

UNIVERSIDADE DA BEIRA INTERIOR



UBI
Covilhã
Portugal

Navegação Autónoma em Loiters Normalizados pela OTAN

Elaborado por
Albano Samuel Queirós Constante

Orientado por
Prof. Kouamana Bousson

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

na
Faculdade de Engenharia
Departamento de Ciências Aeroespaciais

Junho de 2010

*“Para ser grande, sê inteiro: nada
Teu exagera ou exclui.
Sê todo em cada coisa. Põe quanto és
No mínimo que fazes.
Assim em cada lago a lua toda
Brilha, porque alta vive”*

Fernando Pessoa

UNIVERSIDADE DA BEIRA INTERIOR

Resumo

Faculdade de Engenharia

Departamento de Ciências Aeroespaciais

Mestre Engenharia Aeronáutica

Elaborado por Albano Samuel Queirós Constante

Actualmente, o desenvolvimento de orientação, navegação e sistemas de controlo de aeronaves tem sido uma área de forte exploração. Com a necessidade de resposta a sistemas mais confiáveis e robustos, tem existido uma criação de inúmeros métodos relacionados com o desempenho da aeronave.

Pretende-se com este trabalho, propor um método de navegação autónoma em loiters normalizados pela OTAN. Para tal utiliza-se, um método não linear de controlo feedback baseado nas derivadas de Lie para a orientação de navegação de veículos aéreos não tripulados (UAV). Parametrizando cada loiter, obtemos para cada um dos casos, uma trajectória quadridimensional que permitiu uma orientação para uma aeronave, sem ultrapassar as restrições das variáveis de controlo.

A fim de validar o método de navegação autónoma proposto, foram realizadas varias simulações, claramente descritas, utilizando um software de cálculo numérico (*MatlabTM*).

UNIVERSIDADE DA BEIRA INTERIOR

Abstract

Faculdade de Engenharia

Departamento de Ciências Aeroespaciais

Mestre Engenharia Aeronáutica

Elaborado por Albano Samuel Queirós Constante

Currently, the development of guidance, navigation and control systems of aircraft has been an area of intense exploration. In response to the need for more reliable and robust systems, several methods have been created and related to the performance of the aircraft.

With this work we propose a method of autonomous navigation on loiters standardized by NATO. For this, we use a nonlinear feedback control based on the Lie derivatives to guide the navigation of unmanned aerial vehicles (UAV). Parameterizing each loiter, we obtain for each case, a path that enables a four-dimensional orientation of an aircraft without going beyond the constraints of the control variables.

To validate the proposed autonomous navigation method, we carried out several simulations, clearly described, using the numerical calculation software (*MatlabTM*).

Agradecimentos

Para que a realização desta dissertação fosse possível, não me limitei a contar comigo! Mas, com confiança, humildade e perseverança, tive de solicitar por diversas vezes o auxílio de vários “amigos”, porque no final, penso tratá-los a todos de “amigos”. Por essa razão, quero em primeiro lugar dirigir os mais sinceros agradecimentos ao meu professor, Doutor K. Bousson, por ter sido o meu guia e consultor, já que foi graças a ele que esta dissertação nasceu e se desenvolveu, sob o olhar atento do “mestre”.

Não podia deixar também de agradecer a toda a minha família em especial aos meus pais e aos meus irmãos, bem como aos meus companheiros e amigos durante estes seis anos, em especial ao André Silva e ao José Novais. Deixo também o meu apreço ao Paulo Machado, à Mafalda Gouveia e à Filipa Romão por todo o apoio e motivação dado.

Obrigado

Conteúdo

Resumo	ii
Abstract	iii
Agradecimentos	iv
Lista de Figuras	vii
Lista de Tabelas	viii
Nomenclatura	ix
1 A Problemática do Controlo Autónomo de Trajectória	1
1.1 Arquitectura Funcional do Sistema de Controlo de UAV [1]	3
1.1.1 Core UCS [1]	5
1.1.2 Data Link Interface [1]	6
1.1.3 The Command and Control Interface [1]	7
1.1.4 Vehicle Specific Module [1]	8
1.1.5 Human Computer Interface [1]	9
1.1.6 Command and Control Interface Specific Module [1]	12
1.2 Loiters	12
1.2.1 Descrição	12
1.2.2 Algoritmo de navegação em loiter [2]	13
1.2.2.1 Navegação num loiter Circular (Figura 1.6)	13
1.2.2.2 Navegação em Hipódromo e na “Figura-8”(Figura 1.6)	14
1.3 Controlo de Trajectórias de Referência	16
1.3.1 Trajectórias Tridimensionais para Aeronaves Autónomas	16
1.3.2 Optimização e Controlo de Trajectrias 4D Definidas por <i>Waypoints</i> [3]	18
1.3.2.1 Formulação do Problema	18
1.3.2.2 Modelo Dinâmico	19
1.3.2.3 Resolução do Problema de Navegação 4D	20
1.3.3 Orientação Segundo uma Trajectória Desejada Definida por <i>Waypoints</i> [4]	22
1.3.4 Síntese	25
1.4 Objectivo da Dissertação	26
2 Método de Navegação Autónoma em Loiters	27

2.1	Controlo Não Linear	27
2.2	Modelo Dinâmico de Navegação	29
2.3	Loiters	31
2.3.1	Loiter Circular	31
2.3.2	Loiter “Figura-8 ”	32
2.3.3	Loiter Hipódromo	35
2.4	Algoritmo	38
3	Simulação e Resultados	39
3.1	Simulação do Loiter Circular	40
3.2	Simulação do Loiter “Figura-8 ”	43
3.3	Simulação do Loiter Hipódromo	45
4	Conclusões	47
	Bibliografia	49
	Anexos	50

Lista de Figuras

1.1	Exemplo de missões de UAV	1
1.2	Arquitetura Funcional UCS [1]	3
1.3	Papel do DLI [1]	7
1.4	Tipos de dados CCI [1]	8
1.5	Função do VSM [1]	10
1.6	Parâmetros dos loiters [1]	13
2.1	Loiter Circular	32
2.2	Loiter Forma-8	32
2.3	Loiter Forma-8	35
3.1	Loiter circular - 3D	40
3.2	Loiter circular - Vista superior	41
3.3	Loiter circular - Erro experimental	41
3.4	Loiter “Figura-8 ”- 3D	43
3.5	Loiter “Figura-8 ”- Vista superior	43
3.6	Loiter “Figura-8 ”- Erro experimental	44
3.7	Loiter hipódromo - 3D	45
3.8	Loiter hipódromo - Vista superior	45
3.9	Loiter hipódromo - Erro experimental	46

Lista de Tabelas

1.1	Parâmetros de loiter [2]	15
3.1	Valores iniciais	39

Nomenclatura

OTAN	Organização do Tratado do Atlântico Norte
STANAG	NATO Standardisation Agreement
UAV	Unmanned Aerial Vehicle
AV	Aerial Vehicle
NIIA	NATO ISR Interoperability Architecture
ISR	Intelligence Surveillance Reconnaissance
LOI	Level of Interoperability
CONOPS	Concept of Operations
UCS	UAV Control System
C4I	Command, Control, Communications, Computers and Intelligence
CUCS	Core UAV Control System
DLI	Data Link Interface
VSM	Vehicle Specific Module
CCISM	Command and Control Interface Specific Module
HCI	Human Computer Interface
GDT	Ground Data Terminal
GPS	Global Positioning System
PMP	Princípio Máximo de Pontryaguin
SW	Synthetic-Waypoint
SW	Synthetic-Waypoint Guidance
DTW	Direct - To - Waypoint
TTW	Track - To - Waypoint
PMP	Princípio Máximo de Pontryaguin
2D	Duas Dimensões
3D	Três Dimensões

$4D$	Quatro Dimensões
Q, Q_1, Q_2	Peso das Matrizes de Estado
e	Erro
g	Aceleração Gravítica
γ	Ângulo de Trajectória
h	Passo - Tempo
q	Trajectória de Referência
V	Velocidade
w	Velocidade Angular
x	Vector do Sistema de Estado
u	Vector do Sistema de Controlo
x	Coordenada Cartesiana
y	Coordenada Cartesiana
ψ	Ângulo de Rumo
z	Altitude
R	Raio de Loiter
d	Comprimento de Loiter

Capítulo 1

A Problemática do Controlo Autónomo de Trajectória

A recente procura de veículos aéreos não tripulados para vários tipos de aplicações civis e militares, Figura 1.1, tem originado um progresso na tecnologia de orientação baseada no sistema de posicionamento global, através de coordenadas que identificam um ponto num espaço físico, designados por *waypoints*.

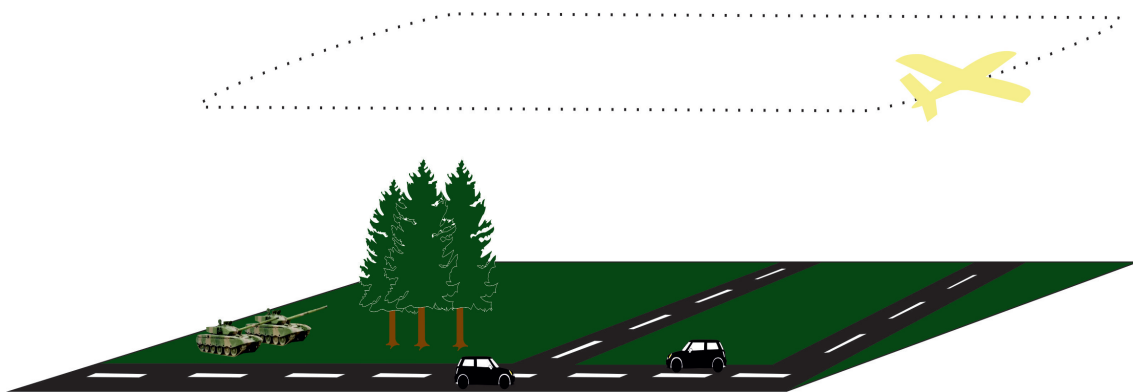


FIGURA 1.1: Exemplo de missões de UAV

A orientação para a navegação consiste em fornecer referências (por exemplo, velocidade, posição e ângulo de trajectória de voo) para o módulo de controlo, de modo a gerar uma trajectória de voo passando por uma sequência especificada de *waypoints* dados, vindo de um banco de dados de *waypoint* fornecido por um operador. Uma missão comum para o transporte é normalmente dividida em fases diferentes, como descolagem, subida, cruzeiro, aproximação e

aterragem. Um UAV é normalmente projectado para missões alternativas. Fases de voo, como subidas ou *loiters* (fase de voo que consiste num voo cruzeiro sobre uma região delimitada num determinado tempo), são previamente planeadas e cada uma delas é definida por uma trajectória particular a fim de respeitar certas restrições e especificações impostas pelas autoridades, operadores e/ou pelos projectistas. Normalmente, um controlador automático específico de voo é projectado para uma determinada fase do voo.

A navegação por *waypoint* é principalmente encontrada em missões para veículos aéreos tripulados. No caso do voo de veículos aéreos não tripulados, a navegação pura por *waypoint* ocorre apenas numa parcela relativamente pequena da missão. Suponha-se que a um UAV tenha sido atribuído como tarefa a vigilância de uma área, por exemplo, no caso de acompanhamento de incêndios ou vigilância de fronteiras, ou mesmo em caso de imagens de área. Nessas situações, a trajectória do veículo deve seguir um determinado padrão, que pode diferir de um mero *waypoint* de navegação. No entanto, por mais que a trajectória de referência seja complicada, pode sempre ser referenciada numa sequência de *waypoints*. No caso de navegação por *loiter*, a altitude de voo constante, é possível restringir o modelo dinâmico em duas dimensões, pois o *loiter* geralmente ocorre em altitude constante. O *loiter*, trajectória contínua, consiste numa trajectória em voo de cruzeiro para um determinada quantidade de tempo sobre uma área específica. A fase de *loiter* ocorre, para a aviação em geral, geralmente no final do plano de voo, normalmente, quando o avião aguarda uma autorização a aterragem. No entanto, algumas aeronaves não tripuladas utilizam-na para fins especiais, como reconhecimento, controlo ou vigilância, pode haver uma fase em *loiter* pleno voo. A STANAG 4586 [1] define um conjunto de padrões *loiter* ao longo do qual um UAV deve navegar em caso de vigilância e monitorização de uma área.

A STANAG 4586 é um protocolo padronizado pela OTAN para o controlo de UAVs. A sua finalidade é promover a interoperabilidade no controlo e comunicação com UAVs para os membros da OTAN. A capacidade para controlar mutuamente os veículos aéreos e as suas cargas, bem como a rápida divulgação da informação resultante de diferentes escalões de comando que requerem sistemas utilizados de UAV que sejam interoperáveis. Actualmente, muitos sistemas de UAVs não são completamente interoperáveis. As interfaces definidas no NIIA fornecem a interoperabilidade dos sistemas ISR nos níveis 1 e 2 para sensores digitais compatíveis com as STANAGs referenciadas e aplicáveis. Os sistemas de UAVs têm sido projectados e adquiridos, e contêm elementos do sistema que geralmente são únicos e específicos. Estes não têm interfaces de padrão entre os elementos do sistema o que resulta numa variedade de sistemas não interoperáveis. A fim de permitir a interoperabilidade dos sistemas de UAVs, a aplicação de normas

importantes para um sistema de interfaces e de funções são exigidas. O objectivo da STANAG 4586 é especificar as interfaces que devem ser implementadas a fim de alcançar o necessário e operacionalmente viável LOI de acordo com o respectivo sistema de UAVs CONOPS (Conceito de Operações) aplicável ao sistema específico e teatro de operações. Isso será realizado através da implementação de interfaces padrão no sistema de controlo UAV (UCS) para se comunicar com diferentes UAVs e cargas úteis, bem como com diferentes sistemas C4I. A implementação de interfaces padrão facilitará também a integração de componentes de diferentes fontes, bem como a interoperabilidade dos sistemas legados. As normas na STANAG 4586, que são identificadas como obrigatórias, devem ser implementadas a fim de obter a carta de intenções necessárias. O cumprimento da STANAG 4586 vai facilitar este processo e, provavelmente, as combinações de sistemas de UAV podem ser certificadas com antecedência.

