

FACULDADE DE TECNOLOGIA DE SÃO PAULO

CURSO SUPERIOR DE TECNOLOGIA EM MICROELETRÔNICA

Desenvolvimento de gráfico de controle de frequências com dados simulados de batimentos cardíacos com sinais senoidais usando *Python*.

Alunos: Pedro Henrique Caetano Bento
Ricardo Magalhaes Cruz Alves Silva

Orientador: Prof. Me. Carlos Takeo Akamine

SÃO PAULO 2022

AGRADECIMENTOS

Agradecemos ao corpo docente pelas oportunidades de aprendizado e crescimento acadêmico durante estes anos na FATEC SP como um todo. Agradecemos também a parentes e familiares que sempre nos sustentaram, apoiaram e contribuíram para o nosso desenvolvimento pessoal e acadêmico.

Agradecemos a colegas que, em momentos oportunos, compartilharam comigo seus aprendizados e mantiveram a cordialidade para além do caminho apenas do conhecimento tecno-científico.

Agradecemos ao nosso orientador, professor Carlos Takeo, pelos ensinamentos edificantes durante a elaboração deste trabalho, além das disciplinas ministradas ao longo do curso.

“É muito melhor lançar-se em busca de conquistas grandiosas, mesmo expondo-se ao fracasso, do que alinhar-se com os pobres de espírito, que nem gozam muito nem sofrem muito, porque vivem numa penumbra cinzenta, onde não conhecem nem vitória, nem derrota.”
(Theodore Roosevelt)

RESUMO

Este trabalho obteve um simulador de sinais usando linguagem de programação *Python*, de forma a descobrir padrões através de simulação utilizando a transformada de Fourier. Os sinais simulados foram as senoidais, ondas quadradas, pulsos periódicos (semelhante ao delta de Dirac), ondas de dente de serra e o ruído branco. O sinal de ruído branco permite recriar situações mais realísticas de um batimento cardíaco e utilizada para avaliar a carta de controle que monitora as variações de frequências. Os resultados mostram que o uso da linguagem *Python* para o monitoramento da frequência usando a Transformada Rápida de Fourier é uma boa técnica para ser utilizada em situações reais de problema que contenha eventos periódicos.

Palavras-chave: Sinal; simulação; Transformada Rápida de Fourier

ABSTRACT

This work obtained a signal simulator using the *Python* programming language, in order to discover patterns through simulation using the Fourier transform. The simulated signals were sinusoidal, square waves, periodic pulses (similar to Dirac delta), sawtooth waves and white noise. The white noise signal allows you to recreate more realistic situations of a heartbeat and used to evaluate the control chart that monitors the frequency variations. The results show that the use of the *Python* language for frequency monitoring using the Fast Fourier Transform is a good technique to be used in real problem situations that contain periodic events.

Keywords: Signal; simulation; Fast Fourier Transform

LISTA DE FIGURAS

Figura 1. Sinal em tempo contínuo (a) e em tempo discreto (b).....	14
Figura 2. Sinal senoidal de amplitude A , fase ϕ e período T	15
Figura 3. Ondas Quadradas de período T	16
Figura 4. Representação da função de Dirac.	17
Figura 5: Sinal de dente de serra.	17
Figura 6. Carta de Controle em condição de Normalidade.....	20
Figura 7. Carta de Controle em condição de Anormalidade no processo.....	20
Figura 8. Senoidais com três frequências (50, 55 e 60 ondas por minuto).	24
Figura 9. Frequência obtida pela FFT.....	24
Figura 10. Sinal de Ruído branco com distribuição normal.....	25
Figura 11. Gráfico das amplitudes e frequências observadas por FFT.....	25
Figura 12. Ondas de Dirac.	26
Figura 13. Gráfico das amplitudes e frequências observadas por FFT.....	26
Figura 14. Onda quadrada com frequência de 80.	27
Figura 15. Gráfico das amplitudes e frequências observadas por FFT.....	27
Figura 16. Onda dente de serra com frequência de 80.....	28
Figura 17. Gráfico das amplitudes e frequências observadas por FFT.....	28
Figura 18. Onda dente de serra com frequência de 80.....	29
Figura 19. Transformada de Fourier dos sinais com ruído branco variando o valor do sigma.....	30
Figura 20. Simulação de batimento cardíaco com frequência de 60 bpm.....	31
Figura 21. Simulação de batimento cardíaco com frequência de 60 bpm (dente de serra).	31
Figura 22. Carta de controle ao supor normalidade da variação da frequência em 10%.	32
Figura 23. Carta de controle com anormalidade da variação da frequência em 30%	33

