

29

**UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA E DE SISTEMAS
MECÂNICOS**

*nota final
9,81 (nove e oito)*
Wagner

MONITORAMENTO VISUAL DE AERONAVES PARA TELEOPERAÇÃO

Trabalho de formatura apresentado à
Escola Politécnica da Universidade
de São Paulo para obtenção do título
de graduação em Engenharia

Elcio Ricardo Lora
Guilherme Oliveira Mello Mazzini

Orientador: Fabio G. Cozman

Área de Concentração:
Engenharia Mecatrônica

São Paulo
2005

**UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA E DE SISTEMAS
MECÂNICOS**

MONITORAMENTO VISUAL DE AERONAVES PARA TELEOPERAÇÃO

Trabalho de formatura apresentado à
Escola Politécnica da Universidade
de São Paulo para obtenção do título
de graduação em Engenharia

**Elcio Ricardo Lora
Guilherme Oliveira Mello Mazzini**

Orientador: Fabio G. Cozman

**Área de Concentração:
Engenharia Mecatrônica**

**São Paulo
2005**

DEDALUS - Acervo - EPMN



31600011857

1495121

FICHA CATALOGRÁFICA

Lora, Elcio Ricardo

Monitoramento visual de aeronaves para teleoperação/ E. R.
Lora; G. O. M. Mazzini; -- São Paulo, 2005.

65p.

Trabalho de Formatura - Escola Politécnica da Universidade
de São Paulo. Departamento de Engenharia Mecatrônica e
Sistemas Mecânicos

1.Telemetria 2.Aeronaves 3.Simulação computacional
4.Operação de aeronaves I.Universidade de São Paulo. Escola
Politécnica. Departamento de Engenharia Mecatrônica e
Sistemas Mecânicos II.t.

DEDICATÓRIA

À Madalena, minha mãe e Edelcio, meu pai, pelo exemplo de dedicação, obstinação, força de vontade e honestidade. Por terem sempre lutado pelo meu futuro e quererem sempre o melhor. Pelo conforto nas horas que necessitei e pelo apoio nas horas mais complicadas. Por serem o espelho daquilo que desejo sempre me tornar.

Ao meu irmão Edelcio, pelo exemplo de força de vontade e companheirismo. Por ter me ajudado nas horas em que precisei.

Aos meus amigos e colegas que tornaram essa jornada de cinco anos menos árdua, e em especial aos amigos que fiz durante o curso de engenharia.

Elcio

À minha família, principalmente minha mãe Neuza e minha irmã Ana Paula, pois sem o apoio delas o esforço para concluir este trabalho seria muito maior.

À minha namorada Amanda, pelo carinho, apoio e paciência por todos os momentos em que tive que trocá-la por um computador.

Guilherme

AGRADECIMENTOS

Ao Prof. Dr. Fabio G. Cozman, nosso orientador, pela ajuda neste projeto.

À Escola Politecnica, por ajudar na nossa formação acadêmica.

RESUMO

Este trabalho tem como objetivo efetuar a simulação de uma transmissão em tempo real de dados entre uma aeronave e uma estação no solo. Para realizá-la são enviadas imagens de câmeras instaladas no avião, além de informações de sensores como o GPS e unidade de medição inercial (UMI), para a estação-base. Após o recebimento desses dados, a estação no solo faz a sua interpretação para monitorar a trajetória da aeronave, viabilizando a pilotagem remota.

Um requisito importante para este projeto é a transmissão de vídeo sem atrasos significativos e, para isso, houve a escolha do tipo de transmissão de vídeo em modo analógico. Devido à presença de quatro câmeras no avião deve ser utilizado um chaveamento para escolher a transmissão entre as câmeras.

A transmissão de dados de telemetria dá-se através de dois modems de rádio freqüência: instalados um no avião e outro na estação-base. Para que isto seja possível, utiliza-se o protocolo TCP/IP e a comunicação via sockets.

Ao final é apresentada uma interface gráfica ao usuário mostrando os dados principais de um avião como altitude, velocidade, nível de combustível e bateria além de outros.

ABSTRACT

This work aims at simulating a real-time data transmission with an aircraft and a base station. In order to perform it, video from cameras installed inside the airplane and also information from sensors like GPS and IMU (inertial measurement unit) are sent to the base station. After receiving the data, the ground station interprets it and then it becomes possible to monitor the airplane trajectory and to perform remote driving.

An important requirement to this project is video transmission without significant delay and so, the choice of video transmission was the analog. Due to the presence of four cameras in the aircraft, it should be used a switching system to choose from which camera the image will be displayed.

Telemetry data is transmitted by two radio-frequency modems: one installed inside the airplane and the other connected to the base station. The TCP/IP protocol and socket communication have been employed in this communication scheme.

A graphical user interface, showing main airplane data such as altitude, speed, fuel level, energy supply and others, was developed during the work.

LISTA DE FIGURAS

Figura 1. Potência de irradiação de uma antena isotrópica	7
Figura 2. Potência de irradiação de um dipolo de meia onda.....	8
Figura 3 - Polarização de uma antena.....	9
Figura 4 - Dimensões de um dipolo de meia onda	10
Figura 5 - Potência de irradiação de um dipolo de meia onda.....	17
Figura 6 - Sistema de transmissão de vídeo digital	27
Figura 7 - Sistema de transmissão de vídeo analógico	29
Figura 8 - Tela do Flight Simulator	31
Figura 9 – Componentes de Hardware	32
Figura 10 - Exemplo de GridBagLayout	50
Figura 11 - Exemplo de SpringLayout	51
Figura 12 - Exemplo de aplicativo que utiliza Timer	53
Figura 13 - Interface otimizada.....	56
Figura 14 - Ponteiro no formato GIF	59
Figura 15 - Interface final	64

SUMÁRIO

1. Introdução	5
2. Revisão Bibliográfica	7
2.1 Antenas: alguns tipos básicos	7
2.1.1. Antena isotrópica	7
2.1.2. Ganho de uma antena.....	8
2.1.3 Polaridade da radiação	9
2.1.4 Dipolo de meia onda.....	9
2.2. Transmissão de Dados Via Rádio.....	11
2.2.1. Espectro Magnético	11
2.2.2. Transmissão de Rádio.....	14
2.2.3. Transmissão de Microonda.....	15
2.2.4. Alguns termos e definições.....	16
2.3. Video.....	18
2.3.1. Video Analógico	18
2.3.2. Video Digital	19
2.4. Linguagens de Programação.....	19
2.4.1 Java	21
3. Análise de Requisitos	24
3.1 Requisitos de Hardware	24
3.2 Requisitos de Software	25
4. Projeto	26
4.1 Idéias Preliminares.....	26
4.1.1 Transmissão de Vídeo.....	27
4.1.2 Transmissão de Dados de Telemetria	30
4.1.3 Interface Gráfica	30
4.2 Soluções adotadas	32
4.2.1 Hardware do sistema.....	32
4.2.2 Comunicação Módulo Aéreo/Módulo Base	38
4.2.3 Interface Gráfica	47
4.2.4 Execução de processos em paralelo.....	65
5 Análises e Conclusão	67
5.1 Comunicação Módulo Aéreo e Módulo Base.....	67
5.2 Interface	68
Referências	69

1. INTRODUÇÃO

Esse projeto de formatura é parte integrante de um projeto maior que tem como objetivo final obter uma aeronave capaz de ser pilotado remotamente enviando informações a uma estação base em solo.

O projeto foi subdividido em áreas as quais todas estão relacionadas a um projeto de formatura dividido basicamente em 4 categorias: Controle (identificação de planta), Estruturas, Aerodinâmica e Transmissão de vídeo e Telemetria.

Dada a experiência de alguns integrantes desse projeto maior em eventos e competições de aeromodelos (Aerodesign – SAE) decidiu-se então realizar esse projeto relativo a UAVs (Unmanned Aerial Vehicles – Veículos aéreos não tripulados).

A respeito dos UAVs, muitos setores da economia mundial têm demonstrado grande interesse nesse novo mercado devido a sua grande versatilidade podendo, os UAV's, serem muito úteis tanto em aplicações militares quanto civis. Aproximadamente US\$ 2.1 bilhões serão investidos no mercado militar em 2005. Em aplicações civis, a atual tecnologia pode ser aplicada para monitoramento de costas e fronteiras, monitoramento preditivo, fotografia aérea entre muitas outras.

O presente projeto de formatura trata da parte de Transmissão de vídeo e Telemetria, tendo como objetivo fazer a comunicação entre a aeronave e a estação base, através de comunicação por rádio-freqüência. Nessa comunicação estão envolvidas as informações necessárias para pilotagem remota, como visualização de câmeras e recepção dos dados de sensores enviados pelo computador embarcado na aeronave. Por fim, é apresentada uma interface gráfica para representar esses dados.

O trabalho está organizado da seguinte forma: inicialmente é feita a revisão bibliográfica sobre antenas, ondas de rádio, vídeo (digital e analógico) e linguagens de programação. Em seguida, é feita a análise de requisitos de hardware e software. Segue-se a seção de projeto, apresentando as idéias preliminares e as soluções adotadas. Ao final são descritas as principais conclusões.

2. REVISÃO BIBLIOGRÁFICA

2.1 ANTENAS: ALGUNS TIPOS BÁSICOS

Antenas são dispositivos passivos, que transmitem ou recebem radiações eletromagnéticas. Neste tópico, algumas considerações básicas sobre este importante componente das telecomunicações.

2.1.1. Antena isotrópica

Uma antena isotrópica pode ser considerada como um elemento puntiforme, cuja potência irradiada (ou recebida) é a mesma em todas as direções (Π da Fig. (1)).

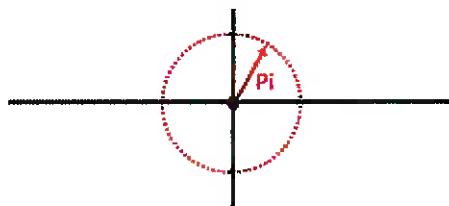


Figura 1. Potência de irradiação de uma antena isotrópica

Na prática ela não existe. É apenas um modelo matemático para comparação com antenas reais. Pode ser simulada de forma aproximada por uma combinação de dipolos de meia onda.

As antenas reais não são isotrópicas, isto é, a potência irradiada (ou recebida) varia de acordo com a orientação (considerando-se o aspecto tridimensional). Algumas antenas práticas irradiam de forma aproximadamente uniforme em um determinado plano.

2.1.2. Ganho de uma antena

O conceito de ganho de uma antena deve ser entendido de forma diferente do de um amplificador. Antenas são elementos passivos, não amplificam sinais. O ganho de uma antena expressa a relação com uma antena de referência. Veja exemplo a seguir.

A Fig. (2) dá a curva aproximada da potência irradiada por um dipolo de meia onda. Um vetor traçado do centro do dipolo até um ponto qualquer da curva representa a potência irradiada na direção do vetor. Assim, a potência máxima irradiada é dada pelo vetor P (ou o oposto de 180° , na outra parte da curva).



Figura 2. Potência de irradiação de um dipolo de meia onda

Considere agora uma antena isotrópica conforme item anterior, na mesma posição do dipolo e alimentada com a mesma potência da linha de transmissão. Ela irradia uma potência máxima P_i , que é a mesma para todas as direções. Então, o ganho do dipolo de meia onda tendo como referência a antena isotrópica é dado pela relação entre essas potências, expressa em

decibéis. Portanto, ganho = $10 \log (P/P_i)$. E o valor encontrado é simbolizado por dBi, para indicar a antena isotrópica como referência. Uma antena isotrópica tem ganho de 0 dBi.

