

ARMANDO ALVES NETO

**GERAÇÃO DE TRAJETÓRIAS PARA VEÍCULOS AÉREOS
AUTÔNOMOS NÃO-TRIPULADOS**

Belo Horizonte
08 de agosto de 2008

ARMANDO ALVES NETO

ORIENTADOR: MARIO FERNANDO MONTENEGRO CAMPOS

**GERAÇÃO DE TRAJETÓRIAS PARA VEÍCULOS AÉREOS
AUTÔNOMOS NÃO-TRIPULADOS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Belo Horizonte

08 de agosto de 2008

Alves Neto, Armando

A474g Geração de Trajetórias Para Veículos Aéreos Autônomos
 Não-Tripulados / Armando Alves Neto. — Belo Horizonte,
 2008

xxiv, 100 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais

Orientador: Mario Fernando Montenegro Campos

1. VAANTS. 2. Planejamento de Trajetórias. 3. Controle
de Aeronaves. I. Título.

CDU 007.52



UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Geração de Trajetórias Para Veículos Aéreos Autônomos Não-Tripulados

ARMANDO ALVES NETO

Dissertação defendida e aprovada pela banca examinadora constituída por:

Prof. MARIO FERNANDO MONTENEGRO CAMPOS – Orientador
Universidade Federal de Minas Gerais

Prof. LUIZ CHAIMOWICZ
Universidade Federal de Minas Gerais

Prof. LEONARDO ANTÔNIO BORGES TÔRRES
Universidade Federal de Minas Gerais

Belo Horizonte, 08 de agosto de 2008

Dedico este trabalho aos meus pais, Celso e Luci.

Agradecimentos

Agradeço em primeiríssimo lugar àquelas pessoas que dão sentido a minha vida. Meu pai Celso e minha mãe Luci, que fizeram de mim aquilo que sou hoje. Meu irmão Celsinho, companheiro de todas as horas. Grazielli, minha noiva, cujo apoio e a paciência foram indispensáveis nos momentos mais difíceis.

Em segundo lugar, agradeço ao Professor Mario Campos pela orientação não apenas técnica, mas também emocional e espiritual, e ao Professor Paulo Iscold que contribuiu ativamente com este trabalho. Estendo ainda o agradecimento a todos os orientadores e professores que passaram pela minha vida, que acabaram se tornando amigos queridos por mim.

Agradeço ainda a CAPES pelo apoio financeiro nesses quase 18 meses de trabalho. O mesmo para o Departamento de Ciência da Computação e todos os seus funcionários e colaboradores.

Por último, mas não menos importante, gostaria de agradecer aos colegas do *VeRLab* pelo apoio, atenção e pelas divertidas horas de trabalho proporcionadas. Agradeço aos Professores Chaimo e Guilherme, aos doutorandos Vilar, Pedro, Wagner e Marcelo, aos mestrandos Erickson, Lara, Paulo, Renato e Douglas, aos graduandos Leandro, Dimas, Vitor e Renato, e ao Wolmar, nosso grande colaborador.

A todos, muito obrigado!

Resumo

Neste trabalho é realizado um estudo sobre o problema da geração de trajetórias para veículos aéreos autônomos não-tripulados. O objetivo principal é prover ferramentas de planejamento para robôs aéreos, levando em conta algumas de suas principais restrições físicas de movimento. Para isso, são discutidas inicialmente algumas das técnicas mais utilizadas para o planejamento de movimento de robôs autônomos terrestres, cujo espaço de navegação é bidimensional. Duas técnicas em especial (o *Dubins' Path* e o Hodográfico Pitagoreano) são analisadas em detalhes, uma vez que essas levam em consideração as principais restrições cinemáticas estudadas neste trabalho.

O foco principal deste texto é a geração de trajetórias no espaço tridimensional, e por isso, são analisadas também algumas das técnicas mais recentemente utilizadas para esse fim. Duas novas abordagens são propostas neste trabalho. A primeira constitui uma extensão do caminho ótimo de *Dubins* para o espaço 3D. A segunda promove a unificação das duas técnicas citadas anteriormente para o caso 2D, visando produzir curvas no espaço que sejam realizáveis por um veículo aéreo específico.

Descreve-se ainda neste trabalho, a implementação de um sistema *Hardware-in-the-Loop*, utilizado para a realização de testes com o intuito de validar as metodologias propostas. Esse sistema utiliza um simulador de vôo como plataforma virtual para o estudo dos módulos de controle e planejamento de veículo aéreo autônomo real. Tais módulos são implementados em um computador de bordo, que por sua vez é conectado ao simulador de vôo via interface de rede. Assim, um modelo (matemático) aerodinâmico de um veículo virtual é utilizado como aeronave de testes para as tarefas de navegação e planejamento de trajetórias no *hardware* embarcado. Outros testes são ainda realizados utilizando-se o modelo matemático de um robô aéreo real desenvolvido pela Universidade Federal de Minas Gerais.

Abstract

In this work a study on the problem of the trajectory generation for unmanned aerial vehicles is carried through. The main objective is to provide tools of path planning for aerial robots, taking into account some of its main physical constraints of movement. For this, some of the more used techniques for motion planning of grounded robots are argued initially, whose the navigation space is bidimensional. Two techniques in special (Dubins' Path and the Pythagorean Hodograph) are analyzed in details, once these take in consideration the main studied kinematic constraints in this work.

The main focus of this text is the trajectory generation in the three-dimensional space, and therefore, some of the techniques more recently used for this end are also analyzed. Two new boardings are proposals in this work. The first one constitutes an extension of the Dubins' optimal path for the 3D space. The second promotes the unification of the two techniques previously cited for the 2D case, aiming at to produce curves in the space that are realizable for a specific air vehicle.

It describes still in this work, the implementation of a Hardware-in-the-Loop system, used for the accomplishment of tests with intention to validate the methodologies proposals. This system uses a flight simulator as virtual platform for the study of the control and planning modules of the real autonomous air vehicle. Such modules are implemented in an embedded computer, that in turn is connected to the flight simulator via net interface. Thus, a (mathematician) aerodynamic model of a virtual vehicle is used as aircraft of tests for the navigation and trajectory planning tasks in the embedded hardware. Other tests still are carried through using the mathematical model of a real aerial robot developed by the Universidade Federal de Minas Gerais.

Sumário

1	Introdução	1
1.1	Definição do Problema	1
1.2	Motivação	2
1.3	Contribuições do Trabalho	3
1.4	<i>UAVs</i> no Mundo	3
1.5	Breve Histórico dos VAANTS Brasileiros	6
1.6	Os VAANTS na UFMG	9
1.7	Organização do Texto	11
2	Trabalhos Relacionados	13
2.1	Planejamento de Movimento	13
2.2	Primeiras Técnicas de Planejamento	14
2.3	Técnicas de Planejamento para Robôs Terrestres	15
2.4	Técnicas de Planejamento para Robôs Aéreos	18
3	Metodologia	23
3.1	Planejamento de Trajetórias Bidimensionais	23
3.1.1	Formalização do Problema	24
3.1.2	<i>Dubins' Path</i> 2D	26
3.1.3	Hodográfico Pitagoreano	34
3.2	Planejamento de Trajetórias Tridimensionais	38
3.2.1	O Problema no Espaço Tridimensional	38
3.2.2	<i>Dubins' Path</i> 3D	41
3.2.3	Hodográfico Pitagoreano Tridimensional	47
3.3	Planejamento Utilizando Curvas de Bézier de 7 ^a Ordem	49
3.3.1	Caso Bidimensional	49
3.3.2	Caso Tridimensional	53
4	Introdução a Dinâmica de Aeronaves	59
4.1	Restrições Cinemáticas e Características Dinâmicas	59
4.1.1	Definições Básicas	60
4.1.2	Cálculo das Restrições	63

5 Arcabouço Experimental	67
5.1 Sistemas de Simulação - Revisão	67
5.2 Sistema <i>HWIL</i>	69
5.2.1 <i>FlightGear Flight Simulator</i>	69
5.2.2 Arquitetura de <i>Hardware</i>	70
5.2.3 Estratégias de Controle	71
5.3 Aeronave <i>AqVS</i>	77
6 Resultados	79
6.1 <i>Dubins' Path 3D</i>	79
6.2 Trajetórias Utilizando Curvas de Bézier de 7 ^a Ordem	82
6.3 Simulação de Vôo	83
6.3.1 Sistema <i>HWIL</i> e o <i>FlightGear</i>	84
6.3.2 Aeronave <i>AqVS</i>	89
7 Conclusões	91
7.1 Discussão dos Resultados	91
7.2 Direções Futuras	92
Referências Bibliográficas	95

Listas de Figuras

1.1	<i>Sperry's Aerial Torpedo</i> , considerado o primeiro <i>UAV</i> do mundo.	4
1.2	<i>V-1</i> , <i>UAV</i> alemão utilizado na II Guerra Mundial.	5
1.3	Projeto <i>Aerosonde</i> (McGeer e Vagners, 1999).	6
1.4	<i>Acauã</i> , o primeiro projeto de VAANT brasileiro, desenvolvido no ano de 1985 (Brandão et al., 2007).	7
1.5	Aeronave <i>ARARA II</i> (de Oliveira Neris, 2001).	7
1.6	<i>Aurora</i> : VAANT dirigível, desenvolvido em 1997 (Ramos, 2002).	8
1.7	Projeto <i>VANT</i> : aeronave <i>Harpia</i> , criada em 2004 (Brandão et al., 2007).	8
1.8	Projeto <i>HELIX/Gyron</i> : VAANT de asa rotativa, 1994 (Gyron, 1998).	9
1.9	<i>CEA-101 CB.1 Gaivota</i> (CEA, 2008).	9
1.10	Projeto <i>CIDA</i> (Controle e Implementação de um Dirigível Autônomo) - (VeR-Lab/DCC/UFMG).	10
1.11	Protótipo do <i>SiDeVAAN-01 "WatchDog"</i> (Campos et al., 2007).	11
1.12	Projeto <i>AqVS</i> : primeiro VAANT de pequeno porte da UFMG (Iscold, 2007).	11
2.1	Curvas do <i>Dubins' Path</i> (Shkel e Lumelsky, 2001).	16
2.2	<i>Simple Continuous Curvature path</i> (Scheuer e Fraichard, 1997).	17
2.3	Hodográfico Pitagoreano (Shanmugavel, 2007).	18
2.4	Navegação de robôs aéreos baseada em diagramas de <i>Voronoi</i> (Bortoff, 2000).	19
2.5	Extensão do <i>Dubins' Path</i> para o espaço 3D (Ambrosino et al., 2006).	20
2.6	Hodográfico Pitagoreano tridimensional (Shanmugavel, 2007).	21
3.1	Configuração arbitrária dos <i>waypoints</i> inicial e final.	25
3.2	Exemplos de <i>Dubins' Path</i> para os tipos <i>CLC</i> e <i>CCC</i> .	27
3.3	Perfil de curvatura do caminho de <i>Dubins</i> para o caso longo.	27
3.4	Perfil de curvatura do caminho de <i>Dubins</i> para o caso curto.	28
3.5	Problema representado na Figura 3.1, transformado para o novo referencial.	29
3.6	Cálculo do <i>Dubins' Path</i> para o caminho do tipo <i>CLC</i> .	31
3.7	<i>Dubins' Path</i> , resultado final.	32
3.8	Comparação entre o <i>HP</i> e o <i>DP</i> .	35
3.9	Comparação entre as funções de curvatura do <i>HP</i> e o <i>DP</i> .	35
3.10	Configuração dos <i>waypoints</i> inicial e final no espaço tridimensional.	39
3.11	Referencial <i>Frenet-Serret</i> .	40

3.12	Variação temporal do referencial <i>Frenet-Serret</i>	40
3.13	Representação dos planos utilizado no cálculo do <i>Dubins' Path</i> 3D.	43
3.14	Exemplo da variação de altitude no plano longitudinal.	44
3.15	Configuração arbitrária dos <i>waypoints</i> inicial e final, para caso 3D.	47
3.16	Comparação entre o HP 3D e o <i>Dubins' Path</i> 3D.	48
3.17	Resultado final baseado no caso do caminho longo de <i>Dubins</i>	53
3.18	Perfil de curvatura de Figura 3.17.	53
3.19	Resultado final baseado no caso do caminho curto de <i>Dubins</i>	54
3.20	Perfis de curvatura da Figura 3.19.	54
3.21	Resultado final baseado no caso do caminho curto do <i>DP</i> 3D.	56
3.22	Perfis de curvatura da Figura 3.21.	56
3.23	Perfis de torção da Figura 3.21.	56
3.24	Perfil de inclinação da Figura 3.21 para as Curvas de Bézier de 7 ^a ordem.	57
3.25	Resultado final baseado no caso do caminho longo do <i>DP</i> 3D.	57
3.26	Perfis de curvatura da Figura 3.25.	57
3.27	Perfis de torção da Figura 3.25.	58
3.28	Perfil de inclinação da Figura 3.25 para as Curvas de Bézier de 7 ^a ordem.	58
4.1	Referencial <i>NED</i> (Norte, Leste e Para baixo) e referencial do corpo da aeronave (<i>ABC</i>).	60
4.2	Orientação do referencial da aeronave em relação ao referencial <i>NED</i>	61
4.3	Momentos angulares que compõem a atitude de uma aeronave no espaço de configurações tridimensional.	61
4.4	Orientação do referencial do vento.	63
5.1	Modelo digital de uma aeronave <i>Rascal 110</i> implementado no <i>FlightGear Flight Simulator</i>	70
5.2	Computador de bordo, módulo <i>PC/104</i> , utilizado no Projeto <i>SiDeVAAN</i> (UFMG).	71
5.3	Arquitetura de comunicação entre o computador de bordo <i>PC/104</i> e o simulador de vôo <i>FlightGear Flight Simulator</i>	71
5.4	Diagrama de controle de velocidade da aeronave.	74
5.5	Diagrama de controle de altitude da aeronave.	75
5.6	Diagrama de controle de orientação da aeronave.	75
5.7	Diagrama de controle de navegação da aeronave.	76
5.8	Modelo látero-direcional do <i>AqVS</i>	78
5.9	Modelo longitudinal do <i>AqVS</i>	78
6.1	Iterações no cálculo do <i>Dubins' Path</i> 3D para o conjunto de <i>waypoints</i>	80
6.2	Resultado final do <i>Dubins' Path</i> 3D para o conjunto de <i>waypoints</i>	81
6.3	Avaliação das restrições cinemáticas para o <i>Dubins' Path</i> 3D.	81
6.4	Comparação entre o caminho de <i>Dubins</i> e as trajetórias baseadas em Béziers de 7 ^a ordem.	82

6.5	Comparação entre o caminho de <i>Dubins</i> e o Hodográfico Pitagoreano 3D.	83
6.6	Avaliação das restrições cinemáticas para as curvas de Bézier de 7 ^a ordem.	83
6.7	Avaliação das restrições cinemáticas para o Hodográfico Pitagoreano.	84
6.8	Teste para determinação do raio mínimo de curvatura da aeronave.	85
6.9	Teste para determinação da máxima taxa de torção da aeronave.	86
6.10	Resposta ao degrau do sistema de controle de altitude.	87
6.11	Controle de navegação da aeronave à altitude constante.	88
6.12	Teste de geração de trajetórias utilizando o <i>Dubins' Path</i> 3D no sistema <i>HWIL</i> . .	88
6.13	Trajetória projetada para o aeromodelo <i>AqVS</i>	89

Listas de Abreviações

<i>ABC</i>	Referencial do corpo da aeronave
<i>AqVS</i>	Avião que Voa Sozinho
<i>CEA</i>	Centro de Estudos Aeronáuticos
<i>CG</i>	Centro de Gravidade
<i>CIDA</i>	“Controle e Implementação de um Dirigível Autônomo”
<i>DP</i>	<i>Dubins' Path</i>
<i>DP 3D</i>	<i>Dubins' Path</i> Tridimensional
<i>FDMs</i>	<i>Flight Dynamic Models</i>
<i>FGFS</i>	<i>FlightGear Flight Simulator</i>
<i>HWIL</i>	<i>Hardware-in-the-loop</i>
<i>MACSIN</i>	Grupo de Modelagem, Análise e Controle de Sistemas Não-lineares
<i>NED</i>	Referencial <i>North, East and Down</i>
<i>HP</i>	Hodográfico Pitagoreano
<i>HP 3D</i>	Hodográfico Pitagoreano Tridimensional
<i>PID</i>	Controlador Proporcional, Integral e Derivativo
<i>SiDeVAAN</i>	“Simulação e Desenvolvimento de Veículos Aéreos Autônomos Não-Tripulados”
<i>SUAV</i>	<i>UAV</i> de pequeno porte
<i>UAVs</i>	<i>Unmanned Aerial Vehicles</i>
<i>UFMG</i>	Universidade Federal de Minas Gerais
<i>VAANTs</i>	Veículos Aéreos Autônomos Não-tripulados
<i>VeRLab</i>	Laboratório de Visão Computacional e Robótica

Listas de Símbolos

α	ângulo de ataque da aeronave
β	ângulo de derrapagem da aeronave
δ_a	<i>ailerons</i> da aeronave
δ_e	profundor (<i>elevator</i>) da aeronave
δ_r	leme (<i>rudder</i>) da aeronave
δ_{th}	motor (<i>throttle</i>) da aeronave
κ_{max}	curvatura máxima de um veículo não-holonômico
$\kappa(t)$	função de curvatura da curva $r(t)$
ρ_{min}	raio de curvatura mínimo de um veículo não-holonômico
Φ	ângulo de rolamento da aeronave (<i>Roll</i>)
Ψ	ângulo de orientação (guinada) da aeronave no plano XY (<i>Yaw</i>)
σ_{min}	raio de torção mínimo de um veículo não-holonômico
τ_{max}	torção máxima de um veículo não-holonômico
$\tau(t)$	função de torção da curva $r(t)$
θ_{max}	ângulo máximo de subida (ou descida) da aeronave
Θ	ângulo de arfagem da aeronave (<i>Pitch</i>)
$B(t)$	polinômio de <i>Bernstein</i>
\mathcal{B}	trajetória final calculada
\mathbf{B}	vetor binormal do referencial <i>Frenet-Serret</i>
\mathcal{D}	conjunto de curvas de <i>Dubins</i> (<i>Dubins' Set</i>)
e_{Dn}	erro derivativo filtrado
e_n	erro entre a variável de processo controlada (y_n) e o valor de referência (r_n)
k_ρ	ganho do raio de curvatura para o cálculo do <i>DP</i> 3D
K_p	ganho proporcional do controlador PID
k_s	ganho do raio de curvatura para o cálculo do <i>DP</i>
\mathbf{N}	vetor normal do referencial <i>Frenet-Serret</i>
\mathbf{p}_i	pontos de controle da curva de <i>Bézier</i>
P	velocidade angular no eixo \mathbf{X} do referencial <i>ABC</i>
\mathbf{P}_i	ponto de partida de uma trajetória $r(t)$
\mathbf{P}_f	ponto de chegada de uma trajetória $r(t)$
$\vec{P}(t)$	vetor posição do veículo no espaço cartesiano

Q	velocidade angular no eixo Y do referencial <i>ABC</i>
r_n	sinal de referência desejado para a variável controlada (y_n)
$\vec{r}(t)$	curva (trajetória) entre \mathbf{P}_i e \mathbf{P}_f
R	velocidade angular no eixo Z do referencial <i>ABC</i>
$s(t)$	função de comprimento da curva $r(t)$
\mathbf{T}	vetor tangente do referencial <i>Frenet-Serret</i>
T_d	ganho derivativo do controlador PID
T_i	ganho integral do controlador PID
T_s	intervalo de amostragem dos sinais do controle discreto
u_n	sinal de saída do controlador no instante de tempo discreto n
U	velocidade linear no eixo X do referencial <i>ABC</i>
V	velocidade linear no eixo Y do referencial <i>ABC</i>
V_T	velocidade do ar (<i>airspeed</i>) medida na aeronave
W	velocidade linear no eixo Z do referencial <i>ABC</i>
y_n	variável de processo a ser controlada
z	altitude da aeronave em relação ao nível do mar

Capítulo 1

Introdução

Neste capítulo inicial, serão apresentadas as motivações, as definições principais e as contribuições deste trabalho. Será apresentado, também, um breve histórico sobre alguns dos trabalhos previamente realizados no Brasil e no mundo, na área de Veículos Aéreos Autônomos Não-Tripulados, abordando em particular os projetos já desenvolvidos pela Universidade Federal de Minas Gerais. O objetivo é evidenciar tanto a viabilidade quanto a necessidade de estudos futuros na área da robótica aérea, sobretudo com relação ao controle, planejamento de trajetórias e navegação.

1.1 Definição do Problema

Neste trabalho discute-se o problema do planejamento de trajetórias para Veículos Aéreos Autônomos Não-tripulados (VAANTS). Tais veículos, em geral, são robôs implementados sobre plataformas aéreas, como aeronaves de asa fixa (aviões em geral), asa rotativa (helicópteros e quadrirotores), entre outras (dirigíveis, balões, etc). Os VAANTS representam uma categoria especial de robôs móveis, necessitando de tipos de sensores e técnicas de controle específicas para a realização de tarefas.

Assim como no caso dos robôs terrestres, os robôs aéreos necessitam de alguma capacidade deliberativa que lhes permita realizar tarefas de maneira autônoma, ou com o mínimo necessário de intervenção humana. Uma das tarefas mais fundamentais é o planejamento do movimento do veículo. Dado um conjunto de pontos definidos no espaço de navegação desse sistema, planejar o movimento significa determinar uma maneira de se atingir cada um desses pontos, levando-se em consideração características como as restrições de movimento do robô, o tempo gasto e a energia necessária. Ao se considerar o parâmetro tempo, por exemplo, conceituam-se dois tipos de planejamento do movimento: o planejamento de caminhos e o planejamento de trajetórias.

A definição de uma trajetória é quando o planejamento do movimento do robô envolve algum tipo de restrição temporal. Em contrapartida, isso não acontece no caminho, o que permite que o veículo se movimente pelo ambiente, livre do parâmetro de tempo. O problema tratado aqui é o planejamento de trajetórias para robôs aéreos, onde são impostas restrições

temporais a esses por meio da velocidade de seus centros de massa, além é claro das demais restrições cinemáticas e dinâmicas de locomoção dos veículos no espaço.

Dada uma seqüência de pontos-alvos (*waypoints*) definidos para o veículo no espaço, o objetivo é gerar funções de navegação que permitam ao robô alcançar todos esses pontos de maneira eficiente do ponto de vista da energia gasta, e ainda respeitando as limitações do veículo utilizado. As principais restrições cinemáticas consideradas aqui são as máximas taxas de curvatura e torção, e o máximo ângulo de subida (ou de descida) da aeronave. Aspectos dinâmicos de controle serão tratados mais adiante neste texto, e apresentam uma influência considerável sobre o planejamento da trajetória.

1.2 Motivação

Nos últimos anos diversas áreas do conhecimento, como as Engenharias Aeroespacial, Mecânica, Eletrônica e de Controle, além da Ciência da Computação e a Robótica, têm convergido para um objetivo comum: o desenvolvimento de Veículos Aéreos Autônomos Não-Tripulados, do inglês *Unmanned Aerial Vehicles (UAVs)*.

Os VAANTS são, em geral, aeronaves de pequeno e médio porte (como modelos de aviões, helicópteros ou dirigíveis) dotadas de alguma capacidade (“inteligência”) computacional que permita a realização de tarefas com certo grau de autonomia. Sejam remotamente operados, ou totalmente independentes da intervenção humana, esses veículos são robôs capazes de desempenhar diversas tarefas, que vão desde seguir trajetórias de vôo pré-programadas até a execução de missões mais complexas, como o monitoramento de grandes áreas florestais e urbanas.

A maior parte das aplicações de VAANTS nasceu dentro de órgãos militares de pesquisa, e tiveram forte desenvolvimento para fins de reconhecimento, monitoração e ações ofensivas contra postos inimigos. Atualmente, porém, outras aplicações de maior interesse para o setor civil vêm sendo alvo de pesquisa e desenvolvimento, possibilitando a utilização industrial e comercial desses veículos. Dentre diversos exemplos, podemos citar:

- monitoramento de áreas com finalidades ambientais (incêndios, desmatamentos, mapeamentos rurais);
- vigilância e monitoramento de regiões urbanas;
- busca de sobreviventes de naufrágios e desastres naturais;
- pulverização de plantações;
- inspeção de linhas de transmissão de energia elétrica;
- transporte de cargas;
- análises atmosféricas.

No combate a incêndios de grande porte, por exemplo, os riscos envolvidos para pilotos e bombeiros justificam amplamente a utilização de robôs como esses. Trata-se certamente de uma área de pesquisa bastante promissora, que demanda um esforço conjunto de especialistas em diversas áreas, e é onde a Universidade Federal de Minas Gerais (UFMG) tem tido papel importante no estudo e desenvolvimento no país.

1.3 Contribuições do Trabalho

Basicamente, são três as contribuições que se pretende oferecer com este trabalho. A primeira delas é a realização de um estudo comparativo de duas técnicas de geração de trajetórias para robôs aéreos, sendo uma delas bastante utilizada para robôs terrestres. São diversas as técnicas propostas na literatura, sendo muitas delas aplicáveis a manipuladores robóticos e robôs terrestres com movimento bidimensional. Em menor número encontram-se as alternativas para o problema da geração de trajetórias no espaço tridimensional, que são as de mais interesse para este trabalho. O objetivo é estudar a viabilidade de extensão de técnicas do 2D para o 3D.

A segunda contribuição diz respeito à apresentação de uma técnica, desenvolvida neste projeto, para a geração de trajetórias que sejam próximas ao caminho ótimo, mas que leve em consideração as restrições cinemáticas de uma aeronave. Conforme será descrito mais adiante, o melhor caminho (ou caminho mais curto) entre dois pontos, dadas as restrições não-holonômicas de um robô aéreo, torna-se irrealizável na prática, devido a problemas que serão discutidos mais adiante. Com a técnica aqui proposta, espera-se gerar um caminho que seja factível para um robô real, e que se aproxime ao máximo do caminho ótimo.

Por fim, a última contribuição é a utilização de um simulador de vôo para testes de estratégias de planejamento e controle para *UAVs*. Serão descritas mais adiante, algumas considerações necessárias para a construção de um sistema *Hardware-in-the-loop (HWIL)*, para o qual foi utilizado o mesmo computador de bordo embarcado no veículo do projeto “Simulação e Desenvolvimento de Veículos Aéreos Autônomos Não-Tripulados” (*SiDeVAAN*). A utilização de um simulador de vôo para esses casos proporciona não somente uma redução de custos de projeto, como também uma maior flexibilidade nos experimentos realizados em vôo.

1.4 *UAVs* no Mundo

A história dos veículos aéreos autônomos não-tripulados no mundo é bastante antiga. Tão antiga quanto a própria história da aviação. Uma das primeiras referências sobre o desenvolvimento de Veículos Aéreos Não-Tripulados no mundo é atribuída a um brasileiro. Registros mostram que o padre Bartolomeu Lourenço de Gusmão foi o responsável pela construção do primeiro VAANT de que se tem notícia, datado do início do século XVIII. Tratava-se de um pequeno modelo de balão de ar quente, ainda sem qualquer tipo de controle. O primeiro vôo

demonstrativo foi realizado em oito de outubro de 1709 em Lisboa, Portugal, na presença do rei João quinto e de toda a corte portuguesa (Brandão et al., 2007).

Outros documentos relatam que em 22 de Agosto de 1849, os Austríacos teriam enviado cerca de 200 balões não-tripulados à cidade de Veneza na Itália, região essa que era dominada pela Áustria na época. Esses balões carregavam bombas temporizadas para explodir sobre a cidade, e eram controlados pela simples ação do vento, o que causava uma grande incerteza com relação à precisão dos ataques. Mesmo assim, esse fato é tido como uma das primeiras intervenções militares da história utilizando-se *UAVs*.

Oficialmente, acredita-se que o primeiro *UAV* do mundo tenha sido o *Sperry's Aerial Torpedo*, construído no ano de 1916 por *Lawrence* e *Elmer Sperry* (Camacho e Yuhas, 2004). Combinando dois sistemas giroscópicos (um para estabilização e outro para direcionamento), esse veículo realizou seu primeiro vôo bem sucedido em seis de Março de 1918 em Copiague, Long Island, NY (Figura 1.1), tendo *Lawrence* como passageiro. Também chamado de *flying bomb*, esse é considerado um dos precursores dos mísseis-guiados modernos, tendo sido utilizado em testes já durante a Primeira Grande Guerra Mundial (Bone e Bolkcom, 2003).



Figura 1.1: *Sperry's Aerial Torpedo*, considerado o primeiro *UAV* do mundo.

Por mais de dez anos após o fim da Primeira Grande Guerra, o desenvolvimento de veículos aéreos não-tripulados sofreu uma pequena estagnação, e a maioria dos projetos em andamento visavam aplicações em tarefas de treinamento. Um destaque nessa época foi o *DH.82B Queen Bee*, considerado o primeiro *UAV* retornável e reutilizável do mundo. Apesar do título, esse veículo foi construído com o propósito de servir de alvo para missões de treinamento de atiradores antiaéreos da Real Marinha Britânica. O *Queen Bee*, cujo primeiro vôo foi realizado em 1935, era rádio-controlado e podia voar a uma altitude de cerca de 17000 pés (algo em torno de 5000 metros), com um raio operacional de 300 milhas (aproximadamente 480 quilômetros). Mais de 380 desses veículos foram utilizados até o ano de 1947.

Em meados da década de 40, auge da Segunda Grande Guerra Mundial, a Alemanha financiou a construção de um *UAV* que tinha a missão exclusiva de atacar alvos não-militares. Era o *Vergeltungswaffe* (a arma da vingança), mais conhecido como *V-1*, que deu início à crescente utilização de veículos aéreos não-tripulados para fins bélicos no mundo (Bone e Bolkcom, 2003). Esse veículo era uma verdadeira “bomba voadora”, apresentando uma expectativa de

vida muito curta, podendo realizar apenas um vôo bem-sucedido. O *V-1* era lançado por uma longa catapulta, e podia voar a uma velocidade de cerca de 750 km/h. Em seu primeiro vôo, essa aeronave fez um vôo programado de 240 quilômetros, matando mais de 900 pessoas (e ferindo outras 35000) em cidades inglesas no ano de 1944.

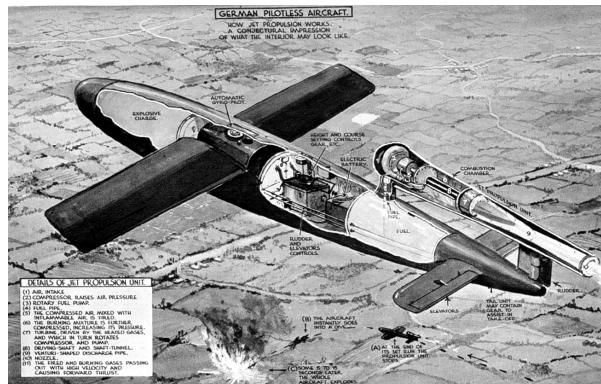


Figura 1.2: *V-1*, UAV alemão utilizado na II Guerra Mundial.

A investida alemã obrigou os países aliados a desenvolver tecnologias capazes de neutralizar as armas do *III Reich*. Um exemplo foi o desenvolvimento, por parte da Unidade Aérea Especial da Marinha dos Estados Unidos, de dois veículos remotamente controlados, o *PB4 Y-1* e o *BQ-7*. Adaptados a partir de aeronaves tripuladas de combate (o *PB4 Y-1 Liberators* e *B-17s*), esses veículos tinham uma missão bastante específica: destruir as bases de lançamento do *V-1* situadas na França dominada. Pilotos comandaram esses aviões em direção ao seus alvos até posições seguras, de onde se ejetavam, para que esses fossem tele controlados até os pontos de impacto.

Nas décadas que se seguiram, os avanços tecnológicos permitiram a utilização de *UAVs* em missões mais estratégicas, como tarefas de reconhecimento e espionagem (Bone e Bolkcom, 2003). Nos anos 60, aeronaves como o *AQM-34 Ryan Firebee* e o *D-21* (ambas produzidas nos EUA) deram apoio as tropas americanas durante a Guerra do Vietnã.

Até então, os *UAVs* existentes possuíam algumas desvantagens relativas a dificuldades de operação e manutenção. As aeronaves eram, em geral, grandes e muito caras, o que limitava o uso em operações civis e militares. Porém, na década de 1970 surgiu a era dos *UAVs* modernos, desenvolvidos para serem menores, mais baratos e mais eficientes. É por volta dessa época que Israel se destaca no desenvolvimento de veículos não-tripulados, com o *Firebee 1241* e posteriormente o *Scout*.

Dessa época até então, diversos outros projetos de veículos aéreos foram desenvolvidos ao redor do mundo, tanto no setor militar quanto civil. Vale destacar alguns deles, como o *RQ-1 Predator*, um dos mais promissores *UAVs* de guerra da década de 90; o *Helios*, uma aeronave autônoma alimentada por painéis solares e desenvolvida pela *NASA* (Agência Espacial Americana); e a aeronave Australiana *Aerosonde*, a primeira a cruzar o oceano Atlântico de maneira totalmente autônoma (McGeer e Vagners, 1999). Um histórico mais completo sobre os *UAVs* pode ser encontrado em (Camacho e Yuhas, 2004), (Valavanis, 2007)

e em diversas outras fontes, como a Internet.



Figura 1.3: Projeto *Aerosonde* (McGeer e Vagners, 1999).

1.5 Breve Histórico dos VAANTs Brasileiros

O Brasil atualmente faz parte de um seletivo grupo de países que domina o conhecimento e a tecnologia para a produção de VAANTs. Essa história começa na década de 70, quando o *CTA* (Comando-Geral de Tecnologia Aeroespacial situado em São José dos Campos - SP), em conjunto com o *IPqM* (Instituto de Pesquisas da Marinha) e o *CTEX* (Centro Tecnológico do Exército), despertaram o interesse por esses veículos. Os primeiros projetos desenvolvidos nessa área passaram a apresentar um caráter especialmente militar, onde o principal interessado era o *COMAER* (Comando Brasileiro da Aeronáutica).

Foi dessa parceria que nasceu, no ano de 1984, o primeiro projeto de um *UAV* brasileiro, chamado de Projeto *Acauã*. A aeronave não-tripulada e rádio-controlada (Figura 1.4) foi produzida com o objetivo de auxiliar estudos climáticos, ambientais e ecológicos, além de outras atividades civis, como inspeção de dutos de gás e óleo. O primeiro vôo realizado com sucesso pelo projeto foi registrado no dia 11 de outubro de 1985, e o veículo tinha a capacidade de alcançar uma velocidade de vôo de cruzeiro de até 100 km/h, transportando uma carga de 14 kg. Sua envergadura era de aproximadamente 5,1 metros, por 4,8 metros de comprimento e 120 kg de peso total (Brandão et al., 2007).

Outro projeto de grande repercussão nessa área foi o *ARARA* (Aeronaves de Reconhecimento Assistidas por Rádio e Autônomas), que teve início no fim dos anos 90. O projeto que surgiu de uma parceria em o Laboratório de Computação de Alto Desempenho do Departamento de Ciência da Computação e Estatística da USP de São Carlos, e a *EMBRAPA-CNPDIA*, tinha por finalidade a construção de veículos aéreos autônomos para o monitoramento de plantações e reservas ecológicas do país, substituindo em alguns casos, as aeronaves convencionais.

A Figura 1.5 apresenta uma das aeronaves produzidas pelo projeto *ARARA*. O *ARARA II* foi planejado de modo a minimizar os custos e o tempo de sua construção, apresentando assim uma estrutura composta de materiais bastante simples. Seu peso final era de 4,5 kg, podendo



Figura 1.4: *Acauã*, o primeiro projeto de VAANT brasileiro, desenvolvido no ano de 1985 (Brandão et al., 2007).

transportar uma carga útil inferior a 1,5 kg (de Oliveira Neris, 2001). A decolagem do veículo era realizada utilizando-se uma catapulta, e sua aterrisagem se dava por meio de um sistema pára-quedas. Além dos diversos sensores de vôo, uma pequena câmera era embarcada para ser utilizada na tarefa de monitoramento visual.