Figura 24. Carta de controle com anormalidade da variação da frequência em 10%
.....34

LISTA DE ABREVIATURAS E SIGLAS

ECG	Eletrocardiograma
T	Temperatura
EEG	Eletroencefalograma
GSR	Resposta Galvânica da pele
EMG	Eletromiograma
RSP	Respiração
LSC	Limite Superior de Controle
LIC	Limite Inferior de Controle
LC	Linha Central
FFT	Fast Fourier Transform

LISTA DE TABELAS

Tabela 1: Tabela da Frequência máxima de acordo com a idade.....20

Tabela 2. Valores das frequências obtidos por FFT.....28

ÍNDICE

1	INTRODUÇÃO.....	12
2	OBJETIVOS.....	13
3	PARTE TEÓRICA.....	13
3.1	Tipos de Sinais.....	13
3.2	Sinais de tempo contínuo e de tempo discreto.....	14
3.3	Sinal Senoidal.....	14
3.4	Sinal Quadrada.....	15
3.5	Sinal da função delta.....	16
3.6	Sinal “dente de serra”.....	17
3.7	Transformada de Fourier.....	18
3.8	Transformada Contínua de Fourier.....	18
3.9	Transformada Discreta Fourier.....	19
3.10	Transformada Rápida de Fourier.....	19
3.11	Carta de controle.....	19
3.12	Método Monte Carlo.....	20
3.13	Python.....	21
3.14	Sinais corpóreos.....	22
4	RESULTADOS.....	23
4.1	Simulador de Sinais.....	23
4.2	Ondas Senoidais.....	23
4.3	Ruído Branco.....	24
4.4	“Delta de Dirac” ou de Pulso Unitário.....	26
4.5	Ondas quadradas.....	27
4.6	Ondas de dente de serra.....	28
4.7	Análise do número de pontos para identificação das frequências...29	
4.8	Análise do sinal senoidal com ruído branco.....	29
4.9	Simulação de frequência cardíaca e atuação da carta de controle..30	
4.10	Carta de controle.....	32
5	CONCLUSÕES.....	34
6	TRABALHOS FUTUROS.....	35

7	REFERÊNCIAS.....	36
8	ANEXO.....	38
8.1	Importação das bibliotecas.....	38
8.2	Criação das Variáveis.....	38
8.3	Desenvolvimento das ondas senoidais.....	38
8.4	Transformada de Fourier das Ondas senoidais.....	39
8.5	Desenvolvimento do ruído branco.....	39
8.6	Transformada de Fourier ruído branco.....	39
8.7	Transformada de Fourier da onda senoidal com ruído branco.....	40
8.8	Transformada de Fourier do sinal senoidal com ruído branco alterando o valor do sigma.....	41
8.9	Desenvolvimento onda Quadrada.....	42
8.10	Transformada de Fourier da onda quadrada.....	42
8.11	Desenvolvimento do “delta de Dirac”	42
8.12	Transformada de Fourier da onda “delta de Dirac”	43
8.13	Desenvolvimento da onda dente de serra.....	43
8.14	Desenvolvimento carta de controle.....	43

1 INTRODUÇÃO

Os sinais estão presentes no cotidiano e como exemplo tem-se a temperatura ambiente, o som de música, nível de poluição do ar, frequência cardíaca, temperatura do corpo entre outros. Os indicadores do mercado como a cotação do dólar, o índice Ibovespa e a taxa de inflação também podem ser considerados como sinais.

