



**FACULDADE DE TECNOLOGIA SENAI CIMATEC  
PROGRAMA DE POS-GRADUAÇÃO STRICTO SENSO  
GESTÃO E TECNOLOGIA INDUSTRIAL  
MODELAGEM COMPUTACIONAL E TECNOLOGIA INDUSTRIAL**

**JOSERLAND SOUZA SANTOS**

**UM SISTEMA TOLERANTE A FALHAS DE SENSORES EM  
ROBÔS AUTÔNOMOS NO CUMPRIMENTO DA MISSÃO**

Salvador  
2017

JOSERLAND SOUZA SANTOS

**UM SISTEMA TOLERANTE A FALHAS DE SENSORES EM  
ROBÔS AUTÔNOMOS NO CUMPRIMENTO DA MISSÃO**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação Stricto Sensu da Faculdade de Tecnologia SENAI CIMATEC como requisito parcial para a obtenção do título de Mestre em Modelagem Computacional e Tecnologia Industrial.

Orientador: Prof.Dr. Roberto Luiz Souza Monteiro

Co-Orientação: Prof. Dr. HermanAugustoLepikson

Salvador  
2017

Ficha catalográfica elaborada pela Biblioteca da Faculdade de Tecnologia SENAI CIMATEC

S237s Santos, Joserland Souza

Um sistema tolerante a falhas de sensores em robôs autônomos no cumprimento da missão / Joserland Souza Santos. – Salvador, 2017.

86 f. : il. color.

Orientador: Prof. Dr. Roberto Luiz Souza Monteiro.

Coorientador: Prof. Dr. Herman Augusto Lepikson.

Dissertação (Mestrado em Modelagem Computacional e Tecnologia Industrial) – Programa de Pós-Graduação, Faculdade de Tecnologia SENAI CIMATEC, Salvador, 2017.

Inclui referências.

1. Tolerância a falhas. 2. Robô autônomo. 3. Sensores - Robô. 4. Lógica fuzzy. I. Faculdade de Tecnologia SENAI CIMATEC. II. Monteiro, Roberto Luiz Souza. III. Lepikson, Herman Augusto. IV. Título.

CDD: 629.892

JOSERLAND SOUZA SANTOS

## **UM SISTEMA TOLERANTE A FALHAS DE SENSORES EM ROBÔS AUTONOMOS NO CUMPRIMENTO DA MISSÃO**

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre em Modelagem Computacional e Tecnologia Industrial, Faculdade de Tecnologia SENAI CIMATEC.

Aprovada em 28 de março de 2017.

Roberto Luiz Souza Monteiro – Orientador \_\_\_\_\_  
Doutor em Difusão do Conhecimento pela Universidade Federal da Bahia  
Faculdade de Tecnologia SENAI CIMATEC

Herman Augusto Lepikson – Co-Orientador - \_\_\_\_\_  
Doutor em Engenharia Mecânica pela Universidade Federal de Santa Catarina  
Faculdade de Tecnologia SENAI CIMATEC

Hugo Saba - \_\_\_\_\_  
Doutor em Difusão do Conhecimento pela Universidade Federal da Bahia  
Universidade do Estado da Bahia

Valéria L. da Silva - \_\_\_\_\_  
Doutora em Física pela Universidade Estadual de Campinas; Pós-Doutora, Bellcore-EUA  
Faculdade de Tecnologia SENAI CIMATEC

Valter de Senna - \_\_\_\_\_  
Pós-Doutor em Probabilidade e Estatística pela University of Southampton  
Faculdade de Tecnologia SENAI CIMATEC

Dedico essa dissertação a Paula, minha esposa, por toda a ajuda e paciência nesta fase.

## AGRADECIMENTOS

Agradeço aos professores Roberto Luiz Souza Monteiro e Herman Augusto Lepikson pela parceria e orientações ao longo do processo.

Agradeço aos professores Valter de Senna, Hugo Saba, Valeria L. da Silva pelas preciosas contribuições na banca de qualificação.

Agradeço a toda a equipe do Programa de Pós Graduação em Modelagem Computacional e Tecnológica Industrial.

## RESUMO

Ao longo dos anos, tem-se investido cada vez mais em sistemas inteligentes, tais como veículos autônomos, robôs de companhia, robôs médicos, entre outros. Nestes sistemas, as consequências de falhas ou faltas de um componente podem ser catastróficas. Com isso, considerar a possibilidade de desempenho errôneo por causa de falhas em alguns dos componentes é um requisito de fundamental importância desses sistemas. Uma falha consiste no funcionamento inadequado de um componente (sensor, atuador etc.) e é sintoma de um possível defeito do sistema. Quando o sistema tem a capacidade de lidar com situações de falhas em alguns de seus componentes de forma autônoma e mantendo sua estabilidade é denominado de sistema tolerante a falhas. Tendo isso em vista, o objetivo geral deste trabalho foi desenvolver um controle tolerante a falhas com lógica *Fuzzy* para sensores de distância em um robô autônomo no cumprimento de sua missão. Foi desenvolvida uma plataforma de robô capaz de se locomover em uma pista sem esbarrar em obstáculos, para isso o robô foi equipado com sensores de distância posicionados estrategicamente para que o mesmo identificasse os obstáculos e desviasse antes de colidir. Foi utilizado o método FMEA para identificar as possíveis falhas e o que deveria ser feito caso as falhas ocorressem. Esses dados foram comparados entre o desempenho do robô trabalhando com todos os sensores funcionando corretamente e com a falha de um sensor por vez ou dois sensores simultâneos. Notou-se a partir disso comportamentos bem diferentes a depender do sensor que entre em falha. Enfim, concluiu-se que o sistema tolerante a falhas desenvolvido oferece um bom desempenho para a simulação de alguns sensores simulados e em outros casos o desempenho não é tão satisfatório.

Palavras-chave: Tolerância a falhas. Robô autônomo. Sensores. Lógica *Fuzzy*.

## **ABSTRACT**

Throughout the years investments in intelligent systems have been increasing, such as autonomy vehicles, company robots, medical robots and among others. In these systems, some consequences of failures or a lack of components can be catastrophic. Therefore, considering a possibility of malfunction due to failures in some components of these systems is a fundamentally important requisite. A failure consists of inadequate operation in a component in this system (sensor, actuator and so on), and it is a symptom of a possible defect of the system. When a system has the capability to cope, autonomously, with failure situations in some of its components and it still maintains its stability this kind of system is denominated system fault-tolerant. Having this conception in view, the general goal of this work was develop a control fault-tolerant with fuzzy logic for distance sensors in a autonomy robot in the fulfilment of its mission. A platform of robot that can go through a lane without colliding with obstacles was developed, in order to do this the robot was equipped with distance sensors placed strategically in order to identify the obstacles and so that it can divert from obstacles before touch them. The method FMEA was used to identify possible failures and what should be done in case of failure. These data were compared with the performance of two different robots. The first one worked with all sensor running correctly and second with a failure in one sensor at a time or in two of them simultaneously. Thus was possible to check different behaviours depending on the sensor that was disabled or in failure. Ultimately, it is noticeable that the developed system fault-tolerant offers a satisfactory performance to some of the simulated sensors, whereas the performance in some other cases was not really good enough.

Keywords:Faulttolerance.Autonomousrobot.Sensors.Fuzzy Logic.



## LISTA DE FIGURAS

<b>Figura 2.1</b>	Princípios de tolerância a falha em sistemas robóticos baseado nos conceitos de Crestani, Godary-DejeaneLapierre.....	23
<b>Figura 3.1</b>	Dimensões da base do robô.....	27
<b>Figura 3.2</b>	Robô utilizado para o projeto.....	28
<b>Figura 3.3</b>	Placa controladora Arduino.....	29
<b>Figura 3.4</b>	Ponte H L298.....	30
<b>Figura 3.5</b>	Guia de posicionamento do sensor.....	31
<b>Figura 3.7</b>	Chassi com posicionamento dos sensores.....	32
<b>Figura 3.8</b>	Integração entre os módulos.....	33
<b>Figura 3.9</b>	Desenho da pista de testes.....	34
<b>Figura 4.1</b>	Estrutura geral do sistema tolerante a falhas.....	42
<b>Figura 4.2</b>	Esquema do sistema tolerante a falhas.....	42
<b>Figura 4.3</b>	Representação gráfica dos valores linguísticos das variáveis de entrada.....	43
<b>Figura 4.4</b>	Representação gráfica dos valores linguísticos das variáveis de saída.....	44

## LISTA DE QUADROS

<b>Quadro 3.1</b>	Escala de severidade do efeito da falha.....	35
<b>Quadro 3.2</b>	Etapas do processo para falha de 1 sensor.....	36
<b>Quadro 3.3</b>	Etapas do processo para falha de 2 sensores simultaneamente..	37
<b>Quadro 5.1</b>	Parâmetros de operação do robô.....	49
<b>Quadro 5.2</b>	Resultados dos testes com todos os sensores funcionando.....	50
<b>Quadro 5.3</b>	Simulação do sensor SDL utilizando SEL.....	51
<b>Quadro 5.4</b>	Simulação do sensor SEL utilizando SDL.....	51
<b>Quadro 5.5</b>	Simulação do sensor SFutilizando SEF-SDF.....	52
<b>Quadro 5.6</b>	Simulação do sensor SEF utilizando SF-SEL-SDF (Caso A) .....	53
<b>Quadro 5.7</b>	Simulação do sensor SEFutilizando SF-SDL-SDF (Caso B) .....	54
<b>Quadro 5.8</b>	Simulação do sensor SDF utilizando SF-SDL-SEF (Caso A) .....	54
<b>Quadro 5.9</b>	Simulação do sensor SDF utilizando SF-SEL-SEF (Caso B).....	54
<b>Quadro 5.10</b>	Simulação dos sensores SF e SDL utilizando SDF-SEF-SEL.....	55
<b>Quadro 5.11</b>	Simulação dos sensores SF e SEL utilizando SDF-SEF-SDL.....	56
<b>Quadro 5.12</b>	Simulação dos sensores SEF e SDL utilizando SDF-SF-SEL.....	57
<b>Quadro 5.13</b>	Simulação dos sensores SEF e SEL utilizando SDF-SF-SDL.....	57
<b>Quadro 5.14</b>	Simulação dos sensores SDF e SEL utilizando SEF-SF-SDL.....	58
<b>Quadro 5.15</b>	Simulação dos sensores SDF e SDL utilizando SEF-SF-SEL.....	58

## LISTA DE SIGLAS

CD	Curva para a direita
CE	Curva para a esquerda
CM	Centímetro
FKF	FederatedKalmanFilter
FMEA	Análise do Modo de Efeito de Falha
FSDF	<i>FuzzySensorDireitaFrente</i>
FSDL	<i>Fuzzy Sensor DireitaLado</i>
FSEF	<i>Fuzzy Sensor Esquerda Frente</i>
FSEL	<i>Fuzzy Sensor Esquerda Lado</i>
FSF	<i>Fuzzy Sensor Frontal</i>
kB	Kilobyte
mA	Miliampère
NPR	Número de Prioridade de Risco
SDF	Sensor Direita Frente
SDL	Sensor Direita Lado
SEF	Sensor Esquerda Frente
SEL	Sensor Esquerda Lado
SF	Sensor Frontal
SO	Sistema Operacional
V	Volt

# SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>17</b>
2.1 SISTEMAS TOLERANTES A FALHAS E ROBÓTICA AUTÔNOMA .....	17
2.2 TOLERÂNCIA A FALHAS .....	20
<b>2.2.1 Princípios de tolerância a falha em sistemas robóticos</b> .....	<b>21</b>
2.3 LÓGICA <i>FUZZY</i> .....	23
2.4 SENSORES .....	25
<b>3 O MÉTODO</b> .....	<b>27</b>
3.1 PLATAFORMA MECÂNICA .....	27
3.2 PLATAFORMA ELETRÔNICA .....	29
3.3 PLATAFORMA SENSORES .....	30
3.4 INTEGRAÇÃO DAS PLATAFORMAS .....	33
3.5 CENÁRIO PARA VALIDAÇÃO .....	33
3.6 PROBLEMA DE TOLERÂNCIA A FALHAS .....	34
<b>4 DESENVOLVIMENTO DO ALGORITMO</b> .....	<b>41</b>
4.1 ESTRUTURA DO SISTEMA TOLERANTE A FALHAS .....	44
4.2 REGRAS NEBULOSAS .....	46
<b>4.2.2 - Falha no Sensor Frontal – SF</b> .....	<b>47</b>
<b>4.2.3 - Falha no Sensor Esquerda Frente – SEF ou Sensor Direita Frente – SDF</b> .....	<b>47</b>
<b>4.2.4 - Falha no Sensor Direita Lado – SDL e Sensor Esquerda Lado – SEL</b> .....	<b>48</b>
<b>5 IMPLEMENTAÇÃO E ANÁLISE DOS RESULTADOS</b> .....	<b>50</b>
5.1 PARÂMETROS DE MEDIÇÃO .....	50
5.2 RESULTADOS DOS TESTES COM TODOS OS SENSORES FUNCIONANDO CORRETAMENTE .....	52
5.3 RESULTADOS DOS TESTES COM FALHA DE 1 SENSOR .....	52
<b>5.3.1 Resultados em caso de falha no Sensor Direita Lado – SDL e Sensor Esquerda Lado - SEL</b> .....	<b>53</b>
<b>5.3.2 Resultados em caso de falha no Sensor Frente – SF</b> .....	<b>54</b>
<b>5.3.3 Resultados em caso de falhado Sensor Direita Frente - SDF e Sensor Esquerda Frente - SEF</b> .....	<b>55</b>
5.4 RESULTADOS DOS TESTES COM FALHA DE DOIS SENSORES SIMULTÂNEAMENTE .....	57
<b>5.4.1 Resultados em caso de falha das combinações SF-SDL ou SF-SEL</b> .....	<b>57</b>
<b>5.4.2 - Resultados em caso de falha das combinações SEF-SDL, SEF-SEL, SDF-SEL, SDF-SDL</b> .....	<b>59</b>
<b>6 LIMITAÇÕES DA PLATAFORMA</b> .....	<b>62</b>
<b>7 CONSIDERAÇÕES FINAIS</b> .....	<b>65</b>
<b>REFERÊNCIAS</b> .....	<b>68</b>

<b>ANEXO A – <i>Datasheet</i> da Ponte H L298 .....</b>	<b>71</b>
<b>ANEXO B – <i>Datasheet</i> do sensor Sharp GP2Y0A21YK.....</b>	<b>85</b>

# 1 INTRODUÇÃO

Com o advento do desenvolvimento de plataformas de hardware e software dotadas de maior poder de processamento, menor custo, tamanho e consumo de energia, os sistemas embarcados tornaram-se uma excelente alternativa para o desenvolvimento de robôs autônomos com a finalidade de realizar tarefas “insalubres” para o ser humano.

Atualmente, tem-se que os robôs autônomos podem funcionar de duas formas: de forma supervisionada, na qual as informações de seu funcionamento são enviadas para um operador humano que monitora todo o funcionamento do robô, ou de forma totalmente autônoma, em que o robô toma suas próprias decisões com base na leitura e interpretação dos sinais de seus sensores sem a supervisão do homem.

Para que um robô autônomo seja eficiente em suas missões, deve ser capaz de lidar com situações adversas em seu ambiente de atuação que podem causar danos à sua estrutura e/ou a seus sensores, por isso os sistemas tolerantes a falhas são muito utilizados nestes casos, pois a ocorrência de uma falha durante o seu funcionamento pode causar danos irreversíveis, além de consumir muito tempo e dinheiro para reparo (CARDOZO, 2013).

Para resolver esta necessidade de tornar o robô tolerante a falhas de sensores de forma eficaz, este trabalho inspira-se nas habilidades humanas, ou seja, quando um indivíduo perde algum sentido (olfato, visão, audição) seu cérebro começa a explorar mais as informações provenientes dos sentidos que não foram afetados com o objetivo de tentar compensar a perda. Por exemplo, caso uma pessoa perca a visão, o cérebro aprende a realocar a parte não utilizada de processamento de imagens por causa da deficiência para outras funções necessárias como memória, linguagem e habilidade musical. Deste modo, uma pessoa cega consegue sentir o mundo ao seu redor de uma forma diferente, porém funcional, para a execução da maioria das suas tarefas cotidianas (Amedi et al., 2005). Sendo assim, por que não trazer essa habilidade para os robôs autônomos, já que os mesmos precisam lidar com possíveis falhas de seus sensores e continuar suas atividades com desempenho satisfatório?!

Vários trabalhos presentes na literatura discutem aspectos de como lidar com detecção, isolamento e tolerância a falhas em sistemas críticos/dinâmicos. Fernandes (2007) utilizou redes neurais para a detecção e isolamento de falhas em sistemas dinâmicos; Bueno (2006) também utilizou redes neurais para detectar falha de sensores em reatores nucleares; Zhu et al (2015) abordam uma estratégia inteligente tolerante a falhas de sistemas de navegação de veículos autônomos subaquáticos, entre outros. Apresenta-se, neste trabalho, a aplicação inteligência computacional baseada na Lógica *Fuzzy* que foi desenvolvida por Zadeh(1965) com o objetivo de reconstruir a informação do sensor em falha através da análise dos sinais dos demais sensores criando um sistema inspirado na reação humana perante a falta de algum sentido.

Segundo Cardozo (2013), os robôs autônomos têm despertado grande interesse em vários setores, a exemplo do industrial, para a manutenção e inspeção de equipamentos de difícil acesso (plataformas de exploração e dutos de distribuição de gás e petróleo); militar, para fins bélicos; científico, para realização de pesquisas, entre outros. Este trabalho visa apresentar uma forma de aumentar a robustez dos robôs autônomos contra falhas ou defeitos dos seus sensores utilizando técnicas de inteligência artificial baseada na observação de características humanas com um baixo custo, pois utiliza o hardware já existente nos robôs autônomos.

Segundo Weber (2003), as falhas estão associadas ao universo físico e, caso não sejam tratadas, podem levar a um erro que pertence ao universo da informação que, por sua vez, pode ocasionar um defeito, que é o universo do usuário. Não necessariamente uma falha se transforma em um erro e um erro se transforma em um defeito; isso depende das condições às quais o sistema é submetido e do seu projeto. Por exemplo, caso um microprocessador tenha uma falha de projeto em uma região específica do seu núcleo, a mesma não necessariamente irá evoluir para um erro, pois aquela região pode nunca ser usada e, por consequência, o sistema nunca apresentará aquela falha, ou por outro lado, caso esta região seja acionada oportunisticamente por algoritmo interno para compensar falha em outra região o sistema pode entrar em colapso. Falhas são coisas inevitáveis, pois os equipamentos envelhecem e sofrem interferência do ambiente externo, por isso os sistemas autônomos têm que, de forma muito

eficiente, detectar, isolar e tolerar as falhas provenientes do funcionamento do robô autônomo, principalmente em seus sensores que são responsáveis por informar as condições do ambiente à sua volta e possibilitar que as missões sejam efetuadas com sucesso. Sem a informação confiável dos sensores é basicamente impossível a execução das missões de forma satisfatória.

Durante a execução de suas missões um veículo autônomo está sujeito a um ambiente de trabalho imprevisível e agressivo, sendo assim, existe um elevado risco para que o mesmo apresente falhas - inclusive em seus sensores. Para que essas falhas não comprometam o funcionamento do robô autônomo, o mesmo tem que ser tolerante a falha ou defeito de alguns dos seus componentes, sensores inclusos.

Segundo Weber (2003), as técnicas de tolerância a falhas estão basicamente em duas classes: (i) mascaramento, que se baseia na redundância (hardware, software, informação ou tempo) fazendo com que as falhas não se manifestem como erros já que são mascaradas na origem, ou (ii) identificação da falha e reconfiguração do sistema para que o mesmo se adapte à nova realidade. A junção destas técnicas será apresentada neste trabalho e corresponde à detecção, localização e mascaramento de falhas através da reconfiguração do sinal do sensor para o sistema operacional (SO) de um robô autônomo, utilizando o mesmo hardware. Obtem-se assim um sistema tolerante a falhas que não aumente o custo com equipamentos e nem o peso do robô - o que é uma variável muito importante para o seu desempenho.

Tendo em vista o cenário descrito neste trabalho, foi desenvolvido um sistema tolerante a falhas baseado em Lógica *Fuzzy* que tem a capacidade de efetuar a leitura dos sensores e, ao mesmo tempo, caso um sensor esteja em falha, emular o sinal do sensor defeituoso com base nas informações dos demais sensores. Sendo assim, caso exista uma diferença entre o sinal lido e o sinal emulado, é detectada a presença de uma falha. Após essa etapa, a falha é localizada e mascarada para o SO, pois a informação proveniente do sensor é descartada e, em seu lugar, é utilizada a informação emulada proveniente dos sinais dos demais sensores.

O objetivo geral deste trabalho foi *desenvolver um sistema tolerante a falhas utilizando lógica Fuzzy para otimizar o desempenho de sensores em robôs autônomos*. E os objetivos específicos foram: (i) desenvolver um sistema de



monitoramento que reaja automaticamente na ocorrência de falhas de sensor de proximidade; (ii) elaborar um modelo para teste do algoritmo proposto.

A relevância do trabalho se mostra ao buscar desenvolver um modelo de como o robô autônomo deve proceder caso haja alguma falha nos seus sensores de modo que, mesmo com falha, o robô consiga completar sua missão com eficiência.

A dissertação possui 6 capítulos, além da presente introdução.

O capítulo 2 – *Fundamentação teórica* traz o referencial teórico com os trabalhos que foram relevantes para o problema aqui tratado.

O capítulo 3 – *O método* aborda o método utilizado para o desenvolvimento da proposta; apresenta as plataformas usadas para construção do robô, o cenário para validação e o problema de tolerância a falhas.

Já o capítulo 4 – *Desenvolvimento do algoritmo* expõe a modelagem do sistema proposto, ou seja, a aplicação do modelo Fuzzy, assim como as regras lógicas necessárias para o funcionamento do sistema.

No capítulo 5 – *Implementação e análise dos resultados* são apresentados os testes realizados com o robô e são discutidos os resultados obtidos.

O capítulo 6 – *Limitações da plataforma* expõe as limitações e imprecisões do modelo que em algumas situações interferiram no deslocamento do robô. Por fim, o capítulo 6 – *Considerações finais*.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados alguns trabalhos que aplicaram as técnicas de sistemas tolerantes a falhas em robôs autônomos e que serviram de base e inspiração para esta pesquisa. Apresenta-se também uma breve introdução sobre lógica *Fuzzy*, além de conceitos importantes sobre tolerância a falhas e a classificação, segundo a literatura, dos conceitos de falha, defeito e erro.

### 2.1 SISTEMAS TOLERANTES A FALHAS E ROBÓTICA AUTÔNOMA

No artigo de Zhu *et al.* (2015) é abordada uma estratégia inteligente tolerante a falhas para sistemas de navegação de veículos autônomos subaquáticos com o objetivo de garantir segurança e confiabilidade. Nessa estratégia é usado o *federatedKalmanfilter (FKF)*, que é concebido para fundir vários subsistemas, tais como sistema de navegação inercial, bússola magnética, velocidade e topologia do terreno de navegação. Segundo os autores, o problema na utilização desse método se dá devido aos níveis de ruído que interferem no sistema, por isso foi utilizada a lógica *Fuzzy* para criar um controle adaptativo que se ajusta automaticamente visando neutralizar o efeito do ruído.

O trabalho apresentado na presente dissertação baseia-se em um robô terrestre que também utiliza sensores para navegação, porém, devido à natureza dos sinais dos sensores utilizados aqui serem mais simples do que os utilizados por Zhu *et al.* (2015), optou-se por utilizar a lógica *Fuzzy*, não para diminuir o efeito do ruído no sistema, mas para o motor principal do sistema ser tolerante a falhas. Com isso, espera-se ter um sistema tolerante a falhas eficiente e que tenha uma menor complexidade matemática e computacional do que o proposto por Zhu *et al.* (2015) - já que não foi utilizado aqui o filtro FKF (*FederatedKalmanFilter*) -; desse modo, podemos embarcá-lo em microcontroladores com baixa capacidade de processamento e, conseqüentemente, com um menor consumo de energia.

