

João Victor Batista Gordo

**Implementação de uma plataforma de teste de
estratégia de controle de VANTs utilizando robô
terrestre autônomo de alta velocidade**

São Carlos, Brasil

4 de abril de 2017

João Victor Batista Gordo

**Implementação de uma plataforma de teste de estratégia de
controle de VANTs utilizando robô terrestre autônomo de alta
velocidade**

Universidade de São Paulo

Escola de Engenharia de São Carlos

Departamento de Engenharia Elétrica e de Computação

Orientador: Eduardo do Valle Simões

São Carlos, Brasil

4 de abril de 2017

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

Agradecimentos

Eu gostaria de agradecer à comunidade que vive o paradigma revolucionário de disseminação livre e gratuita de ideias e trabalhos. Este projeto é um fruto tímido de uma árvore regada pelos esforços de cada uma destas pessoas que, de gota em gota, clareiam as águas turvadas pelo egoísmo e nutrem a germinação de verdadeiros pomares que estão por vir. Quem sabe um dia o único obstáculo ao aprendizado e construção do conhecimento seja a força de vontade...

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (...) As palavras-chave devem figurar logo abaixo do resumo, antecidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

Palavras-chaves: latex. abntex. editoração de texto.

Abstract

This is the english abstract.

Key-words: latex. abntex. text editoration.

Lista de ilustrações

Figura 1 – Módulo de Rádio Frequência baseado no <i>transceiver</i> da Nordic nRF24L01+ .	31
Figura 2 – Diagrama da Classe RF24	47
Figura 3 – Estrutura Analítica do Projeto	48
Figura 4 – Diagrama Elétrico	49
Figura 5 – Fluxograma da Rotina de Desvio de Obstáculos	51
Figura 6 – Três tipos de modulação PWM: <i>trailing edge</i> , <i>leading edge</i> e <i>both edges</i> , de cima pra baixo, respectivamente.	52

Lista de tabelas

Tabela 1	–	50
----------	-------------	----

Lista de abreviaturas e siglas

CC	Corrente Contínua
PWM	<i>Pulse Width Modulation</i>
ESC	<i>Electronic Speed Controller</i>
IDE	<i>Integrated Development Environment</i>
ISM	<i>Industrial, Scientific and Medical</i>
UHF	<i>Ultra High Frequency</i>
SPI	<i>Serial Peripheral Interface</i>
USB	<i>Universal Serial Bus</i>
CRC	<i>Cyclic Redundancy Check</i>
GFSK	<i>Gaussian Frequency Shift Keying</i>
BLDC	<i>Brushless Direct Current</i>
FIFO	<i>First In First Out</i>
VANT	<i>Veículo Aéreo Não Tripulado</i>
UBEC	<i>Ultimate Battery Eliminator Circuit</i>
LiPo	Lítio Polímero
MIFA	<i>Meandered Inverted-F Antenna</i>
ICSP	<i>In-Circuit Serial Programming</i>

Lista de símbolos

α	coeficiente de atenuação do meio.
τ	tempo de meia volta da onda propagante, i.e. <i>time of flight</i> , do sensor.
Δ	distância medida nos sensores do tipo <i>time of flight</i> .
ν	velocidade de propagação no meio da onda ou partícula nos sensores do tipo <i>time of flight</i> .

Sumário

1	INTRODUÇÃO	21
2	EMBASAMENTO TEÓRICO	23
2.1	Sistemas Reativos	23
2.1.1	Paradigma Reativo como Robótica Bioinspirada	23
2.1.2	Características	24
2.2	Arquitetura MOSA	24
3	MATERIAIS	27
3.1	Motor <i>Brushless</i>	27
3.2	ESC	28
3.3	Sensor Ultrassônico	28
3.3.1	Princípio de Funcionamento	28
3.3.2	Limitações	29
3.3.2.1	Variação na velocidade de propagação da onda acústica	29
3.3.2.2	Direcionalidade	29
3.3.2.3	Resposta no Ambiente Alvo	29
3.3.2.4	<i>Foreshortening</i>	30
3.3.2.5	Reflexão especular e <i>Crosstalk</i>	30
3.3.2.6	Tempo de Resposta	30
3.4	Módulo de rádio frequência	30
3.5	Arduino	31
3.6	Bateria	32
3.6.1	<i>C rate</i>	32
4	MÉTODO	33
4.1	Estratégia <i>bottom-up</i>	33
4.2	Arquitetura Reativa	33
4.3	Subsistema de Locomoção	33
4.4	Subsistema de Percepção	34
4.5	Subsistema de Comunicação	34
4.6	Subsistema de Navegação	35
4.7	Integração dos Subsistemas	36
5	RESULTADOS	37
5.1	USS	37
6	CONCLUSÃO	39
	REFERÊNCIAS	41

	APÊNDICES	45
	APÊNDICE A – APÊNDICES	47
A.1	Diagrama da Classe RF24	47
A.2	Estrutura Analítica do Projeto	48
A.3	Esquemático do Robô	49
A.4	Tabela	49
A.5	Fluxograma da Rotina de Desvio de Obstáculos	49
A.6	Sistemas de Tempo Real	49
A.7	CRC	50
A.8	MIFA	52
A.9	efeito hall	52
A.10	campo girante	52
A.11	<i>schmitt trigger</i>	52
A.12	PWM	52
A.13	GFSK	53
A.14	banda ISM	53
A.15	SPI	53
	ANEXO A – ANEXOS	55

1 Introdução

2 Embasamento Teórico

2.1 Sistemas Reativos

“Representações explícitas e modelos atrapalham. No fim das contas, a melhor representação do mundo é ele mesmo.”

Brooks, R.A [1] - tradução livre -

2.1.1 Paradigma Reativo como Robótica Bioinspirada

De acordo com Rodney Brooks [1], para o desenvolvimento da inteligência no seu sentido mais estrito e genuíno, são condições suficientes que o indivíduo, que denominaremos agente, tenha as seguintes faculdades: mobilidade dentro de um ambiente dinâmico no qual esteja inserido, percepção do que se passa nas suas adjacências e, por fim, manutenção da própria sobrevivência. Em suma, habilidades como o raciocinar, comunicar-se, gerar conhecimento nada mais são do que comportamentos complexos, consequências simples do fato de existirmos e termos o poder de reagir dentro do meio em que vivemos.

Para aprofundarmos a discussão e esclarecermos como se daria esse processo de aprimoramento dos agentes, é preciso fornecer uma definição mais rigorosa do termo “comportamento”. Numa tradução livre: “comportamentos são mapeamentos diretos de informações sensoriais recebidas em padrões de ações motoras, desempenhadas para se cumprir uma tarefa. Matematicamente, seria uma função transferência que transforma dados dos sensores em comandos para os atuadores” [2].

Nos animais, a transformação de percepção em ação está subordinada à existência de estímulos específicos, de natureza interna ou externa ao agente, que podem ser entendidos como sinais de controle que permitem ou inibem determinados comportamentos [2]. A título de ilustração: ao avistar uma presa - informação sensorial - o predador somente a ataca - comportamento - caso esteja com fome - estímulo interno; ou quando afastamos a mão - comportamento - ao tocarmos uma panela quente - a informação sensorial seria a temperatura da panela enquanto o estímulo externo é o fato de que ela excede uma dada temperatura.

Com base em estudos da etologia, o comportamentos dos animais podem ser inatos ou aprendidos, e a sua inteligência pode ser decomposta verticalmente em camadas de comportamentos, cada qual acessa os sensores e atuadores do agente de maneira independente das demais [2]. Isto é, o indivíduo inicia sua existência com um conjunto comportamentos inatos de autopreservação mas, ao longo da sua vida, outros novos vão surgindo, podendo: refinar comportamentos pré-existentes, negá-los (completamente) ou agregar-se a eles sem produzir conflitos, i.e. trabalhando paralelamente com os que lhe são ancestrais. Desta forma, os dois

primeiros casos podem ser entendidos como uma reutilização de camadas inferiores da inteligência, enquanto o último consiste na adição de mais uma camada.