Os sinais ou dados observados ao longo do tempo podem ser coletados em tempo contínuo ou tempo discreto. A taxa de inflação (mensal) são dados observados em tempo discreto (mensalmente) e as medidas de temperatura observada num termômetro são classificados como de tempo contínuo. Situações em que as medições são realizadas através de sensores com registro no computador devem ser por amostragem e consideradas como discreta.

Os estudos dos sinais podem ser realizados no domínio do tempo, mas, em situações que ocorrem periodicidade (simples ou múltipla) a análise no domínio do tempo pode ficar limitado e, o uso do domínio da frequência são a melhor alternativa em obter informações dos dados.

Estudo dos sinais corpóreos de indivíduos em situações de estresse ou ansiedade podem alterar durante certo período e podem indicar situações críticas e a obtenção em “tempo real” dessas condições permite compreender esses fenômenos através dos dados. O presente trabalho tem o intuito de simular vários tipos de sinais e obter um procedimento para monitorar uma frequência criando um gráfico semelhante à carta de controle e acredita-se que um possível uso dessa ferramenta seja no monitoramento da frequência cardíaca.

As ferramentas utilizadas para o trabalho são a linguagem de programação *Python* que contém várias bibliotecas como o *matplotlib* e o *numpy* e que agilizam no desenvolvimento dos simuladores e a criação de novas ferramentas antes de sua aplicação no mundo real.

2 OBJETIVOS

O trabalho tem como objetivo simular sinais e observar as frequências através da transformada de Fourier. Os sinais simulados serão o senoidal, as ondas quadradas, a função de “Dirac”, o dente de serra e o ruído branco. O ruído branco será utilizado para a simular condições “reais” e avaliar o comportamento da transformada de Fourier como instrumento para as medidas de frequências. Os sinais simulados do “batimento cardíaco” serão avaliados em várias condições de variação de sua frequência e avaliar como a carta de controle funcionaria como ferramenta de monitoramento do controle da frequência.

3 PARTE TEÓRICA

No dia a dia e durante quase todos os momentos, lidamos com sinais, sinais estes que contêm ou demonstram algo que de alguma forma nos trazem informação sobre algum tipo de anomalia, perturbação da natureza ou simplesmente um ruído. O som de um aparelho como radio, ou mesmo na transmissão de alguma mensagem através da voz seria um exemplo de uso para identificação de um indivíduo através da voz ou indicação de anomalias nas cordas vocais. Outro exemplo seria os sinais de um eletrocardiograma onde o resultado dele possibilita que o médico indique uma condição de normalidade ou não.

No entanto, o que está em pauta é como obter e analisar as frequências desses sinais que são nada mais do que um indicativo de atividade elétrica do coração, quando a energia elétrica é distribuída de forma que ele se torne um sinal senoidal, ou seja com uma determinada frequência e amplitude.

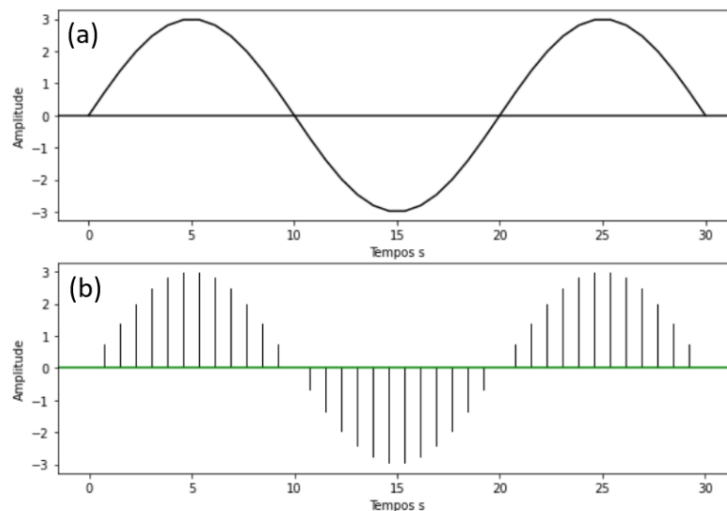
3.1 Tipos de Sinais

Existem uma grande quantidade de tipos de sinais, os exemplos citados anteriormente possuem uma variável independente e são denominados como unidimensionais. Os vídeos são exemplos de sinal bidimensional, uma de iluminação e outra de cores. A projeção holográfica são sinais tridimensionais com três variáveis de posição. (TOKIO, 2003)