Malar (2016) apresenta o desenvolvimento de um sistema de detecção de falhas nas juntas de um robô manipulador industrial através da análise de ruído e de vibrações da articulação do robô. O autor utilizou uma rede neural artificial capaz de se adaptar às mudanças de carga e outros parâmetros de funcionamento. Foi usado

um sistema híbrido no qual as redes neurais são utilizadas para aprendizado do modo de falha e a lógica *Fuzzy* trabalha com a imprecisão dos dados; desse modo, utilizam-se as características da falha apresentadas pela rede para auxiliar o processo de detecção e isolamento.

A partir desse método, Malar (2016) efetua o diagnóstico das falhas que acontecem no braço robótico. Já aqui, nesta dissertação, se fez uso de uma forma diferente para diagnóstico de falhas: o sistema proposto efetua a comparação entre o sinal que está sendo simulado pelo sistema e o sinal real do sensor e, caso haja divergência, o sinal do sensor é ignorado e o sinal simulado é levado em consideração. Com isso temos um modelo bem mais simples e que consegue identificar a falha e tornar o sistema tolerante, porém, sem efetuar o diagnóstico. Mas como o objetivo desta dissertação é possibilitar que, mesmo com falha, o robô execute sua missão de forma satisfatória, para simplificar o modelo optou-se em não efetuar o diagnóstico de falhas.

O artigo de Hafez e Kamel (2016) apresenta um controle tolerante a falhas para cooperação de veículos aéreos não tripulados autônomos no qual foi implementado um modelo linear descentralizado de controle preditivo tendo como principal contribuição a utilização da lógica *Fuzzy* para tomar a decisão de resolver o problema da reconfiguração da formação de vôo caso algum elemento apresente falha. Ou seja, cada indivíduo do grupo tem seu próprio controlador *Fuzzy* e, caso encontre uma falha, o mesmo sai da formação e informa aos demais que está fora da formação e qual deve ser a nova configuração de vôo.

A pesquisa apresentada nesta dissertação utiliza uma estratégia parecida com a proposta de Hafez e Kamel, (2016), pois também utiliza a lógica *Fuzzy* para detecção de falhas e correção de problemas provenientes dessas falhas de modo que outros elementos do sistema possam assumir o papel do elemento em falha. Os dois autores têm um modelo no qual vários robôs cooperam entre si para manter sua missão com eficiência, mesmo em caso de falha. Já nesta dissertação, como utilizamos somente 1 (um) robô, trabalhamos com a cooperação da informação de diferentes sensores para serem tolerantes a falhas; mas ambos os modelos utilizam um controlador *Fuzzy* para a tomada de decisão.

Crestani, Godary-Dejean e Lapierre (2015) apresentam uma abordagem que integra os princípios de tolerância a falhas no desenho de uma arquitetura de controle em tempo real de um robô móvel, analisando os aspectos construtivos de hardware e software para identificar assinaturas de comportamentos defeituosos, diagnosticar e recuperar falhas de modo eficiente na etapa de projeto com o objetivo de criar um método de desenvolvimento de projeto de robôs autônomos. Esses três autores desenvolveram um método para a concepção completa do robô e de seus possíveis modos de falhas. Para isso, utilizaram ferramentas clássicas para diagnosticar eventuais falhas de projeto, como a análise do modo do efeito de falha, o FMEA (DAĞSUYU, 2016). Como nesta dissertação o objetivo não está na montagem do robô e, sim, na construção de um sistema tolerante a falhas de sensores, utilizou-se apenas a etapa do FMEA que efetua uma análise do modo e efeito das falhas dos sensores, ou seja, com essa ferramenta podemos identificar cada sensor em falha e quais as possíveis decisões que o robô deve tomar.

O FMEA foi desenvolvido na década de 50 pelas forças armadas dos Estados Unidos para o processo de construção de aeronaves. Logo após, na década de 1960, a NASA integrou essa ferramenta ao seu programa espacial e a *Ford* foi a primeira indústria a utilizar essa ferramenta (CHIOZZA, PONZETTI, 2009).

O FMEA é um método proativo que evita falhas em sistemas e processos. Usa três fatores de risco para determinar a causa, o efeito e as ações que se deve executar no caso de ocorrência de falhas. Esses fatores são: a frequência com que a falha ocorre, a facilidade de sua detecção e a sua gravidade. Com isso gera-se uma saída resultante do produto dos três fatores para formar o número de prioridade de risco (NPR), que sinaliza quais as falhas mais críticas para o sistema e por isso devem ter sua correção priorizada. Nesta dissertação somente será considerado o fator de gravidade da falha, pois não se tem dados históricos para determinar a frequência de ocorrência da falha. Além disso, como o objetivo principal é que o sistema identifique a falha e trate a mesma para não comprometer o funcionamento do robô, não foi considerada a facilidade de sua detecção, ou seja, para o sistema não interessa se a falha pode ou não acontecer, interessa o modo de ação caso a falha ocorra. Interessa somente identificar quais ações devem ser tomadas para manter o funcionamento do sistema de uma forma aceitável caso a falha ocorra.

O tema também foi abordado na tese de Caldas (2004). A tese apresenta uma metodologia de desenvolvimento de controle para robôs móveis de arquitetura híbrida integrando tolerância a falhas adaptativa através de redundâncias que existam em um único robô ou distribuídas nas diversas competências individuais de outros agentes. Já esta dissertação pretende que o robô não precise de redundância em seus sensores, ou seja, o objetivo é que a informação de outros sensores possa ser usada para a simulação do valor do sensor que estaria com defeito, neste caso o robô não necessita de uma redundância já que o objetivo é reconstruir a informação faltante através dos sinais dos outros sensores.

Por fim, temos a dissertação de Cardozo (2013). Em seu trabalho, a autoradesenvolveu um controle tolerante a falhas que consegue reagir de forma autônoma à falha de propulsores em um veículo subaquático por meio de um controle que utiliza a lógica *Fuzzy* como a inteligência do controlador. Na presente pesquisa também é abordada a lógica *Fuzzy* como a inteligência do controlador, mas aqui ela foi usada para reagir às falhas dos sensores de medição de distância com o objetivo de possibilitar que o robô navegue no ambiente mesmo com falhas em alguns de seus sensores.

## 2.2 TOLERÂNCIA A FALHAS

Tolerância a falhas é uma área de estudo bastante abrangente e evoluída, porém na aplicação de suas técnicas ainda não existe uma solução única (ideal) que contemple todo tipo de projeto devido às especificidades de cada projeto (CALDAS, 2004).

Ainda de acordo com Caldas (2004), a tolerância a falhas é basicamente a capacidade que um sistema tem de, mesmo na presença de defeitos ou falhas, fornecer suas saídas de forma correta.

Crestani, Godary-Dejean e Lapierre (2015) abordam a importância da utilização de técnicas de tolerância a falhas em robôs móveis autônomos no cumprimento de sua missão, pois os mesmos podem sofrer avarias e geralmente não são projetados para lidar de forma eficiente com falhas ou defeitos. Sendo assim, percebe-se que o projeto de sistemas confiáveis é uma questão crítica que

pode salvar vidas humanas e reduzir impactos técnicos e financeiros devido a falhas catastróficas em projetos de alto custo.

Para trabalhar com tolerância a falhas é importante analisar os estados indesejados que podem vir a ocorrer nos sistemas de robôs autônomos, mas antes disso convém entender alguns conceitos básicos sobre falha, defeito e erro (CALDAS, 2004):

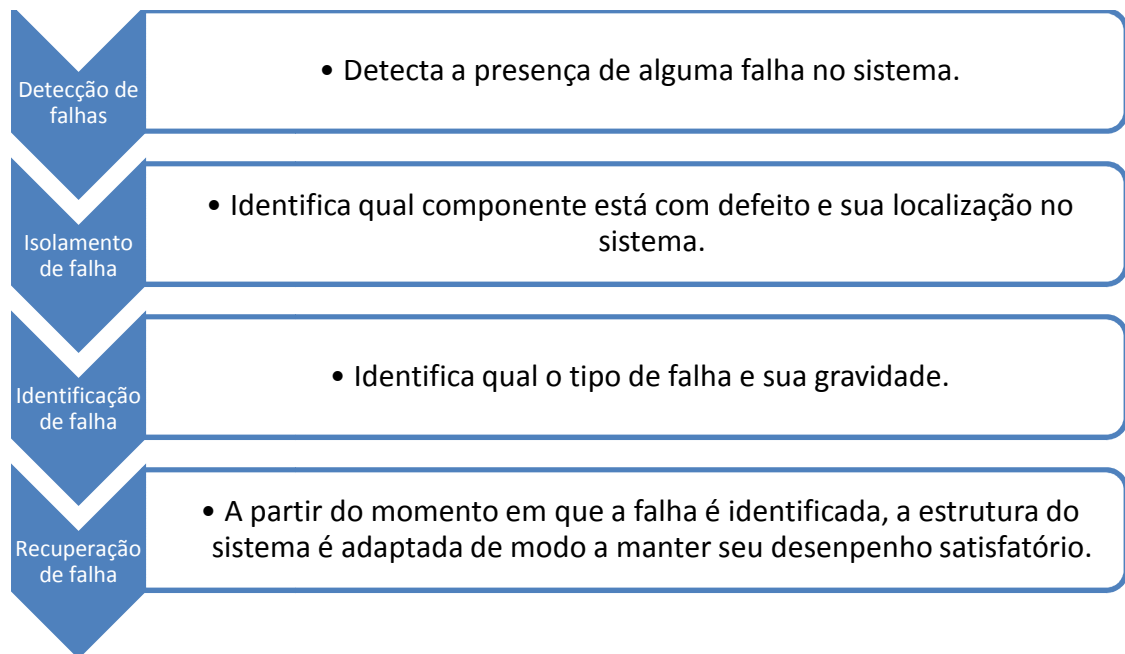
- a) Defeitos: são consequências de erros que são perceptíveis ao usuário; pode ser tanto de hardware (travamento da junta de um robô, por exemplo), ou de software quando provocam erros na execução de um programa.
- b) Falhas: são os sintomas de um defeito, pertencem ao universo físico e não necessariamente se manifestam. Por exemplo, um robô pode possuir uma falha de projeto em determinada parte de seu sistema, porém aquela parte pode nunca ser utilizada, o que fará com que a falha não ocorra.
- c) Erros: estão no universo da informação e, basicamente, são manifestações das falhas, ou seja, o erro aparece quando a falha ocorre e o sistema apresenta um comportamento indesejado.

Deste modo, uma falha não necessariamente leva a um erro, pois aquela parte do sistema pode nunca ser acessada e um erro não necessariamente leva a um defeito, pois o sistema pode utilizar de outras informações para que seu desempenho seja mantido de forma satisfatória.

### **2.2.1 Princípios de tolerância a falha em sistemas robóticos**

Conforme apresentado por Crestani, Godary-Dejean e Lapierre (2015), um sistema robótico tolerante a falhas é capaz de manter a sua estabilidade em um desempenho aceitável quando ocorrerem falhas em seus componentes e deve seguir quatro princípios básicos: detecção, isolamento, identificação e recuperação de falhas, conforme apresentado na Figura 2.1.

**Figura 2.1** - Princípios de tolerância a falha em sistemas robóticos.



Fonte: Crestani, Godary-Dejean e Lapierre (2015).

A partir do momento em que se deseja que um sistema mantenha o seu funcionamento normal na presença de falhas é necessário que o mesmo seja projetado apresentando caminhos alternativos para que as informações do processo se mantenham confiáveis. Para isso, é necessário que o sistema contemple as etapas apresentadas na Figura 2.1, em que no processo de detecção de falhas são comparados os valores obtidos de um módulo com um conjunto de valores esperados e, caso haja alguma divergência, é detectada a presença de uma falha. Em seguida, é necessário que o sistema identifique qual módulo está com problema e, com isso, possa isolá-lo do sistema. Na seqüência, a falha deve ser identificada juntamente com a sua gravidade perante o sistema. A correta identificação da falha é fundamental para que a etapa de recuperação seja eficiente. Esse processo de recuperação pode ser feito através de redundância de módulos físicos ou lógicos ou através da utilização de outras informações para substituir as informações do módulo com defeito (CALDAS, 2004).

## 2.3 LÓGICA FUZZY

A lógica *Fuzzy* foi desenvolvida por Zadeh (1965) com o objetivo de tratar os aspectos vagos da informação através de regras lingüísticas, ou seja, o processo de inferência *Fuzzy* é capaz de gerar valores de saída a partir de valores de entrada baseados em regras lingüísticas. Essas regras baseiam-se na opinião de especialistas humanos que descrevem seu conhecimento do processo através de regras lógicas, prexemplo: **Se** condição **Então** conclusão, ou seja, **Se**  $a = b$  **Então** retorne  $c$ .

Segundo Simões e Shaw (2007), o processo de extração de informação dos especialistas humanos e construção do modelo *Fuzzy* são compostos pelos seguintes blocos funcionais:

- Interface de fuzzificação
- Base de conhecimento
- Lógica de tomada de decisões
- Interface de defuzzificação

Essa estrutura representa o funcionamento do controlador *Fuzzy*, onde ocorre a transformação dos dados do mundo real para os dados *Fuzzy*, desta forma é utilizada a base de conhecimento para tomada de decisões lógicas e, por fim, os dados *Fuzzy* são traduzidos novamente para os dados do mundo real, para que haja a resposta do sistema *Fuzzy* aos estímulos de entrada.

As variáveis de entrada de um sistema *Fuzzy* geralmente são provenientes de sensores presentes em equipamentos nos quais se quer implementar o controlador *Fuzzy*. Esses sensores possuem uma faixa de operação que é utilizado como um universo de discurso pré-definido para cada variável de entrada. O procedimento de fuzzificação converte o valor da faixa de operação em uma interface entre  $[0,1]$  (MARTINS, 2014). A isso chamamos de interface de fuzzificação. A base de conhecimento é formada pela base de dados e a base de regras *Fuzzy*, que é composta a partir das informações extraídas do conhecimento do especialista do processo. De acordo com Simões e Shaw (2007), a base de dados fornece



definições numéricas usadas no conjunto numérico *Fuzzy*, já a base de regras serve como parâmetros lógicos da estratégia do controlador *Fuzzy*. Ainda de acordo com esses mesmos autores:

A lógica de tomada de decisões, que é incorporada na estrutura de inferência da base de regras, usa implicações *Fuzzy* para simular tomadas de decisões humanas. Ela gera ações de controle – consequentes – inferidas a partir de um conjunto de condições de entradas – antecedentes. [...] A defuzzificação consiste em obter-se um único valor, discreto, utilizável numa ação de controle concreta no mundo real, a partir de valores *Fuzzy* de saída obtidos. Este único valor discreto representa um compromisso entre os diferentes valores *Fuzzy* contidos na saída do controlador. (SIMÕES e SHAW 2007, p. 46/47).

Os principais métodos utilizados para a defuzzificação são: Centro-da-Área (C-o-A) que calcula o centróide dos resultados da saída *Fuzzy* conforme apresentado na equação 2.1.

$$u = \frac{\sum_{i=1}^N ui \cdot \sum_{k=1}^N \mu_{out}(ui)}{\sum_{i=1}^N \sum_{k=1}^N \mu_{out}(ui)} \quad \text{Equação 2.1}$$

Onde  $u$  é o resultado do centróide da área,  $\mu_{out}(ui)$  é a área de uma função de pertinência,  $ui$  é a posição do centróide da função de pertinência individual. Já o Centro-do-Máximo (C-o-M) utiliza a mesma técnica do (C-o-A) porém só considera seus máximos e é representada pela equação 2.2.

$$u = \frac{\sum_{j=1}^N ui \mu_{out}(ui)}{\sum_{j=1}^N \mu_{out}(ui)} \quad \text{Equação 2.2}$$

E por fim a Média do Máximo (M-o-M) que trabalha com a média dos máximos representada pela equação 2.3.

$$u = \sum_{m=1}^M \frac{um}{M} \quad \text{Equação 2.3}$$

Onde  $um$  é o  $m$ -ésimo elemento de discurso e  $M$  é o número total desse elemento.

## 2.4 SENSORES

Sensores são elementos com a capacidade de “sentir” determinadas características do ambiente ao seu redor e traduzi-las em dados processáveis. Existem diversos tipos de sensores, cada um com sua aplicação específica, com grandes diferenças na sua composição e estrutura física, mas com um único objetivo: ser a porta de entrada das informações do ambiente ao seu redor (CALDAS, 2004).

Pereira, Amorim e Castro (2003) dizem que sensor é o dispositivo que implementa a monitoração física de um fenômeno ambiental e que a análise de suas informações pode gerar relatórios de análise de desempenho dos equipamentos monitorados, ou seja, um sensor produz uma resposta mensurável a mudanças de condições físicas, tais como temperatura, pressão, vazão, luminosidade, entre outros.

Megerian, Koushanfar, Veltri e Potkonjak (2002), informam que um sensor consiste basicamente de: detector de hardware, memória, bateria, processador embutido e transmissor e, em sua maioria compartilham de duas características básicas: quanto maior a distância menor a precisão da medição e quanto menor for o nível de interferência e ruído menor será o seu efeito sobre o erro de medição.

No caso da robótica autônoma - foco deste trabalho -, alguns tipos de sensores são utilizados para a identificação de obstáculos ao redor. Quanto maior for o conjunto de sensores, mais preciso será o processo de identificação de obstáculos. Porém isso causa um impacto diretamente proporcional no custo, consumo de energia e no processamento computacional do robô.

Como já foi apresentado aqui, os sensores estão sujeitos a erros, incertezas e falhas, e objetivando uma melhor qualidade da informação da presença e distância de obstáculos, é comum se utilizar um conjunto de sensores redundantes. Com o objetivo de reduzir custo e consumo de energia do robô autônomo pode-se utilizar o conjunto de informações provenientes de vários sensores para que haja uma fusão de dados que permita uma otimização do desempenho mesmo na presença de

falhas, sem a necessidade de que sejam utilizadas redundâncias de hardware. Isso pode levar a uma redução de custo e maior robustez do robô.

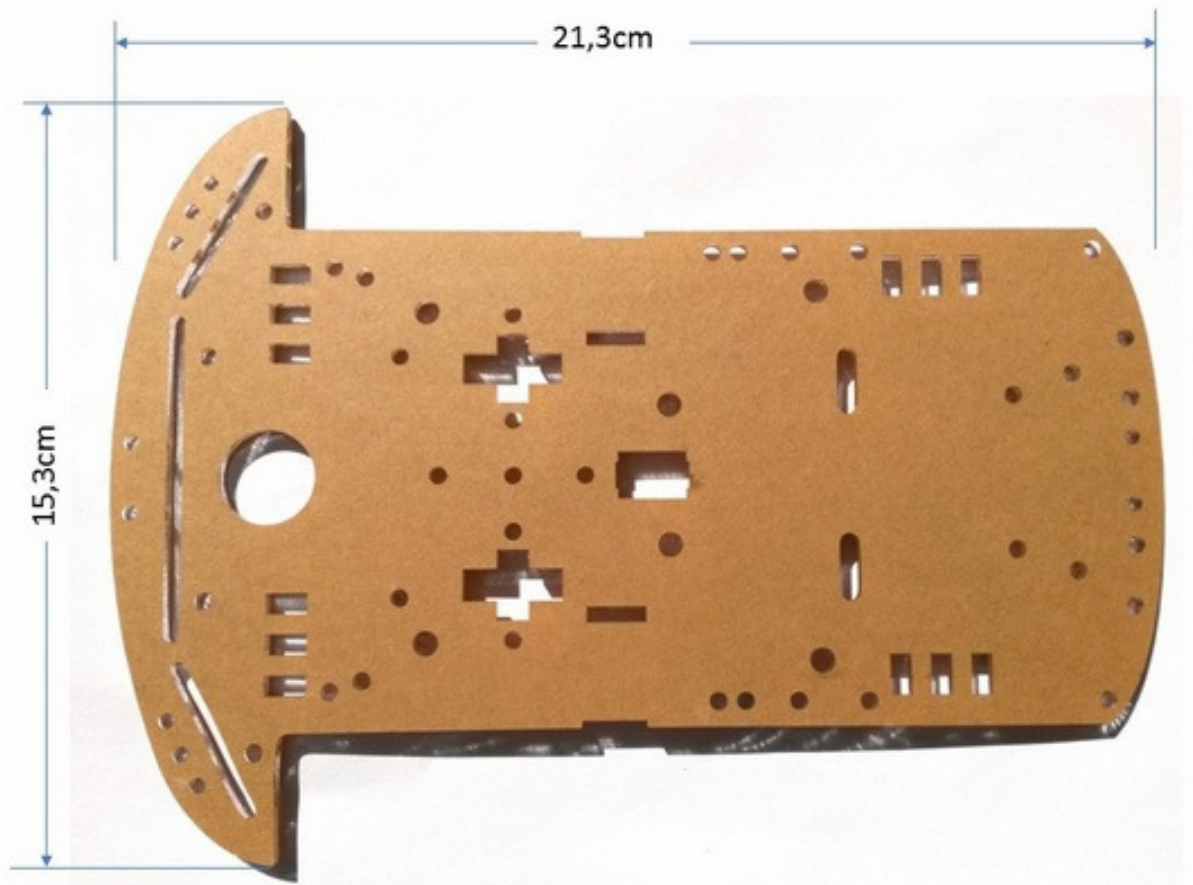
Há também os casos em que as informações de outros sensores presentes em outras plataformas externas podem ser integradas aos dados de processamento de sinais. Para isso, podem ser utilizadas redes de sensores sem fio que possibilitam a transmissão de informação entre as diferentes plataformas (CALDAS, 2004).

## 3 O MÉTODO

### 3.1 PLATAFORMA MECÂNICA

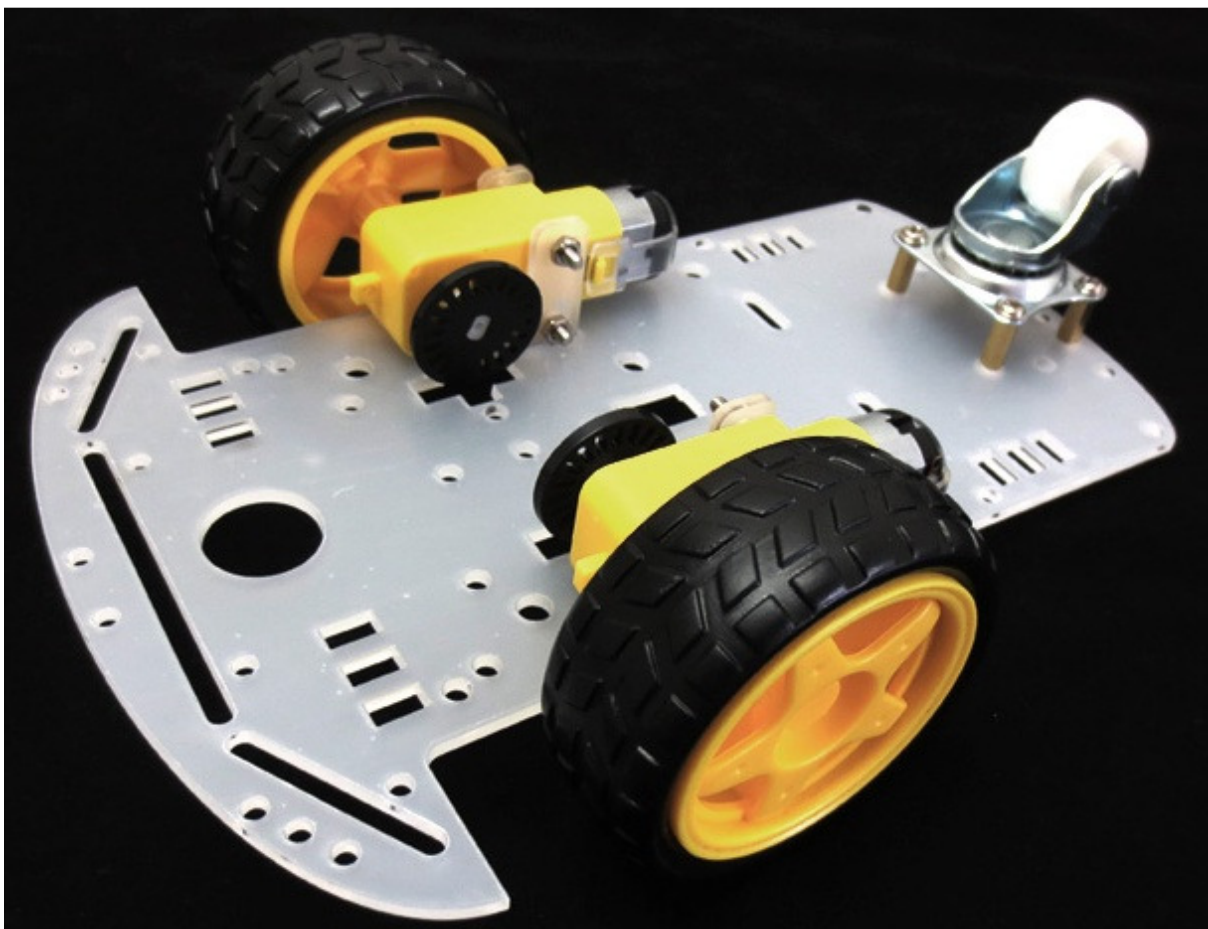
Conforme anunciado na introdução, o projeto tem como objetivo o desenvolvimento de um sistema tolerante a falhas de sensores em robôs autônomos. Para efetuar os testes do sistema proposto foi desenvolvida uma plataforma (Figuras 3.1 e 3.2) e uma pista que servem para validar o sistema dentro de uma determinada condição controlada.

