

UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

RAFAEL SCHMITZ DE OLIVEIRA

ROBÔ MÓVEL UTILIZANDO LÓGICA FUZZY NA PLATAFORMA ARDUINO

CRICIÚMA

2017

RAFAEL SCHMITZ DE OLIVEIRA

ROBÔ MÓVEL UTILIZANDO LÓGICA FUZZY NA PLATAFORMA ARDUINO

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel em Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. MSc. Paulo João Martins

Coorientadora: Profa. Dra. Merisandra Côrtes
de Mattos Garcia

CRICIÚMA

2017

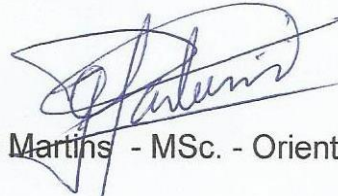
RAFAEL SCHMITZ DE OLIVEIRA

ROBÔ MÓVEL UTILIZANDO LÓGICA FUZZY NA PLATAFORMA ARDUINO

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Automação.

Criciúma, 20 de junho de 2017.

BANCA EXAMINADORA



Prof. Paulo João Martins - MSc. - Orientador



Profª. Merisandra Cortes de Mattos Garcia – Dra. - (UNESC) -
Coorientadora



Prof. Sergio Coral - Especialista - Esp. - (UNESC)



Prof. Valter Blauth Junior - Especialista - Esp. - (UNESC)

Aos meus pais e familiares que sempre me apoiaram.

AGRADECIMENTOS

Quero agradecer primeiramente a Deus pelo dom da vida, a minha família, aos meus pais que me ajudaram a realizar mais esta etapa na minha vida, ao meu irmão que sempre esteve junto me incentivando.

Meus agradecimentos aos meus colegas de sala que me proporcionaram que a caminhada até aqui pudesse ser mais amena e prazerosa, aos professores que fizeram o possível e o impossível para garantir o melhor aprendizado tanto para mim quanto aos meus colegas.

Agradeço a minha instituição e a todos que fazem parte da Universidade do Extremo Sul Catarinense – UNESC, ser referência no ensino de qualidade, a todos meu muitíssimo obrigado.

Por fim e não menos importante incondicionalmente ao meu orientador Paulo João Martins e minha coorientadora Merisandra Côrtes de Mattos Garcia, pelas suas dedicações, compreensão e por ter confiado na minha capacidade para chegar até aqui, obrigado a todos sem vocês não seria possível a realização desse sonho.

“O homem é tão bom quanto o seu desenvolvimento tecnológico o permite ser.”

George Orwell

RESUMO

A robótica é um campo que está cada vez mais presente tanto para fins militares, quanto sociais, como saúde, domésticos, entre outros. Com os avanços uma subárea deste campo vem se destacando que são as criações de robôs móveis autônomos, com inteligência aplicada ou não. Deste modo, esta pesquisa tem como principal objetivo criar um protótipo de veículo autônomo aplicando lógica *fuzzy* para o seu deslocamento por meio de um microcontrolador arduino. O projeto é composto por uma placa arduino uno R3, sensor ultrassônico HC-SR04, ponte H para alimentar os motores e um servomotor para permitir que o sensor verifique as direções. A lógica *fuzzy* aplicada no projeto é responsável por controlar a velocidade em relação ao obstáculo, permitindo-o realizar os desvios necessários durante a trajetória. A mesma utiliza-se de três regras para realizar as comparações e determinar qual será a melhor velocidade a ser aplicada em relação ao obstáculo encontrado. Seus resultados foram satisfatórios para os quesitos de controle de velocidade em relação a distância por meio da lógica *fuzzy*, o sensor utilizado para a detecção de obstáculos HC-SR04, a função realiza cinco leituras para cada direção, devido ao tempo de resposta das ondas sonoras do sensor, pois acima de cinco leituras havia um atraso e o mesmo colidia com o obstáculo. A lógica *fuzzy* aplicada a área de robótica tem obtido bons resultados, e tem auxiliado a obter a solução para alguns problemas de percurso, velocidade, entre outros.

palavras-chave: Lógica Fuzzy. Arduino. Robô Autônomo.

ABSTRACT

Robotics is a field that is increasingly present for military as well as social purposes, such as health, domestic, among others. With the advances a subarea of this field has been emphasizing that they are the creations of autonomous mobile robots, with intelligence applied or not. In this way, this research has as main objective to create a prototype of autonomous vehicle applying fuzzy logic for its displacement by means of an arduino microcontroller. The design consists of an arduino board R3, ultrasonic sensor HC-SR04, H-bridge to feed the motors and a servomotor to allow the sensor to check the directions. The fuzzy logic applied in the project is responsible for controlling the speed in relation to the obstacle, allowing it to realize the necessary deviations during the trajectory. The same is used of three rules to make the comparisons and determine what will be the best speed to be applied in relation to the obstacle encountered. Its results were satisfactory for the velocity control tasks in relation to the distance by means of the fuzzy logic, the sensor used for the detection of obstacles HC-SR04, the function performs five readings for each direction, due to the response time of the sound waves Of the sensor, because over five readings there was a delay and the same collided with the obstacle. The fuzzy logic applied to the robotics area has obtained good results, and has helped to obtain the solution for some problems of course, speed, among others.

Keywords: Fuzzy Logic. Arduino. Autonomous Robot.

LISTA DE ILUSTRAÇÕES

Figura 1 - Tipos de robôs móveis	111
Figura 2 - Tipos de rodas	113
Figura 3 - Modelo unicycle	114
Figura 4 - Modelo cinemático para robô unicycle	116
Figura 5 - Método de transmissão-recepção	119
Figura 6 – Método pulso-eco.....	119
Figura 7 – Sensor ultrassônico HC-SR04	120
Figura 8 – Servomotor.....	123
Figura 9 – Motor DC utilizado no protótipo	124
Figura 10 – Placa arduino UNO e seus componentes	128
Figura 11 - Mapa de pinagem Atmega168 e Atmega328P	130
Figura 12 - Especificações Técnicas do Arduino Uno	132
Figura 13 - L293D utilizado na ponte H.....	134
Figura 14 - Shield L293D Driver Ponte H	135
Figura 15 - Funcionamento interno de uma ponte H.....	136
Figura 16 - Comportamento gráfico da lógica booleana.....	137
Figura 17 - Comportamento gráfico da lógica <i>fuzzy</i>	138
Figura 18 – Sistema de funcionamento de lógica <i>fuzzy</i>	140
Figura 19 – Formato de função de pertinência triangular	141
Figura 20 - Formato de função de pertinência trapezoidal	142
Figura 21 – Expressão de função de pertinência Gaussiana	143
Figura 22 - Função de pertinência <i>fuzzy Singleton</i>	143
Figura 23 - Função de pertinência para a variável linguística altura	145
Figura 24 - Classes da biblioteca eFLL.....	148
Figura 25 - Fluxograma do funcionamento interno.....	154
Figura 26 - Modelagem do protótipo	155
Figura 27 – Estrutura do protótipo.....	156
Figura 28 - Ligação do sensor ao protótipo	156
Figura 29 - Teste do sensor	157
Figura 30 - Pseudocódigo para capturar a distância do obstáculo.....	158
Figura 31 - Modelagem fuzzy	158
Figura 32 - Função de pertinência trapezoidal	160

Figura 33 - Gráfico das funções de pertinência da variável distância	160
Figura 34 - Gráfico das funções de pertinência da variável velocidade	161
Figura 35 - Gráfico do grau de pertinência	164
Figura 36 - Apresentação no monitor serial do exemplo inicializado na fuzificação	166
Figura 37- Ângulo de posicionamento do servomotor	167
Figura 38 - Funções de locomoção	168
Figura 39 - Função de melhor distância	169
Figura 40 - Leitura do sensor verificada no monitor serial.....	171
Figura 41 - Teste de locomoção e distância.....	172

LISTA DE TABELAS

Tabela 1 - Base de regras	162
---------------------------------	-----

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
A/D	Analógico para Digital
AEC	Controle de Exposição
AGC	Controle automático de ganho
AGV's	Veículos Guiados Automaticamente
AWB	Balanço de Branco Automático
CA	Corrente alternada
CAN	<i>Controller Area Network</i>
CC	Corrente Continua
cm	Centímetros
CPU	<i>Central Processing Unit</i>
D/A	Digital para Analógico
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory</i>
eFLL	<i>Embedded Fuzzy Logic Library</i>
EREF	<i>Executive real estate finance</i>
fps	<i>Frames Per Second/Quadros Por Segundo</i>
Full HD	<i>Full high definition</i>
FTDI	<i>Future Technology Devices International</i>
g	Gramas
GB	Gigabyte
GHz	Gigahertz
GPS	<i>Global Positioning System</i>
Hz	Hertz
IDE	<i>Integrated Development Environment</i>
I ² C	<i>Inter-Integrated Circuit</i>
JTAG	<i>Joint Test Access Group</i>
K	Kilo
kB	Kilobyte
KTS	Kang-Takagi-Sugeno
KHz	Quilohertz
LTAs	<i>Lighter-Than-Air</i>

LCD	<i>Liquid Crystal Display</i>
LED	<i>Light Emitting Diode</i>
mA	Miliampere
MHz	Megahertz
mm	Milímetro
m/s	Metro por Segundo
NASA	<i>National Aeronautics and Space Administration</i>
rpm	Rotação por minuto
PWM	<i>Pulse Width Modulation</i>
RAM	<i>Random Access Memory</i>
RISC	<i>Reduced Instruction Set Computing</i>
ROM	<i>Read-only memory</i>
SONAR	<i>Sound Navigation and Ranging</i>
SPI	<i>Serial peripheral interface</i>
SRAM	<i>Static Random Access Memory</i>
UART	<i>Universal Asynchronous Receiver/ Transmitter</i>
UNESC	Universidade do Extremo Sul Catarinense
μS	Microsegundo
USB	<i>Universal Serial Bus</i>
Uespi	Universidade Estadual do Piauí
v	Volts
VGA	<i>Video Graphics Array</i>

SUMÁRIO

1	INTRODUÇÃO	101
1.1	OBJETIVO GERAL	102
1.2	OBJETIVOS ESPECIFICOS	102
1.3	JUSTIFICATIVA	103
1.4	ESTRUTURA DO TRABALHO	104
2	ROBÓTICA	106
2.1	VEÍCULOS AUTÔNOMOS E ROBÔS	107
2.1.1	Robô móvel autônomo	110
2.1.2	Tipos de robôs móveis	111
2.2	CINEMÁTICA	114
2.2.1	Modelos cinemáticos e restrições	115
2.3	SENSORES	116
2.3.1	Sensor ultrassônico	118
2.3.2	Sensor HC-SR04	119
2.3.3	Atuadores	121
3	MICROCONTROLADORES	125
3.1	ARDUINO	126
3.1.1	Entrada e saída	130
3.1.2	Memória arduino Uno	132
3.1.3	Alimentação	132
3.1.4	Shield L293D driver ponte H	133
4	LÓGICA FUZZY	136
4.1	CONTROLE FUZZY	138
4.1.1	Fuzificação	140
4.1.2	Base de regras fuzzy	144
4.1.3	Inferência fuzzy	145
4.1.4	Defuzificação	146
4.1.5	Biblioteca eFLL	147
5	TRABALHOS CORRELATOS	149
5.1	APLICAÇÃO DA LÓGICA FUZZY NO CONTROLE DA VELOCIDADE DE UM PROTÓTIPO MÓVEL	149

5.2	DESENVOLVIMENTO DE UMA BIBLIOTECA FUZZY PARA O CONTROLE AUTÔNOMO DE UM ROBÔ MÓVEL EM AMBIENTE DESCONHECIDO	150
5.3	NAVEGAÇÃO EM AMBIENTES DESCONHECIDOS POR ROBÔ MÓVEL AUTÔNOMO BASEADO EM PLATAFORMA ARDUINO COM LINGUAGEM C.....	151
5.4	AUTOBOMOUS ROBOT OBSTACLE AVOIDANCE USING A FUZZY LOGIC TRONTROL SCHEME	152
6	PROTÓTIPO DESENVOLVIDO COM INTEGRAÇÃO DA PLATAFORMA ARDUINO COM LÓGICA FUZZY	153
6.1	MONTAGEM DA ESTRUTURA	155
6.2	TESTE DO SENSOR HC-SR04.....	157
6.3	MODELAGEM FUZZY	158
6.3.1	Definição das funções de pertinência.....	159
6.3.2	Definição da base de regras	162
6.3.3	Fuzificação	162
6.3.4	Avaliação das regras	163
6.3.5	Defuzificação.....	163
6.4	MOVIMENTAÇÃO DE ATUADORES	166
7	ANÁLISE DOS RESULTADOS OBTIDOS	170
8	CONCLUSÃO.....	173
	REFERÊNCIAS.....	175

1 INTRODUÇÃO

A utilização da automação de processos por meio da robótica vem sendo estudada e pesquisada a anos por diferentes instituições do mundo, além de contribuir à sociedade, também proporciona o desenvolvimento de estratégias de conhecimento, como a engenharia e informática (DIAS; COSTA; CARVALHO, 2012).

Um sistema computacional com propósitos específicos pode ser considerado como um sistema embarcado, geralmente, os mesmos possuem tamanhos reduzidos, e trabalham de forma autônoma. Como exemplo destes modelos pode-se citar equipamentos médicos especializados; celulares; sistemas automobilísticos como injeção eletrônica, freios; entre outros (JUNG et al, 2005).

O robô móvel pode ser utilizado em uma vasta gama de tarefas, desde transporte de peças até o reconhecimento de ambientes que oferecem riscos, como crateras, ambientes espaciais, áreas radioativas, entre outros. O que difere deste modelo para os robôs manipuladores é a sua liberdade de locomoção. Portanto, o seu deslocamento exige um maior grau de comunicação com o ambiente, o que necessita da interação de sensores para a captação do ambiente e eventualmente a detecção do resultado das ações do próprio robô.

Segundo Gudiwin (2005) e Wolf et al. (2009), robôs móveis são objetos de intensa atividade de pesquisa, devido a facilidade de realizar testes por meio de simuladores. Outro fator importante é a complexidade específica ao seu funcionamento, assim tornando-o um laboratório de pesquisa para a inteligência artificial.

A robótica utiliza-se de um conjunto vasto de sensores como, câmeras, ultrassom, infravermelho entre outros, que permitem identificar o ambiente. Os robôs móveis deslocam-se por meio de rodas, pernas, esteiras, ou mecanismos semelhantes, assim permitindo-o a utilização dos mesmos para os mais diversos fins (RUSSELL; NORVIG, 2013).

Segundo Almeida (2012) e Luger (2013), apesar de possuir uma vasta quantidade de pesquisa sobre o assunto, o tema continua sendo desafiador, pois realizar uma tarefa de alto nível como, locomover-se em uma sala com obstáculo necessita de uma sequência de ações atômicas¹ para realizá-la.

Deste modo, pretende-se integrar o sensor ultrassônico com a plataforma arduino e lógica *fuzzy*, para realizar o deslocamento de um robô autônomo em ambientes desconhecidos.

Considerando-se o exposto esta pesquisa consiste no desenvolvimento de um protótipo de veículo autômato por meio de lógica *fuzzy* para realizar o controle de velocidade em relação ao obstáculo durante o percurso.

1.1 OBJETIVO GERAL

Esta pesquisa tem como objetivo desenvolver um protótipo de um veículo autônomo, utilizando inteligência artificial por meio de lógica *fuzzy*, e sensor ultrassônico integrado ao Arduino.

1.2 OBJETIVOS ESPECIFICOS

Os objetivos específicos deste Trabalho de Conclusão de Curso são:

- a) conhecer os projetos existentes para a montagem de um veículo autônomo de custo mais acessível;
- b) utilizar a linguagem de programação Arduino;
- c) obter a distância de desvio de obstáculo por meio de sensor ultrassom;
- d) aplicar testes no protótipo desenvolvido;

¹ É uma ação que é indivisível, não pode ser quebrada numa sequência simples de ações mais simples.

e) aplicar lógica fuzzy no controle do velocidade em relação ao obstáculo.

1.3 JUSTIFICATIVA

O arduino é um projeto que engloba *software* e *hardware* para processar entradas e saídas por meio de conexões entre os dispositivos. O arduino também é conhecido como plataforma embarcada devido a integração de sensores com sua plataforma (MCROBERTS, 2011).

Dentre os fatores que levaram a escolha do arduino para a automação embarcada, é por se tratar de uma plataforma gratuita. Deste modo, pode-se trabalhar com projetos livres, sendo possível a prototipação de seus equipamentos eletrônicos de *hardware* e *software*. Sendo assim o mesmo permite a criação de um robô móvel.

Segundo Russell e Norvig (2013), os robôs são agentes físicos que realizam tarefas por meio de seus atuadores e sensores. Deste modo, o mesmo é equipado por atuadores como pernas, rodas, articulações e garras. Estes atuadores têm como principal propósito exercer força física sobre o ambiente, enquanto os sensores detectam o ambiente. A maioria dos robôs existentes se enquadra em uma das seguintes categorias: fixos, humanóides e *off-road*.

Para o desenvolvimento deste projeto aplicou-se o modelo *off-road* por meio de Inteligência Artificial (I A) que surgiu após o fim da Segunda Guerra Mundial, sendo uma das ciências mais recentes da atualidade. Há I.A estrutura e automatiza tarefas intelectuais, portanto torna-se um campo universal nestes aspectos (RUSSELL; NORVIG, 2004).

Portanto, a criação de um protótipo veicular autônomo utilizando I A por meio de lógica *fuzzy*, tem como finalidade controlar a velocidade em relação ao obstáculo encontrado no ambiente. Este método permite a locomoção em locais que necessitem de cuidados devido as suas estruturas, ou que apresente algum risco aos seres humanos.

Dentre os fatores que levaram a escolha de lógica *fuzzy* para o controle de velocidade em relação ao obstáculo são:

- a) permite possuir inúmeras variáveis de entrada e saída sendo capaz de processar a ação semelhante ao operador humano;
- b) permite a representação linguística das definições vagas e imprecisas pertinente ao raciocínio humano (SHAW; SIMÕES, 2007).

1.4 ESTRUTURA DO TRABALHO

O projeto é formado por oito capítulos que apresentam as métricas e etapas realizadas. O primeiro demonstra a introdução para o assunto abordado, o objetivo geral, os objetivos específicos e a justificativa para a realização do projeto.

No segundo capítulo aborda-se a robótica, os veículos autônomos, os robôs e suas classificações, robô móvel autônomo, os tipos de robôs móveis, quais tipos de autonomia existem para robôs, neste capítulo também é abordado a utilização da cinemática em robôs unicyclos.

O terceiro capítulo é dedicado ao microcontrolador, também ao histórico do arduino como seu surgimento, ao modelo arduino UNO, entradas e saídas, alimentação, memória, sensores e suas classificações na robótica, a micro câmera do Samsung Galaxy J7 na captura de imagens, sensor ultrassônico e sensor HC-SR04 e suas características. Outra abordagem muito interessante que este capítulo faz, é sobre a biblioteca eFLL, onde a mesma permite aplicação de lógica *fuzzy* com mais facilidade na plataforma arduino.

O quarto capítulo é dedicado à lógica *fuzzy*, como seu surgimento, também é abordado o funcionamento de um controlador *fuzzy*, juntamente a base de regras, controlador *fuzzy*, variáveis linguísticas e funções de pertinência.

No quinto capítulo, são demonstrados os trabalhos permitindo um maior embasamento no desenvolvimento do trabalho.

O sexto capítulo apresenta o que foi desenvolvido neste projeto, bem como a metodologia, e os materiais utilizados para o desenvolvimento do mesmo.

No Sétimo capítulo encontra-se os resultados obtidos com o desenvolvimento deste trabalho, bem como suas vantagens e desvantagens.

No oitavo e último capítulo aborda-se a conclusão alcançada com o desenvolvimento deste trabalho, bem como suas vantagens e a possibilidade de trabalhos futuros do mesmo.

2 ROBÓTICA

A robótica é uma área de pesquisa multidisciplinar que engloba vários conteúdos de diferentes campos de pesquisa, desta forma auxilia na construção de robôs para os mais diversos fins (POZZEBON; FRIGO, 2013).

A robótica foi definida e explicada por volta de 1940 quando Isaac Asimov criou e propôs as 3 leis a seguir:

“1.^a) Um robô não pode fazer mal a um ser humano e nem consentir, permanecendo inoperante, que um ser humano se exponha a situação de perigo.

2.^a) Um robô deve obedecer sempre às ordens de seres humanos, exceto em circunstâncias em que essas ordens entrem em conflito com a 1^a lei.

3.^a) Um robô deve proteger a sua própria existência, exceto em circunstância que entrem em conflito com a 1^a e 2^a leis.” (ROMANO, 2002, p.1).

Algum tempo depois e de realizar experimentos, Isaac Asimov inseriu a lei zero onde diz: (HALFPAP; SOUZA; ALVES, 2007).

0^a) “Um robô não deve fazer mal à humanidade, ou, permanecer passivo numa situação que prejudique a humanidade” (ROMANO, 2002, p.1).

A primeira patente registrada na história de um robô controlado por computadores foi em 1954, registrado no nome de *George Devol*, mas isso só foi possível graças a invenção do transistor em 1948 que permitiu o controle dos robôs pelos computadores (DE PIERI, 2002).

Em 1983 o método *freeway*, serve para planejamento de trajetória, porém possuía restrições para as criações e manutenções de mapas do ambiente. Já em 1986 começou a utilização de leituras baseada em seus sensores. Esta última arquitetura possibilitou realizar pesquisas de métodos reativos e auxiliou para a construção de robôs móveis (DE PIERI, 2002).

No ano de 1990 surgiram novas arquiteturas onde ocorreram evoluções como nas arquiteturas de Mataric, que observou a necessidade de melhorar a capacidade dos robôs, com mapas criados por meio de marcas detectadas no ambiente. No mesmo ano *Zelinski* pesquisou um método que se

baseava no mapeamento em tempo real. A partir do ano de 1994 a robótica evoluiu principalmente na área da robótica móvel (DE PIERI, 2002).

A *National Aeronautics and Space Administration* (NASA) com parceria com a *Carnegie Mellon University*, projetaram um robô para inspecionar vulcões em atividade, em julho de 1997 foi enviado o robô “*Soujourner*” para a expedição a Marte onde chegou ao seu destino no ano de 2000. Já no ano de 2000 a *Carnegie Mellon University*, projetou um robô de quatro rodas utilizando técnicas de Inteligência Artificial para coletar amostras de meteoritos caídos na Antártida, onde o mesmo já classificava as pedras (DE PIERI, 2002).

A evolução dos robôs móveis com sensores e raciocínio existentes no mercado, tem como principal inspiração, criações de séculos passados, como por exemplo, a criação de Leonardo Da Vinci do século XV nos anos noventa o robô Humanoide ou as criações do século XX como a máquina *Speculatrix* dos anos cinquenta e *Shakey* dos anos setenta (SECCHI, 2008).

Os robôs chegaram ao mercado industrial no começo dos anos 60 de forma mais significativa, devido às possibilidades que o mesmo oferecia ao processo produtivo (SECCHI, 2008).

Então a partir desta década houve uma evolução dos princípios de sensores, inteligência artificial e processamento de imagens, tendo como objetivo proporcionar robôs com capacidade de atuar em diversos ambientes e que possuíssem uma estrutura robusta (DE PIERI, 2002).

Com o avanço da robótica surgiu a noção de célula de fabricação robotizada e iniciou-se o novo passo na automação industrial, que por sua vez deixou muito mais flexível o modo de produção. Então na década de 80 começou-se à utilização de veículos guiados automaticamente (AGV's) (SECCHI, 2008).

2.1 VEÍCULOS AUTÔNOMOS E ROBÔS

O desenvolvimento de veículos autônomos necessita da integração de diversos componentes de *software* e dispositivos como: sensores,

atuadores e computacionais, deste modo o desenvolvimento dos mesmos se torna uma tarefa muito complexa a ser realizada. Portanto, torna-se inviável o desenvolvimento de veículos sem o uso de bibliotecas e ferramentas especializadas. Na atualidade os grupos de pesquisa e desenvolvimento em veículos autônomos vêm utilizando plataformas de código aberto nas implementações realizadas, o que leva há muitas vantagens em respeito aos outros tipos de plataforma existentes, principalmente nos quesitos de reaproveitamento de código, padronização e redução de tempo no desenvolvimento deste tipo de projeto (OSÓRIO, 2014).

Um estudo realizado pela J.D. *Power and Associates* apontou que 30% das pessoas têm o interesse de adquirir um veículo autônomo. Devido ao aumento da procura deste tipo de veículos as principais montadoras de carros como General Motors, Ford, Mercedes-Benz, Volkswagen, Audi, BMW, Volvo, Cadillac, entre outras, estão investindo cada vez mais em sistemas que possibilitem o desenvolvimento de veículos autônomos (YVKOFF, 2012, tradução nossa).

Na década de 80 começou a utilização de AGV's, porém com algumas limitações, pois se os mesmos encontrassem um obstáculo na área de trabalho ou fora do âmbito industrial os veículos não fariam nenhuma ação programada. Com o intuito de melhorar neste sentido, criou-se um veículo capaz de interagir com o ambiente por meio de seus sensores e de seu estado atual, ou seja, um veículo com percepção e com um grau maior de inteligência para suas atividades. Com estes avanços na década de 90 surgiram os robôs (SECCHI, 2008).

A palavra robô vem de origem tcheca – *robot* – que tem como significado trabalho forçado (HALFPAP; SOUZA; ALVES, 2007). Os termos "robótica" e "automação" tem um significado técnico preciso. De acordo com o *Robotics and Automation Society* do Instituto de Eletrônica e Engenheiros Elétricos, "Robótica se concentra em sistemas com sensores e atuadores que operam de forma autônoma ou semiautônoma, em cooperação com os seres humanos". A robótica procura, enfatizar a inteligência e adaptabilidade para lidar com ambientes não estruturados. Pesquisa de automação enfatiza a

eficiência, a produtividade, qualidade e confiabilidade, com foco em sistemas que operam de forma independente, muitas vezes em ambientes estruturados durante longos períodos, e na estruturação explícita de tais ambientes (ESTADOS UNIDOS DA AMÉRICA, 2013 tradução nossa).

A robótica vem crescendo nos últimos anos em diversas áreas de aplicação, com isso a mesma vem contribuindo para a economia, meio ambiente e o bem-estar dos seres humanos, isto indica que nos próximos anos a este meio poderá estar tão presente quanto à informática e a comunicação (BUENO et al, 2014).