3.2 Sinais de tempo contínuo e de tempo discreto

Sinais definidos em todo o instante de tempo são denominados de sinais de tempo contínuo e, quando registrados em determinados instantes de tempo são considerados sinais de tempo discreto. A representação da senoide apresentado na Figura 1a é um sinal de tempo contínuo, já o sinal da Figura 1b é um sinal de tempo discreto, uma vez que está mostrando os valores nos instantes de tempo 0, 1, 2..., N. (TOKIO,2003)

Figura 1. Sinal em tempo contínuo (a) e em tempo discreto (b).



Fonte: Elaboração própria.

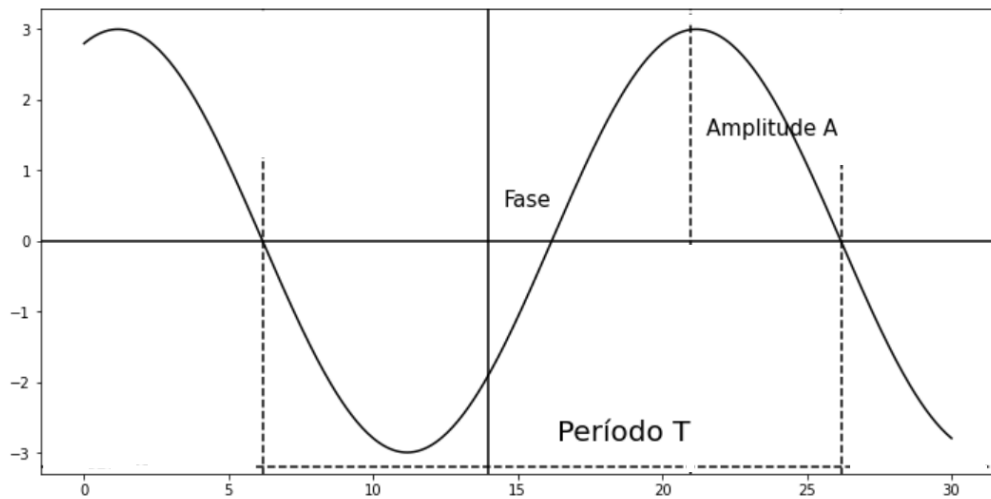
3.3 Sinal Senoidal

Um sinal senoidal é representado por: $f(t) = A \cos(\omega_0 t + \phi)$ onde t é o tempo, A é a amplitude, ω_0 é a frequência angular, medida em radianos por unidade de tempo, $1/f_0 = 2\pi/\omega_0$ é a frequência medida em ciclos por unidade do tempo e ϕ é a fase, medida em radianos. A relação entre o período, frequência e frequência angular é dada pela equação (1).

$$T_0 = \frac{2\pi}{\omega_0} = \frac{1}{f_0} \quad (1)$$

A Figura 2 ilustra a forma da onda senoidal com a descrição da amplitude, frequência, período e fase.

Figura 2. Sinal senoidal de amplitude A, fase ϕ e período T.



Fonte: Elaboração própria.

