

UNIVERSIDADE DA BEIRA INTERIOR



UBI
Covilhã
Portugal

Optimização e Controlo de Trajectórias 4D Definidas por Waypoints

Elaborado por

Paulo Filipe Faria Machado

Orientado por

Professor Dr. Kouamana Bousson

Dissertação de Mestrado submetida à Universidade da Beira Interior
para obtenção do grau de Mestre

na

Unidade de Ciências da Engenharia
Departamento de Ciências Aeroespaciais

Junho de 2008

“Aqueles que por obras valerosas se vão da lei da morte libertando.”

Luís Vaz de Camões

UNIVERSIDADE DA BEIRA INTERIOR

Resumo

Unidade de Ciências da Engenharia
Departamento de Ciências Aeroespaciais

Dissertação de Mestrado

por Paulo Filipe Faria Machado

O problema da optimização e controlo de trajectórias 4D é uma temática ainda pouco abordada. A navegação 4D consiste em definir *waypoints*, para uma dada missão, com o tempo de chegada especificado em cada um deles. A única possibilidade de projectar trajectórias capazes de executar este tipo de missão é recorrer ao controlo óptimo, também designado por optimização de trajectória. Para resolver o problema de optimização de trajectória recorre-se à aplicação dos métodos indirectos e dos métodos directos. Os métodos indirectos assentam no princípio do máximo de Pontryagin, enquanto que nos métodos directos é necessário transcrever o problema para um problema de programação não linear. Em problemas complexos, que é o caso do projecto de trajectórias 4D, são normalmente usados os métodos directos, pelo que, estes precisam de esquemas de integração para discretizar as equações dinâmicas do problema. A maior parte dos métodos de integração derivam dos esquemas de Runge-Kutta, e dum método bastante popular é o método da Colocação. Actualmente os métodos pseudoespectrais começam a ser a melhor opção para resolver alguns dos problemas mais complexos de optimização de trajectória. O método pseudoespectral utilizando polinómios de Chebyshev consegue solucionar o problema de optimização de trajectórias 4D, conseguindo resolver duas missões propostas. É proposto uma abordagem de um método de controlo predictivo que permite controlar um veículo aeroespacial através de uma trajectória de referência, este tipo de controlo denomina-se por controlo predictivo de passo único.

Agradecimentos

Presto o mais sincero agradecimento ao Professor Doutor Kouamana Bousson, pois sem a sua orientação, sem o seu apoio e sem a sua dedicação este trabalho não era possível.

Conteúdo

Resumo	ii
Agradecimentos	iii
Lista de Figuras	vi
Lista de Tabelas	ix
Nomenclatura	x
1 Estado da Arte	1
1.1 Introdução	1
1.2 Posição do Problema	2
1.3 Trabalhos Existentes	3
2 Revisão da Optimização	6
2.1 Revisão: O que é a Optimização?	6
2.2 Modelo Matemático da Optimização	6
2.3 Definições	7
2.4 Programação Não Linear	7
2.4.1 Métodos Iterativos	7
2.4.1.1 Existência do Mínimo Global	8
2.4.1.2 Métodos Iterativos Descendentes	8
2.4.1.3 Método do Gradiente	8
2.4.2 Optimização sem Restrições	9
2.4.2.1 Escolha da Direcção de Descida	10
2.4.2.2 Escolha do Passo de Descida	12
2.4.2.3 Critério de Paragem	13
2.4.3 Optimização com Restrições	13
2.4.3.1 Restrições de Igualdade	13
2.4.3.2 Restrições de Desigualdade	14
2.5 Aplicação	15
3 Optimização de Trajectória	19
3.1 Formulação do Problema de Optimização de Trajectória	19

3.1.1	Tipos de Problemas	20
3.1.1.1	Problema de Mayer	21
3.1.1.2	Problema de Lagrange	21
3.1.1.3	Problema de Bolza	21
3.1.1.4	Tranformação do Problema de Bolza num Problema de Mayer	21
3.2	Métodos de Procura de Trajectórias Óptimas	22
3.2.1	Métodos Indirectos	22
3.2.1.1	Descretização do Método	24
3.2.2	Métodos Directos	25
3.2.2.1	Colocação	26
3.2.2.2	Métodos de Runge Kutta	28
3.2.2.3	Pseudoespectral	29
3.3	Aplicação	32
3.3.1	Descrição do Problema de Bryson-Ho	32
3.3.2	Solução Numérica para o Problema de Optmização de Trajectória	33
4	Problema de Navegação e Controlo 4D	43
4.1	Modelação Geral do Problema de Navegação 4D	44
4.1.1	Formulação do Problema	44
4.1.2	Modelo Dinâmico	45
4.2	Resolução do Problema de Navegação 4D	46
4.2.1	Trajectórias Óptimas	48
4.2.1.1	Trajectórias Óptimas Segundo Métodos Directos	48
4.3	Controlo de Trajectórias de Referência	51
4.3.1	Controlo Preditivo	51
4.3.1.1	Teoria do Controlo Predictivo de passo Único	51
4.3.1.2	Controlo de Trajectórias 4D	54
4.4	Aplicação	56
4.4.1	<i>Exemplo I</i>	56
4.4.1.1	Plano de Voo	56
4.4.1.2	Trajectória Óptima	57
4.4.2	<i>Exemplo II</i>	66
4.4.2.1	Plano de Voo	66
4.4.2.2	Trajectória Óptima	66
4.4.2.3	Controlo de Trajectória de Referência	74
5	Conclusão	83
A	Dados do SkyGu@rdian	87
	Bibliografia	88

Lista de Figuras

1.1	Navegação por Waypoints	2
2.1	Representação gráfica dos valores da tabela (à esquerda) e da função de Hardy (à direita)	17
3.1	Distância Radial vs Tempo de Simulação para $N = 10$ nós	36
3.2	Velocidade Radial vs Tempo de Simulação para $N = 10$ nós	37
3.3	Velocidade Tangencial vs Tempo de Simulação para $N = 10$ nós	37
3.4	Ângulo do Sistema propulsivo vs Tempo de Simulação para $N = 10$ nós	38
3.5	Distância Radial vs Tempo de Simulação para $N = 30$ nós	38
3.6	Velocidade Radial vs Tempo de Simulação para $N = 30$ nós	39
3.7	Velocidade Tangencial vs Tempo de Simulação para $N = 30$ nós	39
3.8	Ângulo do Sistema propulsivo vs Tempo de Simulação para $N = 30$ nós	40
3.9	Distância Radial vs Tempo de Simulação para $N = 50$ nós	40
3.10	Velocidade Radial vs Tempo de Simulação para $N = 50$ nós	41
3.11	Velocidade Tangencial vs Tempo de Simulação para $N = 50$ nós	41
3.12	Ângulo do Sistema propulsivo vs Tempo de Simulação para $N = 50$ nós	42
4.1	Comportamento da Longitude ao longo do Tempo	60

4.2	Comportamento da Latitude ao longo do Tempo	60
4.3	Comportamento da Altitude ao longo do Tempo	61
4.4	Projecção no terreno através <i>Google Hearth</i> da Trajectória obtida pelo método da Colocação	61
4.5	Projecção no terreno através <i>Google Hearthd</i> a Trajectória obtida pelo método Pseudoespectral	62
4.6	Projecção no terreno através <i>Google Hearth</i> da Trajectória obtida pelo método da Colocação (a branco) e Pseudoespectral (a vermelho)	62
4.7	Comportamento da Velocidade ao longo do Tempo	63
4.8	Comportamento da Ângulo de Trajectória ao longo do Tempo	63
4.9	Comportamento do Rumo ao longo do Tempo	64
4.10	Comportamento do controlo u_1 ao longo do Tempo	64
4.11	Comportamento do controlo u_2 ao longo do Tempo	65
4.12	Comportamento do controlo u_3 ao longo do Tempo	65
4.13	Comportamento da Longitude ao longo do Tempo	68
4.14	Comportamento da Latitude ao longo do Tempo	69
4.15	Comportamento da Altitude ao longo do Tempo	69
4.16	Trajectória 3D	70
4.17	Projecção no terreno através <i>GoogleHearthd</i> da Trajectória obtida pelo método da Pseudoespectral	70
4.18	Comportamento da Velocidade ao longo do Tempo	71
4.19	Comportamento da Ângulo de Trajectória ao longo do Tempo	71
4.20	Comportamento do Rumo ao longo do Tempo	72
4.21	Comportamento do controlo u_1 ao longo do Tempo	72

4.22	Comportamento do controlo u_2 ao longo do Tempo	73
4.23	Comportamento do controlo u_3 ao longo do Tempo	73
4.24	Comportamento da Longitude na trajectória nominal e na controlada . . .	75
4.25	Comportamento da Latitude na trajectória nominal e na controlada . . .	76
4.26	Comportamento da Altitude na trajectória nominal e na controlada . . .	76
4.27	Trajectória 3D nominal e controlada	77
4.28	Projecção no terreno através <i>GoogleHearth</i> da Trajectória de referência (branco) e da Trajectória realizada (azul)	78
4.29	Comportamento da Velocidade na trajectória nominal e na controlada . . .	79
4.30	Comportamento do Ângulo de Trajectória na trajectória nominal e na controlada	79
4.31	Comportamento do Rumo na trajectória nominal e na controlada	80
4.32	Comportamento do pranchamento durante a trajectória	80
4.33	Comportamento da argagem durante a trajectória	81
4.34	Comportamento da taxa de rolamento durante a trajectória	81
4.35	Comportamento da taxa de arfagem durante a trajectória	82
4.36	Comportamento da taxa de guinada durante a trajectória	82

Lista de Tabelas

2.1	Tabela de Tracção	18
3.1	Dados normalizados do problema	34
3.2	Valores das variáveis de estado no final da transferência de órbita	36
4.1	Coordenadas dos Waypoints entre Castelo Branco e Covilhã	57
4.2	Condições de fronteira do vector de estado e de controlo	58
4.3	Valores das variáveis de estado no final da trajectória	59
4.4	Coordenadas dos Waypoints para uma missão em circuito	66
4.5	Valores das variáveis de estado no final da trajectória	67
4.6	Erro RMSE entre a trajectória nominal e a efectuada pelo controlador	75

Nomenclatura

λ	Longitude
φ	Latitude
R_e	Raio da Terra
h	Altitude em relação ao nível médio das águas do mar
τ	Tempo de chegada aos waypoints
V	Velocidade
γ	Ângulo de Trajectória
ψ	Ângulo de Rumo
ω	Velocidade angular da Terra
T	Tracção
L	Sustentação
D	Arrasto
C_L	Coefficiente de Sustentação
C_D	Coefficiente de Arrasto
C_Y	Coefficiente de Força Lateral
α	Ângulo de Ataque
β	Ângulo de Derrapagem
θ	Ângulo de Argagem
ϕ	Ângulo de Rolamento
p	Taxa de Rolamento
q	Taxa de Arfagem
r	Taxa de Guinada
g	Aceleração da gravidade
ρ	Massa específica
S	Área Alar
m	Massa do corpo
I_x	Momento de Inércia em relação x
I_y	Momento de Inércia em relação y
I_z	Momento de Inércia em relação z
I_{xz}	Momento de Inércia em relação xz

<i>C_m</i>	Coeficiente de Momento de Arfagem
<i>C_l</i>	Coeficiente de Momento de Rolamento
<i>C_n</i>	Coeficiente de Momento de Guinada

*Dedico calorosamente esta Dissertação, aos meus queridos pais
António Machado e Ilza Machado e ao meu mestre e amigo Sensei
Profírio Isidoro que me ensinou aquilo que eu mais gosto*

Capítulo 1

Estado da Arte

1.1 Introdução

O tema deste trabalho é encontrar um método, que definindo uma determinada missão através de *waypoints* (com localização e tempo de chegada ao *waypoint*), se consiga gerar uma trajectória óptima que passe por todos os *waypoints* minimizando os atrasos de chegada a cada um deles.

Na realidade não existe estudos concretos sobre este tema, existe apenas um estudo que tenta aproximar-se sobre esta temática [10] mas não resolve o problema 4D, resolvendo sim o problema 3D e para além do mais utiliza restrições dinâmicas da aeronave para resolver o problema de navegação. O que é feito por certas entidades que trabalham sobre optimização de trajectória para problemas de navegação, é pegar numa determinada missão definida por *waypoints* e dividi-la nos vários segmentos (trajecto entre *waypoints*), tratando cada segmento como uma trajectória única.

No Capítulo 3 irá ser compreendido o problema de optimização de trajectória, irão ser descritos os métodos que resolvem o problema de optimização de trajectória, métodos directos e métodos indirectos. Já no Capítulo 4 irá ser aplicado a optimização de trajectória recorrendo unicamente aos métodos directos para resolver o problema de navegação 4D, ainda durante este capítulo irá ser aplicado um tipo de controlo predictivo que resolve o problema do controlo segundo uma trajectória de referência.

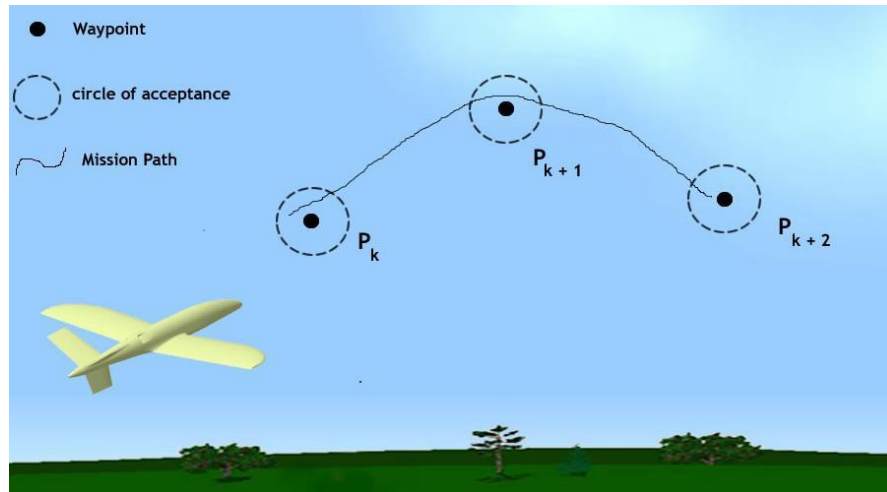


FIGURA 1.1: Navegação por Waypoints

1.2 Posição do Problema

A orientação para a Navegação consiste em fornecer a Velocidade V_{ref} o rumo ψ_{ref} e o ângulo de trajetória γ_{ref} para gerar uma trajetória de voo que passe através de uma sequência pré-definida de *waypoints* $P_0, P_1, \dots, P_k, \dots, P_N$ (figura (1.1)), com a possibilidade de $P_N = P_0$. Os *waypoints* são colocados numa base de dados pelo operador.

Para situações aplicáveis à vigilância, o veículo pode ser guiado através de uma trajetória de referência que pode ser um *loiter* ou qualquer outra trajetória de referência contínua.

Na navegação por *waypoints* pode acontecer que a aeronave não se consiga passar através de um *waypoint* vizinho devido às restrições dinâmicas e estruturais, por exemplo, saturação dos controlos, limite de rumo ou razão de subida.

Cada *waypoint* P_k é descrito numa base de dados, tendo para cada k , um vector de quatro dimensões:

$$P_k = (\lambda_k \varphi_k h_k \tau_k)^T \quad (1.1)$$

Com,

- λ_k : longitude, em radianos
- φ_k : latitude, em radianos
- h_k : altitude em relação ao nível do mar, em metros
- τ_k : tempo de chegada ao waypoint, em segundos

Esta é a formulação geral da navegação definida por *waypoints4D*, ou seja, é imposto um tempo de chegada a cada *waypoint* com vista a minimizar, por exemplo, o consumo de combustível.

O problema do atraso nos vários waypoints tem de ser tido em conta, pois é mais que provável, que as condições atmosféricas influenciem tempo de voo entre *waypoints*. Outro problema não menos importante, é o de minimizar os desvios de trajectória em relação à rota ortodrómica.

Pretende-se então optimizar este problema de modo a que as situações acima referidas sejam ultrapassadas.

1.3 Trabalhos Existentes

Um dos objectivos de um veículo aéreo não tripulado, por exemplo, é alcançar o seu destino final sem que colida com qualquer obstáculo. Já para um míssil cruzeiro, é necessário orientar este em direcção ao alvo passando por várias áreas sem que o sistema defensivo terrestre o detecte. Para conseguir estes objectivos convém que seja gerada uma boa trajectória nominal, que pode ser projectada por vários métodos. Um desses métodos é o projecto de trajectórias para *waypoints* pré-definidos de acordo com a missão.

As leis de orientação são frequentemente usadas para gerar trajectórias que passam através de waypoints específicos.

Actualmente distinguem-se basicamente dois tipos de trabalhos relativos à optimização de trajectórias definidas por *waypoints*,

- i) Trabalhos que utilizam apenas as equações de navegação [9, 12] em que as trajectórias são projectadas sem terem em consideração limites estruturais e dinâmicos tais como limites de manuverabilidade, saturação dos comandos, raio de volta mínima a uma determinada velocidade.
- ii) Trabalhos que para além de utilizarem as equações de navegação entram com as equações da dinâmica do voo [11].

De uma maneira geral, o modelo das equações de navegação assenta na seguinte estrutura:

$$\dot{\lambda} = \frac{V \cos \gamma \sin \psi}{(Re + h) \cos \varphi} \quad (1.2)$$

$$\dot{\varphi} = \frac{V \cos \gamma \sin \psi}{Re + h} \quad (1.3)$$

$$\dot{h} = V \sin \gamma \quad (1.4)$$

onde

Nos trabalhos em que se utiliza apenas as equações de navegação, constata-se que o modelo de orientação não é necessariamente viável para determinadas aeronaves. Utilizam-se modelos 3D em que o vector de estado é dado pela velocidade V , pelo ângulo de trajectória γ e pelo ângulo de rumo ψ [9]. Modelos que podem também ser faseados em dois modelos 2D, um no plano horizontal e outro no plano vertical, cujo modelo do plano horizontal é utilizado para definir a trajectória requerida (por exemplo uma trajectória de reconhecimento), utilizando-se o modelo do plano vertical para especificação do relevo do terreno ou da altura da trajectória [12]. Este tipo de abordagem tem dificuldades em encontrar trajectórias óptimas entre *waypoints*.

Por outro lado um método óptimo de geração de trajectórias 3D foi proposto para a passagem de *waypoints* específicos [11]. Neste caso para além das equações acima utilizadas recorreu-se também às equações da dinâmica de voo:

$$\dot{V} = \frac{1}{m} F_T - g \sin \gamma + \omega^2 R \cos \varphi (\sin \gamma \cos \varphi - \cos \gamma \sin \psi \sin \varphi) \quad (1.5)$$

$$\begin{aligned} V \dot{\gamma} = & \frac{1}{m} F_N \cos \phi - g \cos \gamma + \frac{V^2}{R} \cos \gamma + 2\omega V \cos \psi \cos \varphi \\ & + \omega^2 R \cos \varphi (\cos \gamma \cos \varphi + \sin \gamma \sin \psi \sin \varphi) \end{aligned} \quad (1.6)$$

$$\begin{aligned} V \dot{\psi} = & \frac{1}{m} \frac{F_N \sin \phi}{\cos \gamma} - \frac{V^2}{R} \cos \gamma \cos \psi \tan \varphi \\ & + 2\omega V (\tan \gamma \sin \psi \cos \varphi - \sin \varphi) - \frac{\omega^2}{\cos \gamma} \cos \psi \sin \varphi \cos \varphi \end{aligned} \quad (1.7)$$

$$F_T = T \cos \alpha - D \quad (1.8)$$

$$F_N = T \sin \alpha + L \quad (1.9)$$

$$D = \frac{1}{2}\rho V^2 C_D S \quad (1.10)$$

$$L = \frac{1}{2}\rho V^2 C_L S \quad (1.11)$$

$$C_L = C_{L_\alpha} \alpha \quad (1.12)$$

O problema de optimização foi formulado,

$$\min J = \int_{t_0}^{t_f} \|u(t)\|_R^2 dt \quad (1.13)$$

sujeito às equações dinâmicas,

$$\dot{x} = f(x, u, t) \quad (1.14)$$

e resolvido por uma técnica directa de optimização.

Devido ao facto de o tempo de passagem por cada waypoint não ser especificado, surge então, o aparecimento de vários tempos intermédios inespecíficos. Introduzindo variáveis auxiliares para tempos inespecíficos, através da normalização da variável tempo, o problema foi reformulado num problema de optimização convencional com tempo fixo. Este método é limitado ainda por ser necessário encontrar uma estimativa inicial óptima da trajectória para que o problema convirja e tenha uma precisão aceitável, e por outro lado o modelo proposto para o guidance é bastante complexo exigindo bastante tempo de computação. Em todos estes trabalhos existem pontos que os autores desprezam, para que o problema se torne mais fácil de resolver, tais como o tempo de sobrevoo/chegada nos *waypoints*, a imposição de um limite de tempo voo entre waypoints.

O estudo sobre optimização e controlo de trajectórias tem sido vasto durante os últimos anos, apesar disso, os modelos que os autores utilizam são bidimensionais ou tridimensionais, não correspondendo perfeitamente à realidade. Não se realizaram ainda estudos sobre trajectórias 4D. A vantagem deste tipo de trajectória é que impondo um tempo de passagem pelos *waypoints* é possível constantemente otimizar a trajectória de forma a cumprir os requisitos propostos. Este novo método proposto é o objectivo deste trabalho.

Capítulo 2

Revisão da Optimização

Na actualidade e de há uns anos a esta parte, tem-se abordado e desenvolvido inúmeros métodos de optimização de trajectória ou controlo óptimo paralelamente ao desenvolvimento espacial.

Irá ser feita uma breve revisão de algumas noções e conceitos que irão ser utilizados no processo de optimização do problema proposto.

2.1 Revisão: O que é a Optimização?

Todas as situações no dia a dia que requerem fazer uma escolha de um objectivo em mente, envolvem alguma optimização. De uma maneira geral, pode-se afirmar que optimização é o processo de fazer a melhor escolha baseada em algumas possibilidades.

Para entender os conceitos de uma maneira mais formal, é preciso olhar para o modo de definir optimização matematicamente.

2.2 Modelo Matemático da Optimização

Comece-se então, por definir restrição. Seja X o conjunto de restrições que podem ser definidas num espaço finito (por exemplo: $X \subseteq \mathbb{R}^n$ ou $X \subseteq \{0, 1\}$) ou num espaço infinito (por exemplo: $X \subseteq H$ em que \mathbf{H} é o espaço de Hilbert).

Posto isto, é também necessário definir **função de custo** como sendo o mapa do conjunto de restrições de X para \mathbb{R} , onde f é um escalar. A notação para a **função de custo** é:

$$f : X \rightarrow \mathbb{R} \tag{2.1}$$

Por fim diz-se que o objectivo da optimização é encontrar $x^* \in X$ tal que $f(x^*) \leq f(x), \forall x \in X$. Com outra notação:

$$\min(f(x)), \forall x \in X \quad (2.2)$$

2.3 Definições

Para que se melhor entenda estes problemas, considere as seguintes definições:

Definição 1. Diz-se que o conjunto $X \subseteq \mathfrak{R}^n$ é convexo se para $\forall x, y \in X$ e $\lambda \in [0, 1]$, o vector $\lambda x + (1 - \lambda)y \in X$.

Definição 2. Diz-se que a função $f : \mathfrak{R} \rightarrow \mathfrak{R}$ é convexa se o *epígrafo* $\text{epi}(f) = \{(x, y) \in \mathfrak{R}^{n+1} | f(x) \leq y\}$.

O *epígrafo* de uma função, é conjunto de pontos que estão acima dessa função.

Definição 3. x^* é um mínimo local se a desigualdade $f(x^*) \leq f(x), \forall x \in X$ tal que $\|x - x^*\| < \epsilon$.

x^* é um mínimo estritamente local se $f(x^*) \leq f(x), \forall x \in X$ com $x \neq x^*$ tal que $\|x - x^*\| < \epsilon$. Note-se que o mínimo global é o mais pequeno dos mínimos locais

Definição 4. O ponto $x^* \in X$ é um mínimo global (óptimo) se $f(x^*) \leq f(x), \forall x \in X$. x^* é um mínimo estritamente global se $f(x^*) \leq f(x), \forall x \in X$ com $x \neq x^*$

2.4 Programação Não Linear

Nesta secção vai ser explicado de forma geral a metodologia da programação não linear. Esta é apenas um conjunto de métodos que pretendem descobrir qual o mínimo ou máximo, local ou se possível global, de uma determinada função $f(x)$ que seja não linear. Por isso programação não linear e optimização não linear são conceitos iguais.

2.4.1 Métodos Iterativos

Os métodos para a optimização de funções de várias variáveis podem ser agrupados em duas classes, métodos de procura que utilizam apenas valores da função, comparando-os, para progredir em direcção ao minimizante e métodos do gradiente que utilizam não só valores de $f(x)$, mas também informação relativa às derivadas, na forma do vector gradiente e/ou da matriz Hessiana.

2.4.1.1 Existência do Mínimo Global

Considere-se o problema $\min_{x \in X} f(x)$, $x \in \mathfrak{R}^n$, então existem duas possibilidades:

- 1) Se o conjunto $\{f(x)|x \in X\}$ não possuir fronteiras inferiores, então não existe o uma solução óptima para o problema.
- 2) Se o conjunto $\{f(x)|x \in X\}$ possuir delimitação inferior, nesta situação existem dois casos para provar a que solução é óptima.

Teorema 1. (Teorema de Weierstrass') O mínimo de f num domínio X existe se f é contínua.(i.e. X é compacto em \mathfrak{R}^n):

Teorema 2. O mínimo global existe se f é contínuo e coercivo, e X é fechado. (i.e. $f(x^k) \rightarrow \infty, \forall \|x^k\| \rightarrow \infty$).

2.4.1.2 Métodos Iterativos Descendentes

Considere-se a minimização de uma função sem restrições diferenciável e contínua $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$. Supondo que f é delimitada inferiormente, $f(x) \geq L, \forall x \in \mathfrak{R}^n$. Os métodos iterativos descendentes, de maneira geral, começam com um ponto arbitrário e em seguida geram novos pontos tal que f esteja a decrescer, isto é:

- i) Arbitra-se um ponto $x^0 \in \mathfrak{R}^n$.
- ii) Gera-se sucessivamente x^1, x^2, \dots, x^k tal que $f(x^{k+1}) < f(x^k)$.

2.4.1.3 Método do Gradiente

O método do gradiente é um dos métodos mais utilizados para optimização. Considere-se por isso uma função $f : X \rightarrow \mathfrak{R}$ e que f é da classe C^1 em X . Quer-se então resolver o seguinte problema:

$$\min_{x \in X} f(x) \tag{2.3}$$

Gradiente de uma função

Gradiente de uma função $f(x)$, $x \in \mathfrak{R}^n$ é o vector definido por:

$$\nabla f(x) = \left[\frac{\delta f(x)}{\delta x_1} \quad \frac{\delta f(x)}{\delta x_2} \quad \dots \quad \frac{\delta f(x)}{\delta x_n} \right]^T \tag{2.4}$$

O gradiente de uma função $f(x)$ em $\forall x$ é, a direcção de subida máxima quando se desloca ao longo de f a partir de x .

Um deslocamento sobre f a partir de x na direcção $\nabla f(x)$ resulta numa maior subida do que em qualquer outra direcção com um percurso do mesmo comprimento. Do mesmo modo, um percurso na direcção oposta ($-\nabla f(x)$) resulta na descida máxima, assim sendo, seguindo a direcção oposta ao gradiente no percurso sobre a função f acaba-se por encontrar o mínimo de f .

O método do gradiente é um método iterativo de procura que parte de um ponto inicial $x^0 \in X$. Para que se atinja o mínimo de f a sequência de pontos $x^k (k = 0, 1, \dots)$ é gerada tal que:

$$x^{k+1} = x^k + \alpha^k d^k \quad (2.5)$$

Onde, para cada passo k , α^k e d^k são designados por *passo de descida* e *direcção de descida* respectivamente. Implicando $\nabla f(x^k)^T d^k < 0$, para que se garanta a descida. O caso básico é quando $d^k = -\nabla f(x)$. O processo iterativo acaba quando $\|d^k\| \leq Tol$, seja Tol um valor próximo de zero.

Uma das vantagens do método do gradiente e de suas variantes é que convergem rapidamente. No entanto este método requer a derivabilidade da função a ser minimizada e o cálculo do gradiente (composto por derivadas parciais) desta função em cada iteração. O método do gradiente é recomendado nas seguintes condições:

- i) A dimensão do problema (isto é o número de variáveis monodimensionais envolvidas no problema) é relativamente pequena.
- ii) A expressão da função f não é muito complexa, isto é, a avaliação de f não demora muito.
- iii) A função f é derivável e as derivadas parciais podem ser calculadas facilmente.
- iv) Pretende-se achar apenas um minimizador local da função f .

2.4.2 Optimização sem Restrições

Considere-se neste tópico um problema de optimização que não está sujeito a restrições. Suponha-se que são escolhidas n variáveis x para minimizar o escalar *função de custo* ou *função objectiva* f . As condições necessárias para que x^* seja um ponto satisfatório são:

$$c(x^*) = \nabla_x f(x^*) = \begin{bmatrix} \frac{\delta f(x^*)}{\delta x_1} \\ \frac{\delta f(x^*)}{\delta x_2} \\ \vdots \\ \frac{\delta f(x^*)}{\delta x_n} \end{bmatrix} = 0 \quad (2.6)$$

Se for usado o método de Newton para encontrar o ponto onde o gradiente é zero, deve-se computar a direcção de descida usando:

$$H(x)d = -c(x) \quad (2.7)$$

onde $H(x)$ é a matriz *Hessiana* simétrica das segundas derivadas da função de custo. Apesar disto existem ainda alguns obstáculos à utilização deste procedimento.

O primeiro é que $c(x) = 0$ é uma condição necessária mas não suficiente, porque o ponto de gradiente nulo tanto pode ser um máximo ou um mínimo. No ponto mínimo a matriz *Hessiana* é positivamente definida, mas pode não o ser para um ponto longe da solução.

O segundo, é que existe alguma ambiguidade na escolha da função de mérito, se a direcção de descida é usada para estabilizar método.

2.4.2.1 Escolha da Direcção de Descida

Método de Newton

O método de Newton não é nada mais que a aplicação directa da condição $\nabla^2 f(x^k)d^k = -\nabla f(x^k)$, então:

$$d^k = (\nabla^2 f(x^k))^{-1}(-\nabla f(x^k)) \quad (2.8)$$

$$x^{k+1} = x^k + \alpha^k (\nabla^2 f(x^k))^{-1}(-\nabla f(x^k)) \quad (2.9)$$

assumindo que existe inversa.

