

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ricardo Silveira Rodrigues

**UM ESTUDO SOBRE TÉCNICAS DE DESCRIÇÃO DE PERCURSO DE
VEÍCULOS USANDO A PLATAFORMA ARDUINO**

Santa Maria, RS
2016

Ricardo Silveira Rodrigues

**UM ESTUDO SOBRE TÉCNICAS DE DESCRIÇÃO DE PERCURSO DE VEÍCULOS
USANDO A PLATAFORMA ARDUINO**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação.**

ORIENTADORA: Prof.^a Marcia Pasin

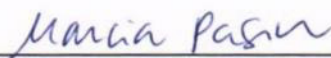
Trabalho de Graduação N^o 425
Santa Maria, RS
2016

Ricardo Silveira Rodrigues

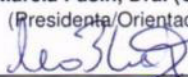
**UM ESTUDO SOBRE TÉCNICAS DE DESCRIÇÃO DE PERCURSO DE VEÍCULOS
USANDO A PLATAFORMA ARDUINO**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

Aprovado em 15 de dezembro de 2016:



Marcia Pasin, Dra. (UFSM)
(Presidente/Orientadora)



Mateus Beck Rutzig, Dr. (UFSM)



João Carlos Damasceno Lima, Dr. (UFSM)

Santa Maria, RS
2016

AGRADECIMENTOS

Agradeço à minha família, especialmente aos meus pais, Nilsa e Darlan que sempre me incentivaram e me apoiaram em todas as decisões da minha vida.

Agradeço aos meus colegas pela parceria durante toda a graduação, seja nas festas ou nos estudos.

Agradeço aos meus amigos por estarem sempre presentes e entenderem a minha ausência nos finais de semestre.

Agradeço à minha orientadora Marcia Pasin, pelo último ano de orientação e por permitir que a realização deste trabalho fosse possível.

RESUMO

UM ESTUDO SOBRE TÉCNICAS DE DESCRIÇÃO DE PERCURSO DE VEÍCULOS USANDO A PLATAFORMA ARDUINO

AUTOR: Ricardo Silveira Rodrigues

ORIENTADORA: Marcia Pasin

Este trabalho apresenta um serviço de descrição de percurso para uma rede veicular usando como suporte a plataforma robótica Arduino. Descrição de percurso é um serviço chave no contexto de redes veiculares e serve como base para outros serviços, como navegação, regulação de tráfego e geração de relatórios estatísticos. Plataformas de prototipação, como a Arduino, permitem investigar e avaliar novos serviços antes de disponibilizá-los na prática, enquanto a tecnologia para redes veiculares não está amplamente implantada. Um dos principais problemas com os sistemas de descrição de percurso é que eles dependem de serviços de localização, que não operam plenamente em condições específicas. Por exemplo, montanhas e prédios altos podem afetar a precisão do GPS. A solução aqui apresentada colabora para preencher esta lacuna. O ambiente opera sobre um veículo deslocando-se em um percurso previamente definido. Um conjunto de sensores de baixo custo, baseados na tecnologia de SONAR, detecta e localiza o veículo mesmo em ambientes fechados. Os resultados obtidos pela solução implementada corresponderam ao esperado. Com os valores oferecidos pelo conjunto de SONARs é possível inferir o percurso realizado por um veículo.

Palavras-chave: Localização. Percurso. Arduino. Descrição. Veículo. Sensor

ABSTRACT

A STUDY OF VEHICLE ROUTE DESCRIPTION TECHNIQUES USING ARDUINO PLATFORM

AUTHOR: Ricardo Silveira Rodrigues

ADVISOR: Marcia Pasin

This work presents a route description service for a vehicular network using the Arduino robotic platform as support. Route description is a key service in the context of vehicular networks. It serves as the basis for other services such as navigation, traffic regulation and statistical reporting. Prototyping platforms, such as Arduino, enable to investigate and evaluate new solutions before making them available in practice, while the target technology is not widely deployed. One of the main limitations of route description systems is the need of location services, which do not fully operate under specific conditions. For example, mountains and high buildings can affect GPS outcome. The solution presented here helps to fill this gap. Our Solution operates in a scenario in which a vehicle is following a previously defined route. A set of low cost sensors, based on SONAR technology, detects and locates the vehicle even indoors. The results obtained by the implemented solution corresponded to the expected one. With the values offered by the set of SONARs, it is possible to infer the route performed by the vehicle.

Keywords: Vehicles. Arduino. Localization. Sensor. Path Description

LISTA DE FIGURAS

Figura 2.1 – Distância d pelo método ToA	16
Figura 2.2 – Método de trilateração	17
Figura 2.3 – Método <i>Time Difference of Arrival</i> com três nós âncoras	18
Figura 2.4 – Método <i>Angle of Arrival</i> com duas estações	19
Figura 3.1 – Exemplo de descrição de percurso/ <i>tracking</i>	23
Figura 3.2 – Imagem ilustrativa de um cenário para o sistema	24
Figura 3.3 – Foto do SONAR implementado	25
Figura 3.4 – Precisão do sensor ultrassônico HC-SR04	26
Figura 3.5 – Etapas da detecção do veículo pelo SONAR	27
Figura 3.6 – Representação do cálculo para localização do veículo	28
Figura 4.1 – Imagem ilustrativa do cenário do experimento	32
Figura 4.2 – Gráfico de dispersão do experimento estático	34
Figura 4.3 – Gráfico de dispersão do experimento dinâmico	35

LISTA DE TABELAS

Tabela 2.1 – Comparação entre as técnicas <i>range-based</i>	20
Tabela 4.1 – Posições originais do veículo	33

LISTA DE ABREVIATURAS E SIGLAS

<i>AoA</i>	<i>Angle of Arrival</i>
<i>DoA</i>	<i>Direction of Arrival</i>
<i>FSPL</i>	<i>Free Space Path Loss</i>
<i>4WD</i>	<i>Four-Wheel Drive</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>GPS</i>	<i>Global Positioning System</i>
<i>RF</i>	<i>Radio Frequency</i>
<i>RSS</i>	<i>Received Signal Strength</i>
<i>RSSF</i>	<i>Rede de Sensores Sem Fio</i>
<i>SPI</i>	<i>Serial Peripheral Interface</i>
<i>SONAR</i>	<i>Sound Navigation and Ranging</i>
<i>TDoA</i>	<i>Time Difference of Arrival</i>
<i>ToA</i>	<i>Time of Arrival</i>
<i>UWB</i>	<i>Ultra-wide-band</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	MOTIVAÇÃO	10
1.2	DESCRICAÇÃO DO PROBLEMA	10
1.3	OBJETIVOS	11
1.3.1	Objetivo geral	11
1.3.2	Objetivos específicos	11
1.4	ESTRUTURA DO TEXTO	11
2	FUNDAMENTOS E REVISÃO DE LITERATURA	12
2.1	REDES VEICULARES (VANETS)	12
2.2	PLATAFORMA ARDUINO	12
2.3	LOCALIZAÇÃO DE VEÍCULOS	13
2.3.1	Taxonomia de localização	13
2.3.1.1	<i>Localização local e localização global</i>	13
2.3.1.2	<i>Ambientes estáticos e dinâmicos</i>	14
2.3.1.3	<i>Procedimentos ativos e passivos</i>	14
2.3.1.4	<i>Único robô versus múltiplos robôs</i>	15
2.3.2	Técnicas de Localização	15
2.3.2.1	<i>Time of Arrival</i>	16
2.3.2.2	<i>Time Difference of Arrival</i>	17
2.3.2.3	<i>Received Signal Strength</i>	18
2.3.2.4	<i>Angle of Arrival</i>	19
2.3.2.5	<i>Comparação entre as técnicas</i>	19
2.4	TRABALHOS RELACIONADOS	20
2.4.1	Pozyx	20
2.4.2	RADAR	21
2.4.3	<i>A Map Matching Method for GPS Based Real-Time Vehicle Location</i>	21
2.4.4	<i>Automatic Vehicle Location in Cellular Communications Systems</i>	21
3	ELABORAÇÃO DO SISTEMA: INFERINDO O PERCURSO DO VEÍCULO	22
3.1	AUTO-LOCALIZAÇÃO	22
3.2	DESCRIÇÃO DE PERCURSO	22
3.3	ARQUITETURA	23
3.3.1	Cenário	23
3.3.2	SONAR	24
3.3.3	Localização	26
3.3.4	Implementação	28
3.3.4.1	<i>Arduino</i>	28
3.3.4.2	<i>Central de controle</i>	29
4	AValiação EXPERIMENTAL	31
4.1	CONFIGURAÇÃO	31
4.2	AMBIENTE DO EXPERIMENTO	31
4.3	EXPERIMENTO	32
4.3.1	Experimento estático	32
4.3.1.1	<i>Resultados</i>	33
4.3.2	Experimento dinâmico	34
4.3.2.1	<i>Resultados</i>	35

5	CONCLUSÃO	36
5.1	CONSIDERAÇÕES FINAIS.....	36
5.2	TRABALHOS FUTUROS	36
	REFERÊNCIAS BIBLIOGRÁFICAS	37

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Descrição de percurso é um serviço chave no contexto de redes veiculares e serve como base para outros serviços, como navegação e controle estatístico. Na descrição de percurso, a utilização de um sistema de localização é essencial para que seja possível determinar as posições do veículo em diversos instantes de tempo durante o percurso. Esses sistemas de localização precisam estar disponíveis o tempo inteiro e em qualquer lugar.

