

UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO

DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS

ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Igor de Oliveira Silvestre

Infraestrutura de comunicação via radiofrequência em um aerogerador com aerofólio cabeado

[Florianópolis]

[2020]

Igor de Oliveira Silvestre

Infraestrutura de comunicação via radiofrequência em um aerogerador com aerofólio cabeado

Trabalho Conclusão do Curso de Graduação em Engenharia de Controle e Automação do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia de Controle e Automação

Orientador: Prof. Dr. Marcelo De Lellis Costa de Oliveira

Florianópolis

2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Silvestre, Igor

Infraestrutura de comunicação via radiofrequência em um
aerogerador com aerofólio cabeado / Igor Silvestre ;
orientador, Marcelo De Lellis Costa de Oliveira, 2020.

71 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia de Controle e Automação,
Florianópolis, 2020.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Energia eólica
com aerofólios cabeados. 3. Comunicação via radiofrequência.
4. Protocolos de comunicação. I. De Lellis Costa de
Oliveira, Marcelo. II. Universidade Federal de Santa
Catarina. Graduação em Engenharia de Controle e Automação.
III. Título.

Igor de Oliveira Silvestre

Infraestrutura de comunicação via radiofrequência em um aerogerador com aerofólio cabeado

Esta monografia foi julgada no contexto da disciplina DAS5511: Projeto de Fim de Curso e aprovada na sua forma final pelo Curso de Engenharia de Controle e Automação.

Florianópolis, 27 de janeiro de 2020.

Prof. Leandro Buss Becker, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Marcelo De Lellis Costa de Oliveira, Dr.
Orientador

Prof. Werner Kraus Junior, Dr.
Avaliador

Prof. Marcelo De Lellis Costa de Oliveira, Dr.
Presidente da banca

Márcio Sumariva Nandi,
Debatedor

Henrique Morales Busiquia,
Debatedor

Este trabalho é dedicado à minha família, Margareth, Marcio e Iago, que sempre estiveram ao meu lado.

AGRADECIMENTOS

A esta universidade e seu corpo docente, que me deram as oportunidades e os meios para eu realizar um sonho pessoal.

Ao meu orientador, pela oportunidade, suporte, confiança e sobretudo respeito.

Aos meus pais, pelo incentivo em persistir e apoio incondicional.

A minha vó, pelo acolhimento, ensino e carinho.

Ao meu irmão, pelo auxílio intelectual e emocional, por padecer de minhas frustrações e me incentivar diariamente.

“Those who can imagine anything, can create the impossible.”

(Alan Turing)

Resumo

Pioneiro na América Latina, o grupo de pesquisa UFSCkite pretende desenvolver a geração de energia via aerofólios cabeados em âmbito nacional e em um futuro próximo inserir na matriz energética esta tecnologia desenvolvida, assim tornando a energia eólica viável em localidades nas quais atualmente ela não é. O grupo UFSCkite atualmente está realizando testes com a unidade de voo na configuração *hardware-in-the-loop* e para permitir a operação do aerogerador devemos estabelecer um *link* de comunicação sem fio entre a unidade de solo e a de voo para realizar a troca de dados importantes para a dinâmica do aerogerador, como por exemplo a configuração de parâmetros do controlador e as variáveis de sensores da unidade de voo. A transferência de dados já vinha sendo abordada pelo grupo UFSCkite onde um sistema de levantamento do perfil de vento vem sendo desenvolvido. Neste sistema a radiofrequência serviu como *link* de comunicação utilizando módulos *swarm bee* da Nanotron e portanto naturalmente esta estrutura foi utilizada durante esta monografia para desenvolver um protocolo de comunicação. Para o desenvolvimento deste trabalho foi realizado a instrumentação, definição dos pacotes que devem ser trocados entre unidade de voo e solo, elaboração do protocolo e finalmente a realização de testes para avaliar o que foi desenvolvido. Com a realização de testes foi averiguado os tempos médios de ciclo de mensagem, verificado se transporte via radiofrequência de pacotes informados via *Publish-Subscribe* funciona de forma adequada e analisada a perda de pacotes envolvidas no protocolo de comunicação. A principal contribuição deste trabalho para o grupo UFSCkite é o protocolo de comunicação em si, que possibilita a realização de testes com a unidade de voo acompanhando o aerofólio cabeado.

Palavras-chave: Energia eólica com aerofólios cabeados. Comunicação via radiofrequência. Protocolos de comunicação.

Abstract

South America's pioneer, the UFSCkite research group intends on developing airborne wind energy on national ground and insert the developed technology into Brazil's energy grid in the near future, thus making wind energy viable in locations where it is not currently. The UFSCkite group is currently conducting tests with the flight unit in a hardware-in-the-loop configuration and to allow the airborne wind turbine to operate we must establish a communication link between the ground unit and the flight unit so we can transfer important data needed for the generator's dynamic, like for example the parameters for the controller and the flight's unit sensors variables. Data transmission was already being addressed by the UFSCkite group where a system to establish a wind profile is being developed. On this system radio frequency served as the communication link by utilizing swarm bee modules by Nanotron and therefore it is natural that this structure was utilized to develop a communication protocol. For the development of this work it was performed the instrumentation, definition of packages to be exchanged between flight and ground units, preparation of the protocol and finally tests to evaluate what was developed were carried out. With the performance of tests, the average message cycle times were verified. It was checked if the radio frequency communication of packages informed via Publish-Subscribe works properly and the loss of packages involved in the communication protocol was analyzed. The main contribution of this work for the UFSCkite group is the communication protocol itself, which makes it possible to carry out tests with the flight unit following the wired airfoil.

Key-words: Airborne wind energy. Radiofrequency communication. Communication protocols.

Lista de ilustrações

Figura 1 – Velocidade média do vento no Brasil (a 100m do nível do solo)	17
Figura 2 – Grupos de pesquisa AWE pelo mundo	18
Figura 3 – Geração eólica convencional, AWE com geração em voo e em solo . . .	19
Figura 4 – Esquemático do gerador UFSCkite	20
Figura 5 – Modelo simplificado de comunicação via radiofrequência terrestre . . .	23
Figura 6 – Tipos de propagação de ondas de rádio	28
Figura 7 – Reflexão e refração	29
Figura 8 – Difração	30
Figura 9 – Dispersão	30
Figura 10 – Interferência	31
Figura 11 – Efeito Doppler	32
Figura 12 – Múltiplo acesso por FDMA, TDMA e CDMA	34
Figura 13 – Sistema de comunicação via RF proposta por MAURER	36
Figura 14 – BeagleCore	37
Figura 15 – Raspberry Pi Zero W	37
Figura 16 – <i>Swarm bee</i> DK plus e LE, respectivamente	38
Figura 17 – Comando SDAT como exemplo de uso da API dos <i>swarm bees</i>	38
Figura 18 – Estrutura modular de software da unidade de voo	39
Figura 19 – Exemplo pacote de dados LCM	40
Figura 20 – Diagrama de representação da comunicação geral do sistema	42
Figura 21 – Sistema supervisorio UFSCkite	43
Figura 22 – Modelo de ponto de massa do aerofólio em coordenadas polares e car- tesianas	45
Figura 23 – Modelo de quatro cabos e variáveis <i>depower</i> e <i>steering</i>	46
Figura 24 – Máquina de estados do protocolo de comunicação	47
Figura 25 – Diagrama de sequência UML para o protocolo de comunicação da uni- dade de solo	48
Figura 26 – Esboço inicial dos pacotes de dados LCM	49
Figura 27 – Versão final dos pacotes de dados LCM	50
Figura 28 – <i>Frame</i> de 65 bytes do pacote “fu_to_gu”	51
Figura 29 – <i>Frame</i> de 123 bytes do pacote “gu_to_fu”	51
Figura 30 – Configuração da porta serial para comunicação USART	53
Figura 31 – Configuração dos pinos A_MODE e MOD_EN	54
Figura 32 – Esboço da implementação do protocolo de comunicação na unidade de solo	55

Figura 33 – Implementação simplificada do protocolo de comunicação na unidade de solo	57
Figura 34 – Implementação do módulo “newradio” da estrutura de software da GU	59
Figura 35 – Tempos de ciclo de mensagem	60
Figura 36 – Distribuição normal do tempo de ciclo de mensagem	61
Figura 37 – Tempos de ciclo de mensagem para pacotes de 10 e 128 bytes, respectivamente	62
Figura 38 – Distribuição normal do tempo de ciclo de mensagem para pacotes de 10 bytes	62
Figura 39 – Configuração para testes da comunicação LCM na unidade de solo . . .	63
Figura 40 – Confirmação da comunicação <i>Publish-Subscribe</i> de Python para C . . .	64
Figura 41 – Confirmação da comunicação <i>Publish-Subscribe</i> de C para Python . . .	65
Figura 42 – Confirmação do transporte de dados entre unidade de solo e de voo . .	66
Figura 43 – Corrupção dos pacotes	67
Figura 44 – Perda de pacotes por <i>timeout</i>	68

Lista de tabelas

Tabela 1 – Capacidade mundial instalada	16
Tabela 2 – Características e aplicações das frequências de rádio	26
Tabela 3 – Geração de ligações específicas do idioma com a biblioteca LCM	41

Lista de abreviaturas e siglas

RF	Radiofrequência
GU	Ground Unit
FU	Flight Unit
AWE	Airborne Wind Energy
WLAN	Wireless Local Area Network
AM	Amplitude Modulation
FM	Frequency Modulation
SHF	Super High Frequency
EHF	Extremely High Frequency
FDMA	Frequency Division Multiple Access
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
LCM	Lightweight Communications and Marshalling
SCADA	Supervisory Control and Data Acquisition
OPC	Open Platform Communications
USART	Universal Synchronous Asynchronous Receiver Transmitter

Sumário

1	INTRODUÇÃO	16
1.1	Objetivo e metodologia	20
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	Histórico das redes de comunicação via RF	22
2.2	Arquitetura do sistema de radiofrequência	23
2.3	Classificação do espectro de rádio	24
2.4	Características das ondas de rádio	27
2.5	Efeitos de propagação das ondas de rádio	28
2.6	Técnicas de comunicação via radiofrequência	32
3	INFRAESTRUTURA DE HARDWARE E SOFTWARE	35
3.1	Trabalhos correlatos	35
3.2	Placas de desenvolvimento	36
3.3	Módulos <i>swarm bee</i> de radiofrequência	37
3.4	Estrutura de software	39
3.5	Biblioteca LCM	40
3.6	Linguagens de programação utilizadas	41
3.7	Supervisório SCADA	42
4	DESENVOLVIMENTO DO PROTOCOLO DE COMUNICAÇÃO	44
4.1	Hardware-in-the-loop	44
4.2	Modelagem do protocolo	46
4.3	Elaboração do pacote de dados	48
4.4	Configuração do módulo <i>swarm bee</i> em Python	52
4.5	Implementação do protocolo em Python	54
4.6	Implementação final do protocolo	56
4.7	Novo módulo “newradio”	58
5	RESULTADOS	60
5.1	Tempo de ciclo de mensagem	60
5.1.1	Distribuição normal do tempo de ciclo de mensagem	61
5.1.2	Variação de acordo com tamanho da mensagem	61
5.2	Comunicação LCM	63
5.3	Transporte dos dados	65
5.4	Perda de pacotes	66

6	CONCLUSÕES E PERSPECTIVAS	69
	REFERÊNCIAS	70

1 Introdução

A maioria da energia consumida pelo mundo até hoje é produto de fontes fósseis (sendo o Brasil uma exceção, com suas hidrelétricas), que atualmente estão enfrentando problemas de segurança, viabilidade econômica e sustentabilidade. Para enfrentar esses problemas diversos países estão decretando políticas energéticas com o intuito de aumentar a participação de energias renováveis na matriz energética mundial. No âmbito dessas políticas, destacam-se as seguintes medidas (CHERUBINI et al., 2005).

- Em setembro de 2009, a União Europeia e os líderes do G8 concordaram que as emissões de carbono devem ser cortadas em 80% até 2050, de acordo com a Comissão Europeia (2011).
- Protocolo de Kyoto, segundo o qual um dos compromissos de 15 membros da União Europeia é reduzir suas emissões coletivas em 8% no período entre 2008-2012.
- O pacote de legislações de clima e energia para 2020, que obriga os membros da União Europeia a reduzir suas emissões em, no mínimo, 20% até 2020, de acordo com a Comissão Europeia (2007).

Nesse contexto, a energia eólica vem crescendo continuamente no mundo como fonte renovável, como é ilustrado na Tabela 1:

Tabela 1 – Capacidade mundial instalada

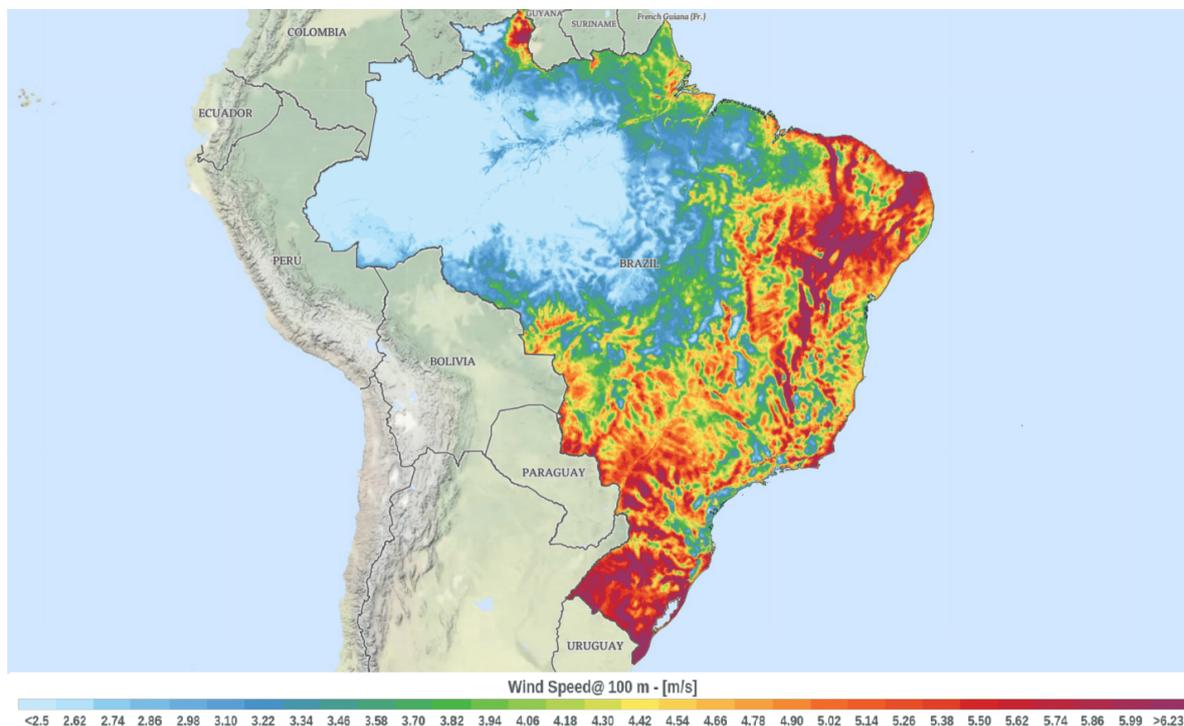
Country/Region1	2018	2017	2016	2015
China	216'870	195'730	168'730	148'000
United States	96'363	88'775	82'033	73'867
Germany	59'313	56'190	50'019	45'192
India**	35'017	32'879	28'279	24'759
Spain*	23'494	23'026	23'020	22'987
United Kingdom	20'743	17'852	14'512	13'614
France	15'313	13'760	12'065	10'293
Brazil**	14'490	12'763	10'800	8'715
Canada	12'816	12'239	11'898	11'205
Rest of the World*	102'138	93'173	85'582	76'653
Total geral	596'556	546'388	486'939	435'284

Fonte: WWEA (2018)

Ocupando o oitavo lugar, é visível que o Brasil vem ao longo dos últimos anos aumentando sua capacidade instalada. Entretanto também é visível que ainda existe um

longo caminho para aproveitar o potencial eólico brasileiro, principalmente no sul e nordeste do país, como se pode ver na Figura 1.

Figura 1 – Velocidade média do vento no Brasil (a 100m do nível do solo)



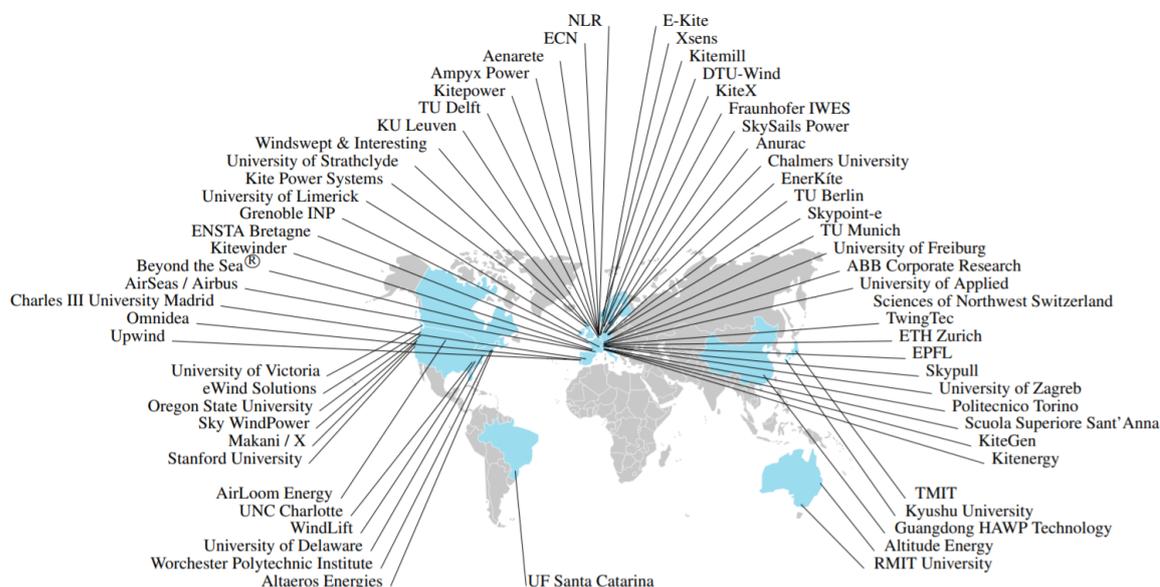
Fonte: Global Wind Atlas (2020)

Visando expandir ainda mais o aproveitamento da energia eólica, permitindo a sua exploração em localidades nas quais atualmente ela não é viável, a tecnologia que utiliza aerofólios cabeados, em inglês *Airborne Wind Energy (AWE)*, vem sendo o foco de diversas pesquisas ao redor do mundo. A ideia principal é a de substituir as pás de uma turbina eólica convencional por um aerofólio que tem capacidade de permanecer no ar por forças aerodinâmicas ou aerostáticas. Tal aerofólio é preso por um ou mais cabos ao solo de forma a substituir a torre de uma turbina convencional. Com estas substituições, estima-se que seja possível gerar energia elétrica com um custo menor e por meio de um sistema de geração que apresenta vantagens sobre os atuais aerogeradores. Entre essas vantagens, destacam-se:

- a substituição da torre por um ou mais cabos permite o aerofólio operar em maiores altitudes e portanto interceptar ventos mais fortes e consistentes;
- a tecnologia AWE reduz substancialmente os custos para se construir um aerogerador, especialmente de material, transporte e instalação.

