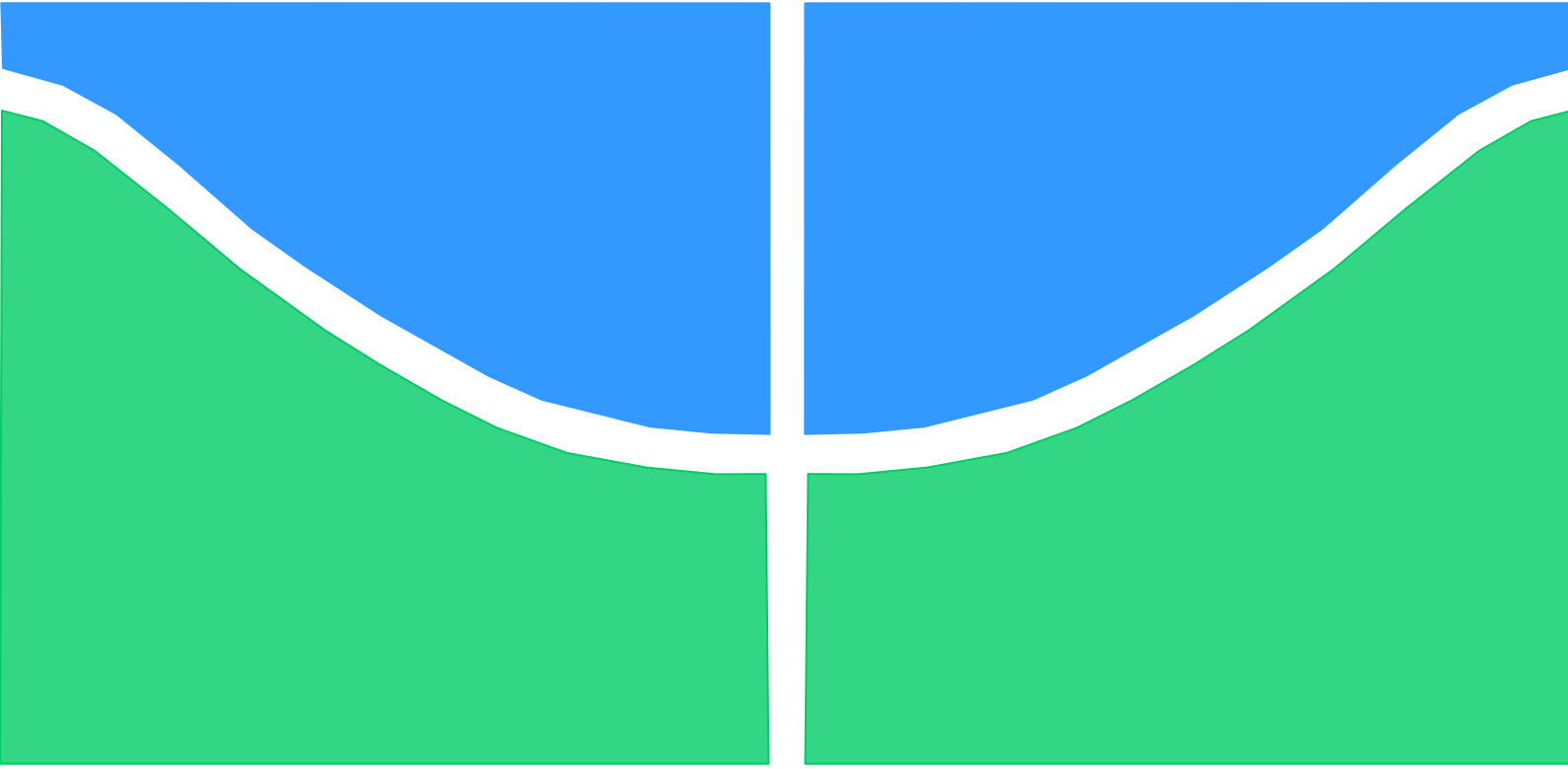


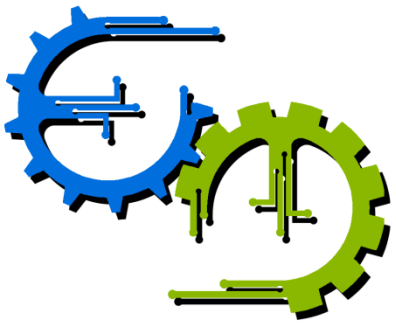
TRABALHO DE GRADUAÇÃO

CARACTERIZAÇÃO DE UM RADAR *MMWAVE* PARA MEDIÇÃO DE ESPAÇOS 3D

Por,

Vitor Miranda Kurovski

Brasília, Maio de 2021



ENGENHARIA
MECATRÔNICA
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

Caracterização de um radar *mmWave* para medição de espaços 3D

POR,

Vitor Miranda Kurovski

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação.

Banca Examinadora

Prof. Jones Yudi Mori Alves da Silva, UnB/ ENM (Orientador)	
Prof. Gerardo Antonio Idrobo Pizo, UnB/ FGA	
Prof. Renato Coral Sampaio, UnB/ FGA	

Brasília, Maio de 2021

FICHA CATALOGRÁFICA

KUROVSKI, VITOR MIRANDA
Caracterização de um radar *mmWave* para medição de espaços 3D.

[Distrito Federal] 2021.

xii, 74p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, Ano). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1.Sensor 2.Radar
3.Nuvem 4.Pontos

I. Mecatrônica/FT/UnB II. Graduação

REFERÊNCIA BIBLIOGRÁFICA

KUROVSKI, VM, (2021). Caracterização de um radar *mmWave* para medição de espaços 3D. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº09, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 74p.

CESSÃO DE DIREITOS

AUTOR: Vitor Miranda Kurovski

CARACTERIZAÇÃO DE UM RADAR *MMWAVE* PARA MEDIÇÃO DE ESPAÇOS 3D.

GRAU: Engenheiro ANO: 2021

Grupo de Automação e Controle
Faculdade de Tecnologia
Universidade de Brasília
CEP: 71.745-403 – Brasília – DF - Brasil

AGRADECIMENTOS

Gostaria de agradecer a todos que me apoiaram, me deram suporte e me acompanharam nessa jornada, especialmente aos meus pais.

Vitor Miranda Kurovski.

RESUMO

Este trabalho apresenta o estudo de software e hardware de sensores radares de ondas milimétricas e sua aplicação em mapeamento de espaços em três dimensões. É feita uma exposição de algumas das várias tecnologias de mapeamento como comparação com a tecnologia alvo deste trabalho. Os conceitos básicos da tecnologia radar de ondas milimétricas são descritos e em seguida o funcionamento específico dos sensores da empresa Texas Instruments. Por fim um conjunto de experimentos dos sensores radares é efetuado em um ambiente controlado, mostrando que o sensor radar consegue mapear espaços em três dimensões, mas é menos preciso se comparado a tecnologias lasers.

Palavras Chaves: Sensor, Radar, Nuvem, Pontos.

ABSTRACT

This work presents the study of software and hardware for millimeter wave radar sensors and their application in 3-dimensional space mapping. An exhibition is made of some of the various mapping technologies as a comparison with the target technology of this work. The basic concepts of millimeter wave radar technology are described and then the specific operation of the sensors of the company Texas Instruments. Finally, a set of experiments on radar sensors is performed in a controlled environment showing that the radar sensor can map spaces in 3 dimensions, but is less accurate compared to laser technologies.

Keywords: Sensor; Radar; Software; Cloud; Points

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

Símbolos Latinos

A	Área	[m^2]
Cp	Calor específico a pressão constante	[kJ/kg.K]
h	Entalpia específica	[kJ/kg]
$\dot{m}$	Vazão mássica	[kg/s]
T	Temperatura	[^oC]
U	Coeficiente global de transferência de calor	[$W/m^2.K$]

Símbolos Gregos

□	Difusividade térmica	[m^2/s]
□	Variação entre duas grandezas similares	
□	Densidade	[m^3/kg]

Grupos Adimensionais

Nu	Número de Nusselt
Re	Número de Reynolds

Subscritos

amb	ambiente
ext	externo
in	entrada
ex	saída

Sobrescritos

$\dot{}$	Variação temporal
$\bar{}$	Valor médio

Siglas

ABNT	Associação Brasileira de Normas Técnicas

CAPÍTULO 1 – INTRODUÇÃO

1.1 Descrição do problema

A medição dos espaços 3D é um problema que vem ganhando uma atenção maior conforme o campo da robótica avança. Para desenvolver a programação de um robô, deve se levar em conta o espaço no qual esse equipamento será inserido. Na maioria das vezes conhecemos as dimensões desse espaço, ou utilizamos as informações disponíveis para realizar essa programação. Mas, em geral, as dimensões são obtidas manualmente e é feita uma aproximação desse espaço em um modelo 3D. Com o avanço da automação, é necessário o mapeamento de espaços mais precisos e de aquisição mais rápida. Uma das técnicas para esse mapeamento é a aquisição de nuvem de pontos a partir de sensores para a geração de superfícies. Usando essa técnica, o mapeamento de espaços 3D com precisão na ordem dos centímetros necessita de uma geração de nuvem de pontos com uma precisão igual ou maior, além de uma velocidade de aquisição que possibilite a aplicação em sistemas autônomos como robôs de navegação.

Uma prática comum para a criação de um modelo 3D é a utilização de imagens feitas por câmeras, que são processadas e convertidas em uma nuvem de pontos. Esse método tem uma precisão limitada e geralmente é acompanhado de outro sensor ou medições manuais para complementação. Ocorre que a captura de imagens gera arquivos pesados, sobrecarregando a memória e o processamento, o que dificulta a portabilidade e a automação do processo.

Existem também os sensores que fazem a medição de distância de objetos, como os ultrassônicos, sensores fotoelétricos e sensores lasers; este último sendo mais empregado em espaços abertos para medição topográfica. No caso dos ultrassônicos e fotoelétricos, a precisão é baixa e os sensores sofrem interferência do ambiente, resultando em ruído nas medições.

Já os sensores lasers são fortemente presentes e têm várias aplicações, seja para nuvem de pontos precisa ou mapeamento topográfico, que é a construção do relevo de uma região. Os lasers também são usados na tecnologia chamada de LIDAR, Light Detection and Ranging, e são concorrentes fortes dos sensores radares,

por serem precisos nas aplicações em várias áreas. Mas o objetivo do trabalho é explorar as ondas milimétricas para que existam mais opções de tecnologia, no futuro, para o desenvolvimento de nuvem de pontos.

Os sensores radares *mmWave* são relativamente novos no mercado. A proposta é de um sensor altamente preciso, que sofre pouca ou nenhuma interferência, com informações simplificadas, ocupam pouca memória e têm baixo consumo de energia. Além disso, permitem um arco de medição, minimizando a movimentação do sensor ou instalação de partes móveis e, consequentemente, com menor ocorrência de erros de medição; ou seja, todas as vantagens dos anteriores. Assim, serão mostradas a eficiência e as possibilidades de utilização desse sensor.

1.2 Objetivos

O objetivo deste trabalho é descrever a tecnologia de ondas milimétricas, proporcionar uma noção do funcionamento dos sensores utilizados e mostrar as vantagens e desvantagens deles. Mais especificamente, será mostrada a teoria da utilização das ondas milimétricas, com ênfase em como o sensor trata esses dados e devolve as informações.

1.2.1 Objetivos específicos

Podemos dividir os objetivos específicos deste trabalho em cinco:

- Breve descrição de algumas tecnologias existentes para comparativo com a tecnologia radar de ondas milimétricas.
- Explicar o conceito básico da tecnologia radar de ondas milimétricas.
- Explicar e mostrar o caminho de dados do sensor radar escolhido
- Explicar o processo de escolha do sensor utilizado neste trabalho
- Demonstrar o funcionamento do sensor através de experimentos

1.3 Estrutura do trabalho

Este trabalho está dividido em quatro eixos principais, sendo o primeiro eixo, a

teoria envolvendo o sensor e sua tecnologia. O segundo eixo trata da calibração do sensor. No terceiro eixo será apresentada a amostragem de pontos, utilizando o Matlab e, por fim, no quarto eixo, a utilização de programas de terceiros para o processamento da nuvem de pontos.

CAPÍTULO 2 – INTRODUÇÃO TEÓRICA

Neste capítulo faremos primeiro uma breve introdução a algumas das principais tecnologias existentes utilizadas no mercado, para que seja possível estabelecer uma comparação com a tecnologia de ondas milimétricas. Em seguida será explicado o conceito base do funcionamento das ondas radares de modulação contínua e o caminho de dados do sensor escolhido para este trabalho. Por fim, serão mostradas brevemente algumas das muitas técnicas de transformação de nuvem e pontos em superfícies.

1.4 Tecnologias existentes para medição de ambientes

1.4.1 Tecnologias de processamento de imagem

Uma nuvem de pontos em três dimensões pode ser retirada usando um conjunto de imagens. Essa tecnologia é bastante acessível, pois são necessárias somente câmeras digitais. Entretanto, há considerável perda de precisão, bem como a ocorrência de significativa interferência ambiental. Podemos contar também com a utilização de dispositivos que geram imagens de calor, sendo o princípio de tratamento da imagem o mesmo, somente com informações diferentes.

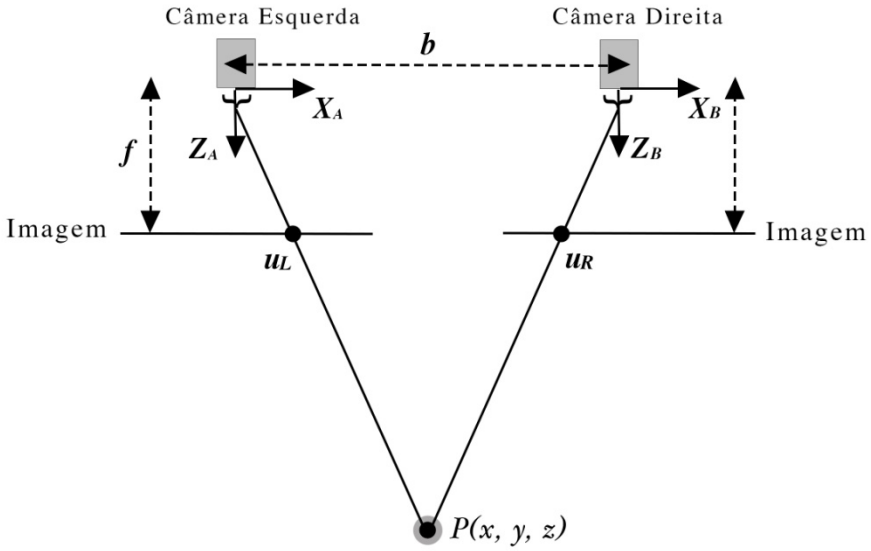


Figura 2.1 – Esquema de funcionamento do método de visão estéreo. [12]

Uma das técnicas é a utilização de duas câmeras a uma distância *b* entre as duas. Ao analisar as pequenas diferenças entre as imagens das câmeras, é possível determinar as distâncias dos pontos na imagem. Esta técnica é chamada de visão estéreo. A vantagem é que o processo é considerado passivo, ou seja, menos energia é gasta para efetuá-la.

Outra técnica é o conhecimento prévio das dimensões do ambiente ou de algum objeto no ambiente, seja por meio de outro sensor ou medições reais. Assim, é possível estimar a distância entre os pixels e criar uma nuvem de pontos com a ajuda das informações de cor de cada pixel. Essas técnicas funcionam bem em ambientes controlados ou já conhecidos, mas caem em eficiência em ambientes com perturbações, como luz ou poeira.

1.4.2 Tecnologias lasers

Os sensores lasers são considerados referência atualmente, quando se trata de geração de nuvem de pontos e leitura de superfícies. Uma das diferenças em relação aos sensores radares é a frequência das ondas eletromagnéticas. Enquanto os sensores radares trabalham em uma escala de gigaheartz (GHz), os sensores lasers começam nas centenas de GHz e alcançam até Terahertz (THz). Para

mapeamento em três dimensões, há várias tecnologias utilizadas, mas existem duas que a literatura destaca: triangulação e os sensores de varredura a laser, também conhecido como LIDAR [13]. A seguir, eles serão introduzidos brevemente para que possamos comparar com os sensores radares.

1.4.2.1 Triangulação

O método de mapeamento 3D por triangulação consiste em um emissor laser que projeta padrões geométricos sobre o objeto ou superfície que se deseja escanear. Então, uma câmera digital é posicionada a uma distância conhecida do sensor, formando assim um triângulo entre o sensor e o objeto. A câmera captura o padrão geométrico projetado na superfície. Como sabemos a distância entre o emissor e a câmera e o ângulo de rotação do sensor, é possível determinar as coordenadas (x,y,z) de um determinado ponto na superfície através de análises geométricas e trigonométricas [14]. A Figura 2.2 mostra o sensor em ação.

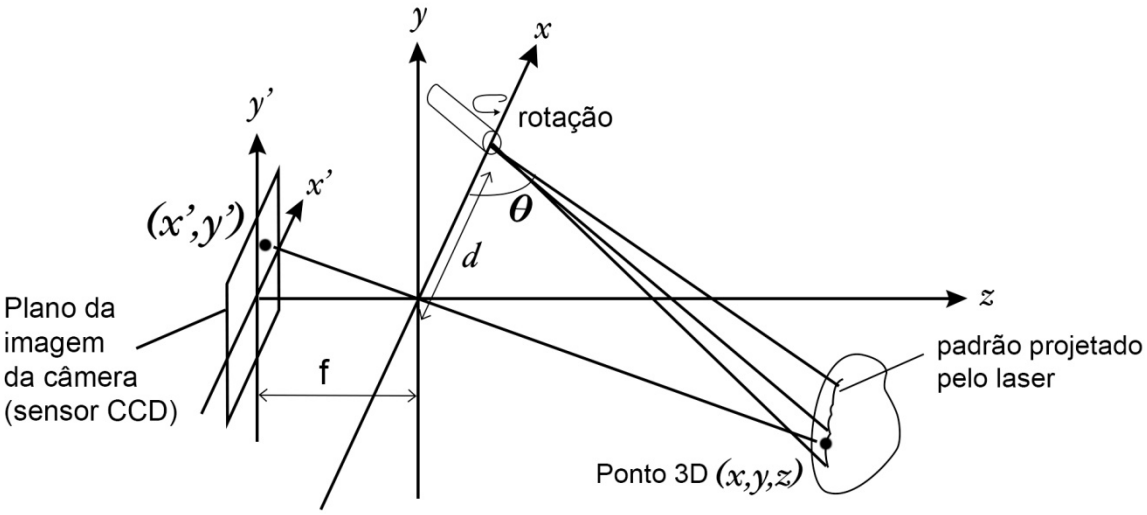


Figura 2.2 – Esquema de funcionamento do método de triangulação [12]

Esta tecnologia tem uma alta resolução, mas com alcance pequeno. Isso inviabiliza a utilização em ambientes abertos ou objetos grandes que estejam fora do alcance da câmera. É, porém, uma tecnologia de baixo custo se comparada a outras tecnologias.

1.4.2.2 LIDAR

Um dispositivo LIDAR é composto por um emissor de luz e um fotorreceptor. A técnica se baseia em determinar as distâncias entre o dispositivo e pontos da superfície. Ao incidir sobre a superfície, parte da luz emitida é refletida na direção do dispositivo, que identifica a onda através do fotorreceptor [12].

Existem duas maneiras de calcular a distância através desse método. A primeira delas é denominada time off light, na qual se mede o tempo entre a emissão de um pulso de luz e a recepção do pulso refletido. Com essa informação, podemos retirar a distância entre o sensor e o dispositivo, de acordo com a equação (1):

$$D = \left(\frac{1}{2}\right) ct \qquad (1)$$

Onde *c* é a velocidade da luz e *t* é o tempo medido entre emissão e recepção. A Figura 2.3 a seguir mostra o esquema de funcionamento dessa técnica.

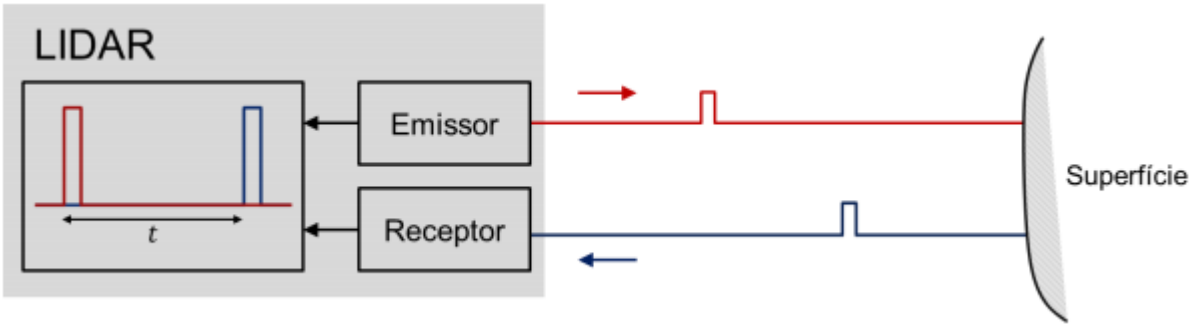


Figura 2.3 – Esquema de funcionamento do LIDAR – Time off light [12]

A segunda estratégia de cálculo da distância tem como princípio a diferença de fase entre uma onda emitida e a onda refletida na superfície. Para isso, a onda de luz é modulada senoidalmente e incidida sobre a superfície. A onda vai ser refletida pela superfície e lida pelo receptor; então, ela é comparada com a onda emitida. Sabendo que as duas têm a mesma frequência *f*, na equação (2), *c* seria a velocidade da luz, *f* é a diferença de fase entre as ondas emitida e recebida.

$$d = \frac{c\varphi}{4\pi f} \qquad (2)$$

A Figura 2.4 mostra o esquema de funcionamento.

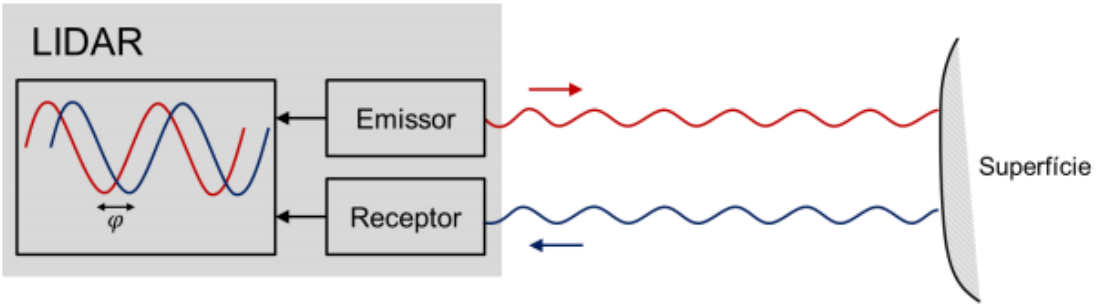


Figura 2.4 – Esquema de funcionamento do LIDAR – Diferença de fase [12]

No caso de varredura de superfícies, o sensor laser pode ser auxiliado por dispositivos optomecânicos. Se for adicionado um espelho poligonal rotacionado por um servo motor à frente do emissor, os dados recebidos podem ser processados por um módulo de processamento. À medida que o servo motor é acionado, o espelho realiza uma varredura em um plano da superfície. Esse tipo de sistema LIDAR é muito utilizado para varreduras bidimensionais, e alguns sistemas utilizam elementos ópticos em mais de uma dimensão para dados em três dimensões. No entanto, esse último é utilizado em topografia por ser muito pesado, apresentar um custo elevado e por ser lento, se comparado as outras tecnologias [12].

