

Filipe Varela

# Vigilância Distribuída com Navegação Autónoma de Aeronaves não Tripuladas



Universidade Lusófona de Humanidades e Tecnologias  
2008

Filipe Varela

# Vigilância Distribuída com Navegação Autónoma de Aeronaves não Tripuladas



*Relatório de Trabalho Final*

Universidade Lusófona de Humanidades e Tecnologias

2008

# Abstract

Recent developments in embedded technology and industrial production of micro electromechanical systems gave birth to an enormous variety of industrial grade single board computing platforms and solid state sensors. The availability of such products costing a fraction of previous existing items triggered the feasibility of many autonomous robotic systems which were exclusively attained only by highly funded projects each developed by an enormous amount of highly trained individuals.

Over the last few years, and as consequence of recent global conflicts and monitoring needs, we have witnessed an increasing interest in such autonomous robotic systems. One of such systems is the AMAV (autonomous micro aerial vehicle) which is the focus of this project.

The following sections illustrate the concept, design and implementation of one such system. Its main focus is that of showing that the use of low cost off the shelf components is fit for the design of a working prototype which may be used as a basis for several dedicated byproducts. This prototype is effectively a sandbox for future projects.

The prototype itself is a product of the integration of several hardware and software components, each performing an exclusive task and reporting its activity to a centralised monitoring module. These components may be enumerated as an array of sensor modules, a sensor fusion module, an array of actuator elements, an actuator controller module, an autonomous navigation module, a microware video transmitter and a microware radio transceiver modem for telemetry and control of the platform. The interactions of all modules will be illustrated on a block diagram and each of the individual modules will be described in detail.

The primary working objective is the full autonomous navigation of a sequence of geo-referenced waypoints or, alternatively, semi autonomous navigation in which case, only some of the control surfaces are under computer authority and the others may be remotely controlled by a groundstation unit which, in turn, has two different approaches to control the platform. One is to set the desired states - heading, altitude and airspeed - and the other is real time relay of data produced by a human interface device (HID) such as a joystick or gamepad. Additionally, the groundstation may instruct the platform to act solely upon commands from a conventional radio control transmitter in use by a human.

All of these operational modes may be manually toggled in real time by the human in control of the system. This allows for several interesting functions such as ATOL (automated take off and landing) and MTOL (manual take off and landing).

The groundstation itself, while not a part of the specification in this project, is a fully functional Mac OS X application which displays a traditional HUD (heads up display) inside an OpenGL context. This context is overlaid on top of a real time video feed from the platform, effectively producing an augmented reality context with several points of interest such as real world object positions, ie: airports, nav aids, vor, dme, air routes for air traffic avoidance, and several others.

Because the hardware itself is also not part of the specification, a simulator integration module has been added to the project. This module acts as a compile time toggled sensor and actuator module. The platform determines whether it is sensing virtual sensors and acts upon that result to transmit simulator control packets to the flight simulator itself. At the time of this writing only one such simulator mode was written. That was the X-Plane module. As a sidenote, X-Plane is a fully FAA certified flight instruction system and is used by many aircraft manufacturers to test performance of aircraft before actual production.

# Conteúdo

<b>Abstract</b>	<b>3</b>
<b>Índice de Figuras</b>	<b>8</b>
<b>Índice de Equações</b>	<b>9</b>
<b>1 Introdução</b>	<b>10</b>
<b>2 Enquadramento Teórico</b>	<b>11</b>
2.1 Sensores . . . . .	12
2.1.1 Sensores Inerciais . . . . .	12
2.1.1.1 Complementaridade dos Sensores . . . . .	14
2.1.2 Sensor de Campo Magnético . . . . .	15
2.1.3 Sensor de Posicionamento . . . . .	16
2.1.4 Sensores de Pressão . . . . .	17
2.1.4.1 Altitude . . . . .	17
2.1.4.2 Velocidade Relativa à Massa de Ar . . . . .	17
2.1.5 Sensor de Ultrasom . . . . .	18
2.2 Multiplexagem de Dados Sensoriais . . . . .	19
2.3 Filtragem, Comunicações e Navegação . . . . .	20
2.3.1 Sistema Operativo . . . . .	20
2.3.2 Unidades de Execução . . . . .	20
2.3.2.1 Aquisição e Filtragem . . . . .	21
2.3.2.2 Navegação . . . . .	21

2.3.2.3	Telemetria . . . . .	21
2.3.2.4	Controlo . . . . .	22
2.3.2.5	Actuação . . . . .	22
2.4	Estação Terrestre . . . . .	22
2.4.1	Cockpit Virtual . . . . .	23
2.4.2	Mapa . . . . .	23
<b>3</b>	<b>Método</b>	<b>25</b>
3.1	Criação de Rotas . . . . .	25
3.1.1	Divisão da Área a Vigiar . . . . .	25
3.2	Navegação de Rotas . . . . .	27
3.2.1	Controlo PID . . . . .	27
3.2.1.1	Integrator Windup . . . . .	29
3.3	Comunicação Entre Controlador e Aeronaves . . . . .	29
3.3.1	Telemetria Aeronave para Estação . . . . .	30
3.3.2	Estação para Aeronave . . . . .	30
3.4	Distribuição de Rotas . . . . .	31
<b>4</b>	<b>Resultados</b>	<b>33</b>
4.1	Forma de Avaliação . . . . .	33
4.2	Precisão de Métodos de Navegação . . . . .	35
4.3	Coerência de Performance Entre Vãos . . . . .	36
4.4	Problemas Identificados . . . . .	37
4.4.1	Aceleração Repentina . . . . .	37
4.4.2	Subida Acentuada . . . . .	37
<b>5</b>	<b>Conclusões</b>	<b>39</b>
5.1	Considerações sobre aplicação GroundStation . . . . .	39
5.2	Sugestões para problemas encontrados . . . . .	40
5.2.1	Navegação . . . . .	40
5.2.2	Aceleração Repentina . . . . .	40

5.2.3 Subida Acentuada . . . . .	40
<b>Referências</b>	<b>42</b>
Referências Web . . . . .	42

# Lista de Figuras

2.1	Diagrama de Componentes . . . . .	11
2.2	Integração de Medidas de Sensores . . . . .	13
2.3	Cockpit Virtual . . . . .	23
2.4	Mapa de Operações . . . . .	24
3.1	Divisão em Faixas . . . . .	25
3.2	Divisão em Bandas . . . . .	26
3.3	Cálculo de Largura . . . . .	26
4.1	Plano de Voo . . . . .	34
4.2	Plano de Voo - Aeronave 0 - Sobreposição com trajecto . . . . .	35
4.3	Plano de Voo - Aeronave 1 - Sobreposição com trajecto . . . . .	36
4.4	Trajectos de aeronaves 0 e 1 - Rota distribuída . . . . .	37



# Lista de Equações

2.1 Integração de medição de giroscópio . . . . .	13
2.2 Velocidade angular devolvida por giroscópio . . . . .	13
2.4 Kalman Filter: Previsão . . . . .	14
2.5 Kalman Filter: Correção . . . . .	15
2.5 Cálculo de altitude a partir de pressão . . . . .	17
3.2 Cálculo de áreas de terreno a cobrir . . . . .	25
3.2 PID: Forma Geral . . . . .	27
3.3 PID: Cálculo de delta de deflecção . . . . .	29
3.4 PID: Cálculo de deflecção final . . . . .	29

# 1

## Introdução

O trabalho aqui relatado é uma implementação de um sistema de navegação autónoma para aeronaves não tripuladas, de pequeno porte, baseado em sensores, actuadores e electrónica off-the-shelf. Pretende especificar uma plataforma de baixo custo para utilização pedagógica, de investigação, e de utilização em ambientes reais como, por exemplo, monitorização de incêndios florestais, controlo de fronteiras, obtenção de dados atmosféricos, aplicações de segurança e militares, mas também para base de trabalho para software de controlo, como por exemplo, controladores PID para sistemas autónomos.

