

UNIVERSIDADE ESTADUAL DE MARINGÁ  
CENTRO DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA QUÍMICA  
CURSO DE ENGENHARIA ELÉTRICA

PAULO ANDRÉ CAMARGO COCCO

**PROTÓTIPO DE CARRO ROBÔ AUTÔNOMO DESVIADOR DE OBSTÁCULOS  
UTILIZANDO SENSORES ULTRASSÔNICOS E MÉTODO *VIRTUAL FORCE  
FIELD (VFF)* ADAPTADO**

TRABALHO DE CONCLUSÃO DE CURSO

MARINGÁ – PR  
2018

PAULO ANDRÉ CAMARGO COCCO

**PROTÓTIPO DE CARRO ROBÔ AUTÔNOMO DESVIADOR DE OBSTÁCULOS  
UTILIZANDO SENSORES ULTRASSÔNICOS E MÉTODO *VIRTUAL FORCE  
FIELD (VFF)* ADAPTADO**

Trabalho de Conclusão de Curso  
apresentado ao curso de Engenharia  
Elétrica da Universidade Estadual de  
Maringá, como requisito parcial para a  
obtenção do título de Engenheiro  
Eletricista.

Orientador: Prof. Dr. Rafael Krummenauer

MARINGÁ - PR  
2018

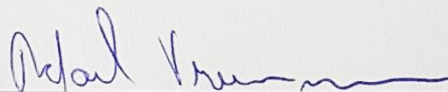
PAULO ANDRÉ CAMARGO COCCO

**PROTÓTIPO DE CARRO ROBÔ AUTÔNOMO DESVIADOR DE  
OBSTÁCULOS UTILIZANDO SENSORES ULTRASSÔNICOS E  
MÉTODO *VIRTUAL FORCE FIELD* (VFF) ADAPTADO**

Trabalho de Conclusão de Curso  
apresentado ao curso de Engenharia  
Elétrica da Universidade Estadual de  
Maringá, como requisito parcial para a  
obtenção do título de Engenheiro  
Eletricista.

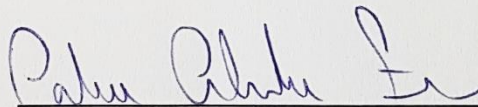
Maringá, 05 de dezembro de 2018.

BANCA EXAMINADORA



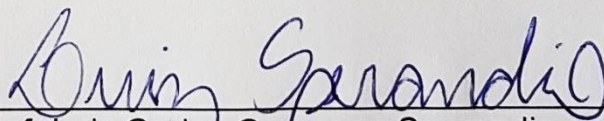
---

Prof. Dr. Rafael Krummenauer  
Universidade Estadual de Maringá – DEQ/UEM



---

Prof. Carlos Alexandre Ferri  
Universidade Estadual de Maringá – DEQ/UEM



---

Prof. Luiz Carlos Campana Sperandio  
Universidade Estadual de Maringá – DEQ/UEM

## **AGRADECIMENTOS**

Agradeço primeiramente à Deus, que sempre me deu sustento, forças e direção, em especial durante esse período de graduação. Sou grato à minha família, meus pais Italo e Keyla e minha irmã Ana Beatriz, pelos conselhos e pela paciência e compreensão nos momentos difíceis. Agradeço ao meu orientador, professor Rafael, que sempre foi solícito e esteve disposto a tirar dúvidas e dar sugestões, me ajudando e dando direcionamento na confecção deste trabalho. A todos os professores, que se dedicaram imensamente a transmitir e fornecer um conhecimento acadêmico sólido. Aos amigos do curso, que caminharam juntos durante este período de graduação, sempre buscando a excelência no aprendizado.

## RESUMO

COCCO, Paulo André Camargo. **Protótipo de Carro Robô Autônomo Desviador de Obstáculos Utilizando Sensores Ultrassônicos e Método *Virtual Force Field* (VFF) Adaptado**. 2018. 51 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Elétrica. Universidade Estadual de Maringá. Maringá, 2018.

Este trabalho trata do desenvolvimento de um protótipo de carro-robô autônomo, comandado por um microcontrolador, capaz de desviar de obstáculos, utilizando uma adaptação do método *Virtual Force Field* (VFF) para cálculo de direção do robô. O projeto consistiu no desenvolvimento de um carrinho de 3 rodas, com estrutura de triciclo (2 rodas traseiras para tração e 1 dianteira para a direção), utilizando a plataforma Arduino Nano (microcontrolador ATmega328P) para o controle, 3 sensores ultrassônicos HC-SR04 (1 na frente, 1 na posição frente-direita e 1 na frente-esquerda) para a verificação de obstáculos, 2 motores CC para providenciar o torque às rodas traseiras, 1 driver L298N para controlar os motores de corrente contínua e 1 servomotor SG5010 (preso à roda dianteira) para mudar a direção do movimento. Todo o processamento e cálculo de velocidade e direção é executado no próprio Arduino. Com o método proposto, o carrinho foi capaz de desviar dos obstáculos no ambiente de testes, buscando evitar rotas de colisão.

**Palavras-chave:** Carro-robô. Desviador de Obstáculos. Arduino Nano. Virtual Force Field. VFF.

## ABSTRACT

COCCO, Paulo André Camargo. **Autonomous Obstacle Avoidance Robot Car Prototype Using Ultrasonic Sensors and Adapted *Virtual Force Field* (VFF) Method**. 2018. 51 p. Final Paper (Graduation) – Electrical Engineering. Universidade Estadual de Maringá. Maringá, 2018.

This work deals with the development of an autonomous robot car prototype, controlled by a microcontroller, able to avoid obstacles, using an adapted *Virtual Force Field* (VFF) method, to calculate the robot's direction. The design consisted of the development of a 3-wheel car, with tricycle structure (2 rear wheels for traction and 1 front for steering), using the Arduino Nano platform (ATmega328P microcontroller) for control, 3 ultrasonic sensors HC-SR04 (1 in front, 1 in front-right and 1 in front-left) for obstacle checking, 2 DC motors to provide torque to the rear wheels, 1 L298N driver to control DC motors and 1 servomotor SG5010 (attached to the front wheel) to change the direction of movement. All processing and calculation of speed and direction is performed on the Arduino itself. With the proposed method, the car was able to avoid obstacles in the test environment, trying to avoid collision routes.

**Keywords:** Robot Car. Obstacle Avoidance. Arduino Nano. Virtual Force Field. VFF.

## LISTA DE FIGURAS

Figura 1: Tipos de rodas - (a) tradicional, (b) castor, (c) sueca, (d) esférica. ....	14
Figura 2: Robô de três rodas, sendo duas fixas, fornecendo torque. ....	14
Figura 3: Empilhadeira autoguiada, com 3 rodas. ....	15
Figura 4: Automóvel Reliant Regal Mk VI, de 1962, com 3 rodas. ....	15
Figura 5: Robô móvel autônomo, relação de percepção, decisão e ação. ....	16
Figura 6: Representação do vetor $AB$ . ....	18
Figura 7: Representação do vetor $u + v = AC$ . ....	18
Figura 8: Representação do vetor $u - v$ . ....	18
Figura 9: Representação do vetor $\alpha v$ , sendo $\alpha$ 1, 2, -1, 12 e 2, respectivamente. ....	19
Figura 10: Princípio de funcionamento do VFF. ....	21
Figura 11: Analogia da chave desligada, ligada e ligada em períodos. ....	22
Figura 12: Exemplos de sinais PWM, com <i>duty cycle</i> diferentes. ....	23
Figura 13: Funcionamento de um sensor ultrassônico. ....	24
Figura 14: Representação de sensor ultrassônico em funcionamento. ....	24
Figura 15: Representação de um servomotor. ....	25
Figura 16: Exemplo da relação PWM - servomotor. ....	26
Figura 17: Esquema de funcionamento de um motor CC. ....	27
Figura 18: Estrutura ilustrativa de uma ponte H, com chaves como transistores. ....	27
Figura 19: Ilustração de ponte H de transistores. ....	28
Figura 20: Ponte H invertendo o sentido da corrente em uma carga. ....	28
Figura 21: Arduino UNO, placa com microcontrolador ATmega328P. ....	29
Figura 22: Arduino Nano. ....	31
Figura 23: Pinagem do Arduino Nano. ....	31
Figura 24: Sensor ultrassônico HC-SR04. ....	32
Figura 25: Vista superior do robô e sensores. ....	33
Figura 26: Servomotor SG-5010. ....	34
Figura 27: Módulo ponte H L298N. ....	35
Figura 28: Portas do módulo L298N. ....	36
Figura 29: Motor com caixa de redução. ....	36
Figura 30: Esquemático do circuito do carro robô. ....	38
Figura 31: Exemplo de simulação. ....	40
Figura 32: Fluxograma de funcionamento do <i>loop</i> . ....	42
Figura 33: Fluxograma de funcionamento do procedimento ligado ao Timer2. ....	43

Figura 34: Placas de circuito impresso utilizadas na construção do robô. ....	44
Figura 35: Sensores e Arduino Nano montados nas placas. ....	44
Figura 36: Protótipo finalizado.....	45
Figura 37: Vista superior do protótipo. ....	45
Figura 38: Vista inferior do protótipo. ....	46



## SUMÁRIO

1. INTRODUÇÃO .....	11
1.1 JUSTIFICATIVA.....	12
2. FUNDAMENTAÇÃO TEÓRICA.....	13
2.1 CONCEITOS DE ROBÓTICA MÓVEL .....	13
2.2 SOMA VETORIAL E VETOR RESULTANTE .....	17
2.3 <i>VIRTUAL FORCE FIELD</i> (VFF).....	19
2.4 SINAL PWM.....	22
2.5 SENSOR ULTRASSÔNICO .....	23
2.6 SERVOMOTOR .....	25
2.7 MOTOR CC .....	26
2.8 PONTE H.....	27
2.9 MICROCONTROLADOR .....	28
3. DESENVOLVIMENTO .....	30
3.1 PROJETO ESTRUTURAL DO PROTÓTIPO.....	30
3.1.1 Arduino Nano .....	30
3.1.2 Sensor Ultrassônico HC-SR04.....	32
3.1.3 Servomotor SG-5010 .....	33
3.1.4 Módulo Ponte H L298N .....	34
3.1.5 Motor CC com Caixa de Redução.....	36
3.1.6 Esquema Elétrico .....	37
3.2 ADAPTAÇÃO DO MÉTODO <i>VIRTUAL FORCE FIELD</i> .....	39
3.3 PROGRAMAÇÃO E CONSTRUÇÃO DO PROTÓTIPO .....	41
3.3.1 Programação.....	41
3.3.2 Construção .....	44
4. RESULTADOS E DISCUSSÃO.....	47
5. CONCLUSÕES .....	48
5.1 TRABALHOS FUTUROS.....	48

## 1. INTRODUÇÃO

A robótica móvel é um tema muito relevante atualmente e, devido à sua importância e sua enorme área de aplicações, cada vez mais estudos e pesquisas estão sendo feitos acerca dela. Suas aplicações abrangem desde robôs voltados à área doméstica (aspirador de pó autônomo) e urbana (veículos semiautônomos e autônomos), até militar e de segurança civil (transporte de suprimentos, explorador de ambientes hostis, desarmador de bombas, robô de busca e resgate em ambientes inacessíveis) [1].

Um robô móvel autônomo necessita conhecer os seus arredores e, com base nisso, tomar a decisão correta de como agir para desempenhar sua função da melhor forma [2,3]. Para isso, na área da robótica móvel, são utilizados sensores para a obtenção de dados do ambiente, juntamente com algum método de controle, que faça o tratamento destes dados, a interpretação e conversão deles em comandos para o robô executar no mundo físico. Atualmente existem diversas abordagens para lidar com os dados fornecidos pelos sensores, utilizando controles PID e redes neurais, por exemplo [4,5]. Destaca-se uma linha de pesquisa que se utiliza de sensores ultrassônicos, juntamente com um método chamado *Virtual Force Field* (VFF), que se baseia na ideia de vetores imaginários exercendo forças no robô, para detecção de obstáculos e cálculo de direção e velocidade [6,7,8].

O intuito desse projeto foi construir um robô com rodas de pequena dimensão, controlado por microcontrolador e estruturado em forma de triciclo (forma bastante utilizada em aplicações com veículos autoguiados [2,9]), que, de forma autônoma, realiza medições com sensores ultrassônicos acoplados à sua estrutura e desvia de obstáculos, mudando a direção e a velocidade de seu movimento. O método implementado para escolher a direção a ser tomada é uma adaptação do já existente *Virtual Force Field* (VFF). O método implementado utiliza os valores fornecidos pelos sensores para gerar vetores de leitura espacial do ambiente e, por meio de operações entre esses vetores, é gerado um vetor resultante, com a direção desejada a ser seguida pelo carrinho. Dessa forma, o projeto alia conhecimentos de robótica móvel, eletrônica analógica e digital, programação de microcontroladores e geometria analítica.

## 1.1 JUSTIFICATIVA

Atualmente existe um grande interesse na área de robôs autônomos e semiautônomos, com o foco cada vez maior em criar robôs que tomam decisões inteligentes por conta própria e interagem com outras entidades de forma significativa, como seres humanos, animais ou até outros robôs [3].

O protótipo desenvolvido nesse projeto se trata de um robô veículo autônomo básico, que pode servir de base para futuros estudos e projetos na área de assistência de condução autônoma de veículos. O projeto também possui um caráter acadêmico, com o intuito de fornecer experiência e material de estudo para laboratórios do curso de Engenharia Elétrica da Universidade Estadual de Maringá, reunindo conhecimentos multidisciplinares vistos ao longo do curso de graduação da universidade.

## 2. FUNDAMENTAÇÃO TEÓRICA

Nesta parte do trabalho é apresentada uma revisão de literatura sobre a teoria aplicada na implementação do protótipo.