Método do tipo Quasi-Newton

O método de Newton precisa do cálculo das segundas derivadas na forma da matriz Hessiana. Estas derivadas podem ser difíceis de calcular quando $f(x)$ tem uma forma

complicada. Assim os seguintes métodos têm como característica estimar a matriz Hessiana inversa G e determinar a direcção de descida, através de,

$$d^k = -G^k \nabla f(x^k) \quad (2.10)$$

D.E.P(Davidon-Fletcher-Powell)

$$G_{D.F.P}^{k+1} = G^k - \frac{G^k q^k q^{kT} G^k}{q^{kT} G^k q^k} + \frac{s^k s^{kT}}{s^{kT} q^k} \quad (2.11)$$

D(Davidon)

$$G_D^{k+1} = G^k + \frac{(s^k - G^k q^k)(s^k - G^k q^k)^T}{s^{kT} q^k - q^{kT} G^k q^k} \quad (2.12)$$

B.F.G.S(Broyden-Fletcher-Goldfarb-Shanno)

$$G_{B.F.G.S}^{k+1} = \left(I - \frac{s^k q^{kT}}{s^{kT} q^k}\right) G^k \left(I - \frac{q^k s^{kT}}{q^{kT} s^k}\right) + \frac{s^k s^{kT}}{s^{kT} q^k} \quad (2.13)$$

com,

$$s^k = \alpha^k d^k = x^{k+1} - x^k \quad (2.14)$$

$$q^k = \nabla f(x^{k+1}) - \nabla f(x^k) \quad (2.15)$$

Em todos estes métodos a matriz G^0 pode ser arbitrária, desde que seja definida positivamente e simétrica, pois alguns métodos conservam estas propriedades de iteração para iteração. Na maior partes dos casos assume-se $G^0 = I$, resultando d^0 como a direcção de descida máxima

Método dos Gradientes Conjugados

Nos problemas considerados de grandes dimensões, em que o número de varáveis é elevado, interessa evitar o cálculo das segundas derivadas de $f(x)$ e até a resolução do sistema de equações lineares para o cálculo da direcção de procura é desaconselhável. Assim os métodos baseados nas equações de Newton e os do tipo Quasi-Newton não devem ser implementados. Um desses métodos, entre outros, é o de *Polack-Ribière*,

$$d^0 = \nabla f(x^0) \quad (2.16)$$

$$\beta^k = \frac{(\nabla f(x^{k+1}) - \nabla f(x^k))^T \nabla f(x^k)}{\|\nabla f(x^k)\|^2} \quad (2.17)$$

$$d^{k+1} = -\nabla f(x^{k+1}) + \beta^k d^k \quad (2.18)$$

2.4.2.2 Escolha do Passo de Descida

Pretende-se agora o passo de descida α^k que garanta a convergência da função f . Alguns dos métodos mais usados na escolha do passo de descida são:

- i) α^k é escolhido tal que $f(x^k + \alpha^k d^k) = \min_{\alpha > 0} f(x^k + \alpha d^k)$. (*Regra da Minimização*)
- ii) α^k é escolhido como sendo uma constante, $\alpha^k = s, \forall k$. (*Passo de descida constante*)
 É obvio que com a escolha deste passo de descida, os pontos gerados não convergem necessariamente.
- iii) Para $k = 1$, $\alpha^k = s$, em cada passo $k + 1$ se $\alpha^k = s$ não for suficientemente pequeno para que a desigualdade $f(x^k + \alpha^k d^k) < f(x^k)$ seja cumprida, então tentar $\alpha^k = \beta^k s$ para $\beta \in [0, 1]$. (*Regra da Redução Sucessiva*)

Método de Armijo

O método de Armijo não é nada mais que uma regra para calcular o passo de descida. Não foi referida no item anterior pois esta regra situa-se entre a Regra da Minimização e a Regra da Redução Sucessiva. Considere-se então $s > 0$ com sendo o passo de descida inicial, $\sigma \in [0, 1]$ como sendo o parâmetro de tolerância e $\beta \in [0, 1]$ o valor pelo qual o passo de descida é reduzido em cada iteração. Seja $\alpha^k = \beta^{m^k} s$, onde $m^k \in 0, 1, 2, \dots$ é o primeiro inteiro tal que:

$$f(x^k) - f(x^k + \beta^{m^k} s d^k) \geq -\sigma \beta^{m^k} s \nabla f(x^k)^T d^k \quad (2.19)$$

onde o termo do lado direito é uma quantidade positiva.

2.4.2.3 Critério de Paragem

Os métodos utilizados na resolução de um problema de programação não linear, são iterativos. O processo gera uma sequência de aproximações x^k ao minimizante, x^* da função objectiva $f(x)$. O valor da função objectiva é muitas vezes o único elemento que pode ser usado para se verificar o processo de minimização. Quando o valor de $f(x)$ não diminui ao longo de um conjunto de iterações, assume-se que não é possível baixar mais o seu valor e que se atingiu um mínimo. No entanto, é quase impossível afirmar se o mínimo atingido é o mínimo global ou um dos locais. Nenhuma das técnicas iterativas mais usadas em optimização não linear, garante a convergência para um mínimo global, quando existem vários mínimos. O critério de paragem pode então ser atingido quando $\|x^{k+1} - x^k\| \leq \epsilon_1$ e $\|f^{k+1} - f^k\| \leq \epsilon_2$. Se $f(x)$ tiver derivadas parciais e contínuas pode-se também verificar se $\nabla f(x^{k+1}) \rightarrow 0$.

Método de Wolfe

é um critério de paragem para problemas de programação não linear que considera três condições, $\frac{\|x^{k+1} - x^k\|}{\|x^{k+1}\|} \leq \epsilon_1$, $\frac{\|f^{k+1} - f^k\|}{\|f^{k+1}\|} \leq \epsilon_2$, $\|\nabla f(x^{k+1})\| \leq \epsilon_3$, onde ϵ_1, ϵ_2 e ϵ_3 são quantidades pequenas e positivas, sendo a norma euclidiana mais aconselhável neste caso.

Método de Gill e Murray

este critério é baseado também em três condições, $\|x^{k+1} - x^k\| \leq \epsilon(1 + \|x^{k+1}\|)$, $\|f^{k+1} - f^k\| \leq \epsilon^2(1 + \|f^{k+1}\|)$ e $\|\nabla f(x^{k+1})\| \leq \epsilon^{\frac{1}{3}}(1 + \|f^{k+1}\|)$, onde ϵ é uma quantidade pequena e positiva que determina a precisão dos resultados no processo iterativo.

2.4.3 Optimização com Restições

2.4.3.1 Restições de Igualdade

Seja $f(x)$ a função objectiva $f(x)$, então pretende-se,

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} L(x, \lambda) \\ & \text{sujeito a :} \\ & h_i(x) = 0, \quad i = 1, \dots, m \end{aligned} \tag{2.20}$$

onde $m \leq n$.

É então definida a **função de Lagrange**, $L : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ como:

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i h_i(x) \quad (2.21)$$

onde $\lambda_i = \lambda_1, \dots, \lambda_m$ são os multiplicadores de Lagrange. Pretende-se agora encontrar as condições necessárias da função f relativamente às restrições h_i .

As condições necessárias para que (x^*, λ^*) seja um minimizador da função de Lagrange são:

$$\nabla_x L(x^*, \lambda^*) = \nabla_x f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla_x h_i(x^*) = 0 \quad (2.22)$$

e

$$\nabla_\lambda L(x^*, \lambda^*) = \sum_{i=1}^m h_i(x^*) = 0 \quad (2.23)$$

A expressão anteriores podem ser rescritas fazendo uma mudança de variável:

$$z = \begin{pmatrix} x \\ \lambda \end{pmatrix} \in \mathfrak{R}^{n+m}, \quad G(z) = 0 \quad (2.24)$$

este sistema representa as condições de Karush-Kuhn-Tucker (KKT) para um problema com restrições de igualdade. As condições KKT são condições necessárias para que seja garantida a convergência localmente, caso a função $f(x)$ seja convexa então as condições KKT são suficientes para poder afirmar que existe um mínimo local.

Para resolver o problema com equações não-lineares basta, por exemplo, recorrer-mos ao método de Newton para encontrar a solução.

2.4.3.2 Restrições de Desigualdade

Uma importante generalização para o problema de optimização ocorre quando são impostas restrições de desigualdade.

Para minimizar a função objectiva $f(x)$ que satisfaça r restrições de desigualdade,

$$\begin{aligned}
& \min_{x \in \mathfrak{R}^n} f(x) \\
& \text{sujeito a :} \\
& h_i(x) = 0, \quad i = 1, \dots, m \\
& g_j(x) \leq 0, \quad j = 1, \dots, r
\end{aligned} \tag{2.25}$$

Assim para que x^* seja um mínimo local neste tipo de problemas, tem de existir um vector único $\lambda^* \in \mathfrak{R}^m$ e $\mu^* \in \mathfrak{R}^r$ tal que:

$$\nabla_x L(x^*, \lambda^*, \mu^*) = \nabla_x f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla_x h_i(x^*) + \sum_{j=1}^r \mu_j^* \nabla_x g_j(x^*) = 0 \tag{2.26}$$

com

$$\begin{aligned}
& \mu_j^* \geq 0, \quad \forall j = 1, 2, \dots, r \\
& \mu_j^* = 0, \quad \text{se } \forall j \notin \{j \in \{1, 2, \dots, r\} | g_j(x^*) = 0\}
\end{aligned} \tag{2.27}$$

A solução das equações acima ou equações KT são a base de alguns algoritmos não-lineares. Estes algoritmos tentam calcular os multiplicadores de lagrange directamente. Os métodos derivados do método de Newton garantem a convergência das equações de Karush-Kuhn-Tucker (KKT).

2.5 Aplicação

Para que se melhor entenda o processo de optimização, vai ser dado um exemplo de aplicação das ferramentas de optimização.

Considere-se então $\chi : \mathfrak{R} \rightarrow \mathfrak{R}^+ = [0, +\infty[$ em que χ é uma função de Hardy definida por,

$$\chi(r) = \sqrt{\sigma^2 + r^2}, \quad \forall r \in \mathfrak{R} \text{ e } \sigma \geq 0 \tag{2.28}$$

As funções de Hardy são muitas vezes utilizadas para realizar, interpolações, aproximação de outras funções, entre outras coisas, pois conseguem transformar um conjunto de dados numa função. Posto isto, pretende-se, por exemplo, aproximar uma tabela de tracção T cujo valor é função da velocidade V e da altitude h , $T = T(V, h)$.

Utilizando a tabela (2.1), as funções de Hardy para cada ponto $r_k = [V \ h]^T$ são,

$$\chi^k : \chi^k(r) = \sqrt{\sigma^2 + \|r - r^k\|^2} \tag{2.29}$$

em que

$$\min\{r^k\} \leq \sigma \leq \sqrt{\|r^k\|^2} \quad (2.30)$$

Com isto pode ser definida uma combinação linear de funções de Hardy na forma,

$$\Gamma(r) = \sum_{k=1}^n \alpha^k \chi^k(r) \quad (2.31)$$

Por fim, a função de custo a ser minimizada será $\min_{\alpha} J(\alpha)$, com

$$J(\alpha^1, \dots, \alpha^n) = \sum_{i=1}^M (\Gamma(r^i) - T(r^i))^2 \quad (2.32)$$

O resultado da minimização será os valores de α que permitem a aproximação da tabela, deste modo pode ser utilizada teoria descrita nos itens anteriores para a procura de $\alpha = [\alpha^1, \dots, \alpha^n]^T$.

Resultados da *Optimização*

A ferramenta de utilizada para *optimização* foi a função *fsolve* da *Optimization Toolbox* do **MatLab**, onde foi possível observar a convergência do método. A aplicação desta ferramenta culminou na convergência do método (2.1).

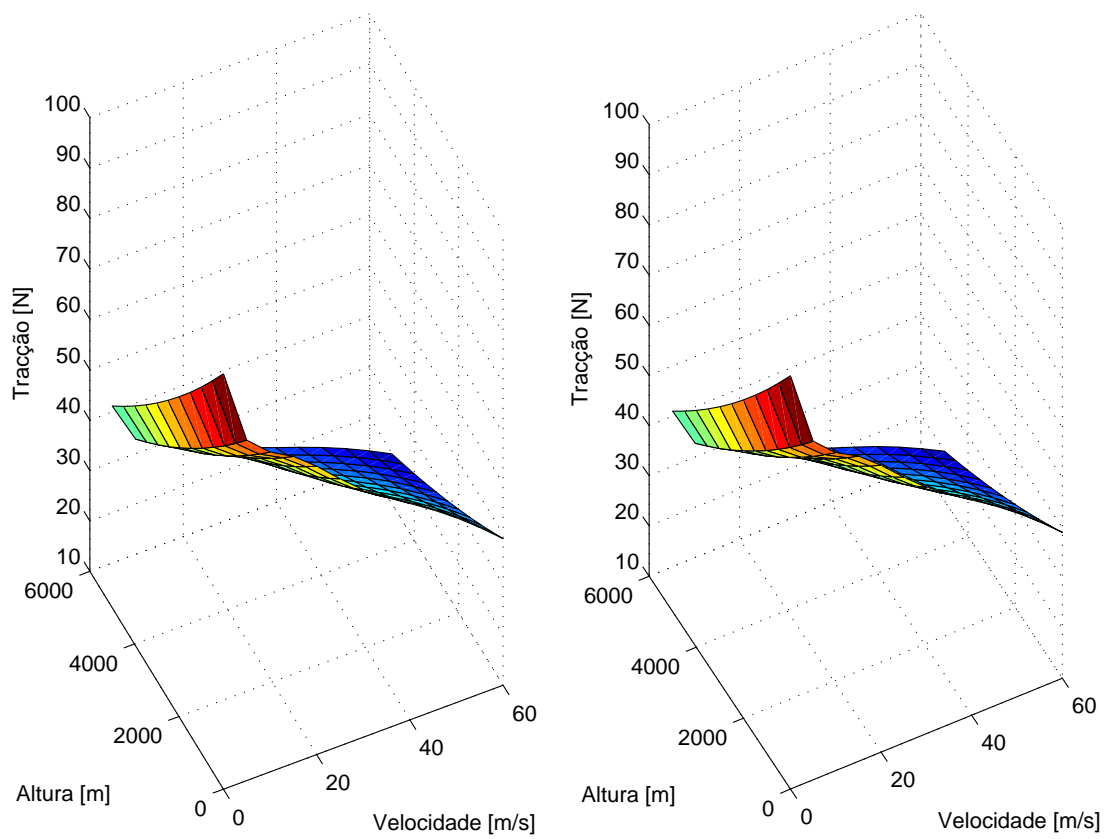


FIGURA 2.1: Representação gráfica dos valores da tabela (à esquerda) e da função de Hardy (à direita)

Tracção [N]	Velocidade [m/s]												
	0	5	10	15	20	25	30	35	40	45	50	55	60
0	92,3	77,4	73,4	70,8	67,1	62,9	59,1	55,8	52,92	49,9	46,6	43,0	39,07
500	87,3	73,2	69,4	66,9	63,4	59,4	55,8	52,7	49,9	47,1	44,0	40,5	36,8
1000	82,5	69,1	65,5	63,2	59,9	56,1	52,6	49,7	47,1	44,4	41,4	38,1	34,6
1500	77,8	65,2	61,8	59,6	56,4	52,9	49,6	46,9	44,4	41,8	39,0	35,9	32,5
2000	73,4	61,4	58,2	56,1	53,1	49,7	46,7	44,1	41,8	39,3	36,6	33,7	30,4
2500	69,1	57,8	54,8	52,8	50,0	46,8	43,9	41,5	39,2	36,9	34,4	31,6	28,5
3000	64,9	54,3	51,5	49,6	46,9	43,9	41,2	38,9	36,8	34,6	32,2	29,5	26,6
3500	61,0	51,0	48,3	46,5	44,0	41,1	38,6	36,5	34,5	32,4	30,1	27,6	24,8
4000	57,1	47,7	45,3	43,6	41,2	38,5	36,1	34,1	32,2	30,3	28,1	25,7	23,1
4500	53,5	44,6	42,3	40,7	38,4	35,9	33,7	31,8	30,1	28,2	26,2	23,9	21,4
5000	49,9	41,7	39,5	38,0	35,8	33,5	31,4	29,7	28,0	26,3	24,3	22,2	19,8

TABELA 2.1: Tabela de Tracção

Capítulo 3

Optimização de Trajectória

Neste capítulo irá ser abordado a *Teoria do Controlo Óptimo*. São várias as áreas que recorrem a esta metodologia, entre as quais, o ramo Aeroespacial, a Robótica e até mesmo a Economia. Os problemas de controlo óptimo ou optimização de trajectória podem ser interpretados como uma extensão de um problema de programação não linear para sistemas dinâmicos.

3.1 Formulação do Problema de Optimização de Trajectória

Um problema de optimização de trajectória tem as seguintes características,

- i) O problema aplica-se aos sistemas dinâmicos,
- ii) Pode haver restrições nalguns parâmetros ou variáveis do sistema considerado,
- iii) O problema implica a determinação de uma trajectória óptima que satisfaça um critério de performance especificado,
- iv) A determinação da trajectória óptima requer a procura de uma sequência de controlos para mandar o veículo seguir esta trajectória.

A optimização de trajectórias consiste em achar controlos para mandar um sistema seguir uma trajectória que optimize um dado critério de performance.

Muitos problemas de optimização no domínio aeroespacial, em robótica e mesmo em economia podem ser reformulados sob a forma de problemas de controlo óptimo. O problema geral da optimização de trajectórias pode ser expresso da seguinte forma,

onde x , u e p são respectivamente os vectores de estados, de controlos e de parâmetros, e t o tempo,

A dinâmica do sistema é descrita por um sistema de equações diferenciais,

$$\begin{aligned} \dot{x} &= f(t, x(t), u(t), p) \\ t_0 &\leq t \leq t_f \\ p_{min} &\leq p \leq p_{max} \end{aligned} \tag{3.1}$$

O problema consiste em achar um vector de controlo $u(t)$ ao longo do tempo para otimizar uma função de custo, *Índice de Performance* do tipo,

$$J(u) = \Phi(t_f, x(t_f), u(t_f), p) + \int_{t_0}^{t_f} L(t, x(t), u(t), p) dt \tag{3.2}$$

onde Φ e L são funções com valores reais.

Em alguns problemas pode-se vir a ter restrições em relação à trajectórias e/ou aos controlos permitidos,

$$h(t, x(t), u(t), p) = 0 \tag{3.3}$$

$$g(t, x(t), u(t), p) \leq 0 \tag{3.4}$$

Pode-se também ter condições de fronteira,

$$\Psi(x(t_0), x(t_f)) = 0 \tag{3.5}$$

3.1.1 Tipos de Problemas

Estando definido o problema de optimização de trajectória, é necessário definir o conceito de *índice de performance*. O *índice de performance* não é nada mais que a função de custo que se pretende minimizar num problema de programação não linear. Em optimização de trajectória a definição do *índice de performance* assenta em três problemas distintos.

3.1.1.1 Problema de Mayer

Seja o *índice de performance* J , que envolve apenas os estados iniciais e finais:

$$J[x(t), u(t)] = \Phi[t_N, x(t_N)] \quad (3.6)$$

então o problema é denominado de *Mayer*.

3.1.1.2 Problema de Lagrange

Caso sejam definido o tempo inicial t_0 e final t_f , e se pretenda minimizar o seguinte *índice de performance*,

$$J[x(t), u(t)] = \int_{t_0}^{t_f} L[x(t), u(t)] dt \quad (3.7)$$

então o problema é chamado de *Lagrange*.

3.1.1.3 Problema de Bolza

A junção do problema de *Mayer* e de *Lagrange* original o caso gerar que é denominado por problema de *Bolza*.

$$J[x(t), u(t)] = \Phi[t_N, x(t_N)] + \int_{t_0}^{t_f} L[x(t), u(t)] dt \quad (3.8)$$

3.1.1.4 Transformação do Problema de Bolza num Problema de Mayer

É frequentemente utilizada a passagem de uma formalização segundo Bolza para uma formalização segundo Mayer, pois o *Índice de Performance* resultante é menos complexo. Considere-se então a dinâmica de um determinado sistema $\dot{x} = f[x(t), u(t)]$, e o *Índice de Performance* representado em (3.8), em que $\dot{y} = L[x(t), u(t)]$. Assim,

$$y(t_f) = \int_{t_0}^{t_f} L[x(t), u(t)] dt \quad (3.9)$$

Aplicando a mudança de variável,

$$z = \begin{pmatrix} x \\ y \end{pmatrix} \quad (3.10)$$

temos que,

$$\dot{z} = \hat{f}[z(t), u(t)] = \begin{bmatrix} f(x, u) \\ L(x, u) \end{bmatrix} \quad (3.11)$$

O resultado é o seguinte *Índice de Performance* na formalização de Mayer,

$$J[x(t), u(t)] = \Phi[t_N, x(t_N)] + y(t_f) \quad (3.12)$$

3.2 Métodos de Procura de Trajectórias Óptimas

Os métodos de resolução dos problemas de optimização de trajectórias podem ser divididos em dois grupos, os métodos *indirectos* e os métodos *directos*.

3.2.1 Métodos Indirectos

Nos métodos *indirectos*, o problema de optimização de trajectória é transformado num problema de fronteira baseado na formulação das condições necessárias de optimalidade da trajectória através do Hamiltoniano.

Para que se melhor entenda a abordagem indirecta, considere-se um problema de optimização, com tempo final t_f não especificado, e vector de estado $x = (x_1, x_2, \dots, x_N)^T$ previamente formulado. Pretende-se então encontrar o vector de controlo $u = (u_1, u_2, \dots, u_M)^T$ que minimize o seguinte *índice de performance*:

$$J = \Phi[t_f, x(t_f)] + \int_{t_0}^{t_f} L[x(t), u(t)] dt \quad (3.13)$$

restringido pelas equações dinâmicas

$$\dot{x} = f[x(t), u(t)] \quad (3.14)$$

satisfazendo ainda, as condições iniciais

$$t_0 = 0, \quad x(t_0) = x_0 \quad (3.15)$$

e as condições finais

$$\Psi[x_f(t_f)] = 0 \quad (3.16)$$

onde Φ e L são escalares e Ψ um vector de $(p + 1) \times 1$.

O tempo final é livre e o valor óptimo será determinado. As condições a serem encontradas na otimização de trajectória resultam do diferencial de primeira e segunda ordem do *índice de performance extendido*:

$$\hat{J} = G[x(t_f), \nu] + \int_{t_0}^{t_f} [H[x(t), u(t), \lambda(t)] - \lambda(t)^T \dot{x}] dt \quad (3.17)$$

onde a função G e o Halmitoniano H são definidos por,

$$G = \Phi[x(t_f)] + \nu^T \Psi[x(t_f)] \quad (3.18)$$

$$H = L[x(t), u(t)] + \lambda^T f[x(t), u(t)] \quad (3.19)$$

O resultado do primeiro diferencial do *índice de performance extendido* é,

$$d\hat{J} = (G_{t_f} + L_f + G_{x_f} f_f) dt_f + (G_{x_f} - \lambda_f^T) \delta x_f + \int_{t_0}^{t_f} [(H_x + \dot{\lambda}^T) \delta x + H_u \delta u] dt \quad (3.20)$$

A determinação dos multiplicadores de Lagrange $\lambda(t)$ e ν é obtido por anulação dos coeficientes δx e de df_f . O resultado é:

$$\begin{aligned} \dot{\lambda} &= -H_x^T \\ \lambda_f &= G_{x_f}^T \end{aligned} \quad (3.21)$$

e

$$G_{t_f} + L_f + G_{x_f} f_f = 0 \quad (3.22)$$

com isto pode-se afirmar que $d\hat{J} = 0$ se,

$$H_u = 0 \quad (3.23)$$

Assim, para encontrar o minimizador de \hat{J} , conseqüente maximizador do Hamiltoniano, tem que se resolver as seguintes equações de Euler-Lagrange,

$$\begin{aligned} \dot{x} &= f[x(t), u(t)] \\ \dot{\lambda} &= -H_x^T[x(t), u(t), \lambda(t)] \\ 0 &= H_u^T[x(t), u(t), \lambda(t)] \end{aligned} \quad (3.24)$$

Esta formulação é o chamado princípio de Pontryagin, e representa as condições necessárias (não suficientes) para que u seja um minimizador do *índice de performance extendido*. O princípio do máximo de Pontryagin afirma que se um vector de controlo u satisfizer as restrições mencionadas no problema de optimização de trajectórias, então se o *Índice de Performance* J for minimizado, necessariamente o controlo u maximiza o Hamiltoniano H .

Para definir as condições suficientes é necessário calcular o diferencial de segunda ordem do *índice de performance extendido* \hat{J} ,

$$d^2\hat{J} = [\delta x_f^T dt_f] \begin{bmatrix} G_{x_f x_f} & \Omega_{x_f}^T \\ \Omega_{x_f} & \hat{\Omega} \end{bmatrix} \begin{bmatrix} \delta x_f \\ \delta t_f \end{bmatrix} + \int_{t_0}^{t_f} [\delta x^T \delta u^T] \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix} \quad (3.25)$$

Assim, para garantir que u seja um minimizador local é preciso que u cumpra não só as condições necessárias, mas também as seguintes condições,

$$\begin{bmatrix} G_{x_f x_f} & \Omega_{x_f}^T \\ \Omega_{x_f} & \hat{\Omega} \end{bmatrix} > 0 \quad , \quad \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} > 0 \quad (3.26)$$

ou seja, as matrizes anteriores têm de ser definidas positivamente.

3.2.1.1 Descretização do Método

Considere-se o tempo t compreendido entre o tempo inicial t_0 e o tempo final t_f , $t_0 < t < t_f$, com $\lambda(t_0) = \lambda_0$, $x(t_0) = x_0$ e $u(t_0) = u_0$. Pode-se então definir,

$$H_0(t, x, u, \lambda_0) = L(t, x, u) + \lambda_0^T f(t, x, u) \quad (3.27)$$

Tendo em conta as condições necessárias (3.24), as condições iniciais, é possível discretizar o problema da seguinte maneira,

Seja N o número de pontos ou nós de uma determinada trajectória, então $h = \frac{t_f - t_i}{N}$.

Aplicando, por exemplo, o método de *Euler explícito* ao sistema de equações diferenciais para cada $k = 0, 1, \dots, N - 1$, temos,

$$x_{k+1} = x_k + hf(t_k, x_k, u_k) \quad (3.28)$$

$$\lambda_{k+1} = \lambda_k - h \frac{\partial H(t_k, x_k, u_k)}{\partial x} \quad (3.29)$$

$$0 = \frac{\partial H(t_k, x_k, u_k)}{\partial u} \quad (3.30)$$

Caso seja possível resolver o sistema discretizado em cada iteração, e seja verificado as condições (3.26) é possível afirmar que u_k é um minimizador do *índice de performance extendido*.

3.2.2 Métodos Directos

Nos métodos *directos*, o problema de optimização de trajectória é transformado num problema de optimização de parâmetros conhecido como *programação não linear*. O processo de conversão, começa por dividir o intervalo de tempo, do problema de controlo óptimo, num número finito de subintervalos, o tempo final de cada segmento denomina-se por *nó*. Então, a ideia básica é escolher os controlos e/ou estados nos vários nós (também denominados por parâmetros de controlo e parâmetros de estado) que serão as variáveis a ser encontrados através do processo de optimização não linear.

Encontrado os parâmetros finais de um dado problema de controlo óptimo, é possível reconstruir o histórico, em relação ao tempo, das variáveis de controlo e estado através de um qualquer método de interpolação polinomial.

Existem quatro classes de métodos para conversão de um problema de controlo óptimo num problema de optimização de parâmetros, são elas,

i) Unicamente parametrização dos controlos.

- ii) Parametrização dos controlos, e parametrização dos estado em alguns nós.
- iii) Parametrização do estado e do controlo.
- iv) Unicamente parametrização do estado.

Tendo em conta o *índice de performance* na forma da equação (3.13) e as suas respectivas restrições, tranforme-se então este problema num problema de programação não linear. Assim, para o tempo t compreendido entre o tempo inicial t_0 e o tempo final t_f , $t_0 < t < t_f$, e uma trajectória dividida em $N - 1$ secções, obtemos um *índice de performance* na sua forma descretizada no caso de um problema com critério integral,

$$J[u(t_0), u(t_1), \dots, u(t_N)] = \Phi[x(t_N)] + \sum_{i=1}^M \alpha_i L[x(t_i), u(t_i)] \quad (3.31)$$

Com o *índice de performance* descretizado, é necessário encontrar um método que possibilite descretizar as restrições de estado. A descretização não é nada mais que a integração numérica efectuada às equações diferenciais $\dot{x} = f(x(t), u(t))$, onde t_0, x_0, u_0 , são conhecidos. No problema de optimização de trajectória, são apenas consideradas integrações de passo fixo. As integrações explícitas são derivadas da aproximação de Runge-Kutta, enquanto que as integrações implícitas derivam de diferentes métodos como Runge-Kutta, diferenças finitas, Newton-Coates e Colocação sendo o resultado praticamente o mesmo.

De seguida são apresentados alguns métodos de descretização que possibilitam a transcrição do problema de optimização de trajectória para o problema de programação não linear,

3.2.2.1 Colocação

No método de descretização *trapezoidal*, as variáveis de programação não linear são,

$$y = [t_0, t_1, \dots, t_f, x(t_0), u(t_0), x(t_1), u(t_1), \dots, x(t_f), u(t_f), p]^T \quad (3.32)$$

As equações de estado são aproximadamente satisfeitas definindo os chamados *defeitos*,

$$\zeta_{k+1} = x_{k+1} - x_k - \frac{h_k}{2} [f(x_{k+1}, u_{k+1}) + f(x_k, u_k)] \quad (3.33)$$

em que $k = 0, 1, \dots, N - 1$, sendo N o número de nós e $h_k = t_{k+1} - t_k$.

A discretização também pode ser de Hermite-Simpson, alterando-se neste caso, as variáveis de programação não linear, pois nestas são acrescentados valores do estado e controlo entre nós, assim,

$$y = [t_0, t_1, \dots, t_f, (x, u, u_m)_i, (x, u, u_m)_{i+1}, \dots, (x, u)_f, p]^T \quad (3.34)$$

em que $i = 0, 1, \dots, N - 2$, e $(u_{m_0}, u_{m_1}, \dots, u_{m_{N-2}})$ são os valores dos controlos no meio de cada segmento.

Para este algoritmo de discretização os *resíduos* são dados por,

$$\zeta_k = x_{k+1} - x_k - \frac{h_k}{6} [f(x_{k+1}, u_{k+1}) + 4f(x_{m_k}, u_{m_k}) + f(x_k, u_k)] \quad (3.35)$$

onde $f(x, u)$ representa as equações de movimento avaliadas nos nós e intervalos entre nós.

