

**INSTITUTO FEDERAL**

Minas Gerais

Campus Betim

# APRESENTAÇÃO DO TRABALHO FINAL

**Disciplina:** Processamento de Sinais

**Prof.:** Leandro Freitas

**Série/Semestre:** 6° / 2022-2

**Data:** 13 / 12 / 2022

**Curso:** Engenharia de Controle e Automação

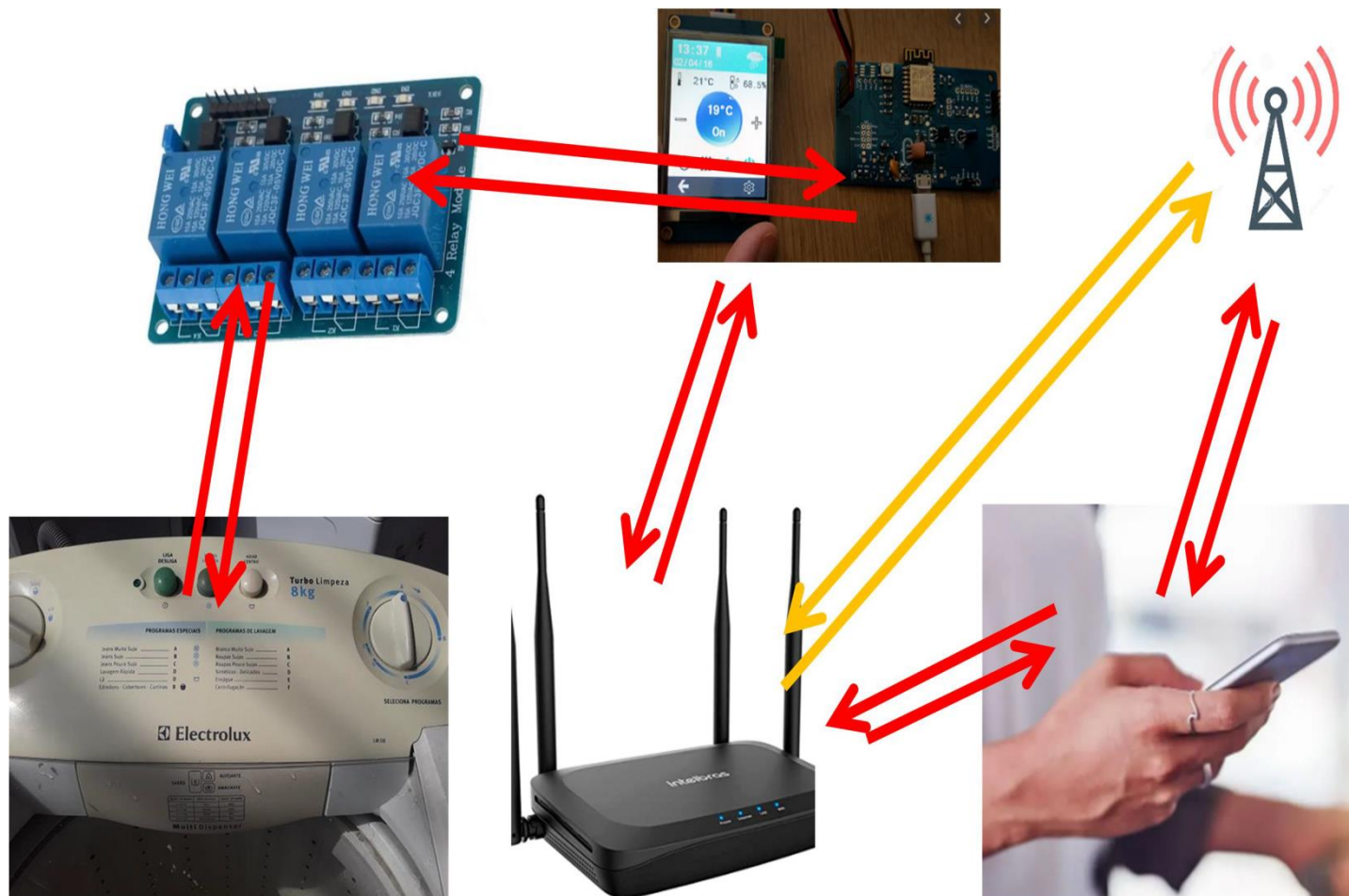
**Nome da aluna:** Alice Nascimento

Registro Acadêmico:						
0	0	2	9	5	0	9

**Nome do aluno:** Lúcio Chagas

Registro Acadêmico:						
0	0	3	3	0	4	7

## TRABALHO DE CONCLUSÃO DE CURSO



# PROJETO TCC

**OBJETIVO:** Fazer um protótipo para atualizar tecnologicamente as máquinas de lavar roupas;

**ESPECÍFICO 1:** Construção de equipamentos eletroeletrônicos com tecnologia atualizada para retrofit em máquinas de lavar com vários anos de utilização de qualquer marca e modelo;

**ESPECÍFICO 2:** Controlar, **monitorar** e parametrizar a máquina de lavar roupas utilizando uma IHM fixa na máquina e/ou aplicativo no celular de qualquer distância conectados por redes de computador (Wifi);