Pode-se ainda acoplar um motor elétrico a um LIDAR 2D, convertendo o sistema de varredura para 3D. Existem três métodos possíveis para realização desta montagem, cada um com suas vantagens e desvantagens. Como o objetivo deste trabalho não é o estudo de sensores lasers, será feito um breve resumo de cada montagem, mas Wulf e Wagner (2003) fazem uma análise detalhada destes métodos [12].

1.4.2.2.1 Rolling

Na configuração *rolling*, o dispositivo LIDAR é rotacionado em torno do eixo horizontal, coincidindo com a reta que divide pela metade o campo de varredura do

laser (Figura 2.5). A vantagem deste método é a montagem, já que o sentido de rotação é o mesmo do motor, o acoplamento mecânico entre estes elementos é relativamente mais simples, ocupa menos espaço e coloca o sensor perto do centro de rotação do motor, diminuindo o esforço do mesmo. Esse método tem uma grande quantidade de pontos à frente do sensor, onde se localiza o ponto focal; logo, essa região fica mais detalhada. Essa tecnologia torna-se perfeita para aplicações que pedem um nível de detalhe elevado nessa região.

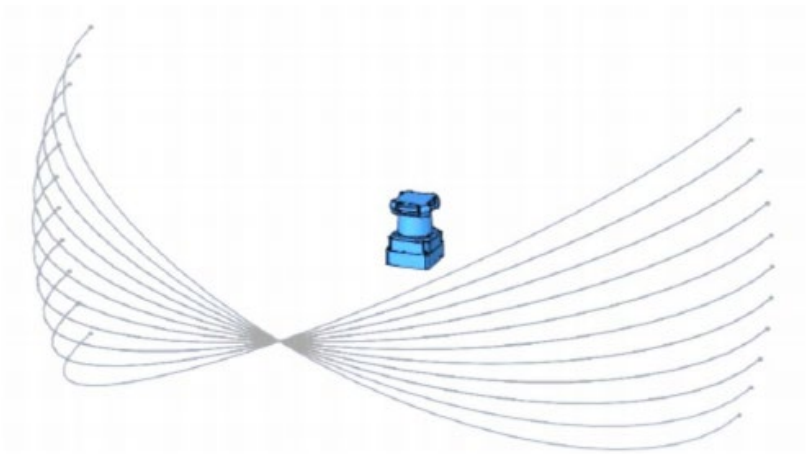


Figura 2.5 – Aspecto de cobertura configuração *rolling* [12]

1.4.2.2.2 Pitching

Na configuração *pitching*, o sensor é rotacionado em torno do eixo horizontal e perpendicular ao eixo de rotação do *rolling* (Figura 2.6). As regiões de maior densidade de pontos estão à direita e à esquerda do sensor, e a região frontal vai ter um mapeamento mais espesso e uniforme [12]. Por causa disso, os pontos à frente do sensor são detectados mais rapidamente, tornando-se adequado para aplicações onde o tempo de decisão com as informações precisa ser curto. Porém, o aparato mecânico é complexo e ocupa espaço maior.

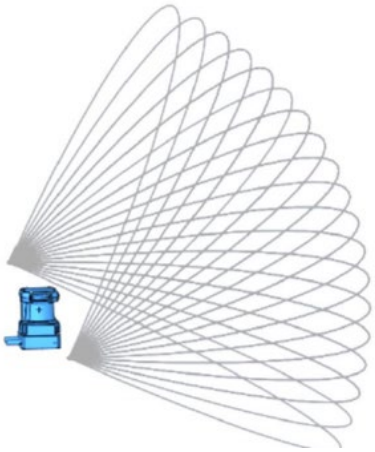


Figura 2.6 – Aspecto de cobertura configuração *pitching* [12]

1.4.2.2.3 Yawing

Por último, o método *yawing* baseia-se na rotação do sensor em torno do eixo vertical (Figura 2.7). Esse método permite um mapeamento em 360 graus, visto que as regiões com mais pontos estão localizadas acima e abaixo do sensor, tornando essa aplicação mais apropriada para ambientes que necessitam de uma visão maior que 180 graus. O acoplamento mecânico depende da configuração adotada. Caso seja com uma das laterais voltada para baixo, a implementação é parecida com o *pitching*. Se for com a parte de trás do sensor voltada para baixo, a montagem se assemelhará à de *rolling* (Figura 2.7).

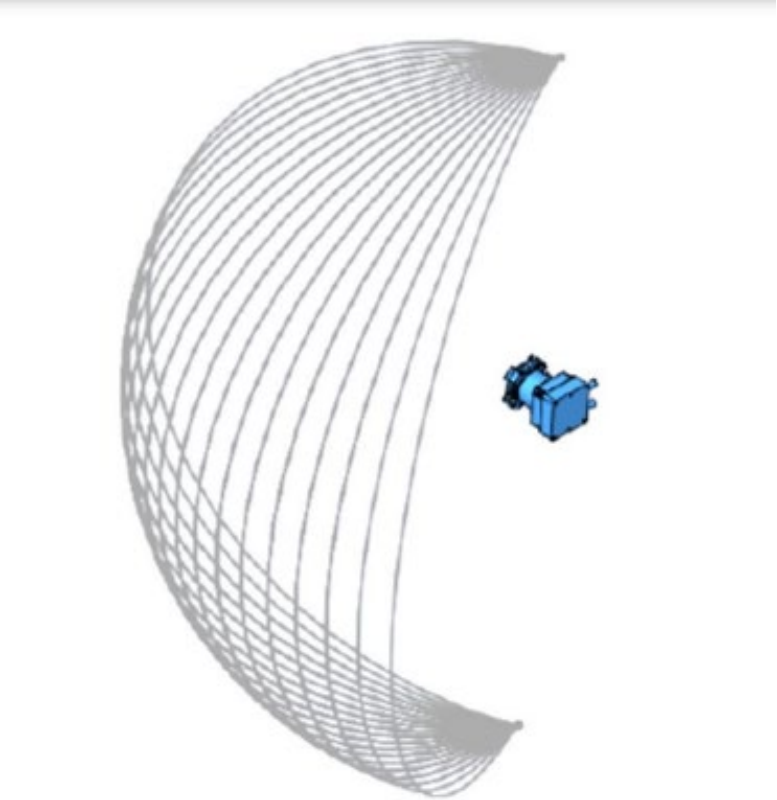


Figura 2.7 – Aspecto de cobertura configuração yawing [12]

1.4.3 Sensores Ultrassônicos

Sensores ultrassônicos têm o mesmo princípio de funcionamento dos sensores radares: a emissão de uma onda mecânica, com frequência superior a 20 Khz, na direção do objeto e à espera do retorno dela após um tempo *t*. Com essas informações, é possível calcular a distância até o objeto. O sensor ultrassônico pode ser utilizado para varreduras em duas dimensões, mas por ser muito sensível a ruídos do ambiente, como vento, poeira e outras vibrações na mesma frequência dele, tem aplicações para distâncias pequenas, se comparado aos sensores radares. Porém, no meio aquático, o ultrassônico tem maior eficiência, pois o som se propaga melhor. Essa tecnologia é conhecida como sonar.

Esse sensor também tem baixa precisão, pois tem dificuldade de detectar objetos pequenos. É preciso que o objeto seja grande o suficiente para refletir uma potência razoável de volta ao emissor, sem desviar a onda.

1.5 Sensores radares

Os sensores escolhidos para a elaboração do projeto são os sensores radares. A utilização desses sensores vem crescendo, principalmente no meio automobilístico, utilizado principalmente para medir a velocidade relativa dos carros. Também vem crescendo seu uso no meio industrial, pois é um sensor muito bom para mapear equipamentos em chão de fábrica, podendo aumentar o potencial para automatização.

Os sensores radares funcionam de forma parecida com os sensores ultrassônicos, ou seja, um emissor emite uma onda eletromagnética no meio, que rebate nos objetos e retorna com certo atraso e potência à antena receptora, e assim sendo possível mensurar a distância, velocidade e ângulo do objeto. Uma das diferenças entre os dois sensores é a frequência de operação, sendo que os sensores radares operam na faixa de 30GHz a 300GHz, fazendo com que interferências sejam menos frequentes e mais suaves.

A classe de sensores radares utilizada para o projeto é a de ondas milimétricas (*Millimeterwave - mmWave*), que usa ondas curtas eletromagnéticas.

Um radar *mmWave* transmite sinais com um comprimento de onda no range milimétrico. Então, este sinal tem um comprimento de onda curto no espectro eletromagnético, sendo uma das vantagens dessa tecnologia. Logo, o equipamento para a emissão e recepção de uma onda desse comprimento é relativamente pequena. Outra grande vantagem é a alta precisão, já que uma onda com uma frequência de 80 GHz tem aproximadamente 4 mm de comprimento de onda, possibilitando a habilidade de detectar movimentos com diferença de frações de milímetros. Os sensores escolhidos estão na faixa de 60GHz até 80Ghz.

Este sensor implementa uma classe tecnológica de onda milimétrica chamada de frequência contínua de modulação de onda (FMCW). Como o nome sugere, este sensor emite uma onda contínua para medir a posição, ângulo e velocidade do objeto, bem diferente dos sensores de modulação por pulso de curta duração. A vantagem dessa tecnologia é a medição de velocidade mais precisa pelo efeito *Doppler*.

1.6 Introdução teórica aos sensores radares

A seguir será explicado de forma simplificada como funciona o cálculo da distância, velocidade e ângulo pelo hardware do sensor.

1.6.1 Processo para o cálculo da distância

O conceito fundamental no sistema de radares é o envio de ondas eletromagnéticas em certa direção e a detecção da refração das mesmas ondas. No caso do sensor utilizado, as ondas são enviadas sem pausa, ou seja, um sinal contínuo, não um pulso como na maioria dos sensores desse tipo. Logo, para não haver interferências, aumentamos a frequência do sinal com o tempo. Esse tipo de sinal também é chamado de *chirp* [1].

A Figura 2.8 mostra um sinal desse tipo, com amplitude constante, mas frequência variante no tempo.

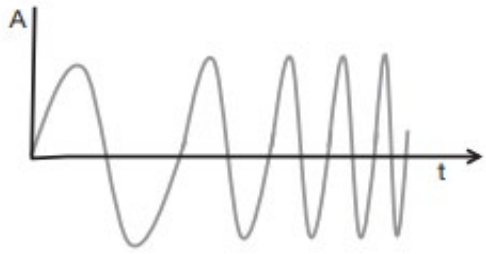


Figura 2.8 – Sinal *chirp* [1]

A Figura 2.9 mostra o mesmo sinal *chirp* com a frequência em função do tempo. Ele é caracterizado por uma frequência de início (*fc*), banda (*B*) e duração (*Tc*).

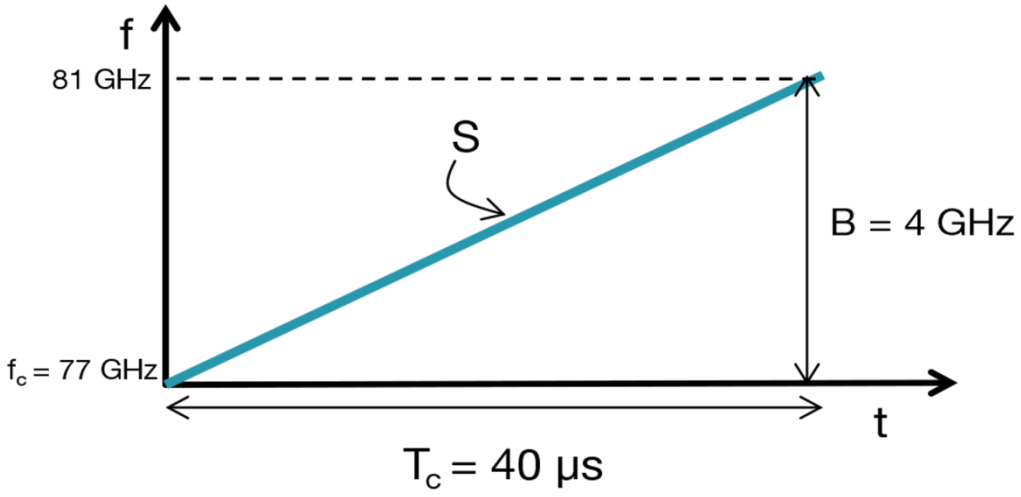


Figura 2.9 - Sinal *chirp* em frequência na função do tempo [1]

A reta (*S*) mostra a velocidade de variação da frequência do sinal. Na Figura 2.9 temos *fc* = 77 Ghz, *Tc* = 40 *µs*, *B* = 4 Ghz e *S*= 100 Mhz/*µs*.

Um sistema de radar FMCW transmite um sinal *chirp* e captura o sinal refletido pelos objetos no caminho. A Figura 2.10 mostra um diagrama de blocos simplificado de componentes de rádio frequência.

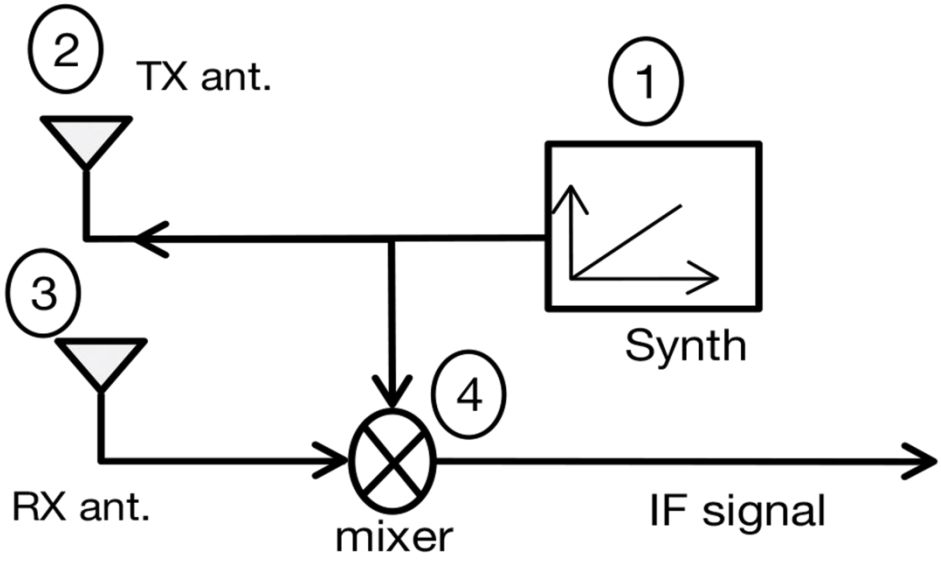


Figura 2.10 -Diagrama de blocos de um Radar FMCW [1]

O sistema vai operar da seguinte maneira:

- O sintetizador (*synth*) gera um sinal *chirp*.
- O sinal é transmitido pela antena (*TX* ant.).
- O sinal é refletido por um objeto, gerando um sinal *chirp* refletido. Esse sinal então é captado pela antena de recepção (*RX* ant.).
- Um somador de sinais combina os dois sinais para gerar um sinal de frequência intermediária (*IF*).

Um somador de sinais é um componente eletrônico que combina dois sinais de frequências diferentes para gerar um novo sinal com uma nova frequência. Para dois sinais senoidais na entrada do somador de sinais *x1* (1) e *x2* (2):

$$x1 = sin(\omega 1t + \phi 1) \quad (3)$$

$$x2 = sin(\omega 2t + \phi 2) \quad (4)$$

A saída x_{out} terá uma frequência instantânea igual à diferença das frequências instantâneas dos sinais de entrada. A fase da saída será igual à diferença das fases dos sinais de entrada, gerando a equação (5):

$$x_{out} = \sin[(\omega_1 - \omega_2)t + (\phi_1 - \phi_2)] \quad (5)$$

O diagrama de frequência em função do tempo da Figura 2.11 mostra o sinal enviado pela antena de emissão (*TX*) e o sinal recebido pela antena de recepção (*RX*). É possível notar que o sinal lido é o sinal emitido com um atraso.

O tempo de *delay* (*t*) pode ser matematicamente deduzido e a equação (6) aparece:

$$t = \frac{2d}{c} \quad (6)$$

Onde *d* é a distância do objeto até o sistema radar e *c* é a velocidade da luz.

Para obter o sinal resultante representado em frequência do somador de sinais, basta subtrair as duas retas na Figura 2.11. A distância entre as retas é fixa, significando que a reta resultante da subtração vai ter uma frequência fixa. Se olharmos a Figura 2.11, veremos que essa frequência será *St*, lembrando que essa frequência só é válida enquanto os dois sinais se sobrepõem [1].

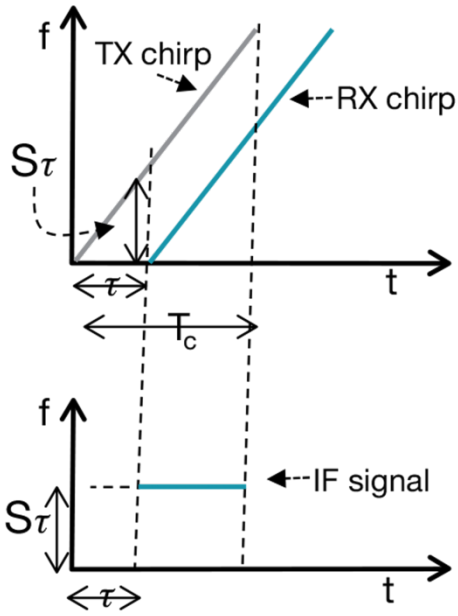


Figura 2.11 – A frequência resultante é constante [1]

O sinal de saída é um sinal senoidal, pois tem uma frequência constante.

A fase inicial do sinal de saída é a diferença entre o sinal enviado pela antena *TX* e o sinal recebido pela antena *RX*, no tempo correspondente ao começo do sinal de saída, e será igual à equação (7):

$$\phi_o = 2\pi f_c\, t \qquad (7)$$

Matematicamente, podemos substituir a equação (6) na equação (7), e fazendo algumas simplificações teremos a equação (8):

$$\phi_o = \frac{4\pi d}{\lambda} \qquad (8)$$

Resumindo, para um objeto a uma distância *d* do radar, o sinal de saída será uma onda senoidal (equação 9):

$$Asin(2\pi f_0 t + \phi_0) \qquad (9)$$

onde $f_0 = \frac{s2d}{c}$ e $\phi_0 = \frac{4\pi d}{\lambda}$.

Assumimos até agora que o radar só está detectando um único objeto. Vamos

analisar caso haja vários objetos a serem detectados. A Figura 2.12 mostra três diferentes sinais de retorno recebidos por três objetos diferentes. Cada sinal *chirp* tem um atraso diferente, representando cada objeto diferente. Os diferentes sinais recebidos vão gerar um sinal de saída com vários harmônicos, cada um com uma frequência constante diferente [1].

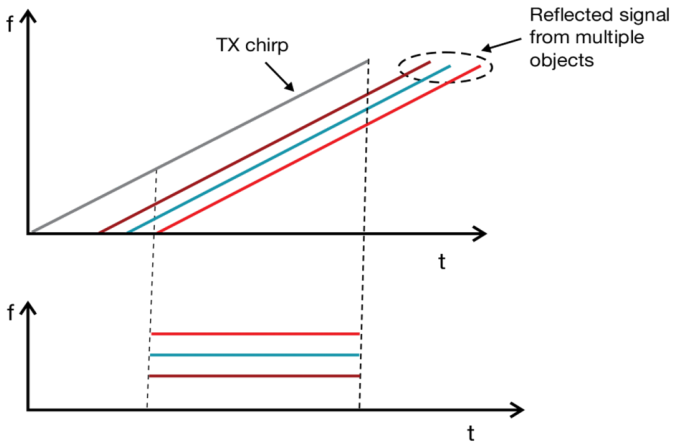


Figura 2.12 - Vários objetos detectados [1]

Neste caso, o sinal deve ser processado usando uma Transformada de Fourier para separar os diferentes harmônicos. Este processamento vai resultar em um espectro de frequência que vai separar os picos de cada harmônico, mostrando a presença de cada objeto em suas respectivas distâncias.

1.6.1.1 Resolução de contraste

A resolução de contraste do sistema radar é a habilidade de diferenciar dois ou mais objetos. Quando dois objetos se movem para perto um do outro, em certo ponto, o sistema radar não vai conseguir distingui-los como objetos separados. A Teoria da Transformada de Fourier diz que, para aumentar essa resolução, devemos aumentar o comprimento do sinal de leitura [1]. No caso do sensor utilizado, para aumentarmos o comprimento do sinal de saída, devemos aumentar o tamanho da banda proporcionalmente. Com este aumento, teremos um sinal de saída com dois ou mais picos representando os objetos.

A Teoria da Transformada de Fourier também postula que em uma determinada janela de observação (*T*) podem ser identificados componentes de frequência que estão separados por mais que *1/T* Hz [1]. Isso significa que um sinal de saída com dois harmônicos pode ser resolvido em frequência, desde que a diferença de frequência respeite a equação (10):

$$\Delta f > \frac{1}{Tc} \quad (10)$$

Onde *Tc* é a janela de observação.

Como $\Delta f > \frac{S2\Delta d}{c}$, a equação (10) pode ser representada por $\Delta d > \frac{c}{2Stc} = \frac{c}{2B}$, sendo *B* = *S*Tc*.

Logo, a resolução de contraste será expressa pela banda dada pelo sinal *chirp*.

$$d_{Res} = \frac{c}{2B} \quad (11)$$

Então, um sinal *chirp* com alguns Ghz de frequência vai ter uma resolução na casa dos centímetros.

1.6.2 Processo para o cálculo da velocidade

A medida de velocidade de um objeto não é significativa para o projeto, já que estamos assumindo que todos os objetos presentes no ambiente estarão em repouso. Por isso, a explicação de como funciona será resumida [1].

Para melhor simplificar utilizaremos a notação de fasor.

