

Universidade Federal de Juiz de Fora  
Instituto de Ciências Exatas  
Programa de Pós-Graduação em Física

**Alaor do Nascimento Freire Diniz**

**Sistema interferométrico com tratamento de imagem**

Juiz de Fora  
2019

**Alaor do Nascimento Freire Diniz**

**Sistema interferométrico com tratamento de imagem**

Dissertação apresentada ao Programa de Pós-Graduação em Física da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Mestre em Física.

Orientador: Dr. José Paulo Rodrigues Furtado de Mendonça

Coorientador: Dr. Rodrigo Alves Dias

Juiz de Fora

2019

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF  
com os dados fornecidos pelo(a) autor(a)

Diniz, Alaor do Nascimento Freire.

Sistema interferométrico com tratamento de imagem / Alaor do Nascimento Freire Diniz. – 2019.

81 f. : il.

Orientador: Dr. José Paulo Rodrigues Furtado de Mendonça

Coorientador: Dr. Rodrigo Alves Dias

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Física, 2019.

1. Interferometria. 2. Variação de diferença de fase. 3. Variação do índice de refração. 4. Imagem interferométrica. I. Mendonça, José Paulo Rodrigues Furtado, orient. II. Dias, Rodrigo Alves III. Título.

**Alaor do Nascimento Freire Diniz**

**" Sistema interferométrico com tratamento de imagem."**

Dissertação apresentada ao Programa de Pós-graduação em Física, da Universidade Federal de Juiz de Fora como requisito parcial a obtenção do grau de Mestre em Física. Área de concentração: Física

Aprovada em 05 de agosto de 2019.

BANCA EXAMINADORA



Prof. Dr. José Paulo Rodrigues Furtado de Mendonça - Orientador  
Universidade Federal de Juiz de Fora - UFJF



Prof. Dr. Rodrigo Alves Dias - Coorientador  
Universidade Federal de Juiz de Fora - UFJF



Prof. Dr. Pedro Claudio Guarinho de Moraes  
Universidade Federal de São João del Rei - UFSJ



Prof. Dr. Benjamin Fragneaud  
Universidade Federal de Juiz de Fora - UFJF

## AGRADECIMENTOS

Primeiramente a Deus que permitiu que tudo isso acontecesse, ao longo de minha vida, e não somente nestes anos como universitário, mas em todos os momentos é o maior mestre que alguém pode conhecer.

Alan do Nascimento Freire Diniz, meu irmão. Agradeço pela confiança e incentivos depositados nessa minha árdua aventura pelo mundo da Física. Não fosse ele, não estaria aqui escrevendo essas linhas convicto de ter feito a escolha certa. Meus pais João Mendes Diniz Filho e Jovelina Regina do Nascimento Freire Diniz, os quais sempre estiveram ao meu lado apoiando todas as minhas decisões.

José Paulo Rodrigues Furtado de Mendonça, meu orientador, o qual me mostrou que a física é complicada, entretanto, ela nos fornece a compreensão de incríveis fenômenos. Meus co-orientadores Rodrigo Alves Dias e Wallon Anderson Tadaiesky Nogueira que tiveram calma e paciência para suprir todas as dúvidas ao longo desse caminho. Tenho muito que agradecer aos meus Orientadores e espero continuar a trabalhar com eles.

Também não posso deixar de agradecer a todos os professores do UFJF. Em especial, Pablo Zimmermann Coura e Sidiney de Andrade Leonel que participaram diretamente na minha formação e proporcionaram muitas gargalhadas na sala do café.

Não posso esquecer de meus amigos que, de um modo ou de outro, fizeram parte do meu cotidiano nestes dois anos da UFJF.

Por fim, gostaria de agradecer a UFJF, o departamento de Física e seus funcionários, principalmente ao Domingos e Pedro Paulo, pela sua paciência e competência. Agradeço também às agências de fomento CAPES, CNPq, FAPEMIG, FINEP

"Se você está pensando em desistir, então desista de pensar."

**Zyzz**

## RESUMO

Nesta dissertação será apresentada a automação e o tratamento de imagens interferométricas do Interferômetro de Atividade Metabólica (*Metabolic Activity Interferometer - MAI*). Este equipamento permite analisar em tempo real as atividades metabólicas de organismos biológicos. As modificações apresentadas do MAI, nesta dissertação, foram desenvolvidas com o objetivo de realizar a atualização dos *softwares* e *hardwares*, com o propósito de obter medidas, da variação da diferença de fase e do índice de refração de uma forma mais rápida. Buscamos também a redução da eletrônica do equipamento, a fim de que, no futuro, possamos transformar tal aparato em uma tecnologia que possivelmente de ser comercializada no mercado. No decorrer do trabalho, apresentamos as alterações no MAI e detalhes de como os dados interferométricos foram analisados juntamente com os dados obtidos de variação de diferença de fase e variação do índice de refração.

Palavras-chave: Interferometria. Variação de diferença de fase. Variação do índice de refração. Imagem interferométrica.

## ABSTRACT

In this dissertation we will present the automation and the treatment of interferometric images of the Metabolic Activity Interferometer (MAI). This equipment allows to analyze in real time the metabolic activities of biological organisms. The modifications presented by the MAI in this dissertation were developed with the objective of updating the *softwares* and *hardwares*, in order to obtain faster measurements of phase difference and refractive index and the reduction of equipment electronics, so that in the future transform such apparatus into a technology that can be commercialized in the market. Throughout the work we present the changes in the MAI and details of how the interferometric data were analyzed together with the data obtained from phase difference variation and refractive index variation.

Keywords: Interferometry. Phase difference variation. Refractive index variation. Interferometric image.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Esquema de duas ondas eletromagnéticas percorrendo caminhos óticos diferentes até se interferirem . . . . .	15
Figura 2 – Esquematização básica do interferômetro de Fabry-Perot, imagem da referência [1], onde $\mathbf{E}_0$ é a onda incidente inicial $\mathbf{E}_1$ , $\mathbf{E}_2$ , $\mathbf{E}_3$ e $\mathbf{E}_4$ são as ondas transmitidas após a onda percorrer o caminho $D$ , $\mathbf{E}'_1$ é a onda refletida ao incidir na placa superior e $\mathbf{E}'_2$ , $\mathbf{E}'_3$ e $\mathbf{E}'_4$ são as ondas transmitidas após sofrerem reflexão na placa inferior . . . . .	17
Figura 3 – Arranjo básico de um interferômetro de Michelson, imagem retirada [2]: S é a fonte que emite o feixe laser, B.S é o divisor de feixe e $B_1$ e $B_2$ os caminhos percorridos pelo feixe após passar pelo B.S . . . . .	18
Figura 4 – Projeção de uma imagem interferométrica, onde $t_1$ e $t_2$ variáveis temporais e adotando valores de $a = 0.083s^{-1}$ , $b = 0s^{-1}$ . . . . .	21
Figura 5 – Módulo da transformada de Fourier de uma imagem interferométrica adotando $a = 0.083s^{-1}$ e $b = 0s^{-1}$ . No qual (a) e (b) representa sua visualização em 3D e sua projeção em $u$ e $v$ respectivamente. . . . .	21
Figura 6 – Projeção de uma imagem interferométrica, onde $t_1$ e $t_2$ variáveis temporais e adotando valores de $a = 0.083s^{-1}$ , $b = -0.02s^{-1}$ . . . . .	22
Figura 7 – Módulo da transformada de Fourier de uma imagem interferométrica adotando $a = 0.083s^{-1}$ e $b = -0.02s^{-1}$ . No qual (a) e (b) representa sua visualização em 3D e sua projeção em $u$ e $v$ respectivamente. . . . .	22
Figura 8 – Efeito de Leakage, onde a curva de azul mostra um sinal co-senoidal, a curva de vermelho é a janela introduzida do sinal co-senoidal inicial e a curva de verde representa a discretização obtida com a introdução da janela. Imagem modificada da referência [3] . . . . .	23
Figura 9 – Módulo da transformada de Fourier de uma imagem interferométrica adotando $a = 0.083s^{-1}$ e $b = 0s^{-1}$ . No qual (a) representa a transformada de Fourier sem utilização da janela <i>Hanning</i> , (b) a transformada de Fourier utilizando a janela <i>Hanning</i> , (c) representação da transformada de Fourier em 3D sem utilização da janela <i>Hanning</i> e (d) representação da transformada de Fourier em 3D utilizando a janela <i>Hanning</i> . . . . .	24
Figura 10 – Módulo da transformada de Fourier de uma imagem interferométrica adotando $a = 0.083s^{-1}$ e $b = -0.02s^{-1}$ . No qual (a) representa a transformada de Fourier sem utilização da janela <i>Hanning</i> , (b) a transformada de Fourier utilizando a janela <i>Hanning</i> , (c) representação da transformada de Fourier em 3D sem utilização da janela <i>Hanning</i> e (d) representação da transformada de Fourier em 3D utilizando a janela <i>Hanning</i> . . . . .	25

Figura 11 – Transformada de Fourier em aumentando o range dos valores de $t_1$ e $t_2$ na equação $I(t_1, t_2) = \cos[2\pi(at_1 + bt_2)]$ , onde adotamos $b = -0,02s^{-1}$	26
Figura 12 – Foto do MAI, onde (A) Disco de aço inoxidável, utilizado para sustentar os portas amostras; (B) conjunto laser com fotorresistor; (C) o servo motor; (D) contador de voltas; (E) interferômetro de Michelson, cujo aparatos ópticos estão abaixo da placa de aço inoxidável; (F) Fototransistor e (G) dois porta amostras, frisando que os porta amostras passam por baixo do sistema interferométrico descrito por (E).	27
Figura 13 – Representação esquemática do MAI, onde onde (A) é Disco de aço inoxidável, utilizado para sustentar os portas amostras; (B) representa o conjunto laser com fotorresistor; (C) o servo motor; (D) contador de voltas; (E) interferômetro de Michelson, cuja os aparatos ópticos estão abaixo da placa de aço inoxidável e (F) o Fototransistor	28
Figura 14 – Representação de um Arduíno Mega 2560, exibindo as portas digitas e analógicas e (A), que é o local para o cabeamento para realizar a comunicação direta com o computador.	29
Figura 15 – Foto retirada do motor de passo, onde (A) mostra a posição do motor de passo e (B) representa as engrenagens responsável pelo giro do disco.	30
Figura 16 – Esquema ilustrativo da eletrônica responsável pela rotação do disco juntamente com o Arduíno.	30
Figura 17 – Foto retirada do conjunto laser com LDR e do servo motor, mecanismos responsáveis de realizar o travamento do disco	31
Figura 18 – Esquema ilustrativo da eletrônica responsável pelo controle do conjunto laser com LDR juntamente com o servo motor e com o Arduíno.	31
Figura 19 – Foto retirada do contador de voltas.	32
Figura 20 – Zoom realizado da figura 19, com o proposito de observar o ressalto posto no disco	32
Figura 21 – Fluxograma da lógica do código responsável pelo funcionamento da parte mecânica.	33
Figura 22 – Fluxograma da lógica do código responsável pelo funcionamento da parte mecânica.	34
Figura 23 – Esquema ilustrativo da eletrônica responsável pelo controle do conjunto responsável pela aquisição de dados.	34
Figura 24 – Foto retirada do osciloscópio exibindo o sinal PWM enviado ao amplificador operacional, onde o eixo da ordenada está numa escala de $5V$ .	35
Figura 25 – Foto retirada do osciloscópio exibindo o sinal PWM modulado em um alcance de $-12V$ a $+12V$ , onde o eixo da ordenada está numa escala de $5V$ .	35

Figura 26 – Foto retirada do osciloscópio exibindo a rampa formada pelo amplificador operacional que varia de $-12V$ a $+12V$ , onde o eixo da ordenada está numa escala de $5V$ . . . . .	36
Figura 27 – Foto retirada do osciloscópio exibindo a rampa normalizada para que o valor da tensão ficasse entre de $0V$ e $5V$ , onde o eixo da ordenada está numa escala de $2V$ . . . . .	36
Figura 28 – Gráfico mostrando os sinais do <i>PZT</i> e fototransistor. As curvas de azul e vermelho são, respectivamente, os sinais de tensão no <i>PZT</i> e no fototransistor. . . . .	38
Figura 29 – Gráfico mostrando o sinal do <i>PZT</i> normalizado, representado pela curva azul, juntamente com a curva de ajuste indicada como laranja e o sinal lido pelo fototransistor identificada com verde. Os valores eixo da ordenada está normalizado de 1 à -1 e o eixo das abscissas corresponde o valor obtido do Arduíno . . . . .	38
Figura 30 – Imagem interferométrica <i>Ar/Ar</i> projetada no eixo dos tempos, onde o eixo da abscissa representa $t_1(s)$ que é o tempo de formação de um padrão de interferência e $t_2(s)$ , tempo necessário para a geração da imagem, no eixo da ordenada. . . . .	40
Figura 31 – Transformada de Fourier da imagem interferométrica <i>Ar/Ar</i> projetada no plano das frequências, onde os eixos das ordenadas e abscissas possuem unidade de frequência angular. Observe que há um ponto mais intenso na posição zero do eixo das ordenadas. Este ponto, representa o valor da delta discutido no capítulo 2. . . . .	41
Figura 32 – Módulo da transformada de Fourier sobre uma das deltas de Dirac do ao realizar o corte na transformada de Fourier da Imagem <i>Ar/Ar</i> . . . . .	42
Figura 33 – Frequências de $u$ e $v$ representadas, respectivamente, pelas curvas azul e vermelha, mostrando os valores das frequências angulares obtidos do espaço recíproco para <i>Ar/Ar</i> . . . . .	43
Figura 34 – Imagem interferométrica <i>Água/Água</i> projetada no eixo dos tempos: o eixo da abscissa representa $t_1(s)$ , que é o tempo de formação de um padrão de interferência. Já no eixo das ordenadas $t_2(s)$ é o tempo necessário para a geração da imagem. . . . .	44
Figura 35 – Transformada de Fourier da imagem interferométrica <i>Água/Água</i> projetada no plano das frequências, onde os eixos das ordenadas e abscissas possuem unidade de frequência angular. Observe que há dos pontos mais intensos na posição zero do eixo das ordenadas. Estes pontos representam os valores das deltas discutidos no capítulo 2. . . . .	44
Figura 36 – Representação da transformada de Fourier em 3D sobre uma das deltas de Dirac para a imagem interferométrica <i>Água/Água</i> . . . . .	45

Figura 37 – Frequências de $u$ e $v$ , representadas respectivamente pelas curvas azul e vermelha mostram valores das frequências angulares obtidos do espaço recíproco para $\acute{A}gua/\acute{A}gua$ . . . . .	46
Figura 38 – Imagem interferométrica $\acute{A}gua + NaCl/\acute{A}gua + NaCl$ projetada no eixo dos tempos, onde o eixo da abscissa representa $t1(s)$ que é o tempo de formação de um padrão de interferência e $t2(s)$ , tempo necessário para a geração da imagem, no eixo da ordenada. . . . .	47
Figura 39 – Frequências de $u$ e $v$ , representadas respectivamente pelas curvas azul e vermelha, mostram valores das frequências angulares obtidos do espaço recíproco para $\acute{A}gua + NaCl/\acute{A}gua + NaCl$ . . . . .	48
Figura 40 – Esquema do interferômetro de Milcheson, onde $l_1$ é o tamanho total do braço de controle, $l_2$ é o tamanho do braço de amostra e $lv$ representa a espessura do vidro fixado no portas amostras. . . . .	50
Figura 41 – Variação do índice de refração da imagem interferométrica para $Ar/Ar$ por $\Delta t$ , onde $1e^{-8}$ , representa $1 \cdot 10^{-8}$ . . . . .	52
Figura 42 – Variação do índice de refração da imagem interferométrica para $\acute{A}gua/\acute{A}gua$ por $\Delta t$ , onde $1e^{-8}$ , representa $1 \cdot 10^{-8}$ . . . . .	53
Figura 43 – Variação do índice de refração da imagem interferométrica para $Agua + NaCl/\acute{A}gua + NaCl$ onde $\Delta t$ possui unidade de tempo . . . . .	54

## SUMÁRIO

<b>1</b>	<b>Introdução . . . . .</b>	<b>12</b>
1.1	Objetivos . . . . .	12
1.2	Estrutura do Trabalho . . . . .	13
<b>2</b>	<b>Fundamentação Teórica . . . . .</b>	<b>14</b>
2.1	Interferência . . . . .	14
2.2	Interferômetros . . . . .	17
2.3	Modelo teórico para geração de uma imagem . . . . .	19
<b>3</b>	<b>Aparato Experimental . . . . .</b>	<b>27</b>
3.1	Montagem mecânica . . . . .	27
3.2	Eletrônica . . . . .	29
<b>4</b>	<b>Resultados . . . . .</b>	<b>37</b>
4.1	Geração de imagens interferométricas . . . . .	37
4.2	Obtenção da variação da diferença de fase . . . . .	40
4.3	Obtenção da variação do índice de refração . . . . .	50
<b>5</b>	<b>CONCLUSÕES . . . . .</b>	<b>55</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>57</b>
	<b>ANEXO A – Código para geração de um modelo de imagem interferométrica . . . . .</b>	<b>59</b>
	<b>ANEXO B – Código para automatização do aparato . . . . .</b>	<b>61</b>
B.1	Funcionamento mecânico do aparato . . . . .	61
B.2	Código para realizar as medias . . . . .	64
	<b>ANEXO C – Código para a coleta e tratamento de dados . . . . .</b>	<b>69</b>
C.1	Código principal . . . . .	69
C.2	Biblioteca criada . . . . .	71
	<b>ANEXO D – Circuito . . . . .</b>	<b>80</b>

## 1 Introdução

O estudo em tempo real da variação do índice de refração de organismos microscópicos surgiu para acompanhar o crescimento e controle de bactérias [4]. A partir desta motivação, começou o desenvolvimento do Interferômetro de Atividade Metabólica (MAI), um equipamento com o propósito de medir a mudança de fase para a determinação do índice de refração de uma dada amostra com o passar do tempo.

O desenvolvimento do MAI começou em 2003, sendo descrito inicialmente na referência [4]. Neste artigo, utilizou-se a técnica de interferometria para realizar o estudo de da taxa de crescimento de bactérias, analisando a variação do índice de refração do meio onde a mesma estava localizada. Ressaltando que na referência [4], todas as análises foram realizadas sobre apenas uma única amostra. Nas referências [5] e [6] identificam-se alterações realizadas no equipamento em relação à referência [4], a saber do controle térmico e posições das amostras. Estas alterações possibilitou o estudo de análise de vinte amostras consecutivas, as quais serão mostradas mais claramente no decorrer desta dissertação.

### 1.1 Objetivos

O objetivo desta dissertação é realizar alterações no MAI que está fora de operação desde 2009 [2]. Este equipamento já mostrou ser eficiente para estudar a evolução de taxa bacteriana [4], entretanto visamos ampliar o seu campo de estudo para outras substâncias como por exemplo: medidas de características físicas do plasma sanguíneo, taxa de crescimento de cianobactérias, cinética química e foto-resinas. Cada procedimento adotados nesta dissertação busca melhorar o equipamento utilizando diferentes ferramentas para o funcionamento do MAI, com o propósito de desenvolver um equipamento comercializável. Esta dissertação também tem como objetivo, propor um modo diferenciado de obter a diferença de fase, a qual se distingue das técnicas utilizadas nas referências, [4], [5], [6] e [2], que é a criação de uma imagem formada por padrões de interferência sequenciados no tempo, que será denominada de imagem interferométrica.

As modificações efetuadas na eletrônica do MAI se fez necessárias para utilizar *hardwares* recentes, com o propósito de obter um equipamento que opere por técnicas modernas e que proporcione que o aparato seja comandado por um ambiente de desenvolvimento integrado<sup>1</sup>. Visto que, a versão anterior do MAI utilizava um Controlador de Interface Periférica (*Peripheral Interface Controller-PIC*), que é uma ferramenta obsoleta atualmente, entretanto vislumbramos ferramentas que facilitariam o controle do MAI. Detalharemos estas modificações no decorrer da dissertação.

A principal motivação desta dissertação é a reconstrução do MAI e a implementação

---

<sup>1</sup> Ambiente de Desenvolvimento Integrado, ou IDE, do inglês Integrated Development Environment, é um um ambiente integrado para desenvolvimento de software [7]

das imagens interferométricas, ressaltando que não há no mercado, até a presente data, um equipamento que realiza os procedimentos dos quais o MAI já mostrou a ser capaz de executar.

## 1.2 Estrutura do Trabalho

Esta dissertação está estruturada em cinco capítulos.

A introdução é apresentada no capítulo 1. O capítulo 2 expõe de maneira breve a teoria necessária para entender as técnicas interferométricas e em seguida, uma breve introdução a respeito de interferômetros e um estudo sobre a criação de um modelo de uma imagem interferométrica. A montagem do MAI está retratada no capítulo 3, onde apresenta-se mostra a montagem mecânica do MAI juntamente com as alterações eletrônicas realizadas sobre o equipamento. As análises feitas estão descritas no capítulo 4, onde apresentamos como realizamos a construção de uma imagem interferométrica, juntamente com os resultados obtidos da variação da diferença de fase e variação do índice de refração, no capítulo 5 são apresentadas as conclusões, as considerações finais e as perspectivas futuras para trabalho.

## 2 Fundamentação Teórica

Neste capítulo será apresentado os conceitos básicos necessários para o bom entendimento da técnica utilizada nesta dissertação. Na primeira seção, apresenta-se os conceitos básicos de interferência. Também veremos de forma sucinta os tipos de interferômetros, especificadamente o Interferômetro de Michelson assim como os conceitos envolvidos no mesmo. Na sequência, apresenta-se o estudo da análise de Fourier para a elaboração de modelos de imagens interferométricas. Vale ressaltar que todo desenvolvimento deste trabalho foi executado no Laboratório de Física Aplicada da Universidade Federal de Juiz de Fora (UFJF).

A interferometria é uma técnica aplicável em muitos ramos da ciência. Sendo um processo físico muito utilizado para realizar medidas para obter informações sobre o índice de refração do material a ser analisado [8]. Uma aplicação atual desta técnica é o conhecido LIGO ( abreviação para *Laser Interferometer Gravitational-Wave Observatory - LIGO*), que possuem montagens de Interferômetros com braços de mais de 4 km [9].

### 2.1 Interferência

Para entender a técnica de interferometria, mencionada é necessário compreender o conceito de interferência. A interferência pode ser entendida como a superposição de duas, ou mais onda, em uma dada região. Considerando  $n$  campos elétricos  $\mathbf{E}_n$  e utilizando o principio de superposição, o campo total será dado por [10],

$$\mathbf{E} = \mathbf{E}_{(1)} + \mathbf{E}_{(2)} + \dots + \mathbf{E}_{(n)} \quad \text{ou} \quad \mathbf{E} = \sum_{i=1}^n \mathbf{E}_i \quad (2.1)$$

Como é bem conhecido, a irradiância de um campo elétrico está relacionada com o vetor de Poynting que é a potência por unidade de área [11]. Sua definição é

$$\mathbf{S} = \mathbf{E} \times \mathbf{H} \quad (2.2)$$

onde  $\mathbf{E}$  e  $\mathbf{H}$  são os campos elétricos e magnéticos da onda eletromagnética respectivamente. A média temporal do vetor de Poynting  $\mathbf{S}$  é dada por

$$\langle \mathbf{S} \rangle = \frac{1}{2} \text{Re}(\mathbf{E} \times \mathbf{H}^*) = \text{Re}\{\mathbf{S}^\dagger\}, \quad (2.3)$$

Agora, a fim de simplificar o problema, considera-se duas ondas planas oriundas de pontos diferentes do espaço, monocromática, de mesma frequência e em um meio homogêneo, Fig. 1, [1, 12].