2.1.3 Polaridade da radiação

O ângulo que a antena faz com o plano horizontal determina a orientação dos campos elétrico e magnéticos irradiados, os quais são perpendiculares entre si.

Para maior eficiência do conjunto transmissor e receptor, as antenas de ambos devem ter a mesma polarização.

A Fig. (3) dá uma representação gráfica.



Figura 3 - Polarização de uma antena

2.1.4 Dipolo de meia onda

É um tipo básico de antena, formado por dois condutores retilíneos, cada um de comprimento de $1/4$ do comprimento de onda da radiação a ser transmitida ou recebida (Fig. (4)).

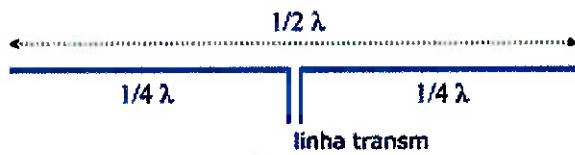


Figura 4 - Dimensões de um dipolo de meia onda

No vácuo (e de forma aproximada para o ar), a relação entre o comprimento de onda e a freqüência é dada por: $\lambda = 3 \times 10^8 / f$, onde λ é o comprimento de onda em metros e f , a freqüência em hertz. A constante 3×10^8 é a velocidade aproximada de propagação (velocidade da luz).

Desde que a velocidade de propagação nos meios condutores é menor, na prática os comprimentos das antenas são cerca de 95% dos calculados pela fórmula anterior.

Para dipolos de meia onda, a impedância na freqüência de ressonância é aproximadamente 72 ohms (lembra que impedância não significa necessariamente um resistor físico. Afinal os elementos são eletricamente separados. É uma característica que pode ser calculada). Um dipolo de meia onda apresenta um ganho de 2,14 dBi. Alguns fabricantes de antenas indicam o ganho tendo como referência o dipolo de meia onda. Assim, para efeito de comparação, é importante saber a referência, pois há uma diferença de 2,14 dB entre as duas.

2.2. TRANSMISSÃO DE DADOS VIA RÁDIO

2.2.1. Espectro Magnético

Quando se movem, os elétrons criam ondas eletromagnéticas que podem se propagar pelo espaço (mesmo no vácuo). Essas ondas foram descobertas pelo físico britânico James Clerk Maxwell, em 1865 e produzidas e observadas pelo físico alemão Heinrich Hertz, em 1887. O número de oscilações por segundo de uma onda eletromagnética é chamado freqüência (f), e é medido em Hz (em homenagem a Heinrich Hertz). A distância entre dois pontos máximos (ou mínimos) consecutivos na onda é chamada de comprimento de onda (λ).

Anexando uma antena de tamanho apropriado a um circuito elétrico, as ondas eletromagnéticas podem ser transmitidas e recebidas com eficiência à longa distância. Todas as comunicações sem fio são baseadas nesse princípio (Tanenbaum, 1996).

No vácuo, todas as ondas eletromagnéticas se propagam na mesma velocidade, independentemente da sua freqüência. Essa velocidade, geralmente chamada de velocidade da luz, c , é aproximadamente 3×10^8 m/seg, aproximadamente 1 ft. (30 cm.) por nanosegundo. No cobre ou na fibra óptica, essa velocidade decresce aproximadamente para 2/3 desse valor e se torna ligeiramente dependente da freqüência (Yacoub, 1993).

A relação fundamental entre f , λ e c (no vácuo) é:

$$\lambda f = c \quad (1)$$

Como c é uma constante, sabendo-se f , pode-se encontrar λ e vice versa. Por exemplo, ondas de 1 MHz têm aproximadamente 300 metros e ondas de 1 cm têm uma freqüência de 30 GHz.

As ondas de rádio, microondas, infravermelho e porções de luz visíveis do espectro podem ser utilizadas para a transmissão de informações através da modulação da amplitude, freqüência ou fase das ondas. A luz ultravioleta, o raio-X e os raios gama deveriam ser melhores devido às suas altas freqüências; mas eles são difíceis de produzir, modular, não se propagam bem através de construções e são nocivos para os seres vivos (Tanenbaum, 1996). As bandas LF, MF, HF, VHF, UHF, SHF, EHF e THF são nomes oficiais do ITU (International Telecommunication Union). Esses nomes são baseados no comprimento de onda. Os termos LF, MF e HF se referem a Low, Medium e High Frequency, respectivamente. Quando esses nomes foram associados, ninguém esperava superar freqüências de 10 MHz; por isso, as bandas superiores posteriormente foram chamadas de Very, Ultra, Super, Extremely e Tremendously High Frequency.

O volume de informação que uma onda eletromagnética pode transportar está diretamente relacionado com a sua largura de banda. Utilizando tecnologias atuais, é possível codificar alguns bits por Hertz em baixas freqüências; no entanto, esse número pode subir para 40 sob determinadas condições em altas freqüências. Um cabo com 500 MHz de largura de banda pode transportar muitos gigabits por segundo (Tanenbaum, 1996).

Se a Eq. (1) for resolvida para f e derivada em relação a λ é obtido:

$$\frac{df}{d\lambda} = -\frac{c}{\lambda^2} \quad (2)$$

Se for aplicado o método de diferenças finitas ao invés da derivação, e somente forem observados os valores absolutos, obtém-se:

$$\Delta f = \frac{c\Delta\lambda}{\lambda^2} \quad (3)$$

Com base na largura de uma banda de comprimento de onda $\Delta\lambda$, é possível calcular a banda de freqüência correspondente, Δf , e, a partir daí, a taxa de dados que a banda pode transportar. Quanto mais larga a banda, mais alta a taxa de dados. Como um exemplo: se for considerada uma banda de 1,30 microns tem-se: $\lambda = 1,3 \times 10^{-6}$ e $\Delta\lambda = 0,17 \times 10^{-6}$, resultando em um Δf de 30 THz.

Foram criadas organizações nacionais e internacionais para regulementar o uso do espectro eletromagnético. Nos Estados Unidos, o FCC (Federal Communications Commission) aloca espectro para rádios AM e FM, televisão, telefonia celular, operadoras de telefonia, polícia, militares, navegação, governo e muitos outros usuários. No Brasil, essa responsabilidade é delegada à ANATEL (Agência Nacional de Telecomunicações). Em âmbito mundial, uma agência do ITU-R (International Telecommunication Union – Radiocommunications Conference), antiga WARC (World Administrative Radio Conference). Muitas vezes, as agências nacionais não seguem as recomendações da WRC. Conseqüentemente, dispositivos que utilizam rádio freqüência podem funcionar apenas em um certo país ou região.

2.2.2. Transmissão de Rádio

As ondas de rádio são relativamente fáceis de gerar, percorrem longas distâncias e atravessam construções com certa facilidade. Devida a essas características, elas são largamente utilizadas para comunicação, tanto em interiores como em espaço aberto. As ondas de rádio também são omnidirecionais, o que significa que elas podem trafegar em todas as direções a partir da sua origem. O transmissor e o receptor não precisam estar cuidadosamente alinhados.

As propriedades das ondas de rádios dependem da freqüência. Em baixas freqüências, as ondas de rádio atravessam bem os obstáculos, mas a potência cai rapidamente com o aumento da distância entre o transmissor e o receptor. Em altas freqüências, as ondas de rádio tendem a percorrer linhas retas, mas são refletidas por obstáculos. As ondas de rádio também são absorvidas pela chuva. Em todas as freqüências, as ondas de rádio estão sujeitas à interferências de motores e outros equipamentos elétricos.

Devido à capacidade que as ondas de rádio têm de percorrer longas distâncias, a interferência entre os usuários é um problema. Por essa razão, todos os governos exercem um controle sobre os transmissores de rádio, concedendo apenas algumas exceções.

Em bandas VLF, LF e MF, as ondas de rádio seguem a superfície terrestre. Em baixas freqüências, essas ondas podem ser detectadas a até 1000 km de distância; em altas freqüências, esse raio de ação é bem menor. Ondas de rádio que operam nessas bandas facilmente atravessam construções. O principal problema na utilização dessas freqüências em comunicações de dados é a baixa largura de banda que elas oferecem (ver equação 1).

Nas bandas HF e VHF, as ondas tendem a ser absorvidas pela terra. Entretanto, as ondas que chegam à ionosfera, uma camada de partículas carregadas que circunda a terra a uma altitude

de 100 a 500 km, são refratadas por ela e enviadas de volta para a terra. Em determinadas condições atmosféricas, os sinais podem ser refletidos várias vezes. Operadores de rádio amador utilizam essas bandas para comunicação em longas distâncias.

2.2.3. Transmissão de Microondas

Acima de 100 MHz, as ondas trafegam em linha reta e, por essa razão, podem ser captadas com mais facilidade. Ao contrário das ondas de rádio de freqüências mais baixas, as microondas não atravessam os prédios. Algumas ondas podem ser refratadas nas camadas atmosféricas mais baixas e, consequentemente, a sua chegada pode ser mais demorada do que as ondas diretas. Esse efeito é chamado de multipath fading, ou fadding por múltiplos caminhos, e costuma provocar sérios problemas. Ele depende do tempo e da freqüência (Tanenbaum, 1996).

Atualmente é muito comum a utilização de bandas até 10 GHz. A partir de 8 GHz surge um novo problema: a absorção pela água.

Para aumentar a flexibilidade e diminuir a burocracia na alocação do espectro eletromagnético, as organizações responsáveis pela regulamentação do seu uso criaram bandas (faixas de freqüência) de uso livre, ou seja, não é necessário obter nenhum tipo de licença para utilizar essas bandas. Elas são chamadas de bandas ISM (Industrial, Scientific and Medical). Além dela, existem outras bandas ISM como a de 902 a 928 MHz e a de 5725 a 5850 MHz; contudo elas podem ser utilizadas somente em alguns países, dentre eles o Brasil. (ANATEL, 2001).

Logicamente, para prevenir abusos, essas organizações impuseram um conjunto de regras para a utilização dessas bandas e somente produtos certificados são permitidos. Essas regras especificam a máxima potência transmitida na banda e fora dela (para não poluir as faixas de freqüência adjacentes).

2.2.4. Alguns termos e definições

Esta parte visa explicar alguns termos relacionados a antenas e rádio freqüência usados quando lidamos com uma instalação de um sistema de rádio freqüência.

- **dB**

Abreviação para decibels. Expressa a relação entre dois valores.

