

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Leandro Magno Gomides de Sousa

**Modelagem e compensação de erro de sensores e
atuadores baseados em Arduino**

Monte Carmelo, Brasil

2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Leandro Magno Gomides de Sousa

**Modelagem e compensação de erro de sensores e atuadores
baseados em Arduino**

Trabalho de conclusão de curso apresentado à
Faculdade de Computação da Universidade Fe-
deral de Uberlândia, Minas Gerais, como re-
quisito exigido parcial à obtenção do grau de
Bacharel em Sistemas de Informação.

Orientador: Leandro Nogueira Couto

Coorientador: Mirella Silva Junqueira

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Monte Carmelo, Brasil

2018

Leandro Magno Gomides de Sousa

Modelagem e compensação de erro de sensores e atuadores baseados em Arduino

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado. Monte Carmelo, Brasil, 17 de Dezembro de 2018:

Prof. Dr. Leandro Nogueira Couto
Orientador

Profa. Ma. Mirella Silva Junqueira
Coorientadora

Profa. Dra. Ana Cláudia Martinez

Prof. Dr. Jefferson Rodrigo de Souza

Monte Carmelo, Brasil
2018

Dedico este trabalho aos meus pais, irmão, minha esposa Andreia, e aos meus orientadores Leandro Nogueira Couto e Mirella Silva Junqueira, e a toda minha família que com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Agradecimentos

Agradeço a Deus, primeiramente, que permitiu que tudo isso acontecesse, ao longo da minha vida, e não somente nestes anos como universitário, mas em todos os momentos. A esta universidade, à direção, administração e seu corpo docente por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

À minha família e amigos tenho um agradecimento muito especial porque acreditaram em mim desde o primeiro instante. Sou quem sou porque vocês estiveram e estão sempre ao meu lado.

A todos os colegas, companheiros que estiveram ao meu lado diretamente ou indiretamente durante esta conquista, o meu muito obrigado.

"Não importa aonde você parou... Em que momento da vida você cansou... O que importa é que sempre é possível e necessário "Recomeçar". Recomeçar é dar uma chance a si mesmo... É renovar as esperanças na vida e o mais importante... Acreditar em você de novo. " -
Carlos Drummond de Andrade

Resumo

Neste trabalho temos como objetivo a modelagem de erros e o desenvolvimento de um algoritmo para compensar os erros presentes em sensores e atuadores robóticos de baixo custo compatíveis com a plataforma Arduino, para navegação de um robô autônomo. Para isto, propõe-se o estudo dos erros inerentes às leituras dos sensores, como o erro na distância medida por um sensor sonar, em comparação com medidas reais de distância, e erros de atuação de atuadores, como as divergências de distância percorrida por motores e rodas sobre terrenos diferentes. A modelagem e análise destes erros permitem a proposta de um algoritmo capaz de compensar os erros, permitindo que medidas mais precisas sejam obtidas e que o erro cumulativo seja mitigado, além de oferecer uma compreensão importante das particularidades, limitações e possíveis aplicabilidades destes equipamentos em situações reais.

Lista de ilustrações

Figura 1 – Ciclo de interação entre robô e ambiente. Retirado de (PIERI, 2002). . .	13
Figura 2 – Unimate	16
Figura 3 – Asimo	17
Figura 4 – Stanley, vencedor do Grand Challenge em 2005 (THRUN et al., 2006) . .	18
Figura 5 – Motor CC 6V	20
Figura 6 – Motor DC 3-6V com Caixa de Redução e Eixo Duplo	20
Figura 7 – Placa de desenvolvimento Arduino Uno.	22
Figura 8 – Curva da distribuição Normal. Retirado de (FREUND, 2006)	23
Figura 9 – Curvas da variação de exemplos de valores positivos e negativos para Kurtosis. Retirado de (GEORGE; MALLERY, 2016)	24
Figura 10 – Curvas da variação de exemplos de valores positivos e negativos para Skewness. Retirado de (GEORGE; MALLERY, 2016)	24
Figura 11 – Gráfico de exemplo do teste Shapiro-Wilk. Retirado de (ACTION, 2017)	25
Figura 12 – Modelo Ultrassônico HC-SR04.	28
Figura 13 – Robô montado com kit robótico proposto.	28
Figura 14 – Exemplo da medição realizada.	29
Figura 15 – Resultados das leituras Sensor 1	32
Figura 16 – Resultados das leituras Sensor 2	33
Figura 17 – Resultados das leituras dos atuadores nos quatro terrenos.	34
Figura 18 – Gráficos da regressão linear dos dois sonares.	39

Lista de tabelas

Tabela 1	– Média das leituras dos sonares em CM	35
Tabela 2	– Média de deslocamento em relação ao tempo	36
Tabela 3	– Análises de normalidade Sensor 1	36
Tabela 4	– Análises de normalidade Sensor 2	37
Tabela 5	– Análises de normalidade Atuadores	38
Tabela 6	– Média das leituras dos sonares em CM após compensar os erros	40
Tabela 7	– Porcentagem da média de erro das leituras dos Sensores	40

Lista de abreviaturas e siglas

VANTs	Veículos Aéreos Não Tripulados
RM	Robótica Móvel
CC	Motores de Corrente Contínua
DC	Direct Current
GPS	Global Positioning System
mm	Milímetros
cm	Centímetros
m	Metro
CBIE	Congresso Brasileiro de Informática na Educação.

Sumário

1	INTRODUÇÃO	12
1.1	Desafios em Robótica Móvel	14
1.2	Problema	15
1.3	Objetivos	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos específicos	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Robótica	16
2.2	Sensores e Atuadores	18
2.3	Arduino	21
2.4	Estudo dos Erros	22
2.4.1	Regressão Linear	25
3	METODOLOGIA	27
3.1	Composição do Robô	27
3.2	Coleta de dados dos Sensores e atuadores	28
3.2.1	Sensores	28
3.2.2	Atuadores	30
3.2.3	Análises das leituras dos Sensores e Atuadores	30
4	EXPERIMENTOS E RESULTADOS	32
4.1	Resultados	32
4.1.1	Leitura dos Sensores	32
4.1.2	Leituras Atuadores	33
4.2	Média	35
4.2.1	Sensores	35
4.2.2	Atuadores	35
4.3	Testes de normalidade	36
4.3.1	Sensores	36
4.3.2	Atuadores	37
4.4	Regressão linear	38
4.4.1	Regressão Linear dos Sensores	38
4.4.2	Código de compensação de erro	38

5	CONSIDERAÇÕES FINAIS	41
5.1	Publicação gerada	41
5.2	Trabalhos futuros	42
	REFERÊNCIAS	43
	ANEXO A – CÓDIGO DO SONAR	45
	ANEXO B – CÓDIGO DOS ATUADORES	46
	ANEXO C – CÓDIGO DE CALIBRAGEM	48
	ANEXO D – CÓDIGO PARA OBTER LEITURA REAL DO SONAR	51

1 Introdução

A robótica tem proporcionado novas soluções para antigos problemas, permitindo com que muitas tarefas trabalhosas, suscetíveis a falhas humanas e perigosos sejam automatizados. Por exemplo, com o uso em aplicações domésticas (aspiradores de pó), industriais (transporte automatizado, soldas automatizadas), Urbanas (transporte, cadeiras de rodas automatizadas), militares (sistemas de monitoramentos aéreos remotos, VANTs) e de segurança e defesa civil e militar (nos controles de patrulhamento e resgate de ambientes hostis) (WOLF et al., 2009).

A robótica também tem sido usada como ferramenta de ensino de programação, também melhorando de forma prática a reflexão e compreensão do conteúdo em disciplinas de Física e Matemática (SOUSA et al., 2016), além de catalisar o aprendizado e criatividade dos alunos em diversas disciplinas ou até mesmo incentivar a consulta a especialistas fora do ambiente escolar em busca da solução de um novo problema, através da integração entre diferentes áreas do conhecimento, quebrando as fronteiras estabelecidas anteriormente entre diferentes conteúdos e mostrando o potencial da interdisciplinaridade (PIO; CASTRO; JÚNIOR, 2006).

Neste contexto pode-se observar a importância do desenvolvimento em pesquisas relativas à robótica no país, promovendo uma revolução que transformar à nossas vidas no futuro. O desenvolvimento da Robótica Móvel(RM) está associado de forma direta a diversas carreiras como Engenharia Mecânica, Engenharia Elétrica, Engenharia da Computação e nas diferentes áreas da computação (WOLF et al., 2009).

WOLF et al. aponta que a área da computação em particular possui um papel importante no suporte de conhecimentos e técnicas que irão permitir o desenvolvimento de sistemas de controles mais robustos, seguros, autônomos e inteligentes para estes robôs móveis.

PIO; CASTRO; JÚNIOR define um robô como um agente autônomo capaz de obter informações de um ambiente permitindo utilizar essas informações para formar uma base de conhecimento que será utilizada para se deslocar de uma forma segura e significativamente intencional (um processo chamado de navegação) para atuar e executar tarefas, de uma forma mais flexível onde não existam limites geográficos, pois robôs móveis não precisam ser fixados como a maioria dos robôs convencionais nas indústrias, sendo capazes de se locomover por meios de pernas, rodas, esteiras, por empuxo ou flutuação.

MARCHI et al. inclui como elemento importante da definição um robô, a utilização

de sensores e atuadores que agem sob o controle de um sistema computacional equipando um dispositivo mecânico que permite executar tarefas repetitivas, enfadonhas ou insalubres, classificando os robôs em três tipos:

- Teleoperados - Controlados por um operador informando os movimentos que o robô deve executar.
- Semi-autônomos - Onde o operador indica através de uma programação (macro de comando) o que deve ser executado e o robô o faz sozinho.
- Autônomos - Capazes de realizar uma tarefa sozinho, tomando suas próprias decisões, com base nas informações obtidas pelos sensores os quais são equipados. Essa autonomia é um desafio de grande interesse em muitas aplicações.

A interação da robótica móvel com o ambiente ocorre através de ciclos de percepção e ação que consistem em três passos fundamentais:

- Obtenção de informações do ambiente através dos sensores;
- Processamento das informações e seleção das ações a serem executadas;
- Execução das ações planejadas através dos atuadores

Onde o robô realiza por diversas vezes esse ciclo de percepção e ação, que consiste na modificação do estado do ambiente em busca da realização de tarefas (PIERI, 2002), conforme demonstra a Figura 1.

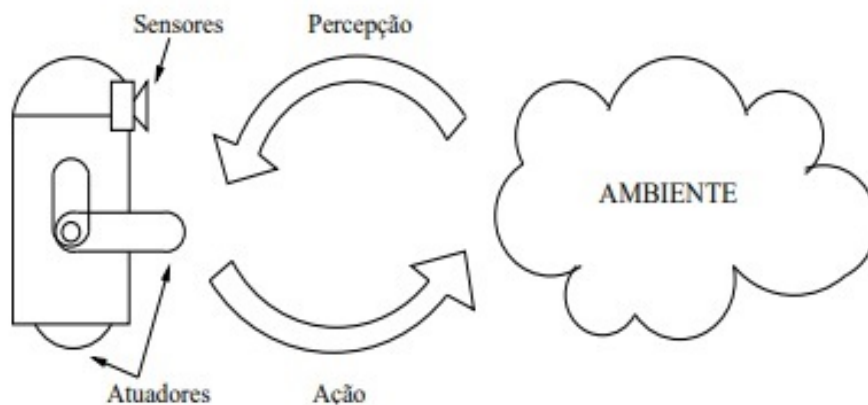


Figura 1 – Ciclo de interação entre robô e ambiente. Retirado de (PIERI, 2002).