**Figura 3.1**– Dimensões da base do robô



Fonte: <http://carrinhocontroladoporsensoroptico.blogspot.com.br/>

**Figura 3.2-** Robô utilizado para projeto



Fonte: <https://multilogica-shop.com/tutorial/montagem-passo-passo-do-kit-chassi-robotico-2wd>

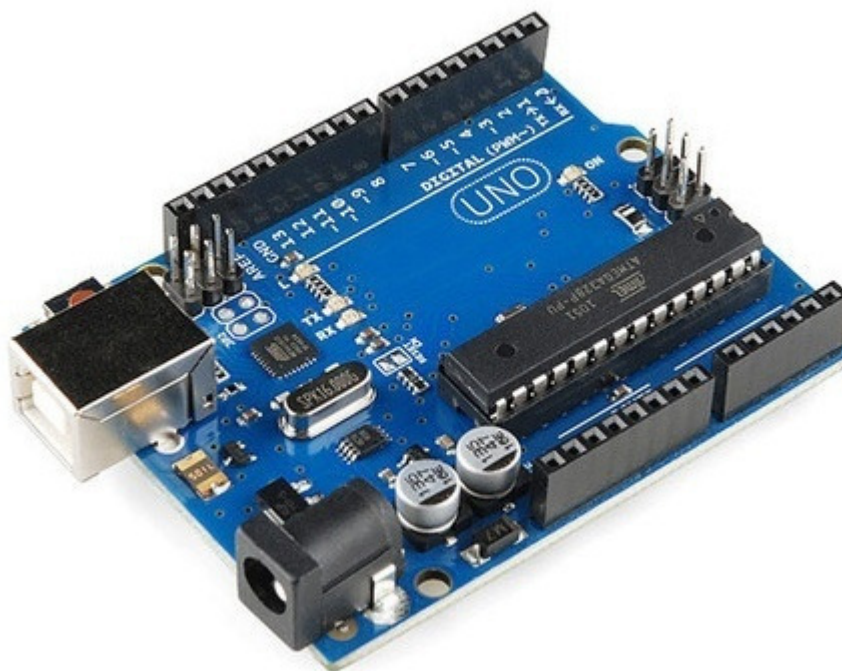
A plataforma apresentada é composta por dois motores independentes, cada um conectado a uma roda e uma caixa de redução com uma relação de 120:1, podendo chegar a uma velocidade de 200 rotações por minuto alimentado com 6 Volts ou com 90 rotações por minuto quando alimentado com 3V. O chassi também conta com três rodas: duas rodas emborrachadas conectadas aos motores com diâmetro de 7 cm e largura de 2,5cm e uma roda “livre” universal que pode ser adquirida em qualquer casa de material de construção.

Esta plataforma permite que o robô consiga girar em seu próprio eixo no momento em que acionamos a roda direita no sentido contrário da roda esquerda, ou ir para frente ou para trás acionando as rodas no mesmo sentido. Isso possibilita que o robô efetue um giro tanto para a direita como para esquerda, para frente ou para trás, tendo liberdade de seguir para qualquer direção.

### 3.2 PLATAFORMA ELETRÔNICA

Para efetuar o controle da movimentação do robô foi utilizada uma placa controladora da marca Arduino Uno (Figura 3.3), que possui um micro controlador Atmega328 de 16MHz de velocidade de processamento, com 6 pinos de entrada ou saída de sinais analógicos de 0 V a 5 V e 14 pinos de entrada ou saída de sinais digitais com tensão de operação de 5 V. A placa pode ser alimentada com uma tensão entre 7 V a 12 V, porém, devido a um regulador de tensão interno, trabalha com uma tensão operacional de 5 V. A placa também possui uma memória interna de 32kB para armazenamento de programas (ARDUINO, 2017). Sua programação pode ser feita em linguagem C dentro do seu próprio ambiente de programação. Esta plataforma foi escolhida por ser *open source* e de baixo custo, além de ter uma grande comunidade desenvolvedora de projetos.

**Figura 3.3** – Placa controladora Arduino UNO



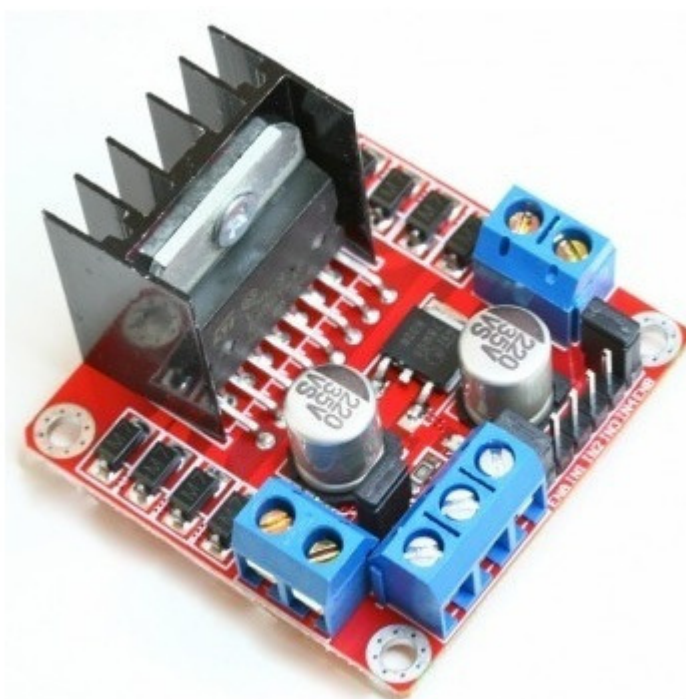
Fonte:(<http://www.filipeflop.com/pd-6b58d-placa-uno-r3-cabo-usb-para-arduino.html>)

A controladora Arduino UNO é responsável pela leitura e processamento dos sinais provenientes dos sensores dentro das regras do sistema *Fuzzy* desenvolvido nesta pesquisa e do sistema de controle principal do robô. Assim, após o processamento das informações, a placa Arduino Uno envia os sinais para o

acionamento dos motores de deslocamento. Esses motores, por sua vez, são controlados pela Ponte H L298(Figura 3.4), que pode controlar o sentido de giro de 2 motores de até 2A trabalhando com uma tensão que pode variar de 4 V até 35 V. (WEBTRONICO, 2017). O *datasheet* da Ponte H L298 encontra-se no Anexo A.

Como a placa controladora Arduino Uno trabalha com sinais de 5 V, conforme mencionado anteriormente, optou-se por padronizar os valores de tensão do robô com o objetivo de facilitar o envio de sinais entre os módulos e eliminar a necessidade de conversores de tensão, o que aumentaria o peso e o custo do robô.

**Figura 3.4** – Ponte H L298



Fonte:<http://www.webtronico.com/modulo-driver-motor-dc-motor-de-passo-l298n.html>

### 3.3 PLATAFORMA SENSORES

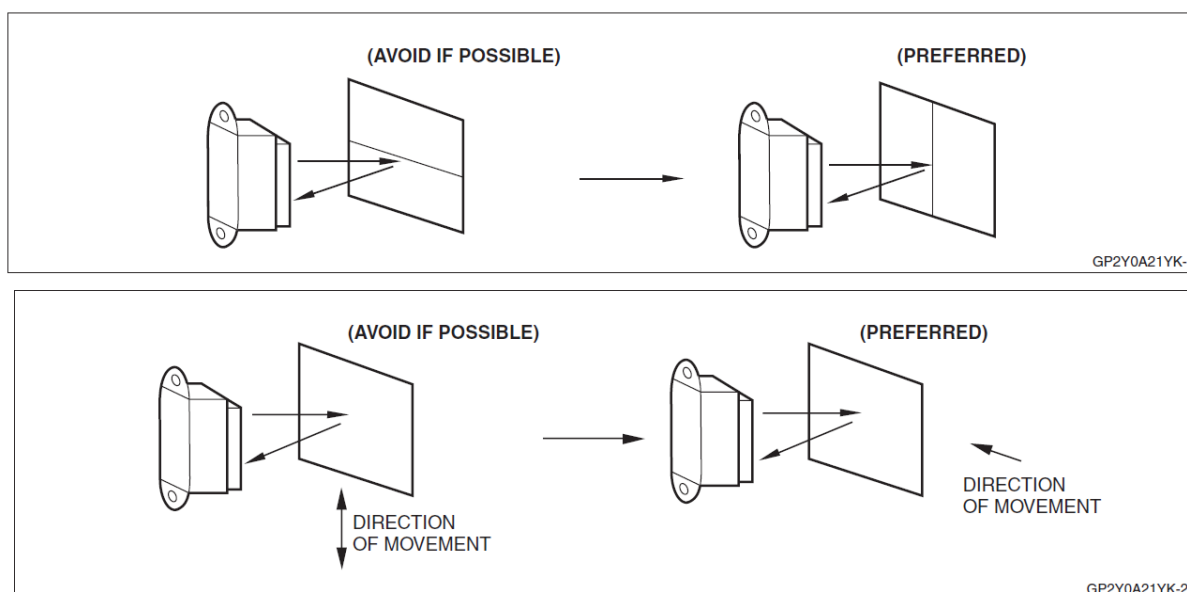
A plataforma do robô é composta por 5 sensores infravermelho modelo SHARP GP2Y0A21YK que efetua a medição de distância de 10 cm a 80 cm contados a partir da lente do sensor. Pode ser alimentado com uma tensão contínua de 4,5 V a 5,5V. Para efetuar a medição de distância, o sensor baseia-se na emissão de um feixe de luz infravermelha que atinge o objeto que se deseja medir e com base na intensidade de reflexão da luz o mesmo emite um sinal analógico que pode variar de



1,65V para 10 cm e 2,15V para a distância de 80 cm. Esses valores, segundo o fabricante (*datasheet* no AnexoB), são verdadeiros quando o sensor é alimentado com uma tensão de 5 V e trabalha a uma temperatura de 25°C.

Ainda de acordo com o fabricante, para o correto posicionamento do sensor se deve seguir as orientações apresentadas na Figura 3.5, onde o modo correto de posicionamento de leitura do sensor deve ser observado. Na primeira condição GP2Y0A21YK-1 deve-se, se possível, evitar o posicionamento do sensor na vertical quando o objeto a ser identificado tiver uma divisão na horizontal. No caso GP2Y0A21YK-2, se o sensor estiver na posição vertical é preferível que o objeto a ser medido se desloque na horizontal. No robô desenvolvido não temos nenhuma situação que reflita a condição apresentada na Figura 3.5 em GP2Y0A21YK-1, pois a superfície da parede da pista é lisa. Porém, na condição GP2Y0A21YK-2, devido às características físicas do suporte dos sensores não foi possível o posicionamento da forma que o fabricante prefere, ou seja, o sensor na posição vertical e o deslocamento do objeto medido na posição perpendicular ao sensor. Sendo assim, devido a condições físicas e por se tratar de uma recomendação e não de uma determinação do fabricante, optou-se por posicionar o sensor horizontalmente e o deslocamento da pista em relação ao robô segue o mesmo sentido.

**Figura 3.5** – Guia de posicionamento do sensor



Fonte: Manual da Sharp GP2Y0A21YK

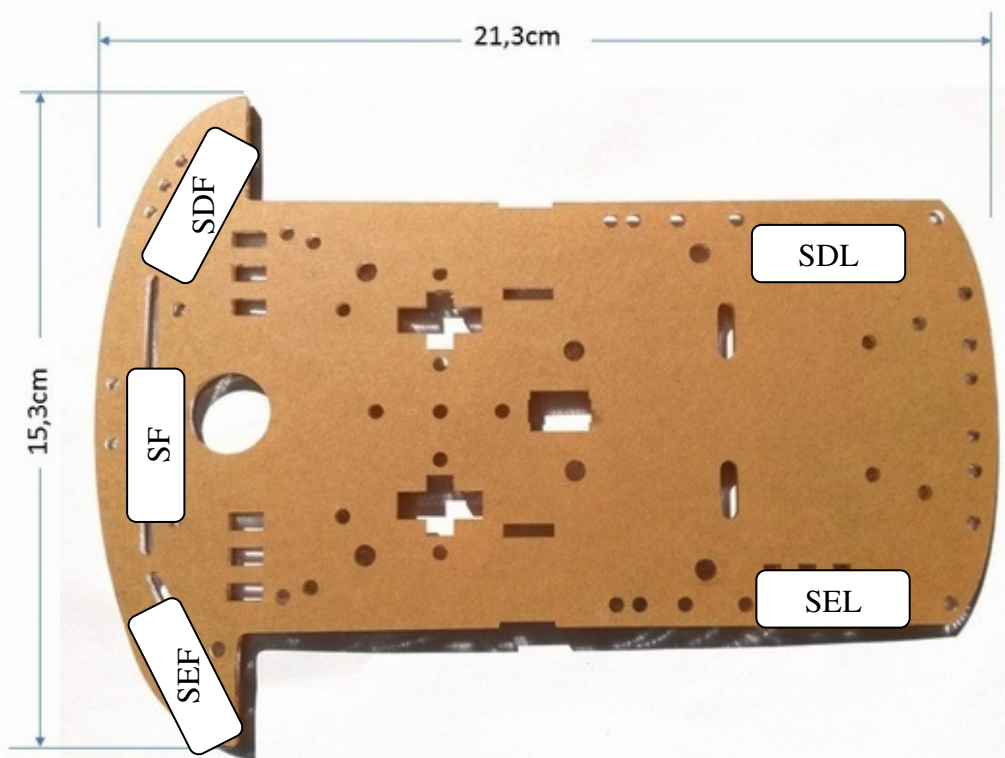
A escolha desse tipo de sensor se deu pelo seu baixo custo e pela facilidade de integração com a plataforma de controle Arduino UNO, pois podem ser alimentada



pela mesma fonte de energia - 5 V - e as portas de entradas analógicas do Arduino UNO são compatíveis com o sinal de saída do sensor, não sendo necessário nenhum hardware adicional de conversão de padrão elétrico.

Para efetuar o deslocamento correto do robô na pista de testes, os sensores foram posicionados de forma que houvesse uma leitura de  $180^\circ$  em torno do robô, possibilitando-lhe seguir a pista sempre em frente sem colidir com os obstáculos. Foi necessário o posicionamento estratégico dos sensores conforme apresentado na Figura 3.6, na qual vemos o **sensor direita lado** (SDL) posicionado a  $0^\circ$  do chassi com o objetivo de, juntamente com o **sensor esquerda lado** (SEL), posicionado a  $180^\circ$ , fornecer informações sobre qual direção o robô deve tomar em caso de curva conforme a sua distância das laterais da pista. Os sensores **esquerda frente** (SEF) e **sensor direita frente** (SDF), posicionados a  $50^\circ$  e  $130^\circ$  respectivamente, são responsáveis juntamente com o **sensor frontal** (SF), posicionado a  $90^\circ$ , por identificar a presença de curvas e obstáculos no caminho à frente do robô. Além disso, os sensores SEF e SDF também fornecem a informação para o alinhamento do robô no centro da pista quando este percorre uma reta.

**Figura 3.7** – Chassi com posicionamento dos sensores.

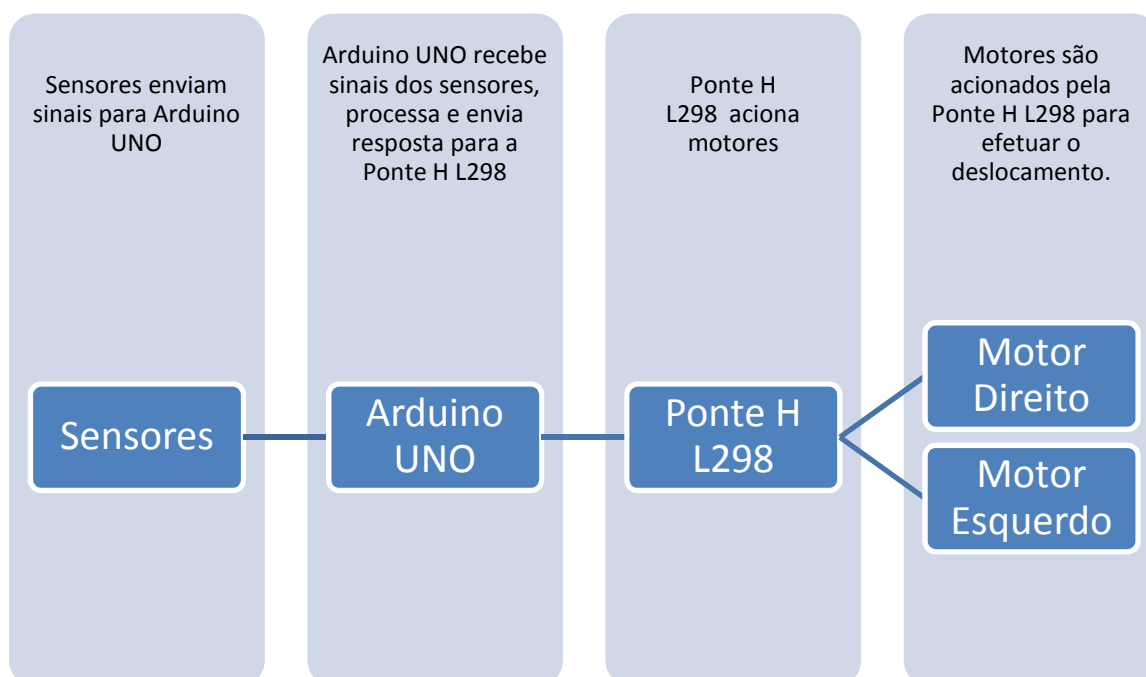


Devido à faixa de medição do sensor ser de 10 cm a 80 cm significa que, entre 0 cm e 10 cm tem-se uma zona morta. Por isso, optou-se então por recuar todos os sensores da margem da plataforma em 1,5 cm. Tal ação visa possibilitar que o veículo se desloque em pistas mais estreitas, pois desse modo a distância entre o robô e a parede da pista pode ser no mínimo maior ou igual a 8,5cm e não 10 cm - o que ocorreria se o sensor estivesse posicionado na borda da plataforma.

### 3.4 INTEGRAÇÃO DAS PLATAFORMAS

A integração entre os módulos eletrônicos pode ser vista na Figura 3.8, que apresenta o fluxo dos sinais elétricos do robô. Na primeira parte os sinais dos sensores são coletados e processados pelo controlador Arduino UNOe, com base nessas informações, são acionados os motores através da Ponte H L298que fará com que o robô efetue seu deslocamento sem colidir com os obstáculos.

**Figura 3.8**– Integração entre os módulos do robô



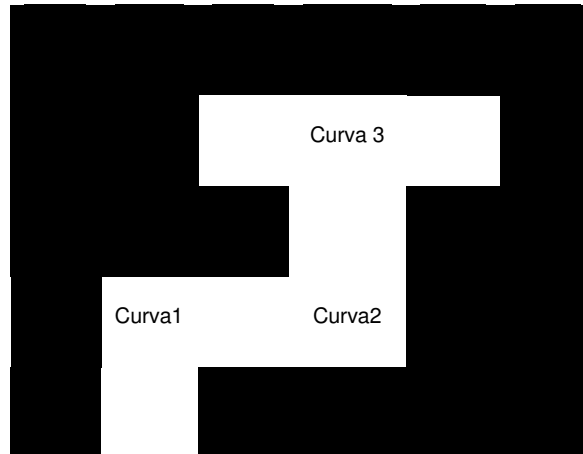
Fonte:Elaboração do autor

### 3.5 CENÁRIO PARA VALIDAÇÃO

Para efetuar a validação do sistema embarcado no robô foi desenvolvida uma pista de testes (Figura 3.9) que possibilitasse as três condições possíveis para o

deslocamento do robô: virar para direita de forma obrigatória (Curva 1), virar para a esquerda de forma obrigatória (Curva 2) e uma condição de decisão onde fosse possível virar tanto para a esquerda como para direita (Curva 3).

**Figura 3.9** - Desenho da pista de testes.



Fonte: Elaboração do autor.

A distância entre uma curva e outra é de 44 cm de largura e 120 cm de comprimento. Essas dimensões foram definidas com base no tamanho do robô e na zona morta do sensor, ou seja, a soma da largura do robô 15,3 cm mais a zona morta dos sensores 8,5cm (10 cm – 1,5 do recuo do sensor na plataforma) mais a distância de segurança 5,9 cm para cada lado que é definida pela distância necessária para que o robô efetue a curva sem colidir com as laterais da pista. Estas dimensões definem a largura da pista. Já o comprimento foi definido em 120 cm devido ao padrão de fabricação das placas laterais da pista.

### 3.6 PROBLEMA DE TOLERÂNCIA A FALHAS

Neste trabalho será considerado o comportamento do robô com todos os sensores funcionando corretamente e com falha de um ou dois sensores simultaneamente. Em todas essas condições espera-se que o robô consiga cumprir sua missão sem esbarrar em nenhum obstáculo.

A partir do momento que existe a falha de algum sensor, o sistema tem que ser capaz de reconstruir a informação do sensor em falha tendo como base as

informações dos demais sensores presentes no robô com o objetivo de eliminar a redundância do hardware.

O sensor será considerado em falha quando a medida de distância enviada pelo mesmo for igual a 0 e estiver diferente do valor simulado pelo sistema.

Na sequência, será analisado o comportamento do robô autônomo na possibilidade de falha de seus sensores, tendo como base as características abordadas na seção 3.3. Conforme apresentado, os sensores estão posicionados de forma que podemos detectar obstáculos em 5 pontos diferentes que cobrem 180° do robô. São apresentadas duas situações de operação: Caso 1 – Falha de um sensor; Caso 2 – Falha de dois sensores.

Para avaliar as falhas que podem ocorrer no sistema foi utilizada a técnica do FMEA - apresentada na seção 2.1 -, cujas informações estão nos Quadros 3.1 e 3.2. O primeiro quadro apresenta uma escala de 1 a 5 que determina o impacto causado pelas falhas dos sensores. Na escala é possível classificar o quão a falha é impactante para o sistema podendo ser igual a 1 caso não tenha efeito e igual a 5 caso o robô sofra uma pane geral em seus sensores ou apresente falhas em mais de dois sensores simultaneamente.

**Quadro 3.1-** Escala de severidade do efeito da falha

Severidade do efeito		Classificação
Extremo	Defeito catastrófico: falha de três ou mais sensores. Robô sofre pane geral. Impossibilitado de navegar.	5
Alto	Falha muito grave compromete dois sensores simultaneamente. Robô tem seu número de sensores reduzido e sua eficiência comprometida. Não é possível reconstruir o sinal perdido com as informações dos outros sensores.	4
Moderado	Falha muito grave compromete dois sensores simultaneamente. Robô tem seu número de sensores reduzido e sua eficiência comprometida, porém é possível reconstruir o sinal perdido com as informações dos outros sensores.	3
Baixo	Falha grave compromete um sensor. Robô tem seu número de sensores reduzido e sua eficiência comprometida, porém é possível reconstruir o sinal perdido com as informações dos outros sensores.	2
Nenhum	Sem efeito	1

Fonte: Elaboração do autor.

Os quadros 3.2 e 3.3 apresentam as etapas do processo que estão suscetíveis a falhas de identificação dos obstáculos nos seus respectivos ângulos, tanto de forma individual, como na falha de identificação de dois obstáculos

simultâneos. Cada etapa do processo (primeira coluna) tem um modo de falha potencial (segunda coluna), ou seja, a etapa do processo não pode ser cumprida em virtude de uma potencial falha. Em seguida, na terceira coluna, temos o potencial efeito da falha, ou seja, o efeito de cada falha caso o sensor não consiga identificar o obstáculo naquele ângulo, deste modo, podemos determinar o grau de severidade (quarta coluna) de cada falha no desempenho do robô com base nas regras estabelecidas no Quadro 3.1. Por fim, temos na quinta coluna as ações a serem tomadas para minimização do efeito da falha.