Atualmente existem mais de 55 grupos de pesquisa em AWE ao redor do mundo. Cada um desses grupos utiliza um modelo de aerogerador, com suas vantagens e desvantagens. Uma grande parte destes grupos pode ser visualizada na Figura 2.

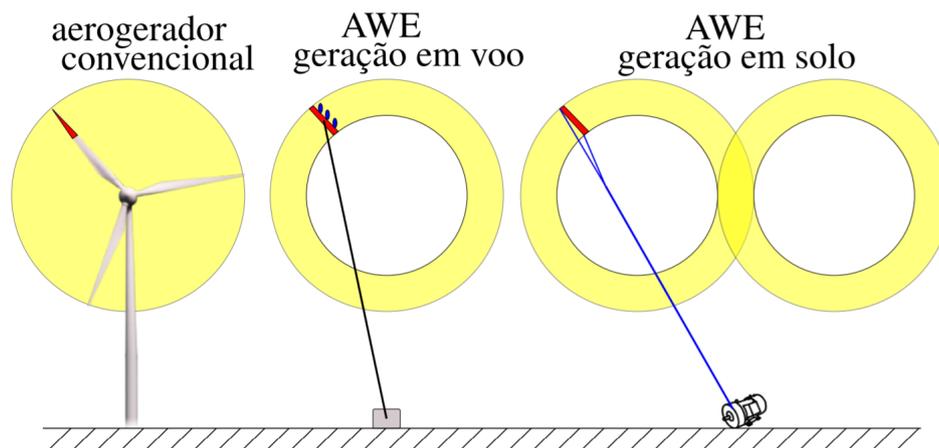
Figura 2 – Grupos de pesquisa AWE pelo mundo



Fonte: (SCHMEHL, 2018).

Existem duas configurações básicas de geração com tecnologia AWE, ilustradas na Figura 3 ao lado da tecnologia eólica convencional. Uma delas é a geração em voo, em que são montadas turbinas elétricas leves e pequenas ao longo de aerofólio rígido (asa). Este realiza uma trajetória de voo similar a de uma pá de um aerogerador convencional (turbina de eixo horizontal montada sobre torre). Já na outra variante a geração é realizada em solo, segundo a qual a energia é gerada por meio do desenrolamento de um cabo preso, em uma ponta, ao aerofólio e, na outra, a uma máquina elétrica. Neste tipo de geração o aerofólio realiza a trajetória de uma lemniscata (“oito deitado”) para garantir maior tração possível sem acumular torção no cabo.

Figura 3 – Geração eólica convencional, AWE com geração em voo e em solo



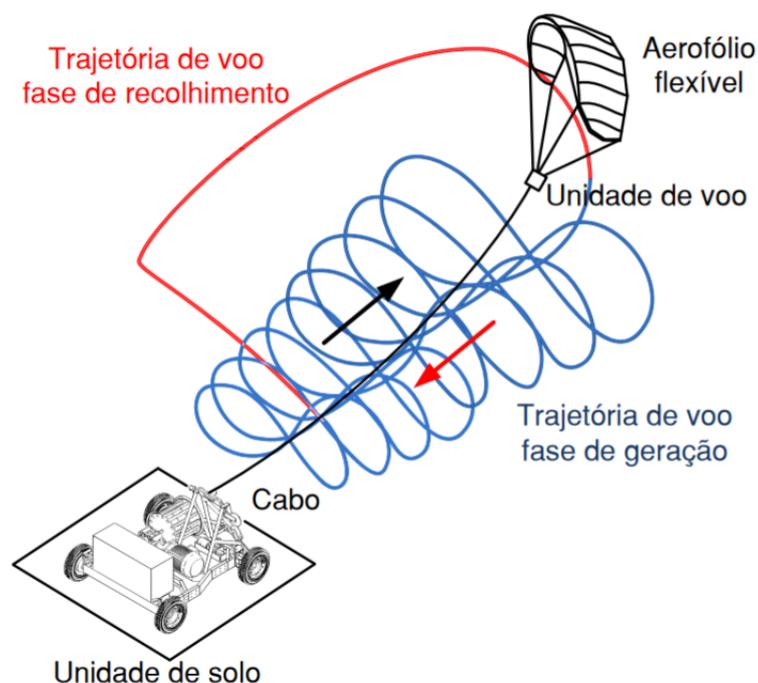
Fonte: (OLIVEIRA, 2016).

Pioneiro na América Latina, o grupo de pesquisa UFSCkite, sediado na Universidade Federal de Santa Catarina em Florianópolis, busca desenvolver a geração de energia via aerofólios cabeados em âmbito nacional. O grupo optou por desenvolver um aerogerador com a tecnologia AWE realizando a geração de energia em solo, principalmente porque essa configuração apresenta um custo de desenvolvimento bastante inferior ao da geração em voo.

Aerogeradores AWE com geração no solo, conforme Figura 4, operam em um ciclo composto por duas fases, descritas a seguir.

- Fase de **geração** de energia. O aerofólio puxa e desenrola o cabo de um carretel acoplado a um gerador até que um certo comprimento do cabo tenha sido desenrolado. O carretel e gerador localizam-se na chamada unidade de solo.
- Fase de **recolhimento** do aerofólio. Com a fase anterior completa, um servomotor abrigado na chamada unidade de voo, suspensa próximo ao aerofólio, atua em um cabo de comando que reconfigura o aerofólio para uma condição de baixa eficiência aerodinâmica. Logo então a máquina elétrica passa a operar como motor, recolhendo o cabo (e o aerofólio). Quando o comprimento do cabo alcança o valor inicial, tem-se início um novo ciclo, com uma nova fase de geração. Como a força de tração do cabo é muito baixa durante a fase de recolhimento, e esta é realizada rapidamente, tem-se um saldo bastante positivo de energia gerada em um ciclo.

Figura 4 – Esquemático do gerador UFSCkite



Fonte: Adaptado de Schmidt et al. (2017).

No gerador sendo desenvolvido pelo grupo UFSCkite, como ilustrado na Figura 4, existe uma unidade de voo que conecta os cabos de tração do aerofólio (no bordo de ataque) ao cabo de tração único que chega à unidade de solo. Além disso, a unidade de voo é equipada com uma eletrônica embarcada que realiza, para citar alguns exemplos, o cálculo da ação de controle da trajetória de voo, a reconfiguração do aerofólio entre as condições de alta e baixa eficiência aerodinâmica, e a comunicação com a unidade de solo. Esta comunicação é imprescindível para o funcionamento seguro e eficiente do aerogerador, permitindo a atualização periódica de *setpoints* de controle, bem como a telemetria de variáveis importantes sobre a operação do sistema.

1.1 Objetivo e metodologia

Como objetivo geral do trabalho tem-se o desenvolvimento de um protocolo e infraestrutura de comunicação que utiliza módulos (*transceivers*) de radiofrequência para a transferência de dados entre as unidades de solo e de voo do aerogerador, possibilitando a troca de informações integras e a uma taxa de atualização suficiente para o controle e monitoramento do sistema.

Metodologia

A metodologia para se alcançar o objetivo estabelecido é composta das seguintes etapas, dispostas de forma cronológica.

- Instrumentação da comunicação e configuração de *hardware*.
- Definição dos pacotes de dados que devem ser trocados entre as unidades de solo e voo.
- Elaboração do protocolo de comunicação.
- Realização de testes para se comparar os dados obtidos via radiofrequência com aqueles via rádio.
- Aprimoramento do protocolo de comunicação, visando melhoria do desempenho.

Estrutura do documento

O restante do documento está organizado da seguinte maneira. O segundo capítulo é destinado a uma revisão dos conceitos básicos de comunicação utilizando radiofrequência, além de abordar também outros aspectos técnicos, como os diferentes tipos de produtos que podem ser encontrados no mercado e diferentes abordagens na utilização da radiofrequência para comunicação. O terceiro capítulo apresenta o equipamento utilizado para a realização dos experimentos, como as placas de desenvolvimento e módulos de comunicação via radiofrequência *swarm bee*, da fabricante Nanotron. O quarto capítulo traz o desenvolvimento do protocolo de comunicação em si e discorre sobre e sua integração com a estrutura de *software* já presente no grupo UFSCkite. O quinto capítulo apresenta os resultados obtidos com o protocolo de comunicação desenvolvido, incluindo uma análise sobre estes resultados para levantamento dos pontos negativos e positivos alcançados durante o desenvolvimento. Por fim, as conclusões e as perspectivas de trabalhos futuros são apresentadas no capítulo 6.

2 Fundamentação teórica

Sistemas de comunicação via radiofrequência existem há muitos anos e são utilizados em aplicações desde o acionamento do alarme de veículos até a operação de satélites. A comunicação sem fio vem sendo a área que mais cresce na indústria da comunicação e está alterando o modo como negócios são realizados no mundo. Por exemplo, redes locais sem fio (WLAN) permitem que funcionários se movimentem pelo ambiente de trabalho sem a restrição de fios, enquanto redes *ad hoc*, como as criadas via *bluetooth*, permitem a transferência e sincronização de dados entre dispositivos. De forma geral estas tecnologias oferecem redução de custos e maior recursos para diversas aplicações, desde o varejo até o chão de fábrica.

Este capítulo apresenta um resumo conciso dos dispositivos de radiofrequência (RF), incluindo uma discussão dos conceitos e definições gerais de sistemas sem fio baseados em RF e uma breve explicação sobre a propagação das ondas de rádio.

2.1 Histórico das redes de comunicação via RF

A comunicação utilizando radiofrequência começou na virada do século 20, quando Marconi, utilizando patentes de Tesla, estabeleceu o primeiro sistema de rádio bem-sucedido em 1895, transmitindo sinais de rádio a uma distância de 2 quilômetros. Ele conduziu experimentos adicionais até 1901, quando transmitiu sinais de rádio através do Oceano Atlântico, da Inglaterra a Terra Nova, na costa leste canadense, a cerca de 2.700 quilômetros de distância. Até este experimento, somente códigos telegráficos eram transmitidos. Contudo, em 1906, Reginald Fessenden realizou a primeira comunicação via rádio da fala humana a uma distância de 18 quilômetros (MAITRA, 2003).

Nos primeiros anos da comunicação via radiofrequência, a transmissão de rádio foi a tecnologia de comunicação sem fio mais implantada no mundo. A invenção do tubo de vácuo e triodo de vácuo acelerou o avanço na transmissão de sinais de voz por rádio. Primeiramente foi utilizada modulação de amplitude (AM) e, posteriormente, modulação de frequência (FM), introduzida no final da década de 1930 (MARK; ZHUANG, 2003). Muitos outros sistemas de radiofrequência foram introduzidos entre o final da década de 1920 e meados da década de 1950, como a televisão, rádio unidirecional, bidirecional e o radar. Outro marco foi testemunhado no final da década de 1970, que marcou o início do crescimento dos rádios móveis celulares e serviços de comunicação pessoal. O primeiro celular móvel analógico comercial de sucesso foi demonstrado em 1979. Atualmente, a comunicação sem fio de todos os tipos abundam em nossa sociedade (UGWEJE, 2004).

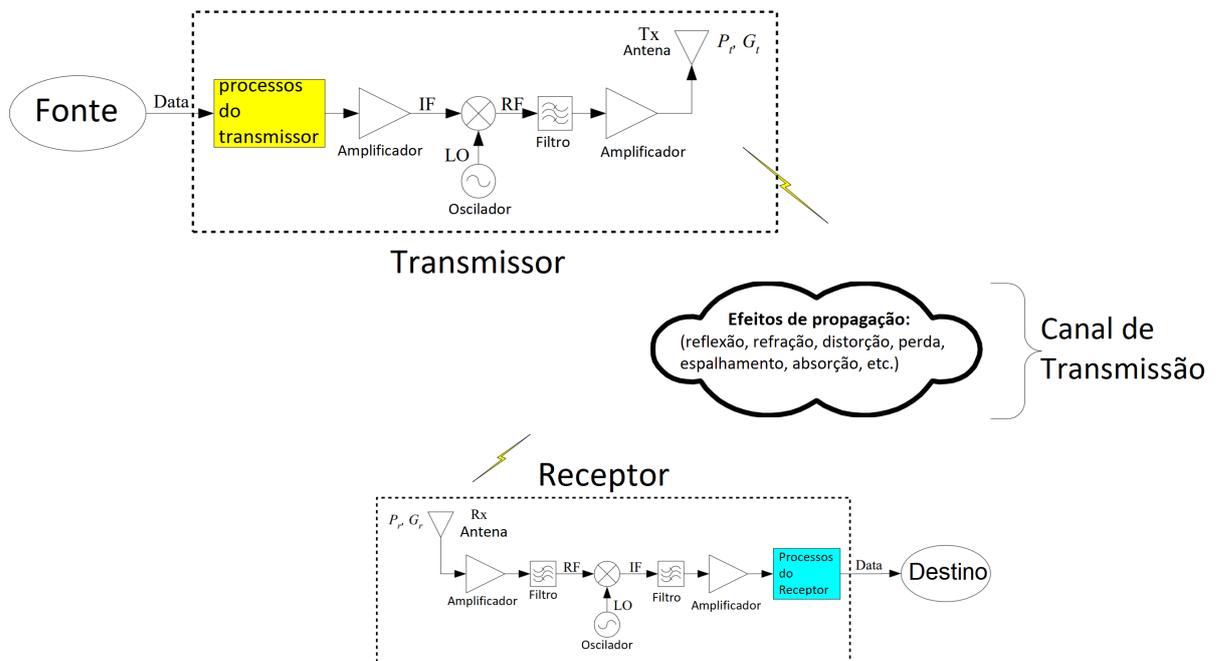
2.2 Arquitetura do sistema de radiofrequência

Nos sistemas de comunicação via radiofrequência, ondas de rádio são usadas para transferir informações entre um transmissor (Tx) e um receptor (Rx). Os sistemas de RF podem ser classificados como terrestres e espaciais. Alguns exemplos de sistemas terrestres incluem microondas ponto a ponto, WLANs e a rádio móvel celular. Os sistemas de microondas terrestres são limitados em relação à distância em que o sinal pode ser propagado. Para remediar isso, torres de relé com antenas direcionais são frequentemente usadas para fornecer um caminho desobstruído.

Os sistemas espaciais (por exemplo, satélites) são semelhantes aos sistemas terrestres de microondas, exceto pelo fato de que os sinais viajam das estações terrestres para um satélite (*uplink*), de onde é enviado de volta para outra estação terrestre terrestre (*downlink*). Isso resulta em uma área de cobertura muito maior que aquela dos sistemas terrestres.

Um sistema de comunicação sem fio típico é mostrado na Figura 5. Ele consiste em uma fonte de informações, um subsistema de *hardware* chamado transmissor, o canal ou meio pelo qual o sinal viaja, outro subsistema de *hardware* chamado receptor, e um destino para a informação.

Figura 5 – Modelo simplificado de comunicação via radiofrequência terrestre



Fonte: original

O transmissor e o receptor são sistemas complexos que consistem em muitos componentes internos, como demonstra simplificada a Figura 5. Os componentes são indicados como processos de transmissor, processos de receptor, amplificadores, osciladores, filtros e antenas. Os processos do transmissor representam funções do transmissor, como modulação, codificação, conversão de analógico para digital e multiplexação. Os processos do receptor, por outro lado, denotam funções inversas, como desmodulação, decodificação, conversão digital-analógica e demultiplexação (WEISMAN, 2003).

A fonte fornece as informações ao transmissor na forma de áudio, vídeo, dados ou uma combinação dos três. A configuração de transmissor e receptor é usada para converter o sinal em um formato adequado para transmissão e depois converter o sinal de volta à sua forma original. Isso é realizado através de um processo de modulação (ou codificação) no lado Tx e desmodulação (ou decodificação) no lado Rx. O sinal é propagado pelo canal de transmissão que, na comunicação sem fio via rádio, não necessita de meio físico para se propagar. Algumas consequências da propagação são a adição de ruído e interferência ao sinal, que geralmente acarretam atenuação, distorção e, eventualmente, erro no sinal recebido.

Os ruídos introduzidos no canal de propagação podem ser de origem humana, por exemplo devido à rede de distribuição de energia elétrica, de natureza atmosférica, como as atividades elétricas em nuvens ou ionosfera, ou ainda de natureza extraterrestre, como ondas eletromagnéticas e partículas de alta energia geradas por erupções solares ou de outras estrelas. Portanto o canal de propagação de RF é variável no tempo, de tal forma que a sua eficácia pode variar devido às causas supracitadas. Esta variação no tempo deve ser considerada durante o projeto de qualquer sistema de comunicação sem fio.

