

**JOSUÉ JUNIOR GUIMARÃES RAMOS**

**CONTRIBUIÇÃO AO DESENVOLVIMENTO DE  
DIRIGÍVEIS ROBÓTICOS**

**FLORIANÓPOLIS  
2002**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA**  
**ELÉTRICA**

**CONTRIBUIÇÃO AO DESENVOLVIMENTO DE**  
**DIRIGÍVEIS ROBÓTICOS**

Tese submetida à  
Universidade Federal de Santa Catarina  
como parte dos requisitos para a  
obtenção do grau de Doutor em Engenharia Elétrica.

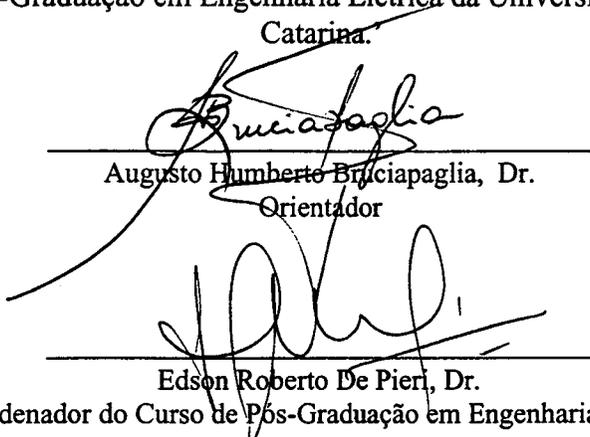
**JOSUÉ JUNIOR GUIMARÃES RAMOS**

Florianópolis, Março de 2002.

# CONTRIBUIÇÃO AO DESENVOLVIMENTO DE DIRIGÍVEIS ROBÓTICOS

Josué Junior Guimarães Ramos

‘Esta Tese foi julgada adequada para obtenção do Título de Doutor em Engenharia Elétrica, Área de Concentração em Sistemas de Informação, e aprovada em sua forma final pelo Curso de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’



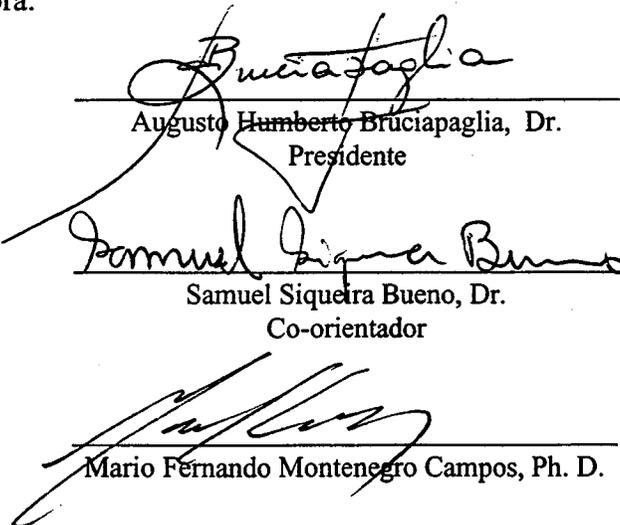
---

Augusto Humberto Brúciapaglia, Dr.  
Orientador

---

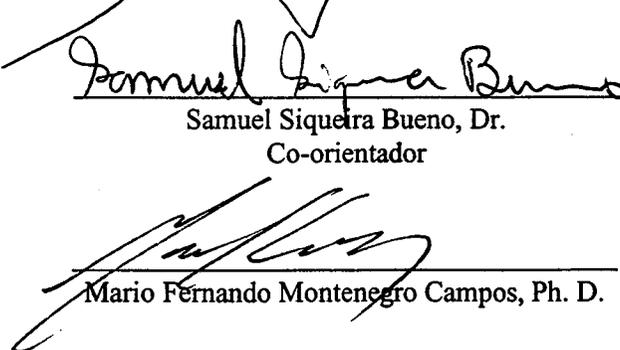
Edson Roberto De Pieri, Dr.  
Coordenador do Curso de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:



---

Augusto Humberto Brúciapaglia, Dr.  
Presidente

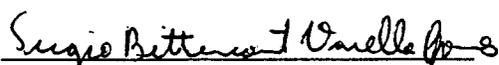


---

Samuel Siqueira Bueno, Dr.  
Co-orientador

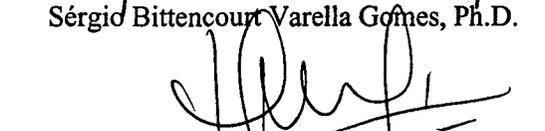
---

Mario Fernando Montenegro Campos, Ph. D.



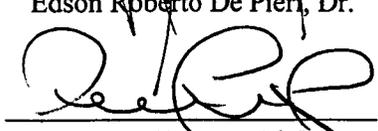
---

Sérgio Bittencourt Varella Gomes, Ph.D.



---

Edson Roberto De Pieri, Dr.



---

Werner Kraus Jr, Ph.D.

## AGRADECIMENTOS

Dedico este trabalho a Oswaldo e Terezinha, meus pais, por todo o apoio e carinho dados ao longo de toda a minha vida. Dedico também a Simony e Clara por todo o apoio durante estes anos.

Dedico também este trabalho ao povo brasileiro, principal financiador dos trabalhos realizados no projeto AURORA.

Agradeço ao meu irmão Osvaldo Tadeu, por estar sempre ao meu lado me incentivando.

Agradeço a toda a equipe do LRVC do CenPRA e do DAS / UFSC pelo constante incentivo e inspiração. Em especial, agradeço ao Augusto, Samuel e Reid pelo ótimo trabalho de orientação. Agradeço ao Marcel pelo constante apoio durante a realização do trabalho, ao Ely, José Raul e Sérgio pela base que me supriram para a realização dos trabalhos. Também agradeço ao Silvio, Mirisola e Bruno pela constante disponibilidade e apoio.

Agradeço em especial ao Edeneziano, Koyama e Jorge pelo apoio na revisão do texto.

Lembro também o nome de todos que participaram do Projeto AURORA, ou contribuíram de alguma forma com o projeto e com os quais eu tive o prazer de trabalhar:

<i>Alberto Elfes</i>	<i>Gustavo de Sousa Pavani</i>	<i>Marcel Bergerman</i>
<i>Bruno Guedes Faria</i>	<i>João Paulo G. Batistela</i>	<i>Odir Spada Júnior</i>
<i>Cleyner Soares Pereira</i>	<i>José Reginaldo H. de Carvalho</i>	<i>Omar Esteves Duarte Filho</i>
<i>Conrad Tadashi Fujiwara</i>	<i>José Raul Azinheira</i>	<i>Patrick Rives</i>
<i>Ely Carneiro de Paiva</i>	<i>Hichiro Yamaguchi</i>	<i>Ricardo da Rocha Frazzato</i>
<i>Fábio A.G. F. dos Santos</i>	<i>Kiyoshi Asanuma</i>	<i>Roberto Tavares Filho</i>
<i>Felipe de Alvarenga Leite</i>	<i>Luiz Gustavo Bizarro Mirisola</i>	<i>Rodrigo Paniago Peixoto</i>
<i>Gabriel Cardoso Martins</i>	<i>Luiz Gustavo C. do Nascimento</i>	<i>Samuel Siqueira Bueno</i>
<i>Gabriel Esteves Marques Pinto</i>		<i>Silvio Mano Maeta</i>

A realização deste projeto teve o apoio financeiro do CenPRA, CNPq e FAPESP.

Resumo da Tese apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Doutor em Engenharia Elétrica.

## **CONTRIBUIÇÃO AO DESENVOLVIMENTO DE DIRIGÍVEIS ROBÓTICOS**

**Josué Junior Guimarães Ramos**

Março/2002

Orientador: Augusto Humberto Brucciapaglia, Dr.

Co-orientador: Samuel Siqueira Bueno, Dr.

Área de Concentração: Sistemas de Informação.

Palavras-chave: dirigível robótico, *software* e *hardware* para controle e navegação, simulação, arquitetura de *software* robótico.

Número de Páginas: 292

Este trabalho aborda o desenvolvimento e a implementação de um protótipo de dirigível robótico não-tripulado com capacidade de realizar vôos autônomos reais seguindo trajetórias definidas por pontos de passagem. Para contemplar este objetivo: i) realizou-se modificações estruturais na mecânica e aerodinâmica do dirigível utilizado, adequando-o ao uso como veículo robótico; ii) estabeleceram-se infra-estruturas, embarcada e em terra, de componentes de *software* e de *hardware*, envolvendo sensores, atuadores, processadores e *software* tempo real; iii) introduziu-se aprimoramentos em um modelo dinâmico do dirigível considerando notadamente os aspectos de força de propulsão e influência da dinâmica do vento; com base neste novo modelo, foram desenvolvidos ambientes de simulação em SIMULINK/MATLAB e em Java/VRML e servidores de simulação; iv) implementou-se um ambiente de suporte ao desenvolvimento e à operação do dirigível robótico; v) concebeu-se e implementou-se um sistema de controle e navegação, compreendendo algoritmos de seguimento de trajetória e de perfil de altitude entre pontos de passagem; este sistema foi analisado e seus parâmetros de sintonia ajustados tanto em simulação quanto em vôos reais, e, finalmente; vi) estabeleceu-se um protótipo de arquitetura de *software* robótico para o dirigível testado apenas em simulação, contemplando aspectos deliberativos e reativos de uma missão robótica completa.

Abstract of Thesis presented to UFSC as a partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

## **CONTRIBUTION TO THE DEVELOPMENT OF ROBOTIC AIRSHIPS**

**Josué Junior Guimarães Ramos**

March/2002

Advisor: Augusto Humberto Bruciapaglia, Dr.

Co-advisor: Samuel Siqueira Bueno, Dr.

Area of Concentration: Information Systems

Keywords: Airship, robotics, software, hardware, control, navigation, simulation.

Number of Pages: 292

This work focuses on the development and implementation of a prototype of an unmanned robotic airship, able to execute autonomous real flights with trajectory tracking through way-points. To accomplish this task: i) the airship's mechanical and aerodynamic components were improved; ii) it was established onboard and ground-based infrastructures of hardware and software components, including sensors, actuators, processors and real time software; iii) improvements were introduced in the airship dynamic model considering propulsion forces and wind dynamics influence; based on this new model, airship simulators in SIMULINK/MATBLAB and Java/VRML environments as well as simulator servers, were developed; iv) a software environment to support the development and operation of the robotic airship was built; v) a control and navigation system was implemented, including trajectory and altitude profile tracking through way-points; this system was analyzed and its tuning parameters were adjusted both in simulation and in real flight, and, finally; vi) a prototype of a robotic software architecture was established in simulation, dealing with deliberative and reactive aspects of a complete robotic mission.

# ÍNDICE

<b>ÍNDICE DE FIGURAS</b>	<b>XIX</b>
<b>LISTA DE SÍMBOLOS</b>	<b>XXIII</b>
<b>1. INTRODUÇÃO</b>	<b>1</b>
1.1 Objetivos	1
1.2 Motivação	2
1.3 Dirigíveis	4
1.3.1 Aplicações e Tendências	4
1.3.2 Dirigíveis não-Tripulados	8
1.4 Dirigíveis para Missões de Inspeção Aérea e Monitoramento Ambiental	11
1.5 Projeto AURORA	12
1.6 Contribuições desta Tese	13
1.7 Organização do Texto	15
<b>2. ESPECIFICAÇÃO DE REQUISITOS</b>	<b>17</b>
2.1 Introdução	17
2.2 Descrição dos Requisitos para um Dirigível Robótico Autônomo	17
2.2.1 Componentes Básicos de um Dirigível Robótico.	18
2.2.2 Capacidades do Dirigível Robótico	19
2.2.3 Recursos de Suporte	22
2.3 Especificação de Requisitos para esta Tese	26
2.4 Projeto Conceitual	28
2.4.1 Mapeamento do Modelo Conceitual para um Dirigível Robótico	30
2.5 Arquitetura Funcional	31
2.6 Conclusões	32
<b>3. MODELO MATEMÁTICO DO DIRIGÍVEL</b>	<b>35</b>
3.1 Introdução	35
3.2 Princípio de Operação dos Dirigíveis	36
3.2.1 Forças Aerostáticas	36
3.2.2 Forças Aerodinâmicas	37
3.2.3 Forças Dinâmicas	38
3.2.4 Forças de Propulsão	38
3.3 Sistemas de Coordenadas e Principais Atuadores no Dirigível	39

3.3.1	Sistemas de Coordenadas	39
3.3.2	Atuadores no Dirigível	40
<b>3.4</b>	<b>Os Termos do Modelo Dinâmico</b>	<b>41</b>
3.4.1	Vetor Velocidade	42
3.4.2	Matriz de Massa	42
3.4.3	Vetor de Forças Dinâmicas	44
3.4.4	Efeito do Vento	44
3.4.5	Vetor de Forças Aerodinâmicas	45
3.4.6	Vetor de Gravidade e Flutuação	47
3.4.7	Modelo da Força de Propulsão	47
<b>3.5</b>	<b>Particularização do Modelo Matemático para o Dirigível AS800</b>	<b>49</b>
<b>3.6</b>	<b>Aprimoramentos no Modelo Matemático do Dirigível AS800</b>	<b>50</b>
3.6.1	Extrapolação dos Coeficientes Aerodinâmicos para Ângulos Maiores que 30°	50
3.6.2	Inclusão do Efeito do Fluxo de Ar Induzido pelos Motores nas Superfícies Aerodinâmicas	52
3.6.3	Inclusão de um Modelo para a Turbulência Atmosférica	53
<b>3.7</b>	<b>Avaliação do Modelo Matemático do Dirigível</b>	<b>53</b>
<b>3.8</b>	<b>Conclusões</b>	<b>55</b>
<b>4.</b>	<b>ALGORITMOS DE CONTROLE</b>	<b>57</b>
<b>4.1</b>	<b>Introdução</b>	<b>57</b>
<b>4.2</b>	<b>Controle Automático para Dirigíveis</b>	<b>58</b>
4.2.1	Técnicas de Navegação e Controle Automático em Outros Veículos Aéreos	58
4.2.2	Publicações sobre Controle de Dirigíveis	60
<b>4.3</b>	<b>Controle e Seguimento de Trajetória</b>	<b>65</b>
4.3.1	Controle de Trajetória em Veículos Aéreos e Manipuladores	65
4.3.2	Algoritmo para Controle de Trajetória do Dirigível	67
<b>4.4</b>	<b>Controle de altitude</b>	<b>71</b>
<b>4.5</b>	<b>Conclusões</b>	<b>74</b>
<b>5.</b>	<b>ARQUITETURA DE <i>SOFTWARE</i> ROBÓTICO PARA O DIRIGÍVEL</b>	<b>77</b>
<b>5.1</b>	<b>Introdução</b>	<b>77</b>
<b>5.2</b>	<b>Robôs</b>	<b>78</b>
<b>5.3</b>	<b>Arquiteturas de Software para Robôs Autônomos</b>	<b>80</b>
<b>5.4</b>	<b>Arquiteturas Híbridas Utilizando Três Camadas</b>	<b>83</b>
5.4.1	Estados	83
5.4.2	Implementação	83
5.4.3	Comportamentos	83
5.4.4	Executivo	83
5.4.5	Planejador	86
5.4.6	Arquitetura de Três Camadas para o Caso Multi-agentes	86
<b>5.5</b>	<b>Modelo Conceitual de Arquitetura para o Dirigível</b>	<b>86</b>

5.5.1	Agente em Terra - AT	87
5.5.2	Agente Embarcado - AE	89
<b>5.6</b>	<b>Conclusões</b>	<b>90</b>
<b>6.</b>	<b>SIMULADORES DE DIRIGÍVEL</b>	<b>91</b>
<b>6.1</b>	<b>Introdução</b>	<b>91</b>
<b>6.2</b>	<b>Simuladores de Veículos Aéreos</b>	<b>92</b>
<b>6.3</b>	<b>Primeira Versão do Simulador</b>	<b>93</b>
<b>6.4</b>	<b>Ambiente CACSD para Desenvolvimento de Sistemas de Controle</b>	<b>95</b>
<b>6.5</b>	<b>Servidor de Simulação de Dirigível em MATLAB</b>	<b>97</b>
<b>6.6</b>	<b>Simulador de Dirigíveis Baseado em Java / VRML</b>	<b>99</b>
6.6.1	Mundo VRML	101
6.6.2	Classe que Contém os Estados e Comandos do Simulador – classe interfaca	102
6.6.3	Classe de Controle do Dirigível – classe simvrml	102
6.6.4	Console do Dirigível – classe console	102
6.6.5	Execução do Modelo Matemático – classe runmod	104
6.6.6	Componentes Adicionais	105
6.6.7	O Simulador como uma Ferramenta de Treinamento	105
<b>6.7</b>	<b>Servidor de Simulação Baseado em Java</b>	<b>106</b>
<b>6.8</b>	<b>Conclusões</b>	<b>107</b>
<b>7.</b>	<b>AMBIENTE DE DESENVOLVIMENTO E OPERAÇÃO</b>	<b>109</b>
<b>7.1</b>	<b>Introdução</b>	<b>109</b>
<b>7.2</b>	<b>Ambientes de Suporte ao Desenvolvimento e Operação para Veículos Robóticos</b>	<b>110</b>
7.2.1	Ambiente para o Veículo da NPS	111
7.2.2	Outros Exemplos de Ambientes de Desenvolvimento e Operação	112
<b>7.3</b>	<b>Visão Geral do Ambiente</b>	<b>113</b>
<b>7.4</b>	<b>Estrutura do Ambiente de Desenvolvimento e Operação.</b>	<b>114</b>
<b>7.5</b>	<b>Protótipo do Ambiente de Desenvolvimento e Operação</b>	<b>116</b>
7.5.1	Infra-estrutura de Hardware e Comunicação do Protótipo	116
7.5.2	Componentes de Software do Protótipo	117
7.5.3	Estação Embarcada	118
7.5.4	Estação de Terra	119
7.5.5	Interface Homem - Máquina	121
7.5.6	Resultados do Desenvolvimento e Uso do Protótipo	124
<b>7.6</b>	<b>Evolução dos Componentes do Ambiente de Desenvolvimento e Operação</b>	<b>124</b>
7.6.1	Evolução dos Módulos de Kernel Embarcado e em Terra	125
7.6.2	Programação de Tarefas por Pontos de Passagem e Sintonização de Controladores	128
7.6.3	Visualizador 3D de vôos	132
<b>7.7</b>	<b>Conclusões</b>	<b>132</b>

<b>8.</b>	<b>IMPLEMENTAÇÃO DE ALGORITMOS DE CONTROLE</b>	<b>135</b>
8.1	Introdução	135
8.2	Diagrama Geral do Sistema de Controle Implementado	136
8.3	<b>Controle de Trajetória: Simulação</b>	<b>139</b>
8.3.1	Trajectoria de Referência	139
8.3.2	Influência dos Parâmetros de Sintonia dos Controladores	140
8.4	<b>Controle de Altitude: Simulação</b>	<b>149</b>
8.4.1	Ajuste dos Controladores sem a Presença de Vento.	150
8.4.2	Efeito da Execução Simultânea dos Algoritmos de Controle de Trajetória e de Altitude para Mudanças de Direção	153
8.4.3	Controle de Altitude e de Trajetória com Vento	155
8.4.4	Comportamento em Função da Flutuabilidade	158
8.4.5	Efeito da Conversão “+” para “x” nas Superfícies Aerodinâmicas de Atuação nos Controles de Altitude e de Trajetória	160
8.4.6	O Efeito da Turbulência	164
8.4.7	Resumo dos Resultados	165
8.5	<b>Controle de Trajetória: Resultados Obtidos em Vôo</b>	<b>167</b>
8.5.1	Trajectoria de Referência	168
8.5.2	Vôo do dia 7 de Julho de 2001	169
8.5.3	Vôos de 4 e 5 de Outubro de 2001	173
8.6	<b>Controle de Altitude: Resultados em Vôo</b>	<b>176</b>
8.6.1	Vôos de 24 e 30 de Julho de 2001	177
8.7	Conclusões	181
<b>9.</b>	<b>PROTÓTIPO DA ARQUITETURA DE SOFTWARE ROBÓTICO PARA O DIRIGÍVEL</b>	<b>185</b>
9.1	Introdução	185
9.2	Instanciação da Arquitetura de <i>Software</i> Robótico no Ambiente de Desenvolvimento e Operação do Dirigível	186
9.3	<b>Implementação da Arquitetura de Software Robótico</b>	<b>187</b>
9.3.1	Estrutura dos Executivos Escritos em TDL	187
9.3.2	Aspectos de Tempo Real	189
9.3.3	Diagrama Completo da Implementação da ASR	190
9.4	<b>Estudos de Caso com a Arquitetura de <i>Software</i> Robótico</b>	<b>191</b>
9.4.1	Testando o Planejador Principal em Ações Deliberativas e Reativas.	192
9.4.2	Testando o Planejador de Tratamento de Contingências	197
9.5	Conclusões	201
<b>10.</b>	<b>CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS</b>	<b>203</b>
10.1	Conclusões	203
10.2	Trabalhos Futuros	206

<b>ANEXO 1.RECURSOS DE SUPORTE PARA DESENVOLVIMENTO DO DIRIGIVEL ROBÓTICO</b>	<b>209</b>
<b>A1.1</b> <b>Introdução</b>	<b>209</b>
<b>A1.2</b> <b>O Dirigível</b>	<b>210</b>
A1.2.1    Envelope	210
A1.2.2    Superfícies Aerodinâmicas:	211
A1.2.3    Gôndola	214
A1.2.4    Rádio-controle, Transmissor e Receptor e Servo Atuadores	217
<b>A 1.3</b> <b>Detalhamento da Infra-estrutura de <i>Hardware</i> e Comunicação</b>	<b>218</b>
A1.3.1    Computadores e Elementos de Comunicação	220
A1.3.2    Atuadores	227
A1.3.3    Sensores	228
<b>A1.4</b> <b><i>Software</i></b>	<b>234</b>
A1.4.1    O Linux Tempo Real	234
A1.4.2    Arcabouço do Software	237
<b>A1.5</b> <b>Conclusões</b>	<b>239</b>
<b>ANEXO 2.APRIMORAMENTO DO SISTEMA DE PROPULSÃO DO DIRIGÍVEL AS800</b>	<b>241</b>
<b>ANEXO 3.DETALHAMENTO DA RELAÇÃO O ENTRE O TRABALHO DE TESE E OUTROS TRABALHOS</b>	<b>245</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>251</b>



## ÍNDICE DE FIGURAS

Figura 1.1: Dirigíveis disponíveis atualmente.	5
Figura 2.1: Conceito geral do dirigível robótico.	18
Figura 2.2: Relação entre componentes do sistema robótico e componentes de suporte.	28
Figura 2.3: Componentes associados às atividades de desenvolvimento/operação.	29
Figura 2.4: Ambiente de desenvolvimento e operação.	29
Figura 2.5: Componentes e inter-relacionamentos para um dirigível robótico.	30
Figura 2.6: Arquitetura funcional do dirigível robótico.	31
Figura 3.1: Operação dos balonetes num dirigível não rígido.	37
Figura 3.2: Componentes principais para a atuação num dirigível.	38
Figura 3.3: Sistemas de coordenadas do dirigível e inercial e principais variáveis associadas.	39
Figura 3.4: Cauda em “x” (visão posterior).	40
Figura 3.5: Ângulos de incidência $\alpha$ e $\beta$ .	46
Figura 3.6: Curva do coeficiente de sustentação ( $c_L$ ) em função de $\alpha$ , com $\beta=20^\circ$ , $\eta=\zeta=\xi=0$ .	51
Figura 3.7: Comandos aplicados ao leme de profundidade.	54
Figura 3.8: Resultados para mesma entrada de controle.	54
Figura 4.1: Diagrama de referência para controle direção em aviões.	66
Figura 4.2: A técnica ATGS.	67
Figura 4.3: Variáveis para seguimento de trajetória.	68
Figura 4.4: Algoritmo de controle de trajetória.	71
Figura 4.5: Diagrama de referência para controle de altitude.	72
Figura 4.6: Algoritmo de controle de altitude.	74
Figura 5.1: Árvore de tarefas.	85
Figura 5.2: Visão geral da ASR para o dirigível robótico	87
Figura 6.1: Saída típica de dados no ambiente de simulação básico MATLAB.	94
Figura 6.2: Bloco Principal do ambiente de CASD.	96
Figura 6.3: Diagrama em Simulink do modelo matemático do dirigível.	97
Figura 6.4: Blocos de comunicação por <i>socket</i> no Simulink.	98
Figura 6.5: Esquema do ensaio - modelo em Matlab/Simulink e painel em Java - comunicação via <i>socket</i> .	99
Figura 6.6: Interface do ambiente de simulação Java/VRML.	100
Figura 6.7: Estrutura e componentes principais do ambiente Java / VRML de simulação.	101
Figura 6.8: Componentes acionais de <i>software</i> e <i>hardware</i> para treinamento de pilotagem.	105
Figura 7.1: Componentes do mundo virtual da <i>Naval Post Graduate School</i> .	112
Figura 7.2: Estrutura conceitual do Ambiente de Desenvolvimento e Operação.	114
Figura 7.3: Diagrama Geral do Ambiente de Desenvolvimento e Operação.	115
Figura 7.4: Esquema de conexão dos equipamentos.	116
Figura 7.5: Componentes do protótipo do Ambiente de Desenvolvimento e Operação.	117
Figura 7.6: Componente gráfica da estação de terra com visualização de seus estados.	122

Figura 7.7: Painel de instrumentos e painel de comando via mouse.	122
Figura 7.8: Esquema de <i>playback</i> de vôos.	123
Figura 7.9: Esquema simplificado da estrutura para o software do módulo do <i>Kernel</i> Embarcado e módulo do <i>Kernel</i> em Terra.	126
Figura 7.10: Diagrama de classes dos módulos que compõem o sistema embarcado	127
Figura 7.11: Diagrama de seqüência da inicialização do sistema embarcado.	128
Figura 7.12: Interface gráfica do sistema de programação de tarefas.	129
Figura 7.13: Interface para ajuste de ganhos e definição de missão.	130
Figura 7.14: Visualização 3D dos estados do dirigível.	132
Figura 8.1: Diagrama geral do sistema de controle de trajetória e altitude.	136
Figura 8.2: Retas de referência e critério para transição entre retas.	139
Figura 8.3: Efeito da variação dos ganhos $K_{RT}$ e $K_{RRT}$ .	141
Figura 8.4: Obtenção de um conjunto de ganhos P e V.	142
Figura 8.5: Ajuste do ganho I com vento de 2,5 m/s	144
Figura 8.6: Efeito da variação do ganho $K_{RT}$ no comportamento em trajetória.	145
Figura 8.7: Trajetória completa com variação de vento.	146
Figura 8.8: Erro de trajetória com variação de vento.	147
Figura 8.9: Efeito da turbulência na resposta do controlador de trajetória	149
Figura 8.10: Ajustes dos parâmetros do controlador de altitude.	150
Figura 8.11: Efeito na altitude devido à mudança abrupta de trajetória	153
Figura 8.12: Efeito da variação da altitude com o vento (a 2,5m/s) e sua direção	155
Figura 8.13: Efeito da variação da altitude com o vento (a 4m/s) e sua direção	156
Figura 8.14: Efeito da variação da altitude com o vento (a 6m/s) e sua direção	157
Figura 8.15: Influência da variação da fluatibilidade.	159
Figura 8.16: Comportamento das varias variáveis para trajetória completa considerando a conversão de “+” para “x” para o conjunto de ganhos “A” da Tabela 8.2.	161
Figura 8.17: Comportamento das varias variáveis para trajetória completa considerando a conversão de “+” para “x” para o conjunto de ganhos “B” da Tabela 8.2.	162
Figura 8.18: Comportamento das varias variáveis para trajetória completa considerando a conversão de “+” para “x” para o conjunto de ganhos “C” da Tabela 8.2.	163
Figura 8.19: Trajetória completa considerando a conversão de “x” para “+” para o conjunto de ganhos A (a), B (b) e C (c) especificados na Tabela 8.2.	164
Figura 8.20: Efeito da turbulência na resposta do controlador de altitude.	165
Figura 8.21: Retas de referência para a trajetória e critério para transição entre retas.	169
Figura 8.22: Passos para ajuste do controlador de trajetória.	170
Figura 8.23: Melhor resultado dos ensaios do dia 7 de julho de 2001.	171
Figura 8.24: Melhor resultado de trajetória em 7 de julho.	171
Figura 8.25: Erro de trajetória “ $\delta$ ”, ângulo de guinada “ $\psi$ ” e leme de direção “ $\zeta$ ” para melhor trajetória.	172
Figura 8.26: Resultados obtidos nos vôos de 4 de Outubro.	174

Figura 8.27: Resultados obtidos nos ensaios de 5 de Outubro de 2001.	174
Figura 8.28: Melhor trajetória, correspondente ao Ensaio 7.	176
Figura 8.29: Trajetória resultante de controle simultâneo trajetória-altitude (Ensaio 1).	178
Figura 8.30: Variáveis do vôo de 24/7/2001.	178
Figura 8.31: Trajetória resultante de controle simultâneo trajetória-altitude (ensaio 4)	179
Figura 8.32: Vista geral do comportamento em trajetória e altitude.	180
Figura 8.33: Vista detalhada do comportamento em trajetória e altitude.	181
Figura 8.34: Piloto Bruno G. Faria e dirigível ao fundo em vôo automático.	183
Figura 9.1: Ambiente de <i>software</i> utilizado para o desenvolvimento da ASR.	186
Figura 9.2: Código simplificado do <i>software</i> de um Executivo	188
Figura 9.3: Estrutura do Protótipo Básico da Arquitetura de <i>Software</i> Robótico	191
Figura 9.4: A mudança no vento muda o plano de vôo.	195
Figura 9.5: Replanejamento devido a consumo de combustível.	196
Figura 9.6: Variáveis associadas a uma contingência de parada num motor numa reação dos comportamentos embarcados.	198
Figura 9.7: Detalhamento da posição e ângulo de guinada do dirigível.	200
Figura 9.8: Trajetória inicial e sua alteração como resultado de contingências tratadas pelo PTC.	200
Figura A1.1: O dirigível AS800-1.	210
Figura A1.2: O dirigível AS800-2.	211
Figura A1.3: Disposição geral dos componentes do dirigível.	212
Figura A1.4: Visão posterior do dirigível e montagens das superfícies aerodinâmicas.	212
Figura A1.5: Equivalência entre a atuação dos lemes em “+” e dos lemes em “x” (vista posterior).	212
Figura A1.6: Superfície aerodinâmica do AS800-1 (coberta) e sua extensão para uso no AS800-2.	214
Figura A1.7: Detalhe da disposição dos componentes embarcados na gôndola original.	214
Figura A1.8: Nova gôndola.	215
Figura A1.9: Motor e conjunto de hélices já utilizados.	216
Figura A1.10: Novo mecanismo de vetorização.	217
Figura A1.11: Conexão dos equipamentos.	218
Figura A1.12: Detalhamento da conexão dos componentes de <i>hardware</i> do dirigível.	219
Figura A1.13: Caixa contendo a eletrônica principal.	220
Figura A1.14: Componentes da estação de terra.	221
Figura A1.15: Sistema PC104 embarcado.	222
Figura A1.16: CAN4EPP da Zanthic.	223
Figura A1.17: Micro-controlador - MCC.	224
Figura A1.18: MiniModul 515C da Phytec.	225
Figura A1.19: Diagrama de operação do comutador de modo de operação e sua caixa.	225
Figura A1.20: Rádio Modem DGR-115.	226
Figura A1.21: Enlace de vídeo.	227
Figura A1.22: Interface com atuadores.	228

Figura A1.23: GPS embarcado e sua interface.	229
Figura A1.24: Bússola eletrônica e inclinômetro TCM2.	230
Figura A1.25: Sonda de vento ADM.	232
Figura A1.26: Interface com a sonda de vento.	233
Figura A1.27: Conexão via rede CAN dos sensores adicionais.	233
Figura A1.28: Esquemas do tacômetro e sensor de nível de combustível.	234
Figura A1.29: Arcabouço dos aplicativos RTLinux.	238
Figura A2.1: Corte da hélice e características dos diferentes perfis.	241
Figura A2.2: Exemplo de perfil de uma seção transversal de hélice levantado numa máquina de medição 3D	242
Figura A2.3: Curvas característica de $C_T$ e $C_p$ para hélices.	242
Figura A2.4: Ensaio da força de propulsão com o método do pêndulo.	243
Figura A3.1: Trabalhos associados ou trabalhos que se originaram de atividades realizadas neste trabalho de tese.	246

## LISTA DE SÍMBOLOS

A: vetor de forças e momentos aerodinâmicos,

AE: agente embarcado

AIAA: American Association for Aeronautics and Astronautics

AS800: dirigível utilizado como plataforma experimental utilizada nesta tese

ASR: arquitetura de *software* robótico

AT: agente em terra

AURORA: do acrônimo em Inglês para *Autonomous Unmanned Remote monitoring Robotic Airship*

CACSD: do Inglês *computer aided control system design*

CAD: do Inglês *computer aided design*

CAN: do Inglês *controller area network*

$c_D$ : coeficiente de arrasto

CE: comportamentos embarcados

CenPRA: Centro de Pesquisas Renato Archer

CG: centro de gravidade

$c_l$ : coeficiente de momento rolagem ou *rolling*

$c_L$ : coeficiente de sustentação

$c_m$ : coeficiente de momento de arfagem ou *pitching*

CMU: Carnegie Mellon University

$c_n$ : coeficiente de momento guinada ou *yawing*

CT: comportamentos em Terra

CV: centro de volume

$c_Y$ : coeficiente de forças laterais

DGPS: do Inglês *Differential Global Positioning System*

EE: Executivo Embarcado

ET: Executivo em terra

FBW: do inglês *flight by wire*

$F_d$ : o vetor de forças e momentos dinâmicos

G: vetor de forças e momentos gravitacionais

GPS: do Inglês *global positioning system*

IHM: interface homem-máquina

ISA: do Inglês *international standard atmosphere*

LTA: do Inglês *lighter-than-air*

M: matriz de massa,

MCD: matriz dos cosenos diretores

MKE: módulo do *kernel* embarcado

MKT: módulo *do kernel* em terra

N: Norte

P: vetor de forças e momentos de propulsão

p: velocidade de rolagem

PI: controlador proporcional-integral

PID: controlador proporcional-integral-derivativo

PIV: controlador proporcional-integral com realimentação de velocidade

PP: planejador principal

PTC: planejador de tratamento de contingências

q: velocidade de arfagem

r: velocidade de guinada

RTFifo: do Inglês *real time* FIFO

RTLinux: do Inglês Real time Linux

SAS: do Inglês *Stability Augmentation System*

SCD: sistema de coordenadas X, Y, Z, associado ao corpo dirigível

SCI: sistema de coordenadas inercial

TDL: do Inglês task description language

u: velocidade à frente no SCD

UAV: do Inglês *unmanned aerial vehicles*

UML: do Inglês *unified modeling language*

v: velocidade lateral no SCD

VRML: do Inglês *virtual reality modelling language*

w: velocidade vertical no SCD

x: vetor de velocidades lineares e angulares

YEZ-2A: Nome de dirigível

$\alpha$  (Alfa): ângulo de ataque

$\beta$  (Beta): ângulo de incidência lateral

$\delta$  (Delta): erro efetivo de trajetória

$\delta_1, \delta_2, \delta_3, \delta_4$  (Delta): deflexões individuais das superfícies aerodinâmicas

$\delta_a$ : erro previsto de trajetória

$\phi$  (Fi): ângulo de rolagem

$\eta$  (Eta) : deflexões do leme de profundidade

$\theta$  (teta): ângulo de arfagem

$\zeta$  (Zeta) : deflexão do leme de direção)

$\rho$  (Rhô): Símbolo para densidade

$\mu$  (Mu): ângulo de vetorização

$\psi$  (Psi): ângulo de guinada

$\xi$ , (Csi) : deflexão que introduz um efeito de rolagem (aileron)

## 1. INTRODUÇÃO

Os trabalhos desenvolvidos nesta tese envolvem duas Instituições: O Departamento de Automação e Sistemas da Universidade Federal de Santa Catarina (DAS / UFSC) e o Laboratório de Robótica e Visão Computacional do Centro de Pesquisas Renato Archer (LRVC / CenPRA), ex Centro Tecnológico para Informática (CTI).

O contexto dessa tese é o Projeto AURORA do acrônimo em Inglês para *Autonomous Unmanned Remote monitoring Robotic Airship* (dirigível autônomo robótico não tripulado para monitoramento remoto).

No AURORA tem-se como objetivo o desenvolvimento de estratégias de controle e de navegação autônomas para dirigíveis não tripulados a serem usados como plataforma aérea para a inspeção e monitoramento ambiental.

### 1.1 Objetivos

Os Objetivos desta tese são:

**Estabelecer a infra-estrutura de *software* e *hardware* e componentes de *software* para a operação autônoma de um dirigível robótico.**

Esta infra-estrutura é composta por sensores, processadores e alguma capacidade de autonomia para o dirigível, englobando tanto o veículo em si quanto a interface em terra com o operador. Ela constitui a base para, no âmbito do projeto AURORA (a ser descrito adiante), possibilitar a evolução gradual para soluções mais completas e complexas, do controle e navegação de dirigíveis, tendo como objetivo futuro do projeto a execução completamente autônoma de missões.

**Oferecer um ambiente de suporte ao desenvolvimento/operação do dirigível robótico.**

Este ambiente deve oferecer suporte às etapas de desenvolvimento e de operação do veículo robótico. Em termos de desenvolvimento, o que abrange também a integração, o teste e a validação de sistemas, ele deve suportar a evolução gradual: i) da concepção do *hardware* e *software* e, ii) do projeto de sistemas de controle, de estratégias de navegação e controle/execução de missão. Para tanto, o ambiente necessita prover meios para a

simulação de dirigíveis. Finalmente, o ambiente deve ser transparente, de forma que ao se substituir o simulador pelo veículo real, seja possível a operação do dirigível em vôos reais.

**Estabelecer um protótipo de arquitetura de software robótico para suportar aspectos deliberativos e reativos em uma missão<sup>1</sup>.**

Neste protótipo, os objetivos são de propor um modelo de arquitetura, coordenando as ações de deliberação e reação entre a estação embarcada e a estação de terra, e implementar um subconjunto desta arquitetura, validando-a em simulação. De forma similar ao exposto anteriormente, este protótipo constitui a base para a evolução gradual das políticas de deliberação e reação para as missões autônomas no AURORA .

**Operar o dirigível robótico em condições climáticas ideais, com a realização de movimentos através de pontos de passagem.**

A fim de validar tanto o ambiente de desenvolvimento e operação quanto as soluções de *hardware* e *software* concebidas e efetivamente implementadas no dirigível e em terra, o dirigível robótico deve ser capaz de estabilizar-se em condições ambientais favoráveis (i.e., com perturbações de pequena amplitude devidas ao vento) e realizar navegação por pontos pré-especificados.

## 1.2 Motivação

Veículos aéreos não tripulados possuem um grande potencial não explorado em tarefas de inspeção de grandes estruturas como linhas de transmissão ou oleodutos, prospecção mineral e arqueológica, planejamento urbano, monitoramento de tráfego, policiamento, retransmissão de sinais de rádio e vídeo, além de pesquisa e monitoramento ambiental, climatológica e de biodiversidade. No que diz respeito a esta última, na seqüência referida apenas como monitoramento ambiental, subtarefas incluem o sensoriamento e monitoramento de florestas, reservas ecológicas e parques nacionais, levantamentos de uso e ocupação do solo, estudos agropecuários, previsão de colheitas, medição da qualidade do ar e de níveis de poluição sobre centros urbanos e industriais, etc. A informação obtida nestes casos pode ser usada como base de estudos ecológicos e para o estabelecimento de políticas de desenvolvimento sustentado.

---

<sup>1</sup> Com relação a este objetivo cabe salientar o apoio do CNPq, através de uma bolsa de doutorado sanduíche, junto à equipe do Dr. Reid Simmons na CMU, USA, no período de março de 2000 a fevereiro de 2001.

Atualmente, muitas das tarefas acima descritas são realizadas a partir de sensoriamento remoto ou, mais precisamente, de dados obtidos através de sensores e câmeras instalados em balões, satélites ou aviões. Tais veículos, entretanto, possuem inúmeras desvantagens. Balões não são manobráveis e, portanto, não é possível ao seu usuário definir com rigor a área a ser sobrevoada. Imagens de satélites disponíveis para fins civis possuem baixa definição espacial e temporal, além de limitações nas faixas do espectro disponíveis. Levantamentos aéreos, embora permitam ao usuário controlar a área a ser sobrevoada, a resolução dos dados e as faixas do espectro utilizadas, requerem a garantia do conforto e da segurança da equipe de bordo, sendo portanto bastante dispendiosos.

Os avanços recentes nas áreas de sensores, interpretação sensorial e sistemas de controle e navegação têm levado ao desenvolvimento de vários sistemas robóticos autônomos não-tripulados com diferentes níveis de autonomia. Por autonomia entende-se a capacidade do sistema robótico efetuar sensoriamento sobre si mesmo e sobre o ambiente em que se encontra, interpretar a partir daí a sua situação, e tomar decisões face a um objetivo explicitado em termos de missão a ser cumprida. Progressos maiores têm sido registrados nas áreas de veículos não-tripulados semi-autônomos terrestres (UGVs, do inglês *unmanned ground vehicles*) e submarinos (UUVs, do inglês *unmanned underwater vehicles*), também têm sido alcançados progressos na área de veículos robóticos aéreos (UAVs, do inglês *unmanned aerial vehicles*). Embora UAVs sejam importantes componentes de missões de reconhecimento (WONG, 1997, ELFES *et al.*, 1998a), agências como a NASA utilizam plataformas aéreas para pesquisa climática e ambiental, como as desenvolvidas pelo projeto ERAST (do Inglês *Environmental Research Aircraft and Sensor Technology* -Tecnologia de Aeronaves e Sensores para Pesquisa Ambiental) (ERAST, 2001), que visa o desenvolvimento de veículo aéreos não tripulados para monitoramento ambiental. Estes veículos são tipicamente comandados por controle remoto durante as fases críticas de suas missões, utilizando sistemas de navegação a bordo para executar o seguimento de trajetórias pré-programadas como no caso dos veículos Helios (HELIOS, 2001) ou Theseus (THESEUS, 1997).

O desenvolvimento de veículos robóticos aéreos autônomos não tripulados permite a aquisição de dados de forma que o usuário possa determinar a área a ser monitorada, a resolução espacial e temporal dos dados e, também, o sensor ideal para cada tipo de

missão, a custos relativamente baixos. Isto levará a uma expansão do uso científico e civil de dados aéreos e a benefícios sociais e econômicos significativos.

Entre os veículos utilizados como UAVs, os mais comuns são aviões de pequeno tamanho, seguidos por helicópteros. Dirigíveis, também conhecidos como veículos mais leves que o ar (LTA, do inglês *lighter-than-air*) têm sido foco recente de atenções, conforme detalhado a seguir.

### **1.3 Dirigíveis**

Os dirigíveis são veículos aéreos que, da mesma forma que balões, utilizam dispositivos conhecidos como envelope, para acondicionar um gás mais leve que o ar, requerido para gerar a força de sustentação para que este flutue no ar. Ao contrário dos balões, eles são manobráveis, em consequência do movimento gerado pela força de propulsão exercida por hélices, e pela presença de lemes em posições estratégicas de sua carenagem (KHOURY, 1999).

#### ***1.3.1 Aplicações e Tendências***

Os dirigíveis foram usados pela primeira vez em 1852, quando Henri Giffard voou no primeiro veículo com estas características. Foi o brasileiro Santos Dumont quem inventou o balonete (MOWFORTH, 1991), resolvendo o problema de manutenção da forma de dirigíveis infláveis. Ele construiu um dirigível com motor à gasolina e, em 1901, conseguiu realizar um percurso de 11Km sobre Paris, de Saint-Cloud à Torre Eiffel e vice-versa, num tempo menor que 30 minutos, ganhando um prêmio de 125.000 Francos (ELE-LIB 2001) e demonstrar de forma definitiva a dirigibilidade na navegação aérea.

Desde o início do século XX até o espetacular acidente com o dirigível Hindenburg, em 1937, os dirigíveis foram um importante meio de transporte. O Hindenburg, ainda detém o recorde de maior veículo aéreo construído, com seus 240m de comprimento (TURNER, 2000).

Os dirigíveis tiveram uso militar em bombardeamento e observação na 1ª Guerra Mundial e veículos de patrulhamento e guerra anti-submarinos na 2ª Guerra Mundial, com aproximadamente 170 unidades operacionais, das quais 16 operaram nas costas do Nordeste brasileiro. Em 1952, o Snowbird ou B2 voou durante 264,2h por uma distância de

aproximadamente 16.000Km, sem reabastecimento, mostrando a capacidade de operação de tais veículos (TURNER, 2000).

Após a década de 50 os dirigíveis foram quase que exclusivamente utilizados em propaganda e cobertura de eventos, tendo em vista a sua alta visibilidade, sendo adotados por empresas como Good Year (Good-Year, 1997), Fuji, Budweiser (AIR-FR, 2001). A Figura 1.1 (AIR-FR, 2001) mostra a disponibilidade de dirigíveis de grande porte no mundo; dos 38 citados, mais de 30 são utilizados em propaganda.

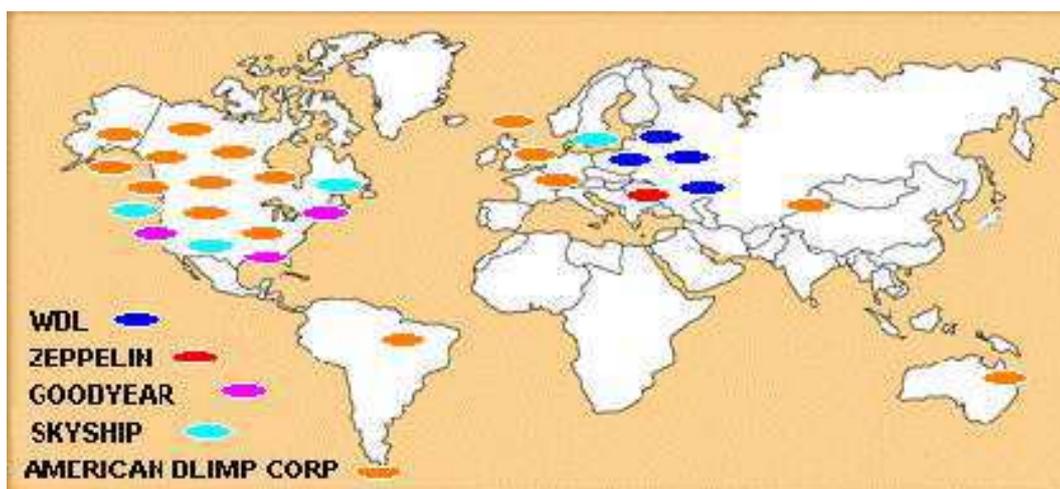


Figura 1.1: Dirigíveis disponíveis atualmente.

Com a crise do petróleo no final da década de 70, e a busca por aeronaves mais econômicas, o dirigível voltou a ser considerado como uma alternativa. O projeto do veículo Sentinel-1000, construído pelo Westinghouse Defense Group para a marinha americana, foi desenvolvido visando o uso em comunicações, transporte aéreo e repressão ao tráfico de drogas (HOTAIR, 2001). Ele marcou o início do ressurgimento dos dirigíveis com aplicação de componentes e técnicas do final do século XX. Como exemplos, pode-se citar a construção em material composto das superfícies aerodinâmicas (MAT-CP, 2001), um sistema elétrico/eletrônico de comando FBW (do inglês *flight by wire*) (KHOURY, 1999), que provocou uma melhoria considerável na qualidade de operação do veículo, em relação aos sistemas puramente mecânicos baseados em cabos. A Tabela 1.1 apresenta as suas características (SKYSHIP, 2001).

O final da década de 90 presenciou o ressurgimento do Zeppelin-NT (ZEPNT, 2001) com um inovador sistema de propulsão montado na popa, para melhorar a manobrabilidade em terra do veículo e com aplicação voltada para transporte de

passageiros com fins turísticos e coleta de imagens e dados. A Tabela 1.1 também apresenta as características do Zeppelin-NT.

<b>Veículo</b>	<b>Dimensões comp. x di-âmetro (m)</b>	<b>Volume do gás (m<sup>3</sup>)</b>	<b>Velocidade (km/h)</b>	<b>Carga Útil (ton.)</b>	<b>Tripulação (pilotos e pessoal de solo)</b>	<b>Motores (HP)</b>
Sentinel-1000	68 x 16,6	10000	96 110 (max)	2,4	25	2 x 300
Zepelin-NT	75 x 19,5	8225	(*)	1	2 e 12 passageiros	(*)
Cargo-Lifter	260 x 82	550000	(*)	160 (3200)m <sup>3</sup>	(*)	(*)

(\*) dado desconhecido

Tabela 1.1: Características do Sentinel-1000, Zepelin-NT e Cargo-lifter.

Uma iniciativa que merece ser reportada é o projeto Cargo-Lifter (do alemão levantador de cargas), com objetivo de transporte de cargas volumosas e pesadas com custos elevados de transporte, como geradores de usinas hidroelétricas. O peso das cargas que poderão ser carregadas por este veículo estão na faixa de 160 toneladas e 3200m<sup>3</sup> (50m x 8m x 8m) (CLF, 2001). As suas características também são apresentadas na Tabela 1.1. Segundo a empresa, o dirigível e sua logística já foram avaliados em simulação e, brevemente, serão testados em protótipos (CLF, 2001a).

<b>Veículo</b>	<b>Empresa</b>	<b>Previsão</b>	<b>Comprimento (m)</b>	<b>Carga Paga (ton.)</b>	<b>Motores(KW)</b>
CL 160	Cargolifter	2002	260	160	5700
SkyCat 20	ATG	2001	82	20	(*)
SkyCat 200	ATG	-	185	200	(*)
SkyCat 1000	ATG	-	307	1000	(*)
DPD 5000	RAS-(RAS, 2001)	-	126,8	-	(*)

(\*) dado desconhecido

Tabela 1.2: Projetos de dirigíveis para transporte de cargas.

Outra iniciativa ainda mais recente, o Sky Cat (SKYCAT, 2001), baseado na idéia do Cargo-Lifter como veículo de transporte de carga pesada, usa o conceito de formato de

casco tipo catamarã e princípios de aterrissagem similares aos dos *hovercraft*. Isto, segundo AGUIRRE (2001), superará os problemas de manobrabilidade em terra dos dirigíveis. O veículo proposto para tal terá largura máxima de 136m, comprimento de 307m, altura total de 77m, envelope com 2.000.000m<sup>3</sup> e capacidade de carga de 1000 toneladas.

A Tabela 1.2 (BLPEUR, 2001) resume os projetos de dirigíveis para transporte e sua características gerais e previsão de prazos de construção.

Uma outra aplicação vislumbrada para dirigíveis é como antenas em sistemas de comunicação, estacionados em altitudes de cerca de 20Km, conhecidos como sistemas HALE (do Inglês *High Altitude Long Endurance* – alta altitude longa duração), que apresentariam grandes vantagens sobre satélites. Como exemplo tem-se os projetos do (AIST) *National Institute of Advanced Industrial Science and Technology* do Japão (ONDA,1999, AIST, 2001) e da Universidade de Stuttgart (HALP, 2000, 2001) e mais duas propostas de projetos comerciais o StratSat da SkyCat (SKYCAT, 2001a) e o da SkyStation (SKYSTA, 2001).

Esta aplicação, segundo estudos recentes mostrados em HALP (2000) e ONDA (2001), deverá ainda superar os seguintes desafios tecnológicos:

- i) lançamento do veículo - pois o balonete no lançamento deveria ocupar aproximadamente 90% do volume;
- ii) dimensões, pois o veículo teria 300m comprimento e 120m de diâmetro para suportar as células solares (estimativa de peso 2 ton., área de 6000 m<sup>2</sup>), para suprir energia ao sistema de propulsão elétrico de 500KW, e baterias para armazenar energia para noite (7,5MWh);
- iii) as células solares em si;
- iv) a variações de temperatura entre o dia e a noite.

Apesar desses desafios, essa aplicação de dirigível parece ter um grande potencial.

Outra área em que têm aparecido propostas para a aplicação de dirigíveis é a aquisição de imagens e monitoramento ambiental. Para o caso de dirigíveis não tripulados cita-se, como exemplo, os projetos de dirigíveis com acionamento por energia solar, como o Lotte da Universidade de Stuttgart na Alemanha (LOTTE, 2001) ou do AZTEC iniciativa de alunos da Virgínia Tech nos EUA (AZTEC, 2001). Ainda com este objetivo, citam-se

projetos em fase inicial como o EnviroBlimp da Carnegie Mellon University – EUA (KANTOR, 2001), ou a proposta de dirigível robótico do LAAS – França (LACROIX, 2000, 2001). O projeto AURORA (ELFES *et al.*, 1998a, 1998b), iniciado em 1996 no CenPRA no Brasil, apresentado na Seção 1.5, que objetiva inicialmente o ensaio de conceito com dirigível de pequenas dimensões e, depois, o uso de grandes dirigíveis para a aplicação em monitoramento ambiental, é uma iniciativa pioneira em escala mundial.

Dirigíveis tripulados para aplicação em monitoramento ambiental como uma possível área de utilização, pode-se citar o USLTA (USLTA, 2001) e o Ecoblímp para monitoramento de vida marinha na região da Florida, EUA (ECOBLIMP, 2001).

### ***1.3.2 Dirigíveis não-Tripulados***

Os dirigíveis não tripulados podem ser tele-operados, autônomos ou quase autônomos, como será visto a seguir.

#### ***Dirigíveis Tele-operados***

Já no início da década de 80, começaram a aparecer dirigíveis tele-operados, utilizando componentes de aeromodelismo, em dimensões variando de 1 a 15m (AIRSPEED, 2001, SKYMIDIA, 2001, SKYBUS, 2001). Eles têm encontrado aplicação em lazer e propaganda em locais fechados, como centros comerciais e estádios. Mais recentemente, eles têm sido utilizados em propaganda e tomadas de imagens aéreas em locais abertos. O preço do uso destes dirigíveis no exterior é de cerca de US\$ 1200 por dia para dirigíveis com até 5 metros para uso em locais fechados e de cerca US\$ 4000 por dia para dirigíveis para uso externo, como o Skibus (SKYBUS, 2001).

Os pequenos dirigíveis, originalmente tele-operados também, têm sido usados em aplicações para comando remoto via Internet ou estudos para vôos em formação de veículos aéreos em ambientes fechados (MIT-FORM, 2001).

#### ***Dirigíveis Autônomos (ou quase autônomos)***

A seguir são descritos os projetos que apontam para a realização de vôos autônomos com dirigíveis. Considera-se que o veículo alcançou este resultado, ou está caminhando para tal, quando foram disponibilizados imagens ou dados que permitam verificar os resultados obtidos na realização dos ensaios.

<b>Dirigível</b>	<b>Skybus</b>	<b>LOTTE</b>	<b>SASS_LITE</b>	<b>AURORA-I</b>
<b>Tipo</b>	não rígido	não rígido	Não rígido	não rígido
<b>Volume</b>	84 m <sup>3</sup>	109 m <sup>3</sup>	976 m <sup>3</sup>	30 m <sup>3</sup>
<b>Comprimento</b>	15,5m	15,6m	30,5m	11,5m
<b>Diâmetro</b>	3,1m	4 m	7,6m	3,1m
<b>Volume do Balonete</b>	8% do envelope	8% do envelope	9% do envelope	10% do envelope
<b>Peso total</b>	91 kg	98 kg	(*)	23 kg
<b>Carga útil</b>	11 kg	6Kg	190Kg	7 kg
<b>Duração da missão</b>	5h	Máxima documentada 2,5h	12 a 24h	1H
<b>Máxima velocidade relativa ao ar</b>	76 km/h	46 km/h	72 km/h	50 Km/h
<b>Altitude</b>	360m	1000m	1000m	300m
<b>Alcance de comunicação</b>	GSM teoricamente sem limite, mas rádio controle 1,5Km	não especificado	100 Km	20 Km rádio modem, 1,5Km rádio-controle
<b>Geradores</b>	2 x 100W	1132 W (Célula solar 7,02 m <sup>2</sup> )	2x1600watts - 24 VDC.	Não possui
<b>Motores</b>	2 motores a Gasolina Zenoah G445	1 motor elétrico	2 motores de combustão	2x OS90
<b>Potência</b>	2 x 4,5 HP	1x2HP	2x 43HP (*)	2 x 3,5 HP
<b>Vetorização</b>	não		sim	sim
<b>Hélices</b>	(*)	800 rpm e 1,7m de diâmetro	(*)	0,4 m de diâmetro

Tabela 1.3: Dirigíveis com potencial para vôos autônomos onde (\*) indicam dados imprecisos ou desconhecidos

### SASS-LITE

É uma família de dirigíveis rádio controlados, desenvolvidos para monitoramento e reconhecimento de longa duração e outras aplicações militares. BOSCHMA (2001), afirma que este é o primeiro dirigível de grande porte com capacidade de operação autônoma. Este fato é corroborado por imagens de todo processo de vôo autônomo realizado em 1998 (SASS, 2001), mas só disponibilizadas em meados de 2001. Os sensores básicos do dirigível citados são GPS e outros sensores inerciais. Possui sistema de comunicação com alcance de 100Km com tripla redundância. Outros detalhes não são fornecidos.

Detalhes dos sensores não são fornecidos, bem como não são reveladas informações sobre tipo de trajetória executada (informa-se que é via pontos de passagem), algoritmos, etc. Presume-se das fotos disponibilizadas na Internet, que a decolagem e aterrissagem são manuais.

### Skybus

O Skybus (SKYBUS, 2001) é baseado no dirigível Wasp RPB (AHA, 2001), que foi reprojetoado, remanufaturado e acrescido de capacidade de operação autônoma,

capacidade esta não documentada por nenhum meio, exceto um *press release* emitido pela empresa. Estas informações foram divulgadas pela Internet em 2001 (SKYBUS, 2001). Cita-se que a capacidade de operação autônoma é implementada por controle de altitude, controle de velocidade e de movimentos através de pontos de passagem. Quanto à forma de operação, é citado que a aeronave é constantemente acompanhada por um veículo em terra, de tal forma que o piloto possa realizar pilotagem manual em caso de problemas. As suas características gerais constam da Tabela 1.3, e algumas características particulares estão descritas a seguir:

- Possui computador embarcado com GPS, sensor de altitude, sensor barométrico, velocidade do ar, temperatura, pressão e medidas elétricas;
- Transmissão de dados de vôo para estação de terra via GSM (Global System for Mobile communications), adequada para regiões onde há cobertura garantida para este serviço, como na Europa;
- Transmissão de vídeo do dirigível para a estação de terra com alcance de 2 Km;
- Estação de terra móvel via *Lap Top* com os transmissores conectados a este e monitoramento dos dados de vôo através de interface homem-máquina similar ao cockpit de um avião. Possui *software* gráfico para planejamento de vôos. O *software* controla o vôo da aeronave enviando e recebendo dados desta, e tem vários recursos de verificação e ajustes;
- São possíveis 16 pontos de passagem incluindo altitude e velocidade para cada ponto e mudanças de pontos durante o vôo;
- A decolagem e aterrissagem são manuais;
- Possui chave para mudar do modo de operação de manual para automático;
- Possui 2 motores com geradores, 2 baterias, 2 motores para enchimento de balonetes, 4 servos para controle de guinada, 2 receptores de rádio controle embarcados operando em 2 frequências diferentes, isto é, possui redundância contra falhas em alguns equipamentos;
- Possui recursos embarcados para prevenir falhas de GPS, mas isso não é explicitado e, em caso de falha do receptor de rádio-controle, há mudança para modo automático e o veículo irá para um ponto pré-definido;
- Tempo de vôo de 5 horas, com velocidade máxima de 70 Km/h, velocidade máxima de vento de 40 Km/h.

Enfim, são informadas as diversas características do veículo, entretanto o ponto ainda questionável são os resultados reais do seus vôos autônomos, que não foram mostrados.

## LOTTE

O projeto Lotte (LOTTE, 2001) visa a construção de um dirigível, que utiliza como fonte de energia células solares montadas sobre o envelope. Ele tem sido utilizado como um dirigível pilotado remotamente, com várias horas de vôo desde 1993. Ultimamente, a equipe do projeto vem introduzindo instrumentação na perspectiva de viabilizar o desenvolvimento do controle automático do veículo, conforme mostrado em (WIMMER, 2001), no qual se apresenta a projeto de um controlador de altitude e detalhes da instrumentação para uso em identificação e controle automático. Para tal foi construído um sistema utilizando um barramento de campo para conexão dos sensores e um esquema para chaveamento de modo manual para automático comandado pelo piloto. Como sensores principais são utilizados servo-acelerômetros, giroscópios de fibra ótica, sonda de vento e GPS.

### **1.4 Dirigíveis para Missões de Inspeção Aérea e Monitoramento Ambiental**

Para a realização de diversas tarefas de inspeção aérea e monitoramento ambiental, citadas na Seção 1.2, é requerida uma plataforma de aquisição de dados capaz de operar a baixa velocidade e baixa altitude. Tal plataforma deve ainda ser capaz de:

- Pairar acima de uma área;
- Ter capacidade de voar por longos períodos, para estudos de longa duração;
- Gerar baixo ruído e turbulência de tal forma a não perturbar o ambiente sendo medido e monitorado;
- Gerar baixa vibração, para diminuir a influência desta nos sensores e suas leituras, bem como no próprio veículo;
- Ser capaz de decolar e aterrissar verticalmente para que as operações de manutenção e reabastecimento possam ser feitas sem a necessidade de pista;
- Ser manobrável;
- Ter uma grande capacidade de carga em relação ao peso do veículo;
- Ter um baixo custo de operação e;

- Apresentar baixo risco em caso de falha no veículo.

Dos quatro veículos aéreos possíveis - aviões, helicópteros, dirigíveis e balões - os últimos não são aqui considerados por não serem manobráveis. Os três veículos restantes são comparados na Tabela 1.4 (ELFES *et al.*, 1998a) em relação aos requisitos definidos acima, onde uma alta concordância com o requisito é indicada com três marcas, e uma baixa concordância com uma marca.

Requisitos Operacionais	Avião	Helicóptero	Dirigível
Baixo custo de operação	✓	✓✓	✓✓✓
Vôo de longa duração	✓✓	✓	✓✓✓
Capacidade de pairar	✓	✓✓✓	✓✓
Razão carga útil / peso	✓✓	✓	✓✓✓
Grande manobrabilidade	✓✓	✓✓✓	✓
Baixo ruído e turbulência	✓	✓	✓✓
Decolagem e aterrissagem vertical	✓	✓✓✓	✓✓✓
Baixo consumo de combustível	✓✓	✓	✓✓✓
Baixo risco em caso de falha	✓	✓	✓✓✓
Relação capacidade de carga / volume	✓✓	✓✓	✓
Baixa vibração	✓✓	✓	✓✓✓

Tabela 1.4: Comparação de veículos aéreos como plataformas para missões de pesquisa e monitoramento ambiental.

Pode-se inferir da Tabela 1.4 que dirigíveis são, em média, mais adequados às tarefas relacionadas à inspeção e monitoramento ambiental que aviões ou helicópteros. Isto é devido, principalmente, ao fato de que a maior parte de sua sustentação vem de forças aerostáticas, em vez de forças aerodinâmicas. Portanto, um dirigível consome pouca energia (combustível) para flutuar no ar. Uma consequência desta propriedade aerostática é que dirigíveis necessitam para propulsão motores menos potentes que aviões e helicópteros, o que produz em consequência menor ruído, vibração, turbulência e menor consumo de combustível. Além disso, em caso de falha, o veículo desce lentamente reduzindo os riscos, ao contrário dos demais.

## 1.5 Projeto AURORA

O projeto AURORA do CenPRA, como explicitado anteriormente, visa o desenvolvimento de estratégias de controle e de navegação autônomas para dirigíveis não tripulados a serem usados como plataforma aérea para a inspeção e monitoramento

ambiental. Ele está baseado em trabalhos prévios realizados pelos pesquisadores do CTI, envolvendo o desenvolvimento de um veículo autônomo submarino ([ELFES, 1990) e de um helicóptero remotamente pilotado (RAMOS, 1995).

O AURORA foi concebido como um projeto multi-fase (Tabela 1.5), com uma seqüência de protótipos a serem desenvolvidos, capazes de, sucessivamente, atingirem tempos de missão e alcances maiores, com níveis incrementais de autonomia, evoluindo de um sistema teleoperado para um sistema telemonitorado autônomo.

	AURORA-I	AURORA-II	AURORA-III
Duração média de missão (h)	1-2	8	> 24
Distância média (km)	1-10	10-50	> 100
Capacidade de carga (Kg)	10	50	> 100

Tabela 1.5: fases do Projeto AURORA.

O projeto AURORA encontra-se em sua primeira fase, chamada de AURORA-I, em cujo contexto se enquadra esta tese.

## 1.6 Contribuições desta Tese

Esta tese contribui para o desenvolvimento de sistemas de controle e navegação automáticos para dirigíveis no projeto AURORA, cujo objetivo é de desenvolver e implementar os recursos tecnológicos necessários, demonstrando a viabilidade técnica do projeto com a realização de aplicações de baixa complexidade. Ressalta-se que, na evolução do projeto AURORA, a meta é se atingir um significativo grau de autonomia durante todas as fases de operação do dirigível, da decolagem à aterrissagem, incluindo a habilidade de planejar e executar sensoriamento e navegação, diagnosticar e recuperar-se de falhas, e de uma forma adaptativa replanejar missões com base em informação sensorial própria, incluindo visão computacional. Estes objetivos assim explicitados são um diferencial importante em relação aos demais projetos citados anteriormente.

Neste contexto, desde março de 2000 já vêm sendo executados vôos com controle de trajetória por pontos de passagem, (RAMOS, 2001a, 2001b) e, mais recentemente, vôos com controle de trajetória e altitude, apresentados inicialmente em MAETA (2001). Mais detalhes do veículo, da infra-estrutura para operação autônoma e resultados de simulação e experimentais, são apresentações no decorrer deste trabalho.

Esta tese se caracteriza pelo seu cunho eminentemente tecnológico e, conseqüentemente, esta também é a ênfase principal de seus resultados. Neste contexto, e considerando os trabalhos de desenvolvimento e integração requeridos para a criação de um dirigível robótico, as principais contribuições e outros resultados relevantes incluem:

- O estabelecimento de metodologia para o desenvolvimento e operação de dirigíveis robóticos;
- A evolução eletromecânica e aerodinâmica do dirigível, transformando um veículo originalmente rádio-controlado num dirigível capacitado a operar como veículo robótico;
- O desenvolvimento e aprimoramento da infra-estrutura computacional e sensorial embarcada e de terra, incluindo soluções específicas de *software* e *hardware*;
- A concepção e desenvolvimento de um ambiente integrado para apoiar o desenvolvimento e operação do sistema robótico;
- O aprimoramento de modelo matemático de dirigíveis, incorporando um modelo de propulsão particularizado para o veículo utilizado, e desenvolvimento de simuladores de dirigíveis em diferentes ambientes para apoio às atividades de projeto treinamento e operação;
- A reinterpretação, sintonia e testes de algoritmos de controle utilizados em vôo real;
- O desenvolvimento e testes em simulação de um protótipo de arquitetura de *software* robótico para tratar aspectos de deliberação e reação no dirigível autônomo;
- A validação experimental do protótipo de um dirigível robótico autônomo, com capacidade de vôo por pontos de passagem.

Além dos resultados acima citados, acrescenta-se a formação de pessoal através da orientação, de fato, de 2 dissertações de mestrado e de trabalhos de iniciação científica, bem como a publicação em temas associados a esta tese de 2 artigos em periódicos como

co-autor; 9 artigos como autor principal e outros 19 como co-autor em conferências nacionais e internacionais com comitê de avaliação.

Como explicitado anteriormente, este trabalho de tese caracteriza-se por seu cunho tecnológico, integrando diferentes domínios do conhecimento, e além disso insere-se em um projeto institucional - o Projeto AURORA. É portanto inevitável, e mesmo salutar, a contribuição ou participação de muitas pessoas no seu desenvolvimento, inclusive pesquisadores, estudantes de pós graduação e de iniciação científica. O Anexo 3 caracteriza as interações entre este trabalho de tese e outros trabalhos, desenvolvidos por outros membros do AURORA, que aqui são utilizados ou dele se originaram.

## 1.7 Organização do Texto

O presente trabalho, além deste capítulo introdutório, conta com outros nove capítulos e três anexos.

O Capítulo 2 trata da especificação de requisitos para o desenvolvimento do dirigível robótico autônomo, gerando uma descrição conceitual contemplando o desenvolvimento e operação do dirigível, caracterizando os principais componentes requeridos para o desenvolvimento do dirigível robótico.

Os Capítulos 3 a 5 estendem desenvolvimentos encontrados na literatura, para o caso do dirigível; assim:

- O Capítulo 3 apresenta o modelo matemático do dirigível;
- O Capítulo 4 apresenta a revisão da literatura no que concerne o controle de dirigíveis e, os controladores que compõem o sistema de controle e navegação deste.
- O Capítulo 5 apresenta um panorama a cerca das capacidades dos robôs, com ênfase nas motivações que levaram ao desenvolvimento de arquiteturas de software robótico híbridas. São então descritos os principais componente dessas arquiteturas, apresentando com maiores detalhes os seus princípios de operação. O capítulo se encerra com a apresentação do modelo conceitual da arquitetura de *software* proposta para o dirigível.

Os Capítulos 6 a 9 tratam das implementações; assim:

- O Capítulo 6 apresenta os simuladores desenvolvidos a partir do modelo matemático do dirigível;
- O Capítulo 7 apresenta o ambiente integrado concebido para o desenvolvimento e operação do dirigível;
- O Capítulo 8 apresenta estudos sobre a implementação do sistema de controle e navegação do dirigível, com os resultados obtidos tanto em simulação quanto em voo real, validando as capacidades de realização de movimentos básicas que permitirão o desenvolvimento de um dirigível robótico autônomo;
- O Capítulo 9 apresenta o mapeamento do modelo conceitual da arquitetura de *software* robótico do dirigível no ambiente de desenvolvimento e operação, gerando-se um protótipo simulado dessa arquitetura. Então, são apresentados os resultados obtidos em estudos de caso simulados utilizando este protótipo, configurando uma primeira aproximação da solução a ser implantada no veículo robótico.

O Capítulo 10 apresenta a avaliação dos resultados obtidos, as conclusões do trabalho e sugestões de trabalhos futuros.

Os três anexos detalham aspectos abordados neste trabalho; assim:

- O Anexo 1 descreve o dirigível em si e a sua evolução eletro-mecânica para torná-lo um veículo robótico. Ele descreve também a infra-estrutura completa de *software* e *hardware*, abrangendo as soluções estabelecidas para a integração dos diferentes componentes do dirigível robótico;
- O Anexo 2 apresenta a caracterização e avaliação do sistema de propulsão utilizado para o dirigível;
- O Anexo 3 apresenta com maiores detalhes a relação entre os trabalhos desenvolvidos nesta tese e outros trabalhos.

## **2. ESPECIFICAÇÃO DE REQUISITOS**

### **2.1 Introdução**

Considerando o contexto de desenvolvimento de um dirigível robótico autônomo, este capítulo tem como objetivo estabelecer a fundamentação deste sistema, originando um modelo conceitual na forma definida por (LARMAN, 1998). Este modelo possibilitará a explicitação dos principais componentes do dirigível robótico autônomo, divididos em diferentes categorias, e suas relações. A importância do conceito de componentes e das interações entre eles, tanto em termos de *software* quanto de *hardware*, reside na possibilidade de troca e reutilização destes componentes da forma mais natural possível. Assim, idealmente, a mudança num dos componentes não afetará o sistema como um todo (FRASER, 1994). A adoção desta metodologia permite a evolução gradual do dirigível robótico, através de etapas sucessivas e de complexidade crescente, organizando e, por conseguinte, reduzindo o esforço de desenvolvimento.

Neste capítulo, a Seção 2.2 trata dos requisitos que definem o sistema, ou seja "**o que**" é esperado que o dirigível robótico autônomo realize. Um subconjunto desses requisitos define o contexto desta tese; ele é explicitado na Seção 2.3. Na Seção 2.4 apresenta-se o projeto conceitual, no qual se baseia o desenvolvimento e implementação das soluções necessárias, de forma a satisfazer os requisitos propostos para o trabalho de tese, gerando um primeiro protótipo de dirigível robótico; ao mesmo tempo, este modelo conceitual visa permitir a evolução do protótipo, de forma a contemplar, posteriormente, os demais requisitos que caracterizam o dirigível robótico autônomo. A Seção 2.5 apresenta a arquitetura funcional implementada no dirigível; finalmente a Seção 2.6 apresenta as conclusões do Capítulo.

### **2.2 Descrição dos Requisitos para um Dirigível Robótico Autônomo**

Como salientado no primeiro capítulo, na primeira fase do Projeto AURORA, denominada AURORA I, objetiva-se desenvolver a base tecnológica e científica para um dirigível robótico autônomo e a validação deste conceito, através de missões de pequena complexidade, com duração de no máximo uma hora e carga útil limitada a 10 Kg. Mesmo considerando-se a reduzida complexidade das missões, ditada principalmente pelas restrições do próprio veículo usado nesta primeira fase (detalhes do AS800 são fornecidos

no Anexo 2), o dirigível robótico deve evoluir gradualmente de um sistema teleoperado para um sistema telemonitorado e dotado de capacidade de operação autônoma, atingindo gradualmente capacidades análogas às de robôs inteligentes. Em termos metodológicos, esta evolução enfoca inicialmente aplicações e cenários de utilização simples que progressivamente vão ganhando maior complexidade, agregando-se gradualmente ao veículo as capacidades de autonomia e inteligência.

Além do aspecto de autonomia, para o desenvolvimento do dirigível robótico são necessários componentes de suporte, tipicamente infra-estruturas de *hardware* e *software* e interface de operação, além componentes de apoio, como ambientes de simulação. Esses assuntos são tratados a seguir.

### 2.2.1 Componentes Básicos de um Dirigível Robótico.

Veículos robóticos envolvem, em geral, pelo menos três componentes básicos (ARMINGOL, 1999, FRASER, 1994): o veículo em si e a infra-estrutura de processamento nele presente, uma infra-estrutura de processamento para a operação e monitoramento remoto do veículo e meios de comunicação entre ambas infra-estruturas.

Analogamente, para o caso do dirigível robótico, a Figura 2.1 mostra o conceito geral do sistema, constituído de três componentes básicos: a estação embarcada montada no dirigível, a estação de terra e o sistema de comunicação entre as estações.



Figura 2.1: Conceito geral do dirigível robótico.

Na Estação Embarcada estão montados os sensores internos necessários à operação autônoma do dirigível robótico; os sensores de missão, específicos para a realização de uma dada aplicação de inspeção ou monitoração aérea; os atuadores que conferem dirigibilidade ao veículo; os computadores embarcados; além dos rádios de comunicação com a estação de terra. Todos estes componentes devem estar devidamente interligados para assegurar o funcionamento do dirigível como um veículo robótico. Também estão montados na gôndola o sistema de suprimento de energia para alimentação elétrica do conjunto e o reservatório de combustível.

A Estação de terra, móvel, é composta de uma infra-estrutura de processamento e comunicação, montada no local de lançamento do dirigível e destinada principalmente a apoiar a realização de missões e prover a interface entre o veículo e seu operador. Ela é composta de computadores, rádios para comunicação com a estação embarcada localizada no veículo, sensores de apoio ao vôo e recursos para comunicação externa, provendo também meios para a difusão de dados coletados pelo dirigível quando de sua operação. Também compõem a estação de terra um mastro portátil de atracamento e um veículo para transporte de equipamentos.

### ***2.2.2 Capacidades do Dirigível Robótico***

Uma característica do processo de desenvolvimento tecnológico numa área ainda não completamente estabelecida, como no caso geral de robôs e, mais ainda, no caso de dirigíveis robóticos, que constitui uma área de pesquisa recente na comunidade científica, é a evolução gradual do sistema para níveis de desenvolvimento e cenários de aplicação com capacidades crescentes. A Tabela 2.1 mostra estes cenários com evolução de sistema teleoperado para sistema robótico autônomo telemonitorado. Verifica-se pela tabela que, na medida que o cenário evolui, o nível de abstração em termos de especificação de missões aumenta: se no cenário mais simples tem-se uma ação unicamente "operacional", desempenhada pelo piloto, os cenários mais abstratos envolvem majoritariamente uma ação "tática e estratégica" que é contemplada por um conjunto de soluções que agregam autonomia ao sistema. Assim, nesta evolução gradual, as especificações e implementações vão sendo continuamente elaboradas e adaptadas segundo os diferentes cenários do dirigível robótico.

Cenário	Capacidade
1- Dirigível comandado por controle remoto	Comando dos dispositivos de atuação do veículo diretamente pelo piloto em terra, que acompanha visualmente a sua evolução, seja por observação direta, seja indiretamente através de imagens geradas por câmeras a bordo do veículo
2- Dirigível teleoperado e telemonitorado	Comando do veículo e observação de sua situação através de sensores e pelo envio ao solo da situação instantânea do veículo medida pelos sensores
3- Dirigível robótico supervisionado	Capacidade de executar repetidamente comandos de movimentos pré-estabelecidos e observação da sua situação por dados de telemetria.
4- Dirigível robótico autônomo	Capacidade de executar e alterar planos de ação em função de informações captadas pelo sistema de percepção. A especificação da missão é feita num alto nível de abstração.

Tabela 2.1: Cenário para evolução do dirigível robótico.

A seguir, apresenta-se um conjunto de capacidades idealizadas para o dirigível robótico e que correspondem aos componentes básicos de um sistema robótico inteligente, baseando-se nos conceitos apresentados em FRASER (1994) e ARMINGOL (1999). Elas incluem: planejamento e controle de missão; navegação/controle de movimentos e percepção.

### ***Planejamento e Controle de Missão***

Conforme mostrado em FRASER (1994), o planejamento e controle da missão de um sistema robótico traduz especificações num alto nível de abstração para níveis sucessivamente mais baixos e, baseado nas informações do sistema de percepção, adapta o plano às mudanças do ambiente. O planejamento e controle de missão definem o que deve ser executado pelo sistema de Navegação / Controle de movimentos em termos da rota a ser seguida, locais a serem visitados, precisão requerida e temporizações, se necessárias. No caso do dirigível robótico, o planejamento e controle de missão contempla dois aspectos:

i) Antes do início da missão é realizado um planejamento prévio, normalmente decompondo a missão em uma seqüência de tarefas, levando-se em consideração as limitações inerentes do veículo e aquelas impostas tanto pelo ambiente onde ele evolui quanto pela aplicação específica;

ii) No decorrer da missão, o controlador de missão, além de gerar comandos de navegação para que se cumpra o planejado, deve também acompanhar a sua execução; neste acompanhamento, o controle de missão deve ter capacidades para auto diagnóstico e conseqüente auto correção, replanejando a missão se necessário for para adaptá-la automaticamente às situações detectadas pelo sistema de percepção. Por exemplo, na ocorrência de imprevistos como mudanças climatológicas bruscas, mudanças naturais no decorrer da missão decorrentes de leituras dos sensores específicos da missão, como a detecção de locais interessantes a serem monitorados que não estavam previstos ou, então, devido a condições excepcionais como consumo anormal de combustível ou mesmo a constatação da degradação ou falha de um componente.

### ***Navegação / Controle de Movimentos***

A Navegação / Controle de Movimentos consiste em duas atividades distintas: a navegação, que é a definição dos caminhos ou a rota a ser seguida, respeitando obstáculos naturais e, o controle fino dos movimentos do veículo.

#### **Navegação**

O veículo deve ter capacidade de navegação, ou seja, dispor de trajetórias geradas em função dos objetivos de uma determinada missão. As trajetórias podem ser definidas *a priori* ou determinadas em tempo real. Para tanto, devem ser consideradas restrições associadas tanto ao próprio veículo (como raio mínimo de curvatura, potência dos motores e não-holonomicidade) (LATOMBE, 1991), como aquelas inerentes ao relevo, orientação e intensidade de vento ou, ainda, regiões “proibidas” por conterem obstáculos ou condições climáticas adversas.

Em sua forma mais simples, a definição *a priori* da trajetória a ser seguida pode ser feita a partir da descrição de coordenadas de latitude e longitude (*waypoints*) associadas a um perfil de altitude, em função do relevo da região, decompondo o vôo segundo primitivas básicas como decolagem, vôo cruzeiro, vôo pairado e aterrissagem.

Num nível maior de complexidade, as informações usadas pelo sistema de navegação para definir *on line* o caminho a ser seguido, poderão ser obtidas a partir de radares, ou outros sensores visuais, utilizando informações do local sendo visitado como por exemplo: marcas de terreno, margens de rios e de lagos, contorno da costa, tubulações e linhas de transmissão.

### Controle de movimentos

O veículo deve ter a capacidade de controle de movimentos, ou seja, capacidade de executar as primitivas de voo descritas anteriormente como decolagem, voo cruzeiro, voo pairado, e aterrissagem, atendendo a especificações de desempenho frente a eventuais perturbações presentes no meio ambiente, principalmente ventos e turbulências.

O controle de dirigíveis, apresentado no Capítulo 4, constitui a base para a automatização das primitivas de voo. Eventualmente, poderão ser usadas, como alternativas para controle dos movimentos, técnicas de visão computacional de alta performance, como o *visual servoing* (ESPIAU, 1992)

### ***Percepção***

O sistema de percepção deve possibilitar a geração de informações sensoriais com diferentes níveis de abstração, adequados ao nível do módulo que o utiliza, dentro do sistema autônomo. Assim, para um algoritmo de controle de movimentos, são necessárias as medidas de posição e atitude do veículo; por outro lado, em níveis mais estratégicos, o tipo de informação requerida pode ser mais abstrato, variando desde valores binários informando se o movimento do veículo está sendo executado normalmente, até medidas de probabilidade associadas a eventos em um processo decisório.

### **2.2.3 Recursos de Suporte**

Para o desenvolvimento de um dirigível robótico, são necessários recursos de suporte sobre os quais são implantadas as capacidades enumeradas anteriormente, dotando o veículo de autonomia. Tipicamente, esses recursos envolvem infra-estruturas de *hardware* e *software* para integração de componentes, e uma interface homem-máquina para a operação do veículo. Eles são tratados a seguir.

### ***Software de Tempo Real***

Para o desenvolvimento de tarefas críticas, principalmente as que atendem a dispositivos que enviam sinais importantes, como a maioria dos sensores internos e rotinas de controle do veículo, deve ser usado um sistema operacional que, além de ser multitarefa, possua características *hard real time*, isto é, garanta o atendimento de interrupções de diferentes graus de prioridade.

### ***Arquitetura de Software Robótico***

O *software* robótico deverá ser implementado com características de arquitetura híbrida permitindo a integração de capacidades deliberativas com reativas, conceitos descritos no Capítulo 5, as quais oferecem construções que facilitam o desenvolvimento de aplicações visando o monitoramento e controle de missões e seu replanejamento.

### ***Interface Homem-máquina***

Como foi explicitado anteriormente, a estação de terra deve incorporar uma interface homem-máquina (IHM) que constitui o elo de ligação entre o operador e o dirigível robótico. O papel e requisitos da IHM dependem do tipo de interação entre o operador e o veículo.

Para a teleoperação, a IHM deve fornecer ao operador as informações necessárias sobre o estado do dirigível, como posição, altitude, atitude, e do ambiente em que ele se encontra, como dados meteorológicos e imagens do solo obtidas a bordo. Ela deve também prover comandos para que o operador possa efetuar a pilotagem do veículo.

No caso de um veículo robótico supervisionado ou autônomo, além das características citadas, a IHM deve ainda:

i) servir de apoio ao controle e navegação, permitindo por exemplo a integração de algoritmos de planejamento de trajetórias e de controle de movimentos e o ajuste dos parâmetros de sintonia associados aos algoritmos e;

ii) fornecer recursos para a definição, acompanhamento e eventual modificação da missão do veículo.

Os dados de telemetria recebidos devem ser também armazenados, para serem utilizados posteriormente ao voo para, por exemplo, estudos de comportamento da

dinâmica do veículo e procedimentos de identificação de modelos, análise do desempenho dos sistemas de missão, navegação e controle, etc.

Na IHM, é extremamente útil a representação da área sobrevoada pelo veículo aéreo através de mapas topográficos, de forma a dar suporte às tarefas de planejamento e controle de missão e navegação por trajetórias, na definição *a priori* da trajetória a ser seguida, em termos de latitude, longitude (*waypoints*) e perfil altitude por exemplo; durante a execução da missão, os mapas citados devem permitir ainda o georeferenciamento, tanto da localização e orientação do veículo, quanto de informações quantitativas e qualitativas associadas à área sobrevoada e obtidas de sensores específicos da missão. Nesse mesmo contexto, é também relevante a integração à IHM de um sistema de informações geográficas (do Inglês GIS – *Geographic Information System*).

### ***Integração dos Componentes***

O desenvolvimento de um dirigível robótico envolve a integração, pelo uso de arquitetura de *hardware* e *software*, de inúmeros dispositivos, divididos entre componentes embarcados e de terra, abrangendo:

i) os equipamentos da estação de terra - computador, monitor de vídeo, um sensor de referência em um sistema diferencial de posicionamento global (do Inglês DGPS – *Differential Global Positioning System*), e uma estação meteorológica cuja função é de prover dados sobre a região de operação do dirigível, notadamente aqueles associados às perturbações como ventos e chuvas pois estas influenciam diretamente a missão do veículo;

ii) os equipamentos embarcados, como sensores para navegação (acelerômetros, inclinômetros, giroscópios, bússola, altímetro, intensidade e orientação do vento, GPS e câmera montada sobre um *pan tilt*), sensores para a monitoração do dirigível em si (rotação dos motores, nível de combustível, parte elétrica, temperatura), os sensores específicos para a execução de uma determinada missão, vários processadores e um sistema de suprimento de energia para alimentar todos estes equipamentos;

iii) o sistema de comunicação entre o veículo e a estação de terra, compreendendo *links*, via rádio, para transmissão e recepção de dados digitais entre o dirigível e a estação

de terra, e para a transmissão analógica de vídeo e imagens do dirigível para a estação de terra; parte do sistema é embarcado no dirigível e parte compõe a estação de terra.

### ***Ambiente de Apoio ao Desenvolvimento e Operação***

Deve ser disponibilizado um ambiente que apoie às fases de desenvolvimento e operação. Esse ambiente deve auxiliar tanto no desenvolvimento, operação e testes dos algoritmos que controlam o veículo na fase de concepção, desenvolvimento e operação, quanto na realização das missões. Para o desenvolvimento de algoritmos de controle de movimentos e de missão devem ser utilizadas ferramentas de apoio ao seu projeto, como por exemplo o MATLAB<sup>®</sup> (MATHWORKS, 2001).

Devido às características desejadas, este ambiente deve possibilitar diferentes modos de utilização, como simulado, híbrido e real. No modo simulado, ele deve congrega modelos matemáticos para simular com a máxima fidelidade possível os movimentos do veículo e os seus sensores, permitindo o teste simulado dos diferentes comportamentos do veículo, principalmente frente a ventos, turbulências e obstáculos. No modo real, o ambiente destina-se ao ajuste de algoritmos e seus parâmetros, bem como à operação do veículo quando da realização de vôos reais. No modo híbrido o ambiente deve permitir o uso de equipamentos reais em conjunto com outros que são simulados..

### ***Ambiente de Simulação***

Deve ser disponibilizado um ambiente de simulação do dirigível, que utilize um modelo matemático que inclua as forças aeroestáticas, dinâmicas, aerodinâmicas, gravitacionais, de empuxo e de propulsão agindo no veículo, permitindo a obtenção da resposta da aeronave aos diferentes comandos nos seus atuadores, além de considerar a influência de perturbações principalmente ventos ou falhas de motorização.

### ***Tolerância a Falhas***

A concepção do sistema robótico deve ser feita com características que permitam a tolerância a falhas. Isto implica na avaliação da necessidade de redundâncias de componentes e do desenvolvimento de mecanismos de degradação suave no caso de falhas, como abordado em FERNANDEZ (2000).

### ***Veículo***

O dirigível deve possuir características mecânicas que permitam a sua utilização como veículo robótico. Isso inclui: capacidade de carga adequada ao conjunto de componentes utilizados na sua operação; conjunto de propulsão apropriado, que permita enfrentar condições de ventos fracos a moderados sem colocar em risco a integridade do veículo, e que torne possível operações de decolagem e aterrissagem verticais; manobrabilidade adequada nas mesmas circunstâncias citadas, etc..

## **2.3 Especificação de Requisitos para esta Tese**

A seção anterior definiu os requisitos gerais para o dirigível robótico autônomo. O contexto desta tese engloba um subconjunto desses requisitos, explicitados a seguir.

Inicialmente as metas estabelecidas para esta tese foram:

- Construção de um ambiente de desenvolvimento e operação do dirigível que permitisse ensaios em simulação de técnicas de controle de trajetória por pontos de passagem em condições de vento com diferentes intensidades.
- Desenvolvimento de infra-estruturas de *hardware* e de *software* que permitissem ao dirigível a execução de trajetórias via pontos de passagem em condições de pouco vento, validadas experimentalmente.

Avaliando-se as metas especificadas e comparando-as com as capacidades apresentadas na Tabela 2.1, esta tese aborda aspectos relacionados ao desenvolvimento de um dirigível robótico supervisionado, provendo infra-estrutura para o desenvolvimento futuro de um dirigível robótico autônomo, através de componentes que poderão ser reutilizados.

Ao longo do desenvolvimento do trabalho foram incorporadas questões ligadas ao dirigível robótico autônomo, pelo estudo simulado de planejamento e controle de missão, frente a ocorrência de contingências, através da construção de um protótipo de arquitetura híbrida de *software* robótico.

A Tabela 2.2 e Tabela 2.3 apresentam o mapeamento das capacidades e recursos de suporte para o desenvolvimento de um dirigível autônomo, associando-os aos requisitos deste trabalho.