2.1.2 Características

Por estar embasado em ideias da etologia discutidas na seção anterior, o paradigma reativo simplesmente desconsidera a etapa de planejamento existente na tríade 'percepção, planejamento, ação', que sumariza o ciclo de tarefas realizadas por um sistema sob o paradigma hierárquico [2, 3]. Em suma, os comportamentos se dão de acordo com o que o agente percebe que está acontecendo no seu entorno, não são feitas modelagens ou representações do ambiente externo, apenas medições locais e orientadas a comportamentos.

Em decorrência da exclusão da etapa de planejamento, robôs desenvolvidos sob o paradigma reativo costumam ser simples e apresentarem respostas rápidas [3]. Com boas práticas de projeto é possível construir uma robô com: alta coesão, pois comportamentos podem ter acesso direto aos sensores de que necessitam para tomar suas decisões, o que possibilita um alto grau de independência em relação a operações e dados externos entre diferentes módulos ou subsistemas do robô; e baixo acoplamento, pois comportamentos são independentes entre si e, portanto, há pouca ou nenhuma dependência de ligações e interfaces externas a um dado módulo [2].

2.2 Arquitetura MOSA

Arquitetura que propõe dividir o sistema aéreo de navegação autônoma em dois módulos: aeronave e MOSA [4].

O primeiro constitui a porção crítica do sistema embarcado, i.e. segmento cuja falha pode resultar em ao menos um dos seguintes desastres: morte ou lesão de pessoas; destruição ou danos a propriedades, patrimônios ou equipamentos; danos ambientais [5]. Veículos Aéreos Não Tripulados, VANTs, apresentam a tolerância de um erro grave a cada 10^5 ou 10^9 horas de voo [6], o que as caracteriza como sistemas computacionais de tempo real do tipo *hard*. Maiores esclarecimentos acerca destes jargões podem ser encontrados no apêndice.

O segundo corresponde à parte não crítica à segurança, encarregada do controle da navegação e, por conseguinte, da determinação da maior parte dos parâmetros de voo. É caracterizado como um conjunto de sensores inteligentes capazes de cumprir uma missão específica, ou seja, existe uma relação biunívoca entre missão e MOSA, dado que ele consiste no melhor arranjo de sensores para o cenário em questão. Neste contexto, a aeronave é vista unicamente como o meio de transporte dos sensores, enquanto que o módulo MOSA constituiria o 'cérebro' da plataforma, responsável pelo cumprimento da missão e por guiar a aeronave até a sua realização.

No entanto, como a aeronave é o elemento responsável pela garantia da segurança, cabe a ela acatar ou não os comandos do MOSA. E pode, inclusive, optar por readaptar a missão em tempo de voo para se ajustar ao cenário, o que inclui a seleção dos sensores que melhor se encaixam na dada conjuntura.

Isso se dá através de uma matriz de reconfiguração dinamicamente adaptável denominada *Knowledge Based Framework*, seu papel é comparável à expertise de um piloto. Ou seja, um

elemento inteligente capaz de escolher o melhor serviço a ser executado com base em regras e critérios de seleção como resposta em tempo real, segurança, performance.

3 Materiais

- 5 sensores ultrassônicos de distância HC-SR04
- 2 motores *brushless outrunner* Turnigy D2836/9 950KV
- 2 ESCs Hobby King com UBEC de 5.5V/4A: um de 35A e outro de 40A
- 2 módulos de rádio frequência baseados no *transceiver* Nordic nRF24L01+
- bateria LiPo 30C de 2800 mAh
- 2 Arduino Pro Mini
- 1 Conversor/Adaptador USB-Serial PL2303
- 1 Carregador balanceador de bateria IMAX B6-AC

3.1 Motor *Brushless*

São motores síncronos¹ de corrente contínua cuja comutação é feita eletronicamente e não mecanicamente por meio de escovas como nos motores CC comuns, por isso denominados *brushless*. Possui aplicações nas indústrias de automóveis, aeroespacial, médica, de equipamentos de automação industrial e instrumentação. Os motores BLDC apresentam algumas vantagens em relação aos de corrente contínua com escovas e de indução no que concerne a: resposta dinâmica, ruídos de operação, durabilidade (i.e. vida útil), assim como razão do torque pelas dimensões do motor [7].

O rotor consiste de um ímã permanente, já os pólos do estator são formados por enrolamentos, que precisam ser energizados na sequência correta para que um campo magnético girante seja criado. Nas máquinas CC isto é feito mecanicamente através das escovas mas, no caso do BLDC, é preciso que a posição do rotor em relação ao estator seja conhecida para que seja possível fazer o acionamento correto das bobinas. Existem dois meios de se obter esta informação: através de sensores de efeito hall, método empregado neste trabalho, ou processamento da força contra eletromotriz das bobinas do estator.

Sensores de efeito Hall são transdutores analógicos que relacionam a intensidade do campo magnético externo transversalmente disposto a ele em termos de tensão elétrica. Quando associado a um circuito comparador *schmitt trigger*, comporta-se como um sensor digital que aponta quando a intensidade do campo magnético atinge um valor de limiar pré-determinado. Ao dispor sensores deste tipo ao longo do estator, torna-se possível uma estimativa da posição do rotor ao ser feito um estudo comparativo da resposta de cada sensor, cruzando esta informação com a posição que cada um destes se encontra em relação ao estator [8].

Há a possibilidade de fazer a comutação sem empregar qualquer tipo de sensor, logo, trata-se de um método mais barato. Nesse caso, a estimativa da posição do rotor se dá através do

¹ Motores Síncronos: o campo magnético girante do rotor e do estator têm a mesma frequência.

processamento das forças contra-eletromotriz de cada um dos enrolamentos do estator. No entanto, algumas limitações surgem: o motor deve operar acima de uma dada rotação, caso contrário o método não funciona; mudanças bruscas de carga não podem ocorrer; há discontinuidades na resposta do motor quando operando em velocidades acima da taxa de comutação ideal [8].

3.2 ESC

Controlador responsável por processar as informações oriundas dos sensores de efeito Hall do motor BLDC e providenciar o acionamento correto dos enrolamentos do estator para que a velocidade angular se dê de acordo com o sinal de controle é enviado a este dispositivo. No caso dos ESCs utilizados no presente trabalho, este sinal de controle é feito utilizando-se modulação por largura de pulso, i.e. PWM. A frequência de operação varia de acordo com o modelo do controlador e para o caso deste projeto é de 400Hz.

3.3 Sensor Ultrassônico

3.3.1 Princípio de Funcionamento

Utiliza o método *time of flight*, que consiste na medição do intervalo de tempo, igualmente denominado *time of flight*, que uma onda ou partícula leva para percorrer uma determinada distância em um dado meio. Pode ser utilizado para medir: distância, velocidade [9], propriedades do meio de propagação ou da partícula propagante [10, 11].

Para medidores de proximidade, como é o caso de sonares e lasers, um transdutor emissor faz a conversão do sinal elétrico, denominado *trigger*, em um pulso de ondas (acústicas para o caso do sonar e eletromagnéticas para o laser), dando início à medição de tempo. Quando esta onda propagante encontra um objeto que a reflita de volta ao sensor e a intensidade deste sinal recebido, denominado *echo*, está acima de um determinado valor de limiar, o transdutor receptor envia um sinal elétrico que interrompe a contagem de tempo, obtendo-se assim a medida do *time of flight*, τ . Com isso, supondo que a velocidade de propagação, ν , da onda em questão no meio seja conhecida, de acordo com [12], pode-se calcular a distância, Δ , entre o sensor e o objeto que reflete o pulso de ondas pela equação 3.1:

$$\Delta = \frac{\nu \times \tau}{2} \quad (3.1)$$

Quanto ao sensor ultrassônico especificamente, temos que as ondas sonoras utilizadas estão usualmente situadas entre 40kHz e 180kHz, sendo emitidas no formato de pacotes compostos por uma série de pulsos; no caso do sonar utilizado neste trabalho, 8 pulsos de 40kHz. Por se tratarem de ondas mecânicas, é importante que a tensão de limiar, do inglês *threshold*, decresça ao longo do ciclo de leitura da seguinte forma [12]: durante o período denominado *blanking time*[12] ou *dead time*[2] na literatura, o qual engloba o intervalo de emissão das ondas sonoras até o momento em que o diafragma para de oscilar (o que pode constituir alguns milissegundos após a cessação do sinal de *trigger*), a tensão de limiar é muito alta no intuito de eliminar leituras inválidas decorrentes de interferência entre emissor e receptor; em seguida, a tensão de *threshold* se reduz a um valor que permita detecção de obstáculos e vai sendo continuamente decrementada

com o passar do tempo. Isso se dá pelo fato de que a intensidade do sinal acústico, i.e. potência por ângulo sólido, sofre atenuações atmosféricas que variam com a distância percorrida, conforme a equação 3.2 [13], que leva em consideração somente efeitos da divergência esférica e absorção molecular.

$$I = \frac{I_0 e^{-2\alpha R}}{4\pi R^2} \quad (3.2)$$

Em que: α é o coeficiente de atenuação do meio, associado às absorções moleculares, o qual varia em função da frequência da onda emitida assim como de propriedades do meio, e.g. humidade e poeira contida no ar. Para ondas de 40kHz: $0,197 \frac{dB}{m} < \alpha < 0,295 \frac{dB}{m}$.