Essa interação com o ambiente que proporciona o funcionamento fundamental de um robô, permitindo que sinta seu mundo através dos seus sensores, se assemelhando aos seres

humanos e alguns animais que o fazem por intermédio de seus sentidos, diferentes de sistemas ou um conjunto de aparelhos interligados que recebam essas informações por outros tipos de entradas (pela leitura de uma memória, por exemplo), não podendo ser considerados um robô (MATARIC et al., 2014).

1.1 Desafios em Robótica Móvel

O Desenvolvimento de um RM esta diretamente ligado a diversos problemas menores, como localização e navegação, erros de posição, erros de orientação, ruídos nas leituras dos sensores, problemas com a carga de baterias, falhas mecânicas, erros de programação (PIERI, 2002).

PIERI mostra que ao desenvolver um robô procura-se muitas das vezes se espelhar em nossas habilidades, fazendo assim com que máquinas com características humanas nos apresentem uma série de desafios como:

1. Navegação

Criar uma forma de representar o ambiente até um ponto que precisem ser refinados e livres de erros e que consiga maximizar o desempenho do robô.

2. Aprendizado

Ao adicionar novos sensores e atuadores ao robô o tornamos mais complexo, o que também implica que teremos uma complexidade maior no trabalho de programação de ações, calibragem dos sensores e coordenação de movimentos. Quanto mais sensores e atuadores, mais dados o robô precisa gerenciar.

3. Cooperação

Consiste na pesquisa da criação de múltiplos robôs, geralmente de pequeno porte e de complexidade limitada, mas que cooperem entre si para realizar tarefas complexas.

4. Cognição

Um dos grandes desafios é o de explorar os limites da inteligência artificial, e desenvolver uma forma de extrair informações do ambiente, aprender, armazenar o conhecimento e utilizá-lo nas tomadas de decisões, permitindo ao robô atingir um alto nível de autonomia e capacidades cognitivas com relação ao meio exposto.

1.2 Problema

Diante dos fatos expostos sobre a robótica, um grande desafio é obter as informações precisas através dos sensores que servirão de base para que o robô consiga tomar suas próprias decisões. No mercado temos à disposição diversos tipos e modelos de sensores que apresentam dos mais baixos aos mais elevados custos, cada um com níveis diferentes de erro e precisão.

Os sensores de alto custo contam com a desvantagem de serem menos acessíveis, muitas vezes, sendo utilizados em áreas industriais de alto desempenho ou alta precisão que não permitem os mínimos erros. Já os sensores de baixo custo são mais acessíveis permitindo o fácil acesso, sendo apropriados para ensino de programação, pequenas soluções, ou lazer, porém podem contar com uma magnitude de erro maior, prejudicando a precisão da tarefa.

Muitos sensores e atuadores de baixo custo, embora sejam úteis e acessíveis, são pouco documentados e pouco usados em pesquisa e ensino. Isso acontece em parte pelo fato que muitos sensores estão se popularizando apenas recentemente, e em parte por conta da precisão inferior. O que permite realizar, um estudo do perfil de erro de tais sensores e propostas para compensação de erros é importante para aumentar a viabilidade destes dispositivos e esclarecer para quais tarefas são apropriados, dependendo dos requisitos de precisão.

Este trabalho também foi motivado pelo aprendizado e pelas dificuldades encontradas no projeto de ensino de Robótica para alunos do Ensino Médio do Colégio Nossa Senhora do Amparo, realizado na cidade de Monte Carmelo - MG (SOUSA et al., 2016).

1.3 Objetivos

1.3.1 Objetivo Geral

Analisar o perfil de erro e propor um algoritmo capaz de compensar os erros presentes em sensores e atuadores de baixo custo, utilizados em robôs compostos pela plataforma Arduino, em diferentes ambientes.

1.3.2 Objetivos específicos

- Mapear os erros presentes nos sensores e atuadores, em diferentes condições.
- Identificar um padrão no perfil dos erros encontrados.
- Propor um algoritmo capaz de compensar estes erros dependendo do ambiente e do perfil de erro encontrado no sensor ou atuador.

2 Revisão Bibliográfica

2.1 Robótica

Segundo Couto (2012), a robótica é uma ciência recente, que ainda está em processo de se difundir. A data do seu surgimento é difícil de especificar devido às diferentes definições que possui o termo robô. Assim, podemos considerar robôs como máquinas que utilizam de alguns propósitos inteligentes para se tornarem autônomas, como as que se originaram em meados do século XX. Conforme o Unimate (Figura 2), considerado o primeiro robô industrial que dispunha de um braço de aproximadamente duas toneladas para realizar tarefas repetitivas e perigosas através de uma sequência de comandos pré-determinados.



Figura 2 – Exemplo de um braço robótico Unimate¹

Implantado em 1961 esse robô se tratava de uma ferramenta com recursos avançados sem inteligência ou capacidade de interagir com o meio ambiente e tomar decisões. Diante da evolução computacional há uma evolução no robô inteligente, capaz de observar e interagir com o ambiente, procurando a simular a inteligência humana. Tais características de interação e reconhecimentos como de faces, comandos de voz, e capacidade de locomoção cada vez mais avançada através de sensores, foram surgindo robôs mais avançados, e entre eles os robôs móveis, sendo que hoje há até mesmo robôs bípedes como o Asimo da Honda (Figura 3), (COUTO, 2012).

A utilização de robôs autônomos com capacidade de atuação em diversas áreas como exploração, combates a incêndios, vigilâncias e resgates através de um raciocínio inteligente e interação ao meio, tem despertado a atenção de diversos pesquisadores, fazendo com que esses

¹ <https://www.prsrobots.com/unimate.html>, acessado em 08/2018

² <http://asimo.honda.com>, acessado em 08/2018.



Figura 3 – Robô humanóide Asimo, desenvolvido pela Honda²

robôs trabalhem cooperativamente a fim de cumprir uma missão atribuída a eles, com ou sem interação entre eles. A realização dessas tarefas que expõem a vida humana em risco, ou realizada de forma ineficiente e até mesmo exaustiva, sendo realizadas por meio de aplicações robóticas e um grande sonho da sociedade afirma (PESSIN et al., 2016).

Á área da robótica pode ser dividida em várias outras sub-áreas, uma dessas sub-áreas é a RM. Há tarefas que só podem ser realizadas através do deslocamento dos robôs de várias formas nos ambientes terrestres como aquáticos e aéreos, podendo ser esse deslocamento classificado em 3 níveis de autonomia tele-operado, semi-autônomo ou totalmente autônomo. A RM oferece uma grande perspectiva no futuro de aplicações em diversas áreas como indústria, pesquisas, atividades agrícolas, militares e domésticas (COUTO, 2012).

As pesquisas em robótica móvel requerem conhecimentos em diversas áreas, como Computação, Matemática, Física, Engenharia Elétrica e Mecânica dentre outras. Temos como exemplos nos últimos anos a aplicação em algumas áreas, como na agricultura e no transporte, dentre os robôs mais conhecidos são os utilizados em explorações espaciais. Entre os 3 níveis de autonomia de deslocamento, o interesse por robôs móveis autônomos tem obtido um grande crescimento e demanda, devido serem capazes de operar nos ambientes internos e externos, aprendendo a tomar decisões corretas e com êxito diante de suas tarefas sem a intervenção humana (SOUZA, 2014).

Temos como exemplos de aplicações em desenvolvimento de robótica móvel na execu-

ção de tarefas cotidianas dos seres humanos os veículos autônomos inteligentes, que já vem sendo usados, com o intuito da diminuição dos acidentes em ruas e rodovias, assim como redução de custos de viagens e melhoramentos logísticos, permitindo aumentar a eficiência assim como a segurança no trânsito (SOUZA, 2014). Um exemplo do sucesso dos veículos autônomos foi o Grand Challenge de 2005, foi um evento organizado pelo DARPA (agência de pesquisas do Departamento de Defesa dos Estados Unidos), com o intuito de realizar uma competição entre alguns carros autônomos que deveriam navegar um percurso de 11.78 km no deserto (COUTO, 2012). A Figura 4 apresenta o Stanley, veículo terrestre não tripulado ganhador da realização do percurso.

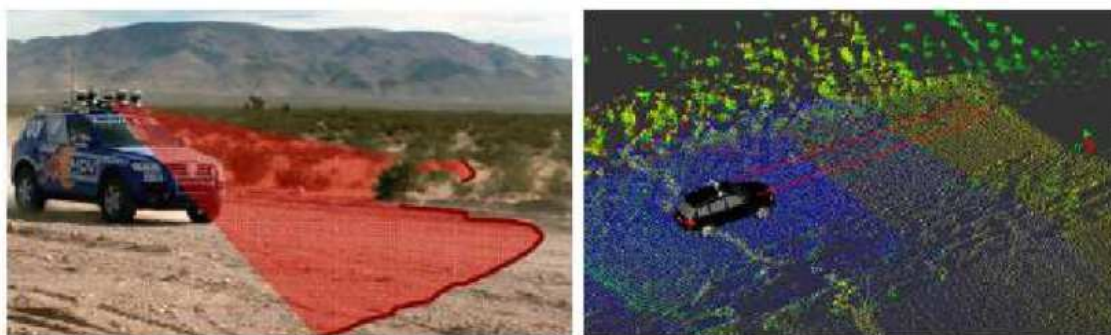


Figura 4 – Stanley, vencedor do Grand Challenge em 2005 (THRUN et al., 2006)

Aplicações na área da robótica com esses níveis de exigência requerem um processamento rápido das informações de objetos dinâmicas e fatores climáticos obtidos através dos diversos sensores, assim como nas tomadas de decisões nos momentos imprevisíveis e imprecisos obtidas pelos sensores e atuadores. Sistemas dessa magnitude devem ser seguros e tolerantes a falhas de forma que não causem danos aos seres humanos ou prejuízos para a população (SOUZA, 2014).

2.2 Sensores e Atuadores

MATARIC et al. afirma que corpos robóticos são diferentes dos seus equivalentes biológicos no que diz respeito à sua construção e ao seu funcionamento. Robôs apresentam dois componentes que permitem ao robô realizar seus objetivos.

Os efetadores, dispositivos que permitem que o robô realize um efeito de impacto sobre o ambiente, através dos atuadores. Os Atuadores podem ser definidos como um mecanismo capaz de efetuar ou executar uma ação ou movimento.

Nos robôs temos como exemplos de atuadores motores elétricos, cilindros hidráulicos ou pneumáticos (usados em braços mecânicos, por exemplo), materiais sensíveis à tempera-

tura e componentes químicos, que reagem com a mesma ideia dos músculos e tendões para movimentarem braços e pernas nos animais, materiais fotorreativos, e dispositivos piezo-elétricos.

Um atuador pode trabalhar de forma ativa que precisa de uma constante alimentação externa como a energia ou passiva que utiliza a energia gerada pela ação da mecânica dos efetadores em sua interação ao ambiente exposto.

Dos tipos de atuadores descritos vamos detalhar um tipo em específico os motores, que são mais comuns de se encontrar na área de robótica. Os motores podem ser classificados como:

- Motores de Corrente Contínua (CC) também conhecidos como motores DC (*Direct Current*) necessitam de uma constante corrente elétrica em uma determinada faixa de tensão, onde a variação da tensão determina a potência em que o motor irá funcionar. Se a tensão aplicada for muito alta, maior potência se obtém do motor, mas se a tensão for baixa, menor será a potência. Esses motores são muito simples e baratos e acessíveis permitindo a compra em grande variedade e escala de tamanho, de acordo com a necessidade e para diferentes tipos de atividades como na robótica, aparelhos de reprodução de mídia de CDs, DVDs, e em maquinários das indústrias siderúrgicas. As Figuras 5 e 6 mostram modelos desse motor.

