligência ela se firma”
Provérbios 24:3*

Resumo

Nesta tese propõe-se o projeto e o desenvolvimento de controladores de formação descentralizados para robôs móveis de diferentes tipos. As soluções baseiam-se na utilização de controladores preditivos não lineares *NMPC* de tempo contínuo para o cálculo das referências de pose e velocidade para cada robô da formação acoplados via função objetivo ou estados. A tarefa definida para os robôs em formação é o seguimento de caminhos espaciais, pré definidos ou adquiridos através de informações visuais, caso em que o controle é realizado diretamente no plano da imagem, através da utilização de características visuais simplificadas. Para o tratamento de formações e caminhos variantes no tempo, propõe-se uma abordagem para garantia de factibilidade recursiva sem demasiado aumento da complexidade algorítmica dos controladores locais. Tal abordagem baseia-se no relaxamento das restrições do problema de otimização sem penalização explícita na função objetivo. São apresentadas duas propostas para o controle de formação utilizando informações visuais, uma baseia-se num novo conceito de câmeras e caminhos virtuais e a outra implementa a estratégia líder seguidor tradicional. Decorrente da não linearidade dos modelos obtidos, surgem limitações nos horizontes de predição e controle, interferindo diretamente nos requisitos de estabilidade. Neste caso, a estabilidade é garantida através da adição de um custo terminal às funções objetivo e uma zona terminal para as restrições. Resultados simulados e experimentais são apresentados para demonstrar o desempenho das estratégias propostas em diversas situações.

Palavras Chave: Robótica móvel, Controle de formação, Seguimento de caminho, Controle preditivo não linear, Visão computacional.

Abstract

This thesis proposes the design and the development of decentralized formation controllers for different types of mobile robots. The solutions are based on the use of continuous time non-linear model predictive formation controllers *NMPC* for the calculus of references of poses and velocity. The task defined for the robots in formation are to track spacial paths, pre-specified or acquired through visual information, in wich case the control is accomplished direct from the image plane. For the handling of time-varying formations and paths, an approach is proposed to guarantee recursive feasibility without too much increase of the algorithmic complexity of the local controllers. Such an approach is based on the constraints relaxation of the optimization problem without any explicit penalty in the objective function. Two approaches are presented for Formation control using visual information, one based on a new concept of virtual cameras and paths and the other one implements the traditional Leader-Follower strategy. Due to the non-linearity of the obtained models, limitations on prediction and control horizons arise, directly interfering on the stability requirements. Simulated and experimental results are shown to demonstrate the good performance of the proposed strategies in different situations.

Keywords: Mobile robots, Formation control, Path following, Non-linear model predictive control, Computer vision.

Lista de Publicações

- Ribeiro, T. T. ; Fernandes, R. O. ; Conceição, A. G. S. *Seguimento de Caminhos Visuais Baseado em NMPC para Robôs Diferenciais*. In: XXI Congresso Brasileiro de Automática - CBA2016, 2016, Vitória-ES. XXI Congresso Brasileiro de Automática - CBA2016, 2016.
- Ribeiro, T. T. ; Fernandes, R. O. ; Conceição, A. G. S. *Garantia de Estabilidade e Factibilidade de Controladores NMPC para o Seguimento de Caminhos Visuais*. In: XXI Congresso Brasileiro de Automática - CBA2016, 2016, Vitória-ES. XXI Congresso Brasileiro de Automática - CBA2016, 2016.
- Ribeiro, T. T. ; Augusto L. da Costa ; Conceição, A. G. S. *Distributed Constrained NMPC with Infeasibility Handling Applied to Formation Control of Nonholonomic Vehicles*. Journal of Control, Automation and Electrical Systems December 2015, Volume 26, Issue 6, pp 599-608.
- Ribeiro, T. T. ; Conceição, A. G. S. ; Inkyu Sa ; Peter Corke. *Nonlinear Model Predictive Formation Control for Quadcopters*. In: 11th IFAC Symposium on Robot Control, 2015, Salvador. 11th IFAC Symposium on Robot Control SYROCO 2015. Volume 48, Issue 19, 2015, Pages 39-44.
- Ribeiro, T. T. ; Augusto L. da Costa ; Conceição, A. G. S. *Controle de Formação baseado em NMPC para Quadricopteros*. In: XX Congresso Brasileiro de Automática, 2014, Belo Horizonte. anais do XX Congresso Brasileiro de Automática - CBA 2014, 2014. p. 1490-1497.
- Ribeiro, T. T. ; Ferrari, R. ; Santos, J. ; Conceição, A. G. S. *Formation control of mobile robots using decentralized nonlinear model predictive control* Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on, Wollongong, NSW, 2013, pp. 32-37.
doi: 10.1109/AIM.2013.6584064.
- Ribeiro, T. T. ; Fróes, S. ; Conceição, A. G. S. *Controle de Formação Baseado em NMPC Aplicado a Robótica Móvel*. In: XI Simpósio Brasileiro de Automação Inteligente, 2013, Fortaleza-CE. Anais do XI Simpósio Brasileiro de Automação Inteligente, 2013.

- Ribeiro, T. T. ; Santos, J. ; Augusto L. da Costa ; Conceição, A. G. S. *Wireless Real-Time System for Omnidirectional Mobile Robots Control*. In: XI Simpósio Brasileiro de Automação Inteligente, 2013, Fortaleza-CE. Anais do XI Simpósio Brasileiro de Automação Inteligente, 2013.

Lista de Figuras

1.1	Problema de infactibilidade inicial do controle de formação baseado em <i>NMPC</i> - Resultado experimental.	5
1.2	Síntese das soluções propostas para os problemas com as abordagens clássicas para controle de formação.	6
2.1	Mecanismo envolvido no andar dos seres humanos. Adaptado de (Siegwart e Nourbakhshi 2004, p. 15)	12
2.2	Relação entre potência específica e velocidade alcançável para diversos mecanismos de locomoção. Adaptado de (Siegwart e Nourbakhshi 2004, p. 15)	12
2.3	Relação entre dois sistemas de coordenadas rotacionados.	14
2.4	Componentes de um robô móvel omnidirecional.	17
2.5	Representação dos sistemas de coordenadas, velocidades e forças.	18
2.6	Robô móvel omnidirecional.	19
2.7	Exemplo de movimentação de robôs diferenciais.	20
2.8	Modelagem cinemática do robô diferencial.	21
2.9	Robôs diferenciais utilizados.	22
2.10	Sistemas de coordenadas para a modelagem do quadricóptero.	23
2.11	Alguns quadricópteros comerciais.	24
2.12	Algumas telas do software <i>Decision</i>	26
2.13	Fluxograma do software <i>Decision</i>	26
2.14	Arquitetura resumida da pilha de protocolos <i>zigbee</i> e um módulo <i>xbee</i> da <i>maxstream</i>	29
2.15	Módulos <i>Flyport</i> utilizados para a comunicação entre robô omnidirecional e <i>PC</i>	30
2.16	Arquitetura de comunicação sistema multi robôs omnidirecionais.	31
2.17	Arquitetura de comunicação sistema multi robôs diferenciais.	32
2.18	Instalação das <i>webcams</i> nos Robôs diferenciais.	33
3.1	Princípio de funcionamento dos controladores preditivos baseados em modelo.	42
3.2	Diferença entre as predições em malha aberta e o comportamento em malha fechada.	50

4.1	Sistemas de coordenadas do robô móvel omnidirecional.	56
4.2	Ilustração do problema de seguimento de caminho.	57
4.3	Estrutura de controle para o robô omnidirecional ¹	61
4.4	Seguimento de caminho circular - Robô omnidirecional - Caso ideal. .	63
4.5	Seguimento de caminho circular com erro de postura inicial elevado. .	64
4.6	Sistemas de coordenadas e representação do problema de seguimento de caminho.	66
4.7	Estrutura de controle para o quadricóptero.	68
4.8	Seguimento de caminhos para quadricópteros: Simulação 1 - Caminho circular fixo.	70
4.9	Seguimento de caminhos para quadricópteros: Simulação 2 - Cami- nhos circulares variantes no tempo.	72
4.10	Formação geométrica das imagens.	74
4.11	Representação da Projeção Perspectiva.	74
4.12	Sistemas de coordenadas utilizadas para o <i>IBVS</i>	75
4.13	Seguimento de caminhos visuais.	77
4.14	Imagem do caminho a ser seguido.	78
4.15	Estrutura de controle baseada em visão para o robô diferencial. . . .	81
4.16	Seguimento de caminhos visuais - Sem perturbações.	83
4.17	Seguimento de caminhos visuais - Com perturbações.	85
4.18	Seguimento de caminhos visuais - Com perturbações.	86
4.19	Seguimento de caminhos visuais - Linha reta.	87
4.20	Seguimento de caminhos visuais - Erros iniciais nos estados.	88
4.21	Primeiro caminho arbitrário.	89
4.22	Seguimento de caminhos visuais - Primeiro Caminho arbitrário - <i>NMPC</i> Nominal.	90
4.23	Seguimento de caminhos visuais - Caminho arbitrário - $\omega = KZ$	91
4.24	Seguimento de caminhos visuais - Primeiro Caminho arbitrário - Folga na restrição da entrada.	92
4.25	Seguimento de caminhos visuais - Caminho arbitrário - Garantia de estabilidade. .	94
4.26	Seguimento de caminhos visuais - Caminho arbitrário - Garantia de factibilidade.	95
4.27	Segundo caminho arbitrário	97
4.28	Seguimento de caminhos visuais - Segundo caminho arbitrário - <i>NMPC</i> nominal.	98
4.29	Seguimento de caminhos visuais - Segundo caminho arbitrário - <i>NMPC</i> com garantia de estabilidade.	99
4.30	Seguimento de caminhos visuais - Segundo caminho arbitrário - <i>NMPC</i> com garantia de factibilidade.	100
4.31	Seguimento de caminhos visuais - Análise da curvatura visual medida. .	101
4.32	Comparação das abordagens para $v = 0, 1s$	102
4.33	Comparação das abordagens para $v = 0, 2s$	104

4.34	Caminho horizontal no plano da imagem - <i>NMPC</i> nominal - $v = 0, 1\text{m/s}$	112
4.35	Caminho horizontal no plano da imagem - <i>NMPC</i> com garantia de estabilidade - $v = 0, 1\text{m/s}$	112
4.36	Caminho horizontal no plano da imagem - <i>NMPC</i> com garantia de factibilidade - $v = 0, 1\text{m/s}$	112
4.37	Caminho vertical no plano da imagem - <i>NMPC</i> nominal - $v = 0, 1\text{m/s}$.	113
4.38	Caminho vertical no plano da imagem - <i>NMPC</i> com garantia de estabilidade - $v = 0, 1\text{m/s}$	113
4.39	Caminho vertical no plano da imagem - <i>NMPC</i> com garantia de factibilidade - $v = 0, 1\text{m/s}$	113
4.40	Caminho vertical no plano da imagem - <i>NMPC</i> nominal - $v = 0, 1\text{m/s}$.	114
4.41	Caminho vertical no plano da imagem - <i>NMPC</i> com garantia de estabilidade - $v = 0, 1\text{m/s}$	114
4.42	Caminho vertical no plano da imagem - <i>NMPC</i> com garantia de factibilidade - $v = 0, 1\text{m/s}$	114
4.43	Caminho vertical no plano da imagem - <i>NMPC</i> nominal - $v = 0, 2\text{m/s}$.	115
4.44	Caminho vertical no plano da imagem - <i>NMPC</i> com garantia de estabilidade - $v = 0, 2\text{m/s}$	115
4.45	Caminho vertical no plano da imagem - <i>NMPC</i> com garantia de factibilidade - $v = 0, 2\text{m/s}$	115
5.1	Representação de um conjunto de robôs sob a forma de grafo - Representação genérica.	120
5.2	Representação de um conjunto de robôs sob a forma de grafo - Estratégia Líder-Seguidor.	121
5.3	Seguimento de caminho coordenado: formação em triângulo.	122
5.4	Estrutura de controle para cada robô omnidirecional.	126
5.5	Robô omnidirecional utilizado.	126
5.6	Controle de formação triangular com caminhos circulares - Robôs omnidirecionais - Caso ideal.	128
5.7	Controle de formação triangular com caminhos circulares - Robôs omnidirecionais - Caminhos e formações variantes no tempo.	130
5.8	Controle de formação triangular com caminhos circulares - Robôs omnidirecionais - Caminhos e formações variantes no tempo (<i>continuação</i>).	131
5.9	Estrutura de controle para cada robô diferencial.	133
5.10	Robô diferencial utilizado.	134
5.11	Controle de formação para robôs diferenciais - Sem relaxamento das restrições.	136
5.12	Controle de formação para robôs diferenciais - Com relaxamento das restrições.	138
5.13	Controle de formação para robôs diferenciais - Entradas ótimas.	139
5.14	Representação do seguimento de caminho coordenado.	140

5.15	Estrutura de controle para cada quadricóptero.	142
5.16	Controle de formação para quadricópteros: Caminhos circulares - Formação triangular variante no tempo.	145
5.17	Controle de formação utilizando informações visuais - Estratégia baseada em câmeras virtuais.	148
5.18	Estrutura de controle para cada robô seguidor - Estratégia baseada em câmeras virtuais.	151
5.19	Robô diferencial utilizado.	151
5.20	Caminho de referência para o robô líder.	152
5.21	Controle de formação utilizando informações visuais - Utilização de câmeras virtuais (continuação).	153
5.22	Estratégia Líder-Seguidor.	154
5.23	Estrutura de controle para cada robô seguidor - Estratégia Líder-Seguidor.	157
5.24	Controle de formação utilizando informações visuais - Estratégia Líder-Seguidor - Caso nominal.	160
5.25	Comparação das curvaturas estimadas pelo sistema visual.	161
5.26	Controle de formação fixas utilizando informações visuais - Estratégia Líder-Seguidor - Caso nominal para os seguidores.	163
5.27	Controle de formação fixas utilizando informações visuais - Estratégia Líder-Seguidor - Garantia de estabilidade para os seguidores.	164
5.28	Imagem capturada ao final do caminho utilizado.	165
5.29	Controle de formação fixas utilizando informações visuais - Estratégia Líder-Seguidor - Garantia de factibilidade para os seguidores.	166
5.30	Comportamento do <i>NMPC</i> nominal na continuação do experimento ao fim do caminho.	167
5.31	Comportamento do <i>NMPC</i> com garantia de estabilidade na continuação do experimento ao fim do caminho.	168
5.32	Comportamento do <i>NMPC</i> com garantia de factibilidade na continuação do experimento ao fim do caminho.	169
5.33	Controle de formações variantes utilizando informações visuais - Estratégia Líder-Seguidor - Caso Nominal.	171
5.34	Controle de formações variantes utilizando informações visuais - Estratégia Líder-Seguidor - Garantia de estabilidade.	172
5.35	Controle de formações variantes utilizando informações visuais - Estratégia Líder-Seguidor - Garantia de factibilidade.	173

Lista de Tabelas

4.1	Matrizes de ponderação do <i>NMPC</i> para seguimento de caminho circular - Robô omnidirecional - Resultado experimental - Caso Ideal. . .	62
4.2	Matrizes de ponderação do <i>NMPC</i> para seguimento de caminho circular - Robô omnidirecional - Resultado experimental - Robô distante do caminho.	62
4.3	Matrizes de ponderação - Simulação 1.	69
4.4	Matrizes de ponderação - Simulação 2.	71
4.5	Matrizes de ponderação.	82
4.6	Matrizes de ponderação.	84
4.7	Matrizes de ponderação.	88
4.8	Matrizes de ponderação.	93
4.9	Matrizes de ponderação.	97
4.10	Análise de desempenho - Comparação das abordagens para o <i>VPF</i> - $v = 0, 1$ [m/s]	104
4.11	Análise de desempenho - <i>VPF</i> com garantia de factibilidade para 2 velocidades	104
4.12	Análise estatística do tempo de processamento - <i>VPF</i> nominal. . . .	107
4.13	Análise estatística do tempo de processamento - <i>VPF</i> com garantia de estabilidade.	107
4.14	Análise estatística do tempo de processamento - <i>VPF</i> com garantia de factibilidade.	107
4.15	Análise estatística de J_{min} - <i>VPF</i> nominal.	108
4.16	Análise estatística de J_{min} - <i>VPF</i> com garantia de estabilidade. . . .	108
4.17	Análise estatística de J_{min} - <i>VPF</i> com garantia de factibilidade. . . .	108
4.18	Análise estatística de $ u_{min} $ - <i>VPF</i> nominal.	109
4.19	Análise estatística de $ u_{min} $ - <i>VPF</i> com garantia de estabilidade. . . .	109
4.20	Análise estatística de $ u_{min} $ - <i>VPF</i> com garantia de factibilidade. . .	109
4.21	Análise de desempenho - <i>VPF</i> nominal	110
4.22	Análise de desempenho - <i>VPF</i> com garantia de estabilidade	110
4.23	Análise de desempenho - <i>VPF</i> com garantia de factibilidade	110
5.1	Matrizes de ponderação para os controladores de formação - Robôs omnidirecionais - Caso ideal.	128

5.2	Matrizes de ponderação para os controladores de formação - Robôs omnidirecionais - Caminhos e formações variantes no tempo.	129
5.3	Matrizes de ponderação.	134
5.4	Matrizes de ponderação.	143
5.5	Matrizes de ponderação.	151
5.6	Matrizes de ponderação.	158
5.7	Matrizes de ponderação para os seguidores	162

Lista de Siglas

<i>MPC</i>	Model Predictive Control
<i>NMPC</i>	Non-Linear Model Predictive Control
<i>IBVS</i>	Image Based Visual Servoing
<i>PBVS</i>	Position Based Visual Servoing
<i>UAVs</i>	Unmanned Aerial Vehicles
<i>NASA</i>	National Aeronautics and Space Administration
<i>VTOL</i>	Vertical Take-off and Landing
<i>SB_{ij}C</i>	Separation Bearing Control
<i>DC</i>	Direct Current
<i>PID</i>	Proporcional Integral e Derivativo
<i>PD</i>	Proporcional e Derivativo
<i>ROS</i>	Robot Operating System
<i>CPU</i>	<i>Central Process Unit</i>
<i>I/O</i>	<i>Input/Output</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>OS</i>	<i>Operational system</i>
<i>MAC</i>	<i>Macintosh Computer ou Model Algorithmic Control</i>
<i>DMC</i>	<i>Dynamic Matrix Control</i>
<i>ARM</i>	<i>Acorn RISC Machine</i>
<i>BSD</i>	<i>Berkeley Software Distribution</i>
<i>BeOS</i>	<i>Be Inc Operational System</i>
<i>DOS</i>	<i>Disk Operating System</i>
<i>GPL</i>	<i>General Public License</i>
<i>GUI</i>	<i>Graphical user interface</i>
<i>RS-232</i>	<i>Recommended Standard 232</i>
<i>bps</i>	<i>Bits per second</i>
<i>IEEE</i>	<i>Institute of Electrical and Electronics Engineers</i>
<i>RF</i>	<i>Rádiofrequência</i>
<i>wi-fi</i>	<i>Wireless Fidelity</i>
<i>WLAN</i>	<i>Wireless Local Area Network</i>
<i>ISM</i>	<i>Industrial, scientific and medical</i>
<i>IP</i>	<i>Internet Protocol</i>
<i>UART</i>	<i>Universal Asynchronous Receiver Transmitter</i>

<i>I2C</i>	<i>Inter-Integrated Circuit</i>
<i>FreeRTOS</i>	<i>Free Real-Time Operational System</i>
<i>API</i>	<i>Application Program Interface</i>
<i>PC</i>	<i>Personal Computer</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>RAM</i>	<i>Random Access Memory</i>
<i>fps</i>	<i>frames per second</i>
<i>UDP</i>	<i>User Datagram Protocol</i>
<i>C</i>	Linguagem de programação
<i>.txt</i>	Extensão dos arquivos de texto
<i>HD</i>	High Definition

Lista de Símbolos

A menos que referência contrária seja fornecida, os símbolos matemáticos a seguir possuem os seguintes significados:

n : ou c	Índice representativo do sistema de coordenadas do robô
m :	Índice representativo do sistema de coordenadas do mundo
R_n^m :	Matriz de rotação do sistema n para o sistema m
$x[m]$:	Coordenada horizontal da posição do robô relativa ao sistema m
$y[m]$:	Coordenada vertical da posição do robô relativa ao sistema m
$\theta[rad]$:	Orientação do robô relativa ao sistema m
$R_{z,\theta}$:	Matriz de rotação de um ângulo θ em torno de z
$x_r[m]$:	Coordenada horizontal da posição de referência
$y_r[m]$:	Coordenada vertical da posição de referência
$\theta_r[rad]$:	Orientação de referência
$v[m/s]$ ou $v_x[m/s]$:	Componente horizontal da velocidade linear do robô
$v_n[m/s]$ ou $v_y[m/s]$:	Componente vertical da velocidade linear do robô
$v_z[m/s]$:	Velocidade linear do robô ao longo do eixo z
$\omega[rad/s]$:	Velocidade angular de giro do robô
L :	Distância entre as rodas e o centro de massa do robô
\mathbf{L} :	Matriz das reduções dos motores
\mathbf{R} :	Matriz dos raios das rodas
l :	Distância entre as rodas
$\omega_r[rad/s]$:	Velocidade angular de referência
$v_{nav}[m/s]$:	Velocidade de navegação do robô
$\{F\}$:	Sistema de coordenadas <i>Serret-Frenet</i>
$\{m\}$:	Sistema de coordenadas do mundo
\mathbb{R} :	Conjunto dos números reais
\mathbb{R}_+ :	Conjunto dos números reais não negativos
\mathbb{Z}_+ :	Conjunto dos números inteiros não negativos
$\overline{\lim} \mathcal{X}$:	Limitante superior do conjunto \mathcal{X}
\mathbb{R}^n :	Espaço dos vetores coluna reais com n elementos
$\mathbb{R}^{g \times n}$:	Espaço das matrizes reais reais com g linhas e n colunas
\mathbb{N} :	Conjunto dos números naturais
\mathbb{C}^n :	Espaço dos vetores coluna complexos com n elementos
$\{c\}$ ou $\{B\}$:	Sistema do robô

$\delta[rad]$:	Ângulo da direção de movimento relativo ao sistema ao sistema c
$\phi[rad]$:	Ângulo da direção de movimento relativo ao sistema ao sistema m
Γ :	Caminho de referência
$R(\cdot)$:	Matriz de rotação
$s[m]$:	Comprimento de caminho
$\dot{s}[m/s]$:	Taxa de atualização das posturas de referência
$\mathbf{x}_e[m]$:	Vetor de erros de estado
$x_e[m]$:	Componente horizontal do erro de estado
$y_e[m]$:	Componente vertical do erro de estado
$\phi_e[rad]$:	Erro do sentido de movimento do robô
$\theta_e[rad]$:	Erro de orientação
$K(\cdot)$:	Função curvatura do caminho Γ
R_i :	Robô i
\mathbf{u}_e :	Vetor de entradas
$p_{ij}(\cdot)$:	Função distância desejada entre dois robôs
\mathbf{E} :	Vetor erros de formação
\mathbf{p} :	Vetor das distancias entre os robôs i e j
\mathbf{s} :	Vetor das posições ao longo do caminho
\mathbf{C} :	Matriz de incidência
\mathbf{x} :	Vetor de estados
\mathbf{u} :	Vetor de entradas
Γ_i :	Caminho de referência para o robô i
Γ_r :	Caminho de referência base
$\theta_{i,c}$:	Ângulo da tangente ao caminho de referência para o robô i
\mathbf{x}_{B_i} :	Vetor das posturas ao longo do caminho base para o robô i
\mathbf{x}_{r_i} :	Vetor das posturas de referência para o robô i
$q_i[m]$:	Distância transversal entre R_i e Γ_r
$P(\cdot)$:	Função comprimento total de caminho
$\xi_i[m]$:	Comprimento do caminho para o robô i
$\dot{\xi}_i[m/s]$:	Taxa de atualização das posturas para o robô i
$v_{nav_B}[m/s]$ ou $u_{f_B}[m/s]$:	Velocidade de navegação base para a formação
$v_{nav_{r_i}}[m/s]$ ou $u_{f_{r_i}}[m/s]$:	Velocidade de navegação de referência para o robô i
$v_{nav_i}[m/s]$ ou $u_{f_i}[m/s]$:	Velocidade de navegação do robô i
$\omega_{r_i}[rad/s]$:	Velocidade angular de referência para o robô i
$K_r(\cdot)$:	Função curvatura do caminho de referência base
$K_i(\cdot)$:	Função curvatura do caminho de referência para o robô i
$\eta_i[m/s]$:	Estado diferença de velocidade
ψ_i :	Aceleração atual do robô i
γ_i :	Aceleração de referência para o robô i
\mathbf{u}_{e_i} :	Vetor de entradas para o robô i
$\mathbf{u}_{e_i}^*$:	Vetor de entradas ótimas para o robô i

\mathbf{x}_{e_i} :	Vetor de estados para o robô i
\mathcal{U}_i :	Conjunto de entradas factíveis para o robô i
\mathcal{X}_i :	Conjunto de estados factíveis para o robô i
T_s :	Período de amostragem
$J(\cdot)$:	Função objetivo
$F(\cdot)$:	Função custo de estágio
\mathbf{Q} :	Matriz de ponderação dos erros de estado
\mathbf{R} :	Matriz de ponderação dos esforços de controle
T_P :	Horizonte de predição
T_C :	Horizonte de controle
ε :	Conjunto de vetores relativos entre os robôs
W_{u_i} :	Constante de ponderação das velocidades relativas
W_{s_i} :	Constante de ponderação dos erros de espaçamento relativo
\mathbf{I}_n :	Matriz identidade de ordem n
$diagonal(\cdot)$:	Matriz diagonal
\min :	Minimize (Problema)
\inf :	ínfimo matemático
$\max(\cdot)$:	Função para a determinação do máximo elemento de um vetor
$f^{[n]}$:	Derivada de ordem n de f
H :	Horizonte para o seguimento de caminhos visuais
z :	Espaçamento lateral entre robô em caminho
θ_r :	Ângulo entre o robô e a tangente ao caminho
lij :	Distância atual entre o robô líder i e o seguidor j
ψ_{ij} :	Ângulo lij e o robô líder
β_{ij} :	Ângulo robô líder i e o seguidor j
l^r_{ij} :	Distância desejada entre o robô líder i e o seguidor j
ψ^r_{ij} :	Ângulo lij e o robô líder

Índice

Agradecimentos	vii
Resumo	ix
Abstract	xi
Publicações	xiii
Lista de Figuras	xv
Lista de Tabelas	xix
Lista de Símbolos	xxiii
Índice	xxvi
1 Introdução	1
1.1 Sobre o controle de formação	2
1.2 Definição do Problema	4
1.3 Motivação e Trabalhos Relacionados	7
1.4 Objetivos e Contribuições	8
1.5 Estrutura do texto	10
2 Robótica Móvel	11
2.1 Locomoção de robôs móveis	11
2.2 Sensores e atuadores	13
2.3 Representação de Posições e Orientações	14
2.4 Modelos Utilizados	15
2.4.1 Robôs móveis omnidirecionais	16
2.4.2 Robôs móveis diferenciais	20
2.4.3 Quadricópteros	22
2.5 Plataforma Experimental	25
2.5.1 O software <i>Decision</i>	25
2.5.2 Arquitetura Descentralizada	27

2.5.3	Sistema Visual	31
2.6	Conclusões	33
3	Controle Preditivo Baseado em Modelo Não linear	34
3.1	Preliminares teóricas	34
3.1.1	Existência e unicidade de soluções	35
3.1.2	Estabilidade Segundo Lyapunov	36
3.2	Controladores <i>NMPC</i>	41
3.2.1	Predição	44
3.2.2	Problema de controle ótimo	46
3.2.3	Otimizadores Não lineares	52
3.3	Conclusões	54
4	Seguimento de Caminho	55
4.1	Seguimento de caminho para robôs omnidirecionais	56
4.1.1	Formalização do Problema	56
4.1.2	Projeto dos Controladores	60
4.1.3	Resultados experimentais	61
4.2	Seguimento de caminhos para quadricópteros	65
4.2.1	Formalização do Problema	65
4.2.2	Projeto dos Controladores	67
4.2.3	Resultados simulados	68
4.3	Controle servo visual	73
4.3.1	Transformação perspectiva	73
4.3.2	Controle Visual Baseado em Imagem (<i>IBVS</i>)	75
4.4	Seguimento de caminhos visuais	76
4.4.1	Formalização do Problema	76
4.4.2	Projeto dos Controladores	79
4.4.3	Resultados	81
4.5	Conclusões	116
5	Controle de Formação	118
5.1	Caracterização da Formação	119
5.2	Controle de Formação para Robôs Ominidirecionais	122
5.2.1	Projeto dos controladores	125
5.2.2	Resultados Experimentais	125
5.3	Controle de Formação para Robôs Diferenciais	132
5.3.1	Resultados experimentais	133
5.4	Controle de Formação para Quadricópteros	140
5.4.1	Projeto dos Controladores	142
5.4.2	Resultado Simulado	143
5.5	Controle de Formação Utilizando Informações Visuais	146
5.5.1	Estratégia baseada em câmeras virtuais	147

5.5.2	Estratégia Líder-Seguidor	154
5.6	Conclusões	174
6	Conclusões e Trabalhos Futuros	176
6.1	Principais contribuições	178
6.2	Perspectivas de Trabalhos Futuros	179
6.3	Considerações Finais	179
	Bibliografia	180

Capítulo 1

Introdução

A busca contínua por soluções inovadoras para os diversos problemas da sociedade contemporânea caracteriza um cenário em que sistemas autônomos possuem grande relevância. A interação entre diversas áreas do conhecimento dá suporte a esta busca, sendo a robótica móvel, um ponto de convergência natural, haja vista o seu caráter interdisciplinar.

Surge como consequência imediata das analogias com as atividades humanas, a necessidade de uma abordagem cooperativa. As investigações sobre os sistemas autônomos cooperativos possuem enfoques que variam num vasto espectro, caracterizado em diversos níveis de abstração.

Dentre os principais benefícios da utilização destes tipos de sistemas, podem ser citados flexibilidade, robustez e adaptabilidade a ambientes desconhecidos (Lim *et al.* 2009a). Tais benefícios são claramente importantes em aplicações tais como missões de busca e resgate (Eoh *et al.* 2013), detecção de incêndio em florestas (Casbeer *et al.* 2006), vigilância (Weisbin e Perillard 1991), dentre outras.

São diversas as inspirações que norteiam o desenvolvimento de sistemas cooperativos, sendo estas resumidamente apresentadas em (Arai *et al.* 2002) para o caso de sistemas multi-robôs. Em (Murray 2007) tem-se uma descrição mais detalhada, feita através da apresentação do ferramental matemático comum a várias abordagens, além de discussões a respeito de áreas técnicas específicas, tais como controle de formação (Egerstedt e Hu 2001), tarefas cooperativas (Miyata *et al.* 2002), planejamento espaço-temporal (Likhachev *et al.* 2002) e consenso (Ren 2007).

Dentre as aplicações destacam-se as denominadas “*testbeds*” que proporcionam ambientes adequados ao desenvolvimento de sistemas para diversos tipos de aplicações, como por exemplo, tem-se a *RoboCup* (Kitano *et al.* 1998, Conceição *et al.* 2006, Stuckler *et al.* 2012, Budden *et al.* 2015) como uma das mais bem difundidas. Nesta aplicação, sistemas multi-robôs são utilizados em ambientes altamente dinâmicos, o que fornece mecanismos eficientes para validação prática.

Muitas das aplicações de controle cooperativo possuem como parte de sua solução tarefas de controle de formação para um dado número de veículos. Trata-se de um dos tópicos mais estudados no âmbito de sistemas multi-robôs (Oh *et al.* 2015).

1.1 Sobre o controle de formação

Uma formação é definida como um arranjo especial de um certo número de robôs para o cumprimento de um determinado objetivo (Nascimento *et al.* 2011). Dentre estes objetivos podem ser citados controle de tráfego aéreo (Tomlin *et al.* 1998), controle de orientação de um grupo de satélites (Burns *et al.* 2000) e realização de tarefas em ambientes hostis (Li *et al.* 2002).

Simplificadamente, pode-se definir como objetivo do controle de formação o rastreamento de um determinado caminho espacial enquanto mantém-se um determinado padrão de espaçamento entre os robôs (Kanjawanishkul e Zell 2008a). De maneira geral, existem três abordagens básicas para a implementação deste rastreamento: estabilização em pontos de trajetória (Fierro e Lewis 1996), rastreamento de trajetórias (Divelbiss e Wen 1997) e seguimento de caminhos (Samson 1995).

Tais abordagens são distintas tanto em sua forma de implementação quanto em termos de aplicabilidade. Normalmente, os pontos, as trajetórias ou caminhos são pré-especificados ou são obtidos de um nível superior que tem algum conhecimento sobre o ambiente de navegação.

Em (Chen e Wang 2005) tem-se sumarizadas as principais características inerentes ao projeto de sistemas para controle de formação de robôs móveis, tais como estabilidade, controlabilidade e aspectos relativos às implementações práticas. Em tal trabalho, as abordagens comumente utilizadas para o controle de formação são divididas em cinco grandes grupos, sendo que as definidas como estrutura virtual, baseada em comportamento e Líder-Seguidor são as mais utilizadas. Estes autores contribuem ainda com uma analogia entre controle de formação e controle clássico ao elucidar como objetivos, controle de rastreamento de formação e controle de regulação de formação.

A estrutura virtual trata o problema de controle de formação como um corpo rígido onde todos os robôs possuem suas posições fixas. Tal abordagem foi originalmente proposta em (Tan e Lewis 1996), sendo bastante utilizada posteriormente em diversas aplicações tais como em (Ren e Beard 2003) para o controle de formação de veículos espaciais e em (Ghommam *et al.* 2008) para robôs do tipo monociclo. Dentre os principais problemas desta abordagem é possível citar o seu mau desempenho no caso regulatório, bem como, o demasiado aumento do custo computacional na medida em que se aumenta o número de veículos.

A abordagem baseada em comportamento é sempre utilizada em conjunto com outra denominada campos potenciais (Gazi 2003). Tal abordagem é definida através da determinação de comportamentos reativos dos elementos da formação, bem como, outros definidos através dos objetivos (Balch e Arkin 1998). Verificam-se exemplos desta abordagem em (Antonelli *et al.* 2006) e (Liang e Xiao 2010). O principal problema com esta abordagem é a necessidade de uma fiel descrição dinâmica do grupo.

Dentre as estratégias para o controle de formação, a mais utilizada é aquela denominada por Líder-Seguidor (Wang 1989). Nesta abordagem, o líder pode ser um

robô real, virtual, ou mesmo um alvo a ser perseguido. Exemplos de utilização desta abordagem podem ser verificados em (Priscoli *et al.* 2015) e (Qiu *et al.* 2016). A principal desvantagem desta abordagem é a ausência de um caminho de realimentação dos seguidores para o líder tornando o sistema suscetível a falhas. Caso os seguidores não consigam regular os erros de formação individuais, a formação será perdida e o líder não irá perceber.

A maioria destas abordagens utiliza controle baseado em modelo, *MPC* (do inglês, *Model Predictive Control*), como estratégia de controle (Xiao *et al.* 2016, Lee e Myung 2015, Fukushima *et al.* 2013). Tal estratégia consiste numa classe de algoritmos que utilizam o modelo explícito dos processos para calcular uma sequência de ajustes no sinal de controle de forma a otimizar o comportamento futuro da saída de uma planta (Camacho e Bordons 2004, Capítulo 1). Dentre as vantagens que viabilizam a escolha deste tipo de estratégia cita-se a sua capacidade de controlar sistemas monovariáveis e multivariáveis, com retardo, instáveis e de fase não-mínima.

Particularmente para o controle de formação, esta estratégia é igualmente aplicável tanto para um grupo de robôs holonômicos como em (Kanjawanishkul e Zell 2008b) quanto para robôs não-holonômicos como em (Lim *et al.* 2009b), com características similares no que concerne aos requisitos de estabilidade. Características importantes, tais como tratamento explícito de restrições e bons níveis de robustez inerente, são contribuições adicionais para a escolha desta estratégia.

Em termos práticos, um aspecto que pode restringir as funcionalidades dos sistemas de controle de formação é a forma como a lei de controle é implementada, ou seja, se é uma estratégia centralizada (Ali e Montenegro 2016) ou descentralizada (Liu *et al.* 2016). A abordagem centralizada demanda mais recursos de hardware e software já que um maior volume de informação será utilizado para o cálculo das ações de controle, já na descentralizada há uma redução neste volume podendo-se até, em alguns casos, simplificar os requisitos de complexidade algorítmica para a manutenção da estabilidade da formação. Em contrapartida, nesta última abordagem, problemas de comunicação tais como perda de informação e latência tendem a ser mais críticos. Em todos os casos existem vantagens e desvantagens sendo em (Das *et al.* 2002) propostos meios de se conciliar os benefícios.

Uma abordagem vantajosa, do ponto de vista prático, baseia-se na utilização direta de controladores preditivos baseados em modelos não lineares, *NMPC* (do inglês, *Nonlinear Model Predictive Control*), para a manutenção das distâncias entre os veículos, objetivando-se o seguimento de caminhos coordenados. Tal manutenção é alcançada através do cálculo das velocidades que cada robô deve ter para o seguimento de caminhos individuais obtidos a partir de um caminho base, previamente definido nas proposições originais. Neste caso, trata-se de uma abordagem descentralizada onde se acoplam, restrições (Richards e How 2004), estados (Jia 2002) ou funções objetivo (Dunbar e Murray 2005).

Dentre estas formas de acoplamento, a mais eficiente do ponto de vista computacional é a baseada em função objetivo, já que o termo de acoplamento pode ser caracterizado adequadamente objetivando a manutenção de requisitos específicos de

otimalidade. Além disso, com esta abordagem não são necessárias novas caracterizações da dinâmica do sistema em função do número de robôs, fator que influencia diretamente na escalabilidade, que por sua vez define a quantidade de dados a serem compartilhados pelos robôs da formação.

1.2 Definição do Problema

As abordagens clássicas para o controle de formação de robôs móveis apresentam os problemas intrínsecos elencados anteriormente. Adicionalmente, os controladores *NMPC* convencionais, também apresentam problemas no que concerne aos requisitos de estabilidade, factibilidade, robustez e custo computacional. Soluções completas e adequadas a diversos tipos de sistemas estão disponíveis (Bemporad e Morari 1999, Mayne *et al.* 2000, Löfberg 2012, He e Han 2012), porém, um compromisso formal para o solucionamento de todos os problemas é bastante complexo e alvo de contínuas investigações.

A própria utilização do *NMPC* para a solução dos problemas de seguimento de caminho e controle de formação, naturalmente produz problemas oriundos de variabilidade temporal das referências de postura e velocidade. Isto se deve, originalmente, à forma como os conjuntos das restrições são definidos, não sendo possível garantir factibilidade inicial tampouco recursiva. As Figuras 1.1(a) e 1.1(b) ilustram o desempenho do controlador proposto em (Ribeiro e Conceição 2012) aplicado ao rastreamento de formações triangulares de três robôs diferenciais, ao longo de caminhos senoidais, quando nenhuma técnica para garantia de factibilidade é utilizada.

Verifica-se que, mesmo de baixa magnitude, os erros iniciais não são regulados devido à infactibilidade inicial, oriunda de violações das restrições associadas aos estados do sistema. Ratificando este problema, a Figura 1.1(c) apresenta o número de amostras gastas no processo de otimização. Sabe-se que, quando forem necessárias mais de uma amostra, as informações de robôs vizinhos não são coletadas adequadamente, inviabilizando o decaimento das funções objetivo locais, implicando em instabilidade da formação, conforme ilustrado na Figura 1.1(d).

Deste modo, confirma-se que, a definição de seguimento de caminhos espaciais como tarefa para a formação, simplesmente considerando o acoplamento via função objetivo, restringe a denominação da abordagem a seguimento de caminho coordenado, já que os erros de formação apenas são corrigidos em instantes em que o problema de otimização for factível. Tal aspecto motiva o desenvolvimento de técnicas para garantia de factibilidade para controladores *NMPC*, sem prejuízos na eficiência computacional.

A utilização de caminhos pré-estabelecidos, em primeira instância, restringe a aplicabilidade a ambientes estáticos, sendo deste modo destituído um dos principais benefícios da utilização de sistemas autônomos cooperativos. Para estender a gama de aplicações, considerando a disponibilidade de sensores visuais de baixo custo, os caminhos de referência podem ser detectados em tempo de execução através de um

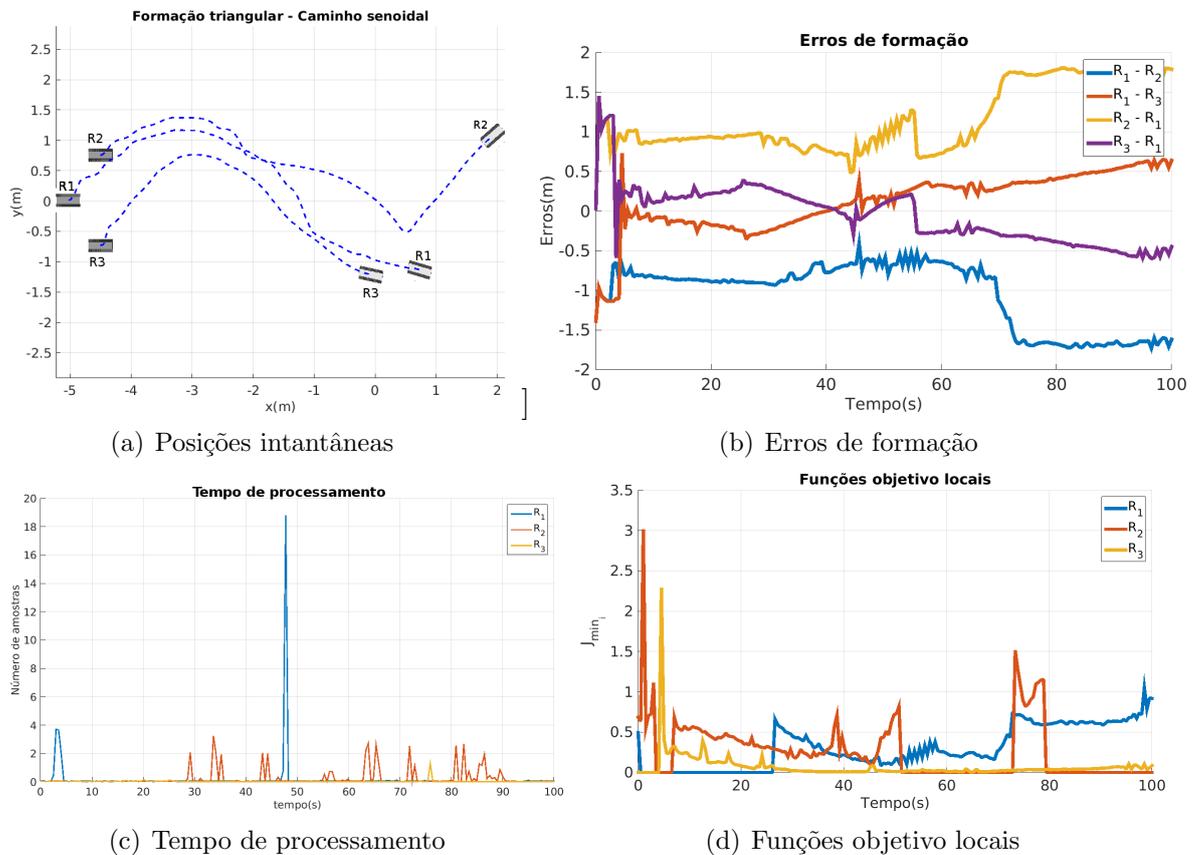


Figura 1.1: Problema de infactibilidade inicial do controle de formação baseado em *NMPC* - Resultado experimental.

sistema de visão computacional.

Para a manutenção dos requisitos de eficiência computacional, é preciso formalizar uma abordagem para seguimento de caminho baseada em visão inspirando-se em técnicas de servovisão baseada em imagem (*IBVS*, do inglês, *Image Based Visual Servoing*) já consolidadas (Chaumette e Hutchinson 2006, Chaumette e Hutchinson 2007).

O Controle Visual baseado em imagem consiste na extração de características visuais simplificadas oriundas de dados de visão computacional, como entrada de uma malha de controle de postura de um sistema dinâmico, sem que seja estimada uma postura absoluta (*PBVS*, do inglês, *Position Based Visual Servoing*) dessa forma o controle se dá diretamente no plano visual reduzindo em larga escala a complexidade algorítmica.

As soluções para o seguimento de caminhos visuais podem ser estendidas para o controle de formação, aumentando a gama de aplicações práticas. Assim como para caminhos pré definidos, as estratégias para controle de formação utilizando informações visuais podem ser concebidas objetivando a aplicação direta do *NMPC* ou através da utilização de uma das abordagens clássicas.

Especificamente, esta tese visa solucionar os seguintes problemas fundamentais:

Problema 1: *Controlar formações de um grupo de robôs de diferentes tipos, baseando-se em NMPC computacionalmente eficiente, com factibilidade e estabilidade garantidas.*

Problema 2: *Generalização das aplicações em potencial através da utilização de informações visuais.*

A solução destes problemas baseia-se na decomposição em subproblemas no âmbito dos sistemas robóticos distribuídos. Aspectos intrínsecos às estratégias de movimentação de robôs móveis e visão computacional são estudados para a geração de soluções parciais que contemplem os problemas fundamentais.

Na medida em que estes subproblemas são solucionados, surgem novas soluções para aqueles outros relacionados a estratégias convencionais, conforme ilustrado no diagrama da Figura 1.2.

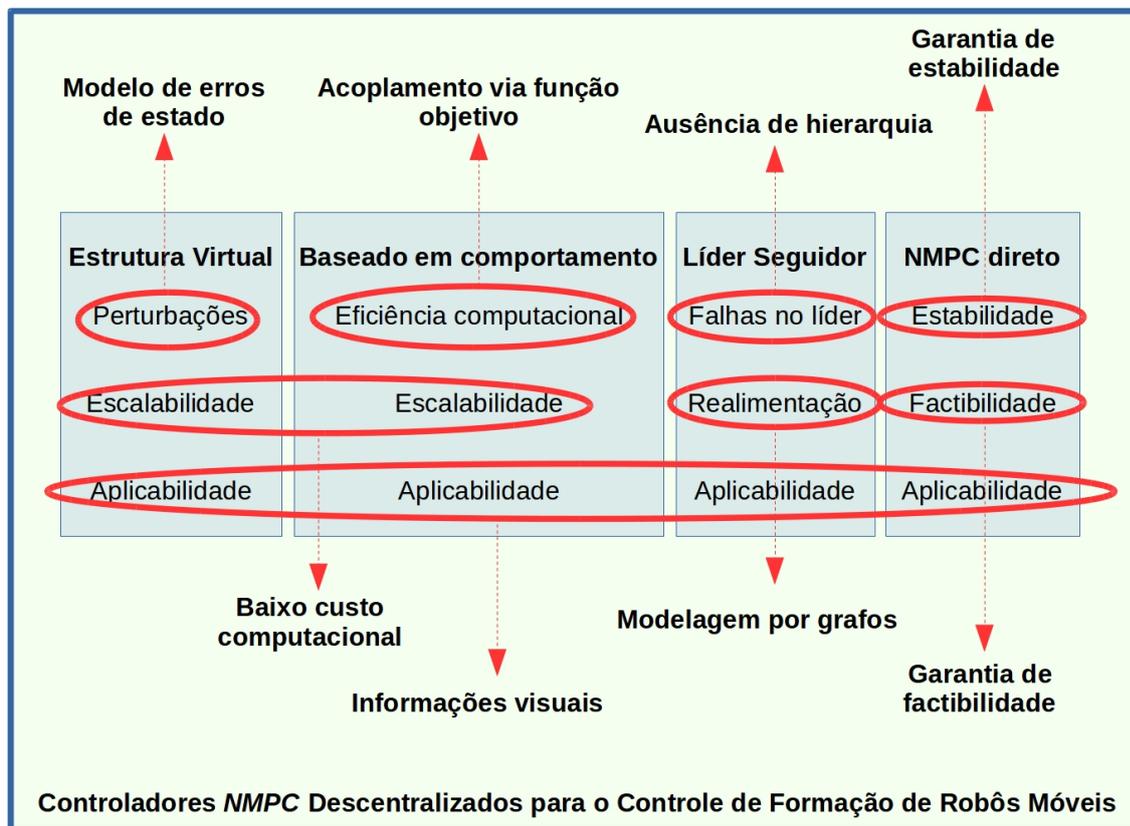


Figura 1.2: Síntese das soluções propostas para os problemas com as abordagens clássicas para controle de formação.

1.3 Motivação e Trabalhos Relacionados

Os impactos sociais previstos através dos diversos tipos de aplicações evidenciam a importância deste tema. Para ilustrar tal importância no cenário tecnológico atual, pode-se citar, por exemplo, três aplicações onde já se verificam avanços substanciais e os sistemas de controle de formação já se encontram em fase de testes: Redes de sensores móveis, sistemas militares e sistemas de transportes.

As redes de sensores móveis são muito utilizadas em pesquisa exploratória como por exemplo em (Monterey 2006) onde tem-se uma rede de veículos autônomos para amostragem ambiental em oceanos. Neste caso a estratégia baseada em comportamento (campos potenciais + líderes virtuais) é utilizada. A NASA aplica controle de formação, através da técnica Líder-Seguidor, para um conjunto de aeronaves não tripuladas, UAV (do inglês, *Unmanned Aerial Vehicle*), numa configuração tal que reduzem as forças de arrasto (Lavretsky 2002). O problema de otimizar o tráfego de veículos em rodovias muito movimentadas pode ser solucionado através do conceito de rodovias inteligentes onde se utiliza controle de formação para a condução de veículos em pelotões com velocidade e espaçamento determinados automaticamente (Path 2006). Normalmente, para esta aplicação, estratégias híbridas são utilizadas para o controle de formação (Baskar *et al.* 2011, EL-Zaher *et al.* 2012, Zhao e Sun 2013, Larson *et al.* 2014).

A concepção de um sistema real e com tecnologia própria, mediante o qual se possam implementar algoritmos para o controle de formação baseado em *NMPC* com factibilidade e estabilidade garantidas, é uma contribuição de grande valia no cenário acadêmico atual. Adicionalmente, a utilização de visão computacional e requisitos explícitos de eficiência computacional, implica no desenvolvimento de uma plataforma experimental que permite atender à demanda por resultados práticos em diversas esferas da robótica móvel, como por exemplo, cooperação em nível cognitivo.

Além disto, os esforços recentemente dedicados em grandes centros à pesquisa sobre sistemas para controle de formação justificam o desenvolvimento de um trabalho desta natureza. Apenas na base de dados do *IEEE*, foram encontrados 6458 artigos publicados em periódicos e conferências nos últimos 10 anos. Além disto, o número de conferências nas áreas de Controle e Automação, Engenharia Elétrica e Ciência da Computação que adicionaram este tema aos seus tópicos de interesse também é bastante elevado.

Dentre estas publicações, algumas se relacionam diretamente com o tema desta tese. Em (Mehrez *et al.* 2014) o desempenho das implementações centralizadas e descentralizadas para a solução do problema de controle de formação baseado em *NMPC* é comparado. São utilizados modelos de robôs diferenciais e resultados simulados são apresentados.

Formações de veículos aéreos não tripulados são tratadas em (Shin e Kim 2009). Três arquiteturas para a implementação dos controladores *NMPC* são propostas, centralizada, centralizada sequencial e descentralizadas. Resultados simulados são também apresentados para validação da proposta.

Problemas oriundos das imperfeições do sistema de comunicação para controladores Líder-Seguidor distribuídos são abordados em (El-Ferik *et al.* 2013). Uma solução baseada em *NMPC* torna desnecessária a caracterização dinâmica de robôs vizinhos e desvio de obstáculos são garantidos. Além disso o problema de controlar a formação de robôs de diferentes tipos é considerado. Este mesmo autor aborda problemas com estabilidade em (El-Ferik *et al.* 2016), contribuindo com uma formulação denominada *Entrada para Estado Praticamente Estável*. Resultados simulados validam a proposta.

Visando o tratamento de formações variantes no tempo, (Kanjanawanishkul e Zell 2008c) propõe uma arquitetura Líder-Seguidor baseada em *NMPC* com estabilidade local garantida através de custo e restrições terminais. Resultados experimentais mostram que, desde que não existam falhas no robô líder, a formação será mantida com previsão de variação ao longo de caminhos pré-definidos. Em (Kanjanawanishkul 2009) este mesmo autor faz análise de estabilidade dos controladores *NMPC* para o controle de formação descentralizado. Resultados experimentais são fornecidos para o seguimento de caminhos e formações constantes e pré definidas.

De maneira geral, a análise crítica destes e de outros trabalhos consultados permite concluir que o problema de controle de formações e caminhos não pré definidos e variantes no tempo, para robôs de diferentes tipos, baseado em *NMPC* com factibilidade e estabilidade garantidas necessita de propostas de soluções efetivas e experimentalmente comprovadas.

1.4 Objetivos e Contribuições

Este trabalho objetiva o desenvolvimento de algoritmos para o controle de formações variantes no tempo para um conjunto de robôs móveis de diferentes tipos. A estratégia escolhida baseia-se na utilização direta de controladores *NMPC*, com factibilidade e estabilidade garantidas, para a manutenção de padrões de formação entre os robôs, enquanto estes seguem caminhos individuais no espaço pré definidos ou obtidos através de características visuais.

Especificamente, numa primeira etapa, de posse de resultados de arquiteturas legadas (Ribeiro *et al.* 2011b, Ribeiro e Conceição 2012, Ribeiro *et al.* 2013), propõe-se uma estratégia baseada no relaxamento das restrições para garantia de factibilidade recursiva. Adaptações feitas ao seguimento de caminho coordenado possibilita a manutenção do acoplamento via função objetivo e denominação apropriada de controle de formação, sendo viabilizado o tratamento de formações e caminhos variantes no tempo. Através da proposta de garantia de factibilidade e estabilidade, pretende-se tornar os termos de acoplamento calculáveis e comunicáveis a todo instante de tempo, de modo que sempre será possível reduzir a função custo mesmo sem restrições explícitas. Ao longo de todo o desenvolvimento são considerados recursos computacionais limitados, o que viabiliza a aquisição de resultados reais.

Em seguida, as estratégias originais para seguimento de caminho e seguimento de

caminho coordenado, desenvolvidas para robôs terrestres, são estendidas para o caso tridimensional, objetivando a utilização de quadricópteros. Estes *UAVs* apresentam grandes vantagens comparativamente a outros tipos de aeronaves. Tal extensão se baseia numa estrutura em cascata composta de *NMPC* e de controladores clássicos.

Considerando que a tarefa para a formação consiste no seguimento de caminhos espaciais, é proposta uma arquitetura baseada em visão computacional com baixa complexidade algorítmica. Para tal fim, características que podem ser facilmente extraídas das imagens são utilizadas na proposição de uma estratégia para seguimentos de caminhos visuais baseado em *NMPC* com garantias de factibilidade e estabilidade. Com a utilização do sistema visual, não existe a necessidade de pré estabelecer caminhos de referência, sendo viabilizada a utilização dos controladores propostos em aplicações mais próximas da realidade tais como navegação em ambientes dinâmicos e não estruturados.

Posteriormente são propostas duas estratégias para controle de formação utilizando informações visuais, que possibilite variação temporal de parâmetros, tanto da formação, quanto dos caminhos seguidos e que mantenha os requisitos de factibilidade, estabilidade e eficiência computacional.

A primeira proposta baseia-se no seguimento de um caminho visual¹ gerador de caminhos virtuais para os elementos da formação que não dispõem de sistema de visão. Por sua vez, na segunda proposta, o seguimento de caminhos visuais é aplicado apenas para um robô líder que envia seus estados atuais para robôs seguidores, sendo, neste caso, implementada a estratégia Líder-Seguidor em que se controla distância e ângulo dos seguidores relativamente ao líder. Em ambos os casos, utiliza-se controladores *NMPC*.

As principais contribuições desta tese estão intimamente relacionadas à extensão da aplicabilidade das técnicas para controle de formação, conforme listagem a seguir:

- Estudo, síntese e implementação prática de abordagens clássicas para o controle de formação baseado em *NMPC*;
- Proposta de garantia de factibilidade para controladores *NMPC* descentralizados;
- Proposição de uma nova abordagem para o seguimento de caminhos variantes no tempo;
- Proposição de uma nova abordagem para o controle de formações variantes no tempo;
- Controle de formação de robôs de diferentes tipos;
- Generalização das aplicações através da utilização de visão computacional.

¹Terminologia originada do termo em inglês, *visual path following* (Frezza et al. 1997).

1.5 Estrutura do texto

O presente trabalho está organizado em capítulos, conforme descrição a seguir:

Capítulo 2: Os sistemas robóticos são caracterizados. Conceitos básicos e os principais modelos dos robôs móveis utilizados são apresentados. Neste capítulo também são fornecidos detalhes das plataformas de simulação e experimental.

Capítulo 3: São apresentados os conceitos teóricos fundamentais inerentes aos controladores *NMPC*. São tratados aspectos relativos à factibilidade, estabilidade, otimalidade e custo computacional.

Capítulo 4: Este capítulo é dedicado à resolução do problema de seguimento de caminhos fixos, variantes no tempo e visuais para robôs de diferentes tipos. São fornecidos detalhes das proposições para garantia de factibilidade e estabilidade de controladores *NMPC* para a solução deste problema. Resultados experimentais e simulados são apresentados e discutidos.

Capítulo 5: Este capítulo é dedicado à resolução do problema de controle de formações fixas, variantes no tempo e utilizando informações visuais para robôs de diferentes tipos. São fornecidos detalhes das proposições para garantia de factibilidade e estabilidade de controladores *NMPC* distribuídos. Resultados experimentais e simulados são apresentados e discutidos.

Capítulo 6: Finalmente, as principais conclusões e expectativas de trabalhos futuros são fornecidas.

Capítulo 2

Robótica Móvel

Robôs móveis são preferencialmente utilizados em situações onde são requeridos elevados níveis de autonomia. Os fundamentos teóricos que caracterizam o funcionamento destes tipos de robôs são, em parte, comuns àqueles utilizados para robôs manipuladores, porém, com particularidades nos domínios mecânico e elétrico.

Como as fontes de energia elétrica são carregamentos obrigatórios dos robôs móveis, para níveis aceitáveis de eficiência, os mecanismos de locomoção devem ser bem caracterizados. Além disso, devido ao elevado grau de interação com o ambiente operacional, os requisitos para especificação dos sensores e atuadores são ainda mais restritivos. Este capítulo se inicia com a classificação básica dos mecanismos de locomoção, sensores e atuadores mais utilizados em robótica móvel.

São obtidos os modelos essenciais ao projeto de controladores para robôs móveis terrestres com locomoção diferencial e omnidirecional e robôs móveis aéreos *VTOL* (do inglês, *Vertical Take-off and Landing*) com múltiplos rotores. Características fundamentais de cada um destes modelos são apresentadas e discutidas.

Detalhamentos das plataformas de simulação e experimental são fornecidos. Especificamente são detalhados aspectos construtivos dos robôs utilizados, sistemas computacionais e estratégias para comunicação multi-robô.

Finalmente, conclusões sobre a robótica móvel no âmbito desta tese são fornecidas.

2.1 Locomoção de robôs móveis

Para o cumprimento dos seus objetivos, robôs móveis necessitam de mecanismos de locomoção que lhes proporcionem a liberdade de movimento requerida. Normalmente, utilizam-se elementos como rodas ou pernas que definem um conjunto de movimentos com imediata contrapartida biológica como, por exemplo, andar ou correr.

O mecanismo relacionado à ação do andar de seres humanos, por exemplo, pode ser aproximado por um polígono rolando, com lados de comprimento d igual à extensão do passo. Quando o tamanho do passo diminui, o polígono se aproxima de um círculo ou roda de raio l , como pode ser visto na Figura 2.1.

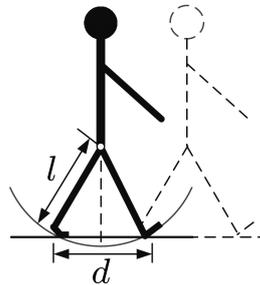


Figura 2.1: Mecanismo envolvido no andar dos seres humanos. Adaptado de (Siegwart e Nourbakhshi 2004, p. 15)

Na robótica móvel, assim como para qualquer sistema biológico, existe uma relação íntima entre energia e velocidade durante a movimentação, implicando no surgimento de restrições para a especificação dos mecanismos de locomoção, já que tais mecanismos definem também o quanto de energia será necessária à movimentação dos robôs. A Figura 2.2 mostra relações entre potência específica e velocidade alcançável para diversos mecanismos de locomoção.

Observa-se que as rodas são os mecanismos mais eficientes, porém sabe-se que tal eficiência dependerá também das condições da superfície em contato com as rodas, já que podem ser incrementadas as forças de atrito.

A locomoção dos robôs móveis é o complemento da manipulação dos robôs estacionários e em ambos os casos a base científica é o estudo de atuadores que geram as forças de interação, sendo assim os mecanismos que definem as propriedades ci-

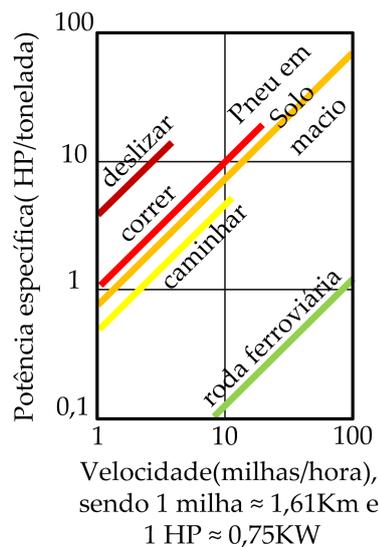


Figura 2.2: Relação entre potência específica e velocidade alcançável para diversos mecanismos de locomoção. Adaptado de (Siegwart e Nourbakhshi 2004, p. 15)

nemáticas e dinâmicas desejadas.

Tanto para manipulação como para locomoção, têm-se os mesmos problemas básicos relativos à estabilidade, características de contato e tipo de ambiente, conforme listados a seguir (Siegwart e Nourbakhshi 2004, Capítulo 2):

- Estabilidade
 - Número e geometria dos pontos de contato;
 - Centro de gravidade;
 - Estabilidade estática/dinâmica;
 - Inclinação do terreno.
- Características de contato
 - Ponto de contato/tamanho do caminho e forma;
 - Ângulo de contato;
 - Fricção.
- Tipos de ambiente
 - Estrutura;
 - Meio.

2.2 Sensores e atuadores

A implementação bem sucedida de qualquer estratégia de controle depende da seleção adequada de dispositivos sensores e atuadores compatíveis com as necessidades do projeto. Particularmente no caso dos sistemas robóticos, os critérios para a escolha destes dispositivos tornam-se cada vez mais rigorosos devido aos requisitos atuais de eficiência e eficácia.

Existe uma grande variedade de sensores aplicados à robótica, como por exemplo, codificadores óticos incrementais (*encoders*), giroscópios, acelerômetros, câmeras, etc., sendo todos estes operando, num cenário ideal, de maneira integrada para o fornecimento das informações precisas sobre os estados do sistema.

Dentre estes sensores, o menos custoso, de precisão regular e altamente integrável são os *encoders*. Tais dispositivos servem para medição indireta da velocidade das rodas para posterior cálculo das velocidades do centro de massa do robô, sendo bastante úteis para estimar as coordenadas do centro de massa do robô. Este mecanismo para medição indireta de posição, conhecido como sistema odométrico, em detrimento da facilidade de implementação, produz erros elevados em função da distância total percorrida.

Um mecanismo de medição de posição bastante confiável são as câmeras, que, com a redução do seu custo, se tornaram muito utilizadas para a correção dos erros

produzidos pelo sistema odométrico ou são fundamentais para controle diretamente no plano da imagem.

Em geral, os atuadores em sistemas robóticos são dos tipos hidráulico, pneumático e elétrico, sendo em robótica móvel predominantemente utilizados os elétricos. Os robôs móveis são dotados de motores que acionam rodas, com as características da aplicação determinando, em primeira instância, as especificações requeridas. Dependendo da aplicação a que se destina o sistema robótico escolhe-se o tipo de motor apropriado, sendo comum a utilização de motores *DC* e servo motores. Tais atuadores são facilmente controláveis por estratégias de controle clássico, cujos algoritmos são de baixa complexidade computacional, sendo portanto facilmente embarcáveis.

Os modelos para o controle destes dispositivos são triviais, sendo que maiores detalhamentos podem ser obtidos em (Ribeiro *et al.* 2011a).

2.3 Representação de Posições e Orientações

Independentemente do tipo de robô móvel a ser utilizado, necessita-se de uma descrição sucinta das relações entre diversos sistemas de coordenadas, sendo comum a fixação de sistemas locais no seu centro de massa e de um sistema global que constitui um referencial inercial conhecido como sistema de referência do mundo. Com tal representação, objetiva-se descrever as posições e orientações do robô através de uma abordagem geométrica que simplifica bastante a tarefa de reproduzir as características de todos os movimentos possíveis.

Em geral, a orientação do robô é fornecida por uma matriz R_n^m conhecida como matriz de rotação que fornece as coordenadas de pontos do sistema n em relação ao sistema m . A Figura 2.3 ilustra os sistemas de coordenadas e as relações entre estes.

A matriz R_n^m é dada por:

$$R_n^m = [x_n^m | y_n^m], \quad (2.1)$$

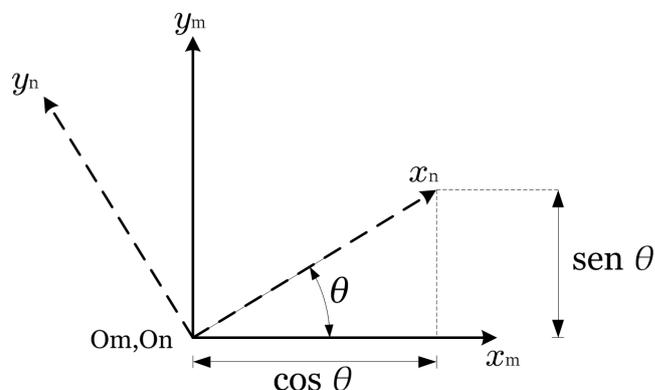


Figura 2.3: Relação entre dois sistemas de coordenadas rotacionados.

sendo x_n^m e y_n^m as coordenadas dos vetores unitários x_n e y_n de $o_n x_n y_n$ com relação ao sistema $o_m x_m y_m$. Assim, tem-se que:

$$x_n^m = \begin{bmatrix} \cos \theta \\ \text{sen } \theta \end{bmatrix}, \quad y_n^m = \begin{bmatrix} -\text{sen } \theta \\ \cos \theta \end{bmatrix}.$$

Substituindo tais expressões em (2.1), obtém-se:

$$R_n^m = \begin{bmatrix} \cos \theta & -\text{sen } \theta \\ \text{sen } \theta & \cos \theta \end{bmatrix}. \quad (2.2)$$

Um ponto qualquer p do sistema n é representado em m pela seguinte expressão:

$$p^m = R_n^m p^n. \quad (2.3)$$

Deste modo, de maneira recursiva, pode-se obter a orientação dos pontos no sistema n com relação ao sistema de referência inercial através de aplicações sucessivas da matriz de rotação R_n^m .

As matrizes de rotação pertencem a um grupo ortogonal especial SO_n e possuem diversas propriedades importantes, as quais podem ser verificadas em (Spong *et al.* 2006, Capítulo 2).

É comum em robótica a utilização do termo pose (postura) para discriminar o conjunto de pontos que determinam a posição e orientação do robô com relação a um dado sistema de coordenadas. Assim, considerando m como sendo indicativo de pontos relativos ao sistema do mundo, define-se a postura do robô como sendo:

$$\xi_m = \begin{bmatrix} x_m \\ y_m \\ \theta_m \end{bmatrix}, \quad (2.4)$$

sendo x_m e y_m posição do centro de massa do robô, e θ_m a orientação, conforme descrito anteriormente.

Os robôs móveis terrestres podem ser considerados como corpos rígidos sobre rodas operando sobre um plano horizontal. A dimensão total deste robô é três sendo esta definida pela postura, onde verificam-se movimentos de translação no plano xy e rotações ao longo do eixo z .

2.4 Modelos Utilizados

Fundamentalmente, os modelos dos robôs móveis classificam-se em cinemáticos e dinâmicos, sendo ambos, numa primeira aproximação, obtidos através de análise fenomenológica.

O objetivo da análise da cinemática dos robôs é obter uma descrição dos seus movimentos sem preocupar-se com as suas causas. Trata-se, portanto, da forma

mais simples de analisar o comportamento de sistemas mecânicos, porém, não menos importante.

Para ambos os robôs terrestres utilizados, as seguintes definições são necessárias:

Definição 2.1. *Cada roda possui apenas um ponto de contato com o solo.*

Definição 2.2. *Ambas as rodas rolam livres de deslizamento na superfície.*

Como as estratégias de controle desenvolvidas neste trabalho possuem como requisito baixos índices de complexidade computacional, são utilizados apenas os modelos cinemáticos direto e inverso.

Em situações em que descrições mais precisas são necessárias, massas e inércias devem ser consideradas, obtendo-se modelos que caracterizam os movimentos dos robôs em termos de forças e torques. Para tal fim, elementos da mecânica Newtoniana ou Lagrangeana podem ser utilizados, fornecendo resultados similares.

2.4.1 Robôs móveis omnidirecionais

Robôs móveis omnidirecionais se caracterizam pela sua capacidade de se movimentar em qualquer direção sem a necessidade de reorientação. Genericamente, essa facilidade de se locomover no plano de navegação define a manobrabilidade dos robôs sendo para os omnidirecionais uma vantagem comparativa com relação a outras configurações tais como diferenciais ou *Ackerman*, sendo, portanto justificada a sua crescente utilização.

A manobrabilidade vai depender do grau de liberdade das rodas e da disposição das mesmas sobre o chassi. Caso se utilizem rodas omnidirecionais, será possível construir robôs com manobrabilidade máxima. A Figura 2.4(a) ilustra uma roda deste tipo.

Uma construção deste tipo de robô com três rodas defasadas no espaço em 120° uma das outras será estável sempre que o seu centro de massa estiver situado dentro do triângulo formado por elas. Robôs omnidirecionais com mais de três rodas necessitam de mecanismos de suspensão adicionais.

Outra característica importante desse tipo de robô é a sua elevada mobilidade, ou seja, sua grande habilidade de mover-se diretamente no ambiente. Isto é justificado pela ausência de restrições relativas à movimentação lateral. A Figura 2.4(b) ilustra uma base omnidirecional com três rodas que apresenta as características supracitadas.



(a) Rodas omnidirecionais.
fonte: site da Vex Robotics¹.



(b) Base com três rodas.

Figura 2.4: Componentes de um robô móvel omnidirecional.

Cinemática dos robôs móveis omnidirecionais

A disposição dos diversos vetores de velocidade no sistema de coordenadas do robô $o_c x_c y_c$ é ilustrada na Figura 2.5. Tal figura representa o diagrama do corpo livre de um robô móvel omnidirecional.

A análise cinemática direta fornece uma descrição do movimento do corpo do robô em função das variáveis das rodas enquanto que com a análise cinemática inversa obtém-se a relação contrária, ou seja, a descrição dos movimentos das rodas em função das variáveis do corpo do robô.

Obtém-se a relação entre as velocidades envolvidas através da expressão a seguir:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R_{z,\theta} \begin{bmatrix} v \\ v_n \\ \omega \end{bmatrix}, \quad (2.5)$$

sendo $R_{z,\theta}$ dada por (2.2).