3.3.2 Limitações

3.3.2.1 Variação na velocidade de propagação da onda acústica

Como citado anteriormente, a medição da distância pressupõe que a velocidade de propagação da onda no meio é conhecida. No entanto, mudanças na temperatura e umidade do fluido em que a onda se propaga podem causar erros de medida não desprezíveis [13].

3.3.2.2 Direcionalidade

O emissor da radiação acústica ultrassônica apresenta um padrão de radiação[14, 15] composto por lobos laterais[14, 15] que não são levados em conta, pois a maioria dos sistemas supõem toda radiação recebida como oriunda do lobo central[14, 15], usualmente modelado como um cone de aproximadamente 30° que varre até 5 metros [2]. De acordo com [16], para o dispositivo utilizado nesse trabalho o ângulo de abertura do feixe é de 15° e o alcance, 4 metros.

Além deste problema, o próprio fato de que direcionalidade do sensor é baixa, i.e. o lobo central é largo, implica numa imprecisão na medida obtida, pois não é possível associar a distância lida a um lugar específico, mas sim à região no espaço coberta pelo lobo central [12].

3.3.2.3 Resposta no Ambiente Alvo

Por ser um sensor refletivo, a performance do sonar é significativamente afetada pelas características do alvo [13]. Um dos problemas decorrentes desse fato é que determinados objetos apresentam elevada taxa de absorção ou, ao contrário, são atravessados pela radiação o que resulta, em ambos os casos, que pouca ou nenhuma energia retorna ao sensor. Dessa forma, estes objetos são invisíveis para o dado método de medição; materiais como espuma, pele e roupas podem absorver as ondas acústicas [12] enquanto objetos com áreas superficiais pequenas, e.g. mesas e cadeiras, podem não ser detectados [2]. Vale ressaltar que as propriedades de reflexão, absorção e transmissão são variáveis com a frequência e do tipo de radiação, acústica ou eletromagnética. Existem outros problemas relativos ao ambiente alvo que não são relacionados à absorção ou transmissão da radiação, mas à reflexão serão tratados nas seções subsequentes separadamente.

3.3.2.4 *Foreshortening*

Como a direcionalidade dos sensores ultrassônicos é baixa, isto é a largura de feixe do lobo central é alta, aproximadamente 30° , quando o alvo a ser detectado não está perpendicularmente posicionado em relação ao eixo acústico do sensor, o cone que formado pelo lobo principal atinge o objeto em instantes diferentes. Consequentemente, retorna ao sensor em instantes diferentes provocando um desvio na leitura da distância, fazendo com que o obstáculo pareça estar mais próximo do que está na realidade. Por isso este problema é denominado *foreshortening*

3.3.2.5 Reflexão especular e *Crosstalk*

Analisando ainda a situação em que o obstáculo não está perpendicular ao eixo acústico do sonar, tems que a onda emitida pode ser refletida de tal forma que não retorne ao sensor, caso este em que o obstáculo não é percebido; outra possibilidade é de que esta onda atinja outras superfícies até que por fim retorne ao sensor, desta forma a medida obtida indica que o alvo encontra-se mais distante do que realmente está, fenômeno denominado reflexão especular [3, 12, 13].

Quando utiliza-se uma matriz de sonares, este problema é agravado, pois pode provocar interferência entre sensores ou, do inglês, *crosstalk*. De modo que além da medida obtida estar errada, o posicionamento estimado do obstáculo será também errôneo [2], afinal pressupõe-se que o sinal de *echo* é oriundo do pulso de ondas emitido pelo próprio dispositivo. No entanto, diferentemente da reflexão especular, este problema pode ser amenizado de diferentes maneiras, vide [17, 18].

3.3.2.6 Tempo de Resposta

3.4 Módulo de rádio frequência

Módulo de rádio frequência (Fig. 1) de baixo custo e consumo cuja faixa de operação situa-se na banda S das ondas UHF (*Ultra High Frequency*), com uma porção dentro da banda ISM ². Algumas informações técnicas [19] de interesse estão listadas abaixo:

- Tensão de alimentação: 1,9V - 3,6V
- Antena em circuito impresso do tipo MIFA (*Meandered Inverted-F Antenna*) [20]
- Frequência de operação: 2,4GHz - 2,525GHz
- Modulação digital do tipo GFSK ³
- Apresenta até 126 canais de comunicação ⁴
- Taxas de bits: 250kbps, 1Mbps ou 2Mbps
- Potências de saída de transmissão: 0dBm, -6dBm, -12dBm e -18dBm

² maiores informações no apêndice

³ maiores informações no apêndice

⁴ Válido apenas para as taxas de 250kbps e 1 Mbps; a 2Mbps este valor cai à metade, i.e. 63 canais.

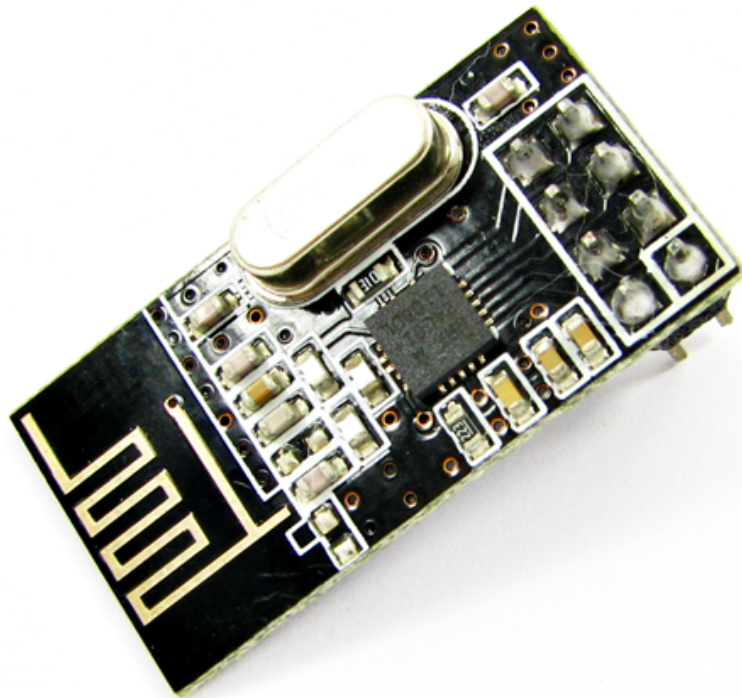


Figura 1 – Módulo de Rádio Frequência baseado no *transceiver* da Nordic nRF24L01+

- Interface com o microcontrolador por SPI ⁵ à taxa de até 10Mbps
- Pinos de entrada tolerantes a até 5V
- Pacotes recebidos verificados automaticamente , certificando-se da validade do endereço apontado e legitimando a integridade do pacote via CRC(*Cyclic Redundancy Check*) ⁶, antes de serem movidos às filas de dados recebidos (*RX FIFO*)
- Receptor envia ao transmissor um pacote de confirmação de recepção dos dados pelo mesmo canal (*acknowledgment packet*).

3.5 Arduino

Trata-se de uma plataforma de prototipação eletrônica aberta, i.e. *open-source hardware*, baseada no microcontrolador de 8 bits da Atmel ATmega328 [21], programável via serial (ICSP) através de um microcomputador, por exemplo, por meio do ambiente de desenvolvimento *Arduino Software IDE*, *open-source software* e encontra-se no GitHub [22]. Para programar este dispositivo, foi utilizado um módulo baseado na ponte USB-Serial PL-2303, cuja descrição detalhada pode ser encontrada em [23].

