

Samuel Pivatto Cavilha

**MEDIÇÃO DE DISTÂNCIA RELATIVA UTILIZANDO
TECNOLOGIA UWB**

Trabalho de Conclusão de Curso
(TCC) submetido ao Curso de
Graduação em Engenharia Elétrica da
Universidade Federal de Santa
Catarina como parte dos requisitos
para obtenção do grau de Bacharel em
Engenharia Elétrica

Orientador: Prof. Richard Demo
Souza, Dr.

Florianópolis
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Cavilha, Samuel
MEDIÇÃO DE DISTÂNCIA RELATIVA UTILIZANDO TECNOLOGIA UWB
/ Samuel Cavilha ; orientador, Richard Demo Souza, 2020.
58 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia Elétrica, Florianópolis, 2020.

Inclui referências.

1. Engenharia Elétrica. 2. Ultra Banda Larga. 3.
Distâncias relativas. 4. Segurança. I. Demo Souza, Richard.
II. Universidade Federal de Santa Catarina. Graduação em
Engenharia Elétrica. III. Título.

Samuel Pivatto Cavilha

**MEDIÇÃO DE DISTÂNCIA RELATIVA UTILIZANDO
TECNOLOGIA UWB**

Este Trabalho de Conclusão de Curso (TCC) foi julgado adequado para obtenção do Título de Bacharel em Engenharia Elétrica e aprovado, em sua forma final, pela Banca Examinadora

Florianópolis, 10 de dezembro de 2020.

Prof. Jean Vianeí Leite, Dr.
Coordenador do Curso de Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Richard Demo Souza, Dr.
Orientador
Universidade UFSC

Prof. Bartolomeu Ferreira Uchôa Filho, Dr.
Universidade UFSC



Prof. Glauber Brante, Dr.
Universidade UTFPR

Este trabalho é dedicado aos meus colegas de classe e aos meus queridos pais.

AGRADECIMENTOS

Gostaria de agradecer à empresa Khomp, que possibilitou um ambiente de pesquisa, incentivo e desenvolvimento para a realização deste trabalho, em especial Heriberto, por ter acompanhado e instruído de perto este trabalho. Agradeço também ao professor Richard, por toda a sua disponibilidade e conhecimento colocado à disposição para o andamento do trabalho.

RESUMO

Neste documento será proposta uma implementação de um sistema para medir distâncias relativas entre dispositivos dentro de uma rede. A tecnologia utilizada para a realização do trabalho foi ultra banda larga, e será apresentado e descrito neste documento as principais características desta tecnologia, os algoritmos utilizados para realizar o cálculo de distâncias, o protocolo de comunicação implementado e algumas funções adicionais. Ao final são apresentados os resultados obtidos juntamente com uma análise, é mostrada a diferença entre os resultados obtidos com a utilização de dois algoritmos distintos e são também propostos trabalhos futuros relacionados ao projeto.

Palavras-chave: UWB. Distância relativa. Segurança.

ABSTRACT

In this document we propose an implementation of a system to measure relative distances between devices inside a network. The technology used will be ultra-wideband (UWB), and in this document we present the main characteristics of this technology, the algorithms used to calculate de distances, the communication protocol implemented and a few additional functionalities. In the end the results will be presented together with a analysis, and the difference between two different algorithms is shown, as well as related future work.

Keywords: UWB. Relative distance. Safety.

LISTA DE FIGURAS

Figura 1 – <i>RTLS</i> funcionando com algoritmo baseado em <i>ToA</i> .	28
Figura 2 - <i>RTLS</i> funcionando com algoritmo baseado em <i>TDoA</i>	29
.....	29
Figura 3 - Faixa de frequência de operação das principais tecnologias de rádio frequência.....	34
Figura 4 - Resumo do conteúdo do kit MDEK1001.....	34
Figura 5 - Esquemático do módulo DMW1001.....	35
Figura 6 - Esquema básico de <i>TWR</i>	36
Figura 7 - Esquema do <i>TWR</i> com resposta adicional.....	37
Figura 8 - Exemplo de um sistema com o protocolo proposto ...	39
Figura 9 - Exemplo do protocolo de comunicação implementado	40
.....	40
Figura 10 - Base de uma mensagem <i>UWB</i>	41
Figura 11 - Transmissão em um ambiente sem linha de visada .	42
Figura 12 - Exemplo dos dados armazenados no acumulador do DW1000.....	43
Figura 13 - Valores do acumulador sob um cenário com obstrução e reflexões.....	44
Figura 14 - Comparação do resultado filtrado contra o resultado normal.....	46
Figura 15 - Diagrama de funcionamento da função de baixo consumo.....	47
Figura 16 - Fluxograma resumido de funcionamento do programa	49
.....	49
Figura 17 - Ambiente de medição.....	50
Figura 18 - Distância medida em um teste inicial.....	51
Figura 19 - Medição estática de 1m.....	51
Figura 20 - Medição estática de 2m.....	52
Figura 21 - Medição estática de 3m.....	52
Figura 22 - Medição estática de 4m.....	53
Figura 23 - Comparação <i>TWR</i> com <i>DS-TWR</i> e 20ms de intervalo	54
.....	54
Figura 24- Comparação <i>TWR</i> com <i>DS-TWR</i> e 50ms de intervalo	55
.....	55
Figura 25 - Comparação <i>TWR</i> com <i>DS-TWR</i> e 100ms de intervalo	55
.....	55

LISTA DE TABELAS

Tabela 1 - Estrutura da mensagem.....	41
Tabela 2 - Análise dos resultados obtidos	53
Tabela 3 - Resultado da comparação entre <i>TWR</i> e <i>DS-TWR</i>	56

LISTA DE ABREVIATURAS E SIGLAS

UWB – Ultra Wide Band

TWR – Two Way Ranging

AoA – Angle of Arrival

TDoA – Time-difference of Arrival

RSS – Received Signal Strength

RTLS – Real Time Location System

DS-TWR – Double-Sided Two Way Ranging

ID – Número de identificação

SUMÁRIO

1	INTRODUÇÃO	27
1.1	SISTEMA DE LOCALIZAÇÃO EM TEMPO REAL	27
1.2	OBJETIVOS	30
1.2.1	Objetivo geral	30
1.2.2	Etapas de Desenvolvimento	30
2	TECNOLOGIA ULTRA BANDA LARGA.....	33
2.1	KIT MDEK1001	34
2.2	ALGORITMO DE CÁLCULO UTILIZADO	35
2.3	PROTOCOLO DE COMUNICAÇÃO IMPLEMENTADO.	38
2.4	ESTRUTURA DA MENSAGEM.....	41
2.5	ALGORITMO PARA DETECÇÃO DE LINHA DE VISADA 42	
2.6	FILTRO 1EURO	45
2.7	FUNÇÕES ADICIONAIS IMPLEMENTADAS	46
2.8	FLUXOGRAMA E FUNCIONAMENTO DO CÓDIGO	47
3	RESULTADOS OBTIDOS	50
3.1	COMPARAÇÃO DO ALGORITMO DE <i>TWR</i> COM O ALGORITMO <i>DS-TWR</i>	54
4	CONCLUSÃO	57
	REFERÊNCIAS	59

1 INTRODUÇÃO

1.1 SISTEMA DE LOCALIZAÇÃO EM TEMPO REAL

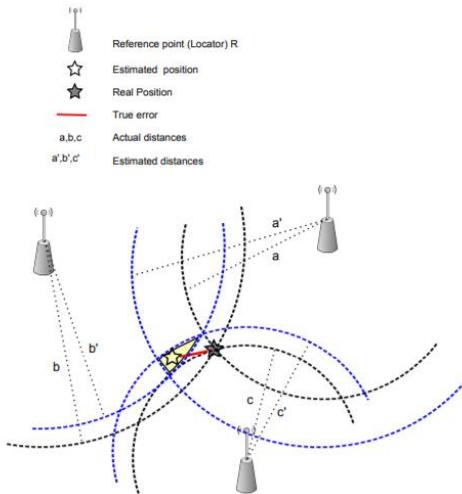
Um sistema de localização em tempo real (*Real Time Location System – RTLS*) é um sistema que tem como objetivo determinar a localização de uma pessoa ou objeto. Esses sistemas não estão ligados a nenhuma tecnologia específica, mas se adequam utilizando a que melhor se encaixa nas necessidades de cada situação. Um sistema *RTLS* é composto primariamente de dois tipos de dispositivos, as âncoras, que são estacionárias, e de posição conhecida, e as etiquetas, ou *tags*, que são os dispositivos que se deseja localizar [Clarinox Technologies Pty Ltd, 2009].