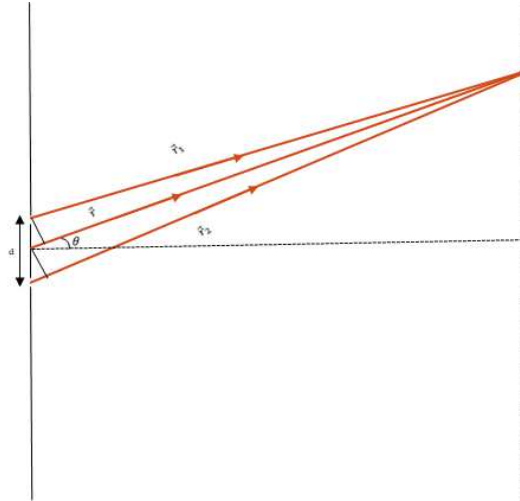


Figura 1 – Esquema de duas ondas eletromagnéticas percorrendo caminhos óticos diferentes até se interferirem

$$\mathbf{E}_{(1)} = \mathbf{E}_1 \exp i(\mathbf{k} \cdot \mathbf{r}_1 - \omega t + \phi_1) \quad (2.4)$$

$$\mathbf{E}_{(2)} = \mathbf{E}_2 \exp i(\mathbf{k} \cdot \mathbf{r}_2 - \omega t + \phi_2),$$

onde,  $\phi_1$  e  $\phi_2$  são as fases absolutas [12],  $\mathbf{k}$  representada o vetor de onda,  $\mathbf{r}_1$  e  $\mathbf{r}_2$  os vetores que define a posição,  $\omega$  a frequência angular da onda,  $t$  o tempo,  $\mathbf{E}_1$  e  $\mathbf{E}_2$  são as amplitudes da onda.

Considerando a diferença de fase como constante, pode-se calcular a irradiância utilizando a equação (2.3). Para calcular a intensidade [11], utiliza-se as duas ondas descritas na equação (2.4), ou seja,

$$\langle \mathbf{S}_1 \rangle = \frac{1}{2} \text{Re}\{\mathbf{E}_1 \times \mathbf{H}_1^*\}, \quad \langle \mathbf{S}_2 \rangle = \frac{1}{2} \text{Re}\{\mathbf{E}_2 \times \mathbf{H}_2^*\}.$$

Assim o vetor de Poynting total é dado por:

$$\langle \mathbf{S}_{tot} \rangle = \frac{1}{2} \text{Re}\{(\mathbf{E}_1 + \mathbf{E}_2) \times (\mathbf{H}_1^* + \mathbf{H}_2^*)\}. \quad (2.5)$$

Desenvolvendo esta conta, ficamos com

$$\langle \mathbf{S}_{tot} \rangle = \langle \mathbf{S}_1 \rangle + \langle \mathbf{S}_2 \rangle + \frac{1}{2} \text{Re}\{\mathbf{E}_1 \times \mathbf{H}_2^* + \mathbf{E}_2 \times \mathbf{H}_1^*\}, \quad (2.6)$$

onde o termo  $\frac{1}{2} \text{Re}\{\mathbf{E}_1 \times \mathbf{H}_2^* + \mathbf{E}_2 \times \mathbf{H}_1^*\}$  corresponde a interferência, isto é

$$\langle \mathbf{S}_{tot} \rangle = \langle \mathbf{S}_1 \rangle + \langle \mathbf{S}_2 \rangle + \langle \mathbf{S}_{int} \rangle. \quad (2.7)$$

Um cálculo mais preciso levaria em consideração a função de correlação, todavia nesta dissertação, não será necessário levar em consideração tal função pois estamos

utilizando um laser de He-Ne que possui um alto grau de coerência [11, 13]. Desta forma a equação (2.6) pode ser escrita como

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos\theta, \quad (2.8)$$

onde

$$\theta = [\mathbf{k} \cdot (|\mathbf{r} - \mathbf{r}_1| - |\mathbf{r} - \mathbf{r}_2|)] + \phi_1 - \phi_2. \quad (2.9)$$

é um termo de fase global.

## 2.2 Interferômetros

Os interferômetros são dispositivos ópticos que utilizam do conceito da interferência para determinar variações de índice de refração de um meio, ou até mesmo o índice de refração absoluto de alguma substância. Para gerar padrões de interferência existem dois tipos de montagens interferométricas que podem ser realizada. Interferômetros de divisão de frente de onda e interferômetros de divisão de amplitude[8].

No interferômetro de divisão de amplitude, a onda primária é dividida em dois feixes que percorrem caminhos diferentes antes de se rearranjar e interferir-se [12]. Exemplo de interferômetros de divisão de amplitude é o interferômetro de Fabry-Perot [1], cuja esquema pode ser visto na Fig 2.

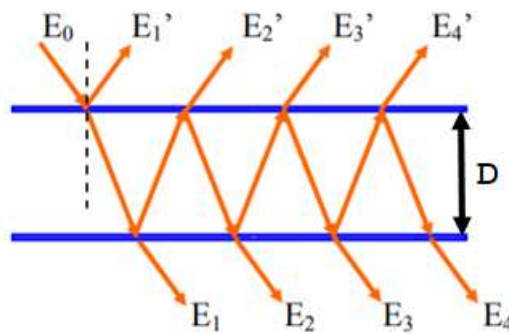


Figura 2 – Esquemática básica do interferômetro de Fabry-Perot, imagem da referência [1], onde  $\mathbf{E}_0$  é a onda incidente inicial  $\mathbf{E}_1$ ,  $\mathbf{E}_2$ ,  $\mathbf{E}_3$  e  $\mathbf{E}_4$  são as ondas transmitidas após a onda percorrer o caminho  $D$ ,  $\mathbf{E}'_1$  é a onda refletida ao incidir na placa superior e  $\mathbf{E}'_2$ ,  $\mathbf{E}'_3$  e  $\mathbf{E}'_4$  são as ondas transmitidas após sofrerem reflexão na placa inferior

Já o interferômetro de divisão de frente de onda utiliza segmentos da frente de onda primária que são usados como fontes para emitir ondas secundárias ou em conjunto com dispositivos ópticos para produzir fontes virtuais de ondas secundárias [12]. Como exemplo de interferômetro de divisão de frente de onda, podemos citar o interferômetro de Michelson, que é o interferômetro utilizado nesta dissertação.

A configuração básica do interferômetro de Milchelson, é composta por: uma fonte de luz, (S), dois espelhos planos, (M1 e M2), um divisor de feixe (*Beam Splitter* - *B.S.*), um compensador (*C*) e um detector onde se formará o padrão de interferência. Este padrão pode ser visualizado conforme a esquematização na Fig. 3. Seu funcionamento é dado da seguinte maneira: A luz originada pela fonte S se divide ao incidir no *B.S.* Os feixes percorreram caminhos diferentes,  $B_1$  e  $B_2$ , até serem refletidos pelos espelhos planos M1 e M2. Após esse processo, os feixes incidem novamente no *B.S.*, onde os feixes serão superpostos novamente e podendo assim observar o padrão de interferência no sensor. Note que na primeira incidência, o feixe que percorreu o caminho  $B_2$  sofreu uma reflexão e o que passou pelo caminho  $B_1$  foi por meio do processo de transmissão. Na segunda incidência o

processo se inverte, o feixe que percorreu o caminho  $B_2$  sofreu uma transmissão e o que passou pelo caminho  $B_1$  foi refletido. O compensador nesse esquema, tem a finalidade de igualar o caminho que os feixes  $B_1$  e  $B_2$  percorram.

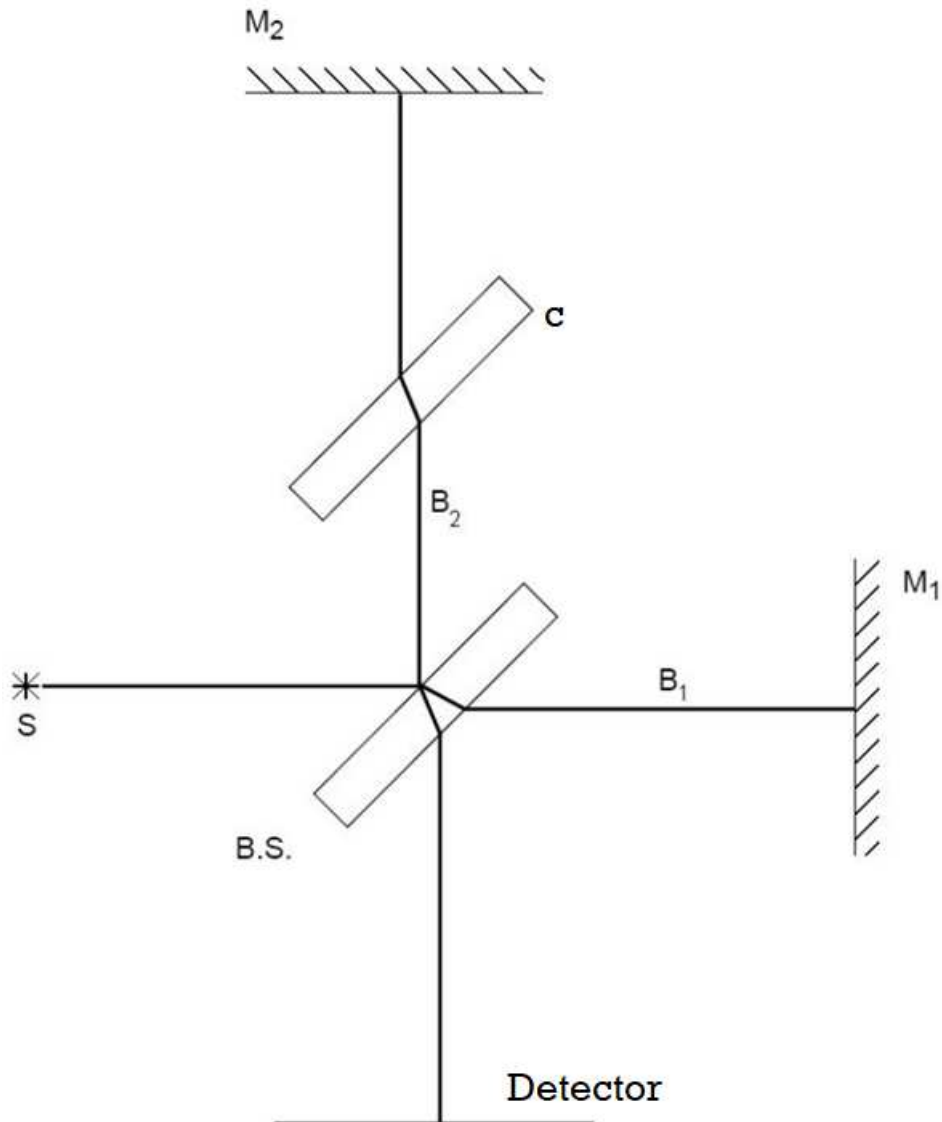


Figura 3 – Arranjo básico de um interferômetro de Michelson, imagem retirada [2]: S é a fonte que emite o feixe laser, B.S é o divisor de feixe e  $B_1$  e  $B_2$  os caminhos percorridos pelo feixe após passar pelo B.S

### 2.3 Modelo teórico para geração de uma imagem

As atividades biológicas podem ser monitoradas a partir da variação de fase ou com a mudança do índice de refração que está relacionado ao metabolismo do organismo. Para alcançar esse objetivo utilizando interferometria, uma quantidade razoável de padrões interferométricos deveria ser medido ao longo do tempo. Posteriormente, ajustes de curvas seriam realizados para tomar a variação de fase que ocorre devido à mudança de índice de refração no meio onde a amostra biológica se encontra.

Neste ponto, o trabalho de ajuste de curvas pode ser inviável em termos computacionais, pois podem levar várias horas para realizar as análises dos dados. Para contornar esse problema, propomos uma nova abordagem que evita a necessidade de fazer ajustes de curvas continuamente sobre padrões interferométricos. Com o intuito de acelerar o processo computacional, geramos uma imagem interferométrica e utilizamos o conceito de transformada de Fourier, [14, 15] para realizar o estudo sobre a imagem.

A Transformada de Fourier de uma função  $f(t)$  é definida da seguinte maneira:

$$g(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) \exp[i\omega t] dt, \quad (2.10)$$

e sua transformada inversa é dado por,

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(\omega) \exp[-i\omega t] d\omega. \quad (2.11)$$

A transformada de Fourier discreta transforma uma sequência de  $N$  números  $\{\mathbf{x}_n\} = x_0, x_1, \dots, x_{N-1}$  em uma nova sequência de números discretos,  $\{\mathbf{X}_k\} = X_0, X_1, \dots, X_{N-1}$  [16], como podemos observar na equação (2.12)::

$$X_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi}{N} kn}, \quad (2.12)$$

e sua transformada inversa é dada por,

$$x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{\frac{i2\pi}{N} kn} \quad (2.13)$$

Também é possível avaliar a equação (2.12) fora do domínio de  $k \in [0, N - 1]$ , isto é, estendendo o período  $N$ . Dessa maneira outras sequências de  $N$  são utilizadas, como por exemplo,  $[-\frac{N}{2}, \frac{N}{2} - 1]$  para  $N$  par, e  $[-\frac{N-1}{2}, \frac{N-1}{2}]$  para  $N$  ímpar [17].

A equação (2.12), pode ser assimilada de algumas maneiras, como os exemplos abaixo [18].:

- Descrever completamente a transformada de Fourier em um tempo discreto (DTFT), em uma sequência periódica que representa apenas componentes discretas da frequência.

- Fornecer esquemas uniformemente espaçados da DTFT contínua em uma sequência de comprimento finito. E, também, é um análogo discreto da fórmula para os coeficientes de uma série de Fourier

A transformada inversa de Fourier equação (2.13) também é periódica em  $N$  no domínio de  $n \in [0, N - 1]$ . Onde todo  $X_k$  é um número complexo que agrupa a fase e a amplitude da componente senoidal ( $e^{i2\pi kn/N}$ ) da função  $x_n$ . A frequência senoidal possui  $k$  ciclos por  $N$  modelos [16]. Sua amplitude e fase são

$$|X_k| = \sqrt{\text{Re}(X_k)^2 + \text{Im}(X_k)^2} \quad (2.14)$$

$$\arg(X_k) = \text{atan2}(\text{Im}(X_k)) \quad (2.15)$$

Nesta dissertação utilizaremos um algoritmo conhecido como transformada discreta de Fourier rápida (FFT) [19] na qual utilizamos a linguagem de programação Python juntamente com o pacote Scipy [20, 21].

Para exemplificar, considere a seguinte equação (2.16) para gerar um modelo de imagens interferométricas,

$$I(t_1, t_2) = \cos[2\pi(at_1 + bt_2)], \quad (2.16)$$

onde  $a$  e  $b$  são constantes que podem assumir quaisquer valores e  $t_1$  e  $t_2$  são variáveis temporais. Para visualizar a equação (2.16) discretizaremos os valores  $t_1 = n\Delta t$  e  $t_2 = m\Delta t$ , com  $\Delta t = 10.0ms$ . Escolhendo os valores  $a = 0.083s^{-1}$ ,  $b = 0s^{-1}$  e os valores máximos para  $n_{max} = N_1$  e  $m_{max} = N_2$  com  $N_1 = 69$  e  $N_2 = 200$  na equação (2.16). Criamos desta forma uma imagem baseada no modelo da equação (2.16) mostrada na Fig. 4.

Uma vez que estamos utilizando imagens para tratar nossos dados as transformadas de Fourier definidas pelas equações (2.12) e (2.13) podem ser escritas da seguinte maneira [22],

$$X_{u,v} = \frac{1}{\sqrt{N_1}} \frac{1}{\sqrt{N_2}} \sum_{n=0}^{N_1-1} \sum_{m=0}^{N_2-1} x_{n,m} e^{-i2\pi(\frac{un}{N_1} + \frac{vm}{N_2})} \quad (2.17)$$

onde  $u$  e  $v$  são os valores das frequências da equação no espaço recíproco. Sua transformada inversa é dado por,

$$x_{n,m} = \frac{1}{\sqrt{N_1}} \frac{1}{\sqrt{N_2}} \sum_{n=0}^{N_1-1} \sum_{m=0}^{N_2-1} X_{u,v} e^{i2\pi(\frac{un}{N_1} + \frac{vm}{N_2})} \quad (2.18)$$

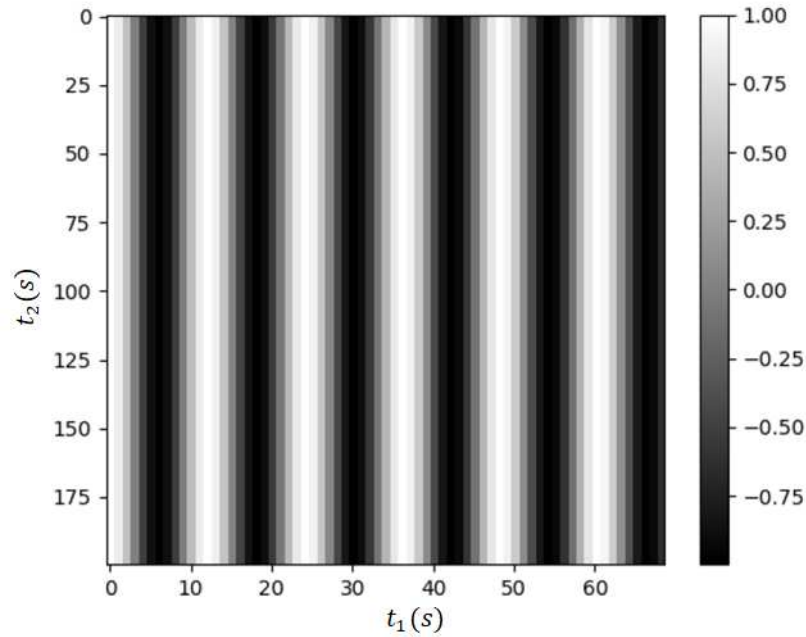


Figura 4 – Projeção de uma imagem interferométrica, onde  $t_1$  e  $t_2$  variáveis temporais e adotando valores de  $a = 0.083s^{-1}$ ,  $b = 0s^{-1}$ .

Ao calcular a transformada de Fourier em duas dimensões [23] ( $t_1$  e  $t_2$ ) da equação (2.16) obtemos duas deltas de Dirac, uma localizada em  $\pm a$  e outra localizada em  $\pm b$ . Como pode ser visto nas imagens representas na fig. ?? do modulo da transformada de Fourier que é definida na equação (5).

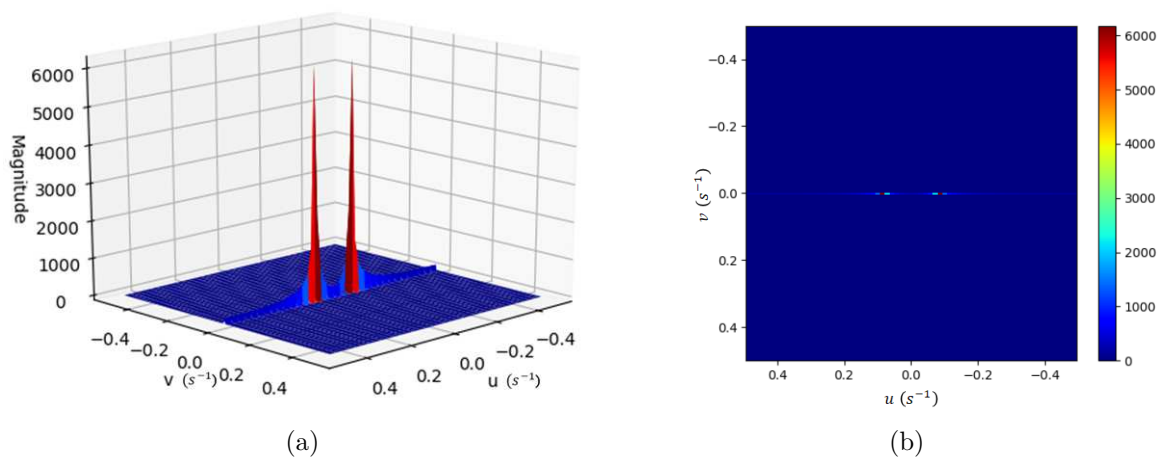


Figura 5 – Módulo da transformada de Fourier de uma imagem interferométrica adotando  $a = 0.083s^{-1}$  e  $b = 0s^{-1}$ . No qual (a) e (b) representa sua visualização em 3D e sua projeção em  $u$  e  $v$  respectivamente.

Isto é, quando  $b$  for igual a zero não há inclinação nos padrões. Agora, adotando um valor diferente de zero, como por exemplo  $b = -0.02s^{-1}$ , encontramos a imagem

interferométrica mostrada na fig. 6. Observe a inclinação entre as franjas claras e escuras, como já esperado.

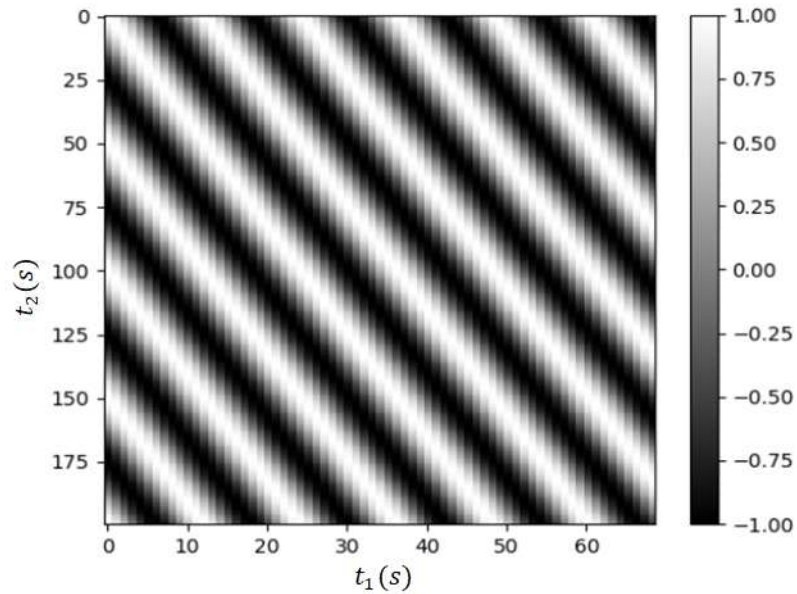


Figura 6 – Projção de uma imagem interferométrica, onde  $t_1$  e  $t_2$  variáveis temporais e adotando valores de  $a = 0.083s^{-1}$ ,  $b = -0.02s^{-1}$ .