**Quadro 3.2** - Etapas do processo para falha de 1 sensor.

Etapas do processo	Modo de falha Potencial	Potencial Efeito da Falha	Severidade	Ações para controle do efeito da falha
Detecção de obstáculo a 90º	Sensor frontal SF gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de colisão com obstáculos frontais	2	Ignore o sinal do sensor SF e utilize os valores do sensor SEF e SDF
Detecção de obstáculo a 50º	Sensor direta frente SDF gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de curva para a direita	2	Ignore o sinal do sensor SDF e utiliza os valores do sensor SF, SEL, SDL e SEF
Detecção de obstáculo a 130º	Sensor esquerda frente SEF gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de curva para a esquerda	2	Ignore o sinal do sensor SEF e utilize os valores do sensor SF, SEL, SDL, SDF
Detecção de obstáculo a 0º	Sensor direita lado SDL gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de obstáculos na lateral direita	2	Ignore o sinal do sensor SDL e utilize os valores do sensor SEL
Detecção de obstáculo a 180º	Sensor esquerda lado SEL gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de obstáculos na lateral esquerda	2	Ignore o sensor SEL e utilize os valores do sensor SDL

Fonte: Elaboração do autor.

Analisando o quadro 3.2, a primeira linha apresenta a etapa em que o robô detecta obstáculos na posição 90º. Neste caso o potencial modo de falha seria do sensor SF que está posicionado à frente do robô e seria responsável por identificar obstáculos que venham à sua frente e, caso o mesmo apresente falha, o sistema deve pegar a informação dos sensores mais próximos que estejam direcionados o mais perto possível do sensor com defeito e utilizar esses sinais para reconstruir o

sinal do sensor em falha. Essa falha tem um grau de severidade 2 porque a falha compromete apenas um sensor e seu sinal pode ser reconstruído pela informação de outros sensores.

Quando ocorre a falha nos sensores laterais SDL e SEL ficam comprometidas a identificação dos obstáculos nas laterais do robô e por isso devem-se utilizar as informações do sensor SEL para simular o SDL e usar o SDL para simular o sensor SEL, pois estão em lados opostos porém simétricos. A sua falha acarreta em que o veículo não consegue identificar obstáculos nas laterais da pista. Já no caso do sensor SDF pode-se utilizar o conjunto de sensores SF-SDL-SEF ou SF-SEL-SEF a depender da disponibilidade dos sensores laterais e no caso do sensor SEF utilizam-se os conjuntos SF-SDL-SDF ou SF-SEL-SDF também a depender da disponibilidade dos sensores laterais, isso traz duas combinações possíveis para a simulação do mesmo sensor, sendo assim tem-se mais opções quando ocorre a falha simultânea de dois sensores e é apresentado no quadro 3.3.

**Quadro 3.3** - Etapas do processo para falha de 2 sensores simultaneamente

Etapas do processo	Modo de falha potencial	Potencial efeito da falha	Severidade	Ações para controle do efeito da falha
Detecção de obstáculo a 130º e 90º	Sensor esquerda frente SEF e sensor frontal SF geram valores diferentes do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de curvas à esquerda e obstáculos à frente	4	Sem ação recomendada.
Detecção de obstáculo a 50º e 90º	Sensor direita frente SDF e sensor frontal SF geram valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de curvas à direita e obstáculos à frente	4	Sem ação recomendada.
Detecção de obstáculo a 130º e 50º	Sensor esquerda frente SEF e sensor direita frente SDF gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de curvas tanto para a direita como para a esquerda	4	Sem ação recomendada

Detecção de obstáculo a 180º e 90º	Sensor esquerda lado SEL e sensor frontal SF geram valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de obstáculos na frente e na lateral esquerda	3	Ignore os sinais dos sensores SEL e SF e considere os valores dos sensores SEF, SDL e SDF
Detecção de obstáculo a 0º e 90º	Sensor direita lado SDL e sensor frontal SF geram valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de obstáculos à frente e na lateral direita	3	Ignore os sinais dos sensores SDL e SF e considere os valores os valores dos sensores SEF, SEL e SDF
Detecção de obstáculo a 180º e 50º	Sensor esquerda lado SEL e sensor direita frente SDF gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de curvas a direita e obstáculos na lateral esquerda	3	Ignore os sinais dos sensores SEL e SDF e considere os valores dos sensores SF, SEF e SDL
Detecção de obstáculo a 50º e 0º	Sensor direita frente SDF e sensor direita lado SDL gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Não identificação de obstáculos em toda a lateral direita do robô	3	Ignore os sinais dos sensores SDF e SDL e considere os valores os valores dos sensores SF, SEL e SEF
Detecção de obstáculo a 180º e 130º	Sensor esquerda lado SEL e sensor esquerda frente SEF gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Não identificação de obstáculos em toda a lateral esquerda do robô	3	Ignore os sinais dos sensores SEL e SEF e considere os valores dos sensores SDF, SDL e SF
Detecção de obstáculo a 180º e 0º	Sensor esquerda lado SEL e sensor direita lado SDL gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de obstáculos nas laterais	4	Sem ação recomendada.
Detecção de obstáculo a 130º e 0º	Sensor esquerda frente SEF e sensor direita lado SDL gera valor diferente do sinal simulado pelo modelo <i>Fuzzy</i>	Possibilidade de não identificação de curvas a esquerda e obstáculos na lateral direita	3	Ignore os sinais dos sensores SEF e SDL e considere os valores os valores dos sensores SF, SDF e SEL

Fonte: Elaboração do autor.

Fazendo uma análise nas etapas do processo onde ocorre a falha de 2 sensores simultaneamente (quadro 3.3) é possível perceber que quando ocorre a falha das combinações de dois sensores da frente do veículo, ou seja, SEF-SF, SDF-SF e SEF-SDF não se pode simular o sinal dos mesmos, pois são necessários

peelo menos dois sensores funcionando corretamente para que o robô consiga identificar obstáculos à frente e efetuar o alinhamento se afastando da lateral da pista durante seu deslocamento em uma reta. Temos também a combinação SEL-SDL, essa combinação não pode ser utilizada devido a um sensor ser responsável pela simulação do outro sendo assim se os dois falharem ambos os sinais não podem ser reconstruídos.

Já quando falham a combinação SEF-SDF, fica comprometida a identificação de curvas e por isso utiliza-se o SF para identificar se o veículo está em uma reta ou não, e os sensores SEL-SDL para verificar o sentido da curva.

Na condição onde as combinações em falha são SF-SEL ou SF-SDL, fica comprometida a identificação de obstáculos à frente e nas laterais do robô; isso prejudica sua centralização na pista e o posicionamento correto para efetuar a curva. Para resolver isso utiliza-se os conjuntos de sensores SEF-SDF-SDL ou SDF-SEF-SEL, respectivamente, para que através dos sensores SEL e SDL se possa ter uma noção de centralização da pista e nos sensores frontais SEF, SDF possa-se identificar a presença de curvas e obstáculos à frente.

Quando falham os conjuntos de sensores SEL-SDF e SDL-SEF temos prejudicada a capacidade de identificar curvas para a direita e obstáculos na lateral esquerda ao mesmo tempo ou curvas para a esquerda e obstáculo para a direita, respectivamente. Desta forma, é possível simular o sinal desses sensores através da combinação SF-SEF-SDL ou SF-SDF-SEL, em que se pode detectar obstáculos à frente com o sensor SF, as curvas através dos sensores SEF ou SDF e SEL ou SDL quando disponível.

Já os casos onde falham a combinação dos sensores SEF-SEL e SDF-SDL o robô perde a capacidade de identificar obstáculos na lateral esquerda ou direita do robô respectivamente, nestes casos são utilizados os sensores dos lados opostos para efetuar a simulação do sinal juntamente com o sensor frontal, ou seja, utilizam-se as combinações SDL-SF-SDF ou SEL-SF-SEF. Com isso utilizamos os sensores SF, SEF ou SDF para identificar obstáculos a frente e os sensores SDL ou SEL para identificar para qual lado se deve seguir.

Analisando grau de severidade das falhas percebe-se que não existe falha que não cause um efeito no sistema. Percebe-se também que o processo de falha exclusivamente de um sensor tem um grau de severidade baixo ou 2, já que podemos reconstruir os sinais perdidos através dos outros sensores. Já quando



falham dois sensores simultaneamente podemos ter a classificação 4, nos casos onde falham os sensores SEL-SDL, SEF-SF, SEF-SDF ou SDF-SF, devido à incapacidade do sistema de reconstruir os sinais perdidos. Porém, nos demais casos onde falham dois sensores podemos reconstruir os sinais perdidos com a informação dos outros sensores, sendo assim têm um grau de severidade 3.

Neste trabalho só foi considerada a falha de até 2 sensores simultâneos, por isso o grau de severidade 5 não foi implementado, uma vez que a falha de três ou mais sensores, independente de quais sejam, inviabiliza o correto funcionamento do sistema devido à perda de, no mínimo, 60% dos sinais e com isso o sistema não tem condições de reconstruir os sinais.

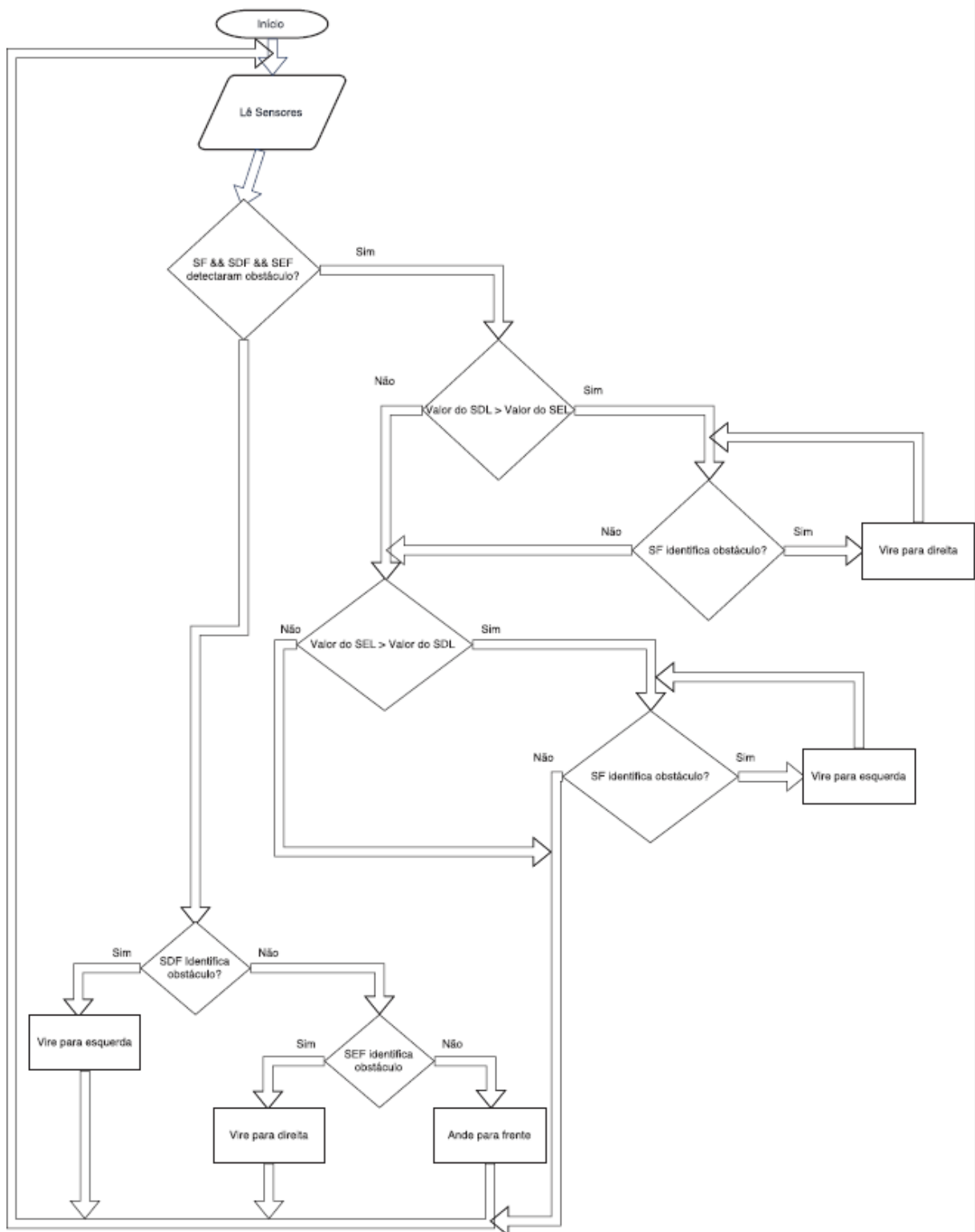
## 4 DESENVOLVIMENTO DO ALGORITMO

O robô apresentado tem um algoritmo de base que é o seu sistema operacional e foi desenvolvido para trabalhar com todos os sensores sem falha. Seu fluxograma pode ser visto na figura 4.1A. Este algoritmo de base não será alterado e, caso haja falha de um sensor, a informação do mesmo tem que vir de forma simulada sem que o sistema de base precise se ajustar à falha do sensor com problema.

O sistema operacional trabalha de forma que primeiramente é efetuado a leitura de todos os sensores e com base nessas informações são tomadas as decisões a partir dos sinais enviados pelos sensores. De início é verificado se existe algum obstáculo à frente comum aos três sensores SF-SDF-SEF; caso não encontre é verificado com os sensores laterais SDF e SEF se foram encontrados obstáculos, caso somente o SDF encontre um obstáculo é acionado o comando de virar para esquerda e caso somente o sensor SEF detecte obstáculo é acionado o comando vire para a direita. Caso nenhum dos dois encontrem obstáculos, o robô deve ir em frente e reiniciar o ciclo.

Porém, caso o conjunto de sensores SF-SDF e SEF detecte um obstáculo significa que o robô está de frente para uma parede em uma curva e, portanto, deve parar e decidir para qual lado virar para efetuar a curva. Isso é feito através dos sensores SDL e SEL: quem estiver mais distante do obstáculo informa para qual lado o robô deve virar, ou seja, se o valor do sinal enviado pelo SDL for maior que o SEL, o robô deve virar para a direita até que o sensor SF não identifique mais obstáculo, garantindo que o robô pode seguir em segurança. O mesmo raciocínio se aplica ao sensor SEL.

**Figura 4.1** - Estrutura geral do sistema operacional do robô



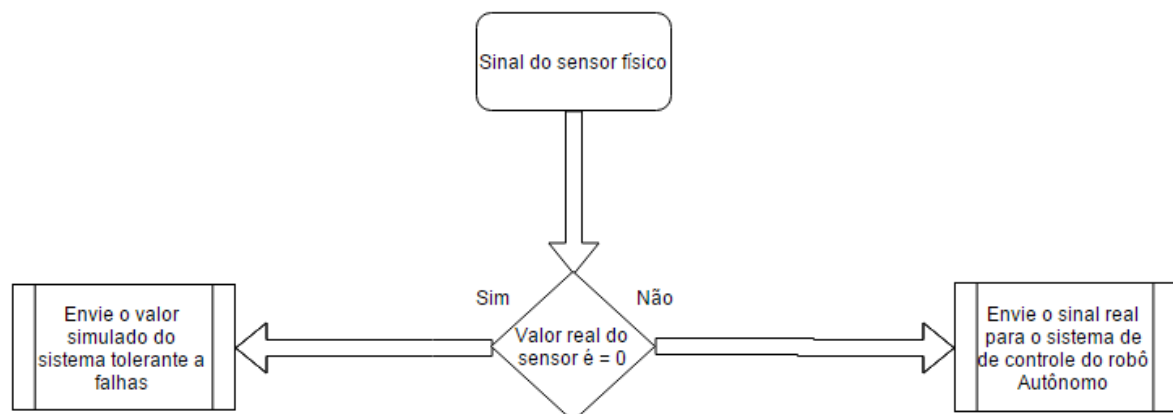
Fonte: Elaboração do autor.

Na estrutura utilizada neste trabalho o sistema de tolerante a falhas (Figura 4.2), é um módulo lógico separado e independente do algoritmo de controle do robô. Isso facilita o processo de desenvolvimento e manutenção do sistema. Seguiu-se, aqui, a mesma linha de Malar (2016) que também retira o processo de detecção e tolerância a falhas do programa principal do robô, diferentemente de Kamel (2016), que trabalham com o sistema tolerante a falhas integrado aos controladores do robô autônomo.

O sistema tolerante a falhas proposto nesta dissertação é responsável por efetuar a leitura dos 5 sensores do robô e, caso algum sensor seja detectado com falha, automaticamente o módulo de detecção de falhas suspende o envio do sinal com falha e envia em seu lugar o sinal construído pelo sistema tolerante a falhas com base na informação dos demais sensores. Essa estratégia faz com que o processo de tolerância a falhas seja transparente para o sistema de controle, o que irá possibilitar uma menor complexidade do seu desenvolvimento e manutenção, além de assegurar a autonomia para o robô, já que este conseguirá tolerar as falhas de seus sensores.

Deste modo, conforme apresentado na Figura 4.2, caso o sinal real do sensor seja igual a 0, isso representa uma falha e assim o sistema tolerante a falha identifica a situação e envia o sinal simulado do sensor específico no lugar do valor real. Devido às características técnicas dos sensores, as faixas de operação estão compreendidos entre 10cm a 80cm.

**Figura 4.2-** Estrutura geral do sistema tolerante a falhas

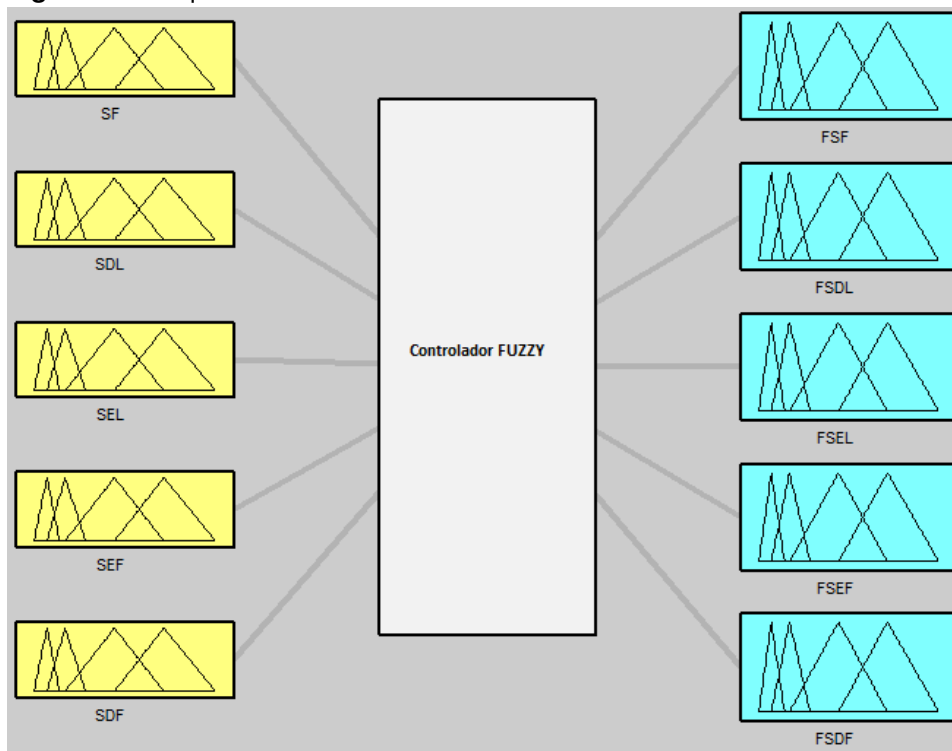


Fonte: Elaboração do autor.

#### 4.1 ESTRUTURA DO SISTEMA TOLERANTE A FALHAS

A estrutura apresentada na Figura 4.3 representa o esquema geral do controlador nebuloso em que as entradas são os sinais originais provenientes dos sensores e a saída são os valores simulados dos sinais dos sensores pela lógica *Fuzzy*. Por este motivo, quando se tratar de sinais simulados pela lógica *Fuzzy* é acrescentada a letra “F” à frente da nomenclatura do sensor, ou seja, o SF seria o sinal original do sensor frontal e o FSF é o seu sinal simulado. As informações provenientes do sistema foram validadas com o software MATLAB com o *toolboxfuzzy*, apresentadas nas figuras 4.3 e 4.4.

**Figura 4.3-** Esquema do sistema tolerante a falhas



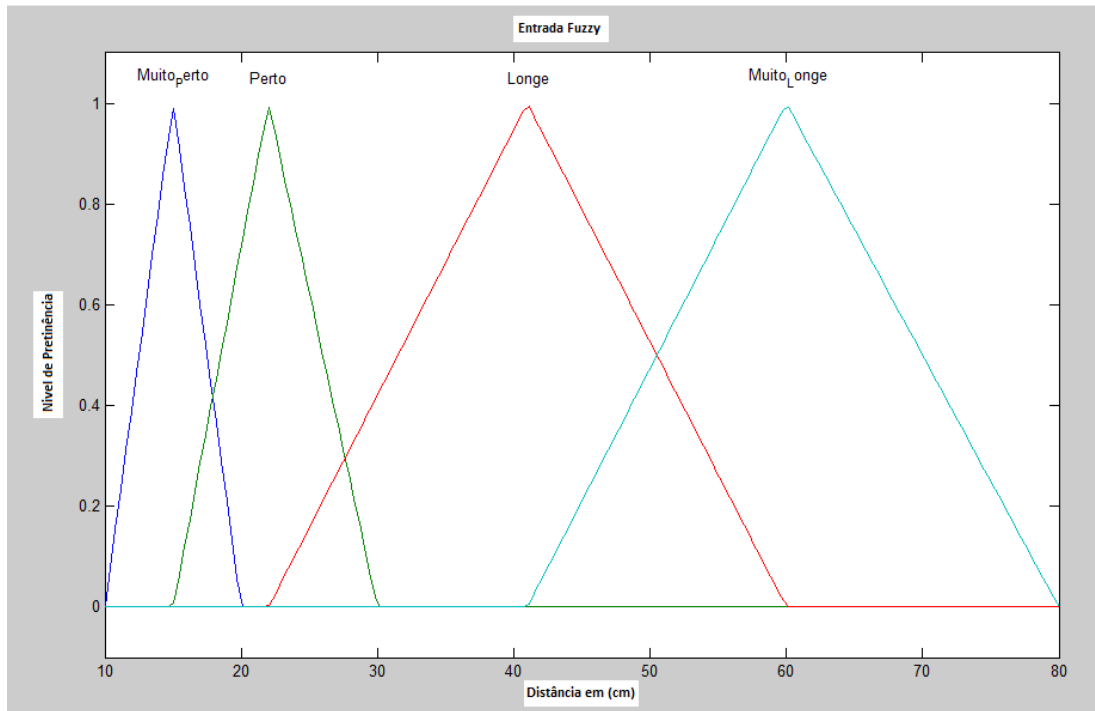
Fonte: Elaboração do autor.

Caldas (2004) apresenta uma estrutura para seu controlador semelhante à apresentada na Figura 4.3, porém em seu trabalho é tratada a informação de falhas em atuadores de um veículo autônomo submarino.

O sistema *Fuzzy* simula o sinal de todos os sensores independentemente de estarem em falha ou não. Porém, só enviam o valor simulado para o sistema operacional do robô caso o sensor físico do robô entre em falha enviando o valor 0 e seja diferente do valor simulado. Os critérios de geração de sinal são determinados

por um controlador nebuloso que tem o seu processo de desfuzzificação calculado pela fórmula do Centro-da-Área apresentado na Equação 2.1.