Uma dos campos explorados pela robótica é a área de robôs autônomos, estes robôs devem realizar tarefas independentemente de como serão realizadas desde que as realizem, e sem que haja há intervenção humana para isto. Mas para que isto ocorra, estes robôs devem ser equipados com sensores e atuadores, e controlados por meio de um sistema computacional (OTTONI; LAGES, 2003).

Os robôs podem ser classificados em três classes principais: industriais; médicos e móveis.

Os industriais possuem uma estrutura mecânica com articulações, que permite adaptar-se para efetivar os comandos necessários passados pelo microprocessador.

Os médicos não estão apenas em salas de cirurgia como se pensa normalmente, os mesmos auxiliam nas tarefas mais árduas, também estão classificados nesta categoria, as próteses inteligentes, que ajudam a realizar funções do próprio sistema nervoso do usuário emitindo por meio dos sinais nervosos ou musculares (SECCHI, 2008).

Já os móveis podem ser subdivididos em três categorias:

- a) não autônomos: também são conhecidos como automatizados e não possuem inteligência própria, ou seja, seu funcionamento se dá através de usuários;
- b) semiautônomos: possui alguma autonomia devido a inteligência integrada em seu mecanismo de controle, porém não permite ter um comportamento adaptativo e auto programado;

- c) totalmente autônomos: seu mecanismo é totalmente inteligente, capaz de tomar decisões de comportamento sem que haja a intervenção humana (MCKERROW, 1991 apud SILVA, 1999).

2.1.1 Robô móvel autônomo

O campo da robótica abrange uma ampla área de tecnologias em que a inteligência artificial é incorporada em máquinas físicas, assim criando sistemas com capacidades muito superiores aos componentes de núcleo sozinho. Tais sistemas robóticos são então capazes de desempenhar tarefas que são realizáveis por máquinas convencionais, por seres humanos que trabalham com ferramentas convencionais. A capacidade de uma máquina para mover por si só, isto é, autônoma, é uma capacidade que abre uma variedade de aplicações que são exclusivamente adequados para sistemas robóticos (BEKEY et al, 2006, tradução nossa).

A robótica móvel possibilita trabalhar com veículos autônomos ou semiautônomos, desta forma este modelo de robótica aumenta as possibilidades da robótica clássica. O modelo dá importância aos problemas complexos dos locais de atuação. Os controles destes veículos provem dos dispositivos de *hardware* e *software* desenvolvidos por especialistas das áreas de engenharia elétrica, engenharia mecânica e ciência da computação (DUDEK; JENKIN, 2000, tradução nossa).

Os robôs móveis da atualidade possuem diversos tipos de sensores para auxiliar na interação com o meio ambiente, como, por exemplo: sensores de temperatura, acelerômetro, infravermelho, toque, umidade, sonar, entre outros, com isso é possível delimitar qual será a estratégia com maior eficiência para uma determinada tarefa (SILVA, 2008).

Segundo Jacobo (2001), a seguir três estratégias de decisão e percepção utilizadas pelos robôs:

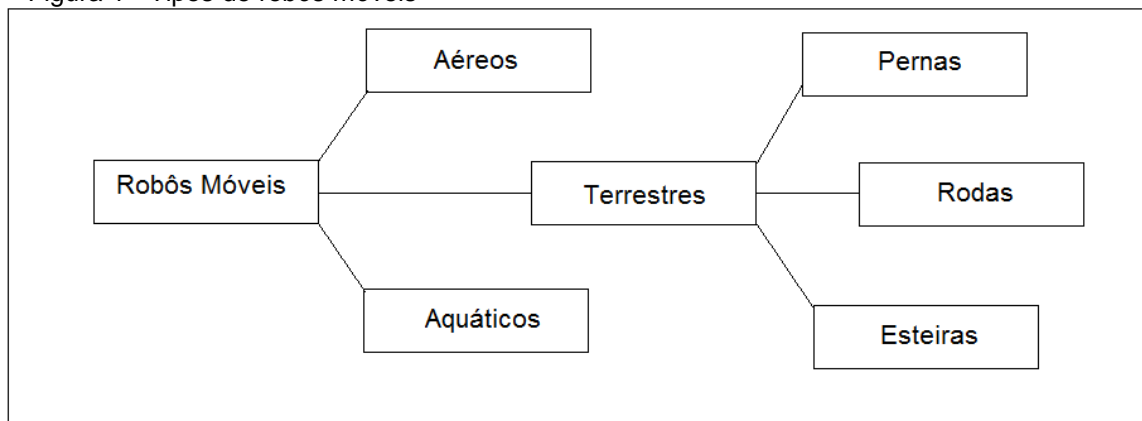
- a) estratégia reativa: baseia-se em ações realizadas por meio de leituras de sensores. Este tipo de controle recebe as leituras do entorno dos sensores e realiza uma determinada ação. Este tipo

- de controle possui uma velocidade de resposta, tem um baixo custo computacional e pode trabalhar com regras e incertezas;
- b) estratégia deliberada: baseia-se nas informações previamente processadas do ambiente e suas tarefas são antecipadamente determinadas para a realização;
- c) estratégia híbrida é uma junção das estratégias anteriores.

2.1.2 Tipos de robôs móveis

Segundo De Pieri (2002), podem ser divididos em três tipos de anatomia:

Figura 1 - Tipos de robôs móveis



Fonte: Adaptado de De Pieri (2002).

Na figura 1 pode-se observar que os robôs móveis estão presentes em diversas áreas de pesquisas como: (DE PIERI, 2002).

- a) robôs aéreos: geralmente são utilizados para a verificação de grandes campos por meio de câmeras acopladas no mesmo, os mais utilizados para realizar este tipo de tarefa são os *Lighter-Than-Air* (LTAs) ou aeromodelos;
- b) robôs aquáticos: em geral são equipados com balões e propulsores, deste modo o veículo pode ficar submerso por alguns metros no fundo de ambientes com água, eles também podem ser equipados com patas para a sua locomoção;

c) robôs terrestres: são facilmente encontrados, e sendo possível utilizar um dos três tipos de atuadores:

- esteiras: mais utilizados em terrenos irregulares devido as pedras existentes ou por ser um terreno fofo, porém sua principal desvantagem, é o gasto de energia que o mesmo precisa para realizar a movimentação de sua esteira de locomoção;
- pernas: é utilizado em terrenos específicos, como por exemplo, locais com escadas. A maior dificuldade neste tipo de projeto é o controle das pernas, que possuem dois graus de liberdade, outro fator importante é o custo para projetos como este, pois cada atuador “perna” necessita no mínimo de dois motores para seu pleno funcionamento;
- rodas: são os mais comuns entre os robôs terrestres, devido a sua facilidade de ser criado, e não necessitar de um *hardware* tão complexo quanto aos demais modelos. Porém sua principal desvantagem de utilização é o seu desempenho em terrenos irregulares, que por sua vez pode não alcançar o desempenho esperado. Geralmente suas rodas devem possuir um raio maior ou igual aos obstáculos que ele irá enfrentar.

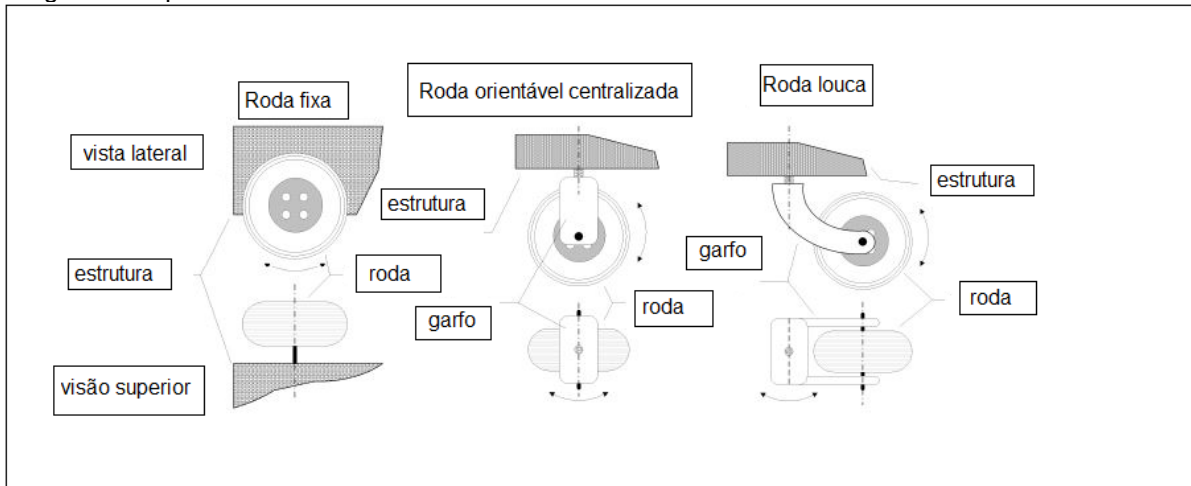
Segundo Secchi (2008), as rodas podem ser classificadas em dois tipos as suecas e as convencionais. As convencionais por sua vez podem ser classificadas em três tipos que são fixa, orientável centralizada e orientável não centralizada (roda castor).

- a) roda fixa: este tipo de roda é encontrado fixado na estrutura do robô, e geralmente é utilizado para dar tração ao mesmo;
- b) roda orientável centralizada: seu movimento por meio de um eixo que corta o seu centro. Este tipo geralmente é utilizado como roda de tração-direção ou apenas de direção;
- c) roda orientável não centralizada (roda louca): conhecida como roda louca ou roda castor é uma modelo cujo seu movimento se dá através de um eixo que não corta o seu centro, mas sim através de um eixo vertical, sua função na estrutura é

proporcionar a estabilidade ao robô e também para lhe dar direção.

Na figura 2 tem-se a estrutura de três tipos de rodas para locomoção de veículos autônomos.

Figura 2 - Tipos de rodas

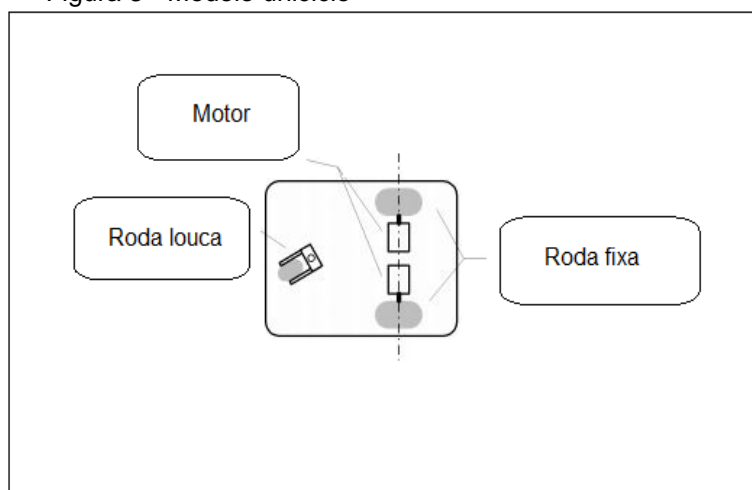


Fonte: Adaptado de Secchi (2008).

Segundo Secchi (2008), robôs unicyclos são considerados por pesquisadores os melhores modelos para a criação de protótipos, por possuírem uma cinemática simples. Este tipo de modelo unicyclo tem sua composição por duas rodas fixas na sua estrutura com controle de tração independentes e uma roda louca para dar estabilidade em sua estrutura física.

A seguir apresenta-se na figura 3 o modelo estrutural de robôs unicyclo, bem como as fixações de cada peça integrante.

Figura 3 - Modelo unicyclo



Fonte: Adaptado de Secchi (2008).

2.2 CINEMÁTICA

É responsável por descrever os movimentos realizados pelo corpo. Geralmente é conhecido como “geometria dos movimentos”. Algumas aplicações de cinemática incluem o projeto de cames², engrenagens, mecanismos, entre outros elementos para as máquinas. A movimentação de uma partícula ou corpo pode ser realizada de várias maneiras, o que difere são de como a informação é coletada e a experiência para realizar a escolha da melhor forma para realizar seu deslocamento. Uma partícula é um corpo com dimensões pequenas comparadas ao tamanho de sua trajetória, deste modo pode-se dizer que o movimento da partícula como a de um ponto (MERIAM; KRAIGE, 2009).

A cinemática tem como principal função fazer com que seja compreendido o comportamento mecânico de um robô, para poder assim realizar com mais eficácia a sua parte de movimentação. Para o entendimento dos movimentos de um robô, precisa-se ter como base a contribuição de cada roda para a sua movimentação, ou seja, cada roda tem influência direta na movimentação final de um robô. Deste modo pode-se afirmar que cada roda colocará restrições em cada movimentação feita pelo mesmo. Deve-se levar

² Parte de uma roda ou eixo giratório ressaltada e projetada para transmitir um movimento alternado ou variável a um outro mecanismo

em questão para restrições, a massa e forças que atuam sobre o protótipo. Um exemplo que se pode obter sobre este assunto é o centro gravitacional alto, que limita o raio das curvas com velocidades altas, pois a uma grande chance do mesmo vir a capotar (BRAGA, 2014).

Estão sendo realizados há algum tempo estudos referentes aos manipuladores robóticos, que por sua vez existem alguns modelos que são mais complexos que os próprios robôs móveis. Um exemplo claro dessa situação é a utilização de várias articulações ao mesmo tempo enquanto um robô pode ser uma máquina simples que possui um controle diferencial (BRAGA, 2014).

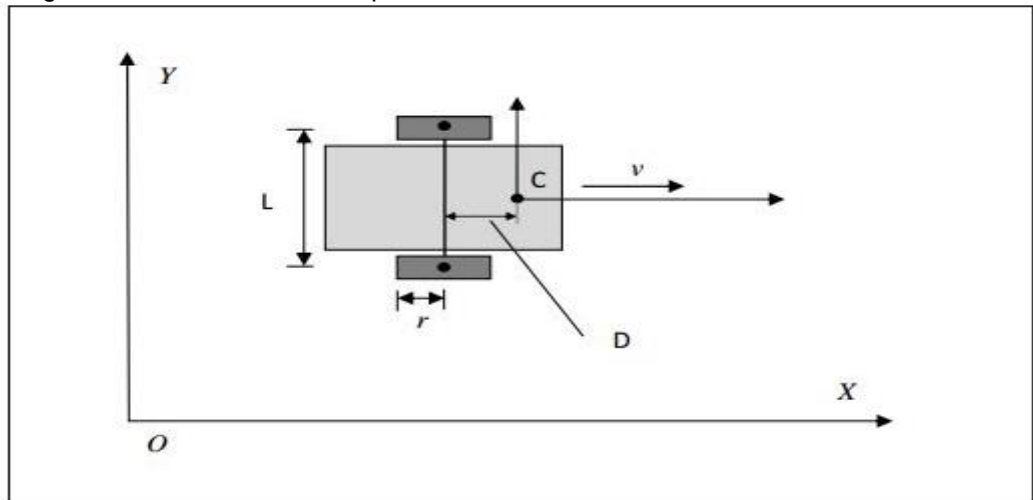
2.2.1 Modelos cinemáticos e restrições

Segundo Braga (2014), cada roda individualmente possui uma contribuição e ao mesmo tempo gera restrições para a sua movimentação. Portanto deve-se observar as rodas presas ao chassi do robô, e assim verificar suas restrições, que devem ser combinadas para que haja uma modelação das restrições globais encontradas no mesmo. Portanto as forças realizadas em cada roda e suas restrições precisam ser colocadas em relação a um referencial claro e fixo.

Santos (2015) um veículo com tração individual de cada roda tem-se elas montadas no mesmo eixo. A sua posição global é apresentado na imagem a seguir no plano $\{X,0,Y\}$.

Na figura 4 pode-se verificar o comportamento da cinemática de um robô uniclo, ou seja, como funciona o seu comportamento em relação aos seus eixos de movimentação.

Figura 4 - Modelo cinemático para robô uniciclo



Fonte: Santos (2015).

Na formula 1 pode-se observar o cálculo utilizado para obter a velocidade média de um corpo, (MERIAM; KRAIGE, 2009).

$$v = \frac{\Delta s}{\Delta t} = S$$

(1)

Onde $\Delta S = S - S_0$ e $\Delta t = t - t_0$

Sendo que:

S → posição final do móvel

S_0 → posição inicial

t_0 → tempo inicial

t → tempo final

O dispositivo robótico deste projeto se encontra no modelo de estratégia reativa e sua categoria pertence à robótica móvel autônoma, sua característica de prototipagem possuem duas rodas motrizes e uma roda louca para permitir seu deslocamento de forma autônoma e de cinemática simples. A captura das informações externas é realizada através de um sensor ultrassônico posicionado na frente do veículo.

2.3 SENSORES

Sensores são componentes elétricos que servem para medir um estímulo externo ao sistema em que estão acoplados, ou seja, funciona como dispositivos de entrada, e suas informações como saídas com base nas medições feitas pelo mesmo. Nem toda entrada é um sensor, porém a maioria delas utilizam-nas para realizar suas determinadas tarefas como, por exemplo, o teclado de um computador, ele não é um sensor, mas utiliza de sensores para saber qual tecla esta sendo pressionada no momento (KARVINEN; KARVINEN, 2014).

Segundo Souza Filho (2009), os sensores utilizados na robótica são classificados da seguinte forma:

- a) sensores de tato: reage a força exercida sobre o objeto. Alguns desses sensores são capazes de medir o nível de força exercida sobre o objeto de contato;
- b) sensores de proximidade e distância: é utilizado para indicar que o sensor esta localizado próximo de um objeto “obstáculo”, mas antes que ocorra a colisão. Quando é possível verificar a distância entre o sensor e o objeto este dispositivo é conhecido como sensor de distância;
- c) sensores de tipos diversos: são classificados nesta categoria os sensores de temperatura, pressão, gases entre outros tipos de fenômenos e;
- d) sensores de visão de máquina: é utilizado para verificar ambientes e interpretá-las para se orientar. Estes tipos de sensores são realizados pela robótica para realizar inspeções e reconhecimento de áreas.

Os sensores podem ser considerados passivos ou ativos. Os passivos como as câmeras, são considerados observadores de ambientes, ou seja, o seu funcionamento se dá através da captura de sinais fornecidos pelo próprio ambiente. Portanto seu consumo de energia é menor do que os ativos. Já os ativos como os sonares necessitam emitir ondas no ambiente para receber informações sobre o ambiente, por este motivo os sensores ativos

acabam necessitando de um consumo maior de energia em relação aos sensores reativos (RUSSELL; NORVIG, 2013).

Para este projeto utilizou-se de um sensor de proximidade e distância, com propósito de identificar obstáculos no ambiente, o modelo aplicado no projeto foi o HC-SR04.

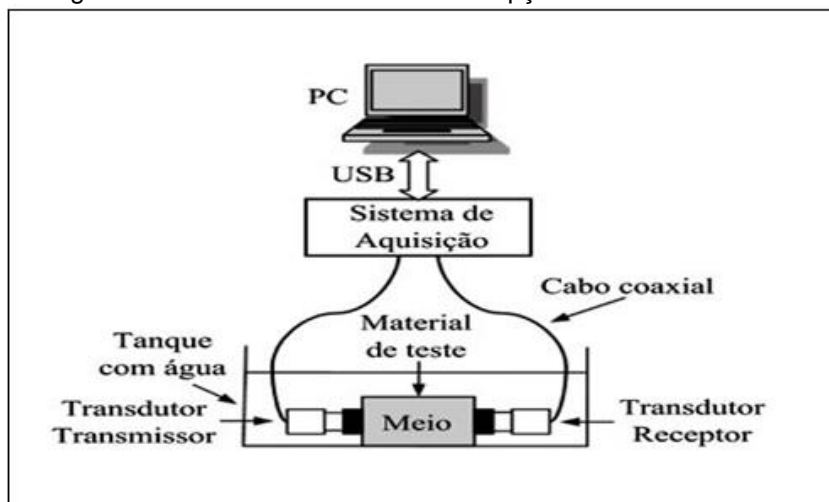
2.3.1 Sensor ultrassônico

O sensor ultrassônico é muito utilizado na área da robótica, sua principal função é emitir um som ultrassônico para detectar a presença de objetos, no qual ao colidir com o obstáculo terá um retorno para o mesmo, isso se dá por meio de ondas emitidas para determinar a distância, o cálculo é baseado no tempo de retorno das ondas ultrassônicas até o sensor. Ele também é conhecido como *Sound Navigation and Ranging* (SONAR), que é utilizado por submarinos para obter a localização de embarcações e também utilizado por morcegos para detecção de suas presas (MCROBERTS, 2011).

Existem diversos tipos de modelos de sensores ultrassom, o que os difere são seus métodos de captura de sinais, deste modo, apresenta-se a seguir os dois modelos mais utilizados.

Portanto na figura 5 apresenta-se a utilização do modelo de transmissão-recepção, que utiliza-se de dois transdutores, um para emitir as ondas ultrassônicas e o outro para receber os sinais emitidos das ondas.

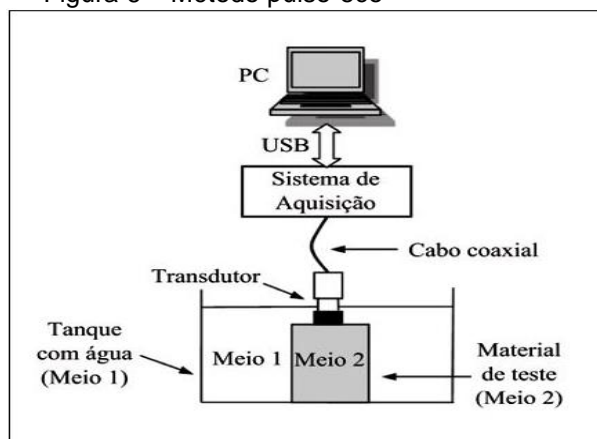
Figura 5 - Método de transmissão-recepção



Fonte: Assef et al. (2009).

Na figura 6 apresenta-se outro modelo de sensor ultrassônico o qual utiliza-se de apenas um transdutor tanto para emitir quanto para receber os sinais das ondas emitidas. Este sinal emitido pelo transdutor terá parte refletida e outra transmitida, deste modo, as ondas refletidas trarão as informações de distância ou velocidade de propagação.

Figura 6 – Método pulso-eco



Fonte: Assef et al. (2009).

2.3.2 Sensor HC-SR04

O sensor HC-SR04 utiliza o método de transmissão-recepção, diferentemente de outros sensores de ultrassom que utilizam o método pulso-eco que usam apenas um transdutor para emitir e receber os sinais das ondas. Já o método transmissão-recepção utilizado HC-SR04 trabalha com dois

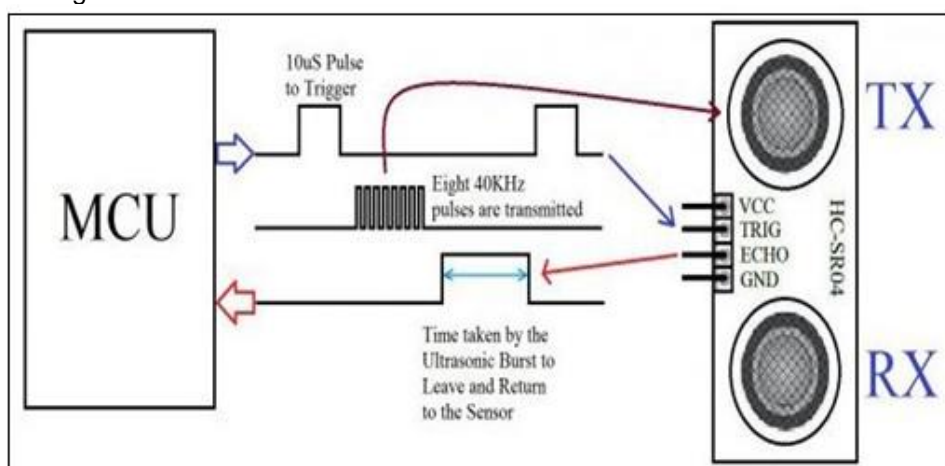
transdutores independentes, isto significa que, quando um pulso elétrico é gerado pelo transdutor emissor o transdutor receptor serve para receber os sinais das ondas ultrassônicas e emitidas (ASSEF et al, 2009).

Possui uma precisão de 3mm em suas medições com um alcance entre 2 cm (centímetros) a 400 cm sem contato com obstáculos, possui transmissor de ultrassom, receptor e circuito de controle. Seu funcionamento básico é:

- a) lançar pelo menos 10 microsegundo (μ S) de alto nível;
- b) seu módulo dispara automaticamente oito 40 kHz para verificação de obstáculos na frente do mesmo, se houver ocorrerá uma ruptura da onda ultrassônica e refletida ao sensor;
- c) se o sinal de volta, por meio de alto nível, o tempo de saída de alta IO duração é o tempo de envio de ultrassom para voltar.
- d) verificar distância = (tempo de alto nível x velocidade do som (340 metro por segundo (m/s)/2 (ELECTFREAKS 2016, tradução nossa)).

Na figura 7 apresenta-se o sensor ultrassônico o qual foi utilizado no projeto, como sua pinagem de alimentação, qual transdutor e responsável por emitir e qual é responsável por receber as ondas sonoras, qual pulso de transmissão são emitidas as ondas entre outras especificações do dispositivo.

Figura 7 – Sensor ultrassônico HC-SR04



Fonte: Electrosome (2014).

Este tipo de sensor tem como principal vantagem a flexibilidade de reconhecimento de objetos de vários tipos de materiais e tamanhos (PEREIRA, 2011).

Segundo Wolf et al. (2009), a robótica possui diferentes dispositivos de *hardware* embarcados para realizar tarefas projetadas. Portanto os principais dispositivos para que isso ocorra são os sensores e atuadores acoplados no robô.

2.3.3 Atuadores

Brugnari e Maestrelli (2010) afirmam que os atuadores realizam comandos manuais ou automáticos com base nas entradas ou critérios a serem seguidos. Os atuadores são responsáveis por transformar sinais elétricos em movimentos capazes de alterar o ambiente que habitam. Os atuadores são considerados transdutores como os sensores (LONGO, 2015). Segundo De Pieri (2002), seu funcionamento se dá através de sua junção com outros dois tipos de sistemas que são:

- a) sistemas de transmissão e redução: são elementos capazes de realizar a movimentação de um lado para outro e alterar a velocidade do seu acionamento. Ainda os mesmos devem trabalhar de forma harmoniosa para que tanto a parte de transmissão, quanto à parte de redução possam realizar seus objetivos sem interferir no funcionamento de partes do robô;
- b) sensores de posição e velocidade: este sistema é responsável por ler as informações de velocidade e posição de cada um dos atuadores de movimentação do robô. Portanto este tipo de sistema é primordial para o funcionamento do robô, pois é através das informações captadas deste sistema que o robô irá gerar suas decisões.