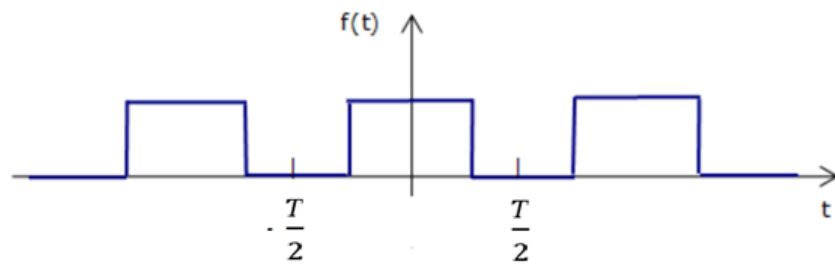
O sinal, representado por meio da equação, é uma aproximação denominada (do ângulo de fase) de senoide eterna, considerando $-\infty < t < +\infty$.

3.4 Sinal da Onda Quadrada

O sinal da onda quadrada é uma forma de onda usada geralmente em eletrônica e as ondas alternam de forma regular e instantânea entre dois níveis conforme mostra a Figura 3. A função que a descreve é a equação (2).

$$f(t) = \begin{cases} A & \text{para } |t| < \frac{T}{2} i \\ 0 & \text{para } |t| > \frac{T}{2} i \end{cases} \quad \text{para } i = \pm 1, \pm 2, \dots \quad (2)$$

Figura 3. Ondas Quadradas de período T.



Fonte: Elaboração própria.

As ondas quadradas são encontradas de modo comum em circuitos diversos, como circuitos de chaveamento digitais e em dispositivos lógicos de dois níveis. Geralmente usadas como sinais de *clock* (relógio), elas são normalmente usadas para identificação de sons como som oco, e como base para sons de instrumentos de sopro.

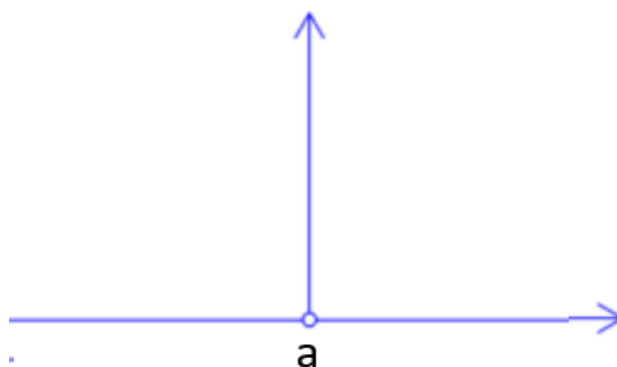
3.5 Sinal da função delta

Na teoria, a função delta de Dirac, é uma linha vertical, a qual vale infinito no ponto **a** e é nula no restante. A integral da função delta de Dirac em todos os pontos é definida como tendo valor 1. (DIRAC,1930)

Podemos considerar essa representação como um retângulo muito estreito e muito alto, com área igual a uma unidade. Muitas vezes, pode ser utilizado como um limite de funções que atendem a essas afirmações. No contexto de processamento de sinais, ela é frequentemente interpretada como um impulso unitário. (DIRAC,1930). A função é descrita na equação (3) e representada na Figura 4.

$$f(t) = \begin{cases} \infty & \text{para } t = a \\ 0 & \text{para } t \neq a \end{cases} \quad (3)$$

Figura 4. Representação da função de Dirac.



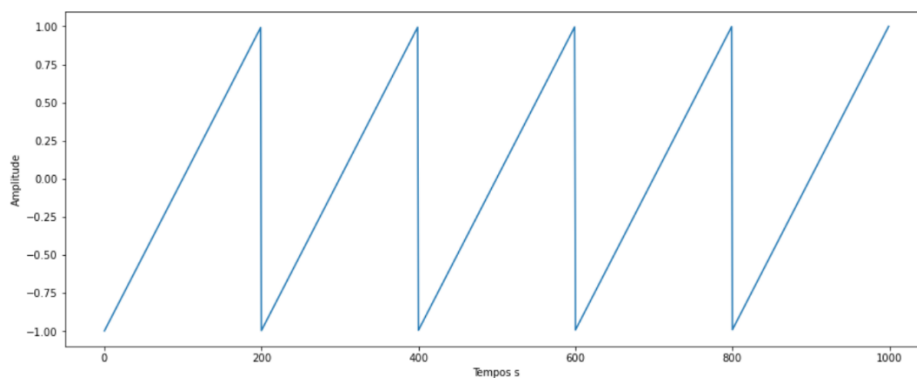
Fonte: Elaboração própria

Teoricamente, não pode ser considerado como uma função, mas sim como um objeto matemático. Isso porque qualquer função que valha zero em todos os pontos exceto num ponto, deve ter integral nula em toda a reta. (DIRAC,1930)