Capacidade	Requisitos para esta tese
Navegação e controle de movimentos	<ul style="list-style-type: none"> <li>• Estabilização do veículo em voo;</li> <li>• Seguimento trajetórias definidas por pontos de passagem, padrões de altitude e velocidade.</li> </ul>
Percepção / Sensores	<ul style="list-style-type: none"> <li>• Infra-estrutura para tratamento de sensores internos requeridos para a execução de primitivas de voo (decolagem, voo cruzeiro, voo pairado, aterrissagem).</li> </ul>
Planejamento / Controle de Missão	<ul style="list-style-type: none"> <li>• Configuração da missão como uma seqüência de movimentos pré-definidos;</li> <li>• Execução da missão levando-se em consideração a ocorrência de eventuais falhas;</li> <li>• Possibilidade de redefinição da missão em execução.</li> </ul>

Tabela 2.2: Mapeamento dos capacidades gerais para os requisitos implementados neste trabalho.

Recursos	Requisitos para esta tese
Ambiente de Desenvolvimento e Operação	<ul style="list-style-type: none"> <li>• Desenvolvimento de recursos que permitam ao ambiente modos de operação: simulado, real e híbrido.</li> </ul>
Modelagem e Simulação de Dirigível	<ul style="list-style-type: none"> <li>• Simulador incluindo o modelo dinâmico do veículo abrangendo as diferentes forças que nele atuam incluindo componentes de atuação representados de forma adequada para a realização de previsões necessárias ao comportamento do veículo frente a ventos e turbulências;</li> <li>• Possibilidade de simular algumas falhas de componentes em simulação.</li> </ul>
Interface homem-máquina	<ul style="list-style-type: none"> <li>• Programação da seqüência de movimentos que constituem a missão, através da especificação de coordenadas geográficas dos pontos de passagem e perfil de altitude do veículo;</li> <li>• Monitoração dos estados do veículo;</li> <li>• Acompanhamento da seqüência de movimentos programada;</li> <li>• Recursos para armazenagem de dados de voo e verificação <i>a posteriori</i> destes;</li> <li>• Alteração dos parâmetros dos algoritmos de controle.</li> </ul>
Integração dos Componentes	<ul style="list-style-type: none"> <li>• Integração dos componentes básicos correspondendo a integração das infra-estruturas de <i>hardware</i> e <i>software</i>, embarcado e em terra, incluindo o conjunto de sensores e atuadores e o sistema de comunicação.</li> </ul>
Arquitetura de <i>Software</i> Robótico	<ul style="list-style-type: none"> <li>• Ensaio preliminar de uma arquiteturas de <i>software</i> robótico, através de estudos simulados de aplicações ligadas a capacidade de planejamento e reação a contingências.</li> </ul>
Veículo	<ul style="list-style-type: none"> <li>• Evolução eletromecânica e aerodinâmica do dirigível tornando-o apto a operação como veículo robótico.</li> </ul>

Tabela 2.3: Mapeamento dos recursos de suporte para os implementados neste trabalho.

## 2.4 Projeto Conceitual

As seções anteriores apresentam as principais capacidades de um sistema robótico inteligente, bem como recursos de suporte, consubstanciados como requisitos no contexto do desenvolvimento de um dirigível robótico autônomo. Mostrou-se também como estes requisitos estão associados às metas definidas nesta tese.

O diagrama mostrado na Figura 2.2 explicita a relação entre o sistema robótico e os recursos de suporte. Observa-se, à esquerda da figura, um sistema robótico conectado aos recursos de suporte utilizados para o seu desenvolvimento e operação, ou seja recursos de IHM, recursos de projeto assistido por computador (do Inglês *Computer Aided Design* — CAD), modelo para simulação (como modelos do veículo, modelos de sensores, etc.) e recursos de armazenamento de dados (base de dados). Nesta figura, a comunicação entre estes componentes é transparente.

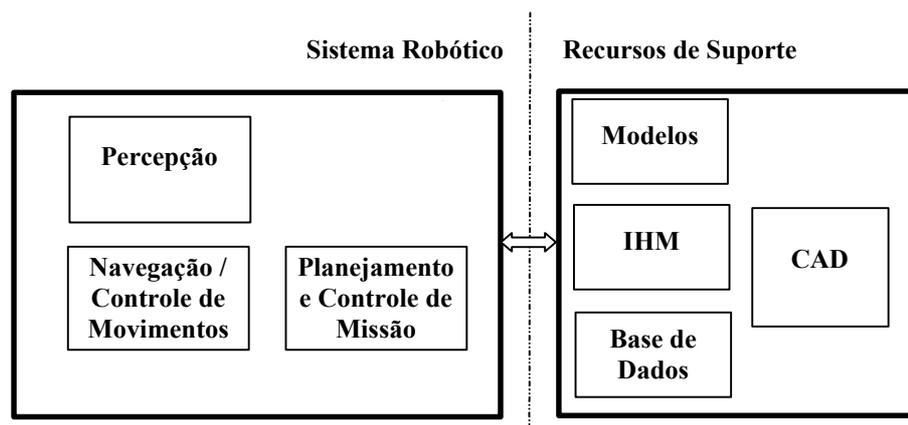


Figura 2.2: Relação entre componentes do sistema robótico e componentes de suporte.

Conforme especificado anteriormente, o projeto de um dirigível robótico autônomo deve contemplar tanto as atividades de desenvolvimento quanto as de operação. É possível, então, estabelecer uma analogia com o modelo CIMOSA (ESPRIT, 1993), que caracteriza o projeto conceitual de um sistema de manufatura em componentes utilizados nas atividades de desenvolvimento e operação, interrelacionados através de uma infra-estrutura de integração composta por elementos de comunicação e base de dados.

exceto que, em atividades de desenvolvimento, são requeridos modelos de simulação do dirigível, de seus sensores e atuadores, enquanto que na operação é requerido o veículo real.

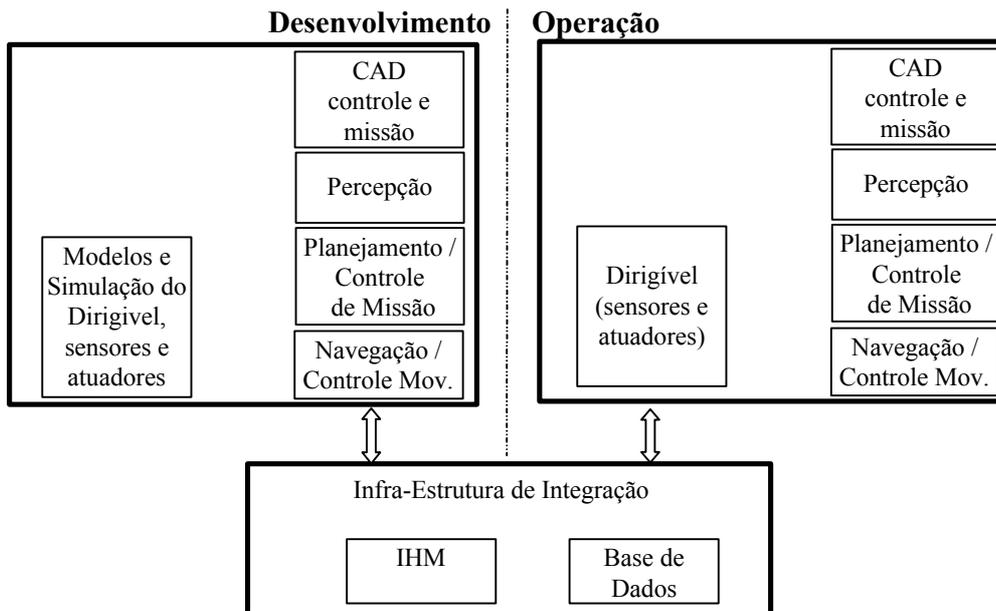


Figura 2.3: Componentes associados às atividades de desenvolvimento/operação.

A Figura 2.3 apresenta os ambientes de desenvolvimento e operação com os seus componentes interligados por uma infra-estrutura de integração. Pode-se observar a interseção existente entre componentes requeridos nas fases de desenvolvimento e operação. Assim, praticamente todos os componentes são utilizados nas duas atividades,

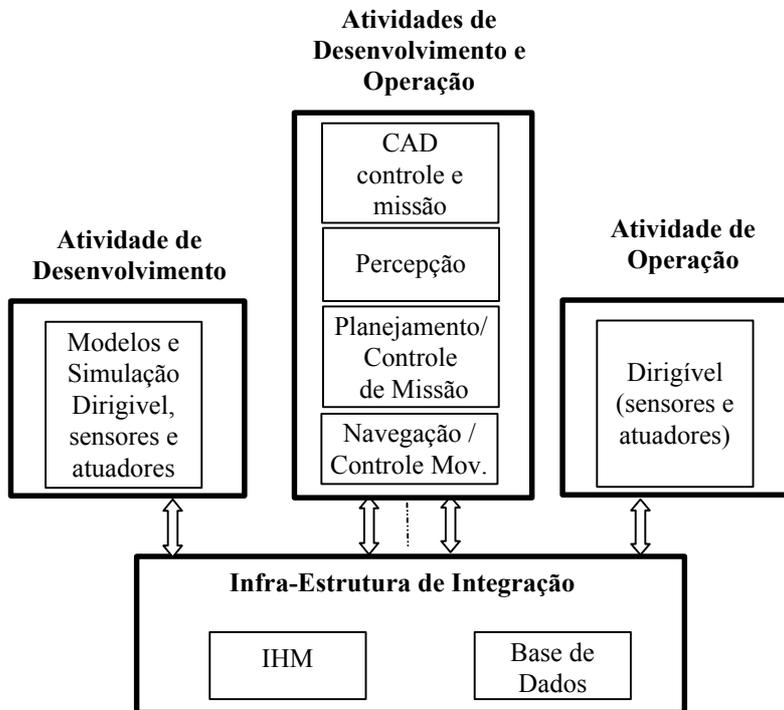


Figura 2.4: Ambiente de desenvolvimento e operação.

A Figura 2.4, apresenta o conceito do ambiente de desenvolvimento e operação para o dirigível robótico. As atividades de desenvolvimento ou de operação podem usar diferentes instâncias desses componentes. Por exemplo, enquanto um usuário ajusta parâmetros de algoritmos em vãos simulados, outro usuário utiliza o mesmo recurso do sistema mas em vôo real; seja em um caso, seja no outro, os componentes são praticamente os mesmos.

### 2.4.1 Mapeamento do Modelo Conceitual para um Dirigível Robótico

Para incrementar a habilidade de operação autônoma do dirigível há duas alternativas básicas de arquitetura:

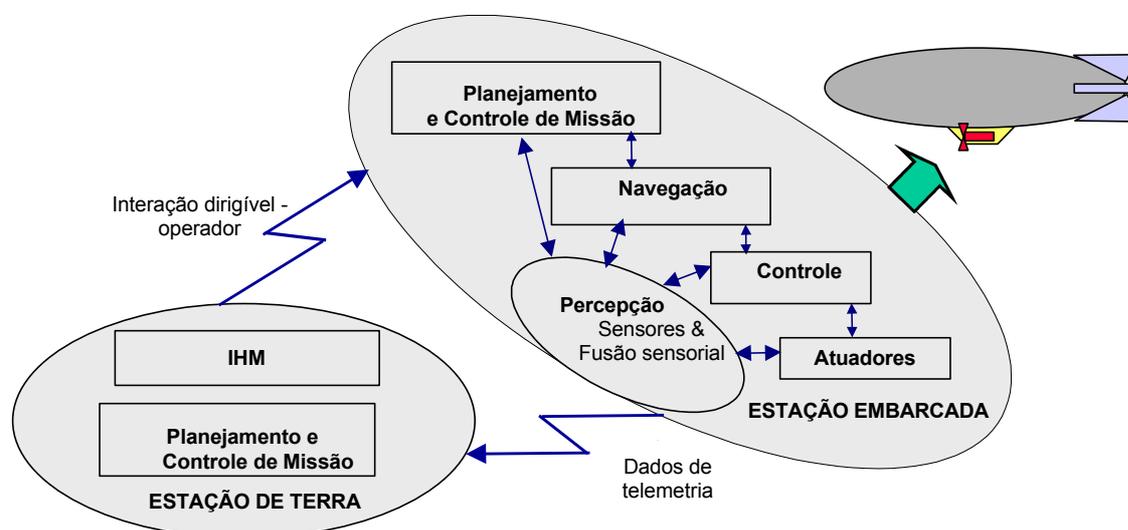


Figura 2.5: Componentes e inter-relacionamentos para um dirigível robótico.

i) Processamento concentrado na estação de terra: os dados sensoriais obtidos na estação embarcada são enviados para a estação de terra, que os processa e envia comandos de movimentação do dirigível para a estação embarcada de forma a realizar a missão atribuída. Neste caso, a “inteligência” do sistema concentra-se na estação de terra; a estação embarcada, devido à menor necessidade de processamento, pode apresentar menores dimensões, peso e consumo de energia. No entanto, um grande inconveniente é a pouca autonomia decisória da estação embarcada, um aspecto preocupante no caso de falha de comunicação, quando a estação embarcada teria a atribuição de modificar a missão em curso e/ou conduzir o veículo para uma situação segura.

ii) Estação embarcada com elevada capacidade de processamento: a Estação embarcada tem capacidade para tratar os dados sensoriais, de diferentes níveis de

complexidade e em diferentes níveis de abstração, e de comandar a movimentação do dirigível de acordo com a missão em curso.

Considerando a característica multisensorial a bordo do dirigível, inclusive a sua evolução para graus de autonomia crescentes, em termos de controle e navegação baseados em imagens por exemplo, decidiu-se por uma estação embarcada com elevado grau de processamento, guardadas as limitações em termos de peso e consumo de energia. Assim, os componentes que congregam as capacidades de autonomia são estruturados conforme mostrado na Figura 2.5.

## 2.5 Arquitetura Funcional

Na Seção 2.2.1, apresentou-se o dirigível robótico como sendo constituído de três elementos básicos: a estação embarcada montada no dirigível; a estação de terra para operar remotamente o veículo, e o sistema de comunicação entre ambos. Nas seções subsequentes, foram relacionados os recursos necessários para a criação do dirigível robótico e estabeleceu-se o modelo conceitual para o seu desenvolvimento e operação.

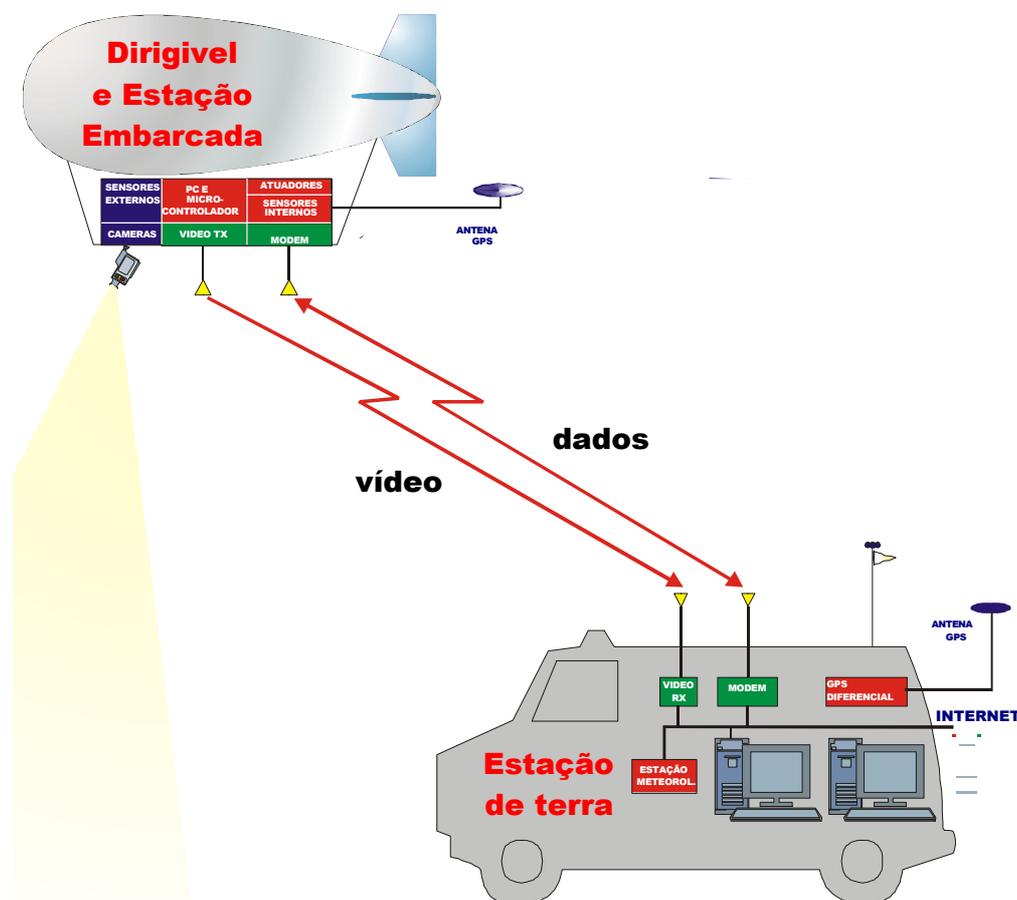


Figura 2.6: Arquitetura funcional do dirigível robótico.

A Figura 2.6 mostra o mapeamento deste modelo conceitual numa arquitetura funcional; ela mostra o dirigível com seus componentes embarcados, a estação de terra com seus componentes, aparecendo também dois enlaces de comunicação: um de vídeo e outro de dados.

Na estação embarcada estão montados os **sensores internos**, que são aqueles necessários à operação autônoma do dirigível robótico; os **sensores externos**, que são específicos para a realização de uma dada aplicação de inspeção ou monitoração aérea; os atuadores que conferem dirigibilidade ao veículo; os computadores embarcados; e rádios para comunicação com a terra. Todos estes componentes devem estar devidamente interligados para assegurar o funcionamento do dirigível como um veículo robótico. Também estão montados na gôndola o sistema de suprimento de energia para alimentação elétrica do conjunto e o reservatório de combustível.

Compõe a estação móvel de terra uma infra-estrutura de processamento e comunicação, montada no local de operação do dirigível ou num veículo que acompanha o deslocamento do dirigível, sendo utilizada para apoiar a realização de missões e prover a interface entre o veículo e operador. Entre os seus componentes estão computadores, rádios para comunicação com a estação embarcada, sensores de apoio ao vôo e recursos para comunicação externa, possibilitando a difusão de dados coletados pelo dirigível quando do desenvolvimento de aplicações.

A arquitetura Funcional é arcabouço para os capítulos subseqüentes, onde se aborda aspectos de desenvolvimento conceitual e implementações. Os componentes básicos de hardware e software e sua integração, constituem os recursos de suporte para o dirigível robótico são os detalhados no Anexo 1.

## 2.6 Conclusões

Apresentou-se neste capítulo a especificação de requisitos para o desenvolvimento do dirigível robótico, o projeto conceitual e a arquitetura funcional adotado para o sistema.

Assim, partindo-se dos componentes básicos de um dirigível robótico, especificou-se as diferentes capacidades a serem integradas para se alcançar um dirigível robótico autônomo, divididas em capacidades de planejamento e controle de missão, navegação / controle de movimentos e percepção; mais os recursos de suporte como *software* de tempo

real, arquitetura de *software* robótico, interface homem-máquina, integração de componentes, ambientes de simulação e apoio ao desenvolvimento e operação e o próprio veículo. Posteriormente realizou-se o mapeamento dessas especificações para as especificações requeridas para esta tese

Apresentou-se também o modelo conceitual de um ambiente de desenvolvimento e operação, composto por componentes que implementam as capacidades requeridas, integrados por uma infra-estrutura de comunicação e apoiados por uma interface homem-máquina e base de dados. Encerrando o capítulo, apresentou-se um mapeamento deste modelo para os componentes básicos do dirigível robótico composto do dirigível, com sua infra-estrutura de sensores e atuadores, e a estação de terra interligados, por um sistema de comunicação, com detalhamento da arquitetura funcional do dirigível. O próximo capítulo inicia a apresentação das bases conceituais associadas a esse trabalho, apresentado o modelo matemático do dirigível.



### 3. MODELO MATEMÁTICO DO DIRIGÍVEL

#### 3.1 Introdução

Modelagem matemática e simulação computacional são utilizadas na análise e no projeto de sistemas de controle para processos não lineares complexos. O objetivo da modelagem é o de expor o funcionamento interno do processo e apresentá-lo numa forma adequada para os estudos que se façam necessários. Assim, num caso particular, ela deve levar em conta o suficiente em termos de atividade do processo de forma a obter um modelo com a precisão requerida, mantendo-o, entretanto, suficientemente simples para permitir a determinação das suas saídas, com os meios disponíveis, em tempos adequados (compromisso simplicidade x precisão).

Em se tratando de veículos aéreos em geral, e dirigíveis em particular, é necessário se dispor de modelos razoavelmente complexos cuja determinação constitui, por si só, tema de pesquisa que extrapola em muito os objetivos desta tese. Optou-se então pelo uso e adaptação ao nosso caso do modelo desenvolvido por GOMES (1990, 1998), que constitui um enfoque reconhecido pelas comunidades acadêmicas e industrial, salientando, antecipadamente, a presença de aproximações nos processos associados às necessárias mudanças de escala entre as aeronaves tratadas por Gomes e no âmbito do presente trabalho.

No modelo matemático obtido por GOMES (1990, 1998), as características aerodinâmicas do dirigível foram obtidas a partir de um extenso conjunto de ensaios em túnel de vento. Numa abordagem alternativa, representada por DELAURIER (1981) e CHELLI (2001), a obtenção dessas características é baseada no uso de técnicas de dinâmica dos fluidos computacional (CFD, do Inglês *Computational Fluid Dynamics*). Segundo KHOURY (1999), o modelo de Gomes constitui um dos mais completos modelos matemáticos de dirigível disponíveis na atualidade.

O capítulo está assim organizado: na Seção 3.2 são mostrados os princípios de operação dos dirigíveis; na Seção 3.3 os sistemas de coordenadas e principais atuadores do dirigível; na Seção 3.4 é apresentado um modelo matemático de dirigível, tal como deduzido por GOMES (1990) para o YEZ-2A, com seus principais termos; na Seção 3.5 é apresentada a particularização do Modelo Matemático para o dirigível AS800 —

plataforma experimental utilizada nesta tese detalhado no Anexo 1 — na Seção 3.6 são apresentados os aprimoramentos realizados neste modelo pelo grupo de pesquisa do AURORA, para a sua adequação às peculiaridades do AS800 e, por fim, na Seção 3.7, como forma de validação, apresentam-se resultados preliminares de comparação do modelo simulado com o veículo real. Finalmente, a Seção 3.8 apresenta as conclusões do capítulo

## 3.2 Princípio de Operação dos Dirigíveis

### 3.2.1 Forças Aerostáticas

Os dirigíveis utilizam como princípio para sua sustentação a denominada sustentação aerostática (MOWFORTH, 1991), isto é, uma sustentação que é independente da velocidade de movimentação, em oposição à sustentação aerodinâmica que depende desta velocidade. O princípio de Arquimedes explica a sustentação aerostática, na qual a força de empuxo  $F$  exercida num corpo é proporcional ao volume ( $vol$ ) de ar deslocado por este e também proporcional à diferença de densidade entre o ar ( $\rho_{ar}$ ) no exterior e o gás ( $\rho_{gás}$ ) no interior do envelope do dirigível, ou  $F = vol [\rho_{ar} - \rho_{gás}]$ , dado que para o gás hélio tem-se  $\rho = 1.06 \text{ Kg/m}^3$  e  $\rho_{AR} = 1,225 \text{ Kg/m}^3$  sob condições ISA a nível do mar. Apesar de mais caro, o gás hélio é o mais usado, visto os riscos de explosão apresentados pelo hidrogênio (MOWFORTH, 1991).

Como muitos dirigíveis são infláveis – também chamados de não-rígidos por não possuírem estrutura – há influência da pressão externa sobre a sua forma. À medida que um dirigível sobe na atmosfera a pressão externa diminui, permitindo a expansão do gás interno, aumentando a tensão no material do envelope, podendo chegar a situações que colocariam em risco a resistência do mesmo. O oposto ocorre nas descidas. Pelo uso de balonetes, que são bolsas de ar dentro do envelope principal que contém o gás de sustentação, o problema é solucionado pela expulsão/admissão de ar de/para os balonetes como mostrado na Figura 3.1. Assim é possível manter-se um diferencial de pressão aceitável com o exterior, considerando-se que a diferença entre a pressão interna e pressão externa deve ser da ordem de 0,5 a 1% (MOWFORTH, 1991).

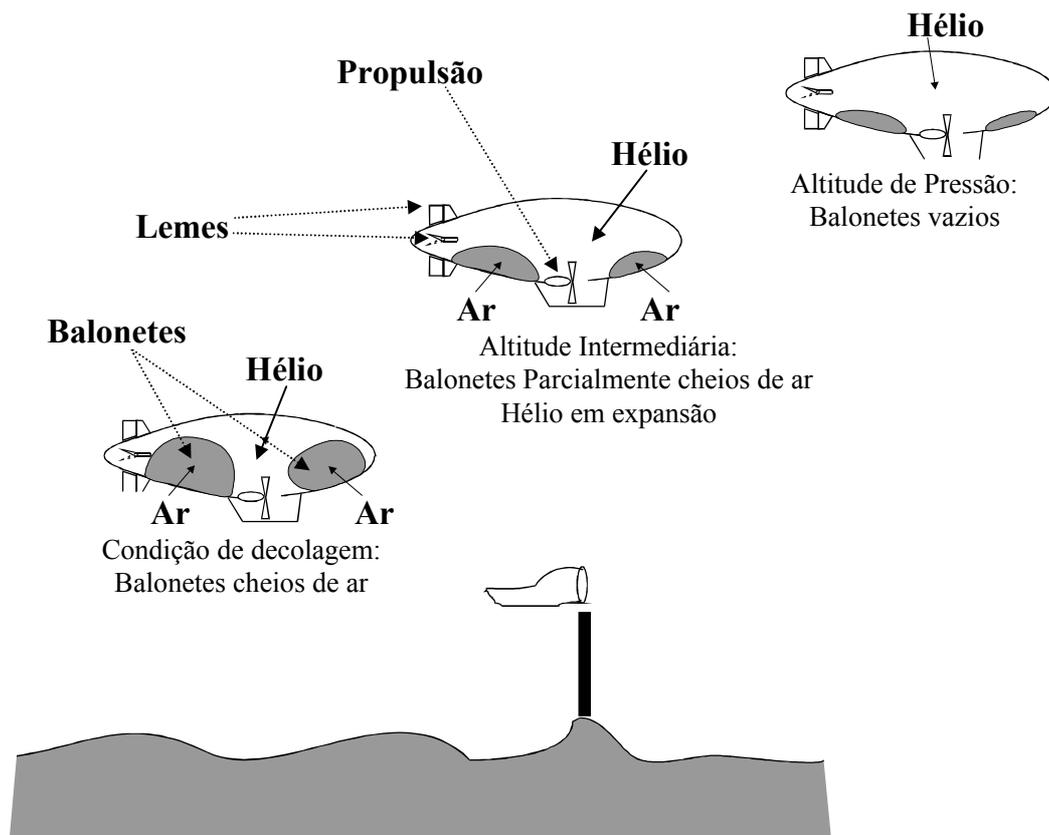


Figura 3.1: Operação dos balonetes num dirigível não rígido.

Caso todo o ar dos balonetes esteja esgotado, a subida será possível apenas com a expulsão do gás de sustentação (o que é oneroso), para garantir o diferencial da pressão. Esta é denominada a altitude de pressão, ou a altitude limite. Usualmente esta é a maior altitude em que o dirigível opera.

### 3.2.2 Forças Aerodinâmicas

As forças aerodinâmicas também atuam nos dirigíveis quando estes se movimentam, distinguindo-se as seguintes forças e momentos (KHOURY, 1999):

- i) arrasto aerodinâmico;
- ii) sustentação aerodinâmica;

iii) a ação das superfícies de atuação aerodinâmicas (lemes ou profundos) mostradas esquematicamente na Figura 3.1 e na Figura 3.2.

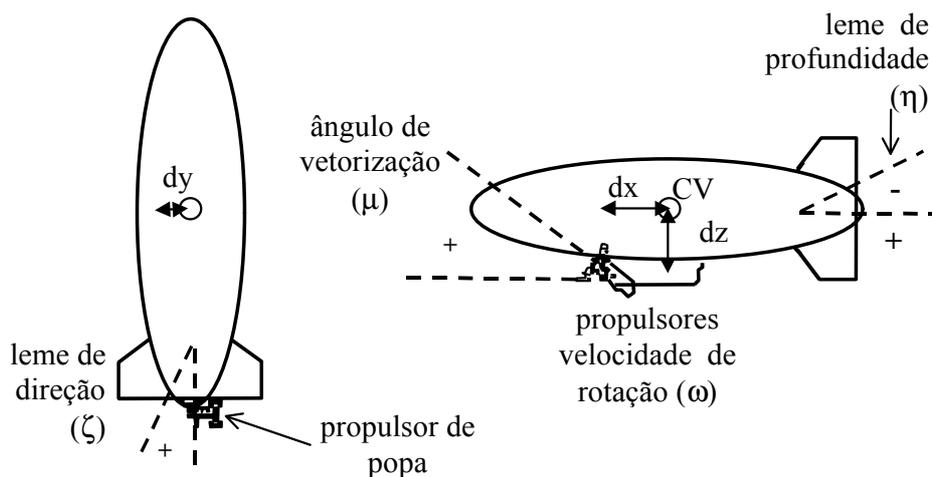


Figura 3.2: Componentes principais para a atuação num dirigível.

### 3.2.3 Forças Dinâmicas

Os principais componentes das forças dinâmicas no dirigível são os associados às massas e inércias virtuais. Eles aparecem de forma significativa em veículos que flutuam num fluido, como submarinos e dirigíveis. Nesses veículos, a massa do fluido por eles deslocada é da ordem de grandeza da massa próprio veículo.

É devido à energia requerida para movimentar o fluido em torno do próprio veículo, quando este se move, que aparecem os efeitos de massas e inércias virtuais. A consequência prática desse é que o veículo aparenta ter massa e inércia maiores que as que ele realmente possui. Em (KHOURY, 1999), encontra-se o desenvolvimento teórico do assunto e é mostrado que, para maioria dos dirigíveis, as massas e inércias virtuais são da ordem de grandeza das massas e inércias do próprio veículo quando em repouso.

### 3.2.4 Forças de Propulsão

Os dirigíveis de grande porte geram suas forças de propulsão através da rotação de hélices acionadas por motores de combustão. Os dirigíveis de pequeno porte também utilizam hélices acionadas por motores elétricos ou motores de combustão baseados em metanol ou gasolina.

Quando são utilizados motores elétricos, a energia provém de baterias ou de células solares. O avião Helios (HELIOS, 2001) é um exemplo de uma aeronave conceitual para ensaiar o uso de veículo aéreo alimentado por células solares (com 62.000 células, 35KW, ocupando uma área de  $137\text{m}^2$ ). O dirigível Lotte é um outro exemplo.

Para facilitar a decolagem e aterrissagem muitos dirigíveis são providos de propulsão vetorizável o que, conforme mostrado na Figura 3.2, é responsável pela variação do ângulo ( $\mu$ ) entre a direção dos eixos de rotação das hélices e a horizontal.

### 3.3 Sistemas de Coordenadas e Principais Atuadores no Dirigível

Nesta seção, define-se inicialmente a notação para o sistema de coordenadas associado ao dirigível e suas variáveis de atuação.

#### 3.3.1 Sistemas de Coordenadas

O sistema de coordenadas  $X, Y, Z$ , associado ao corpo dirigível e denominado SCD, é mostrado na Figura 3.3. O SCD para o dirigível é estabelecido segundo a convenção utilizada em aeronaves (PROUTY, 1986), mas considerando a particularidade de localizar o centro do sistema no centro de volume do dirigível (CV) em lugar de localizá-lo no seu centro de gravidade (CG), o que se justifica pela constante mudança na posição do CG dos dirigíveis devido ao esvaziamento / enchimento dos balonetes para compensar as variações de pressão e manter a forma do envelope (GOMES, 1990). As variáveis introduzidas, associadas ao SCD, são:

- $X, Y, Z$ : os eixos do sistema de coordenadas do dirigível;
- $u, v, w$ : as velocidades lineares relativas aos eixos  $X, Y$  e  $Z$ ;
- $p, q, r$ : as velocidades angulares em torno dos eixos  $X, Y$  e  $Z$ .

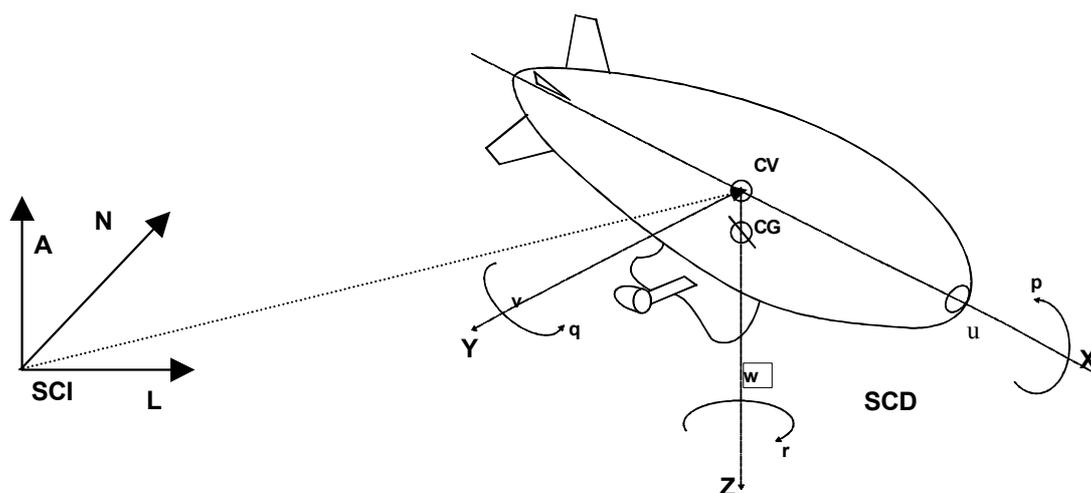


Figura 3.3: Sistemas de coordenadas do dirigível e inercial e principais variáveis associadas.

Se na formulação do modelo matemático do dirigível, como em geral para veículos robóticos, o sistema de coordenadas de referência SCD é localizado no próprio corpo do veículo, por outro lado, na medida que o veículo realiza seu movimento, ele altera sua posição e atitude (orientação) ao longo do tempo, constantemente, também em relação a um segundo sistema de coordenadas, denominado SCI - sistema de coordenadas inercial. O SCI, fixado na terra, permite associar a posição e atitude do veículo a referenciais geodésicos, tais como orientações segundo os eixos Norte, Leste e "Acima" (do Inglês *North, East, Up*), sendo usualmente utilizado nos sistemas de controle e navegação de aeronaves. As transformações entre os referenciais SCD e SCI são abordadas na seqüência.

### 3.3.2 Atuadores no Dirigível

Os principais atuadores de um dirigível e a respectiva notação das variáveis de atuação estão ilustrados na Figura 3.2. Nesta figura, as superfícies de atuação possuem uma configuração em cruz. Os outros elementos de atuação são os propulsores que, além da variação da força de empuxo através da velocidade de rotação dos motores, podem ser vetorizáveis, isto é, ter alterados seus ângulos de orientação com o veículo. Eventualmente, propulsores de popa e / ou proa são montados perpendicularmente ao eixo longitudinal do veículo, para oferecer manobrabilidade adicional em baixas velocidades.

Ao contrário da configuração em "+" para as superfícies de atuação – deflexões  $\zeta$  (do leme de direção) e  $\eta$  (do leme de profundidade), mostrado na Figura 3.2 – o dirigível AS800 possui as superfícies em forma de "x", o que é ilustrado pela Figura 3.4. Os efeitos de ambas configurações sobre o dirigível são explicitados na Seção **A1.2.2**.

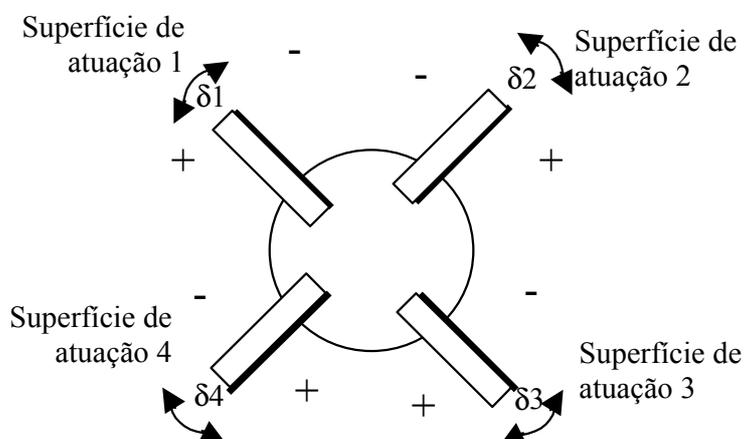


Figura 3.4: Cauda em "x" (visão posterior).

A conversão da atuação em “x” para uma deflexão equivalente em “+”, e vice-versa, é definida pelas Equações 3.1 a 3.3 (GOMES,1990), a seguir, segundo as variáveis introduzidas na Figura 3.2 e Figura 3.4:

$$\eta = (\delta_1 + \delta_2 + \delta_3 + \delta_4) / 4 \quad 3.1$$

$$\zeta = (\delta_1 - \delta_2 + \delta_3 - \delta_4) / 4 \quad 3.2$$

$$\xi = (\delta_1 - \delta_2 - \delta_3 + \delta_4) / 4 \quad 3.3$$

Nota-se que, devido a um grau de liberdade suplementar, além das deflexões  $\zeta$  (do leme de direção) e  $\eta$  (do leme de profundidade), surge uma terceira variável,  $\xi$ , que introduz um efeito de rolagem no dirigível. Usualmente, em dirigíveis rádio controlados, os servos que acionam os pares de aerofólios (1,3) e (2,4) utilizam o mesmo sinal de comando, fazendo com que  $\delta_1 = \delta_3$  e  $\delta_2 = \delta_4$ , implicando neste caso que  $\xi = 0$ , ou seja, não há nenhum comando de rolagem. Nesta condição, tem-se adicionalmente que:

$$\delta_1 = \delta_3 = \eta + \zeta \quad 3.4$$

$$\delta_2 = \delta_4 = \eta - \zeta \quad 3.5$$

No caso do dirigível AS800, cujo sistema de controle é apresentado no Capítulo 4. , não se utiliza o comando de rolagem e  $\xi = 0$ ; no entanto, estudos vêm sendo feitos para utilização deste grau de liberdade suplementar.

### 3.4 Os Termos do Modelo Dinâmico

O modelo dinâmico do dirigível, em seis graus de liberdade, referenciado ao sistema SCD de coordenadas no seu corpo, mostrado na Figura 3.3, é dado por GOMES (1990):

$$M \dot{x} = F = F_d(x) + A(x) + G + P \quad 3.6$$

onde "M" é a matriz de massa, "x" o vetor de velocidades lineares e angulares, "F<sub>d</sub>" o vetor de forças e momentos dinâmicos, "A" o vetor de forças e momentos aerodinâmicos, "G" o vetor de forças e momentos gravitacionais e "P" o vetor de forças e momentos de propulsão.

Cada um dos componentes do modelo será descrito a seguir.

### 3.4.1 Vetor Velocidade

O vetor velocidade "x" contém as três velocidades lineares [u, v, w] e as três velocidades angulares [p, q, r], todas escritas com respeito ao sistema de coordenadas SCD fixado no corpo do dirigível:  $x^T = [u, v, w, p, q, r]$ .

Com objetivo de desenvolvimento de estratégias de controle e navegação, bem como de simulação do modelo, as velocidades são transformadas do sistema de coordenadas no dirigível - SCD - para o sistema de coordenadas inercial - SCI, no qual o vetor de velocidades é denotado por  $x_{SCI}^T = [u_N, v_L, w_A, p_N, q_L, r_A]$ , respectivamente associados às direções Norte, Leste e "Acima".

Para as componentes lineares das velocidades, a matriz de transformação entre os sistemas SCD (i.e. u, v, w) e SCI (i.e.  $u_N, v_L, w_A$ ), é chamada MCD – Matriz dos Cosenos Diretores (CRAIG,1989; GOMES,1990), definida pela Equação 3.7, onde os coeficientes  $\lambda_{ij}$  são denominados parâmetros dos cosenos diretores; a transformação propriamente dita é dada pela Equação 3.8:

$$\text{MCD} = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} \\ \lambda_{21} & \lambda_{22} & \lambda_{23} \\ \lambda_{31} & \lambda_{32} & \lambda_{33} \end{bmatrix} \quad 3.7$$

$$[u_N, v_L, w_A]^T = \text{MCD} [u, v, w]^T \quad 3.8$$

Para as componentes angulares das velocidades, a transformação entre os sistemas SCD e SCI é mais complexa, sendo relacionada com a representação da orientação utilizada. Assim, a taxa de variação da MCD associada às velocidades "p, q, r" é dada pela Equação 3.9, onde  $d\lambda_{ij}/dt$  representa a taxa de variação no tempo para cada um dos componentes da matriz dos cosenos diretores (TITTERTON, 1997).

$$\begin{aligned} d\lambda_{11}/dt &= r \lambda_{12} - q \lambda_{13}, & d\lambda_{12}/dt &= p \lambda_{13} - r \lambda_{11}, & d\lambda_{13}/dt &= q \lambda_{11} - r \lambda_{12}, \\ d\lambda_{21}/dt &= r \lambda_{22} - q \lambda_{23}, & d\lambda_{22}/dt &= p \lambda_{23} - r \lambda_{21}, & d\lambda_{23}/dt &= q \lambda_{21} - r \lambda_{22}, \\ d\lambda_{31}/dt &= r \lambda_{32} - q \lambda_{33}, & d\lambda_{32}/dt &= p \lambda_{33} - r \lambda_{31}, & d\lambda_{33}/dt &= q \lambda_{31} - r \lambda_{32}. \end{aligned} \quad 3.9$$

### 3.4.2 Matriz de Massa

A matriz de massa, "M", incorpora todas as massas e inércias do dirigível, incluindo os termos de massas e inércias virtuais – associados ao fato de estar se tratando

de um veículo flutuante – mais a massa variável dos balonetes. Para o dirigível considerado em GOMES (1990) ou similares, a matriz é dada por:

$$M = \begin{bmatrix} m - X_{\ddot{u}} & 0 & 0 & 0 & m a_x - X_{\dot{q}} & 0 \\ 0 & m - Y_{\ddot{v}} & 0 & -m a_z - Y_{\dot{p}} & 0 & m a_x - Y_{\dot{r}} \\ 0 & 0 & m - M_{\ddot{w}} & 0 & -m a_x - Z_{\dot{q}} & 0 \\ 0 & -m a_z - Y_{\ddot{v}} & 0 & I_x - L_{\dot{p}} & 0 & -J_{xz} \\ -m a_z - M_{\ddot{u}} & 0 & m a_x - X_{\ddot{w}} & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & m a_x - N_{\ddot{v}} & 0 & -J_{xz} & 0 & I_z - N_{\dot{r}} \end{bmatrix} \quad 3.10$$

onde:

$m$  é a massa do dirigível;

$$m_X = m - X_{\ddot{u}}; \quad m_Y = m - Y_{\ddot{v}}; \quad m_Z = m - Z_{\ddot{w}};$$

$X_{\ddot{u}}$ ,  $Y_{\ddot{v}}$ ,  $Z_{\ddot{w}}$  são os termos de massa virtual nos eixos X, Y e Z;

$I_X$ ,  $I_Y$  e  $I_Z$  são momentos de inércia em relação aos eixos X, Y e Z;

$L_{\dot{p}}$ ,  $M_{\dot{q}}$  e  $N_{\dot{r}}$  são os termos de inércia virtual em relação aos eixos X, Y e Z;

$$J_X = I_X - L_{\dot{p}}; \quad J_Y = I_Y - M_{\dot{q}}; \quad J_Z = I_Z - N_{\dot{r}};$$

$$J_{XZ} = I_{XZ} + N_{\dot{p}} = I_{XZ} + L_{\dot{r}}$$

onde;

$I_{XZ}$  é o produto de inércia em relação ao eixo Y, enquanto que  $N_{\dot{p}}$  e  $L_{\dot{r}}$  são termos de inércia virtual;

$a_x$  e  $a_z$  são as coordenadas do CG em relação ao CV nos eixos X e Z; elas são calculadas a cada iteração de simulação, considerando a distribuição das massas no veículo, inclusive as dos balonetes;

$X_{\dot{q}}$ ,  $Y_{\dot{p}}$ ,  $Y_{\dot{r}}$  e  $Z_{\dot{q}}$  são termos de massa virtual e são considerados desprezíveis (GOMES, 1990);

$L_{\dot{v}}$ ,  $M_{\dot{u}}$ ,  $M_{\dot{w}}$  e  $N_{\dot{v}}$  são termos de inércia virtual e são considerados desprezíveis (GOMES, 1990);

Os termos de massa virtual e inércia virtual podem ser estimados conforme descrito em GOMES (1990) usando as relações:

$$X_{\ddot{u}} = -k_1 B/g; \quad Y_{\ddot{v}} = -k_2 B/g; \quad Z_{\ddot{w}} = Y_{\ddot{v}};$$

$$M_{\dot{q}} = -k' [B/g] [(l^2 + d^2) / 20]; \quad N_{\dot{r}} = M_{\dot{q}};$$

onde "B" é a força de flutuação, "g" a aceleração da gravidade, " $k_1$ " e " $k_2$ " são os fatores de inércia de Lamb para movimentos ao longo dos eixos longitudinal (X) e lateral (Y), " $k'$ " é o coeficiente de inércia de Lamb para rotações ao longo do eixo lateral (Y), e "l" e "d" são o comprimento e diâmetros máximos do dirigível respectivamente.

### 3.4.3 Vetor de Forças Dinâmicas

O vetor de forças dinâmicas, "F<sub>d</sub>", contém os termos de Coriolis e centrífugos do modelo dinâmico e é dado por:

$$F_d = [f_1, f_2, f_3, f_4, f_5, f_6]^T \quad 3.11$$

onde:

$$f_1 = -m_Z w q + m_Y r v + m [a_X (q^2 + r^2) - a_Z r p] ;$$

$$f_2 = -m_X u r + m_Z p w + m [-a_X p q - a_Z r q] ;$$

$$f_3 = -m_Y v p + m_X q u + m [-a_X r p + a_Z (q^2 + p^2)] ;$$

$$f_4 = -(J_Z - J_Y) r q + J_{XZ} p q + m a_Z (u r - p w) ;$$

$$f_5 = -(J_X - J_Z) p r + J_{XZ} (r^2 - p^2) + m [a_X (v p - q u) - a_Z (w q - r v)] ;$$

$$f_6 = -(J_Y - J_X) q p - J_{XZ} q r + m [-a_X (u r - p w)].$$

### 3.4.4 Efeito do Vento

Como uma contribuição deste trabalho de tese ao modelo de GOMES (1990), foi introduzido o efeito do vento, que é a principal perturbação que afeta o dirigível em vôo e, por conseguinte, influencia no projeto do sistema de controle e navegação do veículo.

Usualmente, o vento é definido no referencial inercial SCI e suas componentes de velocidade (linear e angular) são transformadas para o referencial SCD através das Equações 3.8 e 3.9 respectivamente.

Considerando-se disponíveis as velocidades do veículo relativas ao referencial SCD ( $x^T = [u, v, w, p, q, r]$ ) e a velocidade do vento em relação a este mesmo referencial ( $x_w^T = [u_w, v_w, w_w, p_w, q_w, r_w]$ ), a velocidade do veículo relativa ao ar é dada pela Equação 3.12:

$$x_a = x - x_w$$

$$x_a = \begin{bmatrix} u_a \\ v_a \\ w_a \\ p_a \\ q_a \\ r_a \end{bmatrix} = \begin{bmatrix} u - u_w \\ v - v_w \\ w - w_w \\ p - p_w \\ q - q_w \\ r - r_w \end{bmatrix} \quad 3.12$$

Uma definição importante é a velocidade linear total relativa ao ar,  $v_{ar\_tot}$ , que é o módulo de  $x_a$ , dado pela equação 3.13

$$v_{ar\_tot} = (u_a^2 + v_a^2 + w_a^2)^{0,5} \quad 3.13$$

Assim, com a obtenção da velocidade do veículo relativa ao ar, este valor é utilizado para calcular as forças aerodinâmicas, conforme é apresentado na próxima seção.

### 3.4.5 Vetor de Forças Aerodinâmicas

O vetor de forças aerodinâmicas, "A", contém os termos aerodinâmicos do modelo, oriundos do corpo do dirigível e das superfícies de controle aerodinâmico. Ele é calculado a partir dos coeficientes aerodinâmicos adimensionais de sustentação ( $c_L$ ), arrasto ( $c_D$ ), forças laterais ( $c_Y$ ), dos momentos de arfagem ou *pitching* ( $c_m$ ), guinada ou *yawing* ( $c_n$ ) e rolagem ou *rolling* ( $c_l$ ), como:

$$A = F(c_i) = [A_X, A_Y, A_Z, A_L, A_M, A_N]^T; \quad 3.14$$

onde  $c_i$  são os coeficientes acima citados;  $A_X$ ,  $A_Y$  e  $A_Z$  são as forças de arrasto, lateral e de sustentação;  $A_L$ ,  $A_M$  e  $A_N$  são os momentos de rolagem, arfagem e guinada respectivamente.

Os coeficientes adimensionais  $c_i$  seguem a convenção padrão para dirigíveis:

$$\begin{aligned} c_D &= A_X / (0.5 \rho v_{ar\_tot}^2 V^{2/3}); \\ c_Y &= A_Y / (0.5 \rho v_{ar\_tot}^2 V^{2/3}); \\ c_L &= A_Z / (0.5 \rho v_{ar\_tot}^2 V^{2/3}) \\ c_l &= A_L / (0.5 \rho v_{ar\_tot}^2 V) \\ c_m &= A_M / (0.5 \rho v_{ar\_tot}^2 V) \\ c_n &= A_N / (0.5 \rho v_{ar\_tot}^2 V) \end{aligned} \quad 3.15$$

onde:  $V$  é o volume do dirigível (em  $m^3$ ),  $v_{ar\_tot}$  é a velocidade relativa ao ar (em m/s) e  $\rho$  é a densidade do ar (em  $Kg/m^3$ ).

Estes coeficientes aerodinâmicos podem ser obtidos:

- i) por medidas diretas em túnel de vento;
- ii) a partir das características geométricas do veículo (UPSON, 1925);
- iii) a partir das derivadas de estabilidade aerodinâmicas.

No presente caso, eles foram determinados diretamente por extensos testes em túnel de vento realizados por Gomes para ângulos de incidência de vento  $\alpha$  e  $\beta$  na faixa de  $-30^\circ$  a  $+30^\circ$ , conforme a Figura 3.5, e organizados em tabelas, uma para cada coeficiente aerodinâmico -  $c_D$ ,  $c_Y$ ,  $c_L$ ,  $c_l$ ,  $c_m$ ,  $c_n$ . Cerca de 120 tabelas assim determinadas são disponíveis em (GOMES, 1990).

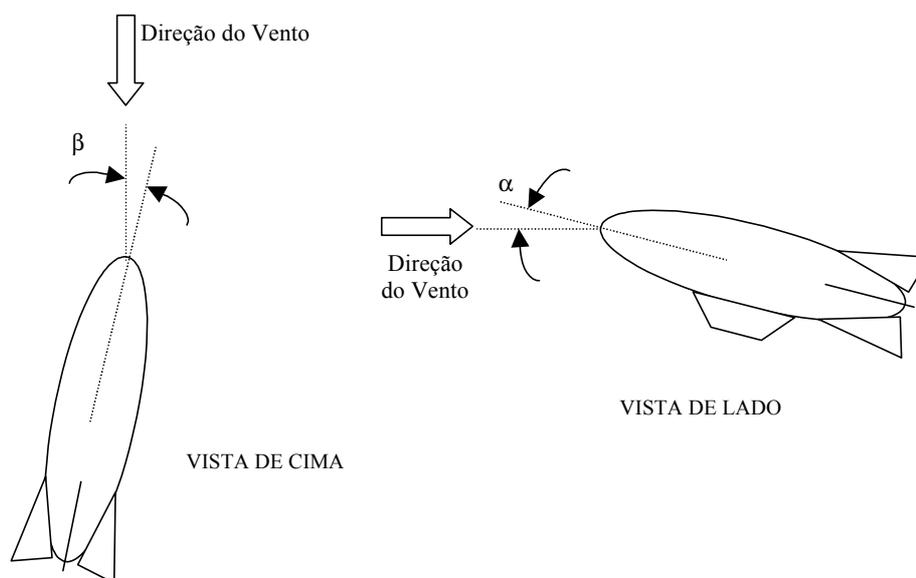


Figura 3.5: Ângulos de incidência  $\alpha$  e  $\beta$ .

Os ângulos de incidência  $\alpha$  e  $\beta$ , mostrados na figura 4 para a situação particular de ensaios em túnel de vento, têm uma caracterização mais genérica quando o dirigível realiza vôos em ambientes reais. Neste caso, os dois ângulos são determinados pela Equação 3.16 a partir das componentes lineares da velocidade relativa ao ar nos três eixos do sistema SCD ( $u_a$ ,  $v_a$ ,  $w_a$ ) e da velocidade linear total relativa ao ar ( $v_{ar\_tot}$ ), definidas pela Equação 3.12 e pela Equação 3.13:

$$\alpha = \text{tg}^{-1}(u_a/w_a) \quad \beta = \text{tg}^{-1}(u_a/v_{ar\_tot}) \quad 3.16$$

Para o acesso às tabelas, são necessários, como dados de entrada, os valores dos comandos das superfícies aerodinâmicas em cruz, pois estas foram construídas segundo este sistema (vide Figura 3.4 e Equações 3.1 a 3.3) - ou seja, os comandos  $\eta$  do leme de profundidade,  $\zeta$  do leme de direção e o efeito resultante de rolagem  $\xi$  bem como os ângulos de ataque  $\alpha$  e de incidência lateral  $\beta$ , obtidos a partir da velocidade relativa ao ar nos três eixos do sistema SCD definidos pela Equação 3.16. A partir desses valores obtém-se das tabelas os coeficientes aerodinâmicos -  $c_D$ ,  $c_Y$ ,  $c_L$ ,  $c_l$ ,  $c_m$ ,  $c_n$ . ou seja,  $c_i = f(\alpha, \beta, \eta, \zeta, \xi)$ .

Então, a partir destes coeficientes, através da Equação 3.15, são calculados as forças  $A_X$ ,  $A_Y$  e  $A_Z$  e os momentos  $A_L$ ,  $A_M$  e  $A_N$

### 3.4.6 Vetor de Gravidade e Flutuação

O vetor “G” representa a diferença entre os vetores associados ao peso do dirigível, “W”, e a força de flutuação, “B”, agindo para cima, transformados para os eixos do dirigível pelos parâmetros dos cosenos diretores  $\lambda_{ij}$ , da matriz MCD (Equação 3.7) e simplificado com base na simetria do veículo com relação ao plano XZ (GOMES, 1990).

$$G = \begin{bmatrix} \lambda_{31}(W - B) \\ \lambda_{32}(W - B) \\ \lambda_{33}(W - B) \\ \lambda_{32} a_z W \\ (\lambda_{32} a_z - \lambda_{33} a_x)W \\ \lambda_{32} a_x W \end{bmatrix} \quad 3.17$$

onde:  $\lambda_{ij}$  são elementos da matriz dos cosenos diretores – MCD, “W” é o peso do dirigível agindo no centro de gravidade CG, “B” é a força de flutuação agindo no centro de Volume – CV – e  $a_x$  e  $a_z$  são as coordenadas do CG nos eixos X e Z respectivamente.

### 3.4.7 Modelo da Força de Propulsão

Como uma evolução aos trabalhos de GOMES (1990), para a modelagem dinâmica do dirigível tornou-se necessária nesta tese a obtenção de um modelo representativo para a força de propulsão, enfocando-se explicitamente o conjunto hélice-motor do AS800. Os principais motivos para tal foram: i) a necessidade de seleção de conjunto hélice-motor com característica de propulsão adequada; ii) o uso do modelo no processo de concepção do sistema de controle.

O vetor "P", que contém os termos de força e momento no sistema de coordenadas do dirigível, é obtido pela equação 3.18. Para isto são requeridos:

- i) as forças de propulsão do motor esquerdo ( $T_{esq}$ ) e motor direito ( $T_{dir}$ ),
- ii) o ângulo de vetorização  $\mu$  do sistema de propulsão (ver Figura 3.2),
- iii) as distâncias  $dx$  (no eixo X),  $dy$  (no eixo Y) e  $dz$  (no eixo Z) da posição do conjunto propulsor ao centro de volume (CV) do dirigível (ver Figura 3.2).

$$P = \begin{pmatrix} (T_{dir} - T_{esq}) \cos(\mu) \\ 0 \\ -(T_{dir} + T_{esq}) \sin(\mu) \\ (T_{dir} - T_{esq}) dy \cos(\mu) \\ (T_{dir} + T_{esq}) (dz \cos(i) - dy \sin(\mu)) \\ (T_{dir} - T_{esq}) dy \cos(\mu) \end{pmatrix} \quad 3.18$$

O modelo matemático da força de propulsão foi obtido através do levantamento das características das hélices, conforme se mostra no Anexo 2. Ele consiste de uma tabela do chamado coeficiente de propulsão " $c_T$ ", característico da hélice, que é obtido a partir das informações de velocidade do veículo em relação ao ar,  $v_{ar\_tot}$ , e da velocidade de rotação dos motores " $n$ "; a partir destes são obtidas as forças de propulsão atuando no veículo (mais detalhes no Anexo 2).

Assim, para a Equação 3.6, em termos de simulação, o vetor "P", de propulsão, é determinado através dos seguintes passos:

1. define-se a velocidade de rotação dos motores esquerdo e direito como uma percentagem da velocidade máxima permitida pelo motor (sendo que nesta situação não se considera nem a dinâmica nem a resposta do motor de combustão, já que não se dispunha da curva de potência destes e não se propôs levantar estes dados no presente trabalho);
2. obtém-se os coeficientes de propulsão " $c_T$ " direito e esquerdo interpolando-se na tabela  $c_T$  função da razão de avanço (apresentado no Anexo 2),  $\gamma$ , dada pela Equação 3.19, onde  $v_{ar\_tot}$  é a velocidade total do veículo relativa ao ar (m/s),  $n$  é a velocidade de rotação da hélice (1/s) e  $D$  é o diâmetro da hélice;

$$\gamma = \frac{v_{ar\_tot}}{n D} \quad 3.19$$

3. calcula-se as forças de propulsão esquerda ( $T_{esq}$ ) e direita ( $T_{dir}$ ) a partir dos coeficientes de propulsão  $c_T$  de cada lado, utilizando a Equação 3.20, onde  $\rho$  é a densidade do ar em  $Kg/m^3$ .

$$T = c_T \rho n^2 D^4 \quad 3.20$$

### 3.5 Particularização do Modelo Matemático para o Dirigível AS800

O modelo original desenvolvido por GOMES (1990), destinava-se ao dirigível YEZ-2A com aproximadamente 130m de comprimento e capacidade de carga de 10 toneladas. A migração dos parâmetros do modelo matemático original para o modelo AS800, realizada principalmente por DE-PAIVA (1998), consistiu:

- Na adequação das massas envolvidas para as massas do AS800: pela realização das medições de todas as massas e tabulação de suas distâncias ao centro de volume CV.
- Na adequação das dimensões e volume para aquelas do AS800.
- No cálculo de novos momentos de inércia e centro de gravidade a partir dos valores das diferentes massas e de suas posições em relação ao centro de volume;
- Na adequação das forças de propulsão para o conjunto hélice-motor conforme a Seção 3.4.7.

A Tabela 3.1 mostra alguns dos parâmetros do AS800 comparados com os do YEZ-2A.

Parâmetro	AS800-1	YEZ-2A
Massa de Hélio (Kg)	3,67	14.222
Volume de Hélio (m <sup>3</sup> )	24	
Massa total (Kg)	26,5	86.816
Momentos de Inércia em relação ao SCD (Kg m <sup>2</sup> )	$I_X = 54,1$ (em torno do eixo X) $I_Y = 108,4$ (em torno do eixo Y) $I_Z = 95,62$ (em torno do eixo Z) $I_{XZ} = 0,33$ (produto de inércia)	$I_X = 1.247.078$ $I_Y = 55.472.495$ $I_Z = 49.248.564$ $I_{XZ} = 1.648.250$
Coordenadas do CG no SCD (SCD localizado no CV) (m)	$a_X = 0,01$ $a_Y = 0,0$ $a_Z = 0,74$	$a_X = -0,13$ $a_Y = 0,0$ $a_Z = 10,43$
Força de Propulsão (N)	80	10.700
Diâmetro (m)	2,26	32
Comprimento (m)	9,5	129,5
Relação comprimento/diâmetro	4,2 : 1	4,04 : 1

Tabela 3.1: Parâmetros do AS800 e do YEZ-2A

Nota-se na tabela a relação comprimento/diâmetro, que é aproximadamente igual em ambos dirigíveis a despeito da enorme diferença dimensional entre ambos. É justamente esta coincidência que permite que as adaptações acima tenham sido efetuadas e ao mesmo tempo mantidos inalterados os coeficientes aerodinâmicos  $-c_D$ ,  $c_Y$ ,  $c_L$ ,  $c_l$ ,  $c_m$ ,  $c_n$  e sua obtenção das tabelas como  $c_i = f(\alpha, \beta, \eta, \zeta, \xi)$ , conforme explicado anteriormente.

### 3.6 Aprimoramentos no Modelo Matemático do Dirigível AS800

Em AZINHEIRA (2001c) são apresentadas as modificações de aprimoramento do modelo decorrentes de observações realizadas sobre o veículo real (AS800). Destacam-se:

#### 3.6.1 *Extrapolação dos Coeficientes Aerodinâmicos para Ângulos Maiores que 30°*

Os coeficientes aerodinâmicos foram levantados por GOMES (1990) em ensaios de túnel de vento para ângulos de incidência  $(\alpha, \beta)$  na faixa de  $-30^\circ$  a  $+30^\circ$  e considerando os comandos  $(\eta, \zeta, \xi)$ . Tomando como base este conjunto de tabelas de dados, uma formulação do modelo aerodinâmico foi desenvolvida considerando-se expressões analíticas para os coeficientes aerodinâmicos. Estes aperfeiçoamentos realizados por Azinheira são descritos em AZINHEIRA (2001c).

As principais evoluções desta nova formulação do modelo em relação ao anterior são:

- i) adaptação das tabelas de coeficientes em função de 5 variáveis  $(\alpha, \beta, \eta, \zeta, \xi)$  para expressões analíticas,
- ii) extensão dos coeficientes aerodinâmicos para ângulos de ataque elevados (90 graus para  $\alpha$  e 180 graus para  $\beta$ ).

O processo de transformação das tabelas para expressões analíticas foi realizado desenhando-se funções bidimensionais  $f(\alpha, \beta)$  dos coeficientes aerodinâmicos oriundos das tabelas de dados, seguido de um procedimento de ajuste de curvas e extração dos parâmetros relevantes de 1ª e 2ª ordens. As funções simplificadoras,  $f(\alpha, \beta)$ , mostradas na Equação 3.21, foram validadas com relação aos dados originais. Erros residuais existem em virtude das simplificações utilizadas, e seus ajustes finais serão realizados através dos ensaios de identificação a serem implementados.

$$c_i = f(\alpha, \beta) \text{fsen}(\alpha, \beta) c_m ; \quad 3.21$$

onde: " $c_i$ " corresponde aos coeficientes  $c_D$ ,  $c_Y$ ,  $c_L$ ,  $c_l$ ,  $c_m$ ,  $c_n$ , " $f(\alpha, \beta)$ " à função de interpolação, " $\text{fsen}(\alpha, \beta)$ " ao ajuste para extrapolação e " $c_m$ " ao comando considerado ( $\eta$ ,  $\zeta$ ,  $\xi$ ).

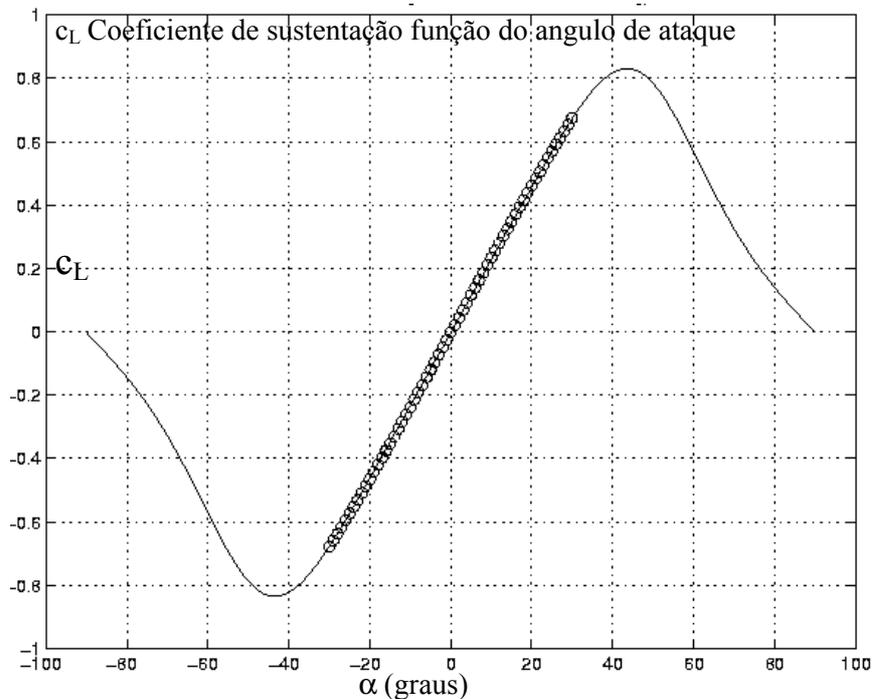


Figura 3.6: Curva do coeficiente de sustentação ( $c_L$ ) em função de  $\alpha$ , com  $\beta=20^\circ$ ,  $\eta=\zeta=\xi=0$ .

Para contemplar movimentos do dirigível em situações onde os ângulos de incidência assumem valores além da faixa original, como em ascensões e descidas verticais, ou próximas delas, ou em situações de vôo pairado, realizou-se a extrapolação destes coeficientes. O processo de extrapolação para altos ângulos de ataque foi realizado adicionando-se termos multiplicativos às funções simplificadoras. Estes termos são funções senoidais, " $\text{fsen}$ " na Equação 3.21, com propriedades particulares que garantem que para pequenos ângulos de incidência os valores resultantes dos coeficientes aerodinâmicos correspondam aos valores originais, enquanto que para ângulos mais altos, estas exibam um comportamento desejável em termos de simetria e saturação. A validação e refinamento destes termos de extrapolação também será realizada através dos ensaios de identificação.

A Figura 3.6 apresenta a curva para o coeficiente de sustentação " $c_L$ ", função do ângulo de ataque " $\alpha$ ", para o ângulo de escorregamento lateral " $\beta$ " igual a  $20^\circ$  e " $\eta$ ,  $\zeta$ , e  $\xi$ "

iguais a zero. Observa-se que os pontos marcados correspondem aos dados originalmente obtidos por GOMES (1990), enquanto que a linha contínua corresponde à função para extrapolação dos valores apresentada em AZINHEIRA (2001c), observando-se uma superposição entre os dados obtidos por GOMES (1990) e a função utilizada.

### ***3.6.2 Inclusão do Efeito do Fluxo de Ar Induzido pelos Motores nas Superfícies Aerodinâmicas***

A prática de vôos com o AS800 mostrou que, devido à reduzida distância entre o grupo propulsor e as superfícies aerodinâmicas, a influência do fluxo de ar induzido pelas hélices dos motores nas superfícies é significativa. Avaliações experimentais indicaram que a velocidade do fluxo de ar induzido chega a 20Km/h. Na medida que o ângulo de vetorização do grupo propulsor se aproxima de  $-90^\circ$  ou de  $+90^\circ$  (i.e., eles se encontram perpendiculares ao eixo longitudinal do veículo), este fluxo praticamente se anula.

Para se considerar este fenômeno foi acrescentado um coeficiente de eficiência, função do ângulo de vetorização, da intensidade da força de propulsão e da área dos propulsores, que incrementa as deflexões comandadas  $\zeta$  (do leme de direção),  $\eta$  (do leme de profundidade), e  $\xi$  (o efeito de rolagem) (AZINHEIRA, 2001c).

Este coeficiente de rendimento (rend) é um fator empírico que incrementa a deflexão em função do ângulo de vetorização, conforme a Equação 3.22a.

$$\text{rend} = 0,25 / (1 + (\mu - \mu_v)^2 / \mu_v^2)$$

$$\text{fator} = 0,25 (1 + (1 + \text{rend}(T_{\text{esq}} + T_{\text{dir}}) / (\rho \text{ Sh } v_{\text{ar\_tot}}^2))^{0,5})^2 \quad 3.22a$$

onde  $\mu$  é o ângulo vetorização,  $\mu_v$  é o ângulo de vetorização para maior fluxo de ar gerado sobre a superfície aerodinâmica (normalmente é nulo) e Sh é a área da hélice de diâmetro D.

Assim os valores comandados são transformados para novos valores considerando a influência do ângulo de vetorização:

$$\zeta_M = \zeta (1 + \text{fator})$$

$$\eta_M = \eta (1 + \text{fator})$$

$$\xi_M = \xi (1 + \text{fator}) \quad 3.22b$$

### 3.6.3 Inclusão de um Modelo para a Turbulência Atmosférica

Na Seção 3.4.4 foi introduzido no modelo a representação de ventos com intensidade fixa – ver Equação 3.12. Um novo modelo de interação com o vento foi desenvolvido por Azinheira e Paiva (AZINHEIRA, 2001c) onde, além de uma intensidade fixa, acrescentou-se mais um termo, estocástico, que utiliza a função de Dryden - clássica em aviãoica (MCLEAN,1990).

## 3.7 Avaliação do Modelo Matemático do Dirigível

Conforme explicitado na Introdução do capítulo, tinha-se como objetivo a obtenção de um modelo matemático para o AS-800. A adoção do enfoque introduzido por Gomes para o dirigível YEZ-2A e sua adaptação às características do AS-800, justificam-se pela mesma razão comprimento/diâmetro de ambas aeronaves (vide Tabela 3.1) o que permite que os coeficientes aerodinâmicos sejam mantidos, tendo sido efetuado o escalamento das demais variáveis. O modelo resultante, e os aprimoramentos nele introduzidos, tornam-no qualitativamente representativo do AS-800, apesar de não retratá-lo de forma exata.

No contexto do presente trabalho, diversos vôos foram realizados, alguns deles comentados no Capítulo 8. Com base na prática de vôo, constatou-se que o dirigível comporta-se como quando simulado, exceção feita principalmente às taxas de guinada e arfagem, indicando que o veículo real apresenta uma dinâmica mais rápida que aquela representada pelo modelo. Esta diferença de comportamento é ilustrada a seguir para a dinâmica longitudinal do dirigível (DE-PAIVA, 2001a). A Figura 3.7 apresenta uma seqüência de comandos aplicados ao leme de profundidade do dirigível ( $\eta$ ), tanto no dirigível real quanto no modelo de simulação. A Figura 3.8 reproduz os comportamentos do dirigível e do modelo simulado, para as variáveis  $u$  - velocidade à frente ( $U$  na Figura 3.8),  $w$  - velocidade vertical ( $W$  na Figura 3.8),  $q$  - velocidade de arfagem ( $Q$  na Figura 3.8) e ângulo de arfagem ( $\Theta$  na Figura 3.8); este último caracteriza o ângulo entre o eixo longitudinal do dirigível e a horizontal. Mesmo não existindo condições idênticas em termos de vento, turbulências e variações de velocidade entre o vôo real e o caso simulado, é possível constatar:

- i) uma grande similaridade na evolução das variáveis e;
- ii) uma maior excursão das variáveis no caso real em comparação ao simulado,

representando nesse caso a dinâmica de maior intensidade do primeiro em relação ao segundo, como constatado na prática de voo.

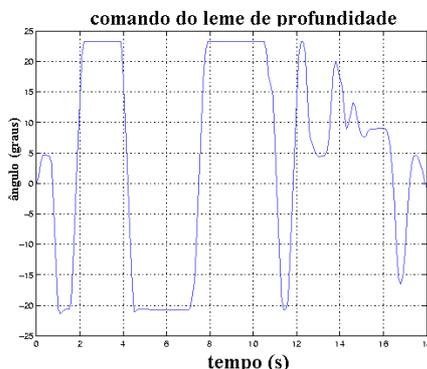
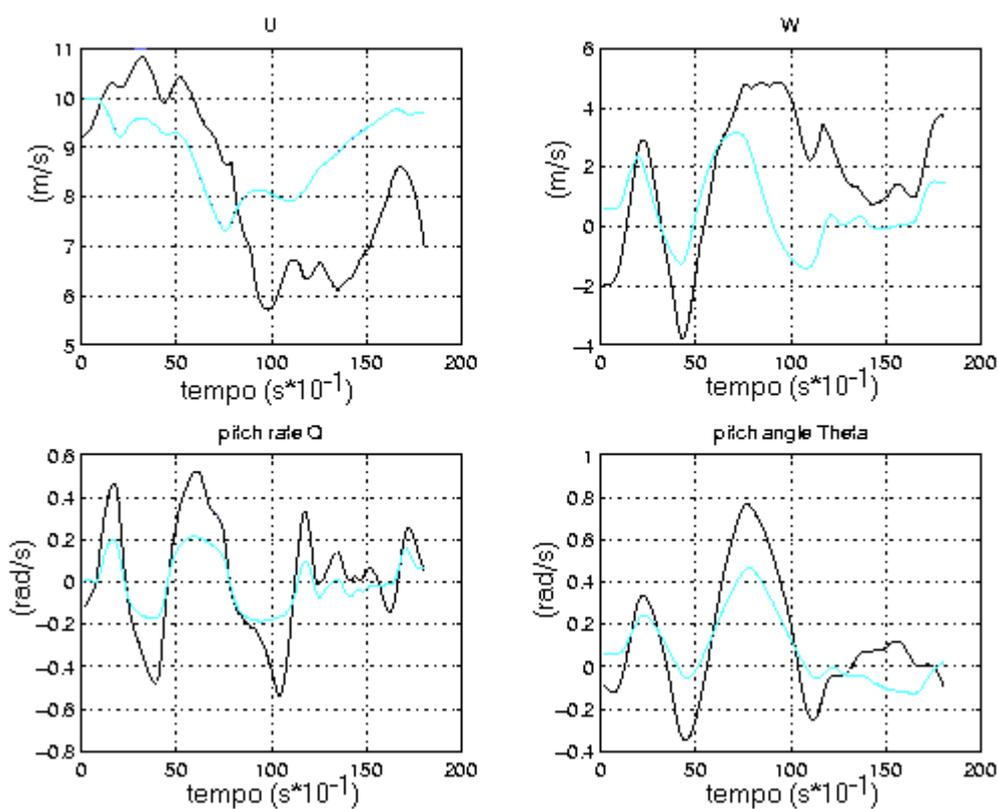


Figura 3.7: Comandos aplicados ao leme de profundidade.



(linha escura real, linha clara simulado)

Figura 3.8: Resultados para mesma entrada de controle.

Assim, para o restante do trabalho, considera-se que se dispõe de um modelo matemático de dirigível qualitativamente representativo da dinâmica do veículo, mas ainda quantitativamente impreciso.

Para tornar o modelo do AS800 quantitativamente correto, são necessários trabalhos adicionais de identificação, ajuste, refinamento e validação do modelo, através de

um procedimento sistemático envolvendo:

- i) o aprimoramento da precisão das medidas das forças de propulsão;
- ii) a realização de ensaios com manobras pré-programadas específicas para excitar separadamente modos do comportamento dinâmico do veículo;
- iii) a aquisição de dados sensoriais do veículo e seu condicionamento;
- iv) o uso de técnicas de identificação para o ajuste de coeficientes estáticos e dinâmicos do modelo do dirigível e;
- iv) a realização de ensaios em túnel de vento, abordando especificamente a forma do envelope do dirigível e a influência das superfícies de atuação nos coeficientes aerodinâmicos.

### **3.8 Conclusões**

Apresentou-se neste capítulo o modelo matemático do dirigível, baseado na formulação desenvolvida por GOMES (1990) que é, reconhecidamente, um dos mais completos enfoques neste domínio. Como resultado deste trabalho de tese, foram realizados aprimoramentos neste modelo, como a inclusão de uma representação mais adequada da força de propulsão e a inclusão do efeito do vento no cálculo das forças aerodinâmicas. Foram apresentados também outros aprimoramentos introduzidos por Paiva e Azinheira, que compreendem a inclusão de um fator empírico para o efeito do vento gerado pelos motores sobre as superfícies aerodinâmicas, a extrapolação dos coeficientes aerodinâmicos para uma faixa de ângulos de incidência mais abrangente e a introdução de um componente estocástico no vento como perturbações.

No final apresentaram-se resultados iniciais de comparação, para o modo longitudinal, entre dados fornecidos pelo modelo e aqueles coletados em voo real do dirigível. Observou-se a proximidade entre ambos os casos, constatando assim a adequabilidade do modelo. Para o restante do trabalho considera-se, portanto, que se dispõe de um modelo matemático de dirigível qualitativamente representativo da dinâmica do veículo, mas ainda quantitativamente impreciso. O refinamento e ajuste do modelo constitui tema de trabalhos futuros, posteriores a esta tese.

O próximo capítulo apresenta aspectos ligados aos Algoritmos de Controle

utilizados no dirigível, que requerem o conhecimento do modelo matemático do dirigível para o seu projeto e ajuste.

## 4. ALGORITMOS DE CONTROLE

### 4.1 Introdução

Conforme especificado na Seção 2.2.2 o dirigível deve ter capacidade de navegação e controle de movimentos, obedecendo a requisitos de estabilizá-lo em seu vôo e, ao mesmo tempo, de executar trajetórias (retilíneas, circulares, etc.) com determinado perfil de altitude, entre pontos de passagem definidos em termos das coordenadas de Latitude e Longitude. Os atuadores disponíveis no AS800 para seu controle, e as denominações das variáveis associadas, são os explicitados na Figura 3.2, da Seção 3.3.2.

Dentre as diversas fases de vôo de dirigíveis, este capítulo se atém exclusivamente ao controle e navegação em vôos na velocidade cruzeiro, por pontos de passagem; ou seja, não são abordadas situações de decolagem, vôo pairado e aterrissagem vertical ou horizontal. Desta forma discute-se na Seção 4.2 a situação das aplicações de sistemas de controle e navegação automáticos para estes veículos, apresentando-se em seguida a perspectiva histórica da aplicação de controle automático em veículos aéreos com o objetivo de sua comparação com a situação atual dos dirigíveis. Esta seção termina com um levantamento de artigos relacionados ao controle de dirigíveis, extraídos das principais periódicos, publicações e conferências na área, resumindo e avaliando o seu conteúdo.

Na Seção 4.3, dando prosseguimento à analogia utilizada na Seção 4.2, são apresentados inicialmente os principais conceitos das técnicas de controle e execução de trajetórias retilíneas e o controle de altitude, utilizados em aviões, helicópteros, mísseis e robôs manipuladores; com base nestes conceitos, aborda-se, então, o problema de controle e navegação de dirigíveis em velocidade de cruzeiro, a técnica de controle utilizada como referência e a técnica proposta para o controle de trajetória dirigíveis, apresentando o diagrama completo do sistema de controle de trajetória e seu mapeamento na infraestrutura embarcada do dirigível. O sistema de controle e navegação do AS800 é baseado nos resultados de estudos desenvolvidos pelos pesquisadores José Raul Azinheira e Ely de Paiva, sendo que nesta tese foram introduzidas modificações tornando-os adequados à implementação e uso reais.

A Seção 4.4 apresenta o problema do controle de altitude partindo da abordagem utilizada em aviões, particularizando-a então para o caso de dirigíveis; esta seção se encerra

com a apresentação do diagrama do sistema de controle de altitude e seu mapeamento para o sistema embarcado. Enfim, a Seção 4.5 apresenta as conclusões do capítulo.

## 4.2 Controle Automático para Dirigíveis

O controle de veículos aéreos tem como objetivos estabilizá-los em seus vôos e fazer com que os vôos sigam padrões de altitude, posição, orientação e velocidade ao longo do tempo, mantendo o curso apesar da presença de perturbações, principalmente as devidas às turbulências que ocorrem no ar. O sistema de controle usa os dados de posição, atitude, velocidades, acelerações do veículo, para determinar os sinais a serem fornecidos aos seus atuadores, de forma a satisfazer os objetivos citados.

Em KHOURY (1999), que através de diversos autores fornece uma análise do estado da arte e tendências em dirigíveis, é relatado que ainda não houve grande progresso no que concerne ao controle automático deste veículos, prevendo-se que é inevitável o uso do computador para o seu controle e principalmente a introdução de piloto automático. Os pilotos automáticos proporcionarão o alívio da carga operacional do piloto, devido às baixas velocidades e grande duração de vôo<sup>2</sup>.

Apresenta-se a seguir um breve resumo histórico da evolução do sistemas de controle automático em outros veículos aéreos objetivando-se uma comparação em relação aos dirigíveis.

### 4.2.1 Técnicas de Navegação e Controle Automático em Outros Veículos Aéreos

Desde o início do século XX, quando os primeiros aviões começaram a voar, até hoje, tem existido conflito entre o que se chama manobrabilidade e estabilidade dessas aeronaves. A estabilidade (ou melhor, a instabilidade própria) diz respeito à existência de alguns modos instáveis como o *dutch roll* (PALLET, 1983), o qual constitui um modo oscilatório instável em torno do eixo longitudinal no veículo, que requer a presença de um piloto para compensá-lo. Na medida em que se estabiliza o veículo, pela introdução de componentes aerodinâmicos, conhecidos como estabilizadores, o veículo fica mais estável porém menos manobrável, ficando mais susceptível às turbulências. Assim faz parte da

---

<sup>2</sup> Por exemplo, o dirigível da Good-Year - *Spirit of Americas* - já levou, conforme depoimento do seu piloto, em torno de 15 horas, em julho de 2001, para fazer o percurso Rio - São Paulo (400Km) com vento contrário

filosofia de projeto dos aviões, deixá-los ligeiramente instáveis no seu projeto aerodinâmico, mas permitindo que sejam manobráveis frente a turbulências, e pilotáveis por seres humanos, eventualmente com a ajuda dos chamados SAS (do Inglês *Stability Augmentation System*, sistema de aumento de estabilidade) (MCLEAN, 1990, BLAKELOCK, 1991).

Desde 1912, quando foi realizado o primeiro vôo com um sistema de controle automático, estabilizando o vôo de um avião em dois eixos, os sistemas de controle automático para esses veículos têm sofrido grande evolução. Assim segundo MCLEAN (1990) e BLAKELOCK (1991):

- Em 1927, um sistema de controle de direção com acionamento pneumático foi testado num Zepelin-LZ127, sendo este sistema chamado de piloto automático;
- Em 1933, Wiley Post realizou a volta ao mundo em 8 dias auxiliado por um sistema que controlava o vôo do seu avião em dois eixos;
- Em 1943 foi realizada a primeira aterrissagem automática na Alemanha;
- Em 1947 um C-54 realizou o primeiro vôo transatlântico - do Canadá à Inglaterra - sob controle automático, da decolagem à aterrissagem. Durante o vôo os dispositivos de comando do avião não foram tocados pelos pilotos, com todos os comandos sendo realizados por um programa armazenado em cartões perfurados.

Enfim, os sistemas de controle e aumento de estabilidade de aeronaves evoluíram muito nos primeiros 50 anos, e mais ainda nos últimos 40 anos do século passado, incorporando toda a evolução da eletrônica em sensores e computadores; no uso da teoria e da prática para o projeto dos sistemas de controle de aeronaves com o auxílio de sistemas SISO e MIMO, de técnicas de controle adaptativo, de controle com estrutura variável, e outros (BLAKELOCK, 1991). De outro lado, os sistemas de navegação (em inglês *guidance*) tem propiciado recursos para a programação de vôos onde o piloto define o local de chegada, e exerce o papel de supervisor, acompanhando mais detidamente a decolagem e a aterrissagem (BLAKELOCK,1991).

Assim, os veículos aéreos têm incorporado sistemas de controle que propiciam o controle de aviões com decolagem vertical, de mísseis que possuem grande precisão, de

helicópteros capazes de pairar automaticamente em condições de turbulência (MCLEAN, 1990, BLAKELOCK, 1991), de helicópteros capazes de executar missões, sem a intervenção do ser humano, utilizando sensores visuais, como na competição de veículos autônomos aéreos (AERC, 2001), sendo previsível a aplicação destas tecnologias em dirigíveis.

#### **4.2.2 Publicações sobre Controle de Dirigíveis**

Os dirigíveis são, na sua maioria, intrinsecamente estáveis (KHOURY, 1999, GOMES, 1990), apresentando no entanto problemas nas manobras em baixa velocidade (DORRINGTON, 1999, MOWFORTH, 1991), pois são muito susceptíveis às turbulências e suas superfícies de atuação aerodinâmica tem capacidade de atuação muito pequena nesta condição. Assim, eles são o oposto dos helicópteros que são muito manobráveis, mas intrinsecamente instáveis. Entretanto, apesar dos problemas de manobrabilidade, visto a disponibilidade de motores para compor sistemas de propulsão com diferentes características, de materiais que não eram disponíveis há 70 anos e de uma variedade de técnicas de projeto e controle de aeronaves, pode se afirmar que as dificuldades que estão sendo encontradas hoje no desenvolvimento de dirigíveis são menores que as que já foram enfrentadas no desenvolvimento de outros veículos aéreos.

Como expresso em KHOURY (1999), até recentemente pouco foi feito no que se refere ao desenvolvimento de sistemas de controle de dirigíveis, tendo em vista seu relativo abandono nos últimos 50 anos. Por outro lado, a existência de vários projetos de dirigíveis concorrentes, permite concluir que parte dos resultados até agora obtidos vem sendo mantidos em sigilo pelas empresas, o que pode ser constatado pela quase que total ausência de artigos que detalhem soluções para os sistemas de controle desses novos dirigíveis, notadamente os projetos comerciais.

A seguir, resumem-se os artigos sobre controle de dirigíveis obtidos de consulta em três fontes, a primeira consiste na pesquisa realizada por Sérgio Gomes (GOMES, 1990), que atuou na área de aerodinâmica de dirigíveis, e repassou as referências obtidas até 1996. A segunda, foi resultado de consulta feita na biblioteca da *Airship Association* em Londres em 1999. A terceira, o constante acompanhamento das publicações divulgadas na LTA-AIAA, *Airship Convention*, *Journal of Airship*, *Journal of Guidance*, IEEE-ICRA, IEEE-IROS.

O autor que tem publicado de forma mais consistente (desde 1981) em temas ligados a estudos de configuração de propulsores e alternativas para controle de dirigíveis é Bellur Nagabushan, atualmente na Universidade de Saint Louis, EUA, destacando-se:

- NAGABUSHAN (1981) analisa a dinâmica e os esquemas de controle para configurações híbridas de dirigível-helicóptero para manobra de cargas pesadas em vôo pairado.
- NAGABUSHAN (1984) examina o uso de propulsão vetorizada para gerar forças em torno do eixo lateral e do eixo longitudinal, e a adição de motores de popa e proa. Mostra a melhoria na capacidade de manobra obtida com tais configurações.
- NAGABUSHAN (1986) mostra as vantagens do uso da propulsão vetorizada em dirigíveis para a diminuição da distância requerida para a decolagem. Também é apresentado o diagrama do sistema de controle do leme de profundidade e de vetorização para a utilização destes recursos em decolagem e aterrissagem.
- NAGABUSHAN (1989) mostra que a inclusão de motores para controle de direção e arfagem montados em popa e proa, aumenta consideravelmente a manobrabilidade em direção e arfagem em baixas velocidades, possibilitando a diminuição da dimensão dos lemes de direção e profundidade.
- NAGABUSHAN (1991) analisa em simulação a manobrabilidade adicional em direção pela inclusão de motor de proa, mostrando os ganhos resultantes.
- NAGABUSHAN (1995) examina em simulação, o efeito da adição de um motor de proa montado perpendicularmente ao eixo longitudinal e um esquema de transição da intensidade de comando deste à medida que a velocidade aumenta, mostrando que em baixas velocidades este motor aumenta a manobrabilidade e que em velocidades maiores é desnecessário.
- NAGABUSHAN (1997) mostra que controladores de guinada utilizando técnicas baseadas em  $H_\infty$  apresentam, em simulação, desempenho superior a controladores utilizando uma estratégia de controle proporcional (P) ou proporcional / derivativo (PD) da guinada.

Um outro grupo de autores são pesquisadores de universidades e centros de pesquisa alemães e da empresa Zeppelin. Destes, destaca-se os seguintes trabalhos:

- ADERMANN (1993) apresenta o uso, em simulação, de técnicas de controle robusto para o controle longitudinal do Zeppelin-NT.
- UNZICKER (1993) apresenta detalhes do projeto aerodinâmico e da especificação do sistema de aumento de estabilidade do Zeppelin-NT. Mostra-se que um objetivo a ser alcançado no seu projeto é deixá-lo ligeiramente instável o que é conseguido com a sua carenagem, objetivando a maior manobrabilidade e economia no peso estrutural. São discutidas as características do sistema de aumento de estabilidade, mostrando que no caso do Zeppelin-NT, um fator crítico é a variação da capacidade de manobra em função da condição de flutuação deste, se leve ou pesado.
- ROTH (1994, 1994a) apresenta o uso de lógica *fuzzy* e *fuzzy* adaptativa para estudos em simulação do controle longitudinal do Zeppelin-NT.
- BUHLMANN (1995) descreve o sistema de propulsão vetorizado do Zeppelin-NT e a infra-estrutura computacional utilizada para o controle.
- KAEMPF (1995, 1997) mostra a aplicação da metodologia clássica de desenvolvimento de sistemas de controle para aeronaves, onde se separam os modos de movimentos laterais e longitudinal, considerando-os desacoplados. Concluiu-se que para o caso do veículo utilizado, o dirigível Lotte, foi necessário adaptar a metodologia em consequência da existência de acoplamento entre os modos neste dirigível, sendo requerido o uso de métodos de controle multivariável para o ajuste dos parâmetros.
- DANNEKER (1999) apresenta o resultado de estudo simulado mostrando as vantagens de manobrabilidade em solo originadas pelo sistema de propulsão de popa utilizado no Zeppelin-NT.
- RUGER *et al.* (1999) estuda a relação entre o arrasto do envelope do dirigível Lotte e o arrasto deste devido à montagem do motor na sua popa.