Segundo Rosário (2005) e Thoamazini; Albuquerque (2011) existem vários tipos de atuadores aplicados a robótica, como:

- a) atuadores hidráulicos: este tipo de atuador tem seu funcionamento de forma linear ou rotativo e se dá por meio de fluídos como óleo ou água. Geralmente são encontrados em robôs que trabalham com uma grande capacidade de carga;
- b) atuadores pneumáticos: este tipo de atuador é semelhante ao anterior, porém seu funcionamento se dá por meio de ar comprimido, e sendo pouco utilizado no segmento da robótica móvel, devido a sua baixa precisão nos movimentos e a alta compressibilidade. Este tipo encontra-se em robôs com movimentos rápidos, mas com pouca precisão. São considerados atuadores pneumáticos pela robótica basicamente dois tipos os cilindros pneumáticos e motores pneumáticos de aletas rotativas pistão axial;
- c) Estes dois modelos apresentados são evitados ao máximo, pois os mesmos acabam gerando muita manutenção devido às instalações de bombas, válvulas, compressores entre outros para que seu funcionamento ocorra de forma correta. Desta forma estes atuadores ficam em muita desvantagem em relação aos atuadores elétricos para robôs móveis;
- d) atuadores elétricos: este tipo de atuador leva como vantagem em relação aos demais atuadores o controle de velocidade calculada e controle de posição. São considerados atuadores elétricos motores de corrente alternada (AC) síncronos e assíncronos, motores de passo, motores de corrente (DC), servomotor (DC), entre outros.

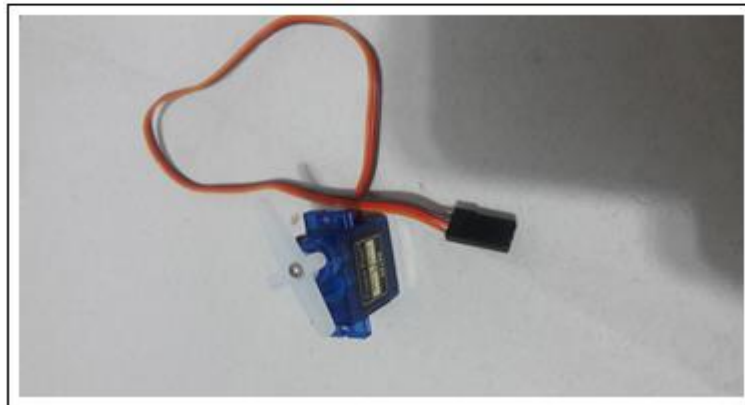
Neste trabalho foram utilizados dois tipos de atuadores elétricos um servomotor (DC) e dois motores (DC) para realizar o funcionamento do protótipo.

O atuador Servomotor (DC): é considerado como um motor (DC), porém com a incorporação de alguns componentes, tornando-o assim, fácil seu uso. Este dispositivo vem equipado geralmente com engrenagens, limitador de eixo, terminais de três pontas para controle de posição, potenciômetro para

voltar à posição e pequenos circuitos integrados para o controle de posição (DE PIERI, 2002).

Na figura 8 apresenta-se o servomotor utilizado neste projeto para realizar a movimentação do sensor com características citadas anteriormente.

Figura 8 – Servomotor

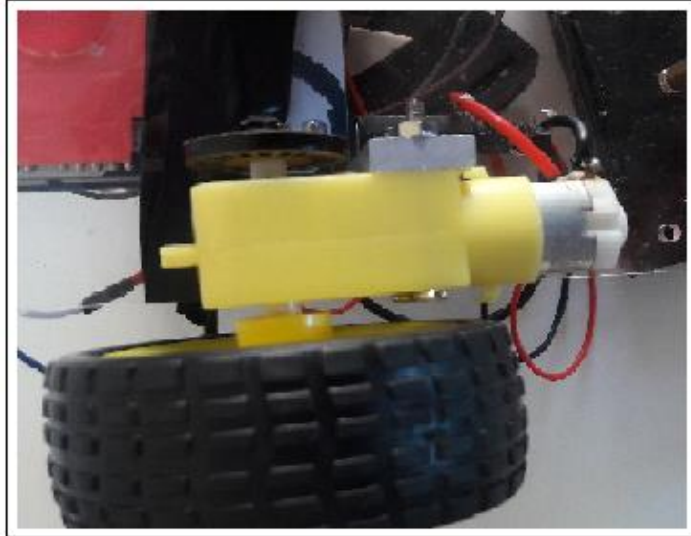


Fonte: Do autor.

Motor (DC): necessita de uma fonte de tensão contínua para realizar o seu funcionamento. Sua estrutura compõe-se de duas partes o indutor e o induzido. O indutor tem como finalidade fornecer o campo magnético para a realização do movimento. Já o induzido tem seu funcionamento ao campo magnético criado por ele mesmo e pelo indutor. Este tipo de motor é os mais utilizados na área da robótica devido a necessidade de uma fonte primaria como pilhas ou baterias e seu fácil controle de manipulação (DE PIERI, 2002).

Na figura 9, tem-se um motor (DC) utilizado no protótipo para realizar sua locomoção através de rodas.

Figura 9 – Motor DC utilizado no protótipo



Fonte: Do autor.

3 MICROCONTROLADORES

Em meados dos anos 70 iniciou-se o uso de microprocessadores em computadores para possuir um melhor desempenho no processamento de dados. O microprocessador Intel foi quem proporcionou este tipo de tecnologia, devido a preocupação em cada vez mais aprimorar seu processamento através desses componentes. A partir da estrutura de um microprocessador e seus periféricos, então um novo componente nasceu o microcontrolador, ou seja, um componente que suportava todo um sistema equivalente a um microprocessador e seus periféricos (MARTINS, 2005).

Um microcontrolador possui acesso aos periféricos de entrada e saída e a memória através de seu processador (OKI; MANTOVANI, 2013). Segundo Martins (2005) eles têm como principal função gerenciar internamente equipamentos eletrônicos, facilitar na manutenção e diminuir o tamanho dos equipamentos, deste modo pode-se dizer que eles estão presentes em quase tudo que envolva eletrônica.

Os microcontroladores têm agregado inúmeras funcionalidades, tais como: gerador interno independente de *clock*; memória *Static Random Access Memory* (SRAM), *Electrically-Erasable Programmable Read-Only Memory* (EEPROM) e *FLASH*; conversores analógico/digital (A/D), digital/analógico (D/A); vários temporizadores/contadores; comparadores analógicos; PWM; diferentes tipos de interface de comunicação, incluindo (universal serial bus) USB, **Universal Asynchronous Receiver/Transmitter** (UART), *Inter-Integrated Circuit* (I²C), *Controller Area Network* (CAN), *Serial Peripheral Interface* (SPI), *Joint Test Access Group* (JTAG); relógios de tempo real; circuitos para gerenciamento de energia no chip; circuitos para controle de *reset*, alguns tipos de sensores; interface para (*liquid crystal display*) LCD; e outras funcionalidades de acordo com o fabricante. (OKI; MANTOVANI,2013, p.3).

Os principais aspectos que diferem os diversos tipos são a velocidade de processamento, a quantidade de pinos de I/O, quantidade de

memória interna para o armazenamento de dados e instruções dos programas, sua arquitetura, modelos e quantidade de periféricos, método de alimentação e as instruções que são disponibilizados nos seus circuitos internos (MARTINS, 2005).

Os microcontroladores ATmega328 da fabricante Atmel utilizados pelo Arduino Uno é composto basicamente de *Central Processing Unit* (CPU), memórias (Random Access Memory (RAM), *read-only memory* (ROM), EEPROM), pinos analógicos, pinos digitais de entrada e saída, conversor de sinal analógico para digital, *pinos Pulse Width Module* PWM, entre outros (ARDUINO, 2016, tradução nossa).

3.1 ARDUINO

O Arduino teve a sua origem no ano de 2005, na cidade Ivrea, na Itália, onde dois professores pesquisadores estavam a procura de uma ferramenta de baixo custo, e pudesse auxiliar o desenvolvimento de projetos das turmas de ambos. Há turma de design do professor Massimo Banzi, e a turma do professor sueco David Guatielles, da Universidade de Malmö. Desta forma, os dois chegaram a um acordo de produzir algo, que viesse a auxiliá-los a encontrar uma solução, assim resolveram desenvolver um microcontrolador de menor custo e de fácil uso e acesso para seus alunos de design e artes, deste modo este microcontrolador surgiu (EVAN; NOBLE; HOCHENBAUM, 2013).

Então David Guatielles desenhou a placa, e David Milles aluno de Massimo programou o *software* para que a placa funcionasse, Gianluca Martino engenheiro local foi contratado por Massimo e aceitou produzir duzentas placas inicialmente (EVANS; NOBLE; HOCHENBAUM, 2013).

As duzentas placas foram facilmente vendidas, e devido a procura de pessoas de outras áreas, teve-se a necessidade de produzir mais unidades para atender a procura, já que seu sistema era de fácil utilização. A sua comercialização era feita em forma de *kit* para facilitar no desenvolvimento de projetos realizados pelos alunos (EVANS; NOBLE; HOCHENBAUM, 2013).

Existem outras plataformas com o mesmo propósito do Arduino, porém não possui as mesmas características e facilidade de utilização e de preço acessível como o mesmo (WERNECK, 2009).

Existem muitos tipos de Arduino e versões como Nano, Mini, *Bluetooth*, outro tipo é o Mega 2560 que por sua vez possui um número maior de pinos de entrada/saída e uma memória maior. As novas placas liberam 1,5 kilobyte (Kb) de memória *flash* que por sua vez tende a ter uma inicialização mais rápida devido ao seu novo *bootloader*, o *Optiboot* (MCROBERTS, 2011).

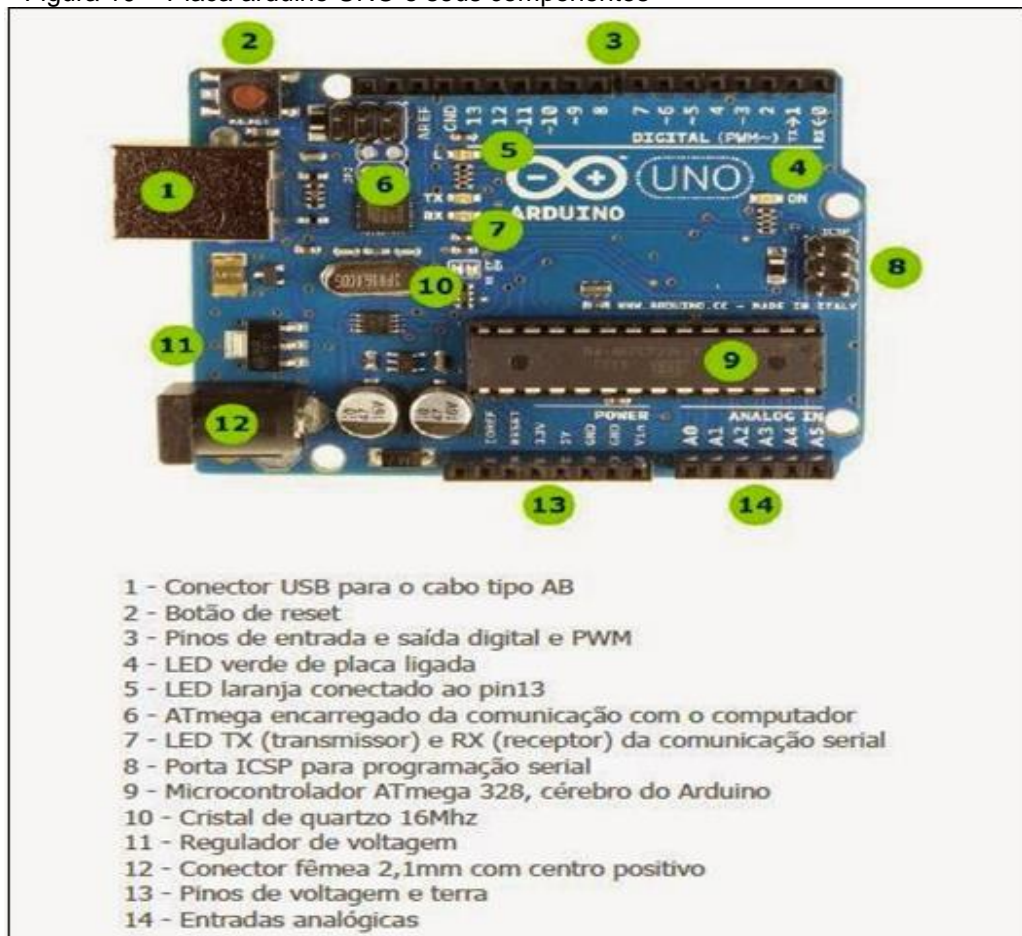
Este circuito é composto por um microcontrolador e seus demais componentes, todos acoplados em uma única placa, onde permite a utilização para os mais diversos fins já que sua plataforma é totalmente livre e não há necessidade de ter um conhecimento avançado em eletrônica (MUNDO ROBÓTICA, 2015).

O Arduino é um computador de pequeno porte, onde seu processamento se dá por meio das informações que chegam por meio das portas de entrada e saída do dispositivo, desta forma pode-se programá-lo para fazer com que os dispositivos conectados a ele consigam interagir com o meio ambiente através de seu *software* e *hardware*, por este motivo sua plataforma é conhecida com computação física ou embarcada (MCROBERTS, 2011).

A sua placa é composta pelos seguintes componentes um microprocessador Atmel AVR, um regulador linear de 5 V, um oscilador (um relógio que envia sinais de frequência especificada que permite seu funcionamento na velocidade correta), pinos de entrada e saída do microcontrolador para conexão em outros circuitos ou sensores. Pode também possuir uma saída USB, para alimentação da placa, recuperação de dados e *upload* (MCROBERTS, 2011).

Na figura 10 apresenta-se uma placa arduino UNO identificando numericamente as porta de entrada e saída e componentes encontrados no dispositivo.

Figura 10 – Placa arduino UNO e seus componentes



Fonte: Eletrônica (2014).

Segundo Evans, Noble e Hochenbaum (2013) a principal diferença entre a versão Uno e suas antecessoras é a utilização de um microcontrolador ATmega8U2 que funciona como um conversor USB para serial, deste modo acabou substituindo o *chipset Future Technology Devices International (FTDI)* que era utilizado nas versões anteriores.

A placa Uno possui inúmeras vantagens em relação a sua antecessora a *Duemilanove*, por não utilizar mais o *chip* FTDI e sim utilizar o ATmega8U2, programado como um conversor USB para serial enquanto o chip FTDI usado na *Duemilanove* leva a porta USB para serial. Portanto deste modo e em primeiro lugar o ATmega tem um custo inferior ao chip FTDI, portanto isso acaba afetando no preço das placas tornando-as mais baratas, e em segundo e principalmente mais importante ele permite que o *chip* USB possua seu *firmware* atualizado, para que apareça no computador como um dispositivo

qualquer, um exemplo é o teclado ou *mouse*. Com essas melhorias ele adquiriu muitas novas possibilidades de ser utilizado no mercado. Mas, com toda a mudança possui seus lados positivos, também acabam possuindo seus lados negativos, ou seja, com a mudança do *chip* USB para este tipo ficou muito mais difícil para os fabricantes de clones criarem clones do Arduino Uno (MCROBERTS, 2011).

Segundo Evans, Noble e Hochenbaum (2013) outra diferença é a segurança que ele proporciona devido a sua tensão integrada de 3,3 V, com isto ocorre uma maior estabilidade de algumas proteções que nas versões anteriores possuíam problemas.

A *Integrated Development Environment* (IDE) utiliza a linguagem *Processing*, que é baseada em C/C++, que também além de ser *open source*, sua interface é amigável, sua plataforma possui uma camada simples de *software* inserida na placa, que é um *bootloader*, deste modo dispensa-se o uso de programadores para o *chip*, já que utiliza a família AVR do fabricante ATMEL, no qual não exige compiladores ou *hardware* adicional (SOUZA et al, 2011).

Sua estrutura é baseada em dois blocos de funções principais, que podem receber outras funções internamente (SILVEIRA, 2012). Segundo o autor ainda esses blocos são constituídos de linguagem C/C++ e suas funções são:

- a) *void setup()*: utilizada para as portas de comunicação serial, onde ocorre a preparação dos pinos do Arduino e inicialização dos valores;
- b) *void loop()*: principal função onde através dela é possível executar as demais funções. Sua execução ocorre logo após a função *void setup()* e permanecerá em execução até que ocorra um comando de parada.

O *software* é de fácil utilização para iniciantes, e também suficientemente flexível para usuários com mais experiências. Ele pode ser utilizado nos sistemas operacionais *Mac*, *Windows* e *Linux* (ARDUINO, 2016, tradução nossa).

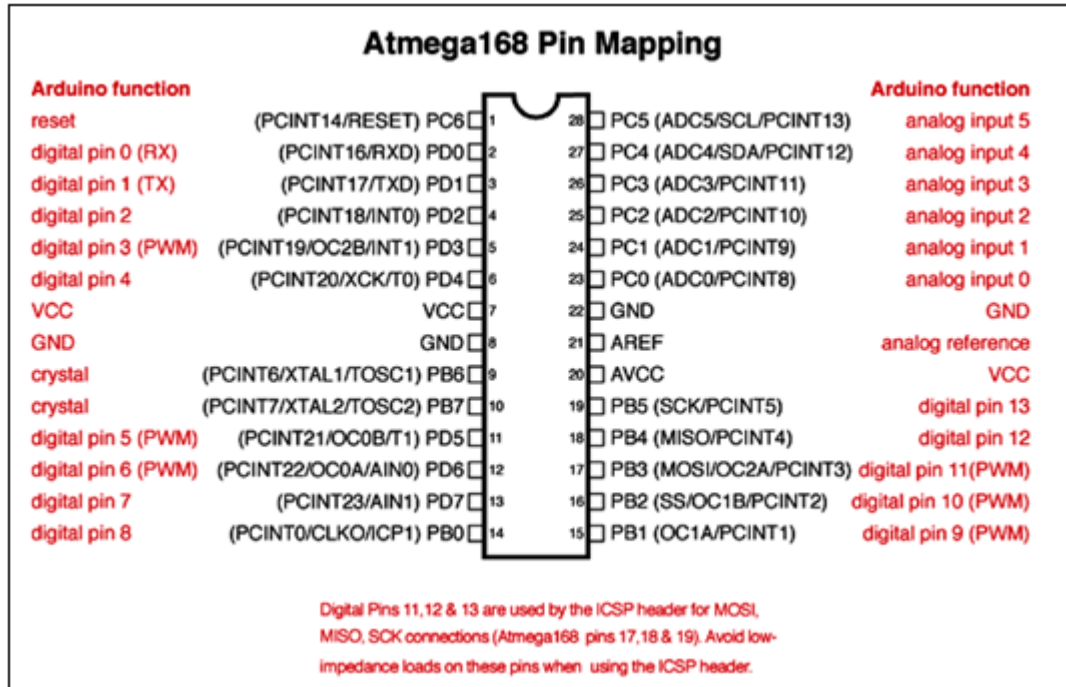
A seguir nas subseções será apresentado algumas das principais características do microcontrolador arduino UNO como: entrada e saída, memória, alimentação. Também será abordado sobre a *shield* L293D *driver* ponte H como, funcionamento interno, possibilidade de conexões com outros dispositivos entre outras características do mesmo.

3.1.1 Entrada e saída

O ATmega8,168 e o ATmega328P possuem a mesma estrutura física, ou seja, são totalmente idênticos isso pode ser notado no mapeamento mostrado na figura abaixo (ARDUINO, 2016, tradução nossa).

Na figura 11 apresenta-se o mapa de pinagem do Atmega168 no qual é idêntico ao Atmega328 o mesmo utilizado no arduino UNO, cujo será utilizado no projeto.

Figura 11 - Mapa de pinagem Atmega168 e Atmega328P



Fonte: Rodrigues (2016).

Utilizando as funções *pinMode()*, *digitalWrite()*, e *digitalRead()* é possível usar qualquer um dos quatorze pinos da placa Uno de forma a definir

o seu funcionamento de entrada ou saída, que trabalham em 5 V. Para que não haja danos permanentes ao microcontrolador as portas de entrada e saída não podem ultrapassar de 40 miliampere (mA). Em média cada pino recebe ou fornece 20mA como padrão de funcionamento e possui um resistor *pull-up* interno de ohm 20-50 kilo (k) (ARDUINO, 2016, tradução nossa).

O Arduino Uno tem suas entradas analógicas rotuladas de A0 à A5 nas quais recebem 10 bits de resolução. Por padrão elas medem 5 V, embora seja possível mudar utilizando a função *analogReference()* e o pino *executive real estate finance* (AREF) (ARDUINO, 2016, tradução nossa).

Alguns pinos tem suas funções especializadas:

- a) serial: 0 (RX) e 1 (TX). Utilizado para receber (RX) e transmitir dados seriais (TX) TTL. Estes pinos são ligados aos pinos correspondentes do chip de ATmega8U2 série USB-TTL.
- b) interrupções externas: 2 e 3. Estes pinos podem ser configurados para disparar uma interrupção por um valor baixo, uma borda de subida ou queda, ou uma mudança de valor.
- c) PWM: 3, 5, 6, 9, 10, e 11. Fornecer saída PWM de 8 bits com a função *analogWrite ()*.
- d) SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estes pinos suportam comunicação SPI usando a biblioteca SPI.
- e) (Light Emitting Diode) LED: 13. Há *um built-in* LED impulsionado por pino digital 13. Quando o pino é de alto valor, o LED está ligado, quando o pino é baixo, ele está fora (desligado).
- f) TWI: A4 ou SDA pinos e A5 ou pino SCL comunicação TWI suporte usando a biblioteca *Wire* (ARDUINO, 2016, tradução nossa).

O PWM é uma técnica utilizada para obter dados analógicos por meios digitais. A maioria dos pinos possuem frequências de sinais aproximadas de 490 Hertz (Hz), porém na placa arduino Uno os pinos 5 e 6 possuem uma frequência maior aproximadamente 980 Hz (ARDUINO, 2016, tradução nossa).

Na figura 12 apresentam-se algumas configurações técnicas do arduino Uno, como por exemplo: peso, quantidade de pinos de entrada analógica, tensão, entre outras características do dispositivo.

Figura 12 - Especificações Técnicas do Arduino Uno

Microcontrolador	ATmega 328P
Tensão operacional	5 V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limite)	6-20V
Digital I/O Pins	14 (dos quais 6 fonecem saída PWM)
PWM Digital I/O Pins	6
Pinos de entrada analógica	6
Corrente Contínua (CC) por I/O Pin	20 mA
Corrente DC 3.3V para Pin	50 mA
Memória Flash	32 KB dos quais 0,5 KB é utilizado por bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)a
Velocidade do relógio	16 Megahertz (MHz)
Comprimento	68,6 (milímetros)mm
Largura	53,4 mm
Peso	25 gramas (g)

Fonte: Arduino (2016).

3.1.2 Memória arduino Uno

O UNO utiliza o ATmega328 que possui 32 KB de memória e 0,5 KB do mesmo é utilizado pelo *bootloader*, também possui 2 KB de memória SRAM e por último 1KB de EEPROM que pode ser controlada pela biblioteca EEPROM (ARDUINO, 2016, tradução nossa).

3.1.3 Alimentação

Segundo Basconcello Filho (2014), o bloco de alimentação deve possuir uma tensão de 7 V à 35 V e sua corrente mínima de 30mA, deste modo

acaba ocorrendo uma filtragem pela fonte para depois regular a tensão para que as duas portas de saídas recebam 5 V e 3,3 V.

Segundo Evans, Noble e Hochenbaum (2013), o cabo USB fornece energia para alimentação da placa do Arduino de 5 V. Essa alimentação ocorre através do ligamento do cabo USB a um computador e a sua placa. Assim possibilitará realizar algumas experiências básicas como, por exemplo, piscar LEDs.

A sua energia pode ser de um adaptador corrente alternada-corrente continua (CA-CC) ou a bateria. Pode-se também ligá-lo através de um *plug* de 2,1 mm de centro positivo à tomada de alimentação encontrada no tabuleiro (ARDUINO, 2016, tradução nossa).

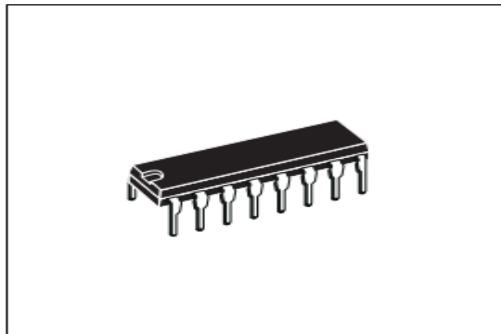
A placa pode funcionar com uma fonte externa de 6 a 20 V. No entanto, se o fornecimento for menor que 7 V o pino de 5 V pode fornecer uma menor quantia de energia e assim deixando-o instável. Mas se ultrapassar mais do que 12 V poderá aquecer a placa e danificá-la, deste modo é recomendado o uso entre 7 a 12 V (ARDUINO, 2016, tradução nossa).

3.1.4 *Shield* L293D driver ponte H

O dispositivo possui capacidade de corrente de saída de 600 mA por canal e suporta de 4,5 à 36 v, está *shield* suporta picos de 1.2 A(ampere), e também contém proteção contra altas temperaturas, sua tensão fica entre 0 à 1,5 v. É recomendado em aplicações que necessitem de ligações entre dispositivos com frequência de até 5 kHz. O mesmo é integrado de alta tensão, e contém *drive* de quatro canais que aceitam cargas indutivas como relés solenóides, DC e motores de passo. Sua estrutura está equipada com uma porta de alimentação, e duas pontes para cada par de canais. O L293D é montado em uma linha de 16 pinos e sua estrutura de plástico, dentre esses pinos, 4 deles são conectados entre si localizados centralmente com função de dissipar a calor gerado (STMICROELECTRONIC, 2003, tradução nossa).

Na figura 13 apresenta-se o desenho do L293D que contém 16 pinos 8 cada lado utilizado na ponte H.