Dentro da tecnologia Ultra Banda Larga (*Ultra Wide Band – UWB*) foram desenvolvidos diferentes algoritmos para se obter a localização, que podem ser classificados dentro seguintes categorias [Alarifi et al. 2016]:

- a. *Time of Arrival (ToA)*
- b. *Time-difference of Arrival (TDoA)*
- c. *Angle of Arrival (AoA)*
- d. *Received Signal Strength (RSS)*
- e. *Hybrid Algorithm*

Os algoritmos de *ToA* se baseiam na medição de distâncias relativas entre uma ou mais *tags* e as âncoras do sistema, como mostra a Figura 1. Essa medição é feita através da estimação do tempo que uma mensagem demora para percorrer o caminho entre o *tag* e a âncora. De posse deste valor é possível obter uma estimativa da distância multiplicando-o pela velocidade da onda no ar (velocidade da luz).

Figura 1 – *RTLS* funcionando com algoritmo baseado em *ToA*

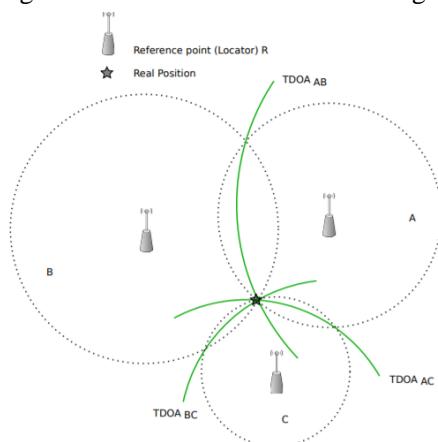


Fonte: Alarifi et al. 2016

A partir das distâncias relativas entre uma *tag* e algumas âncoras, é possível traçar círculos, através dos quais se obtém a estimativa da posição da *tag* no espaço. Para se obter a posição em um ambiente de três dimensões, é necessário saber a distância da *tag* em relação à quatro âncoras diferentes. Estes algoritmos possuem a melhor acurácia e são de fácil implementação, mas necessitam de um número maior de mensagens trocadas, aumentando o tempo de ocupação do canal de transmissão do sistema e consumo de bateria.

Já o algoritmo de *TDoA* funciona através da diferença de tempo de recepção das mensagens transmitidas pela(s) *tag(s)* registradas pelas âncoras do sistema. Cada âncora recebe a mensagem transmitida e carimba o tempo de recebimento. Sabendo a diferença de tempo entre as recepções das âncoras, é possível traçar hipérbolas, que, na sua intersecção, resultarão na estimativa da posição da *tag*, como ilustra a Figura 2.

Figura 2 - RTLS funcionando com algoritmo baseado em TDoA



Fonte: Alarifi et al. 2016

Estes algoritmos necessitam uma sincronização dos *clocks* das âncoras, para que a estimativa seja correta, o que acarreta um sistema mais complexo, ou na necessidade da aplicação de algoritmos de correção para melhorar as estimativas. Por outro lado, estes algoritmos tendem a possuir ótima autonomia de bateria, por necessitarem apenas de transmissões esporádicas da *tag*.

Nos algoritmos baseados em *RSS*, a *tag* mede a força do sinal recebido de diversos transmissores com posição conhecida, e utiliza esses dados para estimar sua distância em relação aos transmissores. Algoritmos baseados em *RSS* não são tão comuns em sistemas para localização em ambientes fechados, por possuírem alta sensibilidade a reflexões e atenuações, mas, por possuírem a característica de a *tag* funcionar apenas como receptora de mensagem, possuem utilidade em algumas situações específicas.

Finalmente, nos algoritmos baseados em *AoA* a direção de chegada de uma mensagem é estimada através de um conjunto de antenas nas âncoras, pela comparação do valor da portadora, ou da amplitude do sinal. Por dependerem de um vetor de antenas e de calibrações muito específicas, os algoritmos baseados em *AoA* possuem uma alta sensibilidade a diversos fatores, o que os torna de uma implementação mais complexa do que os outros métodos.

Neste projeto não será desenvolvido um *RTLS* especificamente, mas será utilizado um algoritmo dentro de categoria de *ToA* conhecido

como *TWR*, a ser apresentado, para realizar a medição relativa de dois dispositivos, porém com a aplicação de algumas variantes que serão apresentadas posteriormente.

1.2 OBJETIVOS

1.2.1 Objetivo geral

O objetivo do projeto é desenvolver um sistema de medição relativa de distância entre *tags*, utilizando a tecnologia Ultra Banda Larga. Este tema surge com o objetivo de responder duas necessidades muito importantes atualmente: a necessidade de distanciamento social, visto a atual pandemia na qual nos encontramos, e distanciamento seguro de máquinas em um ambiente de obras. O distanciamento social é algo que todos nós estamos vivenciando neste período da história, e este trabalho pode contribuir para este controle, principalmente no meio industrial, conseguido realizar este controle em tempo real. Já no ambiente de obras, a distância segura entre máquinas é algo de extrema gravidade, e causa de muitos acidentes, resultando até mesmo na morte de trabalhadores. Com o trabalho proposto, é possível manter as máquinas a uma distância segura, evitando que este tipo de situação possa ocorrer.

A ideia para a realização deste projeto surgiu de uma necessidade de uma empresa relacionada à segurança em seus campos de obras. Muitos acidentes vinham acontecendo, e surgiu a necessidade da implementação de um projeto de monitoramento dos funcionários e das máquinas, para garantir o distanciamento seguro e assegurar que os funcionários estivessem fora das zonas de perigo. Dentro deste ambiente, um dos pontos levantados seria a necessidade de manter uma distância mínima entre as máquinas, demanda que pode ser atendida com a implementação do projeto desenvolvido.

1.2.2 Etapas de Desenvolvimento

Para se alcançar o obtido geral proposto, o projeto foi dividido em pequenas etapas, para facilitar o andamento e a execução do mesmo. A etapa inicial é realizar uma revisão teórica sobre as técnicas de *RTLS*, quais as suas vantagens, desvantagens, assim como qual seria a mais factível com o que estava disponível. Após escolhida a técnica e a tecnologia a serem utilizadas, o próximo passo é o desenvolvimento de um protocolo de comunicação para possibilitar a troca de mensagens

entres os módulos, que foi desenvolvido visando a proposta de realizar um *broadcast*, ou seja, para que não ocorra uma comunicação módulo-a-módulo, para que os módulos possam entrar e sair da rede sem interrupção e para facilitar a comunicação e estruturação do código a ser desenvolvido. Em seguida é desenvolvida a estrutura da mensagem a ser transmitida, ponto importante para reduzir a quantidade de dados enviados, e para que haja uma correta comunicação entre os módulos. Para aprimorar o projeto, ao final é aplicado o filtro 1Euro, para melhorar a aproximação dos resultados, e implementada uma função de baixo consumo, ativada a partir do momento em que um módulo ficar um certo tempo sem receber nenhuma transmissão de outros módulos.

2 TECNOLOGIA ULTRA BANDA LARGA

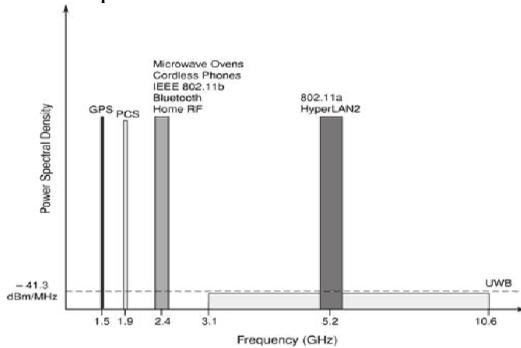
O termo *UWB* foi usado pela primeira vez em 1989, mas é uma tecnologia que já é conhecida desde a década de 60 [Svalastog, 2007]. Seu conceito principal é transmitir e receber pulsos de energia de radiofrequência de duração extremamente baixa, resultando em um espectro de frequência bem amplo. Esta tecnologia também utiliza uma técnica para distribuir a energia pela banda do sinal, resultado em uma baixa densidade espectral de potência [Alarifi et al. 2016]. As baixas frequências dos pulsos possibilitam também que o sinal atravesse obstáculos, como paredes e objetos.

A localização de objeto e pessoas em ambientes fechado tem se tornado uma necessidade cada vez maior no cenário atual, e muitas das tecnologias usadas para ambientes aberto, como *GPS – Global Positioning System*, não apresentam uma acurácia satisfatória. Diante disto, a tecnologia *UWB* está crescendo cada vez mais, principalmente como resposta a estas necessidades, já que seu funcionamento não requer linha de visada e não há interferência com outros dispositivos de comunicação que estejam nas proximidades, já que o espectro utilizado para transmissão *UWB* está localizado entre 3.1 e 4.8 GHz e 6 e 10.6 GHz. O custo dos dispositivos *UWB* também tendem a ser baixos e apresentar um consumo menor do que outras tecnologias, além da capacidade de apresentar erros de localização menores do que 1cm [Alarifi et al. 2016].