Analisando-se as relações entre as velocidades das rodas e as velocidades do centro de massa, obtém-se:

$$v_{m1} = -v_n + \omega L, \quad (2.6)$$

$$v_{m2} = v \cos \gamma + v_n \sin \gamma + \omega L, \quad (2.7)$$

$$v_{m3} = -v \cos \gamma + v_n \sin \gamma + \omega L. \quad (2.8)$$

Matricialmente, tem-se:

¹Disponível em: <<http://www.vexrobotics.com/276-2165.html>> Acesso em mar. 2017

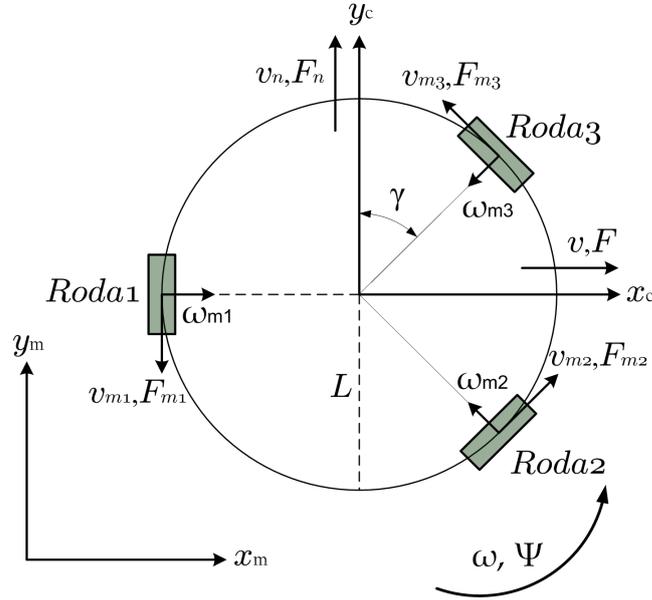


Figura 2.5: Representação dos sistemas de coordenadas, velocidades e forças.

$$\begin{bmatrix} v_{m1} \\ v_{m2} \\ v_{m3} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -1 & L \\ \cos \gamma & \sin \gamma & L \\ -\cos \gamma & \sin \gamma & L \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} v \\ v_n \\ \omega \end{bmatrix}. \quad (2.9)$$

Esta expressão é necessária para a descrição da cinemática inversa dos robôs móveis omnidirecionais.

Para a obtenção do modelo cinemático direto, multiplica-se ambos os lados de (2.9), à esquerda, por B^{-1} , de onde obtém-se:

$$\begin{bmatrix} v \\ v_n \\ \omega \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -1 & L \\ \cos \gamma & \sin \gamma & L \\ -\cos \gamma & \sin \gamma & L \end{bmatrix}^{-1}}_{\mathbf{G}} \begin{bmatrix} v_{m1} \\ v_{m2} \\ v_{m3} \end{bmatrix}. \quad (2.10)$$

A relação entre as velocidades lineares e as velocidades angulares das rodas é obtida a partir da seguinte expressão:

$$\begin{bmatrix} v_{m1} \\ v_{m2} \\ v_{m3} \end{bmatrix} = \mathbf{RL}^{-1} \begin{bmatrix} \omega_{m1} \\ \omega_{m2} \\ \omega_{m3} \end{bmatrix}, \quad (2.11)$$

sendo:

$$\mathbf{L} = \begin{bmatrix} l_1 & 0 & 0 \\ 0 & l_2 & 0 \\ 0 & 0 & l_3 \end{bmatrix} \text{ a matriz das reduções dos motores;}$$

$$\mathbf{R} = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \text{ a matriz dos raios das rodas.}$$

Finalmente:

$$\begin{bmatrix} v \\ v_n \\ \omega \end{bmatrix} = \mathbf{RL}^{-1} \begin{bmatrix} 0 & -1 & L \\ \cos \gamma & \sin \gamma & L \\ -\cos \gamma & \sin \gamma & L \end{bmatrix}^{-1} \begin{bmatrix} \omega_{m1} \\ \omega_{m2} \\ \omega_{m3} \end{bmatrix}. \quad (2.12)$$

Através de (2.5) obtém-se uma justificativa da importância de uma boa análise da cadeia cinemática do sistema robótico, pois, sabe-se que a estimação de posturas de robôs através do sistema odométrico baseia-se na medição das velocidades das rodas, cálculo das velocidades do centro de massa através de (2.10) e posterior integração numérica, de modo que é necessária uma representação a mais fiel quanto possível dos seus movimentos.

A Figura 2.6 ilustra o robô omnidirecional utilizado neste trabalho.

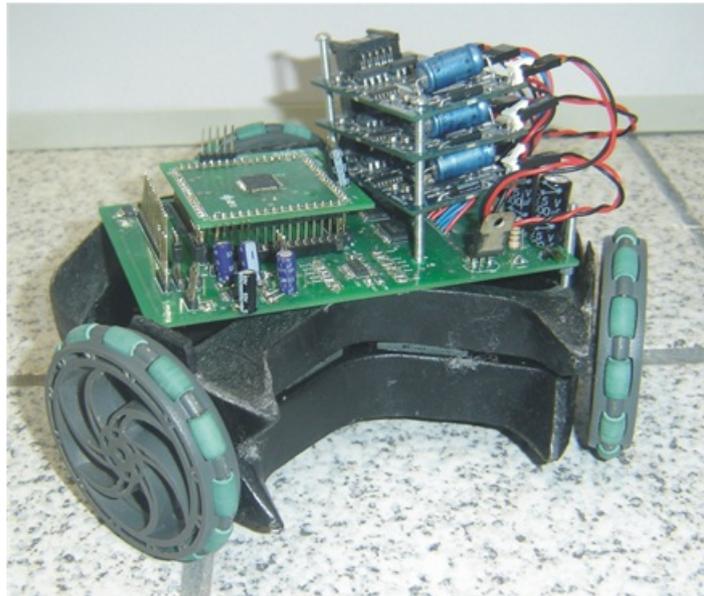


Figura 2.6: Robô móvel omnidirecional.

2.4.2 Robôs móveis diferenciais

Robôs móveis diferenciais com rodas são classificados como robôs não holonômicos e são bastante utilizados em diversas situações, sempre que haja necessidade de um mecanismo cuja descrição dos movimentos seja simplificada. Tal simplificação se deve ao fato de que é preciso definir apenas dois comandos para estabelecer a direção de movimento, especificamente velocidade e sentido de giro de duas rodas.

Sistemas não holonômicos em geral se caracterizam pelo fato de que suas equações diferenciais não são integráveis da origem até a posição final. Especificamente para o caso de robôs diferenciais, isto significa que, mesmo num cenário ideal, a distância percorrida pelas rodas não é suficiente para determinar a posição final do robô, sendo necessário um conhecimento prévio de como o movimento foi realizado. A Figura 2.7 exemplifica esta situação. Observa-se que, mesmo sendo as distâncias iguais ($d_{1_E} = d_{2_E}$, $d_{1_D} = d_{2_D}$ e $d_1 = d_2$), as coordenadas finais são diferentes ($x_1 \neq x_2$ e $y_1 \neq y_2$).

Cinemática dos robôs móveis diferenciais

A Figura 2.8(a) apresenta os sistemas de coordenadas utilizados na modelagem cinemática do robô diferencial. Tem-se O_m a origem do sistema de coordenadas do mundo e O_c a origem do sistema de coordenadas do robô. Considerando que a postura do robô em O_m é completamente especificada por $[x \ y \ \theta]^T$, tem-se o seguinte modelo cinemático:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (2.13)$$

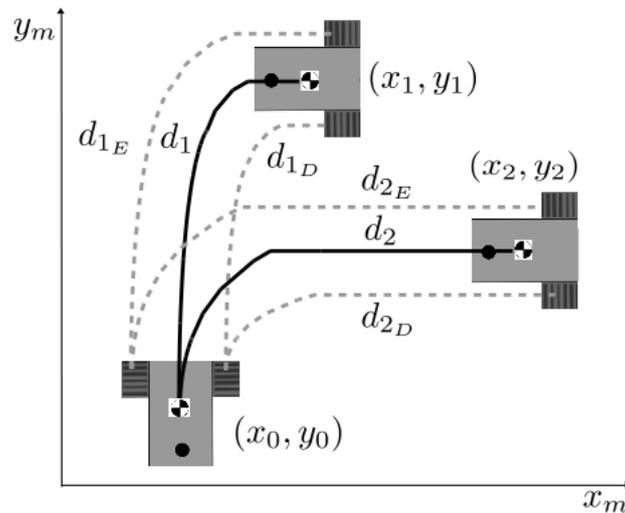


Figura 2.7: Exemplo de movimentação de robôs diferenciais.

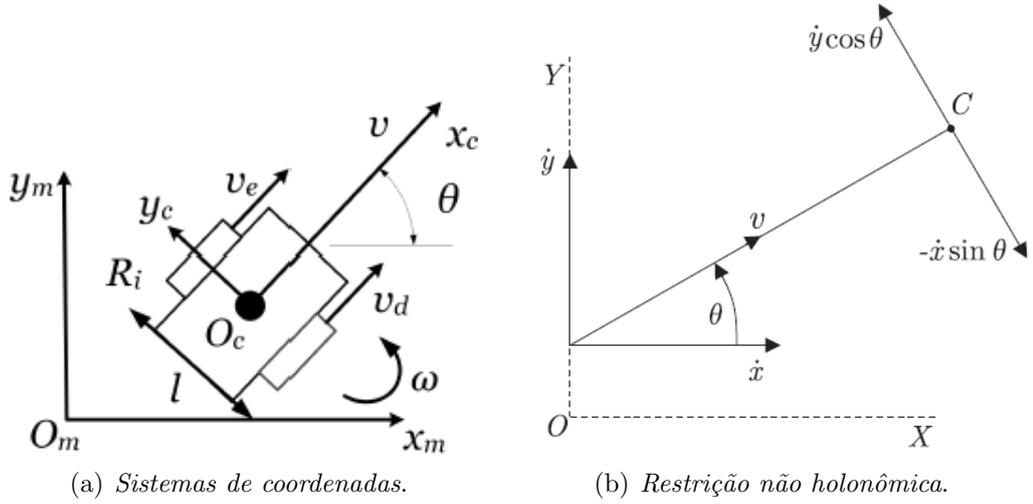


Figura 2.8: Modelagem cinemática do robô diferencial.

Sendo as entradas deste modelo a velocidade linear v na direção de x_c e a velocidade angular ω em torno do eixo normal à página.

Desde que a Definição 2.2 seja satisfeita, a velocidade na direção y_c é nula, ou seja $\dot{y}_c = 0$. Da análise da Figura 2.8(b) é possível concluir que:

$$\begin{bmatrix} -\sin \theta & \cos \theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0. \quad (2.14)$$

Sendo deste modo explicitada a restrição cinemática não holonômica do modelo. Maiores detalhes sobre restrições cinemáticas podem ser encontrados em (Barraquand e Latombe 1989).

A velocidade linear v e a velocidade angular ω relacionam-se com as velocidades lineares das rodas direita v_D e esquerda v_E do seguinte modo:

$$v = \frac{v_D + v_E}{2}; \quad (2.15)$$

$$\omega = \frac{v_D - v_E}{l}. \quad (2.16)$$

A relação entre as velocidades lineares e as velocidades angulares das rodas é obtida a partir da seguinte expressão:

$$\begin{bmatrix} v_D \\ v_E \end{bmatrix} = \mathbf{RL}^{-1} \begin{bmatrix} \omega_D \\ \omega_E \end{bmatrix}, \quad (2.17)$$

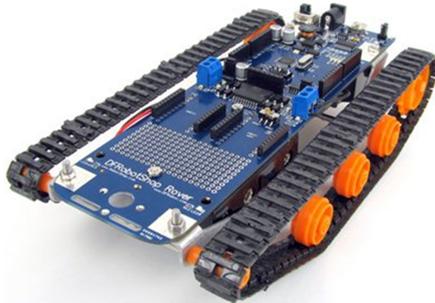
sendo:

$$\mathbf{L} = \begin{bmatrix} l_D & 0 \\ 0 & l_E \end{bmatrix} \text{ a matriz das reduções dos motores;}$$

$$\mathbf{R} = \begin{bmatrix} r_D & 0 \\ 0 & r_E \end{bmatrix} \text{ a matriz dos raios das rodas.}$$

Dos modelos obtidos é possível confirmar que, a despeito do surgimento das restrições não holonômicas, a caracterização dos movimentos dos robôs diferenciais é bastante simples.

A Figura 2.9 ilustra os robôs móveis diferenciais comerciais utilizados neste trabalho.



(a) *Robotshop Rover*.
fonte: site da Robotshop¹.



(b) *Lego Mindstorms EV3*.
fonte: site da Lego².

Figura 2.9: Robôs diferenciais utilizados.

2.4.3 Quadricópteros

Recentemente, grandes esforços estão sendo dedicados ao desenvolvimento de estratégias para controle de quadricópteros. Estes tipos de veículos, caracterizados internacionalmente como *multi-rotor-craft*, possuem vantagens significativas em relação aos helicópteros convencionais, tais como maior manobrabilidade e precisão nos movimentos. Adicionalmente, a disponibilidade de plataformas de pesquisa de baixo custo possibilita o seu uso em várias aplicações da robótica, como pode ser visto em (Sa *et al.* 2013) e (Kim *et al.* 2013), se tornando uma ferramenta importante tanto economicamente quanto tecnicamente.

Embora a modelagem dos quadricópteros seja mais complexa do que a dos robôs terrestres, a potencial desvantagem que surge quando se analisam efeitos aerodinâmicos são compensadas com os baixos níveis de atrito experimentados por este tipo

¹Disponível em: <<http://www.robotshop.com/en/dfrobotshop-rover-tracked-robot-basic-kit.html>> Acesso em jan. 2017

²Disponível em: <<https://www.lego.com/en-us/mindstorms/products/mindstorms-ev3-31313>> Acesso em jan. 2017

de robô, facilitando tarefas de pouso e decolagem. Além disso, a aplicabilidade de um robô com mobilidade em três dimensões é muito mais extensa.

Cinemática dos Quadricópteros

Um quadricóptero, como o utilizado neste trabalho, consiste de quatro rotores conectados a um sistema de coordenadas rígido, conforme ilustrado na Figura 2.10. Trata-se de um sistema mecânico sub atuado com seis graus de liberdade e somente quatro entradas de controle.

Denota-se o sistema de coordenadas do mundo (inercial) por $\{W\}$ e o sistema de coordenadas do corpo fixado no centro de massa do quadricóptero por $\{B\}$.

O vetor $\xi = [x \ y \ z]^T$ representa a posição do centro de massa do quadricóptero expresso em termos do sistema inercial $\{W\}$, e a orientação do veículo é dada por uma matriz de rotação $R_B^W = R$, onde $R \in SO(3)$ é uma matriz de rotação ortonormal (ver (Mahony *et al.* 2012); (Corke 2011), Capítulos 2-4). O vetor $\Phi = [\psi \ \phi \ \theta]^T$ representa os ângulos de Euler fixos nos eixos Z-X-Y (ângulo de guinada, ψ ; ângulo de rolagem, ϕ ; e ângulo de arfagem, θ) usados para descrever a orientação do veículo com respeito a $\{W\}$. É interessante notar que o vetor de velocidade angular ω em $\{B\}$ não é $\dot{\Phi}$, já que o eixo de rotação não estará necessariamente localizado na direção da derivada temporal dos ângulos de Euler. Para obter o vetor de velocidade angular resultante utiliza-se a seguinte relação:

$$\omega = \begin{bmatrix} 1 & 0 & -\text{sen } \theta \\ 0 & \cos \phi & \cos \theta \text{sen } \phi \\ 0 & -\text{sen } \phi & \cos \theta \cos \phi \end{bmatrix} \dot{\Phi}. \quad (2.18)$$

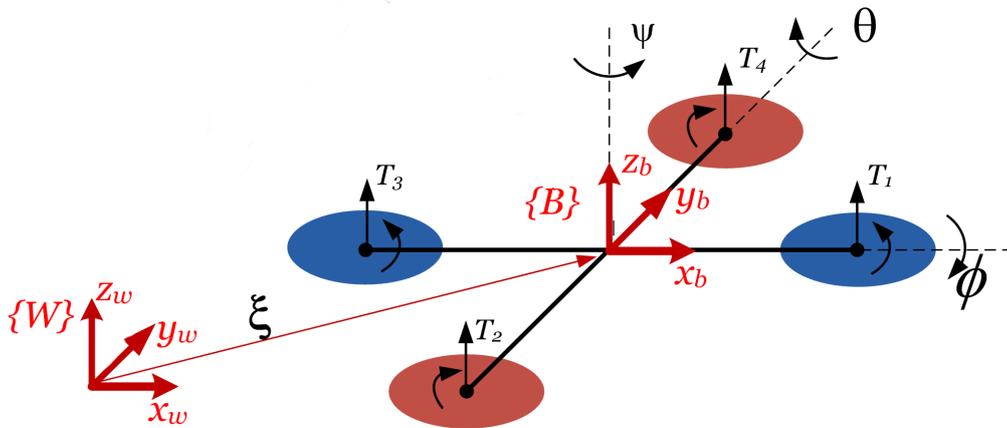


Figura 2.10: Sistemas de coordenadas para a modelagem do quadricóptero.

O modelo cinemático que relaciona as velocidades em $\{B\}$ a $\{W\}$ é dado da seguinte forma:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t)); \\ \mathbf{x}(0) &= \mathbf{x}_0; \\ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} &= R \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix},\end{aligned}\tag{2.19}$$

sendo R a matriz de rotação ortogonal, que converte do sistema de coordenadas do quadricóptero (x_b, y_b, z_b) para o sistema de coordenadas do mundo (x_w, y_w, z_w) e vice-versa, dada por:

$$R = \begin{bmatrix} c_\phi c_\psi - c_\theta s_\phi s_\psi & -c_\psi s_\phi - c_\phi c_\theta s_\psi & s_\theta s_\psi \\ c_\theta c_\psi s_\phi + c_\phi s_\psi & c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\psi s_\theta \\ s_\phi s_\theta & c_\phi s_\theta & c_\theta \end{bmatrix}.\tag{2.20}$$

sendo $c_\bullet = \cos \bullet$ e $s_\bullet = \sin \bullet$

Neste trabalho, as estratégias de controle desenvolvidas para os quadricópteros serão avaliadas através de simulações. Tais robôs são bastantes conhecidos como *drones* e existe uma grande variedade de modelos disponíveis no mercado a Figura 2.11 ilustra alguns destes.



Figura 2.11: Alguns quadricópteros comerciais.

¹Disponível em: <<https://www.parrot.com/fr/drones/parrot-ardrone-20-elite-édition>> Acesso em fev. 2017

²Disponível em: <<http://www.dji.com/phantom-4-pro>> Acesso em fev. 2017

2.5 Plataforma Experimental

É de fundamental importância para o projeto de sistemas de controle a utilização de softwares confiáveis que possibilitem ao desenvolvedor preocupar-se apenas com o atendimento das especificações de desempenho. Atualmente, uma grande variedade de ambientes de desenvolvimento (*IDE's*) que atendem a estes requerimentos estão disponíveis.

Neste trabalho, os algoritmos foram implementados no ambiente *Lazarus*. Trata-se de um software *IDE* compatível com Linux, Windows, *OS/2*, *Mac OS tradicional*, *Mac OS X*, *ARM*, *BSD*, *BeOS*, *DOS* e outros.

São vários os motivos que nortearam a escolha do *Lazarus* como o software *IDE* a ser utilizado no desenvolvimento das atividades. Pode-se citar, por exemplo, o fato deste ser um programa localmente compilado, o que possibilita a geração de executáveis compatíveis com diversas plataformas, a linguagem operacional é o *Object Pascal* proporcionando facilidades na programação, é licenciado em *GPL*, ou seja, os softwares desenvolvidos com ele podem ser distribuídos sob qualquer licença, aumentando assim os atrativos comerciais, etc. Além disso, existe compatibilidade com códigos desenvolvidos em outras linguagens. Neste trabalho, por exemplo, o otimizador foi desenvolvido em linguagem *C* e a comunicação entre as aplicações em *Lazarus* é feita através de objetos estáticos.

Para desenvolvimento das soluções deste trabalho, devem ser utilizados softwares capazes de interagir de maneira satisfatória com o sistema operacional, para que sejam atendidas as necessidades de agendamento de tarefas em tempo real e otimização dos tempos de processamento. Assim, utiliza-se como plataforma computacional o *Lazarus* funcionando em ambiente *Linux*, sendo deste modo facilitada a portabilidade dos códigos fonte desenvolvidos dentre outras necessidades inerentes a este trabalho.

2.5.1 O software *Decision*

O programa utilizado para a implementação do mapa de decisão é o *Decision*. Tal programa foi ajustado e complementado a partir de uma versão desenvolvida em (Conceição 2007) e é responsável pela definição, em todas as instâncias, dos comportamentos dos robôs.

Além da interface *GUI* já desenvolvida, o *Decision* disponibiliza gráficos de todas as grandezas inerentes ao sistema robótico e apresenta a movimentação dos robôs móveis em um campo de futebol, tornando a sua interface ainda mais amigável e ilustrativa. A Figura 2.12 mostra algumas telas do *Decision* e a Figura 2.13 ilustra o seu fluxograma.

O *Decision* é completamente estruturado para operação via eventos, de modo que tem-se grande otimização do esforço computacional.

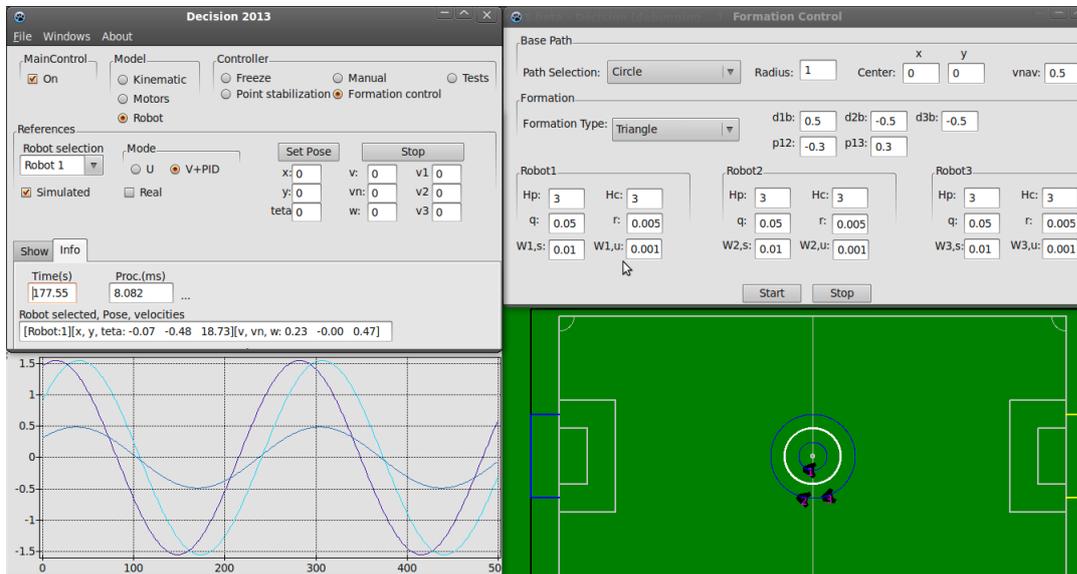


Figura 2.12: Algumas telas do software *Decision*.

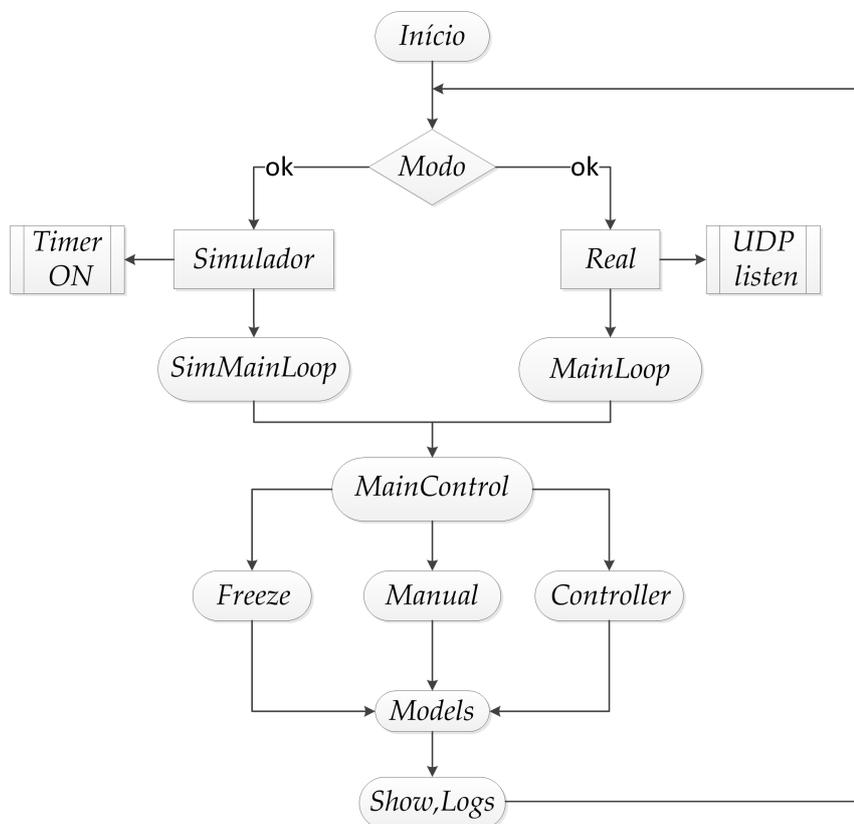


Figura 2.13: Fluxograma do software *Decision*.

Uma das grandes vantagens do *Decision* é que o usuário pode selecionar o modo operacional como sendo de simulação ou experimental. Através do modo simulação têm-se condições de avaliar as estratégias de controle através de modelos teóricos para caracterização prévia de desempenho. Esta característica é de grande importância, pois evita-se que sejam realizados testes agressivos ao hardware que poderiam causar-lhe danos.

Tanto o *loop* de simulação (*SimMainLoop*) quanto o experimental (*Mainloop*) são compostos de um controle principal (*MainControl*) que é responsável pela aquisição de parâmetros de entrada e condicionamento do sistema para a definição de modos de atuação, tais como parado (*freeze*), controle manual dos motores (*Manual*, Controladores de baixo nível) e controladores de postura (*Controller*, Controladores de alto nível). Finalmente, tem-se as tarefas de modelagem (*Models*) e de monitoramento (*Show, Logs*).

Além das janelas ilustradas na Figura 2.13, o *Decision* permite ao usuário modificar parâmetros de modelos e de controladores, alternar os modos de atuação e operacionais, dentre outras características. Maiores detalhes sobre o *Decision* podem ser encontrados em (Ribeiro *et al.* 2011a).

2.5.2 Arquitetura Descentralizada

Os robôs utilizados e seus sistemas microprocessados não possuem capacidade computacional para que sejam embarcados os controladores *NMPC* locais. Nestes robôs, os únicos recursos existentes são relativos à instrumentação básica, ou seja, sensores, atuadores e controladores de baixo nível.

Os controladores de baixo nível são definidos como aqueles responsáveis pelo controle de velocidade das rodas de maneira independente. Tratam-se de controladores *PID* discretos implementados através da modalidade conhecida como algoritmo velocidade. Maiores detalhes sobre os aspectos instrumentais destes robôs são obtidos em (Ribeiro *et al.* 2011a).

Desta forma, para o desenvolvimento dos controladores de formação descentralizados, consideram-se os computadores do tipo *Desktop* (*PC*), responsáveis pela implementação do ambiente de decisão/supervisão, como parte do robô. São utilizados os seguintes (*PCs*):

PC1: Intel® Core™ i5 CPU M 430 @ 2,27 GHz X 4; 5,5 GB de RAM; 64 bits;

PC2 e PC3: Intel® Core™ i7 CPU 4770 @ 3,90 GHz X 4; 8 GB de RAM; 64 bits.

Existe contínua comunicação do robô para estes *PCs* através de pacotes que, além de enviar os dados instrumentais de baixo nível, disparam os eventos a cada período de amostragem. Tais eventos são utilizados para o cálculo das ações de controle.

Comunicação entre robôs e *PCs*

Diversos fatores devem ser analisados antes da definição de uma estratégia comunicação adequada. Aspectos como taxas de comunicação, alcance, latência, eficácia e consumo energético são fundamentais.

Devem ser utilizadas estratégias que disponibilizem uma infraestrutura para comunicação em rede, objetivando a avaliação dos controladores através de diversas topologias. De maneira geral, os aspectos detalhados desta parte do controle automático conhecida como *Networked Control Systems*, podem ser encontrados em (Walsh *et al.* 2002).

Como os robôs móveis estão em contínuo movimento por um espaço de trabalho geralmente desconhecido, estratégias de comunicação sem fio devem ser escolhidas. Assim, todos os elementos norteadores da escolha da estratégia se tornam ainda mais críticos.

Os fatores determinantes da latência em sistemas de comunicação sem fio são, em geral, os métodos de acesso implementados e os protocolos para checagem e correção de erros. Tais protocolos irão definir em primeira instância a eficácia na entrega dos dados.

Consumo energético é sempre uma especificação relevante, já que há uma tendência de miniaturização dos dispositivos, especialmente no caso da robótica móvel, onde é requerida alta densidade de potência por agente com minimização de massa, já que as fontes são carregamentos obrigatórios.

Diante deste cenário, as estratégias de comunicação entre robô e *PC*, dependem do tipo de robô utilizado.

Robôs Diferenciais

Considerando-se as características dos robôs utilizados, a comunicação entre robô diferencial e computador pessoal (*PC*) é feita através do padrão *zigbee* (ZigBee 2011). Tal padrão opera em complementaridade com o padrão *IEEE 802.15.4* (Zheng e Lee 2004) a partir da utilização das camadas física e de enlace.

O padrão *zigbee* implementa redes de baixo consumo energético e com topologias diversificadas, sendo basicamente dois dispositivos a serem definidos: os dispositivos de função completa (*FFD*) e os dispositivos de função reduzida (*RFD*) que implementam coordenadores e dispositivos finais, respectivamente, numa rede *zigbee*.

Os modos de operação de uma rede *zigbee* são denominados “sinalizado” e “não-sinalizado”. No modo “sinalizado” os coordenadores enviam sinais periodicamente para confirmar a presença de outros elementos na rede, sendo com este modo permitido operar em modo *sleep*, já que estes outros elementos apenas necessitam serem ativados no momento da sinalização. No modo “não-sinalizado” os elementos permanecem conectados todo o tempo, incrementando os requisitos energéticos necessários.

Neste trabalho são utilizados os módulos *xbee* da *Maxstream* (MaxStream 2006). Tais módulos oferecem diversas vantagens, tais como facilidade de configuração e

pequeno volume. A Figura 2.14 ilustra resumidamente a arquitetura da pilha de protocolos *zigbee* juntamente com um módulo *xbee*.

Os módulos *xbee* se comunicam a uma taxa fixa de $250Kbps$ no *link* de *RF* e outra configurável, de valor superior limitado em $115,2Kbps$, com as aplicações. Esta diferença se deve à implementação de protocolos de checagem de erros dentre outros fatores. Porém, constata-se, com as implementações práticas de uma plataforma ponto a ponto utilizando os módulos *xbee*, que a comunicação se torna muito imprecisa quando estes operam em sua taxa máxima.

Isto se deve à imprecisão no *clock* destes módulos, fazendo com que a taxa real seja um pouco menor, sendo assim incompatível com o padrão definido no *host* que implementa a aplicação. À medida em que se aumenta esta taxa a imprecisão se torna ainda maior, e, como nesta aplicação não há a necessidade de elevadas taxas, define-se esta como sendo $57,6Kbps$.

Robôs Omnidirecionais

A dinâmica mais rápida do robô omnidirecional utilizado inviabiliza a utilização do padrão *zigbee*. Desta forma, a comunicação entre estes robôs e os *PCs* é feita através do padrão *wi-fi*.

Assim como no padrão *zigbee*, tais módulos operam nas bandas do espectro destinadas à *ISM* (do inglês, *Industrial, Scientific and Medical*), porém, com modulações e protocolos para checagem e correção de erros que permitem taxas de dados elevadas que variam desde $1Mbps$ até $600Mbps$, valores mais do que suficientes à aplicação em questão.

As potências de transmissão permitem alcances superiores aos verificados com o padrão *zigbee*, o que torna o consumo energético dos equipamentos para a implementação desta estratégia, em geral, superiores. As variações no padrão *IEEE 802.11* determinam as especificações de desempenho.

Devido ao reduzido valor do tempo de bit, além da versatilidade na escolha dos protocolos da camada de transporte, os níveis latência para este tipo de estratégia são bem menores do que os verificados com outras, tais como *bluetooth* (Albert *et al.* 2010) ou o próprio *zigbee*.

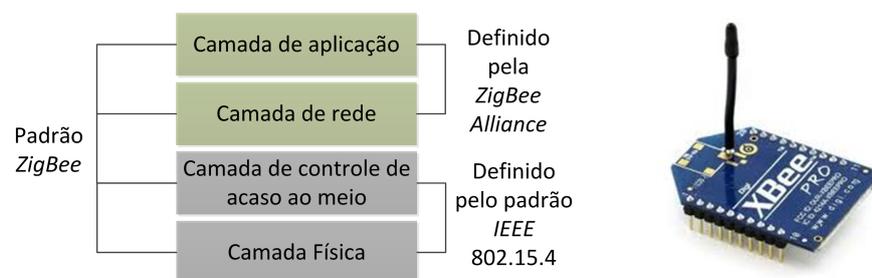


Figura 2.14: Arquitetura resumida da pilha de protocolos *zigbee* e um módulo *xbee* da *maxstream*.

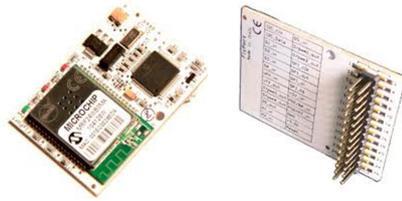


Figura 2.15: Módulos *Flyport* utilizados para a comunicação entre robô omnidirecional e *PC*.

Além destes fatores, a grande disponibilidade de módulos comerciais para a implementação dessa estratégia, aspectos relativos à integridade e segurança dos dados comunicados e a versatilidade na configuração das redes, tornam a sua utilização bastante atrativa.

São dois tipos básicos de configuração das redes sem fio: tipo infra-estrutura e *ad-hoc*. Na infra-estrutura todos os dispositivos se comunicam entre si, geralmente por um centralizador. Já no tipo *ad-hoc*, tem-se uma definição *a priori* de quais dispositivos devem se comunicar. Do mesmo modo que as redes baseadas no padrão ethernet, para as redes *wi-fi* devem ser definidos os endereços *IP*, máscara sub-rede e gateway padrão para cada dispositivo.

Neste trabalho, utilizam-se módulos *wi-fi* denominados *flyport* (openPicus 2012) da *OpenPicus*. Tais módulos são uma combinação de microcontrolador e transceptor *wi-fi*, de código fonte aberto, para soluções de conectividade sem fio através dos padrões *IEEE 802.11 b/g/n*.

Os módulos *Flyport* são de baixo consumo energético e possuem *IDE* e *API* próprios, o que reduz significativamente o tempo desenvolvimento das aplicações, porém, por limitação de hardware, a taxa de comunicação máxima é definida em *2Mbps*. A Figura 2.14 ilustra algumas imagens destes módulos.

Comunicação entre *PCs*

Neste caso, o protocolo *UDP* foi escolhido, principalmente pelas altas taxas de dados verificadas, já que trata-se de serviço sem entrega garantida e não orientado à conexão (Tanenbaum 2002), ou seja, sem nenhum tipo de *Handshake* na comunicação, fatores que são desnecessários à aplicação em questão

As Figuras 2.16 e 2.17 ilustram os detalhes das arquiteturas utilizadas, para o robô omnidirecional e diferencial, respectivamente. Tal arquitetura apresenta aspectos da topologia de rede utilizada, os endereços *IP* e as portas dos *sockets UDP* de cada um destes elementos. Os relógios ilustrados indicam que a base de tempo real é implementada em cada robô.

Em (Ribeiro 2013) são fornecidos maiores detalhes sobre estas arquiteturas.

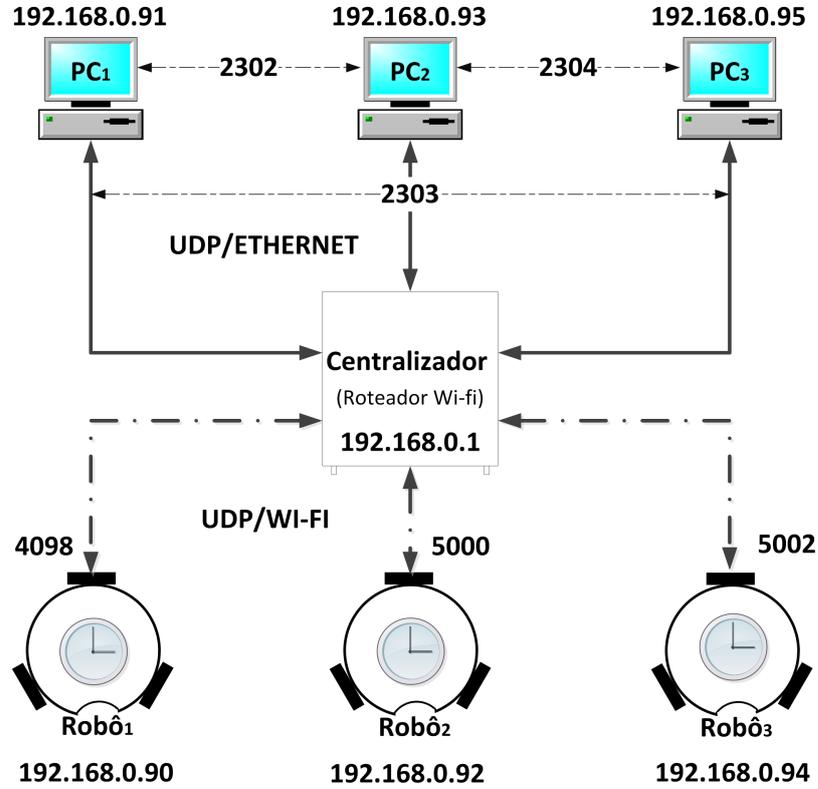


Figura 2.16: Arquitetura de comunicação sistema multi robôs omnidirecionais.

2.5.3 Sistema Visual

Neste trabalho, o sistema visual consiste em *webcams* comerciais instaladas nos robôs diferenciais das figuras 2.9(a) e 2.9(b).

Para o caso do *RobotShop Rover* uma *webcam* comercial e de baixo custo foi instalada com a seguinte postura de offset (relativa ao eixo traseiro): $x_c = 0,16\text{m}$; $y_c = 0\text{m}$; $z_c = 0,18\text{m}$; $\theta_{c_y} = \frac{\pi}{4}$ rad.

Por ser um equipamento de baixo custo e relativamente antigo, informações sobre o sistema óptico não foram obtidas, sendo a calibração do sistema visual feita de maneira *ad-hoc*. A cadência visual desta câmera varia entre 2 e 30 fps, dependendo bastante das condições de luminosidade.

No caso do *Lego EV3* utilizou-se uma *webcam HD-3000* da *microsoft*. Esta câmera filma em *HD* com uma resolução de 720p, razão de aspecto 16:9, e dispõe da tecnologia *TrueColor* que corrige automaticamente brilho e cor em diversas condições de iluminação. Neste caso, a cadência visual varia entre 15 e 30 fps, variando menos em função das condições de luminosidade. Esta câmera foi instalada com a seguinte postura de offset (relativa ao eixo dianteiro): $x_c = 0\text{m}$; $y_c = 0,05\text{m}$; $z_c = 0,2\text{m}$; $\theta_{c_y} = \frac{\pi}{4}$ rad.

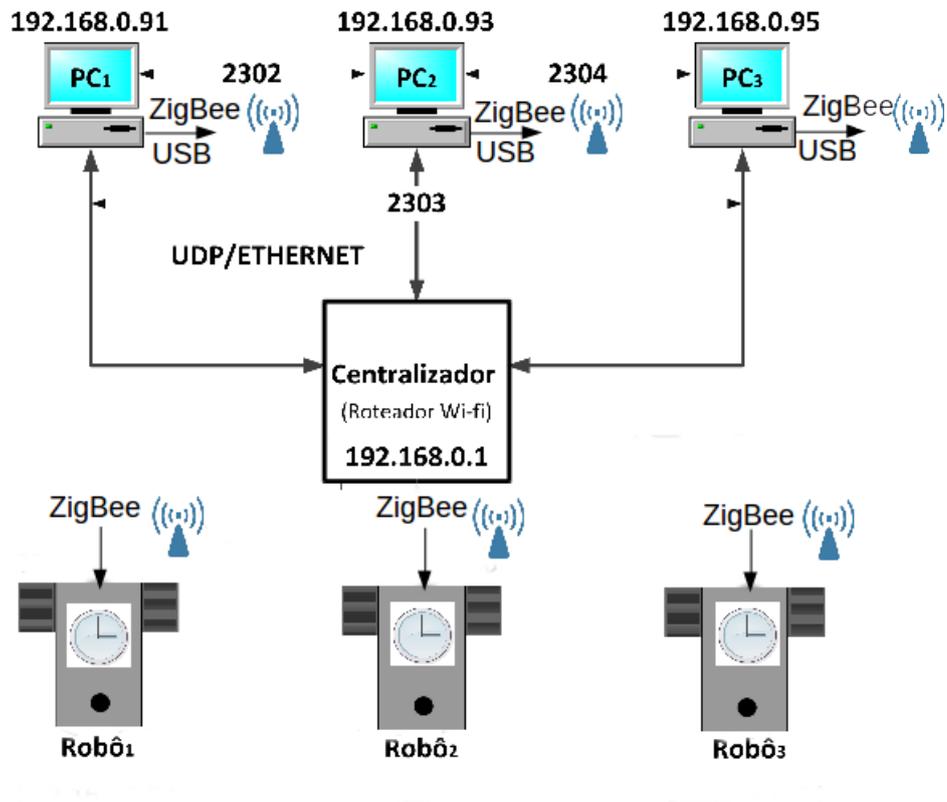


Figura 2.17: Arquitetura de comunicação sistema multi robôs diferenciais.

Ambas as câmeras possuem distância focal $f = 32mm$ e comunicam-se com o *PC* através de portas *USB*.

O processo de calibração do sistema visual é feito através de ajustes entre os valores medidos para características visuais adquiridas no plano da imagem e aquelas medidas plano de navegação. Tal processo é discutido detalhadamente no Capítulo 4.

A Figura 2.18 mostra as referidas instalações.



(a) Montagem da *webcam* no *RobotShop Rover*.



(b) Montagem da *webcam* no *Lego Ev3*.

Figura 2.18: Instalação das *webcams* nos Robôs diferenciais.

2.6 Conclusões

Este capítulo foi dedicado à apresentação dos robôs móveis e dos sistemas fundamentais utilizados neste trabalho. Aspectos construtivos inerentes a qualquer robô móvel e modelos fundamentais ao desenvolvimentos dos controladores para seguimento de caminho e controle de formação foram apresentados e discutidos.

Independente do tipo de robô, é possível observar que os modelos cinemáticos são relativamente simples, sendo, desta forma, possível atender aos requisitos de baixa complexidade algorítmica. Para os controladores propostos, como será visto nos próximos capítulos, fica evidente que tais modelos são suficientes.

Detalhes importantes da plataforma experimental foram fornecidos. Tal etapa é de fundamental importância para que seja possível compreender a relevância dos experimentos realizados. Os próximos capítulos fornecem a complementação necessária na medida em que as estratégias e os objetivos de controle forem apresentados.

Dentre os requisitos fundamentais para a implementação prática de qualquer estratégia de controle de formação, a arquitetura de comunicação possui grande destaque, já que, para o sucesso das abordagens descentralizadas, estratégias de comunicação eficientes em diversos domínios devem ser escolhidas.

Capítulo 3

Controle Preditivo Baseado em Modelo Não linear

Neste capítulo, os fundamentos básicos de projeto dos controladores preditivos baseados em modelos não-lineares são descritos.

Inicialmente, alguns conceitos teóricos essenciais são apresentados. A importância desta etapa é justificada pela necessidade da síntese de conteúdos intrínsecos à teoria de controle não linear que são aplicáveis no âmbito desta tese.

Sabe-se que a aplicabilidade destes controladores possui limitações nos requisitos de factibilidade, estabilidade e custo computacional, sendo necessária, portanto, uma análise detalhada destas características. Em seguida, são apresentados os esquemas convencionais de predição e os formatos gerais de função objetivo e restrições utilizados.

Finalmente as principais estratégias de otimização não linear necessárias à implementação destes controladores são brevemente discutidas.

3.1 Preliminares teóricas

Em diversas situações, as não-linearidades dos processos devem ser consideradas nos projetos dos controladores para que sejam atendidos critérios específicos de desempenho (Camacho e Alba 2013, Capítulo 9). Plantas com condições operacionais cada vez mais rigorosas devem ser controladas de maneira eficiente, e uma solução que vem sendo bastante utilizada baseia-se nos controladores ótimos. Controladores *NMPC* fazem parte desta família de controladores, diferindo fundamentalmente dos controladores *MPC* pela utilização de modelos não-lineares dos processos tanto nos objetivos quanto nas restrições. (Allgöwer *et al.* 2004)

Os modelos utilizados neste trabalho são representados pela seguinte equação diferencial não linear:

$$\dot{x}(t) = f(u, x, t); \quad x(0) = x_0. \quad (3.1)$$

sendo $x \in \mathbb{R}^n$ e $u \in \mathbb{R}^m$ os vetores de estados e entradas, respectivamente. Considerando o determinismo destes modelos, deve-se estudar previamente se soluções existem e se as mesmas são únicas. Além disso, a estabilidade dos modelos deve ser analisada para que se possa estabelecer critérios para a estabilidade do sistema em malha fechada. Finalmente, por motivações práticas, alguma métrica para a obtenção de soluções deve ser adotada.

3.1.1 Existência e unicidade de soluções

É possível explicitar u como função do tempo em (3.1), obtendo-se o seguinte problema do valor inicial:

$$\dot{x} = f(t, x); \quad x(t_0) = x_0, \quad (3.2)$$

Para existência e unicidade da solução de (3.2) a condição de *Lipschitz* deve ser satisfeita. Tal condição é dada pela seguinte inequação:

$$\|f(t, x) - f(t, y)\| \leq L\|x - y\|, \quad (3.3)$$

para todo (t, x) e (t, y) em alguma vizinhança de (t_0, x_0) .

Uma função que satisfaça (3.3) é dita ser *Lipschitz* em x , e a constante positiva L é chamada de constante de *Lipschitz*. Os teoremas enunciados e provados em (Khalil 2002, Capítulo 3) mostram que (3.2) é:

- Localmente *Lipschitz* em x no intervalo $[a, b] \times D \subset \mathbb{R} \times \mathbb{R}^n$ se cada ponto $x \in D$ possui uma vizinhança D_0 tal que f satisfaz (3.3) no intervalo $[a, b] \times D_0$ com alguma constante de *Lipschitz* L_0 ;
- *Lipschitz* em x no intervalo $[a, b] \times W$ se ele satisfaz (3.3) para todo $t \in [a, b]$ e todos os pontos em W com a mesma constante de *Lipschitz* L ;
- Globalmente *Lipschitz* em x se ela é *Lipschitz* em x no intervalo $[a, b] \times \mathbb{R}^n \subset \mathbb{R} \times \mathbb{R}^n$.

A condição (3.3) é muitas vezes chamada de *continuidade de Lipschitz* por ser uma condição mais abrangente do que a continuidade usual definida em um ponto. Por outro lado a continuidade de *Lipschitz* é menos abrangente do que a diferenciabilidade contínua.

3.1.2 Estabilidade Segundo Lyapunov

Inicialmente, os critérios de estabilidade segundo Lyapunov serão apresentados para sistemas autônomos¹ e em seguida para sistemas não autônomos. As provas dos teoremas não são fornecidas, mas podem ser encontradas em (Khalil 2002, Capítulo 4)

Sistemas Autônomos

Seja o seguinte sistema:

$$\dot{x} = f(x), \quad (3.4)$$

onde $f : D \rightarrow \mathbb{R}^n$ é um mapa localmente *Lipschitz* de um domínio $D \subset \mathbb{R}^n$ em \mathbb{R}^n . É possível supor, sem perda de generalidade, que o sistema possui um ponto de equilíbrio na origem de \mathbb{R}^n , ou seja, $f(0) = 0$, já que um ponto de equilíbrio qualquer pode ser deslocado para origem através de mudança de variáveis apropriada.

A seguinte definição estabelece a estabilidade segundo Lyapunov:

Definição 3.1. *O ponto de equilíbrio $x = 0$ de (3.4) é:*

- *Estável se, para todo $\epsilon > 0$, existe $\delta = \delta(\epsilon) > 0$ tal que:*

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \epsilon, \forall t \geq 0. \quad (3.5)$$

- *Instável, se este ponto não é estável.*
- *Assintoticamente estável, se tal ponto é estável e δ pode ser escolhido tal que:*

$$\|x(0)\| < \delta \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0. \quad (3.6)$$

Textualmente, o ponto $x = 0$ será estável sempre que, para cada valor de ϵ existir um valor de δ , possivelmente dependente de ϵ , tal que uma trajetória iniciando em δ na vizinhança de $x = 0$ nunca irá deixar a vizinhança de ϵ . De maneira geral, é possível examinar a estabilidade através da análise das derivadas de algumas funções em especial, sem ser necessário explicitar a solução de (3.4).

O teorema da estabilidade de Lyapunov fornece esta possibilidade.

Teorema 3.1. *Seja $x = 0$ um ponto de equilíbrio de (3.4) e $D \subset \mathbb{R}^n$ um domínio contendo $x = 0$. Seja $V : D \rightarrow \mathbb{R}$ uma função continuamente diferenciável tal que:*

$$V(0) = 0 \text{ e } V(x) > 0 \text{ em } D - \{0\}; \quad (3.7)$$

$$\dot{V}(x) \leq 0 \text{ em } D. \quad (3.8)$$

Então, $x = 0$ é estável. Além disso, se:

$$\dot{V}(x) < 0 \text{ em } D - \{0\}. \quad (3.9)$$

Então, $x = 0$ é assintoticamente estável.

¹O sistema não linear (3.1) é dito autônomo se f não depende explicitamente das variáveis independentes (u e t). Caso contrário, o sistema é chamado não autônomo.

Uma função $V(x)$ continuamente diferenciável satisfazendo (3.7) e (3.8) é chamada *função de Lyapunov* em homenagem ao matemático e engenheiro russo criador da teoria. Uma função $V(x)$ que satisfaça a condição (3.7) é chamada de *definida positiva*. Para uma condição menos rigorosa, ou seja $V(x) \geq 0$ para $x \neq 0$, tal função é chamada de *semi definida positiva*. No sentido contrário, para $V(x) < 0$ e $V(x) \leq 0$, define-se tais funções como sendo *definida negativa* e *semi definida negativa*, respectivamente.

Com esta terminologia é possível enunciar o teorema da estabilidade de Lyapunov do seguinte modo: *A origem será um ponto de equilíbrio estável se existir uma função continuamente diferenciável definida positiva $V(x)$ tal que $\dot{V}(x)$ seja semi definida negativa, e assintoticamente estável se $\dot{V}(x)$ for definida negativa*

Quando a origem é assintoticamente estável, é interessante determinar o quão distante a trajetória inicial pode estar e ainda convergir para a origem quando $t \rightarrow \infty$, para tal determinação define-se a *região de atração*.

Seja $\phi(t; x)$ a solução de (3.4) que inicia com valor x no instante $t = 0$. Então, a região de atração é definida como o conjunto de todos os pontos x tal que $\phi(t; x)$ é definida para todo $t \geq 0$ e $\lim_{t \rightarrow \infty} \phi(t; x) = 0$. Encontrar a região de atração analiticamente pode ser muito difícil ou mesmo impossível. Contudo, funções de Lyapunov podem ser utilizadas para estimar a região de atração, ou seja, encontrar conjuntos contendo a região de atração, porém regiões muito conservadoras poderão ser encontradas.

Genericamente, será possível definir a região de atração como sendo todo o \mathbb{R}^n desde que seja possível mostrar que para qualquer estado inicial x , a trajetória $\phi(t; x)$ aproxima da origem quando $t \rightarrow \infty$, não importando o quão grande $\|x\|$ seja. Se um ponto de equilíbrio, assintoticamente estável na origem, possui esta propriedade, ele é dito ser *globalmente assintoticamente estável*. O seguinte teorema formaliza esta propriedade:

Teorema 3.2. *Seja $x = 0$ um ponto de equilíbrio para (3.4). Seja também $V : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função continuamente diferenciável tal que:*

$$V(0) = 0 \text{ e } V(x) > 0, \forall x \neq 0; \quad (3.10)$$

$$\|x\| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty; \quad (3.11)$$

$$\dot{V}(x) < 0, \forall x \neq 0, \quad (3.12)$$

então $x = 0$ é globalmente assintoticamente estável.

Uma função que satisfaz (3.11) é chamada de radialmente ilimitada. Este teorema é conhecido como teorema de *Barbashin-Krasovskii*.

Princípio da Invariância

Para análise da estabilidade assintótica, de acordo com os Teoremas 3.1 e 3.2, é necessário encontrar uma função de Lyapunov cuja derivada temporal seja definida negativa. Porém, se em um domínio ao redor da origem é possível encontrar uma função de Lyapunov cuja derivada ao longo das trajetórias do sistema é semi definida negativa, e, se é possível determinar que não existem trajetórias que podem permanecer identicamente nos pontos onde $\dot{V}(x) = 0$, excetuando-se a origem, então a origem é assintoticamente estável. Esta ideia provém do *Princípio da invariância de LaSalle*.

Para definir o teorema da invariância de LaSalle são necessárias algumas definições. Seja $x(t)$ uma solução de (3.4). Um ponto p é dito ser um limitante positivo de $x(t)$ se existe uma sequência $\{t_n\}$, $t_n \rightarrow \infty$ quando $n \rightarrow \infty$, tal que $x(t_n) \rightarrow p$ quando $n \rightarrow \infty$. O conjunto de todos os limitantes positivos de $x(t)$ é chamado de conjunto limitante positivo de $x(t)$. Um conjunto M é chamado de conjunto invariante em relação a (3.4) se:

$$x(0) \in M \Rightarrow x(t) \in M, \forall t \in \mathbb{R}. \quad (3.13)$$

Isto significa que, se uma solução pertence a M em algum instante, ela vai pertencer a M para todos os instante passados e futuros. Um conjunto M é dito ser *positivamente invariante* se:

$$x(0) \in M \Rightarrow x(t) \in M, \forall t \geq 0. \quad (3.14)$$

É possível também dizer que $x(t)$ se aproxima de um conjunto M quando t se aproxima de infinito, se para cada $\epsilon > 0$ existe $T > 0$ tal que:

$$dist(x(t), M) < \epsilon, \forall t \leq T, \quad (3.15)$$

onde $dist(p, M)$ denota a distância de um ponto p ao conjunto M , ou seja, a menor distância de p para qualquer ponto de M . Mais especificamente:

$$dist(p, M) = \inf_{x \in M} \|p - x\|. \quad (3.16)$$

O teorema de LaSalle é enunciado do seguinte modo:

Teorema 3.3. *Seja $\Omega \subset D$ um conjunto compacto¹ positivamente invariante em relação a $\dot{x} = f(x)$. Seja $V : D \rightarrow \mathbb{R}$ uma função continuamente diferenciável tal que $\dot{V}(x) \leq 0$ em Ω . Seja E o conjunto de todos os pontos em Ω onde $\dot{V}(x) = 0$. Seja M o maior conjunto invariante em E . Então cada solução iniciada em Ω aproxima-se de M quando $t \rightarrow \infty$.*

¹Um conjunto \mathbb{X} é dito compacto se toda sequência contida em \mathbb{X} possui uma sub sequência que converge para algum ponto de \mathbb{X} . Todo conjunto fechado e limitado é compacto

Para mostrar que $x(t) \rightarrow 0$ quando $t \rightarrow \infty$, é preciso estabelecer que o maior conjunto invariante de E é a origem. Isto é feito mostrando que não existem soluções que podem estar identicamente em E que não seja a trivial $x(t) \equiv 0$. Restringindo o Teorema 3.3 a este caso e fazendo $V(x)$ ser definida positiva, obtém-se os seguintes Corolários que estendem os Teoremas 3.1 e 3.2.

Corolário 3.1. *Seja $x = 0$ um ponto de equilíbrio para (3.4). Seja $V : D \rightarrow \mathbb{R}$ uma função continuamente diferenciável definida positiva num domínio D contendo a origem $x = 0$, tal que $\dot{V}(x) \leq 0$ em D . Seja $S = \{x \in D | \dot{V}(x) = 0\}$ suponha que não exista solução que possa permanecer identicamente em S que não seja a solução trivial $x(t) \equiv 0$. Então a origem é assintoticamente estável.*

Corolário 3.2. *Seja $x = 0$ um ponto de equilíbrio para (3.4). Seja $V : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função continuamente diferenciável, radialmente ilimitada, definida positiva, tal que $\dot{V}(x) \leq 0$ para todo $x \in \mathbb{R}^n$. Seja $S = \{x \in \mathbb{R}^n | \dot{V}(x) = 0\}$ suponha que não exista solução que possa permanecer identicamente em S que não seja a solução trivial $x(t) \equiv 0$. Então a origem é globalmente assintoticamente estável.*

Quando $\dot{V}(x)$ é definida negativa, $S = \{0\}$, fazendo os Corolários 3.1 e 3.2 coincidirem com os Teoremas 3.1 e 3.2, respectivamente.

Sistemas Não Autônomos

Seja o seguinte sistema:

$$\dot{x} = f(t, x), \quad (3.17)$$

onde $f : [0, \infty) \times D \rightarrow \mathbb{R}^n$ é um mapa contínuo por partes em t e localmente *Lipschitz* em x no intervalo $[0, \infty) \times D$, e $D \subset \mathbb{R}^n$ é um domínio que contém a origem. A origem é um ponto de equilíbrio para (3.17) em $t=0$ se:

$$f(t, 0) = 0, \forall t \geq 0. \quad (3.18)$$

Assim como para sistemas autônomos, é possível supor, sem perda de generalidade, que a origem pode ser obtida através de uma translação de um ponto de equilíbrio não nulo do sistema.

As noções de estabilidade e estabilidade assintótica de pontos de equilíbrio para sistemas não autônomos são muito similares àquelas introduzidas na Definição 3.1 para sistemas autônomos. Neste caso, o novo elemento é que, enquanto a solução de um sistema autônomo depende somente de $(t - t_0)$, a solução de um sistema não autônomo dependerá de ambos t e t_0 . Por isso, o comportamento da estabilidade do ponto de equilíbrio será, em geral, dependente de t_0 . A origem $x = 0$ é um ponto de equilíbrio estável para (3.17) se, para cada $\epsilon > 0$, e qualquer $t_0 \geq 0$, existir $\delta(\epsilon, t_0) > 0$ tal que:

$$\|x(t_0)\| < \delta \Rightarrow \|x(t)\| < \epsilon, \forall t \geq t_0. \quad (3.19)$$

A constante δ é geralmente dependente do tempo inicial t_0 . A existência de δ para cada t_0 não necessariamente garante que existirá uma constante δ , dependente somente de ϵ , que irá servir para todo t_0 .

Para enfatizar a dependência do comportamento da estabilidade da origem com o instante inicial t_0 , um refinamento da Definição 3.1 deve ser realizado. Objetiva-se redefinir estabilidade e estabilidade assintótica da origem como uma propriedade uniforme relativa ao instante de tempo inicial.

Definição 3.2. *O ponto de equilíbrio $x=0$ de (3.17) é:*

- *Estável se, para cada $\epsilon > 0$, existir um $\delta = \delta(\epsilon, t_0) > 0$ tal que:*

$$\|x(t_0)\| < \delta \Rightarrow \|x(t)\| < \epsilon, \forall t \geq t_0 \geq 0. \quad (3.20)$$

- *Uniformemente estável se, para cada $\epsilon > 0$, existir um $\delta = \delta(\epsilon) > 0$, independente de t_0 , tal que (3.20) seja satisfeita.*
- *Instável se não é estável.*
- *Assintoticamente estável se é estável e existe uma constante positiva $c = c(t_0)$ tal que $x(t_0) \rightarrow 0$ quando $t \rightarrow \infty$, para todo $\|x(t_0)\| < c$.*
- *Uniformemente assintoticamente estável se é uniformemente estável e existe uma constante positiva c , independente de t_0 tal que para todo $\|x(t_0)\| < c$, $x(t_0) \rightarrow 0$ quando $t \rightarrow \infty$, uniformemente em t_0 ; ou seja, para cada $\eta > 0$, existe $T = T(\eta) > 0$ tal que:*

$$\|x(t)\| < \eta, \forall t \geq t_0 + T(\eta), \forall \|x(t_0)\| < c. \quad (3.21)$$

- *Globalmente uniformemente assintoticamente estável se é uniformemente estável, $\delta(\epsilon)$ pode ser escolhido para satisfazer $\lim_{\epsilon \rightarrow \infty} \delta(\epsilon) = \infty$, e, para cada par de números positivos η e c , existe $T = T(\eta, c) > 0$ tal que:*

$$\|x(t)\| < \eta, \forall t \geq t_0 + T(\eta, c), \forall \|x(t_0)\| < c. \quad (3.22)$$

A teoria da estabilidade de Lyapunov pode ainda ser utilizada para mostrar os limitantes das soluções das equações de estado, mesmo quando não existe um ponto de equilíbrio na origem. A seguinte definição caracteriza esta propriedade:

Definição 3.3. *As soluções de (3.17) são:*

- *Uniformemente limitadas se existe uma constante positiva c , independente de $t_0 \geq 0$, e, para cada $a \in (0, c)$, existe $\beta = \beta(a) > 0$, independente de t_0 tal que:*

$$\|x(t_0)\| \leq a \Rightarrow \|x(t)\| \leq \beta, \forall t \geq t_0. \quad (3.23)$$

- *Globalmente uniformemente limitadas se (3.23) é satisfeita para a arbitrariamente grande.*
- *Uniformemente terminalmente limitada com limite terminal b se existem restrições positivas b e c , independentes de $t_0 \geq 0$, e para cada $a \in (0, c)$, existe $T = T(a, b) \geq 0$, independente de t_0 , tal que:*

$$\|x(t_0)\| \leq a \Rightarrow \|x(t)\| \leq b, \forall t \geq t_0 + T. \quad (3.24)$$

- *Globalmente uniformemente terminalmente limitada se (3.24) é satisfeita para a arbitrariamente grande.*

No caso de sistemas autônomos, a palavra “uniformemente” pode ser removida desde que a solução depende somente de $t - t_0$.

3.2 Controladores *NMPC*

Controle Preditivo Baseado em Modelo (*MPC*), também conhecido como *Receding Horizon Control* é uma família de algoritmos de controle avançado originalmente concebido para o suprimento de demandas da indústria petroquímica. Datam do final da década de 70 as primeiras publicações sobre controladores preditivos, especificamente os algoritmos *MAC* (Richalet *et al.* 1978) e *DMC* (Cutler e Ramake 1980).

O avanço das pesquisas para esta classe de controladores viabilizou a sua utilização nos mais variados processos, tais como nas áreas, automotiva, aeroespacial, processamento de alimentos, dentre outras. Isto se deve, em parte, à sua eficácia no tratamento de problemas multivariáveis, instáveis, com atraso e de fase não mínima (Camacho e Alba 2013, Capítulo 1).

De maneira geral, os controladores preditivos baseados em modelo (*MPC*) são formulados através da solução repetida de um problema de controle ótimo em malha aberta sujeita à dinâmica do sistema e à restrições nas entradas e/ou saídas. Baseando-se em medidas das variáveis do processo num instante t , o controlador prediz o comportamento do sistema ao longo de um dado horizonte de predição T_p e define ao longo de um horizonte de controle T_c (com $T_c \leq T_p$) um perfil de controle em malha aberta tal que um dado objetivo é minimizado considerando-se um conjunto de restrições. A Figura 3.1 ilustra este princípio.

O perfil de controle calculado é mantido até o próximo instante de amostragem T_s . Um efeito de realimentação é adicionado através da repetição deste procedimento a cada período de amostragem, sendo este procedimento definido como *Princípio do Horizonte Móvel*, proposto originalmente por (Propoi 1963).

Decorrente de especificações de desempenho cada vez mais rigorosas, desde a década de 90, muitos esforços têm sido dedicados ao desenvolvimento de algoritmos *NMPC*. Esta classe de controladores utiliza os modelos não lineares dos processos e incluem restrições lineares ou não para a resolução de um problema de otimização não linear.

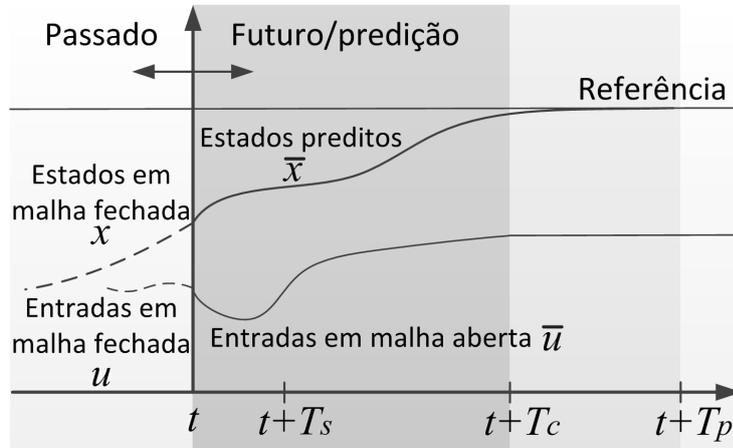


Figura 3.1: Princípio de funcionamento dos controladores preditivos baseados em modelo.

A seguir, tem-se o esquema padrão de funcionamento do *NMPC*:

- 1: Obter estimativas dos estados do sistema.
- 2: Calcular uma entrada ótima que minimiza determinada função objetivo sob um horizonte de predição utilizando o modelo do processo para predição.
- 3: Implementar a primeira parte da entrada de controle ótima até o próximo instante de amostragem.
- 4: Continuar com 1.

Deste esquema de funcionamento, observa-se que os controladores preditivos são estratégias inerentemente discretas, porém, fruto das não linearidades e ausência de algoritmos de discretização eficientes, modelos de tempo contínuo podem ser diretamente utilizados. Neste caso, como os algoritmos permanecem sendo implementados em ambientes digitais, tais controladores pertencem à classe dos sistemas a dados amostrados. Maiores detalhes sobre as diferenças entre estes sistemas e sistemas de tempo discreto podem ser encontrados em (Grüne e Pannek 2011, Capítulo 2).

Os modelos normalmente utilizados para solucionar problemas de seguimento de caminho e controle de formação são não lineares e apresentam dependência temporal implícita, o que viabiliza a escolha de controladores *NMPC* de tempo contínuo. Além disso, a variabilidade dos períodos de amostragens devido a tarefas de sincronização intrínsecas a arquiteturas descentralizadas se torna um grande atrativo à utilização da abordagem de tempo contínuo.

De maneira geral, a formulação matemática destes controladores considera um sistema descrito pela seguinte equação diferencial não-linear:

$$\dot{x}(t) = f(x(t), u(t)); \quad x(0) = x_0, \quad (3.25)$$

sujeito a restrições nas entradas e nas saídas da seguinte forma:

$$u(t) \in \mathcal{U}, \quad \forall t \geq 0, \quad (3.26)$$

$$x(t) \in \mathcal{X}, \quad \forall t \geq 0, \quad (3.27)$$

sendo $x(t) \in \mathbb{R}^n$ e $u(t) \in \mathbb{R}^m$ os vetores de estados e entradas, respectivamente. A princípio, o conjunto \mathcal{U} é compacto e define as entradas factíveis, e o conjunto \mathcal{X} é conectado² e define os estados factíveis. Tais conjuntos são definidos como segue:

$$\mathcal{U} := \{u \in \mathbb{R}^m \mid u_{min} \leq u \leq u_{max}\}, \quad (3.28)$$

$$\mathcal{X} := \{x \in \mathbb{R}^n \mid x_{min} \leq x \leq x_{max}\}. \quad (3.29)$$

Observa-se que para a implementação de controladores *NMPC*, modelos no espaço de estados, normalmente obtidos através de aspectos fenomenológicos dos processos, são necessários. Deste modo, desde que os modelos sejam representativos, devido ao princípio do horizonte móvel, os controladores possuirão características intrínsecas de robustez.

As principais características e propriedades dos controladores *NMPC* são resumidas como segue (Allgöwer *et al.* 2004):

- 1) *NMPC* permite o uso direto de modelos não lineares para predição;
- 2) *NMPC* permite o tratamento explícito de restrições nas entradas e nos estados;
- 3) No controlador *NMPC* um critério de desempenho especificado no domínio do tempo é minimizado *on line*.
- 4) No controlador *NMPC* o comportamento predito é, em geral, diferente do comportamento em malha fechada;
- 5) Para a implementação do *NMPC* a solução em tempo real de um problema de controle ótimo em malha aberta é tipicamente necessária.
- 6) Para executar a predição, os estados do sistema devem ser medidos ou estimados.

²Um conjunto \mathbb{X} é dito conectado se não existem dois conjuntos abertos \mathbb{X}_1 e \mathbb{X}_2 , disjuntos e não vazios, tal que $\mathbb{X} = \mathbb{X}_1 \cup \mathbb{X}_2$.

3.2.1 Predição

De acordo com o esquema padrão de implementação do algoritmo *NMPC*, é necessário estimar a saída do sistema ao longo do horizonte de predição. Desde que existam soluções analíticas para (3.25), estas podem ser utilizadas, sendo, deste modo, constituída uma das principais vantagens do *NMPC* de tempo contínuo.

No caso de sistemas a dados amostrados, em que as soluções analíticas não estão disponíveis, é comum a utilização das funções de Laguerre ((Wang 2001) e (Wang 2009, Capítulo 5)) e expansão por série de Taylor truncada ((Chen *et al.* 1996), (Cao 2005) e (Grüne e Pannek 2011, Capítulo 9)), sendo esta última a estrutura de predição utilizada neste trabalho.

Mais especificamente, admitindo-se que os estados do sistema (3.25) são mensuráveis a todo instante de tempo, a predição de x no instante $t_i + T_s$, é dada como segue:

$$x(t_i + T_s | t_i) = x_{t_i} + T_s x^{[1]} + \frac{T_s^2 x^{[2]}}{2!} + \dots + \frac{T_s^n x^{[n]}}{n!}. \quad (3.30)$$

No presente trabalho, a expressão anterior é truncada em seu termo de primeira ordem e calculada iterativamente ao longo de um horizonte T_p .

Também utiliza-se a série de *Taylor* para parametrizar a trajetória de entrada futura em seu valor atual e em suas m derivadas, ou seja:

$$u(t_i + T_s | t_i) = u_{t_i} + T_s u^{[1]} + \frac{T_s^2 u^{[2]}}{2!} + \dots + \frac{T_s^m u^{[m]}}{m!}. \quad (3.31)$$

Tais valores, truncados em seu termo de primeira ordem, servem de estimativas iniciais das ações de controle, e também são calculadas ao longo de um horizonte T_c através de um procedimento iterativo.