Figura 13 - L293D utilizado na ponte H

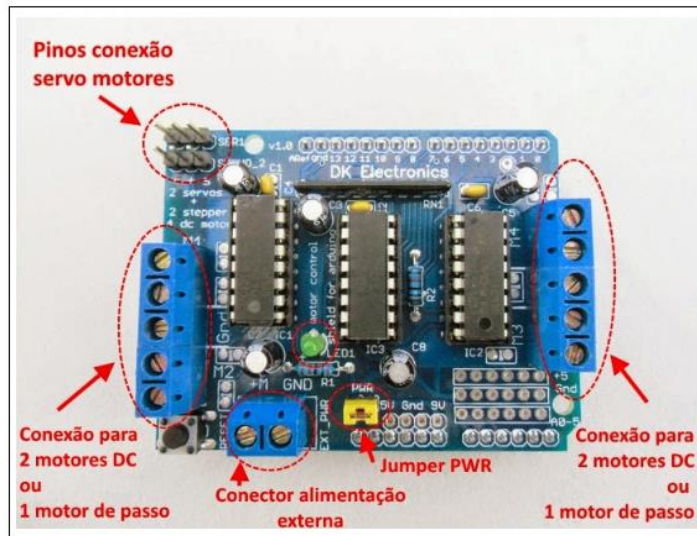


Fonte: StmicroelectroniC (2003)

Segundo Arduino e Cia (2014), a ponte H é adaptável para as placas de arduino Uno e arduino Mega, é composta por dois chips L293D e um CI 74HC595. Está *shield* utiliza os pinos 11, 3, 5 e 6, além dos pinos 4,7, 8 e 12 para poder controlar motores de passo e motores DC, também são utilizados os pinos 9 e 10 para realizar o controle de servomotores. A mesma dispõe de pinos analógicos e digitais para realizar conexões, para os sinais analógicos encontram-se os pinos de A0 até A5 que servem também de pinos digitais de 14 a 19. Encontram-se disponíveis também os pinos 2 e 13.

Na figura 14 apresenta-se as conexões disponíveis para os motores, servomotores e alimentação, a mesma apresenta a localização da luz de led para indicar que existe alimentação, já o jumper PWR deve-se ser retirado quando da existência de alimentação externa.

Figura 14 - Shield L293D Driver Ponte H



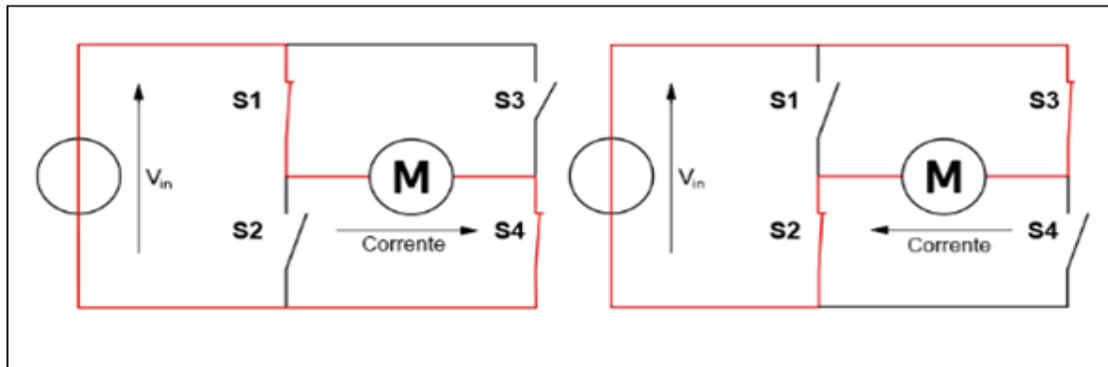
Fonte: Arduino e Cia (2014).

Segundo Arduino e Cia (2014), a ponte H possui um conceito simples de um circuito elétrico paralelo, onde um motor é controlado por quatro chaves e uma fonte de energia. Para realizar este procedimento de chaveamento o L293D utiliza transistores e diodos que possui internamente.

Na figura 15 pode-se observar como é realizado o procedimento de chaveamento para controlar um motor através da ponte H. No primeiro caso, o sentido da corrente será da esquerda para direita devido as portas S1 e S4 estarem acionadas, já no segundo caso, observa-se que as chaves S2 e S3 estão sendo acionadas, portanto seu sentido de corrente é o inverso assim rotacionando para o sentido contrário do anterior, o mesmo também possibilita realizar a frenagem por meio de suas portas.

Deste modo, utilizou-se esta *shield* para receber os valores de saída fornecidos pela lógica fuzzy por meio de sua defuzificação.

Figura 15 - Funcionamento interno de uma ponte H



Fonte: Arduino e Cia (2014).

4 LÓGICA FUZZY

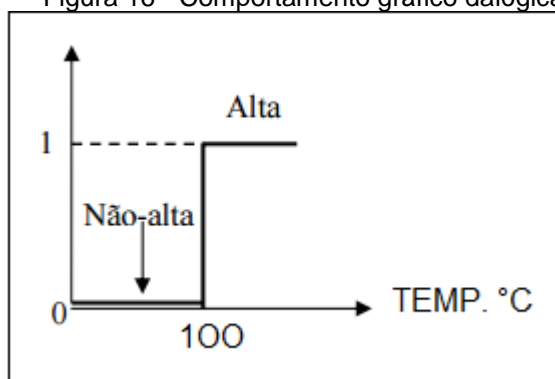
Lógica *Fuzzy* ou Lógica Nebulosa como também é conhecida, teve início no ano de 1965 pelo seu criador o professor Lofti Asker Zadeh. A lógica *fuzzy* tem o objetivo de tratar do aspecto de imprecisão de uma informação, trabalhando com o conceito de intervalo, tendo-se o grau de pertinência que pode variar entre 0 e 1, definido pela função de pertinência (LUGER, 2013).

A lógica *fuzzy* tem como objetivo trabalhar com raciocínio aproximado, com proposições imprecisas, utilizando como principal ferramenta a teoria de conjuntos *fuzzy* (ZADEH 1965, tradução nossa). A teoria dos conjuntos *fuzzy* tem como propósito afirmar o quanto um objeto atende uma descrição vaga, desta forma não é apenas verdade ou falso, mas sim

aproximações dentre os valores 0 e 1 (RUSSELL; NORVIG, 2013; WANG, 1997, tradução nossa).

Na figura 16 pode-se observar o comportamento de um elemento quando pertence ou não a um conjunto utilizando lógica booleana, ou seja, se o elemento pertence ao conjunto alta temperatura seu valor será 1, caso contrário não pertencerá ao conjunto alto.

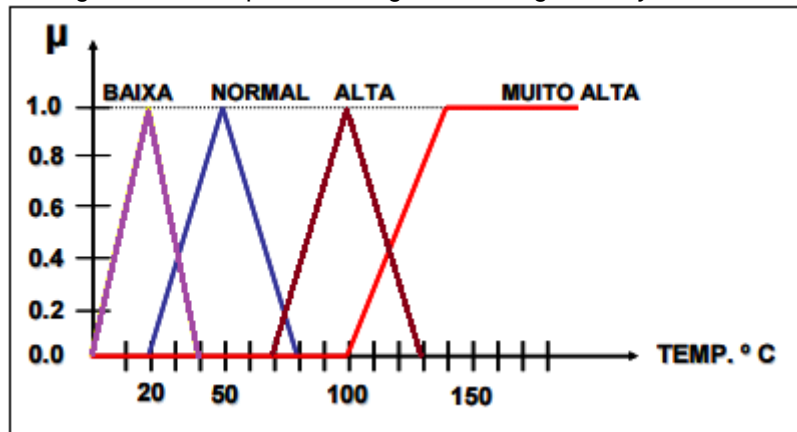
Figura 16 - Comportamento gráfico da lógica booleana



Fonte: Adaptado de Biondi Neto et al. (2006).

Na figura 17 pode-se observar o comportamento de um elemento quando pertence ou não a um conjunto utilizando a lógica *fuzzy*, ou seja, se o elemento não pertence ao conjunto o valor dele a este conjunto será zero, pois este método possibilita trabalhar com valores aproximados, deste modo é possível trabalhar com situações que possuem imprecisões, como por exemplo, se a temperatura estiver em 30°C ela possui um determinado grau de pertinência ao conjunto baixo, normal entre outros.

Figura 17 - Comportamento gráfico da lógica *fuzzy*



Fonte: Biondi Neto et al. (2006).

Segundo Mozelli (2008), a teoria de sistemas *fuzzy* é um método utilizado para resolver problemas complexos, que a necessidade de tratar as informações imprecisas por meio do raciocínio aproximado. Nas últimas décadas, a lógica *fuzzy* tem sido aplicada em diversas áreas, como por exemplo: medicina, economia, sistemas de transportes, entre outros. Segundo Bisc (2013), a Universidade de Berkeley afirma que mais de 16 mil projetos foram patenteados nos Estados Unidos e 7 mil no Japão com a utilização da lógica *fuzzy* desde a publicação do artigo de Zadeh.

Segundo Russell e Norvig (2013), a lógica *fuzzy* expõe a falta de precisão de forma quantitativa, inserindo uma função de pertinência a conjuntos que pode assumir valores entre 0 e 1. A lógica *fuzzy* é utilizada no raciocínio com expressões que descrevem a pertinência de conjuntos difusos. Portanto a lógica difusa é um sistema verdade-funcional.

A lógica *fuzzy* é uma aliada na criação de sistemas de controle, sua utilização permite trabalhar com variáveis reais do tipo: velocidade, temperatura, distância, entre outras. As mesmas são consideradas variáveis linguísticas com valores *fuzzy*, alto, baixo, perto, longe, entre outros (BILOBROVEC; MARÇAL; KOVALESKI, 2004).

4.1 CONTROLE *FUZZY*

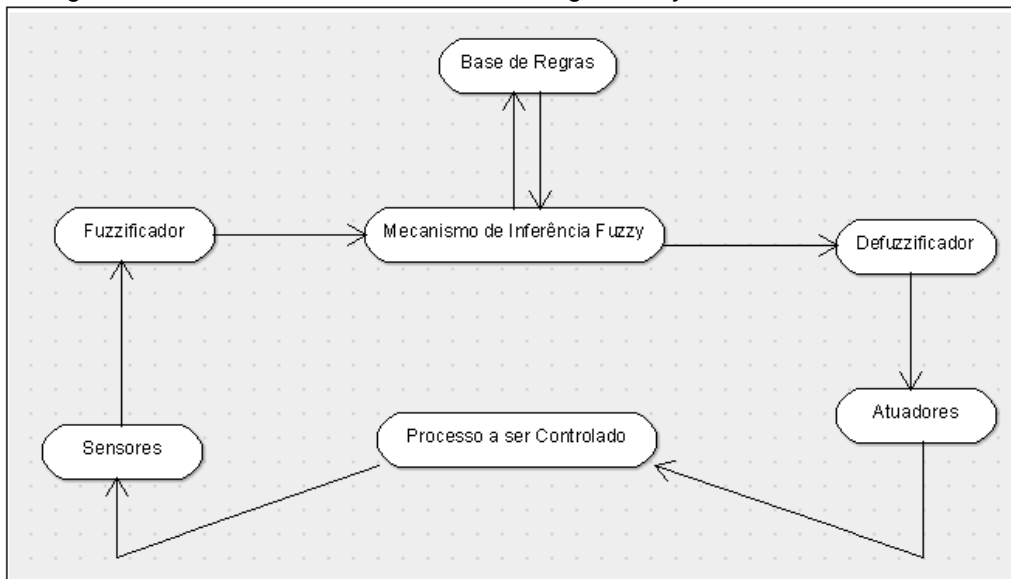
Controle *fuzzy* é responsável por transformar valores reais mapeados entre os parâmetros de entrada e saída em valores utilizados nas regras difusas. O mesmo tem tido sucesso em produtos comerciais como, por exemplo, o ar-condicionado (RUSSELL; NORVIG, 2004).

O controlador *fuzzy* pode ser aplicado em sistemas que não possuam uma estrutura adequada para o seu controle, em virtude da dificuldade ou inviabilidade para a obtenção de um modelo matemático convincente (BILOBROVEC; MARÇAL; KOVALESKI, 2004).

A lógica *fuzzy* utilizada em sistema de controle consiste das seguintes etapas (figura 20):

- a) fuzificador: responsável por mapear os valores lidos por meio dos sensores e transformá-los em informações *fuzzy* (WEBER; KLEIN, 2003).
- b) base de regras *fuzzy*: base de conjuntos de regras do tipo SE-ENTÃO (SILVA, 1999);
- c) mecanismo de inferência *fuzzy*: encarregado pelo mapeamento de conjuntos *fuzzy* em outros conjuntos de acordo com as regras e os valores de pertinência existentes (SILVA, 1999);
- d) defuzificador: é responsável por pesar as diversas respostas fornecidas pelas regras, assim transformando a saída em um número, que resultará em uma ação (SILER; BUCLEY, 2005 tradução nossa, WEBER; KLEIN 2003),

Figura 18 – Sistema de funcionamento de lógica *fuzzy*



Fonte: Adaptado de Weber e Klein (2003).

O controlador fuzzy pode ser projetado de forma a ser de raciocínio dedutivo como os seres humanos, ou seja, realiza deduções por meio das informações conhecidas ou indutivo que permite o aprendizado e generalização pela observação do comportamento do processo em uma situação dinâmica ou que muda conforme o tempo, mas geralmente este tipo de raciocínio é conhecido como controle fuzzy aprendiz ou adaptativo (SIMÕES, 2007).

4.1.1 Fuzificação

O processo de fuzificação é responsável por coletar os valores por meio das funções de pertinência de cada entrada, onde ocorre a manipulação para adequação dos dados para máquina de inferência, deste modo pode-se afirmar que os valores reais de entrada são responsáveis por determinar o grau de pertinência (CRUZ, 2009).

Segundo Biondi Neto et al (2006), a função de pertinência é responsável por refletir a semântica de um conjunto *fuzzy*. Segundo Cruz (2009), existem vários perfis de funções de pertinência que podem ser interpretadas por meio de funções, as mais utilizadas são:

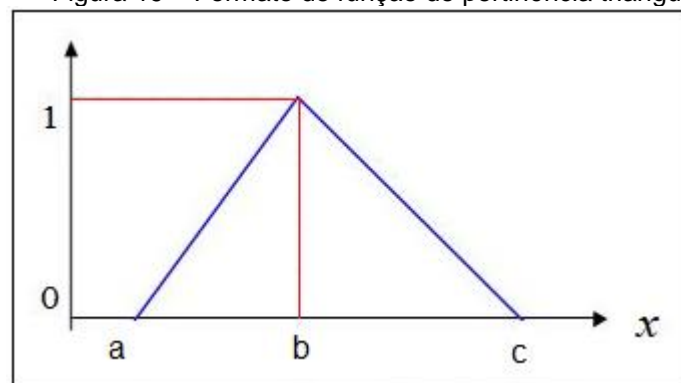
a) função de pertinência triangular: segundo Garcia (2009) é uma

função simples, sua estrutura é composta por três variáveis, a, b e c. Na figura 19 pode-se observar que o formato que se cria no gráfico possui um formato de triângulo. A função triangular é matematicamente definida por:

$$f(x, a, b, c) = \begin{cases} \mu(x) = \frac{x-a}{b-a}, \text{ para } a \leq x < b \\ \mu(x) = \frac{c-x}{c-b}, \text{ para } b \leq x \leq c \\ \mu(x) = 0, \text{ caso contrário} \end{cases} \quad (2)$$

$$f(x, a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

Figura 19 – Formato de função de pertinência triangular



Fonte: Adaptado de Garcia (2009).

b) função de pertinência trapezoidal: diferencia-se da função triangular por utilizar quatro variáveis (GARCIA, 2009).

Na figura 20 pode-se observar que o gráfico trabalha com quatro variáveis e as mesmas interligadas apresentam um formato de trapézio.

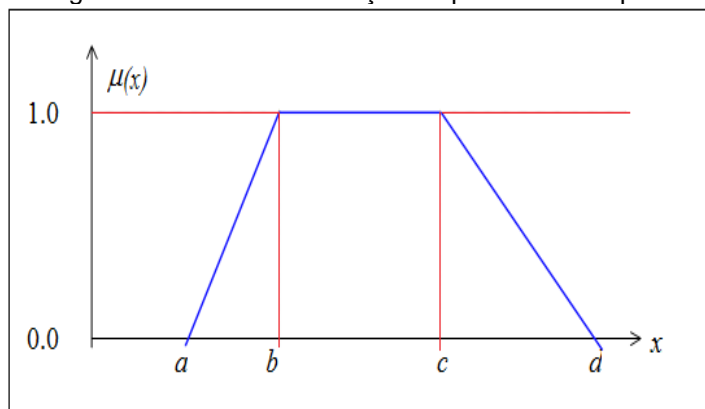
Segundo Biondi Neto et al.,(2006), a função de pertinência

$$f(x,a,b,c,d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}$$

trapezoidal é: (3)

$$f(x,a,b,c,d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right)$$

Figura 20 - Formato de função de pertinência trapezoidal



Fonte: Garcia (2009).

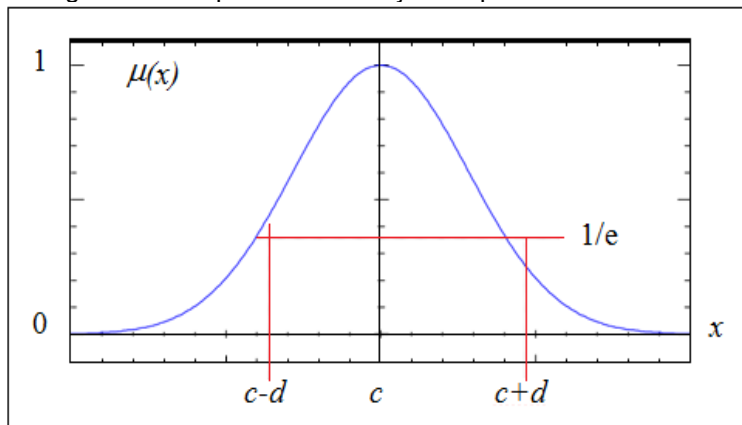
- c) função de pertinência gaussiana: é identificada pelo desvio padrão e pela média agregados a variável de entrada. A mesma apresenta características simétricas “em forma de sino” e possui valores diferentes de zero para todo controle da variável (HORTEGAL, 2016). Na figura 21 apresenta-se a aplicação da função de pertinência Gaussiana em uma distância euclidiana onde $m = c$ e $v = d$.

Segundo Garcia (2009), a expressão para a função de pertinência Gaussiana é:

$$\mu(x) = e^{-\left(\frac{x-m}{v}\right)^2} \quad (4)$$

Onde o m representa a média e v o desvio padrão.

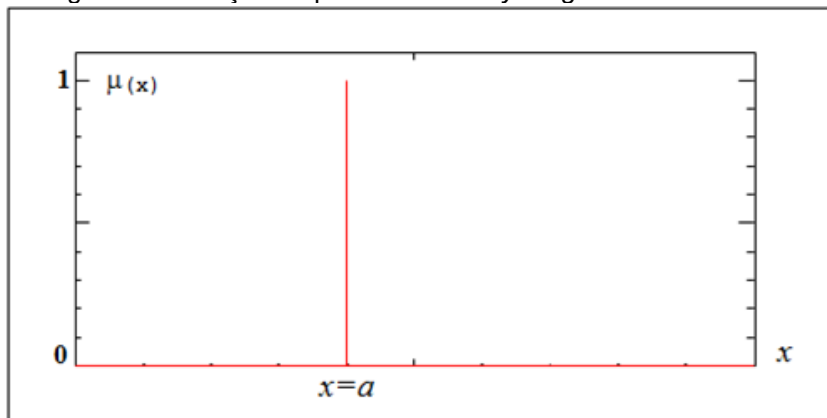
Figura 21 – Expressão de função de pertinência Gaussiana



Fonte: Garcia (2009).

d) função de pertinência *singleton*: o principal benefício deste tipo de função é a facilitação da lógica *fuzzy* no processo de defuzzificação. Em relação as funções de pertinência apresentadas anteriormente, esta função é muito comum entre os termos consequentes, enquanto as demais são encontradas tanto nos termos antecedentes, quanto nos termos consequentes. Na figura 22 pode-se observar que a função *singleton* assumiu o grau de pertinência igual a 1. Portanto, em apenas um ponto o grau de pertinência é igual a 1 no seu domínio e grau 0 nos demais pontos (GARCIA, 2009).

Figura 22 - Função de pertinência fuzzy Singleton



Fonte: Garcia (2009).

4.1.2 Base de regras *fuzzy*

A base de regras é criada durante o projeto do sistema em si, reconhecendo as suas características. A definição das regras é feita por meio do auxílio de um especialista, ou seja, uma pessoa com conhecimento no sistema a ser desenvolvido (BITTENCOURT, 2011). Ainda segundo Bittencourt (2011), após determinar as funções de pertinências das variáveis de entrada e saída, é fundamental conseguir as regras do sistema *fuzzy*, também consideradas como base de conhecimento ou inteligência do sistema.

A lógica *fuzzy* é formada por meio de várias regras compostas de SE-ENTÃO, onde as condições das regras são conhecidas como antecedentes e suas ações são chamadas de conseqüências. Portanto, estas condições verificam as regras de entradas do conjunto *fuzzy* e as ações fornecem as possíveis respostas de saída aceitas pelas variáveis linguísticas definidas em um determinado conjunto *fuzzy* (BARROS; BASSANEZI, 2006).

As variáveis linguísticas possuem as mesmas funções que as variáveis, assumindo valores, porém ao invés de trabalhar com valores numéricos assumem valores linguísticos, ou seja, nomes que são encontrados em um conjunto linguístico. Seus termos são definidos pelos conjuntos *fuzzy* reproduzidos pela função de pertinência denominada como variável linguística. Na teoria do conjunto *fuzzy* as variáveis exatas que pertencem ao conjunto clássico são identificadas como variáveis “verdadeiras ou falsas” (ZADEH, 1965; ZADEH, 1973 apud CRUZ, 2009).

Segundo Zimmermann (1991) apud Cruz (2009), a variável linguística é formada pelo grupo: $\{X, T(X), U, G, M\}$, que possuem as seguintes definições:

a) X é o nome dado a variável linguística (nome do conjunto), por

exemplo, $X = \text{Altura}$;

b) $T(X)$ é um conjunto de termos linguísticos (atributos, adjetivos);

por exemplo, $T = (\text{Altura}) = \{\text{Baixo, Mediana, Alto}\}$;

c) U é o universo de discurso (faixa, intervalo);

por exemplo, $U = [0,1]$;

d) G é uma regra sintática para composição dos termos linguísticos;

por exemplo, $G = \text{Baixo } [X: 0,0 \text{ a } 0,5], \text{ Mediana } [X: 0,3 \text{ a } 0,7] \text{ e Alto}$

$[X: 0,7 \text{ a } 1]$;

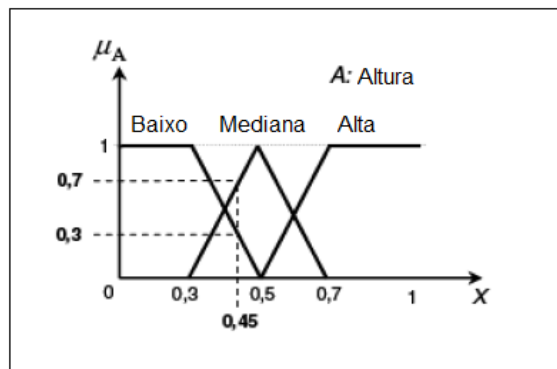
e) M é uma regra semântica que associa cada termo linguístico de G a um Conjunto difuso.

Por exemplo, $M = \text{associação dos termos de } G \text{ aos conjuntos difusos definidos por } \mu_{\text{Baixo}}, \mu_{\text{Mediana}} \text{ e } \mu_{\text{Alto}}$.

Na figura 23 pode-se verificar o funcionamento das variáveis linguísticas para a determinação de uma altura, onde seus valores são baixo, mediana e alta.

altura

Figura 23 - Função de pertinência para a variável linguística



Fonte: Adaptado de Cruz (2009).

4.1.3 Inferência *fuzzy*

A inferência *fuzzy* é responsável por definir a resposta do sistema com relação as entradas apresentadas. Nesta etapa ocorre a avaliação das regras e os dados quando e onde serão utilizados. Neste momento ocorre a combinação dos antecedentes das regras, implicação e *Modus Ponens* (é um

tipo de argumento em que as premissas hipotéticas são testadas e que interferem no resultado) (DADONE, 2001, tradução nossa).

Este processo é encarregado de realizar a tradução matemática por meio das técnicas de lógica *fuzzy*, para cada uma das teorias. Deste modo, encontra-se a saída que por sua vez é o controle a ser adotado pelo controlador definido a partir da entrada de cada conjunto *fuzzy* analisado.

A máquina de inferência pode apresentar os modelos de inferência Kang-Takagi-Sugeno (KTS) e Mamdani (BARROS, BASSANEZI, 2006).

Segundo Martins (2007) o modelo Mamdani também pode ser chamado de modelo linguístico, devido a sua formação de regras.

Originou-se com propósito de tentar controlar um conjunto de turbina a vapor/boiler por meio de regras derivadas de um especialista humano (MAMDANI; ASSILIAN, 1975 apud GARCIA 2009).

Segundo Martins (2007), o modelo de inferência Mamdani é proposto pela seguinte regra:

Regra: Se r é Q e s é W então h é N

Onde Q , N e W são os conjuntos *fuzzy* das variáveis de entrada e saída r , s e y respectivamente.

Geralmente neste modelo, as regras para construção do sistema são fornecidas por operadores humanos especialistas. Portanto, por meio das regras é possível realizar a interpretação linguística do sistema e suas ações.

4.1.4 Defuzificação

Segundo Barros e Bassanezi (2006) a defuzificação é responsável por transformar os valores do conjunto *fuzzy* em valores numéricos (*crisp*). Segundo Martins (2007), existem diversos métodos específicos para realizar esta conversão, cada um com suas vantagens e desvantagens, os mais utilizados são:

- a) máximo: utiliza o valor máximo do conjunto de saída. Não possui um funcionamento satisfatório em saídas com intervalos;

- b) média dos máximos: utiliza a média entre os valores máximos do conjunto de saída *fuzzy*. Deste modo, quando o valor máximo do conjunto é excedido, e quando a média extraída é exatamente um valor com grau de pertinência zero, o método apresenta erro em seus valores; e
- c) centro de gravidade: também conhecido por centróide, necessita de um poder computacional superior aos demais para obter o valor exato.