Esses motores podem utilizar um recurso que permite aumentar a força e o torque de saída, que são as engrenagens. A disposição das engrenagens, por exemplo em séries ou agrupadas, realiza diferentes transformações de forças e promovem uma grande variedade de forças, velocidades e aplicações.

- Servomotores - são motores que permitem o deslocamento do giro do seu eixo para uma posição específica, através da junção de motores CC, engrenagens, sensores de posição do eixo do motor que auxiliam na direção a ser tomada e circuitos eletrônicos que controlam a direção em ângulos por exemplo. Os servomotores são encontrados facilmente em brinquedos, como por exemplo para se controlar a direção de carrinhos remotos e as asas de aviões de aeromodelismo.

Além dos atuadores responsáveis pelo deslocamento, os robôs precisam de dispositivos que permitam extrair as informações sobre si mesmo e do ambiente a sua volta que são necessárias para tomada de decisões, para isso eles disponibilizam dos sensores que os dispositivos responsáveis para coletar essas informações. Os sensores podem ser classificados por diferentes formas como por métodos de medição, passivos ou ativos.

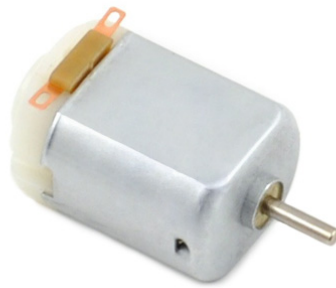


Figura 5 – Motor CC 6V



Figura 6 – Motor DC 3-6V com Caixa de Redução e Eixo Duplo

- Sensores ativos: são aqueles que emitem um tipo de pulso de energia no ambiente e realiza uma estimativa através da energia retornada até eles, como por exemplo os sensores de medida de distância como Sonares e *lasers*.
- Sensores passivos: diferentes dos ativos, estes realizam as estimativas através das informações obtidas nas leituras externas disponíveis no ambiente, como por exemplo sensores de câmera, temperatura, calor, umidade, giroscópios e acelerômetros.

Para WOLF et al., os sensores podem ser divididos em duas classes: proprioceptivos, que são os que realizam medidas dos dados internos do próprio robô como a carga da sua bateria e giro das rodas (odometria), e os sensores extraceptivos ou exteroceptivos, que obtêm as informações do meio ambiente em que o robô está inserido, como o GPS, sonar e câmera, que lhe permitem obter informações de localização.

Em um projeto robótico temos que levar em consideração as diferentes tarefas e necessidades de percepção e precisão que o robô precisará executar para escolher os sensores adequados, que serão capazes de obter as informações necessárias.

Um robô pode ser composto por um ou mais sensores, e essa fusão de sensores pode ser classificada da seguinte maneira:

- **Redundante ou Competitiva:** para se inserir redundâncias nas decisões do robô, que ajuda a reforçar um tipo de medida, podem ser inseridos dois modelos diferentes de sensores para um determinado tipo de leitura. Por exemplo, a medida de distância, para a qual podemos utilizar dois sensores de mesmo tipo, como dois sensores de sonar de marcas diferentes, para inserir redundância na medida, devido ao fato de que os sensores irão obter um mesmo tipo de informação com precisões diferentes.
- **Complementar:** que disponibiliza de dois sensores que fornecem informações diferentes, como por exemplo um sensor térmico para identificar um alvo através de informações de calor e uma câmera para eliminar falsas detecções por meio de movimentos ou de reconhecimento pessoal em uma caso de resgate do corpo de bombeiros, ou um sensor de sonar e um câmera ou um sensor de laser permitindo assim uma maior confiabilidade no processo de localização e navegação.
- **Coordenada:** utilizado-se sensores em uma determinada ordem para identificar a presença de uma pessoa por exemplo em um ambiente utilizando um sensor de reconhecimento de voz, em seguida uma câmera ou um sensor térmico para validar essa presença. Isso é importante porque em um robô real é necessário levar em conta limitações de energia e processamento.

2.3 Arduino

O Arduino é uma ferramenta de prototipagem eletrônica de hardware livre, permitindo assim com que pessoas de diversos níveis de conhecimento tenham acesso ao projeto e desenvolvimento de circuitos eletrônicos. O Arduino possui uma linguagem de programação estruturada baseada em C ou Orientada a Objetos C++, com um conjunto de funções próprio e algumas modificações. A placa tem como objetivo permitir o desenvolvimento rápido de projetos de baixo custo e de fácil utilização, permitindo uma conexão entre outros aparelhos ou dispositivos através de Conexões sem fio ou seriais (VASCONCELOS; JUNIOR; SOARES, 2013), e através de *shields*, que são placas com conexões padronizadas desenvolvidas para projetos que necessitam de funções específicas(ELIAS et al., 2014).

O uso do Arduino viabiliza o desenvolvimento de robôs de baixo custo com o acoplamento de diversos sensores e atuadores compatíveis, tendo em vista a grande quantidade de sistemas microcontroladores e componentes eletrônicos compatíveis existentes no mercado, e a grande quantidade de documentação e bibliotecas disponíveis.

O Arduino UNO, modelo mais popular do Arduino, tem como mecanismo principal um microcontrolador ATMEGA, desenvolvido pela Atmel Corporation, que permite ao desenvolvedor criar diversos tipos de sistemas automatizados, robôs e etc. Ele possui uma porta de comunicação USB para se conectar a um computador, por um cabo USB-AB, usada principalmente para que o desenvolvedor possa programar algoritmos no microcontrolador na linguagem de programação própria do Arduino. A placa Arduino pode ser visualizada na Figura 7.



Figura 7 – Placa de desenvolvimento Arduino Uno.

2.4 Estudo dos Erros

Quando falamos em estudo dos erros estamos referenciando uma análise estatística, em que toda variável aleatória assume uma determinada distribuição e é afetada por fatores aleatórios ou intrínsecos. Na estatística há vários modelos de distribuições teóricas que procuram através de um determinado evento em função de sua ocorrência traçar um comportamento para o mesmo (LEOTTI; COSTER; RIBOLDI, 2012). No caso de sensores, podemos considerar que toda medição M é na verdade uma medição de $M = M_{real} + \epsilon$, sendo ϵ a letra grega comumente usada para indicar o erro da medida. Em muitos casos é possível traçar um perfil para o erro ϵ , de forma a reduzi-lo. De qualquer forma, entender como o erro de sensores influencia o desempenho de certas tarefas é essencial.

Neste trabalho iremos aplicar a análise estatística de distribuição de normalidade. É muito comum que fenômenos naturais como o erro de medida tenham uma distribuição do tipo "normal", também conhecida como a "curva em forma de sino", que tem valor maior em um ponto central e se prolonga indefinidamente em ambas as direções. Visto que ao realizar

diversas aferições de uma determinada medida em um determinado aparelho, não se necessariamente chegará ao mesmo resultado todas as vezes, neste caso obtendo-se um conjunto de valores que oscilam em uma proximidade de um valor central. Através da construção de um histograma com esses valores conseguimos a imagem de uma forma aproximadamente simétrica, denominando assim a "lei normal dos erros" deduzida matematicamente a distribuição normal como a distribuição de probabilidade das observações dos erros por Gauss (CORREIA, 2003), (FREUND, 2006). A Equação 2.1 representa a matemática desse tipo de curva, seguida da Figura 8 que representa o gráfico da curva da distribuição normal.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad -\infty < x < \infty \quad (2.1)$$

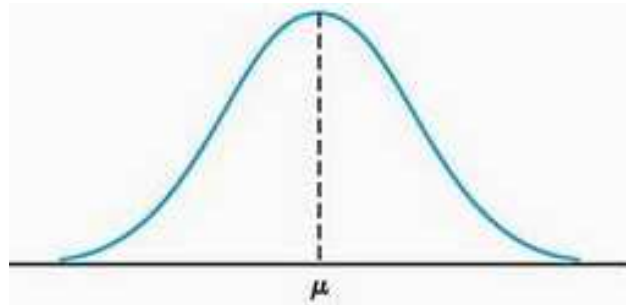


Figura 8 – Curva da distribuição Normal. Retirado de (FREUND, 2006)

Saber se uma determinada medida segue uma distribuição normal é uma informação útil. Para verificar a normalidade da distribuição de uma variável, temos alguns testes para avaliar e quantificar os valores de frequências e assimetria encontrados no conjunto de medidas. Dentre os testes existentes, temos:

- **Kurtosis:** é a medida do "pico" ou "achatamento" de uma distribuição (WESTFALL, 2014). Onde o valor de Kurtosis próximo de zero (0) indica uma forma próxima do normal, valores positivos indicam uma distribuição mais alta do que o normal e um valor negativo indica uma forma mais plana que o normal. Um valor extremamente negativo (por exemplo, <-5,0) indica uma distribuição em que mais valores estão nas extremidades (caudas) da distribuição do que em torno da média, já um valor entre ±1,0 é considerado excelente para a maioria dos propósitos psicométricos, mas um valor entre ±2,0 é em muitos casos também aceitável, dependendo da aplicação em particular. Sendo em alguns casos esses valores apenas diretrizes, pois em outros cenários, podem surgir diferentes critérios, como desvio significativo da normalidade (fora de ±2σ o erro padrão) (GEORGE; MALLERY, 2016), a imagem 9 demonstrar a variação da curva

para valores positivo e negativo, e a Equação 2.2 mostra a fórmula para o cálculo do valor de Kurtosis.

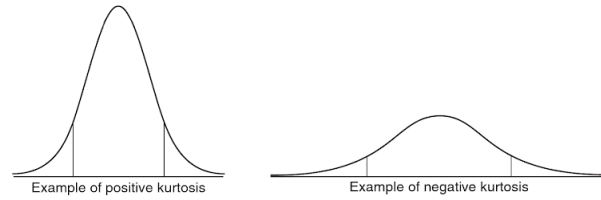


Figura 9 – Curvas da variação de exemplos de valores positivos e negativos para Kurtosis. Retirado de (GEORGE; MALLERY, 2016)

$$k = \sigma^{-4} \sum_{i=1}^G (i - \mu)^4 p(i) - 3 \quad (2.2)$$

- **Skewness:** é a medida em que até que ponto uma distribuição de valores se desvia da simetria em torno da média. Um valor zero (0) representa uma distribuição simétrica ou equilibrada de maneira uniforme, o que é o comportamento esperado de uma distribuição normal. Uma assimetria positiva indica um número maior de valores menores (tanger em sentido contrário, porém está correto). Um valor negativo indica um maior número de valores maiores. Assim como Kurtosis, um valor de assimetria entre $\pm 1,0$ é considerado excelente para demonstrar que uma distribuição é simétrica, mas um valor entre $\pm 2,0$ é, em muitos casos, também aceitável, dependendo da sua aplicação (GEORGE; MALLERY, 2016). A Figura 10 demonstra a variação da curva para valores positivo e negativo, enquanto a Equação 2.3 mostra a fórmula para o cálculo de Skewness.

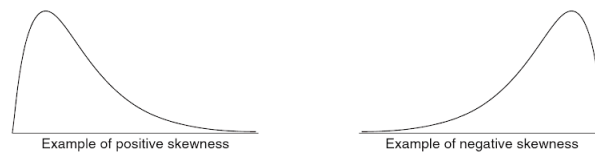


Figura 10 – Curvas da variação de exemplos de valores positivos e negativos para Skewness. Retirado de (GEORGE; MALLERY, 2016)

$$s = \sigma^{-3} \sum_{i=1}^G (i - \mu)^3 p(i) \quad (2.3)$$

- **Shapiro-wilk:** é um teste criado por Shapiro e Wilk (1965) e que pode ser aplicado para amostras de tamanho $4 \leq n \leq 2000$, (FÁVERO; BELFIORE, 2017). Comparando valores de uma amostra a uma distribuição normal modelo de mesma média e variância dos valores encontrados na amostra. Indicando que o teste das amostras são normais quando os dados da amostra não se diferem significativamente de uma distribuição normal onde $p > 0,05$ (não-significativo). No caso contrário, quando $p < 0,05$ (significativo), temos indicativo de que a distribuição da amostra é significativamente diferente de uma distribuição normal (FIELD, 2009). A Figura 11 exemplifica um gráfico gerado pelo teste de Shapiro-Wilk.

