

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

24 de outubro de 2016

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

24 de outubro de 2016

*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 amparam a germinação de verdadeiros pomares. 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+ .	29
Figura 2 – Diagrama da Classe RF24	45
Figura 3 – Estrutura Analítica do Projeto	46
Figura 4 – Diagrama Elétrico	47
Figura 5 – Fluxograma da Rotina de Desvio de Obstáculos	48

Lista de tabelas

Tabela 1	–	44
----------	-------------	----

Lista de abreviaturas e siglas

CC	
PWM	<i>Pulse Width Modulation</i>
ESC	<i>Electronic Speed Controller</i>
IDE	<i>IDE</i>
ISM	<i>ISM</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>UBEC</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	Arquitetura Reativa	23
2.2	Arquitetura MOSA	24
3	MATERIAIS	25
3.1	Motor <i>Brushless</i>	25
3.2	ESC	26
3.3	Sensor Ultrassônico	26
3.3.1	Princípio de Funcionamento	26
3.3.2	Limitações	27
3.3.2.1	Variação na velocidade de propagação da onda acústica	27
3.3.2.2	Direcionalidade	27
3.3.2.3	Resposta no Ambiente Alvo	27
3.3.2.4	<i>Foreshortening</i>	28
3.3.2.5	Reflexão especular	28
3.3.2.6	<i>Crosstalk</i>	28
3.3.2.7	Tempo de Resposta	28
3.4	Módulo de rádio frequência	28
3.5	Arduino	29
3.5.1	Arduino Uno	29
3.5.2	Arduino Pro Mini	30
3.6	Bateria	30
3.6.1	<i>C rate</i>	30
4	MÉTODO	31
4.1	Estratégia <i>bottom-up</i>	31
4.2	Arquitetura Reativa	31
4.3	Subsistema de Locomoção	31
4.4	Subsistema de Percepção	32
4.5	Subsistema de Comunicação	32
4.6	Subsistema de Navegação	34
4.7	Integração dos Subsistemas	34
5	RESULTADOS	35
5.1	USS	35
6	CONCLUSÃO	37

REFERÊNCIAS	39
APÊNDICES	41
APÊNDICE A – APÊNDICES	43
A.1 Sistemas de Tempo Real	43
A.2 PWM	43
A.3 GFSK	43
A.4 banda ISM	43
A.5 SPI	43
A.6 CRC	43
A.7 MIFA	44
A.8 efeito hall	44
A.9 campo girante	44
A.10 <i>schmitt trigger</i>	44
A.11 Diagrama da Classe RF24	44
A.12 Estrutura Analítica do Projeto	44
A.13 Esquemático do Robô	44
A.14 Tabela	44
A.15 Fluxograma da Rotina de Desvio de Obstáculos	45
ANEXO A – ANEXOS	49

1 Introdução

2 Embasamento Teórico

2.1 Arquitetura Reativa

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

Brooks, R.A [1] - tradução livre -

A ideia básica que fundamenta esta arquitetura é de que tudo aquilo conquistado até hoje pelos seres vivos, incluindo a própria racionalidade e inteligência humana, se deu exclusivamente graças às nossas faculdades de percepção, mobilidade e manutenção da própria sobrevivência [1]. Denominemos agentes primitivos aqueles indivíduos cujos comportamentos se reduzem exclusivamente aos seus instintos de auto-preservação. Sendo que, de acordo com [2] numa tradução livre, “comportamentos são mapeamentos diretos das informações sensoriais recebidas em um padrão de ações motoras, utilizadas a fim de se cumprir uma tarefa”

Sob esta ótica, temos que estes agentes primitivos desenvolveram novos comportamentos ao longo de seu processo evolutivo, que nada mais é do que uma consequência das faculdades deste indivíduo de se mover e perceber o mundo, de forma que estes novos comportamentos obtidos se relacionam com os demais de três maneiras possíveis. A primeira consistiria na reutilização de outros comportamentos que lhe são ancestrais, o que seria um refinamento do seu proceder: a criação de uma nova camada comportamental que envolve as ancestrais, análogo à constituição de uma pérola. O segundo meio seria negando comportamentos antecessores, o que constituiria uma mudança de postura. Enquanto que a última maneira seria a simples criação de uma nova conduta paralela, o que pode ser visto como um comportamento mais avançado que se empilha sobre os demais ao invés de envolvê-los ou negá-los[2].