- WIMMER (2001) mostra a metodologia para a identificação das características dinâmicas do dirigível Lotte e se apresenta proposta de esquema de controle de altitude que ainda se encontra em fase de testes.

HOOKWAY (1979) e CAVALCANTI (1981) consideram o uso de estratégias de controle preditivo para compensar atrasos de transporte presentes na resposta dos dirigíveis.

MCISAAC (2000) e ZWAAN (2001) mostram o uso de técnicas de *visual servoing* no controle de um dirigível de pequeno porte para uso em interiores.

Sobre o uso de sistemas de piloto automático em dirigíveis, a informação disponível é muito pequena. O Zeppelin-NT (ZEPNT, 2001) é equipado com um sistema FBW (do inglês *flight by wire*), mas nada é comentado sobre a disponibilidade de piloto automático, exceção feita ao artigo de UNZICKER (1993), que mostra a importância de SAS (do Inglês *Stability Augmentation System*), para este. No caso do Cargo Lifter (CLF, 2001), também nada é comentado. Entende-se, entretanto, que a tarefa do piloto seria em muito facilitada com a presença de piloto automático, considerando-se que a autonomia prevista para esta aeronave é de mais de 24 horas, além da necessidade de vôo pairado sobre o local de carga e descarga.

Outra fonte de publicações sobre controle de dirigíveis relaciona-se ao projeto AURORA, com 40 artigos ligados à área de dirigíveis, destacando-se os seguintes:

- DE-PAIVA *et al.* (1999b) mostram a aplicação simulada de técnicas de controle robusto no controle do ângulo de arfagem de dirigíveis.
- DE-PAIVA *et al.* (1999a) mostram um ambiente MATLAB para desenvolvimento de sistema de controle para o dirigível do projeto AURORA.
- AZINHEIRA *et al.*, (2000, 2000a, 2000b, 2000d) apresentam alternativas nas estratégias de controle de trajetória para dirigíveis, aplicadas em simulação ao caso do AS800.
- AZINHEIRA *et al.* (2000C) mostram estudo simulado de controle de dirigível em vôo pairado.
- DE-PAIVA *et al.* (2001) discutem o uso de um controlador  $H_2/H_\infty$  para controle de guinada.

- RAMOS *et al.* (2001a, 2001b) apresentam resultados práticos de controle de trajetória com um dirigível.
- ELFES *et al.* (2000a, 2000b) apresentam aplicações simuladas, do uso de sistemas de visão e de esquemas de percepção para a navegação de dirigível.
- SILVEIRA *et al.* (2001) apresentam formulação, estudo e simulação para controle por visão (*visual servoing*) do vôo pairado e seguimento de retas para o dirigível.

Da análise da literatura constata-se que Dirigíveis possuem características dinâmicas particulares em função da velocidade e do peso em relação à fluabilidade do veículo. Para médias e altas velocidades, a manobrabilidade é conseguida através das superfícies aerodinâmicas. Em contraposição, para velocidades muito baixas, nas quais se verifica uma redução acentuada na eficiência das superfícies, movimentos de subida e descida devem-se principalmente à vetorização da propulsão, enquanto que a movimentação lateral é conseguida pela diferença de velocidade entre dois motores (quando eles existem) situados a cada lado do eixo longitudinal do veículo – o que é conhecido por propulsão diferencial, ou eventualmente por motores de proa ou popa montados perpendicularmente ao eixo do veículo. Outra característica de dirigíveis a baixas velocidades é um comportamento de fase não mínima. Dirigíveis são, portanto, veículos caracterizados por uma dinâmica multivariável, variante no tempo (em função do peso e da velocidade), mas intrinsecamente estáveis, no plano longitudinal devido à estabilidade de pêndulo. No plano lateral é freqüente a existência de velocidade residuais em guinada (GOMES, 1990).

Constatou-se, também, que parte das publicações analisadas apresenta resultados de estudos visando a melhoria da capacidade de manobra dos dirigíveis através de configurações alternativas de montagem do sistema de propulsão e não sobre o seu controle em malha fechada, o que ainda constitui um assunto em aberto na comunidade.

No caso específico do veículo Lotte, os trabalhos sobre aumento de estabilidade são motivados pela existência de modo instável em determinada faixa de velocidade, pelo fato deste ter seu centro de gravidade muito elevado devido as células solares dispostas sobre a parte superior do envelope.

Os trabalhos realizados no contexto do projeto AURORA sobre técnicas de controle e navegação de dirigível apresentam, na sua maioria, desenvolvimentos em simulação e alguns resultados práticos alentadores, através de implantação e validação em vôos experimentais, o que é abordado neste capítulo e no Capítulo 8.

Com respeito aos trabalhos sobre a aplicação de *visual servoing* com dirigíveis de pequeno porte, os resultados obtidos em locais fechados são interessantes, mas teria que se avaliar a aplicabilidade destes resultados em dirigíveis de maior porte e em ambiente externo; este constitui um assunto em desenvolvimento no AURORA.

Para concluir, nota-se que não há um grande número de artigos nem um grande número de pesquisadores atuando na área de controle de dirigíveis, e menos ainda quando se considera a validação através de resultados práticos experimentais.

### **4.3 Controle e Seguimento de Trajetória**

Nesta seção, apresenta-se inicialmente uma visão geral sobre o problema de controle e seguimento de trajetória para veículos aéreos tais como aviões e para robôs manipuladores. A partir desta base conceitual, aborda-se então o controle de trajetória para dirigíveis.

#### **4.3.1 Controle de Trajetória em Veículos Aéreos e Manipuladores**

Na literatura de referência de sistemas de controle de aviões e helicópteros, a questão do controle de trajetória não é tratada em profundidade, como pode ser visto em MCLEAN (1990) e BAKELOCK (1991). Quando se considera esta questão, ela é abordada em termos de se gerar sinais de controle do veículo para que este atinja um alvo, inclusive móvel. Não há portanto preocupação explícita em se precisar a sua trajetória, mas com a capacidade do veículo seguir e atingir o alvo, através da geração de referências de direção e de velocidade para os respectivos controladores.

No caso de aviões, o controle de trajetória é implementado usando como variável de controle a posição do leme de direção e/ou pela diferença de sustentação nas asas, através dos comandos de propulsão e rolagem (aileron) (MCLEAN, 1990). No caso de aviões comerciais, os sinalizadores de ajuda à navegação e aterrissagem, como o ILS (do Inglês *instrument landing system* - sistema de aterrissagem por instrumentos) (MCLEAN,

1990), são de extrema importância por enviar sinais de referência que são utilizados pelo sistema de controle automático de aterrissagem para controlar direção e altitude do veículo.

A Figura 4.1 (MCLEAN, 1990) mostra o princípio de operação do sistema de controle de direção em aviões. Ele é realizado através de três malhas:

- i) uma malha externa de controle de ângulo de guinada;
- ii) uma segunda malha de controle de ângulo de rolagem e;
- iii) uma malha interna de controle de velocidade de rolagem.

Nota-se que, a partir de uma direção de referência " $\psi_{ref}$ ", são gerados os comandos para a malha interna de controle do ângulo de rolagem, que por sua vez gera comandos para a malha interna de controle de velocidade de rolagem.

Como já se disse, na maioria das situações, as curvas em aviões são feitas através da diferença de sustentação das asas, gerando um efeito de rolagem. Além de " $\psi_{ref}$ ", as variáveis neste esquema de controle são o ângulo de guinada " $\psi$ ", o ângulo de rolagem " $\phi$ ", a velocidade de rolagem " $p$ " e " $\phi_{ref}$ ", que constitui a referência para o controlador do ângulo de rolagem; há também os ganhos dos controladores  $K_1$  – o controlador da malha externa de ângulo de guinada e  $K_2$  – o controlador da malha interna de rolagem. Assim, a trajetória executada pelo veículo segue a evolução das referências de ângulo de guinada no tempo, geradas em determinadas situações por dispositivos como o ILS.

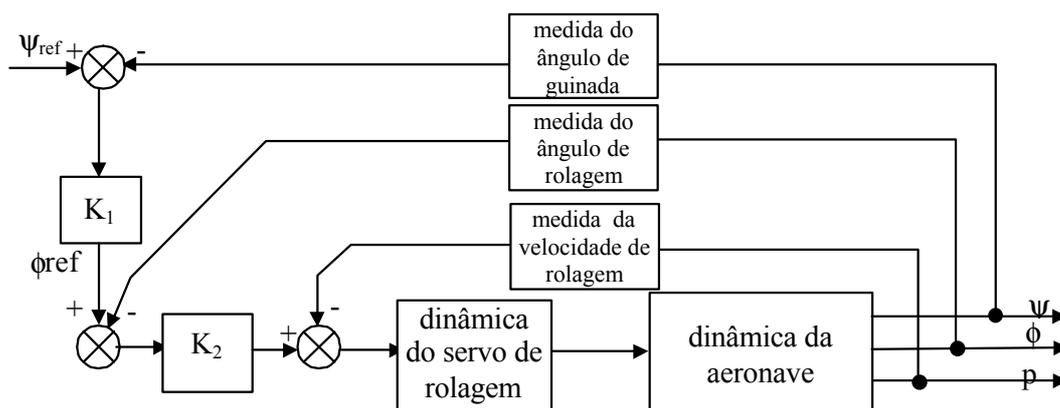


Figura 4.1: Diagrama de referência para controle direção em aviões.

Já no caso de robôs manipuladores, as técnicas de controle de trajetória foram estudadas na década de 80 (PAUL, 1981; HASEGAWA, 1983; CRAIG, 1989), sendo importantes em várias aplicações (por exemplo em soldagem a arco ou em aplicação de

adesivos). Dentre as técnicas de controle destaca-se o ATGS (do Inglês *Autonomous Trajectory Generation Servomechanism* – servo-mecanismo autônomo para geração de trajetórias) (MIZUTANI, 1984; RAMOS, 1986; MADRID, 1990). O ATGS se baseia no uso de medidas como o erro de trajetória " $d$ " e sua taxa de variação para gerar comandos de controle, como mostrados na Figura 4.2.

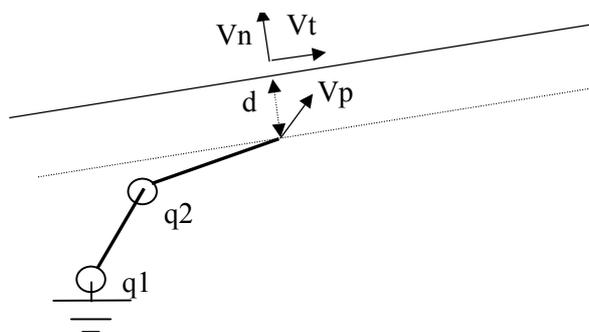


Figura 4.2: A técnica ATGS.

O princípio do ATGS é o de utilizar a distância " $d$ " e também sua taxa de variação " $\dot{d}$ "; a variável " $d$ " corresponde ao erro entre a trajetória de referência, dada pela linha contínua na Figura 4.2, e a situação atual, mostrada pela ponta do manipulador. A partir dessas variáveis, são gerados dois termos para o controle de sua trajetória. Um dos termos, paralelo à trajetória de referência, representado por " $V_t$ " e outro, normal à ela, de correção de erro, chamado " $V_n$ ", que é função do erro de trajetória " $d$ " na figura e de " $\dot{d}$ ". A composição de " $V_t$ " e " $V_n$ " gera a velocidade " $V_p$ ", que comanda o movimento a ser executado pela ponta do manipulador para aproximá-la da trajetória de referência. Para a execução do movimento da ponta, essa velocidade é transformada para as velocidades das juntas " $q_1$ " e " $q_2$ " do robô. O objetivo dos controles tipo ATGS é minimizar o erro de trajetória e a velocidade de erro de trajetória, através da especificação da velocidade espacial do movimento que o efetuador deve executar.

#### 4.3.2 Algoritmo para Controle de Trajetória do Dirigível

O algoritmo de base para o controle de trajetória do dirigível foi desenvolvido por AZINHEIRA et al.(2000), inicialmente no contexto dos projetos ARMOR / SARA (Aeronave Robotizada para Missões de Observação e Reconhecimento / Sistemas Automatizados de Reconhecimento Aéreo) desenvolvidos no IST de Lisboa, (LOURTIE *et al.*, 1995) e posteriormente particularizado para o dirigível do projeto AURORA,

(AZINHEIRA, 2000a). Assim como no ATGS, o algoritmo utiliza, para correção de trajetória, um erro caracterizado por duas componentes: o erro efetivo de posição e, o erro previsto em função da velocidade, constituindo o chamado erro previsto de trajetória em um horizonte à frente (do Inglês *look ahead error*). Este erro, originário das duas componentes, é usado como sinal atuante de uma malha externa de controle de trajetória; o sinal resultante desta malha externa é fornecido a uma malha interna de controle, que regula o ângulo de guinada do dirigível. A configuração do sistema de controle em duas malhas corresponde ao princípio clássico adotado em aviões.

A obtenção do erro previsto de trajetória " $\delta_a$ " (AZINHEIRA, 2000a) é mostrada na Figura 4.3. Nesta figura aparecem:

- i) o dirigível na sua posição e orientação atuais;
- ii) a reta de referência que caracteriza a trajetória a ser seguida, com o ângulo de guinada da trajetória,  $\Psi_{traj}$ , em relação a um sistema de coordenadas inercial denominado SCI, expresso em termos de orientações Norte e Leste, como introduzido na Seção 3.3;
- iii) a velocidade inercial " $V_T$ " do veículo e o ângulo " $\epsilon$ " desta com a trajetória de referência, e;
- iv) a distância a que o dirigível se encontra desta direção de referência, chamada erro efetivo de trajetória " $\delta$ ", uma componente do erro devido à velocidade inercial " $\delta_v$ ", sendo que juntas elas compõem o erro previsto de trajetória " $\delta_a$ ".

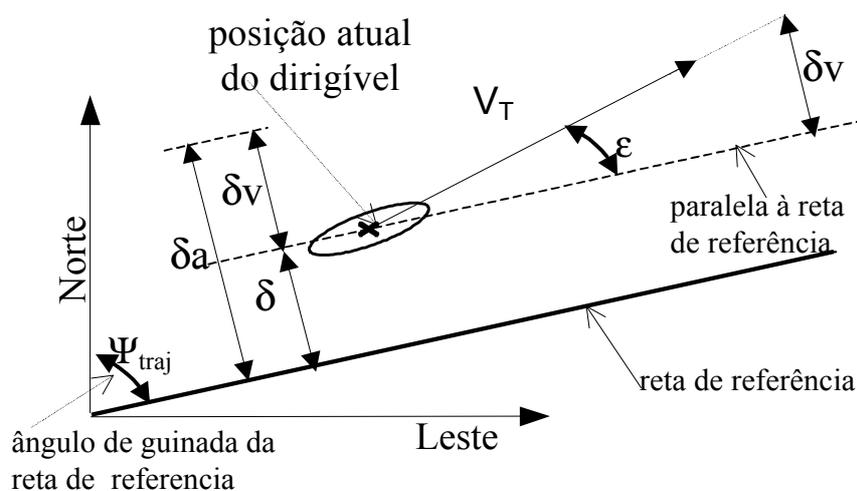


Figura 4.3: Variáveis para seguimento de trajetória.

A estratégia de controle utilizada, baseia-se no fato que a taxa de variação do ângulo " $\epsilon$ " é a própria velocidade de guinada do veículo " $r$ " (Equação 4.1). É então razoável o uso do ângulo de guinada " $\psi$ " para o controle de trajetória, conforme é mostrado em AZINHEIRA *et al.* (2000a):

$$d\epsilon/dt = r \quad 4.1$$

Na Figura 4.3 aparecem o erro efetivo de trajetória " $\delta$ " mais o erro devido à velocidade " $\delta v$ ", como os dois componentes que formam o erro previsto de trajetória " $\delta_a$ ". O erro de velocidade " $\delta v$ " corresponde à taxa de variação do erro efetivo de trajetória, chamado de  $d\delta/dt$ . Utilizando-se uma aproximação de primeira ordem para " $\delta v$ ", o erro previsto de trajetória em um horizonte de previsão à frente " $\Delta t_p$ " definido em AZINHEIRA (2000a) pode ser dado por:

$$\delta_a = \delta + (d\delta/dt) \Delta t_p \quad 4.2$$

Uma expressão equivalente para a equação 4.2 pode ser expressa a partir da velocidade inercial  $V_T$  do veículo. Da Figura 4.3, considerando-se que a velocidade  $V_T$  não sofre variações bruscas no intervalo " $\Delta t_p$ ", e assumindo valores suficientemente pequenos para  $\epsilon$  pode-se observar que:

$$d\delta/dt = V_T \sin(\epsilon) \approx V_T \epsilon \quad 4.3$$

Introduzindo a variável " $V_{Perp}$ ", correspondente à velocidade perpendicular à trajetória:

$$V_{Perp} = V_T \sin(\epsilon) \approx V_T \epsilon \quad 4.4$$

observa-se que:

$$d\delta/dt = V_{Perp} \quad 4.5$$

A estratégia de controle implementada em AZINHEIRA *et al.* (2000a) visa minimizar o erro previsto de trajetória  $\delta_a$  agindo sobre o ângulo de guinada. O algoritmo de controle de trajetória apresentado na Figura 4.4 contempla dois níveis em cascata: o de controle de trajetória (malha externa) e o de controle de ângulo de guinada (malha interna). A malha externa implementada por AZINHEIRA *et al.* (2000a) corresponde a um

controlador PI (proporcional-integral) que atua sobre o erro previsto de trajetória, o qual pode ser escrito como:

$$u(t) = K_p \delta_a(t) + K_i \int \delta_a(t) dt \quad 4.6$$

onde  $u(t)$  é o sinal de controle,  $K_p$  é o ganho proporcional e  $K_i$  é o ganho integral

Substituindo  $\delta_a(t)$ , dado pela equação 4.2, em 4.6, tem-se que:

$$u(t) = K_p [\delta(t) + (d\delta/dt) \Delta t_p] + K_i \int [\delta(t) + (d\delta/dt) \Delta t_p] dt$$

donde

$$u(t) = K_p \delta(t) + K_p \Delta t_p (d\delta/dt) + K_i \int \delta(t) dt + K_i \Delta t_p \int (d\delta/dt) dt \quad 4.7$$

Agrupando-se os termos e realizando-se simplificações obtém-se

$$u(t) = (K_p + K_i \Delta t_p) \delta(t) + K_i \int \delta(t) dt + K_p \Delta t_p (d\delta/dt)$$

E utilizando-se a relação 4.5, que associa  $d\delta/dt$  à  $V_{perp}$ , esta última definida por 4.4, a expressão para o controlador de trajetória resulta na Equação 4.8:

$$u(t) = (K_p + K_i \Delta t_p) \delta(t) + K_i \int \delta(t) dt + K_p \Delta t_p V_{perp} \quad 4.8$$

Assim, da mesma forma que o ATGS, o controlador PI atuando sobre o erro previsto de trajetória " $\delta_a$ ", apresentado em AZINHEIRA *et al.* (2000a), é equivalente a um controlador PIV de trajetória, com ganhos proporcional ( $K_{PT}$ ), integral ( $K_{IT}$ ) e de velocidade ( $K_{VT}$ ), dados pela equação 4.9.

$$K_{PT} = K_p + \Delta t_p K_i \quad 4.9$$

$$K_{IT} = K_i$$

$$K_{VT} = \Delta t_p K_p$$

Nota-se que a componente de velocidade, que usualmente aparece no controle de dispositivos eletromecânicos, corresponde na prática a uma ação derivativa (sobre a posição) a qual introduz um caráter antecipatório à ação de controle. Apesar do horizonte de predição originalmente proposto influenciar também o ganho proporcional, ele afeta diretamente e sobretudo o ganho de velocidade, funcionando como ponderador deste caráter antecipatório, coerente portanto com o aspecto preditivo da formulação original. Estes ganhos " $K_{PT}$ ", " $K_{VT}$ " e " $K_{IT}$ " do controlador de trajetória, e os ganhos " $K_{RT}$ " e " $K_{RRT}$ "

do controlador do ângulo de guinada, são utilizados para a implementação do sistema de controle, conforme o diagrama mostrado na Figura 4.4.

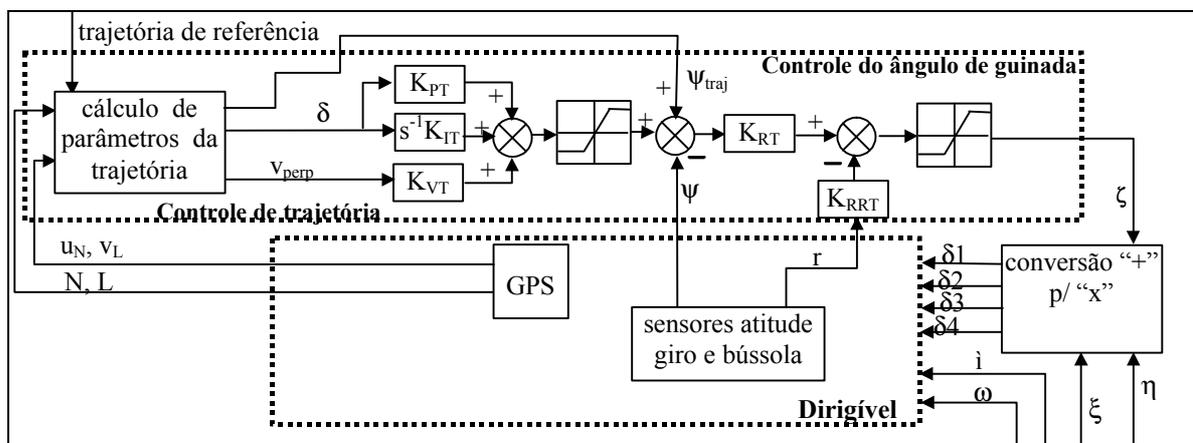


Figura 4.4: Algoritmo de controle de trajetória.

Conforme se mostrou, o controlador de trajetória implementado na malha externa é do tipo PIV, equivalente ao controlador PI atuando sobre o erro previsto de trajetória. A variável de controle da malha externa é a variação necessária no ângulo de guinada para que o veículo corrija a sua trajetória e diminua o erro em relação à reta de referência. As variáveis em questão, expressas em termos de orientações Norte e Leste - NL do sistema de coordenadas inercial SCI, são obtidas a partir dos sensores embarcados no dirigível (para detalhamento dos sensores, vide Seção A1.3.3), que fornecem as posições (Norte - N, Leste - L) e velocidades (Norte -  $u_N$ , Leste -  $v_L$ ). A partir desses valores e da trajetória de referência são obtidos o erro de trajetória " $\delta_a$ ", a velocidade perpendicular " $v_{Perp}$ " e o ângulo de referência da trajetória " $\psi_{traj}$ ". A malha interna corresponde a um controlador PD de guinada, utilizando como referência o ângulo de guinada requerido para correção dos erros de trajetória e utilizando as medições da bússola eletrônica " $\psi$ " e da velocidade de guinada " $r$ " fornecida pelo giroscópio da central inercial. Esta malha determina o comando do leme de direção " $\zeta$ " que, no caso do AS800, é transformado para as variáveis  $\delta_1$ ,  $\delta_2$ ,  $\delta_3$  e  $\delta_4$  (vide equação 3.1 do Capítulo 3) no processo de conversão de comandos configurados em "+" para a deflexão das superfícies de atuação do dirigível montados em "x".

#### 4.4 Controle de altitude

A questão de controle de altitude é aqui discutida, inicialmente, no caso de aviões, helicópteros e mísseis. No caso de aviões os objetivos são de se manter a altitude constante e se ter o controle na taxa de subida e/ou descida, visando assegurar o conforto dos

passageiros. Já no caso de mísseis, o objetivo é fazer com que o veículo acompanhe rapidamente o perfil do terreno, visando, por exemplo, mantê-lo fora do alcance de radares. No caso de helicópteros, em algumas situações como a de voo pairado, é requerida a manutenção da altitude e posição mesmo na presença de perturbações, enquanto que em voo de cruzeiro os objetivos são similares àqueles explicitados para aviões.

A forma clássica de se controlar a altitude em aviões é através do comando do leme de profundidade. Os sistemas de controle, conforme mostrado por MCLEAN (1990), usam uma malha externa com a medida de altitude barométrica e uma malha interna de controle do ângulo de arfagem com a realimentação da velocidade de arfagem, como apresentado na Figura 4.5. Nesta Figura, as variáveis são a altitude de referência " $H_{ref}$ ", a altitude medida pelo altímetro barométrico " $H_{atual}$ ", o ângulo de arfagem " $\theta$ ", a referência do ângulo de arfagem " $\theta_{ref}$ " e a velocidade de arfagem " $q$ "; e o ganho do controlador de altitude " $K_1$ ". MCLEAN (1990) mostra que este esquema apresenta características de desempenho superiores a outras alternativas.

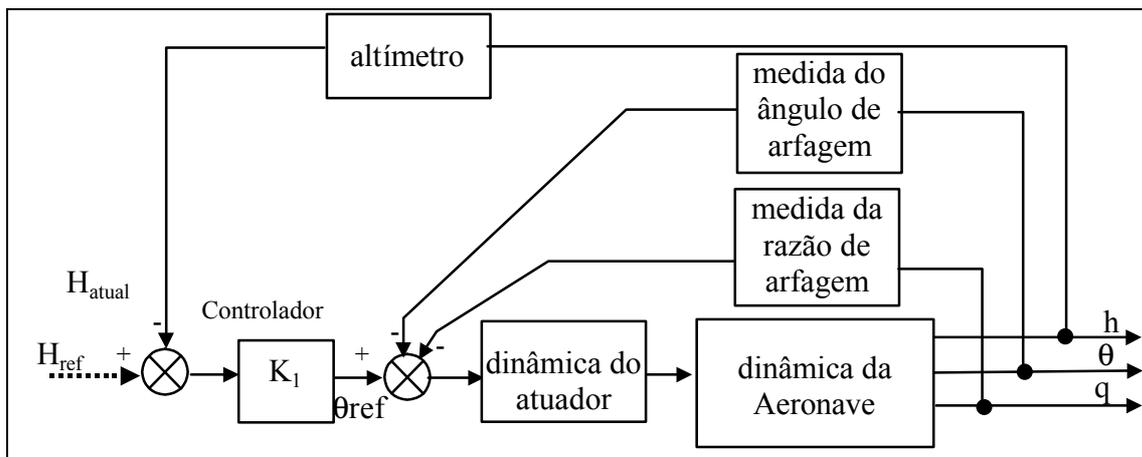


Figura 4.5: Diagrama de referência para controle de altitude.

Também em MCLEAN (1990) é mostrado que:

i) para o controle da taxa de descida para aterrissagem, baseado no sistema ILS, é utilizado o controle direto do ângulo de arfagem, a partir de medidas da taxa de descida e altitude e;

ii) para se implementar um seguimento de terreno em aviões militares são requeridas informações sobre: aceleração vertical, velocidade de subida, altitude, comando de propulsão, variação de propulsão e variação da velocidade relativa ao ar.

No caso de dirigíveis, algumas aplicações podem requerer altitude constante, outras altura constante em relação ao terreno, mas sem as exigências dos aviões militares. Para auxílio ao controle de altitude, alguns veículos como o AS800 dispõem de propulsão vetorizada que pode compensar a situação em que a sustentação aerodinâmica seja insuficiente – a baixas velocidades, por exemplo –, como mostrado em NAGABUSHAN (1986).

Em NAGABUSHAN (1986), WIMMER (2001) e DE-PAIVA (1999b) são mostradas algumas alternativas para o controle da altitude de dirigíveis. O princípio geral é análogo ao mostrado na Figura 4.5, ou seja, uma malha externa a partir da realimentação da altitude, e uma malha interna de arfagem gerando os comandos de controle do leme de profundidade; esta estratégia é usada para o controle da altitude em vôo de cruzeiro, sem o uso da vetorização. Em DE-PAIVA (1999b), também é apresentado o controle da altitude utilizando vetorização e leme de profundidade.

O diagrama de blocos do sistema de controle de altitude utilizado no AS800 para vôo de cruzeiro (ou seja, sem o uso de vetorização pois ela não se faz necessária nessa faixa de velocidades), está mostrado na Figura 4.6. Observa-se que a partir da referência de altitude " $H_{ref}$ ", que reflete ao longo do tempo o perfil de altitude a ser seguido pelo dirigível, e das medidas de altitude atual " $H_{atual}$ ", realizada pela sonda de vento, há uma malha externa de controle de altitude com realimentação da velocidade vertical " $w_A$ ", obtida pelo GPS, e uma malha interna de ângulo de arfagem que utiliza as medidas do ângulo e velocidade de arfagem, realizadas pela central inercial, e determina o comando do leme de profundidade " $\eta$ " para controlar o veículo. No caso do AS800, o comando " $\eta$ " é transformado para as variáveis  $\delta_1$ ,  $\delta_2$ ,  $\delta_3$  e  $\delta_4$  devido à conversão de comandos configurados em "+" para a deflexão dos atuadores do dirigível montados em "x".

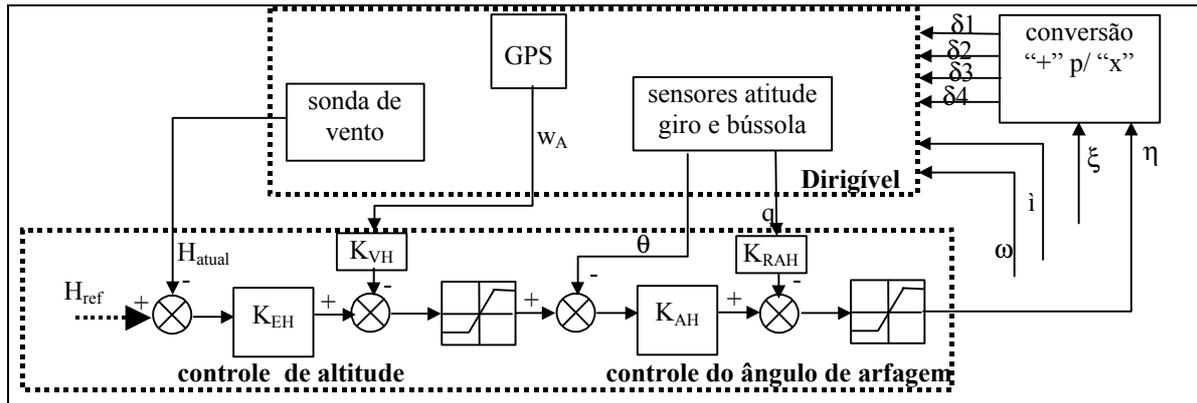


Figura 4.6: Algoritmo de controle de altitude.

Os ganhos utilizados para a implementação do sistema de controle são " $K_{EH}$ " e " $K_{VH}$ " do controlador de altitude, mais os ganhos " $K_{AH}$ " e " $K_{RAH}$ " do controlador do ângulo de arfagem. Os sensores citados são os descritos no Anexo 1 (A1.3.3).

## 4.5 Conclusões

Conforme mostrado inicialmente no capítulo, o resultado da pesquisa bibliográfica relativa ao uso de técnicas de controle automático em dirigíveis permitiu constatar que são reduzidos os desenvolvimentos nesse domínio e menores ainda os resultados de aplicação prática de técnicas de controle e de navegação em termos de seguimento de trajetórias e de perfis de altitude.

Neste sentido, este capítulo apresentou estratégias de controle de trajetória e altitude para veículos aéreos e também para robôs manipuladores, constituindo a base para a apresentação das técnicas de controle para dirigíveis, mostrando suas características gerais.

Em seguida apresentou-se o algoritmo de controle desenvolvido por AZINHEIRA et al. (2000) com característica preditiva utilizado como referência para controle de trajetória. Esta formulação foi então reinterpretada segundo um controlador PID para o Controle de Trajetória. Em seguida apresentou-se o diagrama geral do algoritmo de controle de trajetória para o dirigível AS800, mostrando o mapeamento da infra-estrutura sensorial instalada a bordo do veículo com o diagrama geral de controle.

Por fim foi apresentado o algoritmo utilizado como referência para controle de altitude e o seu mapeamento num diagrama visando a sua implementação, o que inclui a infra-estrutura de sensores instalada no dirigível.

O próximo capítulo apresenta aspectos ligados à Arquitetura de *Software* Robótico para o Dirigível, que utiliza como base comportamentos de controle de trajetória e controle de altitude apresentados no presente capítulo.



## 5. ARQUITETURA DE *SOFTWARE* ROBÓTICO PARA O DIRIGÍVEL

### 5.1 Introdução

O termo robô, originário da palavra Tcheca “robot” (escravo) (CRAIG, 1989), é utilizado para denotar diferentes dispositivos, incluindo manipuladores, veículos móveis, programas de computador etc. Uma característica comum aos chamados robôs é a existência de diferentes níveis de autonomia, isto é, de capacidade de auto-determinação e, por conseguinte, de operar sem intervenção humana.

Além da capacidade de autonomia um outro aspecto a considerar é o nível de inteligência, pois um alto grau de autonomia não implica em um alto grau de inteligência. por exemplo, uma chave termostática possui um alto grau de autonomia e nenhuma inteligência, enquanto que um sistema perito tem uma alta inteligência e nenhuma autonomia. Assim, o grau de autonomia de um sistema depende não apenas do seu nível de inteligência, mas, também, da complexidade e do nível de incerteza do ambiente em que ele atua.

Já foram construídos robôs com um alto grau de autonomia para operarem em ambientes controlados - tipicamente os robôs com aplicação industrial - e também robôs autônomos com capacidade de agir no mundo real, o que se mostra muito mais complexo. Para os robôs autônomos, esta complexidade é caracterizada nas situações reais pela necessidade de atingir metas definidas em alto nível para o robô, que deve ao mesmo tempo ser apto a reagir face à situações inesperadas (COSTE-MANIERE e SIMMONS, 2000). Assim, a arquitetura de *software* que controla o robô, onde a palavra controle possui nesse contexto um significado mais amplo, deve coordenar atividades concorrentes, seqüenciais, monitorar o ambiente através de sensores e tratar exceções (SIMMONS, 1998).

Este capítulo propõe uma primeira abordagem para a arquitetura de *software* robótica (ASR) para o dirigível. Assim, a Seção 5.2 apresenta a caracterização de robôs quanto à sua capacidade de autonomia e adaptação a variações do seu ambiente. A seguir, na Seção 5.3, é apresentado um panorama das arquiteturas de *software* robótico que possibilitam que eventos e decisões sejam tratados pelo sistema no decorrer de sua

operação, mostrando os motivos que levaram as arquiteturas híbridas a serem padrão de fato para os sistemas robóticos autônomos. Na Seção 5.4 caracteriza-se a constituição das arquitetura híbridas e é apresentada a linguagem TDL, uma das linguagens utilizadas para compor o nível executivo de uma arquitetura híbrida.

A Seção 5.5 apresenta uma proposta da arquitetura de *software* robótico para o dirigível, constituindo desta maneira uma primeira solução cujo intuito é prover os recursos básicos para a especificação e execução de ações deliberativas e reativas do dirigível no decorrer de sua operação.

Por fim, na Seção 5.6, são apresentadas as conclusões do capítulo.

## 5.2 Robôs

O progresso recente do desenvolvimento e aplicação de sistemas robóticos autônomos deve-se, em grande parte, à interseção entre a Inteligência Artificial e a Robótica, originando a área de pesquisa chamada de Robótica Inteligente Autônoma (do Inglês *Intelligent Autonomous Robotics*). O uso de inteligência artificial em robôs não só aumenta o nível de abstração no qual as tarefas podem ser representadas, mas também o nível de abstração no qual o sistema é capaz de operação totalmente autônoma.

A Tabela 5.1 (FRASER, 1994) mostra os diferentes níveis de autonomia para robôs e sua designação.

<b>Nível de Autonomia</b>	<b>Designação</b>	<b>Representação das Tarefas</b>	<b>Método de Comando</b>
Baixa	Robô Teleoperado	Geométrica	Joysticks ou similares
Média	Robô Supervisionado	Mista	Comandos de movimento, tarefas simples
Alta	Robô Autônomo	Simbólica	Tarefas complexas e descrição de missões

Tabela 5.1: Níveis de Autonomia em Sistemas Robóticos Inteligentes

O nível de autonomia baixo, chamado teleoperado, é caracterizado por uma alto grau de intervenção humana na operação. Os dispositivos são comandados por teleoperação, utilizando comandos manuais. Nos sistemas com média autonomia, chamados supervisionados, a intervenção humana acontece na definição das tarefas, através de comandos de movimentos, e o seqüenciamento destas, possuindo o robô a inteligência necessária para executar estes comandos, dentro do seu ambiente de operação.

Nos robôs com alta autonomia, chamados de robôs autônomos, o robô recebe uma missão descrita num nível de abstração alto, que consiste em tarefas ou objetivos, associados a prioridades e limitações. A partir dessas especificações em alto nível, o robô deve ser capaz de desenvolver um plano para realizar essas tarefas, decompostas num conjunto de sub-tarefas e uma seqüência de execução. Isto é chamado de **capacidade de planejamento**. A finalização das sub-tarefas pode gerar, recursivamente, outras sub-tarefas com menor nível de abstração e maior nível de detalhamento. Em todo momento o conhecimento do estado do mundo, chamado de **capacidade de percepção**, derivado da informação de sensores, deve ser usado no processo de raciocínio, para adaptar o plano a mudanças do ambiente. Neste processo, a partir de uma descrição abstrata da missão, o robô gera uma seqüência de operações de baixo nível de abstração e pontos no seu espaço de operação, que devem ser atingidos para completar a tarefa. A esta capacidade chama-se **capacidade de navegação**.

Como mostrado em COSTE-MANIERE e SIMMONS (2000) há muitos exemplos de histórias de sucesso no uso de robôs autônomos como:

- i) veículos capazes de dirigir autonomamente em rodovias por centenas de quilômetros;
- ii) a missão do Sojourner em Marte;
- iii) em aplicações na agricultura;
- iv) guias em museus.

Citou-se, na introdução, que a complexidade nos robôs autônomos é caracterizada pela necessidade de atingir metas definidas em alto nível e ao mesmo tempo ser apto a reagir face à situações inesperadas. Isso envolve a coordenação de atividades concorrentes, seqüenciais, monitoração do ambiente através de sensores e tratamento de exceções

(SIMMONS, 1998). Essas concorrências e seqüenciamentos geram a necessidade de planejamento e de sincronismo para coordenar as atividades e evitar conflitos. Os programas de controle de tarefa de robôs são normalmente difíceis de serem escritos e depurados; um dos motivos disso é que o controle de tarefas deve encadeá-las corretamente para que o robô faça suas atividades de um forma concorrente. Outro motivo é o fato do fluxo de controle não ser local; isso ocorre quando há a ocorrência de exceções, cujo tratamento pode mudar totalmente o andamento das atividades que vinham sendo executadas.

A arquitetura de *software* robótico diz respeito à forma que o sistema de controle do robô é organizado, isto é; como ele é dividido em sub-sistemas; como estes sub-sistemas interagem (COSTE-MANIERE e SIMMONS, 2000) e, dentro destes, a forma em que eventos e decisões são manipulados no decorrer da operação do robô.

As arquiteturas de software, apresentadas na próxima seção, correspondem às alternativas para se organizar o desenvolvimento de um software de um sistema robótico.

### **5.3 Arquiteturas de Software para Robôs Autônomos**

Em GAT (1997), onde é apresentado um panorama histórico das arquiteturas de *software* robótico, mostra-se que até 1990 utilizava-se como referência duas arquiteturas para o desenvolvimentos de robôs: as arquiteturas hierárquicas e as comportamentais ou reativas.

A arquitetura hierárquica foi formalizada por Saridis (FRASER,1994), onde a inteligência do controle é distribuída de acordo com o princípio de aumento da abstração com o aumento da inteligência. Assim o planejamento é executado pelos níveis mais altos e o controle é executado pelos níveis mais baixos. Há 3 níveis básicos de controle: organizacional (planejador), coordenação e pilotagem. Os componentes desta hierarquia possuem os seguintes papéis (FRASER,1994):

- O nível planejador aceita e interpreta comandos e as devidas realimentações do resto do sistema, define tarefas a serem completadas e particiona essas tarefas numa seqüência adequada de subtarefas;
- O nível de coordenação recebe instruções do nível planejador e informações de realimentação do sistema para cada subtarefa a ser executada e coordena a

execução do nível inferior. O coordenador pode assinalar índices de desempenho e funções de penalização e também definir uma condição de encerramento das subtarefas;

- O nível de ação de controle ou pilotagem envolve a execução de uma certa ação de movimento e requer a execução de uma estratégia de controle.

Essa estrutura é também referida como Arquitetura de Planejamento - Navegação - Pilotagem (FRASER,1994). Segundo este mesmo autor, inúmeros robôs autônomos foram implementados segundo esta arquitetura, tais como o Robô Hilaire, o CMU Rover, e o Navlab. As principais limitações desse tipo de arquitetura estão relacionadas ao fato de que o fluxo de informação também está estruturado hierarquicamente dentro dessas camadas. Cada nível da hierarquia recebe comandos do nível superior e informações sensoriais do nível inferior, fechando uma malha de realimentação. A frequência de atualização das informações diminui com a subida de nível hierárquico. As restrições de comunicação entre os níveis torna muito difícil a conexão entre o sensoriamento em baixo nível e a ação de processos de alto nível, gerando tempos longos de processamento, uma baixa integração entre sensores e fraca reação a eventos inesperados (FRASER,1994).

Na arquitetura reativa ou comportamental (chamada de *Subsumption* em Inglês) não existe um nível supervisor. Existe um grupo de programas de comunicação conhecidos como “comportamentos”, análogos ao termo “comportamental” em psicologia, que correspondem a comportamentos para o mundo exterior (FRASER, 1994). Dados e controle são distribuídos entre todos os níveis, cada nível processa sua própria informação sensorial e os comandos que são enviados aos atuadores não são passados entre níveis como no caso hierárquico e alguns comportamentos se subordinam(do Inglês *subsume* — subordinam) aos outros.

O que se observava na prática é que implementações realizadas com uma arquitetura *Subsumption*, baseadas principalmente nos trabalhos de BROOKS (1986), apesar de utilizarem computadores mais simples, pareciam possuir mais inteligência por apresentarem respostas mais rápidas. Assim, segundo GAT (1997), enquanto um robô desenvolvido segundo uma arquitetura Hierárquica, estava parado num processo de planejamento até para as ações mais simples, um robô simples, utilizando uma arquitetura Comportamental parecia possuir maior inteligência devido à sua rapidez de ação.

No final da década de 1980, a arquitetura Comportamental também começava a mostrar suas falhas, principalmente na sua incapacidade de seguimento de planos e também devido à falta de repetibilidade na sua operação GAT (1997). A proposta realizada por vários autores, entre os quais se inclui ARKIN (1989), foi a da integração das características dessas duas arquiteturas, gerando o que se chama hoje de arquitetura híbrida.

Segundo GAT (1997), essa integração foi realizada quase que simultaneamente por três pesquisadores: Connel, Gat e Bonasso. A arquitetura de Connel, chamada de SSS (CONNELL, 1992), a de Gat chamada de ATLANTIS (GAT, 1992) e a de Bonasso que veio a se chamar 3T (BONASSO, 1995).

As soluções possuem o seguinte denominador comum: um mecanismo de controle baseado em realimentação com estreita interação entre os sensores e atuadores do robô através de diferentes algoritmos de controle, um planejador deliberativo que opera num ciclo mais lento que o mecanismo de controle e um mecanismo de sequenciamento, que conecta os dois primeiros componentes.

O uso de arquitetura de três camadas já é padrão de fato em robótica (FRASER, 1994, GAT, 1997, COSTE-MANIERE e SIMMONS, 2000), persistindo diferentes denominações para os três diferentes mecanismos das arquiteturas conforme mostra a Tabela 5.2.

Arquitetura	Tipo do Mecanismo		
	Mecanismo de controle	Mecanismo de sequenciamento	Planejador deliberativo
ATLANTIS	<i>Controller</i>	<i>Sequencer</i>	<i>Deliberator</i>
3T	<i>Skill</i>	<i>Skill Manager</i>	<i>Planner</i>
LAAS	<i>Controller</i>	<i>Executive</i>	<i>Planner</i>
Simmons	<i>Behavior</i>	<i>Executive</i>	<i>Planner</i>

Tabela 5.2: Denominações para os diferentes mecanismos nas arquiteturas híbridas.

No decorrer desta tese utiliza-se a terminologia adotada por SIMMONS (1998) com a seguinte tradução para o Português: *Behavior* é Comportamento, *Executive* é Executivo e *Planner* é Planejador.

## **5.4 Arquiteturas Híbridas Utilizando Três Camadas**

Esta sessão resume vários aspectos ligados a arquiteturas de *software* robótico híbridas utilizando três camadas. Tratam-se brevemente aspectos ligados aos estados, à forma de implementação, à operação dos diferentes mecanismos e à extensão dessa arquitetura para o caso de multi-agentes.

### **5.4.1 Estados**

Nas ASR híbridas, é de fundamental importância a existência de estados que caracterizem a situação do robô (GAT, 1997). Nessas arquiteturas a memória do estado interno tem diferentes papéis em cada um dos seus diferentes mecanismos: nos Comportamentos há pouca ou nenhuma lembrança sobre os estados internos; no Executivo há memória sobre os estados passados; já no Planejador os algoritmos realizam previsões sobre estados futuros.

### **5.4.2 Implementação**

A implementação dos três componentes de uma arquitetura híbrida pode ser realizada em diferentes processos numa aplicação multi-tarefa, ou encadeados adequadamente numa aplicação mono-tarefa. No caso multi-tarefa os processos devem poder interagir quando necessário.

### **5.4.3 Comportamentos**

Os Comportamentos são normalmente escritos em linguagem de alto nível como “C”, sendo sua ativação ou desativação coordenada pelo Executivo. Este tem como papel selecionar quais comportamentos são ativados. Em alguns casos isso envolve a determinação da função de transferência que o(s) controlador(es) deve(m) usar num determinado instante, suprindo os seus parâmetros para sua operação.

### **5.4.4 Executivo**

O Executivo deve ter informação sobre a história passada do sistema. Em função de contingências que ocorrem no ambiente de operação do robô, dessa história, deve ser capaz de comandar o seqüenciamento dos comportamentos. Isto pode ser implementado pelo chamado “seqüenciamento condicional” (GAT, 1997).

O seqüenciamento condicional é análogo ao modelo de “instruções” que pode ser utilizado pelas pessoas para execução de tarefas. Ele contém a base computacional para codificar as instruções em relação às contingências. Assim, o Executivo deve possuir construções para responder a contingências, gerenciar múltiplas tarefas paralelas de interação, através de um seqüenciamento condicional. Essas características podem ser obtidas em programas escritos em “C” (GAT, 1997). Entretanto, as construções tornam-se complexas nessa linguagem, sendo benéfico o uso de linguagens especiais, como os RAP (do Inglês *reactive action packages* —pacote de ações reativas) (GAT, 1997), os PRS (do Inglês *procedural reasoning system* — procedimento de raciocínio procedural). Essas linguagem especiais podem ser uma extensão de alguma outra linguagem, como a ESL (*Executive Support Language* —linguagem para apoiar o executivo) (GAT,1997), que é uma extensão do Lisp ou; a TDL (SIMMONS, 1998), que é uma extensão do C++.

### ***A linguagem TDL***

Simmons e seu grupo vem desenvolvendo infra-estruturas para o desenvolvimento de ASR para robôs autônomos há mais de 10 anos, inicialmente através da TCA – *Task Control Architecture* (SIMMONS,1994) e, posteriormente, com a TDL, uma evolução da TCA. Via de regra, ambas utilizam o IPC - *Inter Process Communication* (SIMMONS, 1994) como base para a comunicação entre os diferentes aplicativos; o IPC é citado brevemente na Seção 7.3.

No contexto de uma cooperação firmada com o Dr. Simmons, e de uma missão de doutorado sanduíche realizada na CMU com financiamento de CNPq, a TDL tornou-se a base da arquitetura do *software* robótico do dirigível autônomo. Por outro lado, conforme explicitado na Seção 7.3, o modelo do IPC não foi adotado, sendo a comunicação assegurada pelos *sockets* do protocolo TCP / IP, tanto no Ambiente de Desenvolvimento e Operação, tratado no Capítulo 7, quanto na ASR do dirigível.

A linguagem TDL, baseada na linguagem ESL (SIMMONS, 1998), é uma extensão do C++ concebida para compor o nível executivo de uma arquitetura de software robótico, com recursos para a decomposição de tarefas, a sincronização, o monitoramento e o tratamento de exceções (i.e. contingências).

A execução de árvores de tarefas ou *task trees* constitui um dos recursos para a decomposição tarefas, como está ilustrado na Figura 5.1.

As *task trees* são estruturas que refletem a hierarquia das tarefas a serem executadas. Os dois tipos básicos de nós das árvores são *goals* (nós intermediários) e *commands* (folhas). Os *goals* representam tarefas relativamente complexas que devem ser divididas em sub-tarefas, que são por sua vez divididas em outros *goals* ou *commands*. Os *commands* são tarefas simples que podem ser executadas diretamente pelo robô.

Cada nó é um pedaço de código que executa uma determinada ação, a qual pode ser auto-suficiente ou pode ser subdividida, gerando filhos na árvore. A criação de filhos é feita dentro do código do nó através de uma cláusula *spawn* que pode ser seguida de uma cláusula *with* onde são colocadas as restrições sobre a sub-tarefa.

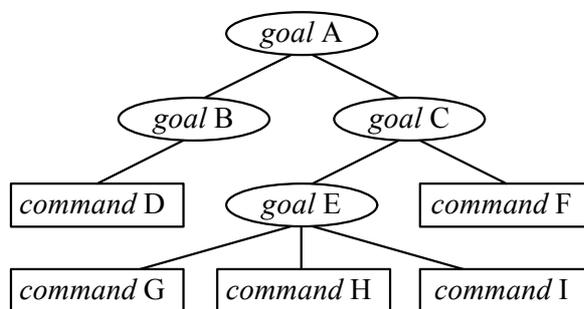


Figura 5.1: Árvore de tarefas.

As ações podem conter código condicional, iterativo e até mesmo recursivo, mas a árvore resultante é sempre uma árvore simples, pois os filhos, mesmo quando são nós do mesmo tipo, são nós inteiramente novos na árvore. Note-se que, como a criação dos filhos se dá dentro do código do nó, o mesmo código pode gerar árvores bem distintas em diferentes execuções.

Outro tipo de recurso é denominado de *monitor*. Enquanto *goals* e *commands* são tarefas que são executadas apenas uma vez a cada vez que são chamados (através de uma cláusula *spawn*), *monitors* são tarefas repetitivas. O *monitor* pode, para uma mesma chamada, ser instanciado várias vezes, periodicamente ou não. Quando o *monitor* é periódico, ele é disparado a cada intervalo de tempo fixo definido na chamada. O *monitor* não periódico é disparado conforme a restrição de ativação que é também definida na sua chamada.

A TDL possui recursos para tratamento de exceções que operam de forma bem parecida com os recursos similares do C++. Quando é constatada uma situação de exceção, seu tratamento consiste na passagem do evento de nó para nó até que seja encontrado um

que contenha um *exception handler* capaz de tratá-la. *Exception handlers* podem ser adicionados incrementalmente nas tarefas durante a programação. Isso facilita o desenvolvimento de um sistema, onde não foram previstas antecipadamente todas as situações possíveis, que podem ser muitas, de anormalidades, correspondendo em muitos casos à ocorrência de contingências.

#### **5.4.5 Planejador**

No Planejador, escrito em linguagem convencional de alto nível, são executados os processos de planejamento que requerem o uso intensivo de recursos computacionais. A interface entre o executivo e o planejador pode ser realizada de duas diferentes formas (GAT, 1997), na primeira ele produz planos para a executivo e na segunda ele responde a questionamentos do executivo. No caso da arquitetura 3T (BONASSO, 1995) a primeira forma é utilizada enquanto que na arquitetura ATLANTIS a segunda.

#### **5.4.6 Arquitetura de Três Camadas para o Caso Multi-agentes**

Em SIMMONS (2000) o conceito de arquitetura de três camadas é estendido explorando o paradigma de multi-agentes (WOOLDRIDGE, 1995). Nesta visão, cada agente é composto das três camadas como numa arquitetura híbrida, sendo que a interação entre os agentes, visando a execução de um trabalho cooperativo, só pode ocorrer entre os mesmos níveis de uma arquitetura de três camadas; isto é, componentes dos níveis de planejamento, executivo e comportamentos só podem interagir com os correspondentes níveis dos outros agentes

### **5.5 Modelo Conceitual de Arquitetura para o Dirigível**

Aborda-se nesta seção o modelo conceitual elaborado para a arquitetura de *software* robótico do dirigível.

A Figura 5.2 apresenta o conceito da ASR do dirigível estabelecido com base na arquitetura utilizada no projeto DIRA, detalhada em SIMMONS (2000), que explora o paradigma de multi-agentes aplicados a ASR híbridas. A ASR do dirigível é híbrida e constituída de dois agentes, o Agente em Terra (AT) e o Agente Embarcado (AE), que trocam informações por meio do sistema de comunicação, neste caso o rádio-modem, sendo implementados nas estruturas de *hardware* e *software* embarcadas e em terra

descritas no Anexo 1.

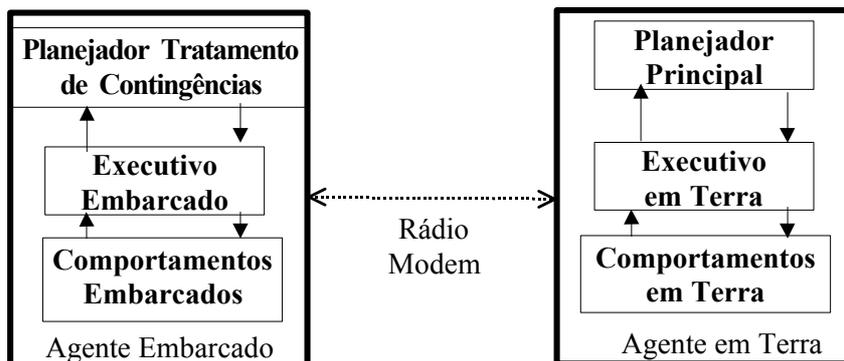


Figura 5.2: Visão geral da ASR para o dirigível robótico

O AT é responsável pela interação com o usuário / operador do dirigível com os equipamentos da Estação de Terra, enquanto que o AE é responsável pela interação com o sistema embarcado, abrangendo os sensores e atuadores bem como as funções de controle e navegação durante um vôo do dirigível. A parte deliberativa é realizada inicialmente pelo AT, que estabelece um plano inicial. Durante a execução deste plano, o encadeamento das tarefas e seu controle é realizado pelo AE.

Com relação à parte reativa da arquitetura, há basicamente dois tipos de contingências a serem tratadas: internas e externas. As internas são devidas ao funcionamento inadequado de qualquer componente das estações Embarcada ou de Terra, enquanto que as contingências externas são tipicamente geradas por perturbações vindas do ambiente onde se está voando, como o vento ou condições meteorológicas. Elas são detectadas pelo exame dos dados de telemetria gerados pelo sistema embarcado e transmitidos para a estação de terra ou por dados detectados em terra. As contingências também variam no grau de seriedade, originando, conseqüentemente, diferentes ações de recuperação com distintos níveis de complexidade.

### 5.5.1 *Agente em Terra - AT*

Como mostrado na Figura 5.2, o Agente em Terra é composto pelo Planejador Principal (PP), pelo Executivo em Terra (ET) e pelos Comportamentos em Terra (CT).

#### ***Planejador Principal –PP***

Previamente ao vôo, o Planejador Principal realiza um plano *a-priori* da missão que objetiva definir o comportamento deliberativo da aplicação. Ele recebe uma requisição do

usuário referente a uma missão definida e a transforma num *script* de vôo, expresso em termos de primitivas de navegação para o dirigível robótico; exemplos de primitivas são: decolar contra o vento até alcançar uma altitude especificada, executar uma determinada trajetória (retas, círculos, etc.) da posição atual para outra posição a uma certa altitude, pairar a determinada posição e altitude sobre uma área de interesse, executar uma trajetória ditada por algum parâmetro de interesse a ser observado pelo dirigível, ir a um determinado local e aterrissar, etc. O plano, na forma deste *script* de vôo, é enviado ao executivo de terra e então ao sistema embarcado, onde será executado.

Ressalta-se ainda que, durante o vôo do dirigível, em caso de contingência importante, o PP pode ser acionado novamente para gerar um plano alternativo, *a-posteriori*. O PP é portanto capaz de uma ação deliberativa (planejamento *a-priori*), seguida por novas deliberações em reação a contingências (planejamento *a-posteriori*). O PP está escrito numa linguagem de alto nível, no caso “C”, e é executado ao nível de usuário, com uma prioridade de execução inferior quando comparada ao sistema operacional ou aos módulos de tempo real.

### ***Executivo em Terra – ET***

A primeira função do ET, antes de iniciar um vôo, é receber o plano de navegação estabelecido pelo PP e enviá-lo ao executivo embarcado. Posteriormente, durante o vôo do dirigível, o ET monitora a execução do plano, com o objetivo de detectar diferenças entre o planejado e o *status* atual do veículo. No caso de uma diferença significativa, o ET disparará a geração de um plano alternativo para se adequar à nova situação. Por outro lado, o ET também monitora a diferença entre a situação ambiental prevista para o vôo e a situação atual: se há uma diferença significativa, indicando a mudança das condições de vôo, ele pode solicitar ao PP a realização de um replanejamento considerando a nova situação. O ET está escrito em TDL.

### ***Comportamentos em Terra – CT***

A principal função dos Comportamentos em Terra é interagir com a interface homem-máquina, com os sensores da estação de terra e com o sistema de comunicação, fazendo-os operar adequadamente. Mesmo se a finalidade dos CT é de fornecer suporte, eles também podem apresentar ações reativas para tratar contingências de baixo nível que

aparecem neste contexto. Os CT estão escritos em RTLinux e são executados como módulos de *kernel*, para satisfazer restrições de tempo real.

### **5.5.2 Agente Embarcado - AE**

O agente embarcado (ver Figura 5.2) é composto pelo Planejador de Tratamento de Contingências (PTC), o Executivo Embarcado (EE) e os Comportamentos Embarcados (CE).

#### ***Planejador de Tratamento de Contingências – PTC***

As contingências em vôo, quando o replanejamento é necessário, normalmente são tratadas pelo Planejador Principal em terra. Entretanto, se uma contingência séria acontece e, por alguma razão, há uma falha de comunicação entre a Estação Embarcada e a de Terra, o AT e o AE não serão capazes de compartilhar informações e, conseqüentemente, de estabelecer a solução necessária. Para tornar o AE capaz de tratar estas situações de emergência, uma versão simplificada do Planejador Principal é introduzida no AE. Esta versão simplificada é justamente o PTC, compatível com as limitações do sistema embarcado em termos de capacidade de processamento e memória. Da mesma forma que o PP, o PTC está escrito numa linguagem de computador de alto nível, no caso “C e é executado o nível de usuário.

#### ***Executivo Embarcado –EE***

O Executivo Embarcado recebe inicialmente um *script* da missão do Agente de Terra e o mapea num conjunto de comportamentos de suporte e de controle e navegação que têm que ser disparados, como algoritmos de controle de velocidade, de altitude e de trajetória, e coordenadas dos sucessivos pontos de trajetória. O EE acompanha então a execução do movimento comandado e, ocorrendo alguma contingência séria, dependendo do estado de comunicação com a terra, poderá disparar o PTC com o objetivo de gerar um plano alternativo de emergência. Da mesma forma que o ET, o EE é escrito em TDL

#### ***Comportamentos embarcados –CE***

Os Comportamentos Embarcados são divididos em dois grupos:

- O primeiro assegura a interação com os sensores embarcados e os atuadores do dirigível (GPS, sensores inerciais, bússola, sonda de vento, motores, vetorização

e superfícies aerodinâmicas) sendo também responsável pelas ações reativas básicas para tratar as contingências de baixo nível associadas a problemas de funcionamento dos sensores e atuadores;

- O segundo grupo engloba os comportamentos diretamente associados ao controle e navegação do dirigível.

Eles constituem um conjunto de movimentos básicos – primitivas de voo tais como: decolar, aterrissar, pairar, voar cruzeiro a uma certa velocidade / altitude, cuja combinação sucessiva deve resultar na execução da missão planejada. Tais comportamentos são executados sob o ambiente RTLinux para satisfazer requisitos de tempo real e têm ligação direta com o sistema de controle e navegação do dirigível apresentado no Capítulo 4.

## 5.6 Conclusões

Este capítulo abordou o estabelecimento de um modelo conceitual para a Arquitetura de *Software* Robótico do Dirigível.

Inicialmente se apresentou um panorama relacionado ao desenvolvimento de sistemas de controle de robôs inteligentes, com sua classificação quanto a sua capacidade de autonomia e adaptação a variações do seu ambiente de operação; posteriormente caracterizou-se os componentes genéricos de um sistema robótico, enfatizando-se a necessidade das arquiteturas de *software* robótico. Na seqüência apresentou-se um panorama das arquiteturas de *software* robótico, mostrando os motivos que levaram as arquiteturas híbridas a serem padrão de fato nos sistemas robóticos, com a caracterização dos principais componentes dessa arquitetura. Apresentou-se também a linguagem de programação TDL, utilizado como linguagem para se escrever o executivo da ASR do dirigível e uma proposta de extensão da arquitetura híbrida para um sistema multi-agente, utilizada como referência para a ASR do dirigível.

Por fim, descreveu-se o modelo conceitual concebido para a ASR do dirigível que utiliza a linguagem TDL, com arquitetura híbrida, estruturada em três camadas, usando o paradigma multi-agente. Foram descritos os componentes dos Agentes Embarcado e em Terra que compõem a base da ASR do dirigível.

## 6. SIMULADORES DE DIRIGÍVEL

### 6.1 Introdução

A partir dos modelos matemáticos de sistemas robóticos são estabelecidos ambientes de simulação, aos quais se agregam ferramentas matemáticas computacionais de análise e síntese, originando ambientes de CACSD (do Inglês *Computer Aided Control System Design*). Ambientes de CACSD de veículos robóticos permitem:

- i) a familiarização do usuário com o comportamento dinâmico do veículo;
- ii) o teste do veículo em diferentes cenários, nem sempre fáceis de serem reproduzidos experimentalmente e, principalmente;
- iii) o desenvolvimento, teste e validação inicial, de forma barata e segura, de métodos de controle e navegação, antes de sua implementação no veículo real, levando-se em consideração, inclusive, aspectos como tempo e esforço computacional requeridos para a implementação real e nível de confiança que se possa depositar num dado sistema de controle associado a partes críticas do processo i.e, sua robustez.

Neste capítulo são apresentados os simuladores de dirigível desenvolvidos na perspectiva do treinamento de pilotos e do desenvolvimento de estratégias de controle e navegação para o Dirigível Robótico. Estes simuladores são calcados no modelo matemático do AS800 apresentado no Capítulo 3. Assim, a Seção 6.2 apresenta o princípio de operação dos simuladores de veículos aéreos e a Seção 6.3 apresenta uma breve exposição da primeira versão do simulador de dirigíveis desenvolvido em ambiente MATLAB. A Seção 6.4 apresenta o CACSD desenvolvido no mesmo ambiente para fins de projeto assistido por computador dos sistemas de controle do dirigível. Pela sua importância como ferramenta utilizada nas tarefas de desenvolvimento do *software* do dirigível, a Seção 6.5 apresenta o servidor de simulação de dirigível desenvolvido utilizando MATLAB. Na seqüência, a Seção 6.6 apresenta o simulador de dirigíveis que usa Java/VRML acessível via Internet, e também utilizado como ferramenta para treinamento de pilotos. Finalmente, a Seção 6.7 apresenta o servidor de simulação de dirigíveis desenvolvido em Java baseado no simulador desenvolvido em Java/VRML e a Seção 6.8 apresenta as conclusões do capítulo.

## 6.2 Simuladores de Veículos Aéreos

Na Seção 3.3.1 foram introduzidos dois sistemas de coordenadas que são utilizados no modelo matemático do dirigível e sua simulação: o SCD, associado ao corpo dirigível e o SCI, associado a um referencial na terra, dito inercial, que permite representar a posição e atitude do veículo com relação a referenciais geodésicos usualmente utilizados nos sistemas de controle e navegação de aeronaves.

A coexistência de ambos referenciais no processo de simulação, introduz a necessidade de transformações entre ambos, o que leva a algumas particularidades no algoritmo de simulação e, em particular, no processo de integração numérica.

Em cada iteração, para a integração do modelo dinâmico e obtenção da nova situação do veículo, i.e. posição e atitude, é requerida a transformação das variáveis do SCD, para o SCI e a integração das variáveis neste último sistema, obedecendo os seguintes passos num processo iterativo (TITTERTON, 1997):

1. Cálculo dos diferentes termos de forças e do vetor de forças generalizadas resultante (forças e momentos),  $F$ , atuando no veículo no instante  $t$ :  $F = F_d(x) + A(x) + G + P$ , segundo a Equação 3.6.
2. Obtenção das acelerações generalizadas (acelerações lineares e angulares), atuando no veículo, a partir das forças generalizadas:  $\dot{x} = M^{-1} F$  (ver Equação 3.6) no SCD. Ressalta-se que embora nenhum estudo tenha sido feito sobre as condições de inversibilidade da matriz  $M$ , problema algum de perda de definição ou singularidade foi constatado em simulação.
3. Obtenção das novas velocidades generalizadas (o vetor "x") no SCD (velocidades lineares e angulares), a partir da integração das acelerações.
4. Transformação das velocidades lineares expressas no SCD para o SCI, a partir das informações de atitude do veículo na iteração anterior ( $t-1$ ), utilizando-se a Equação 3.8.
5. Obtenção da nova atitude do veículo, a partir das velocidades angulares no SCD e da atitude anterior utilizando-se a Equação 6.1.
6. Obtenção da nova posição do veículo pela integração das velocidades lineares.

O método de integração utilizado no procedimento de simulação foi o Runge-Kutta de quarta ordem disponível no MATLAB (MATHWORKS, 2001) ou o apresentado em PRESS (1992). O passo de integração adotado foi de 20 ms, o que é apropriado levando-se em conta a dinâmica do dirigível. Cabe comentar a constatação de problemas numéricos para passos de integração superiores a 30 ms.

O passo 5, que determina a nova atitude do veículo no SCI, é comumente o aspecto de mais difícil compreensão. A dificuldade está relacionada ao entendimento da propagação no tempo (integração) das diferentes representações possíveis para a atitude do veículo no SCI – como os ângulos de Euler (rolagem, arfagem e guinada), os cossenos diretores ou os quaternions (TITTERTON, 1997) em função das velocidades angulares em torno dos três eixos dos sistemas de coordenadas SCD. O simulador de helicóptero desenvolvido por RAMOS (1995), por exemplo, utiliza os ângulos de Euler nesta transformação, enquanto que o simulador desenvolvido por GOMES (1990) utiliza os quaternions. A grande vantagem dos quaternions é a inexistência de singularidades matemáticas, como acontece com os ângulos de Euler (CRAIG, 1989).

A integração dos quaternions em função das velocidades angulares é dada pela Equação 6.1. Assim, dado um quaternion com componentes  $a(t)$ ,  $b(t)$ ,  $c(t)$ ,  $d(t)$ , no instante  $t$ , considerando-se um passo de integração  $\Delta t$ , esta equação permite determinar o seu valor na próxima iteração " $t+1$ " em função das velocidades angulares  $p(t)$ ,  $q(t)$ ,  $r(t)$  (TITTERTON, 1997):

$$\begin{aligned} a(t+1) &= a(t) - 0,5 [b(t) p(t) + c(t) q(t) + d(t) r(t)] \Delta t; \\ b(t+1) &= b(t) + 0,5 [a(t) p(t) - d(t) q(t) + c(t) r(t)] \Delta t; \\ c(t+1) &= c(t) + 0,5 [d(t) p(t) + a(t) q(t) - b(t) r(t)] \Delta t ; \\ d(t+1) &= d(t) - 0,5 [c(t) p(t) - b(t) q(t) - a(t) r(t)] \Delta t. \end{aligned} \tag{6.1}$$

### 6.3 Primeira Versão do Simulador

O simulador de dirigível desenvolvido originalmente por GOMES (1990), calcado no equacionamento apresentado anteriormente, baseava-se na linguagem ACSL (ACSL, 1987), uma variante da linguagem FORTRAN. Para o desenvolvimento do ambiente de

simulação básico voltado ao projeto do dirigível robótico, foi feita a migração do código em ACSL para o MATLAB.

Neste ambiente básico de simulação, através do próprio código do MATLAB, o usuário define a situação de simulação, indicando quais são as variáveis de entrada e o tempo de simulação desejado, então comanda a execução da simulação e, após seu encerramento, escolhe o gráfico de comportamento das variáveis do dirigível para visualizar.

A Figura 6.1 mostra uma tela típica de saída gráfica do ambiente básico de simulação do dirigível, neste caso representando o movimento espacial executado pelo veículo. Estão disponíveis botões, através dos quais o usuário pode escolher as variáveis a serem mostradas, como por exemplo: posições, velocidades e acelerações lineares e angulares e algumas outras variáveis.

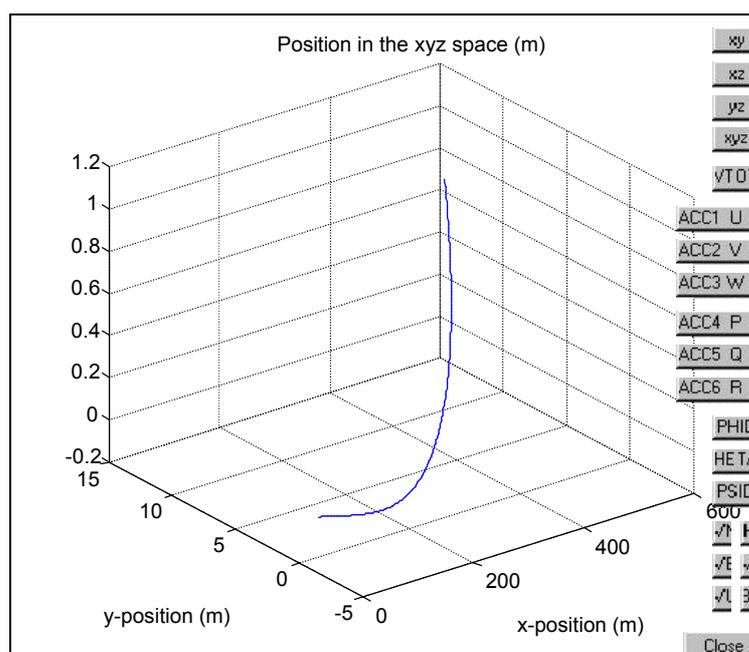


Figura 6.1: Saída típica de dados no ambiente de simulação básico MATLAB.

As principais limitações deste ambiente estão relacionadas com:

- uso da linguagem de comandos do MATLAB para configurar uma simulação — o que limitava os usuários às pessoas com conhecimentos de MATLAB;
- a dificuldade para a visualização dos resultados, e;
- a dificuldade para a verificação do comportamento do veículo durante longos períodos, com o usuário “pilotando-o”.

Estas dificuldades motivaram o desenvolvimento de ambientes com maior facilidade de uso, o que é tratado nas Seções 6.4 a 6.6 e, quando possível, de simulação com visualização do comportamento do dirigível em 3D, com uma interface através da qual o usuário pudesse “pilotar” a aeronave, o que é abordado na Seção 6.6.

#### 6.4 Ambiente CACSD para Desenvolvimento de Sistemas de Controle

O ambiente MATLAB descrito na seção anterior evoluiu<sup>3</sup>, conforme é descrito em DE-PAIVA *et al.* (1999a), com a utilização dos recursos de programação gráfica do SIMULINK (MATHWORKS, 2001), incluindo também facilidades para a inserção de malhas de controle, tornando esta nova versão um ambiente de CACSD para o desenvolvimento dos sistemas de controle e navegação do dirigível robótico.

A Figura 6.2 apresenta o diagrama funcional do bloco principal SIMULINK do ambiente de CACSD. Nesta figura podem ser vistos três blocos importantes: "Dirigível", que simula o dirigível a partir de seu modelo; "Controle Dirigível", que contém algoritmos de controle e navegação (abordados no Capítulo 7); e "Plota", que permite apresentar os resultados gráficos. Aparece também uma conexão, chamada de Xpri, que corresponde aos estados do dirigível.

As variáveis de entrada do bloco Dirigível são (ver Figura 6.2): motor1 e motor2, correspondentes às referências de velocidade de rotação " $\omega$ " dos motores; "ang-vet", o valor de comando do ângulo de vetorização  $\mu$ ; os comandos "ETA" e "ZETA" dos lemes de profundidade " $\eta$ " e direção " $\zeta$ " respectivamente, enquanto "CSI" corresponde ao comando rolagem " $\xi$ "; "u\_wind" corresponde à intensidade do vento com direção norte, "v\_wind" a intensidade do vento com direção leste e "w\_wind" a intensidade do vento com direção acima, isto expresso no SCI, respectivamente  $u_N$ ,  $v_L$  e  $w_A$  e segundo a Seção 3.3.1. Os valores do vento são transformados para o sistema de coordenadas do Dirigível por funções do bloco "Dirigível" utilizando as equações 3.8 e 3.9.

---

<sup>3</sup> Este ambiente foi desenvolvido em conjunto com o Dr. Ely Paiva, do CenPRA, utilizando o primeiro simulador como base.

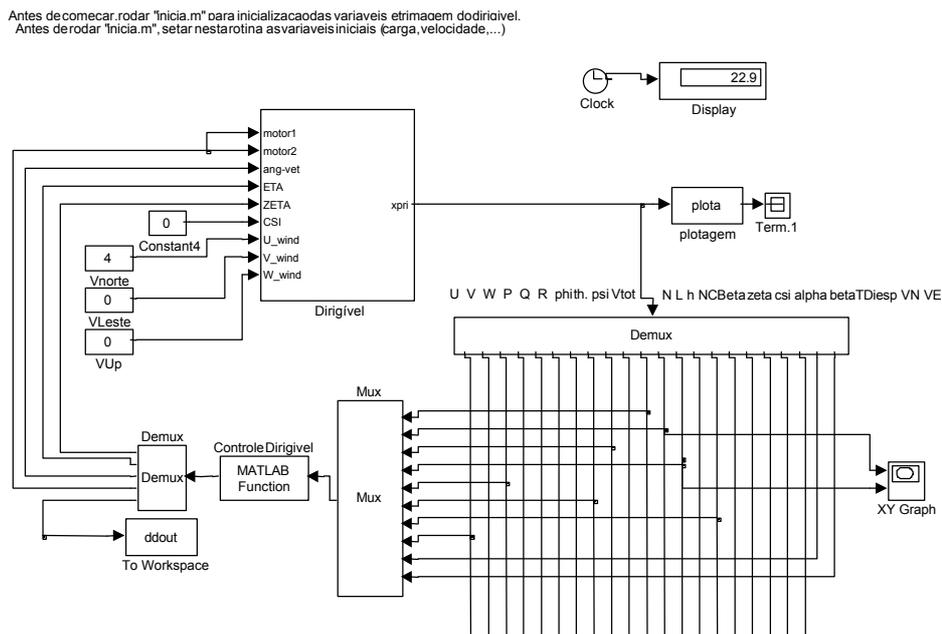


Figura 6.2: Bloco Principal do ambiente de CASD.

O módulo “Dirigível” da Figura 6.2 é detalhado na Figura 6.3, onde se mostra o diagrama em blocos SIMULINK do modelo do dirigível aparecendo as variáveis de entrada, de saída e os principais blocos do modelo propriamente dito, a maioria implementados como funções MATLAB. Assim, considerando-se o modelo descrito na Seção 3.4, aparecem os blocos que, em função dos estados e comandos do dirigível, calculam os diferentes vetores de forças presentes na Equação 3.6 e explicitados anteriormente; eles são denotados por “Propulsão”  $P$ , “Força dinâmica”  $F_d$ , “Força aerodinâmica”  $A$  e “Força Gravitacional”  $G$ . Adicionalmente aparecem dois blocos que alteram variáveis globais de simulação em função do estado do dirigível: o bloco “Matriz de massa” que atualiza a matriz massa  $M$  em função da temperatura, da densidade do ar, e simula a variação de massa dos balonetes em função da altitude; e o bloco “Ângulos de Euler”, que calcula os ângulos de Euler em função do *quaternion* atual do veículo (Equação 6.1). O vetor total de forças,  $F$ , é obtido pelo bloco “Força Resultante”. Os blocos seguintes calculam as acelerações a partir dos valores da matriz de massa e realizam a integração numérica, conforme os passos de simulação descritos no início da Seção 6.2, gerando, no bloco “Variáveis de Navegação”, a posição e a atitude do dirigível expressas no sistema de coordenadas inercial SCI fixado na terra. Por fim, são disponibilizados o vetor de estados completo do passo de simulação, chamado de  $X_{pri}$ , incluindo forças,

acelerações, velocidades e posição do veículo. São providos dois mostradores (“Disp1” e “Disp2”), que mostram a evolução das variáveis durante a simulação.

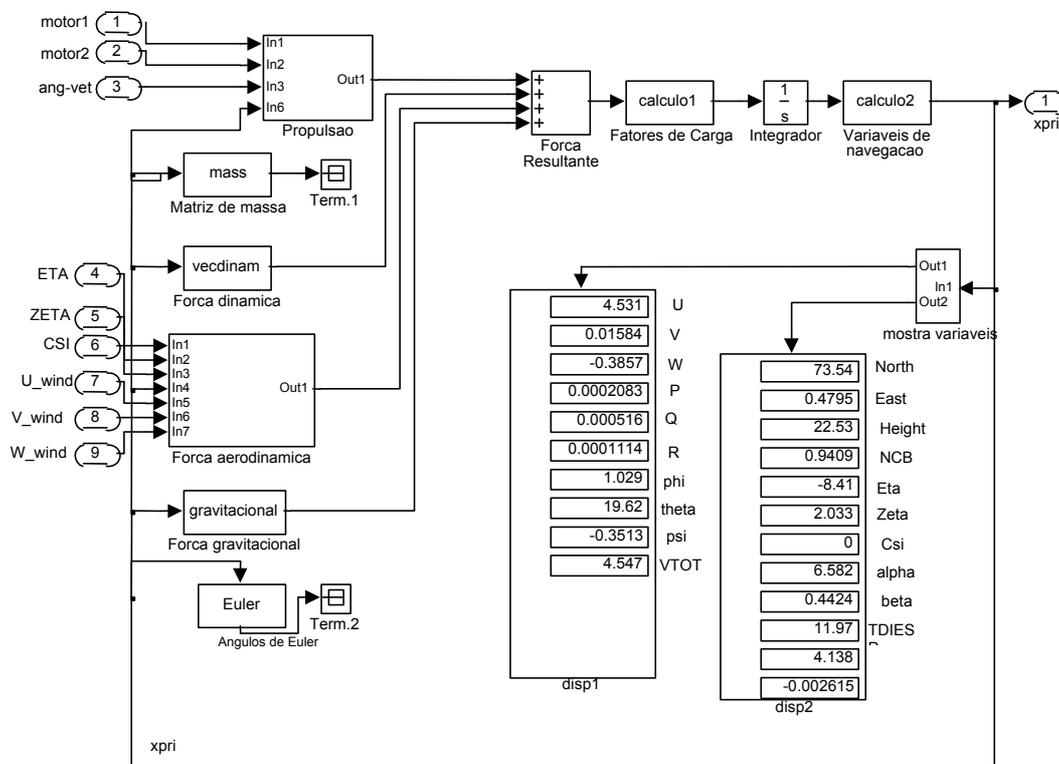


Figura 6.3: Diagrama em Simulink do modelo matemático do dirigível.

## 6.5 Servidor de Simulação de Dirigível em MATLAB

Uma vez estabelecido o ambiente de CACSD, a flexibilização de seu uso em um contexto mais abrangente de apoio ao desenvolvimento de soluções de controle e navegação para o dirigível robótico envolve o conceito de servidor de simulação. O desenvolvimento de um servidor de simulação do dirigível tem o objetivo de oferecer uma ferramenta com interface adequada à política de definição de componentes descrita no Capítulo 2, permitindo o ensaio e testes do *software* embarcado no dirigível em diferentes condições de uso, o que será descrito posteriormente no contexto do Ambiente de Desenvolvimento e Operação, no Capítulo 7.

Neste servidor, descrito em RAMOS (1999), acrescentam-se ao simulador os *sockets* do protocolo TCP/IP como interface para conexão, tendo em vista a limitada capacidade de comunicação entre processos na versão disponível do MATLAB – a versão

5. Apesar da sua simplicidade, os *sockets* proporcionam a possibilidade de construção de sistemas segundo a arquitetura cliente-servidor (SOCK-FAQ, 2001)

A solução adotada para a implementação da conexão entre o MATLAB e os recursos de TCP/IP foi através do uso de MEX-files. Mex-files são programas que podem ser escritos em linguagem C, Fortran ou na linguagem própria do MATLAB e que proporcionam uma nova funcionalidade e uma maior flexibilidade ao pacote MATLAB, acelerando sobremaneira a velocidade de execução. Aproveitando-se dessas facilidades, foram escritas rotinas de comunicação por *sockets* para as plataformas Windows e Unix na linguagem C, usando-se como referência o manual da GNU (GNU, 2001). Todavia, os MEX-files gerados são dependentes da plataforma.

Para se alcançar um nível mais alto de abstração para a implementação da comunicação por *sockets*, foram criados blocos do SIMULINK (vide Figura 6.4), que independem da plataforma, de forma a permitir que o modelo dinâmico do dirigível possa ser executado em diferentes plataformas, necessitando apenas da recompilação dos respectivos MEX-files de comunicação. Os blocos básicos, como mostrado na Figura 6.4, recebem comandos e enviam estados.

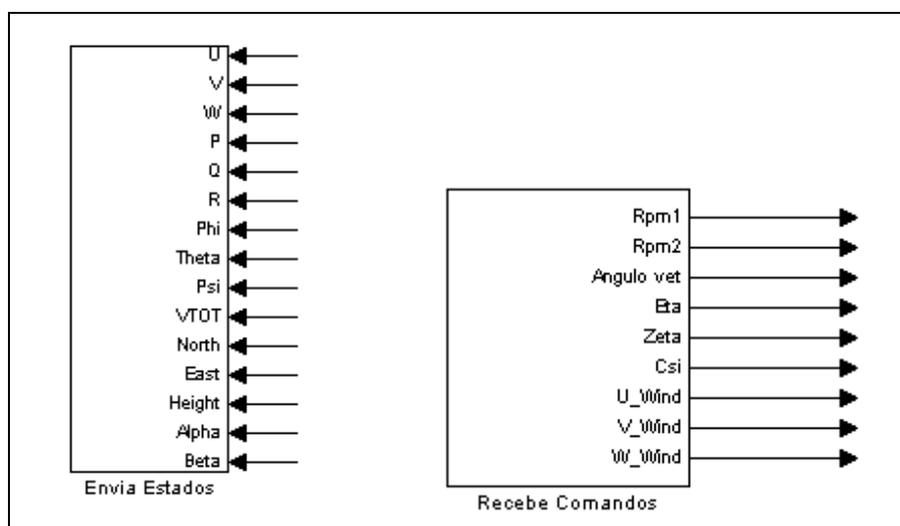


Figura 6.4: Blocos de comunicação por *socket* no Simulink.

Um protótipo inicial deste servidor de simulação, com as características acima, foi concebido e utilizado em ambas plataformas (Windows e UNIX). Devido ao maior potencial do MATLAB em sistema UNIX, quando comparado à versão Windows, decidiu-se aprimorar este protótipo inicial unicamente na sua versão UNIX; dessa maneira, os

recursos citados a partir deste ponto são somente para esta versão, adotando-se portanto uma máquina UNIX para atuar como servidor de modelo MATLAB do dirigível.

O MATLAB provê um recurso denominado Matlab Engine (MATHWORKS, 2001) capaz de ser executado remotamente e de se comunicar com outro processo via *pipe*, que é um esquema de comunicação entre processos intensamente utilizado no mundo UNIX (GNU, 2001). Com tal recurso foi desenvolvido um programa em linguagem C que recebe comandos via *socket* para controlar a execução do modelo dinâmico; há comandos para início, *reset* e finalização da execução da simulação.

Em um dos ensaios feitos para validar o desenvolvimento, usou-se o painel de comandos e instrumentos Java executado em uma máquina Windows (estes elementos são apresentados na Seção 6.6.4) e o modelo dinâmico em ambiente UNIX numa servidora SUN Ultra 2, conforme a Figura 6.5.

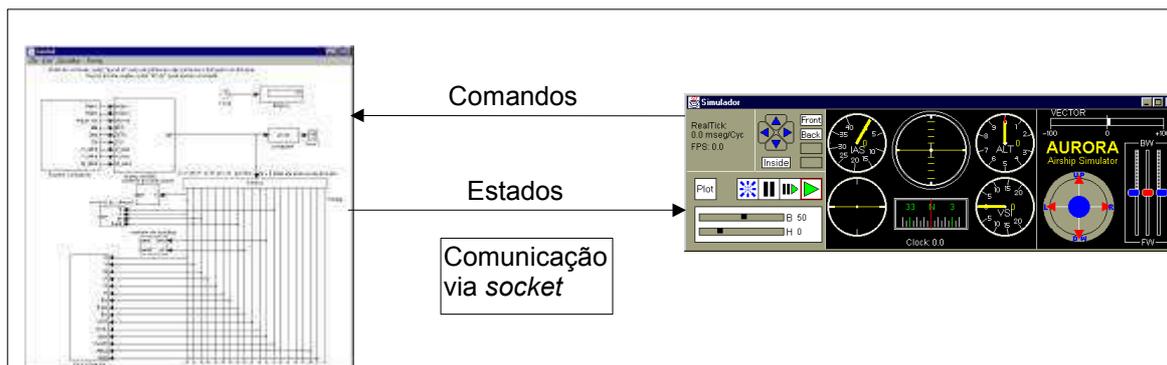


Figura 6.5: Esquema do ensaio - modelo em Matlab/Simulink e painel em Java - comunicação via *socket*.

Dessa forma, o simulador do dirigível implementado em MATLAB / SIMULINK constitui-se em um Servidor de Simulação, que pode ser acessado via *socket* obedecendo-se a um protocolo simples.

## 6.6 Simulador de Dirigíveis Baseado em Java / VRML

Apesar da utilidade e validade dos ambientes de CACSD, a experiência anterior no desenvolvimento do projeto de um helicóptero autônomo (RAMOS, 1995, RAMOS 1995a) mostrou a importância de três recursos adicionais incorporados a um ambiente de simulação:

- representação e animação 3D (não disponível na versão 5 do MATLAB) do dirigível em voo;

- uma interface com o usuário, contemplando comandos de atuação, bem como um painel para a exibição do estado do dirigível, numa forma similar as utilizadas em aviãoica;
- um painel para a configuração e controle da execução da simulação.

O advento da linguagens de programação Java (JAVA, 2001) e VRML (do Inglês *virtual reality modelling language* – linguagem para modelagem de realidade virtual) (VRML, 2001), com a promessa de uma solução multiplataforma e aberta para os vários componentes de um ambiente de simulação do dirigível, motivaram, apesar do custo adicional de ter de se realizar a migração do modelo matemático para Java, a escolha destas para comporem o ambiente, atendendo aos propósitos especificados anteriormente.



Figura 6.6: Interface do ambiente de simulação Java/VRML.

O ambiente de simulação resultante (RAMOS, 1999a) é ilustrado na Figura 6.6, onde a interface do simulador é composta de um *Applet* em Java, integrado em uma simples página HTML, que permite a um usuário, localizado em qualquer parte do mundo, acessar sua URL. A interface carrega todos os arquivos e classes necessárias para o simulador operar no computador do usuário e a simulação é inicializada. Ressalta-se que alguns componentes mostrados nesta página são utilizadas para compor a Interface Homem-Máquina do dirigível, o que é descrito no Capítulo 7.

Na Figura 6.7, que mostra a estrutura do simulador, aparecem os principais componentes responsáveis por sua operação: o Mundo Virtual VRML, e as classes escritas em Java: Simvrml, sockcomm, console, runmod e interfac, e a EAI (*External Authoring Interface*) (EAI, 2001). A EAI é uma biblioteca que acompanha o *plug-in* VRML e permite que programas escritos em Java e JavaScript controlem o conteúdo de um mundo VRML. Os controles disponíveis incluem: criação e remoção de objetos, operações com: rotações, translações, mudanças no ponto de vista do usuário, mudanças de propriedades, etc.

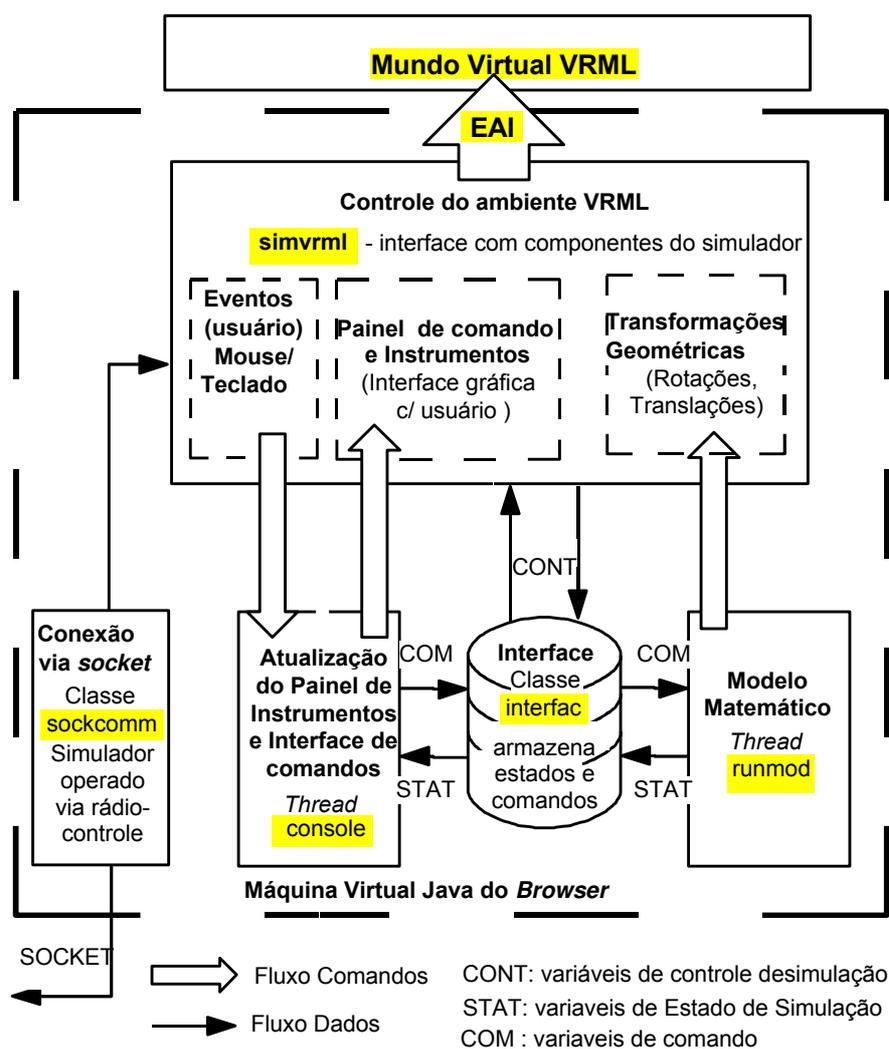


Figura 6.7: Estrutura e componentes principais do ambiente Java / VRML de simulação.