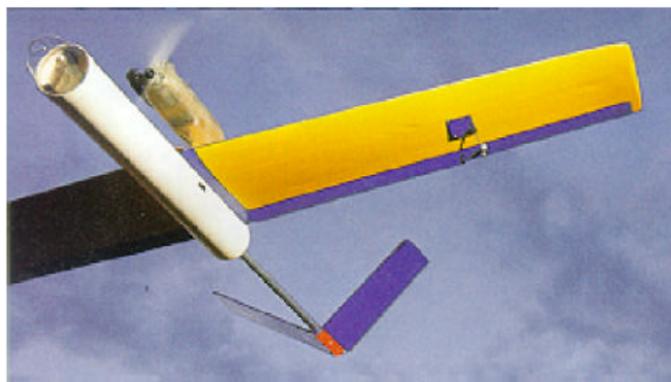


Figura 1.5: Aeronave *ARARA II* (de Oliveira Neris, 2001).

Ainda com relação às iniciativas dos setores acadêmicos, um outro projeto também de destaque foi o Projeto *Aurora* (*Autonomous Unmanned Remote Monitoring Robotic Airship*), nascido no *CenPRA* (Centro de Pesquisas Renato Archer) em 1997 na cidade de Campinas, SP. Trata-se de um dirigível autônomo, e programado para a realização de algumas tarefas de inspeção ambiental, climatológica e de biodiversidade (Ramos, 2002). A grande diferença desse para outros projetos criados até então foi a utilização de um veículo mais leve do que o ar (Figura 1.6), no lugar das já tradicionais aeronaves de asa-fixa e dos helicópteros.

O sistema era dividido em duas partes: a primeira correspondente à estação embarcada, composta pelo próprio dirigível e pelos sensores e atuadores de vôo (Maeta, 2001); e a segunda referente à estação de Terra, responsável pelo processamento externo das informações colhidas pelo robô (Mirisola, 2001). Por tratar-se de um veículo que apresenta uma dinâmica de movimentação mais lenta, suas principais aplicações eram a inspeção visual de reservas ambientais e redes de transmissão de energia elétrica.



Figura 1.6: *Aurora*: VAANT dirigível, desenvolvido em 1997 (Ramos, 2002).

Contudo, o interesse dos órgãos de pesquisa nesse setor permaneceu pouco aquecido, e durante quase duas décadas, muito pouco foi realizado. O interesse foi reavivado no ano de 2004, quando o *CTA*, o *IPqM* e o *CTEx* se uniram à empresa *Avibras Aeroespacial* (*Avibras*, 2008), no desenvolvimento do Projeto *VANT*. Esse projeto unia os principais interesses dos órgãos envolvidos para criar *UAVs* de baixo custo financeiro, alta desempenho de manobrabilidade, inteiramente comercializáveis, e que pudessem ser utilizados em missões de caráter civil. A Figura 1.7 mostra o *Harpia*, o primeiro VAANT de baixo-custo desenvolvido sobre a alcunha do projeto *VANT*. Esse veículo tinha uma missão muito simples e exclusiva, que era servir de alvo para as bases de lançamento de mísseis da Marinha.



Figura 1.7: Projeto *VANT*: aeronave *Harpia*, criada em 2004 (Brandão et al., 2007).

Modelos de aeronaves de asa rotativa também foram transformados em veículos autônomos no Brasil. Em 1991 a empresa *Gyron Sistemas Autônomos* lançou seu primeiro projeto de aeronave robótica, chamada *HELIX* (*Gyron*, 1998). O projeto foi desenvolvido em parceria com a Universidade Federal de Santa Catarina e o *CTI*, hoje *CenPRA* (e Cavalcante, 1994). Tratava-se da construção de um helicóptero não-tripulado e remotamente controlado para fins de inspeção. O protótipo apresentava uma carga útil de cerca de 10 kg, com uma autonomia de voo de 1,5 h e raio operacional de 30 km. A Figura 1.8 apresenta o segundo protótipo da aeronave *HELIX* criada no contexto desse projeto.

Outros esforços, ainda utilizando helicópteros, foram apresentados com projetos desenvol-



Figura 1.8: *Projeto HELIX/Gyron*: VAANT de asa rotativa, 1994 (Gyron, 1998).

vidos pela Universidade de Brasília (Martins et al., 2007), pela Universidade Federal do Rio Grande do Norte, e pela Escola de Engenharia de São Carlos (USP).

A Universidade Federal de Minas Gerais também investiu muito na criação de seus primeiros *UAVs*, conforme apresentado brevemente na próxima seção. Aqui ainda cabe ressaltar que muitos outros projetos brasileiros surgiram ao longo dessas mais de três décadas de avanços tecnológicos dos veículos aéreos autônomos. Um histórico mais completo pode ser encontrado em (Brandão et al., 2007).

1.6 Os VAANTS na UFMG

A UFMG é pioneira no desenvolvimento tanto de aeronaves tripuladas quanto não-tripuladas, e ao longo dos anos tem avançado muito na área de pesquisa dos robôs aéreos. Em meados da década de 1960, foi inteiramente projetada e construída no Departamento de Engenharia Mecânica da UFMG, a primeira aeronave mineira, o *CEA-101 CB.1 Gaivota* (Figura 1.9). Muitos outros projetos de aeronaves tripuladas surgiram desde então, com a criação do Centro de Estudos Aeronáuticos (CEA) do Departamento de Engenharia Mecânica.



Figura 1.9: *CEA-101 CB.1 Gaivota* (CEA, 2008).

O mesmo pioneirismo se refletiu na área dos veículos aéreos autônomos, cujo primeiro projeto foi realizado pelo Laboratório de Visão Computacional e Robótica (VeRLab) do Departamento de Ciência da Computação da UFMG. Com mais de uma década de existência e o

desenvolvimento de inúmeros projetos nas áreas de Robótica e Visão Computacional, o VeR-Lab promoveu em outubro de 2000, a conclusão do Projeto “Controle e Implementação de um Dirigível Autônomo” (*CIDA*). Tratava-se da construção de um pequeno dirigível (Figura 1.10) autônomo, capaz de se movimentar por ambientes fechados, realizando tarefas de navegação por meio de Visão Computacional (Campos e de Souza Coelho, 1999). O principal objetivo desse projeto foi investigar o problema da cooperação entre robôs aéreos e robôs terrestres, uma tarefa de pesquisa em grande atividade atualmente.



Figura 1.10: Projeto *CIDA* (Controle e Implementação de um Dirigível Autônomo) - (VeR-Lab/DCC/UFMG).

No ano de 2003, foi criado na UFMG o Projeto *SiDeVAAN*, cujo principal objetivo era o de apropriar e desenvolver tecnologia nacional capaz de equipar veículos aéreos de asa-fixa, para voar de maneira autônoma (Campos et al., 2007). O projeto contou com a participação do VeRLab, do CEA, e do Grupo de Modelagem, Análise e Controle de Sistemas Não-lineares (MACSIN) do Departamento de Engenharia Elétrica. Cada um desses grupos foi responsável pelo desenvolvimento dos softwares do sistema embarcado, projeto aerodinâmico e construção dos sensores eletrônicos de vôo, respectivamente.

O resultado final foi uma aeronave (Figura 1.11) capaz de levar uma carga de até 30 kg, dotada de uma autonomia de vôo de cerca de 4 horas e podendo atingir uma velocidade média de aproximadamente 200 km/h, tornando-se um dos primeiros VAANTs brasileiros (senão o primeiro) a serem comercializados no país.

Já em 2006, teve início o Projeto Avião que Voa Sozinho (*AqVS*), cujos objetivos eram bastante semelhantes ao do projeto anterior. Porém, a plataforma aérea utilizada nesse caso foi um motoplanador elétrico autônomo, de decolagem por arremesso manual (*hand-launched*) (Figura 1.12). Esse *UAV* de pequeno porte (*SUAV*) foi embarcado com um receptor *GPS* para localização espacial, sensor de pressão absoluta para estimar a altitude, microcâmera para a aquisição de imagens e unidade de aquisição de dados em vôo para controle de navegação (Iscold, 2007). O projeto atualmente encontra-se em pleno desenvolvimento, com centenas de vôos autônomos experimentais já realizados, resultando em dezenas de horas de testes.



Figura 1.11: Protótipo do *SiDeVAAN-01 "WatchDog"* (Campos et al., 2007).



Figura 1.12: Projeto *AqVS*: primeiro VAANT de pequeno porte da UFMG (Iscold, 2007).

1.7 Organização do Texto

Neste capítulo foi apresentada uma pequena introdução do problema tratado neste trabalho. No capítulo seguinte trata-se da revisão bibliográfica dos assuntos tratados ao longo do texto. Diversos trabalhos sobre geração de trajetórias para robôs terrestres são levantados, descrevendo as mais variadas estratégias utilizadas. Duas em particular são consideradas e analisadas em detalhes, gerando um estudo que permite a criação uma nova metodologia de geração de trajetórias tridimensionais que apresente algumas vantagens sobre as demais. Outros trabalhos específicos para robôs aéreos são também considerados.

No Capítulo 3 apresenta-se a metodologia seguida ao longo deste trabalho. A aplicação principal é a geração de trajetórias tridimensionais, cujo estudo é iniciado para o caso da movimentação 2D. O *Dubins' Path* é apresentado como solução para a determinação do caminho ótimo entre dois *waypoints* quaisquer. Uma segunda metodologia, o *Pythagorean Hodograph*, é utilizada para gerar trajetórias com perfis de curvatura mais suaves do que o *Dubins' Path*. Ambos os métodos são apresentados para o caso de geração de caminhos 3D. Uma terceira metodologia é apresentada como resultado final da pesquisa, visando unir as vantagens apresentadas por cada uma dessas duas técnicas.

No Capítulo 4 é feita uma breve introdução sobre o problema da modelagem matemática

de aeronaves de asa-fixa. Uma visão bastante sucinta sobre a dinâmica de veículos aéreos é apresentada, com o intuito de proporcionar uma visão geral sobre o planejamento e a navegação desses veículos. Essa parte é fundamental para o entendimento das relações entre as restrições cinemáticas dos veículos e os aspectos dinâmicos de controle, e abre caminho para estudos futuros mais aprofundados.

O Capítulo 5 descreve a instanciação experimental utilizada neste trabalho. Algumas publicações relativas à simulação e testes *HWIL* para aeronaves são analisadas. Dois modelos matemáticos de diferentes aeronaves são utilizados para testar as estratégias de planejamento propostas. Os passos mais importantes da construção de um sistema *HWIL* são descritos em detalhes, desde a utilização de um simulador de vôo, até a implementação de técnicas de controle para o posicionamento espacial da aeronave.

No sexto capítulo são discutidos os resultados finais do trabalho. Primeiramente são apresentados testes de planejamento de trajetórias para aeronaves com restrições escolhidas arbitrariamente. Em seguida, são apresentados os experimentos realizados para a análise das restrições cinemáticas impostas ao veículo instanciado dentro de um simulador de vôo, após a implantação de sistemas de controle em malha fechada. Os resultados referentes ao planejamento de trajetórias tridimensionais por meio do *Dubins' Path 3D* são também apresentados. Por fim, são discutidos resultados relativos à metodologia proposta para a geração de caminhos com perfis contínuos de curvatura e torção. Nesse caso são considerados tanto o problema bidimensional como o tridimensional.

O capítulo final apresenta as conclusões finais do trabalho. Obviamente, por se tratar de um assunto em uma área em desenvolvimento, muitas possibilidades de trabalhos futuros e novas aplicações surgem diante das questões levantadas ao longo do projeto.

Capítulo 2

Trabalhos Relacionados

Neste capítulo serão apresentados alguns dos principais trabalhos já realizados na área de planejamento de trajetórias, tanto para controle de robôs manipuladores quanto para a navegação de robôs móveis (terrestres e aéreos, entre outros). Metodologias importantes que surgiram inicialmente para robôs com movimentação no plano foram posteriormente estendidas para veículos no espaço tridimensional, onde os problemas de controle e navegação se tornam bem mais complexos. São com base na análise de duas características principais dos caminhos gerados (tamanho e suavidade) que serão escolhidas as melhores estratégias para o desenvolvimento da nossa abordagem.

2.1 Planejamento de Movimento

Existem três contextos distintos para o problema de planejamento de movimento: robótica, inteligência artificial e controle (LaValle, 2006). Em robótica, esse é um dos princípios mais fundamentais, geralmente abordado como o problema de conduzir o robô de um ponto a outro de uma sala evitando as possíveis colisões com obstáculos. Nesse escopo também são consideradas características como incertezas de sensores, existência de múltiplos objetos e restrições dinâmicas do sistema. O planejamento de movimento inclui entre outras características, o planejamento de trajetórias que impõem restrições temporais ao movimento do robô.

Já a inteligência artificial aborda o problema em um nível deliberativo mais elevado, incluindo no planejamento questões como tomadas de decisão, navegação topológica, teoria de jogos e replanejamento. O objetivo é tornar o planejamento de movimento uma tarefa que melhor se adeque à ambientes dinâmicos.

Por fim, do ponto de vista do controle, o planejamento de movimento é historicamente tratado como um problema de mais baixo-nível, onde os principais interesses se concentram nas características de estabilidade, otimalidade e realimentação do sistema. O foco principal é o desenvolvimento de trajetórias realizáveis em malha aberta, levando-se em consideração as restrições cinemáticas e dinâmicas do veículo utilizado.

Seja qual for a abordagem considerada, existe um fim comum ao problema do planejamento de movimento, que é o desenvolvimento de algoritmos que permitam ao robô navegar

por um determinado ambiente. Navegação é, segundo uma definição mais geral (Bowditch, 1995), o processo de planejamento e execução (controle) do movimento de um veículo, dados um ponto de saída e um ponto de chegada no espaço de navegação do mesmo. Tais algoritmos basicamente decompõem as especificações de tarefas de alto-nível em descrições de movimento em baixo-nível, levando em consideração informações como localização e geometria de obstáculos, restrições cinemáticas e geometria do robô, entre outras características, dependendo do escopo do problema tratado.

A abordagem principal seguida neste texto é bastante semelhante aquela tratada pelo controle, uma vez que o principal interesse é a geração de trajetórias para veículos aéreos, em tempo contínuo, e próximas da otimalidade (tamanho mínimo). Considera-se neste trabalho, as principais restrições cinemáticas e dinâmicas impostas aos veículos em questão, objetivando a construção de trajetórias que sejam realizáveis sob certa restrição de tempo. Questões como desvio de obstáculos e incertezas de sensores são retirados do escopo deste trabalho, podendo ser abordados em estudos futuros.

2.2 Primeiras Técnicas de Planejamento

Os primórdios do estudo do planejamento do movimento antecedem o advento dos robôs móveis (Siegwart e Nourbakhsh, 2004). De fato, esse problema foi profundamente analisado frente a utilização dos primeiros robôs manipuladores na área industrial. Um simples braço robótico com seis juntas de rotação já apresentava uma complexidade muito superior a de um robô móvel diferencial.

As primeiras pesquisas relativas a essa área originaram as técnicas de planejamento discreto, as quais são consideradas bastante simples por que, geralmente, o espaço de trabalho do robô é discreto e finito (LaValle, 2006). Os veículos considerados nesses casos são representados por modelos geométricos simples (pontos no espaço) e não existe a necessidade de equações diferenciais ou de modelos matemáticos complexos para descrever seu comportamento. Teorias probabilísticas e cálculo de incertezas também são deixados de lado em prol da simplicidade. São exemplos mais conhecidos dessas técnicas o algoritmo de *Dijkstra* (busca do caminho ótimo em grafos com um alto custo computacional) e o algoritmo *A** (busca de um caminho possivelmente ótimo com baixo custo computacional por meio da utilização de heurísticas).

Em conjunto com essas técnicas, e com bases no conceito de espaço de configurações do robô, surgiram novas abordagens, como o uso de *Road Maps* e da Decomposição por Células (Siegwart e Nourbakhsh, 2004). A técnica de *Road Map* trata o planejamento como um problema de conectividade em um grafo, que por sua vez é construído como uma rede que conecta por meio de arestas, diversos vértices do ambiente, dentre os quais estão os pontos de saída e de chegada do planejamento. Metodologias como a do grafo de visibilidade e dos diagramas de *Voronoi* são utilizadas para a construção desse grafo. Uma variação dessa técnica, conhecida como *Probabilistic Road Map* utiliza funções probabilísticas para a construção do grafo, tornando a metodologia mais interessante do ponto de vista do desvio

de obstáculos (Latombe e Kavraki, 1998).

De forma semelhante, a Decomposição por Células cria um grafo a partir da discretização do ambiente em que o robô navega. Porém, os vértices desse grafo são determinados por meio da divisão do espaço em células. Essas células são classificadas entre áreas livres e áreas ocupadas por obstáculos, e o planejamento calcula o melhor caminho através das células livres.

Em contraponto às técnicas de planejamento discreto, surgiram as técnicas de planejamento contínuo, também inicialmente aplicadas ao problema de controle do movimento de robôs manipuladores. Tais técnicas já incorporavam representações geométricas de robôs e obstáculos no ambiente por meio do estabelecimento de referenciais e de transformações no espaço cartesiano. Modelos como o de *Denavit-Hartenberg* foram utilizados para descrever a cinemática de robôs manipuladores como o *PUMA 560* (Craig, 1986).

Diversos trabalhos produzidos para a navegação de robôs móveis foram inicialmente inspirados nessas e em outras técnicas clássicas de planejamento para robôs manipuladores, como a técnica de Campos Potenciais por exemplo (Khatib, 1986). Aqui são estabelecidas funções que geram ações a partir da posição do robô no ambiente (funções potenciais). Funções de força potencial atrativa são aplicadas sobre o robô pelo ponto final a ser alcançado (*goal*), ao passo que forças repulsivas são geradas pelos obstáculos a fim de se evitar colisões durante o movimento. Um problema já conhecido dessa abordagem é a existência de mínimos locais no campo potencial resultante que impedem que o robô alcance seu objetivo.

Outras, porém, surgiram a partir da consideração de novas características no escopo do problema, como modelagem cinemática dos veículos, sensoriamento do ambiente, entre outras. Alguns desses trabalhos serão abordados na seqüência. Em especial, será considerado o problema da geração de trajetórias para robôs móveis não-holonômicos, ou seja, robôs com restrições holonômicas de movimento. A próxima seção aborda o planejamento de movimento para robôs terrestres, cujo volume da literatura é superior aos dos demais veículos.

2.3 Técnicas de Planejamento para Robôs Terrestres

Os primeiros trabalhos relativos ao planejamento de movimento para robôs móveis foram aplicados a veículos terrestres, que apresentavam apenas dois graus de liberdade (posições *x* e *y* no plano) (LaValle, 2006). A partir de muitos desses trabalhos, foram obtidas técnicas previamente desenvolvidas para os robôs manipuladores para a aplicação em veículos terrestres simples, sem quaisquer representações cinemáticas.

Com a evolução das metodologias discretas e contínuas, surgiram as técnicas de planejamento com restrições, as quais introduziram os problemas de restrições cinemáticas e dinâmicas de movimento dos veículos não-holonômicos à tarefa de planejamento. Isso remete a uma abordagem do problema de navegação mais do ponto de vista da teoria de controle, foco que direciona este trabalho. Existem diferentes tipos de modelos cinemáticos para robôs, cada qual apresentando suas próprias restrições não-holonômicas de movimento (Siegwart e Nourbakhsh, 2004). Alguns modelos de veículos mais simples passaram a incorporar outros graus de liberdade como orientação de movimento, velocidade, entre outras.

Uma das primeiras metodologia à incorporar restrições de movimento ao planejamento de caminhos pode ser vista em (Dubins, 1957) *apud* (Shkel e Lumelsky, 2001). Nesse clássico trabalho, o autor demonstra que, para um veículo não-holonômico em um espaço bidimensional, o menor caminho entre dois *waypoints* (pontos de referência para a navegação ou pontos-alvo de saída e chegada) quaisquer, com direções de movimento pré-estabelecidas, é sempre constituído de três curvas. Segundo o autor, as curvas que tocam os pontos inicial e final são sempre arcos de raio ρ_{min} , que representam a taxa mínima de curvatura permitida para um dado veículo. Já a curva central pode corresponder a um novo arco de mesma curvatura, ou uma linha reta, dependendo da distância entre os dois pontos em questão. Essa abordagem é conhecida como *Dubins' Path (DP)*, e aparece em uma grande quantidade de trabalhos relativos à geração de caminhos para veículos, devido principalmente à natureza ótima do resultado final. O gasto de energia durante a execução desse caminho também é minimizado, o que pode ser bastante interessante devido à restrições de autonomia de energia existentes na maioria dos sistemas robóticos. Detalhes sobre essa técnica serão apresentados no próximo capítulo.

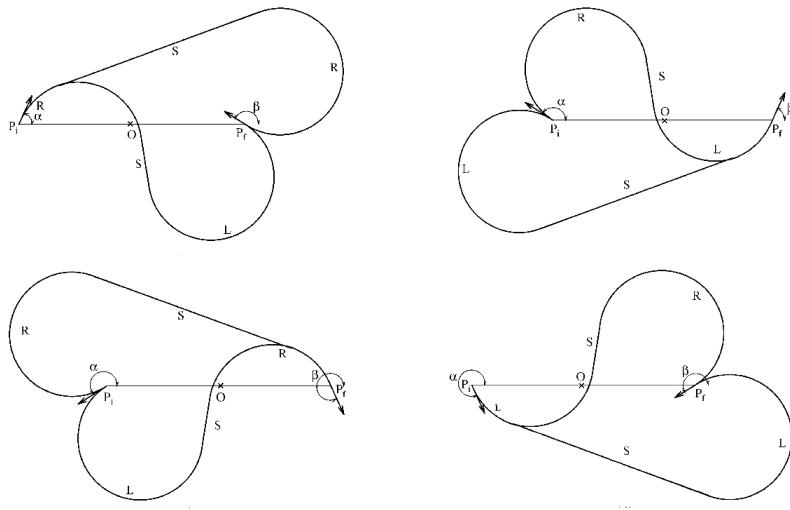


Figura 2.1: Curvas do *Dubins' Path* (Shkel e Lumelsky, 2001).

Porém, há duas desvantagens que limitam a utilização do *DP* para todos os casos de planejamento de caminhos. Uma delas, talvez menos significativa, é a complexidade computacional envolvida. Entretanto, com o rápido desenvolvimento de *hardware* e da melhoria da relação desempenho/consumo, sistemas embarcados cada vez mais poderosos tem sido utilizados em robótica. A outra desvantagem, essa bem mais séria, diz respeito à suavidade do caminho gerado. O *DP* apresenta uma característica de descontinuidade em alguns pontos de sua função de curvatura, que pode tornar o caminho gerado irrealizável por um veículo real.

Para reduzir a complexidade computacional do método de *Dubins*, (Shkel e Lumelsky, 2001) apresentam uma técnica de programação dinâmica para o cálculo do *DP* que leva em consideração duas situações; o caso de longa distância e o caso de curta distância entre os *waypoints*. Se os pontos forem considerados suficientemente distantes, comparativamente ao

tamanho do raio mínimo de curvatura ρ_{min} , então o caminho de *Dubins* pode ser diretamente obtido a partir de uma tabela previamente calculada. Caso contrário, os dois únicos caminhos possíveis para o caso curto (arcos para direita-esquerda-direita, ou esquerda-direita-esquerda) são computados e o menor deles é escolhido.

Já com relação à suavidade da curva, alguns outros métodos têm sido propostos na literatura. Em (Scheuer e Fraichard, 1997), os autores discutem um método muito semelhante à geração do *DP*, denominado *SCC* (*Simple Continuous Curvature path*), capaz de replanejar localmente a curva ótima nos pontos de transição onde ocorre a descontinuidade. Nesses trechos, a curvatura varia linearmente em relação ao comprimento do caminho (Figura 2.2). Esse por sua vez é considerado como sendo sub-ótimo, pois a suavidade alcançada nesse caso é obtida em detrimento de um pequeno aumento no comprimento do caminho de *Dubins*. Tal alternativa mostra-se bastante interessante para sistemas de movimentação bidimensional, entretanto os perfis de curvatura apresentados podem se mostrar pouco suaves, ou tão abruptas quanto o *DP* nos casos particulares em que o comprimento de uma das três curvas for muito menor do que o caminho total.

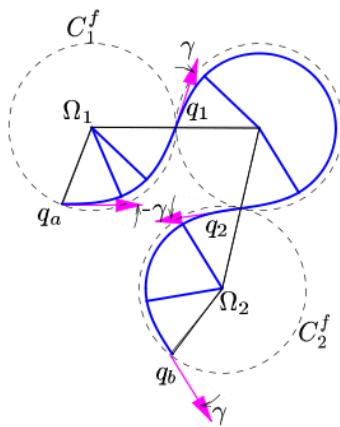


Figura 2.2: *Simple Continuous Curvature path* (Scheuer e Fraichard, 1997).

Os autores de (Maček et al., 2005) propõem um método que busca suavizar um caminho planejado por meio do sistema de controle do robô. A idéia principal é que o controlador de mais baixo nível filtre as variações mais bruscas da curvatura, proporcionando inclusive flexibilidade nas tarefas de replanejamento em tempo real. Várias publicações abordam técnicas semelhantes, porém, por não tratarem exatamente de uma técnica de geração de trajetórias, outros problemas como perturbações externas, ajuste de parâmetros de controle e instabilidade passam a ser considerados, tornando os problemas mais complexos, e as soluções produzidas menos genéricas.

Uma solução alternativa para o problema de suavidade de caminhos, inicialmente proposta em (Komoriya e Tanie, 1989), foi a utilização de curvas polinomiais denominadas *B-splines*. Tais curvas são utilizadas para interpolar *waypoints* em um caminho e apresentam como vantagens a simplicidade matemática de implementação e a continuidade de suas derivadas de mais alta ordem, o que influí diretamente no perfil da função de curvatura.

No trabalho (Shanmugavel, 2007), o autor explora com mais detalhes o problema da falta de suavidade de curvatura gerada pelo *DP*. É feita uma comparação entre esse e outros dois métodos capazes de gerar trajetórias de curvatura contínua no plano bidimensional; o *Clothoid Path* que nada mais é do que a *SCC*, e o *Pythagorean Hodograph*, cujo método é baseado na utilização de curvas de Bézier (Figura 2.3). Uma das conclusões apresentadas é que o *DP* é muito bom para veículos como helicópteros, onde a descontinuidade da curvatura do caminho causa um efeito mínimo de manobrabilidade. Outro ponto importante é que a utilização do Hodográfico Pitagoreano (HP) resolve esse problema, porém gera trajetórias que são muito maiores do que o caminho ótimo em alguns casos.

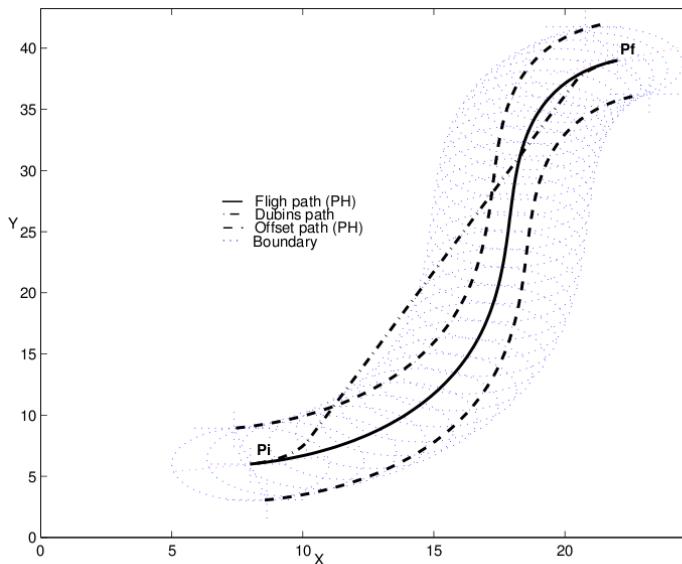


Figura 2.3: Hodográfico Pitagoreano (Shanmugavel, 2007).

Muitos dos trabalhos sobre planejamento de trajetórias publicados na literatura não levam em consideração o problema da suavidade das curvas, talvez por que para um veículo terrestre, as acelerações laterais geradas causem um impacto menos significativo sobre os sistemas de controle. Além disso, é possível que um veículo real siga a trajetória de *Dubins*, desde que reduza sua velocidade de translação (até próximo de zero) nos pontos de descontinuidade da curva, o que é praticamente impossível para veículos aéreos de asa-fixa. Algumas das estratégias propostas na próxima seção também desconsideram essa característica, mas neste trabalho, esse será o foco principal.

2.4 Técnicas de Planejamento para Robôs Aéreos

A principal diferença entre a geração de trajetórias para veículos terrestres e aéreos corresponde ao número de graus de liberdade do espaço de navegação considerado. Os primeiros se movimentam em um plano, e geralmente são caracterizados por sua posição $[x, y]$ e sua orientação ψ nesse plano. Já os veículos aéreos são modelados como corpos rígidos no espaço

tridimensional, caracterizados pela posição $[x, y, z]$ e pela orientação $[\Phi, \Theta, \Psi]$ em relação a algum referencial inercial.

Alguns dos métodos de navegação apresentados acima, para os robôs terrestres não-holônomicos, funcionam muito bem para os veículos aéreos, se sobre esses for imposta a restrição de voarem a uma altitude constante. Porém essa é uma restrição muito forte, se considerarmos que as diversas tarefas de uma missão se dão em lugares de relevo acidentado, onde a distância relativa ao solo varia consideravelmente. Para os *UAVs* é necessário o emprego de técnicas que permitam que os *waypoints* da trajetória estejam em diferentes níveis de altitude do solo e, que ainda assim, possam gerar trajetórias de mínima distância e de curvatura contínua.

Estudos iniciais nessa área aplicaram a restrição de altitude constante em busca de estudar o comportamento dos veículos aéreos frente algumas técnicas de planejamento já bem estabelecidas. Por exemplo, em (Bortoff, 2000) o autor apresenta um método baseado nas técnicas do diagrama de *Voronoi* para gerar um caminho de navegação para veículos aéreos em um cenário composto por n radares inimigos, considerados como obstáculos (Figura 2.4). Apesar de manter a restrição de altitude constante na geração do caminho, o trabalho leva em consideração questões como a dinâmica dos veículos utilizados, e as características móveis dos radares.

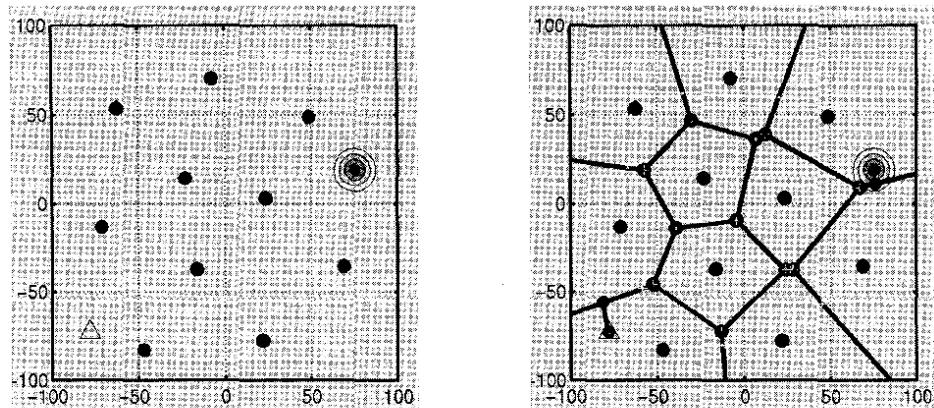


Figura 2.4: Navegação de robôs aéreos baseada em diagramas de *Voronoi* (Bortoff, 2000).

Outro trabalho aplicável a veículos aéreos, mas que desconsidera a variação de altitude é apresentado em (Jia e Vagners, 2004). Nesse trabalho os autores aplicam técnicas de Computação Evolucionária (*EC*) para resolver problemas de planejamento de caminhos em larga escala de maneira ótima, levando ainda em consideração a existência de áreas proibidas no ambiente. Assim como no trabalho anterior, essa metodologia não apresenta a característica de planejamento com restrições, já que não leva em consideração nenhum tipo de modelo cinemático do robô.

Alguns trabalhos empregam diretamente a técnica de *Dubins* para a navegação de robôs aéreos. Como exemplo, os autores de (Jeyaraman et al., 2004) fazem uma discussão sobre a utilização do *DP* no planejamento de caminhos para múltiplos *UAVs*. Um algoritmo de

controle reativo é incorporado ao método de planejamento, criando uma forma de arquitetura híbrida entre os vários veículos. A conclusão é a de que é possível obter bons resultados para a navegação coordenada de um time de veículos aéreos utilizando-se o *DP*.

Um dos primeiros trabalhos a considerar a expansão da técnica de *Dubins* para o espaço tridimensional é (Ambrosino et al., 2006). Basicamente, os autores desacoplam o problema da variação de altitude, dividindo o caminho tridimensional em três partes. A primeira e a última parte servem para orientar a aeronave segundo as configurações iniciais e finais dos *waypoints* especificados. A curva do centro serve para a realização do caminho de *Dubins* em um plano inclinado determinado pela variação de altitude desejada, mantendo-a constante durante todo o caminho. Essa solução é bastante interessante do ponto de vista prático, pois torna o cálculo da variação de altitude bastante simples, levando-se em conta a restrição de máxima taxa de subida de um veículo. Porém o trabalho desconsidera questões como a descontinuidade da taxa de curvatura. A trajetória gerada por essa técnica também tende a se afastar do comprimento ótimo em função de uma dinâmica muito lenta na taxa de subida de uma aeronave. Além disso, não existem garantias de que a técnica sempre funcione para qualquer configuração de *waypoints*.

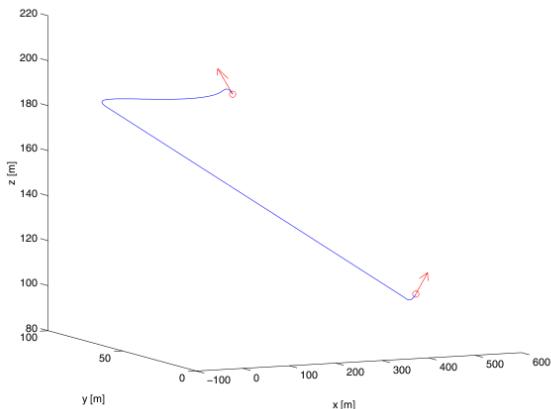


Figura 2.5: Extensão do *Dubins' Path* para o espaço 3D (Ambrosino et al., 2006).

Os autores de (Hwangbo et al., 2007) apresentam um resultado igualmente interessante do ponto de vista da variação de altitude. Apesar de o foco principal do trabalho ser o desvio de obstáculos em tempo real, o planejamento de caminhos antevê a necessidade de distorcer o caminho de *Dubins*, retirando-o do plano se necessário, para gerar caminhos de tamanho reduzido. O problema dessa metodologia é que a orientação longitudinal da aeronave fica restrita à zero nos *waypoints*. Isso pode não ser tão ruim, se uma missão necessitar que o veículo esteja paralelo ao solo durante a passagem por esses pontos, para, por exemplo, tirar fotos de um alvo. O número de aplicações, entretanto, acaba restrito por essa característica.

Novamente em (Shanmugavel, 2007), o autor discute ainda a extensão dos métodos utilizados para a geração de caminhos e trajetórias no espaço 2D (*DP*, *Clothoid Path* e *HP*) para o espaço 3D, onde além da restrição de função de curvatura, passa a ser considerada também, a restrição para a torção espacial. A mesma conclusão com relação ao melhor desempenho

do HP para a navegação 2D foi obtida aqui, e é válida especialmente para a utilização de múltiplos robôs. O trabalho se aprofunda ainda mais nas questões de restrição do movimento da aeronave, levando em consideração não apenas a suavidade do perfil de curvatura da trajetória, mas também a função de torção da curva. A representação de um Hodográfico Pitagoreano tridimensional é apresentada na Figura 2.6.

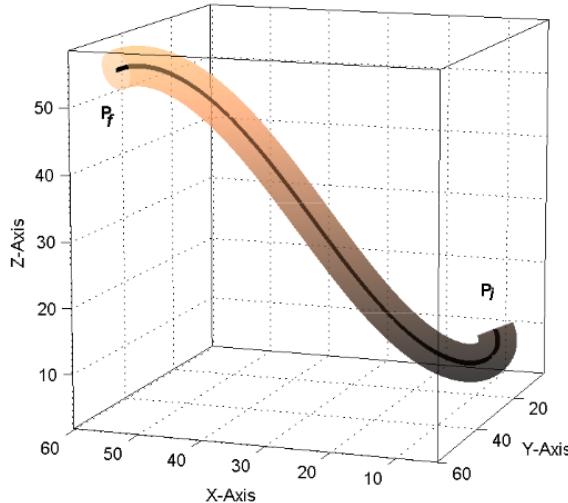


Figura 2.6: Hodográfico Pitagoreano tridimensional (Shanmugavel, 2007).