### 2.1 CONCEITOS DE ROBÓTICA MÓVEL

Um robô móvel, como o nome já sugere, necessita de um certo grau de mobilidade, para se mover pelo ambiente em que está inserido e executar suas funções com êxito. O desafio primário a ser tratado é a forma de locomoção do robô, ou seja, como deve ser o movimento dele no espaço e que mecanismos o robô deve usar para executar este movimento [2]. Existem muitas formas de locomoção e, enquanto alguns robôs procuram imitar as formas já existentes na natureza (movimento de andar, pular, correr, se arrastar), geralmente os robôs móveis se locomovem utilizando mecanismos baseados no movimento com rodas, que já é uma forma muito conhecida e utilizada na maioria dos veículos existentes [2]. Mecanismos que usam como base o movimento de pernas são mais adequados a terrenos irregulares, porém possuem um alto grau de complexidade mecânica, enquanto que mecanismos com rodas são de uma complexidade bem menor e são muito bem adequados ao terreno plano em geral [2]. Nos robôs movidos por rodas, geralmente a questão do equilíbrio não é um problema, pois quase sempre esse tipo de robô é projetado de forma que todas as rodas estão em contato com o terreno (três rodas já são o suficiente para garantir estabilidade, apesar de que robôs com duas rodas também podem ter estabilidade) [2]. Com isso, normalmente neste tipo de robô, o foco está em problemas como tração, controle e manobrabilidade.

Sobre os modelos de roda, existem basicamente quatro tipos: tradicional, castor, sueca e esférica [2]. A figura a seguir ilustra esses quatro tipos.

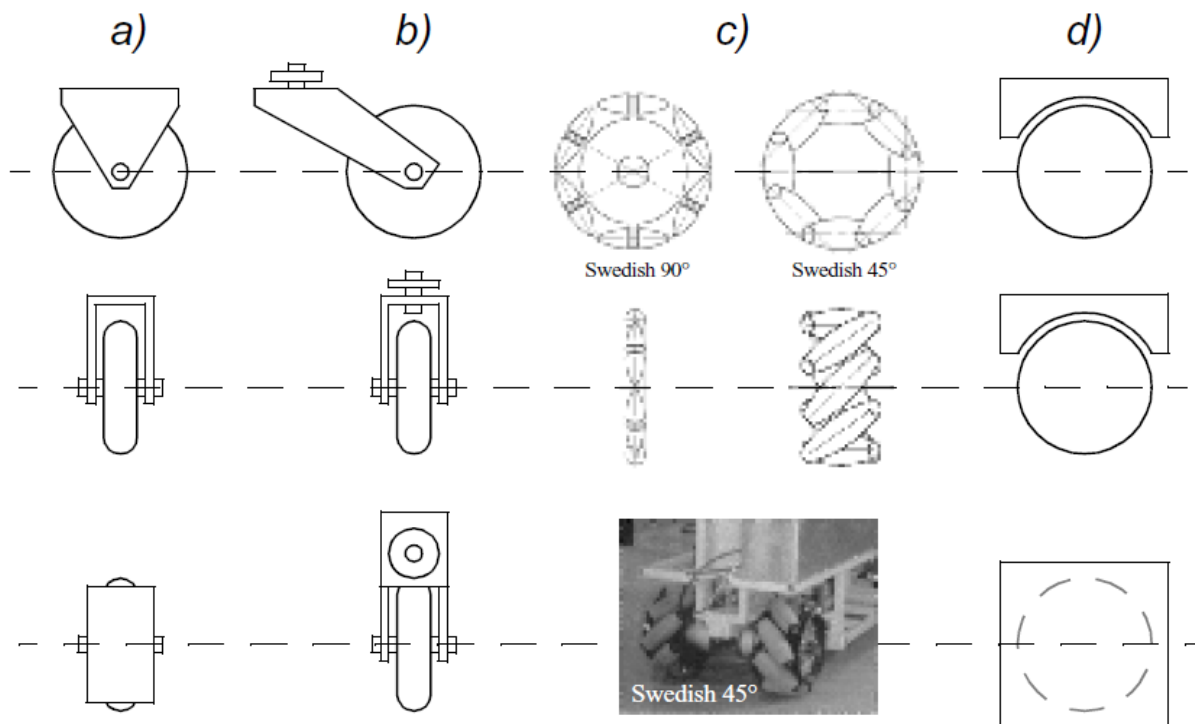


Figura 1: Tipos de rodas - (a) tradicional, (b) castor, (c) sueca, (d) esférica.

Fonte: SIEGWART (2004).

Como já foi dito, é possível ter estabilidade com no mínimo duas rodas, mas convencionalmente, para se ter estabilidade estática, são necessárias no mínimo três rodas, com o centro de massa do robô dentro do triângulo formado pelos pontos de contato das rodas com o terreno [2]. A figura abaixo mostra uma representação da estrutura de um robô móvel em forma de triciclo, com três rodas.

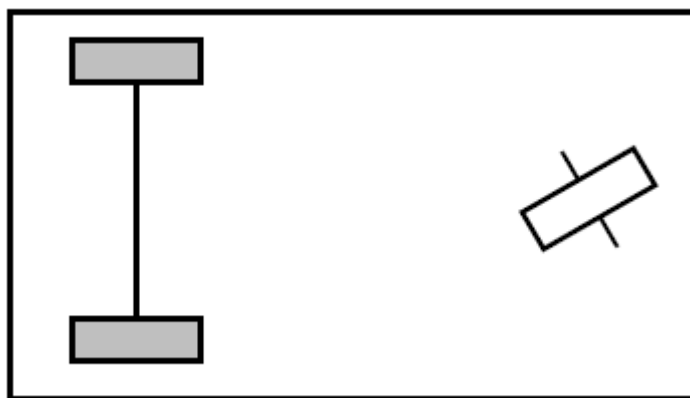


Figura 2: Robô de três rodas, sendo duas fixas, fornecendo torque.

Fonte: SIEGWART (2004).

Esta forma de posicionar as rodas no veículo é bastante utilizada, desde veículos do tipo autoguiados projetados para aplicações industriais, até mesmo automóveis [2,9]. As figuras abaixo mostram esses exemplos de aplicação.



Figura 3: Empilhadeira autoguiada, com 3 rodas.

Fonte: AUTOMATION WORLD (2016).



Figura 4: Automóvel Reliant Regal Mk VI, de 1962, com 3 rodas.

Fonte: WIKIPEDIA (2018).

Para o controle e estimativa de velocidade e direção do carro robô autônomo, existem diversos métodos e abordagens, mas em todas elas a ideia central é a mesma: o robô (agente), munido de sensores dos mais variados tipos, se encontra em um determinado ambiente e, com base em medições realizadas com os sensores (percepção) e no método de controle programado no robô, ele deve tomar uma decisão de movimento (ação), gerando comandos e controlando seus dispositivos de atuação para se mover no ambiente [1]. No caso de um carro robô, os dispositivos de atuação podem ser apenas motores CC com um sistema de redução conectados diretamente às rodas que, girando em velocidades diferentes, conseguem controlar tanto a direção quanto a velocidade do movimento, ou podem estar acompanhados de um mecanismo de direção constituído com servomotores, que mudam a angulação das rodas com relação a estrutura do robô. A figura a seguir ilustra a relação de percepção, decisão e ação do robô móvel com o ambiente.



Figura 5: Robô móvel autônomo, relação de percepção, decisão e ação.

Fonte: WOLF (2009).

Para integrar as informações obtidas por meio de sensores, interpretar os dados e gerar comandos para os atuadores, são usados diferentes métodos dentro da área de robótica móvel. Existem aplicações que utilizam a técnica de controle PID [4], que, com base nos valores medidos dos sensores, calcula um erro (formado pelos termos proporcional, integral e derivativo) e procura minimizá-lo, corrigindo então a rota do carro robô. Um outro método mais recente e complexo é a utilização de uma estrutura conhecida como rede neural [5], baseada na estrutura do cérebro humano para o aprendizado e funcionamento, para escolher a direção a ser seguida. Neste método, a rede neural é treinada com vários padrões de entrada e saída conhecidos e desejados (dados de entrada podem ser as medições dos sensores e os dados de saída, dados para controle dos atuadores) e, com base nessas amostras, a rede neural aprende os padrões de entrada e saída e, assim, gerencia o controle do robô. Também existem métodos, como o *Virtual Force Field*, que se baseiam na exploração de conceitos matemáticos, utilizando conceitos de álgebra linear e geometria analítica para determinar a velocidade e o ângulo de direção do carro robô, com base em medições realizadas com os sensores embutidos [6,7,8].

## 2.2 SOMA VETORIAL E VETOR RESULTANTE

Considerando dois pontos no espaço,  $A$  e  $B$ , um segmento orientado  $AB$  é um conjunto de pontos, que abrange  $A$ ,  $B$  e todos os pontos que passam entre eles (na reta que liga os dois pontos), que possui orientação ( $A$  é a origem, enquanto que  $B$  é a extremidade do segmento orientado). Se os segmentos orientados  $AB$  e  $CD$  tem o mesmo comprimento ( $|AB|$  = tamanho, módulo), direção (paralelos) e sentido (orientação) (ou forem ambos nulos), estes segmentos orientados são equipolentes [12,13].

Formalmente, um vetor  $\overrightarrow{AB}$  é o conjunto de todos os segmentos orientados equipolentes a  $AB$ , e cada um desses segmentos orientados é um representante do vetor  $\overrightarrow{AB}$ . A figura abaixo mostra a representação deste vetor.



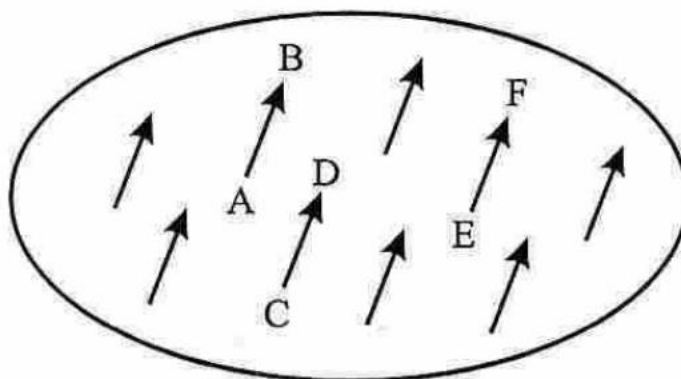


Figura 6: Representação do vetor  $\overrightarrow{AB}$ .

Fonte: WATANABE (1999).

Considerando dois vetores,  $\vec{u} = \overrightarrow{AB}$  e  $\vec{v} = \overrightarrow{BC}$ , a soma  $\vec{u} + \vec{v}$  resulta em  $\vec{u} + \vec{v} = \overrightarrow{AC}$ , que é o vetor resultante da soma. Sabendo que o vetor  $-\vec{v}$  é igual ao vetor  $\vec{v}$ , no sentido oposto, a subtração resulta em  $\vec{u} - \vec{v} = \vec{u} + (-\vec{v})$ . Considerando  $\alpha$  um número real, a multiplicação é  $\alpha \cdot \vec{v} = \alpha \vec{v}$  [12,13]. As figuras a seguir ilustram as operações de soma, subtração e multiplicação por escalar ( $\alpha$ ).

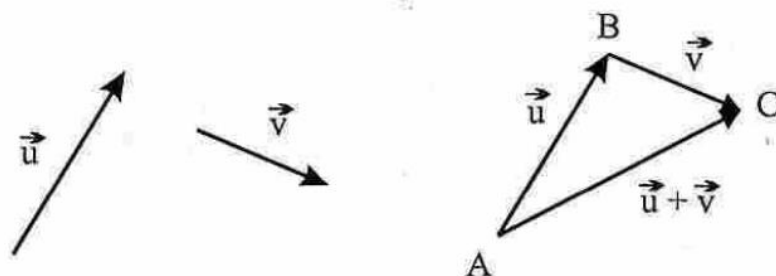


Figura 7: Representação do vetor  $\vec{u} + \vec{v} = \overrightarrow{AC}$ .

Fonte: WATANABE (1999).

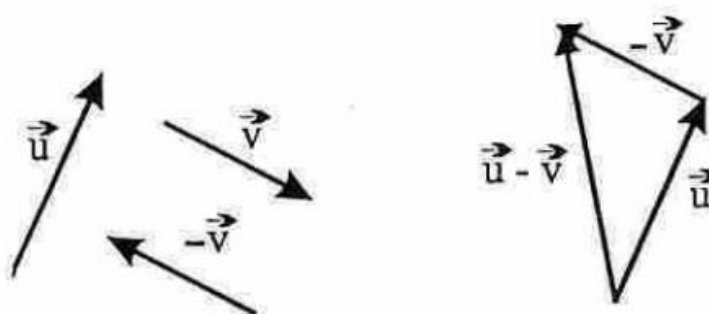


Figura 8: Representação do vetor  $\vec{u} - \vec{v}$ .

Fonte: WATANABE (1999).

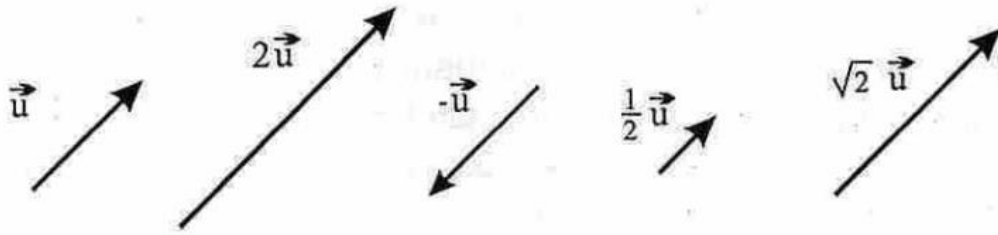


Figura 9: Representação do vetor  $\alpha \vec{v}$ , sendo  $\alpha$  1, 2, -1,  $\frac{1}{2}$  e  $\sqrt{2}$ , respectivamente.