- **Potência RF (RF Power Level)**

É a potência transmitida através de um transmissor RF ou a potência recebida em um receptor. Normalmente expressado em Watts(W) ou dBm. A relação entre Watts e dBm é expressada a seguir:

$$P_{dBm} = 10 \cdot \log P_{mW} \quad (4)$$

- **Atenuação**

A atenuação de um sinal é definida como:

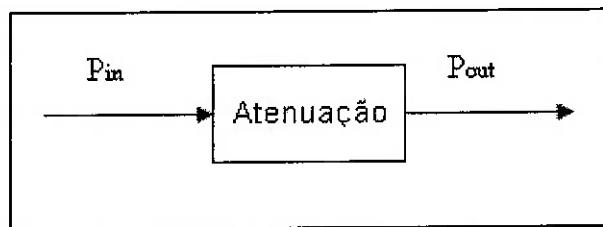


Figura 5 - Potência de irradiação de um dipolo de meia onda

Onde:

P_{in} é a potência RF de entrada

P_{out} é a potência RF de saída

A atenuação é expressa em dB como:

$$PdB = 10 \cdot \log\left(\frac{P_{in}}{P_{out}}\right) \quad (5)$$

- **Perdas durante a trajetória (Path Loss)**

É a atenuação da onda eletromagnética quando se propaga pelo espaço. É também expresso em dB e normalmente depende de:

- Distância entre as antenas de transmissão e recepção
- Caminho estar livre (visada direta) entre as antenas de transmissão e recepção
- Tamanho da antena

É calculada utilizando-se a seguinte fórmula

$$Pl = 32,4 + 20 \cdot \log F + 20 \cdot \log R \quad (6)$$

onde

F é a freqüência do sinal em MHz

R é a distância entre as antenas em km

- **Sensibilidade de um transmissor (*Transmitter Sensitivity*)**

É a potência mínima requerida do sinal de entrada para que possua uma determinada performance

2.3. VIDEO

2.3.1. Video Analógico

O ser humano percebe imagens de uma forma paralela, isto é, todos os objetos da imagem refletem luz ao mesmo tempo e o olho humano capta tais reflexões paralelamente. O nervo óptico possui milhões de conexões que levam a informação, paralelamente, até o cérebro. No mundo eletrônico, é impraticável construir uma conexão com milhares de circuitos; por isso, é utilizada

uma única conexão ou algumas poucas conexões e um sensor de vídeo (câmera) que converte a imagem paralela em um sinal de vídeo (Goularte & Moreira, 1998).

2.3.2. Video Digital

O vídeo digital tem crescido muito nos últimos anos por uma série de razões. A informação digital é mais robusta e pode ser codificada para eliminar erros. Isso significa que perdas na gravação e na transmissão podem ser eliminadas. Outra razão desse crescimento está relacionada com a manipulação de vídeo. As técnicas de gravação e transmissão digital permitem um nível de manipulação de conteúdo impossível de ser feito utilizando as técnicas analógicas (Tektronix, 2001).

Antes de se obter um vídeo digital, é preciso transformá-lo de sua representação analógica para uma representação digital. Esse processo é conhecido como digitalização. O processo de digitalização envolve a conversão do sinal de vídeo analógico em um fluxo de vídeo (*video stream*) digital, utilizando circuitos para captura de vídeo.

2.4. LINGUAGENS DE PROGRAMAÇÃO

Uma linguagem de programação é um método padronizado para expressar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um

programa de computador. Uma linguagem permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias. O conjunto de palavras (tokens), compostos de acordo com essas regras, constituem o código fonte e um software. Esse código fonte é depois traduzido para código de máquina, que é executado pelo processador.

Uma das principais metas das linguagens de programação é permitir que programadores tenham uma maior produtividade, permitindo expressar suas intenções mais facilmente do que quando comparado com a linguagem que um computador entende nativamente (código de máquina). Assim, linguagens de programação são projetadas para adotar uma sintaxe de nível mais alto, que pode ser mais facilmente entendida por programadores humanos. Linguagens de programação são ferramentas importantes para que programadores e engenheiros de software possam escrever programas mais organizados e rapidamente.

Linguagens de programação também tornam os programas menos dependentes de computadores ou ambientes computacionais específicos (propriedade chamada de portabilidade). Isto acontece porque programas escritos em linguagens de programação são traduzidos para o código de máquina do computador no qual será executado em vez de ser diretamente executado. Uma meta ambiciosa do Fortran, uma das primeiras linguagens de programação, era esta independência da máquina onde seria executada. Uma linguagem de programação pode ser convertida em código de máquina por compilação ou interpretação. Se o método utilizado traduz todo o texto do programa (também chamado de código), para só depois executar (ou rodar, como se diz no jargão da computação) o programa, então se diz que o programa foi compilado e que o mecanismo utilizado para a tradução é um compilador (que por sua vez nada mais é do que um programa). A versão compilada do programa tipicamente é armazenada, de forma que o

programa pode ser executado um número indefinido de vezes sem que seja necessária nova compilação, o que compensa o tempo gasto na compilação. Isso acontece com linguagens como Pascal e C.

Se o texto do programa é traduzido à medida que vai sendo executado, como em Javascript, Python ou Perl, num processo de tradução de trechos seguidos de sua execução imediata, então se diz que o programa foi interpretado e que o mecanismo utilizado para a tradução é um interpretador. Programas interpretados são geralmente mais lentos do que os compilados, mas são também geralmente mais flexíveis, já que podem interagir com o ambiente mais facilmente (freqüentemente linguagens interpretadas são também de script).

Embora haja essa distinção entre linguagens interpretadas e compiladas, as coisas nem sempre são tão simples. Há linguagens compiladas para um código de máquina de uma máquina virtual (sendo esta *máquina virtual* apenas mais um software, que emula a máquina virtual sendo executado em uma máquina real), como o Java e o Parrot. E também há outras formas de interpretar em que os códigos fontes, ao invés de serem interpretados linha-a-linha, têm blocos "compilados" para a memória, de acordo com as necessidades, o que aumenta a performance dos programas quando os mesmos módulos são chamados várias vezes, técnica esta conhecida como Just in Time.

2.4.1 Java

Java é uma linguagem de programação orientada a objetos projetada e implementada por um pequeno grupo de pessoas coordenado por James Gosling na empresa Sun Microsystems em

1991. Criada inicialmente com a finalidade de ser utilizada em eletrodomésticos, se tornou popular pelo seu uso na internet.

A linguagem Java foi projetada tendo em vista os seguintes objetivos:

- Orientação a objetos - Baseado no modelo de Smalltalk e Simula67.
- Portabilidade - Independência de plataforma.
- Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP.
- Segurança - Pode executar programas via rede com restrições de execução.

Além disso, podem-se destacar outras vantagens apresentadas pela linguagem:

- Sintaxe similar a Linguagem C.
- Facilidades de Internacionalização - Suporta nativamente caracteres Unicode.
- Simplicidade na especificação, tanto da linguagem como do ambiente de execução (JVM).
 - É distribuída com um vasto conjunto de bibliotecas (ou APIs).
 - Possui facilidades para criação de programas distribuídos e *multi-thread* (múltiplas linhas de execução num mesmo programa).
 - Desalocação de memória automática por processo de garbage collector.
 - Carga Dinâmica de Código - Programas em Java são formados por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização.

Programas em Java não são traduzidos para "linguagem de máquina" como outras linguagens compiladas e sim para uma representação intermediária, chamada de *bytecodes*, que por sua vez são interpretados pelo processador. Os críticos da linguagem afirmam que por causa desse processo, o código *interpretado* Java tem baixo desempenho. Porém, sucessivos avanços

tornaram a otimização da máquina virtual Java (JVM) bastante eficiente, diminuindo a diferença de performance entre Java e as linguagens compiladas tradicionais.

Outro "defeito" apontado pelos críticos da linguagem é de que os *bytecodes* produzidos pelos compiladores Java podem ser facilmente usados num processo de "engenharia reversa" para a recuperação do programa-fonte original, mas esta é uma característica que atinge em menor ou maior grau todas as linguagens compiladas.

Para ser um desenvolvedor da tecnologia Java, é necessário ter conhecimentos sobre o conceito de Orientação a Objeto. A partir daí, é necessário baixar da internet o JDK (Java Development Kit), um Kit de Desenvolvimento do Java, que contém uma coleção com as principais classes de Java e um compilador que transforma sua programação em *bytecodes*. Também existem IDEs de desenvolvimento em Java, como o NetBeans (SUN) e o Eclipse (Patrocinado pela IBM). Nessas IDEs, o desenvolvedor já pode definir a interface com o usuário (parecido com o VB). Para desenvolvedores em UML, as dicas são os modeladores Umbrello, DIA e principalmente o Jude.

3. ANÁLISE DE REQUISITOS

3.1 REQUISITOS DE HARDWARE

- Haverá dois sistemas de transmissão de dados, um relativo a transmissão de vídeo e outro relativo a transmissão de dados de telemetria.
- Existirão dois módulos que aqui serão chamados de Modulo da Base (relativo aos equipamentos que estarão na estação da base) e o Módulo Aéreo (relativo aos equipamentos embarcados na aeronave)
- A transmissão de dados (tanto de vídeo quanto de dados de telemetria) deve ser projetado para ter alcance de 7,5 km.
- Os equipamentos a bordo da aeronave devem ter baixo consumo de energia
- Estarão disponíveis quatro câmeras de vídeo a bordo do avião para possibilitar a visualização de partes importantes da aeronave para o piloto.
- Deverá haver a possibilidade de alterar as câmeras a serem visualizadas através de um “switch” ou a utilização de mais de quatro canais na transmissão dos vídeos através das câmeras.

3.2 REQUISITOS DE SOFTWARE

- Todos os sensores devem ser continuamente monitorados pelo MA para a detecção de falhas. As falhas detectadas pelo MA devem ser transmitidas para o MB;
- Os dados de telemetria e vídeo digital devem ser continuamente transmitidos pelo MA;
- Os dados de telecomando devem ser continuamente transmitidos pelo MB;
- Como haverá quatro câmeras de vídeo, o operador poderá selecionar a câmera desejada;
- O sistema deve apresentar, ao operador, uma interface gráfica intuitiva semelhante ao painel de um avião. No painel devem ser mostrados os instrumentos necessários para a realização de vôo por instrumentos.

4. PROJETO

Como já citado anteriormente, o projeto foi dividido em dois módulos (Módulo Aéreo e Módulo Base). O Módulo Base conta com os equipamento da estação em solo e utiliza um microcomputador utilizando como sistema operacional Windows XP da Microsoft e no Módulo Aéreo um computador de bordo (PC-104) utilizando o sistema operacional QNX.

Este capítulo apresenta detalhes do projeto do sistema. Na seção 4.1 são apresentadas algumas idéias iniciais do projeto, na seção 4.2 as soluções escolhidas, o projeto de hardware do sistema, bem como detalhes sobre o software a ser implementado e uma seção a respeito do protocolo de comunicação entre os módulos do projeto.

4.1 IDÉIAS PRELIMINARES

Dentro deste tópico será abordado inicialmente algumas das soluções possíveis propostas pelos integrantes do grupo referente a este trabalho de formatura, relacionados à transmissão de vídeo, transmissão de dados de telemetria e apresentação da interface gráfica ao usuário.

4.1.1 Transmissão de Vídeo

Foi feito um levantamento através de pesquisa e chegou-se a duas soluções possíveis para a transmissão de vídeo, poderia ela ser feita de maneira analógica ou digital. A escolha das imagens provenientes de cada uma das quatro câmeras posicionadas na aeronave será feita através de um chaveamento.