2.3 Classificação do espectro de rádio

As frequências de rádio constituem uma faixa do espectro eletromagnético que se estende desde 3 kHz até 300 GHz. Todo o espectro de RF é classificado em diferentes bandas e intervalos, baseado nas propriedades de propagação. Sinais de banda base (por exemplo, sinais de áudio) estão na faixa de baixa frequência, abaixo de 30 kHz. Esse intervalo de frequência é classificado como frequência muito baixa, que deve ser convertida em RF antes da transmissão. As ondas de rádio também são caracterizadas pelo seu comprimento de onda como ondas curtas, médias ou milimétricas. Quanto maior a frequência, menor o comprimento de onda, conforme a equação

$$\lambda = c/f_c \text{ ,} \tag{2.1}$$

na qual c é a velocidade da luz e f_c é a frequência portadora. O comprimento de onda (λ) também está relacionado ao comprimento da antena (L), largura de banda do sistema (B) e outros parâmetros práticos do sistema. Em geral, ondas de rádio de alta frequência produzem ondas de comprimento menores, requerem uma antena menor e têm menor eficiência de largura de banda (ρ). São também menos suscetíveis a atividade elétrica atmosférica mas sofrem com distorção térmica e climática atmosférica. A eficiência de largura de banda (ρ) é o número de bits de informação transmitidos por segundo por unidade de largura de banda (Hz), considerando que

$$\rho = \frac{R_b}{B} = \frac{\log_2 M}{BT_s} = \frac{1}{BT_b} \text{ bits/s/Hz} . \quad (2.2)$$

À medida em que ρ cresce, a largura de banda necessária diminui, se for considerada uma taxa de bits (R_b) constante. M representa o número de símbolos que varia com o tipo de modulação utilizada, enquanto T representa o período. Os subscritos s e b se referem a representação em segundos e em bits, respectivamente. Qualquer sistema digital perde eficiência de largura de banda à medida em que BT_b cresce.

As características e aplicações das frequências de rádio estão resumidas na Tabela 2. Dentro de cada faixa de frequência, várias bandas de frequência podem ser designadas para comunicação. Essas bandas são comumente identificadas tanto por sua frequência portadora (f_c) ou por um símbolo de letra – por exemplo, é possível descrever um sistema de RF como operando em bandas C, X, K e Ka no lugar de utilizar as frequências numericamente.

Tabela 2 – Características e aplicações das frequências de rádio

Nome da Banda	Sigla	Frequência	Comprimento da Onda	Exemplos de Uso
<i>Extremely low frequency</i>	ELF	3–30 Hz	100,000–10,000 km	Comunicação com submarinos
<i>Super low frequency</i>	SLF	30–300 Hz	10,000–1,000 km	Comunicação com submarinos
<i>Ultra low frequency</i>	ULF	300–3,000 Hz	1,000–100 km	Comunicação com submarinos/minas
<i>Very low frequency</i>	VLF	3–30 kHz	100–10 km	Navegação, monitor de batimento cardíaco sem fio
<i>Low frequency</i>	LF	30–300 kHz	10–1 km	Rádio amador, onda longa AM
<i>Medium frequency</i>	MF	300–3,000 kHz	1,000–100 m	Onda média AM, rádio amador
<i>High frequency</i>	HF	3–30 MHz	100–10 m	Radar OTH, comunicações de aviação, RFID, telefonia marítima e móvel
<i>Very high frequency</i>	VHF	30–300 MHz	10–1 m	Rádio FM, transmissão de canais de televisão
<i>Ultra high frequency</i>	UHF	300–3,000 MHz	1–0.1 m	Transmissão de canais de televisão, forno microondas, rádio via satélite
<i>Super high frequency</i>	SHF	3–30 GHz	100–10 mm	Radares mais modernos, satélites, transmissão de canais a cabo
<i>Extremely high frequency</i>	EHF	30–300 GHz	10–1 mm	WLAN (802.11ad), radioastronomia
<i>Tremendously high frequency</i>	THF	300–3,000 GHz	1–0.1 mm	Experimentos para substituir raios X, sensoriamento remoto

Devido ao congestionamento ou indisponibilidade do espectro utilizável nas faixas de frequência mais baixas (abaixo de 20 GHz) e à recente demanda por comunicação multimídia com alta taxa de dados, projetistas de sistemas têm direcionado sua atenção para o uso de *super high frequency* (SHF) e *extremely high frequency* (EHF) para a comunicação. A faixa de banda EHF traz diversos benefícios potenciais, como a disponibilidade de espectro utilizável, alta capacidade de taxa de dados e redução na interferência. No entanto, nessas frequências, a distorção atmosférica, especialmente a atenuação devido à chuva, é um problema grave (RAPPAPORT, 2002). A severidade dos efeitos meteorológicos aumenta com o aumento da frequência e, em algumas bandas de frequência, os efeitos meteorológicos podem causar uma redução na amplitude do sinal recebido, despolarização

da onda de rádio e um aumento do ruído térmico.

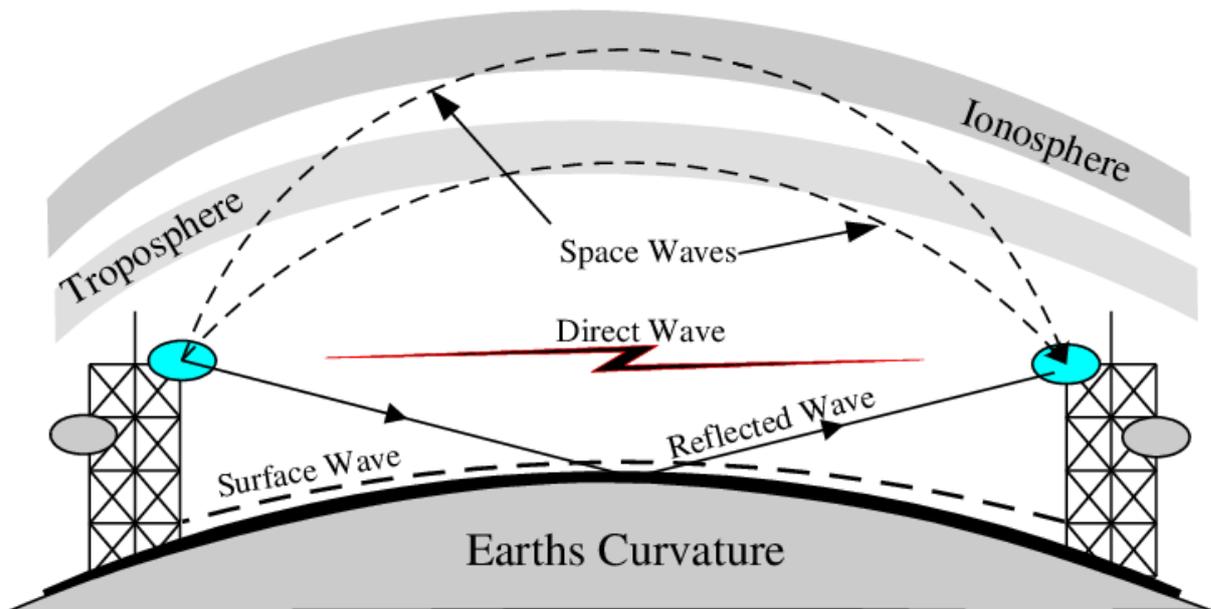
2.4 Características das ondas de rádio

Quando energia elétrica é aplicada a uma antena, esta energia é convertida em ondas eletromagnéticas. No lado do transmissor, a antena converte uma tensão ou corrente variável no tempo em uma onda eletromagnética de propagação variável no tempo. Esta onda de rádio se propaga pelo espaço como um campo eletromagnético itinerante, proporcional à tensão ou corrente que varia no tempo. A energia de propagação da radiofrequência é composta por um campo elétrico e um campo magnético. Os dois campos coexistem porque uma mudança no campo elétrico gera uma mudança correspondente no campo magnético e vice-versa. Já no lado receptor, a antena realiza uma operação inversa de conversão de uma onda eletromagnética de propagação variável no tempo para uma tensão ou corrente que varia no tempo.

A polarização da onda de rádio é algo importante e é determinada pela direção do campo elétrico. Geralmente a construção e orientação da antena determinam esta direção do campo elétrico. A magnitude da potência irradiada na direção da propagação pode ser calculada e é denominada como potência irradiada efetiva, que corresponde à potência máxima irradiada disponível para o transmissor na direção de ganho máximo de antenas direcionais ou isotrópicas. Este valor é uma medida da eficácia de uma antena no direcionamento da potência do transmissor em uma direção específica.

As ondas de rádio se propagam no espaço de diversas maneiras e podem ser classificadas como ondas diretas, de superfície, troposféricas e ionosféricas, como ilustrado na Figura 6. As ondas mais simples são as ondas diretas, que têm o meio de propagação livre de obstruções e são projetadas para viajar em um linha reta entre o transmissor e receptor. Alguns exemplos de aparelhos que utilizam ondas diretas são os rádios *walk talk* e torres de celular. Já as ondas de superfície se propagam, como o nome sugere, ao longo da superfície terrestre e por isso são afetadas pelo terreno ou qualquer obstáculo no caminho. Ao longo da propagação elas são refletidas e se dispersam à medida em que encontram prédios, montanhas ou qualquer outra irregularidade no caminho. Por fim, as ondas troposféricas e ionosféricas se propagam em direção ao espaço sideral mas podem retornar à Terra devido à refração e/ou a dispersão na ionosfera ou troposfera (KAIGARULA, 2013). A troposfera é a camada atmosférica mais próxima da superfície terrestre e se estende até aproximadamente doze quilômetros da superfície. A onda ionosférica é similar à troposférica, porém ela tem de viajar uma maior distância para a refração ocorrer na ionosfera, que localiza-se entre 60 km e 1.000 km da superfície terrestre. Esta onda ionosférica pode ser muito confiável para telemetria, aplicações militares e previsão de tempo.

Figura 6 – Tipos de propagação de ondas de rádio



Fonte: (UGWEJE, 2004)

2.5 Efeitos de propagação das ondas de rádio

O sinal emitido pelo transmissor é sempre diferente do recebido pelo receptor devido aos diversos inconvenientes relacionados à propagação das ondas de radiofrequência, já mencionados. Quando as ondas de rádio são irradiadas a partir de uma antena, a sua propagação é governada pelos mecanismos descritos a seguir.

Propagação pelo espaço livre

Considerada como a propagação ideal para os casos em que existe uma linha de visão direta entre receptor e transmissor separados por uma distância d . Se considerarmos a potência transmitida P_t , a potência recebida P_r pode ser determinada pela equação (Rappaport, 2002)

$$P_r(d) = G_t G_r P_t \frac{\lambda^2}{(4\pi d)^2 L} = P_t \frac{A_{et} A_{er}}{(\pi d)^2 L}, \quad (2.3)$$

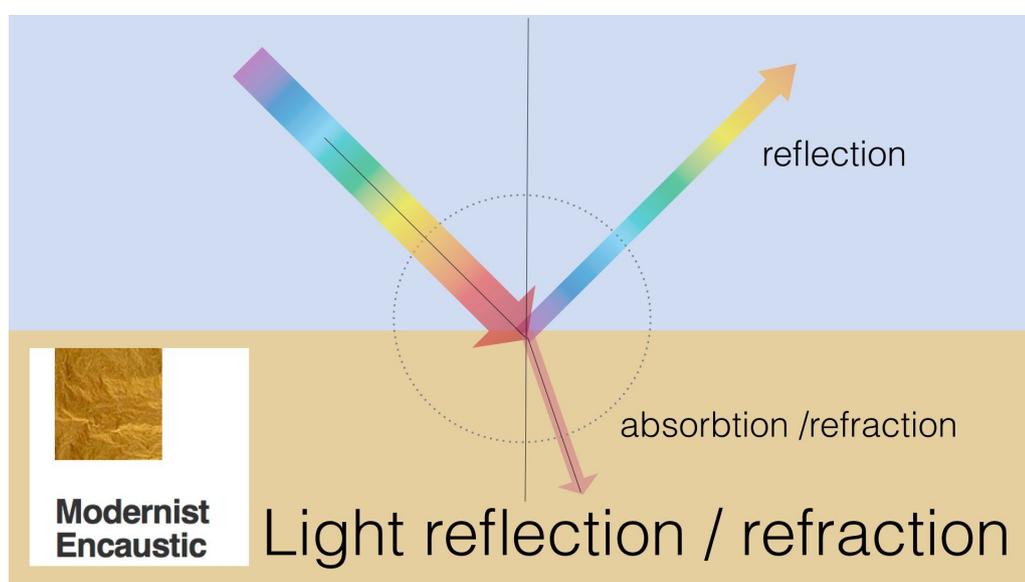
na qual A_e , G e L são respectivamente a área efetiva, ganho da antena e o fator de perda do sistema. Os subscritos t e r se referenciam ao transmissor e receptor respectivamente. Podemos perceber na Equação 2.3 que a potência recebida decai a uma taxa de 20 dB/década com o aumento da distância.

Reflexão e Refração

Quando uma onda de rádio se choca com um obstáculo de dimensões muito grandes se comparadas com o comprimento de onda, a reflexão ocorre. Todas as ondas de rádio sofrem reflexão se o meio de propagação tem suas propriedades físicas alteradas abruptamente. Além disso, dependendo das características do obstáculo, a onda pode ser parcialmente refletida, completamente refletida ou absorvida.

Já a refração ocorre no limiar de dois dielétricos, quando a onda incidente se propaga para outro meio em um determinado ângulo. Quando a onda vai de um meio de propagação de certa densidade para outro com uma densidade diferente, a velocidade da onda muda. Essa mudança de velocidade faz com que a onda se curve no limiar entre os dois meios, e a onda sempre se curvará em direção ao meio mais denso. Podemos visualizar o que acontece com a onda de rádio quando ocorre a reflexão ou a refração na Figura 7.

Figura 7 – Reflexão e refração



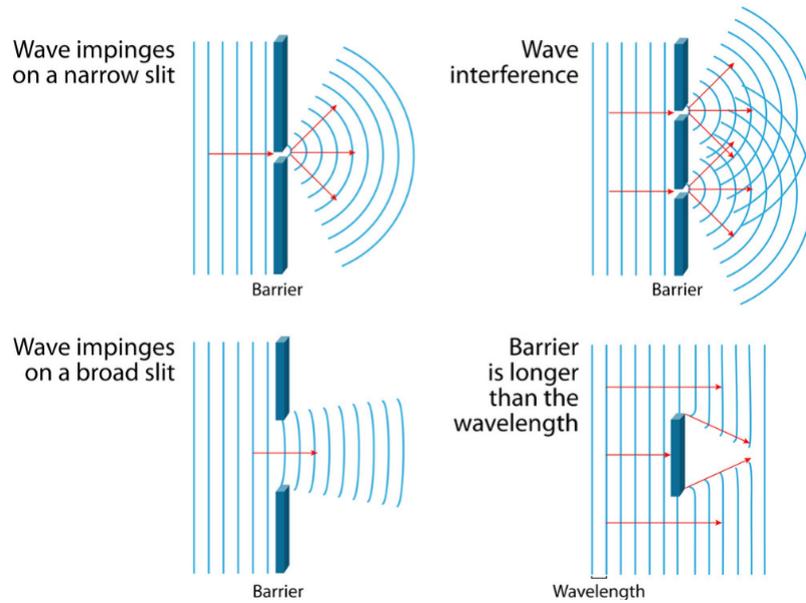
Fonte: Modernist Encaustic (2020)

Difração

A difração é o efeito de propagação que ocorre quando a onda de rádio encontra um obstáculo e tende a se propagar ao redor dos cantos do obstáculo. Para a difração ocorrer o obstáculo deve ter dimensões comparáveis ao tamanho do comprimento de onda. Este efeito de propagação pode ser visualizado na Figura 8.

Figura 8 – Difração

DIFFRACTION OF WAVES

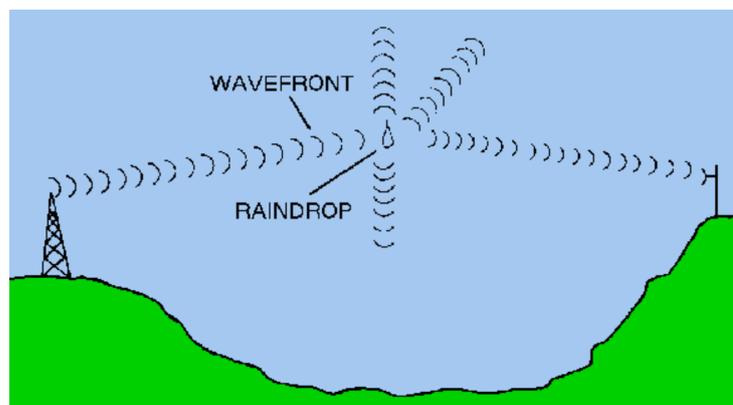


Fonte: VectorStock (2020)

Dispersão

A dispersão acontece quando a onda de rádio encontra um obstáculo com dimensões menores do que o comprimento da onda. Conseqüentemente, o sinal se dispersa para todas as direções. A Figura 9 ilustra o efeito de dispersão quando uma onda de rádio se choca com uma gota de chuva.

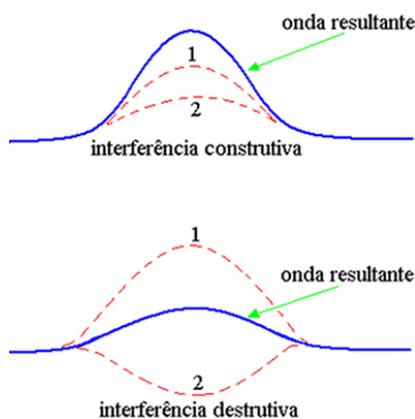
Figura 9 – Dispersão

Fonte: *Navy Electricity and Electronics Training Series* (NEETS, 1998)

Interferência

A interferência pode ocorrer quando a onda transmitida chega a um ponto via dois ou mais caminhos e portanto tem diferença de fase. Com isso, as ondas podem se adicionar construtivamente ou destrutivamente no receptor, como podemos ver na Figura 10.

Figura 10 – Interferência



Fonte: Brasil Escola (2020)

Absorção

O efeito de absorção acontece quando a energia da onda de rádio penetra um objeto e é convertida em calor. A onda incidente penetra o objeto e continua se propagando dentro do objeto, à medida em que a energia da onda cai exponencialmente com a distância percorrida dentro do objeto. Assim, pode ocorrer uma dissipação completa da onda ou ela pode re-emergir do material, com amplitude menor. As partículas da atmosfera absorvem a energia da radiofrequência, o que normalmente só acontece com ondas de frequência maior que 10 GHz.