1.1 Arquitectura Funcional do Sistema de Controlo de UAV [1]

A Arquitectura Funcional UCS necessária para suportar a interoperabilidade entre os futuros sistemas de UAV é ilustrada na Figura 1.2.

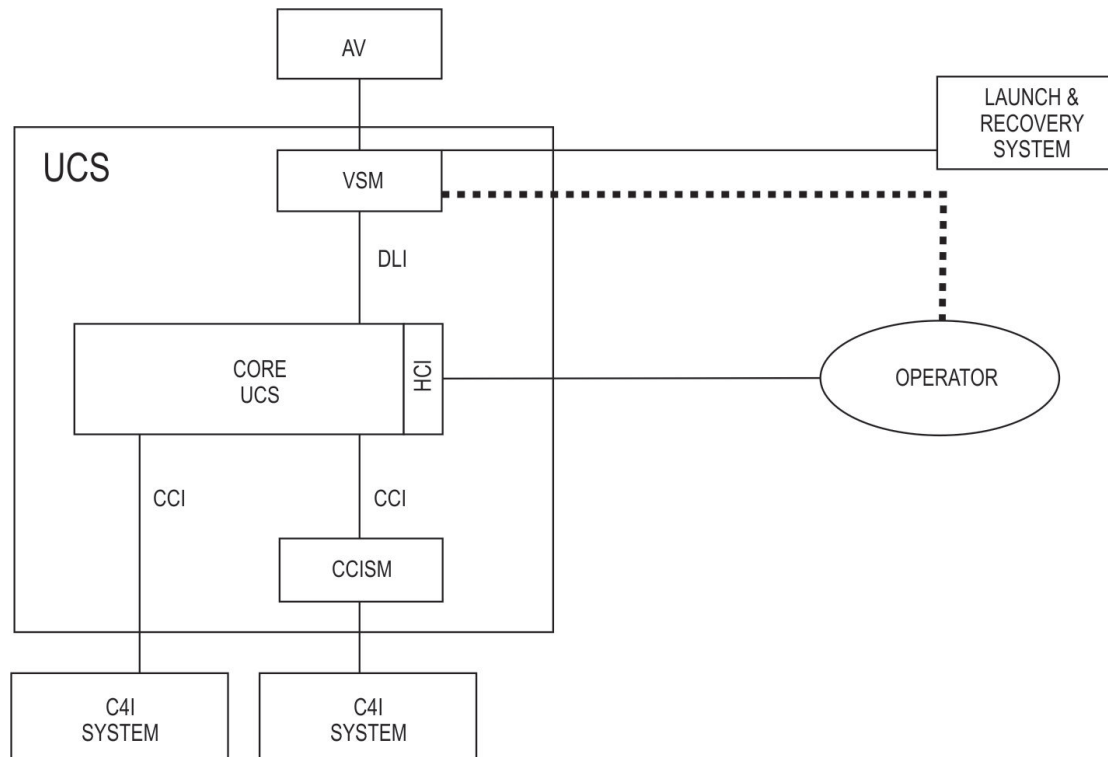


FIGURA 1.2: Arquitectura Funcional UCS [1]

Esta arquitectura estabelece os seguintes elementos funcionais e interfaces:

- Core UAV Control System
- Data Link Interface
- Command and Control Interface
- Vehicle Specific Module
- Command and Control Interface Specific Module
- Human Computer Interface

A STANAG 4586 [1] não é uma tentativa para definir um projecto detalhado nem uma implementação para o CUCS que não especifica que a acomodação da arquitectura funcional face à integração do DLI e CCI, mas recomenda que se siga a aplicação das normas da OTAN e directrizes para o software. Devido à evolução tecnológica, a STANAG 4586 não define um específico COE, mas apenas distingue que o ambiente operacional suporta/integra a rede específica/protocolos de transporte e suporta as aplicações de cada utilizador. No futuro, bem como sistemas legados de UAV, será possível agir com a interoperabilidade através da conformidade com essa arquitectura e com as normas relevantes. O DLI apoiará o legado, bem como os futuros UAVs e toda a tecnologia de veículos aéreos e também todos os propósitos operacionais de um UAV (vigilância, reconhecimento e combate).

Os futuros sistemas de UAV devem utilizar um sistema de ligação de dados compatível com a STANAG 7085. A STANAG 7085 [5] estabelece as normas de interoperabilidade para três classes de dados utilizados para ligações de transmissões de dados ISR. Para aqueles que não o utilizarem, um ADT compatível com GDT deve ser fornecido, a fim de atingir o nível 2, além da interoperabilidade. De modo semelhante, a interface do CCI apoiará o legado, bem como os futuros sistemas C4I. Assim, a interface entre o CUCS e os nós externos C4I devem ser compatíveis com a infra-estrutura do sistema de comunicação utilizado para apoiar as tarefas externas e a divulgação de dados do sensor.

O conceito de uma função VSM é introduzido e fornece unicamente/exclusivamente protocolos de comunicação, calendário de interface e os formatos de dados que os respectivos veículos aéreos requerem. A VSM também irá fornecer qualquer tradução necessária dos protocolos DLI e formatos de mensagens de requisitos para um único veículo aéreo, desde que seja única para

cada veículo aéreo, o fabricante do veículo aéreo, em geral, pode fornecê-lo. Se as ligações de dados utilizados no sistema de UAV não são compatíveis com a STANAG 7085 então, o GDT associado a uma inconformada ligação de dados deve ser fornecida e a interface com a UCS é feita através da função DLI-VSM ou a capacidade de receber e processar dados DLI, especificando a ligação de controlo de dados e mensagens de estado que deve ser incorporado na GDT. A função VSM pode ser hospedada no veículo aéreo e/ou no solo.

O CCISM fornece uma função semelhante à do VSM, isto é, o encapsulamento dos dados CCI e qualquer tradução necessária que seja compatível/interoperável com as ligações de comunicação física entre a UCS e os sistemas C4I. O CCISM pode ser hospedado e co-instalado com o UCS ou por o nó de ligação C4I. A arquitectura do UCS deve prever a integração de um CCISM. O operador do sistema UAV deve fornecer um conjunto padronizado de parâmetros para que o operador possa utilizar para operar/acompanhar o UAVs que lhe fora atribuído. Isto inclui a sua interface com o controlo de autoridade gestão aérea.

O DLI e CCI devem ser implementados com mensagens. Além de apoiar um conjunto de mensagens genéricas, o CUCS será capaz de suportar a capacidade de visualização remota. A capacidade de exibição remota suporta o controlo e acompanhamento de informações da VSM de um veículo específico, na CUCS, para a qual não há nenhuma interface genérica disponível. O quadro para a troca de informações permitirá que a informação seja capaz de passar de um processo para outro na mesma plataforma, entre processos em diferentes plataformas, e mesmo entre diferentes produtos de software e sistemas operacionais.

Os desenvolvedores do VSM são livres de determinar o método de execução de intercâmbio das informações necessárias entre o CUCS e o VSM, utilizando serviços identificados. Esta abordagem centra-se na troca de informações de uma forma que efectivamente exhibe informações de estado do específico veículo aéreo e prevê um controlo efectivo sobre a funcionalidade específica desse veículo.

1.1.1 Core UCS [1]

O CUCS é o principal item na arquitectura de um sistema de controlo de UAV. O CUCS consiste no hardware e no software que, no mínimo, implementa uma interface de ligação de dados, e os itens de configuração de software de computador necessários e do CUCS/HCI. Ele fornece um operador de UAV com a funcionalidade de controlar o UAV em todas as fases de uma missão.

Também fornece uma interface gráfica para ajudar o operador de controlo do veículo. O CUCS controla e monitoriza a aeronave através da comunicação com os veículos (VSM). Cada CUCS pode ser conectado a diversos módulos específicos do veículo, e cada VSM pode ser conectado a diversos CUCSs. Para ajudar a coordenar a rede de CUCSs e VSMs, comando, controlo, comunicações, computadores e informação (C4I) podemos utilizar nós. Dependendo do adequado LOI e das cargas suportadas pelo respectivo Sistema de UAV, o CUCS deve fornecer:

- A funcionalidade e capacidade para receber, processar e divulgar dados do veículo aéreo e de carga, de realizar o planeamento da missão, acompanhar e controlar a carga, monitorar e controlar o veículo aéreo e acompanhar e controlar as ligações de dados;
- Uma arquitectura adicional de suporte de software aberto para futuros veículos aéreos e capacidade de carga;
- O operador de UAV com as ferramentas necessárias, relacionadas com comunicações, tarefas da missão, planeamento da missão, execução e monitorização da missão, recepção de dados, processamento e disseminação de dados;
- A capacidade de acolher a VSM e funções do CCISM.

A CUCS deve ser capaz de receber e enviar certas mensagens de/para as VSMs. Por exemplo, um CUCS conecta a um VSM, enviando um pedido de autorização CUCS para um VSM específico, ou transmitir o pedido a cada VSM disponível. Irá então receber uma resposta do VSM indicando se a VSM permite que o CUCS controle ou monitorize, e nesse caso, em que nível de interoperabilidade (LOI). Depois do CUCS ter sido autorizado pela VSM, pode controlar e monitorizar os VSMs do veículo e/ou cargas permitidas na LOI. Para controlar o veículo aéreo o CUCS precisa de enviar mensagens, tais como comandos de direcção e comandos da missão a carregar, e receber mensagens, por exemplo, contendo os estados de inércia ou estados de operação dos veículos.

1.1.2 Data Link Interface [1]

A DLI é a interface entre CUCSs e VSMs. Prevê um conjunto de mensagens padronizadas para permitir a comunicação entre uma grande variedade de veículos aéreos e estações de controlo. Não é discriminado que tipo de ligação física o DLI deve usar.

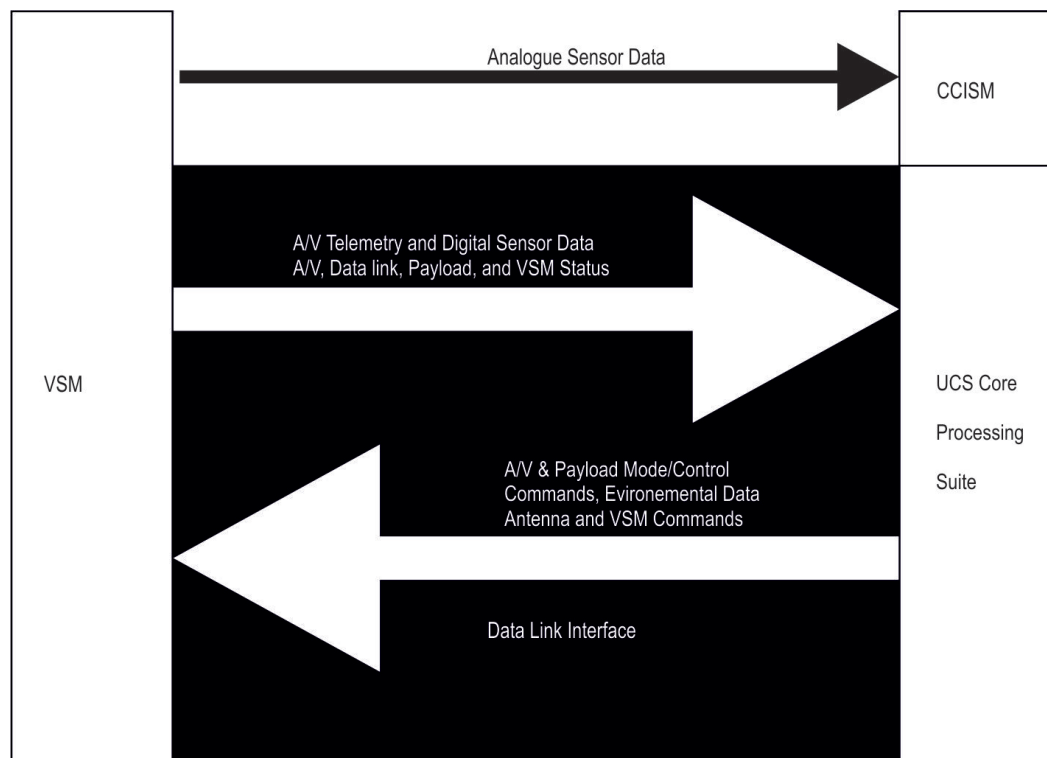


FIGURA 1.3: Papel do DLI [1]

O DLI permite a comunicação entre o VSM e o elemento CUCS do UAV. No princípio a estrutura de mensagens é fornecido pelo DLI com o objectivo de passar o controlo e a informação de estado para os veículos aéreos UAV, cargas úteis, links de dados, bem como advertências e cuidados entre o CUCS e o VSM, sem criar dependências entre os dois componentes. O VSM pode afectar o HCI no CUCS através de “serviços” de metodologia da DLI permitindo que o operador de UAV interaja com o veículo aéreo UAV através da VSM para seleccionar as opções, modos de operação, e outras acções do UAV.

1.1.3 The Command and Control Interface [1]

O CCI é a interface entre o CUCS e os sistemas externos C4I. Ele especifica as exigências de dados que devem ser adoptados para a comunicação entre o núcleo do UCS e todos os usuários finais C4I através de uma interface padrão comum. O CCI está destinado a cobrir todos os tipos de mensagens e dados que precisam de ser trocados nos dois sentidos entre o CUCS e os sistemas C4I durante todas as fases de uma missão de UAV, incluindo:

- Antes do voo: mensagens agendadas, situação táctica, dados ambientais, restrições gerais da missão e os planos da missão;
- Durante o voo: estados e mensagens de serviço, dados da carga, relatórios de progresso;
- Após o voo: estados e mensagens de serviço, dados da carga, relatórios de exploração pós voo, relatórios de missão.