Portanto, para a aplicação da teoria *fuzzy*, utilizou-se de uma biblioteca desenvolvida para a plataforma arduino chamada de *Embedded Fuzzy Logic Library* (eFLL).

4.1.5 Biblioteca eFLL

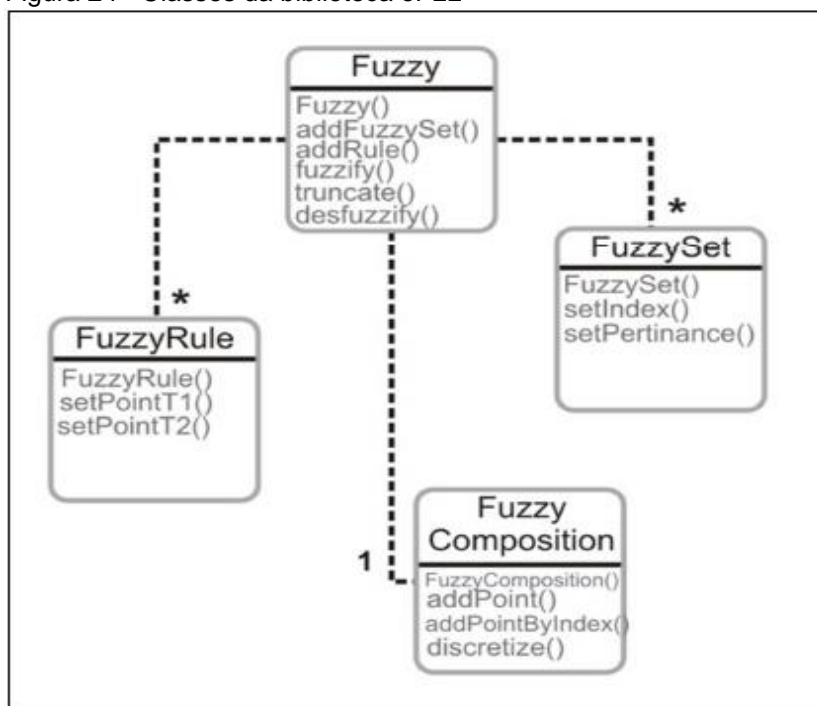
A biblioteca eFLL foi desenvolvida no Brasil pela Universidade Estadual do Piauí – Uespi, sendo orientada a objeto e visa auxiliar em sistemas embarcados com pouco processamento a bordo. Isto não quer dizer que não possa ser utilizada em outras plataformas, porém, para sua utilização necessita-se de comandos realizados na linguagem C++. Sua estrutura possui os conceitos e técnicas *Fuzzy*, ou seja, possui um sistema de regras, fuzificação, máquina de inferência e defuzificação (KRIDI et al., 2014). A biblioteca utiliza-se do processo max- min para a inferência e composição e para a defuzificação o método de centro de área. Uma desvantagem da eFLL é somente o uso das funções trapezoidais, *singleton* e triangulares (GITHUB, 2017). Porém sua vantagem é a interface de fácil configuração, para realizar o sistema *fuzzy*. A mesma possui quatro classes que são: *Fuzzy*, *FuzzySet*, *FuzzyRule* e *FuzzyComposition* que se interligam para realizar o processo de defuzificação (KRIDI et al., 2014).

Segundo Kridi et al. (2014) a classe *Fuzzy.cpp* é denominada como classe principal da biblioteca, deste modo pode-se determinar a quantidade de

entradas a serem utilizadas na aplicação. Na classe principal encontram-se os métodos *fuzzify()* e *truncate()*. O método *fuzzify()* é responsável por alcançar a fuzificação dos valores das entradas, já o método *truncate()* é responsável por apontar o ponto em que a fuzificação atinge os conjuntos das variáveis. A classe *fuzzySet.cpp* tem como finalidade determinar quais conjuntos estão associados a uma determinada entrada e quais valores o representam. Já a classe *fuzzyRule.cpp* gerar as regras que permitem realizar as decisões do sistema *fuzzy*, isto só ocorrerá pelos métodos *addRule()* e *reOrderBaseRules()* que pertencem a classe *Fuzzy* e servem para chamar e organizar respectivamente suas regras. E por último a classe *FuzzyComposition* que tem como finalidade realizar a defuzificação dos valores obtidos pelo sistema *fuzzy*.

Na figura 24 pode-se observar o comportamento entre as funções que compõem a biblioteca eFLL.

Figura 24 - Classes da biblioteca eFLL



Fonte: Kridi et al. (2014).

5 TRABALHOS CORRELATOS

A seguir são apresentados alguns trabalhos correlatos, podendo observar a utilização da computação embarcada.

5.1 APLICAÇÃO DA LÓGICA *FUZZY* NO CONTROLE DA VELOCIDADE DE UM PROTÓTIPO MÓVEL

Para aplicação da lógica *fuzzy* no controle da velocidade de um protótipo móvel o professor João Erivando Soares Marques desenvolveu um veículo autônomo baseado em lógica fuzzy utilizando a plataforma arduino juntamente com sensor ultrassônico SRF08 que recebe os dados lidos da área para ter a distância do obstáculo, após realizar a coleta das informações o sensor passa-as para a placa arduino que baseada na lógica *fuzzy* aplicada no projeto define qual velocidade é adequada a ser utilizada pelos dois servomotores do veículo.

Primeiramente no sistema de controle definiu-se as funções de pertinência para entrada e saída do sistema, que foram três funções para entrada (distância): pequena, segura e grande que são funções trapezoidais e três funções de pertinência para saída (velocidade): lenta, normal e rápida que também são trapezoidais. Logo após foi definido a base de regras do veículo para poder controlar a velocidade em relação aos obstáculos encontrados.

Sua estrutura no código fonte se divide em três partes distintas. Na primeira parte encontram-se as bibliotecas e as funções que fazem parte da lógica *fuzzy*, na segunda parte encontra-se as bibliotecas e as funções para o comando do servomotores a partir das informações vindas do controlador *fuzzy*, e por último encontra-se a implementação da lógica responsável por capturar as informações através do sensor ultrassônico.

Nos resultados obtidos o protótipo apresentou-se de forma satisfatória para o controle de velocidade em relação ao obstáculo. Para que o protótipo reduzisse a velocidade conforme a aproximação do obstáculo utilizou-se o PWM para variar o ciclo de trabalho de acordo com a distância, podendo assim controlar a velocidade dos motores corretamente por meio da tensão rms (CIÊNCIA & DESENVOLVIMENTO, 2016).

5.2 DESENVOLVIMENTO DE UMA BIBLIOTECA FUZZY PARA O CONTROLE AUTÔNOMO DE UM ROBÔ MÓVEL EM AMBIENTE DESCONHECIDO

De acordo com Kridi et al., (2014), uma biblioteca aplicando lógica *fuzzy* foi desenvolvida na universidade estadual do Piauí. A mesma possui um algoritmo orientado a objetos criado na linguagem C++, e teve como propósito auxiliar no desenvolvimento de sistemas *fuzzy* em plataformas de *hardware* embarcados como microcontroladores com limitações em seus processamentos. Isto é possível porque a biblioteca eFLL como é chamada possui uma implementação simples e direta, sua estrutura é composta por quatro classes que são: *Fuzzy*, *FuzzySet*, *FuzzyRule* e *FuzzyComposition* onde as mesmas se interligam para realizar a modelagem da defuzificação. A classe *Fuzzy* cpp, é denominada como classe principal da biblioteca, deste modo

pode-se determinar a quantidade de entradas a serem utilizadas na aplicação. Na classe principal encontra-se os métodos *fuzzify()* e *truncate()*. O método *fuzzify()* é responsável por alcançar a fuzzificação dos valores das entradas, já o método *truncate()* é responsável por constatar o ponto em que a fuzzificação atinge o conjuntos das variáveis. A classe *fuzzySet.cpp* tem como finalidade determinar quais conjuntos estão associados a uma determinada entrada e quais valores o representam. Já a classe *fuzzyRule.cpp* tem como intuito a geração das regras, permitindo realizar as decisões do sistema *fuzzy*, por meio dos métodos *addRule()* e *reOrderBaseRules()* que pertencem a classe *Fuzzy* e servem para chamar e organizar respectivamente suas regras. E por último a classe *FuzzyComposition* que tem-se como finalidade realizar a defuzzificação dos valores obtidos pelo sistema *fuzzy*.

Para realizar a aplicação da eFLL utilizou-se um robô móvel com chip ATmega328P presente na placa arduino *Duemilanove*. Na aplicação da biblioteca no robô móvel, o mesmo deverá se locomover por um pequeno trecho sem colidir com os obstáculos encontrados no percurso. Para realizar esta tarefa o robô está equipado com um sonar, e dois sensores infravermelho que serão capazes de perceber o ambiente onde encontra-se, o mesmo também possui três servomotores para realizar sua locomoção, onde dois são encarregados pela movimentação das rodas e o terceiro encarregado de movimentar o sensor ultrassônico.

Nos testes iniciais o robô não evitava as colisões com os obstáculos, portanto criou-se mais dois conjuntos e diminuiu-se as leituras dos sensores infravermelhos para resolver este problema. Deste modo foi possível observar uma melhora significativa no seu desempenho, e assim realizar o que se havia sido proposto com o projeto.

5.3 NAVEGAÇÃO EM AMBIENTES DESCONHECIDOS POR ROBÔ MÓVEL AUTÔNOMO BASEADO EM PLATAFORMA ARDUINO COM LINGUAGEM C

O trabalho desenvolvido por Vitor Martes Sternlicht, Valdenice Minatel Melo de Cerqueira, Rodrigo da Silva Viana, teve como objetivo a criação de um robô na plataforma arduino com capacidade de mapear o ambiente. O mesmo está equipado com um sensor ultrassônico para realizar a captação de obstáculos pelo caminho percorrido, um encoder (sensor de rotação), sensores de fotodiodo, dois motores de correntes continua com caixa de redução e um *shield Bluetooth* capaz de se comunicar com o computador em tempo real. O protótipo foi desenvolvido C, C++ e .NET, comunicação por *bluetooth*, seu conceito de programação foi no modelo de pilha avançado com armazenamento em *arrays*, juntamente foi desenvolvido um software que apresenta inúmeras opções de controle capaz de construir em tempo real o mapa do ambiente, e que roda no sistema operacional *Windows*.

O protótipo segue a exploração para o lado leste até não possuir opção de deslocamento então o mesmo retorna procurando posições adjacentes no percurso para explorar, então volta sua a posição inicial e desloca-se para o sentido o sentido oeste para realizar a exploração do mapa com o mesmo procedimento realizado a leste. A demarcação no mapa ocorre quando o robô desloca-se o seu próprio tamanho que é de (17 cm x 17 cm).

Ao longo do desenvolvimento do projeto ocorreram alguns problemas, como por exemplo, o alinhamento do protótipo já que os motores possuíam variações entre si nas suas velocidades, para sanar este problema foi utilizado enconder para aprimorar os movimentos do mesmo. Porém o maior desafio encontrado para a realização deste projeto foi realizar a conectividade entre o computador e o *bluetooth*.

5.4 AUTOBOMOUS ROBOT OBSTACLE AVOIDANCE USING A FUZZY LOGIC TRONTROL SCHEME

Neste projeto o acadêmico William Martin propôs um robô autônomo para desviar de obstáculos por meio de lógica *fuzzy* e a plataforma arduino para controle. Para realizar o mesmo, o acadêmico utilizou-se de um veículo de brinquedo em forma de tanque.

Segundo Martin (2009) o desenvolvimento do mesmo necessitou de três sensores infravermelho para medições de distâncias entre 10 e 80 cm. Os mesmos foram posicionados sobre a estrutura do mesmo, sendo, um ao centro, um a direita e um a esquerda, permitindo assim um melhor desempenho do robô.

Para a locomoção criou-se um esquema de controle por meio de lógica *fuzzy* em relação ao obstáculo. Já para controlar os dois motores, utilizou-se do PWM juntamente com circuito integrados (L2722). Esses circuitos integrados permitem a alimentação dos motores com energia reduzida vindo do PWM.

Foram definidos três conjuntos de função de pertinência, distância, velocidade de translação e velocidade de rotação, sendo feita por um total de duas funções de pertinência de entrada e oito de saída. Também foram utilizadas as funções triangular e trapezoidal, porém modificou-se o método de calcular devido a utilização de um microcontrolador. Para uma melhor eficiência dos conjuntos de dados reduziu-se de um segundo para 40 milissegundos.

Uma vez que os dados de entradas sejam fuzificados, as oito regras definidas são executadas em ordem atribuindo valores para ambas as velocidades. Estes resultam em valores múltiplos para cada componente de saída, tendo necessidade de levar o máximo desses valores para cada componente de saída. Por fim os valores de saída foram defuzificados usando o método do produto- máximo resultando no controle de velocidade dos motores.

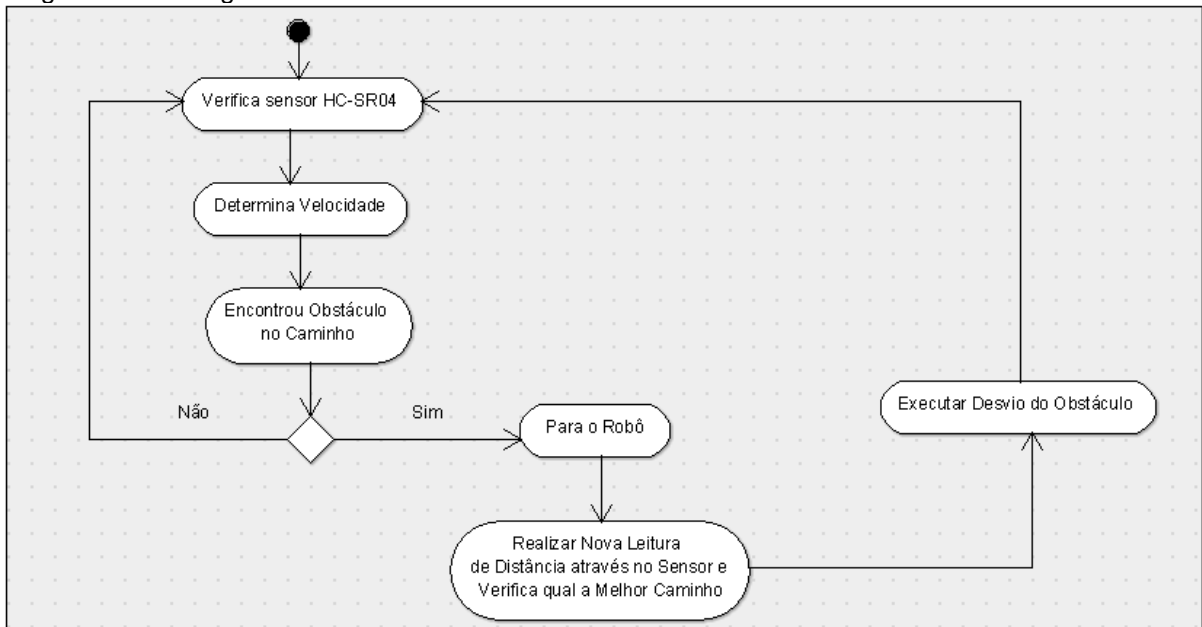
O esquema de controle *fuzzy* utilizado permitiu que o robô atendesse de forma rápida e satisfatória. Durante os testes Martin (2009) cita problemas na detecção de obstáculo devido a forma geométrica dos mesmos. Apesar disso os resultados apresentaram satisfatórios. Para trabalhos futuros Martin indica a utilização de um servo para realizar a captura de sensores em 180° garantindo um maior grau de confiabilidade.

6 PROTÓTIPO DESENVOLVIDO COM INTEGRAÇÃO DA PLATAFORMA ARDUINO COM LÓGICA FUZZY

A área da robótica, mais especificamente a de robótica móvel vem apresentando uma evolução de seus métodos tecnológicos, conforme mencionado anteriormente. O avanço desta área beneficia a sociedade em geral, devido a sua vasta área de aplicação. No desenvolvimento deste protótipo aplicou-se os conceitos aprendidos durante a pesquisa. Este capítulo apresenta a modelagem e montagem do desenvolvimento do protótipo idealizado a partir dos conceitos aqui abordados, ele se divide em quatro subseções: a primeira aborda os componentes utilizados para a montagem da estrutura do protótipo, a segunda apresenta os testes do sensor HC-SR04, no terceiro apresenta a aplicação de lógica *fuzzy*, como suas regras, defuzzificação, variáveis linguísticas, entre outras, e por fim, na quarta sessão são apresentados os métodos responsáveis por determinar as direções a serem realizadas.

Apresenta-se na figura 25 o funcionamento interno do protótipo por meio de um fluxograma de atividade.

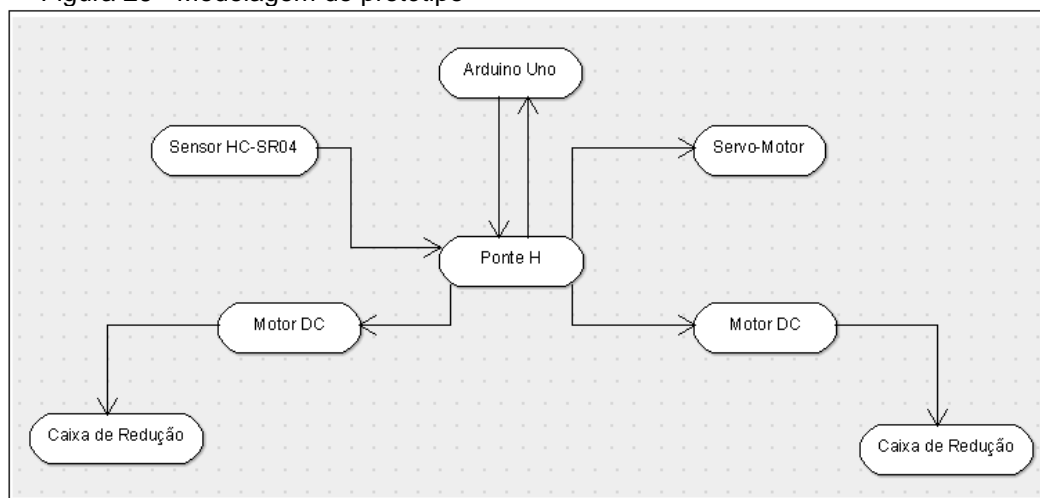
Figura 25 - Fluxograma do funcionamento interno



Fonte: Do autor.

Já o modelo da estrutura do protótipo como pode-se observar na figura 26, foram interligados por meio de seus pinos o motor *shield* l293d - driver ponte H e o arduino UNO, logo após conectou-se os dois motores DC na ponte H que possuem as caixas de redução em suas extremidades para o encaixe das rodas. Posteriormente, foram conectados o servomotor e o sensor ultrassônico para a realização das leituras no ambiente.

Figura 26 - Modelagem do protótipo



Fonte: Do autor.

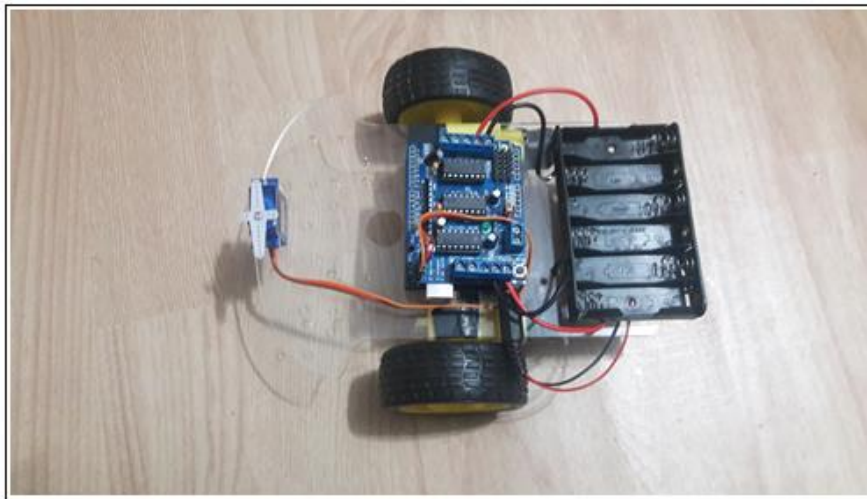
6.1 MONTAGEM DA ESTRUTURA

Para o desenvolvimento do protótipo foram comprados todos os componentes necessários. A alimentação era insuficiente, foi necessário aumentar a carga com duas baterias extras. Foi anexado o arduino a estrutura do protótipo e conectou-se a ponte H sobre a mesma, em seguida conectaram-se os cases das pilhas para a alimentação das placas, logo após inseriu-se os fios de alimentação dos dois motores DC na ponte H. Depois deste procedimento conectou-se o servomotor que será responsável por movimentar o sensor ultrassônico para realizar a captura das informações do ambiente.

Na figura 27 mostra-se a estrutura descrita acima com as conexões dos motores na ponte H, o case de pilhas utilizadas no projeto, o posicionamento das rodas e o servomotor fixado na frente e bem centralizado

na base, onde é encarregado de movimentar o sensor tanto para o lado esquerdo quanto para o direito.

Figura 27 – Estrutura do protótipo



Fonte: Do autor.

Teve-se a necessidade de realizar a troca da base do protótipo, pois a sua base original de acrílico quebrou, portanto criou-se uma nova base com a mesma modelagem, porém de madeira compensada, após realizar a troca da base e conectar as fiações, o próximo passo para a conexão do servomotor e o sensor ultrassônico HC-SR04 e realizar suas ligações com a *shield* I293d - driver ponte H como pode-se observar na figura 28.

Figura 28 - Ligação do sensor ao protótipo



Fonte: Do autor.

6.2 TESTE DO SENSOR HC-SR04

Para a validação do sensor ultrassônico HC-SR04 utilizou-se uma régua para verificação da distância em entre o sensor e o obstáculo, no caso uma caixa de papelão serviu como obstáculo. Ao deslocar-se a caixa sobre a régua é possível determinar a distância em cm entre a caixa e o sensor, para verificar se a leitura estava de acordo com a distância utilizou-se o monitor serial da plataforma arduino onde possibilitou a verificação das leituras realizadas pelo sensor. Deste modo verificou-se a precisão de leitura do sensor em relação ao obstáculo.

Na figura 29 pode-se observar o posicionamento do sensor em relação ao obstáculo.

Figura 29 - Teste do sensor



Fonte: Do autor.

Na figura 30 apresenta-se um trecho do pseudocódigo responsável por realizar a captura e calcular a distância entre o protótipo e o obstáculo. Inicialmente é desligada a emissão do som, logo após aguarda-se 4 μ s para iniciar a emissão do som, o mesmo permanece ligado por 20 μ s emitindo ondas sonoras, então auto desligasse e aguarda 10 μ s para receber a onda sonora.

Posteriormente, o mesmo liga o receptor e verifica quantos pulsos retornou, então é realizado o cálculo que determina a distância em cm.

Figura 30 - Pseudocódigo para capturar a distância do obstáculo

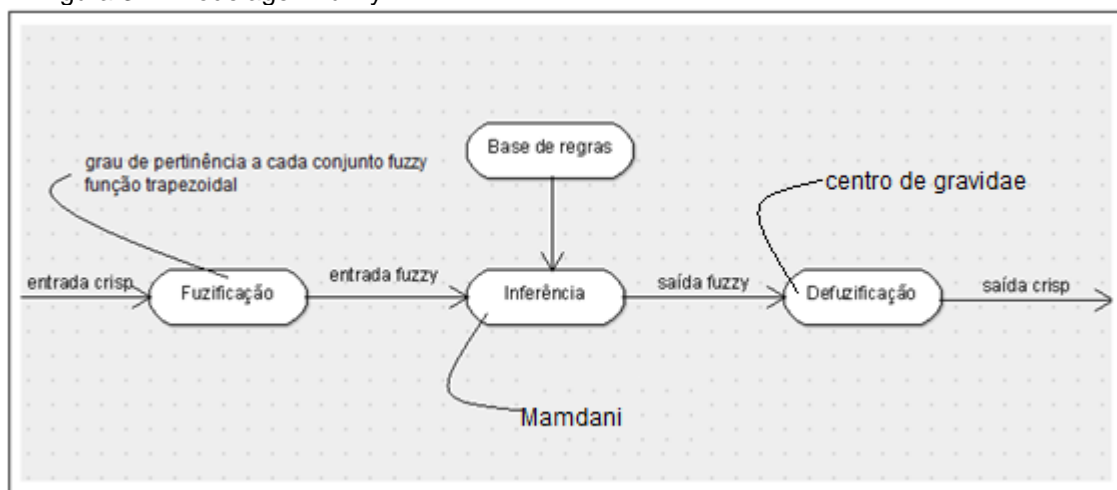
```
digitalWrite(HC_SR04_TRIGGER, LOW);  
delayMicroseconds(4);  
digitalWrite(HC_SR04_TRIGGER, HIGH);  
delayMicroseconds(20);  
digitalWrite(HC_SR04_TRIGGER, LOW);  
delayMicroseconds(10);  
long pulse_us = pulseIn(HC_SR04_ECHO, HIGH);  
dist = pulse_us / 59;  
delayMicroseconds(300);
```

Fonte: Robô, (2014).

6.3 MODELAGEM FUZZY

As etapas necessárias para a elaboração da lógica *fuzzy* iniciaram com aquisição do conhecimento, seguida pela modelagem *fuzzy* e desenvolvimento do modelo a ser aplicado. Deste modo, pode-se observar na figura 31 a modelagem *fuzzy* aplicada no projeto.

Figura 31 - Modelagem fuzzy



Fonte: Do autor.

A modelagem fuzzy compreendeu inicialmente a definição das funções de pertinência da variável de entrada e de saída do protótipo,

posteriormente realizou-se a definição da base de regras, e as etapas propriamente ditas referentes ao desenvolvimento de um sistema fuzzy que compreende a fuzificação, avaliação das regras e defuzificação.