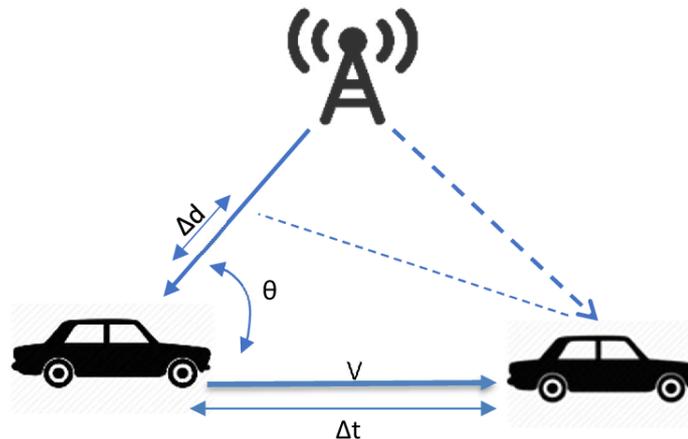
Efeito Doppler

Se trata da mudança na frequência devido à diferença na velocidade entre o transmissor e receptor. A frequência f_d devido ao efeito Doppler pode ser calculada pela equação

$$f_d = f_m \cos \theta = \frac{v \cos \theta}{\lambda} = \frac{v_{eff}}{\lambda} = \frac{v_{eff} f_c}{c}, \quad (2.4)$$

Para um objeto se movendo em velocidade constante v , a frequência da onda portadora f_c é alterada em uma quantidade máxima $f_m = v/\lambda$, v_{eff} a velocidade do objeto e θ o ângulo entre transmissor e receptor. Este ângulo é demonstrado na Figura 11.

Figura 11 – Efeito Doppler



Fonte: (HAMID et al., 2019)

O resultado do efeito Doppler é uma onda que chega no receptor com modulação de fase e frequência aleatória. O processamento da informação transmitida por essa onda exigirá o uso de técnicas de detecção de fase e frequência diferenciais.

2.6 Técnicas de comunicação via radiofrequência

Devido a estes diversos efeitos negativos na propagação das ondas de rádio, pode-se perceber que o canal sem fio não é um meio de propagação muito confiável. Para contornar estes problemas, algumas técnicas foram desenvolvidas, entre as quais o alargamento de espectro, acesso múltiplo, diversidade, equalização e codificação. Nesta seção serão introduzidas as técnicas mais utilizadas para garantir a confiança do meio de propagação.

Alargamento do espectro

A ideia por trás dessa técnica é a de transformar um sinal com largura de banda B_s em um sinal semelhante a um ruído com um largura de banda muito maior B_{ss} . Isto é alcançado pela modulação dos dados com uma sequência de ruído pseudo-aleatório a uma taxa muito maior que a taxa de dados. A capacidade C do canal pode ser determinada pela equação

$$C = B \log_2(1 + SNR) , \quad (2.5)$$

na qual B é a largura de banda (Hz) e SNR é a razão entre sinal e ruído. Nota-se que quando a largura de banda é aumentada, é possível reduzir a razão entre sinal e ruído

sem perder capacidade. O principal parâmetro de sistemas de alargamento de espectro é o ganho de processamento G_p , definido como

$$G_p = \frac{B_{ss}}{B_s} = \frac{T_b}{T_c}, \quad (2.6)$$

sendo T_b e T_c o período de bits e o período da sequência de ruído pseudo-aleatório. O ganho G_p pode ser considerado como o aumento em performance decorrido do alargamento de espectro. Ele também determina o número de usuários que podem ser permitidos por um sistema e é usado para descrever a fidelidade ganha no sinal ao custo de largura de banda (RAPPAPORT, 2002). Para sistemas de alargamento de espectro é vantajoso que o ganho G_p seja o mais alto possível, pois dessa forma maior será a capacidade do sistema de suprimir interferência. Técnicas de alargamento de espectro são utilizadas nas redes de telefonia celular e em satélites de posicionamento global (GPS).

Diversidade

Diversidade é uma técnica utilizada para aumentar a performance do sistema de radiofrequência que se baseia em disponibilizar uma ou mais cópias similares de um sinal que se deseja transmitir. Assim, são emitidas duas ou mais cópias similares de um mesmo sinal, que sofrem enfraquecimento de forma independente. Existem diversas técnicas de combinação de diversidade, as quais podem ser baseadas em frequência, espaço (antena), ângulo de chegada, polarização e tempo de chegada. Por exemplo, na diversidade baseada em espaço o sinal transmitido é recebido por N antenas diferentes. A recepção por diversidade aumenta a confiabilidade do sistema sem aumentar a potência transmitida e a largura de banda do canal (YACOUB, 1993).

Outras técnicas comuns de combinação de diversidade são a combinação de razão máxima, combinação por ganho igual e a seleção de diversidade. Na combinação de razão máxima, os sinais recebidos são pesados e adicionados de forma a enfatizar os sinais mais confiáveis e suprimir os sinais menos confiáveis. Já na combinação por ganho igual, os sinais recebidos são pesados e combinados sem levar em conta as intensidades dos sinais individualmente. Por fim, na seleção de diversidade, o sinal mais desejável é selecionado e os outros são descartados.

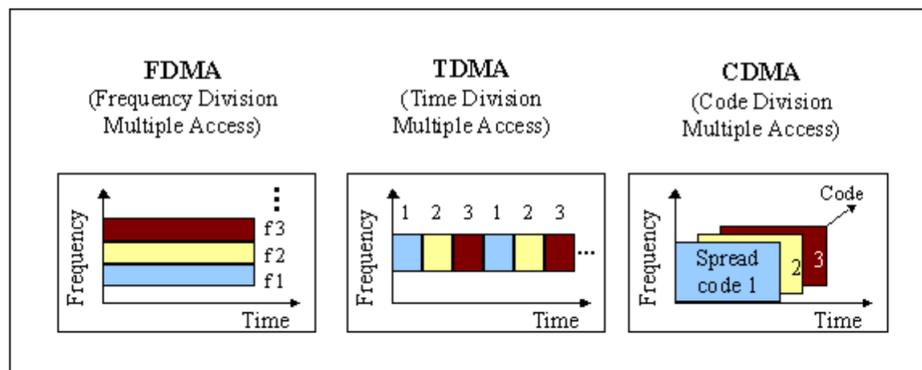
Acesso múltiplo

O espectro de radiofrequência é finito e portanto é necessário compartilhar os recursos entre os usuários. Técnicas de acesso múltiplo são as estratégias principais para o compartilhamento de recursos e se baseiam em protocolos de multiplexação, os quais permitem que mais de um transceptor (dispositivo que combina um transmissor e um

receptor para obter ambas funções em um só aparelho) compartilhem o mesmo meio. Isto pode ser alcançado utilizando o tempo, frequência e código, dando origem a três técnicas populares conhecidas como múltiplo acesso por: divisão de frequência (FDMA); por divisão de tempo (TDMA); e por divisão de código (CDMA).

No múltiplo acesso por frequência, todo o espectro é dividido em bandas menores, que são atribuídas para os usuários de acordo com a demanda. Já no múltiplo acesso por tempo, o tempo disponível é dividido em *frames* de duração igual. Somente um usuário pode transmitir ou receber a cada *frame* de tempo, formando assim um estrutura de tempo cíclica para ordenar os acessos dos usuários. Por fim, o múltiplo acesso por código atribui para cada usuário uma sequência de código única “PN”, assim possibilitando que diversos usuários utilizem a mesma largura de banda simultaneamente. Essas três técnicas de múltiplo acesso e suas diferenças podem ser visualizadas na Figura 12.

Figura 12 – Múltiplo acesso por FDMA, TDMA e CDMA



Fonte: International Telecommunication Union (2020)

3 Infraestrutura de hardware e software

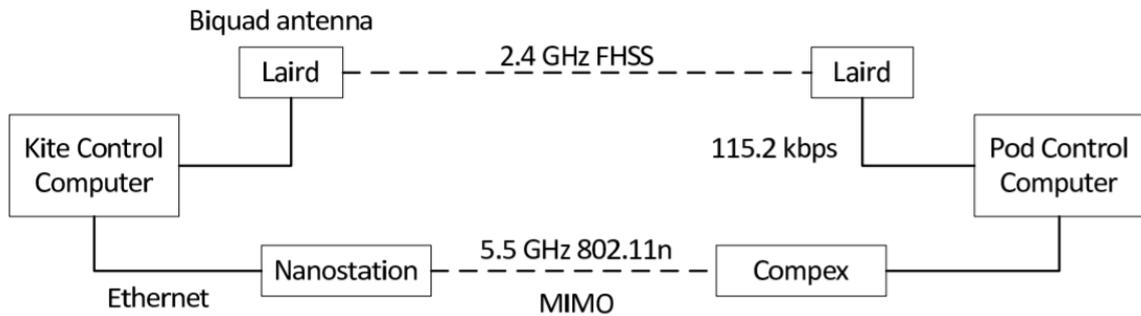
Como discutido na Seção 1.1, o objetivo geral deste trabalho é o estabelecimento de uma infraestrutura de comunicação entre as unidades de solo e de voo do aerogerador, incluindo o desenvolvimento de um protocolo de comunicação a nível de aplicação. O primeiro passo para alcançar este objetivo consiste na instrumentação do sistema de comunicação via radiofrequência. Deste modo, neste capítulo serão discutido os componentes de *hardware* e *software* que foram utilizados durante a realização do trabalho, assim como uma breve introdução a outros trabalhos desenvolvidos pelo mundo para sanar a comunicação entre unidade de solo e voo em sistemas AWE.

3.1 Trabalhos correlatos

Existem diversos métodos de controle do aerofólio, dois deles ilustrados na Figura 22, e algumas táticas para realizar a comunicação entre a unidade de controle em voo e a unidade de solo. Esta comunicação pode ser realizada através de uma rede cabeada ou sem fio. Entretanto, as redes cabeadas adicionam peso à estrutura e são suscetíveis à danificação caso ocorra uma falha mecânica. Portanto, pela natureza do aerogerador, é mais adequado que a comunicação entre unidades de solo e voo sejam realizadas sem fios.

Uma das implementações de comunicação via radiofrequência entre unidades de voo e solo foi proposta por MAURER (2012), onde se foi utilizado de um *link* “duplo” de comunicação, onde um deles é considerado o *link* principal e o outro o secundário. Esta configuração proposta por MAURER, de acordo com o autor, resulta em uma redução na perda de pacotes envolvidas na comunicação via radiofrequência e possibilita a comunicação através do *link* secundário caso o principal esteja indisponível. Esta configuração pode ser visualizada na Figura 13.

Figura 13 – Sistema de comunicação via RF proposta por MAURER



Fonte: (MAURER, 2012).

Outra configuração, proposta por FAGIANO et al., utiliza do protocolo ZigBee (IEEE 802.15.4). Este protocolo é ideal para redes sem fio de sensoriamento que requerem alta confiança, baixo custo de energia e uma taxa de transferência de dados mais baixa. A configuração proposta por FAGIANO et al. utiliza um *link* de comunicação entre dois módulos de radiofrequência, um no solo e outro na unidade de voo, logo é mais semelhante ao que foi desenvolvido durante esta monografia. Entretanto, o autor não entra em detalhes de *hardware* e de como o protocolo de comunicação foi estabelecido.

3.2 Placas de desenvolvimento

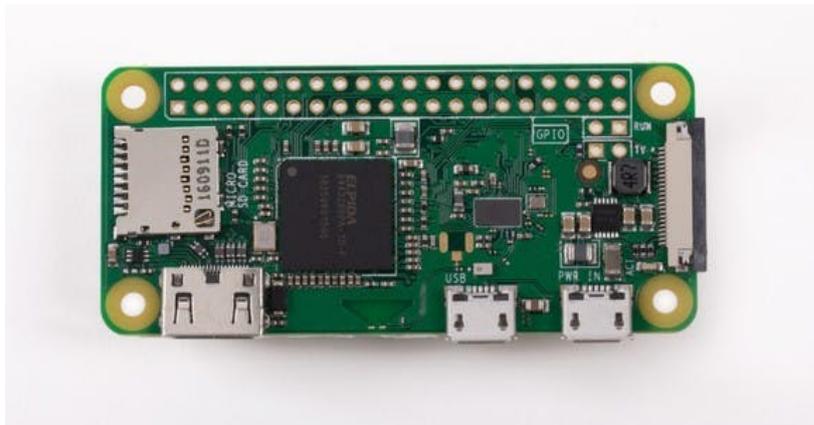
Como ilustrado na Figura 4, o protótipo de aerogerador AWE é composto basicamente por uma unidade de solo e por uma unidade de voo que acompanha o aerofólio cabeado. Em cada uma das unidades existe uma eletrônica embarcada que, entre seus componentes, destacam-se microcomputadores embarcados. Na unidade de solo está instalada uma BeagleCore, placa baseada na plataforma Beaglebone, que é mais adequada para utilização em eletrônicas embarcadas sujeitas a campo magnéticos de intensidade elevada (neste caso, devido principalmente à operação do inversor de frequência da máquina elétrica). Esta placa de desenvolvimento pode ser vista na Figura 14. Já na unidade de voo está instalada uma Raspberry Pi Zero W, que pode ser vista na Figura 15.

Figura 14 – BeagleCore



Fonte: BeagleCore™

Figura 15 – Raspberry Pi Zero W



Fonte: Raspberry Pi Foundation

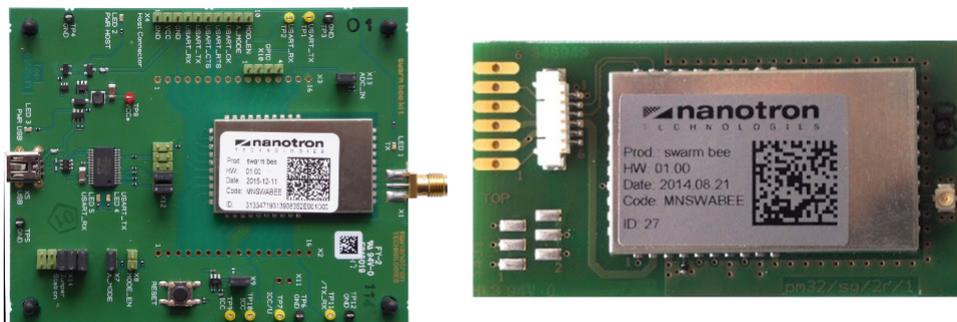
3.3 Módulos *swarm bee* de radiofrequência

Para realizar a comunicação via radiofrequência foram utilizado os módulos *swarm bee*, produzidos pela empresa Nanotron. Estes módulos já estavam sendo utilizados pelo grupo UFSCkite para a medição da distância entre um módulo instalado no aerofólio e quatro módulos fixos no chão, em um algoritmo de lateração para a estimativa da posição e velocidade do aerofólio cabeado. Estes módulos utilizam uma modulação de alargamento de espectro *chirp*, que utiliza pulsos de *chirp* modulados em frequência linear de banda larga para codificar informações. O *chirp* é basicamente um sinal senoidal que

aumenta e diminui de frequência com o tempo, normalmente existindo uma expressão polinomial entre tempo e frequência.

Os módulos *swarm bee* utilizam medições do tempo de voo para estimar a distância entre dois dispositivos. Apesar de esta técnica estar sujeita a efeitos de múltiplos caminhos, ela consegue alcançar uma precisão significativamente maior em comparação com outras técnicas, como a avaliação da potência do sinal recebido. Existem algumas versões disponíveis dos módulos *swarm bee*. Na unidade de solo é utilizada uma versão de desenvolvimento, que é maior, pois não existe praticamente restrição de espaço para a sua instalação. Já na unidade de voo é utilizada uma versão menor e mais leve, devido a restrições de espaço e peso da eletrônica embarcada. Estas duas versões podem ser vistas na Figura 16.

Figura 16 – *Swarm bee* DK plus e LE, respectivamente



Fonte: Nanotron

O fabricante dos módulos *swarm bee* disponibiliza para o usuário uma interface de programação de aplicações (API). Trata-se de uma lista de comandos que o usuário pode enviar para o módulo em código ASCII de modo a abstrair o nível mais baixo da implementação da radiofrequência. Um desses comandos, muito utilizado para enviar dados, pode ser visualizado na Figura 17. Neste caso, SDAT (*send data*) é o nome do comando. O próximo byte “1” é um *byte* de configuração, os *bytes* “1F318052001A” correspondem ao ID do módulo destinatário, os *bytes* “02” representam o tamanho da mensagem (em *bytes*), e “FA13” é o *payload* em si a ser enviado.

Figura 17 – Comando SDAT como exemplo de uso da API dos *swarm bees*

```
SDAT <Option> <ID> <Len> <Data>:
```

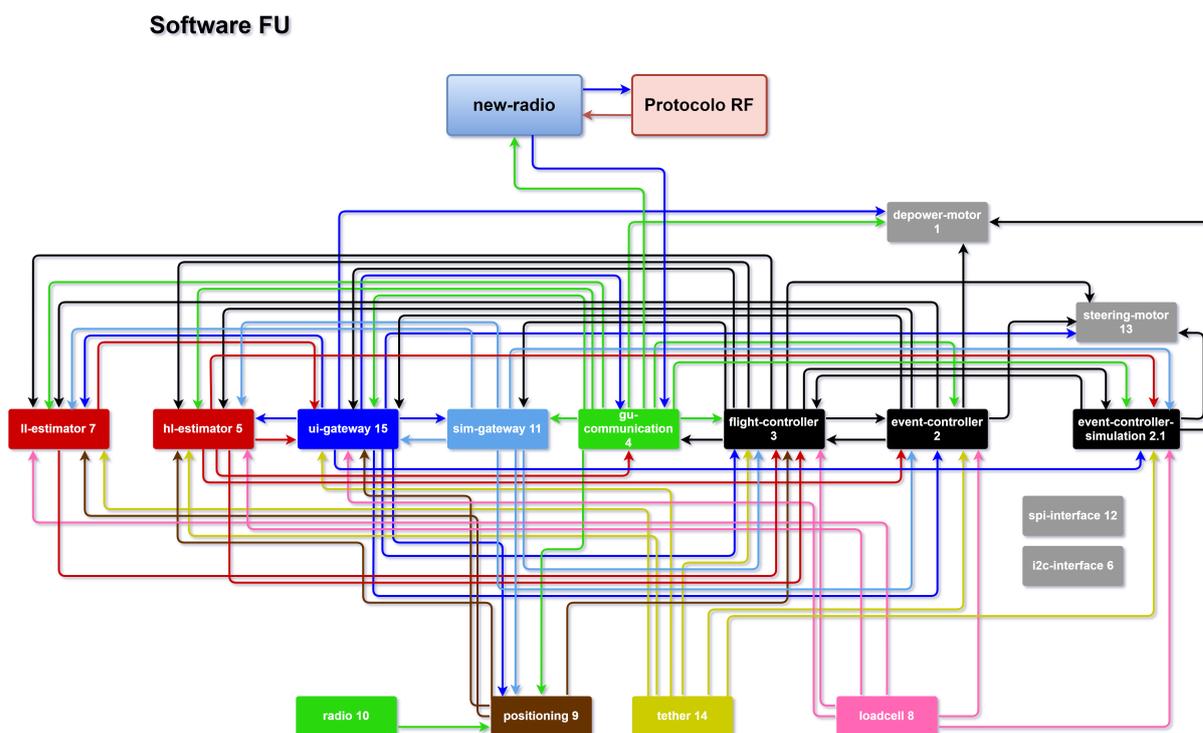
```
Ex : SDAT 1 1F318052001A 02 FA13
```

Fonte: Nanotron

3.4 Estrutura de software

Além dos elementos de *hardware*, uma estrutura de *software* é continuamente desenvolvida no projeto UFSCkite. Pode contribuir com essa estrutura durante esta monografia incorporando um novo módulo escrito em C e o protocolo de comunicação em Python nesta estrutura de *software*. Esta estrutura inclui toda a lógica de sensoriamento, filtragem e estimação, controle, atuação e comunicação de dados. Esta estrutura foi desenvolvida de forma modularizada para facilitar o entendimento e a interação dos desenvolvedores com o código. Esta modularização pode ser visualizada na Figura 18, onde as setas representam a direção do fluxo de informações entre os módulos.