Algumas informações técnicas [24] de interesse estão listadas abaixo:

⁵ maiores informações no apêndice

⁶ Vide apêndice para uma breve explicação sobre CRC

- Dimensões: 17,78mm x 33mm
- Tensão de alimentação recomendável: 5V - 12V
- Memória
 - Flash: 32kB
 - SRAM: 2kB
 - EEPROM: 1kB
- 20 portas digitais de entrada/saída, das quais 6 podem ser usadas como saídas PWM
- 6 portas de entrada analógicas
- *clock* de 16MHz

3.6 Bateria

Baterias do tipo LiPo são uma das mais indicadas para veículos elétricos e híbridos, tanto quanto para equipamentos eletrônicos portáteis; no entanto, alguns cuidados precisam ser tomados ao manipulá-la por serem sensíveis a sobrecarga ou descarga abrupta. Logo, por questões de segurança e eficiência é necessário haver um sistema eletrônico para gerenciar a recarga deste dispositivo, o qual monitora a tensão cada uma das células assim como a temperatura em pontos específicos [25]. Neste trabalho foi utilizado o carregador IMAX B6-AC para fazer este serviço.

3.6.1 *C rate*

É um parâmetro que descreve a corrente de descarga da bateria em relação à sua capacidade nominal [26]. Vide a Eq. 3.3 para um exemplo ilustrativo baseado na bateria utilizada neste projeto.

$$30C = \frac{I_{descarga}}{2.800mAh} \Rightarrow I_{descarga} \approx 10.7A \quad (3.3)$$

4 Método

4.1 Estratégia *bottom-up*

A estratégia gerencial e organizacional *bottom-up* foi utilizada no desenvolvimento deste projeto. Em função da natureza modular e orientada a comportamentos de arquiteturas reativas [2], a adoção deste método de gerenciamento é quase que uma escolha natural. O robô foi dividido em quatro subsistemas, desenvolvidos e testados separadamente: percepção, locomoção, comunicação e navegação (que, no caso deste projeto, consiste no desvio de obstáculos em si). Nesta fase de implementação, boas práticas de engenharia de software foram tidas como prioridade, buscando uma implementação que apresente baixo acoplamento e alta coesão com a expectativa de desenvolver um código que possa ser facilmente entendido e reutilizado em futuros trabalhos afins. Em seguida se deu a etapa de integração das partes para, a posteriori, serem feitos testes no conjunto, conforme ilustra o diagrama 3:

4.2 Arquitetura Reativa

Optou-se por uma arquitetura de controle fortemente baseada nas informações sensoriais, sem delongas em processamento de sinal para ajustar os dados dos sensores a um modelo ou representação de mundo preconcebido. A razão dessa escolha é decorrência da necessidade de uma resposta rápida do sistema, vantagem da arquitetura reativa em função da sua simplicidade [3].

A latência inerente à obtenção dos dados dos sonares [27] associada à alta velocidade de operação do robô é a causa desta restrição temporal. Como é imprescindível colher dados do ambiente externo a uma taxa que dê um panorama atualizado do que está se passando ao redor do robô [1], reduzir o tempo de resposta do sistema possibilita que o desvio de obstáculos ocorra de maneira mais suave. Haja vista que se a detecção for feita com antecedência, medidas menos bruscas podem ser adotadas; em contraste com o caso em que a latência é alta a ponto de que percepção das barreiras no caminho se dê na proximidade do veículo.

Os comportamentos implementados no robô se restringem às diferentes manobras de evasão, adotadas com base na proximidade de obstáculos dos cinco sensores, vide 1. Os estímulos reguladores consistem na recepção, via RF, de um comando que incite o robô a navegar e do aval mediante recebimento de uma mensagem nos últimos 500ms informando se o botão de segurança foi acionado.

4.3 Subsistema de Locomoção

Numa visão geral, temos que o Arduino é responsável por emitir um sinal de controle, modulado em largura de pulso, ao ESC. Este, por sua vez, é incumbido de energizar os devidos enrolamentos do estator a fim de que o motor BLDC atinja, o mais breve possível, a velocidade desejada, expressa pelo sinal de controle. Resumidamente: o Arduino comanda, o ESC acata a

ordem e conduz o motor a cumprí-la utilizando os recursos da bateria. O robô apresenta tração dianteira e os motores estão fixos no chassi, logo, faz curvas quando há diferença de velocidade entre os motores.

O código fonte responsável pela produção do pulso PWM nas portas do Arduino foi desenvolvido por Sam Knight e disponibilizado ao público para utilização e modificações de qualquer natureza. Esta biblioteca, denominada PWM, pode ser encontrada no GitHub [28].

4.4 Subsistema de Percepção

O *software* que manipula os sensores ultrassônicos foi aperfeiçoado aos poucos. Primeiramente, buscou-se fazer o dispositivo funcionar, utilizando funções prontas e, portanto, não otimizadas de bibliotecas do Arduino. Em seguida, foi construída a matriz de sensores que, conforme a Fig. 4, tem o pino de *trigger* comum a todos sonares; no entanto, a priori, a leitura dos sonares era feita sequencialmente utilizando o código citado. O próximo passo, naturalmente, foi fazer com que os cinco sensores fossem lidos paralelamente, aproveitando o fato de todos dispararem juntos, a fim de minimizar o tempo de resposta na leitura da matriz.

Foram feitos testes mais rigorosos nessa última configuração com o objetivo de certificar se há de fato a necessidade de estipular um intervalo mínimo entre leituras sucessivas dos sonares ou se seria possível que esta latência fosse dinâmica, atrelada ao sensor cujo obstáculo encontra-se mais distante. Em suma, foi verificado se ciclos de leitura menores do que os 60ms sugeridos em [16] realmente ocasionam aumento na incidência de erros nas medidas; os detalhes acerca destes testes constam na seção de resultados.

A decisão de disparar todos os sonares simultaneamente foi feita com o intuito de reduzir o número de portas utilizadas no Arduino, assim como aumentar a taxa de obtenção dos dados, i.e. a largura de banda, conforme a terminologia adotada em [3]. No entanto, as consequências desta deliberação são severas: agravamento dos fenômenos de *foreshortening* e *crosstalk* [18].

4.5 Subsistema de Comunicação

Este segmento teve como alicerce a biblioteca denominada RF24, disponível em [29], responsável por todo o controle em baixo nível do *transceiver* nRF24L01+. Foi implementada em C++ e consiste numa única classe, RF24 (vide Fig.2), que provê acesso às funcionalidades básicas do *transceiver* como controle da potência de transmissão do sinal, escolha do canal a ser utilizado, tanto quanto funções que permitem enviar dados por um canal previamente aberto, ler dos canais em que o dispositivo se comporta como receptor; a documentação completa da classe pode ser encontrada em [30]. Assim como todos os códigos de terceiros e programas utilizados nesse projeto, sua utilização é aberta ao público gratuitamente, conforme os termos de uso.

Na definição do escopo do projeto, o papel do módulo de radiofrequência seria de simplesmente garantir a segurança e integridade do robô. Neste caso, uma comunicação *simplex* seria suficiente para cumprir a tarefa. O módulo transmissor, localizado no acionador remoto, enviava ao robô o nível lógico lido do botão de segurança. O *transceiver* do robô assumia o papel de receptor e enviava os dados recebidos por comunicação serial ao Arduino, que ordenava

a parada dos motores caso a mensagem indicasse que o botão estava desligado ou se nenhum pacote fosse detectado num período pré-determinado de 1 segundo.

No entanto, após concluir o sistema de acionamento sem fio, concebeu-se a ideia de sofisticar a utilização do módulo de radiofrequência, implementando uma interface de comando capaz de alterar e supervisionar os parâmetros e dados sensoriais do robô, com o intuito de facilitar a etapa de testes com o veículo em movimento, objetivando evitar ao máximo a necessidade de reprogramá-lo.