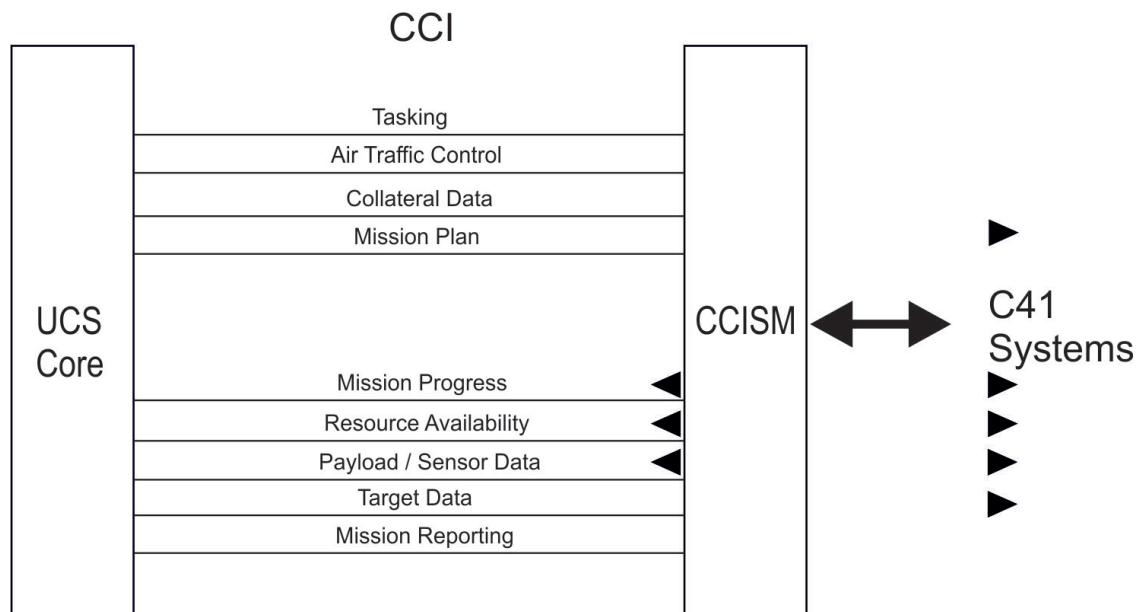


FIGURA 1.4: Tipos de dados CCI [1]

1.1.4 Vehicle Specific Module [1]

A VSM é uma função específica do veículo aéreo que serve a interface DLI entre os elementos CUCS e o sistema do veículo aéreo. O VSM deve isolar o CUCS de especialidades específicas de interface do veículo aéreo através da manutenção de controlo de circuito fechado e comunicação com o veículo aéreo e a sua carga útil seguindo os protocolos específicos do veículo aéreo, o calendário e os métodos de codificação. A de desembarque em movimento, de recuperação de emergência, etc. O VSM deve ser responsável pelas seguintes funções onde não fazem parte do AV ou do GDT:

- Traduzir os dados a partir da representação utilizada pelo CUCS (mensagens definidas pelo DLI) para específicas representações do veículo de, e vice-versa.

- Agindo como um depósito e servidor de dados para um específico veículo (como a configuração do veículo e as limitações de desempenho) e métodos.
- Empacotamento e desempacotamento de uma ligação de dados para otimizar a largura de banda de transmissão, quando necessário.
- Gestão de interfaces necessárias para controlar e monitorizar dados da operação.
- Gestão de interfaces necessárias para controlar e acompanhar o lançamento e a recuperação (L/R) de sistemas relacionados com os respectivos veículos.

O módulo VSM recebe dados do DLI e transmite os dados para um veículo aéreo. Este garante a conformidade com protocolos como o STANAG 4586, e actua como uma ponte entre a DLI e o veículo aéreo. Um VSM pode controlar um ou mais aviões e um CUCS pode ligar-se ao VSM para assumir o controlo, ou a acompanhar, uma aeronave ou uma carga transportada por uma aeronave. A VSM pode estar localizado na mesma máquina que o CUCS, ou numa máquina diferente no solo, ou pode ser dividido entre um terminal de dados no terreno, e um terminal de dados aéreo transportado a bordo de um veículo aéreo. Cada VSM pode controlar várias aeronaves, bem como estando ligado a diversos CUCSs. Quando um VSM está ligado a vários CUCSs, apenas um deles pode controlar cada aeronave num determinado momento. No entanto, os outros CUCSs podem receber dados, e controlar o peso nas aeronaves, se a VSM permitir. Depois da VSM ter autorizado o CUCS para um certo nível de interoperabilidade, a VSM precisa estar pronta para receber outras mensagens do CUCS, tais como mensagens de configuração e comandos de direcção. A VSM também envia continuamente mensagens de estado para autorizar os CUCSs, dizendo-lhes os estados de inércia, estados de operação dos veículos aéreos, etc.

1.1.5 Human Computer Interface [1]

O HCI é um elemento integrante do CUCS. Os requisitos do HCI devem facilitar a integração dos sistemas de UAV da OTAN numa infra-estrutura de campo de batalha conjunta da OTAN através de cinco níveis de interoperabilidade (LOI). O HCI tem definido um conjunto de requisitos obrigatórios e recomendações para permitir a interoperabilidade entre o utilizador de UAV das Nações OTAN. Estão classificados sob os seguintes títulos:

- Requisitos Gerais

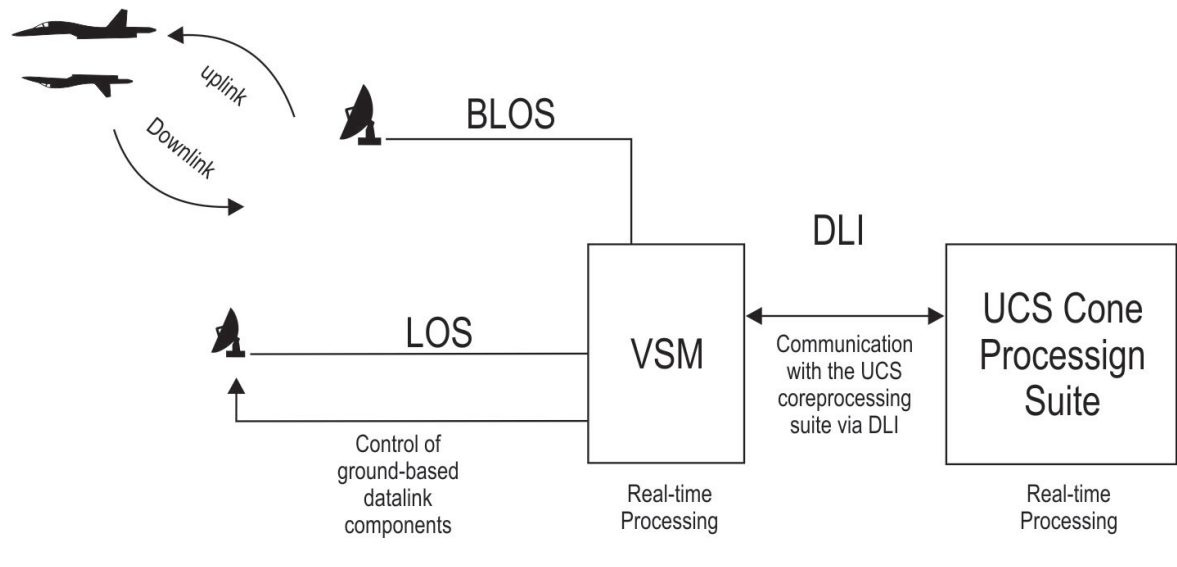


FIGURA 1.5: Função do VSM [1]

O operador deve ter a capacidade de inserir e sincronizar uma vez com o sistema UAV e com sistemas C4I aplicáveis.

- Configuração do UCS

O HCI deve fornecer ao operador a capacidade de gerar, receber, visualizar, editar e enviar tipos de mensagens que foram definidas na STANAG 4586 como aplicáveis ao necessário LOI. O operador deve ser também capaz de alterar globalmente as unidades de medida.

- Planeamento da Missão

O planeamento da missão inclui todos os aspectos de planeamento de todas as fases da missão contidas no MO/ATO (por exemplo, antes e no decorrer do voo para o AV (aerial vehicle), de carga, de ligação de dados e de comunicações). Para os CUCS's que tenham a capacidade de gerar plano(s) de missão, o HCI deve permitir um operador qualificado para criar, editar e salvar um plano ou planos de missão. Para planos de missão desenvolvidos externamente ao CUCS, o HCI deve permitir um operador qualificado para importar, visualizar e salvar um plano de missão. O HCI deve permitir um operador qualificado para actualizar (sem enviar) um plano de missão actual a qualquer momento antes ou durante o voo. O HCI deve fornecer um operador qualificado com a capacidade de transmitir para o AV planos de missão.

- Controlo do Veículo Aéreo

O HCI deve prever controlos e displays para o controlo/accompanhamento do veículo em todos os apoiados modos de voo. O operador deve ter a capacidade de passar o controlo do AV para outro UCS com um operador, qualificado, e acompanhar o estado da entrega conforme os mecanismos definidos no DLI.

- Operador de Monitorização e Controlo

O HCI deve fornecer uma visualização da imagem para permitir ao operador visualizar arquivos de imagem de fontes externas ao C4I.

- Controle e Monitorização da Carga

As funções de controlo de carga HCI definido dentro do CUCS serão genéricas para os tipos de carga, ao invés de cargas específicas, sempre que possível. Os tipos de carga são sensores de imagem (passivos), sensores de imagem (sensores activos/radar), Laser Based Payloads, repetição de comunicações, lojas(armazém) de cargas. O HCI deve prever os controlos suficientes e displays para controlar cargas e todas as funções associadas para apenas aquelas cargas que tenham sido validadas com o CUCS actual. Um operador qualificado deve ter a capacidade de transmitir e de receber controlo (transferência) de carga útil do AV de/para outro sistema de controlo e monitorização de carga através dos mecanismos pelo DLI. Para cargas que geram imagens de movimento, o operador deverá ter um visor de imagens de movimento. Lojas (de venda) de cargas são consideradas aquelas que estão dispensados do UAV como parte dos objectivos da missão UAV. Isso pode incluir o lançamento de armas ou de implantação de sensores remotos, etc. O mecanismo de libertação para a carga deve ser claramente identificado e rotulado ao operador. Um bloqueio de segurança deve funcionar de tal forma que o operador não possa libertar a carga inadvertidamente.

- Advertências, Cuidados e Avisos

Avisos, advertências e alertas informam o operador sobre qualquer condição incomum ou crítica. O HCI deve fornecer a capacidade de exibir e gerir alertas, advertências e alertas definidos na STANAG 4586.

- Gestão das Comunicações

A Gestão das Comunicações controla as ligações de comunicações entre a UCS e o UAV. Isto inclui qualquer tipo de antenas ou ligações de dados adicionais necessários para suportar uma carga específica (por exemplo, uma carga CR). Ele fornece ao operador no CUCS a capacidade de configurar as ligações de dados e de alterar um número de parâmetros do Terminal Aéreo de Dados (ADT) e do GDT. Embora a maioria dos parâmetros de ligação de dados seja controlada a partir do CUCS, algumas funções específicas de ligações de dados terão de ser controlados através da DLI. O HCI deve fornecer ao operador a capacidade de abrir e controlar as ligações de comunicações entre o CUCS e outros órgãos externos, tais como, Sistemas C4I através da ligação CCI, Controle de tráfego aéreo por meio de voz e ligações de dados, e ADT/GDT. O CUCS deve fornecer uma antena/dados de estado da ligação. Esta exposição não tem necessariamente de ser separada do controlo/monitorização do AV.

1.1.6 Command and Control Interface Specific Module [1]

O CCISM destina-se essencialmente para a comunicação com sistemas legados C4I que não são directamente compatíveis com padrões especificados na STANAG 4586, protocolos ou camada física. Quando os futuros sistemas C4I forem desenvolvidos espera-se que estes sejam compatíveis com a STANAG 4586, em cada caso haverá uma ligação directa sem a necessidade de uma CCISM intermediária.

1.2 Loiters

1.2.1 Descrição

A STANAG 4586 [1] especifica diferentes tipos de *loiters* que podem ser comandados pelo CUCS se o veículo aéreo tiver capacidade para que os possa implementar. Estes tipos de *loiters*, o círculo, o hipódromo e a “Figura-8”, [1], são mostrados na Figura 1.6. É possível configurar determinados parâmetros do *loiter*, tais como o raio do *loiter*, rumo, comprimento do *loiter*, etc., através de uma mensagem de configuração *loiter*.

Os parâmetros do *loiter* à passagem num determinado *waypoint* para a realização do *loiter* encontram-se descritos na tabela 1.1.

