

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Fábio Hideki Sigaki da Rocha

**Estudo de telemetria para aquisição de dados durante
ensaio de voo de avião**

São Carlos

2018

Fábio Hideki Sigaki da Rocha

**Estudo de telemetria para aquisição de dados durante
ensaio de voo de avião**

Monografia apresentada ao Curso de Engenharia Elétrica com Ênfase em Eletrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Amílcar Careli César

**São Carlos
2018**

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

He	Hideki Sigaki da Rocha, Fabio Estudo de telemetria para aquisição de dados durante ensaio de voo de avião / Fabio Hideki Sigaki da Rocha; orientador Amílcar Careli César. São Carlos, . Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, . 1. Telemetria. 2. GPS. 3. Google Earth. 4. Aquisição de Dados. 5. Ensaio de voo de avião. I. Título.
----	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

FOLHA DE APROVAÇÃO

Nome: Fábio Hideki Sigaki da Rocha

Título: "Estudo de telemetria para aquisição de dados durante ensaio de voo de avião"

Trabalho de Conclusão de Curso defendido e aprovado
em 24 / 11 / 2018,

com NOTA 8,0 (oito, zero), pela Comissão Julgadora:

Prof. Titular Amílcar Careli César - Orientador - SEL/EESC/USP

Prof. Dr. Leonardo André Ambrosio - SEL/EESC/USP

Mestre André Luiz Ferraz Lourenço - Doutorando - SEL/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

RESUMO

ROCHA, F. **Estudo de telemetria para aquisição de dados durante ensaio de voo de avião.** 2018. 38p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2018.

O presente trabalho tem como objetivo o desenvolvimento de um *software*, que é usado como suporte durante um ensaio de voo para a visualização do avião, em tempo real, utilizando o *Google Earth* como ferramenta visual. Para isso, o programa recebeu dados coletados de um sistema presente no avião. Para realizar a visualização do avião, foi utilizado um sistema de telemetria para adquirir dados em tempo real de latitude, longitude, altitude, inclinação, rotação e velocidade e enviar do sistema a bordo do avião para a estação em solo. A estação em solo recebe e processa os dados adquiridos e o *software* desenvolvido faz a conexão com o Google Earth utilizando um formato de arquivo específico chamado KML. O resultado do projeto foi o desenvolvimento do software GECliente, que coleta dados e mostra a posição no mapa do avião em tempo real durante um ensaio de voo.

Palavras-chave: Telemetria. GPS. Google Earth. Aquisição de Dados. Ensaio de voo de avião.

ABSTRACT

ROCHA, F. **Telemetry study for data acquisition during flight test**. 2018. 38p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2018.

The present work aims at the development of software that is use as support during a flight test to visualize,in real-time, the airplane in 3D using Google Earth as visual tool. The software receives and processes data collected from an on-bord system. To perform the visualization of the airplane, the telemetry system was used to acquire real-time data on latitude, longitude, altitude, tilt, heading, holl and speed to send from the on-board system to the ground station. The ground station processes the acquired data and the software that has been developed connect to Google Earth using a specific file format called KML. The developed software, named GECliente, collects data and shows on the map the position of the airplane in real time during a flight test.

Keywords: Telemetry. GPS. Google Earth. Data acquisition. Flight test.

LISTA DE FIGURAS

Figura 1 – Representação do sistema de telemetria de uma Formula 1.	12
Figura 2 – Sistema de Telemetria.	15
Figura 3 – Multiplexação de um sinal.	17
Figura 4 – Estrutura de um pacote PCAP com arquivos iNET-X.	19
Figura 5 – Estrutura completa de um pacote de dados.	20
Figura 6 – Representação do GEClient no sistema de telemetria.	23
Figura 7 – Representação da latitude e longitude no globo terrestre.	24
Figura 8 – Exemplo KML NetworkLink.	25
Figura 9 – Exemplo de KML Camera.	26
Figura 10 – Representação da função <i>Orientation</i>	27
Figura 11 – Representação da função <i>LookAt</i>	28
Figura 12 – Interface do Software desenvolvido GEClient.	29
Figura 13 – Fluxograma do GEClient.	31
Figura 14 – Representação de uma simulação de ensaio usando rede local.	32
Figura 15 – Simulação para uma altitude de visualização de 400 mil metros.	33
Figura 16 – Simulação para uma altitude de visualização de 50 mil metros com <i>Tilt</i> = 0° e <i>Heading</i> = 0°.	34
Figura 17 – Simulação para uma altitude de visualização de 50 mil metros com <i>Tilt</i> = 45° e <i>Heading</i> = 315°.	35
Figura 18 – Simulação visão <i>Cockpit</i>	35

LISTA DE ABREVIATURAS E SIGLAS

GPS	Global Position System
IEEE	Instituto de Engenharia Elétrica e Eletrônica
iNet	integrated Network-Enhanced Telemetry
KML	Keyhole Markup Language
LAN	Local Area Network
MAC	Media Access Control
PCAP	Packet Capture
PCM	Pulse-code modulation
TDM	Time-division multiplexing
VS	Visual Studio

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Objetivo	13
1.2	Estrutura da monografia	13
2	CONCEITOS GERAIS	14
2.1	Telemetria	14
2.1.1	O que é Telemetria	14
2.1.2	Panorama de um sistema de Telemetria	14
2.2	Sistema em avião	16
2.2.1	Aquisição de dados	16
2.2.2	Multiplexação	16
2.2.3	PCM	17
2.2.4	<i>Ethernet</i>	18
2.2.5	IEEE 802.3	18
2.2.6	iNET-X	18
2.2.7	PCAP	19
2.3	Estação em solo	21
2.3.1	<i>Setup</i>	21
2.3.2	Reconstrução PCM	21
2.3.3	Processamento em tempo real	21
2.3.4	Distribuição de dados	22
3	METODOLOGIA	23
3.1	GPS	23
3.2	Google Earth	24
3.3	.KML	24
3.3.1	KML <i>NetworkLink</i>	25
3.3.2	KML Camera	25
3.4	<i>Visual Studio</i>	27
3.5	<i>WxWidgets</i>	28
3.6	GECient	28
4	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	32
5	CONCLUSÃO E TRABALHOS FUTUROS	36

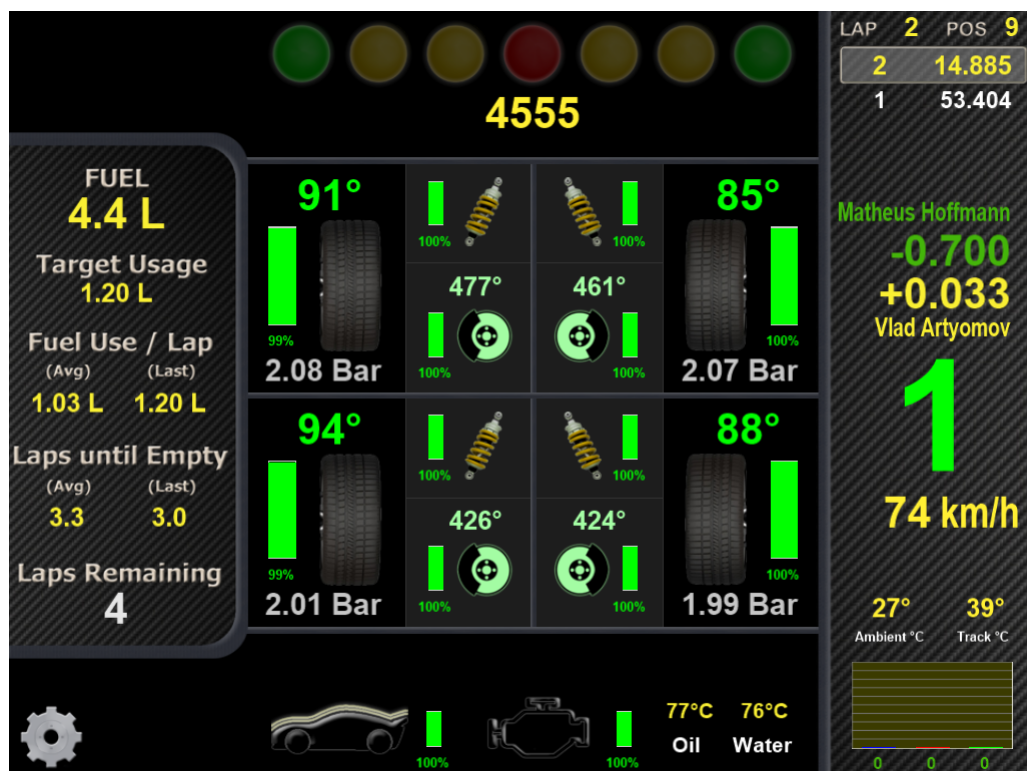
REFERÊNCIAS 37

1 INTRODUÇÃO

Pode-se definir telemetria como um sistema de transmissão de dados que hoje é, amplamente, utilizado em carros de Fórmula 1 e em ensaios de voos aeronáuticos. Os primeiros registros da utilização de um sistema rústico de telemetria datam de 1930, quando um balão equipado com transceptores de radiofrequência foi usado para captar informações sobre o clima (KERR; MCELROY, 1976).