Sistemas de localização de propósito geral são essencialmente baseados em GPS (*Global Positioning System*) ou em serviços oferecidos por rede de computadores (através de protocolos 3G, 4G e Wi-Fi). Esses sistemas funcionam relativamente bem em locais abertos, onde não existem edificações ou montanhas interferindo e bloqueando o sinal desses sistemas. Já em ambientes fechados, como túneis e estacionamentos, o sinal do GPS é bloqueado e, conseqüentemente, a localização fica indisponível nessas situações.

Uma solução para que a localização esteja disponível em locais fechados é fazer o uso de rede de sensores para calcular e determinar posições no ambiente. Uma rede de sensores é uma infraestrutura composta de elementos capazes de medir o ambiente, computar e se comunicar para observar ou reagir à um determinado evento (SOHRABY; MINOLI; ZNATI, 2007). Com isso, sensores espalhados em um ambiente fechado podem detectar e determinar a localização de veículos.

1.2 DESCRICAO DO PROBLEMA

Serviços de localização, idealmente, deveriam estar disponíveis todo tempo e em qualquer lugar. Porém, a maioria das tecnologias hoje tem limitações. Por exemplo, com o GPS somente é possível localizar objetos em ambientes que possuam disponibilidade de sinal. Isso afeta qualquer tipo de lugar que, de alguma forma, bloqueia esse sinal, como edificações, montanhas ou túneis.

Além disso, muitas vezes é necessário que a localização seja feita usando apenas recursos locais, ou seja, quando é preciso determinar a localização de algum dispositivo em um ambiente específico, aonde o importante são os pontos de referência desse ambiente, isso pode ser exemplificado com a planta de uma edificação, onde coordenadas globais não são importantes mas sim coordenadas locais.

1.3 OBJETIVOS

1.3.1 Objetivo geral

O presente trabalho tem como objetivo contribuir para o desenvolvimento e disponibilidade das redes veiculares com uma implementação de baixo custo e complexidade, visando suprir a necessidade de se ter uma descrição de percurso de veículos em ambientes fechados o que não contam com sistemas de GPS.

Para isso, a plataforma Arduino oferece um ambiente de prototipação de baixo custo para o desenvolvimento desse sistema. Com ela, é possível construir e montar a rede de sensores e também simular a movimentação e descrição de percurso veicular usando como suporte a plataforma Arduino 4WD (*Four-Wheel Drive*).

1.3.2 Objetivos específicos

Mais precisamente, neste trabalho é feito um estudo sobre diferentes técnicas de localização em redes de sensores sem fio (RSSF) e é proposto um sistema de descrição de percurso que foi implementado na plataforma Arduino utilizando sensores ultrassônicos. Enquanto a tecnologia para redes veiculares não é amplamente disponibilizada, plataformas de prototipação, como a Arduino, permitem investigar e avaliar novos serviços.

1.4 ESTRUTURA DO TEXTO

Este trabalho está organizado como segue. O Capítulo 2 apresenta a fundamentação teórica: plataforma Arduino e técnicas de localização, e os trabalhos relacionados. O Capítulo 3 apresenta a arquitetura da solução aqui proposta, de forma detalhada. O Capítulo 4 apresenta a avaliação da solução proposta. O Capítulo 5 apresenta as conclusões e direções futuras.

2 FUNDAMENTOS E REVISÃO DE LITERATURA

Este capítulo trata dos fundamentos teóricos necessários para o entendimento do sistema proposto e do atual estado da arte. A seção 2.1 introduz o conceito de redes veiculares. A seção 2.2 apresenta um breve resumo da plataforma Arduino. A seção 2.3 os conceitos básicos de localização e técnicas comumente usadas na literatura e uma comparação entre elas. A seção 2.4 apresenta os trabalhos relacionados.

2.1 REDES VEICULARES (VANETS)

Redes veiculares ou VANETs (Vehicular Ad Hoc Network) são redes formadas por veículos e por equipamentos espalhados pelas estradas que trocam mensagens entre si. VANETs possuem diversas aplicações na vida real, como controle automático de vias, disseminação de informações sobre acidentes, acionamento automático de freios do veículo, entre outras.

VANET está compreendida no campo de MANET (Mobile Ad Hoc Network), que são redes móveis auto adaptativas, porém com foco nos problemas veiculares. Essa tecnologia vem ganhando bastante atenção ultimamente devido a modernização das vias e o aparecimento de veículos autônomos ou automáticos. Além disso, VANETs podem ser utilizadas como ferramenta para a redução de congestionamentos nas cidades fazendo o controle inteligente de sinalleiras.

2.2 PLATAFORMA ARDUINO

Arduino¹ é uma plataforma eletrônica de código aberto composta por *hardware* e *software* usada para projetar e desenvolver protótipos e produtos eletrônicos de baixo custo. A plataforma segue a filosofia *easy-to-use*, isto é, é feita para que seja de fácil uso e acesso, tanto para iniciantes quanto para pessoas com experiência na área.

O *hardware* consiste em uma placa única, projetada com um microcontrolador Atmel AVR de 8 bits com capacidade de fazer leitura e escrita através de portas analógicas e digitais. O projeto original inclui todas as partes eletrônicas necessárias para que uma placa seja considerada uma placa Arduino e possa se comunicar com o microcontrolador: regulador, cristal de relógio, interface *USB-to-serial* e uma interface programável SPI (*Serial Peripheral Interface*) para substituir o *bootloader* (ARDUINO, 2016).

¹www.arduino.cc

O *software* que compõe a plataforma é a Arduino IDE (*Integrated Development Environment*), que é uma aplicação multiplataforma desenvolvida na linguagem Java, onde é possível escrever, compilar e carregar códigos para sua placa. A Arduino IDE suporta as linguagens C e C++ utilizando algumas regras para organizar o códigos.

A plataforma Arduino é compatível com projetos robóticos, destacando-se a plataforma veicular. A placa Arduino pode ser acoplada a um chassi com 4 rodas e motores individuais para cada roda, assim formando a estrutura de um veículo. A placa faz o controle dos motores que movem as rodas, fazendo assim, o veículo se movimentar. Além disso, outros sensores podem ser usados para aumentar a capacidade de coisas que o veículo pode fazer.

2.3 LOCALIZAÇÃO DE VEÍCULOS

Localização é um dos principais problemas para descrição de percurso de veículos, mas sempre é um problema difícil de ser resolvido. Segundo (THRUN; BURGARD; FOX, 2005), a descrição da taxonomia dos problemas de localização ajuda a compreender as necessidades deste serviço. A seguir, é discutida a taxonomia dos problemas de localização (seção 2.3.1). Na sequência (seção 2.3.2), são apresentadas as diferentes técnicas de localização de objetos.

2.3.1 Taxonomia de localização

Tipicamente a problemática da localização envolve um objeto (robô ou, neste caso, um veículo) e um mapa. O mapa é uma informação conhecida e serve como entrada para a resolução do problema.

2.3.1.1 *Localização local e localização global*

O problema da localização depende do escopo que cada situação, as vezes é necessário saber a posição em termos globais e em outras situações a localização em um lugar específico.