**Figura 4.4-** Representação gráfica dos valores linguísticos das variáveis de entrada



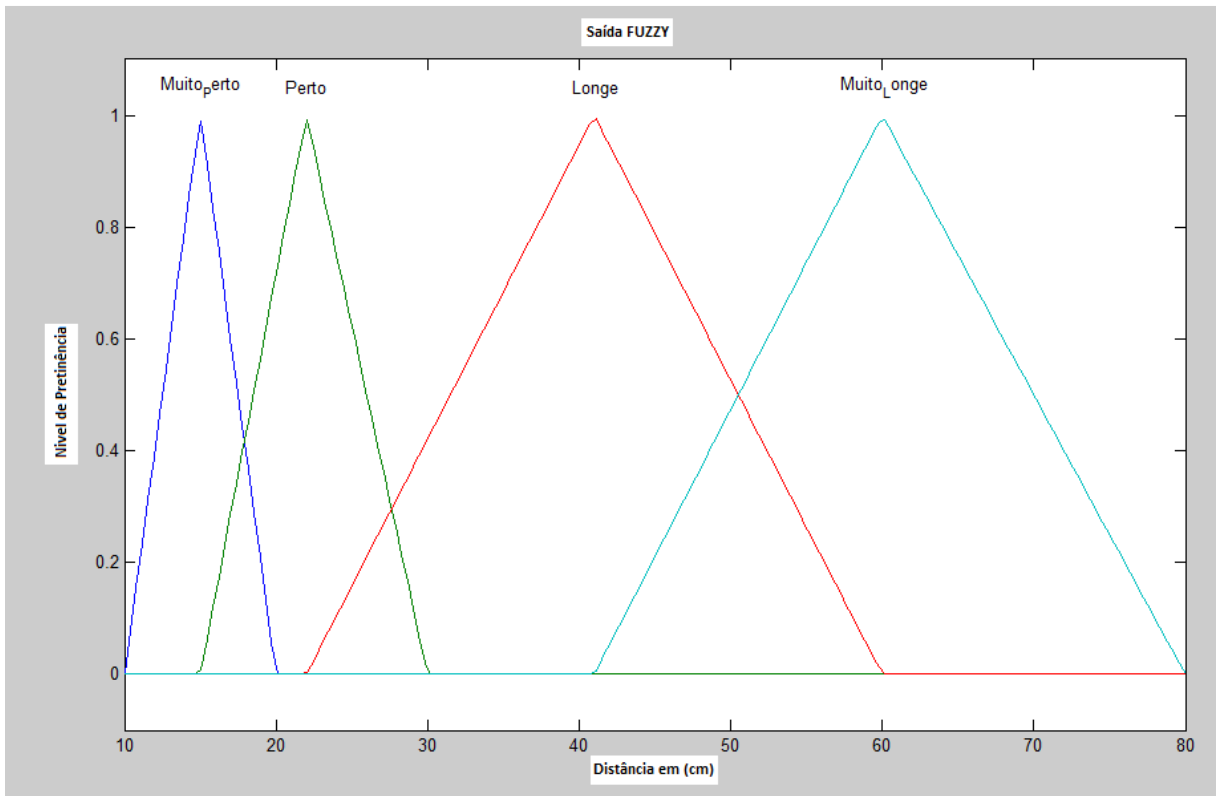
Fonte: Elaboração do autor

As variáveis nebulosas de entrada e saída foram implementadas em quatro categorias conforme Figura 4.4 e Figura 4.5. Essas categorias são:

- **Muito Perto:** varia entre 10 cm e 15 cm. Nesta distância, o robô precisa tomar uma ação imediata para não colidir.
- **Perto:** varia entre 15cm e 30 cm. Essa variável identifica a aproximação do obstáculo.
- **Longe:** varia entre 22 cm e 60 cm. Nesta faixa estão reunidos os valores que possuem até 16% de erro de medição devido a ruído externos.
- **Muito Longe:** varia entre 41cm e 80 cm. Essa faixa compreende os valores que possuem até 48% de erro de medição devido a ruídos externos.

Devido ao nível de imprecisão as variáveis linguísticas 'Longe' e 'Muito Longe' são pouco utilizadas para a construção das regras da lógica *Fuzzy*. Tal imprecisão é discutida com mais detalhes no capítulo 6 – Limitações da plataforma.

**Figura 4.5-** Representação gráfica dos valores linguísticos variáveis de saída



Fonte: Elaboração do autor

## 4.2 REGRAS NEBULOSAS

As regras nebulosas foram desenvolvidas tendo em consideração a disposição geométrica dos sensores apresentada na Figura 3.7 e conforme a opinião do especialista. As entradas do sistema nebuloso são os valores reais de todos os 5 sensores presentes no robô. As saídas do sistema nebuloso são os sinais simulados de todos os sensores com base na informação dos demais sensores.

Tendo como base a disposição geométrica dos sensores e o conhecimento intrínseco do especialista para o desenvolvimento da lógica *Fuzzy*, foram levantados conjuntos de regras para a simulação dos sensores com base nas informações dos Quadro 3.2 e 3.3, nos quais são apresentadas todas as combinações possíveis de falhas de 1 ou de 2 sensores simultaneamente, assim como o conjunto de sensores que podem ser utilizados para a simulação do sinal original. Ou seja, é possível ter mais de uma possibilidade de simular o mesmo sensor. Na sequência, são exploradas essas possibilidades e as respectivas regras do sistema *Fuzzy*. Devido à simetria dos sensores SDF e SEF e dos sensores SDL e SEL, a descrição das regras foram apresentadas em um único tópico em decorrência da similaridade das regras.

#### 4.2.2 - Falha no Sensor Frontal – SF

A principal função do sensor SF é identificar a aproximação de uma curva juntamente com os sensores SEF e SDF.

Para que seja constatada uma curva, os três sensores SF-SDF-SEF têm que concordar que o obstáculo está “Muito Perto”. Para a simulação do sensor FSF foram utilizadas as seguintes regras.

- Se (SEF ou SDF é Muito Perto) então (FSF is Muito Perto)
- Se (SEF ou SDF é Perto) então (FSF é Perto)

Na simulação do SF, os sensores SEF e SDF são utilizados, ou seja, caso o SDF ou SEF esteja “Perto” a saída FSF é igual a “Perto”, ou caso esteja “Muito Perto” a saída FSF deverá ser “Muito perto”.

#### 4.2.3 - Falha no Sensor Esquerda Frente – SEF ou Sensor Direita Frente – SDF

Os sensores SEF e SDF têm duas funções; a primeira seria identificar a proximidade do robô com a lateral esquerda ou direita da pista, respectivamente, e a segunda seria identificar, juntamente com o SF, a presença de uma curva.

É possível simular o sensor SEF através de duas combinações de sensores diferentes. A primeira seria pelos sensores SEL-SF-SDF e SDL-SDF-SF. Já o sensor SDF pode ser simulado através das combinações SEL-SF-SEF e SDL-SEF-SF. As seguintes regras são utilizadas para sua simulação dos sensores:

**Caso A** - Simulação SEF através da combinação SEL-SF-SDF:

- Se (SEL é Muito\_Perto) e (SF é Longe) então (FSEF é Muito Perto)
- Se (SDF é Longe) e (SF Perto) então (FSEF é Muito Perto)

Essas regras têm como objetivo simular o sensor SEF de modo que o mesmo consiga diferenciar se o robô está diante de uma curva ou se está desalinhado na pista. Portanto, se o sensor SEL estiver “Muito Perto” e o SF estiver “Longe” significa que a plataforma está inclinada e, por isso, a saída FSEF deve estar “Muito Perto” para que o robô tenha tempo de corrigir seu posicionamento. Isso também se



observa na segunda regra onde se o SDF está “Longe” e o SF está “Perto”; o FSEF tem que estar muito perto para que o robô corrija seu posicionamento na pista.

**Caso B** - Simulação SEF através da combinação SDL-SF-SDF:

- Se (SDL é Perto) e (SF é Longe) então (FSEF é Muito perto)
- Se (SDF é Longe) e (SF é Perto) então (FSEF é Muito Perto)

Neste caso é efetuada a substituição do SEL pelo SDL, o que para a análise só muda a primeira regra onde o SDL (que está do lado oposto do SEL) juntamente com o SF identificam se a plataforma está inclinada ou não. Para isso utiliza-se a relação inversa, ou seja, se no primeiro caso o SEL estava “Muito Perto”, nesse mesmo momento o SDL estava “Perto” da outra lateral da pista.

**Caso A** - Simulação SDF através da combinação SDL-SF-SEF

- Se (SDL é Perto) e (SF é Longe) então (FSDF é Muito perto)
- Se (SEF é Longe) e (SF é Perto) então (FSDF é Muito Perto)

**Caso B** - Simulação SDF através da combinação SEL-SF-SEF

- Se (SEL é Muito\_Perto) e (SF é Longe) então (FSDF é Muito Perto)
- Se (SEF é Longe) e (SF Perto) então (FSDF é Muito Perto)

Na simulação do SDF seguem-se as mesmas regras do da simulação do SEF, porém com os sensores correspondentes às suas respectivas posições simétricas na plataforma.

#### 4.2.4 - Falha no Sensor Direita Lado – SDL e Sensor Esquerda Lado – SEL

A principal função dos sensores SDL e SEL consiste em, após os sensores SF-SEF-SDF identificarem uma curva, informar para qual lado o robô deve seguir. Durante o seu processo de simulação através da lógica *Fuzzy* podemos utilizar o SDL se quisermos simular o sinal do SEL ou utilizar o SEL se quiser simular o sinal do SDL, com base nas regras a seguir.

Caso o SDL entre em falha, o sistema pode simular o sinal do mesmo utilizando a informação do sensor SEL baseado nas seguintes regras:

- Se (SEL é Muito Perto) então (FSDL é Muito Longe)
- Se (SEL é Perto) então (FSDL is Muito Perto)

Ou seja, caso haja uma curva para a direita, o sensor SEL deve estar “Muito Perto” da parede e conseqüentemente o valor simulado FSDL deve estar “Muito Longe”. Porém se a curva for para esquerda o sensor SEL deve estar maior ou igual à distância “Perto”, então isso significa que o valor simulado FSDL está próximo ao obstáculo.

O mesmo se aplica ao SEL, porém de forma oposta, baseado nas seguintes regras:

- Se (SDL é Muito Perto) então (FSEL é Muito Longe)
- Se (SDL é Perto) então (FSEL is Muito Perto)

Para a definição das regras do sistema *Fuzzy* foi levado em conta a opinião do especialista que avalia as posições dos sensores no robô e define qual sensor tem mais peso no processo de simulação de cada conjunto de sensores em falhas. Deste modo, a simulação de alguns sensores pode ser feita de forma mais exata e com menor custo computacional, pois as regras implementadas são as mínimas necessárias para a correta simulação do sistema. A redução do número de regras é necessária devido à baixa capacidade de processamento do micro controlador da placa Arduino UNO, por isso somente foram implementadas as regra essenciais para correto funcionamento do sistema. Caso ocorra alguma condição que não está prevista pelas regras estabelecidas, o sistema automaticamente envia um sinal com o valor -1 para a saída do sensor a ser simulado; essa condição é indiferente para o sistema de controle do robô e não afeta suas decisões.

No momento em que falham dois sensores simultaneamente, o sistema apenas verifica quais sensores estão com defeito e utiliza suas regras individuais correspondentes para simulações de seus sinais.

## 5 IMPLEMENTAÇÃO E ANÁLISE DOS RESULTADOS

Essa pesquisa visou desenvolver um sistema tolerante a falhas de sensores que permitisse ao robô autônomo reconstruir o sinal de qualquer um de seus sensores com base na informação dos demais sensores presentes no robô.

Na ocorrência de uma falha, o sistema automaticamente ignora a informação do sensor defeituoso e, em seu lugar, trabalha com as informações dos demais sensores que simulam o sinal do sensor defeituoso, com isso o sistema operacional de controle do robô tem sempre a informação dos sensores de forma transparente a falhas presentes nos mesmos.

### 5.1 PARÂMETROS DE MEDIÇÃO

Com base na pista apresentada na Figura 3.9, existem três ações básicas que o robô deverá cumprir para que se possa considerar que o mesmo teve um desempenho satisfatório: a Curva 1 para direita (CD); Curva 2 para esquerda (CE) e Curva 3, que é um processo de decisão no qual o robô pode decidir se vai para a direita ou esquerda. Essa decisão deve ser tomada pelos sinais dos sensores SEL e SDL, que devem medir a maior distância para o obstáculo possibilitando definir para qual lado o robô deve girar. Neste último caso não foi arbitrado para qual lado o robô deve seguir devido à grande imprecisão dos sensores quando estão medindo distâncias “Muito Longe”; como a oscilação do sinal dos sensores é grande – em torno de 48% – isso causa uma instabilidade nas comparações dos sinais. Por isso, apenas se leva em consideração se o robô consegue virar para a direita ou para a esquerda.

Os parâmetros utilizados para medir o desempenho do robô são apresentados no quadro 5.1. Esses parâmetros são necessários para verificar o desempenho dos sensores físicos e o desempenho da simulação dos sensores pela lógica *Fuzzy*.

**Quadro 5.1-** Parâmetros de operação do robô

<b>Parâmetros de medição de desempenho</b>	
OK	Informa que o robô efetuou a curva de forma satisfatória
OK1	Informa que o robô efetuou a curva de forma satisfatória porém ocorreu colisão de leve com a lateral da pista
NOK1	Informa que o robô causou uma forte colisão na pista, mas conseguiu se recuperar
NOK2	Informa que o robô se perdeu em seu trajeto, mas conseguiu se recuperar
NOK3	Informa que o robô se perdeu no trajeto ou colidiu de forma que não consegue se recuperar
X	Informa que aquela curva não foi executada porque o robô não conseguiu chegar àquele ponto

Fonte: Elaboração do autor.

Ainda conforme apresentado no quadro 5.1, temos algumas nomenclaturas com objetivo de classificar o comportamento do robô durante sua missão. Essa nomenclatura varia desde “**OK**” designando que o robô efetuou a curva sem colidir com os obstáculos; “**OK1**” informando que ocorreram colisões leves com a lateral da pista, mas o deslocamento do robô não foi afetado; “**NOK1**”, o robô teve uma forte colisão na pista, mas conseguiu se recuperar; “**NOK2**”, o robô se perdeu do trajeto porém conseguiu se recuperar; “**NOK3**”, o robô se perdeu no seu trajeto ou colidiu sem conseguir se recuperar e, por fim, “**X**” que informa que a curva não foi executada porque o robô não conseguiu chegar àquele ponto. Com isso, espera-se qualificar e quantificar as informações mais importantes sobre o deslocamento do robô em sua missão.

Todas as situações – robô com todos os sensores funcionando corretamente e robô com falhas, conforme descrição da subseção 4.2 (Regras nebulosas) – foram testadas 10 vezes e seus valores anotados em quadros para realização das análises. Cada quadro se refere a uma situação diferente. Eles são apresentados a seguir.

## 5.2 RESULTADOS DOS TESTES COM TODOS OS SENSORES FUNCIONANDO CORRETAMENTE

Para análise comparativa de desempenho foi testado o veículo com todos os seus sensores em funcionamento correto por 10 vezes. Os resultados dos testes são apresentadas no Quadro 5.2.

**Quadro 5.2-** Resultados do testes com todos os sensores funcionando.

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK1	% NOK1	% NOK2	% NOK3	% X
Curva1	OK	OK	OK	OK	OK	OK	OK	OK1	OK	OK	90%	10%	0%	0%	0%	0%
Curva2	OK	OK	OK	OK1	OK1	OK	OK	OK1	OK	OK	70%	30%	0%	0%	0%	0%
Curva 3	OK	OK1	OK	OK1	OK	OK	OK	OK1	OK	OK	70%	30%	0%	0%	0%	0%
Total Geral											77 %	23%	0%	0%	0%	0%

Fonte:Elaboração do autor

Com base nas informações do quadro acima, pode-se verificar que com todos os sensores funcionando corretamente temos um índice de acerto de 100% que estão divididos entre as nomenclaturas “OK” e “OK1”, ou seja, neste caso, mesmo o robô colidindo 23% das vezes isso não impactou de forma grave a sua missão, pois em todos os casos o robô conseguiu atingir o seu destino de forma satisfatória.

A colisão gerada no teste se dá pela imprecisão da plataforma – que será discutida no item 6 Limitações da plataforma. As caixas de redução têm uma imprecisão que causa uma demora na movimentação das rodas e há um atraso no acionamento/desacionamento da Ponte H L298. Por isso, quando o robô chega perto do obstáculo e o controle envia o sinal de “pare” para as rodas, a plataforma leva um tempo até executar o comando; esse tempo faz com que ocorram algumas pequenas colisões com a pista.

## 5.3 RESULTADOS DOS TESTES COM FALHA DE 1 SENSOR

### 5.3.1 Resultados em caso de falha no Sensor Direita Lado – SDL e Sensor Esquerda Lado - SEL

O processo de falha do sensor SDL e/ou SEL impossibilita o robô de identificar para qual lado deve seguir no momento em que encontra uma curva. Os resultados dos testes desses sensores podem ser vistos nos quadros 5.3 e 5.4. Ambos os quadros têm resultados semelhantes, pois tratam da mesma informação em direções opostas.

**Quadro 5.3-** Simulação do sensor SDL utilizando SEL.

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK 1	% NOK 1	% NOK 2	% NOK 3	% X
Curva 1	OK 1	OK	OK	OK	OK	NOK 3	OK	OK	OK	OK	80 %	10 %	0%	0%	10%	0%
Curva 2	OK	OK	OK	OK	OK	X	NOK 3	OK	NOK 3	OK 1	60 %	10 %	0%	0%	20%	10 %
Curva 3	OK	OK	OK	OK	OK	X	X	OK	X	OK	70 %	0%	0%	0%	0%	30 %
Total Geral											70 %	7%	0%	0%	10%	13 %

Fonte:Elaboração do autor

**Quadro 5.4-** Simulação do sensor SEL utilizando SDL

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK1	% NOK1	% NOK2	% NOK3	% X
Curva 1	NOK 3	OK	OK	OK1	OK1	OK1	OK	OK	OK	OK	60%	30%	0%	0%	10%	0%
Curva 2	X	NOK3	OK	OK	OK	OK	NOK1	OK	NOK3	NOK3	50%	10%	0%	0%	30%	10%
Curva 3	X	X	OK	OK	OK	X	OK	OK	X	X	50%	0%	0%	0%	0%	50%
Total Geral											54%	13%	0%	0%	13%	20%

Fonte:Elaboração do autor

Pode-se perceber que na simulação do SDL o robô executou a curva perfeitamente ou colidiu com levemente com obstáculos em 77% das vezes

conforme apresentados pelas nomenclaturas “OK” e “OK1”. Isso se repete de forma semelhante no sensor SEL onde as nomenclaturas “OK” e “OK1” juntas somam a maior parte das ações totalizando 67%.

Observa-se que os dois casos apresentam ocorrência do tipo “NOK3”, ou seja, em 13% das vezes o SEL e em 10% das vezes o SDL fazem com que o robô se perca no trajeto ou colidasem conseguir se recuperar. Isso pode ser causado pela demora da Ponte H L298 em executar os comandos provenientes do controlador Arduino UNO e quando isso acontece não é possível continuar a missão conforme apresentado no Quadro 5.3 do SDL nos testes 6, 7, 9 e no Quadro 5.4 do SEL nos testes 1, 2, 9, 10.

### 5.3.2 Resultados em caso de falha no Sensor Frente – SF

A falha do sensor frontal aumenta consideravelmente o número de colisões frontais do tipo “NOK3” (Quadro 5.5). Essas colisões aumentam pelo fato dos dois sensores SDF e SEF serem mais lentos para identificar a distância para o obstáculo devido ao seu posicionamento. As colisões ocorrem quando o veículo se aproxima rapidamente de uma curva e pela falta do sensor frontal e pela demora de resposta do sistema eletrônico de acionamento/desacionamento das rodas o robô não consegue desviar a tempo.

Conforme apresentado no quadro 5.5, é possível observar que mesmo com aumento do número de colisões frontais o robô teve êxito em 60% das tentativas de curvas, ou seja, o robô executou de forma correta ou com colisões leves conforme a soma das nomenclaturas “OK” e “OK1”.

**Quadro 5.5-** Simulação do sensor SF utilizando SEF-SDF

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK 1	% NOK 1	% NOK 2	% NOK 3	% X
Curva 1	OK	OK	OK	OK	OK	OK	OK 3	OK	OK 1	OK 1	70 %	20 %	0%	0%	10%	0%
Curva 2	OK 1	OK 1	OK 1	OK 1	OK 3	OK 1	X	OK	OK 3	OK 3	50 %	10 %	0%	0%	30%	10 %
Curva 3	OK	OK 3	OK 3	OK 1	X	X	X	OK 1	X	X	10 %	20 %	0%	0%	20%	50 %
Total Geral											43 %	17 %	0%	0%	20%	20 %





**Quadro 5.7-** Simulação do sensor SEF utilizando SF-SDL-SDF (Caso B)

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK1	% NOK1	% NOK2	% NOK3	% X
Curva 1	NOK2	NOK2	NOK1	NOK3	NOK2	NOK3	NOK2	NOK2	OK	NOK3	10%	0%	10%	50%	30%	0%
Curva 2	NOK1	OK	OK	X	NOK1	X	NOK3	NOK3	NOK3	X	20%	0%	20%	0%	30%	30%
Curva 3	NOK1	NOK2	OK	X	NOK1	X	X	X	X	X	10%	0%	20%	10%	0%	60%
Total Geral											13%	0%	17%	20%	20%	30%

Fonte:Elaboração do autor

**Quadro 5.8 -** Simulação do sensor SDF utilizando SF-SDL-SEF (Caso A)

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK1	% NOK1	% NOK2	% NOK3	% X
Curva 1	OK	OK	NOK1	OK	OK	NOK2	NOK1	OK	OK	OK	70%	0%	20%	10%	0%	0%
Curva 2	NOK3	NOK1	NOK3	NOK1	NOK2	NOK3	NOK3	NOK2	NOK2	NOK3	0%	0%	20%	30%	50%	0%
Curva 3	X	NOK1	X	NOK2	NOK3	X	X	NOK1	NOK1	X	0%	0%	30%	10%	10%	50%
Total Geral											23%	0%	23%	17%	20%	17%

Fonte:Elaboração do autor

**Quadro 5.9-** Simulação do sensor SDF utilizando SF-SEL-SEF (Caso B)

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK1	% NOK1	% NOK2	% NOK3	% X
Curva 1	NOK2	NOK2	NOK3	NOK1	NOK3	NOK3	NOK2	NOK2	NOK2	NOK2	0%	0%	10%	60%	30%	0%
Curva 2	NOK1	NOK1	X	NOK1	X	X	NOK1	NOK1	NOK3	NOK2	0%	0%	50%	10%	10%	30%
Curva 3	OK	NOK1	X	NOK2	X	X	NOK3	NOK3	X	NOK3	10%	0%	10%	10%	30%	40%
Total Geral											4%	0%	23%	27%	23%	23%

Fonte:Elaboração do autor.

Percebe-se também um aumento na quantidade de vezes em que o robô se perde na trajetória e conseguiu se recuperar (“NOK2”) e o número de vezes que o robô se perdeu completamente ou colidiu sem condições de recuperação (“NOK3”). Se forem somados os percentuais dessas ocorrências nas simulações, os valores encontrados são de 37% e 40% na simulação do SEF Caso A e B respectivamente e

de 37% e 50% do SDF Caso A e B respectivamente. Isso revela que a simulação no caso A para os dois sensores é mais eficiente do que no caso B pelo motivo de o caso A sempre utilizar na sua simulação o sensor lateral do mesmo lado do sensor frontal simulado e nas regras *Fuzzy* considerar a distância “Muito Perto” para o mesmo, e no caso B considera-se o valor do sensor lateral do lado oposto ao sensor frontal simulado considerando na regras *Fuzzy* como “Perto” para o sensor lateral.

É importante observar que no caso A da simulação do sensor SDF as curvas feitas sem colisões do tipo “OK” atingiram valores de 23% de ocorrência na quantidade geral das vezes, enquanto as outras simulações se mantiveram entre 4 e 13%. Esse resultado não deve ser levado em consideração pois se observado no quadro 5.8 esses acertos só ocorrem na curva 1, o que indica que o posicionamento inicial do robô influenciou o resultado dos testes.

#### 5.4 RESULTADOS DOS TESTES COM FALHA DE DOIS SENSORES SIMULTÂNEAMENTE

No caso da falha de dois sensores simultâneos as regras *fuzzy* não são alteradas, ou seja, utiliza-se as mesmas regras usadas para a simulação de cada sensor individualmente.