Atualmente, a telemetria é conhecida pela aplicação nos carros de Fórmula 1, nos quais transmite informações, durante as corridas, de dados relevantes ao desempenho do carro (MODONIA; SACCOB; TERKAJ, 2016). Durante o desenvolvimento de um carro de Fórmula 1, sensores são instalados com o objetivo de possibilitar o monitoramento em tempo real e enviar informações do carro, a cada volta, para os boxes de controle, sendo, então, possível realizar o seu monitoramento em tempo real. Um exemplo disso está na Fig.1, na qual podemos verificar os dados sendo transmitidos por essa coleção de sensores embarcados no carro.

Figura 1: Representação do sistema de telemetria de uma Formula 1.



Fonte: projectcarsgame (2012).

A telemetria também teve um grande desenvolvimento em aplicações dentro da indústria aeronáutica, trazendo possibilidades de monitoramento de ensaios de voo em tempo real. Durante

o processo de fabricação de um produto dentro de uma indústria aeronáutica, os ensaios de voo são de grande importância para verificar possíveis falhas e corrigi-las antes da aeronave ser certificada para utilização em voos tripulados. A realização desses ensaios durante o processo de desenvolvimento de um novo avião é importante, uma vez que se investe muito dinheiro para desenvolver e criar uma aeronave que atinja os requisitos de *design* e segurança para atender as necessidades do cliente.

Antes de um avião ser certificado, ele é testado por milhares de horas de voo. Ele é colocado em rigorosos testes, como bruscas variações de temperatura e análise de comportamento em relação a turbulências (KUMAR et al., 2016). Para auxiliar nesses testes, utiliza-se um sistema de telemetria para aquisição de dados remotamente em tempo real (ROLANDO; MINICHINI; ROSSI, 2016).

Além do auxílio durante um ensaio de voo, o sistema de telemetria aviônica é utilizado em aviões não tripulados para enviar informações para a estação em solo (ROLDAN et al., 2016), as quais podem ser desde imagens coletadas pelo avião (C.M.GEVAERT et al., 2017) até dados de GPS (WINTERNITZ et al., 2016). A partir disso, desenvolvemos um *software* que tem como função capturar dados de GPS do avião por meio de um sistema de telemetria.

1.1 Objetivo

O objetivo deste trabalho é desenvolver um software para capturar dados de GPS em tempo real, utilizando:

- a) Dados em tempo real de latitude, longitude e altitude do avião durante um ensaio de voo; e
- b) O Google Earth, para realizar uma visualização em 3D dos dados coletados.

1.2 Estrutura da monografia

No capítulo 2 são abordados conceitos gerais de um sistema de telemetria. Neste capítulo descrevemos como um dado obtido no avião é enviado para a estação em solo e capturado pelo GEClient. Apresentamos as características de um sistema a bordo e da estação em solo. No capítulo 3, descrevemos como o *software* desenvolvido se comunica com o Google Earth. São especificadas funções próprias do Google Earth e o formato do arquivo gerado para a comunicação. No capítulo 4, enumeramos os resultados obtidos no trabalho e o procedimento de teste do *software*. No capítulo 5, são apresentadas as considerações finais.

2 CONCEITOS GERAIS

2.1 Telemetria

2.1.1 O que é Telemetria

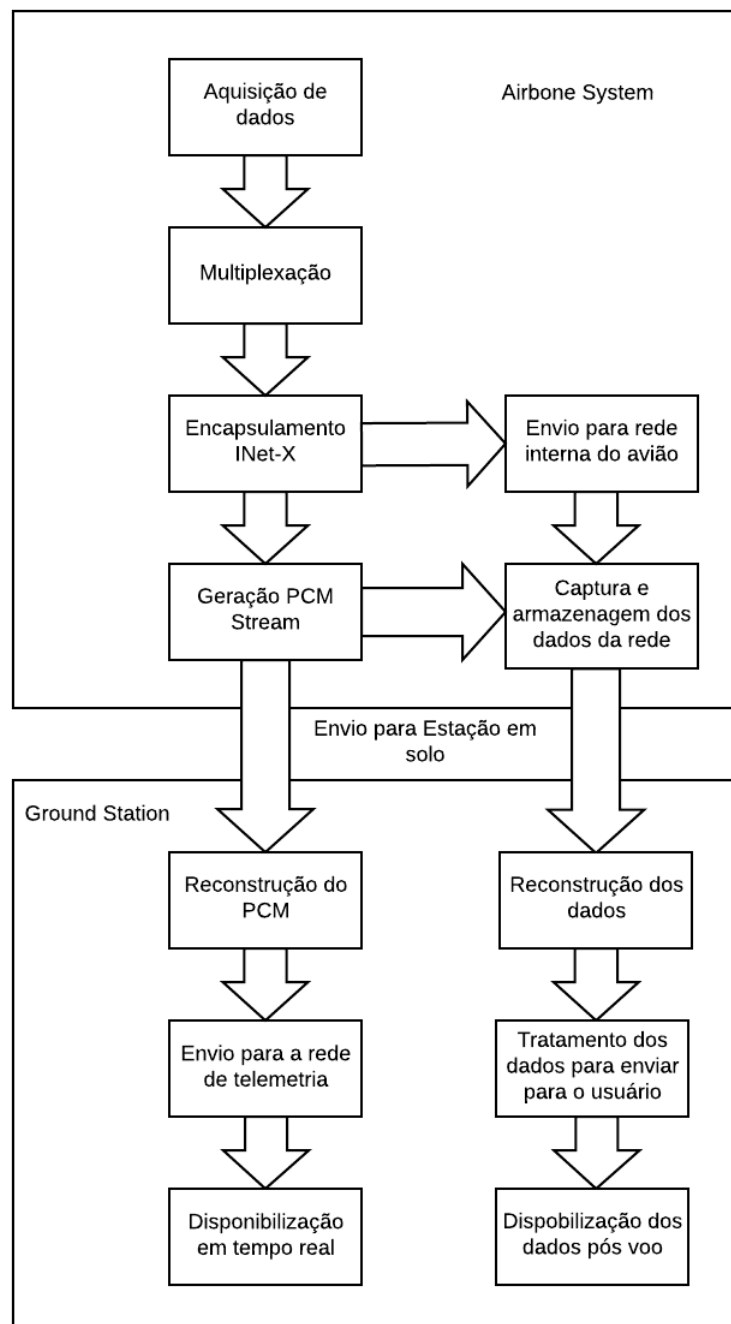
Telemetria é um sistema de comunicação que permite a aquisição e transmissão de dados de forma remota. Essa transmissão pode ser feita por um meio físico, como cabo coaxial, par de fios e fibra óptica, ou via rádio em enlace terrestre ou de satélite. Em um sistema de telemetria podem ser coletados inúmeros tipos de dados, como a leitura de RPM (rotações por minuto), para indicação do contágio do veículo, leitura de odômetro (FENG, 2016) e etc. Por isso, esse processo é muito utilizado para testes em protótipos de carros e aeronaves, nos quais os dados são monitorados em tempo real (DOWNLING; LANCASTER, 1985).

Devido à grande quantidade de informações que podem ser coletadas em um sistema de telemetria, é necessário que essas sejam pré-processadas para melhorar a eficiência de transmissão. Em contrapartida, existiria uma ineficiência, caso cada medição utilizasse um canal de comunicação. Por conta disso, o sistema de telemetria envolve um agrupamento dos dados para serem transmitidos em um *stream*, possibilitando, assim, utilizar apenas um canal de transmissão (DOMINGOS, 2016).

2.1.2 Panorama de um sistema de Telemetria

O sistema de telemetria que foi estudado é próprio para a aquisição de dados de um avião durante um ensaio de voo e pode ser dividido em duas partes: sistema embarcado no avião (*Airbone System*) e a estação em solo (*Ground Station*) (TELEMETRY-WEST, 2013). O *Airbone System* é responsável pela aquisição, modulação e transmissão dos dados durante um ensaio de voo; já a *Ground Station* é responsável pela aquisição, a demodulação e a disponibilização dos dados adquiridos durante um ensaio de voo. Esse processo está ilustrado na Fig. 2.

Figura 2: Sistema de Telemetria.



2.2 Sistema em avião

2.2.1 Aquisição de dados

A aquisição é um processo que converte uma entrada física não elétrica em uma saída elétrica (RUOCCO, 1988). Durante um ensaio de voo, existem vários tipos de sensores que medem informações, como latitude, longitude, velocidade, inclinação do avião.

Na aquisição de dados durante um ensaio de voo, as saídas dos sensores devem ser transformadas, filtradas ou modificadas para ficarem compatíveis com o próximo estágio do sistema de telemetria, que é a multiplexação. A precisão, ou seja, a relação do valor real obtido com o valor medido, pode variar com o tempo, altitude, pressão e temperatura, sendo possível realizar calibrações especificadas pelo fabricante para melhorar a eficiência da aquisição dos sensores.