- localização global calcula ou determina uma posição única no globo terrestre que pode ser lida e reconhecida em qualquer lugar. Um exemplo de dispositivo que usa esse tipo de localização é o GPS, que indica uma coordenada global única no mundo inteiro.

- localização local indica uma posição localmente, geralmente necessita de pontos de referências que já tenham uma posição predefinida no ambiente. Esses pontos de referências podem ser mapas ou plantas de um determinado lugar. A localização local pode ser convertida em coordenadas globais.

Neste trabalho, é enfocada a localização local, onde o ambiente possui coordenadas próprias e os pontos de referências são os sensores utilizados, que são conscientes das suas posições.

2.3.1.2 Ambientes estáticos e dinâmicos

Em relação ao ambiente onde os sistemas de localização operam, pode-se classificá-los em dois tipos, estáticos e dinâmicos.

- ambientes estáticos são ambientes que, além do veículo ou objeto a ser localizado, todos os seus elementos são fixos e não sofrem alterações de suas posições. São ambientes em que a disposição de objetos presentes não muda.
- ambientes dinâmicos, por outro lado, são ambientes em que a disposição dos seus objetos ou elementos mudam periodicamente ou constantemente, além do veículo que está se movendo.

Neste trabalho, o ambiente utilizado para o sistema de localização e descrição de percurso proposto é estático, ou seja, além do veículo se movendo, todos os outros objetos do ambiente serão fixos.

2.3.1.3 Procedimentos ativos e passivos

Os procedimentos de localização também diferem na relação da periodicidade em que a localização será feita, podendo ela ser automática ou manual.

- localização ativa se refere à ação do veículo precisar solicitar para o sistema sempre que ele quiser saber sua localização.
- localização passiva é quando a localização é atualizando automaticamente em intervalos de tempos sem a necessidade do veículo solicitar toda vez que ele precisar da sua posição.

Neste trabalho é enfocada a localização passiva, onde o sistema vai atualizando a localização do veículo automaticamente, sendo assim possível fazer a descrição de percurso desse veículo.

2.3.1.4 Único robô versus múltiplos robôs

Um sistema de localização pode ser capaz de operar de forma orgânica, ou seja, os próprios robôs (ou veículos) podem ser usados como parte do sistema para realizar uma operação.

- localização de robô único e sem comunicação é quando o robô consegue calcular sua posição sem receber informações de fontes de externas para isso.
- localização de múltiplos robôs com cooperação é a capacidade do sistema identificar e localizar mais de um robô simultaneamente. Isso pode ser feito com a troca de informações entre os robôs e entre os componentes da infra-estrutura do sistema de forma colaborativa.

Em contraste, neste trabalho é enfocada a localização de apenas um robô, mas com a cooperação de pontos fixos (nós âncoras). Vários sensores irão trabalhar simultaneamente para identificar e atualizar a posição do veículo.

2.3.2 Técnicas de Localização

A maioria dos sistemas de localização veicular hoje em dia são baseados em GPS, onde por meio de sinais de satélites é possível determinar a posição de um dispositivo em qualquer lugar no planeta.

Esses sistemas baseados em GPS, porém, necessitam que o veículo esteja na linha de visada dos satélites para que funcionem plenamente. Essa limitação impede que esses sistemas de localização funcionem em locais fechados, por exemplo, estacionamentos e túneis, no que se diz respeito a localização veicular.

Como alternativa ao GPS, o conceito de rede de sensores sem fio pode ser utilizado para fazer a localização de dispositivos em locais fechados. Em uma RSSF, os sensores espalhados no ambiente que são conscientes de suas posições, chamados nós âncoras, e um nó móvel que desconhece sua posição, que seria o veículo. Uma RSSF também pode ser formada apenas por nós que são âncoras (MAO et al., 2009).

A localização de nós em RSSF é geralmente classificada em *range-based* ou *range-free* (SOHRABY; MINOLI; ZNATI, 2007). A abordagem *range-based* utiliza as distâncias absolutas ou ângulos entre o nó móvel e os nós âncoras. Na abordagem *range-free*, o nó móvel, apenas através da conexão e proximidade com os nós âncoras, consegue estimar regiões ou áreas em que ele está localizado. Este trabalho irá focar apenas nas técnicas *range-based*.

Existem quatro técnicas *range-based* mais conhecidas na literatura, são elas: *Time of Arrival*, *Time Difference of Arrival*, *Received Signal Strength* e *Angle of Arrival*. Essas

técnicas serão discutidas em mais detalhes no texto que segue.

2.3.2.1 Time of Arrival

Time of Arrival (ToA) é uma técnica que usa o tempo de viagem do sinal entre um receptor e o emissor para determinar a distância entre eles. Para isso é necessário saber a velocidade do sinal e os tempos exatos em que o sinal foi enviado e recebido. Essa técnica pode ser aplicada para diversos tipos de sinais, como RF (*Radio Frequency*), acústico, infravermelho, e ultrassônico (HU; CAO, 2010).

Assumindo que no tempo t_0 um sinal é emitido pelo nó móvel, e no tempo t_i o sinal é recebido por um dos nós âncoras. A distância entre os dois pode ser calculada por

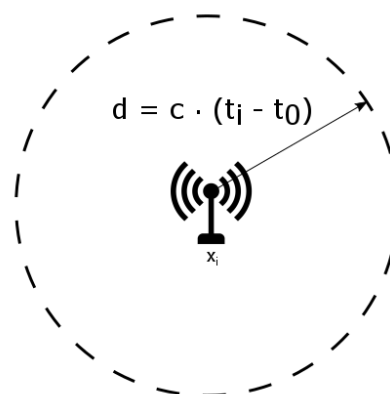
$$d = c \cdot (t_i - t_0), \quad (2.1)$$

onde c é a velocidade de propagação do sinal. Esta equação também pode ser escrita por

$$\|x - x_i\| = c \cdot (t_i - t_0), \text{ para todo } i \in \{1, 2, 3, \dots\} \quad (2.2)$$

Geometricamente, a Equação 2.2 representa um círculo com o centro em x_i e com raio de $c \cdot (t_i - t_0)$, como mostra a Figura 2.1.

Figura 2.1 – Distância d pelo método ToA

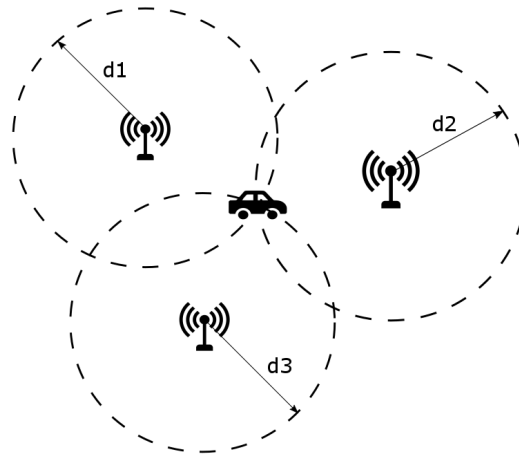


Fonte: Autor.

Para determinar a localização do nó móvel na rede usando TOA, é necessário saber a distância que ele está de pelo menos três nós âncoras, supondo um cenário em duas dimensões (quatro nós âncoras considerando um cenário em três dimensões). Geome-

tricamente, a localização do nó móvel pode ser dada pelo ponto de intersecção entre os círculos formados pelas distâncias entre os nós, essa técnica é chamada de trilateração (multilateração se houver mais de três nós âncoras), como mostra a Figura 2.2.

Figura 2.2 – Método de trilateração



Fonte: Autor.

O maior problema do TOA é a sincronização de tempo entre os nós em uma rede sem fio. Devido à alta velocidade de propagação do sinal e à curta distância entre os nós, qualquer pequena diferença de tempo entre os nós pode gerar erros significativos na precisão nas medidas calculadas.