A plataforma resultante é uma base de trabalho funcional para que, posteriormente, se possa reforçar e tornar robustos os vários componentes com base em cada aplicação específica. Permite que se derivem várias configurações sem passar pelo tradicional problema da reinvenção da roda.

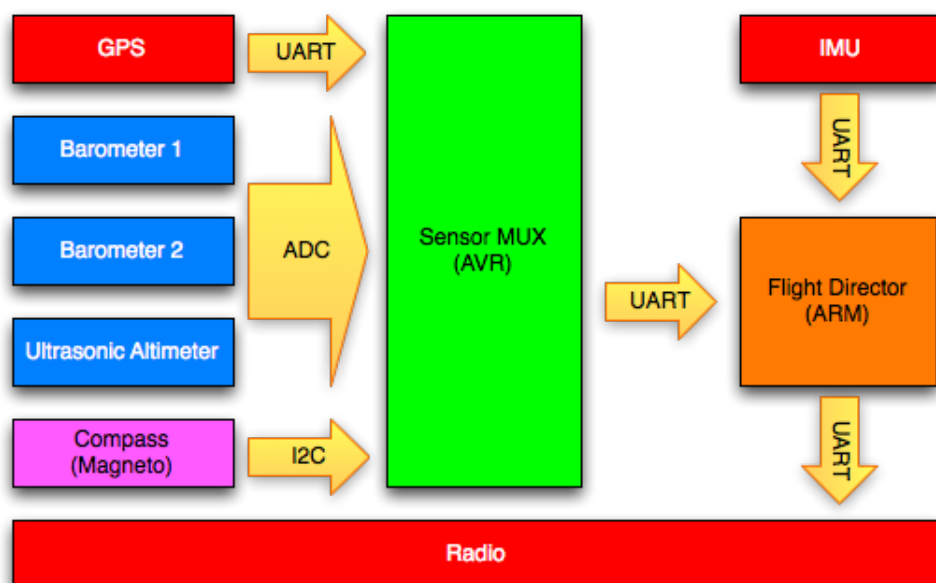
O trabalho aqui relatado pretende desenvolver, sobre o projecto previamente desenvolvido de controlo de aeronaves não tripuladas, um mecanismo de vigilância de uma determinada área geográfica por várias aeronaves, através de técnicas de distribuição de rotas individuais e monitorização simultânea da actividade das mesmas.

## 2

# Enquadramento Teórico

Neste capítulo apresentarei uma descrição completa do sistema proposto. A figura abaixo é um diagrama de blocos com todos os componentes e será descrita a forma proposta de função de cada um além da forma como interagem. Será apresentado o suporte teórico á operação de cada componente e modo de interacção. No capítulo seguinte, demonstrarei como o mecanismo de vigilância distribuído funciona assente nesta arquitectura.

Figura 2.1: Diagrama de Componentes



## 2.1 Sensores

### 2.1.1 Sensores Inerciais

Um sensor inercial é um transducer que reage a alterações de acelerações e rotações. A sua utilização, neste contexto, é o da determinação do estado da aeronave para que o módulo de navegação a consiga controlar. Por exemplo, sendo necessário executar uma curva para ir de um ponto para o outro, a única forma de saber a quantidade de deflecção nas superfícies móveis da aeronave a aplicar, é conhecer a inclinação actual. A diferença da inclinação actual para a desejada a fim de iniciar e manter a curva é um valor que alimenta o módulo de controlo. Sem conhecer a inclinação, não é possível mantê-la dentro de valores seguros. Um piloto humano, em condições de voo VFR (visual flight rules) utiliza a sua visão e ouvidos para determinar a inclinação actual. No entanto, uma máquina não o pode fazer. Em condições IFR (instrument flight rules), um piloto humano é obrigado a ler os instrumentos que lhe indicam o estado actual para controlar a aeronave. O instrumento utilizado é o Horizonte Artificial e é alimentado precisamente por um array de sensores inerciais.

O array de sensores inerciais baseia-se numa unidade IMU (inertial measurement unit) que cria informação de estado - atitude - necessária ao sistema de software que implementa uma INU (inertial navigation unit). O IMU contém três conjuntos de sensores alinhados ortogonalmente em cada um dos três eixos. Cada conjunto contém um acelerómetro biaxial e um giroscópio axial.

Num sistema estático bastaria a presença de um acelerómetro biaxial para determinar a inclinação sobre um eixo através da utilização de trigonometria, visto que a aceleração gravítica se manifesta nos sensores. Como o valor da aceleração  $G$  é constante, seria possível por funções trigonométricas inversas determinar o ângulo necessário para que o valor  $G$  sentido fosse o assinalado pelo sensor. No entanto, num sistema sujeito a acelerações dinâmicas como é o caso de uma aeronave, os sensores de aceleração não conseguem distinguir o componente  $G$  do componente resultante da variação de velocidade (lateral, longitudinal, vertical), logo a estimativa de inclinação é deteriorada pela presença destas acelerações.

Para solucionar este problema, utiliza-se um giroscópio em cada conjunto. A função do giroscópio é medir velocidades angulares num determinado eixo. É razoável presumir que a utilização do giroscópio seria suficiente para a obtenção de estimativa de rotação, mas infelizmente, uma das características das unidades de baixo custo é um fenómeno de drift. Este fenómeno consiste no facto de existir uma margem de erro de variância não zero em cada medição do aparelho. Visto que a característica de medição é uma velocidade angular, é necessário o cálculo integral linear do escalar medido multiplicado pela variação de tempo entre a medida anterior e a presente para determinar o incremento à inclinação actual da seguinte forma

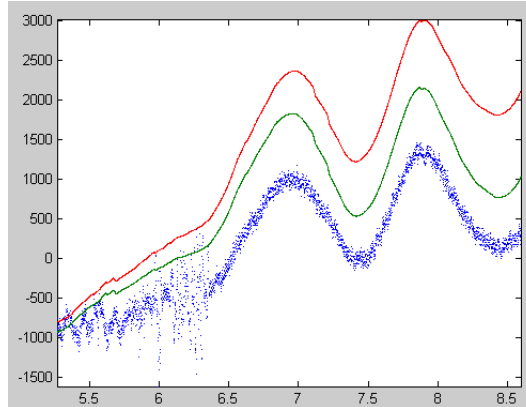
$$\varphi_t = \varphi_{t-1} + \omega \cdot (t - t_{-1}) \quad (2.1)$$

e que se lê: o ângulo  $\varphi$  num determinado instante  $t$  corresponde ao seu valor no instante imediatamente anterior  $t_{-1}$  somado ao produto da velocidade angular  $\omega$  medida pelo giroscópio pela diferença de tempo entre o instante actual e anterior. Isto implica que o erro existente em cada medição é incrementado à estimativa final, pois

$$\omega = x + u \quad (2.2)$$

onde  $x$  é a velocidade angular real e  $u$  é o erro em cada medição, sendo ambos integrados. Ao fim de alguns segundos, a estimativa é completamente inutilizável, como ilustrado na figura seguinte

Figura 2.2: Azul: Acelerómetro; Verde: Real; Vermelho: Giroscópio.



Existe também um problema adicional que é o da necessidade de conhecer com exactidão o estado inicial do sistema para que os incrementos sejam feitos a esses valores, no entanto a presença dos acelerómetros permite conhecê-lo, partindo do princípio que o estado inicial é estático.