Fonte: WATANABE (1999).

Em coordenadas cartesianas, considerando  $\vec{u} = (a, b)$  e  $\vec{v} = (c, d)$ , a soma é  $\vec{u} + \vec{v} = (a + c, b + d)$  e a subtração é  $\vec{u} - \vec{v} = (a - c, b - d)$ . O produto do número real  $\alpha$  com  $\vec{v}$  é  $\alpha \vec{v} = (\alpha c, \alpha d)$  [12,13,14]. Estes conceitos e operações servem de base para o método de tomada de decisão utilizado neste projeto.

### 2.3 VIRTUAL FORCE FIELD (VFF)

O *Virtual Force Field (VFF)* é um método de desvio de obstáculos, empregado dentro da robótica móvel. Seu princípio de funcionamento se baseia na ideia de que os obstáculos atuam no robô móvel exercendo forças de repulsão, cuja magnitude é inversamente proporcional à distância até o robô, enquanto existe uma força de atração na direção desejada de locomoção. O robô se orienta pelo espaço utilizando sensores ultrassônicos [6,7,8].

Neste método os obstáculos são representados da seguinte forma: o ambiente que o robô está inserido é representado numa espécie de grade espacial de certezas (*certainty grid*), em que o espaço é dividido em células, que contém certos valores de certeza, que indicam o grau de convicção de que existe algum obstáculo dentro da célula (quanto maior o valor, maior a certeza) [7,8].

O robô conhece sua posição nessa grade e, à medida que o robô se move, ele realiza medições com os sensores ultrassônicos e as projeta nas células correspondentes na *certainty grid*, formando uma espécie de modelo do ambiente. Se um obstáculo retorna o eco do sinal emitido pelo sensor, a célula correspondente a posição dele é incrementada na grade. Dessa forma, este método considera as medições anteriores, não só as medições realizadas no instante.

Simultaneamente ele examina uma área menor da grade, que equivale ao seu redor. Dentro dessa região examinada, cada célula preenchida aplica uma força de repulsão no robô, empurrando o robô para longe da célula, em que a magnitude da força é diretamente proporcional ao valor da célula e inversamente proporcional à distância entre a célula e o robô (relação não-linear).

A força de repulsão de cada célula pode ser calculada pela seguinte fórmula:

$$F(i, j) = \frac{F_{cr}C(i, j)}{d^2(i, j)} \left[ \frac{x_i - x_0}{d(i, j)} \hat{x} + \frac{y_i - y_0}{d(i, j)} \hat{y} \right] \quad (1)$$

em que

- $F_{cr}$  é uma constante de força de repulsão,
- $d(i, j)$  é a distância entre a célula  $(i, j)$  e o robô,
- $x_0, y_0$  são as coordenadas do robô,
- $x_i, y_i$  são as coordenadas da célula.

Nota-se que as forças de repulsão são inversamente proporcionais à distância ao quadrado.

A força de repulsão resultante ( $F_r$ ) no robô é a soma vetorial de todas as forças individuais provocadas pelas células. Existe também uma força de atração ( $F_t$ ) que atua constantemente sobre o robô, que aponta para a direção desejada de locomoção e tem magnitude constante:

$$F_t = F_{ct} \left[ \frac{x_t - x_0}{d(t)} \hat{x} + \frac{y_t - y_0}{d(t)} \hat{y} \right] \quad (2)$$

em que

- $F_{ct}$  é uma constante de força de atração para o alvo,
- $d(t)$  é a distância entre o robô e o alvo,
- $x_t, y_t$  são as coordenadas do alvo.

A soma vetorial entre a força de atração ( $F_t$ ) e a força de repulsão resultante ( $F_r$ ) gera um vetor  $R$ , cuja direção  $\delta = R/|R|$  é usada como referência na tomada de uma direção por parte do carro robô.

A figura a seguir ilustra o robô inserido no ambiente, a divisão espacial na *certainty grid* e os vetores gerados no processo de tomada de decisão.

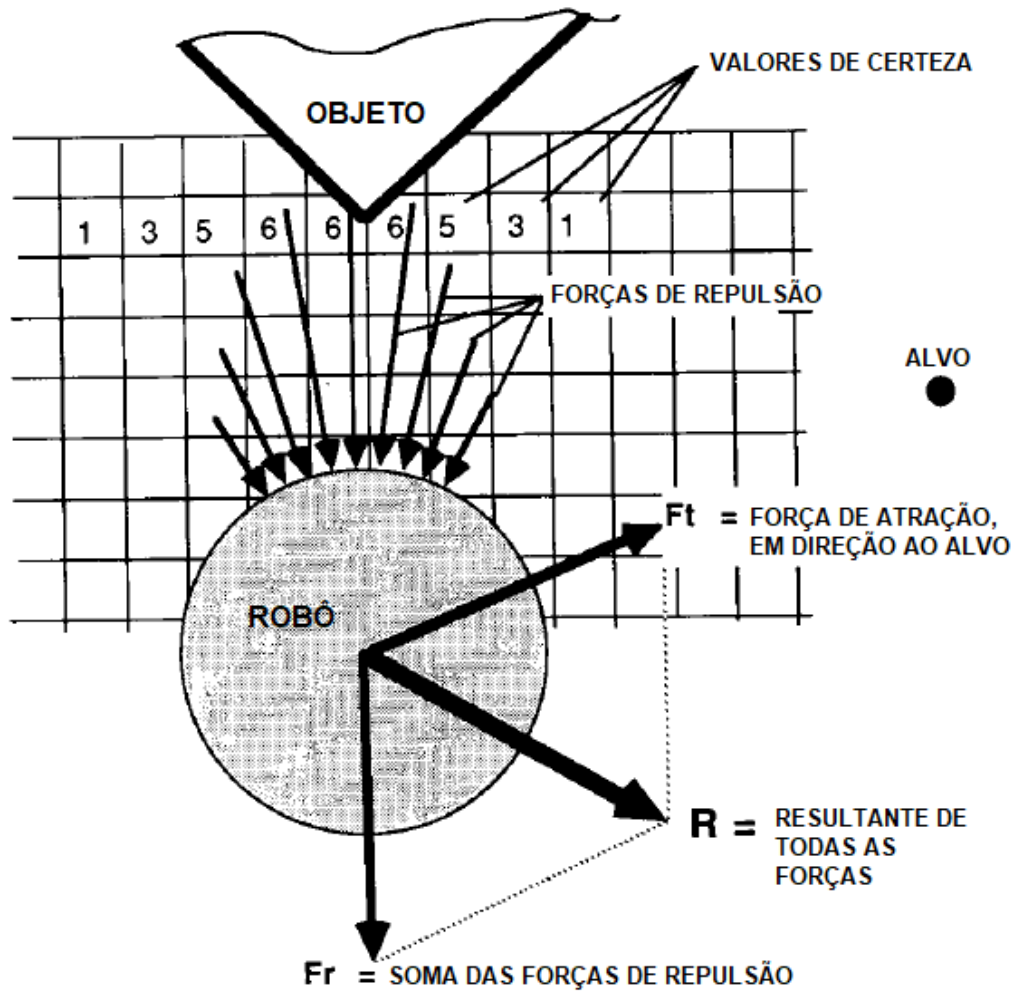


Figura 10: Princípio de funcionamento do VFF.

Fonte: Adaptado de BORENSTEIN (1989).

O robô se utiliza de uma taxa de direção (*steering-rate*)  $\Omega$ , que pode ser calculada pela seguinte fórmula:

$$\Omega = K_s[\theta(-)\delta] \quad (3)$$

em que

$K_s$  é uma constante proporcional para direção (em  $s^{-1}$ ),

$\theta$  é o atual ângulo de direção (em graus),

$(-)$  é um operador que retorna a menor distância entre dois ângulos.

Este método não exige que o robô pare de se mover para adquirir os dados do ambiente por meio dos sensores, de forma que ele consegue lidar com os obstáculos enquanto se movimenta.

A locomoção do robô é caracterizada pelos ajustes da direção e de velocidade. A seção a seguir trata do controle da velocidade.

## 2.4 SINAL PWM

O PWM é uma técnica empregada para fazer com que um sinal digital se comporte como um sinal analógico, por meio de modulações na largura do pulso de uma onda quadrada [15]. O princípio de funcionamento é o de uma chave, ligando e desligando em períodos: se a chave fica sempre desligada, a tensão fornecida é nula, se fica sempre ligada, é máxima, se liga e desliga em tempos controlados, a tensão fornecida é um valor entre 0 e a tensão máxima (por exemplo, se fica ligada 60% do período e desligada nos outros 40%, a tensão fornecida é 60% da tensão máxima), como ilustra a figura a seguir.

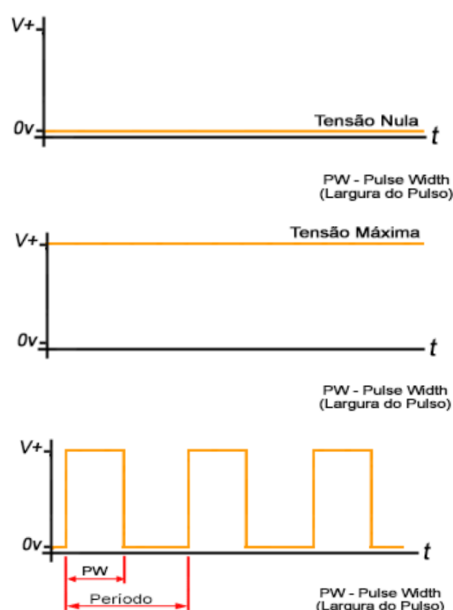


Figura 11: Analogia da chave desligada, ligada e ligada em períodos.

Fonte: MECAWEB (2008-2016).

O que define o valor de tensão fornecido (porcentagem da tensão máxima possível) é o *Duty Cycle* (ciclo de trabalho), que pode ser calculado da seguinte forma [19]:

$$Duty\ Cycle = 100 * \frac{Largura\ do\ Pulso}{Período} \quad (4)$$

em que *Duty Cycle* é o próprio ciclo de trabalho (%) e *Largura do Pulso* e *Período* são o tempo que o sinal fica ligado e o tempo de 1 ciclo da onda, respectivamente (ambos medidos em unidades de tempo). A figura abaixo mostra exemplos de sinais PWM com diferentes ciclos de trabalho.

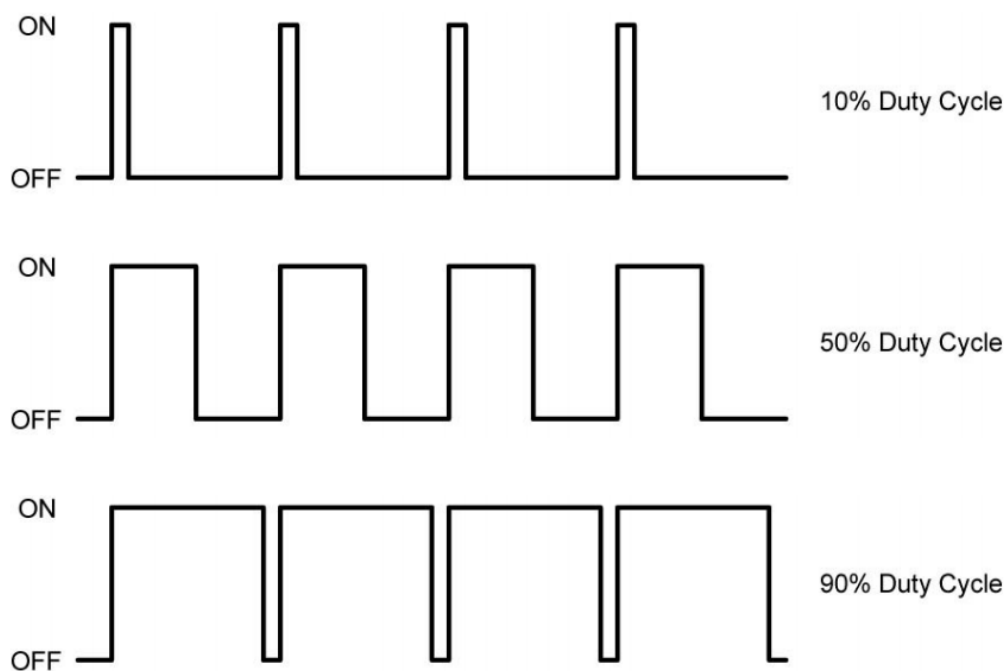


Figura 12: Exemplos de sinais PWM, com *duty cycle* diferentes.

Fonte: MECAWEB (2008-2016).

## 2.5 SENSOR ULTRASSÔNICO

O sensor ultrassônico é um sensor que se utiliza de ondas ou pulsos de acústicos na frequência superiores à audível pelo ouvido humano (acima de 20 kHz), chamados de ultrassom, para fazer medições de distância. Desta forma, este sensor se caracteriza por operar por um tipo de radiação não sujeita a interferência eletromagnética e totalmente limpa [16]. Ele possui a capacidade de detectar objetos em distâncias que variam entre milímetros até vários metros, podendo ser usado para detectar os mais variados tipos de objetos e substâncias. Além da aplicação de medição de distância, o sensor ultrassônico pode ser usado para detectar a passagem de objetos numa linha de montagem, detectar a presença de pessoas ou ainda de substâncias em diversos estados num reservatório permitindo a medida de seu nível [16].

O princípio de funcionamento do sensor ultrassônico é o mesmo do sonar, usado por animais como morcegos para detectar presas e objetos [16]. O sensor possui duas partes, uma emissora e uma receptora. Para realizar a medição de distância do sensor até algum obstáculo, primeiramente ele emite um pequeno pulso sonoro de alta frequência, que se propaga na velocidade do som no meio em que está inserido. Quando este pulso atinge algum objeto, um sinal de eco será refletido para o sensor [16,17]. As figuras abaixo ilustram esse princípio de funcionamento.

