UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

Departamento de Engenharia da Computação e Automação

Plataforma de Simulação para Veleiros Robóticos Autônomos

Andouglas Gonçalves da Silva Júnior

Plataforma de Simulação para Veleiros Robóticos Autônomos

Andouglas Gonçalves da Silva Júnior

Orientador: Rafael Vidal Aroca

Trabalho de conclusão de curso apresentado ao Departamento de Engenharia da Computação e Automação da Universidade Federal do Rio Grande do Norte – UFRN – para a obtenção do título de Bacharel em Engenharia Mecatrônica.

Área de Concentração: Automação e Controle

UFRN - Natal

Novembro de 2013

Agradecimentos

Sumário

1 Introdução **9**

1.1 Motivação 9

1.2 Objetivos 11

2 Revisão Bibliográfica **11**

3 Embasamento Teórico **13**

3.1 Velejo 13

3.1.1 Sobre os Veleiros 13

3.1.2 Como velejar 14

4 Projeto **19**

4.1 Módulo de Atuação 21

4.2 Módulo de Sensoriamento 21

4.2.1 Anemômetro 22

4.2.2 Sensor de Angulação 23

4.3 Módulo de Controle 24

4.3.1 Controle de Velocidade 25

4.3.2 GPS Falso 27

4.3.3 Algoritmo de Navegação 28

4.3.4 Algoritmo de Análise dos Resultados 30

4.4 Módulo de Comunicaçao 30

5 Estado atual do Projeto **31**

5.1 Descrição das atividades realizadas 31

5.2 Resultados Experimentais 33

5.2.1 Trajetória Direta 34

5.2.2 Zig-zag 37

5.3 Resultados Finais 38

5.4 Dificuldades e Limitações 39

6 Conclusões e Trabalhos futuros **39**

7 Referências Bibliográficas **40**

8 Anexos **42**

8.1 Algoritmo do Arduino 42

8.2 Algoritmo de recebimento de dados pelo programa 45
8.3 Algoritmo de envio de dados para o Arduino 45
8.4 Parte do codigo que plota os gráficos na tela. 45

Lista de Figuras

1 Diagrama da patente US 4158263. Figura adaptada de [18]. 12
2 Imagem referente a patente . Figura adaptada de [16] 13
3 Partes principais de um veleiro. Figura adaptada de [11]. 14
4 Vento real e vento aparente. Figura adaptada de [14] 17
5 Manobras de um veleiro com relação a direção do vento. Figura adaptada de [9] 18
6 Diagrama da plataforma completa 20
7 Diagrama da plataforma simplificada 20
8 Ventilador utilizado para simular velocidade e direçao do vento 21
9 Anemômetro utilizado no projeto. 22
10 Potenciômetro utilizado no projeto. Figura adaptada de [13] 24
11 Arduino Uno. Figura adaptada de [2] 25
12 Esquema do circuito para controle dos ventiladores. 25
13 CI 4N35. Figura adaptada de [3]. 26
14 CI MOC3020. Figura adaptada de [4]. 26
15 TRIAC BT 137. Figura adaptada de [7]. 27
16 Vetores de velocidade e direção do vento. 29
17 Vetores de velocidade e direção do vento. 30
18 Circuito montado para o controle dos ventiladores 32
19 Tela de passagem dos parâmetros. 33
20 Tela de gráficos. 33
21 Resultados para situaçao de vela em zero e leme variando. 34
22 Resultados para velocidade do vento 10km/h, sem correnteza. 35
23 Resultados para velocidade do vento 10km/h, com correnteza. 36
24 Resultados para velocidade do vento 15km/h, sem correnteza. 36
25 Resultados para velocidade do vento 10km/h, sem correnteza, em zig-zag, com vela constante. 37
26 Resultados para velocidade do vento 10km/h, sem correnteza, em zig-zag, com latitude constante. 38

Lista de Tabelas

1 Descriçao das partes principais de um veleiro. 15

2 Jargões náuticos 15

3 Manobras de velejo 17

4 Teste de velocidade do ventilador.. 32

Resumo

Neste documento apresenta-se o trabalho de conclusão de curso referente ao projeto de graduação de Andouglas Gonçalves da Silva Júnior, que consiste no projeto e implementação de uma prova de conceito de uma plataforma para simulação de controle de veleiros robóticos autônomos. Neste trabalho foi desenvolvido uma plataforma composta de ventiladores, sensores e microcontrolador que é capaz de simular condições básicas em um ambiente márítimo como: velocidade e direção do vento e correntes marítimas. Além disso, um algoritmo que simula um GPS também foi implementado para aumentar o grau de realismo do sistema como um todo.

Através dessa plataforma é possível simular o deslocamento do veleiro caso ele estivesse em condições reais, de acordo com o controle implementado no mesmo. Desta forma, é possível testar e validar esses algoritmos de navegação para barcos robóticos sem, necessariamente, leva-los para ambientes reais como mares e lagos.

O trabalho final correspondeu ao proposto neste documento e pode ser confirmado através dos resultados obtidos. Experimentos e análises do sistema demosntraram a viabilidade prática de seu uso. Como trabalhos futuros, o sistema proposto pode ser expandido para simular maiores detalhes de ambientes de navegação.

1 Introdução

1.1 Motivação

A utilização de veículos robóticos autônomos começa a ganhar espaço fora do ambiente industrial, onde já são amplamente utilizados. Sua capacidade de locomoção, de extração e envio de dados e sua autonomia, fazem desses equipamentos um grande foco de pesquisas para emprega-los em diversas áreas. Neste trabalho, destaca-se sua utilização no ambiente marítmo.

Pesquisas climatológicas, segurança do litoral, estudo sobre o meio ambiente, qualidade da água e exploração de petróleo são apenas algumas das motivações para utilização dos veículos robóticos autônomos marítimos. Já utilizados em grande empresas exploradoras e que focam em pesquisa, esses veículos se apresentam como uma alternativa para situações adversas e complexas que se pode encontrar em mares ou lagoas e que se torna inviável, ou até impossível, o acesso de seres humanos para extração de dados ou exploração.

Entre as principais vantagens para se utilizar esses veículos, pode-se destacar:

- Facilidade de extrair dados;
- Acesso a lugares complexos e adversos;
- Grande carga de dados transferidos de longas distâncias;
- Segurança elevada, já que não é necessário a atuação do homem diretamente no local explorado.

Porém, também é possível destacar algumas desvantagens da utilização desses veiculos, como:

- Custos altos;
- Necessidade de um estudo elaborado e complexo para construção do robô e do seu controle;
- Foco especial na autônomia da fonte de energia utilizada.

Este último ponto é um fator que se apresenta como um problema na construção desses equipamentos. O gasto excessivo de bateria para propulsão e deslocamento é um ponto que vem gerando grande investimento para se encontrar uma forma de melhorar a autonomia desses veículos.

Neste cenário, um tipo de veículo marítimo se apresenta como uma possível solução para alguns desses problemas: o veleiro. Esse meio de transporte marítmo não necessita de propulsão por motores. Segundo Belcher[17], o uso de barcos com propulsão do vento é uma boa alternativa para as tarefas que exigem missões de longa duração, já que as exigências são menores quando comparados com barcos movidos por motores. Toda a sua navegação é feita através da força do vento agindo nas velas e pelo posicionamento correto do leme. Desta forma, é possível utilizar a maior parte da capacidade da bateria para alimentação

de sensores e sistemas de comunicação e o mínimo possível para locomoção, de forma que se possa obter maior quantidade de dados e tempo de operação autônoma em uma determinada situação.

Um ponto que levou a elaboração desse projeto é a dificuldade, ainda maior, de criar o controle de navegação. Como não consiste apenas no envio de tensão para os motores de propulsão, o projeto apresenta desafios e oportunidades de pesquisa e desenvolvimento. Além de ser necessário encontrar o ângulo certo da vela e do leme, ralacionando-os com velocidade e direção do vento, também é necessário considerar situações como as correntes marítimas, ondas e possíveis obstáculos, o que dificuta ainda mais o desenvolvimento do sistema de controle.

Apesar disso, é possível e exequível a construção dos veículos robóticos autônomos desde que se tenha como validar de forma segura e rápida a evolução do projeto. Isso é necessário pois situações como a necessidade de deslocar esses veleiros para lagos ou piscinas, a falta de controle de situações como velocidade e direção do vento e situações climáticas como chuvas são fatores que deixam a execução do projeto mais complicada e cara.

Esses veleiros robóticos autônomos são um tipo de barco propulsionado pelo vento, assim como os veleiros comuns, mas que podem se deslocar de um ponto a outro de forma autônoma devido a um sistema de controle projetado para definir determinadas angulações da vela e do leme de acordo com a direção do vento e do alvo. Basicamente, dois servo-motores são utilizados para variar essa angulação: um diretamente ligado ao leme e outro ligado a vela por meio de um barbante. Esse barbante é liberado ou retido de acordo com a variação do ângulo do servo. Quando a vela é liberada, o vento atuante faz com que ela varie para um valor que depende de quanto o barbante foi liberado.

Analisando esses fatores é proposta uma plataforma de simulação para veleiros robóticos autônomos que facilite e permita validar os testes de controle e navegação. Em outras palavras, desenvolver todo um sistema inteligente para simular velocidade e direção do vento, correntes marítimas, utilização de GPS e angulação da vela e do leme. Apresentar gráficos que mostrem como o controle está sendo executado e em que direção o barco seguiria se estivesse em condições reais de velejo.

Como o projeto proposto contempla a implementação de muitos detalhes, é apresentada uma arquitetura e ideia do sistema como uma implementação parcial para prova de conceito. É importante saber que a plataforma e o veleiro funcionam de forma independente. A primeira é apenas utilizada para simular o meio ambiente em que a segunda estará condicionada. Desta forma, qualquer barco robótico que seja propelido pelo vento e que utilize GPS como método de posicioinamento pode ser testado no sistema.

No capítulo 2 são apresentados trabalhos relacionados ao projeto. Em seguida, no capítulo 3, um embasamento terórico com relação a navegação de veleiros. No capítulo 4, o que é o projeto. No 5, o estado atual em que o projeto se encontra e, por fim, no capítulo 6 as conclusões e trabalhos futuros.

1.2 Objetivos

Desenvolver uma plataforma de simulação de controle de veleiros robóticos autônomos que seja capaz de analisar a trajetória de um barco robótico, enviado de uma determinada posição inicial para uma posição final que se localize na direção em que o vento simulado aponta. O resultado desta simulação é um gráfico de latitude e longitude que identifica o posicionamento do barco caso ele estivesse em uma situação real de velejo. Além disso, criar o sistema de simulação de GPS, apresentado neste trabalho como falso GPS, que mapeia os dados reais do sistema de posicionamento global para um espaço local de trabalho, podendo assim, relaciona-los e apresentar resultados mais realísticos.

Por fim, deseja-se obter resultados específicos para duas situações encontradas na navegação de veleiros: trajetória retilínea e em zig-zag, podendo as duas ser implementadas na situação em que o vento está na mesma direção do ponto alvo e que são conceituadas de melhor forma no decorrer deste documento.

2 Revisão Bibliográfica

Na literatura atual não são encontrados trabalhos relacionados diretamente a construção de uma plataforma que simule as codições marítimas para navegação de veleiros. Porém, como o foco principal deste trabalho é a análise de controle desses veículos marítimos, são apresentados alguns projetos que já encontram-se em execução ou em desenvolvimento, mostrando-se como se torna importante os estudos nessa área.

O AVALON [21] foi desenvolvido pelo Institudo Federal de Tecnologia de Zurich, na Suíça, no intuito de competir no campeonato de veleiros robóticos autônomos Microtransat [10]. Seu controle é baseado em lógica fuzzy e é um veleiro rápido e preciso. Outro veleiro bastante conhecido e considerado como o primeiro projeto de veleiro robótico autônomo a obter sucesso é o Roboat [23]. Este também participou da competição Microtransat, porém também é utilizado para monitoramento e pesquisa em áreas de interesse. Além desses, o FAST [22] desenvolvido pela Faculdade de Engenhria do Porto, Portugal, também se destaca mundialmente. Seu maior diferencial é a utilização de FPGAs para o sistema de controle digital.

Por fim, em estágio inicial na Universade Federal do Rio Grande do Norte está o N-Boat [15]. Projeto que incentivou o desenvolvimento deste trabalho, está na fase de teste dos controles básicos de um nautimodelo para então partir para a contrução do veleiro em escala maior.

Todos os projetos citados neste tópico e tantos outros que estão em desenvolvimento são possíveis utilizadores da plataforma construída neste trabalho.