6.3.1 Definição das funções de pertinência

Durante a aquisição do conhecimento, definiu-se como variável de entrada a distância e como variável de saída a velocidade. Para a variável distância formou-se os conjuntos *fuzzy*:

Variável de entrada distância → Conjuntos $\left\{ \begin{array}{l} \text{Pequena} \\ \text{Mediana} \\ \text{Grande} \end{array} \right.$

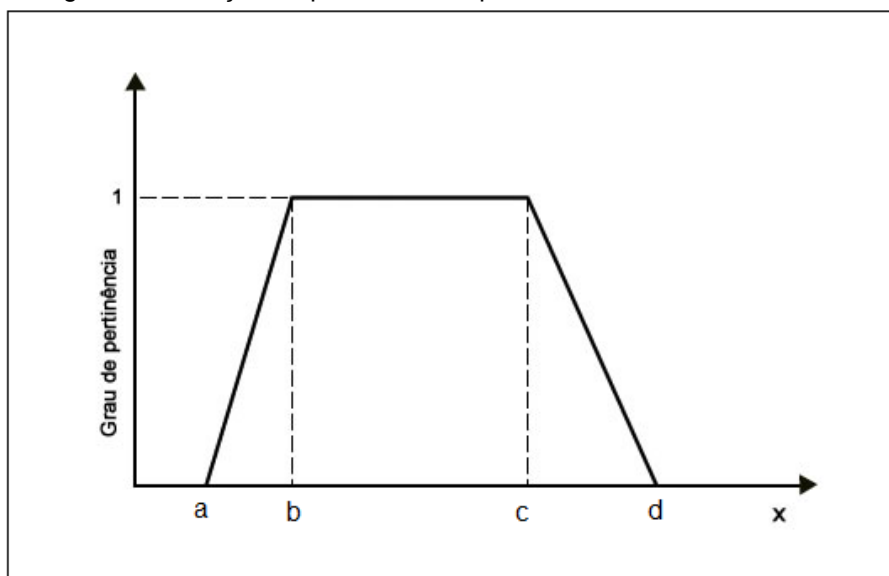
Para a variável velocidade formou-se os conjuntos fuzzy:

Variável de saída velocidade → Conjuntos $\left\{ \begin{array}{l} \text{Baixa} \\ \text{Média} \\ \text{Rápida} \end{array} \right.$

As funções de pertinência para as variáveis de entrada e de saída do protótipo foram definidas por meio da função trapezoidal. Segundo Biondi Neto et al. (2006), o cálculo realizado pela função trapezoidal é dada por:

$$f(x,a,b,c,d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}$$

Figura 32 - Função de pertinência trapezoidal



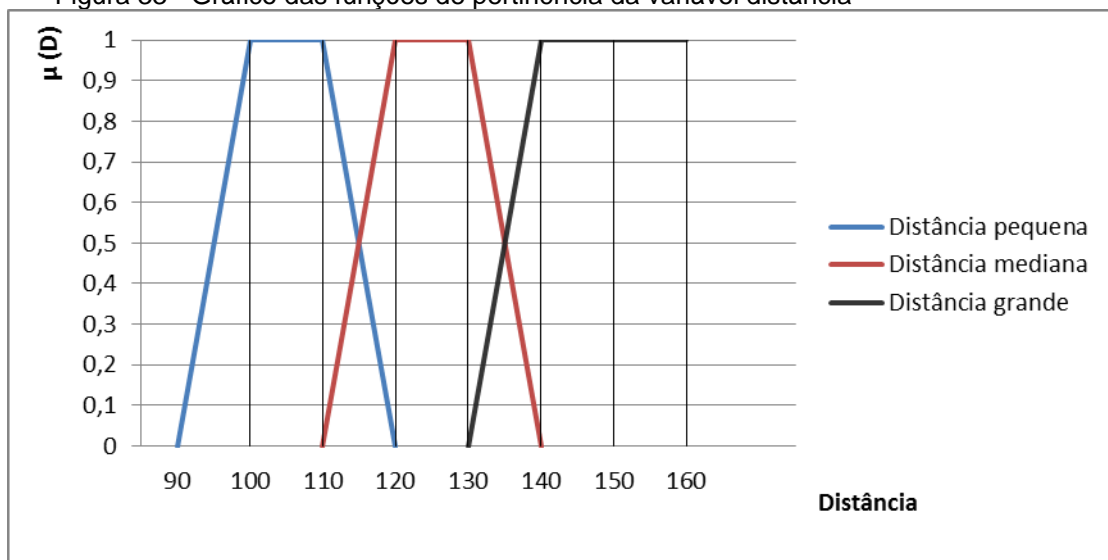
Fonte: Garcia, Melo, Shirru (2009).

Onde:

- a) x é o valor da variável distância;
- b) a e d onde o grau de pertinência é 0;
- c) b e c o ponto onde o grau de pertinência é 1.

A função de pertinência da variável de entrada mostra-se no modelo trapezoidal aplicado no desenvolvimento deste trabalho.

Figura 33 - Gráfico das funções de pertinência da variável distância



Fonte: Do autor.

Tendo-se os seguintes conjuntos fuzzy definidos:

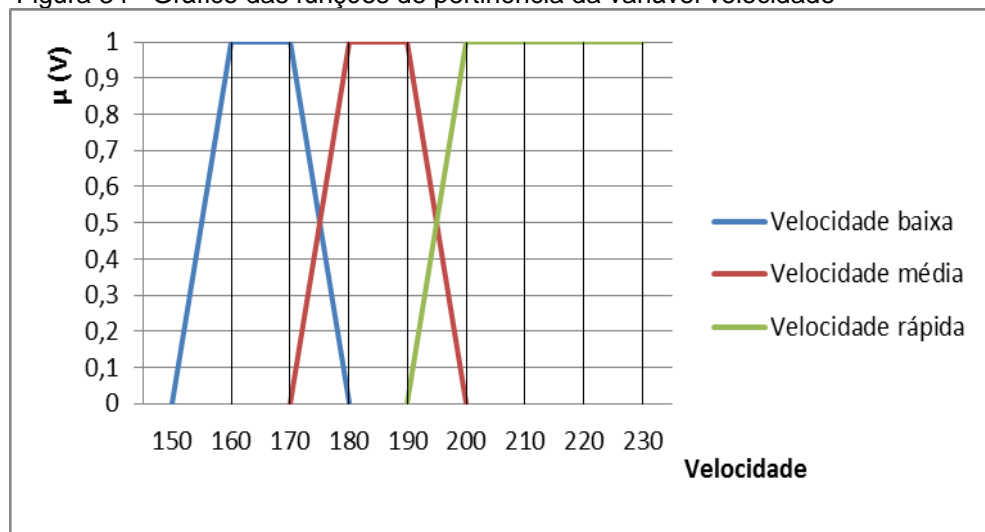
$$\mu(D)_{\text{pequena}} = \begin{cases} 0 & \text{se } D \leq 90 \text{ ou } D \geq 120 \\ \frac{D-90}{100-90} & \text{se } 90 < D < 100 \\ \frac{120-D}{120-110} & \text{se } 110 < D < 120 \\ 1 & \text{se } 100 \leq D \leq 110 \end{cases}$$

$$\mu(D)_{\text{mediana}} = \begin{cases} 0 & \text{se } D \leq 110 \text{ ou } D \geq 140 \\ \frac{D-110}{120-110} & \text{se } 110 < D < 120 \\ \frac{140-D}{140-130} & \text{se } 130 < D < 140 \\ 1 & \text{se } 120 \leq D \leq 130 \end{cases}$$

$$\mu(D)_{\text{grande}} = \begin{cases} 0 & \text{se } D \leq 130 \text{ ou } D \geq 160 \\ \frac{D-130}{140-130} & \text{se } 130 < D < 140 \\ \frac{160-D}{160-150} & \text{se } 150 < D < 160 \\ 1 & \text{se } 140 \leq D \leq 150 \end{cases}$$

As funções de pertinência definidas para a variável de saída velocidade são formadas pelos conjuntos *fuzzy* baixa, média e rápida conforme a figura 34.

Figura 34 - Gráfico das funções de pertinência da variável velocidade



Fonte: Do autor.

$$\mu(V)_{baixa} = \begin{cases} 0 & \text{se } V \leq 150 \text{ ou } V \geq 180 \\ \frac{V-150}{160-150} & \text{se } 150 < V < 160 \\ \frac{180-V}{180-170} & \text{se } 170 < V < 180 \\ 1 & \text{se } 160 \leq V \leq 170 \end{cases}$$

$$\mu(V)_{m\u00e9dia} = \begin{cases} 0 & \text{se } V \leq 170 \text{ ou } V \geq 200 \\ \frac{V-170}{180-170} & \text{se } 170 < V < 180 \\ \frac{200-V}{200-190} & \text{se } 190 < V < 200 \\ 1 & \text{se } 180 \leq V \leq 190 \end{cases}$$

$$\mu(V)_{r\u00e1pida} = \begin{cases} 0 & \text{se } V \leq 190 \text{ ou } V \geq 220 \\ \frac{V-190}{200-190} & \text{se } 190 < V < 200 \\ \frac{220-V}{220-210} & \text{se } 210 < V < 220 \\ 1 & \text{se } 200 \leq V \leq 220 \end{cases}$$

6.3.2 Defini\u00e7\u00e3o da base de regras

O n\u00famero de regras aplicadas no prot\u00f3tipo foi encontrado por meio de um c\u00e1lculo que utiliza-se da quantidade de conjuntos *fuzzy* e vari\u00e1vel de entrada. Portanto o prot\u00f3tipo \u00e9 composto por tr\u00eas regras, que \u00e9 dada pela equa\u00e7\u00e3o $1 \times 3 = 3$ o que sugere a forma\u00e7\u00e3o:

Tabela 1 - Base de regras

Regras	Entradas		Sa\u00edda
1	Se dist\u00e2ncia = pequena	Ent\u00e3o	Velocidade = <i>baixa</i>
2	Se dist\u00e2ncia = <i>mediana</i>	Ent\u00e3o	velocidade = <i>m\u00e9dia</i>
3	Se dist\u00e2ncia = grande	Ent\u00e3o	Velocidade = <i>r\u00e1pida</i>

Fonte: Do autor

6.3.3 Fuzifica\u00e7\u00e3o

A fuzifica\u00e7\u00e3o consiste na identifica\u00e7\u00e3o de um valor fuzzy a cada um dos conjuntos da vari\u00e1vel de entrada, considerando-se um valor *crisp* de

entrada. Portanto, considerando-se que a distância do robô é de 115 cm, calcula-se o grau de pertinência, por meio das funções, para cada um dos conjuntos fuzzy distancia pequena, mediana e grande.

$$\mu_{\text{pequena}}(115) = (120 - 115) / (120 - 110) \quad (\text{opção 1 da função})$$

$$\mu_{\text{pequena}}(115) = 0,5$$

para o conjunto mediana

$$\mu_{\text{mediana}}(115) = (120 - 115) / (120 - 110) \quad (\text{opção 2 da função})$$

$$\mu_{\text{mediana}}(115) = 0,5$$

para o conjunto grande:

$$\mu_{\text{grande}}(115) = 0, \text{ pois } x \leq 115 \quad (\text{opção 3 da função})$$

6.3.4 Avaliação das regras

Os graus de pertinência obtidos por meio da fuzificação são então transferidos para o mecanismo de inferência. O objetivo desta etapa é determinar o grau de pertinência de cada regra.

Portanto como existe apenas um antecedente para cada regra, o grau de pertinência é o próprio grau de pertinência obtido na etapa anterior.

Dando sequência ao exemplo inicializado anteriormente, no caso do robô autônomo o resultado obtido da inferência para 115 cm é:

a) regra 1 (SE distancia = pequena ENTÃO velocidade = baixa):

grau de pertinência = 0,5;

b) regra 2 (SE distancia = mediana ENTÃO velocidade = média):

grau de pertinência = 0,5;

c) regra 3 (SE distancia = grande ENTÃO velocidade = rápida):

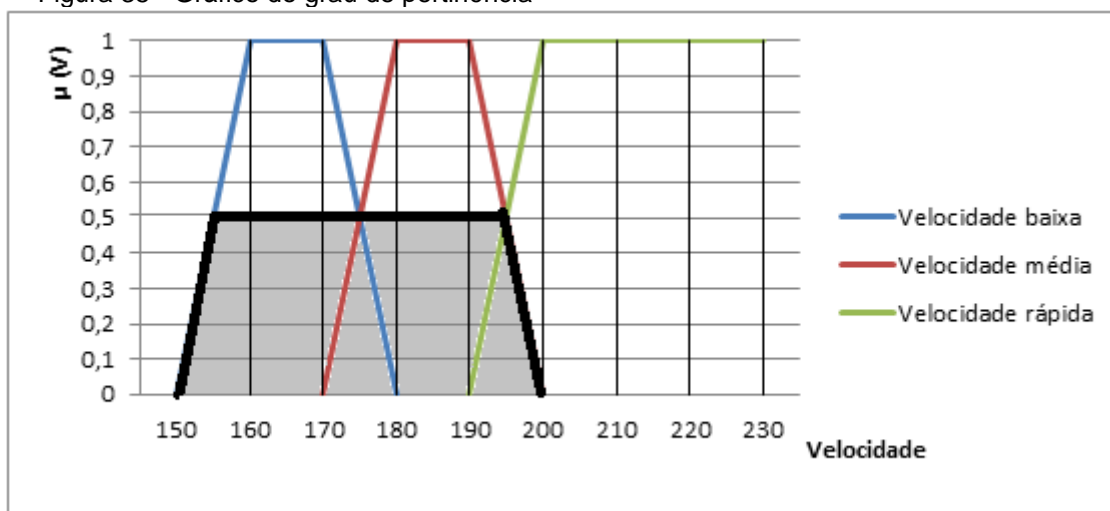
grau de pertinência = 0.

6.3.5 Defuzificação

Após a realização da inferência, procede-se a defuzzificação, a fim de transformar um valor fuzzy de saída em um valor crisp.

Na base de regras do robô autônomo não existem consequentes iguais, deste modo, o resultado de agrupamento é o próprio grau de pertinência. Portanto, o contorno formado pela união dos conjuntos pequena e mediana para a entrada de 115 cm pode-se observar na figura 35.

Figura 35 - Gráfico do grau de pertinência



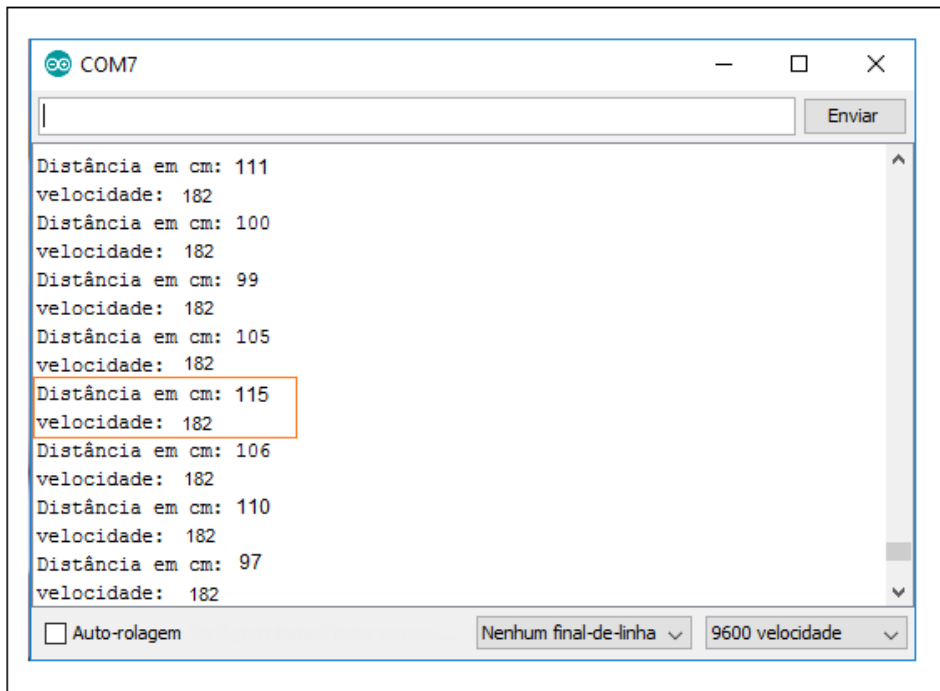
Fonte: Do autor.

Deste modo, por meio do universo de discurso resultante da união dos conjuntos baixa e média, e os graus de pertinência traçados pela união dos dois conjuntos, a fórmula do centro de gravidade foi aplicada. De acordo com Shaw e Simões (2007), a fórmula do centro de gravidade é:

$$u^* = \frac{\sum_{i=1}^N u_i \mu_{Out}(u_i)}{\sum_{i=1}^N \mu_{Out}(u_i)}$$

Onde $\mu_{Out}(u_i)$ são os graus de pertinência que formam o contornos dos conjuntos utilizados e u_i são os valores do universo de discurso que compõe os conjuntos utilizados.

Figura 36 - Apresentação no monitor serial do exemplo inicializado na fuzificação



Fonte: Do autor.

6.4 MOVIMENTAÇÃO DE ATUADORES

Para realizar a movimentação dos atuadores presentes neste projeto, utilizou-se de uma *shield* ponte H que por sua permitiu realizar todas as conexões responsáveis por dar vida ao projeto.

Neste processo foram conectados os dois motores responsáveis por dar tração ao protótipo, também conectou-se o servomotor por meio da pinagem 10 da ponte H, sendo este incumbido de mover o sensor para detectar obstáculos durante o percurso.

Na figura 36 apresenta-se o pseudocódigo responsável por determinar o ângulo de posicionamento do servomotor, que permite realizar as leituras do sensor nas direções desejadas, e iniciando-se na parte central do protótipo com ângulo de 90°.

Figura 37- Ângulo de posicionamento do servomotor

```
.
int calcularDistanciaCentro() {
    servo_ultra_sonico.write(90);
    delay(200);
    int leituraDoSonar = lerSonar();
    delay(600);
    leituraDoSonar = lerSonar();
    delay(600);
    Serial.print("Distancia Centro: ");
    Serial.println(leituraDoSonar);
    return leituraDoSonar;
}
int calcularDistanciaDireita() {
    servo_ultra_sonico.write(30);
    delay(200);
    int leituraDoSonar = lerSonar();
    delay(600);
    leituraDoSonar = lerSonar();
    delay(600);
    Serial.print("Distancia Direita: ");
    Serial.println(leituraDoSonar);
    return leituraDoSonar;
}
int calcularDistanciaEsquerda() {
    servo_ultra_sonico.write(180);
    delay(200);
    int leituraDoSonar = lerSonar();
    delay(600);
    leituraDoSonar = lerSonar();
    delay(600);
    Serial.print("Distancia Esquerda: ");
    Serial.println(leituraDoSonar);
    return leituraDoSonar;
}
```

Fonte: Robô, (2014).

Na figura 37 apresentam-se as funções responsáveis por determinar o deslocamento do protótipo por meio dos valores fornecidos pela variável *output*, assim permitindo realizar os comandos *forward* e *backward* para dar sentido de rotação as rodas. O comando *forward* é responsável por realizar a rotação para trás, já o comando *backward* é responsável por realizar a rotação para frente.

Figura 38 - Funções de locomoção

```
void rotacao_Parado()
{
  Serial.println(" Parar ");
  motor1.run(RELEASE);
  motor2.run(RELEASE);
}
void rotacao_Frente()
{
  Serial.print("Distância: ");
  Serial.println(averageDistance);
  motor1.setSpeed(output);
  motor2.setSpeed(output);
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  delay(100);
  Serial.print("velocidade: ");
  Serial.println(output);
}

void rotacao_Re()
{
  motor1.setSpeed(output);
  motor2.setSpeed(output);
  Serial.println(" ré ");
  Serial.println(output);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  delay(100);
  rotacao_Parado();
}
void rotacao_Direita()
{
  Serial.println(" Para a direita ");
  motor1.setSpeed(output);
  motor2.setSpeed(output);
  motor1.run(FORWARD);
  motor2.run(BACKWARD);
  delay(100);
}

void rotacao_Esquerda()
{
  Serial.println(" Para a esquerda ");
  motor1.setSpeed(output);
  motor2.setSpeed(output);
  motor1.run(BACKWARD);
  motor2.run(FORWARD);
  delay(100);
}
```

Fonte: Adaptado de Robô, (2014).

Para determinar qual seria o melhor caminho a ser percorrido pelo protótipo, o pseudocódigo desenvolvido possui uma função capaz de realizar comparações entre as leituras obtidas pelo sensor nas direções determinadas. O pseudocódigo responsável por esta ação apresenta-se na figura 38.

Figura 39 - Função de melhor distância

```
char calculaMelhorDistancia() {
    int esquerda = calcularDistanciaEsquerda();
    int centro = calcularDistanciaCentro();
    int direita = calcularDistanciaDireita();
    reposicionaServoSonar();
    int maiorDistancia = 0;
    char melhorDistancia = '0';

    if (centro > direita && centro > esquerda) {
        melhorDistancia = 'c';
        maiorDistancia = centro;
    } else if (direita > centro && direita > esquerda) {
        melhorDistancia = 'd';
        maiorDistancia = direita;
    } else if (esquerda > centro && esquerda > direita) {
        melhorDistancia = 'e';
        maiorDistancia = esquerda;
    }
    if (maiorDistancia <= 15) {
        rotacao_Re();
        posicionaCarroMelhorCaminho();
    }
    reposicionaServoSonar();
    return melhorDistancia;
}
```

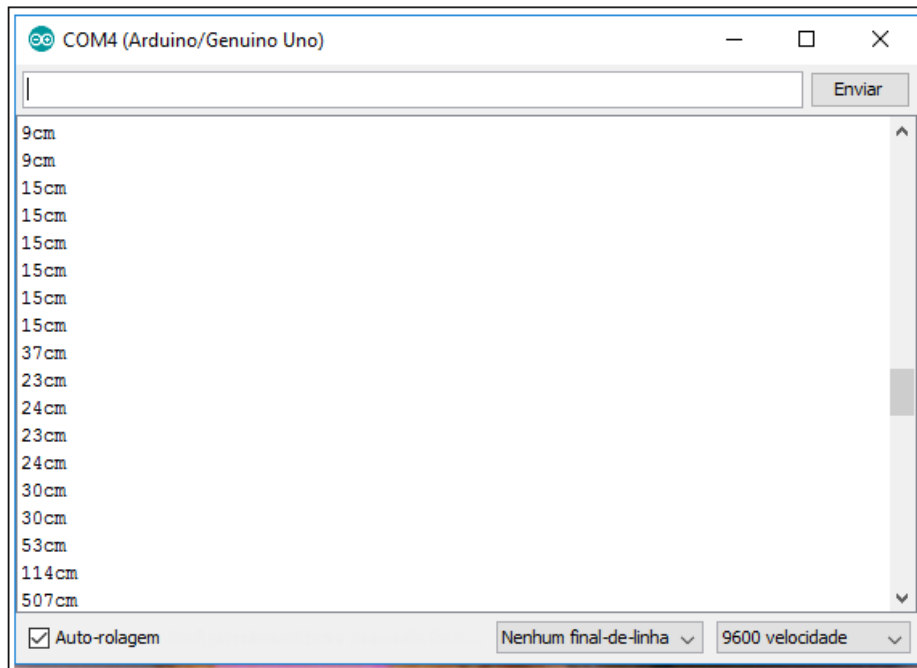
Fonte: Do autor.

7 ANÁLISE DOS RESULTADOS OBTIDOS

Conforme a literatura de Kridi et al. (2014), a utilização de sensores infravermelhos apresentam valores inconsistentes para a função de captura de obstáculos, portanto os mesmos propuseram a utilização de sonares para trabalhos futuros. Portanto, neste projeto utilizou-se de um sensor ultrassônico para realizar as leituras de distância em relação aos obstáculos. Com base nos testes realizados durante o processo de desenvolvimento verificou-se que o mesmo apresentava inconsistência em suas leituras, para amenizar este problema optou-se por diminuir a quantidade de leituras realizadas para aplicar max-min de cada leitura, então observou-se que o tempo de resposta do sensor ultrassônico melhorou significativamente para a detecção de obstáculos, sendo assim capaz de executar sua tarefa sem dificuldade.

Na figura 40 observa-se os resultados obtidos referente as leituras realizadas na figura 29 após alterações no pseudocódigo. Deste modo, nota-se, que conforme o sensor realiza as leituras o serial monitor da própria plataforma arduino apresenta os valores na tela do mesmo, neste caso a distância foi medida em cm.

Figura 40 - Leitura do sensor verificada no monitor serial



Fonte: Do autor.

Sugerido por Martin (2009), optou-se por utilizar um servomotor para movimentar o sensor, sendo possível a obtenção das informações em relação as distâncias de todas as direções desejadas. O método sugerido atingiu as expectativas esperadas para esta etapa do trabalho, e auxiliou na redução de custos do projeto, pois não houve a necessidade de mais sensores acoplados no projeto.

A plataforma arduino utilizada no projeto mostrou-se ser capaz de realizar este modelo de desenvolvimento, pois o mesmo permite conexões com dispositivos externos assim tornando-o uma ferramenta para qualquer área de pesquisa envolvendo robótica móvel ou não. O *software* do arduino permitiu que fosse possível realizar testes de distância e velocidade por meio do monitor serial como mostra a figura 41 que mostra um teste realizado, e também permitiu-se realizar aplicações de bibliotecas para agilizar no processo de desenvolvimento do mesmo, deste modo a plataforma arduino tanto de *hardware* quanto de *software* surpreenderam com as suas possibilidades de utilização no projeto.