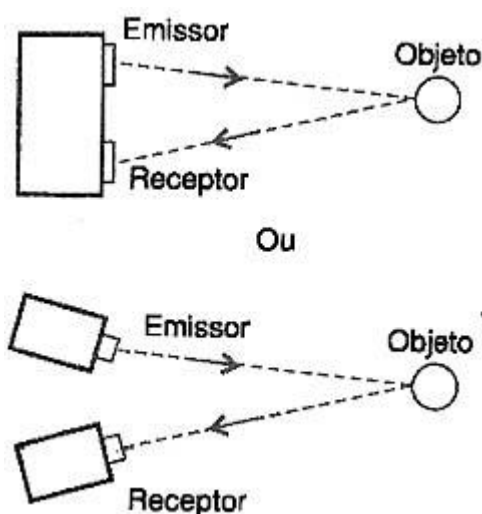


Figura 13: Funcionamento de um sensor ultrassônico.

Fonte: INSTITUTO NEWTON C. BRAGA (2012).

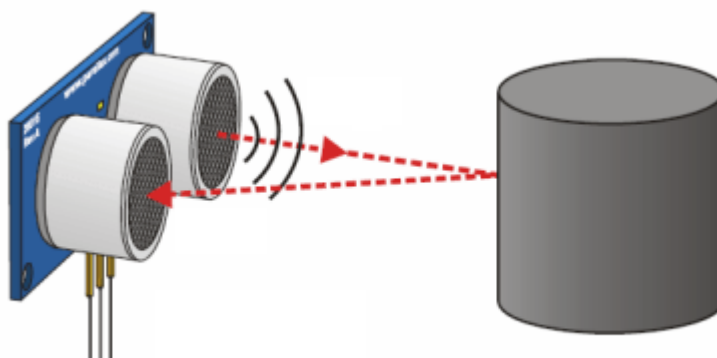


Figura 14: Representação de sensor ultrassônico em funcionamento.

Fonte: VIDA DE SILÍCIO (2017).

Conhecendo a velocidade do som no meio em questão ( $V_{som}$ ) e o tempo entre o sensor emitir e receber o sinal sonoro ( $T$ ), é possível calcular a distância entre o sensor e o objeto detectado ( $\Delta S$ ) [16,17]. Considerando a velocidade do som no ar aproximadamente  $340 \text{ m/s}$ , tem-se que:

$$V_{som} = 340 \frac{\text{m}}{\text{s}} = \frac{\Delta S}{\Delta T} \quad (5)$$

Sabendo que  $T$  é o tempo da onda sonora ser emitida, refletir no objeto e voltar até o sensor, então  $\Delta T = T/2$ , e tem-se que:

$$\frac{\Delta S}{T/2} = 340 \quad (6)$$

Então a distância do sensor até o obstáculo é, em metros:

$$\Delta S = 340 * \frac{T}{2} \quad (7)$$

## 2.6 SERVOMOTOR

Um dos dispositivos mais importantes no controle de robôs e automatismos é o servomotor, que, de forma simplificada, é um motor que permite controlar sua posição angular através de um sinal de entrada [18,19].

O servomotor é constituído basicamente de um circuito de controle e um motor com redução, que movimenta uma alavanca/braço. A partir de um sinal aplicado à entrada deste dispositivo, é possível movimentar o braço até uma posição determinada de modo proporcional. A figura abaixo mostra uma representação de um servomotor.

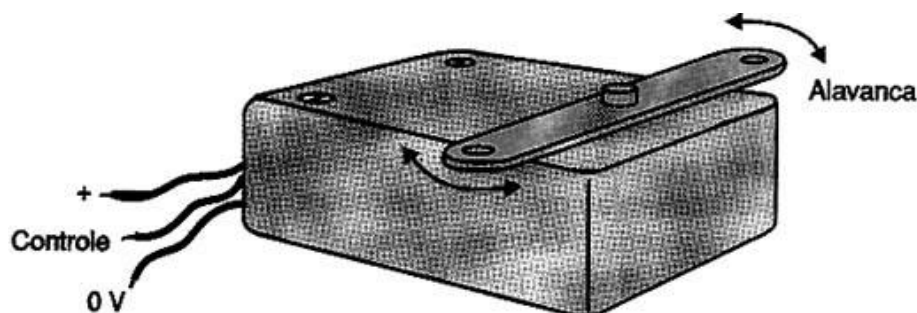


Figura 15: Representação de um servomotor.  
Fonte: INSTITUTO NEWTON C. BRAGA (1999).



O sinal de controle pode ser na forma PWM (Modulação por Largura de Pulso) [19]. Utilizando um sinal PWM como sinal de entrada de um servomotor, o ângulo do braço é proporcional ao *duty cycle* do sinal. Na figura a seguir está exemplificada a relação do sinal PWM com o ângulo da alavanca do servomotor.

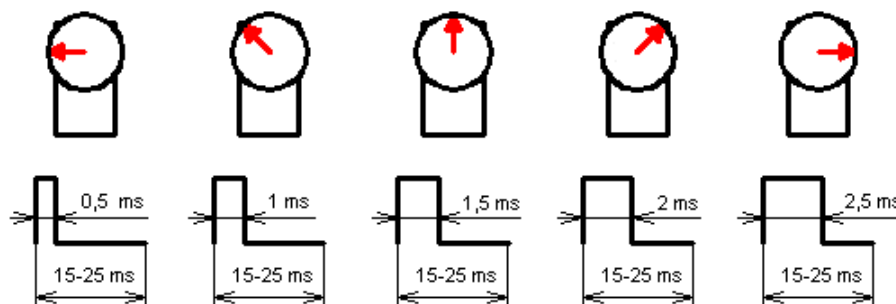


Figura 16: Exemplo da relação PWM - servomotor.

Fonte: ROBÓTICA FÁCIL (2016-2018).

## 2.7 MOTOR CC

Um motor CC (corrente contínua) nada mais é do que uma máquina elétrica rotativa alimentada por corrente contínua, sendo esta alimentação proveniente de uma bateria ou qualquer outro tipo de fonte CC. O motor CC é composto por duas partes: uma parte fixa, denominada estator, e uma parte móvel, chamada rotor, esta acoplada a um eixo, constituindo a parte girante do motor [21]. A comutação (a passagem de corrente elétrica entre o rotor e o estator) pode ser através de escovas ou sem escovas.

O princípio básico de funcionamento do motor CC pode ser descrito da seguinte forma [21,22]: se um condutor conduzindo corrente elétrica é colocado em um campo magnético, este condutor experimenta uma força mecânica, gerando um torque (ortogonal à direção da corrente e ao campo magnético, pela regra da mão esquerda de Fleming) e provocando o giro do eixo do motor. Então se o rotor, composto por espiras (fios enrolados), for alimentado pelo comutador e se o estator for alimentado de forma que trabalhe como ímã, obtém-se o torque e movimento da parte girante do motor. A figura a seguir ilustra esse princípio de funcionamento.

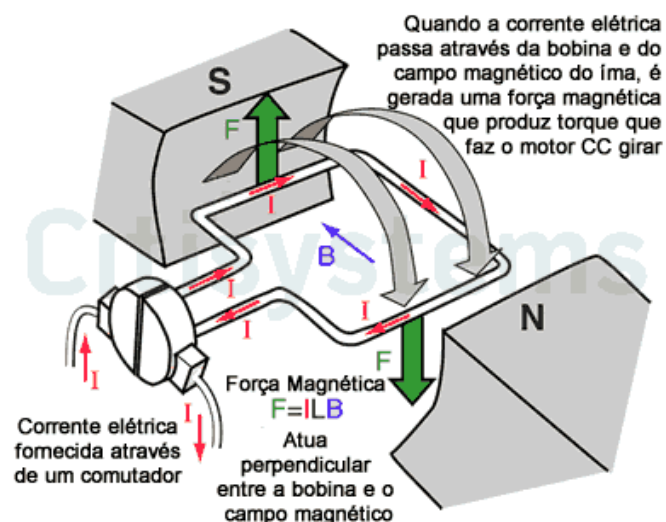


Figura 17: Esquema de funcionamento de um motor CC.

Fonte: CITISYSTEMS (2018).

É possível controlar a velocidade do motor, de acordo com a tensão aplicada nele [21]. A relação da velocidade rotacional do rotor é diretamente proporcional à tensão aplicada.

## 2.8 PONTE H

Uma ponte H é um circuito que permite realizar a inversão de sentido da corrente que flui através de uma carga. É muito utilizada para controlar a direção de rotação de um motor CC [23]. A ponte H é composta por 4 transistores, que funcionam como chaves, dispostas de forma que, alternando elas dois a dois, a tensão entre os dois polos da carga inverte e, dessa forma, a corrente que passa pela carga muda de sentido [23,24]. As figuras a seguir ilustram o conceito de ponte H e seu funcionamento.

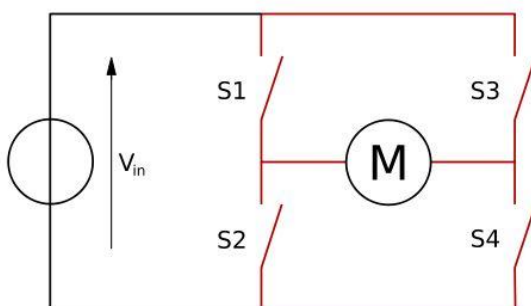


Figura 18: Estrutura ilustrativa de uma ponte H, com chaves como transistores.

Fonte: CITISYSTEMS (2018).

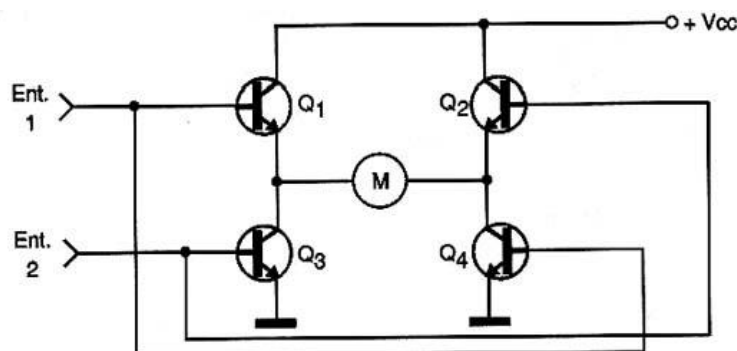


Figura 19: Ilustração de ponte H de transistores.

Fonte: INSTITUTO NEWTON C. BRAGA (1999).

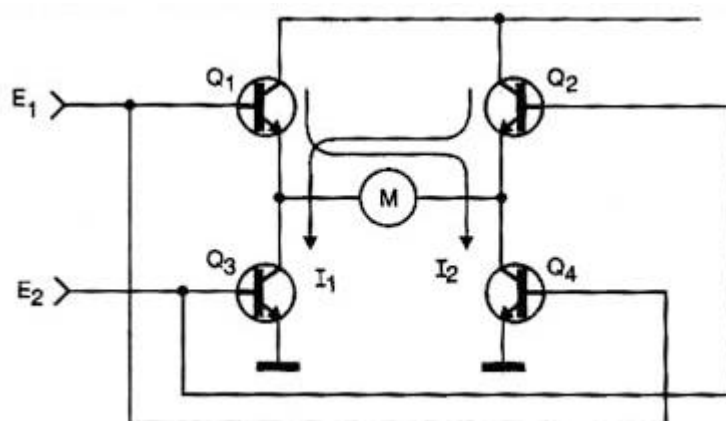


Figura 20: Ponte H invertendo o sentido da corrente em uma carga.

Fonte: INSTITUTO NEWTON C. BRAGA (1999).

## 2.9 MICROCONTROLADOR

Um microcontrolador é um tipo de circuito integrado, que pode ser programado para realizar diferentes funções [25]. Os microcontroladores são utilizados para controlar circuitos a partir de informações passadas a ele, por meio da programação, cuja linguagem e forma de programar varia de microcontrolador para microcontrolador.

Antigamente as montagens eletrônicas eram feitas baseadas em componentes individuais (discretos), ou em diversos circuitos integrados, que exerciam funções únicas e separadas. Com o advento dos microcontroladores, foi possível tornar os projetos muito mais poderosos e compactos [25].

Para os microcontroladores poderem ser usados e programados, necessitam ser montados em placas apropriadas e sua programação é feita em conexão com um computador. Atualmente existem placas prontas, destinadas principalmente à

prototipagem ou aplicações isoladas (que não são industrializadas), que já possuem todos os recursos para a entrada de programação, através de cabo, e conectores para ligar aos circuitos controlados, sensores, outros dispositivos e fonte de alimentação [25]. A figura abaixo mostra um exemplo de placa pronta, o Arduino UNO, que possui o microcontrolador ATmega328P.



Figura 21: Arduino UNO, placa com microcontrolador ATmega328P.

Fonte: FILIPEFLOP (2017).

Um microcontrolador é constituído por cinco blocos [25]:

- *CPU*: é o “cérebro” do microcontrolador, é capaz de realizar operações lógicas com sinais digitais. O processamento do microcontrolador é determinado pelo *clock* (ritmo de operação do microcontrolador, um oscilador de *clock* determina a velocidade com que o microcontrolador opera);
- *Memória de programa*: armazena as instruções (programa) que diz o que o microcontrolador deve fazer. O tamanho dessas memórias varia de acordo com a finalidade e tipo do microcontrolador. São do tipo EEPROM ou Flash, que retém a informação mesmo quando a alimentação é desligada;
- *Memória de dados*: armazena as informações que o microcontrolador colhe durante seu funcionamento. Essa memória não retém informação quando o microcontrolador é desligado;
- *Timers*: realizam contagens precisas de tempo. Permitem ciclos de controle muito mais lentos que o *clock*, que são usados em diversas aplicações no microcontrolador;
- *Portas I/O*: interface do microcontrolador com o mundo exterior. São acessadas através de pinos no circuito integrado.