Além desses projetos, é possivel encontrar algumas patentes que se assemelham a este trabalho. A patente US 4158263 [18], com o título de *Sailing Simulator*, apresenta um reservatório com um trilho circular deslizante anexado a parede, onde o veleiro é amarrado e pode fazer manobras caso algum vento

natural ou artificial seja fornecido dentro dos limites do reservatório. A figura 1 mostra o desenho deste projeto.

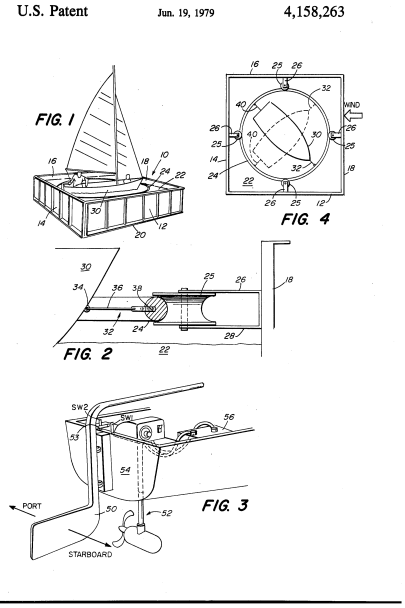

Figura 1: Diagrama da patente US 4158263. Figura adaptada de [18].

Outra patente que apresenta uma estrutura para simulação de veleiros é a US 20110281244 A1[16]. Esta patente propõe a construção de uma plataforma fixa que pode girar ao longo dos três eixos das coordenadas cartesianas, ou seja, varia sua direção roll, pitch e yaw. A figura 2 apresenta a ideia desta patente.

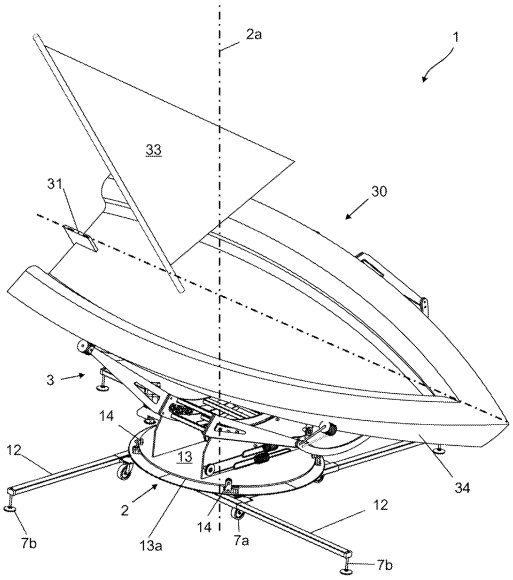

Fig.1

Figura 2: Imagem referente a patente . Figura adaptada de [16]

As duas patentes citadas apresentam apenas uma estrutra mecânica para a simulação de velejo. A plataforma proposta neste trabalho é mais completa e atual, envolvendo dados em tempo real e cálculos matématicos de controle. Além disso, apresenta um foco especial para os veleiros robóticos autônomos, de forma a simula-los de uma forma mais próximo da realidade. Portanto, este projeto pode ser considerado algo inovador e promissor para essa área.

3 Embasamento Teórico

Nesta seção são apresentados alguns conceitos importantes para o entendimento da proposta deste trabalho.

3.1 Velejo

Para a execução do projeto é necessário entender o que são os barcos à vela, como acontece a sua navegação e quais as manobras necessárias para determinadas situações. Esses três pontos são apresentados em seguida.

3.1.1 Sobre os Veleiros

Os veleiros são uma das primeiras e mais revolucionárias invenções da humanidade no campo dos transportes [19]. Bastante utilizado nos tempos antigos, eles foram essenciais no comércio marítimo, viagens

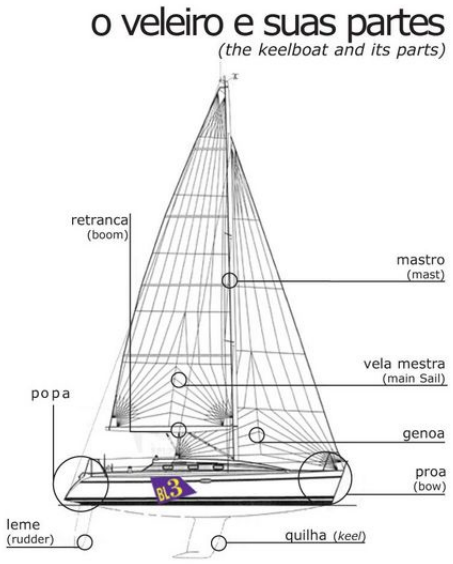

Figura 3: Partes principais de um veleiro. Figura adaptada de [11].

de exploração e intecâmbio cultural. Basicamente, sua propulsão é dada através do vento que aciona as velas e lhes dão movimento. A direção é dada pela angulação do leme que fica em contato direto com a água.

Os veleiros comuns apresentam, basicamente, oito partes essenciais para a sua navegação. Entender cada parte é fundamental para compreender o seu funcionamento e consequentemente como funciona a sua navegação.

A figura 3 apresenta as partes constituintes de um veleiro completo. Em seguida, na tabela 1, é apresentada uma breve descrição das oito partes básicas contidas no veleiro.

Vale salientar que para os veleiros robóticos autônomos a cana do leme não existe, já que o leme gira a partir da rotação do servo-motor acoplado a ele. Além disso, a retranca também é acionada por um servo motor que libera ou puxa uma corda (barbante) e varia a angulação da vela.

3.1.2 Como velejar

Saber como velejar é o que permite desenvolver os controladores dos veleiros robóticos autônomos. A forma de velejar varia de acordo com a posição do vento com relação ao barco. Porém, os velejadores profissionais utilizam uma espécie de algoritmo para cada situação, o que permite a construção dos algoritmos de controle.

Antes de entender como velejar, é necessário entender alguns termos conhecidos como jargões náuticos e bastante utilizados nas manobras dos veleiros reais. A tabela 2 apresenta o que siginifica cada expressão.

Tendo o conhecimento destas expressões é possível se adquirir um melhor entendimento no conceito de navegação, que é de fundamental importância para se compreender como funciona o controle de navegação

Nome	Descrição
Casco	Concha externa do barco que contêm todos os componentes internos do veleiro. Sua forma simétrica dá equilíbrio e reduz o arrasto com a água.
Cana do Leme	Posicionado na parte de trás do veleiro, ou popa, é utilizado para variar o ângulo do leme.
Leme	Componente que fica submerso na água. Como dito anteriormente, é o responsável pelo direcionamento do barco.
Quilha	Localizada na parte de baixo do veleiro, permite que o barco não emborque, atuando como um peso de equilíbrio sob a água.
Retranca	Viga em que se prende o lado horizontal da vela. Esse componente serve como braço de apoio para girar a vela e assim variar a incidência do vento na mesma. Desta forma é possível variar a velocidade do veleiro.
Mastro	Viga em que se prende o lado vertical da vela. Serve para fixar a vela principal ao veleiro.
Vela Principal	Consiste na maior vela e a responsável por receber a maior parte do poder eólico necessário para mover o veleiro.
Bujarrona	Vela secundária, menor que a principal, que permite uma potência adicional à vela principal.

Tabela 1: Descriçao das partes principais de um veleiro.

Jargão	Significado
Estibordo	Corresponde ao lado direito.
Bombordo	Corresponde ao lado esquerdo.
Popa	Parte traseira do barco.
Proa	Parte dianteira do barco.
A ferros	Quando o barco está em direção diretamente oposta ao vento e as velas não se inflam.
A barlavento	Dirigir o barco na direção do vento.
Vento real	Velocidade e direção do vento referenciados a um observador em terra.
Vento aparente	O que se sente quando o barco se move. Combinação do vento real com o criado devido a movimentação do barco.
Ajustar as velas	Posicioná-las para a máxima eficiência.

Tabela 2: Jargões náuticos

de um veleiro robótico autônomo e, consequentemente, os padrões abordados neste trabalho.

O que diferencia a construção deste tipo de controle para outros mais utilizados, como por exemplo os carros robóticos, é que não consiste apenas em enviar uma determinada tensão aos motores e receber de retorno, através de um determinado sensor, uma velocidade ou posicionamento do mesmo. No caso de um veleiro, é necessário fazer o controle de angulação do leme e o de liberação da corda da vela, para que esta alcance uma determinada angulação e receba a ação do vento para aumentar ou diminuir a velocidade e, portanto, faze-lo movimentar-se.

Alguns conceitos básico são importantes para entender a navegação. Esses conceitos são apresentados e explanados abaixo.

Conceito 1 *Resultante*: No deslocamento de uma embarcação há grande influência de forças externas, como por exemplo as correntes marítimas. Estes provocam abatimento ou deriva no rumo da embarcação. Desta forma, não se pode aproar (levar o barco para) diretamente ao objetivo que se quer alcançar. Com isso é preciso escolher uma determinada direção cuja resultante seja uma função da corrente, velocidade do barco e distância a percorrer.

Conceito 2 *Vento real e vento aparente*: Para encontrar a direção da força resultante é necessário levar em consideração aquilo que é vento real e aquilo que é o vento aparente. O vento aparente é um elemento que aparece quando o veleiro está em movimento. Um exemplo simples é quando alguém está a correr em um dia sem vento. A sensação do vento no rosto consiste no vento aparente. Para a navegação é um fator importante para estabelecer a resultante e aproar o veleiro a seguir naquela direção.

O vento aparente nos veleiros autônomos precisa ser levado em consideração, talvez ainda mais do que em um veleiro real. Isso acontece pois o vento aparente, dependendo de onde se encontra instalado o sensor de velocidade do vento, modifica o valor lido, o que dificulta na execução do controle em um ambiente real.

A figura 4 apresenta esses dois primeiros conceitos. Basicamente, o vetor em verde representa o vento induzido obtido a partir da combinação do vento real e do vento aparente. Além disso, o vetor em azul mostra a velocidade e direção do barco de acordo com a resultante.

Conceito 3 *Termos Náuticos*: Através desses conceitos básicos é possível se ter uma ideia de como velejar. Existe manobras estabelecidas para diferentes angulações do vento com relação ao barco e com relação ao objetivo final.

A tabela 3 mostra o nome utilizado para as manobras em termos náuticos. A figura 5 permite uma melhor visualização e entedimento destas manobras.

Analisado esta figura, percebe-se que existe uma situação denominada zona morta, que acontece quando o barco está totalmente contra o vento, não tendo como movimentar-se para frente sem que

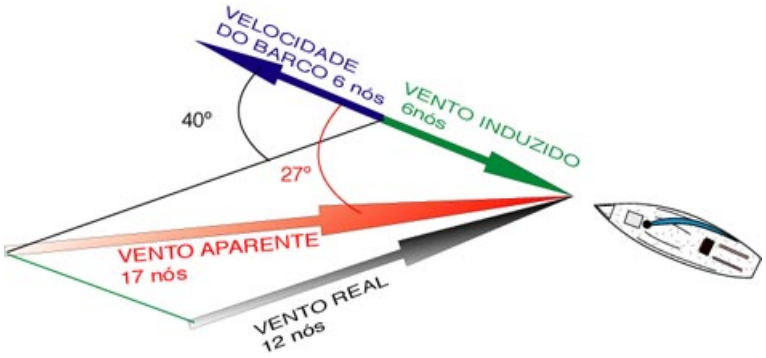

Figura 4: Vento real e vento aparente. Figura adaptada de [14]

Manobra	Descrição
Orçar	Acontece quando a posição do veleiro é perpendicular a direção do vento e o veleiro é direcionado contra o vento.
Arribar	Acontece quando a posição do veleiro é perpendicular a direção do vento e o veleiro é direcionado a favor do vento.

Tabela 3: Manobras de velejo

modifique sua direção. Além disso, existe também uma situação denominada vento em popa que é a base deste trabalho e é explicado um pouco mais adiante.

Para a elaboração deste projeto são apresentadas três situações básicas que podem ser encontradas em uma navegação real em que se deve utilizar uma das manobras citadas anteriomente.

1) Vento na mesma direção do barco

Também conhecido como vento em popa, acontece quando o vento está na mesma direção do barco. Este caso é a mais simples forma de navegação. Basta liberar a vela de forma que fique com uma angulação de 90^{o} com relação ao vento e deixa-lo atuar. Deste modo, é possível alcançar qualquer alvo que esteja em um determinado intervalo de angulação da embarcação com relação ao vento variando minimamente o leme. No caso simples, se o alvo está no mesmo vetor de direção deste vento, o leme não precisa ser acionado.