Determinados modelos apresentam não linearidades sob a forma de produtos de funções não lineares dos estados pelas entradas. Esta classe especial de sistemas, não lineares nos estados e lineares nas entradas, são definidos como sistemas *bilineares*. Exemplos desta classe de sistema são largamente encontrados na teoria de sistemas robóticos não holonômicos, tais como o descrito por (2.13) (LaValle 2006, capítulo 13). Nestes casos, um esquema de predição analítico pode ser implementado. Tal esquema é obtido considerando o sistema da seguinte forma:

$$\dot{x}^{[1]} = f(x(t)) + g(x(t))u(t). \quad (3.32)$$

Considerando-se que $x(t)$ é n vezes derivável em t e tomando estas derivadas, omitindo os termos temporais, obtêm-se:

$$x^{[2]} = f^{[1]} + g^{[1]}u + gu^{[1]} = f^{[1]} + C_0^1 g^{[1]}u + C_1^1 gu^{[1]}, \quad (3.33)$$

$$x^{[3]} = f^{[2]} + C_0^2 g^{[2]}u + C_1^2 g^{[1]}u^{[1]} + C_2^2 gu^{[2]}, \quad (3.34)$$

⋮

$$x^{[n]} = f^{[n-1]} + C_0^{n-1} g^{[n-1]}u + C_1^{n-1} g^{[n-2]}u^{[1]} \dots + C_{n-1}^{n-1} gu^{[n-1]},$$

$$x^{[n]} = f^{[n-1]} + \sum_{i=0}^{n-1} C_i^{n-1} g^{[n-1-i]}u^{[i]}, \quad (3.35)$$

sendo $C_m^p = \frac{m!}{p!(m-p)!}$ a combinação simples de m termos p a p .

Substituindo-se (3.32) e (3.35) em (3.30), obtêm-se:

$$\begin{aligned} x(t_i + T_s|t_i) &= x(t_i) + [T_s f + \frac{T_s^2}{2!} f^{[1]} + \dots + \frac{T_s^n}{n!} f^{[n-1]}] + \\ &+ u [C_0^0 g T_s + C_0^1 g^{[1]} \frac{T_s^2}{2!} + C_0^2 g^{[2]} \frac{T_s^3}{3!} \dots + C_0^{n-1} g^{[n-1]} \frac{T_s^n}{n!}] \\ &+ u^{[1]} [C_1^1 g \frac{T_s^2}{2!} + C_1^2 g^{[1]} \frac{T_s^3}{3!} + \dots + C_1^{n-1} g^{[n-2]} \frac{T_s^n}{n!}] + \\ &+ u^{[2]} [C_2^2 g \frac{T_s^3}{3!} + C_2^3 g^{[1]} \frac{T_s^4}{4!} \dots + C_2^{n-1} g^{[n-3]} \frac{T_s^n}{n!}] + \\ &\quad \vdots \\ &+ u^{[n-2]} [C_{n-2}^{n-2} g \frac{T_s^{(n-1)}}{(n-1)!} + C_{n-2}^{n-1} g^{[1]} \frac{T_s^n}{n!}] + u^{[n-1]} [C_{n-1}^{n-1} g \frac{T_s^n}{n!}]. \end{aligned}$$

Matricialmente, tem-se:

$$x(t_i + T_s|t_i) = x(t_i) + F(t_i + T_s|t_i) + \mathbf{G}(t_i + T_s|t_i)\mathbf{U}(t_i), \quad (3.36)$$

sendo:

$$F(t_i + T_s|t_i) = T_s f + \frac{T_s^2}{2!} f^{[1]} + \dots + \frac{T_s^n}{n!} f^{[n-1]}, \quad (3.37)$$

$$\mathbf{G}(t_i + T_s|t_i) = \begin{bmatrix} C_0^0 g T_s + C_0^1 g^{[1]} \frac{T_s^2}{2!} + C_0^2 g^{[2]} \frac{T_s^3}{3!} + \dots + C_0^{n-1} g^{[n-1]} \frac{T_s^n}{n!} \\ C_1^1 g \frac{T_s^2}{2!} + C_1^2 g^{[1]} \frac{T_s^3}{3!} + \dots + C_1^{n-1} g^{[n-2]} \frac{T_s^n}{n!} \\ \vdots \\ C_{n-1}^{n-1} g \frac{T_s^n}{n!} \end{bmatrix}, \quad (3.38)$$

$$\mathbf{U}(t_i) = [u \quad u^{[1]} \quad \dots \quad u^{[n-1]}]. \quad (3.39)$$

3.2.2 Problema de controle ótimo

De maneira geral, as entradas a serem aplicadas ao processo são soluções, a cada período de amostragem, do seguinte problema de controle ótimo em malha aberta:

$$\min_{\bar{u}(\cdot)} J(x(t), \bar{u}(\cdot)), \quad (3.40)$$

$$\text{sujeito a: } \dot{\bar{x}}(\tau) = f(\bar{x}(\tau), \bar{u}(\tau)), \quad \bar{x}(t) = x(t), \quad (3.41)$$

$$\bar{u}(\tau) \in \mathcal{U}, \forall \tau \in [t, t + T_c], \quad (3.42)$$

$$\bar{u}(\tau) = \bar{u}(t + T_c), \forall \tau \in [t + T_c, t + T_p], \quad (3.43)$$

$$\bar{x}(\tau) \in \mathcal{X}, \forall \tau \in [t, t + T_p], \quad (3.44)$$

com a seguinte função objetivo:

$$J(x(t), \bar{u}(\cdot)) := \int_t^{t+T_p} F(\bar{x}(\tau), \bar{u}(\tau)) d\tau, \quad (3.45)$$

A função F (custo por estágio), é utilizada para definir as especificações de desempenho, podendo assumir diversos formatos a depender das aplicações. Os problemas de seguimento de caminho e controle de formação possuem estruturas específicas para suas funções objetivo, as quais serão apresentadas nos próximos capítulos.

A barra sobre as variáveis das expressões anteriores é para diferenciar as variáveis reais daquelas internas ao controlador, já que em geral se tratam de grandezas distintas. Especificamente, $\bar{x}(\cdot)$ é a solução de (3.41) conduzida pelo sinal de controle interno ao controlador $\bar{u}(\cdot) : [t, t + T_p] \rightarrow \mathcal{U}$ sob a condição inicial $x(t)$.

Após o processo de otimização, apenas o primeiro termo da sequência de controle ótimo obtida é aplicado ao sistema até o período de amostragem seguinte, quando todo o processo se repete.

A despeito de todos os benefícios da utilização do *NMPC*, tal estratégia, no formato apresentado, possui imperfeições que restringem a sua aplicabilidade. Mais especificamente, tais imperfeições dizem respeito aos requisitos de factibilidade e estabilidade.

Factibilidade

O problema de controle ótimo caracterizado anteriormente, deve ser melhor analisado no contexto das aplicações, objetivando-se explicitar os requisitos de factibilidade, característica inerente a qualquer abordagem para garantia de estabilidade de controladores *NMPC*. Primeiramente, os conjuntos \mathcal{U} e \mathcal{X} devem ser caracterizados a fim de definir valores limitantes para as entradas e para os estados no processo de otimização, respectivamente.

O problema de otimização será factível na inicialização se, para um dado estado inicial \mathbf{x}_0 , o conjunto de entradas admissíveis $\mathcal{U}(\mathbf{x}_0)$ ³ não é vazio. Existirá factibilidade ao longo do horizonte de controle quando as seguintes condições são satisfeitas para todo $t_k = 0, 1, \dots, T_c - 1$:

$$u(t_k) \in \mathcal{U}(\mathbf{x}_u(t_k), \mathbf{x}_0); \quad (3.46)$$

$$\mathbf{x}_u(t_{k+1}, \mathbf{x}_0) \in \mathcal{X}, \quad (3.47)$$

sendo:

$u(t_k)$ a sequência de controle atual;

$\mathbf{x}_u(t_k, \mathbf{x}_0)$ a solução de (3.25) no instante t_k .

Desde que somente problemas de controle ótimo factíveis possuem soluções admissíveis, os pontos $\mathbf{x}_0 \in \mathcal{X}$, satisfazendo $\mathcal{U}(\mathbf{x}_0) \neq \emptyset$ a cada instante, são exatamente os pontos para os quais a lei de controle implícita em malha fechada μ_{T_p} é bem definida.

Para formalizar a análise de factibilidade (Grüne e Pannek 2011, Capítulo 8) fornece a seguinte definição:

Definição 3.4. *Seja um conjunto de restrições de estado \mathcal{X} e um horizonte de predição $T_p \in \mathbb{N}$, para o algoritmo *NMPC* descrito por (3.40) até (3.45) tem-se:*

(i) *Um ponto $x \in \mathcal{X}$ é chamado de factível para \mathcal{X} e T_p se $\mathcal{U}(\mathbf{x}_0) \neq \emptyset$*

(ii) *O conjunto factível para \mathcal{X} e T_p é definido como:*

$$F_N := \{x \in \mathcal{X} | x \text{ é factível para } \mathcal{X} \text{ e } T_p\}. \quad (3.48)$$

O conjunto F_∞ é também chamado de núcleo de viabilidade.

(iii) *Um conjunto $A \subseteq \mathcal{X}$ é chamado de recursivamente factível para o horizonte de predição $T_p \in \mathbb{N}$ se $A \subseteq F_N$ e é positivamente invariante à lei de controle *NMPC* implícita μ_{T_p} , isto é, $f(x, \mu_{T_p}(x)) \in A$ é válida para todo $x \in A$.*

³Notação que representa o mapeamento de (3.42), (3.43) e (3.44) para a variável manipulada $\mathbf{u}(\mathbf{t}_k)$.

A propriedade da factibilidade recursiva da definição (iii) garante que para qualquer valor inicial $x \in A$ o *NMPC* em malha fechada irá gerar soluções admissíveis para todos os instantes futuros. Formalmente, o seguinte lema define a factibilidade recursiva:

Lema 3.1. *Seja $\mathcal{A} \subseteq \mathcal{X}$ recursivamente factível para o problema *NMPC* descrito anteriormente com horizonte de predição $T_p \in \mathbb{N}$. Então para cada $\mathbf{x} \in \mathcal{A}$ a solução implícita em malha fechada $\mathbf{x}_{\mu_{T_p}}(t_k, x)$ é bem definida para todo $t_k \in \mathbb{N}_0$ e satisfaz $\mathbf{x}_{\mu_N}(t_k, x) \in \mathcal{A}$ e assim também $\mathbf{x}_{\mu_{T_p}}(t_k, x) \in \mathcal{X}$ para todo $t_k \in \mathbb{N}_0$.*

A prova deste lema é obtida por indução direta usando $\mathbf{x}_{\mu_{T_p}}(t_k, x)$ e algum \mathcal{A} positivamente invariante a μ_{T_p} , isto é, se $f(x, \mu_{T_p}) \in \mathcal{A}$ é válido para todo $x \in \mathcal{A}$.

Ainda em (Grüne e Pannek 2011, Capítulo 8) são propostas duas abordagens para garantia de factibilidade sem qualquer alteração na função objetivo ou restrições. A primeira utiliza propriedades da interação entre f e \mathcal{X} e é independente de qualquer propriedade de estabilidade, neste caso, determina-se o horizonte de predição mínimo para a manutenção da factibilidade para determinados valores limitantes de \mathcal{U} . A segunda abordagem utiliza a estabilidade assintótica em malha fechada para garantir a factibilidade de subconjuntos no espaço de estados.

Para o *MPC* linear muitas abordagens para garantia de factibilidade estão disponíveis. Cita-se, por exemplo, (Rawlings e Muske 1993) que sugere remover as restrições durante um período inicial do horizonte de predição e considerá-las em um momento posterior. (Zafiriou e Chiou 1993) propõe uma metodologia para calcular uma quantidade apropriada de relaxamento das restrições que tornam o problema de otimização factível num caso *SISO*. Seguindo a ideia de relaxamento das restrições, cita-se ainda (Alvarez e de Prada 1997) que aplica relaxamento das restrições tanto nos estados quanto nas entradas do sistema.

Em (Scokaert 1994) são propostas várias estratégias para resolver o problemas de infactibilidade, dentre as quais, destaca-se uma baseada na classificação das restrições em níveis de prioridade satisfazendo apenas os níveis mais altos, relaxando os níveis mais baixos. Já em (Vada *et al.* 2001) também é proposta uma classificação das restrições em níveis de prioridades, porém, um problema de programação linear é resolvido em paralelo com o problema de otimização padrão do *MPC* para a determinação de uma métrica de relaxamento das restrições relaxáveis.

Todas estas proposições objetivam manipular os conjuntos das restrições para a garantia de factibilidade recursiva, sendo necessária uma análise da possibilidade de aplicação ao *NMPC* distribuído. Porém, delimitando-se ao âmbito desta tese, opta-se por uma solução que seja mais simples e eficiente do ponto de vista computacional. Mais especificamente, sabe-se que a definição dos limites de \mathcal{U} e \mathcal{X} baseados em elementos físicos intrínsecos aos sistemas robóticos, a factibilidade inicial e recursiva irá ser limitada a pequenas regiões de atração, o que permite a aplicação do algoritmo nominal apresentado somente para a resolução de problemas muito bem caracterizados e sem variância temporal de parâmetros.

Neste trabalho é proposta uma abordagem para garantia de factibilidade que se aproveita da estrutura dos problemas de seguimento de caminhos e controle de formação para relaxamento das restrições. Mais especificamente, desde que as entradas de controle definidas nos capítulos subsequentes existem apenas no domínio matemático, é possível remover qualquer sentido físico na definição dos limitantes de \mathcal{U} , isto é, valores elevados são definidos com posterior redução a valores admissíveis na plataforma experimental.

Os valores limitantes para as restrições dos erros de estado (limitantes de \mathcal{X}) são definidos como segue:

$$\mathbf{x}^{bound} = \mathbf{x}^{max} + \mathbf{x}^{tol}; \quad (3.49)$$

$$\mathbf{x}^{bound} \in \mathcal{X}, \quad (3.50)$$

sendo \mathbf{x}^{max} um vetor dos erros máximos de cada estado ao longo do horizonte de predição e \mathbf{x}^{tol} um vetor constante das tolerâncias, baseadas nos critérios físicos de cada estado. Mais especificamente, \mathbf{x}^{tol} representa tolerâncias para erros de distância e velocidade, definidas através de critérios práticos, conforme detalhado nos capítulos subsequentes.

O Algoritmo 1 sintetiza a ideia proposta para garantia de factibilidade.

Com este tratamento da factibilidade, subotimalidade proposta por (Scokaert *et al.* 1999) pode ser utilizada como estratégia para garantia de estabilidade, desde que, com esta abordagem, factibilidade implica em estabilidade.

Este tratamento das restrições nos estados é de certa forma similar à abordagem denominada por *restrições brandas* (do inglês, “*soft-constraint*”) (de Oliveira e Biegler 1994), porém sem qualquer penalização explícita na função objetivo, mantendo-se baixas especificações de complexidade computacional.

Tal proposta é utilizada para seguimento de caminhos com elevados erros iniciais e variação temporal de parâmetros. Além disso, para o controle de formação, esta técnica é útil para manutenção das vantagens do acoplamento através da função objetivo já que há garantia de comunicação e cálculo dos termos de acoplamento entre robôs vizinhos a todo instante de tempo.

Algoritmo 1: GARANTIA DE FACTIBILIDADE

Entrada: $\mathbf{x} = [x_0, x_1, \dots, x_{T_p}]^T$, $\mathbf{u} = [u_0, u_1, \dots, u_{T_c}]^T$, \mathbf{x}^{tol}

Saída: \mathcal{U} e \mathcal{X}

1 **início**

2 $\{k \in \mathbb{Z}_+ \mid k > 100\}$
3 $\overline{\lim} \mathcal{U} = k \sum_{i=0}^{i=T_c} |u_i|$
4 $\overline{\lim} \mathcal{X} = \max(\mathbf{x}) + \mathbf{x}^{tol}$

5 **fim**

6 **retorna** \mathcal{U} e \mathcal{X}

Estabilidade

O principal requisito para qualquer sistema de controle refere-se à estabilidade em malha fechada. Conforme elucidado anteriormente, o mecanismo de realimentação dos controladores preditivos baseia-se no Princípio do Horizonte Móvel sendo assim necessária a realização de predições ao longo de um intervalo de tempo factível.

Horizontes de predição finitos não são suficientes para capturar por completo a dinâmica do sistema, de modo que se deve realizar uma definição criteriosa deste parâmetro de sintonia. A Figura 3.2 ilustra a diferença entre os estados preditos e o comportamento em malha fechada dos estados reais para um sistema de segunda ordem genérico.

Em geral, deve-se definir um horizonte de predição longo, idealmente infinito para que se possa garantir estabilidade em malha fechada. Tal requerimento é inviável do ponto vista prático, pois os custos computacionais envolvidos seriam proibitivos, sendo assim, é necessário o uso de horizonte finito.

Existem diversas abordagens que garantem estabilidade de controladores *NMPC*, sendo as principais baseadas na adição de critérios puramente matemáticos à função custo responsáveis pela convergência da dinâmica em malha fechada.

A técnica mais bem difundida, apresentada em (Mayne *et al.* 2000), baseia-se na utilização de um termo de penalidade terminal aos objetivos e uma região terminal implementada através de restrições de igualdade ou desigualdade.

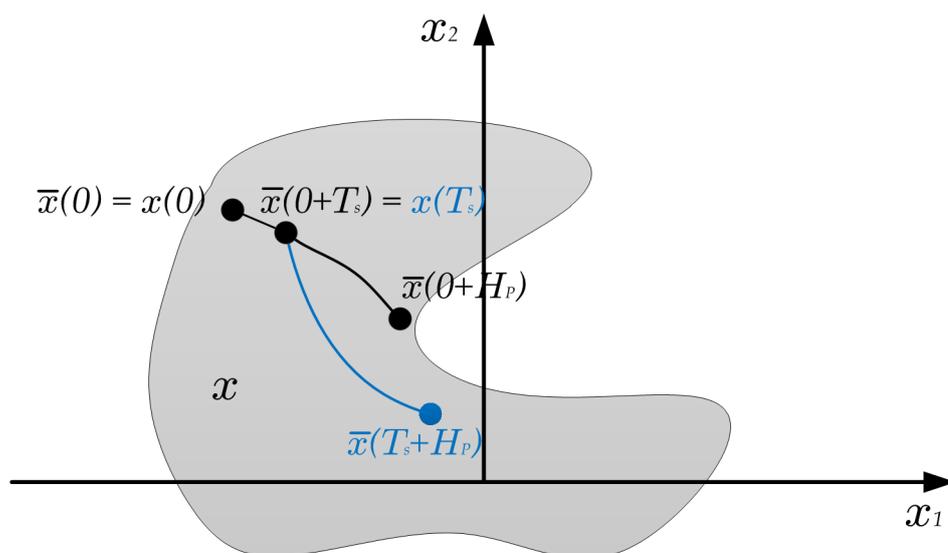


Figura 3.2: Diferença entre as predições em malha aberta e o comportamento em malha fechada.

Neste caso a função objetivo fica da seguinte forma:

$$\min_{\bar{u}(\cdot)} \int_t^{t+T_p} F(\bar{x}(\tau), \bar{u}(\tau)) d\tau + V(\bar{x}(t+T_p)), \quad (3.51)$$

$$\text{sujeito a: } \dot{\bar{x}}(\tau) = f(\bar{x}(\tau), \bar{u}(\tau)), \quad \bar{x}(t) = x(t), \quad (3.52)$$

$$\bar{u}(\tau) \in \mathcal{U}, \forall \tau \in [t, t+T_c], \quad (3.53)$$

$$\bar{u}(\tau) = \bar{u}(t+T_c), \forall \tau \in [t+T_c, t+T_p], \quad (3.54)$$

$$\bar{x}(\tau) \in \mathcal{X}, \forall \tau \in [t, t+T_p], \quad (3.55)$$

$$\bar{x}(t+T_p) \in \Omega, \quad (3.56)$$

onde $V(\bar{x}(t+T_p))$ é o termo de penalidade terminal e Ω é a região terminal.

O seguinte teorema enunciado por (Allgöwer *et al.* 2004) fornece uma métrica para se determinar os termos de penalidade e restrições terminais:

Teorema 3.4. *Assumir que:*

- 1) $\mathcal{U} \subset \mathbb{R}^m$ é compacto, $\mathcal{X} \subset \mathbb{R}^n$ é conectado e a origem está contida no interior de $\mathcal{U} \times \mathcal{X}$.
- 2) O campo vetorial $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ é contínuo em u , localmente Lipschitz em x e satisfaz $f(0, 0) = 0$.
- 3) $F : \mathbb{R}^n \times \mathcal{U} \rightarrow \mathbb{R}$ é contínuo em todos os seus argumentos com $F(0, 0) = 0$ e $F(x, y) > 0 \forall (x, y) \in \mathbb{R}^n \times \mathcal{U} \setminus \{0, 0\}$.
- 4) O termo de penalidade terminal $V : \Omega \rightarrow \mathbb{R}$ é contínuo com $V(0) = 0$ e a região terminal Ω é dada por $\Omega := \{x \in X | V(x) \leq v_1\}$ para algum $v_1 > 0$ tal que $\Omega \subset \mathcal{X}$.
- 5) Existe uma lei de controle contínua local $\mu_{T_p} = k(x)$ tal que $k(x) \in \mathcal{U}$ para todo $x \in \Omega$ e

$$\frac{\partial V}{\partial x} f(x, k(x)) + F(x, k(x)) \leq 0, \quad \forall x \in \Omega. \quad (3.57)$$

- 6) O problema de controle ótimo em malha aberta representado por (3.51) até (3.56) possui solução factível em $t = 0$. Então, para qualquer instante de amostragem $0 < \delta \leq T_p$ o sistema nominal em malha fechada dado por $\dot{x}(t) = f(x(t), \bar{u}^*(t; x(t_j)))$, com \bar{u}^* a sequência de controle ótimo, é assintoticamente estável e a região de atração \mathbf{R} é dada pelo conjunto dos estados para os quais o problema de controle ótimo em malha aberta possui uma solução factível.

Em muitos casos, esta solução não é funcional do ponto de vista prático, pois os custos computacionais envolvidos ainda seriam elevados, não sendo, portanto, garantida a estabilidade devido a problemas de infactibilidade.

Em (Mayne *et al.* 2000) tem-se uma revisão completa das principais abordagens para garantia de estabilidade de controladores *NMPC*.

3.2.3 Otimizadores Não lineares

Neste trabalho, utiliza-se o software *DONLP2* para as tarefas de otimização nas implementações com o software *Decision*. Trata-se de um software livre para finalidades acadêmicas e desenvolvido em linguagem *C*, o que garante total portabilidade a diversas aplicações.

A versão aqui utilizada é denominada *donlp2-intv-dyn* (Spellucci 1998) e trata-se de um otimizador não linear de variáveis contínuas e de propósito geral. Este software é utilizado para a minimização de uma função real não linear e diferenciável f sujeita a restrições de igualdade e desigualdade. Especificamente, os problemas resolvidos por este software possuem a seguinte estrutura geral:

$$\begin{aligned} f(x^*) &= \min\{f(x) : x \in \mathcal{S}\}; \\ \mathcal{S} &= \{x \in \mathbb{R}^n : x_u \leq x \leq x_o, \\ &\quad b_u \leq Ax \leq b_o, \\ &\quad c_u \leq c(x) \leq c_o\}. \end{aligned} \tag{3.58}$$

Sendo A uma matriz de dimensão $\text{nlin} \times n$, sendo nlin o número de restrições lineares e n o número de variáveis a serem minimizadas e c_u e c_o são vetores cujos componentes são avaliações das funções não-lineares que definem as restrições deste tipo e com dimensão nonlin , sendo nonlin o número de restrições não lineares.

O método de otimização não linear utilizado é a programação quadrática sequencial com a utilização da técnica de conjuntos ativos. Dentre as diversas funcionalidades deste programa podem ser citadas a avaliação analítica ou numérica dos gradientes, avaliação externa de função objetivo, etc.

São disponibilizados três métodos de diferenciação numérica, sendo que para redução dos custos computacionais envolvidos nesta etapa, utiliza-se o método das diferenças divididas de Newton. Ao final do processo de otimização é disponibilizado, se necessário, um relatório em formato *txt*.

Um dos benefícios deste software em termos computacionais é a utilização de alocação dinâmica de memória, característica bastante atrativa quando se utilizam sistemas variantes no tempo.

A comunicação entre o software *Decision* e este otimizador é feita através de objetos estáticos gerados pela compilação de códigos-fonte nas linguagens *C* e *Freepascal*, dotados de funções e procedimentos que possibilitam a passagem de parâmetros entre estas diferentes linguagens de programação.

Maiores detalhes sobre a utilização do algoritmo *DONLP2* podem ser encontrados em (Spellucci 1999).

Nas simulações em *Matlab*, utiliza-se o módulo *SOLNP* (Ye 1987). Este módulo resolve problemas gerais de programação não linear do seguinte formato:

$$\begin{aligned} & \min f(x); \\ \text{sujeito a: } & g(x) = 0, \end{aligned} \tag{3.59}$$

$$\begin{aligned} & l_h \leq h(x) \leq u_h, \\ & l_x \leq x \leq u_x. \end{aligned} \tag{3.60}$$

onde $x \in \mathbb{R}^n$, $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, $g(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{m_1}$, $h(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{m_2}$, $l_h, u_h \in \mathbb{R}^{m_2}$ com $l_h < u_h$ e $l_x, u_x \in \mathbb{R}^{m_2}$ com $l_x < u_x$. Em geral, f , g e h são quaisquer funções suaves não lineares que podem ser especificadas e calculadas no *Matlab*.

Adicionando folgas às restrições de desigualdade, o algoritmo *SOLNP* converte o problema anterior em:

$$\begin{aligned} & \min f(x); \\ \text{sujeito a: } & g(x) = 0, \end{aligned} \tag{3.61}$$

$$l_x \leq x \leq u_x. \tag{3.62}$$

A k – ésima principal iteração deste algoritmo resolve um problema de otimização linearmente restrito com o método do Lagrangeano aumentado da seguinte forma:

$$\begin{aligned} & \min f(x) - y^k g(x) + (\rho/2) \|g(x)\|^2; \\ \text{sujeito a: } & J^k(x - x^k) = -g(x^k), \end{aligned} \tag{3.63}$$

$$l_x \leq x \leq u_x. \tag{3.64}$$

onde J^k é uma aproximação numérica para o Jacobiano, dada como segue:

$$J^k = \frac{\partial g}{\partial x} \Big|_{x^k}. \tag{3.65}$$

e y^k é o vetor dos multiplicadores de Lagrange na k – ésima principal iteração. Dentro deste passo, primeiramente verifica-se a factibilidade de x^k , em caso de in-factibilidade, um procedimento adicional baseado em programação linear é chamado para tentar encontrar uma solução factível.

Em seguida implementa-se programação quadrática sequencial para calcular o vetor gradiente g e atualizar a matriz Hessiana H usando a técnica de Broyden-

Fletcher-Goldfarb-Shanno para a função objetivo do Lagrangeano aumentado e então resolver um problema quadrático da seguinte forma:

$$\begin{aligned} \min (1/2)(x - x^k)^T H(x - x^k) + g^T(x - x^k); \\ \text{sujeito a: } J^k(x - x^k) = -g(x^k), \end{aligned} \quad (3.66)$$

$$l_x \leq x \leq u_x. \quad (3.67)$$

Maiores detalhes sobre a utilização do algoritmo *SOLNP* podem ser encontrados em (Ye 1990).

3.3 Conclusões

Este capítulo foi dedicado à apresentação dos controladores *NMPC*. Os fundamentos teóricos essenciais dos sistemas não-lineares, critérios de factibilidade e estabilidade no âmbito destes controladores foram enunciados e discutidos.

Após discussão do seu padrão de funcionamento e particularidades, foi definida a modalidade do *NMPC* utilizada. Especificamente, o conhecimento prévio dos modelos utilizados nesta tese, norteou a escolha da abordagem em tempo contínuo com predição feita a partir de série de Taylor truncada.

Etapa importante deste capítulo, após apresentação da estrutura geral do *NMPC* como um problema de controle ótimo, tendo como requisito de projeto, controladores com baixos níveis de complexidade computacional, foi a apresentação de uma proposta para garantia de factibilidade recursiva que viabiliza a aplicação do *NMPC* de maneira eficiente em cenários mais realísticos. Tal técnica baseia-se no relaxamento das restrições nas entradas e nos estados sem penalização explícita nas funções objetivos, característica que permite a implementação eficiente dos controladores *NMPC* distribuídos, conforme será elucidado no Capítulo 5.

Para garantia de estabilidade, utiliza-se uma técnica clássica baseada na adição de um termo de penalidade terminal e de uma zona terminal para as restrições.

Finalmente, os otimizadores não lineares utilizados foram brevemente apresentados.

Nos próximos capítulos, os controladores *NMPC* apresentados genericamente no presente capítulo serão caracterizados especificamente para a resolução dos problemas de seguimento de caminho e controle de formação de um grupo de robôs.

Capítulo 4

Seguimento de Caminho

Neste trabalho, definem-se como tarefas para os robôs em formação as atividades de seguimento de caminho (Canudas de Wit *et al.* 1993, Jiang e Nijmeijer 1999). Tal abordagem consiste numa das três mais utilizadas para a solução dos problemas de controle de movimentos de robôs móveis, sendo as outras duas denominadas estabilização em pontos de trajetória (Brockett 1983) e rastreamento de trajetórias (Freund e Mayr 1997).

Para a técnica de seguimento de caminho, o caminho a ser seguido deve ser parametrizado em unidades de comprimento de caminho, diferentemente do rastreamento de trajetória, onde o caminho a ser seguido é parametrizado pelo tempo. Uma das vantagens da técnica de seguimento de caminho com relação ao rastreamento de trajetórias é que a omissão dos termos temporais permite uma implementação adequada à concepção de sistemas de tempo real, sendo possível inclusive utilizar implementações de tempo contínuo para a determinação das leis de controle.

Outra vantagem importante se refere ao grau de liberdade adicional para ajustes das velocidades de navegação. No caso do rastreamento de trajetórias, os perfis velocidades são calculados a partir de funções explícitas do tempo.

Na estabilização em pontos não há uma descrição matemática completa do caminho através de quaisquer dos dois tipos de parâmetros anteriormente citados, sendo feita apenas a atualização das referências de postura quando o robô se encontra numa região de tolerância próxima de um ponto da trajetória. Isto consiste numa desvantagem desta técnica, em especial quando utilizada para navegação de robôs com restrições não-holonômicas, já que não poderá ser obtida uma lei de controle contínua via realimentação.

Tem-se como principal desvantagem da técnica para seguimento de caminho a necessidade de completa caracterização de caminhos contínuos previamente especificados.

Este capítulo é dedicado ao projeto e análise de controladores *NMPC* para seguimento de caminhos para robôs omnidirecionais, diferenciais e quadricópteros. Como uma das principais contribuições desta tese, propõe-se uma técnica para seguimento de caminhos baseada em visão computacional. Aspectos de factibilidade e estabi-

lidade são considerados e resultados experimentais e simulados são fornecidos. Em todos os casos, a técnica para garantia de factibilidade, proposta no capítulo anterior, é implementada, sendo possível confirmar melhor desempenho dos controladores *NMPC* que se baseiam nesta solução.

4.1 Seguimento de caminho para robôs omnidirecionais

Para a modelagem aqui realizada, considera-se como premissa que as posturas instantâneas do robô com relação ao sistema de coordenadas do mundo são conhecidas através do sistema odométrico. Além disso as definições 2.1 e 2.2 são válidas.

4.1.1 Formalização do Problema

Inicialmente, define-se uma velocidade de navegação v_{nav} , que representa a velocidade do centro de massa do robô ao longo de um dado caminho num plano, conforme ilustrado na Figura 4.1. Tal velocidade relaciona-se com as velocidades do centro de massa da seguinte forma:

$$\begin{bmatrix} v \\ v_n \end{bmatrix} = \begin{bmatrix} v_{nav} \cos \delta \\ v_{nav} \sin \delta \end{bmatrix}, \quad (4.1)$$

sendo que $\delta = \phi - \theta$, representa o ângulo da direção de movimento com relação ao sistema de coordenadas do robô, e ϕ o ângulo desta mesma direção de movimento com relação ao sistema de coordenadas do mundo.

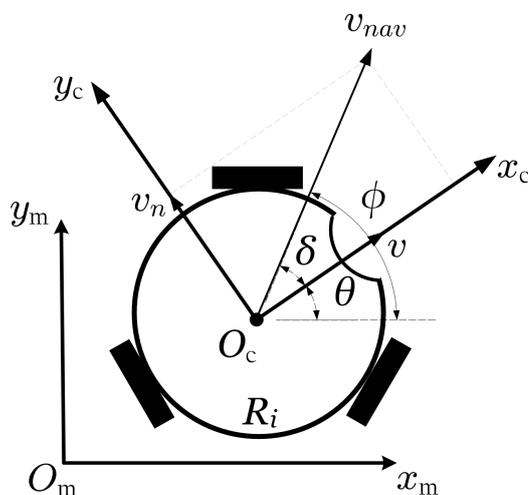


Figura 4.1: Sistemas de coordenadas do robô móvel omnidirecional.

O problema de seguimento de caminho originalmente proposto em (Samson e Ait-Abderrahim 1991) consiste no desenvolvimento de um perfil de velocidade desejado através de v_{nav} , enquanto os controladores atuam sobre a orientação do robô para conduzi-lo ao longo do caminho. As referências neste caso são obtidas pela projeção vertical do veículo ao longo do caminho, o que ocasiona problemas de singularidades quando o robô se encontra em determinadas posições.

Tais singularidades obrigam que a posição inicial esteja restrita a um cilindro ao redor do caminho, com raio inferior a $\frac{1}{C_{c,max}}$ sendo $C_{c,max}$ a máxima curvatura ao longo do caminho. Este problema é analisado em detalhes e solucionado em (Soetanto *et al.* 2003) através da adição de um grau de liberdade na modelagem ao se trabalhar com elementos de geometria diferencial, conforme elucidado a seguir.

A Figura 4.2 ilustra o robô móvel omnidirecional juntamente com um caminho de referência Γ a ser seguido e os sistemas de coordenadas utilizados na modelagem.

O modelo cinemático de robô móvel pode ser definido com relação a um sistema de *serret-frenet* (Mansfield 2010), $\{F\}$, movendo-se ao longo de um caminho de referência. Tal sistema é utilizado de maneira genérica para descrever as propriedades cinemáticas de uma partícula que se move ao longo de uma curva contínua e

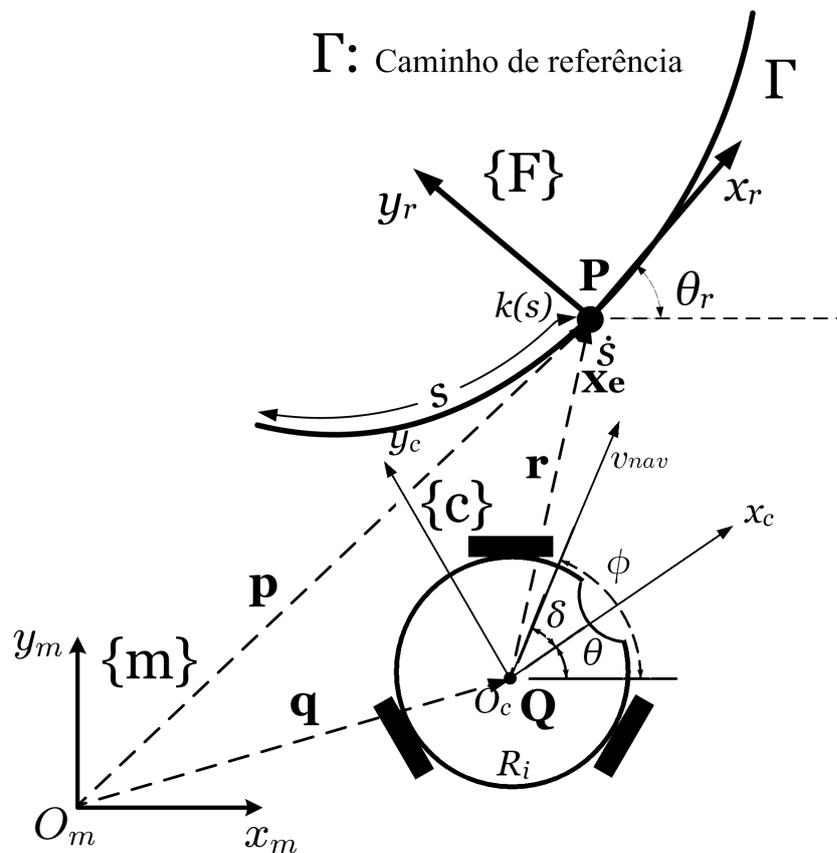


Figura 4.2: Ilustração do problema de seguimento de caminho.

diferenciável em \mathbb{R}^3 .

Tal sistema é utilizado para modelar a movimentação de um veículo virtual que deve ser seguido pelo veículo real. A postura deste veículo virtual é definida por um vetor de estados de referência $[x_r \ y_r \ \theta_r]^T$.

Obtém-se o vetor de erro entre os estados do robô real e os estados de referência com relação a um sistema de coordenadas fixado na extremidade do trecho de caminho s definida pelo ponto \mathbf{P} da seguinte forma:

$$\mathbf{x}_e = \begin{bmatrix} x_e \\ y_e \\ \phi_e \end{bmatrix} = \begin{bmatrix} \cos \theta_r & \sin \theta_r & 0 \\ -\sin \theta_r & \cos \theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_r \\ y - y_r \\ \phi - \theta_r \end{bmatrix}. \quad (4.2)$$

Derivando-se x_e , y_e e ϕ_e com relação ao tempo e fazendo as substituições adequadas, obtêm-se:

$$\dot{x}_e = y_e \dot{\theta}_r + \dot{x} \cos \theta_r + \dot{y} \sin \theta_r - \dot{x}_r \cos \theta_r - \dot{y}_r \sin \theta_r, \quad (4.3)$$

$$\dot{y}_e = -x_e \dot{\theta}_r - \dot{x} \sin \theta_r + \dot{y} \cos \theta_r - \dot{x}_r \sin \theta_r - \dot{y}_r \cos \theta_r, \quad (4.4)$$

$$\dot{\phi}_e = \dot{\phi} - \dot{\theta}_r. \quad (4.5)$$

Sabe-se que a curvatura K em função do comprimento de caminho s é dada por:

$$K(s) = \frac{\dot{\theta}_r}{\dot{s}}. \quad (4.6)$$

A grandeza \dot{s} determina a taxa de atualização das referências, sendo, portanto, definida a taxa de progressão do veículo virtual ao longo do caminho. Tal grandeza relaciona-se com as referências x_r e y_r da seguinte forma:

$$\dot{s} = \sqrt{\dot{x}_r^2 + \dot{y}_r^2}. \quad (4.7)$$

Completa-se a descrição do caminho em função das coordenadas curvilíneas s e θ_r através de:

$$\dot{\theta}_r = \arctan\left(\frac{\dot{y}_r}{\dot{x}_r}\right). \quad (4.8)$$

Destas relações, pode-se verificar que as taxas de variação das coordenadas retangulares de referência podem ser dadas por:

$$\dot{x}_r = \dot{s} \cos \theta_r, \quad (4.9)$$

$$\dot{y}_r = \dot{s} \sin \theta_r. \quad (4.10)$$

A curvatura K também pode ser obtida através destas coordenadas da seguinte forma:

$$K(x_r, y_r) = \frac{\left| \frac{d^2 y_r}{dx_r^2} \right|}{\left[1 + \left(\frac{dy_r}{dx_r} \right)^2 \right]^{\frac{3}{2}}}. \quad (4.11)$$

Substituindo-se (4.6), (4.9) e (4.10) em (4.3), (4.4) e (4.5), obtêm-se:

$$\dot{x}_e = y_e K(s) \dot{s} - \dot{s} + \dot{x} \cos \theta_r + \dot{y} \sin \theta_r, \quad (4.12)$$

$$\dot{y}_e = -x_e K(s) \dot{s} - \dot{s} \sin \theta_r + \dot{y} \cos \theta_r, \quad (4.13)$$

$$\dot{\phi}_e = \dot{\phi} - K(s) \dot{s}. \quad (4.14)$$

Utilizando-se (2.5), obtêm-se:

$$\dot{x}_e = y_e K(s) \dot{s} - \dot{s} + v \cos(\theta - \theta_r) - v_n \sin(\theta - \theta_r), \quad (4.15)$$

$$\dot{y}_e = -x_e K(s) \dot{s} - \dot{s} \sin(\theta - \theta_r) - v_n \cos(\theta - \theta_r), \quad (4.16)$$

$$\dot{\phi}_e = \dot{\phi} - K(s) \dot{s}. \quad (4.17)$$

Finalmente, de (4.1) e das relações angulares definidas, obtêm-se:

$$\dot{x}_e = y_e K(s) \dot{s} - \dot{s} + v_{nav} \cos \phi_e, \quad (4.18)$$

$$\dot{y}_e = -x_e K(s) \dot{s} + v_{nav} \sin \phi_e, \quad (4.19)$$

$$\dot{\phi}_e = \dot{\phi} - K(s) \dot{s}. \quad (4.20)$$

Conforme mencionado anteriormente, translação e rotação do robô móvel omnidirecional podem ser controlados independentemente, sendo possível, deste modo, encontrar valores factíveis de \dot{s} e $\dot{\phi}_e$ que conduzam o erro de estado para zero.

Com o outro grau de liberdade, direciona-se o robô para uma orientação desejada simultaneamente. Considerando tal orientação como sendo o ângulo da tangente ao caminho de referência θ_r e definido-se o erro de orientação com sendo $\theta_e = \theta - \theta_r$, as equações dos erros de estados se tornam:

$$\dot{x}_e = y_e K(s) \dot{s} - \dot{s} + v_{nav} \cos \phi_e, \quad (4.21)$$

$$\dot{y}_e = -x_e K(s) \dot{s} - v_{nav} \sin \phi_e, \quad (4.22)$$

$$\dot{\theta}_e = \omega - \omega_r, \quad (4.23)$$

sendo $\omega = \dot{\theta}$ e $\omega_r = \dot{\theta}_r$.

Redefinindo os sinais de controle da seguinte forma:

$$\mathbf{u}_e = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -\dot{s} + v_{nav} \cos \phi_e \\ v_{nav} \sin \phi_e \\ \omega - \omega_r \end{bmatrix}, \quad (4.24)$$

obtém-se, o seguinte modelo final para os erros de estados:

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} y_e k(s) \dot{s} + u_1 \\ -x_e k(s) \dot{s} + u_2 \\ u_3 \end{bmatrix}. \quad (4.25)$$

Considerando o vetor \mathbf{x}_e como saída, o problema de controlar os movimentos do robô omnidirecional através da técnica de seguimento de caminho pode ser resumido da seguinte forma:

Encontrar \dot{s} e ω , tal que u_1, u_2, u_3, x_e, y_e e θ_e sejam factíveis

Uma peculiaridade desta técnica, a qual pode ser um limitante de sua aplicabilidade, pode ser verificada em (4.11), donde verifica-se que o caminho necessita ser \mathbb{C}^2 -contínuo.

Apesar de solucionar o referido problema de singularidade, com esta abordagem ainda serão necessários sinais de controle elevados quando o robô estiver muito distante do caminho. Uma solução para este problema foi proposta por (Celeste *et al.* 2008) através de saturação dos sinais de controle. Outra solução, baseada num critério empírico, é conhecida como curva de pouso (Kanjawanishkul e Zell 2009), onde define-se uma curva apropriada que o robô deverá seguir em segurança até alcançar o caminho de referência.

No presente trabalho tal problema é solucionado no âmbito do problema de otimização. Mais especificamente, levando em consideração que não existe nenhum conhecimento prévio do ambiente de navegação que possibilite definir rigorosamente as restrições físicas de \mathbf{x}_e , utiliza-se a técnica para garantia de factibilidade proposta no Capítulo 3, sendo os esforços de controle físicos posteriormente saturados em valores praticáveis na plataforma experimental. Tal proposta, aplicada ao problema de seguimento de caminho, consiste numa das principais contribuições originais deste trabalho.

4.1.2 Projeto dos Controladores

Considerando-se a natureza estritamente regulatória do modelo obtido, utiliza-se o seguinte critério de desempenho para a implementação do *NMPC* nominal:

$$F(\mathbf{x}(\tau), \mathbf{u}(\tau)) = \mathbf{x}_e^T \mathbf{Q} \mathbf{x}_e + \mathbf{u}_e^T \mathbf{R} \mathbf{u}_e, \quad (4.26)$$

sendo:

Q: matriz definida positiva que pondera os desvios nos valores dos erros de estado;

R: matriz definida positiva que pondera os desvios nos valores das ações de controle.

Neste trabalho, $T_p = T_c$ e após a resolução do problema de otimização no instante t_k , as velocidades do centro de massa do robô (v , v_n , ω), calculadas de $\mathbf{u}(t_k)_{\text{ótimo}}$, são enviadas para a malha interna embarcada no robô que implementa a estratégia *PID* clássica para o controle das velocidades dos motores ($\omega_1, \omega_2, \omega_3$).

O esquema de controle proposto esta ilustrado na Figura 4.3.

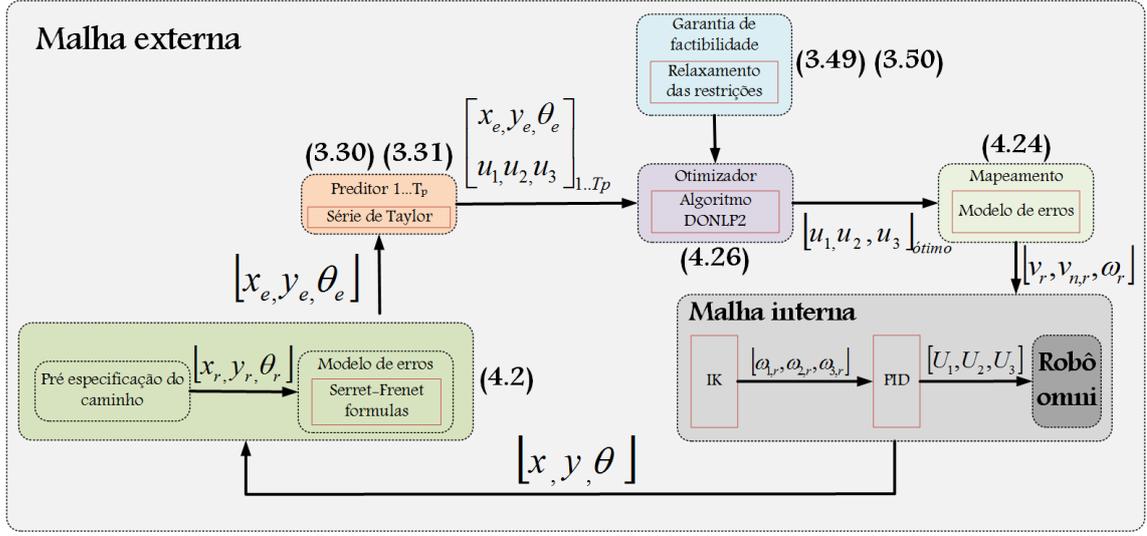


Figura 4.3: Estrutura de controle para o robô omnidirecional ¹.

4.1.3 Resultados experimentais

Primeiramente, é feita uma análise do *NMPC* nominal, para um caso ideal onde o robô se encontra próximo do caminho de referência, mais especificamente, um círculo de raio unitário. Inicialmente, define-se como primeiras coordenadas do caminho os valores $x_{ref}=1\text{m}$, $y_{ref}=0$ e $\theta_{ref}=1,57\text{rad}$ e o robô posicionado sobre este ponto, de modo que o erro inicial com relação ao caminho de referência é nulo.

As coordenadas retangulares deste caminho, já parametrizado em termos de comprimento de caminho s , são dadas pelas expressões a seguir:

$$\begin{aligned} x(s) &= \cos(s) \\ y(s) &= \sin(s) \end{aligned} \quad (4.27)$$

As referências angulares foram definidas como sendo o ângulo da tangente em cada ponto do caminho.

¹Os números no diagrama se referem a equações ao longo do texto.

Os horizontes foram fixados em $T_p = T_c = 3$ amostras e a Tabela 4.1 apresenta os outros parâmetros de sintonia utilizados neste experimento.

Os conjuntos das restrições foram definidos como segue:

$$|u_1| \leq 2; |u_2| \leq 2; |u_3| \leq 12;$$

$$|x_e| \leq 0,05 \text{ m}; |y_e| \leq 0,05 \text{ m}; |\theta_e| \leq 0,3 \text{ rad}.$$

Para $T_s = 0,05s$, foram obtidos os resultados apresentados na Figura 4.4(a). Tal figura ilustra as posturas medidas no plano xy para os primeiros 47,6s de movimento. A evolução temporal das posturas durante 60s é apresentada na Figura 4.4(b).

Os erros de estado são ilustrados na Figura 4.4(c), onde verifica-se o atendimento das tolerâncias especificadas, porém, maiores valores foram obtidos para o erro angular, já que os valores medidos para esta grandeza são mais ruidosos. Os esforços de controle são ilustrados na Figura 4.4(d). Observam-se valores elevados no início da movimentação, porém, satisfazendo as restrições do problema.

Os custos computacionais ilustrados na Figura 4.4(e) evidenciam que foram obtidos valores inferiores ao T_s especificado.

Em seguida, para confirmar uma das vulnerabilidades da abordagem original, em que não se faz nenhum tratamento de infactibilidades, o robô é posicionado em um ponto distante do caminho a ser seguido, para que sejam gerados erros de estado elevados. Mais especificamente, define-se a postura inicial do robô com sendo $x = 0$, $y = 0$ e $\theta = 0$ e as referências do experimento anterior são mantidas, ou seja, com $x_{ref} = 1m$, $y_{ref} = 0$ e $\theta_{ref} = 1,57rad$.

Obviamente, os novos valores iniciais dos erros de estado são maiores do que os limites máximos especificados pelas restrições, o que faz o otimizador acusar infactibilidade. Para a regulação destes erros, novas matrizes de ponderação são definidas e apresentadas na Tabela 4.2.

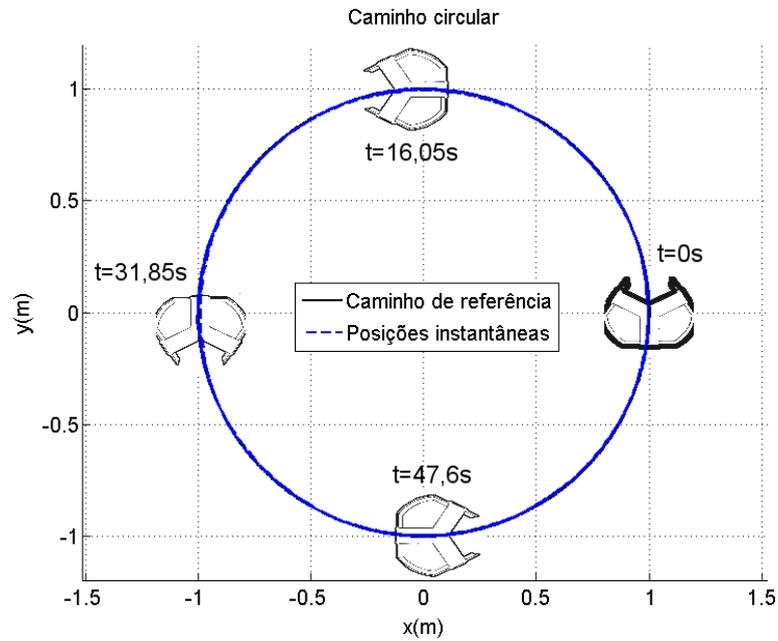
Habilitando o estágio para garantia de factibilidade, implementado através do Algoritmo 1, foram obtidos os resultados apresentados na Figura 4.5.

Q	<i>diagonal</i> (0,05; 0,05; 0,05)
R	<i>diagonal</i> (0,01; 0,01; 0,01)

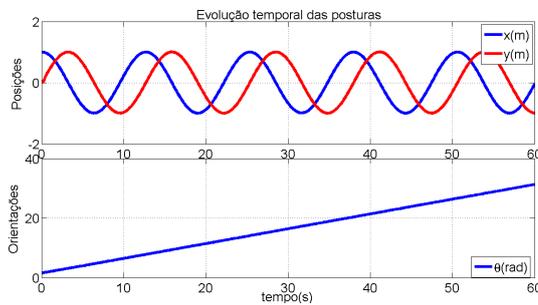
Tabela 4.1: Matrizes de ponderação do *NMPC* para seguimento de caminho circular - Robô omnidirecional - Resultado experimental - Caso Ideal.

Q	<i>diagonal</i> (0,5; 0,5; 0,5)
R	<i>diagonal</i> (100; 100; 100)

Tabela 4.2: Matrizes de ponderação do *NMPC* para seguimento de caminho circular - Robô omnidirecional - Resultado experimental - Robô distante do caminho.



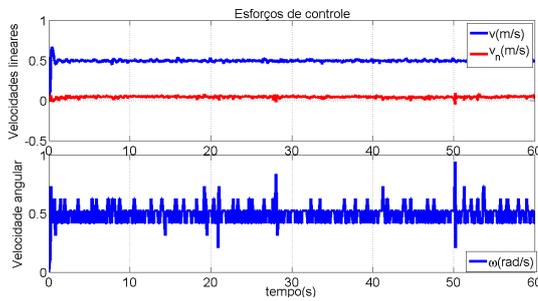
(a) Posturas no plano xy - Instantâneos por volta.



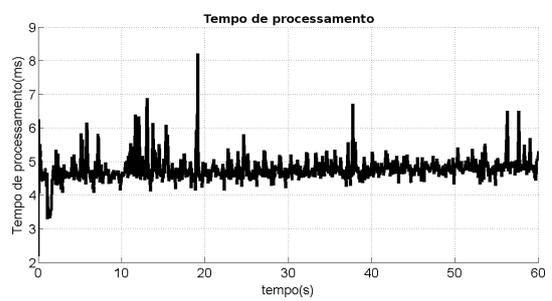
(b) Evolução temporal das posturas.



(c) Erros de estado.



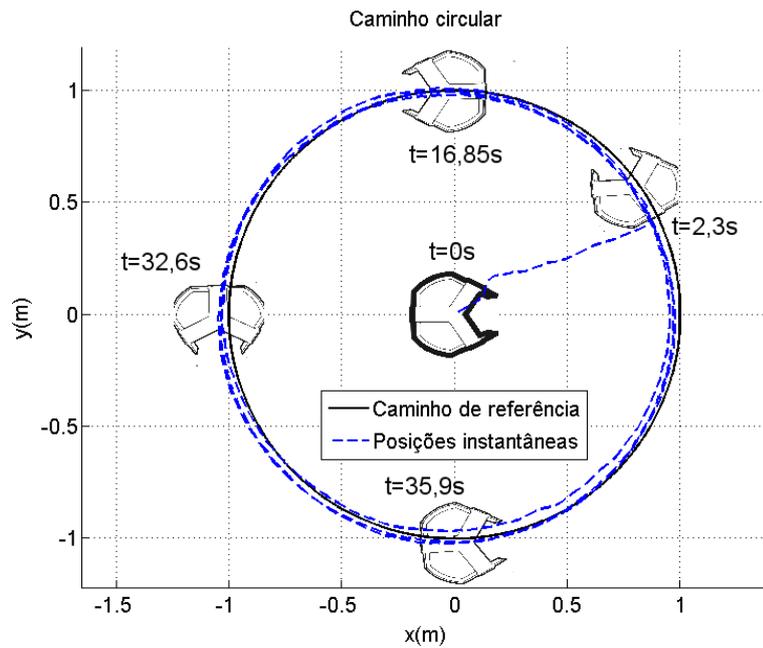
(d) Esforços de controle.



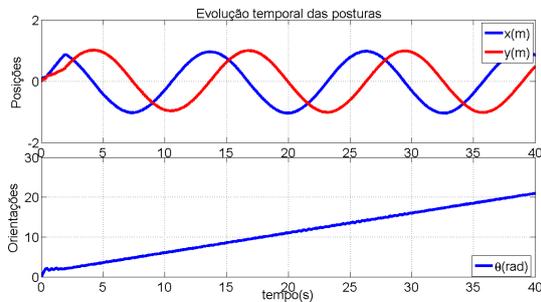
(e) Tempo de processamento.

Figura 4.4: Seguimento de caminho circular - Robô omnidirecional - Caso ideal.

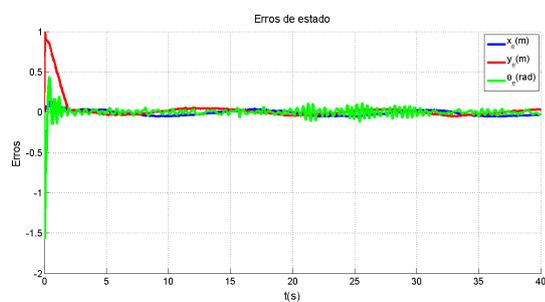
A Figura 4.5(a) ilustra as posturas no plano e a Figura 4.5(b) a sua evolução temporal. Na Figura 4.5(c) verifica-se a rápida convergência dos estados para a zona de tolerância especificada.



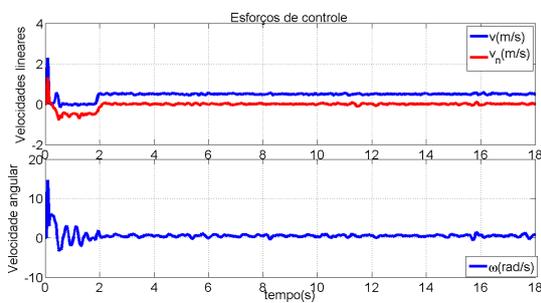
(a) Posturas no plano xy - Instantâneos por volta.



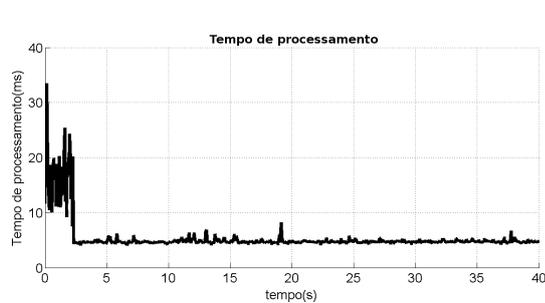
(b) Evolução temporal das posturas.



(c) Erros de estado



(d) Esforços de controle



(e) Tempo de processamento

Figura 4.5: Seguimento de caminho circular com erro de postura inicial elevado.

Verifica-se a partir da Figura 4.5(d) que os picos das ações de controle permaneceram em níveis aproximadamente iguais aos definidos para as restrições, ou seja, não importando os reais valores de u_1, u_2, u_3 ótimos fornecidos pelo otimizador. O tempo de processamento ilustrado na Figura 4.5(e) mostra que, comparado com o resultado anterior, existe maior atividade do sistema computacional para rastrear o caminho de referência distante.

Destes resultados comprova-se a validade da técnica proposta para garantia de factibilidade, uma das principais contribuições originais deste trabalho. Tal proposta permite solucionar um caso mais geral em que o robô se encontra com posturas iniciais diferentes das especificadas para o caminho a ser seguido. Este resultado viabiliza a extensão da técnica para o tratamento de caminhos variantes no tempo, caso a ser considerado para quadricópteros através de simulações e para o controle de formação através de experimentos.

4.2 Seguimento de caminhos para quadricópteros

4.2.1 Formalização do Problema

Os caminhos utilizados neste trabalho, assim como no caso bidimensional, são parametrizados em termos de comprimentos de caminho s . Além disso, tais caminhos são caracterizados por círculos concêntricos com altitude constante, o que define referências nulas para atitude (ângulos de guinada e rolagem).

Assim, as equações cinemáticas para os movimentos rotacionais e translacionais são dadas por:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t)); \\ \mathbf{x}(0) &= \mathbf{x}_0; \\ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix} &= R_o^T(\psi) \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_\psi \end{bmatrix}, \end{aligned} \quad (4.28)$$

sendo $R_o(\psi)$ a matriz de rotação ortogonal, que converte do sistema de coordenadas do mundo (x_w, y_w, z_w) para o sistema de coordenadas do corpo do quadricóptero (x_b, y_b, z_b) e vice-versa, é dada por:

$$R_o(\psi) = \begin{bmatrix} \cos(\psi) & \text{sen}(\psi) & 0 & 0 \\ -\text{sen}(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.29)$$

Neste etapa, uma velocidade de navegação u_f é definida para o quadricóptero no plano XY . Estas velocidades relacionam-se com as velocidades do centro de massa como segue:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} u_f \cos \delta \\ u_f \sin \delta \end{bmatrix}, \quad (4.30)$$

sendo que $\delta = \varphi - \psi$ representa o ângulo da direção de movimento relativo a $\{B\}$ e φ é o mesmo ângulo relativo a $\{W\}$.

Como para os robôs omnidirecionais, o modelo cinemático do quadricóptero pode ser definido em termos de um sistema de *Serret-Frenet*, movendo-se ao longo de um caminho de referência como mostrado na Figura 4.6. Nesta abordagem, as fórmulas de *Serret-Frenet* modelam o movimento de um veículo virtual, com postura $[x_r \ y_r \ z_r \ \psi_r]^T$, que deveria ser seguido pelo veículo real. O vetor de erros entre a postura do quadricóptero real e as posturas de referência relativas a $\{F\}$, é dado como segue:

$$\mathbf{x}_e = \begin{bmatrix} x_e \\ y_e \\ z_e \\ \varphi_e \end{bmatrix} = \begin{bmatrix} \cos \psi_r & \sin \psi_r & 0 & 0 \\ -\sin \psi_r & \cos \psi_r & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_r \\ y - y_r \\ z - z_r \\ \varphi - \psi_r \end{bmatrix}. \quad (4.31)$$

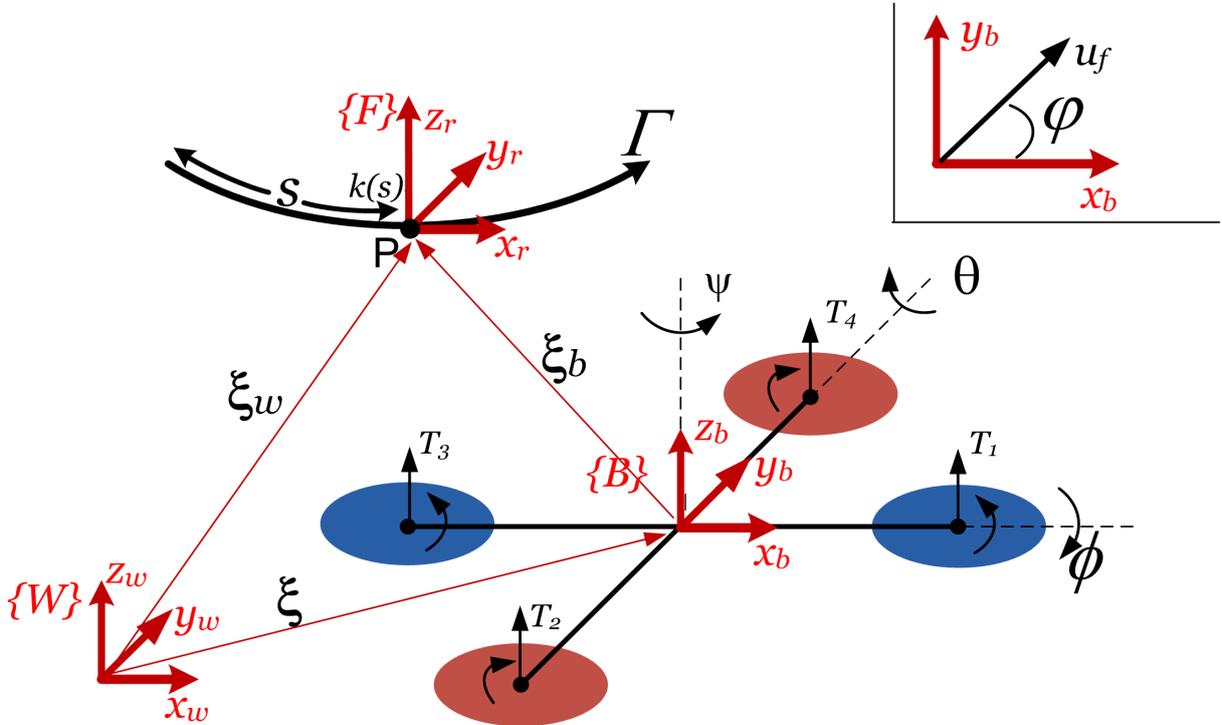


Figura 4.6: Sistemas de coordenadas e representação do problema de seguimento de caminho.

Uma análise detalhada da Figura 4.6 mostra que, usando o modelo cinemático do quadricóptero e considerando o grau de liberdade para definir a altitude e a orientação do quadricóptero, o problema de controle de seguimento de caminho é definido como um problema de controle regulatório. O modelo de erros de estados relativo ao sistema de coordenadas do caminho é dado como segue:

$$\begin{aligned}
\dot{x}_e &= y_e k(s) \dot{s} - \dot{s} + u_f \cos \varphi_e \\
\dot{y}_e &= -x_e k(s) \dot{s} - u_f \sin \varphi_e \\
\dot{z}_e &= v_z - v_{zr} \\
\dot{\psi}_e &= \omega_\psi - \omega_{\psi r},
\end{aligned} \tag{4.32}$$

sendo $k(s)$ a curvatura da seção limitada pelo comprimento de caminho s .

Objetivando a utilização do *NMPC* de tempo contínuo para este problema, definem-se os seguintes sinais de controle:

$$\begin{aligned}
u_1 &= -\dot{s} + u_f \cos \varphi_e \\
u_2 &= v_z - v_{zr} \\
u_3 &= \dot{\varphi} - k(s) \dot{s} \\
u_4 &= \omega_\psi - \omega_{\psi r}.
\end{aligned} \tag{4.33}$$

Considerando que as saídas são os próprios estados, o problema de controlar os movimentos do quadricóptero pode ser resumido da seguinte forma:

Encontrar \dot{s} , v_z e $\omega_{\psi r}$, tal que u_1 , u_2 , u_3 , u_4 , x_e , y_e , z_e e ψ_e sejam factíveis.

Similarmente ao caso do seguimento de caminho para robôs omnidirecionais, esta abordagem ainda vai requerer sinais de controle elevados quando o quadricóptero estiver muito distante do caminho, o que justifica a busca de alternativas que viabilizem implementações reais da técnica. Novamente, a proposta para garantia de factibilidade baseada no relaxamento das restrições é implementada.

4.2.2 Projeto dos Controladores

Assim como no projeto anterior, controladores de natureza estritamente regulatória são necessários. Portanto, o mesmo critério de desempenho, dado por (4.26), é utilizado, Além disso $T_p = T_c$ é mantido.

Neste caso, após a resolução do problema de otimização no instante t_k , as velocidades do centro de massa do quadricóptero ($v_x, v_y, v_z, \omega_\psi$), calculadas de $\mathbf{u}(t_k)_{\text{ótimo}}$, são enviadas para a malha interna que implementa a estratégia *PD* clássica para o controle das velocidades dos rotores ($\omega_1, \omega_2, \omega_3, \omega_4$). Esta malha interna está embarcada nos quadricópteros e foi proposta em (Corke 2011).

O esquema de controle proposto esta ilustrado na Figura 4.7.

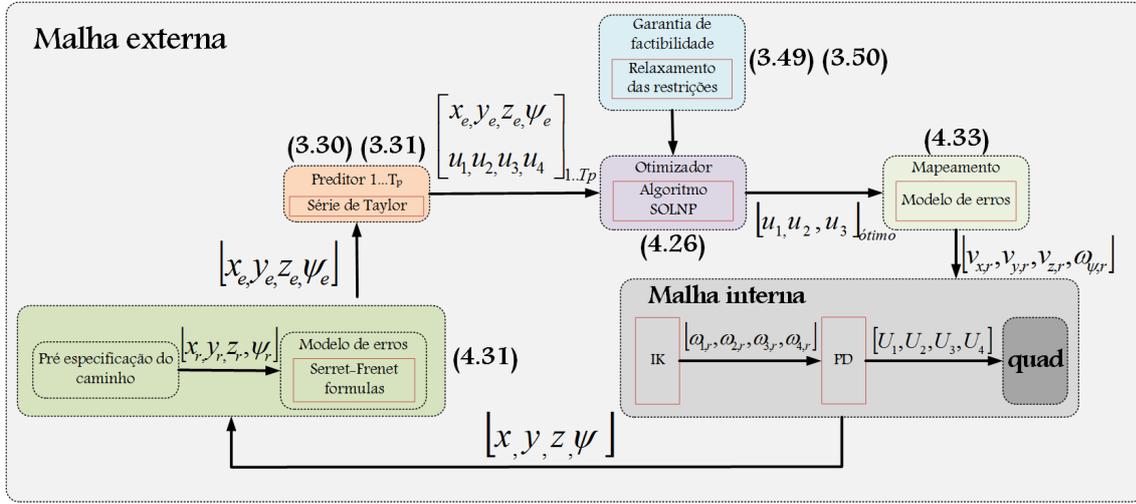


Figura 4.7: Estrutura de controle para o quadricóptero.

4.2.3 Resultados simulados

A estratégia de controle proposta foi avaliada através de simulações que fornecem evidências dos benefícios da técnica quando comparada com outras abordagens que não consideram imperfeições do ambiente de navegação e nem complexidade computacional. Para tal fim, foram considerados erros elevados para o seguimento de caminhos, fixos e variantes no tempo. Em ambos os casos as simulações foram feitas considerando um ambiente realístico onde existem rajadas de vento aleatórias atuando sobre os quadricópteros. A representação matemática destas rajadas é baseada nas especificações militares *MIL-F-8785C* (OH 1982), moduladas por uma portadora em ruído branco com 1% de potência. Na modelagem aerodinâmica do sistema, as rajadas podem ser consideradas distúrbios e neste caso considera-se 2m de comprimento e 0,5m/s de amplitude da rajada.

Todos os resultados foram adquiridos durante 100s de simulação, com $u_f = 0,5\text{m/s}$ e $T_s = 0,05\text{s}$.

Neste caso, define-se o caminho referência como um círculo de raio unitário no plano xy deslocado em z de 10m, ou seja:

$$\begin{aligned} x(s) &= \cos(s); \\ y(s) &= \text{sen}(s); \\ z &= 10. \end{aligned} \quad (4.34)$$

O ângulo de guinada de referência é o ângulo da tangente ao caminho.

Foram utilizados os parâmetros apresentados na Tabela 4.3 e $T_p = T_c = 3$ amos-tras, como horizontes.

Considerando a utilização direta da técnica para garantia de factibilidade, a tolerância para os erros lineares ($x_{e_{tol}}$, $y_{e_{tol}}$ e $z_{e_{tol}}$) foram fixadas em 5cm e para o erro

angular ($\psi_{e_{tol}}$) 0, 1rad. Como parte da proposta, as componentes de \mathcal{U}_i foram definidas suficientemente altas para permitir factibilidade do problema de otimização e posterior saturação em níveis praticáveis. Tais níveis foram definidos como sendo $\pm 2\text{m/s}$ para as velocidades lineares e $\pm 12\text{rad/s}$ para velocidade angular.

Para gerar altos erros iniciais, o quadricóptero foi posicionado distante do caminho de referência. Define-se a posição e a orientação iniciais como sendo $\xi = [0, 0, 0]$ e $\Phi = [\frac{\pi}{2}, 0, 0]$, respectivamente.

A Figura 4.8 ilustra os resultados obtidos no espaço tridimensional. Vistas dos planos XZ e YZ são apresentadas na Figura 4.8(b) e a evolução temporal das posturas são apresentadas na Figura 4.8(c).

Verifica-se o atendimento dos objetivos de controle, desde que houve regulação ao longo do caminho de referência mesmo com um elevado erro inicial para a postura e na presença dos distúrbios causados pelas rajadas de vento. A Figura 4.8(d) mostra a forma de onda deste distúrbio e a Figura 4.8(e) mostra como as velocidades foram afetadas.