Atualmente grandes empresas como Apple [SHANKLAND, Stephen, 2019], Samsung [KJ Kim, 2020] e Xiaomi [Xiaomi team, 2020] estão começando a implementar tecnologia *UWB* em seus novos dispositivos. Como fabricantes de componentes podemos citar empresas como DecaWave (Qorvo) com o *transceiver DW1000* [DW1000 Datasheet, 2015], e também a NXP [NXP – Ultra-Wideband].

A regulação da tecnologia é feita pelo padrão IEEE 802.15.4 [IEEE Std 802.15.4a-2007], o qual especifica frequências de operação, estrutura das mensagens, limite de potência, entre outras características. Na Figura 3 podemos observar o espectro de frequência de algumas tecnologias, e também a densidade de potência permitida para transmissões *UWB*.

Figura 3 - Faixa de frequência de operação das principais tecnologias de rádio frequência

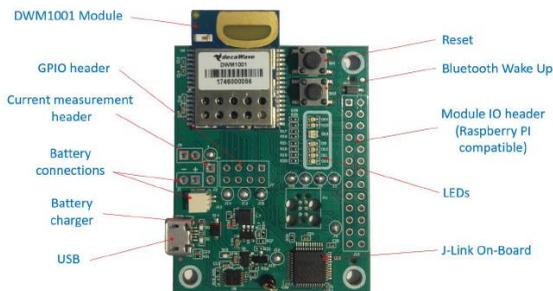


Fonte: IEE 802.15

2.1 KIT MDEK1001

O kit que foi utilizado para a realização do projeto foi o MDEK1001 [MDEK1001 system user manual, Decawave, 2017], da empresa DecaWave. Ao todo no projeto foram utilizados quatro kits idênticos. Na Figura 4 é apresentado o kit, assim como destacadas suas principais funcionalidades.

Figura 4 - Resumo do conteúdo do kit MDEK1001

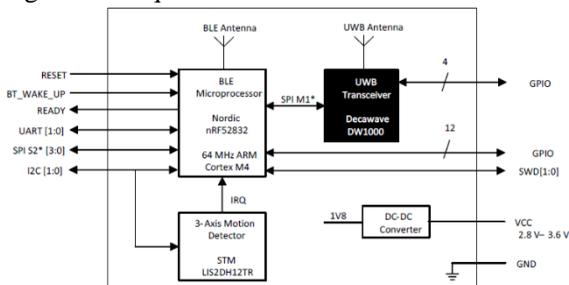


Fonte: MDEK1001 Kit User Manual, 2017, p. 13

O uso deste kit de desenvolvimento facilita e agiliza o processo, por já contar principalmente com a conexão USB, com J-Link embarcado para facilitar a interface e o *debug* do programa. Este kit

contém também o módulo DWM1001, com seu esquemático apresentado na Figura 5.

Figura 5 - Esquemático do módulo DMW1001



Fonte: DWM1001 Datasheet, p. 1.

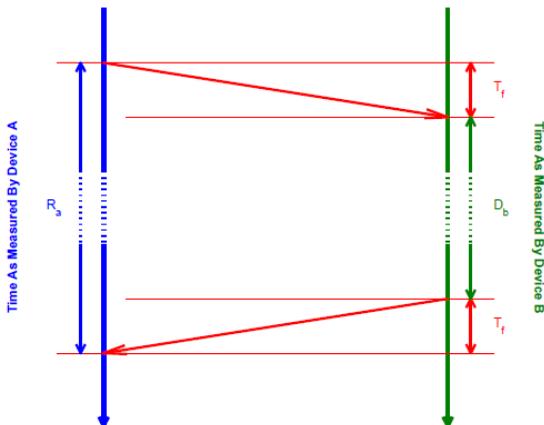
Este módulo contém um micro controlador principal, nRF52832, no qual é carregado o programa principal. Este microprocessador também possui uma interface *Bluetooth low energy* – BLE, mas que não será utilizada neste projeto. O módulo também contém o *transceiver UWB DW1000* [DW1000 Datasheet, 2015], com o qual o microprocessador se comunica através da *SPI – Serial Peripheral Interface*. Todas as funções base para controlar o *transceiver* já foram implementadas e fornecidas pela DecaWave em uma biblioteca. O módulo também conta com um acelerômetro, através do qual será proposto funcionalidades adicionais, e um conversor DC-DC, que não será utilizado neste projeto.

2.2 ALGORITMO DE CÁLCULO UTILIZADO

No início do desenvolvimento do projeto, estava sendo utilizado um algoritmo simples de *TWR – Two Way Ranging* [Decawave, APS013 2015], conforme a Figura 6, onde um dispositivo (A), com seu tempo representado pela linha azul, envia uma mensagem inicial, que após um tempo de voo T_f chega no segundo dispositivo (B), com seu tempo representado pela linha verde. Este dispositivo carimba o tempo de chegada da mensagem, chamado de *timestamp*, e após um certo tempo D_b envia uma resposta contendo o seu tempo de recepção e de envio, que novamente após um tempo T_f chega no dispositivo A, que por sua vez carimba o seu tempo de recepção. Conhecendo os *timestamps* de

ambas as transmissões e recepções, é possível obter o tempo de voo, T_f , que multiplicado pela velocidade da luz, resultará na distância entre os dois dispositivos

Figura 6 - Esquema básico de TWR



Fonte: Dries, Luk, and McLaughlin, 2016, p.2.

Da figura pode-se extrair a equação do tempo total transcorrido entre o envio mensagem inicial e a recepção da mensagem de resposta, R_a :

$$R_a = 2T_f + D_b, \quad (1)$$

da qual pode-se, por sua vez, obter-se o valor de T_f :

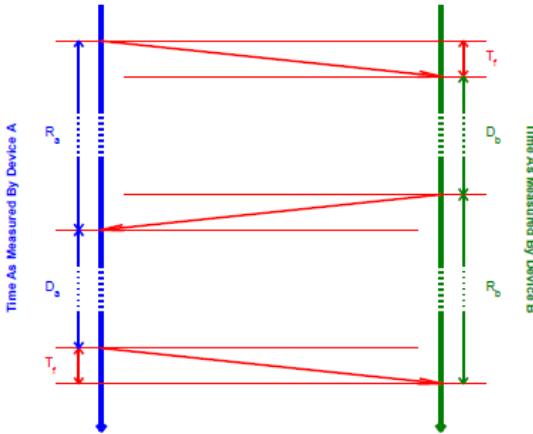
$$T_f = \frac{R_a - D_b}{2}, \quad (2)$$

que multiplicado pela velocidade de propagação da luz no ar resultará na distância entre os módulos. Este algoritmo foi implementado pela sua simplicidade e agilidade no desenvolvimento, importantes principalmente na fase inicial de fundamentação do protocolo de comunicação e testes iniciais.

Foi percebido, porém, que este algoritmo apresentava uma grande dependência em relação ao atraso de resposta D_b , ou seja, quanto mais tempo passava entre o recebimento da mensagem e a transmissão da resposta, maior o erro. Esta característica se mostrou um grande

empecilho, principalmente relacionado às questões de economia de energia, que requerem aumentar o tempo entre as mensagens. Foi encontrado então o algoritmo *Double-Sided Two-Way Ranging – DS-TWR*, proposto em [Neirynek et al. 2016], no qual uma mensagem extra é enviada, como pode ser observado na Figura 7.

Figura 7 - Esquema do *TWR* com resposta adicional



Fonte: Dries, Luk, and McLaughlin, 2016, p.2.