2) Vento contrário ao barco

Quando se veleja contra o vento é necessário utilizar a manobra chamada de Orça, ou bolina cochada. Nesta manobra se veleja a 45^{o} com relação ao vento. Caso a angulação seja menor, não existirá a condição mínima necessária para se locomover, já que o barco estará aproado ao vento. Portanto, o percurso pode ser feito utilizando técnica de zig-zag ou de bordos múltiplos, em que um momento o barco está a esquerda e em outra a direita do vento.

3) Vento perpendicular ao barco

Este caso é conhecido como través, ou largo, e acontece quando se veleja atravessando o vento, ou seja, o vento chegará ao barco perpendicular a ele, entrando pelo costado do veleiro. As velas são ajustadas

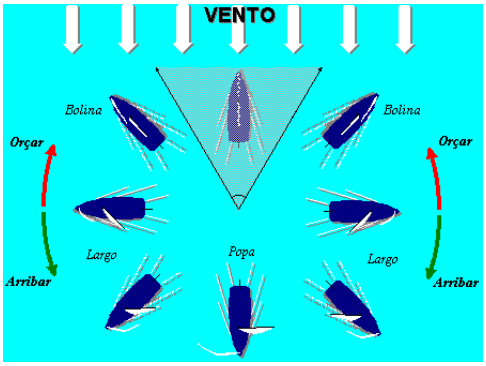

Figura 5: Manobras de um veleiro com relação a direção do vento. Figura adaptada de [9]

para ficarem a 45º do eixo longitudinal do barco. O través é a posição que o veleiro se movimenta com maior velocidade.

Nota-se que em qualquer momento pode-se alcançar qualquer uma dessas posições mostradas anteriormente e, consequentemente, utilizar das manobras relacionadas a essas posições. Por exemplo, mesmo que se esteja velejando na mesma direção do vento, pode-se ter um maior desempenho em velocidade inclinando-se o barco para alcançar a posição través e executar o caminho em zig-zag.

Devido a simplificação imposta neste trabalho, apenas uma das situações acima é simulada na plataforma: o vento na mesma direção do barco em relação ao alvo. Porém, é possível simular a mesma situação para duas posições distintas: vento em polpa e través.

4 Projeto

O projeto consiste na elaboração de uma plataforma de simulação para veleiros robóticos autônomos. Basicamente, o sistema é capaz de analisar o algoritmo de controle de um determinado veleiro robótico e tirar conclusões a partir da sua execução. Desta forma, será possível saber qual o grau de exatidão e precisão deste algoritmo e em quais situações ele é válido. Além disso, o sistema permite realizar todo processo de testes e desenvolvimento de veleiros robóticos em um ambiente de laboratório, evitando gastos de tempo e dinheiro com deslocamento para testes em lagos, rios ou mares. Dessa forma, o ambiente de simulação permite acelerar os testes e desenvolvimento de projetos de barcos robóticos, além de aumentar a segurança dos testes, já que o barco robótico pode ser levado para testes de campo apenas após ter sido testado de diversas formas em laboratório.

Para sua implementação são necessários oito ventiladores dipostos de uma forma que possam simular ventos nas direções dos quatro pontos cardeais (norte, sul, leste e oeste) e quatro pontos colaterais (nordeste, sudeste, noroeste e sudoeste). Além disso, um anemômetro para verificar a velocidade do vento também é utilizado.

Todo esse sistema é interligado a um microcontrolador Atmega328 contido na placa de desenvolvimento Arduino[12] que recebe os valores da posição da vela, do leme e a velocidade do vento medida com o anemômetro e é capaz de atuar nos ventiladores por meio de um circuito que é apresentado no desenvolvimento deste trabalho. Uma das vantagens de se utilizar o Arduino é a facilidade de se empregar novas funcionalidades adaptando uma placa chamada de shield, que extende sua capacidade.

Um computador central é capaz de enviar e receber dados para o Arduino por meio da interface serial. Desta forma, ele recebe os valores obtidos com os algoritmos implementados no microcontrolador e apresenta-os em forma de gráfico para a análise desses dados.

Todo o controle é feito pelo microcontrolador que executa as ações de acionar o ventilador a ser utilizado e sua velocidade através de um algortimo de controle, além de enviar os dados para o computador central.

Por fim, uma tela de supervisório é apresentado no computador central de modo que se análise, em tempo real, as ações que o barco faria em uma situação verídica. Por exemplo, o gráfico do deslocamento da posição inicial à posição final. A figura 6 mostra um esquema de como ficará a estrutura da plataforma.

Nesta figura podemos notar os oito ventiladores classificados de acordo com a direçao do vento que produzem, por isso se apresenta de forma contrária a rosa-dos-ventos. No centro do círculo formado pelos ventiladores estão os sensores envolvidos no projeto. Estes, por sua vez, ligados ao microcontrolador que recebe seus dados, os processa e, tanto atua no acionamento dos ventiladores, como envia as informações para a estação base para ser apresentado nos gráficos.

Porém, como dito anteriormente, foi feita uma simplificação do projeto completo. Portanto, o trabalho atual é desenvolver a plataforma de modo que seja possível verificar a saída do veleiro de um determinado

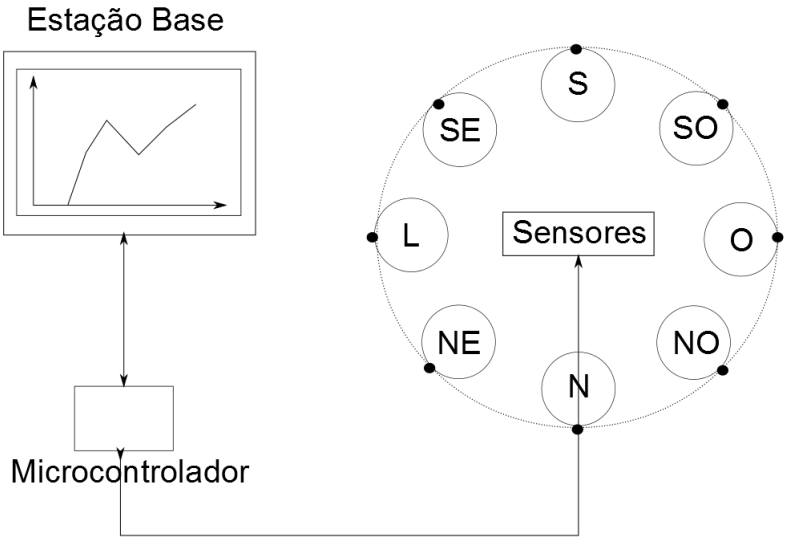

Figura 6: Diagrama da plataforma completa

ponto a outro que esteja no mesmo quadrante e na mesma direção do vento. Com isso, o projeto se resume a três ventiladores posicionados de forma que possa simular a velocidade do vento na direção norte e variações de ângulos em $+45^{o}$(Nordeste) e -45^{o} (Noroeste). A figura 7 mostra a nova estrutura para o projeto simplificado.

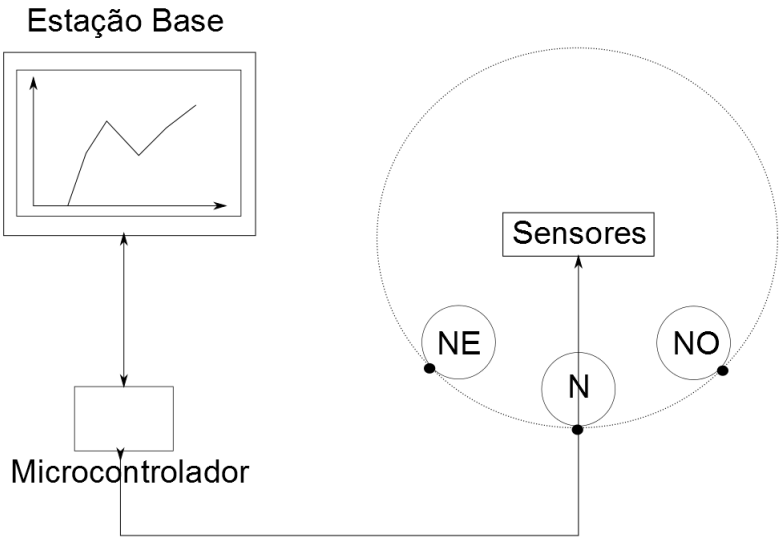

Figura 7: Diagrama da plataforma simplificada

Com essa estrutura é possivel validar a construção da plataforma completa, apenas adicionando novas funcionalidades de controle para outras situações de velejo. Para facilitar a execução do projeto, o mesmo foi divido em quatro módulos distintos: módulo de atuação, módulo de sensoriamento, módulo de controle e módulo de comunicação.

4.1 Módulo de Atuação

O módulo de atuação consiste nos componentes que atuam diretamente no veleiro. No caso simplificado, apenas os ventiladores se apresentam neste grupo.

Os ventiladores utilizados são comerciais de 220V (Figura 8). Através de testes realizados com o anemômetro, constatou-se que sua velocidade varia de 0 à aproximadamente 16km/h. Eles são os responsáveis por simular a velocidade e direção do vento na plataforma. Basicamente, um determinado sinal é enviado para o ventilador que será acionado e então sua velocidade é variada para um valor definido nos parâmetros do projeto. O controle de navegação informa qual ventilador será utilizado em um determinado momento.

Figura 8: Ventilador utilizado para simular velocidade e direçao do vento

Para o controle de velocidade destes ventiladores foi utilizado um circuito elétrico que recebe a tensão da rede e envia uma determinada corrente proporcional ao valor da velocidade desejada. Esse circuito e seu funcionamento é apresentado no módulo de controle devido a sua relação direta com o algoritmo de velocidade do vento.

4.2 Módulo de Sensoriamento

O módulo de sensoriamento corresponde aos sensores envolvidos na construção do projeto. Basicamente, para construir um controle simples de um veleiro é necessário apenas a direção em que o vento se encontra com relação ao norte. Nem sua velocidade é necessária, já que influenciaria apenas na velocidade do barco. Porém, para a simulação na plataforma acontece o contrário. Enquanto a direção do vento não precisa ser verificada, a leitura da sua velocidade é de fundamental importância para que se possa variar a posição do veleiro o mais próximo possível do real. Para o caso simplificado, este módulo consiste em um anemômetro e dois ponteciômetros que funcionam como sensor de angulação da vela e do leme.

4.2.1 Anemômetro

O anemômetro é um sensor que serve para medir a velocidade do vento. Neste trabalho foi utilizado o tipo concha. Neste tipo de anemômetro, o vento faz com que um pólo vertical gire de acordo com a sua velocidade, gerando um valor de tensão correspondente a esse movimento giratório devido a variação da resistência interna.

A sua utilização é de fundamental importância para verificar se a velocidade do vento gerada pelos ventiladores corresponde ao valor recebido como parâmetro inicial. Esta informação é utilizada para calcular o erro no algoritimo de controle e ajustar o valor da entrada para estabilizar e manter a velocidade constante.

O funcionamento do anemômetro é semelhante ao funcionamento de um potenciômetro. Porém, neste caso, a rotação é infinita, permitindo assim a contagem das revoluções em um determinado período de tempo, e, consequentemente, verificar a velocidade do mesmo.

A aquisição dos dados é feita através de um dos 2 pinos que envia o valor da tensão de acordo com a variação da resistência, ou seja, funciona como um divisor de tensão. A figura 9 apresenta o anemômetro usado neste trabalho.

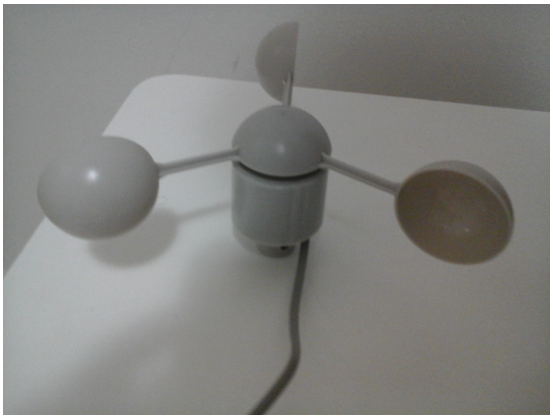

Figura 9: Anemômetro utilizado no projeto.

O algoritmo para o cálculo da velocidade do vento é simples e apresentada na listagem 1. Incialmente, dois inteiros são declarados. Em seguida, uma variável do tipo long é declarada com o valor de 2400 que corresponde a constante para fazer a conversão de milhas - unidade do datasheet - para quilômetro. Então a variável *speed* recebe seu valor adicionado do número de revoluções do anemômetro multiplicado por ele mesmo. Depois esta mesma variável recebe o seu valor dividido por uma constante de tempo de delay, o $MSECS_CALC_VEL_VENTO$. Esse valor é passado para a variável *iSpeed*. Por fim, o resultado final é a variável v, que primeiro recebe o valor de *iSpeed* divido por 10 - valor da casa decimal - e em seguida adiciona esse valor com o resto dessa divisão através da função *mod* para então somar com o valor anterior.