1.2.2 Algoritmo de navegação em loiter [2]

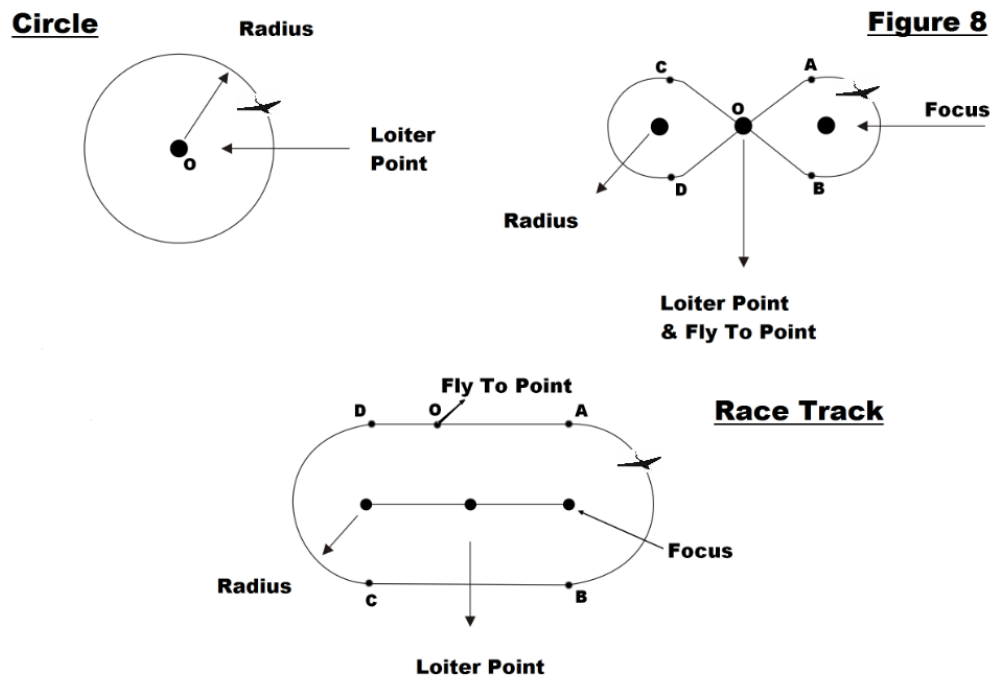


FIGURA 1.6: Parâmetros dos loiters [1]

1.2.2.1 Navegação num loiter Circular (Figura 1.6)

Seja t_{init} o tempo em que começa a navegação num *loiter* circular, e ψ_{init} é a posição inicial no *loiter*. A velocidade, o rumo e o ângulo trajectória de voo, Tabela 1.1, são referências calculadas da seguinte maneira a um tempo t :

1. $V_{ref} = l_{sp}$
2. $Se(l_{dn} = 1)$

$$\Psi_{ref}(t) = \Psi_{init} + \frac{V_{ref}}{l_{rd}}(t - t_{init})$$

3. $Senão(l_{dn} = 2)$

$$\Psi_{ref}(t) = \Psi_{init} - \frac{V_{ref}}{l_{rd}}(t - t_{init})$$

Fim

$$4. \gamma_{ref} = 0$$

No caso de um voo coordenado, o algoritmo de navegação do *loiter* circular, acima descrito, dá origem a um ângulo de inclinação ϕ_{ref} dado por:

$$\phi_{ref} = \tan^{-1} \left(V \frac{\dot{\psi}_{ref}}{g} \right)$$

onde $\dot{\psi}_{ref}$ é a derivada em ordem ao tempo de ψ_{ref} , g é a aceleração gravítica, e $V = l_{sp}$ é a velocidade de voo no *loiter*. Assumindo l_{sp} como constante, o algoritmo do *loiter* circular fornece uma navegação com um ângulo de rolamento constante.

1.2.2.2 Navegação em Hipódromo e na “Figura-8” (Figura 1.6)

No caso de navegação em *loiter* no sentido dos ponteiros do relógio: A navegação no *loiter* começa no ponto O, e continua como:

Enquanto (o tempo final de navegação do *loiter* não foi atingido)

1. Do ponto O realizar uma navegação por *waypoint* ate ao ponto A;
2. Realizar a navegação circular do *loiter* do ponto A para o ponto B;
3. Do ponto B deslocar-se para o ponto C pelo procedimento de navegação por *waypoint*;
4. Realizar a navegação circular do *loiter* do ponto C para o ponto D;
5. Do ponto D deslocar-se para o ponto O através procedimento de navegação por *waypoint*;

Fim

No caso da navegação se realizar no sentido oposto dos ponteiros do relógio todo este procedimento realizado anteriormente deve ser efectuado no sentido contrário.

TABELA 1.1: Parâmetros de loiter [2]

Parâmetros	Nomenclatura	Unidades	Alcance
Tipo de loiter	ι_p	<i>Enumerado</i>	1=Circular 2=Hipódromo 3=Figura-8
Raio de loiter Usado como raio para loiters circulares, bem como raio de semi-círculo em cada ponto final do loiter padrão.	ι_{rd}	<i>m</i>	$1 \leq \iota_{rd} \leq 100000$
Comprimento de loiter Usado para o hipódromo e para a Figura-8 para definir o comprimento, centrado no ponto de loiter na direcção do final rumo do loiter.	ι_{lh}	<i>m</i>	$1 \leq \iota_{lh} \leq 100000$
Rumo de loiter O parâmetro do rumo de loiter, referenciado no ponto de loiter, a partir do norte verdadeiro.	ι_{bh}	<i>rad</i>	$0 \leq \iota_{bh} \leq 2\pi$
Direcção de loiter Define a direcção da volta quando circula o ponto de loiter definido pelo <i>waypoint</i> da posição do veículo.	ι_{dn}	<i>Enumerado</i>	1=Sentido dos ponteiros do relógio 2=Sentido inverso dos ponteiros do relógio
Altitude de loiter	ι_{al}	<i>m</i>	$-1000 \leq \iota_{al} \leq 100000$
Tipo de altitude	a_{tp}	<i>Enumerado</i>	0 = Altitude de pressão 1 = Altitude barométrica 2 = AGL 3 = WGS-84
Velocidade de voo de loiter	ι_{sp}	<i>m/s</i>	$0 \leq \iota_{sp} \leq 10000$
Tipo de velocidade	s_{tp}		0 = IAS 1 = TAS 2 = Velocidade em relação ao solo

1.3 Controlo de Trajectórias de Referência

1.3.1 Trajectórias Tridimensionais para Aeronaves Autónomas

O trabalho apresentado na referência [6], diz respeito a um algoritmo de planeamento de seguimento de trajectória 3D óptimo para um veículo não tripulado, baseado no seguimento de uma trajectória numa estratégia Bang-Zero-Bang. Esse planeamento pode gerar uma sequência de trajectórias elementares 3D para um conjunto de *waypoints* pré-definidos no espaço. O algoritmo proposto usa uma sequência de cinco seguimentos de trajectória elementares para criar um modelo de uma trajectória tridimensional global no espaço.

Os sistemas aéreos não tripulados possuem uma autonomia decisiva, para tal torna-se necessário o desenvolvimento de métodos de controlo e de decisão com o objectivo de executar as várias operações durante uma missão, em particular quando as comunicações entre o operador e os veículos aéreos não tripulados são limitadas bem como no caso de missões de risco.

Um UAV deve ser capaz de seguir o plano de voo pré-definido mas também de criar um novo plano de voo em resposta a eventuais eventos ocorridos durante a missão, que possam invalidar o plano de voo actual.

O planeamento de uma trajectória é um problema de optimização que cria uma trajectória ideal entre duas configurações no espaço de estado, considerando um determinado índice de desempenho (tempo, energia ou a distancia). A sua viabilidade depende da escolha do método de optimização, do índice de desempenho e de uma serie de condicionantes de natureza variada. Este último depende essencialmente do próprio veículo (arquitectura, dinâmica e modos de actuação) e no ambiente em que o veículo se desloca (endurance, altitude, velocidade, modos de descolagem e de aterragem...).

Em geral, as técnicas de planeamento de trajectória podem ser classificadas em três grupos distintos. As técnicas baseadas na teoria de controlo óptimo, na teoria *flatness* [6] e as abordagens probabilísticas. A trajectória óptima pode ser encontrada através da minimização de um determinado índice de desempenho (tempo de execução, o consumo de energia, distancia percorrida...) e tendo em conta algumas restrições da dinâmica e da cinemática. O segundo grupo de técnicas de planeamento faculta uma propriedade estrutural de uma classe de sistemas dinâmicos, onde todos os estados e variáveis de entrada são expressos como funções de uma variável específica e das suas derivadas. Esta variável tem como nome, variável de saída plana. Esta propriedade

pode bem ser usada para simplificar a resolução de um problema de planeamento, quando a caracterização explícita da trajectória é necessária. Diante da dificuldade de generalizar a utilização de métodos analíticos, abordagens heurísticas foram desenvolvidas como as técnicas de mapa das estradas e do *Rapidly Exploring Random Tree*. Estas técnicas utilizam, em geral, um conjunto de primitivas utilizadas para construir os segmentos de todas as maneiras aleatórias.

A abordagem dos autores é parcialmente inspirada no trabalho de Dubin's, a ideia é usar as linhas rectas e circulares para construir a sequência ideal de trajectórias, verificando o Princípio Máximo de Pontryaguin (PMP) para trajectórias elementares óptimas. O seu plano de trajectória pode criar uma trajectória tridimensional no espaço, adicionando um nível constante para as asas nos voos de subida/descida e as hélices verticais para o seguimento inicial da trajectória elementar. Para reduzir a complexidade do problema, os autores utilizaram apenas o seguimento de trajectória como primitivas para criar uma trajectória global.

No caso da formação de UAVs, vários cenários podem ser considerados de acordo com as especificações da missão, o comprimento da trajectória deve ser igual ao segmento entre os *waypoints*, em oposição ao caso do voo livre, onde o objectivo da passagem pelo *waypoint* é para evitar um obstáculo ou uma área proibida. No trabalho realizado pelos autores, empenharam-se apenas pelo caso de tempo mínimo, ou seja, não é necessário que o veículo aéreo passa exactamente pelo ponto de passagem, mas que chegue perto da sua vizinhança. O seguimento de trajectória tem como vantagens, facilitar o planeamento e os problemas de controlo, a técnica de controlo linear pode ser suficiente para estabilizar o veículo na sua vizinhança em condições de seguimento. Outra vantagem é que os coeficientes aerodinâmicos que são variáveis no tempo e no espaço tornam-se estacionários sobre esta condição e a sua identificação torna-se mais fácil.

Com o algoritmo de planeamento de trajectória óptima para um veículo aéreo de asa fixa, baseado no seguimento de trajectória e na estratégia de Bang-Zero-Bang, os autores realizaram dois exemplos dos quais confirmaram que o seguimento de trajectória reduz significativamente a complexidade de um modelo dinâmico de um UAV bem como o problema de controlo de trajectória. O algoritmo usado continha cinco seguimentos de trajectória, o horizontal, a subida, nível das asas para voo de descida, o nível para rotação de voo e hélices verticais com curvatura e taxa de subida constantes. Os autores demonstraram que a estratégia Bang-Zero-Bang satisfaz a condição necessária de Pontryaguin. Os autores destacam uma das limitações deste algoritmo que diz respeito a velocidade do UAV que é mantida constante durante todas as trajectórias elementares.

1.3.2 Optimização e Controlo de Trajectrias 4D Definidas por Waypoints [3]

No trabalho descrito na referência [3], é proposto um método de navegação e controlo a quatro dimensões. O interesse da navegação e controlo de trajectórias 4D, deve-se ao facto de que uma certa aeronave, pode cumprir uma passagem por um determinado *waypoint* num tempo específico. 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. 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 distância. O autor pretendeu 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.

1.3.2.1 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,

$$(P_i, \tau_i) \rightarrow (P_i, [\tau_i^1, \tau_i^2]) \quad (1.1)$$

sendo $[\tau_i^1, \tau_i^2]$ o intervalo de tempo permitido para que se chegue a um determinado *waypoint*. Se $s(\tau)$ for a posição da aeronave, então o problema de navegação 4D está resolvido quando,

$$\|P_i - s(\tau_i)\| \leq \varepsilon \quad \forall i \in \mathfrak{R}^n \quad (1.2)$$

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

A inequação 1.2 pode ainda ser transformada numa igualdade,

$$\|P_i - s(\tau_i)\|^2 \leq \varepsilon^2 \quad \forall i \in \mathfrak{R}^n \quad (1.3)$$

1.3.2.2 Modelo Dinâmico

O modelo dinâmico apresenta duas possibilidades, uma na referência geocêntrica outra na referência geodética.

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

$$\dot{x} = V \cos(\gamma) \cos(\psi) \quad (1.4)$$

$$\dot{y} = V \cos(\gamma) \sin(\psi) \quad (1.5)$$

$$\dot{z} = V \gamma \quad (1.6)$$

$$\dot{V} = u_1 \quad (1.7)$$

$$\dot{\gamma} = u_2 \quad (1.8)$$

$$\dot{\psi} = u_3 \quad (1.9)$$

já na referência geodética basta transformar as equações 1.4, 1.5 e 1.6 para esta referência e o modelo de estado continua igual, assim,

$$\dot{\lambda} = \frac{V \cos(\gamma) \cos(\psi)}{(h + Re) \cos(\varphi)} \quad (1.10)$$

$$\dot{\varphi} = \frac{V \cos(\gamma) \sin(\psi)}{(h + Re)} \quad (1.11)$$

$$\dot{h} = V \sin(\gamma) \quad (1.12)$$

em que $\hat{x} = [x, y, z, V, \gamma, \psi]$ ou $\hat{x} = [\lambda, \varphi, h, V, \gamma, \psi]$ é 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 retrata no fundo a dinâmica da aeronave. O problema para obter uma trajectória que a 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,

$$u_i \in [u_i^{min}, u_i^{max}] \quad (1.13)$$

$$\dot{V} \in [V_i^{min}, V_i^{max}] \quad (1.14)$$

$$\dot{\gamma} \in [\gamma_i^{min}, \gamma_i^{max}] \quad (1.15)$$

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

1.3.2.3 Resolução do Problema de Navegação 4D

Depois do modelo da dinâmica da navegação, foi formulado o problema em termos de controlo óptimo para que depois seja possível aplicar um método de procura de trajectórias óptimas.