2.3.2.2 Time Difference of Arrival

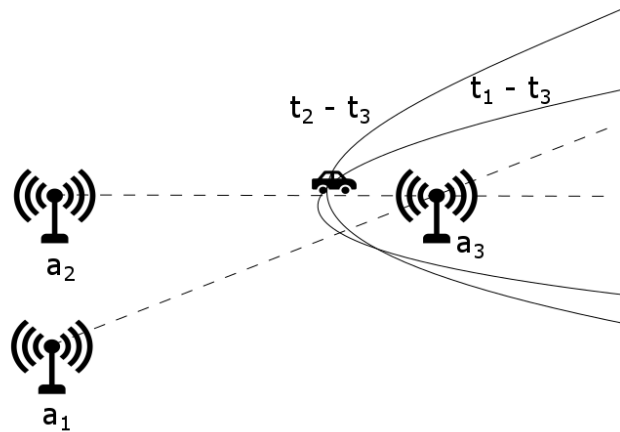
O método *Time Difference of Arrival* (TDoA) calcula a diferença de tempo de voo do sinal entre o nó móvel (emissor) e um par de nós âncoras. Diferentemente do TOA, essa técnica necessita que apenas os nós âncoras estejam com os tempos sincronizados. O nó móvel envia o sinal no instante de tempo t_0 que é recebido pelo nó âncora 1 no tempo t_1 , pelo nó âncora 2 no tempo t_2 , e pelo nó âncora i no tempo t_i . As diferenças de tempo de chegada, nomeadas $(t_2 - t_1)$, $(t_3 - t_1)$, e assim por diante, podem ser escritas como as diferenças de distâncias entre um par de nós e um nó móvel como

$$\|x - x_i\| - \|x - x_j\| = c \cdot (t_i - t_j), \text{ para todo } i, j \in \{1, 2, 3, \dots\} \text{ e } i \neq j. \quad (2.3)$$

Geometricamente, essa equação descreve uma hipérbole que representa as diferenças de tempo de voo entre os nós âncoras i e j . Assim como o TOA, a posição do nó

móvel se dá pela intersecção dessas hipérbolas como mostra a Figura 2.3.

Figura 2.3 – Método *Time Difference of Arrival* com três nós âncoras



Fonte: Autor.

2.3.2.3 Received Signal Strength

Received Signal Strength (RSS) é uma técnica que utiliza a força do sinal recebido para medir distância, a medida que a distância aumenta, a força do sinal diminui correspondentemente. Em teoria, essa técnica é a mais fácil de se implementar, porém a mais difícil de se conseguir resultados precisos.

A técnica RSS é baseada no modelo de *Free Space Path Loss* (FSPL), que é o caminho que um sinal percorre em um espaço vazio sem obstáculos que causem reflexão ou refração. Em um espaço livre, a força do sinal pode ser determinada por

$$P(d) = P_0 - 20 \log_{10} \frac{d}{d_0} \quad (2.4)$$

Depois de calculada a distância entre os nós da rede, o método RSS, assim como o TOA, faz uso do processo de trilateração para determinar a localização.

Em comparação aos métodos TOA e TDOA, o RSS é o mais fácil de ser implementado pois não há necessidade de nenhum tipo de sincronização temporal entre os nós da rede e não necessita de nenhum tipo de *hardware* adicional.

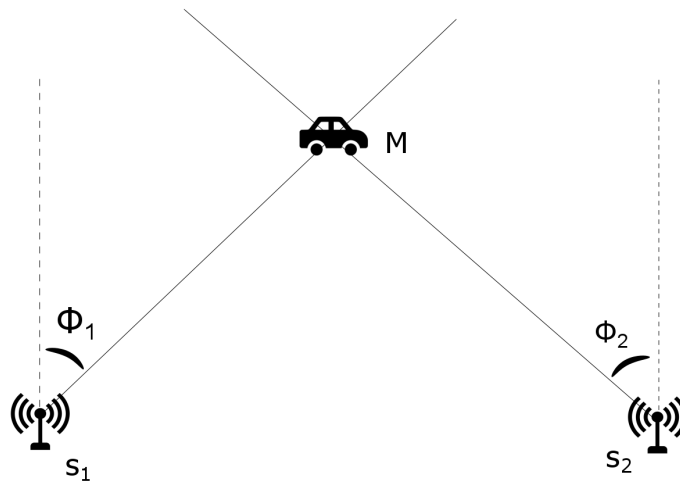
Contudo, o método RSS possui várias desvantagens. A principal delas é que o sinal pode sofrer interferência de diversos fatores do ambiente, como interferência eletromagnética e até mesmo com a umidade do ar.

2.3.2.4 Angle of Arrival

Angle of Arrival (AoA), também conhecida como *Direction of Arrival* (DoA), é uma técnica usada para determinar a direção que um sinal propagado chega em uma estação. Essa estação é composta por um *array* de antenas capazes de receber o mesmo sinal separadamente.

Cada antena presente nesse *array* é posicionada de forma que é possível determinar a diferença de tempo que o sinal chega em cada uma delas e assim calcular a direção do sinal. Com duas estações com as mesmas configurações, posicionadas em locais diferentes, é possível determinar a localização do emissor, que se encontra na intersecção das linhas formadas pelas direções dos sinais.

Figura 2.4 – Método *Angle of Arrival* com duas estações



Fonte: Autor.

A Figura 2.4 demonstra o método AOA com duas estações S_1 e S_2 que, a partir das direções calculadas do emissor (M), é possível inferir os ângulos ϕ_1 e ϕ_2 respectivamente e assim calcular a posição do emissor pela intersecção dos vetores direções formados.

AOA é uma técnica que difere das demais por não usar o sinal emitido para tentar calcular a distância entre nós, mas sim a direção que o sinal chega. A maior desvantagem dessa técnica é o custo de implementação, pois cada estação necessita de diversas antenas para formar o *array*.

2.3.2.5 Comparação entre as técnicas

A Tabela 2.1 apresenta uma comparação entre as técnicas *range-based* vistas. As técnicas estão classificadas em relação à necessidade de sincronização temporal entre

nós da RSSF, complexidade de implementação e custo de implementação.

Tabela 2.1 – Comparação entre as técnicas *range-based*

Técnica	Sincronização	Complexidade	Precisão
<i>Time of Arrival</i>	Necessária	Alta	Alta
<i>Time Difference of Arrival</i>	Necessária entre os nós âncoras	Média	Média
<i>Received Signal Strength</i>	Não necessária	Baixa	Baixa
<i>Angle of Arrival</i>	Não necessária	Alta	Alta

Neste trabalho, é utilizada a técnica de TOA e uma variação de AOA. TOA é utilizada pelos sensores ultrassônicos para a medição de distância. A outra técnica é uma variação da *Angle of Arrival*, pois faz uso do ângulo formado entre o veículo e o nó âncora (sensores ultrassônicos) para calcular a posição.

2.4 TRABALHOS RELACIONADOS

Localização de robôs (objetos móveis) é um assunto que vem ganhando destaque na pesquisa científica. Merecem destaques trabalhos que enfocam localização usando RSSF: (MAREK, 2010), (BISCHOFF et al., 2012), (XU; DING; DASGUPTA, 2011). Porém, é válido destacar quatro projetos: Pozyx, RADAR, *A Map Matching Method for GPS Based Real-Time Vehicle Location* e *Automatic Vehicle Location in Cellular Communications Systems*. Pozyx é sistema de localização em ambientes fechados desenvolvido na plataforma Arduino. RADAR é um trabalho de referência para diversos outros na área.

2.4.1 Pozyx

Pozyx (LABS, 2016) é um sistema de posicionamento para ambiente internos desenvolvido na plataforma Arduino com o objetivo de ser altamente preciso. Através da técnica de TOA e trilateração com nós âncoras, o projeto faz uso da tecnologia de radio UWB (*Ultra-wide-band*) para fazer a comunicação entre os nós e calcular o tempo de voo do sinal. Os produtos desenvolvidos estão à venda no site da empresa com versões para medição de distância, a localização e kit de desenvolvedor.

2.4.2 RADAR

RADAR (BAHL; PADMANABHAN, 2000) é o primeiro sistema de posicionamento para ambientes internos baseado na força do sinal Wi-Fi. A partir de uma rede LAN *Local architecture Network* e a tecnologia de RF *fingerprinting*, o RADAR é capaz de determinar a localização de dispositivos e usuários dentro de edificações. Esse trabalho é base para a maioria dos estudos que utilizam sinais de Wi-Fi para localização em ambientes fechados.