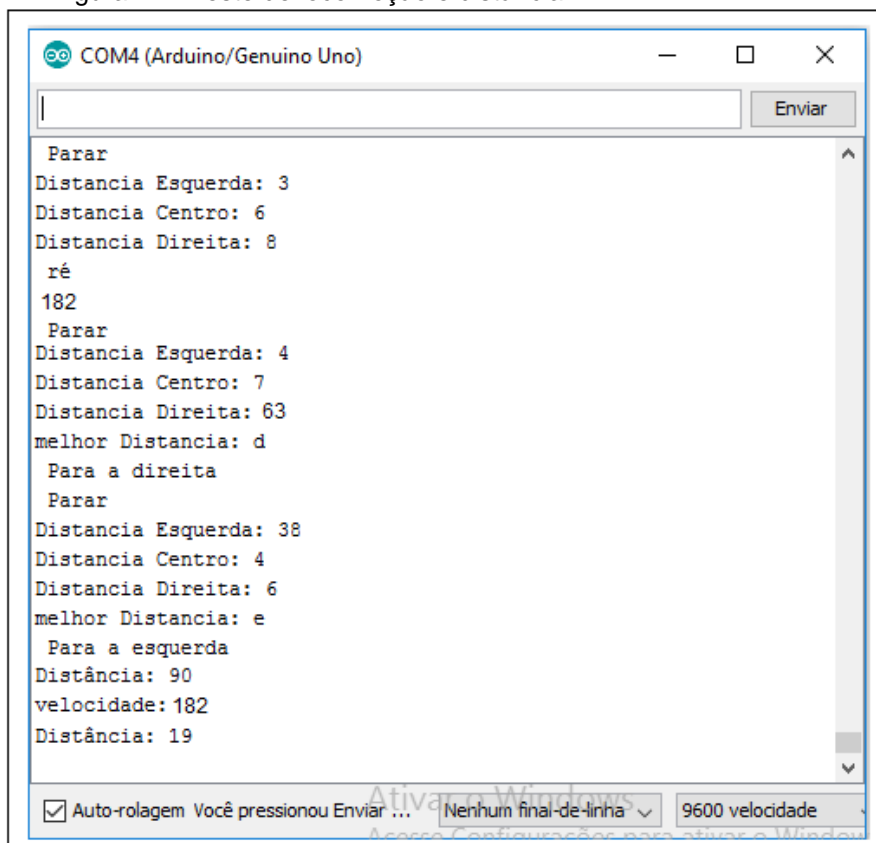
Os testes realizados no protótipo em relação aos obstáculos encontrados pode-se observar por meio do serial monitor do arduino o

comportamento da velocidade em relação à distância do obstáculo, também verificou-se o comportamento do robô quando a distâncias eram menores que 15 cm como mostra a figura 41, na figura 40 apresenta-se um dos testes realizados no sensor HC-SR04.

No mesmo teste de locomoção, pode-se observar que o modelo mostrou-se inconsistente em territórios desnivelados devido sua roda castor, porém em locais nivelados o mesmo apresentou consistência na locomoção.

Na figura 41 apresenta-se por meio do serial monitor os dados de captura e as direções tomadas durante o teste de locomoção.

Figura 41 - Teste de locomoção e distância



Fonte: Do autor.

8 CONCLUSÃO

Este é um projeto desafiador, pois existem diversas áreas de estudo associadas, tal como: mecânica, eletrônica, computação entre outras.

Portanto este trabalho demonstrou como a lógica *fuzzy*, técnica de inteligência artificial, aplicada a plataforma arduino, pode contribuir para a área de robótica móvel que possuem informações vagas e imprecisas. O mesmo apresentou os conceitos de robótica móvel e de lógica *fuzzy*, resultando no desenvolvimento de um protótipo de robô móvel autônomo para o controle de velocidade em relação a distância do obstáculo encontrado no ambiente por meio da biblioteca eFLL.

Devido ao grande campo de atuação da teoria *fuzzy*, não foi a intenção de delinear o assunto e sim como ocorre seu funcionamento.

Durante o projeto, algumas dificuldades foram encontradas tanto na parte teórica quanto na prática. Em relação a teoria, ainda é pequeno o número de materiais que abordam sobre a biblioteca eFLL, apesar de possuir uma quantidade satisfatória de artigos científicos abordando lógica *fuzzy*, o que dificultou a compreensão do funcionamento interno da mesma. Na parte prática, foi determinar quais valores determinariam cada conjunto *fuzzy*, visto que as ações de controle de velocidade em relação ao obstáculo são quase sempre vagas e imprecisas. A plataforma arduino quanto o sensor ultrassônico

permitiram que fosse possível desenvolver um protótipo aplicando a lógica *fuzzy*, sendo assim, ferramentas viáveis para este tipo de projeto.

Com relação ao projeto desenvolvido, os resultados obtidos foram satisfatórios, atingindo o objetivo geral quanto os específicos que consistiam no controle de velocidade em relação ao obstáculo por meio de lógica *fuzzy*, utilização da plataforma arduino, testes para locomoção, sendo possível desenvolver um protótipo de robô móvel autônomo.

Como sugestão para continuidade deste projeto propõe-se:

- i. inserir uma câmera própria para a plataforma arduino;
- ii. implementação de sensores para medição de gases e temperaturas de ambientes;
- iii. controle de bateria;
- iv. utilização de outras plataformas embarcadas como por exemplo *Raspberry Pi*.

REFERÊNCIAS

ALECRIM, Paulo D. de et al . **LOW COST FUZZY SYSTEM APPLIED TO THE CONTROL AND SUPERVISION OF THERMAL ENVIRONMENT IN POULTRY FARMS**. Eng. Agríc., Jaboticabal , v. 37, n. 1, p. 194-205, Feb. 2017 . Available from <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-69162017000100194&lng=en&nrm=iso>. access on 04 June 2017. <http://dx.doi.org/10.1590/1809-4430-eng.agric.v37n1p194-205/2017>.

ALMEIDA, T. S. et al. **Inteligência artificial aplicada à navegação autônoma de robôs**. In: SEMANA DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE SERGIPE, 2., 2012, Itabaiana. Anais eletrônicos... Itabaiana: DSI/UFS, 2012. Disponível em: <www.si.itatechjr.com.br/images/docs/SEMINFO_UFSITA_2012.pdf>. Acesso em: 11 out. 2015.

ARDUINO, Criando Robô Com. **Criando Robô com Arduino**. 2014. Disponível em: <http://www.criandorobocomarduino.com/2014/10/caraut-v2-criando-um-robo-autonomo-com_24.html>. Acesso em: 06 jul. 2016.

ARDUINO. **Site Oficial**. Disponível em: <<http://www.arduino.cc/>>. Acesso em: 27 mar. 2016.

ASSEF, Amauri A. et al. **Sistema parágrafo Geração, Aquisição e Processamento de Sinais de ultra-som**. Sba Controle & Automação , Natal, v. 20, n. 2, p. 145-155, Junho de 2009. Disponível a partir http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592009000200003&lng=en&nrm=iso. Acesso em 07 de mai. de 2016.

BARROS, L. C.; BASSANEZI, R. C. 2006. **Tópicos de Lógica Fuzzy e Biomatemática**, volume 5 Coleção IMECC - Textos Didáticos. IMECC-UNICAMP, Campinas/SP.

BASCONCELLO ,Filho O. Daniel. **Curso de Arduino**. Disponível em http://www.robotizando.com.br/curso_arduino_hardware_pg1.php Acesso em 27 mar. 2016

BATISTA, Alan Vinícius de Araújo. **ROBÔ IRRIGADOR MULTIFUNCIONAL DE BAIXO CUSTO PARA AGRICULTURA FAMILIAR (RIRRIG)**. 2016. 91 f. Dissertação (Mestrado) - Curso de Engenharia Agrícola, Departamento de Engenharia Agrícola, Universidade Federal do Ceará, Fortaleza, , 2016. Disponível em: http://repositorio.ufc.br/bitstream/riufc/18757/1/2016_dis_avabatista.pdf. acesso em: 25 mar. 2017.

BILOBROVEC, Marcelo; MARÇAL, Rui Francisco Martins; KOVALESKI, João Luis. Implementação de um sistema de controle inteligente utilizando a lógica fuzzy. **XI SIMPEP, Bauru/Brasil**, 2004.

BIONDI NETO, Luiz et al. SISTEMA ESPECIALISTA NEBULOSO: MINICURSO. In: XXXVIII SIMPÓSIO BRASILEIRO PESQUISA OPERACIONAL, 38., 2006, Goiânia. **Pesquisa Operacional na Sociedade: Educação, Meio Ambiente e Desenvolvimento**. Goiânia, Go: Sobrapo, 2006. p. 2508 - 2547. Disponível em: http://www.uff.br/decisao/SBPO_Fuzy.pdf. Acesso em: 24 jun. 2016.

BISC. **The Berkeley Initiative in Soft Computing**. California: University of Berkeley. Disponível em: <http://people.eecs.berkeley.edu/~zadeh/stimfl.html>. Acesso em: 04 jun. 2016.

BITTENCOURT, Claudemir Godoy de. **APLICAÇÃO DA LÓGICA FUZZY EM UM SISTEMA DE PRÉ-ANÁLISE DE OSCILOGRAFIAS EM UNIDADES DE GERAÇÃO**. 2011. 75 f. TCC (Graduação) - Curso de Engenharia Elétrica, Coordenação de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná, Pato Branco, Pr, 2011. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/4074/1/PB_COELT_2011_2_01.pdf. Acesso em: 20 jun. 2016.

BORBA, Thiago. **Arduino, Lógica Fuzzy!!!: Sensacional!!**. 2013. Disponível em: <<http://projetosdoborba.blogspot.com.br/2013/06/arduino-logica-fuzzy-sensacional.html>>. Acesso em: 10 out. 2016.

BRAGA, Frederico Castro. **ESTUDO PARA FABRICAÇÃO DE UM PROTÓTIPO DE ROBÔ MÓVEL COM RODAS PARA ASPIRAÇÃO DE PÓ DOMÉSTICA**. 2014. 128 f. TCC (Graduação) - Curso de Engenharia Mecânica, Engenharia Mecânica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014. Disponível em: <<http://monografias.poli.ufrj.br/monografias/monopoli10012128.pdf>>. Acesso em: 10 out. 2016.

BRUGNARI, A.; MAESTRELLI, L. H. M., **AUTOMAÇÃO RESIDENCIAL via WEB**. 2010, 36f. Trabalho de conclusão de curso – Curso de Graduação em Engenharia de Computação – PUC-PR, Curitiba, 2010.

CIÊNCIA & DESENVOLVIMENTO: Aplicação da Lógica Fuzzy no controle da velocidade de um protótipo móvel. Revista Eletrônica da Fainor: Geffer Thiago Batista Correa, 2016. Semestral. Ciência & Desenvolvimento Está Indexada nas Seguintes Bases: Latindex (sistema Regional de Información En Línea Para Revistas Científicas de América Latina, El Caribe, España y Portugal). Doaj (directory Of Open Access Journals). Rcaap (repositório Científico de Acesso Aberto de Portugal). SumÁrios.org (sumários de Revistas Brasileiras). Drji (directory Of Research Journal Indexing). Disponível em: <<http://srv02.fainor.com.br/revista/index.php/memorias/issue/view/16>>. Acesso em: 03 mar. 2016.

CONGRESSO DA SOCIEDADE BRASILEIRA DA COMPUTAÇÃO, 34., 2014, Brasília. **Revista da Sociedade Brasileira de Computação COMPUTAÇÃO BRASIL VEÍCULOS AUTÔNOMOS NÃO TRIPULADOS Conheça esta nova tecnologia que começa a ganhar força no Brasil: SISTEMAS ABERTOS NO DESENVOLVIMENTO DE VEÍCULOS AUTÔNOMOS**. Porto Alegre, Rs: Sbc, 2014. 155 p. Disponível em: <<http://sbc.org.br/component/flippingbook/book/15>>. Acesso em: 18 jun. 2016.

CORTELLETI Daniel - **Introdução à programação de microcontroladores Microchip PIC**, Senai, Rio Grande do Sul, 2006.

CRUZ, Shermila Guerra Santa. **Sistema de Navegação para Veículo Autônomo usando Lógica Difusa**. 2009. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18152/tde-09092009-082215/publico/Shermila.pdf>>. Acesso em: 23 jun. 2016.

DADONE P.(2001), **Design Optimization of Fuzzy Logic Systems**. PhD Dissertation, Virginia Polytechnic Institute, p. 87-93. Blacksburg, Virginia. Maio, 2001.

DE PIERI, EDSON ROBERTO. "**Curso de Robótica Móvel.**" Santa Catarina, Florianópolis (2002).

DIAS, W. S.; COSTA, B. S.; CARVALHO, C. A. **Construção de um Robô Veicular Autônomo**. In: VII Congresso Norte Nordeste de Pesquisa e Inovação, 2012, Palmas - TO. Anais VII Connepi 2012, 2012. Disponível em: <https://pt.scribd.com/user/102658780/Manuel-Pereira>. Acessado em: 12/10/2015.

DUDEK, G.; JENKIN, M. **computational principles of mobile robotics**. Cambridge University Press, 280p, 2000.

ELECFREAKS. **Ultrasonic Ranging HC Module - SR04**. Disponível em: <<http://www.micropik.com/PDF/HCSR04.pdf>>. Acesso em: 23 abr. 2016.

ELECTROSOME. **Interfacing HC-SR04 Ultrasonic Sensor with PIC Microcontroller**. 2014. Disponível em: <<https://electrosome.com/hc-sr04-ultrasonic-sensor-pic/>>. Acesso em: 23 abr. 2016.

ELETRÔNICA. **Placa Arduino - Uma das grandes evoluções da Eletrônica**. 2014. Copyright © 2014 Eletrônica 90. Disponível em: <<http://eletronica90.blogspot.com.br/2014/09/placa-arduino-uma-das-grandes-evolucoes.html>>. Acesso em: 17 maio 2016.

ESTADOS UNIDOS DA AMÉRICA. Georgia Institute Of Technology. Caucus Comitê Consultivo Robotics do Congresso dos Eua (Org.). **A Roadmap for U.S. Robotics From Internet to Robotics**. 2. ed. Washington, Eua: Georgiatech, 2013. 129 p. Disponível em: <[https://robotics-vo.us/sites/default/files/2013 Robotics Roadmap-rs.pdf](https://robotics-vo.us/sites/default/files/2013%20Robotics%20Roadmap-rs.pdf)>. Acesso em: 18 mar. 2016.

GARCIA, André Luís Jorge. **Implementação eletrônica de sistemas Fuzzy**. 2009. 114 f. Dissertação (Mestrado) - Curso de Engenharia Eletrônica, Centro de Tecnologia e Ciências, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2009. Disponível em: <http://www.pel.uerj.br/bancodissertacoes/Dissertacao_Andre_Garcia.pdf>. Acesso em: 23 jun. 2016.

GARCIA, Pauli Adriano de Almada; MELO, P. F. Frutuoso e; SCHIRRU, R.. **Aplicação de um modelo fuzzy DEA para priorizar modos de falha em sistemas nucleares**. *Pesqui. Oper.*, Rio de Janeiro , v. 29, n. 2, p. 383-402,

Aug. 2009 . Available from
<http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382009000200007&lng=en&nrm=iso>. acesso em 21 abr. 2017.
<http://dx.doi.org/10.1590/S0101-74382009000200007>.

GITHUB. **Zerokol/eFLL: eFLL (Embedded Fuzzy Logic Library) is a standard library for Embedded Systems.** Zerokol authored on 16 May 2014. Disponível em: <https://github.com/zerokol/eFLL> Acesso em: 15 mar. 2017.

GOMES, Dennis dos Santos. **Inteligência Artificial: Conceitos e Aplicações.** Revista Olhar Científico, Ariquemes, Ro, v. 1, n. 2, p.234-246, 17 nov. 2010. Semestral. Disponível em:
<<http://www.olharcientifico.kinghost.net/index.php/olhar/article/view/49/37>>. Acesso em: 19 jun. 2016.

GUDWIN, R.R. (2005). **Novas Fronteiras na Inteligência Artificial e na Robótica.** 4º Congresso Temático de Dinâmica, Controle e Aplicações Campinas:SP.:<http://www.dca.fee.unicamp.br/%7Egudwin/ftp/publications/Dinc%7Egudwin.pdf>>. Acesso em: 08 out. 2015.

HALFPAP, Dulce Maria; SOUZA, Gilberto Corrêa de; DA MOTA ALVES, João Bosco. **Robôs como artefatos. Ciênc. cogn.,** Rio de Janeiro , v. 12, p. 203-213, nov. 2007. Disponível em
http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1806-58212007000300019&lng=pt&nrm=iso. Acesso em 29 mai. 2016

HORTEGAL, M.V. (2016). **Aplicação da Lógica Fuzzy no Controle do Desempenho de Estacas Hélice Contínua.** Tese de Doutorado, Publicação G.TD-126/2016, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 157 p. Disponível em:
<http://repositorio.unb.br/handle/10482/22636?mode=full> acesso em 28 mai. 2017.

JUNG, C. R.; Osório, F. S.; Kelber, C.; Heinen, F. (2005) **“Computação embarcada: Projeto e implementação de veículos autônomos inteligentes”**, In: Anais do CSBC’05 XXIV Jornada de Atualização em Informática (JAI). São Leopoldo, RS: SBC, v. 1, p. 1358–1406.
<http://www.lbd.dcc.ufmg.br/colecoes/jai/2005/008.pdf>>. Acesso em: 04 out. 2015.

KARVINEN, Kimmo; KARVINEN, Tero. **Primeiros Passos com Sensores.** São Paulo, Sp: Novatec, 2014. 160 p. Tradução de: Tero Karvinen, Kimmo Karvinen. Disponível em:
<<https://novatec.com.br/livros/sensores/capitulo9788575224021.pdf>>. Acesso em: 09 abr. 2016.

KOHAGURA, Tiago. **Lógica Fuzzy e suas Aplicações**. 2007. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Departamento de Computação, Universidade Estadual de Londrina, Londrina, 2007. Disponível em <<http://www2.dc.uel.br/nourau/document/?view=601>>. Acesso em: 25 mai. 2016.

KRIDI, Douglas Santiago et al. **DESENVOLVIMENTO DE UMA BIBLIOTECA FUZZY PARA O CONTROLE AUTÔNOMO DE UM ROBÔ MÓVEL EM AMBIENTE DESCONHECIDO**. Universidade Estadual do Piauí – UESPI / Laboratório de Informática Aplicada - LAIC. Disponível em: <<http://sistemaolimpo.org/midias/uploads/13fdd3bf2d22ab0aeecff212df10bd4c.pdf>> Acesso em: 17 mar. 2017.

LONGO, Lucas. **INTERNET DAS COISAS: USO DE SENSORES E ATUADORES NA AUTOMAÇÃO DE UM PROTÓTIPO RESIDENCIAL**. 2015. 97 f. TCC (Graduação) - Curso de Engenharia de Computação, Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Pato Branco, 2015. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/5899/1/PB_COENC_2015_2_02.pdf>. Acesso em: 03 abr. 2017.

LUGER, George F. **Inteligência artificial**. 6. ed São Paulo: Pearson Education do Brasil, 2013.

MARTIN, William. **Autonomous robot obstacle avoidance using a fuzzy logic control scheme**. 2009. Disponível em: <http://vision.middlebury.edu/~wcmartin/files/fuzzy_control_robot.pdf>. Acesso em 14 mar. 2017

MARTINS, Wander Ferreira. **CONTROLE FUZZY EM TEMPO REAL, APLICADO AO SISTEMA PLATAFORMA-ESFERA**. 2007. 112 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Ciências em Engenharia Elétrica, Instituto Militar de Engenharia, Rio de Janeiro, 2007. Disponível em: <http://www.pgee.ime.eb.br/pdf/wander_martins.pdf>. Acesso em: 21 abr. 2017.

MCROBERTS, Michael. **Arduino básico**. São Paulo: Novatec Editora, 2011.

MERIAM, J. I.; kraige, I. g. **Mecânica para engenharia**. 6. ed. Rio de Janeiro: LTC, 2009.

MIRANDA; Edvan Cordeiro de et al. **Influência onda Arc Movimento no nível de diluição e no cordão de solda Geometry Usando Processo de Solda TIG com fio automática Fria alimentação**. Soldag. Insp. , São Paulo, v. 20, n. 2, p. 180-190, junho De 2015. Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-92242015000200180&lng=pt&nrm=iso>. acessos em 19 de junho 2016. <http://dx.doi.org/10.1590/0104-9224/SI2002.06>.

MONARD, Maria Carolina; BARANAUKAS, José Augusto. **Aplicações de Inteligência Artificial: Uma Visão Geral**. São Carlos: Instituto de Ciências Matemáticas e de Computação de São Carlos, 2000.

MOZELLI, Leonardo Amaral. UFMG, Belo Horizonte, 03 de outubro de 2008, Controle **Fuzzy para Sistemas Takagi-Sugeno; Condições aprimoradas e aplicações**. Disponível em: http://www.cpdee.ufmg.br/~palhares/dissertacao_mozelli_final.pdf Acesso em 05 jun. 2016.

MUNDO ROBÓTICA: Revista Oficial da Olimpíada Brasileira de Robótica. Sorocaba, Sp: Obr, v. 4, 2015. Quatrimestre. Disponível em: <<http://www.obr.org.br/wp-content/uploads/2015/04/Mundo-Robotica-4.pdf>>. Acesso em: 16 maio 2016.

NASCIMENTO, Susana. Automatizações no inorgânico: aproximações ao estudo social de criaturas artificiais. **Anál. Social**, Lisboa , n. 181, p. 1033-1056, 2006. Disponível em <http://www.scielo.mec.pt/scielo.php?script=sci_arttext&pid=S0003-25732006000400004&lng=pt&nrm=iso>. acessos em 23 jun. 2016.

OTTONI, Guilherme de Lima; LAGES, Walter Fetter. **Navegação de robôs móveis em ambientes desconhecidos utilizando sonares de ultra-som**. Sba Controle & Automação, Campinas , v. 14, n. 4, p. 402-411, dez. 2003 . Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592003000400008&lng=pt&nrm=iso>. Acesso em 29 mai. 2016. <http://dx.doi.org/10.1590/S0103-17592003000400008>.

PEREIRA, L. A. M. **Robô autômato para monitoramento de rebanho e medição da forragem do pasto**. São Paulo-SP: <http://www.teses.usp.br/teses/disponiveis/74/74131/tde-12042010-132133/pt-br.php> . Acesso em: 16 abr. 2016.

PEREZ, Anderson Luiz Fernandes et al. **Uso da Plataforma Arduino para o Ensino e o Aprendizado de Robótica**. In: International Conference on Interactive Computer aided Blended Learning (ICBL). 2013.

POFFO, Robinson et al. Cirurgia robótica em Cardiologia: um Procedimento seguro e efetivo. **Einstein (São Paulo)**, São Paulo, v 11, n.. 3, p. 296-302, setembro de 2013. Disponível em:
<http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1679-45082013000300007&lng=en&nrm=iso>. Acesso em 19 de junho de 2016.
<http://dx.doi.org/10.1590/S1679-45082013000300007>.

POZZEBON, FRIGO. **Robótica no Processo de Ensino e Aprendizagem**. In: International Conference on Interactive Computer aided Blended Learning (ICBL). 2013.

PRIMO, Alex; COELHO, Luciano Roth. **Comunicação e inteligência artificial: interagindo com a robô de conversação Cybelle**. In: MOTTA, L. G. M. et al. (Eds.). **Estratégias e culturas da comunicação ed.Brasília**. Brasília: Editora Universidade de Brasília, 2002, p. 83-106.

RIGNEL, Diego Gabriel de Sousa; CHENCI, Gabriel Pupin; LUCAS, Carlos Alberto. UMA INTRODUÇÃO A LÓGICA *FUZZY*. **Resiget: Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica**, On-line, v. 01, p.17-28, 01 Não é um mês valido! 2011. Disponível em:
<http://www.logicafuzzy.com.br/wp-content/uploads/2013/04/uma_introducao_a_logica_fuzzy.pdf>. Acesso em: 09 out. 2016.

ROBÔ, Seu. **Criando robo com arduino: Criando um robô autônomo com Shield L293D e Arduíno**. 2014. Disponível em:
<<http://www.criandorobocomarduino.com/>>. Acesso em: 09 abr. 2017.

ROBÓTICA LIVRE. **Introdução à programação de microcontroladores**. 2011. Disponível em:
http://www.roboticalivre.com/index.php?option=com_flexicontent&view=items&cid=58&i%20d=70. Acesso em: 06 abr. 2016.

RODRIGUES, Marcelo. **Laboratório de Garagem: Desenvolvedores Independentes de Ciência e Tecnologia**. 2016. Disponível em:
<<http://labdegaragem.com/forum/topics/otimizar-c-digo-arduino-com-registradores-port>>. Acesso em: 26 mar. 2016.

ROMANO, Vitor Ferreira. **Robótica industrial: aplicação na indústria de manufatura e de processos**. São Paulo: Edgard Blücher, 2002. 256 p.

ROSÁRIO, João Maurício. **Princípios de mecatrônica**. Pearson Prentice Hall, 2005.

RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial**. 2. Ed. Rio de Janeiro: Campos, 2004.

RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial**. Rio de Janeiro, Elsevier, 2013.

SABBATINI, Renato M.e.. **W. Grey Walter: A Machina speculatrix**. 1999. Disponível em: <<http://www.cerebromente.org.br/n09/historia/documentos.htm>>. Acesso em: 23 jun. 2016.

SANTOS, Alexsandro José Virgínio dos. **ANÁLISE E CONTROLE DE UM VEÍCULO ROBÓTICO TRACIONADO POR ESTEIRAS**. 2015. 140 f. Tese (Doutorado) - Curso de Engenharia Mecânica, Universidade Federal da Paraíba, João Pessoa, 2015. Disponível em: <<http://tede.biblioteca.ufpb.br/bitstream/tede/7577/2/arquivototal.pdf>>. Acesso em: 04 nov. 2016.

SECCHI, Humberto Alejandro. **Una Introducción a los Robots Móviles**. 2008. 91 f. Monografia (Especialização) - Curso de Instituto de Automática – Inaut, Universidade Nacional de San Juan, San Juan, 2008. Disponível em: <http://www.obr.org.br/wp-content/uploads/2013/04/Una_Introducao_aos_Robos_Moveis.pdf>. Acesso em: 21 jul. 2016.

SHAW, Ian S ; Simões, Marcelo Godoy. **Controle e modelagem fuzzy**. 2. ed. rev. A ampl São Paulo: Blucher, 2007. Xiv, 186 p.

SILER, William; BUCKLEY, James J. **Fuzzy expert systems and fuzzy reasoning**. New Jersey: Wiley, c2005. 405 p. ISBN 0471388599.

SILVA, Joelson Coelho da. **Uma Proposta de Controle Neural Adaptativo para a Navegação de Veículos Autônomos**. 1999. 101 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 1999. Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/18631/000728938.pdf?sequence=1>>. Acesso em: 20 mar. 2016.

SILVEIRA, J. A. D. **Arduino- Cartilha para Programacao em C**. Revista do arduino, 2012. Disponível em: <www.revistadoarduino.com.br>. Acesso em: 02 mai. 2016.

SOUZA, Anderson R. de et al . **A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC**. Rev. Bras. Ensino Fís., São Paulo, v.33, n.1, p.01-05, mar. 2011. Disponível em http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172011000100026&lng=pt&nrm=iso. Acesso em 13 mar. 2016. www.obr.org.br/wp-content/uploads/2015/04/Mundo-Robotica-4.pdf.

STMICROELECTRONIC. **L293D L293DD: PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES**. 2003. Disponível em: <<https://www.arduino.cc/documents/datasheets/L293D.pdf>>. Acesso em: 15 maio 2017.