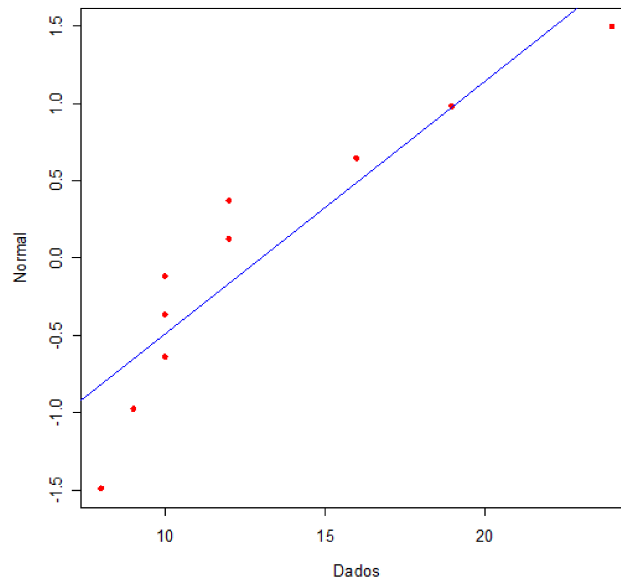


Figura 11 – Gráfico de exemplo do teste Shapiro-Wilk. Retirado de (ACTION, 2017)

2.4.1 Regressão Linear

Outra ferramenta importante quando se trata de estimar o comportamento de dados é a regressão linear. A regressão linear é um método que encontra uma função no formato $y = a + bx$, isto é, uma linha reta, para descrever a relação entre duas variáveis x e y . Para caracterizar a regressão linear para um conjunto de medidas de pontos (x, y) , é calculado o valor de a e b , de acordo com as Equações 2.4 e 2.5, respectivamente. Nas equações, n representa o número de observações.

$$a = \frac{\sum (y) \sum (x^2) - \sum (x) \sum (xy)}{n \sum (x^2) - (\sum (x))^2} \quad (2.4)$$

$$b = \frac{n \sum (xy) - \sum (x) \sum (y)}{n \sum (x^2) - (\sum (x))^2} \quad (2.5)$$

A regressão nos diz qual a melhor curva a ser traçada que se ajusta melhor nesses dados, podendo ser uma reta ou uma curva qualquer. Ao olhar para o gráfico pode se ter um palpite se é uma reta, ou um polinômio de segundo grau, por exemplo.

O importante é que o método lhe diz qual a melhor curva para informa que a distância entre os pontos e a curva ajustada seja a menor possível. Ou seja, minimizando o erro consigo estimar com maior precisão.

3 Metodologia

3.1 Composição do Robô

Para este projeto, foi utilizado um robô controlado pela plataforma Arduino, pelo seu baixo custo e facilidade de uso. A composição do robô consiste de periféricos (sensores e atuadores). Os sensores escolhidos foram o sonar modelo HC-SR04 mostrado na Figura 12, que é um sensor sonar ultrassônico capaz de medir a distância dentro de uma faixa de 2 centímetros a 4 metros, sendo um sensor de baixo custo. Sensores do tipo sonar funcionam com um módulo transmissor e um módulo receptor, além de um circuito de apoio. Como todo sonar, o HC-SR04 calcula a distância de um obstáculo através do tempo de demora para o sinal emitido pelo transmissor chegar no receptor. Sabendo a velocidade do som (que varia de acordo com o meio, sendo 340 m/s no ar, mas chegando a 1500 m/s na água, por exemplo), podemos calcular a distância com a fórmula 3.1.

$$distancia = (tempo \times velocidade\ do\ som / 2) \quad (3.1)$$

Segundo o *datasheet* do dispositivo ¹, a precisão do HC-SR04 pode chegar a 3 milímetros. Nesse caso, 3 milímetros é um limite inferior para a precisão. Essa é uma constatação importante para este trabalho, pois a precisão real pode variar significativamente e esse dado não está presente no *datasheet*.

Os atuadores escolhidos foram quatro rodas, controladas por dois motores (um para cada lado do robô controlando duas rodas) representado pela Figura 6, e uma cabeça mecânica do tipo *pan-tilt*, que permite o posicionamento do sonar acoplado a ela. A Figura 13 mostra um dos robôs usados no projeto.

O kit robótico utilizado também satisfaz os requisitos de usabilidade que foi levantado. Ele permite grande personalização e extensão, já que há uma vasta gama de sensores compatíveis com a plataforma Arduino. A gravação do controlador é bastante fácil, o Arduino utiliza uma IDE (Interface de Desenvolvimento Integrado) de fácil uso, existente para os três grandes Sistemas Operacionais (Windows, Linux e Mac OS), e a interface do robô com o computador ocorre pela porta USB. Além disso, a programação é feita usando uma versão simplificada da linguagem C, que é comumente usada em cursos superiores como linguagem

¹ <https://www.mouser.com/ds/2/813/HCSR04-1022824.pdf>

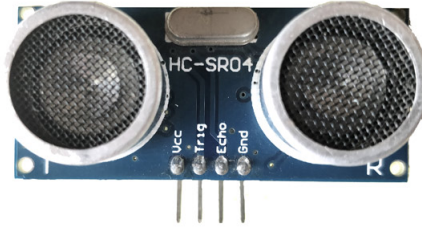


Figura 12 – Modelo Ultrassônico HC-SR04.

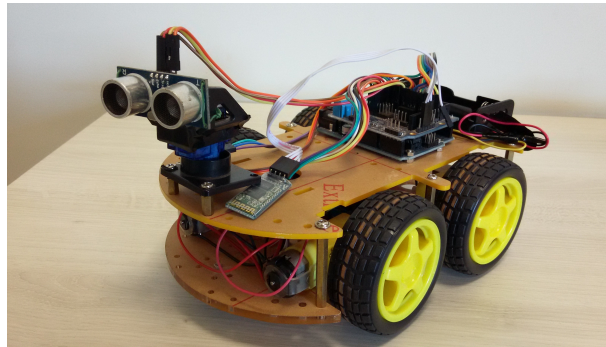


Figura 13 – Robô montado com kit robótico proposto.

para introduzir o ensino de programação, e que permite evitar conceitos diagnosticados como principais causadores de dificuldade aos alunos no aprendizado como gerenciamento de memória e orientação a objetos (MILNE; ROWE, 2002) (ROBINS; ROUNTREE; ROUNTREE, 2003).

Por fim, uma qualidade que o kit robótico proposto apresenta que não se vê em outros kits é que, devido a seu baixo custo e facilidade de acesso no Brasil. O kit proposto, completo com sensores, atuadores, Arduino, baterias e chassi, tem custo cerca de seis vezes menor que o custo do equipamento concorrente, e por ser hardware livre, pode ser usado, modificado e explorado sem restrições.

3.2 Coleta de dados dos Sensores e atuadores

Nesta parte do projeto será especificada a maneira pela qual foi realizada a leitura dos sensores e atuadores presentes no robô.

3.2.1 Sensores

Via de regra, o sensor ultrassônico HC-SR04 é usado para auxiliar na locomoção do robô como critério para tomada de decisão. Muitas vezes, como no caso do SLAM, ou

Simultaneous Localization and Mapping (BAILEY; DURRANT-WHYTE, 2006), o sonar é usado para compensar o erro intrínseco de atuadores como o motor, servindo para validar a distância percorrida, permitindo aumentar a precisão da locomoção a ser realizada pelo robô. Para verificar a precisão real do sensor, em comparação com a precisão informada pelo fabricante, foi realizado alguns testes de leituras com 2 sensores do mesmo modelo.

Os testes foram realizados marcando um ponto inicial em relação a um ponto final (parede) em uma superfície plana, efetuando a medição da distância entre esses 2 pontos com uma trena com precisão de 1 milímetro (fita ou régua flexível utilizada para medir distância em milímetros, centímetros, e metros), em seguida posicionando o sensor sonar no ponto inicial, e registrando o valor de distância obtido. Para cada sonar foram realizadas as medições 3 vezes para um conjunto de distâncias, sendo elas:

- **2 cm** distância mínima de leitura informada pelo fabricante
- **50 cm; 1 m; 1,50 m; 2 m; 2,50 m; 3 m; 3,50 m;** variação de 50 cm.
- **4 m** distância máxima de leitura informada pelo fabricante

A Figura 14 demonstra o procedimento de marcação dos pontos inicial e final.

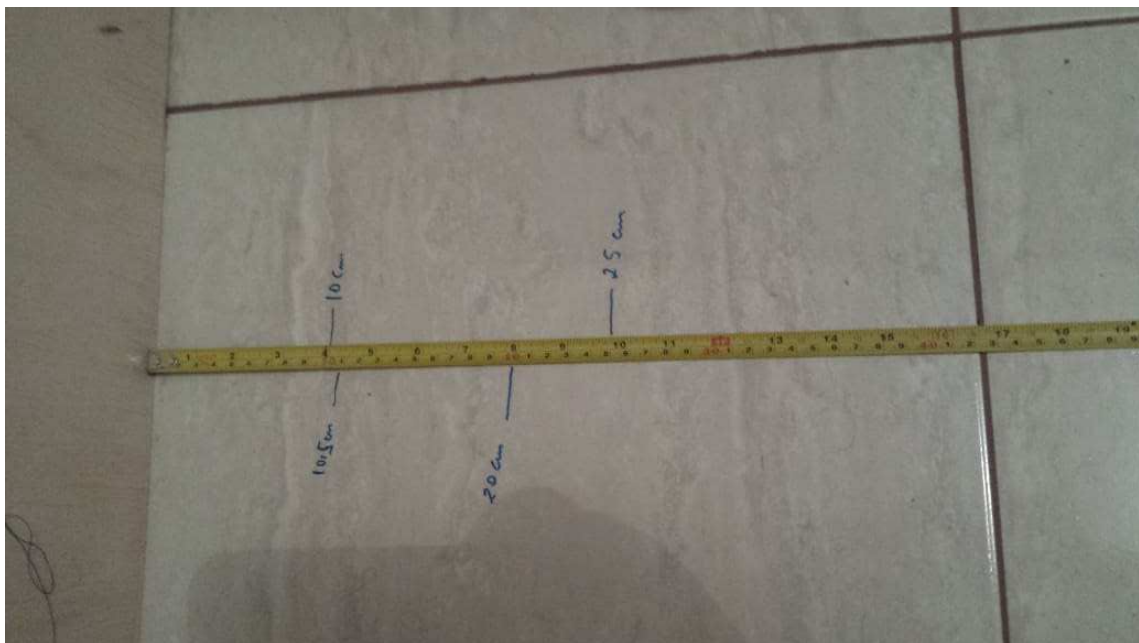


Figura 14 – Exemplo da medição realizada.

Para cada uma das distância a leitura do sonar foi repetida 1000 vezes. O código utilizado para realizar a leitura dos dados do sonar foi adaptado do código disponibilizado

pelo site FILIPEFLOP, utilizando a biblioteca "Ultrasonic.h" para controle do sensor sonar, adicionando um contador para auxiliar na contagem das leituras realizadas conforme demonstrado pelo Anexo A.1.