4.1.1.1 Transmissão Digital

Uma idéia geral é dada na Fig (14) abaixo a respeito da transmissão digital:

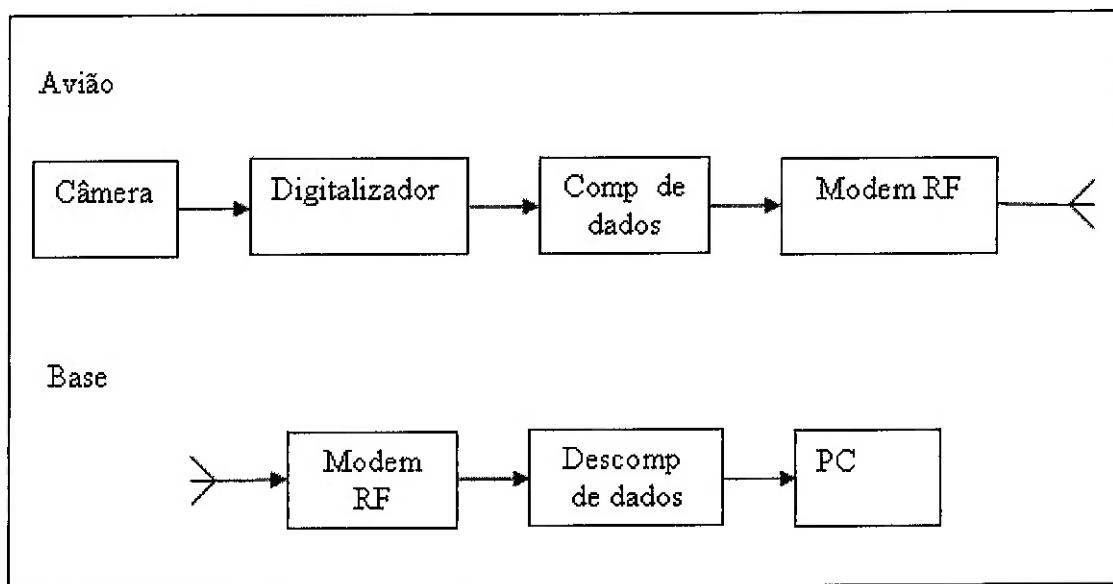


Figura 6 - Sistema de transmissão de vídeo digital

O sistema baseia-se na utilização de um digitalizador que converte o sinal analógico da câmera instalada na aeronave em sinais digitais, após isso há a compressão de dados para possibilitar a transmissão de dados pelo modem de rádio freqüência, na base há outro modem que recebe os dados ocorre então o processo de descompressão de dados que é enviado e processado pelo PC.

Esse processo tem como vantagens ser um sistema bastante modular o que permite a detecção de problemas em camadas, no entanto mostra-se um processo bastante complexo possuindo etapas de compressão e descompressão de dados que quando realizados podem ocorrer atrasos significativos da ordem de segundos, conforme explicado pelo Prof. Guido Stolfi do Departamento de Engenharia de Telecomunicações da Escola Politécnica, em conversa referente ao assunto do trabalho de formatura.

4.1.1.2. Transmissão Analógica

É apresentada na figura abaixo uma outra possível implementação para o sistema de vídeo analógico.

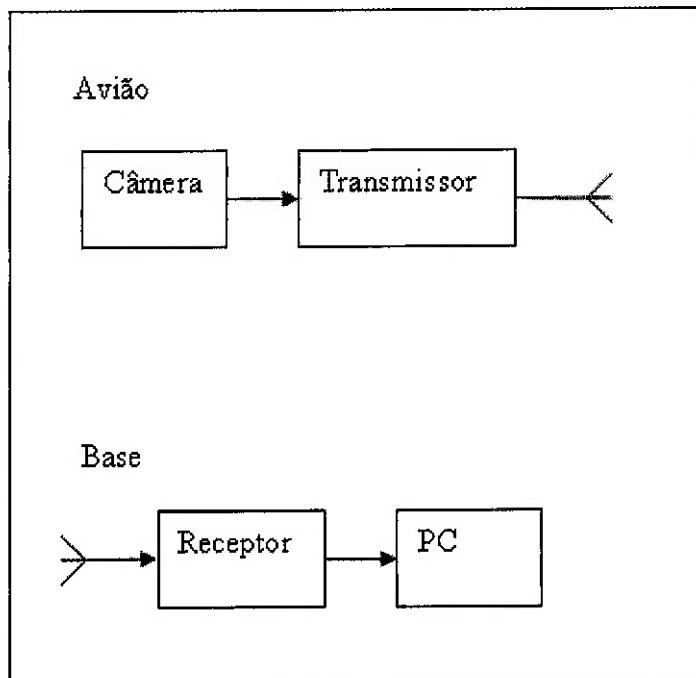


Figura 7 - Sistema de transmissão de vídeo analógico

Existem no mercado soluções já prontas para a transmissão de vídeo analógico principalmente quando falamos de sistemas de vigilância que utilizam transmissões sem-fio. Esse tipo de transmissão tem como vantagens o custo monetário de implementação pois requer menos componentes e trata-se de uma solução mais simples do ponto de vista implementação sendo necessário somente os cálculos e a escolha de antenas que tornem possível a transmissão de vídeo e ainda mais importante é o fato de possuir um atraso pouco significativo em relação ao tempo de reação humano quando no controle da aeronave. No entanto possui dois sistemas sobre os quais não se possui um grande conhecimento que são o Transmissor (também conhecido como Tx) e o Receptor (conhecido como Rx).

4.1.2 Transmissão de Dados de Telemetria

A transmissão de dados de telemetria será feita utilizando-se dois modems de rádio freqüência, sendo um localizado no módulo aéreo e outro localizado no módulo base. Será definido um protocolo de transmissão e recepção de dados para efetuar a comunicação correta entre os dois módulos. A implementação será feita através de uma simulação com dois computadores enviando e recebendo sinal através de modems, pois como já mencionado anteriormente os dados referentes aos sensores não faz parte do escopo desse projeto.

4.1.3 Interface Gráfica

Para o interfaceamento com o usuário será utilizado como base o software Flight Simulator da Microsoft, no entanto ao invés de simularmos graficamente a visão do piloto teremos as imagens de cada uma das câmeras instaladas na aeronave.

A imagem abaixo Fig. (8) ilustra a forma com a qual o projeto final deverá se parecer. As imagens capturadas pelas câmeras estarão na parte superior da tela, e a leitura dos sensores do UAV (telemetria) estará na parte inferior, representada pelos relógios.



Figura 8 - Tela do Flight Simulator

A escolha da linguagem para a produção do software teve de ser baseada não só nos requisitos de software anteriormente apresentados, mas também na experiência dos integrantes do grupo com a linguagem de programação.

4.2 SOLUÇÕES ADOTADAS

4.2.1 Hardware do sistema

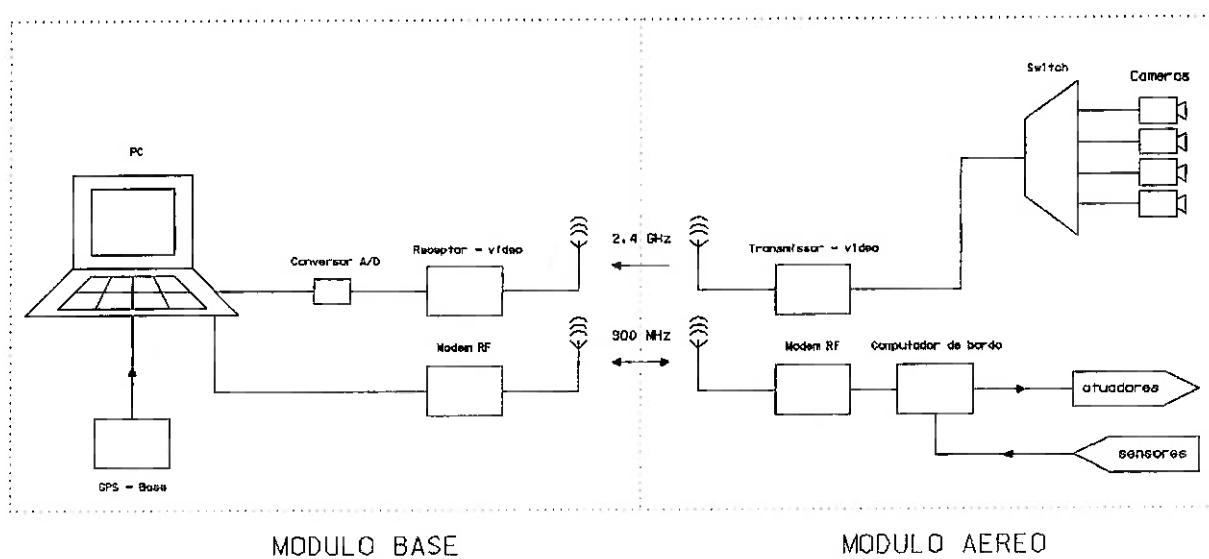


Figura 9 – Componentes de Hardware

A fig. 17 apresenta um diagrama descrevendo os componentes de hardware utilizados no projeto. A seguir, será apresentada uma breve descrição desses componentes.

4.2.1.1 Componentes de Hardware

1. Câmeras de Vídeo

Serão instaladas 4 câmeras de vídeo na aeronave. Duas câmeras, apontando cada uma para uma asa para possibilitar a visualização do piloto dos comandos de aileron, uma será apontada para o bico da aeronave possibilitando a visualização da frente do avião e outra apontando para a parte traseira da aeronave, possibilitando a visualização dos comandos de leme e profundor da aeronave.

2. Atuadores

Através dos sinais emitidos pelo usuário do microcomputador e recebidos no computador de bordo, este será enviado para a ativação ou desativação de dispositivos presentes na aeronave. Durante esta etapa de projeto somente será comandado remotamente pelo módulo da base o dispositivo de emergência do pára-quedas da aeronave. Outros dispositivos de atuação (servomotores responsáveis pela movimentação de leme, ailerons, profundor, motor) serão comandados através de rádio-controles normalmente utilizados em aeronaves comuns.

3. Sensores

Deverão estar presentes na aeronave os seguintes sensores

- Bússola
- GPS
- Unidade de Medição Inercial (UMI)

- Rotação do motor
- Sensor Data Boom
 - Umidade
 - Temperatura
 - Pressão
 - Velocidade
 - Ângulo de *ataque*
 - Ângulo de *sideslip*
- Ultra-som
- Nível de combustível
- Consumo de combustível
- Nível de bateria
- Funcionamento do alternador

4. Switch

É uma chave eletrônica de vídeo e áudio. É composta pelas entradas de áudio e vídeo das 4 câmeras presentes na aeronave e possui uma saída de áudio e vídeo, a seleção de câmeras será feita através dos canais presentes no rádio que controla o avião.

5. Computador de Bordo

Foi selecionado um PC-104, devido as restrições de peso e consumo de energia. O modelo selecionado possui 4Mbytes de memória, 2 MBytes de memória de estado sólido, 4 portas seriais e 1 porta paralela. Na implementação do sistema de monitoramento será

responsável por receber os dados dos sensores e enviá-los ao modem para que sejam recebidos os dados pelo microcomputador e também por enviar o sinal recebido do microcomputador da seleção de câmeras e do acionamento de dispositivos de atuação.

6. Transmissor/Receptor de Vídeo

O fator preponderante para a escolha do tipo de transmissão de vídeo foi o tempo de atraso da transmissão, conforme já citado anteriormente a transmissão de dados digital pode fornecer atrasos da ordem de segundos, o que se tornaria inviável dado que a função da interface é fornecer um meio de visualização para a possibilitar a condução da aeronave pelo piloto, impossibilitando com um atraso dessa ordem a dirigibilidade da aeronave. Utilizaremos a freqüência de 2.4 GHz para a transmissão de vídeo. Seguem os dados e cálculos relativos aos equipamentos necessários para a escolha correta de transmissor e receptor.