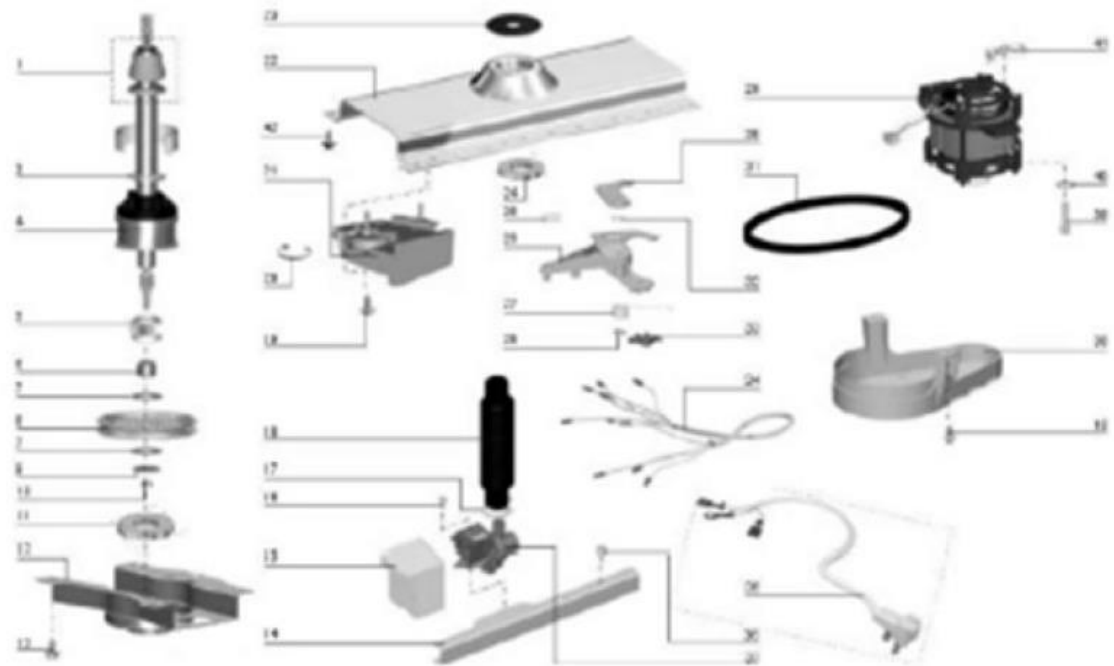
## MÁQUINA DE LAVAR ROUPAS

### VISTA EXPLODIDA MECANISMO



Fonte: Electrolux Manual Lm09

Conjunto Motor



## TANQUINHO DE LAVAR ROUPAS





## EQUIPAMENTOS UTILIZADOS



CODIGO\_ARDUINO | Arduino 1.8.18

Arquivo Editar Sketch Ferramentas Ajuda

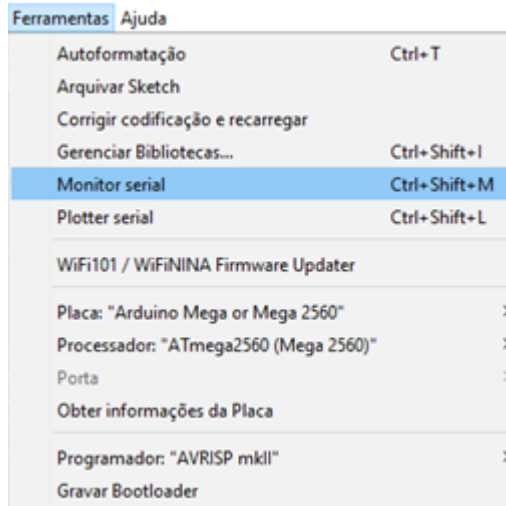


```
int original; // recebe o sinal do sensor de Vibração
unsigned long clocktime;
```

```
void setup() {
  Serial.begin(115200);
}
```

```
void loop() {
  original=analogRead(A0);
  clocktime=millis();
  Serial.print(clocktime);
  Serial.print(",");
  Serial.println(original);

  delay(10);
}
```



MOTOR\_DANIFICADO\_TESTE - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

Millis, Leituras

0,248  
10,306  
20,187  
30,301  
41,241  
52,146  
62,257  
73,124  
83,86  
94,190  
104,75  
115,199  
125,164  
136,78  
147,194  
157,76  
167,88  
178,183  
189,65  
199,186  
209,169  
221,34  
231,158  
241,44  
252,49  
263,152  
273,32  
283,158  
294,142  
305,36  
315,159  
326,51  
336,47  
347,156  
357,31  
368,151  
378,144

**CSV** é um formato de arquivo que significa “comma-separated-values” (**valores separados por vírgulas**);


É um tipo de arquivo de texto fundamental para transferência de informações entre aplicativos diferentes.

- CÓDIGO ARDUÍNO
- INSTALAÇÃO E MONTAGEM
- MONITORAMENTO ON LINE
- LEVANTAMENTO DOS DADOS





# INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS – IFMG/BETIM

 Search or jump to... [Pull requests](#) [Issues](#) [Codespaces](#) [Marketplace](#) [Explore](#)

lucio-paulo / [Processamento\\_Trabalho\\_Final\\_AliceLucio](#) Public [Pin](#) [Unwatch](#) 1 [Fork](#) 0 [Star](#) 0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

main 1 branch 0 tags

[Go to file](#) [Add file](#) [Code](#)

lucio-paulo Delete TESTE.csv 280c2b3 17 hours ago 34 commits		
.gitattributes	Initial commit	7 days ago
MOTOR_BOM_ACOPLADO.csv	Rename MBSA.csv to MOTOR_BOM_ACOPLADO.csv	19 hours ago
MOTOR_BOM_COM_AGITADOR.CSV	Add files via upload	20 hours ago
MOTOR_BOM_DESACOPADO.csv	Rename MBCA.csv to MOTOR_BOM_DESACOPADO.csv	19 hours ago
MOTOR_BOM_SEM_AGITADOR.CSV	Add files via upload	20 hours ago
MOTOR_BOM_TESTE.csv	Rename MOTOR_BOM_TESTE.csv to MOTOR_BOM_TESTe.csv	17 hours ago
MOTOR_DANIFICADO_COM_AGITAD...	Add files via upload	20 hours ago
MOTOR_DANIFICADO_SEM_AGITAD...	Add files via upload	20 hours ago
MOTOR_DANIFICADO_TESTE.csv	Add files via upload	20 hours ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

About

No description, website, or topics provided.