### 6.6.1 Mundo VRML

O mundo VRML é visto no canto esquerdo superior da Figura 6.6. O dirigível é modelado como um elipsóide alongado, com propulsores cilíndricos em cada lado, e possui superfícies aerodinâmicas feitas de poligonais. A vista “*default*” do veículo é como se ele

fosse visto do chão; o usuário, entretanto, pode embarcar em um voo virtual do dirigível clicando no botão “*inside*” embaixo dos comandos do *pan-tilt* e passando a observar o mundo do ponto de vista do piloto. O mundo VRML também mostra o solo feito de plano tecelado, e o céu, nublado para mais realismo. O dirigível é mostrado ao longo do tempo de simulação nas posições e orientações providas pela integração do modelo dinâmico, segundo o referencial inercial SCI. O mundo VRML pode ser também representado em 3 dimensões a partir por exemplo de um mapa topográfico da região a ser sobrevoada pelo dirigível; isso permite um “voo virtual” anterior à missão.

#### **6.6.2 Classe que Contém os Estados e Comandos do Simulador – classe *interfac***

A classe “*interfac*” é o repositório das informações do simulador, contendo variáveis estáticas de controle da simulação tais como: o modo de operação do simulador (por exemplo contínuo ou passo a passo), o estado da aquisição de dados (ligado/desligado), bem como variáveis dinâmicas congregando os comandos e estados do dirigível, cujos valores passados são armazenados em um *buffer*. O uso de instâncias desta classe permite a troca de várias informações entre os demais componentes do simulador.

#### **6.6.3 Classe de Controle do Dirigível – classe *simvrml***

*Simvrml* é a classe principal e a primeira a ser chamada pelo navegador quando a página do simulador é acessada. Ela é responsável pela inicialização de todas as outras classes e pela conexão do mundo VRML ao *Applet* Java, começando a simulação.

Este componente inclui os métodos responsáveis pela atualização da posição e orientação (dentro do mundo virtual) e pela definição do ponto de vista do usuário, a cada passo de simulação. Isto permite ao usuário a visualização dos movimentos que estão sendo executados pelo dirigível.

#### **6.6.4 Console do Dirigível – classe *console***

*Console* é a classe Java da interface homem-máquina do dirigível. Ela inclui os comandos do simulador, os comandos do dirigível, os instrumentos do simulador, os comandos de controle dos balonetes e de variação da flutuabilidade e, também, os comandos de aquisição de dados.

Os comandos do simulador vistos no canto inferior esquerdo da Figura 6.6, permitem ao usuário executar a simulação passo a passo ou continuamente, adquirir dados para posterior visualização gráfica e reinicializar a simulação. É também possível mudar o ponto de vista da representação VRML de uma visão a partir do solo para o ponto de vista de uma câmera virtual embarcada no dirigível (botão *inside*), cuja orientação pode ser comandada mediante os botões denotados por *pan-tilt* logo acima. Adicionalmente, o usuário recebe informações do ciclo atual de simulação (em ms/ciclo), tempo global de simulação e o número de quadros do mundo VRML que estão sendo mostrados por segundo.

Os comandos do dirigível vistos no canto inferior direito da Figura 6.6 permitem ao usuário:

- Fazer com que o dirigível execute movimentos de arfagem (para baixo ou para cima) ou de guinada (para a esquerda ou para a direita) segurando o cursor na região circular da janela.
- Ajustar a velocidade dos motores mantendo o botão esquerdo do cursor pressionado, movimentando-o para cima e para baixo na região mais a direita na figura.
- Vetorizar os motores para cima e para baixo pressionando o cursor no topo desta região.

Em qualquer caso, as coordenadas do cursor são transformadas nos comandos correspondentes e enviadas como entradas para o cálculo do modelo dinâmico.

Balonetes e fluutuabilidade são comandados durante a simulação mediante as duas réguas mostradas no lado inferior esquerdo da Figura 6.6. A régua superior permite mudanças na divisão percentual do volume entre os balonetes dianteiros e traseiros, nota-se que no AS800, há um só balonete, de tal forma que este comando não se aplica. Estes valores afetam o equilíbrio do dirigível. A régua inferior é usada para definir a fluutuabilidade do dirigível, o que corresponde a diferença entre a força aerostática e o peso do dirigível. Na figura o dirigível está voando com fluutuabilidade neutra (“ $H=0$ ” na figura) e os balonetes têm metade do volume total em cada um deles.

O painel do simulador inclui, ainda, métodos para mostrar os instrumentos de vôo do dirigível na forma de um painel comumente encontrado em aviônica, emulando altímetro, bússola, horizonte artificial, etc., permitindo assim acompanhar seu comportamento durante um vôo.

Uma janela de aquisição de dados (que não está presente na Figura 6.6) permite a apresentação gráfica das variáveis armazenadas durante a simulação. A implementação atual permite mostrar os gráficos das posições (cartesiana e atitude), velocidades e acelerações do dirigível.

A classe *Console* é a responsável pela interpretação dos eventos de cursor e teclado gerados pelo usuário e capturados pelo *browser*. Estes eventos são transformados nos comandos de simulador descritos acima. Esta classe é também responsável pela atualização do painel de instrumentos, usando dados de simulação.

O painel de instrumentos do simulador é apresentado no contexto gráfico criado por *simvrml*, mas sua atualização é feita pela classe *console*. Uma figura que contém todos os desenhos do painel é carregada quando o simulador está sendo inicializado. Com o objetivo de ganhar em desempenho, somente são atualizadas as regiões que são efetivamente alteradas no painel de instrumentos (dados numéricos e mostradores dos instrumentos de vôo).

#### **6.6.5 Execução do Modelo Matemático – classe *runmod***

*Runmod* é a classe responsável pela execução do modelo dinâmico do dirigível. É uma *thread* Java responsável pela execução de um passo de simulação do modelo matemático, dando como resultado os novos estados do dirigível. A execução pode ser contínua ou passo a passo, dependendo da configuração escolhida pelo usuário. Esta classe também é responsável pelo comando de atualização do mundo virtual.

O modelo matemático é um objeto em Java que a todo ciclo recebe como entradas os comandos de arfagem e guinada, rotação e ângulo de vetorização dos motores, a percentagem de ocupação dos balonetes e flutuabilidade, e a direção e intensidade do vento, e provê como saída a posição e orientação do dirigível. A saída do modelo dinâmico é usada para atualizar o mundo VRML.

### 6.6.6 Componentes Adicionais

A classe de *plotagem* fornece recursos para a apresentação gráfica de variáveis tais como velocidades, acelerações, orientação, posição e altitude, etc. O usuário pode também definir a direção e velocidade em relação à terra do vento que age sobre o dirigível.

### 6.6.7 O Simulador como uma Ferramenta de Treinamento

Seja durante toda a fase de desenvolvimento do dirigível robótico, seja em situações de emergência durante a sua operação, é necessária a existência de uma unidade de comando remoto (RCU do Inglês remote control unity)) que permita a operação manual do veículo, conforme descrito na Seção 2.2.3.

A forma mais segura e barata de treinar pilotagem via RCU é o uso de simuladores de vôo. Em simuladores de vôo, conectando-se a RCU a um computador, como acontece em aeromodelismo (SIMUL, 2001), o usuário pode familiarizar-se com a interface de rádio controle e com o comportamento do veículo. Assim decidiu-se desenvolver os componentes de *hardware* e *software* adicionais para tornar o simulador uma ferramenta de treinamento de pilotagem do dirigível. A Figura 6.8 apresenta o esquema desenvolvido.

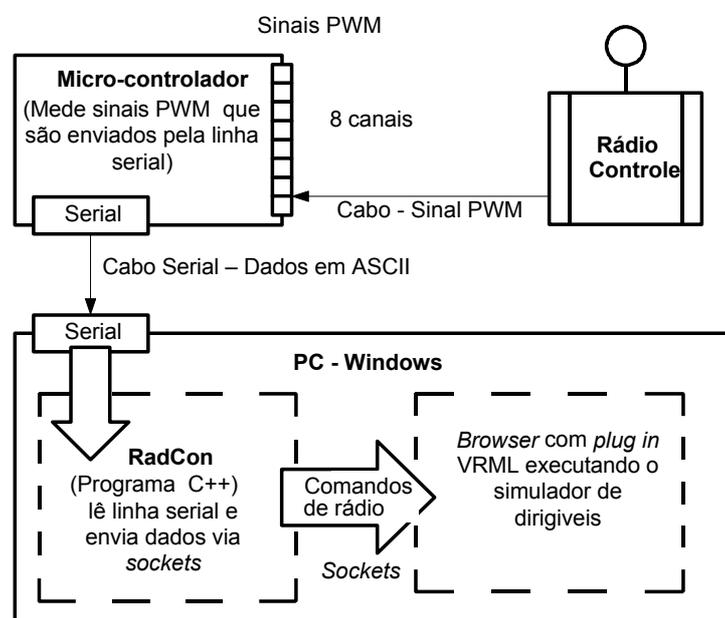


Figura 6.8: Componentes acionais de *software* e *hardware* para treinamento de pilotagem.

A RCU envia comandos em PWM (modulação por largura de pulso) para o micro-controlador, que os converte em dados ASCII, e os envia a qualquer computador externo através de interface serial. No computador, um programa escrito em C++ lê os dados da

porta serial, e os envia via *socket* para o simulador baseado em Java/VRML. Esta implementação foi escolhida devido a características de segurança da máquina virtual Java executando num *browser*, o qual não permite uma comunicação direta com a porta serial a menos que o desenvolvedor esteja bastante familiarizado com os mecanismos de segurança dos *browsers* e seja capaz de superá-los.

Atualmente a ferramenta de treinamento de pilotos permite, a qualquer integrante do projeto, pilotar um dirigível simulado, de uma forma bastante próxima do dirigível real, e usando exatamente a mesma RCU que ele utiliza para controlar remotamente o dirigível quando de vôos reais.

## 6.7 Servidor de Simulação Baseado em Java

Aproveitando-se classes do simulador de dirigíveis em Java/VRML e desenvolvendo-se, adicionalmente, três classes, obteve-se um servidor de simulação que oferece a mesma interface de comunicação do servidor em MATLAB, com a vantagem de poder ser executado em qualquer máquina que execute aplicativos escritos em Java. As três classes implementadas como *threads* Java destinam-se a: envio de estados de simulação via *sockets*; recebimento via *sockets* dos comandos de atuação do dirigível e; recebimento via *sockets* dos comandos de simulação.

Adicionalmente, o servidor de Simulação escrito em Java apresenta uma interface com o usuário na forma de um painel de aviônica pelo uso da classe *console* mostrando a evolução dos estados de simulação da mesma forma que o painel de instrumentos.

Optou-se por excluir o visualizador 3D do simulador de dirigíveis tendo em vista problemas de compatibilidade do “*plug in*” VRML em algumas plataformas, garantindo assim a disponibilidade de um servidor de simulação de dirigíveis em diferentes plataformas.

Este servidor de simulação, foi utilizado para o desenvolvimento da Arquitetura de *Software* Robótico do dirigível, realizada na Carnegie Mellon University (CMU)-EUA, conforme é apresentado no Capítulo 9.

## 6.8 Conclusões

Este capítulo apresentou os simuladores e ambientes de CACSD desenvolvidos para auxiliar no processo de projeto de sistemas de controle e navegação, para a construção e teste do ambiente de desenvolvimento e operação do dirigível e para o treinamento de pilotagem.

Inicialmente apresentou-se a primeira versão do simulador desenvolvido em MATLAB, que evoluiu para um ambiente CACSD de projeto de sistemas de controle de dirigíveis. Em seguida apresentou-se um servidor de simulação desenvolvido em Simulink que utiliza os resultados obtidos no desenvolvimento deste ambiente CACSD.

Na seqüência apresentou-se um simulador de dirigíveis baseado em Java/VRML, disponibilizado via Internet, que além de contribuir para a familiarização com o comportamento do dirigível, constitui ferramenta de treinamento de pilotos. Finalmente foi apresentado o servidor de simulação desenvolvidos em Simulink e em Java para apoiar desenvolvimento e para teste do ambiente de desenvolvimento e operação do dirigível.

Modelo Matemático e simuladores são componentes do Ambiente de Desenvolvimento e Operação do dirigível abordado no próximo capítulo. Eles apoiam o ciclo de desenvolvimento de estratégias de controle e navegação, objeto do Capítulo 8, e a concepção de uma Arquitetura de *Software* Robótico com características deliberativas e reativas, tratada no Capítulo 9.

Ressalta-se que o modelo matemático e os simuladores resultantes são qualitativamente representativos da dinâmica do dirigível e, portanto, aptos à utilização posterior anunciada. Por outro lado o modelo se mostra ainda impreciso conforme já foi enfatizado no Capítulo 3.

## 7. AMBIENTE DE DESENVOLVIMENTO E OPERAÇÃO

### 7.1 Introdução

Dentre os recursos de suporte para o desenvolvimento de um dirigível robótico enumerados na Seção 2.2.3, um deles consiste no estabelecimento de um ambiente que contemple as fases de desenvolvimento e operação. Isso abrange as fases de desenvolvimento tanto do sistema de controle e navegação do dirigível robótico quanto de aplicações do veículo e sua operação durante as missões que desempenha, de forma tal a permitir sua evolução, de dirigível robótico teleoperado para dirigível robótico autônomo. O ambiente deve possibilitar diferentes modos de utilização, como simulado e real. No modo simulado, ele deve congrega simuladores e ferramentas de CACSD baseadas em modelo matemático do dirigível, como os descritos no Capítulo 6, com o objetivo de se definir estratégias de controle, testar conceitos, arquitetura de *software* robótico, etc.. No modo real, o ambiente destina-se ao ajuste de algoritmos de controle e navegação e seus parâmetros, bem como à operação do veículo quando da realização de vôos.

A operação real do veículo envolve também a interface homem-máquina (IHM), situada na Estação de Terra, que constitui o elo entre o operador e o dirigível robótico. Ela deve:

- i) fornecer ao operador dados de telemetria do dirigível e do ambiente em que ele se encontra;
- ii) prover comandos para que o operador possa efetuar a pilotagem do veículo;
- iii) permitir a integração de algoritmos de planejamento de trajetórias e de controle de movimentos e o ajuste dos parâmetros de sintonia desses algoritmos e;
- iv) fornecer recursos para a definição, acompanhamento e eventual modificação da missão do veículo.

Neste capítulo, a IHM efetivamente implementada contempla os itens (i) a (iii), enquanto que o item (iv) restringe-se à definição *a-priori* da trajetória por pontos de passagem e perfil de altitude e, durante a execução da missão, ao acompanhamento da localização e orientação do veículo. Níveis mais elevados de abstração para a definição *a-*

*priori* da missão, seu acompanhamento e eventual modificação, no contexto ainda de simulação, são discutidos no Capítulo 8.

A Seção 7.2 apresenta a motivação para o desenvolvimento de ambientes para o desenvolvimento e operação de robôs, seguida apresentação de um ambiente de desenvolvimento e operação para veículos robóticos subaquáticos que tem sido utilizado como referência dos demais ambientes de desenvolvimento de sistemas robóticos autônomos, principalmente subaquáticos, e apresentam-se referências a outros ambientes de desenvolvimento e operação, a maioria baseados no primeiro.

A Seção 7.3 apresenta a visão geral do Ambiente de Desenvolvimento e Operação do dirigível construída a partir da proposição conceitual apresentada na Seção 2.4. A Seção 7.4 detalha a estrutura real do Ambiente. A Seção 7.5 apresenta o protótipo inicial e seus componentes, incluindo a interface homem-máquina de operação. A Seção 7.6 apresenta a evolução realizada no Ambiente como consequência de duas dissertações de mestrado associadas a esta tese. Finalmente, a Seção 7.7 apresenta as conclusões do capítulo.

## **7.2 Ambientes de Suporte ao Desenvolvimento e Operação para Veículos Robóticos**

A motivação para a existência de ambientes de desenvolvimento do *software* de controle de robôs provém das dificuldades operacionais, custos e riscos envolvidos na realização de ensaios. No caso de robôs subaquáticos o custo operacional para ensaios é muito alto (BRUTZMAN, 1994). Mesmo robôs móveis como o Nomad (XR400, 1999), que não tem um custo elevado para a realização de testes práticos, exceto riscos de acidentes devido a erros de *software*, dispõem de um modo de operação simulado, que permite ao usuário testar o seu *software* de aplicação, previamente ao uso real. Além disso, tais ambientes reduzem tanto o tempo de desenvolvimento em si (incluindo também o teste e validação) quanto os trabalhos de implantação (entende-se implantação como a palavra mais próxima da palavra inglesa *deployment*).

A seguir são apresentados vários ambientes utilizados para o desenvolvimento de robôs, principalmente robôs subaquáticos, que requerem ambientes com mais recursos. A maioria dos ambientes de veículos subaquáticos, baseia-se no ambiente desenvolvido pela NPS (*Naval Post Graduate School* da marinha americana), e apresentam como denominador comum:

- i) o suporte à operação simulada, híbrida ou real;
- ii) modelos matemáticos para simulação do veículo;
- iii) interface com o usuário através de recursos para visualização de dados e visualização 3D do veículo e de seu ambiente;
- iv) modelagem de sensores e atuadores e;
- v) integração dos componentes através de recursos do protocolo TCP/IP.

### 7.2.1 *Ambiente para o Veículo da NPS*

Visando aumentar a confiabilidade do *software* produzido na NPS, (BRUTZMAN, 1994) projetou e implementou um ambiente virtual para suportar as fases de desenvolvimento e operação de veículo autônomo subaquático. O ambiente baseia-se no chamado RBM (*Rational Behavior Model* - modelo de comportamento racional), dividido em níveis: estratégico, tático e de execução. Esses níveis são análogos aos encontrados na pilotagem de navios. O nível estratégico corresponde ao planejamento de longo prazo, executado pelo comandante, o nível tático corresponde aos planejamentos de médio e curto prazos, executado pelo oficial de cabine, e o nível de execução implementado pelos operadores de sonar e de leme. São utilizadas linguagens de programação diferentes para cada nível. O nível estratégico usa tipicamente *backward chaining* e é implementado em linguagem como Prolog. O nível intermediário, ou tático, é orientado a objetos, usa troca de mensagens para comunicação e está programado em linguagem Ada clássica. Finalmente, o nível de execução usa a linguagem 'C'.

Na Figura 7.1, extraída de BRUTZMAN (1994), estão mostrados os componentes deste ambiente, incluindo: o modelo dinâmico do veículo; as interações do veículo com o modelo do mundo e outros componentes. Observa-se nesta a conexão dos elementos utilizando *sockets* do protocolo TCP/IP.

## NPS AUV Underwater Virtual World

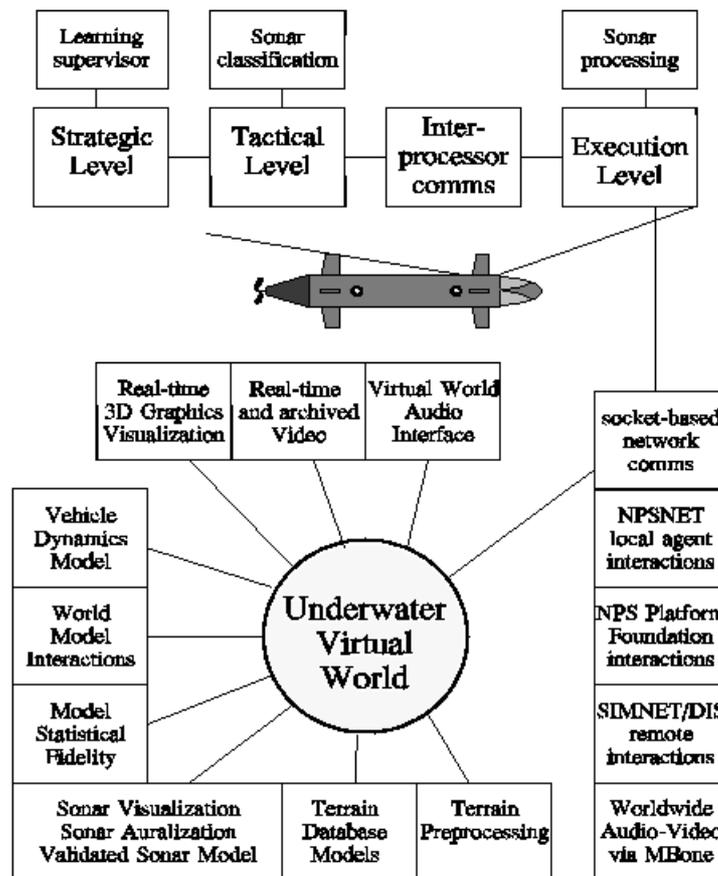


Figura 7.1: Componentes do mundo virtual da *Naval Post Graduate School*.

### 7.2.2 Outros Exemplos de Ambientes de Desenvolvimento e Operação

Listam-se abaixo referências a outros ambientes de desenvolvimento e operação para veículos subaquáticos e um para veículos aéreos:

- IFREMER (*Institut Français de Recherche pour l'Exploitation de la Mer* - Instituto Francês de Pesquisa para a Exploração do Mar) desenvolve o veículo subaquático destinado a operação autônoma, chamado de Vortex, que utiliza a arquitetura chamada ORCAD (BORRELLY *et al.*, 1998);
- DEVrE - *Distributed Environment for Virtual and/or Real Experiments for Underwater Robots* (RIDAO *et al.*, 2001) desenvolvido na Universidad de Girona, Espanha,
- DVECS - *Distributed Virtual Environment Collaborative Simulator* (CHOI, 2001) desenvolvido pelo *Autonomous Systems Laboratory* da Universidade do Hawaii,

- AMGSSS (AMGSS, 1996): o ambiente de desenvolvido no projeto AMGSSS (*Air-Mobile Ground Security and Surveillance System*) para veículos robóticos aéreos utiliza redes sem fio e o protocolo TCP/IP como recursos de comunicação e ensaios prévios de componentes de um helicóptero autônomo.

### 7.3 Visão Geral do Ambiente

O ambiente de desenvolvimento e operação de veículo subaquático *Virtual World for an Autonomous Underwater Vehicle*, desenvolvido pela NPS (BRUTZMAN, 1994), constitui a base para o desenvolvimento de uma solução similar voltada ao dirigível robótico, considerando-se, no entanto, que o cenário de desenvolvimento no caso do dirigível apresenta um caráter evolutivo, com especificações que também evoluem no decorrer do tempo, como já mostrado no Capítulo 2. Adota-se também como princípio geral o uso de componentes que utilizem interfaces padronizadas de comunicação e integração com outros componentes (FRAZER, 1994), de tal forma que, na medida do possível, eventuais alterações de um componente executor de um serviço, não afetem os outros componentes usuários desse serviço (FRAZER, 1994).

A comunicação entre os componentes do ambiente utiliza os *sockets* do protocolo TCP/IP (GRIFFIN, 1998) como infra-estrutura de integração. Eles representam a camada de aplicação de um protocolo clássico, isto é, encapsulam o meio físico e o nível de enlace de uma conexão. Assim, por meio deste recurso, torna-se transparente para o usuário se um outro componente que ele deva conectar para interagir está sendo executado na mesma máquina ou em outra máquina. O nível de abstração da conexão via *socket* é baixo, cabendo ao usuário a responsabilidade sobre o procedimento de estabelecimento e administração da conexão, incluindo até aspectos ligados à conversão da representação dos valores numéricos em diferentes plataformas, como por exemplo, entre plataformas X86 da Intel e Solaris. Outro aspecto para a escolha dos *sockets* relaciona-se ao seu desempenho pois, dada a sua simplicidade, a não inclusão de camadas adicionais de software para implementar níveis de abstração mais elevados incrementa a velocidade das conexões.

Além da solução adotada, foram consideradas outras alternativas para compor a infra-estrutura de integração: os chamados *middleware* como o CORBA (CRB, 2001) ou o IPC, associado à linguagem TDL desenvolvida na CMU (SIMMONS, 1998) - ver Seção 2.2.3. Ambos encapsulam os detalhes do processo de conexão e sua administração,

oferecendo assim uma visão mais abstrata para o usuário, seja através de objetos distribuídos no caso de CORBA, ou de mensagens no caso do IPC. A primeira alternativa, baseada no uso do CORBA, foi descartada em 1998 por razões de custo. Já a segunda foi descartada por não ser uma solução padronizada.

Assim, no contexto do presente trabalho, a configuração conceitual para o ambiente de desenvolvimento e operação é conforme apresentado na Figura 2.4 do Capítulo 3, onde é explicitamente adicionado o meio de comunicação, através de conexões dos componentes do ambiente a redes implementando o protocolo TCP/IP, como é apresentado na Figura 7.2, abaixo.

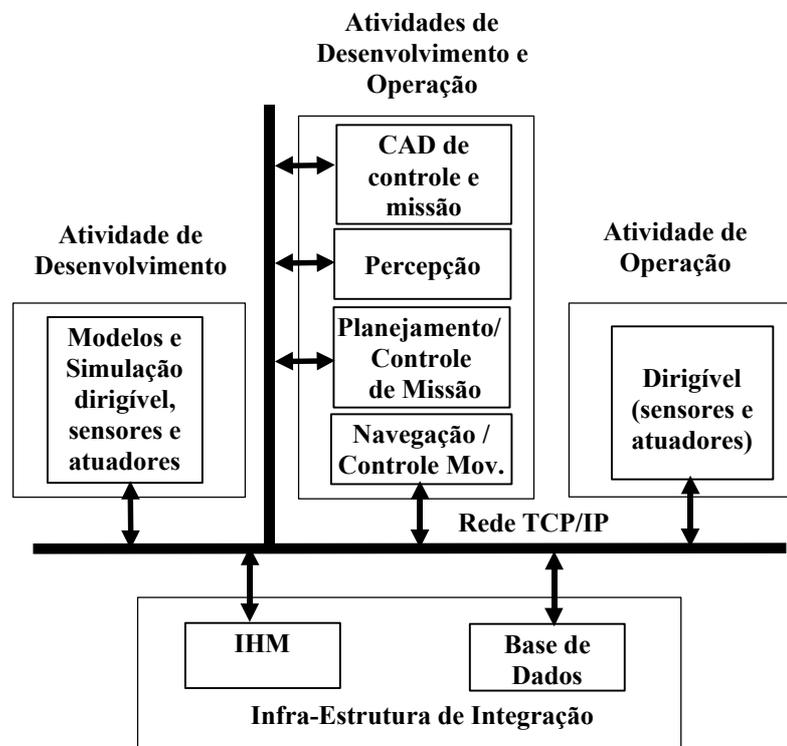


Figura 7.2: Estrutura conceitual do Ambiente de Desenvolvimento e Operação.

#### 7.4 Estrutura do Ambiente de Desenvolvimento e Operação.

A Figura 7.3 mostra o diagrama geral do ambiente de Desenvolvimento e Operação, apresentado em RAMOS (1999). Nesta figura, podem se observar os componentes básicos e de suporte do sistema robótico do dirigível, distribuídos entre a estação embarcada e a estação de terra, além dos componentes de simulação e de projeto assistido por computador, todos conectados através do protocolo TCP/IP.

Observa-se também na figura a existência de três exceções a esse padrão de conexão via TCP/IP. Trata-se das conexões entre a estação de terra e a estação embarcada, chamadas de enlaces de dados e enlace de imagens, além do enlace do RCU. Os dois primeiros correspondem, respectivamente, à conexão via rádio-modem para troca de dados entre as duas estações e à conexão de vídeo que consiste no envio de imagens da estação embarcada para a estação de terra<sup>4</sup>. O enlace denotado por RCU (unidade de comando remoto) corresponde ao recurso de pilotagem manual do dirigível através de rádio-controle convencional, como discutido no Anexo 1.

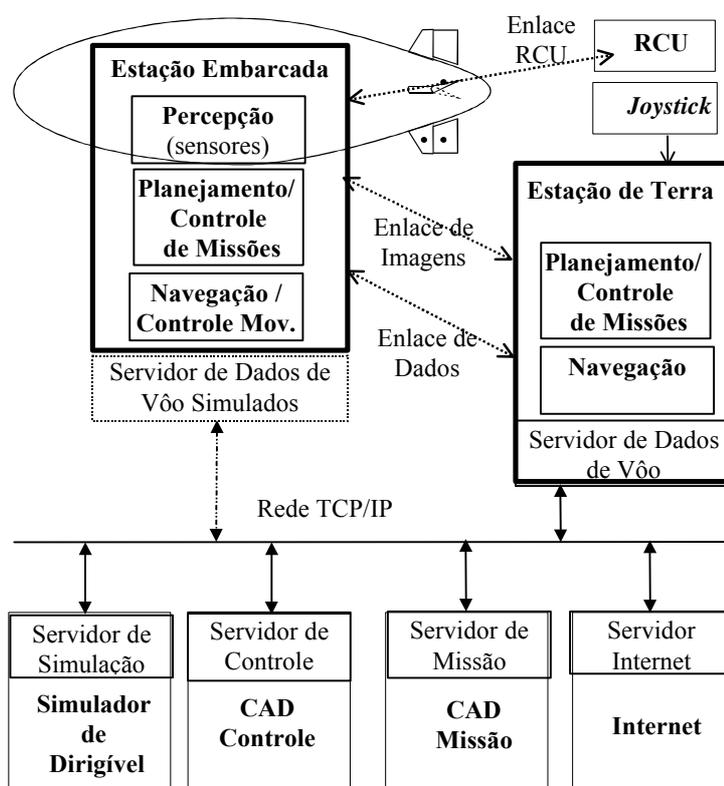


Figura 7.3: Diagrama Geral do Ambiente de Desenvolvimento e Operação.

A Figura 7.3 mostra que em cada componente conectado à rede, aparecem servidores com os mesmos nomes. A função genérica destes servidores é a de tratar as conexões externas para os serviços dos componentes, solução classicamente adotada numa arquitetura cliente-servidor, possibilitada pelo protocolo TCP/IP. O servidor de dados simulados tem papel reservado quando se usa o ambiente no modo simulado de operação, sendo portanto desconectado em vô real conforme será mostrado na Seção 7.5.3. O

<sup>4</sup> Mesmo no caso do rádio-modem, poderiam ser usadas as interfaces clássicas do protocolo TCP/IP para interface serial RS232 como o PPP ou o SLIP (GRIFFIN, 1998), mas isso diminuiria a velocidade de comunicação praticamente à metade, devido ao *overhead* do protocolo.

sistema operacional utilizado neste ambiente é o RTLinux, apresentado na Seção A1.4.1, sobre o qual é montado o arcabouço de software descrito na Seção A1.4.2. Os componentes de *hardware* da estação embarcada e estação de terra são aqueles detalhados na Seção A 1.3.

## 7.5 Protótipo do Ambiente de Desenvolvimento e Operação

Para avaliar e validar o conceito descrito anteriormente, foi implementado um protótipo inicial do ambiente de desenvolvimento e operação. O protótipo contempla a estação embarcada, a estação de terra, o servidor de simulação do dirigível e recursos para telemetria dos estados do dirigível, que se conectam ao servidor de dados da estação de terra.

### 7.5.1 Infra-estrutura de Hardware e Comunicação do Protótipo

Para que o dirigível possa operar de modo autônomo são necessários sensores e infra-estruturas de processamento tanto embarcadas quanto em terra, além de recursos para a comunicação entre ambas. Uma visão geral deste conceito é fornecida pela Figura 7.4. A idéia geral é realizar a conexão entre os elementos embarcados e de terra via rádio-modem, mantendo-se no entanto conexão do rádio controle convencional de forma independente da anterior. A prioridade da conexão quando de operação autônoma é via rádio-modem, enquanto que o rádio controle é usado para pilotagem manual durante o desenvolvimento do projeto, tendo também a função de possibilitar uma alternativa de *back-up*, garantindo assim a segurança da operação do dirigível.

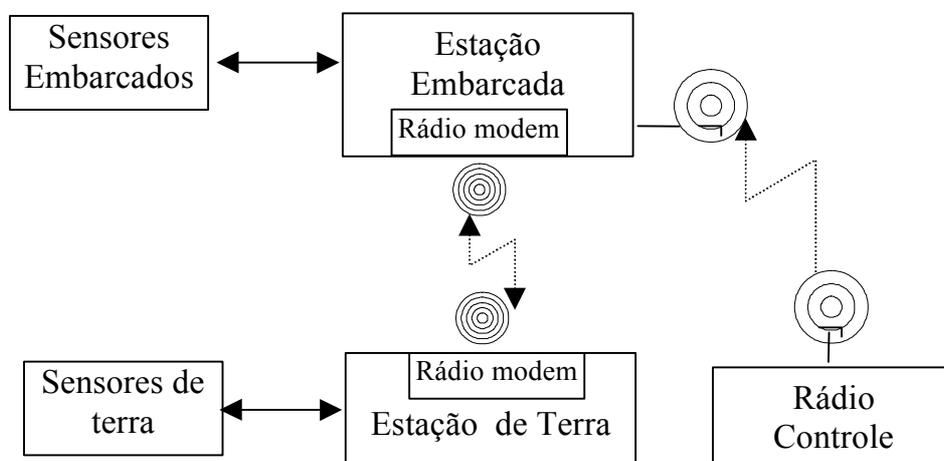


Figura 7.4: Esquema de conexão dos equipamentos.

O Anexo I fornece detalhes da infra-estrutura desenvolvida, que constitui uma contribuição importante deste trabalho.

### 7.5.2 Componentes de Software do Protótipo

A Figura 7.5 apresenta os componentes do protótipo, onde, à esquerda, aparecem os componentes relacionados à estação embarcada e à direita aparecem os relacionados à estação de terra. Observa-se a existência de duas alternativas para a conexão entre as estações: a primeira via rádio-modem (no modo real de operação); a segunda, via RTFifos (no modo simulado de operação), usada durante a fase desenvolvimento.

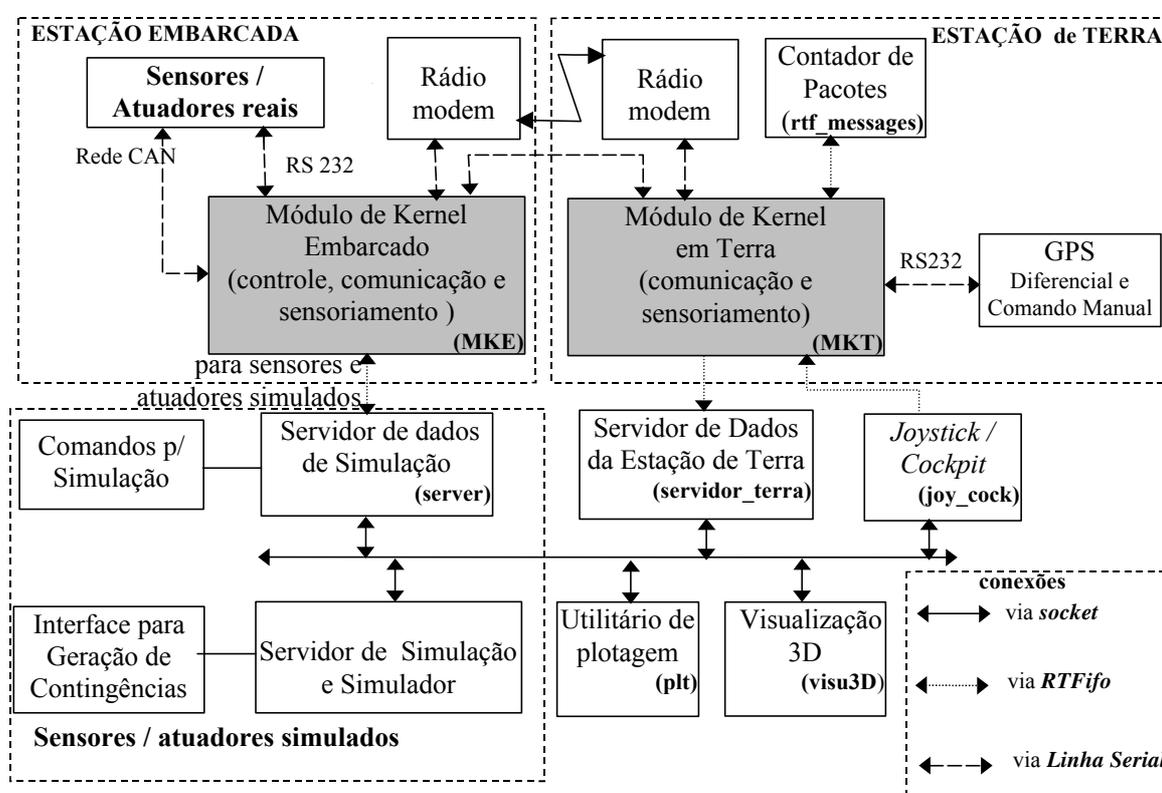


Figura 7.5: Componentes do protótipo do Ambiente de Desenvolvimento e Operação.

Visando atender os requisitos de tempo real, os componentes responsáveis pela interação com os sensores e atuadores, pela comunicação terra-ar-terra e pelo controle e navegação do veículo, foram desenvolvidos como módulos do *Kernel* e executados sob o RTLinux; eles são os chamados Módulo do *Kernel* Embarcado (MKE) e Módulo do *Kernel* em Terra (MKT). A decisão de se incluir um módulo de tempo real em terra, o MKT, visou garantir o atendimento da comunicação terra-ar e também a periféricos essenciais como o

dispositivo de pilotagem manual, conforme descrito na Seção A1.3.1.1, e o GPS Diferencial .

Os componentes da estação embarcada e da estação de terra são detalhados a seguir.

### 7.5.3 *Estação Embarcada*

O diagrama geral do *software* da estação embarcada, mostrado à esquerda da Figura 7.5, apresenta três componentes principais: o MKE, aplicativo RTLinux que constitui o núcleo da estação embarcada; e os denominados *sensores/atuadores reais* e *sensores/atuadores simulados* associados ao simulador do dirigível. O MKE segue os padrões do arcabouço dos aplicativos em RTLinux como descrito na Seção A1.4.2. A existência dos componentes ditos simulados visa possibilitar o teste do *software* de controle e navegação embarcado em modo simulado de operação do ambiente; já no modo real de operação, considera-se como sensores/atuadores os componentes efetivamente presentes no dirigível.

A operação da maioria das rotinas do MKE é independente do modo de operação - se simulado ou real - as únicas rotinas afetadas pela diferenciação quanto ao modo de operação, através de parâmetros de configuração, são as que interagem com os sensores e atuadores. No modo real de operação, as rotinas de tratamento dos sensores/atuadores interagem com a interface à qual o periférico está conectado, por exemplo RS-232 ou rede CAN. No modo simulado, as rotinas de tratamento de sensores/atuadores, no lugar de interagirem com os periféricos, são desviadas para interagir com os sensores/atuadores simulados, neste caso, o servidor de simulação de dirigível.

Pelas limitações de recursos de comunicação para módulos escritos em RTLinux, e do *kernel* Linux, que não podem realizar conexão via *sockets*, a conexão do MKE com o servidor de simulação do dirigível requer a passagem das informações por um intermediador, denominado *servidor de dados de simulação*, como pode ser visto na Figura 7.3 e na Figura 7.5. Assim, é aberta uma conexão via RTFifos entre o MKE e o servidor de dados de simulação, e este abre uma conexão via *sockets* com o servidor de simulação, na máquina em que este último estiver sendo executado. O servidor de dados recebe os comandos via RTFifo e os envia por meio de *sockets* para o simulador e vice versa,

retornando os estados do dirigível virtual. Um recurso adicional do servidor de dados de simulação é oferecer uma interface para comandar a operação do simulador.

Associada ao simulador do dirigível pode estar acoplada, ainda, uma interface para a definição e geração de contingências de vôo, requeridas quando se deseja testar a resposta do sistema a falhas em componentes ou alteração nas condições de vôo, como as mudanças de direção e intensidade do vento. Esta ferramenta é utilizada no ensaio da arquitetura de *software* robótico para o dirigível tratado no Capítulo 9.

Os parâmetros de chamada do MKE definem como este irá operar, se em modo simulado ou real. O MKE executa ciclicamente tarefas de:

- Leitura das medidas feitas pelos sensores reais ou simulados;
- Leitura das informações e comandos enviadas da estação de terra, como os dados do GPS diferencial e/ou os comandos do sistema de pilotagem ou do sistema de controle da execução de missões;
- Execução dos algoritmos de controle e navegação;
- Envio dos comandos para os atuadores simulados ou reais;
- Envio para a estação de terra dos dados de telemetria (leituras dos sensores e os estados do dirigível) e de sua missão, via rádio-modem (no caso real) ou mesmo o rádio-modem ou RTFifos (no caso simulado)

#### **7.5.4 Estação de Terra**

O diagrama geral apresentando o *software* da estação de terra pode ser visto na Figura 7.3 e detalhado à direita da Figura 7.5. Nele aparecem seus principais componentes:

- O módulo de *kernel* da estação de terra (MKT);
- O servidor de dados da estação de terra (servidor-terra);

detalhados a seguir.

#### **O Módulo de Kernel da Estação de Terra –MKT**

Principal componente da estação de terra o módulo de *kernel* (MKT) tem como uma das suas funções realizar a interface com a estação embarcada, através da leitura do canal de comunicação entre as estações, executando ciclicamente tarefas de:

- Leitura das informações de telemetria e do desenrolar das missão enviadas pela estação embarcada via rádio-modem e o envio destes dados, via uma RTFifo, para o servidor de dados da estação de terra;
- Leitura das medidas feitas pelos sensores localizados em terra, como o GPS diferencial, e envio destes dados à estação embarcada;
- Leitura dos comandos de vôo e missão gerados em terra e envio destas informações para estação embarcada;
- Envio do número de pacotes de rádio-modem recebidos para o mostrador do número de pacotes.

O MKT é configurável, de tal forma que nos parâmetros da sua chamada define-se sua forma de operação, real ou simulada. Caso seja operação simulada, define-se também se será utilizado rádio-modem ou uma RTFifo para a interação com a estação embarcada.

O MKT usa dispositivos eficientes de software para leitura/escrita de linhas seriais RS-232, como o RTCom do RTLlinux (RTLINUX, 2001) que realiza a leitura das linhas seriais do rádio-modem, do comando manual via estação de terra e do GPS diferencial.

A interação do MKT com os outros dispositivos, como o servidor de dados da estação de terra e o contador de pacotes de comunicação, é feita via RTFifos.

### ***Servidor de Dados da Estação de Terra***

O servidor de dados da estação de terra, que aparece na Figura 7.5 representado por "servidor-terra", tem o papel de repassar os dados de telemetria para componentes conectados via protocolo TCP/IP. Os dados dos sensores, estados do dirigível e dados de andamento da missão, provenientes da estação embarcada, são enviados do MKT para o servidor de dados da estação de terra via uma RTFifo. O papel deste servidor é receber esses dados e repassa-los a diferentes clientes que os requeiram. Assim, o compartilhamento dos dados de telemetria via um servidor de *sockets* permite aos vários componentes da interface homem-máquina utilizar este recurso para recebimento desses dados como é apresentado a seguir.

### 7.5.5 Interface Homem - Máquina

Os dados de telemetria e de evolução da missão, recebidos na estação de terra, quando visualizados em tempo real através da Interface Homem-Máquina (IHM), permitem ao operador acompanhar o andamento e progresso da missão, reprogramá-la caso situações interessantes ocorram, e até mesmo assumir o comando do dirigível caso surjam eventos inesperados. Neste primeiro protótipo da IHM usam-se conceitos e implementações estabelecidos no ambiente de simulação do dirigível em JAVA/VRML apresentado anteriormente na Seção 6.6.

#### *Armazenamento de Dados de Vôo*

O armazenamento dos dados de telemetria e de evolução da missão é realizado por um cliente do servidor de *sockets* da estação de terra. Este cliente é um programa escrito em "C" que armazena todos os dados em um arquivo, gravando cada pacote de dados juntamente com a hora em que foi recebido, sendo o tempo contado em centésimos de segundo desde que o processo foi inicializado.

#### *Visualização de dados de vôo*

A visualização dos dados é realizada por um segundo cliente do servidor de *sockets* da estação de terra que consiste num *script* escrito em Tcl/Tk cujo aspecto visual é mostrado na Figura 7.6. Este *script*, referenciado como "plt" na Figura 7.5, conecta-se ao servidor por um *socket* e mostra na forma de gráficos e em tempo real, as leituras de telemetria recebidas. O usuário pode escolher quais variáveis são mostradas, além de um gráfico de latitude x longitude, que retrata o movimento do dirigível no mundo. Caso se disponha de um mapa digitalizado da região de vôo, pode-se colocá-lo como fundo deste gráfico, permitindo a visualização do movimento do veículo sobre o mapa em tempo real. Os gráficos são *widgets* da extensão BLT (TCL, 2001) da linguagem Tcl/Tk, que implementam *auto-scroll*, *zoom*, *auto-scale*, e outras funcionalidades. Um arquivo de configuração define os campos do pacote de dados permitindo assim fácil adaptação em caso de mudança de formato do pacote devido, por exemplo, à inclusão de um novo sensor.

Além de desenhar os gráficos o *script* mostra continuamente algumas variáveis de estado críticas do dirigível, como a situação do GPS e do DGPS, a altitude e a temperatura interna da estação embarcada, e outros.

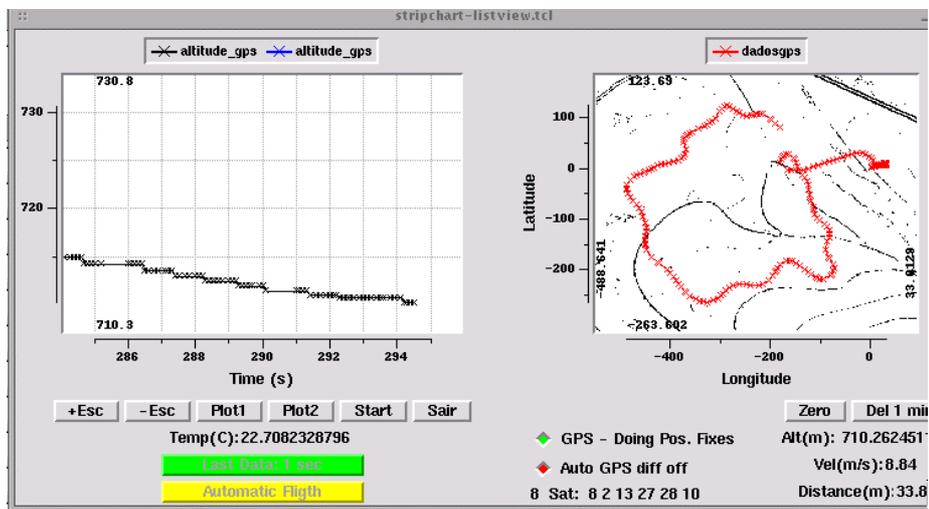


Figura 7.6: Componente gráfica da estação de terra com visualização de seus estados.

### ***Painel de Instrumentos***

A visualização dos dados de telemetria através de um painel de instrumentos similar aos freqüentemente encontrados em aviação, é realizada por um programa escrito em Java (*joy\_cock*), cujo aspecto visual está mostrado na Figura 7.7. Este programa abre um *socket*, e conecta-se com o servidor de dados da estação de terra. Ele utiliza como base a classe console desenvolvida para o simulador de dirigível, descrita na Seção 6.6.4, e mostra a evolução em tempo real do estado do dirigível, em instrumentos do tipo horizonte artificial e bússola. Se configurado da mesma forma que no simulador de dirigível, ele pode enviar comandos para operação manual do dirigível, utilizando, neste caso, uma RTFifo para enviar os comandos para o MKT.



Figura 7.7: Painel de instrumentos e painel de comando via mouse.

### Repetição de vôos

A repetição de vôos é feita por um servidor de *sockets* que possui a mesma interface que o servidor de dados da estação de terra, só que em vez de ler dados de RTFifos, este software, implementado por um *script* Tcl/Tk, lê um arquivo de dados de vôo previamente gravado e permite ao usuário exibi-lo novamente (Figura 7.8). Botões similares aos de um videocassete (*Play, Fast Forward, Stop, etc.*) permitem o controle da transmissão pelo *socket* dos pacotes armazenados. O programa tem um contador de tempo, e leva em conta o tempo armazenado juntamente com cada pacote, para transmitir os pacotes na mesma escala de tempo em que foram recebidos, permitindo uma reprodução realista do vôo. Os mesmos dispositivos IHM já descritos podem ser utilizados para visualização do vôo, possibilitando a apresentação gráfica de dados e outros recursos que forem adicionados.

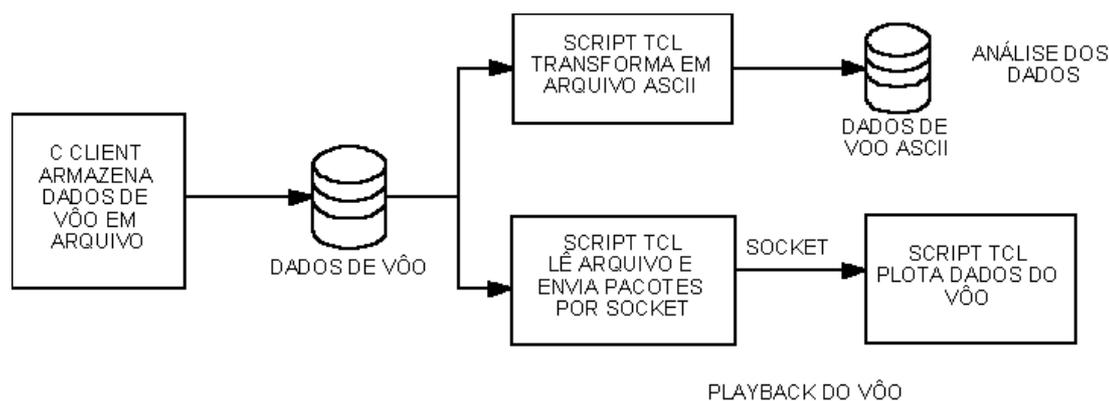


Figura 7.8: Esquema de *playback* de vôos.

É importante notar que este servidor de repetição de vôos não necessita ser modificado em caso de mudanças no formato do pacote, quando outros sensores são acrescentados ao sistema; isto porque supõe-se que o arquivo de dados de vôo contenha pacotes já no formato requerido pelo *script* que desenha os gráficos.

Além da ferramenta de repetição, foi desenvolvido um outro *script* Tcl/Tk que gera, a partir do arquivo de dados de vôo, um arquivo ASCII que contém os nomes das variáveis gravadas, seguidas pelos seus valores. Um arquivo ASCII desta forma é compatível com inúmeros programas, inclusive o ambiente de CACSD do dirigível descrito na Seção 6.4.

### **7.5.6 Resultados do Desenvolvimento e Uso do Protótipo**

Este primeiro protótipo do ambiente de desenvolvimento e operação, compreendendo o *hardware* e *software* básico das estações de terra e embarcada e a comunicação entre ambas, foi utilizado tanto no modo de simulação – ao longo do desenvolvimento dos componentes acima citados, com seus respectivos programas e algoritmos de controle e navegação – quanto no modo real, compreendendo testes de telemetria, sintonia dos parâmetros dos algoritmos de controle e navegação e culminando com a realização do primeiro vôo automático do dirigível com trajetórias contínuas.

Com esta evolução e resultados validou-se o conceito geral deste ambiente, que é calcado no uso de módulos de *kernel* em RTLinux e de *sockets* do protocolo TC/IP para conectar componentes, operando em modos simulado e real. Ele permitiu também avaliar as especificações iniciais e, a partir daí, gerar novas especificações.

Uma avaliação do protótipo indicou ser ele de difícil manutenção e evolução, indicando a necessidade de redefinição dos componentes, utilizando-se metodologias de desenvolvimento de *software* para garantir a evolução do sistema de forma mais estruturada. A adoção destas metodologias e a geração de uma segunda versão do ambiente constituiu o objeto de duas dissertações de mestrado, cujos principais aspectos serão apresentados resumidamente a seguir.

## **7.6 Evolução dos Componentes do Ambiente de Desenvolvimento e Operação**

Nas dissertações de MAETA (2001) e MIRISOLA (2001), que abordam e detalham temas originários e inseridos nos trabalhos desta tese, o protótipo do Ambiente de Desenvolvimento e Operação sofreu vários aprimoramentos. Maeta abordou a evolução da Estação Embarcada e em particular o MKE. Mirisola abordou a evolução da Estação de Terra enfocando o MKT, outros componentes a ela associados e a inclusão, na IHM de operação, de um sistema de programação de tarefas por pontos de passagem, constituindo um módulo básico de especificação de missões.

A principal deficiência apresentada pelo protótipo inicial do ambiente, principalmente do MKE e do MKT, residia na dificuldade de se acrescentar novas funcionalidades, fazendo com que o processo de manutenção e evolução fosse complexo e

ineficiente. A principal causa desta deficiência deveu-se ao desenvolvimento segundo uma metodologia de prototipagem, utilizando técnicas de programação procedurais. Tornou-se necessário o uso, no processo de desenvolvimento, de uma metodologia de projeto que formalmente englobasse e documentasse as várias fases, dando garantia à evolução do software.

### **7.6.1 Evolução dos Módulos de Kernel Embarcado e em Terra**

Para alicerçar a geração de uma nova versão do Ambiente de Desenvolvimento e Operação, MAETA (2001) e MIRISOLA (2001) adotaram um paradigma de programação orientado a objetos. Além disso, a metodologia UML (*Unified Modeling Language*) foi utilizada como ferramenta de desenvolvimento de *software*, englobando as fases clássicas de Especificação de Requisitos, Análise de Requisitos, Projeto e Implementação; ao final do ciclo de desenvolvimento foram gerados os diagramas das classes e diagramas de colaboração (da metodologia UML) para o sistema.

Para estas versões orientadas a objeto do MKE e do MKT, os objetivos considerados foram:

- Encapsular as rotinas de baixo nível como: leitura/escrita de dados das portas seriais, formatação/conversão de dados de sensores e acesso a outros periféricos;
- Separar em módulos os diversos elementos que compõem os sistemas embarcado e de terra, organizando-os segundo critérios de funcionalidade e facilidades de compreensão e manutenção;
- Permitir que novos sensores fossem facilmente acoplados, tanto na estação embarcada como na estação de terra, sem geração de impactos significativos no restante do sistema;
- Simplificar a inserção de algoritmos de controle e navegação nas estações embarcada e de terra, com a sintonização de seus parâmetros “*on line*” a partir da estação de terra, sem impactos significativos no restante dos componentes do sistema;

- Facilitar o processo de conexão dos sistemas de controle de baixo nível embarcados com outros que possuam um nível de abstração mais elevado, como programação de tarefas ou controle de missão;
- Sistematizar e melhorar a qualidade da documentação, facilitando a compreensão do sistema, de seu funcionamento, e também a sua evolução.

Tanto para o MKE como o para o MKT estabeleceu-se analogia com um agente robótico apresentado em MAETA (2001), resultando no diagrama simplificado da nova estrutura do *software* embarcado e de terra, ilustrado na Figura 7.9 (MAETA, 2001), que apresenta na sua parte superior os componentes de *hardware* e na parte inferior os componentes de *software*, que nos seus níveis mais próximos ao *hardware* refletem seus componentes, e nos níveis mais afastados refletem os componentes de nível de abstração maior.

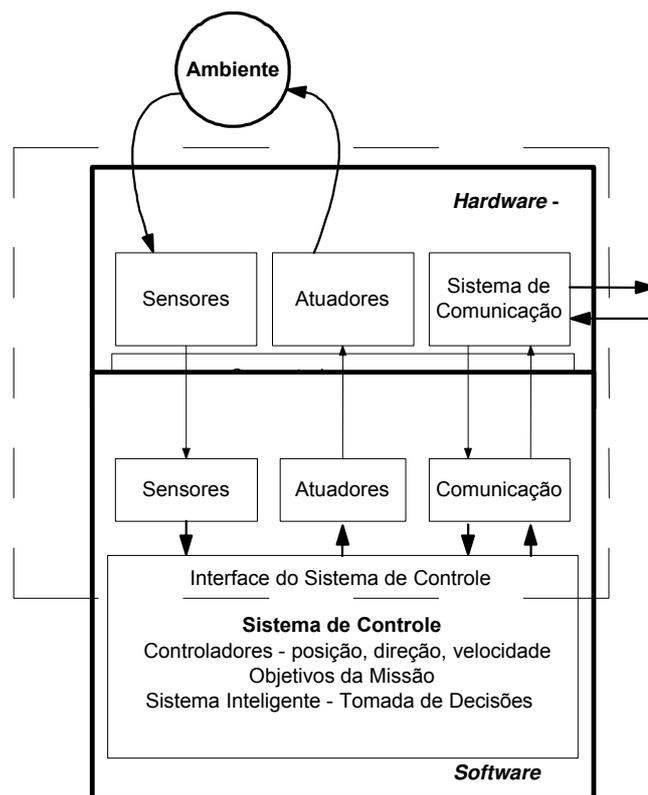


Figura 7.9: Esquema simplificado da estrutura para o software do módulo do *Kernel* Embarcado e módulo *do Kernel* em Terra.

As novas versões do MKT e do MKE compõem-se de quatro elementos principais: sensores (*sensors*), atuadores (*actuators*), sistema de controle (*control*) e sistema de

comunicação (*communication*). No caso da instância implementada pelo MKT, os elementos atuadores não são utilizados.

A Figura 7.10 apresenta o diagrama de classes aplicável tanto para o sistema embarcado (*embedded\_system*) quanto para o sistema em terra. Ele é composto pelos quatro módulos: *sensors*, *communication*, *control* e *actuators* e as suas interfaces públicas. Observa-se na figura que as interfaces públicas das classes correspondem a métodos para a inicialização e execução das tarefas destas e para a troca de mensagens com outros módulos. Observa-se, também, que as classes *sensors*, *control* e *actuators* herdam características da classe *module*, com compartilhamento de métodos para troca de mensagens com outros módulos, inclusive mensagens que podem ser trocadas entre o MKE e o MKT. Vê-se também a classe *canManager*, que é compartilhada pelos módulos *sensors* e *actuators*, pois ela pode ser utilizada para acionamento de atuadores ou para leitura de dados de sensores.

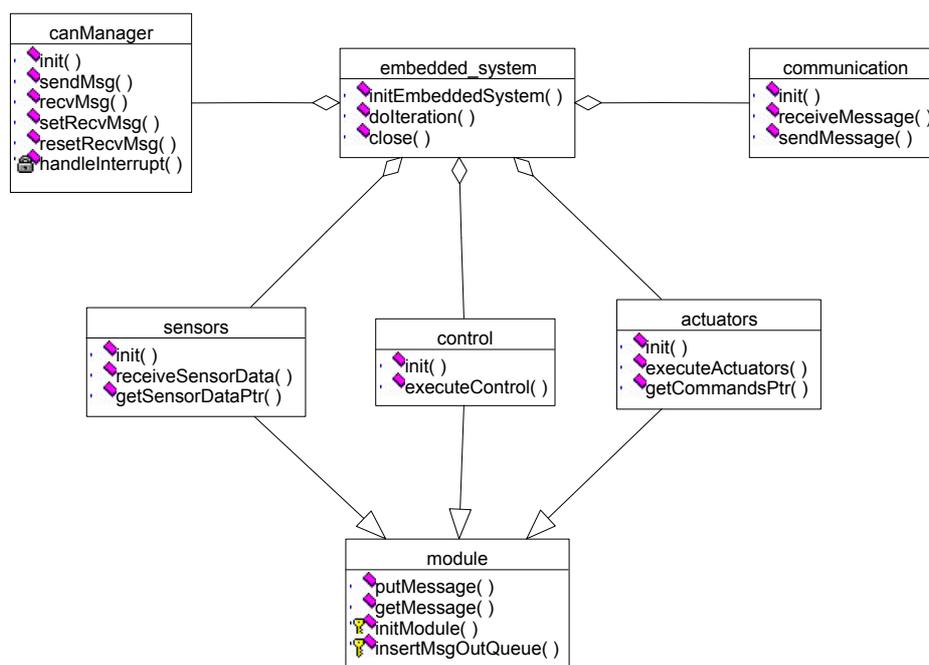


Figura 7.10: Diagrama de classes dos módulos que compõem o sistema embarcado

A Figura 7.11 apresenta um dos diagramas de colaboração elaborados para o MKE, mostrando o relacionamento dos métodos das classes no caso de inicialização do sistema (MAETA, 2001). O Diagrama de Colaboração consiste em caixas representando as classes, linhas verticais partindo das classes e, em linhas horizontais, setas que podem interceptar as linhas verticais indicando ou não a existência de uma colaboração entre as classes.

Assim o diagrama de colaboração da Figura 7.11 mostra a colaboração existente entre as classes do sistema embarcado no seu processo de inicialização: a classe "embarc\_main" do MKE chama o método de inicialização (initEmbeddedSystem) da instância da classe principal do MKE (embedded\_System). A chamada da inicialização do sistema embarcado consiste:

- i) na chamada da inicialização da rede CAN através do método "init" da classe canManager;
- ii) na inicialização dos sensores através do seu método de inicialização que recebe a referência do gerenciador da rede CAN (canManager);
- iii) na inicialização dos atuadores, no caso atual através da especificação da referência do módulo microcontrolador (MCC) descrito no Anexo 1;
- iv) na inicialização dos algoritmos de controle através do seu método "init" e;
- v) na inicialização da comunicação entre os módulos através do método "init".

Há outros diagramas de colaboração como o que descreve um passo de iteração do sistema de controle, etc.

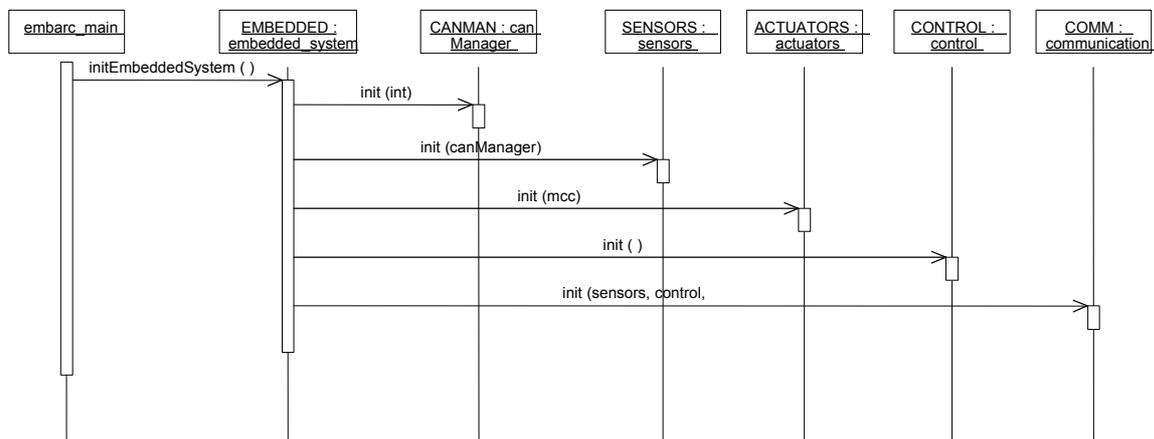


Figura 7.11: Diagrama de seqüência da inicialização do sistema embarcado.

Como já foi dito anteriormente, o MKT (MIRISOLA, 2001) utiliza o mesmo diagrama de classes que o MKE e possui praticamente a mesma estrutura que este. A sua implementação é mais simples, com a instanciação de objetos adequados à estação de terra.

### 7.6.2 Programação de Tarefas por Pontos de Passagem e Sintonização de Controladores

Em (MIRISOLA, 2001) é apresentado o desenvolvimento de um Módulo da IHM que permite ao operador da estação de terra, durante o voo, de interagir com o sistema embarcado com dupla finalidade:

i) ajustar dos parâmetros de configuração e sintonia dos algoritmos embarcados de controle e navegação do dirigível, apresentados no Capítulo 8 e ;

ii) possibilitar a programação de tarefas, estabelecidas nessa versão através da definição de um conjunto de pontos de passagem que representam locais a serem sobrevoados pelo dirigível.

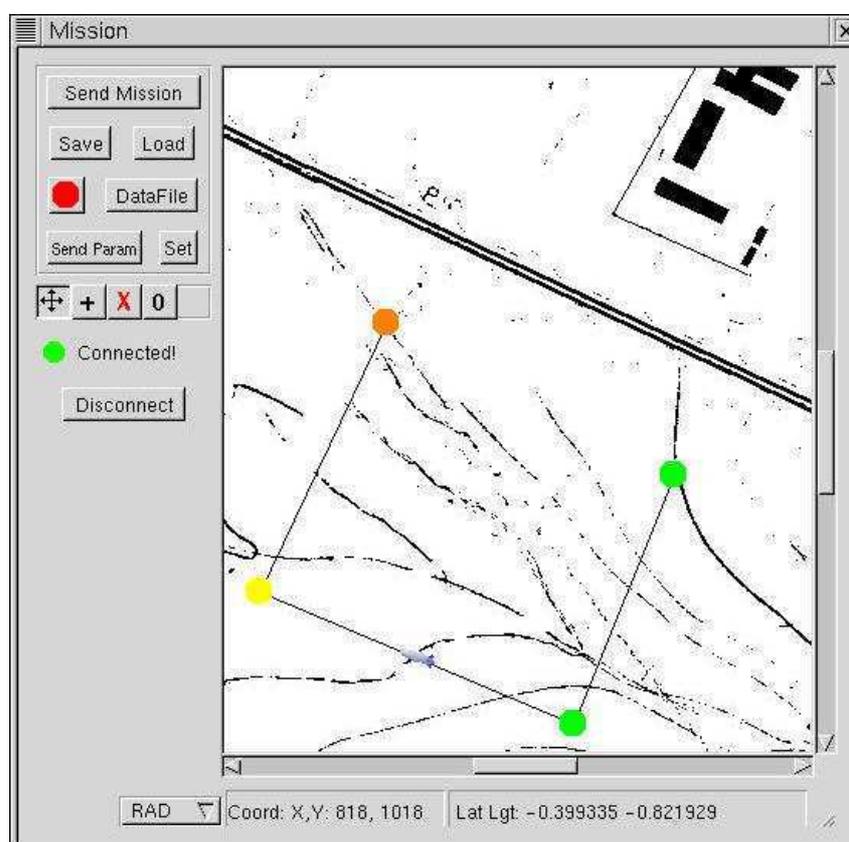


Figura 7.12: Interface gráfica do sistema de programação de tarefas.

A Figura 7.12 apresenta a tela principal do módulo, com seus vários botões de interface com o usuário. No campo à esquerda aparecem botões que são auto explicativos, na medida em que o *mouse* para sobre eles. Existem botões associados à alteração de parâmetros de configuração e de sintonia dos algoritmos de controle no caso os botões "set" e "Send Param", sendo os demais botões associados à definição de uma tarefa através de pontos de passagem como os botões "Send Mission", "Save", "Load", etc.

No caso de definição de uma tarefa por pontos de passagem, mostrado na Figura 7.12, aparece na tela o mapa digitalizado da região de vôo definido através de um arquivo de configuração. O usuário, movimentando o mouse sobre o mapa, pode definir novos pontos por onde o dirigível passará e clicando sobre o local correspondente, incluir este ponto na lista de pontos da tarefa do dirigível. Da mesma forma ele pode apagar pontos da lista, selecionando um ponto com um click do mouse e clicando no botão "X". O Botão "Save" permite ao usuário salvar uma lista de pontos, e o botão "Load" carregar uma lista de pontos. Esta lista de pontos é armazenada sob forma textual permitindo ao usuário editá-la com um editor de texto e alterá-la. Quando da execução de um vôo o usuário pode enviar uma lista de pontos ao sistema embarcado através do botão "Send Mission", sendo que na versão atual esta lista só será aceita pelo sistema embarcado caso o dirigível esteja em modo de pilotagem manual. Imediatamente após o chaveamento de modo manual para automático no rádio-controle do dirigível, este começará a fazer a varredura seqüencial da lista enviada.

Conforme já foi dito, os botões "Set" e "Send Param" estão associados à alteração parâmetros de configuração e de sintonia dos algoritmos de controle. O click no botão "Set" abre a janela de configuração mostrada na Figura 7.13, chamada de "Parameters".

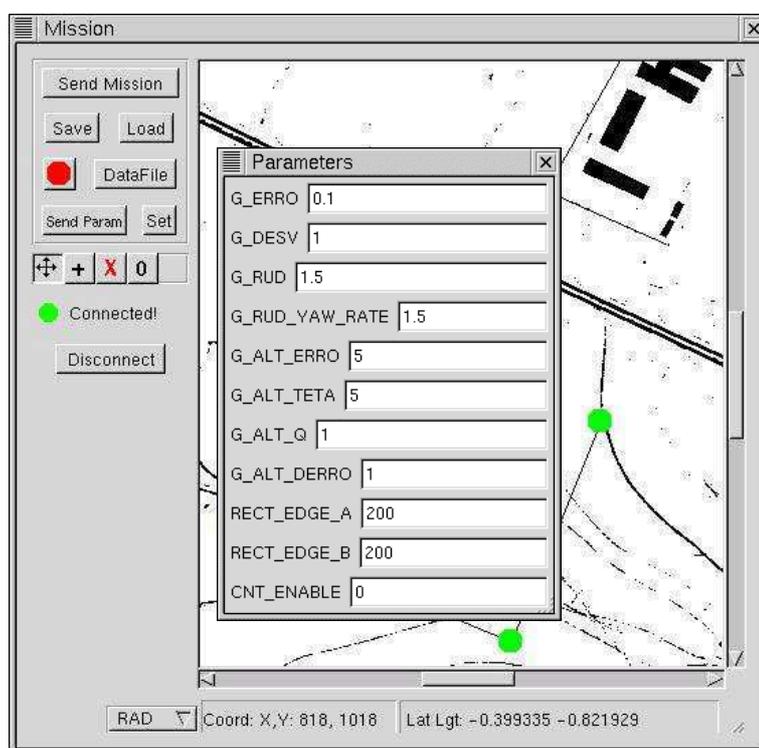


Figura 7.13: Interface para ajuste de ganhos e definição de missão.

O nome e número de parâmetros é definido num arquivo de configuração, aparecendo na janela o número de linhas correspondente ao número de parâmetros. O programa em si não realiza nenhuma crítica nos parâmetros digitados, sendo a estação embarcada a que tem entendimento sobre sua semântica. No caso mostrado na figura são definidas onze parâmetros, dos quais os oito primeiros correspondem a ganhos dos controladores de trajetória e altitude, as linhas chamadas de "RECT\_EDGE\_A" e "RECT\_EDGE\_B" servem para definir duas arestas de um retângulo, caso se use geração automática de pontos de passagem na estação embarcada e a última linha "CNT\_ENABLE" é para definir quais são os controladores que serão habilitados quando do chaveamento para o modo automático, atualmente "0" significa só trajetória, "1" trajetória e altitude, "2" trajetória, altitude e velocidade e "3" só altitude.

O botão "Send Param" da janela principal serve para se enviar via rádio-modem os parâmetros para a estação embarcada, que só os aceita, caso se encontre em modo de pilotagem manual. Observa-se que o piloto deve utilizar o rádio-controle para comandar os atuadores que não foram habilitados para o modo de controle automático, assim no modo "0" ele deve controlar a altitude pelo leme de profundidade e ou vetorização, e a velocidade de rotação dos motores.

Este módulo, por ser compilado, apresenta vantagens de desempenho em relação a aplicativos escritos em Java ou Tcl/Tk, que são interpretados. A base para o seu desenvolvimento em C++ é a biblioteca gráfica orientada a objetos, o "qt", que possui estrutura de dados para facilitar aplicações que usam interfaces gráficas e a classe "*template*" que implementa as estruturas de dados comuns (fila, listas, etc).

Por ter sido desenvolvido segundo uma metodologia orientada a objetos, o módulo pode ser facilmente modificado para atender aos novos tipos de tarefas e novos algoritmos de controle e navegação. Em MIRISOLA, (2001) são explicados detalhes do módulo, como a independência da representação gráfica do objeto e das suas características gerais, ou a forma como seriam introduzidas primitivas adicionais como vôo pairado, decolagem e aterrissagem.

### 7.6.3 Visualizador 3D de vôos

Outra evolução incorporada foi a de visualização 3D do movimento do dirigível, mostrada na Figura 7.14, realizada por um programa cliente escrito em “C”, visu3D, que se conecta ao servidor de dados da estação de terra. Este programa utiliza funções da biblioteca MESA (MESA, 2001), com funcionalidade análoga ao Open-GL, e com base nas informações dos estados, posição e orientação do dirigível, realiza a visualização 3D do veículo. Diferentes pontos de vista podem ser definidos pelo usuário; na Figura 7.14 vê-se o dirigível a partir do solo, enquanto que uma outra possibilidade é a visão do solo a partir de uma câmera virtual embarcada no dirigível.

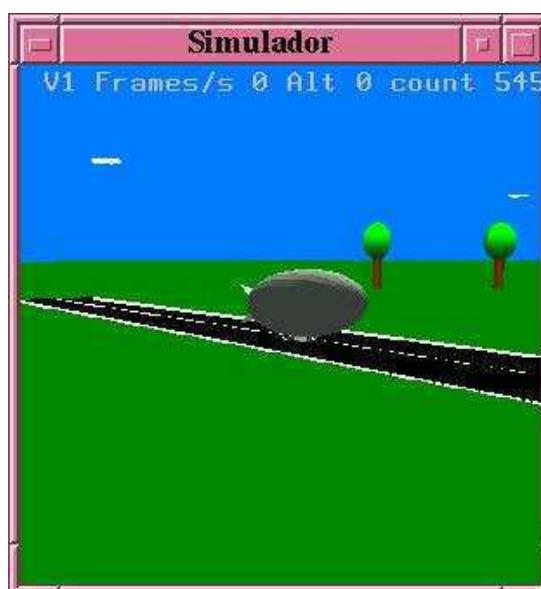


Figura 7.14: Visualização 3D dos estados do dirigível.

Esta solução de visualizador em Mesa/OpenGL para o dirigível foi desenvolvida pela equipe do Dr. José Raul Azinheira do IST, Portugal. Ela foi adotada em substituição à visualização JAVA/VRML apresentada no Capítulo 5, pois esta última necessita de um *plug-in* VRML para Linux, que até muito recentemente não estava disponível.

## 7.7 Conclusões

Este capítulo apresentou inicialmente a motivação para a concepção de um ambiente de suporte ao desenvolvimento e operação do dirigível, tendo como referência outro ambiente desenvolvido para uso com veículos subaquáticos. Abordou-se então a configuração conceitual do ambiente com todos os componentes integrados por uma rede TCP/IP. A seguir foi apresentada a estrutura efetivamente adotada em termos de

modularização e comunicação entre os módulos. Foi detalhado o protótipo inicial, desenvolvido congregando as estações de terra e embarcada, além do ambiente de CACSD do dirigível; o ambiente assim configurado pode ser utilizado no modo simulado (dando suporte ao desenvolvimento) ou no modo real (apoiando a operação do dirigível robótico em seus vôos). Na seção final apresentaram-se brevemente resultados de duas dissertações de mestrado, associadas a esta tese, desenvolvidas na perspectiva da evolução do ambiente, utilizando técnicas de projeto de software e programação orientada a objeto, e da inclusão de um módulo de IHM para a programação de tarefas para o dirigível, em termos de pontos de passagem, para a sintonização dos controladores embarcados e para a incorporação de um visualizador em 3D da situação do dirigível em vôo

O ambiente de desenvolvimento e operação constitui a base para os trabalhos do dirigível robótico, dando suporte à telemetria e armazenamento de dados, à alteração de parâmetros e configuração de algoritmos de controle, à programação de seqüências de vôo, permitindo uma série de testes em vôo real e simulado. Além do mais ele constitui a base para o desenvolvimento da arquitetura de *software* robótico do dirigível tratado no Capítulo 9.

Os programas que compõem este ambiente totalizam aproximadamente vinte mil linhas de código escrito em "C", "C++", Java, Tcl/Tk e assembler. Alguns de seus componentes possuem solução específica para uso no sistema operacional Linux, mais particularmente o RTLinux, sendo que a migração para outros sistemas de tempo real poderá requerer a realização de um trabalho de adaptação.

O próximo capítulo enfoca o sistema de controle e navegação, cuja implementação utilizou o Ambiente de Desenvolvimento e Operação.

## **8. IMPLEMENTAÇÃO DE ALGORITMOS DE CONTROLE**

### **8.1 Introdução**

O Capítulo 4 apresentou um panorama e detalhou conceitualmente os algoritmos que dão capacidade de controle de movimentos e de navegação ao dirigível, obedecendo a requisitos de estabilizá-lo em seu vôo e, ao mesmo tempo, de executar trajetórias retilíneas com determinado perfil de altitude, entre pontos de passagem definidos em termos das coordenadas de Latitude e Longitude.

No processo de desenvolvimento do sistema de controle e navegação, foram utilizados os simuladores e ferramentas de CACSD, descritas no Capítulo 6, e o Ambiente de Desenvolvimento e Operação que os incorpora, apresentado no Capítulo 7. Como precisado anteriormente, o modelo matemático e os simuladores resultantes são qualitativamente representativos da dinâmica do veículo, permitindo o desenvolvimento e análise de estratégias de controle e navegação sob as mais diversas condições, como diferentes níveis de fluabilidade ou intensidade e orientação do vento como perturbação, bem como o estudo da influência de parâmetros de sintonia dos algoritmos de controle e navegação no desempenho e robustez do sistema. Por outro lado, explicitou-se também que no presente momento o modelo ainda é quantitativamente impreciso, de forma que não se pode transferir diretamente os parâmetros de sintonia dos algoritmos de controle e navegação determinados em simulação para os utilizados na implementação real. Assim, nesta tese, adotou-se como metodologia a realização de um extenso conjunto de simulações para estudar a influência dos parâmetros de sintonia, das condições de operação e das perturbações no comportamento do sistema de controle e navegação. As informações assim coletadas constituem o subsídio para a implementação do sistema de controle e navegação na infra-estrutura embarcada do dirigível e para o ajuste dos parâmetros de sintonia em situações reais de vôo.

Considerando os aspectos acima descritos, neste capítulo, a Seção 8.2 apresenta o diagrama global do sistema de controle de trajetória e altitude concebido para o dirigível, baseado no que foi apresentado no Capítulo 4, os parâmetros de sintonia e alguns detalhes sobre a implementação realizada. A Seção 8.3 apresenta um estudo em simulação do algoritmo de controle de trajetória para diferentes condições de operação, mostrando a

influência do ajuste dos parâmetros de sintonia e de condições ambientais, como o vento, no comportamento do algoritmo. Da mesma forma, a Seção 8.4 apresenta o estudo para o controle de altitude isoladamente e conjuntamente com o controle de trajetória, mostrando a influência deste no comportamento do controle de altitude. A Seção 8.5 apresenta resultados de ensaios em vôo do algoritmo de controle de trajetória e a Seção 8.6 apresenta resultados de vôo com controle de trajetória e controle de altitude simultaneamente. A Seção 8.7 apresenta as conclusões do capítulo.

## 8.2 Diagrama Geral do Sistema de Controle Implementado

A Figura 8.1 mostra o diagrama geral do sistema de controle e navegação implantado na infra-estrutura embarcada do dirigível, apresentando uma visão conjunta dos controladores de altitude e de trajetória. Distinguem-se quatro blocos principais: o dirigível com seus sensores e atuadores, o sistema de controle de trajetória, o sistema de controle de altitude, e o sistema de conversão dos comandos de “+” para “x” para a deflexão das superfícies aerodinâmicas.

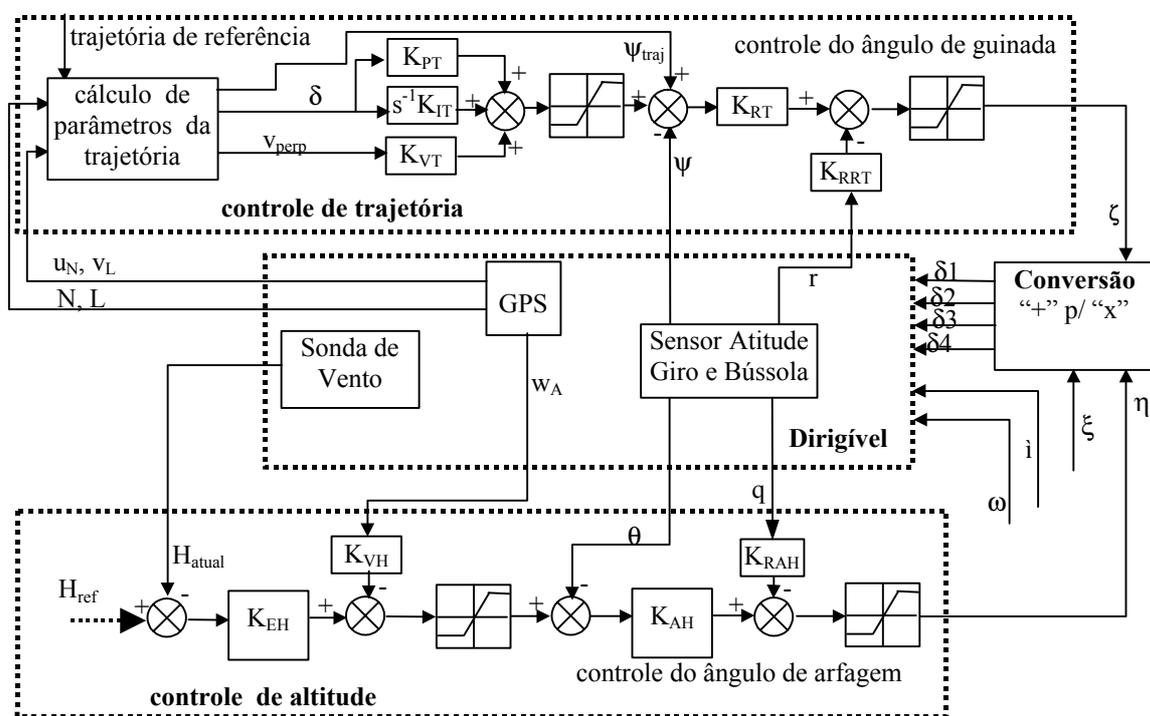


Figura 8.1: Diagrama geral do sistema de controle de trajetória e altitude.

No bloco dirigível são destacados os blocos de sensores: a sonda de vento, o GPS, e os dois sensores de atitude – o sensor TCM e a central inercial apresentados no Anexo 1 –.

A sonda de vento mede a altitude barométrica “ $H_{\text{atual}}$ ”, o ângulo de ataque “ $\alpha$ ”, o ângulo de incidência lateral “ $\beta$ ” e a velocidade total do veículo relativa à velocidade do ar “ $v_{\text{ar\_tot}}$ ”. O GPS mede a latitude e a longitude, que são convertidas para distâncias a partir do equador na direção Norte “N” e a partir do meridiano de Greenwich na direção Leste “L”; ele fornece também a altitude relativa ao nível do mar e as velocidades do veículo na direção norte “ $u_N$ ”, leste “ $v_L$ ” e vertical para cima “ $w_A$ ”. O TCM e a Central Inercial medem a atitude sendo que a última fornece, ainda, informações sobre as velocidades angulares (p, q e r) e as acelerações lineares, nos eixos X, Y e Z do dirigível respectivamente. A atitude medida é representada pelos ângulos de rolagem “ $\phi$ ”, de arfagem “ $\theta$ ”, e de guinada “ $\psi$ ”, sendo os dois primeiros medidos por níveis de bolha no TCM e pela combinação de giroscópio e acelerômetros na central inercial e o de guinada resultante da combinação de medidas de magnetômetros de três eixos e do nível de bolha no TCM, gerando como saída a direção do norte magnético medido pelo sensor. A direção do norte magnético é medida na Central Inercial pela combinação de três magnetômetros, acelerômetro e giroscópios. As velocidades angulares do dirigível são medidas por giroscópios na central inercial.

Das variáveis associadas ao controle de trajetória, são utilizadas as medidas do GPS: a latitude, a longitude e as velocidades do veículo na direção norte “ $u_N$ ” e leste “ $v_L$ ”; do TCM o ângulo de guinada “ $\psi$ ” e da central inercial a velocidade de guinada “r”. Para as variáveis associadas ao controle de altitude, são utilizadas a altitude barométrica “ $H_{\text{atual}}$ ” da sonda de vento; a velocidade vertical “ $w_A$ ” do GPS; o ângulo de arfagem “ $\theta$ ” e a velocidade de arfagem “q” da central inercial.

As variáveis manipuladas são os ângulos dos quatro lemes das superfícies aerodinâmicas montadas em “x” ( $\delta_1$ ,  $\delta_2$ ,  $\delta_3$  e  $\delta_4$ ). Os sinais de comando “ $\eta$ ” do leme de profundidade, ou profundor, utilizado para controle de altitude, do leme de direção “ $\zeta$ ”, usado para controle de trajetória, e de rolagem “ $\xi$ ”, associados a uma configuração em “+”, são combinados para gerar os comandos em “x” correspondentes, conforme as Equações 3.1 a 3.4 da Seção 3.3.2. Como já se mencionou naquela seção, já que no AS800 são utilizados os mesmos sinais para comandar os lemes  $\delta_1$ - $\delta_3$  e  $\delta_2$ - $\delta_4$ , ou seja  $\delta_1=\delta_3$ , e  $\delta_2=\delta_4$ , isto resulta em “ $\xi$ ” nulo. Na presente implementação são fixados pelo operador o ângulo de vetorização “ $\mu$ ” e a rotação dos motores “n”, ditadas pela abertura dos carburadores dos motores.

Os parâmetros de sintonia do sistema de controle são reagrupados na Tabela 8.1. A determinação destes parâmetros, em simulação e em situações reais de vôo, são tratados no restante deste capítulo.

Ganho	Descrição
$K_{PT}$	ganho proporcional da malha externa de controle de trajetória;
$K_{IT}$	ganho integral da malha externa de controle de trajetória;
$K_{VT}$	ganho da taxa de variação de erro de trajetória, na malha externa de controle de trajetória;
$K_{RT}$	ganho proporcional de realimentação do ângulo de guinada da malha interna de controle de trajetória
$K_{RRT}$	ganho proporcional de realimentação da velocidade de guinada da malha interna de controle de trajetória
$K_{EH}$	ganho proporcional de realimentação do erro de altitude da malha externa de controle de altitude
$K_{VH}$	ganho proporcional de realimentação da velocidade vertical da malha externa de controle de altitude
$K_{AH}$	ganho proporcional de realimentação do ângulo de arfagem da malha interna de controle de altitude
$K_{RAH}$	ganho proporcional de realimentação da velocidade de arfagem da malha interna de controle de altitude

Tabela 8.1: Parâmetros de sintonia do sistema de controle e navegação

A implementação dos controladores é realizada da seguinte forma:

- O controlador da malha externa opera no intervalo de amostragem do GPS, que em condições normais é de 1 Hz, enquanto que a malha interna opera à frequência de 10 Hz;
- Na malha externa de controle de trajetória o termo integral do controlador implementado utiliza um método de integração de primeira ordem, com estratégia *anti wind-up* para se evitar a influência da saturação do integrador;
- A malha interna do controlador de ângulo de guinada possui um procedimento que ignora os comandos do controlador de trajetória quando o erro de guinada ultrapassa um determinado limite. Esta solução se faz necessária quando ocorrem transições abruptas de trajetória, evitando que o dirigível assuma direções errôneas.

### 8.3 Controle de Trajetória: Simulação

Apresenta-se a seguir estudos de simulação para o sistema de controle de trajetória do dirigível, usando o CACSD desenvolvido em MATLAB e o Ambiente de Desenvolvimento e Operação descritos nos Capítulos 6 e 7. Os algoritmos de controle já têm suas rotinas implementadas em código "C" compilado, para facilitar a posterior migração do sistema de controle e navegação para o código embarcado. São consideradas, nas simulações, diferentes condições de operação, perturbações, e conjuntos de valores para os parâmetros de sintonia dos controladores.

#### 8.3.1 Trajetória de Referência

A trajetória escolhida para testes dos algoritmos de controle e navegação, mostrada na Figura 8.2, consiste num quadrilátero, com arestas de extensão programáveis (usualmente 200m nos estudos realizados), de forma similar à que se utiliza quando da realização dos testes práticos. O quadrilátero se caracteriza pelo primeiro ponto de passagem (PI-1), primeiro ponto final (PF-1), e pontos subsequentes (PI-2, PF-2, PI-3, PF-3, PI-4 e PF-4); as arestas (PI-1, PF-1), (PI-2, PF-2), (PI-3, PF-3) e (PI-4 e PF-4) constituem os segmentos de reta que formam a trajetória de referência, utilizada pelo sistema de controle de trajetória. O dirigível decola da origem e na direção norte, depois desloca-se para leste, posteriormente para o sul e então para o oeste.

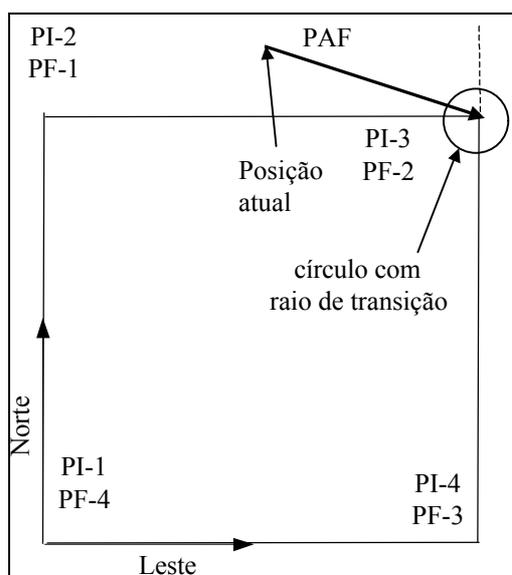


Figura 8.2: Retas de referência e critério para transição entre retas.