Para medir velocidade, um sistema radar FMCW transmite dois sinais *chirp* separados por *Tc*; cada um irá refletir no objeto em movimento e o sinal refletido deve ser processado por uma FFT (Transformada de Fourier Rápida) para detectar a distância do objeto. A distância correspondente para cada sinal *chirp* terá o mesmo pico, na mesma posição, depois da Transformada de Fourier. Mas cada pico terá uma fase diferente; a diferença entre as duas fases corresponde com o movimento do objeto. A diferença de fase pode ser retirada da equação (8), resultando na equação (12):

$$\Delta\phi = \frac{4\pi v T c}{\lambda} \quad (12)$$

Podemos retirar a equação da velocidade utilizando a equação (13):

$$\mathrm{v} = \frac{\lambda \Delta\phi}{4\pi T c} \quad (13)$$

Já que a medição da velocidade é feita através do módulo, pode acontecer ambiguidade, ou seja, a medição só será correta se $|\Delta\phi| < \pi$; usando isso e a equação (13), podemos deduzir que a velocidade será sempre $\mathrm{v} < \frac{\lambda}{4Tc}$. Logo, a velocidade máxima será dita pela equação (14).

$$v_{max} = \frac{\lambda}{4Tc} \quad (14)$$

1.6.3 Processo para o cálculo do ângulo

Um sistema radar pode estimar o ângulo de um sinal refletido de acordo com a Figura 2.13. Este ângulo também é chamado de ângulo de chegada.

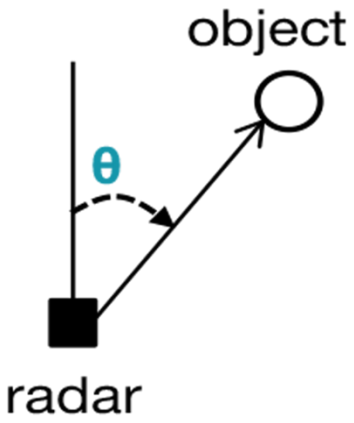


Figura 2.13 – Ângulo de chegada [1]

A estimativa de um ângulo baseia-se na pequena diferença de distância de um mesmo sinal recebido por duas antenas *RX* diferentes. Isso resulta em uma pequena diferença na fase do pico da Transformada de Fourier dos sinais. Essa diferença é usada para determinar o ângulo do objeto em relação ao sistema radar, utilizando duas antenas como mostra a Figura 2.14.

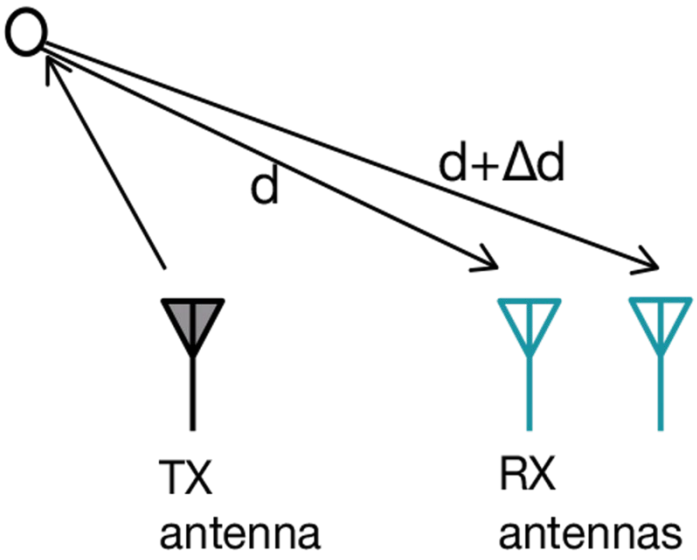


Figura 2.14 – Duas antenas recebendo o sinal refletido [1]

Nesta configuração, podemos determinar matematicamente o ângulo e temos como resultado a equação (15).

$$\Delta\phi = \frac{2\Delta d\pi}{\lambda} \quad (15)$$

Assumindo que a onda viaja em um plano para facilitar a geometria, sabe-se então que $\Delta d = i \sin(\theta)$, onde lambda é a distância entre as antenas. Logo, o ângulo de chegada pode ser medido com a equação (16):

$$\theta = \sin^{-1}\left(\frac{\lambda\Delta\phi}{2\pi l}\right) \quad (16)$$

Como resultado, podemos ver que o valor do ângulo fica cada vez mais preciso conforme o ângulo for menor.

1.6.4 Configuração espacial da antena

Cada versão de EMV tem uma configuração de antenas própria, com posição e quantidade. A Figura 2.15 mostra a configuração espacial da placa adquirida e

utilizada neste trabalho. Onde lambda é igual a 4 milímetros. As antenas Tx1 e Tx3 estão relacionadas com as informações referentes a distância e ângulo de Azimuth, enquanto a antena Tx2 com as informações que ajudam no cálculo da distância, mas também informações sobre o ângulo referente à altura do objeto detectado. Cada antena tem quatro emissores/receptores, ou seja, cada emissão ou recepção de sinal vai conter 4 componentes.

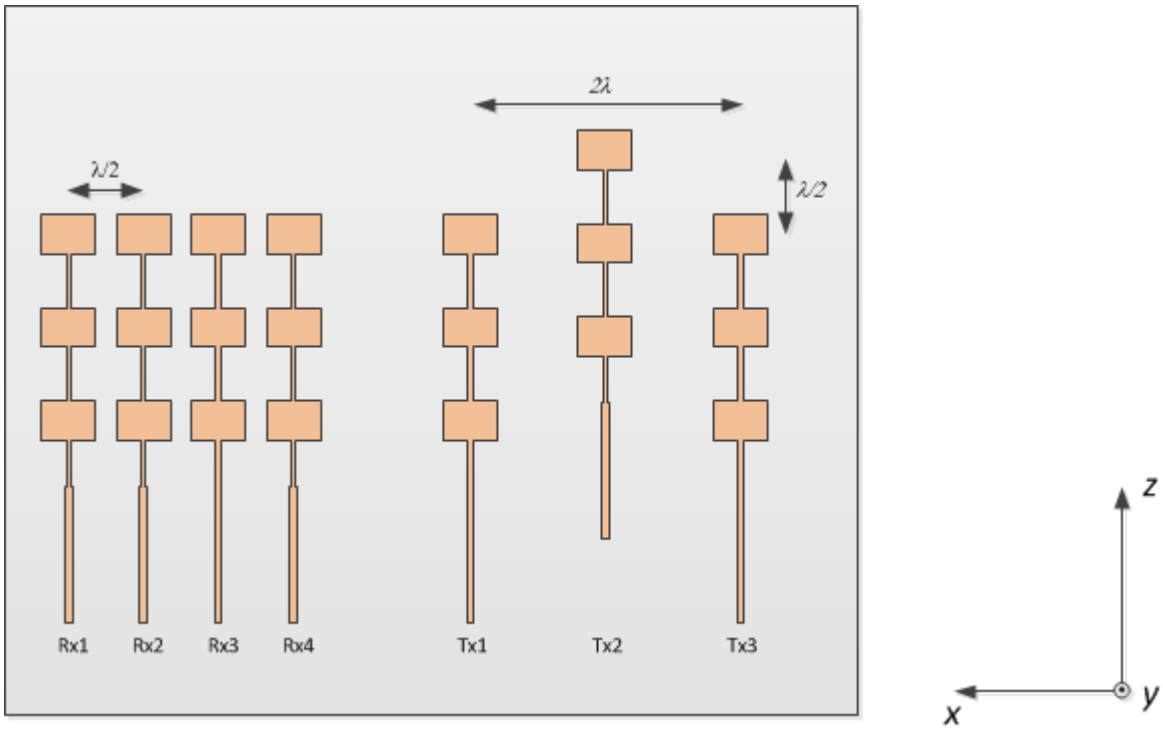


Figura 2.15 - Configuração espacial das antenas de recepção e emissão [6]

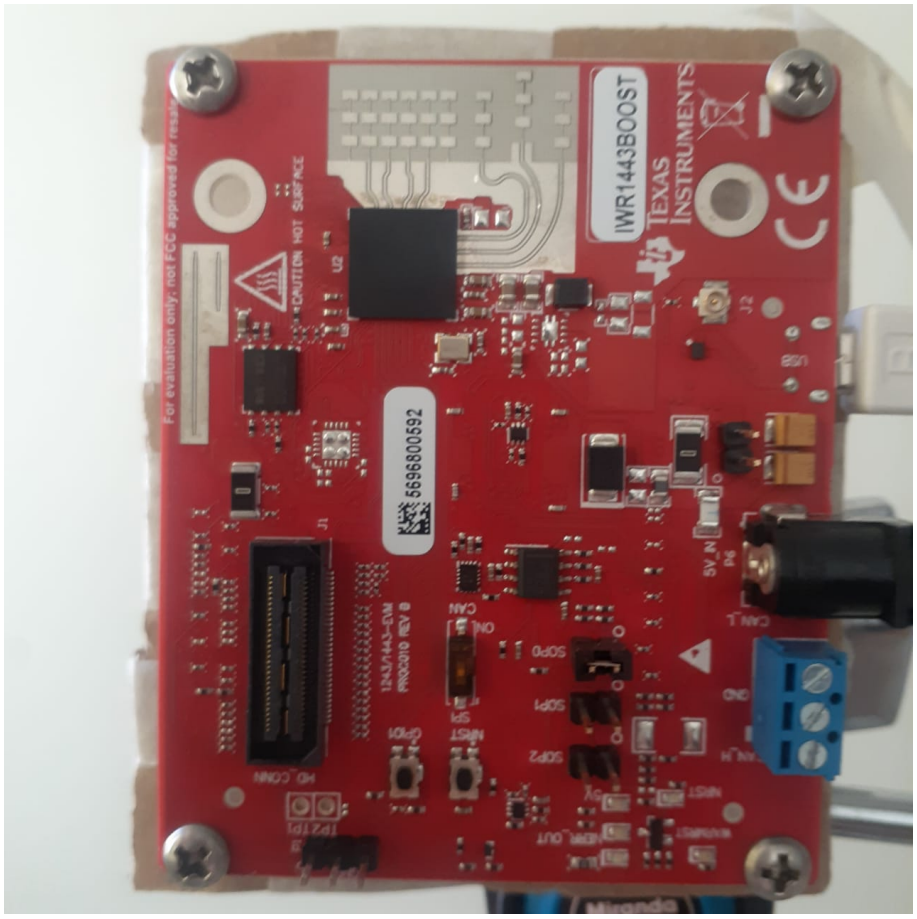


Figura 2.16 – visão frontal do sensor

1.7 Introdução ao funcionamento do sensor (14XXIWR)

O processo básico do funcionamento do sensor foi exposto no capítulo anterior. Ele possui uma programação e um caminho de dados específico para a implementação do cálculo da distância e velocidade. A partir de agora, o *chip* 1443 mais as antenas de emissão e recepção com os componentes que constroem o sensor serão chamados EVM (*Evaluation module*).

Nesta seção será exposto o funcionamento do caminho de dados interno do sensor.

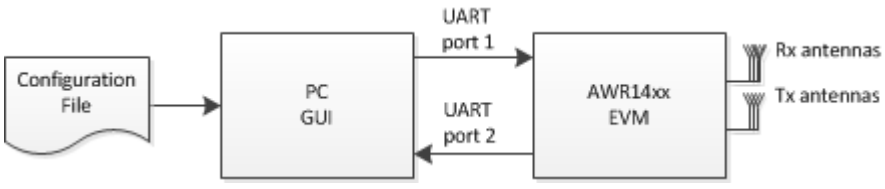


Figura 2.17 – Comunicação entre EVM e o computador [6]

Quando iniciamos o sensor conectado ao computador via porta USB, são executadas algumas tarefas que carregam as informações de configuração das antenas e do chip. Então, o sensor entra em um loop de medições, aguardando novas configurações ou paradas, caso sejam requisitadas pelo usuário. Internamente, isso é controlado por uma máquina de estados (Figura 2.18).

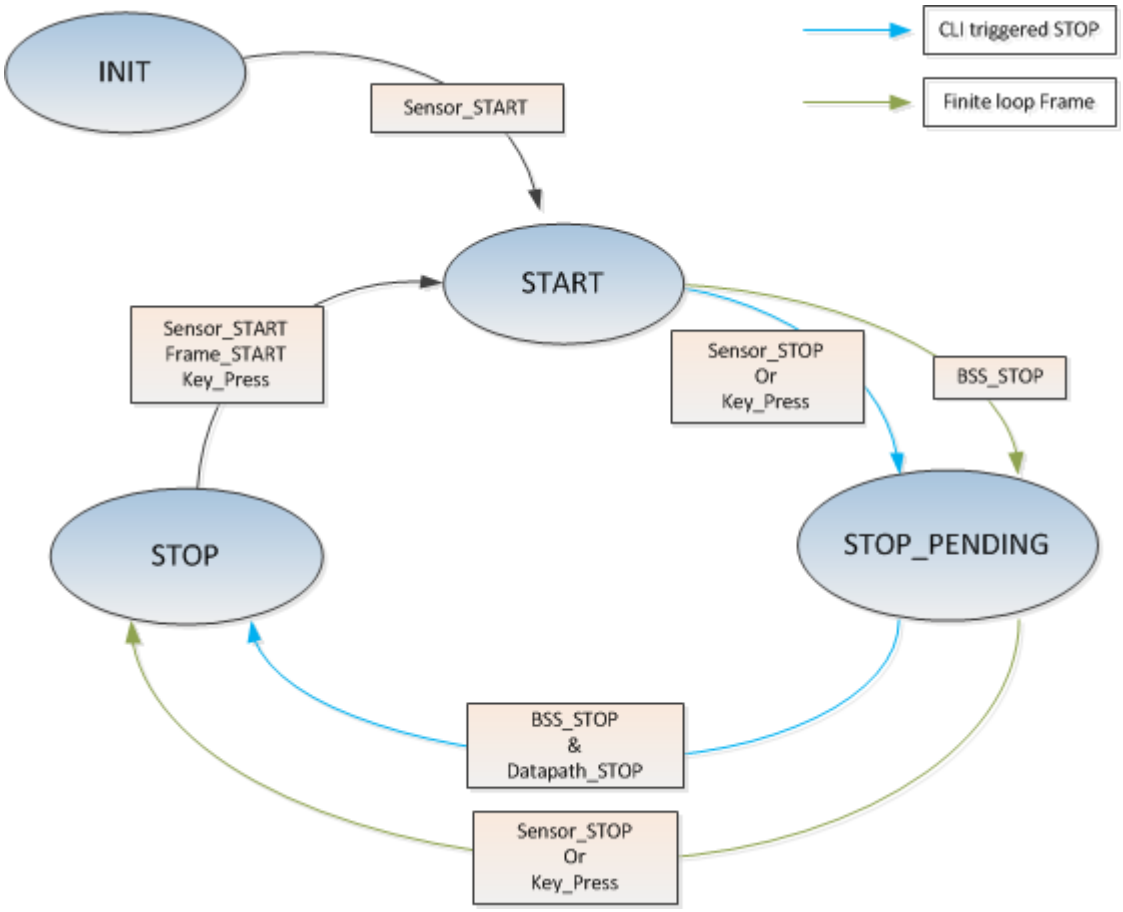


Figura 2.18 – Máquina de estados interna [6]

Onde o evento de começo do sensor é acionado pelo carregamento das configurações ou por comando efetuado pela comunicação USB, pela comunicação também vem o comando de parada do sensor. Por fim, o comando de parada do caminho de dados acontece quando ocorre um erro, como loops infinitos ou erro de parâmetros de configuração. O evento que monitora esses erros é chamado de BSS. Quando o sensor está no estado parado, ele fica em espera de comando de começo.

Quando o sensor termina de adquirir os dados, acontecem o processamento e a transmissão de velocidade de 1D, 2D, CFAR, Azimuth e Elevação em três

coordenadas espaciais (x,y,z), dos objetos detectados. O tempo que o sensor demora a ler é chamado de período de aquisição; o período entre novas medidas é o tempo que o sensor demora para executar todos os cálculos; e a soma dos dois é chamada de Frame. A versão demonstrativa também pode ser configurada para apenas detecção de 2D (velocidade e coordenadas x,y) [6]. Isso pode ser visto na Figura 2.19, que mostra o caminho de dados das informações coletadas pelas antenas até o envio do processamento feito pela porta USB.

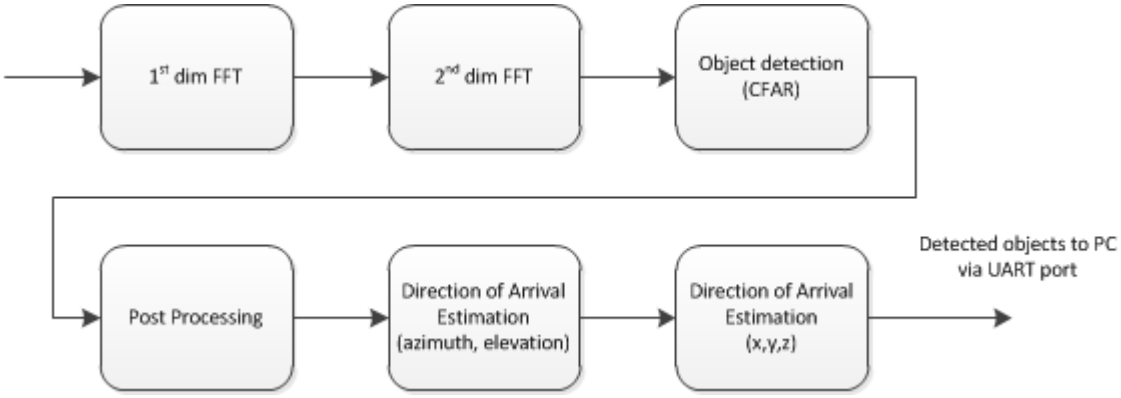


Figura 2.19 – Visão do caminho de dados [6].

O processamento de 1D começa com a emissão de uma onda *chirp* pela antena *Tx1*; isso é chamado de *ping*. Em seguida, a mesma onda *chirp* é transmitida pela antena *Tx3*; isso é chamado de *pong*. Então, a antena *Tx2* envia este sinal ao ambiente, outro *ping*. Esse processo se repete até que sejam completadas dezesseis vezes [6]. Cada onda *chirp* é detectada pelas quatro antenas *Rx*. Essas informações de frequência, conforme são recebidas pelas antenas *Rx*, passam de um conversor analógico para digital, e as informações digitais passam por uma Transformada Rápida de Fourier [11] de 256 pontos e são armazenadas em memória. Após as dezesseis interações, o processo está completo e o software ativa a próxima etapa. Em memória haverá uma matriz cúbica (Figura 2.20) com as informações das Transformadas de Fourier de 1D.

Figura 2.20 – Organização da matriz cúbica de radar [6]

O processamento de 2D pode ser executado de duas formas. A primeira é o processamento de baixa precisão, onde cada linha da matriz vai sofrer outra operação de Transformada Rápida de Fourier de 2 dimensões (FFT-2D) de 16 bits, e em seguida será efetuada uma operação logarítmica de base 2 dos valores absolutos. Ao fim, são somados pares sequenciais das linhas da matriz resultante. O resultado é salvo na memória. Esse resultado é a distância de *Doppler* em magnitude logarítmica. Já o resultado da FFT-2D é salva no lugar da matriz de FFT-1D. O resultado da FFT-2D tem uma precisão de 24 bits, mas para efeitos de armazenamento, esta precisão é convertida para 16 bits. A operação logarítmica então é efetuada em cima desse resultado, o que aumenta picos de ruídos na detecção de objetos [6].

Uma solução é a utilização do processamento de alta precisão. A diferença entre os dois tipos de processamento é que o resultado da FFT-2D passa a ser salvo como um valor de 32 bits antes da operação logarítmica; isso consequentemente aumenta consideravelmente a quantidade de memória necessária. Então, uma solução é não salvar o resultado da FFT-2D na memória, o que significa que em memória irão ter a distância *Doppler* e a matriz de FFT-1D. Mas isso vem com um custo de que no cálculo de ângulo de chegada a FFT-2D tenha que ser recalculada. Então é uma escolha entre tempo de processamento e precisão [6]. O sensor é programado para a realização de alta precisão automaticamente.

A saída do processamento de duas dimensões vai estar em formato Q9, isso significa que 9 bits são bits fracionários.

Em seguida, o processo de detecção de objeto é acionado. Neste processo, a matriz *Doppler* em magnitude logarítmica é analisada, passando por uma operação CFAR (*Constant false alarm rate*). A operação CFAR é um algoritmo adaptativo usado em sistemas radares para a detecção de objetos em um ambiente com ruídos, ecos e interferências. O princípio é criar um limiar para o resultado da matriz de distância *Doppler*; se esse limiar for muito baixo, ruídos e interferências serão detectados como objetos; se ele for muito alto, alguns objetos podem deixar de ser detectados. Em ambientes estáticos, é possível escolher um valor fixo para este limite, mas na maioria dos casos, os ruídos podem variar bastante de intensidade, e então um algoritmo que modifique este limiar para manter uma probabilidade de alarmes falsos. Esse algoritmo é chamado de CFAR [7].

O algoritmo mais simples de CFAR é fazer uma média entre os valores próximos do valor testado para o cálculo de um limiar. Existem técnicas mais avançadas de CFAR onde se medem rigorosamente as estatísticas de ruídos do ambiente para a criação de limiares dinâmicos.

Após o processo de CFAR, as informações de distância e *Doppler* são salvas na memória.

A etapa de pós-processamento ocorre para aplicar configurações escolhidas pelo usuário em tempo real. Como agrupamento de objetos detectados, caso o usuário queira somente um ponto que defina onde o obstáculo foi detectado. No caso de geração de nuvem de pontos, queremos todos os pontos detectados pelo sensor.

No pós-processamento também são preparados os dados para o cálculo do ângulo de chegada e descarte de objetos fora da distância estabelecida pelo usuário. O preparo para a próxima operação é chamado de compensação de *Doppler*, e é feito ao dividir os resultados contidos na matriz resultante da FFT-2D em ângulos de elevação e de Azimuth, utilizando a matriz distância-*doppler*. Isso é feito separando o resultado da FFT-2D de acordo com o que foi recebido pelas antenas; se foi recebido pelas antenas *Tx1* e *Tx3*, a FFT-2D delas é utilizada para o cálculo do ângulo de Azimuth; se foi recebido por *Tx2*, é utilizada para o cálculo do ângulo de elevação. Então, cada informação da antena *Tx2* é rotacionada em 1/3 da diferença de fase *doppler* entre dois *chirps* consecutivos, e cada informação da antena *Tx3* é rotacionada por 2/3 desse valor [6]. O resultado dessas operações é salvo em memória.

Como o uso deste sensor é em um ambiente sem objetos em movimento, essa operação não retornará quase nenhuma mudança nos dados.

Ao fim, esse processo também salva o número de objetos em memória.

O último processamento sobre os dados é o cálculo do ângulo de chegada da onda milimétrica. A primeira etapa é pegar as duas matrizes geradas pelo processo anterior, selecionar os valores complexos das FFTs-2D e salvá-los em memória.