### 2.1.1.1 Complementaridade dos Sensores

Neste momento será evidente que os sensores se complementam pois cada um é excelente na determinação daquilo que o outro não consegue. Nos anos 60, o matemático Rudolf Kalman desenvolveu uma ferramenta matemática que pretendia solucionar estes problemas. Na altura, os melhores sensores tinham características de precisão comparáveis às dos sensores de baixo custo que existem na actualidade, pelo que se compreende a motivação para a criação desta ferramenta.

A ferramenta tomou o nome do seu criador - Kalman Filter. Consiste num filtro recursivo de determinação de um estado de um sistema dinâmico a partir de um conjunto de medições com ruído e/ou incompletas. Desde então foram criadas várias derivações deste filtro. Uma das quais é o Extended Kalman Filter cujas características o tornam perfeito para utilização em sistemas aeronáuticos. De facto, este filtro está presente em todas as unidades de medição inercial da indústria. A descrição matemática do sistema é extremamente complexa. No entanto, como curiosidade, apresentarei a lista sequencial de equações utilizadas no projecto. É, no entanto, importante perceber que o sistema é pré-configurado com estimativas de percentagem de erro divulgadas pelos fabricantes dos sensores e que utiliza um sistema de covariância de erros para tentar determinar, a cada passo, qual a quantidade de erro existente em cada medição. Para tal, utiliza os pontos fortes de cada sensor para minimizar o erro do outro sensor. Esta fase chama-se a fase de previsão:

$$\begin{aligned}\hat{x}_{k|k-1} &= f(\hat{x}_{k-1|k-1}, u_k) \text{ [Far98]} \\ P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + Q_k\end{aligned}\tag{2.3}$$

e é executada sempre que é feita uma medida de velocidade angular e o erro é propagado para o estado. Sempre que é feita uma medida nova de aceleração (e

sua extrapolação para orientação) executa-se a fase de correcção:

$$\begin{aligned}
 \tilde{y}_k &= z_k - h(\hat{x}_{k|k-1}) \\
 S_k &= H_k P_{k|k-1} H_k^T + R_k \\
 K_k &= P_{k|k-1} H_k^T S_k^{-1} \\
 \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{y}_k \\
 P_{k|k} &= (I - K_k H_k) P_{k|k-1}
 \end{aligned} \tag{2.4}$$

O princípio base é o de que quando a aeronave está a navegar de um ponto para outro a uma velocidade fixa, os acelerómetros são livres de erros de acelerações parasitas. Este valor chama-se o corrector de longo prazo, visto que este estado de navegação é predominante em cada sessão de voo. Nos momentos em que este estado se verifica, é possível determinar as inclinações reais, compará-las com a estimativa criada pela medição dos giroscópios e, assim, determinar a acumulação de erros nos mesmos. Este erro é decrementado de cada novo valor vindo de um giroscópio.

### 2.1.2 Sensor de Campo Magnético

Um sensor de campo magnético detecta o campo magnético terrestre para determinar em que direcção está apontado e destina-se a complementar a atitude num dos eixos. Assume a função de bússula e substitui o acelerómetro nesse eixo visto que não é possível a partir dessa informação de aceleração determinar um estado absoluto de direcção. Assim, o magnetómetro é o corrector de longo prazo do giroscópio montado no eixo horizontal da aeronave. Estes sensores, tal como os restantes, só são precisos em circunstâncias muito especiais. Sempre que existir inclinação num dos restantes dois eixos, a sua precisão baixa para valores inutilizáveis. Assim, o giroscópio nesse eixo alimenta um estimator de trigonometria inversa para calcular a rotação real nesse eixo. O método utilizado para a fusão sensorial neste eixo é o mesmo filtro Kalman, com parâmetros de configuração distintos - ajustados às margens de erro do magnetómetro.

É importante ter em consideração que o campo magnético terrestre não é constante. Existem flutuações em zonas geográficas conhecidas. Todos os anos é

realizado um levantamento do estado do campo em posições georeferenciadas em latitude e longitude. Essas flutuações devem ser utilizadas como factor de correcção do magnetómetro, pelo que é necessário, também para este fim, conhecer a posição absoluta da aeronave em termos de latitude e longitude.

### 2.1.3 Sensor de Posicionamento

O sistema de determinação de posição adoptado é, evidentemente, o GPS. O sistema é incrivelmente simples em teoria, mas o facto de utilizar radiação electromagnética sujeita a interferências atmosféricas para transmissão do beacon de cada satélite, exige muita complexidade na filtragem do sinal captado pelos receptores. Como curiosidade, também nos módulos rádio de recepção é utilizado um filtro Kalman com parâmetros específicos a esta aplicação.

O facto de o sistema ser absoluto - ou seja, o de devolver valores que não precisam de transformação adicional para a forma pretendida - não significa que não tenha margens de erro. De facto tem-nas e muitas vezes comprometem toda a precisão do cálculo de posicionamento da aeronave. As condições que afectam a precisão são várias mas quase sempre resultam de sinais distorcidos e de fraca visibilidade de satélites. Em qualquer um dos casos, o responsável é quase sempre a atmosfera terrestre e poluentes que aumentam a condensação de água que absorve alguma da radiação do beacon.

O grau de precisão da medição tem que ser conhecido para que o sistema seja fiável. Assim, utiliza-se uma tabela de equivalência de coeficientes entre zero e um para visibilidade de satélites (um dos valores devolvidos pelo módulo) de tal forma que é assignado o valor um a um sinal com o mínimo de satélites para a precisão máxima do módulo (quatro) e valores inferiores para menos satélites. Pela leitura da secção anterior sobre sensores inerciais, pode concluir-se que um dos subprodutos do processo de filtro Kalman é uma estimativa de erro para cada sensor. Assim, a subtracção de esse erro a cada medição de aceleração linear e a sua transformação trigonométrica de coordenadas locais para o sistema NED (North East Down) permite obter acelerações atribuíveis exclusivamente à deslocação da aeronave. Logo, este



valor é utilizado para complementar a medição do módulo de GPS através da dupla integração da aceleração para velocidade e posição. Mesmo que o módulo de GPS fosse isento de erros, este método continuaria a ser necessário visto que um módulo de alta precisão raramente excede taxas de actualização superiores a 1Hz. Compreende-se que seria muito difícil controlar uma aeronave só podendo contar com medições correctas uma vez por segundo.

## 2.1.4 Sensores de Pressão

### 2.1.4.1 Altitude

Os sensores de pressão são essenciais para a determinação de altitude e velocidade relativa. A altitude devolvida pelo sistema GPS não é muito precisa, pelo que a existência de uma estimativa melhor tem a dupla função de também servir para determinar o grau de precisão actual da leitura do GPS. O seu modo de funcionamento é muito simples. Para altitudes inferiores a 1Km, é trivial determinar uma altitude a partir de uma leitura de pressão atmosférica, pois varia de forma conhecida a cada unidade de distância. Essa variação não é linear mas pode ser considerada como tal para valores de altitude inferiores a 1Km e é traduzida através de uma função implementável num algoritmo simples

$$h = 43634.1 \cdot \left[ \left( \frac{QNH}{QFE} \right)^{0.190255} - 1 \right] \quad (2.5)$$

onde  $QNH$  é o valor de pressão atmosférica ao nível do mar e  $QFE$  o valor de pressão atmosférica no local de medição à altitude a determinar.

### 2.1.4.2 Velocidade Relativa à Massa de Ar

O sensor de pressão utilizado para a velocidade relativa é de importância vital para o controlo do motor da aeronave. A velocidade determinada pelo conjunto GPS + acelerómetros diz respeito à taxa de deslocamento em relação ao solo e não em relação à massa de ar. As velocidades relativas medem-se em relação ao fluido envolvente - no caso a atmosfera e a unidade de medição é o Nó.