Listagem 1 Algoritmo para aquisição dos dados de velocidade do vento

```
void calcVelVento() {
int v, iSpeed;
long speed = 2400;
speed *= numRevsAnemometro;
speed /= MSECS_CALC_VEL_VENTO;
iSpeed = speed;
v = iSpeed / 10;
velocidadeDoVento = v;
v = iSpeed % 10;
velocidadeDoVento += v;
numRevsAnemometro = 0;
}
void ContaPulso() { pulsos++; }
void countAnemometro() { numRevsAnemometro++; }
```

4.2.2 Sensor de Angulação

Para que a angulação da vela e do leme fossem capturadas com o intuito de utiliza-los no algoritmo de controle foram utilizados dois potenciômetros, um para cada componente. Na verdade, não consiste em um sensor comercial, apenas em uma solução encontrada para cobrir a necessidade atual.

O potenciômetro consiste em um equipamento eletrônico que tem a capacidade de variar sua resistência interna. Porém, diferentemente do anemômetro, sua rotação varia dentro de um intervalo de 0 a 270 graus para a angulação e de 0 a 10kΩ para variação da resistência.

Como no caso do veleiro o leme varia de -45º e 45º e a vela de -90º à 90º, o potenciômetro pode ser empregrado neste projeto. Desta forma, a partir do valor da resistência ajustada é possível relacionar uma tensão de saída, em outras palavras, consiste em um divisor de tensão assim como o primeiro sensor. Consequentemente, por esse ajuste ser feito através de um componente giratório, relaciona-se essa tensão a uma angulação. Essa angulação é o que enviado para o algoritmo de controle do sistema.

A relação entre angulação e variação da resistência para a vela é dada por:

x= ($\frac{270}{1023}$)*y

Onde,

x = valor em graus da angulação desejada;

y = valor da resistência dada pelo potenciômetro;

Esse valor de 1023 foi obtido pois o AtMega tem conversor A/D de 10 bits, ou seja, são 1024 níveis de quantização. Portanto, o intervalo de 0 a 1023 vai se relacionar com a variação da resistência entre 0 e 10kΩ.

No caso do leme, como este varia dentro de um intervalo de 90º, foi feito um ajuste para posicionar o valor de 0º na metade deste intervalo e então poder variar a angulação do leme de -45º e 45º. A relação da angulação e variação da resistência para o leme é dada por:

x= ($\frac{45}{511.5}$)*y

Onde,

x = valor em graus da angulação desejada;

y = valor da resistência dada pelo potenciômetro;

O potenciômetro apresenta 3 pinos: dois responsaveis pela diferença de potencial no componente e um terceiro que coleta a tensão devido a variaçao da resistência. A figura 10 mostra um dos potenciômetros utilizados no projeto.

Figura 10: Potenciômetro utilizado no projeto. Figura adaptada de [13]

4.3 Módulo de Controle

O módulo de controle é o responsável por gerenciar e executar toda a lógica contida na plataforma. Esta lógica consiste em alguns pontos importantes no funcionamento do sistema, como:

- qual ventilador deve ser acionado para uma determinada manobra do veleiro;
- a partir das angulações da vela e do leme, qual o percurso que o veleiro seguirá de acordo com a direção do vento;
- tendo como base a velocidade do veleiro, qual o seu posicionamento instantâneo para ser apresentado nos gráficos de resultados.

Dentro deste módulo estão os quatro algoritimos que formam este projeto: controle de velocidade do vento, GPS falso, algoritimo de navegação e análise dos dados.

Além dos algoritmos, este módulo também apresenta a plataforma Arduino que, através do microcontrolador Atmega328, é responsável por gerenciar e executar todos os algoritmos descritos nesta sessão. A figura 11 mostra o Arduino Uno usado neste trabalho.

Esta placa apresenta 6 pinos para sinais analógicos e 14 para digitais, que é suficiente para a montagem do circuito para os três ventiladores. Além disso, como já citado anteriormente, é possível introduzir shields para determinadas funcionalidades de forma simples, permitindo a adição de novas funcionalidades à plataforma de forma rápida e segura. O algoritmo completo embarcado neste Arduino é apresentado no anexo 1.

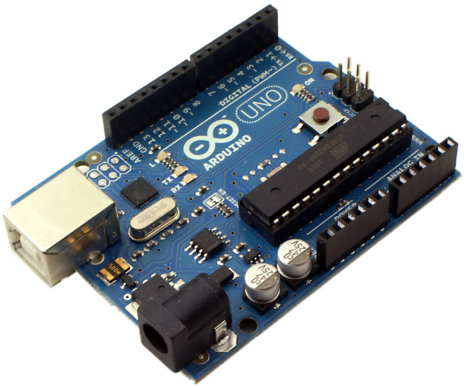

Figura 11: Arduino Uno. Figura adaptada de [2]

4.3.1 Controle de Velocidade

O controle de velocidade dos ventiladores pode ser feito a partir da relação dessa velocidade e a corrente elétrica necessária para aciona-los. O sinal de referência que o controlador utilizará é aquele recebido pelo anemômetro, ou seja, é necessário enviar uma corrente de modo que a velocidade desejada seja igual a obtida pelo sensor.

Para que fosse possível trabalhar com ventiladores comerciais de 220V juntamente com o Arduino, um circuito elétrico foi montado. Esse circuito varia a corrente entregue a carga, nesse caso o ventilador, de acordo com a tensão imposta pelo controlador. O esquema do circuito é mostrado na figura 12.

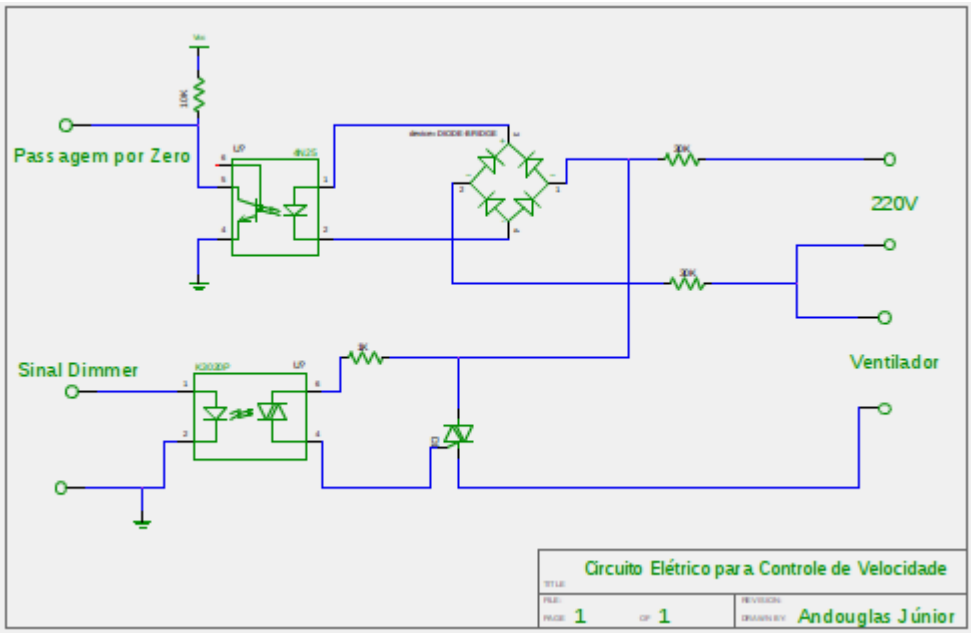

Figura 12: Esquema do circuito para controle dos ventiladores.

Neste ponto é necessário analisar a seguinte situação: além de se ter uma tensão alta, o sinal de corrente da rede elétrica é alternado. Para que seja possível controlar esse sinal entregue a carga é

necessário "cortar" a corrente no ciclo de oscilação. Em outras palavras, a energia enviada a carga é variada quando a onda de corrente desliga por um determinado tempo e depois é religada.

Desta forma é possível entender como foi montado o circuito de controle. Na realidade são dois circuitos diferentes. Um informa ao Arduino cada passagem por zero do sinal senoidal da rede. O outro controla efetivamente, comandado pelo próprio Arduino, permitindo que apenas parte da onda seja entregue a carga.

O primeiro circuito consiste do CI 4N35[5] , dois resistores de 30kΩ e uma ponte retificadora. Este CI é um optoacoplador composto por um fotodiodo e um fototransitor. Sua função é desacoplar eletricamente o circuito que recebe a tensão, daquele que está conectado ao Arduino. Isto é possível pois o fotodiodo, ao conduzir, emite luz que é captada pelo fototransitor, permitindo assim conduzir entre o coletor e o emissor. Desta forma, os dois lados do CI são independentes se analisados eletricamente e acoplados se analisados opticamente. Além disso, o optoacoplador indicará ao Arduino a passagem por zero, indicando assim o início de um novo ciclo. A figura 13 apresenta o CI 4N35.

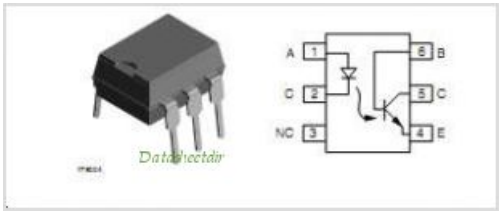

Figura 13: CI 4N35. Figura adaptada de [3].

O segundo circuito é composto por um CI MOC3020[6] (figura 14), um TRIAC bt 137[7] (figura 15) e um resistor de 1kΩ. Esse circuito recebe o sinal de controle do arduino, e o CI, assim como o 4N35, permite a condução para a carga. Desta forma, a carga recebe apenas um valor proporcional a corrente total recebido pelo circuito completo.

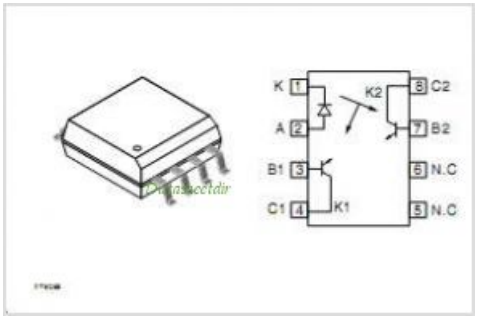

Figura 14: CI MOC3020. Figura adaptada de [4].

Desta forma, utilizando-se destes dois circuitos simples, é possível enviar o sinal desejado para o transistor, e, consequentemente, apenas a parcela desejada para a carga.

No algoritmo de velocidade do vento é necessário avaliar a fase da rede e, a partir do ciclo de oscilação, permitir ou não a passagem de corrente para a carga.

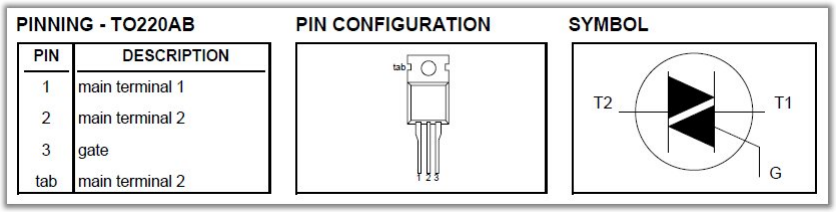

Figura 15: TRIAC BT 137. Figura adaptada de [7].

Portanto, a junção do algoritmo de controle com o circuito implementado permite ajustar a velocidade dos ventiladores e assim simular os ventos que incidem no veleiro em uma situação real.

4.3.2 GPS Falso

No intuito de facilitar a visualização do controle na plataforma de simulação e com o objetivo de aproxima-lo ainda mais do ambiente real foi desenvolvido um algoritmo para simular o posicionamento do barco usando GPS. Para isso, o plano do GPS real foi mapeado para um plano criado virtualmente no projeto. A tela de supervisão imprime o gráfico do GPS real a partir dos dados do GPS virtual.

Basicamente, o sistema marca como parâmetro inicial a posição 0 tanto para latitude, como para logintude. A partir da movimentação simulada do barco, o GPS virtual incrementa utilizando-se de relações trigonométricas a nova posição alcançada. No fim, é feito um mapeamento inverso, retornando assim o local do posicionamento do GPS real.

Para que essa relação fosse possível, o conceito de resultante foi bastante utilizado. Analisando em um âmbito ideal, quanda a angulação da vela está perperdicular ao vento, obtêm-se a maior incidência do vento e consequentemente uma maior velocidade. Da mesma forma, estando a angulação da vela paralela ao vento, não existe força atuando e, portanto, não há velocidade.

Seguindo esse raciocínio, em termos de simulação, é razoável aproximar a relação ângulo da vela e velocidade do vento a partir de relações trigonomêtricas de senos e cossenos.