Figura 18 – Estrutura modular de software da unidade de voo



Fonte: original

Esta troca de informações, representada por linhas com setas na Figura 18, é realizada através do protocolo de comunicação *Publish-Subscribe*, pelo qual os remetentes de mensagens, chamados de *publishers*, não informam especificamente os destinatários, chamados de *subscribers*, mas sim categorizam as mensagens publicadas em classes sem o conhecimento de quantos *subscribers* (se existirem) querem receber determinada mensagem. Da mesma forma, os *subscribers* manifestam interesse em uma ou mais classes de mensagens e recebem apenas mensagens de seu interesse, sem o conhecimento de quantos *publishers* existem, se houver. Na estrutura de *software* de ambas as unidades de solo

e de voo é utilizada a biblioteca LCM, que fornece um modelo de envio de mensagens *Publish-Subscribe*.

3.5 Biblioteca LCM

A biblioteca LCM é um conjunto de ferramentas para o envio de mensagens e organização de dados, direcionada a sistemas de tempo real nos quais a alta largura de banda e baixa latência são importantes. Esta biblioteca fornece um modelo de envio de mensagens de publicação e assinatura e a geração automática de código de “empacotamento e desempacotamento” para a utilização em aplicativos em uma variedade de linguagens de programação.

O primeiro passo para a utilização da biblioteca consiste na definição de um pacote de dados. A biblioteca LCM permite definir esses pacotes como estruturas de dados independentes da linguagem de programação utilizada. Estas estruturas podem ter vários campos, cada um com seu próprio tipo e nome. Alguns desses campos podem ser estruturas próprias ou matrizes. Por exemplo, um pacote pode ser definido como apresentado a Figura 19.

Figura 19 – Exemplo pacote de dados LCM

```
package exlcm;
struct example_t
{
    int64_t    timestamp;
    double    position[3];
    double    orientation[4];
    int32_t    num_ranges;
    int16_t    ranges[num_ranges];
    string    name;
    boolean    enabled;
}
```

Fonte: Repositório github LCM

O arquivo é bem simples e é separado em duas partes: um nome para o pacote, neste caso “exlcm”, e uma definição de estrutura. O pacote define um nome para a estrutura de dados e é mapeado para a construção dependendo da linguagem utilizada. A estrutura é uma lista de campos de dados, cada um com seu nome e tipo. Vários tipos primitivos estão disponíveis para uso, alguns exemplificados na Figura 19.

A ferramenta “lcm-gen”, distribuída juntamente com a biblioteca LCM, converte

as definições de tipo de mensagem em código para as linguagens de programação suportadas e mapeia os tipos de mensagem em estruturas ou classes de dados específicas da linguagem. Cada campo de dados, por sua vez, é mapeado em uma estrutura de dados nativa. Em C, por exemplo, *boolean* corresponde ao tipo “int8_t” e “string” corresponde a um caractere terminado por *NULL*. Além disso, é possível definir matrizes de tamanho fixo ou de comprimento variável. No exemplo da Figura 19, “position” é uma matriz “double” de comprimento três e “range” é uma matriz “int16_t” de comprimento variável. Este comprimento é especificado pelo campo “num_ranges”. Para gerar as ligações específicas do idioma basta executar a ferramenta “lcm-gen” com os argumentos listados na Tabela 3, de acordo com a linguagem de programação de sua escolha.

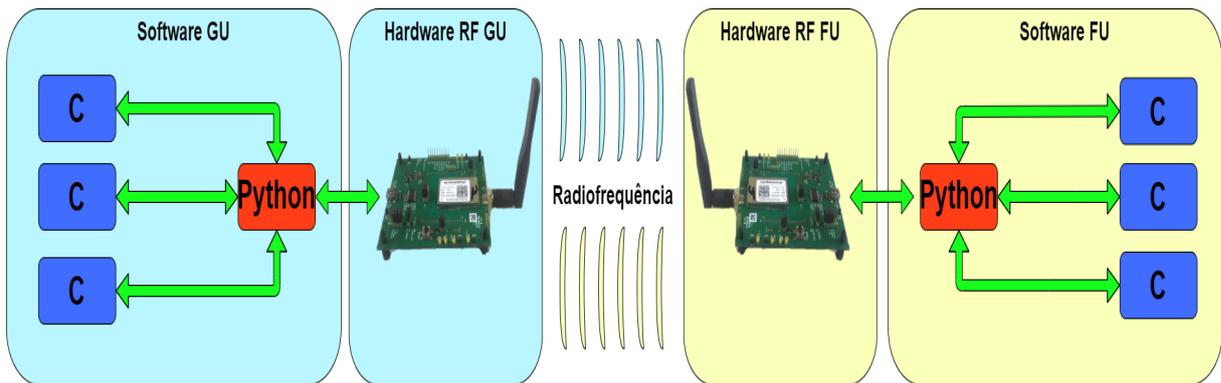
Tabela 3 – Geração de ligações específicas do idioma com a biblioteca LCM

Linguagem	Uso da ferramenta “lcm-gen”
C	lcm-gen -c example_t.lcm
C++	lcm-gen -x example_t.lcm
Java	lcm-gen -j example_t.lcm
Lua	lcm-gen -l example_t.lcm
Python	lcm-gen -p example_t.lcm
C#	lcm-gen -csharp example_t.lcm
MATLAB	Generate Java code
Go	lcm-gen -g example_t.lcm

3.6 Linguagens de programação utilizadas

A estrutura de *software* do aerogerador é formada em sua maioria por códigos escritos na linguagem C. Entretanto, neste trabalho foram desenvolvidos códigos na linguagem Python, pois no projeto UFSCkite já havia sido desenvolvido código para se trabalhar com os módulos *swarm bee* nesta linguagem. Apesar da diferença na linguagem, a biblioteca LCM permite a comunicação *Publish-Subscribe* entre as duas linguagens de forma transparente. Sendo assim, o código desenvolvido durante este trabalho se comporta como um módulo escrito em Python na estrutura de módulos formada primariamente por códigos desenvolvidos em C. Isto pode ser visualizado na Figura 20, onde estão ilustradas as estruturas de *hardware/software* da unidade de solo (GU) e da unidade de voo (FU). As setas verdes representam a comunicação *Publish-Subscribe* provida pela biblioteca LCM; as formas em azul escuro representam os módulos escritos em linguagem C; finalmente, os módulos em vermelho são aqueles desenvolvidos em Python durante a realização deste trabalho.

Figura 20 – Diagrama de representação da comunicação geral do sistema

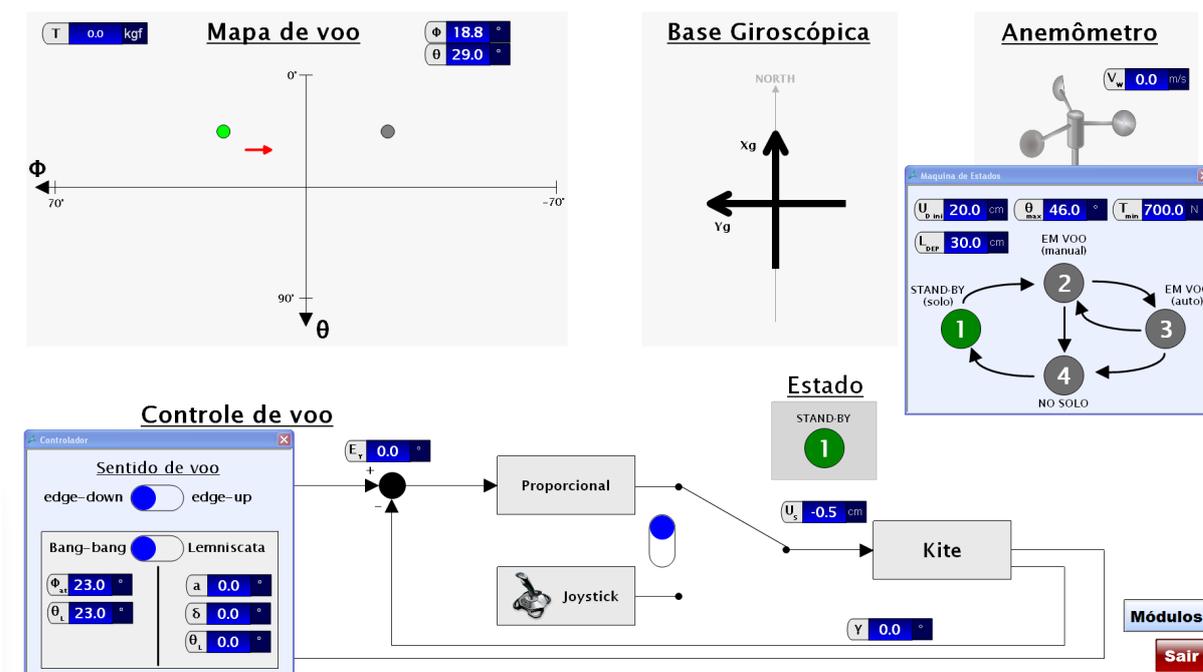


Fonte: original

3.7 Supervisório SCADA

Para realizar a supervisão, alteração de alguns parâmetros de estratégia de controle, e acionamento das máquinas na unidade de solo durante o funcionamento do aerogerador é utilizado um software supervisório. O software utilizado é o WinCC e ele permite a programação de diversas telas que servem para a supervisão e interação com o operador do aerogerador. Este supervisório se comunica com a unidade de solo através de um servidor OPC e é por este canal onde as variáveis são trocadas entre a estrutura de *software* e supervisório. Na Figura 21 é demonstrada uma das telas desenvolvidas no supervisório do aerogerador desenvolvido pelo grupo UFSCkite, na qual é possível monitorar e alterar parâmetros de controle de voo do aerofólio.

Figura 21 – Sistema supervisorio UFSCkite



Fonte: Repositório UFSCkite

4 Desenvolvimento do protocolo de comunicação

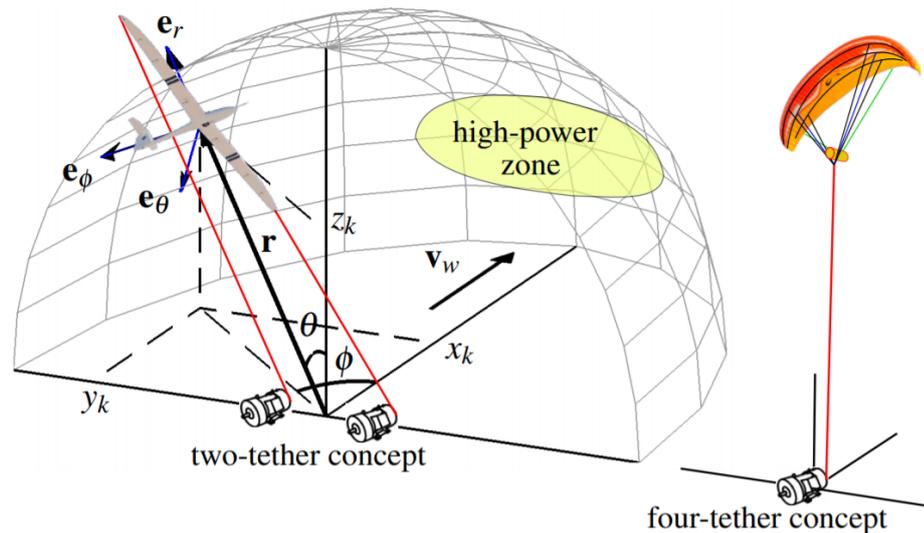
Para alcançar o objetivo geral deste trabalho, uma das principais atividades desenvolvidas foi a concepção de um protocolo de comunicação, implementado em código Python, para ser utilizado por um módulo da estrutura de *software* do aerogerador, tanto no microcomputador da unidade de voo como no da unidade de solo, como ilustrado na Figura 20.

Assim, neste capítulo serão discutidas as etapas no desenvolvimento do protocolo de comunicação, desde a estipulação do pacote de dados que deve ser trocado entre unidades até a fase inicial de colhimento de resultados.

4.1 Hardware-in-the-loop

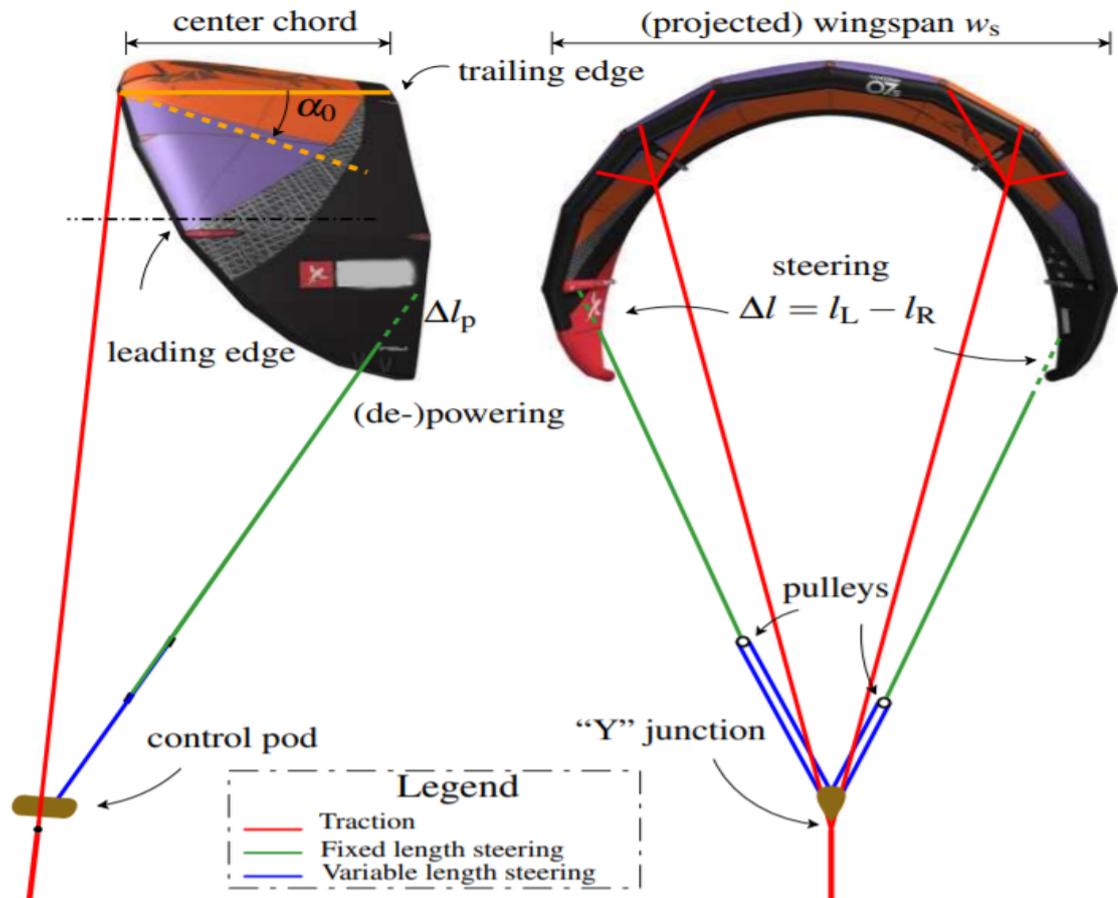
Para melhor entendimento do protocolo desenvolvido e testado no escopo desta monografia, é importante se ter conhecimento da situação atual dos testes realizados no projeto UFSCkite. Atualmente estão sendo realizados testes com a unidade de voo na configuração *hardware-in-the-loop*, ou seja, testes de *hardware* na unidade de voo com informações do aerofólio geradas em simulação computacional. Esta simulação fornece os dados relacionados à dinâmica de voo do aerofólio cabeado, tais como posição (coordenadas esféricas), velocidade, tração e velocidade de desenrolamento do cabo, velocidade do vento etc. Algumas dessas variáveis podem ser visualizadas na Figura 22, que ilustra o modelo de ponto de massa de um aerofólio em coordenadas polares e cartesianas.

Figura 22 – Modelo de ponto de massa do aerofólio em coordenadas polares e cartesianas



Fonte: (OLIVEIRA, 2016)

Na Figura 22 é possível visualizar o vetor de vento nominal v_w , bem como o ângulo polar θ , azimute ϕ e a distância radial r . O aerogerador desenvolvido no projeto UFSCkite utiliza uma configuração de quatro cabos (dois de tração e dois de comando) que saem da unidade de voo em direção ao aerofólio. Conseqüentemente, algumas variáveis adicionais devem ser mencionadas. As duas principais são os comandos de *de-power* e *steering*. O primeiro comando emprega um servomotor para reduzir igualmente o comprimento dos dois cabos de comando que saem da unidade de voo e conectam-se ao bordo de fuga (região “traseira”) do aerofólio. Isto ocasiona uma alteração no ângulo de ataque do aerofólio o que, por sua vez, modifica a sua eficiência aerodinâmica. O uso deste comando é fundamental na transição entre as fases de geração e recolhimento, impactando a eficiência do aerogerador. Já o comando de *steering* permite manobrar o aerofólio em um plano perpendicular ao vetor \mathbf{r} , ou seja, realizar a trajetória de lemniscata, por exemplo. Este comando consiste em gerar, por meio de um segundo servomotor, uma diferença entre o comprimento dos dois cabos de comando já mencionados. A configuração de quatro cabos do aerofólio pode ser visualizada juntamente com as variáveis recém descritas na Figura 23.