O vector de estado e as equações de movimento no meio de cada segmento são dadas por,

$$x_{m_k} = \frac{1}{2}[x_k + x_{k+1}] + \frac{h_k}{8} [f(x_k, u_k) - f(x_{k+1}, u_{k+1})] \quad (3.36)$$

$$f_{m_k} = f(x_{m_k}, u_{m_k}, p, t_k + \frac{h_k}{2}) \quad (3.37)$$

para $k = 0, 1, \dots, N - 2$.

Nestas equações h_k é o intervalo de tempo entre segmentos. Para segmentos de igual duração $h_k = (t_f - t_i)/(N - 2)$.

O método da Colocação, consiste então, em discretizar as equações de estado, ou quaisquer outras restrições, de forma implícita. Para isso pode-se recorrer, à discretização trapezoidal, de Hermite-Simpson, ou qualquer outro método, que permitem contruir os denominados *defeitos* que se tornam restrições no problema de programação não linear.

A aproximação das variáveis de estado e de controlo podem ser feita através funções, no caso da colocação de Hermite-Simpson, os polinómios de terceiro grau de Hermite são utilizados para representar as variáveis de estado, enquanto que as variáveis de controlo podem ser aproximadas por funções constantes, lineares, quadráticas ou polinómios de terceiro grau.

Para a Colocação, o problema de optimização de trajectória, é formulado da seguinte maneira para que seja possível resolver num problema de programação não linear,

$$\min_{y \in \mathfrak{R}^n} J(y) = \Phi(y) + \sum_{i=1}^M \alpha_i L(y) \quad (3.38)$$

tal que

$$x_{k+1} - x_k - \frac{h_k}{2} [f(x_{k+1}, u_{k+1}) + f(x_k, u_k)] = 0 \quad (3.39)$$

$$\Psi_0 = 0 \quad (3.40)$$

$$\Psi_f = 0 \quad (3.41)$$

$$g(y) \leq 0 \quad (3.42)$$

A equação dos *defeitos* neste caso foi representada na sua forma discretizada pelo método trapezoidal, mas podia ter sido representada, por exemplo, pelo método de Hermite-Simpson.

3.2.2.2 Métodos de Runge Kutta

Os métodos de integração, explícitos e implícitos são todos derivados das aproximações de Runge-Kutta. Dado o tempo inicial t_0 , o estado inicial $x_0 = x(t_0)$ e o tempo o passo de integração $h_k = t_{k+1} - t_k$, pode ser obtida a seguinte expressão,

$$x_{k+1} = x_k + h_k \sum_{i=1}^M c_i f_i \quad (3.43)$$

onde

$$f_i = f(t_k + h_k \alpha_i, x_k + h_k \sum_{j=1}^M \beta_{ij} f_j) \quad (3.44)$$

Em que M é o número de funções a ser avaliadas, c , α e β são constantes cujos valores pretendem ser determinados.

Na prática, basta aplicar um dos métodos de Runge-Kutta às equações diferenciais, e daí construir os *resíduos*. Por exemplo, caso seja aplicado o método de Runge-Kutta de 2ª Ordem, os *defeitos* vêm na seguinte forma,

$$\zeta_k = x_{k+1} - x_k - \frac{h_k}{2}(k_1 + k_2) \quad (3.45)$$

$$k_1 = f(t_k, x_k) \quad (3.46)$$

$$k_2 = f(t_k + h_k, x_k + h_k k_1) \quad (3.47)$$

A formulação do problema de optimização de trajectória, com base neste tipo de integração numérica, é em tudo parecida ao da Colocação mudando apenas a descretização das equações de estado.

3.2.2.3 Pseudoespectral

Os métodos pseudo-espectrais têm sido desenvolvidos para os métodos directos, com o intuito de determinar trajectórias óptimas de sistemas não lineares de ordem elevada. Nestes métodos são normalmente usados os polinómios de *Legendre* e *Chebyshev* para aproximarem as variáveis de estado e controlo. Como a abordagem para a utilização dos dois polinómios é a mesma então, optou-se por escolher o método pseudo-espectral baseado em *Chebyshev*.

O procedimento para aproximação das variáveis de estado e de controlo é baseado na interpolação dos polinómios de *Chebyshev* nos respectivos nós de *Chebyshev*. Deste modo convém definir este método,

Seja $N + 1$ o número de nós que definem uma determinada trajectória, os polinómios de *Chebyshev*, são definidos entao, na sua forma trigonométrica como,

$$T_N(t) = \cos(N \cos^{-1}(t)) \quad (3.48)$$

Os respectivos nós, são definidos no intervalo $t \in [-1, 1]$, tomando a forma,

$$t_k = \cos\left(\frac{k\pi}{N}\right) \quad (3.49)$$

com $k = 0, 1, \dots, N$.

A parametrização do estado e do controlo, assenta na definição que se segue, respectivamente,

$$x_i^N(t) = \sum_{k=0}^N x_i(t_k) \varphi_k(t) \quad (3.50)$$

$$u_j^N(t) = \sum_{k=0}^N u_j(t_k) \varphi_k(t) \quad (3.51)$$

sendo o número de variáveis de estado $i = 1, 2, \dots, n$, o número de variáveis de controlo $j = 1, 2, \dots, m$ e $\varphi_k(t)$ os polinómios de *Lagrange* contruídos nos *nós de Chebyshev*,

$$\begin{aligned} \varphi_k(t) &= \frac{(-1)^{k+1}(1-t^2)\dot{T}_N(t)}{c_k N^2(t-t_k)} \\ &= \frac{2}{N c_k} \sum_{l=0}^N \frac{T_l(t_k) T_l(t)}{c_l} \end{aligned} \quad (3.52)$$

com

$$c_k = \begin{cases} 2 & \text{se } k = 0 \vee k = N \\ 1 & \text{se } k = 1, \dots, N-1 \end{cases} \quad (3.53)$$

A particularidade destes polinómios, é que nos *nós de Chebyshev* as variáveis tomam o valor exacto, assim define-se,

$$\varphi_k(t_l) = \delta_{kl} = \begin{cases} 1 & \text{se } k = l \\ 0 & \text{se } k \neq l \end{cases} \quad (3.54)$$

em que δ_{kl} é delta de *Kronecker*.

Neste método, equações de estado são aproximadas por funções calculadas nos *nós*, pelo que não é necessário aproximar a equação diferencial em todo o seu domínio. Assim,

$$\dot{x}_i^N(t_k) = \sum_{l=0}^N C_{kl} x_i(t_l) \quad (3.55)$$

onde,

$$\frac{dC_l}{dt}(t_k) = C_{kl} = \begin{cases} \frac{(2N^2+1)}{6} & k = l = 0 \\ -\frac{(2N^2+1)}{6} & k = l = N \\ -\frac{t_l}{2(1-t_l^2)} & 1 \leq k = l \leq N - 1 \\ \frac{c_k(-1)^{k+l}}{c_l(t_k-t_l)} & k \neq l \end{cases} \quad (3.56)$$

Para facilitar a formulação da programação não linear, use-se a notação,

$$x_{il} := x_i(t_l), \quad u_{jl} := u_j(t_l)$$

com,

$$x_{il} = (x_{i0}, x_{i1}, \dots, x_{iN}), \quad u_{jl} = (u_{j0}, u_{j1}, \dots, u_{jN})$$

Tendo isto, o conjunto das variáveis de programação linear veêm,

$$y = [x_{il}, u_{jl}, \tau_0, \tau_f]$$

em que τ_0 e τ_f são o tempo inicial e final, respectivamente, e τ é dado por,

$$\tau = \frac{(\tau_f - \tau_0)t + (\tau_f + \tau_0)}{2} \quad (3.57)$$

Então, dado um determinado *Índice de Performance*, pretende-se,

$$\min_y J(y) \quad (3.58)$$

sujeito a,

$$\left(\frac{\tau_f - \tau_0}{2}\right) f(x_{il}, u_{jl}) - \sum_{l=0}^N C_{kl} x_i(t_l) = 0 \quad (3.59)$$

$$\Psi_0(x_{0l}, \tau_0) = 0 \quad (3.60)$$

$$\Psi_f(x_{Nl}, \tau_f) = 0 \quad (3.61)$$

$$g(y) \leq 0 \quad (3.62)$$

3.3 Aplicação

No presente item vai-se aplicar a teoria, que até agora tem vindo sendo descrita, a um problema muito estudado, que é o da transferência de órbita de veículo espacial entre o planeta *Terra* e planeta *Marte*. Este problema é também conhecido por *Problema de Bryson-Ho*.

As considerações que vão ser feitas para a resolução deste problema é que, é contínuo e que a órbita é coplanar e de baixo consumo. As órbitas dos planetas são assumidas como coplanares e circulares e a duração da transferência é de 193 dias.

3.3.1 Descrição do Problema de Bryson-Ho

As equações $\dot{x} = f(x, u)$ que representam o movimento de um veículo numa transferência de órbita são,

$$\dot{r} = \frac{dr}{dt} = u \quad (3.63)$$

$$\dot{u} = \frac{du}{dt} = \frac{v^2}{r} - \frac{\mu}{r^2} + \frac{T}{m} \sin \phi \quad (3.64)$$

$$\dot{v} = \frac{dv}{dt} = -\frac{uv}{r} + \frac{T}{m} \cos \phi \quad (3.65)$$

onde r é a posição radial, u é a velocidade radial, v é a velocidade tangencial, T é a tracção do motor, m é a massa do veículo, ϕ é o ângulo definido entre a posição do veículo e a direcção tangencial deste, μ é a constante gravitacional.

O vector de estado x e controlo u são, respectivamente,

$$x = \begin{bmatrix} r \\ u \\ v \end{bmatrix}, \quad u = \begin{bmatrix} \phi \end{bmatrix}$$

A massa do veículo é definida em cada tempo τ por,

$$m(\tau) = m_0(1 - \dot{m}\tau) \quad (3.66)$$

em que \dot{m} é a razão de consumo de combustível do sistema propulsivo e m_0 é a massa inicial do veículo.

As condições Ψ_0 de voo do veículo para que se crie uma orbita inicial circular são,

$$\begin{aligned} r(0) &= r_0 \\ u(0) &= u_0 = 0 \\ v(0) &= v_0 = \sqrt{\frac{\mu}{r_0}} \end{aligned} \tag{3.67}$$

Para que se crie uma orbita circular no tempo final, é necessário impor que a velocidade radial seja zero e a velocidade tangencial seja igual à velocidade circular local, assim temos as seguintes condições de fronteira Ψ_f ,

$$\begin{aligned} u(\tau_f) &= u_f = 0 \\ v(\tau_f) - \sqrt{\frac{\mu}{r(\tau_f)}} &= 0 \end{aligned} \tag{3.68}$$

O que se pretende então, para este problema, é *maximizar* o raio final da órbita circular. A formalização deste problema em termos do problema de optimização de trajectória é a seguinte,

$$\min_{\substack{x \in \mathbb{R}^n \\ u \in \mathbb{R}^m}} J(\tau, x(\tau), u(\tau)) = -r(\tau_f) \tag{3.69}$$

sujeito a

$$\dot{x} = f(x, u) = 0 \tag{3.70}$$

$$\Psi_0 = 0 \tag{3.71}$$

$$\Psi_f = 0 \tag{3.72}$$

$$x_l \leq x \leq x_u \tag{3.73}$$

$$u_l \leq u \leq u_u \tag{3.74}$$

3.3.2 Solução Numérica para o Problema de Optimização de Trajectória

Tendo já sido formulado o problema de Bryson-Ho e visto qual o problema de optimização de trajectória a ser resolvido, à que utilizar um método de procura de soluções óptimas. Os métodos de procura, como já foi referido, dividem-se em métodos *indirectos*

e métodos *directos*. Os métodos *indirectos* para resolverem o problema de optimização da trajectória necessitam das condições de optimabilidade que acentam no princípio de Pontryagin. Estes métodos não são muito utilizados para resolverem problemas complexos pois possuem um raio de convergência pequeno e ainda é necessário um cálculo adicional para verificar as condições suficientes.

Os métodos *directos* têm a vantagem, de não ser preciso construir o *Hamiltoniano*, basta transformar apenas o problema de controlo óptimo num de programação não linear, que a convergência para a solução é garantida (caso seja possível). Por isto, neste trabalho, vão ser apenas estudados os métodos *directos* e dentro destes vão ser aplicados o método da Colocação e o método Pseudo Espectral, que mais à frente será explicado o porquê desta escolha.

A programação deste problema, e de outros no decorrer deste trabalho, foi feita integralmente em *MatLab*[©] utilizando o algoritmo de optimização não linear sujeito a restrições, que também podem ser não lineares, *fmincon* da **Optimization Toolbox**. Este método de programação não linear (*solver*) não é de todo o melhor, pois padece de falta de precisão na convergência da solução e o tempo que leva a encontra-la também é bastante elevado. Existem outras *tollbox* com melhor performance, melhor aproximação da solução e tempo de cálculo para encontrar esta bastante mais rápido. Este facto é devido ao que foi retratado no Capítulo 2, pois o processo de optimização depende muito do tipo de algoritmo utilizado, da forma como é programado e do tipo de linguagem utilizada.

Quanto à resolução do problema de Bryson-Ho, ainda é preciso ter em conta outras considerações, que é o facto da escala ser bastante grande, o que pode levar muito tempo de simulação e podendo mesmo não haver convergência para a solução óptima. Assim, à que normalizar o problema para que a escala seja reduzida e se consiga encontrar uma solução óptima.

	Não Normalizados	Normalizados
T	3.71 N	0.1405
m_0	4535.9 Kg	1
\dot{m}	5.85 Kg/dia	0.07487
τ_f	193 dias	3.32

TABELA 3.1: Dados normalizados do problema

A tabela(3.1) mostra-nos os valores normalizados dos parâmetros necessários à resolução do problema. A tracção adimensional é obtida por $(T/m_0)/(\mu/r_0^2)$, o tempo adimensional obtêm-se através de $\tau_f/\sqrt{r_0^3/\mu}$ e o consumo de combustível adimensional $\dot{m}\tau_f/(m_0\tau_{f_a})$, em que τ_{f_a} é o tempo da transferência de órbita normalizada. Com isto, as variáveis de estado iniciais tomam valores em que, $r_0 = 1$, $u_0 = 0$ e $v_0 = 1$.

Na tabela 3.2 estão indicados os valores finais do vector de estado. As unidades estão em unidades astronómicas. Quanto aos resultados obtidos é possível afirmar que o método pseudoespectral dá valores muito próximos ao dos autores do problema. Não é possível dissertar explicitamente sobre a influência dos nós, pois a partir do momento que o problema está descrito em termos de programação não linear é o *solver* que influencia o resultado, e este à medida que o número de nós aumenta tem mais dificuldade em refinar a solução.

Contudo é possível observar que o método da Colocação maximiza mais a função de custo, mas por outro lado, não possui um resultado como o obtido no método pseudoespectral em relação à velocidade radial, que é o especificado nas condições de fronteira.

O comportamento do estado $u(\tau)$, $v(\tau)$ e $r(\tau)$ quer num método quer noutro são em tudo parecidos aos dos autores, mais no método pseudoespectral, contudo no controlo $\phi(\tau)$ existe diferenças acentuadas. Nas figuras (3.4), (3.8) e (3.12) observa-se que o método Colocação aproxima-se de valores de outras referências, o certo é, que este método pseudoespectral aqui utilizado possui amplitudes menores mas apresentando alguma oscilação. Oscilação esta que poderá ser retirada se a aproximação à função $\phi(\tau)$ for feita, por exemplo por Hardy¹.

Outra ilação que se retira dos resultados, é que para trinta nós os resultados aproximação mais entre os dois métodos, acrescentando a isto o facto do controlo rondar valores ao de outros autores.

Como já foi referido o cálculo deste problema depende muito do método computacional utilizado, mesmo sabendo que o utilizado ao longo do presente estudo, não é de todo o melhor, mas os resultados obtidos comprovam que estes métodos são capazes de resolverem problemas de optimização de trajectória, e para além disto dá-nos boas expectativas em relação aos métodos pseudoespectrais para aplicar ao tema deste trabalho.

¹ver Capítulo 2

	$u(\tau_f)$	$v(\tau_f)$	$J = r(\tau_f)$
	[ua/dia]	[ua/dia]	[ua]
Colocação			
N = 10	0.1218	0.7692	1.5055
N = 30	0.0396	0.7567	1.5997
N = 50	0.0240	0.7542	1.6500
Pseudoespectral			
N = 10	0.0010	0.8157	1.4994
N = 30	0.0010	0.8065	1.5336
N = 50	0.0010	0.8145	1.5035