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Contributors 2

lucio-paulo

AliceMarquesIFMG

## IMPORTAÇÃO DAS BIBLIOTECAS E FUNÇÃO FFT

```
[27] #####
# Importação das Bibliotecas
import numpy as np, scipy as sp, matplotlib.pyplot as plt
from scipy import signal
from scipy.signal.fir_filter_design import fft
import pandas as pd, numpy as np
import requests
%matplotlib inline
#####
#FUNÇÃO FFT
def fourier(x, t, Hz=False, dB=False, deg=False, Nfft=None):

    N = len(x)
    T = t[1] - t[0]

    if Nfft==None:
        Nfft = N

    X_f = fft(x, n=Nfft)
    X_f = X_f/N
    X_f = X_f[:Nfft//2]
    X_f[1:] = 2*X_f[1:]

    if Hz==True:
        f0 = 1/(Nfft*T)
        fHz = np.arange(Nfft)*f0
        f = fHz[:Nfft//2]
    else:
        w0 = 2*np.pi/(Nfft*T)
        wRad = np.arange(Nfft)*w0
        f = wRad[:Nfft//2]

    if dB==False:
        mag = np.abs(X_f)

    else:
        mag = 20*np.log10(np.abs(X_f))

    if deg==False:
        pha = np.angle(X_f, deg=0)
    else:
        pha = np.angle(X_f, deg=1)
```

## 1. (TESTE EM BANCADA) MOTOR EM BOM ESTADO E DANIFICADO

```
#####
# ENDEREÇAMENTO DOS DADOS MEDIDOS
dadosteste= r'https://raw.githubusercontent.com/lucio-paulo/Processamento_Trabalho_Fin
dados = pd.read_csv(dadosteste)

# DECLARAÇÃO DAS VARIÁVEIS
Millis = dados['Millis'].to_numpy()
Leituras = dados['Leituras'].to_numpy()

# GRÁFICO DOMÍNIO DO TEMPO DO SINAL MEDIDO
plt.figure(figsize=(13,8))
plt.subplot(221)
plt.plot(Millis,Leituras,label="Sinal",color = "g")
plt.title("MOTOR BOM TESTE")
plt.xlabel("$Tempo$ [ms]")
plt.ylabel("$Vibração$")
plt.legend(fontsize="18")
plt.xlim(0,10000)

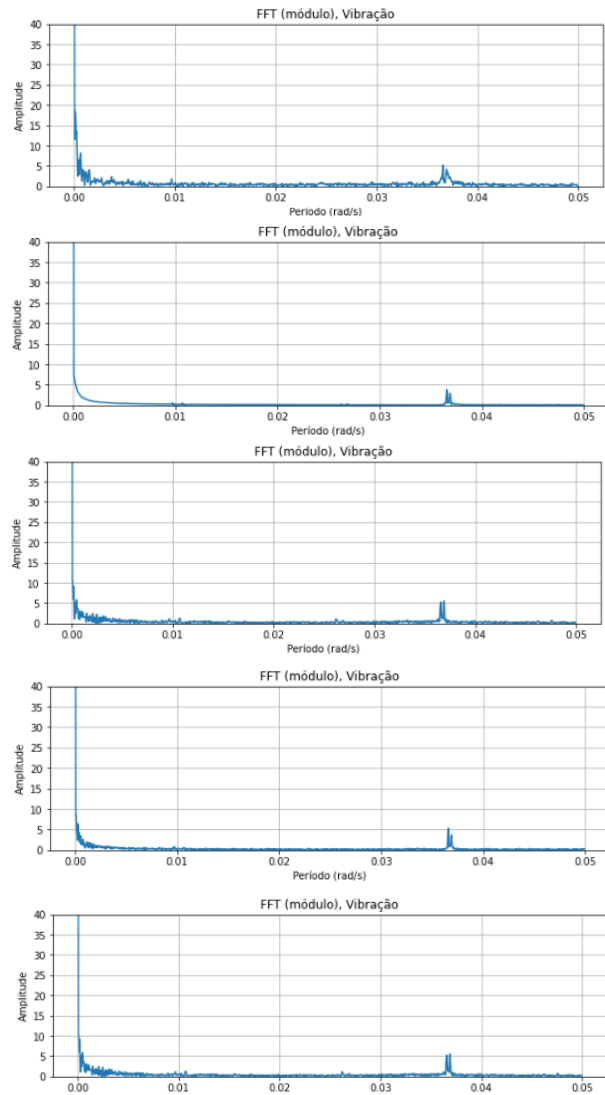
#####
# GRÁFICO DOMÍNIO DA FREQUÊNCIA
Leituras = Leituras.reshape(-1)
mag, pha, f = fourier(Leituras, Millis, Hz=True, dB=False, deg=False, Nfft=None)
plt.subplot(223)
plt.plot(f, mag)
plt.title("FFT (módulo), Vibração")
plt.xlabel("Período (rad/ms)")
plt.ylabel("Amplitude")
plt.ylim(0,20)
plt.grid(1);

#####
# ENDEREÇAMENTO DOS DADOS MEDIDOS
dadosteste= r'https://raw.githubusercontent.com/lucio-paulo/Processamento_Trabalho_Fin
dados = pd.read_csv(dadosteste)

# DECLARAÇÃO DAS VARIÁVEIS
Millis = dados['Millis'].to_numpy()
Leituras = dados['Leituras'].to_numpy()

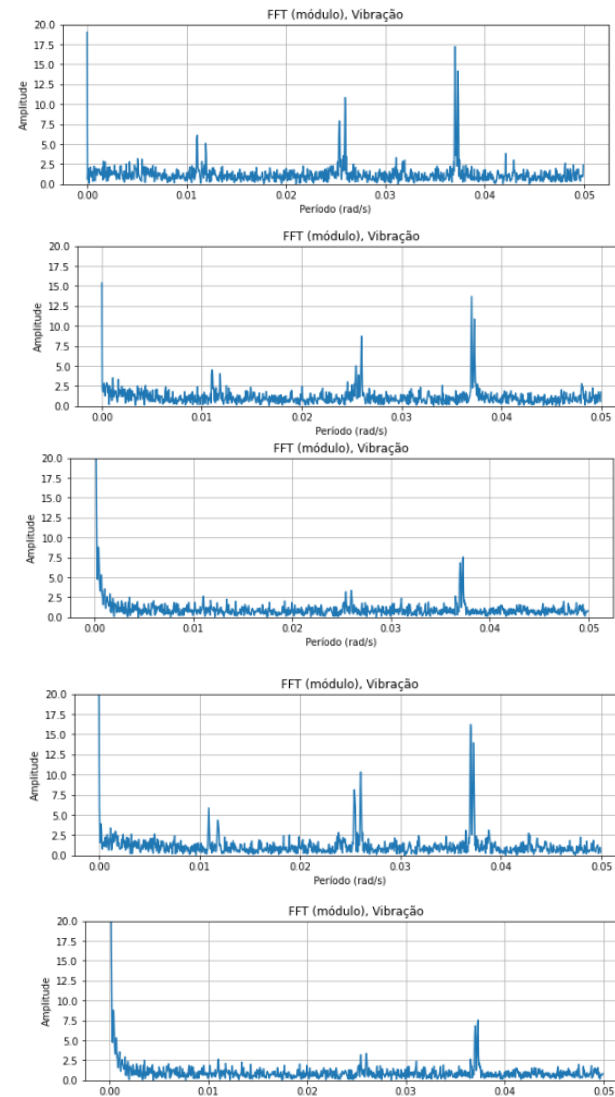
# GRÁFICO DOMÍNIO DO TEMPO DO SINAL MEDIDO
plt.subplot(222)
plt.title("MOTOR DANIFICADO TESTE",color = "r",fontsize="18")
plt.xlabel("$Tempo$ [ms]")
```