Observe que a Fig. 6, não apresentam mais padrões claros e escuros verticais. Agora há declínio nos padrões, uma vez que  $b \neq 0$ . Retirando a transformada de Fourier da Fig. 6, obtemos a Fig. 7

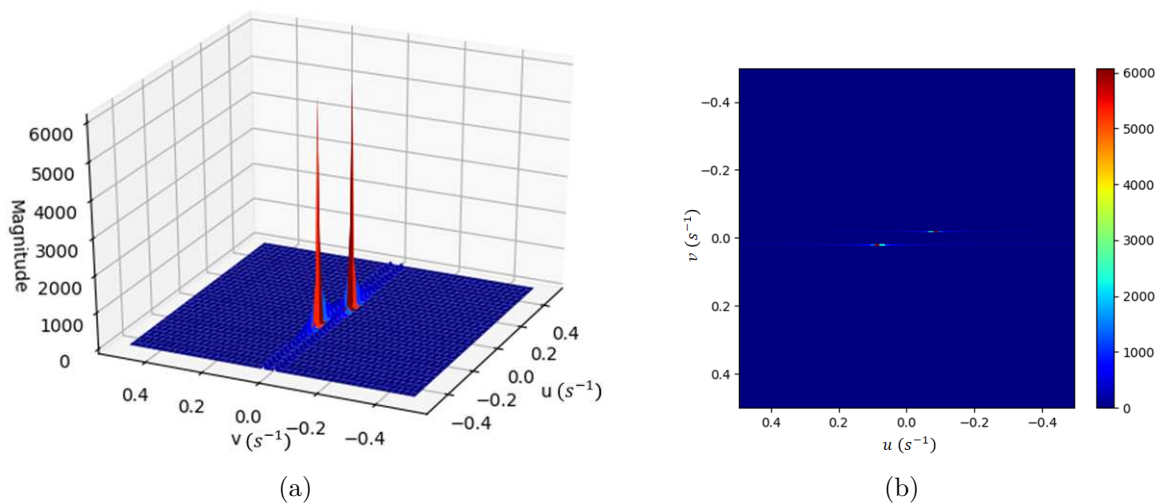


Figura 7 – Módulo da transformada de Fourier de uma imagem interferométrica adotando  $a = 0.083s^{-1}$  e  $b = -0.02s^{-1}$ . No qual (a) e (b) representa sua visualização em 3D e sua projeção em  $u$  e  $v$  respectivamente.

Observe que um acréscimo no valor de  $b$ , no argumento da função cosseno, nos



resulta uma diferença de fase e como estamos analisando de diferença de fase no passar do tempo, obtemos a variação da diferença de fase. Entretanto a transformada de Fourier de uma função cosseno deveria nos retornar duas deltas de Dirac [15], que seriam dois pontos bem definidos na nossa imagem, o que não é observado a partir das Figs. 5 e 7. A discordância do nosso valor ocorre devido efeito de Leakage [24]. Pois ao aplicar a transformada de Fourier é necessário extrair uma parte do sinal de comprimento finito, produzindo assim estimativas incorretas de amplitude e frequência do sinal, devido a esse "vazamento" no sinal deu-se o nome efeito de Leakage, veja mais detalhes na Fig. (8) .

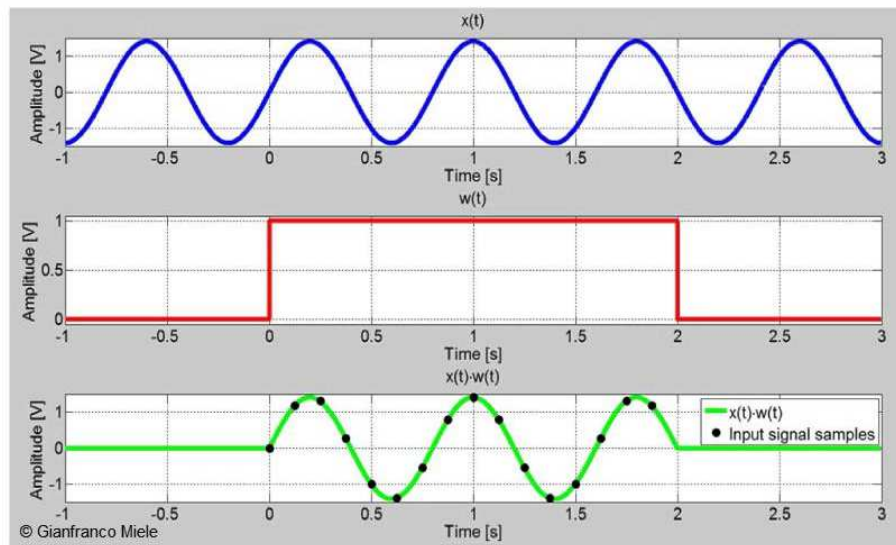


Figura 8 – Efeito de Leakage, onde a curva de azul mostra um sinal co-senoidal, a curva de vermelho é a janela introduzida do sinal co-senoidal inicial e a curva de verde representa a discretização obtida com a introdução da janela. Imagem modificada da referência [3]

Uma maneira de reduzir o efeito Leakage é aplicando uma janela em nosso sinal. Existem uma variedade de janelas para reduzir as dispersões devido o o efeito Leakage [25], nesta dissertação escolhemos utilizar a janela denominada de *Hamming* [21, 24], pois esta janela é útil quando se deseja uma redução da dispersão em relação a frequência, não se importando com a dispersão na amplitude.

Ao aplicar a janela *Hamming* nas transformadas de Fourier obtemos a seguinte redução nas nossas dispersões, que pode ser observada na Fig 9, para a transformada de Fourier da imagem interferométrica adotando  $a = 0.083s^{-1}$  e  $b = 0s^{-1}$ .

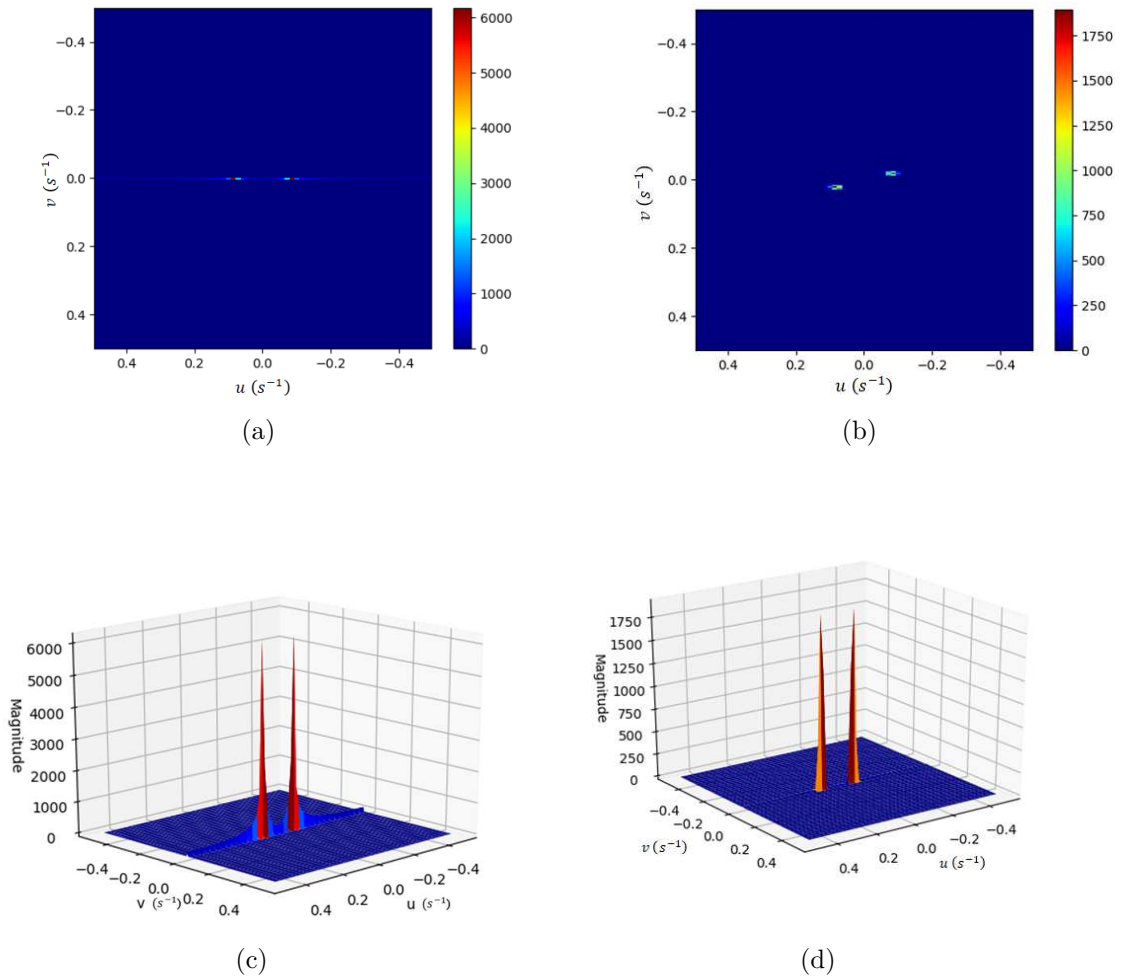


Figura 9 – Módulo da transformada de Fourier de uma imagem interferométrica adotando  $a = 0.083s^{-1}$  e  $b = 0s^{-1}$ . No qual (a) representa a transformada de Fourier sem utilização da janela *Hamming*, (b) a transformada de Fourier utilizando a janela *Hamming*, (c) representação da transformada de Fourier em 3D sem utilização da janela *Hamming* e (d) representação da transformada de Fourier em 3D utilizando a janela *Hamming*.

Atente que a dispersão não foi completamente eliminada utilizando a janela *Hamming*, entretanto houve uma redução perceptível da mesma. Realizando o mesmo processo para imagem interferométrica, no qual os parâmetros da equação 2.16 escolhidos, foram  $a = 0.083s^{-1}$  e  $b = -0.02s^{-1}$ , Fig. 10, percebemos também apenas uma redução na dispersão, isto é, não se pode eliminar totalmente o efeito de Leakage.

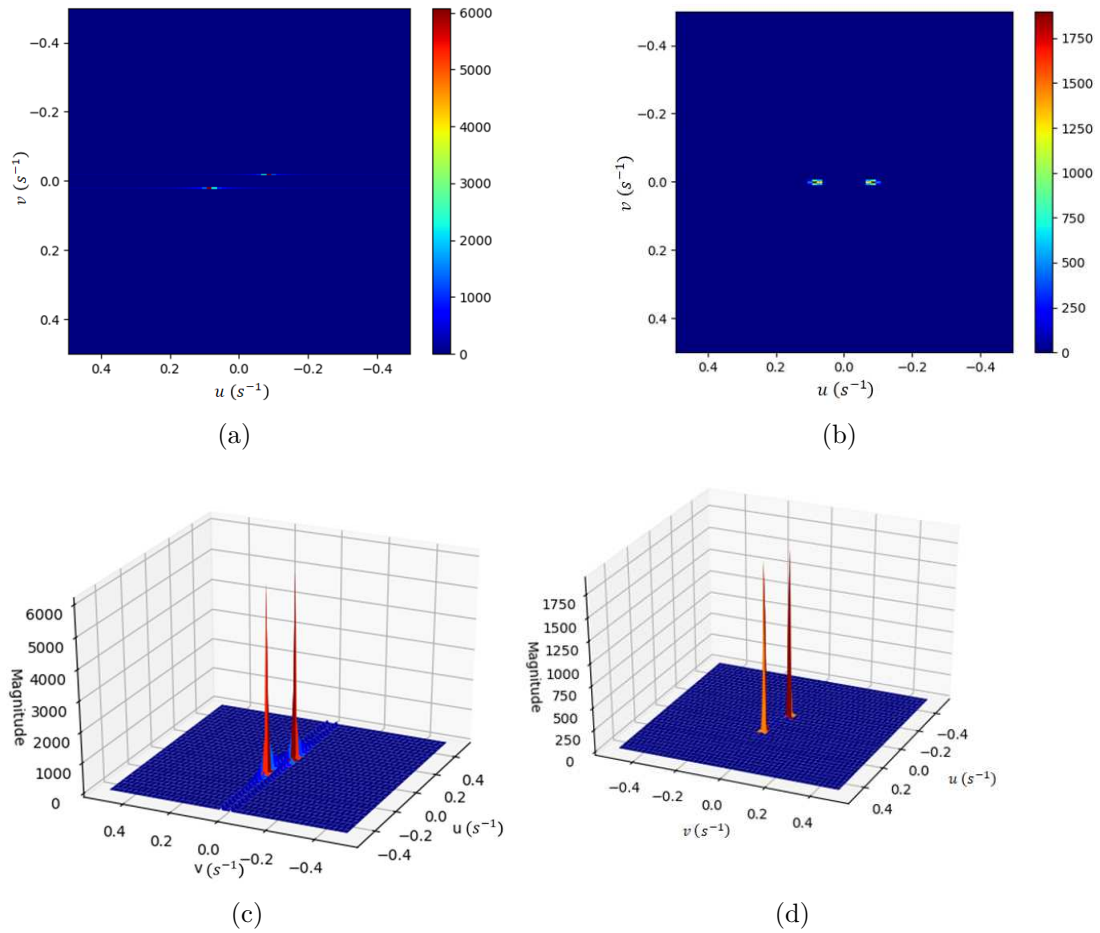


Figura 10 – Módulo da transformada de Fourier de uma imagem interferométrica adotando  $a = 0.083s^{-1}$  e  $b = -0.02s^{-1}$ . No qual (a) representa a transformada de Fourier sem utilização da janela *Hamming*, (b) a transformada de Fourier utilizando a janela *Hamming*, (c) representação da transformada de Fourier em 3D sem utilização da janela *Hamming* e (d) representação da transformada de Fourier em 3D utilizando a janela *Hamming*.

Uma outra maneira de reduzir os efeitos de dispersão é em relação aonde iríamos truncar a série quando realizarmos a transformada de Fourier. Considerando que  $t_1$  e  $t_2$  possa variar de 0 a 1000, isto é, realizamos a transformada de Fourier numericamente onde o somatório será realizada de 0 a 1000. Veja que na Fig. 11 o resultado encontrado são dois pontos, entretanto com pequenas.

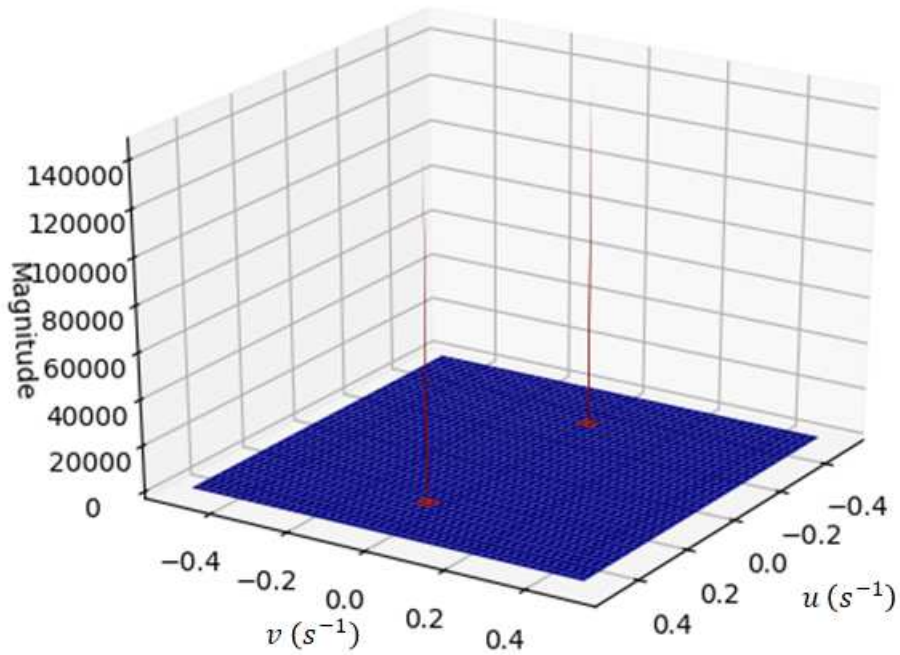


Figura 11 – Transformada de Fourier em aumentando o range dos valores de  $t_1$  e  $t_2$  na equação  $I(t_1, t_2) = \cos[2\pi(at_1 + bt_2)]$ , onde adotamos  $b = -0,02s^{-1}$

Estamos chamando atenção para este fato pois, no capítulo 4, vamos mostrar que os dados experimentais possuem a mesma ordem de grandeza das retratadas nas figuras acima e conseqüentemente obtemos uma dispersão nos nossos resultados experimentais. Destacando que as figs. 4 e 6, também foram geradas para obtermos imagens interferométricas semelhantes as imagens experimentais.

### 3 Aparato Experimental

Neste capítulo iremos expor o funcionamento de nosso equipamento, mostrando a montagem mecânica e, em seguida explicando as alterações realizadas nele, o que levou a construção de uma eletrônica com uma nova lógica de automação e um método de processar imagens.

#### 3.1 Montagem mecânica

Conforme mencionado anteriormente o MAI é um dispositivo baseado no interferômetro de Michelson e sua estrutura pode ser observada na fig. 12.

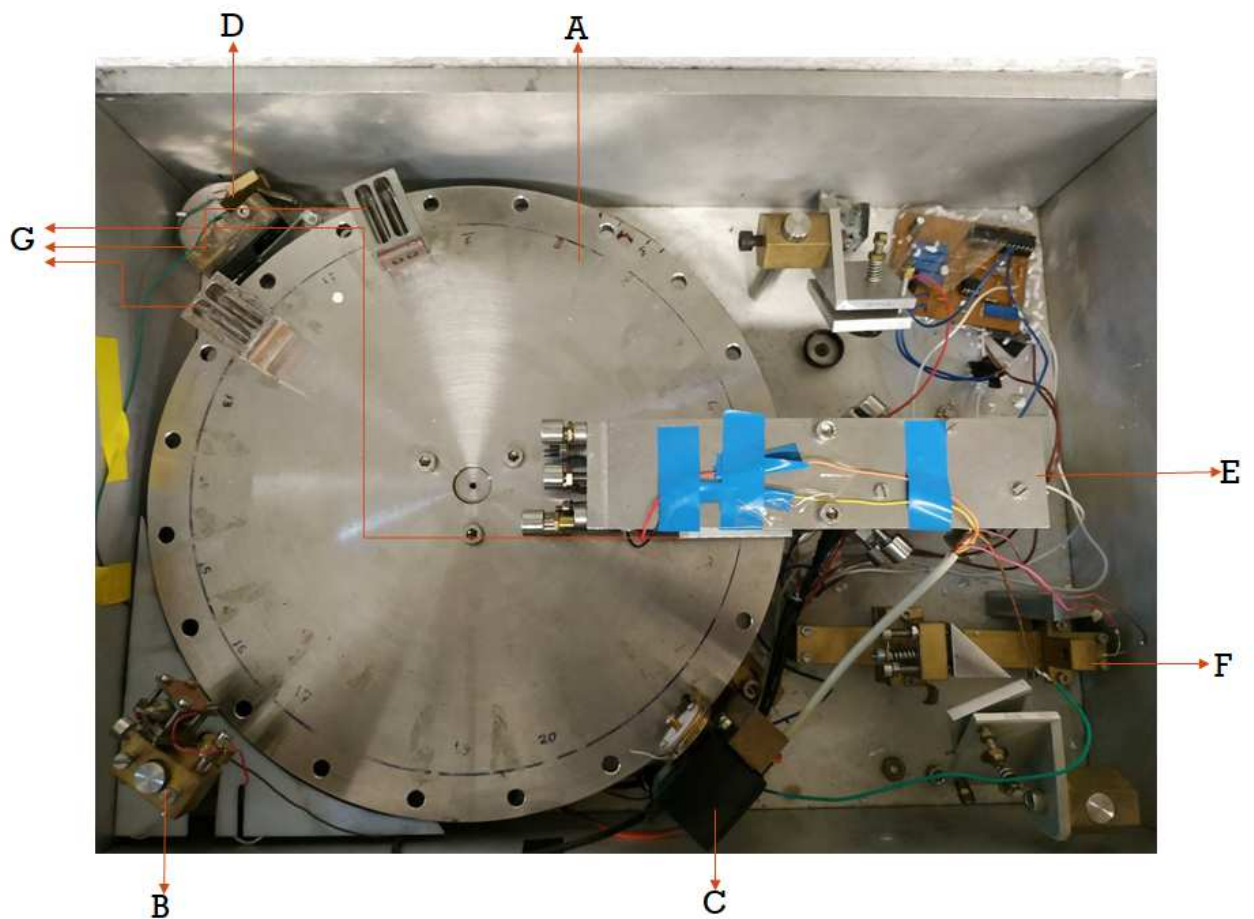


Figura 12 – Foto do MAI, onde (A) Disco de aço inoxidável, utilizado para sustentar os portas amostras; (B) conjunto laser com fotorresistor; (C) o servo motor; (D) contador de voltas; (E) interferômetro de Michelson, cujo aparatos ópticos estão abaixo da placa de aço inoxidável; (F) Fototransistor e (G) dois porta amostras, frisando que os porta amostras passam por baixo do sistema interferométrico descrito por (E).

A construção do MAI foi realizada em 2003 [4]. Para deixar o equipamento operacional novamente, foi necessário realizarmos um estudo com o propósito de compreender

a lógica de funcionamento do equipamento [2]. Seu funcionamento é compreendido de maneira mais clara com base no esquema apresentado na fig. 13.

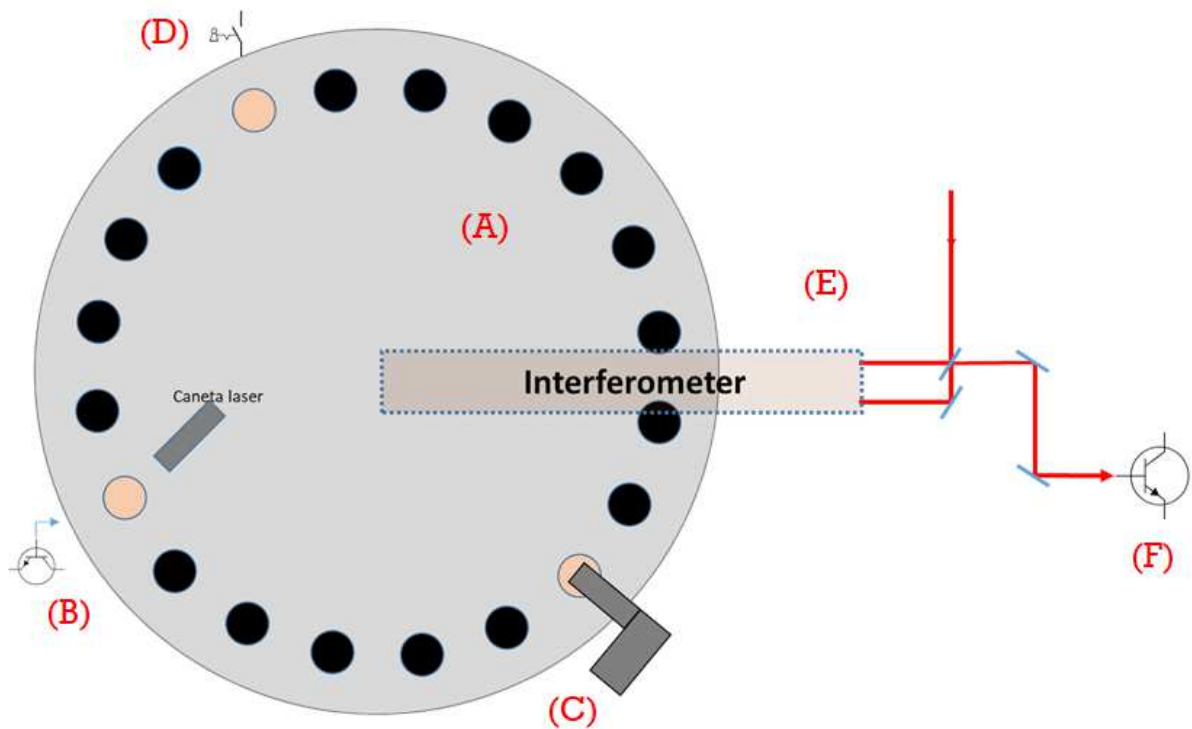


Figura 13 – Representação esquemática do MAI, onde onde (A) é Disco de aço inoxidável, utilizado para sustentar os portas amostras; (B) representa o conjunto laser com fotorresistor; (C) o servo motor; (D) contador de voltas; (E) interferômetro de Michelson, cuja os aparatos ópticos estão abaixo da placa de aço inoxidável e (F) o Fototransistor

Segundo a estrutura demonstrada na fig. 13, podemos subdividir o MAI em seis subsistemas, com o intuito de compreender com mais perfeição como deveria ser a lógica de funcionamento do equipamento, o subsistemas dispõem da seguinte lógica: Primeiramente o disco de aço inoxidável, que dispõe de vinte furos igualmente espaçados, irá girar em uma velocidade contante. Após isso, o fotorresistor do conjunto (B) da fig. 13 (ressaltando que o laser permanece em operação há todo momento), fica a todo momento averiguando um sinal do laser, que só será detectado quando o laser passar por algum furo. No momento que o sinal for detectado o servo motor realizará o travamento do disco, de modo que o porta amostra fique posicionado abaixo do interferômetro. O fototransistor possui a finalidade de identificar os sinais de interferência. O item (D) é um contador de voltas, além de contar quantos ciclos o disco realizou, também permite que tenhamos um controle da primeira a amostra a ser analisada.