Figura 23 – Modelo de quatro cabos e variáveis *depower* e *steering*

Fonte: (OLIVEIRA, 2016)

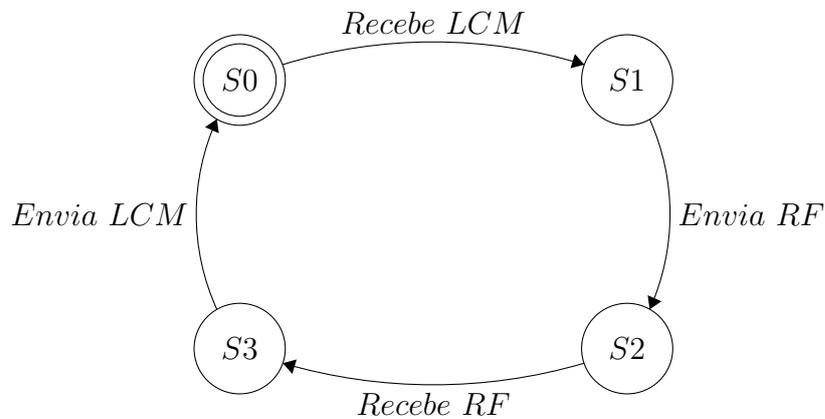
Por fim, é importante frisar que este trabalho foi desenvolvido enquanto o aerogerador do projeto UFSCkite estava em fase de testes do tipo *Hardware-in-the-loop*, portanto o protocolo de comunicação desenvolvido poderá sofrer modificações futuras como preparação para que o aerogerador possa operar em campo.

4.2 Modelagem do protocolo

Uma primeira atividade para estabelecer o comportamento adequado para o protocolo de comunicação é o desenvolvimento de uma máquina de estados, que é demonstrada na Figura 24. Considerando que esta máquina de estados representa o protocolo de comunicação da unidade de solo, temos o estado inicial “S0” do protocolo de comunicação, onde o mesmo espera receber uma mensagem via LCM dos outros módulos da estrutura de software para ser enviada via radiofrequência para a unidade de voo, passando assim para o estado “S1”. Neste estado “S1” é aguardado o envio via radiofrequência através do módulo *swarm bee* da unidade de solo, que leva a máquina de estados para o estado

“S2”. Neste estado “S2”, é esperado o recebimento de uma mensagem via radiofrequência do outro módulo *swarm bee* da unidade de voo, que leva a máquina para o estado “S3” onde deve ser realizado o envio da mensagem recebida via radiofrequência para os outros módulos da estrutura de software da unidade de solo via LCM, retornando a máquina de estados para o estado inicial “S0”.

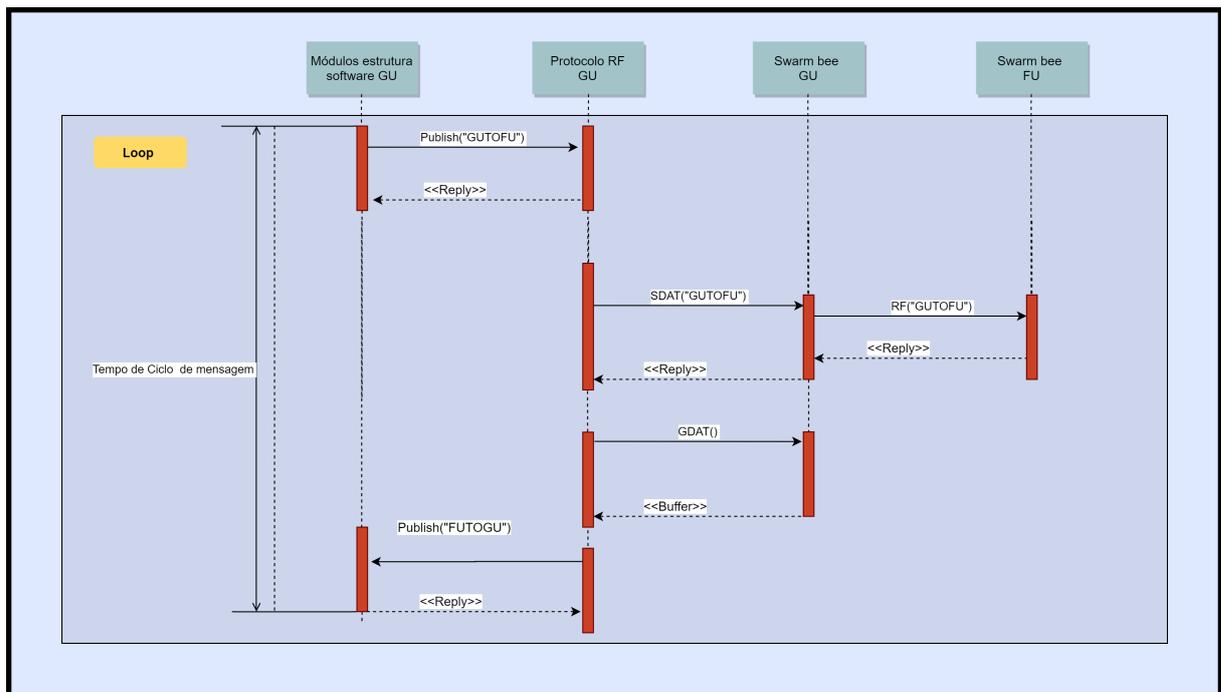
Figura 24 – Máquina de estados do protocolo de comunicação



Fonte: original

Outra atividade desenvolvida durante o desenvolvimento do protocolo de comunicação foi a de conceber o comportamento desejado do protocolo através de diagramas (projeto). O principal diagrama utilizado foi o diagrama de sequência UML, que é muito útil para visualizar a troca desejada de informações entre os agentes envolvidos no protocolo de comunicação. Este diagrama pode ser visualizado na Figura 25.

Figura 25 – Diagrama de sequência UML para o protocolo de comunicação da unidade de solo



Fonte: original

Na Figura 25 é possível visualizar a sequência de troca de informações estabelecida neste protocolo de comunicação entre as unidade de solo e de voo. Primeiramente, é ideal que um pacote “GUTOFU” que se deseja enviar para a unidade de voo seja enviado para o protocolo via *Publish-Subscribe*. Este pacote é enviado da unidade de solo (GU) pela USART para o módulo *swarm bee* utilizando o comando de envio de dados “SDAT”, com os dados que devem ser enviados para a unidade de voo. Após isso, o módulo de radiofrequência realiza de fato a transferência destes dados via ondas de rádio e devolve um “Reply” para o protocolo de comunicação em Python. Tendo realizado o envio de informações via radiofrequência, resta ler o *buffer* do módulo de radiofrequência para verificar se foi recebida alguma informação via radiofrequência. Isto é realizado pelo comando “GDAT”, após o qual os dados lidos são enviados via comunicação *Publish-Subscribe* para os módulos em C da estrutura de *software* da unidade de solo.

4.3 Elaboração do pacote de dados

Levando em conta que o objetivo geral do trabalho é estabelecer uma comunicação via radiofrequência entre as unidades de voo e de solo, deve-se estabelecer quais informações devem ser trocadas entre tais unidades. Considerando que cada uma tem suas responsabilidades no funcionamento do aerogerador e, portanto, interesse em diferentes

variáveis, é de simples entendimento que existirão dois pacotes de mensagem, um para envio da unidade de solo à de voo, e o outro no sentido contrário.

Entretanto é preciso ter em mente as limitações de *hardware* e considerar que o pacote de dados deve ser alocado em um *frame* que será transportado via radiofrequência pelos módulos *swarm bee*. Os módulos de radiofrequência utilizados têm um limite de 128 bytes para o *payload* que será transportado via radiofrequência.

Um esboço inicial destes dois pacotes de dados é apresentado na Figura 26. Nela é possível visualizar as estruturas de dados desenvolvidas em arquivos da biblioteca LCM, tratando-se dos pacotes: (a) “FlightToGround”, ou seja, do pacote de dados que tem como remetente a unidade de voo (FU) e destinatário a unidade de solo (GU); (b) do pacote “GroundToFlight”, que tem como remetente a unidade de solo (GU) e como destinatário, a unidade de voo (FU). As variáveis do pacote “FlightToGround” são as de interesse da unidade de solo e, por consequente, do supervisor SCADA, tendo em vista que este recebe as variáveis via um servidor OPC da unidade de solo. Estas variáveis dizem respeito à dinâmica do voo, como por exemplo os valores dos comandos de *depower* e *steering*, a quantidade de cabo (metros) disponível em carretel e os ângulos polar (θ) e azimutal (ϕ), bem como suas derivadas. Já as variáveis do pacote “GroundToFlight” são aquelas de interesse da unidade de voo e, por conseguinte, do controlador que está encapsulado na estrutura de *software* da unidade de voo, tais como o ganho K_p do controle proporcional do vetor da direção de voo.

Figura 26 – Esboço inicial dos pacotes de dados LCM

<pre> package FlightToGround; struct ftgoutput { int16_t theta; int16_t thetaponto; int16_t phi; int16_t phiponto; int16_t comp_cabo; int16_t mi_steering; int16_t mi_depower; } </pre>	<pre> package GroundToFlight; struct gtfoutput { int16_t theta_l; int16_t phi_atrac; int16_t comp_inicial; int8_t Kp; int16_t Vw; int16_t rponto; int16_t mi_steering; int16_t mi_depower; int8_t maq_estados; } </pre>
--	--

Fonte: original

Entretanto, rapidamente foi percebido que estes dois pacotes, tais como foram concebidos, não iriam satisfazer as necessidades de informações de ambos os lados. Por

exemplo, foi percebido que para realizar o controle manual do aerofólio via *joystick* devemos transportar algumas variáveis relacionadas ao *joystick* para a unidade de voo. Outro exemplo foi a constatação de que a máquina de estados é executada na unidade de voo e então as variáveis *booleanas* devem ser enviadas para a unidade de solo. A partir de então, novas versões destes pacotes foram estabelecidas e se mantiveram durante o restante do desenvolvimento deste trabalho, como demonstra a Figura 27. Nesta versão final, o pacote “fu_to_gu” representa os dados de interesse da unidade de solo (GU) e do supervisor. Já o pacote “gu_to_fu” corresponde aos dados de interesse da unidade de voo (FU). Em sua versão final, o pacote “fu_to_gu” inclui novas variáveis importantes tais como a força de tração do cabo principal, bem como variáveis booleanas associadas a uma máquina de estados que rege o comportamento desejado da unidade de voo.

Figura 27 – Versão final dos pacotes de dados LCM

```
package fu_to_gu;

struct ftgpayload {
    double steering_reference;
    double attractor_signal;

    boolean is_active;
    boolean is_passive;
    boolean is_standby;
    boolean is_setup;
    boolean is_manual_operation;
    boolean is_auto_operation;
    boolean is_failure;

    double theta;
    double phi;
    double theta_d;
    double phi_d;
    double cable_length;
    double traction_force;
}

package gu_to_fu;

struct gtfpayload {
    double tether_phi;
    double tether_phi_d;
    double tether_theta;
    double tether_theta_d;

    double traction;
    int16_t joys;
    int16_t joyd;
    int8_t joyau;
    int8_t joyack;
    int8_t joyman;
    int16_t trac_min;
    double theta_max;
    int16_t ini_depower;
    double theta_center;
    double phi_atr;
    double Kp;

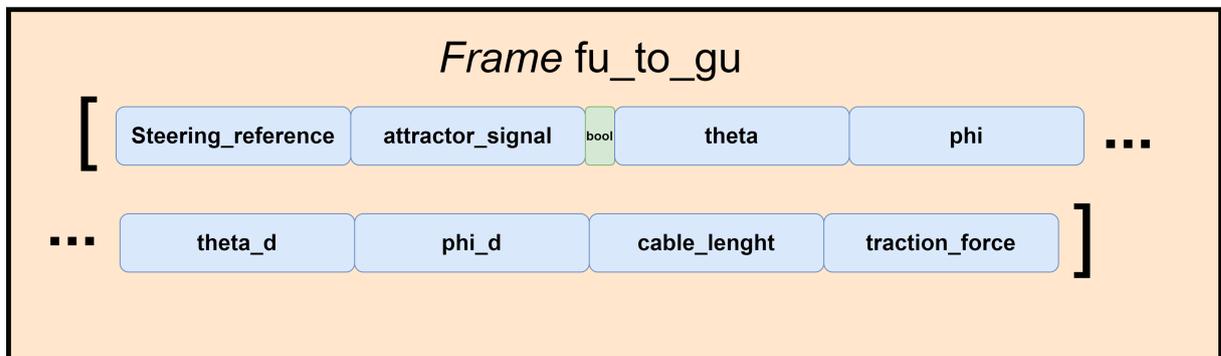
    double depower_value;
    double wind_speed;
    double tether_speed;
    double tether_length;
    double wind_filtered_direction;
}
```

Fonte: original

Após um cálculo rápido, é visível que o pacote fu_to_gu ocupa 8 variáveis *double*, onde cada uma ocupa 8 *bytes* totalizando 64 *bytes*. Além disso temos mais 7 valores do tipo *boolean* que cabem em 1 *byte*, totalizando 65 *bytes* e portanto cabendo tranquilamente

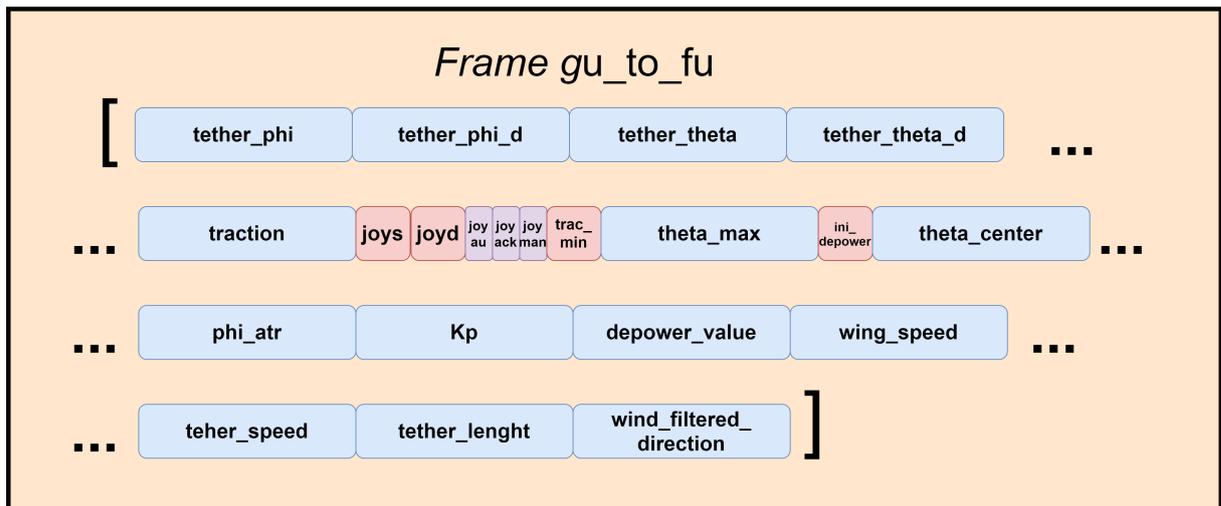
no *frame* máximo de 128 *bytes*. Já o pacote *gu_to_fu* ocupa 123 *bytes* e, embora perto do limite, ainda cabe no *frame* máximo mencionado. Estes encapsulamentos podem ser vistos nas Figuras 28 e 29, nas quais temos os dois pacotes distribuídos pelos *frames* de 65 e 123 *bytes*, respectivamente.

Figura 28 – *Frame* de 65 *bytes* do pacote “*fu_to_gu*”



Fonte: original

Figura 29 – *Frame* de 123 *bytes* do pacote “*gu_to_fu*”



Fonte: original

Na Figura 28 é possível visualizar o *frame* “*fu_to_gu*”, montado praticamente de variáveis “*double*”, exceto pelas variáveis booleanas associadas à máquina de estados. Foi decidido durante o projeto acumular estas variáveis em uma variável de 1 *byte* e reconstruir as variáveis booleanas no lado receptor, neste caso, a unidade de solo. Já o *frame* “*gu_to_fu*” é um pouco mais complexo, tanto devido à sua extensão como ao seu uso de variáveis de tamanho variado. Temos variáveis *double* que ocupam 8 *bytes*, *int_16t* ocupando 2 *bytes* e também *int_8t* ocupando 1 *byte*. Tais variáveis referem-se,

por exemplo, aos *setpoints* de posição dos servomotores de *steering* e *depower*, gerados a partir de um *joystick* utilizado para o controle manual do aerofólio, valor do ganho de controle K_p e da posição azimutal ϕ_{atr} dos atratores do controle de voo, etc.

É importante lembrar que, como mencionado na Seção 4.1, o protocolo atual de comunicação foi desenvolvido com a configuração de testes *hardware-in-the-loop*. Nesta configuração temos a unidade de voo em solo, cujos sensores são substituídos por valores que advêm de uma simulação computacional do voo do aerofólio. Como esta não é a configuração final do aerogerador, é natural esperar que este protocolo possa ser alterado conforme o avanço do projeto. Portanto, o que está descrito nesta monografia se refere ao estado atual de desenvolvimento do aerogerador do projeto UFSCkite.

4.4 Configuração do módulo *swarm bee* em Python

Como descrito na Seção 3.6, já havia sido desenvolvido código para se trabalhar com os módulos *swarm bee* na linguagem Python. Portanto uma estrutura de acesso aos módulos de radiofrequência já era disponível previamente ao início do desenvolvimento deste trabalho. Os módulos *swarm bee* possuem diversos pinos para interação com o usuário, mas os principais utilizados são os seguintes:

- USART_RX – pino pelo qual é realizada a leitura de dados enviados via USART para o módulo *swarm bee*;
- USART_TX – pino pelo qual é realizado o envio de dados enviados via USART do módulo *swarm bee*;
- A_MODE – pino que controla o modo automático do módulo;
- MOD_EN – pino que controla a circulação de corrente no módulo;
- VCC – pino de alimentação;
- GND – pino de aterramento.

Primeiramente, para realizar a comunicação via USART deve-se informar para microcomputador embarcado (BeagleCore ou Raspberry Pi) a porta serial que será utilizada para comunicação USART, como pode ser visualizado no código da Figura 30.