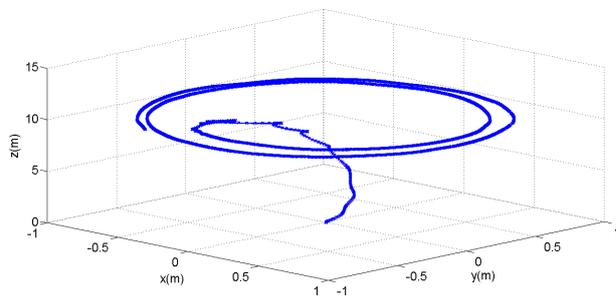
A Figura 4.8(f) apresenta os erros de seguimento de caminho, e, verifica-se que durante as regulações, todos os estados do quadricóptero são afetados, comportamento justificado pela sintonia escolhida, já que foi dada maior importância à regulação de y_e . Em todos os casos é verificada a convergência dos estados para as tolerâncias especificadas.

Os esforços de controle da malha externa são mostrados na Figura 4.8(g). Verifica-se que não foram alcançados os limites de saturação especificados para as variáveis perturbadas, indicando a possibilidade de regulação de erros mais elevados. A Figura 4.8(h) mostra que os esforços de controle na malha interna foram mantidos em valores compatíveis com as especificações do quadricóptero utilizado, mesmo durante a regulação dos erros iniciais elevados e na presença de turbulência.

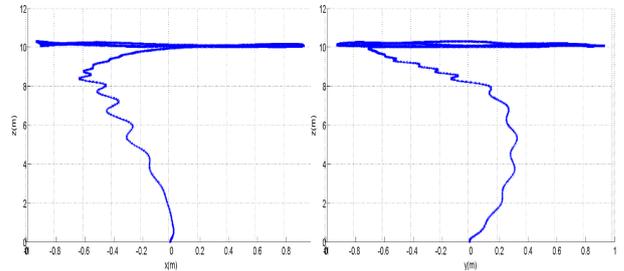
Nesta etapa não foi feita avaliação de esforço computacional, uma vez que para a simulação em *Matlab/Simulink* foram utilizados códigos interpretados, sem garantia de execução de tarefas em tempo real. As medidas fornecidas por funções nativas para medição de tempo de processamento são imprecisas, sendo impossível uma avaliação realística do custo computacional da técnica. Experiências adquiridas com implementações práticas do seguimento de caminho com garantia de factibilidade indicam que a eficiência computacional será mantida, haja vista a similaridade entre os modelos.

Tabela 4.3: Matrizes de ponderação - Simulação 1.

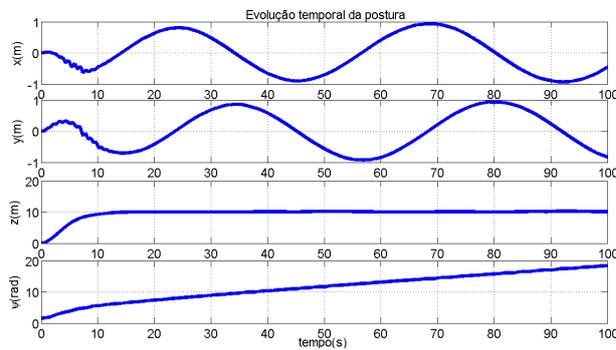
Q	$diagonal(100;1000;0,1;10)$
R	$diagonal(0,01;0,01;0,01;0,01)$



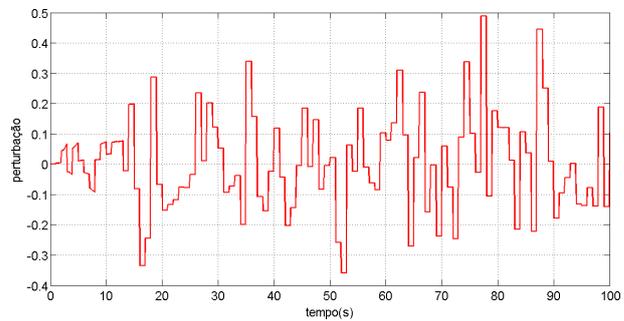
(a) Visão 3D.



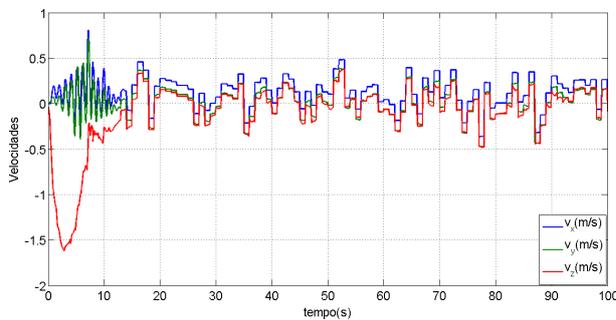
(b) Visões 2D do fundo (X/Z) e do lado (Y/Z).



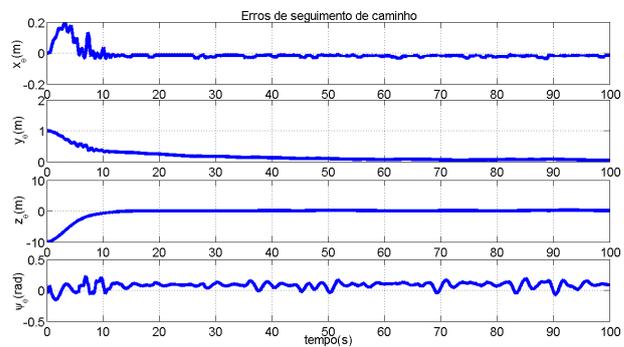
(c) Evolução temporal das posturas.



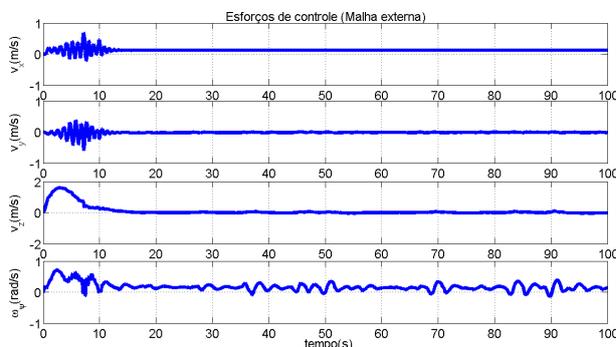
(d) Forma de onda da turbulência.



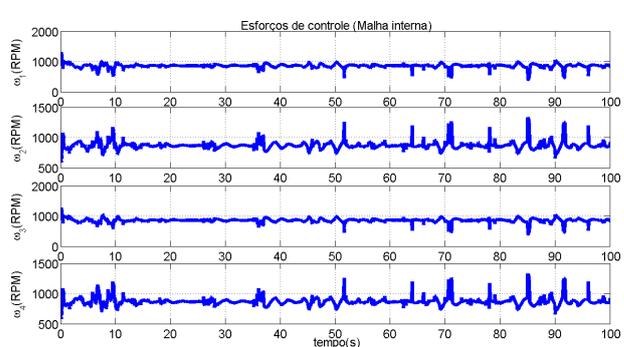
(e) Velocidades perturbadas.



(f) Erros de seguimento de caminho.



(g) Esforços de controle - Malha externa



(h) Esforços de controle - Malha interna

Figura 4.8: Seguimento de caminhos para quadricópteros: Simulação 1 - Caminho circular fixo.

Simulação 2 - Erros iniciais elevados e caminho variante no tempo

Esta simulação tem o objetivo de avaliar o desempenho da técnica proposta para seguimento de caminho em casos de mudanças abruptas nos parâmetros do caminho de referência, ou seja, são aplicadas perturbações diretamente aos erros de seguimento além daquelas já causadas pelo ambiente. Além disso, para ilustrar melhor os efeitos do relaxamento das restrições, a magnitude dos limites de saturação das velocidades foram reduzidos para 0,5. Tal situação poderia ocorrer, por exemplo, por limitações dos atuadores de determinado quadricóptero. Nesta etapa será analisado também o comportamento da função objetivo com e sem a utilização da técnica para garantia de factibilidade.

As perturbações são aplicadas em instantes de tempo bem definidos através das seguintes alterações nos parâmetros dos caminhos:

1. $T_1 = 25s \rightarrow z_r = 5m$ e $\rightarrow r = 2m$; A altitude de referência é reduzida e o raio do caminho circular aumentado.
2. $T_2 = 50s \rightarrow z_r = 7,5m$ e $\rightarrow r = 1,5m$; A altitude de referência é aumentada e o raio do caminho circular reduzido.

Considerando os novos limites de saturação impostos e os resultados obtidos na simulação 1, define-se os parâmetros apresentados na Tabela 5.4.

Os critérios para a escolha destes parâmetros são baseados na redução do tempo de regulação de cada estado. Objetiva-se que tal regulação ocorra dentro dos limites de tempo especificados.

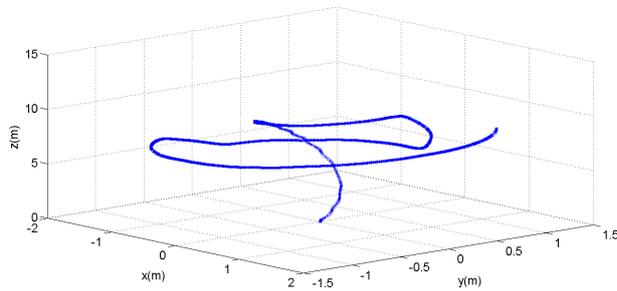
A Figura 4.9(a) ilustra os resultados obtidos no espaço tridimensional. Vistas dos planos XZ e YZ são apresentadas na Figura 4.9(b) e a evolução temporal das posturas são apresentadas na Figura 4.9(c). Verifica-se o atendimento dos objetivos de controle em cada passo de execução através de completa regulação ao longo dos caminhos.

A Figura 4.9(d) apresenta os erros de seguimento de caminho, e, verifica-se regulação mesmo na presença dos dois tipos de perturbação sendo também verificada, neste caso, convergência de todos os estados para os limites de tolerância especificados.

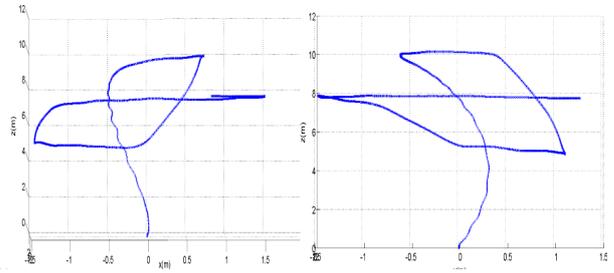
Os esforços de controle da malha externa são mostrados na Figura 4.9(e). Verifica-se que mesmo atingido os limites de saturação para as variáveis com maiores índices de perturbação, as outras ações permanecem em limites factíveis. A Figura 4.9(f) mostra que os esforços de controle na malha interna também foram mantidos em valores compatíveis com as especificações do quadricóptero utilizado, mesmo durante a regulação dos distúrbios e na presença de turbulência.

Tabela 4.4: Matrizes de ponderação - Simulação 2.

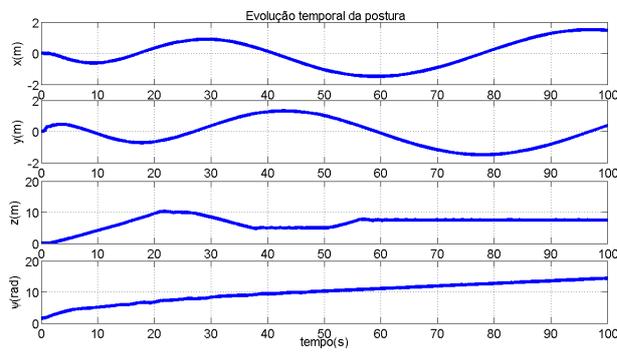
Q	$diagonal(4000;10000;0,01;40)$
R	$diagonal(0,001;0,001;0,001;0,001)$



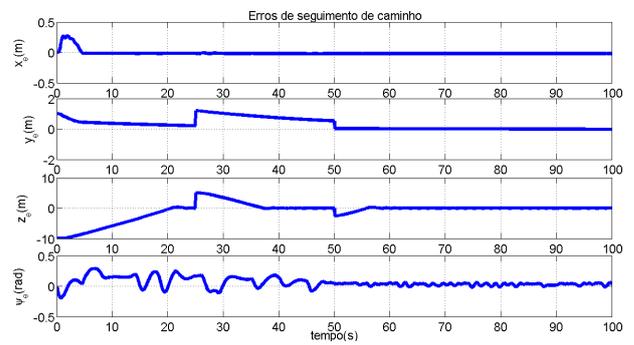
(a) Visão 3D.



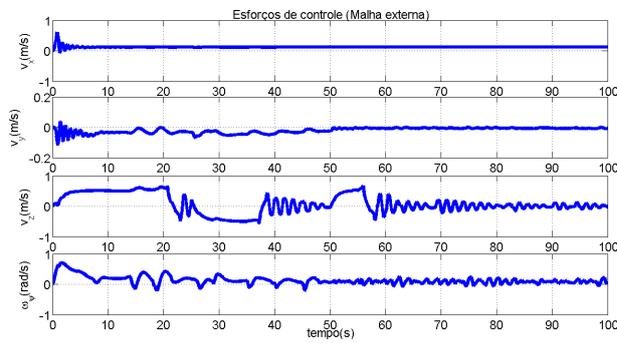
(b) Visões 2D da frente (X/Z) e do lado (Y/Z).



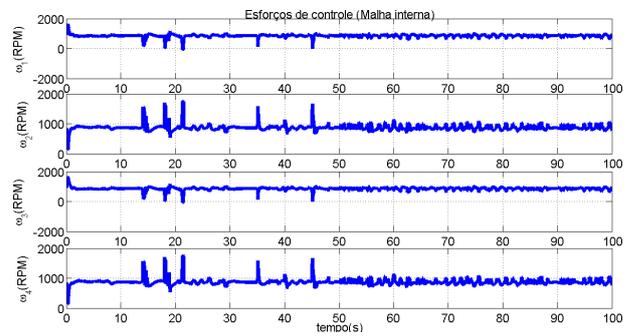
(c) Postura do quadricóptero.



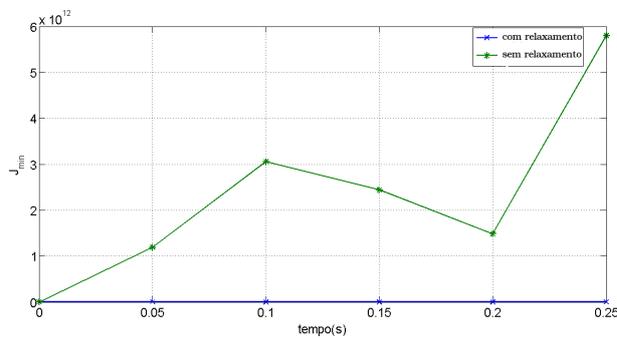
(d) Erros de seguimento de caminho.



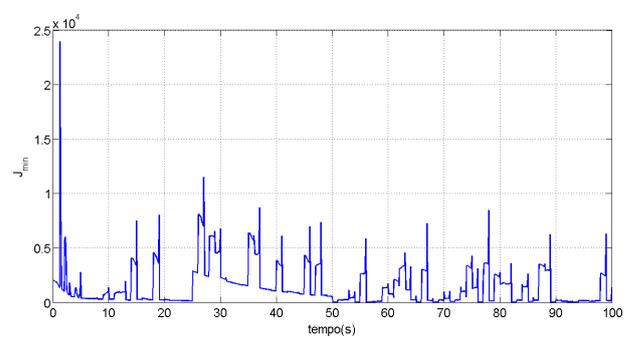
(e) Esforços de controle - Malha externa



(f) Esforços de controle - Malha interna



(g) Comparação das funções objetivo



(h) Análise da função objetivo

Figura 4.9: Seguimento de caminhos para quadricópteros: Simulação 2 - Caminhos circulares variantes no tempo.

Para ilustrar o comportamento da função objetivo, a Figura 4.9(g) mostra uma comparação com um caso em que não se utiliza o relaxamento das restrições para garantia de factibilidade. Observa-se que neste caso, logo nos instantes iniciais, a função objetivo não converge e o algoritmo de otimização acusa infactibilidade. A Figura 4.9(h) ilustra o comportamento da função objetivo ao longo de toda a simulação, quando se utiliza a técnica proposta, sendo verificada completa convergência.

4.3 Controle servo visual

As abordagens clássicas para seguimento de caminho possuem como premissa a existência de caminhos de referência fixos e pré estabelecidos. Isto limita bastante a gama de aplicações, visto que, em diversas situações práticas da navegação robótica os ambientes são dinâmicos e não estruturados. A definição prévia de caminhos contemplando este cenário pode ser uma tarefa bastante complexa.

Com a popularização de câmeras de baixo custo e o desenvolvimento de algoritmos eficientes para visão computacional, a utilização de informações visuais viabilizou a concepção de sensores absolutos de postura, que, em conjunção com outros sensores embarcados, possibilitam a criação de mecanismos eficientes de localização.

Dentre as principais técnicas para controle servo visual destacam-se o *PBVS* e *IBVS*, com esta última mais eficiente do ponto de vista computacional devido a ausência de algoritmos para a estimação da posição (Chaumette e Hutchinson 2006). As ideias elementares por detrás destas técnicas servem de base para a concepção da técnica de seguimento de caminhos visuais, sendo em todos os casos, a representação de pontos no mundo através de pontos no plano da imagem feita através da projeção perspectiva.

4.3.1 Transformação perspectiva

Seja uma câmera C , com distância focal f ; o ponto P de coordenadas (X, Y, Z) no sistema de coordenadas da câmera (P_W^C); e sua projeção p de coordenadas (x, y) no plano da imagem (Figura 4.10).

Usando relações geométricas e conhecimentos básicos de óptica, é possível deduzir que a projeção do ponto P no plano das imagens, p , é dada como segue:

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{f}{Z} & 0 & 0 \\ 0 & \frac{f}{Z} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (4.35)$$

Observa-se a que foi introduzido um componente ao vetor p , que mesmo não tendo significado físico permite que a equação seja escrita na forma homogênea.

A projeção pode também ser representada em termos dos pixels da imagem resultante. Neste caso, como mostrado na Figura 4.11, o sistema de coordenadas adotado admite apenas valores positivos de coordenadas (u, v) , e tem como ponto principal

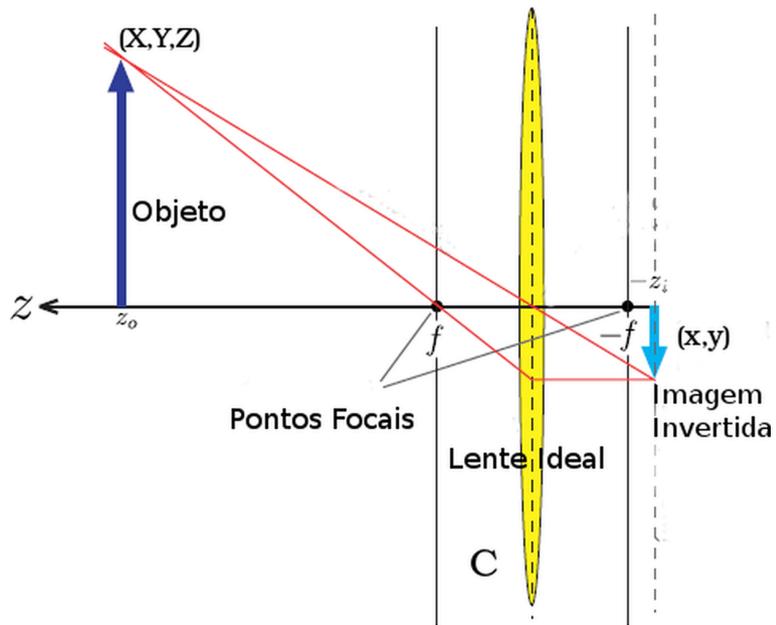


Figura 4.10: Formação geométrica das imagens. Adaptado de: (Corke 2011), p. 252.

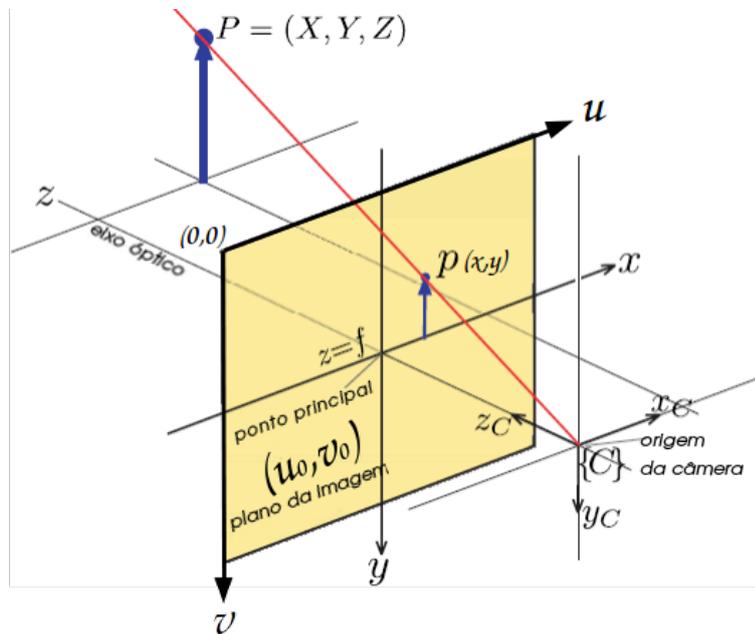


Figura 4.11: Representação da Projeção Perspectiva. Adaptado de: (Corke 2011), p. 252.

(u_0, v_0) . A expressão que define a projeção p em termos de pixels (\hat{p}) é dada como segue:

$$\hat{p} = \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} = \begin{bmatrix} \frac{fX}{\rho_u Z} - u_0 \\ \frac{fY}{\rho_v Z} - v_0 \end{bmatrix}. \quad (4.36)$$

sendo que ρ_u e ρ_v representam a largura e a altura de um pixel, respectivamente, e \hat{u} e \hat{v} representam as coordenadas em relação à origem do plano da imagem, e não ao ponto principal.

Por fim, para representar corretamente o ponto P , é preciso considerar a posição e orientação da câmera em relação ao ponto. Para isso deve-se expressar o ponto P no sistema de coordenadas do mundo. Assim, uma transformação de coordenadas é introduzida na relação anterior, obtendo-se:

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{f}{\rho_u Z} & 0 & u_0 \\ 0 & \frac{f}{\rho_v Z} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} (T_C^O)^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (4.37)$$

Uma análise mais detalhada pode ser encontrada em (Corke 2011), capítulo 11.

Neste trabalho, a transformação perspectiva é também utilizada para o cálculo de estimativas da curvatura do caminho visual.

4.3.2 Controle Visual Baseado em Imagem (*IBVS*)

O Controle Visual Baseado em Imagem pode ser definido como o uso de características visuais, extraídas de dados de visão computacional, como entrada de uma malha de controle de postura de um sistema dinâmico, sem que seja estimada a sua postura real, dessa forma o controle se dá diretamente no plano visual.

Para melhor elucidação, considera-se a situação ilustrada na Figura 4.12, onde a câmera C , localizada no ponto O_c , se move em direção ao ponto P , e se orienta de acordo com um alvo localizado no ponto P_A , ou seja, a postura do robô é corrigida

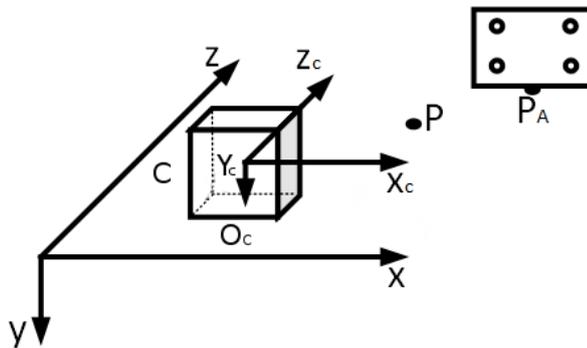


Figura 4.12: Sistemas de coordenadas utilizadas para o *IBVS*.

de acordo com o erro entre as características da imagem inicial e da imagem final do alvo.

De maneira geral, para k parâmetros de características visuais, associa-se o vetor $\kappa = [\kappa_1 \dots \kappa_k] \in \mathcal{K} \subseteq \mathbb{R}^k$ de parâmetros correspondentes, onde \mathcal{K} é o espaço de parâmetros de características da imagem. A dimensão de \mathcal{K} depende das características adquiridas, considerando pontos, por exemplo, cada característica possui dois parâmetros, um para cada eixo do plano da imagem.

Define-se a função erro de posicionamento $\mathbf{e} : \mathcal{K} \mapsto \mathbb{R}^l$, onde $l \leq k$ é o número de graus de liberdade do robô. O objetivo de regular κ até os parâmetros de características desejados κ_d é atingido quando $\mathbf{e}(\kappa) = \kappa - \kappa_d = 0$, controlando implicitamente os movimentos da câmera. A relação entre as velocidades das características e a velocidade espacial da câmera \mathbf{v}_c é dada por:

$$\dot{\kappa} = \mathbf{L}_\kappa \mathbf{v}_c \quad (4.38)$$

sendo $\mathbf{L}_\kappa \in \mathbb{R}^{k \times 6}$ a matriz de iteração relativa a κ (Jacobiano visual).

Para alvos estáticos tem-se:

$$\dot{\mathbf{e}} = \mathbf{L}_s \mathbf{v}_c \quad (4.39)$$

sendo $\mathbf{L}_s = \mathbf{L}_\kappa$

Considerando um decaimento exponencial para os erros ($\dot{\mathbf{e}} = -\lambda \mathbf{e}$) as velocidades de referência para a câmera são calculadas como segue:

$$\mathbf{v}_c = -\lambda \mathbf{L}_s^+ \mathbf{e}, \quad (4.40)$$

onde \mathbf{L}_s^+ é a pseudo inversa da matriz de iteração.

Em alguns casos, como para seguimento de caminhos visuais, onde de maneira geral existem muitos parâmetros por característica, o cálculo desta pseudo-inversa em tempo de execução pode tornar o custo computacional bastante elevado, prejudicando o desempenho, mesmo comparativamente ao *PBVS*, motivando a busca por soluções alternativas para o controle visual.

4.4 Seguimento de caminhos visuais

4.4.1 Formalização do Problema

Objetivando a utilização de características visuais simplificadas, para manutenção dos requisitos de complexidade computacional, utiliza-se a modelagem proposta por (Coulaud *et al.* 2006) para robôs diferenciais. Neste caso, a partir de uma velocidade de navegação v , controla-se a orientação do robô para que o caminho a ser seguido fique centralizado no campo visual da câmera. Para tal fim, a mesma é posicionada com uma posição de offset $[x_c, y_c, z_c]$ e com uma orientação espacial $[\theta_{c_x}, \theta_{c_y}, \theta_{c_z}]$ relativa à postura do robô.

Esta caracterização do seguimento de caminho visual é particularmente aplicável em situações em que o sensor visual está posicionado de frente para o caminho a ser

seguido e com campo de visão limitado. Tal caminho é definido através da extração de parâmetros de características visuais presentes no ambiente de navegação de acordo com objetivos específicos, por exemplo, seguimento de linhas de sinalização em faixas de domínio de rodovias.

Define-se *horizonte* como uma linha reta no solo, paralela a eixo traseiro do robô a uma distância h conforme ilustrado na Figura 4.13. Considera-se o ponto p_r como o ponto de interseção entre o horizonte e o caminho a ser seguido. Se a curvatura do caminho não é muito elevada, este ponto é único, caso contrário p_r será o ponto mais próximo do eixo longitudinal do robô. Define-se z como a distância entre p_r e o eixo longitudinal do robô.

Tanto o horizonte quanto z são características cujos parâmetros podem ser facilmente extraídos das imagens, visto que o horizonte corresponde a uma linha de pixels a uma distância $d(h)$ do ponto principal enquanto z é proporcional ao número de pixels entre o centro do horizonte e p_r . A Figura 4.14 apresenta uma imagem com estas características.

A princípio deve-se considerar que distorções nas imagens podem ser desconsideradas e, como h é constante, para obter z a partir das imagens, existe um único coeficiente de proporcionalidade obtido na fase de calibração do sistema visual.

Como feito nas seções anteriores define-se um sistema de *Serret-Frenet* (\vec{T}, \vec{N}) para caracterizar os movimentos de um veículo virtual posicionado em p_r .

Analisando a Figura 4.14, obtém-se:

$$p_r(s(t)) = p(t) + hx_r(\theta(t)) - z(t)y_r(\theta(t)). \quad (4.41)$$

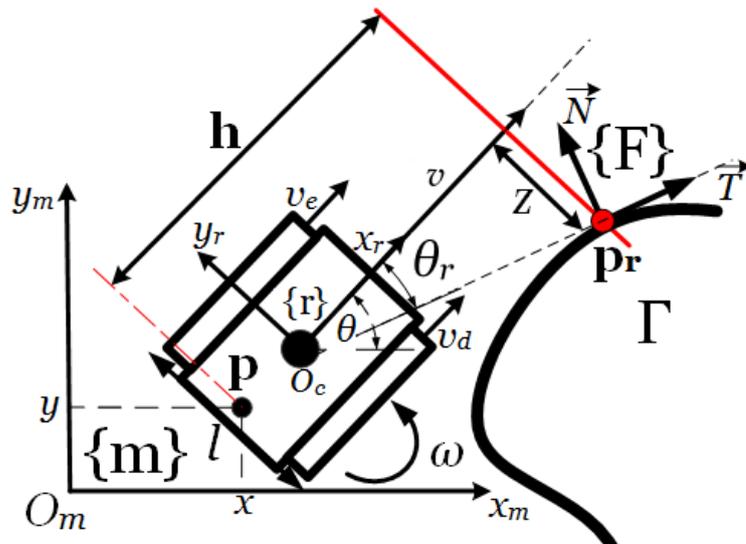


Figura 4.13: Seguimento de caminhos visuais.

Derivando a expressão anterior com relação ao tempo, obtém-se:

$$\dot{s}_T \vec{T}(s) + \dot{s}_N \vec{N}(s) = \dot{x} \vec{x}_r(\theta(t)) + \dot{y} \vec{y}_r(\theta(t)) + h \dot{\theta} \vec{y}_r(\theta(t)) + z(t) \dot{\theta} \vec{x}_r(\theta(t)) - \dot{z} \vec{y}_r(\theta(t)). \quad (4.42)$$

Sabendo-se que $\dot{s}_T = \dot{s}$, $\dot{s}_N = 0$, $\dot{y} \vec{y}_r = 0$ e omitindo as dependências angulares e temporais, obtém-se:

$$\dot{s} \vec{T}(s) = \dot{x} \vec{x}_r + h \dot{\theta} \vec{y}_r + z \dot{\theta} \vec{x}_r - \dot{z} \vec{y}_r. \quad (4.43)$$

A relação entre o sistema de coordenadas do robô e o sistema de *Serret-Frenet* é dada como segue:

$$\begin{bmatrix} \vec{x}_r \\ \vec{y}_r \end{bmatrix} = \begin{bmatrix} \cos \theta_r & \sin \theta_r \\ -\sin \theta_r & \cos \theta_r \end{bmatrix} \begin{bmatrix} \vec{T} \\ \vec{N} \end{bmatrix} \quad (4.44)$$

Projetando esta expressão no sistema de *Serret-Frenet* e substituindo (2.13) obtém-se:

$$\dot{z} = \omega h + (\omega z + v) \tan \theta_r; \quad (4.45)$$

$$\dot{s} = \frac{v + \omega z}{\cos \theta_r}. \quad (4.46)$$

Visto que $\dot{\theta}_r = \omega - \dot{c}(s)$, tem-se:

$$\dot{z} = \omega h + (\omega z + v) \tan(\theta_r); \quad (4.47)$$

$$\dot{\theta}_r = \omega - c(s) \frac{(v + \omega z)}{\cos \theta_r}, \quad (4.48)$$

com $c(s)$ satisfazendo:

$$\dot{c} = \frac{dc}{ds} \frac{(v + \omega z)}{\cos \theta_r}. \quad (4.49)$$

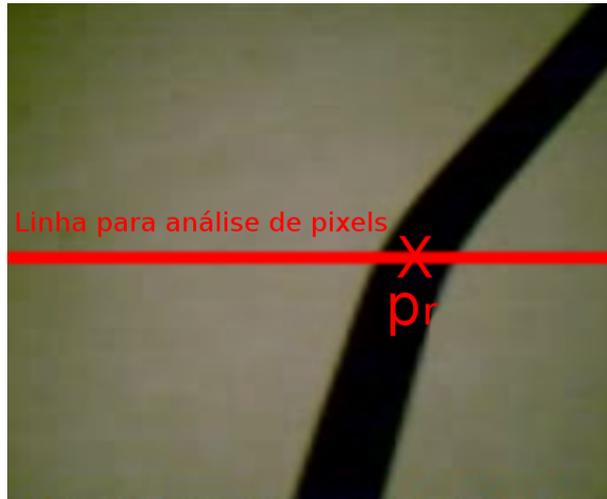


Figura 4.14: Imagem do caminho a ser seguido.

Para $\dot{z} = \dot{\theta}_r = 0$ tem-se os seguintes valores de ω :

$$\omega = \frac{cv}{\cos \theta_r + cz}; \quad (4.50)$$

$$\omega = \frac{v \tan \theta_r}{z \tan \theta_r + h}. \quad (4.51)$$

Igualando-se (4.50) a (4.51), obtém-se:

$$\theta_r = \text{sen}^{-1}(hc). \quad (4.52)$$

Deste modo, verifica-se que uma condição de equilíbrio pode ser alcançada apenas com ajustes em θ_r . Objetivando-se formatar o problema de seguimento de caminhos visuais no âmbito de um problema de controle ótimo, define-se $\mathbf{u}_e = \dot{\theta}_r$, como entrada de controle para o *NMPC*. Assim, tem-se o seguinte modelo para o seguimento de caminhos visuais:

$$\mathbf{x}_e = \begin{bmatrix} \dot{z} \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \omega h + (\omega z + v) \tan(\theta_r) \\ \mathbf{u}_e \end{bmatrix}. \quad (4.53)$$

Considerando que as saídas são os próprios estados, o problema de seguimento de caminhos visuais para robôs diferenciais pode ser resumido da seguinte forma:

Encontrar ω , tal que u_e , z e θ_r sejam factíveis.

Etapa particularmente importante para o sucesso da implementação desta estratégia é o cálculo da curvatura do caminho visual. Sabe-se que tal grandeza é bastante sensível às medidas de z e θ_r , que por sua vez são obtidos de maneira aproximada sem requisitos muitos rigorosos na etapa de calibração do sistema visual.

Uma forma de obter esta curvatura com precisão razoável é obter as coordenadas de 2 pontos vizinhos ao ponto principal e calcular o raio do círculo que passa por esses três pontos. Obviamente, a precisão dos movimentos dos robôs ao longo do caminho visual será de grande relevância para cálculo das curvaturas visuais, conforme será ilustrado com os resultados experimentais.

4.4.2 Projeto dos Controladores

Para solucionar o problema em questão, consideram-se referências constantes para Z e θ_r , o que permite, sem perda de generalidade, a utilização da mesma estrutura para $F(\mathbf{x}(\tau), \mathbf{u}(\tau))$ (4.26) apresentada anteriormente. Neste caso, devido às limitações práticas do sistema visual, especialmente no que se refere ao cálculo da curvatura visual, uma abordagem para garantia de estabilidade se torna necessária. A abordagem escolhida baseia-se na adição de um termo de penalidade e de restrições terminais ao problema de otimização, ou seja, é utilizado o algoritmo *NMPC* formalizado de (3.51) até (3.56).

A seguinte função de Lyapunov é escolhida como termo de penalidade:

$$V(\mathbf{x}_e(t + T_p)) = \frac{1}{2} \mathbf{x}_e(t + T_p)^T \mathbf{P} \mathbf{x}_e(t + T_p) \quad (4.54)$$

sendo $\mathbf{x}_e(t + T_p) = [z_T, \theta_{r_T}]$ o estado terminal e \mathbf{P} definida positiva.

Considerando a ação de controle terminal u_e^T , a seguinte condição deve ser satisfeita:

$$\dot{V}(\mathbf{x}_e(t)) + F(\mathbf{x}_e(t), \mathbf{u}_e(t)) \leq 0. \quad (4.55)$$

Definindo-se $u_e^T = -\alpha \theta_{r_T}$, com $\alpha > 0$ e os seguintes formatos para as matrizes de ponderação:

$$\begin{aligned} Q &= \text{diagonal}(q_{11}, q_{22}); \\ R &= r_{11}; \\ P &= \text{diagonal}(p_{11}, p_{22}), \end{aligned}$$

de (4.54) em (4.55), e utilizando (4.53) tem-se a seguinte condição de estabilidade:

$$\begin{aligned} & p_{11} z_T \dot{z}_T + p_{22} \theta_{r_T} \dot{\theta}_{r_T} + F(t + T_p) \\ &= p_{11} z_T (\omega H + (\omega z_T + v) \tan \theta_{r_T}) + p_{22} \theta_{r_T} u_{1L} + F(t + T_p) \quad (4.56) \\ &= p_{11} z_T (\omega H + (\omega z_T + v) \tan \theta_{r_T}) + p_{22} \theta_{r_T} u_{1L} + q_{11} z_T^2 + q_{22} \theta_{r_T}^2 + r_{11} u_1^2 \\ &= p_{11} z_T (\omega H + (\omega z_T + v) \tan \theta_{r_T}) - \alpha p_{22} \theta_{r_T}^2 + q_{11} z_T^2 + q_{22} \theta_{r_T}^2 + r_{11} \alpha^2 \theta_{r_T}^2 \\ &= q_{11} z_T^2 + \theta_{r_T}^2 (-\alpha p_{22} + q_{22} + r_{11} \alpha^2) + p_{11} z_T (\omega H + (\omega z_T + v) \tan \theta_{r_T}). \end{aligned}$$

Deste modo, tem-se a seguinte condição para a determinação dos pesos das matrizes de ponderação:

$$\alpha p_{22} - q_{22} - r_{11} \alpha^2 \geq q_{11}. \quad (4.57)$$

A região terminal dos estados Ω é definida da seguinte forma:

$$|\theta_{r_T}| \geq |z_T|; \quad (4.58)$$

$$p_{11} z_T (\omega H + (\omega z_T + v) \tan \theta_{r_T}) \leq 0. \quad (4.59)$$

Tal região terminal é limitada pelas entradas de controle do seguinte modo:

$$\omega = \frac{cv - \alpha \theta_{r_T} \cos \theta_{r_T}}{\cos \theta_{r_T} - cz_T}, \quad (4.60)$$

sendo ω restrita da seguinte forma:

$$\omega_{min} \leq \omega \leq \omega_{max}. \quad (4.61)$$

A lei de controle proposta por (Coulaud *et al.* 2006) consiste em encontrar ω proporcional a Z , ou seja, não leva em consideração explicitamente os limites físicos das variáveis de estado. Na próxima seção será fornecida uma comparação entre as duas propostas.

Assim como nos casos anteriores, após a resolução do problema de otimização no instante t_k , as velocidades do centro de massa do robô (v e ω), calculadas de $\mathbf{u}(t_k)_{\text{ótimo}}$, são enviadas para a malha interna, embarcada no robô diferencial, que implementa a estratégia *PID* clássica para o controle das velocidades dos motores (ω_l, ω_r).

O esquema de controle proposto esta ilustrado na Figura 4.15.

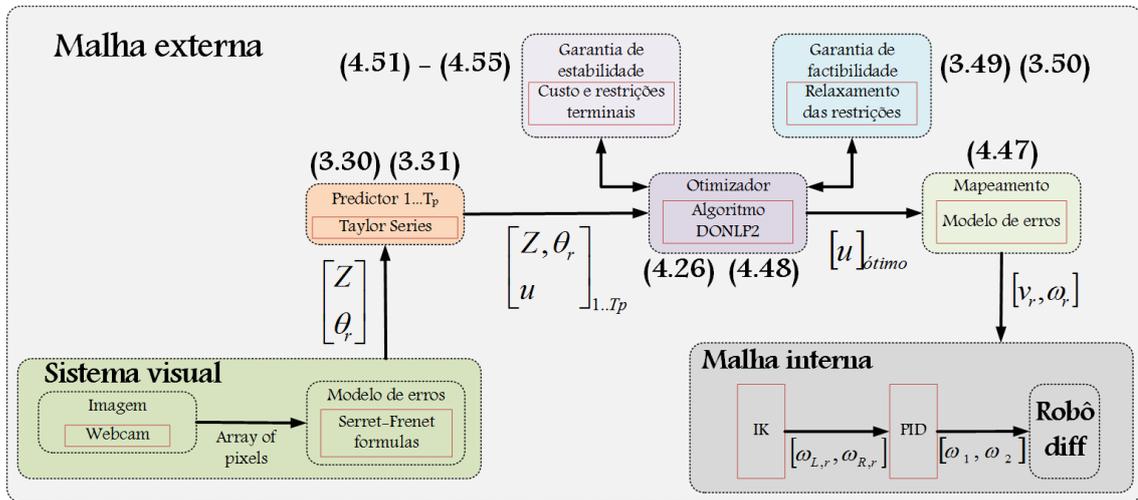


Figura 4.15: Estrutura de controle baseada em visão para o robô diferencial.

4.4.3 Resultados

Nesta etapa, a abordagem proposta para seguimento de caminhos visuais é validada inicialmente através de simulação para o modelo ideal. Em seguida, resultados de experimentos com dois tipos diferentes de robôs diferenciais comerciais são apresentados e discutidos. Conforme elucidado na subseção anterior, o caminho a ser seguido deve estar no campo de visão da câmera, o que limita as possibilidades para definição da postura inicial do robô. Deve-se destacar que, devido à projeção perspectiva, pequenas perturbações no posicionamento do robô, geram grandes perturbações nas imagens, justificando ainda mais a necessidade de utilização da técnica proposta para garantia de factibilidade.

Simulações

Para as simulações, a câmera é posicionada sobre o robô com uma postura de *offset* [$x_c = 0,03\text{m}$, $y_c = 0\text{m}$, $z_c = 0,1\text{m}$, $\theta_{c_y} = \frac{\pi}{4}\text{rad}$]. Utiliza-se uma distância focal $f = 0,01\text{m}$ e o caminho a ser seguido é um círculo desenhado no plano xy .

O robô possui uma postura inicial [$x_r = 1\text{m}$, $y_r = 0\text{m}$, $\theta_y = \frac{\pi}{2}\text{rad}$] relativa ao sistema de coordenadas do mundo. Esta postura inicial gera os seguintes valores iniciais para os erros de seguimento de caminho: [$Z = -0,0013\text{m}$, $\theta_r = -0,068\text{rad}$].

Inicialmente, são realizadas simulações para um caso ideal onde não existem perturbações, em seguida perturbações do tipo degrau são aplicadas às variáveis de estado.

Todos os resultados foram adquiridos durante 150s de simulação, com $v = 0,05\text{m/s}$, período de amostragem $T_s = 0,05\text{s}$ e os horizontes de $T_p = T_c = 3$ amostras foram mantidos.

Simulação 1 - Caso ideal.

Para esta simulação foram utilizadas as matrizes de ponderação apresentadas na Tabela 4.5.

A tolerâncias foram mantidas em 5cm para os erros lineares e 0,1rad para os erros angulares. Estes valores foram adicionados às restrições definidas em \mathcal{X} . Os valores limitantes de \mathcal{U} foram definidos de acordo com as características dos robôs comerciais disponíveis, ou seja, $|u_e| \leq 0,1$.

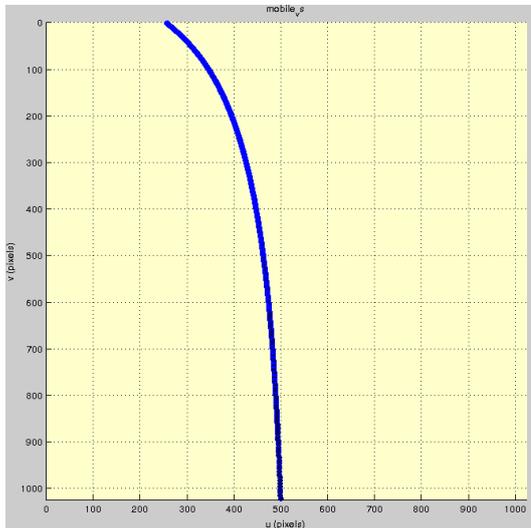
A Figura 4.16(a) mostra a imagem inicial obtida pela câmera, e, desde que o caminho esteja sendo seguido corretamente, esta imagem permanece igual ao longo de toda simulação. Na Figura 4.16(b) tem-se apresentadas algumas posições instantâneas do robô e os caminhos seguidos. Para melhor ilustração deste resultado, a Figura 4.16(c) ilustra a evolução temporal das posturas. Verifica-se que o caminho foi seguido corretamente, mesmo sem nenhuma especificação prévia sobre suas características.

Os erros de seguimento de caminho são ilustrados na Figura 4.16(d), onde se verifica completa satisfação das restrições do problema. A entrada de controle ótima fornecida pelo otimizador está ilustrada na Figura 4.16(e), de onde é calculada a ação de controle física ω a ser enviada ao robô (Figura 4.16(f)). Desta figura observa-se que esforços de controle praticáveis em um cenário real foram obtidos.

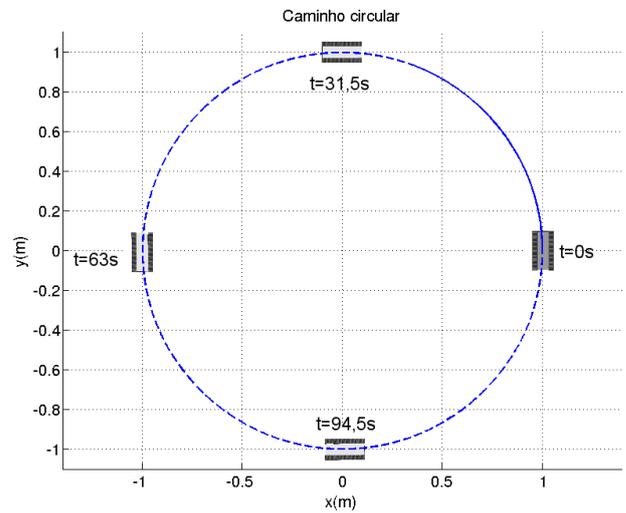
A seguir, uma análise considerando perturbações aplicadas indiretamente aos estados é realizada.

Tabela 4.5: Matrizes de ponderação.

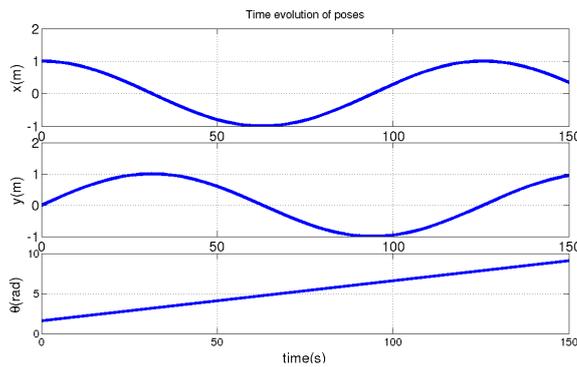
\mathbf{Q}_i	$diagonal(500;500)$
\mathbf{R}_i	1000



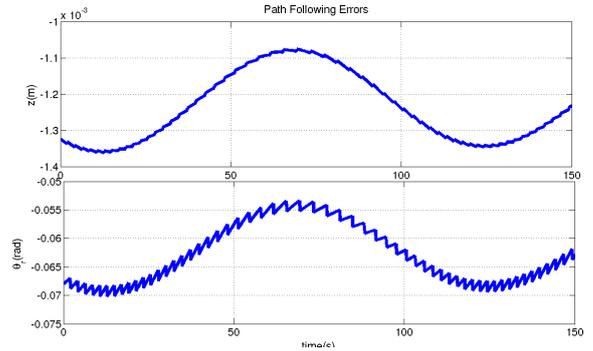
(a) Imagem inicial obtida pela câmera.



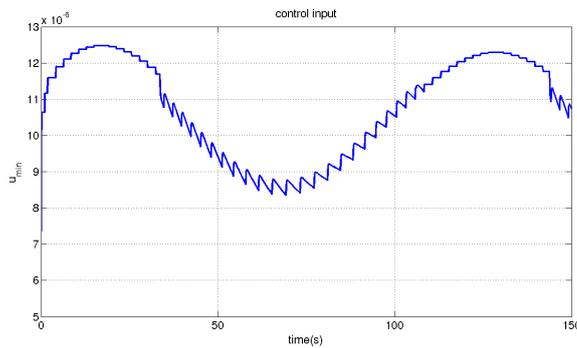
(b) Posições instantâneas do robô.



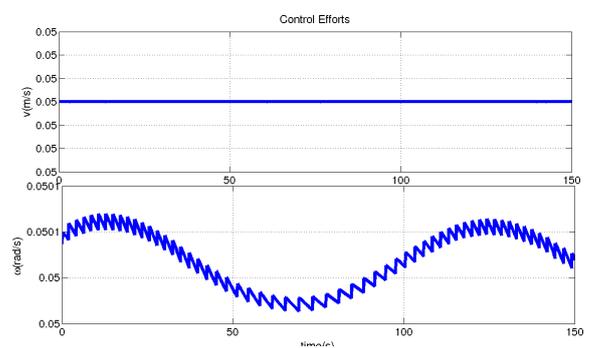
(c) Evolução temporal das posturas.



(d) Erros de seguimento de caminho.



(e) u_{min} .



(f) Esforços de controle.

Figura 4.16: Seguimento de caminhos visuais - Sem perturbações.

Simulação 2 - Com a aplicação de perturbações aos estados.

Nesta etapa é feita uma análise de desempenho quando existem perturbações no ambiente operacional, ou seja, objetiva-se analisar como o sistema se comporta caso o sistema visual seja deslocado de sua posição atual.

Para o modelo dado por (4.53), as referências obtidas através do posicionamento inicial da câmera com relação ao caminho devem ser mantidas, deste modo não é prevista a satisfação de um caso regulatório. Porém, é possível utilizar a mesma estrutura para a função objetivo desde que as referências para os estados estejam dentro dos limites de tolerância especificados.

A perturbação aplicada ao robô é de baixa magnitude, apenas 1cm na sua coordenada y com relação ao mundo. Porém, tal perturbação gera, no plano da imagem, perturbações maiores nas variáveis de estado, mais especificamente $Z = 0,0015$ e $\theta_r = -0,0776$, ficando θ_r superior aos limites especificados pelas restrições. Deste modo, considerando o tratamento para restrições convencional e critérios empíricos de sintonia, será muito difícil satisfazer aos objetivos de controle.

Intuitivamente, modificam-se as matrizes de ponderação para que seja dada maior importância à minimização de θ_r . A Tabela 4.6 apresenta os novos parâmetros.

Para o caso nominal (sem relaxamento das restrições), iniciada a simulação, aplica-se a perturbação descrita em $t = 4,25s$ e o otimizador acusa infactibilidade em $t = 8,2s$, já que os erros continuaram aumentando devido a impossibilidade de se obter as ações de controle necessárias com a sintonia utilizada.

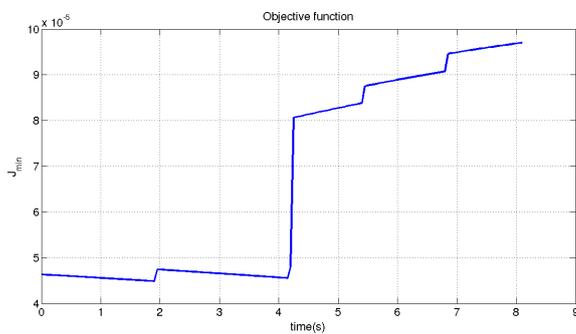
A Figura 4.17(a) mostra como se comporta a função objetivo neste caso. Verifica-se que tal função aumenta continuamente, revelando a impossibilidade em se regular os erros de estados. Tal resultado também indica que a técnica para garantia de estabilidade baseada em custo e restrições terminais, também não será suficiente para a regulação de perturbações visuais de tamanha magnitude.

Para solucionar este problema, considerando os mesmos parâmetros de sintonia, habilita-se o módulo para garantia de factibilidade, utilizando as mesmas métricas descritas nas subseções anteriores. A Figura 4.17(b) ilustra completa convergência da função objetivo neste caso.

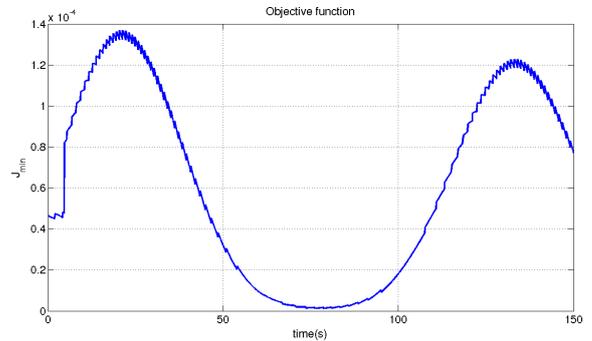
A Figura 4.17(c) apresenta a imagem obtida pela câmera no momento da perturbação, sendo confirmada que pequenas perturbações na estrutura do robô podem levar a perturbações maiores nos estados. Na Figura 4.17(d) tem-se apresentadas algumas posições instantâneas do robô e os caminhos seguidos. Para melhor ilustração deste resultado, a Figura 4.18(a) ilustra a evolução temporal das posturas. Verifica-se que, mesmo com a perturbação, o caminho foi seguido corretamente.

Tabela 4.6: Matrizes de ponderação.

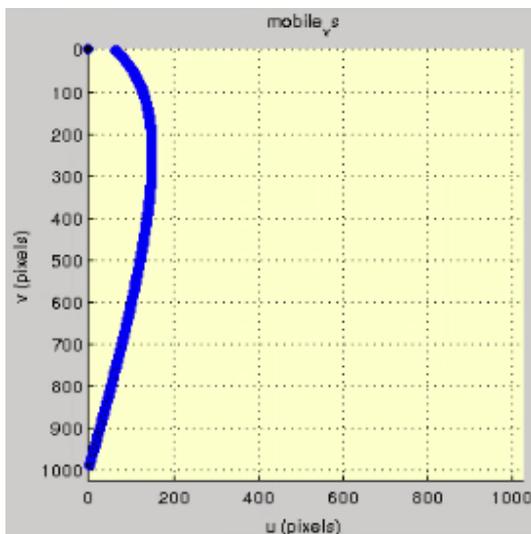
\mathbf{Q}_i	$diagonal(50;50000)$
\mathbf{R}_i	1000



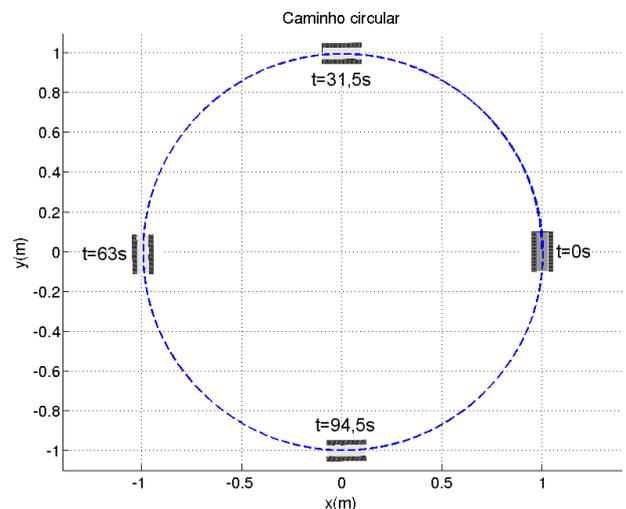
(a) J_{min} sem relaxamento das restrições.



(b) J_{min} com relaxamento das restrições.



(c) Imagem obtida pela câmera no momento da perturbação.



(d) Posições instantâneas do robô.

Figura 4.17: Seguimento de caminhos visuais - Com perturbações.

Os erros de seguimento de caminho são ilustrados na Figura 4.18(b), onde se verifica completa satisfação das restrições do problema. A entrada de controle ótima fornecida pelo otimizador está ilustrada na Figura 4.18(c), de onde é calculada a ação de controle física ω a ser enviada ao robô (Figura 4.18(d)). Desta figura observa-se que esforços de controle praticáveis em um cenário real foram obtidos.

Este resultado confirma a eficiência da técnica proposta, inclusive para o seguimento de caminhos visuais baseado em *NMPC*.

Considerando a plataforma computacional utilizada nestas simulações, avaliações sobre custo computacional não foram realizadas. Como feito para os robôs omnidirecionais, tal avaliação será apresentada através de experimentos práticos.

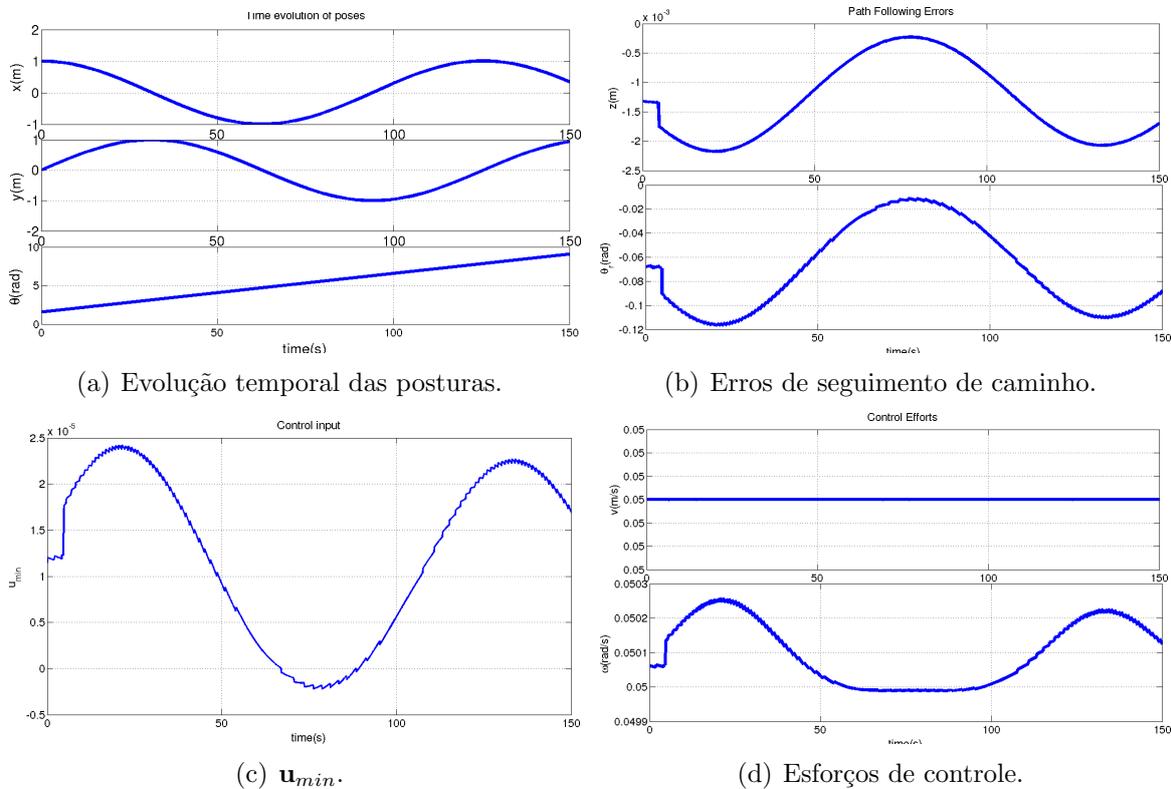


Figura 4.18: Seguimento de caminhos visuais - Com perturbações.

Resultados experimentais com o RobotShop Rover

Caminho 1 - Linha reta.

Inicialmente, objetivando validar a estratégia em um caso simplificado, utiliza-se um caminho em linha reta. Com esta análise será possível também verificar o comportamento do sistema odométrico, servindo de indicativo de sua representatividade para o seguimento de caminhos mais complexos. Utilizando os mesmos parâmetros da Tabela 4.6, os mesmos limitantes para as restrições da simulação 1, $v = 0,05\text{m/s}$, $T_p = T_c = 3$ amostras e $T_s = 0,5\text{s}$ foram adquiridos os resultados ilustrados na Figura 4.19.

Verifica-se que as restrições foram atendidas e que os esforços de controle se mantiveram compatíveis. Apesar disso, as oscilações nos estados e nas ações de controle indicam que novos parâmetros de sintonia devem ser escolhidos para o seguimento de caminhos com curvatura não nula. Deste resultado confirma-se que a evolução dos erros de postura oriundos do sistema odométrico pode limitar a aplicabilidade prática da arquitetura proposta para caminhos pré definidos, sendo justificada assim a utilização do sistema visual. Através da Figura 4.19(b) é possível confirmar a eficiência computacional da técnica.

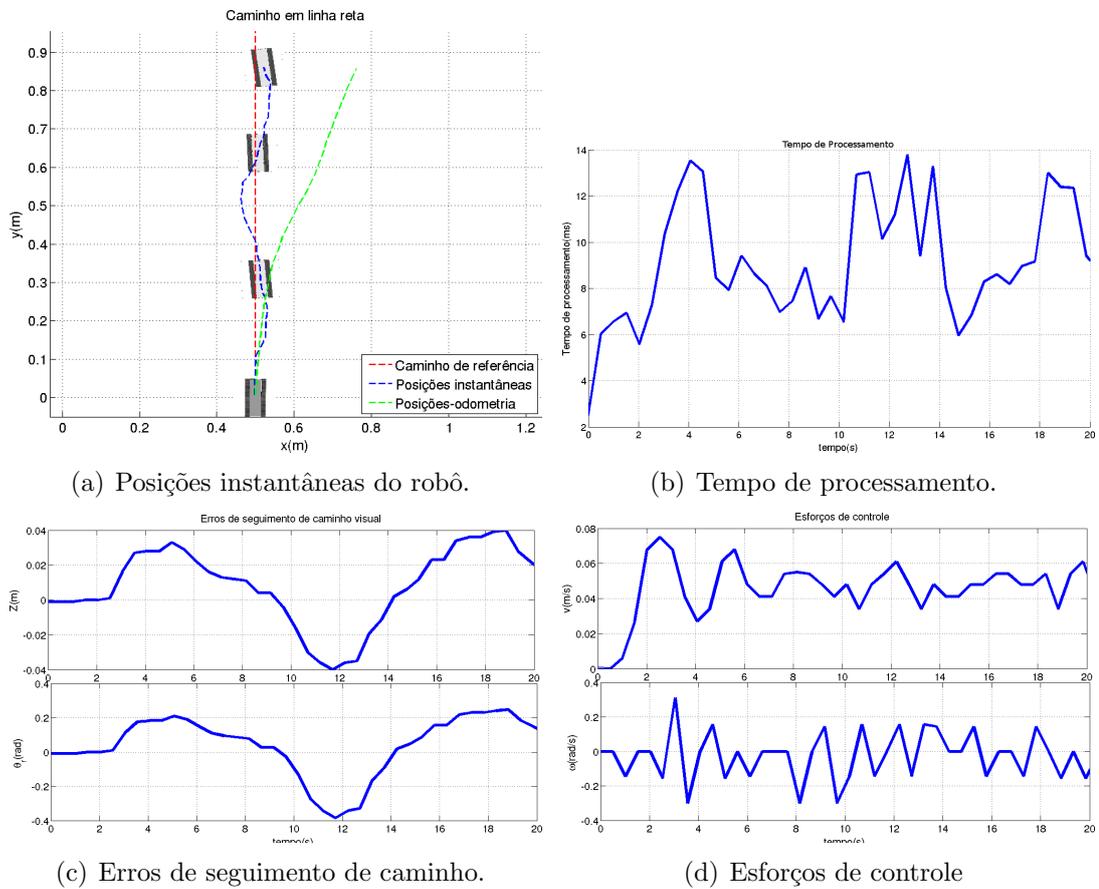


Figura 4.19: Seguimento de caminhos visuais - Linha reta.

Ainda para o caminho em linha reta, foi feita uma avaliação de desempenho considerando a existência de erros iniciais nos estados. A Figura 4.20(a) ilustra o resultado obtido sem o relaxamento das restrições e a Figura 4.20(c) com o relaxamento. Os custos computacionais destes experimentos também são apresentados.

Verifica-se que, sem a utilização da técnica para o relaxamento das restrições, os erros de estado não foram corrigidos de maneira satisfatória, ficando o caminho no limite do campo visual da câmera. Observa-se também que, com a técnica proposta, após a regulação dos erros de estado, os erros aumentaram no sentido oposto até os limites das restrições, sendo necessária uma avaliação do desempenho para outros tipos de caminho. Os custos computacionais se mantiveram baixos em ambos os casos, confirmando mais uma vez a eficiência computacional da técnica.

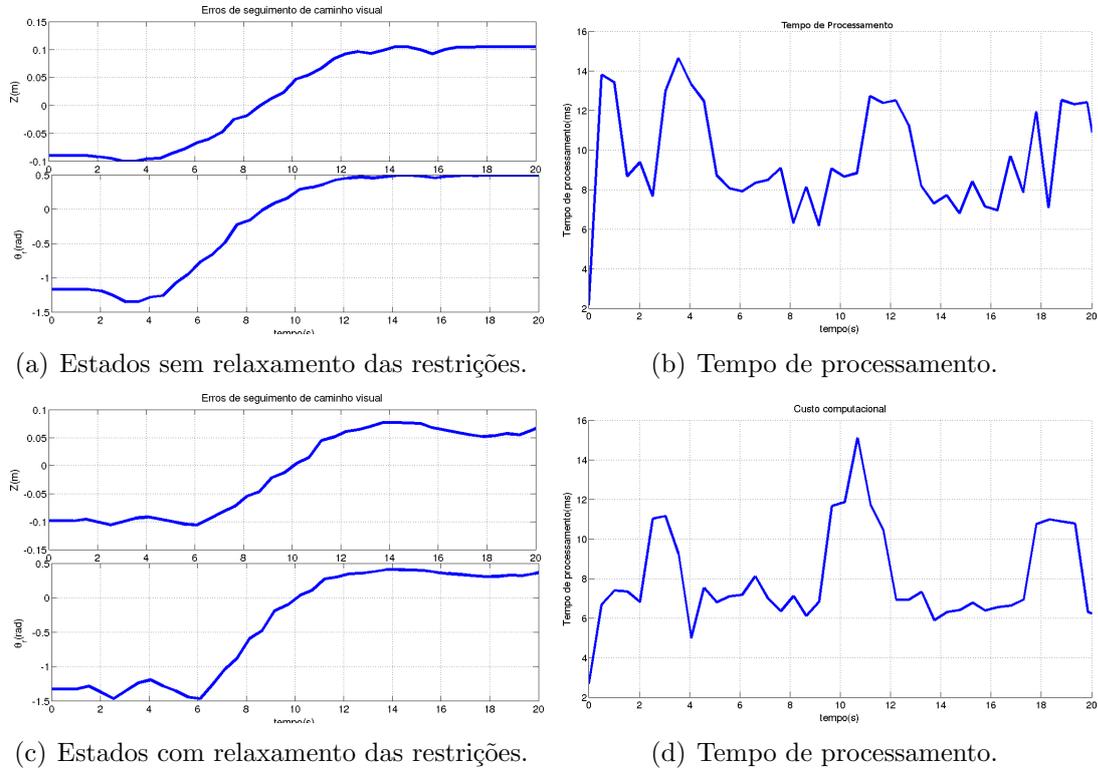


Figura 4.20: Seguimento de caminhos visuais - Erros iniciais nos estados.

Caminho 2 - Caminho arbitrário.

Uma breve análise dos resultados obtidos anteriormente, direcionam para uma avaliação da técnica proposta considerando caminhos mais complexos, que possuam função curvatura diferente de 0 (caminho reto) ou constante (caminho circular). Nesta etapa é feita uma avaliação de desempenho para o caminho arbitrário ilustrado na Figura 4.21. Observam-se trechos de curvatura variável o que permite uma melhor avaliação da estratégia de controle proposta.

Inicialmente, considerando as características do robô mencionadas no Capítulo 2 e do caminho a ser seguido, são adquiridos resultados resultados para $v = 0,03\text{m/s}$. Novos parâmetros de sintonia foram escolhidos para este caso (Tabela 4.7).

Os novos limites para as restrições foram definidos como segue:

$$|u_1| \leq 0,2; |z| \leq 0,05 \text{ m}; |\theta_e| \leq 0,3 \text{ rad}.$$

Para um tempo de amostragem $T_s = 0,5\text{s}$ e definindo-se $T_p = T_c = 3$ amostras,

Tabela 4.7: Matrizes de ponderação.

\mathbf{Q}_i	$diagonal(50;0.5)$
\mathbf{R}_i	10



Figura 4.21: Primeiro caminho arbitrário.

foram adquiridos os resultados ilustrados na Figura 4.22.

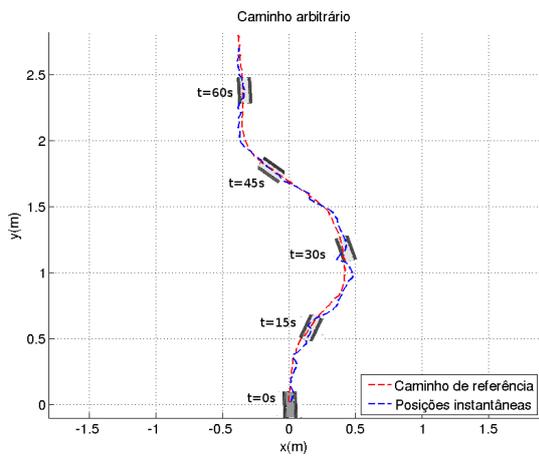
É possível verificar através da Figura 4.22(a), que o caminho foi seguido satisfatoriamente. Considerando a baixa representatividade do sistema odométrico, confirmada através dos resultados obtidos para o caminho 1, a posição calculada por este mecanismo de localização não é mais apresentada, já que o mesmo não possui influência alguma para o seguimento de caminhos visuais.

Verifica-se na Figura 4.22(b) que os tempos de processamento se mantiveram inferiores ao período de amostragem. Destes resultados, observa-se também que as restrições das saídas foram plenamente atendidas (4.22(c)) e os esforços de controle se mantiveram factíveis ao longo do experimento (4.22(d)).

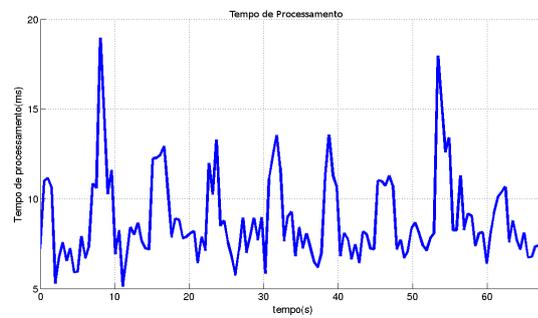
Nesta etapa, uma breve comparação com a técnica original ($\omega = Kz$) (Coulaud *et al.* 2006) é realizada. Considerando $K = 4$, foram obtidos os resultados ilustrados na Figura 4.23.

Observa-se que, a despeito do tempo de processamento inferior a 5% de T_s , os erros de estado foram elevados, alcançando, em determinados instantes, valores próximos do limite definido pelo campo visual da câmera. Tais resultados justificam o desenvolvimento de controladores ótimos para o seguimento de caminhos visuais.