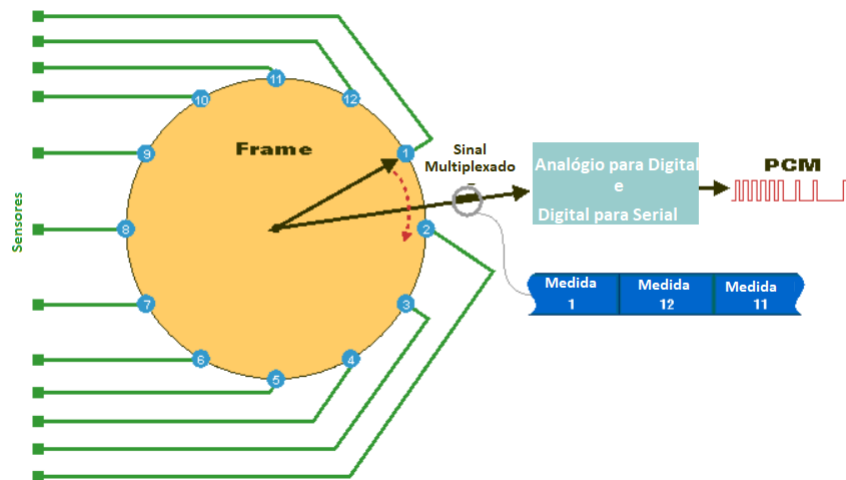
Alguns dos dados adquiridos, como latitude, longitude e altitude, que serão utilizados para o *software* desenvolvido, são transmitidos para a estação em solo para serem processados em tempo real e, além disso, todos os dados coletados são armazenados em mídias físicas que estão a bordo do avião durante o ensaio; posteriormente, os dados armazenados serão processados e disponibilizados após o fim do ensaio, quando a mídia física chegar à estação em solo.

2.2.2 Multiplexação

A multiplexação ou multiplexagem é uma técnica que consiste na combinação de dois ou mais canais de informação em um meio de transmissão. Independente da quantidade de dados a serem monitorados, a complexidade que o sistema de transmissão teria para enviar cada dado, separadamente, seria alta. Isso porque, caso não houvesse a multiplexação, cada canal deveria ser transmitido individualmente, causando, assim, um alto processamento desses dados.

Como há muitos dados, os sistemas de telemetria utilizam multiplexação de sinais e, para o sistema de telemetria em aviações, usa-se o *Time-Division Multiplexing* (TDM). Segundo (COUCH, 2012), TDM é o tempo de intercalação de amostras de várias fontes para que suas informações possam ser transmitidas serialmente por um único canal de comunicação. A Fig. 3 mostra como a multiplexação é realizada.

Figura 3: Multiplexação de um sinal.



Fonte: Telemetry-West (2013).

Como visto na Fig. 3, o sistema é cíclico. Logo, quando todos os canais forem amostrados, a sequência reinicia com o primeiro canal. Devido ao monitoramento contínuo, as amostras devem possuir uma alta taxa de amostragem para evitar que os valores de cada medida tenham distorções. Por isso, a frequência deve ser, pelo menos, duas vezes maior que a taxa de amostragem para que o sinal seja reconstituído (taxa de Nyquist) (OGATA, 2010).

Os dados multiplexados têm dois destinos que serão descritos nos próximos itens deste trabalho. Todo sinal multiplexado é encapsulado ao seu corpo de dados, o padrão *Integrated Network-Enhanced Telemetry* (iNET-X), e esse dado é enviado para a rede interna do avião, onde será capturado em pacotes *Packet Capture* (PCAP); uma pequena e específica parte dos dados é multiplexada e é gerado um PCM *stream*, que será enviado para a estação em solo, para que os dados sejam coletados em tempo real.

2.2.3 PCM

A modulação por código de pulsos (PCM) é, essencialmente, um tipo especial de conversão analógico-digital, no qual a informação analógica instantânea é representada por uma palavra digital (*digital word*) de um fluxo de *bits* em série (*bit stream*) (BELLAMY, 1991).

Um PCM *stream* é gerado através de uma parte dos dados pré-estabelecidos e enviado para a estação em solo. Cada PCM *stream* possui um *Sub-Frame ID* em cada começo de quadro e em todos eles há um quadro de sincronização no seu fim. Esse *frame* de sincronização é importante para que o dado seja reconstruído na *Ground Station*.

2.2.4 Ethernet

Ethernet é um padrão de rede de computadores comumente usado em redes locais (LAN) (STALLINGS, 2000). Foi introduzida, comercialmente, em 1980, e padronizada pela primeira vez em 1983, como IEEE 802.3, e, desde então, tem sido aperfeiçoada para suportar taxas de *bits* mais altas e distâncias de *link* mais longas (COLLINS; BRAY, 2017).

A bordo do avião, há computadores que pré-processam os dados adquiridos nos sensores. Esses computadores estão conectados entre si por uma rede interna cabeada do avião. À medida que os dados são coletados e processados, são enviados para essa rede interna. Dentro do avião, há um processo de armazenamento em forma de PCAP e esse é salvo nas mídias físicas internas no avião.

2.2.5 IEEE 802.3

O Instituto de Engenharia Elétrica e Eletrônica (IEEE) é responsável pela padronização do controle de acesso ao meio (*MAC*) da camada física e da camada de enlace de dados da *Ethernet* com fio, ou seja, define o formato dos pacotes, as tecnologias de transmissão dos *hardwares* de rede, os modos de transmissão e as características de transmissão de uma rede *Ethernet* (IEEE, 2018).

O IEEE 802 possui diversos sub-padrões, dentre eles o IEEE 802.3, que visa facilitar a conectividade entre as máquinas de diferentes fabricantes para a administração da LAN, uma vez que há a especificação das camadas de ligação dos dados do Modelo OSI (Interconexão de Sistemas Abertos).

2.2.6 iNET-X

A padronização iNET (*integrated Network-Enhanced Telemetry*) busca atender os requisitos da rede de instrumentação de ensaios em voo (*Flight Test Instrumentation* - FTI) em relação à transmissão de dados, sincronização de tempo e administração de sistemas. Com essa padronização, há um aumento da flexibilidade no desenvolvimento de sistemas, visto que há uma compatibilidade de equipamentos de diferentes empresas (YOUNG, 2018). O iNET-X é uma versão do iNET com uma melhoria de desempenho.

O padrão iNET-X possui as seguintes características:

- a) Utilização dos padrões IEEE 802.3-2005;
- b) Um único endereço MAC;
- c) Tamanho entre 64 Bytes e 1581 Bytes;
- d) Uso de *Internet Protocol* (IP) na versão 4;
- e) Transmissão em *multicast*;
- f) Utilização de *User Datagram Protocol* (UDP) para a transmissão em tempo real

- g) Utilização do protocolo IEE 1588 *Precision Time Protocol* (PTP) para serviços de rede e de sincronização, sincronizar todos os pacotes gerados em diferentes equipamentos;
- h) *Time Stamp* (horário que o pacote foi enviado) no formato PTP; e
- i) Os pacotes transmitidos como pacote UDP/IP.

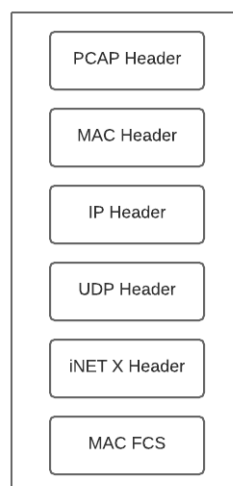
2.2.7 PCAP

Os arquivos PCAP (*Packet Capture*) são os arquivos capturados no tráfego da rede a bordo do avião. Esses arquivos são obtidos durante o ensaio de voo e salvos em mídias físicas, as quais são enviadas para a estação em solo que irá reprocessar os dados posteriormente.

Os arquivos PCAP possuem informações que permitem a leitura e localização de cada pacote de rede através de aplicativos específicos, como o *Wireshark*, que é um *software* que permite ver o fluxo de dados na rede, e gravar esse fluxo à medida que é capturado (LICENSE, 2018). Além disso, a *Microsoft* disponibiliza no sistema operacional *Windows* uma biblioteca chamada WinPcap que facilita a transmissão e recepção dos pacotes de rede (WINPCAP, 2018).