3.2.2 Atuadores

Foram utilizados 4 rodas fixadas em motores de redução DC (corrente contínua), com tensão de operação de 3V a 6V, como atuadores integradas a uma Ponte H Dupla modelo L298N. O nome motor de redução se deve ao fato de o motor usar engrenagens que reduzem a velocidade para gerar torque ².

A coleta de dados dos atuadores foi efetuada através de um teste em que o código de deslocamento é configurado com um valor de tempo determinado de execução da ação de andar. O tempo determinado para esta ação foi dado pela função **delay(2000)**, ou seja, um tempo de 2000 milissegundos, conforme demonstrado no Anexo B.1. Posicionando o robô em um ponto inicial ao executar o código o robô se deslocava pelo tempo determinado e em seguida com o auxílio de uma trena (com precisão de 1 milímetro) foi marcada a distância percorrida pelo robô.

Os testes foram efetuados 15 vezes para 4 diferentes tipos de terreno, sendo eles:

- Terreno plano com grama
- Terreno plano com piso liso
- Terreno com declive com piso de concreto
- Terreno com alicive com piso de concreto

3.2.3 Análises das leituras dos Sensores e Atuadores

Os dados coletados das leituras dos sensores e atuadores foram organizados em tabelas e submetidos a análises de erros através da análise de distribuição de normalidade, pelos testes de normalidade de Shapiro-Wilk, kurtosis e Skewness.

Todos os testes de normalidade foram efetuados com o auxílio de ferramentas de cálculos online sendo elas SciStatCalc ³ para Shapiro-Wilk e Kurtosis, e Skewness Calculator ⁴ para Skewness. Após realizar os testes de normalidade com os resultados dos cálculos foram criados os gráficos.

² <https://wrokits.com.br/catalog/article/7>

³ <http://scistatcalc.blogspot.com/2013/10/shapiro-wilk-test-calculator.html>

⁴ <https://www.easycalculation.com/statistics/skewness.php>

Através da análise dos resultados, foi proposto um algoritmo para correção do erro de leitura do sonar através de calibragem e cálculo do erro estimado. Isto foi realizado através da estimativa de uma função linear que determina o erro de acordo com a distância, por regressão linear. Ou seja, baseado no modelo de função de linha reta $y = a + bx$, deseja-se encontrar uma função que descreve o erro ϵ e a distância d seguindo a função $\epsilon = a + bd$. Medindo a distância d com o sonar, podemos calcular uma estimativa da distância real d_{real} sabendo que $d_{real} = d - \epsilon$.

4 Experimentos e Resultados

Esse capítulo apresentará todos os resultados obtidos: valores obtidos nos testes de normalidade dos sensores e atuadores, e uma análise dos resultados do sonar de forma a buscar uma calibragem do sensor que possa reduzir o erro.

4.1 Resultados

Os resultados dos sensores foram organizados em gráficos que representa a frequência da leitura dos sonar em um total de 1000 leituras para cada distância, posicionado a frente de uma parede de alvenaria em um ambiente externo.

4.1.1 Leitura dos Sensores

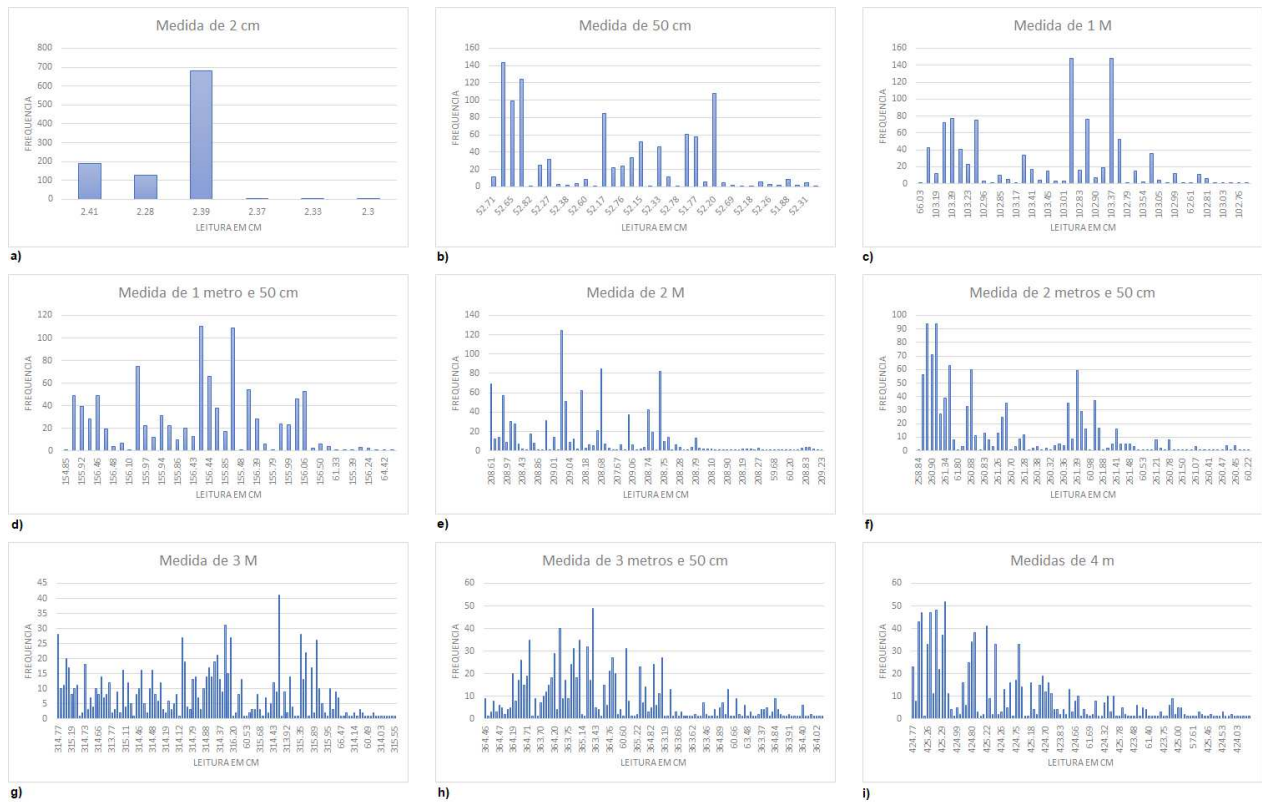


Figura 15 – Resultados das leituras Sensor 1

Analisando os gráficos para cada uma das leituras no Sensor 1, podemos observar que para as distâncias menores que 2 M, temos uma oscilação menor nas frequências das leituras.

Apesar da oscilação da frequência nota-se que as medidas repetidas com maior frequência estão entre o intervalo 2 CM e 10 CM.

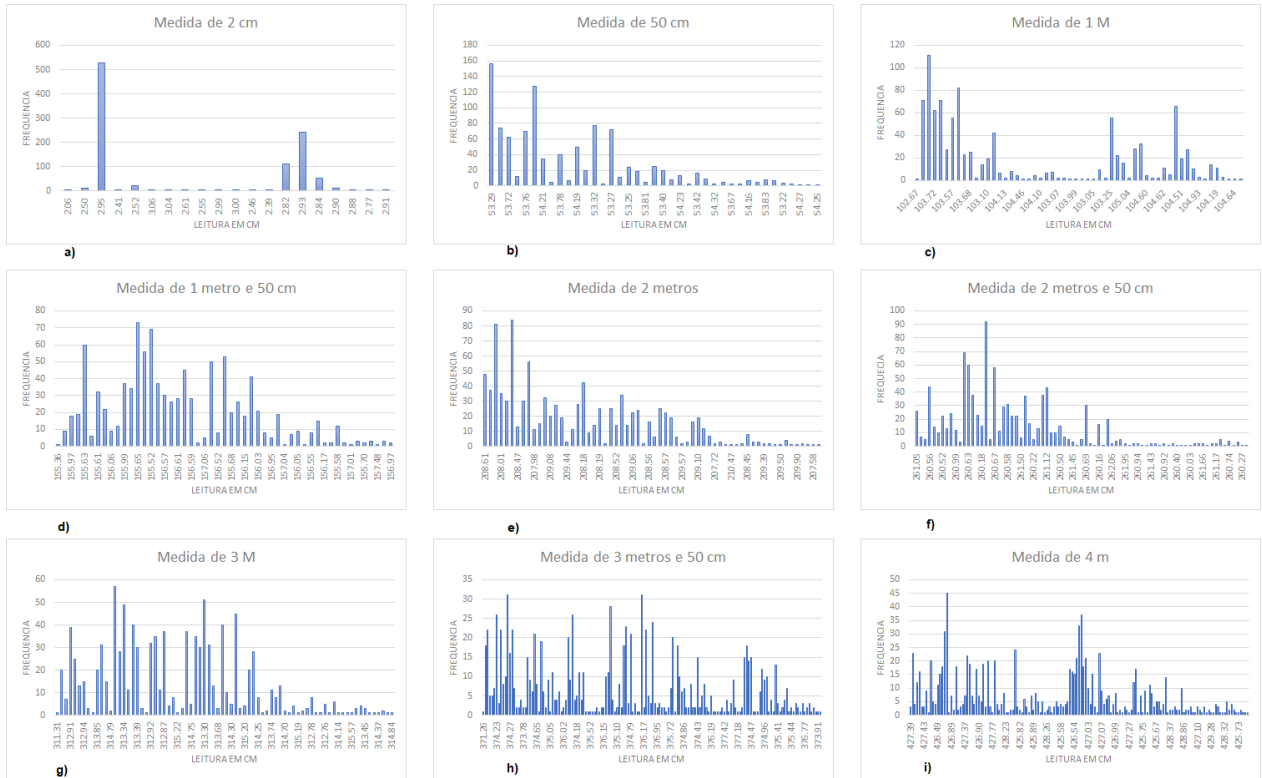


Figura 16 – Resultados das leituras Sensor 2

Em relação às leituras do sonar 1 pode se observar que no Sonar 2 há uma oscilação maior para as medidas acima de 50 CM, e uma frequência de repetição maior entre 3 CM > medida real < 5 CM, conforme apontam os gráficos b, c, d, na Figura 16 e > que 8 CM para as medidas acima de 2 M conforme os gráficos e, f, g, h e i.

4.1.2 Leituras Atuadores

Os resultados dos atuadores foram organizados em gráficos apresentando a frequência das distâncias em que o robô se deslocou para 15 leituras em um terreno específico no tempo de 5 segundos.