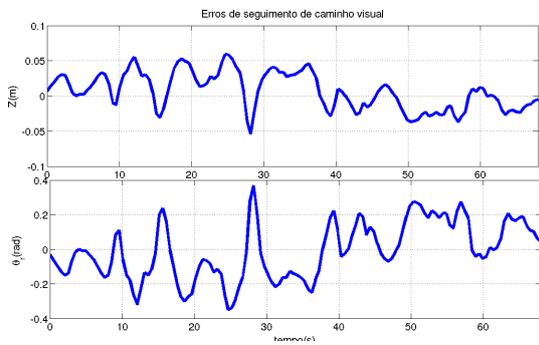
Dos resultados obtidos anteriormente, é possível concluir que, para velocidades de navegação maiores, caso em que os esforços de controle devem ser ainda maiores, devido às características do robô (tipo tanque, ruídos de medição nas velocidades), da câmera (baixas resolução e cadência visual) e do piso (material cerâmico escorregadio) onde os testes foram realizados, os resultados tendem a ser piores. Mantendo-se os mesmos parâmetros do experimento anterior e, objetivando a obtenção de esforços de controle maiores, aumenta-se a restrição da entrada para $|u_1| \leq 1$. A Figura 4.24 ilustra os resultados obtidos.



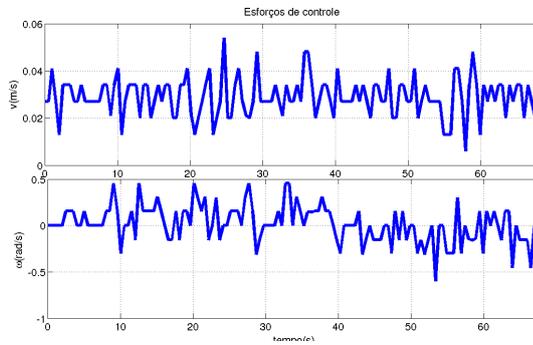
(a) Posições instantâneas do robô.



(b) Tempo de processamento.

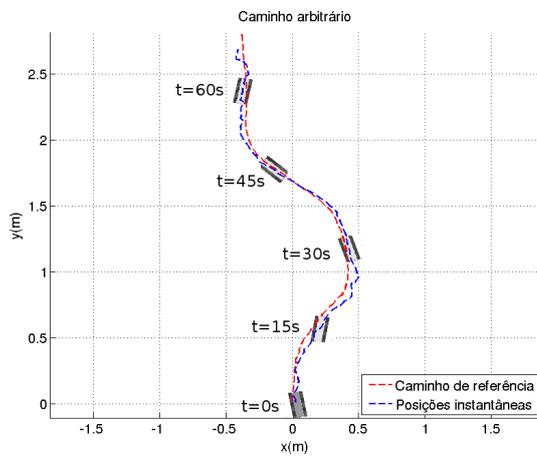


(c) Erros de seguimento de caminho.

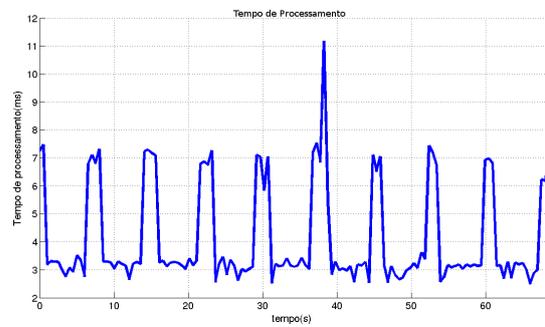


(d) Esforços de controle.

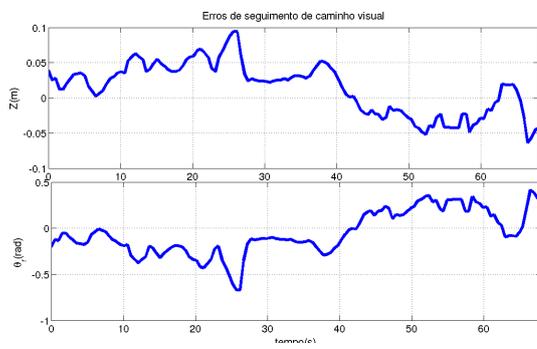
Figura 4.22: Seguimento de caminhos visuais - Primeiro Caminho arbitrário - NMPC Nominal.



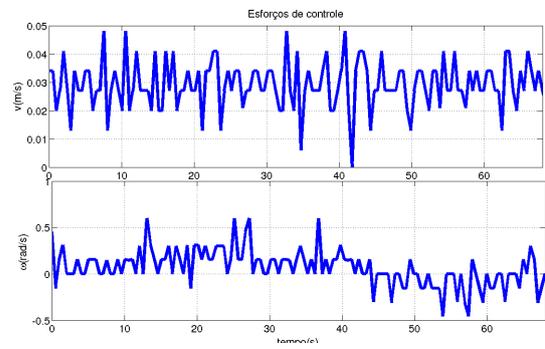
(a) Posições instantâneas do robô.



(b) Tempo de processamento.

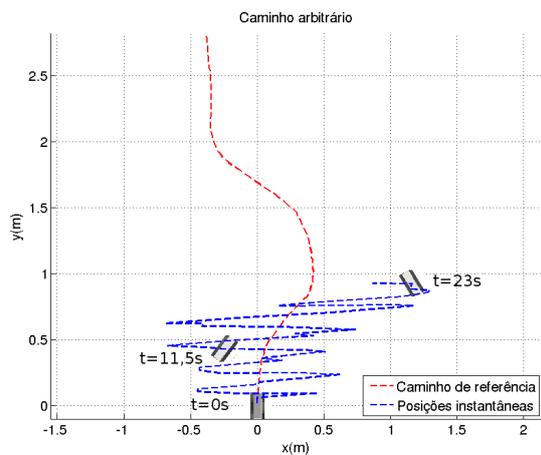


(c) Erros de seguimento de caminho.

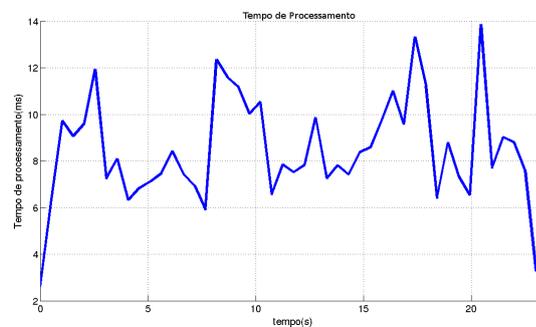


(d) Esforços de controle.

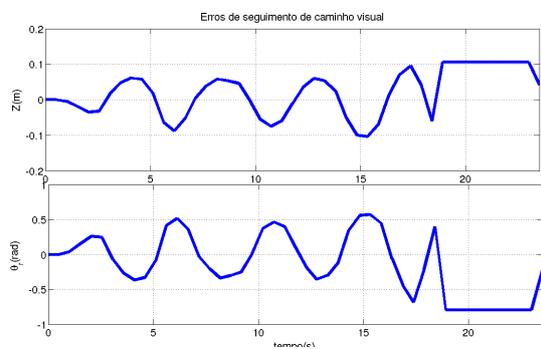
Figura 4.23: Seguimento de caminhos visuais - Caminho arbitrário - $\omega = KZ$.



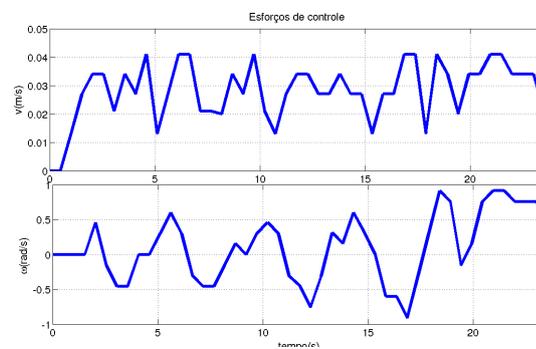
(a) Posições instantâneas do robô.



(b) Tempo de processamento.



(c) Erros de seguimento de caminho.



(d) Esforços de controle.

Figura 4.24: Seguimento de caminhos visuais - Primeiro Caminho arbitrário - Folga na restrição da entrada.

Verifica-se que o sistema ficou instável mesmo com erros iniciais nulos. Tal resultado é justificado pela incapacidade do sistema visual acompanhar as variações em θ_r em tempo de execução. Tal resultado seria ainda pior para velocidades de navegação maiores.

Uma das alternativas para solucionar este problema seria uma nova sintonia do controlador, porém a adequação dos parâmetros seria uma tarefa difícil, haja vista a não linearidade do modelo. Assim, definiu-se que uma estratégia para garantia de estabilidade deve ser utilizada.

Habilitando o estágio para garantia de estabilidade a partir de custo e restrições terminais, para os mesmos parâmetros de sintonia, e, acrescentando-se $P = \text{diag}(100; 100)$ e $\alpha = 2$, foram obtidos os resultados ilustrados na Figura 4.25.

Verifica-se o correto atendimento dos objetivos de controle, porém com a utilização de toda a margem definida para as restrições na saída, o que evidencia a necessidade de ajustes nos parâmetros de sintonia. Apesar da estabilidade garantida, observa-se que, quando o caminho sai do campo visual (fim do caminho em torno de 65s), o sistema voltou a instabilizar, já que neste instante, o problema se tornou infactível.

Outra constatação importante deste resultado refere-se ao tempo de processamento ilustrado na Figura 4.25(d) onde verifica-se que a inclusão do módulo para garantia de estabilidade não prejudicou a eficiência computacional da técnica.

Finalmente é feita uma avaliação habilitando o estágio de relaxamento das restrições de controle e retirando a garantia de estabilidade implementada anteriormente. Objetiva-se verificar o desempenho do controlador com factibilidade garantida.

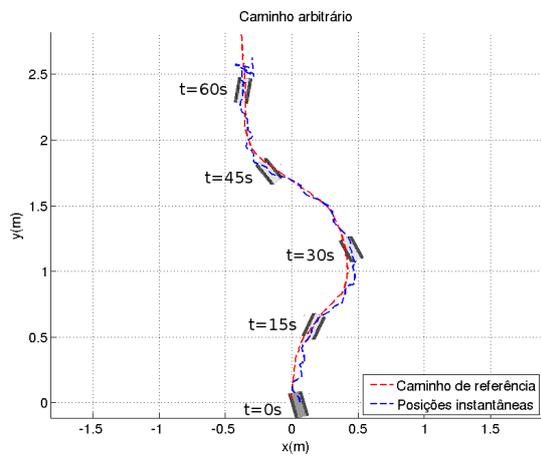
Neste caso, como já foi feita uma avaliação anterior para pequenas perturbações, os resultados são adquiridos para $v = 0.05\text{m/s}$ e os parâmetros de sintonia estão apresentados na Tabela 4.8.

Foram obtidos os resultados ilustrados na Figura 4.26.

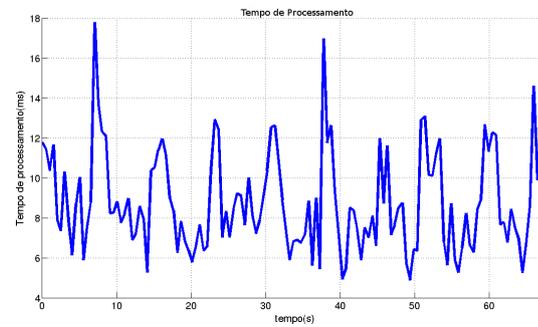
Em 4.26(b) é possível verificar que os custos computacionais foram ainda menores. Em 4.26(c) verifica-se que os erros de estados foram maiores em determinados instantes, porém, nunca saindo do campo visual da câmera. Com esforços de controle compatíveis com a plataforma experimental (4.26(d)) a factibilidade foi mantida a todos os instantes, mesmo ao final do caminho quando o mesmo sai do campo visual da câmera. Tal comportamento é requerido para a interrupção segura do processo além de ser fundamental para o desenvolvimento dos controladores de formação utilizando informações visuais. Estes resultados foram reunidos e publicados em (Ribeiro *et al.* 2016b) e (Ribeiro *et al.* 2016a).

Tabela 4.8: Matrizes de ponderação.

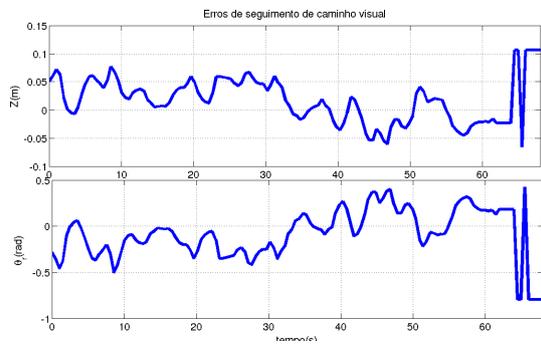
\mathbf{Q}_i	$\text{diagonal}(5000;50)$
\mathbf{R}_i	0.1



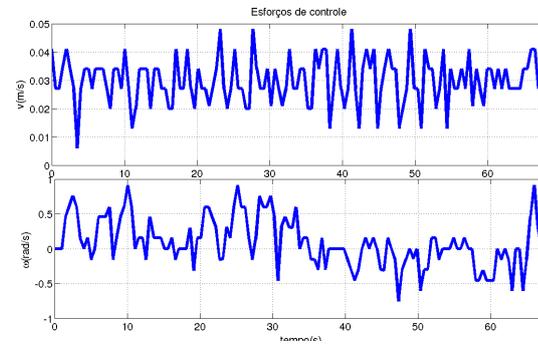
(a) Posições instantâneas do robô.



(b) Tempo de processamento.

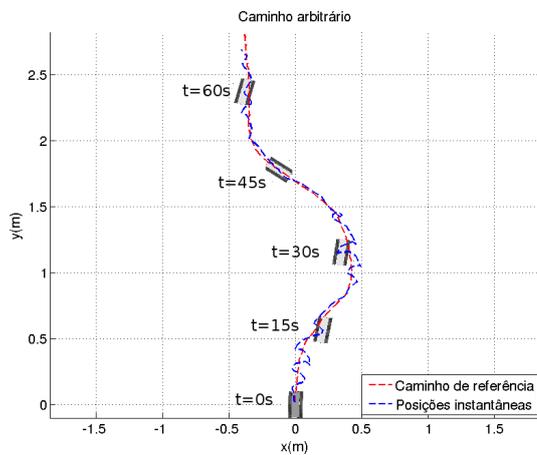


(c) Erros de seguimento de caminho.

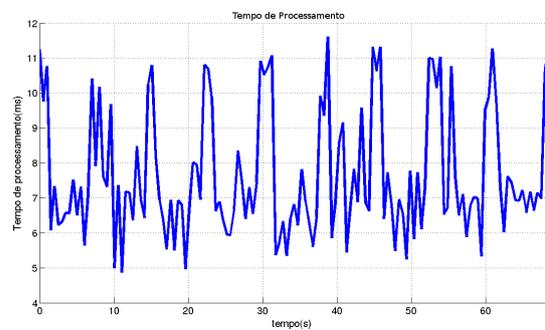


(d) Esforços de controle.

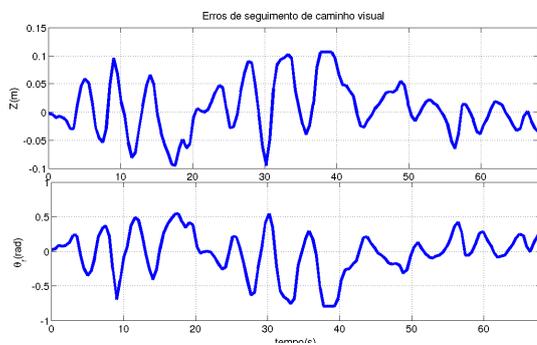
Figura 4.25: Seguimento de caminhos visuais - Caminho arbitrário - Garantia de estabilidade.



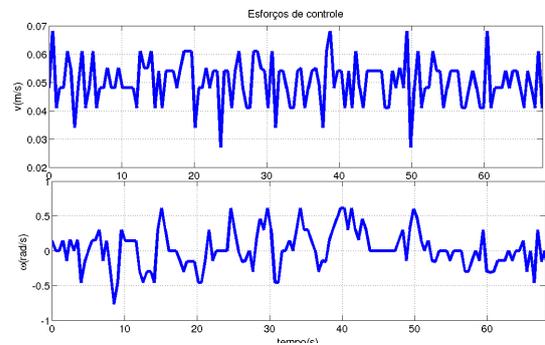
(a) Posições instantâneas do robô.



(b) Tempo de processamento.



(c) Erros de seguimento de caminho.



(d) Esforços de controle.

Figura 4.26: Seguimento de caminhos visuais - Caminho arbitrário - Garantia de factibilidade.

Conforme detalhado no Capítulo 2, os aspectos construtivos deste robô, especialmente com relação aos seus mecanismos sensoriais, inviabilizam a busca por parâmetros de sintonia que possibilitem a aquisição de resultados melhores. Além disso, a própria *webcam* utilizada acrescenta dificuldades adicionais devido a valores baixos de resolução e cadência que impedem uma calibração mais precisa.

Estas imperfeições geram sérias implicações na aquisição das variáveis relacionadas ao seguimento de caminho visual, especialmente a curvatura. Neste caso, tal grandeza não foi ilustrada devido aos elevados ruídos que limitaram uma melhor calibração. A seguir, resultados adquiridos com uma plataforma experimental composta por sensores e atuadores de melhor qualidade quando comparada ao *robotshop rover* e com uma *webcam HD* de maiores resolução e cadência visual, permitirá o detalhamento desta grandeza.

Resultados experimentais com o Lego EV3

A utilização deste robô permite avaliações mais precisas da técnica de seguimento de caminhos visuais, já que o mesmo possui instrumentação básica mais eficaz e maior eficiência energética, dentre outros fatores discutidos no Capítulo 2. A plataforma experimental concebida com este robô viabiliza a sua aplicação para o seguimento de caminhos com maiores curvaturas e comprimento total, permitindo explicitar os requisitos dos controladores de formação utilizando informações visuais.

Para uma primeira avaliação da estratégia com esta nova plataforma, utiliza-se o caminho ilustrado na Figura 4.27. Observa-se que este caminho possui curvatura variável em mais trechos do que o primeiro caminho, tornando esta grandeza ainda mais influenciável. Mantendo a velocidade de navegação $v = 0.05\text{m/s}$, $T_p = T_c = 3T_s$ com $T_s = 0,2\text{s}$, foram escolhidos os parâmetros de sintonia da Tabela 4.9. Tais parâmetros foram escolhidos com o objetivo de suavizar as ações de controle, reduzindo as oscilações verificadas nos experimentos anteriores. Os valores limitantes das restrições foram mantidos os mesmos dos casos anteriores.

Para o caso nominal e sem erros iniciais, foram obtidos os resultados da Figura 4.28. De modo geral, verifica-se que, ao contrário dos resultados anteriores, com esta nova plataforma experimental, o caso nominal para o seguimento de caminhos visuais satisfaz aos objetivos de controle, mesmo para $v = 0,05\text{m/s}$, para um caminho relativamente longo e com trechos mais curvos.

Observa-se, a partir da Figura 4.28(b), que durante a regulação em trechos de maior curvatura o tempo de processamento apresenta picos que, mesmo sendo de baixa magnitude, precisam ser melhor avaliados. Verifica-se também na Figura 4.28(c) que apesar das restrições satisfeitas, os estados chegam próximos dos valores especificados para as restrições, reduzindo a capacidade de regulação nos trechos de maior curvatura. Através da Figura 4.28(d), observa-se que os esforços de controle se mantiveram inferiores aos limites das restrições, sendo possível, desta forma, obter melhores resultados através da escolha de novos parâmetros de sintonia.



Figura 4.27: Segundo caminho arbitrário

Tabela 4.9: Matrizes de ponderação.

\mathbf{Q}_i	$diagonal(10;1)$
\mathbf{R}_i	0.2

Entretanto, para comparar as estratégias propostas, mantém-se os parâmetros de sintonia e habilita-se apenas o módulo para garantia de estabilidade. Mantendo $P = diagonal(100; 100)$ e $\alpha = 2$, foram obtidos os resultados da Figura 4.29.

Observa-se que foram obtidos resultados similares ao caso nominal, variando, notoriamente, o tempo de processamento que apresentou picos maiores e mais frequentes. Mesmo que pequena, é possível visualizar diferenças nos gráficos dos estados e das ações de controle, com o experimento atual apresentando desempenho ligeiramente superior. A vantagem da garantia de estabilidade, neste caso, seria melhor visualizada através da aplicação de maiores perturbações que, comparado ao caso nominal, geraria as ações de controle necessárias à estabilização dos estados na região terminal das restrições.

Complementando esta análise inicial, habilita-se apenas o módulo para garantia de factibilidade. Para os mesmos parâmetros de sintonia, foram obtidos os resultados da Figura 4.30.

Resultados similares foram obtidos, porém, análise detalhada permite visualizar melhor desempenho da técnica, já que os erros de estado se mantiveram próximos do *NMPC* com garantia de estabilidade com tempo de processamento próximo do caso nominal. Conforme discutido em seções anteriores, a principal vantagem desta técnica consiste em regular perturbações além dos limites das restrições.

A Figura 4.31 ilustra a curvatura visual medida em ambos os casos. Sabe-se que esta medida é bastante sensível aos erros de rastreamento, sendo útil, portanto, para evidenciar o melhor desempenho da técnica baseada no relaxamento das restrições. Especificamente para o caso do seguimento de caminhos visuais, esta técnica também permite o desenvolvimento de velocidades maiores, conforme detalhado a seguir.

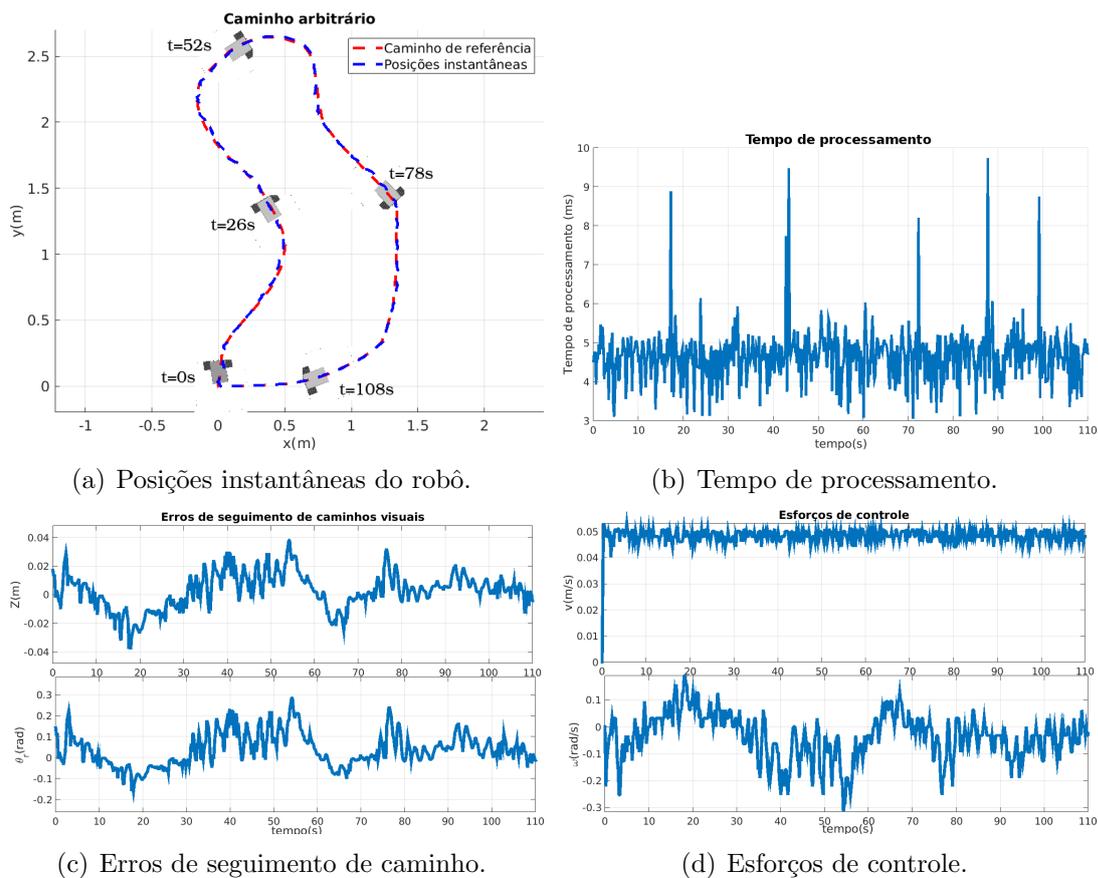
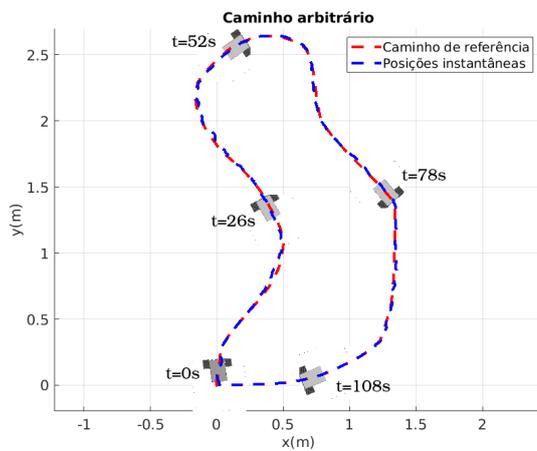
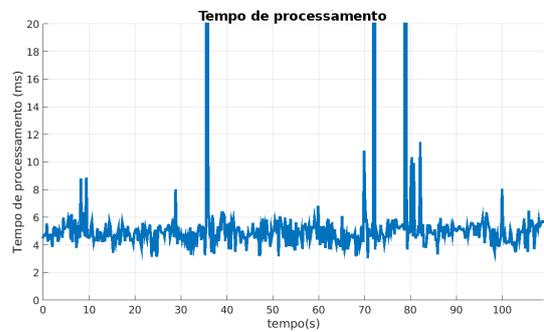


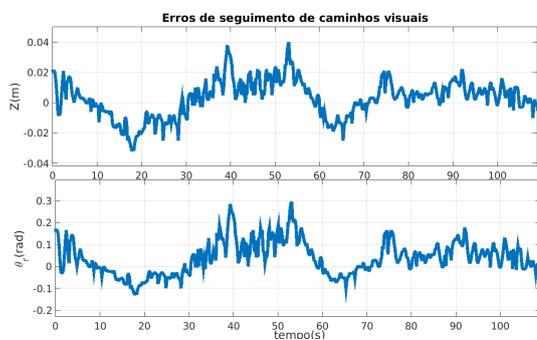
Figura 4.28: Seguimento de caminhos visuais - Segundo caminho arbitrário - *NMPC* nominal.



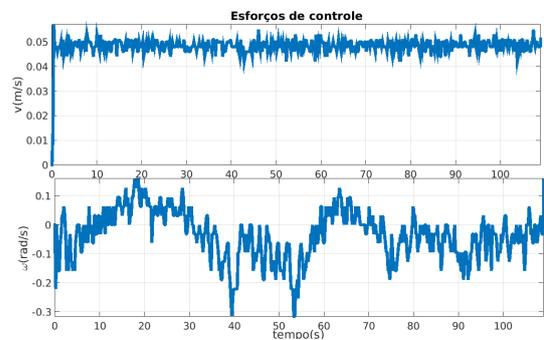
(a) Posições instantâneas do robô.



(b) Tempo de processamento.

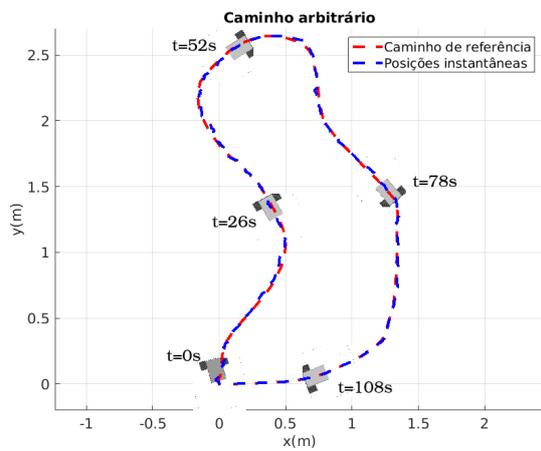


(c) Erros de seguimento de caminho.

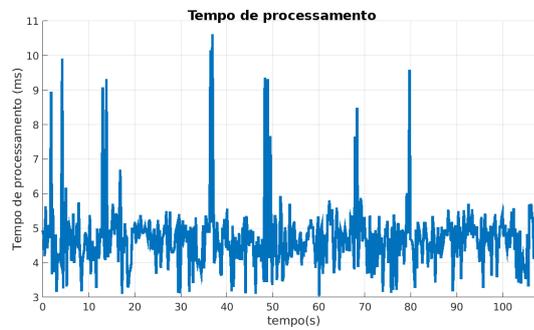


(d) Esforços de controle.

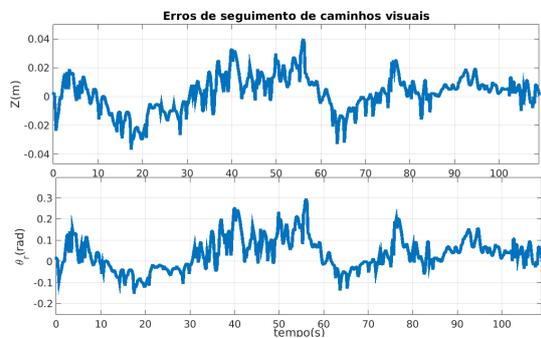
Figura 4.29: Seguimento de caminhos visuais - Segundo caminho arbitrário - *NMPC* com garantia de estabilidade.



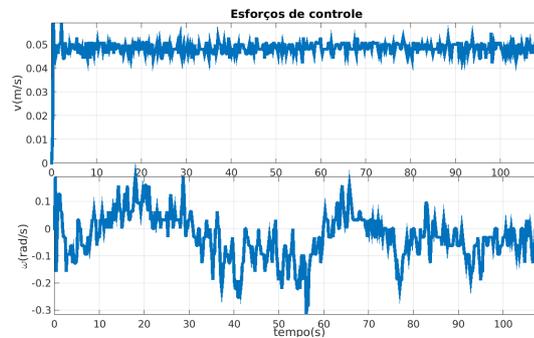
(a) Posições instantâneas do robô.



(b) Tempo de processamento.

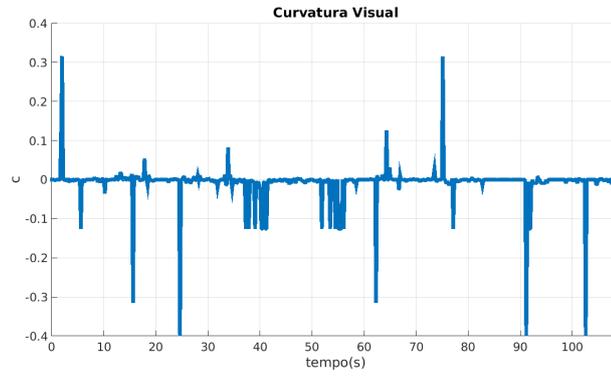


(c) Erros de seguimento de caminho.

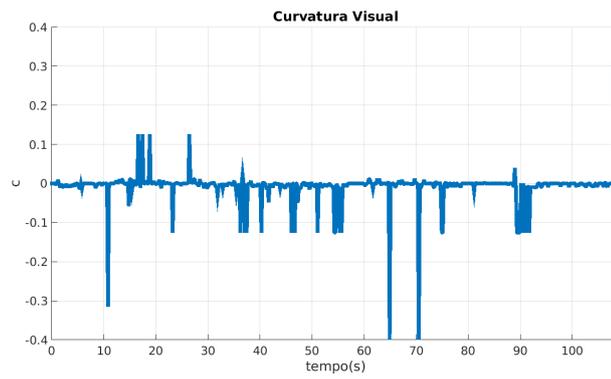


(d) Esforços de controle.

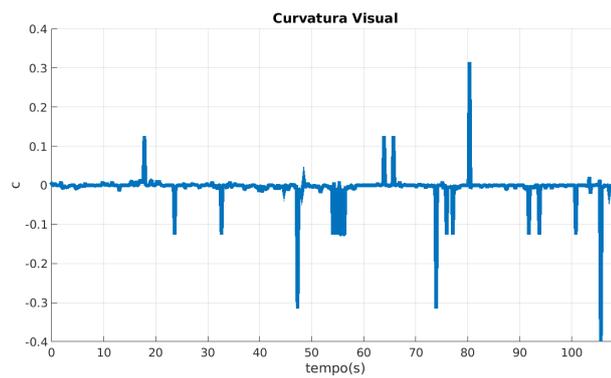
Figura 4.30: Seguimento de caminhos visuais - Segundo caminho arbitrário - *NMPC* com garantia de factibilidade.



(a) Caso nominal.



(b) Garantia de estabilidade.



(c) Garantia de factibilidade.

Figura 4.31: Seguimento de caminhos visuais - Análise da curvatura visual medida.

Influência da velocidade.

Surge como consequência das limitações de cadência do sistema visual, a necessidade de limitar a velocidade de navegação em níveis inferiores aos valores máximos praticáveis com o robô móvel. Esta etapa objetiva analisar o comportamento das estratégias propostas para velocidades maiores, mais especificamente $v = 0,1\text{m/s}$ e $v = 0,2\text{m/s}$.

Dos experimentos anteriores é possível concluir que, caso alguma métrica formal de redução de velocidade em trechos mais curvos fosse utilizada, seria possível obter resultados satisfatórios para ambas as estratégias de controle, porém, é interessante explicitar as limitações das propostas.

Sem alteração alguma nos parâmetros de sintonia, resultados de experimentos com $v = 0,1\text{m/s}$ são apresentados na Figura 4.32.

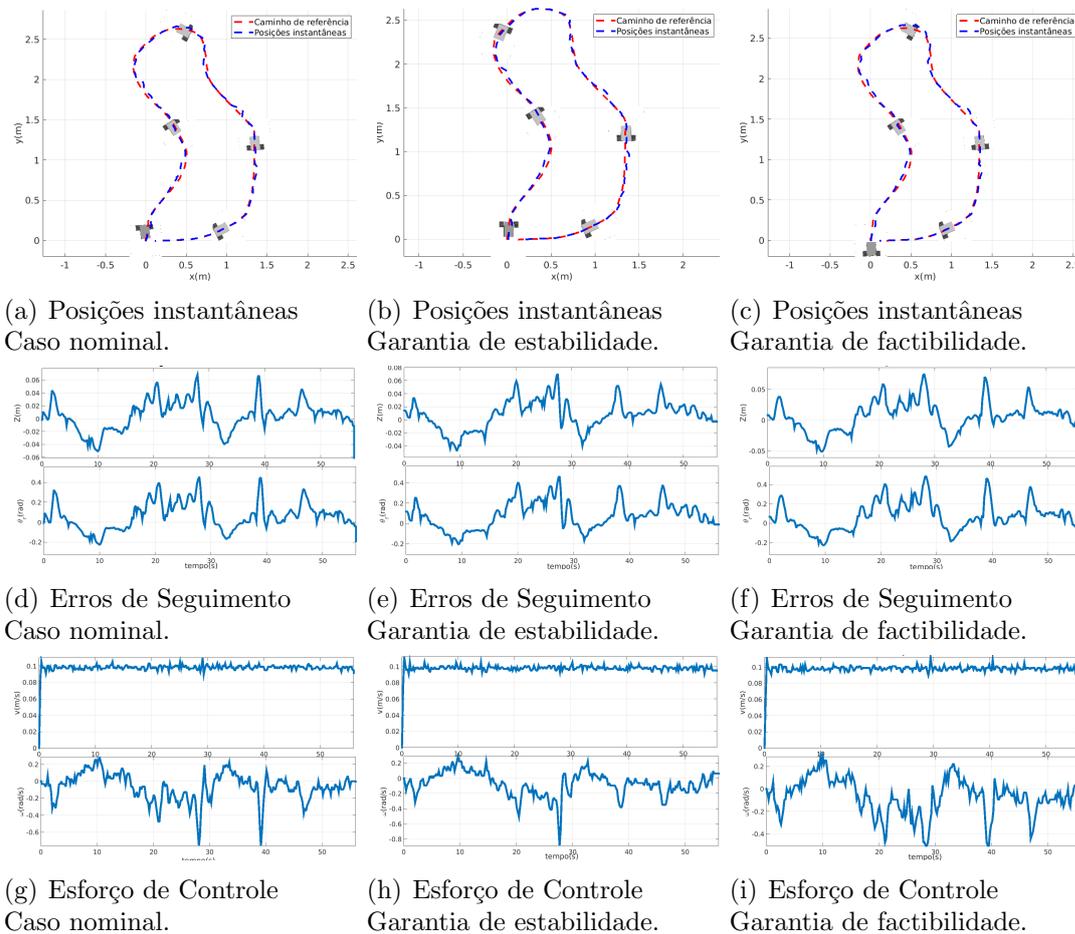


Figura 4.32: Comparação das abordagens para $v = 0,1\text{s}$.

Confirma-se assim, que os resultados com garantia de factibilidade se mantiveram melhores também neste cenário, já que mesmo com as restrições relaxadas, erros de estado de magnitude inferiores foram obtidos, inclusive nos trechos mais curvos. É possível observar que as restrições para os erros de estados foram violadas em determinados momentos para o caso nominal e com garantia de estabilidade.

Ainda para estes experimentos, os desempenhos das abordagens são comparados através de duas métricas quantitativas, mais especificamente, utiliza-se a integral do erro absoluto (IAE) e a variação de controle total (TV). O índice IAE , calculado por $\int_0^{T_{END}} |e(t)| dt$, é muito utilizado para comparação de desempenho de estratégias distintas em experimentos semelhantes, por outro lado o índice TV , calculado por $\sum_{k=0}^{k_{END}} |u(k) - u(k-1)|$, objetiva avaliar o efeito dos ruídos nos sinais de controle.

Para o problema em questão, os erros (z e θ_r) e as ações de controle (v e ω) geram os dados ilustrados na Tabela 4.10. Observam-se valores próximos, porém com leve vantagem para o VPF com garantia de estabilidade em ambos os índices. Em contrapartida, as violações momentâneas nas restrições, elevam o tempo de processamento, sendo este o principal prejuízo desta técnica.

Destes resultados, é natural esperar que ocorra grande deterioração do desempenho para velocidades maiores. Isto se confirma através de experimentos com $v = 0,2\text{m/s}$, ilustrados a Figura 4.33. Observa-se que a única abordagem que consegue seguir o caminho completamente foi a que implementa o relaxamento das restrições, já que em momento algum o caminho saiu do campo visual da câmera.

Para a abordagem nominal, o caminho foi perdido em aproximadamente 15s e para o caso com garantia de estabilidade em aproximadamente 6s, ratificando a importância da factibilidade recursiva para a garantia de estabilidade. Foram feitas modificações na ponderação do custo terminal, porém, ainda assim o caminho não foi seguido completamente, apenas aumentou o intervalo de tempo de seguimento.

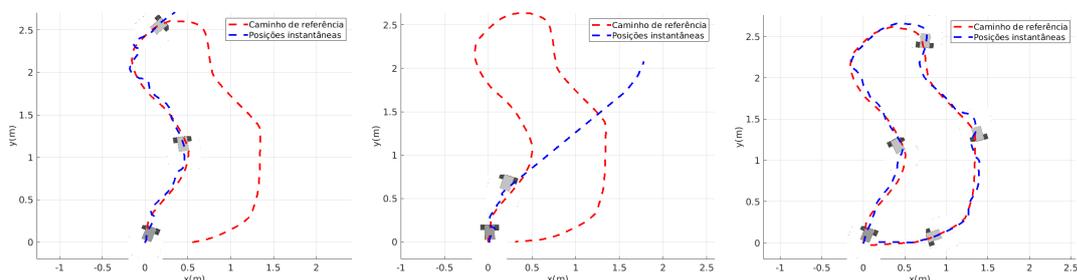
Mesmo tendo bastante reserva para os esforços de controle, o tempo de processamento elevado impede o cálculo das ações de controle em apenas $3T_s$. Neste contexto, relaxar as restrições sem penalização na função objetivo surge como abordagem adequada.

Os índices de desempenho calculados para o VPF com garantia de factibilidade para $v = 0,1\text{m/s}$ e $v = 0,2\text{m/s}$ estão dispostos na Tabela 4.11. Verifica-se que, para o dobro da velocidade, o IAE piorou um pouco, por outro lado houve melhora no TV já que para velocidades maiores, poucas variações foram necessárias para corrigir a pose. Em todos os casos, a técnica para a garantia de factibilidade proposta atende à regulação dos erros de estado com ações de controle adequadas à plataforma experimental e baixo tempo de processamento, mesmo sem modificação alguma na estrutura da função objetivo.

Tal avaliação é importante já que, para os controladores de formação, estão previstas variações de velocidade para coordenação dos movimentos relativos.

Tabela 4.10: Análise de desempenho - Comparação das abordagens para o *VPF* - $v = 0,1$ [m/s]

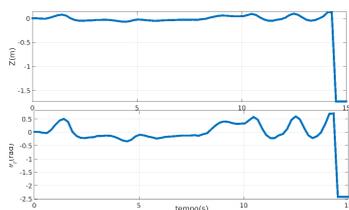
	<i>IAE</i>		<i>TV</i>	
	z_{IAE} [m]	θ_{rIAE} [rad]	v_{TCV} [m/s]	ω_{TCV} [rad/s]
Nominal	1,358	6,966	0,878	15,830
Garantia de estabilidade	1,024	6,584	0,965	12,276
Garantia de factibilidade	1,111	7,139	0,923	13,292



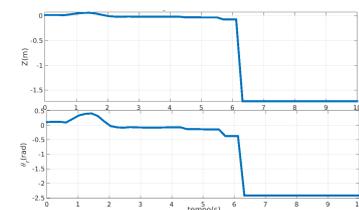
(a) Posições instantâneas
Caso nominal.

(b) Posições instantâneas
Garantia de estabilidade.

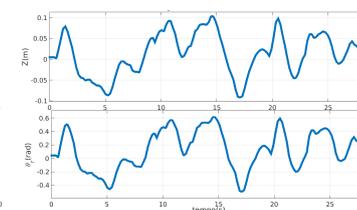
(c) Posições instantâneas
Garantia de factibilidade.



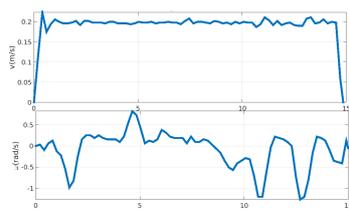
(d) Erros de Seguimento
Caso nominal.



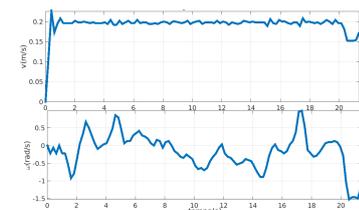
(e) Erros de Seguimento
Garantia de estabilidade.



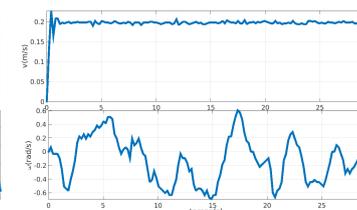
(f) Erros de Seguimento
Garantia de factibilidade.



(g) Esforço de Controle
Caso nominal.



(h) Esforço de Controle
Garantia de estabilidade.



(i) Esforço de Controle
Garantia de factibilidade.

Figura 4.33: Comparação das abordagens para $v = 0,2$ s.

Tabela 4.11: Análise de desempenho - *VPF* com garantia de factibilidade para 2 velocidades

v [m/s]	<i>IAE</i>		<i>TV</i>	
	z_{IAE} [m]	θ_{rIAE} [rad]	v_{TCV} [m/s]	ω_{TCV} [rad/s]
0,1	1,111	7,139	0,923	13,292
0,2	3,039	18,435	0,718	12,786

Influência do horizonte de predição.

Nesta etapa, analisam-se as técnicas propostas através de experimentos nos quais os horizontes de predição T_p e de controle T_c são variados. Mais especificamente, além dos índices de desempenho IAE e TV , objetiva-se analisar como o aumento de tais horizontes influenciam no tempo de processamento T_{proc} , no valor mínimo da função objetivo J_{min} e na quantidade de energia necessária para o seguimento de caminhos visuais, medida através de $|u_{min}|$.

Sabe-se que tais medidas variam bastante em função do caminho, velocidade de navegação, parâmetros de sintonia e plataforma computacional, sendo portanto, os testes realizados para o caminho arbitrário 2, $v = 0,1\text{m/s}$, parâmetros da Tabela 4.9 e para o PC_1 especificado no Capítulo 2. Tais análises são importantes para nortear a escolha da abordagem adequada para os controladores de formação utilizando informações visuais, além de ratificar a necessidade da utilização de valores pequenos para os horizontes.

A Tabela 4.12 apresenta as medidas de tempo de processamento para caso nominal. Confirma-se que tanto o valor médio como as medidas de dispersão (variância e desvio padrão) crescem diretamente com os horizontes. Os grandes valores de variância medidos a partir de $T_p = T_c = 7$ refletem os picos de processamento que surgem nos trechos mais curvos, indicando possível violação das restrições do problema.

Para garantia de estabilidade, verifica-se a partir da Tabela 4.13 que a relação entre tempo de processamento e horizonte continua sendo direta, a menos para para $T_p = T_c > 5$. Porém verifica-se, para estes horizontes, variância bastante elevada, indicando, mais uma vez, picos de processamento que se relacionam com violação das restrições. Os critérios de otimização mais restritivos oriundos da implementação do custo e das restrições terminais justificam a antecipação deste efeito relativamente ao caso nominal.

Conclui-se a análise dos tempos de processamento em função dos horizontes com a técnica para garantia de factibilidade, cujos resultados estão dispostos na Tabela 4.14. Verifica-se que foram obtidos valores menores do que os anteriores, inclusive para horizontes elevados. As folgas acrescentadas às restrições permitem diminuir o tempo de cálculo das ações de controle, sendo possível a aquisição de novas imagens que são utilizadas para a correção dos possíveis erros oriundos das ações de controle anteriormente saturadas.

As Tabelas 4.15, 4.16 e 4.17 apresentam análises do comportamento de J_{min} ao longo do horizontes para as três abordagens. Para uma maior quantidade de variáveis de decisão, espera-se um aumento desta quantidade em todos os cenários, situação confirmada com certo nível de precisão.

Das medidas de dispersão, verifica-se que as características do caminho possuem influência nesta medida, especialmente para horizontes elevados. Os valores baixos de J_{min_m} em ambos os casos, servem de indicativo de que o sistema permaneceu estável. O controlador com garantia de factibilidade forneceu medidas inferiores para horizontes elevados, justificando a importância da manutenção da estrutura

nominal da função objetivo.

Relativamente às ações de controle internas, as Tabelas 4.18, 4.19 e 4.20 apresentam as medidas de $|u_{min}|$ em função dos horizontes. Foram medidos valores próximos em ambos os casos. Conforme esperado, para os casos nominal e com garantia de estabilidade, o aumento dos horizontes incrementa a energia mínima necessária para a regulação dos erros de estado. Para a garantia de factibilidade, o relaxamento das restrições em todas as variáveis de decisão fornece um requisito energético mais uniforme em função dos horizontes.

Finalmente, avaliam-se os índices de desempenho IAE e TV em função dos horizontes. Para o caso nominal a Tabela 4.21 apresenta valores bem próximos a menos do horizonte máximo avaliado ($T_P = T_C = 10$) que apresentou valores elevados.

Com a garantia de estabilidade, a Tabela 4.22 mostra pequeno aumento de ambos os índices com o aumento das ações de controle, indicando redução da capacidade de regulação e rejeição de ruído quando se aumentam os horizontes, comportamento justificado pelo aumento da quantidade de variáveis a serem atraídas para a região terminal das restrições, além do efeito do custo terminal, que deterioram o custo computacional.

Para a garantia de factibilidade, a Tabela 4.23 mostra, mais uma vez, uniformidade dos índices de desempenho, independente dos horizontes adotados, indicando a capacidade de regulação dos erros de estado com baixas variações nos esforços de controle.

De maneira geral, estes resultados indicam que qualquer uma das abordagens propostas é viável, já que possuem requisitos energéticos e computacionais similares para baixos horizontes de predição e controle, comportamento intrínseco à abordagens não lineares. Além disso, considerando todos os critérios estudados, conclui-se a viabilidade da manutenção de $T_p = T_c = 3T_s$ para os próximos experimentos, tanto para o seguimento de caminhos visuais quanto para o controle de formação.

Tabela 4.12: Análise estatística do tempo de processamento - *VPF* nominal.

$T_p = T_c$	T_{proc}	$var(T_{proc})$	$dp(T_{proc})$
1	4,595	0,898	0,948
3	4,800	0,567	0,753
5	5,132	1,396	1,181
7	6,240	114,013	10,678
10	10,672	3,941e+03	62,780
15	39,092	4,312e+04	207,660

Tabela 4.13: Análise estatística do tempo de processamento - *VPF* com garantia de estabilidade.

$T_p = T_c$	T_{proc}	$var(T_{proc})$	$dp(T_{proc})$
1	4,768	2,206	1,485
3	5,074	2,764	1,663
5	7,066	782,469	27,973
7	9,071	1,275e+03	35,700
10	8,193	935,836	30,591
15	41,541	4,180e+04	204,456

Tabela 4.14: Análise estatística do tempo de processamento - *VPF* com garantia de factibilidade.

$T_p = T_c$	T_{proc}	$var(T_{proc})$	$dp(T_{proc})$
1	4,545	0,427	0,653
3	5,513	121,643	11,029
5	7,072	758,819	27,547
7	7,307	886,876	29,871
10	7,334	413,654	20,339
15	7,710	3,295	1,815

Tabela 4.15: Análise estatística de J_{min} - VPF nominal.

$T_p = T_c$	J_{min_m}	$var(J_{min})$	$dp(J_{min})$
1	0,008	1,383e-04	0,019
3	0,011	2,850e-04	0,017
5	0,013	3,729e-04	0,019
7	0,020	0,002	0,040
10	0,049	0,064	0,253
15	0,277	0,543	0,737

Tabela 4.16: Análise estatística de J_{min} - VPF com garantia de estabilidade.

$T_p = T_c$	J_{min_m}	$var(J_{min})$	$dp(J_{min})$
1	0,043	0,044	0,209
3	0,117	0,701	0,838
5	0,032	0,002	0,043
7	0,085	0,023	0,151
10	0,157	0,273	0,523
15	0,539	1,547	1,243

Tabela 4.17: Análise estatística de J_{min} - VPF com garantia de factibilidade.

$T_p = T_c$	J_{min_m}	$var(J_{min})$	$dp(J_{min})$
1	0,009	2,037e-04	0,014
3	0,013	4,313e-04	0,021
5	0,019	0,003	0,055
7	0,017	4,568e-04	0,021
10	0,033	0,002	0,044
15	0,140	0,050	0,224

Tabela 4.18: Análise estatística de $|u_{min}|$ - *VPF* nominal.

$T_p = T_c$	$ u_{min_m} $	$var(u_{min})$	$dp(u_{min})$
1	0,126	0,013	0,112
3	0,125	0,014	0,118
5	0,128	0,013	0,113
7	0,128	0,014	0,120
10	0,125	0,013	0,116
15	0,222	0,116	0,340

Tabela 4.19: Análise estatística de $|u_{min}|$ - *VPF* com garantia de estabilidade.

$T_p = T_c$	$ u_{min_m} $	$var(u_{min})$	$dp(u_{min})$
1	0,157	0,075	0,274
3	0,199	0,184	0,429
5	0,127	0,012	0,111
7	0,131	0,013	0,114
10	0,128	0,013	0,113
15	0,163	0,012	0,141

Tabela 4.20: Análise estatística de $|u_{min}|$ - *VPF* com garantia de factibilidade.

$T_p = T_c$	$ u_{min_m} $	$var(u_{min})$	$dp(u_{min})$
1	0,131	0,011	0,107
3	0,132	0,012	0,108
5	0,135	0,013	0,113
7	0,129	0,011	0,104
10	0,133	0,012	0,107
15	0,133	0,011	0,107

Tabela 4.21: Análise de desempenho - *VPF* nominal

$T_p = T_c$	IAE		TV	
	z_{IAE} [m]	$\theta_{r_{IAE}}$ [rad]	v_{TCV} [m/s]	ω_{TCV} [rad/s]
1	1,022	6,567	0,827	13,040
3	1,015	6,490	0,815	15,798
5	1,047	6,760	0,865	13,872
7	1,031	6,642	0,891	15,673
10	1,045	6,730	0,859	13,995
15	1,849	10,712	1,654	16,961

Tabela 4.22: Análise de desempenho - *VPF* com garantia de estabilidade

$T_p = T_c$	IAE		TV	
	z_{IAE} [m]	$\theta_{r_{IAE}}$ [rad]	v_{TCV} [m/s]	ω_{TCV} [rad/s]
1	1,108	7,297	0,842	16,478
3	1,101	7,228	0,877	18,918
5	1,025	6,661	0,859	13,922
7	1,043	6,808	0,835	14,438
10	1,017	6,643	0,875	14,660
15	1,309	8,150	1,153	16,319

Tabela 4.23: Análise de desempenho - *VPF* com garantia de factibilidade

$T_p = T_c$	IAE		TV	
	z_{IAE} [m]	$\theta_{r_{IAE}}$ [rad]	v_{TCV} [m/s]	ω_{TCV} [rad/s]
1	1,093	7,046	0,983	14,245
3	1,111	7,139	0,919	13,292
5	1,100	7,178	0,985	13,564
7	1,080	6,989	0,889	13,548
10	1,117	7,191	0,867	13,684
15	1,119	7,262	0,901	13,486

Erros no caminho visual.

As análises de desempenho das proposições de controladores *NMPC* para seguimento de caminhos visuais são concluídas através da geração de erros do caminho visual inicialmente capturado pela câmera. Basicamente, posiciona-se o robô em relação ao caminho para que existam três tipos de erros: Caminho visto como uma linha horizontal próximo ao horizonte, caminho visto como uma linha vertical à direita do ponto principal e bifurcação no plano da imagem.

Objetiva-se analisar o comportamento das estratégias em cenários em que existam grandes perturbações no ambiente de navegação. Tal análise é útil para definir os requisitos dos caminhos variantes no tempo, implementados para os controladores de formação utilizando informação visual.

No primeiro caso, os erros de estado não são inicialmente muito elevados mas, a velocidade de navegação constante gera, nos instantes de amostragem seguintes, grande perturbação nos estados que deve ser rejeitada em tempo hábil, sob pena de perda do caminho do campo visual. Além disso, situação similar ocorre quando o robô chega ao fim do caminho, podendo ser útil para a continuidade do seguimento.

As Figuras 4.34, 4.35 e 4.36 ilustram os resultados obtidos para $v = 0,1\text{m/s}$. Verifica-se que ambos os controladores conseguiram regular os erros de seguimento com esforços de controle satisfatórios. Neste caso, a técnica baseada na garantia de factibilidade apresentou regulação mais suave e num intervalo de tempo menor, viabilizando a análise quando existem perturbações iniciais. Para velocidades inferiores, desempenhos similares são obtidos.

Em seguida o caminho é posicionado numa das extremidades do plano da imagem, gerando erros de estado inicialmente elevados, que estão fora dos limites das restrições. As Figuras 4.37, 4.38 e 4.39 apresenta os resultados para $v = 0,1\text{m/s}$. Observa-se que, para os casos nominal e com garantia de estabilidade, os erros regulados momentaneamente, porém, nos instantes posteriores os erros crescem no sentido oposto até a perda total do caminho do campo visual. Para garantia de factibilidade, os erros são regulados satisfatoriamente. Este teste é particularmente importante já que demonstra em termos práticos a importância da factibilidade recursiva para a garantia de estabilidade.

Finalmente, verifica-se o comportamento das estratégias quando existe bifurcação no caminho visual, gerando dois caminhos ao longo do horizonte. Inicialmente, para $v = 0,1\text{m/s}$, foram adquiridos os resultados das Figuras 4.40, 4.41 e 4.42. Observa-se que ambas as estratégias conseguiram regular os erros de estado, sendo mais uma vez destacado o desempenho quando se utiliza a garantia de fatibilidade. Para $v = 0,2\text{m/s}$, apenas esta técnica atende aos requisitos de regulação, conforme ilustrado nas Figuras 4.43, 4.44 e 4.45.

Estes resultados comprovam melhor desempenho dos controladores *NMPC* com garantia de factibilidade para seguimento de caminhos visuais, sendo portanto a abordagem escolhida para o desenvolvimento dos controladores de formação utilizando informações visuais no próximo capítulo.

Caminho horizontal no plano da imagem

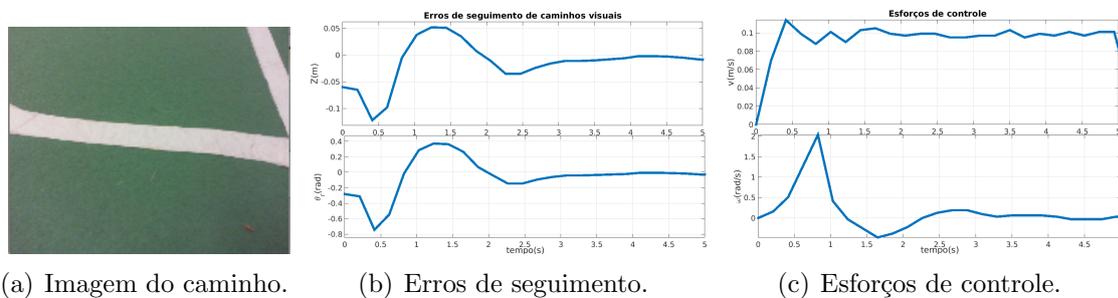


Figura 4.34: Caminho horizontal no plano da imagem - *NMPC* nominal - $v = 0, 1m/s$.

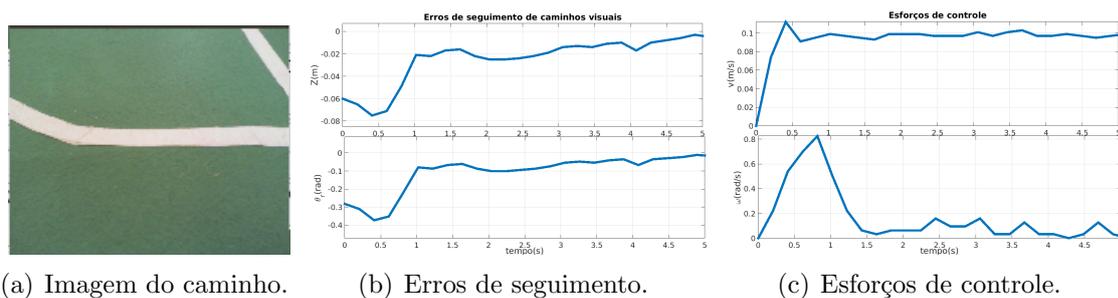


Figura 4.35: Caminho horizontal no plano da imagem - *NMPC* com garantia de estabilidade - $v = 0, 1m/s$.

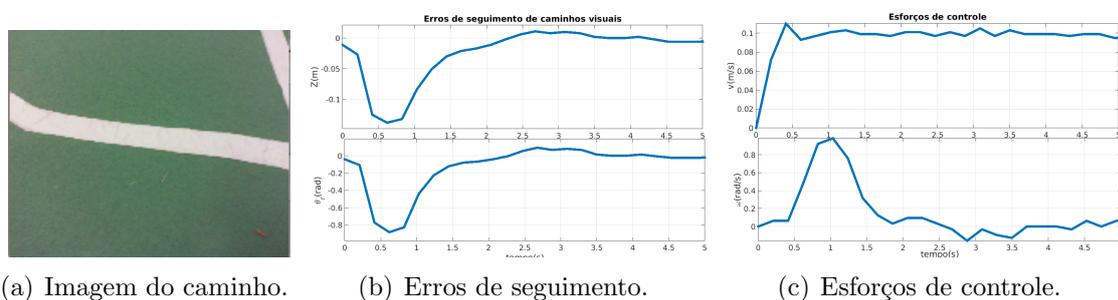


Figura 4.36: Caminho horizontal no plano da imagem - *NMPC* com garantia de factibilidade - $v = 0, 1m/s$.

Caminho vertical no plano da imagem

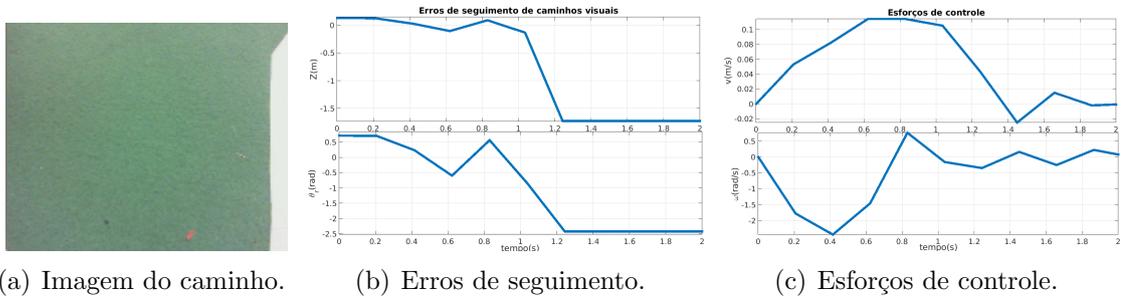


Figura 4.37: Caminho vertical no plano da imagem - *NMPC* nominal - $v = 0, 1m/s$.

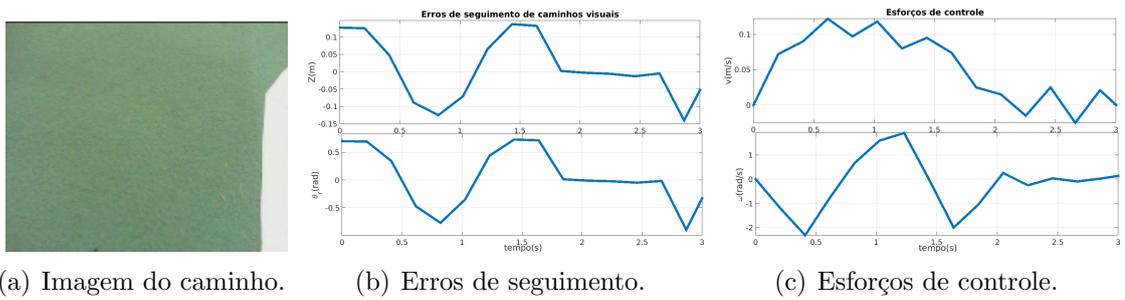


Figura 4.38: Caminho vertical no plano da imagem - *NMPC* com garantia de estabilidade - $v = 0, 1m/s$.

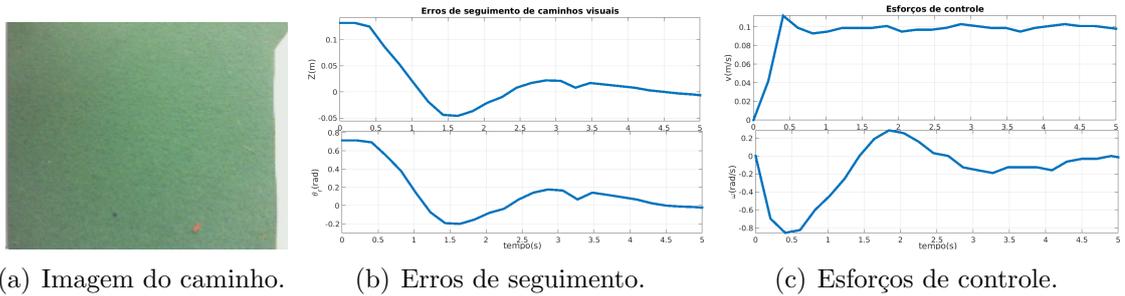


Figura 4.39: Caminho vertical no plano da imagem - *NMPC* com garantia de factibilidade - $v = 0, 1m/s$.

Bifurcação no plano da imagem

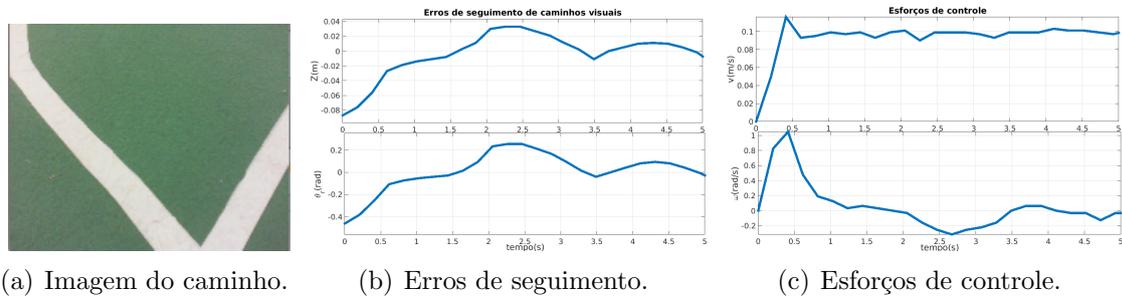


Figura 4.40: Caminho vertical no plano da imagem - *NMPC* nominal - $v = 0,1m/s$.

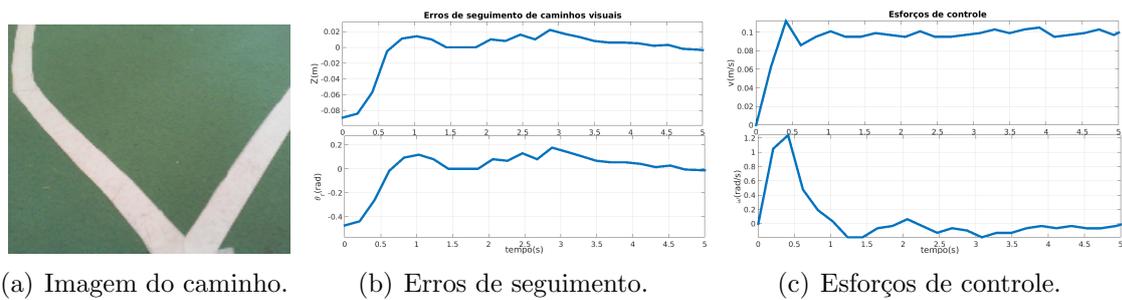


Figura 4.41: Caminho vertical no plano da imagem - *NMPC* com garantia de estabilidade - $v = 0,1m/s$.

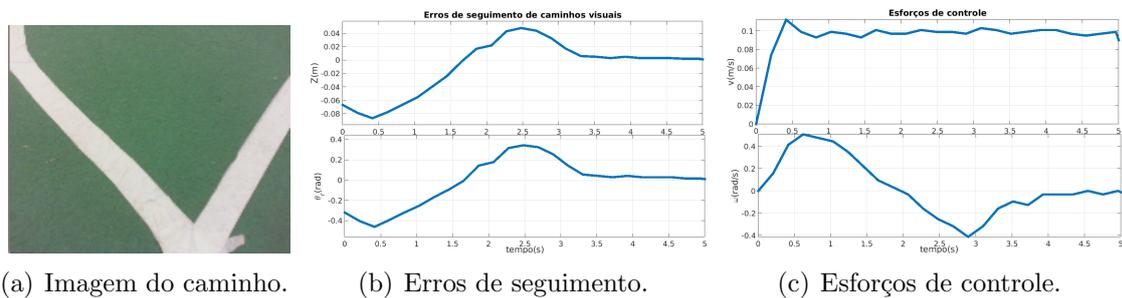


Figura 4.42: Caminho vertical no plano da imagem - *NMPC* com garantia de factibilidade - $v = 0,1m/s$.

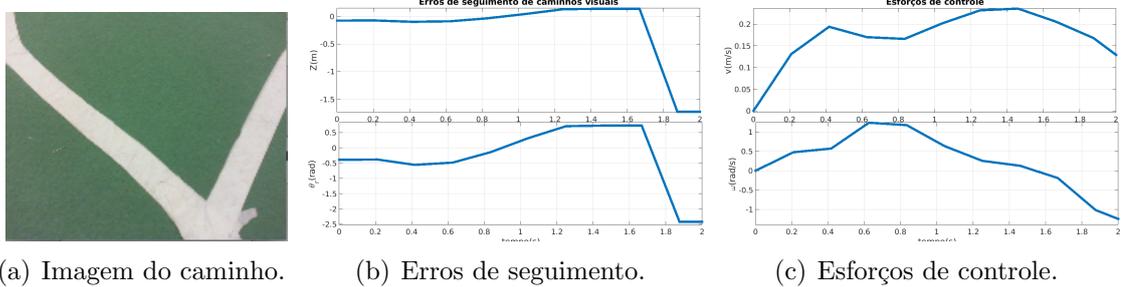


Figura 4.43: Caminho vertical no plano da imagem - *NMPC* nominal - $v = 0,2\text{m/s}$.

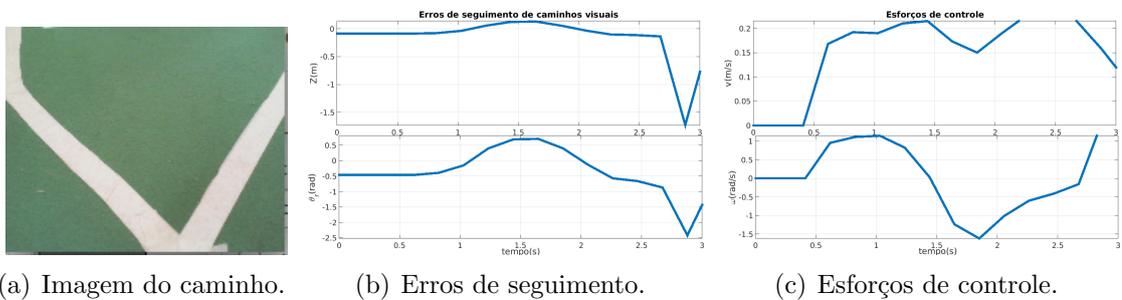


Figura 4.44: Caminho vertical no plano da imagem - *NMPC* com garantia de estabilidade - $v = 0,2\text{m/s}$.

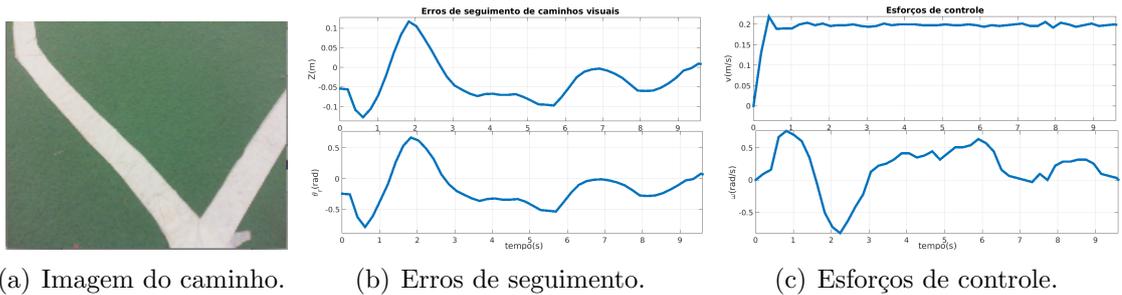


Figura 4.45: Caminho vertical no plano da imagem - *NMPC* com garantia de factibilidade - $v = 0,2\text{m/s}$.

4.5 Conclusões

Este capítulo foi dedicado à formalização e implementação dos controladores *NMPC* para seguimento de caminhos, tarefa definida para os robôs em formação. Foram adquiridos resultados experimentais que validam as abordagens propostas. Para quadricópteros as validações foram feitas através de simulações considerando ambientes realísticos, produzindo resultados bastante conclusivos.

Inicialmente, a proposta original para seguimento de caminhos pré definidos baseado em *NMPC* para robôs omnidirecionais foi avaliada, sendo possível confirmar sua aplicabilidade, desde que os erros de estado não sejam muito elevados. Sob ponto de vista das aplicações, a previsão de regulação de erros elevados é um requisito importante, já que pode haver variações nos parâmetros do caminho, seja por uma necessidade intrínseca ou por imperfeições do ambiente de navegação.

Foi possível confirmar, ainda para os robôs omnidirecionais, que a proposta para garantia de factibilidade baseada no relaxamento das restrições permite o tratamento destes erros elevados com desempenho satisfatório sob o ponto de vista dos requisitos de controle e dos custos computacionais.