Seja $w_{pi} \rightarrow (P_i, \tau_i)$ uma sequência de *waypoints* pré-definidos, w_{p0} e w_{pf} o *waypoint* inicial e final, respectivamente, então o que se quer para uma determinada missão é minimizar o atraso no *waypoint* final w_{pf} . Se $s(\tau)$ for uma trajectória no tempo e espaço contínuo,

$$s(\tau) = \begin{pmatrix} \lambda(\tau) \\ \varphi(\tau) \\ h(\tau) \end{pmatrix} \quad (1.16)$$

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

$$J = \|P_f - s(\tau_f)\|_Q^2 \quad (1.17)$$

em que τ_f é o tempo pretendido para a duração da trajectória e Q é a 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,

$$x = \begin{bmatrix} \lambda \\ \phi \\ h \\ V \\ \gamma \\ \psi \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (1.18)$$

Assim sendo, o problema de optimização de trajectória aparece formulado da seguinte maneira:

$$\min_u J(u) = \|P_f - s(\tau_f)\|_Q^2 \quad (1.19)$$

sujeito a,

$$\dot{x} = f(x, u) \quad (1.20)$$

$$\|P_i - s(\tau_i)\| \leq \epsilon \quad (1.21)$$

$$u_i^{\min} \leq u_i \leq u_i^{\max} \quad (1.22)$$

$$V^{\min} \leq V \leq V^{\max} \quad (1.23)$$

$$\gamma^{\min} \leq \gamma \leq \gamma^{\max} \quad (1.24)$$

Com o problema de navegação descrito num problema de optimização de trajectória, aplica-se então um método de optimização de trajectórias. Existem dois tipos de métodos para resolver o problema de controlo óptimo, os métodos indirectos e os métodos directos. Os métodos indirectos assentam no princípio do mínimo de Pontryagin, a sua implementação para problemas complexos não de todo apropriado pelas razões que já foram explicadas. O autor optou 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 método utilizado para descrever as equações diferenciais de estado foi os métodos pseudoespectrais.

1.3.3 Orientação Segundo uma Trajectória Desejada Definida por *Waypoints* [4]

Como já foi evidenciado anteriormente, o desenvolvimento de orientação, navegação e sistemas de controlo de aeronaves tem sido uma área de forte exploração. Com a necessidade de resposta a sistemas mais confiáveis e robustos, tem existido uma criação de inúmeros métodos relacionados com o desempenho da aeronave. Actualmente, na aviação em geral, o conhecimento de uma trajectória de voo é especificada por *waypoints*, localizados no espaço inercial. Existem dois tipos de métodos comuns para a orientação de uma aeronave convencional, são eles o DTW (Direct-To-Waypoint) e o TTW (Track-To-Waypoint), em relação a trajectórias designadas com *waypoints*. O método DTW permite simplificar questões de comando de rumo do veículo, baseado na diferença angular entre o *waypoint* e o veículo. Quando o veículo atinge o *waypoint*, o sistema de controlo orienta a aeronave para o *waypoint* seguinte através de uma nova ordem. No segundo método, TTW, pretende orientar-se por trajectórias definidas entre *waypoints*. Neste método, a orientação do sistema de controlo visa a minimizar o deslocamento lateral entre a trajectória de voo prevista e a posição da aeronave. Portanto, ao contrário do método DTW, este adiciona uma restrição sobre uma trajectória de voo que este deve seguir para alcançar o *waypoint*. No entanto, ambos os métodos estão longe de serem considerados óptimos. Tal pode ser confirmado nas transições das aeronaves, na trajectória de voo, após atingir um *waypoint*. De modo a suavizar ou minimizar desvios da trajectória de voo, têm sido investigadas estratégias, através de métodos modernos de controlo, tais como *receding-horizon control* e controlo preditivo, para antecipar as mudanças na trajectória de voo e tomar as medidas de controlo antes de chegar ao objectivo, enquanto mantém um desempenho de voo adequado ao veículo.

A orientação de mísseis e sistemas de controlo operam com princípios semelhantes aos algoritmos de orientação e de controlo de aeronaves comerciais, civis, e não tripuladas. Num sistema de mísseis, a primeira função é interceptar o movimento de um alvo através da informação sobre a posição relativa e a velocidade entre o perseguidor e o destino. O PG (Pursuit Guidance)

foi um dos primeiros métodos utilizados para a orientação de mísseis. A teoria de PG baseia-se em manter um rumo directo em direcção ao alvo, dirigindo o ângulo entre o eixo frontal do prosseguidor e a LOS (line of sight) para o alvo. Isto é conseguido, ordenando ao veículo perseguidor que mantenha uma taxa igual à taxa da LOS. À medida que o alvo se move, o perseguidor também se move, mantendo uma linha directa com o alvo para proceder à sua intercepção.

Na sua forma mais simples, o algoritmo PG pode ser expresso matematicamente como

$$x_c = N\dot{\lambda} \quad (1.25)$$

onde x_c é o comando de saída, N é o ganho constante proporcional que multiplica pela taxa LOS, λ .

As soluções deste método não consideram a trajectória tomada ou os níveis de desempenho do sistema exigido pelo perseguidor para atingir o alvo, resultando numa solução longe da óptima. Para resolver este problema de sub-otimização, foram introduzidos parâmetros adicionais para melhorar o desempenho dos mísseis. Uma das melhorias introduzidas foi a emissão dos comandos da aceleração lateral com base no erro de trajectória e da taxa do erro de trajectória do alvo. Outro método tem como intuito modificar o nível de controlo do algoritmo de orientação possuído pelo veículo, ajustando o nível de ganho proporcional.

No trabalho apresentado em [4], é discutido o desenvolvimento de uma lei de orientação em fusão com os conceitos de alvo virtual com os de orientação em perseguição para uma aplicação num sistema de orientação de aeronaves. Foi desenvolvido, pois um algoritmo capaz de seguir uma trajectória de orientação de uma aeronave em perseguição utilizando uma nova designação de *waypoint*, denominado por *synthetic-waypoint*, que utiliza apenas um pequeno conjunto de parâmetros de orientação, além da extensão do conceito do alvo virtual para concluir a orientação da aeronave. O caminho é definido por uma trajectória entre um conjunto mínimo de *waypoints*, em determinados locais, removendo alguns que possam afectar a suavidade da trajectória ou a necessidade de uma alteração lógica da trajectória planeada quando um *waypoint* é atingido. O *synthetic-waypoint* percorre a trajectória entre os *waypoints*, com a aeronave a percorrer com suavidade a trajectória gerada pela sua própria dinâmica.

O controlo de voo da aeronave é direccionado por uma trajectória e pelos parâmetros calculados do ângulo de subida a partir da diferença entre o posicionamento do avião e o *synthetic-waypoint*, através da observação da dinâmica da LOS. A formulação do sistema de controlo é realizada utilizando o projecto de um regulador quadrático linear. As coordenadas (x_a, y_a, z_a) e (x_w, y_w, z_w) definem a posição actual da aeronave e do *synthetic-waypoint* respectivamente. O rumo e os ângulos de subida para a posição do *synthetic-waypoint* podem ser avaliados da seguinte maneira:

$$\begin{aligned}\varphi_c &= \arctan\left(\frac{R_y}{R_x}\right) \\ \gamma_c &= \arcsin\left(\frac{R_z}{R}\right)\end{aligned}\quad (1.26)$$

onde

$$\begin{aligned}R_x &= x_w - x_a \\ R_y &= y_w - y_a \\ R_z &= -z_w + z_a\end{aligned}\quad (1.27)$$

φ_c é variável do rumo e γ_c é a variável do ângulo de subida para o *synthetic-waypoint* desde a posição actual da aeronave. O bloco de orientação é executado dentro do loop no algoritmo de SWG (*Synthetic-Waypoint Guidance*), recebendo os estados da dinâmica da aeronave para gerar comandos de orientação que também estão em conta para o controlo e o sistema de navegação. O sistema de navegação contém a informação da trajectória de voo definida para utilizar a actualização da posição do *synthetic-waypoint*, que é então alimentado de volta para o bloco de orientação, juntamente com os estados da aeronave. O bloco de controlo recebe os comandos de orientação para gerar acções de controlo com base nos estados actuais da aeronave em relação ao *synthetic-waypoint*. Por fim são gerados os controlos usados para orientar a aeronave ao longo da trajectória desejada.

1.3.4 Síntese

Neste capítulo, foram apresentados três métodos de navegação autónoma para veículos aéreos não tripulados. O método apresentado em [6], identifica-se como um método que reduz significativamente a complexidade de um modelo dinâmico de um UAV bem como o problema de controlo. O método permite uma facilidade no planeamento, nos problemas de controlo e tem como mais valia a estagnação dos coeficientes aerodinâmicos que são variáveis no tempo e no espaço, tornando mais fácil a sua identificação. Por outro lado este método apresenta algumas limitações como a da velocidade do UAV que é mantida constante durante todas as trajectórias elementares, bem como todos os valores iniciais de voo que o projectista seria obrigado a saber, como por exemplo, as coordenadas de todos os *waypoints*, não sendo possível acrescentar mais durante o voo, de tal forma que torna o UAV limitado se o operador desejar alterar a sua rota durante a fase de voo. Em contrapartida o trabalho realizado em [3], permite obter uma optimização e controlo de trajectória 4D definidas por *waypoints*, utilizando o método pseudoespectral. Através deste constatou-se que uma determinada aeronave seria capaz de cumprir todas as restrições, como o minimizar o atraso em cada *waypoint*. Ficou demonstrado, pelo método de Medagoda e Gibbens [4], através de simulações não lineares que o desempenho de uma aeronave ao seguir uma trajectória específica é fortemente dependente de mudanças nos parâmetros de orientação, nomeadamente o tempo de horizonte. O algoritmo desenvolvido apresentou, nas simulações não lineares um desempenho dentro do alcance da aeronave para condições de vários tempos de horizonte e em confronto com perturbações atmosféricas. Com o aumento do tempo de horizonte prevê-se uma maior estabilidade na convergência da trajectória de voo em detrimento do tempo de resposta.

Em conclusão é de realçar que todos os métodos presenciados utilizam uma navegação baseada em *waypoints*, sendo que apenas um deles permite uma navegação 4D.

1.4 Objectivo da Dissertação

O avanço da tecnologia tem tornado mais fácil encontrar sensores e sistemas de controlo que realizam o controlo de forma eficaz e de custos reduzidos. Através do desenvolvimento de algoritmos eficazes podemos tomar partido, de forma mais eficiente, das aptidões dos UAVs.

O objectivo deste trabalho é projectar um método de navegação aérea para veículos aéreos não tripulados em *loiters* normalizados pela OTAN, utilizando um método de controlo capaz de funcionar através de trajectórias contínuas, que permita uma navegação eficiente, autónoma e com um tempo de processamento reduzido. Um dos aspectos importante para o uso de uma trajectória contínua deve-se ao facto de que os controladores que utilizam uma navegação baseada em *waypoints* não permitem um seguimento adequado da trajectória de referência e pode, por exemplo, influenciar uma missão de monitoramento de um UAV numa aérea através da captação de imagens. O desenvolvimento do algoritmo de trajectória, *loiters*, tem em conta todos os parâmetros e restrições impostas pela OTAN na STANAG 4586.

Capítulo 2

Método de Navegação Autónoma em Loiters

O progresso na tecnologia de orientação de sistemas de veículos aéreos não tripulados, baseados no sistema de posicionamento global, tem originado o desenvolvimento de vários métodos para o projecto de controladores não lineares.

Neste capítulo será apresentado um método para a orientação de navegação de veículos aéreos não tripulados, em *loiters*. No capítulo anterior, foi abordado um estudo para melhor compreender e explorar conceitos e teorias importantes para a resolução deste método de navegação. A vantagem deste trabalho deve-se ao facto de se utilizar um método de controlo não linear de controlo feedback baseado nas derivadas de Lie de Carloni-Bousson [7], utilizando uma trajectória de referência contínua.

2.1 Controlo Não Linear

Entende-se por controlo não linear o conjunto de procedimentos destinados a fazer com que as variáveis de saída de um sistema não linear se aproximem de uma determinada referência e estabilizem numa vizinhança do seu valor.

Num controlo não linear temos que

$$\dot{x} = f(x) + G(x)u \quad (2.1)$$

no entanto, esta equação pode ser assumida da seguinte forma, no qual o vector de estado é dividido em dois subvectores x_1 e x_2 :

$$\begin{cases} \dot{x}_1 = f_1(x) \\ \dot{x}_2 = f_2(x) + B_2 u \end{cases} \quad (2.2)$$

com

$$x = [x_1, x_2]^T \quad (2.3)$$

e consequentemente:

$$G = \begin{pmatrix} 0 \\ B_2 \end{pmatrix} \quad (2.4)$$

B_2 e G são matrizes de dimensões adequadas. O primeiro vector de estado x_1 na equação 2.2 não está directamente dependente do vector de controlo. O segundo vector de estado x_2 , ao contrário, é composto por esse conjunto de equações diferenciais que envolve o vector de controlo.

Assumindo a trajetória de referência como uma contínua função no tempo

$$q(t) = [q_1(t) \ q_2(t)]^T \quad (2.5)$$

é possível definir o sistema dinâmico como

$$\begin{cases} \dot{q}_1 = f_1(q) \\ \dot{q}_2 = f_2(q) + B_2(q) u_{ref} \end{cases} \quad (2.6)$$

onde u_{ref} é o controlo de referência. O erro entre o actual estado do sistema e a referência é:

$$e(t) = x(t) - q(t) \quad (2.7)$$

A função de custo utiliza um valor constante para o passo-tempo (h) (que pode ser visto como o ganho do controlador)

$$J(u(t)) = \frac{1}{2}e^T(t)Qe(t) + \frac{1}{2}u^T(t)Ru(t) \quad (2.8)$$

onde $Q \in \mathfrak{R}^{n \times n}$ e $R \in \mathfrak{R}^{m \times m}$ são semi-definidas positivamente pelo projectista. Este índice de desempenho expressa o erro de trajectória e o esforço de controlo. Q e R serão definidas como:

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & hQ_2 \end{bmatrix} \quad R = \begin{bmatrix} 0 & 0 \\ 0 & R_2 \end{bmatrix} \quad (2.9)$$

O controlo ideal preditivo para minimizar o índice de desempenho é encontrado através da resolução da equação $\partial J / \partial u = 0$ para o controlo u . Em consequência

$$u(t) = - \{ [\Lambda(h)W(x)]^t Q \Lambda(h) + R \}^{-1} \{ [\Lambda(h)W(x)]^t Q [e(t) + \dot{e}(t)] \} \quad (2.10)$$

onde $\Lambda \in \mathfrak{R}^{n \times n}$ é uma matriz diagonal (n é a dimensão do espaço de estado) em que os elementos da diagonal Λ_{ii} são dados por:

$$\Lambda_{ii}(h) = \frac{h^{r_i}}{r_i!}, \quad i = 1, \dots, n \quad (2.11)$$

e $W \in \mathfrak{R}^{n \times m}$ é uma matriz i linhas definida como:

$$W_i = \left\{ L_{g_j} [L_f^{r_i-1}(x_i)], \dots, L_{g_m} [L_f^{r_i-1}(x_i)] \right\}, \quad i = 1, \dots, n \quad (2.12)$$

onde os vectores g_1, \dots, g_m representam as colunas da matriz G .