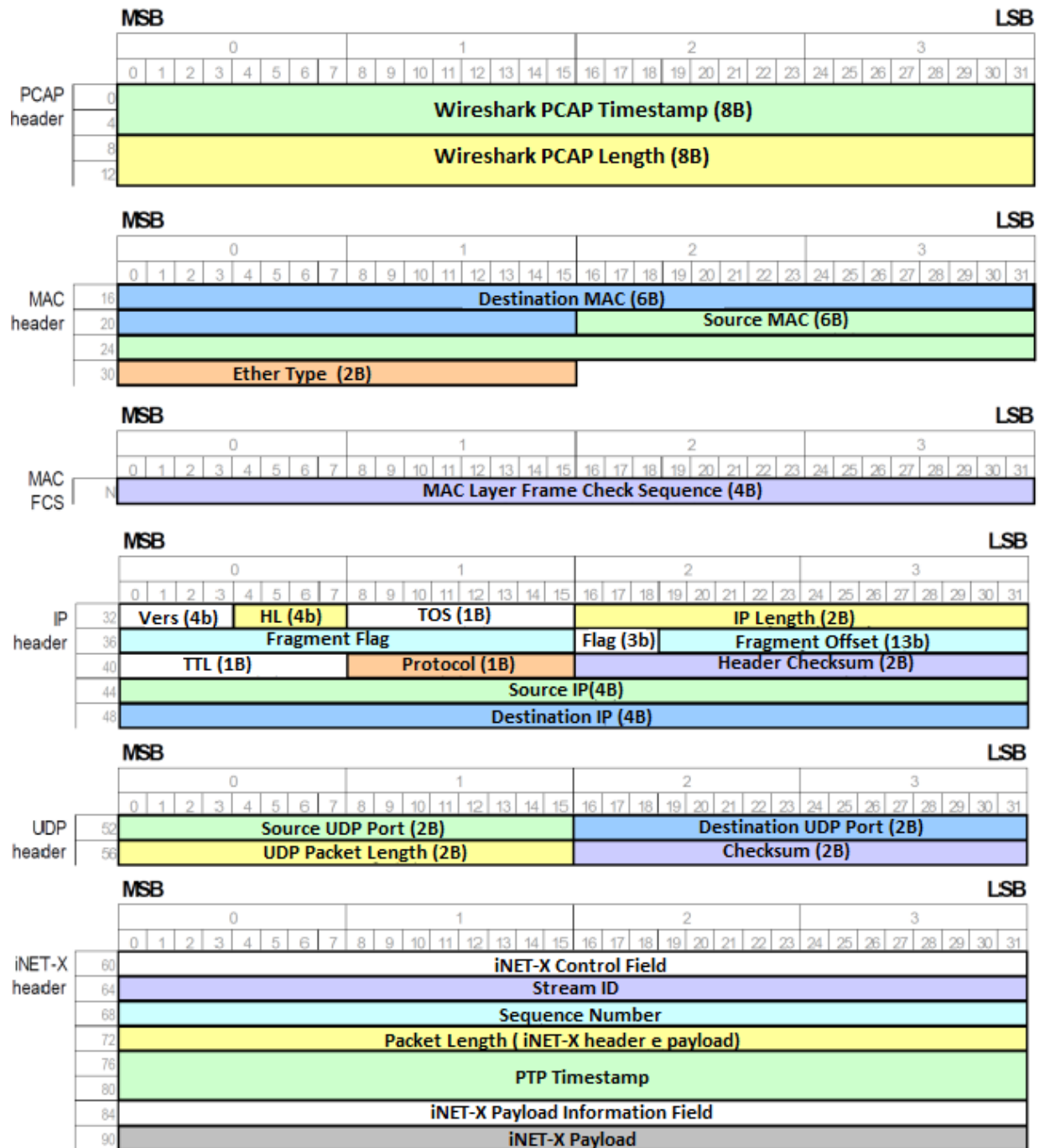
Os pacotes iNET-X são capturados durante sua transmissão em arquivos PCAP. O dado desejado está na região de dados do iNET-X, que, por sua vez, possui encapsulação dos protocolos de transmissão de rede. A Fig.4 ilustra o formato de uma estrutura de um pacote PCAP.

Figura 4: Estrutura de um pacote PCAP com arquivos iNET-X.



Nota-se que, ao final, há a presença do MAC FCS (*Frame Check Sequence*). Essa estrutura de 4 bytes é responsável pela indicação da sequência de quadros, para verificar se houve algum pacote corrompido. A Fig. 5 mostra, detalhadamente, o cabeçalho de um pacote PCAP. Com esse cabeçalho detalhado, é possível recriar o ambiente capturado dentro do avião. As informações dos dados coletados estão encapsuladas no iNET-X *Payload*.

Figura 5: Estrutura completa de um pacote de dados.



Fonte: Controls (2012).

2.3 Estação em solo

2.3.1 Setup

A estação em solo é responsável por receber os pacotes PCM, reconstruí-los e disponibilizá-los para o usuário, de forma contínua e de fácil acesso. A configuração de uma estação terrestre de telemetria inclui:

- a) Definição do sistema de aquisição de dados, incluindo as características dos sensores, bem como a geração do sinal;
- b) Definição dos quadros da telemetria, de modo a adaptar todas as taxas de amostragens, bem como limitações do *hardware* de aquisição;
- c) Calibração dos sensores antes do início do teste de voo, bem como a disponibilização dos dados com as unidades de medidas corretas;
- d) Exibir dados em tempo real para auxiliar o ensaio de voo, e
- e) Definir dados a serem arquivados no disco.

Outro papel fundamental da estação em solo é o armazenamento dos dados de um ensaio de voo. Esses dados podem ser utilizados futuramente e cabe à estação o papel de disponibilizá-los.

2.3.2 Reconstrução PCM

Na estação em solo, o sinal PCM é recebido e é necessário reconstruí-lo para obter os dados com as informações do avião. Devido à distorção que o sinal pode sofrer na transmissão, o sinal de dados PCM recebido deve ser reconstruído antes de utilizá-lo. Antes de serem distribuídos, os pulsos PCM são filtrados, de modo a arredondar suas bordas, reduzindo a largura de banda requerida na transmissão.

A primeira função do processamento de sinal tem como objetivo regenerar o sinal com um número mínimo de erro e, então, a informação do tempo de sincronização é obtida através do sinal regenerado. Esse processamento de sinal é chamado sincronização de *bits*, que é estabelecida através de uma série de pulsos de *clock* que estão sincronizados com o sinal de entrada. Essa sincronização é responsável por classificar cada valor de *bit* dentro do *stream*.

Os dados PCM só poderão ser decodificados corretamente se o tempo de *clock* do sinal for o mesmo da sincronização de *bits* no momento da codificação. Com isso, este método possui uma desvantagem: caso exista um erro de sincronização, o dado é decodificado erroneamente.

2.3.3 Processamento em tempo real

O processamento em tempo real é importante para o sistema de telemetria, uma vez que, com a análise em tempo real, é possível verificar como está a aquisição dos parâmetros durante o ensaio de voo.

Após a reconstrução do PCM, tem-se o acesso dos dados enviados pela aeronave e, assim, pode ocorrer a verificação de sua precisão. Esses dados contêm diversas medições e algumas delas serão usadas para o *software* que foi desenvolvido neste projeto. Esses dados são: latitude, longitude, altitude, inclinação, rotação e velocidade. Com esses parâmetros, pode-se fazer uma visualização em tempo real do avião durante o ensaio. Os dados reconstruídos são disponibilizados em tempo real na rede de telemetria que fica dentro da estação em solo. Esses serão os dados que serão coletados pelo *software* que foi criado.

2.3.4 Distribuição de dados

A distribuição de dados é a principal função da estação em solo. Tanto os dados adquiridos pelo sistema de telemetria quanto os salvos pelas mídias físicas a bordo precisam estar ao alcance do usuário.

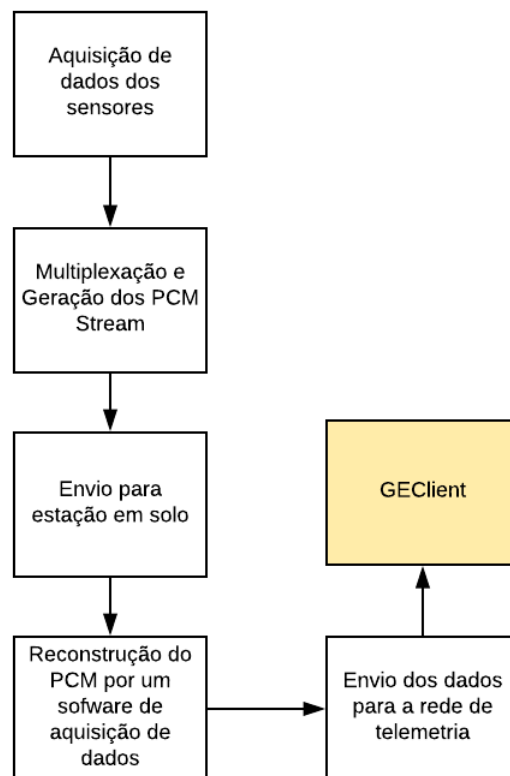
Devido ao alto volume de dados adquiridos em um ensaio, é inviável disponibilizá-los inteiramente para todos os usuários que querem adquiri-los. Por tal motivo, a estação em solo possui um sistema para a requisição desses dados, no qual são solicitados os parâmetros (tipos) dos dados desejados e, também, o período durante o ensaio. Com isso, os dados são filtrados e, finalmente, disponibilizados para o usuário.

3 METODOLOGIA

Na estação em solo, há *softwares* com três principais funções: aquisição, disponibilização e visualização de dados. O *software* que desenvolvemos é de visualização. Criamos um ambiente que facilita a visualização do avião em tempo real durante um ensaio de voo. Para isso, usamos o Google Earth como ferramenta.

A Fig.6 mostra a disposição do *software* no sistema de telemetria. Nota-se que o GEClient utiliza os dados da rede de telemetria que já foram reconstruídos por *softwares* de aquisição.