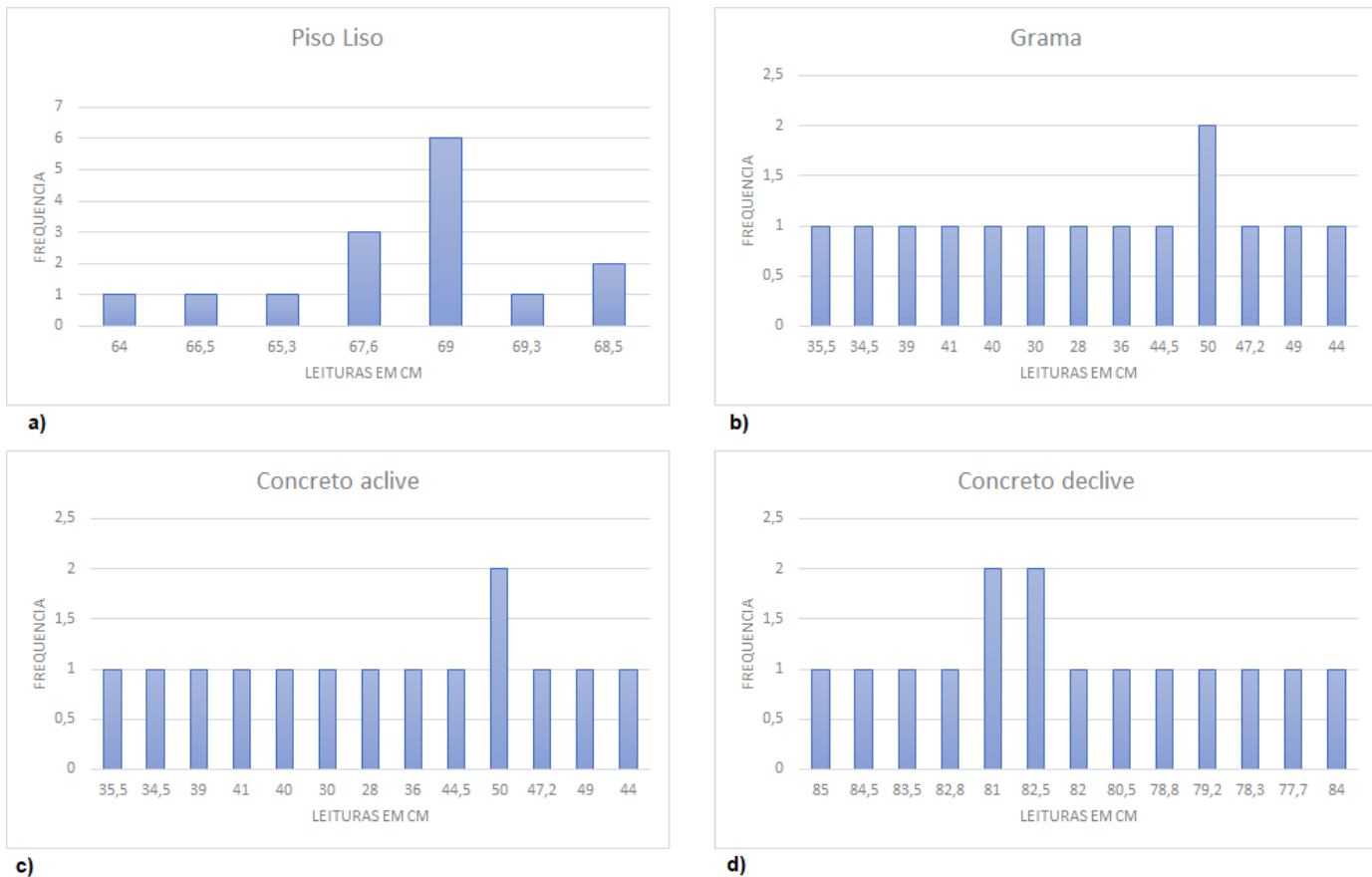


Figura 17 – Resultados das leituras dos atuadores nos quatro terrenos.

Observando de uma forma geral podemos ver que para alguns terrenos um dos testes acaba influenciando na distribuição de normalidade fazendo com que os resultados sejam bastante dependentes do terreno.

Podemos afirmar que esteja dentro do padrão devido as seguintes situações:

- Grama, neste terreno muitos fatores podem influenciar a umidade presente na grama faz com que as rodas do robô possam patinar e ficar em um mesmo local por um tempo, não conseguindo fazer ele deslocar em relação a uma grama seca, outro fator nesse terreno é altura em que a grama se encontra o que pode resultar também no robô em patinar ou ficar preso(embaraçado) na grama, analisando em relação ao tamanho e peso do robô.
- Concreto com acive, neste terreno um dos fatores que podem influenciar o desempenho do atuadores e a fonte de energia a qual o robô está utilizando, devido o terreno ser uma subida com irregularidades irá exercer uma força maior dos atuadores.

- Concreto com declive, neste terreno ao contrário do com alicha os atuadores não precisam exercer uma força maior, mais devido ao peso do robô pode fazer com que ele se desloque a mais do que o desejado.

Dentre os terrenos apresentados o que se manteve uma distância normal percorrida em relação ao tempo determinado foi o terreno de piso liso, onde sua superfície plana não apresentou demais dificuldades aos atuadores.

4.2 Média

4.2.1 Sensores

A Tabela 1 demonstra a média das leituras de dois sensores Ultrassônico do mesmo modelo HC-SR04 e números de serie de fabricação.

Tabela 1 – Média das leituras dos sonares em CM

Valor Real	2	50	100	150	200	250	300	350	400
Média Sonar 1	2,38	52,21	103,11	155,66	204,84	258,46	312,28	358,95	417,88
Erro Médio Sonar 1	0,38	2,21	3,11	5,66	4,84	8,46	12,28	8,95	17,88
Média Sonar 2	2,90	53,57	103,61	156	208,44	260,85	313,38	374,89	426,42
Erro Médio Sonar 2	0,9	3,57	3,61	6	8,44	10,85	13,88	24,89	26,42

É possível observar no caso do Sonar 1 uma precisão parecida com a de 3 milímetros mencionada no *datasheet* para o melhor caso, ou seja, o caso de menor distância (2 cm). No caso do Sonar 2, mesmo na menor distância o erro foi de 9 milímetros. O Sonar 2 teve consistentemente um desempenho pior que o do Sonar 1, o que pode denotar alguma variação devida ao processo de fabricação do dispositivo. Nos dois casos, o erro aumenta com o aumento da distância, o que é esperado devido ao funcionamento do sonar baseado em uma onda mecânica (som). É possível perceber isso observando os erros médios no caso dos dois sonares (mais claramente no caso do Sonar 2). Nesse caso, podemos supor que há uma relação linear entre o erro e a distância, ou seja o erro aumenta de modo proporcional ao aumento da distância.

4.2.2 Atuadores

Como pode se observar pelas medidas das distâncias percorridas em um tempo de 5 segundos o terreno em que o robô obteve maior distância de deslocamento foi o terreno de declive de concreto, o que deve ser visto como indicativo de uma grande influência da

Tabela 2 – Média de deslocamento em relação ao tempo

Deslocamento no tempo de 5 segundos	
Terreno plano	Distância Média em CM
Piso Liso	67,92
Grama	41,59
Concreto Aclive	64,65
Concreto Declive	81,55

inclinação na capacidade do atuador parar no tempo determinado, visto que temos uma diferença pequena em relação ao terreno plano de piso liso com o terreno com aclive de concreto, já a grama em relação ao demais terrenos podemos observar que suas distâncias alcançadas foram as menores, provavelmente em decorrência de derrapagem e baixa aderência, o que implica que em aplicações envolvendo este tipo de terreno há necessidade de uma correção de calibragem.

4.3 Testes de normalidade

4.3.1 Sensores

Neste teste foi realizada a análise dos dados de 2 sensores HC-SR04, as Tabelas 3 e 4 apresentam os valores obtidos em cada um dos testes de normalidade para os sensores.

Tabela 3 – Análises de normalidade Sensor 1

Distância	Shapiro-Wilk	Kurtosis	Skewness
2 cm	0.539565	2.379	-2
50 cm	0.908902	-0.553	0.35
1 m	0.055131	245.837	-15.71
1,5 m	0.054151	194.743	-14.02
2 m	0.148903	33.467	-5.96
2,5 m	0.092437	71.927	-8.6
3 m	0.079250	94.894	-9.85
3,5 m	0.118296	51.754	-7.31
4m	0.123017	2473.758	-6.93

Analisando os dados do Sensor 1 pelo método de Shapiro-Wilk podemos observar que as distribuições das leituras estão um levemente fora da normalidade sendo pouco maiores ou próximas de 0.05. Já em relação a Kurtosis e Skewness, cujos valores devem ser próximos de ± 1.0 ou ± 2.0 como em algumas medidas temos valores acima de 2.0 podemos afirmar que a distribuição está bem distante da curva normal. O Skewness teve grande tendência negativa, com números negativos altos, o que indica que a curva das medidas é assimétrica e

pende para a direita ¹. No caso de Kurtosis, um valor alto como o obtido indica que existem muitos pontos fora da curva esperada, também chamados de *outliers* (WESTFALL, 2014), o que realmente pode ser visto nos gráficos gerados.

Tabela 4 – Análises de normalidade Sensor 2

Distância	Shapiro-Wilk	Kurtosis	Skewness
2 cm	0.506183	13.905	-3.47
50 cm	0.815520	-0.643	0.78
1 m	0.896839	0.564	0.92
1,5 m	0.899697	-0.590	0.6
2 m	0.887332	0.656	0.9
2,5 m	0.913855	-0.011	0.67
3 m	0.893522	1.567	1.02
3,5 m	0.118296	2.754	1.3
4m	0.882607	3.162	1.52

Analizando o Sensor 2 pelos mesmos aspectos apresentados na análise do sensor 1, podemos perceber que para um mesmo modelo de sensor há uma diferença significativa em relação a distribuição de normalidade, visto que os testes para este sensor ficaram muito mais próximos aos limites para caracterização de normalidade dos testes. Com base nestes resultados, é possível afirmar que as medições do Sonar 2 estão muito mais próximas de uma distribuição normal, ou muito próxima, especialmente considerando os valores de Skewness e Kurtosis que estão em sua grande maioria dentro dos intervalos que caracterizam uma distribuição normal.

Estes resultados deixam claro que para um mesmo sensor sonar, de mesmo modelo, adquiridos no mesmo lote de venda, pode haver diferenças significativas de medições e perfil de erro. Por esse motivo, uma calibragem prévia como a que será proposta no fim deste capítulo é essencial em tarefas que exigem precisão de medida.

4.3.2 Atuadores

Os resultados para os testes dos atuadores é apresentada pelo tipo de terreno em que eles foram submetidos, através da análise do espaço amostras de 20 vezes consecutivas para cada terreno em relação ao tempo determinado.

Ao analisarmos a Tabela 5 podemos observar que o terreno de Piso Liso teve uma distribuição próxima à normalidade através dos resultados de Kurtosis e Skewness, mas o teste de Shapiro-Wilk foi maior que o esperado de uma distribuição normal. O mesmo para o teste no terreno de Grama. Para o terreno de concreto com alicate (subida) podemos observar

¹ <https://cnx.org/contents/bE-w34Vi@9/Descriptive-Statistics-Skewness-and-the-Mean-Median-and-Mode>

que sua normalidade esteve abaixo do esperado em relação todos os testes, já para o terreno de concreto declive (descida) podemos observar uma normalidade por Skewness, mas para os demais testes temos uma padrão fora da curva normal.

Tabela 5 – Análises de normalidade Atuadores

Terreno	Shapiro-Wilk	Kurtosis	Skewness
Piso Liso	0.784385	0.379	-1.34
Grama	0.954268	-1.189	-0.26
Concreto aclave	0.319862	8.261	-3.32
Concreto declive	0.956590	5.297	-0.2

Estes testes demonstram a necessidade das três medidas para determinar se uma distribuição é normal, pois três dos terrenos obtiveram valores adequados em pelo uma das medidas, mas a análise completa mostra que não devem ser consideradas distribuições normais.

4.4 Regressão linear

4.4.1 Regressão Linear dos Sensores

Após realizarmos a análise de distribuição de normalidade foi efetuado o cálculo da regressão linear para ambos os sensores sonares. Uma análise do erro permite ver uma relação com aparência linear em ambos os sonares, em que o aumento da distância aumenta o erro da medida. A partir das medidas realizadas a regressão linear oferece uma equação de reta que define uma relação aproximada entre o erro e a distância. A equação é dada através dos valores obtidos em **a** e **b**, calculados pelas Equações 2.4, 2.5 respectivamente. Dispondo os valores das leituras realizadas dos sensores juntamente com a reta calculada para cada sonar em um gráfico, podemos observar a uma reta crescente na dispersão dos dados obtidos nos sensores, é possível verificar que os pontos se localizam próximos à reta calculada, indicando que a linearidade da relação entre erro e distância pode ser esperada, mesmo em sonares de comportamentos tão distintos.