##### 5.4.1 Resultados em caso de falha das combinações SF-SDL ou SF-SEL

A falha do sensor frontal - SF e do sensor direita lado - SDL ou sensor esquerda lado - SEL, compromete o robô no reconhecimento de obstáculos à frente e na identificação de para qual lado ele deve seguir no caso de uma curva. Os resultados dos testes de falhas com o conjunto SF-SDL ou SF-SEL estão apresentados nos quadros 5.10 e 5.11, respectivamente.

**Quadro 5.10-** Simulação dos sensores SF e SDL utilizando SDF-SEF-SEL

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK1	% NOK1	% NOK2	% NOK3	% X
Curva 1	OK	OK	NOK1	OK	OK1	OK	NOK3	NOK1	NOK1	OK	50%	10%	30%	0%	10%	0%
Curva 2	NOK3	NOK1	OK1	NOK3	NOK3	NOK3	X	OK	NOK1	NOK1	10%	10%	30%	0%	40%	10%
Curva 3	X	NOK3	NOK1	X	X	X	X	OK1	OK	OK	20%	10%	10%	0%	10%	50%

Total Geral	27%	10%	23%	0%	20%	20%
-------------	-----	-----	-----	----	-----	-----

Fonte:Elaboração do autor

**Quadro 5.11-** Simulação dos sensores SF e SEL utilizando SDF-SEF-SDL

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK1	% NOK1	% NOK2	% NOK3	% X
Curva 1	OK	OK	OK	OK	NOK1	OK	OK	OK	OK	OK	90%	0%	10%	0%	0%	0%
Curva 2	NOK3	NOK1	NOK1	NOK1	NOK3	NOK1	NOK1	NOK1	NOK3	OK1	0%	10%	60%	0%	30%	0%
Curva 3	X	NOK1	NOK3	OK	X	NOK3	OK	OK	X	OK	40%	0%	10%	0%	20%	30%
Total Geral											43%	3%	27%	0%	17%	10%

Fonte:Elaboração do autor

Percebe-se que nas duas simulações apresentadas nos quadros 5.10 e 5.11 onde o sensor SF é simulado juntamente com os sensores SDL e SEL, o número de colisões fortes do tipo “NOK1” começam a aparecer, o que não acontecia quando os sensores eram simulados individualmente. Isso ocorre na realização de uma curva seja para direita ou para esquerda devido à falta do sensor SF e da demora da Ponte H L298 em responder esse comando, o que faz como que o robô, no momento de efetuar a curva, inicie o giro muito próximo da lateral da pista, gerando uma colisão forte com a lateral, mas na sequência ele consegue se recuperar. Em decorrência, as formas de colisões se tornam mais graves e as colisões mais leves tem sua ocorrência atenuada nas duas simulações em comparação com a simulação individual dos sensores.

Percebe-se também que o número de colisões fortes ou perdas sem recuperação do tipo “NOK3” se mantém próximo dos resultados do SF simulado separadamente. Isso demonstra que o SF exerce maior influência para este tipo de colisão.

Deve-se observar também que na simulação do conjunto SF-SEL a quantidade de curvas bem sucedidas “OK” tem uma maior ocorrência do que no conjunto SF-SDL sendo de 43% e 27% respectivamente. Essa informação não tem muita relevância, pois, como informado anteriormente, a maior parte das ações bem sucedidas estão na primeira curva e isso sofre grande influência da posição inicial do robô.

#### 5.4.2 - Resultados em caso de falha das combinações SEF-SDL, SEF-SEL, SDF-SEL, SDF-SDL

A falha do conjunto de sensores SEF-SDL, SEF-SEL, SDF-SEL, SDF-SDL faz com que o robô tenha prejudicada a sua capacidade de correção do seu alinhamento na pista através dos sensores SEF e SDF e sua capacidade de identificar para qual lado deve girar no caso de uma curva através dos sensores SEL e SDL.

Conforme os resultados apresentados nos quadros 5.12, 5.13, 5.14, 5.15, toda vez que um dos sensores frontais falha, aumenta também o número de colisões graves sem recuperação do tipo “NOK3”. Já nos casos em que falham os sensores laterais do lado oposto aos sensores frontais, ou seja, na combinação SEF-SDL houve falha em 23% das vezes e na combinação SDF-SEL houve falha em 37%. Isso ocorre pelo mesmo motivo das simulações individuais dos sensores SDF e SEF apresentados no item 5.14. Também por esse motivo, a soma das colisões fortes ou perdas com recuperação do tipo “NOK1” e “NOK2” são menores quando falham os sensores do mesmo lado, sendo assim tem-se 33% para as combinações SEF-SEL e SDF-SDL e de 43% e 46% para as combinações SEF-SDL e SDF-SEL.

**Quadro 5.12-** Simulação dos sensores SEF e SDL utilizando SDF-SF-SEL.

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK 1	% NOK 1	% NOK 2	% NOK 3	% X
Curva 1	NOK 2	NOK 3	OK	NOK 2	NOK 2	OK	NOK 2	NOK 2	NOK 3	NOK 2	20 %	0%	0%	60%	20%	0%
Curva 2	NOK 1	X	NOK 1	NOK 3	NOK 3	NOK 3	OK	NOK 2	X	NOK 3	10 %	0%	20%	10%	40%	20 %
Curva 3	NOK 1	X	NOK 3	X	X	X	OK	OK	X	X	20 %	0%	10%	0%	10%	60 %
Total Geral											17 %	0%	10%	23%	23%	27 %

Fonte:Elaboração do autor

**Quadro 5.13 -** Simulação dos sensores SEF e SEL utilizando SDF-SF-SDL

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK1	% NOK1	% NOK2	% NOK3	% X
Curva 1	NOK1	NOK1	OK	NOK2	OK	OK	NOK2	OK	NOK1	NOK2	40%	0%	30%	30%	0%	0%

Curva 2	NOK3	OK	NOK1	NOK3	NOK2	OK	NOK3	NOK3	NOK3	NOK2	20%	0%	10%	20%	50%	0%
Curva 3	X	NOK1	OK	X	NOK1	NOK2	X	X	X	NOK1	10%	0%	30%	10%	0%	50%
Total Geral											23%	0%	23%	20%	17%	17%

Fonte:Elaboração do autor

**Quadro 5.14-** Simulação dos sensores SDF e SEL utilizando SEF-SF-SDL

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK1	% NOK1	% NOK2	% NOK3	% X
Curva 1	NOK2	NOK1	NOK2	NOK2	NOK1	NOK2	NOK2	OK	NOK3	NOK1	10%	0%	30%	50%	10%	0%
Curva 2	NOK3	NOK3	NOK3	NOK3	NOK2	NOK2	OK	NOK3	X	NOK3	10%	0%	0%	20%	60%	10%
Curva 3	X	X	X	X	NOK3	NOK3	NOK3	NOK2	X	X	0%	0%	0%	0%	40%	60%
Total Geral											7%	0%	10%	23%	37%	23%

Fonte:Elaboração do autor

**Quadro 5.15-** Simulação dos sensores SDF e SDL utilizando SEF-SF-SEL

	Testes										Resultados					
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	% OK	% OK1	% NOK1	% NOK2	% NOK3	% X
Curva 1	OK	NOK2	NOK2	OK1	OK1	OK	NOK2	OK1	OK1	OK1	20%	50%	0%	30%	0%	0%
Curva 2	NOK3	NOK3	NOK1	NOK1	NOK2	NOK1	NOK2	NOK3	NOK2	NOK2	0%	0%	30%	40%	30%	0%
Curva 3	X	X	OK	OK	NOK1	NOK3	NOK1	X	NOK1	NOK1	20%	0%	40%	0%	10%	30%
Total Geral											13,5%	17%	23%	23%	13,5%	10%

Fonte:Elaboração do autor

Analisando os quadros é possível perceber, de um modo geral, que a maior parte das perdas do robô com recuperação ou não, se dá basicamente a depender da forma como o robô está posicionado ao entrar na curva, ou seja, caso o robô inicie o procedimento da curva quando está mais próximo da lateral externa, as curvas serão realizadas com maior eficiência devido à melhor referência de medição para os sensores laterais SDL, SEL, SDF e SEF. Entretanto, caso o procedimento seja efetuado mais próximo da lateral interna da curva, aumenta a possibilidade do robô ter maiores perdas de trajetos e colisões irreversíveis.

Neste capítulo foram apresentados os resultados da implementação do sistema nebuloso tolerante a falhas para a simulação de falhas de todos os sensores de forma individual ou de até 2 sensores simultâneos. Conforme dados apresentados, foi possível perceber que o sistema, em alguns casos, consegue reagir de forma eficiente, porém, em outros casos, muitas vezes existem colisões graves ou o robô se perde em sua trajetória.

Também se concluiu que os sensores frontais são mais importantes para o funcionamento do robô, pois quando esses falham o robô tem mais dificuldade para completar a sua missão.

Constatou-se também que hardware exerce um papel muito importante nesse processo, pois a demora da ponte HL298 em enviar os comandos, a imprecisão das caixas de redução e da leitura dos sensores a longa distância interferem nos resultados devido à sua ocorrência ser intermitente.

## 6LIMITAÇÕES DA PLATAFORMA

A plataforma para validação dos resultados apresentada nesta dissertação possui algumas características físicas que acabaram por trazer algumas limitações para o desempenho do robô.

Os sensores utilizados são do fabricante SHARP modelo GP2Y0A21YK. Segundo o fabricante, sua capacidade de medir distâncias está entre 10 cm e 80 cm, porém, durante a sua utilização percebeu-se que o sensor se comportava de forma diferente a cada vez que se aproximava ou se afastava do objeto a ser medido. Para isso foi montada uma bateria de testes para validar o real potencial de medição dos sensores e observar se o sinal emitido por um sensor poderia influenciar a medição de outro sensor. Sendo assim foi traçada uma escala de medição através de uma régua com o objetivo de saber se na referida distância o sensor retorna o seu valor correspondente.

Isso foi feito de forma individual para cada sensor, ou seja, quando a medição era feita para um determinado sensor os demais estavam desligados. Com isso constatou-se que nas medições entre 10cm e 30 cm o erro da medição era nulo, ou seja, o valor medido era igual ao valor real, porém quando se pretendia medir uma distância entre 30cm e 60 cm e 60cm a 80 cm a resposta do sensor oscilava em torno de 16% a 48% de erro respectivamente. Isso ocorreu da mesma forma em todos os sensores e atribui-se essa imprecisão aos ruídos causados pela iluminação ambiente. Quanto mais distante o sensor estava do objeto a ser medido maior foi a sua imprecisão, ou seja, a intensidade de reflexão da luz infravermelha do emissor será menor e com isso a iluminação ambiente introduzia uma maior quantidade de ruído no receptor de luz. Entranto, nas distâncias entre 10 cm e 30 cm, devido à proximidade do sensor com o objeto, existe uma maior intensidade de reflexão o que minimiza/anula o efeito do ruído tornando a medição mais precisa.

Também foi observado se um sensor teria condições de interferir na medição do outro, pois alguns fornecedores de produtos, como TinySine<sup>1</sup> e Sparkfun<sup>2</sup>, informam que existe um ângulo de 40° na zona de medição do sensor e isso poderia interferir nas leituras. Com isso foi consultado o manual do fabricante (Anexo B), mas

---

<sup>1</sup><[http://www.tinyosshop.com/index.php?route=product/product&product\\_id=268](http://www.tinyosshop.com/index.php?route=product/product&product_id=268)>

<sup>2</sup><<https://www.sparkfun.com/products/8958>>

nele não foi encontrado nenhuma informação desse tipo, ou seja, para o fabricante não existe nenhum ângulo que um sensor possa interferir no outro desde que esse não esteja posicionado a sua frente.

Foi então realizado um segundo teste para identificar se existia a possibilidade do emissor de um sensor interferir no receptor do outro. Para isso foi realizado o mesmo teste informado acima, porém com todos os sensores ligados e seus valores foram comparados com os resultados dos sensores quando testados individualmente.

Foi constatado que os valores eram idênticos e, portanto, não existiu interferência entre os sensores, pois os resultados se comportaram da mesma forma. Diante dessas informações foi considerado no trabalho que não existe limitação de ângulo para a medição do sensor.

Em relação à plataforma mecânica, ela é formada por dois motores de corrente contínua para cada roda conectados a uma caixa de redução conforme apresentado no item 3.1. Devido a ser uma plataforma de prototipagem não existe uma boa precisão nos acionamentos das rodas, ou seja, mesmo quando ligados ao mesmo tempo não necessariamente as rodas iniciam o movimento de forma simultânea, pois existe o fator das folgas na caixa de redução que são proveniente da plataforma e que acabam causando um atraso tanto no acionamento quanto na parada do movimento.

Esse fenômeno foi observado quando se aciona a plataforma para frente e a mesma inicia o movimento com um leve desalinhamento para a direita ou para esquerda de forma intermitente. Neste caso não é possível fazer nenhum tipo de correção, pois a caixa de redução é lacrada pelo fabricante. Sendo assim é obrigatória a convivência com essa circunstância.

No que concerne à plataforma eletrônica, esta é composta pela placa Arduino Uno e pela Ponte H L298 apresentados no item 3.2. Dentro dessa plataforma durante o seu uso foi constatado algumas inconsistências em seu funcionamento que causam reações inadequadas no funcionamento do robô, ou seja, foi constatado durante os testes que a Ponte H L298 prioriza a saída do motor que está conectado à saída "A", por isso ele recebe a energização antes do motor que está conectado na saída "B".



Sendo assim, quando a plataforma recebe o comando de “ir para frente” o motor conectado na saída “A” parte antes do motor na saída “B”, isso causa um desalinhamento da plataforma e pode causar instabilidade no deslocamento do robô, pois, assim que a plataforma se desalinha o algoritmo de controle tenta corrigi-la através das informações dos sensores. Porém, como o processo de correção se dá com o giro das rodas no sentido oposto e mesmo assim continua existindo a demora da movimentação das rodas o algoritmo tem que efetuar correções todo o tempo o que pode em algumas situações levar o robô a colidir com obstáculos.

Consultando o manual do fabricante, Anexo A onde foi constatado que isso ocorre devido ao tempo em que o chip L298 leva para acionar as saídas dos motores, além disso foi identificada a existência da diferença do tempo de acionamento dos dois motores, as saídas têm uma demora entre o recebimento da informação para acionar ou parar o motor e sua execução. Em outras palavras, quando o robô está em movimento e é enviado o comando “PARE” ou “CORRIGIR”, ele demora um tempo para responder e por isso com isso são causadas colisões constantes a depender do seu posicionamento na pista.

## 7 CONSIDERAÇÕES FINAIS

Nesta dissertação apresentou-se um sistema de tolerância a falhas de sensores em um robô autônomo baseado na lógica *Fuzzy*. O sistema tolerante a falhas desenvolvido realiza a simulação de sensores defeituosos com base nas informações dos demais sensores não redundantes que estão com seu funcionamento satisfatório. Desta forma, o objetivo do sistema é identificar a falha e automaticamente oferecer uma resposta para a falha do sensor correspondente.

Para atingir os objetivos propostos foi desenvolvido um robô para testes e uma pista para validação dos resultados. Neste processo foram estudados os módulos que compõem o robô tanto na parte mecânica como na eletrônica constatando-se que o robô possui algumas limitações e que essas limitações poderiam, em certas ocasiões, comprometer o correto funcionamento do sistema. Verificou-se também que os valores informados pelo fabricante no range de medição dos sensores apresentam uma taxa de erro a partir da distância de 30 cm e esse erro aumenta conforme aumenta a distância. Além disso, mecanicamente existe a imprecisão do sistema mecânico das caixas de redução dos motores e da plataforma de acionamento dos mesmos.

Quando o sistema opera com todos os sensores funcionando corretamente, ele consegue atingir o seu objetivo em todas às vezes, porém com algumas colisões devido às limitações da plataforma, essas colisões são leves e pouco atrapalham a missão do robô. Entretanto, quando os sensores entram em falha tem-se vários comportamentos a depender do sensor simulado, por exemplo, na perda dos sensores laterais SDL e SEL o robô permanece com um desempenho de no mínimo 70% e 65% de acerto, respectivamente, e mesmo que o robô sofra algumas colisões leves, ele consegue cumprir as três etapas da pista sequencialmente.

Já quando se trabalha com o sensor SF, o robô consegue acertar 70% das decisões para o seu deslocamento, porém em apenas 30% das tentativas o robô consegue completar a pista, isso ocorre devido ao aumento do número de colisões frontais o que muitas vezes leva o robô à região de zona morta dos sensores e,consequentemente, a perda do trajeto.

No momento em que simulamos o sensor SDF e SEF, que têm a função de efetuar o alinhamento do robô na pista foi observado que, em média, em 45% das

tentativas o robô conseguiu cumprir as três curvas do circuito. Com isso podemos concluir que a falha dos sensores frontais são mais graves para o robô.

Quando tratamos da falha de dois sensores simultâneos devemos primeiro observar as possibilidades presentes nas tabelas do FMEA no Capítulo 3. Sendo assim, quando ocorre a falha do sensor SF-SDL ou SF-SEL o robô consegue completar 45% das tentativas 100% do circuito.

Já quando falham as combinações SEF-SDL, SEF-SEL, SDF-SEL, SDF-SDL, foi percebido que quando na combinação o sensor frontal SEF ou SEF falham juntamente com o sensor lateral oposto, ou seja, quando falham os sensores SEF-SDL ou SDF-SEL,consequentemente aumenta a quantidade de falhas. Porém, nesses casos tem-se que, em média, 37,5% das tentativas foram bem sucedidas em percorrer a pista completa.

Enfim, pode-se concluir que a depender do sensor em falha é possível ter um aumento ou redução no nível de acerto. Os sensores mais eficientes na simulação são os sensores laterais SDL e SEL; os outros tiveram mais dificuldade para completar o circuito. Contudo, se for levado em consideração os acertos parciais de todos os sensores, temos um melhor desempenho com relação ao número de acertos em completar o circuito. Um fator importante para a constatação desses resultados são os módulos presentes na plataforma e sua imprecisão, ou seja, caso fossem substituídos os motores de corrente contínua e suas caixas de redução juntamente com a ponte H L298 por servo motores, teríamos uma maior precisão no deslocamento do robô e com isso resultado melhores. Além disso, se poderia utilizar sensores com ranges de operação menores, tal como 4 cm a 20 cm ou 0 cm a 10cm.Issso reduziria a chance do robô entrar na zona morta dos sensores e,consequentemente, se perder menos durante o percurso.

Diante do exposto, compreende-se que a utilização de tolerância a falhas em sistemas robóticos com intenção de eliminar sensores redundantes é eficiente e pode trazer grandes vantagens para sistemas autônomos, pois possibilita que mesmo em caso de falhas o robô consiga completar sua missão ou mesmo retornar à sua base para reparos. Entretanto, deve ser levado em consideração a necessidade de um hardware que consiga ter robustez suficiente para lidar com as falhas que possam ocorrer.

O sistema e a metodologia proposta nessa dissertação podem ser utilizados para melhorar o desempenho de máquinas e equipamentos industriais críticos para

o processo, ou seja, pode-se utilizar nos setores aeronáuticos para o cruzamento de informações dos diferentes sensores presentes na aeronaves e assim aumentar a robustez do sistemas redundantes. Por exemplo, em caso de falha, a redundância dos sensores da aeronave permitirá que a mesma continua sua missão tendo por base os sinais dos demais sensores sem que seja gerada uma pane. Outro caso seria para uso de caldeiras industriais ou compressores de ar, pois com base nas informações dos sensores que não apresentem falhas pode-se desligar emergencialmente o sistema e colocá-lo em um modo seguro antes que o mesmo entre em colapso e cause um acidente.

Por fim, apresentam-se sugestões para o aperfeiçoamento do trabalho na linha de pesquisa, com base nos resultados apresentados:

- Implementar o algoritmo para o cruzamento de vários tipos de sensores diferentes tais como sensores de navegação e de análise do ambiente.
- Ampliar o sistema para que o mesmo consiga tolerar a falha de mais sensores simultâneos.
- Trabalhar com aplicações do modelo em equipamentos industriais.
- Adicionar um sistema de diagnóstico e detecção de falhas mais eficiente de modo que avalie a gravidade da falha e, se possível, considere a parte da informação útil gerada pelo sensor para aumentar a precisão do algoritmo.

## REFERÊNCIAS

AMEDI, A.; MERABET, L.B.; BERMPOHL, F.; PASCUAL-LEONE, A. The occipital cortex in the blind. Lessons about plasticity and vision. *Current Directions in Psychological Science*, Washington, DC, v. 14, n. 6, p. 306–311, 2005.

ARDUINO: página institucional. Versão em inglês. [s.l.]. Disponível em: <<https://www.arduino.cc/en/Main/ArduinoBoardUno>>. Acesso em: 05 jan. 2017.

BUENO, Elaine Inacio. **Utilização de redes neurais artificiais na monitoração e detecção de falhas em sensores do reator IEA-R1**. 2006. Tese de Doutorado. Universidade de São Paulo.

CALDAS, Wilton Speziali. **Tolerância a falhas adaptativa para robôs móveis com arquitetura híbrida**. 2004. Tese, Doutorado em Ciência da Computação. Universidade Federal de Minas Gerais, Belo Horizonte, 2004.

CARDOZO, Daisy Isabel Kang. **Projeto de controle de alocação tolerante a faltas para um veículo autônomo subaquático utilizando lógica nebulosa**. 2013. Dissertação, Mestrado em Engenharia de Sistemas Dinâmicos e Energéticos. Universidade Estadual do Oeste do Paraná, Foz do Iguaçu, 2013.

CHIOZZA, Maria Laura; PONZETTI, Clemente. FMEA: A model for reducing medical erros. *Clinica Chimica Acta*. Vol. 404, Issue 1, 6 June 2009, Pages 75–78. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0009898109001466>>. Acesso em: 15 jan. 2017.

CRESTANI, Didier; GODARY-DEJEAN, Karen; LAPIERRE, Lionel. Enhancing fault tolerance of autonomous mobile robots. *Robotics and Autonomous Systems*, v. 68, p. 140-155, 2015.

FERNANDES, Raphaela Galhardo. **Deteção e isolamento de falhas em sistemas dinâmicos baseados em redes neurais**. 2007. Dissertação, Mestrado em Engenharia Elétrica. Universidade Federal do Rio Grande do Norte, Natal, 2007.

HAFEZ, Ahmed Taimour; KAMEL, Mohamed A. Fault-tolerant control for cooperative unmanned aerial vehicles formation via Fuzzy logic. In: **2016 International Conference on Unmanned Aircraft Systems (ICUAS)**. IEEE, 2016. p. 1261-1266.

MALAR, R. Suja Mani. Fault Detection and Tolerance on Robot Manipulators Locked Joint Failure Using Anfis. *International Journal of Applied Engineering Research*, v. 11, n. 2, p. 887-891, 2016.

MARTINS, Maurício Armada. **Desenvolvimento de funções de lógica difusa para PLC**. 2014. Dissertação, mestrado em Mestrado em Engenharia Electrotécnica e de Computadores. Instituto Superior de Engenharia do Porto, Porto (Portugal), 2014.

MEGERIAN, S.; KOUSHANFAR, F.; VELTRI, G. Qu, G.; POTKONJAK, M., Exposure in wireless sensor networks: theory and practical solutions, *Wireless Networks*, Kluwer Academic Publishers, ISSN 1022-0038, vol. 8, no. 5, 2002, pp 443-454.

PEREIRA, Marluce R.; AMORIM, C. L.; CASTRO, M. C. S. Tutorial sobre redes de sensores. **Cadernos do IME**: Série Informática, v. 14, 2003.

SIMÕES, Marcelo Godoy; SHAW, Ian S. **Controle e modelagem Fuzzy**. São Paulo: Blucher: FAPESP, 2007.

WEBER, Taisy Silva. Tolerância a falhas: conceitos e exemplos. **Apostila do Programa de Pós-Graduação–Instituto de Informática-UFRGS**. Porto Alegre, 2003.

WEBTRONICO. Disponível em: <<http://www.webtronico.com/modulo-driver-motor-dc-motor-de-passo-l298n.html>>. Acesso em: 07 jan. 2017.

ZADEH, Lotfi A. Fuzzy sets. **Information and control**, v. 8, n. 3, p. 338-353, 1965.

ZHU, Yixian et al. An intelligent fault-tolerant strategy for AUV integrated navigation systems. In: **2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)**. IEEE, 2015. p. 269-274.