Descrição	Símbolo	Fórmula	Valor
Distância máxima entre os pontos (km)	R		7,5
Freqüência (MHz)	F		2400
Potência Transmissor (mW)	Pt		1000
Ganho da antena Tx (dBi)	Gt		3
Ganho da antena Rx (dBi)	Gr		3
Comprimento do cabo (Tx)	Lt		0
Comprimento do cabo (Rx)	Lr		5
Atenuação do cabo (dB/m)	C		0,6
Path Loss	Pl	$32,4 + 20 \log(F) + 20 \log(R)$	118
Atenuação Cabo Tx	Ct	$C * Lt$	0
Atenuação Cabo Rx	Cr	$C * Lr$	3
Potencia Transmissor (dBm)	P	$10 \log(Pt)$	30
Potência recebida no Receptor (dBm)	Si	$P - Ct + Gt - Pl + Gr - Cr$	-85
Sensibilidade Receptor (dBm)	Ps		-92

Podemos observar que a potência na entrada do receptor é superior a potência mínima na entrada do mesmo, assim sendo possível transmissão com um equipamento que possua as características acima.

7. Modems de Rádio Freqüência

É responsável pela transmissão de dados entre os módulos aéreo e base. É necessário que a transmissão ocorra nos dois sentidos, tanto do MA para o MB como do MB para o MA, portanto deverá operar em modo *full duplex* sendo possível transmitir e receber dados simultaneamente. Será utilizada a freqüência de 900 MHz para a transmissão de dados. Seguem os dados cálculos relativos aos equipamentos necessários para a escolha correta dos modems.

Descrição	Símbolo	Fórmula	Valor
Distância máxima entre os pontos (km)	R		7,5
Freqüência (MHz)	F		900
Potência Transmissor (mW)	Pt		1000
Ganho da antena Tx (dBi)	Gt		2,1
Ganho da antena Rx (dBi)	Gr		2,1
Comprimento do cabo (Tx)	Lt		0
Comprimento do cabo (Rx)	Lr		5
Atenuação do cabo (dB/m)	C		0,6
Path Loss	Pl	$32,4 + 20 * \log(F) + 20 * \log(R)$	109
Atenuação Cabo Tx	Ct	$C * Lt$	0
Atenuação Cabo Rx	Cr	$C * Lr$	3
Potencia Transmissor (dBm)	P	$10 * \log(Pt)$	30
Potência recebida no Receptor (dBm)	Si	$P - Ct + Gt - Pl + Gr - Cr$	-78
Sensibilidade Receptor (dBm)	Ps		-100

Podemos observar que a potência na entrada do receptor é superior a potência mínima na entrada do mesmo, assim sendo possível transmissão com um equipamento que possua as características acima.

8. Conversor A/D

Trata-se de um periférico que converte o sinal de vídeo analógico em digital para que o mesmo possa ser exibido pelo microcomputador. Possui uma entrada do tipo RCA de vídeo analógico e uma saída USB assim tendo o seu funcionamento similar ao de uma Web Cam para o microcomputador, sendo somente necessária a instalação dos drivers do equipamento.

9. PC

Microcomputador que será utilizado no módulo base, é responsável pelo interfaceamento com usuário (como será explicado na seção 4.2.2), bem como a transmissão de dados de periféricos do módulo base.

10. Diferencial GPS – base

Fornece dados relativos a um ponto fixo na Terra, no caso do módulo base. Esses dados são enviados ao microcomputador através da entrada serial do PC e reenviados ao computador de bordo para que sejam efetuados os cálculos necessários do GPS embarcado.

4.2.2 Comunicação Módulo Aéreo/Módulo Base

Esse tópico trata da transmissão de dados entre a aeronave e a estação de solo. A comunicação entre o módulo aéreo e o módulo base é caracterizada por ser uma comunicação ponto-a-ponto. Os dados transmitidos pelo canal de comunicação entre a estação de solo e o UAV são divididos em dois grupos de dados de acordo com o sentido do fluxo: os dados provenientes do MA e que serão transmitidos para o MB que chamaremos de dados de telemetria e os dados provenientes do MB e serão transmitidos para o MA que chamaremos de dados de telecomando. Também tratará do protocolo e da semântica de comunicação que será utilizada e qual a solução adotada para a comunicação entre os módulos.

4.2.2.1 Dados de telemetria

Para que o usuário possa pilotar a aeronave através do sistema a ser implementado, necessita-se de instrumentos que indiquem o estado da aeronave (telemetria), como por exemplo a direção e a altitude da aeronave e esses dados de telemetria serão gerados pelos sensores presentes na aeronave. Os pacotes de dados de telemetria serão transmitidos do MA para o MB em um dado intervalo de tempo.

Seguem quais serão os dados transmitidos do MA para o MB:

- Rotação do motor

- Umidade
- Temperatura
- Pressão
- Velocidade
- Ângulo de Ataque
- Ângulo de Sideslip
- Consumo de combustível
- Nível de combustível
- Funcionamento do Alternador
- Dados relativos a bussola
- Dados relativos ao GPS
- Dados relativos a UMI

4.2.2.2 Dados de telecomando

Os dados de telecomando são provenientes da estação de solo e são responsáveis pelo controle remoto de alguns dispositivos atuadores da aeronave, ou seja, intervenções do operador por meio de um dispositivo conectado ao microcomputador da estação de solo (mouse, teclado), irão gerar dados que atuarão na aeronave.

Seguem quais serão os dados transmitidos do MB para o MA:

- Atuação

- Comando de câmeras
- GPS diferencial

4.2.2.3 Protocolo de comunicação

Devido à utilização de modems que utilizam a rede ethernet (IEEE 802.3) foi escolhido o protocolo de comunicação TCP/IP para comunicação entre os módulos aéreo e base. Oficialmente esta família de protocolos é chamada, Protocolo Internet TCP/IP, comumente referenciada só como TCP/IP, devido a seus dois protocolos mais importantes (TCP: Transport Control Protocol e IP: Internet Protocol)

- O protocolo IP

O protocolo IP define mecanismos de expedição de pacotes sem conexão.

IP define três pontos importantes:

1. A unidade básica de dados a ser transferida na Internet.
2. O software de IP executa a função de roteamento, escolhendo um caminho sobre o qual os dados serão enviados.
3. Incluir um conjunto de regras que envolvem a idéia da expedição de pacotes não confiáveis. Estas regras indicam como os hosts ou gateways poderiam processar os

pacotes; como e quando as mensagens de erros poderiam ser geradas; e as condições em que os pacotes podem ser descartados.

- O protocolo TCP

TCP é um protocolo da camada de transporte. Este é um protocolo orientado a conexão, o que indica que neste nível vão ser solucionados todos os problemas de erros que não forem solucionados no nível IP, dado que este último é um protocolo sem conexão. Alguns dos problemas com os que TCP deve tratar são:

- pacotes perdidos ou destruídos por erros de transmissão .
- expedição de pacotes fora de ordem ou duplicados.

O TCP especifica o formato dos pacotes de dados e de reconhecimentos que dois computadores trocam para realizar uma transferência confiável, assim como os procedimentos que os computadores usam para assegurar que os dados cheguem corretamente. Entre estes procedimentos estão:

- Distinguir entre múltiplos destinos numa máquina determinada.
- Fazer recuperação de erros, tais como pacotes perdidos ou duplicados.

4.2.2.4 Comunicação por Sockets

Um socket é definido como um ponto terminal para a comunicação. Um par de processos que se comunicam através de uma rede emprega um par de sockets – um para cada processo. Ele é identificado por um endereço IP concatenado com um número de porta.

Os sockets são a base da comunicação em redes TCP/IP e também são muito usados em comunicações entre processos no interior de um mesmo computador. Um socket é uma interface de comunicação bidirecional entre processos. Sockets são representados como descritores de arquivos e permitem a comunicação entre processos distintos na mesma máquina ou em máquinas distintas, através de uma rede.

Serão então utilizados sockets para o estabelecimento de comunicação entre os módulos aéreo e base. No entanto para a comunicação deve-se definir um cliente e um servidor que trocarão informações através do socket aberto.

Algumas características importantes do modelo Cliente/Servidor:

- O cliente conhece o endereço e forma de acesso ao servidor e toma a iniciativa da comunicação
- O servidor é uma entidade passiva, apenas recebendo pedidos dos cliente e respondendo aos mesmos.
- O servidor oferece um serviço específico a seus clientes
- O cliente envia uma requisição de serviço e aguarda uma resposta do servidor.
- As implementações do cliente e do servidor são independentes e autônomas; apenas as seqüências de mensagens trocadas durante a comunicação, que caracterizam o serviço, devem ser respeitadas.

Com base nisso, utilizaremos o PC-104 como servidor, já que nele estarão contidas as informações relativas aos sensores presentes na aeronave e o PC na base como cliente recebendo os dados enviados do módulo aéreo.

4.2.2.5 Comunicação Serial

Também tornou-se necessária a leitura de dados do GPS diferencial presente no módulo base, no entanto o mesmo possui uma interface serial. Foi elaborado então um programa para leitura de porta serial do PC recebendo informações do GPS de acordo com a sua taxa de envio de dados, os dados recebidos pela interface serial são recebidos pelo cliente e enviados ao servidor para que o mesmo efetue os cálculos necessários.

4.2.2.6 Semântica de comunicação

Para que as mensagens sejam entendidas tanto pelo computador presente no módulo base quanto no módulo aéreo foi definida uma semântica de comunicação, determinando assim uma linguagem única dos dados emitidos e recebidos pela base.

A estrutura básica seqüencial de cada mensagem enviada/recebida é a seguinte:

- \$: caractere de início de mensagem (cabeçalho)
- Caractere identificador do observador (controle)

- Caractere identificador da informação trazida pelo sensor (dados A)
- Valor da informação do observador (dados B)
- Caracteres ASCII [CR][LF] indicadores de fim de mensagem (cabeçalho de fim)

A sintática fica então:

\$	Identificador do observador	Identificador do sensor	Dado	[CR][LF]
----	-----------------------------	-------------------------	------	----------

Na mesma mensagem, repetem-se as seções 3 e 4 de acordo com a necessidade de cada observador (quantidade de variáveis).