2.2 Modelo Dinâmico de Navegação

O modelo dinâmico usado para desenvolvimento da navegação em *loiter* é um modelo semi-linearizado para um pequeno ângulo de trajectória.

$$\dot{x} = V \cos(\psi) \quad (2.13)$$

$$\dot{y} = V \sin(\psi) \quad (2.14)$$

$$\dot{z} = V\gamma \quad (2.15)$$

$$\dot{\psi} = \frac{g}{V} \tan \sigma \quad (2.16)$$

$$\dot{\gamma} = f(\delta) \quad (2.17)$$

onde δ é a deflexão das superfícies de controlo verticais, V é a velocidade da aeronave, ψ é o ângulo de rumo, g é a aceleração gravítica e σ é o ângulo de rolamento.

Os controlos serão definidos como:

$$u_1(t) = \tan(\sigma) \quad (2.18)$$

$$u_2(t) = f(\delta) \quad (2.19)$$

O modelo encontra-se sujeito a várias restrições:

- Uma restrição aplicada ao estado, em especial ao ângulo de trajectória

$$|\gamma| \leq 5^\circ \quad (2.20)$$

- Duas outras restrições aplicadas são

$$|\sigma| \leq 80^\circ \quad (2.21)$$

$$|\dot{\gamma}| \equiv |u_2| \leq 30^\circ/s$$

O modelo será dividido num estado x_1 , composto por três elementos e um estado x_2 , composto por dois elementos, isto é:

$$x_1 = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}, x_2 = \begin{Bmatrix} \Psi \\ \gamma \end{Bmatrix} \quad (2.22)$$

2.3 Loiters

O *loiter*, como já referenciado anteriormente, consiste num voo de cruzeiro para um determinado período de tempo sobre uma área específica. Nesta secção proponho um modo de caracterizar cada trajectória contínua dos *loiters* definidos pela OTAN, na normalização STANAG 4586 [1], começando pelo *loiter* circular, de seguida, o *loiter* “Figura-8” e por fim o *loiter* hipódromo.

2.3.1 Loiter Circular

A trajectória do círculo é descrita pelo seguinte sistema contínuo:

$$\bar{x}(t) = R \cos(\omega t) \quad (2.23)$$

$$\bar{y}(t) = R \sin(\omega t) \quad (2.24)$$

$$\bar{z}(t) = \text{Constante} \quad (2.25)$$

$$\bar{\Psi}(t) = \frac{\pi}{2} + \omega t \quad (2.26)$$

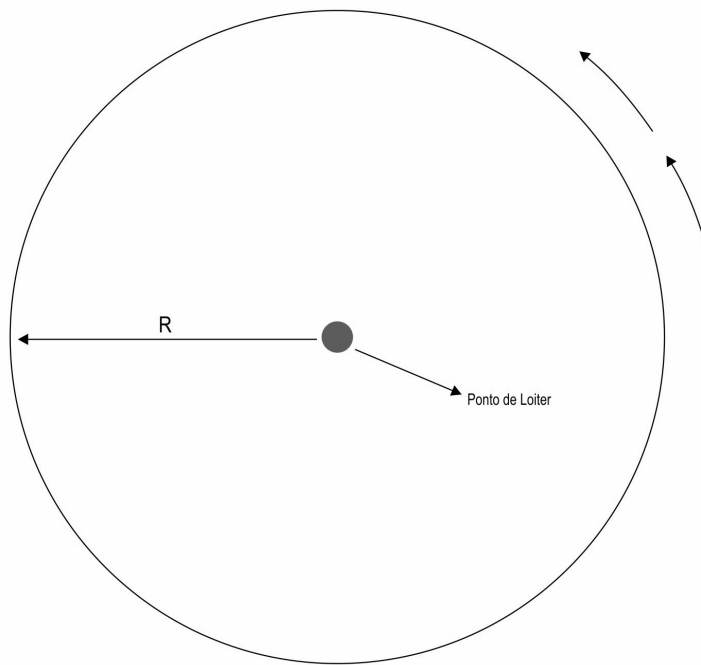


FIGURA 2.1: Loiter Circular

2.3.2 Loiter “Figura-8”

Utilizando as equações parametrizadas do círculo e da recta, descrevemos a trajectória contínua na forma de um oito invertido. Pretendendo tornar mais simples a resolução da parametrização do *loiter*, delimitamos este por tempos. Através destes limites obtivemos várias fases, como podemos observar na figura abaixo.

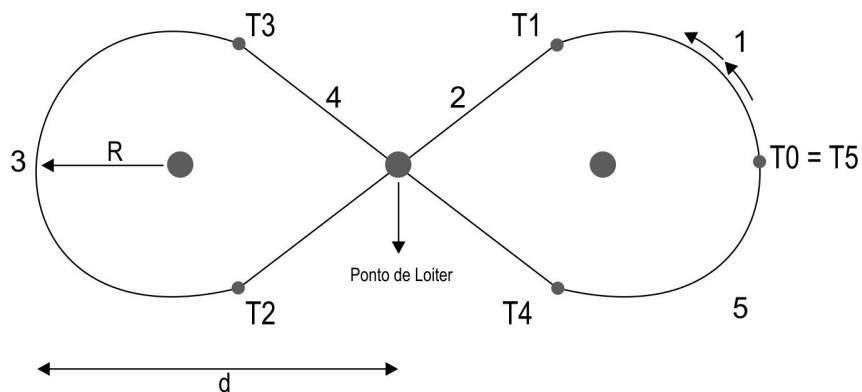


FIGURA 2.2: Loiter Forma-8

Fase 1

A Fase 1, corresponde a fase delimitada pelos tempos T0 e T1.

$$\bar{x}(t) = \frac{d}{2} + R\cos(\omega t) \quad (2.27)$$

$$\bar{y}(t) = R\sin(\omega t) \quad (2.28)$$

$$\bar{z}(t) = Constante \quad (2.29)$$

$$\bar{\psi}(t) = \frac{\pi}{2} + \omega t \quad (2.30)$$

Fase 2

De T1 a T2, passando obrigatoriamente pelo Ponto de *Loiter*, designa-se pela Fase 2.

$$\bar{x}(t) = \frac{d}{2} - V_x t \quad (2.31)$$

$$\bar{y}(t) = R - V_y t \quad (2.32)$$

$$\bar{z}(t) = Constante \quad (2.33)$$

$$\bar{\psi}(t) = \frac{\pi}{2} + \omega t \quad (2.34)$$

Fase 3

À Fase 3, corresponde a fase estremada pelos tempos T2 e T3.

$$\bar{x}(t) = -\frac{d}{2} + R\cos(\omega t) \quad (2.35)$$

$$\bar{y}(t) = -R\sin(\omega t) \quad (2.36)$$

$$\bar{z}(t) = \text{Constante} \quad (2.37)$$

$$\bar{\Psi}(t) = \frac{\pi}{2} + \omega t \quad (2.38)$$

Fase 4

À Fase 4, pertence o intervalo de T3 e T4, passando imperiosamente pelo Ponto de *Loiter*.

$$\bar{x}(t) = -\frac{d}{2} + V_x t \quad (2.39)$$

$$\bar{y}(t) = R - V_y t \quad (2.40)$$

$$\bar{z}(t) = \text{Constante} \quad (2.41)$$

$$\bar{\Psi}(t) = \frac{\pi}{2} + \omega t \quad (2.42)$$

Fase 5

Por fim, à Fase 5, equivale a fase delimitada pelos tempos T4 e T5.

$$\bar{x}(t) = \frac{d}{2} + R\cos(\omega t) \quad (2.43)$$

$$\bar{y}(t) = R\sin(\omega t) \quad (2.44)$$

$$\bar{z}(t) = \text{Constante} \quad (2.45)$$

$$\bar{\Psi}(t) = \frac{\pi}{2} + \omega t \quad (2.46)$$

2.3.3 Loiter Hipódromo

Do mesmo modo que parametrizamos o *loiter* “Figura-8”, iremos parametrizar o *loiter* Hipódromo, utilizando as equações parametrizadas do círculo e da recta. Como tal, este foi da mesma forma dividido em secções/fases para facilitar a sua parametrização.

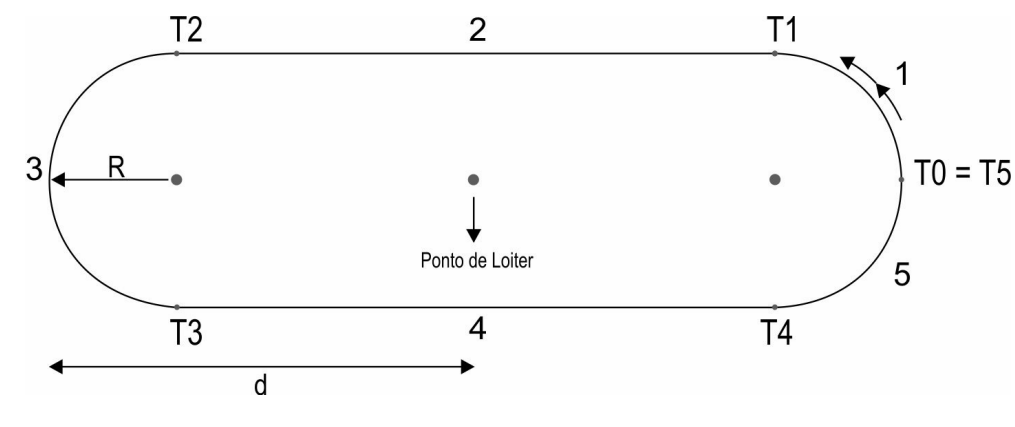


FIGURA 2.3: Loiter Forma-8

Fase 1

A Fase 1, corresponde a fase delimitada pelos tempos T0 e T1.

$$\bar{x}(t) = \frac{d}{2} + R\cos(\omega t) \quad (2.47)$$

$$\bar{y}(t) = R\sin(\omega t) \quad (2.48)$$

$$\bar{z}(t) = Constante \quad (2.49)$$

$$\bar{\Psi}(t) = \frac{\pi}{2} + \omega t \quad (2.50)$$

Fase 2

De T1 a T2 designa-se pela Fase 2.

$$\bar{x}(t) = \frac{d}{2} - V_x t \quad (2.51)$$

$$\bar{y}(t) = R \quad (2.52)$$

$$\bar{z}(t) = \text{Constante} \quad (2.53)$$

$$\bar{\Psi}(t) = \frac{\pi}{2} + \omega t \quad (2.54)$$

Fase 3

À Fase 3, corresponde a fase estremada pelos tempos T2 e T3.

$$\bar{x}(t) = -\frac{d}{2} + R \cos(\omega t) \quad (2.55)$$

$$\bar{y}(t) = R \sin(\omega t) \quad (2.56)$$

$$\bar{z}(t) = \text{Constante} \quad (2.57)$$

$$\bar{\Psi}(t) = \frac{\pi}{2} + \omega t \quad (2.58)$$

Fase 4

À Fase 4, pertence o intervalo de T3 e T4.

$$\bar{x}(t) = -\frac{d}{2} + V_x t \quad (2.59)$$

$$\bar{y}(t) = -R \quad (2.60)$$

$$\bar{z}(t) = Constante \quad (2.61)$$

$$\bar{\Psi}(t) = \frac{\pi}{2} + \omega t \quad (2.62)$$

Fase 5

Por fim, à Fase 5, equivale a fase delimitada pelos tempos T4 e T5.

$$\bar{x}(t) = \frac{d}{2} + R\cos(\omega t) \quad (2.63)$$

$$\bar{y}(t) = R\sin(\omega t) \quad (2.64)$$

$$\bar{z}(t) = Constante \quad (2.65)$$

$$\bar{\Psi}(t) = \frac{\pi}{2} + \omega t \quad (2.66)$$

onde R é o raio do *loiter*, ω é a velocidade angular, z é uma constante altitude, d é a distância desde o ponto de *loiter* até a extremidade da secção circular e $V = V_x + V_y$.

2.4 Algoritmo

O bloco de orientação tem como incumbência proporcionar à aeronave as direcções e velocidade a manter durante a missão. Este recebe o vector posição corrente, o vector posição de referência, a velocidade de referência e o tempo restante e fornece ao bloco de controlo, sob a forma de comandos, a velocidade a adoptar e a atitude que, não poderão exceder os limites previamente estabelecidos. O seguinte algoritmo resolve o problema de navegação aérea autónoma nos três diferentes *loiters*.

1. Descrever os dados iniciais para a realização da simulação (velocidade de *loiter* da aeronave, altitude de *loiter*, raio de *loiter*, aceleração gravítica e tempo de missão);
2. Impor os parâmetros h , Q e R necessário para o controlo da trajectória;
3. Definir a posição actual da aeronave;
4. Calcular os ângulos de referência, γ e ψ , de modo a orientar a aeronave em direcção à trajectória de referência descrita parametricamente;
5. Enviar a velocidade de referência, rumo de referência e ângulo de trajectória para o módulo de controlo.