Compreendido como o MAI operava, uma nova eletrônica foi elaborada com o objetivo de retomar a atividade do equipamento, este assunto será tratado na seção 3.2, onde também deixaremos claro o funcionamento dos subsistemas.

### 3.2 Eletrônica

A eletrônica dos conjuntos citados na seção 3.1 do MAI, estavam dispostas por um microcontrolador PIC-18f4520 [2]. Entretanto, esse microcontrolador não possui uma interface direta com o computador, diante disso decidimos utilizar o Arduino, fig. 14, que é uma placa de desenvolvimento de biblioteca aberta e possui uma interação direta com o computador através de um Ambiente de Desenvolvimento Integrado (*Integrated Development Environment -IDE*), com linguagem de programação baseada em C e C++ [26]

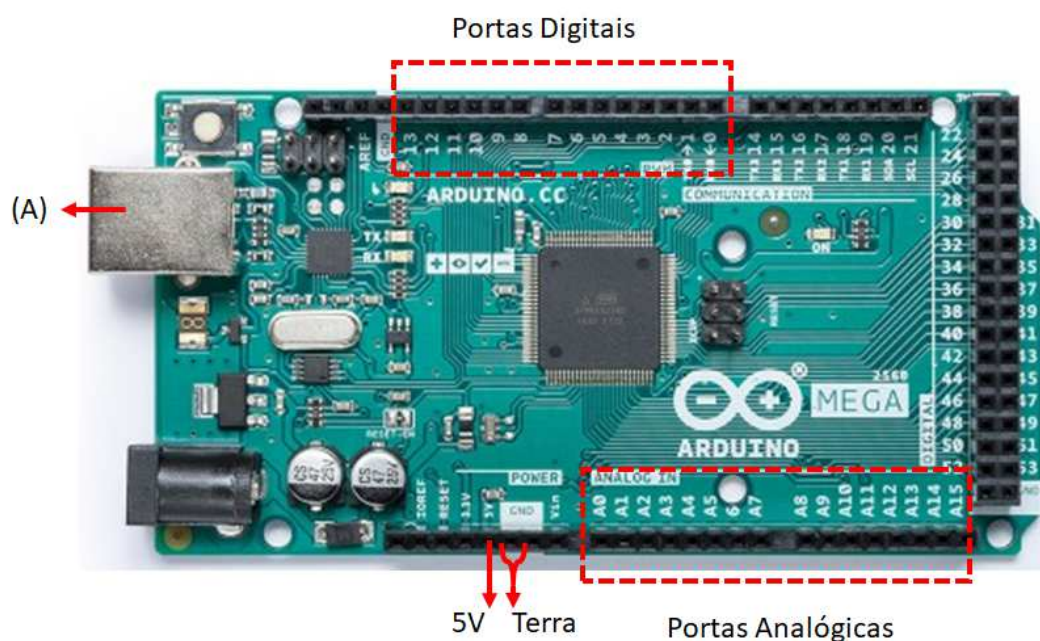


Figura 14 – Representação de um Arduino Mega 2560, exibindo as portas digitais e analógicas e (A), que é o local para o cabeamento para realizar a comunicação direta com o computador.

A fig. 14 apresenta um Arduino Mega 2560, modelo utilizado para realizar esta dissertação. O próprio Arduino dispõem de uma porta que fornece +5V, portas digitais que propicia valores de tensão de 0V ou +5V, portas analógicas que realizam a leituras de sinais em bytes e (A) que representa onde será realizado o cabeamento para realizar a comunicação direta com o computador.

Com a utilização do Arduino, uma nova eletrônica e um novo programa para a automatização do dos sistemas da seção 3.1 necessitariam ser realizados. Devida o uso do Arduino realizamos os circuitos dos subsistemas em blocos, testante cada um deles individualmente.

Começamos com a reconstrução da eletrônica a partir do disco, cuja o seu movimento é realizado por um motor de passo acoplados a engrenagens, conforme pode ser visualizado na fig. 15,



Figura 15 – Foto retirada do motor de passo, onde (A) mostra a posição do motor de passo e (B) representa as engrenagens responsável pelo giro do disco.

Para o controle do motor de passo utilizamos um circuito integrado, ULN 2003A, que é altamente utilizado para realizar o controle de motores de passos. O Arduino controla os pulsos de comando enviados aos terminais do motor, desta forma produzindo um movimento do motor e conseqüentemente causando uma rotação no disco. O circuito esquemático deste subsistema está representado na fig. 16

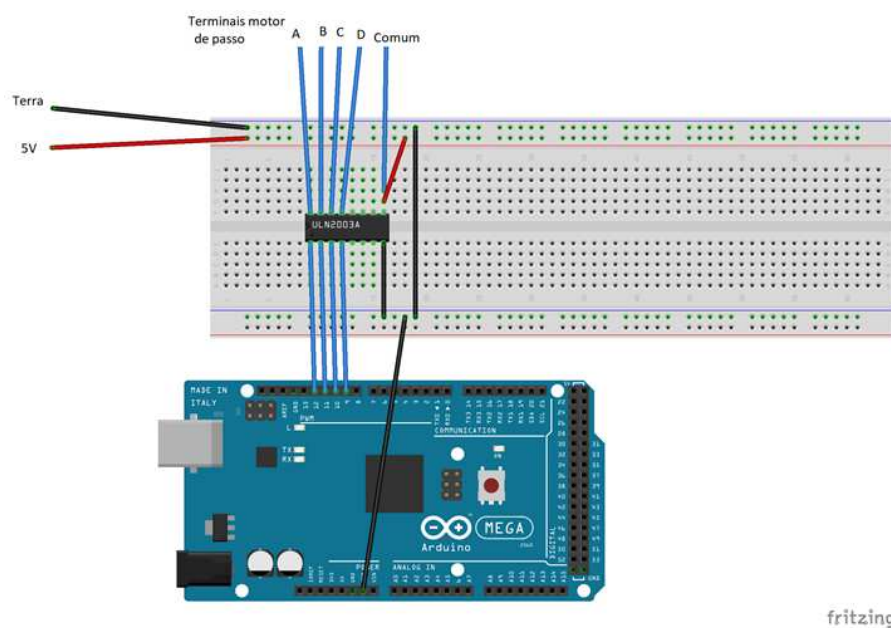


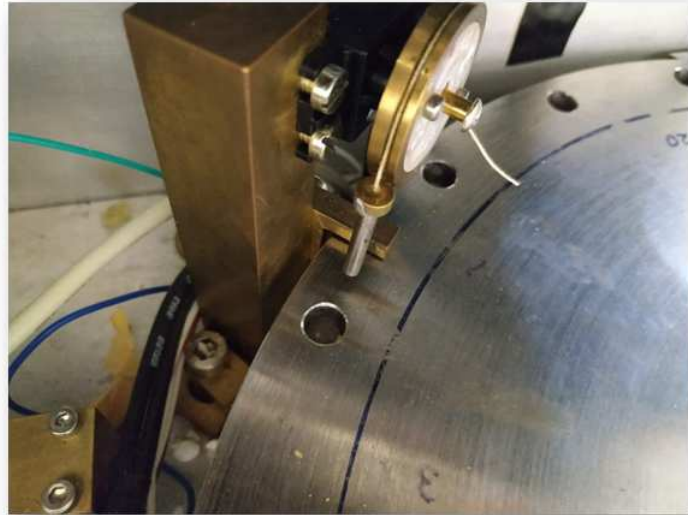
Figura 16 – Esquema ilustrativo da eletrônica responsável pela rotação do disco juntamente com o Arduino.



Na sequência realizamos a eletrônica dos circuitos do conjunto laser com LDR juntamente com o servo motor, tais dispositivos estão representados na fig. 17



(a) Conjunto laser com LDR



(b) Servo motor

Figura 17 – Foto retirada do conjunto laser com LDR e do servo motor, mecanismos responsáveis de realizar o travamento do disco .

Na fig. 17(a) podemos reparar o laser na parte de cima do disco como já mencionado na seção 3.1. Quando o feixe laser incidir em algum orifício o LDR que está disposto abaixo do disco detectará o sinal do feixe laser, retornando um sinal para o Arduino, que retorna um sinal para o servo motor, que possui uma pino acoplado em si, acionando-o e realizando o travamento do disco, para detalhes veja na fig. 17(b). A esquematização eletrônica destes subsistemas estão apresentados na fig. 18.

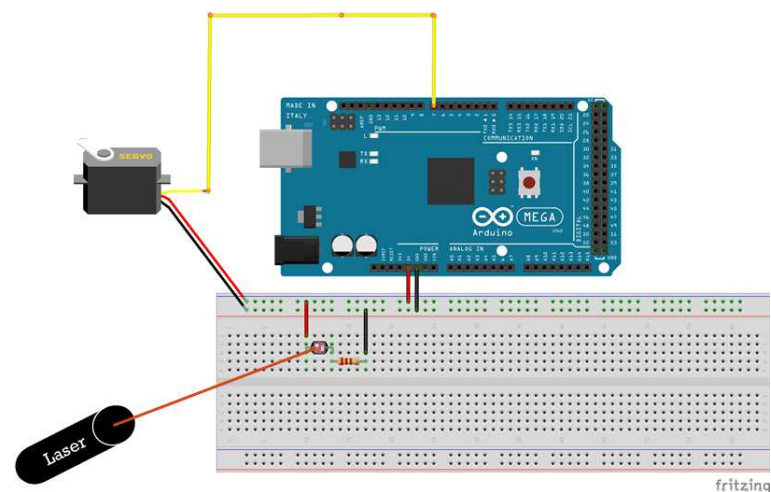


Figura 18 – Esquema ilustrativo da eletrônica responsável pelo controle do conjunto laser com LDR juntamente com o servo motor e com o Arduino.

Na sequência realizamos a elaboração da eletrônica do contador de voltas, fig. 19. Esta ferramenta além de contar quantas voltas o disco realiza, também proporciona que tenhamos o controle da primeira amostra a ser medida.



Figura 19 – Foto retirada do contador de voltas.

O contador de voltas é uma chave que é assinada ao passar por um ressalto localizado em um ponto específico do disco, fig. 20. Ao passar pelo ressalto um sinal é enviado para o Arduino comunicando-o que o disco completou uma volta completa. Este sinal lido pelo Arduino também pode ser entendido como um sinal que a próxima amostra que passar pelo interferômetro é a primeira amostra. Assim, teremos um controle de sempre iniciar as aquisições de dados a partir da primeira amostra.

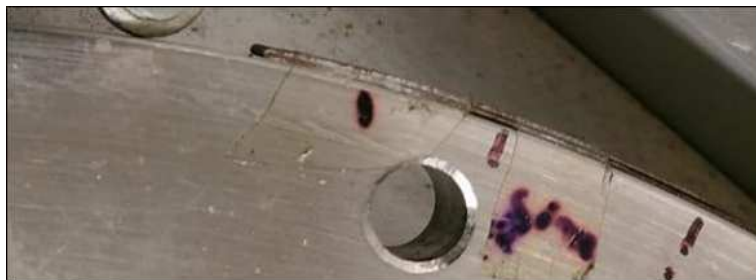


Figura 20 – Zoom realizado da figura 19, com o proposito de observar o ressalto posto no disco

Com os circuitos da parte opto mecânica, responsável por girar e travar o disco, de maneira que os recipientes do porta amostras fique alinhado aos braços do interferômetro, elaboramos um código via Arduino para que esses subsistemas funcionassem como um único sistema. A lógica deste código está representado no fluxograma da fig. 21

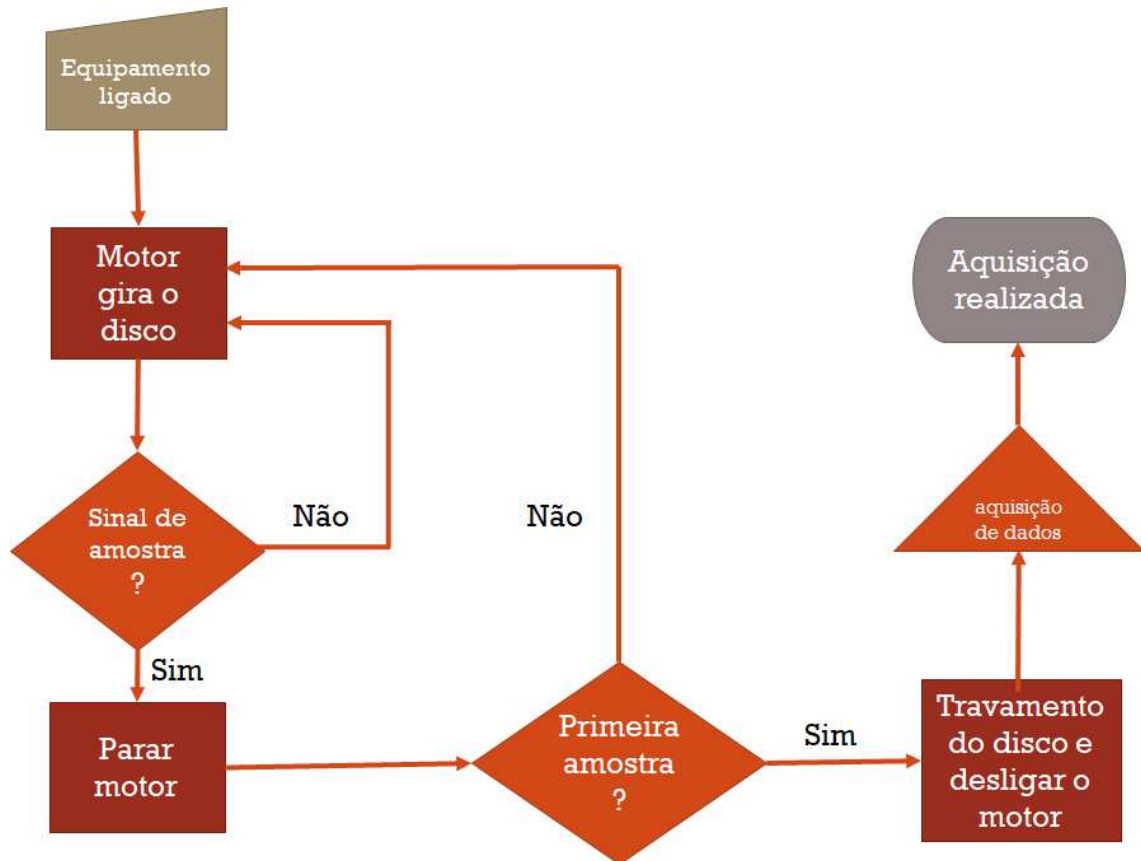


Figura 21 – Fluxograma da lógica do código responsável pelo funcionamento da parte mecânica.

Com o controle das componentes mecânicas realizadas, seguimos para a construção do circuito e da programação do conjunto responsável por realizar a aquisição de dados. Isto é, realizamos uma programação para o controle mecânico, que é apresentado no fluxograma da fig. 21 e uma programação para o controle da aquisição de dados. A explicação da divisão em dois programas é porque, uma vez que, os dois conjuntos estejam funcionando, podemos colocar ambos sistemas funcionando juntamente, realizando um novo código no Arduino. E também porque para gerar as imagens interferométricas e realizar o estudo das mesmas, não é necessário que os dois conjuntos funcionando paralelamente.

O subsistema responsável de realizar as aquisições de dados está representado na fig. 22. Observe que o porta amostras está disposto de maneira que os feixes laser passem pelos seus recipientes. No braço inferior do interferômetro apresentado na fig. 22, está acoplado um piezoelétrico (PZT), que é responsável de realizar uma alteração no caminho óptico, fazendo assim que os nossos padrões de interferência se desloquem de forma constante. O deslocamentos destes padrões nos fornecem uma fase constante. Caso ocorra alguma alteração no recipiente onde se encontra amostra, o deslocamento dos meus padrões de interferência não retornaram ao mesmo ponto, assim obteremos uma diferença de fase. Os padrões de interferência são obtidos a partir do fototransistor.

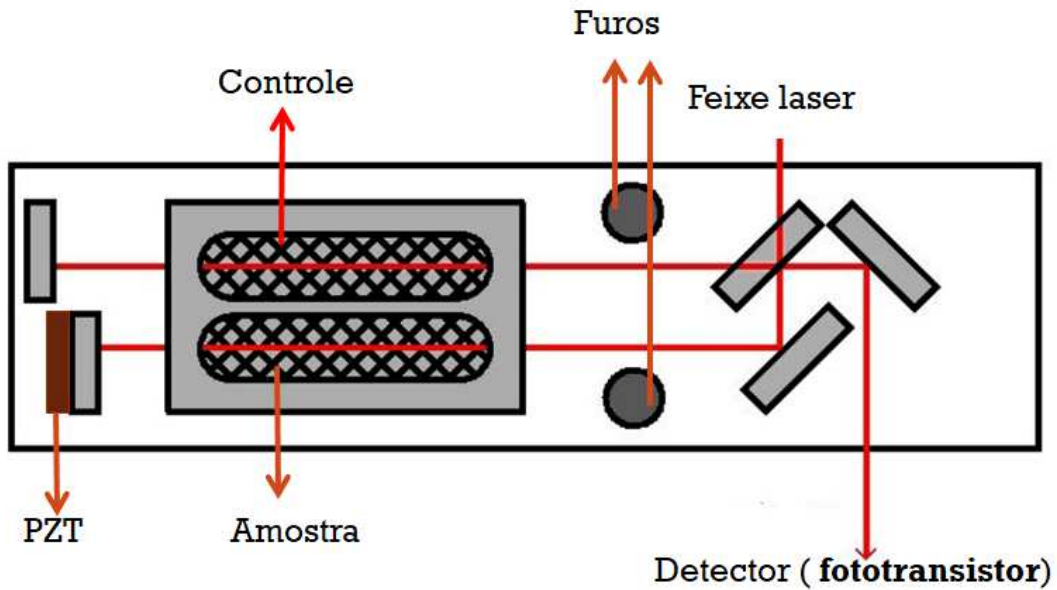


Figura 22 – Fluxograma da lógica do código responsável pelo funcionamento da parte mecânica.

Para causar uma diferença no caminho óptico, aplicamos uma tensão que variava de  $-12V$  a  $+12V$  no PZT. Para progredir essa tensão de  $-12V$  a  $+12V$  elaboramos uma eletrônica baseada em amplificadores operacionais inversores, potenciômetros e capacitor, como pode ser observado no esquema da fig. 23.

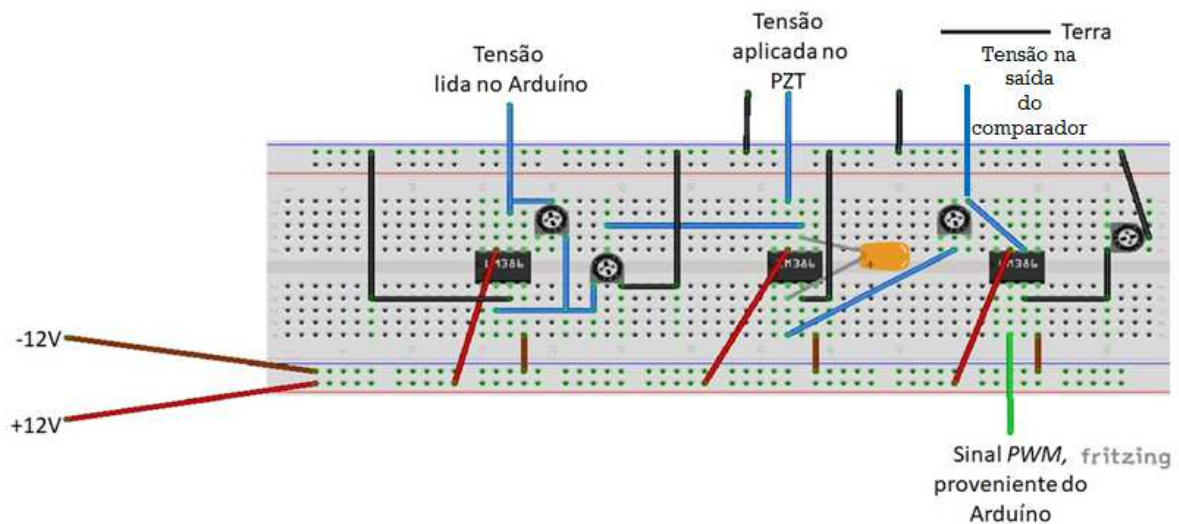


Figura 23 – Esquema ilustrativo da eletrônica responsável pelo controle do conjunto responsável pela aquisição de dados.

Para o funcionamento do circuito da fig. 23 aplicamos através de umas das portas digitais, fig. 14, um sinal PWM *Pulse Width Modulation* na entrada do primeiro amplificador operacional, que é um sinal de uma onda quadrada, onde esse sinal varia de  $0V$  a  $+5V$ . O

sinal de  $0V$  a  $+5V$  do Arduino eram mantidos por um segundo, como podemos ver na fig. 24.

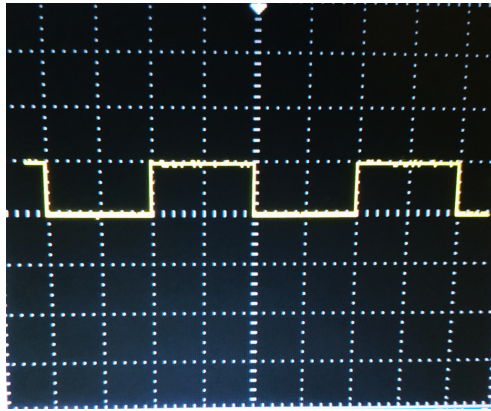


Figura 24 – Foto retirada do osciloscópio exibindo o sinal PWM enviado ao amplificador operacional, onde o eixo da ordenada está numa escala de  $5V$ .

Ao passar pelo primeiro amplificador operacional inversor, o sinal do PWM será comparado em relação a tensão de  $-12V$  a  $+12V$ . Assim obtendo um sinal de uma onda quadrada com um alcance de  $-12V$  a  $+12V$ , como mostrado na fig. 25.