TABELA 3.2: Valores das variáveis de estado no final da transferência de órbita

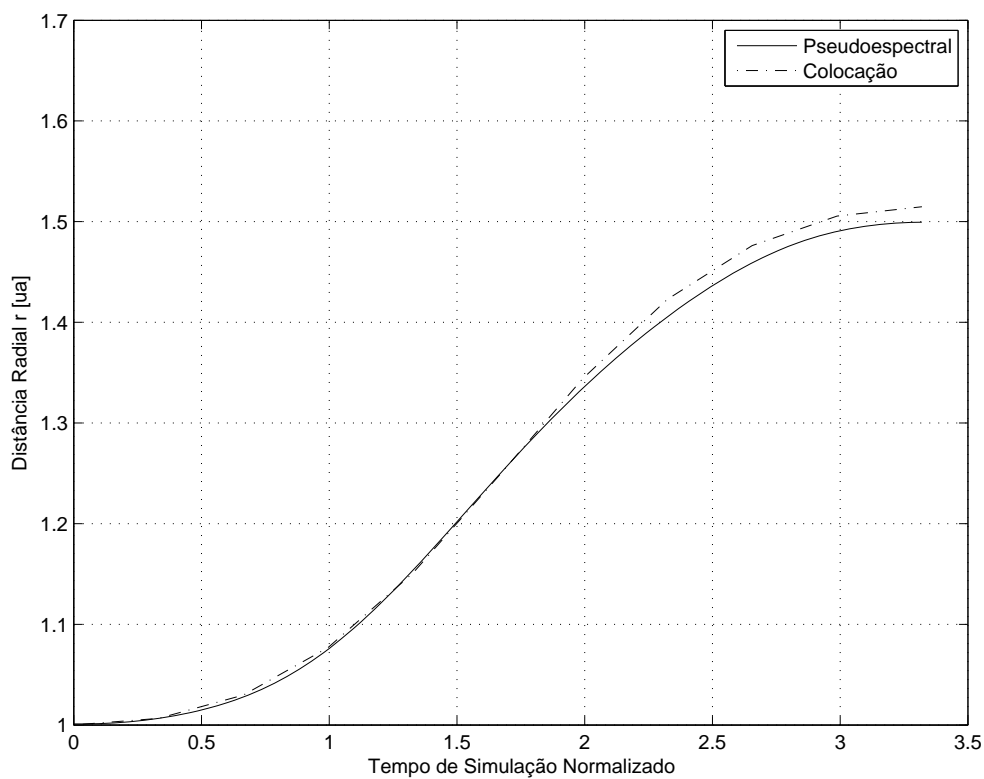


FIGURA 3.1: Distância Radial vs Tempo de Simulação para N = 10 nós

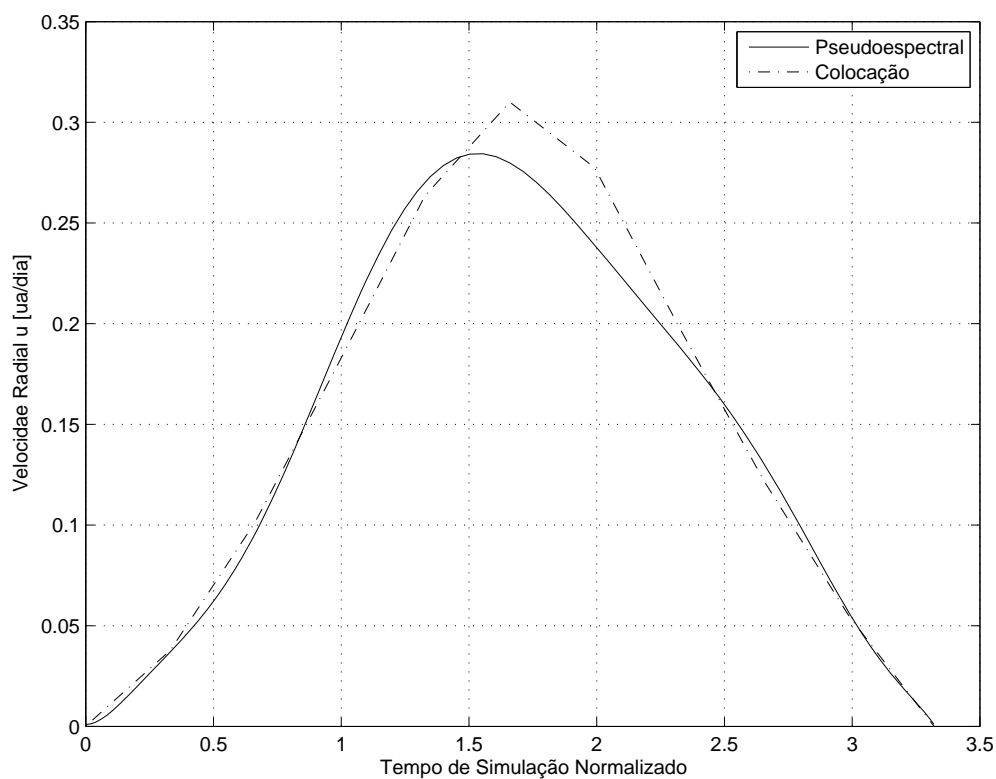


FIGURA 3.2: Velocidade Radial vs Tempo de Simulação para $N = 10$ nós

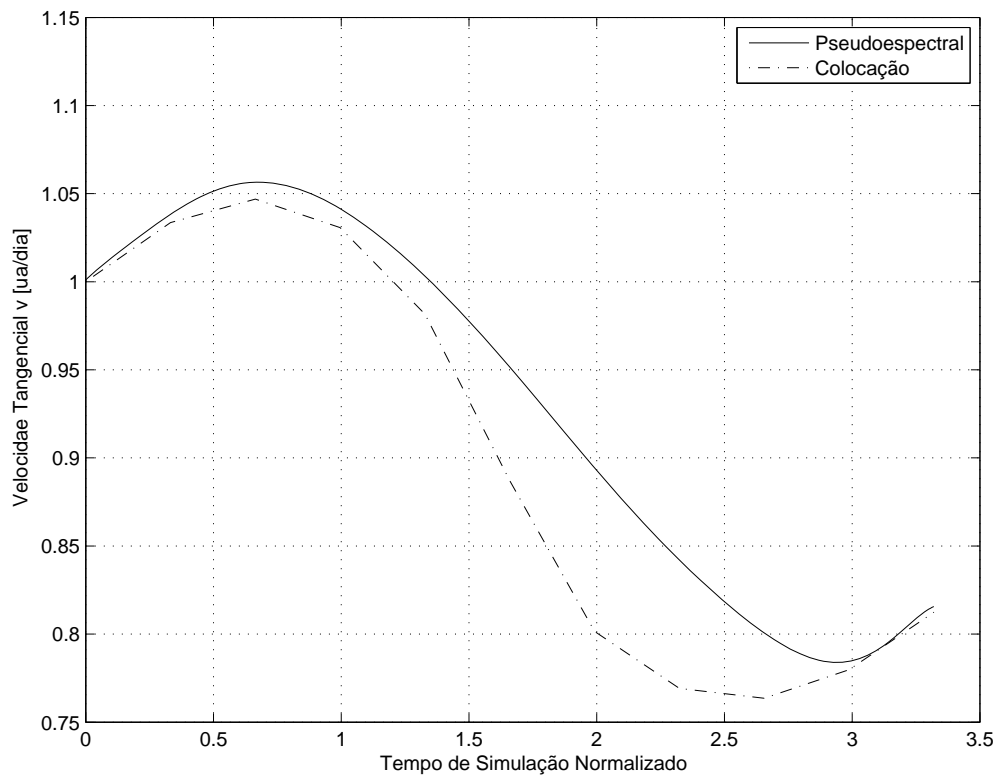


FIGURA 3.3: Velocidade Tangencial vs Tempo de Simulação para $N = 10$ nós

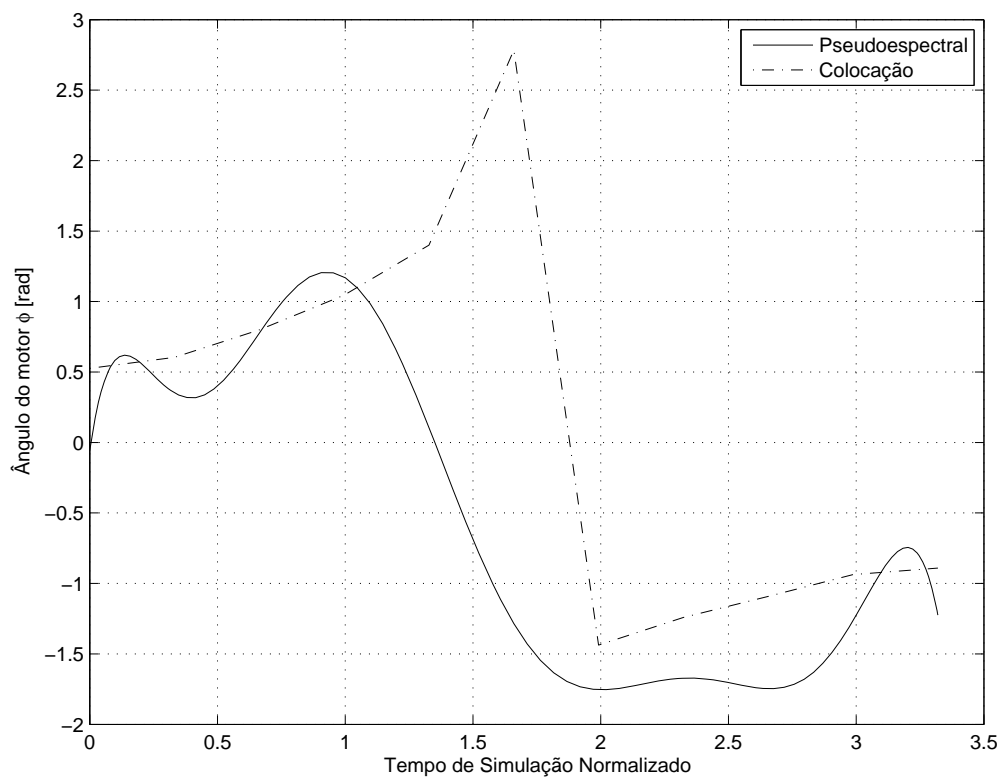


FIGURA 3.4: Ângulo do Sistema propulsivo vs Tempo de Simulação para $N = 10$ nós

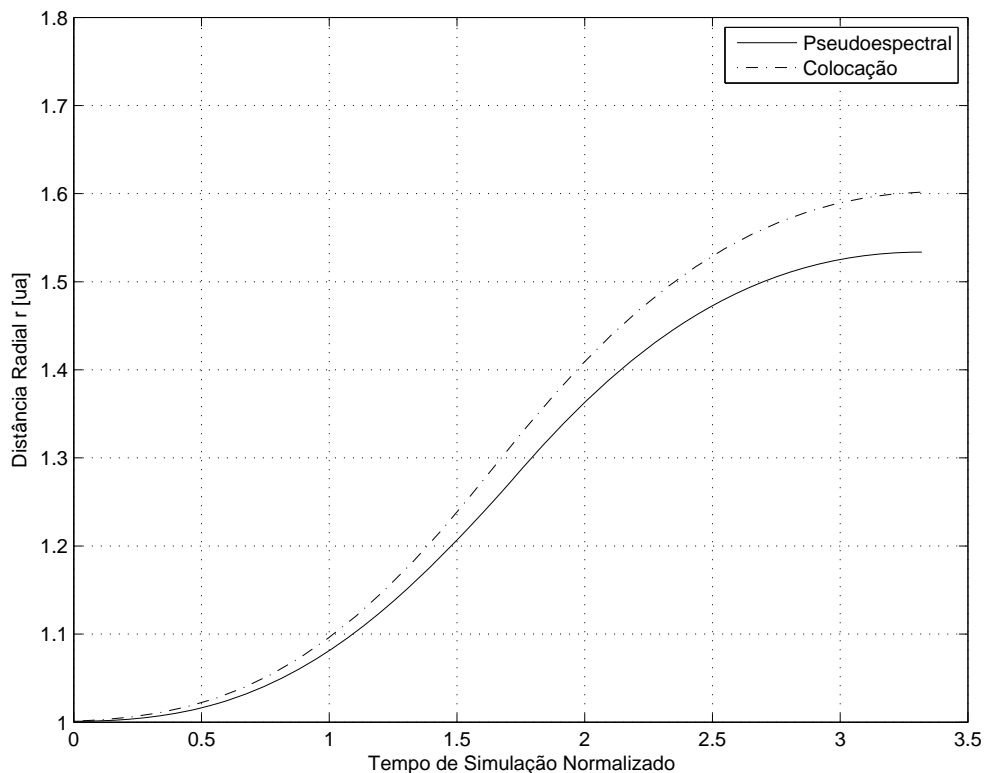


FIGURA 3.5: Distância Radial vs Tempo de Simulação para $N = 30$ nós

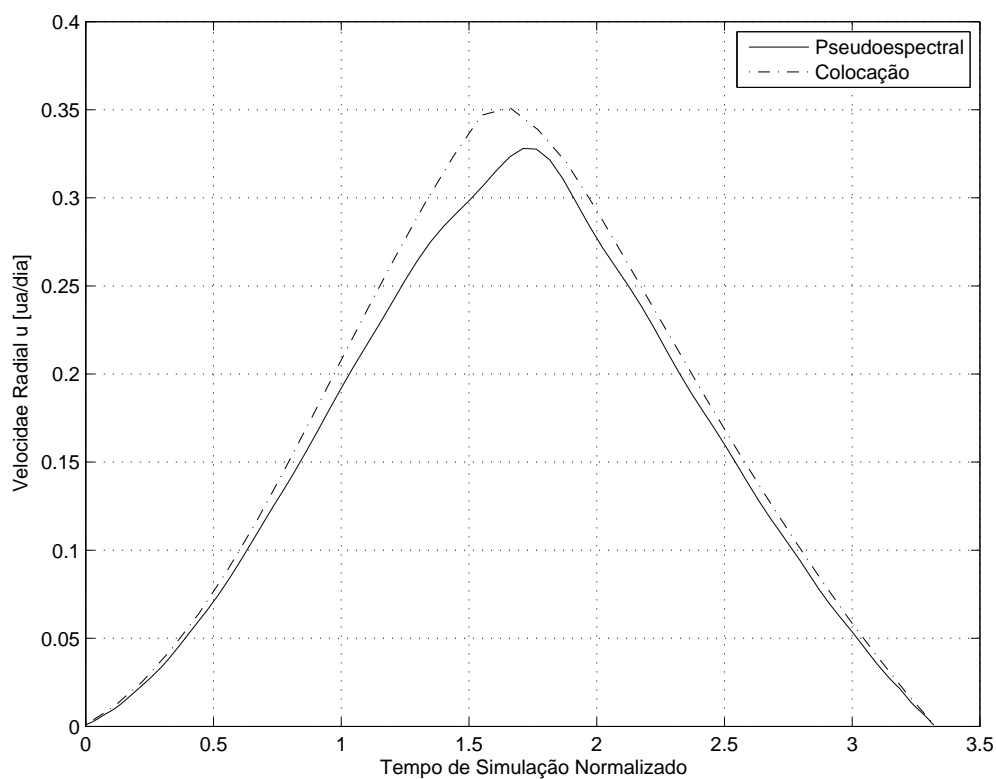


FIGURA 3.6: Velocidade Radial vs Tempo de Simulação para $N = 30$ nós

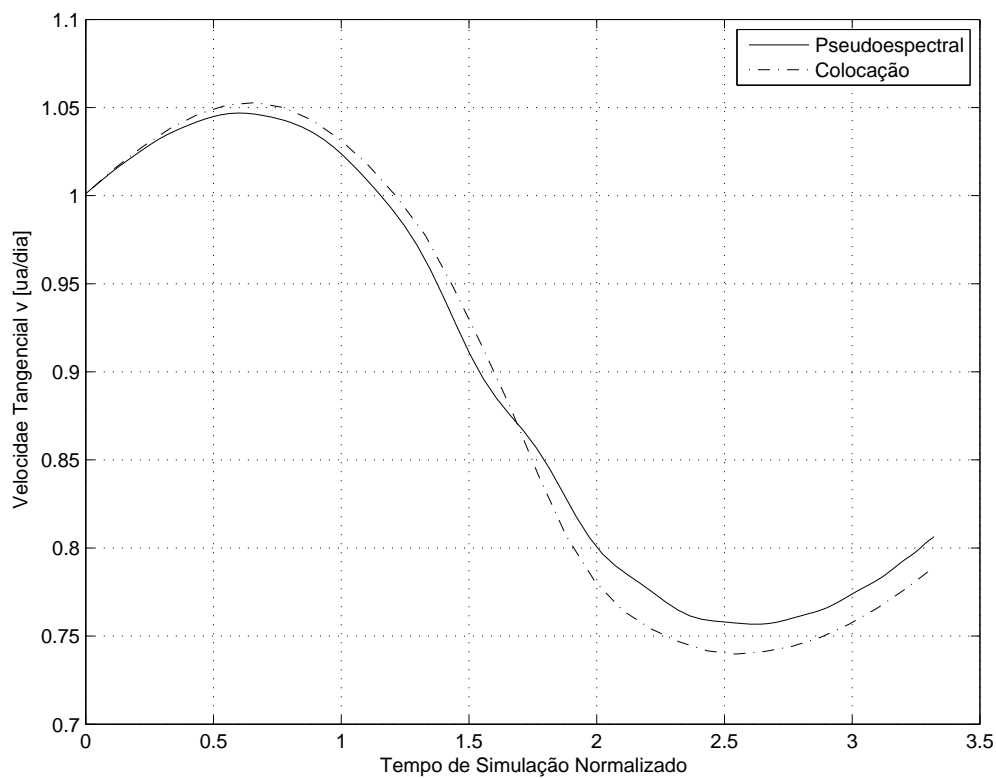


FIGURA 3.7: Velocidade Tangencial vs Tempo de Simulação para $N = 30$ nós

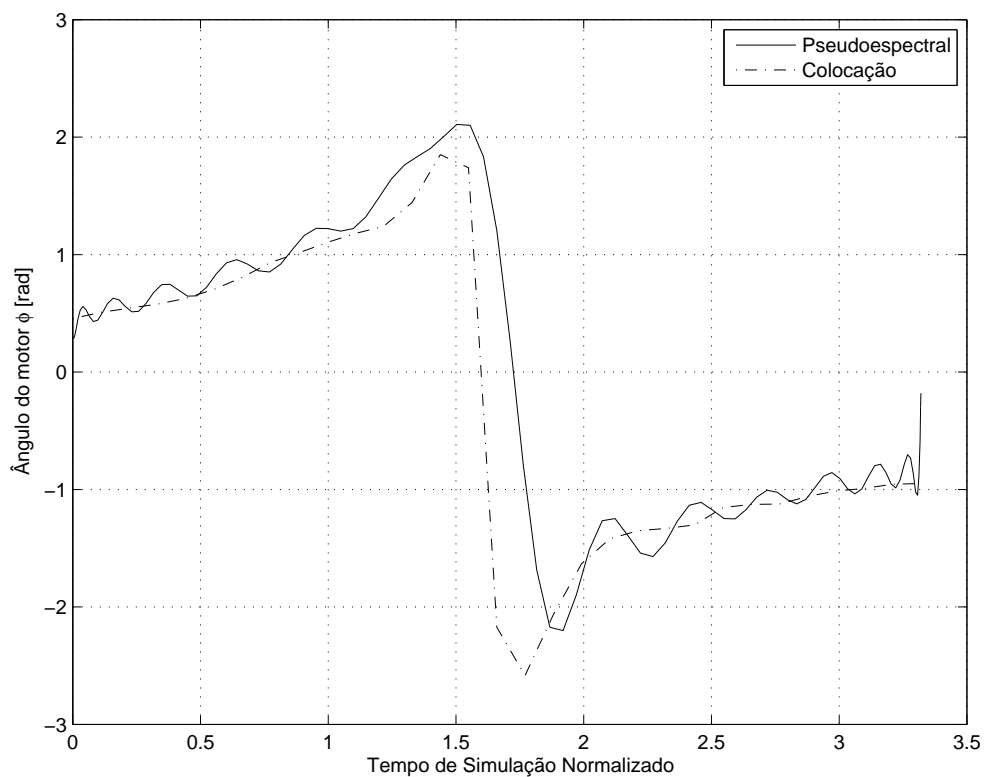


FIGURA 3.8: Ângulo do Sistema propulsivo vs Tempo de Simulação para $N = 30$ nós

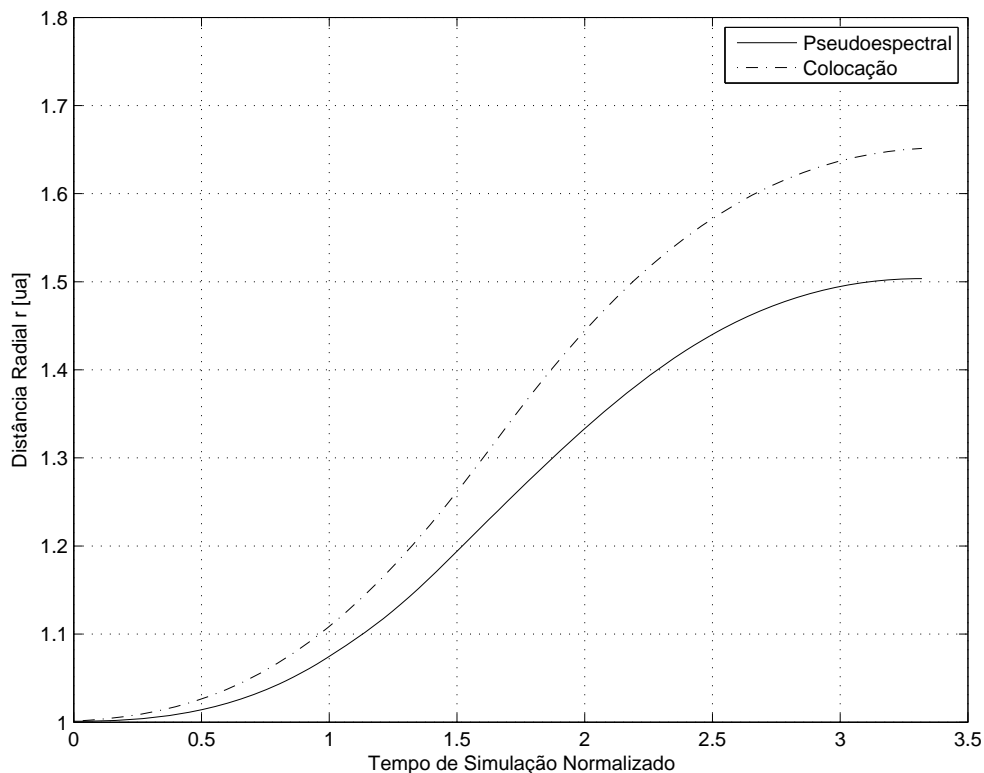


FIGURA 3.9: Distância Radial vs Tempo de Simulação para $N = 50$ nós

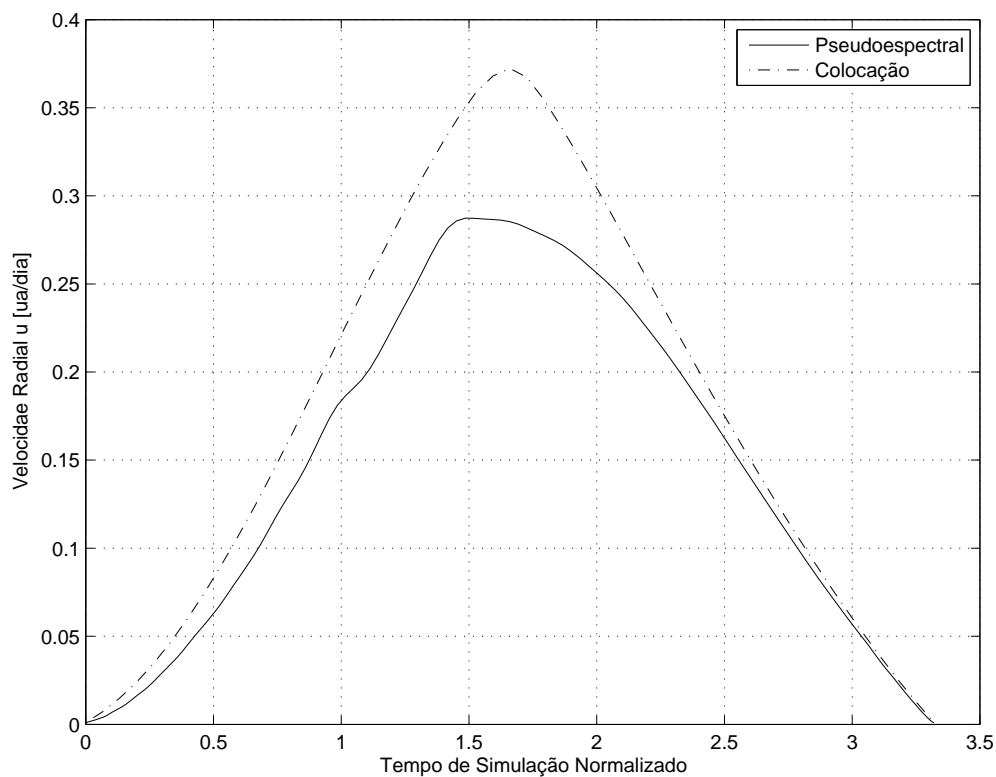


FIGURA 3.10: Velocidade Radial vs Tempo de Simulação para $N = 50$ nós

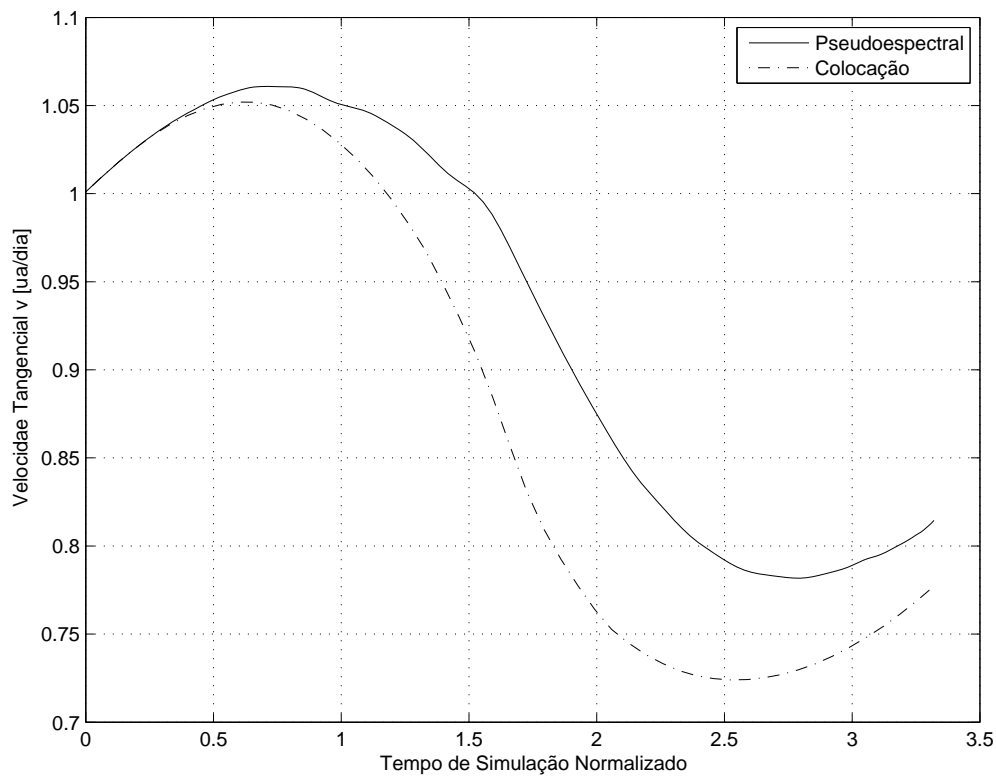
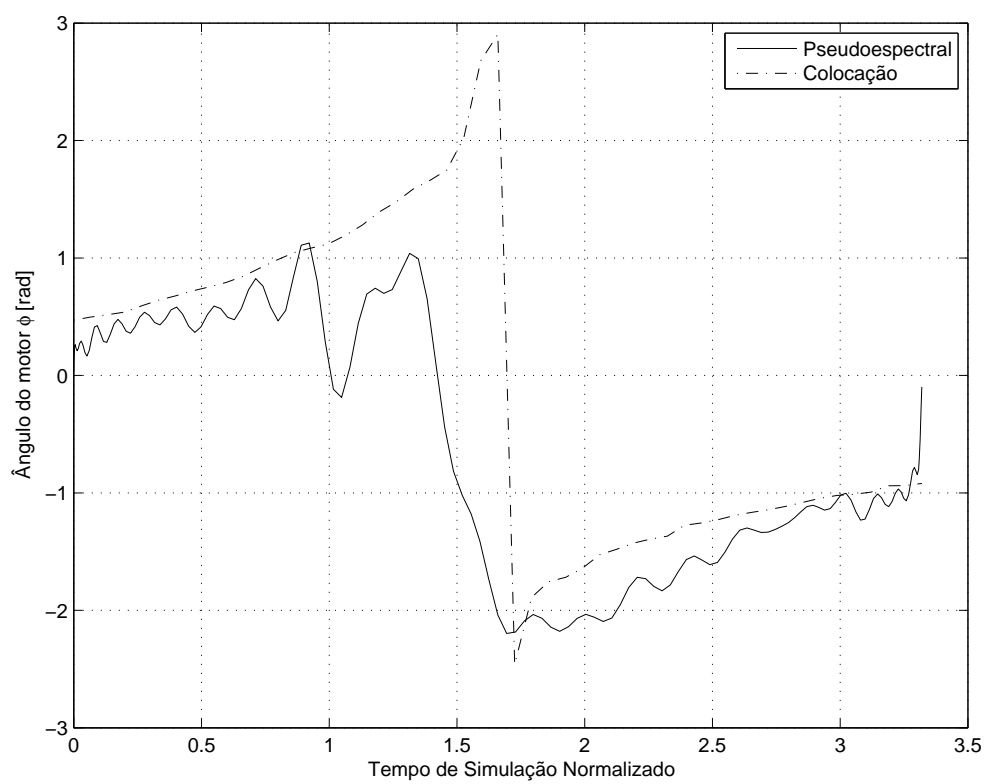


FIGURA 3.11: Velocidade Tangencial vs Tempo de Simulação para $N = 50$ nós

FIGURA 3.12: Ângulo do Sistema propulsivo vs Tempo de Simulação para $N = 50$ nós

Capítulo 4

Problema de Navegação e Controlo 4D

Vai ser abordado o problema da navegação e controlo a quatro dimensões, que é um tema ainda por explorar. O que até agora tem vindo a ser feito no decorrer deste estudo, é compreender e explorar, conceitos e teoria que é de extrema importância na resolução deste problema.

Foi feita uma revisão geral, no Capítulo 2, sobre o funcionamento do processo de optimização e as capacidades da programação não linear. Já no Capítulo 3 obteve-se conhecimento sobre controlo óptimo ou optimização de trajectória, que recorre à programação linear para ser resolvido. A optimização de trajectória, é então, a única ferramenta capaz de ajudar a encontrar trajectórias óptimas num problema de navegação, pois é a maneira pela qual se consegue formular e resolver tal problema.

O interesse da navegação e controlo de trajectórias 4D é vasto, pois o projecto de uma trajectória 4D dá a certeza, que para um determinado *waypoint*, a aeronave, veículo espacial vai passar neste no tempo especificado. Assim, consegue-se garantir um trajecto entre *waypoints* óptimo, o que no problema 3D não é tão fácil porque a trajectória nominal satisfaz apenas o ponto de chegada. Uma trajectória que satisfaz um ponto inicial e um ponto final pode ter não quer dizer que cumpra a distância Euclidiana ou Riemanniana, pois existem uma infinidade de possibilidades que cumprem estas especificações.

A ideia subjacente ao projecto de trajectórias 4D é que especificando o tempo de chegada a um determinado *waypoint*, o veículo aeroespacial vai controlar a velocidade, até onde

for possível, para que cumpra o tempo de chegada. Este facto é de extrema importância na medida em que o trajecto entre dois *waypoints* tende para a menor distância¹.

Nos itens seguintes vai ser modelado e resolvido o problema de navegação e o problema de controlo, aplicando este a dois exemplos práticos. Numa primeira fase irá ser formulado o problema de navegação 4D em seguida será modelado segundo a abordagem da optimização de trajectória e aplicado métodos directos para resolvê-lo. Resolvido o problema de navegação irá ser aplicado um tipo de controlo predictivo de passo único.

4.1 Modelação Geral do Problema de Navegação 4D

Na formulação do problema de Navegação tem que se ter em conta qual é a abordagem que é necessário utilizar para resolve-lo. Este problema pode ser resolvido tendo em conta a dinâmica do veículo aeroespacial ou apenas utilizar as equações de navegação e deixar a dinâmica para o controlo. A segunda hipótese é mais aliciante pois, nesta abordagem, a trajectória nominal é calculada tendo em conta apenas e só o modelo de navegação poupando assim muito tempo de processamento. Caso contrário iria-se ter muitas mais variáveis podendo comprometer a convergência para uma solução óptima.

O que se pretende neste estudo é separar a Navegação do Controlo, pois estes podem ser totalmente independentes, assim é calculada uma trajectória previamente e depois injectada no controlador para que este possa seguir a trajectória.

A grande questão neste momento é saber se a trajectória de referência calculada pode infasível para uma determinada aeronave ou veículo espacial. Esta questão irá ser contornada com veremos na formulação do problema.

4.1.1 Formulação do Problema

Seja o ponto P definido na referência geocêntrica (x, y, z) e um conjunto de pontos $P_i = P_0, P_1, \dots, P_n$, em que n é o número de pontos. Se τ_i for o tempo de chegada a cada ponto, podemos então definir um *waypoint* 4D como,

$$(P_i, \tau_i) \rightarrow (P_i, [\tau_i^1, \tau_i^2]) \quad (4.1)$$

sendo $[\tau_i^1, \tau_i^2]$ o intervalo de tempo permitido para que se chegue a um determinado *waypoint*.

¹Pode ser Euclideana o Riemanniana ,ou outra qualquer, dependendo do modelo utilizado

Se $s(\tau)$ for a posição da aeronave, então o problema de navegação 4D está resolvido quando,

$$\|P_i - s(\tau_i)\| \leq \epsilon \quad \forall i \in \mathfrak{R}^n \quad (4.2)$$

em que ϵ é distância máxima a que aeronave pode sobrevoar um determinado *waypoint*.

A inequação (4.2) pode ainda ser transformada numa igualdade,

$$\|P_i - s(\tau_i)\|^2 = \epsilon^2 \quad \forall i \in \mathfrak{R}^n \quad (4.3)$$

4.1.2 Modelo Dinâmico

O modelo dinâmico de apresenta duas possibilidades, um na referência geocêntrica outro na referência geodética.

Seja $P_i = (x_i, y_i, z_i)$ um determinado ponto na referência geocêntrica e $P_i = (\lambda_i, \varphi_i, z_i)$ um ponto na referência geodética, então, na referância geocêntrica temos o seguinte modelo de estado,

$$\dot{x} = V \cos \gamma \cos \psi \quad (4.4)$$

$$\dot{y} = V \cos \gamma \sin \psi \quad (4.5)$$

$$\dot{z} = V \sin \gamma \quad (4.6)$$

$$\dot{V} = u_1 \quad (4.7)$$

$$\dot{\gamma} = u_2 \quad (4.8)$$

$$\dot{\psi} = u_3 \quad (4.9)$$

já na referência geodética basta transformar as equações (4.4),(4.5) e (4.6) para esta referência e o modelo de estado continua igual, assim,

$$\dot{\lambda} = \frac{V \cos \gamma \cos \psi}{(h + Re) \cos \varphi} \quad (4.10)$$

$$\dot{\varphi} = \frac{V \cos \gamma \sin \psi}{(h + Re)} \quad (4.11)$$

$$\dot{h} = V \sin \gamma \quad (4.12)$$

em que $\hat{x} = [x, y, z, V, \gamma, \psi]$ ou $\hat{x} = [\lambda, \varphi, h, V, \gamma, \psi]$ são o vector de estado, $u = [u_1, u_2, u_3]$ o vector de controlo e Re é o raio da terra.

Este modelo dinâmico conta com equações de navegação e equações muito simples que retratam no fundo a dinâmica da aeronave. O problema de se obter uma trajectória que à posteriori a aeronave não a consiga realizar, é resolvido quando se limita a velocidade V , o ângulo de trajectória γ e as taxas \dot{V} , $\dot{\gamma}$ e $\dot{\psi}$ de acordo com a dinâmica da aeronave a que a trajectória se destina, assim,

$$\begin{aligned} u_i &\in [u_i^{min}, u_i^{max}] \\ \dot{V} &\in [V^{min}, V^{max}] \\ \dot{\gamma} &\in [\gamma^{min}, \gamma^{max}] \end{aligned}$$

A escolha deste modelo dinâmico é devido ao facto de este conter apenas apenas as equações da navegação, permitindo ainda, fazer das variações de Velocidade, Ângulo de Trajectória e Rumos controlos, assim é possível obter Velocidades, Ângulos de Trajectória e Rumos de referência.

4.2 Resolução do Problema de Navegação 4D

Com modelo da dinâmica da navegação, é preciso então, formular o problema em termos de controlo óptimo para que depois seja possível aplicar um método de procura de trajectórias óptimas.

Seja $w_{p_i} \rightarrow (P_i, \tau_i)$ uma sequência de *waypoints* pré-definidos, w_{p_0} e w_{p_f} o *waypoint* inicial e final, respectivamente, então o quer-se para uma determinada missão minimizar o atraso no *waypoint* final w_{p_f} . Se $s(\tau)$ for uma trajectória no tempo e espaço contínuo,

$$s(\tau) = \begin{pmatrix} \lambda(\tau) \\ \varphi(\tau) \\ h(\tau) \end{pmatrix} \quad (4.13)$$

o Índice de Performance J para este problema aparece com a seguinte expressão,

$$J = \|P_f - s(\tau_f)\|_Q^2 \quad (4.14)$$

em que τ_f é o tempo pretendido para a duração da trajectória e Q é uma matriz positivamente definida.

Seja $\dot{x} = f(x, u)$ as equações de estado, x o vector de estado e u o vector de controlo são definidos por,

$$x = \begin{bmatrix} \lambda \\ \varphi \\ h \\ V \\ \gamma \\ \psi \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

então o problema de optimização de trajectória aparece formulado da seguinte maneira,

$$\min_u J(u) = \|P_f - s(\tau_f)\|_Q^2 \quad (4.15)$$

sujeito a,

$$\dot{x} = f(x, u) = 0 \quad (4.16)$$

$$\|P_i - s(\tau_i)\| \leq \epsilon \quad (4.17)$$

$$u_i^{\min} \leq u_i \leq u_i^{\max} \quad (4.18)$$

$$V^{\min} \leq V \leq V^{\max} \quad (4.19)$$

$$\gamma^{\min} \leq \gamma \leq \gamma^{\max} \quad (4.20)$$

4.2.1 Trajectórias Óptimas

Tendo o problema de navegação descrito num problema de optimização de trajectória, é então necessário aplicar um método de optimização de trajectórias. Como já foi referido existem dois tipos de métodos para resolver o problema de controlo óptimo, os métodos indirectos e os métodos directos. Os métodos indirectos acentam no princípio do mínimo de Pontryagin, a sua implementação para problemas complexos não de todo apropriado pelas razões que já foram explicadas. Por isto, optou-se por aplicar os métodos directos a este problema, estes resolvem o problema de optimização de trajectória transformando-o num problema de optimização de parâmetros e através da programação não linear encontram uma solução. O problema agora é saber qual quais os esquemas de integração utilizados para descrever as equações diferenciais de estado. Foi visto que existem vários tipos de destes métodos mas os que se destacam são os esquemas implícitos, e actualmente os métodos pseudoespectrais.

4.2.1.1 Trajectórias Óptimas Segundo Métodos Directos

Será transcrito para um problema de programação linear o problema de optimização de trajectória com base nos métodos da Colocação com integração trapezoidal e Pseudoespectral utilizando os polinómios de Chebyshev.

O problema de optimização de trajectória definido, anteriormente para o problema de navegação 4D, continua a ser o mesmo independentemente do esquema de integração utilizado, mudando apenas o modo como é feita a parametrização.

Para uma dada missão, o cálculo da trajectória deve começar por definir correctamente os *waypoints* que dado veículo aeroespacial percorra, definindo coordenadas e especificando o tempo de chegada a cada *waypoint*. Deve-se ter em atenção que a especificação do tempo de chegada e a própria localização dos *waypoints* deve ser tal que o veículo aeroespacial seja capaz de cumprir, ou seja, deve-se ter em conta a dinâmica do veículo quando se especifica um determinado *waypoint* (e.g. não definir tempos de chegada que o veículo não consiga realizar devido à velocidade de cruzeiro).

Definida a missão, o procedimento seguinte será transcrever o problema de navegação 4D para programação não linear, que irá ser feito de seguida.

Método da Colocação

Dado um conjunto de *waypoints* $w_{p_i} \rightarrow (P_i, \tau_i)$, com $i = 1, 2, \dots, M$. Dividindo a trajectória em N segmentos teremos $(N + 1)$ nós. Se τ_i for o tempo chegada a cada *waypoints*, τ_0 o tempo inicial, τ_f o tempo final e τ_k o tempo em cada nó, temos,

$$\tau_i \subseteq \tau_k \in [\tau_0, \tau_f]$$

podendo acontecer que os dois conjuntos sejam coincidentes,

$$\tau_i = \tau_k$$

As variáveis de programação não linear serão o conjunto das variáveis de estado x e controlo u ao longo de cada nó, assim define-se y como as variáveis de programação não linear,

$$y = [u_0, x_0, u_1, x_1, \dots, u_N, x_N]$$

Tendo isto, basta discretizar as equações de estado com base no método da colocação e teremos o problema de optimização de trajectória para navegação 4D formulado em termos de programação não linear,

$$\min_y J(y) = \|(\lambda_M, \varphi_M, h_M) - (\lambda(\tau_N), \varphi(\tau_N), h(\tau_N))\|_Q^2 \quad (4.21)$$

sujeito a,

$$x_{k+1} - x_k - [f(x_{k+1}, u_{k+1}) + f(x_k, u_k)] = 0 \quad (4.22)$$

$$\|P_i - s(\tau_i)\| \leq \epsilon \quad (4.23)$$

$$u_j^{min} \leq u_j \leq u_j^{max} \quad (4.24)$$

$$V^{min} \leq V \leq V^{max} \quad (4.25)$$

$$\gamma^{min} \leq \gamma \leq \gamma^{max} \quad (4.26)$$

com $k = 0, 1, \dots, N$

Método Pseudoespectral

O método pseudoespectral utilizando polinómios de Chebyshev, possui diferenças em relação aos métodos típicos que derivam dos esquemas de Runge-Kutta. A primeira grande diferença é que os nós têm que ser construídos segundo os nós de Chebyshev t_k , outra diferença é que para se cumprir a passagem nos diferentes *waypoints* terá de

ser calculado o t_k correspondente a τ_i e interpolar a posição através dos polinómios de Chebyshev.

Assim, seja N o número de segmentos em que se divide uma trajectória, os nós de Chebyshev são definidos pela seguinte expressão,

$$t_k = \cos\left(\frac{k\pi}{N}\right), \quad k = 0, 1, \dots, N \quad (4.27)$$

Como foi visto no Capítulo 3 os polinómios de Chebyshev $T(t)$ são definidos para o intervalo $t \in [-1, 1]$ e para cacular τ correspondente a t tem-se,

$$\tau = \frac{(\tau_f - \tau_0)t + (\tau_f + \tau_0)}{2} \quad (4.28)$$

Dadas as variáveis de programação não linear,

$$y = [u_0, x_0, u_1, x_1, \dots, u_N, x_N]$$

definidas nos nós de Chebyshev t_k .

A trajectória contínua no tempo $s(\tau)$, pode ser definida para os nós t_k fazendo corresponder o tempo verdadeiro ao tempo no espaço de Chebyshev $\tau \rightarrow t$, assim a trajectória $s(t)$ vem,

$$s(t) = \begin{pmatrix} \lambda(t) = \sum_{k=0}^N \lambda(t_k) \bar{\varphi}_k(t) \\ \varphi(t) = \sum_{k=0}^N \varphi(t_k) \bar{\varphi}_k(t) \\ h(t) = \sum_{k=0}^N h(t_k) \bar{\varphi}_k(t) \end{pmatrix} \quad (4.29)$$

onde $\bar{\varphi}_k(t)$ são os polinómios de Lagrange contruidos nos nós de Chebyshev (ver Capítulo 3).