Figura 6: Representação do GEClient no sistema de telemetria.



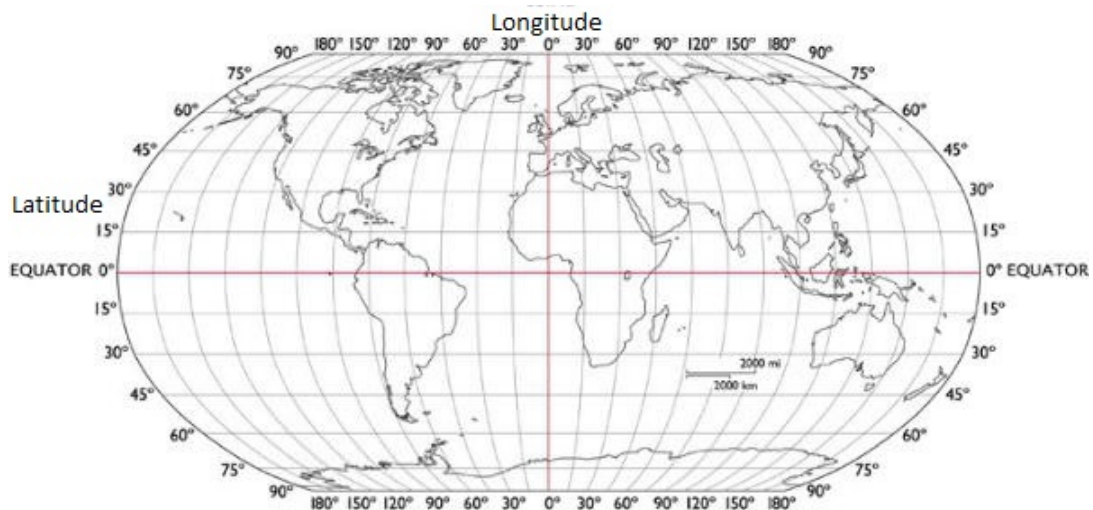
3.1 GPS

O GPS (*Global Position System*) é um sistema que determina a latitude e longitude da posição do receptor (PARKINSON; JR., 1996). A Fig. 7 mostra a latitude e longitude no globo terrestre. Nota-se que a latitude é representada numericamente duas vezes entre 0° e 90° e a longitude é representada numericamente duas vezes entre 0° e 180°. Logo, para representarmos um ponto é necessário indicar o valor numérico da latitude e se esse ponto está a norte ou a sul,

além do valor da longitude, que pode estar a leste ou a oeste (STERN, 2004). São Carlos, por exemplo, está na latitude $22,0154^\circ$ Sul e $47,8911^\circ$ Oeste.

No exemplo da posição de São Carlos, nota-se que tanto a latitude quanto a longitude possuem uma alta precisão (quatro casas decimais) e devido a essa precisão não é possível armazenar esses dados em apenas um parâmetro. Com isso, tanto a latitude quanto a longitude possuem dois parâmetros para representar seu real valor. No exemplo dado, a longitude viria separada em dois parâmetros: um deles teria o valor $47,8^\circ$ e o outro teria $0,0911^\circ$; e a longitude real seria dada pela soma desses dois parâmetros. Desse modo, é possível manter a precisão da latitude e da longitude.

Figura 7: Representação da latitude e longitude no globo terrestre.



Fonte: Phanernsoft (2014).

3.2 Google Earth

O Google Earth é um programa desenvolvido pela empresa Google e tem como principal função a visualização tridimensional do globo terrestre. Esse programa foi escolhido como ferramenta do programa de visualização por causa da facilidade de utilização. O Google Earth identifica e opera com um formato específico de arquivo e, com isso, é possível seu controle através desse arquivo, que tem formato .KML.

3.3 .KML

Keyhole Markup Language (KML) é uma linguagem baseada em XML e serve para expressar anotações geográficas e visualização de conteúdos existentes nessa linguagem, como mapas em 2D e navegadores terrestre em 3D (DEVELOPER, 2018). Para o desenvolvimento desta aplicação foram utilizados dois tipos de KML: o *NetworkLink* e o *Camera*.

3.3.1 KML *NetworkLink*

Como o arquivo KML é lido apenas uma vez pelo Google Earth, é gerado um arquivo de KML que possui uma chamada para outro arquivo KML. Esse KML é um *link* que indica para o Google Earth que o KML apontado deve ser relido constantemente. A Fig. 8 representa um arquivo de KML *NetworkLink*.

Figura 8: Exemplo KML *NetworkLink*.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Folder>
    <name>Network Links</name>
    <visibility>0</visibility>
    <description>Network link example 2</description>
    <NetworkLink>
      <name>View Centered Placemark</name>
      <visibility>0</visibility>
      <open>0</open>
      <refreshvisibility>0</refreshvisibility>
      <flyToView>0</flyToView>
      <Link>
        <href>http://yourserver.com/cgi-bin/viewCenteredPlacemark.py</href>
        <refreshInterval>2</refreshInterval>
        <viewRefreshMode>onStop</viewRefreshMode>
        <viewRefreshTime>1</viewRefreshTime>
      </Link>
    </NetworkLink>
  </Folder>
</kml>
```

Fonte: Developer (2018).

Nota-se a presença de três funções: *NetworkLink*, *href* e *ViewRefreshTime*. O *NetworkLink* é a função que indica o KML que será apontado; o *href* indica o caminho desse KML, que pode ser tanto local quanto remoto; e *ViewRefreshTime* representa o tempo em que o KML apontado deve ser atualizado (no caso do exemplo é de 2 segundos). Com testes realizados, identificamos que o menor tempo de atualização que o Google Earth opera é de 1 segundo.

3.3.2 KML Camera

Este arquivo KML exibe as informações necessárias para a representação do avião no Google Earth. Com isso, esse arquivo é constantemente atualizado pelo *software* que desenvolvemos, para mostrar em tempo real a posição do avião. A Fig. 9 representa um KML com as informações do avião que serão lidas pelo Google Earth.

A representação do modelo 3D da aeronave está descrita pela função *Model* e nota-se que, dentro dessa função, é possível colocar a posição do objeto por meio de sua latitude, longitude e altitude. Além disso, pode-se mudar a orientação do objeto em relação ao solo por meio das funções *heading*, *tilt* e *roll* (todos esses dados são adquiridos pelo sistema de telemetria em

Figura 9: Exemplo de KML Camera.

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <LookAt>
      <altitudeMode>absolute</altitudeMode>
      <longitude>-40</longitude>
      <latitude>-20</latitude>
      <range>10000</range>
      <tilt>0</tilt>
      <heading>0</heading>
    </LookAt>
    <Placemark>
      <name>Aircraft 3D Model</name>
      <Model>
        <altitudeMode>absolute</altitudeMode>
        <Location>
          <longitude>-40</longitude>
          <latitude>-20</latitude>
          <altitude>10000</altitude>
        </Location>
        <Orientation>
          <heading>-4</heading>
          <tilt>0.04</tilt>
          <roll>0.05</roll>
        </Orientation>
        <Scale>
          <x>1</x>
          <y>1</y>
          <z>1</z>
        </Scale>
        <Link>
          <href>http://MediaAircraft </href>
        </Link>
      </Model>
    </Placemark>
  </Document>
</kml>

```

Função LookAt

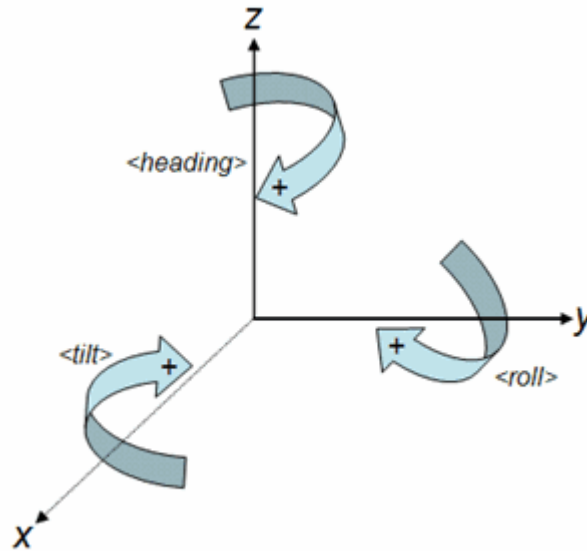
Função para orientar o modelo 3d

Função para representação 3d do avião

tempo real). A Fig. 10 representa como o *heading*, *tilt* e *roll* representam a orientação do modelo 3D. Esses valores são os valores adquiridos pelos sensores.

Outra função que aparece dentro da função *Model* é o *scale* e o *link*. A função *link* é responsável por referenciar a figura que será utilizada pela função *model*. No exemplo da Fig. 9, esta mídia está no endereço *MediaAircraft*; nesse endereço, há arquivos .dae que são representações 3D de cada aeronave. Já a função *scale*, representa a escala real do avião. Essa escala pode ser mudada pela interface do GEClient.

Figura 10: Representação da função *Orientation*.



Fonte: Developer (2018).