3.6 Sinal “dente de serra”

Um sinal dente de serra é um tipo de onda não senoidal encontrada na eletrônica ou no processamento de sinais. Leva o nome devido a sua representação gráfica que se assemelha aos dentes de uma serra. A Figura 5 ilustra a forma do gráfico de dente de serra.

Figura 5: Sinal de dente de serra.



Fonte: Elaboração própria.

3.7 Transformada de Fourier

Um estudo sobre condução de calor realizado pelo matemático e físico Jean Baptiste Joseph Fourier (1768-1830) levou a um resultado de que qualquer forma de onda pode ser representada por uma somatória de funções trigonométricas como senóides e cossenóides com diferentes frequências, amplitudes e fases. Ficou reconhecido como introdutor do estudo da decomposição de funções periódicas em séries trigonométricas convergentes e conhecido como as séries de Fourier.

Um sinal periódico $f(t)$ pode ser decomposto como descrito na equação (4). O sinal $f(t)$, pode ser representada como uma série de senos e cossenos múltiplos da frequência fundamental w somados a uma constante c_0 .

$$f(t) = c_0 + \sum_{n=1}^{\infty} (a_n \cos w_n t + b_n \sen w_n t) \quad (4)$$

Sendo c_0 , a_n e b_n os coeficientes de Fourier dada por:

$$c_0 = \frac{1}{T} \int_0^T f(t) dt ; a_n = \frac{2}{T} \int_0^T f(t) \cos w_n t dt ; b_n = \frac{2}{T} \int_0^T f(t) \sen w_n t dt$$

A frequência fundamental ocorre quando $n = 1$ e T é o período.

3.8 Transformada Contínua de Fourier

Para uma função $f(t)$ contínua e integrável no domínio do tempo. A transformada de Fourier corresponde a soma de exponenciais complexas com frequência angular w e amplitude. A equação (5a) e sua inversa como (5b) indicam que $f(w)$ “carrega” todas as informações de $f(t)$. Lembrando da relação de Euler: $x(t) = e^{itw} = \cos(wt) + i \sen(wt)$ com $i = \sqrt{-1}$.

$$F(f)(w) = \int_{-\infty}^{+\infty} f(t) e^{-itw} dt \quad (5a)$$

$$f(t) = F^{-1}(F(w)) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(w) e^{itw} dw \quad (5b)$$

O termo transformada de Fourier refere-se a ambas as representações no domínio de frequência e a operação matemática que associa a representação no domínio de frequência a uma função temporal.

3.9 Transformada Discreta Fourier

Um sinal em tempo discreto $x[n]$ onde n é um número inteiro tem sua transformada de Fourier em tempo discreto como descrito na equação (6a) e sua inversa como (6b).

$$X(e^{i\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-i\omega n} \quad (6a)$$

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} X(e^{i\omega}) e^{i\omega n} d\omega \quad (6b)$$

3.10 Transformada Rápida de Fourier

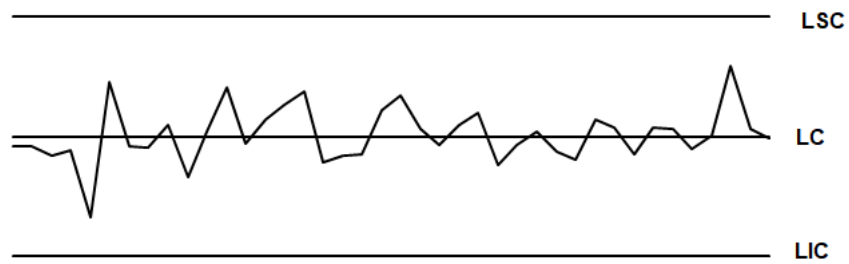
A Transformada Rápida de Fourier é um método numérico otimizado de obtenção dos coeficientes da Transformada Discreta de Fourier. Corresponde a um algoritmo que exige pouco esforço computacional e é utilizado por vários pacotes computacionais e conhecido como "*Fast Fourier Transform*" (FFT).