A formulação do problema de navegação em termos de programação não linear utilizando o método pseudoespectral de Chebyshev descreve-se do seguinte modo,

$$\min_y J(y) = \|(\lambda_M, \varphi_M, h_M) - (\lambda(t_N), \varphi(t_N), h(t_N))\|_Q^2 \quad (4.30)$$

sujeito a,

$$\left(\frac{\tau_f - \tau_0}{2}\right) f(x_k, u_k) - \sum_{l=0}^N C_{kl} x(t_l) = 0 \quad (4.31)$$

$$\|P_i - s(t_i)\| \leq \epsilon \quad (4.32)$$

$$u_j^{min} \leq u_j \leq u_j^{max} \quad (4.33)$$

$$V^{min} \leq V \leq V^{max} \quad (4.34)$$

$$\gamma^{min} \leq \gamma \leq \gamma^{max} \quad (4.35)$$

com $k = 0, 1, \dots, N$ e t_i é o t correspondente aos *waypoints*.

4.3 Controle de Trajectórias de Referência

Resolvido o problema de navegação 4D, pretende-se controlar uma aeronave ao longo da trajectória calculada. Existem alguns métodos que permitem que uma aeronave siga uma determinada trajectória pré-definida, entre os quais o denominado *controle predictivo*.

4.3.1 Controle Preditivo

O controle predictivo recorre ao modelo não linear da dinâmica de um determinado sistema, pelo que é necessário que tal modelo esteja incorporado na própria aeronave. Isto permite, ao contrário de do controle não linear, seguir qualquer tipo de trajectória (dentro dos limites dinâmicos e estruturais).

4.3.1.1 Teoria do Controle Predictivo de passo Único

Dada as equações dinâmicas para um sistema não linear, com a seguinte forma,

$$\dot{x}_1 = f_1(x) \quad (4.36)$$

$$\dot{x}_2 = g(x, u) \quad (4.37)$$

onde $x_1 \in \mathfrak{R}^{n_1}$ são as varáveis do primeiro sistema, $x_2 \in \mathfrak{R}^{n_2}$ as varáveis do segundo sistema, $n_1 + n_2 = n$ a dimensão total dos dois sistemas e $u \in U \subset \mathfrak{R}^m$ o conjunto das varáveis de controlo em que U é o conjunto compacto que define o domínio de controlos factíveis. Assim, temos $x = (x_1 \ x_2) \in \mathfrak{R}^n$ como sendo o vector de estado do sistema.

A equação (4.37) pode-se desenvolver para que se torne mais simples,

$$\begin{aligned}
 \dot{x}_2 &= g(x, u) \\
 &= g(x, u_k) + B(x)\bar{u} \\
 &= g(x, u_k) + B(x)(u - u_k) \\
 &= f_2(x) + B(x)u
 \end{aligned} \tag{4.38}$$

em que u_k é o valor do vector de controlo para um dado tempo τ e $B(x) = \left(\frac{\partial g(x, u)}{\partial u} \right)_{u=u_k}$.

Uma particularidade deste sistema é que f_1 é menos duas vezes diferenciável, f_2 e B são diferenciáveis com gradientes contínuos. O vector de estado parcial x_2 e o vector de controlo u são assumidos como sendo condições de fronteira,

$$\begin{aligned}
 u_{min} &\leq u \leq u_{max} \\
 x_{2min} &\leq x_2 \leq x_{2max}
 \end{aligned} \tag{4.39}$$

Pode existir, na eventualidade, restrições de desigualdade $c(\cdot)$ associadas ao vector de estado parcial x_2 e controlo u .²

$$\begin{aligned}
 c(x_2) &\leq 0 \\
 c(u) &\leq 0
 \end{aligned} \tag{4.40}$$

Supondo que a trajectória de referência $s(t) \in \mathfrak{R}^{n_1}$ dada pelo vector de estado parcial x_1 , é conhecida no intervalo de tempo $[\tau_0, \tau_f]$. Esta trajectória de referência pode ser uma trajectória tempo-espaço bem como de estados fixos. O proposto método de controlo predictivo tem como objectivo determinar os valores do vector de controlo u que permitam cumprir a trajectória nominal $s(t)$.

Tendo agora em atenção, um dado instante de tempo, arbitrário $\tau_k \in [\tau_0, \tau_f]$, o estado conhecido $x_k \equiv x(\tau_k)$ e considerando que $x_{k+1} \equiv x(\tau_{k+1}) = x(\tau_k + \delta_\tau)$ (onde δ_τ é o incremento de tempo) proceda-se à expansão em segunda ordem da série de Taylor sobre valor actual de x_{1k} do vector de estado parcial x_1 e uma expansão em primeira

²As condições de fronteira também podem ser entendidas como restrições de desigualdade

ordem sobre valor actual x_{2_k} do vector de estado parcial x_2 . Isto acontece porque o grau relativo da equação (4.36) é um, já o da equação (4.37) é nulo.

$$x_{1_{k+1}} = x_{1_k} + \delta_\tau f_1(x_k) + \frac{\delta_\tau^2}{2} [F_1 f_1(x_k) + F_2(f_2(x_k) + B(x_k)u_k)] \quad (4.41)$$

$$x_{2_{k+1}} = x_{2_k} + \delta_\tau (f_2(x_k) + B(x_k)u_k) \quad (4.42)$$

em que $F_1 = \left(\frac{\partial f_1(x)}{\partial x_1} \right)_{x=x_k}$ e $F_2 = \left(\frac{\partial f_1(x)}{\partial x_2} \right)_{x=x_k}$ são os *Jacobianos* da matriz f_1 relativo a x_1 e x_2 , respectivamente, em $x = x_k$.

Seja x_1^{ref} e x_2^{ref} os valores de referência dos respectivos estados em $\tau_{k+1} = \tau_k + \delta_\tau$, então para τ_{k+1} o erro dos estados $x_{1_{k+1}}$ e $x_{2_{k+1}}$ em relação à trajectória nominal é dado por,

$$\begin{aligned} e_{1_{k+1}} &= x_1^{ref} - x_{1_{k+1}} \\ e_{2_{k+1}} &= x_2^{ref} - x_{2_{k+1}} \end{aligned} \quad (4.43)$$

com isto é possível formular o problema de controlo num problema de minimização quadrática, que pretende encontrar o controlo u capaz de minimizar a diferença entre a trajectória nominal e a calculada. Numa outra notação,

$$\begin{aligned} \min_{u, \tau} J(u, \tau) &= \frac{1}{2} \|x_1^{ref} - x_{1_{k+1}}\|_{Q_1}^2 + \frac{1}{2} \|x_2^{ref} - x_{2_{k+1}}\|_{Q_2}^2 \\ &= \frac{1}{2} e_{1_{k+1}}^T Q_1 e_{1_{k+1}} + \frac{1}{2} e_{2_{k+1}}^T Q_2 e_{2_{k+1}} \end{aligned} \quad (4.44)$$

sujeito às restrições,

$$\delta_{\tau_{min}} \leq \delta_\tau \leq \delta_{\tau_{max}} \quad (4.45)$$

$$c(u) \leq 0 \quad (4.46)$$

$$c(x_2) \leq 0 \quad (4.47)$$

onde Q_1 e Q_2 são matrizes positivamente definidas.

O erro médio entre a trajectória de referência $s^{ref}(\tau)$ e a trajectória verdadeira $s(\tau)$ é dado pelo *RMSE* (*Root Mean Square Error*), este representa desvio médio que a aeronave faz em relação à trajectória de referência e calcula-se através da seguinte expressão,

$$RMSE = \left(\frac{1}{\tau_f - \tau_0} \int_{\tau_0}^{\tau_f} \|s^{ref}(\tau) - s(\tau)\|^2 dt \right)^{1/2} \quad (4.48)$$

4.3.1.2 Controlo de Trajectórias 4D

Para aplicar o controlo predictivo a um determinado veículo, em primeiro lugar terá de se especificar qual o modelo dinâmico pelo qual se rege. No caso das aeronaves, que é onde vai ser aplicado a este tipo de controlo no decorrer deste estudo é,

$$\dot{V} = \frac{\delta_T T_{max} \cos(\alpha + \epsilon_T) - D}{m} - g \sin \gamma \quad (4.49)$$

$$\dot{\gamma} = \frac{L + \delta_T T_{max} \sin(\alpha + \epsilon_T)}{mV} \cos \phi - \frac{g}{V} \cos \gamma \quad (4.50)$$

$$\dot{\psi} = \frac{L + \delta_T T_{max} \sin(\alpha + \epsilon_T)}{mV \cos \gamma} \cos \phi \quad (4.51)$$

$$\dot{\phi} = p + (q \sin \phi + r \cos \phi) \tan \theta \quad (4.52)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (4.53)$$

$$\dot{p} = \frac{1}{I_x I_y - I_{xz}^2} [I_z (Q S b C_l + (I_y - I_z) q r) + I_{xz} (Q S b C_n + (I_x - I_y + I_z) q p - I_{xz} q r)] \quad (4.54)$$

$$\dot{q} = \frac{1}{I_y} [Q S c C_m + I_{xz} (r^2 - p^2) + (I_z - I_x) z p] \quad (4.55)$$

$$\dot{r} = \frac{1}{I_x I_z - I_{xz}^2} [I_x (Q S b C_n + (I_x - I_z) p q) + I_{xz} (Q S b C_l + (I_y - I_x + I_z) q z - I_{xz} p q)] \quad (4.56)$$

$$\theta = \alpha + \gamma \quad (4.57)$$

No caso de uma aeronave a aplicação deste tipo de controlo tem de ser faziada, denominado controlo em *cascata*, assim na primeira fase temos,

$$x_1 = \begin{pmatrix} \lambda \\ \varphi \\ h \end{pmatrix}, \quad x_2 = \begin{pmatrix} V \\ \gamma \\ \psi \end{pmatrix}, \quad u = \begin{pmatrix} \theta \\ \phi \\ \delta_e \\ \delta_T \end{pmatrix}$$

em que θ e ϕ são denominados controlos primários, δ_e e δ_T controlos secundários.

Do processo de optimização de trajectória tem-se,

$$x_1^{ref} = \begin{pmatrix} \lambda^{ref} \\ \varphi^{ref} \\ h^{ref} \end{pmatrix}, \quad x_2^{ref} = \begin{pmatrix} V^{ref} \\ \gamma^{ref} \\ \psi^{ref} \end{pmatrix},$$

Aplicando o processo de optimização da equação (4.44) é obtido o controlo u capaz de satisfazer a restrições descritas. Na segunda fase do controlo predictivo, θ e ϕ do controlo u obtido anteriormente passa a ser uma referência, pelo que os novos x_1 e x_2 são,

$$x_1 = \begin{pmatrix} \theta \\ \phi \end{pmatrix}, \quad x_2 = \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \quad u = \begin{pmatrix} \delta_a \\ \delta_r \end{pmatrix}$$

$$x_1^{ref} = \begin{pmatrix} \theta^{ref} \\ \phi^{ref} \end{pmatrix}, \quad x_2^{ref} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

onde as taxas são nulas para garantirem as condições de *quasi-equilibrium*.

O processo de optimização para obter os controlos, quer secundários como primários, tem de ser feito através de programas computacionais, ora, como já foi referido para aplicar este tipo de controlo a uma aeronave é necessário que ela contenha no controlador de voo o seu próprio modelo dinâmico. Para que sejam feitos estes dois processos de optimização em tempo útil é necessário que o sistema seja baseado numa linguagem de programação rápida e o algoritmo de optimização seja ele também rápido. O mesmo acontece com a simulação, se nesta não forem utilizados algoritmos de resolução de problemas de programação não linear, rápidos e arquitectados em linguagens rápidas também o tempo de simulação poderá ser demorado.

4.4 Aplicação

Estudo o modo como se estrutura o processo de optimização de trajectória para o problema de navegação 4D, vai ser aplicado a dois tipos de missão, a primeira, uma missão directa entre dois *waypoints* com variação da altitude, típica de um voo comercial. A segunda missão é em circuito para provar as capacidade dos métodos de optimização de trajectória pois a dificuldade de obter uma trajectória óptima é maior. Nesta última missão ainda será aplicado o controlo a uma porção da trajectória, pois a ferramenta computacional não permite para mais.

Em ambos os exemplos vai ser utilizado o *MatLab* tanto para obter as trajectórias óptimas através do programa *fmincon* da **Optimization Toolbox**.

As duas aplicações serão aplicadas a uma aeronave não tripulada, *SkyGu@rdian* existente no Departamento de Ciências Aeroespaciais da Universidade da Beida Interior cujas características estão no Apêndice A.

No problema de navegação 4D trabalhar-se-à na referência geodética com distâncias em quilómetros e e velocidades em quilómetros por hora, isto para que haja uma redução de escala e a convergência se torne possível e mais rápida, caso contrário, como levaria-se muito tempo de simulação podendo mesmo acontecer que não houvesse convergência para uma trajectória óptima.

4.4.1 *Exemplo I*

Nesta primeira aplicação será realizada uma missão entre Castelo Branco, Fundão e Covilhã num voo típico de uma aeronave comercial.

4.4.1.1 Plano de Voo

Na tabela 4.1 está descrito o plano de voo, com todos os *waypoints* que aeronave terá de percorrer, especificando posição e tempo de chegada em cada um deles. Os tempos de chegada a cada waypoint, bem como as altitudes para são escolhidas tendo sempre em conta as restrições dinâmicas e estruturais do UAV SkyGu@dian.

Os *waypoints* 1, 11 e 14 da tabela 4.1 são Castelo Branco, Fundão e Covilhã, respectivamente, os restantes são para ajudar a delinear o trajecto que se pretende que a aeronave faça.

Nº	Longitude λ	Latitude φ	Altura h	Tempo de Chegada τ	
	[deg min seg]	[deg min seg]	[m]	[horas]	
1	7° 29' 35.00" W	39° 49' 25.71" N	400	0.000	Castelo Branco
2	7° 29' 37.00" W	39° 50' 34.82" N	500	0.035	
3	7° 29' 39.00" W	39° 51' 33.38" N	600	0.070	
4	7° 29' 41.00" W	39° 52' 39.86" N	600	0.080	
5	7° 29' 41.50" W	39° 54' 50.26" N	700	0.120	
6	7° 29' 42.00" W	39° 56' 55.38" N	800	0.165	
7	7° 29' 45.00" W	39° 59' 15.04" N	800	0.210	
8	7° 29' 47.00" W	40° 01' 17.12" N	800	0.245	
9	7° 29' 49.00" W	40° 03' 45.92" N	800	0.280	
10	7° 29' 51.00" W	40° 05' 31.38" N	800	0.325	
11	7° 29' 53.00" W	40° 08' 12.56" N	800	0.370	Fundão
12	7° 29' 55.00" W	40° 11' 06.43" N	750	0.415	
13	7° 30' 00.00" W	40° 14' 07.43" N	650	0.450	Covilhã
14	7° 30' 02.00" W	40° 17' 02.02" N	600	0.480	

TABELA 4.1: Coordenadas dos Waypoints entre Castelo Branco e Covilhã

4.4.1.2 Trajectória Óptima

O processo de optimização de trajectória vai ser feito através de dois métodos directos, o método da Colocação com integração trapezoidal e o método pseudoespectral utilizando polinómios de Chebyshev. Os esquemas de Runge-Kutta não foram utilizados, pois o método da Colocação deriva destes e o facto de ser um método implícito dá uma solução mais refinada.

Para este exemplo os dois métodos convergiram para uma solução é necessário desde logo especificar as condições de fronteira das variáveis de estado e controlo (tabela 4.2), ou se quisermos das variáveis de programação não linear.

Como se viu na formulação do problema 4D, as restriões nos *waypoints* são obrigatórias (equação (4.17)) pelo que a estimativa inicial não precisa de ser muito próxima da solução, se for, tanto melhor pois a convergência é mais rápida. O que se faz na prática é arrancar as variáveis no meio das suas condições de fronteira.

	Longitude λ [rad]	Latitude φ [rad]	Altura h [km]	Velocidade V [km/h]	Ângulo de Trajectória γ [rad]	Rumo ψ [rad]	u_1 [km/h ²]	u_2 [rad/h ²]	u_3 [rad/h ²]
Mínimo	$-\pi$	$-\frac{\pi}{2}$	0.4	64.8	-0.175	-2π	-25920	-200 π	-600 π
Maximo	π	$\frac{\pi}{2}$	1.8	108.0	0.175	2π	25920	200 π	600 π

TABELA 4.2: Condições de fronteira do vector de estado e de controlo

	$\lambda(\tau_f)$	$\varphi(\tau_f)$	$h(\tau_f)$	J
	[m]	[m]	[km]	
Colocação	-0.1309	0.7031	0.5998	2.8×10^{-8}
Pseudoespectral	-0.1309	0.6999	0.5058	3.7×10^{-4}

TABELA 4.3: Valores das variáveis de estado no final da trajectória

Relativamente ao número de nós ($N+1$), não existe nenhum número que se deva utilizar, pois esta temática ainda é um campo por explorar. O facto é, que o número de nós é implicado pelo método utilizado e pela quantidade de variáveis de programação não linear que o meio computacional aguenta. O que se fez neste estudo, quanto ao método da Colocação, foi definir o número de nós igual ao número de *waypoints*. O que se fez na realidade foi diminuir a escala do problema, passar para a referência geodética e trabalhar com unidades de maior escala, assim conseguiu-se ter o tempo em horas que faz com que o passo no tempo seja pequeno não afectando o método de integração. Quanto ao método pseudo espectral como o problema é tratado num outro espaço, o tempo está parametrizado e os nós só coincidem com os *waypoints* por mero acaso. Então o número de nós ($N+1$) utilizado foi 20 que foi a partir do momento que houve convergência para uma solução.

O valor do *Índice de Performance* ou função de custo representado na tabela 4.3 é praticamente igual nos dois métodos. As restrições também elas são cumpridas pelo que nas figuras (4.1), (4.2) e (4.3) mostram que para os dois métodos a trajectória passa nos waypoints pré-definidos. Apesar de na figura (4.1) parecer que para o método pseudoespectral existe bastante oscilação, em relação ao método da Colocação é verdade, mas projectada a trajectória no terreno (figuras (4.4) e (4.5) utilizando o *GoogleEarth* consta-se que elas são quase coincidentes (4.6).

Relativamente ao comportamento da Velocidade (figura (4.7)) ela coincide praticamente quer num método quer no outro, o Ângulo de trajectória (figura (4.8)) também tem uma resposta temporal muito parecida na aplicação dos dois métodos diferenciando-se apenas no facto que o método pseudoespectral proporciona uma variação mais suave, o Rumo por sua vez apresenta uma resposta (4.9) com uma diferença de amplitude de 0.5º que não é significativa visto que a trajectória é cumprida nos dois métodos. Quanto aos controlos (figuras (4.10), (4.11) e (4.12)) a única situação a referir é que com o método pseudo espectral as suas respostas são mais suaves.

O facto de o método pseudoespectral possuir respostas mais suaves é importante pois num voo não se quer picos de movimentos quanto mais se suavizar uma trajectória mas probabilidades se tem para fazer um voo seguro.

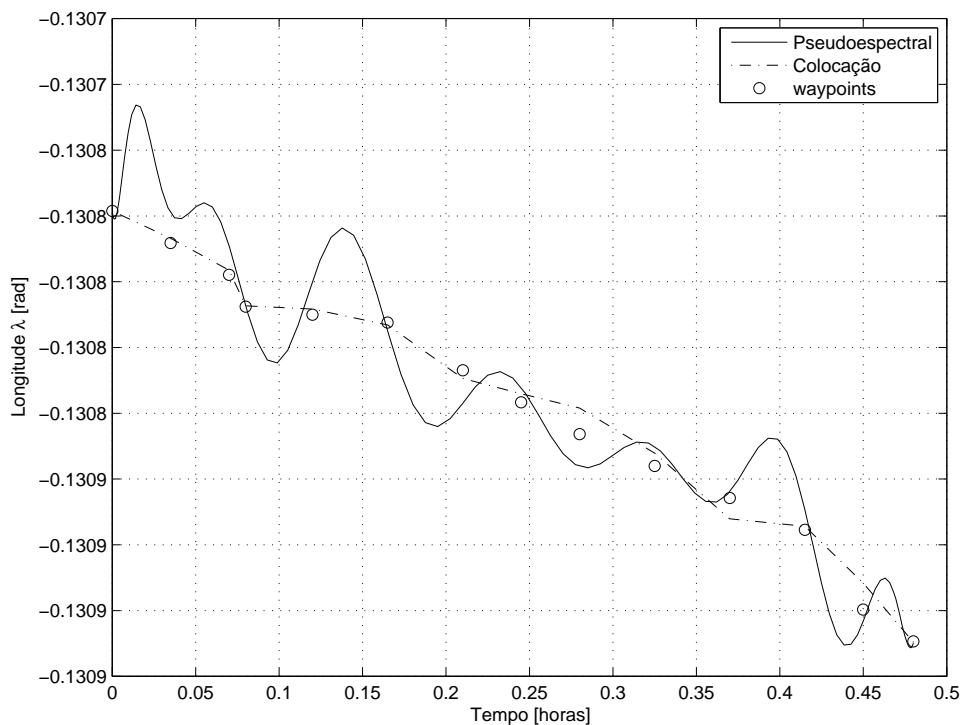


FIGURA 4.1: Comportamento da Longitude ao longo do Tempo

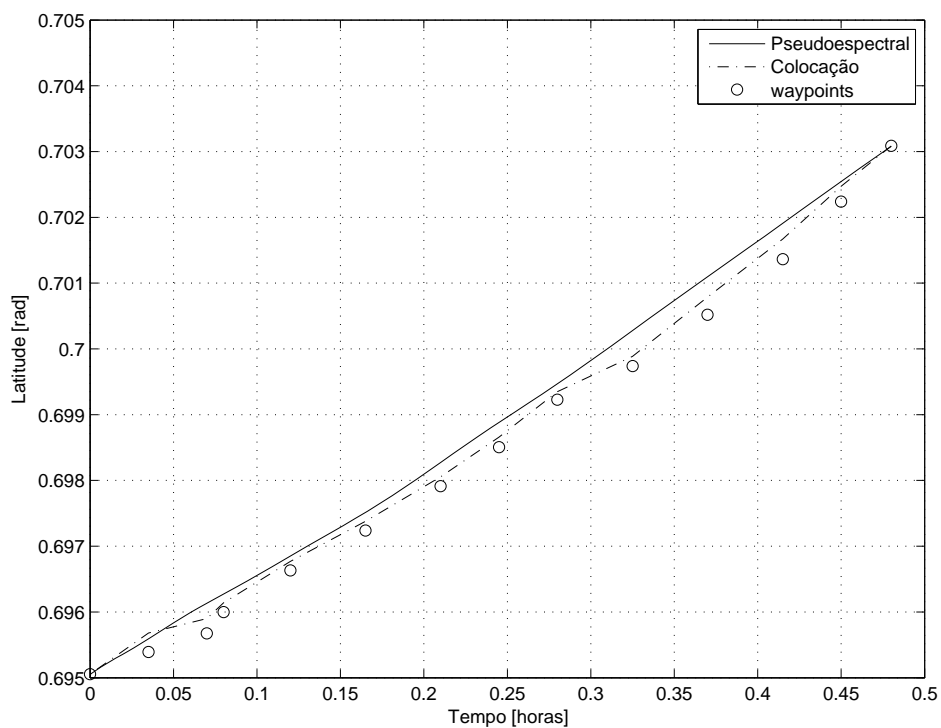


FIGURA 4.2: Comportamento da Latitude ao longo do Tempo

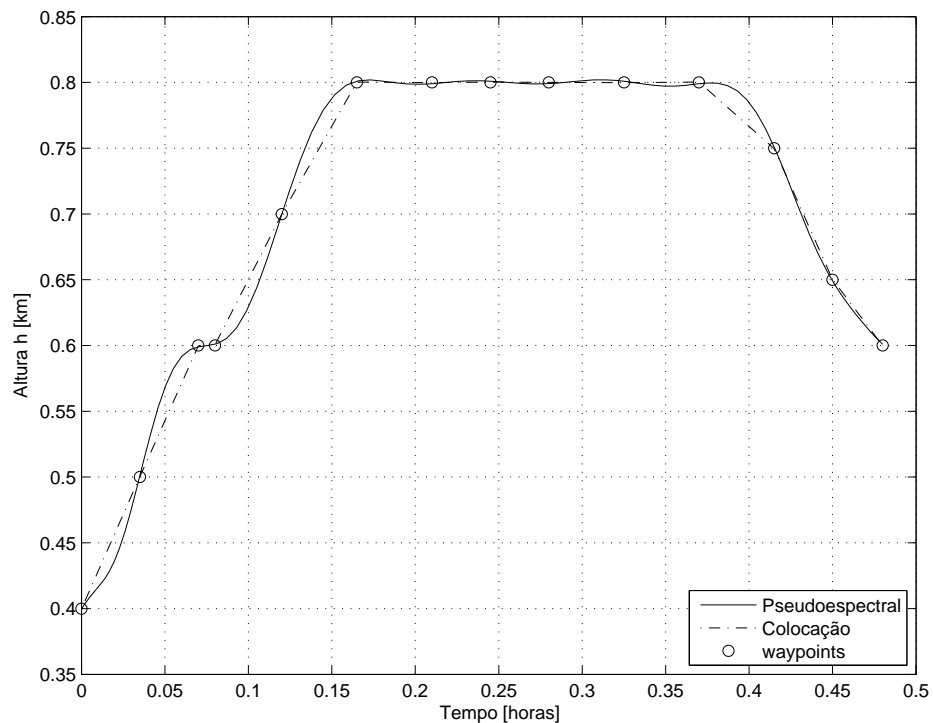


FIGURA 4.3: Comportamento da Altura ao longo do Tempo

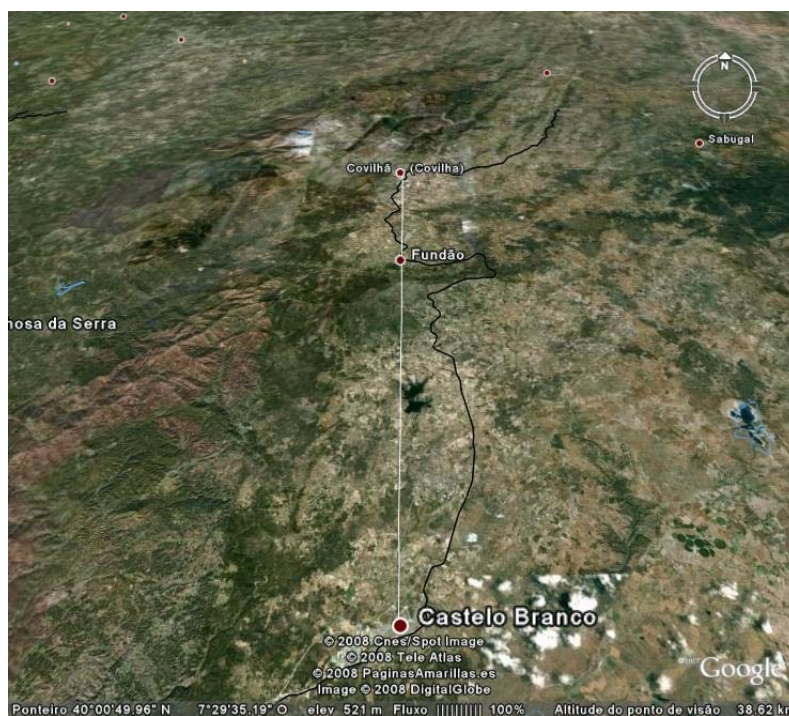


FIGURA 4.4: Projecção no terreno através *Google Earth* da Trajectória obtida pelo método da Colocação

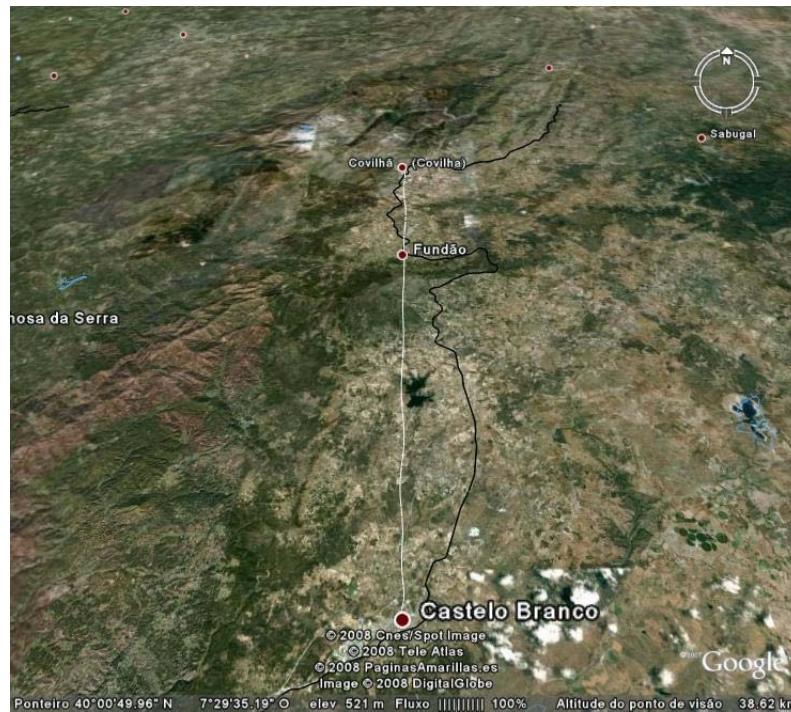


FIGURA 4.5: Projecção no terreno através *Google Earth* da Trajectória obtida pelo método Pseudoespectral

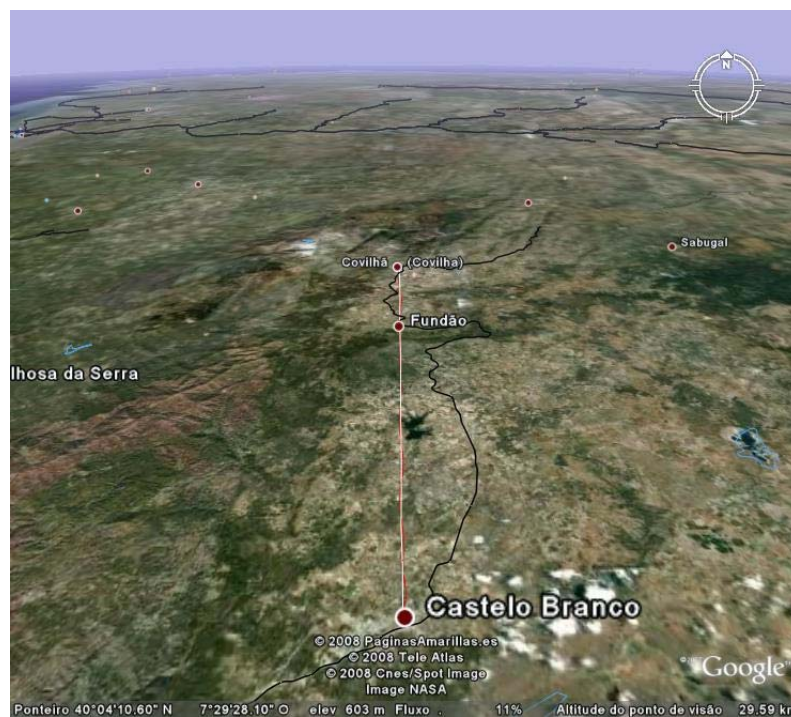


FIGURA 4.6: Projecção no terreno através *Google Earth* da Trajectória obtida pelo método da Colocação (a branco) e Pseudoespectral (a vermelho)