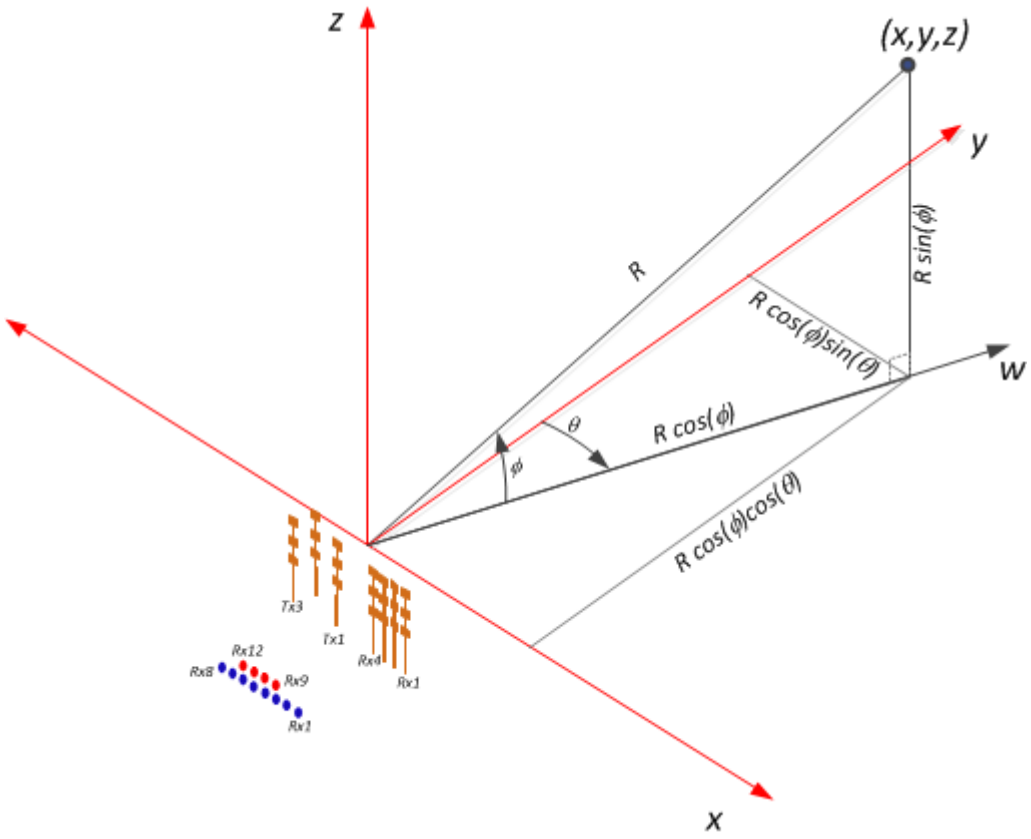


Figura 2.21 – Cálculo do ângulo de chegada e coordenadas do objeto detectado [6]

Utilizando a Figura 2.21 pode-se retirar as equações (17), (18) e (19):

$$x = R\cos(\phi)\sin(\theta) = R\frac{Wx}{\pi} \quad (17)$$

$$z = R\frac{Wz}{\pi} \quad (18)$$

$$y = \sqrt{R^2 - x^2 - z^2} \quad (19)$$

Onde *R* é igual a:

$$R = k_r \frac{c*F_{samp}}{2*S*Nfft} \quad (20)$$

Sendo *kro* índice de distância (retirado na operação CFAR), *c* é a velocidade da luz, *Fsamp* é a frequência medida, *S* a inclinação da reta *chirp* e *Nfft* é o tamanho

da FFT – 1D [6].

Para achar os ângulos θ e ϕ são utilizadas as matrizes com os valores complexos das FFT-2D, onde os valores referentes ao Azimuth (antenas *Tx1* e *Tx3*) serão representados por:

$$A_1 e^{j\psi}[1 \;\; e^{jWx} \; e^{j2Wx} \; e^{j3Wx} \; e^{j4Wx} \; e^{j5Wx} \; e^{j6Wx} e^{j7Wx}] \quad (21)$$

Onde *A1* e *Ψ* são valores arbitrários do primeiro receptor da antena.

E os valores de elevação (antena *Tx2*) serão representados por:

$$A_2 e^{j(\psi+2Wx-Wz)}[1 \;\; e^{jWx} \; e^{j2Wx} \; e^{j3Wx} \;] \quad (22)$$

Wx é a diferença de fase entre duas medidas dos receptores das antenas e *Wz* é a diferença de fase entre o ângulo de Azimuth (*θ*) e a elevação desse receptor acima do plano de Azimuth. Utilizando o que vimos na configuração espacial da antena, podemos concluir que a distância entre ondas interceptando dois receptores da antena *Rx2* é igual a (sabendo que os receptores da antena de elevação têm uma distância *λ* /2):

$$AB = \frac{\lambda}{2} \sin \phi \quad (23)$$

Então utilizando a Figura 2.21, *Wz* vai ser igual a:

$$Wz = \pi \sin \phi \quad (24)$$

Para retirar o valor de *Wx* fazemos uma trigonometria básica utilizando a Figura 2.21, resultando em:

$$Wz = \pi \sin\theta \ \ \cos\phi \quad (25)$$

Se for feita a Transformada de Fourier das equações (21) e (22) e retirarmos seus picos teremos, respectivamente:

$$P_1 = A_1 e^{j\psi} \quad (26)$$

$$P_2 = A_2 e^{j(\psi + 2Wx - Wz)} \quad (27)$$

P1 é o pico do sinal em *Wx* com uma fase de pico sendo *Ψ*. E *P2* também é o pico do sinal em *Wx*, mas a fase de pico será *Ψ + 2*x-Wz*.

Fazendo:

$$P_1 . P_2^* = A_1 A_2 e^{j(\psi - (\psi + 2Wx - Wz))} \quad (28)$$

Teremos:

$$W_x = \angle(P_1 . P_2^* . e^{j2wx}) \quad (29)$$

Se k_{MAX} é o índice do pico em magnitude logarítmica FFT representado pelo índice de sinais na distância $[-\frac{N}{2}, \frac{N}{2} - 1]$, então w_x será:

$$W_x = \frac{2\pi}{N} K_{max} \quad (30)$$

1.7.1 Comunicação com a porta USB

Após as operações anteriores, o EMV 1443 organiza as informações em

pacotes; estes têm um cabeçalho que descreve o pacote e o conteúdo das operações anteriores para um frame de medição.

A composição do cabeçalho tem um total de 320 bits, sendo os primeiros 64 bits chamados de *magic word*; esta é uma informação de sincronização. No momento da leitura dos dados, procuramos esses 16 bits, e ao achá-los, sabemos que é o começo do pacote. Após, vêm informações acerca do próprio pacote e do sensor utilizado, todos com tamanho de 32 bits. Primeiro vem a informação da versão do sensor utilizada para a geração do pacote; em seguida, o tamanho do pacote total, junto do header. A plataforma e o frame referentes àquele pacote vêm em seguida. A sexta informação é o tempo em ciclos da CPU em que o pacote foi criado. A sétima informação é o número de objetos detectados. Depois vem o número de conjunto de informações diferentes, que será detalhado mais à frente. E por último, a configuração avançada de frames – se existir; do contrário é uma sequência de 0) [6].

Além disso, o pacote finaliza com uma operação de *padding* para que o tamanho total dele seja múltiplo de 32 bits.

Após o cabeçalho, vêm os conjuntos de informações, chamados de TLVs. Eles são divididos em seis tipos de conjunto de dados. Para a aplicação deste trabalho, foram usados somente o primeiro e o terceiro tipos de conjuntos, visto que os demais servem para análises mais profundas sobre o comportamento das ondas milimétricas. Todo TLV é composto de um cabeçalho com a informação de tamanho e tipo, e em seguida vêm as informações. O primeiro tipo de TLV é a lista de objetos detectados. Em seu cabeçalho vêm as informações padrões de tamanho e tipo de TLV, e em seguia, cada objeto detectado com informações de posição, velocidade e as indexações. A composição se divide da seguinte maneira (figura 2.22):

- Indexação de distância (16 bits)
- Indexação doppler (16 bits)
- Valor de pico (16 bits)
- Coordenada x (16 bits)
- Coordenada z (16 bits)
- Coordenada y (16 bits)

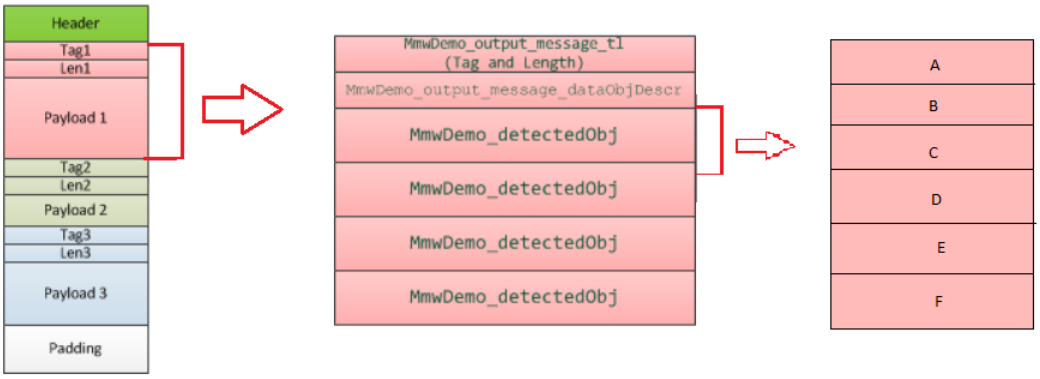


Figura 2.22 – organização dos pacotes (TLVs). Adaptado de [6]

No caso das coordenadas, os valores estão em metros no formato Q9. O segundo tipo de TLV são as informações sobre a magnitude recebida de pontos que não têm velocidade.

Na sequência temos as informações sobre a magnitude de ruídos para cada objeto na velocidade máxima, mas para cenas estacionárias, essa informação vai ser o ruído mínimo no ambiente. O quarto TLV é o mapa de calor dos ângulos de Azimuth; o quinto é o mapa de calor de distância de objetos com velocidade, e, por último, a informação acerca do processamento, como quanto de memória foi gasto e o tempo de execução.

1.8 Técnicas de tratamento de nuvem de pontos

A princípio, uma nuvem de pontos com as informações de posição em um sistema cartesiano pode ser inspecionada e filtrada diretamente, mas são comumente convertidas em um conjunto de polígonos para análise de superfície. No caso do sensor utilizado, teremos informação não só da superfície do objeto, como alguns pontos dentro dele. Isso faz com que a sequência de pontos gerados pelas técnicas a

seguir mostre a espessura dos objetos, dependendo de quão espesso é o objeto.

A técnica utilizada pode mudar, dependendo de como será utilizado o sensor, se móvel ou parado, com movimento circular na vertical e ou na horizontal etc.

Existem várias técnicas para o tratamento de nuvem de pontos para gerar superfície, porém, as duas técnicas que comentaremos são as mais comuns na literatura.

1.8.1 Triangulação Delaunay

Na matemática e geometria computacional, uma triangulação Delaunay, para um dado conjunto de pontos discretos em uma posição geral, é uma triangulação tal que nenhum ponto em p está dentro do círculo circunflexo de qualquer triângulo feito a partir de três pontos. Essa triangulação maximiza o menor ângulo de todos os ângulos de todos os triângulos, para tentar evitar ângulos agudos que podem conter propriedades não desejadas em interpolações.

Para evitar que a triangulação não seja única, uma das condições é que o interior dos círculos não pode conter pontos. Essa triangulação também pode ser aplicada para mais que duas dimensões [12].

Umas das propriedades dessa operação de triangulação é o chamado *flipping*. Essa propriedade é importante, pois, quando dois triângulos que compartilham a mesma borda podem ter a soma dos ângulos opostos iguais ou menores que 180 graus, os triângulos cumprem a condição de Delaunay. Caso contrário, a troca da borda para uma borda que ligue os dois pontos que geram esses ângulos vai produzir triângulos que atendem essa condição. Isso permite que seja possível modificar a forma da superfície e diminuir ângulos agudos.

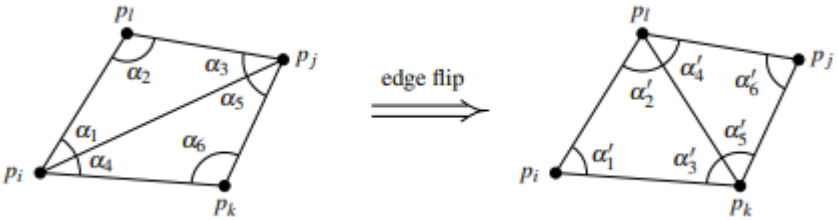


Figura 2.23 – Operação de *flipping* [12]

1.8.2 Alpha shape

Uma alpha shape, ou α-shape, é uma família de curvas simples lineares por partes no plano euclidiano associado com o formato de um grupo de pontos finitos. A-shape é generalização do conceito de encapsulamento convexo, ou seja, todo encapsulamento convexo é uma α-shape, mas nem toda α-shape é um encapsulamento. Encapsulamento convexo de uma forma é o menor convexo que contém essa forma [9].

Fazendo uma analogia, se tivermos uma bola de sorvete com gotas de chocolate, e utilizando uma colher for retirado o soverte da borda sem que nenhuma gota de chocolate seja retirada junto, o resultado será uma α-shape. Considerando que a colher seja circular, o raio de retirada será o fator α [9].

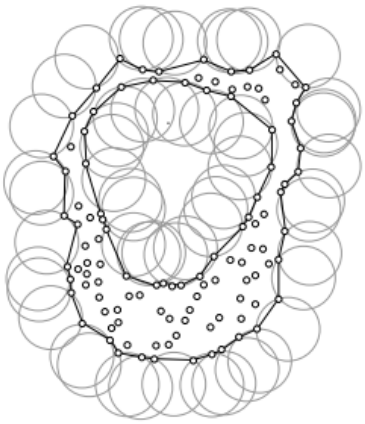


Figura 2.24- Uma α-shape com um valor de α [9]

CAPÍTULO 3 – DESENVOLVIMENTO

1.9 Seleção do Sensor

A primeira etapa do projeto é a escolha do sensor. Foi escolhida a linha de sensores radares da empresa Texas Instruments (TI), pois é uma empresa altamente confiável com uma linha de sensores especificamente desenvolvida para uso automotivo e industrial. Outras empresas oferecem soluções com sensores radares, mas não o sensor para aplicações de usuários, além de que a maioria dessas empresas utiliza alguma tecnologia disponibilizada pela TI, outro motivo pela escolha dessa empresa.

A linha de sensores é chamada de industrial *mmWave sensors* (sensores industriais de onda milimétrica). Esses sensores têm como principal característica o alto desempenho em locais fechados, com alta resolução tanto para distâncias quanto para velocidades.

A empresa disponibiliza quatro tipos de chips de processamento, cada um com um hardware contendo os componentes necessários e antenas do tipo emissor (*TX*) e receptor (*RX*), para aplicações industriais. Essas antenas podem ser arranjadas de várias maneiras, tendo números de emissores e receptores diferentes em um mesmo sensor, além de configurações de duração da onda *chirp* e modulação das ondas. Na Tabela 3.1, pode-se verificar as principais características de cada antena. A antena IWR 1843 é a única antena deste conjunto que não usa a tecnologia de sinais *chirp*, e por isso foi descartada para uso neste projeto.

Tabela 3.1 - Comparativo do cone de detecção angular das antenas [2]

Tipo de antena	Azimuth CDV (deg)	Elevação do CDV (deg)	Resolução angular de Azimuth (deg)	Resolução angular da elevação (deg)	Distância máxima para pessoas(M)
IWR1443BOOST (77GHz)	+/- 60	+/- 15	15	60	70
IWR1642BOOST (77GHz)	+/- 60	+/- 15	15	N/A	90
IWR1843BOOST (77GHz)	+/- 60	+/- 15	15	58	80
IWR6843ISK (60GHz)	+/- 60	+/- 15	15	58	75
IWR6843ODS (60GHz)	+/- 60	+/- 60	29	29	60
IWR6843AoP (60GHz)	+/- 60	+/- 60	29	29	60

Para o projeto de mapeamento de espaços fechados é necessário que o sensor tenha um alto espaço de visão (FOV) e uma alta acurácia e resolução, já que é importante que o sensor devolva a posição exata dos objetos dentro de uma sala. Como o sensor irá trabalhar em um local onde há pouco movimento, serão descartadas as informações de velocidade e a acurácia da mesma. Assim, a Tabela 3.2 contém as informações dos sensores, a diferença entre cada um, o número e antenas de recepção e emissão que cada sensor possui, e a inclinação da curva que mostra a alteração da frequência no sinal *chirp*.

Os sensores adquiridos via Texas Instrument (TI) vêm com o número de antenas fixas, somente mudando a configuração via software quais antenas ativar. Este sensor é chamado de módulo de avaliação (EVM).

A partir da Tabela 3.2, foram escolhidos os sensores IWR6843 e IWR1443, pois os dois têm alta resolução e acurácia para distâncias razoáveis, semelhante ao tamanho de salas. Nota-se principalmente a resolução de 0,04 metro com acurácia de 0,73 centímetro para uma distância de até 30 metros do sensor IWR1443, utilizando uma antena para emissão e recepção com uma taxa de 10 MHz do sinal *chirp* e também o sensor IWR6843 com uma resolução de 0,125 metro e acurácia de aproximadamente 0 centímetro, ou seja, dentro da resolução de distância do sensor a medida será virtualmente exata, para uma distância de 12,79 metros, utilizando três antenas de emissão, quatro de recepção e 10 Hz de variação do sinal *chirp*.

Tabela 3.2- Comparativo entre os sensores e suas configurações [2]

Sensor	Frequência (GHz)	distância máxima (m)	resolução da distância (m)	acurácia da distância (cm)
IWR6843	60.25 - 64	12.79	0.125	0
IWR6843	60.5-64	6.3	0.055	5.5
IWR6843	60.5-64	6.4	0.125	12.5
IWR6843	60.5-64	56	0.49	49
IWR1642	77-81	6.4	0.125	2.3
IWR1642	76-77	30	0.18	3.2
IWR1642	77-81	6	0.05	5
IWR1642	77-81	5.6	0.05	5

IWR1642	77-81	84	0.366	6.68
IWR1642	77-81	200	0.4	7.3
IWR1642	77-81	14	0.12	12
IWR1642	76-77	56.25	0.4151	20.5
IWR1642	76-77	70	0.25	25
IWR1642	76-77	120	1	100
IWR1443	77-81	30	0.04	0.73
IWR1443	77-81	6.4	0.125	2.3
IWR1443	77-81	9.62	0.047	4.7
IWR1443	77-81	100	0.18	10.8

Além dos sensores, também foi adquirida a placa MMWAVEICBOOST, usada para desenvolvimento de software avançado, melhora do processamento do sensor e acompanhamento do código base do sensor em tempo real, ou seja, essa placa permite configurar o sensor, número de antenas ativas e as curvas de sinal *chirp*, para melhor servir ao propósito do projeto.

A seguir, as imagens dos sensores e da placa adquiridos.

Figura 3.1- IWR1443BOOST [2]

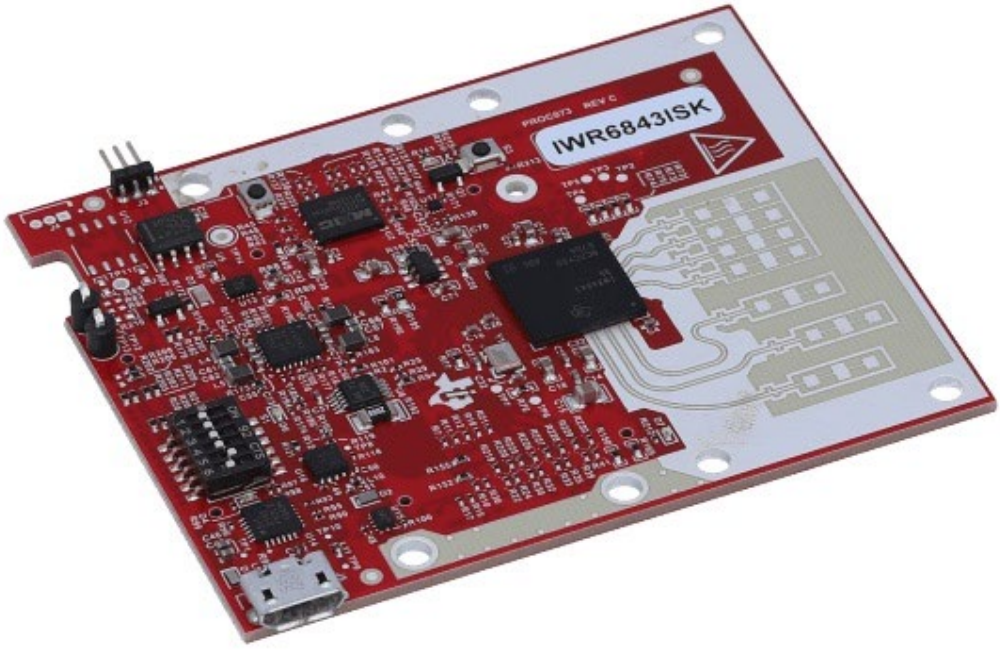


Figura 3.2 - IWR6843ISK [2]

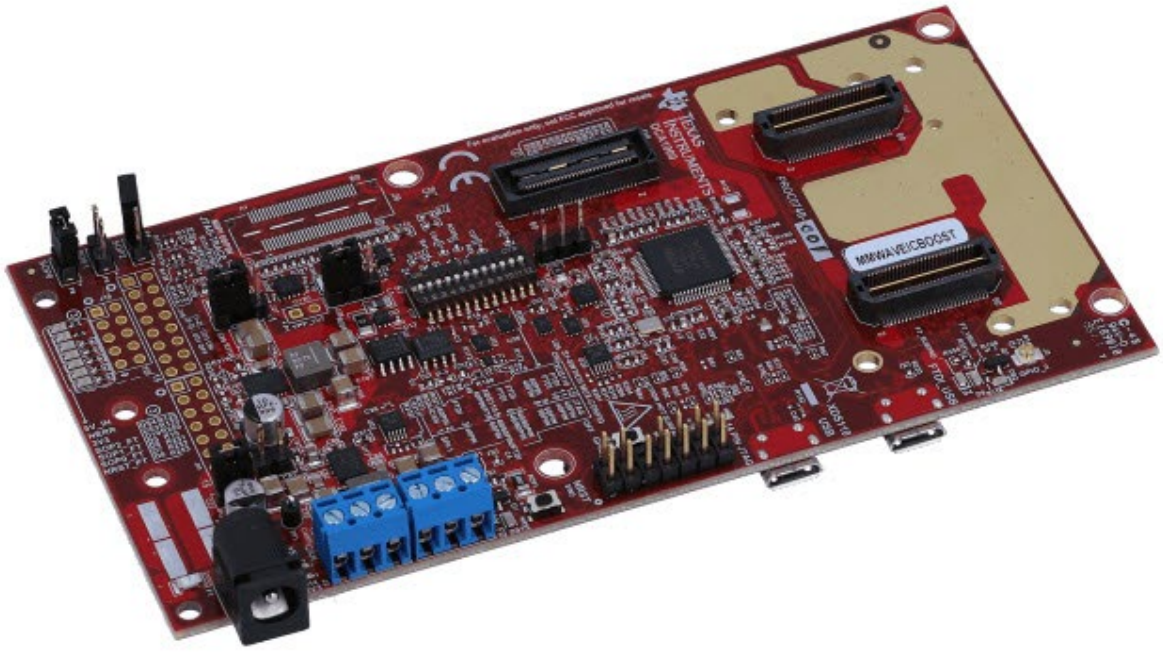


Figura 3.3- MMWAVEICBOOST [2]

1.10 Fixação do sensor em suportes fixo e móvel

O sensor foi montado em uma placa de papelão rígida para criar um suporte universal, a qual poderia ser presa em uma superfície sem que o sensor tocasse na mesma ou ficasse inclinado. Esse suporte então foi colocado em duas posições, uma móvel e uma fixa.