No capítulo seguinte será apresentada a metodologia proposta neste trabalho. A técnica do caminho de *Dubins* será discutida em mais detalhes, pois corresponde à etapa inicial do estudo da geração de trajetória com perfis contínuos de curvatura e torção que se pretende propor. Uma nova restrição (o ângulo máximo de subida da aeronave) é incorporada ao problema de planejamento, sendo considerada de vital importância para o caso de veículos aéreos. O capítulo tem ainda dois aspectos fundamentais: a proposição de uma extensão do *DP* para o espaço tridimensional, e a apresentação de uma nova metodologia de planejamento baseada em curvas de Bézier.

Capítulo 3

Metodologia

Este capítulo descreve a metodologia concebida para se investigar o problema da geração de trajetória para veículos aéreos autônomos não-tripulados sobre restrições cinemáticas de navegação. Inicialmente são descritas duas técnicas de planejamento em espaços contínuos, o *DP* e o *HP*, visando introduzir alguns conceitos importantes para o desenvolvimento do trabalho. O primeiro método é diretamente aplicável a aeronaves cuja variação de orientação e posição são independentes entre si (caso de helicópteros, dirigíveis e quadrirotores). Porém, esse não serve para os demais veículos aéreos, devido ao problema de descontinuidade do caminho de *Dubins*. O segundo método gera trajetórias que podem ser realizadas também por aeronaves de asa-fixa (aviões e planadores). Porém, esse apresenta problemas que impedem o cálculo de bons resultados em algumas situações, tornando-o um método falho. Ao final, descreve-se uma nova técnica que utiliza as duas metodologias citadas para gerar trajetórias que possam ser realizadas por todos os veículos aéreos citados, com algumas vantagens sobre os métodos anteriores.

3.1 Planejamento de Trajetórias Bidimensionais

O primeiro passo no desenvolvimento do trabalho refere-se ao estudo de técnicas de planejamento de trajetórias para veículos no caso bidimensional. O movimento de um robô no plano é caracterizado por meio de três graus de liberdade: dois valores de posição (x e y), e um valor de orientação (ψ) nesse plano.

Ao longo desta seção, duas técnicas são apresentadas em detalhes. O *DP*, discutido pela primeira vez em (Dubins, 1957) *apud* (Shkel e Lumelsky, 2001) apresenta o caminho mais curto entre dois *waypoints* e possui aplicações em diversos trabalhos além da área de Robótica Móvel. O cálculo do *HP* é uma técnica mais recente, apresentada em detalhes em (Shanmugavel, 2007), que visa gerar trajetórias com perfis contínuos em suas funções de curvatura entre dois *waypoints*, corrigindo assim a principal falha da metodologia de *Dubins*. Os conceitos discutidos aqui servirão de base para a posterior apresentação das metodologias de planejamento de movimento no espaço tridimensional na próxima seção.

3.1.1 Formalização do Problema

Existe uma diferença principal entre um caminho e uma trajetória. O caminho é a especificação do movimento de um corpo no espaço. Em outras palavras, é uma função espacial que define uma seqüência de posições a serem seguidas. O *DP*, por exemplo, representa um caminho (ótimo) entre dois pontos orientados em um plano. Já uma trajetória é a especificação do movimento de um corpo ao longo do tempo. Dado um caminho a ser seguido no espaço, a trajetória define uma função temporal que descreve o movimento do corpo através desse caminho. O *HP* é um exemplo de trajetória.

Os diversos tipos de veículos aéreos existentes são capazes de imprimir diferentes perfis de velocidade translacional durante a execução de um caminho, podendo assim executar diferentes tipos de trajetórias. Alguns desses veículos (como helicópteros, quadrirotos e dirigíveis) conseguem permanecer parados no ar (velocidade zero). Outros, porém necessitam de um valor mínimo de velocidade a fim de se manterem suspensos no ar (caso das aeronaves de asa-fixa). No intuito de simplificar a abordagem utilizada para a geração de trajetórias para veículos aéreos, considera-se que a velocidade dos veículos durante a realização do caminho é sempre constante e diferente de zero.

Assim, considerando $\vec{r}(t)$ como a curva que representa uma trajetória de um veículo, então $\dot{\vec{r}}(t)$ apresenta módulo constante ao longo do tempo. Já $s(t)$, que descreve a função de comprimento da curva $\vec{r}(t)$, pode ser calculada por meio da Equação (3.1), variando linearmente em função do tempo, com $\dot{s}(t) = \|\dot{\vec{r}}(t)\|$.

$$s(t) = \int_{t_i}^{t_f} \|\dot{\vec{r}}(t)\| \delta t. \quad (3.1)$$

Dados os pontos de partida e chegada no espaço de atuação de um veículo, planejar uma trajetória significa deliberar sobre a melhor maneira de conduzir esse veículo, levando-se em conta suas restrições de movimento. Sabe-se que o caminho mais curto entre dois pontos no plano cartesiano é dado por uma reta. Assim sendo, considerando-se as técnicas de planejamento discreto vistas no capítulo anterior, onde robôs são modelados como pontos sem restrições holonômicas, o caminho ótimo entre dois pontos é o mais simples possível.

Entretanto, quando se passa a considerar as restrições cinemáticas e dinâmicas dos veículos utilizados, tais técnicas se mostram incompletas. Automóveis e aeronaves de asa-fixa, por exemplo, não são capazes de imprimir velocidades laterais de forma independente, necessitando de manobras (algumas vezes complexas) para alcançar determinadas configurações no espaço. Esses são exemplos de sistemas não-holonômicos, que apresentam, entre várias características, uma velocidade mínima de translação e um raio mínimo de curvatura. Essa última é importante, pois limita o valor máximo de curvatura que uma dada trajetória ou caminho pode possuir para que seja realizada por um robô.

De uma maneira mais formal, podemos definir o problema do planejamento de trajetórias como sendo a escolha de uma curva $\vec{r}(t)$ em \mathbb{R}^2 , que leve o robô de um *waypoint* inicial (\mathbf{P}_i) a um *waypoint* final (\mathbf{P}_f) e que simultaneamente respeite as restrições de movimento do veículo. Ou seja,

$$\begin{aligned}\mathbf{P}_i(x_i, y_i, \psi_i) &= \vec{r}(t_i) \\ \mathbf{P}_f(x_f, y_f, \psi_f) &= \vec{r}(t_f)\end{aligned}\quad (3.2)$$

onde os x' s e y' s correspondem as coordenadas do vetor posição do veículo em relação a um determinado referencial sobre o plano, e os ψ' s correspondem a orientação do mesmo em relação ao eixo **X** tomados positivamente no sentido anti-horário.

Na Figura (3.1) é possível observar um exemplo bastante simples do problema tratado. As coordenadas x_i e y_i representam a posição do robô em um instante de tempo t_i , e a direção segundo a qual o robô deve deixar esse ponto, devido a restrições holonômicas, é representada por ψ_i . O veículo deve então executar uma trajetória que o leve à x_f e y_f chegando a esse ponto com uma direção ψ_f no tempo t_f . Nesse caso ainda, $\vec{r}(t)$ apresenta módulo constante e diferente de zero.

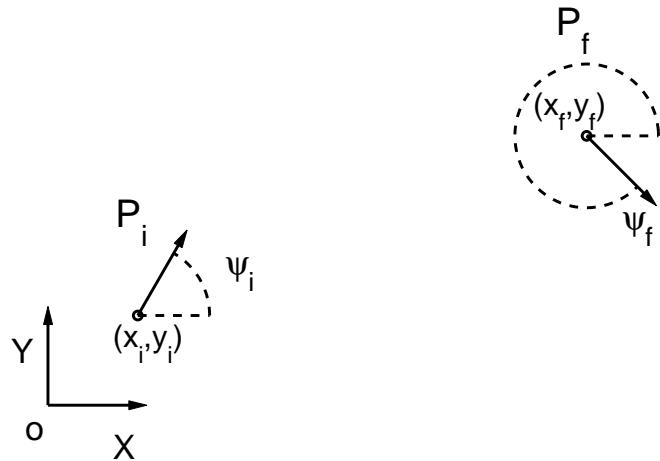


Figura 3.1: Configuração arbitrária dos *waypoints* inicial e final.

Para o caso do movimento em um plano, a principal restrição que atua sobre um veículo não-holonômico é a curvatura máxima (κ_{max}) possível de ser executada. Cada veículo possui um raio mínimo de curvatura que pode ser realizado, devido as suas configurações dinâmicas e cinemáticas, caracterizado por ρ_{min} . Logo, a curva $\vec{r}(t)$ pode ser considerada uma trajetória realizável desde que não contenha nenhum trecho cujo raio de curvatura seja inferior ao mínimo admissível. De uma outra forma, seja $\kappa(t)$ a função que representa a curvatura de $\vec{r}(t)$; pode-se então representar a restrição do veículo segundo a Equação (3.3).

$$|\kappa(t)| \leq \kappa_{max}, \quad (3.3)$$

onde

$$\kappa_{max} = \frac{1}{\rho_{min}}.$$

Além disso, espera-se que a função de curvatura de toda a trajetória seja contínua, a fim de se evitar perturbações causadas por forças laterais que podem desestabilizar o sistema de

controle do veículo. Matematicamente, podemos determinar que se $\vec{r}(t)$ possui derivadas de primeira e segunda ordem contínuas, então a função de curvatura também será contínua ao longo de toda a trajetória.

Sendo a curva $\vec{r}(t)$ representada por um vetor em \mathbb{R}^2 composto por $[x(t), y(t)]$, então a função de curvatura $\kappa(t)$ pode ser expressa por meio da Equação (3.4).

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{\left(\sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}\right)^3}, \quad (3.4)$$

com

$$\|\dot{\vec{r}}(t)\| > 0 \quad \forall \quad t_i \leq t \leq t_f.$$

A restrição imposta a $\dot{\vec{r}}(t)$ representa outra condição de regularidade para $\vec{r}(t)$, que pode ser fisicamente interpretada como a incapacidade de um veículo (como uma aeronave de asa-fixa) em reduzir sua velocidade a zero ao realizar uma trajetória.

Além da função de curvatura de $\vec{r}(t)$ outro ponto importante é o comprimento da trajetória. Os parâmetros t_i e t_f correspondem aos instantes de tempo da passagem do veículo pelos pontos \mathbf{P}_i e \mathbf{P}_f respectivamente. Essa é uma métrica importante, pois se espera gerar trajetórias com comprimentos próximos do mínimo valor possível, a fim de economizar a energia gasta na realização de tarefas. Para um espaço em \mathbb{R}^2 , o comprimento da curva $\vec{r}(t)$ pode ser obtido integrando-se numericamente a Equação 3.5, que por sua vez, é derivada da Equação 3.1.

$$s(t) = \int_{t_i}^{t_f} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} \, dt. \quad (3.5)$$

Em suma, o problema considerado nesta seção é a geração de trajetórias para veículos aéreos com velocidade de translação constante, capazes de realizar curvas com raio mínimo limitado. Tendo apresentado as principais características do problema tratado, será estudada uma técnica bastante utilizada para a geração de caminhos no espaço bidimensional; o caminho de *Dubins*. Suas principais vantagens e desvantagens serão tratadas a seguir.

3.1.2 *Dubins' Path 2D*

No trabalho (Dubins, 1957) *apud* (Shkel e Lumelsky, 2001), o autor discute a geração de curvas de tamanho mínimo com restrições de curvatura, dados os pontos e as tangentes (ou orientações) conforme descrito na Figura 3.1. A principal conclusão é a de que o caminho mais curto entre as poses \mathbf{P}_i e \mathbf{P}_f , pode ser obtido por meio da junção de três curvas simples, denominadas de Conjunto de *Dubins* (ou *Dubins' Set*), representado por \mathcal{D} . A seguir serão apresentadas as principais características do método, e ainda uma simples formalização matemática utilizada para a geração do caminho.

3.1.2.1 Principais Características

Há dois tipos de classificação de caminhos de *Dubins*, especificados em função da configuração dos pontos escolhidos. O tipo *CLC* representa um caminho composto por semicírculos C de raio ρ_{min} em suas duas extremidades, unidos por uma reta L , que é tangente a esses arcos. Já o tipo *CCC* corresponde à união de três arcos de mesmo raio. A Figura 3.2 apresenta dois exemplos desses tipos de curvas de *Dubins* geradas entre dois pontos, onde as setas mostram o sentido do movimento. Para esses dois casos, foi escolhido um valor arbitrário para ρ_{min} .

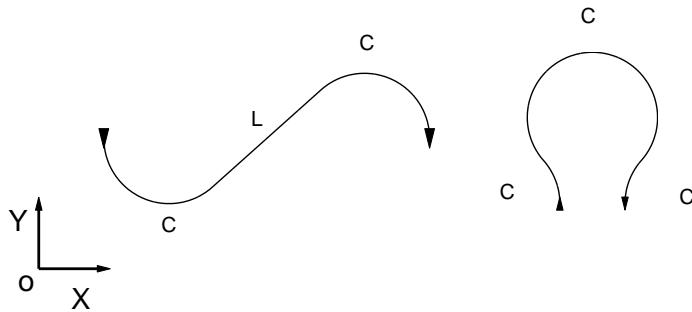


Figura 3.2: Exemplos de *Dubins' Path* para os tipos *CLC* e *CCC*.

Para se calcular a curvatura de uma curva em função do comprimento da mesma ($\kappa(s)$) no plano, pode ser utilizada a Equação 3.6, onde ψ corresponde à orientação tangente em cada ponto dessa curva ao longo de s . Por meio dessa expressão, foram calculadas as funções de curvatura para cada um dos exemplos apresentados acima, cujos resultados são mostrados nas Figuras 3.3 e 3.4, respectivamente.

$$\kappa(s) = \frac{\delta\psi}{\delta s}. \quad (3.6)$$

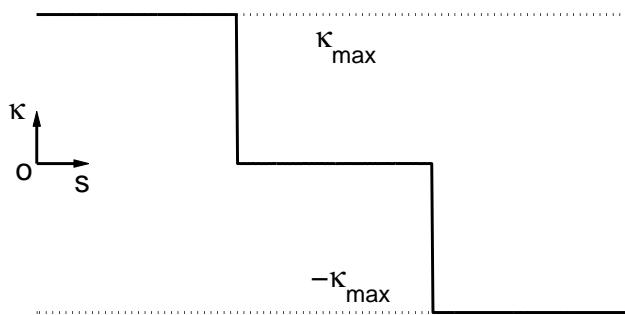


Figura 3.3: Perfil de curvatura do caminho de *Dubins* para o caso longo.

A classificação entre um desses dois tipos de caminhos depende das especificações determinadas para \mathbf{P}_i e \mathbf{P}_f . Basicamente, se a distância euclidiana entre os pontos de chegada e saída do caminho for suficientemente grande, de modo que os arcos inicial e final não apresentam qualquer intersecção, então o caminho é do tipo *CLC*, também chamado de Caminho Longo de *Dubins*. Do contrário, aplica-se o caminho do tipo *CCC*, conhecido como o Caminho Curto de *Dubins*.

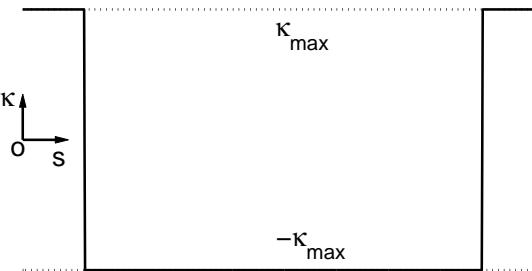


Figura 3.4: Perfil de curvatura do caminho de *Dubins* para o caso curto.

A curvatura instantânea de uma curva é dada pelo inverso do valor do raio de curvatura naquele instante de tempo, o que remete à restrição para a curva $\vec{r}(t)$ apresentada na Equação 3.3. O parâmetro κ_{max} , representado nas figuras por linhas pontilhadas corresponde à curvatura máxima do veículo. Esse valor não deve ser excedido, qualquer que seja o caminho gerado.

Mais especificamente, no caso do caminho de *Dubins*, as curvas geradas com o valor de ρ_{min} , produzem arcos C_s cuja curvatura máxima é sempre equivalente ao máximo aceitável, ou seja, κ_{max} . Observa-se, porém, a existência de descontinuidades na função de curvatura desses caminhos, ou mais especificamente, nas faixas onde ocorrem as junções das sub-curvas do caminho final. Em outras palavras, a união de curvas que apresentam diferentes valores constantes de curvatura, geram variações bruscas em $\kappa(s)$. Como exemplo, o caso apresentado de caminho longo de *Dubins* é caracterizado pela junção de arcos de curvatura com módulo igual à $\frac{1}{\rho_{min}}$ e uma reta de curvatura igual à zero. No caso em que o caminho gerado é do tipo *CCC*, as variações de aceleração são ainda mais bruscas, pois alternam subitamente de guinadas da esquerda para direita, ou vice-versa. Tais efeitos causam acelerações laterais que são indesejáveis do ponto de vista de estabilidade do sistema de controle de um veículo. Assim sendo, não se deve utilizar o *DP* da maneira direta especialmente por causa desse efeito de descontinuidade na função de curvatura.

Outra característica diz respeito ao sinal da função de curvatura, o que determina o sentido (direita ou esquerda) da curva executada. A convenção adotada neste trabalho é que curvas para a direita apresentam sinal negativo e, para esquerda, sinal positivo.

A seguir será apresentada a metodologia proposta para a geração do caminho de *Dubins*. Essa etapa é importante, pois seu resultado será posteriormente utilizado para o cálculo de trajetórias que apresentem funções contínuas de curvatura ao longo do tempo.

3.1.2.2 Geração do Caminho de *Dubins* 2D

O cálculo do *DP* apresentado a seguir é baseado nas técnicas propostas em (Dubins, 1957) e (Shkel e Lumelsky, 2001), sendo esse último especialmente utilizado para classificar o tipo de caminho escolhido. O primeiro passo corresponde ao cálculo da distância euclidiana entre os pontos de saída e chegada do caminho, conforme apresentado na Equação 3.7. A norma do vetor que liga esses pontos é dividida pelo raio mínimo de curvatura do veículo, no intuito de generalizar o algoritmo utilizado para a definição do caminho, e para simplificar os cálculos

posteriormente.

A distância d entre os pontos avaliados em relação à taxa de curvatura será utilizado para determinar o tipo de caminho de *Dubins* (*CLC* ou *CCC*) a ser calculado.

$$d = \frac{\sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}}{\rho_{min}}. \quad (3.7)$$

Para o caso de veículos que apresentam raio mínimo de curvatura igual a zero, o teste é desnecessário, pois o caminho ótimo gerado será sempre do tipo *CLC*. Em outras palavras, tais veículos podem girar em torno do próprio eixo (ρ_{min} nulo), de modo que possam apontar para $[x_f, y_f]$ sem sair de $[x_i, y_i]$, realizando assim um caminho retilíneo entre esses pontos. Algumas aeronaves, como helicópteros, quadrirotores e alguns dirigíveis possuem essa capacidade.

Na sequência, o sistema é rotacionado segundo o ângulo formado pelo vetor $\overrightarrow{\mathbf{P}_i \mathbf{P}_f}$ e o eixo \mathbf{X} do plano XY . Subtraindo-se esse ângulo dos valores de orientação ψ_i e ψ_f dos pontos inicial e final, obtém-se as novas orientações ψ_i' (ponto de saída) e ψ_f' (ponto de chegada) para o novo referencial conforme mostram as Equações 3.8 e 3.9, respectivamente.

$$\psi_i' = \psi_i - \text{atan}2\left(\frac{y_f - y_i}{x_f - x_i}\right), \quad (3.8)$$

$$\psi_f' = \psi_f - \text{atan}2\left(\frac{y_f - y_i}{x_f - x_i}\right). \quad (3.9)$$

Na Figura 3.5 é possível observar a nova representação do problema apresentado na Figura 3.1, em relação ao novo referencial $X'Y'$. Nessa nova configuração, \mathbf{P}_i corresponde a pose $(0, 0, \psi_i')$ e \mathbf{P}_f a $(d, 0, \psi_f')$, e o problema se reduz a geração de caminhos ótimos, onde o raio mínimo de curvatura é unitário.



Figura 3.5: Problema representado na Figura 3.1, transformado para o novo referencial.

Se os semicírculos inicial e final que compõem o caminho se interceptam em mais de um ponto no plano, significa que a distância entre \mathbf{P}_i e \mathbf{P}_f não pode ser interpolada linearmente, gerando um caminho do tipo *CLC*. Dadas essas condições de contorno, é possível demonstrar que o valor de d para o qual os arcos inicial e final se interceptam no plano normalizado, pode ser obtido por meio da Equação 3.10 (Shkel e Lumelsky, 2001).

$$d_{min} = \sqrt{4 - (|\cos \psi_i'| + |\cos \psi_f'|)^2} + |\sin \psi_i'| + |\sin \psi_f'|. \quad (3.10)$$

Com base nesse valor, utiliza-se o critério a seguir para determinar qual tipo de caminho é mais adequado:

$$d \quad \begin{cases} > d_{min} & CLC \text{ ou caminho longo ,} \\ \leq d_{min} & CCC \text{ ou caminho curto .} \end{cases}$$

É importante ressaltar que a transformação feita para o referencial $X'Y'$ serve apenas para classificar o tipo de caminho a ser utilizado. Uma vez escolhido o tipo de caminho, retorna-se ao referencial original.

O passo seguinte é determinar qual dos caminhos possíveis a um mesmo tipo é mais curto. Para o caso do caminho longo, existem quatro possibilidades para se compor o caminho, sendo que uma delas é a ótima. O conjunto de *Dubins* \mathcal{D} é composto pela combinação de curvas C que giram para a esquerda (L) e para a direita (R), unidas por uma reta tangente. Ou seja, $\mathcal{D} = [LSL, RSR, LSR, RSL]$. Da mesma forma, o caso de caminho curto é composto por apenas duas possíveis combinações, sendo elas $\mathcal{D} = [RLR, RLR]$.

Para o caso longo, o caminho de *Dubins* mais curto, dentre os quatro possíveis apresentados, é determinado utilizando-se uma técnica de programação dinâmica, apresentada em (Shkel e Lumelsky, 2001). Essa técnica visa principalmente reduzir o custo computacional empregado na tarefa, eliminando a necessidade do cálculo dos quatro caminhos, para determinar a melhor opção. Os autores apresentam uma tabela de onde se pode obter diretamente o resultado final, em função dos quadrantes a que pertencem os ângulos ψ_i^* e ψ_f^* , no plano coordenado da Figura 3.5.

Uma vez que se conheçam as direções de movimento das curvas inicial e final, e lembrando que tais curvas são unidas por meio de uma reta tangente comum, o próximo passo é determinar o ponto central de cada uma das curvas. As Equações 3.11 e 3.12 fornecem os centros dos arcos inicial e final do caminho de *Dubins*, ponderados pelo valor de ρ_{min} .

$$\mathbf{c}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \rho_{min} \begin{bmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} \cos \psi_i \\ \sin \psi_i \end{bmatrix}, \quad (3.11)$$

$$\mathbf{c}_f = \begin{bmatrix} x_f \\ y_f \end{bmatrix} + \rho_{min} \begin{bmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} \cos \psi_f \\ \sin \psi_f \end{bmatrix}. \quad (3.12)$$

Nesse caso, γ é um ângulo que depende do sentido de movimento da curva analisada. Se os arcos inicial ou final giram para a esquerda (L), então γ assume valor $\frac{\pi}{2}$. Caso contrário, o valor é $-\frac{\pi}{2}$.

Após definir o centro e o sentido de cada curva, é preciso determinar o comprimento do arco gerado, ou em outras palavras, a varredura angular de cada semicírculo. Chamaremos de ψ_a o ângulo do vetor formado entre o centro da primeira curva (\mathbf{c}_i) e o ponto de partida (x_i, y_i) no plano, e ψ_b o ângulo do vetor formado entre \mathbf{c}_f e o ponto em que a reta tangente S toca a curva (Figura 3.6). Tais ângulos determinam a varredura angular do primeiro semicírculo e precisam ser determinados.

Raciocínio semelhante se estende ao arco final, onde ψ_c é o ângulo do vetor composto

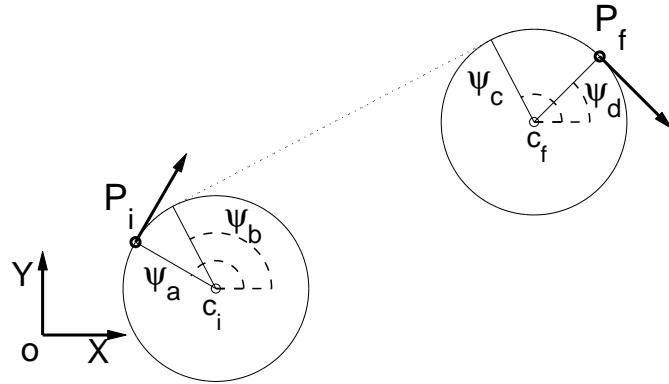


Figura 3.6: Cálculo do *Dubins' Path* para o caminho do tipo *CLC*.

pelo centro da segunda curva (\mathbf{c}_f) e o ponto em que a reta tangente S toca a curva. Já ψ_d corresponde ao ângulo do vetor formado por \mathbf{c}_f e o ponto final (x_f, y_f) .

Dadas as quatro possibilidades anteriores, podemos utilizar as Equações 3.13 e 3.14 para calcular os ângulos dos pontos de chegada e partida para qualquer que seja o tipo de caminho longo.

$$\psi_a = \text{atan}2\left(\frac{y_i - \mathbf{c}_i(y)}{x_i - \mathbf{c}_i(x)}\right), \quad (3.13)$$

$$\psi_d = \text{atan}2\left(\frac{y_f - \mathbf{c}_f(y)}{x_f - \mathbf{c}_f(x)}\right). \quad (3.14)$$

Os ângulos ψ_b e ψ_c determinarão os pontos tangentes da reta S e os semicírculos inicial e final. Para serem determinados exatamente, ψ_b e ψ_c dependem das direções de movimento das curvas. Para cada uma das possibilidades de caminhos, são apresentadas, a seguir, as equações para determinação desses valores:

- Caso *LSL*:

$$\psi_b = \text{atan}2\left(-\frac{\rho_{min}}{\|\mathbf{c}_f - \mathbf{c}_i\|} \frac{\delta c(x)}{\delta c(y)}\right), \quad (3.15)$$

$$\psi_c = \psi_b. \quad (3.16)$$

- Caso *RSR*:

$$\psi_b = \text{atan}2\left(-\frac{\rho_{min}}{\|\mathbf{c}_f - \mathbf{c}_i\|} \frac{\delta c(x)}{\delta c(y)}\right), \quad (3.17)$$

$$\psi_c = \psi_b. \quad (3.18)$$

- Caso *LSR*

$$\psi_b = \text{atan}2 \left[\frac{\rho_{min}}{\|\mathbf{c}_f - \mathbf{c}_i\|} \frac{\delta c(x) \cos(-\xi) - \delta c(y) \sin(-\xi)}{\delta c(x) \sin(-\xi) - \delta c(y) \cos(-\xi)} \right], \quad (3.19)$$

$$\psi_c = \text{atan}2 \left[\frac{\rho_{min}}{\|\mathbf{c}_f - \mathbf{c}_i\|} \frac{\delta c(x) \cos(\pi - \xi) - \delta c(y) \sin(\pi - \xi)}{\delta c(x) \sin(\pi - \xi) - \delta c(y) \cos(\pi - \xi)} \right]. \quad (3.20)$$

- Caso *RSL*

$$\psi_b = \text{atan}2 \left[\frac{\rho_{min}}{\|\mathbf{c}_f - \mathbf{c}_i\|} \frac{\delta c(x) \cos(\xi) - \delta c(y) \sin(\xi)}{\delta c(x) \sin(\xi) - \delta c(y) \cos(\xi)} \right], \quad (3.21)$$

$$\psi_c = \text{atan}2 \left[\frac{\rho_{min}}{\|\mathbf{c}_f - \mathbf{c}_i\|} \frac{\delta c(x) \cos(\xi - \pi) - \delta c(y) \sin(\xi - \pi)}{\delta c(x) \sin(\xi - \pi) - \delta c(y) \cos(\xi - \pi)} \right]. \quad (3.22)$$

onde

$$\delta c(x) = \mathbf{c}_f(x) - \mathbf{c}_i(x)$$

$$\delta c(y) = \mathbf{c}_f(y) - \mathbf{c}_i(y)$$

Nas relações descritas anteriormente, ξ é calculado pela Equação 3.23.

$$\xi = \sin^{-1} \left(\frac{\sqrt{\|\mathbf{c}_f - \mathbf{c}_i\|^2 - 4\rho_{min}^2}}{\|\mathbf{c}_f - \mathbf{c}_i\|} \right), \quad (3.23)$$

Após determinar completamente os arcos inicial e final que compõem o caminho, a reta que os une pode ser calculada por meio da interpolação linear dos pontos tangentes a mesma.

A Figura 3.7 apresenta o resultado final do cálculo do caminho de *Dubins* para um caso particular de *waypoints* \mathbf{P}_i e \mathbf{P}_f e raio ρ_{min} . O caminho ótimo identificado entre essas duas poses é do tipo *RLR*, ou seja, uma curva para a direita, seguido de um caminho em linha reta e posteriormente, outra curva para a direita.

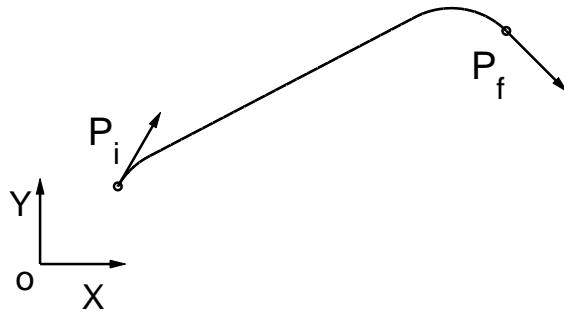


Figura 3.7: *Dubins' Path*, resultado final.

Já no caso de se optar pela geração de um caminho do tipo curto (*CCC*), existem dois resultados possíveis: *LRL* e *RLR*. Como não existe um método para se obter diretamente a melhor opção, como no caso do caminho longo, torna-se necessário o cálculo desses dois

tipos de caminho para a avaliação posterior de qual deles é o mais curto. O impacto do ponto de vista da complexidade computacional envolvida nesse caso é menor, devido ao fato de existirem apenas duas possibilidades de escolha. Porém, conforme já enfatizado, o poder de processamento dos sistemas computacionais aumenta cada vez mais, tornando menos significativa essa característica.

Com relação à determinação dos ângulos dos arcos inicial e final, ψ_a e ψ_d , o procedimento para o cálculo do DP , nesse caso, assemelha-se muito ao caso anterior. Como esses dois arcos são interpolados por meio de um terceiro semicírculo, é necessário especificá-lo inicialmente por meio do cálculo do seu centro. As Equações 3.24 e 3.25 fornecem o valor de \mathbf{c}_m para esses dois casos.

- Caso *LRL*:

$$\mathbf{c}_m = \mathbf{c}_i + 2\rho_{min} \begin{bmatrix} \cos \xi & -\sin \xi \\ -\sin \xi & \cos \xi \end{bmatrix} \frac{(\mathbf{c}_f - \mathbf{c}_i)}{\|\mathbf{c}_f - \mathbf{c}_i\|}. \quad (3.24)$$

- Caso *RLR*:

$$\mathbf{c}_m = \mathbf{c}_i + 2\rho_{min} \begin{bmatrix} \cos -\xi & -\sin -\xi \\ -\sin -\xi & \cos -\xi \end{bmatrix} \frac{(\mathbf{c}_f - \mathbf{c}_i)}{\|\mathbf{c}_f - \mathbf{c}_i\|}. \quad (3.25)$$

O ângulo ξ é fornecido pela Equação 3.26, onde o valor absoluto calculado dentro do arccosseno deve ser inferior a 1. Por fim, os valores de ψ_b e ψ_c podem ser determinados, ainda para os dois casos, seguindo as Equações 3.27 e 3.28, respectivamente.

$$\xi = \frac{\pi}{2} - \frac{1}{2} \cos^{-1} \left(\frac{\|\mathbf{c}_f - \mathbf{c}_i\|^2 - 8\rho_{min}^2}{-8\rho_{min}^2} \right), \quad (3.26)$$

$$\psi_b = \text{atan2} \left(\rho_{min} \frac{\mathbf{c}_m - \mathbf{c}_i}{\|\mathbf{c}_m - \mathbf{c}_i\|} \right), \quad (3.27)$$

$$\psi_c = \text{atan2} \left(\rho_{min} \frac{\mathbf{c}_f - \mathbf{c}_m}{\|\mathbf{c}_f - \mathbf{c}_m\|} \right). \quad (3.28)$$

Finalmente, o caminho mais curto é obtido pela comparação do somatório do comprimento das circunferências de cada um dos três semicírculos em questão. A dimensão de cada semicírculo é calculada por $C = \delta\psi \rho_{min}$, onde $\delta\psi$ é a varredura angular de cada arco.

Essa é uma forma bastante simples de se calcular o caminho de *Dubins*. Existem basicamente duas maneiras de realizar esse cálculo: através da Geometria Euclidiana, conforme visto aqui; e por meio da Geometria Diferencial, que utiliza princípios do cálculo diferencial para gerar o resultado final. Mais detalhes sobre a geração do caminho de *Dubins* utilizando Geometria Diferencial podem ser encontrados em (Shanmugavel et al., 2005). Esse mesmo trabalho também trata da utilização das curvas de Bézier para gerar trajetórias de curvatura contínua, chamadas de Hodográfico Pitagoreano.

3.1.3 Hodográfico Pitagoreano

Uma técnica mais recentemente estudada para a geração de trajetórias de veículos é o Hodográfico Pitagoreano, cujo primeiro estudo foi apresentado no trabalho (Farouki e Sakkalis, 1990). Por definição, o HP é uma curva no plano, cujas derivadas temporais (*hodographs*) satisfazem a condição de *Pitágoras*. Em outras palavras, supondo que as componentes do vetor velocidade $\dot{\vec{r}}(t) = [\dot{x}(t), \dot{y}(t)]$ do ponto (x, y) correspondam aos catetos de um triângulo retângulo, então conforme mostrado na Equação 3.29, existe uma função polinomial $h(t)$ que descreve o comportamento hodográfico da curva $\vec{r}(t)$, tal que

$$h(t)^2 = \dot{x}(t)^2 + \dot{y}(t)^2. \quad (3.29)$$

Essa característica apresenta duas vantagens principais do HP. A primeira é que $h(t)$ pode ser integrada analiticamente para o cálculo do comprimento da trajetória, ao passo que na Equação 3.5, $s(t)$ é obtido por meio de integração numérica, gerando um valor aproximado do real. A segunda é que os pontos gerados a partir da curva, devido à discretização do parâmetro t , apresentam uma distribuição constante, ou seja, para uma dada variação no tempo, a dimensão da curva varia de maneira proporcionalmente linear. Em seguida são apresentadas outras propriedades do HP relativas à sua utilização na geração de trajetórias bidimensionais.

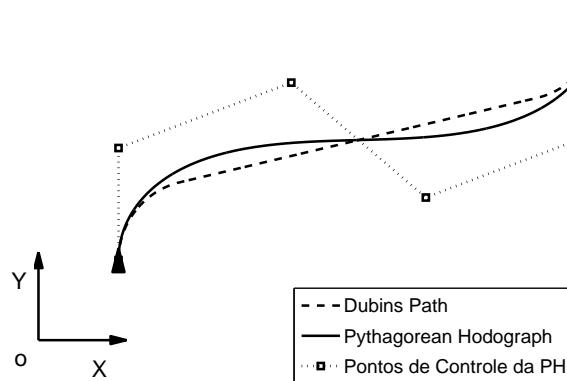
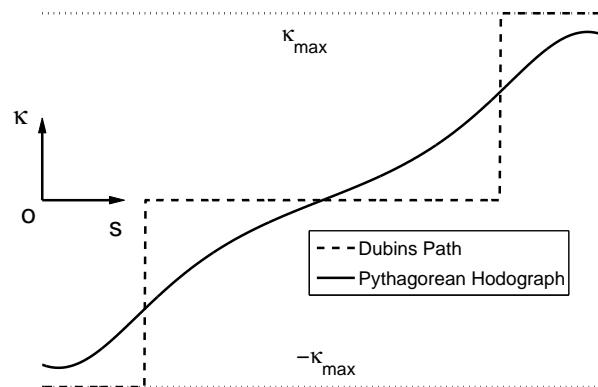
3.1.3.1 Propriedades do HP

Em (Shanmugavel et al., 2007), os autores apresentam uma técnica que permite gerar trajetórias para um veículo autônomo a partir do cálculo do HP. Tal abordagem apresenta duas características fundamentais em relação ao caminho de *Dubins*, uma referente ao comprimento da trajetória gerada e a outra relativa ao problema da descontinuidade da função de curvatura de $\vec{r}(t)$.