A partir da figura é possível extrair as seguintes equações:

$$R_a = 2T_f + D_b, \quad (3)$$

$$R_b = 2T_r + D_a. \quad (4)$$

Multiplicado as equações (3) e (4) e rearranjando os termos obtemos a seguinte equação para o cálculo exato de T_f :

$$T_f = \frac{R_a R_b - D_a D_b}{2(R_a + D_a)} = \frac{R_a R_b - D_a D_b}{R_a + D_a + R_b + D_b}. \quad (5)$$

A deriva de *clock* é um problema de *hardware* causado pela variação da frequência do cristal oscilador devido ao ruído, temperatura, envelhecimento, variação de tensão, entre outros [Chee et al. 2004]. Como a medição de tempo em cada módulo depende diretamente da frequência do cristal, este problema é tido como a principal fonte de erro, e é possível modelá-la da seguinte forma:

$$\hat{R}_a = (1 + e_a)R_a = k_a R_a, \quad (6)$$

$$\hat{R}_b = (1 + e_b)R_b = k_b R_b, \quad (7)$$

$$\hat{D}_a = (1 + e_a)D_a = k_a D_a, \quad (8)$$

$$\hat{D}_b = (1 + e_b)D_b = k_b D_b, \quad (9)$$

onde e_x representa a variação em relação à frequência nominal em cada módulo, e é usualmente expressado em ppm (partes por milhão) [Neiryneck et al. 2016]. A estimativa de T_f considerando o erro se torna:

$$\hat{T}_f = \frac{k_a k_b R_a R_b - D_a D_b}{k_b 2(R_a + D_a)} = k_a \frac{R_a R_b - D_a D_b}{2(R_a + D_a)} = k_a T_f. \quad (10)$$

O erro então pode ser calculado como:

$$\hat{T}_f - T_f = k_a T_f - T_f = e_a T_f. \quad (11)$$

Fica claro na equação (10) que o erro fica na casa de $e_x T_f$, e não depende mais do tempo de resposta. Como exemplo, para um cristal de 125MHz, e um pior caso de ± 20 ppm segundo o padrão IEEE 802.15.4 [IEEE Std 802.15.4a-2007], considerando que T_f normalmente está perto de algumas centenas de nanosegundos, temos um erro no pior dos casos de menos de um pico segundo, ou seja, menos de 3mm. Esta característica possibilitou um grande aumento na precisão do sistema e uma maior flexibilidade na implementação do código.

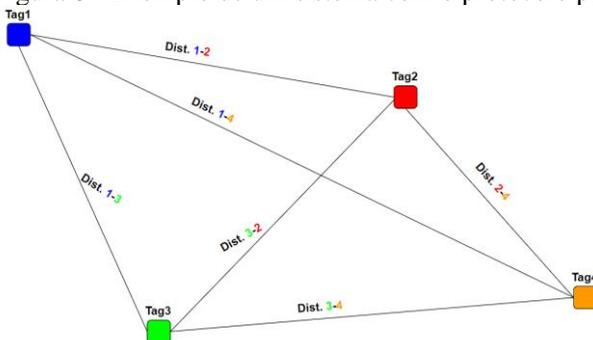
Um ponto a ser salientado é que, apesar do algoritmo precisar de uma mensagem a mais para realizar a estimação em relação ao algoritmo de TWR, é possível utilizar informações utilizadas no cálculo anterior. Na Figura 7, quando o dispositivo B enviar sua próxima mensagem, D_a passará a ser o novo R_a , e D_b passará a ser o novo R_b , de forma com que a cada mensagem recebida seja possível realizar uma nova estimativa

2.3 PROTOCOLO DE COMUNICAÇÃO IMPLEMENTADO

A ideia para o desenvolvimento do protocolo de comunicação foi criar uma rede com transmissões somente *broadcast*, sem que existisse necessidade de comunicação módulo-a-módulo, permitindo com que todos as *tags* no sistema conseguissem saber a a distância relativa uma das outras. Isto permitiu também uma simplicidade maior no protocolo, já que todas as mensagens enviadas e recebidas possuem o mesmo

formato, possibilitando se comunicar com qualquer módulo que esteja no alcance da transmissão. O protocolo também permite que os módulos entrem e saiam do alcance um dos outros sem que haja nenhuma interferência no sistema. Na Figura 8 pode-se observar como funciona um sistema com o protocolo proposto.

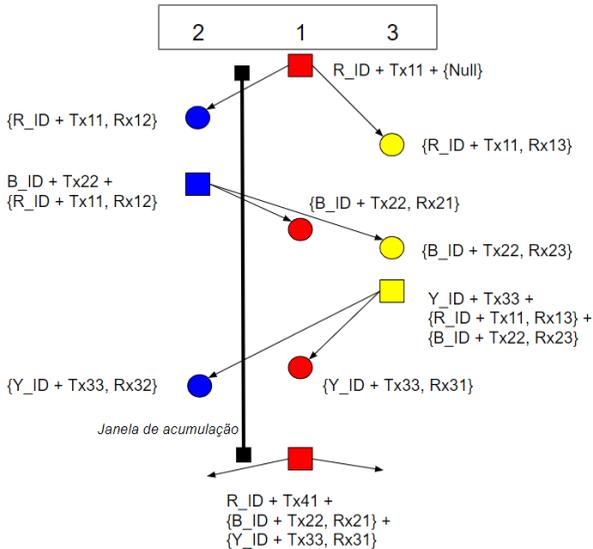
Figura 8 - Exemplo de um sistema com o protocolo proposto



Fonte: Elaborada pelo autor

Nota-se também que é possível que as *tags* calculem as distâncias relativas entre outras *tags* que estão no alcance de comunicação, já que as mensagens são *broadcast*. Na Figura 9 temos uma exemplificação de como o protocolo de comunicação foi implementado.

Figura 9 - Exemplo do protocolo de comunicação implementado



Fonte: Elaborada pelo autor

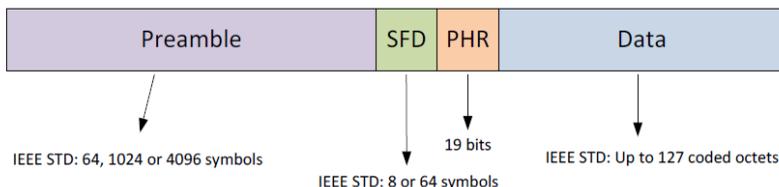
Cada módulo possui um intervalo de tempo para ficar em modo de recebimento, e após esse tempo realiza uma transmissão com os dados acumulados durante essa janela. Quando um módulo envia uma mensagem, esta contém seu número de identificação (ID), o *timestamp* de transmissão e um pacote com os dados acumulados de outras *tags*. Este pacote contém, para cada *tag* da qual foi recebida mensagem durante a janela de recebimento: seu ID, *timestamp* no qual a mensagem foi enviada e *timestamp* no qual a mensagem foi recebida.

Quando um módulo recebe uma mensagem, este salva o tempo em que a mesma foi recebida, e procura no pacote recebido por seu próprio ID. Se o módulo encontrar seu ID, ele salva as informações relativas e verifica se é possível realizar o cálculo da distância, em seguida volta a entrar no modo de recepção.

2.4 ESTRUTURA DA MENSAGEM

A mensagem possui a estrutura base apresentada na Figura 10.

Figura 10 - Base de uma mensagem *UWB*



Fonte: Dw1000 User Manual, 2017, p. 215.

A mensagem é composta pelo *Preamble*, que serve para auxiliar na recepção com sincronização da mensagem e estimação do canal, *SFD* (*frame delimiter sequence*), que é utilizado para estabelecer qual a velocidade da comunicação e *PHR* (*PHY header*), que contém as informações necessárias para decodificar o pacote recebido [IEEE Std 802.15.4a-2007]. Nesta seção será explicado como é composto a parte de *Data* implementada para o funcionamento do projeto.

A estrutura base dos dados está apresentada na Tabela 1.

Tabela 1 - Estrutura da mensagem

Para cada recepção					
HDR	ID	Txa_Ts	R_IDb	Rxb_Ts	Txb_Ts
5 bytes	4 bytes	5 bytes	4 bytes	5 bytes	5 bytes

Fonte: Elaborada pelo autor.

- HDR: Cabeçalho da mensagem, usado para reconhecer módulos da mesma rede;
- ID: Número de identificação do módulo enviando a mensagem;
- Txa_Ts: *Timestamp* que está sendo enviada a mensagem;

Para cada mensagem recebida durante a janela de acumulação:

- R_IDb: Número de identificação do módulo que enviou a mensagem;
- Rxb_Ts: *Timestamp* que foi recebida a mensagem;

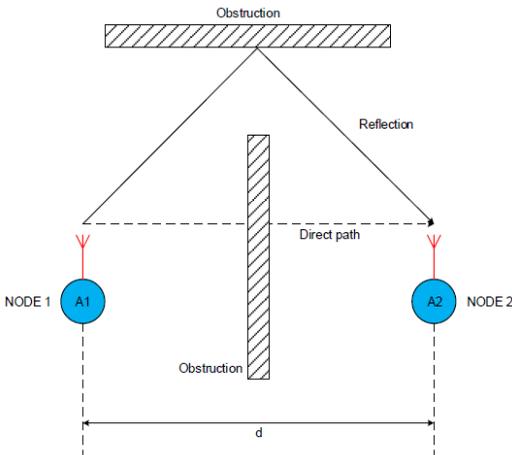
c. T_{xb_Ts} : *Timestamp* que foi enviada a mensagem;

Como apresentado na tabela, a parte de *Data* possuirá 14 Bytes fixos, mais 14 Bytes para cada mensagem recebida na janela de acumulação. Como o padrão IEEE 802.15.4 [IEEE Std 802.15.4a-2007] define um tamanho máximo de 127 bytes para esta parte da mensagem, é possível acumular dados de até 8 módulos diferentes, ultrapassando este limite, as mensagens recebidas de novos módulos serão descartadas e as mensagens recebidas de módulos com informações já obtidas serão utilizadas para atualização.