A posição fixa foi utilizada para a calibragem do sensor utilizando os softwares disponibilizados pela empresa e, ao mesmo tempo, para a construção dos programas em Matlab. A construção da cena e o posicionamento do sensor, neste caso, seguiram as orientações da calibragem sugerida pelo fabricante.

A posição móvel é um tripé com gira de 360 graus e um suporte que permite prender a base. Isso dá mobilidade ao sensor, permitindo giros para a criação de uma nuvem de pontos do ambiente. A montagem do sensor no suporte e no tripé pode ser vista nas Figuras 3.4, 3.5 e 3.6.

Figura 3.4 – fixação do sensor no tripé

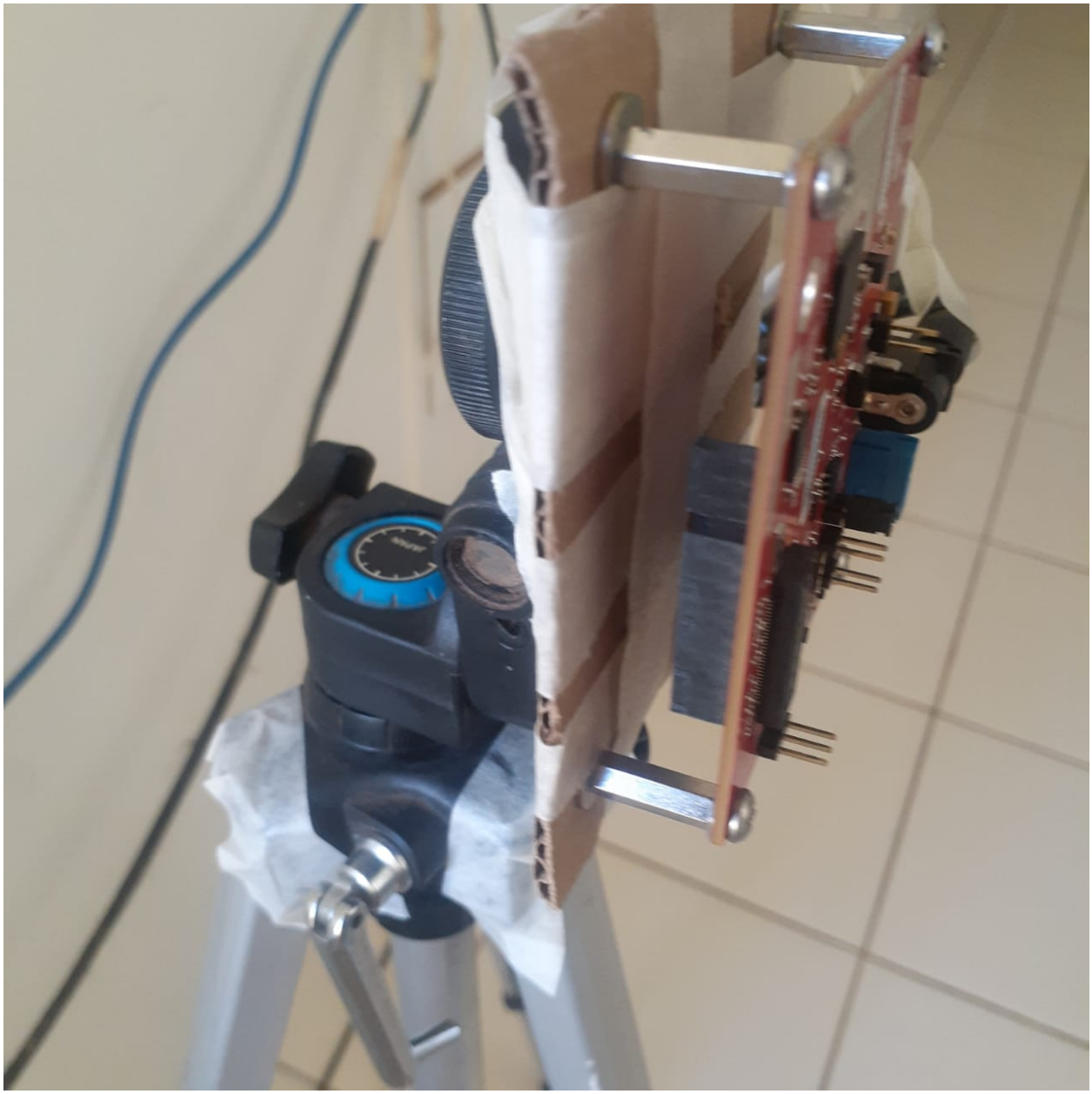


Figura 3.5 – Fixação do sensor no tripé

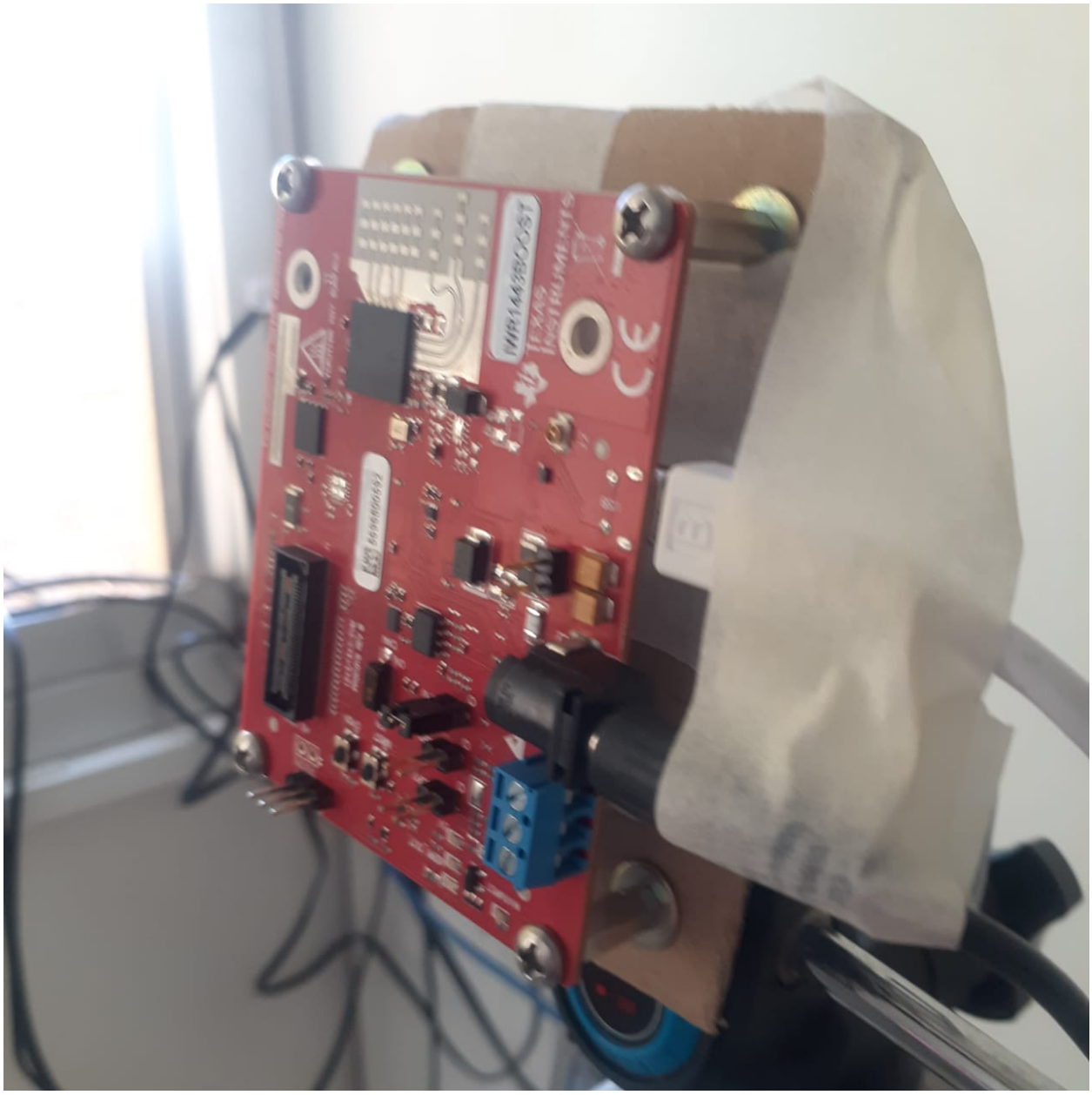


Figura 3.6 – Fixação do sensor visão frontal

1.11 Teste e calibração

Após a aquisição dos sensores, foi feito o download dos softwares disponibilizados pela TI, e a partir dos tutoriais disponibilizados, foram realizadas as devidas calibrações. A empresa recomenda utilizar o exemplo “OUT OFF THE BOX DEMO”, que é um guia rápido passo a passo de como montar, testar e calibrar o sensor.

O guia [3] para o sensor IRW1443 e o guia [4] para o sensor IRW6843 são bem parecidos, mudando somente a forma de configurar cada placa.

Foi utilizado o guia para o sensor IRW1443 [3], calibrados e testados os dois sensores e a placa extra. Nesta seção, serão mostrados um dos testes feitos e um dos exemplos disponibilizados pelo próprio software.

Este exemplo é o recomendado pela TI para calibragem e testagem do sensor, chamado de “OUT OFF THE BOX DEMO”. Nele é mostrado como usar os principais programas necessários para o funcionamento do sensor. A Tabela 3.3 mostra os softwares utilizados neste demo, e que serão utilizados para a próxima parte do projeto.

Tabela 3.3 - Links para os softwares usados [3]

Software	Link para download
TI mmWave SDK	http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/latest/index_FDS.html
mmWave Demo Visualizer	https://dev.ti.com/gallery/view/mmWave/mmWave_Demo_Visualizer/
mmWave Industrial Toolbox	http://dev.ti.com/tirex/explore/content/mmWave_industrial_toolbox_4_5_1/docs/readme.html
uniflash	https://www.ti.com/tool/UNIFLASH
CCS	http://software-dl.ti.com/ccs/esd/documents/ccs_downloads.html#Code_Composer_Studio_Version_8_Downloads

Após a instalação dos softwares, deve ser feito o download do projeto a ser carregado na placa IRW1443. O projeto está disponível no CCS, que é a plataforma de exploração de projetos e laboratórios elaborados pela TI, e pode ser encontrado como na Figura 3.7.

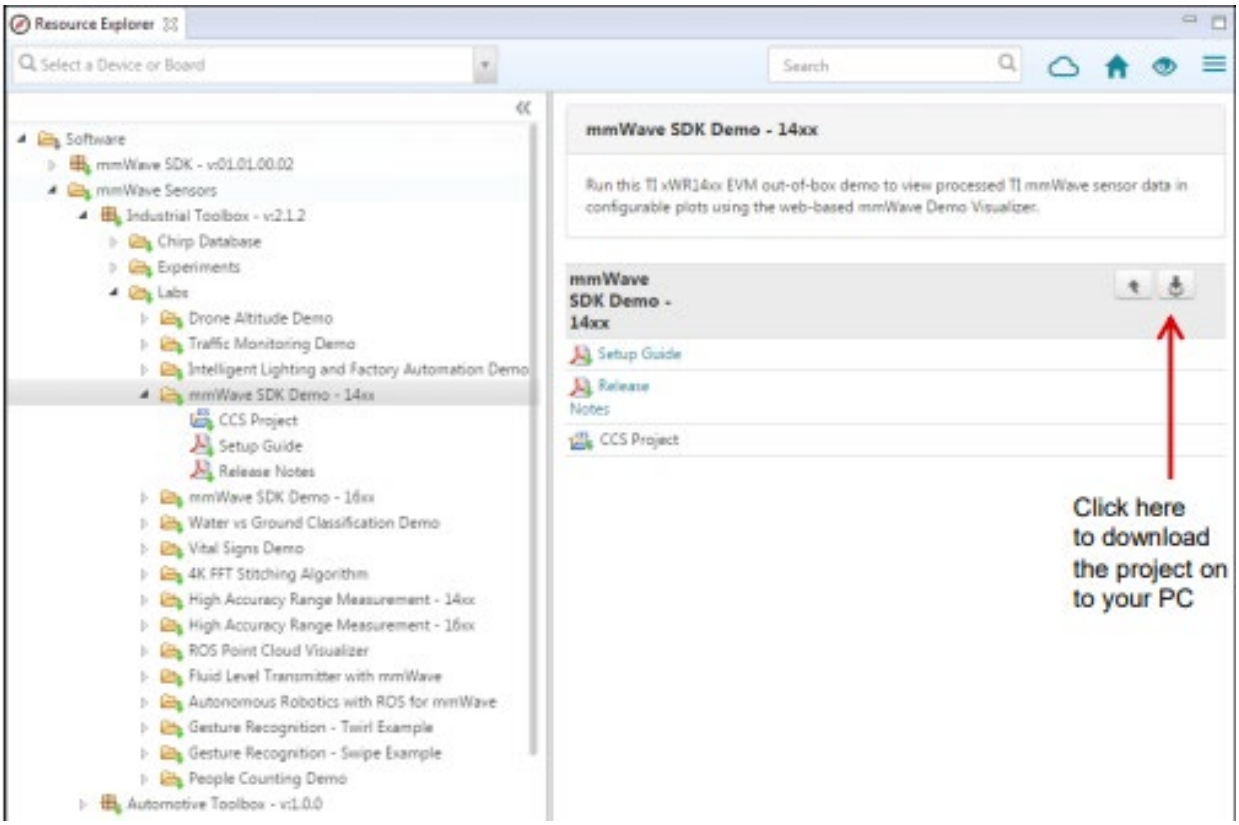


Figura 3.7 – Localização do download do projeto

Após a aquisição do projeto, deve ser feita a compilação do mesmo, que pode ser feita pelo CSS e irá gerar dois arquivos, como na Figura 3.8. Os dois arquivos serão salvos no diretório especificado e terão o nome semelhante a:

- xwr14xx_mmw_demo_mss.xer4f (1)
- xwr14xx_mmw_demo.bin (2)

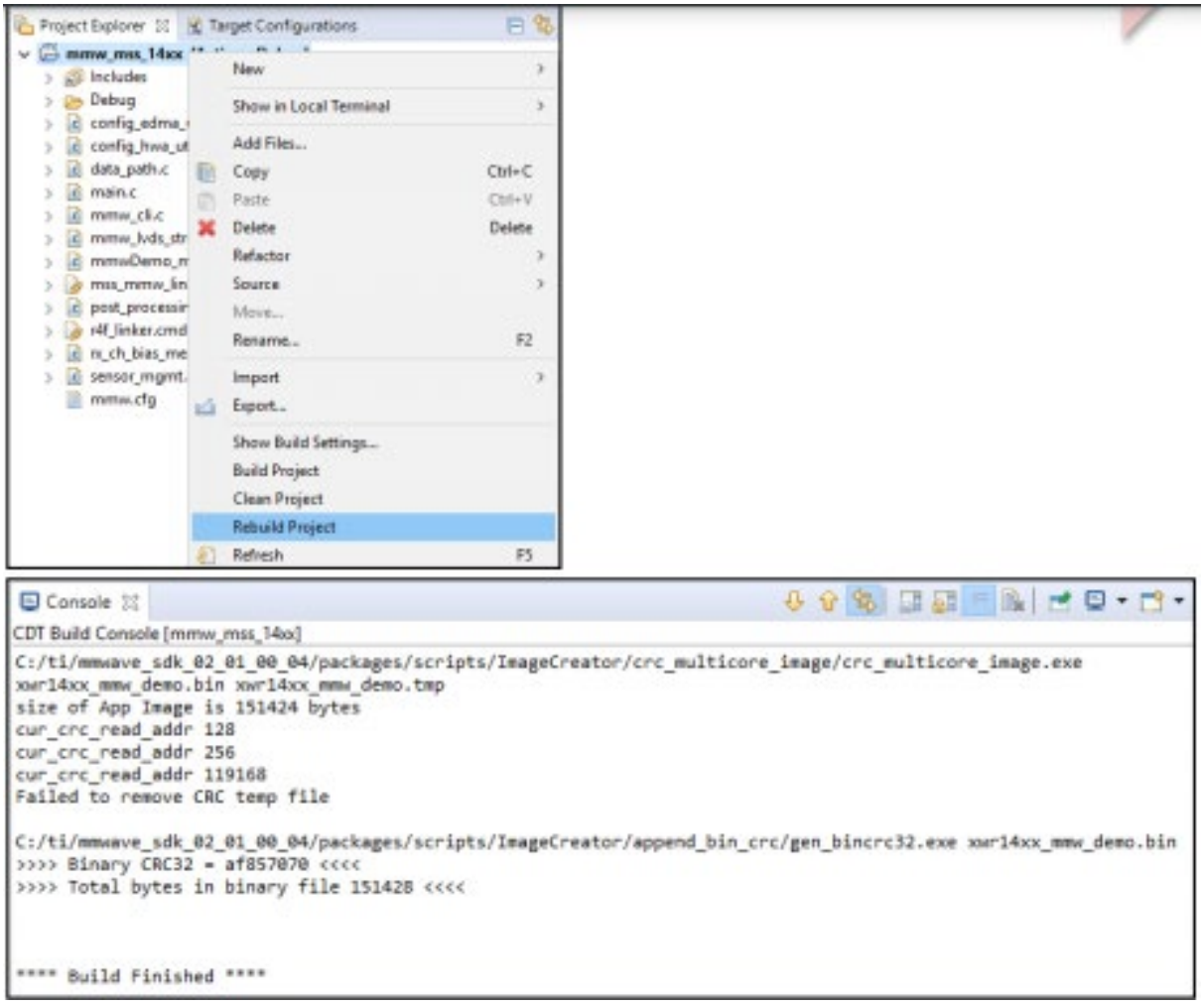


Figura 3.8 - Compilação do projeto

Com a compilação finalizada, é conectado o sensor via cabo USB ao computador utilizado e será feito o download no sensor com as configurações e caminho de dados dele. Existem dois métodos utilizados para isto: o primeiro é o modo de desenvolvimento, que grava uma imagem binária e, neste modo, o sensor roda automaticamente. No segundo método, chamado de modo *Debug*, é feito o download na placa de um executável que é rodado via o programa SDK. O objetivo do trabalho é fazer a caracterização do sensor, logo, o segundo método não é interessante para este trabalho, mas também está explicado no guia e será então explicado o primeiro método.

A placa deve ser colocada primeiro em *flashing mode*, conforme Figura 3.9; em seguida é aberto o programa *UniFlash*. Esse programa vai fazer a gravação da

imagem binária do caminho de dados programado no chip 1443. Selecionamos então a porta correta para a transmissão do arquivo binário, e colocamos o caminho do arquivo conforme Figuras 3.10 e 3.11. Em seguida, gravamos o arquivo binário no sensor, desligamos o mesmo, tiramos o sensor de *flashing mode* abrindo o curto em SOP2 e voltamos a ligar o sensor. O sensor está preparado para receber configurações de antena e entrar em funcionamento.