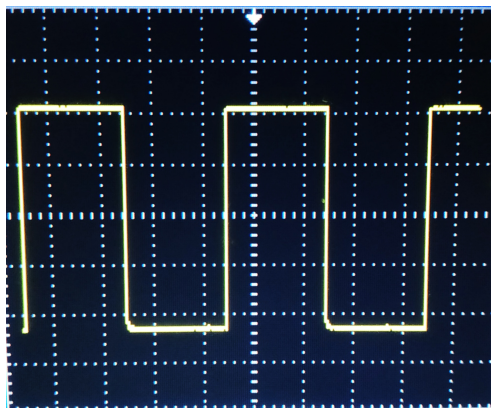


Figura 25 – Foto retirada do osciloscópio exibindo o sinal PWM modulado em um alcance de  $-12V$  a  $+12V$ , onde o eixo da ordenada está numa escala de  $5V$ .

O sinal representado na fig. 25 é destinado ao segundo amplificador operacional, note que no esquema, esse amplificador está vinculado à um capacitor. Esse capacitor produz uma rampa que inicia em  $-12V$  e termina em  $+12V$ , veja detalhes na fig. 26. Essas rampas de subida e descida levam um segundo para serem realizadas, tempo que por ter escolhido que o sinal PWM de  $0V$  a  $+5V$  fossem mantidos por um segundo.

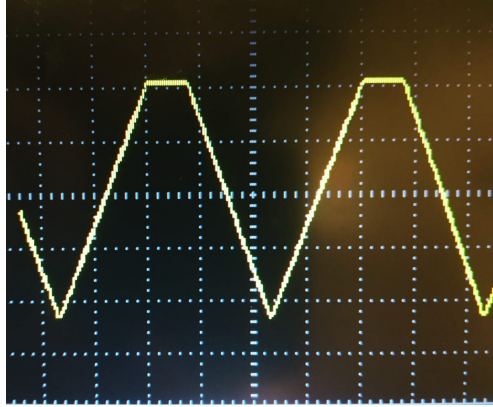


Figura 26 – Foto retirada do osciloscópio exibindo a rampa formada pelo amplificador operacional que varia de  $-12V$  a  $+12V$ , onde o eixo da ordenada está numa escala de  $5V$ .

Repare que na fig. 26 o nosso sinal não alcança os  $+12V$ , pois o capacitor saturou antes alcançar tal tensão. Entretanto, por questão de tempo e também porque esse fato não atrapalha para a geração das imagens interferométricas prosseguimos o trabalho. O sinal exibido na fig. 26 é a tensão que aplicada no PZT, ocasionando assim a alteração do caminho óptico, porém para realizarmos o tratamento de dados de maneira eficiente, é necessário que o Arduino receba o valor da tensão aplicada no PZT. Contudo, como já mencionado o Arduino só opera com valores de  $0V$  a  $+5V$ . Desta maneira o sinal da fig. 26 precisou ser normalizado de modo que permanecesse dentro do sinal de leitura do Arduino. O terceiro amplificador operacional, opera uma redução do sinal da fig. 26, para mais detalhes observe a figura 27, possibilitando assim que o Arduino realize a leitura do tensão aplicada no PZT.

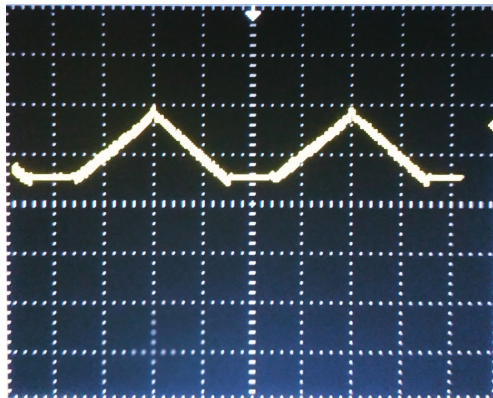


Figura 27 – Foto retirada do osciloscópio exibindo a rampa normalizada para que o valor da tensão ficasse entre de  $0V$  e  $5V$ , onde o eixo da ordenada está numa escala de  $2V$ .

Atente ao fato que na fig. 27 a saturação causada no capacitor, reflete no sinal que é lido pelo Arduino. Contudo tal fato não irá afetar na formação das imagens interferométricas, este fato será melhor explicado de maneira mais clara no capítulo 4.

## 4 Resultados

Neste capítulo apresentaremos o protocolo elaborado para a geração das imagens interferométricas e a obtenção da variação da diferença de fase da mesma. Também mostraremos as análises realizadas de três substâncias diferentes, com o propósito de obter a variação do índice de refração e obter um grau de confiabilidade do equipamento.

### 4.1 Geração de imagens interferométricas

Com a eletrônica e a programação responsável pela aquisição de dados prontas, iniciamos a obtenção de medidas experimentais para a geração das imagens interferométricas. As aquisições dos resultados foram feitas utilizando o Arduino em conjunto com o Python. Os códigos utilizados neste capítulo estão presentes no anexo C.

Para realizar a formação de imagens interferométrica, utilizamos três amostras diferentes:

- Ar em ambos braços do interferômetro, com ambos recipientes do porta amostras abertos;
- Água destilada em ambos braços do interferômetro com ambos recipientes do porta amostras abertos;
- Água destilada com sal de cozinha em ambos braços do interferômetro, onde o recipiente do controle foi lacrada com um fita adesiva e enquanto o recipiente de amostra ficou exposto.

Nesta seção apresentaremos o protocolo que realizamos para a geração das imagens interferométricas utilizando ar em ambos os braços do interferômetro, o mesmo procedimento foi realizado para as outras substâncias. Como mencionado os dados foram salvos no computador utilizando o Arduino juntamente com o Python. Os dados foram salvos realizando a aquisição dos sinais no fototransistor a cada  $10ms$  e salvando tais dados em um arquivo no formato CSV (sigla para Comma Separated Values). Os arquivos foram salvos indicando a data e hora da aquisição e com os valores da tensão no PZT e no fototransistor. Os sinais obtidos na aquisição estão representados na figura 28,

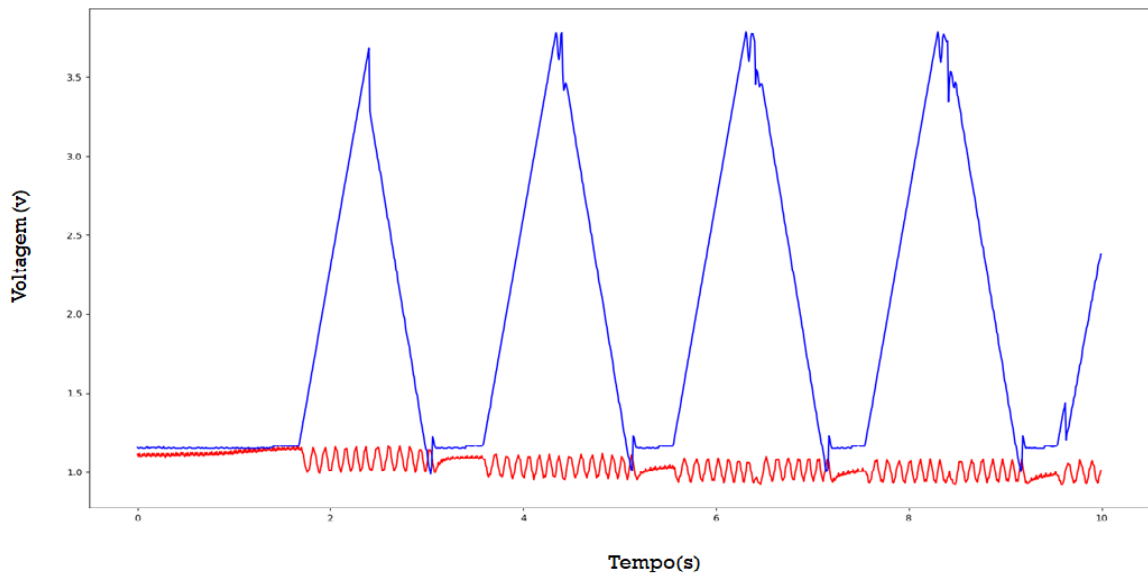


Figura 28 – Gráfico mostrando os sinais do *PZT* e fototransistor. As curvas de azul e vermelho são, respectivamente, os sinais de tensão no *PZT* e no fototransistor.

A curva em azul é o sinal no *PZT* e o sinal gerado pelo fototransistor está em vermelho. Como o objetivo é a construção de uma imagem interferométrica, um protocolo para criação do mesma foi realizada. Primeiramente, um filtro passa baixa foi aplicado na curva do sinal no *PZT*, com o proposito de eliminar as frequências altas dos nossos dados experimentais, presentes no ponto máximo do sinal do *PZT*, veja mais detalhes na figura 28. Aplicado o filtro no sinal do *PZT*, normalizamos o sinas do *PZT* e do fototransistor, desta maneira o ponto máximo da curva do *PZT* ficou mais fácil de ser detectado. Com o máximo identificado,ajusta-se uma curva a esses dados. Para gerar a curva de ajuste, pegamos cinquenta pontos a partir do ponto máximo do sinal do *PZT* e traçamos uma reta que até cortar o zero do eixo das ordenadas, veja mais detalhes na fig. 29.

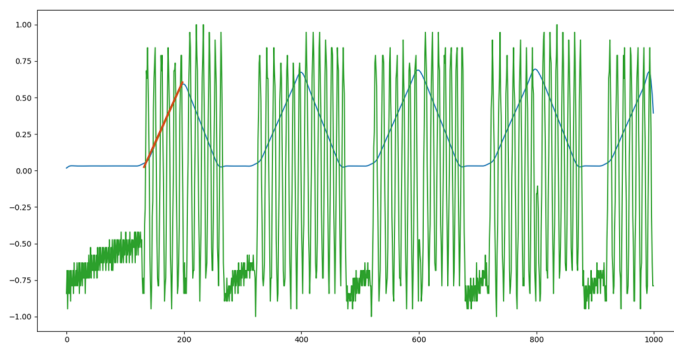


Figura 29 – Gráfico mostrando o sinal do *PZT* normalizado, representado pela curva azul, juntamente com a curva de ajuste indicada como laranja e o sinal lido pelo fototransistor identificada com verde. Os valores eixo da ordenada está normalizado de 1 à -1 e o eixo das abscissas corresponde o valor obtido do Arduino



Em seguida, todos os padrões de interferência abaixo de cada curva de ajuste, foram salvos na linha de uma matriz. Os padrões de interferência salvos continham 69 pontos, isto é, utilizando este método a nossa imagem interferométrica ficou limitada para 69 pontos no eixo da abscissa, veja tal detalhe na imagem interferométrica indicada pela fig. 30, onde os eixos das abscissas é o tempo para formação do padrão interferométrico e o eixo das ordenadas o tempo decorrido para realizar a formação da imagem.

## 4.2 Obtenção da variação da diferença de fase

Para a aferir a variação da diferença de fase a partir das imagens interferométricas, utilizamos os métodos discutidos da seção 2.3. Ressaltando que aqui, não realizamos nenhum filtro na imagem representada pela figura 30 e nas demais imagens interferométricas, os filtros serão aplicados somente nas transformadas de Fourier.

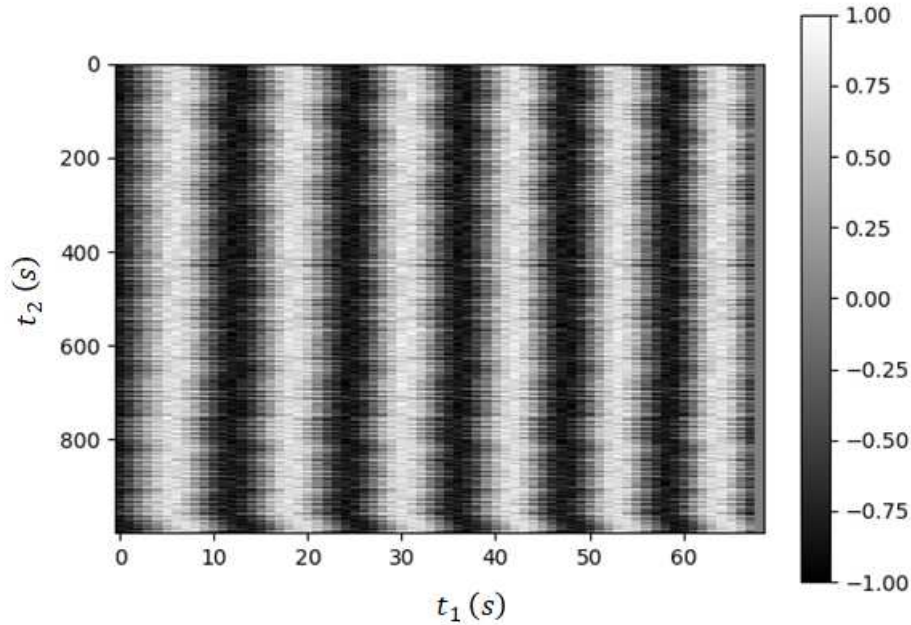


Figura 30 – Imagem interferométrica  $Ar/Ar$  projetada no eixo dos tempos, onde o eixo da abscissa representa  $t1(s)$  que é o tempo de formação de um padrão de interferência e  $t2(s)$ , tempo necessário para a geração da imagem, no eixo da ordenada.

Observe que a imagem representada na fig. 30 é muito semelhante a imagem formada na seção 2.3, porém com mais ruídos. Aqui, representaremos apenas a projeção da imagem interferométrica em  $t1(s)$  e  $t2(s)$ .

Mesmo não conhecendo a função geratriz da imagem indicada pela fig. 30, sabemos que a sua transformada de Fourier nos fornecerá valores relacionado a fase da nossa imagem interferométrica. Utilizando o mesmo procedimento de cálculo numérico descrito na seção 2.3, calculamos a transformada de Fourier da imagem da fig. 30 e obtemos a fig. 31. O tempo para a aquisição de dados para gerar a fig. 30 foi de aproximadamente 33 minutos. Para uma melhor análise, particionamos o imagem interferométrica  $Ar/Ar$  a cada 200 linhas em relação ao eixo das ordenadas, ficando assim com cinco regiões para serem analisadas. Realizando a transformada de Fourier em cada umas das regiões, obtemos assim 5 regiões a serem analisadas. A transformadas de Fourier para a primeira região está apresentada na fig. 31. Repare que na fig. 30 o eixo das abscissas, possui 69

pontos, já o eixo das ordenadas 200, isto é, resultaremos em uma imagem na dimensão de 69 por 200. Este é o motivo pelo qual na seção 2.3, geramos imagens nestas ordens de grandezas e ressaltamos que as nossas imagens possuíam dispersões que poderiam ser reduzidas utilizando um filtro, tais dispersões que também serão encontradas ao realizar a transformada de Fourier nos dados experimentais, entretanto os dados aqui apresentados, já aplicamos o filtro a fim de reduzir as nossas dispersões.

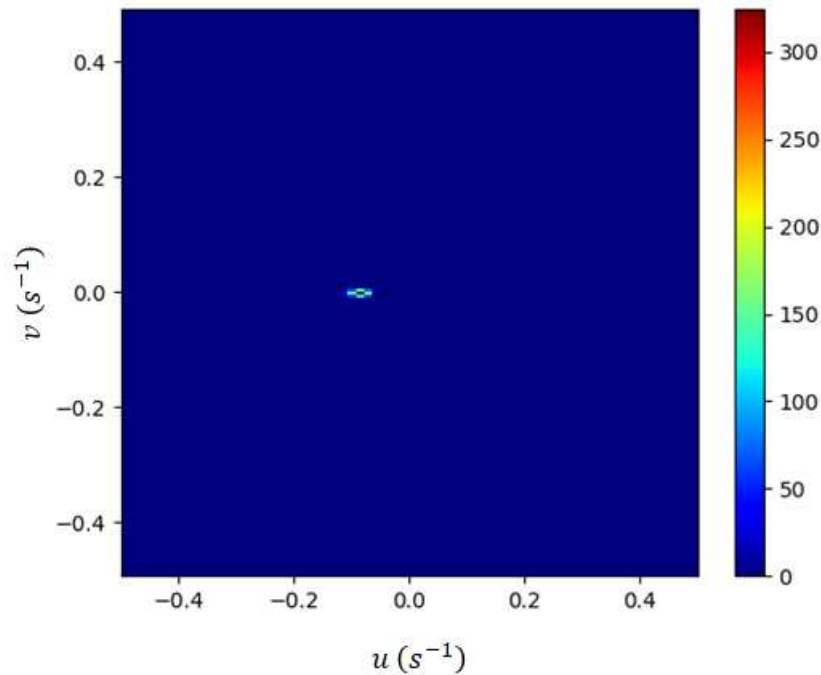


Figura 31 – Transformada de Fourier da imagem interferométrica Ar/Ar projetada no plano das frequências, onde os eixos das ordenadas e abscissas possuem unidade de frequência angular. Observe que há um ponto mais intenso na posição zero do eixo das ordenadas. Este ponto, representa o valor da delta discutido no capítulo 2.

Observe que na fig. 31 tem-se um ponto bem localizado, com uma dispersão ao seu redor. A transformada de Fourier em todas as outras regiões foram similares a da fig. 31. Como já foi mencionado na seção 2.3, se houvesse uma fase na imagem, obteríamos uma inclinação nos padrões da fig. 30. Esse fato aponta que o parâmetro responsável por produzir as inclinações em nossos padrões, adotado como  $b$  na seção 2.3, é aparentemente nulo para a fig. 31.

O fato da fig. 31 apresentar somente uma delta de Dirac e não duas, como encontrado no modelo da seção 2.3, é que realizamos um corte quadricular com as dimensões 2 por 2 em torno de um dos pontos máximos da transformada de Fourier, uma vez que esses pontos são simétricos, não importa o quadrante de onde o ponto está sendo analisado. Com o corte realizado, calculamos a média e o desvio padrão sobre a região formada, observe detalhes na fig. 32.

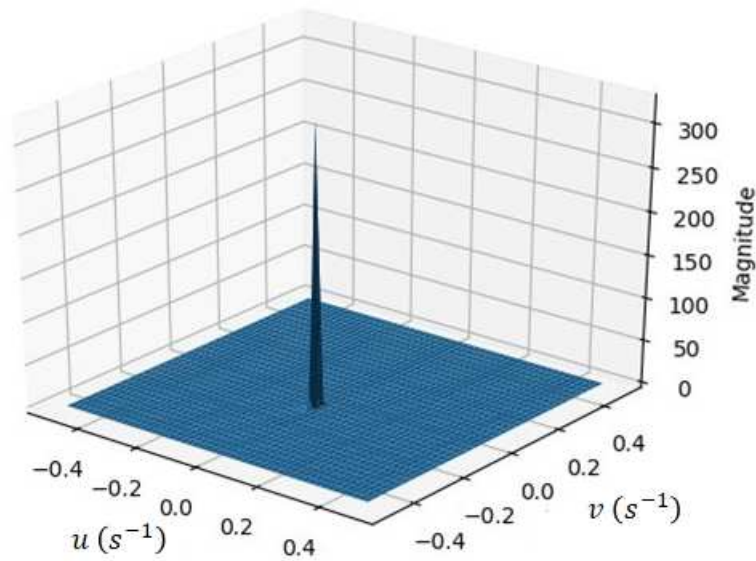


Figura 32 – Módulo da transformada de Fourier sobre uma das deltas de Dirac dos ao realizar o corte na transformada de Fourier da Imagem  $Ar/Ar$ .

Ressaltando que o mesmo procedimento também foi realizado para as outras 4 regiões, entretanto, aqui apresentaremos a figura formada apenas da transformada de Fourier da primeira região. Observe que na fig. 32, ainda possui uma dispersão maior para o eixo  $u$  do que para o eixo  $v$ , mais detalhes na tabela 1.

Tabela 1 – Tabela expondo os valores encontrados para  $u$  e  $v$  após a realização da transformada de Fourier para a análise de  $Ar/Ar$ .

Regiões	Valor médio para $u(s^{-1})$	Desvio Padrão para $u(s^{-1})$	Valor médio para $v(s^{-1})$	Desvio Padrão para $v(s^{-1})$
Região 1	0.08733644	0.00770400	0.00017192	0.00246756
Região 2	0.08725686	0.00763666	0.00004483	0.00240744
Região 3	0.08728101	0.00766297	0.00022693	0.00252288
Região 4	0.08723142	0.00762780	0.00015506	0.00243723
Região 5	0.08727139	0.00767350	0.00000112	0.00239018

A partir da Tab. 1 geramos um gráfico mostrando os valores de  $u$  e  $v$  de todas as regiões, juntamente com a barra de erro, ficando, assim, mais visível de observar as alterações nos resultados.



Figura 33 – Frequências de  $u$  e  $v$  representadas, respectivamente, pelas curvas azul e vermelha, mostrando os valores das frequências angulares obtidos do espaço recíproco para  $Ar/Ar$ .

Note que a barra de erro para  $u$  ficou maior do que a barra de erro de  $v$ , com forme já mencionado a partir da figura 32, que a dispersão em  $u$  seria maior que dispersão para  $v$ . Pois, como já discutido na seção 2.3, a dispersão está relacionada ao efeito de *leakage*. Outro fato que contribui para a dispersão é que esta imagem é resultante de dados experimentais, isto é, existem várias condições que podem contribuir para o aumento de tal dispersão, como por, exemplo: ruídos da eletrônica, alteração abrupta da temperatura (fato que pode ocorrer porque ainda não possuímos um controle térmico), movimentações externas que podem produzir um movimento na mesa onde o equipamento está instalado, entre outros motivos. Entretanto, os valores de  $u$  e  $v$  da nossa imagem não sofreram alterações visíveis.

Com o protocolo definido iniciamos os testes para as outras substâncias. Para água destilada em ambos os braços, obtivemos a seguinte imagem interferométrica, veja a Fig .34.

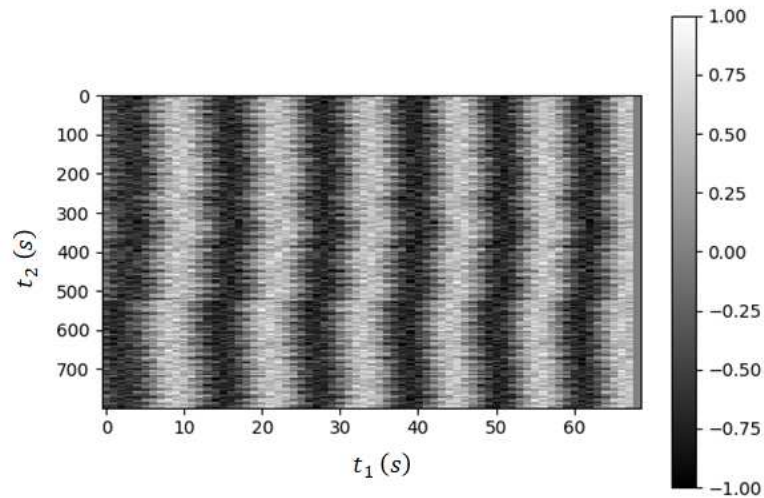


Figura 34 – Imagem interferométrica *Água/Água* projetada no eixo dos tempos: o eixo da abscissa representa  $t_1(s)$ , que é o tempo de formação de um padrão de interferência. Já no eixo das ordenadas  $t_2(s)$  é o tempo necessário para a geração da imagem.

A imagem interferométrica gerada com água em ambos os braços é bem semelhante com a imagem da Fig. 30. Calculando a transformada de Fourier para a imagem interferométrica *Água/Água*, obtivemos a imagem mostrada na Fig 35.