Além da modelagem 3D do avião, é possível ajustar a posição de visão do usuário para a visualização do avião através da função *LookAt*. A Fig. 11 demonstra como a visão do usuário é representada, utilizando as coordenadas do mapa e sua inclinação. Essa função é responsável pelo posicionamento da visão que o usuário terá no Google Earth.

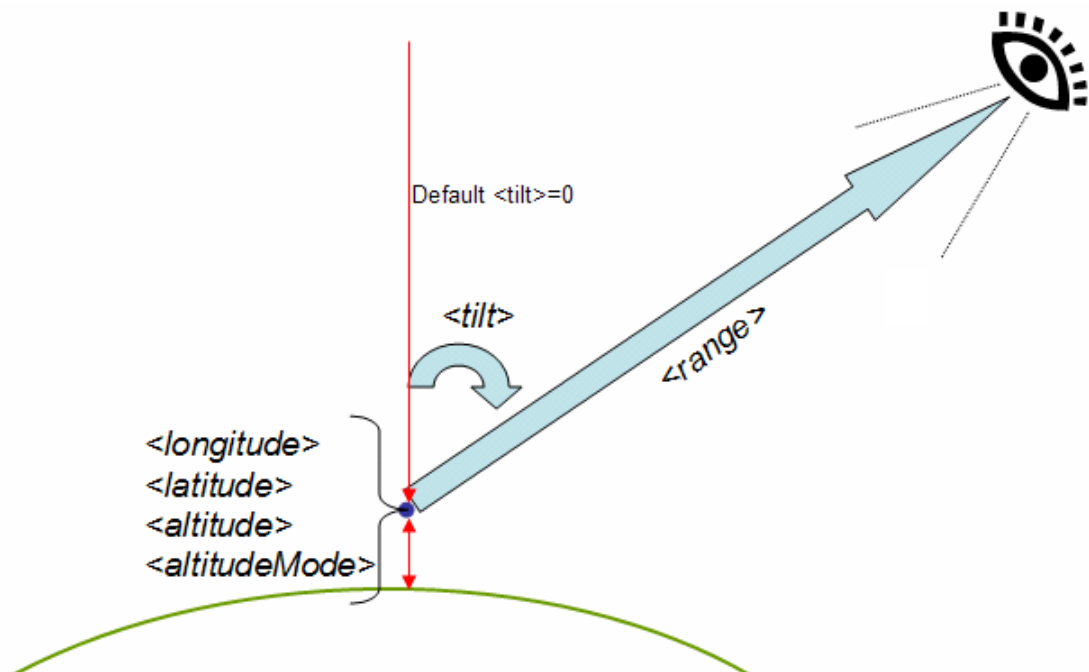
3.4 Visual Studio

O código foi desenvolvido em ambiente *Visual Studio* (VS), da *Microsoft*, em C++. Esse ambiente possui diversas ferramentas que auxiliam o desenvolvedor. O sistema de *Debug* do VS auxilia muito o desenvolvimento do projeto, pois é possível visualizar fluxo do código, bem como a atribuição para cada variável em todo o instante do programa.

Além disso, o *Visual Studio* possibilita importar diversas bibliotecas para auxiliar o projeto. Para este, foi incluída a biblioteca *WxWidgets* para a interface gráfica do programa e a biblioteca *WinCap* para a aquisição mais otimizada dos pacotes de rede.

Com o cabeçalho de um pacote *iNet-X*, é possível retirar as informações necessárias do *payload* dentre aquelas que foram multiplexadas e encapsuladas no formato *iNet-X*. Como cada canal de aquisição de dados possui um parâmetro identificador, pode-se, pela interface, filtrar os dados necessários para o funcionamento do programa (KUROSE; ROSS, 2017).

Figura 11: Representação da função *LookAt*.



Fonte: Developer (2018).

3.5 WxWidgets

WxWidgets (SMART, 2018) é uma biblioteca *open source* que permite a criação de interface gráfica para aplicações *Windows*. O ambiente de desenvolvimento *Visual Studio* possui uma ferramenta chamada MFC para desenvolvimento de interface gráfica, porém essa ferramenta é menos eficiente do que o *WxWidgets*.

Escolhemos utilizar essa biblioteca devido a sua eficiência e pela fácil transição de plataforma. A biblioteca é um pouco mais difícil de ser utilizada pela falta de visualizador gráfico para auxiliar a programação, contudo, torna-se mais fácil de ser modificada quando o *layout* estiver pré-estabelecido.

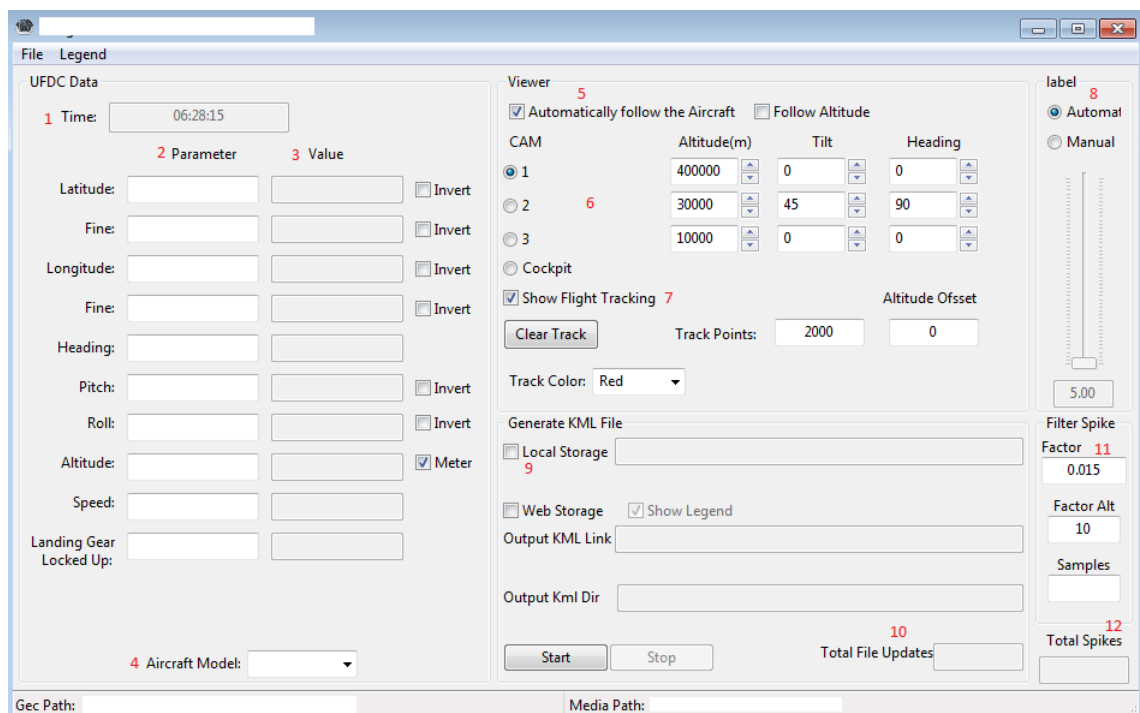
3.6 GEClient

O *GEClient* (*Google Earth Client*) é o *software* desenvolvido que utilizará os dados da rede de telemetria para gerar os arquivos KMLs que serão lidos e plotados no Google Earth. A interface do programa é muito importante para a aplicação, devido à possibilidade de alterações do arquivo KMLs através dela. Como visto, é possível mudar a posição de visão do usuário pelo Google Earth, por meio da função *LookAt* e é por meio dessa interface que o usuário faz essa modificação.

Além disso, é por meio da interface que escolhemos qual é a aeronave na qual está ocorrendo o ensaio. Cada uma das aeronaves possui um determinado parâmetro para cada informação, ou seja, cada uma das delas possui um parâmetro diferente para latitude e longitude.

O GECliente ainda é responsável por armazenar todas as posições do avião durante o período de ensaio. Esses dados são armazenados em um *buffer* interno e, caso o usuário escolha esses dados, são disponibilizados visualmente. A Fig. 12 mostra a interface desenvolvida para o *software* GEClient. Esse *software* irá gerar os arquivos KMLs que serão lidos pelo Google Earth.

Figura 12: Interface do Software desenvolvido GEClient.