Um dos estados de controlo da aeronave é a sua velocidade e é utilizado na codificação de acções como levantar vôo (acelerar até velocidade X), navegar (circular com velocidade mínima para sustentação da massa), aterrar (manter altitude minima acima do solo e velocidade nula causa desaceleração e perda de altitude que força o controlo vertical a inclinar a aeronave para cima para aterragem suave). No entanto, em circunstâncias em que o meio não seja estático em relação ao solo, como acontece na presença de deslocamentos de massas de ar ou variações de pressão locais, a velocidade devolvida pelo módulo de posicionamento não traduz a real. A sustentação da aeronave depende da quantidade de ar que passa nas asas e diferença de velocidades entre a massa superior e inferior. A diferença de velocidade produz um potencial de pressão que origina uma força ascendente (visto que em tudo na natureza, há uma tendência de equilibrio da maior concentração para a menor).

Um dos problemas que resulta da não determinação da velocidade relativa é facilmente ilustrável no seguinte exemplo. Imagine-se uma aeronave de pequenas dimensões de velocidade mínima para sustentação de 20 Nós e sujeita a um vento frontal de 20 Nós, ordenada para manter uma velocidade de 30 Nós. Se o controlo de velocidade fosse alimentado pelo módulo de posicionamento, a velocidade sentida seria equivalente a 10 Nós, o que resultaria num sinal enviado para o motor para aumentar a velocidade visto estar abaixo quer da mínima quer da ordenada. Se utilizada a velocidade relativa, seria mantida a velocidade desejada de 30 nós. Em termos absolutos a aeronave deslocar-se-ia mais devagar, mas manteria um consumo de combustível e energia previsível.

### 2.1.5 Sensor de Ultrasom

Mesmo utilizando um sensor de pressão atmosférica para determinação de altitude, a precisão da medição é constrangida pela conversão do sinal analógico para digital. O precisão do conversor depende do conjunto de valores que o sensor possa transmitir a dividir pela quantidade que valores que o número de bits de precisão permite. No caso dos sensores utilizados pode existir uma margem de erro de aproximadamente 60 centímetros. Para além deste facto, a altitude determinada é sempre relativa ao

nível do mar pelo que não tem em consideração a topografia do terreno. Para efeitos de navegação, este erro é suportável ordenando altitudes superiores à do terreno, no entanto, para aterragem e descolagem automatizadas, pode produzir resultados catastróficos. Assim, opta-se pela utilização de um transceiver de ultrasom. O sensor emite um sinal ultrasónico que reflecte no solo e é captado. O tempo total é dividido por dois e transformado em metade da distância percorrida visto a velocidade do som ser conhecida para uma determinada pressão. Esta distância corresponde à altitude em relação ao solo. O sensor tem precisão de 1 centímetro para distâncias entre dois e 600 centímetros e é utilizado para controlar a fase inicial de descolagem e final de aterragem.

## 2.2 Multiplexagem de Dados Sensoriais

O módulo principal de processamento de navegação, comunicações, controlo e telemetria consiste num microprocessador ARM numa placa com memória e periféricos simples. Este microprocessador não dispõe de uma funcionalidade essencial ao interface com os sensores que é um conversor analógico para digital (ADC). Assim torna-se necessário comunicar com um outro microprocessador que disponha dessa funcionalidade. O módulo escolhido foi uma placa com um microprocessador AVR (Atmega128) com duas portas série TTL, vários canais ADC, I2C e SPI. Existem outras funcionalidades interessantes que não são utilizadas no contexto deste projecto.

Os sensores estão ligados às portas ADC do AVR com excepção do GPS e magnetómetro que utilizam, respectivamente, o BUS I2C e uma porta série TTL. O microprocessador corre código desenvolvido para este projecto que a cada iteração capta os vários dados e os coloca num stream binário que é enviado pela segunda porta série TTL. Não é feito qualquer processamento de sinal (excluindo a conversão ADC) no AVR. Esta aproximação permite que se utilize apenas uma porta série TTL no módulo ARM para obter todos os dados de todos os sensores. A segunda porta TTL do módulo ARM é utilizada para comunicações de telemetria e controlo.

Alternativamente, é possível ligar o módulo de GPS directamente ao módulo

ARM e assim utilizar apenas uma porta TTL do módulo AVR, ficando a segunda disponível para leitura de uma stream binária de dados de todos os acelerómetros e giroscópios presentes no IMU que, por sua vez, consiste numa placa em que todos os sensores enviam dados para um multiplexer ADC que os reenvia para um microprocessador PIC16F88. Este microprocessador tem um bootloader que permite reprogramação do chip através da mesma porta TTL que utiliza para output do stream binário. O módulo AVR é programado em modo directo por uma porta ISP (In System Programming) não carecendo de bootloader.

## 2.3 Filtragem, Comunicações e Navegação

Este módulo é o microprocessador ARM descrito anteriormente. Desempenha as tarefas de filtragem dos dados sensoriais, como descrito anteriormente, cálculos de navegação com base no modo de funcionamento seleccionado (autónomo, por canais, ou relay manual), comunicações de telemetria e recepção de instruções de controlo. O código que desempenha estas funções é o âmbito principal do projecto e a sua teoria de operação é descrita de seguida.

### 2.3.1 Sistema Operativo

O módulo executa um kernel linux modificado para microprocessadores sem MMU e com módulos de kernel para interacção com dispositivos de hardware de baixo nível, como servos RC (radio control) actuados por PWM (Pulse Width Modulation) e PPM.

### 2.3.2 Unidades de Execução

O programa principal lança uma thread POSIX por cada função básica e individual. Cada elemento do diagrama acima representa uma destas unidades de execução. A teoria de operação de cada unidade de execução é descrita de seguida e os detalhes de gestão de concorrência e sincronização são apresentados no capítulo de implementação.

### 2.3.2.1 Aquisição e Filtragem

O processo de filtragem sensorial é executado nesta thread. É preenchida uma estrutura com os dados sensoriais lida de uma porta série TTL, que é processada e alimenta uma outra estrutura de dados filtrados contendo posições, acelerações e inclinações. Esta estrutura é utilizada pelo módulo de navegação.

### 2.3.2.2 Navegação

Esta thread é responsável pela determinação de quantidades de deflecção necessárias em cada superfície móvel da aeronave para cumprir um plano de voo configurado pelo operador humano. Essas quantidades são determinadas pelo modo de operação seleccionado. No caso de funcionamento manual, a thread não executa qualquer acção de cálculo de deflecção, minimizando a quantidade de ciclos de processador. Esta thread utiliza uma série de submódulos do sistema que contém informação específica para cada aeronave como, por exemplo, coeficientes de deflecção específicos para cada envelope aerodinâmico, deflecções máximas a permitir independentes do limite que cada actuador permite, potência de motor, velocidade mínima de sustentação, entre outras. O cálculo de quantidade de deflecção baseia-se na tradicional aproximação PID (Proportional, Integrative, Derivative). A teoria de operação é muito simples. Consiste na atribuição de um valor chamado de erro que é a diferença entre o estado desejado e o actual. Os três coeficientes criam valores que permitem ajustar quão imediata é a deflecção, quanto o tempo desejado até atingir o estado máximo e quão depressa se corrigem erros cujo estado não é atingido dentro do tempo desejado. Os detalhes de utilização deste mecanismo são apresentados no capítulo de implementação.