Capítulo 3

Simulação e Resultados

Neste capítulo apresenta-se as simulações referentes a cada um dos *loiters* descritos na normalização da OTAN, STANAG 4586, de forma a obter o estudo completo da navegação autónoma em *loiter*, utilizando o método descrito no capítulo anterior. O ambiente utilizado para validar o método de navegação autónoma em *loiters* decorre num software de cálculo numérico, *Matlab*TM. De forma a obter uma trajectória contínua que defina a missão e crie uma trajectória que faça a aeronave passar o mais próximo possível da referência, sem provocar a saturação das variáveis de estado e de controlo, no tempo previamente estipulado.

Descrevemos, de seguida, os parâmetros necessários para a simulação do algoritmo de navegação:

O ponto de *loiter* é dado por:

$$x(0) = \begin{Bmatrix} 0 \\ 0 \\ z \end{Bmatrix}$$

TABELA 3.1: Valores iniciais

Parâmetros	Nomenclatura	Unidades	Valor
Raio de loiter	R	m	500
Comprimento de loiter	d	m	1000
Altitude de loiter	z	m	500
Velocidade de loiter	V	m/s	20

Seja

$$Q_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\varepsilon + e_z^2} \end{bmatrix} \quad Q_2 = \begin{bmatrix} h^2 & 0 \\ 0 & h^2 \end{bmatrix} \quad R_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

onde $\varepsilon = 10^{-6}$ é um parâmetro para evitar que a matriz Q_1 seja singular, h é o passo-tempo e

$$e_z = x_z - q_z$$

3.1 Simulação do Loiter Circular

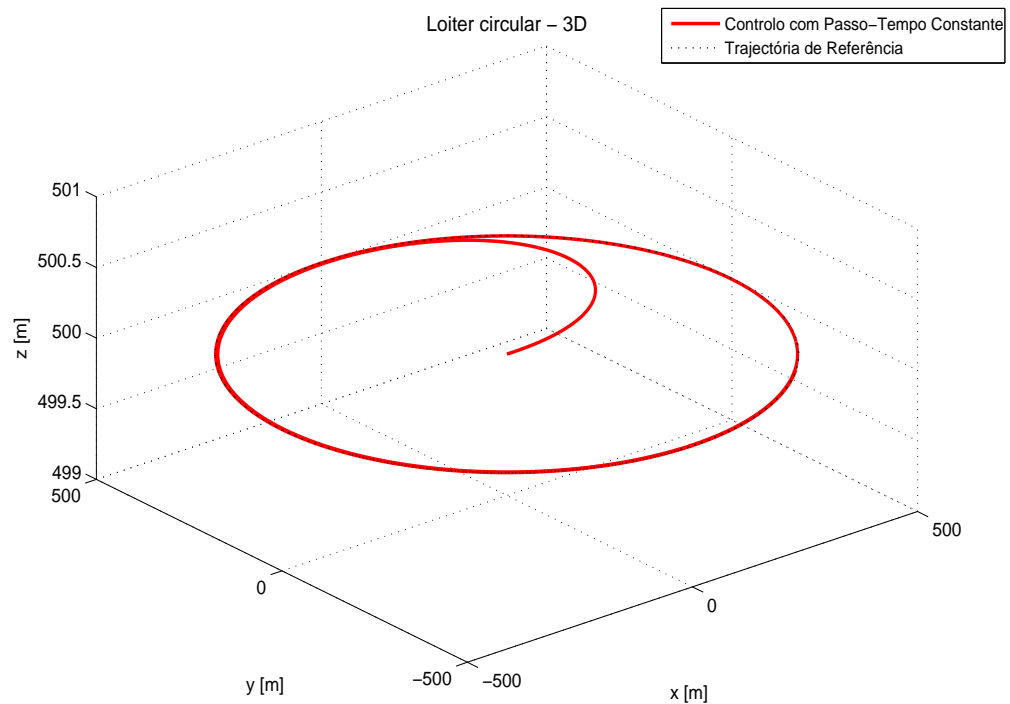


FIGURA 3.1: Loiter circular - 3D

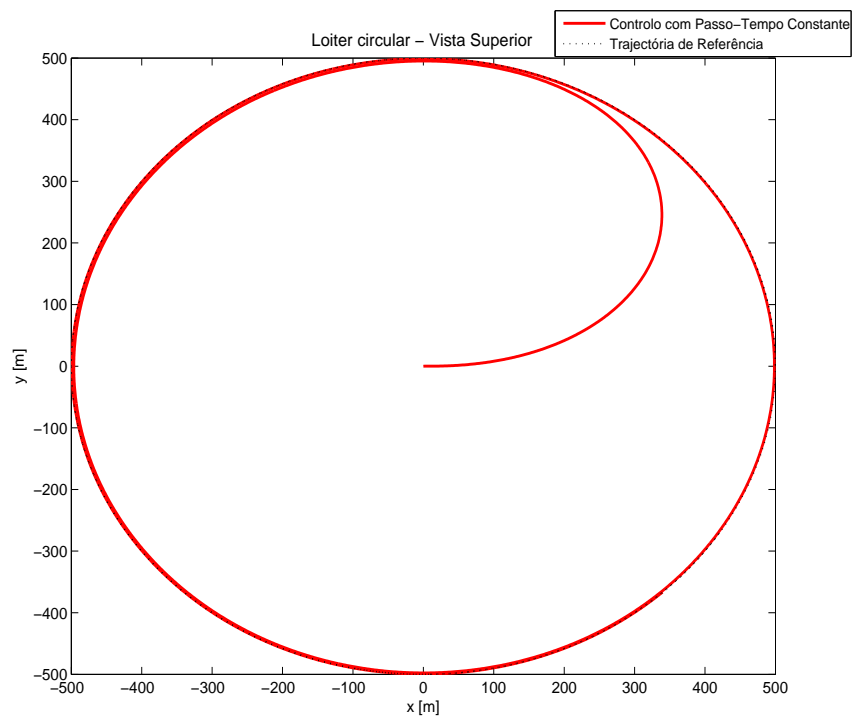


FIGURA 3.2: Loiter circular - Vista superior

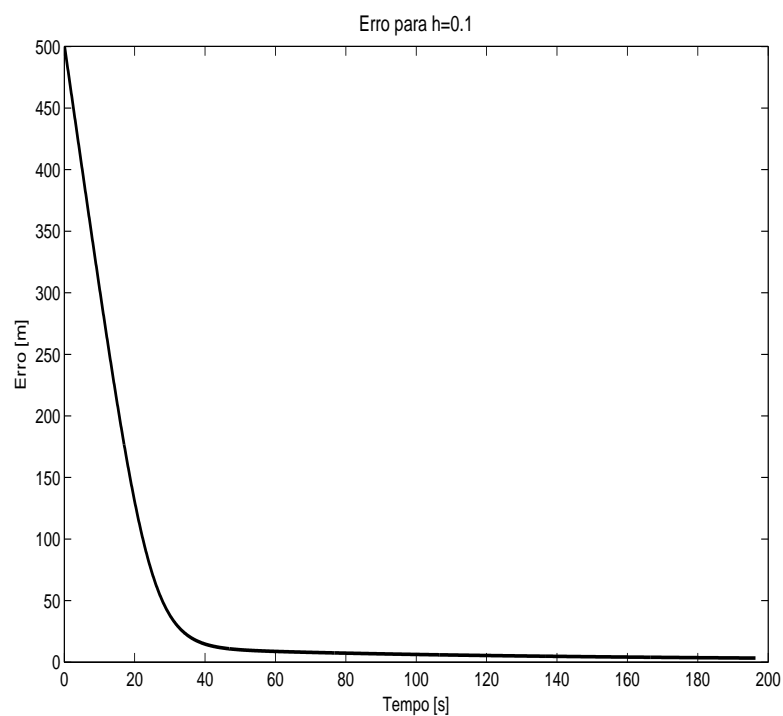


FIGURA 3.3: Loiter circular - Erro experimental

Ao analisar as figuras 3.1 e 3.2 é visível a resposta do sistema para um $h = 0.1s$. A direcção do *loiter*, que define a direcção da volta quando se circula em torno do ponto de *loiter* definido pela posição da aeronave, é feita no sentido contrário ao sentido dos ponteiros do relógio. Com um tempo relativo a uma volta completa, verificamos que a trajectória do controlador aproxima-se da trajectória de referência circular muito rapidamente. Apuramos ainda que o controlo utilizado, bem como a forma de definir uma trajectória contínua, resultaram num controlo adequado ao longo da trajectória de referência, fazendo com que a trajectória controlada coincida exactamente com a trajectória de referência. Os ângulos de trajectória e de rumo são cumpridos pelo controlador, bem como a velocidade constante, acatando sempre com os constrangimentos. A Figura 3.3 mostra o erro experimental entre a trajectória de referência e a trajectória do controlador. O erro é maior no início, porque o controlador tem de deixar a sua posição inicial, ponto de *loiter*, para encontrar a trajectória de referência. Apesar do seu erro inicial, a trajectória chega à referência facilmente. Por fim, verificamos ainda que os resultados da simulação são claramente afectados pelo parâmetro do passo-tempo (h). De forma a melhorar os resultados, adicionamos um novo parâmetro c , mantendo todos os outros constantes. c é um parâmetro positivo e ingressa no sistema multiplicando este pela matriz Q de forma a obtermos um trajectória mais suave e com um erro menor. Para a simulação deste *loiter* foi utilizado um $c = 0.05$.