Ao adicionar essa funcionalidade, surge a necessidade de que ambas partes, i.e. robô e sistema de controle remoto, possam receber e enviar informações um ao outro. Como o *transceiver* utilizado tem a funcionalidade de estabelecer comunicação *half-duplex* para cada canal, i.e. bidirecional mas não simultaneamente, pois o receptor pode inserir dados no pacote de confirmação de recepção, *acknowledgment packet* [19], e a biblioteca RF24 apresenta funções prontas que facilitam o emprego deste recurso, foi possível adicionar essa funcionalidade ao projeto sem a necessidade de utilizar dois canais de comunicação.

A interface de comando implementada abrange as seguintes funções:

- Ajustar a frequência do PWM de cada um dos motores.
- Ajustar a velocidade angular dos motores, que corresponde ao *duty cycle* do sinal de controle, modulado em largura de pulso.
- Enviar parâmetros do robô ao controlador remoto: frequência dos PWMs, velocidades dos motores, leituras dos sensores ultrassônicos, *status* do botão de segurança de acordo com o veículo.
- Energizar os motores na velocidade estipulada enquanto o botão de segurança estiver acionado e não houver obstáculos que representem perigo ao robô.
- Acionar o sistema de navegação autônoma, também subordinado ao botão de segurança, com tentativas de envio das informações sensoriais e comportamentais a cada tomada de decisão do veículo ao controlador remoto sem suspender a movimentação do robô.
- Acionar o sistema de navegação autônoma por um número pré-estabelecido de leituras dos sonares, seguido de envio de todos dados coletados ao controlador remoto com o robô parado.

4.6 Subsistema de Navegação

Consiste na inteligência do robô, isto é, trata-se do conjunto de comportamentos adotados pelo veículo, através dos quais ele é capaz de desempenhar sua função de desvio de obstáculos. Tal qual [31], a área coberta por um dado sensor ultrassônico foi dividida em três regiões: distante, próxima e perigo. Quando a leitura de todos os sonares indica região distante, i.e. obstáculos distam mais do que 3 metros, considera-se que o robô está seguro e pode andar em velocidade máxima; em futuros trabalhos, corresponderá à situação em que o controle do veículo é cedido ao MOSA. Caso a medida de algum dos sensores seja menor do que 1 metro - região de perigo -

entende-se que o robô está na iminência de uma colisão e deve freiar imediatamente. Quando nenhuma destas situações citadas ocorre, isto é, nenhum dos sonares da matriz está na região de perigo, mas há ao menos um deles que não está na região distante, por conseguinte na região próxima, entende-se que há um obstáculo passível de ser contornado.

A estratégia de desvio de obstáculos é semelhante à desenvolvida em [32], define comportamentos bem simples e diretos, como atos reflexos nos animais, garantindo rapidez de resposta uma vez que as leituras dos sensores já foram feitas, vide Fig. 5. Analisa-se cada um dos cinco sonares quanto à região em que se encontra a barreira identificada: 0, para região distante e 1, próxima; cada uma das 32 combinações possíveis apresenta um comportamento correspondente: seguir em frente, fazer uma curva aberta, moderada ou brusca. Na Tabela 1 utiliza-se ‘E’ e ‘D’ para designar curvas à esquerda e direita, respectivamente; enquanto os índices ‘L’, ‘M’ e ‘F’ caracterizam o quão acentuada vai ser a curva: leve, moderada ou fortemente.

4.7 Integração dos Subsistemas

Assim que todos os subsistemas foram implementados, testados e operavam isoladamente de maneira satisfatória, foram feitos testes no conjunto, que indicaram novos problemas a serem tratados. A maioria deles de ordem prática e facilmente contornáveis, no entanto um é digno de nota, pois implicou em uma mudança na disposição física do veículo.

Nos testes feitos, em determinadas condições o robô ia de encontro ao obstáculo ao invés de efetuar o desvio. Após serem analisados os dados dos sensores nessas circunstâncias, notou-se a existência de ruídos em sensores específicos, que causavam a adoção de comportamentos errados.

Ao perceber o problema, novos testes foram engendrados a fim de descobrir a sua natureza. As hipóteses concebidas eram as seguintes: *crosstalk*, curto circuito entre pinos do Arduino, falha na lógica do *software* ou interferência dos motores nos sensores.

Para eliminar a possibilidade de que uma das portas da placa de prototipação estivesse interferindo na outra de alguma maneira, mudou-se a ligação dos sensores ultrassônicos e o problema se manteve. O mesmo foi feito no *software*, i.e. foi alterada a disposição dos sensores ultrassônicos no código. Especificamente falando, foram trocados os parâmetros que correlacionam a ligação física do pino de *echo* dos dispositivos à sua variável correspondente na matriz de estruturas do tipo *sensor_t*, denominada no programa por *USS*, e, mais uma vez, o defeito persistiu. Adicionalmente a essa modificação no *software*, foram feitas medições com os motores desligados, nas quais a falha em questão não ocorreu, evidenciando que a natureza do problema não era do código.

O último teste realizado consistiu em afastar os sensores ultrassônicos dos motores, o que resultou na desapareição das leituras espúrias, confirmando a hipótese de que os BLDC causavam a distorção na percepção do robô. Para sanar essa questão, optou-se por erguer os sensores a uma altura na qual não houvesse interferência suficiente a ponto de provocar erros de medição.

5 Resultados

5.1 USS

[27](página 149) vibração residual do trigger pode causar falsas leituras no echo, talvez esse seja o problema no caso de detecções dinâmicas deixado de lado.

[12] (páginas 126-127) talvez estendendo o *blanking time* consigamos manter as detecções dinâmicas. Como fazer isso:

- colocar um delay antes de entrar no loop
- mexer diretamente no myPulsein() para garantir medidas abaixo de 5 cm não sejam processadas trabalhando nos IFs.

While comparison of consecutive readings is an efficient way for rejecting external erroneous readings, it is unsuitable for reducing crosstalk. [17]

6 Conclusão

Referências

- 1 BROOKS, R. A. Intelligence without representation. *Artificial intelligence*, Elsevier, v. 47, n. 1, p. 139–159, 1991.
- 2 MURPHY, R. *Introduction to AI robotics*. [S.l.]: MIT press, 2000.
- 3 ROMERO, R. A. F. et al. *Robótica Móvel*. [S.l.: s.n.], 2014.
- 4 PIRES, R. et al. Mosa-mission oriented sensor array: A proposal. In: *Proceedings of the XXXVII Conferencia Latinoamericana de Informática-CLEI*. [S.l.: s.n.], 2011. p. 1309–1318.
- 5 STOREY, N. R. *Safety critical computer systems*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1996.
- 6 DILLINGHAM, G. L. *Unmanned Aircraft Systems: Federal Actions Needed to Ensure Safety and Expand Their Potential Uses Within the National Airspace System*. [S.l.]: DIANE Publishing, 2008.
- 7 YEDAMALE, P. *AN885 - Brushless DC (BLDC) Motor Fundamentals*. 2003. Disponível em: <http://ww1.microchip.com/downloads/en/AppNotes/00885a.pdf>. Acesso em: 12/10/2016.
- 8 BROWN, W. *AN857 - Brushless DC Motor Control Made Easy*. 2002. Disponível em: <http://ww1.microchip.com/downloads/en/AppNotes/00857B.pdf>. Acesso em: 12/10/2016.
- 9 COTTER, R. J. Time-of-flight mass spectrometry. In: *Time-of-Flight Mass Spectrometry*. [S.l.: s.n.], 1994.
- 10 AXEL, L.; SHIMAKAWA, B. A.; MACFALL, J. A time-of-flight method of measuring flow velocity by magnetic resonance imaging. *Magnetic resonance imaging*, Elsevier, v. 4, n. 3, p. 199–205, 1986.
- 11 HAYASHI, K.; YOSHINO, K.; INUISHI, Y. Carrier mobilities in insulating polymers measured by time of flight method. *Japanese Journal of Applied Physics*, IOP Publishing, v. 14, n. 1, p. 39, 1975.
- 12 SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D. *Introduction to autonomous mobile robots*. [S.l.]: MIT press, 2011.
- 13 EVERETT, H. *Sensors for mobile robots: theory and application*. [S.l.]: AK Peters, Ltd., 1995.
- 14 BALANIS, C. A. *Antenna theory: analysis and design*. [S.l.]: John Wiley & Sons, 2016.
- 15 POZAR, D. M. *Microwave engineering*. [S.l.]: John Wiley & Sons, 2009.
- 16 ELECFREAKS. Disponível em: <http://www.micropik.com/PDF/HCSR04.pdf>. Acesso em: 18/10/2016.
- 17 BORENSTEIN, J.; KOREN, Y. Error eliminating rapid ultrasonic firing for mobile robot obstacle avoidance. *IEEE Transactions on Robotics and automation*, IEEE, v. 11, n. 1, p. 132–138, 1995.
- 18 SHOVAL, S.; BORENSTEIN, J. Using coded signals to benefit from ultrasonic sensor crosstalk in mobile robot obstacle avoidance. In: IEEE. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*. [S.l.], 2001. v. 3, p. 2879–2884.