A noção de que toda ação é fruto de um comportamento alicerça o paradigma reativo. Temos portanto que a ação está intimamente ligada à percepção através do comportamento. Nesta perspectiva, quando afastamos a mão ao tocarmos uma panela quente, podemos entender que a nossa ação foi consequência de um comportamento que nos impele a distanciarmo-nos de objetos quentes, detectados pelos nossos sensores de calor da pele.

Como consequência desta prerrogativa, 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], pode ser descartada por não ser necessária. Em suma, os comportamentos se dão de acordo com o que está acontecendo no ambiente externo, informação obtida através da leitura dos sensores que compõem a percepção do robô. Nenhuma modelagem ou representação deste ambiente externo é feita, apenas medições locais e orientadas a comportamentos são adotadas.

2.2 Arquitetura MOSA

Arquitetura que propõe dividir o sistema aéreos 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 *Electronic Speed Controlers (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
- 1 Arduino Uno
- 1 Arduino Pro Mini
- 1 Conversor/Adaptador USB-Serial PL2303
- 1 Carregador 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 comutados 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 seu rotor consiste de um ímã permanente, enquanto seu estator apresenta pólos formados por enrolamentos. Como é preciso que os enrolamentos sejam energizados na sequência correta para que um campo magnético girante seja criado de modo a produzir o maior torque. 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 então se faça o acionamento na ordem correta. Existem dois meios de se obter esta informação: sensores de efeito hall, método empregado neste trabalho, ou processamento da força contra eletromotriz dos enrolamentos 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

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

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 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 atmosférica 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

Dentro desta mesma situação em que o obstáculo não está perpendicular ao eixo acústico do sonar, existe a possibilidade de que a onda emitida seja refletida de tal forma que não retorne ao sensor, caso este em que o obstáculo não é percebido. Outra hipótese é de que esta onda refletida para longe do sensor atinja outra(s) superfície(s) até que retorne ao sensor, desta forma a medida obtida indica que o alvo encontra-se mais distante do que realmente está [3, 12, 13].

3.3.2.6 *Crosstalk*

Quando utiliza-se uma matriz de sensores ultrassônicos, o problema da reflexão especular é agravado, pois pode constituir interferência entre sensores. De modo que além da medida feita estar errada, o posicionamento do obstáculo será também errôneo [2]. No entanto, diferentemente da reflexão especular, este problema pode ser amenizado de diferentes maneiras, vide [17, 18]

3.3.2.7 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.

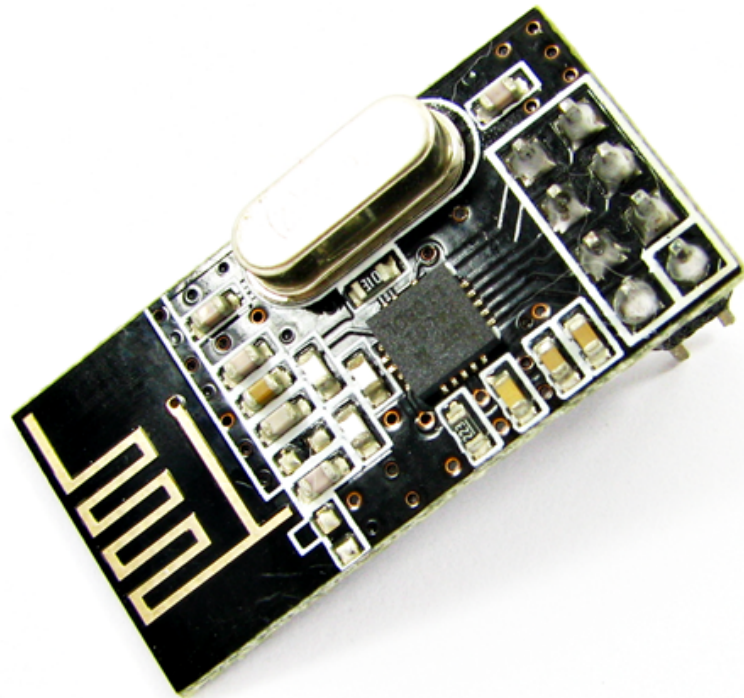


Figura 1 – Módulo de Rádio Frequência baseado no *transceiver* da Nordic nRF24L01+