2.5 ALGORITMO PARA DETECÇÃO DE LINHA DE VISADA

Um dos problemas enfrentados ao se utilizar a tecnologia *UWB* é a operação em ambientes nos quais não existe linha de visada (*line of sight – LOS*). Na Figura 11 temos apresentados os dois principais cenários que podem ocorrer, uma obstrução no caminho de comunicação direto, e uma reflexão em alguma superfície próxima.

Figura 11 - Transmissão em um ambiente sem linha de visada



Fonte: APS006 Part 1 Channel effects on communications range and time stamp accuracy in DW1000 based systems, 2014, p. 14.

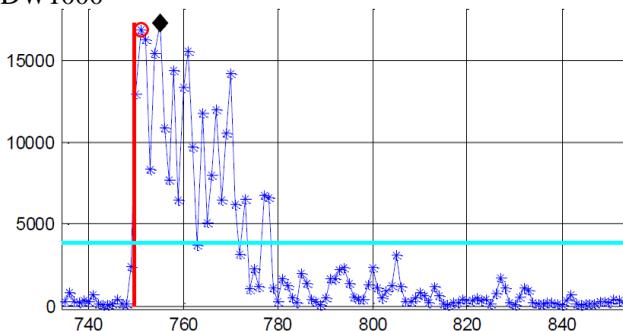
Na situação em que ocorre uma obstrução no sinal, pode ocorrer uma medição errada da distância, visto que uma onda em

radiofrequência se propaga em velocidade menor em outros meios do que no ar. Em situações mais extremas, dependendo do material e da sua espessura, o sinal pode ser atenuado de forma que não possa mais ser reconhecido pelo módulo receptor.

Já na situação na qual ocorre uma reflexão e uma obstrução, como mostra a Figura 11, o sinal refletido percorre uma distância maior do que o sinal do caminho direto, e chega depois ao receptor, mas com amplitude maior, por não ter sido atenuado por obstáculos. Nesta situação, dependendo do nível dos sinais, é possível detectar o sinal de caminho direto. Um problema que acontece nesta situação é quando não existe reflexão, mas somente caminho sem linha de visada, pois o sinal recebido será atenuado e atrasado ao travessar o objeto, fazendo com que seja feita uma estimativa errada ou que a comunicação nem chegue a ser bem sucedida. Esta situação acaba limitando o uso da técnica apresentada somente para cenários com linha de visada, a não ser que informações externas sejam agregadas ao sistema.

Para minimizar os erros causados pela falta de linha de visada, a própria empresa fabricante do kit utilizado, a DecaWave, sugere um algoritmo para a detecção de cenários sem linha de visada [APS006 Part 3, DecaWave, 2016]. O *transceiver* DW1000 possui um acumulador que guarda o valor da amplitude recebida. Um exemplo dos valores do acumulador pode ser visto na Figura 12.

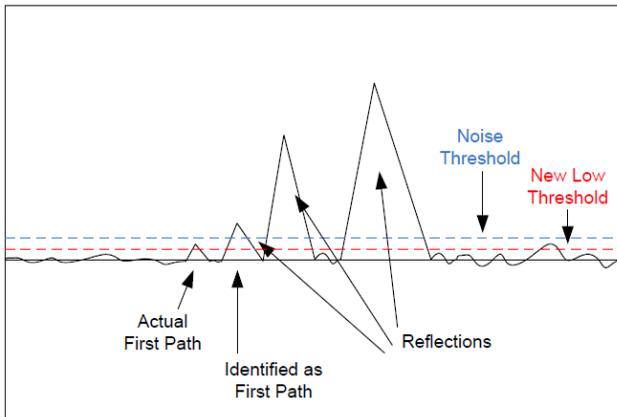
Figura 12 - Exemplo dos dados armazenados no acumulador do DW1000



Fonte: APS006 Part 3 DW1000 Metrics for Estimation of Non-Line Of Sight Operating Conditions, 2016, p. 14.

A linha azul claro na Figura 12 representa o limite de amplitude para interpretação de sinal válido. A Figura 13 mostra um cenário parecido com o mostrado na Figura 11, onde o primeiro sinal sofreu uma atenuação ao ponto de ficar abaixo do limite de detecção, e são recebidos sinais atrasados com maior amplitude, provenientes de reflexões. Desta forma, o primeiro sinal detectado é proveniente de uma reflexão, sendo incorretamente identificado.

Figura 13 - Valores do acumulador sob um cenário com obstrução e reflexões



Fonte: APS006 Part 3 DW1000 Metrics for Estimation of Non-Line Of Sight Operating Conditions, 2016, p. 10.

O algoritmo proposto analisa o conteúdo do acumulador ao receber uma mensagem, e busca picos que não foram detectados (que chegaram com uma amplitude abaixo do limite de detecção), como é observado na Figura 13. Se foram detectados picos abaixo do limite mínimo, o limite de detecção é diminuído, multiplicando o limite anterior por 0,6 [APS006 Part 3, DecaWave, 2016], como mostra a Figura 13. Se algum dos picos detectados ficar acima do novo limite, é considerado que a comunicação ocorreu em um cenário sem linha de visada, e a mensagem é descartada.

Este algoritmo foi implementado, e, ao descartar todas as mensagens que foram recebidas em cenários sem linha de visada, observou-se que erros esporádicos que estavam aparecendo deixaram de acontecer, e a precisão do sistema aumentou.

2.6 FILTRO 1EURO

Para melhorar as estimativas das distâncias obtidas no projeto, foi implementado um filtro sobre os valores de distância estimados, chamado de “1€” (*One Euro filter*). O filtro 1€ usa um filtro passa-baixa de primeira ordem, mas possui uma frequência de corte variável, que se adapta de acordo com a velocidade do sinal de variação do sinal, sua derivada [Casiez et al. 2012]. É um algoritmo de fácil implementação, e nas situações em que poucas características do sinal são conhecidas, apresenta bons resultados.

Uma equação discreta para o filtro de primeira ordem pode ser observada a seguir:

$$\hat{X}_i = (1 - \alpha)\hat{X}_{i-1} + \alpha X_i, \quad (12)$$

onde \hat{X}_i é o valor filtrado, e α é o fator de alisamento. Percebe-se que o sinal filtrado é uma soma do sinal atual ponderada pelo fator α , e da última amostra filtrada ponderada pelo fator de $1 - \alpha$. Reescrevendo α como uma função do período entre as amostras, e da constante de tempo do sistema obtemos:

$$\alpha = \frac{1}{1 + \frac{\tau}{T_e}}, \quad \text{onde} \quad (13)$$

$$\tau = \frac{1}{2\pi f_c}, \quad (14)$$

e f_c é a frequência de corte do filtro. Substituindo (13) e (14) em (12), obtemos:

$$\hat{X}_i = \left(\frac{\tau}{T_e} \hat{X}_{i-1} + X_i \right) \frac{1}{1 + \frac{\tau}{T_e}}. \quad (15)$$

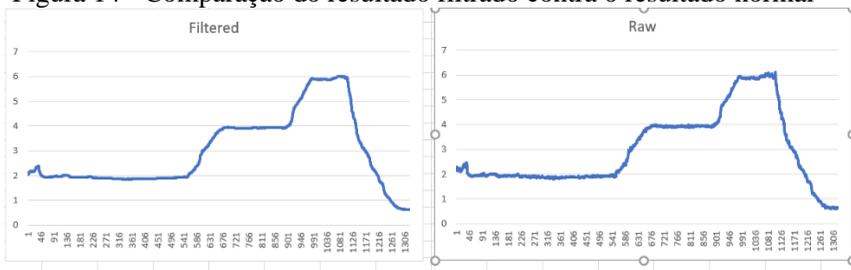
Para a f_c , é proposto a seguinte igualdade [Casiez et al. 2012]:

$$f_c = f_{cmin} + \beta \left| \dot{\hat{X}}_i \right|, \quad (16)$$

onde f_{cmin} é um valor mínimo para a frequência de corte, β é um parâmetro de calibração do filtro para controlar a variação da frequência de corte e $|\dot{\hat{X}}_l|$ é o módulo da derivada do valor filtrado.

Sendo assim, possuímos apenas dois parâmetros de calibração do filtro, f_{cmin} e β . Como os dados da distância não apresentam grandes variações e já é bem preciso, o ganho da implementação do filtro foi mais em um alisamento do gráfico, como pode ser observado na Figura 14. Na imagem da esquerda temos o sinal filtrado e na imagem da direita o sinal original.