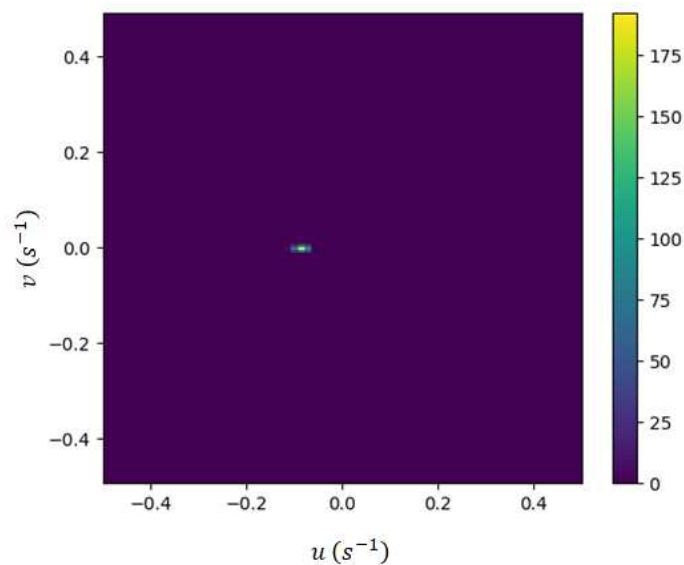


Figura 35 – Transformada de Fourier da imagem interferométrica *Água/Água* projetada no plano das frequências, onde os eixos das ordenadas e abscissas possuem unidade de frequência angular. Observe que há dos pontos mais intensos na posição zero do eixo das ordenadas. Estes pontos representam os valores das deltas discutidos no capítulo 2.

Observe que a Fig. 35, também apresenta um ponto de máximo próximo de zero muito bem definido em relação ao eixo das ordenadas, isto é, assim como observamos nas imagens geradas para  $Ar/Ar$ , o nosso parâmetro, que proporcionaria uma inclinação no nosso padrão interferométrico, deve ter um valor próximo de zero. Ressaltando, que também utilizamos o mesmo o corte quadricular conforme feito para realizar as análises das imagem interferométrica para  $Ar/Ar$ . A fim de obter uma melhor visualização, veja a imagem representada na Fig. 36.

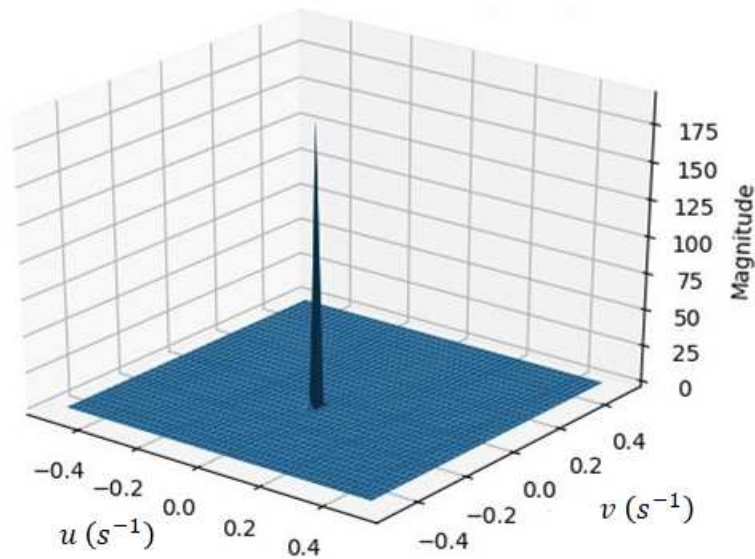


Figura 36 – Representação da transformada de Fourier em 3D sobre uma das deltas de Dirac para a imagem interferométrica  $\hat{A}gua/\hat{A}gua$ .

Assim como feito para a imagem interferométrica para  $Ar/Ar$ , subdividimos em regiões de 200 linhas em relação ao eixo das ordenadas. Entretanto, o tempo para gerar a Fig. 34 ou seja, o tempo de aquisição de dados foi de aproximadamente 23 minutos. Logo esta imagem apresenta apenas 4 regiões. Os valores para  $u$  e  $v$  obtidos, estão descrito na Tab. 2

Tabela 2 – Tabela expondo os valores encontrados para  $u$  e  $v$  após a realização da transformada de Fourier para a análise de  $\hat{A}gua/\hat{A}gua$ .

Regiões	Valor médio para $u(s^{-1})$	Desvio Padrão para $u(s^{-1})$	Valor médio para $v(s^{-1})$	Desvio Padrão para $v(s^{-1})$
Região 1	0.08690831	0.00747792	0.00022481	0.00264715
Região 2	0.08688131	0.00745437	0.000066	0.00269284
Região 3	0.08695998	0.00737938	0.00139309	0.00325233
Região 4	0.08696527	0.00743001	0.000205	0.00244517

Com os dados da Tab. 2, também geramos uma gráfico mostrando os valores de  $u$  e  $v$  para 4 regiões, representado pela figura 37.



Figura 37 – Frequências de  $u$  e  $v$ , representadas respectivamente pelas curvas azul e vermelha mostram valores das frequências angulares obtidos do espaço recíproco para  $\text{Água}/\text{Água}$ .

Observe que o gráfico da Fig. 37, apresenta valores para  $u$  semelhantes com o gráfico da Fig. 33. Entretanto, para as transições da região 2 até a região 4, houve pequenas alterações no valores de  $v$ , No entanto a partir da Tab. 2 é possível notar que os valores estão bem próximo de zero.

O próximo teste foi utilizar água destilada com sal em ambos os braços do interferômetro. A concentração escolhida foi  $2g$  de sal para  $80mL$  de água destilada, ou seja,  $0,025g/mL$ . Colocamos a mesma quantidade do líquido na cavidade de controle e na cavidade da amostra do porta amostra. A cavidade de controle foi lacrada com uma fita adesiva para evitar a evaporação do líquido. Já a cavidade de controle foi mantida aberta, ou seja, a cavidade da amostra estava sujeita a evaporação. O porta amostra foi posicionado no disco conforme indicado na Fig. 22 e, então, começamos a realizar a aquisição de dados. O propósito de realizar a aquisição da solução de  $\text{Água} + \text{NaCl}/\text{Água} + \text{NaCl}$  dessa maneira é para certificar que o método de imagem interferométrica verifica a variação da diferença de fase, e não para estudar a características da solução. O experimento foi realizado em temperatura ambiente, em torno de  $24^{\circ}C$ . A imagem gerada por tal solução está apresentada na Fig. 38. A fim de obter valores mais perceptíveis, a aquisição de dados para esta amostra durou em torno de 3 horas.



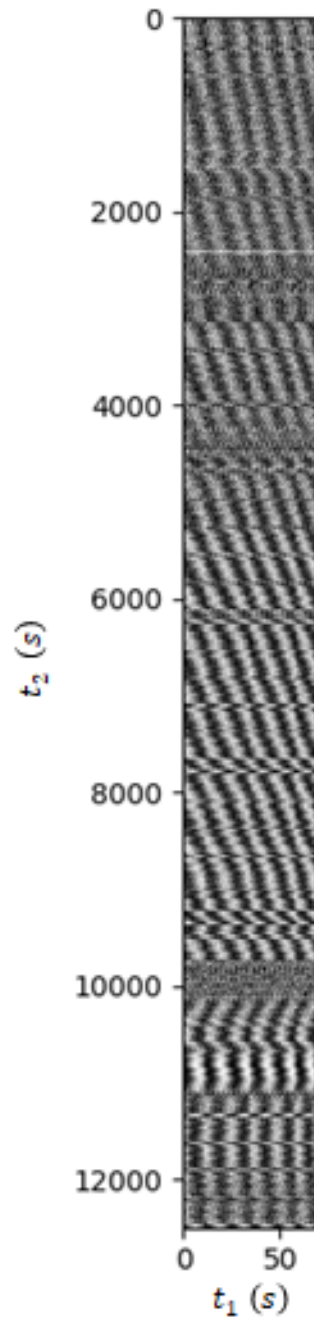


Figura 38 – Imagem interferométrica  $\text{Água} + \text{NaCl} / \text{Água} + \text{NaCl}$  projetada no eixo dos tempos, onde o eixo da abscissa representa  $t_1(s)$  que é o tempo de formação de um padrão de interferência e  $t_2(s)$ , tempo necessário para a geração da imagem, no eixo da ordenada.

Na Fig. 38, podemos observar há uma inclinação das franjas na imagem interferométrica. Significando que tivemos um termo de fase na imagem interferométrica, isto é conseguimos verificar se há mudanças de fase apenas olhando para a imagem in-

terferométrica. Esta fase está correlacionada com o parâmetro  $b$ . Em algumas regiões os padrões ficaram muito ruidosos e em outras regiões a inclinação do padrão sofreu alterações. Uma possível explicação inicial para tais fatos é que a mistura de água com sal não ficou totalmente homogênea. Assim, produzindo a formação de vários sítios de concentração na solução e que as distorções encontradas na imagem, estivesse ligada ao fato da passagem de tais sítios na frente do feixe laser. Entretanto, segundo a referência [27], a alteração de  $0,1^{\circ}\text{C}$  para uma concentração de  $0.3\text{mol/L}$  de  $\text{NaCl}$  em água, pode causar uma mudança até na ordem de  $10^{-5}$  na variação do índice de refração, Consequentemente, contribuindo com erro no valor da variação de fase durante a análise interferométrica. Ressaltando que a concentração utilizada para gerar a imagem da imsal era de aproximadamente  $0.47\text{mol/l}$ , isto é, próximo da concentração utilizada na referencia [27]. Além disso, o nosso equipamento ainda não está disposto de um controle térmico e, como a medida foi realizada durante horas, a variação de temperatura do ambiente, resultou em alterações abruptas nos dados, como pode ser observado no gráfico da Fig. 39.

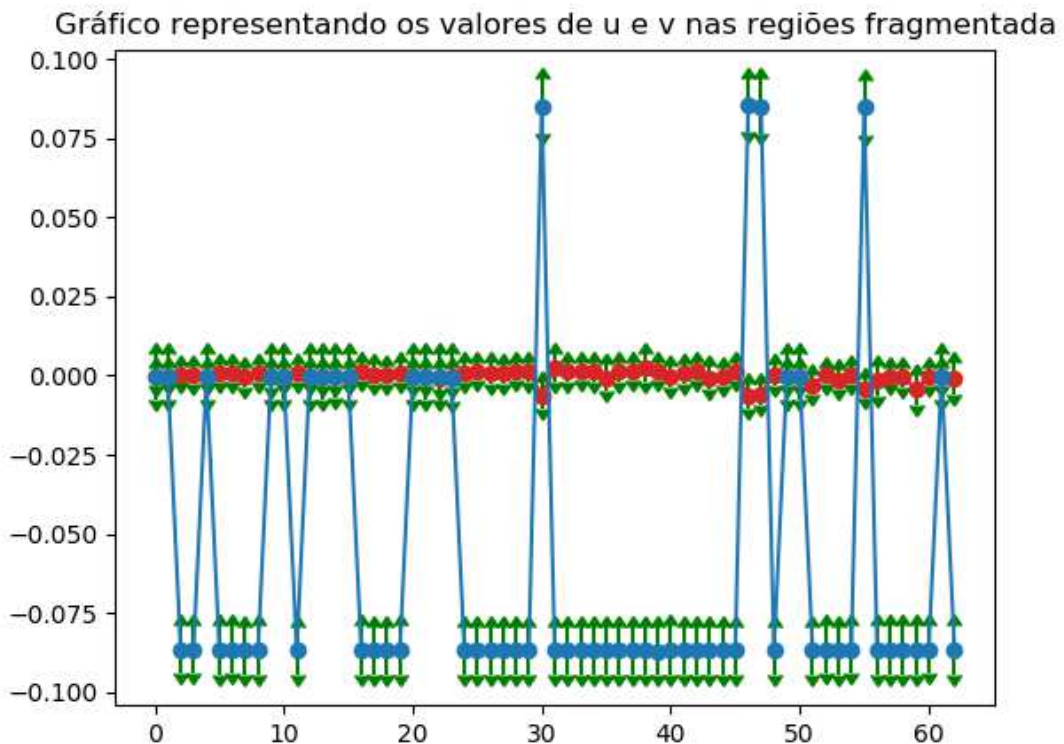


Figura 39 – Frequências de  $u$  e  $v$ , representadas respectivamente pelas curvas azul e vermelha, mostram valores das frequências angulares obtidos do espaço recíproco para  $\text{Água} + \text{NaCl}/\text{Água} + \text{NaCl}$

Os valores das frequências representado na Fig. 39, mostram que há pequenas alterações em  $v$ . Já o valor de  $u$ , que até então manteve constante nas outras análises sofreu mudanças abruptas.

Desta forma, podemos finalizar essa seção, certificado que o método de imagem interferométrica apresentam bons resultados, como observados para as análises realizadas para *Ar/Ar* e *Água/Água*, pois para tais testes, tanto o braço de amostra quanto o de controle, não sofreram nenhum tipo de alteração, pois havia as mesmas condições em ambos recipientes. No entanto para a análise de dados no qual os braços de amostra e de controle estejam sujeito a mudanças é necessário a implementação de um controle térmico caso queira realizar um estudo mais detalhado sobre tal substância, mas como a intenção era apenas gerar uma imagem cuja apresentação uma variação na diferença de fase, o método de imagem interferométrica se torna bem eficaz. Uma vez que é possível observar se houve mudança ou não na fase, apenas olhando para a imagem interferométrica.

### 4.3 Obtenção da variação do índice de refração

Possuindo o valor da variação da diferença de fase, assunto tratado na seção anterior deste mesmo capítulo, é possível obter a variação do índice de refração das substâncias que foram analisadas nesta dissertação. A finalidade de realizar essas análises é para obtermos uma grau de confiabilidade do interferômetro.

Tomando como base o esquema da Fig. 40 e descrevendo o nosso feixe laser como uma onda plana, isto é,  $\mathbf{E} = \mathbf{E}_0 e^{ikz}$ . Podemos descrever um caminho óptico para o braço de controle e para o braço de amostra.

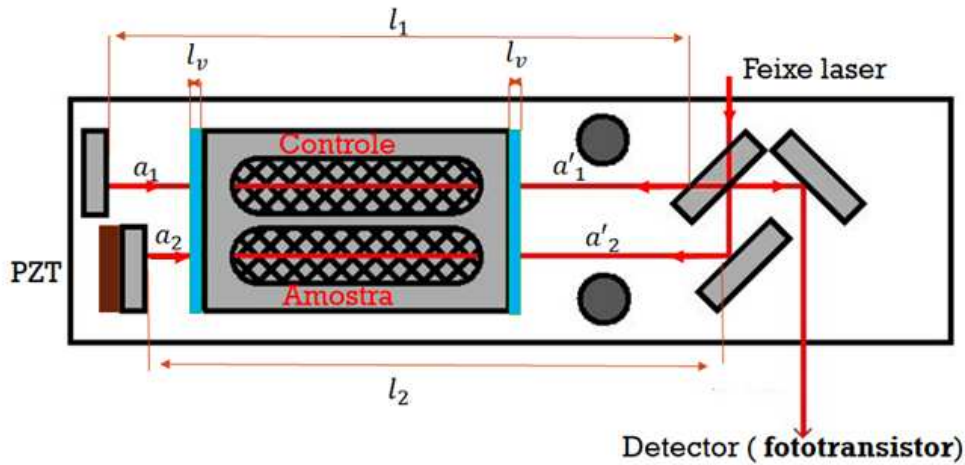


Figura 40 – Esquema do interferômetro de Michelson, onde  $l_1$  é o tamanho total do braço de controle,  $l_2$  é o tamanho do braço de amostra e  $l_v$  representa a espessura do vidro fixado no portas amostras.

A seguir, temos as expressões para as fases.

$$\phi_1 = a_1 \frac{\omega}{c} n_{ar} + \frac{\omega}{c} a'_1 n_{ar} + 2l_v \frac{\omega}{c} n_v + D_c \frac{\omega}{D} n_c \quad (4.1)$$

$$\phi_2 = a_2 \frac{\omega}{c} n_{ar} + \frac{\omega}{c} a'_2 n_{ar} + 2l_v \frac{\omega}{c} n_v + D_a \frac{\omega}{D} n_a + \Delta l \frac{\omega}{c} n_{ar} \quad (4.2)$$

onde  $\Delta l$  é a variação do caminho proporcionado pelo PZT,  $l_1$  e  $l_2$  são os comprimentos totais do braço do interferômetro. Esses podem ser subdivididos em  $a_1$ ,  $a'_1$ ,  $l_v$ ,  $D_c$  (comprimento do ambiente onde ficará o controle),  $a_2$ ,  $a'_2$ ,  $D_a$  (comprimento do ambiente onde ficará a amostra, sendo que  $D_a = D_b$ ),  $\omega$  é a frequência angular do feixe laser He-Ne ( $\lambda 632,8nm$ ),  $c$  a velocidade da luz,  $n_{ar}$  é o índice de refração do ar,  $n_v$  é o índice de refração do vidro,  $n_c$  é o índice de refração do controle e  $n_a$  o índice de refração da amostra. Realizando a diferença de fase  $\Delta\phi$ , obtivemos a seguinte expressão

$$\Delta\phi = \left[ a_2 \frac{\omega}{c} n_{ar} + \frac{\omega}{c} a_2' n_{ar} + 2l_v \frac{\omega}{c} n_v + D_a \frac{\omega}{c} n_a + \Delta l \frac{\omega}{c} n_{ar} \right] - \left[ a_1 \frac{\omega}{c} n_{ar} + \frac{\omega}{c} a_1' n_{ar} + 2l_v \frac{\omega}{c} n_v + D_c \frac{\omega}{c} n_c \right]$$

O nosso sistema está disposto em uma caixa térmica, deste modo, para medidas efetuados em um pequeno intervalo de tempo  $\Delta t$  podemos desprezar as mudanças que podem ocorrer nos comprimentos de  $a_1$ ,  $a_1'$ ,  $l_v$ ,  $D_c$ ,  $a_2$  e  $a_2'$ . Essa consideração deve ser feita pois, segundo a Ref. [27], a variação de temperatura pode resultar em alterações no nosso índice de refração.

Para obter uma diferença de índice de refração da substância contida no recipiente de amostra comparada com a do controle, efetuamos uma variação em  $\Delta\phi$ , isto é, obtivemos uma equação que correlaciona a variação de índice de refração com variação da diferença fase, como mostrado na equação 4.3.

$$\delta[\Delta\phi(t_2 - t_1)] = \frac{\omega}{c} D [\Delta n_{ca}(t_2 - t_1)] \quad (4.3)$$

onde  $\Delta n_{ca}$  é a variação do índice de refração do controle em relação a amostra. Podemos reescrever a equação 4.3 como

$$\Delta n_{ca}(t_2 - t_1) = \frac{c}{\omega D} \delta[\Delta\phi(t_2 - t_1)]. \quad (4.4)$$

Observe que a partir desse cálculo não é possível calcular o índice de refração absoluto, mas somente a variação do índice de refração. Com esse método é possível estudar a evolução de materiais em relação ao tempo, ressaltando que o material precisa ser transparente na faixa de frequência do laser He-Ne.

A partir da equação 4.4 juntamente com os dados obtidos na seção 4.2, é possível calcular a variação do índice de refração para as substâncias analisadas. Para isso é necessário realizar um calculo de propagação de erro das nossas incerteza. Considerando uma função  $f$  arbitrária que possui dependência com variáveis  $A$ ,  $B$ ,  $C$  e  $D$  da forma a apresentada na equação 4.6.

$$f = \frac{AB}{CD}, \quad (4.5)$$

podemos relacionar a variância de  $f$  com o desvio padrão das variáveis  $A$ ,  $B$ ,  $C$  e  $D$ , da seguinte maneira,

$$\left( \frac{\sigma f}{f} \right)^2 \approx \left( \frac{\sigma A}{A} \right)^2 + \left( \frac{\sigma B}{B} \right)^2 + \left( \frac{\sigma C}{C} \right)^2 + \left( \frac{\sigma D}{D} \right)^2. \quad (4.6)$$

Que pode ser reescrita como

$$\sigma f \approx f \sqrt{\left( \frac{\sigma A}{A} \right)^2 + \left( \frac{\sigma B}{B} \right)^2 + \left( \frac{\sigma C}{C} \right)^2 + \left( \frac{\sigma D}{D} \right)^2}, \quad (4.7)$$

onde  $\sigma A$ ,  $\sigma B$ ,  $\sigma C$  e  $\sigma D$  representam o desvio padrão das variáveis  $A$ ,  $B$ ,  $C$  e  $D$ . Assim é possível rescrever a função  $f$  como mostrado na equação 4.8

$$f = (\langle f \rangle \pm \sigma f). \quad (4.8)$$

Podemos relacionar a equação 4.4 com a equação 4.8. Entretanto, desprezaremos o erros de  $c$ ,  $\omega$  e  $D$ . Desta forma, podemos reescrever a equação 4.4 como,

$$\Delta n_{ca} = \frac{c}{\omega D} [\langle \delta[\Delta\phi(t_2 - t_1)] \rangle \pm \sigma[\delta[\Delta\phi(t_2 - t_1)]]], \quad (4.9)$$

isto é, consideraremos apenas a variância da variação da diferença de fase. Adotando valores de  $c = 2.99702547 \cdot 10^{-8} m/s$ ,  $\omega = 2\pi c/\lambda$  e  $D = 0.03385m$ , onde  $c$  é velocidade da luz no ar,  $\lambda$  é o comprimento de onda do laser e  $D$  o comprimento das aberturas do porta amostras. Ressaltando que  $\Delta\phi(t_2 - t_1)$  são os valores encontrados nos gráficos das Fig. 33, 37 e 39, uma vez que o valor de  $v$  de tais gráficos estão correlacionados ao parâmetro  $b$ .

Assim, utilizando os dados da seção 4.2, para  $A_r/A_r$  em ambos os braços do interferômetro obtivemos o gráfico, mostrado na Fig. 41, para a variação do índice de refração.

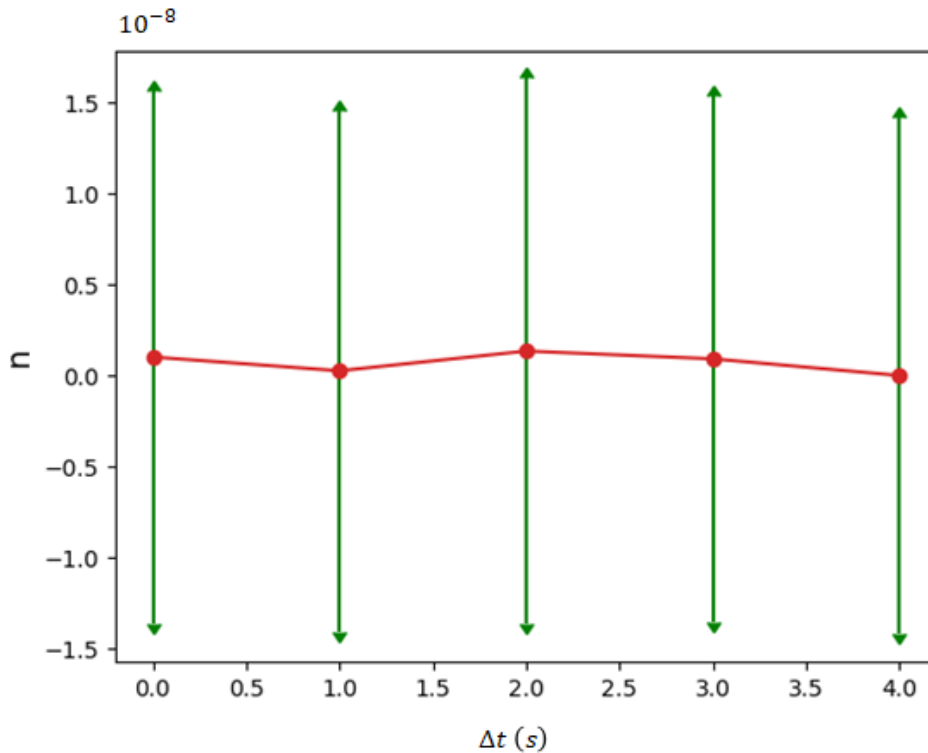


Figura 41 – Variação do índice de refração da imagem interferométrica para  $A_r/A_r$  por  $\Delta t$ , onde  $1e^{-8}$ , representa  $1 \cdot 10^{-8}$ .