- 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

3.5.1 Arduino Uno

Trata-se de uma plataforma de prototipação eletrônica aberta, i.e. *open-source hardware*, baseada no microcontrolador de 8 bits da Atmel ATmega328P [21], programável via USB ou 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]. Algumas informações técnicas [23] de interesse estão listadas abaixo:

⁵ maiores informações no apêndice

⁶ Vide apêndice para uma breve explicação sobre CRC

- Dimensões: 53.4mm x 68.6mm
- Tensão de alimentação recomendável: 7V - 12V
- Fonte de tensão de 3,3V embutida
- 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

3.5.2 Arduino Pro Mini

Apresenta especificações análogas às do Arduino Uno, porém com a desvantagem de não ser programável via USB. No entanto apresenta uma compliância maior da tensão de alimentação recomendável, 5V - 12V, e dimensões reduzidas, 17,78mm x 33mm. O microcontrolador utilizado nesta placa é o ATMega328, cuja descrição também pode ser encontrada em [21]. Para programar este dispositivo, foi utilizado um módulo baseado na ponte USB-Serial PL-2303, cuja descrição detalhada pode ser encontrada em [24].

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 do projeto. Em função da natureza modular e orientada a comportamentos da arquitetura reativa [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 acoplamento¹ e alta coesão² [2] com a expectativa de desenvolver um código que possa ser facilmente entendido e reutilizado. 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 ultrassônicos a um conjunto de regras ou a um modelo 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 na 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.

4.3 Subsistema de Locomoção

Numa visão geral, temos que o Arduino Uno é responsável por emitir o 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 no 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 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

¹ Acoplamento: grau de dependência de ligações e interface externas a um dado módulo

² Coesão: grau de independência em relação a operações e dados externos a um dado módulo

qualquer natureza. Esta biblioteca, denominada PWM, pode ser encontrada no GitHub [28].

O robô apresenta tração dianteira, logo, faz curvas quando há diferença de velocidade entre os motores.

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 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, buscou verificar se ciclos de leitura menores do que os 60ms sugeridos em [16] ocasionam aumento na incidência de erros de 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 Uno e 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 mais 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 usado tanto quanto funções que permitem enviar dados por um canal previamente aberto assim como ler as informações recebidas nos 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.

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, que constitui o acionador remoto, foi ligado ao Arduino Pro Mini, que enviava ao robô o nível lógico lido pelo botão de segurança. O *transceiver* do robô assumia o papel de receptor e enviava os dados recebidos por comunicação serial ao Arduino Uno, que parava os 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. Logo, 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. O *transceiver* utilizado neste projeto 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].

Contudo, o número de *bytes* disponibilizados ao receptor para incorporar sua mensagem no pacote de confirmação não atinge a metade do que o transmissor pode enviar. Além disso, os dados a serem integrados no *acknowledgment packet* precisam ser decididos antes do recebimento do pacote do transmissor, ou seja, a informação que o receptor envia ao transmissor é resposta ao pacote enviado anteriormente, e não ao mais atual. Por causa destes pormenores, optou-se por ignorar essa ferramenta do módulo e implementar uma comunicação através de dois canais, de tal forma que o dispositivo transmissor em um canal é receptor no outro, e vice versa.

A ideia inicial concebida de como estabelecer essa interface de comunicação propunha que cada uma das partes deveria transmitir e enviar uma mensagem a cada iteração de modo bloqueante, ou seja, não poderia haver mudança de estado antes do cumprimento da sua respectiva incumbência. Isto é, podemos entender como uma máquina de dois estados, transmissão e recepção, que faz transição entre estes somente quando recebe o pacote de notificação de entrega e quando recebe uma mensagem, respectivamente. No entanto, esta ideia teve de ser reformulada em função do recorrente surgimento de impasses (*deadlocks*): ambos elementos tentavam transmitir ou receber ao mesmo tempo, de modo que ficavam presos nesta condição pois um aguardava que o outro mudasse de estado. Algumas hipóteses foram formuladas para tentar explicar esse fenômeno, mas sem conclusões cabais, de modo que não serão detalhadas nesse documento.