Figura 14 - Comparação do resultado filtrado contra o resultado normal



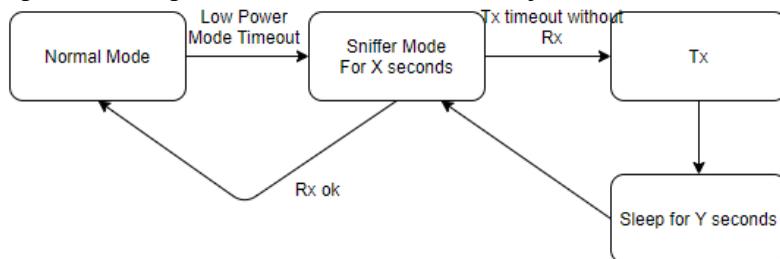
Fonte: Elaborada pelo autor

Foram utilizados valores base de exemplo para os parâmetros do filtro, mas para aplicações onde possa existir uma variação mais rápida na posição relativa entre os módulos, como alguma máquina com maior velocidade, pode-se ajustar os parâmetros do filtro para se melhorar ainda mais o resultado obtido.

2.7 FUNÇÕES ADICIONAIS IMPLEMENTADAS

Como funcionalidade adicional do projeto, foi desenvolvida uma função de baixo consumo, para economizar energia em um ambiente isolado, em que o módulo não recebe nenhuma mensagem de outro módulo. Na Figura 15 temos um resumo da implementação desta função.

Figura 15 - Diagrama de funcionamento da função de baixo consumo



Fonte: Elaborada pelo autor

Quando o módulo está em funcionamento normal, a cada recepção realizada um timer é zerado e iniciado. Se este timer atinge o seu valor final, o módulo entra no modo de baixo consumo. O módulo então passa um certo tempo em modo “*Sniffer*”, onde este fica alternando entre modo de recepção e de baixo consumo, na espera de receber alguma mensagem de outro módulo. Se X segundos passam sem que uma mensagem seja recebida, o modo faz uma transmissão simples para acordar possíveis módulos que estejam em baixo consumo dentro do alcance, depois entra em modo “*Sleep*”, no menor consumo possível, por Y segundos, após os quais ele irá entrar em modo *Sniffer* novamente. Se durante este modo o módulo receber uma mensagem, ele volta ao modo normal de operação.

2.8 FLUXOGRAMA E FUNCIONAMENTO DO CÓDIGO

Na Figura 16 temos apresentado o fluxo resumido de funcionamento do programa. Após as configurações iniciais, as quais englobam parâmetros das transmissões e recepções *UWB* e configurações dos *timers* utilizados, são iniciados os *timers* para controle da função *Low Power* e para o intervalo de transmissões, e habilitado o modo de recepção. O módulo permanece neste estado até que alguma das seguintes interrupções seja acionada:

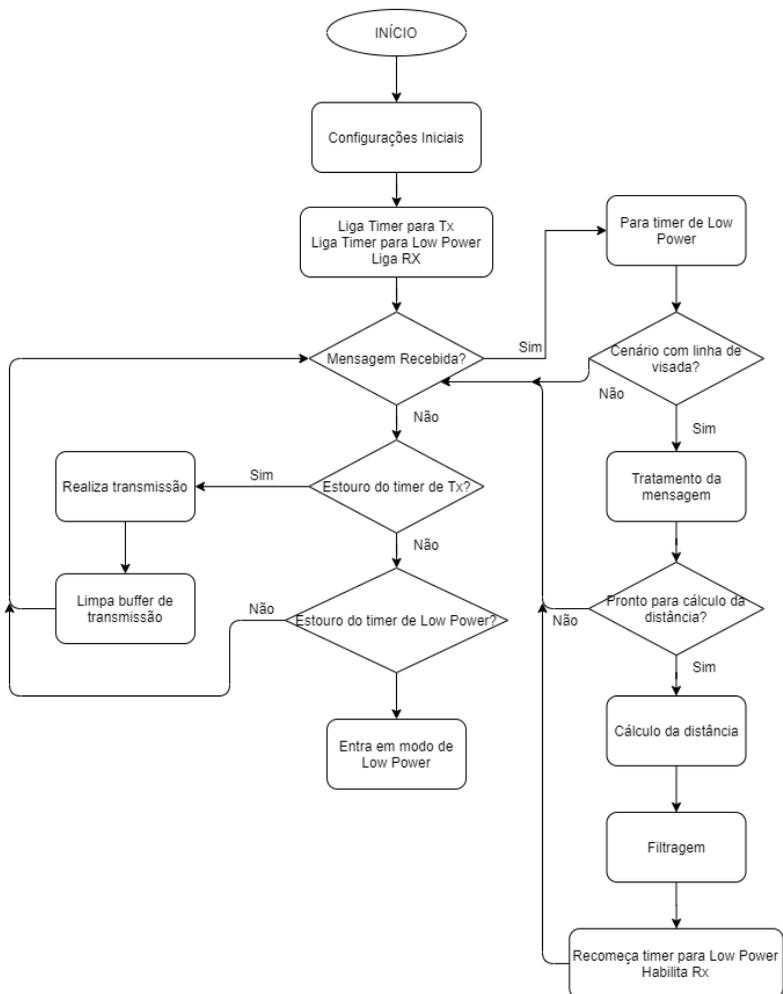
- a. Mensagem recebida
- b. Estouro do *timer* de transmissão
- c. Estouro do *timer* para entrar em modo *Low Power*

Caso ocorra a recepção de uma mensagem, depois de parar o *timer* para entrar na função de *Low Power*, é aplicado o algoritmo proposto de detecção de comunicação sem linha de visada. Se o cenário foi identificado com linha de visada, é passado para a parte de tratamento da mensagem, onde será gravado o *timestamp* de recepção, varrer o pacote enviado procurando o ID próprio, de forma a armazenar os dados para cálculo, validação se foi perdido algum pacote, armazenamento das informações na estrutura criada para cada módulo e armazenamento das informações necessárias no *buffer* de saída. Se o módulo possuir informações necessárias para calcular a distância ao módulo que enviou a mensagem, esta será calculada, depois filtrada. A partir deste ponto podem se tomar as decisões partindo-se do resultado obtido. Ao final, é religado o *timer* para entrar na função de *Low Power*, e habilitado novamente o modo de recepção.

Caso ocorra um estouro do *timer* para se realizar a transmissão, o modo de recepção é pausado, depois é feita a transmissão com o conteúdo atual do *buffer* de saída, que então é limpo para armazenar novas informações.

Caso ocorra um estouro do *timer* para entrar na função de *Low Power*, o módulo entrará neste modo até receber alguma nova mensagem de outra *tag*.

Figura 16 - Fluxograma resumido de funcionamento do programa



Fonte: Elaborada pelo autor

3 RESULTADOS OBTIDOS

Inicialmente foi realizada uma pequena simulação de cenário com os quatro kits disponíveis rodando o código implementado, onde foi possível validar a entrada e saída de dispositivos da rede assim como o correto funcionamento em um ambiente com mais de duas *tags*. Em seguida, com o código implementado rodando em dois módulos, foi feito um experimento para demonstração inicial dos resultados apresentados pelo projeto, em um cenário com linha de visada, e sem obstruções. Na Figura 17 encontra-se uma foto do ambiente de medição.

Figura 17 - Ambiente de medição



Fonte: Acervo do autor

Foram colocadas marcas em 1, 2, 3 e 4 metros. Inicialmente o módulo foi deixado parado na marca de 1m, depois movido para a marca de 2m, e em seguida foi movido continuamente para a marca de 4m. O resultado das distâncias filtradas se encontra na Figura 18

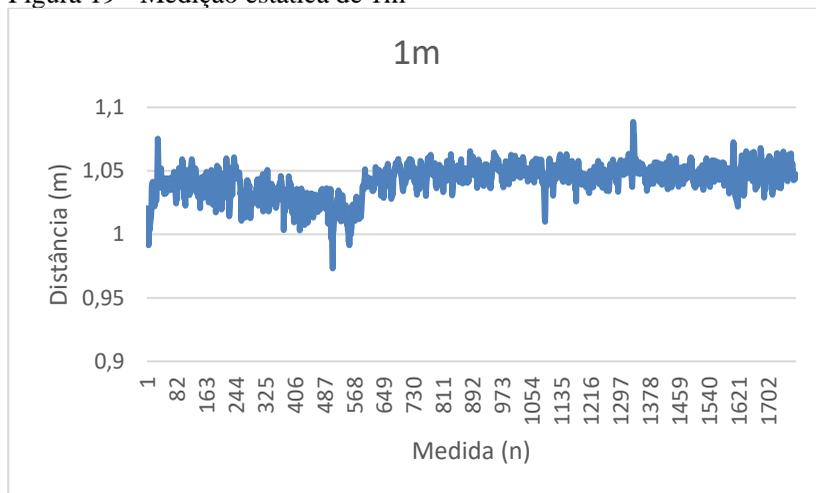
Figura 18 - Distância medida em um teste inicial