Em seguida a abordagem original para seguimento de caminhos, foi estendida para um caso tridimensional, objetivando a utilização de quadricópteros. Considerando algumas hipóteses simplificadoras para a especificação dos caminhos de referência, foram adquiridos resultados tanto para caminhos fixos como variantes no tempo. Nesta etapa, foi possível confirmar que para o caso nominal, em que não se garante factibilidade, não é possível regular erros os elevados que surgem no momento da variação dos parâmetros do caminho de referência, sendo mais vez a técnica para garantia de factibilidade aplicada com sucesso para a resolução deste problema.

Atendendo ao objetivo de generalização das aplicações, uma estratégia baseada em visão computacional foi proposta para robôs diferenciais. Mais especificamente, o problema de seguimento de caminhos visuais foi definido e os controladores *NMPC* implementados. Avaliações prévias do comportamento do sistema visual acoplado ao eletromecânico (robô móvel) evidenciaram a necessidade de se adotar alguma estratégia para garantia de estabilidade, especialmente para plataformas experimentais de baixo custo. Neste caso, utilizou-se a técnica para garantia de estabilidade baseada em termo de penalidade terminal e zona terminal para as restrições.

Os resultados obtidos com este tipo de plataforma evidenciaram a aplicabilidade prática da técnica para três abordagens baseadas em *NMPC*, já que, mesmo com todas as imperfeições, foi possível satisfazer aos objetivos de controle. Além disso, confirma-se também a baixa representatividade das posturas medidas através do sistema odométrico e as vantagens dos controladores ótimos com relação ao proporcional clássico utilizado na abordagem original para seguimento de caminhos visuais.

Para uma plataforma experimental mais bem construída, foram adquiridos resultados ainda mais conclusivos considerando um caminho mais longo e com funções curvaturas mais variantes e descontínuas. Após apresentação de um caso inicial, para velocidades próximas das definidas para os testes com a plataforma anterior,

foram feitas avaliações considerando diversas velocidades, sendo possível concluir que, a única abordagem em que é possível definir maiores velocidades é aquela em que se garante factibilidade. Dos resultados obtidos, conclui-se que para ambos os controladores projetados, considerando as necessidades de variação das velocidades de navegação para coordenação dos movimentos, $v_{nav} = 0,05\text{m/s}$ é suficiente para o seguimento do caminho base.

Além disso foram feitas avaliações quando se variam os horizontes de predição e controle, ocasião em que se confirmou a inviabilidade da escolha de valores elevados para a implementação dos controladores *NMPC*, nominal e com garantia de estabilidade. Ainda nesta etapa, os critérios de desempenho *IAE* e *TV* evidenciaram a superioridade da técnica proposta para garantia de factibilidade de controladores *NMPC*, independente do horizonte escolhido.

De maneira geral, objetivando o projeto de controladores de formação baseados em todos os esquemas de *NMPC* para seguimento de caminhos apresentados neste capítulo, define-se os horizontes como sendo $T_p = T_c = 3T_s$.

Finalmente, foram feitas análises para o seguimento de caminhos visuais com erros iniciais elevados sendo, mais uma vez, confirmado melhor desempenho dos controladores *NMPC* com factibilidade garantida.

Capítulo 5

Controle de Formação

Caracterizada a tarefa para os robôs em formação, este capítulo é dedicado à apresentação das estratégias para coordenação dos movimentos de robôs diferenciais, omnidirecionais e quadricópteros.

Inicialmente, deve-se definir a configuração da formação e a estrutura de comunicação entre os robôs, já que se trata de uma etapa fundamental para especificação dos objetivos de controle. Neste contexto, as duas estruturas utilizadas, uma em que todos os robôs possuem a mesma importância relativa e outra baseada na estratégia *Líder-Seguidor* são caracterizadas sob a forma de grafos.

Em seguida, formaliza-se o problema de seguimento de caminho coordenado para robôs omnidirecionais e são fornecidos resultados experimentais tanto para formações fixas quanto variantes no tempo. Em diversas situações práticas, tais como desvio de obstáculo e escolta, variações nos parâmetros da formação são requisitadas, porém, no estudo aqui realizado, objetiva-se apenas gerar perturbações elevadas para análise mais criteriosa das estratégias de controle propostas. Resultados experimentais ratificam a importância da utilização de uma técnica para garantia de factibilidade de controladores *NMPC*.

Para os robôs diferenciais, os princípios da formalização realizada para os omnidirecionais são mantidos, sendo os modelos ajustados para contemplar a natureza não holonômica do robô. Resultados experimentais são adquiridos considerando um caminho de referência base que apresenta função curvatura variável e são considerados erros iniciais não nulos para enfatizar a importância da técnica proposta.

Posteriormente uma extensão para o caso tridimensional é proposta. São apresentados os requisitos de controle e as principais limitações desta abordagem. Nesta etapa os robôs móveis considerados são os quadricópteros e também são considerados caminhos e formações variantes.

Finalmente, mais uma vez objetivando generalização das aplicações, duas abordagens para controle de formação utilizando informações visuais são propostas para robôs diferenciais, a primeira define o conceito de câmeras e caminhos visuais virtuais e a segunda utiliza a estratégia Líder-Seguidor. Em ambos os casos, são fornecidos resultados experimentais obtidos através de uma arquitetura centralizada

e após análise de desempenho, confirmada a superioridade da segunda abordagem, são detalhadas as suas características fundamentais através de experimentos com a arquitetura descentralizada.

5.1 Caracterização da Formação

As exigências de controladores operando em condições extremas para modelos cada vez mais complexos evidenciaram a necessidade de uma modularização das tarefas. Isto se tornou possível com o aumento da eficiência das estratégias de comunicação, sendo viabilizada a utilização de sistemas distribuídos em diversas situações.

Com este tipo de sistema há uma redução significativa na complexidade dos projetos e nos custos computacionais, já que problemas mais simples são resolvidos localmente, o que implica na redução dos custos financeiros.

Os sistemas distribuídos, também conhecidos como sistemas estruturados ou descentralizados, são normalmente caracterizados por grafos. Tal caracterização é factível já que se trata um objeto matemático preciso e consolidado. Desta forma, uma análise descentralizada de sistemas dinâmicos complexos é realizada através do rigoroso arcabouço matemático da teoria dos grafos (Šiljak 1991).

Neste trabalho, um grupo de robôs cuja formação deve ser controlada é considerado como um sistema distribuído e a representação desta formação sob a forma de grafo determinará quais agentes devem se comunicar, bem como também suas importâncias relativas para o atendimento dos objetivos de controle.

Inicialmente, são considerados grafos não direcionados e conectados já que em parte das formações aqui consideradas o fluxo de informação deverá ser bidirecional e sempre haverá pelo menos um robô posicionado entre outros dois. A Figura 5.1 ilustra genericamente um grafo assim classificado. Nesta Figura, tem-se uma representação sob a forma de grafo de um conjunto de robôs a serem controlados em formação. Para facilitar o entendimento, apenas alguns vértices e arestas são apresentados, definindo um subgrafo. Desconsiderando as importâncias relativas dos vértices deste subgrafo, tal figura representa a árvore¹ de extensão mínima de um grafo completo que representa toda a formação.

O grafo de uma formação qualquer é dado por $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, sendo $\mathcal{V} = 1, 2, \dots, N$ o número de vértices do grafo, com cada vértice representando um robô e $\mathcal{A} \subset \mathcal{V} \times \mathcal{V}$ é o conjunto das arestas, com cada aresta representando um vetor relativo entre dois robôs.

Dois robôs i e j são considerados vizinhos se a aresta $A_{i,j} \in \mathcal{A}$. O conjunto de todos os robôs vizinhos a um dado robô i é denotado por $\mathcal{N}_i \in \mathcal{V}$. Tal conjunto determinará um critério para a definição de quais elementos da formação deverão se comunicar.

Dado um caminho de referência parametrizado em s a ser seguido pela formação, determina-se a distância desejada entre dois robôs i e j por uma função $p_{ij}(s)$, sendo

¹Na teoria dos grafos, uma árvore é um grafo conectado e acíclico.

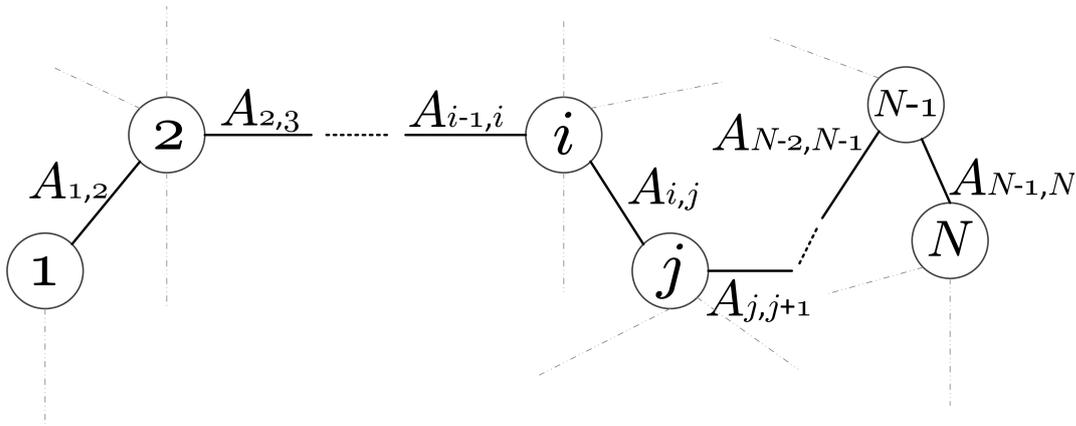


Figura 5.1: Representação de um conjunto de robôs sob a forma de grafo - Representação genérica.

a relação entre a posição do robô i com o robô j ao longo deste caminho dada por:

$$s_j = s_i + p_{ij}(s). \quad (5.1)$$

Como os grafos considerados são não direcionados, tem-se que $p_{ij} = -p_{ji}$.

Define-se um vetor de erros de formação $E \in \mathbb{R}^M$, sendo M o número total de arestas de \mathcal{G} . Este vetor possui componentes $s_{i,e} \in \mathbb{R}$ definidos da seguinte forma:

$$s_{i,e} = s_i - s_j + p_{ij}(s). \quad (5.2)$$

Considerando-se E_i um vetor com todos os componentes os K componentes de E que possuem um termo de acoplamento com o robô i , tem-se que:

$$E_i = \mathbf{C}_i \mathbf{s} + \mathbf{p}, \quad (5.3)$$

sendo:

$\mathbf{p} = [\dots, p_{ij}, \dots]^T$: o vetor das distâncias relativas;

$\mathbf{s} = [\dots, s_i - s_j, \dots]^T$: o vetor das posições curvilíneas relativas ao longo do caminho;

\mathbf{C}_i : a submatriz de incidência de $\mathcal{G} \in \mathbb{R}^{K \times M}$.

A matriz de incidência possui entradas binárias e define a estrutura de um grafo. Sua utilização proporciona cálculos eficientes dos erros numa formação generalizada.

A caracterização de quais elementos devem se comunicar é feita através da matriz de adjacência \mathbf{A} , que também possui uma estrutura binária. Trata-se de uma matriz quadrada de ordem N que possui entradas unitárias quando existe um vértice conectado entre dois nós e nulas em caso contrário.

De maneira geral, estas matrizes são ferramentas úteis para a representação das propriedades de um dado sistema distribuído e, por apresentarem estrutura binária,

são de fácil implementação computacional. Outras estruturas para a caracterização de grafos são estrutura de adjacência e conjuntos.

Os critérios para a escolha de uma representação específica são normalmente baseados em complexidade algorítmica. Como neste trabalho objetiva-se o cálculo do erros de formação em diferentes nós de processamento e determinação prévia do fluxo de informação entre os robôs, opta-se pela utilização das representações matriciais.

Outra grandeza que deve ser controlada para a manutenção da formação é a velocidade de navegação $u_{f_{i,e}}$. Define-se o erro de velocidade, relativo ao caminho base, para o robô i como segue:

$$u_{f_{i,e}} = \frac{P(\Gamma_r)}{P(\Gamma_i)} v_i - \frac{P(\Gamma_r)}{P(\Gamma_j)} v_j \quad (5.4)$$

sendo $P(\Gamma)$ a função comprimento total do caminho Γ .

Define-se o vetor erro de formação local para o robô i , quantidade utilizada como termo de acoplamento entre os robôs, da seguinte forma:

$$\mathbf{f}_{i,e} = \begin{bmatrix} s_{i,e} \\ u_{f_{i,e}} \end{bmatrix} \quad (5.5)$$

Quando a estratégia Líder-Seguidor é utilizada, tem-se a representação da Figura 5.2. Nesta representação tem-se um robô líder i e os seguidores $1, 2, \dots, N - 1, N$. Como neste caso apenas o robô líder envia informação para os seguidores, trata-se de um grafo direcionado, conectado e acíclico, mais especificamente, uma árvore. A menos da direcionalidade, os conceitos enunciados para a representação anterior continuam válidos.

Além disso, devido a importância do robô líder, o mesmo é definido como raiz, sendo deste modo o grafo desta formação definido como árvore enraizada. Para esta abordagem, a priori, não são definidos termos de acoplamento entre os agentes.

Maiores detalhes sobre a teoria dos grafos pode ser encontrada em (Godsil e Royle 2001).

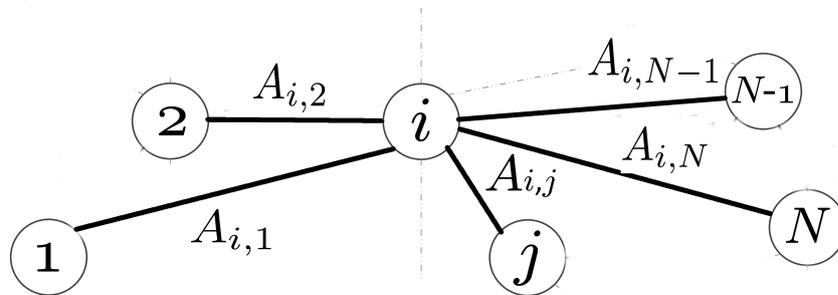


Figura 5.2: Representação de um conjunto de robôs sob a forma de grafo - Estratégia Líder-Seguidor.

5.2 Controle de Formação para Robôs Ominidirecionais

De maneira geral, tem-se um conjunto de N robôs móveis e N caminhos espaciais de referência Γ_i , com $i = 1, \dots, N$, sendo que o robô R_i deve seguir o caminho Γ_i . O objetivo de coordenar o movimento deste grupo de robôs pode ser alcançado através de ajustes nas velocidades de navegação v_{nav_i} e na taxa de progressão do veículo virtual \dot{s}_i .

Neste trabalho, inicialmente define-se um caminho de referência base Γ_r que deve ser seguido em formação pelo grupo de robôs. A Figura 5.3 ilustra esta situação para o caso de três robôs mantendo uma formação em triângulo entre si. Este caminho é parametrizado em função de s_i , é definido previamente considerando-se as necessidades de desvio de obstáculos e deve possuir características especiais como pode ser verificado em (Soetanto *et al.* 2003).

Considerando-se as posturas ao longo do caminho base nos pontos s_i como sendo $[x_{B_i}, y_{B_i}, \theta_{B_i}]^T$ e as distâncias transversais q_i entre estes pontos e o centro de massa dos robôs, obtêm-se as posturas de referência para cada robô i , de acordo com a expressão a seguir:

$$\mathbf{x}_{r_i} = \begin{bmatrix} x_{r_i} \\ y_{r_i} \\ \theta_{r_i} \end{bmatrix} = \begin{bmatrix} x_{B_i} - q_i \sin \theta_{B_i} \\ y_{B_i} + q_i \cos \theta_{B_i} \\ \theta_{B_i} \end{bmatrix}, \quad (5.6)$$

sendo θ_{B_i} o ângulo da tangente à curva de referência no ponto s_i com relação ao sistema de coordenadas do mundo.

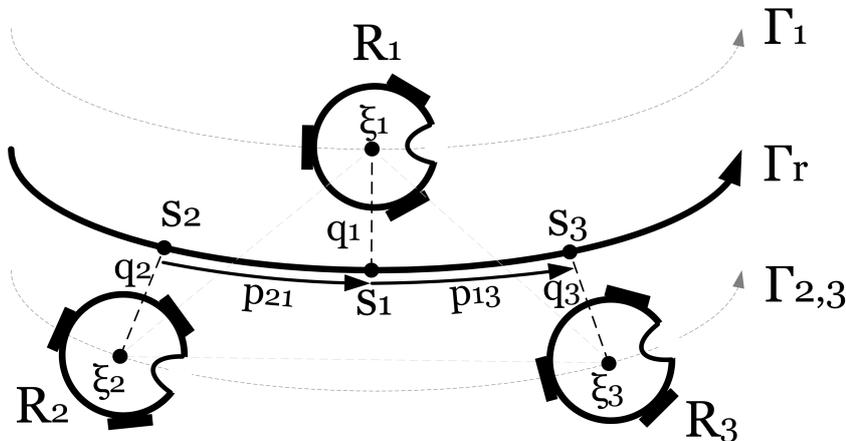


Figura 5.3: Seguimento de caminho coordenado: formação em triângulo.

Determina-se a velocidade de navegação de referência para o i -ésimo robô R_i através da relação entre o comprimento total do caminho de referência para este robô $P(\Gamma_i)$ e o comprimento total do caminho de referência base $P(\Gamma_r)$, ou seja:

$$v_{nav_{r_i}} = \frac{P(\Gamma_i)}{P(\Gamma_r)} v_{nav_B}, \quad (5.7)$$

sendo v_{nav_B} a velocidade de navegação base definida previamente para a formação. Tal formação estará controlada quando as velocidades relativas ao caminho base forem iguais, ou seja, considerando o exemplo da Figura 5.3 o robô interno deve possuir uma velocidade de navegação menor do que a dos robôs externos.

A velocidade angular de referência é determinada para cada robô i , como sendo:

$$\omega_{r_i} = k_i(\xi_i) v_{nav_{r_i}}, \quad (5.8)$$

sendo ξ_i o comprimento do trecho do caminho de referência para o i -ésimo robô e $k_i(\xi_i)$ a curvatura deste caminho dados por:

$$\xi_i = \frac{P(\Gamma_r)}{P(\Gamma_i)} s_i, \quad (5.9)$$

$$k_i(\xi_i) = \frac{k_r(s_i)}{1 - q_i k_r(s_i)}, \quad (5.10)$$

sendo $k_r(s_i)$ a curvatura do caminho de referência base.

O modelo de erros de estado para o i -ésimo robô, utilizando-se as variáveis definidas para os caminhos individuais é dado por:

$$\begin{aligned} \dot{x}_{e_i} &= y_{e_i} k_i(\xi_i) \dot{\xi}_i - \dot{\xi}_i + v_{nav_i} \cos \phi_{e_i}, \\ \dot{y}_{e_i} &= -x_{e_i} k_i(\xi_i) \dot{\xi}_i - v_{nav_i} \sin \phi_{e_i}, \\ \dot{\theta}_{e_i} &= \omega_i - \omega_{r_i}. \end{aligned} \quad (5.11)$$

Conforme já referido anteriormente, neste trabalho, os critérios para o fluxo de informação entre os robôs são dados em termos da teoria dos grafos (Szwarcfiter 1984). O objetivo é determinar a vizinhança de cada elemento do sistema para que se definam quais elementos necessitam se comunicar.

Para o caso da Figura 5.3, por exemplo, pode-se generalizar aquela formação sob a forma de grafo, determinando-se os vértices, as arestas, os vetores relativos e a matriz de adjacência, de onde conclui-se que, para este tipo de configuração, necessitam se comunicar os conjuntos de robôs $\{1,2,3\}$, $\{2,1\}$ e $\{3,1\}$ para a solução dos problemas locais.

Como a velocidade de navegação para cada robô deve ser coordenada com os elementos de sua vizinhança, v_{nav_i} não pode ser mantida constante. Assim, define-se uma nova variável de estado η_i , dada por:

$$\eta_i = v_{nav_i} - v_{nav_{r_i}}, \quad (5.12)$$

sendo $v_{nav_{r_i}}$ calculada por (5.7).

Assim, tem-se que:

$$\dot{\eta}_i = \epsilon_i - \gamma_i. \quad (5.13)$$

Sendo $\epsilon_i = \dot{v}_{nav_i}$ a aceleração atual e $\gamma_i = \dot{v}_{nav_{r_i}}$ aceleração de referência para o robô R_i .

Finalmente, definem-se os sinais de controle para o i -ésimo robô como sendo:

$$\mathbf{u}_{e_i} = \begin{bmatrix} u_{1_i} \\ u_{2_i} \\ u_{3_i} \\ u_{4_i} \end{bmatrix} = \begin{bmatrix} -\dot{\xi}_i + (\eta_i + v_{nav_{r_i}}) \cos \phi_i \\ (\eta_i + v_{nav_{r_i}}) \text{sen } \phi_i \\ \omega_i - \omega_{r_i} \\ \epsilon_i - \gamma_i \end{bmatrix}. \quad (5.14)$$

Assim, o modelo de erros de estado para cada robô R_i é dado como segue:

$$\dot{\mathbf{x}}_{e_i} = \begin{bmatrix} \dot{x}_{e_i} \\ \dot{y}_{e_i} \\ \dot{\theta}_{e_i} \\ \dot{\eta}_i \end{bmatrix} = \begin{bmatrix} y_{e_i} k_i(\xi_i) \dot{\xi}_i + u_{1_i} \\ -x_{e_i} k_i(\xi_i) \dot{\xi}_i + u_{2_i} \\ u_{3_i} \\ u_{4_i} \end{bmatrix}. \quad (5.15)$$

Considerando-se que as saídas são os próprios estados, o problema de controlar os movimentos do i -ésimo robô pode ser resumido como segue:

Encontrar $\dot{\xi}_i$, ω_{r_i} e ϵ_i , tais que u_{1_i} , u_{2_i} , u_{3_i} , u_{4_i} , x_{e_i} , y_{e_i} , θ_{e_i} e η_i sejam factíveis.

Tanto o modelo para seguimento de caminho básico (4.25), quanto o generalizado para um grupo de robôs (5.15), são não-lineares e requerem controladores de natureza estritamente regulatória. Além disso, como nos casos iniciais abordados neste trabalho, os caminhos de referência a serem seguidos são conhecidos, a ideia de predição pode ser satisfatoriamente aplicada.

5.2.1 Projeto dos controladores

Para a solução do problema de controle de formação, utiliza-se para cada robô, a seguinte função F para especificação de desempenho:

$$F_i(\mathbf{x}_i(\tau), \mathbf{u}_i(\tau), \mathbf{f}_{i,e}(\tau)) = \mathbf{x}_{i,e}^T \mathbf{Q}_i \mathbf{x}_{i,e} + \mathbf{u}_{i,e}^T \mathbf{R}_i \mathbf{u}_{i,e} + \sum_{(i,j) \in \mathcal{A}} \mathbf{f}_{i,e}^T \mathbf{W}_i \mathbf{f}_{i,e} \quad (5.16)$$

sendo:

\mathcal{A} : Conjunto de vetores relativos entre os robôs;

\mathbf{Q}_i , \mathbf{R}_i , \mathbf{W}_i : matrizes definidas positivas que ponderam desvios nos valores requeridos.

Assim, o critério a ser minimizado possui fatores relativos aos caminhos individuais a serem seguidos por cada robô e fatores de coordenação dos movimentos entre os mesmos. Este acoplamento via função objetivo permite a manutenção da formação através de ajustes nas velocidades de navegação e nas taxas de atualização nas referências de postura para cada robô.

Após o problema de otimização no instante t_k ser resolvido para R_i , os comprimentos de caminho atuais e os ótimos preditos, além das velocidades de navegação atuais e preditas, são transmitidos para a sua vizinhança. Deve-se notar que os referidos comprimentos devem ser convertidos para o sistema de coordenadas do caminho de referência base antes de serem transmitidos.

Assim como nas implementações do capítulo anterior, existem controladores *PID* embarcados em cada robô e o esquema de controle está ilustrado na Figura 5.4

5.2.2 Resultados Experimentais

Caracterizado o desempenho dos controladores para seguimento de caminho, procede-se com a inclusão dos termos de acoplamento às funções objetivo dos controladores de cada um dos robôs da formação. Conforme já elucidado, novos parâmetros de sintonia são adicionados nesta etapa, sendo especial atenção dedicada àqueles que ponderam os erros de formação.

São apresentados resultados de experimentos adquiridos através de uma arquitetura descentralizada composta de um robô real (R_1 , Figura 5.5) e dois virtuais (R_2 e R_3). Isto se deve à indisponibilidade de três robôs omnidirecionais reais. Porém, além de permitir uma avaliação inicial das propostas, esta forma de validação permite uma avaliação eficiente dos fundamentos em casos de formações com robôs diferentes, como por exemplo, diferenciais ou quadricópteros.

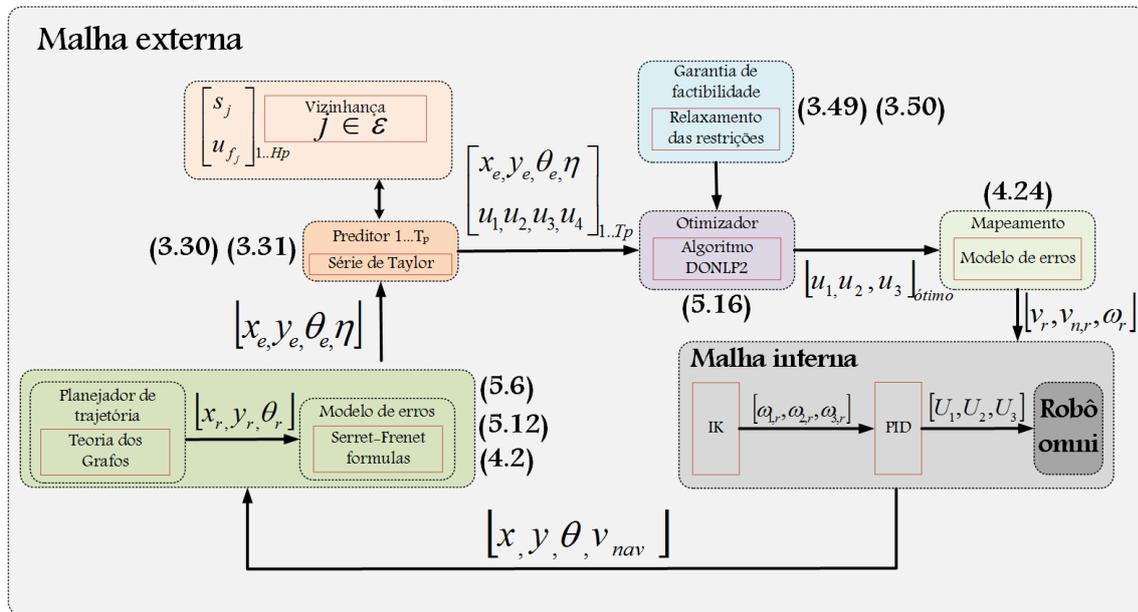


Figura 5.4: Estrutura de controle para cada robô omnidirecional.

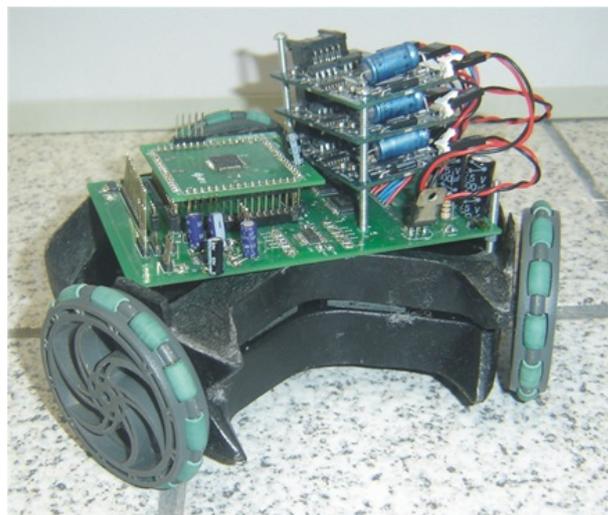


Figura 5.5: Robô omnidirecional utilizado.

Inicialmente são consideradas formações triangulares conforme a representação da Figura 5.3. Neste caso a matriz de adjacência possui a seguinte estrutura:

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (5.17)$$

Deste modo, deve haver fluxo mútuo de informações entre os R_1 e R_2 e entre R_1 e R_3 .

Definem-se caminhos de referência circulares para cada robô. Tais caminhos são caracterizados em função do seguinte caminho de referência base Γ_B :

$$\begin{aligned} x(s_i) &= r_B \cos(s_i); \\ y(s_i) &= r_B \sin(s_i), \end{aligned} \quad (5.18)$$

Sendo $r_B = 1$ o raio do caminho de referência base utilizado nos experimentos. A referência angular continua sendo o ângulo da tangente ao caminho no ponto de referência atual.

As referências para cada robô são calculadas por (5.6) e a velocidade de navegação ao longo do caminho base v_{nav_B} é fixada como sendo igual a $0,5m/s$. A posição inicial para o caminho base foi $x_0 = 1m$ e $y_0 = 0m$ e os parâmetros da formação foram os seguintes: $p_{12} = -0,5m$, $p_{13} = 0,5m$, $q_1 = 0,5m$, $q_2 = -0,5m$, $q_3 = -0,5m$.

Considerando o desempenho obtido com a estratégia para seguimento de caminho, nos experimentos realizados nesta etapa os horizontes de predição T_P e de controle T_C foram mantidos como sendo iguais a $3T_s$, sendo T_s definido neste caso como sendo $0,05s$ devido as características do robô. Os outros parâmetros de sintonia são ajustados em cada experimento e as restrições para o problema de controle de formação são dadas como segue:

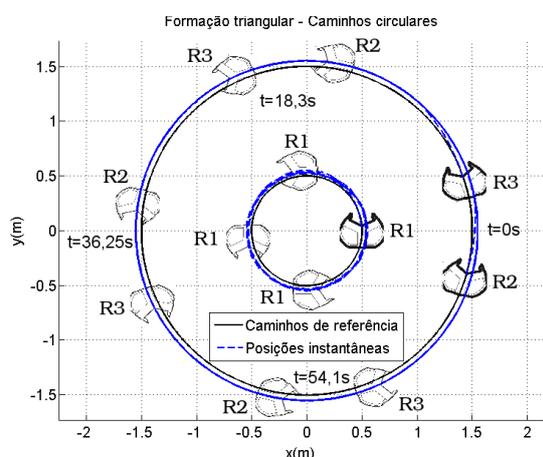
$$\begin{aligned} |u_1| &\leq 0,1; \quad |u_2| \leq 0,1; \quad |u_3| \leq 0,5; \quad |u_4| \leq 0,5; \\ |x_{e_i}| &\leq 0,05 \text{ m}; \quad |y_{e_i}| \leq 0,05 \text{ m}; \\ |\theta_{e_i}| &\leq 0,1 \text{ rad}; \quad |\eta_i| \leq 0,02 \text{ m/s}. \end{aligned}$$

Considerando um caso ideal, onde os erros iniciais são nulos e não há variações nos parâmetros da formação, para os parâmetros de sintonia apresentados na Tabela 5.1, foram obtidos os resultados ilustrados na Figura 5.6.

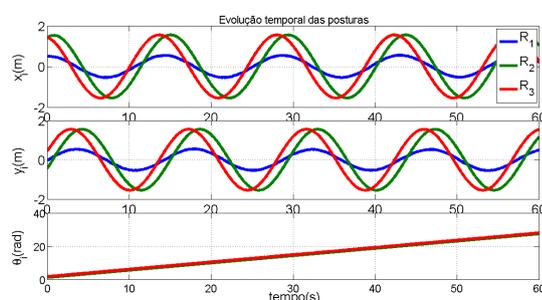
Observa-se o atendimento dos objetivos, já que os caminhos foram seguidos corretamente e os erros de formação se mantiveram inferiores a $0,05cm$ mesmo sem nenhuma restrição explícita. Os erros de seguimento de caminho obtidos com este experimento (5.6(d)) utilizaram toda a margem definida através das restrições, servindo de indicativo de que devem ser feitos novos ajustes dos parâmetros de sintonia.

Tabela 5.1: Matrizes de ponderação para os controladores de formação - Robôs omnidirecionais - Caso ideal.

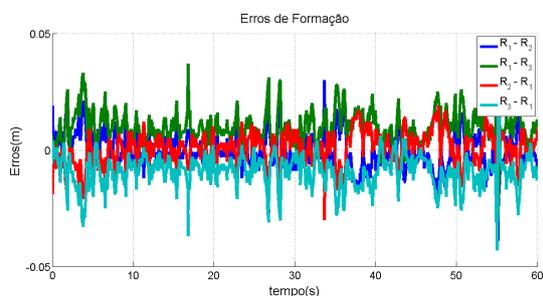
Q_i	$diagonal(0,1; 0,1; 0,0001; 0,1)$
R_i	$diagonal(0,005; 0,005; 0,000005; 0,005)$
W_i	$diagonal(0,05; 0,0001)$



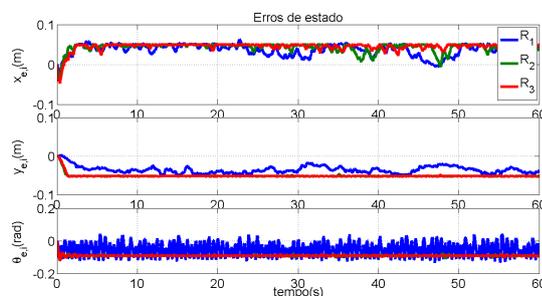
(a) Posturas no plano xy - Instantâneos por volta.



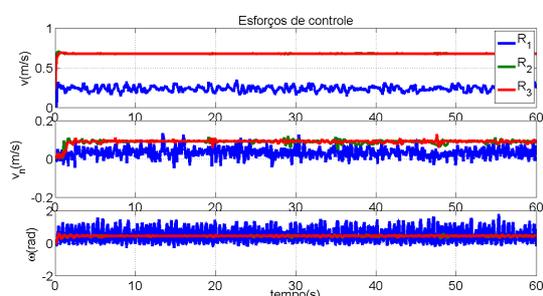
(b) Evolução temporal das posturas.



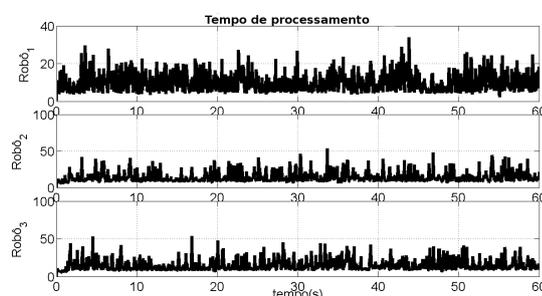
(c) Erros de formação.



(d) Erros de seguimento de caminho.



(e) Esforços de controle.



(f) Tempo de processamento.

Figura 5.6: Controle de formação triangular com caminhos circulares - Robôs omnidirecionais - Caso ideal.

Os esforços de controle apresentados na Figura 5.6(e) são compatíveis com a plataforma experimental e o tempo de processamento 5.6(e) se manteve praticável.

A proposta para garantia de factibilidade baseada no relaxamento das restrições possibilita a utilização da estratégia de controle de formação em casos onde são necessárias mudanças nos caminhos e nas formações. Neste caso, é necessário modificar alguns parâmetros de sintonia em função de objetivos específicos, tais como redução do tempo de regulação da formação ou dos erros de estado.

Os novos parâmetros são apresentados na Tabela 5.2. Deste modo, nesta etapa realiza-se um experimento em que, a cada volta no caminho circular, há uma modificação nos parâmetros da formação e nos caminhos de referência para cada robô. Esta implementação consiste na principal adaptação à estratégia originalmente proposta, que além de não explicitar os termos da formação nas funções objetivos locais, não considera as necessidades de modificação na formação ou no caminhos. Como situação prática onde isto seria necessário, pode-se citar o controle de formação considerando desvios de obstáculos.

Tabela 5.2: Matrizes de ponderação para os controladores de formação - Robôs omnidirecionais - Caminhos e formações variantes no tempo.

\mathbf{Q}_i	$diagonal(0,005; 0,005; 0,000005; 0,005)$
\mathbf{R}_i	$diagonal(0,0005; 0,0005; 0,0000005; 0,0005)$
\mathbf{W}_i	$diagonal(0,08; 0,0001)$

O experimento realizado consiste em oito passos, objetivando espelhar a formação em triângulo anteriormente definida ao longo do caminho base Γ_B :

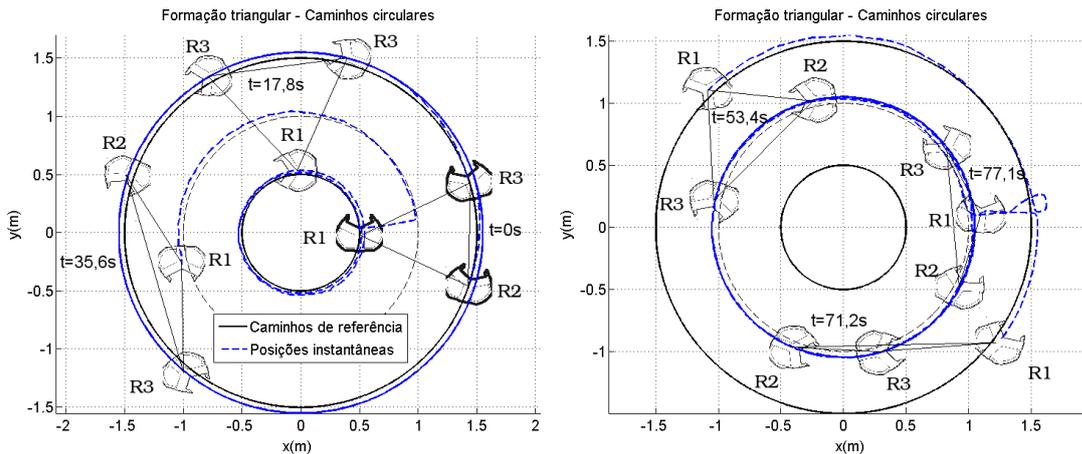
- **Primeira volta:** Formações e caminhos idênticos ao caso original;
- **Segunda volta:** Incremento no espaçamento relativo ($p_{12} = -0.6\text{m}$, $p_{13} = 0.6\text{m}$);
- **Terceira volta:** Mudança do caminho de referência do robô₁ para o caminho base ($q_1 = 0\text{m}$);
- **Quarta volta:** Mudança do caminho de referência do robô₂ para o caminho base ($q_2 = 0\text{m}$);
- **Quinta volta:** Mudança do caminho de referência do robô₃ para o caminho base ($q_3 = 0\text{m}$);

- **Sexta volta:** Mudança do caminho de referência do robô₁ para o caminho definido anteriormente para o robô₂ e para o robô₃ ($q_1 = -0.5\text{m}$);
- **Sétima volta:** Mudança do caminho de referência do robô₂ para o caminho definido anteriormente para o robô₁ ($q_2 = 0.5\text{m}$);
- **Oitava volta:** Mudança do caminho de referência do robô₃ para o caminho definido anteriormente para o robô₁ ($q_1 = 0.5\text{m}$);

Ainda com um robô real (R_1) e dois virtuais (R_2 e R_3) as Figuras 5.7 e 5.8 ilustram os resultados obtidos. A Figura 5.7(a) e 5.7(b) ilustra os resultados através das posturas instantâneas obtidas para os primeiros 77,1s de movimento. A continuação do movimento até 116,8s é ilustrado na Figura 5.8(a). Na Figura 5.8(b) tem-se uma representação mais precisa da movimentação, sendo verificado o atendimento dos objetivos em cada passo. A Figura 5.8(c) ilustra que os erros de formação foram regulados de maneira satisfatória, assim como também os erros de seguimento de caminho apresentados na Figura 5.8(d). Em regime permanente, tais valores se mantiveram inferiores aos limites das restrições.

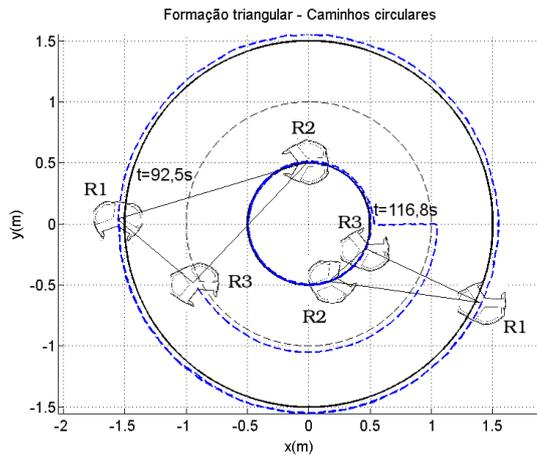
Os esforços de controle apresentados na Figura 5.8(e) permanecem limitados a valores praticáveis com robô utilizado, com exceção dos picos quando a este é solicitada mudança para o caminho externo (sexta volta). A Figura 5.8(f) comprova a manutenção da eficiência computacional da estratégia também para este caso.

Estes resultados foram reunidos e publicados em (Ribeiro *et al.* 2013) e (Trindade Ribeiro *et al.* 2013).

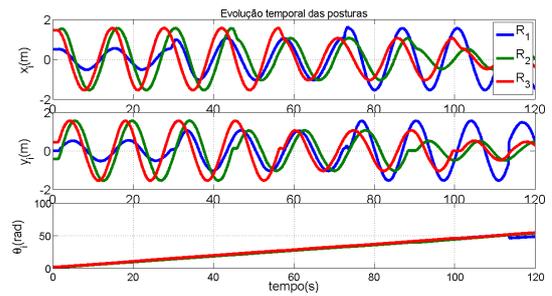


(a) Posturas no plano xy (0 - 35,6s) - Instantâneas por volta. (b) Posturas no plano xy (53,4s - 77,1s) - Instantâneas por volta.

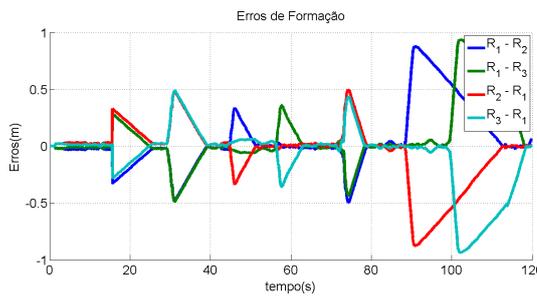
Figura 5.7: Controle de formação triangular com caminhos circulares - Robôs omnidirecionais - Caminhos e formações variantes no tempo.



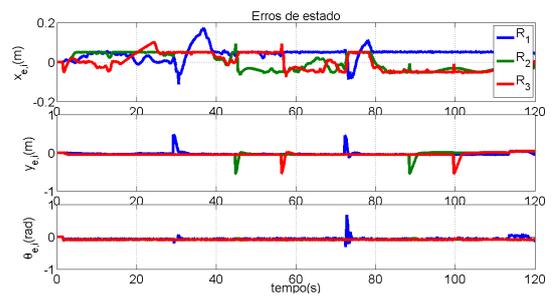
(a) Posturas no plano xy (92,5s - 116,8s) - Instantâneos por volta.



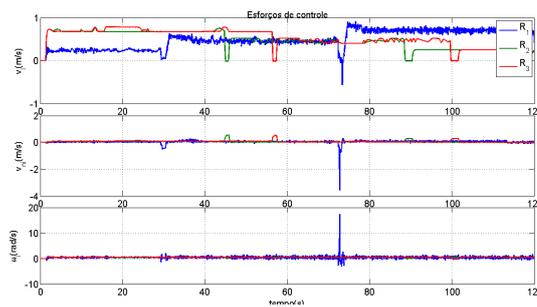
(b) Evolução temporal das posturas.



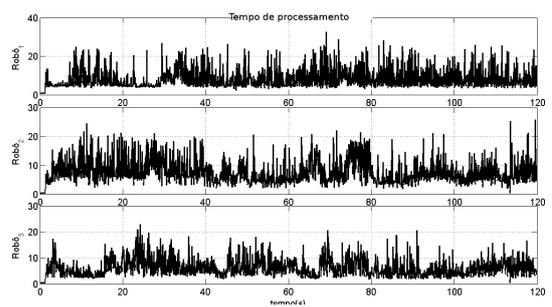
(c) Erros de formação.



(d) Erros de estado.



(e) Esforços de controle.



(f) Esforços computacionais.

Figura 5.8: Controle de formação triangular com caminhos circulares - Robôs omnidirecionais - Caminhos e formações variantes no tempo (*continuação*).

5.3 Controle de Formação para Robôs Diferenciais

A modelagem do controle de formação para robôs diferenciais segue os mesmos princípios utilizados na seção anterior para robôs omnidirecionais, sendo necessário apenas a alteração do modelo para que seja levada em consideração a restrição não holonômica inerente a este tipo de robô. Neste caso, definem-se as seguintes entradas:

$$\mathbf{u}_i = \begin{bmatrix} u_{1_i} \\ u_{2_i} \\ u_{3_i} \end{bmatrix} = \begin{bmatrix} -\dot{\xi}_i + (\eta_i + v_{r_i}) \cos \theta_{e_i} \\ \omega_i - k_i(\xi_i) \dot{\xi}_i \\ \epsilon_i - \gamma_i \end{bmatrix}. \quad (5.19)$$

Assim, o modelo dos erros de estado para cada robô é dado como segue:

$$\dot{\mathbf{x}}_i = \begin{bmatrix} \dot{x}_{e_i} \\ \dot{y}_{e_i} \\ \dot{\theta}_{e_i} \\ \dot{\eta}_i \end{bmatrix} = \begin{bmatrix} y_{e_i} k_i(\xi_i) \dot{\xi}_i + u_{1_i} \\ -x_{e_i} k_i(\xi_i) \dot{\xi}_i + (\eta_i + v_{r_i}) \sin \theta_{e_i} \\ u_{2_i} \\ u_{3_i} \end{bmatrix}. \quad (5.20)$$

Considerando que as saídas são os próprios estados, o problema de controlar os movimentos do *i*-ésimo robô pode ser resumido da seguinte forma:

Encontrar $\dot{\xi}_i$, ω_i e ψ_i , tal que u_{1_i} , u_{2_i} , u_{3_i} , x_{e_i} , y_{e_i} , θ_{e_i} e η_i sejam factíveis.

Considerando a mesma estrutura da função objetivo utilizada para robôs omnidirecionais e que também existem controladores *PID* embarcados, o esquema de controle está ilustrado na Figura 5.9

5.3.1 Resultados experimentais

Esta subseção apresenta resultados experimentais obtidos para o controle de formação de três robôs diferenciais reais (Figura 5.10) ao longo de um caminho de referência fixo e considerando erros iniciais elevados, tanto para os estados, quanto para a formação. Neste caso, a mesma estrutura da matriz de adjacência dada por (5.17) é utilizada.

Todos os experimentos foram feitos considerando um período de amostragem fixo $T_s = 0.5s$. A velocidade de navegação ao longo do caminho base foi definida como sendo $u_{f_{b_i}} = 0,05m/s$. Estes valores foram definidos em decorrência das características dos robôs comerciais utilizados (DFR 2014). Tais robôs são naturalmente lentos (tipo tanque), comunicam-se através do padrão *zigbee* e utilizam a plataforma Arduino. O principal problema das implementações práticas com estes robôs são os ruídos de medição de velocidade, já que são utilizados *encoders* analógicos de baixa resolução o que inviabiliza a utilização de soluções elementares para filtragem.

Considerando as não linearidades e imperfeições já apresentadas, os horizontes foram definidos como sendo $T_p = T_c = 3$ amostras.

Em todos os testes, foram utilizados caminhos senoidais e formações triangulares. O caminho base foi definido como segue:

$$x(s_i) = x_0 + s_i; \quad (5.21)$$

$$y(s_i) = y_0 + A_b \text{sen}(s_i). \quad (5.22)$$

sendo x_0 e y_0 as posições iniciais definidas para o caminho base e A_b a sua amplitude.

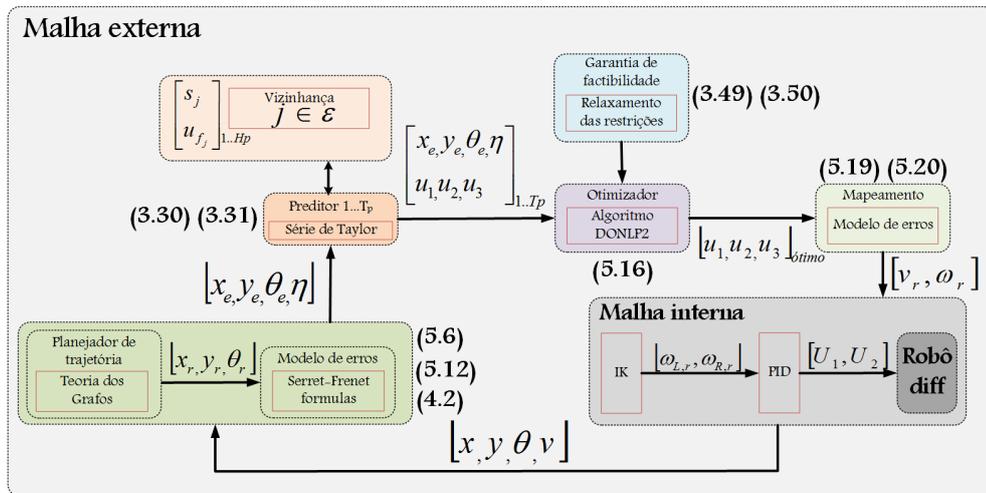


Figura 5.9: Estrutura de controle para cada robô diferencial.



Figura 5.10: Robô diferencial utilizado.

A função curvatura para este tipo de caminho é dada como segue:

$$k(s) = \frac{|\sin(s)|}{[1 + \cos^2(s)]^{\frac{3}{2}}}. \quad (5.23)$$

Esta expressão mostra que caminhos senoidais permitem análises mais criteriosas da estratégia proposta. A posição inicial para o caminho base foi $x_0 = -3,72\text{m}$ e $y_0 = 0,71\text{m}$ e os parâmetros da formação foram os seguintes: $p_{12} = -0,2\text{m}$, $p_{13} = -0,2\text{m}$, $q_1 = 0\text{m}$, $q_2 = 0,2\text{m}$, $q_3 = -0,2\text{m}$. Para gerar os altos erros iniciais requeridos, as posturas iniciais dos robôs foram: $\mathbf{x}_1 = [-5; 0; 0]^T$, $\mathbf{x}_2 = [-4,5; 0,74; 0]^T$, $\mathbf{x}_3 = [-4,5; -0,74; 0]^T$.

Sem relaxamento das restrições

Nesta etapa é feita uma análise prática sem a utilização da técnica proposta para garantia de factibilidade. Assim, será possível verificar que a abordagem para o controle de formação baseado em *NMPC* nominal não contempla situações em que os robôs estejam em pontos distantes dos caminhos de referência e sem formação definida. A Tabela 5.3 apresenta as matrizes de ponderação utilizadas.

As tolerâncias para os erros de estado foram definidas como sendo ($x_{e_i} = y_{e_i} = 5\text{cm}$) e ($\theta_{e_i} = 0,1\text{rad}$), valores compatíveis com os níveis de precisão do mecanismo de localização utilizado (sistema odométrico). Para os erros de velocidade (η_i) as tolerâncias foram definidas como sendo iguais a $0,02\text{m/s}$. Estes valores foram adicionados às restrições definidas por \mathcal{X}_i . Os valores limitantes de \mathcal{U}_i foram definidos de acordo com as características dos robôs reais utilizados. As inequações que definem

Tabela 5.3: Matrizes de ponderação.

\mathbf{Q}_i	$diagonal(1; 2,5; 0,00005; 0,5)$
\mathbf{R}_i	$diagonal(0,05; 0,005; 0,0005)$
\mathbf{W}_i	$diagonal(0,1; 0,0001)$

estas restrições para o otimizador são sumarizadas como segue:

$$|u_1| \leq 0,1; |u_2| \leq 0,1; |u_3| \leq 0,04;$$

$$|x_{e_i}| \leq 0,05 \text{ m}; |y_{e_i}| \leq 0,05 \text{ m};$$

$$|\theta_{e_i}| \leq 0,1 \text{ rad}; |\eta_i| \leq 0,02 \text{ m/s}.$$

Resultados experimentais foram adquiridos durante 215s. A Figura 5.11(a) mostra as poses instantâneas dos robôs e os caminhos seguidos neste experimento. Verificam-se alguns cruzamentos de caminhos, caracterizando possíveis colisões entre os robôs. Isto se deve à incapacidade de se obter as ações de controle necessárias para regular os elevados erros iniciais. Esta figura mostra que após um certo tempo a formação converge mesmo com um desempenho deteriorado. Ao longo deste experimento o otimizador acusa infactibilidade em muitos instantes (ver Figura 5.11(b)), o que compromete a estabilidade do esquema original baseado em *NMPC*.

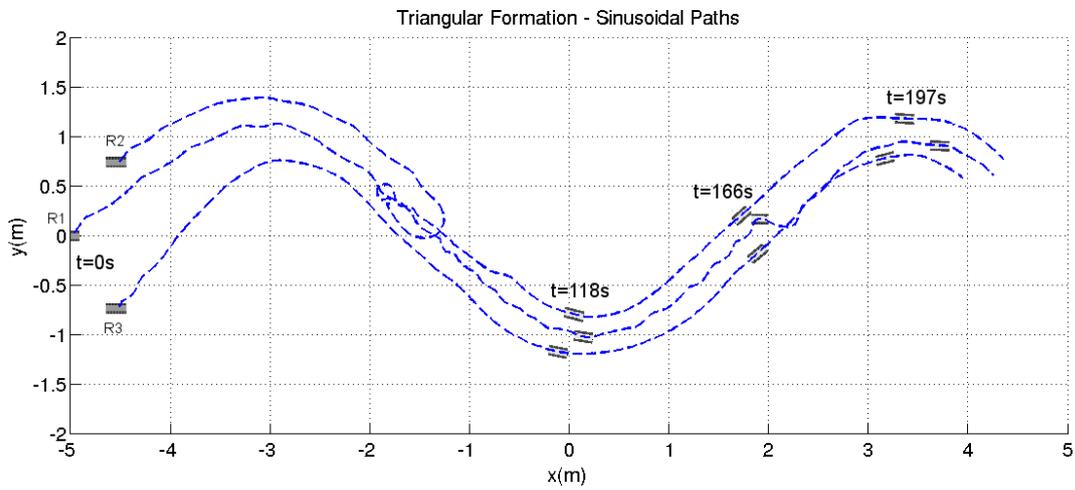
A Figura 5.11(c) ilustra a evolução temporal das posturas neste experimento. Valores elevados para os erros de formação foram obtidos, especialmente no início dos movimentos como pode ser verificado na Figura 5.11(d). Este resultado é justificado pela prioridade na regulação dos erros de seguimento de caminhos locais, desde que tais erros estão dispostos sob a forma de restrição, diferentemente dos erros de formação que possuem influência direta na função objetivo.

Os erros de seguimento de caminho são mostrados na Figura 5.11(e), sendo possível verificar uma regulação lenta e com grandes oscilações ao longo do experimento.

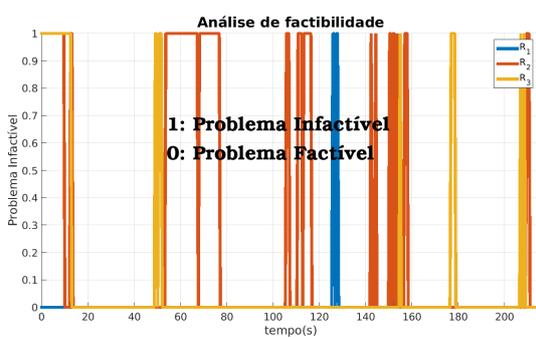
Os esforços de controle são ilustrados na Figura 5.11(f), onde se verifica que velocidades próximas do limite superior praticável pelos robôs móveis foram necessárias.

Como os valores das funções objetivos crescem continuamente devido aos erros de formação elevados, valores crescentes das ações de controle são necessários, e, como eles não são praticáveis, verifica-se instabilidade em diversos instantes do experimento.

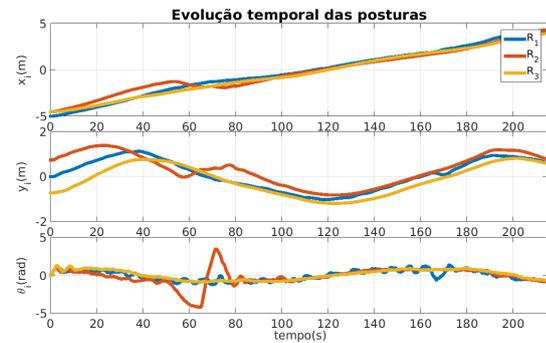
O tempo de processamento é mostrado na Figura 5.11(g). O desempenho degradado, especialmente para R_2 é devido à necessidade de se ajustar os erros de seguimento que surgem ao longo do experimento. Foram obtidos os seguintes tempos de processamento médios: $T_{p_1} = 37,3\text{ms}$; $T_{p_2} = 87,1\text{ms}$; $T_{p_3} = 12,73\text{ms}$. Tais resultados demonstram a impossibilidade de utilizar esta técnica nas situações apresentadas.



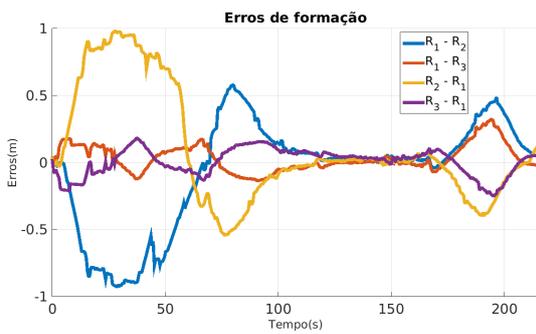
(a) Poses instantâneas dos robôs e caminhos seguidos.



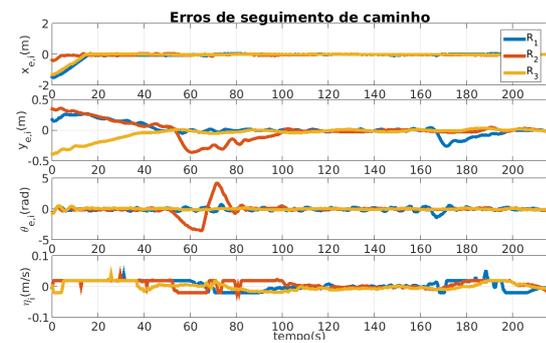
(b) Análise de infactibilidade.



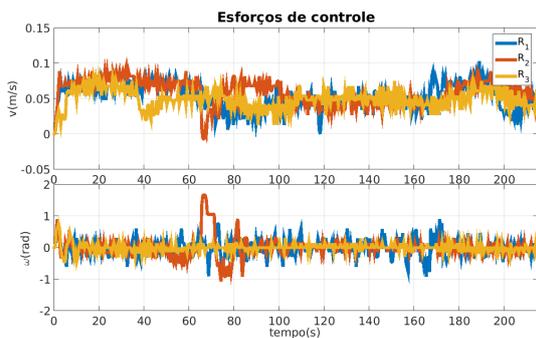
(c) Evolução temporal das posturas.



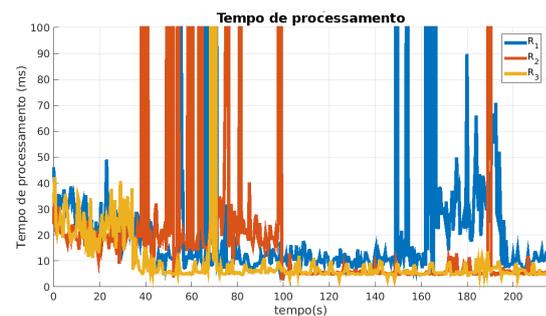
(d) Erros de formação.



(e) Erros de seguimento de caminho.



(f) Esforços de controle.



(g) Custos computacionais.

Figura 5.11: Controle de formação para robôs diferenciais - Sem relaxamento das restrições.

Com relaxamento das restrições

Com base nos resultados anteriores, implementa-se a técnica para garantia de factibilidade, para que se possa posicionar os robôs em pontos distantes dos caminhos de referência e sem formação definida.

Assim é possível definir apropriadamente a técnica proposta como controle de formação baseado em *NMPC*, desde que, fazendo o problema de otimização factível em todos os instantes de tempo, os erros de formação sempre serão bem caracterizados nas funções objetivo locais. Conforme definido anteriormente, esta proposta baseia-se numa política de relaxamento das restrições do problema, onde as restrições nas entradas são definidas com valores elevados e as restrições dos estados são definidas como função dos erros de seguimento de caminho iniciais. Deste modo, as novas restrições do problema de otimização são dadas como segue:

$$|u_1| \leq 10^3; |u_2| \leq 10^3; |u_3| \leq 10^3;$$

$$|x_{e_i}| \leq 0,05 + x_{e_i}^{max} m; |y_{e_i}| \leq 0,05 + y_{e_i}^{max} m;$$

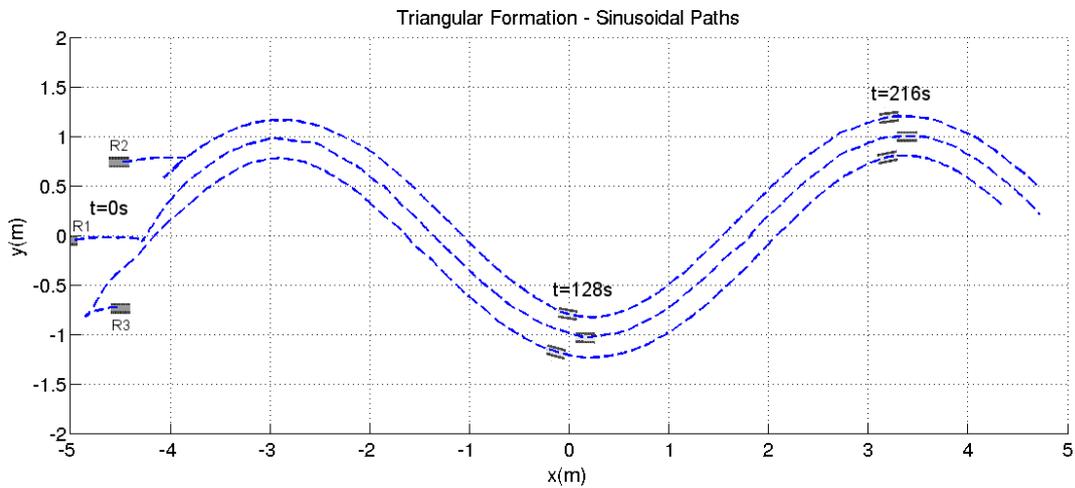
$$|\theta_{e_i}| \leq 0,1 + \theta_{e_i}^{max} rad; |\eta_i| \leq 0,02 + \eta_i^{max} m/s.$$

Utilizando os mesmos parâmetros do experimento da subseção anterior, foram adquiridos resultados por 250s. A Figura 5.12(a) mostra as poses instantâneas dos robôs e os caminhos seguidos neste caso. Observa-se que os robôs convergem suavemente para os caminhos de referência e que os erros de formação são rapidamente regulados, convergindo para zero ao longo do experimento. Para melhor análise, a Figura 5.12(b) mostra que o problema permaneceu factível e a Figura 5.12(c) ilustra a evolução temporal das posturas.

A Figura 5.12(d) mostra que os erros de formação foram regulados em cerca de 30s e a Figura 5.12(e) mostra que os erros de seguimento de caminho foram regulados e mantidos em valores admissíveis a todo tempo.

Os esforços de controle são mostrados na Figura 5.12(f). Observa-se que ações de controle positivas e negativas foram geradas objetivando manter a factibilidade do problema multi objetivo, havendo sempre redução nos erros de formação mesmo sem restrições explícitas.

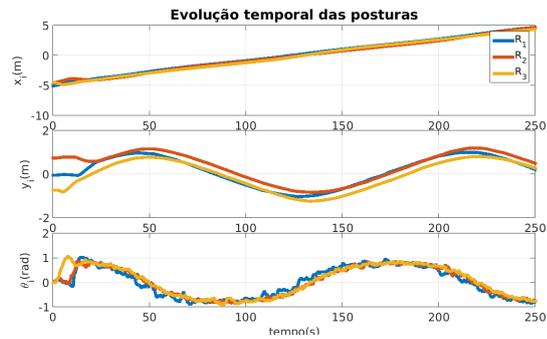
Relativamente ao tempo de processamento deste experimento, observa-se na Figura 5.12(g) a existência de apenas um pico de processamento no início dos movimentos com subsequente manutenção em valores aceitáveis. Os tempos de processamento médios neste caso foram os seguintes: $T_{p_1} = 24,5\text{ms}$; $T_{p_2} = 6,4\text{ms}$; $T_{p_3} = 6,4\text{ms}$, valores muito menores do que aqueles obtidos no experimento anterior.



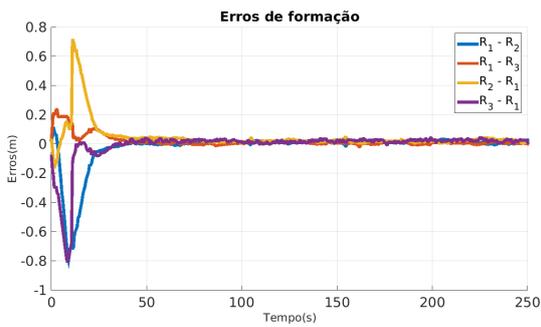
(a) Poses instantâneas dos robôs e caminhos seguidos.



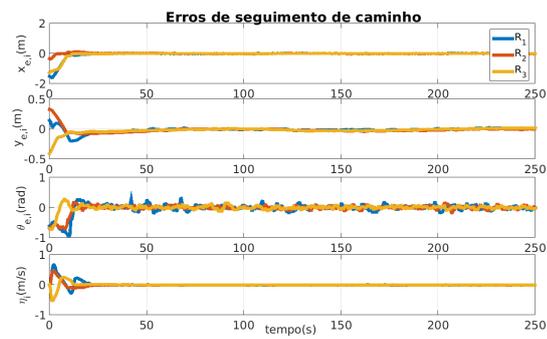
(b) Análise de infactibilidade.



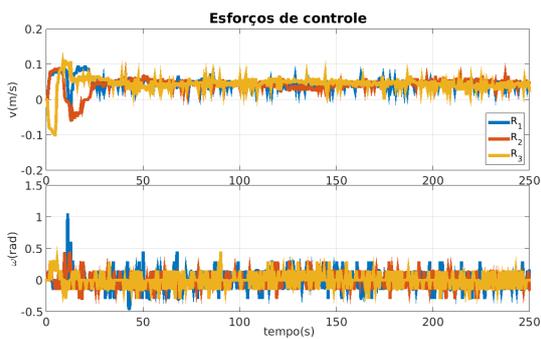
(c) Evolução temporal das posturas.



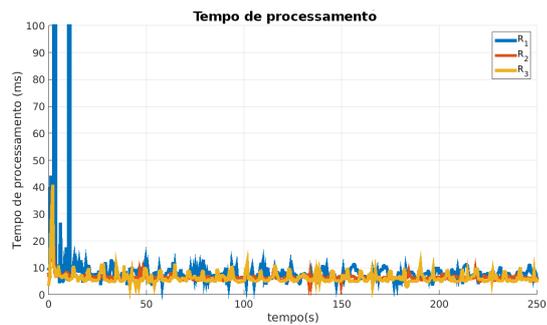
(d) Erros de formação.



(e) Erros de seguimento de caminho.



(f) Esforços de controle.



(g) Custos computacionais.

Factibilidade - Análise comparativa

As variáveis de controle internas do *NMPC*, em ambos os casos, são ilustradas na Figura 5.13.

Observa-se da Figura 5.13(a) que os limites especificados para \mathbf{u}_i foram alcançados em vários instantes. Este resultado mostra que não foi possível regular os erros de seguimento de caminho com as restrições especificadas, especialmente no início do experimento. A Figura 5.13(b) mostra o comportamento utilizando a abordagem proposta. Verifica-se que valores iniciais elevados são gerados o que possibilita a obtenção de valores mínimos factíveis.

A Figura 5.13(c) mostra um comportamento irregular das funções objetivo locais, diferentemente da Figura 5.13(d) que apresenta completa convergência, confirmando os benefícios da estratégia proposta.

Estes resultados foram reunidos e publicados em (Ribeiro *et al.* 2015b).

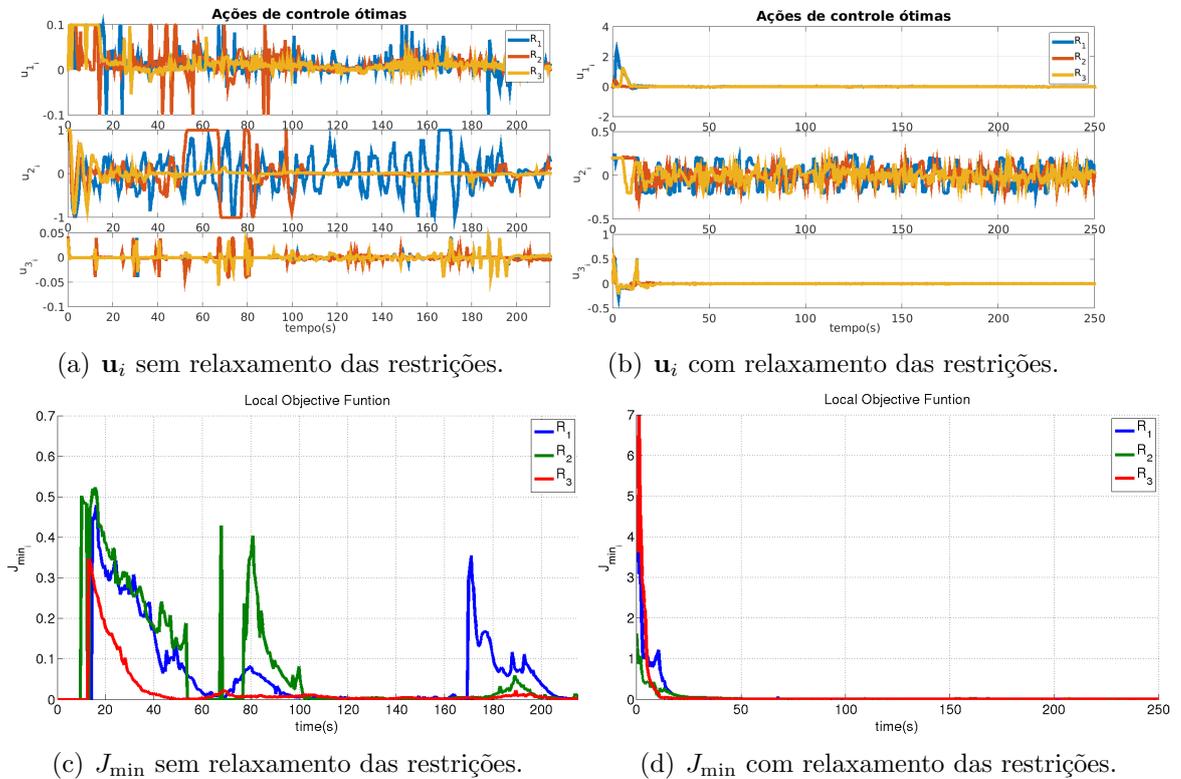


Figura 5.13: Controle de formação para robôs diferenciais - Entradas ótimas.

5.4 Controle de Formação para Quadricópteros

Neste caso, existe um conjunto de N quadricópteros e N caminhos de referência Γ_i , com $i = 1, \dots, N$, e o quadricóptero Q_i deve seguir o caminho Γ_i . O objetivo é coordenar os movimentos deste grupo de quadricópteros e isto pode ser alcançado através de ajustes nas velocidades de navegação u_{f_i} e nas taxas de progressão dos veículos virtuais \dot{s}_i .