## MOTOR BOM



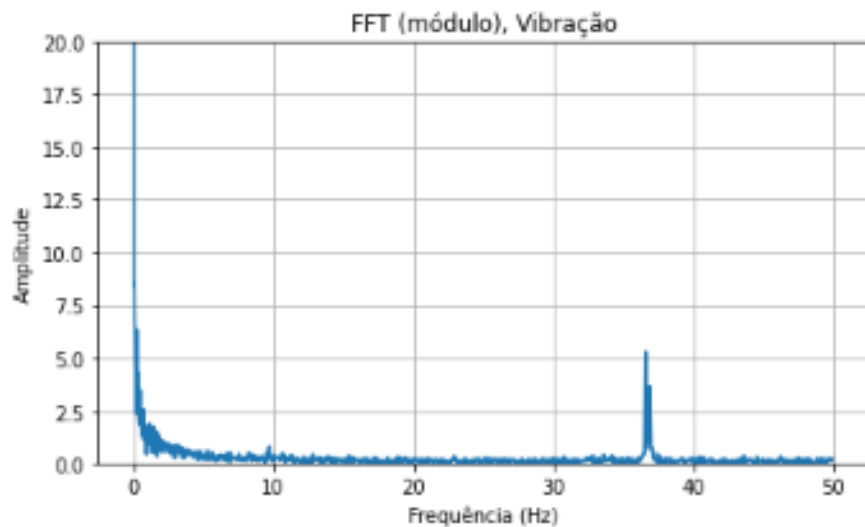
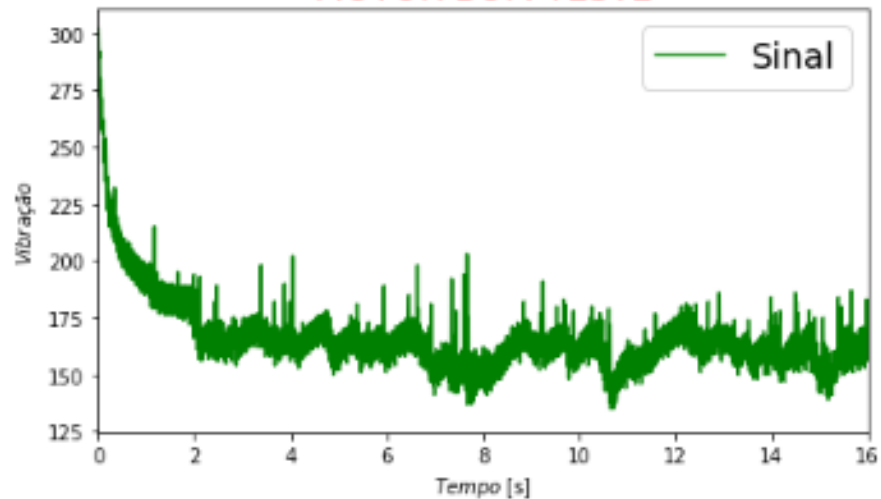
FREQUÊNCIA (KHz)

## MOTOR DANIFICADO

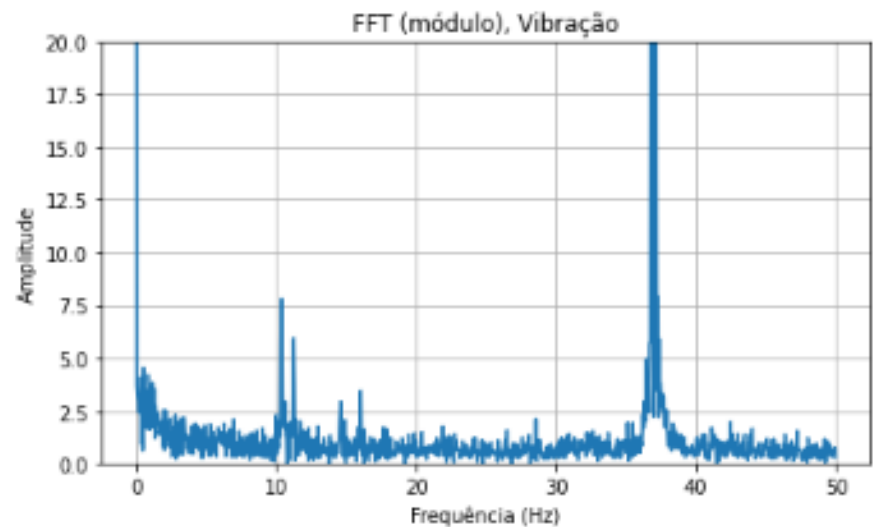
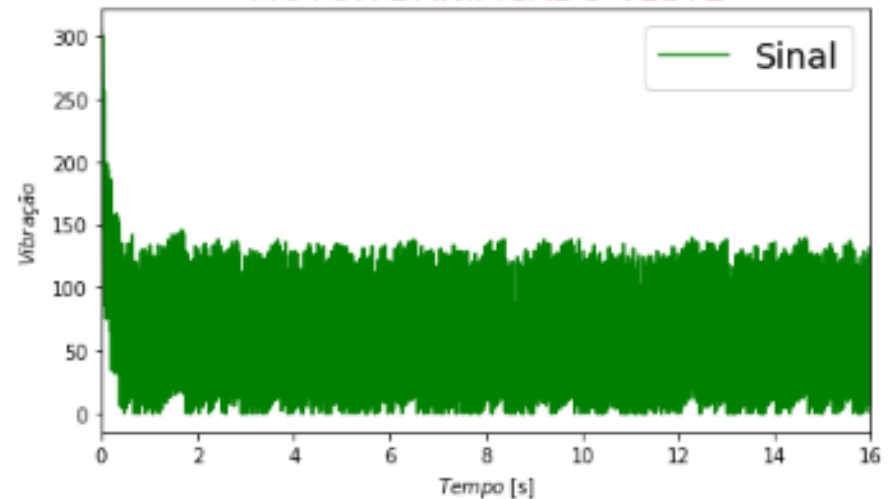