Fonte: Elaborada pelo autor

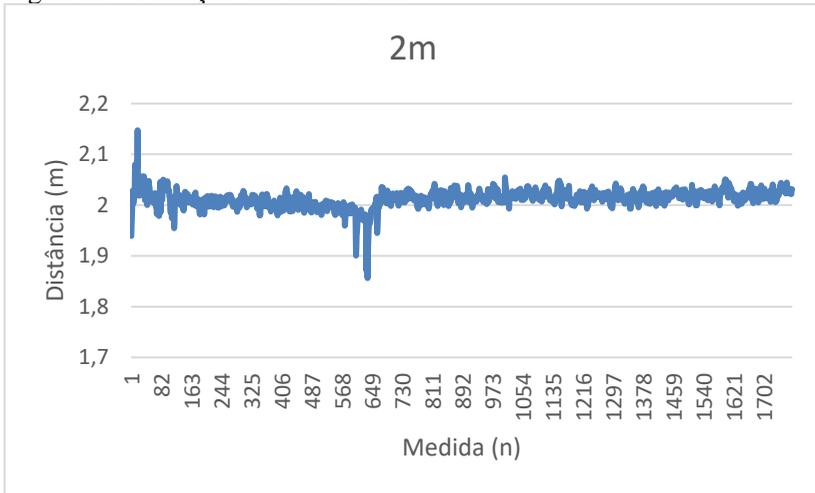
Para análise mais profunda dos resultados, os módulos foram mantidos estáticos em cada marca por um período maior, sendo coletadas 1729 amostras em cada etapa. Os resultados estão apresentados em: Figura 19, Figura 20, Figura 21 e Figura 22.

Figura 19 - Medição estática de 1m



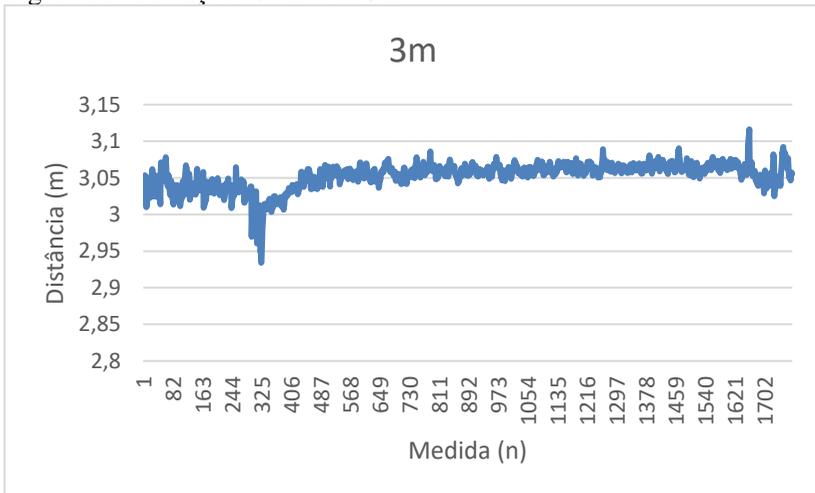
Fonte: Elaborada pelo autor

Figura 20 - Medição estática de 2m



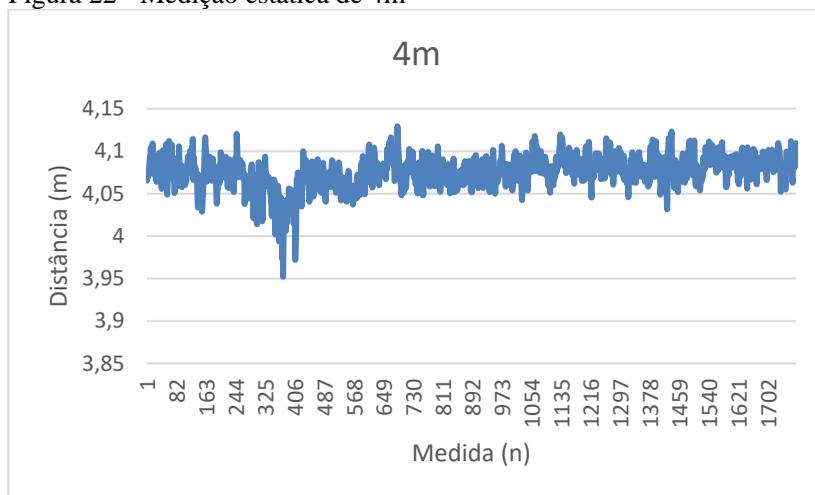
Fonte: Elaborada pelo autor

Figura 21 - Medição estática de 3m



Fonte: Elaborada pelo autor

Figura 22 - Medição estática de 4m



Fonte: Elaborada pelo autor

Observando as figuras, pode-se perceber que, desconsiderando algumas amostras esporádicas, a variação em geral permanece perto de 5cm.

A Tabela 2 apresenta os valores obtidos da média das amostras e da média do módulo do erro.

Tabela 2 - Análise dos resultados obtidos

Distância (m)	Média das amostras (m)	Média do módulo do erro (m)
1	1,042	0,04243
2	2,012	0,01704
3	3,053	0,05360
4	4,075	0,07561

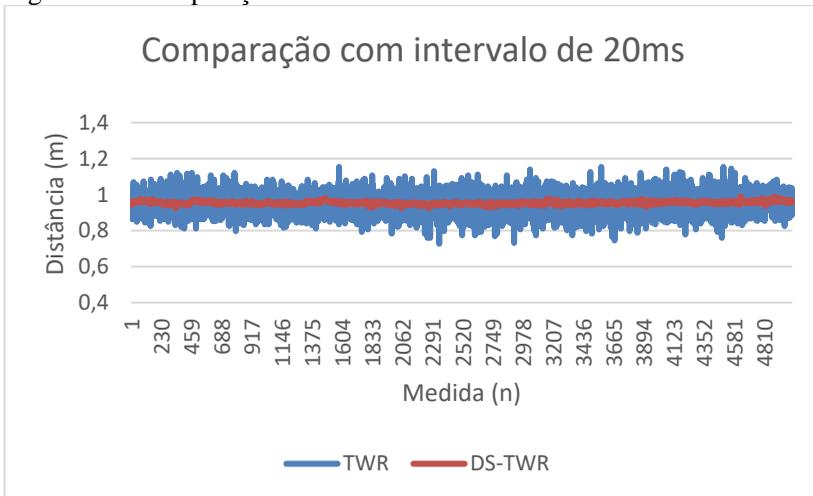
Fonte: Elaborada pelo autor

Considera-se também que podem haver ocorrido pequenos erros nas medições das marcas, e no posicionamento dos módulos, o que poderia explicar um aumento do erro nas medições de 1m em relação às medições de 2m.

3.1 COMPARAÇÃO DO ALGORITMO DE *TWR* COM O ALGORITMO *DS-TWR*

Foi também realizado um teste para que fosse possível visualizar a diferença de resultado entre o método inicial de cálculo que estava sendo utilizado (*TWR*), com o que posteriormente foi implementado (*D-TWR*). Também é possível confirmar o efeito de aumentar o tempo entre as mensagens na estimativa da distância. Foram feitas três etapas de medições, sempre com os módulos estáticos, com linha de visada, e com distância de 1m. Na primeira etapa foi usado um intervalo entre a recepção da mensagem e o envio da resposta de 20ms. O gráfico obtido encontra-se na Figura 23.

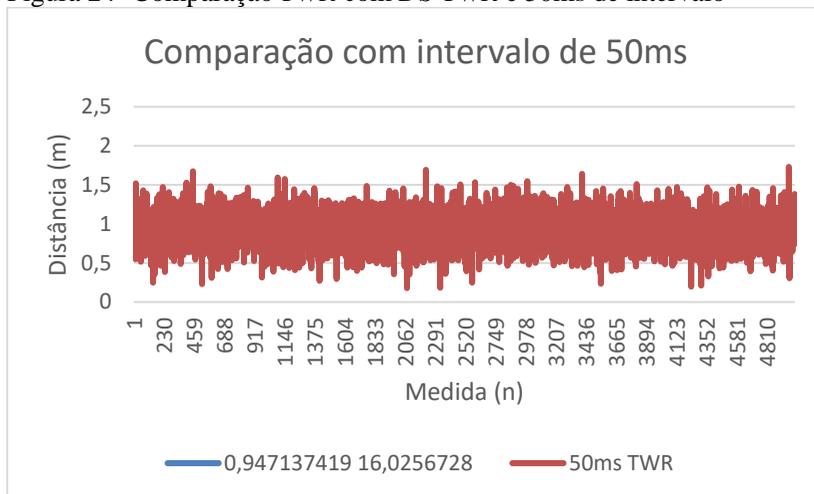
Figura 23 - Comparação *TWR* com *DS-TWR* e 20ms de intervalo



Fonte: Elaborada pelo autor

A segunda etapa de medição foi realizada com intervalo entre a recepção da mensagem e o envio da resposta de 50ms. O gráfico obtido encontra-se na Figura 24. Pode-se perceber que a resposta do algoritmo inicial apresenta um resultado muito pior do que no caso anterior, e o algoritmo novo é muito pouco afetado.