Assim como no caso dos robôs anteriores, um caminho de referência base Γ_r é previamente definido e o mesmo deve ser seguido pelo grupo de quadricópteros. A Figura 5.14 ilustra este cenário para três quadricópteros numa formação triangular.

Considerando os vetores de postura do caminho de referência no ponto s_i como sendo $[x_{i,b}, y_{i,b}, z_{i,b}, \psi_{i,b}]^T$, e as distâncias transversais q_i entre este ponto e o centro de massa dos quadricópteros, os vetores de estado de referência são:

$$\mathbf{x}_{i,r} = \begin{bmatrix} x_{i,r} \\ y_{i,r} \\ z_{i,r} \\ \psi_{i,r} \end{bmatrix} = \begin{bmatrix} x_{i,b} - q_i \sin \psi_{i,b} \\ y_{i,b} + q_i \cos \psi_{i,b} \\ z_{i,b} \\ \psi_{i,b} \end{bmatrix}, \quad (5.24)$$

sendo $\psi_{i,b}$ o ângulo da tangente à curva de referência no ponto s_i , relativo a $\{W\}$.

A velocidade de navegação de referência para o quadricóptero Q_i é determinada

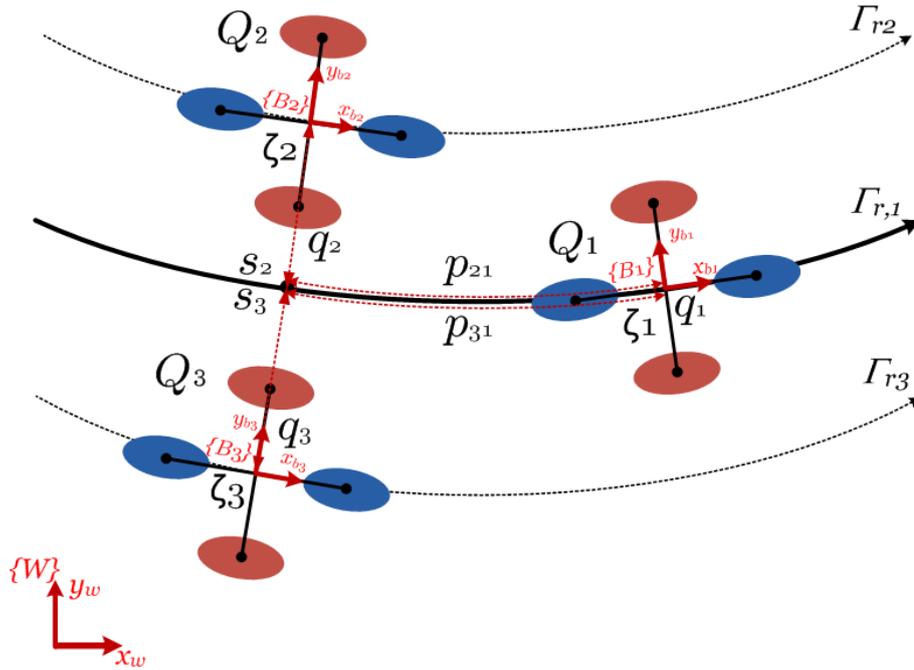


Figura 5.14: Representação do seguimento de caminho coordenado.

através da relação entre o comprimento total do caminho de referência base $P(\Gamma_r)$ e o comprimento total do caminho de referência $P(\Gamma_i)$ como segue:

$$u_{f_{i,r}} = \frac{P(\Gamma_i)}{P(\Gamma_r)} u_{f_{B,r}}, \quad (5.25)$$

sendo $u_{f_{B,r}}$ a velocidade de navegação base previamente definida para a formação.

A formação é controlada quando as velocidades relativas ao caminho base são as mesmas. Considerando o exemplo da Figura 5.14, o quadricóptero externo deve ter uma velocidade de navegação maior do que a do quadricóptero interno. A velocidade angular de referência é determinada para cada quadricóptero i como sendo:

$$\omega_{i,r} = k_i(\zeta_i) u_{f_{r,i}}, \quad (5.26)$$

sendo ζ_i o comprimento do caminho de referência para o i -ésimo quadricóptero e $k_i(\zeta_i)$ é a curvatura deste caminho dada, como segue:

$$\zeta_i = \frac{P(\Gamma_r)}{P(\Gamma_i)} s_i, \quad (5.27)$$

$$k_i(\zeta_i) = \frac{k_r(s_i)}{1 - q_i k_r(s_i)}, \quad (5.28)$$

sendo $k_r(s_i)$ a curvatura do caminho de referência base.

A velocidade linear de referência para o controle de altitude é definida para cada quadricóptero como segue:

$$v_{z_{r_i}} = K_z (z_{r_i} - z_i), \quad (5.29)$$

sendo K_z constante que depende da máxima velocidade vertical permissível em determinada plataforma experimental.

Assim como nas modelagens anteriores, já que a velocidade de navegação para cada quadricóptero deve ser coordenada com os elementos da vizinhança, u_{f_i} não pode ser constante. Assim, uma nova variável de estado η_i é definida, dada por $\eta_i = u_{f_i} - u_{f_{i,r}}$, sendo $u_{f_{i,r}}$ calculada por (5.25).

Assim, tem-se que:

$$\dot{\eta}_i = \epsilon_i - \gamma_i, \quad (5.30)$$

sendo ϵ_i e γ_i as acelerações, atuais e de referência, respectivamente.

Finalmente, os sinais de controle para cada quadricóptero são definidos da seguinte forma:

$$\mathbf{u}_{i,e} = \begin{bmatrix} u_{i,1} \\ u_{i,2} \\ u_{i,3} \\ u_{i,4} \\ u_{i,5} \end{bmatrix} = \begin{bmatrix} -\dot{\zeta}_i + (\eta_i + u_{f_{i,r}}) \cos \varphi_{i,e} \\ v_{z_i} - v_{z_{i,r}} \\ \dot{\varphi}_{i,e} - k_i(\zeta_i) \dot{\zeta}_i \\ \omega_{\psi_i} - \omega_{\psi_{i,r}} \\ \epsilon_i - \gamma_i \end{bmatrix}. \quad (5.31)$$

Assim, o modelo final para os erros de estado é dado como segue:

$$\dot{\mathbf{x}}_{i,e} = \begin{bmatrix} \dot{x}_{i,e} \\ \dot{y}_{i,e} \\ \dot{z}_{i,e} \\ \dot{\varphi}_{i,e} \\ \dot{\psi}_{i,e} \\ \dot{\eta}_i \end{bmatrix} = \begin{bmatrix} y_{i,e}k_i(\zeta_i)\dot{\zeta}_i + u_{i,1} \\ -x_{i,e}k_i(\zeta_i)\dot{\zeta}_i + (\eta_i + u_{f,i,r}) \text{sen } \varphi_{i,e} \\ u_{i,2} \\ u_{i,3} \\ u_{i,4} \\ u_{i,5} \end{bmatrix}. \quad (5.32)$$

Considerando que as saídas são os próprios estados, o problema de controlar os movimentos do i -ésimo quadricóptero pode ser resumido da seguinte forma:

Encontrar $\dot{\zeta}$, v_{z_i} , $\omega_{\psi_{i,r}}$ e ϵ_i , tal que $u_{i,1}$, $u_{i,2}$, $u_{i,3}$, $u_{i,4}$, $u_{i,5}$, $x_{i,e}$, $y_{i,e}$, $z_{i,e}$, $\psi_{i,e}$ e η_i sejam factíveis.

5.4.1 Projeto dos Controladores

Considerando a mesma estrutura da função objetivo utilizada para os robôs anteriores e que existem controladores PD embarcados nos quadricópteros o esquema de controle está ilustrado na Figura 5.15

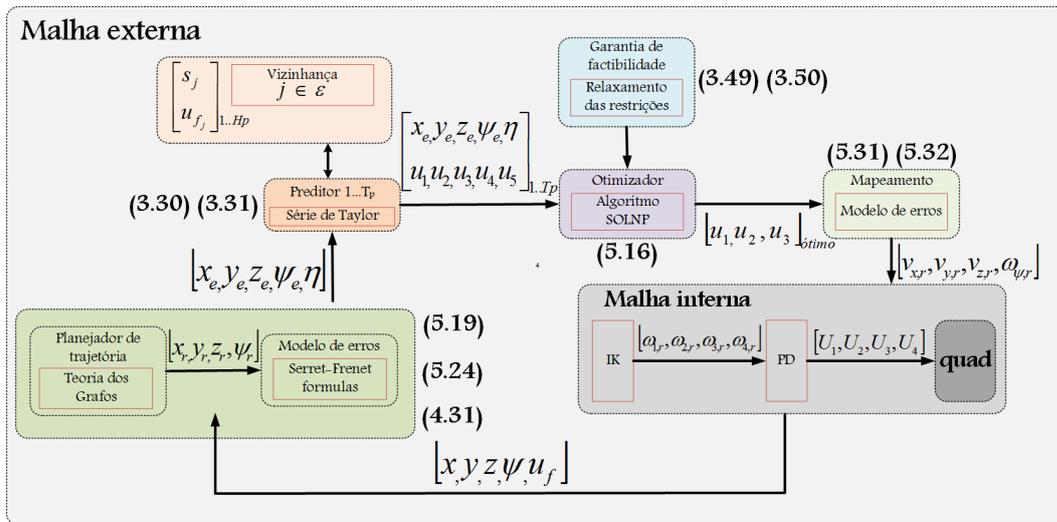


Figura 5.15: Estrutura de controle para cada quadricóptero.

5.4.2 Resultado Simulado

A estratégia de controle proposta foi avaliada através de simulações que evidenciam o desempenho em situações em que existem erros elevados, sendo neste caso considerados também erros de formação. O caminho base é definido como segue:

$$\begin{aligned}x(s_i) &= 1,5 \cos(s_i); \\y(s_i) &= 1,5 \sin(s_i); \\z &= 5.\end{aligned}\tag{5.33}$$

O ângulo de guinada de referência é o ângulo da tangente a cada caminho individual Γ_i .

Foram utilizados os seguintes parâmetros para formação desejada: $p_{12} = -0,5\text{m}$, $p_{13} = 0,5\text{m}$, $q_1 = 0\text{m}$, $q_2 = 1\text{m}$, $q_3 = -1\text{m}$. Os horizontes foram mantidos os mesmos e as matrizes de ponderação são apresentadas na Tabela 5.4.

Considerando ainda as mesmas especificações apresentadas no Capítulo 4 para as restrições e para as rajadas de vento aleatórias, foram adquiridos resultados por 150s, com $u_{fB,r} = 0,5\text{m/s}$ e um mesmo período fundamental de amostragem.

Para gerar altos erros iniciais, os quadricópteros foram posicionados distantes dos caminhos de referência e sem formação definida. As posições iniciais neste caso são: $\xi_1 = [1,5,0,0]$, $\xi_2 = [0,5,0,0]$ e $\xi_3 = [2,5,0,0]$. A orientação inicial para todos os i -quadricópteros é: $\Phi_i = [\frac{\pi}{2}, 0, 0]$.

A simulação consiste em 3 eventos principais objetivando analisar o comportamento da estratégia proposta em casos de caminhos e formações variantes no tempo:

1. $T_1 = 25\text{s} \rightarrow z_{1,r} = z_{2,r} = 7\text{m}$: As altitudes de referência para Q_2 e Q_3 são modificadas para 7m.
2. $T_2 = 50\text{s} \rightarrow q_3 = 0\text{m}$: O caminho de referência de Q_3 é modificado, o que representa um distúrbio de -1m de amplitude em $y_{3,e}$.
3. $T_3 = 75\text{s} \rightarrow q_2 = 0\text{m}$, $p_{12} = 1\text{m}$, $p_{13} = -1\text{m}$: O caminho de referência Q_2 é também modificado e o espaçamento entre os quadricópteros modificado para 1m, o que representa um distúrbio de 1m de amplitude em $y_{2,e}$.

É possível verificar, que tais as mudanças nos parâmetros da formação adiciona perturbações diretamente nas funções objetivos locais.

Tabela 5.4: Matrizes de ponderação.

\mathbf{Q}_i	$diagonal(1000;3000;20;3000;1000;400)$
\mathbf{R}_i	$diagonal(3;250;3;25;5)$
\mathbf{W}_i	$diagonal(200; 0,1)$

A Figura 5.16(a) ilustra os resultados obtidos no espaço tridimensional e a Figura 5.16(b) apresenta a evolução temporal das posturas.

Verifica-se o atendimento dos objetivos de controle, desde que houve regulação ao longo dos caminhos de referência em cada passo mesmo na presença dos distúrbios causados por mudanças nos parâmetros da formação e turbulência causada pelas rajadas de vento.

A Figura 5.16(c) apresenta os erros de seguimento de caminho, e, verifica-se que durante as regulações, outros estados do mesmo quadricóptero são afetados, comportamento justificado pela sintonia escolhida e a necessidade de regular os erros de formação. Este resultado mostra também que os estados de outros quadricópteros não foram afetados durante a regulação dos distúrbios em T_1 e T_2 . Em T_3 os estados de Q_1 são afetados para regular o erro de formação aplicado. Em todos os casos é verificada a convergência dos estados para as tolerâncias especificadas.

O máximo erro de formação em regime permanente entre as configurações intermediárias foi de 0,4m com convergência para 0m ao final do experimento, como pode ser visto na Figura 5.16(d), mostrando a eficácia do acoplamento via função objetivo. O desempenho durante o transitório prova a robustez da técnica, desde que distúrbios abruptos de 2m nos erros de formação foram rejeitados.

Os esforços de controle da malha externa são mostrados na Figura 5.16(e). Verifica-se que quando alcançado o limite de saturação para a variável perturbada, as outras permanecem em valores factíveis, mantendo a estabilidade e demonstrando a eficiência da técnica. A Figura 5.16(f) mostra que os esforços de controle na malha interna foram mantidos em valores compatíveis com as especificações dos quadricópteros utilizados, mesmo durante a regulação dos distúrbios e na presença de turbulência.

Para ilustrar o comportamento da função objetivo, a Figura 5.16(g) mostra uma comparação com um caso em que não se utiliza a técnica proposta. Observa-se que neste caso a função objetivo não converge e o algoritmo de otimização acusa um problema de infactibilidade. A Figura 5.16(h) ilustra o comportamento da função objetivo ao longo de toda a simulação utilizando a técnica proposta, sendo mostrada a completa convergência mesmo após os distúrbios.

Estas análises demonstram as vantagens da aplicação da técnica proposta para garantia de factibilidade, também para o caso de robôs aéreos não tripulados. Obviamente, numa projeção para validação através de experimentos, alguns ajustes devem ser feitos, especialmente nos parâmetros de sintonia.

Adicionalmente, é possível concluir que proposta aumenta a viabilidade das aplicações reais, já que, conforme validado para para outros robôs, o tempo de processamento desta abordagem é bastante reduzido, o que permite implementar técnicas seguras e eficientes de pouso e decolagem.

Estes resultados foram reunidos e publicados em (Ribeiro *et al.* 2014) e (Ribeiro *et al.* 2015a).

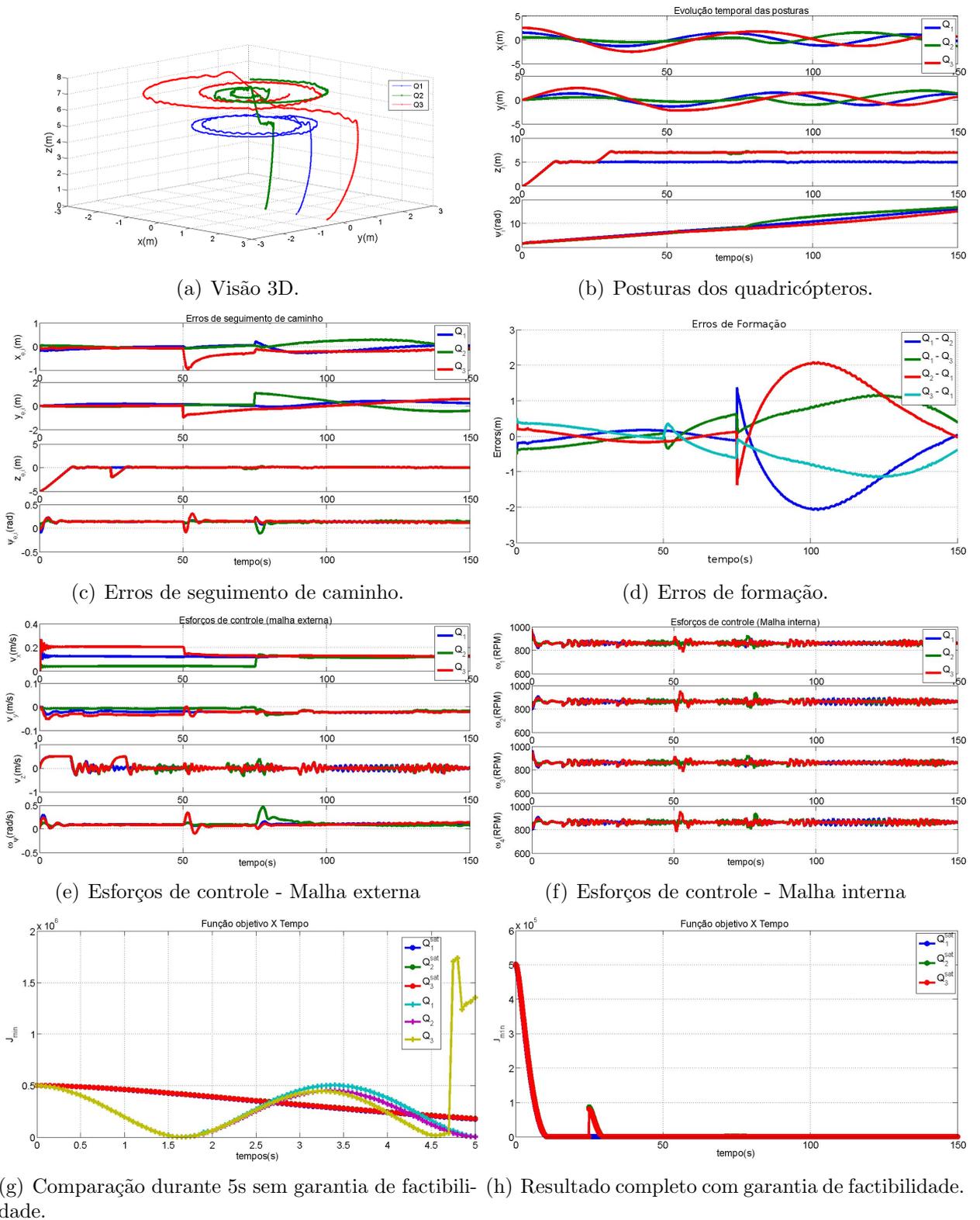


Figura 5.16: Controle de formação para quadricópteros: Caminhos circulares - Formação triangular variante no tempo.

5.5 Controle de Formação Utilizando Informações Visuais

Todas as abordagens para controle de formação apresentadas anteriormente possuem como premissa a existência de um caminho de referência que gera, a partir dos parâmetros da formação, caminhos individuais para todos os robôs. Para o caso de caminhos visuais, o caminho de referência é arbitrário e adquirido em tempo de execução, inviabilizando parametrização prévia dos caminhos de referência em termos de comprimento de caminho.

Uma solução trivial para a manutenção das propostas anteriores seria a instalação de câmeras em todos os robôs. Neste caso, os caminhos individuais são também adquiridos através de informações visuais e o seguimento de caminhos visuais coordenados seria obtido através do acoplamento das funções objetivos locais. Esta solução é inviável do ponto de vista prático já que haverá a necessidade de pré especificação de caminhos locais baseados nos parâmetros da formação, conflitando com a ideia de seguimento de caminhos arbitrários.

Outra solução, seria baseada na utilização de apenas uma câmera para um dos robôs da formação, denominado líder, que segue um caminho de referência visual e envia sua postura instantânea para os outros robôs estimarem seus próprios caminhos de referência. Porém, os resultados obtidos no capítulo anterior demonstram que tal solução também é inviável, pois, exigiria uma calibração perfeita dos sistemas odométrico e visual.

Alternativamente, ao invés de serem estimados caminhos contínuos que dependem de grandezas bastante sensíveis a erros no sistema visual, tais como curvatura e comprimento de caminho, seria possível estimar trajetórias de referência (Kanjawanishkul e Zell 2008c), porém, a implementação seria inviável já que pode não ser possível encontrar uma única função temporal representativa de todos os caminhos seguidos pelo líder, interferindo fundamentalmente no cálculo das velocidades de referência, pois sabe-se que as mesmas dependem da existência de derivadas contínuas em todos os pontos da trajetória.

Finalmente, ainda considerando a postura do líder conhecida a todos os instantes de amostragem, seria possível implementar a técnica de estabilização em pontos de trajetória para os seguidores, porém, a ausência de uma lei de controle contínua, potencialmente contribui para a não estabilização da formação. Além disso, os erros de regime permanente quando se trabalha em coordenadas cartesianas (Kuhne *et al.* 2005) exigiria a utilização de modelos em coordenadas polares, gerando implicações práticas devido ao elevado nível de ruído presente na medição dos ângulos.

Diante deste cenário, propõe-se duas abordagens para o controle de formação utilizando informação visual. A primeira baseia-se na utilização de uma câmera real para o robô líder e câmeras virtuais em cada seguidor que, desde que a formação esteja controlada, receberá a mesma imagem do líder. A segunda proposta é mais imediata e baseia-se na estratégia Líder-Seguidor básica, onde apenas o robô líder possui uma câmera. Em ambos os casos obtém-se modelos de erros de estado e aplica-se

controladores *NMPC*. Resultados experimentais evidenciam melhor desempenho da segunda proposta, sendo para esta, aplicadas as técnicas para garantia de estabilidade e factibilidade.

Nesta etapa, são consideradas formações triangulares com três robôs diferenciais, porém os resultados podem ser estendidos para n robôs de diferentes tipos e formações geométricas diversas.

5.5.1 Estratégia baseada em câmeras virtuais

Neste caso, a ideia de geração de caminhos virtuais a partir do caminho real base seguido por um robô líder é mantida. Para este fim, pressupõe-se que o caminho gerado é idêntico ao original transladado no plano e uma câmera virtual é instalada nos robôs seguidores. Admite-se ainda que, enquanto a formação estiver controlada, ou seja, distância transversal e longitudinal ao longo do caminho de referência em valores próximos daqueles especificados pelos parâmetros da formação, a imagem capturada pelas câmeras dos seguidores são idênticas às imagens capturadas pelo líder. A Figura 5.17 ilustra esta ideia para um robô líder i e um robô seguidor j .

A pose de referência para o robô j é obtido da seguinte forma:

$$\begin{bmatrix} x_{r_j} \\ y_{r_j} \\ \theta_{r_j} \end{bmatrix} = \begin{bmatrix} x_{B_j} - q_j \sin \theta_{B_j} \\ y_{B_j} + q_j \cos \theta_{B_j} \\ \theta_{B_j} \end{bmatrix}, \quad (5.34)$$

sendo:

$$\begin{bmatrix} x_{B_j} \\ y_{B_j} \\ \theta_{B_j} \end{bmatrix} = \begin{bmatrix} x_i + p_{ji} \cos \theta_i \\ y_i + p_{ji} \sin \theta_i \\ \theta_i \end{bmatrix}, \quad (5.35)$$

Da análise dos vetores da Figura 5.17 obtém-se as seguintes relações:

$$\theta_{ej} = \theta_{ei} + \theta'_i; \quad (5.36)$$

$$z_j = (z_i + z'_i) \cos \theta'_i. \quad (5.37)$$

sendo z'_i e θ'_i dados por:

$$z'_i = \frac{1}{\cos^2 \theta'_i} (z_i \sin^2 \theta'_i + H_j \sin \theta'_i + y_{ej}); \quad (5.38)$$

$$\theta'_i = \theta_j - \theta_i. \quad (5.39)$$

Substituindo (5.38) em (5.37) obtém-se:

$$z_j = \frac{z_i + H_j \sin \theta'_i + y_{ej}}{\cos \theta'_i}. \quad (5.40)$$

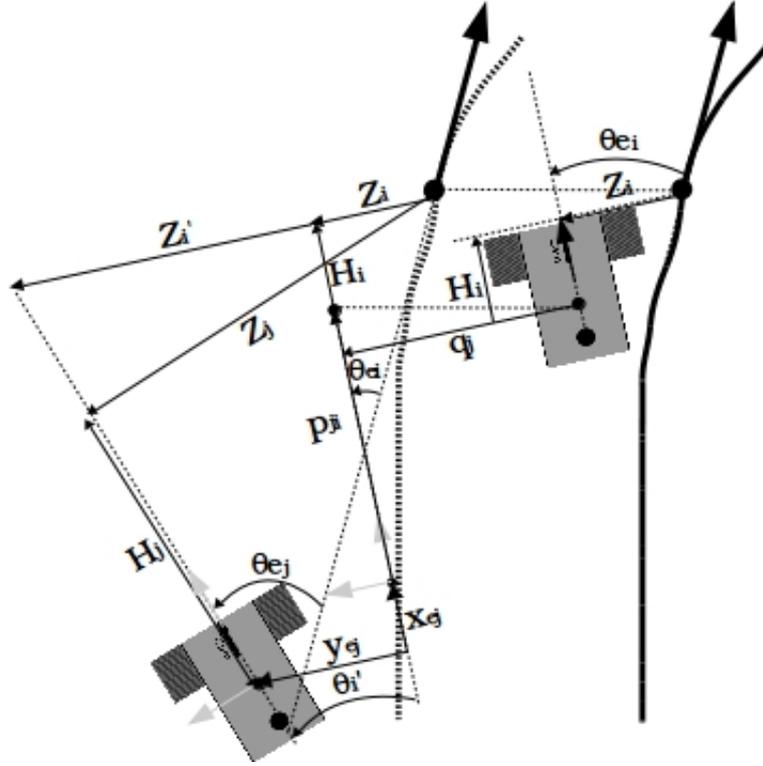


Figura 5.17: Controle de formação utilizando informações visuais - Estratégia baseada em câmeras virtuais.

com:

$$\begin{bmatrix} x_{ej} \\ y_{ej} \end{bmatrix} = \begin{bmatrix} \cos \theta'_i & \text{sen } \theta'_i \\ -\text{sen } \theta'_i & \cos \theta'_i \end{bmatrix} \begin{bmatrix} x_j - x_{rj} \\ y_j - y_{rj} \end{bmatrix}; \quad (5.41)$$

$$H_j = (H_i + p_{ji} + x_{ej}) \cos \theta'_i - (z_i + y_{ej}) \text{sen } \theta'_i. \quad (5.42)$$

Completa-se assim a representação das variáveis de estado para os seguidores em termos daquelas calculadas através de medidas da câmera real instalada no líder.

De (5.36), (5.40) e (5.42), observa-se que:

$$\lim_{\theta'_i, y_{ej} \rightarrow 0} z_j = z_i; \quad (5.43)$$

$$\lim_{\theta'_i \rightarrow 0} \theta_{ej} = \theta_{ei}; \quad (5.44)$$

$$\lim_{\theta'_i, x_{ej}, y_{ej} \rightarrow 0} H_j = H_i + p_{ji}. \quad (5.45)$$

O que permite concluir que a imagem capturada pela câmera virtual será a mesma da real sempre que a formação estiver controlada.

Os modelos dos erros de estado para os seguidores são obtidos de maneira similar ao obtido para o robô líder, porém, atentando para o fato de que H_j não é constante. Além disso, como as velocidades devem ser coordenadas, os erros de velocidade são definidos como um estado adicional. Tais modelos são apresentados a seguir:

$$\dot{z}_i = \omega_i H_i + (\omega_i z_i + v_i) \tan(\theta_{ri}); \quad (5.46)$$

$$\dot{\theta}_{ri} = u_{1i} = \omega_i - c_i(s_i) \frac{(\omega_i z_i + v_i)}{\cos \theta_{ri}}; \quad (5.47)$$

$$\dot{\eta}_i = u_{2i} = \dot{v}_i - \dot{v}_{ri}. \quad (5.48)$$

$$\dot{z}_j = \omega_j H_j + (\omega_j z_j + v_j + \dot{H}_j) \tan(\theta_{rj}); \quad (5.49)$$

$$\dot{\theta}_{rj} = u_{1j} = \omega_j - c_j(s_j) \frac{(\omega_j z_j + v_j + \dot{H}_j)}{\cos \theta_{rj}}; \quad (5.50)$$

$$\dot{\eta}_j = u_{2j} = \dot{v}_j - \dot{v}_{rj}. \quad (5.51)$$

com $c_i(s_i)$ satisfazendo:

$$\dot{c}_i = \frac{dc_i}{ds_i} \frac{(\omega_i z_i + v_i)}{\cos \theta_{ri}}, \quad (5.52)$$

e com $c_j(s_j)$ satisfazendo:

$$\dot{c}_j = \frac{dc_j}{ds_j} \frac{(\omega_j z_j + v_j + \dot{H}_j)}{\cos \theta_{rj}}. \quad (5.53)$$

Sendo \dot{H}_j dado por:

$$\dot{H}_j = (\dot{x}_{ej} - \dot{\theta}'_i(z_i + y_{ej})) \cos(\theta'_i) - (\dot{z}_i + \dot{y}_{ei} + \dot{\theta}'_i(H_i + p_{ji} + x_{je})) \sin(\theta'_i), \quad (5.54)$$

e \dot{x}_{ej} , \dot{y}_{ej} e $\dot{\theta}'_i$ dados por:

$$\begin{aligned} \dot{x}_{ej} &= y_{ej} \omega_i - v_i + v_j \cos \theta'_i; \\ \dot{y}_{ej} &= -x_{ej} \omega_i + v_j \sin \theta'_i; \\ \dot{\theta}'_i &= \omega_j - \omega_i. \end{aligned} \quad (5.55)$$

Considerando novamente que as saídas são os próprios estados, o problema de controlar os movimentos de cada um dos k robôs pode ser resumido da seguinte forma:

Encontrar ω_k e v_k , tal que u_{1k} , u_{2k} , x_k , θ_{rk} e η_k sejam factíveis.

Da análise de (5.53), observa-se que \dot{c}_j deve satisfazer a um critério que depende de \dot{H}_j , \dot{x}_{ej} , \dot{y}_{ej} e $\dot{\theta}'_i$ evidenciando que, para a regulação de θ_{rj} , critérios físicos mais restritivos devem ser levados em consideração durante o processo de sintonia já que tal estado depende diretamente da curvatura.

Para os seguidores, o controle de formação baseado no acoplamento via função objetivo deve ser adaptado para que se contemple a necessidade de convergência das variáveis $\theta'_i, x_{ej}, y_{ej}$, sem comprometimento do custo computacional, o que certamente aconteceria caso fossem definidos novos estados. De (5.36), verifica-se que a convergência de θ'_i para a origem é dependente de variáveis de estado cujos limites físicos são pré especificados, sendo necessário apenas definir uma métrica para as outras variáveis. Tal métrica é dada como segue:

$$d_{j,e} = \|x_{ej} - y_{ej}\|_2. \quad (5.56)$$

O termo de acoplamento adaptado assume a seguinte forma:

$$\mathbf{f}_{j,e} = \begin{bmatrix} d_{j,e} \\ u_{f_{j,e}} \end{bmatrix}, \quad (5.57)$$

com $u_{f_{j,e}}$, neste caso, dado por:

$$u_{f_{ji,e}} = \frac{1}{1 - c_j q_{ji}} v_j - \frac{1}{1 - c_i q_{ij}} v_i. \quad (5.58)$$

A estrutura da função objetivo é a mesma dos controladores de formação das seções anteriores e a estrutura de controle para os robôs seguidores é apresentada na Figura 5.18.

Resultados Experimentais

Nesta etapa, são realizados experimentos utilizando o robô diferencial *Lego EV3* como líder real (R_1 , Figura 5.19), e robôs diferenciais simulados como seguidores (R_2 e R_3). Primeiramente, para simplificar a implementação, utiliza-se uma abordagem sequencial, onde um mesmo *PC* é responsável pelo cálculo de todas as ações de controle, sendo deste modo eliminados problemas de comunicação que poderiam surgir com a rede multi robô.

O caminho de referência escolhido para o robô líder está ilustrado na Figura 5.20 e os resultados são adquiridos para $v_{navB} = 0,05\text{m/s}$, período fundamental de amostragem de $T_s = 0,2\text{s}$, os mesmos parâmetros da formação definidos na subseção 5.3 e mesma matriz de adjacência dada por (5.17). Os parâmetros de sintonia são os mesmos para todos os robôs e estão dispostos na Tabela 5.5.

Os limites para as restrições foram definidos como segue:

$$|u_1| \leq 5; |u_2| \leq 0,05; |d_e| \leq 0,05 \text{ m}; |\theta_e| \leq 0,5 \text{ rad}; ; |\eta| \leq 0,3 \text{ m/s}.$$

Definindo-se $T_p = T_c = 3$ amostras, foram adquiridos os resultados ilustrados na Figura 5.21.

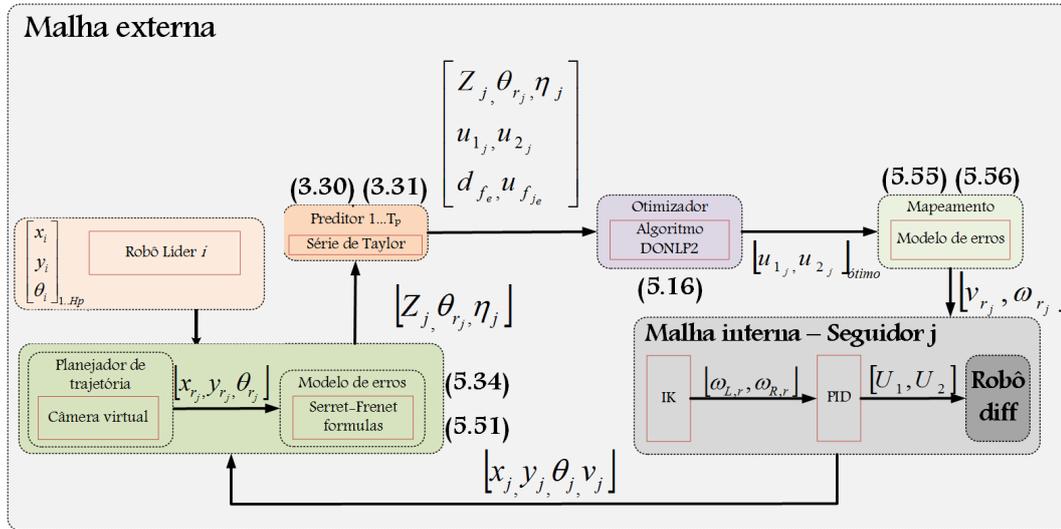


Figura 5.18: Estrutura de controle para cada robô seguidor - Estratégia baseada em câmeras virtuais.



Figura 5.19: Robô diferencial utilizado.

Tabela 5.5: Matrizes de ponderação.

\mathbf{Q}_i	$diagonal(50;0,005;25)$
\mathbf{R}_i	$diagonal(0,1;;0,1)$
\mathbf{W}_i	$diagonal(1;0,001)$

É possível verificar através da Figura 5.21(a), que o caminho foi seguido satisfatoriamente por todos os robôs. A Figura 5.21(b) ilustra as soluções dos problemas de otimização locais. Verifica-se que, ao longo do trecho de maior curvatura, há um aumento das funções objetivos e ações de controle internas que satisfazem as restrições especificadas mas apresentam oscilações.



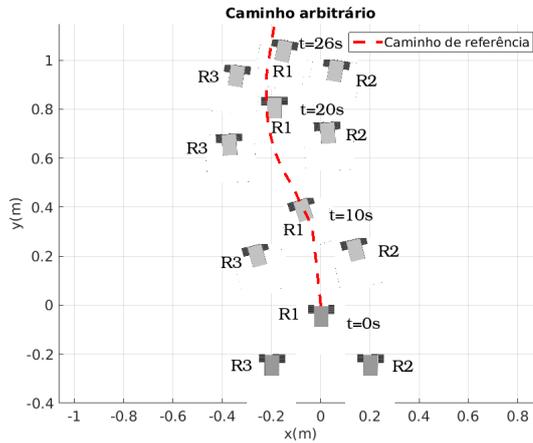
Figura 5.20: Caminho de referência para o robô líder.

Os erros de seguimento de caminho se mantiveram inferiores aos limites das restrições, conforme ilustrado na Figura 5.21(c), porém, é possível verificar que os erros angulares para os seguidores aumentam durante os trechos de maior curvatura, resultado esperado, já que, a inclusão de \dot{H}_j em \dot{c}_j dificulta uma melhor regulação deste erro. Verifica-se também que, a elevação do termo de acoplamento na função objetivo do líder, faz seu controlador apresentar um comportamento oscilatório, sendo necessário explicitar os requisitos de estabilidade para um melhor ajuste dos parâmetros de sintonia, especialmente maior peso relativo para as entradas.

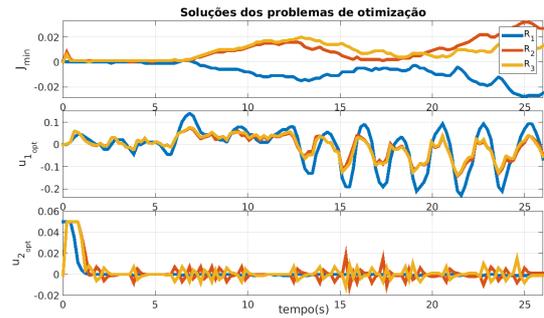
O erros de formação estão ilustrados na Figura 5.21(d), sendo possível verificar a dificuldade em regular os erros de formação durante os trechos de maior curvatura.

Os esforços de controle se mantiveram dentro de limites praticáveis (5.21(e)) e os tempos de processamento permaneceram muito inferiores ao período de amostragem (5.21(f)), mesmo implementando todos os controladores em um único nó de processamento.

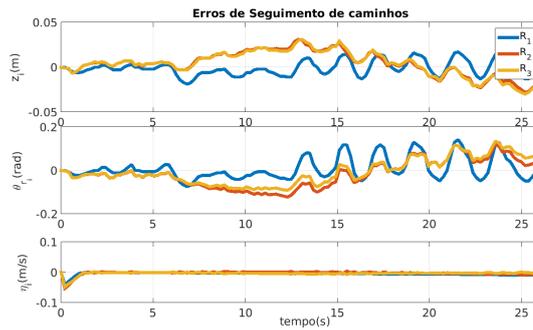
Tais resultados foram obtidos após recalibração do sistema visual, dando maior atenção ao cálculo da curvatura capturada pelo robô líder, para que fosse possível fornecer uma boa estimativa desta grandeza para os seguidores. A grande sensibilidade dos estados com relação a esta grandeza estimula a busca por outras alternativas para o controle de formação utilizando informações visuais.



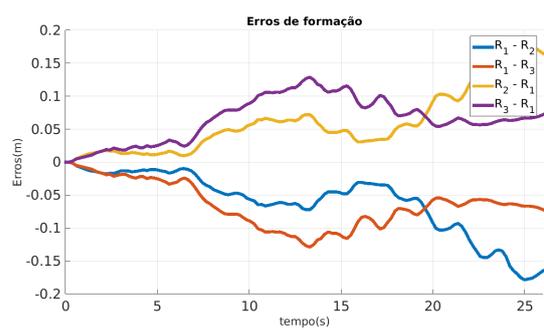
(a) Posições instantâneas dos robôs.



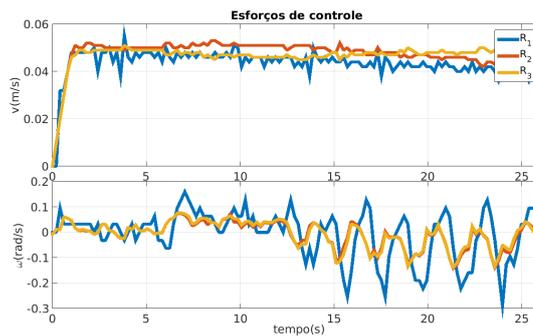
(b) Soluções do problema de otimização.



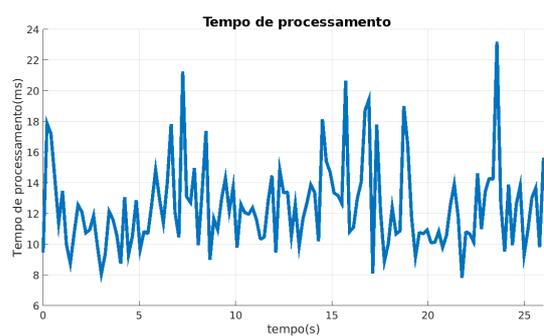
(c) Erros de seguimento de caminho visual.



(d) Erros de formação.



(e) Esforços de controle.



(f) Tempo de processamento.

Figura 5.21: Controle de formação utilizando informações visuais - Utilização de câmeras virtuais (continuação).

5.5.2 Estratégia Líder-Seguidor

O desempenho da abordagem proposta anteriormente em trechos de elevada curvatura motiva a busca de soluções alternativas. Uma das estratégias mais utilizadas para controle de formação é denominada por Líder-Seguidor, também conhecida como $SB_{ij}C$ (do inglês, Separation and Bearing Control). Isso se deve, em parte, à simplicidade de sua implementação. Tal abordagem é plenamente aplicável quando se considera que apenas um robô possuirá câmera para a implementação do seguimento de caminho visual e todos os n seguidores devem seguir este líder com distância e orientação pré especificadas através dos parâmetros da formação.

A Figura 5.22 ilustra o robô líder R_i e o robô seguidor R_j utilizados na modelagem da estratégia Líder-Seguidor. Novamente, os modelos obtidos são facilmente generalizados para um grupo de n robôs e admite-se que, a postura do robô líder é conhecida a todos os instantes de tempo através dos sistema odométrico.

Por análise vetorial, obtém-se a distância atual $l_{ij} \in \mathbb{R}_{>0}$ entre R_i e R_j da seguinte forma:

$$l_{ij} = \sqrt{(x_i - x_j - d_j \cos \theta_j)^2 + (y_i - y_j - d_j \sin \theta_j)^2} \quad (5.59)$$

sendo d_j a distância entre o centro de gravidade e a frente dos robôs diferenciais.

O ângulo de orientação de R_i relativo a l_{ij} é dado por:

$$\psi_{ij} = \pi + \phi_{ij} - \theta_i \quad (5.60)$$

sendo ϕ_{ij} calculado da seguinte forma:

$$\phi_{ij} = \tan^{-1} \left(\frac{y_i - y_j - d_j \sin \theta_j}{x_i - x_j - d_j \cos \theta_j} \right) \quad (5.61)$$

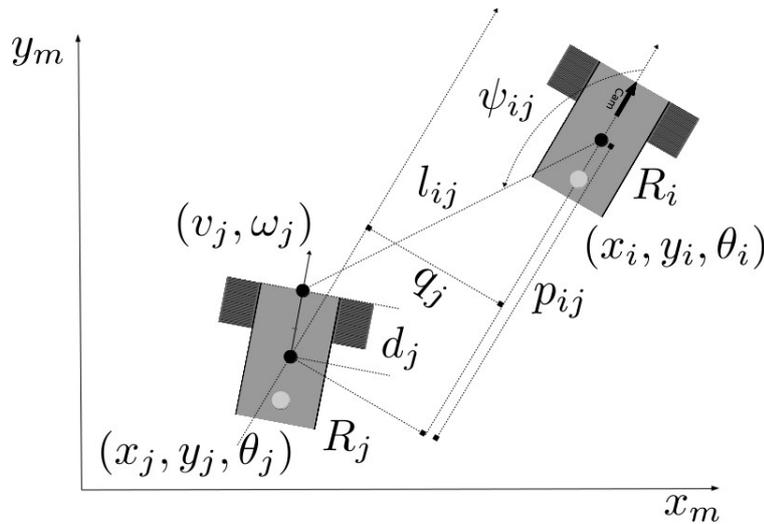


Figura 5.22: Estratégia Líder-Seguidor.

Após derivação de (5.59) e (5.60) obtém-se:

$$\dot{l}_{ij} = \frac{1}{l_{ij}} \left[(x_i - x_j - d_j \cos \theta_j)(\dot{x}_i - \dot{x}_j + d_j \sin \theta_j \dot{\theta}_j) + (y_i - y_j - d_j \sin \theta_j)(\dot{y}_i - \dot{y}_j - d_j \cos \theta_j \dot{\theta}_j) \right]; \quad (5.62)$$

$$\dot{\psi}_{ij} = \frac{1}{l_{ij}^2} \left[(x_i - x_j - d_j \cos \theta_j)(\dot{y}_i - \dot{y}_j - d_j \cos \theta_j \dot{\theta}_j) - (y_i - y_j - d_j \sin \theta_j)(\dot{x}_i - \dot{x}_j + d_j \sin \theta_j \dot{\theta}_j) \right] - \dot{\theta}_i. \quad (5.63)$$

Da análise da Figura 5.22 e de (5.61) obtém-se as seguintes relações:

$$x_i - x_j - d_j \cos \theta_j = l_{ij} \cos \phi_{ij}; \quad (5.64)$$

$$y_i - y_j - d_j \sin \theta_j = l_{ij} \sin \phi_{ij}. \quad (5.65)$$

Substituindo tais relações, o modelo cinemático do líder i e do seguidor j e utilizando propriedades trigonométricas elementares em (5.62) e (5.63) obtém-se:

$$\dot{l}_{ij} = v_i \cos(\theta_i - \phi_{ij}) - v_j \cos(\theta_j - \phi_{ij}) + d_j \omega_j \sin(\theta_j - \phi_{ij}); \quad (5.66)$$

$$\dot{\psi}_{ij} = \frac{1}{l_{ij}} [v_i \sin(\theta_i - \phi_{ij}) - v_j \sin(\theta_j - \phi_{ij}) + d_j \omega_j \cos(\theta_j - \phi_{ij})] - \omega_i. \quad (5.67)$$

Define-se os seguintes ângulos auxiliares:

$$\gamma_{ij} = \beta_{ij} + \psi_{ij}; \quad (5.68)$$

$$\beta_{ij} = \theta_i - \theta_j. \quad (5.69)$$

Finalmente, tem-se o seguinte modelo cinemático:

$$\dot{l}_{ij} = v_j \cos \gamma_{ij} + d_j \omega_j \sin \gamma_{ij} - v_i \cos \psi_{ij}; \quad (5.70)$$

$$\dot{\psi}_{ij} = \frac{1}{l_{ij}} (-v_j \sin \gamma_{ij} + d_j \omega_j \cos \gamma_{ij} + v_i \sin \psi_{ij}) - \omega_i, \quad (5.71)$$

As distâncias e os ângulos de referência, em termos dos parâmetros da formação são obtidos da seguinte forma:

$$l_{ij}^r = \sqrt{(-p_{ij} - d_j)^2 + q_j^2}; \quad (5.72)$$

$$\psi_{ij}^r = \pi - \tan^{-1} \left(\frac{q_j}{-p_{ij} - d_j} \right). \quad (5.73)$$

Define-se um modelo de erros como segue:

$$e_l = l_{ij}^r - l_{ij}; \quad (5.74)$$

$$e_\psi = \psi_{ij}^r - \psi_{ij}; \quad (5.75)$$

$$e_\theta = \beta_{ij}. \quad (5.76)$$

Considerando referências constantes, após derivação temporal, obtém-se o seguinte modelo cinemático para os erros:

$$\dot{e}_l = -v_j \cos \gamma_{ij} - d_j \omega_j \sin \gamma_{ij} + v_i \cos \psi_{ij}; \quad (5.77)$$

$$\dot{e}_\psi = \frac{1}{l_{ij}} (v_j \sin \gamma_{ij} - d_j \omega_j \cos \gamma_{ij} - v_i \sin \psi_{ij}) + w_i; \quad (5.78)$$

$$\dot{e}_\theta = \omega_i - \omega_j. \quad (5.79)$$

Para a aplicação no *NMPC*, definem-se as seguintes entradas:

$$u_{1j} = v_i \cos \psi_{ij} - v_j \cos \gamma_{ij}; \quad (5.80)$$

$$u_{2j} = \omega_i - \omega_j. \quad (5.81)$$

Finalmente, considerando os erros como os estados, l_{ij} e ψ_{ij} como saídas, tem-se o seguinte modelo:

$$\dot{e}_l = u_{1j} - d_j \omega_j \sin \gamma_{ij}; \quad (5.82)$$

$$\dot{e}_\psi = \frac{1}{l_{ij}} (v_j \sin \gamma_{ij} - d_j \omega_j \cos \gamma_{ij} - v_i \sin \psi_{ij}) + w_i; \quad (5.83)$$

$$\dot{e}_\theta = u_{2j}. \quad (5.84)$$

Deste modo, considerando que o robô líder resolve localmente o seu problema de seguimento de caminhos visuais, o problema de controlar os robôs seguidores em formação através da estratégia Líder-Seguidor pode ser resumido da seguinte forma:

Calcular w_j e v_j tal que u_{1j} , u_{2j} , \dot{e}_l , \dot{e}_ψ e \dot{e}_θ sejam factíveis.

Do modelo obtido, observa-se que não existe nenhuma relação explícita com a curvatura ou qualquer outra grandeza crítica do caminho visual seguido pelo robô líder, sendo, desta forma, desacoplados os estados do mesmo com relação aos seguidores. Nominalmente, a função objetivo segue a mesma estrutura utilizada para a resolução dos problemas de seguimento de caminho, ou seja, não existe termo de acoplamento na função objetivo, já que, o acoplamento, neste caso, é feito via restrições para os controladores locais dos j seguidores. A estrutura de controle para cada um destes seguidores é apresentada na Figura 5.23

Garantia de estabilidade

Assim como para o seguimento de caminhos visuais, para os robôs seguidores utiliza-se a abordagem para garantia de estabilidade baseada na adição de um termo de penalidade e de restrições terminais ao problema de otimização.

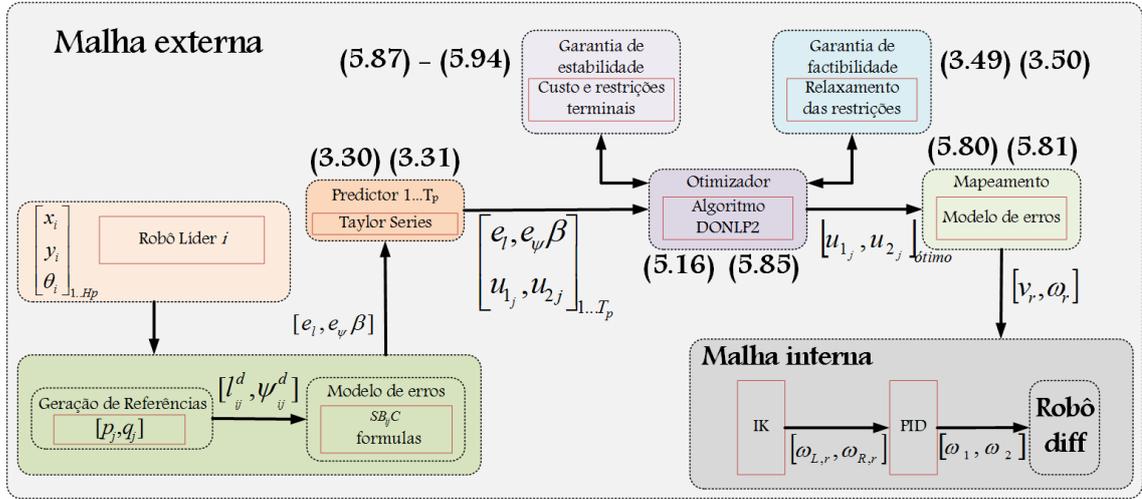


Figura 5.23: Estrutura de controle para cada robô seguidor - Estratégia Líder-Seguidor.

A seguinte função de Lyapunov é escolhida como termo de penalidade:

$$V(\mathbf{x}_e(t + T_p)) = \frac{1}{2} \mathbf{x}_e(t + T_p)^T \mathbf{P} \mathbf{x}_e(t + T_p), \quad (5.85)$$

sendo $\mathbf{x}_e(t + T_p) = [e_{l_T}, e_{\psi_T}, e_{\theta_T}]^T$ o estado terminal e \mathbf{P} definida positiva.

Para a ação de controle terminal u_e^T , a seguinte condição deve ser satisfeita:

$$\dot{V}(\mathbf{x}_e(t)) + F(t, \mathbf{x}_e(t), \mathbf{u}_e(t)) \leq 0. \quad (5.86)$$

Considerando $u_{eT} = [-\alpha_1 e_{l_T}, -\alpha_2 e_{\theta_T}]^T$, com α_1 e $\alpha_2 \in \mathbb{R}_+$ e os seguintes formatos para as matrizes de ponderação:

$$Q = \text{diagonal}(q_{11}, q_{22}, q_{33});$$

$$R = \text{diagonal}(r_{11}, r_{22});$$

$$P = \text{diagonal}(p_{11}, p_{22}, p_{33}),$$

de (5.85) em (5.86), tem-se a seguinte condição de estabilidade:

$$\begin{aligned} & p_{11} e_{l_T} \dot{e}_{l_T} + p_{22} e_{\psi_T} \dot{e}_{\psi_T} + p_{33} e_{\theta_T} \dot{e}_{\theta_T} + F(t + T_p) \\ = & p_{11} e_{l_T} (u_{1j} - d_j \omega_j \sin \gamma_{ij}) + p_{22} e_{\psi_T} \left[\frac{1}{l_{ij}} (v_j \sin \gamma_{ij} - d_j \omega_j \cos \gamma_{ij} - v_i \sin \psi_{ij}) + w_i \right] + \\ & + p_{33} e_{\theta_T} u_{2j} + q_{11} e_{l_T}^2 + q_{22} e_{\psi_T}^2 + q_{33} e_{\theta_T}^2 + R u_{eT} \\ = & (-p_{11} \alpha_1 + q_{11} + r_{11} \alpha_1^2) e_{l_T}^2 - (p_{33} \alpha_2 - q_{33} - r_{22} \alpha_2^2) e_{\theta_T}^2 \\ & + q_{22} e_{\psi_T}^2 - p_{11} d_j \omega_j \sin \gamma_{ij} e_{l_T} + p_{22} e_{\psi_T} \left[\frac{1}{l_{ij}} (v_j \sin \gamma_{ij} - d_j \omega_j \cos \gamma_{ij} - v_i \sin \psi_{ij}) + w_i \right]. \end{aligned}$$

Deste modo, tem-se as seguintes condições para a determinação dos pesos das matrizes de ponderação:

$$\begin{aligned} -p_{11}\alpha_1 + q_{11} + r_{11}\alpha_1^2 &\leq 0; \\ p_{33}\alpha_2 - q_{33} - r_{22}\alpha_2^2 &\geq q_{22}. \end{aligned}$$

A região terminal dos estados Ω é definida da seguinte forma:

$$|e_{\theta_T}| \geq |e_{\psi_T}|; \quad (5.87)$$

$$\omega_j \sin \gamma_{ij} e_{l_T} \geq 0; \quad (5.88)$$

$$(v_j \sin \gamma_{ij} - d_j \omega_j \cos \gamma_{ij} - v_i \sin \psi_{ij} + w_i) e_{\psi_T} \leq 0; \quad (5.89)$$

$$|e_{l_T}| \leq \frac{v_j \cos \gamma_{ij} - v_i \cos \psi_{ij}}{\alpha_1}; \quad (5.90)$$

$$|e_{\theta_T}| \leq \frac{\omega_j - \omega_i}{\alpha_2}. \quad (5.91)$$

sendo as ações de controle físicas restritas da seguinte forma:

$$v_{j_{min}} \leq v_j \leq v_{j_{max}}; \quad (5.92)$$

$$\omega_{j_{min}} \leq \omega_j \leq \omega_{j_{max}}. \quad (5.93)$$

Resultados Experimentais

Inicialmente, objetivando comparar o desempenho das duas propostas, mantém-se a implementação sequencial, mesmo caminho e parâmetros da formação utilizados na subseção anterior. Velocidade de navegação e período de amostragem também são os mesmos. Os parâmetros de sintonia utilizados são apresentados na Tabela 5.6

A matriz de adjacência, neste caso, assume o seguinte formato:

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.94)$$

Os limites para as restrições foram definidos como segue:

$$|u_1| \leq 0.2; |u_2| \leq 5; |e_l| \leq 0,05 \text{ m}; |e_\psi| \leq 0,5 \text{ rad}; |e_\theta| \leq 1,5 \text{ rad}.$$

Tabela 5.6: Matrizes de ponderação.

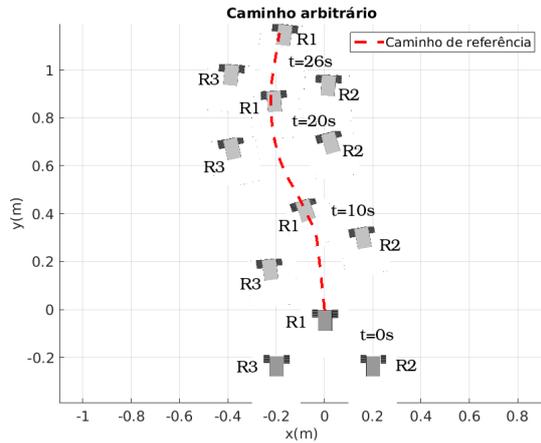
\mathbf{Q}_i	<i>diagonal</i> (50;0,5)
\mathbf{R}_i	0,1
\mathbf{Q}_j	<i>diagonal</i> (200;10;1)
\mathbf{R}_j	<i>diagonal</i> (0,1;0,001)

Mantendo-se $T_p = T_c = 3$ amostras, ainda com R_1 real, R_2 e R_3 virtuais, foram adquiridos os resultados ilustrados na Figura 5.24.

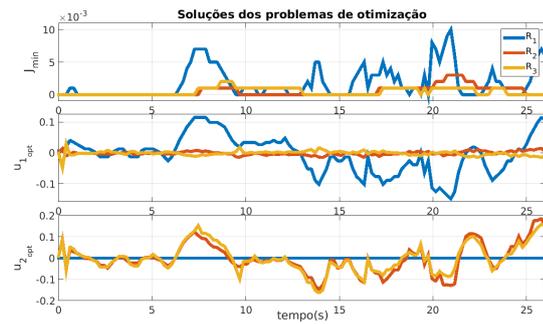
É possível verificar através da Figura 5.24(a), que o caminho foi seguido corretamente pelo robô líder, e a formação triangular foi controlada a todos os instantes. A Figura 5.24(b) ilustra as soluções dos problemas de otimização locais. As funções objetivas locais minimizadas assumiram valores inferiores ao caso anterior e as ações de controle mínima satisfizeram as restrições, sem maiores oscilações, independente da curvatura do caminho. Os erros de estado se mantiveram inferiores aos limites das restrições, conforme ilustrado na Figura 5.24(c) para o líder e na Figura 5.24(d) para os seguidores.

Ainda na Figura 5.24(d), é possível confirmar a boa regulação dos erros de formação, inclusive nos trechos mais curvos, já que distâncias inferiores a 0,02m e ângulos inferiores a 0,2 rad foram obtidos.

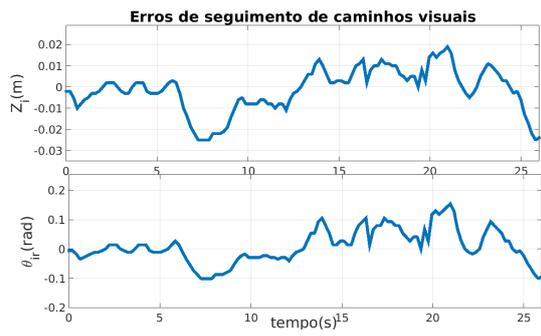
Os esforços de controle necessários a este experimento estão ilustrados na Figura 5.24(e), de onde verifica-se que as velocidades lineares e angulares foram ajustadas corretamente mesmo sem nenhuma especificação prévia como existente na abordagem da subseção anterior. Através da Figura 5.24(f), confirma-se a manutenção da eficiência computacional dos controladores de formação.



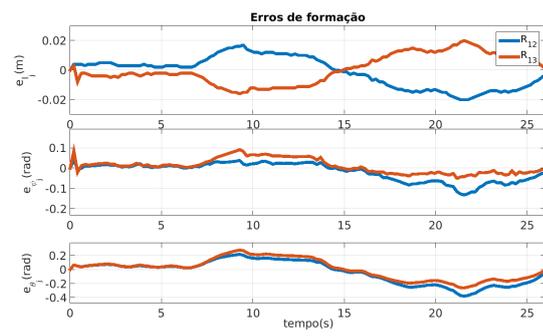
(a) Posições instantâneas dos robôs.



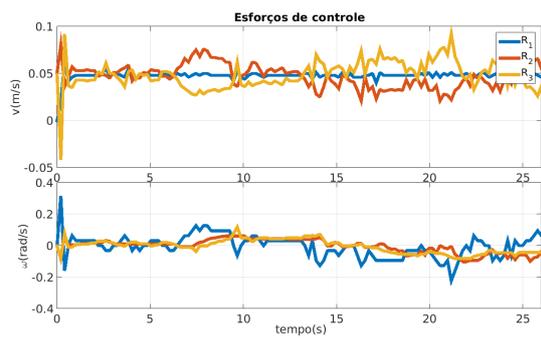
(b) Soluções do problema de otimização.



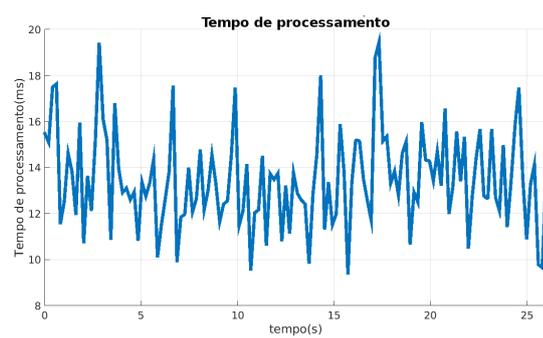
(c) Erros de seguimento de caminho visual.



(d) Erros de formação.



(e) Esforços de controle.



(f) Tempo de processamento.

Figura 5.24: Controle de formação utilizando informações visuais - Estratégia Líder-Seguidor - Caso nominal.

Conforme elucidado no capítulo anterior, a estimativa da curvatura em um ponto do caminho visual é bastante sensível aos erros de seguimento. Com o objetivo de comparar as curvaturas adquiridas pelo robô líder nos dois casos, tal grandeza é avaliada ao longo deste experimento e do anterior, conforme ilustrado na Figura 5.25.

Verifica-se que, com a abordagem baseada em câmeras virtuais, tal grandeza é bastante variável, influenciando bastante para o cálculo das velocidades de referências, o que justifica o desempenho inferior em trechos curvos. Para estratégia Líder-Seguidor, tal grandeza apresenta variação apenas em instantes de elevada curvatura, reflexo do bom desempenho dos controladores.

Para complementar a validação desta abordagem para controle de formação é necessário descentralizar o cálculo das ações de controle. Desde que atrasos e falhas de comunicação possam ser desprezados, tem-se como principal vantagem o aumento do poder computacional, já que existirá um nó de processamento para cada robô, permitindo a avaliação do desempenho da técnica em situações mais extremas, além de viabilizar a implementação das técnicas para garantia de estabilidade e factibilidade para os robôs seguidores.

No contexto atual, tais situações mais extremas podem ser representadas por caminhos de grande comprimento e com trechos de elevada curvatura. Desde que apenas o robô líder implementa a técnica de seguimento de caminho visual, problemas oriundos do cálculo do comprimento de caminho e curvatura visuais, não influenciam diretamente nos controladores locais dos seguidores. Para tal fim, utiliza-se o mesmo caminho arbitrário da Figura 4.27.

Considerando o desempenho superior do controlador *NMPC* com garantia de factibilidade para seguimento de caminhos visuais, tal estratégia é utilizada para robô líder real (R_1) em todas as avaliações aqui realizadas. Além disso, os robôs seguidores (R_2 e R_3) continuam sendo virtuais.

Inicialmente, avalia-se o desempenho desta proposta para formação triangular fixa, com mesma matriz de adjacência do experimento anterior (5.94). As matrizes de

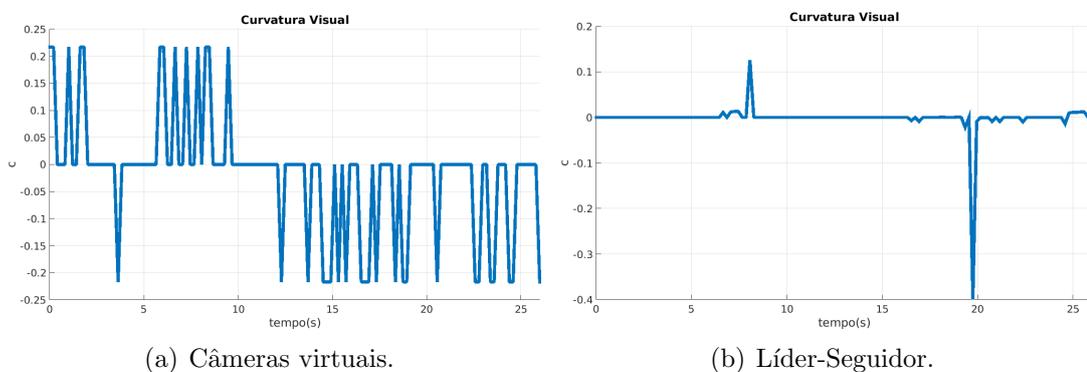


Figura 5.25: Comparação das curvaturas estimadas pelo sistema visual.

ponderação são fornecidas pela Tabela 5.7 e são definidos mesmos valores limitantes para as restrições, velocidade de navegação e horizontes do experimento anterior. Com os seguidores implementando o *NMPC* nominal, foram obtidos os resultados ilustrados na Figura 5.26

A Figura 5.26(a) ilustra algumas poses instantâneas no plano, sendo possível confirmar que a formação foi devidamente controlada.

Para explicitar os requisitos de factibilidade e estabilidade desta técnica, as soluções dos problemas de otimização locais são apresentadas na Figura 5.26(b). Verifica-se que a função objetivo se manteve com baixo valor, inclusive nos trechos mais curvos e que as ações de controle internas ótimas permaneceram dentro dos limites das restrições. Porém, mesmo que de baixa magnitude, os aumentos pontuais nas funções objetivos dos seguidores motiva a análise da estabilidade para o tratamento de formações variantes no tempo.

Os erros de seguimento de caminhos visuais e os erros de formação estão dispostos nas Figuras 5.26(c) e 5.26(d), respectivamente, sendo satisfeitas, em todos os casos, as restrições nas saídas. Os esforços de controle se mantiveram compatíveis com plataforma experimental (5.26(e)) e a eficiência computacional da técnica foi preservada também para este experimento (5.26(f)).

Em seguida, habilita-se o estágio para garantia de estabilidade apenas para os robôs seguidores com $P = \text{diagonal}(200, 200, 200)$, $\alpha_1 = 2$ e $\alpha_2 = 1$. A Figura 5.27 apresenta os resultados obtidos. Observa-se que o caminho visual foi seguido em formação corretamente, porém, para a manutenção da estabilidade, a função objetivo se tornou bem maior devido ao custo terminal. Ainda na Figura 5.27(b) verifica-se que todas as ações de controle ótimas se mantiveram dentro dos limites das restrições.

Os erros de formação ilustrados na Figura 5.27(d) indicam desempenho inferior, já que maior parte da zona das restrições foi utilizada, porém, sem violação alguma. A Figura 5.27(f) mostra que, diferentemente da garantia de estabilidade para seguimento de caminhos visuais, o desempenho computacional foi bastante degradado, neste caso, o tempo de processamento ultrapassou o período de amostragem em diversos momentos, para ambos os seguidores.

Dentre os potenciais prejuízos desta degradação do tempo de processamento, cita-se, por exemplo, a falta de capacidade de processamento para a execução de algoritmos adicionais, tal como filtro de Kalman para melhor estimativa da pose do robô líder.

Tabela 5.7: Matrizes de ponderação para os seguidores

\mathbf{Q}_j	$\text{diagonal}(200;0,1;0,001)$
\mathbf{R}_j	$\text{diagonal}(0,00001;0,0001)$

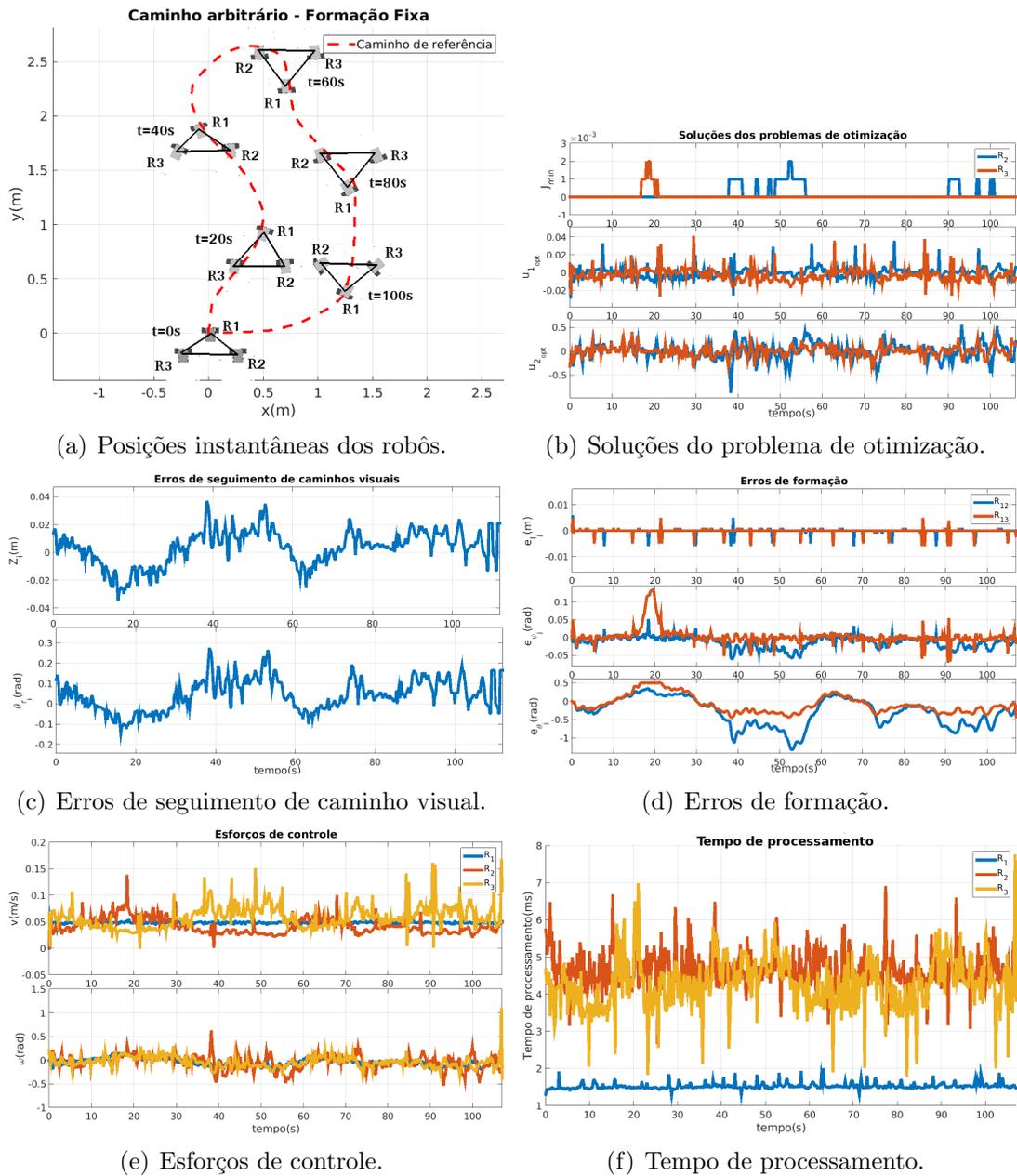
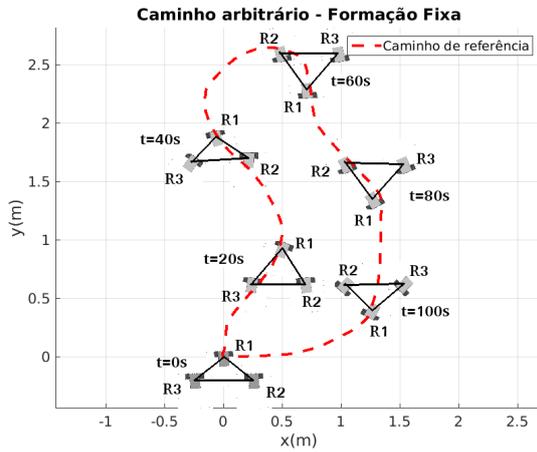


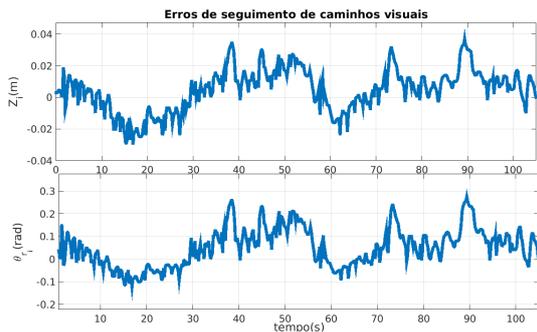
Figura 5.26: Controle de formação fixas utilizando informações visuais - Estratégia Líder-Seguidor - Caso nominal para os seguidores.