### 2.3.2.3 Telemetria

A thread de telemetria é vital para a monitorização remota da aeronave. Prepara um stream binário com os dados sensoriais filtrados e estado do plano de voo actual e envia-o pela porta de série TTL ligada ao modem rádio. É a unidade de execução mais simples e compacta de todo o sistema.

#### 2.3.2.4 Controlo

A thread de controlo é aláloga à de telemetria em sentido inverso. Permite a recepção de dados de controlo da estação remota e modificação de estados internos de acordo com as ordens. É esta unidade que permite, por exemplo, a comutação entre modo autónomo, por canais ou manual, o upload de planos de voo em tempo real, mesmo durante a execução de um plano anteriormente enviado.

#### 2.3.2.5 Actuação

Esta thread é responsável pela transformação dos valores de actuação recebidos da estação terrestre ou calculados pelo módulo de navegação em valores dentro da escala aplicável aos servos electromecânicos que movem as superfícies de actuação. A actuação em si é realizada pelo módulo de kernel que utiliza os valores aqui transformados e os envia para o receptor de rádio que, tal como num tradicional modelo telecomandado, está ligado aos servos. Este receptor foi transformado de tal modo que o canal que transporta o sinal multiplexado com todos os comandos para os vários sinais para cada servo foi cortado e cada ponta ligada aos pinos general purpose configurados pelo módulo de kernel acima referido. Assim, em modo manual, o módulo lê os comandos enviados pelo transmissor de rádio de modelo tradicional e reenvia-os pela outra ponta. Em modo assistido ou autónomo, ignora esses dados e injecta os calculados no receptor.

Esta forma de funcionamento serve a dupla função de permitir a utilização de um tradicional rádio de modelos e ainda a de evitar ter pinos desperdiçados para ligação a cada servo, no microprocessador ARM.

## 2.4 Estação Terrestre

A estação terrestre é uma aplicação Mac OS X que mostra um cockpit virtual em realidade aumentada e um mapa de área de operações com pistas visuais da rota comum e representação de bandas de monitorização assignadas a cada aeronave.

### 2.4.1 Cockpit Virtual

Esta janela contém um plano bidimensional no qual é desenhada uma textura OpenGL contendo uma frame de um dispositivo de captura de imagem, como por exemplo, o receptor da câmara wireless a bordo da aeronave. Sobre esse plano são desenhadas várias formas e informações que representam o estado da aeronave, um horizonte artificial e alguma informação de controlo como os estados desejados em voo autónomo e potência de motor.



Figura 2.3: Controlo por canais e selecção de tipo de controlo

### 2.4.2 Mapa

Nesta janela é apresentado um mapa da área de operações sobre o qual se podem desenhar os vários waypoints que constituem a missão. Por waypoint é possível editar com pormenor a sua latitude, longitude, altitude e velocidade de chegada desejada em Nós. A imagem pode ser criada com um pequeno utilitário desenvolvido para o projecto que transporta do repositório de tiles georeferenciados do google o conjunto de imagens que constitui a área e constrói a imagem maior que representa toda a área. Após o desenho do plano de missão é possível parametrizar qual dos pontos constitui

o início da área de separação das aeronaves que vai ser alvo de vigilância. Para tal, é necessário fornecer ao sistema valores que indicam quantas aeronaves estão envolvidas na missão e qual a largura da área a observar em metros. Desta forma, o sistema desenha sobre a área a observar, algumas pistas visuais sobre qual a área assignada a cada aeronave.



Figura 2.4: Desenho de missão e scripts por waypoint



# 3

## Método

### 3.1 Criação de Rotas

#### 3.1.1 Divisão da Área a Vigiar

A definição de cada rota individual dever-se-à conseguir através do método de divisão por bandas. Uma área qualquer é circunstrita por um rectângulo dividido em um número de faixas  $n = \textit{aparelhos}$  paralelas - ilustrado em 3.1.

Figura 3.1: Divisão em Faixas:  $L_p$  = Faixa de um elemento



De seguida, cada faixa assignada a uma aeronave é dividida num número de bandas paralelas a percorrer em S de tal forma que cada uma corresponda à área que a câmara de abertura  $\phi$  consegue cobrir - ilustrado em 3.2.

Para calcular a largura de cada banda,  $L_{ps}$ , utiliza-se o método ilustrado em 3.3 em que  $\phi$  é a abertura da câmara,  $\delta$  é metade desse ângulo, e a forma de cálculo é

$$L_{ps} = 2 \cdot \tan \delta \cdot h \quad (3.1)$$

$$N_{bandas} = L_p / (2 \cdot \tan \delta \cdot h)$$

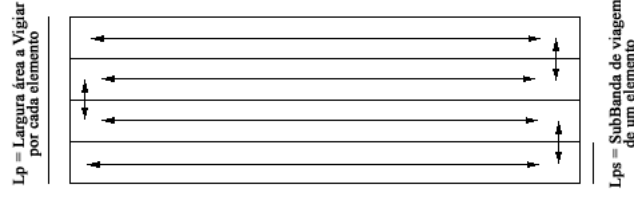
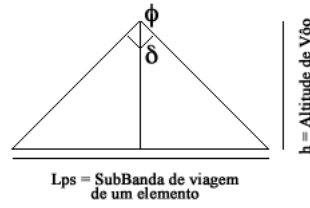
Figura 3.2: Divisão em Bandas:  $L_p$  = Faixa de um elemento

Figura 3.3: Cálculo de Largura



No entanto, a divisão poderá ter resto. Assim, adiciona-se um a  $N_{bandas}$  e corrige-se a largura de cada banda, diminuindo a sua dimensão. Deste modo, garante-se que não há nenhuma zona de fora, mas com a consequência de alguns pontos serem alvo de duas passagens.

Cada aparelho tem como ponto inicial de rota o ponto médio do segmento de largura da primeira banda. A circulação é feita pela navegação até ao final da banda e mudança para banda adjacente em sentido oposto, fechando um ou várias formas 8. Assim, o percurso é percorrido um  $n$  número de vezes ou durante um tempo  $\Delta t$ .

O método aqui delineado tem como inconveniente o facto de todos os cálculos serem feitos na groundstation. Neste ambiente, o sistema de coordenadas local no mapa é em pixels. Por este motivo, é necessário alguma complexidade algorítmica adicional para fazer corresponder uma distância bidimensional em pixels a uma altura - necessária para cálculo de zona coberta pela câmara - em pés. Visto a terceira dimensão não estar representada, é necessário partir do princípio que a escala de altitude é igual à horizontal. Assim, determina-se a quantidade de metros por pixel na escala horizontal, faz-se a conversão para pés e utiliza-se o coeficiente encontrado para determinar a quantidade de pixels correspondentes à altitude programada em cada waypoint. Assim, é possível desenhar uma pista visual que represente a área de

terreno coberta pela câmara com base no método trigonométrico na equação 3.1.

## 3.2 Navegação de Rotas

A navegação dos waypoints definidos por missão faz-se através de um sistema de dois loops de controlo de baixo e alto nível. O loop de alto nível determina a posição e estado da aeronave e calcula qual a direcção para atingir o próximo waypoint com a altitude e velocidade desejados. Os valores encontrados - direcção a tomar, velocidade alvo e velocidade de subida necessária - são passados para o loop de baixo nível - o de controlo de superfícies móveis - que alimenta um algoritmo por superfície e controlo de motor através do cálculo de diferença entre estado desejado e actual. Este algoritmo é executado para os vários componentes de um estado - a altitude, posição, atitude e velocidade da aeronave. O algoritmo de controlo calcula as deflecções necessárias nas várias superfícies móveis para passar do estado actual para o desejado através de um processo de teoria de controlo baseado no sistema PID - proportional, integral, derivative.