No que tange à questão do comprimento da trajetória gerada, a técnica do HP produz resultados que, na maior parte das vezes, se aproximam da curva ótima discutida anteriormente. De fato, essa é uma questão muito importante devido às restrições de autonomia de energia de alguns sistemas, e precisa ser levada em consideração na etapa de planejamento. Na Figura 3.8 é possível observar um resultado comparativo entre as duas técnicas, geradas a partir de uma mesma configuração de poses \mathbf{P}_i e \mathbf{P}_f . Nesse caso simples, a diferença no tamanho dos caminhos gerados foi inferior a 5%, valor esse que tende a diminuir em função do aumento da distância entre *waypoints*.

Entretanto, existem casos em que o comprimento do HP é muito maior do que o *DP*. Esse problema se caracteriza especialmente em casos onde os *waypoints* estão muito próximos, como no caso do caminho curto de *Dubins*.

Já com relação à questão da curvatura, o HP apresenta uma vantagem significativa em relação à técnica de *Dubins*. Enquanto essa última possui uma função de curvatura descontínua, a primeira é suave, conforme pode ser observado na Figura 3.9. Ambas as curvas geradas são limitadas pelo valor máximo de curvatura possível para o veículo (κ_{max}).

Figura 3.8: Comparação entre o HP e o *DP*.Figura 3.9: Comparação entre as funções de curvatura do HP e o *DP*.

Uma das conclusões levantada pelos autores em (Shanmugavel et al., 2007) é que existe uma solução de compromisso entre o comprimento de uma curva e o perfil de sua função de curvatura. O *DP* é de fato o caminho ótimo entre dois pontos, em parte por negligenciar a continuidade em $\kappa(s)$, a ponto de torná-la irrealizável por um robô real com restrições cinemáticas e dinâmicas. O HP, por outro lado, é calculado de forma iterativa, aumentando a trajetória gerada a fim de atingir os limites de curvatura desejados, porém mantendo-a contínua.

Outra vantagem dessa técnica é que os resultados produzidos geram valores máximos de curvatura que são inferiores ao valor máximo permitido, determinado por κ_{max} . É possível assumir que existe uma região ao redor da trajetória (um tubo) que configura uma área livre para a navegação do robô, que sofre com erros de posicionamento devido a imperfeições dos sensores ou devido a imposição de perturbações no sistema de controle (como rajadas de vento lateral). Tal vantagem, porém, não está vinculada exclusivamente à essa técnica.

Entretanto essa técnica apresenta outras características, não tão benéficas, que precisam ser levadas em conta. Uma delas é que a trajetória gerada é calculada de maneira iterativa. Alguns parâmetros do algoritmo são ajustados de forma recursiva, após várias iterações, de modo a se alcançar os limites de curvatura desejados. Isso pode tornar o algoritmo de planejamento mais caro computacionalmente do que a técnica de *Dubins*.

Outra desvantagem importante (e que também se aplica ao *DP*) diz respeito à composição de trajetórias determinadas por múltiplos *waypoints* no plano. Supondo uma trajetória c_1 gerada através das poses \mathbf{P}_a e \mathbf{P}_b , e uma outra c_2 , gerada entre as poses \mathbf{P}_b e \mathbf{P}_c , a junção de c_1 e c_2 provavelmente irá gerar uma trajetória final cuja função de curvatura é descontínua no ponto de transição dessas curvas. Como não há controle, na técnica do HP, dos valores de curvatura nos pontos extremos da curva $\vec{r}(t)$, a curvatura final da trajetória será descontínua.

Ao final desse capítulo, será proposta uma metodologia que visa gerar trajetórias de curvatura contínua, seguindo os princípios do HP, mas que permita a junção de diversas curvas, sem os problemas citados anteriormente. O cálculo do comprimento da trajetória gerada é obtido através de uma função analítica bem determinada (tal qual acontece com o HP), e o valor gerado é limitado a valores próximos da curva ótima. Assim como o HP, essa metodologia utiliza as curvas de Bézier para o planejamento de trajetórias.

3.1.3.2 Curvas de Bézier

Uma função cuja representação matemática respeita o princípio do HP é a curva de Bézier (Barnhill e Riesenfeld, 1974). Essa curva representa uma função polinomial paramétrica, e foi publicada pela primeira vez em 1962 por *Pierre Bézier*, sendo atualmente bastante utilizada na área de Computação Gráfica. A curva de Bézier é geralmente representada pela Equação 3.30, onde n determina a ordem da função, \mathbf{p}_i corresponde ao i -ésimo ponto de controle da Bézier, e $B(t)$ é o polinômio que controla o tipo da curva utilizada, também chamado de Polinômio de *Bernstein*.

$$\vec{r}(t) = \sum_{i=0}^n \mathbf{p}_i B_i^n(t), \quad (3.30)$$

onde

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i, \quad 0 \leq t \leq 1. \quad (3.31)$$

A técnica apresentada em (Shanmugavel et al., 2007), utiliza uma curva de Bézier para gerar uma trajetória de curvatura contínua entre dois *waypoints*, \mathbf{P}_i e \mathbf{P}_f . Nesse método, um polinômio de 5^a ordem é escolhido justamente por apresentar a ordem mínima necessária para a qual, a Equação 3.30 possua pelo menos um ponto de inflexão. Essa curva é também conhecida como *Quintic Pythagorean Hodograph*.

Curvas de mais alta ordem são encontradas na literatura, em diversas aplicações. Entretanto, para a geração de caminhos, o aumento de n torna cada vez mais complexo o problema de ajuste dos pontos de controle da curva de Bézier, especialmente devido à restrição de mínimo raio de curvatura. Para calcular a trajetória gerada pelo HP, o método determina a localização dos pontos de controle da curva de Bézier no plano *XY*.

Uma importante propriedade das curvas de Bézier é que os pontos de controle inicial e final (\mathbf{p}_0 e \mathbf{p}_5 respectivamente) sempre determinam as extremidades da função. Logo, a interpolação da curva sempre passa por esses pontos, os quais são determinados pelos

parâmetros de posição dos *waypoints* \mathbf{P}_i e \mathbf{P}_f .

Outra característica muito importante é que o valor da tangente à curva na extremidade inicial é sempre determinada em função do vetor $\overrightarrow{\mathbf{p}_0\mathbf{p}_1}$, onde \mathbf{p}_1 representa o segundo ponto de controle da curva de Bézier. De maneira equivalente, o vetor $\overrightarrow{\mathbf{p}_{n-1}\mathbf{p}_n}$ determina o valor da tangente à curva no ponto final, onde \mathbf{p}_{n-1} corresponde ao penúltimo ponto de controle. No caso de um polinômio de quinta ordem, a tangente final é dada pelo vetor $\overrightarrow{\mathbf{p}_4\mathbf{p}_5}$.

Essas duas propriedades permitem criar uma curva que passa pelos pontos determinados por \mathbf{P}_i e \mathbf{P}_f , com as respectivas orientações desejadas. Utiliza-se a Equação 3.32 para se determinar quatro dos seis pontos de controle necessários para a determinação do HP.

$$\begin{aligned} \mathbf{p}_0 &= [x_i, y_i], \\ \mathbf{p}_1 &= \mathbf{p}_0 + \frac{c_0}{5}[\cos \psi_i, \sin \psi_i], \\ \mathbf{p}_4 &= \mathbf{p}_5 - \frac{c_5}{5}[\cos \psi_f, \sin \psi_f], \\ \mathbf{p}_5 &= [x_f, y_f], \end{aligned} \tag{3.32}$$

onde novamente x_i, y_i e ψ_i correspondem aos parâmetros da pose inicial, e x_f, y_f e ψ_f são os parâmetros da pose final da trajetória. Já $c_0 \in [1, \infty]$ e $c_5 \in [1, \infty]$ são variáveis cujos valores são incrementados durante a iteração do algoritmo, visando atingir a restrição de curvatura desejada.

O principal problema dessa metodologia está em determinar os dois pontos de controle restantes da curva, \mathbf{p}_2 e \mathbf{p}_3 . Para isso, utiliza-se a solução proposta em (Farouki e Neff, 1995), onde tais pontos são calculados a partir dos valores determinados na Equação 3.32, e em função do polinômio $h(t)$. Existem, entretanto, quatro possíveis soluções para o problema, gerando com isso quatro curvas diferentes. A melhor trajetória é escolhida a partir do resultado que minimiza a função de energia de curvatura (Equação 3.33) definida em (Farouki, 1996).

$$\varepsilon = \int_{t_i}^{t_f} \kappa(t)^2 h(t) \delta t. \tag{3.33}$$

Após a determinação de todos os pontos de controle da curva, um algoritmo iterativo descrito em (Shanmugavel et al., 2006), calcula-se os valores de c_0 e c_5 para os quais a restrição de curvatura máxima é satisfeita. A principal desvantagem desse processo é que a convergência desses valores pode demandar um número muito grande ou mesmo infinito de iterações, dependendo da configuração das poses escolhidas (*waypoints* muito próximos, por exemplo). Além disso, os valores de curvatura nos pontos extremos da curva estão associados aos três pontos mais próximos de cada extremidade. Logo, não é possível controlar a função de curvatura de cada trajetória exatamente sobre os *waypoints*, gerando junções de caminhos que são descontínuos em $\kappa(t)$.

Na próxima seção, discute-se mais a fundo a questão da geração de trajetórias tridimensionais no espaço. Conforme será discutido, a metodologia de *Dubins* apresenta extensões para

esse problema que são meras aproximações para o caminho tridimensional ótimo. Já a técnica do HP é facilmente extensível para o caso 3D, porém os problemas levantados anteriormente ainda permanecem.

3.2 Planejamento de Trajetórias Tridimensionais

O estudo da geração de trajetórias para veículos terrestres pode ser encontrado em um grande volume de trabalhos na literatura. Um grande número de soluções utilizando diferentes abordagens são discutidas, algumas das quais foram apresentadas no capítulo anterior. De fato, tanto a geração de trajetória quanto outros problemas relativos ao controle de navegação, localização e mapeamento (questões fundamentais da robótica móvel) são mais simples de serem estudos no caso bidimensional, exatamente por apresentarem um número menor de restrições a serem consideradas.

Porém, ao se levar em conta o movimento no espaço tridimensional, novas questões passam a ser incorporadas ao problema, em virtude do aumento do número de graus de liberdade do veículo. Sabe-se que três variáveis (duas posições e uma orientação) são suficientes para se descrever completamente a configuração de um corpo rígido no plano. Já para um veículo aéreo no espaço são necessárias três posições e três ângulos de orientação, totalizando seis graus de liberdade.

No que se segue, apresenta-se a formalização do problema de geração de trajetória para o caso tridimensional. Uma proposta de extensão do caminho de *Dubins* é descrita, visando construir caminho que respeitem as restrições aqui consideradas para o robô aéreo. Ainda, uma generalização das curvas de *Bézier* para o espaço tridimensional permite a geração de trajetórias para veículos aéreos com as mesmas vantagens e desvantagens apresentadas pela técnica do HP 2D. Todas essas novas considerações serão abordadas na próxima seção, com o objetivo de fundamentar a proposição de um novo algoritmo para a geração de trajetórias ao final deste capítulo.

3.2.1 O Problema no Espaço Tridimensional

Generalizando a idéia anterior, dadas duas poses (ou vetores de configuração) \mathbf{P}_i e \mathbf{P}_f em \mathbb{R}^n onde n corresponde à dimensão do espaço cartesiano, gerar uma trajetória significa determinar uma curva $\vec{r}(t)$ que leve o veículo da configuração inicial até a configuração final, levando em consideração suas restrições dinâmicas e cinemáticas nas n dimensões.

Além das descrições em termos de posicionamento (x, y) no plano do chão, uma nova variável passa a ser considerada para o caso tridimensional, a altitude (z), de modo que $\vec{r}(t) = [x(t), y(t), z(t)]$. Além disso, duas outras variáveis descrevem especificações para a orientação do veículo no espaço (ψ, θ). O ângulo ψ determina a orientação de um *waypoint* paralelamente ao plano terrestre, como no caso bidimensional. Já o ângulo θ descreve a inclinação normal do *waypoint* também em relação ao plano terrestre. Ambos os ângulos são descritos em relação ao eixo **X** de um referencial fixo. A descrição formal do problema, nesse caso, passa a ser apresentada pela Expressão 3.34.

$$\begin{aligned}\mathbf{P}_i(x_i, y_i, z_i, \psi_i, \theta_i) &= \vec{r}(t_i) \\ \mathbf{P}_f(x_f, y_f, z_f, \psi_f, \theta_f) &= \vec{r}(t_f)\end{aligned}\quad (3.34)$$

A Figura 3.10 apresenta graficamente o problema, dadas as especificações de dois *waypoints* \mathbf{P}_i e \mathbf{P}_f , para o caso tridimensional. O veículo aéreo inicia o movimento na posição $[x_i, y_i, z_i]$ em um dado instante de tempo t_i , orientado segundo os ângulos ψ_i e θ_i . Espera-se que em um outro instante de tempo t_f , o mesmo alcance a posição $[x_f, y_f, z_f]$, com orientações espaciais ψ_f e θ_f . Pode-se ver que as retas formadas a partir dessas duas configurações não são coplanares, o que impede a utilização da técnica de *Dubins* diretamente.

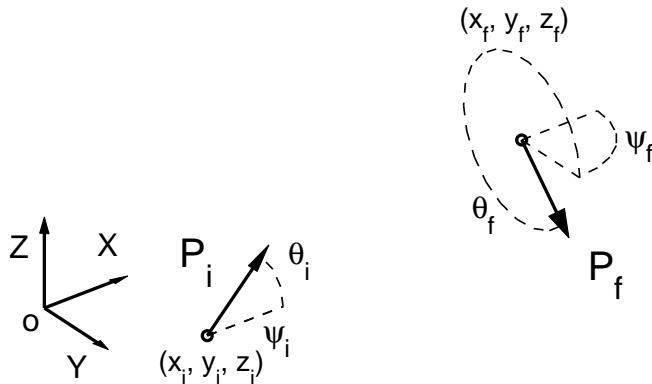


Figura 3.10: Configuração dos *waypoints* inicial e final no espaço tridimensional.

Como no caso bidimensional, a curva $\vec{r}(t)$ produzida deve respeitar a restrição de curvatura máxima κ_{max} definida pelo tipo de veículo aéreo utilizado. Porém nesse caso, o raio de curvatura varia com as três dimensões do espaço. Para calcular a curvatura de uma curva $\vec{r}(t)$ para o caso n -dimensional, utiliza-se a Equação 3.35, a qual corresponde a uma generalização da Equação 3.4. Ainda como no caso bidimensional, é desejável que $\vec{r}(t)$ apresente uma função de curvatura contínua, gerando uma trajetória tridimensional contínua para a aeronave.

$$\kappa(t) = \frac{|\dot{\vec{r}}(t) \times \ddot{\vec{r}}(t)|}{|\dot{\vec{r}}(t)|^3}. \quad (3.35)$$

Além da restrição de curvatura máxima, duas novas restrições são incorporadas ao problema da geração de trajetórias no espaço tridimensional. São elas a torção máxima da curva no espaço (τ_{max}) e o ângulo máximo de subida (θ_{max}).

A torção $\tau(t)$, também chamada de segunda curvatura, é definida como a taxa de oscilação (variação) do plano de curvatura de uma curva no espaço (Kreyszig, 1991). Em outras palavras, é uma medida da rotação da curva em torno do vetor tangente a mesma. No campo da Geometria Diferencial, as características de curvatura e torção são suficientes para especificar completamente uma curva no espaço tridimensional (no plano apenas a curvatura é suficiente, já que a torção de um plano é nula).

Para melhor caracterizar esses conceitos, define-se um referencial no espaço composto por três vetores (\mathbf{T} , \mathbf{N} e \mathbf{B}) mutuamente ortogonais e cuja origem situa-se em um dado ponto

pertencente à curva $\vec{r}(t)$. O vetor \mathbf{T} corresponde ao vetor tangente à curva $\vec{r}(t)$ em cada instante de tempo, que por sua vez é paralelo ao vetor velocidade do veículo. O vetor normal \mathbf{N} descreve a direção do centro de curvatura no mesmo instante de tempo e é ortogonal a \mathbf{T} . Completando o referencial, o vetor \mathbf{B} , chamado de binormal, é orientado na direção ortogonal ao plano de movimentação, sendo calculado como $\mathbf{T} \times \mathbf{N}$. Esse referencial é chamado de Referencial *Frenet-Serret*, e é ilustrado na Figura 3.11.

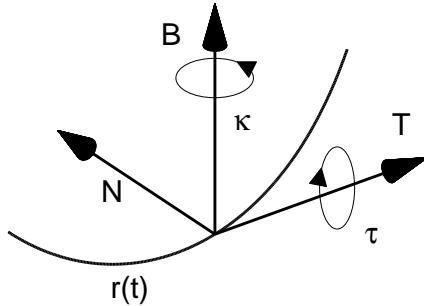


Figura 3.11: Referencial *Frenet-Serret*.

Conforme se pode observar, a curvatura é definida como a velocidade angular em torno do vetor binormal, ao passo que a torção representa a velocidade angular em torno do vetor tangente. Por convenção, curvatura e torção são positivas seguindo a regra da mão direita. O *Frenet-Serret* representa um referencial móvel e variante no tempo em função de $\vec{r}(t)$, conforme visto na Figura 3.12.

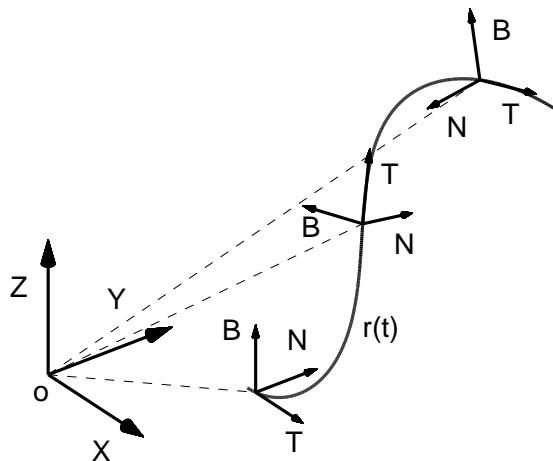


Figura 3.12: Variação temporal do referencial *Frenet-Serret*.

Logo, $\kappa(t)$ é contínua se as direções de \mathbf{T} e \mathbf{N} variarem continuamente ao longo da curva. No caso do *DP*, sabemos que a tangente da curva é sempre contínua, porém o centro de curvatura varia bruscamente devido a junção das três curvas do conjunto de *Dubins*. Ainda nesse caso, a curva é confinada a um único plano no espaço, e consequentemente a função de torção é nula.

A torção provoca variações na direção do vetor \mathbf{B} , torcendo a curva $\vec{r}(t)$ no espaço. Cada veículo possui um valor máximo de torção suportada para a navegação no espaço, a qual deve

ser considerada no cálculo de $\vec{r}(t)$. Tal valor pode ser determinado ainda por meio da Equação 3.36, onde σ_{min} caracteriza o raio mínimo de torção da aeronave.

$$\tau_{max} = \frac{1}{\sigma_{min}}. \quad (3.36)$$

Além disso, assim como no caso da curvatura, variações bruscas de $\tau(t)$ podem provocar efeitos indesejáveis sobre a movimentação de um veículo real, e por isso devem ser evitadas. Assim sendo, é desejável que a função de torção de $\vec{r}(t)$ seja contínua ao longo do tempo. Para calcular essa função, utiliza-se a Equação 3.37, generalizada para o espaço \mathbb{R}^n .

$$\tau(t) = \frac{\dot{\vec{r}}(t) \cdot [\ddot{\vec{r}}(t) \times \dddot{\vec{r}}(t)]}{|\dot{\vec{r}}(t) \times \ddot{\vec{r}}(t)|^2}. \quad (3.37)$$

O ângulo máximo de subida (θ_{max}) por sua vez, representa outra restrição importante para trajetórias de veículos aéreos. Ele descreve o valor máximo de inclinação da trajetória (subida ou descida) que pode ser executada pelo veículo aéreo durante o vôo (Equação 3.38). No caso das aeronaves que realizam pouso e decolagem vertical (*VTOL*) como helicópteros, quadrirotores, balões e alguns dirigíveis, essa restrição é menos importante, já que os limites para θ_{max} nesses casos são mais abrangentes. Entretanto, para as demais aeronaves, esse é um fator que deve ser levado em consideração, sob pena de tornar a trajetória irrealizável para o veículo.

$$\theta(t) = \text{atan2}\left(\frac{\delta z}{\delta t}\right) = \text{atan2}\left(\|\dot{\vec{r}}(t)\| \frac{\delta z}{\delta s}\right). \quad (3.38)$$

Valores que excedem o ângulo máximo de subida podem causar efeitos indesejáveis, como por exemplo, a perda de sustentação nas asas da aeronave, também conhecido como estol no jargão aeronáutico. Não necessariamente os limites de subida e descida são equivalentes, entretanto, isso será assumido aqui para fins de simplicidade. Assim sendo, uma curva $\vec{r}(t)$ é realizável se não apresenta nenhum trecho cujo ângulo de inclinação é superior a θ_{max} .

Dessa forma, o conjunto de restrições cinemáticas que devem ser impostas a trajetória $\vec{r}(t)$ são descritas conforme é visto na Equação 3.39. Tais características devem ser satisfeitas para qualquer tipo de veículo aéreo, realizando qualquer trajetória $\vec{r}(t)$.

$$|\kappa(t)| \leq \kappa_{max} \quad \& \quad |\tau(t)| \leq \tau_{max} \quad \& \quad |\theta(t)| \leq \theta_{max}. \quad (3.39)$$

3.2.2 Dubins' Path 3D

A literatura apresenta alguns poucos trabalhos sobre a geração de caminhos tridimensionais como extensões do *DP* 2D. Como exemplo, podemos citar os estudos de (Hwangbo et al., 2007), (Shanmugavel, 2007) e (Ambrosino et al., 2006). A primeira alternativa limita os ângulos de inclinação da curva (θ) a zero graus nos *waypoints*. Tal restrição ajuda a simplificar o problema, porém limita muito o número de aplicações. A segunda alternativa oferece solução apenas para o caso longo do caminho de *Dubins*, deixando o caso curto de lado. A terceira

por sua vez gera um caminho onde os movimentos longitudinais (normais ao plano da Terra) e látero-direcionais (paralelos ao plano da Terra) não ocorrem simultaneamente, gerando bons resultados apenas em casos muito particulares.

Todos esses trabalhos propõem aproximações para o caminho tridimensional, sem, entretanto garantir a otimalidade do resultado final. Além disso, nenhum deles considera a restrição de ângulo máximo de subida da aeronave de maneira direta, o que pode comprometer o resultado final.

A seguir será apresentada uma proposta de extensão do *DP* 2D para o espaço tridimensional. O caminho gerado visa unir as poses \mathbf{P}_i e \mathbf{P}_f , considerando todas as restrições descritas acima, porém deixando de lado o problema da continuidade das funções de curvatura e torção. A principal razão que motiva o cálculo do *Dubins' Path* Tridimensional (*DP* 3D) neste trabalho é a sua posterior utilização para a geração de trajetórias de curvatura e torção contínuas, conforme apresentado na próxima seção. Além disso, sabendo das limitações existentes no caminho de *Dubins*, tais resultados podem ser utilizados como base de comparação com as demais técnicas existentes.

3.2.2.1 Principais Considerações

Em primeiro lugar, é importante observar que a metodologia proposta aqui para a geração do *DP* 3D não garante uma trajetória ótima entre dois *waypoints* quaisquer, assim como acontece nos demais trabalhos citados. Até o momento, não foram encontradas publicações que apresentem o caminho tridimensional ótimo, a partir das restrições assumidas neste trabalho.

Para resolver esse problema de maneira simples, será considerada a existência de dois planos. O primeiro será chamado de plano látero-direcional, onde ocorrerão as variações de posição nos eixos **X** e **Y** do espaço. O segundo será o plano longitudinal, sobre o qual se darão as variações no eixo **Z** em função dos dois primeiros. O problema da construção do caminho tridimensional será reduzido ao cálculo de duas curvas bidimensionais sobre esses dois planos. Um algoritmo iterativo será utilizado para garantir que a curva respeite as restrições de curvatura, torção e ângulo de subida especificadas.

3.2.2.2 Geração do Caminho 3D

Considerando-se a Figura 3.13 é possível observar a decomposição do espaço tridimensional em dois planos. O plano látero-direcional (equivalente ao plano da Terra) representa exatamente o espaço de configurações para o planejamento bidimensional estudado anteriormente. A linha pontilhada desenhada sobre esse plano representa o *DP* 2D (\mathcal{D}_{lat}), calculado entre os *waypoints* $\mathbf{P}_{lat|i} = [x_i, y_i, \psi_i]$ e $\mathbf{P}_{lat|f} = [x_f, y_f, \psi_f]$, que por sua vez foram derivados dos *waypoints* tridimensionais \mathbf{P}_i e \mathbf{P}_f .

Já o plano longitudinal (normal ao plano da Terra) representa um espaço bidimensional composto pelos eixos **Z** e **S**, onde o primeiro corresponde a altitude no espaço 3D, e o segundo representa a função de comprimento $s(t)$ de \mathcal{D}_{lat} , dada pela Equação 3.5. Sobre esse plano será calculado o caminho \mathcal{D}_{lon} , correspondente a variação de altitude ao longo do caminho látero-

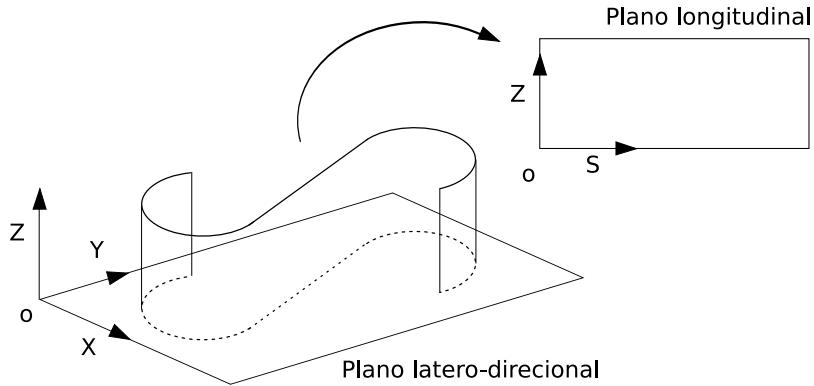


Figura 3.13: Representação dos planos utilizado no cálculo do *Dubins' Path* 3D.

direcional. Para isso serão considerados os *waypoints* $\mathbf{P}_{lon|i} = [0, z_i, \theta_i]$ e $\mathbf{P}_{lon|f} = [s_f, z_f, \theta_f]$, também derivados das poses tridimensionais, onde s_f constitui o comprimento total de \mathcal{D}_{lat} .

Ainda com relação a geração do caminho látero-direcional, será utilizado um raio de curvatura equivalente a $k_\rho \rho_{min}$, onde $k_\rho \in [1, \infty]$ representa um fator multiplicativo para o raio mínimo da curva 2D. Inicialmente unitário, esse fator é incrementado de maneira iterativa, aumentando assim o comprimento de \mathcal{D}_{lat} a cada interação do algoritmo. Esse é um ponto fundamental na metodologia utilizada, conforme discutido mais adiante. Essa etapa pode ser representada de maneira simples pela Expressão 3.40.

$$\mathcal{D}_{lat} = \text{dubinspath}(\mathbf{P}_{lat|i}, \mathbf{P}_{lat|f}, k_\rho \rho_{min}). \quad (3.40)$$

O passo seguinte é o cálculo da variação de altitude ao longo do caminho látero-direcional gerado anteriormente. Inicialmente é preciso verificar se o comprimento do caminho gerado s_f é suficiente para comportar a variação de altitude desejada ($z_f - z_i$). Em outras palavras, supondo que para uma dada variação de altitude, o veículo tenha que percorrer uma determinada distância no plano XY (devido as suas características não holonômicas), é necessário lançar mão de um teste que verifique se s_f é ou não suficiente. Primeiramente, calcula-se a distância d_z conforme apresentado na Equação 3.41. Para o caso de veículos que apresentam ρ_{min} igual à zero (helicópteros e outras aeronaves de pouso e decolagem vertical), a variação de altitude é sempre possível para qualquer s_f .

$$d_z = \frac{\sqrt{s_f^2 + (z_f - z_i)^2}}{\rho_{min}}. \quad (3.41)$$

Em seguida, calcula-se o valor mínimo aceitável para o comprimento s_f da curva em função da variação de altitude requerida, por meio da Equação 3.42.

$$d_{z_min} = \sqrt{4 - (|\cos \theta_i| + |\cos \theta_f|)^2} + |\sin \theta_i| + |\sin \theta_f|, \quad (3.42)$$

onde

$$\theta_i' = \theta_i - \text{atan}2\left(\frac{z_f - z_i}{s_f}\right), \quad (3.43)$$

e

$$\theta_f' = \theta_f - \text{atan}2\left(\frac{z_f - z_i}{s_f}\right). \quad (3.44)$$

Os ângulos θ_i' e θ_f' são obtidos, respectivamente, por meio das Equações 3.43 e 3.44, que por sua vez dependem dos ângulos de inclinação estabelecidos para \mathbf{P}_i e \mathbf{P}_f . Vale lembrar que os valores absolutos desses ângulos devem ser inferiores à θ_{max} para que o caminho seja realizável.

O teste é concluído com a utilização da Equação 3.45 para determinar se é possível gerar o caminho tridimensional. Caso a condição não seja satisfeita, o caminho \mathcal{D}_{lat} gerado será considerado insuficiente para promover a variação de altitude desejada, e o algoritmo deve partir novamente do início, com uma valor maior de k_ρ . Isso acontece iterativamente até que a condição seja satisfeita.

$$d_z \begin{cases} > d_{z_min} & \text{variação de altitude aceitável ,} \\ \leq d_{z_min} & \text{variação de altitude não-aceitável .} \end{cases} \quad (3.45)$$

Quando for encontrado um caminho bidimensional longo o suficiente para comportar a variação de altitude, o passo seguinte é traçar a variação de z ao longo da dimensão da curva. De maneira análoga ao caso 2D, existirá um certo conjunto de curvas no plano longitudinal que irão compor a variação de altitude ao longo do eixo **S**, que varia linearmente em função da imposição de velocidade constante considerada anteriormente. É possível representar esse conjunto como sendo $\mathcal{D}_{lon} = [USU, DSD, USD, DSU]$, onde D correspondem à arcos de raio ρ_{min} para baixo (*Down*), S são variações lineares, e U são arcos de mesmo raio, porém orientados para cima (*Up*). A Figura 3.14 representa um dos possíveis perfis de variação da altitude ao longo do caminho tridimensional.

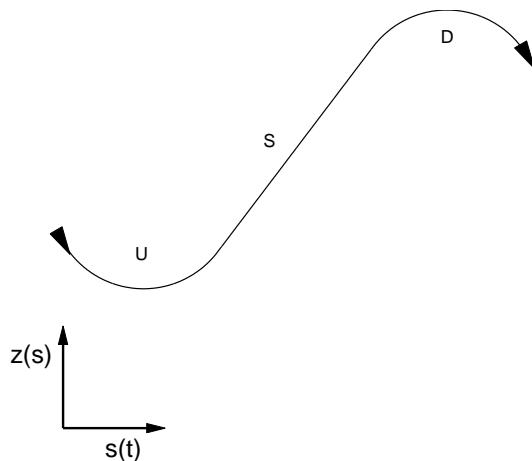


Figura 3.14: Exemplo da variação de altitude no plano longitudinal.

Para estimar o centro de curvatura da variação de z nos pontos de saída e chegada de cada *waypoint*, utilizam-se as Equações 3.46 e 3.47, onde γ assume valor $-\frac{\pi}{2}$ se θ é negativo, e $\frac{\pi}{2}$ caso contrário.

$$\mathbf{c}_{zi} = \begin{bmatrix} 0 \\ z_i \end{bmatrix} + \rho_{min} \begin{bmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix}, \quad (3.46)$$

$$\mathbf{c}_{zf} = \begin{bmatrix} s_f \\ z_f \end{bmatrix} + \rho_{min} \begin{bmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} \cos \theta_f \\ \sin \theta_f \end{bmatrix}. \quad (3.47)$$

De maneira equivalente ao caso bidimensional, é possível calcular os ângulos internos dos arcos inicial e final do \mathcal{D}_{lon} . As Equações 3.48 e 3.49 fornecem, respectivamente, os valores iniciais e finais em relação ao centro das curvas calculados anteriormente.

$$\theta_a = \text{atan2}\left(-\frac{z_i - \mathbf{c}_{zi}(z)}{\mathbf{c}_{zi}(s)}\right), \quad (3.48)$$

$$\theta_d = \text{atan2}\left(\frac{z_f - \mathbf{c}_{zf}(z)}{s_f - \mathbf{c}_{zf}(s)}\right). \quad (3.49)$$

Já os demais ângulos que formam a varredura dos arcos inicial e final (θ_b e θ_c , respectivamente) são calculados em função do tipo de caminho escolhido:

- Caso *USU*:

$$\theta_b = \text{atan2}\left(-\frac{\rho_{min}}{\|\mathbf{c}_{zf} - \mathbf{c}_{zi}\|} \frac{\delta c(s)}{\delta c(z)}\right), \quad (3.50)$$

$$\theta_c = \theta_b. \quad (3.51)$$

- Caso *DSD*:

$$\theta_b = \text{atan2}\left(-\frac{\rho_{min}}{\|\mathbf{c}_{zf} - \mathbf{c}_{zi}\|} \frac{\delta c(s)}{\delta c(z)}\right), \quad (3.52)$$

$$\theta_c = \theta_b. \quad (3.53)$$

- Caso *USD*

$$\theta_b = \text{atan2}\left(\frac{\rho_{min}}{\|\mathbf{c}_{zf} - \mathbf{c}_{zi}\|} \frac{\delta c(s) \cos(-\xi_z) - \delta c(z) \sin(-\xi_z)}{\delta c(s) \sin(-\xi_z) - \delta c(z) \cos(-\xi_z)}\right), \quad (3.54)$$

$$\theta_c = \text{atan2}\left(\frac{\rho_{min}}{\|\mathbf{c}_{zf} - \mathbf{c}_{zi}\|} \frac{\delta c(s) \cos(\pi - \xi_z) - \delta c(z) \sin(\pi - \xi_z)}{\delta c(s) \sin(\pi - \xi_z) - \delta c(z) \cos(\pi - \xi_z)}\right). \quad (3.55)$$

- Caso *DSU*

$$\theta_b = \text{atan}2\left(\frac{\rho_{min}}{\|\mathbf{c}_{zf} - \mathbf{c}_{zi}\|} \frac{\delta c(s) \cos(\xi_z) - \delta c(z) \sin(\xi_z)}{\delta c(s) \sin(\xi_z) - \delta c(z) \cos(\xi_z)}\right), \quad (3.56)$$

$$\theta_c = \text{atan}2\left(\frac{\rho_{min}}{\|\mathbf{c}_{zf} - \mathbf{c}_{zi}\|} \frac{\delta c(s) \cos(\xi_z - \pi) - \delta c(z) \sin(\xi_z - \pi)}{\delta c(s) \sin(\xi_z - \pi) - \delta c(z) \cos(\xi_z - \pi)}\right). \quad (3.57)$$

onde

$$\delta c(s) = \mathbf{c}_{zf}(s) - \mathbf{c}_{zi}(s)$$

$$\delta c(z) = \mathbf{c}_{zf}(z) - \mathbf{c}_{zi}(z)$$

e ξ_z pode ser obtido por meio da Equação 3.58.

$$\xi_z = \sin^{-1}\left(\frac{\sqrt{\|\mathbf{c}_{zf} - \mathbf{c}_{zi}\|^2 - 4(\rho_{min})^2}}{\|\mathbf{c}_{zf} - \mathbf{c}_{zi}\|}\right), \quad (3.58)$$

Por fim, a curva tridimensional é obtida a partir da composição do primeiro e do segundo caminho, conforme apresentado na Expressão 3.59, onde x , y e z são funções do comprimento $s(t)$ da curva no plano látero-direcional.

$$\vec{r}(t) = [\mathcal{D}_{lat}(x), \mathcal{D}_{lat}(y), \mathcal{D}_{lon}(z)]. \quad (3.59)$$

Dois últimos testes são ainda requeridos para a validação do resultado final. O primeiro referente a restrição de ângulo máximo de subida da curva, cujos pontos mais críticos ocorrem nas extremidades e na reta central do eixo **Z**. Como as inclinações dos pontos extremos (θ_i e θ_f) são limitados pelo valor máximo, basta verificar a inclinação da curva central (reta). Conseqüentemente o módulo dessa inclinação deve ser inferior ao valor de θ_{max} da aeronave para que o resultado seja válido.