- 19 SEMICONDUCTOR, N. nrf24l01+ single chip 2.4 ghz transceiver product specification v1. 0. *nRF24L01+ datasheet*. Disponível em: <<http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01>>. Acesso em: 05/10/2016.
- 20 CYPRESS SEMICONDUCTOR CORPORATION. *AN91445 - Antenna Design and RF Layout Guidelines*. [S.l.], 2016. Rev. D. Disponível em: <<http://www.cypress.com/documentation/application-notes/an91445-antenna-design-and-rf-layout-guidelines>>. Acesso em: 07/10/2016.
- 21 ATMEL CORPORATION. *ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES IN-SYSTEM PROGRAMMABLE FLASH - DATASHEET*. [S.l.]. Disponível em: <http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf>. Acesso em: 05/10/2016.
- 22 ARDUINO. Disponível em: <<https://github.com/arduino/Arduino>>. Acesso em: 05/10/2016.
- 23 PROLIFIC TECHNOLOGY INC. *PL-2303 Edition USB to Serial Bridge Controller Product Datasheet*. [S.l.], 2005. Rev. 1.6. Disponível em: <<http://www.electronicaestudio.com/docs/PL2303.pdf>>. Acesso em: 07/10/2016.
- 24 ARDUINO. Disponível em: <<http://www.arduino.org/products/boards/arduino-uno>>. Acesso em: 07/10/2016.
- 25 BARONTI, F. et al. Hierarchical platform for monitoring, managing and charge balancing of lipo batteries. In: IEEE. *2011 IEEE Vehicle Power and Propulsion Conference*. [S.l.], 2011. p. 1–6.
- 26 TEAM, M. *A Guide to Understanding Battery Specifications*. [S.l.]: December, 2008.
- 27 JONES, J. L.; FLYNN, A. M. *Mobile robots: inspiration to implementation*. [S.l.]: AK Peters, Ltd., 1993.
- 28 MANIACBUG. Disponível em: <<https://github.com/kmark/Captain/tree/master/CaptainShip/PWM>>. Acesso em: 17/10/2016.
- 29 MANIACBUG. Disponível em: <<https://github.com/maniacbug/RF24>>. Acesso em: 16/10/2016.
- 30 MANIACBUG. Disponível em: <<https://github.com/maniacbug/RF24>>. Acesso em: 19/10/2016.
- 31 GAGEIK, N.; MÜLLER, T.; MONTENEGRO, S. Obstacle detection and collision avoidance using ultrasonic distance sensors for an autonomous quadrocopter. *University Of Würzburg, Aerospace Information Technology (Germany), Würzburg September*, 2012.
- 32 DONGYUE, Q.; YUANHANG, H.; YUTING, Z. The investigation of the obstacle avoidance for mobile robot based on the multi sensor information fusion technology. *Int. J. Mat. Mech. Manuf*, v. 1, p. 366–370, 2013.
- 33 SILBERSCHATZ, A. et al. *Operating system concepts*. [S.l.]: Addison-Wesley Reading, 1998. v. 4.
- 34 STIGGE, M. et al. Reversing crc—theory and practice. Citeseer, 2006.
- 35 PETERSON, W. W.; BROWN, D. T. Cyclic codes for error detection. *Proceedings of the IRE*, v. 49, n. 1, p. 228–235, 1961.

-
- 36 BROWN, W. *Analog Pulse Width Modulation*. 2013. Disponível em: <<http://www.ti.com/lit/ug/slau508/slau508.pdf>>. Acesso em: 03/04/2017.

Apêndices

APÊNDICE A – Apêndices

A.1 Diagrama da Classe RF24

RF24
<pre> +Power Amplifier Level: rf24_pa_dbm_e = RF24_PA_MIN , RF24_PA_LOW, RF24_PA_HIGH, RF24_PA_MAX, RF24_PA_ERROR +CRC Length: rf24_crclength_e = RF24_CRC_DISABLED , RF24_CRC_8, RF24_CRC_16 +Data Rate: rf24_datarate_e = RF24_1MBPS = 0, RF24_2MBPS, RF24_250KBPS -ce_pin: uint8_t -csn_pin: uint8_t -wide_band: bool -p_variant: bool -payload_size;: uint8_t -ack_payload_available: bool -dynamic_payloads_enabled: bool -ack_payload_length: uint8_t +pipe0_reading_address: uint64_t #csn(mode:int): void #ce(level:int): void #read_register(reg:uint8_t,buf:uint8_t*,len:uint8_t): uint8_t #read_register(reg:uint8_t): uint8_t #write_register(reg:uint8_t,buf:const uint8_t*,len:uint8_t): uint8_t #write_register(reg:uint8_t,value:uint8_t): uint8_t #write_payload(buf:const void*,len:uint8_t): uint8_t #read_payload(buf:void*,len:uint8_t): uint8_t #flush_rx(): uint8_t #flush_tx(): uint8_t #get_status(): uint8_t #print_status(status:uint8_t): void #print_observe_tx(value:uint8_t): void #print_byte_register(name:const char*,reg:uint8_t,qty:uint8_t=1): void #print_address_register(name:const char*,reg:uint8_t,qty:uint8_t=1): void #toggle_features(): void RF24(_cepin:uint8_t,_cspin:uint8_t) +begin(): void +startListening(): void +stopListening(): void +write(buf:const void*,len:uint8_t): void +available(): bool +read(buf:void*,len:uint8_t): bool +openWritingPipe(address:uint64_t): void +openReadingPipe(number:uint8_t,address:uint64_t): void +setRetries(delay:uint8_t,count:uint8_t): void +setChannel(channel:uint8_t): void +setPayloadSize(size:uint8_t): void +getPayloadSize(): uint8_t +getDynamicPayloadSize(): uint8_t +enableAckPayload(): void +enableDynamicPayloads(): void +isPVariant(): bool +setAutoAck(enable:bool): void +setAutoAck(pipe:uint8_t,enable:bool): void +setPALevel(level:rf24_pa_dbm_e): void +getPALevel(): rf24_pa_dbm_e +setDataRate(speed:rf24_datarate_e): bool +getDataRate(): rf24_datarate_e +setCRCLength(length:rf24_crclength_e): void +getCRCLength(): rf24_crclength_e +disableCRC(): void +printDetails(): void +powerDown(): void +powerUp(): void +available(pipe_num:uint8_t*): bool +startWrite(buf:const void*,len:uint8_t): void +writeAckPayload(pipe:uint8_t,buf:const void*,len:uint8_t): void +isAckPayloadAvailable(): bool +whatHappened(tx_ok:bool&,tx_fail:bool&,rx_ready:bool&): void +testCarrier(): bool +testRPD(): bool +isValid(): bool </pre>