A passagem de um waypoint para o seguinte é feita avaliando a distância entre a projecção bidimensional da posição actual sobre a superfície terrestre e a mesma projecção do waypoint destino. Quando esta distância é inferior a um determinado valor em metros, o loop de alto nível avança o índice de waypoint destino.

### 3.2.1 Controlo PID

O processo de controlo PID baseia-se no cálculo de três coeficientes de ganho que são multiplicados pelo erro calculado e somados para criação do valor final enviado para a superfície móvel. Os três componentes são calculados pela forma geral

$$u_n = K_x \cdot (r_n - y_n) \quad (3.2)$$

onde  $K_z$  é um coeficiente de ganho de um dos três componentes,  $r_n$  é a referência de estado desejado e  $y_n$  é o valor de estado actual.

A componente P (proportional) é um valor directamente proporcional ao erro,

apenas multiplicado por um coeficiente de ganho. Este ganho permite ajustar quão importante é o erro no cálculo final. Em situações muito simples, este componente por si só permite um controlo estável de alguns sistemas, mas tende a estabilizar o sistema abaixo do estado desejado. Por exemplo, se o aplicarmos a um controlo de velocidade de um automóvel cuja velocidade desejada fosse  $100 \text{ Km}\cdot\text{h}^{-1}$  e a velocidade actual fosse inferior, o erro seria a diferença entre as duas. No entanto, à medida que o erro fosse diminuindo pela aceleração imposta pela componente P, o input para acelerador também diminuiria até chegar a um ponto em que seria insuficiente para fazer aumentar a velocidade. Faria sentido, neste caso, utilizar derivadas de primeira ordem para alimentar o sistema.

No caso de uma aeronave perfeitamente equilibrada, um input de superfícies móveis nem sempre permite que a mesma se mantenha nesse estado, devido a factores de desequilíbrio externos, como variações de densidade do ar e ao próprio vento. Neste caso é necessário utilizar uma componente que nos indique a acumulação de erro ao longo de vários instantes. Desta forma, quanto mais tempo demorarmos a chegar ao estado desejado (já que o controlo P pára perto dele), maior é a influência de I no estado final.

O problema inerente ao controlo I é o de que não tem em conta o quão depressa está a chegar ao estado desejado. Assim, à medida que o ganho I aumenta para atingir o objectivo, a velocidade de aproximação a esse estado também aumenta, fazendo com que facilmente se ultrapasse esse limite, invertendo o sinal de P e I e fazendo com que o estado actual oscile numa forma sinusoidal em torno do valor de referência. Para evitar que isto aconteça, utiliza-se a componente D (derivative) que nos permite ter uma noção de velocidade de aproximação. Quanto mais perto estamos do destino, mais baixo o valor de output D. Assim evita-se o *overshoot* do valor de referência. O cálculo de valor final em cada superfície é feito pela seguinte forma utilizada em projectos de controlo de veículos e simulações.<sup>1</sup>

---

<sup>1</sup>Contribuição de Roy Ovesen de autoria não identificada para o simulador de voo opensource FlightGear - Ver pormenor: <http://www.flightgear.org>. Implementado em pid.c. Navegação high level implementado em navigation.c.

$$\Delta u_n = K_p \cdot \left[ (eP_n - eP_{n-1}) + \left( \frac{T_s}{T_i} \cdot e_n \right) + \frac{T_d}{T_s} \cdot (eDf_n - 2 \cdot eDf_{n-1} + eDf_{n-2}) \right] \quad (3.3)$$

$$u_n = u_{n-1} + \Delta u_n \quad (3.4)$$

### 3.2.1.1 Integrator Windup

Integrator Windup é um problema associado ao sistema de controlo PID que se caracteriza pelo aumento da componente I na impossibilidade de atingir o estado destino. Pegando no exemplo de controlo de velocidade anterior, imagine-se o caso de um automóvel a subir uma colina a uma velocidade inferior à desejada, mas com o acelerador já no máximo. Neste caso, a impossibilidade do motor desenvolver potência suficiente para aumentar a velocidade faz com que a componente I aumente muito, visto que esta é alimentada pelo acumular do erro. A cada instante que passa, o valor pode efectivamente duplicar. O problema só se manifesta quando, ao chegar ao topo da colina, o automóvel retomar a marcha sem esforço. Nesta situação, o valor anteriormente alto de I faz com que o automóvel acelere demasiado e potencialmente muito para lá do pretendido. O sistema torna-se instável, imprevisível e perigoso.

A forma de contornar o problema pode ser estabelecer um limite máximo para o componente I ou, mais elegantemente, aplicar o limite máximo em conjunto com uma componente dependente de tempo e erro que faça com que a subida de I seja não linear. O ajuste desta componente deve ser feito por experimentação e erro em cada uma das superfícies a controlar.

## 3.3 Comunicação Entre Controlador e Aeronaves

Para enviar às várias aeronaves os dados de cada missão é necessário uma forma de comunicação. No modelo não simulado existe apenas um meio de comunicações rádio. Por natureza o canal é partilhado por todos os intervenientes tal como numa rede WiFi. Assim, e dependendo da configuração do modem rádio, é possível endereçar individualmente as aeronaves apenas através do conteúdo do datagrama. Para que a

simulação seja uma réplica do ambiente real, as comunicações da groundstation para as aeronaves utilizam o mecanismo de broadcast. Desta forma, o formato do datagrama é exactamente igual ao do ambiente de produção e o código que executa a criação e parsing dos datagramas é partilhado.

O facto de o canal de comunicações real fornecer uma abstracção de alto nível de um canal serial normal acessível através de leitura e escrita em file descriptors (e que em ambiente simulado tem uma camada de tradução para UDP) faz com que seja necessário marcar cada datagrama com um marcador de início, tamanho, e fim. Existem vários tipos de datagrama a circular no meio de comunicações. Apresento abaixo os vários formatos dos mesmos.

### 3.3.1 Telemetria Aeronave para Estação

Este é o datagrama enviado por cada aeronave para a groundstation e que contém informação de estado da mesma. Resume-se a uma sequência de valores com informação de posição e orientação. Numa futura revisão deverá ser extendido para enviar um bloco completo de informação dos sensores como, por exemplo, número de satélites visíveis pelo módulo GPS. Como esta informação não compromete a utilização da groundstation, poderá ser enviada a uma frequência inferior à anterior.

### 3.3.2 Estação para Aeronave

A estação terrestre envia para a aeronave datagramas para funções distintas para comutar o modo de operação entre manual telecomandado, manual com dispositivo HID, assistido por canal ou canais e missão (navegação autónoma); ou ainda para, em modo missão, enviar para a aeronave os dados da missão.

- char Header
- char PayloadType
- PayloadData
- char Footer

Onde PayloadType faz comutar de imediato o modo de controlo com os dados do campo seguinte da seguinte forma:

PayloadType:0, PayloadData contém um vector de 4 floats com as deflecções de superfícies móveis desejadas

PayloadType:1, PayloadData contém um vector de 3 floats com os estados desejados (heading, altitude, airspeed)

PayloadType:2, PayloadData tem um byte inicial com o número de waypoints de payload e seguido por um vector de waypoints indicando, cada um, estado desejado numa sequência de 4 floats (latitude, longitude, airspeed, altitude). O módulo flight-director carrega estes valores para uma linked list que contém a missão em memória.