O segundo teste é relativo ao máximo valor de torção da curva, que ocorre nos pontos extremos de $\vec{r}(t)$. Isso acontece devido ao fato de as funções de curvatura das curvas nos planos longitudinal e látero-direcional serem igualmente máximas nesses pontos. Os valores de torção nos extremos podem ser calculados por meio da Equação 3.37, e devem inferiores em módulo a τ_{max} .

Ambos os testes podem ser indiretamente satisfeitos a partir do aumento do valor de k_ρ , o que conseqüentemente implica no aumento do comprimento da curva final. Tanto a torção quanto a inclinação da curva (além da curvatura) tendem a diminuir a cada iteração do algoritmo. Existe, entretanto, um ponto de falha dessa técnica, que ocorre quando caminho gerado no plano látero-direcional é dado por apenas uma reta, e a variação de altitude não é factível para esse trecho. Nesse caso particular, o aumento do valor de k_ρ não causa qualquer efeito, causando uma situação de não convergência para o resultado final. Esse problema pode ser facilmente resolvido, criando-se *waypoints* extras entre os dois previamente especificados.

O resultado final da aplicação do DP 3D às poses \mathbf{P}_i e \mathbf{P}_f da Figura 3.10 pode ser visto na Figura 3.15. É possível observar que a projeção da curva tridimensional gerada é equivalente

à curva bidimensional calculada no plano XY , o que acontece devido ao primeiro caminho ter servido de base para o cálculo do caminho final.

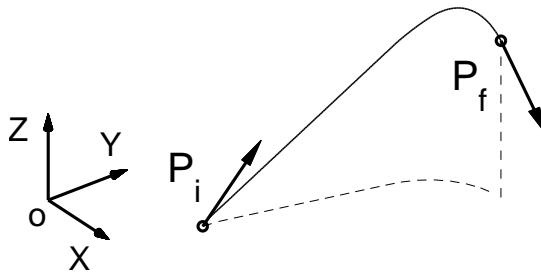


Figura 3.15: Configuração arbitrária dos *waypoints* inicial e final, para caso 3D.

O comprimento total do caminho 3D gerado é dado por meio da Equação 3.60.

$$s(t) = \int_{t_i}^{t_f} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} dt. \quad (3.60)$$

O caminho gerado por meio dessa metodologia não garante a otimalidade do resultado. A projeção do caminho em XY pode inclusive não corresponder ao melhor caminho bidimensional, caso mais de uma iteração seja necessária para a convergência do algoritmo. Embora respeite as restrições de curvatura, torção e ângulo de subida especificados, uma desvantagem que permanece é a descontinuidade nas funções, tanto de $\kappa(t)$ quanto de $\tau(t)$.

Na seqüência, apresenta-se a extensão do HP para o caso tridimensional, gerando assim o Hodográfico Pitagoreano Tridimensional (HP 3D). O estudo dessa técnica, associada ao caminho 3D de *Dubins*, servirá de auxílio para o desenvolvimento de uma nova técnica para a geração de trajetórias com funções contínuas de curvatura e torção, apresentado na última seção deste capítulo.

3.2.3 Hodográfico Pitagoreano Tridimensional

A técnica do HP, apresentada em (Shanmugavel et al., 2007), foi discutida anteriormente para o caso bidimensional. Em seu trabalho, o autor demonstra que é possível gerar uma trajetória de curvatura contínua entre dois *waypoints* (P_i e P_f) utilizando-se curvas de Bézier de quinta ordem. Essas curvas são configuradas de modo a respeitar o princípio de *Pitágoras* (Equação 3.29), pelas razões explicadas anteriormente. Ainda nesse trabalho, é possível ver também a demonstração da técnica estendida para o espaço de movimentação tridimensional. Nesse caso, o problema passa a apresentar um grau de liberdade a mais (o eixo **Z**), e uma nova condição, dada pela Equação 3.61, passa a ser considerada.

$$h(t)^2 = \dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2. \quad (3.61)$$

De forma equivalente ao caso bidimensional, deve-se definir seis pontos de controle da curva de Bézier, representados por p_i , com $i = 0 \dots 5$. Quatro desses pontos podem ser determinados a partir dos parâmetros que compõem as poses P_i e P_f , conforme se pode observar na Equação

3.62.

$$\mathbf{p}_0 = [x_i, y_i, z_i],$$

$$\begin{aligned} \mathbf{p}_1 &= \mathbf{p}_0 + \frac{c_0}{5}[\cos \psi_i \cos \theta_i, \sin \psi_i \cos \theta_i, \sin \theta_i], \\ \mathbf{p}_4 &= \mathbf{p}_5 - \frac{c_5}{5}[\cos \psi_f \cos \theta_f, \sin \psi_f \cos \theta_f, \sin \theta_f], \end{aligned} \quad (3.62)$$

$$\mathbf{p}_5 = [x_f, y_f, z_f],$$

onde novamente $x's$, $y's$, $z's$, $\psi's$ e $\theta's$ são os parâmetros das poses inicial e final, e $c_0 \in [1, \infty]$ e $c_5 \in [1, \infty]$ são variáveis de iteração, utilizadas para gerar uma curva dentro dos padrões de curvatura e torção determinados.

Novamente o problema de determinação do HP 3D se resume ao cálculo dos pontos de controle \mathbf{p}_2 e \mathbf{p}_3 . Utilizando-se a metodologia apresentada em (Farouki e Han, 2006), bastante semelhante ao caso 2D, temos novamente um algoritmo iterativo para o cálculo da curva que melhor se ajuste às restrições de curvatura e torção impostas para o veículo aéreo. A Figura 3.16 apresenta um resultado comparativo entre a técnica apresentada anteriormente para o cálculo do *DP* 3D, e o HP 3D.

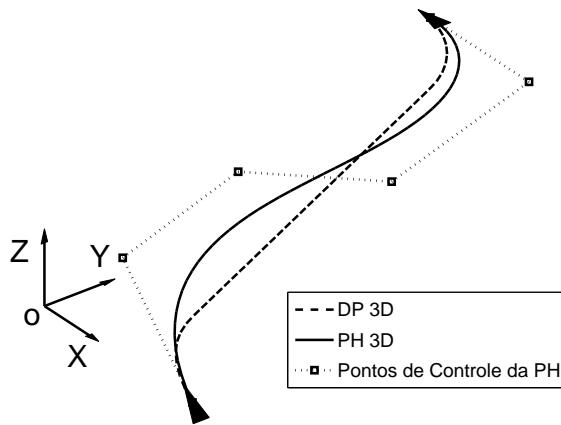


Figura 3.16: Comparaçāo entre o HP 3D e o *Dubins' Path* 3D.

Os mesmos problemas referentes à utilização do HP 2D podem ser identificados em sua extensão para o espaço tridimensional, como as descontinuidades na união de duas ou mais curvas devido a consideração de múltiplos *waypoints*, maior complexidade computacional em relação a técnica de *Dubins* e não convergência do algoritmo para determinadas poses. Além disso, um problema mais sério é que a metodologia não leva em conta a restrição de ângulo máximo de inclinação da aeronave, o que não garante que a trajetória seja realizável. Veículos aéreos com restrições κ_{max} e τ_{max} equivalentes podem apresentar diferentes valores de θ_{max} , necessitando de trajetórias diferentes.

3.3 Planejamento Utilizando Curvas de Bézier de 7^a Ordem

Nas seções anteriores foram considerados dois tipos de algoritmos de planejamento de movimentos para veículos não-holonômicos. O primeiro foi o *DP*, que produz caminhos de comprimento ótimo (no caso 2D) entre dois *waypoints* quaisquer, mas que problemas de descontinuidade em sua função de curvatura. Já o segundo foi o *HP*, uma técnica de geração de trajetória para robôs baseadas em curvas de curvatura contínua. Ambas as metodologias foram analisadas tanto para o caso bidimensional, quanto tridimensional.

Porém, apesar das vantagens apresentadas pelo *HP*, existem algumas desvantagens que devem ser levadas em consideração. Portanto, será apresentada nesta seção, uma metodologia que visa resolver todos os problemas citados anteriormente. Além disso, os resultados gerados por essa técnica apresentam valores de comprimento, em geral, próximos do caminho ótimo.

3.3.1 Caso Bidimensional

Uma das principais características do *DP* é que o caminho total é composto pela união de três curvas (semicírculos e retas). Conforme visto anteriormente, existem dois tipos de composições possíveis: a composição do tipo *CCC*, referente ao caminho do tipo curto; e a do tipo *CLC*, correspondente ao caminho do tipo longo.

Analizando esse aspecto do *DP*, é possível generalizar essa idéia para promover a construção de uma trajetória pela união de três curvas quaisquer (que não simplesmente arcos e retas), de modo que a função de curvatura final apresentada seja contínua. Em outras palavras, é possível unir três funções para gerar uma trajetória contínua entre dois *waypoints*. Essas funções devem, individualmente, respeitar as restrições consideradas para o problema. Assim sendo, tal trajetória pode ser representada por meio do conjunto \mathcal{B} , composto por três funções $[B_1, B_2, B_3]$, onde cada função B_j apresenta uma função contínua de curvatura, e valores equivalentes para $\kappa(t)$ em suas duas extremidades.

Uma vantagem imediata proporcionada pela utilização desse conjunto é a continuidade em $\kappa(t)$, não apenas de uma trajetória $\vec{r}(t)$ específica entre dois *waypoints*, mas também para a junção de diversas trajetórias que passam por diversos *waypoints* no plano. Além disso, espera-se que a curva total resultante apresente um comprimento relativamente próximo ao caminho ótimo, algo que é considerado fundamental nesse problema. Espera-se ainda que a determinação das funções do conjunto \mathcal{B} possa ser realizada em tempo computacional finito.

Um tipo de função que pode ser utilizada com essa finalidade é a curva de Bézier. Conforme visto anteriormente, a metodologia do *HP* apresenta funções contínuas de curvatura ao longo de toda a trajetória, porém as curvas geradas não exibem valores semelhantes para $\kappa(t)$ em seus pontos extremos $\vec{r}(t)|_{t=0}$ e $\vec{r}(t)|_{t=1}$.

Uma propriedade bastante interessante das curvas de Bézier é que o vetor \mathbf{T} tangente a curva $\vec{r}(t)$ no ponto inicial $r(0)$ é sempre paralelo ao vetor $\overrightarrow{\mathbf{p}_0\mathbf{p}_1}$, formado pelos dois pontos de controle iniciais. O mesmo é observado para o ponto final da curva $r(1)$, cuja tangente é paralela ao vetor $\overrightarrow{\mathbf{p}_{n-1}\mathbf{p}_n}$, formado pelos dois pontos de controle finais. Essa característica é útil, pois serve para orientar a curva segundo as especificações de orientação ψ de cada

waypoint na etapa de planejamento.

Outra propriedade que também pode ser bastante útil refere-se ao valor da curvatura nos pontos extremos de cada curva. Observando a Equação 3.4 é possível ver que a curvatura de uma função depende das derivadas parciais de primeira e segunda ordens da mesma. Conclui-se então que para a curva de Bézier, as curvaturas inicial e final dependem da configuração dos três pontos de controle mais extremos dessa curva ($[\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2]$ e $[\mathbf{p}_{n-2}, \mathbf{p}_{n-1}, \mathbf{p}_n]$, respectivamente). Matematicamente, a curvatura inicial de uma curva de Bézier pode ser determinada por meio da Equação 3.63 (Sederberg, 2007).

$$\kappa(t)|_{t=0} = \frac{(n-1)}{n} \frac{\|\mathbf{p}_2 - \mathbf{p}_1\|}{\|\mathbf{p}_1 - \mathbf{p}_0\|} \sin \sigma, \quad (3.63)$$

onde σ representa o ângulo formado entre os vetores $\overrightarrow{\mathbf{p}_0\mathbf{p}_1}$ e $\overrightarrow{\mathbf{p}_1\mathbf{p}_2}$. Se esse ângulo for proporcional à π por qualquer valor inteiro, maior ou igual a zero, então $\kappa(0)$ será nulo. Se σ é nulo, então os vetores em questão são colineares com sentidos equivalentes. O mesmo raciocínio se estende para a curvatura $\kappa(1)$, no outro extremo da curva.

Logo, é possível gerar uma curva de Bézier com valores inicial e final de curvatura nulos, por meio do alinhamento dos três pontos de controle mais próximos de cada extremidade da mesma. Uma vez que tais curvas apresentam funções contínuas em $\kappa(t)$ com valores nulos nas extremidades, a união de duas ou mais curvas sempre produzirá trajetórias com funções de curvatura contínuas.

Tal suposição é válida apenas para polinômios cuja ordem é igual ou superior à cinco. Entretanto, a utilização de uma função de quinta ordem (como é o caso da *Quintic Pythagorean Hodograph*) limita em muito o projeto de curvas, já que os seis pontos de controle ficam restritos à condição de colinearidade. Por outro lado, a complexidade exigida no projeto de curvas de mais alta ordem fica reduzida, devido à mesma condição, de modo que se pode utilizar uma técnica bastante semelhante ao cálculo do HP para uma curva de Bézier de 7ª ordem, conforme apresentado na seqüência.

Para garantir que o conjunto \mathcal{B} não constitua uma trajetória muito maior do que o caminho tridimensional de *Dubins*, tomemos esse como base para a construção das três curvas de Bézier. O primeiro passo é: dadas as poses inicial e final, calcular o DP , conforme apresentado na Expressão 3.64.

$$\mathcal{D} = \text{dubinspath}(\mathbf{P}_i, \mathbf{P}_f, k_s \rho_{min}), \quad (3.64)$$

onde $k_s > 1$ constitui um ganho que é acrescido de forma iterativa pelo algoritmo, com a finalidade de restringir a curva à κ_{max} .

Como já é sabido, \mathcal{D} é composto por três curvas, onde a primeira e a terceira representam arcos de raio $k_s \rho_{min}$. Utilizando a informação dessas duas curvas, será realizada na seqüência, a decomposição dos *waypoints* inicial e final em quatro outros, descritos como

$$\begin{aligned}
\mathbf{P}_1 &= \mathbf{P}_i, \\
\mathbf{P}_2 &= \left(\mathbf{c}_i(x) + k_s \rho_{min} \cos \psi_b, \mathbf{c}_i(y) + k_s \rho_{min} \sin \psi_b, \psi_b \pm \frac{\pi}{2} \right), \\
\mathbf{P}_3 &= \left(\mathbf{c}_f(x) + k_s \rho_{min} \cos \psi_c, \mathbf{c}_f(y) + k_s \rho_{min} \sin \psi_c, \psi_c \pm \frac{\pi}{2} \right), \\
\mathbf{P}_4 &= \mathbf{P}_f,
\end{aligned}$$

onde $[\mathbf{c}_i(x), \mathbf{c}_i(y)]$ e $[\mathbf{c}_f(x), \mathbf{c}_f(y)]$ representam as coordenadas do centro da primeira e da terceira curva de \mathcal{D} , respectivamente (Equações 3.11 e 3.12), e ψ_b e ψ_c são os ângulos internos observados na Figura 3.6. A redundância na determinação dos ângulos de cada pose pode ser resolvida em função do sentido da curvatura de cada curva, sendo que pela convenção adotada, curvas para a esquerda correspondem a sinais positivos.

A partir desse ponto é possível definir \mathcal{B} como um conjunto de três curvas de Bézier de 7^a ordem, onde a primeira curva define uma trajetória de \mathbf{P}_1 para \mathbf{P}_2 , a segunda de \mathbf{P}_2 para \mathbf{P}_3 e a última de \mathbf{P}_3 para \mathbf{P}_4 . Assim sendo, dada uma das curvas do conjunto, unindo os respectivos *waypoints* de partida $\mathbf{P}_a = [x_a, y_a, \psi_a]$ e de chegada $\mathbf{P}_b = [x_b, y_b, \psi_b]$ no plano, seis de seus oito pontos de controle podem ser determinados a partir da Equação 3.65.

$$\begin{aligned}
\mathbf{p}_0 &= [x_a, y_a], \\
\mathbf{p}_1 &= \mathbf{p}_0 + \frac{s_j}{2\pi} [\cos \psi_a, \sin \psi_a], \\
\mathbf{p}_2 &= \mathbf{p}_1 + \frac{s_j}{2\pi} [\cos \psi_a, \sin \psi_a], \\
\mathbf{p}_5 &= \mathbf{p}_6 - \frac{s_j}{2\pi} [\cos \psi_b, \sin \psi_b], \\
\mathbf{p}_6 &= \mathbf{p}_7 - \frac{s_j}{2\pi} [\cos \psi_b, \sin \psi_b], \\
\mathbf{p}_7 &= [x_b, y_b],
\end{aligned} \tag{3.65}$$

onde a variável s_j para $j \in [1, 2, 3]$ representa o comprimento de uma das três curvas de \mathcal{D} correspondente a curva de \mathcal{B} avaliada.

O problema é semelhante a geração do HP, restando dois pontos de controle a serem determinados (\mathbf{p}_3 e \mathbf{p}_4). Tais pontos podem ser calculados utilizando-se a Equação 3.66, a qual constitui uma adaptação da metodologia apresentada em (Farouki e Neff, 1995).

$$\begin{aligned}
\mathbf{p}_3 &= \mathbf{p}_2 + \frac{1}{5} [u_0 u_1 - v_0 v_1, u_0 v_1 + u_1 v_0], \\
\mathbf{p}_4 &= \mathbf{p}_3 + \frac{2}{15} [u_1^2 - v_1^2, 2u_1 v_1] + \frac{1}{15} [u_0 u_2 - v_0 v_2, u_0 v_2 + u_2 v_0].
\end{aligned} \tag{3.66}$$

Os parâmetros $[u_0, u_1, u_2]$ e $[v_0, v_1, v_2]$ representam os coeficientes dos polinômios $u(t)$ e $v(t)$ respectivamente, utilizados no projeto do HP de quinta ordem (Shanmugavel et al., 2007).

$$(u_0, v_0) = \sqrt{\frac{5}{2}} \left[\sqrt{\|\Delta \mathbf{p}_1\| + \Delta x_1}, \frac{\Delta y_1}{|\Delta y_1|} \sqrt{\|\Delta \mathbf{p}_1\| - \Delta x_1} \right]. \quad (3.67)$$

$$(u_2, v_2) = \pm \sqrt{\frac{5}{2}} \left[\sqrt{\|\Delta \mathbf{p}_5\| + \Delta x_5}, \frac{\Delta y_5}{|\Delta y_5|} \sqrt{\|\Delta \mathbf{p}_5\| - \Delta x_5} \right]. \quad (3.68)$$

$$(u_1, v_1) = -\frac{3}{4} [u_0 + u_2, v_0 + v_2] \pm \sqrt{\frac{1}{2}} \left[\sqrt{c+a}, \frac{b}{|b|} \sqrt{c-a} \right], \quad (3.69)$$

onde

$$\begin{aligned} \Delta x_k &= x_{k+1} - x_k, \\ \Delta y_k &= y_{k+1} - y_k, \end{aligned}$$

e

$$\Delta \mathbf{P}_k = [\Delta x_k, \Delta y_k].$$

Já a , b e c podem ser determinados conforme a seguir:

$$a = \frac{9}{16}(u_0^2 - v_0^2 + u_2^2 - v_2^2) + \frac{5}{8}(u_0 u_2 - v_0 v_2) + \frac{15}{2}(x_5 - x_2),$$

$$b = \frac{9}{8}(u_0 v_0 + u_2 v_2) + \frac{5}{8}(u_0 v_2 + v_0 u_2) + \frac{15}{2}(y_5 - y_2),$$

$$c = \sqrt{a^2 + b^2}.$$

Assim como acontece no cálculo do HP, existem quatro possíveis soluções para a trajetória desejada, isso devido a redundâncias existentes nas Equações 3.68 e 3.69. Utiliza-se a Equação 3.33 para determinar a solução que apresenta o perfil da função de curvatura mais suave dentre todas possíveis.

Após o cálculo das três curvas do conjunto \mathcal{B} , resta verificar se a função de curvatura da trajetória final não excede o valor permitido κ_{max} . Uma vez que $\kappa(t)$ tenha sido calculado no passo anterior, a verificação é bastante simples. Caso essa condição não tenha sido alcançada, o valor de k_s deve ser incrementado, e uma nova iteração do algoritmo é requisitada. Isso se repete até que a curva respeite a restrição.

A Figura 3.17 apresenta uma comparação entre os resultados gerados por essa técnica e pelo *DP*, para um mesmo par de *waypoints*. A curva \mathcal{B} foi gerada sobre um caminho ótimo do tipo *CLC*, apresentando um comprimento cerca de 10% maior do que o mínimo possível. O ganho disso pode ser visto na Figura 3.18, onde a função de curvatura apresenta um resultado contínuo e limitado pelo raio mínimo especificado.

Outro resultado, para o caso do caminho curto de *Dubins*, pode ser observado na Figura

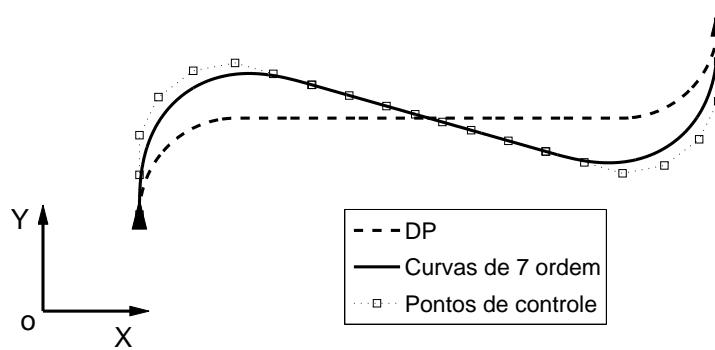
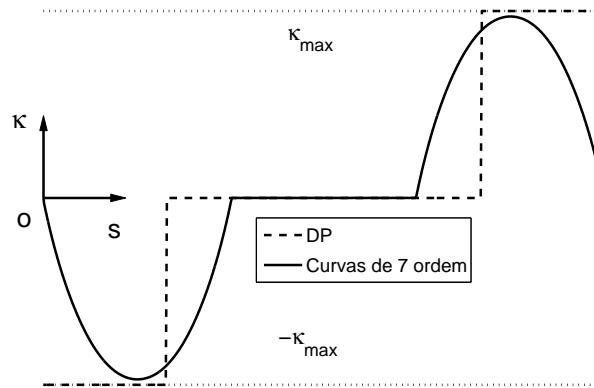
Figura 3.17: Resultado final baseado no caso do caminho longo de *Dubins*.

Figura 3.18: Perfil de curvatura de Figura 3.17.

3.19. O resultado é importante, pois permite a realização de manobras do robô em pequenas áreas e em curtas distâncias, ainda mantendo o perfil da função de curvatura contínua, conforme é mostrado na Figura 3.20.

Apesar do cálculo da trajetória nesse caso se dar de forma iterativa, como no caso do HP, a convergência do método tende a ser mais rápida (mesmo com o cálculo de três curvas), devido ao fato de \mathcal{B} ser baseada em uma curva de *Dubins*. Além disso, sempre é possível encontrar uma trajetória, já que sempre existe um conjunto \mathcal{D} para quaisquer dois *waypoints* especificados (Shkel e Lumelsky, 2001). Por fim, o comprimento da trajetória obtida é, em geral, próximo ao caminho ótimo, se comparado a alguns casos particulares do HP, já que se baseia em DP com raios de curvatura k_s vezes ρ_{min} , onde k_s é em geral inferior a 2.

3.3.2 Caso Tridimensional

De forma semelhante ao caso bidimensional, é possível gerar uma trajetória para uma aeronave no espaço tridimensional a partir do cálculo do DP 3D apresentado anteriormente. Entretanto, nesse caso não apenas a restrição de curvatura deve ser considerada, mas também a torção e o ângulo máximo de subida especificados.

$$\mathcal{D} = \text{dubinspath3D}(\mathbf{P}_i, \mathbf{P}_f, k_s \rho_{min}, \sigma_{min}, \theta_{max}), \quad (3.70)$$

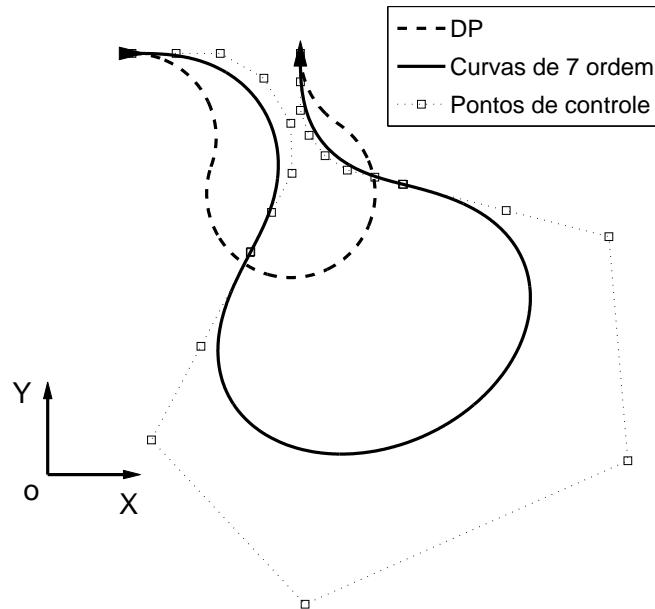


Figura 3.19: Resultado final baseado no caso do caminho curto de *Dubins*.

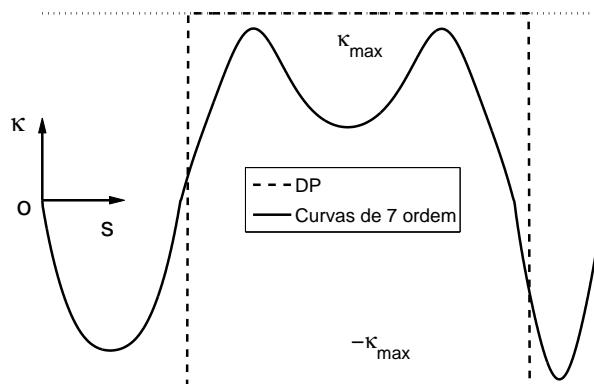


Figura 3.20: Perfis de curvatura da Figura 3.19.

Novamente, k_s representam um fator multiplicativo que será utilizado para que a trajetória possa respeitar simultaneamente as três restrições consideradas.

$$\mathbf{P}_1 = \mathbf{P}_i,$$

$$\mathbf{P}_2 = \left(x_2, y_2, z_2, \psi_b \pm \frac{\pi}{2}, \theta_b \pm \frac{\pi}{2} \right),$$

$$\mathbf{P}_3 = \left(x_3, y_3, z_3, \psi_c \pm \frac{\pi}{2}, \theta_c \pm \frac{\pi}{2} \right),$$

$$\mathbf{P}_4 = \mathbf{P}_f,$$

onde

$$\begin{aligned}x2 &= \mathbf{c}_i(x) + k_s \rho_{min} \cos \psi_b \cos \theta_b, \\y2 &= \mathbf{c}_i(y) + k_s \rho_{min} \sin \psi_b \cos \theta_b, \\z2 &= \mathbf{c}_i(z) + k_s \rho_{min} \sin \theta_b,\end{aligned}$$

e

$$\begin{aligned}x3 &= \mathbf{c}_f(x) + k_s \rho_{min} \cos \psi_c \cos \theta_c, \\y3 &= \mathbf{c}_f(y) + k_s \rho_{min} \sin \psi_c \cos \theta_c, \\z3 &= \mathbf{c}_f(z) + k_s \rho_{min} \sin \theta_c,\end{aligned}$$

com $\mathbf{c}_j(z) = \mathbf{c}_{zj}(z)$, e θ_b e θ_c calculados pelo *DP* 3D. Ainda, com relação à redundância dos ângulos, ψ' s para a esquerda e θ' s para cima determinam sinais positivos.

Como no caso bidimensional, a determinação dos oito pontos de controle de cada curva racional de Bézier do conjunto \mathcal{B} determinam a trajetória. Seis desses pontos podem ser determinados por meio da Equação 3.71.

$$\begin{aligned}\mathbf{p}_0 &= [x_a, y_a, z_a], \\ \mathbf{p}_1 &= \mathbf{p}_0 + \frac{s_j}{2\pi} [\cos \psi_a \cos \theta_a, \sin \psi_a \cos \theta_a, \sin \theta_a], \\ \mathbf{p}_2 &= \mathbf{p}_1 + \frac{s_j}{2\pi} [\cos \psi_a \cos \theta_a, \sin \psi_a \cos \theta_a, \sin \theta_a], \\ \mathbf{p}_5 &= \mathbf{p}_6 - \frac{s_j}{2\pi} [\cos \psi_b \cos \theta_b, \sin \psi_b \cos \theta_b, \sin \theta_b], \\ \mathbf{p}_6 &= \mathbf{p}_7 - \frac{s_j}{2\pi} [\cos \psi_b \cos \theta_b, \sin \psi_b \cos \theta_b, \sin \theta_b], \\ \mathbf{p}_7 &= [x_b, y_b, z_b].\end{aligned}\tag{3.71}$$

O ganho s_j representa o mesmo calculado no caso bidimensional, porém correspondendo à dimensão da curva no espaço 3D. De forma equivalente, os pontos \mathbf{p}_3 e \mathbf{p}_4 podem ser calculados a partir da extensão da metodologia vista em (Farouki e Han, 2006) para o caso tridimensional.

Como acontece no caso 2D, k_s é utilizado para aumentar o comprimento da trajetória gerada, de modo que o resultado alcance as restrições impostas para a curva 3D, assim como acontece no *DP* 3D. Um exemplo comparativo entre a técnica do *DP* 3D proposta anteriormente, e essa última pode ser vista na Figura 3.21. O caminho de *Dubins* gerado é do tipo *CCC*. As funções de curvatura, torção e inclinação pode ser conferidas nas Figuras 3.22, 3.23 e 3.24, respectivamente.

De forma equivalente, pode-se verificar um exemplo de trajetória baseada no caminho longo do *DP* 3D e suas principais características nas Figuras 3.25, 3.26, 3.27 e 3.28.

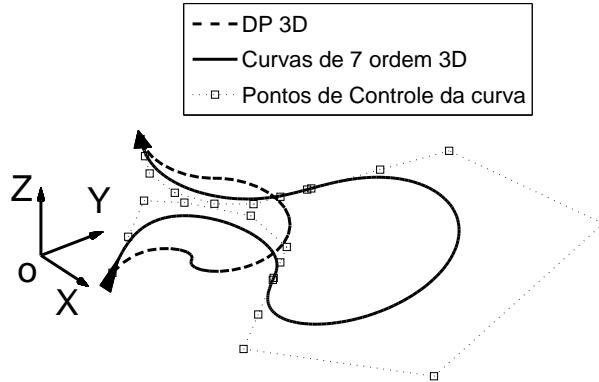


Figura 3.21: Resultado final baseado no caso do caminho curto do *DP 3D*.

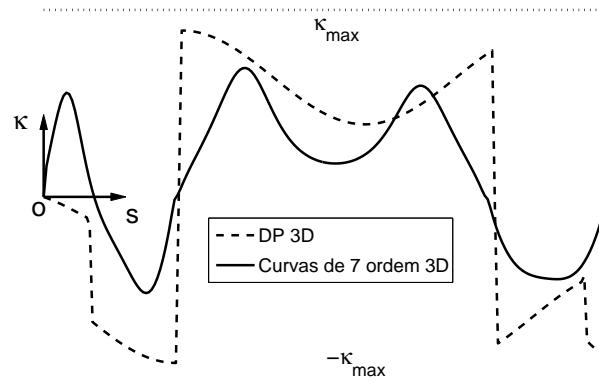


Figura 3.22: Perfis de curvatura da Figura 3.21.

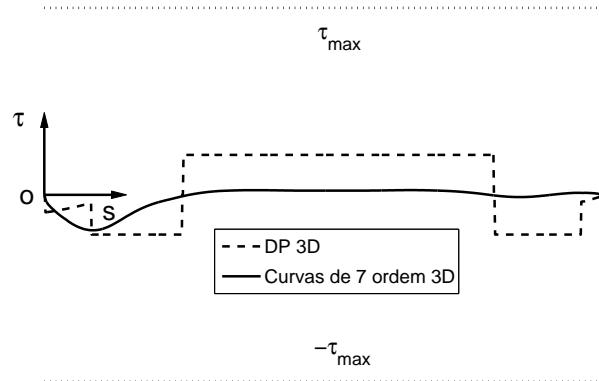


Figura 3.23: Perfis de torção da Figura 3.21.

Nesse capítulo foram apresentadas duas metodologias de planejamento de movimentos para veículos aéreos. A primeira representa uma adaptação do caminho de *Dubins* bidimensional para o espaço 3D. A segunda é uma técnica de geração de trajetórias a partir da utilização de curvas de Bézier de 7^a ordem. No próximo capítulo são apresentadas algumas relações entre as técnicas aqui propostas e as questões dinâmicas e cinemáticas dos veículos aéreos. No penúltimo capítulo são apresentadas as principais conclusões obtidas a partir da metodologia considerada.

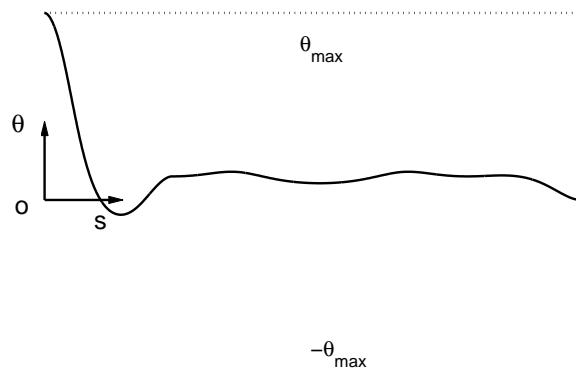


Figura 3.24: Perfil de inclinação da Figura 3.21 para as Curvas de Bézier de 7^a ordem.

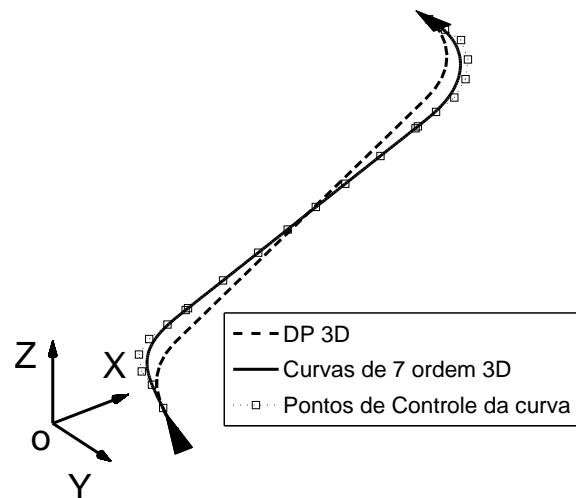


Figura 3.25: Resultado final baseado no caso do caminho longo do *DP 3D*.

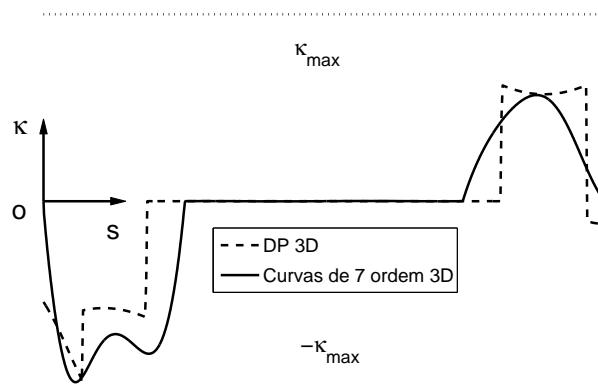


Figura 3.26: Perfis de curvatura da Figura 3.25.

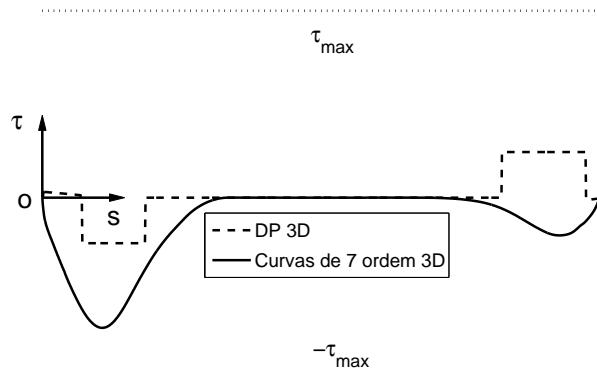


Figura 3.27: Perfis de torção da Figura 3.25.