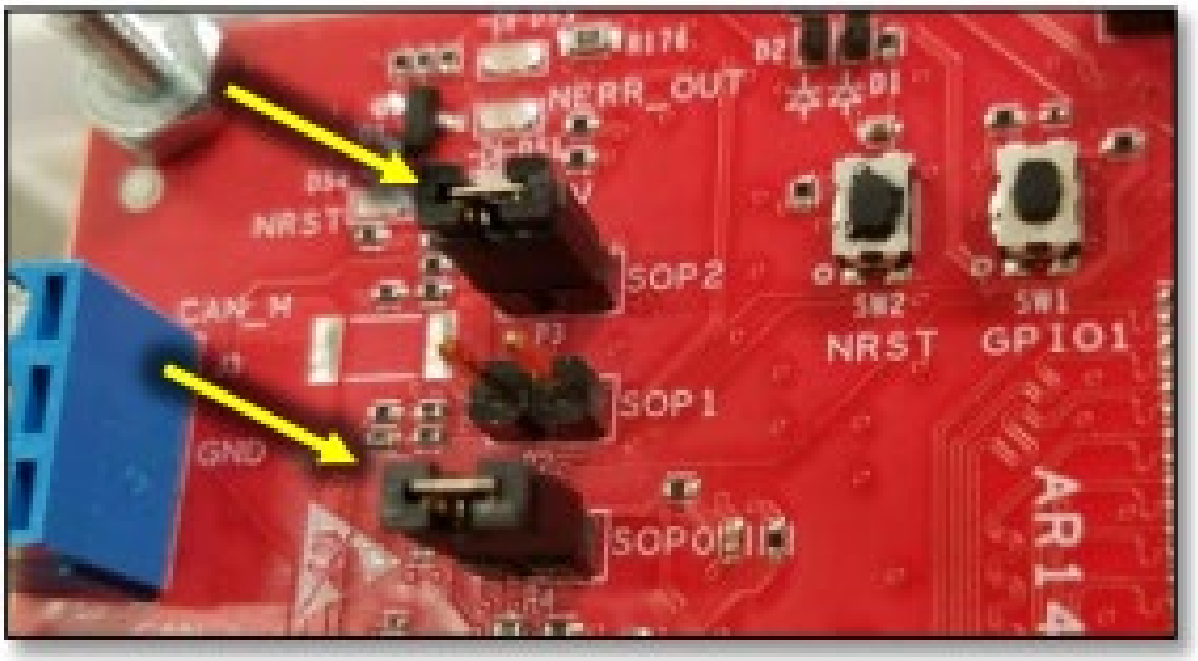


Figura 3.9 - Placa do sensor em flashing mode

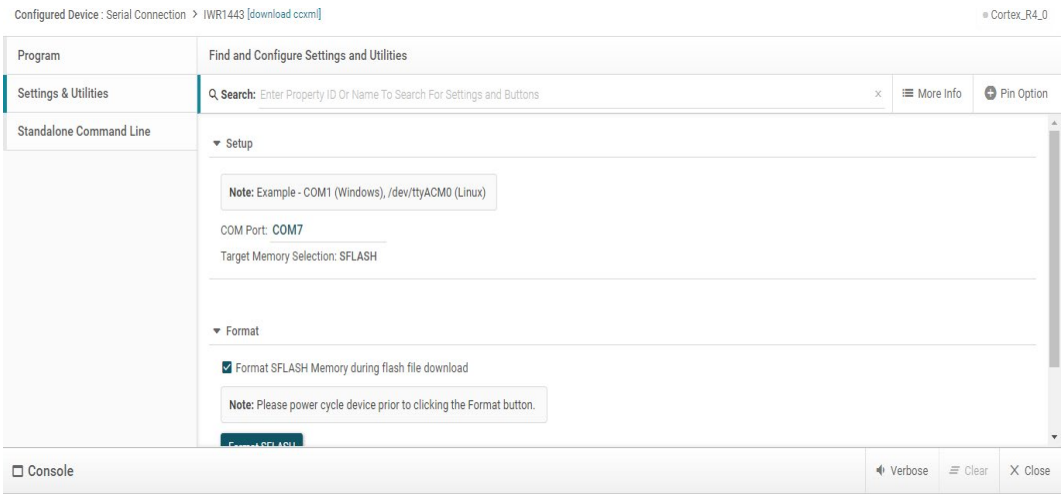


Figura 3.10 - Selecionando a porta

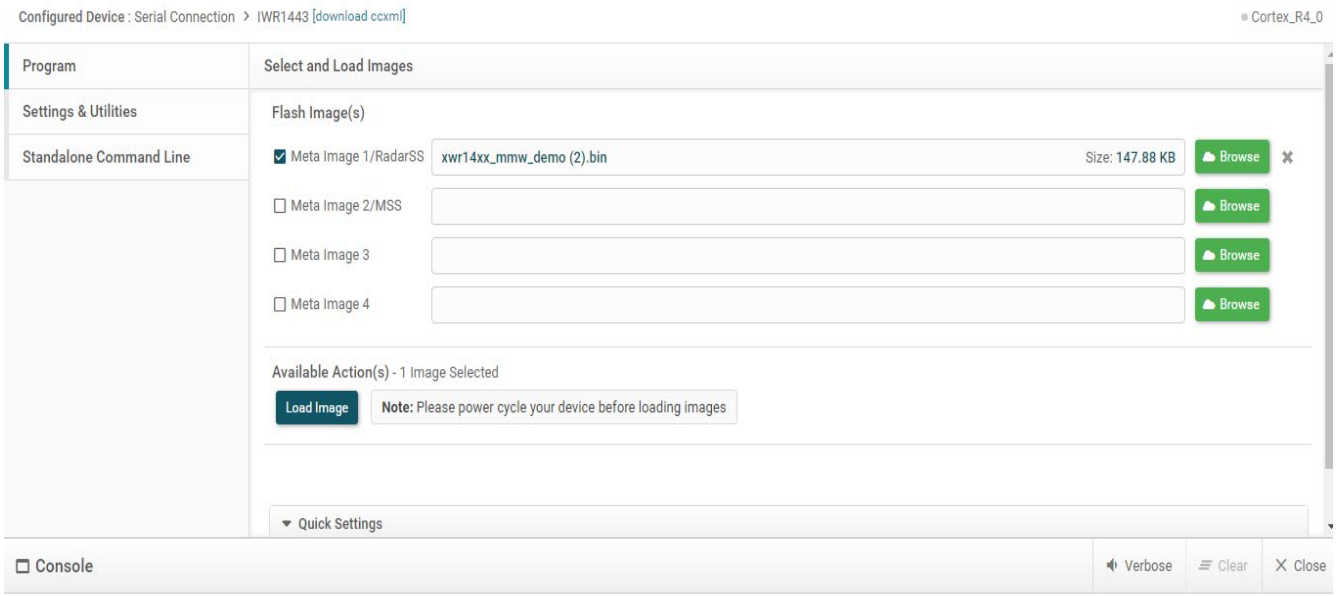


Figura 3.11 – Selecionando arquivo binário

Para visualizar o sensor em funcionamento, usaremos o Programa *mmWave Demo Visualizer*. Esse programa pode ser acessado em nuvem pelo navegador seguindo o link disponível na tabela de programas. É feita então a configuração e, a partir dela, pode ser utilizada a aba de gráficos (*plots*) para verificar a resposta do sensor. Essa configuração inclui quantidade de antenas com a resolução do ângulo de medição, qual vai ser a prioridade dos pontos ou objetos gerados e a banda de frequência. Também podem ser selecionados os parâmetros referentes à cena. Neste caso, para verificar a calibração do sensor, seguiremos as configurações da Figura 3.12. Essas configurações permitem verificar o erro com os pontos gerados em 3D, a quantidade e, no caso deste teste, verificar como o sensor reponde de frente a uma parede de tijolos em duas distâncias diferentes.

Essa configuração foi escolhida em razão da cena que será descrita mais adiante. Por causa das propriedades do sensor, foi definida uma distância máxima de 2,42 metros, visto que leituras de outros objetos que estão fora da cena poderiam atrapalhar na validação da precisão do sensor. A velocidade máxima é a menor possível, pois queremos somente pontos estáticos, e a resolução de velocidade e distancia são consequências dessas escolhas. Como não vamos detectar objetos em movimento, a taxa de quadros não é importante.

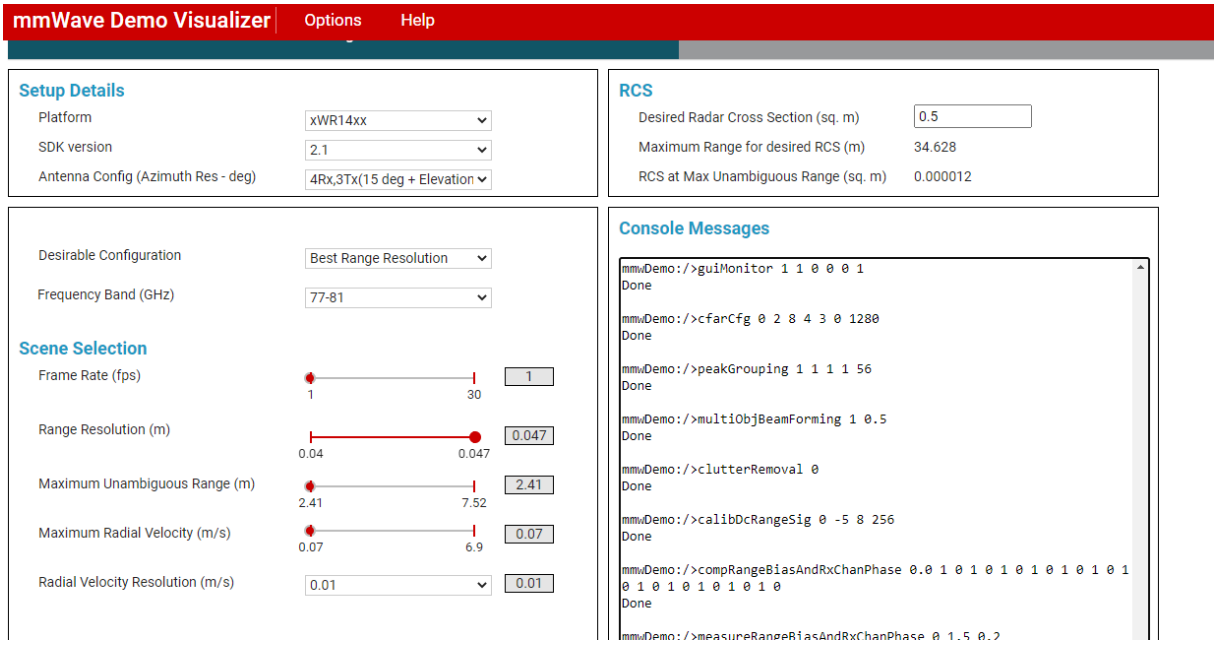


Figura 3.12 – Configurações selecionadas

Enviamos a configuração ao sensor e verificamos os gráficos gerados. O visualizador demonstrativo somente converte os pontos e as informações enviadas pelo sensor; ele não disponibiliza essas informações tratadas, somente no formato que chega na porta USB. Esse visualizador serve somente para calibração e teste do sensor. Essa calibração é feita pela aba de configuração. É possível ainda alterar aspectos relacionados ao funcionamento do sensor modificando os arquivos que gravam a imagem no sensor.

Nas Figuras 3.13 e 3.14 consta o teste executado para a calibração do sensor. Foi utilizada uma configuração de geração de pontos 3D. Logo, o primeiro gráfico da esquerda para a direita na Figura 3.10 mostra os pontos que o sensor captou; na sequência temos o gráfico que mostra a distância de pontos sem nenhuma velocidade. Podem ser adicionados outros gráficos, mas a princípio esses são os dois gráficos que interessam para este trabalho. Desabilitamos também as opções de agrupamento de distância e velocidade, visto que queremos ver a nuvem de pontos e como ela se comporta.

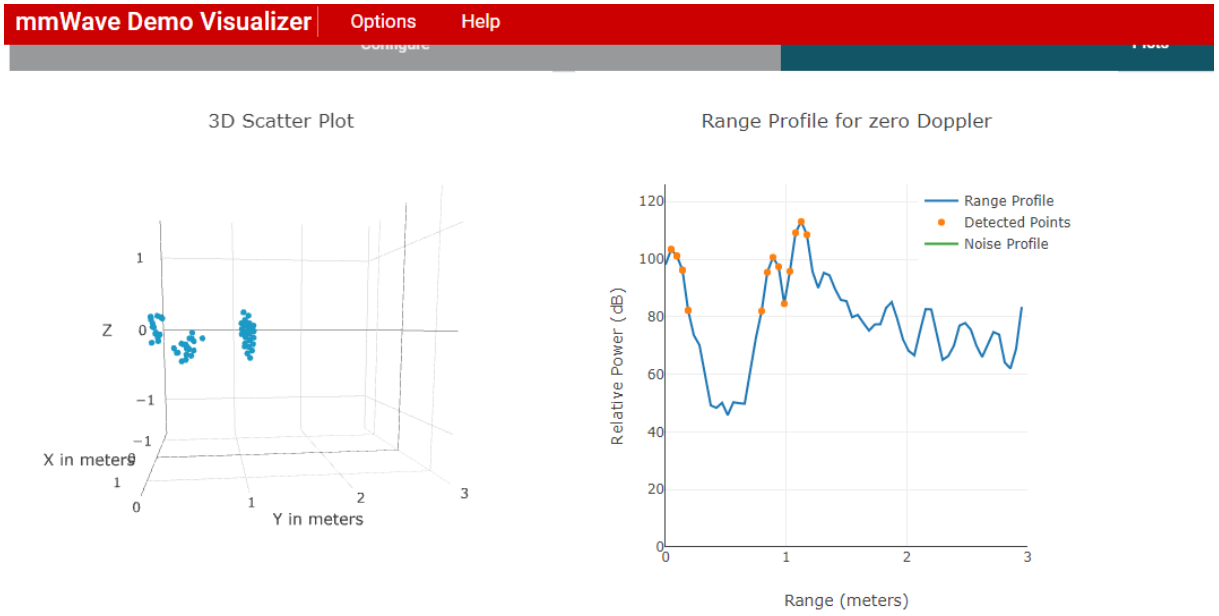


Figura 3.13 – Distância da parede de 105,5 centímetros e 113,5 centímetros acima do chão

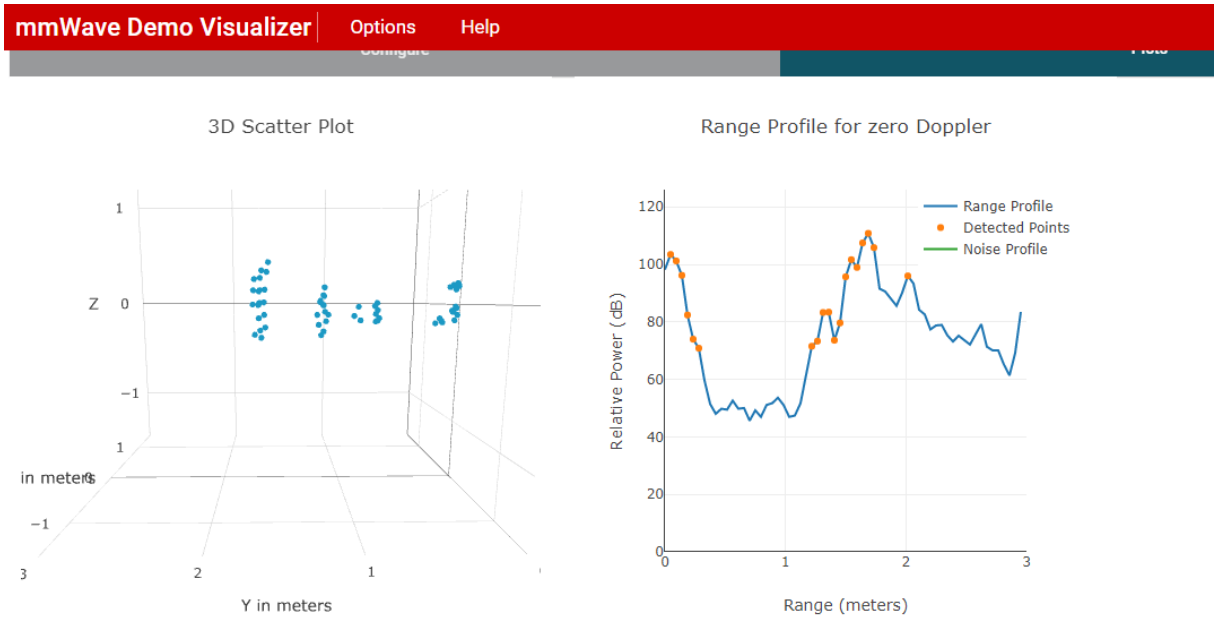


Figura 3.14 – Distância da parede de 162 centímetros e 113,5 centímetros do chão

É possível navegar nos gráficos e verificar as coordenadas dos pontos mostrados. No caso da imagem 18, o primeiro ponto sinalizando o encontro da parede diretamente à frente do sensor (eixo y) está a 103,98 centímetros de distância, ou seja, dentro da resolução escolhida de 4,7 centímetros. No caso da imagem 19, o primeiro ponto está a 163 centímetros de distância. É importante notar que alguns pontos na lateral esquerda do sensor foram capturados, isto porque a janela, que será

comentada mais à frente no cenário, estava a 70 centímetros do sensor, ou seja, dentro do cone de visão do sensor.

Com esse teste completado, é possível criar o programa em *matlab* que irá ler as informações do sensor, construir a cena e mostrar o comportamento do sensor quando é gerada uma nuvem de pontos rotacionando o sensor.

1.12 Construção dos programas

Para a leitura das informações enviadas pelo sensor, utilizamos as informações expostas na seção 2.4.1 deste trabalho. Assim, foi construído um *parser* que recebe os pacotes das informações em arquivo binário, lê o arquivo, separa as informações que interessa para a construção da nuvem de pontos e faz as conversões necessárias para que seja mostrado em uma única imagem.

O funcionamento do *parser* é simples. Utilizando as funções de leitura do Matlab, lemos a quantidade de *bits* necessária para verificar o começo do pacote, que pode ser visto na Figura 3.15, que chamamos de *magicWord*. Após verificar o começo do pacote, fazemos as leituras nos tamanhos descritos na seção 2.4.1.

```
while (~feof(file))
tamanhodestepacote =0;
header = fread (file , 4 , 'uint16')% magic word
if (length(header) < 4)
 'erro no header, finalizando operação (header vazio)'
 continue;
end
if (header(1) ~= 258 || header(2)~=772 || header(3)~=1286 || header(4)~=1800)
    'erro no header, tentando novamente(header errado)'
    fseek(file, 6 , 'cof');
    continue;
end
header = dec2hex(header);
```

Figura 3.15 – Leitura do header

No pacote em geral, o primeiro grupo de informações são os pontos lidos pelo sensor. O TLV com as informações dos pontos é lido conforme a Figura 3.16, e por

fim guardado para futuro processamento. Como mencionado na secção de dados enviados ao usuário, também lemos o TLV de ruído, para podermos descartar pontos em movimento que podem ser erros de leitura.

```
if(identificadordotvl==1) % detected objects
    medidas=medidas+1;
    descritor=fread(file,2,'uint16');
    nuemrodeobjetoslocais=descritor(1); %prieiro argumento sendo o numero de objetos neste tvl
    xyzq=descritor(2); %segundo argumento sendo o formato das cordenadas
    i=0;
    range_index=[];
    doppler_index=[];
    peak_value=[];
    x_coordinate=[];
    y_coordinate=[];
    z_coordinate=[];
    while(i~=nuemrodeobjetoslocais)
        i=i+1;
        range_index=[range_index,fread(file,1,'uint16')];
        doppler_index=[doppler_index,fread(file,1,'uint16')];
        peak_value=[peak_value,fread(file,1,'uint16')];
        % x,y,z podem ser retirados dividindo por 2^xyzq
        x_coordinate=[x_coordinate,fread(file,1,'int16')/2^xyzq];
        y_coordinate=[y_coordinate,fread(file,1,'int16')/2^xyzq];
        z_coordinate=[z_coordinate,fread(file,1,'int16')/2^xyzq];
    end
```

Figura 3.16 – Leitura dos pontos

Por fim, é feita a transformação dos pontos para a cena, pois os pontos são sempre em referência ao sensor. O processo se repete para cada angulação do sensor. Isto é feito realizando uma transformação simples de rotação de coordenadas para a que será usada como principal, ou seja, a posição do centro do sensor com o eixo y apontado para a janela. É feito também um deslocamento de 7,3 centímetros, pelo fato de o sensor estar a essa distância do centro do tripé utilizado para o movimento angular. Podemos ver essa operação no código na Figura 3.17.

```
while (j<=sizedepontos(2))
    b = pontosx(j);
    c = pontosy(j);
    a=(b^2)+(c^2);
    if b~=0
    alfa = atan(c/b);
    else
        alfa = 1.5708;
    end
    d = sqrt(a);
    if b<0
        d=-d;
    end
    deslocamentosensorx=   0.073*cos(rotacao+90);
    deslocamentosensory=    abs(0.073*sin(rotacao+90));
    pontosx(j) =deslocamentosensorx + d*cos(rotacao+alfa);
    pontosy(j) =deslocamentosensory + abs(d*sin(rotacao+alfa));
    j= j+1;
end
plot3 (pontosx,pontosy,pontosz,'.','color','red')
hold on;
pause();
contador= contador+1;
x = [x;pontosx(:)] ;
y = [y;pontosy(:)] ;
z = [z;pontosz(:)] ;
```

Figura 3.17 – Deslocamento dos pontos para o plano cartesiano centralizado no sensor

Ao fim do processo, é criado um arquivo *.XYZ* para que possa ser exportado para outros programas.

Após as primeiras calibrações, nota-se uma quantidade de pontos muito próxima do sensor, perto ou abaixo do limite de diferenciação de objeto. Esses pontos foram considerados ruídos, porque acontece uma atenuação de ruídos existentes no ambiente em posições aleatórias. Nem sempre a atenuação acaba com o ruído por completo, fazendo com que o sensor interprete esse ruído como pontos próximos a ele. Por este motivo, foram removidos esses pontos e a análise de pontos foi feita visualmente.

Além disso, o programa cria, com as medidas reais, as paredes, janelas e obstáculos, caso haja algum, para facilitar na análise da precisão dos pontos.

1.13 Construção da cena

Para executar as medições como simulação para ambientes internos, a cena foi montada seguindo a Figura 3.18. Nela, foram colocados quatro tipos de matérias diferentes, comuns em ambientes fechados, sendo a cor azul para parede de tijolos com 19,5 centímetros de espessura; a cor amarela para uma janela de vidro com

periferia de metal, sendo o vidro com 3 centímetros de espessura e o metal com 7,5 centímetros de espessura; e por último uma parede de madeira, com aproximadamente 6 centímetros de espessura, representada em preto. A janela está situada a 63 centímetros do chão. O ambiente original pode ser visto na Figura 3.19.

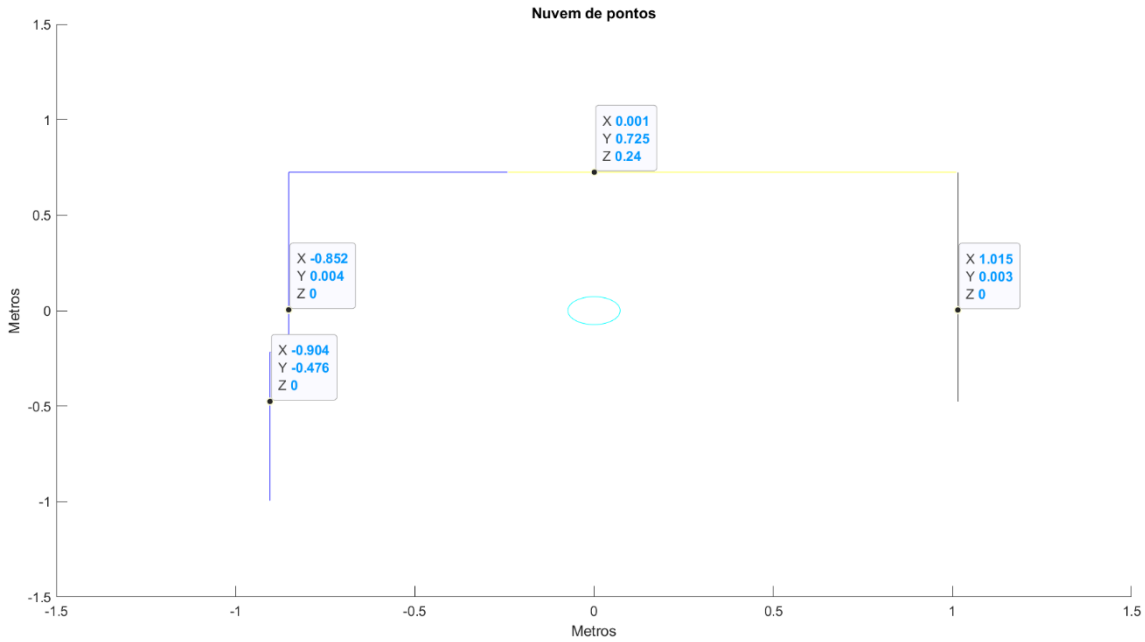


Figura 3.18 – Visão de cima da cena (coordenada X por Y)

O sensor foi colocado na posição em azul claro, sendo a distância para os limites mostrada na Figura 3.18. As medidas foram efetuadas em duas alturas: a primeira de 54,3 centímetros e a segunda de 110 centímetros.

Figura 3.19 – Foto da cena

CAPÍTULO 4 – EXPERIMENTOS E DISCUSSÕES

1.14 Experimentos

Os experimentos foram divididos em duas etapas. A primeira etapa é a medição da cena somente com pontos em duas dimensões; a segunda, com medições em três dimensões. Foram efetuadas medidas em duas alturas diferentes e em 180 graus, com um passo de 30 graus entre cada posição do sensor. Ao fim foram montadas quatro nuvens de pontos, duas medidas na altura de 53,5 centímetros e as outras duas na altura de 110 centímetros. A resolução de rotação no eixo x-y é de 30 graus. Essa resolução foi escolhida, pois o tripé no qual o sensor está conectado somente permite fazer variações de 30 graus com precisão.