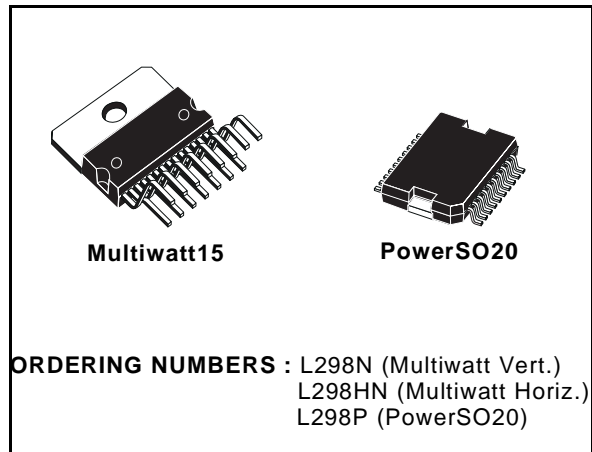
**ANEXO A – *Datasheet* da Ponte H L298**

## DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

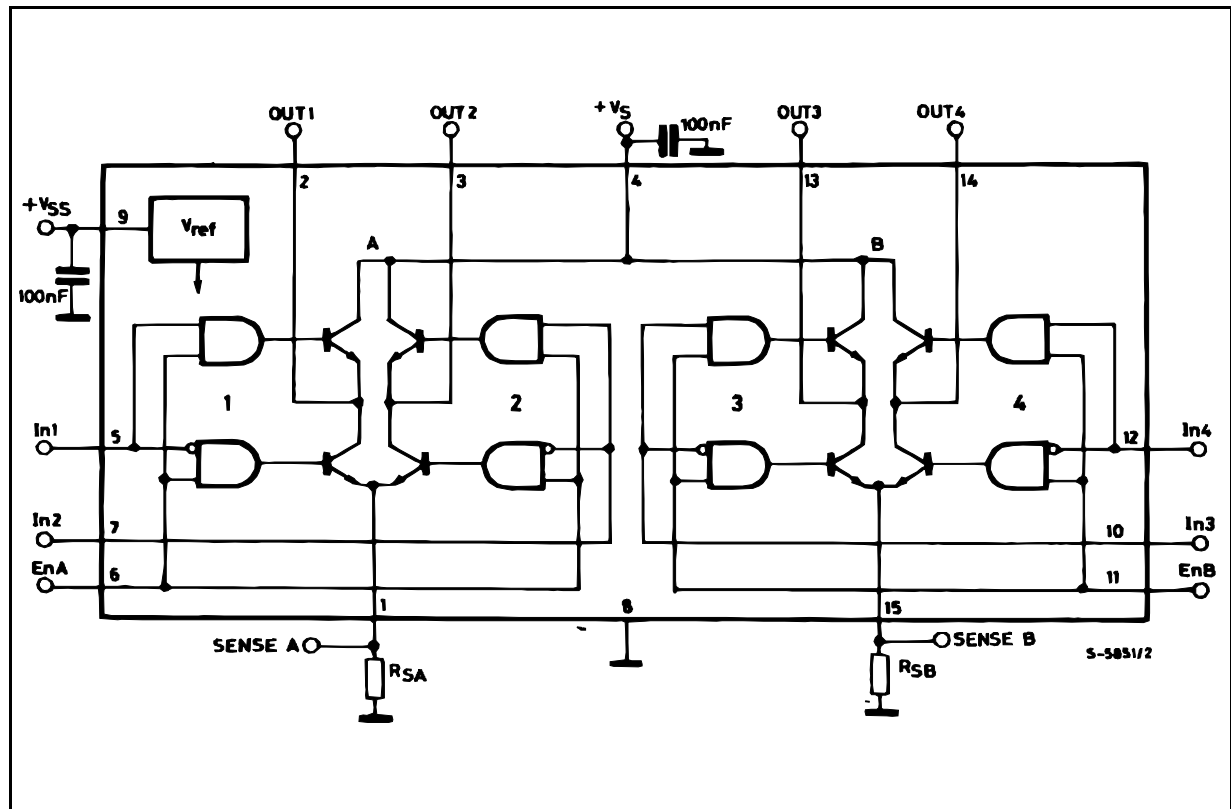
### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

### BLOCK DIAGRAM

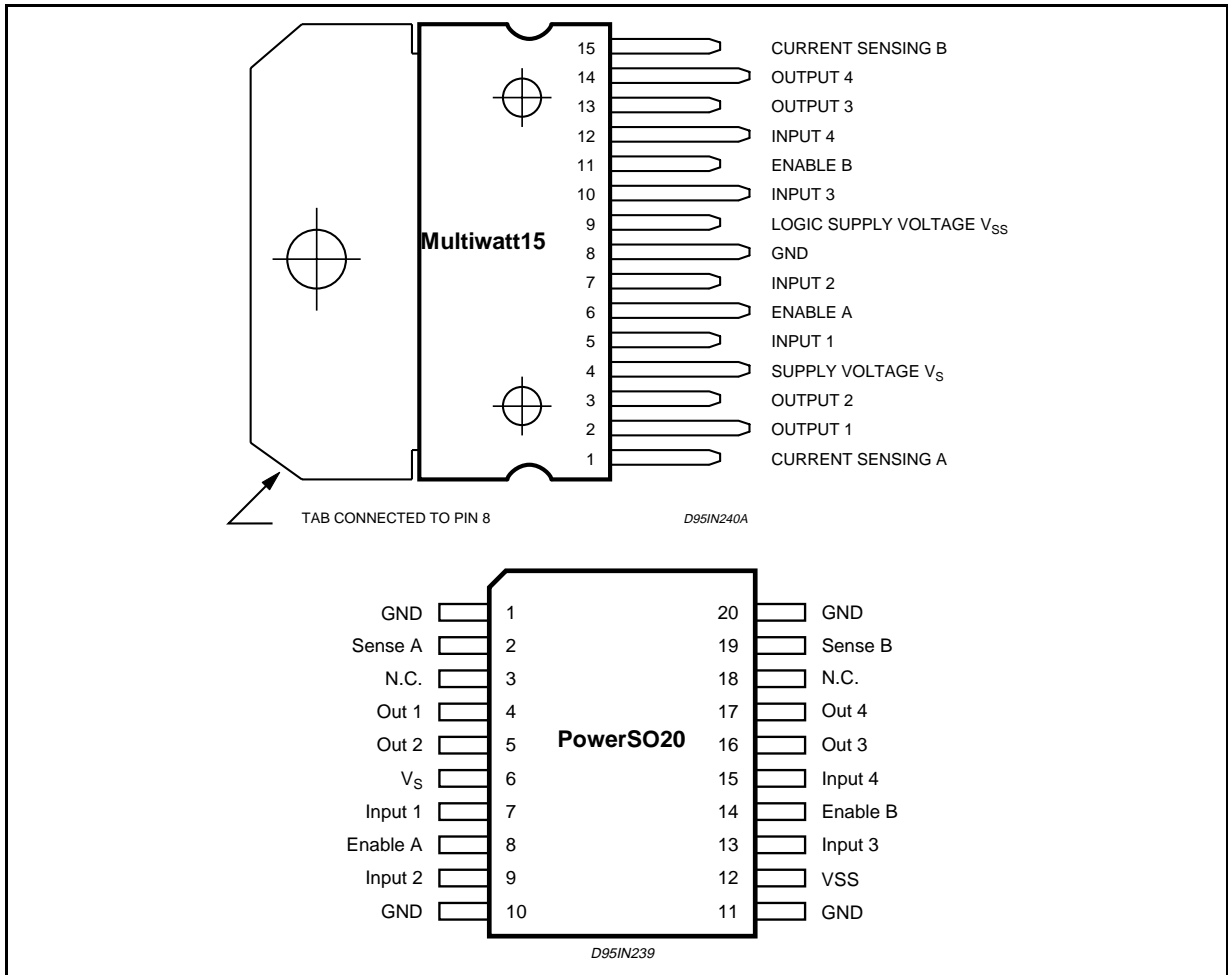




**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value	Unit
$V_S$	Power Supply	50	V
$V_{SS}$	Logic Supply Voltage	7	V
$V_I, V_{En}$	Input and Enable Voltage	-0.3 to 7	V
$I_O$	Peak Output Current (each Channel)		
	- Non Repetitive ( $t = 100\mu s$ )	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$ )	2.5	A
	-DC Operation	2	A
$V_{sens}$	Sensing Voltage	-1 to 2.3	V
$P_{tot}$	Total Power Dissipation ( $T_{case} = 75^\circ C$ )	25	W
$T_{op}$	Junction Operating Temperature	-25 to 130	$^\circ C$
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	$^\circ C$

**PIN CONNECTIONS (top view)**



**THERMAL DATA**

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th j-case}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(\*) Mounted on aluminum substrate



**PIN FUNCTIONS** (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V <sub>s</sub>	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V <sub>SS</sub>	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

**ELECTRICAL CHARACTERISTICS** (V<sub>S</sub> = 42V; V<sub>SS</sub> = 5V, T<sub>j</sub> = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V <sub>S</sub>	Supply Voltage (pin 4)	Operative Condition	V <sub>IH</sub> +2.5		46	V
V <sub>SS</sub>	Logic Supply Voltage (pin 9)		4.5	5	7	V
I <sub>S</sub>	Quiescent Supply Current (pin 4)	V <sub>en</sub> = H; I <sub>L</sub> = 0 V <sub>i</sub> = L V <sub>i</sub> = H		13 50	22 70	mA mA
		V <sub>en</sub> = L V <sub>i</sub> = X			4	mA
I <sub>SS</sub>	Quiescent Current from V <sub>SS</sub> (pin 9)	V <sub>en</sub> = H; I <sub>L</sub> = 0 V <sub>i</sub> = L V <sub>i</sub> = H		24 7	36 12	mA mA
		V <sub>en</sub> = L V <sub>i</sub> = X			6	mA
V <sub>iL</sub>	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V <sub>iH</sub>	Input High Voltage (pins 5, 7, 10, 12)		2.3		V <sub>SS</sub>	V
I <sub>iL</sub>	Low Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = L			–10	μA
I <sub>iH</sub>	High Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = H ≤ V <sub>SS</sub> –0.6V		30	100	μA
V <sub>en</sub> = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V <sub>en</sub> = H	Enable High Voltage (pins 6, 11)		2.3		V <sub>SS</sub>	V
I <sub>en</sub> = L	Low Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = L			–10	μA
I <sub>en</sub> = H	High Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = H ≤ V <sub>SS</sub> –0.6V		30	100	μA
V <sub>CEsat</sub> (H)	Source Saturation Voltage	I <sub>L</sub> = 1A I <sub>L</sub> = 2A	0.95	1.35 2	1.7 2.7	V V
V <sub>CEsat</sub> (L)	Sink Saturation Voltage	I <sub>L</sub> = 1A (5) I <sub>L</sub> = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V <sub>CEsat</sub>	Total Drop	I <sub>L</sub> = 1A (5) I <sub>L</sub> = 2A (5)	1.80		3.2 4.9	V V
V <sub>sens</sub>	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T <sub>1</sub> (V <sub>i</sub> )	Source Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (2); (4)		1.5		μs
T <sub>2</sub> (V <sub>i</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		0.2		μs
T <sub>3</sub> (V <sub>i</sub> )	Source Current Turn-on Delay	0.5 V <sub>i</sub> to 0.1 I <sub>L</sub> (2); (4)		2		μs
T <sub>4</sub> (V <sub>i</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.7		μs
T <sub>5</sub> (V <sub>i</sub> )	Sink Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		0.7		μs
T <sub>6</sub> (V <sub>i</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.25		μs
T <sub>7</sub> (V <sub>i</sub> )	Sink Current Turn-on Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		1.6		μs
T <sub>8</sub> (V <sub>i</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.2		μs
f <sub>c</sub> (V <sub>i</sub> )	Commutation Frequency	I <sub>L</sub> = 2A		25	40	KHz
T <sub>1</sub> (V <sub>en</sub> )	Source Current Turn-off Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (2); (4)		3		μs
T <sub>2</sub> (V <sub>en</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		1		μs
T <sub>3</sub> (V <sub>en</sub> )	Source Current Turn-on Delay	0.5 V <sub>en</sub> to 0.1 I <sub>L</sub> (2); (4)		0.3		μs
T <sub>4</sub> (V <sub>en</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.4		μs
T <sub>5</sub> (V <sub>en</sub> )	Sink Current Turn-off Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (3); (4)		2.2		μs
T <sub>6</sub> (V <sub>en</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.35		μs
T <sub>7</sub> (V <sub>en</sub> )	Sink Current Turn-on Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (3); (4)		0.25		μs
T <sub>8</sub> (V <sub>en</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V<sub>sens</sub> min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

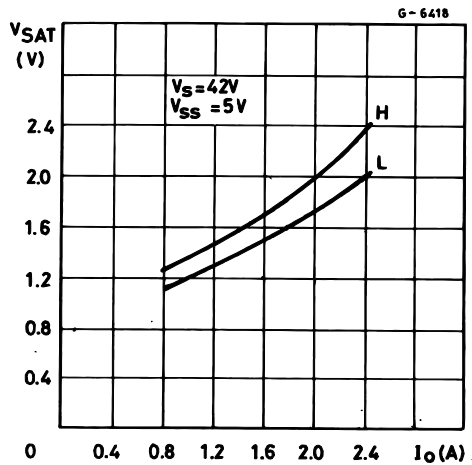
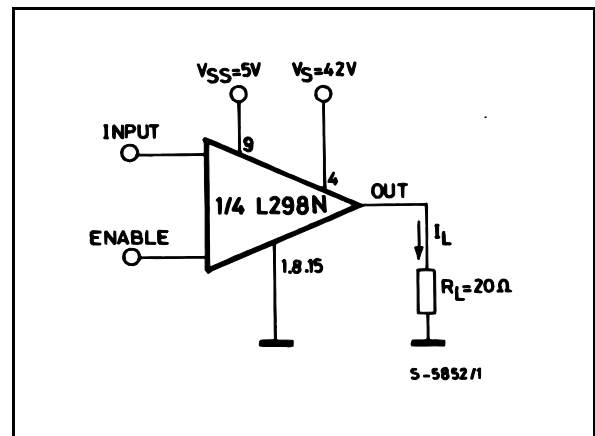
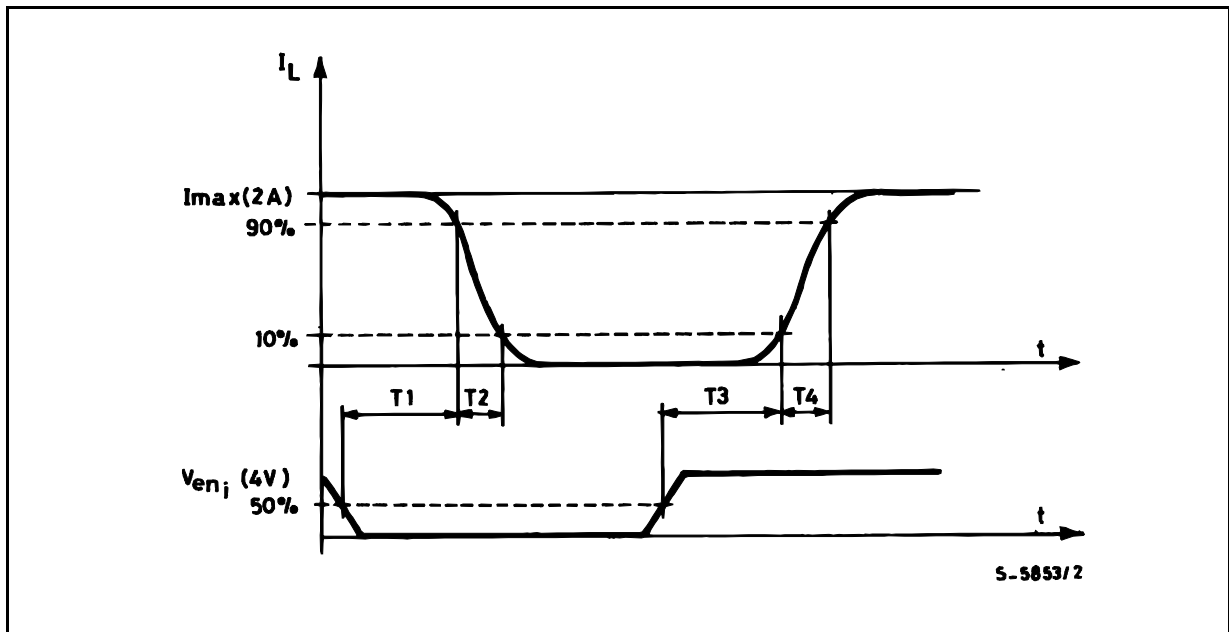


Figure 2 : Switching Times Test Circuits.

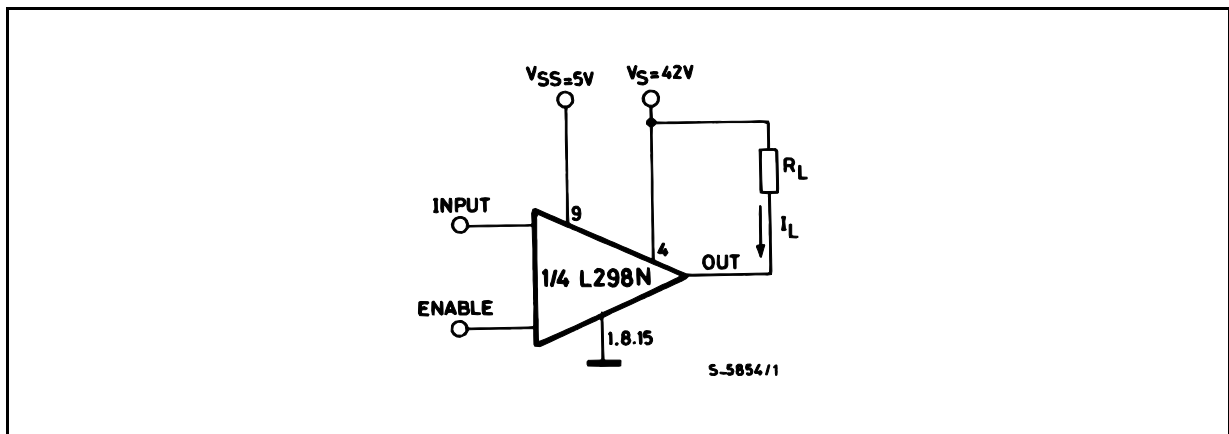


Note : For INPUT Switching, set EN = H  
 For ENABLE Switching, set IN = H

**Figure 3 :** Source Current Delay Times vs. Input or Enable Switching.



**Figure 4 :** Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H  
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

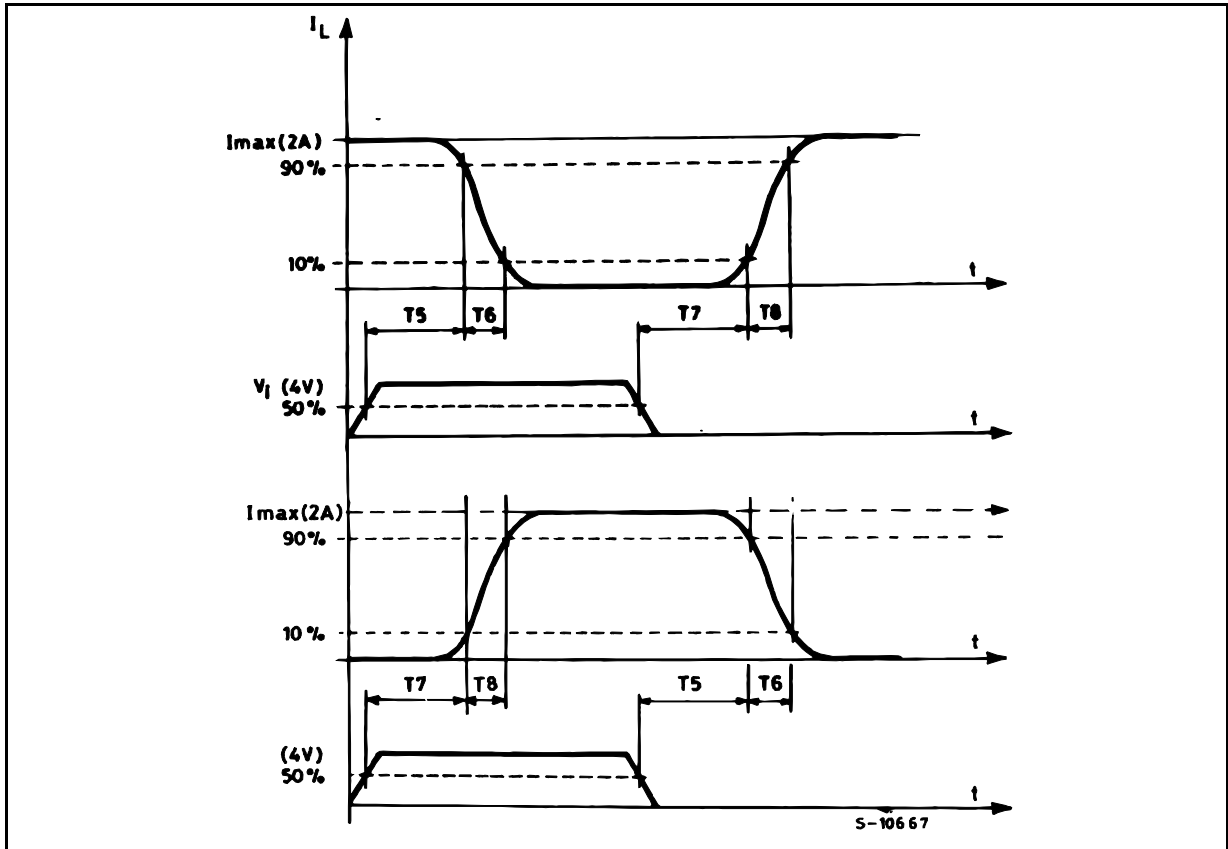
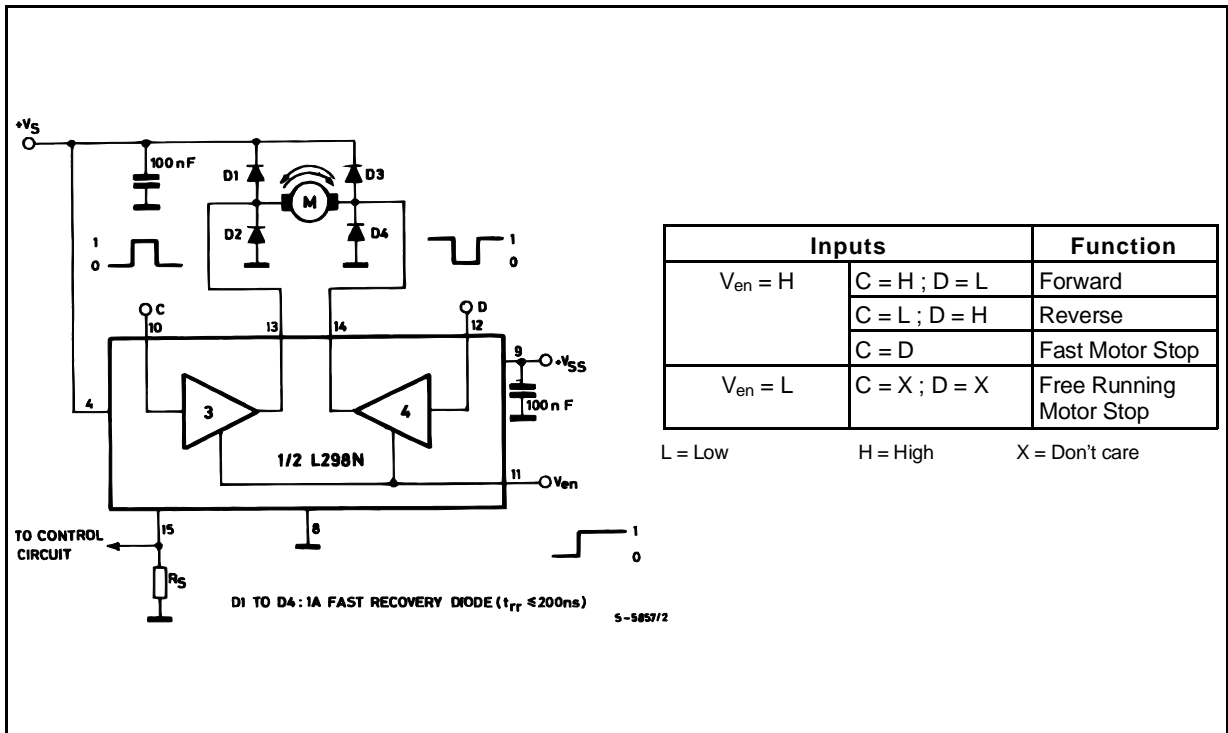
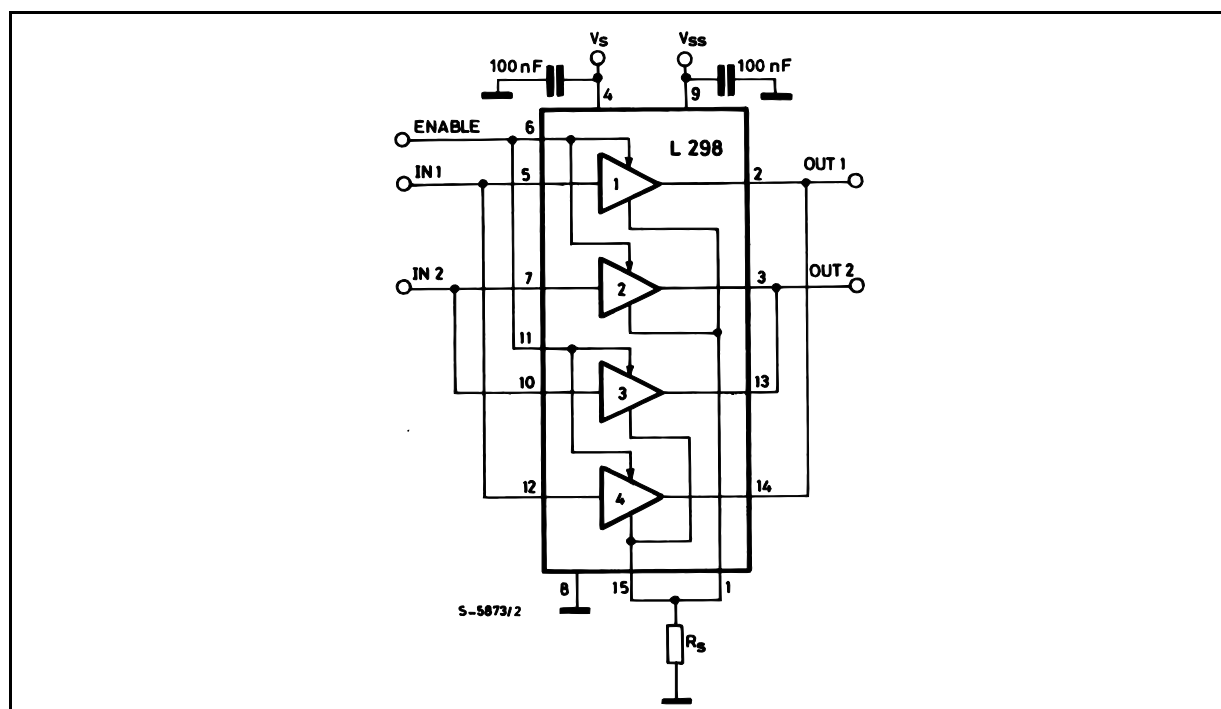


Figure 6 : Bidirectional DC Motor Control.