Figura 2 – Diagrama da Classe RF24

A.2 Estrutura Analítica do Projeto

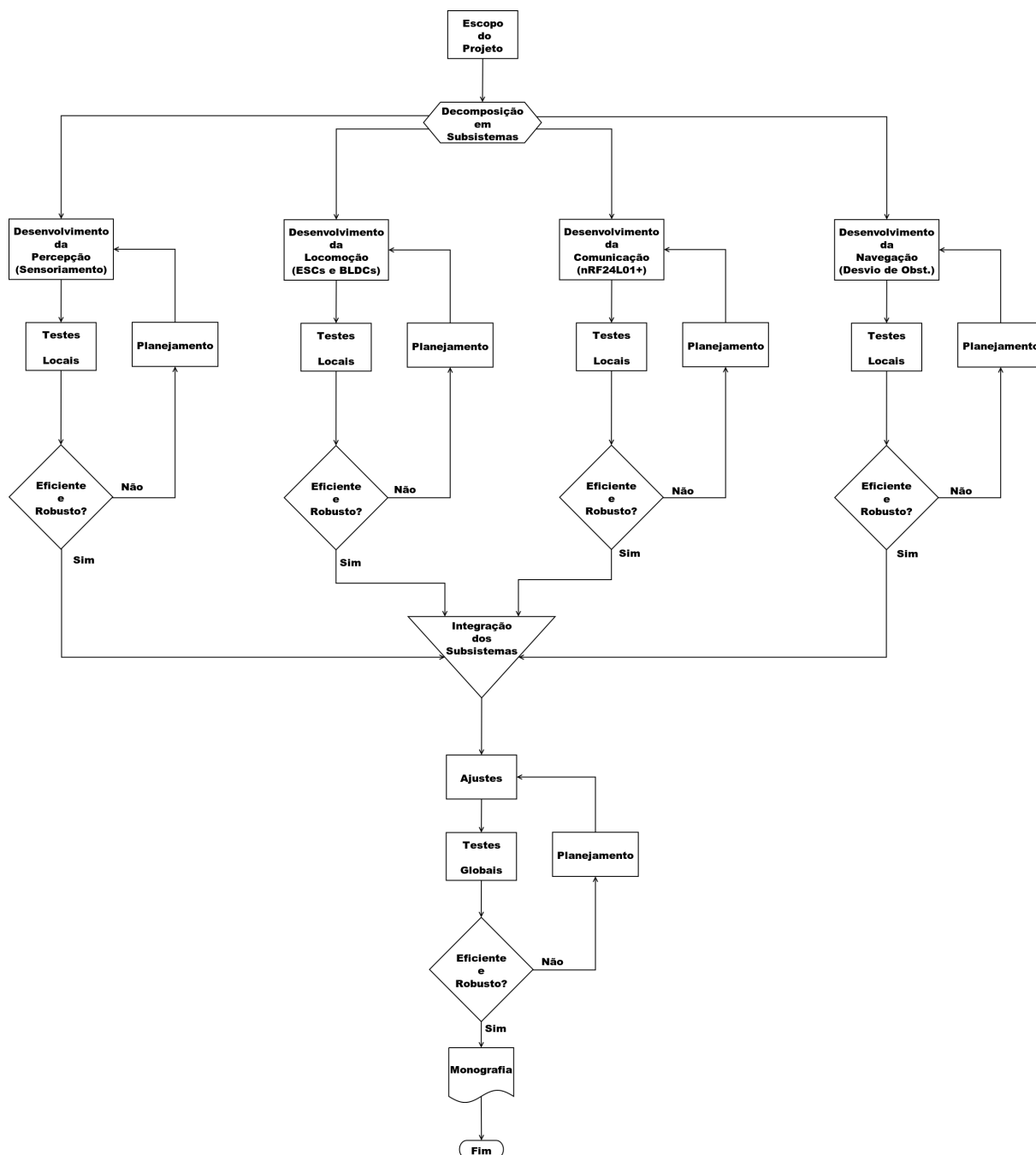


Figura 3 – Estrutura Analítica do Projeto

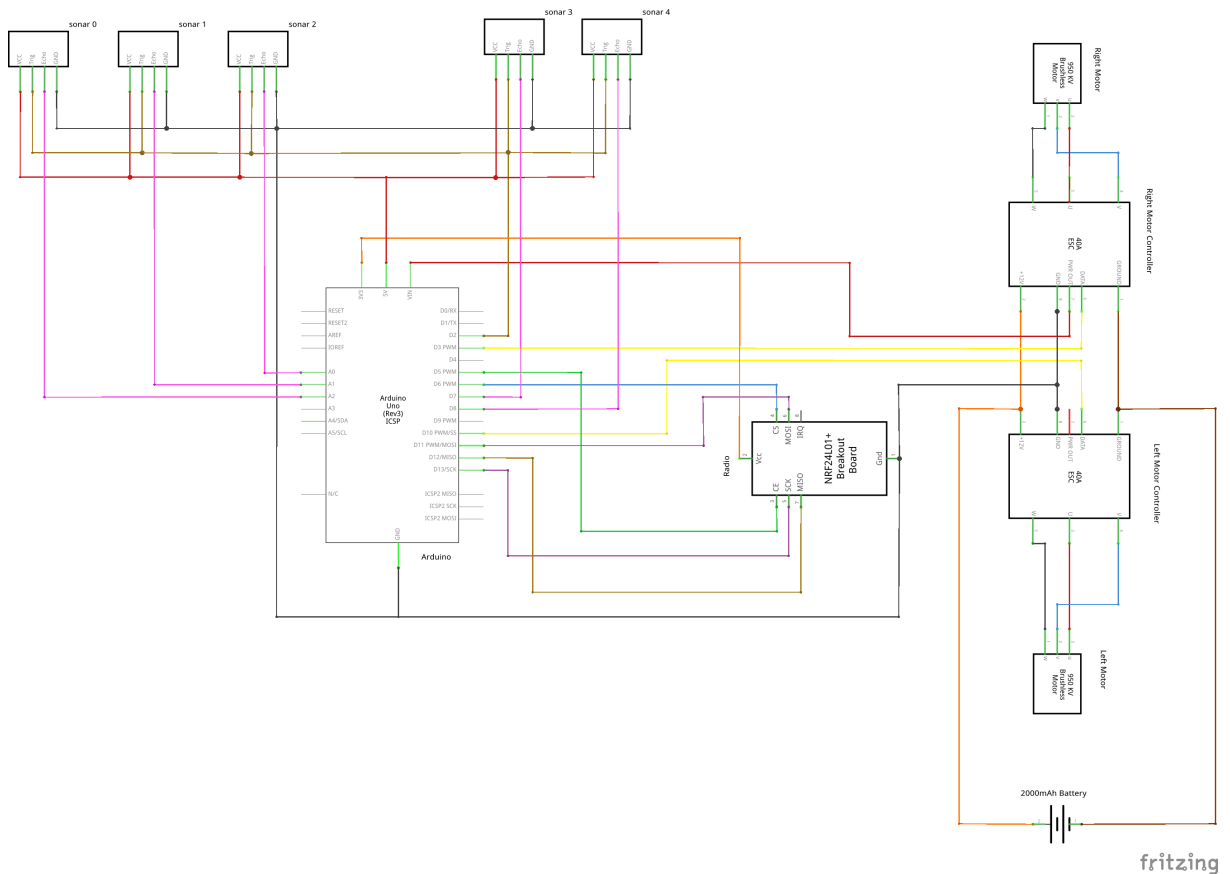


Figura 4 – Diagrama Elétrico

A.3 Esquemático do Robô

A.4 Tabela

A.5 Fluxograma da Rotina de Desvio de Obstáculos

A.6 Sistemas de Tempo Real

São sistemas computacionais que dependem não somente que os dados computados estejam corretos, mas que sejam obtidos dentro de um intervalo de tempo pré determinado, que pode ser maior ou menor de acordo com a aplicação. Na literatura, este período em que se espera que a resposta do sistema se dê é denominado *deadline*. Sistemas de tempo real podem ser classificados em dois tipos: *soft* ou *hard*.

Sistemas *soft* são menos restritivos, tolerando eventuais perdas de *deadline*; ao contrário dos sistemas *hard*, em que estas perdas não são aceitáveis.

Algumas características típicas, apesar de não obrigatórias, de sistemas de tempo real são limitações com relação ao tamanho, propósito específico e baixo custo [33].