Para a aquisição dos dados foi utilizado o programa *Demo Vizualizer*, que permite a extração dos dados enviados pelo sensor à porta USB, criando um arquivo binário com os dados. Não há diferença entre a leitura feita diretamente pela porta ou pelo arquivo gerado pelo demo, mas por causa de limitações computacionais, foi utilizada a geração do arquivo do programa mencionado e, assim, realizar a leitura das medidas após todas elas serem feitas. Mas a leitura em tempo real é possível.

Após a aquisição dos arquivos binários é utilizado o programa *parserdotripe* (Aprendi e A) para fazer a leitura do arquivo e organizar as informações em uma matriz com três colunas e o número de pontos detectados como linhas. Plotamos então os pontos e as margens do ambiente para análise de resultados.

1.14.1 Medida 1 (altura de 53.5 cm com 2 dimensões)

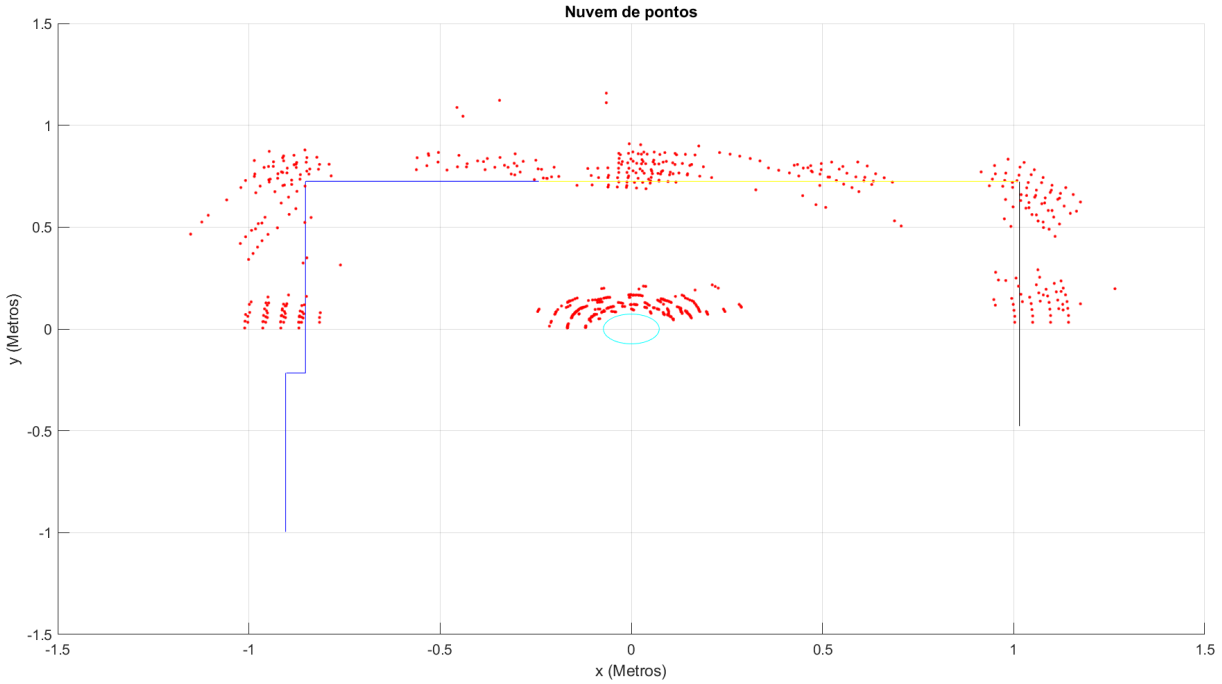


Figura 4.1 – Visão superior dos pontos

Analisando a distribuição da nuvem de pontos em duas dimensões, é perceptível como o sensor identifica cada material e cada parede. Em azul temos as paredes de tijolos, onde os primeiros pontos que mostram o contato com um objeto estão com as medidas dentro da resolução estabelecida na configuração do sensor. A espessura dela também poder ser vista pela concentração dos pontos. Alguns pontos de exemplo comparados com a distância real e espessura da parede para vários experimentos podem ser vistos na Tabela 4.1 no capitulo 4.2:

Em amarelo, a posição da janela, os pontos seguem um padrão parecido com os pontos referentes à parede de tijolos, ou seja, os primeiros pontos estão dentro da resolução escolhida para o sensor. Entretanto, a concentração de pontos que definiria a espessura da janela não está correta, já que a espessura da janela e vidro é de 2 centímetros, e a diferença entre o ponto de começo e fim de objeto tem uma espessura de 20 centímetros. Isto acontece porque a altura de medição do sensor está levemente abaixo do limite entre a parede e o vidro. Logo, se for levada em consideração a espessura da parede, teremos uma medida dentro da resolução.

Por fim, os pontos situados onde está a parede de madeira, em preto, têm uma distribuição que não condiz com os outros materiais, isto porque, analisando os pontos nessa região, nota-se que os pontos de começo de objeto estão no limite da resolução do sensor, o que ainda é aceitável para uma medida precisa, mas os pontos que definiriam a espessura têm uma distância de aproximadamente 14 centímetros, bem diferente da espessura real de 5 centímetros. Um estudo sobre o comportamento do sensor quando medindo madeira é uma das sugestões para o futuro, mas esse não é o propósito deste trabalho.

Notam-se também ‘buracos’ na continuidade das medições. Isto acontece por dois motivos: primeiro, é a resolução de rotação do tripé que é de 30 graus; segundo, a quantidade de pontos lidos por cada passo, que tem um limite de 100 pontos por frame, para não haver sobrecarga do processo de aquisição de dados ao computador utilizado.

Esses ‘buracos’ podem ser resolvidos utilizando um passo de medição menor, e com um computador com melhor processamento é possível aumentar o número de pontos adquiridos por frame.

Por último, devem ser comentados os pontos que estão perto do sensor. Tratam-se de ruídos provenientes do ambiente e de interferências do suporte do sensor. Os erros de ambiente deveriam ser menores pela aplicação da operação CFAR, mas esses erros podem significar que o limiar calculado pelo sensor na utilização do CFAR está baixo. A solução seria a mudança de alguns parâmetros antes da gravação do arquivo binário que é feito antes da calibração. Já os erros provenientes da interferência do suporte é que o sensor está detectando o suporte e ou o tripé em seu limite de angulação. Por serem pontos com distâncias muito baixas e no limite angular de detecção, tornam-se pontos com um grau de erro muito maior e, por isso, não confiáveis.

A quantidade total de pontos desta medida foi de 1.034 pontos.

1.14.2 Medida 2 (altura de 77.4 cm com 3 dimensões)

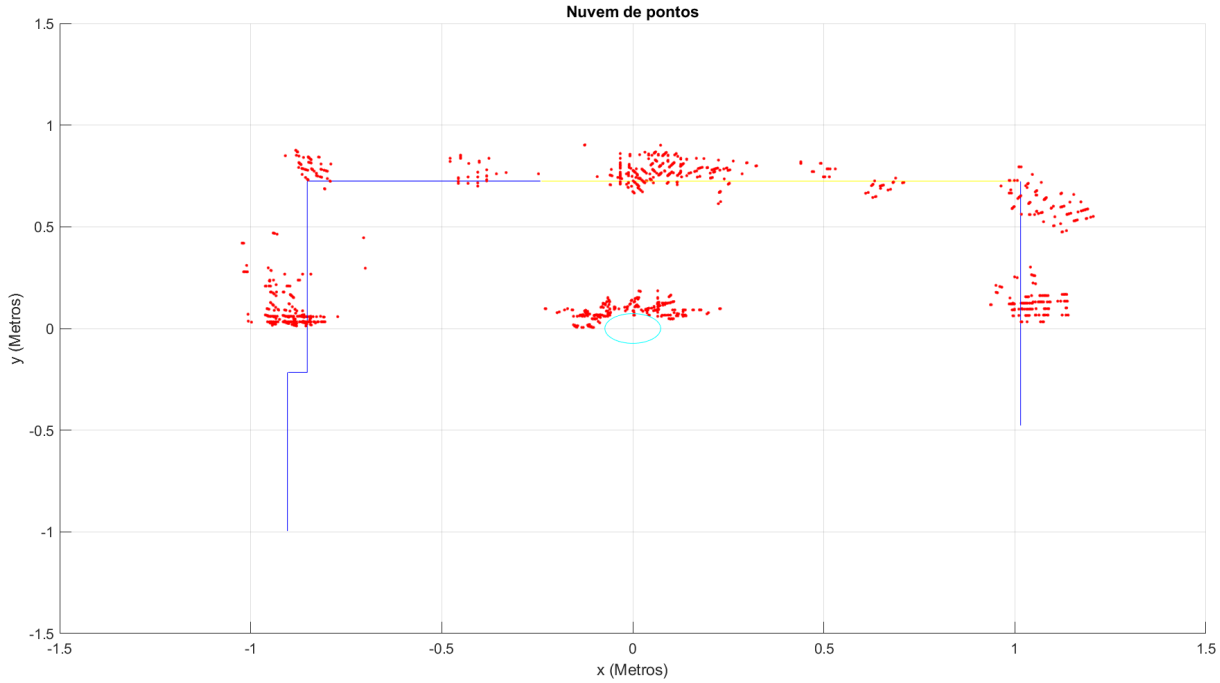


Figura 4.2 – Visão superior dos pontos (x,y)

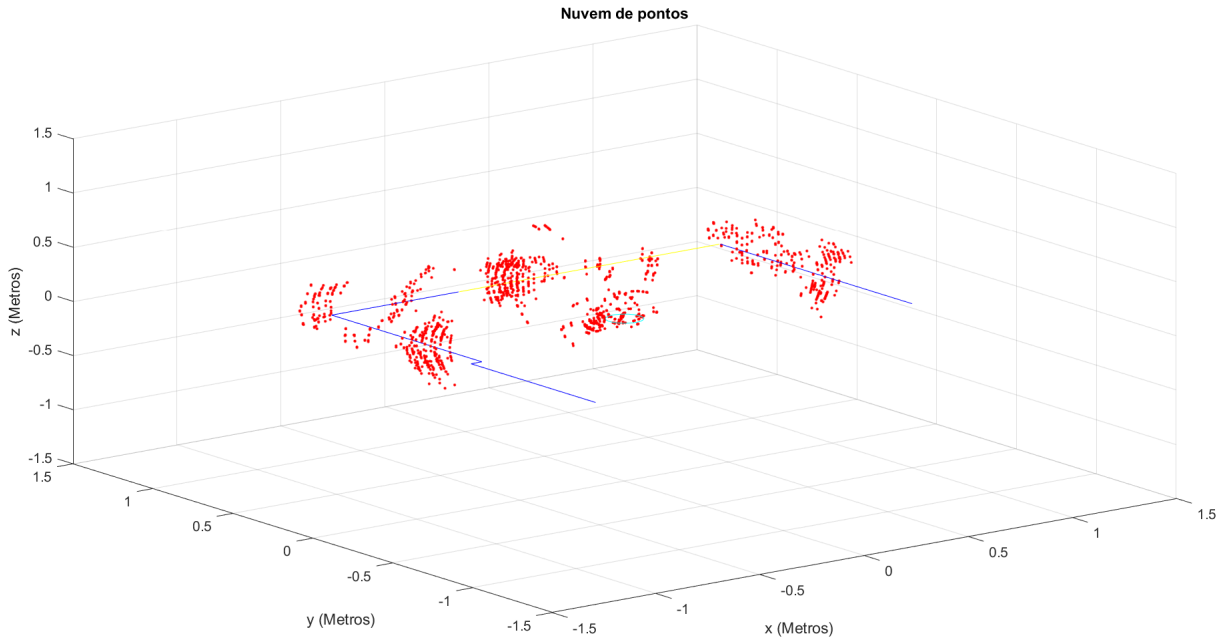


Figura 4.3 – Visão de todos os eixos

Os pontos na medida de três dimensões seguem os mesmos padrões que na medida com duas dimensões quando são analisadas as coordenadas do plano xy (Figura 4.2). Mas, agora, temos informação de elevação também (Figura 4.4). É possível verificar a distribuição dos pontos dentro dos materiais, enquanto na parede temos uma distribuição em forma de uma onda. Quando olhamos a distribuição dos pontos na região da janela, não temos uma forma definida. Isto porque, nessa região, temos o encontro de três materiais: a parede no limite do frame da janela, o frame da janela que é feito de metal e a janela feita de vidro.

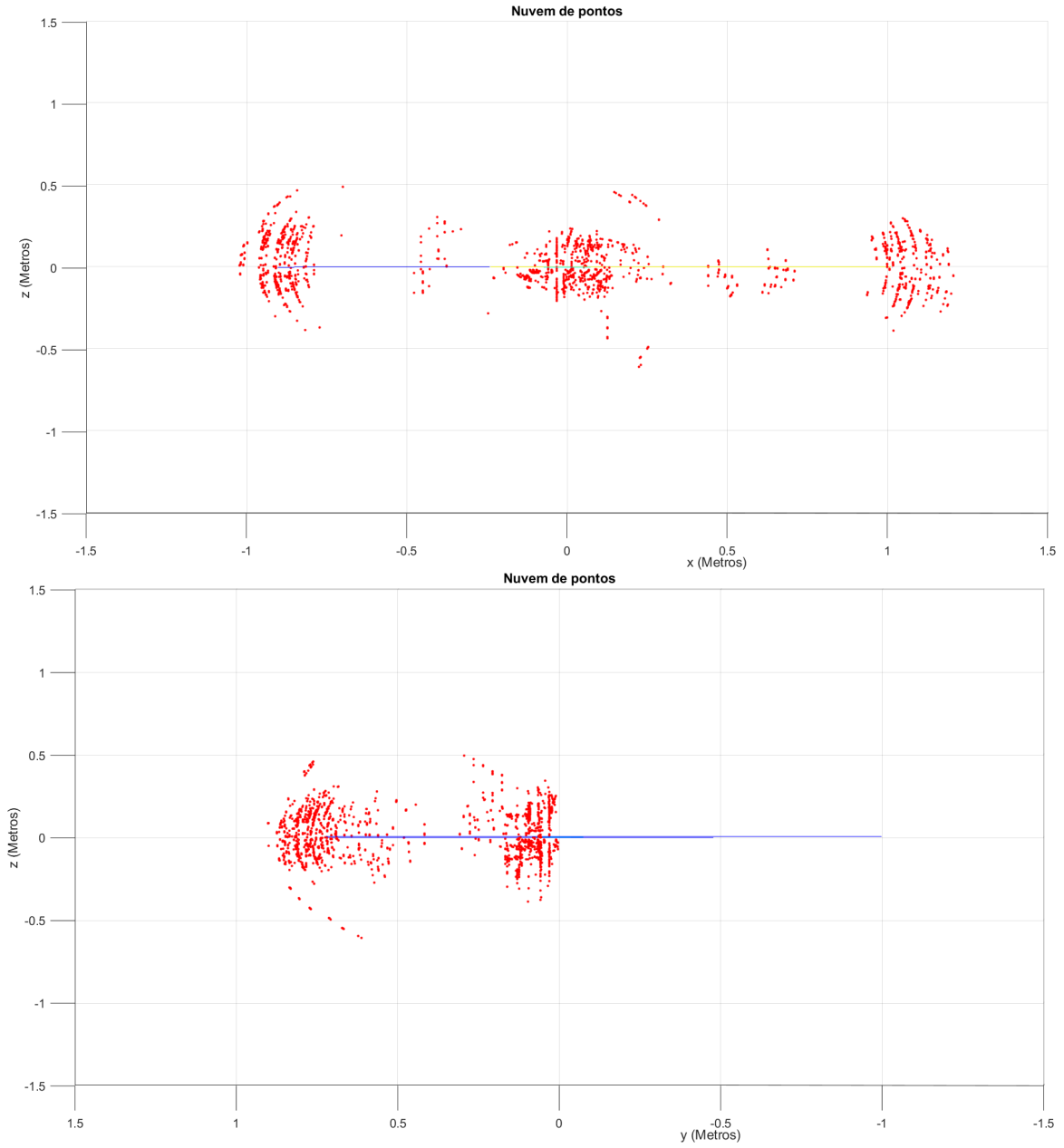


Figura 4.4 – As duas vistas com relação à altura

Nesta medida foram gerados 4902 pontos

1.14.3 Medida 3 (altura de 110 cm com 2 dimensões)

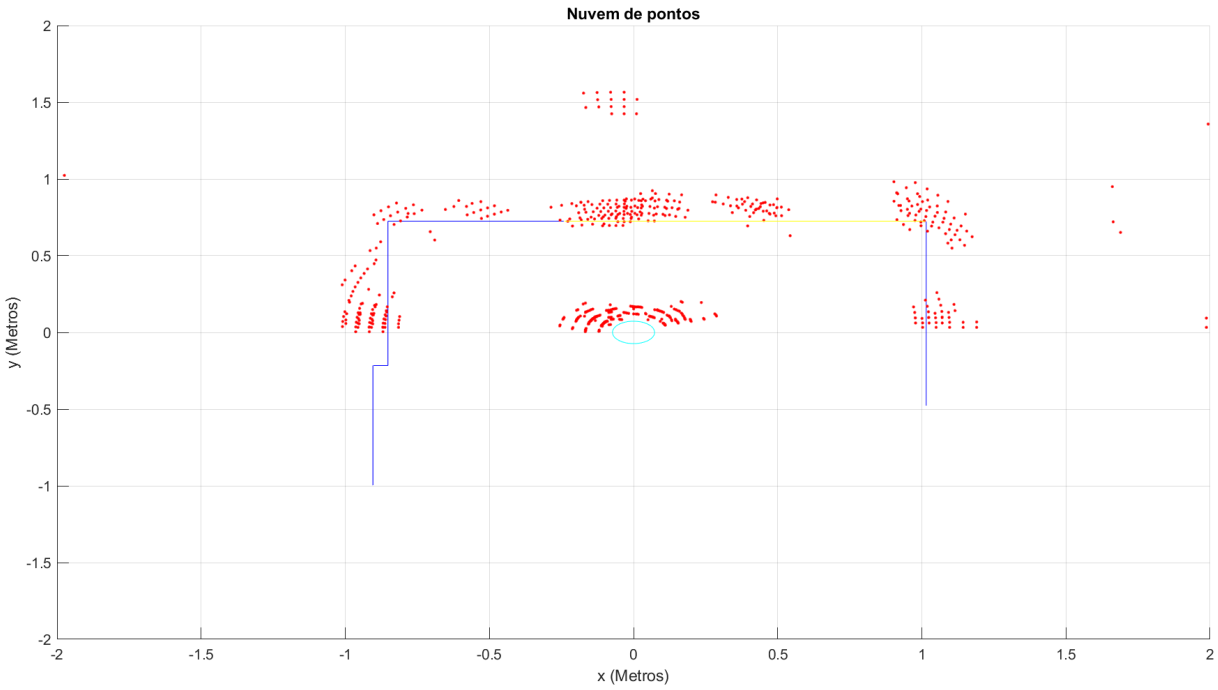


Figura 4.5 – visão superior dos pontos

Agora a medida é feita a uma altura de 110 centímetros. Se compararmos com as medidas feitas a 53,5 centímetros, não há diferença no padrão de distribuição dos pontos, com uma exceção: agora a altura está na metade da altura da janela, o que significa que o único material à frente do sensor é o vidro. Mas como pode ser analisado na Figura 4.5, a espessura na região da janela continua a mesma. Isso requer um estudo específico do comportamento do sensor em vidros. Alguns pontos foram identificados passando a região da janela, detectando objetos do lado de fora da cena. Esses pontos são ruídos, provavelmente produzidos por causa do vidro, já que nessas coordenadas não existe nenhum obstáculo nessa altura.

Nesta medida foram gerados 7.267 pontos.

O aumento considerável na quantidade de pontos é proveniente da interação do sensor com o vidro, gerando detecção de pontos que não existem.

1.14.4 Medida 4 (altura de 110 cm com 3 dimensões)

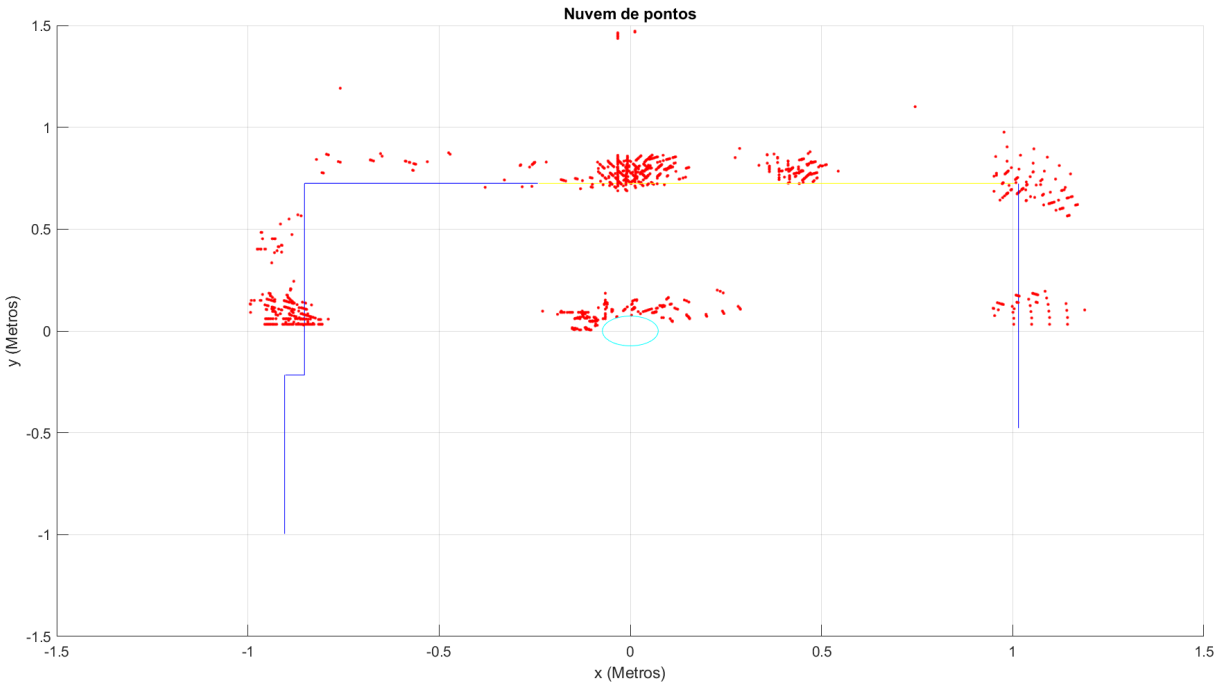


Figura 4.6 – Visão superior dos pontos (xy)

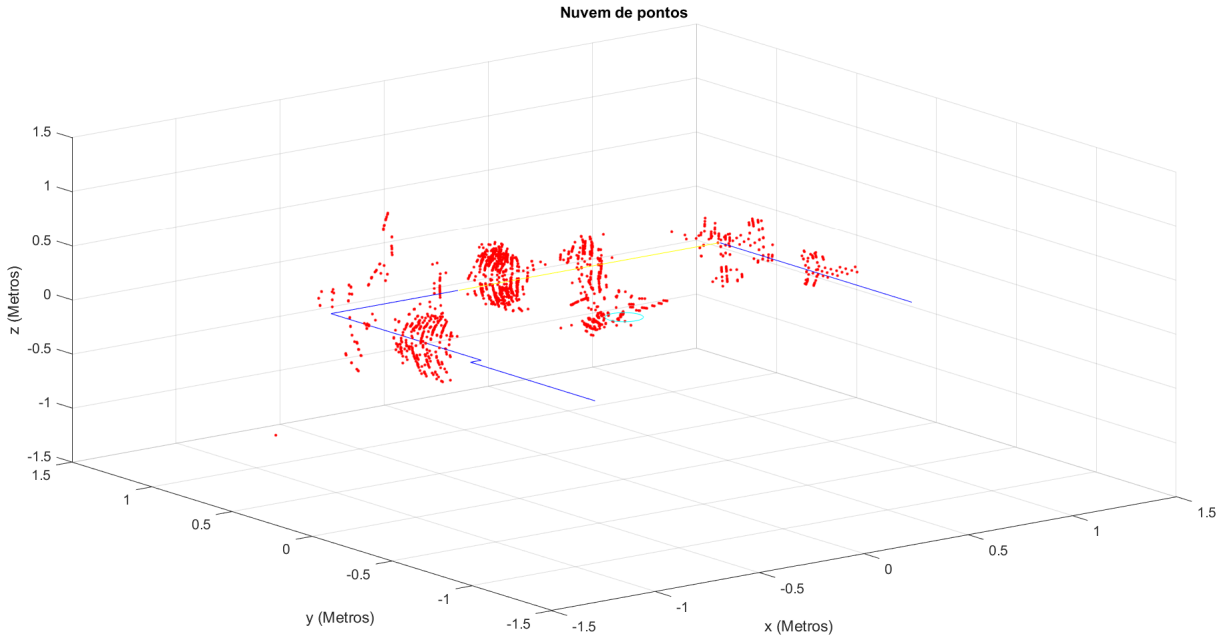


Figura 4.7 – Imagem de todas as coordenadas (xyz)

Nesta medição encontra-se uma pequena diferença em relação à medição 2. Comparando a Figura 4.5 com a Figura 4.6, nota-se um aumento nos pontos na região da janela, e agora não há interferência dos outros materiais. Por isso, os pontos na região da janela têm um padrão de onda, mesmo que as dimensões das medidas estejam fora da resolução do sensor. Por isso é importante que em trabalhos futuros seja empregada uma atenção especial ao material vidro.