### 3.4 Distribuição de Rotas

Para que seja possível cada aeronave vigiar uma determinada parte do terreno, é necessário saber a qual se deve assignar cada faixa do mesmo. Uma possibilidade é o de as aeronaves negociarem entre si quem vigia que parte, mas o facto de serem sistemas diferentes com uma base temporal não comum torna-o difícil. Existem métodos de o garantir, mas num meio com comunicações sem garantias de fiabilidade é vital que o operador humano consiga prever qual o aparelho que vai executar determinada tarefa. Por este motivo, o método de distribuição de rotas é o mais simples possível e consiste na utilização de endereçamento aplicativo de cada aeronave com base num índice sequencial. A cada um destes índices, faz-se corresponder o mesmo valor em índice de faixa de terreno.

Esta forma de assignação de tarefas permite simplificar toda a geração de rotas. Assim, a aplicação de monitorização envia exactamente o mesmo payload com dados de missão a cada aeronave, sendo que cada uma determina para cada waypoint, se o deve seguir ou se é objectivo de outra aeronave<sup>2</sup>. Pegando neste conceito, pode-se pensar numa revisão futura em que há handling de falhas incapacitantes de uma

---

<sup>2</sup>Cada waypoint tem os campos: latitude, longitude, altitude, airspeed, uavid onde uavid é o índice da faixa a que pertence e, portanto, da aeronave que o vai sobrevoar

aeronave, permitindo que outra a ela adjacente execute um plano de vôo cujo conjunto de waypoints seja a justaposição de ambos os subconjuntos.



# 4

## Resultados

### 4.1 Forma de Avaliação

Para a avaliação dos resultados obtidos com o sistema proposto foram feitos 3 vôos simulados com o mesmo plano de vôo e foi feita uma análise de desempenho a cada simulação. Este método permite verificar o quão estável e previsível o sistema é (coerência de resultados entre os vôos) e qual a precisão dos métodos de navegação e comunicações (qualidade dos dados de cada vôo). O ambiente de simulação é o simulador X-Plane e a aeronave utilizada foi o modelo da plataforma real para a qual os coeficientes de ganho do módulo PID foram otimizados (Great Planes PT60<sup>1</sup>). Foram simuladas duas aeronaves em simultâneo em dois computadores com processadores de endianness distintos<sup>2</sup>.

O plano de vôo consiste em descolagem autónoma, navegação até ponto de separação de aeronaves, navegação de cada segmento de forma individual em cada simulador até ponto de reunião, continuação para alinhamento com pista de aterragem e aterragem autónoma. O modo loop não estava activo pelo que a simulação pára quando a aeronave se imobiliza após aterragem.

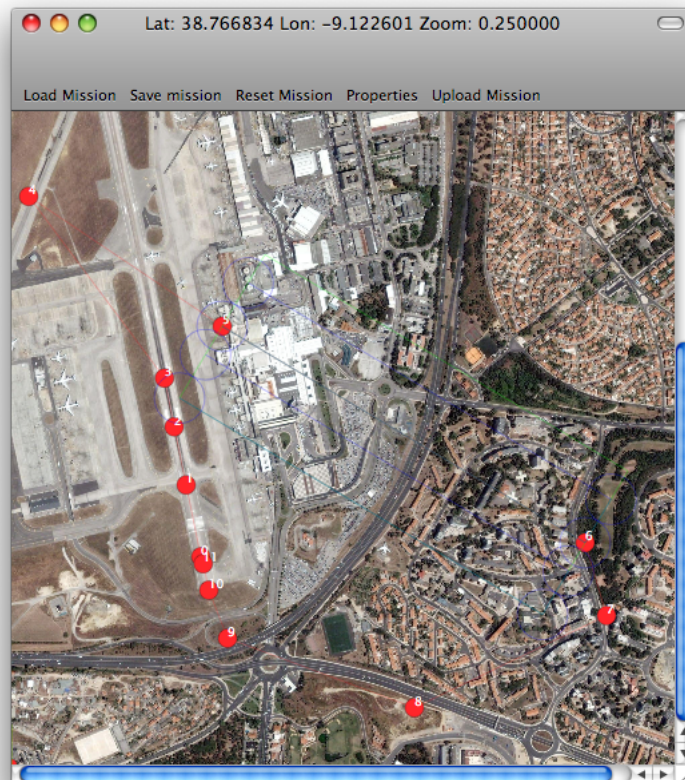
---

<sup>1</sup>Envelope aerodinâmico utilizado com muita frequência em projectos de vôo autónomo pelas características de estabilidade e tolerância a vôo lento que apresenta

<sup>2</sup>Para teste de fiabilidade de comunicações, visto os pacotes serem dados binários. Foram utilizados um PowerPC970 (PPC64) e um CoreDuo (IA32)

Para uma análise mais simples dos resultados obtidos, apresento uma imagem do plano de voo - figura 4.1 - tal como representado na groundstation com anotações de cada pista visual. De seguida, farei a sobreposição do percurso percorrido por cada uma das duas aeronaves - figuras 4.2 e 4.3 - que participam na missão sobre o plano de voo e na última imagem apresento ambas sobrepostas simultaneamente - figura 4.4. Esta análise permite avaliar a precisão dos métodos de navegação utilizados. Apresentarei também uma versão desta última imagem para cada um dos três vôos realizados para comparação e determinação da coerência de resultados entre os vôos. O plano de voo<sup>3</sup> assenta sobre a área do Aeroporto Internacional da Portela<sup>4</sup>. Consiste

Figura 4.1: Plano de Voo



numa descolagem autónoma da pista 35, monitorização duma área sobre a Avenida

---

<sup>3</sup>Fig 4.1

<sup>4</sup>ICAO: LPPT

de Berlim e aterragem na mesma pista. Foi desenhado para ser executado por duas aeronaves a uma altitude de 400 pés<sup>5</sup>. O simulador foi preparado para ignorar toda a topografia de tal forma a que a altitude na área - cerca de 300 pés - foi correspondida a 0 para simplificar a análise posterior dos vôos. Para facilitar a leitura do plot de trajecto de cada aeronave sobre o plano de vôo, foram aplicadas transparências às layers de plot nas imagens. Como nota adicional, o ponto de separação das aeronaves em rotas independentes é o waypoint 4 e o de rendezvous é o 7. Os waypoints 5 e 6 são os designadores de área de vigilância. Permitem, juntamente com um parâmetro de largura da área, circunscrever a mesma e assignam automaticamente os waypoints imediatamente antes e depois como ponto de separação e junção de percursos.

Figura 4.2: Plano de Vôo - Aeronave 0 - Sobreposição com trajecto



## 4.2 Precisão de Métodos de Navegação

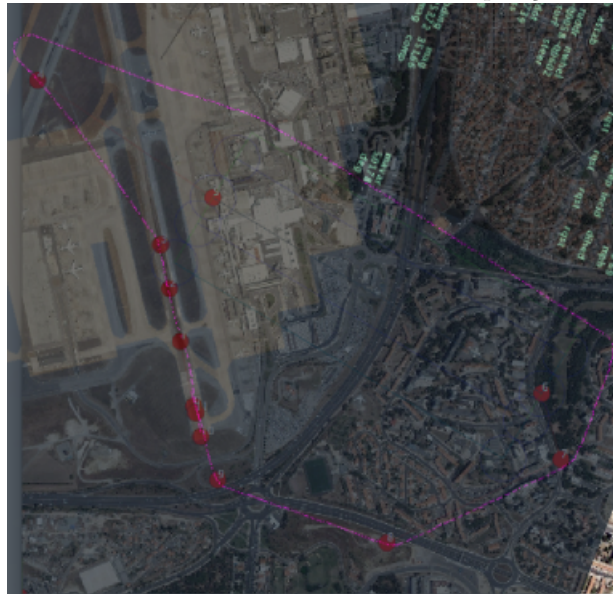
Após análise de resultados obtidos com simulador, constata-se que o plano foi seguido com um desvio marginal do percurso pretendido. No entanto, verifica-se que na execução de curvas superiores a 45 graus, a aeronave tende a entrar no waypoint

---

<sup>5</sup>Cerca de 120 metros

seguinte com um desvio lateral considerável que é corrigido não de imediato, mas durante todo o percurso para o próximo waypoint. O motivo deste desvio é o de que a velocidade da aeronave impõe-lhe um raio de curvatura demasiado alargado. Este problema e outros foram identificados e serão apresentadas, no capítulo seguinte, possíveis resoluções.