Seguem os códigos para cada informação relativa aos observadores:

Sistema Energético (Energy)	
Identificação do Observador	E
Informação do Sensor	Descrição
A	tensão do alternador
I	corrente do alternador
B	tensão da bateria 12V
J	corrente da bateria 12V
C	tensão da bateria 5V
K	corrente da bateria 5V

Sistema de Propulsão (Thrust)	
Identificação do Observador	T
Informação do Sensor	Descrição
τ	temperatura interna
r	rotação do motor
m	nível de combustível principal
i	nível de combustível intermediário
F	consumo de combustível

Sensor Air Data Boom	
Identificação do Observador	A
Informação do Sensor	Descrição
u	velocidade do ar
β	ângulo de sideslip
α	ângulo de ataque
τ	temperatura do ar
h	umidade do ar
b	barômetro

Bússola (Compass)	
Identificação do Observador	C
Informação do Sensor	Descrição
ϕ	ângulo de yaw
θ	ângulo de pitch
Φ	ângulo de roll
X	componente x do campo magnético
Y	componente y do campo magnético
Z	componente z do campo magnético
τ	temperatura interna

Unidade de Medição Inercial (Inertial)	
Identificação do Observador	I
Informação do Sensor	Descrição
a	aceleração na direção x
b	aceleração na direção y
c	aceleração na direção z
p	taxa de rotação x
q	taxa de rotação y
r	taxa de rotação z
θ	ângulo de pitch
Φ	ângulo de roll
τ	Temperatura
t	Tempo

GPS da Base (Base)	
Identificação do Observador	B
Somente envio de dados recebidos através da porta serial	

Sensor Ultrasom (Ultrasound)	
Identificação do Observador	U
Informação do Sensor	Descrição
h	altura

GPS móvel (Rover)	
Identificação do Observador	R
Informação do Sensor	Descrição
w	semana
t	tempo
x	posição em x
y	posição em y
z	posição em z
u	componente x da velocidade
v	componente y da velocidade
w	componente z da velocidade
a	aceleração na direção x
b	aceleração na direção y
c	aceleração na direção z
P	status posição: intensidade do sinal
V	status velocidade: intensidade do sinal
A	status aceleração: intensidade do sinal
h	altitude
f	latitude
g	longitude
H	standard altitude deviation
F	standard latitude deviation
G	standard longitude deviation
D	datum
U	undulation
S	solution status

4.2.3 Interface Gráfica

4.2.3.1 Java/Swing

Quando se usa Java/Swing são criadas aplicações que rodam no mínimo em 4 plataformas diferentes: Windows, Linux, Solaris e Mac OS. O Java oferece um nível de abstração quando se constrói interfaces gráficas maior do que qualquer outra linguagem, porque ele deve suportar muitas plataformas diferentes. Como o projeto atual é interligado com outros projetos, que utilizarão sistemas operacionais diferentes, vemos a vantagem da linguagem Java. Assim, levando também em consideração que essa linguagem de programação já é conhecida dos autores do projeto, optou-se pela sua utilização.

Para obter-se a interface gráfica desejada, podemos utilizar a biblioteca gráfica de Java, Swing. Essa biblioteca permite e gerencia o uso de janelas, botões, gráficos, imagens, animações, etc.

Pode-se usar também a API 2D Java, que permite que sejam adicionados imagens de alta qualidade ao programa.

No release 1.4 da plataforma Java, a API Swing tem 17 packages públicos:

Javax.accessibility	javax.swing.plaf	javax.swing.text.html
Javax.swing	javax.swing.plaf.basic	javax.swing.text.parser
Javax.swing.border	javax.swing.plaf.metal	javax.swing.text.rtf
Javax.swing.colorchooser	javax.swing.plaf.multi	javax.swing.tree
Javax.swing.event	javax.swing.table	javax.swing.undo
javax.swing.filechooser	javax.swing.text	

Com essas bibliotecas, podemos construir a GUI (Graphical User Interface) desejada.

Temos à disposição, no Swing, elementos como bordas, botões, imagens, painéis e animações.

Botões são um conjunto de elementos que disparam eventos; mais exemplos são dados na tabela

(1) abaixo:

Tabela 1 - Eventos da biblioteca Swing e seus listeners associados

Ação que resulta no evento	Action Listener
Usuário clica em um botão, aperta Enter em uma caixa de texto ou escolhe um item em um menu	ActionListener
Usuário fecha um frame (janela principal)	WindowListener
Usuário clica em uma região da janela	MouseListener
Usuário move o mouse sobre uma janela	MouseMotionListener
Componente se torna visível	ComponentListener
Componente toma o foco do teclado	FocusListener
A seleção da tabela ou da lista muda	ListSelectionListener
Alguma propriedade em um componente muda, como o texto em um rótulo	PropertyChangeListener

4.2.3.1.1 Usando gerenciador de layouts

Uma das ferramentas mais importantes do Swing é o gerenciador de layouts. Com ele, pode-se definir como será a janela do aplicativo a ser programado, definindo como esta janela será dividida em painéis.

Alguns layouts de uso geral são os seguintes:

- BorderLayout
- BoxLayout

- CardLayout
- FlowLayout
- GridBagLayout
- GridLayout
- SpringLayout

Para uso neste trabalho, os mais importantes layouts, que permitem uma melhor personalização da janela, são:

- **GridBagLayout**

GridBagLayout é um gerenciador de layout flexível. Ele alinha os componentes posicionando-os dentro de uma grade de células, permitindo que alguns componentes ocupem mais de uma célula. As linhas da grade podem ter diferentes alturas, e as colunas podem ter diferentes larguras.



Figura 10 - Exemplo de GridBagLayout

- **SpringLayout**

SpringLayout também é um gerenciamento flexível, para ser utilizado por construtores de Gui's. Ele permite que se especifique precisamente o relacionamento entre as bordas de cada componente da janela.

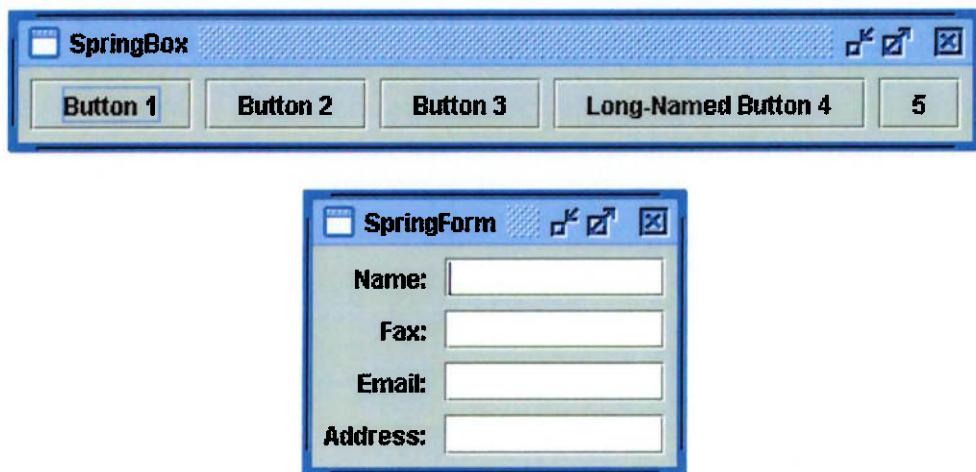


Figura 11 - Exemplo de SpringLayout

4.2.3.1.2 Imagens múltiplas

Quando trabalhamos com imagens múltiplas, temos que escolher se trabalhamos carregando todas as imagens inicialmente ou carregamos à medida em que elas são necessárias (processo conhecido como “carregamento de imagens preguiçoso”).

Para carregar as imagens, usamos o `getSystemResource`, um método que procura o endereço da classe para achar os arquivos de imagem.

4.2.3.1.3 Temporizadores do Swing

Um temporizador Swing (da biblioteca javax.swing.Timer) dispara uma ou mais ações depois de um tempo específico.

Em geral, é recomendável usar temporizadores (timers) Swing ao invés de temporizadores de uso geral, para processos relacionados a GUI. Isso porque os timers Swing dividem o mesmo pré-existente processo de temporização, e os processos relacionados ao GUI são executados automaticamente nos processos disparadores de evento.

Os timers Swing podem ser usados de duas formas:

- Percorrer o processo uma vez, após um atraso
- Percorrer um processo repetidamente

Quando se cria um timer, deve ser especificado um action listener, que será notificado quando o temporizador atingir a contagem. O método actionPerformed neste listener deve conter o código para os processos a serem seguidos. Ao criar um timer, também é especificado o número de milissegundos necessários para dispará-lo.

Abaixo há uma foto de uma aplicação que usa um timer e uma barra de progresso. A cada disparo do timer, a barra de progresso é atualizada.

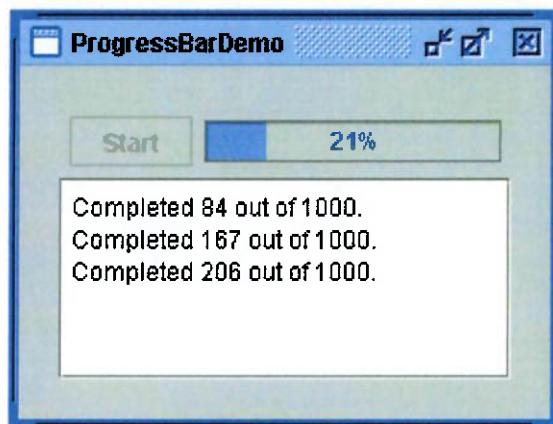


Figura 12 - Exemplo de aplicativo que utiliza Timer

A tabela seguinte mostram os construtores e métodos utilizados da biblioteca javax.swing.Timer. Esta API se divide em duas categorias:

- Criando e inicializando o Timer
- Usando o Timer

Tabela 2 - Criando e inicializando um Timer Swing

Método ou Construtor	Objetivo
Timer(int, ActionListener)	Cria um temporizador Swing. O argumento int especifica o número de milissegundos de parada entre os eventos.
void setDelay(int)	Define ou recebe o número de milissegundos de parada.
int getDelay()	
void setInitialDelay(int)	Define ou recebe o número de milissegundos a serem esperados antes de disparar a ação do primeiro evento.
int getInitialDelay()	
void setRepeats(boolean)	Define ou recebe se o temporizador usará repetição.
boolean isRepeats()	
void setCoalesce(boolean)	Define ou recebe se o temporizador reúne múltiplas ações pendentes em uma única ação.
boolean isCoalesce()	
addActionListener(listener)	Adiciona ou remove um action listener
removeActionListener(listener)	

Tabela 3 - Utilizando um Timer Swing

Método	Objetivo
void start()	Liga o temporizador. Restart também cancela ações pendentes
void restart()	
void stop()	Desliga o temporizador
boolean isRunning()	Recebe a informação de temporizador ligado ou não

4.2.3.1.4 Conclusão do método I

Através das características apresentadas nos itens anteriores, é possível descrever a solução utilizada para a construção da interface gráfica. Utilizando-se o gerenciador de layouts, separa-se a janela do programa em displays diferentes, cada display representando um relógio de medição. Cada relógio terá múltiplas imagens, cada uma representando uma posição do ponteiro. A imagem a ser mostrada depende do valor da leitura dos sensores, de acordo com a faixa de valores obtidos. Fazendo com que o timer atualize essas imagens em um tempo razoável, teremos o efeito de animação, com os ponteiros mostrando em tempo real os valores corretos.

Essa solução tem dois problemas:

- 1) É pouco sofisticada, pois a resolução da leitura depende da quantidade de imagens disponíveis para representar as posições dos ponteiros;
- 2) Dependendo do número de imagens, o programa pode se tornar pesado. Exemplo: Se tivermos 8 sensores, com 100 imagens diferentes cada um, e levando em consideração que cada imagem de display tem em média 8 Kbytes, teremos 6,25 MegaBytes apenas de imagens.

Apesar disso, há a vantagem da simplicidade: podemos nos concentrar, inicialmente, em conseguir a comunicação entre os modems e usar os valores obtidos para construir a interface gráfica usando essa técnica. Após os testes necessários, podemos partir para o refinamento do programa, substituindo essas imagens por animações “reais”.

Também pode ser possível o uso do programa dessa forma, mesmo acrescentando mais imagens para se melhorar a resolução. Isso dependerá dos testes de desempenho.