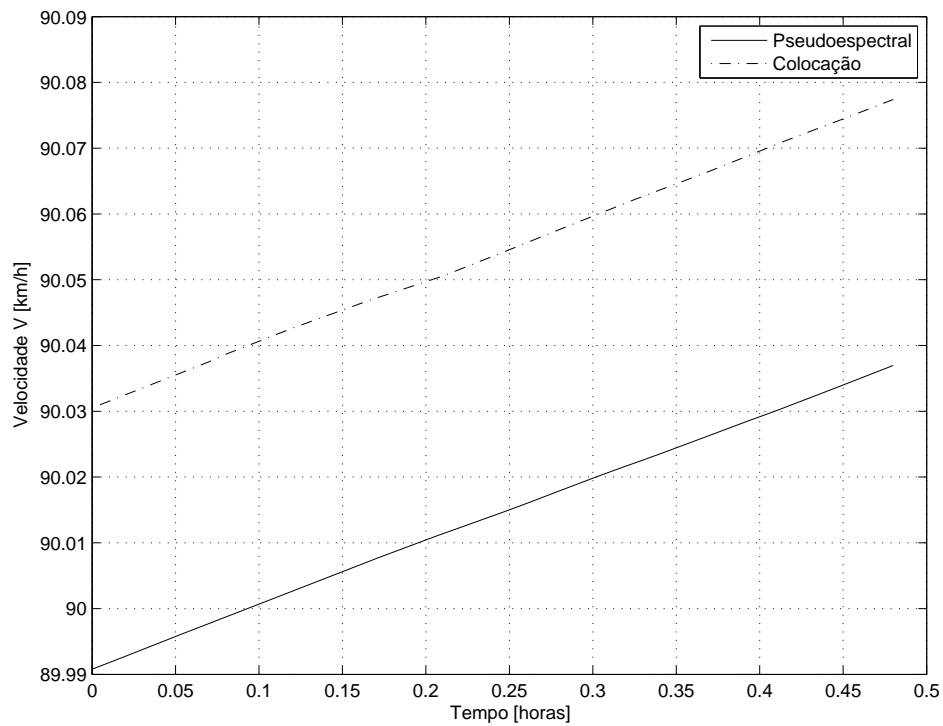


FIGURA 4.7: Comportamento da Velocidade ao longo do Tempo

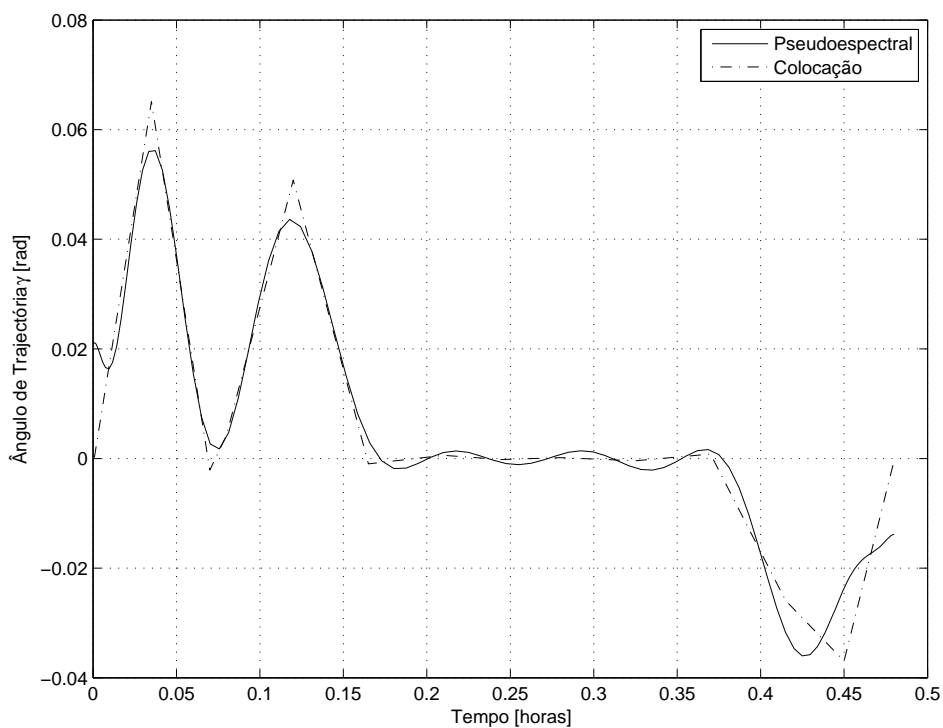


FIGURA 4.8: Comportamento da Ângulo de Trajetória ao longo do Tempo

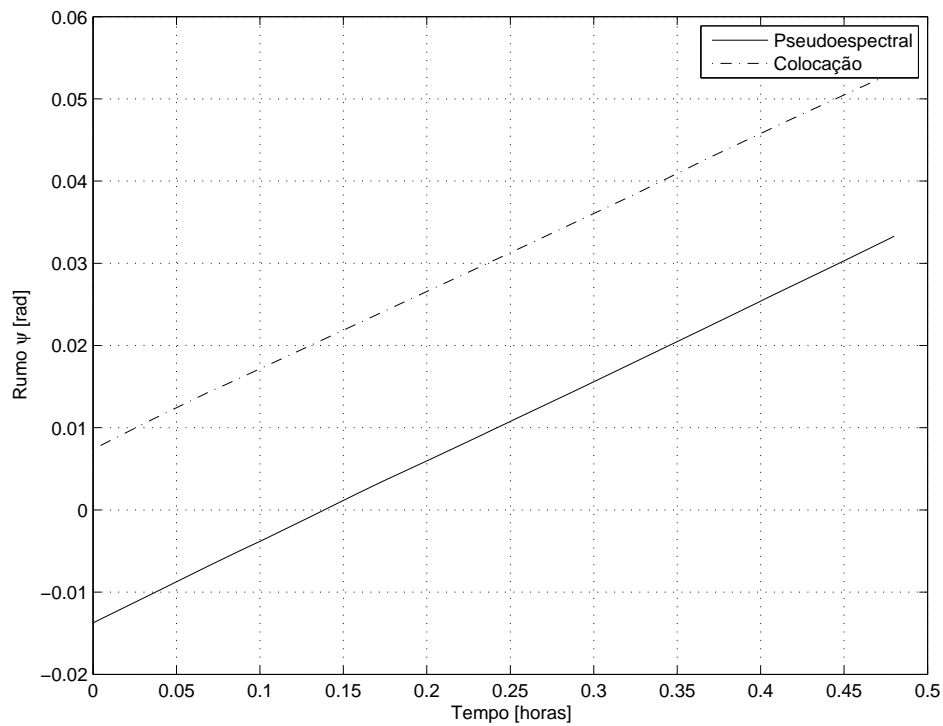


FIGURA 4.9: Comportamento do Rumo ao longo do Tempo

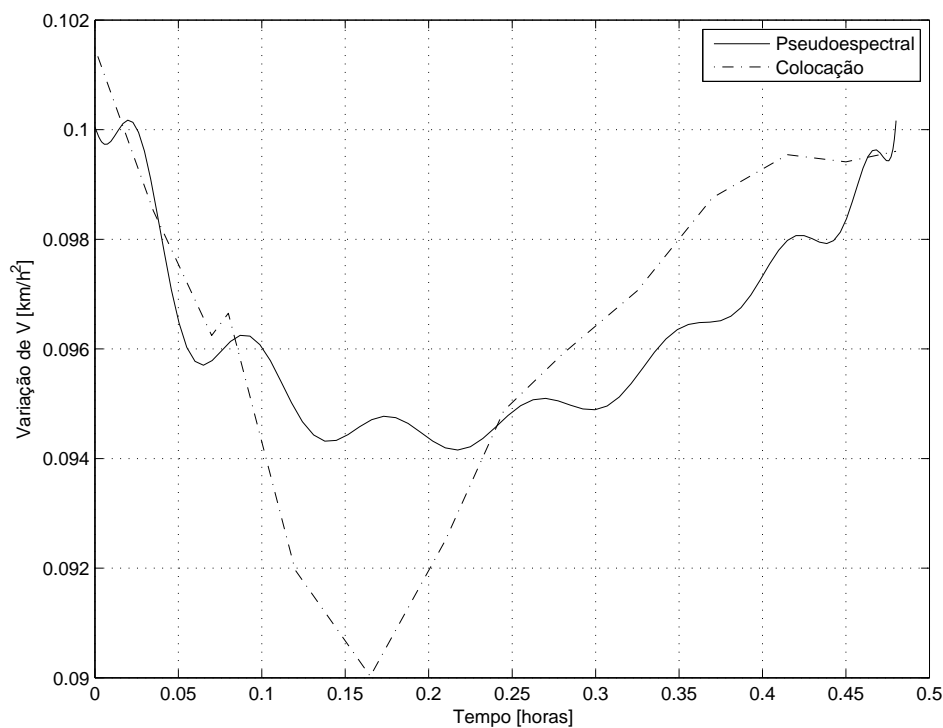


FIGURA 4.10: Comportamento do controle u_1 ao longo do Tempo

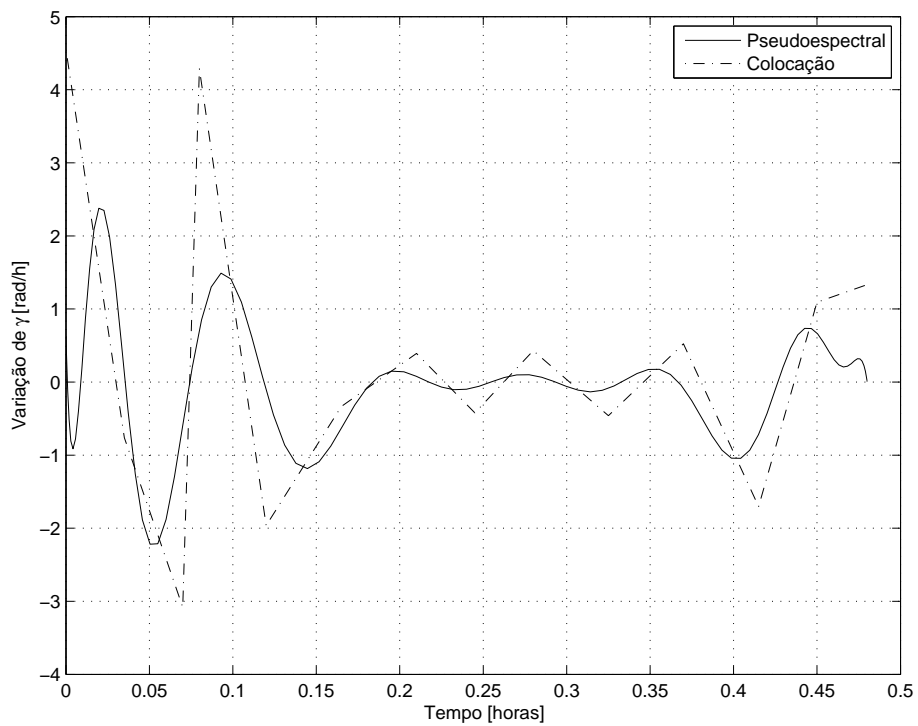


FIGURA 4.11: Comportamento do controle u_2 ao longo do Tempo

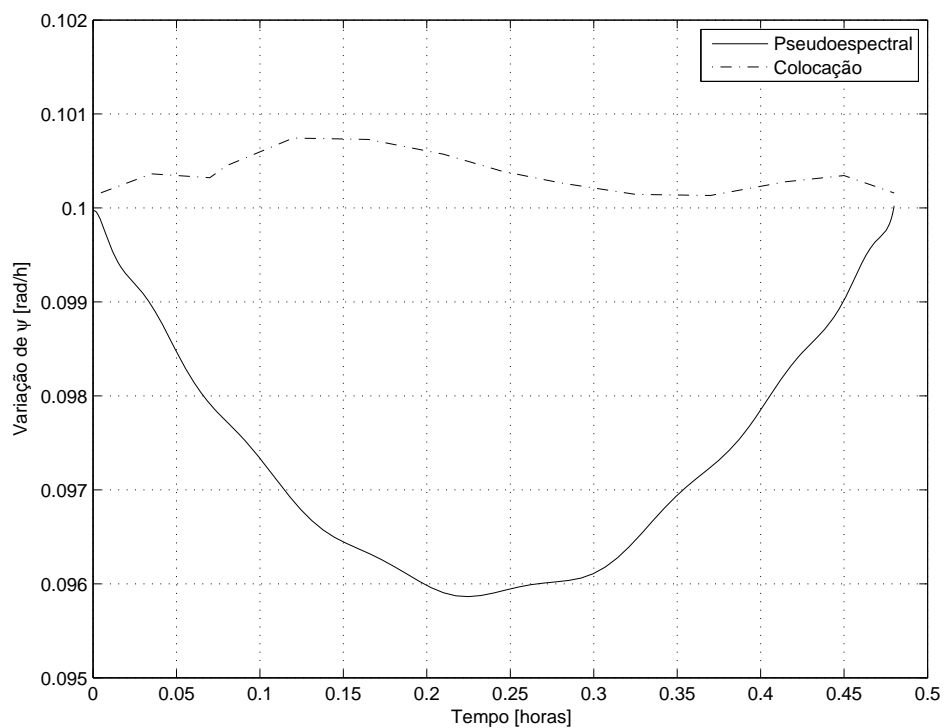


FIGURA 4.12: Comportamento do controle u_3 ao longo do Tempo

Nº	Longitude λ	Latitude φ	Altura h	Tempo de Chegada τ	
	[deg min seg]	[deg min seg]	[m]	[h]	
1	7° 28' 45.66"	40° 15' 54.29"	700	0.000	LPCV
2	7° 29' 37.84"	40° 15' 55.50"	750	0.014	
3	7° 30' 28.55"	40° 15' 57.79"	800	0.023	Refúgio
4	7° 31' 36.32"	40° 16' 38.79"	1100	0.051	S.Martinho
5	7° 32' 05.08"	40° 17' 35.86"	1500	0.072	St.Maria
6	7° 31' 27.13"	40° 18' 19.47"	1350	0.085	
7	7° 30' 32.76"	40° 18' 45.15"	1250	0.097	
8	7° 29' 45.49"	40° 18' 58.96"	1150	0.120	Vila do Carvalho
9	7° 28' 32.44"	40° 19' 05.28"	1000	0.136	
10	7° 27' 24.73"	40° 18' 53.08"	850	0.157	Teixoso
11	7° 26' 56.44"	40° 17' 52.36"	810	0.175	
12	7° 27' 09.48"	40° 16' 45.91"	760	0.193	
13	7° 27' 37.41"	40° 16' 14.20"	730	0.213	Boidobra
14	7° 28' 14.22"	40° 15' 59.65"	710	0.221	
15	7° 28' 45.66"	40° 15' 54.29"	700	0.232	LPCV

TABELA 4.4: Coordenadas dos Waypoints para uma missão em circuito

4.4.2 *Exemplo II*

Neste exemplo a missão que se propõe é efectuar um voo em circuito em torno do Conselho da Covilhã com início e fim aeródromo deste (LPCV), passando por algumas das localidades deste conselho.

Este exemplo feito prepositadamente visto que conseguir trajectórias em volta é mais complicado, e assim testa-se a capacidade da optimização de trajectória e dos métodos a que se recorre no âmbito deste trabalho.

4.4.2.1 Plano de Voo

O plano de voo para esta missão está especificado na tabela 4.4. A escolha dos waypoints, da mesma forma como se procedeu no *exemplo I*, teve de ter em conta factores dinâmicos da aeronave, como velocidade de cruzeiro, ângulo de ataque máximo, etc.

4.4.2.2 Trajectória Óptima

A abordagem para este problema é em tudo igual ao exemplo anterior, com as restrições das variáveis de estado e controlo (tabela 4.2). Aplicando os dois métodos que resolvem o problema de optimização de trajectória constatou-se que o método da Colocação não consegue convergir, mesmo tentando aumentar o número de nós não foi possível obter

	$\lambda(\tau_f)$	$\varphi(\tau_f)$	$h(\tau_f)$	J
	[m]	[m]	[km]	
Pseudoespectral	-0.1305	0.7026	0.5749	3.7×10^{-7}

TABELA 4.5: Valores das variáveis de estado no final da trajectória

dados consistentes para este método. Já para o método pseudoespectral conseguiu-se uma trajectória nominal que passou por todos os waypoints pré-definidos. A tabela 4.5 mostra os valores finais que a função de custo pretende minimizar.

O número de nós ($N + 1$) utilizados foi 25, o porquê deste número é devido ao facto que para números de nós inferior a este o método não convergia para uma solução, a partir deste valor o tempo de calculo já se tornava elevado pois o número de variáveis aumenta nove vezes mais (6 variáveis de estado e 3 de controlo) e a partir do momento que tem uma trajectória que cumpra todas as restrições não é necessário refinar o método.

Relativamente aos dados obtidos, constatou-se que o método pseudoespectral muito em voga na actualidade, foi capaz de encontrar uma solução capaz de cumprir todos as restrições (minimizar o atraso em cada waypoint), as figuras (4.13), (4.14) e (4.15) mostram isso mesmo, o único problema acontece no *waypoint* final, pois como não faz parte das restrições do programa computacional utilizado, mas sim da função de custo, este não consegue refinar solução, no entanto a restante trajectória é conseguida. Em relação à Longitude e Latitude (figuras (4.13) e (4.14) respectivamente), a trajectória não poderia ser mais suave, já a Altitude (figura (4.15)) apresenta alguma oscilação, não é muita porque se projectar-mos a trajectória em 3D vê-mos que não existem oscilações muito acentuadas. Isto acontece porque os tempos de chegada para cada *waypoint* são muito próximos uns dos outros e o tempo para finalizar a trajectória é relativamente pequeno em relação à velocidade de cruzeiro da aeronave, o que significa que a trajectória terá de ser feita a uma velocidade no limiar da velocidade de cruzeiro (figura (4.18)), implicando variações constantes de acelerações (figura (4.21)) e do Ângulo de Trajectória (figura (4.19)) para que não se ultrapasse a restrição imposta para a velocidade.

Os controlos representados nas figuras (4.21), (4.22) e (4.23), que não são nada mais que as variações da Velocidade, Ângulo de trajectória e Rumo, são por si só um indicador do que uma aeronave com estas características dinâmicas terá de realizar para cumprir esta trajectória com estas especificações.

Como já foi referido a pré-definição dos *waypoints*, para este plano de voo, foi proposta visto que à partida seria uma trajectória difícil de realizar. Tanto é, que aplicando o método da Colocação não é possível obter uma trajectória nominal o que não acontece com o método pseudoespectral. Sendo assim, este último dá boas indicações para

resolver todo o tipo destes problemas. O próximo passo será encontrar programas computacionais com melhor capacidade em termos de procura de soluções em programação não linear e tempo de convergência para a solução.

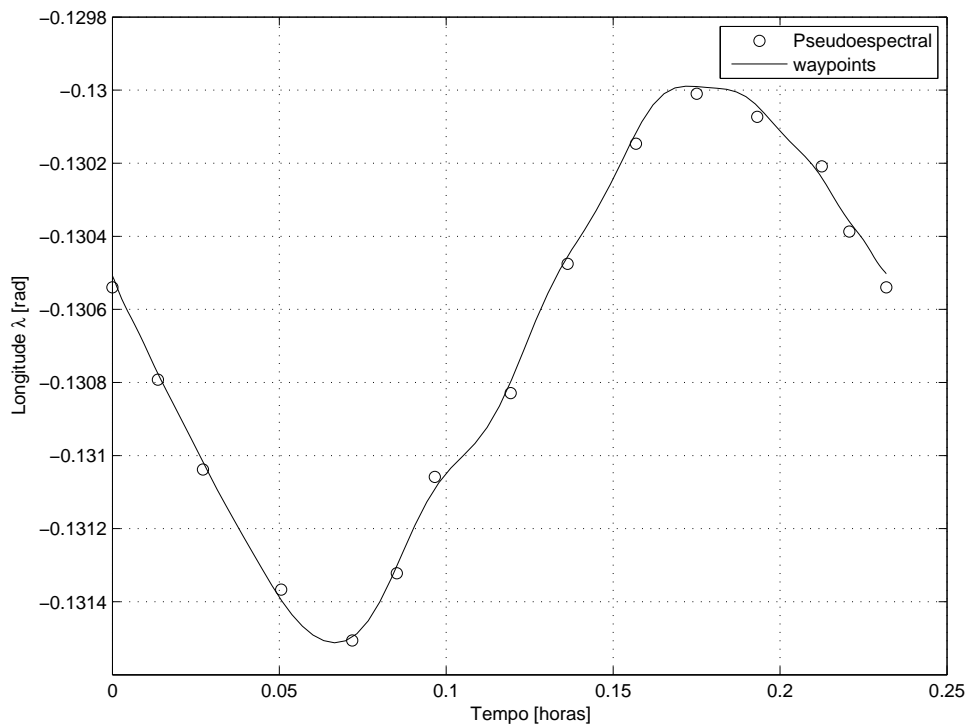


FIGURA 4.13: Comportamento da Longitude ao longo do Tempo

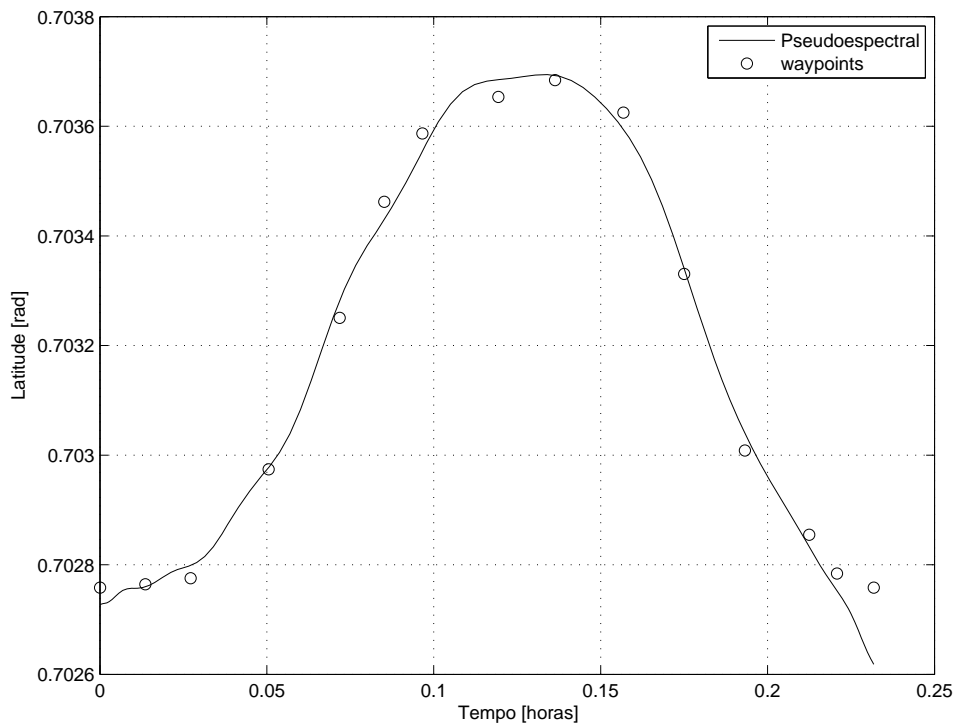


FIGURA 4.14: Comportamento da Latitude ao longo do Tempo

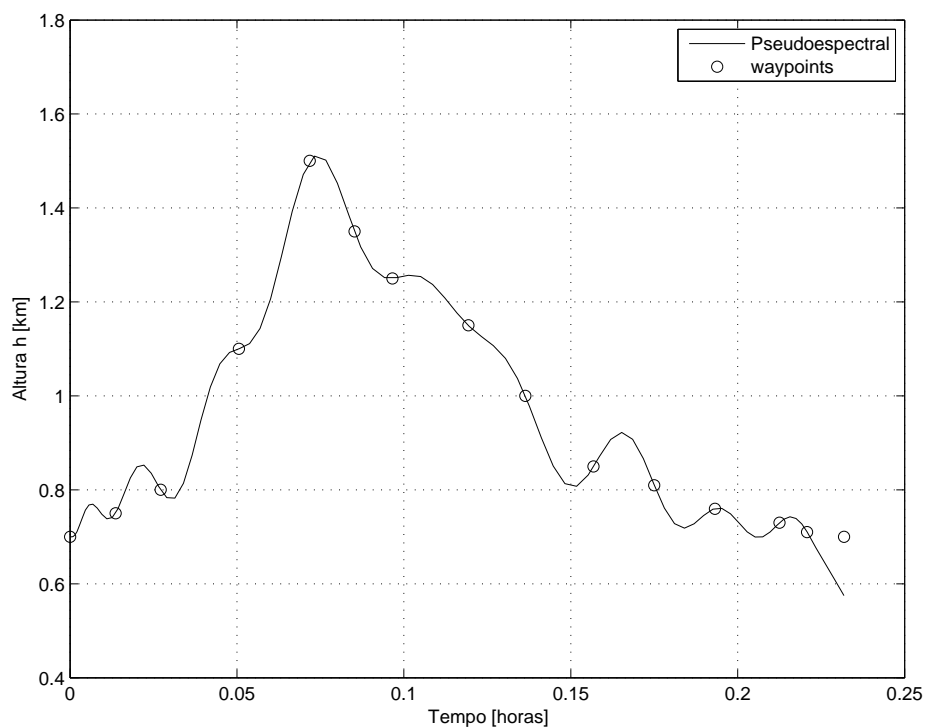


FIGURA 4.15: Comportamento da Altitude ao longo do Tempo

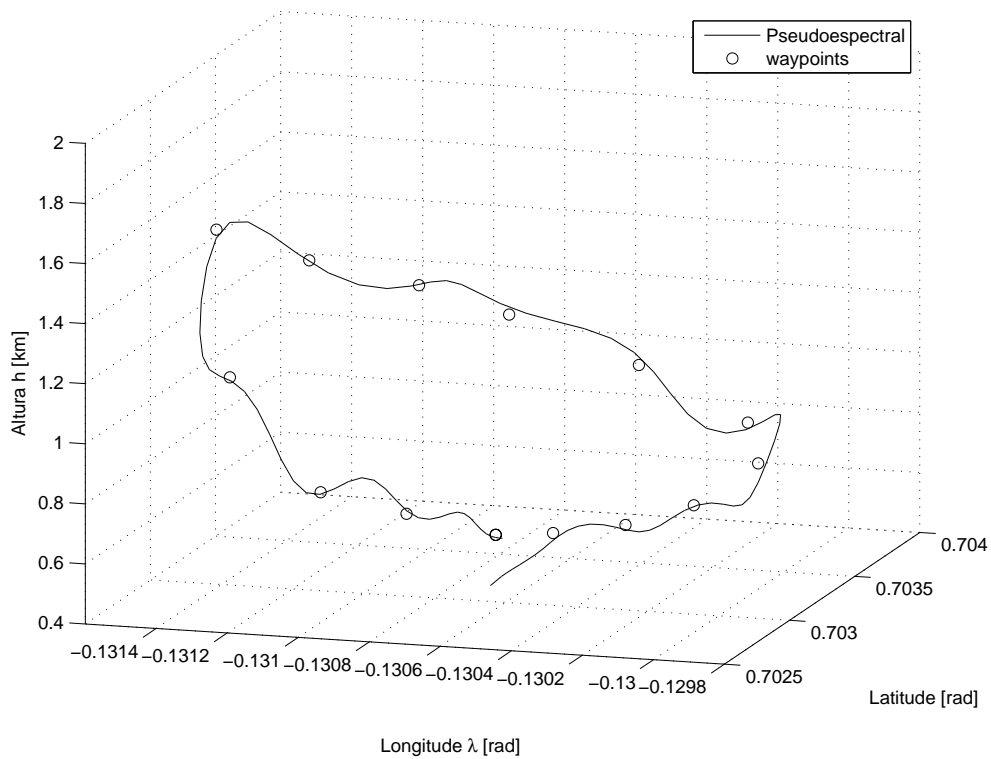


FIGURA 4.16: Trajectória 3D

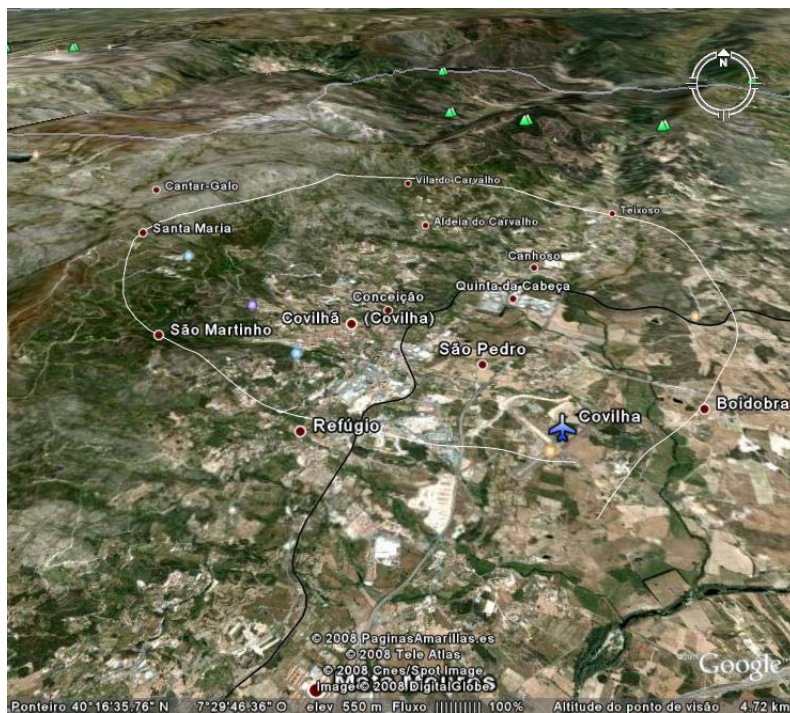


FIGURA 4.17: Projecção no terreno através *GoogleEarth* da Trajectória obtida pelo método da Pseudoespectral