2.4.3 *A Map Matching Method for GPS Based Real-Time Vehicle Location*

Nesse trabalho (JAGADEESH; SRIKANTHAN; ZHANG, 2004), é descrita uma implementação de um método localização em tempo real usando *map matching* com baixo custo de processamento. Map matching é uma técnica que utiliza informações conhecidas do mapa para melhorar a precisão das posições calculadas pelo GPS.

2.4.4 *Automatic Vehicle Location in Cellular Communications Systems*

Nesse trabalho (SONG, 1994), é implementada uma técnica de localização veicular utilizando redes de celulares. Com um aparelho de celular incorporado ao veículo, é possível calcular sua localização usando a atenuação do sinal transmitido, desde que o veículo esteja transitando dentro de uma área com cobertura celular.

3 ELABORAÇÃO DO SISTEMA: INFERINDO O PERCURSO DO VEÍCULO

Este capítulo descreve o processo de desenvolvimento do sistema visando alcançar os objetivos deste trabalho. Este sistema tem o objetivo de apresentar uma solução para a problemática de localização e descrição de percurso veicular utilizando a plataforma de prototipagem Arduino e diferentes sonares. Cada SONAR possui um sensor ultrassônico.

Inicialmente são definidos os conceitos de auto-localização e descrição de percurso. Depois será descrita a arquitetura do sistema elaborado.

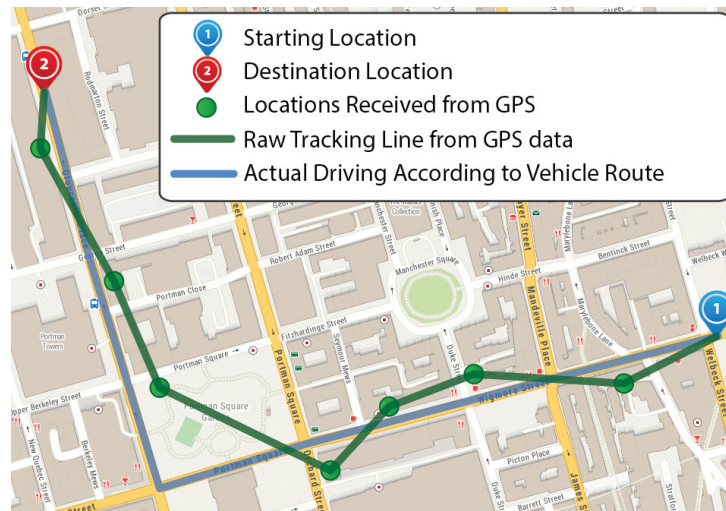
3.1 AUTO-LOCALIZAÇÃO

Auto-localização é a capacidade de um veículo calcular ou solicitar de alguma ferramenta externa sua posição atual em um ambiente local ou global. Essa solicitação pode ser passiva ou ativa, ou seja, é possível solicitar a posição apenas no momento desejado ou solicitar que a posição seja atualizada em um intervalo de tempo constante durante um trajeto. Como visto no Capítulo , existem diversas técnicas que podem ser utilizadas para localização veicular.

3.2 DESCRIÇÃO DE PERCURSO

Descrição de percurso ou *tracking*, pode ser definido como uma sequência de posições em intervalos de tempo a partir de um ponto de partida até um ponto de chegada. Com esses pontos, é possível traçar um caminho feito pelo veículo. É de se esperar que, quanto mais pontos e menor o intervalo de tempo, mais preciso é o traçado do percurso.

Figura 3.1 – Exemplo de descrição de percurso/tracking



Fonte: We-edit (2015).

A Figura 3.1 representa um sistema de descrição de percurso em uma ferramenta que usa GPS. Partindo de um ponto de localização inicial (*Starting Location*), é possível traçar um percurso até um ponto de destino (*Destination Location*) coletando e ligando pontos de localização pelo caminho em diferentes instantes de tempo (*Raw Trackin Line from GPS data*). Nesse exemplo, existem 7 pontos de controle (pontos verdes), onde foi recebida a posição do dispositivo durante o percurso. Aumentando o número de pontos onde o veículo recebe a posição pelo GPS, é possível aumentar a precisão do sistema.

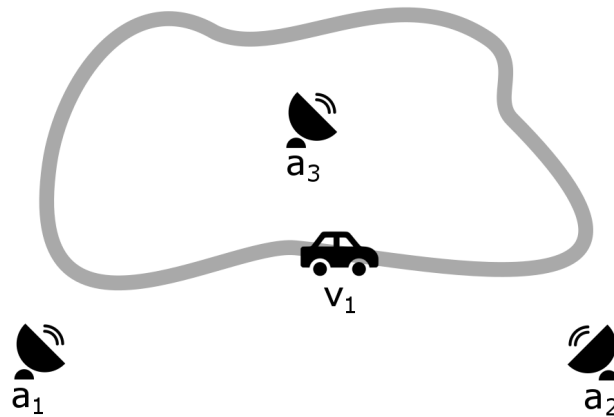
3.3 ARQUITETURA

A arquitetura do sistema desenvolvido pode ser separada em 2 componentes principais, o Arduino que controla o SONAR (*Sound Navigation and Ranging*) implementado e o computador que faz o papel da central de controle.

3.3.1 Cenário

Para que o sistema funcione, é necessário que o ambiente onde ele irá operar seja conhecido. O sistema utiliza de localização local, ou seja, coordenadas válidas apenas para o ambiente e que são conhecidas através pontos de referências, podendo ser um mapa do ambiente ou posições preestabelecidas.

Figura 3.2 – Imagem ilustrativa de um cenário para o sistema



Fonte: Autor.

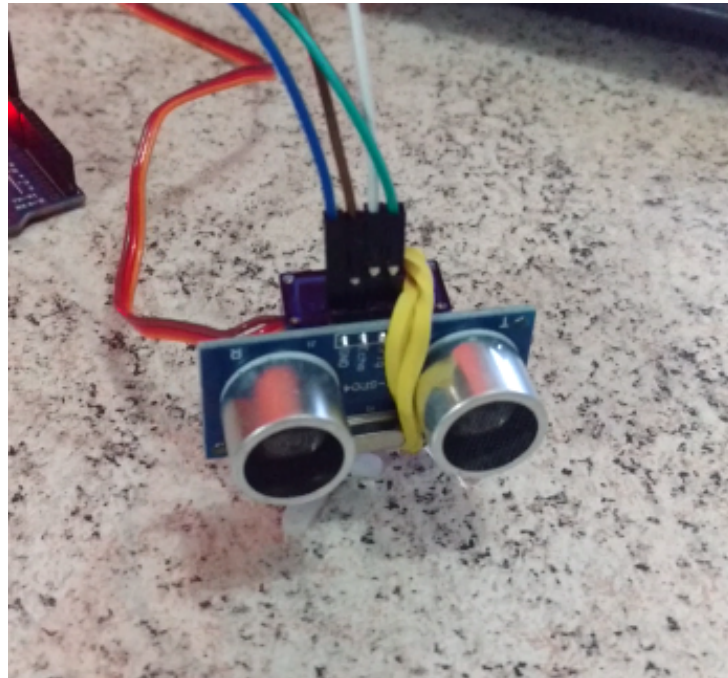
A Figura 3.2 ilustra um cenário para o funcionamento do sistema. Onde os sensores a_1, a_2 e a_3 são conscientes de suas posições e servem como pontos de referência para calcular a posição do veículo v_1 .

3.3.2 SONAR

Sound Navigation and Ranging ou Navegação e Determinação de Distância por Som, é um instrumento que foi inicialmente usado para detecção e localização de submarinos em tempos de guerra.

O SONAR implementado é composto por um sensor ultrassônico HC-SR04 e um servo motor 9g SG90, que são conectados à uma placa Arduino UNO como mostra a Figura 3.3.

Figura 3.3 – Foto do SONAR implementado

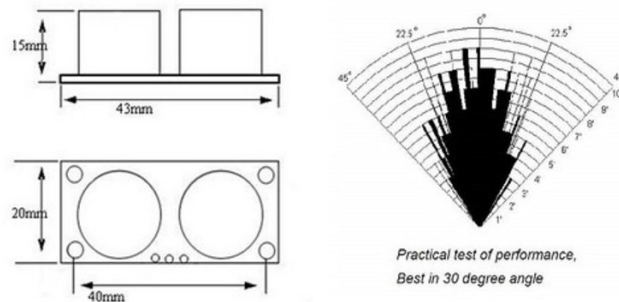


Fonte: Autor.