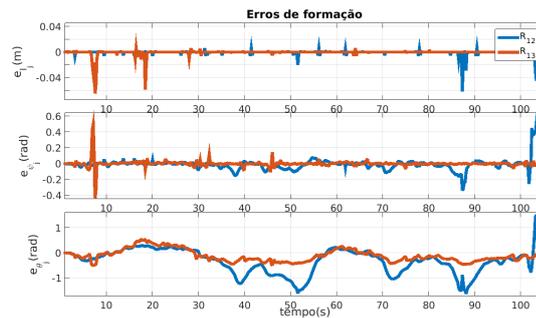
(a) Posições instantâneas dos robôs.



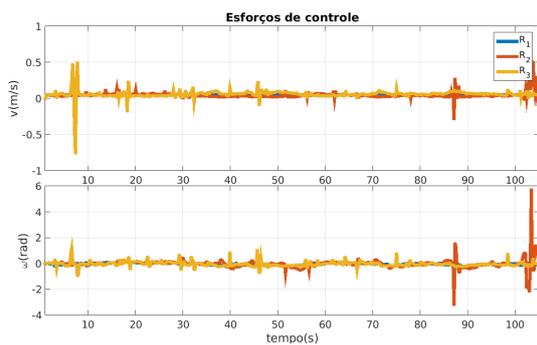
(b) Soluções do problema de otimização.



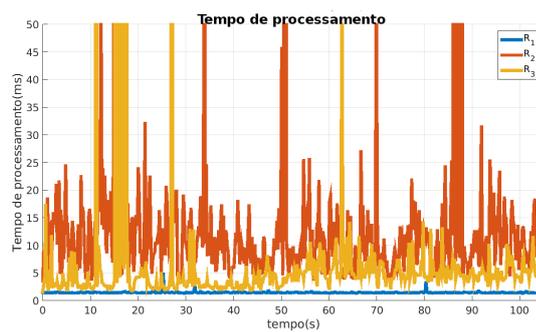
(c) Erros de seguimento de caminho visual.



(d) Erros de formação.



(e) Esforços de controle.



(f) Tempo de processamento.

Figura 5.27: Controle de formação fixas utilizando informações visuais - Estratégia Líder-Seguidor - Garantia de estabilidade para os seguidores.

Ainda considerando caminhos fixos, avalia-se o desempenho da proposta quando se utiliza apenas a estratégia para garantia de factibilidade para os robôs seguidores. Dos resultados anteriores, conclui-se que, para os mesmos parâmetros de sintonia, a menos do tempo de processamento, resultados similares devem ser obtidos. Neste caso, os limitantes das restrições são dados como segue:

$$|u_{j1}| \leq 10^3; |u_{j2}| \leq 10^3;$$

$$|e_{l_j}| \leq 0,05 + e_{l_j}^{max} m; |e_{\psi_j}| \leq 0,5 + e_{\psi_j}^{max} rad; |e_{\theta_j}| \leq 1,5 + e_{\theta_j}^{max} rad.$$

A Figura 5.29 apresenta os resultados obtidos sendo possível verificar o correto atendimento dos objetivos de controle. A Figura 5.29(b) mostra que, assim como no caso nominal, também houve aumentos pontuais de baixa magnitude na função objetivo local e entradas internas ótimas apresentaram comportamento similar. A grande vantagem neste caso, consiste em manter a factibilidade independente dos erros de estado atuais.

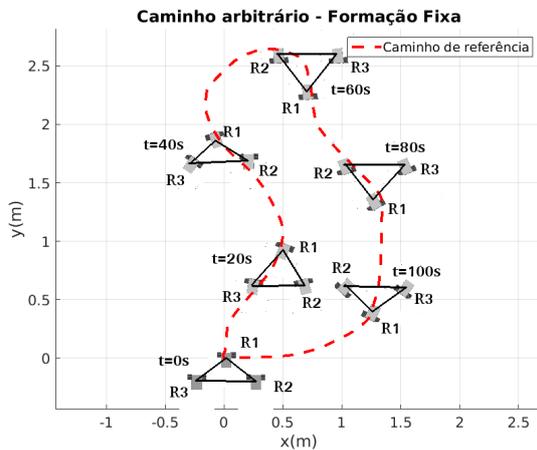
Na Figura 5.29(d) é possível verificar que, mesmo com as restrições relaxadas, os valores dos erros de estado foram ainda menores que todos os casos anteriores, sendo, portanto, a formação mais bem controlada. Além disso os esforços de controle e computacional se mantiveram próximos do caso nominal.

Conforme discutido no capítulo anterior, a implementação da técnica para garantia de factibilidade no robô líder, viabiliza a regulação de erros elevados, sendo necessário avaliar como se comportam os seguidores em formação quando surgem mudanças abruptas no caminho visual. Da continuação dos experimentos anteriores, pelo fato do caminho arbitrário ser fechado, surge uma variação abrupta no caminho visual sob a forma de descontinuidade, conforme ilustrado na Figura 5.28.

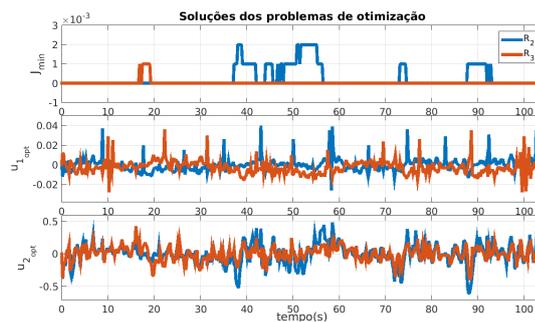
Este tipo de perturbação exige esforços de controle elevados do robô líder que quando aplicados ao sistema odométrico, produzem medições de deslocamentos no plano que tendem a ser bastante ruidosas. Tais deslocamentos são estritamente necessários para a implementação das técnicas baseadas em Líder-Seguidor, sendo portanto, uma forma importante de evidenciar aspectos positivos das mesmas.



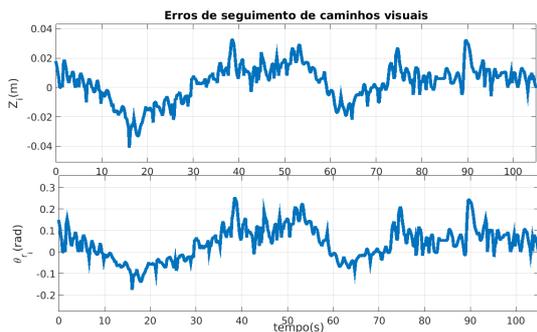
Figura 5.28: Imagem capturada ao final do caminho utilizado.



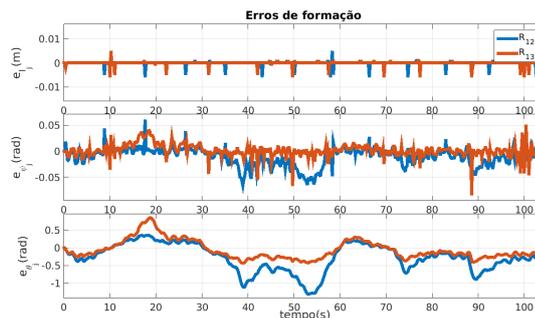
(a) Posições instantâneas dos robôs.



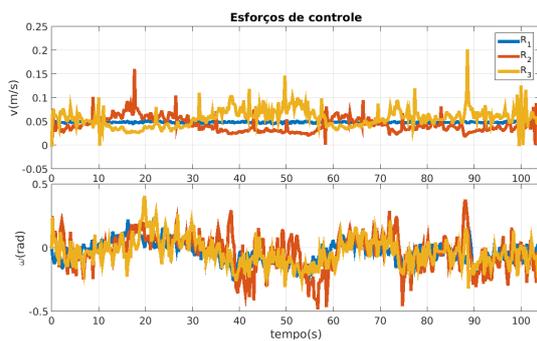
(b) Soluções do problema de otimização.



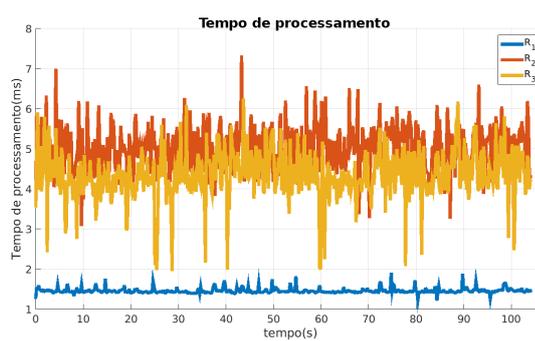
(c) Erros de seguimento de caminho visual.



(d) Erros de formação.



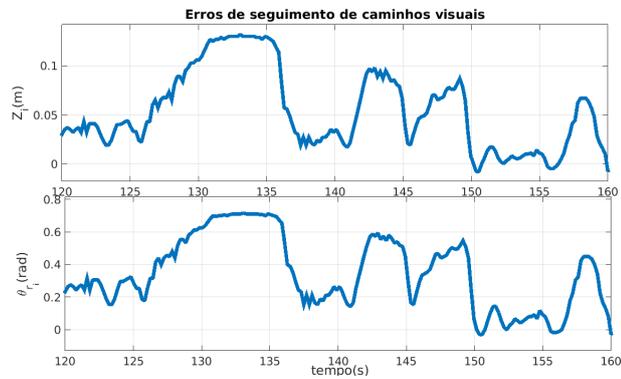
(e) Esforços de controle.



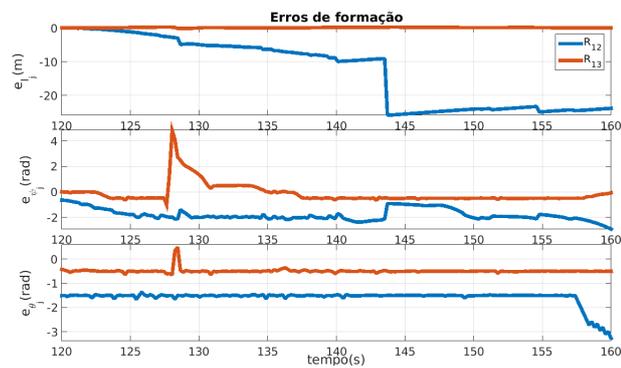
(f) Tempo de processamento.

Figura 5.29: Controle de formação fixas utilizando informações visuais - Estratégia Líder-Seguidor - Garantia de factibilidade para os seguidores.

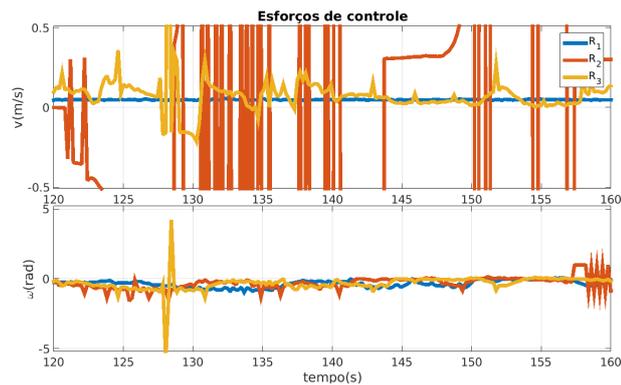
As Figuras 5.30, 5.31 e 5.32 ilustram o comportamento das três propostas neste cenário.



(a) Erros de seguimento de caminho.

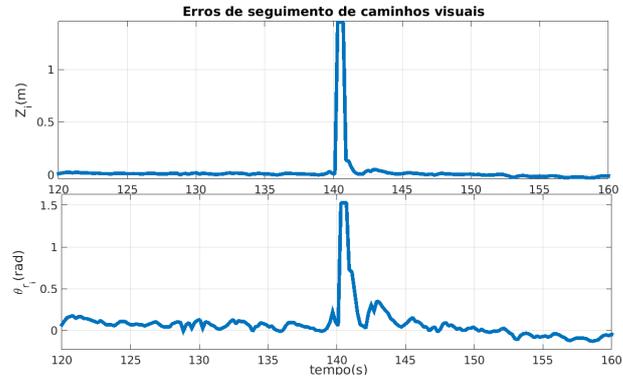


(b) Erros de formação.

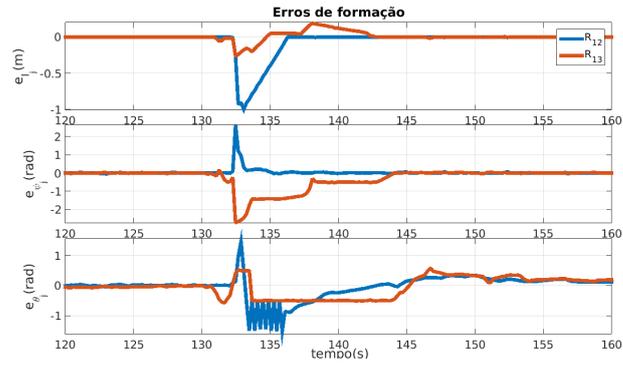


(c) Esforços de controle.

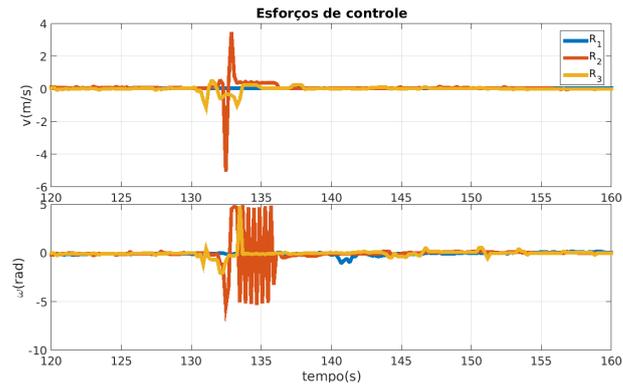
Figura 5.30: Comportamento do *NMPC* nominal na continuação do experimento ao fim do caminho.



(a) Erros de seguimento de caminho.

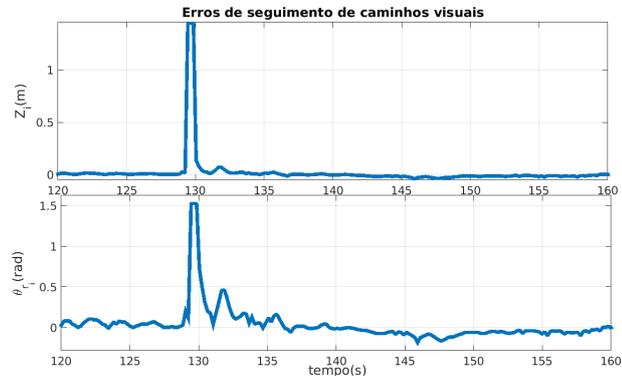


(b) Erros de formação.

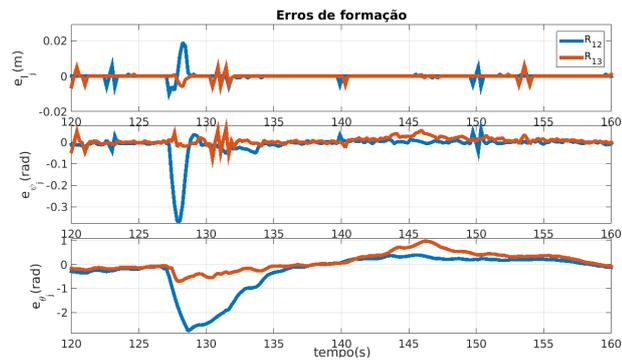


(c) Esforços de controle.

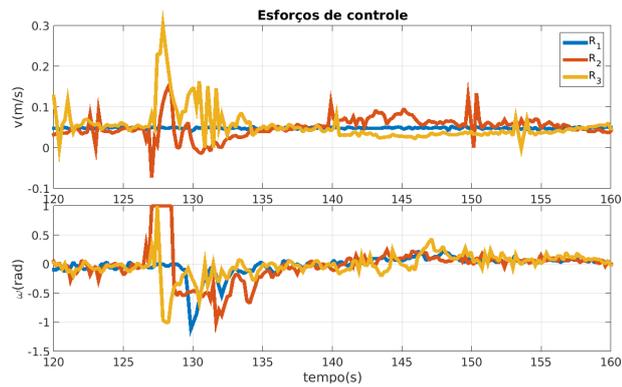
Figura 5.31: Comportamento do *NMPC* com garantia de estabilidade na continuação do experimento ao fim do caminho.



(a) Erros de seguimento de caminho.



(b) Erros de formação.



(c) Esforços de controle.

Figura 5.32: Comportamento do *NMPC* com garantia de factibilidade na continuação do experimento ao fim do caminho.

Verifica-se que os controladores *NMPC* com garantia de estabilidade e com garantia de factibilidade conseguiram controlar a formação durante a regulação ao longo do caminho visual. Porém, os esforços de controle ilustrados em 5.31(c) não são praticáveis, sendo indicativo da não aplicabilidade prática da técnica para garantia de estabilidade quando todos os robôs forem reais.

Este resultado ratifica mais uma vez as melhores características de desempenho dos controladores *NMPC* com garantia de factibilidade em condições de funcionamento adversas.

Finalmente, para o mesmo caminho de referência, são consideradas formações variantes no tempo. Especificamente, partindo da mesma configuração dos experimentos anteriores, os parâmetros da formação são modificados em três intervalos de tempo específicos, sendo que ao final, retoma-se o formato geométrico original, mais especificamente:

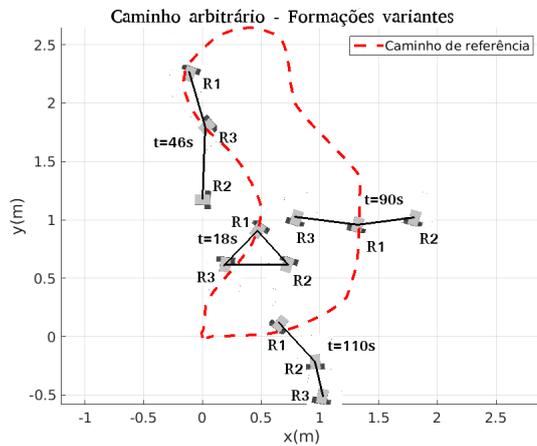
- $20 \leq t < 60$: Formação em pelotão - ($q_2 = q_3 = 0\text{m}$, $p_{21} = 1, \text{m}$ e $p_{31} = 0,5\text{m}$);
- $60 \leq t < 100$: Formação em coluna - ($q_2 = 0.5$, $q_3 = -0.5\text{m}$, $p_{21} = p_{31} = 0\text{m}$);
- $t \geq 100$: Formação triangular original.

Diferentemente das abordagens para controle de formações variantes ao longo de caminhos pré-especificados, em que tais modificações de parâmetros geram perturbações diretamente nas funções objetivos locais, neste caso, as perturbações são aplicadas diretamente às restrições dos controladores dos seguidores, o que tende em aumentar demasiadamente o custo computacional, não sendo possível regular os erros em um intervalo de tempo satisfatório.

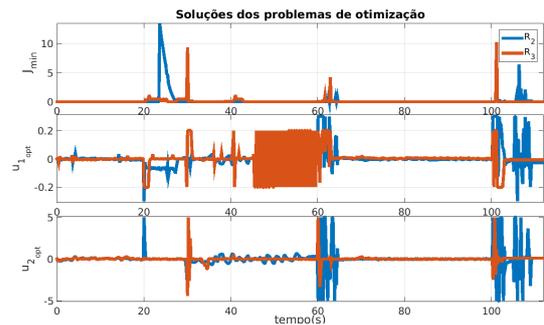
Com os mesmos parâmetros de sintonia e velocidade de navegação, foram obtidos os resultados ilustrados nas Figuras 5.33, 5.34 e 5.35.

Observa-se que a proposta para garantia de factibilidade é a única abordagem que consegue controlar formações variantes no tempo com esforços de controle praticáveis. Através do relaxamento das restrições, foi possível manter a factibilidade recursiva, fazendo os erros serem reduzidos, a cada instante de amostragem, para valores físicos permissíveis no ambiente de navegação.

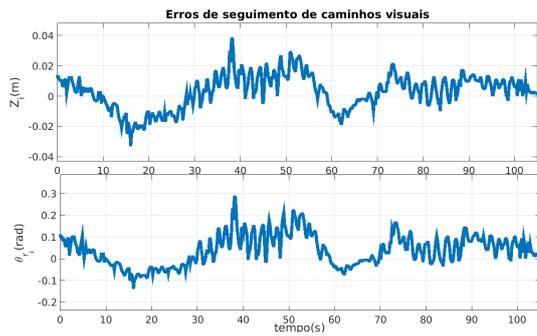
Para esta abordagem, é interessante notar mais uma vez que, ações de controle internas de grande magnitude são geradas, com posterior saturação das ações físicas em valores compatíveis com o tipo de robô utilizado. As perturbações foram reguladas em menos de 5s e o tempo de processamento se manteve sempre inferior a 10ms, o que confirma a eficiência da técnica.



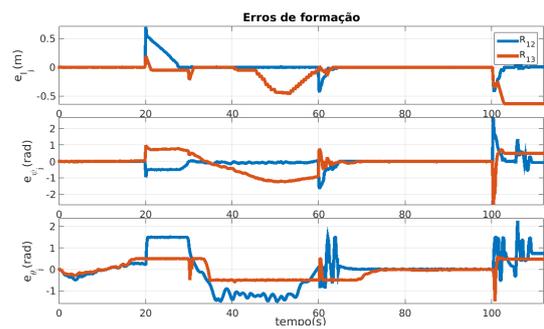
(a) Posições instantâneas dos robôs.



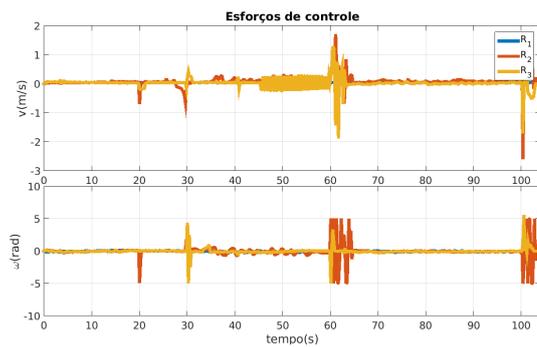
(b) Soluções do problema de otimização.



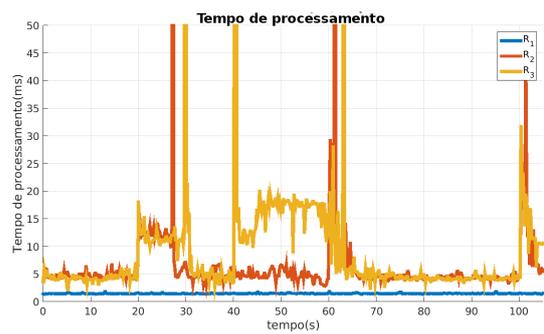
(c) Erros de seguimento de caminho visual.



(d) Erros de formação.

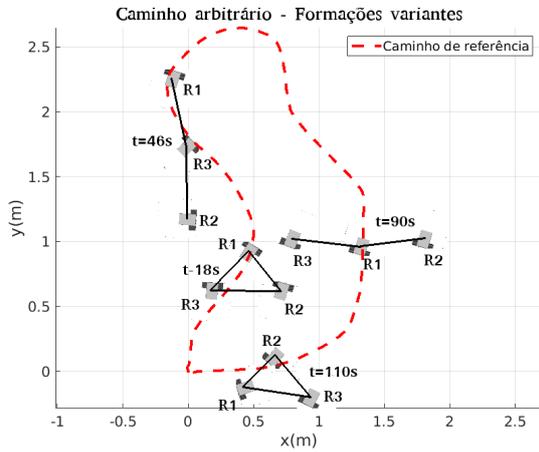


(e) Esforços de controle.

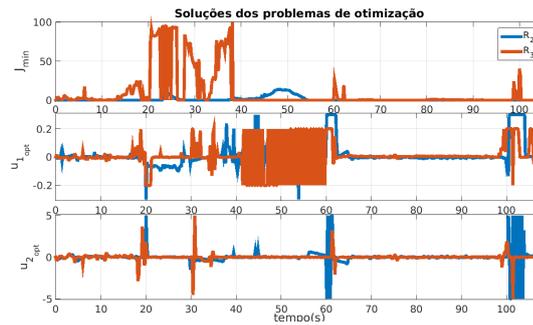


(f) Tempo de processamento.

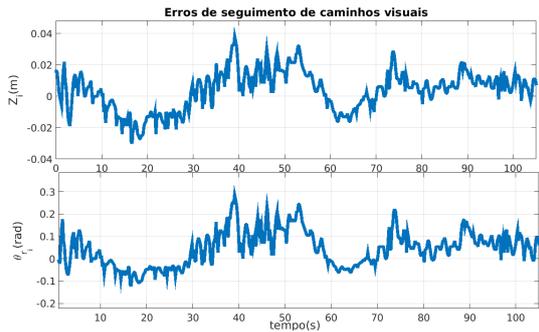
Figura 5.33: Controle de formações variantes utilizando informações visuais - Estratégia Líder-Seguidor - Caso Nominal.



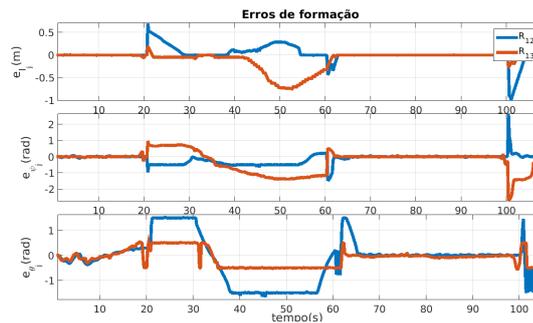
(a) Posições instantâneas dos robôs.



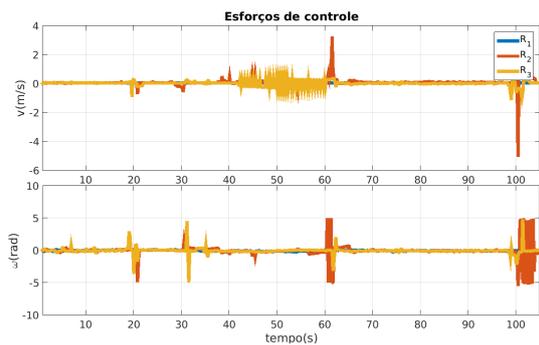
(b) Soluções do problema de otimização.



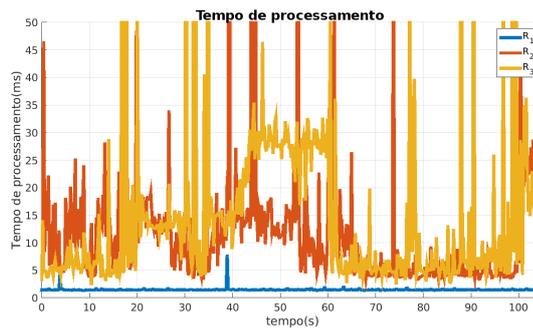
(c) Erros de seguimento de caminho visual.



(d) Erros de formação.

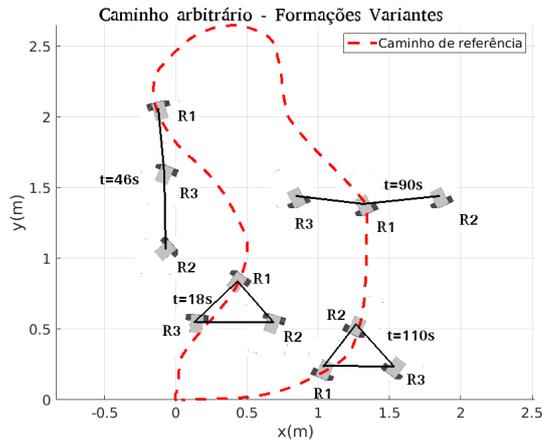


(e) Esforços de controle.

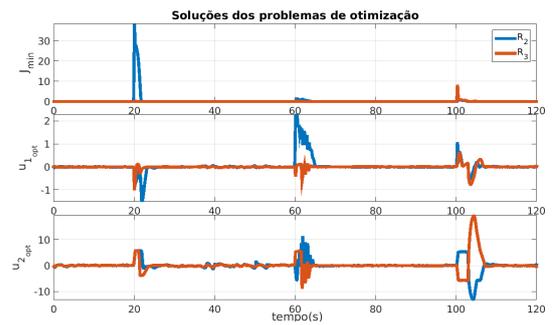


(f) Tempo de processamento.

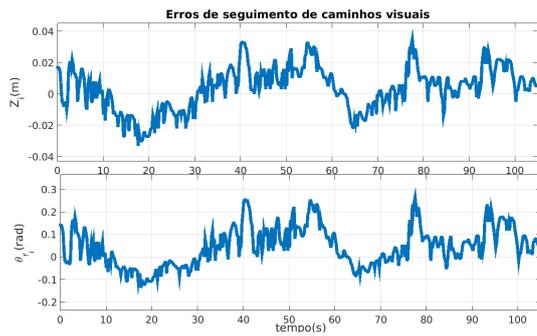
Figura 5.34: Controle de formações variantes utilizando informações visuais - Estratégia Líder-Seguidor - Garantia de estabilidade.



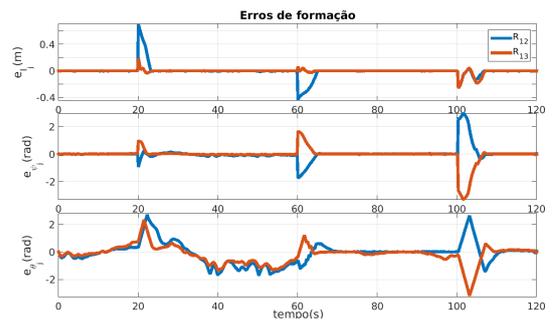
(a) Posições instantâneas dos robôs.



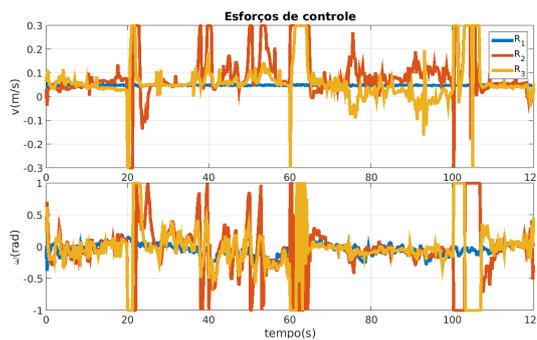
(b) Soluções do problema de otimização.



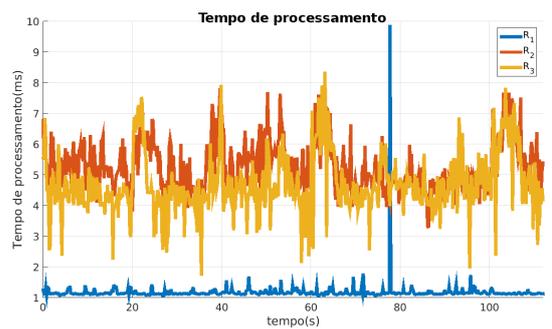
(c) Erros de seguimento de caminho visual.



(d) Erros de formação.



(e) Esforços de controle.



(f) Tempo de processamento.

Figura 5.35: Controle de formações variantes utilizando informações visuais - Estratégia Líder-Seguidor - Garantia de factibilidade.

5.6 Conclusões

Este capítulo foi dedicado à formalização dos problemas de controle de formação, ao projeto e desenvolvimento de controladores *NMPC* para diversos tipos de robôs móveis, mais especificamente, omnidirecionais, diferenciais e quadricópteros.

A caracterização do fluxo de informação entre os elementos da formação é feita sob a forma de grafos. Trata-se de uma forma eficiente e reconhecida de caracterizar como deve ser feita a troca de informações entre os robôs em formação, já que dispõe de um ferramental matemático preciso e consolidado com aplicação direta para o controle automático de sistemas distribuídos.

Inicialmente, os aspectos teóricos básicos necessários à implementação dos controladores descentralizados foram apresentados. Em seguida, as duas topologias implementadas neste trabalho, especificamente, formação genérica e Líder-Seguidor, foram apresentadas e discutidas neste contexto.

O controle de formação ao longo de caminhos pré especificados foi inicialmente formalizado para robôs omnidirecionais. Com os problemas de seguimento de caminhos locais devidamente caracterizados, foi definida a estrutura da função objetivo utilizada no algoritmo *NMPC* descentralizado.

Foram fornecidos resultados experimentais através de uma arquitetura composta de um robô real e dois virtuais, inicialmente para caminhos circulares fixos, em seguida para caminhos e formações variantes no tempo. Estes últimos resultados foram viabilizados pela utilização da estratégia proposta para garantia de factibilidade, já que os controladores *NMPC* nominais não teriam condições de regular as perturbações oriundas das variações de parâmetros.

Tais perturbações são aplicadas diretamente nas funções objetivos através do termo de acoplamento e nas restrições devido as especificações atualizadas para os caminhos de referência locais, de modo que, no caso nominal, ações de controle além dos limites físicos impostos pelas restrições seriam necessárias, tornando o problema infactível.

Para o caso de robôs diferenciais, foram realizados ajustes nos modelos de seguimento de caminhos para contemplar a natureza não holonômica deste tipo de robô. Mantendo-se as estruturas das funções objetivo locais, foram adquiridos resultados experimentais com três robôs reais posicionados para geração de erros elevados, tanto para a formação quanto para o seguimento de caminhos.

Com estes resultados ficaram mais uma vez evidenciados os benefícios proporcionados pela técnica para garantia de factibilidade, já que, mesmo com as imperfeições da plataforma experimental utilizada, tais resultados foram satisfatórios. Nesta etapa ainda foram feitas análises das variáveis internas dos controladores *NMPC* com e sem o relaxamento das restrições, sendo confirmadas experimentalmente as limitações da abordagem original para controle de formação descentralizado.

Em seguida, os conceitos do controle de formação para veículos terrestres foram estendidos para veículos aéreos não tripulados, mais especificamente os quadricópteros. Os modelos de seguimento de caminhos locais foram ajustados, e, também

com manutenção das estruturas das funções objetivos, foram adquiridos resultados simulados que consideram um cenário realístico com rajadas de vento aleatórias e formações variantes no tempo.

Posteriormente, considerando a disponibilidade de apenas um robô equipado com câmera, define-se uma arquitetura composta de um robô líder real e robôs seguidores virtuais. Foram propostas duas soluções para o controle de formação utilizando informações visuais, uma baseando-se no conceito de câmeras e caminhos virtuais e outra utilizando a estratégia básica denominada por Líder-Seguidor.

A primeira proposta pressupõe através de relações geométricas elementares que, quando a formação está controlada, as imagens capturadas pelo câmera real são idênticas àquelas capturadas pelas câmeras virtuais instaladas nos robôs seguidores. Neste caso, a manutenção dos termos de acoplamento, baseando-se em comprimento de caminho, necessita de novos algoritmos para estimação de caminhos virtuais adequados. Porém, limitando-se ao âmbito desta tese, optou-se por definir uma nova métrica para avaliação dos erros de formação.

Através de uma arquitetura composta por um robô real e dois virtuais, foram adquiridos resultados experimentais demonstrando que esta técnica, devido à grande sensibilidade dos erros de formação à curvatura, apresenta dificuldade de regulação em trechos mais curvos e isto se deve, fundamentalmente, à indisponibilidade de mecanismos precisos para a calibração do sistema visual, notadamente para o cálculo da curvatura no ponto principal da imagem.

Finalmente a estratégia Líder-Seguidor básica foi apresentada. Tal estratégia baseia-se no controle da distância e ângulo relativos entre líder e seguidores sendo os próprios modelos de erros no espaço de estados representativos da formação, ou seja, neste caso o acoplamento entre os robôs é feito via estados. Para este caso, além do *NMPC* nominal, implementam-se as técnicas para garantia de estabilidade baseada em custo e restrições terminais e para garantia de factibilidade baseada no relaxamento das restrições.

Para avaliação desta proposta, são considerados dois tipos de caminhos visuais, objetivando evidenciar a independência das características do caminho visual seguido pelo robô líder, um caminho pequeno, aberto e com trechos de baixa curvatura, idêntico ao utilizado nos experimentos da proposta anterior e outro caminho longo, fechado e com trechos de maior curvatura utilizado para avaliação do seguimento de caminho visual apresentado e discutido no Capítulo 4.

Considera-se ainda a aplicabilidade das três modalidades de *NMPC* para formações variantes no tempo, sendo possível confirmar, novamente, o desempenho superior da técnica proposta para garantia de factibilidade.

Todos os controladores desenvolvidos foram validados através de uma plataforma experimental que possui como único mecanismo de localização absoluta um sistema odométrico, já que o desenvolvimento de algoritmos para localização, não faz parte do escopo deste trabalho. As soluções propostas podem ser implementadas em plataformas que dispõem de mecanismos mais precisos de localização, sendo possível a obtenção de resultados ainda melhores.

Capítulo 6

Conclusões e Trabalhos Futuros

Esta tese foi dedicada ao desenvolvimento de algoritmos para a solução de problemas de controle de formação de robôs móveis através de Controladores Preditivos baseados em Modelos Não Lineares, descentralizados. Trata-se de um tema bastante atual e de grande relevância, já que engloba controle de formação, sub tarefa de fundamental importância para o controle cooperativo de robôs móveis e controladores *NMPC*, estratégia bastante difundida em diversos setores da sociedade e cujo número de aplicações é crescente.

O Capítulo 1 fornece uma revisão bibliográfica atualizada do estado da arte e define os problemas de controle automático a serem solucionados. São também apresentadas motivações em diversos domínios para o desenvolvimento de trabalhos desta natureza além dos objetivos gerais e específicos.

A busca das soluções para os problemas enunciados no primeiro capítulo se inicia no Capítulo 2, após caracterização básica dos sistemas robóticos, apresentação dos modelos dos robôs utilizados e detalhamentos da plataforma experimental. Tendo sempre como requisito o desenvolvimento de soluções de baixa complexidade algorítmica, foram obtidos os modelos cinemáticos dos robôs e caracterizados os sistemas computacional e de comunicação. Adicionalmente, a importância desta etapa é evidenciada pelo estabelecimento de níveis de aprofundamento compatíveis com os recursos tecnológicos disponíveis além de demonstrar que as soluções propostas são passíveis de validação através de simulações e experimentos.

Os fundamentos dos controladores *NMPC* são apresentados no Capítulo 3. Inicialmente, são fornecidas algumas preliminares teóricas relativas aos sistemas não lineares, destacando-se o critério de estabilidade segundo Lyapunov, abordagem utilizada neste trabalho para análise de estabilidade e o princípio da invariância de LaSalle utilizado para a análise de factibilidade e estabilidade.

Após detalhamento da estrutura de predição utilizada, o problema de controle ótimo básico é apresentado, bem como, suas vantagens e desvantagens, destacando-se a ausência natural de garantias de factibilidade e estabilidade, aspectos fundamentais para pleno atendimento dos objetivos de controle, especialmente quando se trata de sistemas não lineares, em que os horizontes de predição e controle não podem ser

indefinidamente grandes.

Ainda neste capítulo, são fornecidas as análises do problema da factibilidade, sendo proposta uma estratégia para a sua garantia baseada no relaxamento das restrições. Fundamentalmente, estabelece-se o aproveitamento da estrutura dos modelos a serem controlados para a aplicação de métricas de relaxamento de ambas as restrições, nas entradas e nos estados, sem acréscimo de termo de penalidade na função objetivo, visando não comprometer o custo computacional.

Para garantia de estabilidade, é apresentada uma estratégia consolidada que baseia-se na utilização de um termo de penalidade terminal e uma região terminal para as restrições. Dos requisitos para implementação desta técnica, é possível verificar o invariável incremento da complexidade algorítmica.

O problema de seguimento de caminho, tarefa definida para os robôs em formação, é abordado no Capítulo 4. Inicialmente considerando caminhos fixos e pré especificados, formaliza-se o problema para robôs omnidirecionais e a estratégia de controle baseada em controladores *NMPC* é apresentada. Resultados experimentais demonstram bom desempenho da técnica, e, objetivando incrementar a sua aplicabilidade prática, implementa-se a estratégia proposta para garantia de factibilidade, permitindo o tratamento de erros iniciais elevados, aspecto de fundamental importância para o seguimento de caminhos variantes no tempo.

Em seguida uma extensão da técnica para quadricópteros é proposta. Após a apresentação da estratégia de controle, são apresentados resultados simulados considerando um ambiente realístico em que existem rajadas de vento aleatórias. Tais resultados utilizam a proposta para garantia de factibilidade para uma primeira avaliação do seu comportamento no caso de caminhos variantes no tempo. Nesta etapa, realiza-se uma avaliação das variáveis internas do *NMPC* sendo possível confirmar os benefícios da técnica.

Objetivando generalizar as aplicações, é proposta uma abordagem para seguimento de caminhos visuais baseado em *NMPC* para robôs diferenciais. Tal abordagem consiste em controlar o espaçamento lateral e o ângulo de erro entre um caminho adquirido por uma câmera através de características visuais simplificadas. Diante das imperfeições práticas dos sistemas visuais (cadência visual e resolução) e eletromecânicos (limitações dos sensores e atuadores), implementa-se pela primeira vez a técnica para garantia de estabilidade.

Resultados simulados e experimentais demonstram a aplicabilidade da técnica para três modalidades de controladores *NMPC*, nominal, com estabilidade garantida e com factibilidade garantida. Além disso, nesta etapa ainda é fornecida uma comparação da proposta um controlador proporcional elementar. Tais resultados são adquiridos com dois robôs comerciais distintos, sendo possível validar as propostas em situações reais opostas.

Este capítulo é finalizado com avaliações práticas da influência das velocidades de navegação, variação nos horizontes de controle e predição e tipos diferentes de erros iniciais elevados, onde confirma-se mais uma vez desempenho superior dos controladores *NMPC* com garantia de factibilidade.

Finalmente, o Capítulo 5 trata das estratégias para controle de formação dos três tipos de robôs móveis. Tal capítulo se inicia com a discussão de um dos pré requisitos fundamentais para a implementação de qualquer estratégia de controle descentralizada, a caracterização do fluxo de informações. Especificamente para o controle de formação de robôs móveis, tem-se na teoria dos grafos o suporte necessário para este desenvolvimento.

Em seguida, considerando caminhos pré especificados, projetam-se controladores de formação para robôs omnidirecionais baseados em *NMPC* para o seguimento de caminhos coordenados. Basicamente, tal coordenação de movimentos é alcançada através de ajustes nas velocidades de navegação e nas taxas de atualização das referências de postura para cada robô. Novamente, a proposta para garantia de factibilidade é aplicada para o tratamento de caminhos e formações variantes no tempo e resultados experimentais são apresentados e discutidos.

Os modelos obtidos para o robô omnidirecional são estendidos para os diferenciáveis e quadricópteros e são fornecidos resultados experimentais e simulados, sempre considerando maior aplicabilidade prática. Nesta etapa, também são fornecidas análises das variáveis internas dos controladores *NMPC* locais durante o processo de otimização.

Propostas para o controle de formação utilizando informações visuais são formalizadas ao final deste capítulo. Destaca-se o desempenho da arquitetura composta de um líder que segue um caminho visual arbitrário e seguidores que implementam a técnica Líder-Seguidor básica. Foram projetados controladores *NMPC* descentralizados em três modalidades distintas, mais especificamente, caso nominal, com garantia de estabilidade e com garantia de factibilidade. Resultados experimentais demonstram desempenho superior do *NMPC* com garantia de factibilidade tanto para formações fixas quanto para formações variantes no tempo.

6.1 Principais contribuições

A seguir, são listadas as principais contribuições desta tese ao controle de formação de robôs móveis:

- Validação experimental de controladores *NMPC* descentralizados para controladores de formação de robôs móveis;
- Desenvolvimento de uma estratégia para garantia de factibilidade experimentalmente comprovada para seguimento de caminho e controle de formação de robôs móveis terrestres e aéreos;
- Seguimento de caminhos variantes no tempo baseado em *NMPC* computacionalmente eficiente para robôs móveis terrestres e aéreos;
- Controle de formações variantes no tempo baseado em *NMPC* computacionalmente eficiente para robôs móveis terrestres e aéreos;

- Uma nova abordagem para seguimento de caminhos visuais arbitrários com factibilidade e estabilidade garantidas para robôs diferenciais experimentalmente comprovada;
- Novas propostas para o controle de formação baseado em *NMPC* utilizando informações visuais com factibilidade e estabilidade garantidas para robôs diferenciais experimentalmente comprovada.

6.2 Perspectivas de Trabalhos Futuros

Ao longo do desenvolvimento desta tese, surgiram várias direções interessantes à continuidade da pesquisa nas áreas de controle de formação, controle preditivo não linear, sistemas descentralizados e controle visual, conforme listagem a seguir:

- Avaliação das estratégias propostas para controle de formação considerando formações heterogêneas, constituídas de robôs terrestres e aéreos não tripulados;
- Aplicação de um procedimento ótimo de sintonia dos controladores *NMPC*;
- Prova formal da garantia de factibilidade recursiva da técnica baseada no relaxamento das restrições;
- Avaliações de escalabilidade das proposições;
- Formalização das transições entre as configurações intermediárias das formações variantes no tempo;
- Formalização do procedimento de calibração do sistema visual, especialmente para aquisição de valores mais precisos de curvatura.

6.3 Considerações Finais

É lícito ressaltar que uma contribuição adicional desta tese se refere à utilização de dispositivos e equipamentos de baixo custo financeiro e facilmente disponíveis no mercado, o que de certo modo contribuem para que as avaliações das estratégias propostas tenham sido feitas próximas de um cenário de “pior caso”, o que aumenta as expectativas de aplicação real das mesmas em diversos setores da sociedade.

Os objetivos principais foram alcançados, fruto da modularização das atividades e busca de entendimento nas diversas áreas do conhecimento que constituem a robótica móvel, tais como computação e mecânica. Para isso, foi de fundamental importância a pesquisa direcionada e contato com especialistas nas áreas afins.

Os resultados obtidos introduzem um conjunto de propostas que podem ser reproduzidas e são absolutamente passíveis de integração com os outros níveis requeridos à completa autonomia de um sistema robótico.

Bibliografia

- Albert, Jeremie, Tegawendé F. Bissyandé, Yérom-David Bromberg, Serge Chaumette e Laurent Réveillère (2010). Ubipan: A bluetooth extended personal area network. Em: *CISIS 2010, The Fourth International Conference on Complex, Intelligent and Software Intensive Systems, Krakow, Poland, 15-18 February 2010* (Leonard Barolli, Fatos Xhafa, Salvatore Vitabile e Hui-Huang Hsu, Eds.). IEEE Computer Society. pp. 774–778.
- Ali, Q. e S. Montenegro (2016). Explicit model following distributed control scheme for formation flying of mini uavs. *IEEE Access* **4**, 397–406.
- Allgöwer, Frank, Rolf Findeisen e Zoltan K. Nagy (2004). Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Engrs.* **35**(3), 299–315.
- Alvarez, Teresa e César de Prada (1997). Handling infeasibilities in predictive control. *Computers & Chemical Engineering* **21**, S577 – S582.
- Antonelli, Gianluca, Filippo Arrichiello, Stefano Chiaverini e Koustubh J. Rao (2006). Preliminary experiments of formation control using the null-space-based behavioral control. *Proceedings 8th IFAC Symposium on Robot Control, Bologna* p. 6.
- Arai, T., E. Pagello e L.E. Parker (2002). Guest editorial: Advances in multirobot systems. *Robotics and Automation, IEEE Transactions on* **18**(5), 655–661.
- Balch, T. e R.C. Arkin (1998). Behavior-based formation control for multirobot teams. *Robotics and Automation, IEEE Transactions on* **14**(6), 926–939.
- Barraquand, J. e J. C. Latombe (1989). On nonholonomic mobile robots and optimal maneuvering. Em: *Proceedings. IEEE International Symposium on Intelligent Control 1989*. pp. 340–347.
- Baskar, L. D., B. De Schutter, J. Hellendoorn e Z. Papp (2011). Traffic control and intelligent vehicle highway systems: a survey. *IET Intelligent Transport Systems* **5**(1), 38–52.

- Bemporad, Alberto e Manfred Morari (1999). *Robust model predictive control: A survey*. pp. 207–226. Springer London. London.
- Brockett, Roger W. (1983). Asymptotic stability and feedback stabilization. Em: *Differential Geometric Control Theory* (R. S. Millman R. W. Brockett e H. J. Sussmann, Eds.). pp. 181–191. Birkhauser. Boston.
- Budden, D. M., P. Wang, O. Obst e M. Prokopenko (2015). Robocup simulation leagues: Enabling replicable and robust investigation of complex robotic systems. *IEEE Robotics Automation Magazine* **22**(3), 140–146.
- Burns, R., C.A. McLaughlin, J. Leitner e M. Martin (2000). Techsat 21: formation design, control, and simulation. Em: *Aerospace Conference Proceedings, 2000 IEEE*. Vol. 7. pp. 19–25 vol.7.
- Camacho, E.F. e C. Bordons (2004). *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer London.
- Camacho, E.F. e C.B. Alba (2013). *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer London.
- Canudas de Wit, C., H. Khennouf, C. Samson e O.J. Sordalen (1993). Nonlinear control design for mobile robots. Em: *Recent Trends in Mobile Robots* (Y. F. Zheng, Ed.). World Scientific Series in Robotics and Automated Systems, Vol. 11. World Scientific.
- Cao, Yi (2005). A formulation of nonlinear model predictive control using automatic differentiation. *Journal of Process Control* **15**(8), 851 – 858.
- Casbeer, David W., Derek B. Kingston, Al W. Beard, Timothy W. McLain, Sai ming Li e Raman Mehra (2006). Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Sciences* **37**, 360.
- Celeste, W. C., T. F. B. Filho e M. S. Filho (2008). Controle de seguimento de caminho com sinais de comando limitados aplicado a robôs móveis unicyclos. Em: *Anais do Congresso Brasileiro de Automática*.
- Chaumette, F. e S. Hutchinson (2006). Visual servo control. i. basic approaches. *Robotics Automation Magazine, IEEE* **13**(4), 82–90.
- Chaumette, F. e S. Hutchinson (2007). Visual servo control. ii. advanced approaches [tutorial]. *Robotics Automation Magazine, IEEE* **14**(1), 109–118.
- Chen, Chieh-Li, Lin Sy-Hong e Cha’o-Kuang Chen (1996). Application of taylor transformation to nonlinear predictive control problem. *Applied Mathematical Modelling* **20**(9), 699 – 710.

- Chen, Yang Quan e Zhongmin Wang (2005). Formation control: a review and a new consideration. Em: *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*. pp. 3181–3186.
- Conceição, A. G. S., T. T. Ribeiro e A. P. Moreira (2006). Architecture of cooperation for multi-robot systems. *1st IFAC Workshop on Multivehicle Systems*.
- Conceição, André Gustavo Scolari (2007). Controlo e cooperação de robôs móveis autónomos omnidireccionais. Tese de doutoramento. Faculdade de Engenharia. Universidade do Porto.
- Corke, P. (2011). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Tracts in Advanced Robotics. Springer.
- Coulaud, J.-B., G. Campion, G. Bastin e M. De Wan (2006). Stability analysis of a vision-based control design for an autonomous mobile robot. *Robotics, IEEE Transactions on* **22**(5), 1062–1069.
- Cutler, C.R. e B.C. Ramake (1980). Dynamic matrix control- a computer control algorithm. Em: *Automatic Control Conference, San Francisco*. pp. 774–778.
- Das, A.K., R. Fierro, V. Kumar, J.P. Ostrowski, J. Spletzer e C.J. Taylor (2002). A vision-based formation control framework. *Robotics and Automation, IEEE Transactions on* **18**(5), 813–825.
- de Oliveira, Nuno e Lorenz T Biegler (1994). Constraint handing and stability properties of model-predictive control. *AIChE journal* **40**(7), 1138–1155.
- DFR (2014). *DFRobotShop Rover USER GUIDE Rev 2.4*.
- Divelbiss, A. W. e J. T. Wen (1997). Trajectory tracking control of a car-trailer system. *IEEE Transactions on Control Systems Technology* **5**(3), 269–278.
- Dunbar, William B. e Richard M. Murray (2005). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica* **42**(4), 549–558.
- Egerstedt, M. e Xiaoming Hu (2001). Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation* **17**(6), 947–951.
- El-Ferik, S., B. A. Siddiqui e F. L. Lewis (2013). Distributed nonlinear mpc formation control with limited bandwidth. Em: *2013 American Control Conference*. pp. 6388–6393.
- El-Ferik, S., B. A. Siddiqui e F. L. Lewis (2016). Distributed nonlinear mpc of multi-agent systems with data compression and random delays. *IEEE Transactions on Automatic Control* **61**(3), 817–822.

- EL-Zaher, Madeleine, Baudouin Dafflon, Franck Gechter e Jean-Michel Contet (2012). Vehicle platoon control with multi-configuration ability. *Procedia Computer Science* **9**, 1503 – 1512.
- Eoh, Gyuhoo, Jeong S. Choi e Beom H. Lee (2013). Faulty robot rescue by multi-robot cooperation. *Robotica* **31**, 1239–1249.
- Fierro, R. e F. L. Lewis (1996). Practical point stabilization of a nonholonomic mobile robot using neural networks. Em: *Proceedings of 35th IEEE Conference on Decision and Control*. Vol. 2. pp. 1722–1727 vol.2.
- Freund, E. e R. Mayr (1997). Nonlinear path control in automated vehicle guidance. *Robotics and Automation, IEEE Transactions on* **13**(1), 49 –60.
- Frezza, R., S. Soatto e G. Picci (1997). Visual path following by recursive spline updating. Em: *Proceedings of the 36th IEEE Conference on Decision and Control*. Vol. 2. pp. 1130–1134 vol.2.
- Fukushima, H., K. Kon e F. Matsuno (2013). Model predictive formation control using branch-and-bound compatible with collision avoidance problems. *IEEE Transactions on Robotics* **29**(5), 1308–1317.
- Gazi, Veysel (2003). Swarm aggregations using artificial potentials and sliding mode control. *IEEE Transactions on Robotics* **21**, 1208–1214.
- Ghomman, J., M. Saad e F. Mnif (2008). Formation path following control of unicycle-type mobile robots. Em: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. pp. 1966–1972.
- Godsil, C.D. e G. Royle (2001). *Algebraic Graph Theory*. Graduate Texts in Mathematics. Springer.
- Grüne, L. e J. Pannek (2011). *Nonlinear Model Predictive Control: Theory and Algorithms*. Communications and Control Engineering. 1st ed. ed.. Springer.
- He, Y. Q. e J. D. Han (2012). Nonlinear model predictive control with regulable computational cost. *Asian Journal of Control* **14**(1), 300–307.
- Jia, Dong (2002). Min-max feedback model predictive control for distributed control with communication. Em: *American Control Conference, 2002. Proceedings of the 2002*. Vol. 6. pp. 4507 – 4512.
- Jiang, Zhong-Ping e H. Nijmeijer (1999). A recursive technique for tracking control of nonholonomic systems in chained form. *Automatic Control, IEEE Transactions on* **44**(2), 265 –279.

- Kanjanawanishkul, K. (2009). Formation control of omnidirectional mobile robots using distributed model predictive control. Em: *2009 Second International Conference on Robot Communication and Coordination*. pp. 1–7.
- Kanjanawanishkul, K. e A. Zell (2008a). Distributed model predictive control for coordinated path following control of omnidirectional mobile robots. Em: *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*. pp. 3120–3125.
- Kanjanawanishkul, K. e A. Zell (2008b). A model-predictive approach to formation control of omnidirectional mobile robots. Em: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. pp. 2771–2776.
- Kanjanawanishkul, K. e A. Zell (2008c). A model-predictive approach to formation control of omnidirectional mobile robots. Em: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 2771–2776.
- Kanjanawanishkul, Kiattisin e Andreas Zell (2009). Path following for an omnidirectional mobile robot based on model predictive control. Em: *ICRA*. pp. 3341–3346.
- Khalil, H.K. (2002). *Nonlinear Systems*. Pearson Education. Prentice Hall.
- Kim, Suseong, Seungwon Choi e H.J. Kim (2013). Aerial manipulation using a quadrotor with a two dof robotic arm. Em: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. pp. 4990–4995.
- Kitano, H., Asada M., I. Noda e H. Matsuura (1998). Robocup: robot world cup. *IEEE Robotics Automation Magazine* **5**(3), 30–36.
- Kuhne, F., W. F. Lages e J. M. G. da Silva (2005). Point stabilization of mobile robots with nonlinear model predictive control. Em: *IEEE International Conference Mechatronics and Automation, 2005*. Vol. 3. pp. 1163–1168 Vol. 3.
- Larson, J., K. Liang e K. H. Johansson (2014). A distributed framework for coordinated heavy-duty vehicle platooning. *IEEE Transactions on Intelligent Transportation Systems* **16**, 419–429.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press. Cambridge, U.K. Available at <http://planning.cs.uiuc.edu/>.
- Lavretsky, E. (2002). F/a-18 autonomous formation flight control systems design. In aiaaGNC, pages AIAA 2002-4757.
- Lee, S. M. e H. Myung (2015). Receding horizon particle swarm optimisation-based formation control with collision avoidance for non-holonomic mobile robots. *IET Control Theory Applications* **9**(14), 2075–2083.

- Li, Sai-Ming, J.D. Boskovic, S. Seereeram, R. Prasanth, J. Amin, R.K. Mehra, R.W. Beard e T.W. McLain (2002). Autonomous hierarchical control of multiple unmanned combat air vehicles (ucavs). Em: *American Control Conference, 2002. Proceedings of the 2002*. Vol. 1. pp. 274–279 vol.1.
- Liang, Xiannuan e Yang Xiao (2010). Studying bio-inspired coalition formation of robots for detecting intrusions using game theory. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **40**(3), 683–693.
- Likhachev, M., M. Kaess e R. C. Arkin (2002). Learning behavioral parameterization using spatio-temporal case-based reasoning. Em: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*. Vol. 2. pp. 1282–1289.
- Lim, Heonyoung, Yeonsik Kang, Jongwon Kim e Changwhan Kim (2009a). Formation control of leader following unmanned ground vehicles using nonlinear model predictive control. Em: *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*. pp. 945–950.
- Lim, Heonyoung, Yeonsik Kang, Jongwon Kim e Changwhan Kim (2009b). Formation control of leader following unmanned ground vehicles using nonlinear model predictive control. Em: *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*. pp. 945–950.
- Liu, Z., W. Chen, J. Lu, H. Wang e J. Wang (2016). Formation control of mobile robots using distributed controller with sampled-data and communication delays. *IEEE Transactions on Control Systems Technology* **24**(6), 2125–2132.
- Löfberg, Johan (2012). Oops! i cannot do it again: Testing for recursive feasibility in {MPC}. *Automatica* **48**(3), 550 – 555.
- Mahony, R., V. Kumar e P. Corke (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *Robotics Automation Magazine, IEEE* **19**(3), 20–32.
- Mansfield, Elizabeth Louise (2010). *A Practical Guide to the Invariant Calculus*. number 26 Em: *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press.
- MaxStream, Inc (2006). *Product Manual, v1.xAx - 802.15.4, Protocol For OEM RF Module Part Numbers: XB24-...-001, XBP24-...-001*. 1st ed.. Digi international.
- Mayne, D. Q., J. B. Rawlings, C. V. Rao e P. O. M. Scokaert (2000). Constrained model predictive control: Stability and optimality. *AUTOMATICA* **36**, 789–814.

- Mehrez, M. W., G. K. I. Mann e R. G. Gosine (2014). Formation stabilization of nonholonomic robots using nonlinear model predictive control. Em: *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*. pp. 1–6.
- Miyata, N., J. Ota, T. Arai e H. Asama (2002). Cooperative transport by multiple mobile robots in unknown static environments associated with real-time task assignment. *IEEE Transactions on Robotics and Automation* **18**(5), 769–780.
- Monterey, Bay Aquarium Research Institute (2006). Autonomous ocean sampling network. <http://www.mbari.org/aosn>.
- Murray, Richard M. (2007). Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control* **129**(5), 571–583.
- Nascimento, T. P., F. A. Fontes, A. P. Moreira e A. G. S. Conceição (2011). Leader following formation control for omnidirectional mobile robots: The target chasing problem. *8th International Conference on Informatics in Control, Automation and Robotics (ICINCO)* p. 10.
- OH, Air Force Wright Aeronautical Labs Wright-Patterson AFB (1982). Background information and user guide for mil-f-8785c, military specification - flying qualities of piloted airplanes. Technical report.
- Oh, Kwang-Kyo, Myoung-Chul Park e Hyo-Sung Ahn (2015). A survey of multi-agent formation control. *Automatica* **53**, 424 – 440.
- openPicus (2012). *FLYPORT Wi-Fi Datasheet*. 9th ed.. www.openpicus.com.
- Path, California (2006). California partners for advanced transit and highways. <http://www.path.berkeley.edu>.
- Priscoli, F. D., A. Isidori, L. Marconi e A. Pietrabissa (2015). Leader-following coordination of nonlinear agents under time-varying communication topologies. *IEEE Transactions on Control of Network Systems* **2**(4), 393–405.
- Propoi, A. I (1963). Use of lp methods for synthesizing sampled data automatic systems. Em: *Automatic Remote Control*. p. 24.
- Qiu, Z., L. Xie e Y. Hong (2016). Quantized leaderless and leader-following consensus of high-order multi-agent systems with limited data rate. *IEEE Transactions on Automatic Control* **61**(9), 2432–2447.
- Rawlings, J. B. e K. R. Muske (1993). The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control* **38**(10), 1512–1516.

- Ren, W. (2007). Consensus strategies for cooperative control of vehicle formations. *IET Control Theory Applications* **1**(2), 505–512.
- Ren, Wei e R.W. Beard (2003). A decentralized scheme for spacecraft formation flying via the virtual structure approach. Em: *American Control Conference, 2003. Proceedings of the 2003*. Vol. 2. pp. 1746–1751.
- Ribeiro, T. T., Augusto L. da Costa e A. G. S Conceição (2014). Controle de formação baseado em nmpc para quadricopteros.. Em: *Anais do XX CBA, Congresso Brasileiro de Automática, 2014*. pp. 1490–1497.
- Ribeiro, T. T., J. T. Santos, A. S. Conceição e A. L. Costa (2011a). Sistema microprocessado de tempo real aplicado à robótica móvel. *Revista Jr de Iniciação Científica em Ciências Exatas e Engenharia* **2**, 23–31.
- Ribeiro, T. T., R. O. Fernandes e A. G. S. Conceição (2016a). Garantia de estabilidade e factibilidade de controladores nmpc para o seguimento de caminhos visuais.. Em: *Anais do XXI CBA, Congresso Brasileiro de Automática, 2016*. pp. 338–343.
- Ribeiro, T. T., R. O. Fernandes e A. G. S. Conceição (2016b). Seguimento de caminhos visuais baseado em nmpc para robôs diferenciais.. Em: *Anais do XXI CBA, Congresso Brasileiro de Automática, 2016*. pp. 332–337.
- Ribeiro, Tiago T., André G.S. Conceição, Inkyu Sa e Peter Corke (2015a). Nonlinear model predictive formation control for quadcopters*. *IFAC-PapersOnLine* **48**(19), 39 – 44. 11th {IFAC} Symposium on Robot Control {SYROCO} 2015Salvador, Brazil, 2628 August 2015.
- Ribeiro, Tiago T., Augusto L. Costa e André G. S. Conceição (2015b). Distributed constrained nmpc with infeasibility handling applied to formation control of nonholonomic vehicles. *Journal of Control, Automation and Electrical Systems* **26**(6), 599–608.
- Ribeiro, Tiago Trindade (2013). Controle de Formação em Tempo Real Baseado em NMPC para Robôs Móveis Omnidirecionais. Dissertação de mestrado. Universidade Federal da Bahia.
- Ribeiro, Tiago Trindade e André Gustavo Scolari Conceição (2012). Controladores preditivos baseados em modelos não lineares para o seguimento de caminho coordenado entre robôs moveis omnidirecionais. Em: *Anais do XIX CBA, Congresso Brasileiro de Automática, 2012*. pp. x–x.
- Ribeiro, Tiago Trindade, Jessivaldo Santos, André Gustavo Scolari Conceição e Augusto Loureiro da Costa (2013). Wireless real-time system for omnidirectional mobile robots control. Em: *Anais do XI SBAI, Simpósio Brasileiro de Automação Inteligente, 2013*. pp. x–x.

- Ribeiro, Tiago Trindade, Jovelino Torres dos Santos, Jessivaldo Santos, André Gustavo Scolari Conceição e Augusto Loureiro da Costa (2011*b*). Real-time microprocessed system applied to mobile robots control. Em: *IEEE IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, LARC 2011, Bogota, Colombia, October 1-4, 2011*. pp. 1–6.
- Richalet, J., A. Rault, J. L. Testud e J. Papon (1978). Paper: Model predictive heuristic control. *Automatica* **14**(5), 413–428.
- Richards, A. e J. How (2004). A decentralized algorithm for robust constrained model predictive control. Em: *American Control Conference, 2004. Proceedings of the 2004*. Vol. 5. pp. 4261–4266 vol.5.
- Sa, Inkyu, Stefan Hrabar e Peter Corke (2013). Outdoor flight testing of a pole inspection uav incorporating high-speed vision. Em: *The International Conference on Field and Service Robotics*.
- Samson, C. (1995). Control of chained systems application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control* **40**(1), 64–77.
- Samson, Claude e Karim Ait-Abderrahim (1991). Mobile robot control part 1: Feedback control of a non-holonomic mobile robots. Technical Report 1281. INRIA.
- Scokaert, Pierre O M. (1994). Constrained Predictive Control. Phd thesis. Univ. Oxford, UK.
- Scokaert, P.O.M., D.Q. Mayne e J.B. Rawlings (1999). Suboptimal model predictive control (feasibility implies stability). *Automatic Control, IEEE Transactions on* **44**(3), 648–654.
- Shin, J. e H. J. Kim (2009). Nonlinear model predictive formation flight. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **39**(5), 1116–1125.
- Siegwart, R. e I. R. Nourbakhshi (2004). *Introduction to Autonomous Mobile Robots*. 1 ed.. Ronald C. Arkin. London.
- Soetanto, D., L. Lapiere e A. Pascoal (2003). Adaptive, non-singular path-following control of dynamic wheeled robots. Em: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*. Vol. 2. pp. 1765–1770 Vol.2.
- Spellucci, P. (1998). An sqp method for general nonlinear programs using only equality constrained subproblems. Em: *Mathematical Programming*. Vol. 82. pp. 413 – 448.
- Spellucci, P. (1999). Donlp2 short users guide.

- Spong, M. W., S. Hutchinson e M. Vidyasagar (2006). *Robot Modeling and Control*. 1st ed.. JOHN WILEY and SONS, INC.. New York.
- Stuckler, J., D. Holz e S. Behnke (2012). Robocup@home: Demonstrating everyday manipulation skills in robocup@home. *IEEE Robotics Automation Magazine* **19**(2), 34–42.
- Szwarcfiter, Jayme Luiz (1984). *Grafos e Algoritmos Computacionais*. Campus / Elsevier.
- Tan, Kar-Han e M.A. Lewis (1996). Virtual structures for high-precision cooperative mobile robotic control. Em: *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*. Vol. 1. pp. 132–139 vol.1.
- Tanenbaum, Andrew S. (2002). *Computer networks (4. ed.)*. Prentice Hall.
- Tomlin, C., G.J. Pappas e S. Sastry (1998). Conflict resolution for air traffic management: a study in multiagent hybrid systems. *Automatic Control, IEEE Transactions on* **43**(4), 509–521.
- Trindade Ribeiro, T., R. Ferrari, J. Santos e A.G.S. Conceicao (2013). Formation control of mobile robots using decentralized nonlinear model predictive control. Em: *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on*. pp. 32–37.
- Vada, Jostein, Olav Slupphaug, Tor A. Johansen e Bjarne A. Foss (2001). Linear {MPC} with optimal prioritized infeasibility handling: application, computational issues and stability. *Automatica* **37**(11), 1835 – 1843.
- Šiljak, Dragoslav D. (1991). *Decentralized control of complex systems*. Mathematics in Science and Engineering. Elsevier Science.
- Walsh, G.C., Hong Ye e L.G. Bushnell (2002). Stability analysis of networked control systems. *Control Systems Technology, IEEE Transactions on* **10**(3), 438 –446.
- Wang, Liuping (2001). Continuous time model predictive control design using orthonormal functions. *International Journal of Control* **74**(16), 1588–1600.
- Wang, Liuping (2009). *Model Predictive Control System Design and Implementation Using MATLAB*. 1st ed.. Springer Publishing Company, Incorporated.
- Wang, P. K. C. (1989). Navigation strategies for multiple autonomous mobile robots moving in formation. Em: *Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS '89. Proceedings., IEEE/RSJ International Workshop on*. pp. 486–493.

- Weisbin, C. e D. Perillard (1991). R & d profile jet propulsion laboratory robotic facilities and associated research. *Robotica* **9**, 7–21.
- Xiao, H., Z. Li e C. L. Philip Chen (2016). Formation control of leader #x2013;follower mobile robots #x2013; systems using model predictive control based on neural-dynamic optimization. *IEEE Transactions on Industrial Electronics* **63**(9), 5752–5762.
- Ye, Yinyu (1987). Interior Algorithms for Linear, Quadratic, and Linearly Constrained Non-Linear Programming. Tese de Doutorado. Department of ESS, Stanford University.
- Ye, Yinyu (1990). *SOLNP Users Guide - A Nonlinear Optimization Program to Matlab*. 1st ed.. University of Iowa. Iowa City, Iowa 52242.
- Zafriou, E. e H. W. Chiou (1993). Output constraint softening for siso model predictive control. Em: *1993 American Control Conference*. pp. 372–376.
- Zhao, Li e Jian Sun (2013). Simulation framework for vehicle platooning and car-following behaviors under connected-vehicle environment. *Procedia - Social and Behavioral Sciences* **96**, 914 – 924.
- Zheng, J. e M. J. Lee (2004). A comprehensive performance study of ieee 802.15.4. *IEEE Press Book*.
- ZigBee, Alliance (2011). *ZigBee Specification*. 1st ed.. ZigBee alliance. <http://www.zigbee.org/>.