FREQUÊNCIA (KHz)

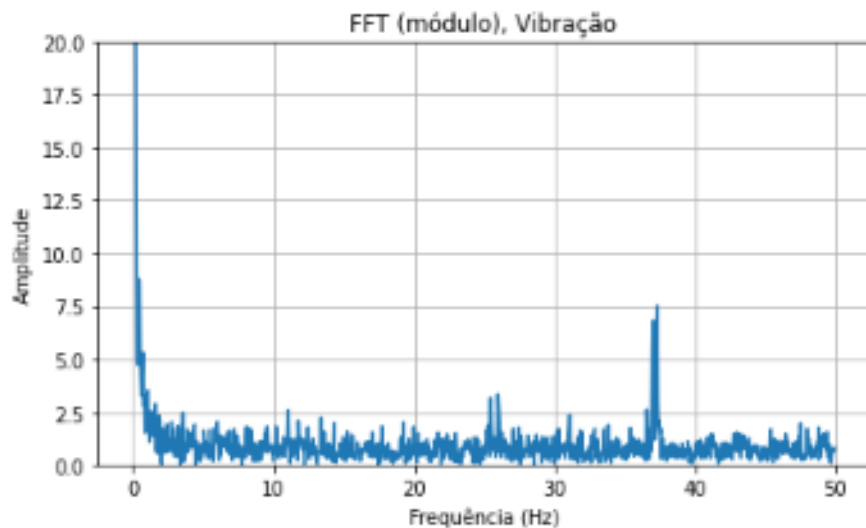
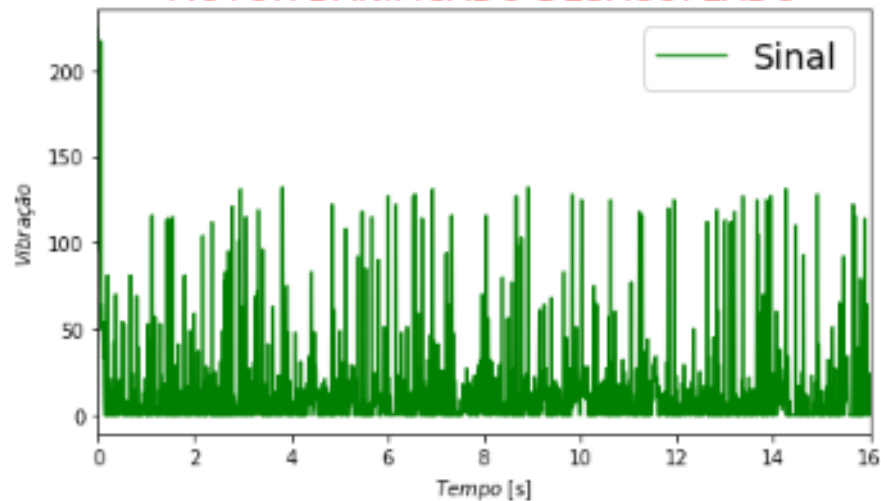
MOTOR BOM TESTE



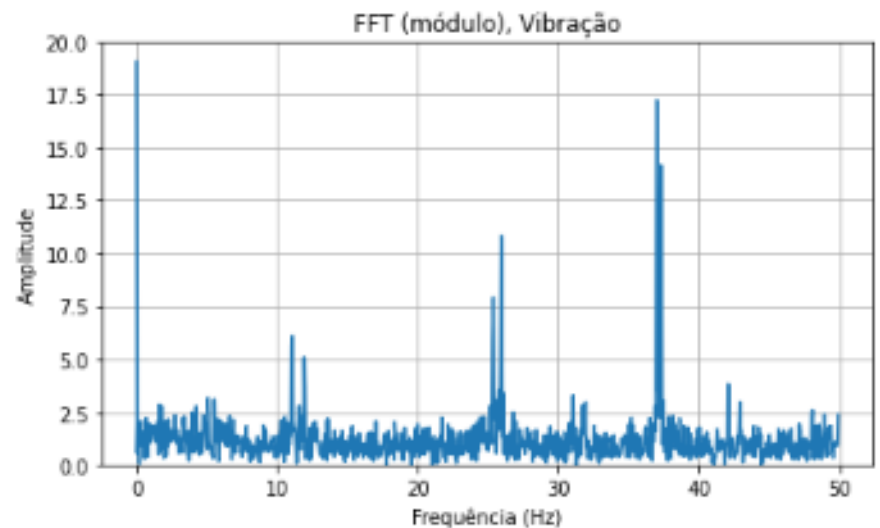
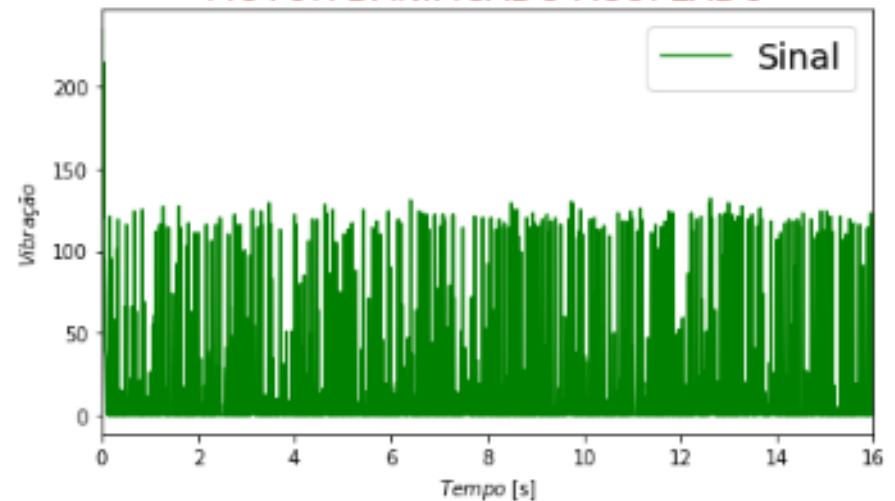
MOTOR DANIFICADO TESTE