O sensor ultrassônico é responsável por detectar e calcular a distância de obstáculos na sua frente. Ele emite uma onda sonora que é refletida quando colide com algum obstáculo. A distância é calculada através da medição do intervalo de tempo que a onda sonora é emitida até o momento que ela retorna refletida. Essa técnica é conhecida como *Time of Arrival*, vista na Seção 2.3.2.1, porém, utilizando ondas sonoras. Quando o transmissor e receptor fazem parte do mesmo dispositivo é chamado de monoestático, que é o caso do sensor HC-SR04.

A onda sonora emitida pelo sensor ultrassônico não representa uma linha reta para onde ele está apontando, ela se espalha pelo ambiente em um formato de cone, ou seja, um obstáculo detectado não está necessariamente na linha direta de visão do sensor. O sensor utilizado (HC-SR04) possui uma melhor performance numa abertura de 30 graus (TECHNOLOGIES, 2016), como podemos ver na Figura 3.4.

Figura 3.4 – Precisão do sensor ultrassônico HC-SR04



Fonte: Technologies (2016).

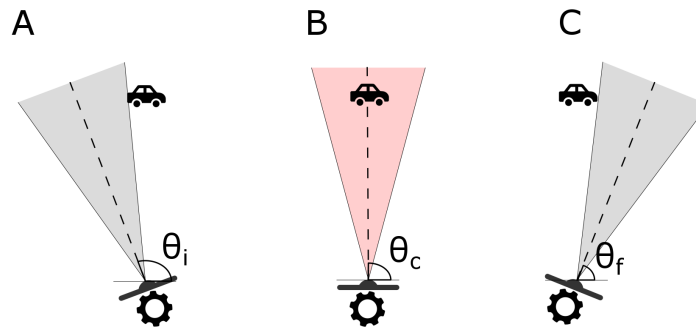
Para que o sistema seja capaz de calcular a posição do veículo, o sensor é acoplado ao servo motor. O servo motor tem a capacidade de rotacionar o sensor em seu próprio eixo, fazendo com que aumente a área de detecção e permitindo calcular a posição do veículo. Através da plataforma Arduino, é possível controlar precisamente qual o grau de rotação que o eixo do servo motor está rotacionado e, conseqüentemente, a direção para qual o sensor está apontando.

Além de controlar o sensor e o motor, o Arduino é responsável por juntar essas informações com a posição do SONAR no ambiente, que é feita de forma manual, e enviar para a computador, que opera como central de controle. A cada rotação do servo motor é enviada uma mensagem contendo essas informações.

3.3.3 Localização

O método de localização proposto usa elementos da técnica de *Angle of Arrival* vista na Seção 2.3.2.4. Porém, nesta implementação a localização do veículo é calculada usando o ângulo formado pela direção em linha reta quando o sensor está apontando para o veículo em relação a um de seus eixos. Diferentemente da técnica de AOA, nesta implementação a distância já é calculada pelo sensor ultrassônico através do método TOA, e não há necessidade da utilização de um vetor de antenas para calcular a direção do sinal. A problemática dessa solução é que, como visto anteriormente, o sensor ultrassônico detecta obstáculos de uma área em formato de cone, impossibilitando determinar se o veículo está absolutamente na frente do sensor.

Figura 3.5 – Etapas da detecção do veículo pelo SONAR



Fonte: Autor.

Para resolver esse problema, é necessário saber quando o centro do cone de visão do sensor está apontado para o veículo. Isso só é possível porque o SONAR implementado não é estático e fica rotacionando em torno do seu eixo com a ajuda do servo motor. A Figura 3.5 ilustra o funcionamento da solução. Na etapa A, o sensor, que está rotacionando no sentido horário, detecta o veículo na extremidade direita do seu campo de visão, quando o centro do campo de visão forma um ângulo θ_i . Então, o sensor irá detectar o veículo até ele sair do seu campo de visão. Na etapa C, o sensor registra o ângulo em que o veículo é detectado pela última vez nessa rotação, que é denotado por θ_f . Agora é possível descobrir o ângulo de rotação do sensor quando ele está objetivamente de frente para o veículo, basta calcular o ângulo que está no meio entre θ_i e θ_f que é ilustrado na etapa B pelo ângulo θ_c .

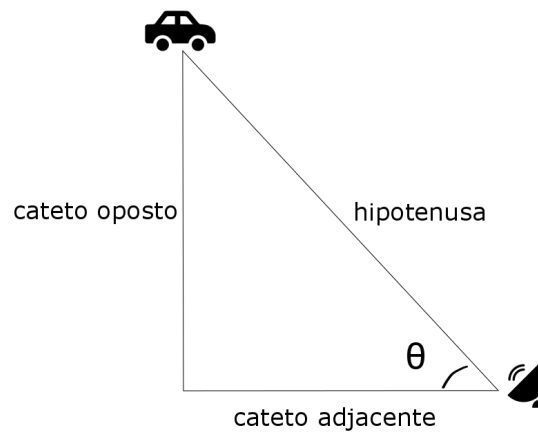
Sabendo o ângulo formado pela linha reta entre o sensor e o veículo, e a distância entre eles, é possível determinar a posição do veículo. O SONAR é consciente de sua posição no ambiente e serve como referência para a posição do veículo. Essa posição é calculada resolvendo as equações abaixo

$$\cos \theta = \frac{\textit{cateto adjacente}}{\textit{hipotenusa}} \quad (3.1)$$

$$\sin \theta = \frac{\textit{cateto oposto}}{\textit{hipotenusa}} \quad (3.2)$$

Nas Equações 3.1 e 3.2, θ representa o ângulo formado pela linha entre o veículo e o sensor, e a *hipotenusa* representa a distância entre eles. Resolvendo as equações, temos que os valores de *cateto adjacente* e *cateto oposto*, representam as distâncias nas coordenadas x e y, respectivamente, que o veículo está do sensor. Como a posição do SONAR é conhecida, basta somar com os valores calculados para descobrir a posição do veículo. A Figura 3.6 representa esse cálculo.

Figura 3.6 – Representação do cálculo para localização do veículo



Fonte: Autor.

3.3.4 Implementação

A implementação do sistema descrito pode ser dividida entre a parte do Arduino e da central de controle. O programa carregado na placa Arduino tem as funções de medir as distâncias calculadas pelo sensor, rotacionar o motor em um arco de rotação controlado, concatenar essas informações e enviá-las em formato de mensagem para a central de controle. Já o programa que roda na central de controle tem as funções de ler as informações passadas pelo Arduino, interpretá-las, calcular a posição do veículo e salvar essas posições em um arquivo.

3.3.4.1 Arduino

Um código escrito para a plataforma Arduino é estruturado de maneira que é preciso estabelecer duas funções:

- *setup()*: essa função é responsável pela configuração inicial do sistema. É onde são definidas as portas usadas na placa e estados iniciais dos componentes do sistema.
- *loop()*: é a função que é chamada sequencialmente durante toda a execução do algoritmo e é onde o código principal deve estar. É responsável por alterações de estados dos componentes, atualização de variáveis e chamada de outras funções.

No sistema implementado, a função *setup()* é responsável pela configuração inicial do servo motor e do sensor ultrassônico. São definidas as portas que cada um está conectado à placa.

O corpo principal do sistema está localizado na função *loop()*. Nessa função é definido como o servo motor e o sensor irão se comportar. Em primeiro lugar, é criado um laço de repetição do tipo *for* que servirá para definir os graus de rotação do servo motor. Os intervalos desse laço são definidos pelos graus mínimo e máximo que o servo motor irá rotacionar. Para cada iteração desse laço, é calculada a distância lida pelo sensor ultrassônico naquele grau de rotação do servo motor. Essas informações são concatenadas em forma de uma *string*, separadas por vírgulas e enviadas para a central de controle via portas seriais. A cada iteração do laço, é enviada uma mensagem.

3.3.4.2 Central de controle

A central de controle é a parte do sistema que processa as informações enviadas pelo Arduino. A central foi implementada usando a linguagem C#, que possui uma biblioteca para operações com portas seriais simples e robusta.

Primeiro, é criada uma *Thread* de execução para cada placa Arduino conectada com o computador. Essa *Thread* é responsável por ler a mensagem recebida do Arduino e em seguida calcular a posição do veículo, caso ele tenha sido detectado.