Caso fatores como correnteza e arrasto sejam desconsiderados, a velocidade do barco consiste justamente da velocidade do vento multiplicada pelo valor do seno do ângulo da vela. Para este trabalho foi utilizado esse princípio no cálculo da velocidade do barco.

Outro fator importante é a relação do valor simulado com o posicionamento real do GPS. Para fazer este mapeamento foi feito alguns cálculos básicos. Considerando o diâmetro total da Terra aproximadamente igual a 12.700 Km[20]. Com esse dado, é possível obter o tamanho da circuferência apenas multiplicando-o com PI. O resultado é de aproximadamente 40.000 Km. Dividindo esse valor por 360, obtêm-se que 1 grau corresponde à 111,12 km. Portanto, a partir da velocidade é possível saber qual foi a movimentação do barco na escala real.

Listagem 2 Algoritmo do GPS falso

```
void gps(){
double Umgrau = 87.2; //Km
velocidadeBarco = velocidadeDoVento/3600;
delay(500);
double v = velocidadeBarco*Umgrau*sin(converteGraus(potenciomentroVela))*(1 − correnteza);
if(potenciometroLeme>400 && potenciometroLeme<700){ val = 551.5; }
latitude = latitude + latitudeCC + v;
if(!(potenciomentroLeme== 551.5))
{
longitude = longitude + longitudeCC +
+ velocidadeBarco*Umgrau*cos(converteGrausLeme(potenciometroLeme+551.5))*v; }
Serial.print(latitude);
Serial.print(",");
Serial.print(longitude);
Serial.print(",");
Serial.print(potenciometroLeme);
Serial.print(",");
Serial.print(potenciometroVela); Serial.print("\n"); }
float converteRadVela(int valor){ return (270.0/1023.0)*(2*3.14/360.0)*valor; }
float converteRadLeme(int valor){
if(valor >= 511.5){
return (45.0/511.5)*(2*3.14/360.0)*valor; }
else{ return (-1)*(45.0/511.5)*(2*3.14/360.0)*valor; }
}
```

4.3.3 Algoritmo de Navegação

A navegação do veleiro na plataforma é o objetivo principal deste projeto. Para que se obtivesse os resultados desejados foi necessário entender a forma de velejo e alguns fatores importantes como a resultante do vento. O algoritmo é básico e foi feito desconsiderando fatores como mudança bruscas de vento, obstáculos e ondas.

A partir da direção do vento é analisado a angulação da vela e do leme para então modificar os ventiladores utilizados e suas velocidades na plataforma. Basicamente, como o sistema apresenta apenas duas situações, ao analisar a angulação do leme e perceber que não há variação, o algoritmo calcula que não é necessário acionar os outros ventiladores, deixando o ventilador acionado a partir do parâmetro inicial e na velocidade desejada. Caso o leme modifique sua angulação, o sistema verifica o valor dessa variação e aciona os ventiladores que direcionam o vento na posição 45º e -45º graus com relação ao norte. A situação que nao se encontra inserida nesses dois padrões ainda não pode ser observada pelo sistema.

Para entender melhor como acontece a navegação na plataforma basta analisar a figura 16 e entender como funciona o cálculo vetorial.

A figura 16 apresenta um diagrama com a direção do vento, o veleiro e a angulação da vela e do leme. O vetor Vv consiste na direção da velocidade da vela caso esta estivesse livre. Porém, como ela está ligada ao veleiro e o leme permanece na posição 0, em uma análise ideal, o veleiro não poderia se deslocar para o lado em que aponta o vetor Vvx (componente X do vetor Vv). Desta forma, o vetor Vvy, componente y do vetor Vv, é o responsável pela parcela da velocidade da resultante do vento que, efetivamente, levará

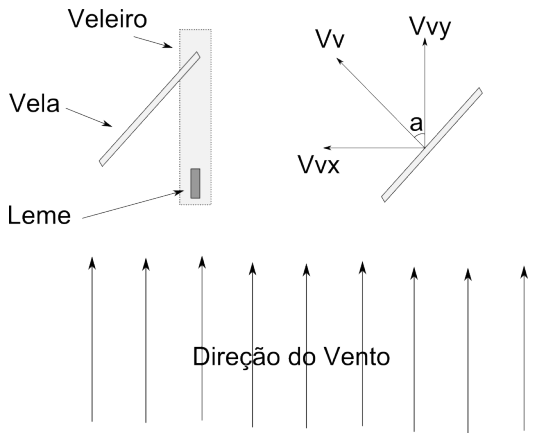

Figura 16: Vetores de velocidade e direção do vento.

a um deslocamento do veleiro. Desconsiderando forças contrárias, essa seria justamente a velocidade do barco para aquela determinada angulação. Em termos matemáticos:

$Vvy = Vv * cos(a)$

Onde, a correponde ao ângulo formado entre a vela e a direção do vento.

Com isso, se a vela está com uma angulação de 90 graus, em um modelo ideal, o veleiro se movimentaria com a maior velocidade possível, já que:

$a = 0^{o}$

$cos(0) = 1$

E portanto:

$Vvy = Vv$

Sabe-se que em um ambiente real isto não acontece devido as forças contrárias que atuam na movimentação do veleiro e, principalmente, devido a aerodinâmica da vela que influência consideravelmente nesse deslocamento. Porém, para termos de simplificação, consiste em uma aproximação razoável.

No caso do leme, sua angulação irá modificar a direção do barco de uma forma linear, ou seja, ao longo do tempo, caso o ângulo do leme seja variado, o barco se posicionará em um mesmo ângulo. Porém, é importante perceber que a mudança de angulação do leme provoca uma mudança de angulação da vela com relação ao vento que precisa ser analisada (Figura 17).

Analisando a figura 17 percebe-se que quando a angulação do leme varia em um ângulo b, o ângulo a também varia, caso a vela permaneça constante. Um ângulo menor consiste em uma velocidade maior se for levado em conta as considerações apresentadas anteriormente. Portanto, caso seja necessário permanecer em uma velocidade constante, basta que a vela seja reposicionada a partir do posicionamento do leme. Esta situação consiste em um controle básico em que este trabalho é baseado.

Para calcular o posicionamento do barco quando o leme é variado, desconsiderando uma variação da vela, basta analisar o ângulo be relaciona-lo com o vetor Vvy da seguinte forma:

$Vl = Vvy * sen(b)$

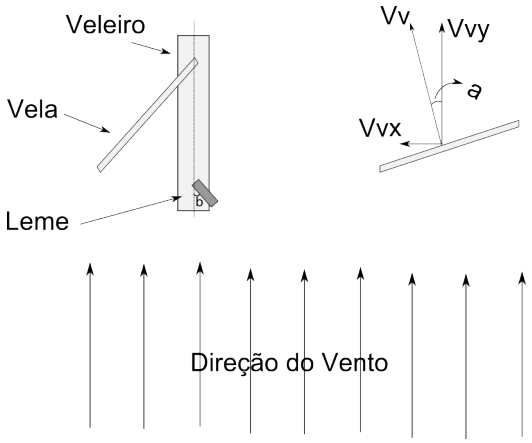

Figura 17: Vetores de velocidade e direção do vento.

Onde Vl é a componente da velocidade na direção X com relação a velocidade do barco e b é a angulação do leme com relação ao barco.

4.3.4 Algoritmo de Análise dos Resultados

O objetivo principal do projeto é apresentar a validade do controle implementado em um determinado veleiro robótico. Para tal, é importante analisar o deslocamento do veleiro através dos eixos longitudinal e latitudinal, a angulação da vela e do leme para se ter uma ideia de que, de acordo com a manobra que deveria ser realizada, aqueles ângulos condizem com o esperado.

O gráfico gerado na tela de supervisório consiste nos valores obtidos a partir do algoritmo do GPS falso em conjunto com a lógica para a relação com a velocidade do barco. Caso não tenha sido especificado nenhuma força contrária ao movimento do veleiro, como por exemplo alguma corrente, a força resultante de velocidade consiste da relação da angulação da vela e da posição do vento.

Os dados que são apresentados em supervisório são a latitude, longitude, angulação da vela e angulação do leme em instantes de tempo pré-definidos. Como esses dados são apresentados é visto mais adiante.

4.4 Módulo de Comunicaçao

Neste projeto são utilizados um Arduino e um computador para a estação base. Para que estes dois compontentes possam interagir entre si, existe o módulo de comunicação. Esse módulo é o responsávael por enviar as informações dos parâmetros inicias para o Arduino e responder o valor do posicionamento do GPS e da vela e do leme para ser mostrado no gráfico.

Para tal, foi implementado um algortimo para fazer a comunicação com a porta USB do arduino através da API RXTX[1] em Java[8]. O processo é síncrono e acontece na seguinte ordem:

1. Início da simulação: os valores dos parâmetros inicias são enviados para o Arduino.

2. O Arduino executa o algoritmo de velocidade e posicionamento, e reenvia para o supervisório os

dados desejados.

3. O supervisório plota os valores recebidos nos gráficos e apresenta os resultados em tela.

Com o intuito de facilitar essa comunicação foi criada uma espécie de protocolo que é entendida tanto pelo computador da estação base, como no microcontrolador. No pacote de dados enviados estão os campos de velocidade inicial e número do ventilador a ser acionado de acordo com a direção do vento que este deve gerar. Na resposta, estão os valores obtidos do algoritmo de posicionamento do GPS (latitude e longitude) e a angulação da vela e do leme para que possam ser inserido no gráfico.

5 Estado atual do Projeto

Nesta sessão é apresentada a situação atual em que o projeto se encontra, tudo aquilo que já foi implementado e os resultados que já foram obtidos a partir de alguns testes básicos.

Vale salientar que a necessidade de construção desta plataforma corresponde as dificuldades encontradas pela equipe do projeto N-Boat da própria Universidade Federal do Rio Grande do Norte em realizar testes para corrigir determinados problemas em seus algoritmos de controle.

Para a execução deste trabalho foi de fundamental importância e participação dos componentes da equipe do N-Boat, a saber: Rafael Vidal Aroca (orientador deste trabalho); Luiz Marcos Garcia Gonçalves; André Paulo Dantas de Araújo e Einstein Santos mestrandos em Engenharia da Computação e Elétrica com tema também relacionado a veleiros robóticos; aluno de graduação em Computação Davi Henrique e aluno de graduação em Ciências e Tecnologia Marcus Azevedo.

5.1 Descrição das atividades realizadas

Os objetivos foram alcançados exceto no que se diz respeito ao controle de velocidade do vento. Pela dificuldade de controlar o sistema por causa da corrente elétrica na rede ser alternada, em alguns momentos, a velocidade do ventilador fica muito baixa, não permitindo analisar os dados de velocidade do anemômetro. Porém, todos os outros pontos descritos como objetivos do trabalho estão em pleno funcionamento. Isso inclui os algoritmos implementados, a lógica e a parte eletrônica. Além disso, é importante destacar que o algoritmo necessário para controle apresentado inicialmente funciona caso seja possível fazer o mapeamento preciso da fase da tensão da rede.

Para a exemplificar a funcionalidade desse algoritmo foi realizado testes que relacionam o controle à velocidade do ventilador. A variável *speed* utilizada no algoritmo pode variar de 0 a 240, obtendo assim a menor e maior velocidade, respectivamente. A tabela 4 apresenta esses testes com valores intermediários para a variável *speed*.

Para que o algoritmo de controle da velocidade do vento realmente funcionasse, mesmo que de uma forma mais simples do que o que seria implementada, foi desenvolvido um controle ON/OFF que controla a

Valor	Velocidade aproximada (km/h)
0	0
60	3,6
120	7,8
180	11,2
240	16

Tabela 4: Teste de velocidade do ventilador..

velocidade do ventilador para a indicada nos parâmetros iniciais. O circuito montado a partir do diagrama da figura 12 é apresentado na figura 18.

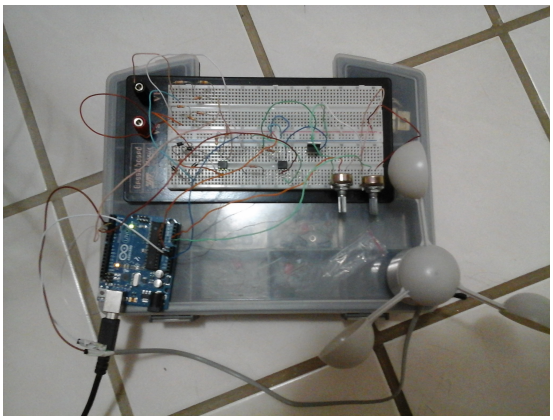

Figura 18: Circuito montado para o controle dos ventiladores

Outro ponto importante é que pela dificuldade de se obter ventiladores iguais, os testes com os três ventiladores ficaram comprometidos, precisando ajustar a velocidade do vento em software. Isso se deu pois existe ventiladores que geram uma velocidade maior com um alcance maior que outros. Desta forma, é necessário que todos os ventiladores da plataforma sejam iguais para que os resultados sejam validados.