MOTOR DANIFICADO DESACOPLADO

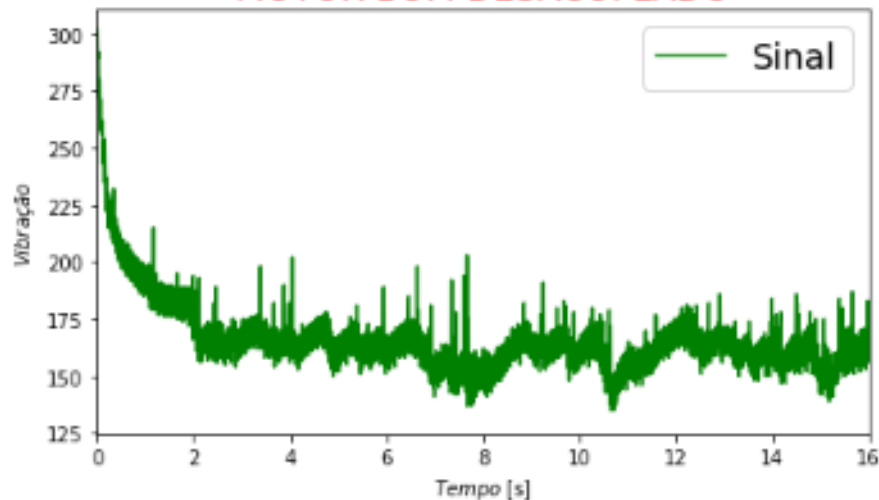


MOTOR DANIFICADO ACOPLADO

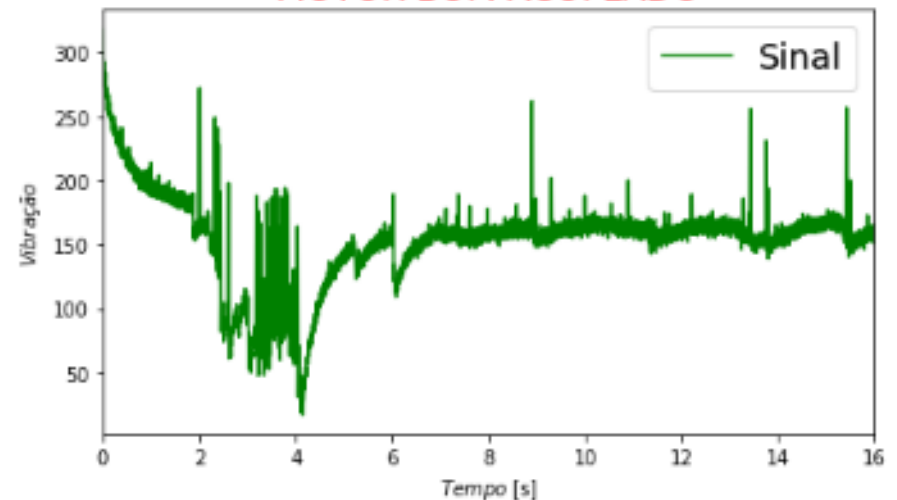




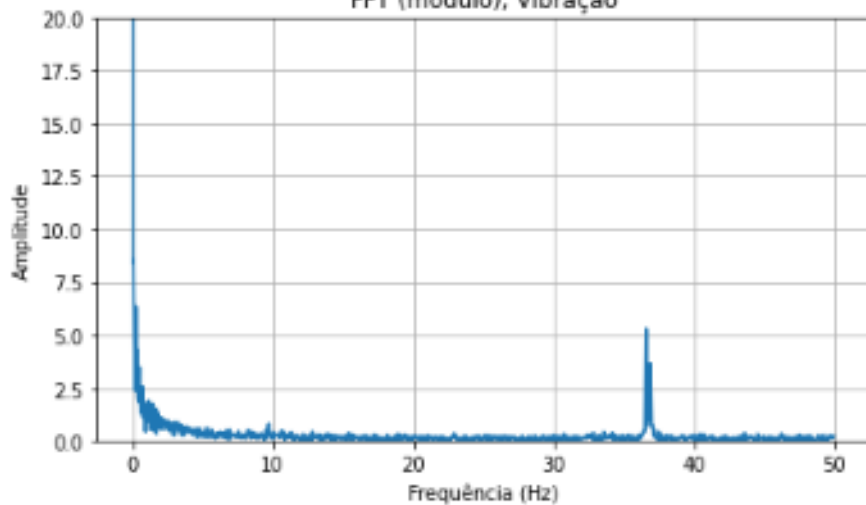
MOTOR BOM DESACOPLADO



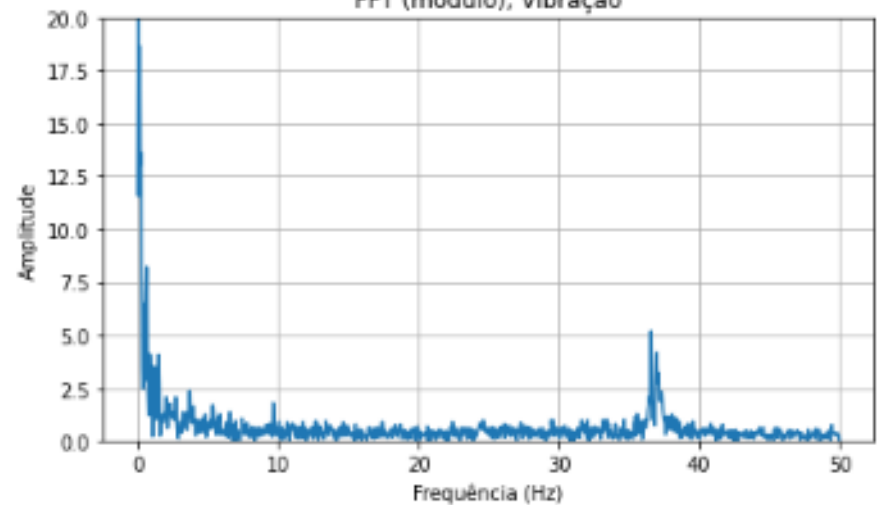
MOTOR BOM ACOPLADO



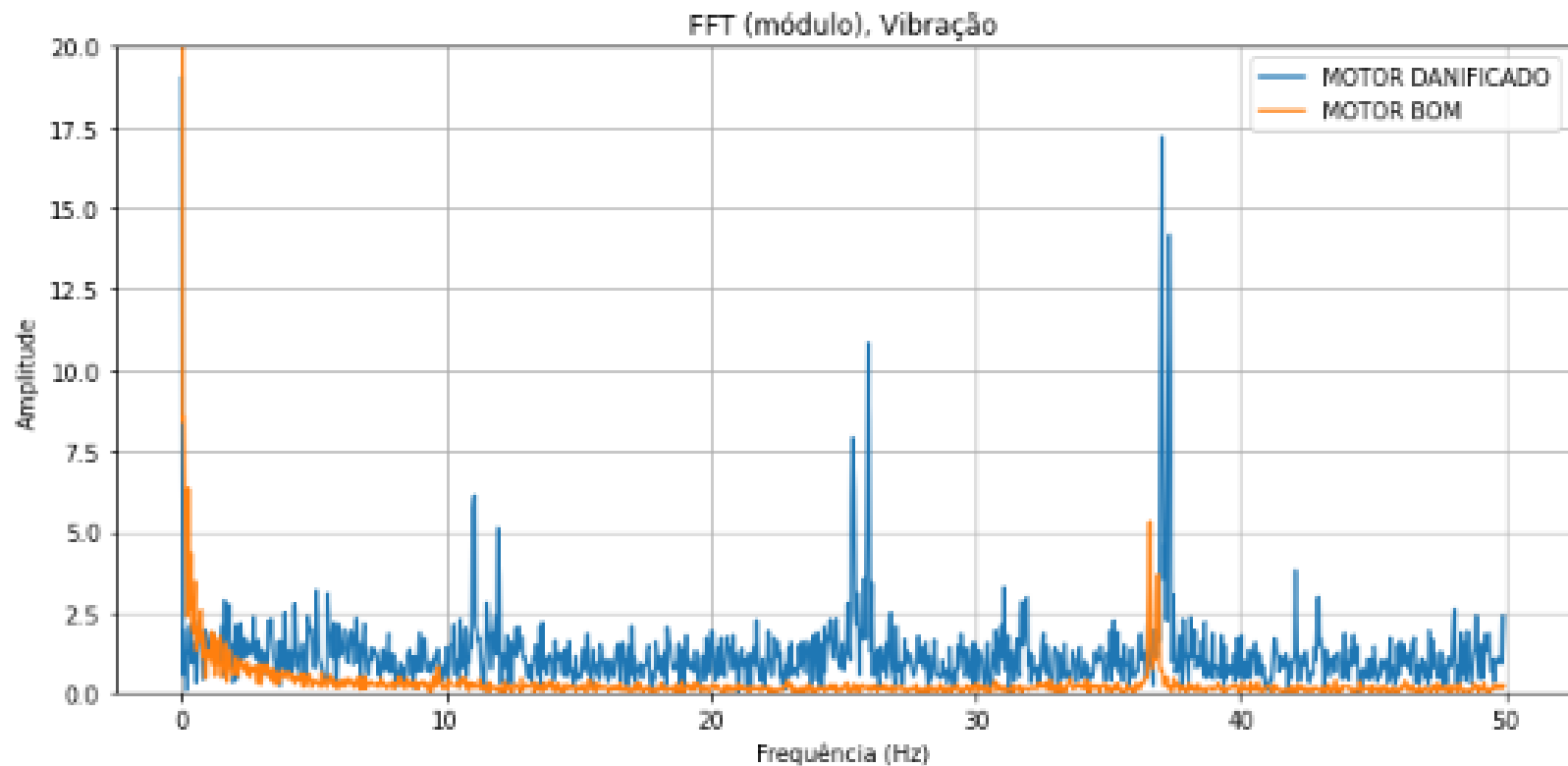
FFT (módulo), Vibração



FFT (módulo), Vibração



## GRÁFICO DOS MOTORES BOM E DANIFICADO NO DOMÍNIO DA FREQUÊNCIA



## 5. PROJETO FILTRO IIR DETECÇÃO DA FALHA

```
from os import WIFSTOPPED
#####
# ESPECIFICAÇÃO DO FILTRO
Ap = 0.5 # Atenuação máxima dB na faixa de passagem
Ar = 65 # Atenuação mínima dB na faixa de rejeição
Gp = -Ap # Ganho mínimo dB na faixa de passagem
Gr = -Ar # Ganho máximo dB na faixa de rejeição

wp1 = 34 # Frequência banda de passagem (Hz)
wr1 = 36 # Frequência banda de rejeição (Hz)
wr2 = 38 # Frequência banda de rejeição (Hz)
wp2 = 40 # Frequência banda de passagem (Hz)