A mudança entre as os segmentos de reta - as trajetórias de referência - no quadrilátero, o que é extensível a outras configurações, é feita segundo dois critérios. O

primeiro usa a distância da posição atual do dirigível ao ponto final da trajetória atual, dada pelo módulo do vetor PAF mostrado na Figura 8.2, que deve ser menor que um raio, chamado de raio de transição, em torno do ponto final. O segundo critério usa a projeção do vetor PAF na trajetória de referência. Se esta for negativa (i.e. além do ponto final da trajetória), significa que o veículo já ultrapassou o prolongamento do segmento de reta seguinte; caso o valor seja positivo, o dirigível ainda não alcançou este prolongamento. Verificando-se a satisfação de um desses dois critérios, é definida uma nova trajetória de referência para o sistema de controle de trajetória, através do novo par de pontos – inicial e final – e da aresta correspondente.

### ***8.3.2 Influência dos Parâmetros de Sintonia dos Controladores***

O procedimento para sintonia dos controladores da malha externa de trajetória e da malha interna de ângulo de guinada, em simulação, seguiu o procedimento clássico para controladores PID. Inicialmente se sintoniza os ganhos da malha interna de guinada, posteriormente os ganhos das ações proporcional e derivativa (PD) e finalmente o ganho da ação integral (I) da malha externa de trajetória

A seqüência de ajustes consistiu na obtenção de um conjunto de ganhos PD que reduzisse o erro de trajetória. Inicialmente buscou-se um conjunto de ganhos sem vento, correspondente à situação ideal; depois foi realizada a avaliação destes ganhos em ventos de fracos a moderados, ou seja, ventos da ordem de 25% e 40% da velocidade de vôo, especificada em 9 m/s.

#### ***Sintonia dos Parâmetros da Malha Interna de Controle de Guinada***

Esta seção mostra, através de exemplos, a influência dos parâmetros da malha interna de controle de guinada ( $K_{RT}$  e  $K_{RRT}$ ), sem o controle de trajetória. A Figura 8.3 apresenta o efeito de diferentes valores dos ganhos  $K_{RT}$  e  $K_{RRT}$  no comportamento do controlador de ângulo de guinada. As referências de ângulo de guinada são sucessivamente  $0^\circ$ ,  $90^\circ$  e  $180^\circ$ , com essas referências sendo mantidas até que veículo cruzasse a reta com coordenada Norte igual a 200m, depois Leste de 200m e então coordenada Norte nula. O critério de ajuste foi de primeiro se incrementar o ganho da malha de ângulo de guinada ( $K_{RT}$ ) para posteriormente, se necessário, variar o ganho da malha de velocidade de guinada ( $K_{RRT}$ ).

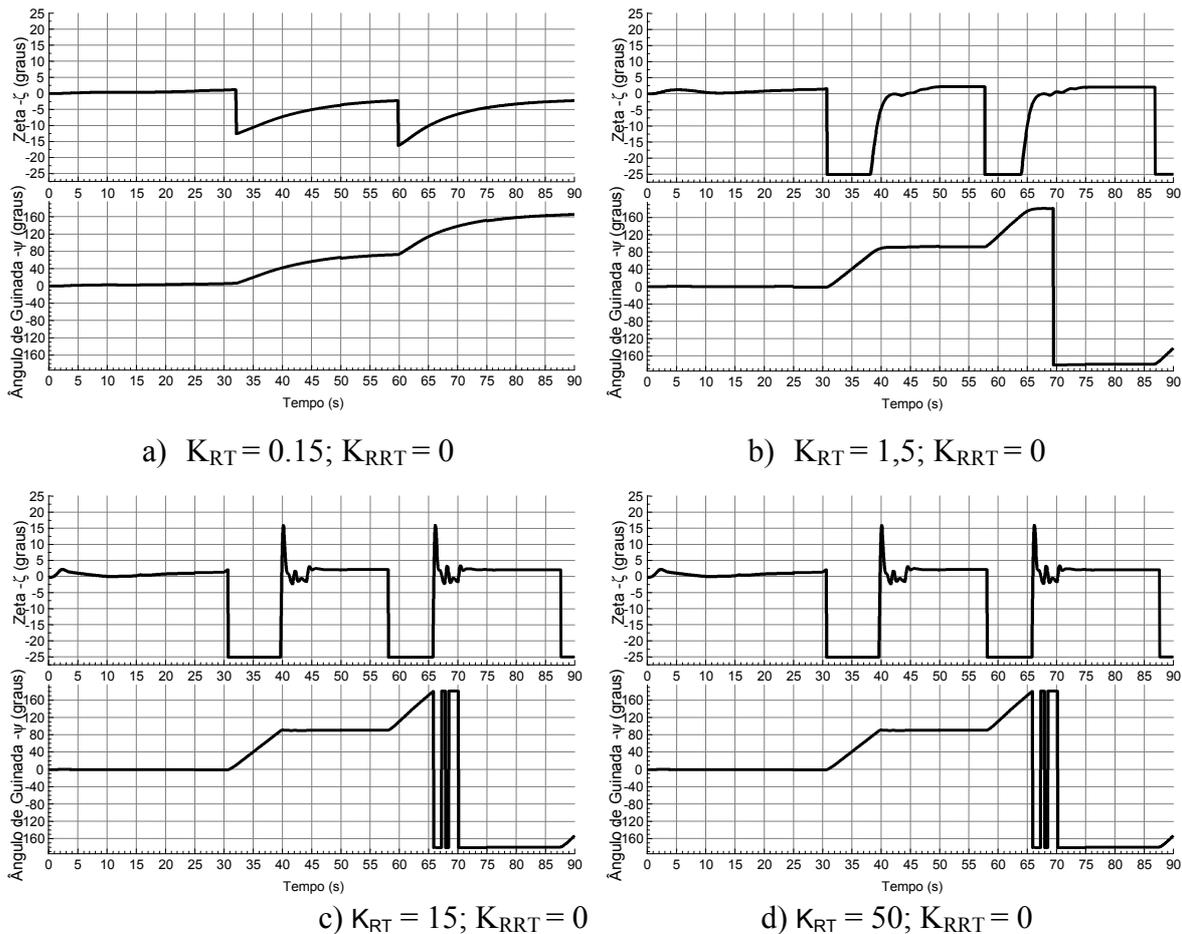


Figura 8.3: Efeito da variação dos ganhos  $K_{RT}$  e  $K_{RRT}$ .

Dentre as várias simulações realizadas, onde alguns dos resultados foram apresentados na Figura 8.3, observa-se:

- O conjunto "a" está super-amortecido, não conseguindo acompanhar as referências;
- Os conjuntos "c" e "d" já começam a apresentar alguma oscilação no sinal de controle "ζ", apesar de não ser observado visualmente este efeito no ângulo de guinada "ψ".
- O conjunto "b" consegue acompanhar as referências, com um pequeno erro de regime permanente, sem oscilação no sinal de controle "ζ";
- A utilização do ganho  $K_{RRT}$  não trouxe benefício perceptível, tendo sido adotado o valor nulo para este parâmetro.

Assim, para a malha interna de controle de guinada adotou-se  $K_{RT} = 1,5$  e  $K_{RRT} = 0$ , correspondendo ao conjunto "b".

### *Sintonia dos Parâmetros PV do Controlador de trajetória sem a Presença de Vento.*

O objetivo desta seção é mostrar a influência dos parâmetros  $K_{PT}$  e  $K_{VT}$  no comportamento da malha externa do controlador de trajetória, sem a presença de vento, quando de mudanças abruptas de direção. Mantendo-se a malha de controle de guinada com os ganhos escolhidos na seção anterior ( $K_{RT} = 1,5$   $K_{RRT} = 0$ ), foram realizadas diversas simulações com diferentes valores para os parâmetros  $K_{PT}$  e  $K_{VT}$ .

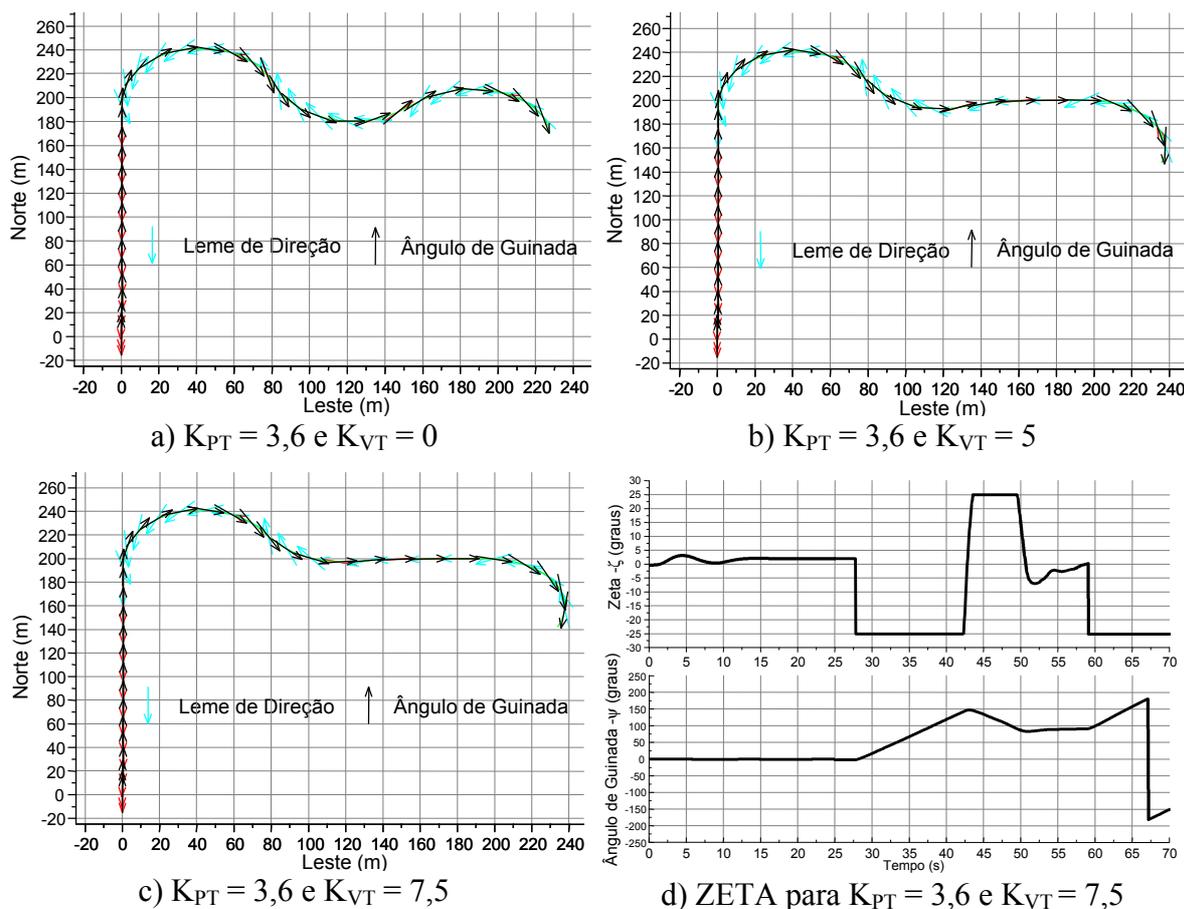


Figura 8.4: Obtenção de um conjunto de ganhos P e V.

Apresenta-se na Figura 8.4 a melhor seqüência de resultados para o ajuste dos ganhos PV. As figuras mostram as trajetórias executadas no plano NL, sendo que nelas são apresentados vetores representando o ângulo de guinada do veículo (em tom mais escuro) e a posição do leme de direção em relação a este (em tom mais claro). A seqüência ilustra o efeito de variação do ganho derivativo  $K_{VT}$  no comportamento do controlador de trajetória para um ganho proporcional ajustado anteriormente ( $K_{PT} = 3,6$ ). Procurou-se um valor do ganho derivativo que fornecesse amortecimento crítico na transição de uma reta para outra,

com uma mudança de direção abrupta de  $90^\circ$ . O mais adequado foi o par  $K_{PT} = 3.6$  e  $K_{VT} = 7,5$ , cujo comportamento é mostrado na Figura 8.4-c, enquanto que na Figura 8.4-d tem-se o comportamento do ângulo de guinada e o comando do leme de direção " $\zeta$ " correspondentes.

Observa-se nesta figura uma fuga de até 40 metros da trajetória (200 m Norte), causada pela mudança abrupta de direção. Na figura "d" observa-se o ângulo de guinada " $\psi$ " e o comando do leme de direção " $\zeta$ ", sendo que é visível a condição de saturação do leme de direção na transição de trajetória.

### ***Sintonia do Ganho Integral***

Após o ajuste dos ganhos P e V aborda-se a questão da influência do ganho integral no comportamento do sistema em malha fechada. O critério utilizado para verificação foi através de uma condição de vento com intensidade fixa de 2,5m/s e direção Norte, correspondendo a 25% da velocidade de vôo do dirigível. O vento - principal perturbação do vôo do dirigível, produz o desvio na trajetória a ser compensado pelo termo integral, o que é pouco perceptível em simulações sem vento.

A Figura 8.5 mostra a seqüência de ajustes do ganho integral  $K_{IT}$  face à presença de vento, onde os ganhos proporcional e derivativo são  $K_{PT} = 3,6$  e  $K_{VT} = 7,5$  e os ganhos do controlador de ângulo de guinada são  $K_{RT} = 1,5$  e  $K_{RRT} = 0$ . Na figura aparecem a direção do vento "V" e a direção de referência para o Norte "N", sendo apresentados a trajetória e vetores representando o ângulo de guinada do veículo (em tom mais escuro) e a posição do leme de direção em relação a este (em tom mais claro).

Na Figura 8.5-a ( $K_{IT} = 0$ ) há um erro de regime permanente na faixa de 5 m no segundo trecho (direção Leste); este erro foi reduzido com o uso de um ganho integral  $K_{IT} = 0,1$ , conforme mostrado na Figura 8.5-b, diminuindo um pouco mais com  $K_{IT} = 0,5$ , conforme mostrado na Figura 8.5-c. No entanto, na Figura 8.5-c há uma maior oscilação nas variáveis, sendo que o aumento do ganho integral para  $K_{IT} = 2$  leva o sistema a tornar-se instável, como observado na Figura 8.5-d. Adotou-se o valor  $K_{IT}$  de 0,1.

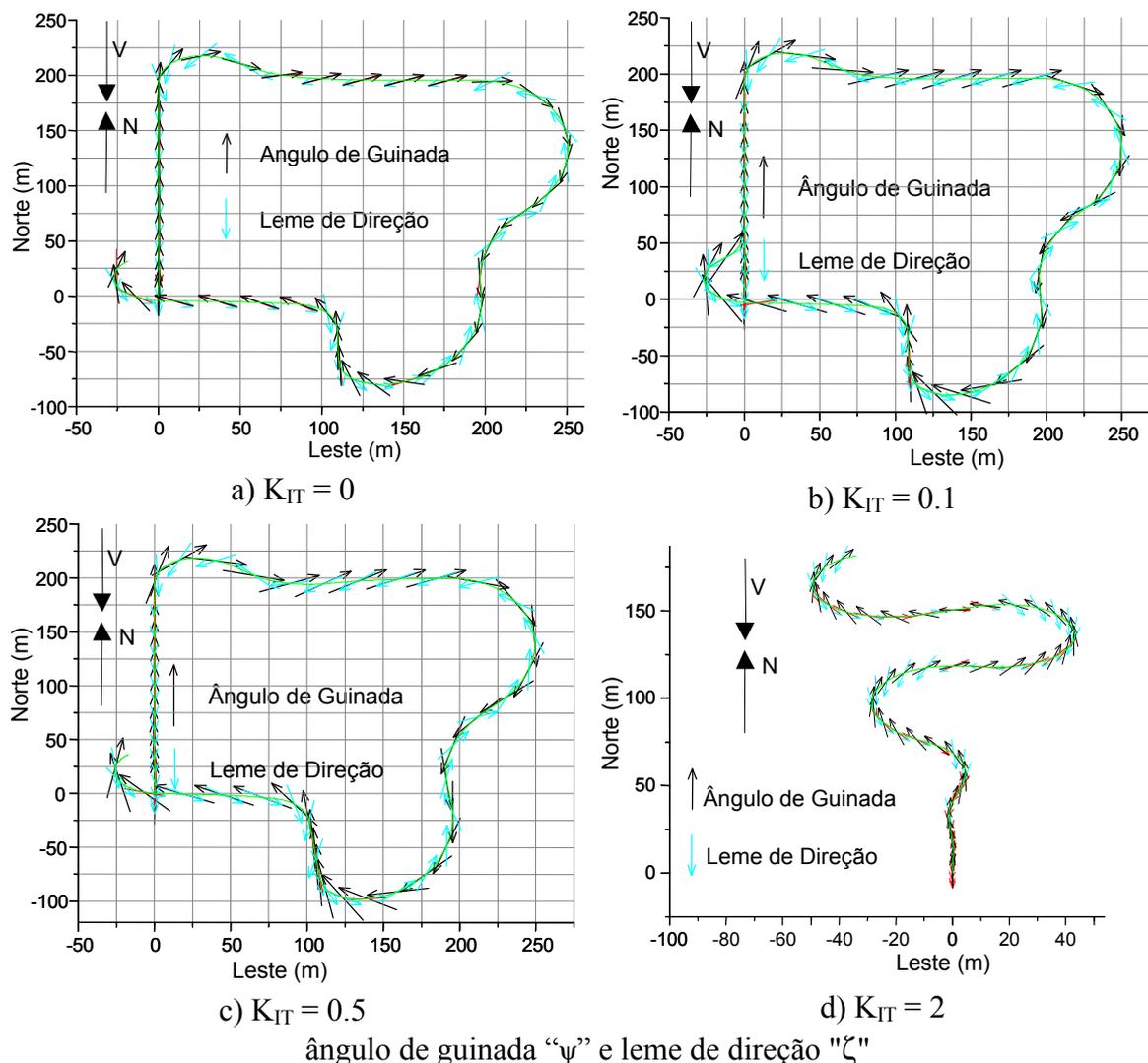


Figura 8.5: Ajuste do ganho I com vento de 2,5 m/s

### ***Influência dos Parâmetros de Sintonia do Controlador da Malha Interna de Guinada***

Na página 140 haviam sido estabelecidos  $K_{RT} = 1,5$  e  $K_{RRT} = 0$  para o controlador da malha interna de guinada. O objetivo dessa seção é mostrar o efeito da variação do parâmetro de sintonia  $K_{RT}$  no desempenho do controlador de trajetória cujos ganhos são os estabelecidos nas seções anteriores, ou seja os ganhos proporcional ( $K_{PT} = 3,6$ ), integral ( $K_{IT} = 0,1$ ) e derivativo ( $K_{VT} = 7,5$ ). Assim variou-se  $K_{RT}$  entre 0,15 e 15. Os resultados são mostrados na Figura 8.6 que, além de apresentar a trajetória seguida, apresenta também o ângulo de guinada (em tom mais escuro) e a posição do leme de direção relativa à posição do veículo (em tom mais claro), sendo que na Figura 8.6-d são apresentados o comando do leme de direção para o ganho  $K_{RT} = 1,5$  em tom mais escuro e  $K_{RT} = 15$  em tom mais claro.

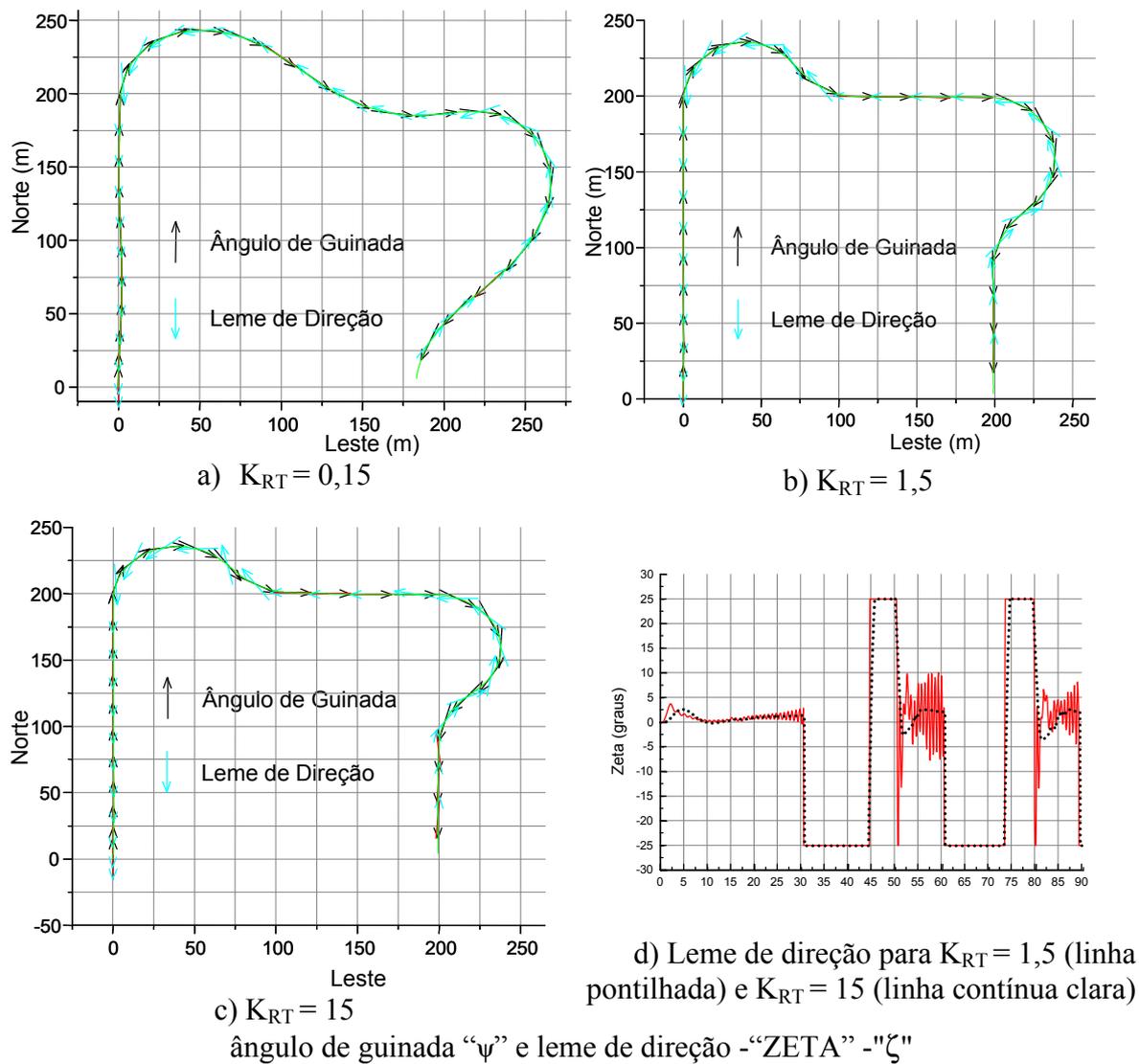


Figura 8.6: Efeito da variação do ganho  $K_{RT}$  no comportamento em trajetória.

Conforme pode-se observar na Figura 8.6-a, a utilização de um ganho  $K_{RT}$  de baixo valor não chega a degradar o comportamento em trajetória, se comparado com o desempenho apresentado na página 140. Mesmo com o incremento do parâmetro  $K_{RT}$ , conforme pode ser observado nas figuras b e c, quase não se diferencia o efeito desses na trajetória efetuada. A figura d reproduz o comportamento do comando do leme de direção, mostrando que o ganho  $K_{RT} = 15$  (em linha contínua com tom claro) apresenta oscilações, enquanto que o ganho  $K_{RT} = 1,5$  (em linha pontilhada com tom escuro) não as apresenta. O resultado apresentado confirma a escolha de  $K_{RT} = 1,5$  feita na página 140.

### *Influência de Vento Constante na Execução de Trajetória Completa*

O vento constitui a principal causa de erros na execução das trajetórias. A primeira razão é que, ele afeta o comportamento do veículo diferentemente, dependendo da sua direção relativa ao veículo, como vento de popa, proa ou lateral. Com o vento de proa, a velocidade do veículo relativa ao ar será maior, fazendo com que as superfícies de atuação aerodinâmicas gerem momentos maiores; por outro lado, quando o dirigível recebe vento de popa, a velocidade relativa ao ar é menor e as superfícies são menos eficientes. Um vento lateral provoca o deslocamento do veículo da trajetória, aumentando o erro de regime permanente da trajetória.

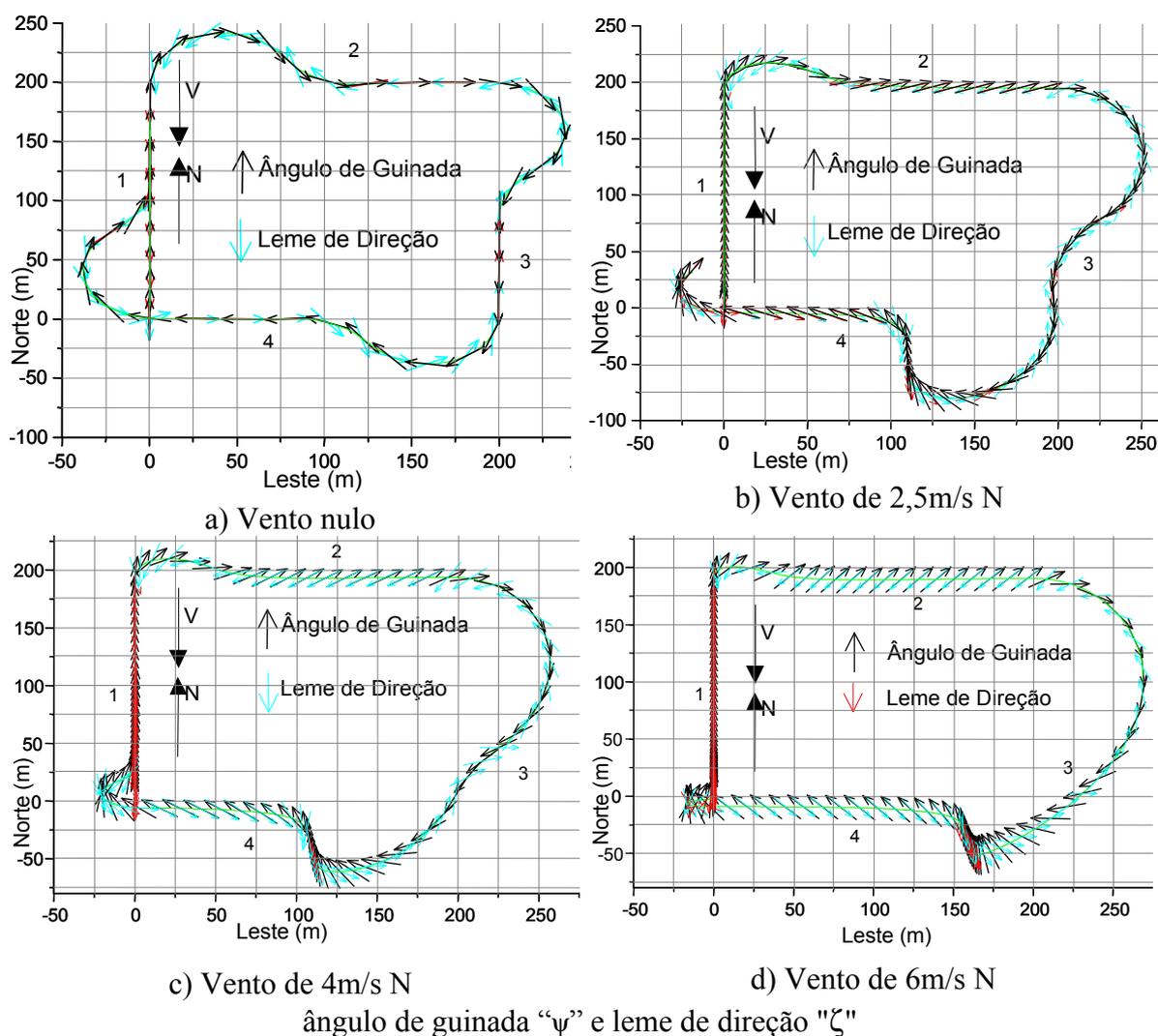


Figura 8.7: Trajetória completa com variação de vento.

Assim, após ter sido efetuado o ajuste básico para os parâmetros de sintonia: ganhos proporcional ( $K_{PT} = 3,6$ ), integral ( $K_{IT} = 0,1$ ) e derivativo ( $K_{VT} = 7,5$ ), além de ganho do ângulo de guinada ( $K_{RT} = 1,5$ ) e da velocidade de guinada ( $K_{RRT} = 0$ ), como

ensaio final verificou-se o comportamento do controlador de trajetória frente a diferentes condições de vento constante. As diferentes condições de vento são retratadas na Figura 8.7, abrangendo desde vento nulo até vento de 6m/s, ou seja, um valor correspondente a 66% da velocidade máxima alcançada pelo dirigível em simulação; o vento tem direção norte. Além disso, nesta figura, são apresentados a direção do vento "V", a direção de referência para o Norte "N" e vetores representando o ângulo de guinada do veículo (em tom mais escuro) e a posição do leme de direção em relação a este (em tom mais claro).

A Figura 8.8 apresenta o erro de trajetória para as diferentes condições de vento e trechos de trajetória numerados de "1" a "4" para as trajetórias mostradas na Figura 8.7.

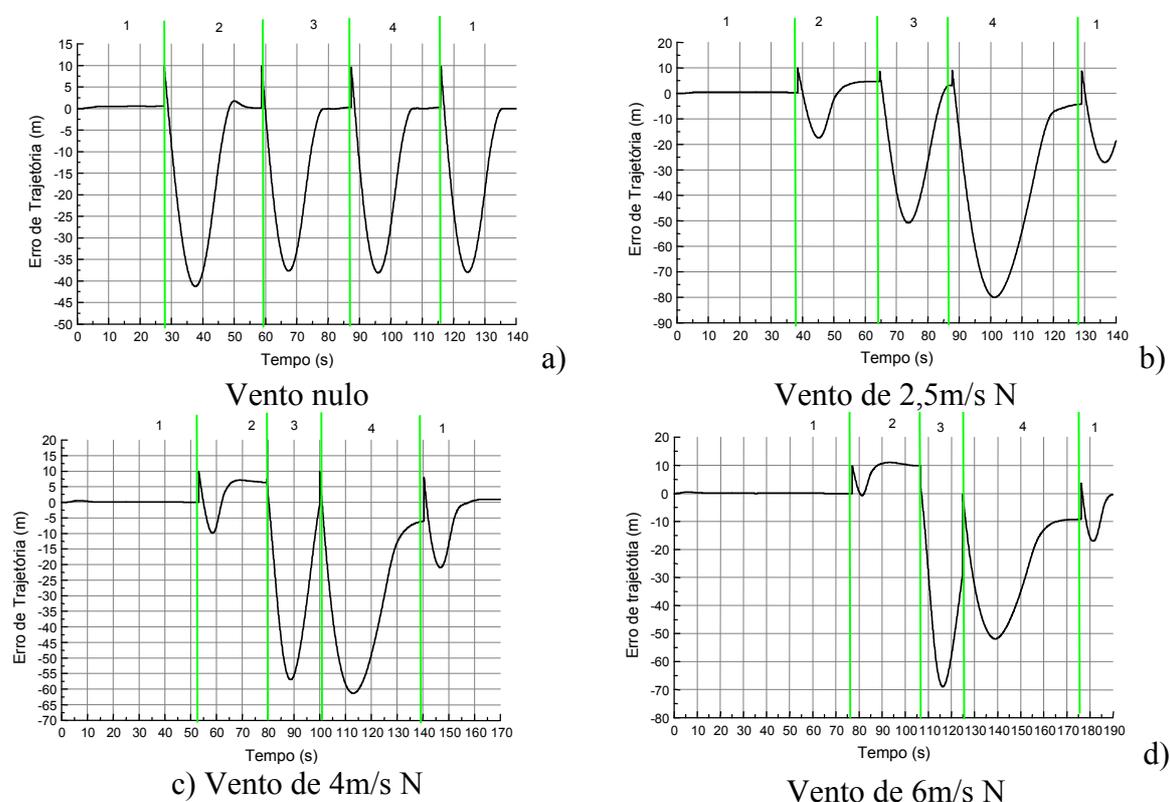


Figura 8.8: Erro de trajetória com variação de vento.

Assim, dos resultados mostrados na Figura 8.7 e na Figura 8.8:

- observando-se o afastamento das setas, que são plotadas a intervalos regulares, pode-se deduzir que o trecho "1" apresenta menor velocidade inercial e maior velocidade relativa ao ar, onde as superfícies aerodinâmicas possuem maior eficiência em relação aos outros trechos;
- observando-se o trecho "2", na medida em que o vento aumenta, o ângulo entre o veículo e a trajetória aumenta – o dirigível avança “de lado”, o que possibilita

a geração de uma componente de velocidade contra o vento, necessária para a realização do movimento; no limite, este ângulo tenderia a atingir  $90^\circ$ , mesmo assim o erro em regime permanente para este trecho aumenta com a intensidade do vento;

- observando-se o trecho "3", verifica-se uma degradação na trajetória que cresce com a intensidade do vento de popa, sendo que mesmo com uma maior velocidade inercial, observada pelo afastamento de setas em tempo menor para executar o trecho, a velocidade relativa ao ar é menor e as superfícies aerodinâmicas possuem menor eficiência em relação aos outros trechos;

As características mostradas acima ilustram a capacidade do algoritmo de controle de trajetória de compensar a ação do vento no vôo do dirigível, sendo inevitável alguma deterioração na trajetória em condições de vento forte.

### ***Efeito da Turbulência no controle de trajetória***

Na Seção 3.6.3 mencionou-se a inclusão da função Dryden como uma componente estocástica para a turbulência do vento. Um dos parâmetros da função Dryden é a severidade da condição atmosférica, com um parâmetro que varia de 1 para tempo calmo a 7 para uma condição extremamente turbulenta. A Figura 8.9 apresenta o efeito da turbulência para o caso de controle de trajetória, usando-se para os parâmetros de sintonia dos controladores os mesmos valores citados anteriormente. Além da trajetória é apresentado um vetor representando o ângulo de guinada do veículo.

Este resultado mostra que situações imprevistas, como a ocorrência de turbulências, afetam em muito o desempenho do controlador de trajetória, provocando desvios nesta de 15m para condições de tempo calmo (intensidade da turbulência 1) até 25m com pequenas instabilidades (intensidade da turbulência 2,3), diminuindo a repetibilidade da trajetória sendo executada.

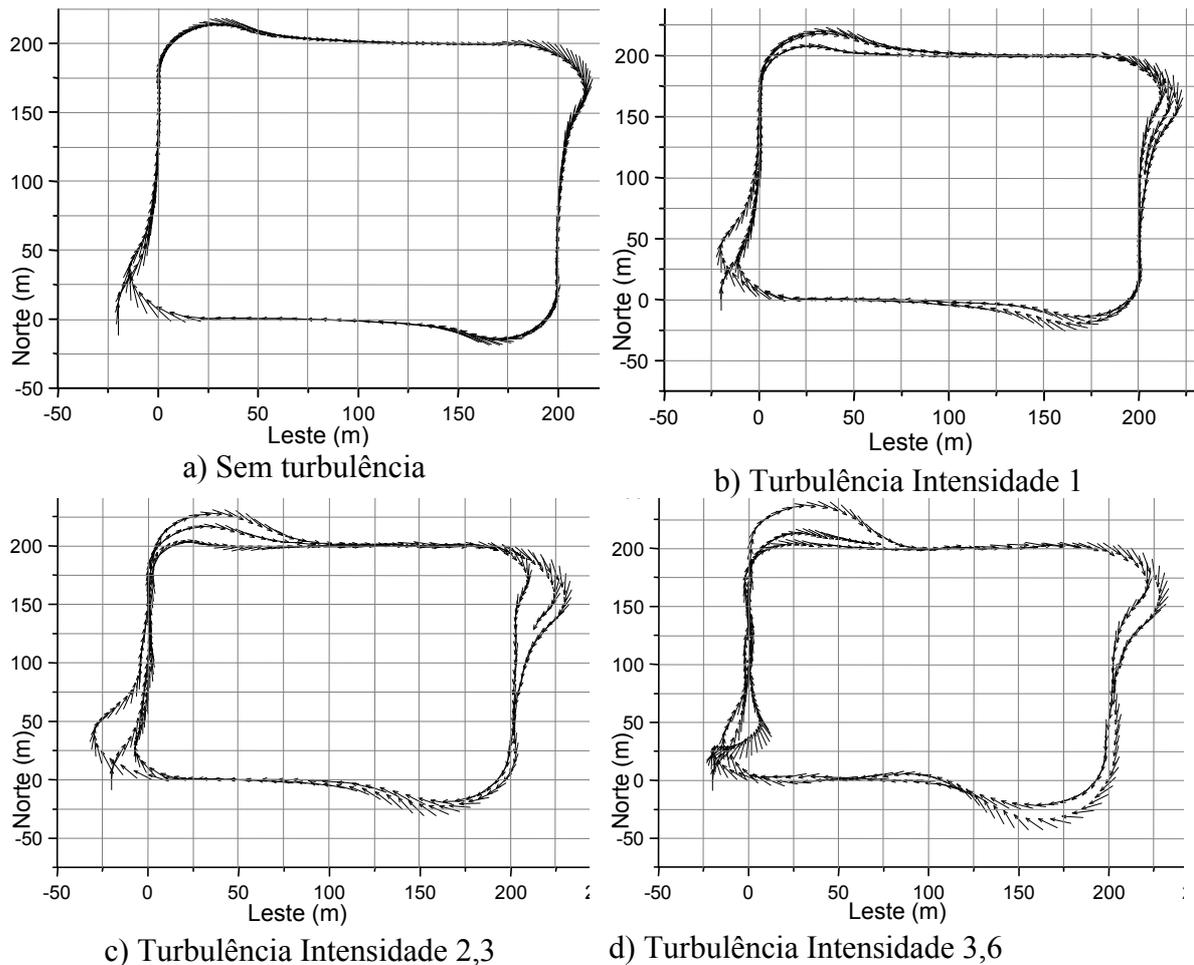


Figura 8.9: Efeito da turbulência na resposta do controlador de trajetória

## 8.4 Controle de Altitude: Simulação

Esta seção apresenta o ajuste dos parâmetros de sintonia e o estudo do comportamento do algoritmo de controle de altitude descrito na Seção 8.2. Os parâmetros de sintonia em questão são  $K_{EH}$  (ganho proporcional de realimentação do erro de altitude),  $K_{VH}$  (ganho proporcional de realimentação da velocidade vertical),  $K_{AH}$  (ganho proporcional de realimentação do ângulo de arfagem) e  $K_{RAH}$  (ganho proporcional de realimentação da velocidade de arfagem), segundo diagrama de controle da Figura 8.1 e a Tabela 8.1.

Da mesma forma que para o algoritmo de controle de trajetória, as simulações utilizam o ambiente de CACSD em MATLAB e o Ambiente de Desenvolvimento e Operação, com os algoritmos de controle implementados em código "C", compilados para posterior migração para a infra-estrutura embarcada. No ajuste dos ganhos do algoritmo de controle de altitude são considerados os diversos fatores envolvidos no comportamento

deste, incluindo a influência: dos parâmetros do próprio algoritmo, da transição de trajetórias, da direção do vento, da fluutuabilidade do dirigível, da configuração dos atuadores aerodinâmicos e de perturbações como vento e turbulências. Como critério de desempenho foram considerados o comportamento na manutenção da altitude de referência frente às perturbações e o comportamento em decolagem, que deve ser realizada em menor tempo sem a ocorrência de sobre-elevação. Ressalta-se que o procedimento de decolagem aqui considerado é o mais simplista, onde para uma determinada velocidade de rotação dos motores, de valor constante, usa-se somente a deflexão das superfícies aerodinâmicas como elementos de atuação. Não se recorre portanto à vetorização do grupo propulsor, nem se contempla procedimentos de decolagem vertical.

Nesta seção é também analisado o funcionamento conjunto dos controladores de trajetória e de altitude.

#### 8.4.1 Ajuste dos Controladores sem a Presença de Vento.

O ajuste dos parâmetros do algoritmo de controle de altitude foi realizado inicialmente sem a presença de vento. Considera-se que o veículo saia de altitude nula e tenha iniciado procedimento de decolagem no instante  $t=0$  para uma altitude de referência de 50m e cada um dos motores comandados a 8000 rpm.

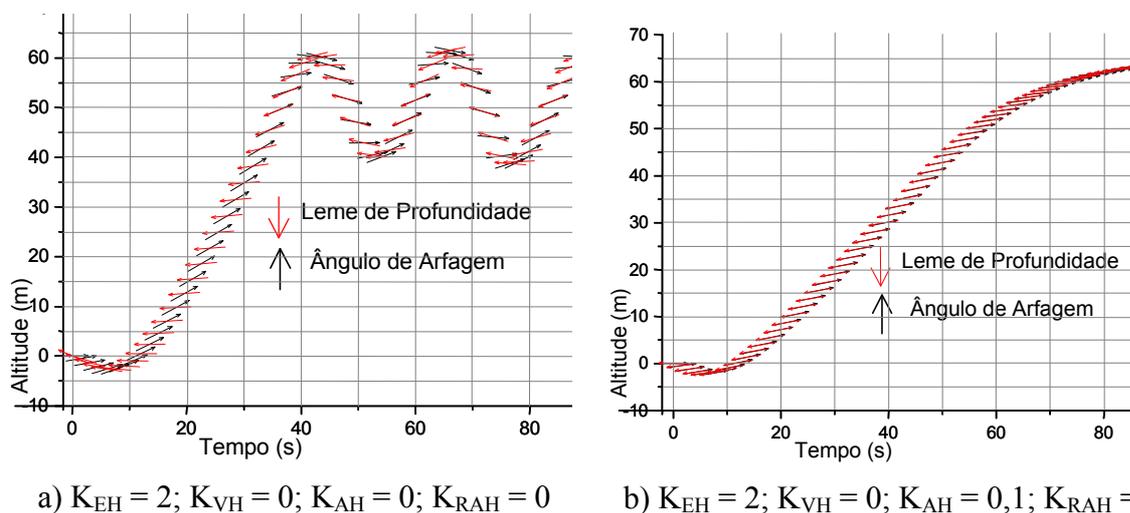


Figura 8.10: Ajustes dos parâmetros do controlador de altitude.

A Figura 8.10 apresenta o comportamento do algoritmo de controle de altitude para diferentes valores dos parâmetros do controlador. Ela também apresenta vetores que representam o ângulo de arfagem instantâneo do veículo em relação ao eixo horizontal (em

tom mais escuro), e o comando do leme de profundidade em relação à direção do ângulo de arfagem (em tom mais claro). Isto permite a visualização simultânea da altitude, ângulo de arfagem e do comando do leme de profundidade.

A Figura 8.10-a apresenta a resposta quando se utiliza unicamente o erro de altitude para comandar os lemes de profundidade, sem a malha interna de controle do ângulo de arfagem; as demais figuras apresentam o comportamento do algoritmo para diferentes valores dos parâmetros dos controladores.

Nas condições de simulação não se detectou melhoria no desempenho com a realimentação da velocidade vertical e da razão de arfagem, através dos ganhos  $K_{VH}$  e  $K_{RAH}$  respectivamente. Observa-se da Figura 8.10 -c a f, da Figura 8.10 -i e da Figura 8.10 -j, que com o incremento do ganho  $K_{AH}$  da malha interna de ângulo de arfagem reduz-se a oscilação de altitude, mostrando que este ganho tem um efeito análogo a um aumento de amortecimento. A única exceção a essa regra foi o caso "j", onde foi requerido também o uso da realimentação de velocidade vertical ( $K_{VH} \neq 0$ ), para se conseguir diminuir a sobre-elevação na altitude.

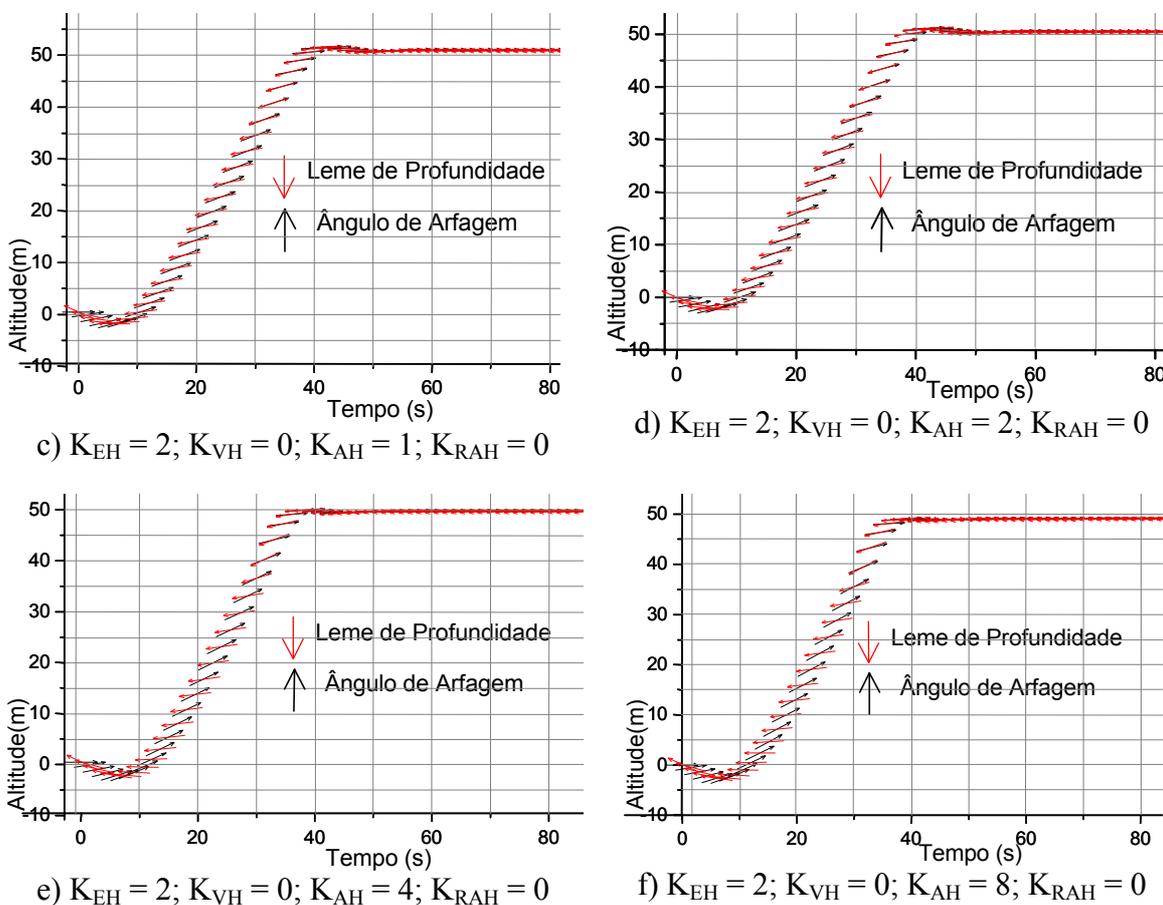
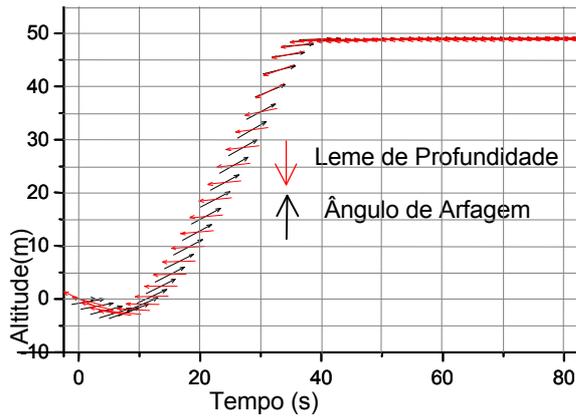
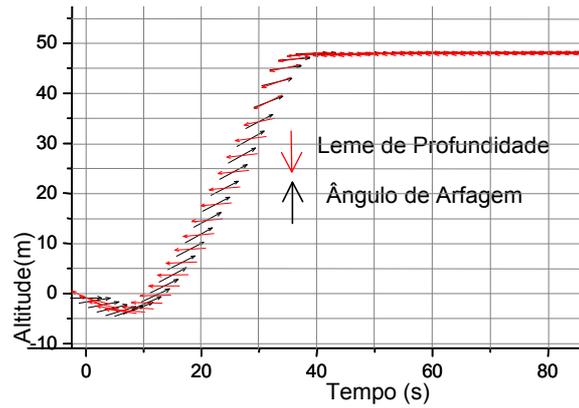


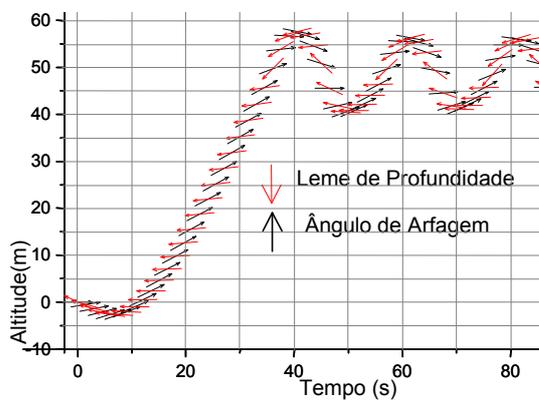
Figura 8.10 (cont): Ajustes dos parâmetros do controlador de altitude.



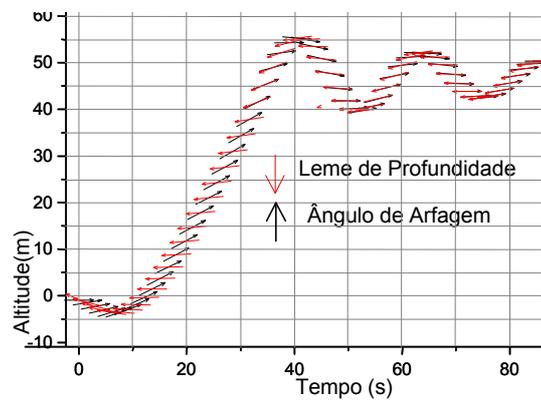
g)  $K_{EH} = 2; K_{VH} = 0; K_{AH} = 20; K_{RAH} = 0$



h)  $K_{EH} = 4; K_{VH} = 0; K_{AH} = 20; K_{RAH} = 0$



i)  $K_{EH} = 8; K_{VH} = 0; K_{AH} = 20; K_{RAH} = 0$



j)  $K_{EH} = 8; K_{VH} = 20; K_{AH} = 40; K_{RAH} = 0$

Figura 8.10 (cont): Ajustes dos parâmetros do controlador de altitude.

O conjunto de ganhos dos controladores que resultaram em melhor desempenho na Figura 8.10 são apresentados na Tabela 8.2.

Referência	Figura 8.10	$K_{EH}$	$K_{VH}$	$K_{AH}$	$K_{RAH}$
A	“e”	2	0	4	0
B	“f”	2	0	8	0
C	“h”	4	0	20	0

Tabela 8.2: Conjunto de ganhos que apresentaram melhor desempenho para o controle de altitude em simulação.

No que segue, os conjuntos de ganhos da Tabela 8.2 são avaliados em outras condições.

### 8.4.2 Efeito da Execução Simultânea dos Algoritmos de Controle de Trajetória e de Altitude para Mudanças de Direção

Nesta seção é avaliado o comportamento dos conjuntos de ganhos A, B e C da Tabela 8.2 frente à execução simultânea dos algoritmos de controle de trajetória e altitude, com cada um dos motores comandados a 8000 rpm. O controle de trajetória tem como ganhos  $K_{PT} = 3,6$ ,  $K_{IT} = 0,1$ ,  $K_{VT} = 7,5$ ,  $K_{RT} = 1,5$  e  $K_{RRT} = 0$ . A trajetória consiste num quadrilátero com 300m de aresta, havendo portanto mudanças abruptas de direção; o intuito é de observar o efeito destas mudanças, e a ação do controle de trajetória, no comportamento do algoritmo de controle de altitude.

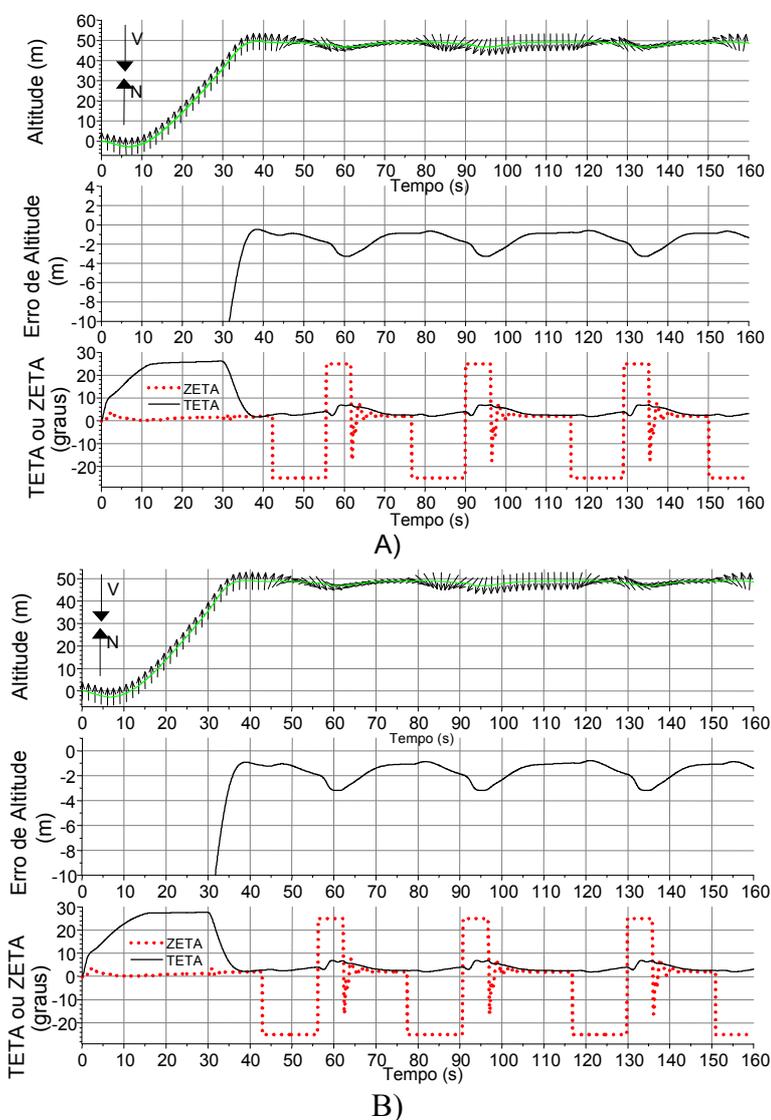


Figura 8.11: Efeito na altitude devido à mudança abrupta de trajetória

Assim, para os conjuntos de ganhos A, B e C, a Figura 8.11 apresenta a altitude do dirigível, o erro de altitude, o comando do leme de direção "ζ" (ZETA) (traço pontilhado),

o ângulo arfagem “ $\theta$ ” (TETA) (traço contínuo) e um vetor sobreposto à altitude do veículo (em tom mais escuro) representando a sua direção instantânea, com seu ângulo de guinada “ $\psi$ ” sendo medido em relação à direção Norte, mostrada na figura como “N”, permitindo observar visualmente a direção do veículo em relação à direção do vento, que é indicada na figura como “V”.

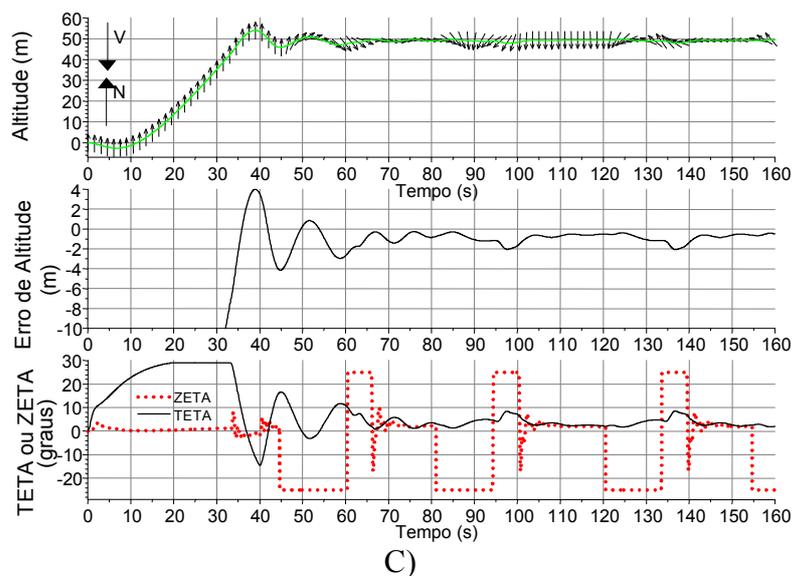


Figura 8.11(cont): Efeito na altitude devido à mudança abrupta de trajetória.

Os resultados apresentados na Figura 8.11 permitem observar:

- Que após o fim das oscilações advindas da mudança abrupta de direção, o erro máximo de altitude é de 3m para os conjuntos "A" e "B" e de 1m para o conjunto "C".
- Que há coincidência da saturação do comando do leme de direção com o momento em que se dá a mudança de direção, e a existência de um pico no erro de altitude neste momento, mostrando a existência da influência da execução da curva com a perda de altitude do veículo. Isto é causado pela diminuição da velocidade relativa do veículo com o ar devido à realização da curva, como consequência do aumento de arrasto causado pela atuação das superfícies aerodinâmicas e pela existência de uma maior área frontal do veículo em relação ao ar nesta situação.
- Que adicionalmente pode-se observar que o ângulo de arfagem (TETA) possui valores sempre maiores que zero, para que o veículo possa gerar a força de sustentação requerida pelo seu peso, pois neste ensaio ele voa “pesado” em 3,5 Kg (o aspecto de flutuabilidade é abordado especificamente na Seção 8.4.4).

### 8.4.3 Controle de Altitude e de Trajetória com Vento

Visando melhor observar o comportamento do algoritmo de controle de altitude operando conjuntamente com o de trajetória, frente a diferentes orientações de vento em relação ao veículo, realizou-se movimentos simultâneos com controle de altitude e de trajetória. A trajetória especificada para realização pelo dirigível corresponde ao mesmo quadrilátero descrito na Seção 8.3.1, bem como os ganhos já citados:  $K_{PT} = 3,6$ ,  $K_{IT} = 0,1$ ,  $K_{VT} = 7,6$ ,  $K_{RT} = 1,5$  e  $K_{RRT} = 0$ .

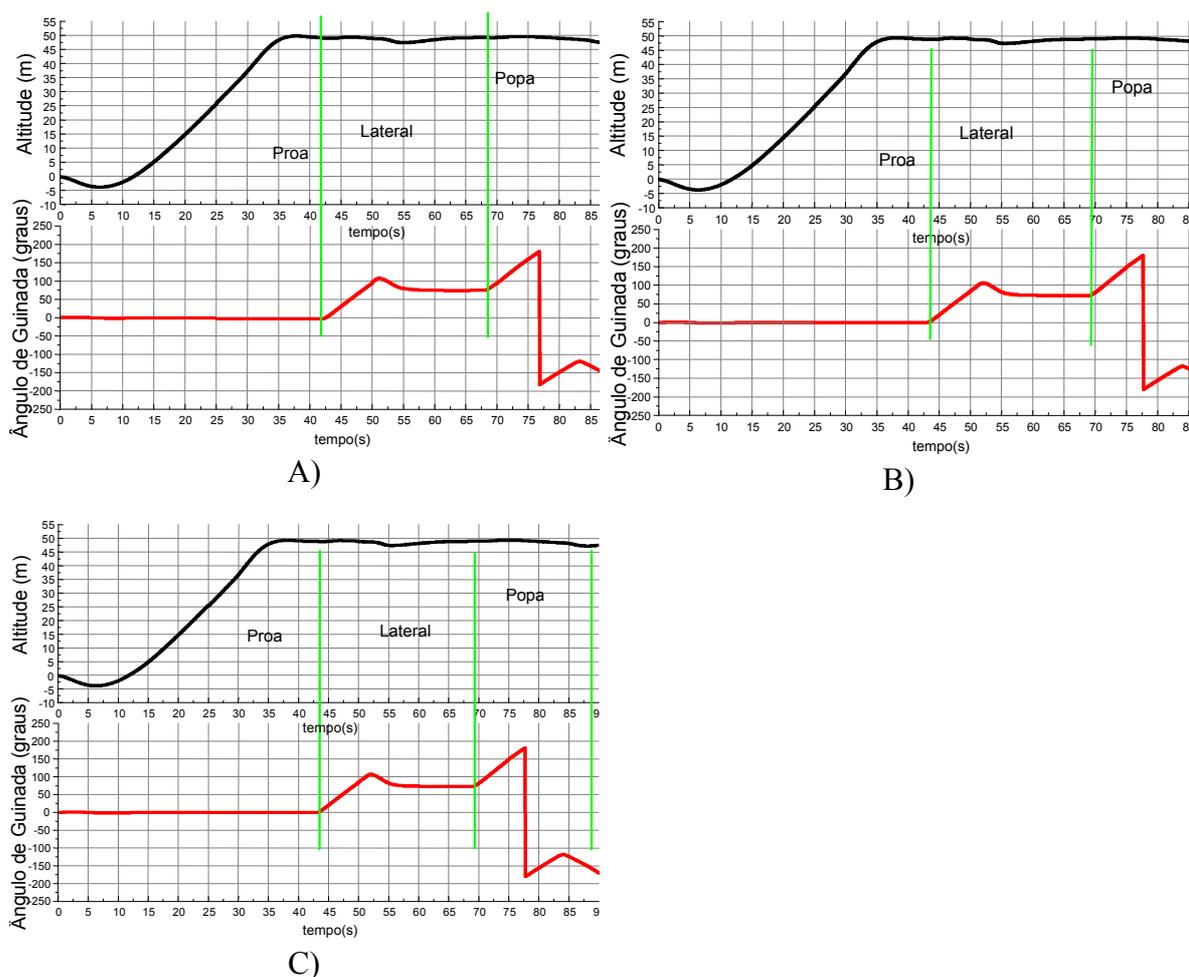


Figura 8.12: Efeito da variação da altitude com o vento (a 2,5m/s) e sua direção

O vento foi fixado com direção Norte e com intensidade dependente do ensaio. O vento com direção fixa e o uso de controle de trajetória permite posicionar o dirigível em diferentes situações de vento dependendo do trecho, ou seja vento de proa, vento lateral, vento de popa e novamente vento lateral; por conseguinte, nestas situações, diferentes valores de velocidade relativa ao ar são encontradas, o que sabidamente influencia o comportamento dinâmico do dirigível.

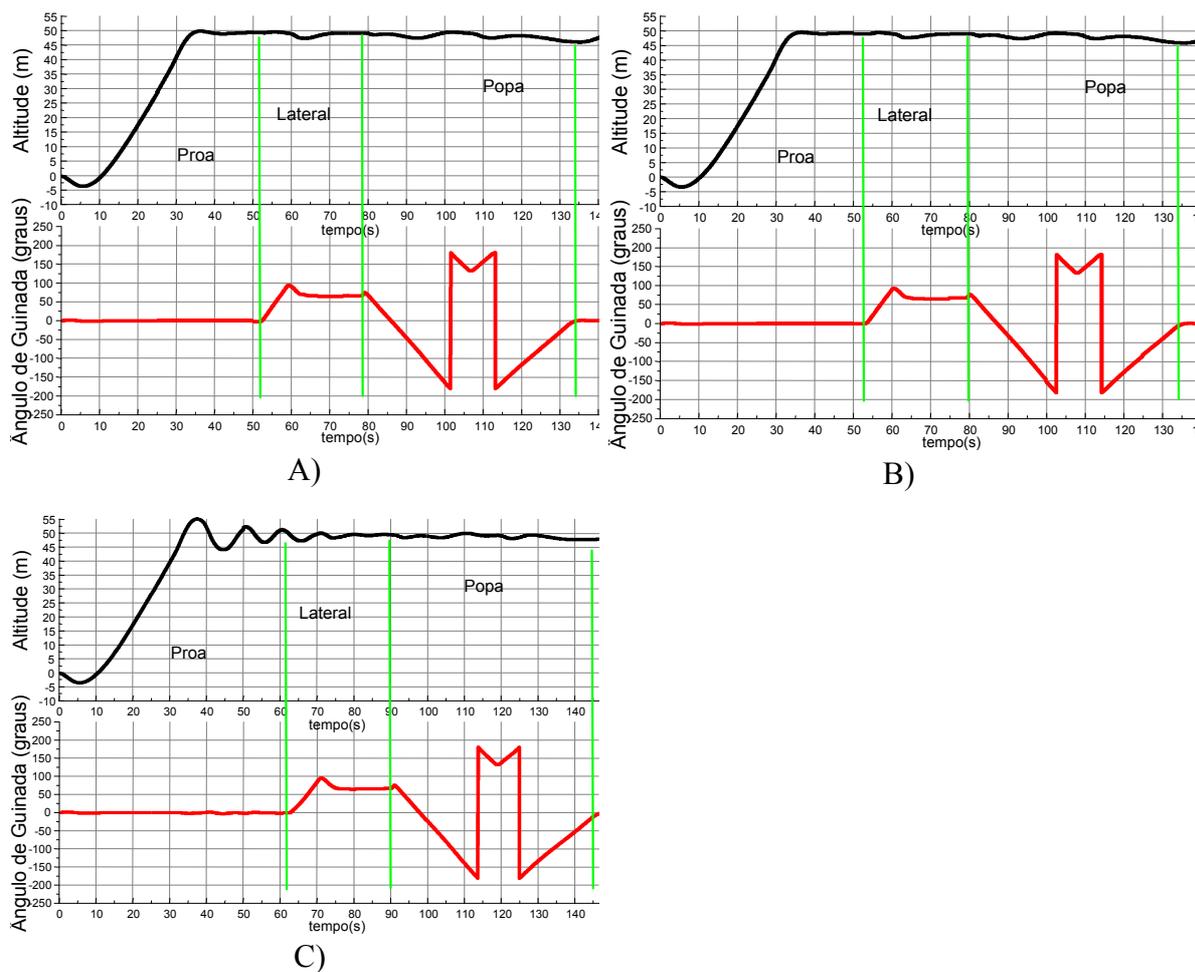


Figura 8.13: Efeito da variação da altitude com o vento (a 4m/s) e sua direção

No ensaio, devido à trajetória em forma de quadrilátero, houve transições drásticas entre os segmentos de reta que a compunham, de maneira a avaliar também o efeito destas transições no comportamento do sistema de controle. Da Figura 8.12 à Figura 8.14, mostra-se o comportamento em altitude do dirigível para os ganhos mostrados na Tabela 8.2, com ventos fracos (2.5m/s), ventos médios (4 m/s) e ventos fortes (6 m/s), com os motores comandados a 8000 rpm. A título de comparação, em situação sem vento, com esta rotação dos motores a velocidade do dirigível seria de 9 m/s.

Nas figuras são apresentadas a altitude juntamente com o ângulo de guinada atual do veículo “ $\psi$ ”, bem como a situação do veículo em relação ao vento, ou seja vento de proa, vento lateral ou vento de popa.

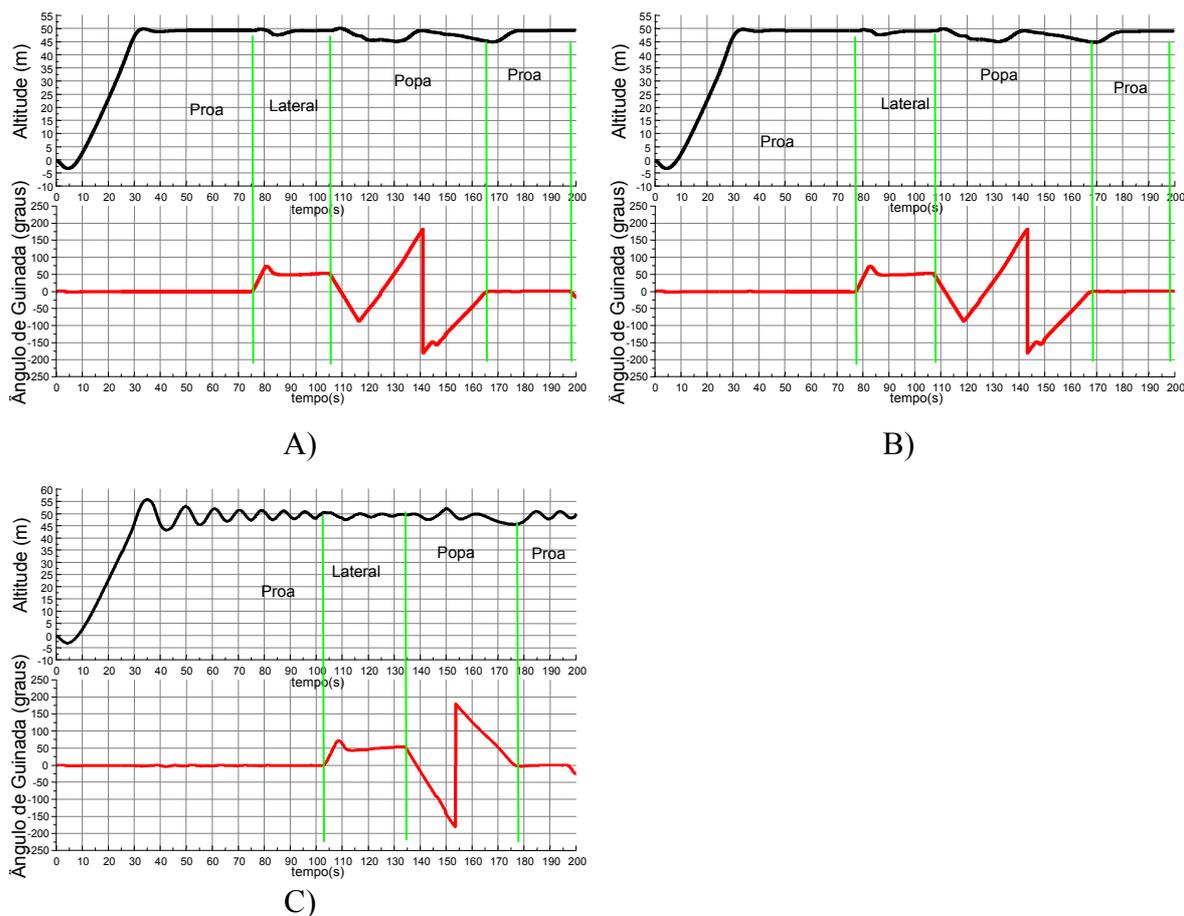


Figura 8.14: Efeito da variação da altitude com o vento (a 6m/s) e sua direção

Observa-se dos resultados mostrados que, após o fim das oscilações advindas da mudança abrupta de direção na trajetória, o erro máximo de altitude é de 3m para os conjuntos "A" e "B", e de 2m para o conjunto "C" para ventos fracos e moderados nos casos da Figura 8.12 e da Figura 8.13. Em todos os casos, coincidindo com a mudança de direção da trajetória e com a fase de estabelecimento de nova direção, há um pico no erro de altitude, mostrando a existência da influência da execução da curva com a perda de sustentação do veículo, que é causada pela diminuição da velocidade relativa do veículo com o ar, conjugado ao fato de o dirigível estar operando “pesado” em 3,5Kg como comentado anteriormente; este aspecto de fluvariabilidade é abordado na próxima seção.

Dos resultados obtidos, comparando-se com a situação sem vento mostrada na Seção 8.4.2, observou-se que:

- No trecho onde o vento é de proa e, portanto, a velocidade relativa ao ar é maior, o conjunto C de ganhos da Tabela 8.2 mostrou maior sobre-elevação e sub-amortecimento, indicando a necessidade de ganhos de menor valor;
- No trecho com vento lateral, os controladores conseguiram recuperar a altitude face à perturbação de mudança de trajetória, sendo que a oscilação quando da utilização do conjunto C foi menor;
- No trecho com vento de popa, a velocidade relativa ao ar é menor, há uma menor eficiência aerodinâmica e, à exceção do conjunto C, todos os outros conjuntos possuem erros de altitude maiores e respostas mais lentas, mostrando a necessidade de ganhos mais elevados.

Este exemplo, face à variação de comportamento dinâmico do dirigível em função da velocidade relativa ao ar (no caso, devido ao vento), indica novamente que o uso de parâmetros fixos, ajustados para uma determinada situação de vôo, não é adequado para cobrir todas as situações de velocidade relativa ao ar com o mesmo desempenho.

#### **8.4.4 Comportamento em Função da Flutuabilidade**

Esta seção aborda a influência da flutuabilidade do dirigível no comportamento de seu controle em altitude e a influência da execução conjunta do controle de trajetória. Como foi explicitado na Seção 3.4.6, a noção de flutuabilidade diz respeito ao peso do dirigível, mais especificamente, à diferença entre as forças aerostática e gravitacional que nele atuam.

Nas seções precedentes, os conjuntos de ganhos da Tabela 8.2, bem como os parâmetros  $K_{PT} = 3,6$ ,  $K_{IT} = 0,1$ ,  $K_{VT} = 7,6$ ,  $K_{RT} = 1,5$  e  $K_{RRT} = 0$  para o controle de trajetória, foram obtidos com o veículo 3,5Kg pesado; o comportamento do dirigível, controlado por estes conjuntos, foi mostrado na Seção 8.4.2. Em contraposição a esta situação, a Figura 8.15 apresenta o comportamento do dirigível operando com flutuabilidade neutra e vento nulo, quando da realização simultânea de controle de altitude e trajetória, para os conjuntos de ganhos do controle de altitude mostrados na Tabela 8.2, mantendo-se os ganhos já citados para o controle de trajetória. Nesta figura são apresentados a altitude, o erro de altitude, o comando do leme de direção " $\zeta$ " (ZETA) (pontilhado em tom mais claro), o ângulo de arfagem " $\theta$ " (Teta) e um vetor indicando o

ângulo de guinada atual do veículo “ $\psi$ ” sobreposto à sua altitude , representando a direção instantânea do veículo.

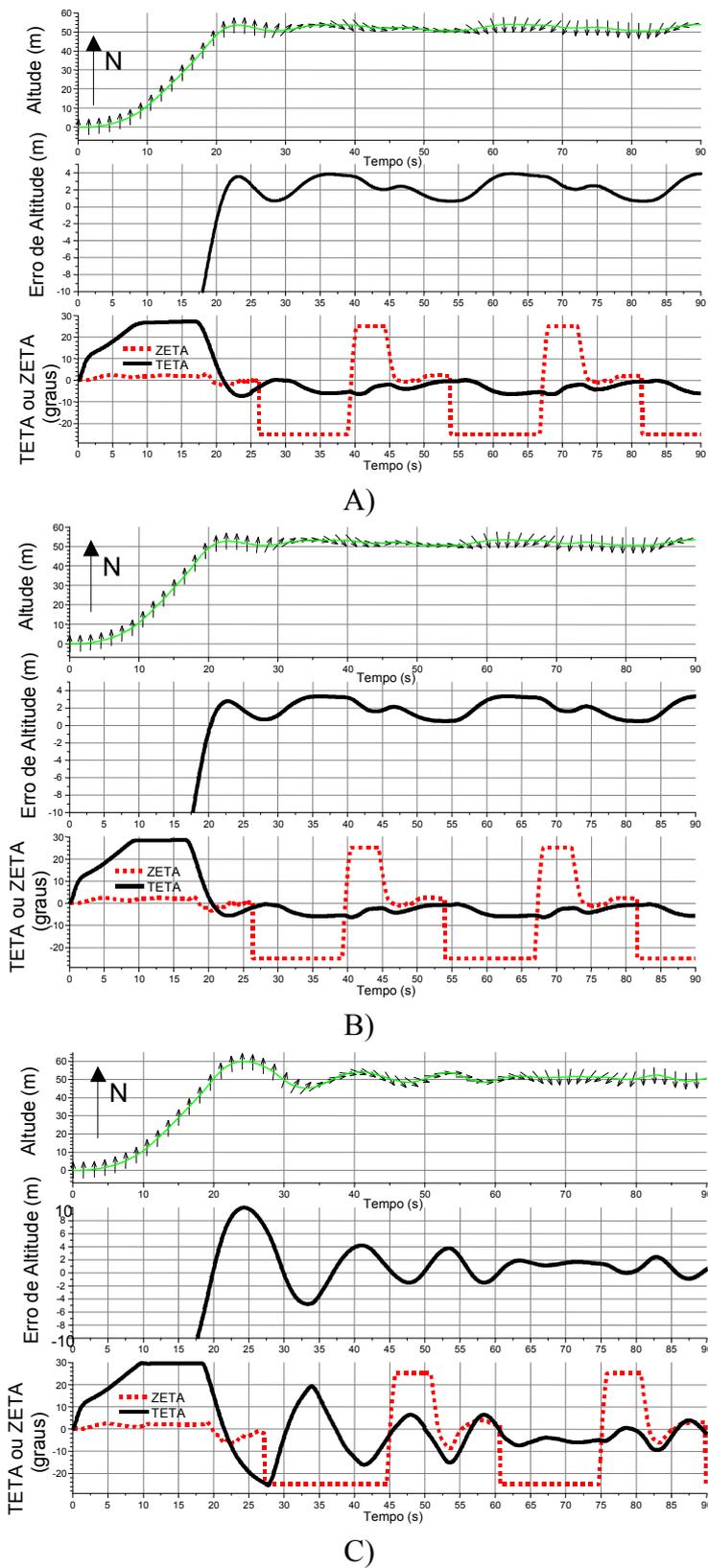


Figura 8.15: Influência da variação da flutuabilidade.

Comparando-se a Figura 8.15 (flutuabilidade neutra) com a Figura 8.11 (flutuabilidade de 3,5 Kg “pesado”), que à exceção da flutuabilidade foram obtidas nas mesmas circunstâncias, observa-se que:

- quando de uma mudança de trajetória, o veículo ganha altitude quando está neutro e perde altitude quando está pesado;
- com os parâmetros de sintonia utilizados, o erro tende a ser positivo na condição de equilíbrio aerostático e negativo na situação pesado; note-se que, fora da situação de neutralidade, o vôo pesado leva naturalmente a um erro de regime devido ao controle ser unicamente proporcional;
- na condição de flutuabilidade neutra o dirigível está usando os lemes de profundidade para evitar que o veículo suba, observando-se que durante todo o movimento o ângulo de arfagem (TETA) “ $\theta$ ” é negativo, indicando que o veículo possui um ângulo de ataque negativo, gerando uma força aerodinâmica para baixo.
- na condição de vôo “pesado” ao contrário, para compensar o peso do veículo e evitar que ele perca altitude, o ângulo de arfagem é positivo e a força aerodinâmica resultante é para cima.

#### ***8.4.5 Efeito da Conversão “+” para “x” nas Superfícies Aerodinâmicas de Atuação nos Controles de Altitude e de Trajetória***

Conforme mostrado na Seção 3.3.2, a conversão de comandos do arranjo de superfícies aerodinâmicas de atuação em cruz “+” para a configuração em “x” do AS800, quando não se utiliza comando de rolagem (i.e. “ $\xi$ ” = 0), é dada pela Equação 3.4 e Equação 3.5, reproduzidas a seguir. Nesta condição, onde  $\eta$  e  $\zeta$  são os comandos correspondentes do leme de profundidade e de direção respectivamente, tem-se que:

$$\delta_1 = \delta_3 = \eta + \zeta \quad 8.1$$

$$\delta_2 = \delta_4 = \eta - \zeta \quad 8.2$$

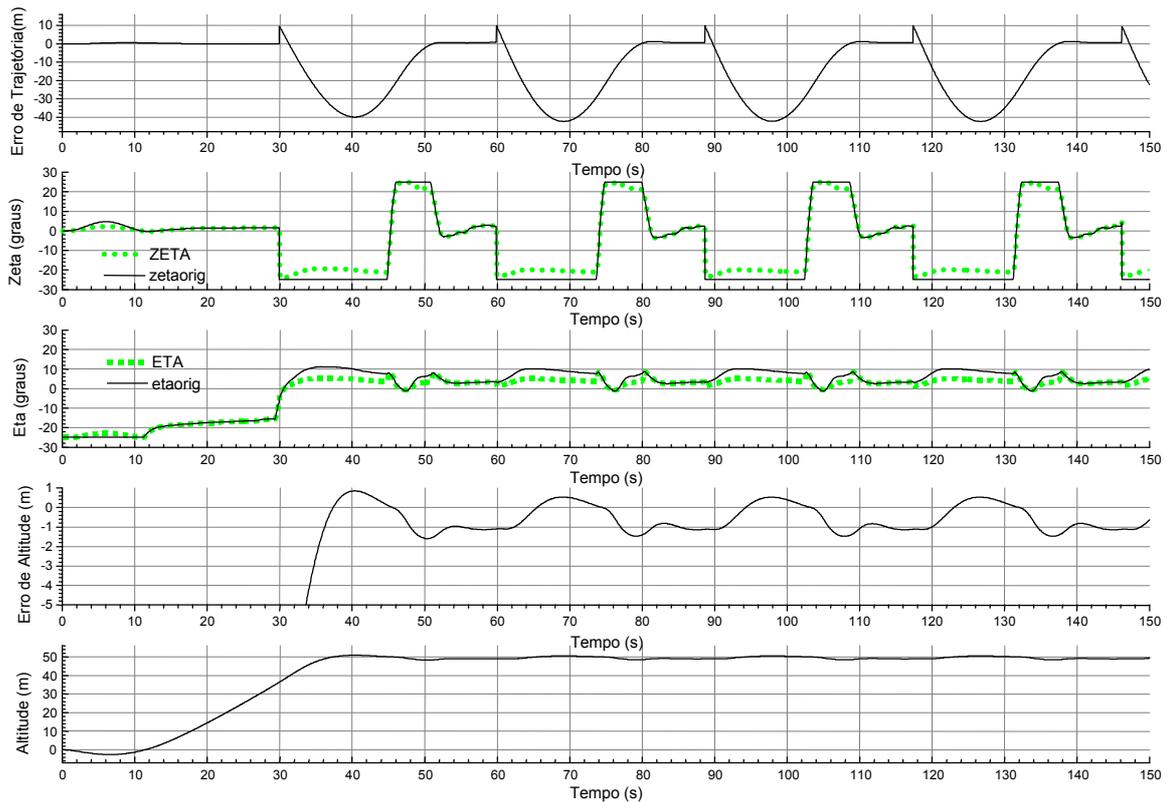


Figura 8.16: Comportamento das varias variáveis para trajetória completa considerando a conversão de “+” para “x” para o conjunto de ganhos “A” da Tabela 8.2.

Como no caso do AS800 as deflexões  $\delta_i$  ( $i = 1, 2, 3, 4$ ) são limitadas a  $\pm 25^\circ$ , combinações de  $\eta$  e  $\zeta$  gerando  $\delta_i$  que ultrapassem estes valores, serão automaticamente limitadas a  $\pm 25^\circ$ . A ocorrência deste efeito de saturação é comum em manobras abruptas, como quando de mudanças importantes na direção de trajetória. Nestas situações de saturação, as relações dadas pelas equações acima perdem sua validade, introduzindo conseqüentemente comandos que não respeitam as componentes originais de arfagem e guinada, resultando em movimentações inesperadas do dirigível. Nas transformações onde há saturação, para evitá-la e, ao mesmo tempo manter a proporção entre as componentes de arfagem e guinada, os valores de  $\eta$  e  $\zeta$  poderiam ser ponderados igualmente, o que ainda não foi utilizado no dirigível, seja em simulação, seja no sistema real embarcado.

Mantendo-se os parâmetros do controlador de trajetória anteriores ( $K_{PT} = 3,6$ ,  $K_{IT} = 0,1$ ,  $K_{VT} = 7,6$ ,  $K_{RT} = 1,5$  e  $K_{RRT} = 0$ ), a Figura 8.16 até Figura 8.18 mostram o efeito da saturação na conversão de “+” para “x”, quando se utiliza  $\eta_{MAX} = \zeta_{MAX} = 25^\circ$ , e  $\eta_{MIN} = \zeta_{MIN} = -25^\circ$ , com o conjunto de ganhos A, B e C especificados na Tabela 8.2. Na figura tem-se a altitude, o erro de altitude, o comando original do leme de direção (zetaorig) “ $\zeta$ ” e

do leme de profundidade " $\eta$ " (etaorig) em linhas contínuas, isso antes da conversão de "+" para "x"; então tem-se comando do leme de direção " $\zeta$ " (ZETA) e o comando do leme de profundidade " $\eta$ " (ETA) em linhas pontilhadas após o ciclo completo de conversão de "+" para "x", saturação e posterior reconversão para "+", indicando assim o que foi efetivamente comandado.

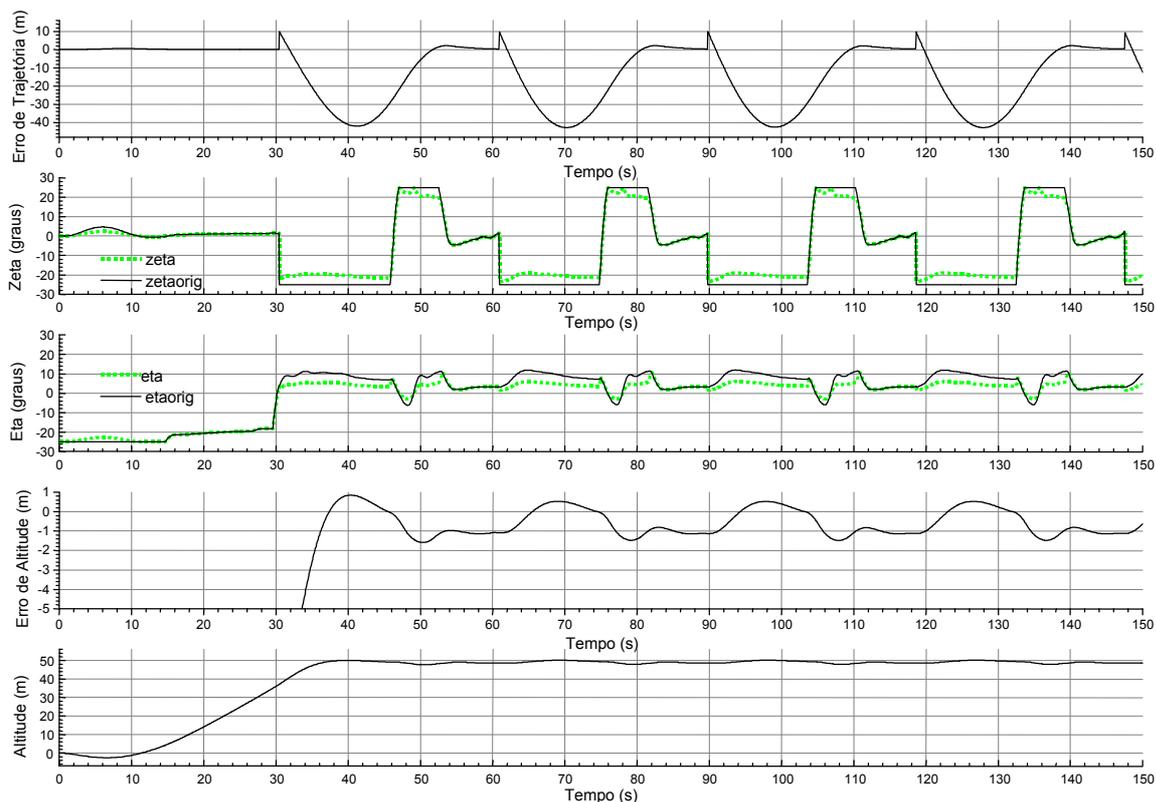


Figura 8.17: Comportamento das varias variáveis para trajetória completa considerando a conversão de "+" para "x" para o conjunto de ganhos "B" da Tabela 8.2.

Observa-se na Figura 8.16 a Figura 8.18 que o erro absoluto de altitude é de 2m para o caso "A" e "B" e oscilante em torno 2m para o caso "C", enquanto isso, o erro de trajetória máximo incrementou de aproximadamente 45m para o caso "A" e "B" para valores em torno de 75m para o caso "C".

Em relação à influência da conversão "x" para "+" nos comandos já no caso "A", pode-se notar como o valor do comando do leme de direção " $\zeta$ " que deveria se manter saturado tem seu valor real influenciado pelo o comando do leme de profundidade " $\eta$ ". Esta situação torna-se mais drástica quando o ganho do controlador de altitude aumenta, acontecendo a saturação simultânea dos dois comandos, fazendo com que os comandos caíssem de 25 para 12,5. Isto é observado com mais clareza entre  $t = 32$  e  $t = 48$ ,

principalmente na Figura 8.18. A consequência geral é a diminuição da intensidade dos comandos originais após a transformação, influenciada pelo fato de que o mesmo leme tem que “atender” os dois pedidos de comando para executar o movimento.

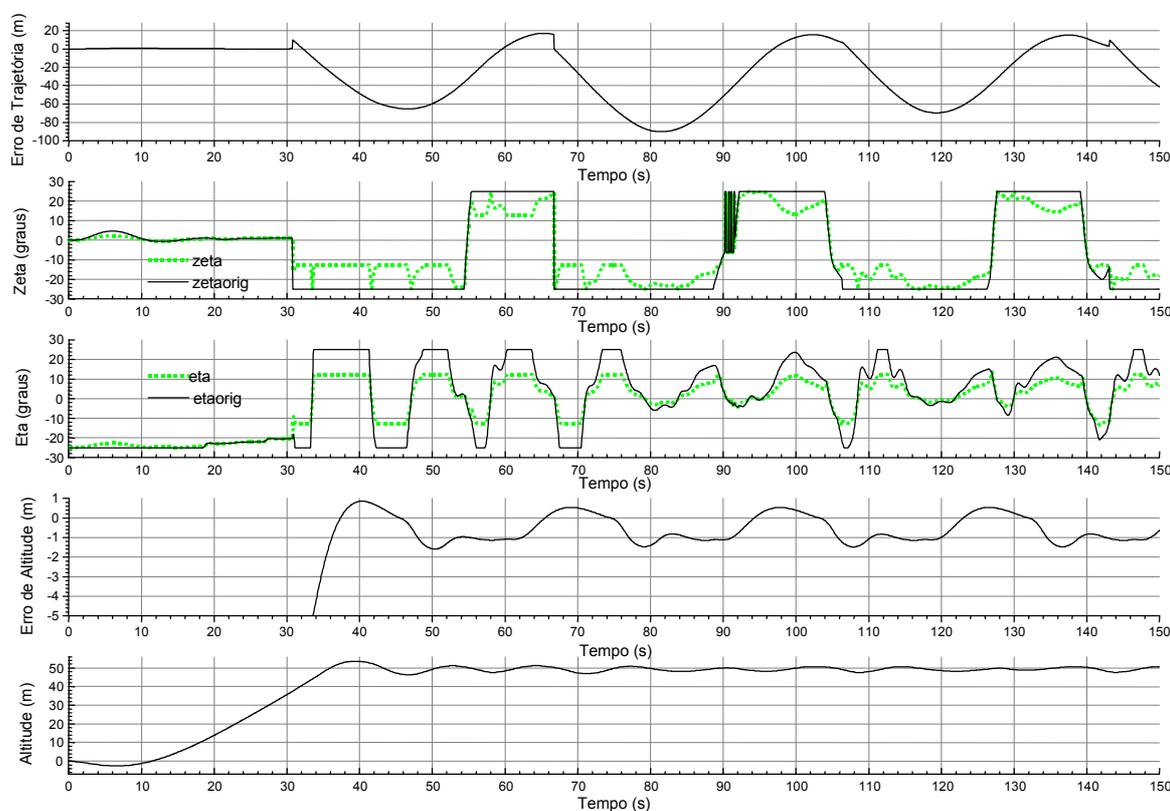


Figura 8.18: Comportamento das varias variáveis para trajetória completa considerando a conversão de “+” para “x” para o conjunto de ganhos “C” da Tabela 8.2.