4.2.3.2 Interface gráfica/Graphics

A interface apresentada anteriormente é considerada ultrapassada, pois há perda de informação da câmera quando uma parte da tela é tomada por relógios. Houve, então, alterações no projeto. A alternativa sugerida é usar a interface como abaixo:

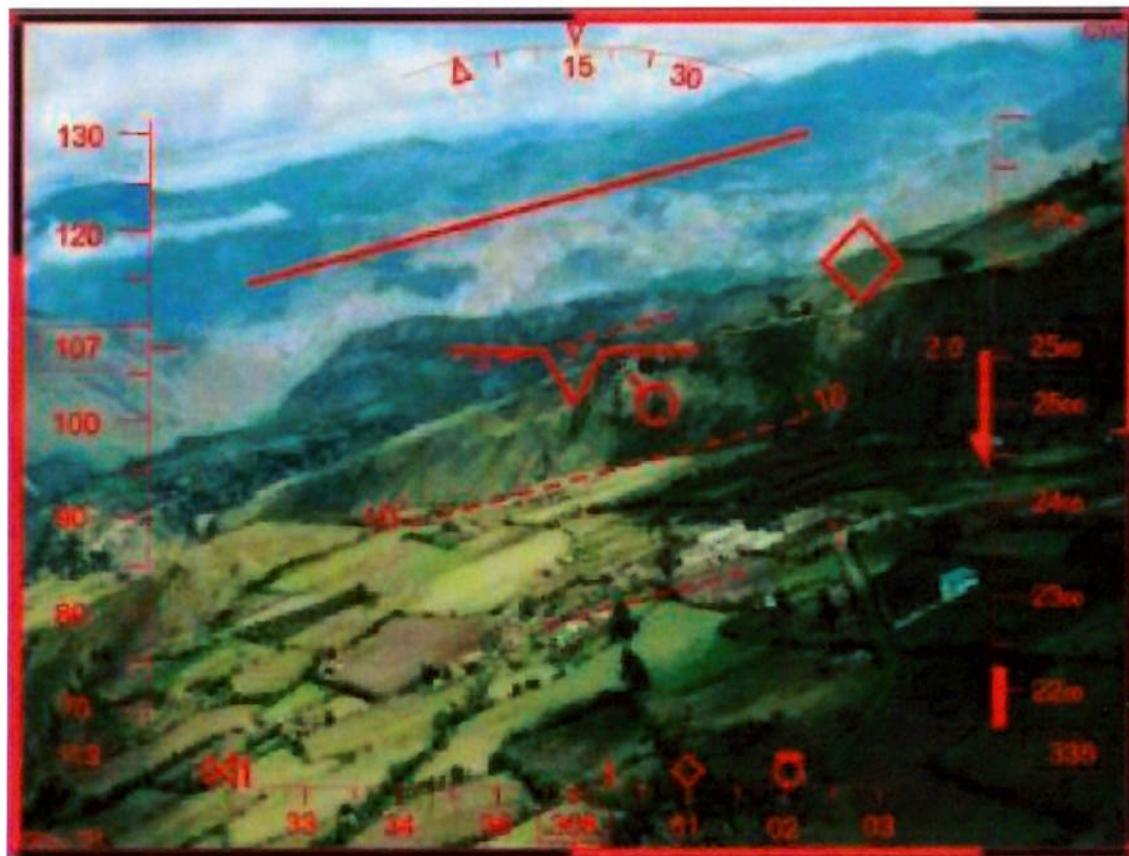


Figura 13 - Interface otimizada

Assim, os mostradores estão sobrepostos à própria imagem da câmera, sem perda de informação.

Como os mostradores devem estar sobre a imagem filmada, não podemos simplesmente criar painéis utilizando o Swing da forma exposta anteriormente. Será usada, então, a classe Graphics (java.awt.Graphics). . Essa classe tem várias ferramentas utilizadas para desenho, como criação de imagens, movimentação e rotação de figuras, etc. Alguns métodos importantes serão descritos a seguir.

4.2.3.2.1 Requisitos

Alguns requisitos devem ser preenchidos pela nova definição de interface:

- Alguns marcadores (altitude, velocidade e bússola) devem ser visualizados em apenas uma área definida da tela, apesar de sua escala ser relativamente grande.
- Alguns marcadores devem mudar de cor, para chamar a atenção do usuário.
- Os marcadores de pitch e roll, além do nível bolha, devem seguir uma relação de rotação e translação entre de si, de forma que seja visível em um elemento só a forma de movimentação do UAV

A visualização desses requisitos ficará mais clara nas seções seguintes.

4.2.3.2.2 O método paint

É o método utilizado para desenhar um componente, seja para escrever uma letra, desenhar uma linha ou um círculo.

Como para cada vez que um método é desenhado ele invoca o método paint, ele pode ser chamado diversas vezes em um programa. Exemplos: o método deve ser chamado quando a

imagem é desenhada pela primeira vez, ou quando o usuário cobre o programa com uma janela e o descobre novamente.

4.2.3.2.3 O método update

O método update limpa a superfície do componente, desenha seu fundo e chama o método paint para pintar o conteúdo novamente novamente. Isto é conveniente pois não é necessário desenhar o componente inteiro dentro de um método paint, pois o fundo já é preenchido. Então, só é preciso desenhar novamente o que está por cima.

4.2.3.2.4 Repainting

Como dito anteriormente, os métodos paint e update desenham a superfície da imagem e atualizam quando há outro evento, como uma janela ser sobreposta a janela do programa. Porém, há casos em que é necessário redesenhar a imagem de qualquer forma. Um exemplo é quando alteramos a propriedade de uma imagem e desejamos mostrar essa alteração imediatamente. É o caso da animação da interface, que é o objetivo do trabalho.

4.2.3.2.5 Uso de GIFs

Para representarmos os mostradores, utilizamos imagens em formato GIF. Esse formato foi o escolhido pois permite que se defina seu fundo como transparente. Assim, no caso do GIF abaixo mostrado, que tem o fundo em branco definido como transparente, a parte visível na interface é apenas o marcador em verde.

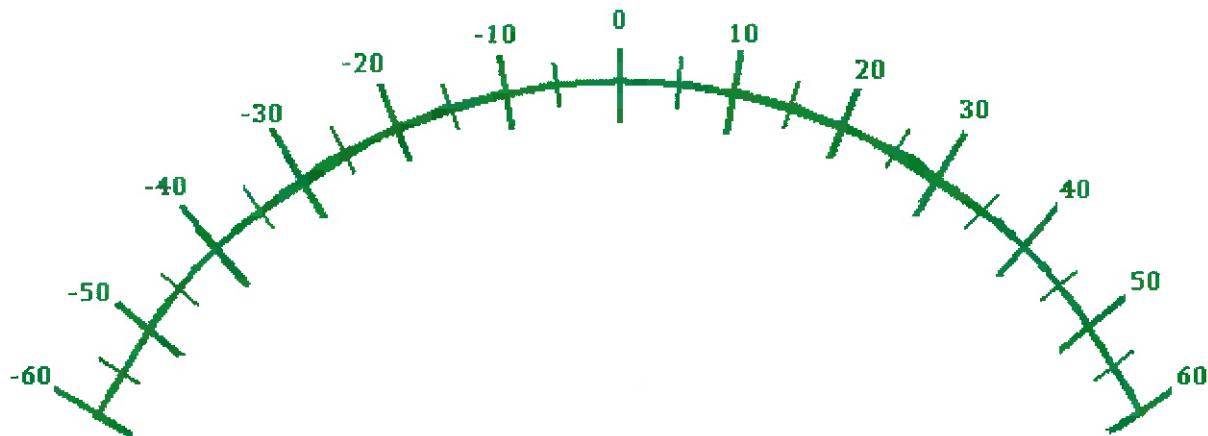


Figura 14 - Ponteiro no formato GIF

Figuras em GIF também podem ser transladadas e rotacionadas, requisitos essenciais para a construção da interface gráfica desejada. Para que se leve uma figura no formato GIF para a estrutura do programa, utiliza-se o método

```
Image arquivo = getImage(getCodeBase(),"arquivo.gif"),
```

onde a figura “arquivo.gif” deve ser colocada na pasta raiz do programa. Assim, a variável “arquivo” do tipo Image representa o arquivo desejado.

4.2.3.2.6 Classes Graphics e Graphics2D

As classes Graphics e Graphics2D são as classes que realizam as operações gráficas, modificam figuras e se relacionam com o método paint citado acima. A classe Graphics é responsável pelos elementos mais simples da interface, como desenho de linhas e barras. A classe Graphics2D altera as propriedades de figuras. Como ela estende a classe Graphics, todas as funcionalidades anteriores se mantêm, permitindo que usemos apenas um elemento Graphics2D no programa. Os principais métodos utilizados são:

- scale (double x, double y) – a imagem tem seu tamanho alterado de um fator x na horizontal e um fator y na vertical. Necessário para tornar a imagem capturada da câmera (640 X 480) compatível com o tamanho da tela (1024 X 768);
- rotate (double ângulo, double x, double y) – rotaciona a imagem desejada em torno do ponto (x,y) de acordo com o ângulo dado;
- translate (double x, double y) – translada a imagem na horizontal e na vertical;
- setColor e setFont – esses métodos definem o tamanho da fonte e a cor que as imagens serão desenhadas na tela;
- setClip (rect) – determina uma área específica da tela para que a imagem seja desenhada. Neste caso, definimos um retângulo rect, e de acordo com esse retângulo (com origem e tamanho) a área de desenho é definida;
- drawString – usada para escrevermos os números e siglas necessárias na interface;
- fillRect (int a, int b, int c, int d) – define o desenho de um retângulo, de acordo com as coordenadas dadas pelos inteiros. Esse desenho é necessário para que façamos as barras medidoras;

- draw Image (Image imagem, int x, int y, ImageObserver) – método mais importante utilizado. É o método que desenha as imagens desejadas na tela, de acordo com as coordenadas especificadas.

4.2.3.2.8 Captura da câmera

Para visualização da câmera, utilizamos uma API Java chamada JMF (Java Media Framework). Esta API é responsável por trabalhar com os drivers e dispositivos disponíveis, definindo quais são os formatos de mídia que podem ser utilizados. No caso do nosso trabalho, as funções necessárias envolvem o gerenciamento de vídeo. O código utilizado é o seguinte:

```
CaptureDeviceInfo deviceInfo = CaptureDeviceManager.getDevice("driver");
player = Manager.createRealizedPlayer(deviceInfo.getLocator());
player.start();
FrameGrabbingControl frameGrabber =
(FrameGrabbingControl)player.getControl("javax.media.control.FrameGrabbingControl");
Buffer buf = frameGrabber.grabFrame();
background = new BufferToImage((VideoFormat)buf.getFormat()).createImage(buf);
```

CaptureDeviceInfo é uma classe que guarda as informações referentes ao dispositivo capturado. Essa captura é feita através do método getDevice da classe CaptureDeviceManager, e função

start inicia o dispositivo. O método grabFrame retorna um frame da câmera, que é transformado em uma variável do tipo Image, podendo ser utilizada na interface.

De acordo com as funções expostas acima, vemos que é possível capturar imagens da câmera em momentos específicos e transformá-las em um arquivo JPEG, utilizando a função apropriada. Assim, ela funciona também como uma câmera fotográfica.