A Figura 18, apresenta a regressão linear dos dados para cada um dos sensores. Para os cálculos, foram usados os valores médios de erros calculados anteriormente.

4.4.2 Código de compensação de erro

Considerando o resultado dos teste de normalidade e da regressão linear, foi desenvolvido um algoritmo capaz de calcular a calibragem através das fórmulas da regressão linear,

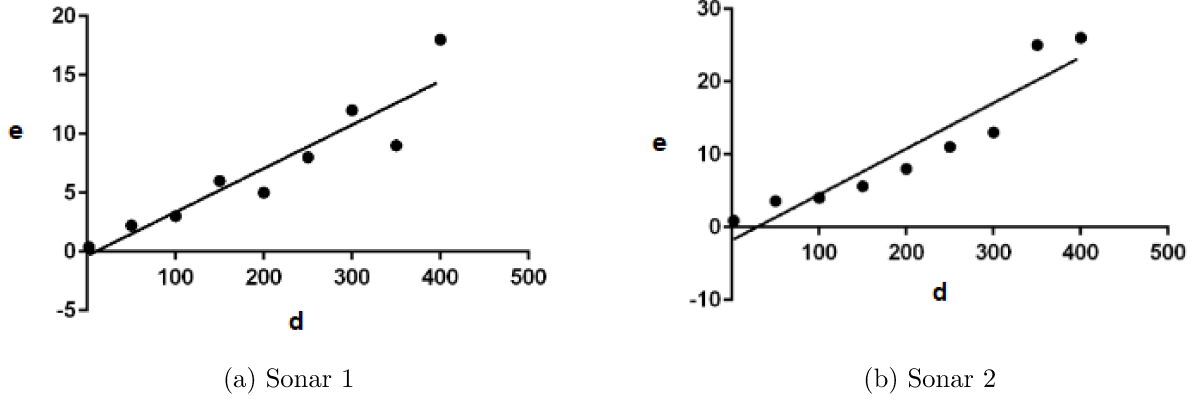


Figura 18 – Gráficos da regressão linear dos dois sonares.

permitindo assim através de simples modificação no algoritmo de leitura do sonar, obter-se uma precisão maior da distância real. Neste código proposto, o algoritmo foi dividido em 2 partes, o primeiro algoritmo sendo responsável por medir as distâncias e calcular o erro em um ambiente controlado, em que sabemos as medidas corretas, para realizar calibragem do sensor. O segundo algoritmo aplica os valores encontrados no algoritmo 1 sobre a leitura realizada pelo sensor, corrigindo o valor lido para um provavelmente mais próximo do real, conseguindo assim reduzir o erro e obter uma estimativa da distância real do obstáculo até o sensor.

O primeiro algoritmo consiste em ler um conjunto de medidas a distâncias pré-definidas, utilizando estes valores no cálculo de uma regressão linear específica para o erro do sonar sendo utilizado. A saída deste algoritmo são os valores de **a** e **b**, que são coeficientes da equação da reta que representa o erro do sonar. No anexo C.1 apresentamos a implementação deste algoritmo na linguagem do Arduino.

O segundo algoritmo é responsável por receber os valores dos coeficientes **a** e **b** já calculados para uma função que irá obter a leitura do sonar, e corrigir esta leitura com os valores dos coeficientes, compensando assim o erro do sonar e retornando o valor de distância real estimado. Unindo as equações apresentadas no capítulo anterior, especificamente a equação que relaciona a medida real e a medida com erro embutido $d_{real} = d - \epsilon$ e a equação da reta $\epsilon = a + bd$, temos a equação 4.1 que oferece a estimativa da distância real.

$$d_{real} = \frac{d - a}{1 + b} \quad (4.1)$$

A implementação na linguagem do Arduino do segundo algoritmo encontra-se no anexo D.1.

Após aplicar o algoritmo de calibragem de erros nas leituras presentes na Tabela 1, podemos observar através da nova Tabela 6, gerada que a leitura para cada um dos sensores obteve uma precisão maior em relação a leitura sem o algoritmo, tendo em alguns casos que o valor obtido foi exatamente igual ao valor real medido pela trena. O erro negativo, nesse caso, significa que a compensação de erro gerou uma estimativa menor que a medida real.

Para fins de comparação entre as Tabelas 1 e 6, no caso do sonar 1, o somatório do módulo dos erros médios arredondados antes da calibragem era de 63,77 CM, e depois da calibragem foi de 12,86 CM. No caso do sonar 2, o somatório do módulo dos erros médios arredondados antes da calibragem era de 98,56 CM, e depois da calibragem foi de 22,70 CM.

Tabela 6 – Média das leituras dos sonares em CM após compensar os erros

Distância Real	2	50	100	150	200	250	300	350	400
Media Sonar 1	2,63	50,69	99,77	150,45	197,87	249,58	301,48	346,48	403,31
Erro média sonar 1	0,63	0,69	-0,23	0,45	-2,13	-0,42	1,48	-3,52	3,31
Media Sonar 2	4,42	52,09	99,17	148,45	197,78	247,09	296,51	354,37	402,85
Erro média sonar 2	2,42	2,09	-0,82	-1,55	-2,21	-2,91	-3,49	4,37	2,84

Isso representa uma redução significativa dos erros, de mais de quatro vezes nos dois casos, considerando a média de medidas tomadas para diversas distâncias dentro do intervalo mínimo e máximo de operação dos sonares, o que demonstra a valiosa contribuição desta proposta de calibragem.

A Tabela 7 faz uma comparação em porcentagens da média do erro de leitura antes de aplicar o algoritmo de calibragem e após sua aplicação para ambos os sensores.

Tabela 7 – Porcentagem da média de erro das leituras dos Sensores

Distância Real	2	50	100	150	200	250	300	350	400
% do erro médio Sonar 1 sem calibragem	19	4,42	3,11	3,77	2,42	3,40	4,09	2,56	4,47
% do erro médio Sonar 1 após calibragem	31	1,38	0,23	0,3	1,06	0,28	0,49	1,00	0,82
% do erro médio Sonar 2 sem calibragem	45	7,14	3,61	3,85	4,22	4,34	4,62	7,11	6,60
% do erro médio Sonar 2 após calibragem	121	4,18	0,82	1,03	1,10	1,16	1,16	1,25	0,71

Ao analisar a Tabela 7 é possível observar que após a utilização do algoritmo de calibragem reduzimos a margem de erro da leitura em $\pm 1,00\%$, para as distâncias acima de 2 cm. Com exceção ao Sonar 2 que esta margem aplica-se as distâncias acima de 50 cm. De modo geral pode ser observar que o algoritmo reduziu o erro de leitura em $\pm 80\%$ para ambos os sensores.

5 Considerações finais

Neste trabalho tivemos como objetivo a modelagem e correção de erros presentes em sensores e atuadores robóticos de baixo custo compatíveis com a plataforma Arduino, para navegação de um robô autônomo. Para isto, propôs-se o estudo dos erros inerentes às leituras dos sensores, como o erro na distância medida por um sensor sonar, em comparação com medidas reais de distância, e erros de atuação de atuadores, como as divergências de distância percorrida por motores e rodas sobre terrenos diferentes.

Assim traçando um perfil de erro para 2 sensores HS-SR04 e um atuador composto por rodas e um motor DC. Nos resultados para o sensor pudemos identificar uma divergência de leitura entre os 2 sensores sonares para algumas medidas em particular, e analisando individualmente cada sensor pudemos notar que ambos apresentaram algumas divergência em suas leituras em relação ao valor real. O mesmo podemos notar para os atuadores que para cada terreno o mesmo atuador teve um desempenho diferente, sendo o erro maior em um determinado terreno e menor em outros.

Através desta análise foi possível propor um algoritmo que seja capaz de compensar o erro do sonar, através de cálculo efetuado com a entrada de um conjunto de leituras de distância e informações do terreno, permitindo assim realizar a calibragem. Mostramos que a calibragem e a estimativa da medida real da nossa proposta permite obter uma precisão significativamente maior em relação as leituras dos sensores. Por esse motivo, oferecemos o código Arduino completo de calibragem e correção e sugerimos a implementação de nossa abordagem para qualquer aplicação em que haja necessidade de precisão maior na medida de distância do sonar, e potencialmente outros sensores.

5.1 Publicação gerada

A seguinte publicação foi gerada pelo autor durante seu curso de graduação, juntamente com seu orientador e colaboradores, tendo assunto relacionado com a pesquisa desenvolvida neste trabalho:

Sousa, Leandro, et al. "Ensino de Programação em Robótica Móvel no Ensino Fundamental e Médio." Anais dos Workshops do Congresso Brasileiro de Informática na Educação. Vol. 5. No. 1. 2016.

5.2 Trabalhos futuros

Seguindo o desenvolvimento deste trabalho com trabalhos futuros, os itens abaixo são sugestões de diversas possibilidades de pesquisa podem ser exploradas a partir dos resultados obtidos neste trabalho:

- Realizar a análise para múltiplos motores DC, oferecendo um método de calibragem para eles também.
- Realizar a análise para diferentes tipos de sensores como o mesmo propósito como os sensores de localização. A proposta de calibragem é válida para qualquer sensor linear, e os sensores de baixo custo do Arduino podem se beneficiar de mais precisão em muitas tarefas.
- Efetuar um estudo da relação de como o resultado de sensor pode implicar positivamente e ou negativamente em outro sensor, como nos classificados como complementares, através da compensação de erro por diversos sensores, compondo um sistema robusto e de baixo custo.
- Aplicar o método de correção em um caso prático de navegação que use o sonar, por exemplo, numa tarefa de navegação e mapeamento SLAM.
- Aperfeiçoar o algoritmo para permitir a comparação de precisão entre sensores.

Referências

- PIERI, E. R. D. Curso de robótica móvel. *Santa Catarina, Florianópolis*, 2002. Citado 3 vezes nas páginas 7, 13 e 14.
- THRUN, S. et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, Wiley Online Library, v. 23, n. 9, p. 661–692, 2006. Citado 2 vezes nas páginas 7 e 18.
- FREUND, J. *Estatística aplicada: economia, administração e contabilidade*. 11^a. [S.l.]: Ed, 2006. Citado 2 vezes nas páginas 7 e 23.
- GEORGE, D.; MALLERY, P. *IBM SPSS Statistics 23 Step by Step: A Simple Guide and Reference*. Taylor & Francis, 2016. ISBN 9781134793334. Disponível em: <<https://books.google.com.br/books?id=vKLOCwAAQBAJ>>. Citado 3 vezes nas páginas 7, 23 e 24.
- ACTION, P. "Teste de Shapiro-Wilk". 2017. Acessado em 16-11-2018. Disponível em: <<http://www.portalaction.com.br/inferencia/64-teste-de-shapiro-wilk>>. Citado 2 vezes nas páginas 7 e 25.
- WOLF, D. F. et al. Robótica móvel inteligente: Da simulação às aplicações no mundo real. In: SN. *Mini-Curso: Jornada de Atualização em Informática (JAI), Congresso da SBC*. [S.l.], 2009. p. 13. Citado 2 vezes nas páginas 12 e 20.
- SOUZA, L. et al. Ensino de programação em robótica móvel no ensino fundamental e médio. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2016. v. 5, n. 1, p. 399. Citado 2 vezes nas páginas 12 e 15.
- PIO, J. L. de S.; CASTRO, T. H. C. de; JÚNIOR, A. N. de C. A robótica móvel como instrumento de apoio à aprendizagem de computação. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2006. v. 1, n. 1, p. 497–506. Citado na página 12.
- MARCHI, J. et al. Navegação de robôs móveis autônomos: estudo implementação de abordagens. Florianópolis, SC, 2001. Citado na página 12.
- MATARIC, M. J. et al. Introdução à robótica. Ed Unesp, 2014. Citado 2 vezes nas páginas 14 e 18.
- COUTO, L. N. *Sistema para localização robótica de veículos autônomos baseado em visão computacional por pontos de referência*. Dissertação (Mestrado) — Universidade de São Paulo, 2012. Citado 3 vezes nas páginas 16, 17 e 18.
- PESSIN, G. et al. Ambiente virtual para simulação de robôs móveis inteligentes aplicados no combate de incêndios florestais. *Disciplinarum Scientia/ Naturais e Tecnológicas*, v. 7, n. 1, p. 147–165, 2016. Citado na página 17.