A Figura 8.19 apresenta a trajetória resultante para o conjunto de ganhos A, B e C especificados na Tabela 8.2, com a resposta em altitude mostrada na Figura 8.18.

Pode-se observar que a trajetória correspondente à Figura 8.19-c nitidamente apresenta erro de trajetória superior aos casos “A” e “B”. Esta é a condição na qual o leme de direção “ $\zeta$ ” e o leme profundidade “ $\eta$ ”, como pode ser observado na Figura 8.18, têm seus comandos reduzidos de  $25^\circ$  para  $12,5^\circ$  numa condição de curva, fazendo com que o raio de curvatura da curva seja maior em consequência do menor comando efetivo do leme de direção.

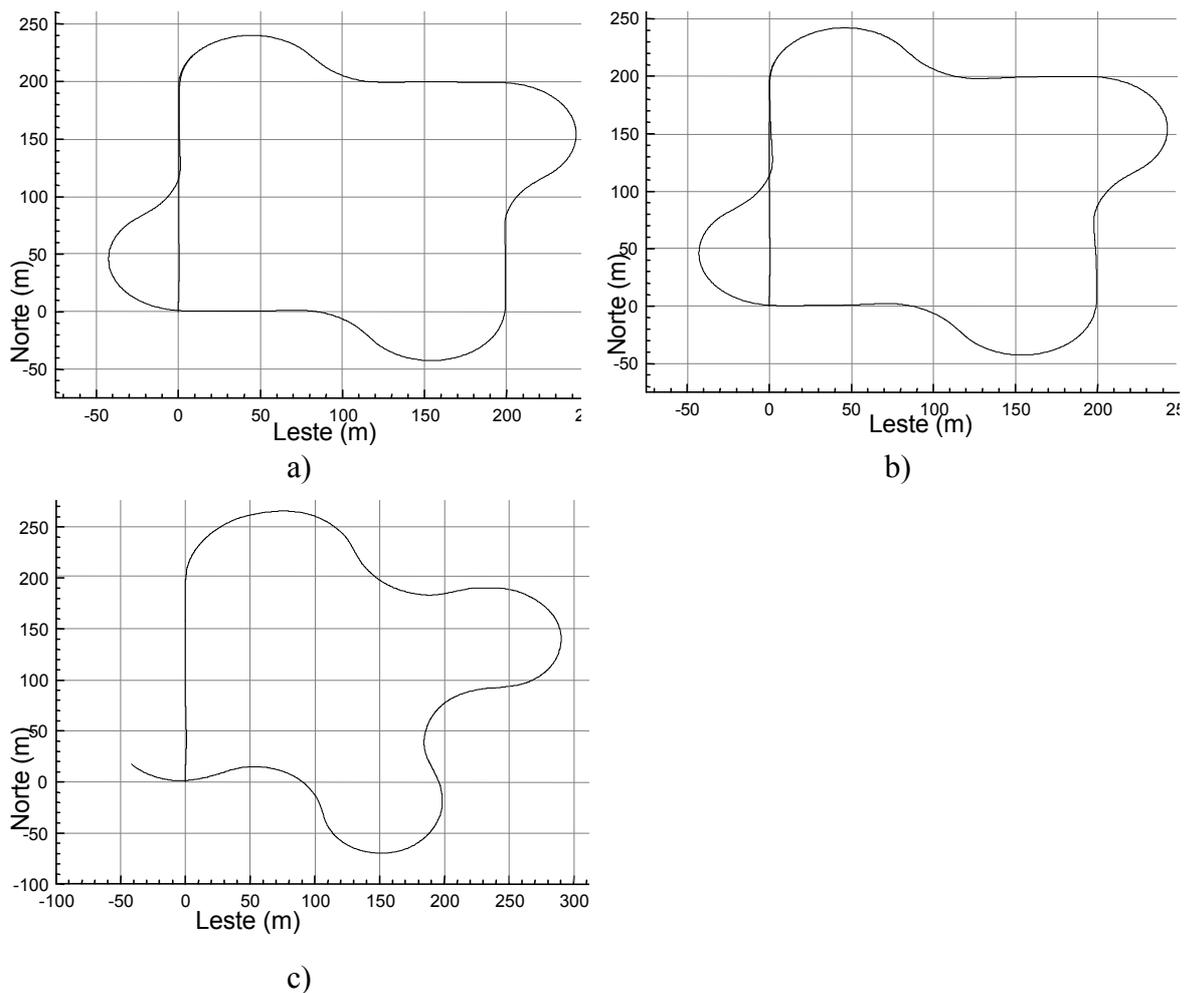
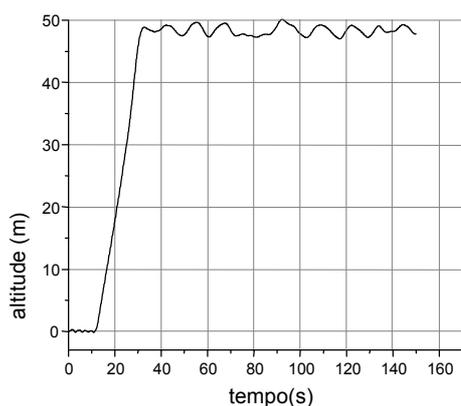


Figura 8.19: Trajetória completa considerando a conversão de “x’ para “+” para o conjunto de ganhos A (a), B (b) e C (c) especificados na Tabela 8.2.

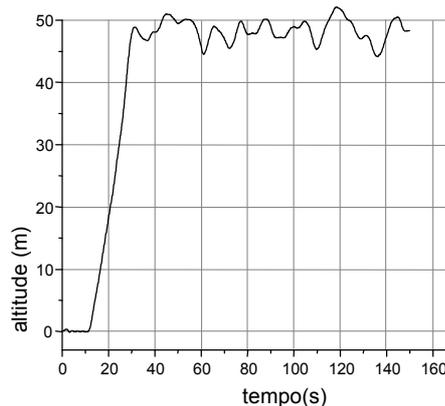
A influência da conversão “+’ para “x” dos comandos do leme de direção e profundidade pode ser facilmente evitada pelo uso de valores limites adequados para cada controlador, como no exemplo acima, de  $12,5^\circ$ , para os limites de comando de cada controlador, em vez de  $25^\circ$ . Esse valor pode ser estipulado por um sistema supervisor em função da manobra requerida pelo veículo, se curva, ou se subida; dando assim, mais autoridade à manobra mais importante naquele momento.

#### 8.4.6 O Efeito da Turbulência

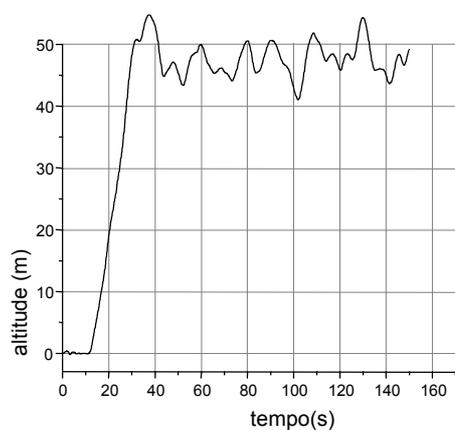
A Figura 8.20 apresenta o comportamento do controle de altitude em turbulência, para o conjunto de ganhos correspondentes ao caso B da Tabela 8.2. Como explicado anteriormente na página 148, o nível de turbulência varia de 1 para tempo calmo a 7 para uma condição extremamente turbulenta.



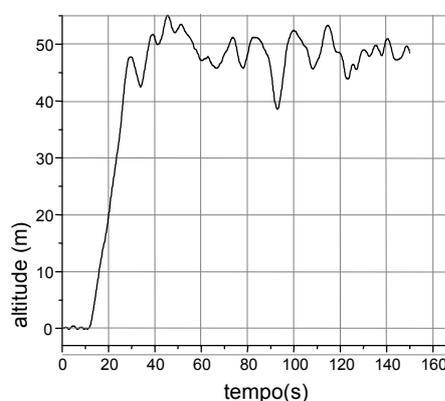
a) Turbulência Intensidade 1



b) Turbulência Intensidade 2.3



c) Turbulência Intensidade 3.6



d) Turbulência Intensidade 5

Figura 8.20: Efeito da turbulência na resposta do controlador de altitude.

Observa-se pela Figura 8.20 que, na medida em que se aumenta a intensidade da turbulência, o erro máximo em altitude aumenta de 2m na intensidade “1” para até 10m com intensidade forte “5”.

#### 8.4.7 *Resumo dos Resultados*

Os resultados de ajuste dos parâmetros de sintonia associados aos algoritmos de controle de trajetória e altitude, e a verificação de seu comportamento em diferentes condições, permitem algumas conclusões.

Os algoritmos utilizados se mostraram simples porém efetivos e uma vez sintonizados os parâmetros para uma condição de voo, o desempenho do sistema é satisfatório. Por outro lado, situações distintas de operação, como perturbações e condições de voo, notadamente a velocidade relativa ao ar que é dependente da orientação do veículo

em relação à direção do vento, faz com que o dirigível já não se comporte como na condição de vôo na qual o controle foi ajustado. Assim, conjuntos fixos de parâmetros de sintonia dos controladores não são adequados para todas as condições de vôo, havendo a necessidade da consideração do uso de técnicas mais elaboradas, como por exemplo *gain scheduling* (do Inglês escalonamento de ganhos), em função da velocidade relativa do veículo. Isto permitiria melhor explorar as variações das características aerodinâmicas do veículo com a condição de vôo e evitar a degradação de desempenho de uma condição para outra.

A avaliação do comportamento do controlador de trajetória com a variação dos parâmetros de sintonia da malha interna de controle do ângulo de guinada mostrou que:

i) o ajuste de parâmetros para um comportamento super-amortecido diminui a capacidade de seguimento de trajetória;

ii) o ajuste de parâmetros para que este tenha uma pequena tendência a sobre-elevação provoca oscilações no comando do leme de profundidade, sem conseqüências para a trajetória e;

iii) que a solução mais adequada é o uso de parâmetros de sintonia que produzam comportamentos próximos ao amortecimento crítico.

O uso de transições não suaves entre dois seguimentos de trajetória tem que ser reavaliado, pois este é um outro fator de degradação de desempenho, por obrigar os controladores a trabalharem em saturação durante o período de transição.

O ensaio conjunto do algoritmo de controle de trajetória com o algoritmo de controle de altitude, mostrou que as principais influências estiveram associadas às transições abruptas na trajetória e à conversão dos comandos de “+” para “x”. Em relação à conversão “+” para “x”, percebeu-se a importância da utilização de valor limite flexível para cada comando, de forma a este estar associado à condição de vôo e a um sistema supervisor. Seria papel do sistema supervisor definir a importância de um comando em relação aos demais em função da condição de vôo, por exemplo requerendo guinada de emergência seria dada prioridade ao leme de direção, utilizando uma participação flexível de um comando em relação aos demais na ocupação das deflexões das superfícies aerodinâmicas.

Os resultados obtidos em simulação permitiram também validar o código em "C" dos algoritmos de controle de trajetória e de altitude, implementados efetivamente no sistema embarcado no veículo e forneceram as direções para o processo de ajuste prático dos parâmetros dos algoritmos do dirigível em vôos reais.

## 8.5 Controle de Trajetória: Resultados Obtidos em Vôo

Esta seção mostra os resultados do procedimento prático de ajuste de parâmetros do algoritmo de controle de trajetória em vôos reais executados pelo dirigível, o que se faz necessário devido à diferença entre o modelo e ambientes associados disponíveis para simulação e a dinâmica real do veículo, conforme explicitado na introdução deste capítulo. O procedimento para a configuração do experimento e ajuste dos parâmetros de sintonia em vôo utiliza a interface do programador de missão mostrada na Seção 7.6.2, que permite:

i) especificar as coordenadas do quadrilátero que forma a trajetória, ou utilizar uma seqüência de pontos definidos pelo sistema de programação de missão;

ii) definir os valores dos ganhos dos algoritmos descritos na Tabela 8.1 ou seja  $K_{PT}$ ,  $K_{IT}$ ,  $K_{VT}$ ,  $K_{RT}$  e  $K_{RRT}$  para controle de trajetória e  $K_{EH}$ ,  $K_{VH}$ ,  $K_{AH}$  e  $K_{RAH}$  do controle de altitude e;

iii) os algoritmos de controle habilitados (usado no caso de vôos com controle de trajetória, de altitude, ou ambos, como será visto posteriormente).

Ressalta-se que alterações nos valores dos parâmetros e na habilitação de controladores pode ser efetuada durante o vôo. Durante toda a execução do vôo, os dados de telemetria gerados no sistema embarcado são transmitidos para a estação de operação em terra e nela armazenados, como descrito na Seção 7.5.5. Estes dados são utilizados para a análise do comportamento do sistema de controle e navegação e de sua ação sobre o veículo.

Entre os anos de 1999 e 2000 foram realizados aproximadamente 15 vôos, nos quais se inclui o primeiro vôo com controle de trajetória em 3 de março de 2000. Para o desenvolvimento e aprimoramento do controlador de trajetória no decorrer do ano de 2001, foram realizados 16 vôos com sucesso, dentre aproximadamente 30 tentativas, correspondendo a aproximadamente 120 horas de trabalho de campo, onde para cada vôo

aguardou-se a existência de condições meteorológicas adequadas para a coleta e análise de informações sobre o comportamento do dirigível e do seu sistema de controle – tipicamente previsão de ventos inferiores a 15 km/h – monitoradas contínua e automaticamente a partir de informações disponíveis na Internet a partir da CEPAGRI-UNICAMP (CEPAGRI, 2001), do CPTEC-INPE (CPTEC, 2001) e do Centro Integrado de Meteorologia e Recursos Hídricos de Santa Catarina (CLIMERH, 2001). A maioria dos vôos deu-se no início da manhã ou final da tarde (tipicamente entre 7:00 e 11:00, e entre 16:00 e 18:00 respectivamente), envolvendo preparativos prévios e posteriores ao vôo de aproximadamente 1 hora cada, e congregando grupos de três a cinco pessoas. Das aproximadamente 14 tentativas não efetivadas, algumas não ocorreram devido a mudanças bruscas nas condições meteorológicas, panes e outros fatores.

A realização de vôos é uma atividade contínua, necessária seja à evolução do entendimento sobre o comportamento do dirigível e sua modelagem, seja para se aprimorar o sistema de controle e navegação.

Do total de vôos efetuados, separou-se para apresentação e discussão dois conjuntos de resultados, um realizado no dia 7 de julho de 2001 e outro nos dias 4 e 5 de outubro de 2001. Em todos eles, a decolagem e aterrissagem do dirigível são efetuadas por controle remoto, pelo piloto em terra. Com o dirigível já estabilizado em vôo de cruzeiro, o piloto comanda a transição do modo manual para o modo automático de operação; a partir daí o dirigível evolui sob a ação do sistema de controle e navegação, permanecendo o piloto como supervisor do andamento do vôo e, posteriormente, reassumindo a pilotagem do veículo até a aterrissagem.

### **8.5.1 *Trajétoria de Referência***

Nos vôos reais, tendo o dirigível atingido uma condição estável de vôo de cruzeiro sob comando manual, o piloto aciona a transição para o modo automático de operação; as coordenadas do veículo neste instante constituem o ponto inicial (PI-1) do quadrilátero que configura a trajetória de referência, sendo a direção da primeira aresta dada por  $\psi_{ares}$ , que é o ângulo de guinada do veículo neste mesmo instante. Os demais pontos do quadrilátero (PF-1, PI-2, PF-2, PI-3, PF-3, PI-4 e PF-4) são definidos a partir do valor de  $\psi_{ares}$  e daquele especificado pelo usuário para o tamanho das arestas, de forma similar à descrita na Seção 8.3.1 e exemplificada na Figura 8.21. Estes pontos e as arestas entre eles configuram a

trajetória de referência para o sistema de controle de trajetória. As transições entre os segmentos de reta (arestas do quadrilátero) são efetuadas como se mostrou na Seção 8.3.1.

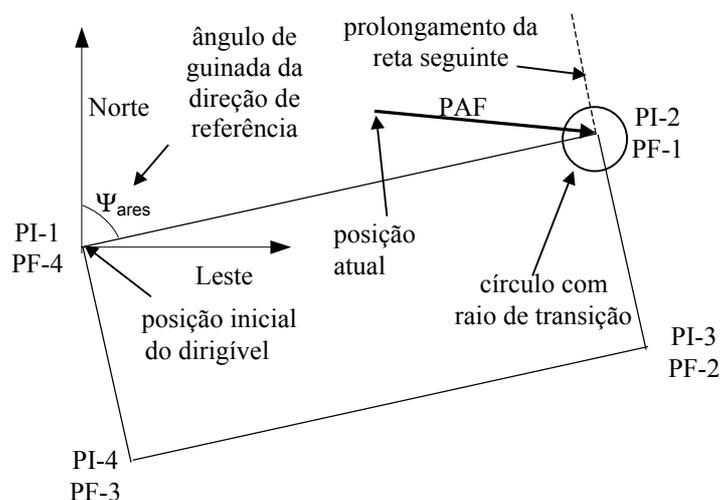


Figura 8.21: Retas de referência para a trajetória e critério para transição entre retas.

### 8.5.2 Vôo do dia 7 de Julho de 2001

Dos dados de telemetria do vôo do dia 7 de julho de 2001, são apresentados alguns resultados de um procedimento iterativo para ajustes do algoritmo de controle de trajetória, onde a trajetória de referência é idêntica em todos os casos, formada por um mesmo quadrilátero. Partindo-se de um conjunto inicial de parâmetros, variou-se alguns deles na busca do melhor comportamento do sistema. A Tabela 8.3 mostra o conjunto de valores de ganhos explorados nesses ensaios.

A Figura 8.22 mostra a trajetória executada (latitude e longitude) para alguns dos conjuntos de ganhos mostrados na Tabela 8.3. Alguns ensaios foram abortados já no seu início, como é o caso dos ensaios 1, 2 e 3, por ser visível a inadequação dos seus parâmetros.

Ganhos/Ensaio	1	2	3	4	5	6	7
$K_{PT}$	0,1	10	5	3	1	1	1
$K_{IT}$	1	1	0,5	0,5	0,1	0,05	0,1
$K_{VT}$	0	0	0	0	0	0	0
$K_{RT}$	1	10	5	3	3	3	1,5
$K_{RRT}$	0	0	0	0	0	0	0

Tabela 8.3: Conjunto de ganhos testados nos dia 7 de julho de 2001

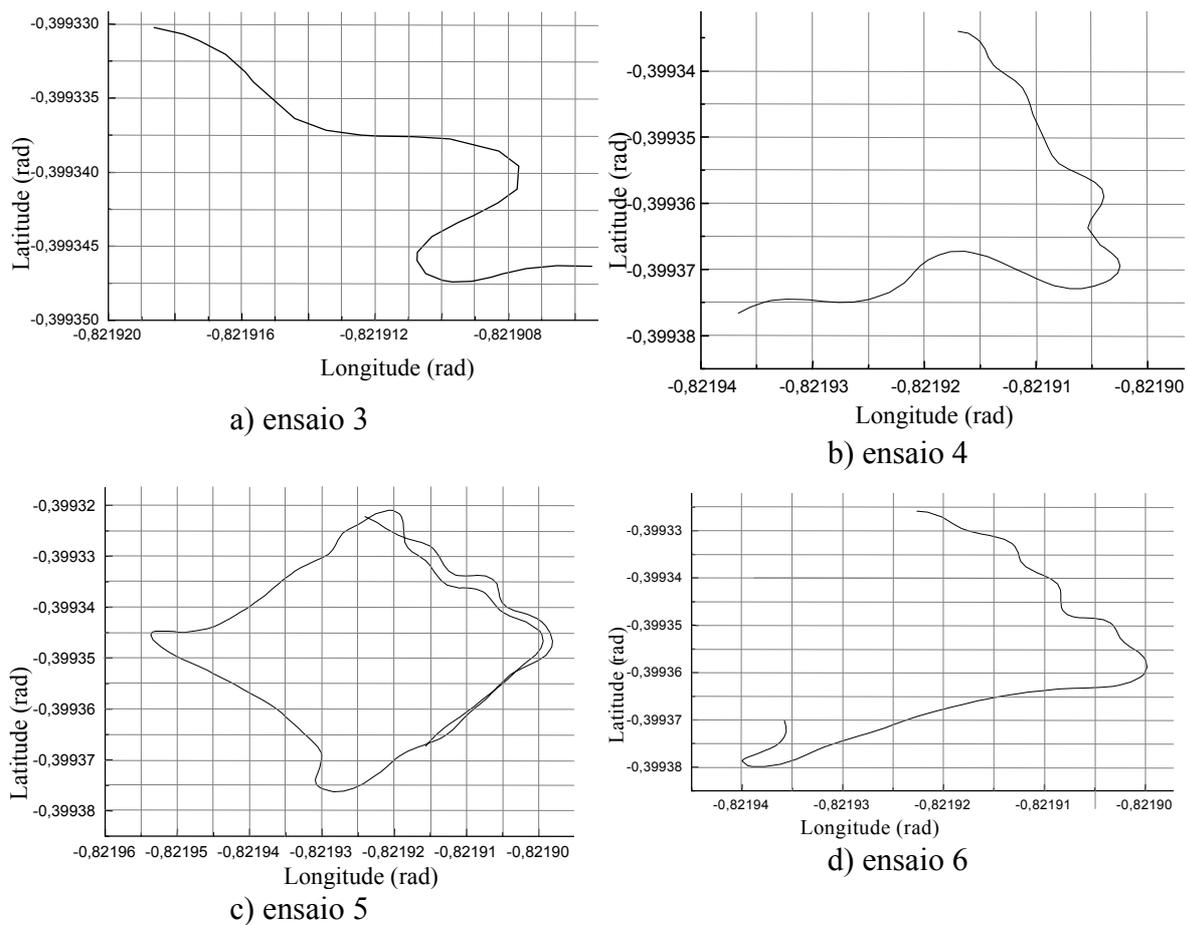


Figura 8.22: Passos para ajuste do controlador de trajetória.

A Figura 8.22-b apresenta um conjunto de ganhos que produz uma trajetória aceitável, mas ainda com alguma oscilação; a partir daí foi se variando os ganhos, principalmente o ganho de erro da trajetória “ $K_{PT}$ ” e do controlador de guinada “ $K_{RT}$ ”, até se obter o resultado mostrado na Figura 8.23, que foi considerado o melhor conjunto de valores, para os parâmetros de sintonia, correspondente ao ensaio 7 mostrado na Tabela 8.3.

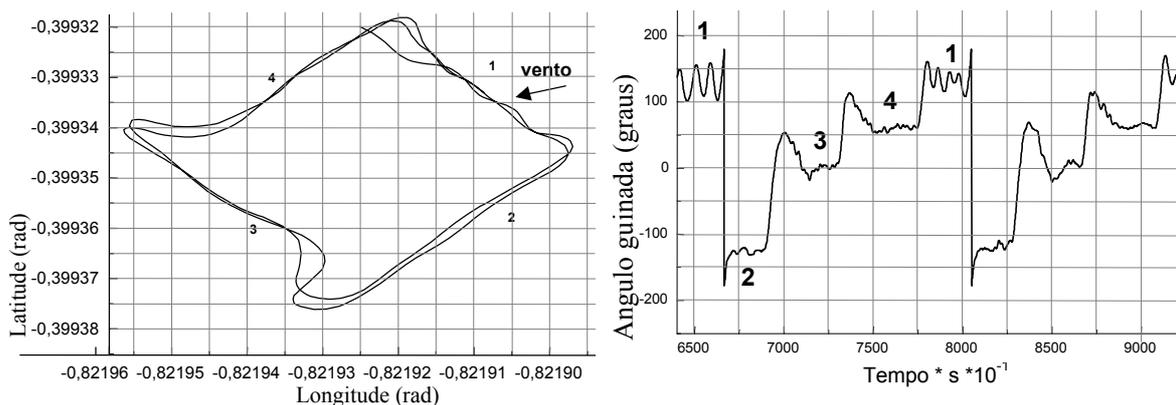


Figura 8.23: Melhor resultado dos ensaios do dia 7 de julho de 2001.  
Latitude e longitude e ângulo de guinada “ $\psi$ ”

Observando-se a Figura 8.23, onde se indica a direção do vento no momento do vôo, nota-se uma maior oscilação no trecho marcado por "1" que é a região mais sujeita a turbulências, por estar mais próxima de árvores. Nos trechos "2" e "3", há uma maior fuga quando da comutação da trajetória devido à presença de vento de popa. No trecho "4", a comutação da trajetória se dá de forma suave devido à presença de vento de proa.

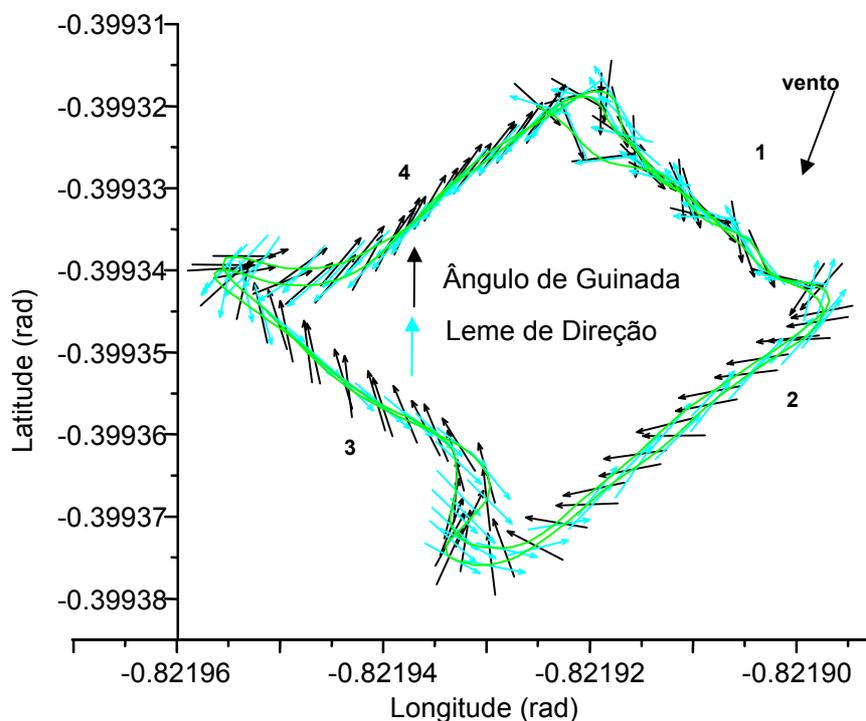


Figura 8.24: Melhor resultado de trajetória em 7 de julho.  
latitude, longitude, ângulo de guinada “ $\psi$ ” e leme de direção (ZETA) -“ $\zeta$ ”

A Figura 8.24 apresenta a trajetória juntamente, com vetores representando o ângulo de guinada (em tom escuro) e o comando do leme de direção (em tom claro). Pode-se observar que:

- os trechos executados com menor velocidade são os 1 e 4, onde se visualiza uma maior concentração de flechas indicando menor velocidade inercial e maior velocidade relativa ao ar devido a uma importante componente de vento de proa;
- os trechos de maior velocidade são os 2 e 3 (i.e. com menor concentração de setas), com maior velocidade inercial e menor velocidade relativa ao ar devido à preponderância de um vento de popa;
- nos trechos 2 e 3, o dirigível desloca-se apontando para dentro da trajetória, e no trecho 4 para fora dela, compensando a influência do vento proveniente do norte, ao passo que no trecho 1, devido à turbulência, ele oscila;
- no trecho 1 a orientação do dirigível é mais oscilante que nos outros, devido às turbulências geradas pelos obstáculos próximos, como árvores e construções.

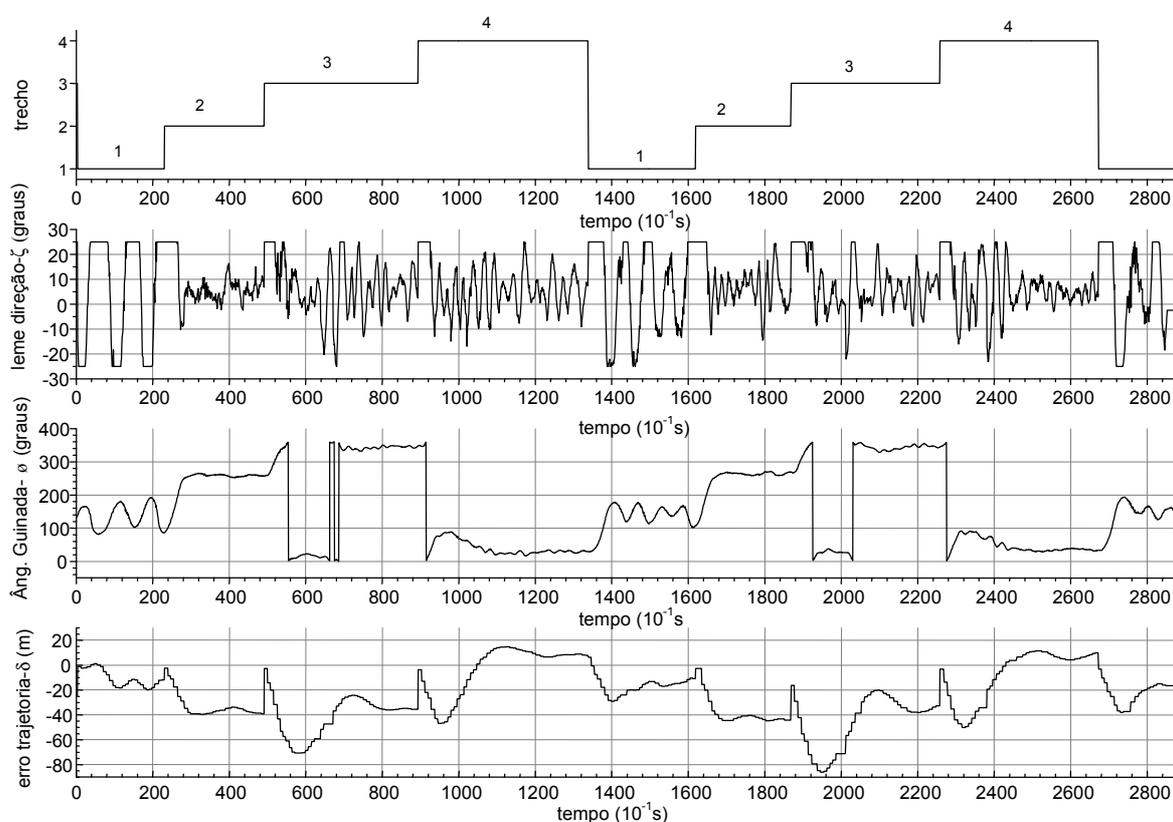


Figura 8.25: Erro de trajetória “ $\delta$ ”, ângulo de guinada “ $\psi$ ” e leme de direção “ $\zeta$ ” para melhor trajetória.

A Figura 8.25 apresenta o erro de trajetória “ $\delta$ ”, isto é, a distância entre a posição do dirigível e a reta de referência (vide Figura 4.3); o comando do leme de direção (ZETA) -“ $\zeta$ ” e o ângulo de guinada “ $\psi$ ”, além da indicação dos trechos da trajetória (Trecho) correspondente à numeração dos trechos da Figura 8.23.

Observa-se na Figura 8.25, como já tinha sido constatado anteriormente, que o trecho "1" é o que apresenta maiores oscilações nas variáveis devido à turbulência na perturbação pelo vento, enquanto que nos demais trechos as oscilações são menores. Neste trecho 1, inclusive, o comando do leme de direção opera praticamente entre limites de saturação, como se vê na Figura.

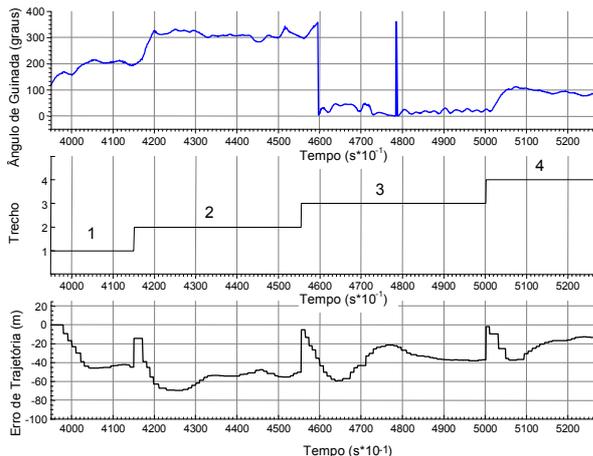
### 8.5.3 Vôos de 4 e 5 de Outubro de 2001

Esta seção apresenta os resultados dos vôos realizados nos dias 4 e 5 de Outubro de 2001. Nestes resultados é mostrada a influência da variação dos ganhos de  $K_{VT}$  que se manteve constante no primeiro exemplo. Foram realizadas variações em  $K_{PT}$  e em  $K_{VT}$  até o sistema ficar sub-amortecido, para então se incrementar finalmente o ganho integral ( $K_{IT}$ ). A Tabela 8.4 mostra o conjunto de valores de ganhos explorados nesses ensaios. Adicionalmente numa avaliação preliminar dos resultados do dia 4 de outubro utilizou-se  $K_{RRT}$  diferente de zero para se incrementar o amortecimento na malha do ângulo de guinada. A seguir são apresentados os gráficos e comentados os resultados.

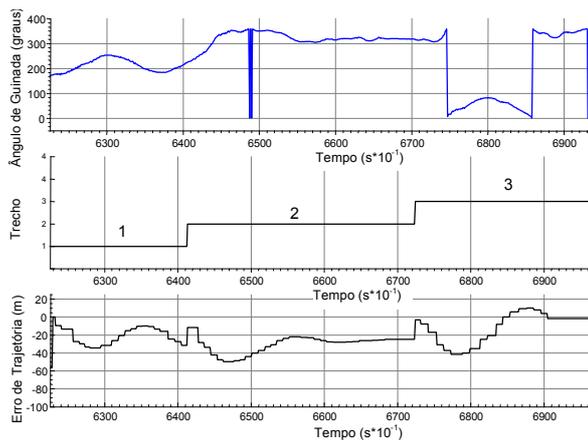
Ensaio	Vôo de 4 Outubro			Vôo de 5 Outubro							
	1	2	3	1	2	3	4	5	6	7	8
$K_{PT}$	1,03	1,2	2,2	4,2	4,4	2,2	2,3	1,6	3,1	2,4	2,7
$K_{IT}$	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,3	0,4
$K_{VT}$	0,3	2	4	8	16	4	6	3	7,5	4,5	4,5
$K_{RT}$	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,5
$K_{RRT}$	0	0	0	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,5

Tabela 8.4: Conjunto de ganhos testados nos dia 4 e 5 de outubro de 2001.

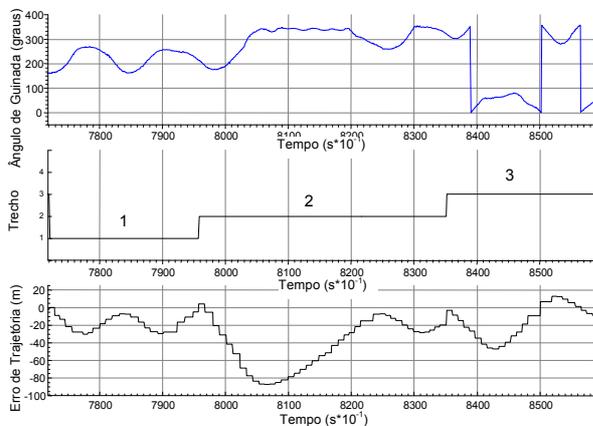
A Figura 8.26 e a Figura 8.27 mostram respectivamente os resultados do conjunto de ganhos da Tabela 8.4 correspondentes aos vôos dos dia 4 e 5 de Outubro de 2001. Elas apresentam o erro de trajetória “ $\delta$ ”, ângulo de guinada “ $\psi$ ” e os trechos de vôo (Trecho) em função do tempo segundo a convenção mostrada na Figura 8.24 e na Figura 8.28.



a) Ensaio 1

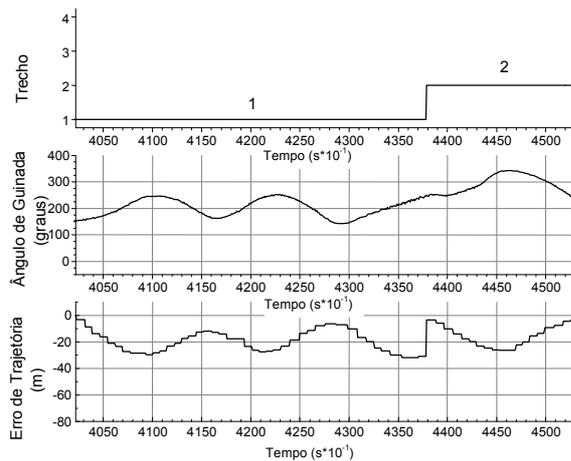


b) Ensaio 2

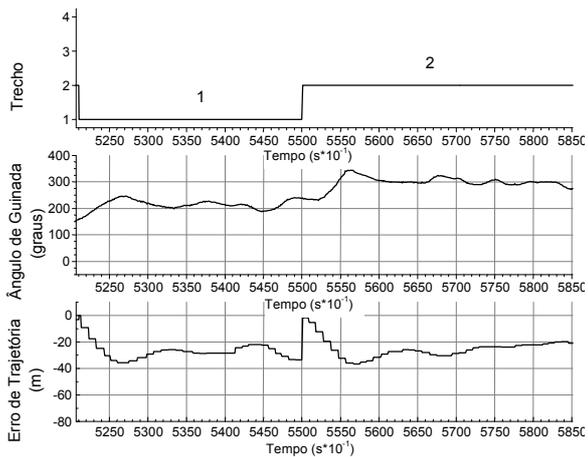


b) Ensaio 3

Figura 8.26: Resultados obtidos nos vôos de 4 de Outubro.

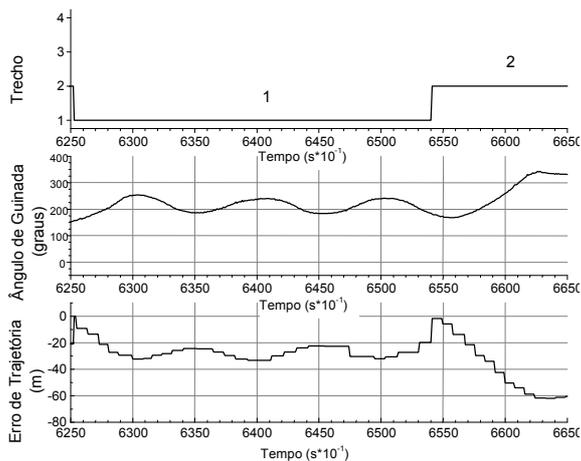


a) Ensaio 1

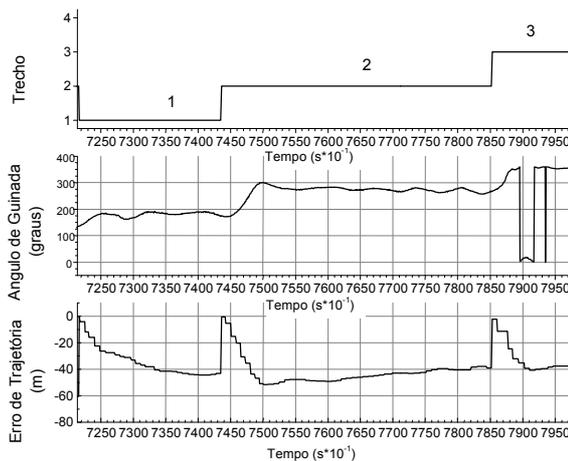


b) Ensaio 2

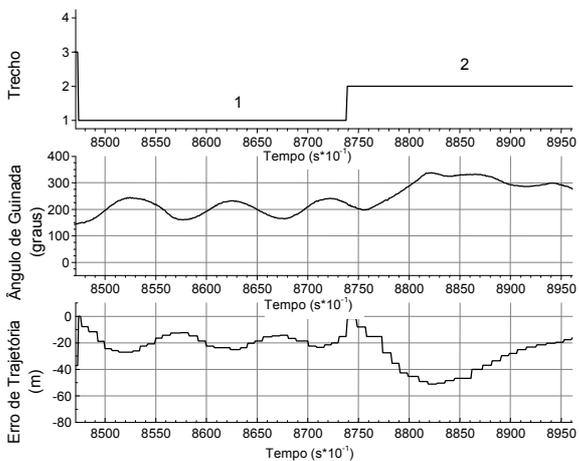
Figura 8.27: Resultados obtidos nos ensaios de 5 de Outubro de 2001.



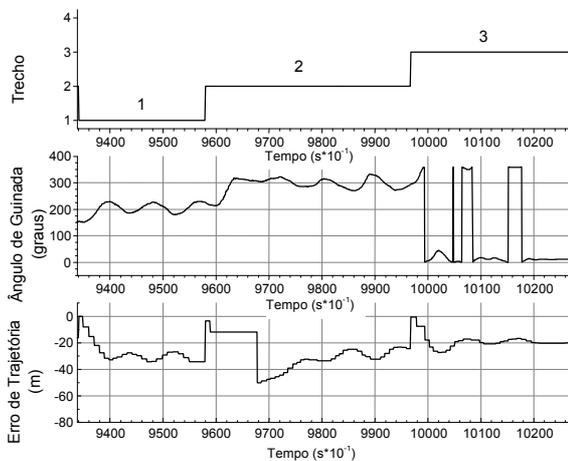
c) Ensaio 3



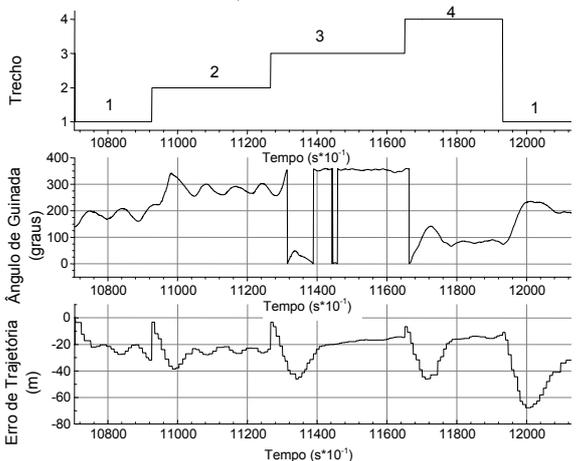
d) Ensaio 4



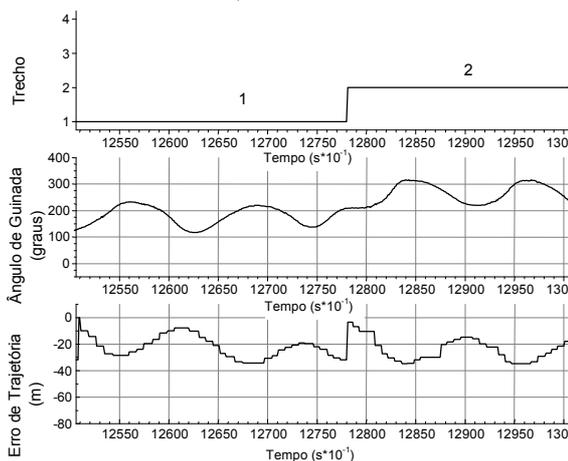
e) Ensaio 5



f) Ensaio 6



g) Ensaio 7



h) Ensaio 8

Figura 8.27(cont.): Resultados obtidos nos ensaios de 5 de Outubro de 2001. erro de trajetória “ $\delta$ ” e ângulo de guinada “ $\psi$ ”

Dentre os conjuntos de ganhos mostrados na Tabela 8.4, o que levou ao melhor resultado foi o utilizado no ensaio 7 do dia 5 de outubro, produzindo um erro médio de trajetória de -24,21m e desvio padrão de 11,33 m numa aresta de 200m. Comparando-se com os dados de vôo do dia 7 de Julho, quando se obteve um erro médio de trajetória similar (-23,64m) com desvio padrão duas vezes maior (22,43m), pode-se afirmar que o melhor ajuste dos ganhos derivativos e integral no ensaio 7 levou a um desempenho superior para o sistema de controle de trajetória. A Figura 8.28 mostra o gráfico latitude-longitude para esta situação.

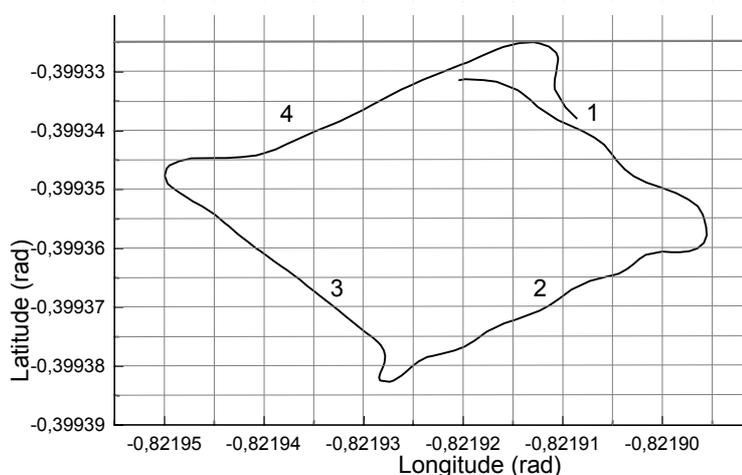


Figura 8.28: Melhor trajetória, correspondente ao Ensaio 7.

## 8.6 Controle de Altitude: Resultados em Vôo

De forma similar à descrita na seção precedente, diversos vôos vêm sendo realizados, com armazenagem dos dados de telemetria gerados a bordo do dirigível, para se avaliar especificamente, em condições práticas reais, o controle de altitude e seu funcionamento em conjunto com o controle de trajetória. Nos ensaios, a decolagem e a aterrissagem do dirigível são comandadas pelo piloto em terra; quando o dirigível se encontra em vôo de cruzeiro, o piloto comanda a transição do modo manual para o modo automático de operação, ativando o controle de trajetória, o controle de altitude, ou ambos. Nota-se que procedimentos automáticos de decolagem, tratados anteriormente em simulação, na Seção 8.4, não são contemplados nos ensaios de vôo.

Dentre os vôos realizados, esta seção apresenta os dois melhores resultados obtidos, relativos a vôos realizados em 24 e 30 de Julho de 2001. Ressalta-se que o ajuste do controle de altitude e de sua atuação com o controle de trajetória, para situações reais de

vôo, encontra-se em desenvolvimento, de maneira que é possível aprimorar os resultados iniciais aqui apresentados.

### 8.6.1 Vôos de 24 e 30 de Julho de 2001

Foram realizados nesses dias vários ensaios do controlador de altitude juntamente com o controlador de trajetória. Nestes ensaios foi avaliada a utilização de ganhos elevados para a malha externa e para a malha interna de controle de altitude, visando comparar com resultados prévios. Os parâmetros do algoritmo de controle de trajetória são similares aos utilizado no vôo de 7 de Julho, descrito na seção precedente; os parâmetros dos controladores de trajetória e de altitude são apresentados na Tabela 8.5 (vide a Tabela 8.1 para a descrição destes parâmetros).

Vôos de 24 de julho e 30 de Julho de 2001				
Ganhos/Ensaio	1	2	3	4
$K_{PT}$	1,5	1,5	1,5	1
$K_{IT}$	0,1	0,1	0,1	0.2
$K_{VT}$	0	0	0	0
$K_{RT}$	1,5	1,5	1,5	1,5
$K_{RRT}$	0	0	0	0
$K_{EH}$	5	5	10	5
$K_{VH}$	1	1	1	1
$K_{AH}$	5	5	5	5
$K_{RAH}$	5	10	10	10

Tabela 8.5: Parâmetros utilizados no ensaio de controle simultâneo de altitude e trajetória.

A Figura 8.29 apresenta a trajetória latitude-longitude executada no primeiro ensaio da Tabela 8.5.

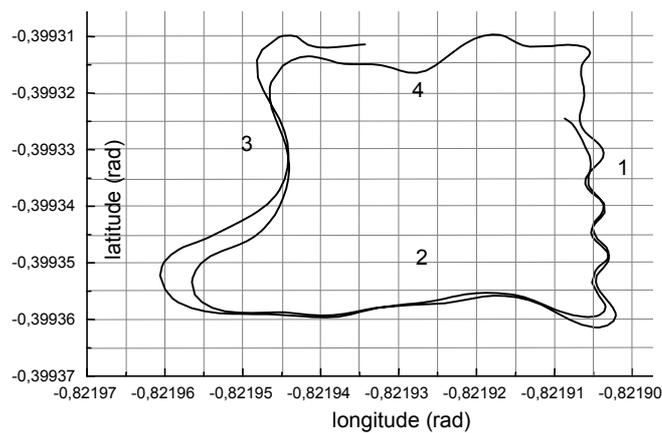
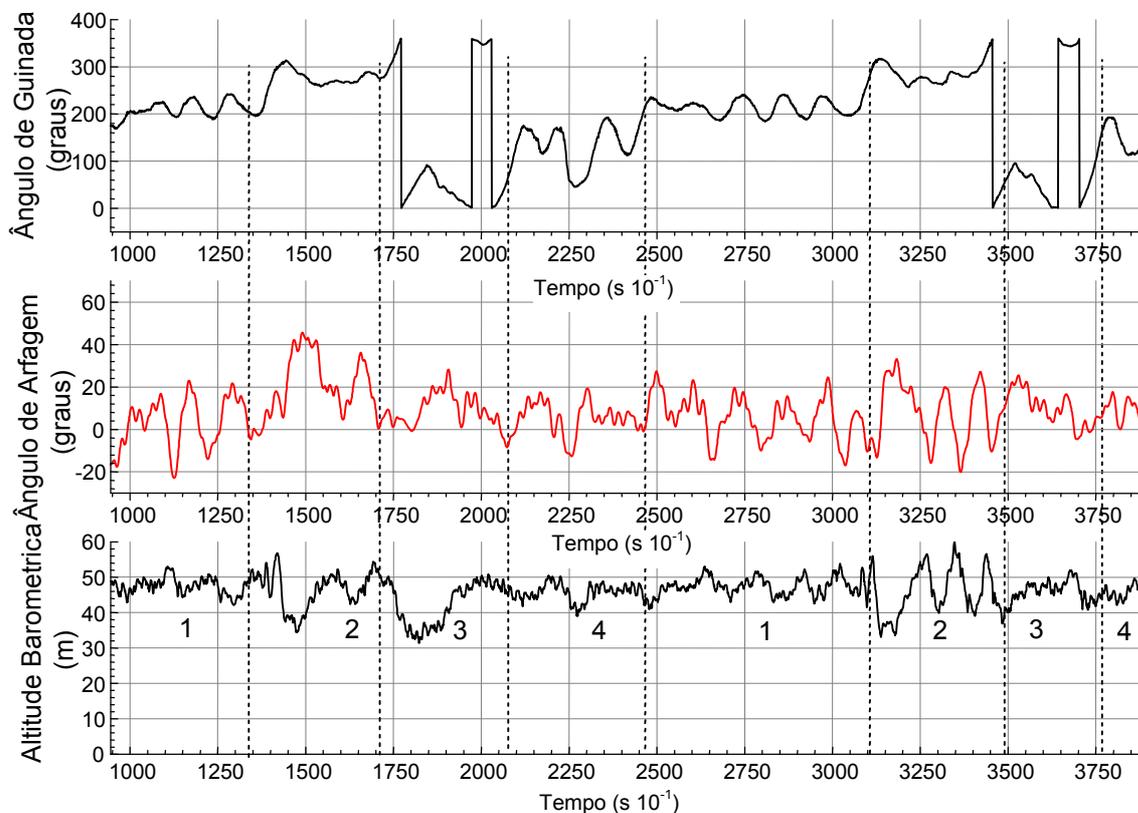


Figura 8.29: Trajetória resultante de controle simultâneo trajetória-altitude (Ensaio 1).



altitude barométrica, ângulo de arfagem “ $\theta$ ” e ângulo de guinada “ $\psi$ ”

Figura 8.30: Variáveis do voo de 24/7/2001.

A Figura 8.30 mostra a altitude barométrica o ângulo de arfagem “ $\theta$ ”, e o ângulo de guinada “ $\psi$ ”. O instante de comutação para controle automático foi  $t = 94,5s$ ; neste instante, a altitude em relação ao ponto de decolagem (48 m) foi considerada como referência para o controlador de altitude. A resposta em altitude corresponde a duas voltas de uma trajetória segundo um quadrilátero com aresta de 200m conforme mostrado na Figura 8.29; os instantes de comutação de trajetória entre as arestas sucessivas são

indicados pelas marcas verticais na figura (tracejadas), sendo que os números correspondem às arestas como indicado na Figura 8.29.

Observa-se na Figura 8.30 que o desvio máximo de altitude oscila entre -16m e +12m, coincidindo com mudanças no ângulo de guinada, correspondendo justamente ao momento da transição abrupta de trajetória do trecho “1” para “2” e depois de “3” para “4”, repetindo-se na volta seguinte, como já se observou em outras situações. Nos trechos onde não há esta transição, a variação de altitude se situa na faixa de 6 metros, mas pode-se observar que as variações em altitude estão em fase com o ângulo de guinada, indicando a influência do controlador de trajetória no controlador de altitude ou vice-versa.

A seguir apresenta-se os resultados do vôo realizado no dia 30 de Julho de 2001 correspondente ao ensaio 4 da Tabela 8.5, sendo que neste ensaio incrementou-se ainda mais o ganho " $K_{RAH}$ " da razão de arfagem em relação ao mostrado previamente. A Figura 8.31 reproduz a trajetória realizada neste vôo.

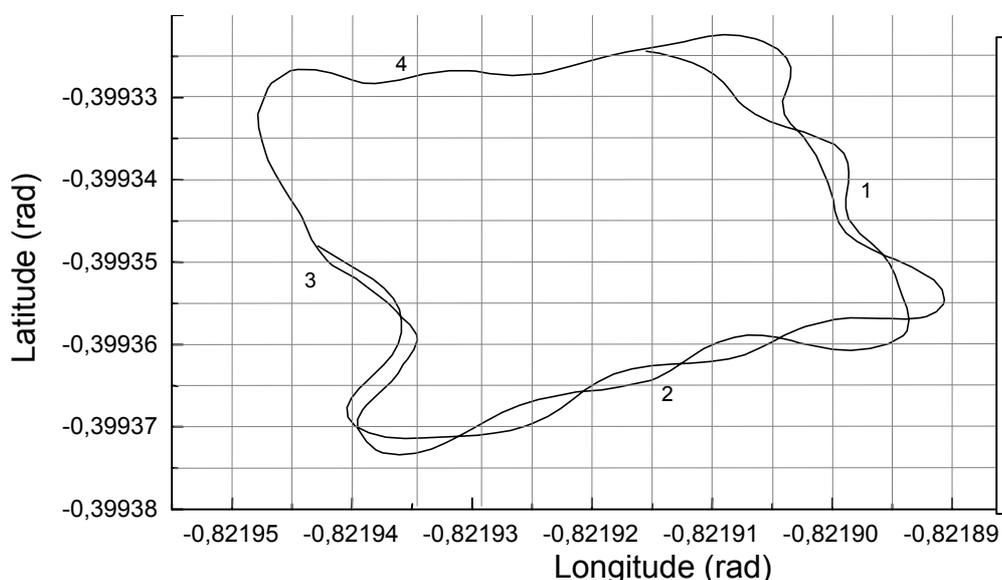


Figura 8.31: Trajetória resultante de controle simultâneo trajetória-altitude (ensaio 4)

A Figura 8.32 apresenta a os comandos dos lemes de direção e profundidade, altitude barométrica, o erro de trajetória “ $\delta$ ”, o trecho da trajetória sendo executado (Trecho "1", "2", "3" ou "4"). Na parte superior desta mesma figura, aparecem os comandos dos lemes de direção e profundidade (ETA e ZETA); eles serão detalhados mais à frente pois na escala da figura esses dados estão muito pouco visíveis, devido aos ganhos elevados da malha interna de arfagem (ZETA), que oscila entre os valores máximo e mínimo.

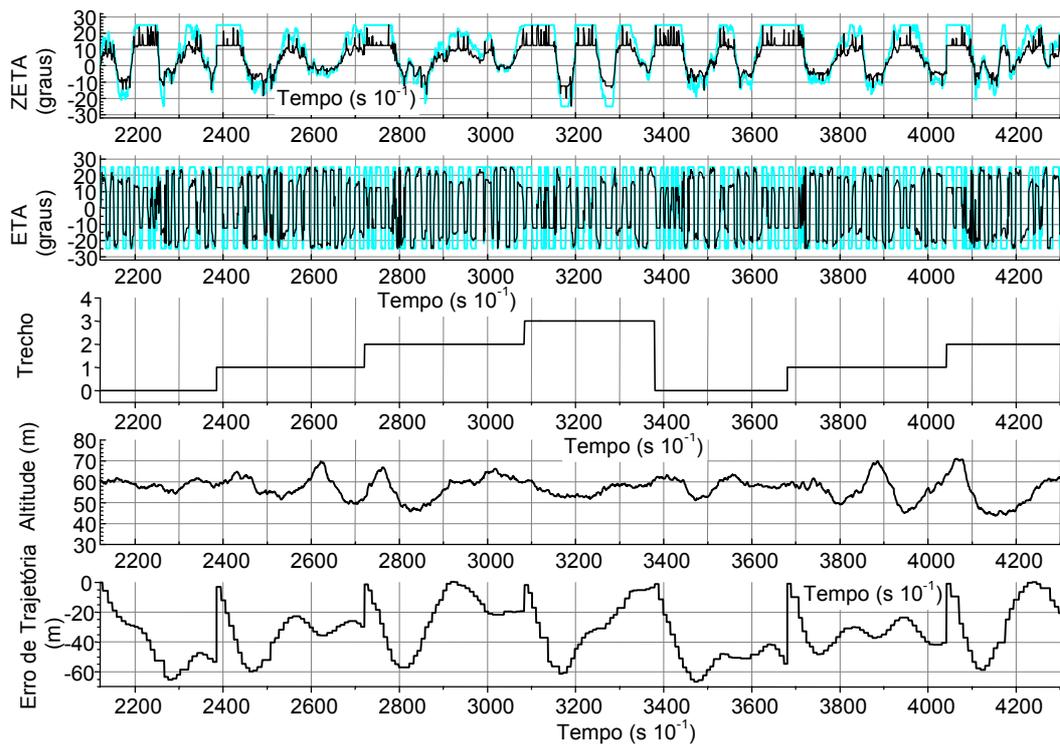


Figura 8.32: Vista geral do comportamento em trajetória e altitude.

Pode-se observar da Figura 8.33 que é visível a existência de relação entre o erro de trajetória e a transição abrupta de trajetória, gerada pela mudança das retas de referência, ocasionando, na configuração parâmetros dos controladores utilizada, a saturação do leme de direção. Além disso nota-se também o efeito por acoplamento no leme de direção (ZETA) pelo comando do leme de profundidade (ETA)), que está oscilando entre os valores máximo e mínimo de saturação, devido a ganhos evidentemente elevados da malha de arfagem, causado pela conversão de "+" para "x" como se mostrou na Seção 8.4.5. A Figura 8.33 apresenta uma visão detalhada da figura precedente. É possível visualizar os comandos do leme de direção: " $\zeta$ " (ZETA) (em tom mais claro) antes da conversão de "+" para "x" – ver Equações 8.1 e 8.2 – e os valores resultantes efetivamente aplicados depois da saturação introduzida pelas superfícies de atuação, obtidos da reconversão de "x" para "+" (em tom mais escuro). Da mesma forma aparecem os comandos leme de profundidade antes " $\eta$ " (ETA) em tom mais claro e depois da conversão (em tom mais escuro).

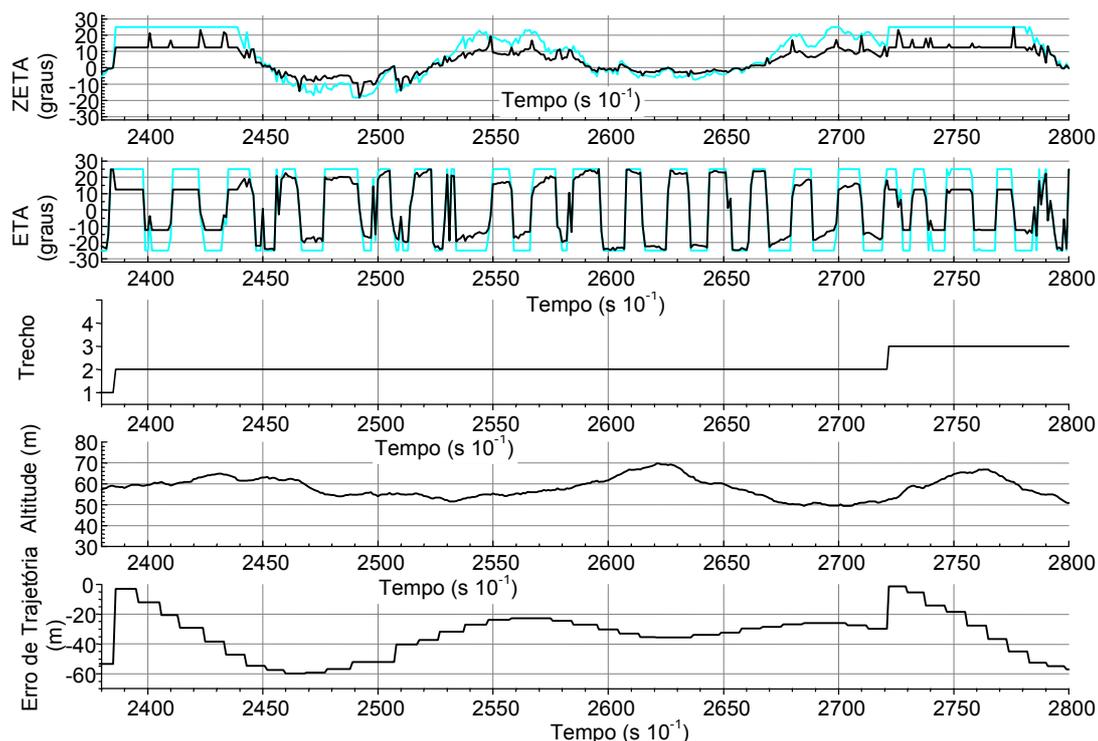


Figura 8.33: Vista detalhada do comportamento em trajetória e altitude.

Observa-se nesta figura que o comando de leme de profundidade (ETA) está oscilando entre os valores máximo e mínimo de saturação, devido a ganhos evidentemente elevados da malha de arfagem. O efeito desta oscilação é a modulação do comando de guinada com esta frequência, provocando um resultado efetivo de um ganho menor em trajetória, principalmente em regiões onde se satura o comando de guinada como na transição de trajetórias, prejudicando o desempenho do sistema no seguimento de trajetória, da mesma forma que se observou na Seção 8.4.5.

Os resultados apresentados em controle simultâneo de altitude e trajetória são os primeiros obtidos, pois o processo de ajuste ainda se encontra em andamento. Entretanto, estes resultados obtidos permitiram a validação do sistema de controle e navegação embarcados.

## 8.7 Conclusões

Características de operação de dirigíveis apresentadas na Seção 3.2, ensaios com modelos e dados de literatura como as citadas no Capítulo 3, e a prática de vôo com o dirigível AS800, permitem concluir que:

- O dirigível é um sistema multivariável, com acoplamento entre as dinâmicas lateral e longitudinal;
- Constatou-se em simulação e na prática que os atuadores possuem características de resposta variáveis em função da velocidade e situação de voo: as superfícies aerodinâmicas perdem a sua eficácia a velocidades muito baixas, condição em que a propulsão e sua vetorização (e acionamento diferencial dos motores ou um motor de cauda perpendicular ao eixo do veículo, quando estas soluções existem) constituem a principal forma de atuação, e vice-versa para velocidades de intermediárias a altas; isso leva à necessidade de uma variação contínua na configuração variáveis controladas  $\Leftrightarrow$  variáveis de atuação segundo as diferentes situações de voo - decolagem, voo de cruzeiro, voo pairado, aterrissagem vertical ou horizontal.
- O Comportamento dinâmico é influenciado pelo grau de fluvariabilidade do veículo – desde neutro até diferentes níveis de voo pesado; mesmo a partir de um determinado peso como condição inicial, a variação da fluvariabilidade no decorrer do voo se dá pela redução do volume de combustível ou da alteração do volume de hélio devido à superaquecimento ou altitude.

Assim os dirigíveis apresentam características peculiares que tornam complexo o desenvolvimento de estratégias de controle e navegação automáticas, tais como:

- A dinâmica não linear;
- Modos mal amortecidos, resultando oscilações laterais e longitudinais importantes para manobras de subida / descida e de guinada;
- O dirigível possui sensibilidade ao vento, cuja orientação e intensidade constituem a mais importante perturbação.

Neste cenário, este capítulo apresentou o diagrama geral dos algoritmos de controle de trajetória para o dirigível AS800, mostrando o mapeamento da infra-estrutura sensorial instalada a bordo do veículo com o diagrama geral de controle.

Foram então apresentados estudos em simulação do algoritmo de controle de trajetória e a sintonia de seus parâmetros, mostrando os principais elementos que

influenciam no seu comportamento, principalmente o vento e a transição abrupta entre segmentos de reta de trechos de trajetória sucessivos.

Posteriormente, os estudos em simulação voltaram-se ao algoritmo de controle de altitude, abordando inclusive a sua interação com o controle de trajetória, mostrando também os principais elementos que os influenciam, como o vento e a transição abrupta de trajetória, e a degradação no desempenho devido à saturação quando da conversão da atuação aerodinâmica de “+” para “x”

Na seqüência, calcado nas informações obtidas em simulação, vários vôos reais do dirigível foram apresentados, abrangendo a sintonia dos ganhos dos controladores e a avaliação do sistema de controle e navegação em seguimento de trajetória e perfil de altitude em situações reais de utilização.

Devido à introdução no dirigível de capacidades para a realização de vôos com controle simultâneo de altitude e de trajetória – um dos principais requisitos para o dirigível robótico e objetivo deste trabalho de tese –, tornou-se possível obter uma fotografia, mostrada na Figura 8.34, na qual o piloto observa o desenrolar de um vôo autônomo do dirigível.



Figura 8.34: Piloto Bruno G. Faria e dirigível ao fundo em vôo automático.



## 9. PROTÓTIPO DA ARQUITETURA DE *SOFTWARE* ROBÓTICO PARA O DIRIGÍVEL

### 9.1 Introdução

No Capítulo 5 introduziu-se o conceito de *arquitetura de software* robótico (ASR) que descreve a forma em que eventos e decisões são tratados pelo sistema robótico no decorrer de sua operação. Citou-se também a tendência ao uso de arquiteturas híbridas, que contemplam tanto a deliberatividade, isto é realização de tarefas programadas de forma explícita, quanto a reatividade, ou seja, a resposta a contingências, como eventos inesperados acontecendo no próprio robô ou no seu ambiente de operação. Nas situações reais, este nível de controle é, portanto, complexo, pois deve ao mesmo tempo atingir metas definidas em um plano de ação e também reagir face as situações imprevistas, compreendendo portanto, a coordenação de atividades concorrentes, realizando o seqüenciamento e a sincronização de tarefas e tratando as eventuais contingências. Nesse contexto, explicitou-se na Seção 2.2.3 que uma arquitetura de *software* robótico de tipo híbrido constitui um dos recursos de suporte para o desenvolvimento de um dirigível autônomo. Comentou-se também que, dada à complexidade acima citada, é pertinente o uso de infra-estruturas que facilitem o desenvolvimento desta arquitetura.

Este capítulo tem por objetivo apresentar o ensaio em simulação da arquitetura de *software* robótico concebida para o dirigível, usando como ferramenta o protótipo do Ambiente de Desenvolvimento e Operação do dirigível, descrito na Seção 7.5. A ASR do dirigível é construída sobre a linguagem a TDL, já apresentada no Capítulo 5.

Este capítulo está assim organizado: na Seção 9.2 apresenta-se a instanciação do modelo conceitual da Arquitetura proposta para o dirigível (mostrada no Capítulo 5) no ambiente de desenvolvimento e operação do dirigível (apresentado no Capítulo 7). A Seção 9.3 aborda aspectos voltados à implementação, tais como os aplicativos escritos em TDL, aspectos de tempo real e de comunicação e um diagrama em blocos completo do *software* robótico do dirigível; na Seção 9.4 mostram-se as capacidades de deliberação e reação da ASR do dirigível através de estudos de caso simulados e, finalizando, na Seção 9.5 são apresentadas as conclusões.

## 9.2 Instanciação da Arquitetura de *Software* Robótico no Ambiente de Desenvolvimento e Operação do Dirigível

Na Figura 7.3 (Seção 7.4) apresentou-se a estrutura do Ambiente de Desenvolvimento e Operação do dirigível, onde os componentes podem estar localizados em processadores diferentes comunicando-se via TCP/IP. A Figura 9.1, a seguir, mostra a instanciação do protótipo da ASR do dirigível, utilizando o conceito de multi-agentes, para os componentes do Ambiente de Desenvolvimento e Operação apresentado no Capítulo 7.

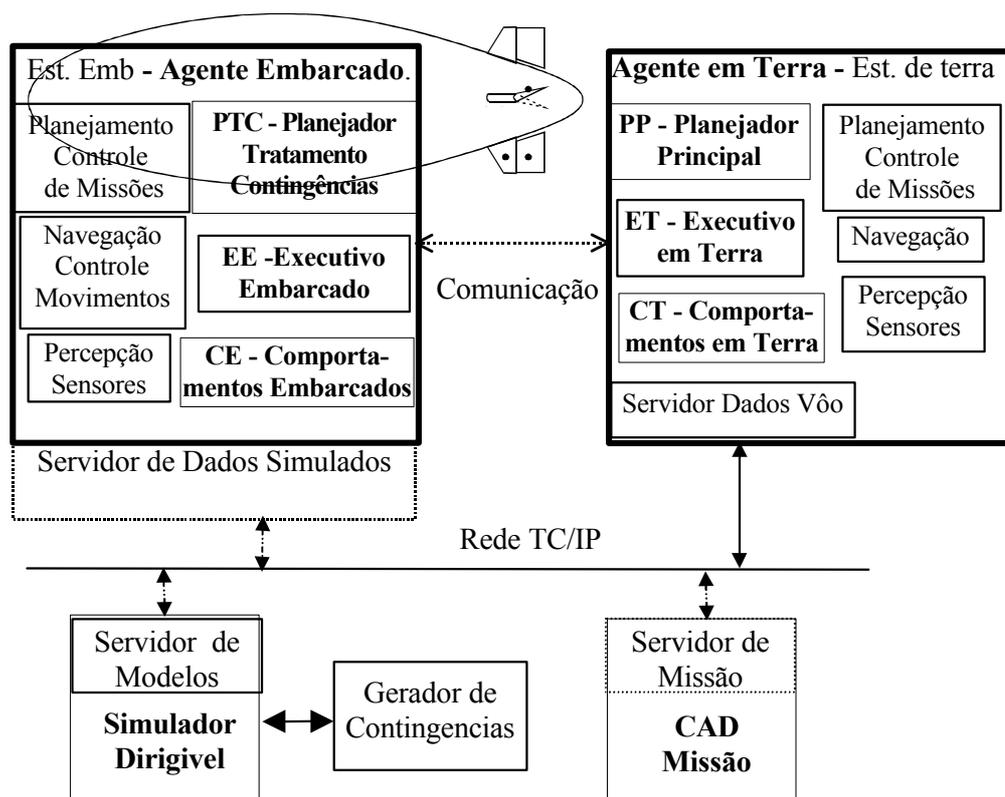


Figura 9.1: Ambiente de *software* utilizado para o desenvolvimento da ASR.

Observa-se na Figura 9.1 que o Agente Embarcado (AE) está associado à Estação Embarcada e o Agente em Terra (AT) está associado à Estação de Terra, sendo que a estrutura de hardware destas estações está descrita no Anexo I.

No caso do AE, o mapeamento dos seus componentes para os componentes da Estação Embarcada se dá como segue:

i) o PTC executa as funções de planejamento de missão enquanto que o EE executa a função de controle de missão;

ii) a navegação e controle de movimentos se dá em parte no EE e em parte nos CE - responsáveis pelos controles dos movimentos em interação com os sensores e do sistema de percepção;

Com relação ao sistema de percepção, vale ressaltar que os sensores e atuadores não aparecem explicitamente nas arquiteturas em três camadas, mas são utilizados pelos comportamentos para realizarem sua ação. O mapeamento do AT para a Estação de Terra ocorre de forma análoga, com planejamento e controle de missões, sendo a parte de planejamento executada no PP e a de controle da missão executada no ET; o tratamento dos sensores em terra e da interface homem-máquina são realizado por comportamentos em terra – CT.

Observa-se também, na Figura 9.1, que o bloco CAD de missão (projeto de missão assistido por computador) está construído com linhas pontilhadas, significando que o desenvolvimento deste componente não foi contemplado neste protótipo da ASR, sendo a concepção da missão realizada pelo operador. Também aparece o Gerador de Contingências associado ao simulador do dirigível, que é fundamental para se estudar o comportamento do sistema frente a eventos. Este gerador permite a introdução de perturbações tais como mudanças de intensidade e direção de vento, falhas em motores, etc.

### **9.3 Implementação da Arquitetura de Software Robótico**

O estabelecimento da base de uma ASR para o dirigível envolve a integração de aplicativos escritos em TDL ao ambiente de *software* do dirigível, (vide Capítulo 5). Neste contexto, nesta seção apresenta-se inicialmente a estrutura dos componentes executivos embarcados e em terra (EE e ET), escritos em TDL; na seqüência discutem-se as questões de tempo real associadas à solução desenvolvida e, finalmente, apresenta-se o diagrama em blocos completo da estrutura e comenta-se sobre a comunicação entre os aplicativos escritos em TDL e os demais aplicativos no *software* do dirigível.

#### **9.3.1 Estrutura dos Executivos Escritos em TDL**

O objetivo desta seção é mostrar, de uma forma simplificada, como se implementou a primeira versão dos executivos da arquitetura de software robótico, tanto da estação

embarcada quanto da de terra, utilizando os recursos dos *monitors* e *exception handlers* da TDL anteriormente apresentados.

A função dos *monitors* é realizar a verificação periódica de estados do dirigível, ou de quaisquer variáveis de interesse no cenário de vôo, e, em caso de ocorrência de alguma contingência, tomar alguma providência através da ativação de um *exception handler*. Podem ser definidos tantos *monitors* quantas forem as variáveis de interesse.

O código simplificado que implementa um executivo é ilustrado na Figura 9.2, tendo sido colocados números de linha em itálico para maior clareza. São declarados: i) um *exception handler* “*airshipproblem*”, iniciando na linha 5 e terminando na linha 9; ii) um *monitor* periódico chamado “*watchfailure*” iniciando na linha 11 e encerrando na linha 19. Na linha 21, a cláusula *spawn watchfailure* dispara a execução do *monitor*. O monitor examina periodicamente variáveis de interesse e se, como escrito por exemplo na linha 14, a variável “*estadox*” não for “*true*”, indicando que uma falha aconteceu, dispara na linha 17 o *exception handler* “*airshipproblem*”, através do comando TDL *fail*. O *exception handler* “*airshipproblem*” termina a tarefa atual na linha 6, trata o problema na linha 7 e ativa um outro monitor, “*thecontrolmonitor*”, que, no caso, controla o movimento do dirigível, na linha 8.

```

1 exception airshipproblem ();
2 exception handler catchproblems (tcm_task_tree_ref thecontrolmonitor)
4 handles airshipproblem
5 {
6   currenttask terminate;
7   treat_problem(problem, activation_parameters)
8   thecontrolmonitor activate ;
9 }
10
11 monitor watchfailure ()
12 with serial, period 0:0:1.0, maximum trigger 1
13 {
14   if(estadox != true)
15     {
16       trigger ();
17       fail airshipproblem (problem);
18     }
19 }
//-----
21 spawn watchfailure

```

Figura 9.2: Código simplificado do *software* de um Executivo

### 9.3.2 Aspectos de Tempo Real

Um dos aspectos considerados na integração de módulos em TDL ao *software* do dirigível (no caso, ao seu Ambiente de Desenvolvimento e Operação) é o da garantia de que o *software* desenvolvido em TDL não comprometa as questões de tempo real do sistema de controle e navegação do dirigível, já que a versão da TDL atualmente disponível opera no nível usuário do Unix, isto é, como todo o sistema Linux, não apresenta características *hard real time*.

A pergunta a ser respondida na análise deste requisito requer a avaliação da necessidade do requisito *hard real time* para os aplicativos escritos em TDL e a verificação dos mecanismos que poderiam ser utilizados para prevenir eventuais problemas que pudessem vir a ser causados pela ausência da característica tempo real na linguagem.

A avaliação das circunstâncias que levam a aplicativos não-*real-time* a apresentarem problemas de tempo de resposta, ou não atenderem a sinais no exato momento de sua ocorrência, mostra que as causas podem estar relacionadas à taxa de ocupação do processador no momento da ocorrência de eventos ou à simultaneidade de eventos. O comparativo feito em LINUXCP (2001) mostra a capacidade de resposta a um evento externo de um aplicativo escrito em RTLinux comparado com um aplicativo escrito em Linux, em função da taxa de ocupação do sistema operacional. O resultado é que, ao contrário do que ocorre com um aplicativo escrito em RTLinux, o atraso de resposta do aplicativo Linux cresce com a taxa de ocupação deste. Como a responsabilidade por responder e atender a eventos externos, no caso do ambiente do dirigível, é do RTLinux, a resposta a estes eventos não será prejudicada com a introdução dos aplicativos escritos em TDL.

Resta, enfim, a questão do atraso da resposta dos aplicativos escritos em TDL aos sinais originados do RTLinux, o que depende da taxa de ocupação do sistema computacional. A solução para reduzir este risco é aumentar a prioridade do aplicativo escrito em TDL em relação à de outros aplicativos ou, então, verificar qual a taxa de ocupação do sistema computacional quando da incorporação da TDL. Constatou-se que a taxa de ocupação do sistema computacional é baixa, pois o intervalo de amostragem para o dirigível - 100 ms - pode ser considerado grande ao se levar em conta tanto a velocidade

dos processadores utilizados atualmente no *hardware* embarcado e de terra e ao *software* neles executados, quanto a dinâmica (lenta) do próprio veículo.

Levando-se em conta estes aspectos, considera-se que é possível assegurar a integridade de uma aplicação de supervisão TDL, executada em Linux a nível de usuário, mostrando-se a solução utilizando TDL adequada para a implementação efetiva da arquitetura de *software* robótica do dirigível.

### **9.3.3 Diagrama Completo da Implementação da ASR**

A Figura 9.3 apresenta o diagrama em blocos completo do protótipo de *software* robótico do dirigível resultante da integração da ASR à estrutura do Ambiente de Desenvolvimento e Operação original. Comparando este diagrama ao apresentado na Figura 7.5 da Seção 7.5, constata-se que, para compor o nível executivo, foram acrescentados ao Ambiente de Desenvolvimento e Operação dois elementos: um embarcado (EE) e outro em terra (ET), escritos em TDL, que interagem com aplicativos executados em RTLinux (MKE no sistema embarcado ou MKT em terra), detalhados nas Seções 7.5.3 e 7.5.4. A função destes executivos é coordenar a operação do dirigível, exercendo o papel de integração entre os níveis de planejamento em terra e embarcado (PP e PTC respectivamente) escritos em linguagem convencional, e os comportamentos do dirigível em terra e embarcados (CT e CE respectivamente) escritos em RTLinux.

A Figura 9.3 ilustra também os aspectos de comunicação entre os diferentes módulos. Como MKE e MKT são módulos de *Kernel* e EE e ET são programas executados a nível de usuário, utiliza-se RTFifos e memória comum para realizar as interações com o EE ou o ET, pelos motivos já apresentados no Capítulo 6.

A memória comum é utilizada para passar os estados do dirigível, medidos pelos seus sensores, enquanto que as RTFifos são utilizadas para o envio de comandos ou sinalizações de eventos.

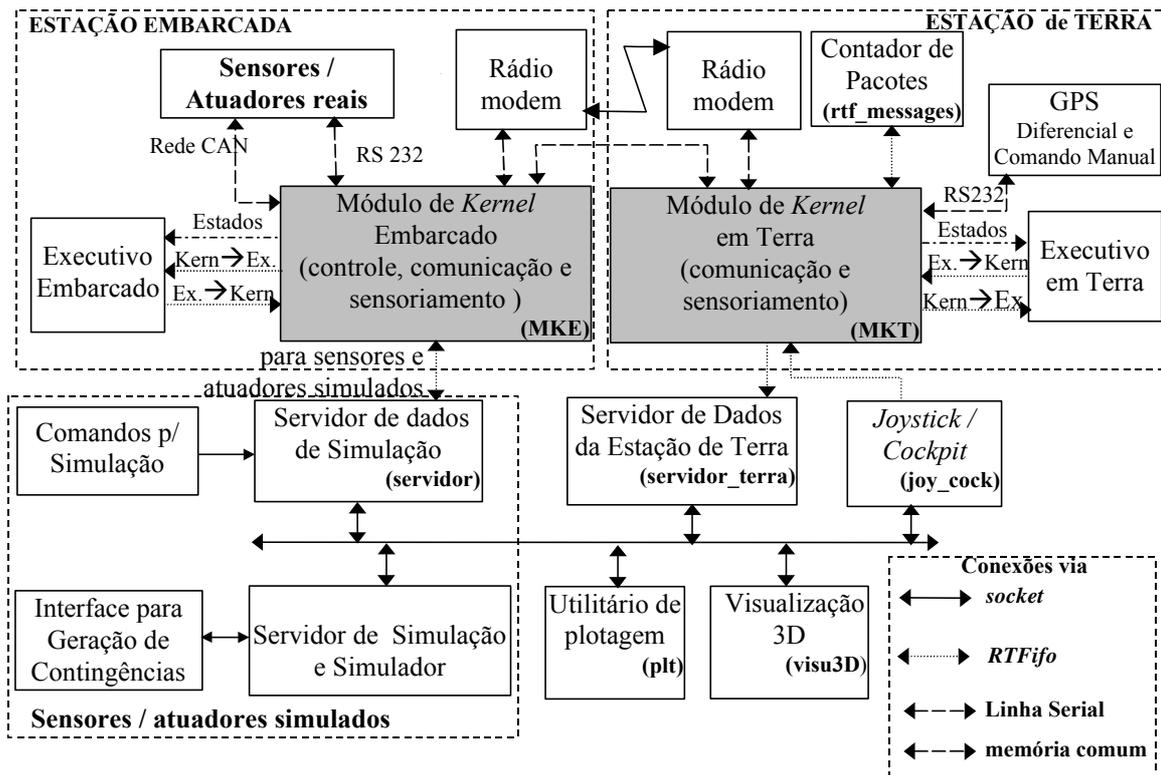


Figura 9.3: Estrutura do Protótipo Básico da Arquitetura de *Software* Robótico

#### 9.4 Estudos de Caso com a Arquitetura de *Software* Robótico

O objetivo desta seção é apresentar estudos de caso, simulados, para avaliar o funcionamento e adequação deste primeiro protótipo desenvolvido para a ASR do dirigível. Os estudos e as simulações são realizados utilizando-se o Ambiente de Desenvolvimento e Operação do dirigível, que incorpora a ASR como descrito anteriormente.

Nos estudos, a ASR executa inicialmente uma etapa de pré-planejamento, originando um conjunto de tarefas em um contexto deliberativo. Posteriormente, ao longo do voo simulado que executa o planejamento original, diferentes contingências são introduzidas, com níveis de gravidade variados. A ASR deve então detectar essas contingências e executar procedimentos de reação através de novas tarefas ou mesmo de ações de replanejamento. É importante chamar atenção que não faz parte dos objetivos desta tese se aprofundar em algoritmos de planejamento, que na maioria das situações como as mostradas nos estudos de caso, se transformam em problemas computacionalmente complexos, visto se tratarem de problemas tipo NP completo. Ou seja, o objetivo é mostrar que se dispõe de uma infra-estrutura de *software* capaz de abrigar as situações vividas por

um dirigível considerado como um sistema robótico. O planejamento inicial e os outros replanejamentos são simples e limitam-se ao contexto e objetivos explicitados.

Dois diferentes experimentos foram usados para desenvolver e testar a parte deliberativa e, mais importante, os componentes e estratégias reativas da ASR. No primeiro experimento, após o estabelecimento de um plano de vôo e do início da execução do mesmo, acontece uma contingência séria que conduz ao replanejamento pela ativação do Planejador Principal (PP), no Agente em Terra (AT). No segundo exemplo, também acontecem contingências sérias e os comportamentos reativos associados ao replanejamento também são necessários; no entanto, neste caso, a perda de comunicação entre o Agente de Terra e o Agente Embarcado (AE) torna o cenário um pouco mais difícil já que a reação de replanejamento necessita ser assegurada pelo Planejador de Tratamento de Contingências (PTC) do AE.

#### ***9.4.1 Testando o Planejador Principal em Ações Deliberativas e Reativas.***

Este experimento cobre dois casos diferentes envolvendo o PP no AT: um com replanejamento devido à variação na direção do vento e outro devido a consumo de combustível.

##### ***Caso 1 - Replanejamento Devido a Alteração no Vento***

No começo deste experimento, num passo deliberativo, uma especificação de missão é transformada num plano inicial viável para a operação do dirigível. A partir daí, o dirigível começa seu vôo, seguindo a trajetória planejada, e então uma contingência séria acontece, envolvendo a mudança drástica da direção do vento. Esta contingência conduz a um comportamento reativo de replanejamento, com o objetivo de definir trajetórias alternativas neste novo contexto; seguindo este novo plano, a missão é então executada até o final.

A decomposição temporal do estudo é a seguinte:

1. O usuário especifica um perfil de missão pela definição das localidades onde serão executadas tarefas com o dirigível; em cada localidade, as tarefas associadas são constituídas de um conjunto de sub-tarefas. Cada localidade corresponde a um nó em

um grafo, sendo que o usuário define também um “valor de lucro” para cada nó, indicando assim sua importância.

Neste estudo, há dez localidades (ou nós). A situação inicial é de vôo sob vento nulo, e a quantidade de combustível no dirigível permite a execução de toda a missão. Como já se mencionou previamente, deve-se observar que na medida que se aumenta o número de nós, o problema de planejamento cresce em termos de complexidade, podendo, no limite, face aos recursos disponíveis nas infra-estruturas embarcadas e em terra tornar-se intratável computacionalmente.

2. O PP, baseado nos “valores de lucro” e no combustível disponível no dirigível, escalona os nós e define uma seqüência ótima de trajetória em termos de consumo de combustível para a situação atual de intensidade e orientação de vento; dado que na condição inicial o combustível é suficiente, não há necessidade de priorizar os nós mais lucrativos. Para encontrar a solução, o planejador constrói uma matriz de consumo de combustível para toda combinação de deslocamentos entre os diferentes nós, mais o consumo para a execução das tarefas nos próprios nós, conforme mostrado na Tabela 1.

Nó	0	4	7	3	2	8	1	5	9	6
0 (1)	0,00	0,27	0,35	0,14	0,43	0,35	0,09	0,32	0,31	0,04
4(10)	0,27	0,00	0,09	0,23	0,29	0,25	0,27	0,06	0,04	0,24
7 (7)	0,35	0,09	0,00	0,29	0,24	0,23	0,33	0,03	0,05	0,32
3 (4)	0,14	0,23	0,29	0,00	0,30	0,22	0,06	0,27	0,27	0,14
2 (6)	0,43	0,29	0,24	0,30	0,00	0,08	0,36	0,27	0,28	0,42
8 (5)	0,35	0,25	0,23	0,22	0,08	0,00	0,28	0,24	0,25	0,34
1 (3)	0,09	0,27	0,33	0,06	0,36	0,28	0,00	0,31	0,31	0,11
5 (8)	0,32	0,06	0,03	0,27	0,27	0,24	0,31	0,00	0,02	0,29
9 (9)	0,31	0,04	0,05	0,27	0,28	0,25	0,31	0,02	0,00	0,28
6 (2)	0,04	0,24	0,32	0,14	0,42	0,34	0,11	0,29	0,28	0,00

Tabela 9.1: Matriz de consumo de combustível para o planejamento inicial.

A primeira linha e coluna desta tabela representam os nomes dos nós a serem visitados pelo dirigível, enquanto as demais colunas representam o consumo de combustível, em litros, para se deslocar entre os nós a determinada velocidade cruzeiro (5 m/s). Observa-se que a matriz é simétrica já que o vento é nulo. Aparece nesta tabela o resultado de aplicação do algoritmo de caixeiro viajante tipo *Nearest Neighbor* (MOSCATO 2001). Assim, observa-se que o vizinho mais próximo do nó inicial “0” é o ponto “6”, sendo escolhido como nó seguinte. O nó seguinte ao “6” é o “1”. Assim, na primeira coluna,

os números entre parênteses representam a ordem do nó na trajetória. Este procedimento resulta na missão mostrada na Figura 9.4-a.