O processo de detecção do veículo mostrado na Figura 3.5 é feito como mostra o algoritmo abaixo.

Algorithm 1 Valor Absoluto

```

1: função THREAD(x)
2:   se distancia < LIMITE então
3:     detectado.Add(distancia, angulo)
4:   senão
5:     se distancia > LIMITE && detectado.lenght > MARGEM então
6:       centro = detectado[detectado.lenght/2]
7:       detectado.clear()
8:     fim se
9:   fim se
10: fim função

```

O Algoritmo 1 descreve como é calculado o ângulo central formado quando o sensor está apontando para o veículo. Se a distancia recebida pelo Arduino é menor que um certo *LIMITE*, é adicionada no vetor *detectado*, juntamente com o ângulo também recebido pelo

Arduino. Caso a distancia recebida for menor que um *LIMITE*, é verificado se o tamanho do vetor é maior que uma certa *MARGEM*, isso é feito para garantir que o objeto detectado é grande suficiente para ser um veículo, e evita falsos positivos ou ruídos detectados pelo sensor. Se o tamanho do vetor é maior que essa *MARGEM*, a posição do meio do vetor é considerada como o centro do veículo. Depois de descoberto o ângulo correto e o distância relativa à este ângulo, é calculada posição do veículo resolvendo as Equações 3.1 e 3.2. Após calculada, a posição do veículo ocorre a escrita em um arquivo de texto. Esse arquivo é usado para fazer a descrição de percurso do veículo a partir das posições detectadas.

4 AVALIAÇÃO EXPERIMENTAL

Para avaliar o sistema implementado de acordo com o Capítulo 3, foram feitos experimentos em um ambiente controlado com objetivo de comparar os resultados inferidos pelo sistema com as medições feitas manualmente.

4.1 CONFIGURAÇÃO

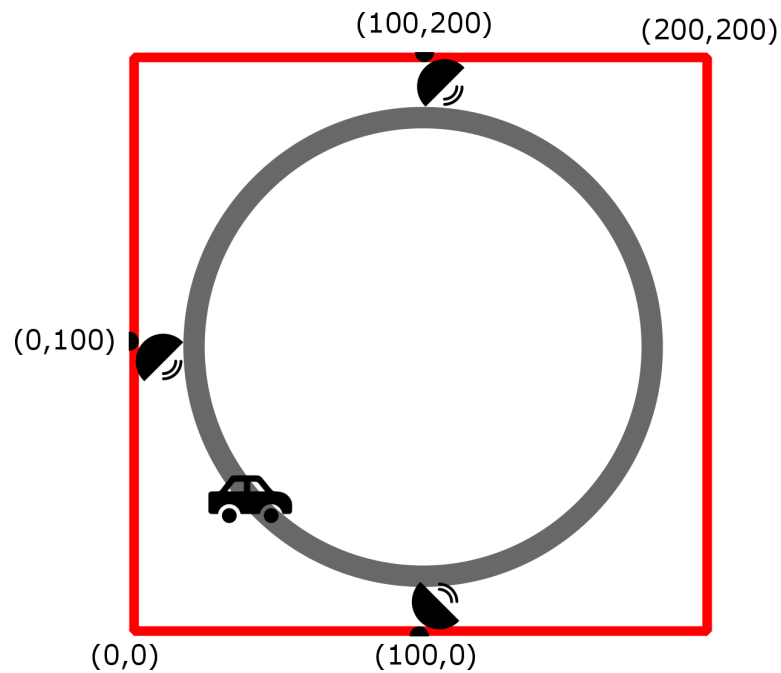
Para a realização do experimento foi usado um computador com as seguintes configurações: processador Intel (R) Core(TM) i5-2450M CPU @ 2.50GHz e 6,00 GB RAM de memória. Rodando em um sistema operacional Windows 7 Home Premium de 64 bits. Foi utilizada a versão do Visual Studio Community 2013 e da Arduino IDE 1.6.8. Esse computador fez o papel de central de controle.

A placa Arduino utilizada foi modelo UNO. Os sensores ultrassônicos foram os HC-SR04 e o servo motor foi o 9g SG90. Para conectar a placa ao computador foram utilizados cabos do tipo USB-A/B.

4.2 AMBIENTE DO EXPERIMENTO

O ambiente usado no experimento consiste em uma área quadrada delimitada por fitas com as dimensões de 200cm x 200cm, e uma pista no interior dessa área, também delimitada por fitas, a qual o veículo irá percorrer, como mostra na Figura 4.1.

Figura 4.1 – Imagem ilustrativa do cenário do experimento



Fonte: Autor.

4.3 EXPERIMENTO

Foram realizados dois tipos de experimentos no sistema proposto, experimento estático e experimento dinâmico. No primeiro, o veículo é posicionado em posições predefinidas de forma manual na área delimitada. No experimento dinâmico o carro está em movimento percorrendo a pista demarcada.

Nos dois experimentos realizados a distribuição dos SONARs pelo ambiente foi feita da seguinte maneira: SONAR 1 na posição (100,0), SONAR 2 na posição (0,100) e SONAR 3 na posição (100,200). Os três SONARs foram posicionados de maneira que o ângulo 0 coincida com a linha sobre qual os três estão posicionados.

4.3.1 Experimento estático

Nesse experimento, o veículo foi posicionado manualmente em oito posições diferentes dentro da área delimitada. O experimento visava medir a precisão das posições do veículo calculadas pelos SONARs e compará-las com as posições originais. As posições

do veículo foram definidas de acordo com a Tabela 4.1 e representam coordenadas que estão sobre a pista.

Tabela 4.1 – Posições originais do veículo

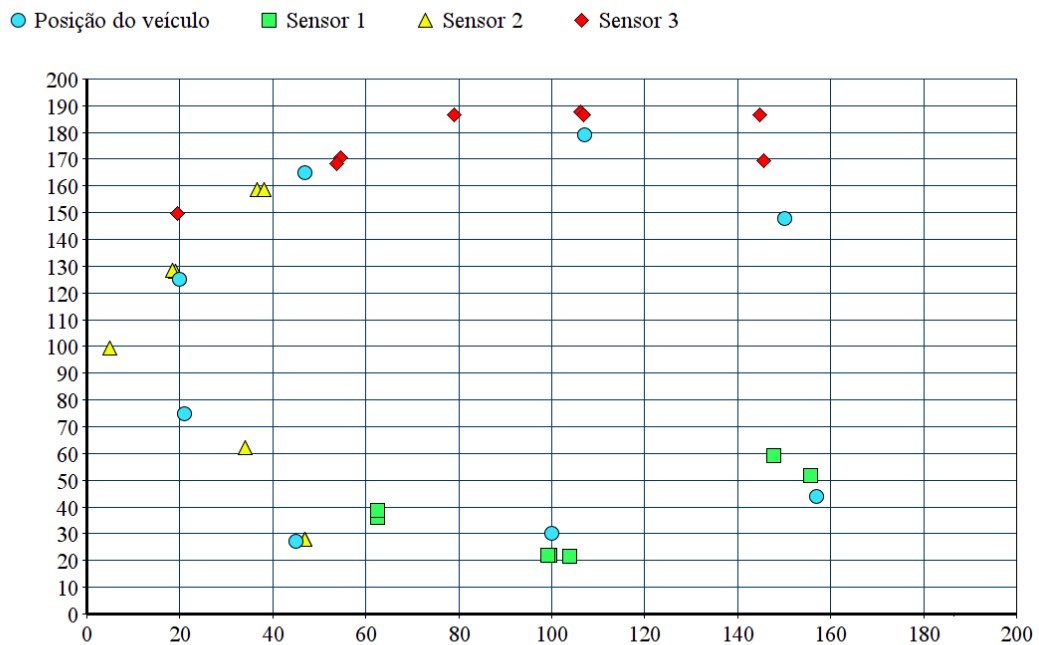
Posição	Eixo X	Eixo Y
Posição 1	21.0	75.0
Posição 2	45.0	27.0
Posição 3	100.0	30.0
Posição 4	157.0	44.0
Posição 5	150.0	148.0
Posição 6	107.0	179.0
Posição 7	47.0	165.0
Posição 8	20.0	125.0

Essas posições são referentes ao centro do veículo, que possui dimensões de 20cm x 20cm, portanto, é necessário considerar uma margem de 10cm em cada eixo para as posições calculadas pelo sistema.