### 3. DESENVOLVIMENTO

Esta parte do trabalho foi dividida em três etapas:

- Projeto da estrutura do robô protótipo;
- Adaptação do método *VFF* para o robô proposto;
- Programação e construção do protótipo.

#### 3.1 PROJETO ESTRUTURAL DO PROTÓTIPO

O protótipo proposto neste projeto é um carro robô autônomo estruturado na forma de triciclo, com a roda de direção frontal do tipo tradicional, e com torque sendo fornecido pelas duas rodas traseiras. A parte elétrica é alimentada por um conjunto de seis pilhas de 1,5V cada (totalizando 9V em série) e é constituída por um Arduino Nano, três sensores ultrassônicos HC-SR04 (para realizar a medição de distância até os obstáculos), um servomotor SG-5010 (preso à roda dianteira de direção, para controlar a direção do carro), um módulo ponte H L298N e dois motores CC com caixa de redução (anexos às rodas traseiras).

##### 3.1.1 Arduino Nano

O Arduino é uma plataforma *open-source* de prototipagem eletrônica com hardware e software fácil de utilizar [27]. Existem diversos tipos de sensores e atuadores compatíveis com a plataforma, disponibilizados em módulos que podem ser facilmente ligados ao Arduino. Por se tratar de uma plataforma de fácil interação e por possuir um custo baixo, é comum o Arduino ser utilizado no meio acadêmico.

O Arduino pode ser encontrado em várias versões. A versão Nano contém o microcontrolador ATmega328P e todos os outros componentes necessários para trabalhar com ele, sendo necessária apenas uma fonte de alimentação, que pode ser via cabo USB ou ligada diretamente nos pinos correspondentes. Ele possui um cristal de quartzo de 16MHz usado como *clock* e não é necessário o uso de um gravador externo, já que ele possui um *bootloader* gravado no microcontrolador. A figura a seguir mostra o Arduino Nano utilizado neste projeto.

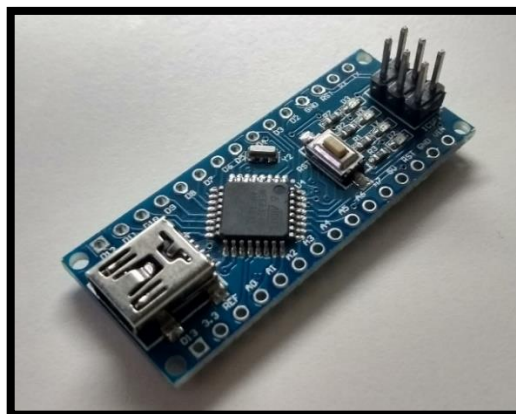


Figura 22: Arduino Nano.

Fonte: Autoria própria (2018).

A placa opera em 5 V, possui 8 pinos analógicos de entrada e 22 pinos digitais de entrada e saída (6 são PWM) [28,29]. A figura abaixo mostra a pinagem da placa.

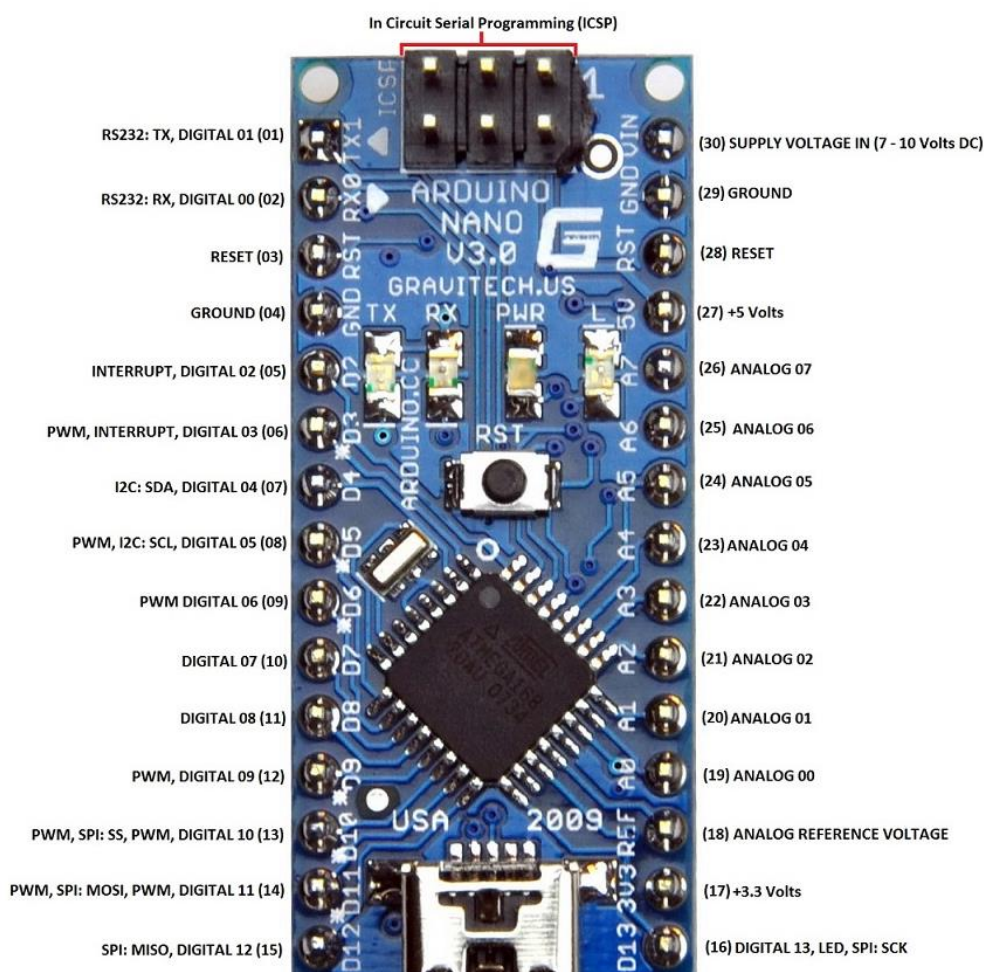


Figura 23: Pinagem do Arduino Nano.

Fonte: CIRCUITS TODAY (2018).

A programação do Arduino é feita através do *software* livre Arduino IDE (*Integrated Development Environment*), em uma linguagem baseada na linguagem C/C++.

O ATmega328P possui 3 temporizadores, sendo dois de 8 bits e um de 16 bits. Esses temporizadores são utilizados para contagem de eventos externos, temporização, geração de sinais PWM, interrupções periódicas e medidas de intervalos de pulsos. Cada temporizador tem suas características próprias e são usados para fins específicos. É possível programar esses temporizadores para que executem outras funções em vez das funções pré-programadas neles [30,31].

O Timer0 é um temporizador de 8 bits e é utilizado para funções como *delay()*, *millis()* e *micros()*, como essas funções são bastante utilizadas, é recomendável evitar o uso deste temporizador para outras funções. O Timer1 é um temporizador de 16 bits e é utilizado para controle de servos motores, enquanto que o Timer2 é um temporizador de 8 bits e é utilizado para a função *tone()* [31].

### 3.1.2 Sensor Ultrassônico HC-SR04

O HC-SR04 é um módulo sensor ultrassônico medidor de distâncias e obstáculos. Sua região de detecção é de 2 cm a 4 m à frente do sensor, com precisão de 3 mm. O módulo inclui os transmissores e receptores ultrassônicos, além de um circuito de controle [32].

Esse módulo trabalha numa tensão de 5 V, corrente de 15 mA, frequência de 40 kHz (frequência do sinal ultrassônico), com ângulo de medição de 15°. A figura abaixo mostra um dos sensores ultrassônicos utilizados neste projeto.

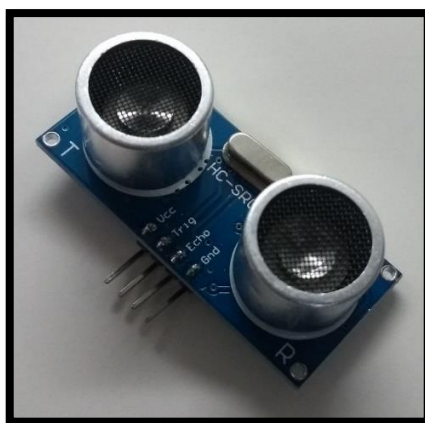


Figura 24: Sensor ultrassônico HC-SR04.

Fonte: Autoria própria (2018).

O HC-SR04 possui quatro pinos: *Vcc* e *Gnd* (alimentação do módulo) e *Trig* (*trigger*) e *Echo*.

No robô proposto, são utilizados 3 sensores ultrassônicos HC-SR04, posicionados cada um com uma determinada angulação no carro (um posicionado na direita-frente, um voltado totalmente para a frente e o último posicionado na esquerda-frente), para detecção de obstáculo vindos de diferentes direções.

A angulação dos sensores da direita-frente, frente e esquerda-frente deve ser de  $60^\circ$ ,  $90^\circ$  e  $120^\circ$ , respectivamente, com relação ao eixo horizontal, analisando o robô de uma perspectiva superior, com sua frente voltada pra cima, como ilustra a figura abaixo.

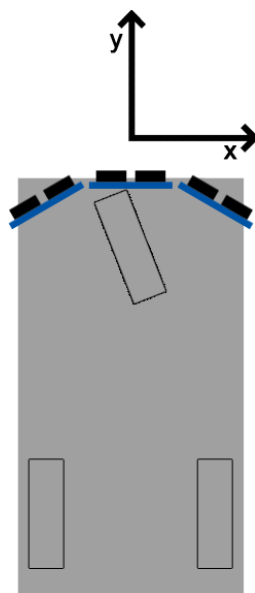


Figura 25: Vista superior do robô e sensores.

Fonte: Autoria própria (2018).

### 3.1.3 Servomotor SG-5010

O Servo Tower Pro SG-5010 é um servo de alta qualidade e excelente para as aplicações em projetos mecatrônicos [33].

Sua tensão de operação é entre 4,8 V e 6 V e possui uma faixa de rotação de  $180^\circ$ . O torque máximo varia entre 8 kg.cm (alimentado com 4,8 V) e 11 kg.cm (alimentado com 6 V). A velocidade de giro varia entre 0,17 seg/ $60^\circ$  (4,8 V) e 0,14 seg/ $60^\circ$  (6 V). A imagem a seguir mostra um servomotor SG-5010, idêntico ao utilizado neste projeto.





Figura 26: Servomotor SG-5010.

Fonte: FILIPEFLOP (2017).

Possui 3 fios, sendo que dois (vermelho e marrom) são para alimentação (positivo e *ground*, respectivamente) e um (laranja) é a entrada para o sinal de controle (PWM).

Neste projeto, a roda de direção do robô proposto deve ter uma amplitude de movimento de 90° total, 45° para cada lado, partindo da posição central (90°). Com isso, o servomotor só pode se movimentar de 45° a 135°.

#### 3.1.4 Módulo Ponte H L298N

O módulo de ponte H L298N é um *driver* de potência e é muito usado em aplicações de robótica que utilizam motores CC, que demandam correntes maiores do que o microcontrolador pode fornecer. Esse módulo possui o CI L298N, que consiste de duas pontes H (pontes A e B), para ser usado com dois motores. Por se tratar de pontes H, é possível inverter o sentido de rotação dos motores controlados [34,35]. A figura a seguir mostra o *driver* L298N utilizado neste projeto.

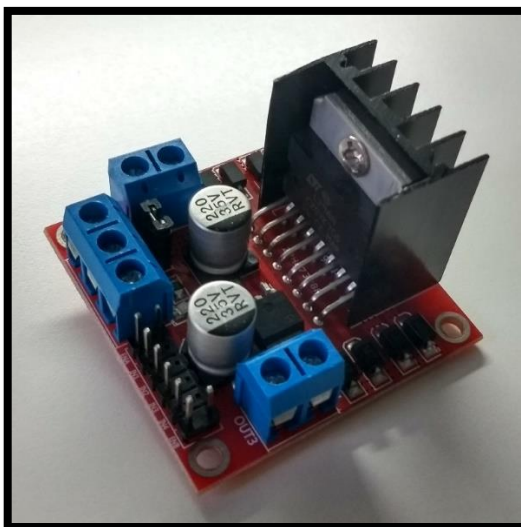


Figura 27: Módulo ponte H L298N.

Fonte: Autoria própria (2018).

O módulo possui as seguintes portas de entradas e saídas:

- *Motor A (Out1 e Out2)*: Saídas da ponte A, é onde um motor é ligado;
  - *Motor B (Out3 e Out4)*: Saídas da ponte B, é onde o outro motor é ligado;
  - *6-35 V*: Porta para alimentação da placa (6-35 V);
  - *Ativa 5 V*: Pode ser jumpeado, quando isso acontece, a placa utiliza um regulador de tensão integrado para fornecer 5 V na porta 5 V, para alimentar outros componentes como microcontrolador e sensores;
  - *5 V*: Saída de 5 V caso Ativa 5 V esteja jumpeado. Caso não exista uma alimentação de 6-35 V para a placa, ela pode ser alimentada com 5 V por essa porta;
  - *Ativa MA (Enable A)*: Pode ser jumpeado, quando isso acontece, o motor gira na velocidade máxima. Se o jumper for removido, é possível controlar a velocidade do motor A aplicando uma tensão entre 0 e 5 V;
  - *Ativa MB (Enable B)*: Mesma coisa do Ativa MA, mas para o motor B;
  - *IN1 e IN2*: usados para controlar o sentido de rotação do motor A;
  - *IN3 e IN4*: usados para controlar o sentido de rotação do motor B.
- A figura logo a seguir indica as entradas e saídas diretamente na placa.