3. O PP envia esta seqüência, na forma de um *script* de vôo, ao ET, que envia o plano ao EE.
4. O EE armazena o *script* de vôo e inicia sua execução pela ativação da seqüência de comportamentos requeridos para executar as tarefas: primeiro ele comanda o comportamento de decolagem, seguido pelos comportamentos de navegação apropriados. Assim que um passo programado é concluído, o fato é notificado ao ET. Se nada extraordinário acontece, o plano completo será executado e a missão concluída com sucesso. A Figura 9.4-a mostra esta situação: o plano original corresponde à linha clara – a seqüência 0-6-1-3-8-2-7-5-9-4-0 –, onde o campo reservado ao vento (*wind*) é nulo. A linha escura que aparece nesta figura representa tarefas já executadas.
5. Durante o vôo do dirigível, o gerador de contingências introduz modificações na velocidade e direção do vento. O ET detecta a nova situação e reage, pedindo ao PP um novo plano (uma ação de replanejamento). O novo plano é gerado seguindo o mesmo procedimento já descrito e é carregado no ET substituindo o plano original. Nesta situação já se tinha executado o plano até o nó “1” inclusive, sendo que o evento ocorreu quando o veículo já executava missões associadas ao nó “3”; sendo assim, a tarefa é replanejada a partir do nó “3”. A Tabela 9.2 apresenta a matriz de consumo de combustível com os nós faltantes, isto é: 3-8-2-7-5-9-4-0, de forma análoga ao mostrado na Tabela 9.1, observando-se que a matriz consumo de combustível não é mais simétrica tendo em vista a existência do vento.

nó	0	4	7	3	2	8	5	9
0	0,00	1,08	1,35	0,26	1,16	0,89	1,28	1,25
4 (4)	0,04	0,00	0,28	0,03	0,33	0,20	0,20	0,17
7 (7)	0,07	0,04	0,00	0,03	0,20	0,09	0,03	0,05
3 (1)	0,15	0,96	1,20	0,00	0,93	0,65	1,14	1,12
2 (3)	0,28	0,49	0,57	0,14	0,00	0,03	0,57	0,58
8 (2)	0,26	0,59	0,74	0,11	0,28	0,00	0,71	0,71
5 (6)	0,06	0,02	0,08	0,03	0,25	0,12	0,00	0,02
9 (5)	0,05	0,01	0,12	0,03	0,27	0,15	0,04	0,00

Tabela 9.2: Matriz de consumo de combustível para replanejamento.

6. A Figura 9.4-b mostra a situação quando do replanejamento citado anteriormente devido à alteração significativa na intensidade do vento. Assim, as tarefas já executadas aparecem em linha tracejada e a seqüência replanejada aparece em linha contínua

destacada, sendo que a seqüência 3-8-2-4-9-5-7-0, representa o novo plano que é executado até o fim.

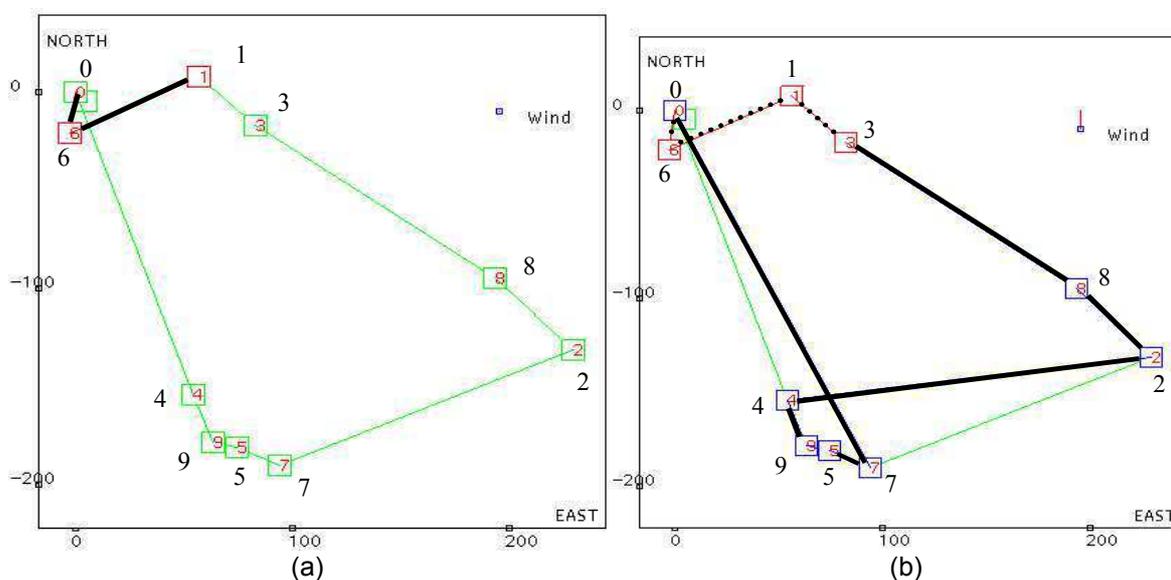


Figura 9.4: A mudança no vento muda o plano de vôo.

### ***Caso 2 - Replanejamento Devido ao Consumo de Combustível***

Neste segundo caso de estudo, a contingência séria que acontece envolve o consumo excessivo de combustível, que torna-se insuficiente para completar todas as tarefas. Este evento conduz a um comportamento relativo de replanejamento com o objetivo de selecionar as tarefas mais prioritárias para serem executadas com o combustível restante.

A existência de uma quantidade apropriada de combustível é vital para um vôo seguro do dirigível. Dada uma missão em termos de localidades associadas a nós em um grafo, como explicitado anteriormente, a política de monitoração do consumo de combustível continuamente verifica:

- i) os nós já completados e a quantidade de combustível usado;
- ii) a velocidade do vento e sua orientação;
- iii) os nós a serem executados e a predição do consumo de combustível;
- iv) o nível de combustível no dirigível.

Se desse conjunto de informações for constatada a insuficiência de combustível, comportamentos relativos serão disparados de acordo.

No começo deste experimento, um passo deliberativo traduz a missão num plano inicial; o dirigível começa então seu vôo, seguindo a trajetória planejada. No decorrer do vôo, é constantemente verificado o consumo e a disponibilidade de combustível para a missão. Ao ser detectada uma quantidade insuficiente de combustível, o PP é chamado para gerar um novo plano; neste novo plano, são suprimidos os nós com baixo lucro em número suficiente para permitir que as tarefas nos nós de maior prioridade sejam realizadas.

Esse estudo de caso é exatamente similar ao anterior até o seu item 3, o que resulta na missão mostrada na Figura 9.4-a e que é repetida na linha mais clara da Figura 9.5, (seqüência 0-6-1-3-8-2-7-5-9-4-0). O dirigível inicia a sua movimentação e, quando ele se encontra no nó “3”, é detectada uma quantidade insuficiente de combustível para execução de toda missão. Sendo assim, o PP é chamado para gerar um novo plano. Neste novo plano, são suprimidas as tarefas com baixo lucro em número suficiente para permitir que as tarefas de maior prioridade sejam realizadas com o combustível restante. O novo cenário é mostrado na Figura 9.5, onde aparece em linha tracejada a seqüência 0-6-1-3 do planejamento inicial que já tinha sido executada e, em linha contínua, o novo plano: a seqüência 3-4-7-0.

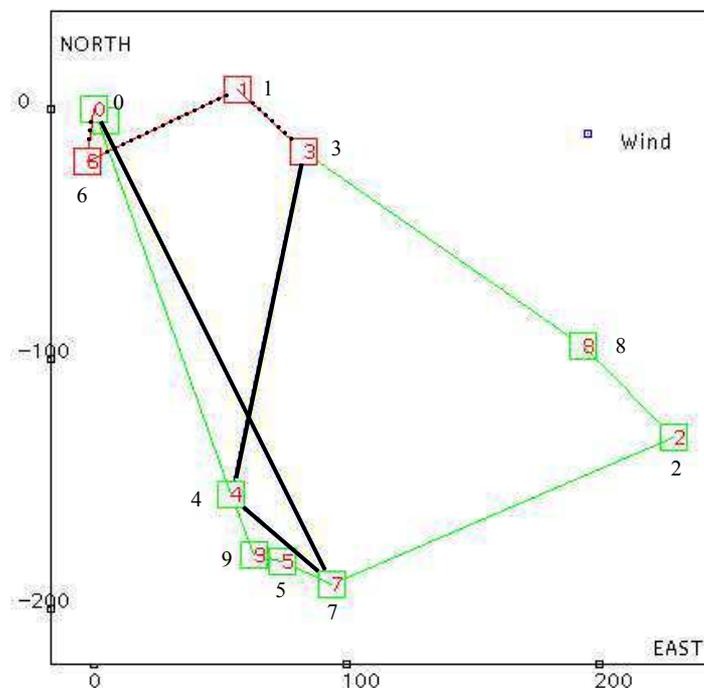


Figura 9.5: Replanejamento devido a consumo de combustível.

#### 9.4.2 Testando o Planejador de Tratamento de Contingências

Neste experimento, o dirigível foi designado para executar uma missão cobrindo uma área restrita. Durante a operação do dirigível um dos motores falha e como reação a esta contingência a ASR tem que:

- i) abortar a execução da missão original;
- ii) selecionar, de um conjunto conhecido *a priori*, um lugar seguro para aterrissar;
- iii) ir para este lugar e executar uma operação de aterrissagem.

O critério para a seleção do local leva em consideração dois aspectos em relação à posição do dirigível no momento da falha:

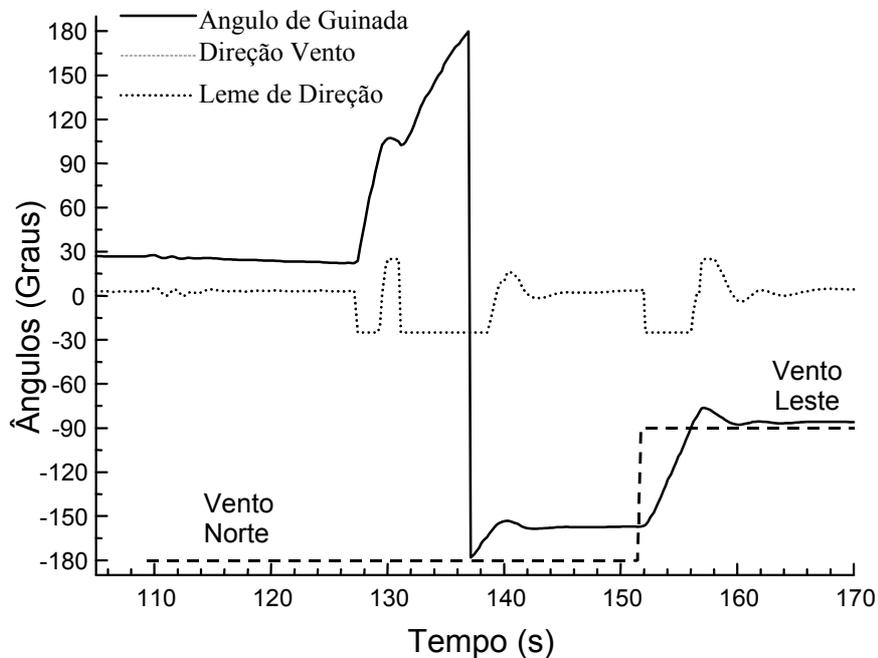
- i) proximidade (i.e., menor distância) e;
- ii) a direção que mais se aproxime da orientação do vento a favor; a motivação é que não seja necessário gastar muito combustível ou tempo para voar contra o vento.

Numa condição de operação normal, tal tipo de contingência seria tratada através de ações de replanejamento executadas pelo PP, em terra. Entretanto neste caso, é assumido que, devido à falha no sistema de comunicação, não há conexão entre o AE e o AT, resultando que o tratamento de contingências tenha que ser assegurado pelo Planejador de Tratamento de Contingências – PTC, no AE.

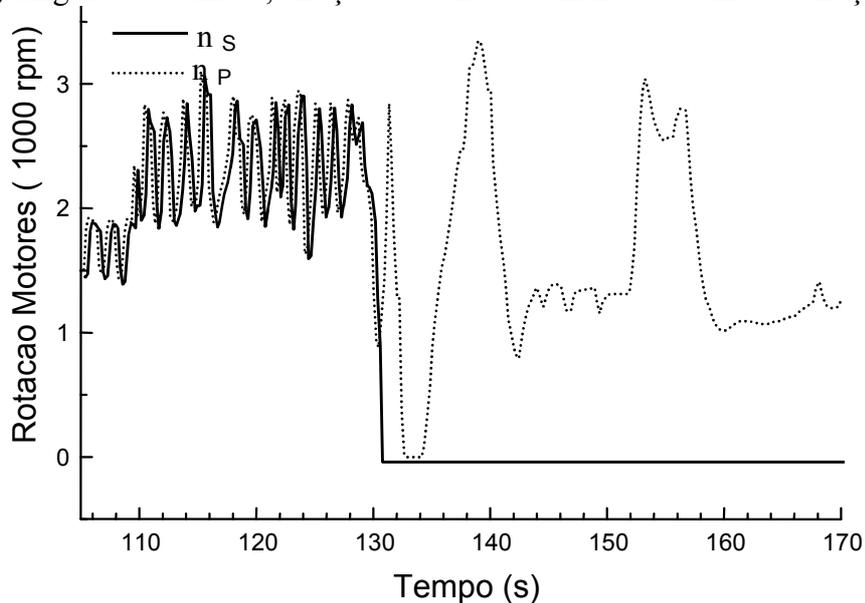
No início do estudo, os passos de 1 a 3 descritos na Seção 9.4.1, mostrado na página 192, se repetem. A partir deste ponto, a aeronave já se encontra em vôo cruzeiro, com vento nulo; em fases programadas o gerador de contingências começa então a introduzir as diferentes perturbações. A decomposição temporal da continuidade do estudo é a que se segue:

1. Uma perturbação de vento com intensidade e orientação fixas, neste caso de baixa intensidade e com direção sul, é introduzida a partir de  $t = 110$  s, como se observa na Figura 9.6-a. Nota-se também, pelas variáveis leme de direção e ângulo de guinada, que esta perturbação não chega a afetar a trajetória do dirigível.
2. Em  $t = 130$  s, o gerador de contingências provoca uma falha no motor direito, que se soma à perturbação precedente. Observa-se na Figura 9.6-b a curva denotada por  $n_s$  assumir valor nulo.

3. Inicialmente, um dos comportamentos embarcados o (CE) causa uma reação do controle de trajetória, o que é mostrado na Figura 9.6-a. Ele altera o leme de direção, ( $\zeta$ ) já a partir de  $t = 130$ s, inicialmente tentando manter a trajetória na direção Norte entre  $t = 130$  s e  $t = 132$  s, como se vê pela variável ângulo de guinada ( $\psi$ ).



a) Ângulo de Guinada, direção do vento e comando do leme de direção.



b) Rotação dos motores

Figura 9.6: Variáveis associadas a uma contingência de parada num motor numa reação dos comportamentos embarcados.

4. Entrementes, o EE detecta a falha do motor e pede um plano alternativo de vôo ao PTC, abortando a missão sendo executada. O PTC seleciona, de um conjunto conhecido

*a priori*, um local seguro para aterrissagem que implique em menor consumo e que tenha direção mais próxima à direção do vento. Neste caso é escolhido um ponto localizado ao Sul da posição atual. Então o PTC comunica a nova solução ao EE, que a envia ao CE, provocando o encaminhamento do dirigível para um local de aterrissagem de emergência na direção Sul.

5. Considerando o novo plano, o EE comanda o dirigível na direção do local escolhido de aterrissagem, mas adotando uma estratégia econômica de vôo para gastar menos combustível, uma vez que ele está usando um só motor; um menor consumo é conseguido orientando o veículo numa direção que se aproxime ao máximo da situação de vento a favor. O início deste processo pode ser observado na Figura 9.6-a entre  $t = 132$  s e  $t = 142$  s, quando o comando de guinada ( $\zeta$ ) e o ângulo de guinada ( $\theta$ ) se estabilizam. O ângulo de guinada se estabiliza numa direção próxima a  $-180^\circ$  (direção Sul), condizente com o vento Norte.
6. O gerador de contingências muda a direção do vento para Leste (isto é  $-90^\circ$ ), em  $t = 152$  s, mantendo a sua intensidade, o que pode ser visto na Figura 9.6-a.
7. O EE detecta a mudança na direção do vento que ocorre exatamente durante um procedimento de encaminhamento do dirigível para um local de aterrissagem de emergência na direção Sul. Com este fato novo, o EE chama novamente o PTC para verificar a viabilidade de escolher outro lugar para aterrissar; um novo local de aterrissagem é efetivamente adotado e a trajetória do dirigível muda de acordo, como mostrado na parte superior da Figura 9.6-a, entre  $t = 152$  s e  $t = 160$  s.

A Figura 9.7 mostra a posição do dirigível e o seu ângulo de guinada durante o intervalo de  $t = 110$  s a  $t = 170$  s, notando-se claramente que o dirigível, inicialmente apontando para direção Nordeste, toma a direção Sudoeste numa fase (após a falha no motor – passo 4 acima) e depois a direção Oeste (após a mudança na orientação do vento – passo 6 acima).

8. O dirigível chega ao local final de aterrissagem. A Figura 9.8 mostra a trajetória do dirigível cobrindo todas as fases descritas anteriormente. Ela retrata o pequeno período onde foi executada a trajetória inicialmente planejada, seguida por duas trajetórias e respectivos locais emergenciais de aterrissagem que resultaram sucessivamente dos comportamentos reativos de emergência face à parada de um motor e posteriormente devido à mudança na direção do vento detalhados na Figura 9.7.

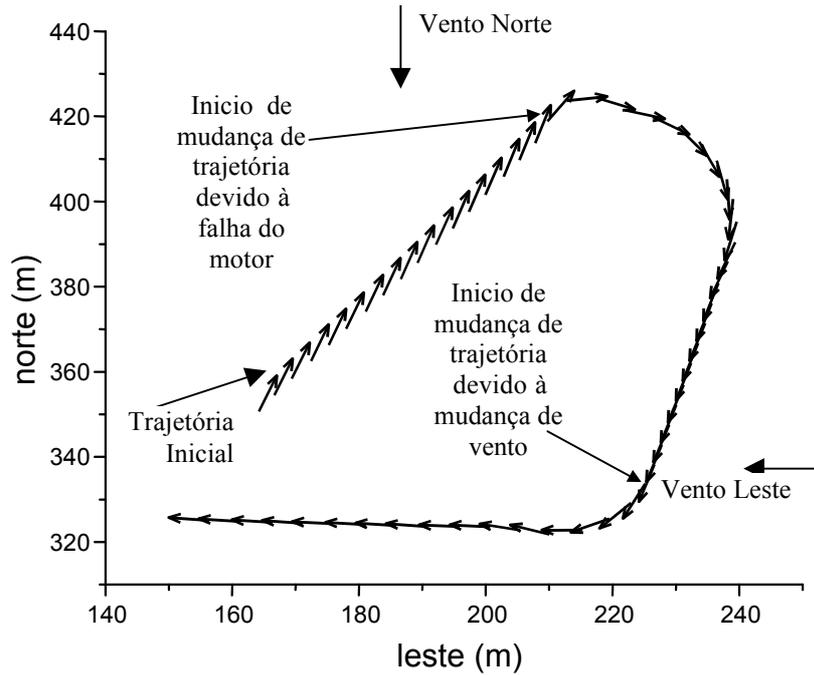


Figura 9.7: Detalhamento da posição e ângulo de guinada do dirigível.

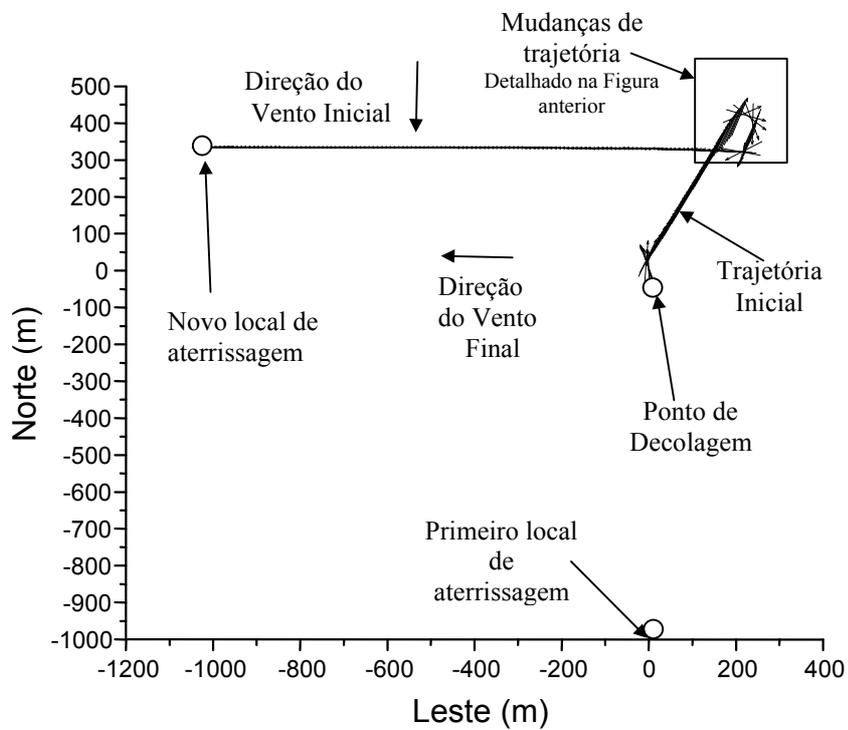


Figura 9.8: Trajetória inicial e sua alteração como resultado de contingências tratadas pelo PTC.

## 9.5 Conclusões

Este capítulo abordou a implementação e a realização de testes em simulação de um protótipo para a Arquitetura de *Software* Robótico do Dirigível. Inicialmente foi mostrado o mapeamento do conceito da ASR para o Ambiente de Desenvolvimento e Operação do dirigível utilizado no processo de desenvolvimento e ensaios da ASR. Então, foram mostrados estudos de caso onde a ASR estabelece planos de vôo (deliberação) e trata diferentes níveis de contingências (reação), que resultam no replanejamento da missão.

O conceito e as ferramentas utilizadas para comporem o ASR do dirigível se mostraram adequados pois contemplam os aspectos de deliberação e reação, possuindo potencial para implementação no veículo real pelo que ficou demonstrado nos exemplos. O uso da TDL facilitou o processo de desenvolvimento do arcabouço da arquitetura de *software* robótico do dirigível, encapsulando detalhes que sem o auxílio desta ferramenta deveriam ser manipulados diretamente.



## 10. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Os objetivos estabelecidos para esta tese foram:

- Estabelecer a infra-estrutura de *software* e *hardware* e componentes de *software* para a operação autônoma de um dirigível robótico;
- Oferecer um ambiente de suporte ao desenvolvimento e operação do dirigível;
- Estabelecer um protótipo inicial de arquitetura de software robótico para suportar aspectos deliberativos e reativos em uma missão;
- Operar um dirigível em condições climáticas ideais, com a realização de movimentos através de pontos de passagem.

Estes objetivos configuram um contexto de pesquisa eminentemente tecnológico, integrando diferentes áreas do conhecimento, envolvendo desenvolvimento e realização prática, estabelecendo um conjunto de soluções de cuja atuação harmônica resulta um dirigível robótico autônomo.

Apresentam-se neste capítulo, como conclusões, comentários sobre temas específicos abordados nesta tese. Ao seu final, são apontados cenários para trabalhos futuros a partir dos resultados alcançados.

### 10.1 Conclusões

Nesta tese, foi estabelecida uma infra-estrutura básica de *software* e *hardware* para suportar a operação autônoma do dirigível robótico, funcionalmente dividida entre dois conjuntos – embarcado e em terra – que se comunicam entre si. Esta infra-estrutura utiliza, no sistema embarcado, micro-controladores e computador da família padrão PC/104 para integrar os elementos de sensoriamento, atuação e comunicação. Em terra, ela utiliza computadores compatíveis com a família IBM/PC. A adoção de rede CAN para a integração dos componentes assegura a flexibilidade necessária para a adição de novos elementos. A base do *software* desta infra-estrutura é o sistema operacional Linux Tempo Real.

A partir da base de *software* e *hardware*, concebeu-se um Ambiente de Desenvolvimento e Operação e realizou-se um protótipo deste, congregando as estações de

terra e embarcada. Este protótipo teve seus componentes aprimorados por duas dissertações de mestrado associadas a esta tese. Esses aprimoramentos foram concebidos seguindo o princípio de orientação a objeto, organizando o processo de desenvolvimento e propiciando condições para a evolução do sistema. O ambiente apoia as fases de desenvolvimento e operação do dirigível, operando em dois modos: simulação ou real. Quando utilizado em modo de simulação, faz parte deste ambiente um simulador de dirigível. O modo de simulação permite desenvolver, testar e validar os algoritmos de controle e navegação, bem como os programas associados ao planejamento e controle das missões a serem desempenhadas pelo veículo. Durante o vôo, em modo real de operação, o ambiente permite: a interação entre operador e dirigível através da interface homem-máquina, a realização de telemetria com o armazenamento dos dados sensoriais e de controle, a configuração e sintonia de algoritmos de controle, a definição e acompanhamento da missão, o que envolve a Arquitetura de Software Robótico. Ainda no modo real, após o vôo, o ambiente oferece meios para a visualização de dados e para a realização de repetição animada do vôo, inclusive com visualização 3D.

Para atender à necessidade de simuladores de dirigível, voltados principalmente ao desenvolvimento e ensaios prévios de estratégias de controle, ao modelo matemático de dirigível originalmente desenvolvido por GOMES em 1990 foi incorporado um modelo de propulsão adequado ao dirigível utilizado e incluída a influência do vento como uma das perturbações. A partir deste modelo matemático foram desenvolvidos servidores de simulação para atender à política de uso de componentes e de distribuição do ambiente de desenvolvimento e operação. Esses servidores constituíram um ambiente de CACSD em MATLAB, desenvolvido conjuntamente com outros pesquisadores no contexto específico desta tese e outro ambiente em Java / VRML, acessível via Internet, que contribuiu para familiarização com o comportamento do dirigível. Uma versão modificada desse simulador constituiu uma ferramenta para treinamento da tele-operação do dirigível.

Para viabilizar o atendimento da meta de realização de navegação por pontos de passagem, a partir dos trabalhos de Azinheira e De-Paiva, reinterpretou-se esse esquema de controle. Procedeu-se então à estudos de sintonia e testes em simulação e em vôos reais, deste sistema controle e navegação, que incorpora técnicas de controle de trajetória e altitude. O estudo de fatores que influenciam o desempenho do sistema de controle e navegação abordou ventos, turbulências atmosféricas, flutuabilidade, configuração dos

lemes de direção e profundidade, bem como os próprios parâmetros de sintonia dos controladores, e também, a operação simultânea dos algoritmos de controle de trajetória e de altitude. Dois desses fatores se destacam: a velocidade do veículo relativa ao ar, influenciada, principalmente, pelo vento e turbulências que mudam constantemente com a situação de vôo, alterando o comportamento dinâmico do veículo; e a execução de transições abruptas nas referências de trajetória quando da ação conjunta dos controles de trajetória e altitude, pois neste caso a ocorrência de saturação dos atuadores leva à perda de desempenho do sistema de controle.

Assim, face à variação na dinâmica do dirigível em função da condição de vôo – dependente principalmente da velocidade do veículo relativa ao ar –, constata-se a inadequabilidade do uso de parâmetros de sintonia fixos nos controladores. Para se obter um melhor desempenho do sistema de controle e navegação, uma alternativa é o uso de recursos de escalonamento de ganhos para variar os parâmetros de sintonia em função da condição de vôo. Em relação às mudanças de trajetória, essas devem ser mais suaves, o que pode ser obtido na prática pelo uso de curvas ou uma sequência de retas de pequeno comprimento e orientação variável, que realizem o processo de mudança de direção passo a passo, ou da utilização de funções de interpolação que variem gradativamente o raio de curvatura da trajetória. Conforme foi constatado, outro fator determinante e importante, que tem que ser considerado em situações de saturação, é o uso de valores variáveis para os limites de saturações dos comandos de arfagem, guinada e rolagem, determinando-os por um sistema de supervisão que privilegie o comando mais importante numa dada condição de vôo.

O desenvolvimento de um modelo conceitual de uma arquitetura de software robótico para o dirigível, e sua avaliação em simulação no ambiente de desenvolvimento e operação, além de constituir um primeiro protótipo de arquitetura, permitiram traçar direções para a implementação de recursos que tratem contingências e as integrem adequadamente com capacidades de planejamento e deliberação no dirigível.

Em termos de *hardware* eletro-mecânico, novos componentes e soluções foram gerados – principalmente sistema de vetorização e motorização, gôndola, elementos de atuação das superfícies aerodinâmicas – de forma a adequar o dirigível original à operação como veículo robótico.

O dirigível robótico, com certo nível de operação autônoma, estabelecido neste trabalho de tese, é inovador no cenário tecnológico científico, tendo originado 2 artigos em periódicos e 28 artigos em conferências.

Adicionalmente, este trabalho cumpriu uma outra função que é a de contribuir à formação de recursos humanos através de diversos trabalhos de iniciação científica e da orientação técnica de duas dissertações de mestrado (MAETA, 2001) e (MIRISOLA, 2001), que constituíram importante fator de apoio aos resultados alcançados no desenvolvimento desta tese.

Como conclusão final, o desenvolvimento e a integração coesa de diferentes soluções tecnológicas, cujo resultado foi o estabelecimento de um dirigível robótico com um nível de autonomia relevante, validado experimentalmente, constitui o cerne e contribuição deste trabalho de tese. Guardadas as limitações intrínsecas devido ao porte do veículo, tais como sensibilidade ao vento e reduzida autonomia (i.e. tempo de vôo), este dirigível robótico mostra-se apto à realização de aplicações piloto e, ao mesmo tempo, constitui uma plataforma experimental que permite a continuidade e evolução do Projeto AURORA, calcado em grande parte neste trabalho.

## **10.2 Trabalhos Futuros**

Como trabalhos futuros a serem desenvolvidos, dando continuidade aos aqui descritos, sugere-se:

- A incorporação de características no sistemas embarcado e de terra para permitir a operação a distâncias maiores entre ambos, como a inclusão de redundância no enlace de comunicação terra-ar e o uso formas alternativas de comunicação, tais como linhas de telefones celulares digitais ou comunicação via satélite;
- A introdução de estratégias de controle mais adequadas à dinâmica variável do dirigível, como o caso de controle com escalonamento de ganhos dos controladores e o tratamento de condições de saturação nos atuadores, agregando um nível inteligente de supervisão;
- O incremento das capacidades do sistema de controle com a inclusão de controle de velocidade; a decolagem, aterrissagem e vôo pairado automáticos, com a inclusão de sensores adicionais;

- A evolução do sistema de navegação, tratando trajetórias diferentes de semi-retas e incorporando transições suaves entre diferentes orientações levando-se em conta a manobrabilidade do dirigível;
- O uso do ambiente de desenvolvimento e operação para a coleta de dados e posterior uso em procedimentos de identificação, visando o ajuste e validação do modelo matemático do dirigível;
- O uso do ambiente de desenvolvimento e operação para o desenvolvimento e ensaios de técnicas de controle baseadas em visão (já se encontram em desenvolvimento por pesquisadores do Projeto AURORA);
- A implementação, nas infra-estruturas embarcada e de terra, do protótipo da arquitetura de *software* robótico do dirigível, agregando capacidades de deliberação e reação em vôos reais;
- A introdução, no sistema embarcado, de um módulo de fusão sensorial e percepção, gerando informações a serem utilizadas pelo sistema de controle e navegação, bem como pela arquitetura de *software* robótico do dirigível;
- A introdução, no sistema robótico, de conceitos e soluções visando a tolerância à falhas;
- O uso automático de informações meteorológicas para planejar ou replanejar o vôo de dirigíveis com longa autonomia;
- O uso de dados obtidos pelo dirigível para realizar a operação integrada e cooperativa com robôs móveis em terra;
- A extensão de todas as soluções técnico científicas para dirigíveis não tripulados de maior porte, cobrindo maior área e com maior autonomia de vôo, conforme a evolução planejada no Projeto AURORA.

No contexto de aplicações, a realização de missões piloto dentre as já identificadas no Projeto AURORA, - contribuições advindas de diversas instituições -, tais como:

- Setor de Petróleo e gás (PETROBRAS - Petróleo Brasileiro, e TBG - Transportadora Brasileira do Gasoduto Bolívia-Brasil), envolvendo como principais aplicações:
  - Dutos: monitoração de erosão, ocupação e atividade humana, e de “pontos notáveis”, ao longo da faixa de dutos, através da inspeção no espectro de luz visível; localização de vazamentos de gás e óleo, através da inspeção no espectro

infra-vermelho ou baseado em sensores especiais de gases; o uso de radar com penetração ao solo para inspeção de dutos;

- Meio Ambiente: monitoração da superfície do solo e de águas em caso de vazamento, através de inspeção visual ou do uso de sensores especiais; medição de qualidade de ar;
- Inspeção visual: principalmente na Amazônia, observação de estações de produção e poços de perfuração em locais de difícil acesso, bem como acompanhamento da logística de transporte por rios;
- Agricultura de Precisão (EMBRAPA - Empresa Brasileira de Pesquisa Agropecuária, através das unidades CNPTIA e CNPDIA): aplicações em agricultura e zootecnia de precisão, visando estudos no espectro visível e infra-vermelho para avaliação de nível de umidade e nutrientes no solo, associando-os a níveis de produção agrícola.
- Amazônia: (INPA - Instituto Nacional de Pesquisas da Amazônia): vôos estacionários e geração de imagens em região de estudo nas proximidades de Manaus, no contexto de um sistema para a classificação automática e estudo de espécies a partir de imagens de copas de árvores.
- Ecologia (Departamento de Ecologia e Biologia Evolutiva da UFSCAR e Instituto de Biologia da UNICAMP): geração de imagens de áreas de interesse ecológico, como parques, reservas e regiões lacustres presentes em SP e MG.
- Controle ambiental: (SABESP e CETESB, em SP): geração de imagens para avaliação ambiental em solo e cursos d' água e represas; aferição de níveis de poluição no ar.
- Aerofotogrametria: (TECNOMÉTRICA) Aerofotogrametria de baixo custo para planejamento urbano em cidades de pequeno porte.
- Militar: (2a. Cia. de Comunicações Blindadas do Exército, em Campinas): geração de imagens da área de treinamento militar, e outras aplicações.

## ANEXO 1. RECURSOS DE SUPORTE PARA DESENVOLVIMENTO DO DIRIGÍVEL ROBÓTICO

### A1.1 Introdução

Conforme descrito na Seção 2.2.3, para o desenvolvimento de um dirigível robótico é necessário um conjunto de recursos de suporte constituindo a base tecnológica sobre a qual são implantados metodologias e respectivos algoritmos para dotar o dirigível de capacidade de operação autônoma.

Dentre os recursos de suporte, foram explicitados: a integração de diferentes componentes de *hardware*, como processadores, sensores, elementos de atuação no dirigível e elementos de comunicação; o *software* de tempo real necessário para prover a observância a requisitos de tempo na execução das diversas rotinas computacionais que compõem o sistema robótico; e a arquitetura de *software* robótico, destinada a permitir a programação, controle e execução da missão. Estes recursos são organizados de forma a originar as infra-estruturas de *hardware* e *software* tanto para a estação embarcada no dirigível quanto para a estação de terra destinada à sua operação, compreendendo ainda a comunicação entre ambas. Estas infra-estruturas devem apresentar características de funcionalidade, flexibilidade e robustez. Além desses recursos, foi também mencionada a necessidade de o dirigível possuir uma estrutura mecânica apta ao desenvolvimento e uso do veículo como um sistema robótico.

O objetivo deste anexo é apresentar os recursos de suporte que foram concebidos para permitir o desenvolvimento de um dirigível robótico capaz de desenvolver vôos autônomos supervisionados de uma forma confiável. Assim, na Seção A1.2 apresenta-se o dirigível em sua forma original como veículo rádio controlado, bem como os aprimoramentos nele introduzidos para torná-lo um veículo robótico. A Seção A 1.3 é dedicada à apresentação da infra-estrutura de *hardware*, compreendendo tanto a estação embarcada quanto a de terra, detalhando os sensores, os atuadores, os computadores e elementos de comunicação entre ambas estações. Na Seção A1.4 são apresentados os elementos de infra-estrutura de *software*, incluindo o Linux tempo real e o arcabouço do *software* para o sistema de controle e navegação.

## A1.2 O Dirigível

O dirigível utilizado no AURORA-I (vide Capítulo 1), mostrado na Figura A1.1, é o AS800 fornecido pela empresa *Airspeed Airships* (AIRSPEED, 2001). Este dirigível foi o escolhido por ser o que apresentava maior capacidade de carga dentre os dirigíveis rádio controlados disponíveis no mercado em 1997, quando foi efetuada a sua aquisição. A primeira versão deste dirigível, chamada de AS800-1, tem aproximadamente 9m de comprimento e 2,3m de diâmetro, utiliza 24m<sup>3</sup> de hélio, o que propicia uma carga paga (*pay load*) de 9Kg ao nível do mar. A segunda versão, chamada AS800-2, tem 10,5m de comprimento, 3m de diâmetro, utilizando 30m<sup>3</sup> de hélio. Devido aos 4Kg adicionais do novo envelope, a carga útil é de 11Kg. Ambas versões destinam-se originalmente à pilotagem por controle remoto, isto é, não possuem os recursos específicos necessários à realização de vôos autônomos. O desenvolvimento desses recursos é um dos enfoques do presente trabalho.

Na Figura A1.1 estão ilustrados os principais componentes do AS800, como envelopes (AS800-1, no caso), superfícies aerodinâmicas, elementos de propulsão, e elementos de apoio (mastro de atracamento e carrinho para apoio ao transporte). Eles são detalhados a seguir.

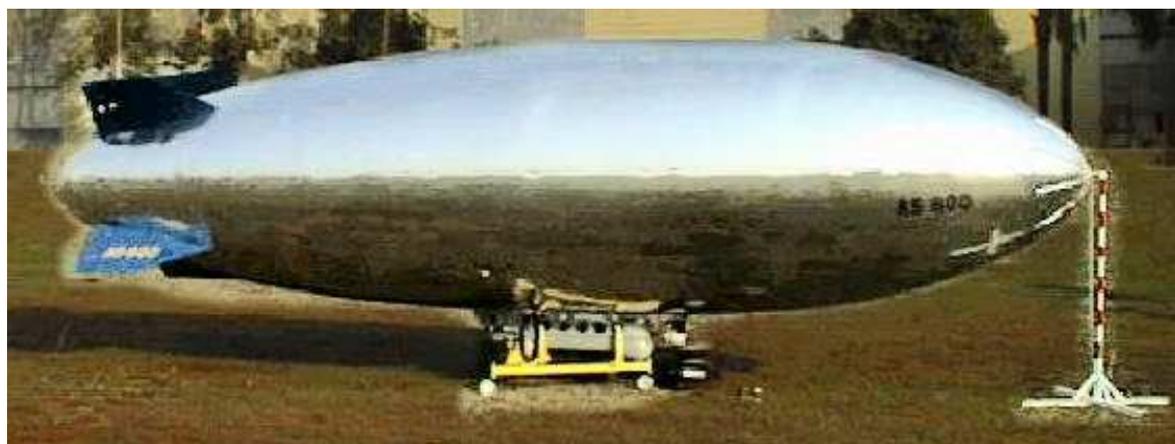


Figura A1.1: O dirigível AS800-1.

### A1.2.1 Envelope

O envelope é do tipo não-rígido, construído a partir de folhas de material plástico unidas através da sobreposição e aquecimento das mesmas. No AS800-1 ( Figura A1.1) o material utilizado é o *mylar* (duas camadas plásticas com uma película de alumínio em seu interior), enquanto que no AS800-2, mostrado na Figura A1.2, usa-se o *nylon*. O *mylar* é

mais leve que o *nylon*, porém o último apresenta uma resistência superior. O envelope possui uma série de engates e velcros que permitem fixar os demais componentes: gôndola, superfícies aerodinâmicas, bico para atracação.



Figura A1.2: O dirigível AS800-2.

A Figura A1.3 apresenta a disposição geral dos componentes sobre o envelope notando-se, além das superfícies aerodinâmicas, a presença:

- i) da válvula para escape de hélio, que é aberta em situações de emergência em que se deseja a diminuição da força aerostática;
- ii) do ponto de fixação da antena do GPS sobre o envelope e da sonda de vento junto ao bico de atracamento (descritas na seção A1.3.3.4);
- iii) da gôndola (descrita na seção A1.2.3), onde estão instalados componentes embarcados e o sistema de propulsão.

#### **A1.2.2 Superfícies Aerodinâmicas:**

As superfícies aerodinâmicas são estruturas construídas em madeira balsa, utilizada em aeromodelismo para se obter maior leveza, recobertas com uma película plástica termo-retrátil. Cada superfície é dotada em seu interior de um servo-motor que permite a movimentação de uma aleta articulada.

As formas típicas de fixação das superfícies de atuação sobre o envelope de dirigíveis são: a configuração em “+”, ilustrada esquematicamente na Figura A1.4-a, com os lemes de direção e de profundidade, ou a configuração em 'x', como na fotografia mostrada na Figura A1.4-b.

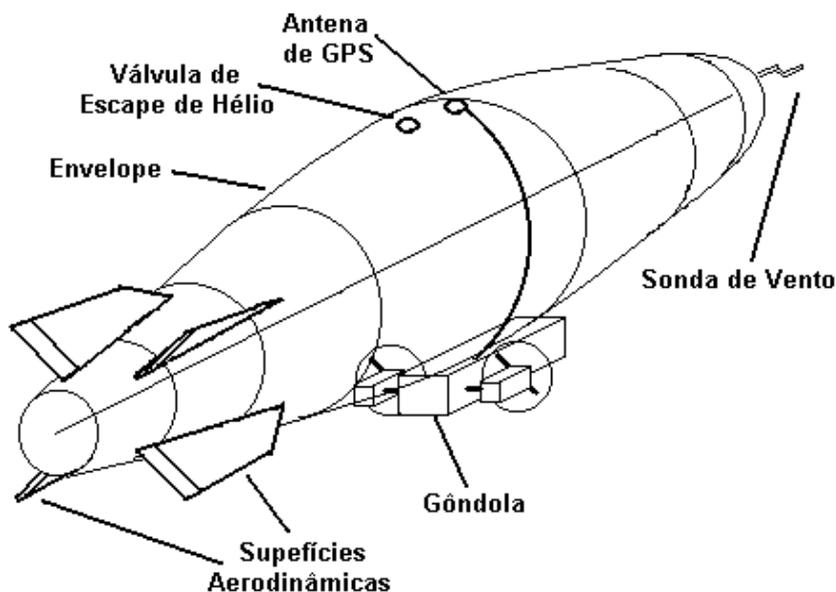


Figura A1.3: Disposição geral dos componentes do dirigível.

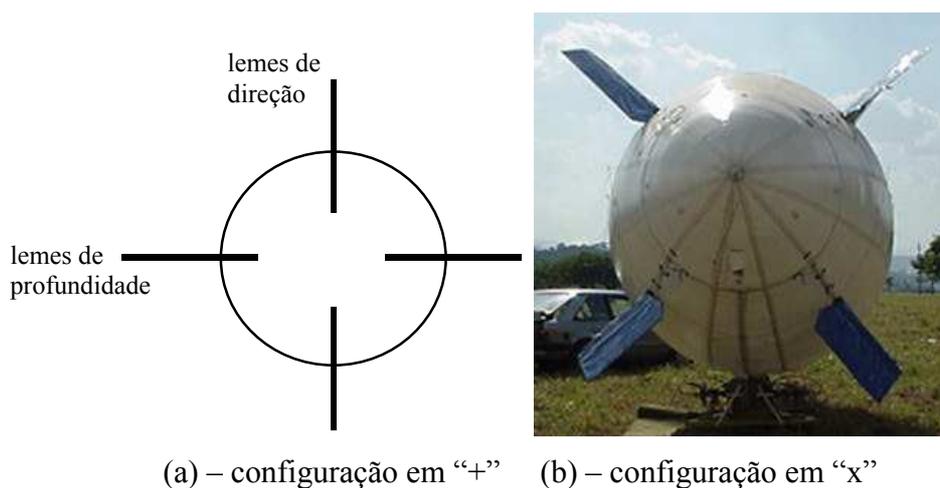


Figura A1.4: Visão posterior do dirigível e montagens das superfícies aerodinâmicas.

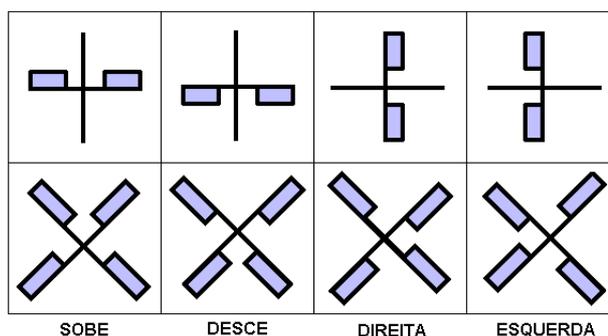


Figura A1.5: Equivalência entre a atuação dos lemes em “+” e dos lemes em “x” (vista posterior).

O entendimento do princípio de atuação em “+” é mais intuitivo, visto que a mudança de direção, como uma guinada é exercida pelo leme de direção e a mudança de

altitude é exercida pelo leme de profundidade ou *profundor*. Para atuadores em “x”, para se obter uma mudança de direção, tal como uma guinada, ou variação de altitude, é requerida uma combinação de movimentos das quatro superfícies. A Figura A1.5 (MAETA, 2001) apresenta uma equivalência entre a atuação em “+” e “x” para o mesmo comando.

Adicionalmente aos comandos de guinada e profundor, desde que se possua comando individual de cada um dos lemes, é possível uma outra combinação de comandos que induz o giro do dirigível em torno do seu eixo longitudinal, esse comando é chamado de comando de rolagem, também muito conhecido como comando de "aileron".

Consultas ao fabricante do AS800 bem como ao Dr. Sérgio Varella Gomes, especialista em dirigíveis e autor referência neste trabalho (GOMES, 1990), bem como a prática de vôo com o dirigível rádio-controlado, mostraram que:

i) A atuação em “x” é mais eficiente para um comando individual, guinada ou subida, principalmente em situações de emergência, quando pode se requerer que se privilegie um comando em prejuízo do outro, podendo-se, neste caso, utilizar uma área até duas vezes maior para realizar um comando individual. Já para comandos compostos como curvas ascendentes ou descendentes a atuação em “+” é mais eficiente;

ii) A atuação em “x” permite maiores ângulos entre o veículo e o plano horizontal no procedimento de decolagem, com menor risco de que as superfícies aerodinâmicas da parte inferior toquem o chão.

No AS800, as superfícies de atuação são montadas sobre o envelope numa configuração em “x”.

Na evolução do AS800-1 para o AS800-2, devido ao novo envelope, estimativas iniciais mostram um aumento de 30% da massa virtual e a quase duplicação da inércia virtual, dificultando as manobras em curvas; isto levou à necessidade de aumento da área de deflexão das superfícies aerodinâmicas, incrementando-se assim a ação (autoridade) das mesmas; o aumento da área de deflexão foi da ordem de duas vezes, o que pode ser visto na Figura A1.6, onde se mostra uma superfície aerodinâmica em processo de duplicação.



Figura A1.6: Superfície aerodinâmica do AS800-1 (coberta) e sua extensão para uso no AS800-2.

### A1.2.3 Gôndola

A gôndola original do veículo rádio controlado possui uma carcaça construída utilizando-se fibra de vidro e de carbono com uma estrutura interna feita em madeira balsa, com o *design* clássico de gôndolas de dirigível. Dentro da estrutura da gôndola estão instalados o sistema de vetorização, tanques de combustível, receptor de rádio controle e baterias. Na parte externa da gôndola, estão instalados os motores que, juntamente com o sistema de vetorização, compõem o conjunto de propulsão do dirigível.

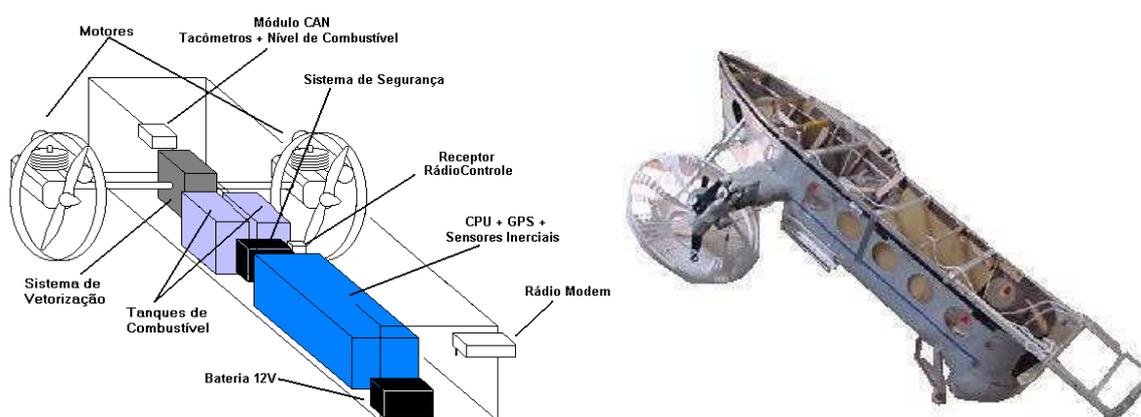


Figura A1.7: Detalhe da disposição dos componentes embarcados na gôndola original.

A vetorização consiste na orientação dos motores para cima ou para baixo em torno da posição neutra (horizontal). No intuito de tornar o AS800 um veículo robótico, instalou-se no interior da gôndola original, como ilustrado na Figura A1.7, a caixa básica do sistema

embarcado, contendo computadores e sensores. No seu exterior foi instalado o rádio-modem. Estes componentes estão descritos na Seção A 1.3.

Visando tornar a gôndola mais adequada para o uso do dirigível como veículo robótico, reformulou-se o seu *design* (PEIXOTO, 2001). A Figura A1.8 mostra este novo conceito, onde há maior acessibilidade aos componentes e se isola o sistema responsável pela propulsão, fonte de vibrações, do sistema contendo sensores e computadores embarcados. Esta isolação pode incluir a montagem de elementos isoladores na fixação do sistema de propulsão, se necessário.



Figura A1.8: Nova gôndola.

Partes da nova gôndola, como o sistema de suporte dos motores, sistema de proteção das hélices e o bloco que contém os tanques de combustível e baterias, foram construídas usando a técnica de prototipagem rápida em Duraform GF, uma poliamida de patente da *DTM Corporation* (PEIXOTO, 2001). A partir da geração de um sólido 3D em CAD, e após um processo de CAM, a peça é produzida pela máquina Sinterstation 2000 que utiliza um processo de sinterização por camadas a laser. Os demais elementos com finalidade estrutural, como o quadro de sustentação, trens de pouso e outros componentes, foram construídos com perfis de alumínio.

#### **A1.2.3.1 Motores de Combustão e Hélices:**

No AS800-2 são utilizados dois motores de aeromodelismo - modelo OS-91FX (OSE, 2001). Eles alcançam uma rotação máxima de 15.000 RPM (dependendo da hélice utilizada), fornecendo a potência de 3,5 HP cada e uma força total de aproximadamente 80N. Alterando-se as hélices consegue-se variar a rotação máxima e a curva de potência que pode ser obtida. Esses motores estão substituindo os antigos que equipavam o AS800-

1, os quais além de possuírem potência menor, geravam mais ruído e consumiam uma quantidade maior de combustível, reduzindo a autonomia de voo.

No processo de aprimoramento do conjunto de motorização fornecido originalmente no AS800-1, a fim de encontrar um conjunto apropriado para as novas dimensões do AS800-2, foi necessário realizar uma avaliação teórica e prática do conjunto, isto é, dos motores e diferentes opções de hélices. Detalhes desse estudo são apresentados no Anexo 2.

A Figura A1.9 mostra detalhes do motor, hélice e carcaça de proteção ao redor da hélice construído em prototipagem rápida (PEIXOTO, 2001). Além disso, mostra-se um conjunto de hélices utilizadas para se encontrar uma hélice adequada para o sistema de propulsão do dirigível.



Figura A1.9: Motor e conjunto de hélices já utilizados.

#### ***A1.2.3.2 Sistema de Vetorização dos Motores:***

O sistema original de vetorização dos motores do dirigível foi projetado para uma faixa limitada, coerente com o uso do veículo de forma rádio-controlada, onde a flexibilidade de manobra não é fator determinante. Com este sistema, os motores eram vetorizados entre  $0^\circ$  (horizontal) e  $120^\circ$  (para cima) (ver Figura 3.2 da Seção 3.2). Esses valores mostraram-se insuficientes para operações de decolagem e aterrissagem verticais, ou voo pairado com maior capacidade de ajuste, que requerem um ângulo de vetorização total de  $180^\circ$ , ou seja  $120^\circ$  para cima e  $60^\circ$  para baixo para ser possível a geração de força de propulsão forçando a descida do veículo. Outro problema do mecanismo original era sua baixa confiabilidade, devido ao seu sistema de funcionamento através de varetas que

transmitiam o torque de servo-motores para o eixo de vetorização, o que provocou alguns acidentes em consequência de quebra das engrenagens dos servo-motores.

O mecanismo substituto desenvolvido, mostrado na Figura A1.10, possui acionamento mais confiável através de uma montagem de polias sincronizadas. O curso de operação deste elemento é de  $180^\circ$  ( $120^\circ$  para cima,  $60^\circ$  para baixo), o que é possibilitado pela relação de redução das polias.



Figura A1.10: Novo mecanismo de vetorização.

#### **A1.2.4 Rádio-controle, Transmissor e Receptor e Servo Atuadores**

O conjunto básico de comando e atuação fornecido originalmente com o dirigível encontra-se facilmente em lojas de aeromodelismo. Ele consiste de um rádio transmissor e seu receptor e de servos, normalmente servos de posição.

O dirigível original, rádio controlado, utiliza como rádio-controle um modelo Futaba PCM1024 totalmente digital (FUTABA, 2001). Ele permite modulação em frequência FM ou PCM (do inglês PCM – *pulse code modulation*) para transmissão dos sinais de comando. Nos vôos por rádio controle é utilizada a modulação PCM, que permite maior alcance e menor interferência do sinal. O alcance é de 1,2 km em visada direta.

Na versão original do AS800 os servo motores estão conectados diretamente ao receptor de rádio, recebendo deste um sinal de modulação por largura de pulso (do inglês PWM – *pulse width modulation*) onde o valor da largura do pulso corresponde à referência

do ângulo de posicionamento do servo-motor. Os servo-motores comandam a abertura dos carburadores dos dois motores a combustão, os ângulos das superfícies aerodinâmicas e o ângulo de vetorização.

### A 1.3 Detalhamento da Infra-estrutura de *Hardware* e Comunicação

Conforme é mostrado na Seção 7.5.1 o esquema de conexão dos componentes de *hardware* do protótipo é o mostrado na Figura 7.4 repetido na Figura A1.11.

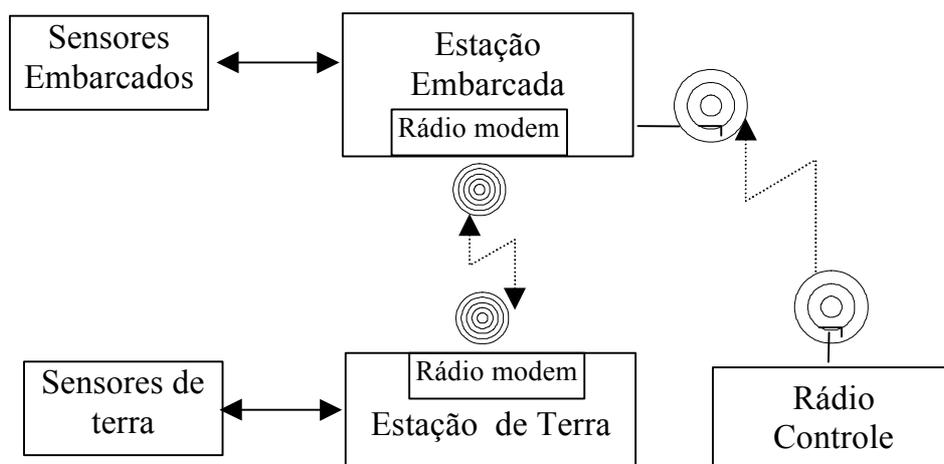


Figura A1.11: Conexão dos equipamentos.

A Figura A1.12 fornece detalhes da infra-estrutura desenvolvida, que constitui uma contribuição importante deste trabalho. Esta infra-estrutura é descrita nas subseções a seguir. Cabe ressaltar que os componentes do sistema embarcado foram selecionados segundo os seguintes critérios:

- i) tamanho e pesos reduzidos;
- ii) baixo consumo de energia;
- iii) imunidade a vibrações;
- iv) facilidade de configuração e programação utilizando ferramentas existentes no mercado e;
- v) baixo custo.

Os sensores foram selecionados segundo um critério adicional: estes devem ser capazes de funcionar dentro dos intervalos de valores ditados pela dinâmica do dirigível (muito mais lenta que a de outros veículos aéreos). Cabe ressaltar os componentes de

hardwrae são escolhidos dentre de opções disponíveis no mercado e que freqüentemente esta infra-estrutura é aprimorada, através da incorporação de soluções mais modernas e de melhor desempenho.

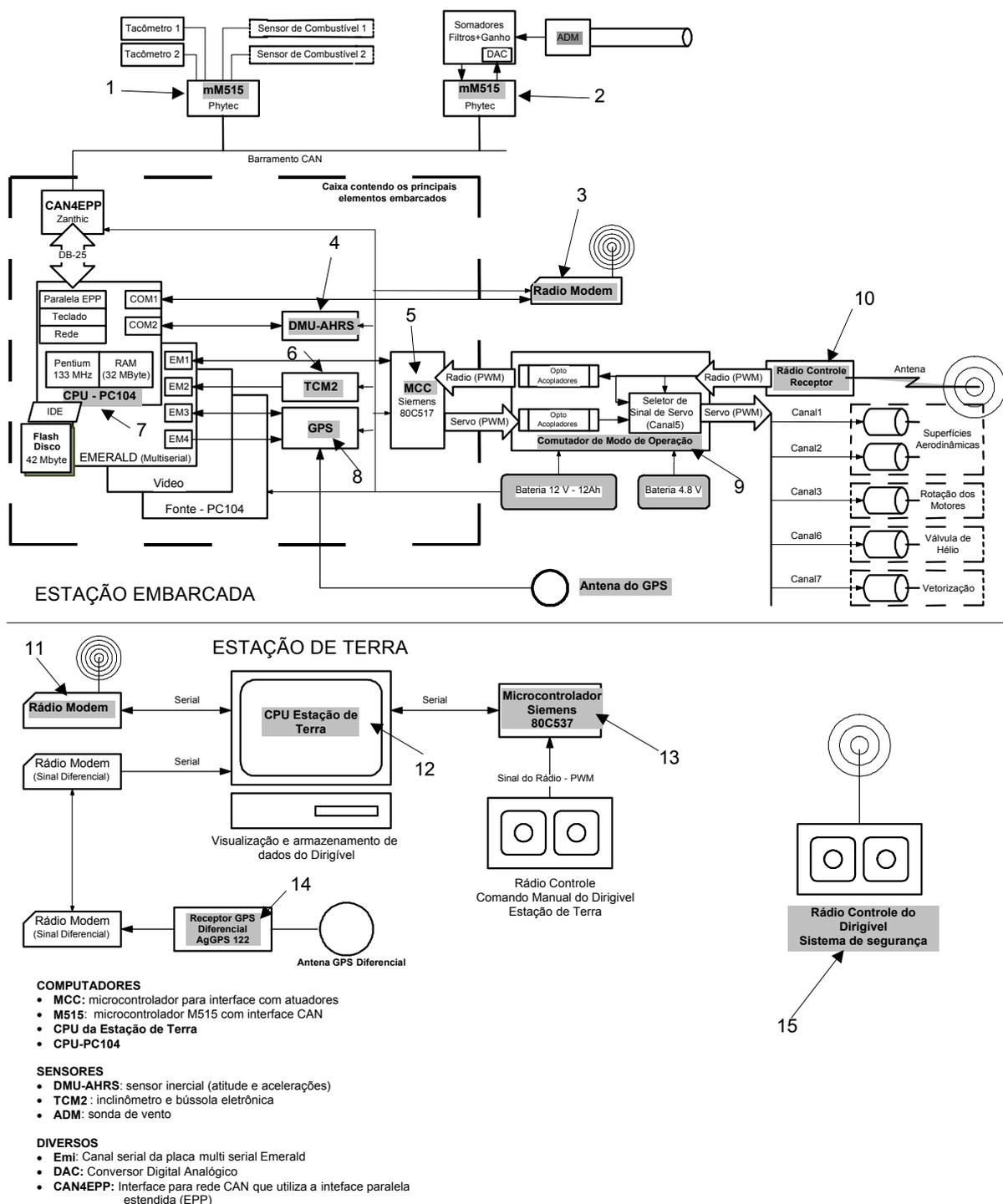


Figura A1.12: Detalhamento da conexão dos componentes de *hardware* do dirigível.

### A1.3.1 Computadores e Elementos de Comunicação

Excluindo os processadores que fazem parte dos sensores, a infra-estrutura computacional desenvolvida dispõe das seguintes unidades de processamento,.

- CPU principal embarcada – PC104
- Micro-controladores com interface CAN - m515;
- Micro controlador 517 (MCC) responsável pela geração dos sinais de comandos dos servo-motores;
- *Hardware* dedicado para comutação do modo de operação;
- Elementos de comunicação;
- CPU da estação de Terra.

Estes elementos são detalhadas a seguir.

A Figura A1.13 (ver também na Figura A1.12 próximo aos indicadores 4, 6, 7 e 8) mostra a caixa principal da eletrônica embarcada, aparecendo nesta a CPU principal, o micro-controlador (MCC), o sensor de atitude TCM2, o receptor de GPS e a central inercial; estes sensores são apresentados na Seção A1.3.3

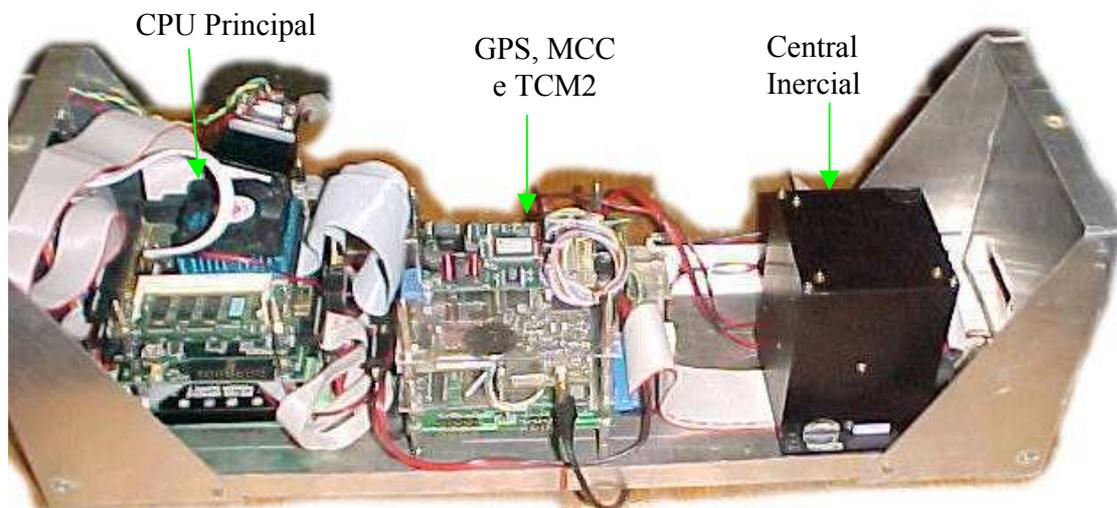


Figura A1.13: Caixa contendo a eletrônica principal.

#### A1.3.1.1 CPU da Estação de Terra

A CPU Principal de Terra utilizada previamente, mostrada na Figura A1.14-a (ver também na Figura A1.12 o indicador 12), é composta por um computador Pentium II com uma placa multiseriada, modelo IOAT66 (BRI, 2001), que possui seis portas seriais com velocidades até 115 kbps. A esta placa estão acoplados o rádio modem que realiza a

comunicação com o sistema embarcado e o sistema de GPS diferencial. É possível também a conexão de um sistema de comando manual convencional através de um *hardware* dedicado (ver também na Figura A1.12 o indicador 13), que foi desenvolvido baseado em micro-controlador 80c537 (SIEM517, 1994) e que realiza a interface e conversão dos sinais PWM de um rádio controle convencional. Esta solução, entretanto, não está sendo utilizada.

Posteriormente, a estação de terra foi migrada para um computador portátil com duas interfaces seriais, o que é mostrado na Figura A1.14-b.

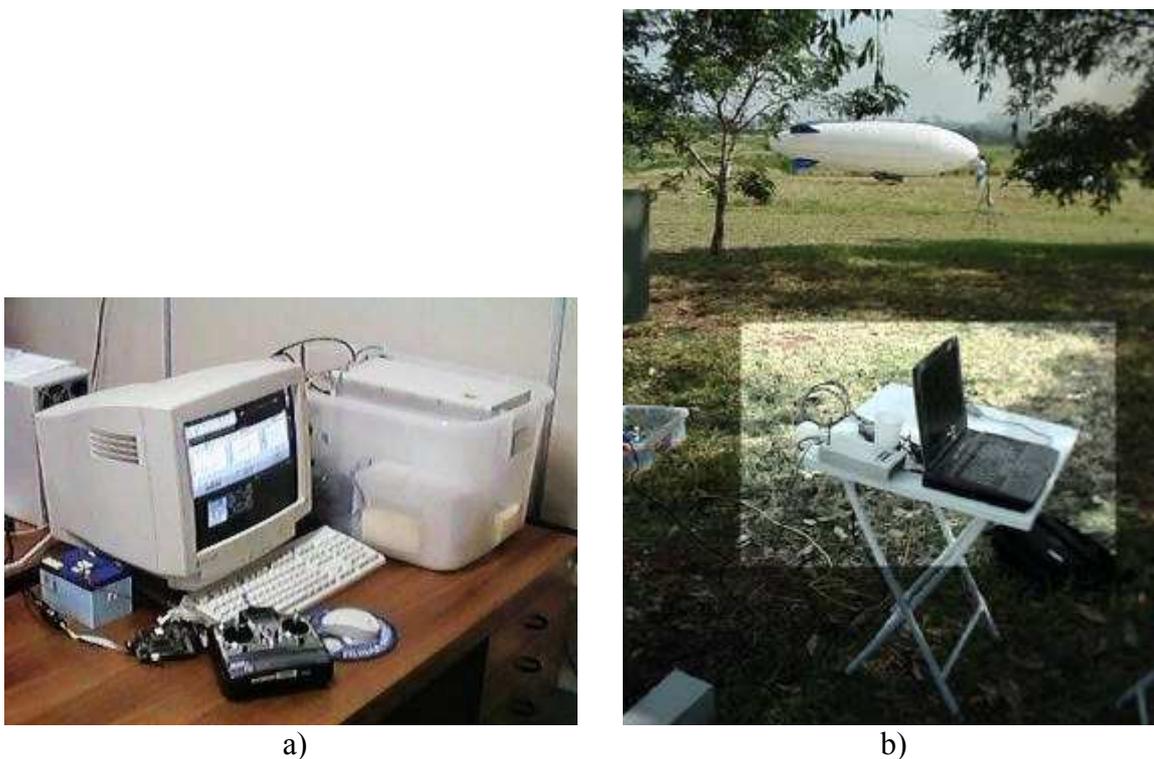


Figura A1.14: Componentes da estação de terra.

### ***A1.3.1.2 Computador Principal Embarcado***

Em MAETA (2001), são apresentadas e discutidas diferentes alternativas para sistemas embarcados, com variada capacidade de processamento, abrangendo micro-controladores, DSPs e Transputers. Conforme explicitado anteriormente, a infra-estrutura de processamento embarcada deve ser capaz de executar todo o processamento necessário para a operação autônoma do dirigível, bem como prever a evolução gradual do nível de autonomia do dirigível robótico, na qual será incrementada progressivamente sua capacidade de percepção e tomada de decisão, utilizando diversos sensores, inclusive de

visão. Neste contexto, a estação embarcada deve prover uma elevada capacidade de processamento.

A CPU adotada como principal componente do sistema embarcado dedicado, mostrada na Figura A1.15 (ver também na Figura A1.12 o indicador 7), é do tipo PC-104 (PC/104, 2001), padrão amplamente utilizado em sistemas industriais embarcados. Cada módulo PC-104 possui dimensões reduzidas (3,6 x 3,8 polegadas), baixo consumo de energia e um barramento vertical que permite que os módulos possam ser empilhados, reduzindo o espaço ocupado. A solução utilizada no dirigível é composta pelos seguintes módulos:

- CPU MOPS lcd5 (JUMPTEC, 2001): possui processador Pentium 133 MHz, 32 Mbytes de RAM, 2 portas seriais, 1 interface paralela EPP, interface de rede 10Base-T Ethernet, interface para teclado, interface IDE.
- Placa Multiserial EMERALD - MM (DIAMOND, 2001): com 4 portas seriais que podem ser do tipo RS-232 ou RS-485. Como a maioria dos sensores possui saídas digitais via linha serial RS-232, optou-se por utilizar essa placa para realizar o interfaceamento entre a CPU e alguns sensores.
- Fonte de Alimentação V104 (TRI-M, 2001) que a partir de uma fonte de 12V (uma bateria comum), gera alimentação para o conjunto dos módulos PC-104.

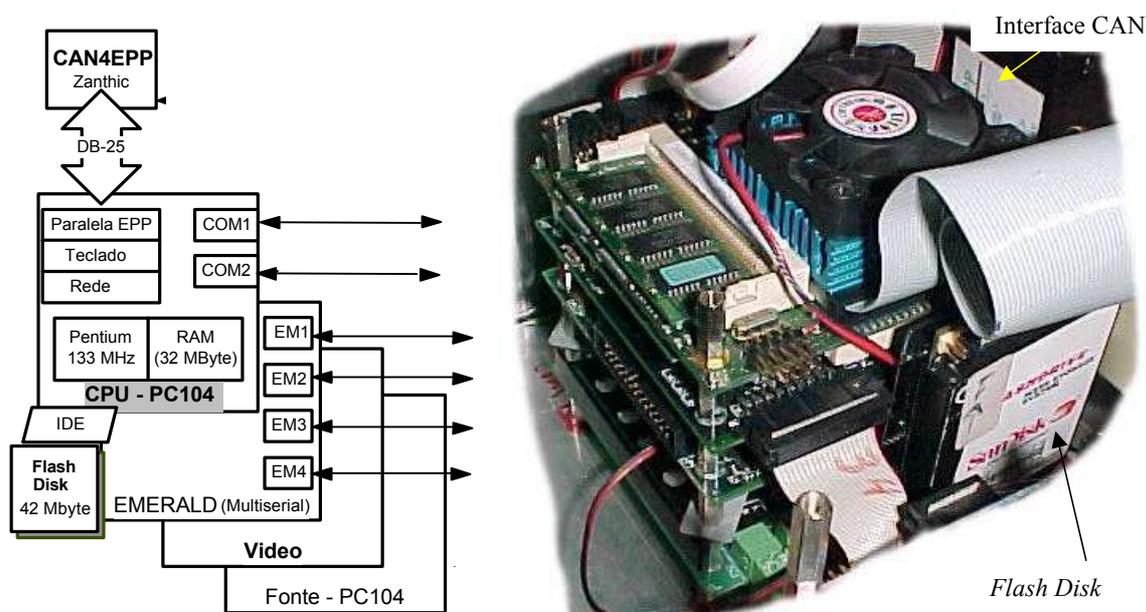


Figura A1.15: Sistema PC104 embarcado.

- Adaptador de Vídeo VGA PCM-3510: A placa de vídeo tem a função de permitir que o sistema possa ser configurado e testado em terra, pois,

conectando-se um monitor e um teclado à CPU embarcada, pode-se utilizar o sistema como um PC normal. Através da conexão Ethernet, pode-se ligar o PC em uma rede e realizar a transferência de novos módulos de *software*.

- Unidade de disco tipo *flash-disk*: Como o sistema embarcado é sujeito a vibrações e a possíveis choques mecânicos, a utilização de um disco rígido comum seria muito problemática. Um disco rígido comum também possui um consumo de energia razoável o que provocaria uma diminuição da autonomia das baterias existentes no sistema. Para resolver esse problema, adotou-se um *flash disk* (SAN, 2001) com 42 Mbytes de capacidade onde foi instalado um núcleo mínimo do sistema operacional Linux que será descrito posteriormente. Esse disco de estado sólido possui tolerância à vibração, além de apresentar menor consumo de energia.
- Interface para CAN – CAN4EPP: O CAN (CAN, 2001) é um protocolo para comunicação serial que suporta, de forma eficiente, controle distribuído de tempo real com um grau elevado de segurança. É largamente utilizado em sistemas automotivos devido ao seu baixo custo, elevada capacidade de transmissão (cerca de 1 Mbit/s) e taxa reduzida de erros. Uma rede CAN permite que um número muito grande de nós possam ser inseridos em uma mesma linha serial, facilitando assim a expansão do sistema embarcado pois não há mais a limitação do número de portas seriais para se realizar a conexão de novos sensores ou outros componentes (FARIA, 2001). A interface com a rede CAN é realizada pelo módulo CAN4EPP, mostrado na Figura A1.16 que é conectado via porta paralela.

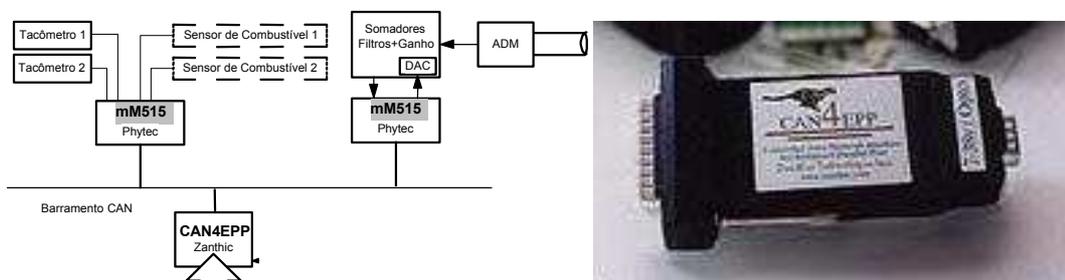


Figura A1.16: CAN4EPP da Zanthic.

### A1.3.1.3 Microcontrolador - MCC

O micro-controlador utilizado é o Infineon 80C517 (SIEM517, 1994) mostrado na Figura A1.17 (ver também na Figura A1.12 o indicador 4), cujos *hardware* e *software* foram desenvolvidos para atender as necessidades do projeto, desempenhando as seguintes funções:

- realizar medições de sinais analógicos de sensores e convertê-los em formato digital. Para isso possui conversor AD de 10 bits;
- adquirir e medir sinais PWM do receptor de rádio controle;
- gerar sinais PWM para comandar os servos;
- prover a comunicação com o PC embarcado através de um interface serial RS-232;
- enviar medidas de sinais analógicos e PWM (rádio controle) e receber dados para geração de sinais PWM (comandos para servos).

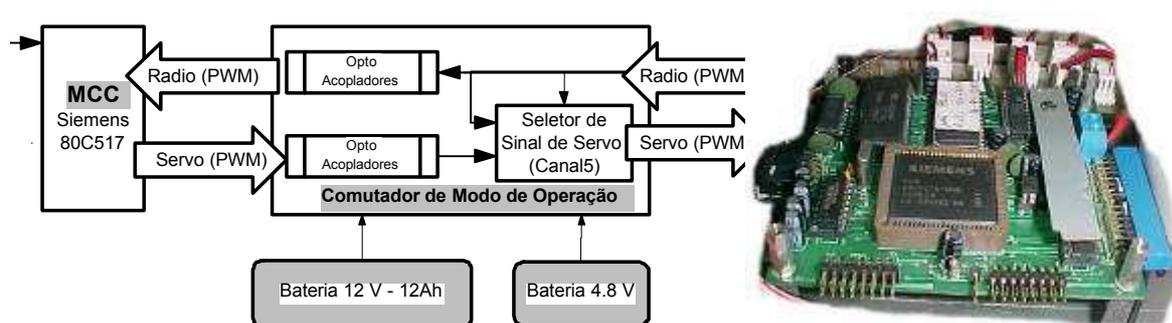


Figura A1.17: Micro-controlador - MCC.

### A1.3.1.4 Módulo 515C

O *MiniModul 515C* (MM515C, 2001) é o módulo básico através do qual os elementos sensoriais são integrados à rede CAN (ver Figura A1.18 e ver também na Figura A1.12 os indicadores 1 e 2). Ele possui interface CAN, e está sendo utilizado como interface para a sonda de vento, para os sensores de rotação e os sensores de nível de combustível, detalhados posteriormente. Este módulo possui um micro-controlador Infineon 80C515 de 8 bits (SIEM515, 1997) que provê a interface CAN e ainda um conversor AD de 10 bits, uma interface RS-232, portas de interrupção externa, *timers* e portas digitais.

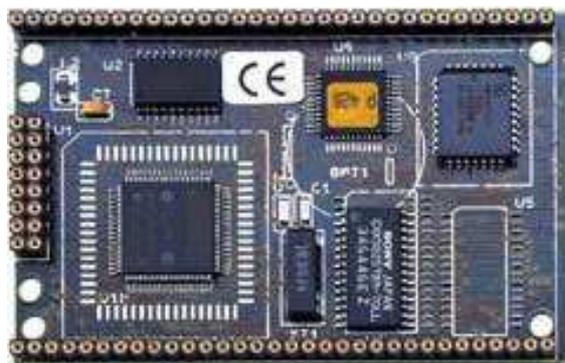


Figura A1.18: MiniModul 515C da Phytec.

#### A1.3.1.5 Comutador de Modo de Operação

O dirigível foi construído originalmente como veículo de aeromodelismo rádio-controlado, sendo conduzido pelo piloto em terra, o que é denominado modo manual de operação. Como veículo robótico, ele deve ser operado automaticamente pelo sistema embarcado, mas mantendo a possibilidade de operação manual por razões de segurança. É necessário, portanto que o dirigível possa operar nesses dois modos, automático ou manual, e assim, foi desenvolvido um dispositivo de *hardware* para comutação de modo de operação.

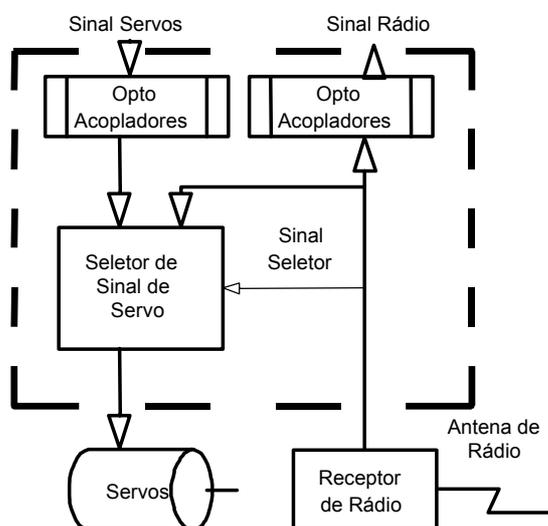


Figura A1.19: Diagrama de operação do comutador de modo de operação e sua caixa.

Esse comutador de modo de operação, mostrado na Figura A1.19 (ver também na Figura A1.12 o indicador 9), permite selecionar que os sinais enviados aos atuadores do dirigível sejam originários tanto da CPU embarcada (no modo automático), quanto do rádio controle (no modo manual). Este comutador age como sistema de segurança, pois caso ocorra uma falha na CPU, um comportamento inadequado dos algoritmos de controle

e navegação, ou mesmo para realizar uma transição entre modo de pilotagem manual e automático, o piloto do dirigível, que sempre está presente nos ensaios, pode comandar o chaveamento.

Este circuito também desacopla eletricamente a CPU e o rádio controle através de opto-acopladores, impedindo que panes do sistema elétrico da CPU possam causar danos sobre o receptor de rádio-controle. O conjunto CPU e sensores é alimentado por uma bateria diferente da que alimenta o conjunto receptor de rádio e servos, permitindo uma forma de controle mesmo em caso de falha elétrica no sistema que alimenta a CPU.

#### ***A1.3.1.6 Elementos de Comunicação***

Para comunicação entre os diferentes processadores são utilizadas conexões de rádio, linha serial, rede CAN, e conexões via rede local a nível físico e protocolo TCP/IP em nível de aplicação (GRIFFIN, 1998).



Figura A1.20: Rádio Modem DGR-115.

A conexão via rádio entre o sistema embarcado e a estação de terra é composta de dois enlaces: um para envio de imagens de vídeo geradas por uma câmera a bordo, e outro para transmissão de dados de telemetria e de operação. O sistema para transmissão de dados entre a CPU de terra e a CPU embarcada é composto por um par de rádio-modems tipo frequência espalhada DGR-115 (FREEWAVE, 2001) – ver Figura A1.20 (ver também na Figura A1.12 os indicadores 3 e 11), operando em uma frequência de 902-928 Mhz, com uma taxa de transmissão de 115,2 Kbps. Estes rádio-modems são conectados às interfaces seriais RS-232 do computador embarcado e da estação de terra. Quando conectados, os rádios fornecem um enlace de comunicação que não se distingue de uma linha serial comum. Os rádios possuem internamente um mecanismo de detecção de erro utilizando CRC (verificação por redundância cíclica - do Inglês *cyclic redundancy code*) e realizam a retransmissão de pacotes.

Uma forma de comunicação alternativa seria obtida utilizando rádio Ethernet (PROXIM, 2001), que possibilitaria a comunicação como se fosse uma interface física de rede local. No entanto esta alternativa se limita a um alcance tipicamente na faixa de 1 Km, enquanto que o alcance do rádio modem é muito maior, chegando a cerca de 30 km (com visada direta e utilizando as antenas especificadas pelo fabricante). Este maior alcance determinou a adoção da solução rádio modem.

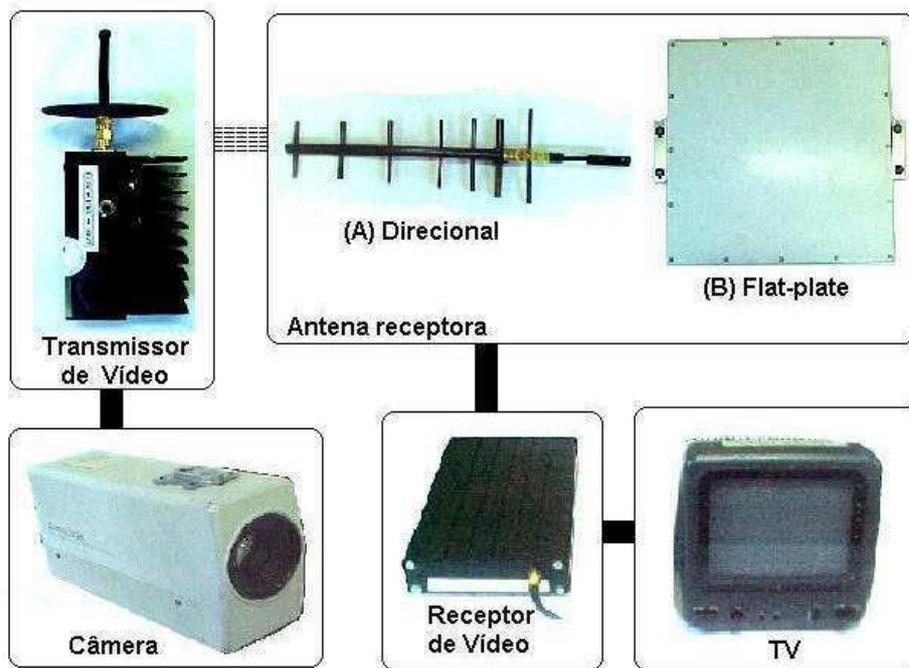


Figura A1.21: Enlace de vídeo.

O sistema de vídeo utilizado é composto por uma câmera CCD (do Inglês *Charged Coupled Device*), um transmissor e um receptor com antena do tipo *flat plate* que permite um desempenho superior. Este sistema foi fornecido pela PULNIX (2001) (Figura A1.21) (MAETA, 2001)

Compõem ainda o nível físico de comunicação as interfaces tipo RS-232 montadas na estação embarcada e na estação de terra, interfaces e periféricos conectados na rede CAN da estação embarcada e dispositivos utilizando conexões via protocolo TC/IP

### A1.3.2 Atuadores

Conforme descrito na Seção A1.3.1.5, um dispositivo de comutação permite a operação do dirigível no modo manual, recebendo comandos gerados pelo piloto em terra

diretamente do receptor do rádio controle ou, então, no modo automático onde os comandos recebidos através do módulo MCC são gerados na estação embarcada.

Em ambos casos, são gerados sinais PWM que definem os ângulos de posicionamento dos servo controles que comandam os atuadores do dirigível, ilustrados na Figura A1.22 (ver também na Figura A1.12 próximo ao indicador 9):

- Motores de combustão: dois servos (um para cada motor) controlando a velocidade de rotação dos motores (regula a abertura do carburador).
- Vetorização: um servo que atua na variação angular do sistema de vetorização;
- Superfícies Aerodinâmicas: quatro servos (um para cada superfície aerodinâmica) acoplados dois a dois. A deflexão das superfícies é de  $-30$  a  $+30$  graus.
- Válvula de Escape de Hélio: um servo atuando sobre uma válvula (aberta/fechada), que está montada na parte superior do envelope (ver a Figura A1.3), tendo como função a liberação de hélio em caso de necessidade emergencial.

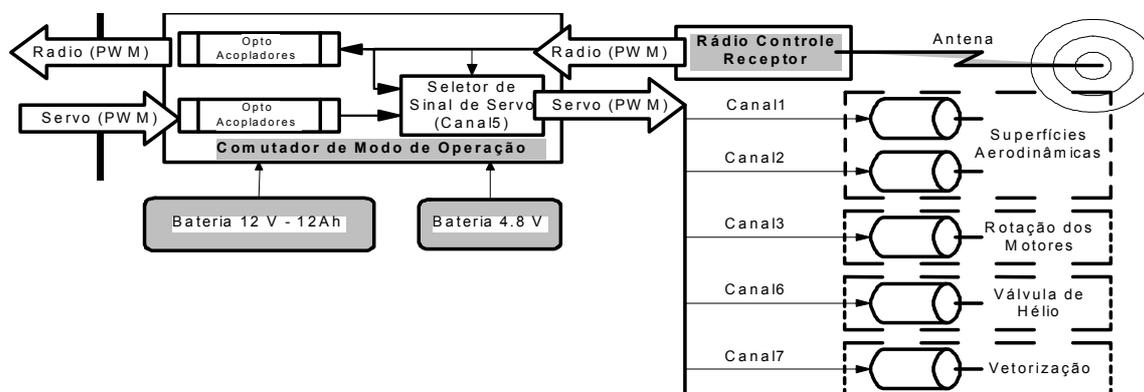


Figura A1.22: Interface com atuadores.

### A1.3.3 Sensores

Os sensores têm como principal função permitir a operação autônoma do dirigível e, neste sentido, devem propiciar informações de posição, velocidade e atitude (PVA) ao longo do tempo. Essas informações constituem as variáveis básicas para o sistema de controle e navegação do veículo. Outros sensores são também necessários, seja para fornecer variáveis utilizadas nas malhas de controle, seja para prover informações a respeito do estado do dirigível durante a sua operação.

Para determinação da PVA do dirigível, os sensores utilizados são: GPS com sua estação de base, giro-inclinômetros, acelerômetros, bússola, sonda de vento e tacômetros. A Figura A1.12 mostra o diagrama de conexão desses componentes da estação embarcada.

#### ***A1.3.3.1 GPS e Sistema de Correção Diferencial***

O sistema de GPS utilizado, mostrado na Figura A1.23 (ver também na Figura A1.12 os indicadores 8 e 14), é o SvecSix (TRIMBLE, 2001), composto por um conjunto de placa com saída serial digital e uma antena receptora. O sistema de GPS capta sinais de satélites e determina a posição do receptor nas três dimensões: latitude, longitude e altitude, mais as velocidades inerciais em três eixos. A precisão do sinal obtido pelo sensor de GPS embarcado, antes da interrupção do sinal de erro propositadamente introduzido pelo Departamento de Defesa americano, encontrava-se na faixa de 50 metros em latitude e longitude, e de 100 metros em altitude. O uso de um outro GPS fixo, conectado à estação de terra, enviando sinais de correção ao GPS embarcado, para que este operasse no chamado modo diferencial, permitia o aumento da precisão para cerca de 5 metros para dados de latitude e longitude, e 10 metros para a altitude.

Após a interrupção do sinal de erro acima referido, em maio de 2000 (FURUNO, 2001), tem-se observado uma melhora na desempenho do receptor de GPS, tendo sido obtida uma precisão na faixa de 5m para latitude e longitude, em modo não diferencial, e 3m em modo diferencial. Segundo FURUNO (2001), com o fim deste sinal a variância do erro de posição cai para 30m sem o DGPS e para 2 a 3m com o DGPS.

Sistemas de GPS mais modernos, que medem a diferença de fase dos sinais recebidos (*carrier phase*), conseguem atingir precisão da ordem de alguns centímetros.



Figura A1.23: GPS embarcado e sua interface.

Nome do Sinal	Unidade	Precisão
Longitude	graus	5 m
Latitude	graus	5 m
Altitude	m	10 m
Velocidade eixo X [positivo para leste]	m/s	-
Velocidade eixo Y [positivo para norte]	m/s	-
Velocidade eixo Z [positivo para cima]	m/s	-

Tabela A1.1: Medidas realizadas pelo GPS.

O módulo de GPS utilizado opera com frequência de amostragem máxima de 1 Hz, e suas principais medidas estão mostradas na Tabela A1.1.

### A1.3.3.2 *Bússola Eletrônica e Inclinômetro TCM2*

O módulo TCM2 da Precision Navigation (PRECISION, 2001), mostrado na Figura A1.24 (ver também na Figura A1.12 próximo ao indicador 6), fornece dados de atitude (ângulos de rolagem, arfagem e guinada) e temperatura, através de uma interface serial RS-232. Os dados de inclinação são obtidos através de um sensor de nível de bolha, que utiliza um princípio de medição de capacitância para a determinação da atitude. O valor de guinada (bússola) é obtido através de um conjunto de magnetômetros dispostos em três eixos ortogonais. A medida da bússola é corrigida internamente utilizando dados do inclinômetro.

O equipamento embarcado, principalmente a CPU, gera campos elétricos e magnéticos que não são desprezíveis e que afetam as medidas efetuadas pelos magnetômetros do TCM. Mesmo realizando a calibração do equipamento para compensar esses efeitos, o erro apresentado nas medidas é superior ao indicado pelo fabricante.