No quesito de comunicação, quatro dados importantes para a validação do controle são transmitidos e entregues via porta serial a estação base. Esses dados são de latitude, longitude, angulação do leme e da vela.

Para a simulação do GPS, o algoritmo do falso GPS recebe as informações de posição real e a partir do mapeamento para o espaço de trabalho do projeto, retorna um valor para a tela de supervisório. Porém, neste ponto, ainda não é possível verificar o quão preciso se encontra já que os testes realizados no sistema ainda não foram realizados em um ambiente real.

Como ainda não foi feito nenhum algoritmo de controle para o veleiro todos os testes estão sendo feito de forma manual, variando a angulação da vela e do leme apenas modificando o potenciômetro. Basicamente, é como se estivesse sendo ultizado um rádio para controlar o veleiro.

Por fim, os dados recebidos pelo computador da estação base são plotados nos gráficos e apresentados em tela. Esses gráficos mostram o percurso que o veleiro percorreria caso aquelas determinadas especificações inicias fossem seguidas, ou seja, apresenta justamente o objetivo final para este trabalho.

Um programa em Java foi desenolvido para melhorar a visualização dos dados recebidos e o envio dos parâmetros iniciais para a plataforma. A tela inicial desse programa é mostrado na figura 19.

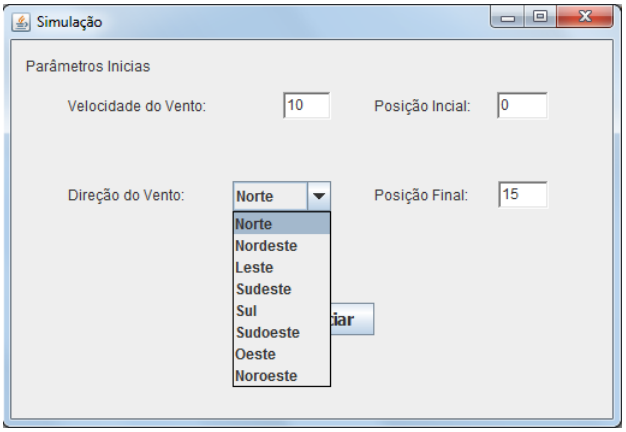

Figura 19: Tela de passagem dos parâmetros.

Nesta tela é possível informar ao sistema quatro parâmetros básicos para controlar a plataforma: velocidade e direção do vento, posição inicial e posição final. Quando o botão iniciar é pressionado, o programa já recebe as informações do arduino e as plota nos gráficos como apresentado na figura 20.

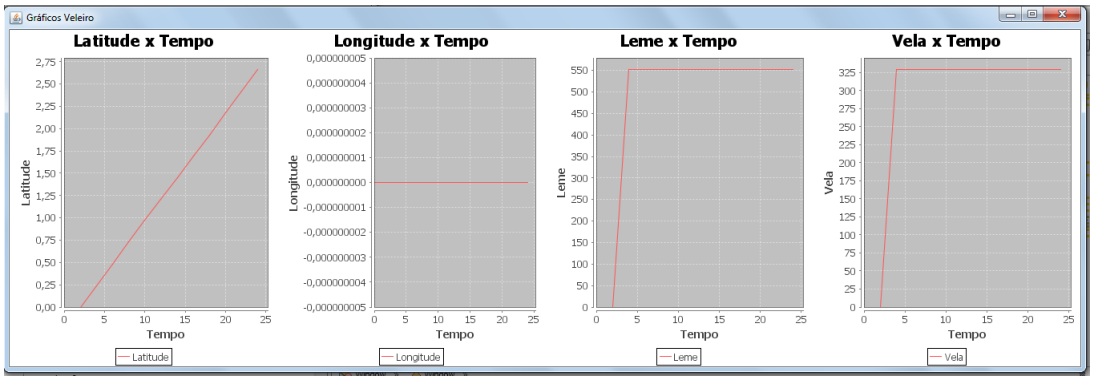

Figura 20: Tela de gráficos.

Esta tela consiste na vizualização final da plataforma. No decorrer do tempo, os gráficos vão variando de acordo com as modificações que ocorrem no sistema. Nesta tela são mostrados os dados de latitude, longitude, angulação do leme e angulação da vela.

5.2 Resultados Experimentais

Inicialmente, para a obtenção dos resultados, foi feito um teste básico para a análise do algoritmo de controle de navegação. Como já citado anteriormente, o leme não tem interferência nenhuma na velocidade de deslocamento do barco. A sua função é apenas no direcionamento. Para se analisar a validade básica do algoritmo, o primeiro teste foi feito deixando a posição da vela em zero. Desta forma, o vento está perpendicular a vela e, teoricamente, não há movimentação do veleiro.

A velocidade do vento colocada como parâmetro inicial foi de 10km/h, a posição inicial com latitude e longitude nula e o vento direcionado ao norte. Além disso, é importante salientar que os testes foram feitos movendo diretamente os potenciometros como se estivesse sendo utilizado um rádio.

A latitude é dada em decimal, como se o barco estivesse iniciando da posição zero e a partir daí variasse até um determinado ponto.

A figura 21 mostra os resultados para esta situação.

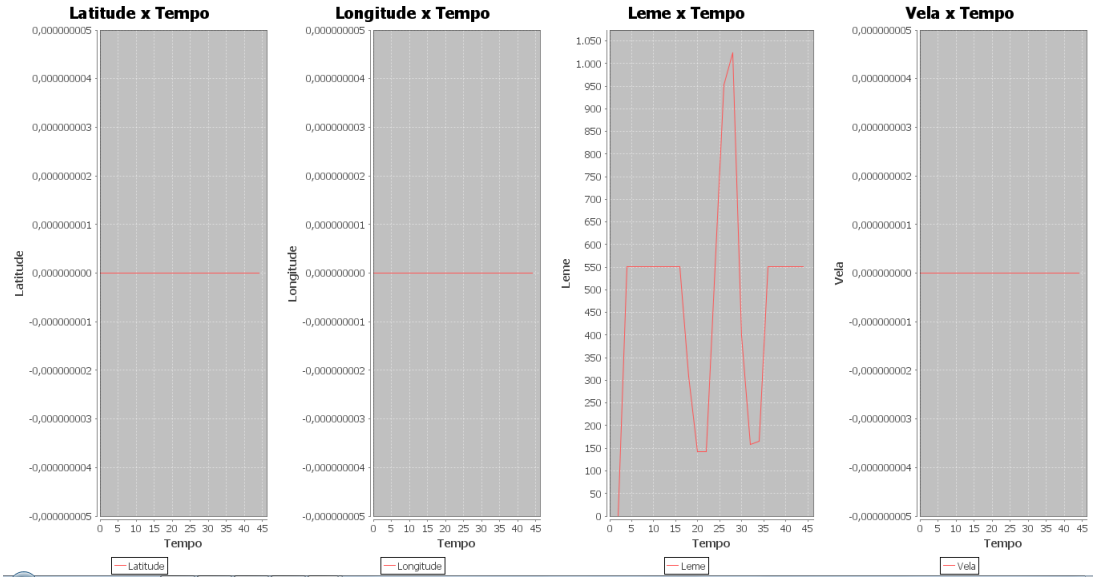

Figura 21: Resultados para situaçao de vela em zero e leme variando.

Percebe-se que angulação do leme foi variada dentro de um determinado intervalo. Porém, como a angulação da vela permaneceu em zero, não houve nenhuma movimentação do barco. Esse dado pode ser constatado a partir dos gráficos de latitude e longitude, ambos com relação ao tempo, que não tiveram variação.

Os próximos testes correspondem ao objetivo deste trabalho. Consiste na movimentação do barco para um ponto que sua direção coincide com a direção do vento. Como já exposto anteriormente, essa situação pode ser executada de duas formas básicas: diretamente e em zig-zag.

5.2.1 Trajetória Direta

Na trajetória direta o barco posiciona a vela a 90º e deixa que o vento atue fazendo com que o barco siga em direção reta, já que o leme permanece na posição zero.

Para esta situação foi considerado a velocidade do vento de 10km/h, direcionado ao norte e sem forças contrárias (correnteza ou arrasto). A figura 22 apresenta os gráficos obtidos para esta situação.

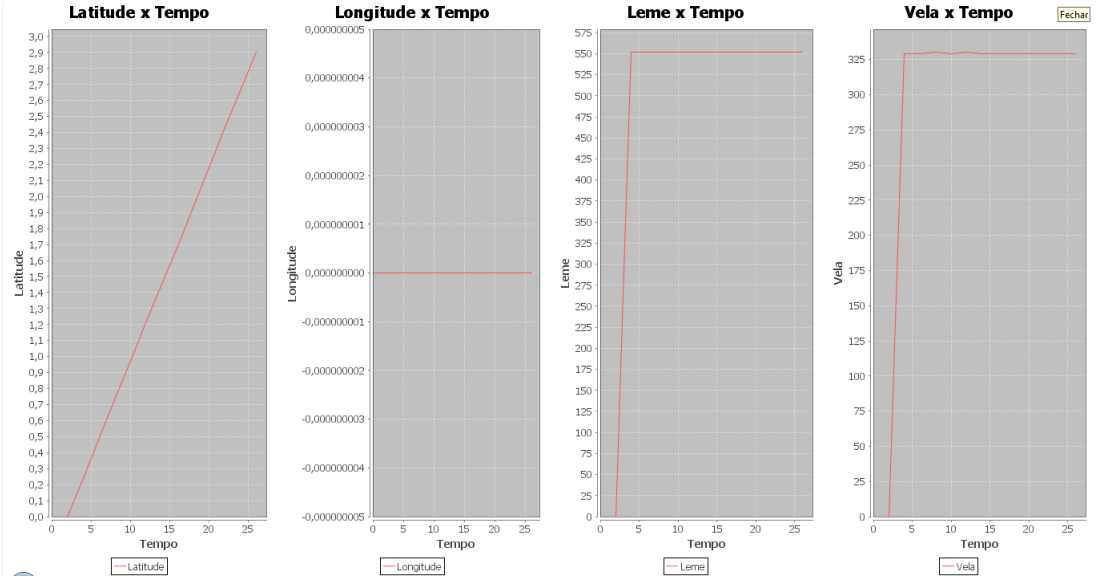

Figura 22: Resultados para velocidade do vento 10km/h, sem correnteza.

Analisando os gráficos é possível perceber que a vela foi levada para uma posição próxima à 90º (correspondente a aproximadamente 341 na escala do potenciômetro). Neste instante, a latitude começou a variar e, até que a vela alcançasse o seu valor final, sua velocidade foi aumentando gradativamente. O leme permaneceu em zero graus fazendo com que a longitude não modificasse o seu valor. Desta forma, houve um deslocamento retilíneo apenas na direção latitudinal.

Vale salientar que esta é uma situação ótima, desconsiderando correnteza e o arrasto. Além disso, é como se o barco se direcionasse para um ponto na mesma posição longitudinal.

Para essa mesma situação foi feito outro teste considerando uma pequena força contrária de aproximadamente 10% do valor da velocidade instantânea do barco, que pode ser entendida como uma correnteza ou arrasto. A figura 23 mostra os resultados obtidos.

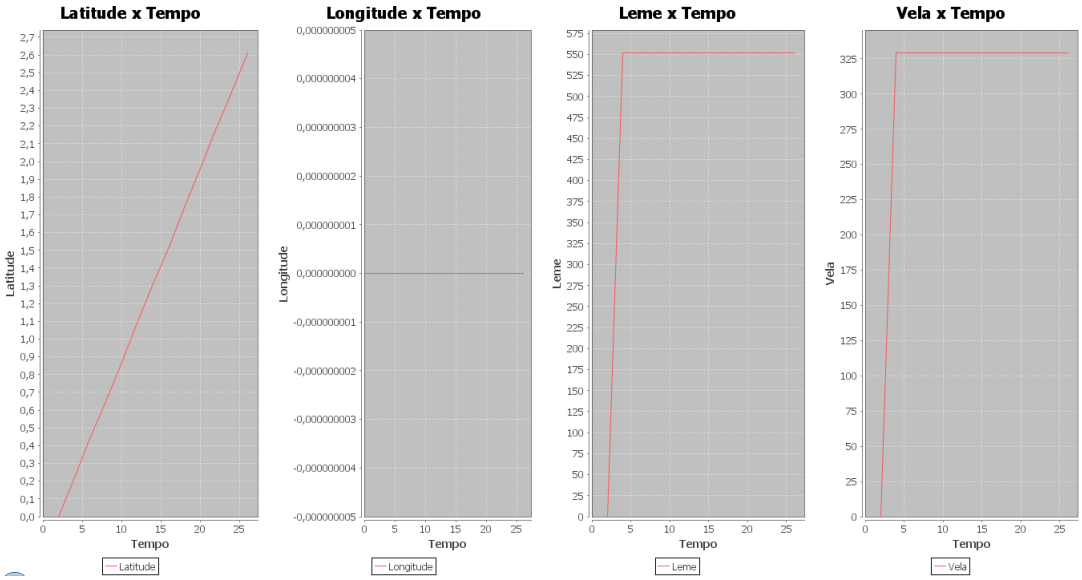

Figura 23: Resultados para velocidade do vento 10km/h, com correnteza.