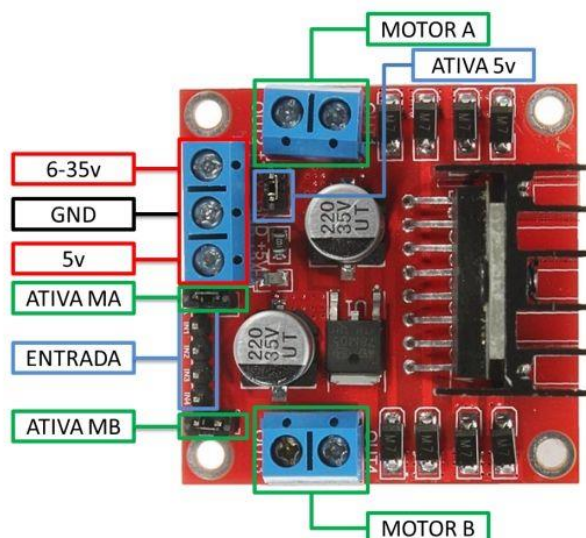


Figura 28: Portas do módulo L298N.

Fonte: ARDUINO PORTUGAL (2017).

### 3.1.5 Motor CC com Caixa de Redução

Neste projeto foram utilizados dois motores CC de ímã permanente, de 3-6 V, muito usados em aplicações de robótica. Estes motores já vêm acompanhados de caixa de redução, que tornam eles muito adequados ao projeto. A figura abaixo mostra um dos motores utilizados nesse projeto.

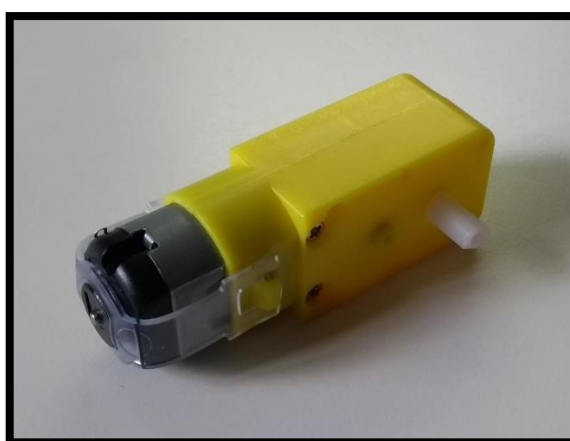


Figura 29: Motor com caixa de redução.

Fonte: Autoria própria (2018).

A redução é na razão de 1:48 e o torque varia de 0,35 Kg.cm (alimentado com 3 V) a 0,80 Kg.cm (6 V) [36].

### 3.1.6 Esquema Elétrico

O circuito funciona basicamente da seguinte forma: a alimentação é feita por pilhas em série, que alimentam o *driver* L298N. O *driver* alimenta os motores e também providencia alimentação para o restante do circuito, que é constituído pela parte de controle (Arduino Nano), direção (servomotor SG-5010) e detecção (sensores ultrassônicos HC-SR04).

O circuito trabalha com os níveis de tensão 0 V (*ground*), 5 V e 9 V (fornecido por 6 pilhas de 1,5 V em série). A tensão de 9 V é utilizada para alimentar o *driver* L298N, que alimenta os motores. O *driver* utiliza o regulador de tensão de 5 V interno para disponibilizar esta tensão (pelo pino 5 V) para o restante do circuito, que trabalha com 5 V. O *ground* é comum a todos os componentes.

Os pinos utilizados no Arduino Nano são ligados da seguinte forma:

- D2: ligado ao pino *trigger* do sensor da esquerda;
- D3: ligado ao pino *echo* do sensor da esquerda;
- D4: ligado ao pino *trigger* do sensor da frente;
- D5: ligado ao pino *echo* do sensor da frente;
- D6: ligado aos pinos *Enable A* e *Enable B* do L298N;
- D7: ligado ao pino *trigger* do sensor da direita;
- D8: ligado ao pino *echo* do sensor da direita;
- D13: ligado ao pino de comando do servo motor (fio laranja);
- A0: ligado ao pino IN1 do L298N;
- A1: ligado ao pino IN2 do L298N;
- A2: ligado ao pino IN3 do L298N;
- A3: ligado ao pino IN4 do L298N;
- 5V: ligado no 5 V do circuito;
- GND: ligado no *ground* do circuito.

A figura a seguir mostra o esquema elétrico deste projeto, indicando as ligações entre os componentes.

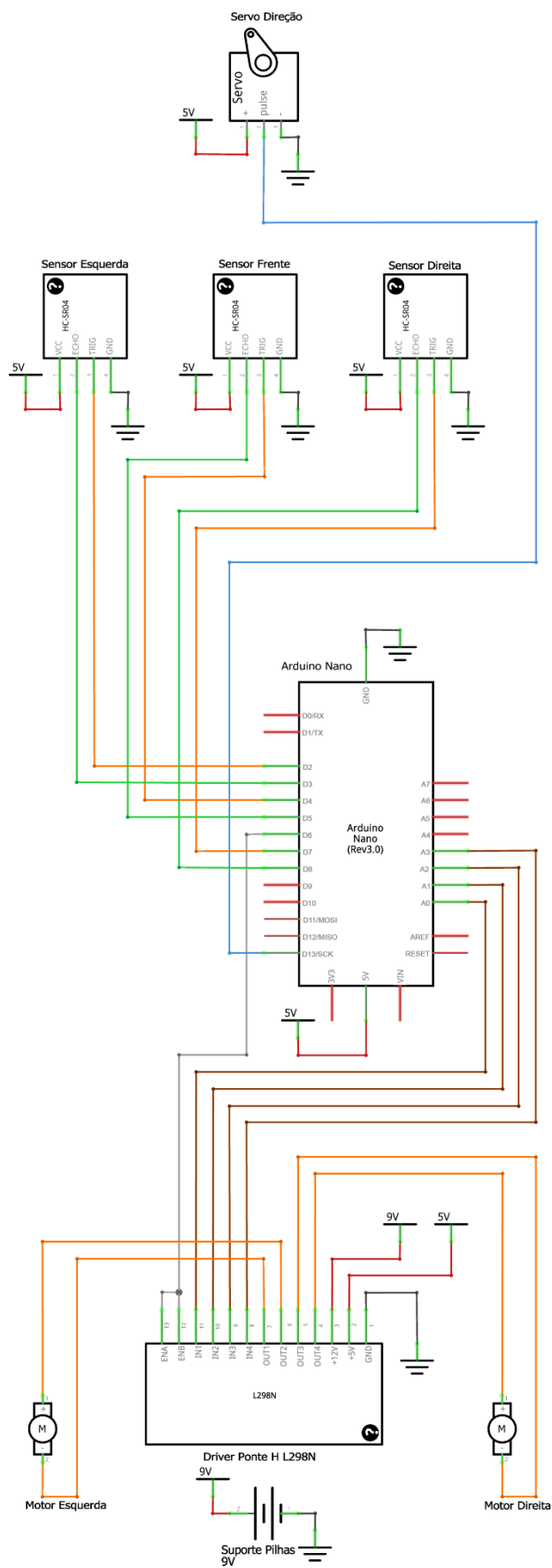


Figura 30: Esquemático do circuito do carro robô.

Fonte: Autoria própria (2018).

### 3.2 ADAPTAÇÃO DO MÉTODO *VIRTUAL FORCE FIELD*

O carro robô possui três sensores presos em sua estrutura, angulados a 60° (sensor direita-frente), 90° (sensor frontal) e 120° (sensor esquerda-frente). Para estabelecer uma região de trabalho limitada (para que obstáculos muito distantes não interfiram no funcionamento do robô), é estabelecido um valor máximo de medição (de 40 cm), de forma que o sensor só realiza medições até este valor (se ele mede um valor maior, o valor considerado é o valor máximo estabelecido). Durante o funcionamento do robô, os sensores realizam as medições de distância até algum obstáculo, dentro da região de detecção.

Para o controle de direção, no *VFF* tradicional o robô divide o espaço ao seu redor em células, na *certainty grid*, e consegue se localizar espacialmente, conhecendo suas coordenadas de posição entre as células, que exercem forças de repulsão no carrinho, de acordo com a distância e o valor de convicção de existência de obstáculo. Como o carro robô proposto neste projeto não tem esta capacidade de se localizar espacialmente, conseguindo apenas realizar medições de distâncias dele até algum obstáculo, sem saber sua posição no espaço, foi feita uma adaptação do método, deixando de utilizar a *certainty grid*. Dessa forma a adaptação desconsidera os valores de certeza (não existem mais células) e reduz o número de elementos que exercem forças de repulsão no carrinho, já que com a *certainty grid* todas as células ao redor do carrinho (dentro da região de detecção e que possuem algum valor de certeza) exercem forças nele, enquanto que na adaptação sem a *certainty grid* só podem existir 3 forças de repulsão, já que o carrinho só possui as medições de distância dele até algum obstáculo no momento (3 sensores realizando medições).

Enquanto que no *VFF* tradicional as forças de repulsão são inversamente proporcionais à distância ao quadrado, a adaptação considera uma relação linear entre a distância medida do obstáculo e a magnitude do vetor de repulsão gerado, de forma que o módulo do vetor é nulo quando a distância medida é o valor máximo estabelecido (não tem obstáculo dentro da região de detecção) e máximo quando a distância é nula (magnitude é o valor da distância máxima).

No *VFF* tradicional é especificado um ponto de destino (coordenadas) para o robô e ele deve chegar até o alvo, desviando dos obstáculos que aparecem no trajeto. A proposta deste projeto é criar um carro robô que procure andar sempre para a frente, desviando dos obstáculos que aparecerem no caminho, com isso o ponto de destino é fixo, sempre à frente do carro. Dessa forma o vetor que representa a força de atração

$F_t$  está sempre “puxando” o robô para frente, num ângulo de  $90^\circ$  com o eixo horizontal. Acerca da magnitude fixa do vetor  $F_t$ , seu valor é 40 (cm) e foi obtido por meio de simulações em um editor de planilhas.

Feitas as adaptações, o método funciona da seguinte forma: primeiramente os sensores ultrassônicos realizam medições de distância até algum obstáculo e, com esses valores, são criados os vetores que representam as forças de repulsão do obstáculo sobre o robô (relação linear). O vetor de repulsão resultante  $F_r$  é criado a partir da soma vetorial dos vetores de repulsão. O vetor de atração  $F_t$  independe de valores de medição, já que ele está sempre posicionado apontando para a direção frontal do robô, a  $90^\circ$ . O vetor resultante  $R$ , cujo ângulo é o desejado para deslocamento, é criado a partir da soma vetorial de  $F_r$  e  $F_t$ . A imagem abaixo mostra uma simulação deste método em funcionamento, indicando os obstáculos e os vetores criados no processo.

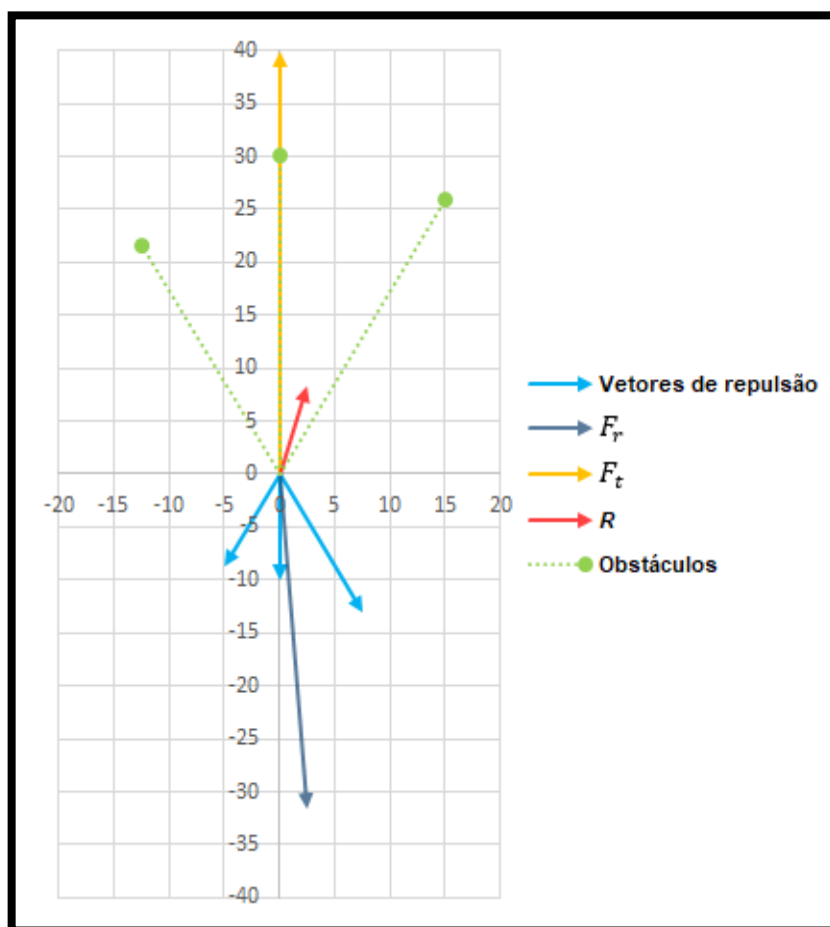


Figura 31: Exemplo de simulação.

Fonte: Autoria própria (2018).

O controle de velocidade do robô é feito utilizando como parâmetro o vetor  $R$ , mais especificamente sua magnitude. Se o robô não encontra obstáculos,  $R$  tem magnitude máxima, portanto o carrinho anda na velocidade máxima. À medida que surgem vetores de repulsão, o módulo de  $R$  diminui, então a velocidade também diminui.

### 3.3 PROGRAMAÇÃO E CONSTRUÇÃO DO PROTÓTIPO

#### 3.3.1 Programação

A programação do carro robô funciona da seguinte forma: existe uma rotina principal, que funciona em *loop*, que realiza as medições de distância dos sensores ultrassônicos. A fim de suavizar os valores, é utilizado um procedimento auxiliar que, quando chamado, realiza uma média das 3 últimas medições. Ainda dentro do *loop* é realizado o controle de direção e velocidade, por meio do *VFF* adaptado.