4.3.1.1 Resultados

Os resultados do experimento feito com o veículo estático nas posições descritas acima geraram o gráfico mostrado na Figura 4.2.

Figura 4.2 – Gráfico de dispersão do experimento estático



Fonte: Autor.

O gráfico contém as posições originais do veículo e as posições calculadas pelos três SONARs posicionados pelo cenário. Considerando a margem de 10cm em relação às dimensões do veículo, os resultados foram satisfatórios. Analisando os pontos no gráfico, é possível traçar um trajeto que se aproxima muito da pista desenhada no cenário. Para um sistema de descrição de percurso, esses pontos são suficientes para que, com um auxílio de um mapa, traçar um caminho percorrido pelo veículo. Os pontos que se distanciam muito das posições originais do veículo são efeitos de ruídos causados pelo ambiente.

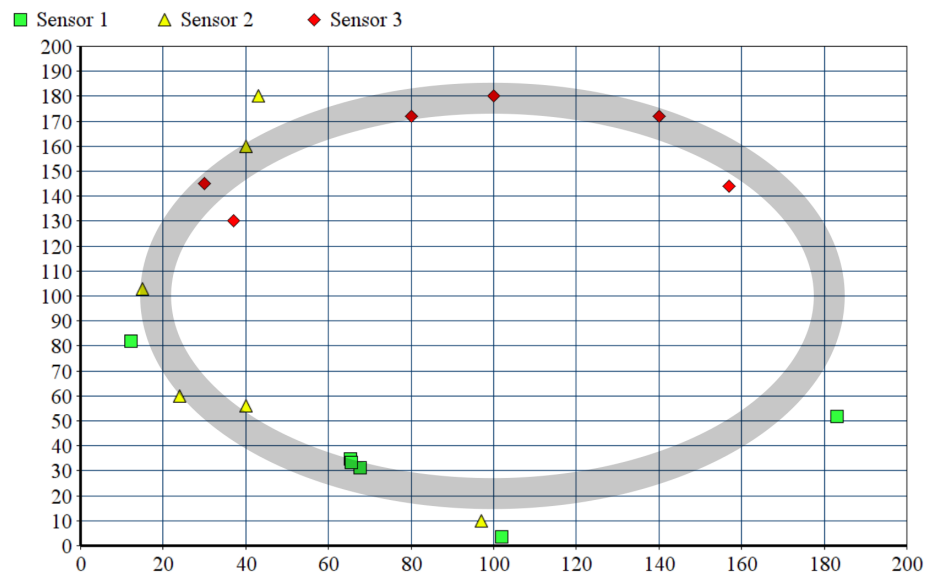
4.3.2 Experimento dinâmico

O experimento dinâmico foi feito com o carro em movimento, percorrendo a pista feita dentro do ambiente 200x200. Os SONARs foram posicionados nas mesmas posições do experimento estático e o carro utilizado foi o mesmo. Nesse experimento o carro percorreu a pista 3 vezes.

4.3.2.1 Resultados

Os resultados do experimento realizado com o veículo se movimentando pela pista pode ser representado pelo gráfico mostrado na Figura 4.3.

Figura 4.3 – Gráfico de dispersão do experimento dinâmico



Fonte: Autor.

Analisando esse gráfico, é possível notar que as posições calculadas pelos SONARs são muito próximas da pista. Se considerar a margem relativa ao veículo, os resultados apresentados podem ser facilmente posicionados dentro da pista.

Mesmo com os desvios na detecção de valores calculados pelo sistema no experimento feitos, muitas aplicações na área de VANTES não necessitam de uma alta taxa de precisão. Para a descrição de percurso, esses valores são suficientes para que se possa projetar o trajeto do veículo.

5 CONCLUSÃO

5.1 CONSIDERAÇÕES FINAIS

Este trabalho tinha como objetivo apresentar técnicas de descrição de percurso e propor uma implementação utilizando a plataforma Arduino. Além disso, também tinha como objetivo ajudar a desenvolver a área de VANETs.

Um dos principais problemas com os sistemas de descrição de percurso e localização existentes é que eles não funcionam plenamente ou totalmente em ambientes fechados. Foi apresentada uma implementação que contempla essa necessidade. Utilizando a plataforma Arduino foi descrito um sistema que, usando sensores de baixo custo, é capaz de detectar e localizar veículos em ambientes fechados.

A plataforma Arduino foi essencial para o desenvolvimento do sistema. A implementação do sistema de localização foi feita em cima da plataforma e ainda foi possível realizar experimentos com um veículo em um ambiente controlado graças a plataforma Arduino 4WD.

5.2 TRABALHOS FUTUROS

A tendência é que a produção de veículos autônomos e automatizados cresça com o passar dos anos. Com isso, a área de pesquisa em VANETs tende a crescer junto. Existem muitos desafios a serem enfrentados nessa área como o aprimoramento da comunicação intraveicular, controle automatizados de semáforos, localização em tempo real em locais fechados, estacionamento automatizado, entre outros. Utilizando a plataforma 4WD também é possível desenvolver a pesquisa na área de automatização veicular.

A partir deste trabalho, existe a possibilidade do desenvolvimento e implementação de outras técnicas de localização e descrição de percurso utilizando a plataforma Arduino e compará-las em questão de custo, precisão, disponibilidade e complexidade. Utilizando a plataforma 4WD também é possível investir em técnicas de automatização veicular.

REFERÊNCIAS BIBLIOGRÁFICAS

ARDUINO. **What makes an Arduino board an Arduino?** 2016. Disponível por WWW em <https://www.arduino.cc/en/Main/Policy>, Acesso em 20/10/2016.

BAHL, P.; PADMANABHAN, V. N. Radar: an in-building rf-based user location and tracking system. In: **INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.** [S.l.: s.n.], 2000. v. 2, p. 775–784 vol.2. ISSN 0743-166X.

BISCHOFF, O. et al. Design and implementation of an ultrasonic localization system for wireless sensor networks using angle-of-arrival and distance measurement. **Procedia Engineering**, v. 47, p. 953 – 956, 2012. ISSN 1877-7058. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877705812043676>>.

HU, F.; CAO, X. **Wireless Sensor Networks: Principles and Practice.** 1st. ed. Boston, MA, USA: Auerbach Publications, 2010. ISBN 1420092154, 9781420092158.

JAGADEESH, G.; SRIKANTHAN, T.; ZHANG, X. A Map Matching Method for GPS Based Real-Time Vehicle Location. **Journal of Navigation**, v. 57, p. 429–440, Set 2004.

LABS, P. **Pozyx.** 2016. Disponível por WWW em <https://www.pozyx.io/>, Acesso em 20/10/2016.

MAO, G. et al. **Localization Algorithms and Strategies for Wireless Sensor Networks.** Hershey, PA: Information Science Reference - Imprint of: IGI Publishing, 2009. ISBN 1605663964, 9781605663968.

SOHRABY, K.; MINOLI, D.; ZNATI, T. **Wireless Sensor Networks: Technology, Protocols, and Applications.** [S.l.]: Wiley-Interscience, 2007. ISBN 0471743003.

SONG, H.-L. Automatic vehicle location in cellular communications systems. **IEEE Transactions on Vehicular Technology**, v. 43, n. 4, p. 902–908, Nov 1994. ISSN 0018-9545.

TECHNOLOGIES, C. **Manual HC-SR04.** 2016. Disponível por WWW em https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB_x2qR4vP8saG73rE/edit, Acesso em 06/12/2016.

THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic robotics.** Cambridge (Mass.) (London): the MIT Press, 2005. ISBN 0-262-20162-3.

WE-EDIT. **How to create an exact tracking route from GPS data with animated icon?** 2015. Disponível por WWW em <http://www.we-edit.de/gis/question/how-to-create-an-exact-tracking-route-from-gps-data-with-animated-icon-150967.html>, Acesso em 08/12/2016.

XU, E.; DING, Z.; DASGUPTA, S. Source localization in wireless sensor networks from signal time-of-arrival measurements. **IEEE Transactions on Signal Processing**, v. 59, n. 6, p. 2887–2897, Junho 2011. ISSN 1053-587X.