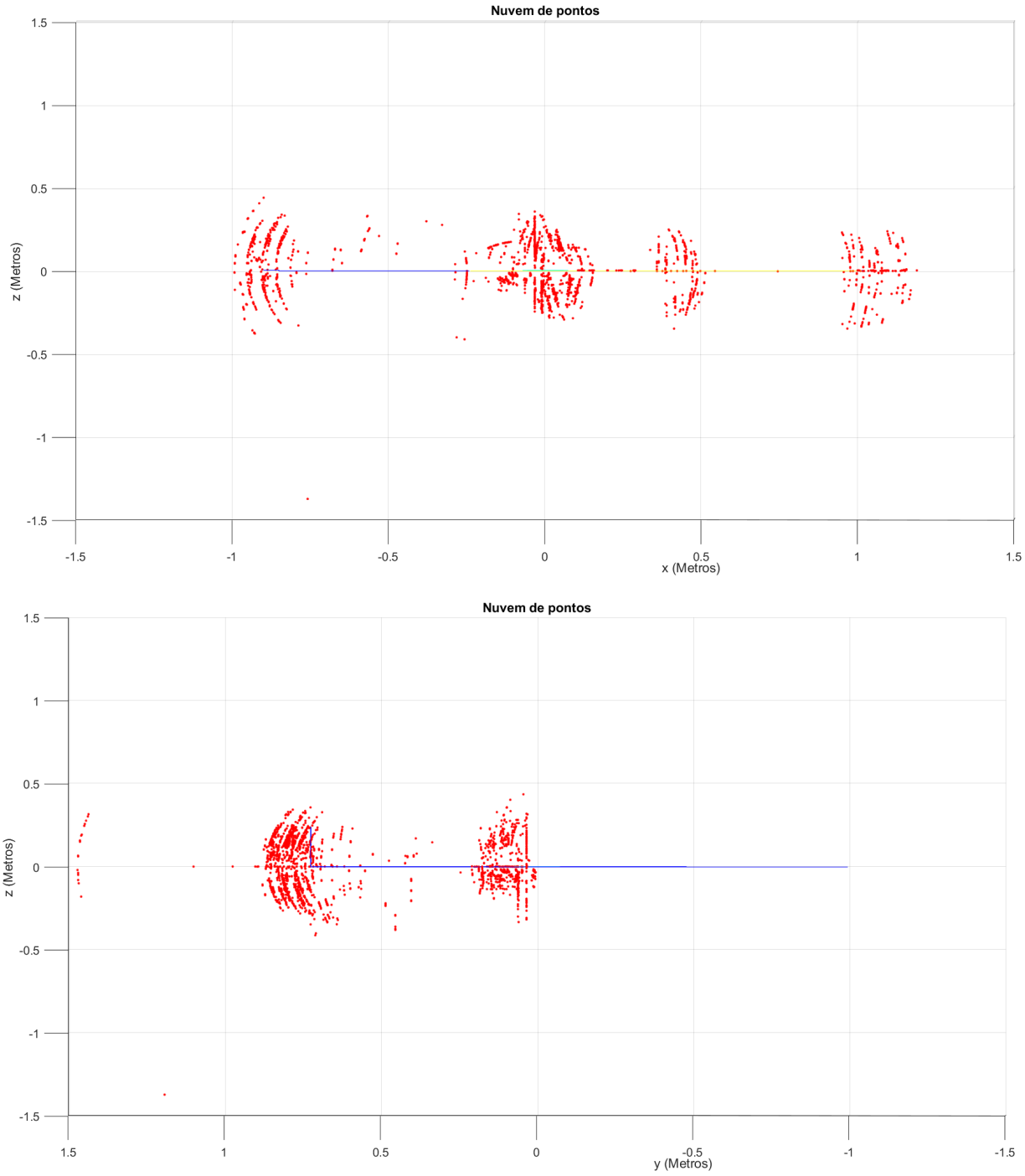


Figura 4.8 – Visões da altura em relação às coordenadas x e y.

Nesta medida foram gerados 5.198 pontos.

1.15 Comparativo das medidas

Na tabela 4.1 foram colocados pontos de cada media, onde os pontos 1 são os primeiros pontos de detecção da parede de tijolos, os pontos 2 são a detecção da janela e os pontos 3 a detecção da parede de madeira. Todas as medidas reais foram retiradas com fita métrica com uma precisão de 1 milímetro.

Tabela 4.1 - tabela comparativo de medidas

Medida	Ponto 1			Ponto 2			Ponto 3		
	x	y	z	x	y	z	x	y	z
Real	-0.852	0.000	0.000	0.000	0.725	0.000	1.015	0.000	0.000
1	-0.815	0.056	NA	-0.012	0.694	NA	0.948	0.144	NA
2	-0.807	0.056	-0.107	0.038	0.666	0.185	0.960	0.206	0.109
3	-0.815	0.032	NA	-0.036	0.697	NA	0.978	0.061	NA
4	-0.789	0.058	0.332	-0.008	0.688	0.093	0.952	0.075	0.212

CAPÍTULO 5 – CONCLUSÃO

Este trabalho apresentou uma introdução a sensores radares e à tecnologia FMCW. Também foi demonstrado o funcionamento do sensor em um ambiente fechado e com diferentes materiais. Em geral, o sistema apresentou medidas precisas e um funcionamento aceitável nesse tipo de ambiente. Entretanto, podemos observar que o sensor teve uma resposta sobre a largura dos materiais não muito satisfatória.

Se compararmos com as tecnologias existentes lasers, essas que se assemelham bastante no funcionamento e poderiam ser aplicadas igualmente, podemos notar que elas são mais precisas que as tecnologias radares por alguns poucos centímetros. A tecnologia de radares, porém, continua avançando e melhorando, e pode-se esperar uma redução nessa diferença de precisão. Neste trabalho não foi testada a precisão de velocidade e, por isto, não é possível comparar com aplicações lasers.

5.1 Sugestões para trabalhos futuros

O próximo passo para o estudo dos sensores radares é um estudo para resposta para diferentes materiais e diferentes espessuras. Como pudemos ver, houve uma discrepância entre os tamanhos reais e os tamanhos medidos, em especial nos materiais de vidro e madeira.

Um dos avanços sugerido para a medição de espaços 3D com sensores radares é o acoplamento do sensor a motores, com precisão de rotação como os motores de passo, para aumentar a resolução de rotação do sensor no eixo horizontal e ampliar a quantidade de amostras, reduzindo os ‘buracos’ que foram vistos na seção de experimentos.

REFERÊNCIAS

[1] **The fundamentals of millimeter wave sensors** - Cesar Iovescu Radar, SandeepRao,TexasInstrument. Disponível em: <https://www.ti.com/lit/wp/spyy005a/spyy005a.pdf?ts=1610092031157&ref_url=https%253A%252F%252Fwww.ti.com%252Fsensors%252FmmWave-radar%252Findustrial%252Fapplications%252Fapplications.html> acesso em 05/07/2020

[2] Tabela 2- Comparativo entre os sensores e suas configurações. Disponível em: <http://dev.ti.com/tirex/explore/node?node=ALYHkY2xyhq2-EgWIUuw-Q__VLyFKFf__LATEST > acesso em 10/10/2020

[3] Guia "**OUT OFF THE BOX**" para o sensor IRW1443. Disponível em: <https://dev.ti.com/tirex/explore/node?node=ABasos6ekSODOVYPuftL5w__VLyFKFf__LATEST > acesso em 10/10/2020

[4] Guia "**OUT OFF THE BOX**" para o sensor IRW6843. Disponível em: <http://dev.ti.com/tirex/explore/node?node=ADSocWGX8FJCytCWREvHzw__VLyFKFf__LATEST > acesso em 10/10/2020

[5] ALDRICH Robert, **Laser Fundamentals Manual**. Disponível em: https://fas.org/man/dod-101/navy/docs/laser/fundamentals.htm

[6] Manual 1443. **Doxygen IWR1443**, disponibilizado pela Texas Instruments. disponível em: https://www.ti.com/tool/MMWAVE-SDK

[7] I. S. Reed, Y. L. Gauand T. K. Truong, **CFAR detection and estimation for STAP radar in IEEE Transactions on Aerospace and Electronic Systems**, vol. 34, no. 3, 1998 p. 722-735

[8] Gonzales, R.C., Woods, R.E., **Digital ImageProcessing**, USA Addison-Wesley Pubisshing Company, 1993.

[9] FISCHER, Kaspar **Introduction to Alpha Shapes**. V. 2000

[10] WATSON, A.; ULRICH, J. **Pidar: 3D Laser Range Finder.** Senior Design Documentation – University of Central Florida ,2014.

[11] Fonseca. João Neto. "**Aplicação da Transformada de Fourier no PROCESSAMENTO DIGITAL DE IMAGENS**''. Aracaju- Se, novembro de 1999.

[12] Torres Ferreira, Guilherme “**sistema de mapeamento tridimensional de ambientes**”. Trabalho de conclusão de curso apresentado à escola de engenharia São Carlos, da universidade de São Paulo.

[13] MORALES, J. ET AL. **Design and development of a fast and precise Low-cost 3d Lasr rangedinder in Mechatronics** (ICM) IEEE INTERNATIONAL CONFERENCE ON IEEE, 2011. P. 621-626,2011

[14] França, J.; GAZZIRO, M, **A 3D scanning system based on laser triangulation and variable field of view**. IEE International Conference on Image Processing, ICIP 2005. IEE, vol.5

APÊNDICE - A

Código em matlab

```
clear all
close all
arquivos(1,:) = ['090.dat'];
arquivos(2,:) = ['060.dat'];
arquivos(3,:) = ['030.dat'];
arquivos(4,:) = ['000.dat'];
arquivos(5,:) = ['-30.dat'];
arquivos(6,:) = ['-60.dat'];
arquivos(7,:) = ['-90.dat'];
angulopre = [-90,-60,-30,0,30,60,90];
contador = 1;
sindo=0;
rotacao=0;
figure;
paredey =  0.723:-0.01:-0.480;
paredex = 1.015+0*paredey;
paredez = 0+0*paredey;
plot3(paredex,paredey,paredez,'-','Color','blue');
hold on;
title('Nuvem de pontos');
xlabel('x (Metros)');
ylabel('y (Metros)');
zlabel('z (Metros)');
xlim([-2 2]);
ylim([-2 2]);
zlim([-2 2]);
%sensor
raio = 0.073
angulos = 0:1:360;
angulos = deg2rad(angulos);
paredex = (raio*cos(angulos));
paredey = (raio*sin(angulos));
paredez = 0+0*paredex;
plot3(paredex,paredey,paredez,'-','Color','cyan');
hold on;
paredey = -0.216:0.01:0.725;
paredex = -0.852+0*paredey;
paredez = 0+0*paredey;
plot3(paredex,paredey,paredez,'-','Color','blue');
hold on;
paredex = -0.852:-0.01:-0.904;
paredey = -0.216+0*paredex;
paredez = 0+0*paredex;
plot3(paredex,paredey,paredez,'-','Color','blue');
hold on;
paredey = -0.216:-0.01:-1;
paredex = -0.904+0*paredey;
paredez = 0+0*paredey;
plot3(paredex,paredey,paredez,'-','Color','blue');
hold on;

paredex = -0.852:0.01:-0.239;
```

```
paredey = 0.725+0*paredex;
paredez = 0+0*paredex;
plot3(paredex,paredey,paredez,'-','Color','blue');


%janela

paredex = -0.239:0.01:1.015;
paredey = 0.725+0*paredex;
paredez = 0+0*paredex;
plot3(paredex,paredey,paredez,'-','Color','blue');
hold on;

paredez = 0:0.01:0.24;
paredex = -0.239+0*paredez;
paredey = 0.725+0*paredez;
plot3(paredex,paredey,paredez,'-','Color','blue');
hold on;

paredez = 0:0.01:0.24;
paredex = 1.015+0*paredez;
paredey = 0.725+0*paredez;
plot3(paredex,paredey,paredez,'-','Color','blue');
hold on;


%janela

paredex = -0.239:0.01:1.015;
paredey = 0.725+0*paredex;
paredez = 0.24+0*paredex;
plot3(paredex,paredey,paredez,'-','Color','yellow');
hold on;


%obstaculo

% paredex = 0.375:0.01:1.015 ;
% paredey = -0.480+0*paredex;
% paredez = 0+0*paredex;
% plot3(paredex,paredey,paredez,'-','Color','green');
% hold on;
%pause();
x=[];
y=[];
z=[];
while( contador<=7)
%pause();
%rotacao= input('angulo de rotação ');
rotacao=deg2rad(angulopre(contador));
%nomearquivo=input('digite o nome do arquivo: ','s');
pontosx = [];
pontosy = [];
pontosz = [];
file = fopen (arquivos(contador,:));
%cabeçalho
range_indext = [];
doppler_indext = [];
peak_valuet = [];
```

```
medidas =0;
while (~feof(file))
tamanhodestepacote =0;
header = fread (file , 4 , 'uint16');% magic word
if (length(header) < 4)
 'erro no header, finalizando operação (header vazio)'
 continue;
end
if (header(1) ~= 258 || header(2)~=772 || header(3)~=1286 ||
header(4)~=1800)
    'erro no header, tentando novamente(header errado)'
    fseek(file, 6 , 'cof');
    continue;
end
header = dec2hex(header);

versao = fread (file, 1 , 'uint32');; % MajorNum * 2^24 + MinorNum * 2^16 +
BugfixNum * 2^8 + BuildNum
versao = dec2hex(versao); % MajorNum * 2^24 + MinorNum * 2^16 + BugfixNum *
2^8 + BuildNum

tamanho  = fread (file, 1 , 'uint32'); %tamanho do pacote

plataforma  = fread (file, 1 , 'uint32');;%plataforma de desenvolvimento
plataforma = dec2hex(plataforma);

frames  = fread (file, 1 , 'uint32');%numero de frames

tempo = fread (file,1 , 'uint32');%tempo em clicos de cpu da duração do
pacote

objetos = fread (file,1 , 'uint32');%numero de objetos detectados

tvls = fread (file, 1 , 'uint32');%numero de tvls

%framesub = fread (x, 1 , 'uint32')%For Advanced Frame config, this is the
sub-frame number in the range 0 to (number of subframes - 1). For frame
config (not advanced), this is always set to 0
aux=0;
%pacote= [ tvls , 3]
% tvls

while (aux<tvls)

identificadordotvl=fread(file,1,'uint');
tamanhodotvl=fread(file,1,'uint');
tamanhodestepacote = tamanhodotvl+tamanhodestepacote;
conteudo=[];

if(identificadordotvl==1) % detected objects
    medidas=medidas+1;
    descritor=fread(file,2,'uint16');
    nuemrodeobjetoslocais=descritor(1); %prieiro argumento sendo o numero
de objetos neste tvl
    xyzq=descritor(2); %segundo argumento sendo o formato das cordenadas
    i=0;
    range_index=[];
    doppler_index=[];
    peak_value=[];
```

```
    x_coordinate=[];
    y_coordinate=[];
    z_coordinate=[];
    while(i~=nuemrodeobjetoslocais)
        i=i+1;
        range_index=[range_index,fread(file,1,'uint16')];
        doppler_index=[doppler_index,fread(file,1,'uint16')];
        peak_value=[peak_value,fread(file,1,'uint16')];
        % x,y,z podem ser retirados dividindo por 2^xyzq
        x_coordinate=[x_coordinate,fread(file,1,'int16')/2^xyzq];
        y_coordinate=[y_coordinate,fread(file,1,'int16')/2^xyzq];
        z_coordinate=[z_coordinate,fread(file,1,'int16')/2^xyzq];
    end


elseif(identificadordotvl==2) % range profile

    lixo = fread (file,tamanhodotvl,'uint8');
    %size_range = 256;

    %range_profile_log=fread(file,size_range,'uint16');

    %conteudo={range_profile_log};

elseif(identificadordotvl==3) % noise profile

    lixo = fread (file,tamanhodotvl,'uint8');

    %size_range=256;

   % noise_profile_log=fread(file,size_range,'uint16');

    %conteudo={noise_profile_log};

elseif(identificadordotvl==4)

    lixo = fread (file,tamanhodotvl,'uint8');

    %virtual_ant_num=rx_num*tx_num;

    %size_rahp=range_bins_num*virtual_ant_num;

    %tmp_ra_heatmap=fread(file,size_rahp*2,'uint16');

    %tmp_complex=zeros(size_rahp,1);

    %dif=0;
    %for i=1:2:size_rahp*2-1

     %   index=i-dif;

     %   tmp_complex(index)=complex(tmp_ra_heatmap(i+1),tmp_ra_heatmap(i));

      %  dif=dif+1;

   % end
   % ra_heatmap=reshape(tmp_complex.',virtual_ant_num,range_bins_num);
```

```
    %ra_heatmap=ra_heatmap.';

    %conteudo={ra_heatmap};

elseif(identificadordotvl==5)
    lixo = fread (file,tamanhodotvl,'uint8');

    %doppler_bins_num=chirpsperframe/tx_num;

    %size_rdhp=range_bins_num*doppler_bins_num;

    %tmp_rd_heatmap=fread(file,size_rdhp,'uint16');

    %rd_heatmap=reshape(tmp_rd_heatmap.',doppler_bins_num,range_bins_num);

   % rd_heatmap=rd_heatmap.';

    %conteudo={rd_heatmap};

elseif(identificadordotvl==6)

    lixo = fread (file,tamanhodotvl,'uint8');

    %interframe_process_time=fread(file,1,'uint');

   % transmit_output_time=fread(file,1,'uint');

    %interframe_prcocess_margin=fread(file,1,'uint');

    %interchirp_process_margin=fread(file,1,'uint');

   % active_frame_cpu_load=fread(file,1,'uint');

    %interframe_cpu_load=fread(file,1,'uint');


%conteudo={interframe_process_time;transmit_output_time;interframe_prcocess
_margin;interchirp_process_margin;active_frame_cpu_load;interframe_cpu_load
};
end
%TLV={identificadordotvl;tamanhodotvl;conteudo};
%pacote(aux+1,:)= TLV;
aux= aux+1;
end

padding = -(tamanhodestepacote+7*4+2+tvls*8)+tamanho;
if(padding>0)
lixo = fread(file,padding-6,'uint8');
end
range_indext = [range_indext,range_index];
doppler_indext = [doppler_indext,doppler_index];
peak_valuet = [peak_valuet,peak_value];
pontosx = [pontosx,x_coordinate];
pontosy = [pontosy,y_coordinate];
pontosz = [pontosz,z_coordinate];
```

```
end
fclose(file);

%rotação dos pontos caso necessario
j=1;
sizedepontos=size(pontosx);
while (j<=sizedepontos(2))
    b = pontosx(j);
    c = pontosy(j);
    a=(b^2)+(c^2);
    if b~=0
    alfa = atan(c/b);
    else
        alfa = 1.5708;
    end
    d = sqrt(a);
    if b<0
        d=-d;
    end
    deslocamentosensorx=   0.073*cos(rotacao+90);
    deslocamentosensory=    abs(0.073*sin(rotacao+90));
    pontosx(j) =deslocamentosensorx + d*cos(rotacao+alfa);
    pontosy(j) =deslocamentosensory + abs(d*sin(rotacao+alfa));
    j= j+1;
end
plot3 (pontosx,pontosy,pontosz,'.','color','red')
hold on;

contador= contador+1;
x = [x;pontosx(:)] ;
y = [y;pontosy(:)] ;
z = [z;pontosz(:)] ;
%sindo=input('sair?[1]continuar?[2] ');



end
grid on;
%plot3(x,y,z,'.','color','red');
hold off;
%remover pontos perto do sensor(ruidos)
n=1;
aux=size(x)
while n<aux(1)
    if (abs(x(n))<0.3 && abs(y(n))< 0.3)
        x(n)=0;
        y(n)=0;
        z(n)=0;
    end
    n=n+1;
end
%shp = alphaShape(x,y,z,0.3);
figure;
plot3(x,y,z,'.','color','red');
figure;
%plot(shp);
shp = alphaShape(x,y,0.1);
shp.RegionThreshold = 0.001;
plot(shp)
grid on;
p = [x y z] ;
```

```
fid = fopen('test.xyz', 'wt');
fprintf(fid, [repmat('%.5f\t', 1, size(p,2)-1) '%f\n'],p.');
fclose(fid);

% [X,Y] = meshgrid(-2:0.1:2)
% Z=0*X+0*Y;
% surface(X,Z,Y);
```