Existe também uma rotina, ligada ao estouro do Timer2, que funciona em intervalos periódicos (a cada 10 milissegundos, aproximadamente) e realiza a mudança de velocidade e direção dos atuadores, em pequenas etapas a cada estouro do Timer2, de forma que o movimento dos atuadores possa ser suave e não dependa do fluxo de programa da rotina principal.

O código fonte é estruturado da seguinte forma:

- Inclusão de bibliotecas e declaração de variáveis;
- Código da função ligada ao estouro do Timer2 (controle dos atuadores) e das outras funções auxiliares (função de cálculo de distância medida por um sensor e procedimento de cálculo de média das últimas medições);
- Procedimentos principais *setup* e *loop*:
  - *setup*: configuração dos pinos de entrada e saída, configurações iniciais de velocidade e direção, configuração do estouro do Timer2, primeiras medições dos sensores e criação do vetor  $F_t$ ;
  - *loop*: medições dos sensores (cálculo da média com as últimas medições), controle de direção e velocidade.

A seguir estão representados fluxogramas de funcionamento dos principais procedimentos da programação do robô: procedimento do *loop* e procedimento ligado ao estouro do Timer2, que controla os atuadores em etapas.



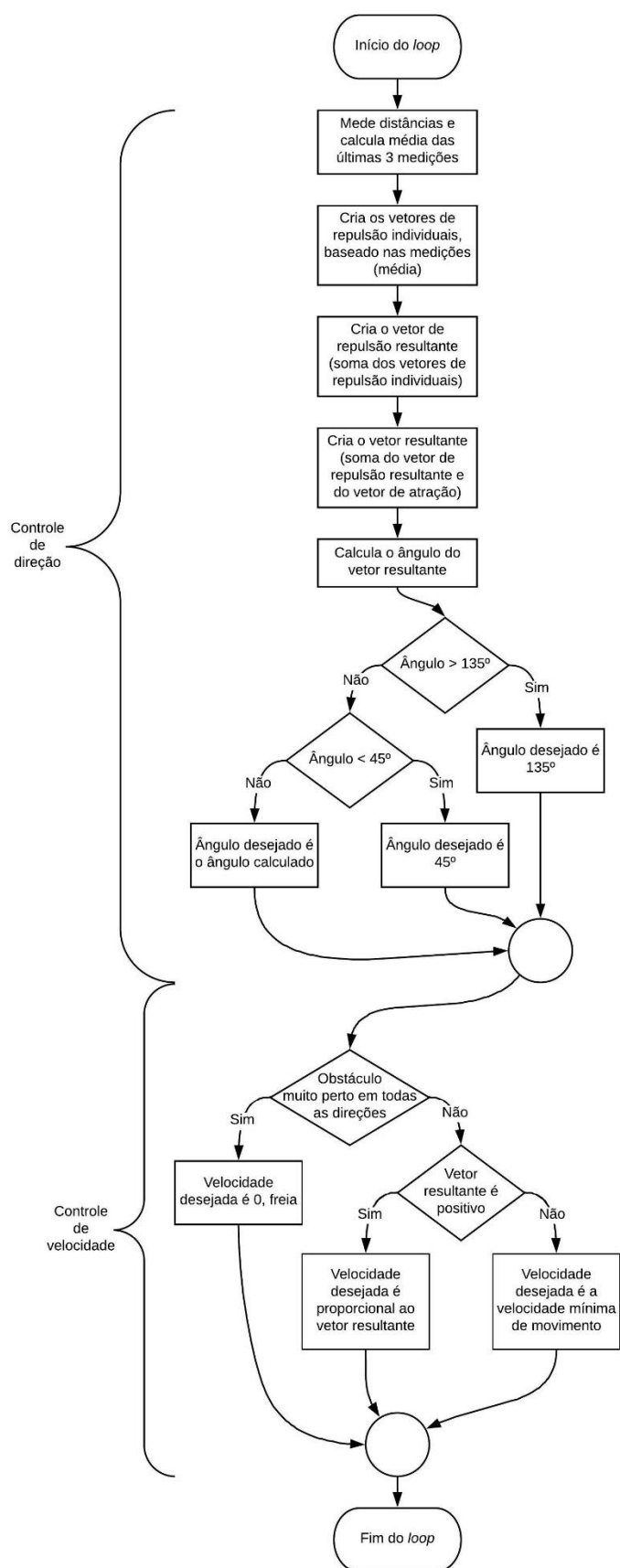


Figura 32: Fluxograma de funcionamento do *loop*.

Fonte: Autoria própria (2018).

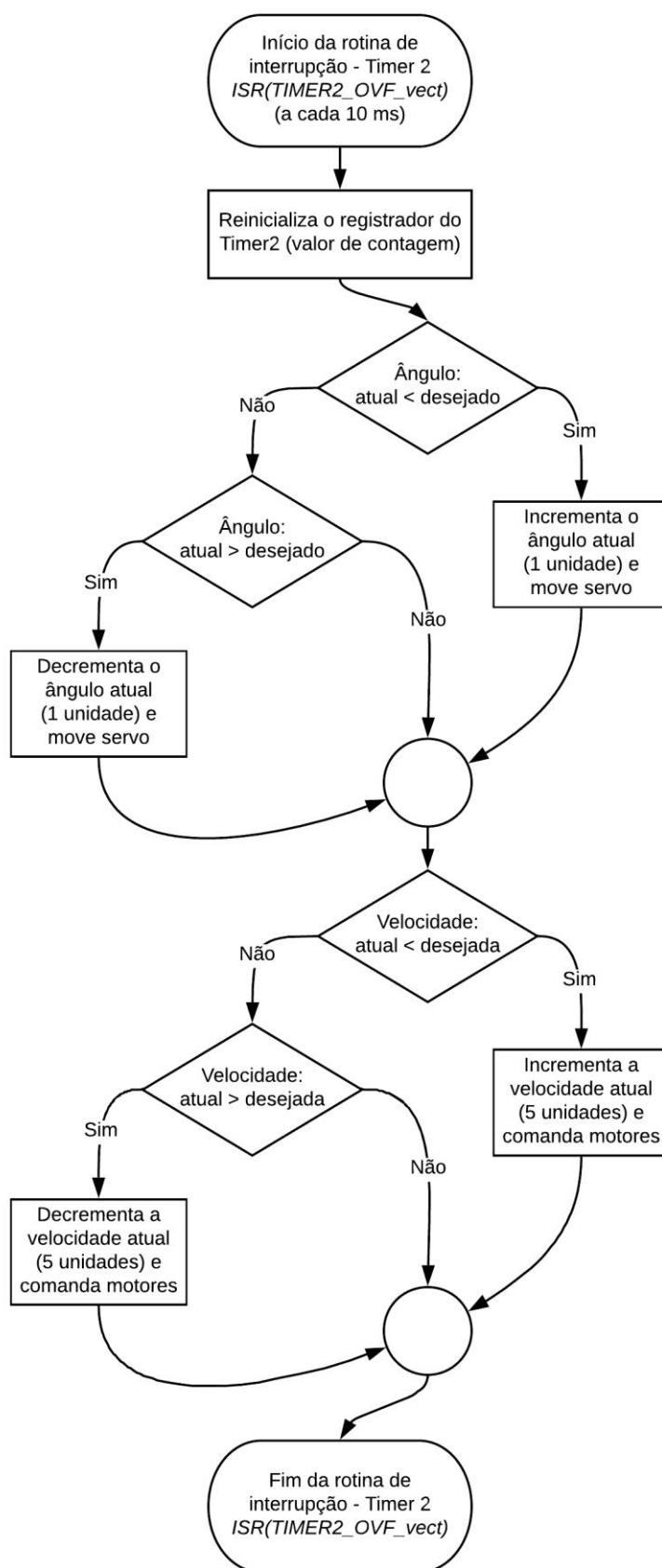


Figura 33: Fluxograma de funcionamento do procedimento ligado ao Timer2.

Fonte: Autoria própria (2018).

### 3.3.2 Construção

A estrutura do carro robô consiste de uma placa de madeira, em que os componentes do projeto foram montados e fixados, com auxílio de outras peças de metal e parafusos.

Como existem vários componentes que compartilham da mesma alimentação (5 V e *ground*), foram fabricadas duas placas de circuito impresso para tornar a montagem mais fácil. As figuras a seguir contém imagens das placas utilizadas e do protótipo finalizado.

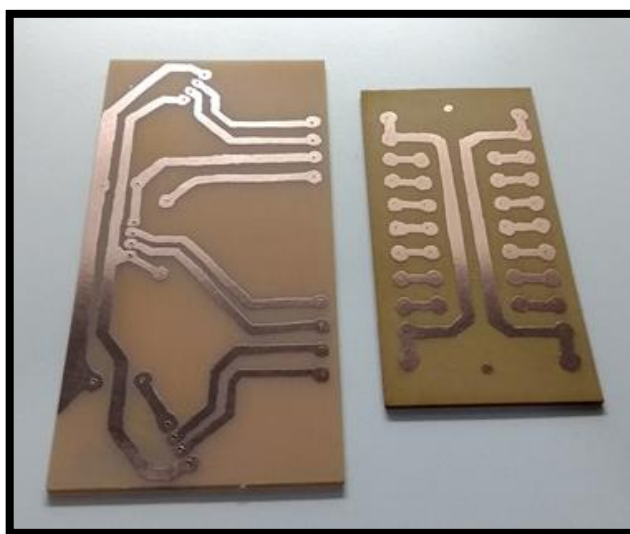


Figura 34: Placas de circuito impresso utilizadas na construção do robô.

Fonte: Autoria própria (2018).

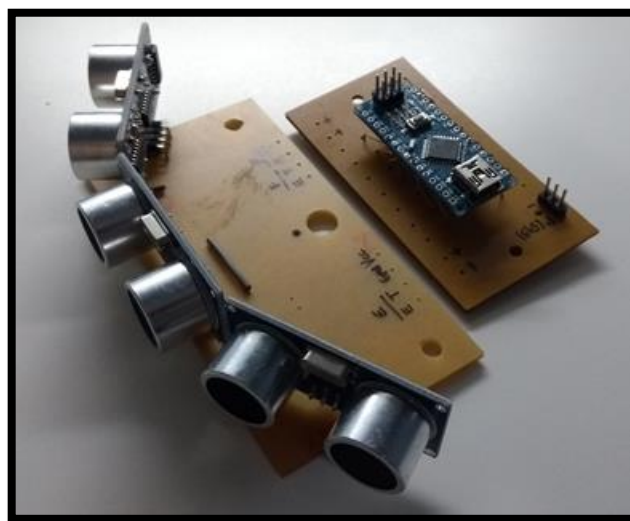


Figura 35: Sensores e Arduino Nano montados nas placas.

Fonte: Autoria própria (2018).

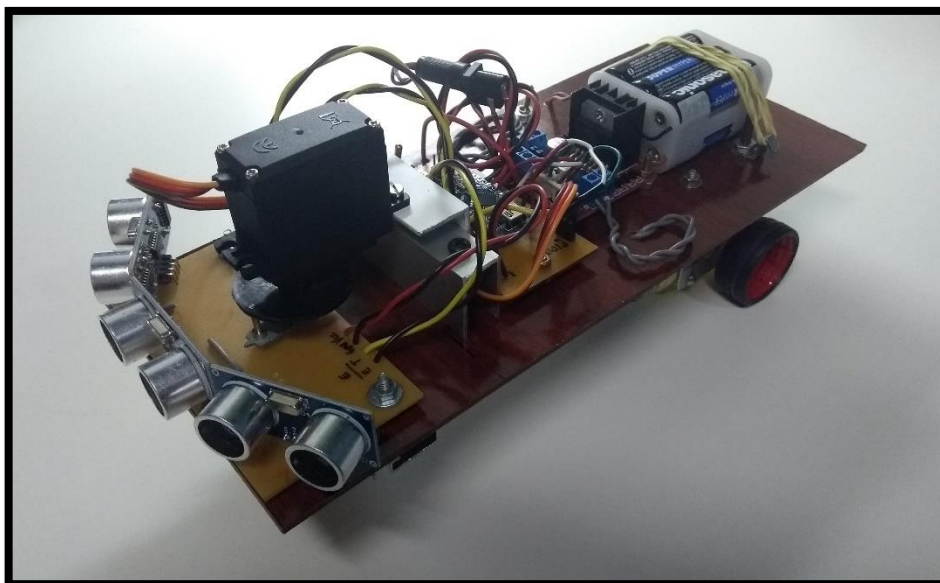


Figura 36: Protótipo finalizado.

Fonte: Autoria própria (2018).

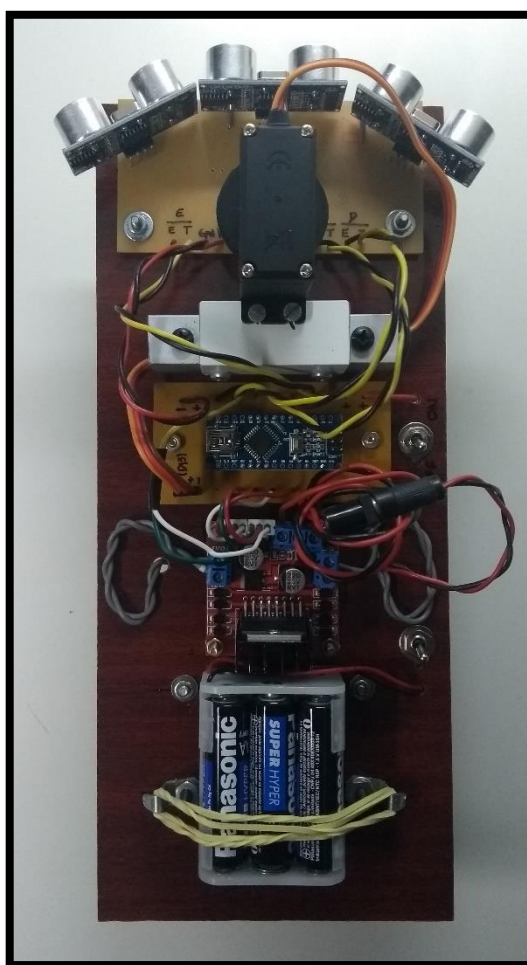


Figura 37: Vista superior do protótipo.