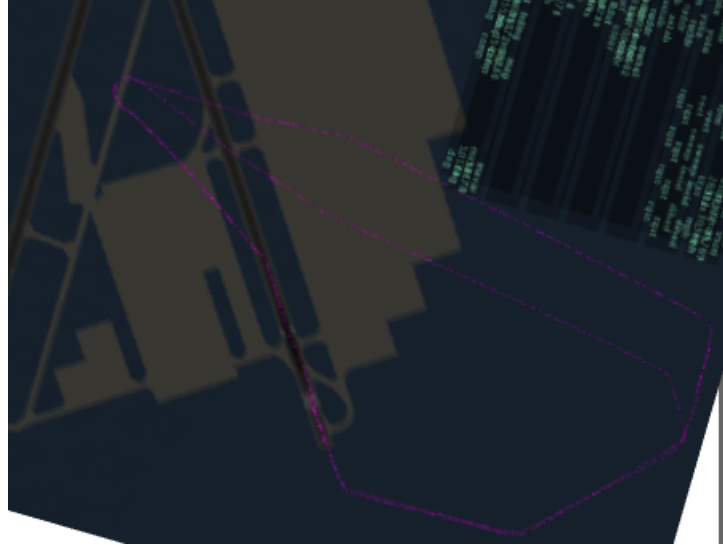
Figura 4.3: Plano de Vôo - Aeronave 1 - Sobreposição com trajecto



### 4.3 Coerência de Performance Entre Vôos

Constatou-se o que já era esperado por observação do comportamento de uma única aeronave em modo de vôo não distribuído e com loop de waypoints activo (repetição interminável da missão) que é uma tendência para repetir exactamente o mesmo percurso com desvios laterais e verticais de centímetros. Tal deve-se à estabilidade da aeronave, à invariância do clima simulado e à precisão milimétrica dos algoritmos de controlo de alto e baixo nível.

Figura 4.4: Trajectos de aeronaves 0 e 1 - Rota distribuída



## 4.4 Problemas Identificados

Para além do problema identificado em 4.2, foram identificados alguns outros que podem condicionar a integridade física dos aparelhos.

### 4.4.1 Aceleração Repentina

No caso de se ordenar a uma aeronave uma aceleração repentina, por exemplo, ordenando uma aceleração de zero até velocidade destino logo no primeiro waypoint, a potência debitada pelo motor e as forças de torção que impõe sobre o corpo da aeronave fazem com que esta se incline para um lado e para cima de forma muito acentuada. Este facto dificulta a obtenção da velocidade ordenada e faz com que seja aumentado o input do motor. Neste caso, o outcome mais provável é que o aparelho se despenhe a baixa altitude a com muita energia de torção, pondo em causa a sua integridade e a dos hipotéticos operadores humanos.

### 4.4.2 Subida Acentuada

Na mesma linha, se se ordenar a uma aeronave que suba de tal forma que a diferença de altitudes seja maior que um certo valor dependente da velocidade, é possível que

entre em inclinação acentuada que, por sua vez, faz diminuir a velocidade, fazendo, de novo, aumentar o input do motor e levando à mesma situação descrita anteriormente.

# 5

## Conclusões

Este projecto demonstrou que é, de facto, possível a construção de um sistema de monitorização distribuída com um mínimo de complexidade, e com potencial para servir de base a outros projectos de melhoramento e aumento de robustez. Foi também possível o desenho de uma aplicação de controlo que permite a um operador sem experiência ou formação específica controlar várias aeronaves em simultâneo recorrendo a um interface simples e intuitivo. O comportamento das aeronaves mostrou-se estável, assinalando a robustez dos mecanismos de controlo. O método de construção de rotas individuais foi posto em prática em relativamente pouco tempo, mesmo sendo o ponto principal do projecto, evidenciando a sua simplicidade. E a análise de resultados mostra que, esta simplicidade não constituiu entrave à boa performance do algoritmo nem à qualidade do resultado pretendido.

### 5.1 Considerações sobre aplicação GroundStation

O software desenvolvido para o projecto está apto a funcionar em vários sistemas embedded, tendo em consideração as necessárias adaptações pontuais às diferenças de hardware inerentes. No entanto, saliento que a aplicação GroundStation foi desenvolvida numa tecnologia que, embora flexível e poderosa, não é utilizável em sistemas que não Mac OS X. Fica como consideração para desenvolvimentos futuros, o port da

aplicação para um ambiente verdadeiramente cross-platform como, por exemplo, Qt<sup>1</sup>, wxWidgets ou até mesmo Java.

## 5.2 Sugestões para problemas encontrados

### 5.2.1 Navegação

Como mencionado no capítulo anterior, em 4.2, foi identificado um problema que resulta do método de navegação utilizado. Este não tem em conta que após atingir um waypoint, é necessário um período de tempo até que aeronave se posicione em rota para o seguinte. Uma solução possível é a determinação, em tempo real, do raio de curva necessário para entrar no próximo waypoint com a velocidade desejada, e iniciar a curva antes do waypoint de viragem. Uma forma de visualizar o conceito é pensar no trajecto que uma aeronave teria a executar um plano de vôo com uma viragem de 90 graus sem curva. Antes de atingir o ponto de viragem, a aeronave iniciaria a curva por forma a entrar no trajecto seguinte sem desvio lateral.

### 5.2.2 Aceleração Repentina

Este problema pode ser mitigado através da construção cuidada de um plano de vôo cuja sequência de waypoints para descolagem e aterragem tenha aumentos graduais de velocidades alvo. Existe uma outra possibilidade que consiste em modificar o controlo PID de potência de motor para que o erro que alimenta o algoritmo não seja a diferença entre velocidade actual e desejada, mas sim um erro de aceleração entre o máximo seguro e zero. Este valor seria, anteriormente, determinado por análise do erro de velocidade já mencionado.

### 5.2.3 Subida Acentuada

Tal como no ponto anterior, é possível modificar o controlo PID de deriva de forma a que seja alimentado pela diferença entre velocidade vertical desejada e actual, sendo

---

<sup>1</sup>Isento de custos para fins não comerciais



o valor desejado calculado com base na diferença de altitudes, distância ao próximo waypoint e velocidade em relação à massa de ar. Isto asseguraria uma subida linear.

# Referências

- [Far98] Jay A. Farrell. *The Global Positioning System And Inertial Navigation*. McGraw-Hill Professional, 1998.
- [Koc03] Stephen Kochan. *Programming in Objective C*. Sams, 2003.
- [OB05] OpenGL Architecture Review Board, Dave Shreiner, Mason Woo, Jackie Neider, Tom Davis. *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL(R), Version 2*. Addison-Wesley Professional, 2005.
- [OB03] OpenGL Architecture Review Board, Dave Shreiner. *OpenGL(R) Reference Manual: The Official Reference Document to OpenGL, Version 1.4*. Addison-Wesley Professional, 2003.

## Referências Web

*Aviation Formulary V1.43* <http://williams.best.vwh.net/avform.htm>

*X-Plane* <http://www.x-plane.com>

*FlightGear* <http://www.flightgear.org>