ws = 100 # Frequência de amostragem (Hz)

wp1a = 2*np.pi*wp1 # Frequência banda de passagem (rad/s)
wr1a = 2*np.pi*wr1 # Frequência banda de rejeição (rad/s)
wr2a = 2*np.pi*wr2 # Frequência banda de rejeição (rad/s)
wp2a = 2*np.pi*wp2 # Frequência banda de passagem (rad/s)
wsa = 2*np.pi*ws # Frequência banda de passagem (rad/s)

T = 2*np.pi/ws # seg. período de amostragem
Ta = 1/wsa # seg. período de amostragem

# Ondulações nas bandas de passagem e rejeição
dp = 1-10**(Gp/20) # ondulação na faixa de passagem
dr = 10**(Gr/20) # ondulação na faixa de rejeição
```

```
#####
#3 PROJETO DO FILTRO IIR

z1, p1, k1 = signal.iirdesign(wp=[wp1/(ws/2), wp2/(ws/2)],
                             ws=[wr1/(ws/2), wr2/(ws/2)], gpass=Ap,
                             gstop=Ar, analog=False, ftype='butter',
                             output='zpk')
sys1=signal.dlti(z1, p1, k1, dt=1/ws)

w = np.logspace(2, 3, 1000)
w1, mag1, phase1 = signal.dboe(sys1)

#w1 = w1*ws/2/np.pi
#w1 = w1*(ws/2)

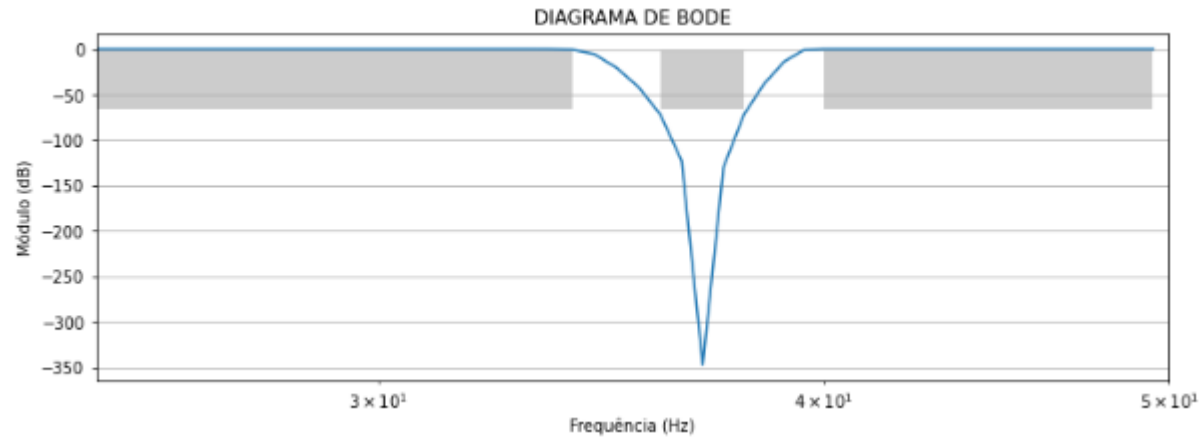
# Transformação para conferir as frequências analógicas
w1= w1/2/np.pi

plt.figure(figsize=(12,4))
plt.semilogx(w1, mag1);

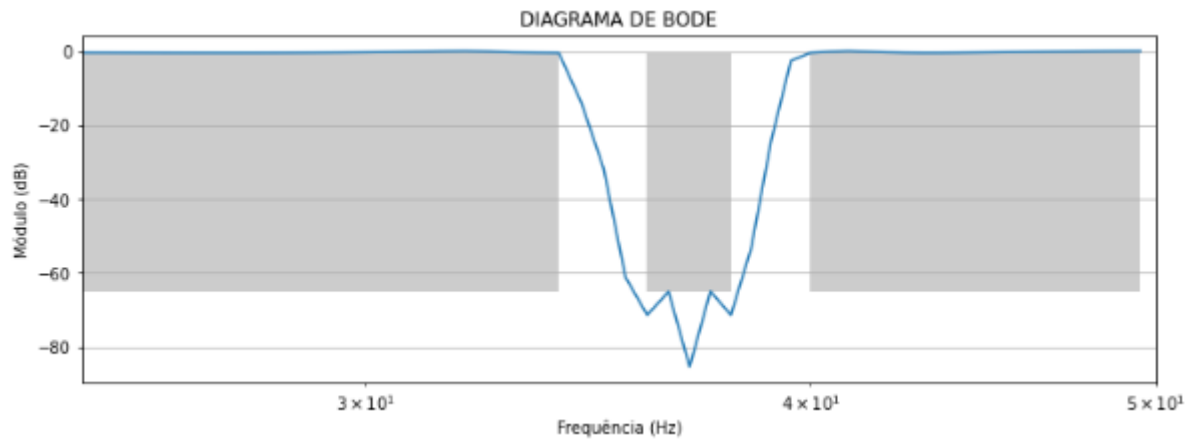
plt.fill([min(w1), wp1, wp1, min(w1)], [Gr, Gr, Gp, Gp], '0.8', lw=0) # Banda de passa
plt.fill([max(w1), wp2, wp2, max(w1)], [Gr, Gr, Gp, Gp], '0.8', lw=0) # Banda de passa
plt.fill([wr1, wr2, wr2, wr1], [Gr, Gr, Gp, Gp], '0.8', lw=0) # Banda de rejeição

plt.title("DIAGRAMA DE BODE")
plt.xlabel("Frequência (Hz)")
plt.ylabel("Módulo (dB)")
plt.xlim((25,50))
plt.grid(1)
plt.show();
```

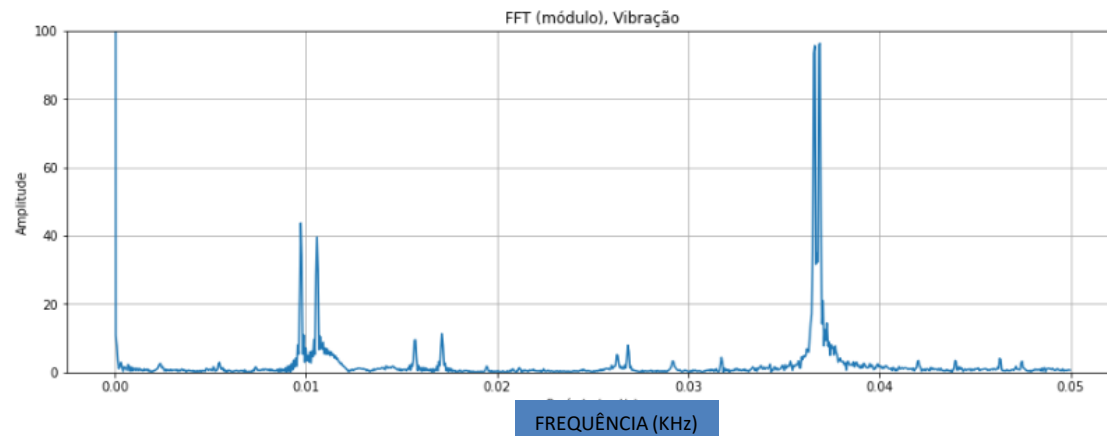
## FILTRO BUTTERWORTH



## FILTRO ELIPTICO



## TANQUINHO MAIS POTENTE MOTOR\_BOM\_SEM\_AGITADOR



## TANQUINHO MAIS POTENTE MOTOR\_BOM\_COM\_AGITADOR