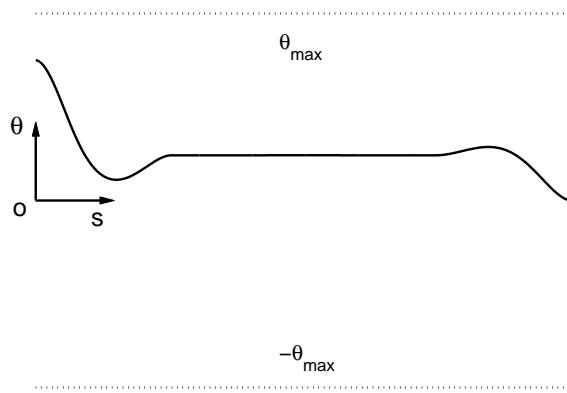


Figura 3.28: Perfil de inclinação da Figura 3.25 para as Curvas de Bézier de 7^a ordem.

Capítulo 4

Introdução a Dinâmica de Aeronaves

Neste capítulo é realizada uma breve descrição sobre a modelagem cinemática e dinâmica de um veículo aéreo, mais especificamente de uma aeronave de asa-fixa. O objetivo é estabelecer uma relação entre as restrições cinemáticas consideradas anteriormente e as características mais particulares de uma aeronave. Não se espera com isso cobrir todo o escopo relativo a modelagem de veículos, o que em si constitui um assunto muito vasto e complexo. Alguns dos conceitos aqui apresentados são fundamentais para o entendimento da parte experimental deste trabalho, descrita no próximo capítulo.

4.1 Restrições Cinemáticas e Características Dinâmicas

Os veículos aéreos constituem um dos maiores desafios do ponto de vista da Engenharia. Em primeiro lugar por se tratarem de sistemas extremamente complexos, geralmente modelados por equações matemáticas não-lineares e compostos por múltiplas entradas (sensores) e múltiplas saídas (atuadores). Em segundo lugar pela necessidade de se projetar sistemas que sejam seguros, já que uma aeronave representa um processo crítico, que lida com vidas humanas na maioria das vezes. Mesmo um veículo aéreo autônomo pode representar uma ameaça às pessoas, se mal projetado.

A metodologia apresentada no capítulo anterior descreve técnicas para a geração de trajetórias tridimensionais para veículos aéreos. Tais veículos são descritos de maneira genérica por meio de suas restrições cinemáticas, como os raios mínimos de curvatura e torção, e o máximo ângulo de subida. Entretanto, a maneira pela qual essas restrições se caracterizam depende, entre outros fatores, do comportamento dinâmico da aeronave específica.

Nesta seção serão apresentados alguns conceitos básicos sobre a modelagem de aeronaves. Conforme mencionado no capítulo anterior, aeronaves de asa-fixa apresentam restrições cinemáticas mais fortes do que as demais (helicópteros, dirigíveis, etc), e por isso, suas características serão avaliadas mais a fundo neste capítulo. Isso não significa, porém, que a dinâmica desses veículos seja mais complicada do que a dos outros tipos de aeronaves. De fato, é possível controlar uma aeronave de asa-fixa com um simples conjunto de controladores, como por exemplo o Controlador Proporcional, Integral e Derivativo (PID), conforme discutido no capí-

tulo seguinte. Estudos mais aprofundados sobre a modelagem e o controle de veículos aéreos podem ser encontrados em (Stevens e Lewis, 1992), (McLean, 1990) e (Valavanis, 2007).

4.1.1 Definições Básicas

Para simplificar o estudo da cinemática e dinâmica, as aeronaves são geralmente modeladas como corpos rígidos no espaço tridimensional. Para tal, é estabelecido um conjunto de referenciais em relação aos quais será realizada a modelagem do problema. É muito comum no estudo da dinâmica de aeronaves a consideração de um referencial fixo ao plano da Terra, chamado de Referencial *North, East and Down (NED)*. Esse é considerado um referencial inercial (o que de fato não é verdade), e é utilizado para estabelecer uma referência para o posicionamento da aeronave no espaço. A Figura 4.1 representa o referencial *NED* fixado sobre o plano terrestre, e cujo eixo **X** aponta para o Norte, o eixo **Y** para o Leste e **Z** para o centro da Terra.

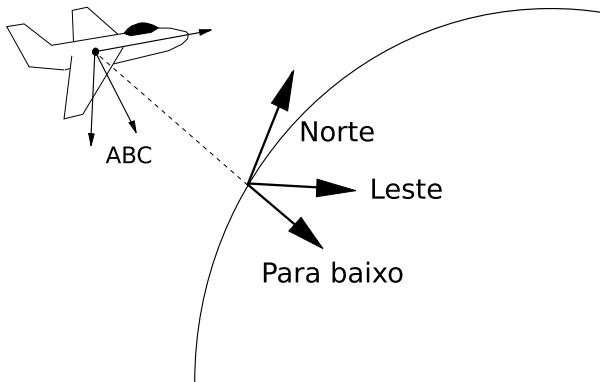


Figura 4.1: Referencial *NED* (Norte, Leste e Para baixo) e referencial do corpo da aeronave (*ABC*).

Assim, a posição da aeronave é representada pelo vetor $\vec{P}(t) = [x_N(t), y_E(t), z_D(t)]$ em relação ao referencial *NED*, representando três dos seis graus de liberdade do corpo no espaço. Os outros três representam as orientações espaciais do veículo, que nesse caso são modelados por meio de ângulos de *Euler*, descritos pelo vetor $\vec{\Phi}(t) = [\Phi(t), \Theta(t), \Psi(t)]$. Também definidos pela nomenclatura *roll*, *pitch* e *yaw*, esses ângulos descrevem a rotação da aeronave em relação ao referencial fixo na Terra.

Supõe-se nesse caso a existência de um referencial fixo ao corpo da aeronave, conhecido como Referencial do corpo da aeronave (*ABC*), conforme é visto também na Figura 4.1. Por convenção, o eixo **X** (eixo longitudinal) desse referencial é disposto de maneira longitudinal ao corpo da aeronave, conforme é apresentado na Figura 4.2. Já **Y** (eixo lateral) é orientado lateralmente, enquanto que **Z** (eixo normal) é apontado para baixo. Dessa forma, a seqüência de rotações para os ângulos de *Euler* é descrita por Ψ (rotação no eixo **Z** do corpo), seguido de Θ (rotação no eixo **Y** do corpo), e por fim Φ (rotação no eixo **X** do corpo).

Com base nesses princípios, são estabelecidos três tipos de momentos angulares para os veículos no espaço. O momento de rolamento (*roll*) representa as forças que causam as

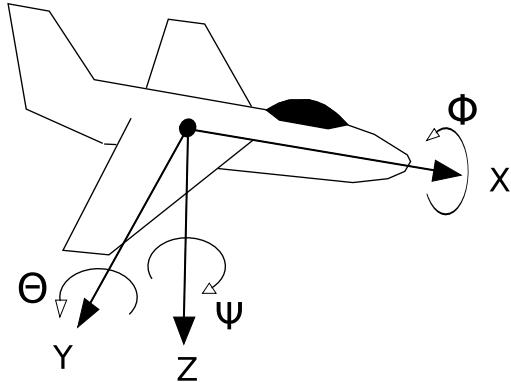


Figura 4.2: Orientação do referencial da aeronave em relação ao referencial *NED*.

variações angulares em Φ ao longo do eixo longitudinal do referencial *ABC*. O momento de arfagem (*pitch*), por sua vez, atua produzindo variações angulares em Θ ao longo do eixo lateral. Já o momento de guinada (*yaw*) caracteriza a variação ao longo do eixo normal por meio do ângulo Ψ . Os três momentos podem ser observados separadamente na Figura 4.3.

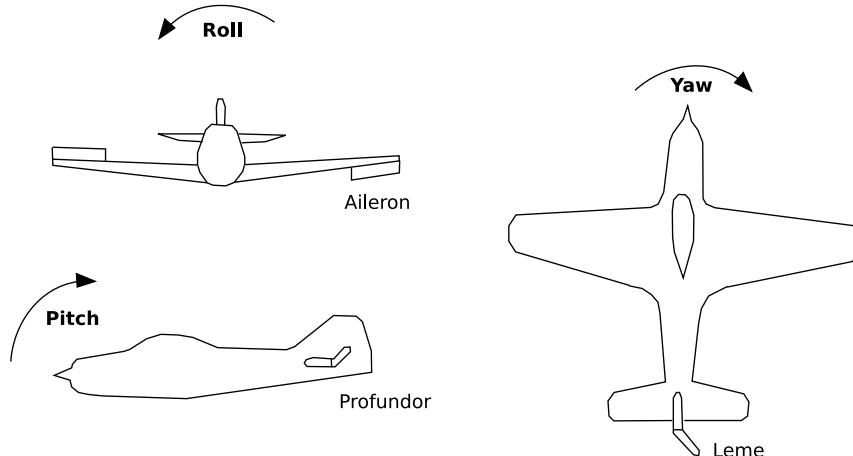


Figura 4.3: Momentos angulares que compõem a atitude de uma aeronave no espaço de configurações tridimensional.

Essas forças de rotação são, em geral, produzidas por meio da atuação em determinadas superfícies de controle da aeronave. Nessa mesma figura é possível observar a relação entre um conjunto simples de atuadores e os momentos provocados para o caso de uma aeronave de asa-fixa.

Denomina-se δ_a a variável que determina o comando de *aileron*, localizado nas asas da aeronave, e que causa grande parte do momento de rolamento. Já δ_e corresponde ao comando de profundor (*elevator*), responsável pelo momento de arfagem do veículo. A mesma relação é estabelecida entre δ_r , o comando de leme (*rudder*) e o momento de guinada da aeronave. Entretanto, por se tratar de um sistema com uma dinâmica extremamente complexa, essa divisão não é tão clara na prática. Por exemplo, o leme pode causar tanto o momento de guinada quanto o de rolamento, mesmo que em menor intensidade.

A variação angular da aeronave no espaço, causada por esses momentos, e medida em relação ao referencial *NED*, pode ser descrita por meio da Equação 4.1.

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} P + Q \tan \Theta \sin \Phi + R \tan \Theta \cos \Phi \\ Q \cos \Phi - R \sin \Phi \\ Q \frac{\sin \Phi}{\cos \Theta} + R \frac{\cos \Phi}{\cos \Theta} \end{bmatrix}, \quad (4.1)$$

sendo que

$$\vec{\omega}_{ABC} = \begin{bmatrix} P \\ Q \\ R \end{bmatrix}$$

representam as velocidades angulares relativas aos eixos **X**, **Y** e **Z** do referencial *ABC*, respectivamente. Essa é a equação cinemática que descreve o comportamento da atitude da aeronave em função do tempo. Já $\vec{\omega}_{ABC}$ representa um vetor cuja variação temporal é descrita por meio de equações dinâmicas de momento, baseadas nas Leis de *Newton*, e influenciadas diretamente pelos parâmetros específicos de cada aeronave, como superfícies de controle, matriz de inércia, posição do Centro de Gravidade (CG), entre outros (Stevens e Lewis, 1992).

Outra relação cinemática que descreve o comportamento do sistema é a Equação 4.2, onde a variação da posição do CG da aeronave pode ser descrita em função das velocidade medidas nos eixos do corpo. Por uma questão de simplicidade c foi utilizado para representar a função coseno e s para a função seno.

$$\begin{bmatrix} \dot{x}_N \\ \dot{y}_E \\ \dot{z}_D \end{bmatrix} = \begin{bmatrix} U(c\Theta c\Psi) + V(-c\Phi s\Psi + s\Phi s\Theta c\Psi) + W(s\Phi s\Psi + c\Phi s\Theta c\Psi) \\ U(c\Theta s\Psi) + V(c\Phi c\Psi + s\Phi s\Theta s\Psi) + W(-s\Phi c\Psi + c\Phi s\Theta s\Psi) \\ Us\Theta - Vs\Phi c\Theta - Wc\Phi c\Theta \end{bmatrix}, \quad (4.2)$$

sendo que

$$\vec{v}_{ABC} = \begin{bmatrix} U \\ V \\ W \end{bmatrix}$$

determinam as velocidades de translação relativas aos eixos **X**, **Y** e **Z** do referencial *ABC*, respectivamente. Essas velocidades também variam em função das características dinâmicas de cada veículo, como a massa, as forças de sustentação, arrasto e tração, entre outras.

Existe ainda um terceiro referencial considerado no caso da modelagem de aeronaves de asa-fixa, chamado de Referencial do Vento. Como no caso do *ABC*, esse referencial também possui sua origem fixada ao CG da aeronave. Entretanto seus eixos são definidos em função da direção de movimento da aeronave no espaço. Conforme é possível observar na Figura 4.4, o eixo **X** desse novo referencial coincide com a direção do vento relativo, porém com o sentido contrário. Essa descrição é válida, porém, apenas para o caso em que a atmosfera está parada,

ou seja, na ausência de vento propriamente dito.

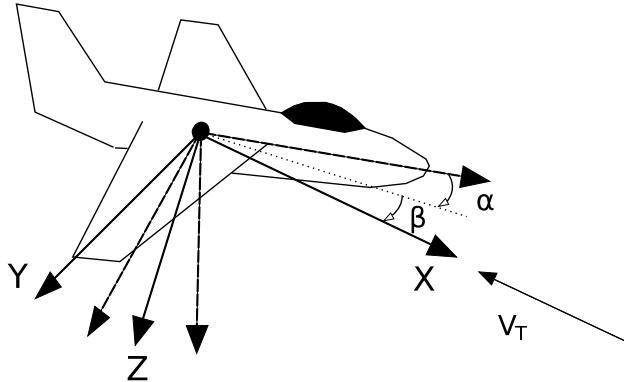


Figura 4.4: Orientação do referencial do vento.

A relação existente entre o referencial ABC e o referencial do vento constitui uma seqüência de duas rotações, determinadas pelos ângulos α e β . O ângulo α , conhecido como ângulo de ataque, descreve a orientação longitudinal do eixo do corpo da aeronave em relação a sua direção de movimento (contra o vento relativo). Esse parâmetro é de fundamental importância, pois dele depende a força de sustentação nas asas da aeronave. Já β é conhecido como ângulo de derrapagem, descrevendo a orientação lateral do referencial do corpo em relação ao vento. Esses dois ângulos são, em geral, próximos de zero para o caso das aeronaves de asa-fixa.

É possível ainda estabelecer uma relação de transformação entre a velocidade medida em relação ao eixo do vento e as velocidades lineares do eixo ABC , conforme apresentado na Equação 4.3.

$$\begin{bmatrix} V_T \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sqrt{U^2 + V^2 + W^2} \\ \text{atan}2\left(\frac{W}{U}\right) \\ \sin^{-1}\left(\frac{V}{V_T}\right) \end{bmatrix}. \quad (4.3)$$

Nesta seção foi apresentada uma visão bastante concisa sobre a modelagem de uma aeronave de asa-fixa. Na seqüência, os parâmetros descritos anteriormente serão utilizados para estabelecer uma relação entre as restrições cinemáticas de movimento e o comportamento temporal do modelo apresentado.

4.1.2 Cálculo das Restrições

Existem diversas maneiras de se estimar as restrições cinemáticas de um veículo aéreo para a aplicação da metodologia de geração de trajetórias descrita no capítulo anterior. Uma delas seria calcular os valores máximos de curvatura, torção e ângulo de subida a partir da medição dos sensores da aeronave durante o vôo. Os valores de posição e velocidade a princípio, seriam

suficientes para resolver as Equações 3.35, 3.37 e 3.38, a partir das quais seriam retirados os valores máximos. Essa técnica, porém, implica em expor a aeronave real a condições extremas de vôo, o que nem sempre seria possível devido aos riscos envolvidos. Entretanto, será o método utilizado na etapa experimental, uma vez que as aeronaves consideradas são modelos virtuais de veículos reais.

Outra alternativa seria projetar estratégias de controle capazes de limitar os valores de κ_{max} , τ_{max} e θ_{max} , tornando tais valores bem conhecidos. Isso pode ser feito por meio da aplicação de saturações à matemática de controle, o que em geral, restringe o comportamento dinâmico do sistema controlado. Essa restrição é, de certa forma, conveniente no caso de veículos aéreos, pois aumenta a estabilidade e simplifica a tarefa de controle no caso de sistemas multi-variáveis.

Uma terceira alternativa seria estimar essas restrições a partir do modelo matemático do veículo, (descrito na seção anterior para o caso de aeronaves de asa-fixa). Sem dúvida, essa não seria a melhor escolha, uma vez que a modelagem de um sistema como esse é uma tarefa bastante custosa. Entretanto, um modelo razoavelmente bom da aeronave pode servir a diversas finalidades (testes, projetos, estudos) que compensam esse investimento. De fato, a terceira alternativa pode vir a incluir a segunda, que por sua vez pode ainda incluir a primeira.

Nessa seção serão apresentados os passos iniciais para se obter as restrições cinemáticas a partir do modelo dinâmico de uma aeronave de asa-fixa. O raciocínio será descrito de maneira genérica, a partir dos parâmetros cinemáticos do modelo de corpo rígido de uma aeronave. A variação de tais parâmetros está diretamente ligada ao comportamento dinâmico de cada aeronave em particular, conforme visto a seguir.

Inicialmente, supõe-se a existência de um referencial de *Frenet-Serret*, cuja origem coincide com o CG da aeronave. Conforme discutido no capítulo anterior, esse referencial é composto por três vetores unitários (**T**, **N** e **B**) mutuamente ortogonais. O vetor **T** representa a direção tangente à curva em um determinado instante de tempo t , cuja direção é dada por $\dot{\vec{r}}(t)$. Fazendo uma analogia com a trajetória da aeronave, a direção de **T** é dada pelo vetor direção da aeronave no espaço, que coincide com o vetor velocidade do vento (\vec{v}_W) descrito em relação ao referencial *NED*. A relação é descrita por meio da Equação 4.4.

$$\dot{\vec{r}}(t) = \vec{v}_{NED} = B^T S \vec{v}_W, \quad (4.4)$$

onde

$$B = R_{NED2ABC} = \begin{bmatrix} c\Theta c\Psi & c\Theta s\Psi & -s\Theta \\ -c\Phi s\Psi + s\Phi s\Theta c\Psi & c\Phi c\Psi + s\Phi s\Theta s\Psi & s\Phi c\Theta \\ s\Phi s\Psi + c\Phi s\Theta c\Psi & -s\Phi c\Psi + c\Phi s\Theta s\Psi & c\Phi c\Theta \end{bmatrix} \quad (4.5)$$

representa a matriz de rotação que transforma um ponto do referencial *NED* para o referencial *ABC*,

$$S = R_{W2ABC} = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \cos\alpha\sin\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \quad (4.6)$$

é a matriz que transforma representações vetoriais do referencial do vento para o referencial ABC , e

$$\vec{v}_W = \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} \quad (4.7)$$

representa o vetor correspondente à direção do vento representado no referencial dos eixos do vento. Pelo mesmo raciocínio, a derivada de $\dot{\vec{r}}(t)$, utilizada nos cálculos de **N** e **B** (Kreyszig, 1991), pode ser obtida por meio da Equação 4.8.

$$\ddot{\vec{r}}(t) = \dot{\vec{v}}_{NED} = (\dot{B}^T S + B^T \dot{S}) \vec{v}_W, \quad (4.8)$$

onde \vec{v}_W é considerado um vetor constante com módulo igual à V_T e as derivadas das matrizes podem ser computadas a partir das relações diferenciais estabelecidas em 4.1 e nas variações dos ângulos de ataque e de derrapagem (Stevens e Lewis, 1992).

Por fim, o valor máximo de curvatura (κ_{max}) da trajetória da aeronave no espaço pode ser calculada a partir da maximização dos parâmetros cinemáticos, conforme mostrado na Equação 4.9.

$$\kappa_{max} = \max \left\{ \frac{|\vec{v}_{NED} \times \dot{\vec{v}}_{NED}|}{|\vec{v}_{NED}|^3} \right\}. \quad (4.9)$$

Já para o cálculo de τ_{max} , a idéia é equivalente, conforme pode ser visto na Equação 4.10.

$$\tau_{max} = \max \left\{ \frac{\vec{v}_{NED} \cdot [\dot{\vec{v}}_{NED} \times \ddot{\vec{v}}_{NED}]}{|\vec{v}_{NED} \times \dot{\vec{v}}_{NED}|^2} \right\}. \quad (4.10)$$

Da mesma forma, o ângulo de subida da trajetória da aeronave pode ser calculado a partir da Equação 4.11.

$$\theta(t) = \text{atan2}(\dot{z}_D) \quad (4.11)$$

Assim, fazendo-se uma analogia com a Equação 3.38, é possível demonstrar que θ_{max} é dado pela Equação 4.12.

$$\theta_{max} = \max \left\{ \|\vec{v}_{NED}\| \text{atan2}(\cos\alpha\cos\beta s\Theta - \sin\beta s\Phi c\Theta - \cos\alpha\sin\beta c\Phi c\Theta) \right\}. \quad (4.12)$$

Vale lembrar que os parâmetros cinemáticos utilizados para o cálculo dessas restrições dependem fortemente das características dinâmicas de cada veículo. Percebe-se que é possível projetar sistemas de controle que estabeleçam os valores das restrições (dentro de certas

condições) conforme descrito anteriormente. Mas a principal conclusão obtida aqui é a de que as restrições cinemáticas consideradas dependem fortemente da velocidade de vôo da aeronave, dentre outras características. Esse é o principal motivo pelo qual V_T é mantido constante para os testes realizados.

Capítulo 5

Arcabouço Experimental

Este capítulo descreve o aparato experimental utilizado neste trabalho. Utilizando a metodologia para a geração de trajetórias tridimensionais proposta anteriormente, foi executado o planejamento de navegação para duas diferentes aeronaves em ambiente de simulação. A primeira foi um aeromodelo virtual implementado dentro de um simulador de vôo, para o qual foram implementadas estratégias de controle para a navegação. Nesse caso foi criado um sistema *Hardware-in-the-loop (HWIL)* a partir de um computador de bordo utilizado no Projeto *SiDeVAAN*. A segunda aeronave foi o modelo matemático do VAANT construído no Projeto *AqVS*, que já contava com algoritmos de controle e navegação projetados para a aeronave real.

5.1 Sistemas de Simulação - Revisão

Sistemas *HWIL* são ferramentas de grande importância em projetos de engenharia. Foram inicialmente utilizados nos setores aeronáuticos e aeroespaciais, e atualmente a utilização de testes *HWIL* estende-se a diversas áreas de pesquisa, auxiliando a minimizar o tempo e os custos envolvidos no desenvolvimento de sistemas complexos (Gholkar et al., 2004). As principais tarefas envolvidas na elaboração de tais sistemas são a modelagem e a simulação. A modelagem permite reproduzir o comportamento de sistemas reais por meio de representações matemáticas, tão complexas quanto necessário, mas ainda mais simples do que os sistemas reais. Já a simulação permite que esses modelos sejam utilizados em lugar de sistemas reais para a realização de testes em caráter experimental a custos mais baixos, e com maior rapidez.

A principal característica de um sistema *HWIL* é a capacidade de execução em tempo real. Obviamente, para avaliar o comportamento de estratégias de controle e planejamento de um processo real, como um sistema robótico, deve-se levar em conta a dinâmica do processo. Sistemas de controle discreto, por exemplo, dependem de parâmetros como o intervalo de tempo de amostragem, e o atraso de tempo de resposta, para manter sua estabilidade durante a operação.

Existem diversos trabalhos na literatura que tratam do tema sobre ambientes de simulação para veículos aéreos. Em (Göktogian et al., 2003), os autores discutem a arquitetura de um

sistema *HWIL* de tempo real para *UAVs*, onde em especial, é apresentado um arcabouço de comunicação multi-sistemas. Deste trabalho nasceu o *RMUS* (*Real-Time Multi UAV Simulator*), uma biblioteca desenvolvida exclusivamente para a simulação de missões com o *UAV Brumby Mk III* do *Australian Centre for Field Robotics* (Wong, 2006). São estabelecidos vários mecanismos de teste e validação para missões, que permitem o estudo em um nível mais alto de inteligência para aeronaves autônomas. Em trabalhos posteriores, como (Göktogian e Sukkarieh, 2005) e (Göktogian et al., 2006), os autores estendem as funcionalidades desta bibliotecas, objetivando possíveis utilizações de técnicas baseadas em Visão Computacional e de Realidade Aumentada, para a navegação de *UAVs*.

Além da característica multi-agente, importante para estabelecer as etapas do planejamento de missões, outro fator importante a ser considerado no estágio de simulação é o caráter de visualização gráfica do ambiente. Alguns trabalhos, como (Castillo-Effen et al., 2005), por exemplo, apresentam plataformas com ambientes que permitem a inspeção visual do comportamento dos veículos, como trajetórias realizadas, acelerações laterais indesejáveis, entre outras características. Vale destacar também a aplicação de técnicas de navegação baseada em imagens, e outras estratégias fundamentadas na área de Visão Computacional.

Aspectos mais afins a simulação *HWIL* são abordados por (Gholkar et al., 2004), em um trabalho voltado para os testes de *hardware* de um *Micro-UAV* (*MUAV*), utilizando modelagem de sensores e simulação de dinâmica de vôo em um sistema *Linux* de tempo real. Os testes simulados permitiram a correção de várias situações que poderiam ser críticas em condições reais de vôo, justificando plenamente o uso necessário das ferramentas de simulação.

Os autores de (King et al., 2006) apresentam neste trabalho todo um arcabouço para o controle e coordenação de múltiplos veículos aéreos. Dentre as principais etapas do projeto, são apresentados os testes *HWIL* realizados com um sistema comercial. Um *cluster* de computadores utilizados para o planejamento em tempo real são conectados aos computadores de bordo de oitos pequenos *UAVs* produzidos no *MIT*. Posteriormente aos testes de simulação, foram realizados testes reais, e os resultados foram comparados à simulação.

Em (Sorton e Hammaker, 2005), os autores criam um sistema *HWIL* de baixo custo, utilizando apenas um simulador de vôo de código-aberto e uma ferramenta de cálculos matemáticos (*MatLab*). O simulador em questão é o *FlightGear Flight Simulator (FGFS)*, um pacote computacional desenvolvido inicialmente para fins acadêmicos, mas que tem se tornado um dos simuladores de vôo mais utilizados. Algumas de suas vantagens, além de ser em código-aberto, são a visualização gráfica, os protocolos de comunicação que suportam inúmeras máquinas para simulação multi-agente e modelagem dinâmica bastante realista.

Tomando como base essa última referência, foi desenvolvido nessa etapa do trabalho um sistema *HWIL* baseado no simulador de vôo *FGFS* para a realização de testes experimentais, conforme será descrito a seguir. Utilizando um computador de bordo de um *UAV* real, foram realizados diversos experimentos de controle e planejamento de trajetórias para uma dada aeronave virtual instanciada dentro do simulador.

5.2 Sistema *HWIL*

O passo inicial dessa fase do trabalho foi a implantação de um sistema *HWIL* para a realização de testes experimentais em ambiente simulado. Conforme mencionado anteriormente, a utilização desse recurso provê diversos benefícios, especialmente nas etapas iniciais de projeto, como redução nos custos experimentais, tratamento modular dos subsistemas da plataforma real e realização de testes em tempo real, entre outros.

Foi utilizado o simulador de vôo *FGFS*, o qual foi configurado para receber comandos de atuação e ler dados de sensores de aeronaves via interface de rede. Assim, foi estabelecida uma conexão entre o simulador e o computador de bordo de um *UAV* real, para o qual foram programadas estratégias de controle e de planejamento de trajetória para a aeronave virtual.

5.2.1 *FlightGear Flight Simulator*

O *FGFS* (Olson, 2007) é um simulador de vôo criado e gerenciado por *Curt Olson*, cujo principal objetivo é o de fornecer um sofisticado sistema de simulação de vôo para fins acadêmicos, como pesquisas na área de aviação, treinamento de pilotos e afins. Desenvolvido em linguagem de programação C++, a qualidade proporcionada por este *software* é equiparável aos mais avançados simuladores comerciais do mundo, como o *Flight Simulator* da *Microsoft*, apesar de ser de livre distribuição. Essa é uma das principais vantagens que tornam o *FGFS* tão utilizado e conhecido ao redor do mundo.

Outras vantagens de se utilizar esse simulador são: a característica de código-aberto, que permite modificar ou acrescentar módulos de programação; a compilação multi-plataforma que permite a utilização em diversos sistemas operacionais; e a implementação de diversos modelos conhecidos de dinâmica de vôos, também chamados *Flight Dynamic Models (FDMs)*.

Mas a maior vantagem da utilização do *FGFS* é a possibilidade do estabelecimento de conexões para a comunicação via rede com outros processos, o que permite a interação de uma instância do simulador, com outras instâncias na mesma ou em outras máquinas. Com isso, é possível utilizar o simulador como sendo um modelo de aeronave virtual para o sistema *HWIL*, permitindo a elaboração de testes de estratégias de controle e planejamento de trajetórias, por exemplo.

Via protocolos de rede é possível amostrar dados dos diversos sensores de uma aeronave, processá-los por meio de um controlador externo, e gerar ações de controle de volta para o simulador; ou ainda inicializar diversas estâncias do simulador em diferentes máquinas, estabelecendo uma rede de comunicação para o planejamento de missões com múltiplos *UAVs*. Na Figura 5.1, podemos ver a representação do *UAV (Rascal 110)*, um dos muitos tipos de modelos de aeronave implementados no *FGFS*, que vão desde pequenos aeromodelos como este, até aviões comerciais de grande porte como o *Boeing 747*.

Um dos principais modelos de dinâmica de vôo utilizados pelo simulador é o *JSBSim* (Berndt, 2004), uma sofisticada ferramenta, também de código-aberto, desenvolvida e distribuída gratuitamente. Essa ferramenta modela as forças e momentos aerodinâmicos utilizando um método clássico para determinação de coeficientes, que incluem efeitos de rotação da Terra



Figura 5.1: Modelo digital de uma aeronave *Rascal 110* implementado no *FlightGear Flight Simulator*.

(acelerações centrífuga e de Coriolis) e fenômenos atmosféricos, tornando a simulação bastante realista.

Outra importante vantagem do *FGFS* é a possibilidade de se extrair imagens da tela do simulador e enviá-las via rede para outros processos, o que permite, por exemplo, a implantação de um sistema de navegação baseada em Visão Computacional, para testes em missões mais sofisticadas, como inspeção visual de grandes áreas.

5.2.2 Arquitetura de *Hardware*

Para a realização de testes em *HWIL* foi utilizado o computador de bordo do Projeto *SiDe-VAAN*. Trata-se de um microcomputador padrão *PC/104* (Figura 5.2), com um processador NS GX1, DRAM de 512 MB e 256 KB de EPROM. Possui ainda saída VGA, entrada para *mouse* e teclado, duas portas seriais RS232, duas portas USB e conector 10/100 Base-T Ethernet. Acoplado ao módulo CPU existe uma placa de aquisição de dados de 16 canais, com entradas de ± 5 e ± 10 volts e resolução de 12 bits. Esse módulo é utilizado para digitalizar a leitura dos sensores da aeronave durante o vôo. Por fim, existe ainda um módulo de aquisição *GPS* para localização do veículo. As dimensões do computador são de aproximadamente $10 \times 10 \times 10$ centímetros, com um peso inferior a 800 gramas.

Esse *PC/104* foi equipado com sistema operacional *Linux* de tempo real (*RTAI*), no intuito de prover uma ferramenta de controle mais robusta ao sistema. O computador ainda é capaz de se comunicar, por meio de uma interface serial, com um controlador de servos *PicoPic* instalado na aeronave, que permite acionar eletronicamente as superfícies de controle como leme, profundor e *ailerons*, e comandar a potência do motor.

Utilizando-se da capacidade do *FGFS* de se comunicar com outros processos via interface de rede, foram estabelecidos dois canais de comunicação entre o *PC/104* e a máquina responsável pela execução do simulador, para a troca de informações em tempo real. Em um deles, trafegam dados referentes aos sensores da aeronave no sentido do simulador para o computador de bordo, enquanto que no outro, trafegam, no sentido inverso, sinais de comando para os atuadores do veículo.



Figura 5.2: Computador de bordo, módulo *PC/104*, utilizado no Projeto *SiDeVAAN* (UFMG).

A comunicação entre os *hardwares* é feita por meio do protocolo de rede *UDP*, escolhido por apresentar maior velocidade na troca de informações, apesar de sua menor confiabilidade. Na Figura 5.3 é possível ver a arquitetura de comunicação, composta por dois canais unidirecionais. A estrutura *FGNetFDM* representa um pacote de dados contendo todas as informação de vôo da aeronave (latitude, longitude, altitude, velocidade, pressão atmosférica, etc), ao passo que a *FGNetCtrls* corresponde aos dados que comandarão o veículo (aceleração, posição do trem de pouso e superfícies de controle).

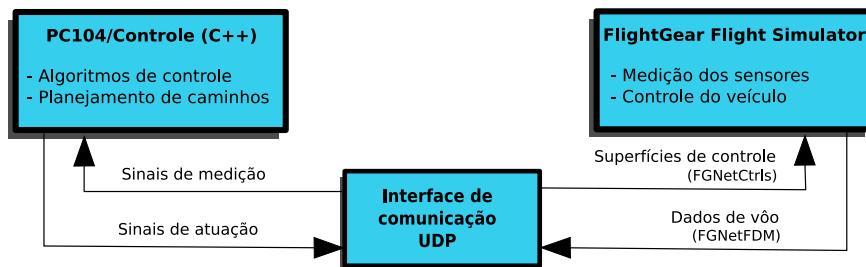


Figura 5.3: Arquitetura de comunicação entre o computador de bordo *PC/104* e o simulador de vôo *FlightGear Flight Simulator*.

5.2.3 Estratégias de Controle

A seguir são apresentadas as estratégias de controle implementadas no computador de bordo. Elas foram projetadas para o controle do modelo simulado da aeronave *Rascal 110*. São quatro os módulos básicos de controle, sendo o último deles, a composição dos três primeiros. São eles:

- controle de velocidade, necessário para manter a velocidade do veículo constante durante a execução da trajetória, conforme assumido na etapa de planejamento;
- controle de altitude, responsável por controlar longitudinalmente a aeronave, levando-a a altitudes estabelecidas pela trajetória tridimensional $\vec{r}(t)$;

- controle de orientação, capaz de controlar a direção de movimento da aeronave em relação ao plano da Terra, tomando como principal referência, o eixo Norte do referencial NED;
- controle de navegação, capaz de conduzir a aeronave até pontos de latitude, longitude e altitude especificados por uma dada missão, no espaço tridimensional de navegação.

Essa estrutura de controle não representa nem a melhor e nem a única possível. De fato, o controle serve apenas para permitir que uma aeronave seja capaz de seguir, de maneira autônoma, a trajetória programada. O desacoplamento dos movimentos tem a vantagem de permitir que a tarefa de controle seja modularizada de uma maneira bastante simples. A desvantagem é a perda de desempenho do sistema do ponto de vista dinâmico, já que os movimentos do veículo aéreo ficam bastante restritos, conforme discutido mais adiante.

Em seguida, apresenta-se mais detalhadamente cada um dos módulos de controle. Um tipo bastante simples de controlador é utilizado para compor os módulos de controle.

5.2.3.1 Controlador PID

O controlador utilizado nessa tarefa é o PID. É caracterizado por apresentar em sua saída, um sinal de comando que é proporcional ao valor, à derivada (ou diferença) e à integral (ou somatório) do sinal de entrada. O sinal de entrada de um controlador é geralmente o erro entre o valor da variável do processo a ser controlada e a referência desejada para esta. Por convenção, denominam-se u_n e e_n a saída e a entrada do controlador, respectivamente no instante de tempo discreto n .

Existem diferentes formas de se modelar matematicamente a equação de um controlador deste tipo. Como o objetivo foi aumentar a suavidade do processo, a ação de controle foi implementada de maneira incremental conforme apresentado na Equação 5.1. Em cada intervalo de tempo é calculado apenas um pequeno incremento Δu_n (em vez de um novo u_n), que é acrescentado ao último sinal de comando gerado (u_{n-1}). Isso evita que variações muito bruscas sejam aplicadas as entradas de controle da aeronave.

$$u_n = u_{n-1} + \Delta u_n. \quad (5.1)$$

O incremento Δu_n , por sua vez, é calculado pela Equação 5.2, onde K_p , T_i e T_d são os parâmetros de ganho proporcional, tempo integral e tempo derivativo, respectivamente. O ajuste destes parâmetros é responsável pelo bom desempenho do sistema de controle, incluindo a estabilidade do processo.

$$\Delta u_n = K_p \left[(e_n - e_{n-1}) + \frac{T_s}{T_i} e_n + \frac{T_d}{T_s} (e_{Dn} - 2e_{Dn-1} + e_{Dn-2}) \right], \quad (5.2)$$

onde

$$e_n = r_n - y_n \quad (5.3)$$

e

$$e_{Dn} = \frac{e_{Dn-1} - y_n \left(\frac{10T_s}{T_d} \right)}{\left(\frac{10T_s}{T_d} + 1 \right)}. \quad (5.4)$$

Por se tratar de um sistema de controle discreto, temos o fator T_s , que representa o tempo de ciclo de uma interação do controlador, ou em outras palavras, o inverso da freqüência de controle em tempo real.