Tabela 1 –

USS_0	USS_1	USS_2	USS_3	USS_4	Ação
0	0	0	0	1	E_L
0	0	0	1	0	E_M
0	0	0	1	1	E_M
0	0	1	0	0	E_F
0	0	1	0	1	E_F
0	0	1	1	0	E_F
0	0	1	1	1	E_F
0	1	0	0	0	D_M
0	1	0	0	1	D_M
0	1	0	1	0	E_F
0	1	0	1	1	E_M
0	1	1	0	0	D_F
0	1	1	0	1	E_F
0	1	1	1	0	E_F
0	1	1	1	1	E_F
1	0	0	0	0	D_L
1	0	0	0	1	Frente
1	0	0	1	0	E_M
1	0	0	1	1	E_M
1	0	1	0	0	D_F
1	0	1	0	1	E_F
1	0	1	1	0	E_F
1	0	1	1	1	E_F
1	1	0	0	0	D_M
1	1	0	0	1	D_M
1	1	0	1	0	D_M
1	1	0	1	1	Frente
1	1	1	0	0	D_F
1	1	1	0	1	D_F
1	1	1	1	0	D_F
1	1	1	1	1	E_F

A.7 CRC

Método de detecção de erros aleatórios, isto é, de dados corrompidos ao longo do processo de transmissão ou armazenamento da informação por exemplo por ruídos, mas não por um agente ‘inteligente’ externo que modifique os dados transmitidos, tal qual um *malware* [34].

Consiste essencialmente em uma divisão polinomial [34], logo, pode ser implementado em *hardware*, utilizando-se apenas registradores de deslocamento com conexões realimentadas [35], assim como em *software*. Em suma, trata-se de acrescentar à mensagem digital original um sufixo, que tem seu valor definido por operações realizadas em função da mensagem binária que intenta-se enviar e de um polinômio gerador. Para o *transceiver* nRF24L01+, dois polinômios geradores são utilizados: Eq. A.1 quando o dado cíclico adicionado é de 1 *byte*, e Eq. A.2, para 2 *bytes* [19].

Para uma descrição completa de como é implementado este método, vide [34, 35].

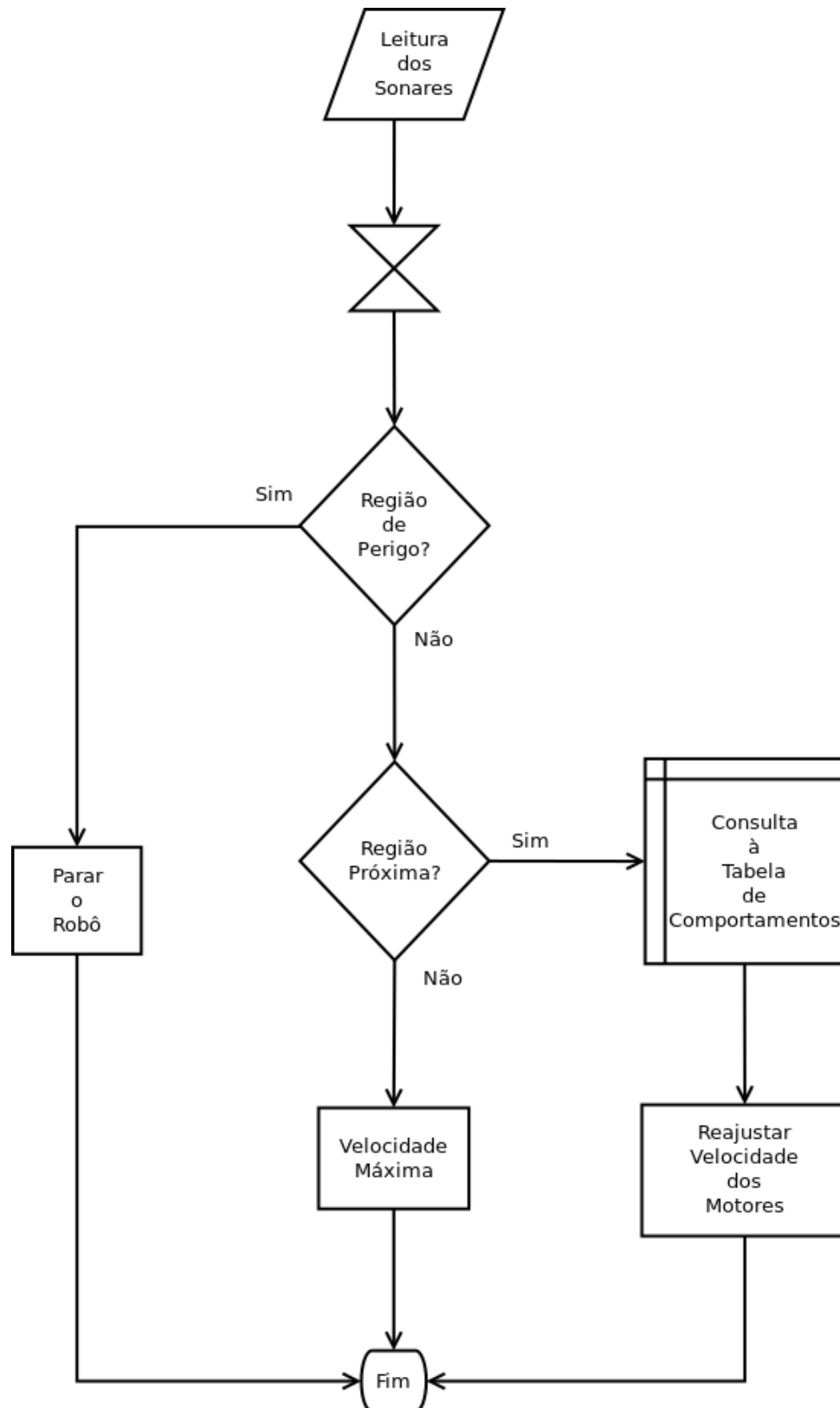


Figura 5 – Fluxograma da Rotina de Desvio de Obstáculos

$$G(X) = X^8 + X^2 + X + 1 \quad (\text{A.1})$$

$$G(X) = X^{16} + X^{12} + X^5 + 1 \quad (\text{A.2})$$

A.8 MIFA

A.9 efeito hall

A.10 campo girante

A.11 *schmitt trigger*

A.12 PWM

A modulação por largura de pulso é uma técnica de modulação que consiste em amostrar e codificar o sinal correspondente à mensagem na largura de um trem de pulsos de amplitude fixa, i.e. cada amostra da mensagem é convertida em um pulso retangular cuja duração expressa a amplitude do sinal modulante.

Um modulador PWM pode ser implementado utilizando-se um circuito comparador não inversor em cuja entrada inversora aplica-se uma onda do tipo dente de serra (*trailing edge modulation*), dente de serra invertida (*leading edge modulation*) ou triangular (*modulation on both edges*), cuja frequência corresponderá à amostragem da modulação **citation needed!!!**, e na entrada não inversora, o sinal modulante, conforme ilustra a 6. Desta forma, quando a tensão da

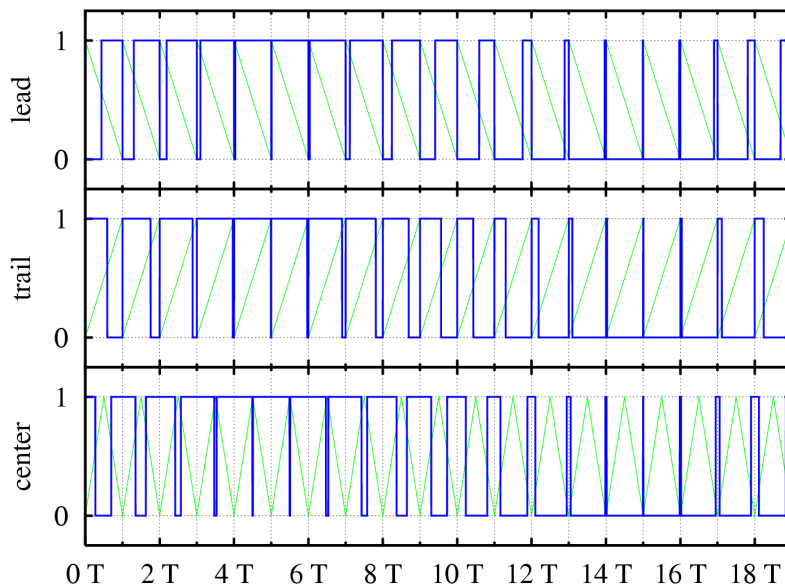


Figura 6 – Três tipos de modulação PWM: *trailing edge*, *leading edge* e *both edges*, de cima pra baixo, respectivamente.

mensagem excede a amplitude da portadora, tem-se um sinal alto na saída, já no caso contrário, baixo. para uma descrição detalhada do circuito e simulações vide [36]

A.13 GFSK

A.14 banda ISM

A.15 SPI

ANEXO A – Anexos