Para contornar este empecilho, implementou-se uma rotina de tratamento de impasses que gera um número pseudo-aleatório entre dois valores pré-estabelecidos que, caso seja par implica que o módulo deve inverter o seu estado, e.g. se estiver transmitindo, passa a receber; caso seja ímpar, o número em questão corresponde ao intervalo máximo, em milissegundos, que o dispositivo deve permanecer no estado em que se encontra antes de retornar a esta função, no caso de não conseguir estabelecer o contato. Toda vez que essa rotina é acionada, aumenta-se o intervalo que delimita o número aleatório gerado em valor fixo pré determinado.

A interface de comando foi desenvolvida com o intuito de facilitar a etapa de testes com o robô em movimento, objetivando evitar ao máximo a necessidade de reprogramá-lo. Para isso, as seguintes funcionalidades foram implementadas:

- 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

Tal qual [31], a área coberta por um 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 de um dado sensor, 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, o que corresponde à região de perigo, entende-se que o robô está na iminência de uma colisão e que deve, portanto, 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, e sim 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, 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

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 ARDUINO. Disponível em: <<http://www.arduino.org/products/boards/arduino-uno>>. Acesso em: 07/10/2016.
- 24 PROLIFIC TECHNOLOGY INC. *PL-2303 Edition USB to Serial Bridge Controller Product Datasheet*. [S.l.], 2005. Rev. 1.6. Disponível em: <<http://www.electronicastudio.com/docs/PL2303.pdf>>. 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.

Apêndices

APÊNDICE A – Apêndices

A.1 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].

A.2 PWM

A.3 GFSK

A.4 banda ISM

A.5 SPI

A.6 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].

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

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

A.7 MIFA

A.8 efeito hall

A.9 campo girante

A.10 *schmitt trigger*

A.11 Diagrama da Classe RF24

A.12 Estrutura Analítica do Projeto

A.13 Esquemático do Robô

A.14 Tabela

Tabela 1 –

USS_0	USS_1	USS_2	USS_3	USS_4	Action
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