O erro, considerado como entrada do controlador, é calculado pela diferença entre o valor da variável de processo a ser controlada (aqui denominada y_n) e o valor de referência desejado para esta (chamado de r_n), conforme apresentado na Equação 5.3. O controlador atua no sentido de minimizar o valor de e_n calculado ao longo do tempo.

Ainda para o cálculo de Δu_n , a derivada da saída é calculada por meio de uma aproximação numérica, em função de e_{Dn} , chamado de erro derivativo filtrado, calculado pela Equação 5.4. Nesse caso, o erro é tomado como sendo $-y_n$, assumindo-se r_n igual à zero, o que acarreta em uma diminuição dos sinais de alta-freqüência de e_n . A ação derivativa nessa estrutura de controle apresenta um efeito estabilizante e ajuda a rejeitar perturbações na saída.

5.2.3.2 Ajuste dos Controladores

Existem diversos métodos empregados para realizar o ajuste de controladores, alguns mais simples e menos confiáveis, e outros mais sofisticados e robustos. Um dos métodos mais utilizados é o ajuste de ganhos por meio de tentativa e erro, onde os parâmetros do controlador são variados, de maneira aleatória (ou não), visando o melhor desempenho da malha fechada de controle. Tal método é atrativo especialmente por não necessitar de nenhum conhecimento sobre o sistema ou da teoria de controle envolvida. Entretanto, apresenta a principal desvantagem do dispêndio excessivo de tempo para o alcance de bons resultados.

Outra técnica mais simples e melhor fundamentada é o método de *Ziegler-Nichols* (Åstrom e Hagglund, 2004). Esse processo propõe alguns passos que auxiliam na sintonia de controladores do tipo P (proporcional), PI (proporcional e integral) e PID, minimizando o tempo gasto e praticamente eliminando a necessidade de análises mais profundas das malhas de controle. O problema com esse método, como no caso anterior, é a necessidade de se interagir com o processo em funcionamento, o que pode ser inviável em casos como processos industriais, sistemas embarcados de aeronaves, entre outros de custo muito elevado. Já no caso de simulação de sistemas, o método pode ser utilizado sem restrições.

5.2.3.3 Controle de Velocidade

A primeira estratégia de controle implementada foi o controle de velocidade. Durante o vôo de uma aeronave, pode-se identificar dois tipos diferentes de velocidade: a velocidade em relação ao vento (V_T), também conhecida com *airspeed*; e a velocidade em relação ao solo (V). A principal diferença entre as duas está basicamente na forma com a velocidade é

medida. Enquanto para a primeira utiliza-se um tubo de *Pitot* (Doeblin, 2004), para a última utiliza-se um *GPS*, por exemplo.

Como o intuito nesse caso é simplesmente manter a velocidade de vôo aproximadamente constante durante a execução de uma trajetória, será utilizado a variável V_T como variável de controle, uma vez que os sensores de velocidade do ar podem ser amostrados a uma freqüência mais alta do que o *GPS*, por exemplo. A Figura 5.4 apresenta o diagrama de blocos do sistema de controle de velocidade, modelado de uma forma bastante simples.

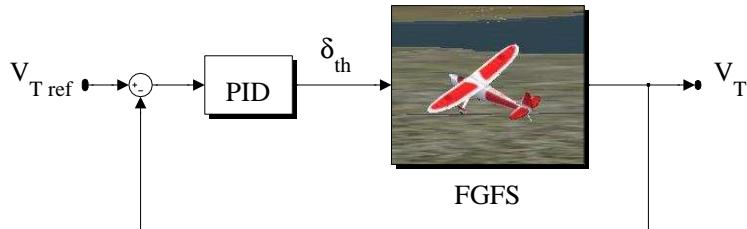


Figura 5.4: Diagrama de controle de velocidade da aeronave.

Aqui, o erro e_n é calculado como sendo a diferença entre o sinal de referência r_n representado por $V_{T\ ref}$, e a variável de processo y_n equivalente à V_T , que é o que se deseja controlar. Ambas são medidas em fps¹. Utiliza-se um controlador PID para calcular a ação de comando u_n , que neste caso representa o atuador de aceleração do motor da aeronave δ_{th} (*throttle*).

5.2.3.4 Controle de Altitude

O controle de altitude é bastante importante para os veículos aéreos autônomos, especialmente por que muitas das missões outorgadas podem incluir trajetórias por grandes áreas com diferentes níveis de terreno, como o caso da inspeção de linhas de transmissão de energia, por exemplo. Mesmo nos casos em que desejamos utilizar as estratégias de geração de trajetórias bidimensionais para veículos aéreos, é preciso considerar a restrição de movimentação apenas em planos de altitude constante, sendo necessário para isso, um sistema de controle eficiente, capaz de manter o nível sem grandes variações.

Novamente, há mais de uma forma de se medir a altitude de um veículo. Podem ser utilizados sensores analógicos baseados em pressão atmosférica, mais precisos, rápidos e confiáveis, ou o próprio *GPS*, menos comum devido aos problemas de imprecisão e falhas. Existem ainda dois tipos de altitudes medidas: a chamada altitude barométrica que representa a elevação do veículo em relação ao nível do mar; e a pressão relativa, que é calculada da mesma forma, porém tomando-se como base a altitude no local da decolagem (nível do aeroporto).

O *FGFS*, nesse caso, fornece apenas a altitude barométrica, a qual é utilizada diretamente na malha de controle. Na Figura 5.5 podemos observar o diagrama de blocos do sistema de controle de altitude, que atua em dois estágios separadamente. O primeiro estágio é composto por um controlador PID, que recebe como entrada, o erro referente à altitude desejada (z_{ref}) e a altitude real medida na aeronave (z), ambas descritas em relação ao nível do mar, e

¹Feet per second - Pés por segundo.

dimensionadas em metros no simulador. Na saída é apresentado um sinal de controle referente à taxa de variação da altitude (*climb rate*), que é utilizado como sinal de referência (\dot{z}_{ref}) no segundo estágio de controle.

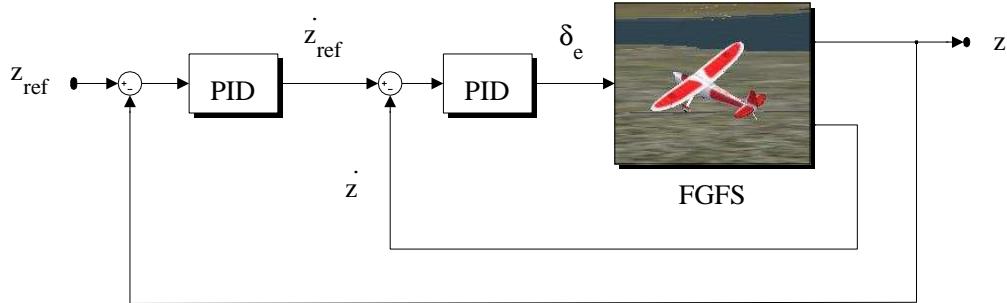


Figura 5.5: Diagrama de controle de altitude da aeronave.

A referência produzida é então comparada ao sinal do sensor de um velocímetro vertical (\dot{z}) instalado na aeronave, gerando um novo sinal de erro que alimenta o controlador do tipo PID, que finalmente gera um sinal de comando de atuação para a aeronave. A variável δ_e representa a deflexão da superfície de controle chamada de profundor (*elevator*).

5.2.3.5 Controle de Orientação

O objetivo do sistema de controle de orientação é manter a aeronave seguindo um curso de vôo pré-estabelecido, tomando-se como referência à indicação de Norte. Em outras palavras, controlar o ângulo de guinada (Ψ) correspondente a uma direção de vôo, paralela ao plano do chão, como se o veículo se comportasse como um robô terrestre movimentando-se no plano terrestre.

Assumindo o modelo látero-direcional da aeronave, serão utilizados aqui os momentos de rolamento e guinada para o controle de orientação do veículo. A Figura 5.6 apresenta o diagrama de controle empregado nesse caso. Novamente temos um controle em dois estágios, onde cada estágio é composto por um controlador do tipo PID apresentado anteriormente.

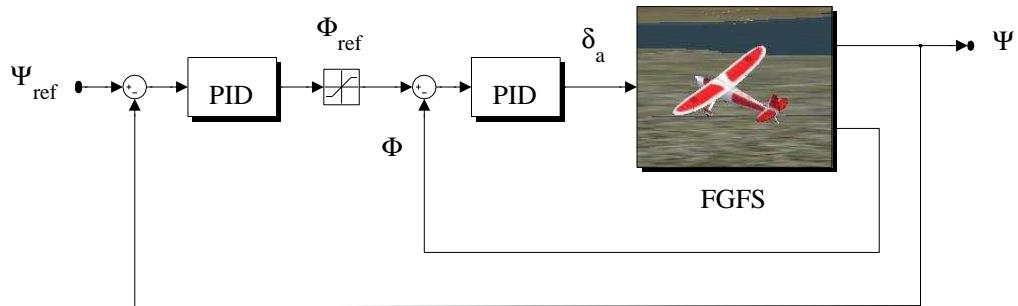


Figura 5.6: Diagrama de controle de orientação da aeronave.

O parâmetro Ψ_{ref} representa um curso de navegação a ser seguido pelo robô, dado em graus. O valor de 0° (equivalentemente a 360°), por definição, representa a direção Norte. Já Ψ corresponde a uma medição do sinal de orientação da aeronave, que novamente pode vir de

diferentes tipos de sensores embarcados (*GPS*, bússola, sensores inerciais). Mais comum para esses casos é a utilização de magnetômetros, que estimam a orientação do campo magnético da Terra.

O erro de orientação alimenta o controlador do primeiro estágio, que gera um sinal de referência para o próximo, denominado Ψ_{ref} . Para fins de aumento da estabilidade do sistema como um todo, o momento de rolamento será utilizado para controlar a orientação da aeronave, e por esta razão, o segundo estágio de controle funciona como um sistema de controle do ângulo de rolamento. O novo erro gerado, produz na saída do segundo controlador, um sinal de controle para os ailerons da aeronave, que são atuados para realizar curvas e orientar a aeronave segundo Ψ_{ref} .

Outro ponto importante é a inclusão de um sistema de saturação capaz de limitar a saída do controlador nesse caso. Ele impede que os *ailerons* da aeronave se movimentem em amplitudes muito elevadas, limitando o ângulo de rolamento. Sua principal função é evitar que a aeronave se incline demais ao fazer uma curva, gerando assim uma possível falta de sustentação que venha a influenciar o controle de altitude.

5.2.3.6 Controle de Navegação

Um nível acima na hierarquia do sistema de controle de baixo-nível, encontra-se o controle de navegação da aeronave. A principal função desse sistema é gerar sinais de referência para os controladores descritos acima, baseado nas especificações de localização da aeronave geradas pelos sistema de planejamento.

Dado um *waypoint* \vec{P}_{ref} referente ao vetor $[x_{ref}, y_{ref}, z_{ref}]$ formado pelas variáveis de longitude, latitude e altitude, respectivamente, o controle de navegação atua no sentido de reduzir o erro em relação ao \vec{P} real medido pelos sensores da aeronave em um determinado instante de tempo. O sistema pode ser esquematicamente representado pela Figura 5.7, onde se pode ver que o controle de posição gera sinais separados para cada um dos três controladores descritos anteriormente, a fim de minimizar o erro de posição.

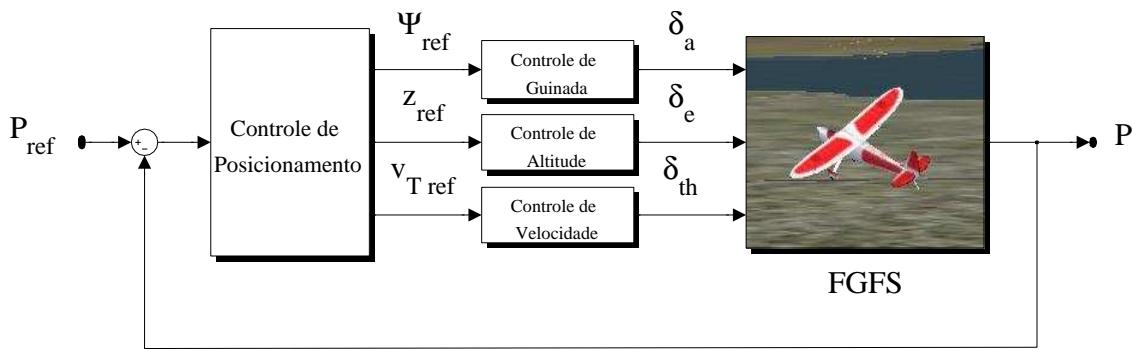


Figura 5.7: Diagrama de controle de navegação da aeronave.

O curso de vôo Ψ_{ref} pode ser determinado simplesmente como a orientação de um robô se movimentando em um plano. O controlador de posição utiliza a Equação 5.5 para estimar o valor da referência para o controlador de orientação, onde x_{ref} e y_{ref} são os valores de referê-

cia para a longitude e latitude da aeronave, respectivamente. Já x e y são os valores medidos de longitude e latitude, facilmente obtidos por meio de um *GPS*. Essa é uma das formas mais comuns de se medir a posição, e infelizmente está sujeita aos problemas já comentados.

$$\Psi_{ref} = \frac{\pi}{2} - \text{atan2} \left(\frac{y_{ref} - y}{x_{ref} - x} \right). \quad (5.5)$$

O *waypoint* é finalmente alcançado quando a distância euclidiana no espaço tridimensional entre \vec{P}_{ref} e \vec{P} é menor ou igual a certo limiar ($dist_{min}$), previamente escolhido, conforme calculado na Equação 5.6.

$$e_n = \sqrt{(x_{ref} - x)^2 + (y_{ref} - y)^2 + (z_{ref} - z)^2} \leq dist_{min}. \quad (5.6)$$

5.3 Aeronave *AqVS*

Em uma segunda etapa de experimentos, foi utilizado o modelo matemático do VAANT *AqVS* para testar a metodologia proposta. Essa aeronave não-tripulada de pequeno porte foi construída dentro do contexto do Projeto *SiDeVAAN*, resultando em um motoplanador autônomo com um grande potencial de realização de tarefas de reconhecimento e monitoramento de terrenos (Iscold, 2007). Centenas de vôo autônomos já foram realizados com sucesso desde o ano de 2006. A Tabela 5.1 apresenta alguns dos dados técnicos dessa aeronave.

Tabela 5.1: Dados técnicos do *AqVS*.

Envergadura	2,00 m	Comprimento	1.6 m
Carga alar	1 kgf	Velocidade de cruzeiro	50 km/h
Peso Vazio	1.2 kgf	Peso Total	2.5 kgf
Teto Oper.	150m AGL	Raio Oper.	10 km

O modelo matemático utilizado neste trabalho foi concebido em duas etapas. Para facilitar as tarefas de controle e simulação de veículos aéreos, em geral, utiliza-se a estratégia de dividir o comportamento da aeronave em dois tipos de movimentos (Stevens e Lewis, 1992).

O primeiro é o movimento látero-direcional, onde se descreve espacialmente a aeronave apenas em função de seus movimentos paralelos ao plano da Terra. Nessa aproximação, apenas as variações de posição nos eixos **X** e **Y** do referencial fixo e a orientação em Ψ são considerados, conforme ocorre no caso bidimensional. Considera-se, nesse caso específico, que o ângulo de arfagem Θ , a velocidade angular Q , a velocidade linear W e o ângulo de ataque α sejam nulos durante a realização do movimento.

O segundo é o movimento longitudinal, que gera a variação de altitude da aeronave ao longo do eixo **Z** da Terra. Para esse caso, considera-se nulos os valores do ângulo de rolamento Φ , das velocidades angulares P e R , e do ângulo de derrapagem β . As Figuras 5.8 e 5.9 apresentam diagramas esquemáticos dos modelos látero-direcional e longitudinal utilizados.

Nessa etapa de experimentos, foram utilizados os dois modelos separadamente. As trajetórias tridimensionais produzidas na descrição da metodologia foram decompostas como

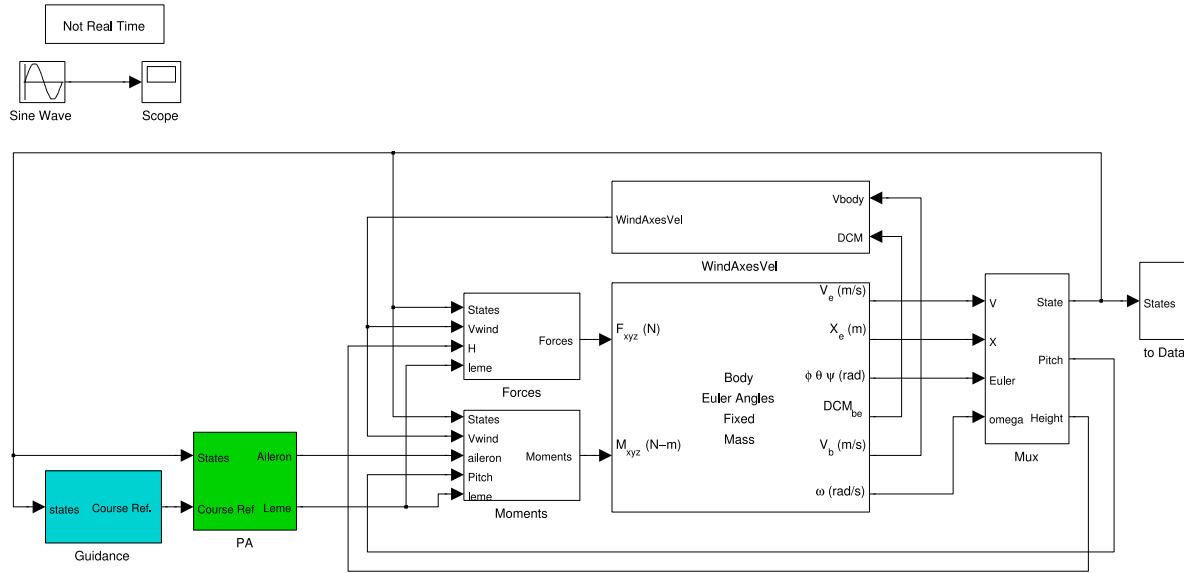


Figura 5.8: Modelo látero-direcional do AqVS.

Modelo Dinamico de Corpo Rígido

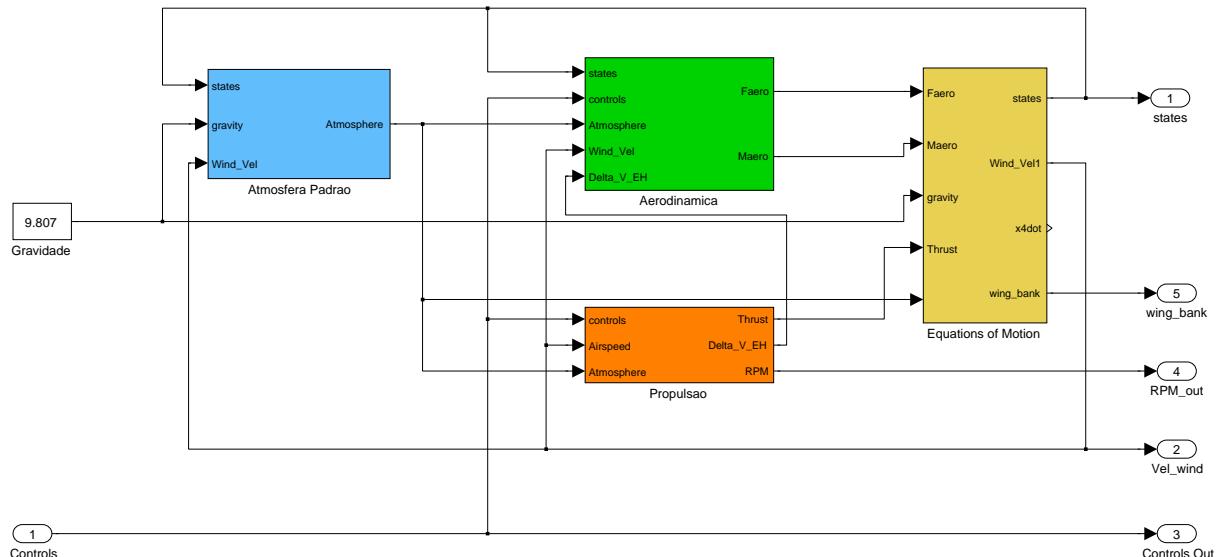


Figura 5.9: Modelo longitudinal do AqVS.

movimentos em **X** e **Y** (látero-direcionais) e movimentos em **Z** (longitudinais). A estrutura de controle utilizada no modelo para seguir trajetórias foi à mesma utilizada na aeronave real.

Capítulo 6

Resultados

Neste capítulo são discutidos alguns dos principais resultados obtidos neste trabalho. Inicialmente são apresentados os caminhos gerados a partir da técnica do *DP* 3D proposta, para um dado conjunto de *waypoints* tridimensionais. O mesmo conjunto é considerado na seqüência, porém utilizando-se a técnica para a geração de trajetórias com perfis contínuos de curvatura e torção. Uma comparação entre esses dois resultados é realizada em termos do comprimento dos caminhos gerados. Por fim, são apresentados conjuntos de trajetórias aplicáveis as duas aeronaves autônomas consideradas no capítulo anterior.

6.1 *Dubins' Path* 3D

Nesta seção são apresentados alguns resultados relativos ao planejamento de caminhos utilizando-se o método do *DP* 3D proposto neste trabalho. Para ilustrar a aplicação dessa técnica, considerou-se nesse primeiro teste uma aeronave cujos valores de restrições foram arbitrariamente escolhidos, a saber:

$$\begin{aligned}\rho_{min} &= 10 \text{ metros}, \\ \sigma_{min} &= 100 \text{ metros}, \\ \theta_{max} &= \frac{\pi}{4} \text{ radianos}.\end{aligned}$$

Após essas definições, foram estabelecidos sete pontos no espaço, pelos quais a aeronave deveria passar. Esses *waypoints* foram ordenados de forma arbitrária, e referenciados por índices de j a p , conforme mostrado a seguir:

$$\begin{aligned}\mathbf{P}_j &= (0, 0, 20, 0, 0), \\ \mathbf{P}_k &= (50, 0, 40, \frac{\pi}{2}, \frac{\pi}{4}), \\ \mathbf{P}_l &= (50, 50, 60, \frac{\pi}{2}, 0), \\ \mathbf{P}_m &= (50, 100, 60, -\frac{\pi}{2}, 0), \\ \mathbf{P}_n &= (50, -50, 80, \pi, -\frac{\pi}{6}) \\ \mathbf{P}_o &= (0, 60, 40, \frac{5\pi}{4}, 0) \text{ e} \\ \mathbf{P}_p &= (0, 30, 20, \pi, 0).\end{aligned}$$

Para cada par de *waypoints*, tomados seqüencialmente nesse conjunto, foi efetuado o cálculo do *DP* 3D, considerando-se as restrições cinemáticas impostas. Uma vez que a técnica proposta apresenta um caráter iterativo, um novo caminho tridimensional é gerado a cada passo. Conforme pode ser visto na Figura 6.1, cada caminho requer um número diferente de iterações, com diferentes tempos computacionais gastos para a produção do resultado final.

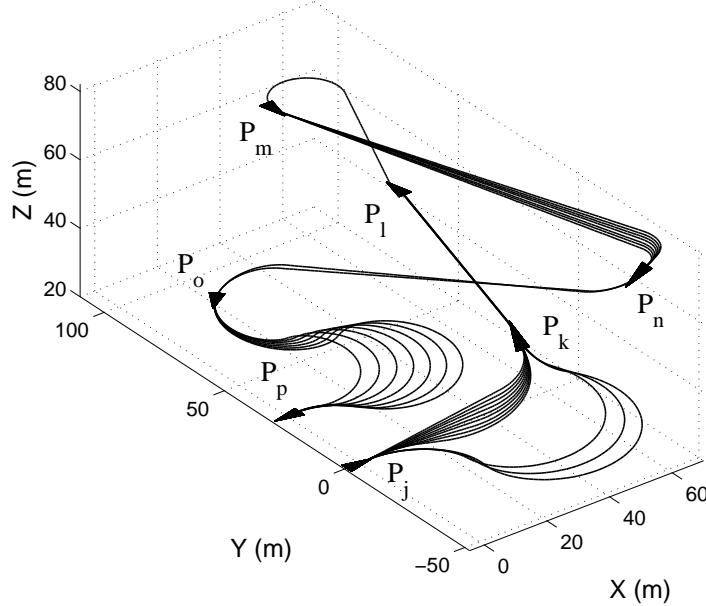


Figura 6.1: Iterações no cálculo do *Dubins' Path* 3D para o conjunto de *waypoints*.

O resultado final da aplicação do *DP* 3D ao conjunto de *waypoints* pode ser visto na Figura 6.2. Já os perfis de curvatura, torção e inclinação são apresentados na Figura 6.3. É possível notar que, não apenas os valores restritivos foram respeitados (o que já era de se esperar), como também os perfis de $\kappa(t)$ e $\tau(t)$ apresentaram descontinuidades.

É preciso ressaltar que o conjunto de *waypoints* foi escolhido de forma a gerar as mais diversas situações para a resolução do algoritmo. As configurações escolhidas podem ser consideradas críticas, especialmente por apresentarem pontos muito próximos uns dos outros.

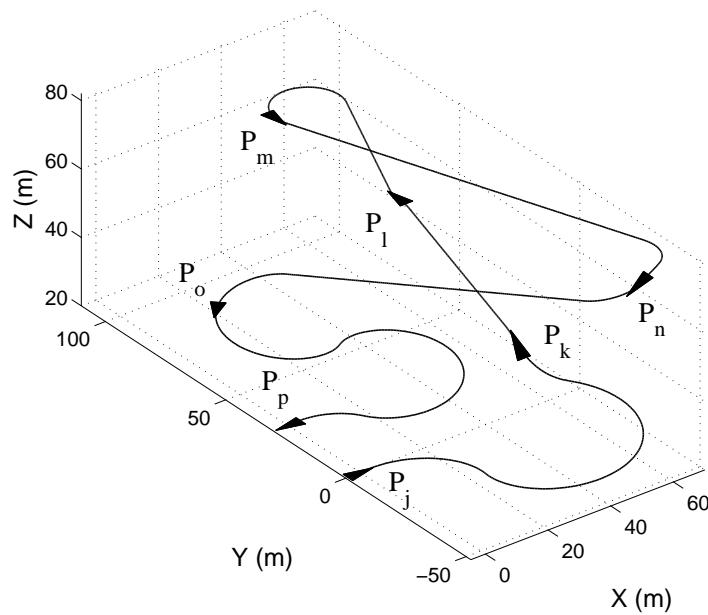


Figura 6.2: Resultado final do *Dubins' Path 3D* para o conjunto de *waypoints*.

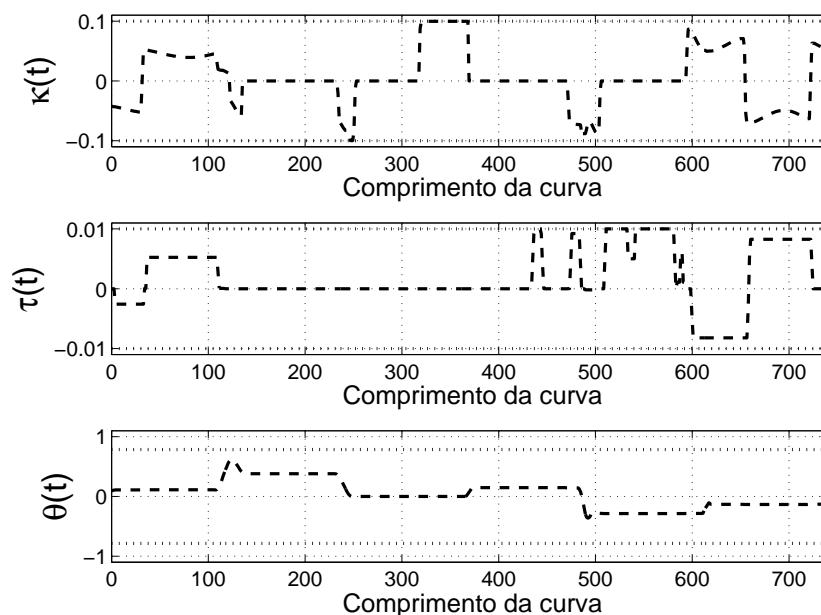


Figura 6.3: Avaliação das restrições cinemáticas para o *Dubins' Path 3D*.

6.2 Trajetórias Utilizando Curvas de Bézier de 7^a Ordem

O mesmo teste foi aplicado para o caso da metodologia de geração de trajetórias tridimensionais com perfis contínuos de curvatura e torção. As mesmas restrições e o mesmo conjunto de *waypoints* foram utilizados para traçar os resultados dessa etapa.

A Figura 6.4 apresenta um resultado comparativo entre o *DP* 3D gerado anteriormente e essa técnica. Pelos valores dos comprimentos calculados para cada uma dessas curvas, conclui-se que a curva de 7^a ordem é aproximadamente 28% maior do que o caminho de *Dubins*.

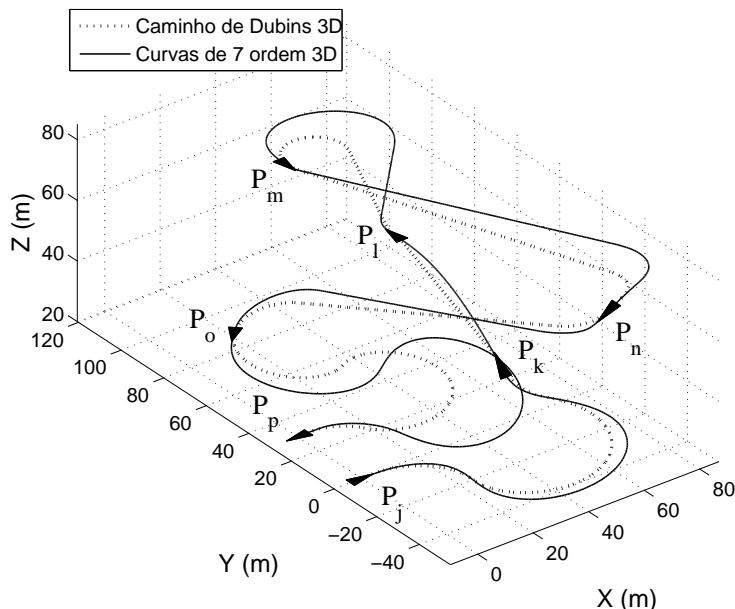


Figura 6.4: Comparação entre o caminho de *Dubins* e as trajetórias baseadas em Béziers de 7^a ordem.

Esse resultado é significativo, se comparado aquele gerado pelo método do HP 3D, visto na Figura 6.5. Nesse caso, o comprimento da curva é superior a 120% do *DP* 3D. Além disso, a convergência do algoritmo para esse caso específico despendeu de um tempo cerca de 10 vezes maior do que as outras técnicas.

Já os comportamentos dos parâmetros de restrição das curvas podem ser comparados pela análise das Figuras 6.6 e 6.7. No caso das curvas de 7^a ordem, constata-se que os perfis de curvatura e torção são contínuos ao longo de toda a trajetória, o mesmo não ocorrendo com o HP 3D.

Novamente, o conjunto de *waypoints* utilizado incorpora situações bastante críticas para o problema de planejamento. É preciso ressaltar que se observou empiricamente que a medida que as poses tendam a se afastar no espaço, os resultados gerados pelas duas metodologias (pelo menos em termos do comprimento da curva) tendem a se tornar equivalentes.

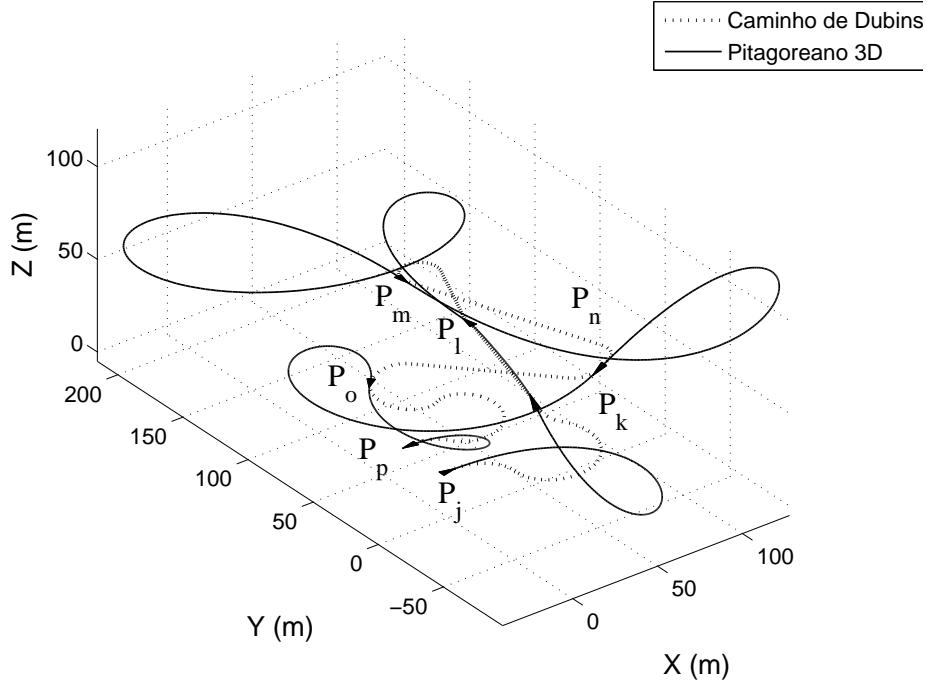


Figura 6.5: Comparação entre o caminho de *Dubins* e o Hodográfico Pitagoreano 3D.

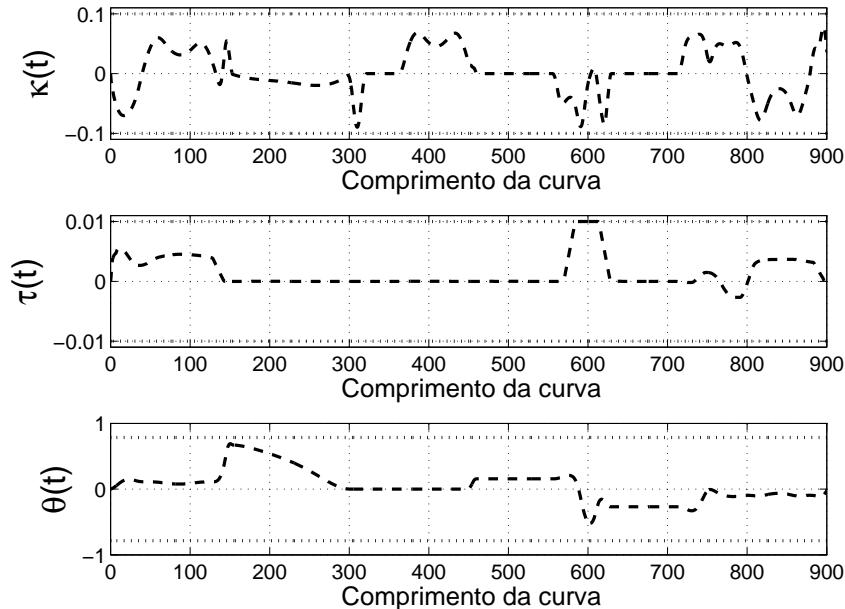


Figura 6.6: Avaliação das restrições cinemáticas para as curvas de Bézier de 7^a ordem.