Figura A1.24: Bússola eletrônica e inclinômetro TCM2.

O TCM pode operar a uma frequência máxima de 30 Hz, podendo oferecer medidas brutas, ou processadas. A Tabela A1.2 apresenta as medidas processadas realizadas pelo

TCM.

Nome do Sinal	Unidade	Precisão
Ângulo de Rolagem [-50 a +50 graus]	graus	0,4 grau
Ângulo de arfagem [-50 a +50 graus]	graus	0,4 grau
Ângulo de Guinada [0 a 360 graus]	graus	1 grau
Temperatura [-20 a 70 graus]	graus celsius	-

Tabela A1.2: Medidas Realizadas pelo TCM.

### A1.3.3.3 Central Inercial - Giro-inclinômetro de Estado Sólido

O giro-inclinômetro (ver na Figura A1.12 o indicador 4 ou na Figura A1.13), ou central inercial DMU-AHRS (DMU, 1999) fornece via interface RS-232 as medidas de atitude (rolagem, arfagem e guinada), velocidades angulares em três eixos ortogonais, e acelerações lineares nestes mesmos eixos. Esta central inercial utiliza como sensores três magnetômetros, três acelerômetros e três giroscópios, e incorpora um DSP de alto desempenho para fornecer as medidas corrigidas. Os elementos sensoriais são do tipo *micro-machined device*, o que lhes confere uma grande durabilidade e resistência, não necessitando de calibração.

As medidas da DMU-AHRS e do TCM são similares mas baseiam-se em diferentes princípios, pelo menos para as medidas de rolagem e arfagem, um utilizando informação de giroscópios e acelerômetros e outro nível de bolha. Serão realizadas algumas avaliações para se decidir o abandono definitivo do TCM como fonte das informações de atitude.

A Central Inercial DMU-AHRS pode operar a uma frequência máxima de 30 Hz, podendo oferecer medidas brutas, ou processadas. No caso de medidas processadas a Tabela A1.3 apresenta aquelas disponibilizadas pelo sensor.

Nome do Sinal	Unidade	Precisão
Angulo de Rolagem (em torno do eixo X) [-90 a +90 graus]	graus	1 grau
Angulo de Arfagem (em torno do eixo Y) [-90 a +90 graus]	graus	1 grau
Angulo de Guinada (em torno do eixo Z) [-180 a +180 graus]	graus	1 grau
Velocidade de rolagem	graus/s	-
Velocidade de arfagem	graus/s	-
Velocidade de guinada	graus/s	-
Aceleração X	m/s <sup>2</sup>	-
Aceleração Y	m/s <sup>2</sup>	-
Aceleração Z	m/s <sup>2</sup>	-

Tabela A1.3: Medidas processadas realizadas pelo DMU-AHRS.

#### A1.3.3.4 Sonda de Vento

A sonda de vento ADM - do Inglês *air data measurement* - (AZINHEIRA *et al.*, 1999), vista na Figura A1.25 (ver também na Figura A1.12 indicador 2), é utilizada para realizar a medida de velocidade relativa de vento sobre o dirigível nos três eixos e também da altitude através da pressão atmosférica (altímetro barométrico). Os sensores utilizados realizam a medição de pressão do ar em diferentes pontos no corpo da sonda. A partir desses valores pode-se calcular as velocidades relativas que atuam sobre o dirigível.

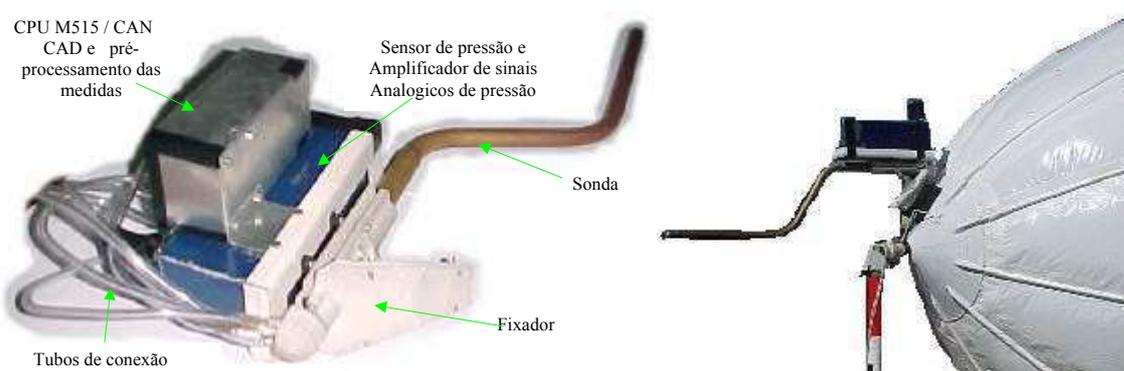


Figura A1.25: Sonda de vento ADM.

Como a velocidade de operação do dirigível é baixa (menos que 50 km/h) e em geral as sondas de vento existentes destinam-se a aviões que operam em uma faixa de velocidades muito maior, a sonda de vento utilizada precisou ser construída e calibrada para esse propósito específico. Ela foi desenvolvida pela equipe do Dr. José Raul Azinheira do Instituto Superior Técnico de Lisboa, Portugal.

A sonda está posicionada junto do bico de atracção, na extremidade frontal do dirigível, a 5m de distância da CPU embarcada localizada na gôndola. Os sinais da sonda de vento são analógicos e, tendo em vista problemas de ruídos e de peso adicional em cabeamento devido à distância da sonda à eletrônica embarcada, tornou-se necessário o desenvolvimento e montagem de uma interface próxima à sonda para a conversão AD destes sinais. Utilizou-se um módulo micro-controlador com interface CAN. Este micro-controlador, por possuir conversor analógico-digital de 10 bits, resolução menor que a necessária que era de no mínimo 12 bits, requereu o desenvolvimento de um circuito adicional de pré-amplificação para se obter a resolução adequada para o sinal de pressão absoluta. O esquema de conexão é ilustrado na Figura A1.26.

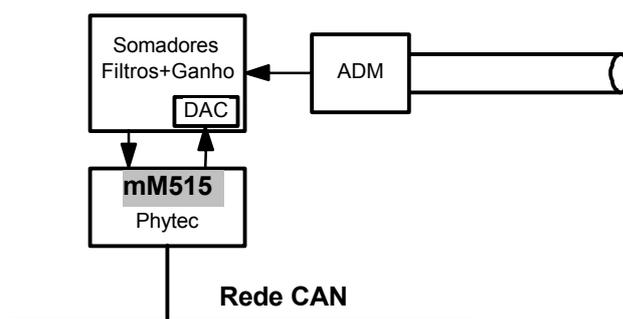


Figura A1.26: Interface com a sonda de vento.

A sonda de vento opera a uma frequência programável máxima até na ordem de 30 Hz, oferecendo medidas brutas, ou seja, um valor digital correspondente à medida do sensor, necessitando um processamento *a posteriori*. A Tabela A1.4 resume suas medidas após o processamento.

Nome do Sinal	Unidade	Precisão
Ângulo de ataque vertical	graus	1 grau
Ângulo de ataque lateral	graus	1 grau
Pressão dinâmica (velocidade do vento)	mmHg	-
Pressão estática (altitude barométrica)	mmHg	-

Tabela A1.4: Medidas realizadas pela Sonda de Vento.

#### A1.3.3.5 Sensores para Monitoramento do Estado do Dirigível

Outros sensores (FRAZZATO, 2001) são necessários tanto para prover informações adicionais para o sistema de controle quanto para realizar a monitoração de seu *status*, i.e., verificar o correto funcionamento dos dispositivos mecânicos, sua integridade estrutural e capacidade de voo nos casos onde o dirigível está fora do alcance visual da equipe em terra.

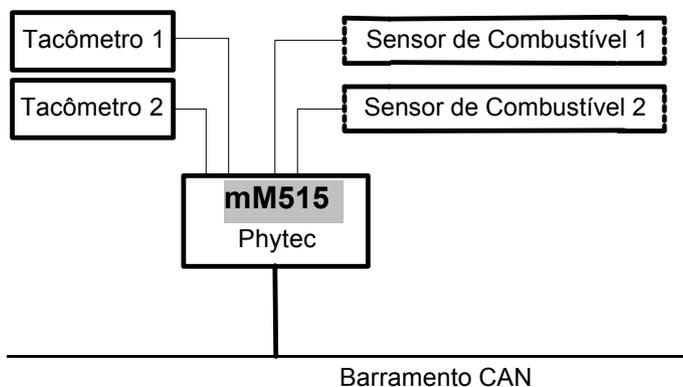


Figura A1.27: Conexão via rede CAN dos sensores adicionais.

São necessários dispositivos apropriados para a medição de: velocidade de rotação das hélices, nível de combustível, ângulo de vetorização do sistema de propulsão, carga das baterias e pressão interna do envelope do dirigível. Destes, foram desenvolvidos os tacômetros e sensores de nível de combustível (um para cada tanque), conectados através de um micro-controlador tipo M515 via o barramento CAN do dirigível, conforme mostra a Figura A1.27 (ver também na Figura A1.12 o indicador 1).

Os tacômetros mostrados na Figura A1.28-a utilizam princípio ótico, baseando-se na interrupção de fonte de luz infra-vermelha pelas pás da hélice. A medição de nível de combustível, Figura A1.28-b, utiliza um sensor baseado no princípio capacitivo.

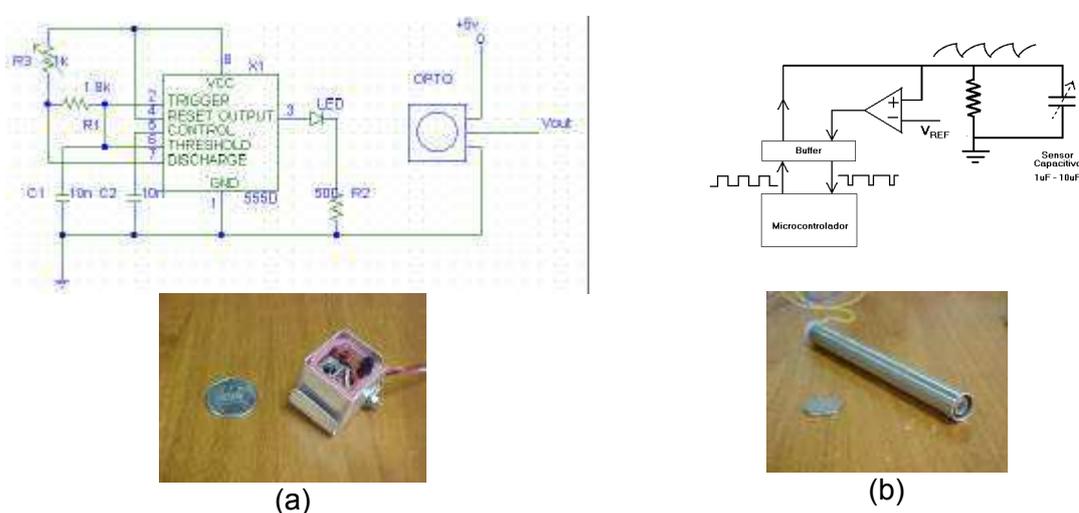


Figura A1.28: Esquemas do tacômetro e sensor de nível de combustível.

## A1.4 Software

O objetivo desta seção é apresentar os elementos de infra-estrutura de *software* incluindo o Linux tempo real e o arcabouço do *software* para os sistema de controle e navegação embarcados bem como para a estação de operação em terra.

### A1.4.1 O Linux Tempo Real

Muitas alternativas de sistemas tempo real estão disponíveis no mercado. Em termos de sistemas abertos, ultimamente têm se destacado os chamados Linux\_tempo\_real (LINUX-RT, 2001; RTAI, 2001), que podem representar uma alternativa a alguns sistemas comerciais. Entre os sistemas comerciais, cita-se: Chorus (CHR, 2001), Lynx (LYNX, 2001), QNX (QNX, 2001), Vx-Works (VXWORKS, 2001).

Conforme explicitado, há sistemas comerciais para se assegurar o atendimento de requisitos de tempo real. Outra alternativa são sistemas abertos. O Linux tempo real (LTR) foi escolhido como o software para se implementar as tarefas com requisitos de tempo real. O principal motivo da sua escolha foi o de gerar conhecimento e ao mesmo tempo avaliar a viabilidade do uso e o desempenho de um sistema operacional de tempo real que segue os princípios da fundação para o *software* livre (do inglês *free software foundation*) (FREESOFT, 2001). Como atividade prévia à adoção do uso do LTR, foi feito um levantamento da existência de aplicações análogas, que também apresentassem restrições de tempo real, como controle de máquinas ferramenta (EHMC, 2001, LME, 2001) e um laboratório para o desenvolvimento de sistemas de controle em tempo real (RTCL, 2001) com inúmeros aplicativos já desenvolvidos. Também foram feitos alguns experimentos-piloto que mostraram a viabilidade do uso do LTR.

O sistema operacional Linux (LINUX, 2001) tem tido grande disseminação nos últimos anos e tem representado uma alternativa, com código fonte aberto, aos sistemas operacionais com código fechado e proprietários. Comunidades trabalhando nos diferentes componentes desse sistema, e utilizando a Internet como meio de interação, têm propiciado um grande desenvolvimento ao Linux, dando seguimento aos ideais da fundação para o *software* livre.

Uma das limitações do Linux reside no fato deste não ser *hard real time* system (em português sistemas com restrições severas de tempo) isto é, apesar de ser multitarefa, ele não garante o atendimento dos requisitos de tempo para a execução de tarefas; por exemplo a operação de chaveamento de contexto (*swap*) pode prejudicar o atendimento da interrupção de um periférico. Em aplicações de controle a perda de um requisito de tempo pode ser catastrófica (OVRTL, 2001). Uma alternativa é a adoção dos chamados “Linux de tempo real” (em Inglês *real time Linux*).

Atualmente há duas versões disponíveis para o real time Linux, o *Real Time Application Interface* (RTAI) desenvolvido pelo *Departimento di Ingegneria Aerospaziale* do *Politecnico di Milano* (RTAI, 2001), e o RTLinux desenvolvido pelo *New Mexico Institute of Technology* (RTLINUX, 2001). O RTLinux foi o primeiro Linux de tempo real desenvolvido, mas a falta de adaptação para as novas versões do Kernel Linux, principalmente a transição da versão 2.0 (monoprocessador) para 2.2 (multiprocessadores)

e o não acordo do pessoal do New Mexico Institute of Technology, hoje na empresa FSMLABS, para desenvolvimento conjunto da plataforma de linux tempo real, conduziu ao surgimento do RTAI, segundo seus próprios desenvolvedores (RTAI, 2001b).

A solução adotada na maioria das versões de Linux tempo real, atualmente disponíveis, foi o desenvolvimento de um executivo de tempo real, que opera com maior prioridade e então fazer com que as outras tarefas do Linux sejam tarefas deste executivo. Na opinião dos desenvolvedores do RTAI a sua modificação no *kernel* é menos invasiva que a modificação feita pelo RTLinux.

Para o dirigível adotou-se o RTLinux versão desenvolvido pela New Mexico Institute of Technology. A principal razão desta adoção foi que esta era a única opção disponível por volta de 1998, apesar de cada vez mais seus criadores estarem se desviando dos objetivos da *Free Software Foundation*. Atualmente o RTLinux continua a ser usado por questões de compatibilidade.

As características gerais do RTLinux segundo seu criador (RTLINUX, 2001b) foram baseadas nas premissas de que componentes de *hard real time* não são compatíveis com alocação dinâmica de recursos, sincronizações complexas ou qualquer recurso que introduza atrasos. A premissa utilizada no seu desenvolvimento é se um serviço ou uma operação não possui características de tempo real, ela deve ser provida em Linux e não em componentes do RTLinux.

A configuração mais utilizada de RTLinux oferece *tasks* primitivas com memória alocada estaticamente, sem proteção no espaço de memória, com um escalonador que utiliza prioridade fixa, sem nenhuma proteção contra escalonamentos impossíveis e a desabilitação de interrupções por *hardware*. Essa configuração usa memória compartilhada como a única primitiva de sincronização entre tarefas de tempo real. Como mecanismo de comunicação entre as tarefas de tempo real e tarefas normais, o RTLinux oferece filas chamadas de “*real time FIFOS*” –RTFifo. As RTFifos correspondem a dispositivos de entrada e saída que são abertos pelos comandos de manipulação de dispositivos de entrada e saída do Unix. Na medida que o usuário incrementa seus requisitos, como o *kernel* foi projetado com módulos substituíveis, ele pode acrescentar módulos que implementem as características requeridas, como por exemplo: diferentes tipos de escalonadores de tarefas, módulos que implementam semáforos, memória comum, comunicação serial.

#### ***A1.4.1.1 Distribuição Compacta do Linux***

O crescimento de aplicações desenvolvidas em Linux embarcadas ou embutidas (do inglês *embedded*) em sistemas com pequena capacidade de memória e disco, motivou a comunidade do Linux a desenvolver distribuições deste extremamente compactas. O projeto do roteador Linux (ROUTER, 2001) desenvolveu uma distribuição que ocupa um disquete de 3 polegadas e meia. Esse sistema, quando se carrega, é executado ocupando 4 Megabytes dos chamados RAMDISK, ou seja, memórias que simulam disco. No presente trabalho, inicialmente, uma versão reduzida do RTLinux foi gerada a partir de modificações da distribuição do projeto roteador, e da recompilação deste. Atualmente já são disponíveis versões compiladas do RTLinux baseadas no projeto Linux Router como em RTLINUX (2001c) e estas vêm sendo usadas na evolução do *software* do dirigível.

No dirigível, os arquivos do sistema operacional são carregados na estação embarcada a partir do *flash-disk*, mas sua execução se dá na RAMDISK. O motivo do uso de RAMDISK em vez de *flash-disk* como memória temporária, deve-se ao fato que esse tipo de dispositivo tem um limite máximo de escrita de aproximadamente 1 milhão de vezes, o que é um número perfeitamente viável de ser alcançado no caso da execução de um sistema operacional.

#### ***A1.4.2 Arcabouço do Software***

Já foram apresentadas as características gerais dos chamados Linux tempo real e justificada a adoção do RTLinux pela necessidade de garantir a execução de tarefas críticas no tempo. Também foi dito que os recursos de comunicação entre aplicativos executando a nível de usuário e a nível de RTLinux são as chamadas RTFifos ou memória compartilhada. Há ainda outros limitantes em relação a estes mecanismos de comunicação: as RTFifos só podem ser lidas por um processo, gerando problemas quando há vários aplicativos que requerem o mesmo conjunto de dados, e o uso de memória comum requer o uso de mecanismos similares aos de sessão crítica, para garantir a integridade dos dados.

Considerando-se os fatores limitantes descritos anteriormente, decidiu-se definir um arcabouço de *software* que estabelece claramente a interação entre processos de tempo real e processos sendo executados a nível de usuário, tanto no computador embarcado quanto no computador de terra.

O diagrama deste arcabouço é apresentado na Figura A1.29, onde o módulo escrito em RTLinux aparece contrastado. Este módulo utiliza uma das capacidades de seu escalonador, que é a de execução de uma tarefa periódica, operando em sincronismo com um relógio. A cada período esta tarefa pode realizar as seguintes funções:

- i) *pooling* das mensagens chegadas via RTFifos;
- ii) leitura dos dados dos sensores;
- iii) cálculos associados aos algoritmos de controle e navegação;
- iv) envio dos comandos para os atuadores;
- v) envio de mensagens via RTFifos para outros aplicativos;

vi) envio de mensagens e de dados de telemetria (correspondentes a informações sobre a situação de vôo, medidas pelos sensores) da estação embarcada para a estação de terra, ou recebimento de comandos e outros dados da estação de terra para a estação embarcada.

Descreveu-se acima o caso da estação embarcada, já que a versão que opera na estação de terra é uma simplificação da estação embarcada.

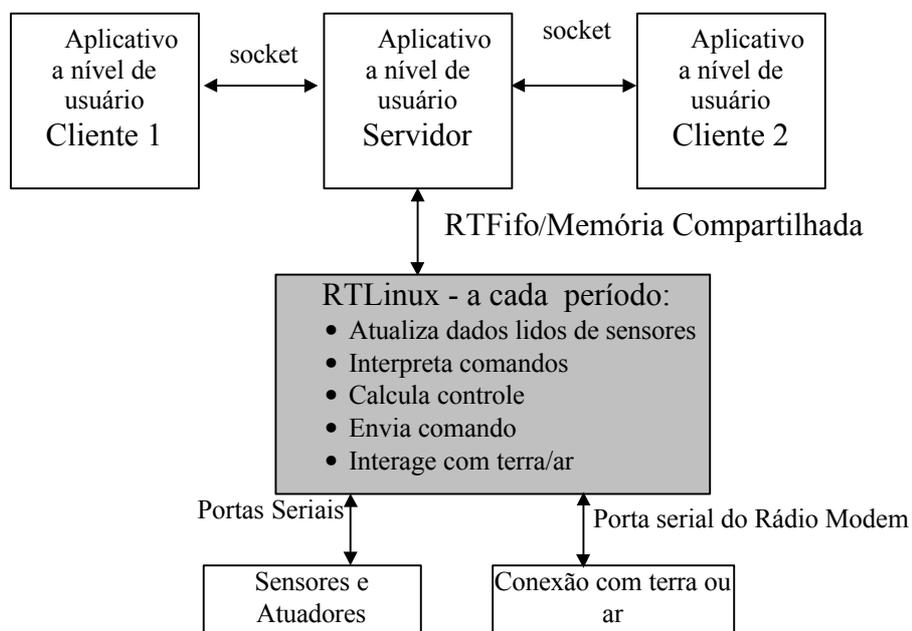


Figura A1.29: Arcabouço dos aplicativos RTLinux.

No caso de mais de um aplicativo em nível de usuário requerer um mesmo conjunto de dados do aplicativo escrito em RTLinux, decidiu-se pela criação de aplicativos

servidores que interagem com o aplicativo RTLinux. Este aplicativo interage através de RTFifos, obtendo os dados requeridos, repassando-os para os demais aplicativos via sockets do protocolo TCP/IP. Esta situação acontece com os dados de telemetria.

Este conjunto de soluções constitui a base de um ambiente integrado que congrega as infra-estruturas embarcada e em terra, suportando tanto a fase de desenvolvimento quanto o uso do dirigível em situações reais de aplicação; este ambiente é o detalhado no Capítulo 7.

## **A1.5 Conclusões**

Este anexo apresentou o conjunto dos recursos de base para dotar o dirigível de capacidade de operação autônoma.

A primeira Seção tratou do veículo propriamente dito e das evoluções construtivas que sofreu para poder satisfazer às necessidades de operação como veículo robótico e voar com segurança. Isto incluiu: a alteração de dimensões e posicionamento das superfícies aerodinâmicas, requeridas na segunda versão do envelope utilizada; a evolução no conjunto de propulsão, com substituição de motores e hélices até que se obtivesse um conjunto adequado; um novo sistema de vetorização e o desenvolvimento de uma nova gôndola que distribui e isola os elementos mecânicos de propulsão dos componentes eletrônicos embarcados, além de melhorar, também, as condições de acessibilidade ao conjunto embarcado.

A segunda Seção apresentou a infra-estrutura completa de *hardware* e comunicação concebida neste trabalho, detalhando seus componentes.

A terceira Seção descreveu o RTLinux e seu uso como sistema operacional de tempo real, e concluiu apresentando o arcabouço do software do dirigível robótico.

## ANEXO 2. APRIMORAMENTO DO SISTEMA DE PROPULSÃO DO DIRIGÍVEL AS800

Conforme visto no Anexo 1, no processo de aprimoramento do conjunto de motorização fornecido originalmente no AS800-1, a fim de encontrar um conjunto apropriado para as novas dimensões do AS800-2, foi necessário realizar uma avaliação teórica e prática do conjunto, isto é, dos motores e diferentes opções de hélices.

As características de propulsão de hélices podem ser obtidas em HEPERLE (2001). O usuário deve definir os perfis aerodinâmicos na raiz e na ponta da hélice, a partir de uma lista disponível de modelos padrão de hélices. Então, deve-se fornecer os valores da corda da hélice e ângulo de ataque (ângulo entre a corda e a referência horizontal) para diferentes raios, isto é, sucessivas seções transversais da raiz até a ponta da hélice, conforme mostrado na Figura A2.1.

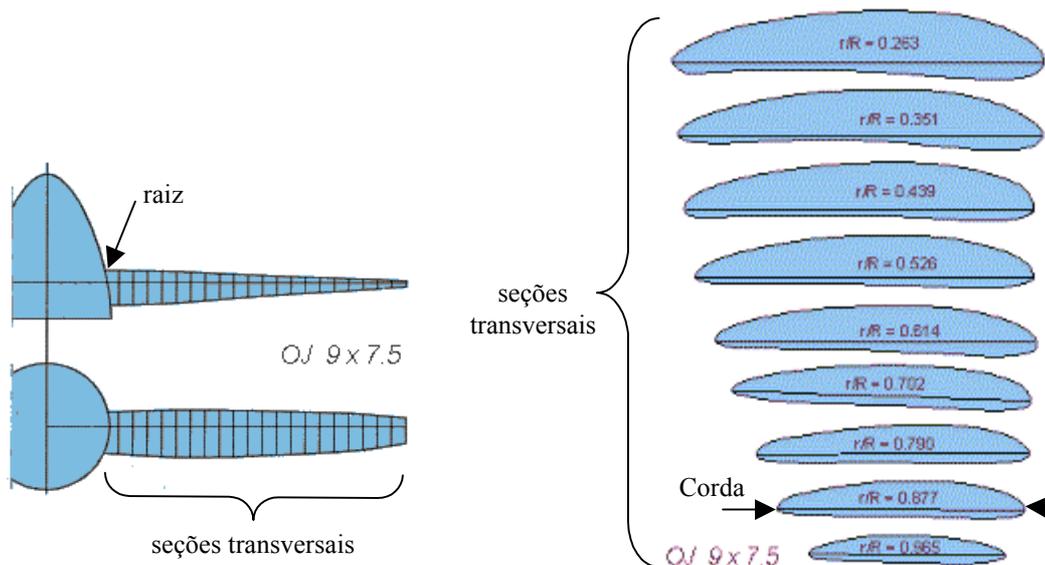


Figura A2.1: Corte da hélice e características dos diferentes perfis.

O levantamento das características dimensionais e de forma da hélice acima citadas, foram realizadas através de digitalização, por uma máquina de medir coordenadas 3D, dos perfis aerodinâmicos (para diferentes seções transversais). A Figura A2.2 mostra um perfil levantado com esta máquina para uma hélice real, tipo Bolley 13x10.

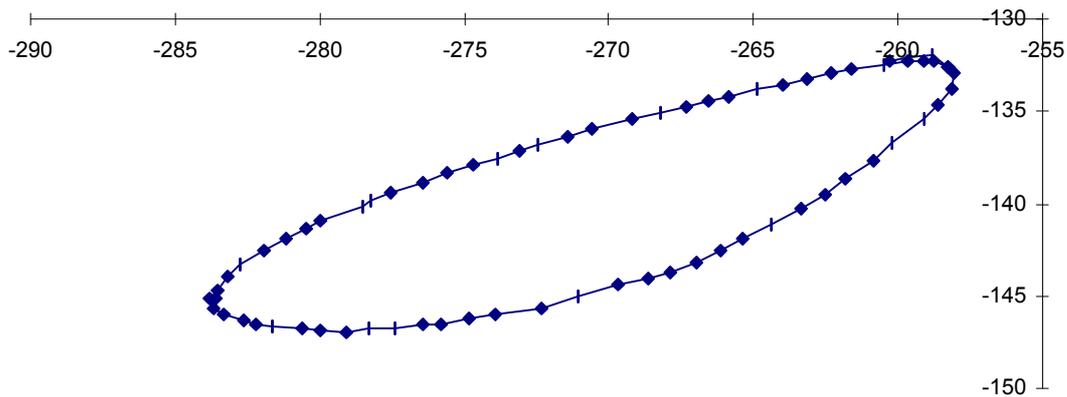


Figura A2.2: Exemplo de perfil de uma seção transversal de hélice levantado numa máquina de medição 3D

Com os dados da hélice fornecidos ao site (HEPPERLE, 2001), este disponibiliza curvas e tabelas com as características da hélice expressas através de parâmetros não dimensionais: coeficiente de propulsão ( $C_T$ ) e coeficiente de potência ( $C_P$ ), ilustrados respectivamente pelas Figura A2.3 "a" e "b"; ambos coeficientes são função da razão de avanço ( $\gamma$ ), isto é, da relação entre a velocidade à frente e velocidade da ponta da hélice, definida como:

$$\tilde{a} = \frac{v}{n D} \quad \text{A1.1}$$

Onde:

$v$  é a velocidade do veículo relativa ao ar (m/s),

$n$  é a velocidade de rotação da hélice (1/s),

$D$  é o diâmetro da hélice (m).

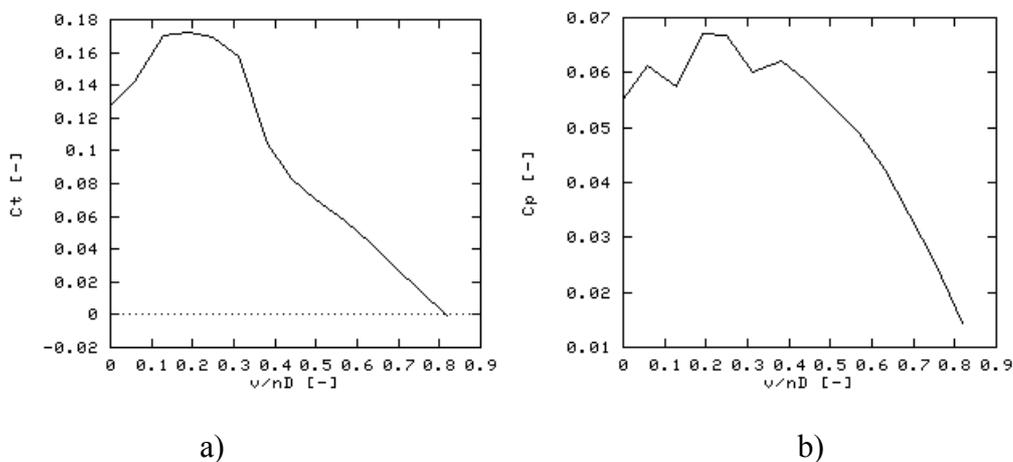


Figura A2.3: Curvas característica de  $C_T$  e  $C_P$  para hélices.

A Figura A2.3 mostra as curvas características para um conjunto de hélices utilizado no AS800.

A força de propulsão  $T$  (em N) e a potência  $P$  (em W), requerida para movimentar a hélice são obtidos respectivamente como:

$$T = C_T \rho n^2 D^4, \quad (\text{A1.2})$$

$$T = C_T \rho n^3 D^5, \quad (\text{A1.3})$$

onde  $\rho$  é a densidade do ar em  $\text{Kg/m}^3$ .

A eficiência da hélice é definida como:

$$\eta = \frac{v}{n D} \frac{C_T}{C_P}. \quad (\text{A1.4})$$

Para a confirmação experimental das características de propulsão das hélices foram realizados ensaios, utilizando-se o método do pêndulo com a gôndola real, isto é, medindo-se o deslocamento horizontal sofrido pela massa acionada pelos motores em diferentes rotações. A montagem é mostrada na Figura A2.4.

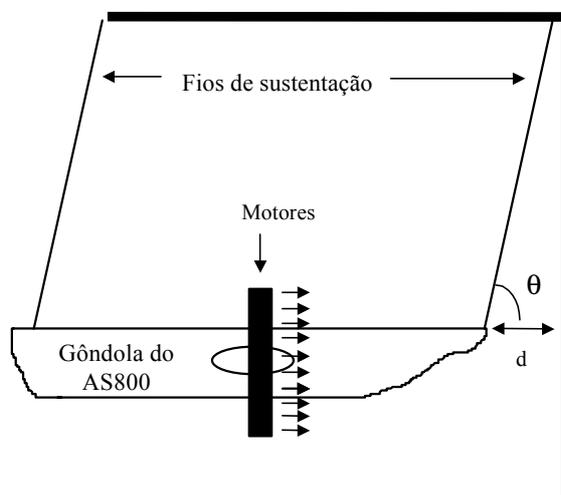


Figura A2.4: Ensaio da força de propulsão com o método do pêndulo.

Sabendo-se o peso da gôndola e medindo-se valores de “ $d$ ” para um conjunto de valores de rotação dos motores, é possível estimar a força exercida pelo motor como sendo a componente horizontal da tensão sobre ponto onde o fio de sustentação prende-se à gôndola.

A Tabela A2.1 mostra um exemplo de resultado deste ensaio, onde há uma comparação entre os resultados práticos e os teóricos para uma hélice tipo Boley, diâmetro de 12,5 polegadas e passo de 6 polegadas, deslocando uma massa de 8,5Kg e utilizando o motor OS61.

Rotação(rpm)	d (m)	Força Medida (N)	Força Teórica(N)	Medido/Teórico
11900	2,40	45,31	56,84	0,80
10500	1,96	31,34	44,25	0,71
8700	1,38	19,46	30,38	0,64
7700	1,10	14,93	23,80	0,63

Tabela A2.1: Resultados das medidas de força de propulsão segundo método do pêndulo.

Diversos ensaios foram realizados para diferentes tipos de hélice. Os erros tipicamente situaram-se na faixa de 20 a 40%. Deve-se chamar atenção que para a obtenção da força teórica, uma diferença de 1,6 mm (5%) no diâmetro da hélice, provoca uma diferença no valor da força teórica de 20%, pois o fator de conversão de  $C_T$  para  $T$  é proporcional à quarta potência do diâmetro. Assim, considerando-se a limitação na precisão das medidas, permite-se considerar como razoáveis os resultados obtidos para o modelo de propulsão da hélice. São previstos levantamentos mais precisos usando-se tacômetros e dinamômetro.

### **ANEXO 3.**

## **DETALHAMENTO DA RELAÇÃO O ENTRE O TRABALHO DE TESE E OUTROS TRABALHOS**

Como explicitado no capítulo introdutório desta dissertação, o trabalho desenvolvido nesta tese caracteriza-se por seu cunho predominantemente tecnológico, integrando diferentes domínios do conhecimento, no âmbito de um projeto institucional - o Projeto AURORA. Frizou-se também a importante participação, no AURORA, de um grupo de pesquisadores, estudantes de pós graduação e bolsistas de iniciação científica. A ação coordenada dos membros deste grupo no AURORA reflete-se também neste trabalho de tese, hora fornecendo-lhe subsídios fundamentais que, aqui integrados, permitem a criação de um dirigível robótico, hora dando continuidade e aprofundando os trabalhos aqui iniciados, contribuindo do mesmo modo para a criação de um dirigível robótico. O objetivo deste Anexo é caracterizar as interações entre este trabalho de tese e outros trabalhos que aqui são utilizados bem como aqueles que dele se originaram.

Além do autor desta tese, denotado por RAMOS, as pessoas que contribuíram tecnicamente no desenvolvimento dos trabalhos aqui apresentados são citadas a seguir; as respectivas filiações institucionais são explicitadas, excetuando-se aquelas relacionadas ao ex-CTI / CenPRA:

- Dr. Alberto Elfes, referido como ELFES;
- Dr. Marcel Bergerman, referido como BERGERMAN;
- Dr. Samuel Siqueira Bueno, referido como BUENO;
- Dr. Ely Carneiro de Paiva referido com DE-PAIVA;
- Dr. José Raul Azinheira, do IST de Lisboa, referido como AZINHEIRA;
- Dr. Reid Simmons da CMU, USA, referido como SIMMONS;
- Dr. Sérgio Varella Gomes, consultor, referido como GOMES;
- Dr. Augusto Bruciapaglia, do DAS-UFSC, referido com BRUCIAPAGLIA;
- Kioyshi ASANUMA, referido como ASANUMA;
- Silvio Mano Maeta, cujo mestrado foi desenvolvido no AURORA, referido como MAETA;
- Luiz Gustavo Bizarro Mirisola, cujo mestrado foi desenvolvido no AURORA, referido como MIRISOLA;

- Bruno Guedes Faria, cujo mestrado encontra-se em desenvolvimento no AURORA, referido como FARIA;
- Os bolsistas de Iniciação Científica Fabio Augusto Guimarães, referido como GUIMARÃES, Luiz Gustavo Nascimento, referido como NASCIMENTO, e Rodrigo Paniago Peixoto, referido como PEIXOTO.

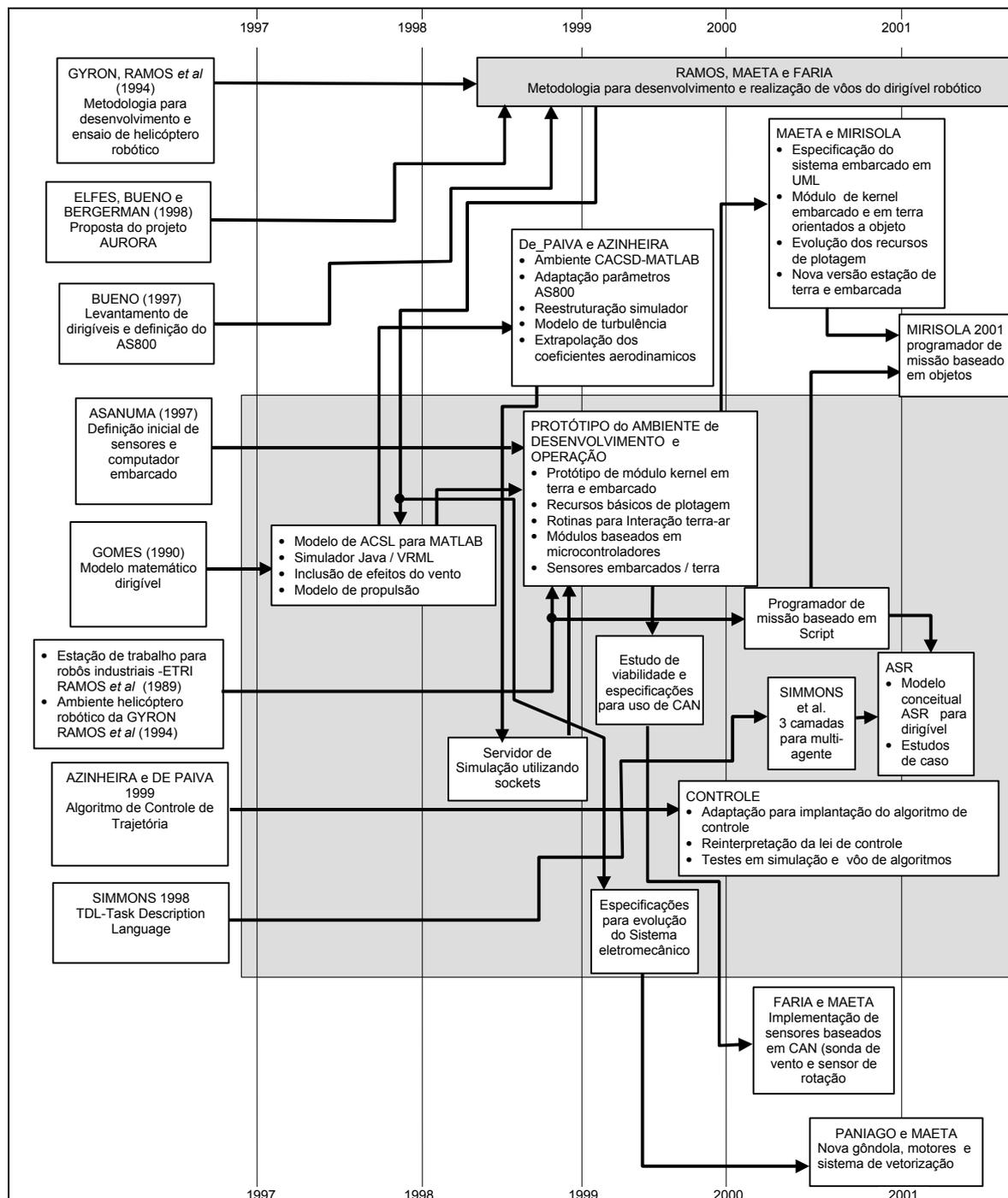


Figura A3.1: Trabalhos associados ou trabalhos que se originaram de atividades realizadas neste trabalho de tese.

A Figura A3.1 mostra, de forma esquemática e numa evolução temporal a partir de 1997, os trabalhos que serviram como base ou influenciaram ou foram influenciados pelos desenvolvimentos do presente trabalho de tese - este ultimo salientado com fundo cinza. . Nesta figura, distinguem-se 7 linhas de trabalho, associadas:

- à metodologia para desenvolvimento e realização de vôos do dirigível robótico;
- ao desenvolvimento do modelo matemático; ao desenvolvimento do sistema de controle e navegação;
- ao desenvolvimento da arquitetura de *software* robótico;
- ao desenvolvimento da infra-estrutura de hardware (sensores e processadores);
- à infra-estrutura de *software* e do ambiente de desenvolvimento e operação;
- ao aprimoramento da infra-estrutura eletromecânica.

### ***Proposta inicial***

O desenvolvimento de um dirigível Robótico Autônomo foi originalmente proposto por ELFES, em 1996. Daí originou-se o Projeto AURORA, cuja estruturação inicial, entre 1996 e 1997, deveu-se a BUENO, BERGERMAN e ELFES. Em 1997, RAMOS iniciou sua participação no AURORA, quando já fazia parte do programa de doutoramento da UFSC.

### ***Desenvolvimento do Modelo Matemático***

Os trabalhos de RAMOS ligados ao desenvolvimento do modelo matemático foram baseados no modelo concebido por GOMES (1990). A partir deste modelo, originalmente codificado na linguagem ACSL, foram desenvolvidos simuladores em MATLAB (ver Seção 6.3) e em Java/VRML (ver Seção 6.6). DE-PAIVA realizou aprimoramentos no ambiente MATLAB tornando-o mais adequado para uma ferramenta de projeto de sistema de controle auxiliado por computador (ver Seção 6.4), que posteriormente foi aprimorado ainda mais por AZINHEIRA. Em relação a componentes do modelo em si, motivado por necessidades ligadas ao desenvolvimentos dos vôos, RAMOS incluiu o efeito do vento em 3 eixos (ver Seção 3.4.4), que posteriormente foi aprimorado por AZINHEIRA com a inclusão de um modelo de turbulência em 6 eixos (ver Seção 3.6.3); finalmente RAMOS também agregou um modelo mais adequado para a força de propulsão (ver Seção 3.4.7). Adicionalmente, AZINHEIRA realizou:

i) a interpolação dos coeficientes aerodinâmicos por funções analíticas (ver Seção 3.6.1);

ii) a extrapolação desses coeficientes para ângulos de arfagem e guinada superiores ao tratados por GOMES (ver Seção 3.6.1), e;

iii) a inclusão dos efeitos do fluxo de ar gerado pelos motores nas superfícies aerodinâmicas (ver Seção 3.6.2).

Os servidores de simulação criados por RAMOS, utilizando *sockets*, desenvolvidos a partir dos simuladores, foram importantes recursos para a criação, por RAMOS, do ambiente de desenvolvimento e operação. Deve-se destacar que: MAETA, como estudante em trabalho de iniciação científica, introduziu aprimoramentos na interface gráfica do simulador Java/VRML e foi responsável pela realização do interface que possibilita esse simulador ser utilizado com ferramenta para treinamento de pilotos (ver Seção 6.6.7); MIRISOLA, como estudante em trabalho de iniciação científica, estabeleceu as bases para interligação dos simuladores via *sockets*.

### ***Infra-Estruturas de Hardware e Software e Ambiente De Desenvolvimento e Operação***

O Ambiente de Desenvolvimento e Operação é baseado na Estação de Trabalho para Robôs Industriais (ETRI) desenvolvida no CenPRA (entre 1983 e 1990), na qual RAMOS teve participação e também no ambiente concebido para o Projeto Helix, de um helicóptero robótico (RAMOS, 1995), desenvolvido pela empresa Gyron Tecnologia contando com a cooperação do CenPRA e da UFSC e (entre 1991 e 1996), no qual também RAMOS teve participação ativa.

A partir dessa base de conhecimento adquirido e do levantamento de sensores e infraestrutura de hardware para processadores embarcados realizado por ASANUMA, foi desenvolvido por RAMOS o *hardware* (incluindo computadores, micro-controladores) que integra os sensores e atuadores do dirigível (este *hardware* está descrito no Anexo I). Uma evolução importante desta infra-estrutura de *hardware*, realizada por FARIA e MAETA, foi a inclusão da rede CAN a partir de estudo de viabilidade, especificações e de interfaces e do protótipo em *hardware* estabelecidas precedentemente por RAMOS.

Em relação ao sistema de desenvolvimento e ao arcabouço do *software* do dirigível, ambos baseiam-se nos conceitos da ETRI e do helicóptero do Projeto Helix. Trabalhos de

iniciação científica desenvolvidos por NASCIMENTO, GUIMARÃES e MIRISOLA concluíram pela viabilidade do uso do RTLinux para compor a base do *software* e pela necessidade da utilização do uso de aplicativos servidores, conforme descrito no Anexo I. A partir dessa base foram desenvolvidos por MIRISOLA, como um trabalho de iniciação científica, os protótipos dos módulos de *Kernel* de terra e embarcado e os correspondentes aplicativos servidores, definidos os recursos básicos de plotagem de dados de telemetria. GUIMARÃES, no seu trabalho de iniciação científica, tratou também da questão de detecção e tratamento de erro de comunicação terra-ar.

O *software* embarcado nos micro-controladores foi desenvolvido por RAMOS, MAETA e FARIA. O protótipo deste ambiente, com capacidades para telemetria de dados, teve seu desenvolvimento inicial finalizado no início de 1999 com a participação intensa de seis estudantes de iniciação científica. Os primeiros testes com a integração dos algoritmos de controle foram realizados em simulação por RAMOS e MAETA no decorrer de 1999. Os trabalhos práticos dos mestrados de MAETA e MIRISOLA, dando continuidade e aprofundando as atividades de iniciação científica (ver Seção 7.6) contemplaram a organização e os aprimoramentos que se mostraram necessários no ambiente, em termos de *hardware* e *software*, englobando as infra-estruturas embarcada e em terra.

Paralelamente, durante sua estadia no grupo de SIMMONS na CMU, RAMOS utilizou o ambiente anteriormente estabelecido para desenvolver o modelo conceitual da arquitetura de *software* robótico do dirigível. Para tal foi elaborado um sistema de programação de missão que utiliza scripts, baseado em solução similar previamente criada para o projeto Helix. MIRISOLA, na sua dissertação de mestrado, aperfeiçoou o programador de missão, tornando-o orientado a objetos e integrando-o ao ambiente de desenvolvimento e operação mostrado na Seção 7.6.2.

### ***Sistema de Controle e Navegação.***

O algoritmo controle de trajetória, com fundamentação preditiva, foi proposto por AZINHEIRA e DE-PAIVA conforme mostrado na Seção 4.3.2. Posteriormente, a migração e adaptações necessárias para conversão de diagrama de blocos para código procedural, agregando também estratégias de dessaturação e tratamento diferenciado de erro quando em transição de trajetória, foi realizado por RAMOS sob orientação de BUENO e BRUCIAPAGLIA. A codificação em "C" e o testes em simulação no *hardware* embarcado

do dirigível deveu-se a MAETA. Este código foi adaptado para operar no arcabouço de *software* do dirigível e foi utilizado no primeiro voo com controle de trajetória em março de 2000, com participação intensa da equipe de estudantes da equipe do AURORA na preparação de todos os componentes de hardware e software requeridos para tal. Posteriormente, em consequência das dificuldades práticas para o ajuste dos parâmetros de sintonia do algoritmo, RAMOS sob orientação de BRUCIAPAGLIA reinterpretou a lei de controle transformando-a em uma estrutura PIV (mostrado na Seção 4.3.2), e realizou ensaios desta tanto em simulação quanto em voo sob diferentes condições, conforme é abordado na Seção 8.3.

O algoritmo de controle de altitude foi inicialmente proposto por DE-PAIVA; posteriormente RAMOS o estudou em simulação sob orientação de BRUCIAPAGLIA (Seção 8.4). RAMOS e MAETA realizaram sua migração para o sistema embarcado, e conduziram todos os ensaios práticos que possibilitaram a obtenção dos resultados com controle de altitude e trajetória atuando conjuntamente, como mostrado na Seção 8.6.

### ***Arquitetura de Software Robótico***

A proposição de uma arquitetura de *software* robótico para o dirigível, e a criação de um primeiro protótipo por RAMOS, resultou de estudos prospectivos por ele realizados, resumidos nas Seções 5.1 e 5.4. O modelo conceitual desta arquitetura de *software* baseia-se nos trabalhos de SIMMONS, que propõe uma arquitetura de três camadas para sistemas multi-agentes, Task Description language —TDL por ele criada (apresentada na Seção 5.4.6).

### ***Evolução Eletro-Mecânica Do Dirigível***

Os trabalhos de aperfeiçoamento da infra-estrutura eletro-mecânica incluindo a aerodinâmica foram motivados por conclusões estabelecidas da prática de voo, coordenados por RAMOS e com especificação através de consenso da equipe. Esses trabalhos foram implementados na sua maioria por MAETA e PEIXOTO.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ACSL 1987 ACSL -*Advanced Continuous Simulation Language Reference* Mitchell and Gauthier Associates Edition 4.2 1987
- ADERMANN 1993 ADERMANN, H. ; ROTH, H. *Robust Control of the Zeppelin NT Airship*; in: PROCEEDINGS OF INTERNATIONAL AIRSHIP CONFERENCE, in Stuttgart 1993
- AERC 2001 Aerial Robotics Competition Home Page  
<http://avdil.gtri.gatech.edu/AUVS/IARCLaunchPoint.html>  
 Endereço válido: 16/09/2001
- AGUIRRE 2001 AGUIRRE, A.A. *Vigencia centenaria de los zepelines em*  
<http://www.hinterlandnews.com/newpages/9comercioext/zepelines.htm>  
 m (Endereço válido: 14/08/2001).
- AHA 2001 Advanced Hybrid Airship  
<http://www.ahausa.com>  
 Endereço válido: 22/06/2001
- AIR-FR 2001 Página com informações sobre dirigíveis  
 [<http://perso.wanadoo.fr/blimp/dirig%20modernes.htm>]  
 (Endereço válido: 14/08/2001).
- AIRSPEED 2001 Airspeed Airships Homepage  
<http://www.airship-association.org/airspeed.html> (Endereço válido: 22/06/2001)
- AIST 2001 Stratospheric Platform  
<http://www2.crl.go.jp/mt/b181/research/spf/index-e.htm>  
 (Endereço válido: 14/08/2001).
- ALAMI 1998 ALAMI, R.; CHATILA, R., et al. *An Architecture for Autonomy*. In *International Journal of Robotics Research*, v. 17 n.4 abril 1998
- ARKIN 1989 ARKIN, R. *Towards the unification of navigational planning and reactive control*. AAI Spring Symposium on Robot Navigation, Março 1989
- AMGSS 1996 HGUYEN, H.G.; MARSH, W. *Aspects of the Air-Mobile Ground Security and Surveillance System Prototype* Publicado em *Unmanned Systems*, Vol. 14, No. 1, Winter 1996.
- ARMINGOL 1999 ARMINGOL, J.M. *Notas de curso sobre Robôs Inteligentes*, ministrado na UFSC em 1999
- AZINHEIRA 1999 AZINHEIRA, J. R.; OLIVEIRA, H. V.; ROCHA, B. F. - *Sistema de Medições Aerodinâmicas para um Dirigível do Projeto AURORA – Relatório de Calibração e Manual do Utilizador*. Instituto de Engenharia Mecânica, Instituto Superior Técnico, Lisboa, Portugal.

- AZINHEIRA 2000 AZINHEIRA, J.R., DE PAIVA, E.C., RAMOS, J.J.G., BUENO, S.S.B. *Guidance of an Autonomous Unmanned Airship with the use of Differential Propulsion*. In PROC. XIII BRAZILIAN CONFERENCE ON AUTOMATION, Florianópolis, SC, Brasil, 2000.
- AZINHEIRA 2000a AZINHEIRA, J. R.; PAIVA, E. C.; RAMOS, J. J. G.; BUENO, S.S. *Mission Path Following for na Autonomous Unmanned Airship* In PROC. IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, San Francisco, EUA, Abril, 2000.
- AZINHEIRA 2000b AZINHEIRA, J.R., DE PAIVA, E.C., RAMOS, J.J.G., BUENO, S.S.B. *Hovering Control of an Autonomous Unmanned Airship*. In PROC. 4TH PORTUGUESE CONFERENCE ON AUTOMATIC CONTROL ; Guimarães, Minho, Portugal; October 2000.
- AZINHEIRA 2000c AZINHEIRA, J.R.; DE PAIVA, E.C.; RAMOS, J.J.G.; BUENO, S.S. *Guidance of an Autonomous Unmanned Airship*. In PROC. L5TH BRISTOL INTERNATIONAL UAV SYSTEMS CONFERENCE, Bristol, Inglaterra, April 2000.
- AZINHEIRA 2000d AZINHEIRA, J.R.; PAIVA, E.C.; RAMOS, J.J.G.; BUENO, S.S. *Guidance control strategies for an autonomous unmanned airship*. In Proc. 6TH INTERNATIONAL IFAC SYMPOSIUM ON ROBOT CONTROL, Vienna, Austria, September 2000, pp. 517-522.
- AZINHEIRA, DE- PAIVA et al 2001a AZINHEIRA, J.R., DE PAIVA, E.C., BUENO, S.S. *Influence of Wind Speed on Airship Dynamics*. Journal of Aircraft” (submetido).
- AZINHEIRA, DE- PAIVA et al 2001b Azinheira, J.R., de Paiva, E.C., Carvalho, J.R.H., Ramos, J.J.G., Bueno, S.S., Bergerman, M., Ferreira, P.A.V. *Lateral/Directional Control for an Autonomous Unmanned Airship*. Aircraft Engineering and Aerospace Technology, vol 73 NB 5 2001
- AZINHEIRA, DE- PAIVA et al 2001c Azinheira, J.R., de Paiva, E.C., Ramos, J.J.G., Bueno, S.S., Bergerman, M. *Extended Dynamic Model for AURORA Robotic Airship* 14<sup>th</sup> AIAA Lighter-Than-Air Technical Committee Convention and Exhibition Akron, Ohio, USA; July 2001.
- AZTEC 2001 University of Virginia Solar Airship Program home page <http://www.virginia.edu/~secap/> (Endereço válido: 14/08/2001).
- BLAKELOCK 1991 BLAKELOCK, J.H. *Automatic Control of Aircraft and Missiles 2*. Ed., New York: John Wiley & Sons, 1991.
- BLPEUR 2001 Blimp Europa Home Page <http://perso.wanadoo.fr/blimp/dirig%20modernes.htm> Endereço válido: 16/09/2001
- BONASSO 1995 BONASSO, P.; KORTENKAMP, D.; MILLER, D; SLACK, .M. *Experiences with an Architecture for Intelligent, Reactive Agents*, Journal of Artificial Intelligence Research, 1997

- BORRELLY 1998 BORRELLY, J.J.; COSTE-MANIERE, E; ESPIAU, B. et al. The *ORCCAD Architecture*. In *International Journal of Robotics Research*, v. 17 n.4 april 1998
- BOSCHMA 2001 BOSCHMA, J.H. *Adaptation of Cycloidal Propulsion to The Autoblomp Unmanned Airship*. In: 14th AIAA Lighter-Than-Air Technical Conference, (14: jul. 2001:Akron-EUA). Proceedings AIAA, Akron, USA, 2001.
- BRI 2001 BOCA Research Inc.  
<http://www.bocaresearch.com>  
(Endereço válido: 22/06/2001).
- BRIGHT 2001 BRIGHT T.A., GETLEIN, S.; *Remote Sensing User's Guide*, Documento publicado pelo U.S. Army Environmental Center, Março 2001
- BROOKS 1986 Brooks, R. *A Robust Layered Control System for a Mobile Robot*, IEEE Journal of Robotics and Automation, Vol. 2, No. 1, March 1986.
- BRUTZMAN 1994 BRUTZMAN, D. P. *A virtual world for an Autonomous Underwater Vehicle*, Ph.D. Thesis, Naval Postgraduate School, USA, Dec, 1994
- BUHLMANN 1995 BUHLMANN, K. *Thrust Vector and Flight Steering of Zeppelin Airship LZ N07*. In PROCEEDINGS AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, AIAA, USA, 1995
- CAN 2001 Robert Bosch GmbH - CAN Specification Version 2.0, Setembro de 1991.
- CAVALCANTI 1981 CAVALCANTI, S. *Predictive Steering of Dirigibles using switching curve approach* In PROCEEDINGS AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, AIAA, USA, 1981
- CEPAGRI 2001 CEPAGRI UNICAMP Metereologia  
<http://www.cpa.unicamp.br/>  
Endereço válido: 16/09/2001
- CHELLI 2001 CHELLI, E., DALA, L., et al *Aerodynamics Investigation of the 1/80 CL 160 P1 Airship*. In: 14th AIAA Lighter-Than-Air Technical Conference, (14: jul. 2001:Akron-EUA). Proceedings AIAA, Akron, USA, 2001.
- CHOI 2001 CHOI, S.K. YUH, J. *A Virtual Collaborative World Simulator for Underwater Robots using Multi-Dimensional, Synthetic Environment*, In Proc. IEEE INT. CONFERENCE ON ROBOTICS AND AUTOMATION ICRA' 2001, Seoul, Korea.
- CHR 2001 Home Page do Chorus  
[http://www-inf.int-evry.fr/~adm\\_chor/faq.html](http://www-inf.int-evry.fr/~adm_chor/faq.html)  
(Endereço válido: 14/08/2001).

- CLF 2001 Cargo Lifter Home Page  
[http://www.cargolifter.com/2001/inhalt/loesung\\_cl160\\_ueberblick\\_e.html](http://www.cargolifter.com/2001/inhalt/loesung_cl160_ueberblick_e.html) (Endereço válido: 14/08/2001).
- CLF 2001a Processo de descarga para o Cargo Lifter  
[http://www.cargolifter.com/2001/inhalt/loesung\\_cl160\\_lastaustausch\\_verfahren\\_e.html](http://www.cargolifter.com/2001/inhalt/loesung_cl160_lastaustausch_verfahren_e.html) (Endereço válido: 14/08/2001).
- CLIMERH 2001 Centro Int. de Meteorologia e Rec. Hídricos de SC  
<http://www.climerh.rct-sc.br/mapavento06.html>
- CONNELL 1992 CONNELL, J. *SSS: A Hybrid Architecture Applied to Robot Navigation* In Proc. IEEE International Conference on Robotics and Automation - ICRA'92..
- COSTE-MANIERE & SIMMONS 2000 COSTE-MANIERE, E.; SIMMONS, R. *Architecture, the Backbone of Robotic Systems*. In Proc. IEEE International Conference on Robotics and Automation, San Francisco, EUA, Abril, 2000.
- CPTEC 2001 Modelo Regional ETA – Meteogramas (Previsao Campinas)  
<http://www3.cptec.inpe.br/cgi-bin/etagram.cgi?campo=XCPN&data=00>  
 Endereço válido: 16/09/2001
- CRAIG 1989 CRAIG, J. *Introduction to Robotic- Mechanics and Control* 2. Ed., New York: Addison-Wesley, 1989.
- CRB 2001 Oficial CORBA home Page  
<http://www.corba.org/>  
 Endereço válido: 14/08/2001
- DANNEKER 1999 DANNEKER, S. *Distributed Thrust and Autonomous Ground Handling*. In PROCEEDINGS AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, AIAA, USA, 1999
- DELAURIER 1981 DE LAURIER, J.D.; EVANS, J.R. *The Shenandoah Flies Again: A Computer Simulation*. 4th AIAA Lighter-Than-Air Technology Conference, Annapolis, Maryland, USA.
- DE-PAIVA 1998 PAIVA, E.C. *Escalonamento e adaptação do modelo e simulador para o caso particular do dirigível AS800*. Relatório Técnico CenPRA , abril de 1998
- DE-PAIVA 2001a PAIVA, E.C. *Relatório Anual de Atividades à FAPESP Processo n. 98/13563-1*, junho 2001
- DE-PAIVA et al 1999a PAIVA, E.C.; BUENO, S.S.; GOMES, S.B.V. ; RAMOS, J.J.G.; BERGERMAN, M. *A Control System Development Environment for AURORA's Semi-Autonomous Robotic Airship*. In Proc. IEEE International Conference on Robotics and Automation - ICRA'99, Detroit, USA.

- DE-PAIVA et al 1999b PAIVA, E.C.; BUENO, S.S.; E BERGERMAN, M. (1999). *A Robust Attitude Control System for AURORA's Semi-Autonomous Robotic Airship*. In Proc. 13th AIAA Ligher-Than-Air Technology Conference, Norfolk, VA.
- DE-PAIVA et al 2001 DE PAIVA, E.C.; CARVALHO, J.R.H.,; FERREIRA, P.A.V.; AZINHEIRA, J.J.R. (2001) *An H2/H-infinity Heading Controller for AURORA-I Semi-Autonomous Robotic Airship*. In Proc. 14TH AIAA LIGHTER-THAN-AIR TECHNOLOGY CONFERENCE, Akron, Ohio, USA.
- DIAMOND 2001 Diamond Systems: PC/104  
<http://www.diamondsystems.com>  
 Endereço válido: 14/08/2001
- DMU 1999 DMU-AHRS User's Manual – Revision 1.2 July 1999. CrossBow Technology Inc. <http://www.xbow.com> Endereço válido: 14/08/2001
- DORRINGTON 1999 DORRINGTON, G.E. *Some General Remarks on the design of Airships* IN PROCEEDINGS 13TH AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, (13: jul. 1999:Norfolk-EUA). AIAA, Norfolk, USA, 1999.
- EAI 2001 Extern Authoring Interface Specification  
[http://www.vrml.org/fs\\_specifications.htm](http://www.vrml.org/fs_specifications.htm)  
 Endereço válido: 14/08/2001
- ECOBLIMP 2001 ECOBLIMP Home Page  
<http://www.ecoblimp.com>  
 Endereço válido: 16/09/2001
- EHMC 2001 The Enhanced Machine Controller  
<http://www.isd.mel.nist.gov/projects/emc/emc.html>  
 Endereço válido: 14/08/2001
- ELE-LIB 2001 Electronic Library  
<http://www.encyclopedia.com/articles/00228.html>  
 Endereço válido: 16/09/2001
- ELFES 1990 ELFES, A.; *Autonomous Underwater Vehicle (AUV) Architecture*. IBM T. J. Watson Research Center Internal Report, 1990.
- ELFES 2000a ELFES, A.; BERGERMAN, M.; CARVALHO, J.R.H.; *Towards Dynamic Target Identification Using Optimal Design of Experiments* IEEE International Conference on Robotics and Automation, São Francisco, CA, EUA, Abril 2000.
- ELFES 2000b ELFES, A.; BERGERMAN, M.; CARVALHO, J. R. H. *Dynamic target identification by an aerial robotic vehicle* Dynamische Perzeption, Baratoff, G. and Neumann, H. (eds.), AKA, Berlin, September 2000.

- ELFES et al. 1998a ELFES, A.; BUENO, S.S.; BERGERMAN, M.; RAMOS, J.J.G. *A semi-autonomous robotic airship for enviromental monitoring missions*. Proceedings International Conference on Robotics and Automation, Leuven, Bélgica, Maio 1998.
- ELFES et al. 1998b ELFES, A.; BUENO, S.S.; BERGERMAN, M.; RAMOS, J.J.G.; PAIVA, E.C.; ASANUMA, K.; BEIRAL, J.A.R. *Projeto AURORA*. Anais XII Congresso Brasileiro de Automática, Uberlândia, MG, Set. 1998, pp. 79-84.
- ENVIROBLIMP 2001 Enviroblimp  
<http://www.frc.ri.cmu.edu/projects/enviroblimp/>  
 (Endereço válido: 14/08/2001).
- ERAST 2001 Home Page do Projeto ERAST  
<http://www.dfrc.nasa.gov/Projects/Erast/erast.html>  
 Endereço válido: 16/09/2001
- ESPIAU 1992 ESPIAU, B.; CHAUMETTE, F.; RIVES, P. *A new approach to visual servoing in robotics* IEEE Transactions on Robotics and Automation, vol. 8, no. 3, 1992, pp. 313-326
- ESPRIT 1993 ESPRIT Consortium AMICE. *CIMOSA Open Systems Architecture for CIM*, 2nd, revised and extended edition, Research Reports ESPRIT, Project 688/5288, AMICE, Volume 1. Berlin: Springer-Verlag, 1993.
- FARIA 2001 FARIA, B.G.; MAETA, S.M.; RAMOS, J.G.; BERGERMAN, M.; BUENO, S.S. *Um barramento de campo Controller Area Network para um dirigível autônomo não tripulado*. XVI Congresso Brasileiro de Engenharia Mecânica - COBEM 2001; Novembro 2001, Uberlândia, MG
- FERNANDEZ 2000 FERNANDEZ, J.L. *Supervision, detection, diagnosis and exception recovery in autonomous mobile robots*, Ph.D. Thesis Dept. de Ingeniería de Sistemas y Automática, Universidad de Vigo, Vigo, Spain, Feb. 2000.
- FRASER 1994 FRASER, R. *Embedded Command and Control Infrastructures for Intelligent Autonomous Systems*, Ph.D. Thesis, Department of Aeronautics and Astronautics, University of Southampton, England, May, 1994.
- FRAZZATO 2001 FRAZZATO, R. *Sensores para monitoramento do status do dirigível* Relatório de Trabalho de Iniciação científica programa PIBIC, julho 2001
- FREESOFT 2001 Free Software Foundation  
<http://www.free-soft.org/>  
 Endereço válido: 14/08/2001

- FREEWAVE 2001 FreeWave Technologies Home Page.  
<http://www.freewave.com>  
 Endereço válido: 14/08/2001
- FURUNO 2001 Furono Professional Marine Products  
<http://www.furuno.com/news/saoff.html>  
 Endereço válido: 16/09/2001
- FUTABA 2001 Home Page da Futaba  
<http://www.futaba.com>  
 (Endereço válido: 14/08/2001).
- GAT 1992 GAT, E. *Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots* in American Association for Artificial Intelligence Conference 1992.
- GAT 1997 GAT, E. *On Three-Layer Architectures* Aparece em Artificial Intelligence and Mobile Robots, MIT Press, 1998
- GNU 2001 Home Page da GNU  
<http://www.gnu.org>  
 Endereço válido: 14/08/2001
- GOMES 1990 GOMES, S.B.V. *An investigation of the flight dynamics of airships with application to the YEZ-2A.*, Cranfield,1990 Ph.D. Thesis, College of Aeronautics, Cranfield University, England.
- GOOD-YEAR 1997 The Blimp Book Ed., Mill Valley,EUA: Squarebooks, 1997.
- GRIFFIN 1998 GRIFFIN, I.; NELSON, J. *Linux Network Programming* Linux Journal, nº46 february 1998
- GUDWIN 1999 GUDWIN, RICARDO RIBEIRO - *Notas de aula do curso : IA005 – Introdução à Teoria de Agentes*, Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas (FEEC-UNICAMP), 1999.
- GYRON-CTI 1994 Documentos Internos de Projeto de Desenvolvimento de Helicóptero Autônomo Gyron Tecnologia / CTI, 1994.
- HAGENLOCHER 1993 HAGENLOCHER, K.; MANDEL, M. *The Zeppelin LZ N07 project* IN PROC. OF THE 1<sup>ST</sup> INTERNATIONAL AIRSHIP CONFERENCE, 1993.
- HALP 2000 REHMET, M.; KROEPLIN, B.H. – *Recent development on High Altitude Platforms* PROCEEDINGS IACE2000 - INTERNATIONAL AIRSHIP CONVENTION AND EXHIBITION – Airship Association, Alemanha, 2000.
- HALP 2001 Hohen Platform  
<http://www.isd.uni-stuttgart.de/arbeitsgruppen/airship/halp/index.htm>  
 (Endereço válido: 14/08/2001).

- HASEGAWA 1983 HASEGAWA, K.; MIZUTANI, T. *On the Autonomous Trajectory Generating Servomechanism for Manipulator Control System Science VIII*, International Conference on Systems, Wroczaw, Poland, 1983
- HELIOS 2001 Home Page do Projeto Helios  
<http://www.dfrc.nasa.gov/Projects/Erast/helios.html>  
 Endereço válido: 16/09/2001
- HEPPERLE 2001 Martin Hepperle Home page for airfoils  
<http://members.tripod.de/MartinHepperle/Airfoils/index.htm>  
 (Endereço válido: 14/08/2001).
- HOOKEYWAY 1979 HOOKEYWAY, R. O. *Progress in Airship Control*. International Symposium on the Economics and Technology of modern Airships, Paris, 1979
- HOTAIR 2001 Repositório de informações sobre dirigíveis  
[\[http://www.hotairship.com/database/wai.html](http://www.hotairship.com/database/wai.html)  
 (Endereço válido: 14/08/2001).
- JAVA 2001 Java Home Page  
<http://www.sun.com/java>  
 Endereço válido: 14/08/2001
- JUMPTec 2001 JUMPTec Industrial Hardware.  
<http://www.jumpotec.de>  
 Endereço válido: 14/08/2001
- KAEMPF 1995 KAEMPF, B. G.; WELL, K. H. *Attitude Control for a Remoted Controlled Airship*, In : PROC. OF THE 11TH LTA SYSTEMS CONFERENCE, 1995
- KAEMPF 1997 KAEMPF, B. G. *Dynamics and Control of a Small Airship*, Proc. of the 2nd International Airship Conference, 1997.
- KANTOR 2001 KANTOR, G.; WETTERGREEN, D.; OSTROWSKI, J.; SINGH S. *Collection of Environmental Data from an Airship Platform* Proceedings of Intelligent Systems and Advanced Manufacturing. Symposium of SPIE (International society for Optical Engineering) n° 4571, 2001
- KHOURY 1999 KHOURY, G.A.; GILLET, J.D. *Airship Technology* 1. Ed., Cambridge: Cambridge University Press, 2000.
- LACROIX 2000 LACROIX, S. *Toward autonomous airships: research and developments at LAAS/CNRS*. In: PROCEEDINGS IACE2000 – INTERNATIONAL AIRSHIP CONVENTION AND EXHIBITION – Airship Association, Alemanha, 2000.
- LACROIX 2001 LACROIX, S. *Developments on autonomous airship control at LAAS/CNRS* In: PROCEEDINGS 14TH AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, (14: jul. 2001: Akron-EUA). Proceedings AIAA, Akron, USA, 2001.

- LAMB 1918 LAMB, H. *The Inertia Coefficients of an Ellipsoid Moving in Fluid*, British Aeronautical Research Committee Report and Memoranda No. 623., 1918
- LARMAN 1998 LARMAN, C. *Applying UML and Patterns An Introduction to Object Oriented Analysis and Design*. 1. Ed., New York: Printice Hall, 1998.
- LATOMBE 1991 LATOMBE, J.C. *Robot motion planning*. Kluwer Academic Publishers, Boston, 1991
- LINUX 2001 Linux Home Page  
<http://www.linux.org/>  
Endereço válido: 14/08/2001
- LINUXCP 2001 Comparação Linux com RTLinux  
<http://www.electro.fisica.unlp.edu.ar/rt/index.html>  
ultima visita em 25 de junho de 2001
- LinuxRT 2001 Home Page do RTAI  
<http://www.realtimelinux.org/about.phtml>  
(Endereço válido: 14/08/2001).
- LME 2001 LME Hexapod Machine  
<http://www.i-way.co.uk/~storrs/lme/LMEHexapodMachine.html>  
Endereço válido: 14/08/2001
- LOTTE 2001 Lotte Home Page  
<http://www.isd.uni-stuttgart.de/arbeitsgruppen/airship/lotte/>  
(Endereço válido: 14/08/2001).
- LOURTIE *et al.* 1995 LOURTIE, P. ;AZINHEIRA,J.R., RENTE, J.P.; FELICIO, P. (1995) *ARMOR project – autonomous flight capability*, AGARD FVP95 Specialist Meeting on Design Operation of Unmanned Air Vehicles, Turquia
- LYNX 2001 Home Page da Lynx Works  
<http://www.lynxworks.com/>  
(Endereço válido: 14/08/2001).
- MADRID 1990 MADRD, M.; PALHARES, A. *Controle de Posição e velocidade de Robôs Manipuladores - Desempenho com Juntas em Malha Aberta e Malha Fechada*. 8º Congresso Brasileiro de Automática, set. 1990, Belém, Brasil
- MAETA 2001 MAETA, S.M. *Desenvolvimento da Infraestrutura Embarcada do Projeto AURORA*. Dissertação de Mestrado, Universidade Estadual de Campinas – Unicamp, Instituto de Computação, Programa de Pós Graduação em Computação, Orientador Samuel Bueno, Julho 2001.
- MANDEL 1997 MANDEL, M. *Controllability and Associated Loads of the Zeppelin LZ 07*. In PROCEEDINGS AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, AIAA, USA, 1997

- MAT-CP 2001 Fabricante de Material composto  
<http://www.etcusa.com/hpc/tailfin.htm>  
 (Endereço válido: 14/08/2001).
- MATHWORKS 2001 Home Page da MATHWORKS INC.  
<http://www.mathworks.com>  
 Endereço válido: 14/08/2001
- MCISAAC 2000 MCISAAC, K.; OSTROWSKI, J. *Visual Servoing with Dynamics: Control of an Unmanned Blimp*. IEEE International Conference on Robotics and Automation, San Francisco, EUA, Abril, 2000.
- MCLEAN 1990 MCLEAN, D. *Automatic Flight Control Systems*. 1. Ed., New York: Printice Hall, 1991.
- MESA 2001 High Performance 2D/3D Graphics  
<http://www.opengl.org/>  
 Endereço válido: 14/08/2001
- MIRISOLA, 2001 MIRISOLA, L.G.B. *Desenvolvimento da Estação de Terra do Projeto AURORA*. Dissertação de Mestrado, Universidade Estadual de Campinas – Unicamp, Instituto de Computação, Programa de Pós Graduação em Computação , Orientador Samuel Bueno, Julho 2001.
- MIT-FORM 2001 Formação de Dirigíveis no MIT  
<http://www.mit.edu/people/jhow/ff/blimps/blimps.html>  
 (Endereço válido: 14/08/2001).
- MM515C 2001 MiniMODUL-535/515C – Hardware Manual –Phytec.  
<http://www.phytec.com>  
 Endereço válido: 14/08/2001
- MOSCATO 2001 Traveling Salesman Problem (TSP) Home Page  
[http://www.densis.fee.unicamp.br/~moscato/TSPBIB\\_home.html.htm](http://www.densis.fee.unicamp.br/~moscato/TSPBIB_home.html.htm)  
 (Endereço válido: 14/08/2001).
- MOWFORTH 1991 MOWFORTH , E. An Introduction to the Airship. England:.Airship Association Publication N° 3, 1991,
- NAGABUSHAN 1981 NAGABUSHAN, B.L.; TOMLINSON, N.P. *Dynamics and Control of a Heavy Lift Airship in cross wind* IN PROCEEDINGS AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, AIAA, USA, 1981
- NAGABUSHAN 1984 NAGABUSHAN, B.L.; FAISS, G.D. *Thrust Control of V/STOL Airship*. *Journal of Aircraft* v. 21, n. 6 1984
- NAGABUSHAN 1986 NAGABUSHAN, B.L.; TOMLINSON, N.P. *Thrust vectored Takeoff, Landing and Ground Handling of an Airship*. *Journal of Aircraft* v. 23, n. 3 1986
- NAGABUSHAN 1989 NAGABUSHAN, B.L. *Control Configured Airship Design* IN PROCEEDINGS AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, AIAA, USA, 1989

- NAGABUSHAN 1991 NAGABUSHAN, B.L.; Pasha, R. *Analyses of Airship Lateral Maneuverability* IN PROCEEDINGS AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, AIAA, USA, 1991
- NAGABUSHAN 1995 NAGABUSHAN, B.L.; TAN, S. *Directional Control of and Advanced Airship* IN PROCEEDINGS AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, AIAA, USA, 1995
- NAGABUSHAN 1997 TAN, S.; NAGABUSHAN, B.L. *Robust Heading-Hold Autopilot for an Advanced Airship* IN PROCEEDINGS AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, AIAA, USA, 1997
- ONDA 1999 ONDA, M. *Design Considerations on Stratospheric LTA Platform*. In 13TH AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, (13: jul. 1999:Norfolk-EUA). Proceedings AIAA, Norfolk, USA, 1999.
- ONDA 2001 ONDA, M. *A Feasibility Study on a Solar Powered Stratospheric Long-Endurance LTA Platform with 40 m/s Air Speed*. In: 14TH AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, (14: jul. 2001:Akron-EUA). Proceedings AIAA, Akron, USA, 2001.
- OSE 2001 Home Page da OS Engines  
<http://www.osengines.com>  
Endereço válido: 14/08/2001
- OVRTL 2001 An Overview of Realtime in the Linux Kernel  
<http://www.realtimelinux.org/about.phtml>  
Endereço válido: 14/08/2001
- PALLET 1983 PALLET; E. H. *Automatic Flight Control*. 2. Ed., New York: Granada Publishing, 1983.
- PAUL 1981 PAUL, R. *Robot Manipulators* MIT Press, 1981
- PC104 2001 PC/104 Consortium.  
<http://www.pc104.org>  
Endereço válido: 14/08/2001
- PEIXOTO 2001 PEIXOTO, R.P.; MAETA, S.M.; YAMAGUCHI, H.; et al *Desenvolvimento de elementos mecânicos para um dirigível robótico não-tripulado através de prototipagem rápida*. XVI Congresso Brasileiro de Engenharia Mecânica - COBEM 2001; Novembro 2001, Uberlândia, MG
- PEREIRA 2001 Pereira, C. S.; Bueno, S.S.; Amaral, W.C.; Paiva, E.C.; Maeta, S.M.; Ramos, J.G. *Formulação de um Sistema de Navegação Inercial para um Dirigível Não-Tripulado: Primeiros Resultados*. XVI Congresso Brasileiro de Engenharia Mecânica - COBEM 2001; Novembro 2001, Uberlândia, MG

- PRECISION 2001 Precision Navigation Inc.  
<http://www.pnicorp.com>  
 Endereço válido: 14/08/2001
- PRESS 1992 PRESS, W.; TEUKOLSKY, S. et al. *Numerical Recipes in C* 2nd edition Cambridge University Press pp. 39-40, Cambridge, 1992
- PROUTY 1986 PROUTY, R. *Helicopter Performance, Stability, and Control*. Robert E. Krieger, 1986.
- PROXIM 2001 PROXIM Wireless LAN White Paper  
<http://www.proxim.com/learn/library/whitepapers/pdf>  
 Endereço válido: 14/08/2001
- PULNIX 2001 PULNiX Europe Ltda  
<http://www.pulnix.com>  
 Endereço válido: 14/08/2001
- QNX 2001 QNX Software Systems  
<http://www.qnx.com/>  
 (Endereço válido: 22/06/2001).
- RAMOS 1986 Ramos, J.; *Geração de trajetórias para Robôs Manipuladores: Aspectos Cinemáticos e Computacionais*, Dissertação de Mestrado, FEE UNICAMP, 1986.
- RAMOS 1995 RAMOS, J.J.G. e NEVES, O. *Environment for unmanned helicopter control system development: application examples* Proceedings 4th IEEE Conference on Control Applications, Albany, NY, Sep. 1995
- RAMOS 1995a RAMOS, J.J.G. E NEVES, O. *A Transputer Based Fuzzy Control System for an unmanned helicopter control* Proceedings IFISA, São Paulo, 1995
- RAMOS 1998 RAMOS, J.J.G.; BUENO, S.S.; MAETA, S.M.; PAIVA, E.C. et al. *Project AURORA: Autonomous Unmanned Remote Monitoring Robotic Airship*. 2<sup>nd</sup> International Airship Convention and Exhibition, Bedford, UK, 1998
- RAMOS 1999 RAMOS, J.J.G.; MAETA, S.M.; MIRISOLA, L.G.B.; BERGERMAN, M.; BUENO, S.S.; PAVANI, G.S.; BRUCIAPAGLIA, A. *A software environment for an unmanned autonomous airship*. International Conference on Advanced Intelligent Mechatronics, Atlanta, GA, EUA, September, 1999
- RAMOS 1999a RAMOS, J.J.G.; MAETA, S.M.; BERGERMAN, M.; BUENO, S.S.; MIRISOLA, L.G.B.; BRUCIAPAGLIA, A. *Development of a VRML/Java unmanned airship simulating environment*. IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyongju, Korea, October 1999, pp. 1354-1359.

- RAMOS 2000 RAMOS, J.J.G.; PAIVA, E.C.; MAETA, S.M.; MIRISOLA, L.G.B.; AZINHEIRA, J.R. ET AL *Project AURORA: A Status Report* 3rd International Airship Convention and Exhibition, Friedrichshafen, Alemanha, Julho 2000,
- RAMOS 2001a RAMOS, J.J.G.; DE PAIVA, E.C.; CARVALHO, J.R.H.; FERREIRA, P.A.V.; AZINHEIRA, J.R.; BUENO, S.S; MAETA, S.M.; MIRISOLA, L.G.B.; FARIA, B.G.; BERGERMAN, M.; ELFES, A. *Path Tracking Flight Test of an Autonomous Unmanned Robotic Airship*. In Proc. 3<sup>RD</sup> INTERNATIONAL CONFERENCE ON FIELD AND SERVICE ROBOTICS, Junho, 2001, Otaniemi, Espoo, Finlândia.
- RAMOS 2001b RAMOS, J.J.G.; PAIVA, E. C.; AZINHEIRA, J. R.; BUENO, S. S.; MAETA, S. M.; MIRISOLA, L. G. B.; BERGERMAN, M. *Autonomous Flight Experiment with a Robotic Unmanned Airship*. In Proc. IEEE INT. CONFERENCE ON ROBOTICS AND AUTOMATION ICRA' 2001, Seoul, Korea.
- RAS 2001 Ros Aero Systems  
<http://perso.wanadoo.fr/blimp/RAS.htm>  
(Endereço válido: 14/08/2001).
- RIDAO et al 2001 RIDAO, P.; BATLLE, J. et al *A Distributed Environment for Virtual and/or Real Experiments for Underwater Robots* In Proc. IEEE INT. CONFERENCE ON ROBOTICS AND AUTOMATION ICRA' 2001, Seoul, Korea.
- ROTH 1994 ROTH, H; ADERMANN. H. *Flight Control Concept of the Zeppelin NT Airship Using Fuzzy Logic* in: PROCEEDINGS OF INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING 1994
- ROTH 1994a ROTH, H; ADERMANN. H. *Adaptive Fuzzy Control for the Zeppelin New Technology Airship* PROCEEDINGS OF INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING 1994
- ROUTER 2001 Home Page do Projeto do Roteador Linux  
<http://www.linuxrouter.org/>  
Endereço válido: 14/08/2001
- RTAI 2001 Real Time Application Interface – RTAI  
<http://www.rtai.org/>  
Endereço válido: 14/08/2001
- RTAI 2001 An Overview of Realtime in the Linux Kernel  
<http://www.rtai.org>  
(Endereço válido: 14/08/2001).
- RTAI 2001b History of RTAI  
<http://www.aero.polimi.it/projects/rtai/history.htm>  
Endereço válido: 14/08/2001

- RTCL 2001 *Real Time Controls Laboratory*  
<http://rtic-lab.sourceforge.net/>  
 Endereço válido: 14/08/2001
- RTLINUX 2001 Real Time Linux Home Page  
<http://www.rtlinux.org/>  
 Endereço válido: 14/08/2001
- RTLINUX 2001b Real Time Linux White Paper  
<http://www.rtlinux.org/documents/papers/whitepaper.html>  
 Endereço válido: 14/08/2001
- RTLINUX 2001c Mini RTLinux  
<http://www.rtlinux.org/minirtl.html>  
 Endereço válido: 14/08/2001
- RUGER et al 1999 RUGER, U.; BLANK, F; KROPLIN,B. *Acquisition of Flight Data Such as Drag and Propeller Thrust by a Remote Controlled Solar Airship*. In PROCEEDINGS AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, AIAA, USA, 1999
- SAN 2001 San Disk - Flash Data Storage Products  
<http://www.sandisk.com>  
 Endereço válido: 14/08/2001
- SASS 2001 Imagens Vôo Autonomo SASS\_LITE  
<http://www.boschaero.com/man4/9-11-98.htm>  
 (Endereço válido: 14/08/2001).
- SIEM515 1997 Manual do usuário do Micro-controllador Single-Chip 80c515-80C535 da Infineon, novembro de 1997
- SIEM517 1994 Manual do usuário do Micro-controllador Single-Chip 80c517-80C537 da Infineon, maio de 1994
- SIMMONS 1994 SIMMONS, R. *Structured Control for Autonomous Robots*, IEEE Transactions on Robotics and Automation, vol 10, No 1, Feb 1994
- SIMMONS 1998 SIMMONS, R. AND APFELBAUM, D. *A Task Description Language for Robot Control*, In Proc. Conference on Intelligent Robotics and Systems, Vancouver Canada, October 1998
- SIMMONS 2000 SIMMONS, R; SINGH, S.; HERSHBERGER, D.; RAMOS, J.J.G; SMITH, T.; *First Results In The Coordination Of Heterogeneous Robots For Large-Scale Assembly* Seventh International Symposium On Experimental Robotics, ISER 2000, Haway, USA, Dezember 2000.
- SILVEIRA et al. 2001 SILVEIRA, G.; CARVALHO, J.; MADRID, M.; BUENO S.; RIVES, P. *Lateral Control of an Aerial Unmanned Robot Using Visual Servoing Techniques* Proceedings of the IEEE 2<sup>nd</sup> Workshop on Robot Motion and Control, Polônia, 2001.

- SIMUL 2001 Real Flight Simulator  
<http://www.qnx.com/>  
 (Endereço válido: 22/06/2001).
- SKYBUS 2001 Sky Bus Home page  
[http:// http://www.skybus.ch/](http://www.skybus.ch/)  
 (Endereço válido: 14/08/2001).
- SKYCAT 2001 SkyCat Home page  
[http:// http://www.skycattech.com/](http://www.skycattech.com/)  
 (Endereço válido: 14/08/2001).
- SKYCAT 2001a SkyCat Stratospheirc Platform  
[http://www.airship.com/prod/stratsat\\_frames.htm](http://www.airship.com/prod/stratsat_frames.htm)  
 (Endereço válido: 14/08/2001).
- SKYMEDIA 2001 Home Page da empresa SkyMidia  
<http://www.skymidia.com.br/index.html>  
 (Endereço válido: 14/08/2001).
- SKYSHIP 2001 Global Skyships Home Page  
<http://www.globalskyships.com/gss3.htm>  
 Endereço válido: 16/09/2001
- SKYSTA 2001 Sky Station Home page  
<http://www.skystation.com>  
 (Endereço válido: 14/08/2001).
- SOCK-FAQ 2001 Unix Socket faq  
<http://www.lcg.org/sock-faq/>  
 (Endereço válido: 14/08/2001).
- TCL 2001 TCL Developer Site  
<http://tcl.activestate.com/>  
 Endereço válido: 14/08/2001
- THESEUS 97 Projeto Theseus  
[http://www.hiflight.com/gallery\\_theseus.html](http://www.hiflight.com/gallery_theseus.html)  
 Endereço válido: 16/09/2001
- TITTERTON 1997 TITTERTON, D.; WESTON, J. *Strapdown Inertial Navigation Technology*. 1Ed Peter Peregrinus Ltd. United Kingdom, 1997.
- TRI-M 2001 Tri-M Systems Inc. - PC/104 Embedded Hardware Solutions  
<http://www.Tri-M.com>  
 Endereço válido: 22/06/2001
- TRIMBLE 2001 Trimble Navigation  
<http://www.trimble.com>  
 Endereço válido: 14/08/2001
- TURNER 2000 TURNER, A. *Development of a Semi-Autonomous Control System for the UVA Solar Airship Aztec*. Virginia-EUA,2000, Dissertação em Engenharia Aeroespacial- School of Engineering and Applied Science University of Virginia

- UNZICKER 1993 UNZICKER, S. *Aspect of Flight Dynamics Determining the Zeppelin NT Configuration* In: PROC. OF THE 1<sup>ST</sup> INTERNATIONAL AIRSHIP CONFERENCE, 1993.
- UPSON 1925 UPSON, R.; KLIKOFF, W. *Application of Practical Hydrodynamics to Airship Design*. British Aeronautical Research Committee Report and Memoranda, N<sup>o</sup> 405, 1925
- USLTA 2001 US LTA Home Page  
<http://www.us-lta.com/remotesensing.html>  
 Endereço válido: 16/09/2001
- VRML 2001 VRML Home Page  
<http://www.vrml.org>  
 Endereço válido: 14/08/2001
- VXWORKS 2001 Wind River OS/Run-Time Products - VxWorks 5.4 -  
<http://www.windriver.com/products/html/vxwks54.html> (Endereço válido: 22/06/2001).
- WIMMER 2001 WIMMER, D.; WELL, K. Instrumentation, identification and control of Airship LOTTE. In: PROC. 14TH AIAA LIGHTER-THAN-AIR TECHNICAL CONFERENCE, (14: jul. 2001:Akron-EUA). Proceedings AIAA, Akron, USA, 2001.
- WONG 1997 WONG; K.C.; *Unmanned Aerial Vehicles (UAVs) - Are They Ready This Time? Are We?*  
[http://www.aero.usyd.edu.au/wwwdocs/UAV\\_RAeS\\_prez\\_26Nov97.PDF](http://www.aero.usyd.edu.au/wwwdocs/UAV_RAeS_prez_26Nov97.PDF) (Endereço válido: 16/09/2001).
- WOOLDRIDGE 1995 WOOLDRIDGE, M. J.; JENNINGS, N. R. *Intelligent agents: Theory and practice*. Knowledge Engineering Review, 1995
- XR400 1999 Monad XR400 Hardware Manual, março 1999
- ZEPNT 2001 Zeppelin NT Home page  
[http://www.zeppelin-nt.com/index\\_uk.html](http://www.zeppelin-nt.com/index_uk.html)  
 (Endereço válido: 14/08/2001).
- ZWAAN 2001 ZWAAN, S. *Vision based station keeping and docking for floating robots* Dissertação de Mestrado em Engenharia Eletrônica Universidade Técnica de Lisboa Instituto Superior Técnico; maio de 2001