**Figure 7 :** For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



## APPLICATION INFORMATION (Refer to the block diagram)

### 1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor ( $R_{SA}$  ;  $R_{SB}$ .) allows to detect the intensity of this current.

### 1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

## 2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of Vs that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

## 3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ( $t_{tr} \leq 200$  nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Schottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

**Figure 8 :** Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

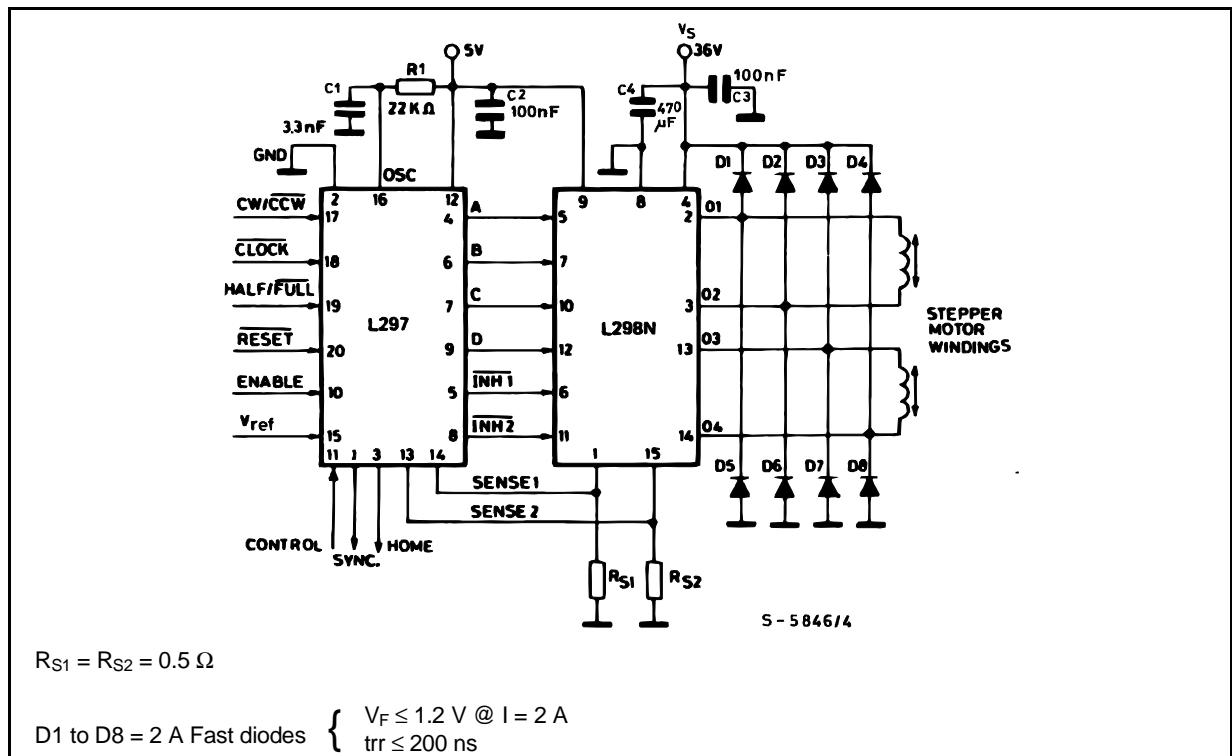


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

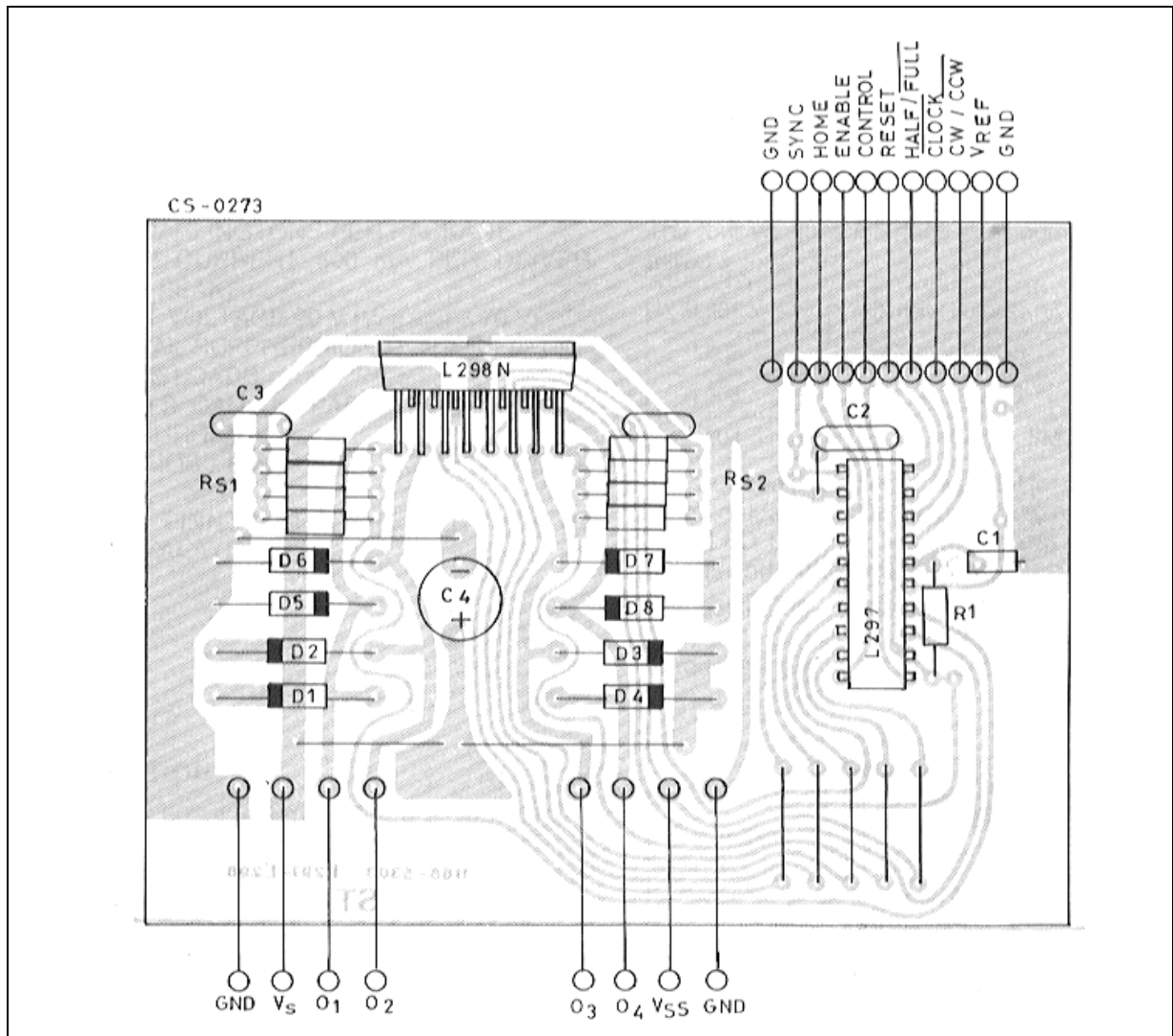
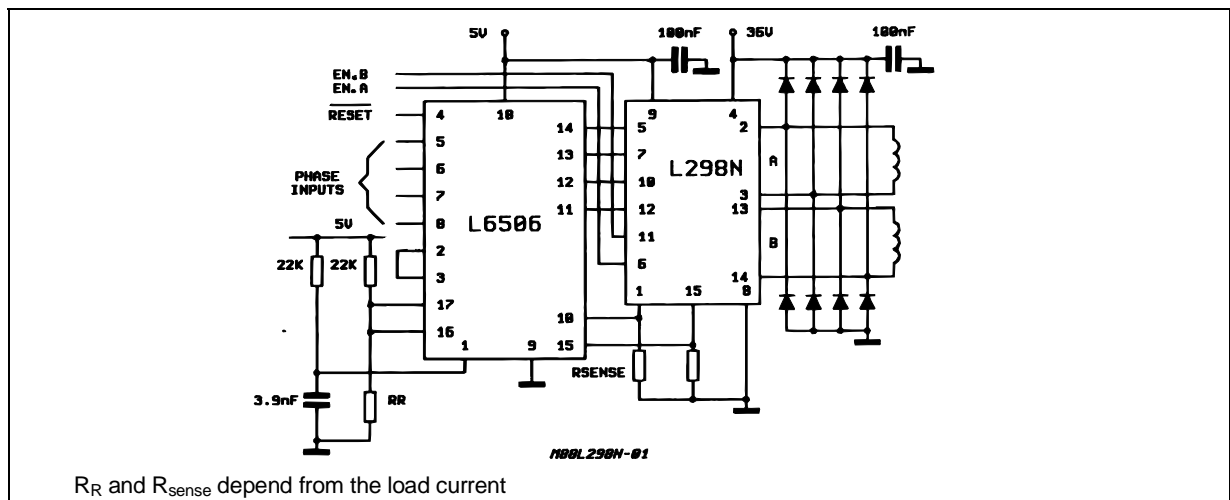


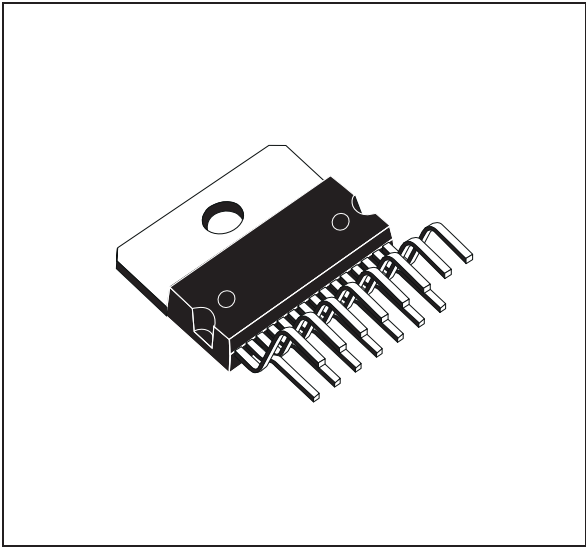
Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.



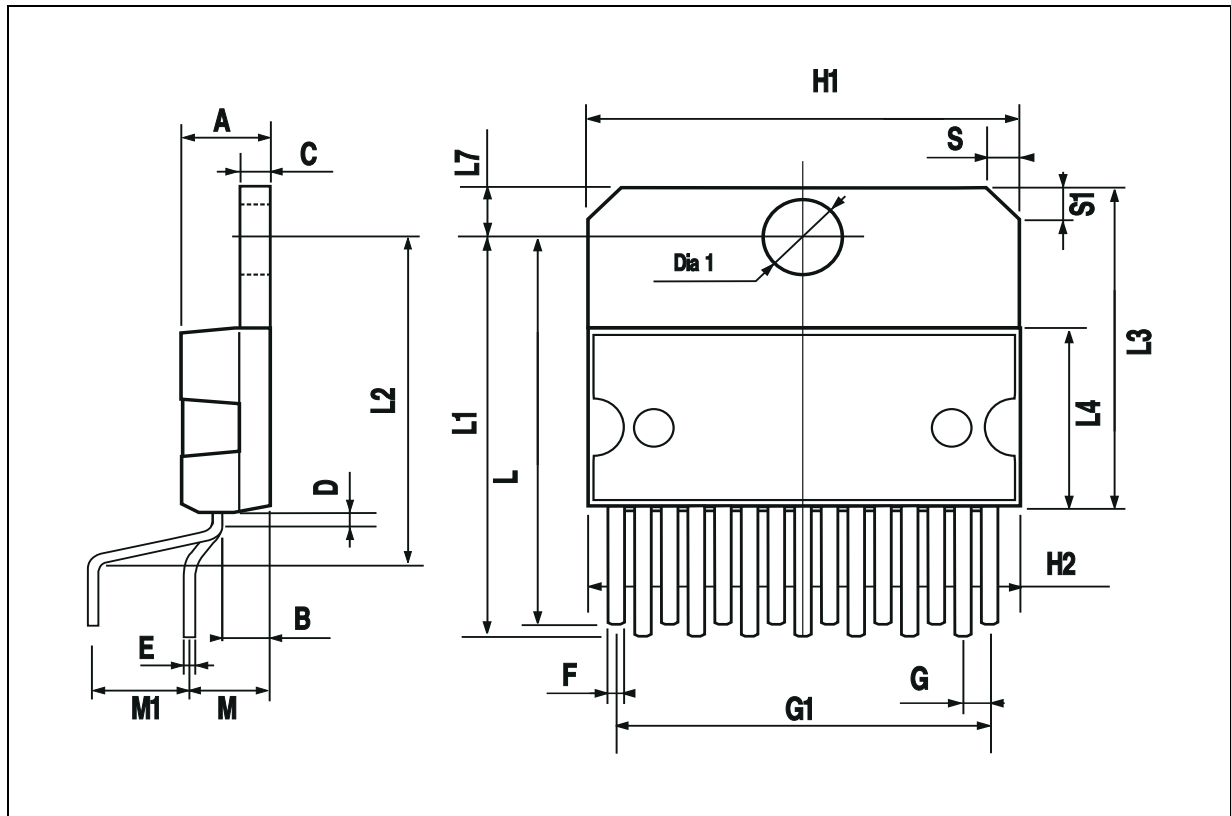


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

**OUTLINE AND MECHANICAL DATA**

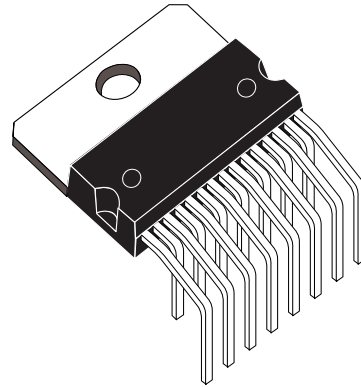


**Multiwatt15 V**

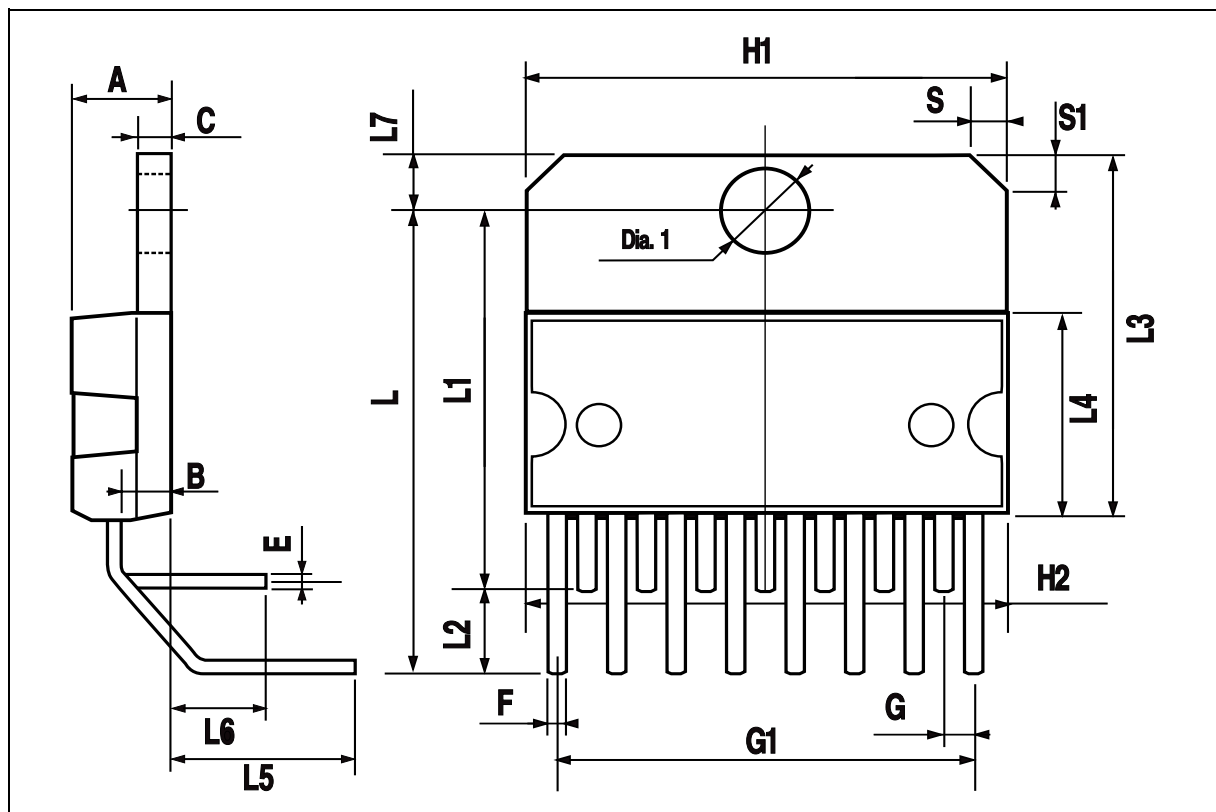


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

## OUTLINE AND MECHANICAL DATA



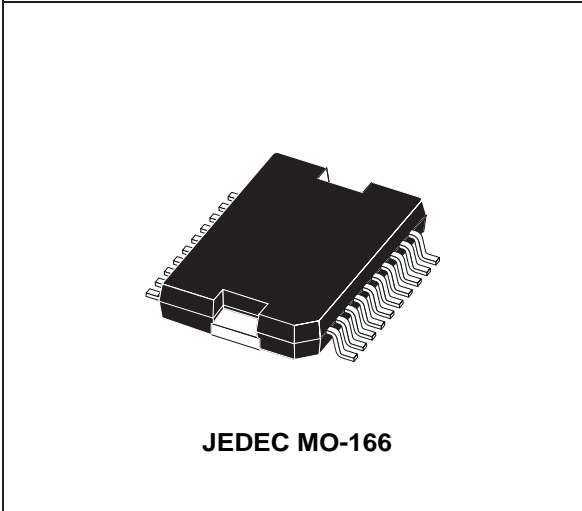
**Multiwatt15 H**



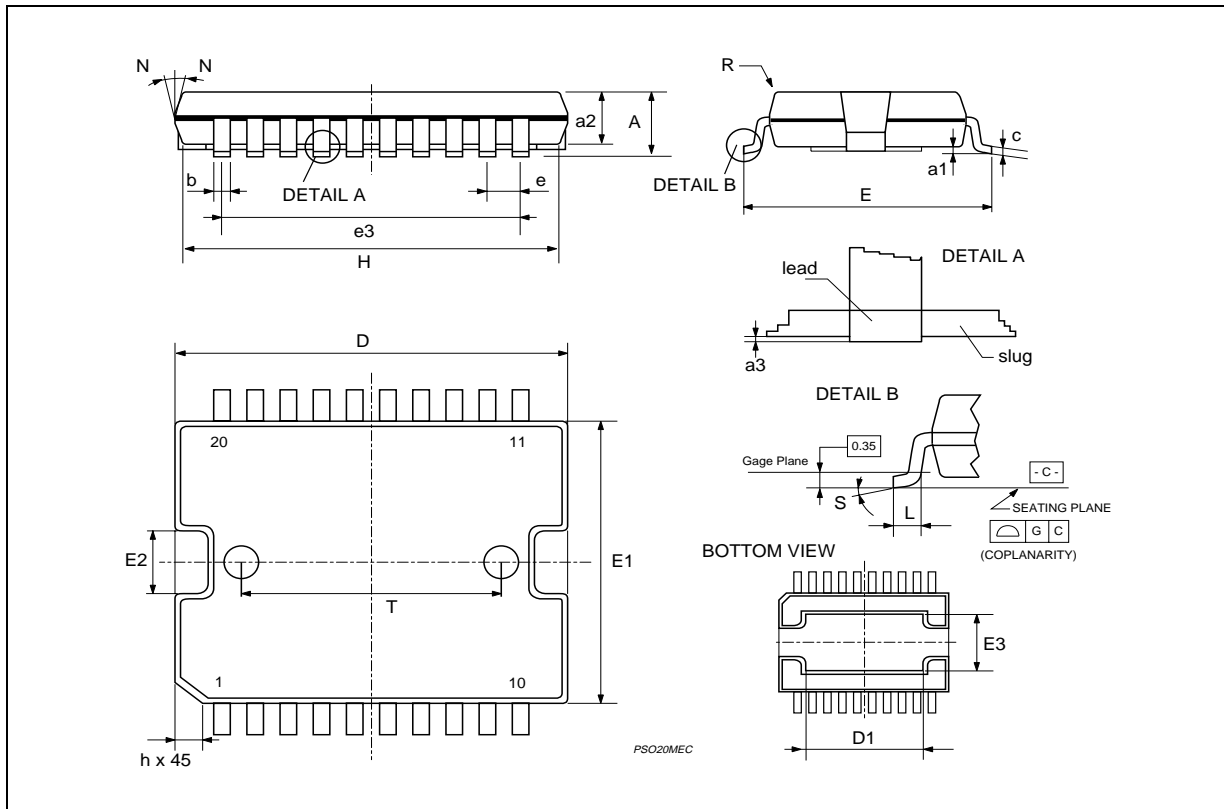
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

(1) "D and F" do not include mold flash or protrusions.  
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").  
 - Critical dimensions: "E", "G" and "a3"

**OUTLINE AND MECHANICAL DATA**



**PowerSO20**



Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics  
© 2000 STMicroelectronics – Printed in Italy – All Rights Reserved  
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco -  
Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

<http://www.st.com>

**ANEXO B** – *Datasheet* do sensor Sharp GP2Y0A21YK