6.3 Simulação de Vôo

Foram ainda realizados testes com a utilização de dois veículos aéreos simulados, conforme descrito na parte experimental deste trabalho. O passo inicial para os dois casos foi o levantamento dos valores das restrições cinemáticas de cada veículo. A partir desses dados, foram geradas trajetórias tridimensionais para esses veículos, as quais foram utilizadas como

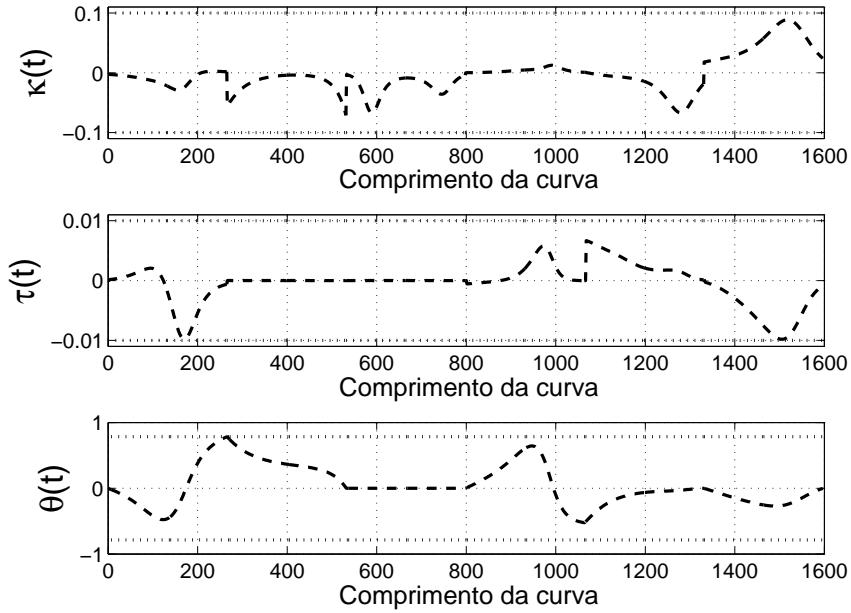


Figura 6.7: Avaliação das restrições cinemáticas para o Hodográfico Pitagoreano.

referência de navegação.

6.3.1 Sistema *HWIL* e o *FlightGear*

Inicialmente, foram realizados diversos experimentos para a implantação do sistema *HWIL*. Tendo como base o modelo matemático da aeronave *Rascal 110* (Figura 5.1) instanciada no *FGFS*, levantou-se os parâmetros dos controladores de cada módulo do sistema de controle descrito no capítulo anterior. Os valores de K_p , T_i e T_d foram determinados segundo a metodologia em malha aberta de *Ziegler-Nichols* apresentada em (Åstrom e Hagglund, 2004). Essa foi uma etapa muito importante, porque o ajuste desses parâmetros determina diretamente os valores das restrições cinemáticas da aeronave.

Muitas vezes, os sistemas de controle projetados para atuar em plantas cuja dinâmica é muito rápida e instável (como é o caso das aeronaves leves), promovem um aumento no tempo de resposta das variáveis controladas, a fim de tornar o processo estável como um todo. Para tornar mais simples a tarefa de controle de navegação de uma aeronave, como é o caso do aeromodelo utilizado, os módulos de controle foram projetados para reduzir a capacidade de manobra do veículo, tornando-o tão lento (e consequentemente menos instável) quanto um avião de médio e grande porte.

Essa estabilidade foi alcançada não somente reduzindo a velocidade de resposta por meio dos algoritmos de controle, mas também limitando os valores que esses aplicam às superfícies de atuação da aeronave. Por exemplo, no primeiro estágio do sistema de controle de orientação observado na Figura 5.6, os valores calculados para o ângulo de rolamento da aeronave são limitados ao intervalo de ± 20 graus, para evitar que a aeronave perca a sua altitude enquanto se orienta. Esse obviamente é um fator limitante para a taxa máxima de curvatura (κ_{max}) que a aeronave é capaz de realizar. Outro fator seria a velocidade do veículo em relação ao

vento, que é mantida constante em cerca de 60 km/h em todos os testes realizados para esse veículo.

Os testes apresentados a seguir foram utilizados para determinar as restrições de movimento da aeronave, como o raio mínimo de curvatura ρ_{min} , o raio mínimo de torção (σ_{min}) e o ângulo máximo de subida θ_{max} .

6.3.1.1 Determinação do Raio Mínimo de Curvatura

O raio mínimo de curvatura foi determinado a partir de um teste de controle simulado, onde foi comandado ao veículo que mantivesse um vôo à altitude e velocidade constantes. Assim, foi estabelecida uma referência de orientação de exatamente 90 graus para o lado esquerdo em relação ao curso real da aeronave. Isso fez com que o veículo se mantivesse voando em círculos, aplicando o valor máximo de curvatura permitida.

O mesmo teste foi aplicado também em curvas para o lado direito, no intuito de observar a influência do vento lateral no módulo de controle de orientação. Utilizando-se uma rajada de vento com velocidade de aproximadamente 10 quilômetros por hora, e soprando com um curso de cerca de 300 graus em relação ao Norte da Terra, nenhuma influência significativa foi observada.

A partir da trajetória observada na Figura 6.8, e utilizando-se as Equações 3.3 e 3.4, o raio mínimo de curvatura para o aeromodelo, dados os ajustes de controle determinados, foi de aproximadamente 0,004 graus (valor dado em termos da longitude e latitude no plano terrestre) ou aproximadamente 450 metros. Apesar de o teste ter sido realizado apenas para o caso de um movimento látero-direcional, esse valor será assumido como sendo a curvatura espacial, já que o controle de altitude foi projetado para tornar o raio de curvatura do movimento longitudinal maior do que esse último.

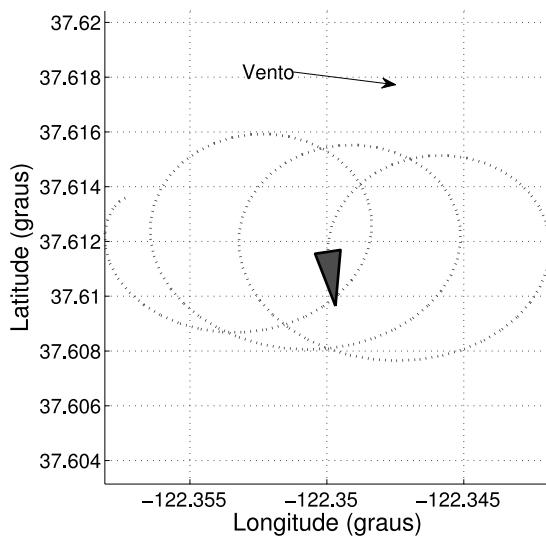


Figura 6.8: Teste para determinação do raio mínimo de curvatura da aeronave.

6.3.1.2 Determinação do Raio Mínimo de Torção

Um teste semelhante ao anterior foi utilizado para determinar o valor de σ_{min} da aeronave utilizada. Porém, ao invés de forçar a máxima curvatura do veículo mantendo o movimento sobre um plano fixo de altitude, aplicou-se um degrau de referência para variar maximamente a distância do aeromodelo até o solo. Isso acarretou uma trajetória de subida em espiral (Figura 6.9), cujos valores apresentados de curvatura e torção são constantes ao longo do tempo. O mesmo teste foi aplicado também para uma trajetória em descida, sem qualquer diferença perceptível no resultado final.

Para calcular o valor de torção máxima da aeronave, o resultado da trajetória foi aplicada à Equação 3.37. Porém, como o FGFS fornece valores de altitude em metros, e de longitude e de latitude em graus, foi necessário utilizar uma função de conversão de unidades para equiparar os eixos do espaço tridimensional. Com isso, o valor obtido para σ_{min} foi de aproximadamente 0,0156 graus, ou cerca de 1750 metros.

O valor do raio de curvatura obtido nesse teste foi aproximadamente equivalente ao valor calculado anteriormente.

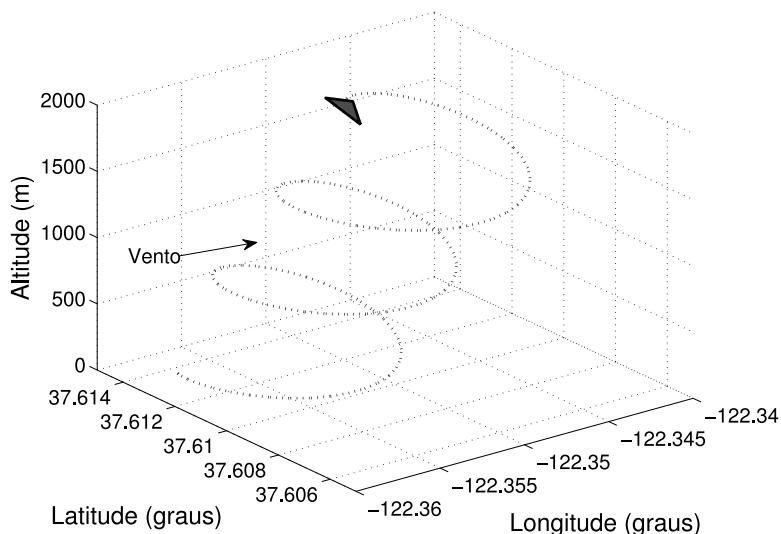


Figura 6.9: Teste para determinação da máxima taxa de torção da aeronave.

6.3.1.3 Determinação do Ângulo Máximo de Subida

A Figura 6.10 apresenta a resposta ao degrau do sistema de controle de altitude. Nesse teste simples, a aeronave foi comandada para voar em linha reta mantendo sua altitude constante a cerca de 100 metros do nível do mar. Em um determinado instante de tempo, a referência do controlador foi alterada para 300 metros, conforme é possível observar no gráfico. A linha contínua representa a resposta da aeronave com controle em malha fechada, após a mudança de comando na referência do controlador de altitude. O tempo de assentamento para o caso da aplicação de um degrau de 200 metros foi de aproximadamente 30 segundos.

Observando-se o valor da inclinação da curva durante a execução desse trecho da trajetória, constatou-se que o valor de θ_{max} atingido para esse caso foi inferior à 5 graus. Esse é um valor relativamente baixo se comparado à capacidade de manobra do aeromodelo em malha aberta. Porém, como foi dito anteriormente, essa é uma das consequências de se aplicar um controlador visando estabilizar o comportamento desse sistema.

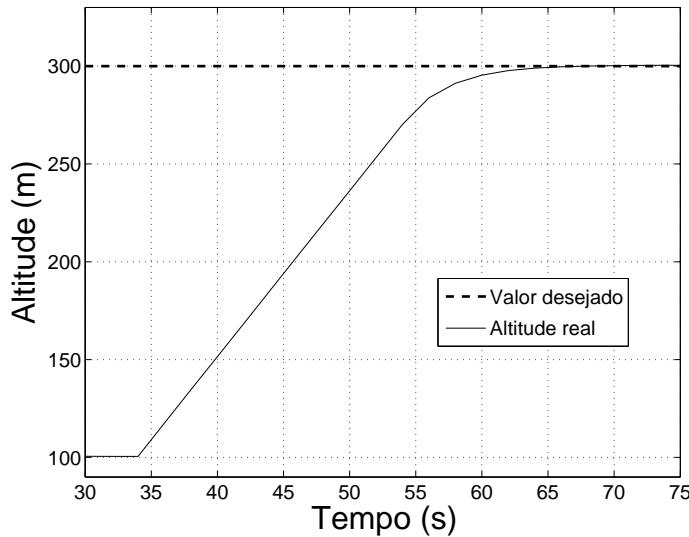


Figura 6.10: Resposta ao degrau do sistema de controle de altitude.

6.3.1.4 Testes de Navegação

Por fim, foram realizados alguns testes para avaliar o comportamento do sistema de controle de navegação implementado. Em dois desses testes foram utilizadas as metodologias de geração de trajetórias com perfis contínuos de curvatura e torção. Inicialmente foi construída uma pequena trajetória bidimensional no plano de latitude e longitude da Terra, conforme é visto na Figura 6.11. A altitude foi mantida constante em cerca de 200 metros acima do nível do mar, e uma rajada de vento de cerca de 15 quilômetros por hora foi aplicada à uma orientação de 300 graus.

Ao observar a trajetória realizada pelo aeromodelo frente ao resultado planejado, é possível perceber a influência da perturbação causada pelo vento sobre o sistema de controle de navegação. A intensidade de vento aplicada, entretanto é relativamente baixa se comparada à algumas condições reais de vôo. Isso demonstra a necessidade de se elaborar algoritmos de controle que sejam mais robustos à esse tipo de ação externa.

Conclusão equivalente pode ser tirada a partir de Figura 6.12, onde foi utilizada uma trajetória tridimensional, gerada a partir das restrições calculadas anteriormente, e do conjunto de *waypoints* apresentado a seguir.

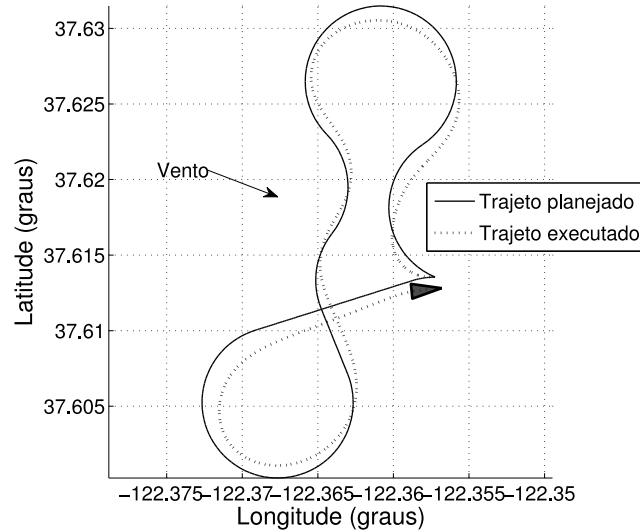


Figura 6.11: Controle de navegação da aeronave à altitude constante.

$$\begin{aligned}
 \mathbf{P}_j &= (-122.3572449, 37.61354309, 100, \frac{3\pi}{2}, 0), \\
 \mathbf{P}_k &= (-122.3811577, 37.62361430, 200, 0, 0), \\
 \mathbf{P}_l &= (-122.4022365, 37.59248146, 400, \frac{\pi}{2}, 0) \text{ e} \\
 \mathbf{P}_m &= (-122.3781880, 37.58263326, 500, \pi, 0).
 \end{aligned}$$

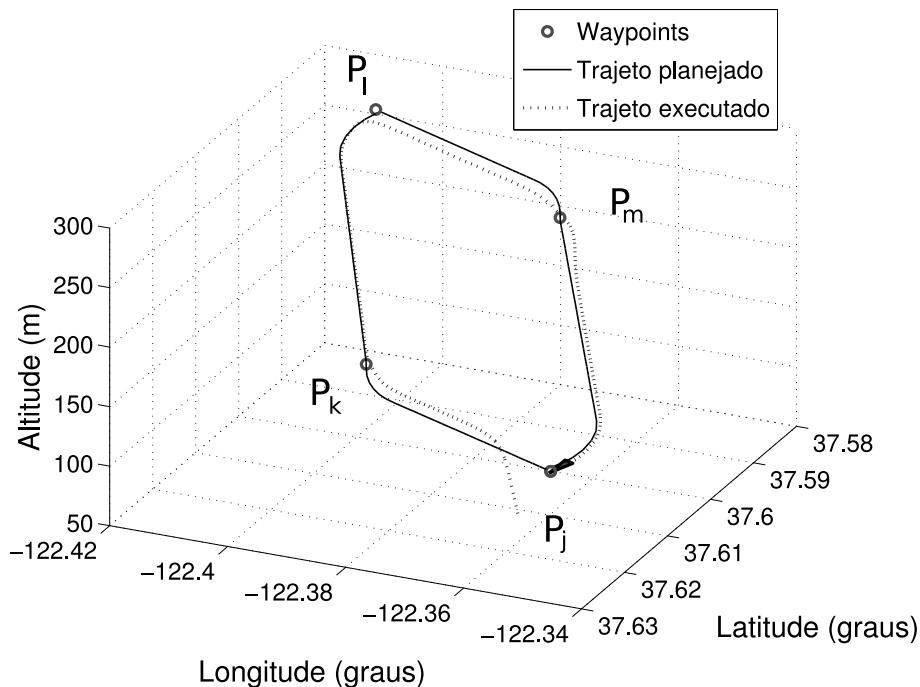


Figura 6.12: Teste de geração de trajetórias utilizando o *Dubins' Path 3D* no sistema *HWIL*.

6.3.2 Aeronave *AqVS*

Como no caso anterior, foram realizados testes para a identificação dos parâmetros restritivos da aeronave. A diferença nesse caso é que foram utilizados dois modelos separados. O modelo longitudinal foi empregado para se determinar o ângulo de subida do *AqVS*. Já o modelo látero-direcional foi utilizado para gerar estimativas do valor do raio mínimo de curvatura. O raio de torção, por sua vez, foi variado de forma arbitrária em diversos experimentos, até que uma trajetória factível fosse encontrada. Os resultados encontrados foram,

$$\begin{aligned}\rho_{min} &= 150 \text{ metros}, \\ \sigma_{min} &= 1000 \text{ metros}, \\ \theta_{max} &= \frac{\pi}{30} \text{ radianos.}\end{aligned}$$

Novamente, um conjunto arbitrário de cinco *waypoints* foi escolhido para uma missão qualquer. Foi gerada uma trajetória a partir da metodologia das curvas de Bézier de 7^a ordem. O resultado final pode ser visto na Figura 6.13.

$$\begin{aligned}\mathbf{P}_j &= (0, 0, 1000, 0, 0), \\ \mathbf{P}_k &= (2000, 0, 1010, 0, \frac{\pi}{40}), \\ \mathbf{P}_l &= (2000, 2000, 1020, 0, 0), \\ \mathbf{P}_m &= (2000, 0, 1010, 0, -\frac{\pi}{40}) \text{ e} \\ \mathbf{P}_n &= (0, 200, 1000, \pi, 0).\end{aligned}$$

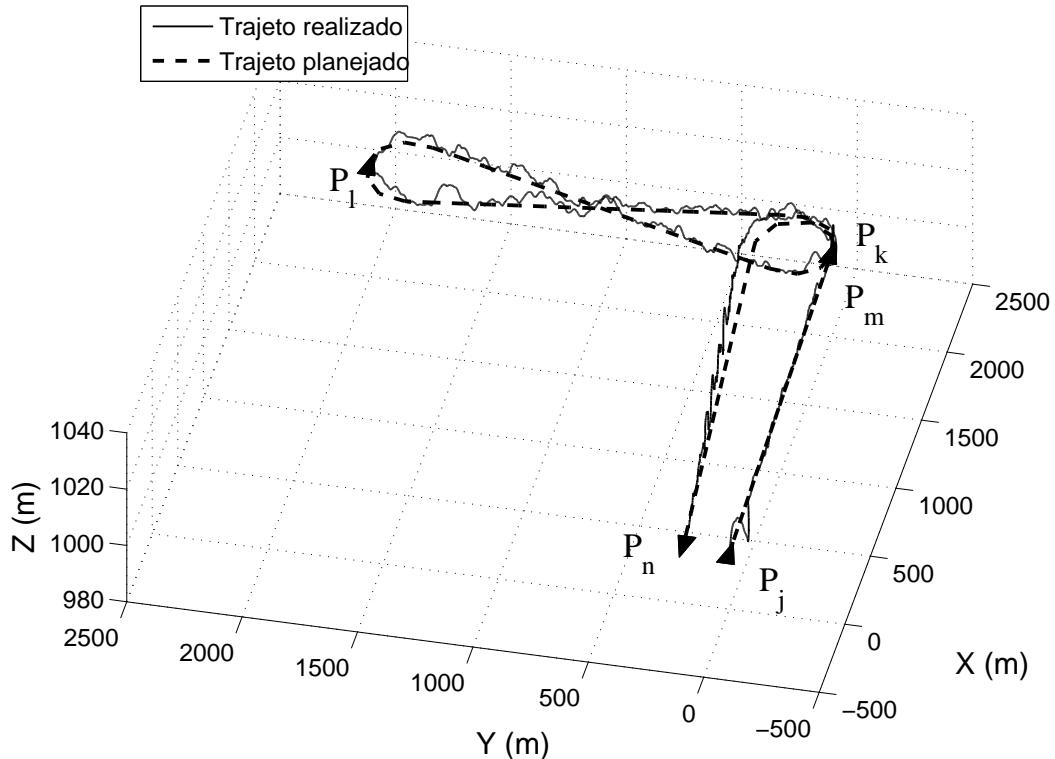


Figura 6.13: Trajetória projetada para o aeromodelo *AqVS*.

O resultado apresentado mostra que a trajetória calculada é realizável pelo modelo do VAANT utilizado. Um detalhe importante visto aqui são as oscilações apresentadas ao longo do eixo longitudinal de navegação do veículo, possivelmente causadas por um mal ajuste do controle de altitude da aeronave, ou pela simulação de ruído nos sensores. No próximo capítulo são apresentadas as principais conclusões retiradas deste trabalho, tal como os possíveis trabalhos futuros.

Capítulo 7

Conclusões

Neste capítulo são apresentadas as conclusões finais obtidas com este trabalho. Nessa etapa são sumarizadas as principais discussões levantadas ao longo de todo o texto. Por fim, são descritos também alguns dos possíveis caminhos (ou trajetórias) a serem explorados em trabalhos futuros.

7.1 Discussão dos Resultados

Neste trabalho apresentou-se um estudo sobre o planejamento de trajetórias tridimensionais para veículos aéreos. Esses veículos foram caracterizados por meio de três restrições cinemáticas: máxima curvatura (κ_{max}), máxima torção (τ_{max}) e máximo ângulo de subida (θ_{max}). Curvatura e torção são características que dependem das derivadas do vetor posição $\vec{r}(t)$ das aeronaves no espaço, sendo estudadas principalmente no ramo da Geometria Diferencial. Já a inclinação depende da relação entre a variação máxima possível de altitude e do comprimento da curva (calculado em função de x e y).

Foram apresentadas duas técnicas, uma relativa ao planejamento de caminhos (o *DP 3D*) e outra referente ao planejamento de trajetórias (curvas de Bézier de 7^a ordem). O *DP 3D* representa uma extensão do caminho bidimensional ótimo entre dois *waypoints* para o caso tridimensional. A principal desvantagem dessa metodologia é a descontinuidade nos perfis de curvatura e torção apresentados pela curva final.

Já a segunda metodologia é baseada no cálculo de curvas de Bézier de 7^a ordem, cujos princípios são semelhantes ao método do Hodográfico Pitagoreano tridimensional. A conclusão obtida a partir dos resultados é que essa técnica apresenta algumas vantagens significativas em relação as demais avaliadas. A principal delas é a continuidade nos perfis de curvatura e torção para o caso de junção de múltiplas curvas, conforme havia sido estipulado anteriormente.

Além disso, dependendo do conjunto de *waypoints* utilizados, as trajetórias geradas por essa metodologia se aproximam mais do *DP 3D*, em termos do comprimento, do que os resultados produzidos pelo *HP 3D*. Isso acontece, em especial, para conjunto com *waypoints* muito próximos, como aquele apresentado na etapa experimental. Nos demais casos, nenhuma das técnicas se sobressai nesse aspecto.

Outra vantagem, que acaba surgindo como consequência da anterior, é que o tempo gasto para a geração de trajetórias é bem menor nesse caso, do que no HP 3D. Além disso, a metodologia proposta converge para um resultado, sempre que for possível encontrar um caminho tridimensional de *Dubins*.

O *DP* 3D, por sua vez, apresenta uma convergência bastante rápida se comparada ao HP 3D, uma vez que é baseado no cálculo do caminho bidimensional ótimo. Cabe ainda relembrar que a técnica proposta neste trabalho para o cálculo do caminho tridimensional de *Dubins* é uma mera aproximação para o caminho 3D ótimo, o qual ainda não é conhecido, dadas as restrições abordadas neste contexto. Isso fica claro, por exemplo, no caso de falha do algoritmo, apontada na descrição do método.

7.2 Direções Futuras

Esse trabalho abre um grande número de possibilidades para o desenvolvimento de projetos futuros. De imediato, é possível pensar em implantar as técnicas de planejamento aqui propostas no *UAV AqVS* real. Espera-se que o comportamento apresentado seja bastante semelhante àquele experimentado em simulação.

De outro ponto de vista, há que se investigar mais a fundo a natureza da otimalidade do *DP* 3D proposto neste trabalho. Ainda representa um desafio para a matemática encontrar o caminho tridimensional ótimo, dadas as restrições cinemáticas consideradas aqui.

O planejamento de trajetórias em ambientes com obstáculos constitui outro problema bastante desafiador. O mesmo é válido para as metodologias que levam em consideração os erros existentes nos sensores dos veículos. Outro problema interessante seria investigar a relação entre as características dinâmicas de um veículo aéreo específico, a fim de gerar a melhor trajetória entre duas poses quaisquer no espaço. Ou ainda, expandir esse conceito para veículos com características cinemáticas e dinâmicas genéricas, com a incorporação de outros tipos de restrição, como por exemplo, a máxima velocidade de vôo. Esse é um parâmetros importante, conforme foi constatado na descrição da modelagem de veículos aéreos.

Outro desafio seria a expansão das técnicas de geração de trajetórias para conjuntos com múltiplos veículos aéreos com características heterogêneas. Já existem, por exemplo, técnicas capazes de gerar trajetórias para *swarms* de robôs aéreos, ou mesmo para a cooperação entre veículos aéreos e terrestres.

Um outra possibilidade ainda seria expandir as funcionalidades do sistema *HWIL* proposto. Esse sistema permite uma grande flexibilidade no que tange a questão da modelagem de veículos aéreos. No simulador de vôo utilizado, é possível utilizar os diversos modelos aerodinâmicos existentes para construir modelos de aeronaves reais das mais variadas categorias, inclusive para a simulação de múltiplos *UAVs* com características heterogêneas. Utilizando-se técnicas de modelagem e identificação de sistemas, é possível levantar os parâmetros e coeficientes aerodinâmicos de veículos para sua posterior representação matemática no ambiente de simulação. Não apenas as aeronaves de asa fixa, mas também helicópteros e dirigíveis virtuais podem ser facilmente criados dentro do simulador, desde que seus modelos sejam conhecidos.

Por fim, seria ainda possível o desenvolvimento de estratégias de controle alternativas e mais sofisticadas do que aquelas previamente programadas. Conforme pode ser visualizado, torna-se necessária à implementação de algoritmos de controle em mais baixo-nível que sejam menos susceptíveis a influências externas indesejáveis, como as rajadas de vento lateral, por exemplo. Em outras palavras, os controladores utilizados nas tarefas de posicionamento espacial devem apresentar maiores índices de rejeição a perturbações para melhorar o desempenho do sistema.

Referências Bibliográficas

- Ambrosino, G.; Ariola, M.; Ciniglio, U.; Corraro, F.; Pironti, A. e Virgilio, M. (2006). Algorithms for 3D UAV Path Generation and Tracking. In *Proceedings of the IEEE Conference on Decision and Control*, pp. 5275–5280, San Diego, CA, USA.
- Avibras (2008). Avibras aeroespacial s.a. <http://www.avibras.com.br/> (Acesso em Janeiro 2008).
- Barnhill, R. E. e Riesenfeld, R. F., editores (1974). *Computer Aided Geometric Design*. Academic Press, Inc., Orlando, FL, USA.
- Berndt, J. S. (2004). JSBSim: An Open Source Flight Dynamics Model in C++. In *Proceedings of AIAA Modeling and Simulation Technologies Conference and Exhibit*, Providence, RI. AIAA Paper 2004-4923.
- Bone, E. e Bolkcom, C. (2003). Unmanned Aerial Vehicles: Background and Issues for Congress. Technical Report RL31872, USA Congress.
- Bortoff, S. A. (2000). Path Planning for UAVs. In *Proceedings of the IEEE American Control Conference (ACC)*, pp. 364–368, Chicago, Illinois.
- Bowditch, N. (1995). *The American Practical Navigator - An Epitome of Navigation*. National Imagery and Mapping Agency, Bethesda, Maryland, 9 edição.
- Brandão, M. P.; Gomes, V. A.; d'Oliveira, F. A. e Bueno, C. A. E. (2007). UAV Activities in Brazil. In *First Latin-American Unmanned Aerial Vehicle (UAV) Conference*, Panama City.
- Camacho, L. e Yuhas, C. (2004). Civil UAV Capability Assessment. Technical report, NASA.
- Campos, M. F. M. e de Souza Coelho, L. (1999). Autonomous Dirigible Navigation Using Visual Tracking and Pose Estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pp. 2584–2589, Detroit, Michigan.
- Campos, M. F. M.; Iscold, P. H. A.; Tôrres, L. A. B. e Aguirre, L. A. (2007). SiDeVAAN: Simulação e Desenvolvimento de Veículos Aéreos Autônomos e Não-Tripulados. In *VIII Simpósio Brasileiro de Automação Inteligente*, Florianópolis, SC.

- Castillo-Effen, M.; Alvis, W. e Castillo, C. (2005). Modeling and Visualization of Multiple Autonomous Heterogeneous Vehicles. In *U.S. Navy Gran*, volume 3, pp. 2001– 2007.
- CEA (2008). Centro de Estudos Aeronáuticos. <http://www.demec.ufmg.br/Cea/> - acesso 12/02/2008.
- Craig, J. J. (1986). *Introduction to Robotics*. Addison Wesley.
- de Oliveira Neris, L. (2001). Um Piloto Automático para as Aeronaves do Projeto ARARA. Master's thesis, Laboratório de Computação de Alto Desempenho, Departamento de Ciência da Computação e Estatística, USP, São Carlos, SP.
- Doeblin, E. O. (2004). *Measurement Systems - Application and Design*. Mc Graw Hill, Singapore, 5 edição.
- Dubins, L. E. (1957). On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *American Journal of Mathematics*, 79:497–516.
- e Cavalcante, C. M. C. (1994). Sistemas de Navegação para Helicópteros Não-tripulados Utilizando Controlador Nebuloso. Master's thesis, Universidade Federal de Santa Catarina, Florianópolis, SC.
- Farouki, R. T. (1996). The Elastic Bending Energy of Pythagorean-Hodograph Curves. *Computer Aided Geometric Design*, 13:227–241.
- Farouki, R. T. e Han, C. Y. (2006). Algorithms for Spatial Pythagorean-Hodograph Curves. *Geometric Properties for Incomplete Data*.
- Farouki, R. T. e Neff, C. A. (1995). Hermite Interpolation by Pythagorean Hodograph Quintics. *Mathematics of Computation*, 64:1589–1609.
- Farouki, R. T. e Sakkalis, T. (1990). Pythagorean Hodographs. *IBM Journal of Research and Development*, 34:736–752.
- Gholkar, A.; Isaacs, A. e Arya, H. (2004). Hardware-In-Loop Simulator for Mini Aerial Vehicle. Technical Report 400076, Department of Aerospace Engineering, IIT Bombay, Powai, Mumbai, India.
- Gyron (1998). *Sistema HELIX: Histórico, Perguntas mais Freqüentes, Descritivo Técnico, Especificações Técnicas, Aplicações, Competência Tecnológica, Equipe, Artigos Publicados, Parcerias*. Gyron Sistemas Autônomos, Florianópolis, SC, Brasil.
- Göktogian, A. H.; Nettleton, E.; Ridley, M. e Sukkarieh, S. (2003). Real Time Multi-UAV Simulator. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pp. 2720– 2726, Taipei, Taiwan.

- Göktogian, A. H. e Sukkarieh, S. (2005). An Augmented Reality System for Multi-UAV Missions. In *Australian Centre for Field Robotics*, Sydney, Australia.
- Göktogian, A. H.; Sukkarieh, S.; Cole, D. e Thompsom, P. (2006). Airborne Vision Sensor Detection Performance Simulation. In *Australian Centre for Field Robotics*, Sydney, Australia.
- Hwangbo, M.; Kuffner, J. e Kanade, T. (2007). Efficient Two-phase 3D Motion Planning for Small Fixed-wing UAVs. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1035–1041, Roma, Italy.
- Iscold, P. (2007). Development of a Small Unmanned Aerial Vehicle for Aerial Reconaiscence. In *International Congress of Mobility Engineering*, São Paulo, SP.
- Jeyaraman, S.; Tsourdos, A.; Rabbath, C.-A. e Gagnon, E. (2004). Formalised Hybrid Control Scheme for a UAV Group using Dubins Set and Model Checking. In *Proceedings of the IEEE Conference on Decision and Control*, volume 4, pp. 4299–4304.
- Jia, D. e Vagners, J. (2004). Parallel Evolutionary Algorithms for UAV Path Planning. In *AIAA 1st Intelligent Systems Technical Conference*, Chicago, Illinois.
- Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research*, 5(1):90–98.
- King, E.; Kuwata, Y. e How, J. P. (2006). Experimental Demonstration of Coordinated Control for Multi-vehicle Teams. *International Journal of Systems Science*, 35(6).
- Komoriya, K. e Tanie, K. (1989). Trajectory Design and Control of a Wheel-type Mobile Robot Using B-spline Curve. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 398–405, Tsukuba, Japan.
- Kreyszig, E. (1991). *Differential Geometry*, volume 1. Dover Publications, New York.
- Latombe, J.-C. e Kavraki, L. E. (1998). Probabilistic Roadmaps for Robot Path Planning.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, Urbana, Illinois, U.S.A. Disponível em <http://planning.cs.uiuc.edu/>.
- Maeta, S. M. (2001). Desenvolvimento da Infra-estrutura Embarcada do Projeto AURORA. Master's thesis, Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP.
- Martins, A.; Bó, A. L.; Borges, G. e Ishihara, J. (2007). Design and Experimental Evaluation of Rotor Speed Regulators for Model Helicopters in a Test Bench. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–6, San Diego.

- Maček, K.; Petrovic, I. e Siegwart, R. (2005). A Control Method for Stable and Smooth Path Following of Mobile Robots. In *Proceedings of the 2nd European Conference on Mobile Robots - ECMR*, pp. 128–133, Ancona, Italy.
- McGeer, T. e Vagners, J. (1999). Wide-scale Use of Long-range Miniature Aerosondes Over the World's Oceans.
- McLean, D. (1990). *Automatic Flight Control Systems*. Prentice Hall, New York.
- Mirisola, L. G. B. (2001). Desenvolvimento da Estação de Terra do Projeto AURORA. Master's thesis, Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP.
- Olson, C. (2007). FlightGear Flight Simulator. <http://www.flightgear.org> (Acesso em 12 de Agosto de 2007).
- Ramos, J. J. G. (2002). *Contribuição ao Desenvolvimento de Dirigíveis Robóticos*. PhD thesis, Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, SC.
- Scheuer, A. e Fraichard, T. (1997). Continuous-Curvature Path Planning for Car-Like Vehicles. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pp. 997–1003, Grenoble, France.
- Sederberg, T. W. (2007). *Computer Aided Geometric Design*, chapter 2, pp. 17–36. Brigham Young University.
- Shanmugavel, M. (2007). *Path Planning of Multiple Autonomous Vehicles*. PhD thesis, Defence College of Management and Technology, Cranfield University, Shrivenham.
- Shanmugavel, M.; Tsourdos, A.; Zbikowski, R. e White, B. A. (2005). Path Planning of Multiple Uavs using Dubins Sets. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit (GN&C)*, San Francisco, California.
- Shanmugavel, M.; Tsourdos, A.; Zbikowski, R. e White, B. A. (2007). 3D Path Planning for Multiple UAVs using Pythagorean Hodograph Curves. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit (GN&C)*, Hilton Head, South Carolina.
- Shanmugavel, M.; Tsourdos, A.; Zbikowski, R.; White, B. A.; Rabbath, C. A. e Léchevin, N. (2006). A Solution to Simultaneous Arrival of Multiple UAVs using Pythagorean Hodograph Curves. In *Proceedings of the IEEE American Control Conference (ACC)*, pp. 2813–2818, Minneapolis, Minnesota, USA.
- Shkel, A. M. e Lumelsky, V. (2001). Classification of the Dubins Set. In *Robotics and Autonomous Systems*, volume 34, pp. 179–202.

- Siegwart, R. e Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*. MIT Press, Cambridge, Massachusetts, London, England.
- Sorton, E. F. e Hammaker, S. (2005). Simulated Flight Testing of an Autonomous Unmanned Aerial Vehicle Using FlightGear. In *American Institute of Aeronautics and Astronautics*, Fairmont, West Virginia. Institute for Scientific Research, Inc.
- Stevens, B. L. e Lewis, F. (1992). *Aircraft Control and Simulation*. John Wiley & Sons, New York.
- Valavanis, K. P., editor (2007). *Advances in Unmanned Aerial Vehicles: State of the Art and Road to Autonomy*. Springer, Tampa, Florida, USA.
- Wong, K. C. (2006). UAV Design Activities in a University Environment. Technical report, School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, Sydney, Australia.
- Ästrom, K. J. e Hagglund, T. (2004). Revisiting the Ziegler-Nichols Step Response Method for PID Control. *Journal of Process Control*, 14:635–650.