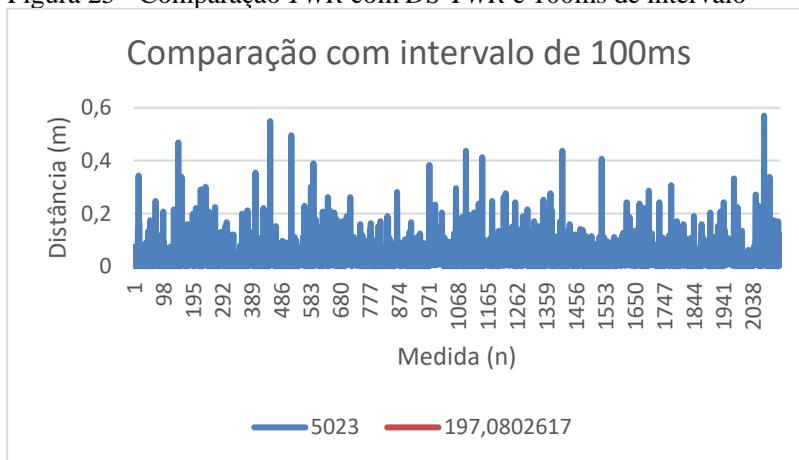
Figura 24- Comparação *TWR* com *DS-TWR* e 50ms de intervalo



Fonte: Elaborada pelo autor

Já no último cenário foi utilizado um intervalo de 100ms, e o gráfico obtido encontra-se na Figura 25. Neste caso a estimativa com o *TWR* fica muito degradada, possuindo valores de 0m até 1,93m. A resposta do *DS-TWR*, porém, se mantém com pouca alteração, pois, como exposto anteriormente, seu erro não depende do intervalo entre as mensagens.

Figura 25 - Comparação *TWR* com *DS-TWR* e 100ms de intervalo



Fonte: Elaborada pelo autor

Na Tabela 3 temos os valores para a média do módulo do erro obtida para o algoritmo de *TWR* assim como o desvio padrão das amostras de *TWR* e a média para cada caso. Percebe-se como o módulo do erro e o desvio padrão crescem, mesmo com um valor médio dos valores não sofrendo tanta alteração.

Tabela 3 - Resultado da comparação entre *TWR* e *DS-TWR*

Intervalo (ms)	Média do módulo do erro (m)	Desvio Padrão	Média (m)
20	0,0638	0,05648	0,94714
50	0,1656	0,19808	0,93342
100	0,2478	0,29934	0,91188

Fonte: Elaborada pelo autor

Através dos resultados apresentado, fica clara a vantagem do método de *DS-TWR* sobre o método de *TWR*. Com o primeiro, além de o projeto apresentar estimativas melhores, também se ganha uma flexibilidade maior na implementação da comunicação, na alocação dos tempos de transmissão e no gerenciamento do consumo. Por fim, também foi possível validar a correta implementação do algoritmo dentro do projeto.

4 CONCLUSÃO

Este trabalho apresentou a implementação de um algoritmo que utiliza tecnologia *UWB* para calcular a distância relativa entre módulos. O algoritmo proposto foi implementado de forma com que os módulos possam entrar e sair livremente da rede, sem necessidade de haver nenhuma âncora no sistema, e nenhum módulo com funções de gerenciamento da rede, o que facilita a implementação do sistema.

Foi feita uma explanação da tecnologia utilizada, e apresentados todos os pontos que precisaram ser estudados e desenvolvidos para a realização do projeto, como o estudo do algoritmo de cálculo mais adequado para a situação apresentada e suas fontes de erro, a implementação de uma análise do cenário em que ocorreu a comunicação e a implementação de um filtro para melhorar a qualidade das estimativas de distância.

Os resultados obtidos foram satisfatórios e apresentam erros compatíveis com a tecnologia utilizada, e também uma resposta rápida. Estas características tornam o projeto proposto uma solução válida para os problemas que se propõem solucionar.

Como trabalhos futuros, é sugerida a realização de uma simulação, abordando as diferentes variáveis do problema, como: tamanho da área a ser monitorada, número de módulos, tempo de transmissão, entre outras. Esta simulação teria como principal objetivo a análise de desempenho do sistema proposto em um ambiente de funcionamento real, apresentando também a chance de perda de mensagens.

Este conteúdo é de significativa importância para uma boa interpretação das capacidades e limites do sistema proposto, assim como para poder detectar qual o impacto do resultado na modificação de cada variável presente no projeto.

Além da realização da simulação, também é proposta uma análise mais profunda, assim como uma melhor modelagem em relação às influências do meio no comportamento do sistema.

REFERÊNCIAS

Neiryneck, D., Luk, E. and McLaughlin, M., 2016, October. An alternative double-sided two-way ranging method. In 2016 13th workshop on positioning, navigation and communications (WPNC) (pp. 1-4). IEEE.

Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M.A. and Al-Khalifa, H.S., 2016. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16(5), p.707.

Clarinox Technologies Pty Ltd. "Real Time Location Systems." 2009.

Tjoa, Randolph, Kim Loon Chee, P. K. Sivaprasad, S. V. Rao, and Joo Ghee Lim. "Clock drift reduction for relative time slot TDMA-based sensor networks." In 2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No. 04TH8754), vol. 2, pp. 1042-1047. IEEE, 2004.

"IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs): Amendment 1: Add Alternate PHYs," in IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006) , vol., no., pp.1-210, 31 Aug. 2007, doi: 10.1109/IEEESTD.2007.4299496.

Decawave Ltd "APS006 Part 1 Channel effects on communications range and time stamp accuracy in DW1000 based systems." Application Note v 1.03, 2014.

Decawave Ltd "APS006 Part 3 DW1000 Metrics for Estimation of Non-Line Of Sight Operating Conditions." Application Note v 1.1, 2016.

Casiez, Géry, Nicolas Roussel, and Daniel Vogel. "1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2012.

DW1000 Datasheet, Version 2.09, Decawave. 2015.

DW1001 Datasheet, Version 1.10, Decawave. 2017.

MDEK1001 sytem user manual, version 1.0, Decawave, Ltd, 2017

Decawave Ltd.DW1000 USER MANUAL, <http://www.decawave.com/>, 2015.

Decawave Ltd “APS013: DW1000 and two-way ranging” Application Note v 2.3, 2015.

What is RTLS? | An Introduction to Real-Time Location Systems. Disponível em: <<https://www.atlasrfidstore.com/what-is-rtls-an-introduction-to-real-time-location-systems/>> Acesso em 06 nov. 2020

Svalastog, M.S. Indoor Positioning-Technologies, Services and Architectures. Cand Scient Thesis, University of Oslo, Oslo, Norway, 2007.

SHANKLAND, Stephen “Apple built UWB into the iPhone 11. Here's what you need to know (FAQ)”, 14 set. 2019. Disponível em: <https://www.cnet.com/news/apple-built-uwb-into-the-iphone-11-heres-what-you-need-to-know-faq/>. Acesso em 26 nov. 2020

XIAOMI TEAM. “XIAOMI INTRODUCES GROUNDBREAKING UWB TECHNOLOGY”, 13 out. 2020. Disponível em: <https://blog.mi.com/en/2020/10/13/xiaomi-introduces-groundbreaking-uwb-technology/>. Acesso em 26 nov. 2020

KJ Kim. “Samsung Expects UWB To Be One of the Next Big Wireless Technologies”, 12 out. 2020. Disponível em: <https://news.samsung.com/global/samsung-expects-uwb-to-be-one-of-the-next-big-wireless-technologies>. Acesso em 26 nov. 2020

NXP. Ultra-Wideband (UWB). Disponível em: https://www.nxp.com/applications/enabling-technologies/connectivity/ultra-wideband-uwb:UWB?gclid=CjwKCAiAnvj9BRA4EiwAuUMDf1_KH6ZwyaajGK4S3-SDNHNVoKBLv9ygAv7E8-0B6_A2dTEm4IEAdBoCZYsQAvD_BwE. Acesso em 25 nov. 2020