TECHNOLOGYES, Omni Vision. **OV7670 Datasheet**. Sunnyvale, Ca Usa: Orleans Drive, 2005. 30 p. Disponível em: <<https://www.voti.nl/docs/OV7670.pdf>>. Acesso em: 16 maio 2017.

YVKOFF, L. **Many car buyers show interest in autonomous car tech**. (2012). Disponível em: <https://www.cnet.com/roadshow/news/many-car-buyers-show-interest-in-autonomous-car-tech/> acessado em 15 out. 2016.

WANG, L i-Xin. **A couse if fuzzy systems and control**. London: Prentice Hall, 1997. 424 p. ISBN 0135408822.

WERNECK, Pedro. **Introdução ao Arduino**. 2009. Disponível em: <http://www.sabereletronica.com.br/secoes/leitura/1307>. Acesso em : 18 fev. 2017

WOLF, Denis et al. **Robótica Móvel Inteligente:: Da Simulação às Aplicações no Mundo Real**. In: CONGRESSO DA SBC – SOCIEDADE BRASILEIRA DE COMPUTAÇÃO (BRASIL), 29., 2009, Bento Gonçalves. **Anais...** . Bento Gonçalves: Sbc, 2009. p. 279 - 330. Disponível em: <http://osorio.wait4.org/palestras/texto/JAI2009_Completo_Revisado.pdf>. Acesso em: 12 mar. 2017.

ZADEH, L. A. **Fuzzy sets: Information and Control**. [S.l.]: Berkeley : Elsevier, v. 3, 1965. p. 239 - 354.

APÊNDICE(S)

APÊNDICE A – ARTIGO

Robô Móvel Utilizando Lógica Fuzzy na Plataforma Arduino

**Rafael S. de Oliveira¹, Paulo João Martins¹ Merisandra Côrtes de Mattos
Garcia¹**

¹Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias (UNACET) – Universidade do Extremo Sul Catarinense (UNESC) – Av. Universitária, 1105 – Caixa Postal 3167 – 88806-000 – Santa Catarina – SC – Brasil

rafael-el-el@hotmail.com, pjm@unesc.net, mem@unesc.net

Abstract. *Robotics is a field that is increasingly present for military as well as social purposes, such as health, domestic, among others. With the advances a subarea of this field has been emphasizing that they are the creations of autonomous mobile robots, with intelligence applied or not. Therefore this article describes the development of a standalone vehicle prototype applying fuzzy logic to its displacement by means of an Arduino microcontroller.*

Resumo. *A robótica é um campo que está cada vez mais presente tanto para fins militares, quanto sociais, como saúde, domésticos, entre outros. Com os avanços uma subárea deste campo vem se destacando que são as criações de robôs móveis autônomos, com inteligência aplicada ou não. Portanto este artigo descreve o desenvolvimento de um protótipo de veículo autônomo aplicando lógica fuzzy para o seu deslocamento por meio de um microcontrolador arduino.*

1. Introdução

A robótica utiliza-se de um conjunto vasto de sensores como, câmeras, ultrassom, infravermelho entre outros, que permitem identificar o ambiente. Os robôs móveis deslocam-se por meio de rodas, pernas, esteiras, ou mecanismos semelhantes, assim permitindo-o a utilização dos mesmos para os mais diversos fins (RUSSELL; NORVIG, 2013). Segundo Almeida (2012) e Luger (2013), apesar de possuir uma vasta quantidade de pesquisa sobre o assunto, o tema continua sendo desafiador, pois realizar uma tarefa de alto nível como, locomover-se em uma sala com obstáculo necessita de uma sequência de ações atômicas para realizá-la.

Segundo Gudiwin (2005) e Wolf et al. (2009), robôs móveis são objetos de intensa atividade de pesquisa, devido a facilidade de realizar testes por meio de simuladores. Outro fator importante é a complexidade específica ao seu funcionamento, assim tornando-o um laboratório de pesquisa para a inteligência artificial.

O robô móvel pode ser utilizado em uma vasta gama de tarefas, desde transporte de peças até o reconhecimento de ambientes que oferecem riscos, como crateras, ambientes espaciais, áreas radioativas, entre outros. O que difere deste modelo para os robôs manipuladores é a sua liberdade de locomoção. Portanto, o seu deslocamento exige um maior grau de comunicação com o ambiente, o que necessita da interação de sensores para a captação do ambiente e eventualmente a detecção do resultado das ações do próprio robô.

2. Metodologia

A área da robótica, mais especificamente a de robótica móvel vem apresentando uma evolução de seus métodos tecnológicos, conforme mencionado anteriormente. O avanço desta área beneficia a sociedade em geral, devido a sua vasta área de aplicação. No desenvolvimento deste protótipo aplicou-se os conceitos aprendidos durante a pesquisa. Este capítulo apresenta a modelagem e montagem do desenvolvimento do protótipo idealizado a partir dos conceitos aqui abordados, ele se divide em quatro sessões: a primeira aborda os componentes utilizados para a montagem da estrutura do protótipo, a segunda apresenta os testes do sensor HC-SR04, no terceiro apresenta a aplicação de lógica fuzzy, como suas regras, defuzzificação, variáveis linguísticas, entre outras, e por fim, na quarta sessão são apresentados os métodos responsáveis por determinar as direções a serem realizadas.

Apresenta-se na figura 1 o funcionamento interno do protótipo por meio de um fluxograma de atividade.

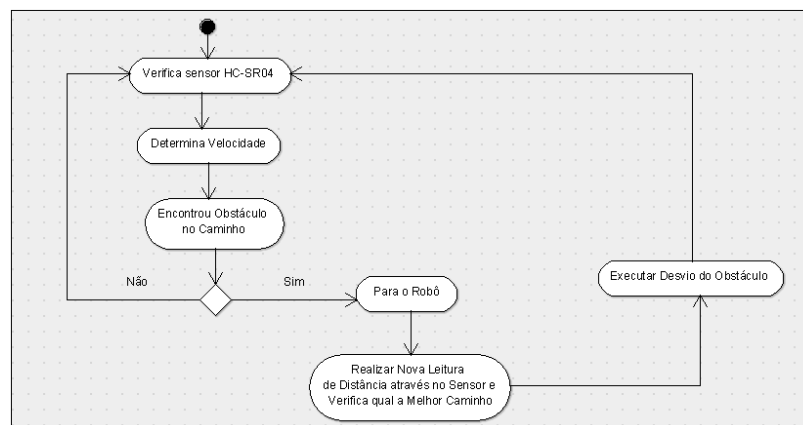
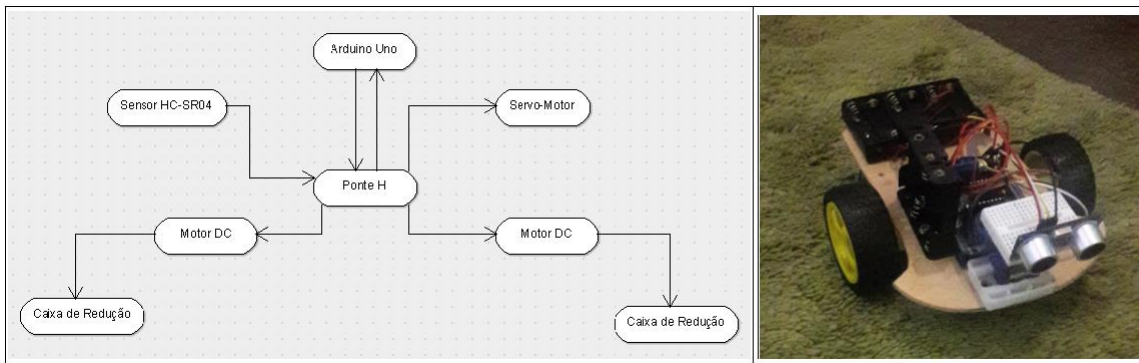


Figura 1 - Fluxograma do funcionamento interno

2.1 Montagem da Estrutura

O modelo da estrutura do protótipo como pode-se observar na figura 2, foram interligados por meio de seus pinos o motor *shield* l293d - driver ponte H e o arduino UNO, logo após conectou-se os dois motores DC na ponte H que possuem as caixas de redução em suas extremidades para o encaixe das rodas. Posteriormente, foram conectados o servomotor e o sensor ultrassônico para a realização das leituras no ambiente.



Figur

a 2 - Estrutura do protótipo

2.2 Teste do Sensor HC-SR04

Para a validação da distância obtida pelo sensor ultrassônico HC-SR04 utilizou-se uma régua para verificação entre o sensor e o obstáculo, no caso uma caixa de papelão serviu como obstáculo. Ao deslocar-se a caixa sobre a régua é possível determinar a distância em cm entre a caixa e o sensor, para verificar se a leitura estava de acordo com a distância e utilizou-se o monitor serial da plataforma arduino onde possibilitou a verificação das leituras realizadas pelo sensor.



Figura 3 - Teste do Sensor

2.3 MODELAGEM FUZZY

A modelagem fuzzy compreendeu inicialmente a definição das funções de pertinência da variável de entrada e de saída do protótipo, posteriormente realizou-se a definição da base de regras, e as etapas propriamente ditas referentes ao desenvolvimento de um sistema fuzzy que compreende a fuzificação, avaliação das regras e defuzificação.

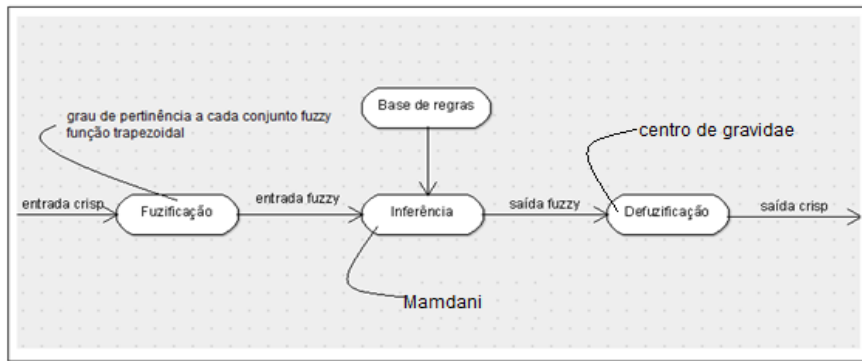


Figura 4 - Modelagem fuzzy

2.3.1 Definição das funções de pertinência

Durante a aquisição do conhecimento, definiu-se como variável de entrada a distância e como variável de saída a velocidade. Para a variável distância formou-se os conjuntos *fuzzy*:

Variável de entrada distância → Conjuntos $\left\{ \begin{array}{l} \text{Pequena} \\ \text{Mediana} \\ \text{Grande} \end{array} \right.$

Para a variável velocidade formou-se os conjuntos *fuzzy*:

Variável de saída velocidade → Conjuntos $\left\{ \begin{array}{l} \text{Baixa} \\ \text{Média} \\ \text{Rápida} \end{array} \right.$

As funções de pertinência para as variáveis de entrada e de saída do protótipo foram definidas por meio da função trapezoidal. Segundo Biondi Neto et al. (2006), o cálculo realizado pela função trapezoidal é dada por:

$$f(x,a,b,c,d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}$$

Onde:

x é o valor da variável distância;

a e d onde o grau de pertinência é 0;

b e c o ponto onde o grau de pertinência é 1.

A função de pertinência da variável de entrada mostra-se no modelo trapezoidal aplicado no desenvolvimento deste trabalho.

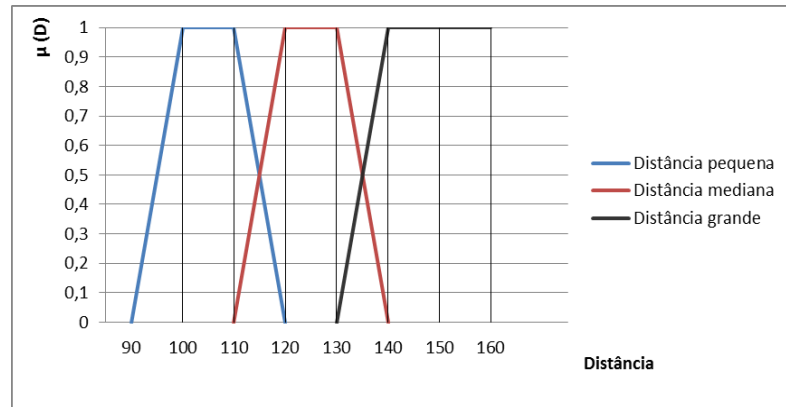


Figura 5 - Gráfico das funções de pertinência da variável distância

Tendo-se os seguintes conjuntos fuzzy definidos:

$$\mu(D)_{\text{pequena}} = \begin{cases} 0 & \text{se } D \leq 90 \text{ e } D \geq 120 \\ \frac{D-90}{100-90} & \text{se } 90 < D < 100 \\ \frac{120-D}{120-110} & \text{se } 110 < D < 120 \\ 1 & \text{se } 100 \leq D \leq 110 \end{cases}$$

$$\mu(D)_{\text{mediana}} = \begin{cases} 0 & \text{se } D \leq 110 \text{ e } D \geq 140 \\ \frac{D-110}{120-110} & \text{se } 110 < D < 120 \\ \frac{140-D}{140-130} & \text{se } 130 < D < 140 \\ 1 & \text{se } 120 \leq D \leq 130 \end{cases}$$

$$\mu(D)_{\text{grande}} = \begin{cases} 0 & \text{se } D \leq 130 \text{ e } D \geq 160 \\ \frac{D-130}{140-130} & \text{se } 130 < D < 140 \\ \frac{160-D}{160-150} & \text{se } 150 < D < 160 \\ 1 & \text{se } 140 \leq D \leq 150 \end{cases}$$

As funções de pertinência definidas para a variável de saída velocidade são formadas pelos conjuntos *fuzzy* baixa, média e rápida conforme a figura 6.

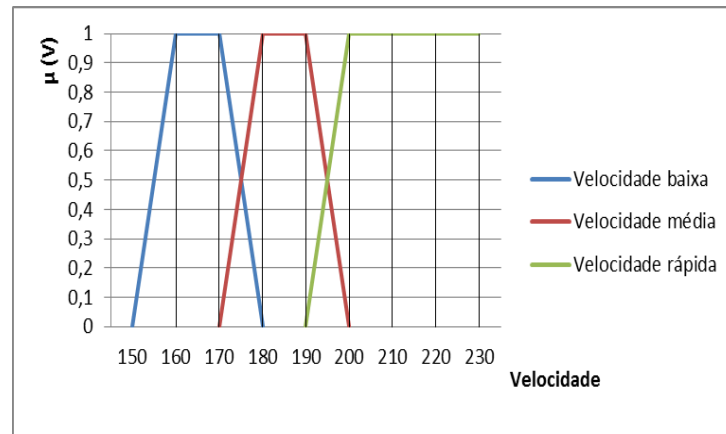


Figura 6 - Gráfico das funções de pertinência da variável velocidade

$$\mu(V)_{\text{baixa}} = \begin{cases} 0 & \text{se } V \leq 150 \text{ e } V \geq 180 \\ \frac{V-150}{160-150} & \text{se } 150 < V < 160 \\ \frac{180-V}{180-170} & \text{se } 170 < V < 180 \\ 1 & \text{se } 160 \leq V \leq 170 \end{cases}$$

$$\mu(V)_{\text{média}} = \begin{cases} 0 & \text{se } V \leq 170 \text{ e } V \geq 200 \\ \frac{V-170}{180-170} & \text{se } 170 < V < 180 \\ \frac{200-V}{200-190} & \text{se } 190 < V < 200 \\ 1 & \text{se } 180 \leq V \leq 190 \end{cases}$$

$$\mu(V)_{\text{rápida}} = \begin{cases} 0 & \text{se } V \leq 190 \text{ e } V \geq 220 \\ \frac{V-190}{200-190} & \text{se } 190 < V < 200 \\ \frac{220-V}{220-210} & \text{se } 210 < V < 220 \\ 1 & \text{se } 200 \leq V \leq 220 \end{cases}$$

6.3.2 Definição da Base de Regras

O número de regras aplicadas no protótipo foi encontrado por meio de um cálculo que utiliza-se da quantidade de conjuntos *fuzzy* e variável de entrada. Portanto o protótipo é composto por três regras, que é dada pela equação $1 \times 3 = 3$ o que sugere a formação:

Regras	Entradas		Saida
1	Se distância = pequena	Então	Velocidade = <i>baixa</i>
2	Se distância = <i>mediana</i>	Então	velocidade = <i>média</i>
3	Se distância = grande	Então	Velocidade = <i>rápida</i>

Tabela 1 - Base de regras

6.3.3 Fuzificação

A fuzificação consiste na identificação de um valor fuzzy a cada um dos conjuntos da variável de entrada, considerando-se um valor *crisp* de entrada. Portanto, considerando-se que a distância do robô é de 115 cm, calcula-se o grau de pertinência, por meio das funções, para cada um dos conjuntos fuzzy distancia pequena, mediana e grande.

$$\mu_{\text{pequena}}(115) = (120 - 115) / (120 - 110) \quad (\text{opção 1 da função})$$

$$\mu_{\text{pequena}}(115) = 0,5$$

para o conjunto mediana

$$\mu_{\text{mediana}}(115) = (120 - 115) / (120 - 110) \quad (\text{opção 2 da função})$$

$$\mu_{\text{mediana}}(115) = 0,5$$

para o conjunto grande:

$$\mu_{\text{grande}}(115) = 0, \text{ pois } x \leq 115 \quad (\text{opção 3 da função})$$

6.3.4 Avaliação das Regras

Os graus de pertinência obtidos por meio da fuzificação são então transferidos para o mecanismo de inferência. O objetivo desta etapa é determinar o grau de pertinência de cada regra.

Portanto como existe apenas um antecedente para cada regra, o grau de pertinência é o próprio grau de pertinência obtido na etapa anterior.

Dando sequência ao exemplo inicializado anteriormente, no caso do robô autônomo o resultado obtido da inferência para 115 cm é:

- d) regra 1 (SE distancia = pequena ENTÃO velocidade = baixa): grau de pertinência = 0,5;
- e) regra 2 (SE distancia = mediana ENTÃO velocidade = média): grau de pertinência = 0,5;
- f) regra 3 (SE distancia = grande ENTÃO velocidade = rápida): grau de pertinência = 0.

6.3.5 Defuzificação

Após a realização da inferência, procede-se a defuzificação, a fim de transformar um valor fuzzy de saída em um valor *crisp*. Na base de regras do robô autônomo não existem consequentes iguais, deste modo, o resultado de agrupamento é o próprio grau de pertinência. Portanto, o contorno formado pela união dos conjuntos pequena e mediana para a entrada de 115 cm pode-se observar na figura 7.

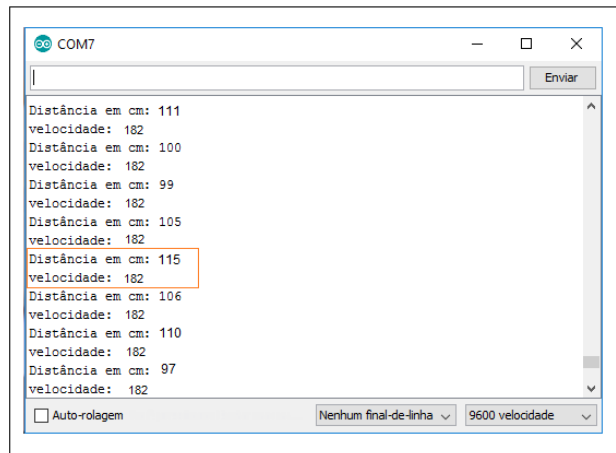


Figura 8 - Apresentação no monitor serial do exemplo inicializado na fuzificação

3. Análise dos Resultados Obtidos

A plataforma arduino utilizada no projeto mostrou-se ser capaz de realizar este modelo de desenvolvimento, pois o mesmo permite conexões com dispositivos externos assim tornando-o uma ferramenta para qualquer área de pesquisa envolvendo robótica móvel ou não.

Conforme a literatura de Kridi et al. (2014), a utilização de sensores infravermelhos apresentam valores inconsistentes para a função de captura de obstáculos, portanto os mesmos propuseram a utilização de sonares para trabalhos futuros. Neste projeto utilizou-se de um sensor ultrassônico para realizar as leituras de distância em relação aos obstáculos. Com base nos testes realizados durante o processo de desenvolvimento verificou-se que o mesmo apresentava inconsistência em suas leituras, para amenizar este problema optou-se por diminuir a quantidade de leituras realizadas para aplicar max-min de cada leitura, então observou-se que o tempo de resposta do sensor ultrassônico melhorou significativamente para a detecção de obstáculos, sendo assim capaz de executar sua tarefa sem dificuldade.

Na figura 4 observa-se os resultados obtidos referente as leituras realizadas na figura 9 após alterações no pseudocódigo. Deste modo, nota-se, que conforme o sensor realiza as leituras o serial monitor da própria plataforma arduino apresenta os valores na tela do mesmo, neste caso a distância foi medida em cm.

No mesmo teste de locomoção, pode-se observar que o modelo mostrou-se inconsistente em territórios desnivelados devido sua roda castor, porém em locais nivelados o mesmo apresentou consistência na locomoção.

Na figura 9 apresenta-se por meio do serial monitor os dados de captura e as direções tomadas durante o teste de locomoção

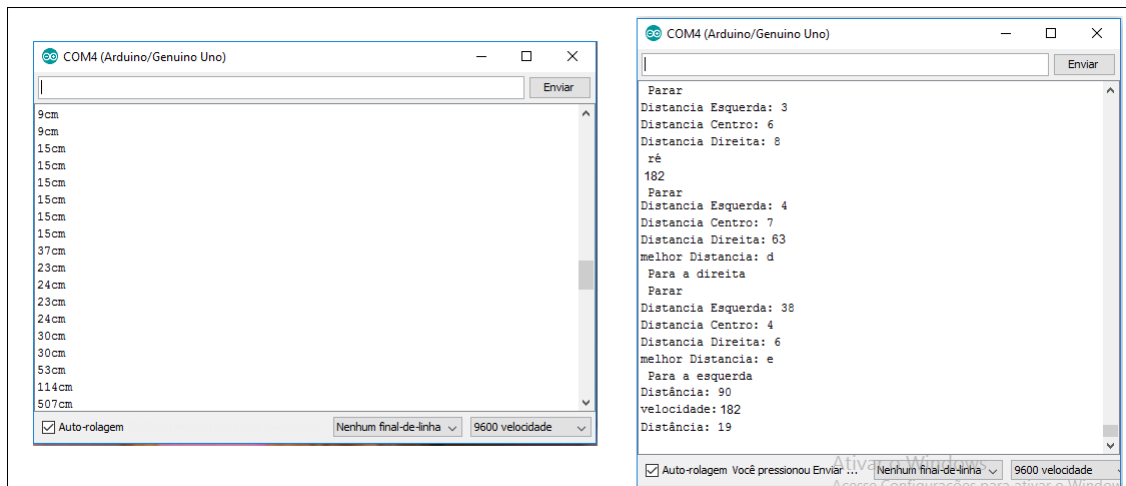


Figura 9 - Leitura do sensor verificada no monitor serial

4. Conclusão

Portanto este trabalho demonstrou como a lógica *fuzzy*, técnica de inteligência artificial, aplicada a plataforma arduino, pode contribuir para a área de robótica móvel que possuem informações vagas e imprecisas. O mesmo apresentou os conceitos de robótica móvel e de lógica *fuzzy*, resultando no desenvolvimento de um protótipo de robô móvel autônomo para o controle de velocidade em relação a distância do obstáculo encontrado no ambiente por meio da biblioteca eFLL.

Devido ao grande campo de atuação da teoria *fuzzy*, não foi a intenção de delinear o assunto e sim como ocorre seu funcionamento.

Durante o projeto, algumas dificuldades foram encontradas tanto na parte teórica quanto na prática. Em relação a teoria, ainda é pequeno o número de materiais que abordam sobre a biblioteca eFLL, apesar de possuir uma quantidade satisfatória de artigos científicos abordando lógica *fuzzy*, o que dificultou a compreensão do funcionamento interno da mesma. Na parte prática, foi determinar quais valores determinariam cada conjunto *fuzzy*, visto que as ações de controle de velocidade em relação ao obstáculo são quase sempre vagas e imprecisas. A plataforma arduino quanto o sensor ultrassônico permitiram que fosse possível desenvolver um protótipo aplicando a lógica *fuzzy*, sendo assim, ferramentas viáveis para este tipo de projeto.

Referências

ALMEIDA, T. S. et al. **Inteligência artificial aplicada à navegação autônoma de robôs**. In: SEMANA DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE SERGIPE, 2., 2012, Itabaiana. Anais eletrônicos... Itabaiana: DSI/UFS, 2012. Disponível em:

<www.si.itatechjr.com.br/images/docs/SEMINFO_UFSITA_2012.pdf>. Acesso em: 11 out. 2015.

GUDWIN, R.R. (2005). **Novas Fronteiras na Inteligência Artificial e na Robótica**. 4º Congresso Temático de Dinâmica, Controle e Aplicações Campinas:SP.:http://www.dca.fee.unicamp.br/%7Egudwin/ftp/publications/Dincom05_Gudwin.pdf>. Acesso em: 08 out. 2015.

KRIDI, Douglas Santiago et al. **DESENVOLVIMENTO DE UMA BIBLIOTECA FUZZY PARA O CONTROLE AUTÔNOMO DE UM ROBÔ MÓVEL EM AMBIENTE DESCONHECIDO**. Universidade Estadual do Piauí – UESPI / Laboratório de Informática Aplicada - LAIC. Disponível em: <<http://sistemaolimpo.org/midias/uploads/13fdd3bf2d22ab0aeecff212df10bd4c.pdf>> Acesso em: 17 mar. 2017.

LUGER, George F. **Inteligência artificial**. 6. ed São Paulo: Pearson Education do Brasil, 2013.

RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial**. Rio de Janeiro, Elsevier, 2013.

WOLF, Denis et al. Robótica Móvel Inteligente:: Da Simulação às Aplicações no Mundo Real. In: CONGRESSO DA SBC – SOCIEDADE BRASILEIRA DE COMPUTAÇÃO (BRASIL), 29., 2009, Bento Gonçalves. **Anais**. Bento Gonçalves: Sbc, 2009. p. 279 - 330. Disponível em: <http://osorio.wait4.org/palestras/texto/JAI2009_Completo_Revisado.pdf>. Acesso em: 12 mar. 2017.