Neste caso, a latitude também aumentou já que a situação era a mesma do primeiro caso, porém sua variação foi mais lenta devido a correnteza adicionada ao sistema. Nota-se que, por exemplo, o ponto 1.4 da latitude foi alcançado no instante de tempo 15, enquanto na primeira situação este mesmo ponto foi alcançado aproximadamente no instante 13.

Por fim, foi analisado a situação em que o barco estava a uma velocidade de 15km/h (5 km a mais que nas duas situações anteriores), sem correnteza. O resultado é apresentado na figura 24.

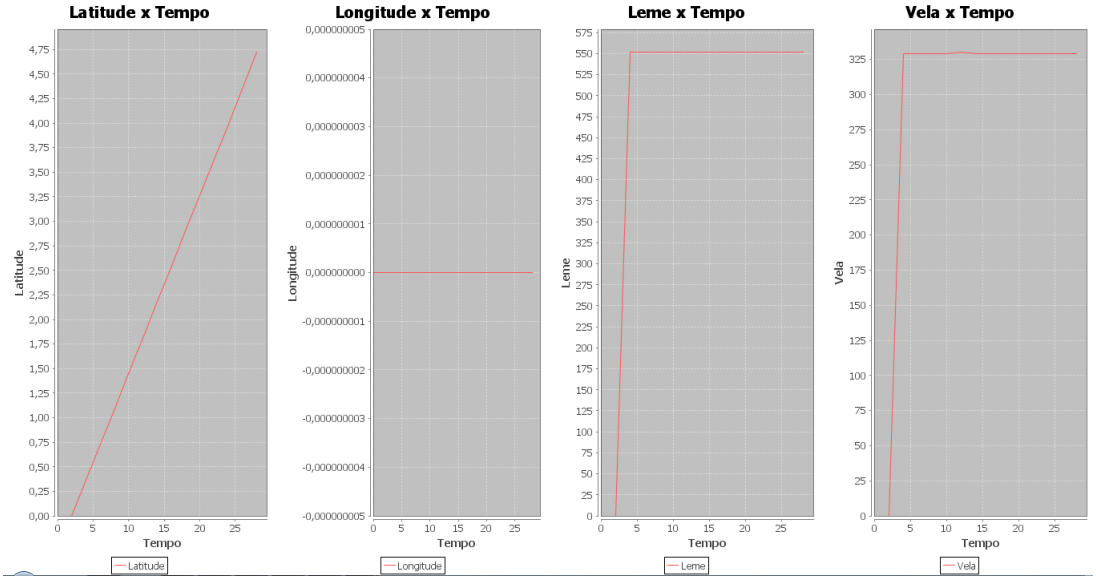

Figura 24: Resultados para velocidade do vento 15km/h, sem correnteza.

Se comparado com a situação com o vento a 10 km/h e sem correnteza percebe uma maior velocidade para alcançar uma determinada latitude devido a velocidade do vento. No mesmo tempo, 25 segundos, o veleiro alcançou uma latitude de aproximadamente 2.6 para a primeira situação e aproxiamadamente

4.75 para a ultima situação.

5.2.2 Zig-zag

A segunda situação é a que o barco movimenta-se ao ponto final em zig-zag. Em situações reais, esta forma apresenta um melhor desempenho devido a aerodinâmica da vela. Portanto, em competições que a velocidade é um ponto considerável esta situação é bastante utilizada.

Para a obtenção destes resultados, a vela foi mantida com um valor constante de aproximadamente 90º e o leme variava a angulação. A figura 25 apresenta os resultados para esta situação. Por dificuldades de obtenção de ventiladores iguais, a direção do vento foi variada em software. Porém, como a angulação da vela não muda, como já mostrado no item referente ao algoritmo de navegação, a partir da mudança de direção do veleiro a angulação da vela com relação a direção do vento modifica, variando assim sua velocidade e consequentemente seu deslocamento, como pode ser visto no gráfico da latitude. Em uma situação de velejo real, seria necesário modificar a angulação da vela caso desejasse que a velocidade permanecesse constante. Esta situação é mostrada logo em seguida.

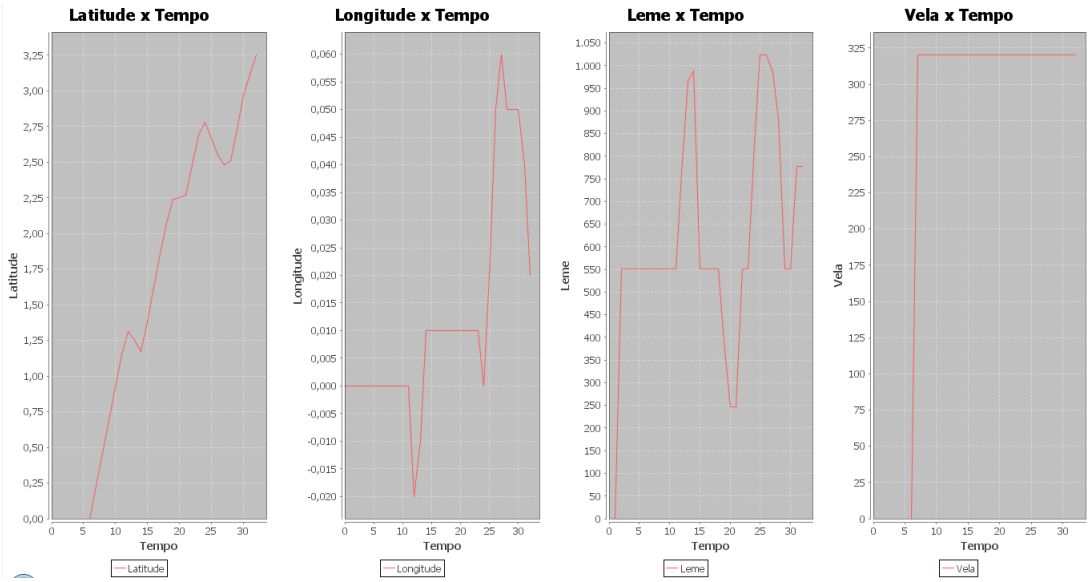

Figura 25: Resultados para velocidade do vento 10km/h, sem correnteza, em zig-zag, com vela constante.

A figura 26 mostra a situação em que se deseja manter o deslocamento do veleiro constante. Para isso, é necessário modificar a angulação da vela para que sempre permaneça com a mesma angulação em relação ao vento.

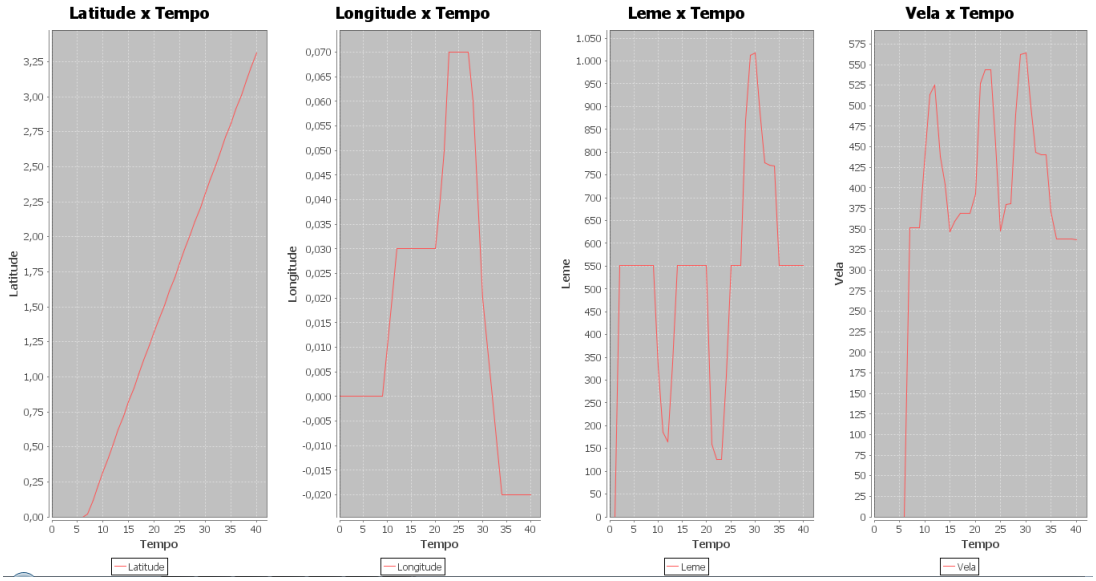

Figura 26: Resultados para velocidade do vento 10km/h, sem correnteza, em zig-zag, com latitude constante.

O gráfico referente a vela mostra a variação de sua angulação relacionada com o posicionamento do leme. Sempre que há uma variação na direção do veleiro, a vela é reajustada para permanecer com a mesma angulação com relação ao vento. Por exemplo, no intervalo entre aproximadamente 8 e 13 a angulação do leme foi variada para próximo de 12º. Esse movimento faz com que o barco gire e mude a angulação da vela com relação ao vento. Por isso, neste mesmo instante, a vela modificou sua angulação para um valor maior do que ela se encontrava e permaneceu com seu valor constante em relação ao vento. Isso pode ser visto a partir do gráfico de latitude que permaneceu constante.

Outros testes não foram realizados para esta situações pela dificuldade em alcançar a mesma variação para o leme de forma manual. Porém, com esses dois testes é possível confirmar a lógica do algoritmo de navegação e valida-lo para as duas situações propostas neste trabalho.

5.3 Resultados Finais

Analisando os resultados numéricos e os objetivos deste trabalho é possível afirmar que este projeto obteve um êxito no seu desenvolvimento e execução. O foco em retornar um possível trajeto para o barco foi alcançado, validando assim um determinado controle que venha a ser utilizado no barco.

Os problemas relacionados ao controle e número de ventiladores não invalidam a veracidade de execução da plataforma completa. Alguns ajustes no circuito eletrônico e a obtenção de mais ventiladores iguais permitem o fechamento total da plataforma e suas funcionalidade.

5.4 Dificuldades e Limitações

A grande dificuldade encontrada foi o controle de velocidade dos ventiladores. Por ser corrente alternada (AC), é necessário um estudo mais aprofundado do que se diz respeito a análise da fase, ciclo e variações da rede. Deseja-se que este problema seja resolvido na execução de trabalhos futuros.

Além disso, a aquisição dos ventiladores para testes também se apresentou como um grande obstáculo. Caso fosse comprado, os valores ficariam alto pela quantidade e qualidade dos ventiladores necessários. Este foi um dos motivos pelo qual o projeto foi simplificado.

Na questão das limitações algumas considerações precisaram ser feitas para que se podesse obter resultados para este trabalho. A não implementação de fatores como o arrasto e correnteza para a situação em zig-zag deixam a plataforma de simulação com uma característica ideal, ainda um pouco distante das reais requeridas para este projeto. Além disso, não considerar a aerodinâmica da vela é um ponto que influência bastante na velocidade final e, consequentemente, modifica os valores finais dos resultados.

6 Conclusões e Trabalhos futuros

A utilização de uma plataforma para simulação de veleiros autônomos facilita o desenvolvimento dos algoritmos de controle, que muitas vezes, apresentam uma implementação complexa e de testes difíceis.

Desta forma, não é necessário deslocar o veleiro para os ambientes de testes, nem se preocupar, inicialmente, com situações adversas relacionados a situação do tempo e do clima. Para a validação inicial do controle, basta que o barco consiga finalizar o percurso desejado, a partir das especificações impostas ao sistema.

Além disso, o desenvolvimento deste trabalho permitiu aplicar os conhecimentos adquiridos durante o curso de uma forma prática e satisfatória. A implementação dos algoritmos, a montagem do circuito e a necessidade de deixar o sistema autônomo, correspondem aos princípios básicos da mecatrônica e, portanto, aquilo que se deseja alcançar através deste curso.