Fonte: Autoria própria (2018).

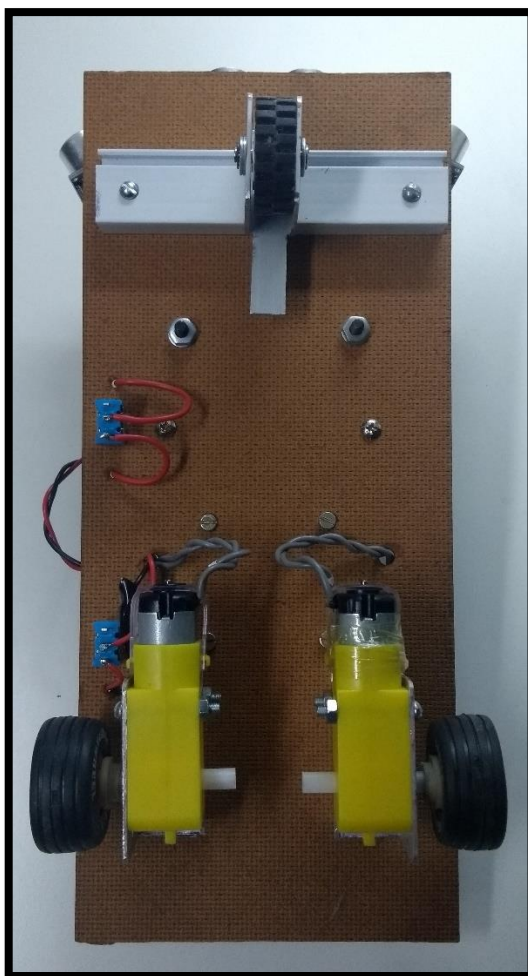


Figura 38: Vista inferior do protótipo.

Fonte: Autoria própria (2018).

#### 4. RESULTADOS E DISCUSSÃO

O protótipo foi testado em um ambiente controlado, com obstáculos previamente selecionados. Foi possível perceber que a adaptação do método *VFF* funcionou muito bem para o projeto proposto, fazendo com que o robô procurasse desviar dos obstáculos com antecedência e de forma suave. À medida que o obstáculo se aproximava do carrinho, ele reduzia a sua velocidade e mudava a angulação da roda de direção, buscando um caminho livre de obstáculos. Se o carrinho tinha obstáculos muito próximos em todas as direções, ele freava, já que não era mais possível o movimento.

Percebeu-se que o robô se comportava de diferentes formas, dependendo do material e da forma do obstáculo. Foi perceptível uma grande dificuldade do carrinho em detectar materiais que tinham superfície macia e/ou irregular, porém isto vem da natureza do sensor ultrassônico, que apresenta desempenho inferior em face de obstáculos com as características citadas, e não tem relação com o método adotado de desvio de obstáculos.

Pelo fato do protótipo ser um robô de movimento, com velocidade ajustável de acordo com o cenário, fica difícil demonstrar com figuras o comportamento do robô frente a cada situação. Nos endereços web permanentes <https://youtu.be/UI7By7wHMCw>, <https://youtu.be/bLChOkn4aVA> e <https://youtu.be/o07kzCjjRbU> pode ser avaliado adequadamente o comportamento de condução autônoma do protótipo construído.

## 5. CONCLUSÕES

A robótica móvel é um tema em ascensão e muito relevante atualmente, com inúmeros estudos e pesquisas sendo realizados, que tem inúmeras aplicações nas mais diversas áreas, como robôs domésticos, militares e os usados na segurança civil.

O robô móvel proposto nesse projeto, que se utiliza de uma adaptação do método *Virtual Force Field* para controle de direção, foi capaz de desviar de obstáculos com sucesso em um ambiente de testes, apresentando melhor desempenho frente a obstáculos previamente selecionados, cuja detecção era garantida, fazendo com que as limitações dos sensores de detecção não influenciassem o funcionamento do robô.

O protótipo desenvolvido poderá servir de base para futuros estudo e projetos acadêmicos na área.

### 5.1 TRABALHOS FUTUROS

Visando a evolução do protótipo desenvolvido neste trabalho, listamos algumas perspectivas:

- Implementação da capacidade de localização espacial, permitindo assim a implementação do método *VFF* tradicional;
- Acréscimo de mais sensores ultrassônicos, inclusive na traseira do robô, a fim de aumentar a capacidade de percepção do ambiente;
- Implementação do movimento de ré, caso o robô se encontre numa situação que o único movimento possível seja para trás;
- Torná-lo semiautônomo, por meio de controle remoto, caso seja necessário intervir no robô durante seu funcionamento;
- Utilizar uma rede neural para controle de direção e velocidade, utilizando o método *VFF* para criar padrões de treinamento para a rede neural.

## REFERÊNCIAS

- [1] WOLF, Denis et al. **Mini-curso 6: Robótica Inteligente: Da Simulação às Aplicações no Mundo Real**. In: JAI - XXVIII JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA, 28., 2009, Bento Gonçalves/RS. Robótica Móvel Inteligente: Da Simulação às Aplicações no Mundo Real. São Paulo: Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2009. p.01-51.
- [2] SIEGWART, Roland; NOURBAKHS, Illah R. **Introduction to Autonomous Mobile Robots**. MIT Press. 2004. XIII+321 p.
- [3] GE, Shuzhi S.; LEWIS, Frank L. **Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications**. CRC/Taylor & Francis, 2006. 709 p.
- [4] HOW, Jonathan P. **Controlling Self Driving Cars**. 2015. Disponível em: <<https://youtu.be/4Y7zG48uHRo>>. Acesso em: 09 nov. 2018.
- [5] ALMEIDA, Jefferson S. et al. **Localization System for Autonomous Mobile Robots Using Machine Learning Methods and Omnidirectional Sonar**. IEEE Latin America Transactions, [s.l.], v. 16, n. 2, p.368-374, fev. 2018. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tla.2018.8327388>.
- [6] BORENSTEIN, Johann; KOREN, Yoram. **Obstacle avoidance with ultrasonic sensors**. IEEE Journal on Robotics and Automation, v. 4, n. 2, p. 213-218, abr. 1988. Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/56.2085>.
- [7] BORENSTEIN, Johann; KOREN, Yoram. **Real-time obstacle avoidance for fast mobile robots**. IEEE Transactions on Systems, Man, and Cybernetics, v. 19, n. 5, p. 1179-1187, set.-out. 1989. Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/21.44033>.
- [8] BORENSTEIN, Johann; KOREN, Yoram. **The vector field histogram-fast obstacle avoidance for mobile robots**. IEEE Transactions on Robotics and Automation, v. 7, n. 3, p. 278-288, jun. 1991. Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/70.88137>.
- [9] BORENSTEIN, Johann et al. **Where am I? Sensors and Methods for Mobile Robot Positioning**. University of Michigan. 1996. 281 p.
- [10] HAND, Aaron. **AGVs Roll Into Food and Bev Plants**. Disponível em: <<https://www.automationworld.com/article/food-and-beverage/agvs-roll-food-and-bev-plants>>. Acesso em: 13 nov. 2018.



- [11] VAUXFORD. **1962 Reliant Regal Mk VI 750cc Front**. Disponível em: <[https://en.wikipedia.org/wiki/Reliant\\_Regal#/media/File:1962\\_Reliant\\_Regal\\_Mk\\_VI\\_750cc\\_Front.jpg](https://en.wikipedia.org/wiki/Reliant_Regal#/media/File:1962_Reliant_Regal_Mk_VI_750cc_Front.jpg)>. Acesso em: 13 nov. 2018.
- [12] MELLO, Dorival A.; WATANABE, Renate G. **Vetores e Uma Iniciação à Geometria Analítica**. Livraria da Física. 1999. 185 p.
- [13] BOULOS, Paulo; CAMARGO, Ivan. **Geometria Analítica: Um Tratamento Vetorial**. McGraw-Hill. 1987. 385 p.
- [14] CAMPAGNER, C. A. **Soma de vetores: Adição gráfica e por decomposição**. Disponível em: <<https://educacao.uol.com.br/disciplinas/matematica/soma-de-vetores-adicao-grafica-e-por-decomposicao.htm>>. Acesso em: 13 nov. 2018.
- [15] MECAWEB. **PWM - Modulação Por Largura de Pulso**. Disponível em: <[http://www.mecaweb.com.br/eletronica/content/e\\_pwm](http://www.mecaweb.com.br/eletronica/content/e_pwm)>. Acesso em: 14 nov. 2018.
- [16] BRAGA, Newton C. **Como funcionam os sensores ultrassônicos (ART691)**. Disponível em: <<http://www.newtoncbraga.com.br/index.php/como-funciona/5273-art691>>. Acesso em: 14 nov. 2018.
- [17] MOTA, Allan. **HC-SR04 – Sensor Ultrassônico de distância com Arduino**. Disponível em: <<https://portal.vidadesilicio.com.br/hc-sr04-sensor-ultrassonico>>. Acesso em: 14 nov. 2018.
- [18] BRAGA, Newton C. **Como funcionam os servos (MEC066a)**. Disponível em: <<http://www.newtoncbraga.com.br/index.php/robotica/5164-mec066a>>. Acesso em: 14 nov. 2018.
- [19] MOTA, Allan. **O que é Servomotor? Controlando um Servo com Arduino**. Disponível em: <<https://portal.vidadesilicio.com.br/o-que-e-servomotor>>. Acesso em: 14 nov. 2018.
- [20] ROBÓTICA FÁCIL. **Servo**. Disponível em: <<http://dyor.roboticafacil.es/en/servo>>. Acesso em: 14 nov. 2018.
- [21] SILVEIRA, Cristiano B. **Motor CC: Saiba como Funciona e de que forma Especificar**. Disponível em: <<https://www.citisystems.com.br/motor-cc/>>. Acesso em: 14 nov. 2018.
- [22] TIPLER, Paul A.; MOSCA, Gene. **Física para Cientistas e Engenheiros: Eletricidade e Magnetismo, Óptica. Vol. 2**. LTC. 2011. 556 p.

- [23] REIS, Fábio. **Como funciona uma Ponte H – Controle direcional de motores DC**. Disponível em: <<http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/como-funciona-uma-ponte-h-controle-direcional-de-motores-dc>>. Acesso em: 14 nov. 2018.
- [24] BRAGA, Newton C. **Ponte-H com controle PWM (mec009)**. Disponível em: <<http://www.newtoncbraga.com.br/index.php/robotica/1213-ponte-h-com-pwm>>. Acesso em: 14 nov. 2018.
- [25] BRAGA, Newton C. **O básico sobre os Microcontroladores – parte 1 (MIC139)**. Disponível em: <<http://newtoncbraga.com.br/index.php/eletronica/52-artigos-diversos/13263-o-basico-sobre-os-microcontroladores-parte-1-mic139>>. Acesso em: 14 nov. 2018.
- [26] FILIPEFLOP. **Placa Uno R3 + Cabo USB para Arduino**. Disponível em: <<https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino>>. Acesso em: 14 nov. 2018.
- [27] ARDUINO. **WHAT IS ARDUINO**. Disponível em: <<https://www.arduino.cc>>. Acesso em: 16 nov. 2018.
- [28] ARDUINO. **Arduino Nano**. Disponível em: <<https://store.arduino.cc/usa/arduino-nano>>. Acesso em: 16 nov. 2018.
- [29] JOHN. **Arduino Nano Tutorial – Pinout & Schematics**. Disponível em: <<http://www.circuitstoday.com/arduino-nano-tutorial-pinout-schematics>>. Acesso em: 16 nov. 2018.
- [30] SOUZA, Fábio. **Timers do ATmega328 no Arduino**. Disponível em: <<https://www.embarcados.com.br/timers-do-atmega328-no-arduino>>. Acesso em: 21 nov. 2018.
- [31] QUEIROZ, Tiago F. **Tutorial: Executando funções em intervalos de tempo fixos (timers) com Arduino**. Disponível em: <<http://labdegaragem.com/profiles/blogs/tutorial-executando-fun-es-em-intervalos-de-tempo-fixos-timers>>. Acesso em: 21 nov. 2018.
- [32] ELEC FREAKS. **Ultrasonic Ranging Module HC – SR04**. Disponível em: <<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>>. Acesso em: 17 nov. 2018.
- [33] FILIPEFLOP. **Servo Tower Pro SG-5010 + Acessórios**. Disponível em: <<https://www.filipeflop.com/produto/servo-tower-pro-sg-5010-acessorios>>. Acesso em: 17 nov. 2018.

- [34] GUEDES, Mariana. **Módulo Ponte H L298N – O primeiro passo para montar seu robot com Arduino**. Disponível em: <<https://www.arduinoportugal.pt/modulo-ponte-h-l298n-primeiro-passo-montar-robo-arduino>>. Acesso em: 17 nov. 2018.
- [35] STMICROELECTRONICS. **DUAL FULL BRIDGE DRIVER**. Disponível em: <[https://www.sparkfun.com/datasheets/Robotics/L298\\_H\\_Bridge.pdf](https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)>. Acesso em: 17 nov. 2018.
- [36] FILIPEFLOP. **Motor DC 3-6V com Caixa de Redução e Eixo Duplo**. Disponível em: <<https://www.filipeflop.com/produto/motor-dc-3-6v-com-caixa-de-reducao-e-eixo-duplo>>. Acesso em: 17 nov. 2018.