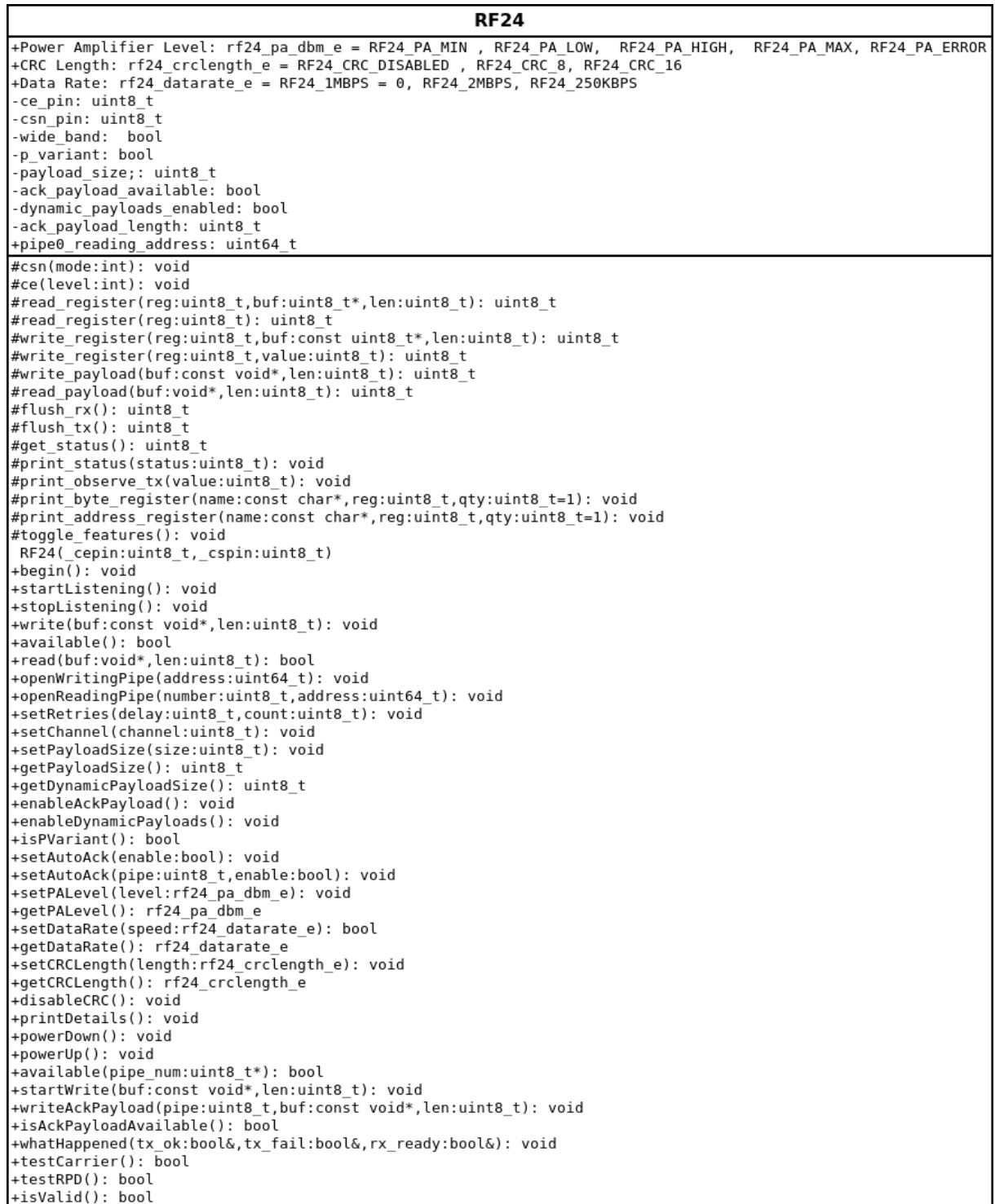


Figura 2 – Diagrama da Classe RF24

A.15 Fluxograma da Rotina de