SOUZA, J. R. d. *Navegação autônoma para robôs móveis usando aprendizado supervisionado*. Tese (Doutorado) — Universidade de São Paulo, 2014. Citado 2 vezes nas páginas 17 e 18.

VASCONCELOS, W. D. S.; JUNIOR, I. F. V.; SOARES, A. A. A. F. Estudo e desenvolvimento de uma rede sem fio baseada em bluetooth sobre a ferramenta de prototipagem arduino. In: *IX Congresso de Iniciação Científica do IFRN*. [S.l.: s.n.], 2013. Citado na página 21.

ELIAS, A. A. de A. et al. Ardweather: Uma estação meteorológica baseada no arduino e em web services restful. In: *Proceedings of Safety, Health and Environment World Congress*. [S.l.: s.n.], 2014. v. 14, p. 44–48. Citado na página 21.

LEOTTI, V. B.; COSTER, R.; RIBOLDI, J. Normalidade de variáveis: métodos de verificação e comparação de alguns testes não-paramétricos por simulação. *Revista HCPA. Porto Alegre. Vol. 32, no. 2 (2012), p. 227-234*, 2012. Citado na página 22.

CORREIA, M. S. B. B. Probabilidade e estatística. 2003. Citado na página 23.

WESTFALL, P. H. Kurtosis as peakedness, 1905–2014. rip. *The American Statistician*, Taylor & Francis, v. 68, n. 3, p. 191–195, 2014. Citado 2 vezes nas páginas 23 e 37.

FÁVERO, L.; BELFIORE, P. *Manual de análise de dados: Estatística e modelagem multivariada com Excel®, SPSS® e Stata®*. Elsevier Editora Ltda., 2017. ISBN 9788535285055. Disponível em: <<https://books.google.com.br/books?id=SmlaDwAAQBAJ>>. Citado na página 25.

FIELD, A. *Descobrimo a estatística usando o SPSS - 2.ed.* Bookman, 2009. ISBN 9788536320182. Disponível em: <<https://books.google.com.br/books?id=Zq059wGcnvWC>>. Citado na página 25.

MILNE, I.; ROWE, G. Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies*, Springer, v. 7, n. 1, p. 55–66, 2002. Citado na página 28.

ROBINS, A.; ROUNTREE, J.; ROUNTREE, N. Learning and teaching programming: A review and discussion. *Computer science education*, Taylor & Francis, v. 13, n. 2, p. 137–172, 2003. Citado na página 28.

BAILEY, T.; DURRANT-WHYTE, H. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, IEEE, v. 13, n. 3, p. 108–117, 2006. Citado na página 29.

FILIFELOP, S. *"Código Sensor Ultrassônico HC-SR04"*. Acessado em 16-08-2018. Disponível em: <<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>>. Citado na página 30.

ANEXO A – Código do Sonar

```
1 //Programa: Conectando Sensor Ultrassonico HC-SR04 ao Arduino
2 //Autor: Adaptado de FILIPEFLOP
3
4 //Carrega a biblioteca do sensor ultrassonico
5 #include <Ultrasonic.h>
6 //Define os pinos para o trigger e echo
7 #define pino_trigger 4
8 #define pino_echo 5
9
10 //Inicializa o sensor nos pinos definidos acima
11 Ultrasonic ultrasonic(pino_trigger, pino_echo);
12
13 int i;
14 void setup(){
15     Serial.begin(9600);
16     Serial.println("Lendo dados do sensor...");
17     i=0;
18 }
19 void loop(){
20     if(i>1001){
21         Serial.print("END");
22         delay(10000);
23         i=0;
24     }
25     float cmMsec, inMsec; //Le as informacoes do sensor, em cm.
26     long microsec = ultrasonic.timing();
27     cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
28     //Exibe informacoes no serial monitor
29     Serial.print(cmMsec);
30     Serial.print("; ");
31     Serial.println(i);
32     delay(500);
33     i++;
34 }
```

Listing A.1 – Código comunicação com o Sonar

ANEXO B – Código dos Atuadores

```
1 // Motor 1
2 int dir1PinA = 2;
3 int dir2PinA = 3;
4 int speedPinA = 9;
5
6 // Motor 2
7 int dir1PinB = 4;
8 int dir2PinB = 5;
9 int speedPinB = 10;
10
11 void setup() {
12     Serial.begin(115200);
13     //Define os pinos controladores da ponte h dupla L298N
14     pinMode(dir1PinA, OUTPUT);
15     pinMode(dir2PinA, OUTPUT);
16     pinMode(speedPinA, OUTPUT);
17     pinMode(dir1PinB, OUTPUT);
18     pinMode(dir2PinB, OUTPUT);
19     pinMode(speedPinB, OUTPUT);
20
21     delay(3000);
22 }
23
24 void loop() {
25     //Para Frente
26     analogWrite(speedPinA, 255);
27     digitalWrite(dir1PinA, HIGH);
28     digitalWrite(dir2PinA, LOW);
29     analogWrite(speedPinB, 255);
30     digitalWrite(dir1PinB, LOW);
31     digitalWrite(dir2PinB, HIGH);
32     delay(2000);
33
34     //Stop
35     analogWrite(speedPinA, 255);
```

```
36     digitalWrite(dir1PinA , LOW);  
37     digitalWrite(dir2PinA , LOW);  
38     analogWrite(speedPinB , 255);  
39     digitalWrite(dir1PinB , LOW);  
40     digitalWrite(dir2PinB , LOW);  
41     delay(5000);  
42 }
```

Listing B.1 – Código comunicação com os Atuadores

ANEXO C – Código de Calibragem

```
1 //Programa: Calibragem do sonar
2 //Autor: Leandro Magno
3
4 //Carrega a biblioteca do sensor ultrassonico
5 #include <Ultrasonic.h>
6
7 //Define os pinos para o trigger e echo
8 #define pino_trigger 4
9 #define pino_echo 5
10
11 //Inicializa o sensor nos pinos definidos acima
12 Ultrasonic ultrasonic(pino_trigger, pino_echo);
13 //variaveis
14 int medidas[9]; //guarda as leituras realizadas
15 //Captura somatoria de y(erros), x(valor real), x ao quadrado e x*y
16 float somaY=0, somaX=0, somaQX, somaXY;
17 float cmMsec, inMsec;
18
19 //----Funcoes-----
20 //ler a distancia do sonar
21 float lerDistancia(){
22     long microsec = ultrasonic.timing();
23     cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
24     return cmMsec;
25 }
26 //calcular as somatorias e salvar a distancia lidas
27 void salvarCalcularMedidas(){
28
29     int dist = 50; //variavel para informar a distancia a ser lida;
30
31     //Este trecho sera responsavel por ler os dados da posicao do
32     //sonar para cada uma das medidas informadas;
33     Serial.println("Posicione o sonar a cada distancia informada e
34     pressione enter para realizar a leitura");
```

```
34 Serial.println("Informe a leitura para a medida de 2 cm");
35 medidas[0]= lerDistancia()-2; //receber a diferenca do valor lido
    pelo sonar em relacao ao valor real
36 Serial.read(); //ler a tecla enter
37
38 somaY=medidas[0]; //receber o primeiro valor da somatoria do erro.
39 somaX=2; //recebe o primeiro valor da somatoria da leitura;
40 somaQX=2*2; //recebe o primeiro valor da somatoria de x ao quadrado
41 somaXY=medidas[0]*2; //recebe o primeiro valor da somatoria
42
43 //loop para realizar a leitura das distancia e grava no vetor;
44 for(int j=1; j < 9; j++){
45     Serial.print("Informe a leitura para a medida de ");
46     Serial.println(dist + "CM");
47     Serial.read(); //ler a tecla enter.
48     medidas[j]= lerDistancia()-dist;
49     somaY+=medidas[j];
50     somaX+=dist;
51     somaQX+=dist*dist;
52     somaXY+= medidas[j]*dist;
53     dist+=50;
54 }
55 }
56 //calcula o valor da expressao(formula) a
57 float valorA(){
58
59     float part1, part2, result;
60     //calcula a parte superior da expressao a;
61     part1 = (somaY*somaQX)-(somaX*somaXY);
62     //calcula a parte inferior da expressao onde 9 equeivale ao n;
63     part2 = (9*somaQX - (somaX*somaX));
64     result = part1/part2; //calcula a divisao da expressao;
65     return result;
66 }
67 //calcula o valor da expressao b
68 float valorB(){
69
70     float part1, part2, result;
71     //calcula a parte superior da expressao b onde 9 equeivale ao n;
```

```
72  part1 = (9*somaXY)-(somaX*somaY);
73  //calcula a parte inferior da expressao onde 9 equeivale ao n;
74  part2 = (9*somaQX - (somaX*somaX));
75  result = part1/part2; //calcula a divisao da expressao;
76  return result;
77  }
78  void setup(){
79    Serial.begin(9600);
80    Serial.println("Lendo dados do sensor...");
81  }
82  void loop(){
83
84    salvarCalcularMedidas();
85    Serial.print("O valor encontrado para A e: ");
86    Serial.println(valorA());
87    Serial.print("O valor encontrado para B e: ");
88    Serial.println(valorB());
89  }
```

Listing C.1 – Código de calibragem para Sensores Ultrassonico

ANEXO D – Código para obter leitura real do Sonar

```
1 //Programa: Distancia Real do sonar
2 //Autor: Leandro Magno
3
4 //Carrega a biblioteca do sensor ultrassonico
5 #include <Ultrasonic.h>
6 //Define os pinos para o trigger e echo
7 #define pino_trigger 4
8 #define pino_echo 5
9
10 //Inicializa o sensor nos pinos definidos acima
11 Ultrasonic ultrasonic(pino_trigger, pino_echo);
12
13 float a,b; //recebe os dados da calibragem
14 float cmMsec, inMsec;// variaves para receber os dados do sonar
15 void setup() {
16     Serial.begin(9600);
17     Serial.println("Informe os dados da calibragem para:");
18     Serial.print("A: ");
19     a=Serial.read();
20     Serial.print("\nB: ");
21     b=Serial.read();
22 }
23 //funcao para ler distancia sonar
24 float lerDistancia(){
25     long microsec = ultrasonic.timing();
26     cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
27     return cmMsec;
28 }
29
30 //funcao para calcular a distancia real.
31 float obtemDistanciaReal(float a, float b){
32     return (( lerDistancia() - a ) / (1+ b));
33 }
```

```
34 void loop() {  
35     Serial.print("A distancia lida pelo Sonar: ");  
36     Serial.println(lerDistancia());  
37     Serial.print("A distancia real: ");  
38     Serial.println(obtemDistanciaReal(a,b));  
39 }
```

Listing D.1 – Código leitura real do Sensor Ultrassonico