3.2 Simulação do Loiter “Figura-8”

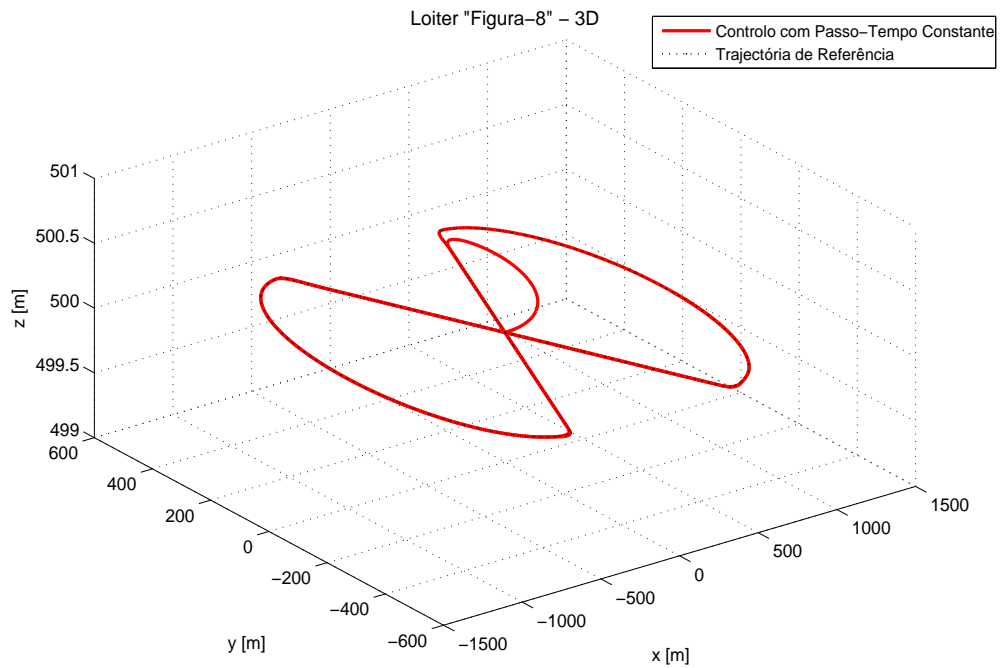


FIGURA 3.4: Loiter “Figura-8” - 3D

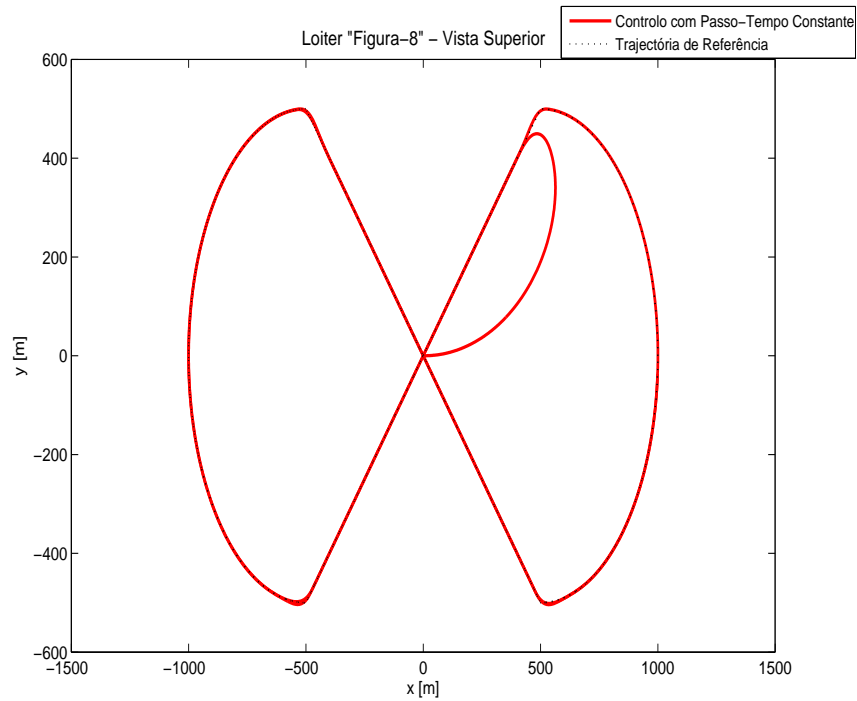


FIGURA 3.5: Loiter “Figura-8” - Vista superior

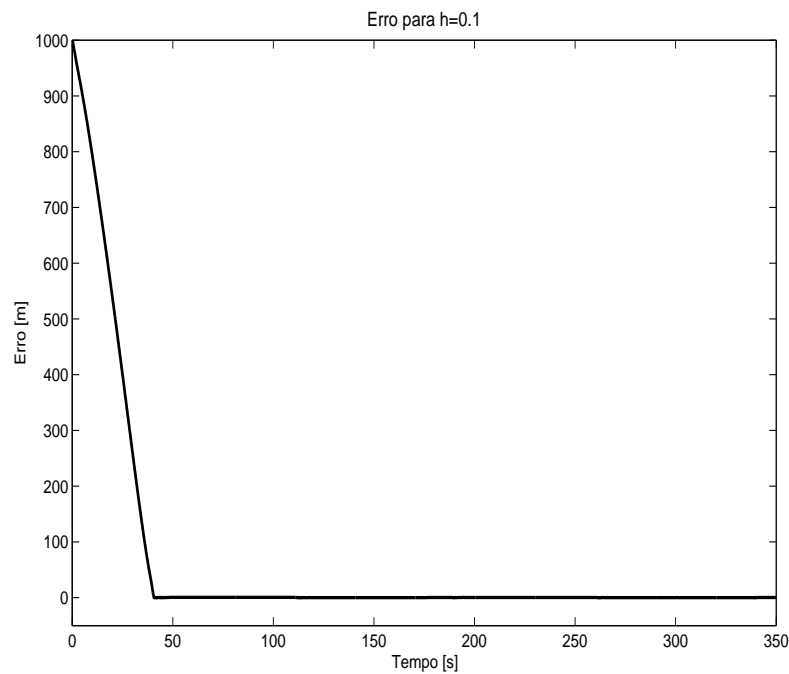


FIGURA 3.6: Loiter “Figura-8” - Erro experimental

Tal como sucede no *loiter* circular, a direcção do *loiter* “Figura-8” é feita no sentido contrário ao sentido dos ponteiros do relógio. Verificamos também para o tempo relativo a uma volta completa, Figuras 3.4 e 3.5, que a trajectória realizada pelo controlador aproxima-se da trajectória de referência muito rapidamente, utilizando um $h = 0.1s$ e um parâmetro $c = 0.5$. Mais uma vez o método de controlo e a forma de descrever a trajectória contínua resultou numa trajectória suave e coincidente com a trajectória de referência, originando um pequeno erro após a chegada da trajectória do controlador à trajectória de referência (Figura 3.6). Existe ainda um pequeno erro pouco significativo, mas que é resultado da transição de uma passagem de fases descrita na parametrização do *loiter* “Figura-8”.

3.3 Simulação do Loiter Hipódromo

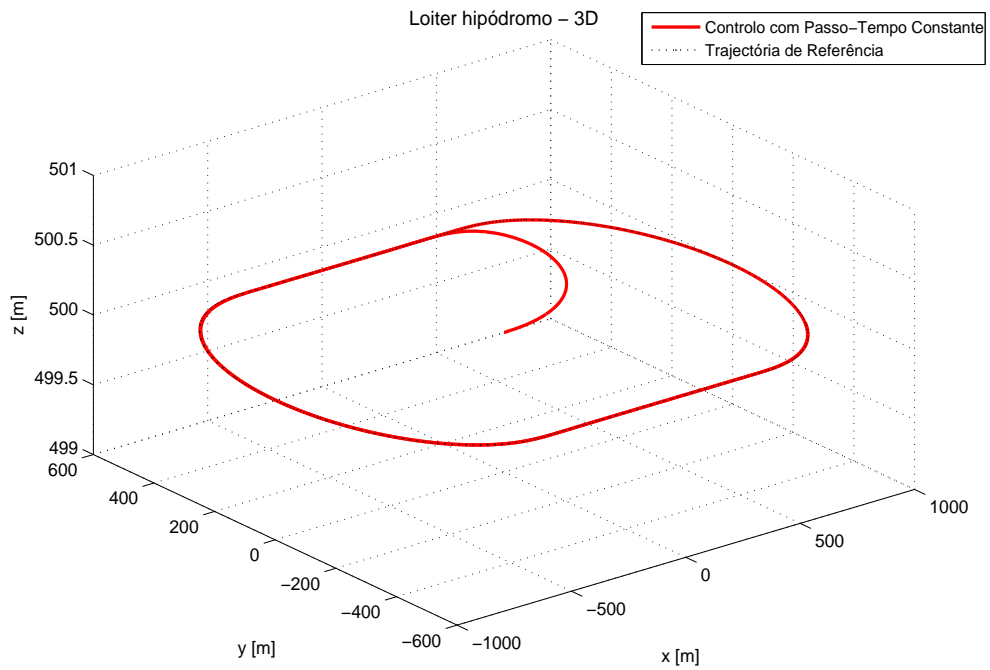


FIGURA 3.7: Loiter hipódromo - 3D

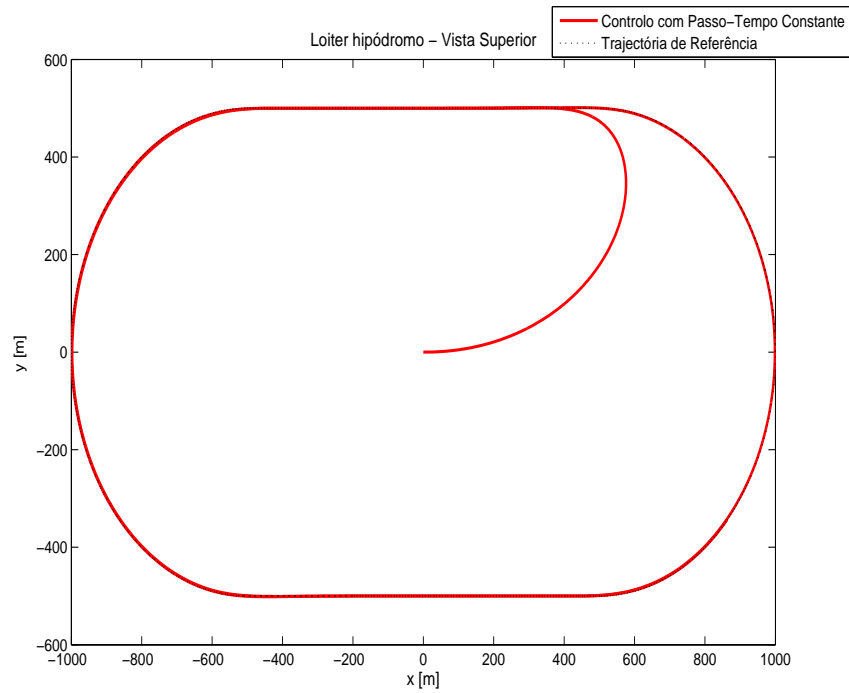


FIGURA 3.8: Loiter hipódromo - Vista superior

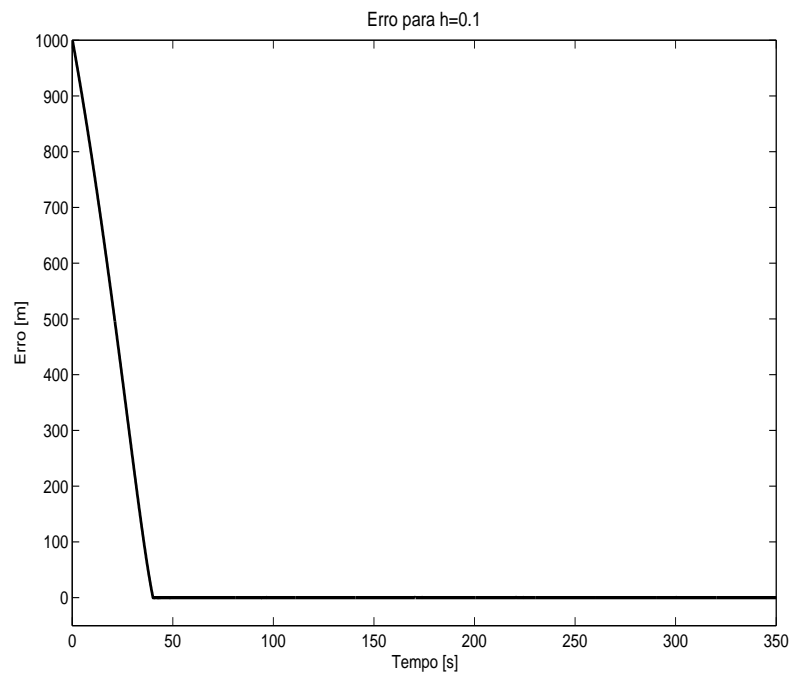


FIGURA 3.9: Loiter hipódromo - Erro experimental

Por fim, tal como aconteceu nos *loiter* circular e *loiter* “Figura-8”, a resposta do sistema foi bem sucedida, utilizando um $h = 0.1s$ e um parâmetro $c = 0.1$ (Figuras 3.7 e 3.8). A trajetória do controlador aproxima-se da trajetória de referência muito rapidamente chegando depois a coincidir praticamente com a referência. O erro experimental para uma volta completa é mostrado na Figura 3.9 e comprova o que foi dito anteriormente. O erro revela-se mais elevado no início, até que a trajetória do controlador intersecte a trajetória de referência. Tal com ocorre no *loiter* “Figura-8”, o *loiter* em forma de hipódromo também possui um pequeno erro resultado da mudança de fase da sua parametrização. Apesar do seu erro inicial, a trajetória chega à referência facilmente. O sentido da simulação do *loiter* em forma de hipódromo é realizado, como nos dois outros *loiters* apresentados, no sentido contrário ao sentido dos ponteiros do relógio.

Capítulo 4

Conclusões

Neste capítulo são apresentadas as conclusões do trabalho desenvolvido ao longo da dissertação, o que se aprendeu e passos para futuro desenvolvimento.

Este trabalho teve como objectivo a navegação autónoma aérea em *loiters* definidos pela STANAG 4586. Para cumprir tal objectivo, no desenvolvimento deste trabalho destacaram-se duas principais áreas, o planeamento da trajectória e o controlo necessário para o seu seguimento. Para tal foi necessário recorrer a um estudo sobre trabalhos executados nas várias vertentes desta temática, desde o planeamento das trajectórias aos vários métodos de controlo aplicados.

No capítulo 2, é apresentado o modelo de planeamento para a orientação e seguimento de uma trajectória de referência contínua 4D, como é o dos *loiters*. O planeamento do *loiter* circular foi, de certo modo, o mais simples de implementar, visto que apenas recorremos as equações parametrizadas do círculo para descrever a trajectória contínua. De seguida, sucederam-se os *loiters* “Figura-8” e Hipódromo, podendo resumir o seu planeamento da mesma maneira, visto terem sido adoptados os mesmos métodos para a sua realização. Para tal, a maneira mais simples e mais precisa encontrada foi utilizando as equações parametrizadas da recta e do círculo, dividindo os *loiters* em fases correspondentes a cada geometria. Por conseguinte, conseguimos descrever todos os *loiters* da STANAG 4586 de uma forma clara e concisa.

Determinado o método de planeamento de trajectória de referência, utilizou-se um método de controlo não linear para o controlo necessário para o seu seguimento. O método escolhido foi um método baseado nas derivadas de Lie proposto por Carloni e Bousson [7]. Verificamos que este método se adequa bastante a uma qualquer trajectória de referência contínua, como é o caso dos *loiters*. Realizamos as simulações correspondentes a cada *loiter* e averiguamos ao longo destas

que certos parâmetros, como h , Q , R , utilizados no projecto do controlador afectavam realmente o desempenho do controlador. Ficou demonstrado, depois de ter encontrado os parâmetros exactos para que o controlador cumprisse com o seguimento da trajectória, que este método de planeamento e controlo de seguimento de trajectória 4D teve capacidade de controlar a aeronave ao longo da trajectória pré-definida e com bastante precisão.

Como trabalho futuro propomos o desenvolvimento de um bloco que receba a velocidade bem como a atitude, sob a forma de comandos, para que a aeronave adopte, em cada instante, a condição de voo desejada, através do accionamento dos actuadores. Assim, todo este conjunto, poder-se-ia agrupar a uma plataforma que utilize o protocolo STANAG 4586 a fim de permitir a interoperabilidade entre vários sistemas de UAVs.

Bibliografia

- [1] NATO STANAG 4586 Ed. 2.5. Standard Interfaces for UAV Control Systems (UCS) for NATO UAV Interoperability. NATO Standardization Agency, Washington DC, 2007.
- [2] Bousson K. Geodetic Waypoint Navigation Guidance. *Department of Aerospace Sciences, University of Beira Interior, Covilhã, Portugal*, 2008.
- [3] Bousson K. and Machado P. 4D Flight Trajectory Optimization Based on Pseudospectral Methods. *International Conference on Aeronautical and Astronautical*, Amsterdam, September 2010.
- [4] Medagoda E. and Gibbens P. Synthetic-Waypoint Guidance Algorithm for Following a Desired Flight Trajectory. *Journal of Guidance, Control, and Dynamics*, 33(2), March–April 2010.
- [5] NATO STANAG 7085 Ed. 3. Interoperable Data Links for Intelligence, Surveillance, and Reconnaissance (ISR) Systems. NATO Standardization Agency, Washington DC, 2009.
- [6] Boukraa D., Bestaoui Y., and Azouz N. Three Dimensional Trajectory Generation for an Autonomous Plane. *International Review of Aerospace Engineering (I.RE.AS.E)*, 1(8):355–365, August 2008.
- [7] Carloni G. and Bousson K. A Nonlinear Control Method for Autonomous Navigation Guidance. *International Review of Aerospace Engineering (I.RE.AS.E)*, 1(8):378–389, August 2008.
- [8] Rugarn J. Rapid Development of Realistic UAV Simulations. *Department of Computer and Information Science, Linköpings University*, 1, February 2009.
- [9] Lu P. Nonlinear Predictive Controllers for Continuous Systems. *Journal of Guidance, Control, and Dynamics*, 17(3), 1994.

Anexos