Desvio de Obstáculos

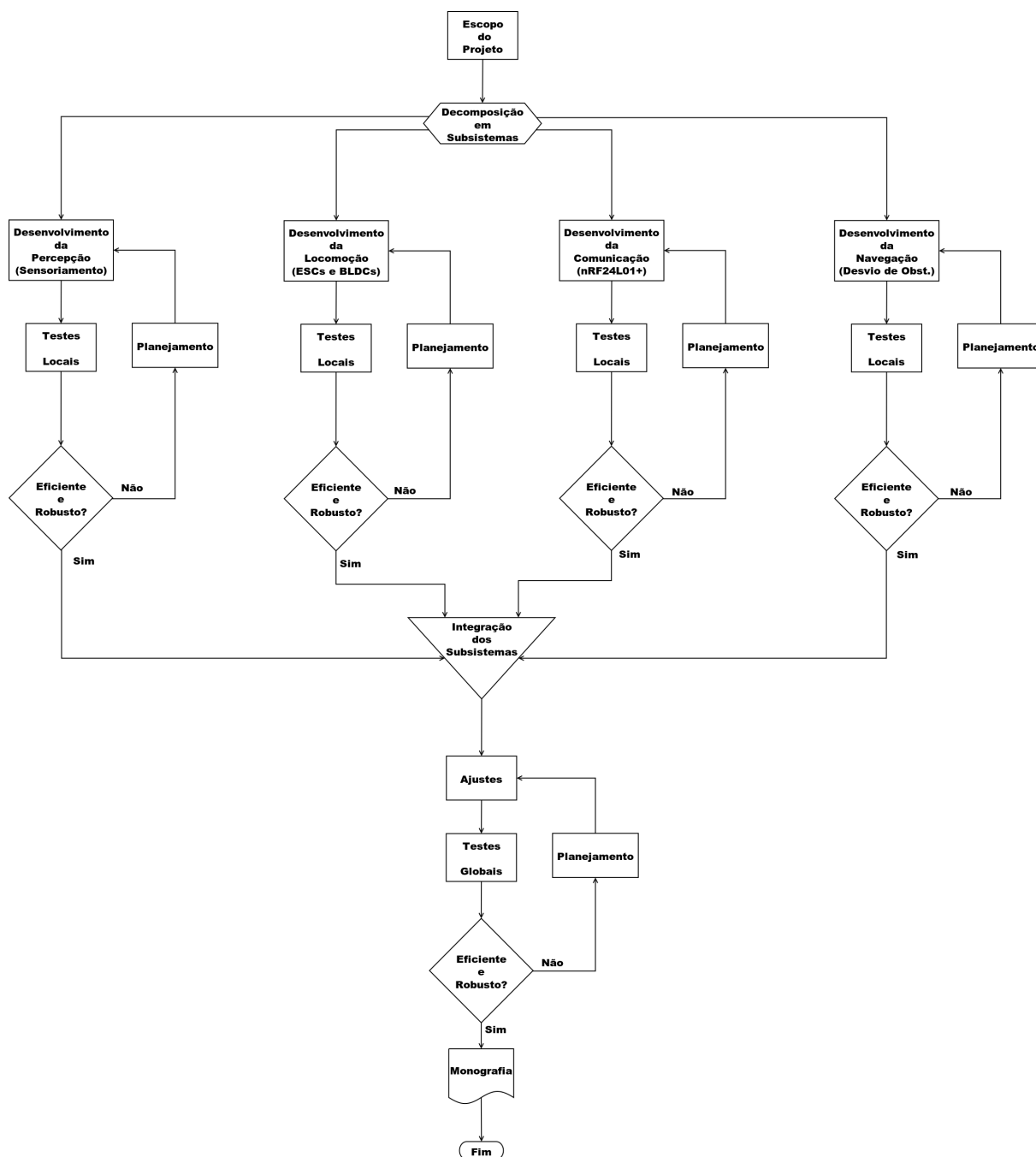
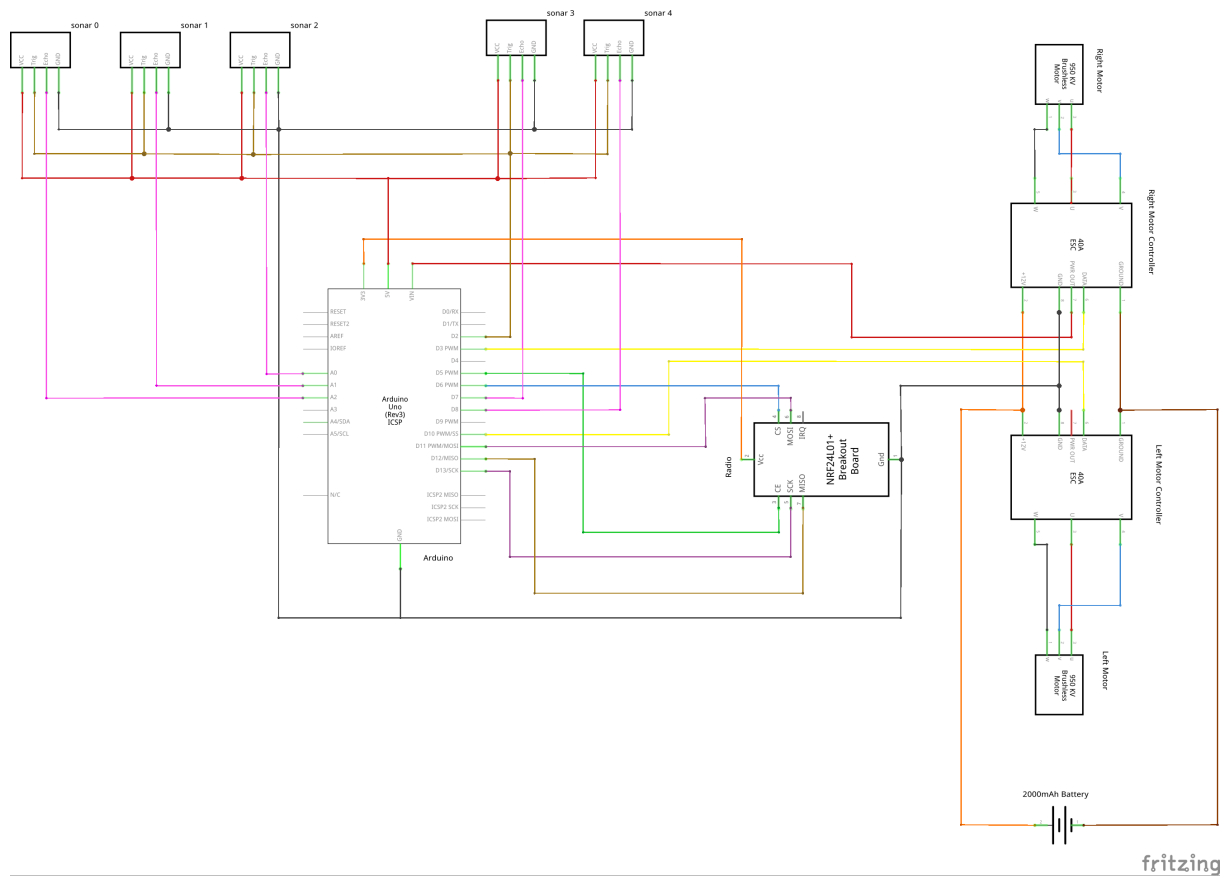