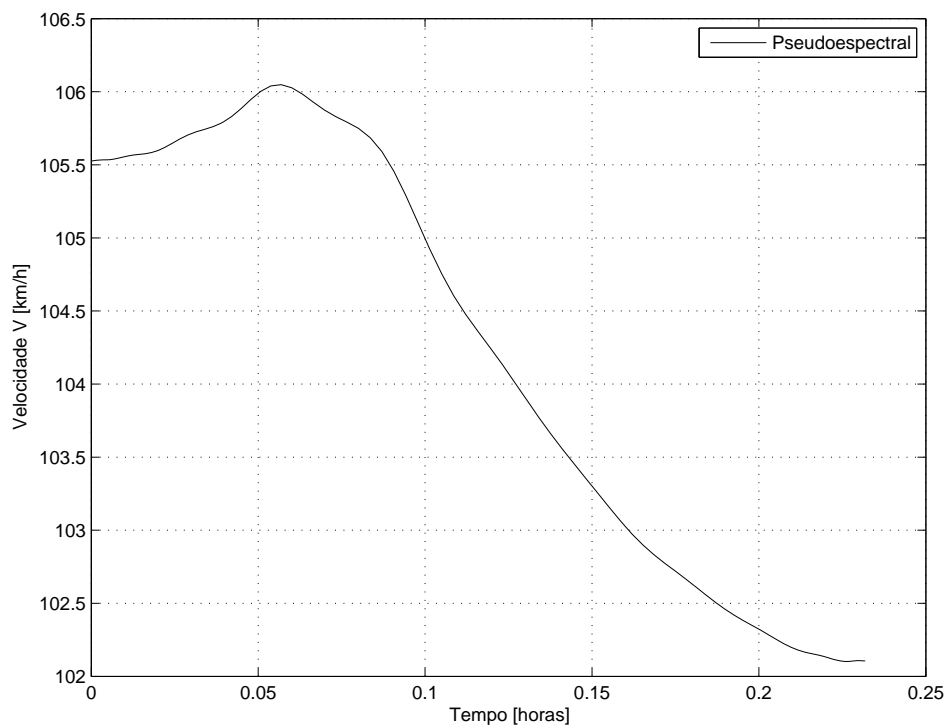


FIGURA 4.18: Comportamento da Velocidade ao longo do Tempo

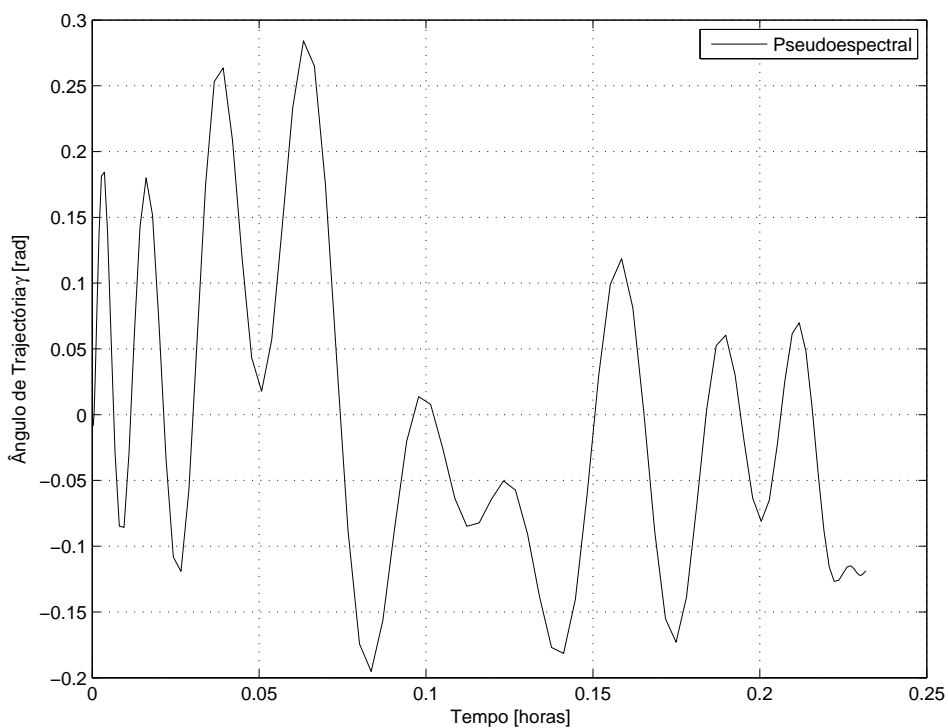


FIGURA 4.19: Comportamento da Ângulo de Trajectória ao longo do Tempo

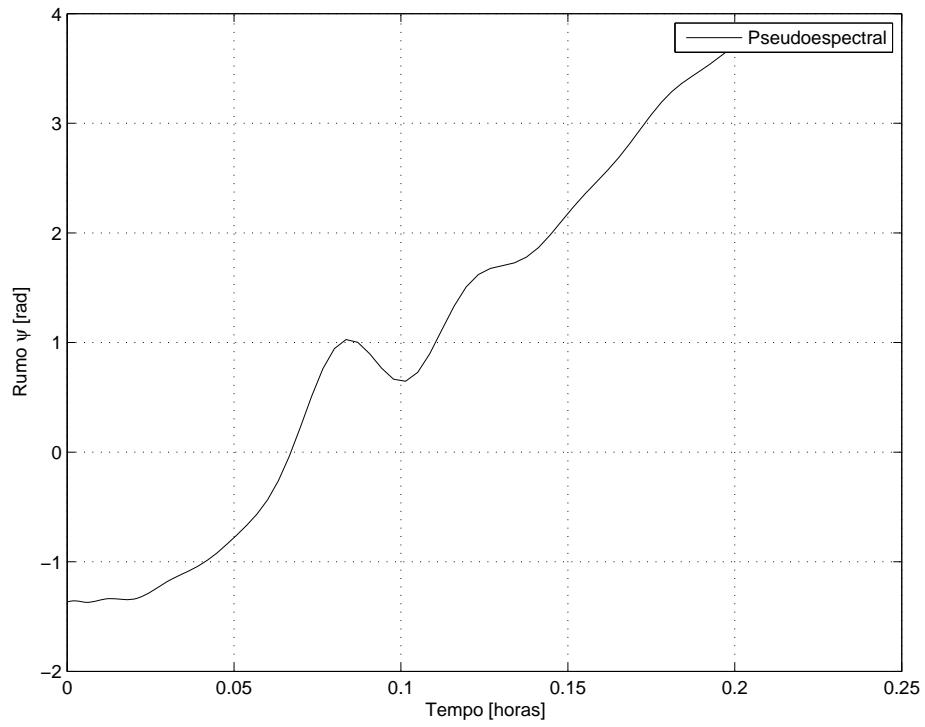


FIGURA 4.20: Comportamento do Rumo ao longo do Tempo

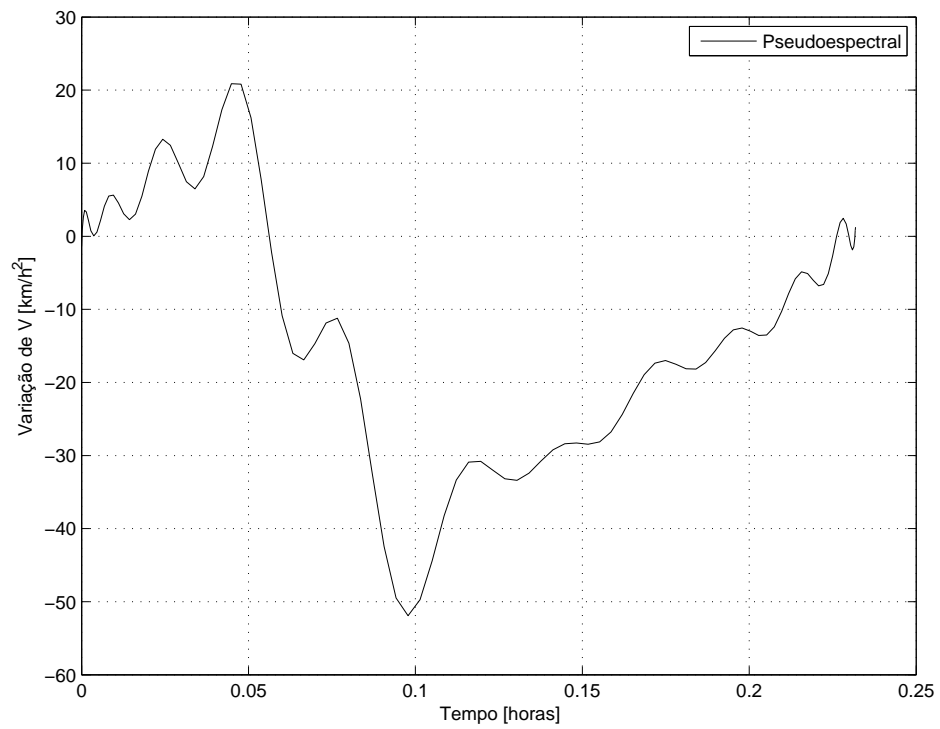


FIGURA 4.21: Comportamento do controle u_1 ao longo do Tempo

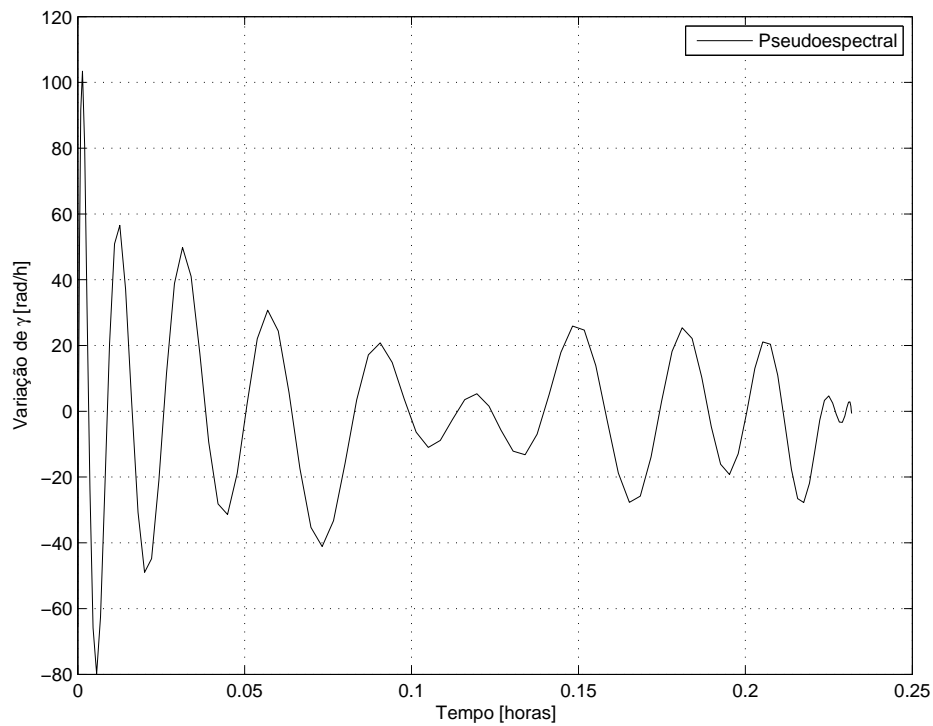


FIGURA 4.22: Comportamento do controle u_2 ao longo do Tempo

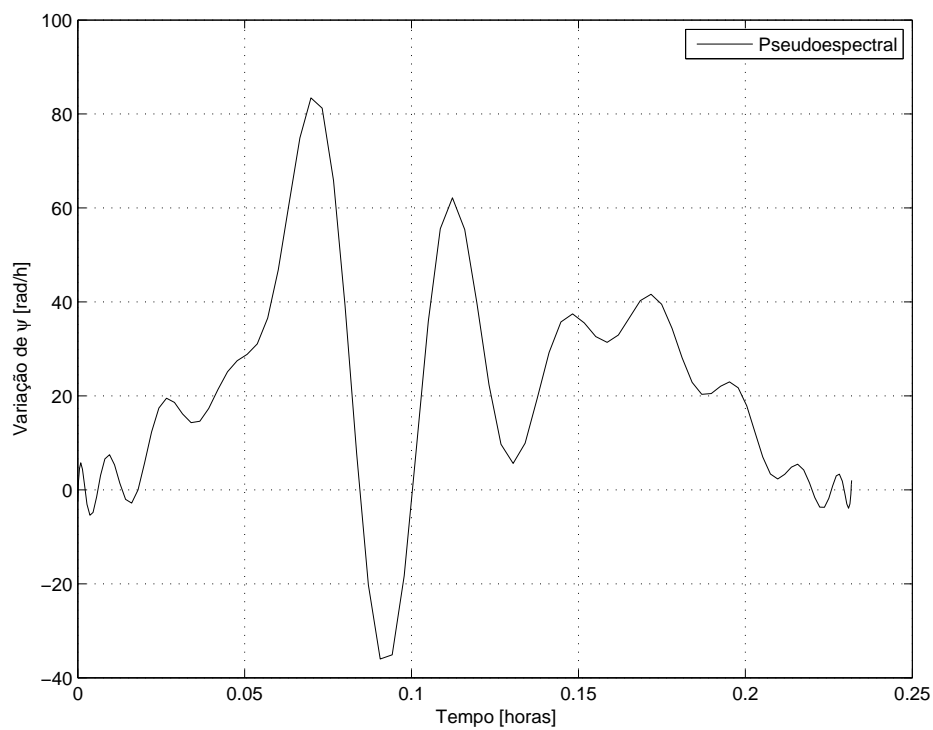


FIGURA 4.23: Comportamento do controle u_3 ao longo do Tempo

4.4.2.3 Controlo de Trajectória de Referência

Com a trajectória nominal obtida, o próximo passo será controlar a aeronave ao longo da referência. O método que foi descrito é o denominado controlo predictivo que permite controlar sistemas não lineares através de uma determinada referência. O que se pretendia, neste exemplo, era determinar o controlo ao longo de toda a trajectória, optou-se no entanto por fazer o controlo num troço da trajectória, isto porque para minimizar a equação (4.44) foi utilizado o algoritmo de optimização não linear *fmincon* do *MatLab*, que demora muito tempo de simulação se tivermos uma tolerância da solução na ordem dos 10^{-3} .

Esta tolerância é o mínimo que se pode impor, pelo menos para esta ferramenta, pois caso contrário iria-se ter um erro entre a trajectória nominal e a controlada que não era aceitável. Assim, realizou-se a simulação de um voo controlado entre o aeródromo da Covilhã e a Freguesia do Refúgio.

Poderia-se pensar, em analogia ao que se na optimização de trajectória, uma mudança de escala. Neste caso a mudança de escala iria prejudicar o processo de optimização, pois se passasse-mos as distâncias para quilómetros e o tempo para horas, as taxas de arfagem, rolamento e guinada ficariam com valores escalares muito elevados o que dificultaria a convergência. Por isto, tem de se simular com o tempo em segundos e as distâncias em metros, o que implica ainda mais tempo de simulação.

Constata-se nas figuras (4.24), (4.25) e (4.26) que o controlo predictivo é capaz de controlar a aeronave ao longo de uma trajectória de referência. O facto da trajectória controlada não coincidir exactamente com a trajectória de referência que pode ser observado no exemplo 3D (figura 4.27) ou na projecção no *Google Earth* é devido ao facto da tolerância de minimização ser de 10^{-3} , e como é obvio o facto da Latitude e Longitude serem trabalhadas em radianos implicando que erros na ordem da milésima em relação à trajectória nominal, provocam discrepâncias entre trajectórias na ordem das centenas de metros.

O Ângulo de Trajectória e o Rumo (figuras 4.30) e 4.31) são praticamente cumpridos pelo controlador (caso contrário não era possível seguir a trajectória nominal), já a Velocidade (figura 4.29) apresenta alguma oscilação, este acontece porque no modelo dinâmico da aeronave é utilizada uma tracção máxima média, inicialmente aproximou-se a tabela 2.1 pelas funções multiquadráticas de Hardy (ver Capítulo 2). Apesar da aproximação da Tracção máxima pela Tracção máxima média ser grosseira é facto é que o tempo de simulação é reduzido em muito.

s	V	γ	ψ
[m]	[m/s]	[rad]	[rad]
0.2310	2.0523	0.0041	0.0003

TABELA 4.6: Erro RMSE entre a trajectória nominal e a efectuada pelo controlador

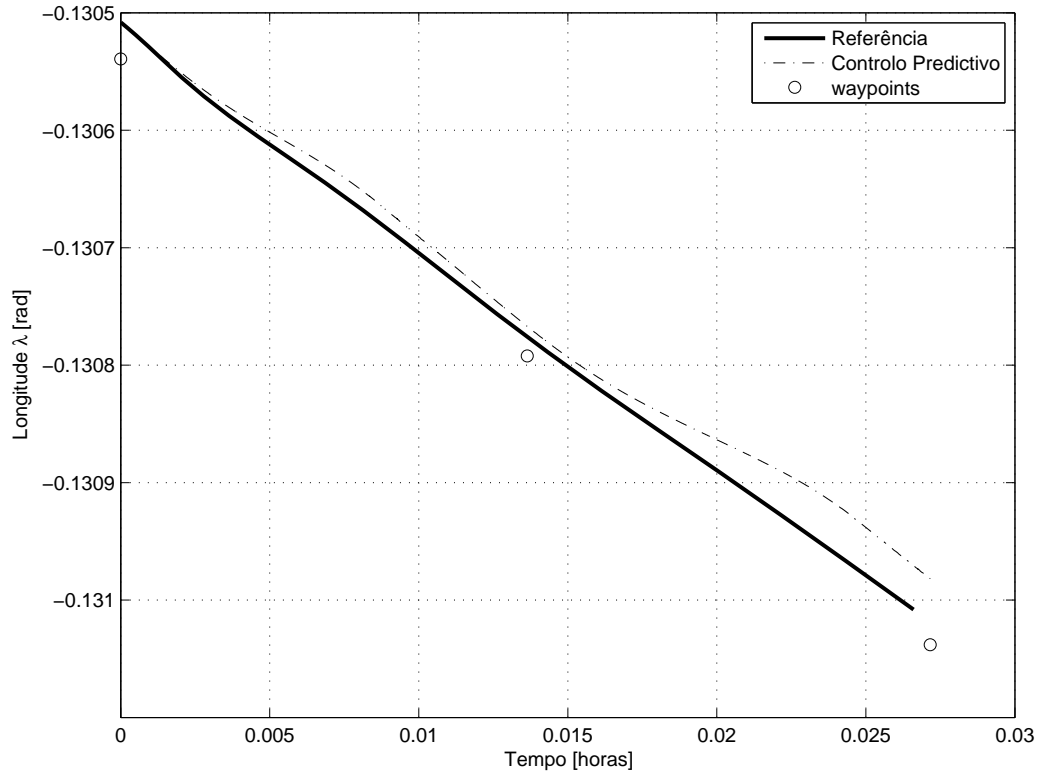


FIGURA 4.24: Comportamento da Longitude na trajectória nominal e na controlada

O erro médio $RMSE$, calculado pela equação (4.48), entre a trajectória nominal e a produzida pelo controlador para as variáveis de referência está descrito na tabela 4.6.

Relativamente às restantes variáveis de estado do modelo dinâmico da aeronave, observe-se que o ângulo de pranchamento ϕ representado na figura (4.32) é praticamente nulo que era de esperar visto neste troço não existe nenhuma volta para a aeronave efectuar. Quanto ao ângulo de arfagem θ (figura (4.33)), este possui uma resposta parecida ao Ângulo de trajectória pois estão relacionados através da equação do trim 4.57. O ruído que apresenta na realidade não é factível pois a dinâmica da aeronave não o permite.

Quanto às taxas p , q e r , a sua resposta temporal tende a permanecer nula (figuras (4.34), (4.35) e (4.36)), excepto a taxa de arfagem q que tem de variar para que seja possível cumprir a trajectória em altitude.

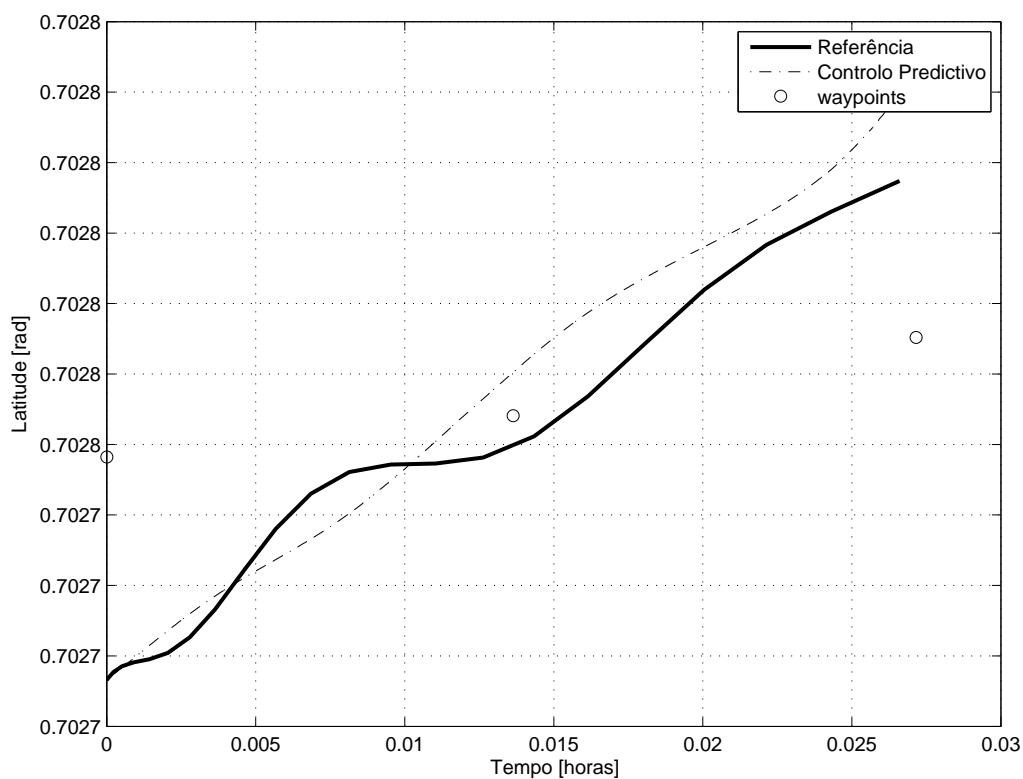


FIGURA 4.25: Comportamento da Latitude na trajetória nominal e na controlada

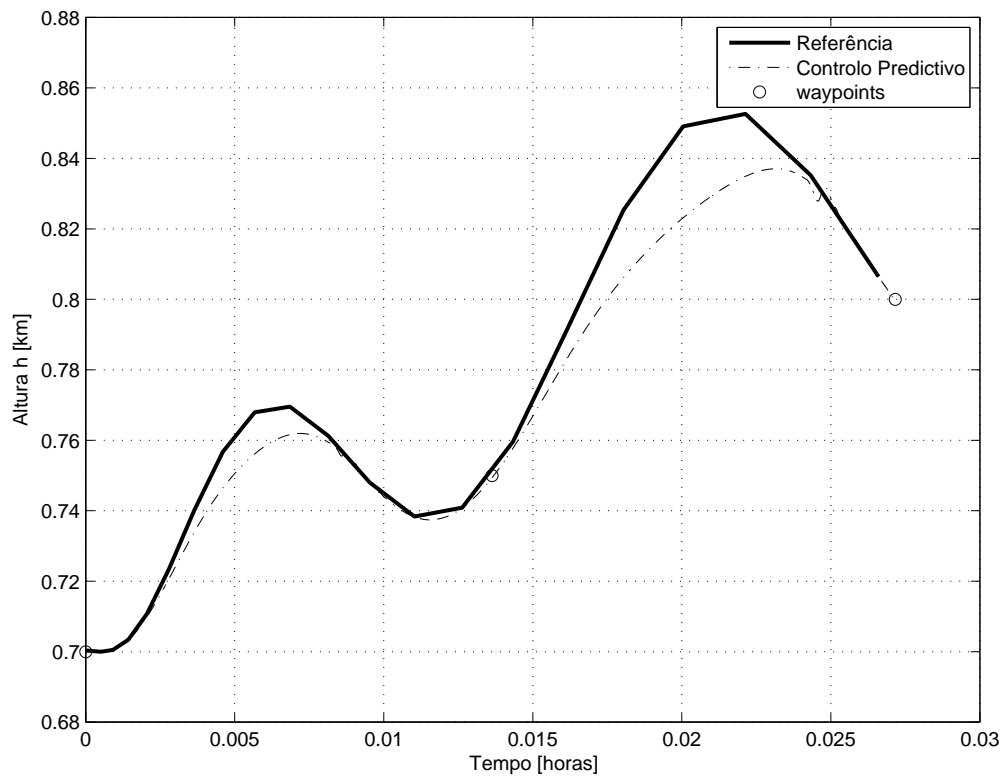


FIGURA 4.26: Comportamento da Altitude na trajetória nominal e na controlada

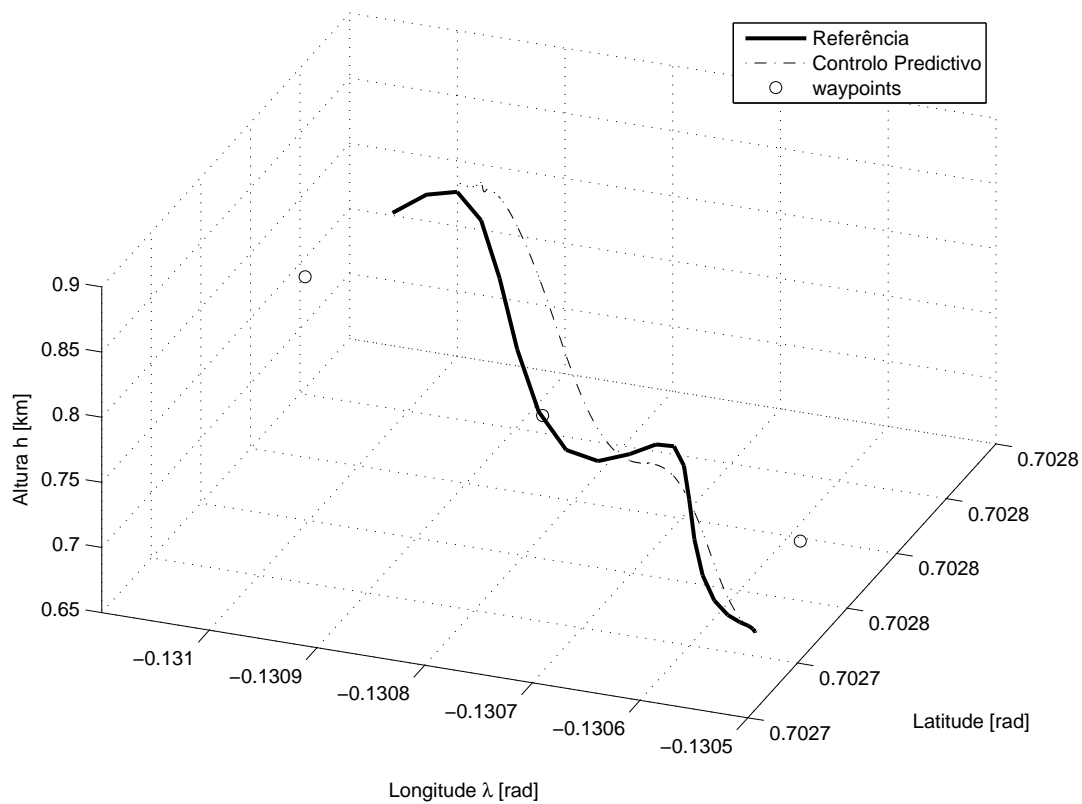


FIGURA 4.27: Trajectória 3D nominal e controlada

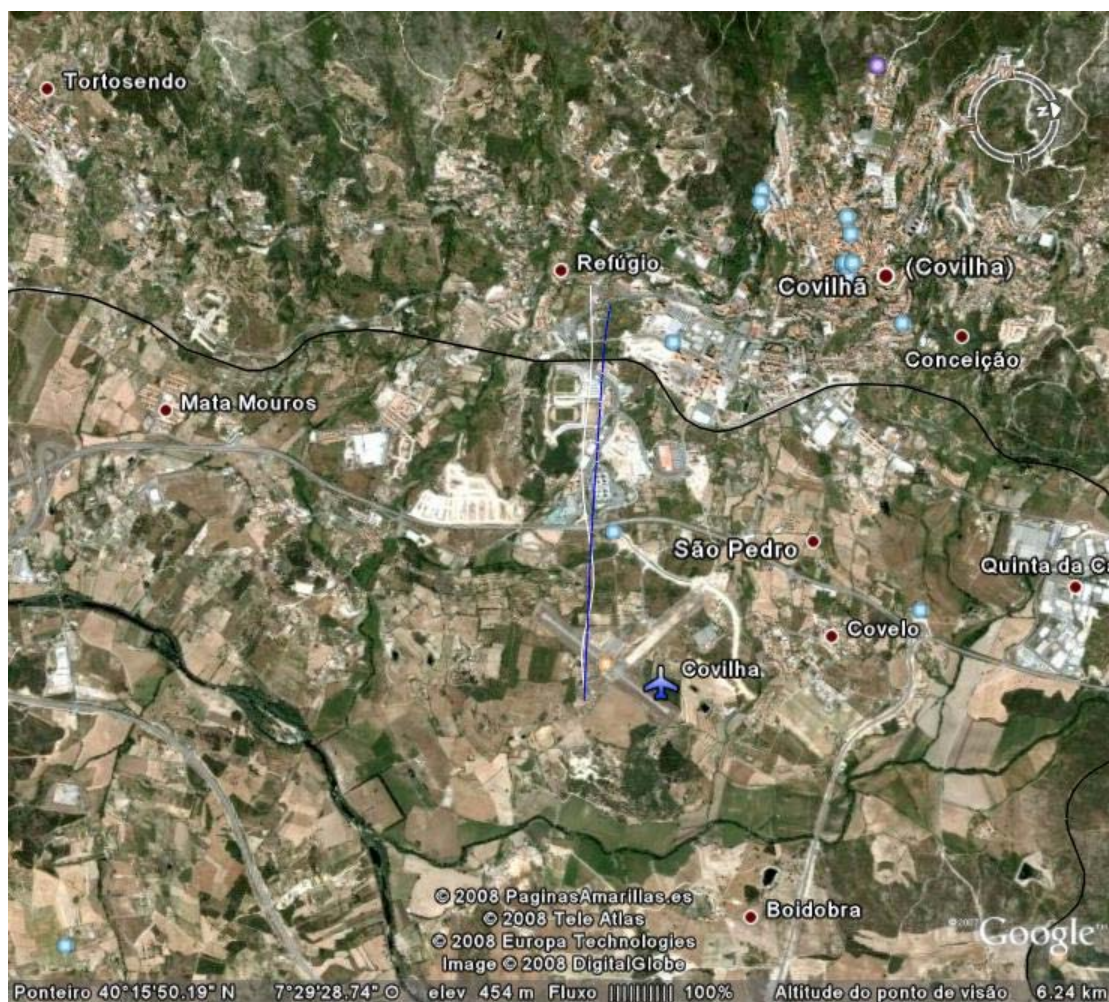


FIGURA 4.28: Projecção no terreno através *GoogleEarth* da Trajectória de referência (branco) e da Trajectória realizada (azul)