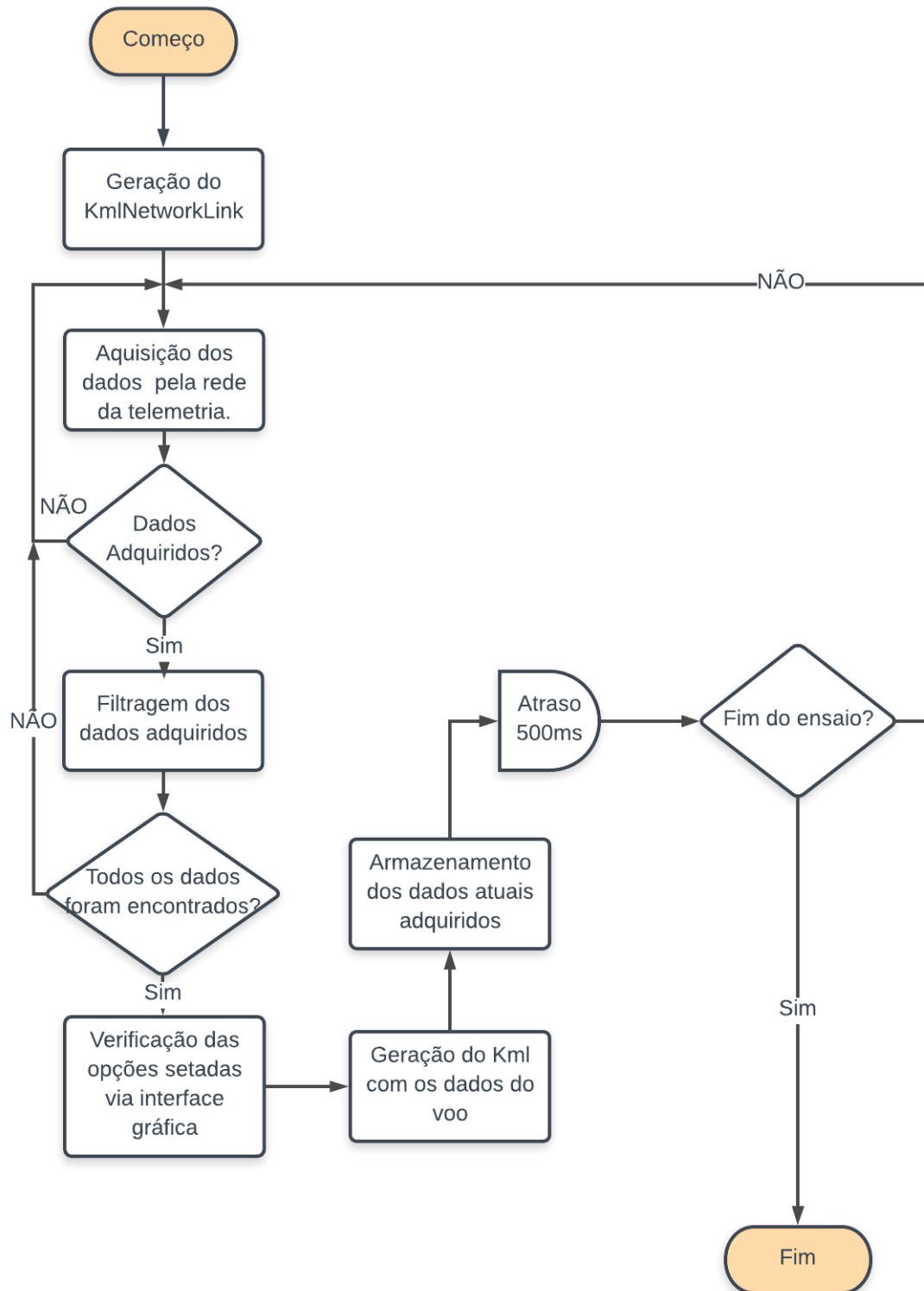
O GECliente possui as seguintes funcionalidades:

1. Tempo adquirido pelo sistema operacional a bordo do avião;
2. Parâmetros a serem filtrados. Observa-se que a latitude e a longitude possuem dois parâmetros: latitude e o *fine*. A latitude real é a soma desses dois parâmetros. Além disso, é possível verificar o botão *Invert*, que é utilizado para mudar o hemisfério, uma vez que as medições do GPS do avião não possuem distinção de hemisfério (positivo ou negativo) conforme visto na Fig. 7;
3. Valor filtrado para o parâmetro colocado;
4. A aeronave em que está acontecendo o ensaio. Cada aeronave possui uma representação 3D diferente;
5. Opção para o usuário escolher a posição de visualização no Google Earth;

6. Quando a opção do item acima está acionada, é possível variar a posição da visão conforme visto na Figura 11. Há uma opção de *cockpit* que irá representar a visão da cabine de controle do avião no Goolge Earth;
7. Opção para mostrar o trajeto do avião durante o ensaio de voo;
8. Ajusta o tamanho do avião 3D no Google Earth. Na opção automático, é realizada uma conta, dependendo do ponto de visão e da altitude do avião, a fim de tornar possível a visualização do avião;
9. Opção para a geração dos arquivos KMLs. Futuramente o GEClient irá gerar uma KML que será lida remotamente;
10. Total de *updates* que o arquivo KML sofreu durante a execução do programa;
11. Opção para filtrar a aquisição de dados. Caso a variação de uma aquisição for menor que os valores descritos nessa opção, o programa irá desconsiderar o novo valor obtido; e
12. Número de vezes que o filtro funcionou.

O fluxograma da Fig. 13 demonstra como é feita a aquisição dos dados e a geração do KML que será lido pelo Google Earth.

Figura 13: Fluxograma do GEClient.

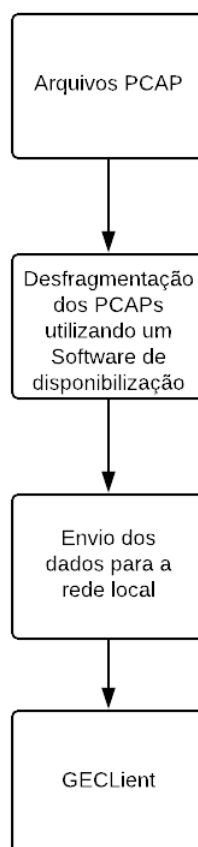


4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

O *software* desenvolvido atendeu às expectativas com relação à visualização da posição do avião em tempo real. O GEClient foi testado em diversos ensaios de voo e apresentou ótimo desempenho. Nas simulações, realizamos testes de horas de voo e o programa conseguiu capturar todos os pontos durante esse tempo. O programa ficou leve, utilizando de pouco mais de 40 MB de memória RAM, o GEClient teve um bom desempenho.

Para a validação do *software*, foi desenvolvido um sistema que simula um ambiente de telemetria durante um ensaio de voo. Para simular este ambiente, foram utilizados dados de ensaios anteriores que foram armazenados em formato PCAP. Esses dados são desfragmentados e os pacotes iNET-X inseridos em uma rede local do mesmo jeito que foi capturado a bordo do avião, simulando os pacotes iNET-X que são recebidos pelo sistema de telemetria durante um ensaio. Essa simulação está exemplificada na Fig 14.

Figura 14: Representação de uma simulação de ensaio usando rede local.



Com esse ambiente de simulação foi possível verificar se os valores obtidos pelo GEClient

estavam coerentes. Para realizar essa verificação foi analisado o valor mostrado *value*, da Fig. 12, para cada parâmetro e comparado com a aquisição dos dados por um *software* de visualização que é utilizado para visualização de dados.

Verificado a aquisição de dados foi possível testar a interação GEClient e Google Earth. Com isso, variou-se os valores da atitude, *tilt* e *heading* e verificou-se o comportamento do Google Earth. Com isso, existe uma variação da função *LookAt* o que implica em uma variação da posição de visão do usuário. As Figuras 15, 16 e 17 representam algumas dessas variações via GEClient. Já a simulação da Fig. 18 representa a visão que o piloto possui durante o ensaio.

Figura 15: Simulação para uma altitude de visualização de 400 mil metros.



As Figuras 15, 16, 17 e 18 representam a imagem de um avião durante um ensaio de voo pelo Google Earth. Nestas figuras, o arquivo KML *NetWorklink* já foi lido e este fica indicando para o Google Earth que o arquivo KML Camera seja relido em certos intervalos de tempo. O arquivo KML Camera detém as informações necessárias para a visualização e os resultados são os obtidos nas figuras. Nota-se que as Figuras 15 e 16 não possuem ângulo de inclinação e a única diferença é a distância da visão em relação ao solo (400 mil metros e 50 mil metros, respectivamente). Já a simulação da Fig. 17 possui as mesmas características de visão da Fig. 16 mas com a diferença que foi selecionado um *tilt* = 45° e um *heading* = 315° o que deixa o ângulo de visão um pouco inclinado sobre o avião.

Figura 16: Simulação para uma altitude de visualização de 50 mil metros com $Tilt = 0^\circ$ e $Heading = 0^\circ$.