Figura 30 – Configuração da porta serial para comunicação USART

```
import subprocess
import signal
import time
import re
from periphery import GPIO, Serial, SerialError, GPIOError

class BaseModule():

    def __init__(self):
        try:
            self.serial = Serial("/dev/ttyO1", baudrate=115200)
            print("Serial available.")
            ...
            ...
```

Fonte: original

Em relação aos pinos A_MODE e MOD_EN, a configuração essencial para o funcionamento do módulo *swarm bee*, já disponível previamente em código Python, pode ser visualizada na Figura 31. O método de classe “configureSwarmBeePins” atribui nível alto de tensão a um pino da placa de desenvolvimento e, a outro pino, nível baixo. Estes dois pinos se referem à configuração dos pinos A_MODE e MOD_EN dos módulos de radiofrequência. A_MODE se refere ao modo automático de funcionamento do módulo, que não é interessante para o desenvolvimento deste trabalho pois se trata de uma rotina desenvolvida pela empresa Nanotron que realiza automaticamente medições de distância entre os módulos de radiofrequência e transmite via radiofrequência estas distâncias. Portanto, o pino A_MODE deve ser atribuído o nível baixo de tensão. Já o pino MOD_EN controla a liberação de corrente para o módulo, logo deve ser atribuído um nível alto de tensão.

Figura 31 – Configuração dos pinos A_MODE e MOD_EN

```
def configureSwarmBeePins(self):
    done = False
    counter = 0
    configWait = 0.3
    while not done:
        if counter > 10:
            print("Could not configure Swarm Bee pins.")
            return -1
        try:
            self.swarmBeePinModEn = GPIO(67, "high")
            self.swarmBeePinAMode = GPIO(68, "low")
            done = True
            time.sleep(configWait)
            print("Configuration of Swarm Bee pins completed.")
        except PermissionError:
            counter += 1
            print("Permission Error: could not configure Swarm Bee
                  pins." +
                  " Retrying.")
            ...
            ...
```

Fonte: original

4.5 Implementação do protocolo em Python

Uma vez estabelecida a configuração dos pinos, se pode partir para a implementação do protocolo de comunicação. Uma codificação inicial é apresentada na Figura 32. Nela, é definida uma mensagem hexadecimal a ser enviada via radiofrequência, neste caso “FFFFFFFF”, é realizado em sequência o cálculo do tamanho da mensagem (parâmetro importante para o uso do comando SDAT mencionado na Figura 17), informado o ID do módulo de radiofrequência da unidade de voo, e a mensagem é enviada utilizando o método de classe “sendDataToNode”, que encapsula as linhas de código necessárias para enviar o comando SDAT para o módulo *swarm bee*. Com isso, torna-se possível realizar uma leitura do que se encontra no *buffer* de mensagens do módulo de radiofrequência com o método de classe “getData”, que contém o código necessário para realizar o comando GDAT (leitura do *buffer* do módulo *swarm bee*). Este valor lido do *buffer* é alocado na variável “received”, para então ser realizado um teste simplificado com o objetivo de verificar se o que foi recebido contém dados. Este teste é realizado verificando o comprimento da variável “received”, que se obter dados deve ter um comprimento maior que 10. Quando não se tem nada escrito no *buffer* a variável “received” armazena a *string* "00" e quando

existe dados na variável “received” ela armazena, por exemplo, a *string* "0A 00000000000B 00010002000304000500060007000809", onde temos o tamanho da mensagem, ID do remetente e a mensagem, respectivamente.

Figura 32 – Esboço da implementação do protocolo de comunicação na unidade de solo

```
def GroundDataCycle(self):
    i = 0
    while(i<=30):
        i = i + 1
        messageToSend = "FFFFFFFF"
        messageLength = str(len(messageToSend)//2)
        flightID = self.controlledModule.intToFullHexStr(2)
        self.controlledModule.sendDataToNode(0, flightID,
            messageLength, messageToSend)

        received = self.controlledModule.getData()
        if(len(received)>5):
            print("recebi msg da flight ---> enviar fu_to_gu para
                modulos da ground")
            ...
            ...
```

Fonte: original

Embora este seja o essencial do protocolo de comunicação, diversos esforços adicionais tornaram-se necessários ao longo do desenvolvimento deste trabalho como, por exemplo:

- integração com a estrutura de *software* e, portanto, implementação de um *Publish-Subscribe* para fazer a interface Python-C;
- construção da mensagem a ser enviada em função das variáveis recebidas pelos módulos em C;
- realização do tratamento da mensagem recebida, consistindo na alocação de variáveis e posterior *publish* destas variáveis para quem estiver interessado (módulos em C).
- implementação de *threads* para tornar possível o recebimento assíncrono de mensagens via *subscribe*, juntamente com o funcionamento de uma rotina que envia e recebe dados via SDAT/GDAT.

4.6 Implementação final do protocolo

Com as adaptações necessárias do protocolo em mente (mencionadas na Seção 4.5), novas implementações do protocolo de comunicação foram surgindo até a sua última versão, cuja síntese consta na Figura 33.

Entre as principais modificações com relação à implementação anterior, uma inscrição é feita logo no início do código, no tipo de mensagem “GUTOFUtopy”, que corresponde às mensagens publicadas pelos módulos em linguagem C e carrega as variáveis que os módulos da estrutura de *software* da unidade de solo desejam enviar para a unidade de voo via radiofrequência. Esta inscrição passa como parâmetro o tipo de mensagem desejado e o manipulador da mensagem – função que deve ser chamada quando uma mensagem do tipo desejado é recebida. Este manipulador não está demonstrado na Figura 33, entretanto ele simplesmente utiliza funções fornecidas pela biblioteca LCM e aloca variáveis para armazenar os dados que o pacote contém.

Figura 33 – Implementação simplificada do protocolo de comunicação na unidade de solo

```

def GroundDataCycle(self):
    i = 0
    subscription = lc.subscribe("GUTOFUtopy", gtf_handler)
    x = threading.Thread(target=sub_function)
    x.start()

    while(i<=30):
        i = i + 1
        messageToSend =
            self.controlledModule.intToHexStr(tether_phi,6)+
            self.controlledModule.intToHexStr(phi_atrac,6)+
            self.controlledModule.intToHexStr(comp_inicial,6)+...

        messageLength = str(hex(len(messageToSend)//2)[2:])
        flightID = self.controlledModule.intToFullHexStr(22)
        self.controlledModule.sendDataToNode(0, flightID,
            messageLength, messageToSend)

        received = self.controlledModule.getData()
        if(len(received)>10):
            print("recebi msg da flight ---> pub ftg para ground")
            msgpub = ftgpayload()
            ... # TRATAMENTO DA MENSAGEM RECEBIDA VIA RF
            ... # PARA ENVIO DELA VIA PUB/SUB
            lc.publish("FlightToGround", msgpub.encode())

    global stop_threads
    stop_threads = True
    x.join()
    lc.unsubscribe(subscription)
    return 0

```

Fonte: original

Além disso, nesta implementação aprimorada do protocolo de comunicação, a mensagem a ser enviada via radiofrequência é construída a partir das variáveis recebidas do pacote “GUTOFUtopy” e tratada pelo manipulador “gtf_handler”. As variáveis têm que ser transformadas em *strings* de hexadecimais, pois este é o formato exigido das entradas do módulo *swarm_bee*. Uma leve alteração também foi feita no cálculo do tamanho da mensagem a ser enviada, que era calculado erroneamente na base decimal, e deve ser também uma variável *string* em base hexadecimal.

Após estas conversões, a mensagem recebida via radiofrequência é tratada. Isto é realizado atribuindo-se valores para variáveis baseados na varredura da *string* “received”. A figura 33 mostra uma versão simplificada deste processo devido ao número grande de linhas

de código para construir as mensagens a serem enviadas e tratar as mensagens recebidas. Por fim, quando o final da execução do código é alcançado, é atribuído o valor *TRUE* a uma *flag* “stop_threads”, sinalizando o encerramento de *threads* adicionais. Juntamente com este processo é realizada uma “liberação” da inscrição feita em mensagens do tipo “GUTOFUtopy” e o protocolo é encerrado.

A implementação ilustrada na Figura 33 refere-se ao protocolo desenvolvido para a unidade de solo, com mensagens destinadas à unidade de voo. Para a direção contrária a implementação segue a mesma estrutura, com diferenças apenas nas variáveis veiculadas.

4.7 Novo módulo “newradio”

Para complementar o protocolo de comunicação via radiofrequência implementado em Python, foi desenvolvido um novo módulo em C. Este desempenha as seguintes funções:

- inscrever-se em mensagens que os outros módulos em linguagem C desejam enviar via radiofrequência;
- publicar estas mensagens utilizando a biblioteca LCM, por exemplo utilizando o pacote “GUTOFUtopy” apresentado na Figura 33;
- executar o código de protocolo de comunicação desenvolvido em Python, assim encapsulando em um módulo escrito em linguagem C o funcionamento do protocolo de mensagens desenvolvido em Python.

A implementação deste novo módulo pode ser visualizada na Figura 34, contendo em código C as funções recém descritas acima.

Figura 34 – Implementação do módulo “newradio” da estrutura de software da GU

```
1 void receive_gutofu(raw_msg *raw, char *channel,
2   gu_to_fu_payload *msg, newradio_module *m) {
3     // CRIAR A MENSAGEM PARA SER ENVIADA PARA
4     PYTHON
5     gu_to_fu_payload msgtopy;
6     msgtopy.tether_phi = msg->tether_phi;
7     msgtopy.tether_phi_d = msg->tether_phi_d;
8     ...
9     gu_to_fu_payload_publish(lcm, "GUTOFUcopy",
10    &msgtopy);
11 }
12 void *subscribeftg(void *joca){
13
14 fu_to_gu_payload_subscribe(lcm, "FlightToGround",
15   &my_handler, NULL);
16 }
17 void setup(newradio_module *m) {
18     int j = 1;
19     pthread_t subscribeftg_thread;
20
21     module_subscribe(m, gu_to_fu_payload,
22       "GUTOFU", receive_gutofu);
23 }
```

Fonte: original

Com o módulo ilustrado na Figura 34 é criado uma *thread* adicional para se inscrever no pacote “FlightToGround”, na linha 19, que representa os pacotes recebidos via radiofrequência pela unidade de solo. É também realizada a inscrição no pacote “GUTOFU” durante o *setup* do módulo, na linha 21. Isto serve para atualizar o módulo com as variáveis que precisam ser enviadas para a unidade de voo. Com o tratador da mensagem “GUTOFU”, denominado “receive_gutofu”, é realizada a leitura dos dados que devem ser enviados via radiofrequência e, posteriormente, a publicação para o código em Python na linha 9.

5 Resultados

Com o desenvolvimento e implementação do protocolo de comunicação realizados, pode-se partir para o recolhimento e análise dos resultados, a serem tratados neste capítulo.

5.1 Tempo de ciclo de mensagem

A averiguação do tempo necessário para realizar um ciclo de mensagem, caracterizado pelo envio e recebimento de uma mensagem via radiofrequência, é algo de suma importância tendo em vista que este tempo deve ser pequeno o suficiente para não prejudicar as malhas de controle implementadas na estrutura de software da unidade de voo. Para esta averiguação foi realizado o registro dos tempos ao final de cada rotina de envio/recebimento de mensagens via RF na unidade de solo (que envia um pacote de 123 *bytes*). Posteriormente, esse arquivo de registro foi analisado, com resultados apresentados na Figura 35.

Figura 35 – Tempos de ciclo de mensagem

```

---
Initialization Completed
---

2020-02-12 21:15:36.834692
2020-02-12 21:15:36.868785
2020-02-12 21:15:36.923452
2020-02-12 21:15:36.978726
2020-02-12 21:15:37.033547
2020-02-12 21:15:37.088334
2020-02-12 21:15:37.143020
2020-02-12 21:15:37.176898
2020-02-12 21:15:37.231759
2020-02-12 21:15:37.286413
...
...

```

Fonte: original

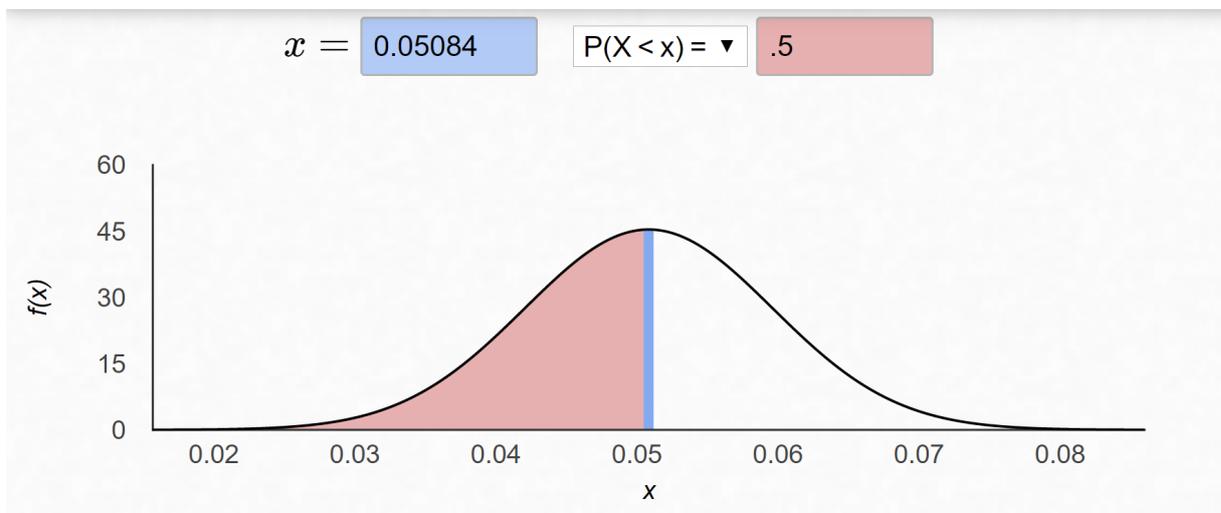
Por inspeção, nota-se que o tempo médio de ciclo de mensagem é algo em torno de 51 ms. Este valor representa o protocolo onde é realizado um envio e em sequência uma leitura do *buffer* do módulo *swarm bee*, entretanto se desejado diminuir o tempo entre

envios seria possível alterar o sequenciamento do protocolo de forma a priorizar envios. Outra possibilidade seria a adição de uma *thread* para a recepção de mensagens, assim existiria uma *thread* para o envio de mensagens e outra para o recebimento.

5.1.1 Distribuição normal do tempo de ciclo de mensagem

Para se caracterizar a duração provável de um ciclo de mensagem, pode-se determinar experimentalmente uma função densidade de probabilidade. Nos experimentos realizados com um *dataset* de dois mil valores de tempo de ciclo de mensagem, a função densidade de probabilidade mostrou-se ser gaussiana (normal), conforme resultados na Figura 36.

Figura 36 – Distribuição normal do tempo de ciclo de mensagem



Fonte: original

5.1.2 Variação de acordo com tamanho da mensagem

Algo que foi percebido durante os testes de levantamento do tempo de ciclo de mensagem é que este tempo varia de acordo com o tamanho de dados enviados no pacote. Por exemplo, o módulo *swarm bee* suporta um pacote de tamanho máximo de 128 *bytes*, e quando os pacotes enviados/recebidos via radiofrequência são de 10 *bytes* o tempo de ciclo de mensagem é consideravelmente menor do que quando se envia/recebe pacotes de 128 *bytes*. Este comportamento pode ser visualizado na Figura 37, onde temos os arquivos de registro de tempo nas duas situações mencionadas acima.

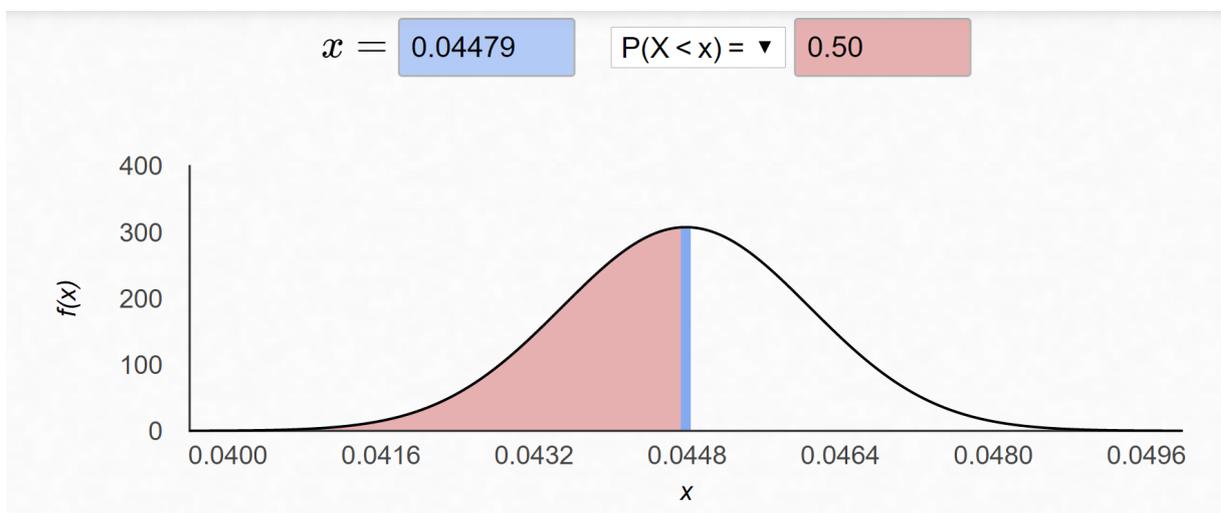
Figura 37 – Tempos de ciclo de mensagem para pacotes de 10 e 128 bytes, respectivamente

<pre> --- Initialization Completed --- 2020-02-12 18:38:03.295044 2020-02-12 18:38:03.339874 2020-02-12 18:38:03.384412 2020-02-12 18:38:03.428946 2020-02-12 18:38:03.473218 2020-02-12 18:38:03.518217 2020-02-12 18:38:03.563624 2020-02-12 18:38:03.608440 </pre>	<pre> --- Initialization Completed --- 2020-02-12 18:29:02.951556 2020-02-12 18:29:02.985483 2020-02-12 18:29:03.049166 2020-02-12 18:29:03.082937 2020-02-12 18:29:03.148880 2020-02-12 18:29:03.182557 2020-02-12 18:29:03.246839 2020-02-12 18:29:03.280502 </pre>
--	--

Fonte: original

Pela Figura 37 percebe-se que o tempo médio de ciclo de mensagem para a situação com um pacote de 10 *bytes* é de 44 ms, enquanto para o pacote de 128 *bytes* é de aproximadamente 51 ms. Esta variação pode ser atribuída à lógica de baixo nível utilizada pelos módulos de radiofrequência. Entretanto esta lógica não é disponível para o usuário/desenvolvedor, que interage com o módulo através da API fornecida. Uma segunda distribuição normal pode ser construída baseada neste *dataset* para avaliar a probabilidade do tempo de ciclo de mensagem para pacotes de 10 *bytes*, como apresentado na Figura 38.