Figura 3 – Estrutura Analítica do Projeto



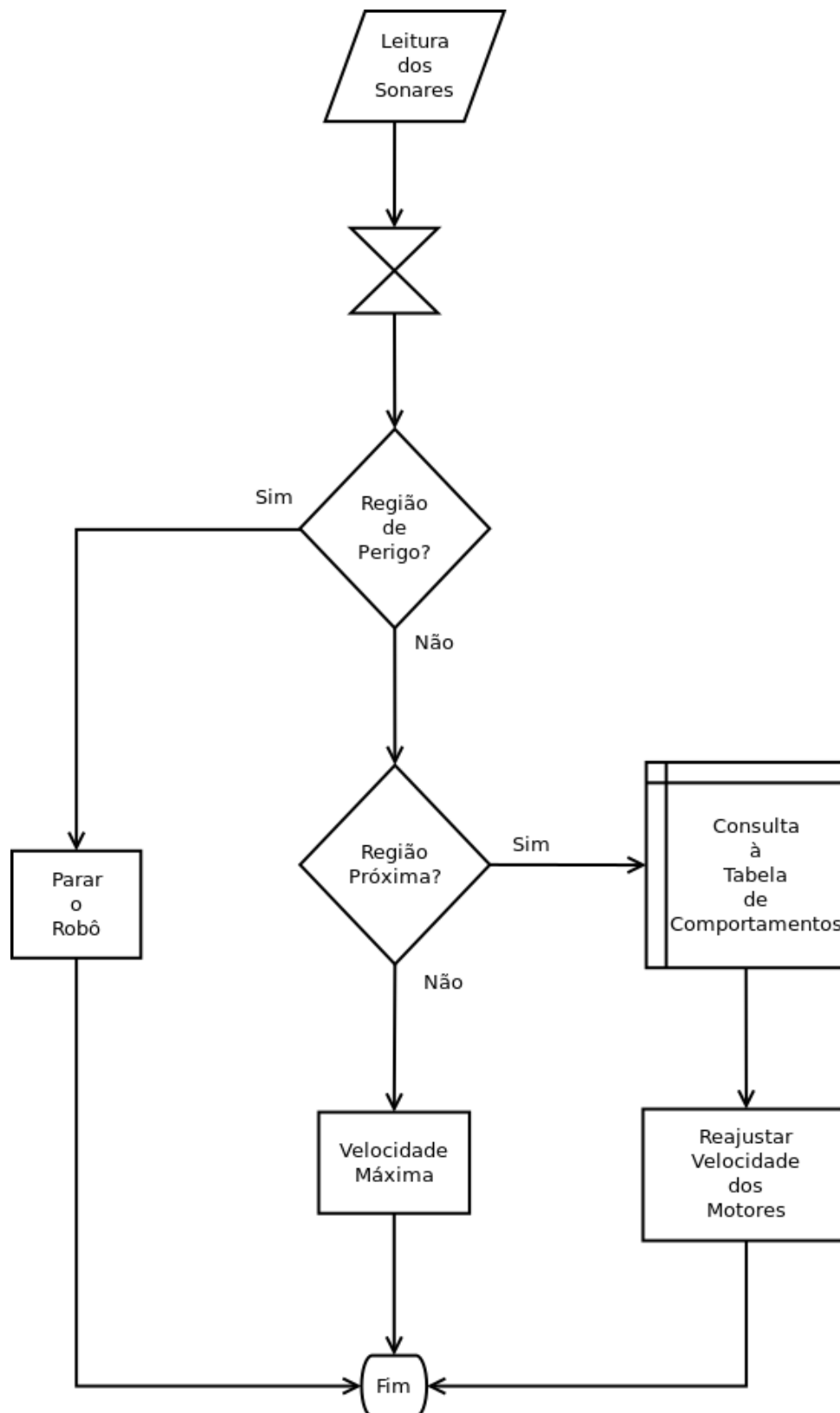


Figura 5 – Fluxograma da Rotina de Desvio de Obstáculos

ANEXO A – Anexos