Observe que na Fig. 41, que a variação do índice de refração está próximo de zero, pois o eixo das ordenadas está em uma escala de  $1e^{-8}$ . Fato esse que mostra que o nosso interferômetro possui uma precisão de analisar a diferença de índice de refração na ordem de grandeza  $1e^{-8}$ . Este resultado também é esperado para a variação do índice de refração para *Água/Água*,

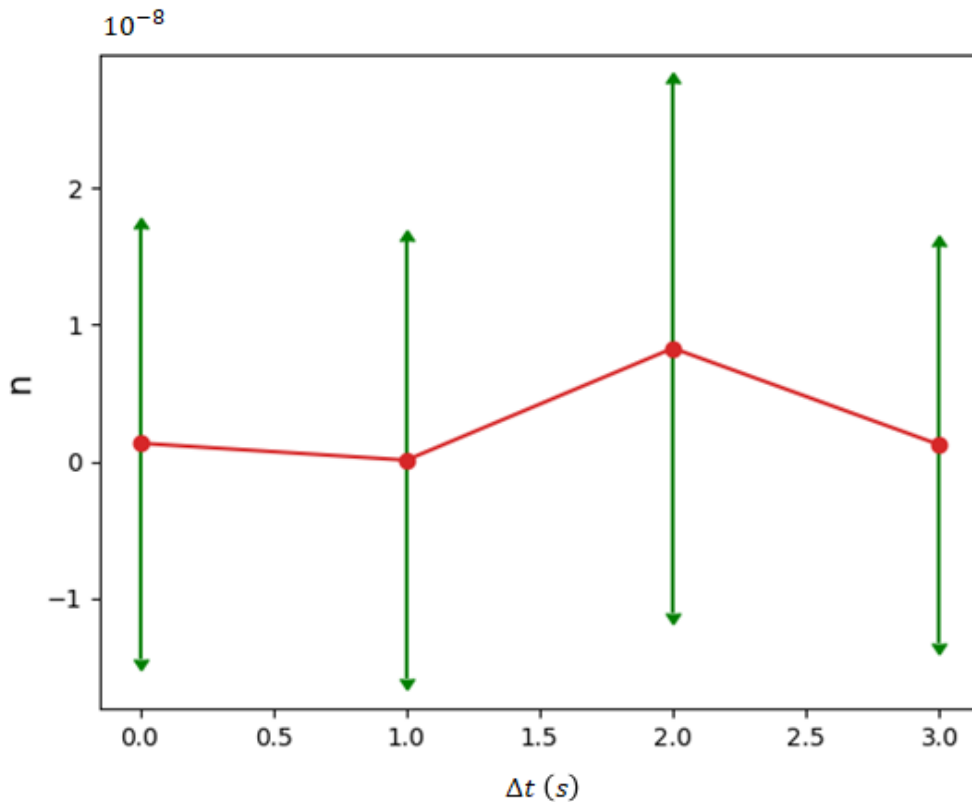


Figura 42 – Variação do índice de refração da imagem interferométrica para *Água/Água* por  $\Delta t$ , onde  $1e^{-8}$ , representa  $1 \cdot 10^{-8}$ .

Observe que a variação do índice de refração para *Água/Água*, o gráfico da Fig. 42, possui uma inclinação na curva, mostrando que houve uma variação no índice de refração. No entanto, como os dados para *Ar/Ar*, o nosso eixo das ordenadas possui a ordem de grandeza de  $1e^{-8}$ , tornando assim esta variação desprezível.

Se realizarmos o mesmo processo para *Água + NaCl/Água + NaCl* obtemos o gráfico representado na Fig. 43

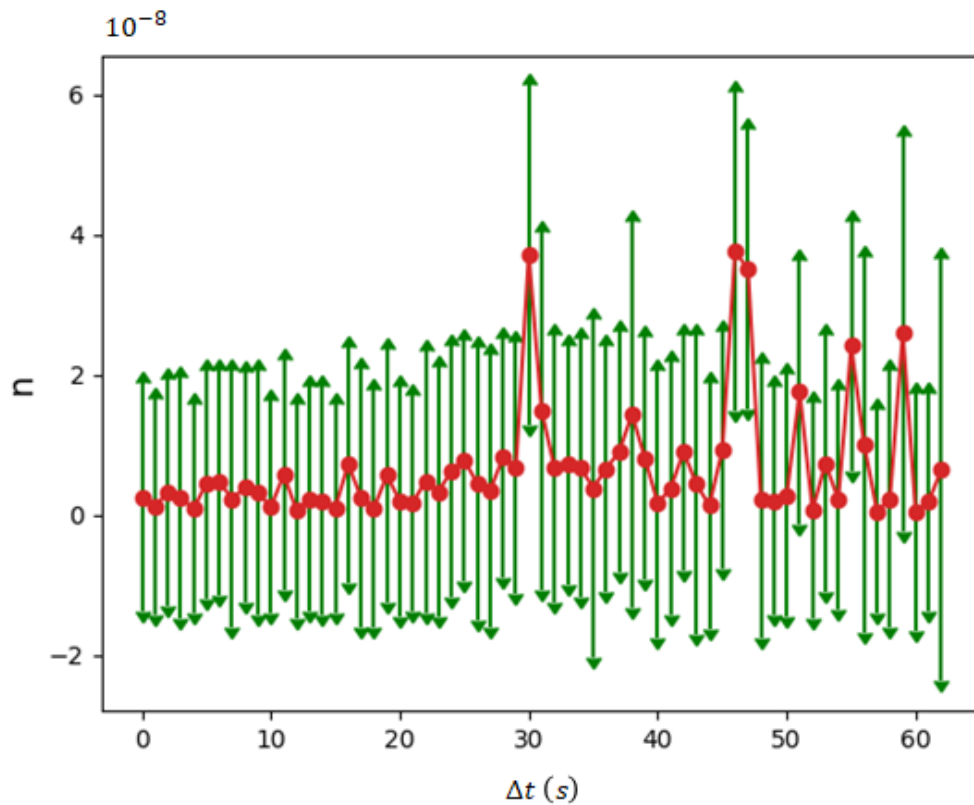


Figura 43 – Variação do índice de refração da imagem interferométrica para *Água+NaCl/Água+NaCl* onde  $\Delta t$  possui unidade de tempo

A Fig. 43 também possui o eixo da ordenada na ordem de grandeza de  $1e^{-8}$ . No entanto, é possível observar variações do índice de refração ao passar ao passar do tempo. Algumas variações são repentinas, entretanto como já mencionado o objetivo é apenas expor que a partir de uma imagem interferométrica é possível determinar variações no índice de refração, propósito que alcançamos com exatidão.



## 5 CONCLUSÕES

As alterações realizadas no interferômetro de atividade metabólica MAI, utilizando uma eletrônica mais compacta e a técnica de tratamento de imagens interferométricas, são apresentadas no decorrer deste capítulo, juntamente com as perspectivas futuras sobre o mesmo.

Mesmo que o modelo teórico de uma imagem interferométrica, nos mostrasse que iríamos obter uma dispersão causando assim variações em nossas medidas, por meio de métodos computacionais conseguimos diminuir as dispersões resultando em resultados apreciáveis.

A linguagem Python, utilizada para realizar o modelo teórico e a análise dos dados experimentais, mostrou-se uma ótima ferramenta para realizar tratamentos de dados e geração de imagens e gráficos. Como o Python é uma linguagem de código aberto, há um vasto número de bibliotecas que podem ser utilizadas e alteradas para diferentes tarefas.

O Arduíno assim como o Python também possui uma linguagem de código aberto e o seu uso está sendo muito disseminado para realizar automatizações de equipamentos. O custo benefício de uma placa de Arduíno é outro motivo da utilização do mesmo. Como o programa é salvo em seu próprio *chip* é possível criar uma placa Arduíno própria de tamanho reduzido.

O controle da parte mecânica se fez muito eficaz. Podendo agora efetuar mudanças em sua lógica diretamente pelo computador, sem a necessidade de alterar a eletrônica.

A análise dos dados obtidos experimentalmente mostrou que o nosso equipamento possui confiabilidade para realizar medidas de variação de índice de refração, uma vez que realizamos o cálculo de variação de índice de refração para diferentes substâncias e conseguimos calcular uma variação do índice de refração na ordem de  $1 \cdot 10^{-8}$ .

Com a reconstrução do interferômetro de atividade metabólica MAI será possível realizar alterações diretamente em sua estrutura mecânica a fim de tornar o equipamento mais compacto e buscar meios de comercializar o mesmo. Futuramente implementaremos um controle térmico e refaremos os mesmos estudo realizado nesta dissertação com o propósito de garantir a precisão do equipamento.

O método proposto de imagem de interferência mostrou rápido e eficaz. Entretanto, ainda como estamos no início do trabalho, melhorias estão sendo e serão realizadas com o propósito de melhorarmos o método de analisar as imagens interferométricas.

Apesar dos resultados encontrados nesta dissertação sejam satisfatório, é importante salientar que a técnica interferométrica possui uma limitação para o nosso equipamento, pois só é possível analisar materiais que sejam transparente para a frequência do laser utilizado.

Concluimos ressaltando que os objetivos propostos, remontagem do MAI e a implementação da técnica de imagem interferométrica foram alcançados e se mostrou útil para realizar o estudo de variação de índice de refração.

## REFERÊNCIAS

- [1] S. C. Zílio, “Óptica moderna: fundamentos e aplicações,” 2001.
- [2] E. S. de Lima Filho, “Medida interferométrica automática de atividade bacteriana,” 2009.
- [3] Instrumentation and M. channel, “*Spectral leakage*.” Disponível em: <<https://www.youtube.com/watch?v=8AA-Do9VeJQ>>. Acesso em 31 de Julho de 2019, 2016.
- [4] D. F. Jardim, R. B. S. Neto, R. Machado, F. Aarestrup, J. De Mendonça, and B. Lesche, “Observing bacterial activity interferometrically,” *European Biophysics Journal*, vol. 32, no. 2, pp. 159–162, 2003.
- [5] R. B. S. Neto, J. P. R. F. de Mendonça, and B. Lesche, “Determination of absolute values of refractive index of liquids using an interferometric method,” *Revista de Física Aplicada e Instrumentação*, vol. 17, no. 2, 2004.
- [6] R. Machado, E. Lima Filho, D. Jardim, M. Ferreira, C. De Faria, R. Duarte, and B. Lesche, “Metabolic activity interferometer: description and calibration of an interferometric method to measure growth of mycobacteria,” *European biophysics journal*, vol. 38, no. 1, pp. 111–119, 2008.
- [7] A. F. Núñez, “Uma introdução ao arduino e suas aplicações na robótica,” *Acessado em: 2ª quinzena de novembro de*, 2016.
- [8] P. Hariharan, *Optical Interferometry*, 2e. Elsevier, 2003.
- [9] A. Abramovici, W. E. Althouse, R. W. Drever, Y. Gürsel, S. Kawamura, F. J. Raab, D. Shoemaker, L. Sievers, R. E. Spero, K. S. Thorne, *et al.*, “Ligo: The laser interferometer gravitational-wave observatory,” *science*, vol. 256, no. 5055, pp. 325–333, 1992.
- [10] G. R. Fowles, *Introduction to modern optics*. Courier Corporation, 1989.
- [11] M. Born and E. Wolf, *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013.
- [12] E. Hecht, “Optics, ed,” *MA: Addison-Wesley Publishing Company*, 1998.
- [13] H. M. Nussenzveig, *Curso de física básica: Eletromagnetismo (vol. 3)*. Editora Blucher, 2015.
- [14] J. W. Brown and R. V. Churchill, *Fourier series and boundary value problems*, vol. 6. McGraw-Hill New York, 2001.
- [15] G. B. Arfken and H. J. Weber, “Mathematical methods for physicists,” 1999.
- [16] A. N. Akansu and H. Agirman-Tosun, “Generalized discrete fourier transform with nonlinear phase,” *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4547–4556, 2010.
- [17] MathWorks, “*fftshift*.” Disponível em: <<https://www.mathworks.com/help/matlab/ref/fftshift.html>>. Acesso em 31 de Julho de 2019, 2019.

- [18] B. Santhanam and T. S. Santhanam, “Discrete gauss-hermite functions and eigenvectors of the centered discrete fourier transform,” in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07*, vol. 3, pp. III–1385, IEEE, 2007.
- [19] S. C. Chapra and R. P. Canale, *Métodos numéricos para engenharia*. McGraw-Hill, 2008.
- [20] V. et all., “*Spicy Lecture Notes*.” Disponível em: <<http://scipy-lectures.org/index.html>>. Acesso em 31 de Julho de 2019, 2019.
- [21] “*Signal Processing*.” Disponível em: <<https://docs.scipy.org/doc/scipy-1.0.0/reference/generated/scipy.signal.hamming.html>>. Acesso em 31 de Julho de 2019, 2019.
- [22] S. Massar and P. Spindel, “Uncertainty relation for the discrete fourier transform,” *Physical review letters*, vol. 100, no. 19, p. 190401, 2008.
- [23] E. Butkov, *Física matemática*. Livros Técnicos e Científicos, 1988.
- [24] S. Arar, “*DFT Leakage and the Choice of the Window Function*.” Disponível em: <<https://www.allaboutcircuits.com/technical-articles/the-dft-leakage-and-the-choice-of-the-window-function2/>>. Acesso em 31 de Julho de 2019, 2017.
- [25] N. Instruments, “*Understanding FFTs and Windowing*.” Disponível em: <<http://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf>>. Acesso em 31 de Julho de 2019, 2019.
- [26] M. H. M. Ferreira and M. C. S. Barcelos, “Desenvolvimento de um heliostato com sistema de controle de posicionamento utilizando arduino,” 2017.
- [27] R. S. Neto, “Técnica interferométrica para medida de alta precisão do índice de refração de líquidos transparentes,” Master’s thesis, Universidade Federal de Juiz de Fora, 2002.

## ANEXO A – Código para geração de um modelo de imagem interferométrica

```

1
2 import matplotlib.pyplot as plt
3 import matplotlib.image as mpimg
4 import numpy as np
5 import pandas as pd
6 from os import walk
7 from scipy.signal import argrelextrema
8 from scipy import optimize
9 from scipy import fftpack
10 from matplotlib.colors import LogNorm
11 from mpl_toolkits.mplot3d import Axes3D
12 from matplotlib import cm
13
14 from scipy.signal import get_window
15
16
17 rawdata=[]
18 width,length=69,200
19 print(width,length)
20 imageRaw = np.zeros((width, length)) # sample 2D array
21
22 w = get_window('hamming', width)
23 w1 = get_window('hamming', length)
24
25 for h1 in range(width):
26 for l1 in range(length):
27 a=0.083/3
28 b=-0.02
29 imageRaw[h1][l1]=1.0*np.cos(2*np.pi*(a*h1+b*l1))*w[h1]*w1[l1]
30
31
32
33 x = np.linspace(0, width, width)
34 y = np.linspace(0, length, length)
35 X,Y = np.meshgrid(x,y)
36
37 fig1 = plt.figure(1)
38 ax1 = fig1.gca(projection='3d')
39 fig2 = plt.figure(2)
40 ax2 = fig2.gca(projection='3d')
41
42 ax1.plot_surface(X,Y,np.transpose(imageRaw), cmap=cm.jet)
43 #ax1.scatter3D(X,Y,imageRaw, cmap=cm.jet)
44 azimuth=90

```

```

45 elevation=90
46 ax1.view_init(azimuth,elevation)
47 ax1.set_xlabel('x')
48 ax1.set_ylabel('y')
49 ax1.set_zlabel('Int')
50
51 im_fft = fftpack.fft2(np.transpose(imageRaw))
52 im_fft = fftpack.fftshift(im_fft)
53
54 FreqCompRows = np.fft.fftfreq(imageRaw.shape[0],d=1)
55 FreqCompCols = np.fft.fftfreq(imageRaw.shape[1],d=1)
56 FreqCompRows = np.fft.fftshift(FreqCompRows)
57 FreqCompCols = np.fft.fftshift(FreqCompCols)
58
59 S,D = np.meshgrid(FreqCompRows, FreqCompCols)
60
61 #ax2.scatter3D(X,Y,np.abs(im_fft), cmap=cm.jet)
62 ax2.plot_surface(S, D, np.abs(im_fft), cmap=cm.jet)
63
64 ax2.view_init(azimuth,elevation)
65 ax2.set_xlabel('kx')
66 ax2.set_ylabel('ky')
67 ax2.set_zlabel('Magnitude')
68
69 Nspec=1
70 plt.figure(3)
71 plt.subplot(111)
72 plt.imshow(np.transpose(imageRaw), cmap="gray", aspect=1.0)
73 plt.colorbar(orientation='vertical', fraction=.1)
74
75 plt.figure(4)
76 plt.subplot(111)
77 plt.imshow(np.abs(im_fft), aspect=Nspec, cmap="gray",
78 extent=[-FreqCompRows[0],FreqCompRows[0],-FreqCompCols[0],FreqCompCols
79 [0]])
79 #plt.imshow(np.abs(im_fft), aspect=1.0, norm=LogNorm(vmin=5))
80 plt.colorbar(orientation='vertical', fraction=.1)
81 plt.show()
82
83 modfft = np.abs(im_fft)
84 maxElement = np.argmax(modfft)
85 iindex, jindex= np.unravel_index(modfft.argmax(),modfft.shape)
86 print( "j=",jindex,"i=" , iindex)
87 kx,ky = FreqCompRows[jindex],FreqCompCols[iindex]
88 print(kx,ky)

```

## ANEXO B – Código para automatização do aparato

### B.1 Funcionamento mecânico do aparato

```
1 /*Declaracoes do motor */
2
3 int wireA=12;
4 int wireB=11;
5 int wireC=10;
6 int wireD=9;
7 int delayi=10;
8 int delayj=10;
9
10 /*Declaracaoe do laser/ldr*/
11
12 const int ldr = A0;
13 int ldrValue = 0;
14
15 /*Declaracoes do servo motor*/
16
17 #include <Servo.h>
18
19 Servo meuservo;
20
21 int angulo = 95;
22 int teta = 20;
23
24 /*Declaracoes do contador de voltas*/
25
26 const int CV = A1;
27 int contavoltas = 0;
28
29 /int cont = 0;
30
31 void setup(){
32
33 Serial.begin(9600);
34 meuservo.attach( 7 );
35 pinMode(wireA, OUTPUT);
36 pinMode(wireB, OUTPUT);
37 pinMode(wireC, OUTPUT);
38 pinMode(wireD, OUTPUT);
39 }
40
41 void loop(){
42
43 int i, nmed=1;
```

```
44 int j;
45 int k;
46
47 while(Serial.available()){
48 char ch = Serial.read();
49 // Serial.println(ch);
50
51 while(ch=='1'){
52 CountGyre();
53 sensor();
54 while (ldrValue>=1021){
55 sensor();
56 motorpassoH();
57 }
58 if(ldrValue<=1019){
59 motorpassoSTOP();
60 delay(500);
61
62 for(k=0; k<=15; k++){
63 motorpassoH();
64 }
65 delay(500);
66
67 meuservo.write(teta);
68 for (i=0; i<=45; i++){
69 motorpassoH();
70 }
71
72 motorpassoSTOP();
73 delay(1000);
74
75 //Serial.print(nmed++);
76
77 //Serial.println("Medida Feita"); momento que a medida seria realizada
78
79 for (j=0; j<40; j++){
80 motorpassoAH();
81 }
82 meuservo.write(angulo);
83 delay(1);
84 for (j=0; i<=100; i++){
85 motorpassoH();
86 }
87 }
88 }
89 }
90 if(ch=='2'){
```



```
91 motorpassoSTOP();
92 }
93 }
94
95 //funcao responsavel pela leitura do do fotoresistor
96
97 void sensor(){
98   ldrValue = analogRead(ldr);
99   Serial.println (ldrValue);
100 }
101
102 //funcao responsavel por girar o motor de passo no sentido horario
103
104 void motorpassoH(){
105   digitalWrite(wireC,HIGH);
106   delay(delayi);
107   digitalWrite(wireC,LOW);
108   digitalWrite(wireA,HIGH);
109   delay(delayi);
110   digitalWrite(wireA,LOW);
111   digitalWrite(wireD,HIGH);
112   delay(delayi);
113   digitalWrite(wireD,LOW);
114   digitalWrite(wireB,HIGH);
115   delay(delayi);
116   digitalWrite(wireB,LOW);
117 }
118
119 //funcao responsavel por parar o motor de passo
120
121 void motorpassoSTOP(){
122   digitalWrite(wireC,LOW);
123   delay(delayi);
124   digitalWrite(wireC,LOW);
125   digitalWrite(wireA,LOW);
126   delay(delayi);
127   digitalWrite(wireA,LOW);
128   digitalWrite(wireD,LOW);
129   delay(delayi);
130   digitalWrite(wireD,LOW);
131   digitalWrite(wireB,LOW);
132   delay(delayi);
133   digitalWrite(wireB,LOW);
134 }
135
136 //funcao responsavel por girar o motor de passo no sentido anti-horario
137
```

```

138 void motorpassoAH(){
139   digitalWrite(wireB,HIGH);
140   delay(delayj);
141   digitalWrite(wireB,LOW);
142   digitalWrite(wireD,HIGH);
143   delay(delayj);
144   digitalWrite(wireD,LOW);
145   digitalWrite(wireA,HIGH);
146   delay(delayj);
147   digitalWrite(wireA,LOW);
148   digitalWrite(wireC,HIGH);
149   delay(delayj);
150   digitalWrite(wireC,LOW);
151 }
152
153 // funcao responsavel pelo controle do contador de voltas
154
155 void CountGyre(){
156
157   contavoltas = analogRead (CV);
158   Serial.print("DATA,TIME,");
159   Serial.print(contavoltas);
160   Serial.print(",");
161   Serial.print(ldrValue);
162   Serial.print(",");
163 }

```

## B.2 Código para realizar as medias

```

1   //defindo funcoes globais
2   enum {
3   ANALOG_INPUT_PIN1 = A0, // ler o sinal do fototransistor
4   ANALOG_INPUT_PIN2 = A1, // ler de saida do sinal do PZT
5   UPDATE_INTERVAL_MICROS = 10000,
6   MODE_2_CONSECUTIVE_READS = 1000,
7   intervalPZT = 1000,
8   };
9
10
11
12   boolean PZTstate = false;
13
14   unsigned long previousMillisPZT=0;
15   unsigned long next_update;
16
17   int reads;
18   int mode;
19