4.2.3.2.8 Animação

A animação utiliza a classe Thread para obtermos a sensação de movimento. Threads são componentes utilizados para multiprocessamento, e serão detalhados na seção 4.2.4. O Thread é utilizado através do código:

```
runner = new Thread (this);  
runner.start();
```

Neste caso, é criado um novo Thread, chamado runner. O comando start o inicia. O Thread roda enquanto o programa não for fechado. Dentro do programa, é usado o comando sleep, que faz com que o Thread espere um determinado tempo em milissegundos. A cada período determinado pelo sleep, é construída a imagem que aparece na tela. Com o comando runner.sleep (46), conseguimos obter um programa que roda em uma velocidade razoável. Para construir a imagem a ser desenhada, é necessário ter um componente chamado BufferedImage. Esse componente permite que façamos imagens complexas, com várias elementos. Isso porque o BufferedImage

age como uma memória, guardando os elementos que são dispostos ao longo do programa. Quando queremos colocar na tela o desenho pronto, chamamos a função `drawImage(BufferedImage)`.

Estruturalmente, o programa funciona da seguinte forma: são feitas as transformações de coordenadas necessárias (como rotação, translação, ou determinação de área de desenho), e desenha-se um elemento, colocando-o no `BufferedImage`, através do método `drawImage`. Ao fim do período do Thread, é desenhado o próprio `BufferedImage`, dentro do método `paint`.

4.2.3.2.9 Interface obtida e descrição

Obtivemos a interface mostrada na Fig. 15, mostrada a seguir.

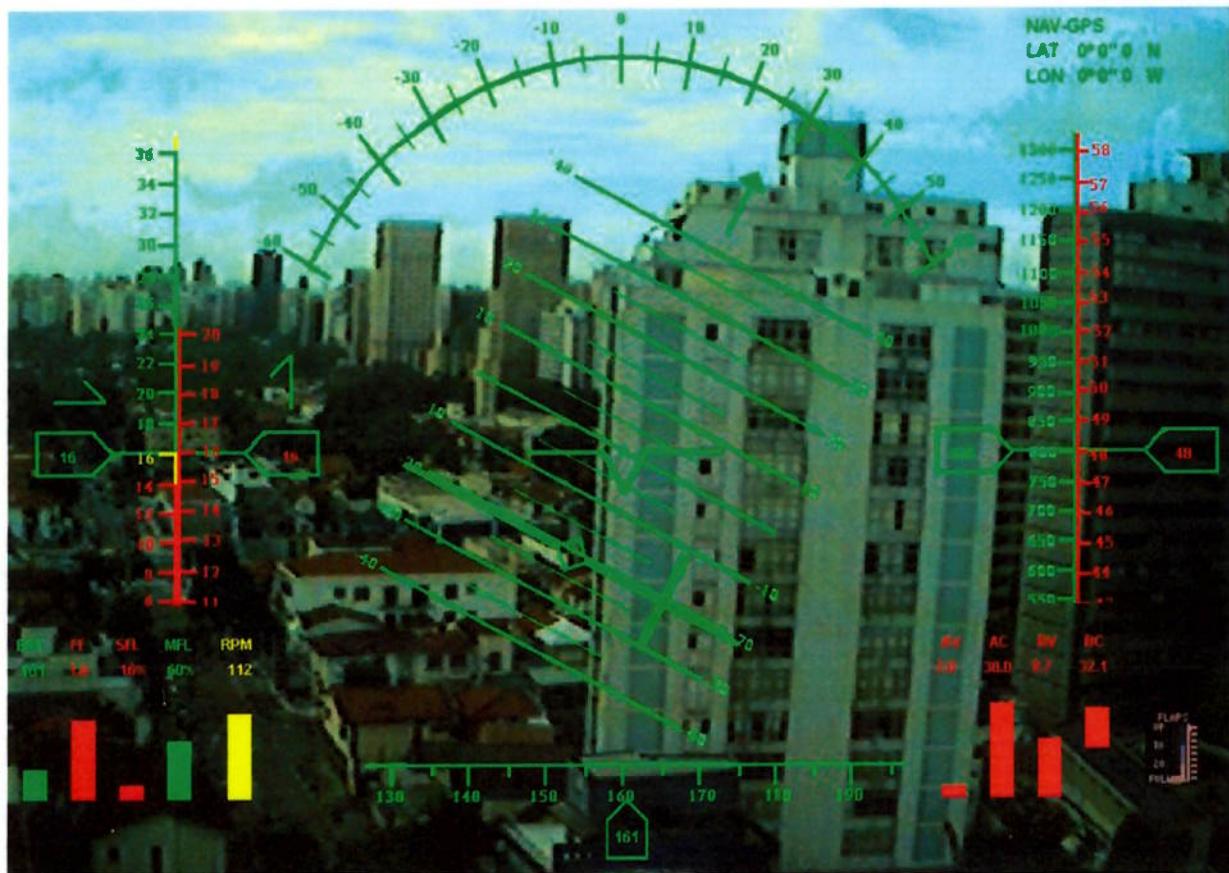


Figura 15 - Interface final

Na parte superior, temos o marcador de ângulo de rolagem, onde seu ponteiro é a seta que fica logo abaixo. Essa seta gira juntamente com o marcador do ângulo de arfagem, cujo ponteiro é dado pela reta que se move verticalmente neste marcador. O inclinômetro, ou nível bolha, é a reta vertical que acompanha o ponteiro do ângulo de arfagem. O ângulo de guinada é dado pelos valores da escala que fica na parte de baixo da tela.

As barras de medição que ficam nos cantos inferiores direito e esquerdo medem tensões, correntes, temperatura, rotação do motor e nível de combustível. Mudam de cor entre vermelho, amarelo e verde de acordo com a periculosidade da situação (baixo nível de combustível é vermelho, por exemplo).

As escalas que ficam à esquerda e à direita movem-se verticalmente, e indicam os valores de velocidade e altitude, respectivamente. O valor atual obtido pelos sensores é destacado na área próxima à escala, e a cor também se altera de acordo com os valores.

Na extrema direita, na parte de baixo da tela, temos o marcador de flaps. Na parte superior, há a marcação de latitude e longitude.

4.2.3.2.12 Conclusão do método II

Apesar de, inicialmente, a mudança no projeto ter se mostrado complexa, a solução encontrada se mostrou mais eficiente. Isso porque, além do uso do método paint e threads ser mais “elegante” em relação à linguagem, o método tem uma performance bem melhor.

Não é preciso ter um ponteiro para cada posição ou dado a ser mostrado, mas sim alterar a posição de cada ponteiro.

4.2.4 Execução de processos em paralelo

Para resolver o problema de executar-se diferentes processos em paralelo, como apresentação de vídeo, recepção e transmissão de dados via ethernet e serial, bem como a apresentação para o usuário desses dados para o usuário foi utilizado o conceito de threads que foi aplicado no programa apresentado.

Um thread é um único fluxo seqüencial de controle dentro de um programa. Ele permite que se façam várias tarefas simultâneas dentro de um programa, sendo desnecessário esperar que um determinado método acabe para que se comece outro. Temos a seguir alguns dos benefícios de trabalhar com threads:

- Capacidade de resposta: permite-se a um programa continuar executando mesmo se parte dele estiver bloqueada ou realizando uma operação prolongada, aumentando desse modo a capacidade de resposta ao usuário
- Compartilhamento de recursos: os threads compartilham, por default, a memória e os recursos do processo ao qual pertencem. A vantagem de compartilhamento do código é permitir que uma aplicação tenha diversos threads de atividade, todos dentro do mesmo espaço de endereçamento
- Economia: a alocação de memória e de recursos para a criação de processos é custosa. Por outro lado, os threads compartilham recursos do processo ao qual pertencem, é mais econômico criar threads e comutar seus contextos

5 ANÁLISES E CONCLUSÃO

Podemos separar o projeto em duas partes principais a serem analisadas a comunicação entre módulos aéreo e base e a interface com o usuário.

5.1 COMUNICAÇÃO MÓDULO AÉREO E MÓDULO BASE

Com a escolha do hardware, dos protocolos e métodos adequados de comunicação foi possível obter uma simulação fiel ao que realmente ocorrerá na comunicação entre os módulos já citados, obedecendo a semântica estabelecida no início do projeto.

A presença de atrasos no processo de comunicação é inevitável, no entanto, o atraso mostrou-se pouco significativo, de modo a não interferir no processo de pilotagem remota, ou seja, a transmissão e recepção de dados do módulo base não é um ponto restritivo a pilotagem da aeronave.

O programa apresentado mostrou-se compatível com o esperado. Apresenta-se dentro dos requisitos tanto de hardware quanto de software propostos pelo projeto.

5.2 INTERFACE

Visualmente, a interface preencheu os requisitos. As informações são passadas de forma correta e organizada na tela, e toda a movimentação exigida foi obtida. Os marcadores se mantêm sobrepostos à imagem capturada da câmera, permitindo que toda a tela esteja disponível para a visualização do percurso feito pela aeronave.

Quanto à performance, há um pequeno atraso na exibição da imagem capturada, um pouco menor do que um segundo. É necessário fazer testes para verificar o quanto este atraso é significativo. Como o avião não tem como objetivo fazer manobras complexas, é possível que o atraso seja aceitável. Em compensação, pode haver dificuldades no pouso e decolagem da aeronave, pois são momentos críticos na pilotagem.

REFERÊNCIAS

ANATEL. Quadro de distribuição de faixas de freqüência no Brasil. Disponível em: <<http://www.anatel.gov.br/index.asp?link=/Radiofrequencia/qaff.pdf>>. Acesso em: 02 de Abril de 2005

Antenas – Informações básicas. Disponível em: <myspace.eng.br/eng/antena1.asp>. Acesso em: 27 de março de 2005

Extropia/Tutorials/Introduction to Java. Disponível em: <<http://www.extropia.com/tutorials/java/toc.html>>. Acesso em: 10 de Setembro de 2005

Goodrich, M. T., Tamassia, R., 2002, “**Estrutura de Dados e Algoritmos em Java**”, 2 ed. Porto Alegre – RS, BRA: Bookman Companhia Editora.

Goularte, R., Moreira, E. S., 1998, “**Produção de vídeo digital: teoria e prática**”, Notas didáticas do ICMC-USP 30, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, Brasil.

A Java GUI Programmer's Primer - Table of Contents. Disponível em: <http://www.scism.sbu.ac.uk/jfl/jibook/>. Acesso em: 05 de Abril de 2005

Lathi, B. P., 1988, “**Modern digital and analog communication systems**”, Bhagwandas Pannalal

Microsoft, Software Flight Simulator

Portal Java. Disponível em: <<http://www.portaljava.com.br/home/index.php>>. Acesso em: 02 de Abril de 2005

Silberschatz, Galvin and Gagne, 2001, “**Operating System Concepts**”, 6 ed, Addison-Wesley

Tanenbaum, A. S., 1996, “**Computer Networks**”, 3 ed. Upper Saddle River – NJ, USA: Prentice Hall.

Walrath, K., Campione, M., Huml, A., Zakhour, S., 1998, “**The Java Tutorial Continued: The Rest of the JDK**”, Bk&CD-Rom edition Addison-Wesley Professional

Walrath, K., Campione, M., Huml, A., Zakhour, S., 2004, “**The JFC Swing Tutorial: A Guide to Constructing GUIs**”, 2 ed. Addison-Wesley Professional

Wikipedia – Linguagem de programação. Disponível em:
http://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o. Acesso em: 23 de Março de 2005

Yacoub, M. D., 1993, “**Foundations of mobile radio engineering**”. Boca Raton – FL, USA: CRC Press.