Com relação aos trabalhos futuros, deseja-se dar continuidade a este trabalho no programa de pós graduação desta universidade. Para finalizar este projeto e alcançar o seu desenvolvimento completo é importante que:

- todos os ventiladores estejam disponíveis;
- se analise a aerodinâmica da vela;
- outras manobras sejam implementadas para validar outros tipos de controle;
- condições ambientais mais realísticas como: correntes marítimas, arrasto, ondas e mareação sejam adicionadas ao projeto.

Estes pontos serão os próximos passos para a construção desta plataforma.

7 Referências Bibliográficas

Referências

[1] Api rxtx. On-Line. Disponível em: http://rxtx.qbang.org/wiki/index.php/Main Page. Acesso em 05/11/2013.

[2] Arduino uno. On-Line. Disponível em: http://www.liquidware.com/system/0000/3648/Arduino Uno Angle.jpg. Acesso em 28/10/2013.

[3] Circuito integrado 4n35. On-Line. Disponível em: http://www.datasheetdir.com/4N35 Optocouplers. Acesso em 05/11/2013.

[4] Circuito integrado moc 3020. On-Line. Disponvíel em: http://www.datasheetdir.com/IL388DAA Optocouplers. Acesso em 05/11/2013.

[5] Datasheet ci 4n35. On-Line. Disponível em: http://measure.feld.cvut.cz/groups/edu/osv/4n35.pdf. Acesso em 05/11/2013.

[6] Datasheet ci moc 3020. On-Line. Disponível em: http://pdf.datasheetcatalog.net/datasheet/ motorola/MOC3021.pdf. Acesso em 05/11/2013.

[7] Datasheet triac bt137. On-Line. Disponível em: http://www.nxp.com/documents/data sheet/BT137 SERIES.pdf. Acesso em 05/11/2013.

[8] Java. On-Line. Disponível em: http://www.java.com/pt BR/. Acesso em 05/11/2013.

[9] Manobras de um veleiro com relação a direção do vento. On-Line. Disponível em: http://www.hobbys.com.br/centraldeveleiros/curso 20velejar arquivos/image012.gif. Acesso em 28/10/2013.

[10] The microtransat challenge. On-Line. Disponível em: http://www.microtransat.org/. Acesso em 30/10/2013.

[11] Partes do veleiro. On-Line. Disponível em: http://www.nauticurso.com.br/partes-veleiros.html. Acesso em 20/10/2013.

[12] Plataforma arduino. On-Line. Disponível em: http://arduino.cc/. Acesso em 30/10/2013.

[13] Potenciômetro utilizado no projeto. On-Line. Disponível em: http://www.huinfinito.com.br/344-438-thickbox/potenciometro-linear-10k-cursor-metal-16mm.jpg. Acesso em 29/10/2013.

[14] Vento real e vento aparente. On-Line. Disponível em: http://www.nautica.com.br/colunas/colunistas/sergiocaetano/050902/2 ventoaparente.jpg. Acesso em 28/10/2013.

[15] Marcus V. A. Silva Rafael V. Aroca Luiz M. G. Gonçalves Andouglas G. S. Junior, André P. D. Araújo. N-boat: An autonomous robotic sailboat. *LARS*, page 6, 2013.

[16] Giovanni Ausenda. Simulation apparatus for sailing boat, 2011.

[17] Bill. Belcher. *Wind-vane Self-Steering*. International Marine, 1982.

[18] Jr. Clyde R. Weihe. Sailing simulator, 1977.

[19] Cristen Conger. Como funcionam os veleiros. On-Line. Disponível em: http://viagem.hsw.uol.com.br/barco-a-vela2.htm. Acesso em 30/10/2013.

[20] Paulo Araújo Duarte. Estrutura da terra. On-Line. Disponível em: http://www.cfh.ufsc.br/ pduarte/dadosterra.html. Acesso em 28/10/2013.

[21] CÉDRIC PRADALIER ROLAND Y.SIEGWART HENDRIK ERCKENS, GION-ANDRI BUSSER. Avalon - navigation strategy and trajectory following controller for an autonomous sailing vessel. *IEEE Robotics & Automation Magazine*, page 10, 2010.

[22] Nuno A. Cruz José C. Alves. Fast - an autonomous sailing platform for oceanographic missions. *IEEE*, page 7, 2008.

[23] Karim Jafarmadar Roland Stelzer. The robotic sailing boat asv roboat as a maritime research platform. *International HISWA Symposium on Yacht Design and Yacht Construction*, page 13, 2012.

8 Anexos

8.1 Algoritmo do Arduino

```
1  /*
2  * Algoritmo para o controle da plataforma de
3  * simulação para veleiros robóticos autônomos
4  * Author: Andouglas Gonçalves da Silva Júnior
5  */
6  #define loadR 4
7  #define uint unsigned int
8  #define ulong unsigned long
9  #define PINO_ANEMOMETRO 3
10 #define MSECS_CALC_VEL_VENTO 1000
11 #define MSECS_CALC_DIR_VENTO 5000
12  volatile unsigned int pulsos = 0;
13  double velocidadeDoVento = 10;
14  double velocidadeCC = 5;
15  double velocidadeBarco = 0;
16  double latitude = 0;
17  double longitude=0;
18  volatile int numRevsAnemometro = 0;
19  ulong nextCalcVelocidade;
20  ulong nextCalcDir;
21  ulong time;
22  volatile int power = 100;
23  int potPinoUm = 1;
24  int potPinZero = 0;
25  int ledPin = 13;
26  int valLeme = 0;
27  int valVela = 0;
28  double latitudeCC=0;
29  double longitudeCC=0;
30  double latitudeFF = 20;
31  double longitudeFF = 0;
32  float correnteza = 0;
33
34  void setup() {
35          Serial.begin(9600);
36          pinMode(loadR, OUTPUT);
37          attachInterrupt(0, zero_crosss_int, RISING);
38          pinMode(PINO_ANEMOMETRO, INPUT);
39          digitalWrite(PINO_ANEMOMETRO, HIGH);
40          attachInterrupt(1, countAnemometro, FALLING);
41          nextCalcVelocidade = millis() + MSECS_CALC_VEL_VENTO;
42          nextCalcDir = millis() + MSECS_CALC_DIR_VENTO;
43          digitalWrite(2, LOW);
44          attachInterrupt(0, zero_crosss_int, CHANGE);
45  }
46
47  void loop() {
48          time = millis();
```

```
49          if (time >= nextCalcVelocidade) {
50                  calcVelVento();
51          nextCalcVelocidade = time + MSECS_CALC_VEL_VENTO;
52          }
53          controleSimples();
54          if(latitude < latitudeFF){
55                  gps();
56          }
57          valLeme = analogRead(potPinoUm);
58          valVela = analogRead(potPinoZero);
59          delay(1000);
60  }
61
62  //Controle ON/OFF utilizado.
63  void controleSimples(){
64          long erro = velocidadeDoVento - velocidadeCC;
65          Serial.print("Erro: ");
66          Serial.print(erro);
67          Serial.print("\n");
68          int k = 15;
69          if(erro > 0){
70                  power = 0;
71          }else{
72                  power = 100;
73          }
74  }
75
76  //Algoritmo de controle para o projeto final.
77  void zero_crosss_int() {
78          // Cálculo do ângulo de disparo: 60Hz-> 8.33ms (1/2 ciclo)
79          // (8333us - 8.33us) / 256 = 32 (aprox) int powertime = (32*(256-power));
80          // Mantém o circuito desligado por powertime microssegundos
81          delayMicroseconds(powertime);
82          // Envia sinal ao TRIAC para que ele passe a conduzir
83          digitalWrite(loadR, HIGH);
84          // Espera alguns microssegundos para que o TRIAC perceba o pulso
85          delayMicroseconds(8.33);
86          // Desliga o pulso
87          digitalWrite(loadR, LOW);
88  }
89
90  //=========================================================
91  //ANEMOMETRO - Calculando a velocidade e mostrando
92  // Fator de conversao indicado no datasheet:
93  // 1 rev/sec = 1.492 mph = 2,4 km/h
94  // Agradecimento: Marcus Azevedo que me ajudou na construção deste algoritmo
95  //=========================================================
96  void calcVelVento() {
97          int v, iSpeed;
98          // Produzira 10 milhas por hora ou 24 km/h
99          long speed = 2400;
100         speed *= numRevsAnemometro;
```

```
101         speed /= MSECS_CALC_VEL_VENTO;
102         iSpeed = speed;
103         v = iSpeed / 10;
104         velocidadeDoVento = v;
105         v = iSpeed % 10;
106         velocidadeDoVento = velocidadeDoVento + v;
107         numRevsAnemometro = 0; // Resetando a contagem
108 }
109 // Rotina que é chamada a cada meio ciclo, incrementando a variável pulsos.
110 void ContaPulso() {
111         pulsos++;
112 }
113 void countAnemometro() {
114         numRevsAnemometro++;
115 }
116
117 //=========================================================
118 // GPS - Retornado GPS
119 //=========================================================
120 void gps(){
121         double Umgrau = 87.2; //Km
122         velocidadeBarco = velocidadeDoVento/3600;
123         delay(500);
124         double v = velocidadeBarco*Umgrau*sin(converteGraus(val2))*(1 - correnteza);
125         if(valLeme>400 && valLeme<700){
126                 valLeme = 551.5;
127         }
128         latitude = latitude + latitudeCC + v;
129         if(val != 551.5){
130                 longitude = longitude + longitudeCC +
131                 + velocidadeBarco*Umgrau*cos(converteGrausLeme(valLeme+551.5))*v;
132         }
133         Serial.print(latitude);
134         Serial.print(",");
135         Serial.print(longitude);
136         Serial.print(",");
137         Serial.print(valLeme);
138         Serial.print(",");
139         Serial.print(valVela);
140         Serial.print("\n");
141 }
142
143 float converteGraus(int valor){
144         return (270.0/1023.0)*(2*3.14/360.0)*valor;
145 }
146 float converteGrausLeme(int valor){
147         if(valor >= 511.5){
148                 return (45.0/511.5)*(2*3.14/360.0)*valor;
149         }else{
150         return (-1)*(45.0/511.5)*(2*3.14/360.0)*valor;
151         }
152 }
```

8.2 Algoritmo de recebimento de dados pelo programa

```
1  /*
2  * Método responsável por receber os dados do Arduino
3  */
4  public synchronized void serialEvent(SerialPortEvent oEvent)
5  {
6          if (oEvent.getEventType() == SerialPortEvent.DATA_AVAILABLE) {
7                  try {
8                          String inputLine = input.readLine();
9                          System.out.println(inputLine);
10                         String[] resultado = inputLine.split(",");
11                         i++;
12                         serieLatitude.add(i, Float.parseFloat(resultado[0]));
13                         serieLongitude.add(i, Float.parseFloat(resultado[1]));
14                         serieLeme.add(i, Float.parseFloat(resultado[2]));
15                         serieVela.add(i, Float.parseFloat(resultado[3]));
16                 } catch (Exception e) {
17                         System.err.println(e.toString());
18                 }
19         }
20 }
```

8.3 Algoritmo de envio de dados para o Arduino

```
1  /*
2  * Métod responsável por enviar os dados para Arduino
3  */
4  public synchronized void writeData(String data)
5  {
6          //data é igual a (velocidadeDoVento, direcaoDoVento);
7          System.out.println(data);
8          try {
9                  output.write(data.getBytes());
10         } catch (Exception e) {
11                 System.out.println("A porta não pode ser acessada!");
12         }
13 }
```

8.4 Parte do codigo que plota os gráficos na tela.

```
1          coleccionLatitude.addSeries(serieLatitude);
2          coleccionLongitude.addSeries(serieLongitude);
3          coleccionLeme.addSeries(serieLeme);
4          coleccionVela.addSeries(serieVela);
5          coleccionLatLong.addSeries(serieLatLong);
6
7          graficoLatitude = ChartFactory.createXYLineChart("Latitude x Tempo", "Latitude",
8                  "Tempo", coleccionLatitude, PlotOrientation.VERTICAL, true, true, false);
9          graficoLongitude = ChartFactory.createXYLineChart("Longitude x Tempo", "Longitude",
10                 "Tempo", coleccionLongitude, PlotOrientation.VERTICAL, true, true, false);
11         graficoLeme = ChartFactory.createXYLineChart("Leme x Tempo", "Leme",
12                 "Tempo", coleccionLeme, PlotOrientation.VERTICAL, true, true, false);
```

```
13          graficoVela = ChartFactory.createXYLineChart("Vela x Tempo", "Vela",
14                  "Tempo", coleccionVela, PlotOrientation.VERTICAL, true, true, false);
15  ...
```