```

```
20 int data;
21 // codigo de configuracao que executa uma vez
22 void setup()
23 {
24   Serial.begin(115200);
25   pinMode(6, OUTPUT);
26   //analogReference(EXTERNAL);
27   mode = 0;
28 }
29
30 // definindo a funcao mode1
31 //Que vai ser chamada no main()
32 //Esse modo le as entradas analogicas e envia pela serial
33 void mode1() {
34
35   mode3();
36
37   int analog_value1 = analogRead(ANALOG_INPUT_PIN1);
38   int analog_value2 = analogRead(ANALOG_INPUT_PIN2);
39   Serial.print(analog_value1);
40   Serial.print(";");
41   Serial.println(analog_value2);
42   delay(10);
43
44 }
45
46 //Definindo a funcao mode2
47 //Ela vai ser chamada la no main()
48 void mode2() {
49
50   mode3();
51
52   //A funcao micros() serve pra pegar o tempo q se passou desde o startup
       do arduino
53
54   unsigned long current_micros;
55   current_micros = micros();
56
57
58   //Este IF so vai ser rodado a cada UPDATE_INTERVAL_MICROS Micros
59   //Sendo assim, o restante do codigo no loop pode rodar varias vezes sem
       que este pedaco de codigo seja
60 executado
61   //Essa e uma forma bastante usual de ter duas tarefas rodando ao mesmo
       tempo, uma com uma frequencia mais
62 alta e uma com uma frequencia mais baixa
63   if(current_micros >= next_update) {
```

```

64 int analog_value1 = analogRead(ANALOG_INPUT_PIN1);
65 int analog_value2 = analogRead(ANALOG_INPUT_PIN2);
66 next_update = next_update + UPDATE_INTERVAL_MICROS;
67 reads--;
68 byte b1, b2 , b3, b4;
69
70 //Fazendo uma operaco E com os bits
71 b1 = analog_value1 & 0xFF;
72 //Aqui alem do E realiza um shift de bits (>>)
73 b2 = ( analog_value1 >> 8 ) & 0xFF;
74
75 b3 = analog_value2 & 0xFF;
76 b4 = ( analog_value2 >> 8 ) & 0xFF;
77 //Serial.println(analog_value1 ,BIN);
78 Serial.write(b1);
79 Serial.write(b2);
80 Serial.write(b3);
81 Serial.write(b4);
82
83 //Serial.print(analog_value1);
84 //Serial.print(";");
85 //Serial.println(analog_value2);
86
87 if(reads <= 0) {
88 mode = 0;
89 Serial.write(0xFF);
90 Serial.write(0xFF);
91 Serial.write(0xFF);
92 Serial.write(0xFF);
93 }
94
95 }
96
97 }
98
99 //Aqui ele ta definindo a funcao mode3
100 //Que vai ser chamada de dentro dos outros modes
101
102 void mode3(){
103
104 //Essa funcao e parecida com o micros() a diferenca e a unidade de
    medida
105 unsigned long currentMillis = millis();
106
107 //Este IF so vai ser rodado a cada intervalPZT millis
108 //Sendo assim, o restante do codigo no loop pode rodar varias vezes sem
    que este pedao de codigo seja

```

```
109 executado
110 //Essa e uma forma bastante usual de ter duas tarefas rodando ao mesmo
    tempo, uma com uma frequencia mais
111 alta e uma com uma frequencia mais baixa
112 if ((unsigned long)(currentMillis - previousMillisPZT) >= intervalPZT)
    {
113 PZTstate = !PZTstate;
114 digitalWrite(6, PZTstate);
115 previousMillisPZT = currentMillis;
116 }
117 }
118 //codigo principal que ser executado repetidas vezes.
119 void loop() {
120
121 //Aqui eespera para ler valores da serial
122 //Dependendo do que vem pela serial ele define o mode
123 if(Serial.available() > 0) {
124
125 int rcv;
126 rcv = Serial.read();
127 //se ele recebe
128 if(rcv == '1') {
129 mode = 1;
130 } else if(rcv == '2') {
131 mode = 2;
132 reads = MODE_2_CONSECUTIVE_READS;
133 next_update = micros();
134 } else if(rcv == '0') {
135 if(mode == 2) {
136 Serial.write(0xFF);
137 Serial.write(0xFF);
138 Serial.write(0xFF);
139 Serial.write(0xFF);
140 }
141 mode = 0;
142 }
143 }
144
145 //Baseado no valor de Mode (que ele pega da serial) ele define o
    comportamento do arduino
146 //Depois chamamos as funcoes mode que criadas
147 //As funcoes mode sao definidas por ele mais pra cima no codigo
148 switch(mode) {
149 case 1:
150 model();
151 break;
152 case 2:
```

```
153 mode2();  
154 break;  
155 default:  
156 delay(200);  
157 break;  
158 }  
159  
160 }
```

## ANEXO C – Código para a coleta e tratamento de dados

## C.1 Código principal

```

1 #####
2 #Importando bibliotecas
3 import matplotlib.pyplot as plt
4 import matplotlib.image as mpimg
5 from mpl_toolkits.mplot3d import Axes3D
6 #from matplotlib.tri import Triangulation
7 import numpy as np
8 import pandas as pd
9 from os import walk
10 from scipy.signal import argrelextrema
11 from scipy import optimize
12 from scipy import fftpack
13 from scipy import ndimage
14 import csv
15 import os
16 from bibliotecas import *
17
18 #####
19 #criando e salvados as pastas onde serao salvos os arquivos
20
21 cmd = "rm -rf Arquivos"
22 returned_value = os.system(cmd) # returns the exit code in unix
23 print('Apagando file Arquivos:', returned_value)
24 cmd = "mkdir Arquivos"
25 returned_value = os.system(cmd) # returns the exit code in unix
26 print('Criando file Arquivos:', returned_value)
27 Nmed=200 #selecionando quando medidas serao realizadas
28 #
29     #####
30 # realizando medidas a partir de uma comunicacao com o Arduino e
31     direcionando aonde os arquivos serao salvos
32 for y in range(Nmed):
33     print(y)
34     coletamedida(name="aquil",direterotio="Arquivos/",portaserial="/dev/
35         ttyACM0")
36 #####
37 #abrindo os arquivos e salvando eles em uma matriz para a geracao da
38     imagem
39 mypath="Arquivos"
40 filenamesdata = []

```

```

39 for (dirpath, dirnames, filenames) in walk(mypath):
40     filenamesdata.extend(filenames)
41     break
42     filenamesdata.sort()
43     analise_medidas(filenamesdata, Nmed, namefile_rawdata='rawdata.dat')
44     namefile='rawdata.dat'
45     imageRaw=readdata(namefile)
46     readfile(filenamesdata)
47     file_name_out='image1.png'
48     plot_image(imageRaw, file_name_out)
49
50     #####
51
52     #separando as matrizes em dois eixos e salvando cada um em uma matriz
        diferente
53     cmd = "rm kxky.dat"
54     returned_value = os.system(cmd)
55     print('returned value:', returned_value)
56     arquivo = open('kxky.dat', 'a')
57     fomart = '%.8f \t %.8f \t %.8f \t %.8f\n'
58     #####
59     Ly_divImage=200 # dividindo a imagem
60     kxdata=[]
61     kydata=[]
62     kxdata_e=[]
63     kydata_e=[]
64
65     #####
66
67     #Crinado as tranformadas de Fourier
68     for i in range ( 0, len(imageRaw), Ly_divImage ):
69         index = i
70         fr,fc,fftim=fourier_plt(imageRaw[i:i+Ly_divImage])
71         kx_m,ky_m,iindex, jindex=maxargmax(fr,fc,fftim)
72         lx,ly=shape(fftim)
73         mask_square=mask(lx,ly,iindex,jindex)
74         kx1,ky1,dkx1,dky1=med_dis_prop(fr,fc,fftim,mask_square)
75         plot_spectrum(fr,fc,fftim,mask_square)
76         plot_spectrum_3d(fr,fc,fftim,mask_square)
77         kxdata.append(kx1)
78         kydata.append(ky1)
79         kxdata_e.append(dkx1)
80         kydata_e.append(dky1)
81         arquivo.write(fomart %(kx1,ky1,dkx1,dky1))
82         arquivo.write("Media kx:%.8f, Std kx:%.8f, Media ky%.8f, Std ky:%.8f" %(
            kx1,ky1,dkx1,dky1) )
83     arquivo.close()

```



```

84 plt.plot(kxdata)
85 plt.plot(kydata)
86 plot_kxky(kxdata,kydata,kxdata_e,kydata_e)
87 plt.show()
88 print(np.average(kydata),np.std(kydata))

```

## C.2 Biblioteca criada

```

1 ##### importando as bibliotecas
2 import matplotlib.pyplot as plt
3 import matplotlib.image as mpimg
4 from mpl_toolkits.mplot3d import Axes3D
5 import numpy as np
6 from matplotlib.colors import LogNorm
7 from matplotlib import cm
8 import serial
9 import struct
10 import csv
11 import codecs
12 from datetime import datetime
13 import time
14 import matplotlib.pyplot as plt
15 import pandas as pd
16 from os import walk
17 from scipy.signal import argrelextrema
18 from scipy import optimize
19
20 #####
21
22 #comunicacao para salvar os dados do Arduino
23 #Aqui pode escolher o nome dos arquivos e em qual pasta ele sera salvo
24 def
25 coletamedida(name="h2o-h2otampado",direterotio="Arquivos/",portaserial="
    /dev/ttyACM0",baudrate=115200)
26 :
27 max_valid_read = 1023
28 reference_voltage = 5.0
29 update_interval = 1e-2
30 now = datetime.now()
31 year=str(now.year)
32 month=str(now.month)
33 day=str(now.day)
34 hour=str(now.hour)
35 minute=str(now.minute)
36 second=str(now.second)
37 filename=direterotio+name+year+"_"+month+"_"+day+"_"+hour+"-"+minute+"-"
    +second+".csv"
38 ser = serial.Serial( portaserial , baudrate )

```

```

39 time.sleep(1.5)
40 ser.flush()
41 ser.write('2'.encode())
42 r = 0
43 r1 = 0
44 r_time = 0
45 analog_reads = [];
46 analog_reads1 = [];
47 analog_times = [];
48 with open(filename, 'w') as saida:
49     escrever = csv.writer(saida)
50     i=0
51     data=[]
52     for i in range(6):
53         while r <= max_valid_read:
54             b1 = ord(ser.read(1))
55             b2 = ord(ser.read(1))
56             b3 = ord(ser.read(1))
57             b4 = ord(ser.read(1))
58             r = b1 + b2*256
59             r1 = b3 + b4*256
60             #print(r,r1)
61             if(r <= max_valid_read):
62                 analog_reads.append((r*reference_voltage)/max_valid_read)
63                 analog_reads1.append((r1*reference_voltage)/max_valid_read)
64                 analog_times.append(r_time)
65                 r_time = r_time + update_interval
66                 escrever.writerow([r,r1])
67                 data.append([r,r1])
68             ser.close()
69         return data
70
71 #####
72
73 #Import data using pandas
74 #####
75
76 #funcao responsavel pela leitura dos arquivos salvos
77 def read_sep_plot(filename):
78     column_names = ['ldr', 'rampa']
79     dataset_path="Arquivos/"+filename
80     raw_dataset = pd.read_csv(dataset_path, header=None, sep=',', skiprows=0)
81     raw_dataset.columns = column_names
82     dataset = raw_dataset[: -1].copy()
83     return dataset
84
85 #####

```

```

86
87 #Filtros gaussiano para tratar os dados
88 def gaussian_kernel(size):
89     size = int(size)
90     x = np.mgrid[-size:size+1]
91     g = np.exp(-(x**2/float(size)))
92     return g / g.sum()
93
94 #####
95
96 #Filtros passa baixa para remover frequencias altas
97 def LowpassFilter(fc=0.05):
98     b = 0.08
99     N = int(np.ceil((4 / b)))
100    if not N % 2: N += 1
101    n = np.arange(N)
102    sinc_func = np.sinc(2 * fc * (n - (N - 1) / 2.))
103    window = 0.42 - 0.5 * np.cos(2 * np.pi * n / (N - 1)) + 0.08 * np.cos(4
        * np.pi * n / (N - 1))
104    sinc_func = sinc_func * window
105    sinc_func = sinc_func / np.sum(sinc_func)
106    return sinc_func
107 #####
108 Reta de ajuste
109 def lin_func(x, a, b):
110     f=a+b*x
111     return f
112
113 #####
114
115 #funcao que utiliza os filtros definidos acima para recolher apenas as
    os padroes realacionado a subida da rampa
116 def analise_medidas(filenameesdata, Nmed, namefile_rawdata='rawdata.dat'):
117     gaussian_kernel_array = gaussian_kernel(size=10)
118     sinc_func=LowpassFilter(fc = 0.05)
119     rawdata=[]
120     for i in filenameesdata[0:Nmed]:
121         data=read_sep_plot(i)
122         max_rampdata=data['rampa'].max()
123         min_rampdata=data['rampa'].min()
124         data['rampa']=(data['rampa']-min_rampdata)/max_rampdata
125         conv_ldrdata_sinc = np.convolve(data['rampa'], sinc_func, 'same')
126         maximum=argrelextrema(conv_ldrdata_sinc, np.greater)
127         x,y=[], []
128         for maxx in maximum[0]:
129             if (conv_ldrdata_sinc[maxx] > 0.5):
130                 x.append(maxx)

```

```

131 y.append(conv_ldrdata_sinc[maxx])
132 data['ldr']=(data['ldr']-data['ldr'].min())/data['ldr'].max()
133 data['ldr']=(data['ldr']-data['ldr'].max()*0.5)/(data['ldr'].max()*0.5)
134 for j in range(len(x)):
135 x_data,y_data=[],[]
136 fin=x[j]
137 ini=x[j]-50
138 for i1 in range(ini,fin):
139 x_data.append(i1)
140 y_data.append(conv_ldrdata_sinc[i1])
141 x_data=np.asarray(x_data)
142 y_data=np.asarray(y_data)
143 params, params_covariance = optimize.curve_fit(lin_func, x_data, y_data,
144 p0=[2, 2])
145 x_data1,y_data1=[],[]
146 ini=int(-params[0]/params[1])
147 for i1 in range(ini,fin):
148 x_data1.append(i1)
149 y_data1.append(data['ldr'][i1])
150 x_data1=np.asarray(x_data1)
151 y_data1=np.asarray(y_data1)
152 rawdata.append(y_data1)
153 X = range(len(conv_ldrdata_sinc))
154 plt.plot(X, conv_ldrdata_sinc, x_data, lin_func(x_data,*params),data['
155 ldr'])
156 plt.plot(X, conv_ldrdata_sinc, x_data, lin_func(x_data,*params),x_data,
157 data['ldr'][x_data[0]:x_data[-1]+1])
158 plt.show()
159 #####
160
161 # Aqui para frente s o as fun es respons veis pela cria o da
162 imagem as fun es realizam #a FFT
163 print(len(rawdata))
164 width=len(rawdata)
165 dimmm_min=100000
166 for k1 in range(width):
167 dim=len(rawdata[k1])
168 if(dim<dimmm_min):
169 dimmm_min=dim
170 length=dimmm_min
171 file_raw = open(namefile_rawdata,'w')
172 rawformat='{0:4f} '
173 rawformat1='\n'
174 imageRaw = np.zeros((width, length))

```

```

175 for k1 in range(width):
176 dim=len(rawdata[k1])
177 for l1 in range(length):
178 file_raw.write(rawformat.format(rawdata[k1][l1]))
179 imageRaw[k1][l1]=rawdata[k1][l1]
180 file_raw.write(rawformat1)
181 file_raw.close()
182
183 #####
184
185 def readdata(namefile):
186 file_raw = open(namefile,'r')
187 rawdata=[]
188 i=0
189 width_min=10000
190 for line in file_raw:
191 i=i+1
192 if (len(line)<width_min):
193 width_min=len(line)
194 line1=line.replace("\n","0.00")
195 linedata=line1.split(' ')
196 rawdata.append(linedata)
197 width,length=np.shape(rawdata)
198 width=int(width)
199 imageRaw = np.zeros((width, length))
200 for yii in range(width):
201 for xii in range(length):
202 imageRaw[yii][xii]=np.float(rawdata[yii][xii])
203 return imageRaw
204 def shape(imageRaw):
205 return np.shape(imageRaw)
206 def mask(lx,ly,Nx,Ny):
207 mask1=np.zeros((lx,ly))
208 print(Nx,Ny)
209 mask1[Nx-20:Nx+20,Ny-5:Ny+5]=1.0
210 plt.imshow(mask1, cmap="gray")
211 plt.show()
212 exit()
213 return mask1
214 def plot_image(imageRaw,file_name_out):
215 savename="Interferometric Image"
216 fig=plt.figure()
217 plt.title('Imagem interferometrica          gua          ')
218 plt.xlabel('t1(s)')
219 plt.ylabel('t2(s)')
220 plt.imshow(imageRaw, cmap="gray")
221 plt.axes().set_aspect(0.05)

```

```

222 fig.savefig(savename, format="png")
223 plt.show()
224 fig.clf()
225 def fourier_plt(im):
226 from scipy import fftpack
227 im_fft = fftpack.fft2(im)
228 im_fft = fftpack.fftshift(im_fft)
229 FreqCompRows = np.fft.fftfreq(im.shape[0],d=1)
230 FreqCompCols = np.fft.fftfreq(im.shape[1],d=1)
231 FreqCompRows = np.fft.fftshift(FreqCompRows)
232 FreqCompCols = np.fft.fftshift(FreqCompCols)
233 return FreqCompRows, FreqCompCols, im_fft
234
235 #####
236
237 def plot_spectrum(FreqCompRows, FreqCompCols, im_fft, mask):
238 modfft= np.abs(im_fft)
239 dkx=FreqCompRows[1]-FreqCompRows[0]
240 dky=FreqCompCols[1]-FreqCompCols[0]
241 norm=0.0
242 for i in range(len(FreqCompRows)):
243 for j in range(len(FreqCompCols)):
244 norm=norm+modfft[i][j]*dkx*dky
245 Prob_modfftnorm= mask*modfft/norm
246 savename="Tranformada de Fourier"
247 fig=plt.figure()
248 plt.imshow(np.abs(im_fft), norm=LogNorm(vmin=5),
249 extent=[FreqCompRows[0],-FreqCompRows[0],FreqCompCols[0],-FreqCompCols
250 [0]])
250 plt.imshow(Prob_modfftnorm,
251 extent=[FreqCompRows[0],-FreqCompRows[0],FreqCompCols[0],-FreqCompCols
252 [0]], cmap='gray')
252 plt.title('Fourier transform')
253 plt.xlabel('u(1/s)')
254 plt.ylabel('v(1/s)')
255 plt.colorbar()
256 plt.show()
257 fig.savefig(savename, format="png")
258 def plot_spectrum_3d(FreqCompRows, FreqCompCols, im_fft, mask):
259 savename="Tranformada em 3D"
260 fig=plt.figure()
261 ax2 = fig.gca(projection='3d')
262 S,D = np.meshgrid(FreqCompRows, FreqCompCols)
263 modfft= np.abs(im_fft)
264 dkx=FreqCompRows[1]-FreqCompRows[0]
265 dky=FreqCompCols[1]-FreqCompCols[0]
266 norm=0.0

```

```

267 for i in range(len(FreqCompRows)):
268 for j in range(len(FreqCompCols)):
269 norm=norm+modfft[i][j]*dkx*dky
270 Prob_modfftnorm= mask*modfft/norm
271 z = np.transpose(Prob_modfftnorm)
272 ax2.plot_surface(S, D, np.transpose(Prob_modfftnorm))
273 )
274 plt.title('Tranformade de Fourier em 3D          gua          ')
275 plt.xlabel('u(1/s)')
276 plt.ylabel('v(1/s)')
277 ax2.set_zlabel('Magnitude')
278 plt.show()
279 fig.savefig(savename, format="png")
280
281 #####
282
283 #funcoes responsaveis pela criacao analise dos resultados obtidos
284 def maxargmax(FreqCompRows, FreqCompCols, im_fft):
285 modfft= np.abs(im_fft)
286 maxElement = np.argmax(modfft)
287 iindex, jindex= np.unravel_index(modfft.argmax(), modfft.shape)
288 kx, ky = FreqCompRows[iindex], FreqCompCols[jindex]
289 return kx, ky, iindex, jindex
290 def save(imageRaw, y):
291 savename= + "_x={}_y={}_from={}_to={}.png".format(imageRaw, y, index, index
+200)
292 fig.savefig(savename)
293 def med_in_max(kx, ky, FreqCompRows, FreqCompCols, im_fft, Nmed):
294 kxmed=0.0
295 kymed=0.0
296 kxmed2=0.0
297 kymed2=0.0
298 modfft= np.abs(im_fft)
299 kxmin=kx-Nmed
300 kxmax=kx+Nmed
301 kymin=ky-Nmed
302 kymax=ky+Nmed
303 aux=0
304 for i in range(kxmin, kxmax):
305 kxmed=kxmed+modfft[i][ky]
306 kxmed2=kxmed2+modfft[i][ky]**2
307 aux=aux+1
308 kxmed=kxmed/(aux)
309 kxmed2=kxmed2/(aux)
310 aux=0
311 for j in range(kymin, kymax):
312 kymed=kymed+modfft[kx][j]

```

```

313 kymed2=kymed2+modfft[kx][j]**2
314 aux=aux+1
315 kymed=kymed/(aux)
316 kymed2=kymed2/(aux)
317 varkx=np.sqrt(kxmed2-kxmed**2)
318 varky=np.sqrt(kymed2-kymed**2)
319 return kxmed,kymed,varkx,varky
320 def med_dis_prop(FreqCompRows,FreqCompCols,im_fft,mask):
321 modfft= mask*np.abs(im_fft)
322 dkx=FreqCompRows[1]-FreqCompRows[0]
323 dky=FreqCompCols[1]-FreqCompCols[0]
324 norm=0.0
325 for i in range(len(FreqCompRows)):
326 for j in range(len(FreqCompCols)):
327 norm=norm+modfft[i][j]*dkx*dky
328 Prob_modfftnorm = mask*modfft/norm
329 kxmed=0.0
330 kymed=0.0
331 desvx=0.0
332 desvy=0.0
333 for i in range(len(FreqCompRows)):
334 for j in range(len(FreqCompCols)):
335 probdk=mask[i,j]*Prob_modfftnorm[i][j]*dkx*dky
336 kxmed=kxmed+FreqCompRows[i]*probdk
337 kymed=kymed+FreqCompCols[j]*probdk
338 for i in range(len(FreqCompRows)):
339 for j in range(len(FreqCompCols)):
340 probdk=mask[i,j]*Prob_modfftnorm[i][j]*dkx*dky
341 desvx=desvx+np.abs(FreqCompRows[i]-kxmed)*probdk
342 desvy=desvy+np.abs(FreqCompCols[j]-kymed)*probdk
343 print("kxmed,kymed=",kxmed,kymed)
344 print("desvx,desvy=",desvx,desvy)
345 return kxmed,kymed,desvx,desvy
346
347 #####
348
349 def plot_kxky(kx,ky,kx_e,ky_e):
350 color = 'tab:red'
351 color1 = 'tab:blue'
352 savename="Evolution of fx and fy in time"
353 fig=plt.figure()
354 x = np.arange(0, len(kx))
355 plt.errorbar(x, kx, color=color, yerr=kx_e, linestyle='-', ecolor='g',
356 marker="*")
357 plt.errorbar(x, ky, yerr=ky_e, linestyle='-', ecolor='g', marker="*")
358 plt.title('Evolu o de u e v ao longo do tempo')
359 plt.show()

```



```
359 fig.savefig(savename, format="png")
360
361 #####
```

**ANEXO D – Circuito**