Figura 38 – Distribuição normal do tempo de ciclo de mensagem para pacotes de 10 bytes

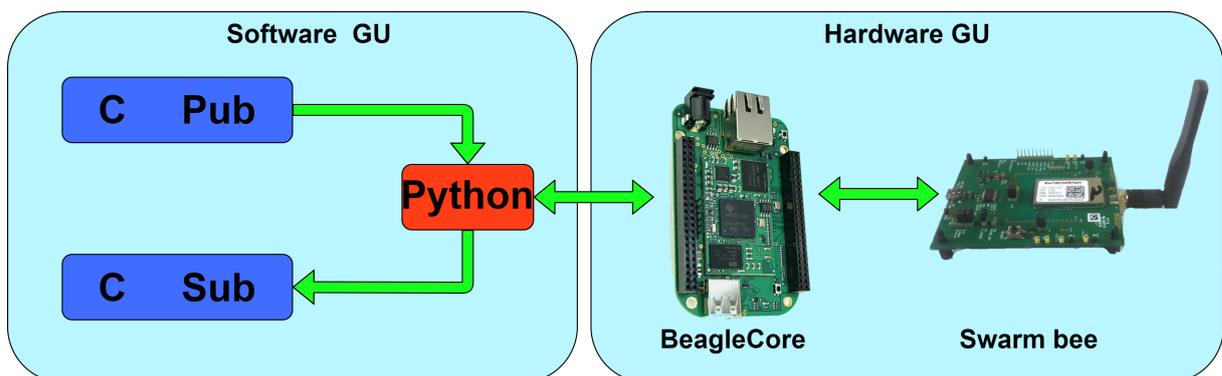


Fonte: original

5.2 Comunicação LCM

Para testar a comunicação via *Publish-Subscribe* foram desenvolvidos alguns códigos de inscrição e publicação para trabalharem em conjunto com os códigos desenvolvidos em Python. Com o auxílio destes códigos, é possível avaliar se os protocolos de comunicação estão publicando as mensagens recebidas via radiofrequência e se estão atualizando os dados a serem enviados à medida em que estes são publicados por terceiros. A configuração destes testes pode ser visualizada na Figura 39, onde temos os códigos em C de publicação e inscrição, o protocolo de comunicação desenvolvido em Python, o microcomputador BeagleCore e o módulo *swarm bee* de radiofrequência.

Figura 39 – Configuração para testes da comunicação LCM na unidade de solo



Fonte: original

O primeiro teste realizado avalia se a comunicação entre o protocolo implementado em Python está se comunicando adequadamente com a estrutura de *software* em C (este comportamento está diagramado na Figura 20). O resultado desse teste é apresentado na Figura 40, na qual é ilustrado o protocolo de comunicação da Figura 33 informando quais comandos estão sendo enviados e lidos dos módulos de radiofrequência *swarm bee*. Além disso, existe o código de inscrição que se inscreve via LCM no pacote de dados “GroundToFlight” e que avisa quando receber este pacote.

Figura 40 – Confirmação da comunicação *Publish-Subscribe* de Python para C

<pre> 2020-02-12 16:31:36.583237 Command Sent: SDAT 0 00000000000B f 00000000000000000000000000000001 Reply: =1 Command Sent: GDAT Reply: =7B 00000000000B 0000000000000000000000000000... recebi pacote RF da GU ---> pub GUTOFU para modulos da FU 2020-02-12 16:31:36.639382 </pre>	<pre> Esperando msg GroundToFlight... 2020-02-12 16:31:36.637096 2020-02-12 16:31:36.649338 2020-02-12 16:31:36.659882 2020-02-12 16:31:36.670402 Received message "GroundToFlight" </pre>
Protocolo RF da unidade de voo	Inscrito do pacote “GroundToFlight”

Fonte: original

Na figura 40 pode-se ver que no instante de tempo “16:31:36.639382” o protocolo de comunicação realiza um “GDAT” e recebe um pacote da unidade de solo. Este pacote deve ser publicado para que os outros módulos da estrutura de *software* da unidade de voo possam receber estas variáveis atualizadas. Com o auxílio de um código de inscrição no pacote ‘GroundToFlight’, que representa os módulos da estrutura de *software* da unidade de voo que desejam receber atualizações das variáveis do solo, é possível visualizar que em “16:31:36.670402” este inscrito recebe uma atualização do pacote. Assim verifica-se que a comunicação *Publish-Subscribe*, que parte do protocolo de comunicação em Python para a estrutura de módulos em C, funciona corretamente.

Outra face da comunicação *Publish-Subscribe* que deve ser avaliada, para garantir um funcionamento adequado do protocolo de comunicação, é a comunicação via *Publish-Subscribe* que parte dos módulos em C e informa para o protocolo em Python as variáveis que devem ser transportadas pelo canal de radiofrequência. Esta comunicação é ilustrada no diagrama da Figura 20, enquanto os resultados podem ser visualizados na Figura 41.

Figura 41 – Confirmação da comunicação *Publish-Subscribe* de C para Python

<pre> modulo GU publica GUTOFU 2020-02-12 21:39:42.806729 modulo GU publica GUTOFU 2020-02-12 21:39:42.914118 modulo GU publica GUTOFU 2020-02-12 21:39:43.016119 modulo GU publica GUTOFU 2020-02-12 21:39:43.118656 modulo GU publica GUTOFU 2020-02-12 21:39:43.220995 </pre>	<pre> 2020-02-12 21:39:42.773431 Command Sent: SDAT 0 000000000016 7b 000000000000000000000000... Reply: Command Sent: GDAT Recebi msg de modulos C para enviar GUTOFU via RF Reply: 2020-02-12 21:39:42.808866 </pre>
Módulo publicador em C	Protocolo RF da unidade de solo

Fonte: original

Na figura 41 percebe-se que no instante de tempo “21:39:42.808866” o protocolo de comunicação na unidade de solo recebe um pacote LCM de “GUTOFU”. Este pacote se refere ao pacote enviado pelo código publicador no instante “21:39:42.806729”. Este resultado nos permite concluir que a comunicação entre os módulos em C da estrutura de *software* e o protocolo de comunicação em Python está funcionando adequadamente.

5.3 Transporte dos dados

Por fim, é possível verificar se a medida que o *frame* de radiofrequência é alterado e enviado pelo transmissor, este *frame* é recebido adequadamente pelo receptor. Isto pode ser visualizado na Figura 42.

Figura 42 – Confirmação do transporte de dados entre unidade de solo e de voo

<pre>2020-02-12 21:57:42.459149 Sent: SDAT 0 000000000016 0A 00000000000000000000000000000000 Reply: =0 2020-02-12 21:57:43.106451 Sent: SDAT 0 000000000016 0A 00010002000304000500060007000809 Reply: =0 2020-02-12 21:57:43.759241 Sent: SDAT 0 000000000016 0A 0002000300040500060007000800090A Reply: =0</pre>	<pre>2020-02-12 21:57:42.622817 Command Sent: GDAT Reply: =0A 00000000000B 00000000000000000000000000000000 2020-02-12 21:57:43.306732 Command Sent: GDAT Reply: =0A 00000000000B 00010002000304000500060007000809 2020-02-12 21:57:43.984733 Command Sent: GDAT Reply: =0A 00000000000B 0002000300040500060007000800090A</pre>
Protocolo RF da unidade de solo	Protocolo RF da unidade de voo

Fonte: original

Na Figura 42 pode ser visto que na hora “2020 -02 -12 21:57:43.106451” um pacote com variáveis atualizadas é enviado via radiofrequência pela unidade de solo. Este pacote é recebido pela unidade de voo na hora “2020 -02 -12 21:57:43.306732”. Este resultado comprova que a comunicação via radiofrequência está funcionando adequadamente, juntamente com a comunicação *Publish-Subscribe*.

5.4 Perda de pacotes

Um dos principais agravantes da comunicação sem fio é a perda dos pacotes enviados. Durante o desenvolvimento do protocolo duas causas deste problema foram constatadas e seguem:

- Perda de pacotes por corrompimento : ocorre quando os pacotes transmitidos chegam no receptor alterados.
- Perda de pacotes por *timeout* : quando os pacotes demoram muito tempo para chegar ao receptor e são descartados.

A perda de pacotes por corrompimento, no âmbito do protocolo de comunicação desenvolvido, ocorre esporadicamente no cenário onde o módulo transmissor *swarm bee* realiza um “SDAT” e o módulo receptor um “GDAT” ao mesmo tempo. O resultado deste cenário é a recepção de um pacote corrompido, como ilustra a Figura 43.

Figura 43 – Corrupção dos pacotes

<pre> 2020-02-12 22:38:11.852456 Sent: SDAT 0 000000000016 0A 00000000000000000000000000000000 Reply: =0 2020-02-12 22:38:12.482149 Sent: SDAT 0 000000000016 0A 00010002000304000500060007000809 Reply: =0 2020-02-12 22:38:13.084692 Sent: SDAT 0 000000000016 0A 0002000300040500060007000800090A Reply: =0 </pre>	<pre> 2020-02-12 22:38:11.876213 Command Sent: GDAT Reply: =0A 00000000000B 00000000000000000000000000000000 2020-02-12 22:38:12.483657 Command Sent: GDAT Reply: =0A 00000000000B 000100FA48B96DD75FA29431CE54B3A2 2020-02-12 22:38:13.112655 Command Sent: GDAT Reply: =0A 00000000000B 0002000300040500060007000800090A </pre>
Protocolo RF da unidade de solo	Protocolo RF da unidade de voo

Fonte: original

Como é visível na Figura 43, quando o módulo transmissor realiza um “SDAT” ao mesmo tempo que o módulo receptor realiza um “GDAT” em “22:38:12.48” o pacote lido durante o comando “GDAT” se encontra corrompido. Para lidar com esta situação, uma solução simples encontrada foi a adição de uma “chave” ao final do pacote. Deste modo, durante a leitura de *buffer* realizada pelo comando “GDAT” no protocolo de comunicação, é possível verificar com uma boa precisão se o pacote foi corrompido. Por exemplo, é possível adicionar ao final do *frame* do pacote uma *string* hexadecimal “AA” de 1 *byte*. Desta maneira, durante a leitura do *buffer* é possível analisar o último *byte* recebido e caso este for divergente da *string* hexadecimal “AA” o pacote pode ser descartado.

Já no segundo caso, a perda de pacotes por *timeout* ocorre quando os pacotes de dados são perdidos durante a transmissão, chegam atrasados ao seu destino ou nem conseguem alcançar o receptor. Normalmente um dos componentes de um protocolo de comunicação é o tempo de *timeout*, que se trata do tempo que o transmissor espera para receber a mensagem de confirmação do receptor (ACK). Entretanto, este *timeout* já é estabelecido pela API fornecida pela Nanotron juntamente com os módulos de radio-frequência *swarm bee*, este tempo é de 24 μ s. Na Figura 44 é demonstrada uma situação onde o tempo de *timeout* é alcançado e o pacote é descartado para dar continuidade ao protocolo de comunicação.

Figura 44 – Perda de pacotes por *timeout*

<pre>2020-02-12 23:13:39.520485 Sent: SDAT 0 000000000016 0A 00000000000000000000000000000000 Reply: =1 2020-02-12 23:13:40.124485 Sent: SDAT 0 000000000016 0A 00010002000304000500060007000809 Reply: =1 2020-02-12 23:13:40.779658 Sent: SDAT 0 000000000016 0A 0002000300040500060007000800090A Reply: =1</pre>	<pre>2020-02-12 23:13:39.795842 Command Sent: GDAT Reply: =00 2020-02-12 23:13:40.412449 Command Sent: GDAT Reply: =00 2020-02-12 23:13:40.963566 Command Sent: GDAT Reply: =00</pre>
Protocolo RF da unidade de solo	Protocolo RF da unidade de voo

Fonte: original

Como é visível na Figura 44, o protocolo da unidade de solo tenta enviar para a unidade de voo pacotes via radiofrequência entretanto o *timeout* de 24 μ s é alcançado e o pacote é descartado, sinalizado por “Reply: =1”. A perda de pacotes por *timeout* ocorre esporadicamente durante o funcionamento do protocolo de comunicação e devido a estrutura do protocolo não se trata de um problema muito agravante, devido ao fato que os pacotes que são trocados entre unidades de solo e voo possuem estruturas estáticas e são transmitidos de forma intermitente. Portanto, caso algum pacote seja perdido por *timeout*, brevemente outro pacote será enviado com as mesmas variáveis atualizadas.

6 Conclusões e perspectivas

Como pôde-se perceber ao final do Capítulo 5, o objetivo geral de conceber e implementar um protocolo de comunicação via radiofrequência entre as unidades de solo e voo do aerogerador com aerofólio cabeado foi alcançado. Para isto foi preciso primeiramente se familiarizar com os conceitos de comunicação via radiofrequência e a sua implementação através dos módulos de radiofrequência *swarm bee*. Com este protocolo desenvolvido, o grupo de pesquisa UFSCkite pode usufruir de um canal de comunicação entre as duas unidades, algo essencial para o funcionamento do aerogerador. O protocolo se demonstrou hábil em transportar as informações necessárias estipuladas na Seção 4.3 e o transporte de uma maior quantidade de informações é possível com o ajuste dos tipos de dados utilizados no pacote LCM.

O protocolo desenvolvido durante esta monografia será de maior benefício para o projeto UFSCkite após o término dos testes de *hardware-in-the-loop* com a unidade de voo. Dessa forma, com a comunicação via radiofrequência funcionando corretamente, será possível alçar voo com a unidade de voo e usufruir de um canal de comunicação eficiente e robusto entre a unidade de voo e de solo, algo essencial para o funcionamento do aerogerador.

Trabalhos futuros

Existem algumas possibilidades de trabalhos futuros em sequência ao desenvolvido durante esta monografia. Como a comunicação via radiofrequência ainda precisa ser avaliada na situação de operação real, na qual a unidade de voo acompanha o aerofólio cabeado e a unidade de solo gera energia, uma primeira possibilidade de trabalho futuro seria a avaliação desta comunicação nesse cenário real de operação. Outra possibilidade seria a otimização do protocolo desenvolvido durante esta monografia. Melhorando assim os tempos de ciclo de mensagem, redução de perda de pacotes e outras possíveis melhorias.

Referências

- 2020 climate energy package. Acessado em janeiro 2020 em <https://ec.europa.eu/clima/policies/strategies/2020>, 2007. Citado na página 16.
- 2050 long-term strategy. Acessado em janeiro 2020 em <https://ec.europa.eu/clima/policies/strategies/2050>, 2011. Citado na página 16.
- BRASIL-ESCOLA. Interferência de ondas Acessado em janeiro 2020 em <https://brasilecola.uol.com.br/fisica/interferencia-ondas.htm>, 2020. Citado na página 31.
- CHERUBINI, A. et al. Airborne wind energy systems: A review of the technologies. *Renewable and Sustainable Energy Reviews*, v. 51, p. 1461–1476, 2005. Citado na página 16.
- FAGIANO, L. et al. Automatic crosswind flight of tethered wings for airborne wind energy: modeling, control design and experimental results. 2013. Citado na página 36.
- GLOBAL Wind Atlas. Acessado em janeiro 2020 em <https://globalwindatlas.info>, 2020. Citado na página 17.
- HAMID, A. F. B. A. et al. Path loss with doppler shift effect and handover delay analysis on cellular communication for connected car application in urban area. 2019. Citado na página 32.
- INTERNATIONAL-TELECOMMUNICATION-UNION. Acessado em janeiro 2020 em <https://www.itu.int/osg/spu/ni/3G/technology/>, 2020. Citado na página 34.
- KAIGARULA, S. Review of radio propagation properties and applications in different frequency bands. 2013. Citado na página 27.
- MAITRA, A. Early history of wireless communications. 2003. Citado na página 22.
- MARK, J. W.; ZHUANG, W. Wireless communications and networking. 2003. Citado na página 22.
- MAURER, A. R. S. Design of a fast and reliable wireless link for kite power systems. 2012. Citado 3 vezes nas páginas 10, 35 e 36.
- MODERNIST-ENCAUSTIC. Modernist Encaustic Acessado em janeiro 2020 em <http://modernistencaustic.com/portfolio-items/reflection-refraction/>, 2020. Citado na página 29.
- NEETS. NAVY ELECTRICITY AND ELECTRONICS TRAINING SERIES Acessado em janeiro 2020 em <https://www.fcctests.com/neets/Neets.htm>, 1998. Citado na página 30.

- OLIVEIRA, M. D. L. C. de. Airborne wind energy with tethered wings: modeling, analysis and control. Dissertação(Doutorado) UFSC-Florianopolis 2016. Citado 3 vezes nas páginas 19, 45 e 46.
- RAPPAPORT, T. Wireless communications: Principles and practice (2^a ed.). 2002. Citado 2 vezes nas páginas 26 e 33.
- SCHMEHL, R. Airborne wind energy advances in technology development and research. 2018. Citado na página 18.
- SCHMIDT, E. et al. State estimation of a tethered airfoil for monitoring. *IFAC*, 2017. Citado na página 20.
- UGWEJE, O. Radio frequency and wireless communications security. 2004. Citado 2 vezes nas páginas 22 e 28.
- VECTORSTOCK. VectorStock Acessado em janeiro 2020 em <https://www.vectorstock.com/royalty-free-vector/wave-diffraction-vector-10497494>, 2020. Citado na página 30.
- WEISMAN, C. J. The essential guide to rf and wireless (2^a ed.). 2003. Citado na página 24.
- WORLD Wind Energy International. Acessado em janeiro 2020 em <https://library.wwindea.org/global-statistics-2018-preliminary>, 2018. Citado na página 16.
- YACOUB, M. D. Foundations of mobile radio engineering. 1993. Citado na página 33.