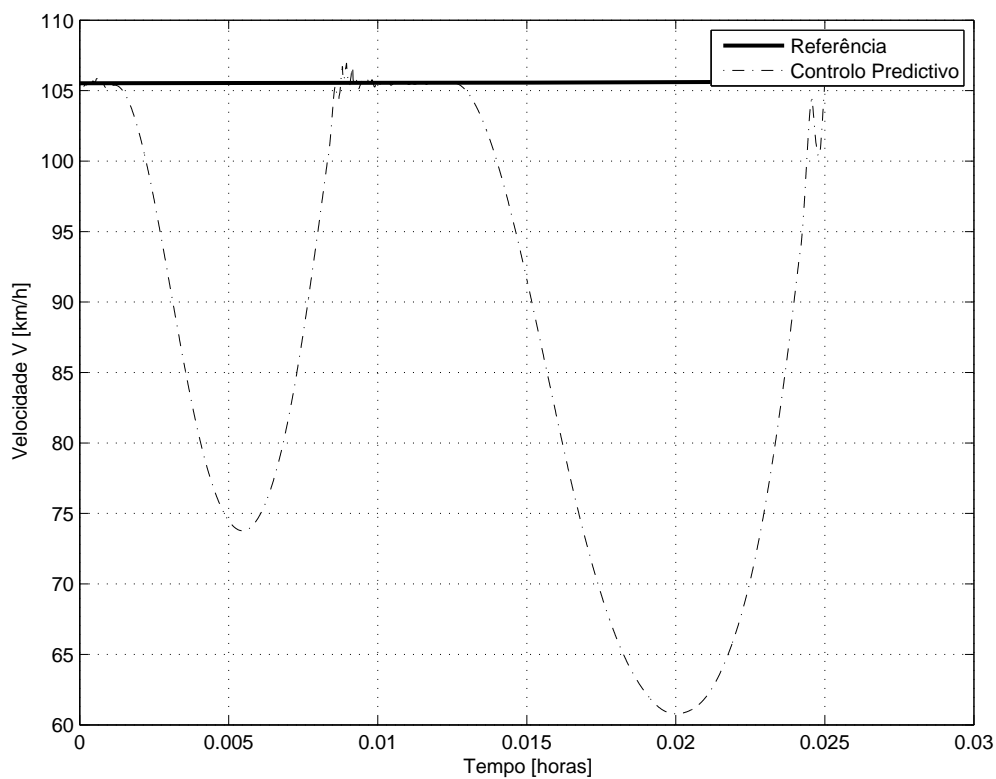


FIGURA 4.29: Comportamento da Velocidade na trajetória nominal e na controlada

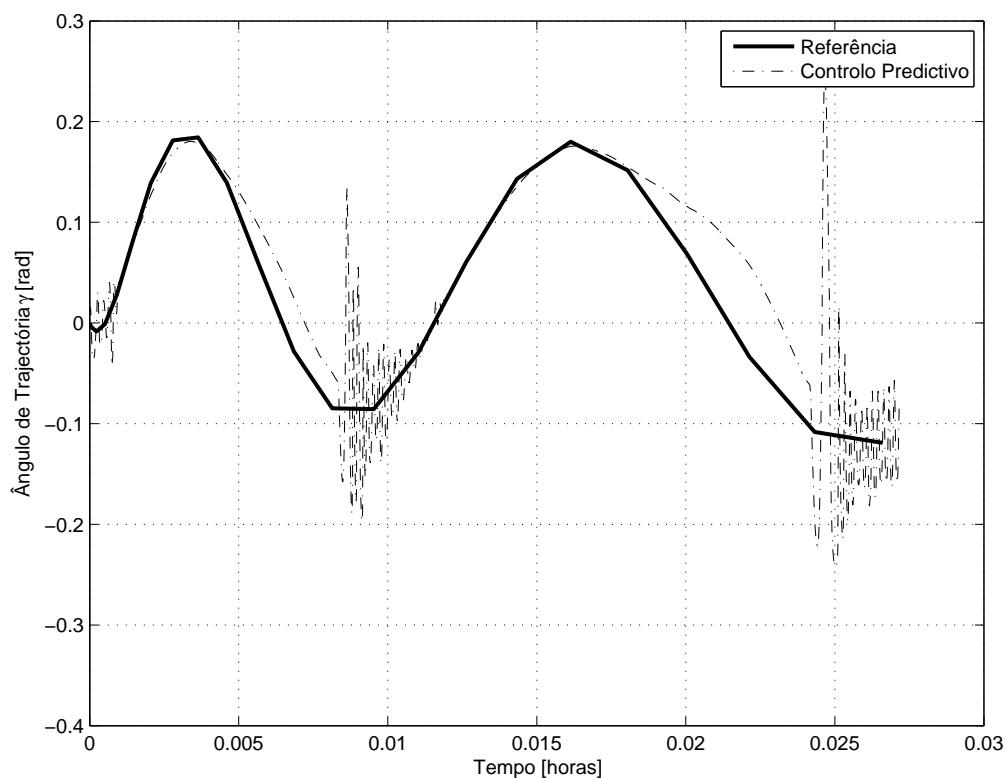


FIGURA 4.30: Comportamento do Ângulo de Trajetória na trajetória nominal e na controlada

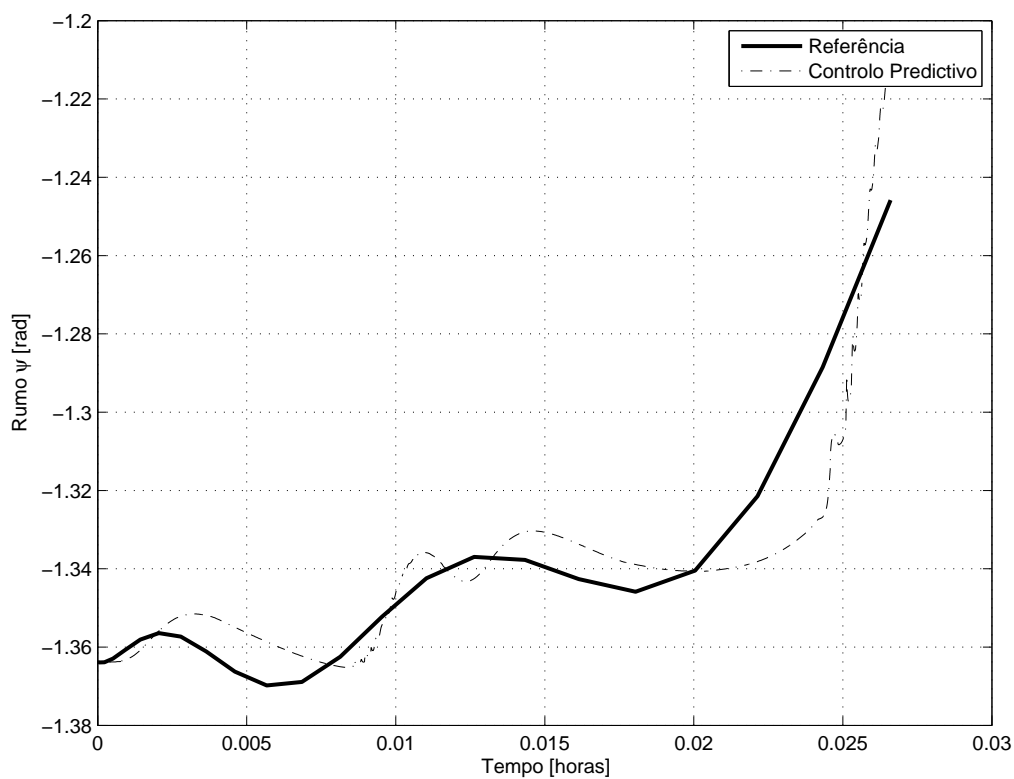


FIGURA 4.31: Comportamento do Rumo na trajetória nominal e na controlada

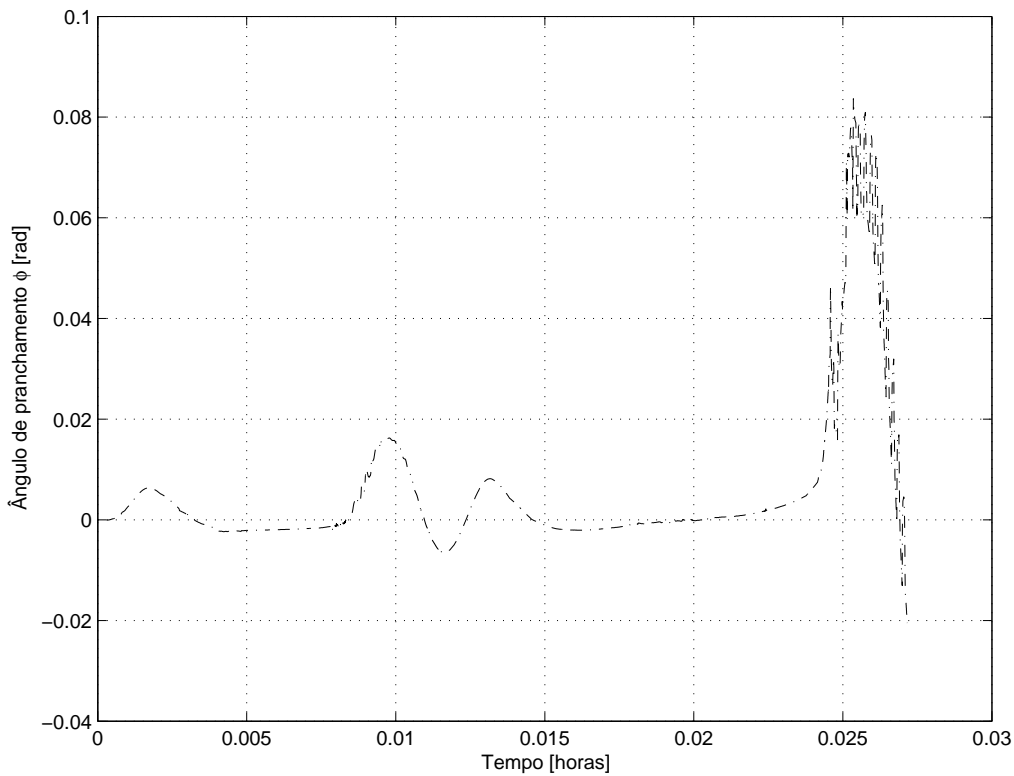


FIGURA 4.32: Comportamento do pranchamento durante a trajetória

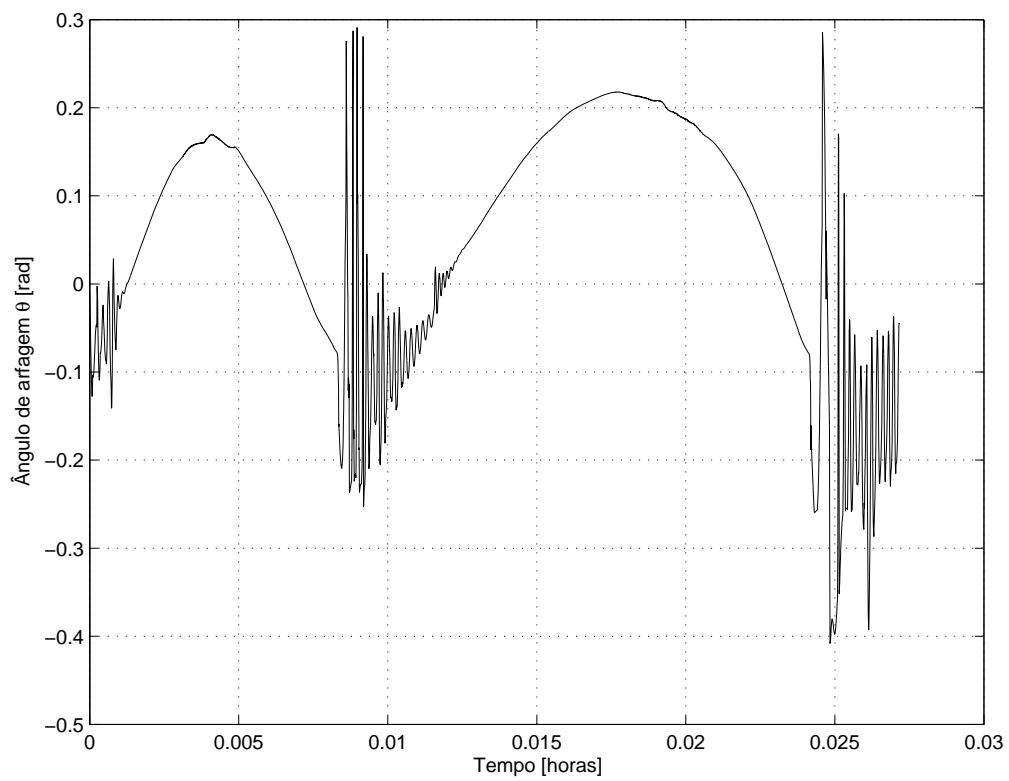


FIGURA 4.33: Comportamento da argagem durante a trajectória

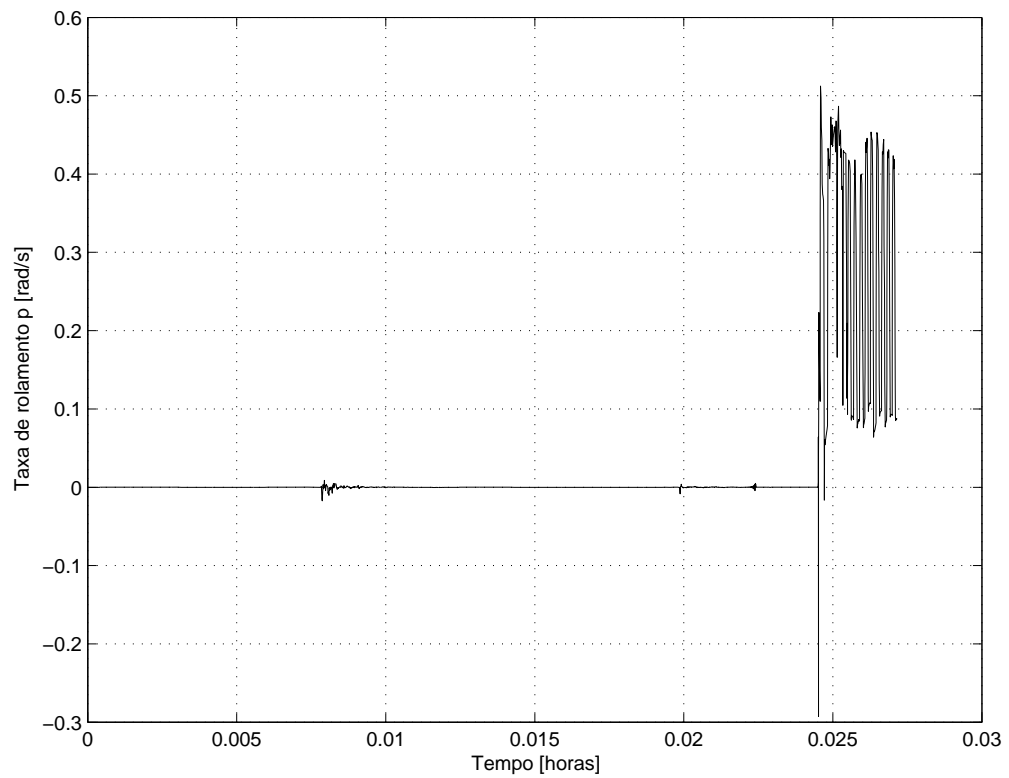


FIGURA 4.34: Comportamento da taxa de rolamento durante a trajectória

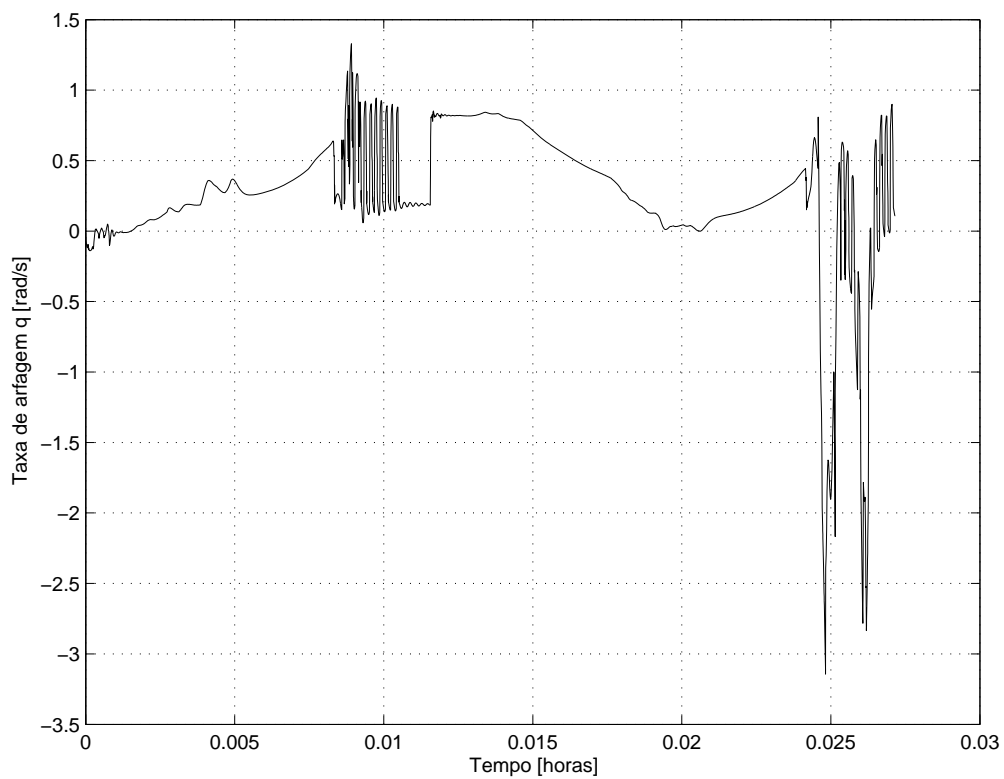


FIGURA 4.35: Comportamento da taxa de arfagem durante a trajetória

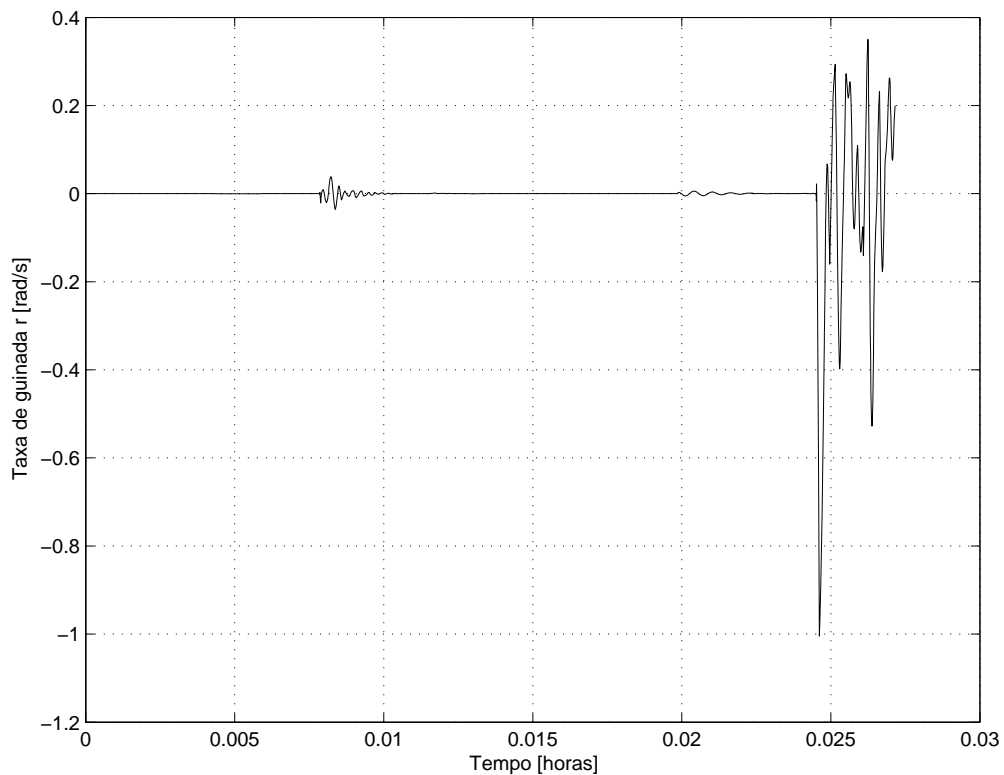


FIGURA 4.36: Comportamento da taxa de guinada durante a trajetória

Capítulo 5

Conclusão

Este trabalho tem como objectivo formular, modular, aplicar e resolver o problema de optimização de trajectórias 4D definidas por *waypoints*. Para que tal aconteça, foi necessário antes demais, saber onde é que o problema se encontra na actualidade.

O projecto de trajectórias 4D só pode ser resolvido através do processo de optimização. No Capítulo 2 foi revisto o modo de funcionamento e as bases matemáticas necessárias para se poder compreender e aplicar métodos de optimização. Durante este estudo conclui-se que o tema deste trabalho só poderia ser resolvido através da programação não linear sujeita a restrições.

Revista a teoria dos métodos de optimização, o próximo passo foi compreender em que se basei o controlo óptimo, também denominado, em Aeronáutica, de optimização de trajectória (Capítulo 3). Em optimização de trajectória existem dois métodos que resolvem este problema, os métodos indirectos e os métodos directos. Os métodos indirectos para resolverem o problema de optimização de trajectória recorrem ao princípio do máximo de Pontryagin que estabelecem um conjunto de condições necessárias para a solução do problema. Por sua vez, os métodos directos, acentam todos eles no princípio de transcrever o problema de optimização de trajectória para um problema de programação não linear. Para que seja possível esta transcrição é necessário discretizar as equações da dinâmica do sistema e as restrições. É neste ponto onde reside a possível resolução do problema deste trabalho.

Visto que as equações de estado são todas elas diferenciais, os métodos de discretização acentam nos esquemas de integração. Como já foi referido alguns destes métodos são bastantes conhecidos nomeadamente aqueles que derivam dos esquemas de Runge-Kutta, entre eles o método da Colocação com integração trapezoidal utilizado neste estudo. Esta

escolha justifica-se pelo facto dos métodos implícitos serem melhores integradores que os explícitos.

Outros métodos capazes de resolverem problemas de integração são os métodos pseudoespectrais, estes não são tão conhecidos como os esquemas de Runge-Kutta, mas são bastante utilizados para problemas dinâmicos, e de uns anos a esta parte, começam a ser utilizados em problemas de optimização de trajectória. O método pseudoespectral utilizado neste trabalho é o que recorre aos polinómios de Chebyshev para parametrizar as variáveis de estado e de controlo.

Estes métodos foram testados no problema de Bryson-Ho, por ser um problema ainda muito em aberto e que a maioria dos autores de trabalhos sobre optimização de trajectória faz questão de referenciar. Constatou-se que os dois métodos convergiram para soluções não muito dispares entre eles e em relação ao que outros autores obtiveram. Contudo o método pseudoespectral aqui utilizado dá soluções coerentes relativamente aos autores do problema.

Quanto ao tema deste trabalho, o que se fez foi separar o problema de navegação do problema de controlo, assim determina-se uma trajectória nominal para uma determinada missão e depois controla-se a aeronave segundo essa trajectória.

No problema de optimização de trajectórias 4D, utilizou-se os métodos de transcrição directa para que fosse possível transformar o problema de optimização de trajectória num problema de programação não linear. A abordagem é diferente nos dois métodos, mas em ambos foi necessário aplicar uma mudança de escala pois o programa computacional utilizado para resolver o problema de programação não linear padece de falta de rapidez na convergência e de refinamento da solução. Para ambas as missões a maneira como se procedeu na transcrição directa foi impor que os nós da trajectória fossem os *waypoints*, assim os nós t_k são os tempos de chegada a cada *waypoint*. Já no método pseudoespectral tal não pode acontecer pois os polinómios de Lagrange são construídos nos nós de Chebyshev, e necessário aplicar uma mudança de variável ao tempo para que seja possível passar para o espaço de Chebyshev.

Aplicou-se este problema a duas missões concretas. A primeira para uma missão com início em Castelo Branco até à Covilhã com passagem pelo Fundão. Esta missão é directa, apenas se definiu *waypoints* com diferentes altitudes para que se pudesse simular uma missão idêntica a um voo comercial. A segunda missão a ser aplicada este problema é a uma missão em torno da Covilhã, com início e fim no aerodromo (LPCV).

No momento de planear o voo é necessário ter em conta algumas características dinâmicas da aeronave (Velocidade e Ângulo trajectória máximo e mínimo), no presente estudo

aplicou-se ao UAV SkyGu@rdian. Os *waypoints* não só dão a indicação do ponto tridimensional que a aeronave deve sobrevoar como também o tempo a que deve chegar a este, é a isto que se denomina por navegação 4D. Então, o que se pretende neste problema é minimizar os atrasos de passagem em cada *waypoint* de modo a que se consiga executar toda a missão.

O problema do número de nós a aplicar à trajectória quando se utiliza o método da Colocação é então, igual ao número de *waypoints* (caso contrário não é possível obter uma trajectória nominal óptima), pelo que se deve ter um número de *waypoints* que permita que o passo de integração seja suficiente para não acumular erros. No método pseudoespectral o número de nós é um problema em aberto, porque depende também da ferramenta computacional utilizada.

Nas missões a que foram aplicadas o problema de optimização, o método pseudoespectral consegui encontrar trajectórias óptimas, já o método da Colocação para a missão em circuito não consegui convergir para uma solução. Na aplicação para a missão de Castelo Branco à Covilhã o resultado, na utilização dos dois métodos foi satisfatório, podendo-se concluir que para trajectórias com pouca curvatura o método da Colocação funciona perfeitamente apesar do método pseudoespectral permitir obter uma trajectória mais suave. Relativamente à missão em circuito, a conclusão é imediata, pois o método pseudoespectral é o consegue obter trajectórias óptimas. É de salientar que pode ser possível obter trajectórias ainda mais suaves para a missão em circuito, o facto é que o programa computacional não permite experimentar um maior número de nós, pois o número das variáveis de programação não linear aumenta nove vezes mais (para este problema).

Determinada a trajectória de referência utilizou-se um tipo de controlo não linear denominado por controlo predictivo de passo único. Este controlo mostrou ter capacidades de controlar a aeronave ao longo da trajectória, o facto de se ter apenas simulado um troço da missão em circuito foi devido ao facto de se não possuir ferramentas capazes de resolver problemas de programação não linear em tempo útil e com bastante precisão. O que é certo é que ficou comprovado que o controlo predictivo dá excelentes indicadores que pode ser possível controlar uma aeronave.

Em suma, o tema do trabalho proposto foi ultrapassado. Consegui-se modelar e resolver o problema de optimização e controlo de trajectórias 4D definidas por *waypoints*. Os resultados obtidos foram bastante animadores pois não tendo acesso programas computacionais arquitectados em linguagens mais rápidas e com melhores algoritmos, conseguiu-se à mesma comprovar e resolver um problema ainda não estudado. Futuramente este

tema pode ser explorado, na tentativa de encontrar maneiras de suavizar o os controlos no problema de optimização de trajectórias 4D, e ainda ser testado em programas computacionais mais rápidos para poder corroborar as conclusões deste trabalho.

Apêndice A

Dados do SkyGu@rdian

m	c	b	I_x	I_y	I_z	I_{xz}
[kg]	[m]	[m]	[kg.m ²]	[kg.m ²]	[kg.m ²]	[kg.m ²]
25	0.40	4.32	7.06	9.30	16.13	0.08

CL_0	CL_α	CL_{δ_e}	CL_q	CD_0	K_{CD}	CY_β	CY_{δ_a}	CY_{δ_r}
0.5986	4.6910	0.4051	0.0000	0.0336	0.0293	-0.3132	0.0000	0.0365

Cm_0	Cm_α	Cm_{δ_e}	Cm_q	$Cm_{\dot{\alpha}}$	Cl_β	Cl_r	Cl_p	Cl_{δ_a}
0.0392	-0.2367	-5.5262	-0.8544	-2.1104	-0.0583	0.1731	-0.5399	0.0570

Cl_{δ_r}	Cn_β	Cn_p	Cn_{δ_a}	Cm_{δ_r}
0.0355	-0.0211	-0.0738	-0.0063	-0.0076

Bibliografia

- [1] F. Benyah and L. S. Jennings. Regularization for optimal control computation. Report WA 6907, Centre for Applied Dynamics and Optimization, Department of Mathematics, University of Western Australia.
- [2] F. Benyah and L. S. Jennings. The l-curve in regulation of optimal control computation. *J. Austral. Math. Soc.*, 40:138–172, September 1998.
- [3] J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, March-April 1998.
- [4] K. Bousson. Chebychev pseudospectral trajectory optimization of differential inclusion models. In *SAE World Aviation Congress, 2003 Aerospace Congress and Exhibition, Montreal, Canada*, pages 2003–01–3044. SAE, September 2003.
- [5] K. Bousson. Single gridpoint dynamic programming for trajectory optimization. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, pages AIAA 2005–5902. AIAA, August 2003.
- [6] E. M. da G.P. Fernandes. *Computação Numérica*. Universidade do Minho, 1997.
- [7] F. Fahroo and I. M. Ross. Costate estimation by a legendre pseudospectral method. *Journal of Guidance, Control, and Dynamics*, 24(2):270–277, March-April 2001.
- [8] D. G. Hull. Conversion of optimal control problems into parameter optimization problems. *Journal of Guidance, Control, and Dynamics*, 20(1):57–60, January-February 1997.
- [9] P. L. Innocenti, M. and D. Turra. Guidance of unmanned air vehicles based on fuzzy sets and fixed waypoints. *J. Guidance*, 27(5):715–719, November 2003.
- [10] P. Lu. Regulation about time-varying trajectories: Precision entry guidance illustrated. *Journal of Guidance, Control, and Dynamics*, 22(6):784–790, November-December 1999.

-
- [11] G. Moon and Y. Kim. Flight path optimization passin through waypoints for autonomous flight control systems. *Engineering Oprimization*, 37(7):755–774, October 2005.
- [12] I. Whang and T. Whang. Horizontal waypoint guidance design using optimal control. *IEEE Trans. Aero. Electron. Syst*, 38(3):1116–1120, July 2003.