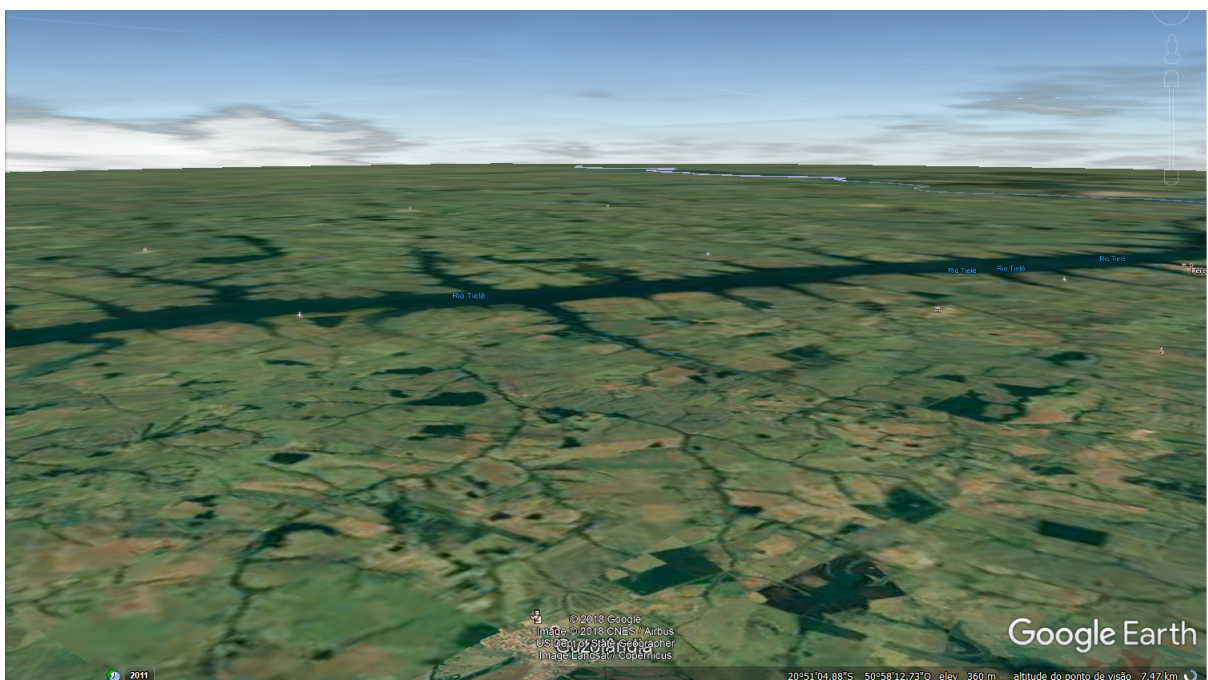
Durante os testes, o GECliente conseguiu ler e salvar mais de 60 mil pontos, sem que o Google demorasse para ler e o GECliente tivesse problema ao escrevê-lo. Com o desenvolvimento do projeto, verificamos duas propriedades que o programa Google Earth possui. A primeira é a limitação da atualização para a releitura do arquivo KML. Com os testes feitos, foi possível verificar que a releitura era de no mínimo um segundo, ou seja, a posição do avião só seria atualizada a cada 1 segundo, o que é um tempo aceitável, visto que há pouca variação da posição do avião durante esse intervalo.

Além disso, foi percebido que o Google Earth restringe a modificação do arquivo enquanto ele estava sendo lido, o que acarretou em diversos erros de leitura e escrita. Para resolver isso, o arquivo KML foi gerado em baixo nível, estabelecendo uma propriedade, para que o arquivo fosse denominado compartilhado; desse modo, não houve mais problemas de escrita ou leitura do arquivo.

Figura 17: Simulação para uma altitude de visualização de 50 mil metros com $Tilt = 45^\circ$ e $Heading = 315^\circ$.



Figura 18: Simulação visão *Cockpit*.



5 CONCLUSÃO E TRABALHOS FUTUROS

O sistema de telemetria aviônica é bastante complexo e de extrema importância para o processo de testes de novos produtos, uma vez que é responsável por todos os dados adquiridos durante um ensaio de voo. O sistema é dividido em dois principais sistemas: o sistema a bordo, que é responsável pela aquisição e armazenagem dos dados de voo, e a estação em solo, que é responsável pela disponibilização dos dados de voo.

O desenvolvimento do *software* GEClient atendeu às expectativas esperadas. Pode-se chegar a essa conclusão graças aos resultados apresentados, que mostram que o GEClient lê e salva as informações de uma forma rápida e sem dificuldades, sem que o Google Earth tenha problemas de leitura nos pontos. Com relação a isso, entendeu-se que o Google Earth apresentou uma limitação de atualização de um segundo para a leitura de arquivos .KML, além de restringir a modificação do arquivo enquanto ele estava sendo lido.

Atualmente, o GEClient está em produção, auxiliando durante os ensaios de voo em uma empresa no ramo aeronáutico, porém, futuramente, será possível realizar a visualização remota do avião durante um ensaio de voo, o que, hoje, só é possível nos computadores conectados à rede da telemetria. Há um projeto futuro que irá permitir o acesso remoto desses dados por meio de um *link* URL, possibilitando a visualização do ensaio em tempo real fora da rede de telemetria.

REFERÊNCIAS

- BELLAMY, J. **Digital Telephony**. Coppel,TX: John Wiley & Sons, 1991.
- C.M.GEVAERT et al. Informal settlement classification using point-cloud and image-based features from uav datag. **ISPRS Journal of Photogrammetry and Remote Sensing**, 2017.
- COLLINS, C. J.; BRAY, J. R. In situ crosstalk measurements of long cables – the multi-network analyzer method. **2017 IEEE International Symposium on Electromagnetic Compatibility Signal/Power Integrity (EMCSI)**, 2017.
- CONTROLS, C. W. Technical note 50. **Curtiss Wright Controls - Avionics Electronics**, 2012.
- COUCH, L. W. **Digital And Analog Communication Systems**. 8. ed. Boston: Pearson, 2012.
- DEVELOPER, G. **KML develop**. 2018. Acesso em 20 set. 2018. Disponível em: <<https://developers.google.com/kml/documentatio/kmlreference>>.
- DOMINGOS, H. B. R. High performance multi-vendor network data acquisition platform. **36th European Telemetry and Test Conference**, 2016.
- DOWNLING, D.; LANCASTER, R. A. Remote aircraft maintenance monitoring using telemetry. **International Telemetry Conference Proceedings**, International Foundation for Telemetry, 1985. Disponível em: <<https://repository.arizona.edu/handle/10150/615737>>.
- FENG, D. L. e. K. L. T. The research of smart ex-telemetry system of launch vehilce. **36th European Telemetry and Test Conference**, 2016.
- IEEE. **IEEE**. 2018. Acesso em 22 set. 2018. Disponível em: <<https://www.ieee.org/>>.
- KERR, J.; MCELROY, C. Measurement of stratospheric nitrogen dioxide from the aes stratospheric balloon program. **Atmosphere**, 1976.
- KUMAR, A. et al. Wireless telemetry system for real-time estimation of ship air wakes with uavs. **Mechatronics**, 2016.
- KUROSE, J. F.; ROSS, K. W. **Computer Networking: A Top-Down Approach**. 7. ed. Boston: Pearson, 2017.
- LICENSE, G. G. P. **Wireshark**. 2018. Acesso em 22 set. 2018. Disponível em: <<https://www.wireshark.org/>>.
- MODONIA, G. E.; SACCOB, M.; TERKAJ, W. A telemetry-driven approach to simulate data-intensive manufacturing processes. **49th CIRP Conference on Manufacturing Systems (CIRP-CMS 2016)**, 2016.
- OGATA, K. **Engenharia de Controle Moderno**. 5. ed. São Paulo: Prentice Hall, 2010.
- PARKINSON, B. W.; JR., J. J. S. **Global Positioning System: Theory and Applications, Volume II**. Washington: American Institute of Aeronautics and Astronautics, 1996.

PHANERNISOFT. **Latitude e Longitude**. 2014. Acesso em 28 set. 2018. Disponível em: <<http://www.phaneronsoft.com.br/2014/06/26/converter-latitude-e-longitude-para-dms-decimal-to-degrees-minutes-seconds/>>.

PROJECTCARSGAME. **F1 Telemetry**. 2012. Acesso em 1 nov. 2018. Disponível em: <<http://forum.projectcarsgame.com/>>.

ROLANDO, A.; MINICHINI, L.; ROSSI, F. An ieee 802.11 based telemetry system for ultralight machines flight testing. **etc2016 - 36. European Telemetry and Test Conference**, 2016.

ROLDAN, J. J. et al. Determining mission evolution through uav telemetry by using decision trees. **Conference: Conference: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**, 2016.

RUOCCO, S. **Robot sensors and transducers**. Berlin: Springer Science & Business Media, 1988.

SMART, J. **WxWidgets**. 2018. Acesso em 21 set. 2018. Disponível em: <<https://www.wxwidgets.org>>.

STALLINGS, W. **Local and Metropolitan Area Networks**. 6. ed. New Jersey: Prentice-Hall, 2000.

STERN, D. P. **Latitude and Longitude**. 2004. Acesso em 28 set. 2018. Disponível em: <<https://www-istp.gsfc.nasa.gov/stargaze/Slatlong.htm>>.

TELEMETRY-WEST, L. C. **Telemetry Tutorial**. L-3 Communications Telemetry-West, 2013. Disponível em: <<https://www.yumpu.com/en/document/view/10390099/telemetry-tutorial-l-3-communications>>.

WINPCAP. **WinPcap**. 2018. Acesso em 22 set. 2018. Disponível em: <<https://www.winpcap.org/>>.

WINTERNITZ, L. M. B. et al. Sextant x-ray pulsar navigation demonstration: Flight system and test results. **2016 IEEE Aerospace Conference**, 2016.

YOUNG, T. Integrated network enhanced telemetry(inet): Impact to the telemetry community for the ettc 2018. **European Test and Telemetry Conference**, 2018. Disponível em: <<https://repository.arizona.edu/handle/10150/615737>>.