3.11 Carta de controle

A carta de controle foi criada por Shewart no Bell Labs na década de 20. A técnica tem como objetivo reconhecer os desvios em relação a um comportamento normal e identificar possíveis melhorias em processos produtivos. É um procedimento gráfico que incorpora técnicas estatísticas com a facilidade visual de identificar anomalias em processos. (SILVEIRA,2013)

A carta contém três linhas horizontais denominadas de Limite Superior de Controle (LSC), Limite Inferior de Controle (LIC) e Linha Central (LC) e vários pontos interligados que correspondem a uma estatística de amostras coletadas ao longo do tempo conforme ilustra a Figura 6. As estatísticas podem ser a média, amplitude ou desvio padrão amostral entre outras medidas.

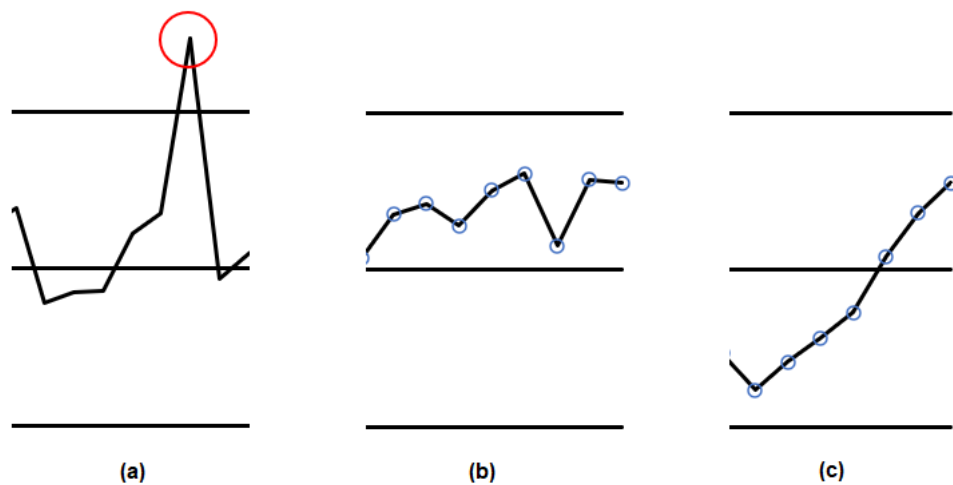
Figura 6. Carta de Controle em condição de Normalidade



Fonte: Elaboração própria.

Uma condição indicando anormalidade ocorre quando um ponto fica fora dos limites de controle (Figura 7a) mas, existem outros padrões de pontos que podem indicar anormalidade como sete consecutivos numa mesma faixa em relação a LC (Figura 7b) ou em sequência ascendente ou descendente (Figura 7c).

Figura 7. Carta de Controle em condição de Anormalidade no processo.



Fonte: Elaboração própria.

3.12 Método Monte Carlo

A Simulação Monte Carlo foi criada pelo matemático John Von Neumann durante a Segunda Guerra Mundial no projeto *Manhattan* (associado à bomba atômica) e o termo faz referência a região com vários cassinos.

Resumidamente, o método gera dados aleatórios associados a eventos que representam situações reais (simplificadas) a fim de criar uma solução para

problemas de difícil determinação analítica. Geralmente usados para solução em problemas de física e de matemática de grande dificuldade ou de haver impossibilidade de ser solucionados experimentalmente. (ROSENBLUTH,1955)

O método utiliza modelos probabilísticos pré-determinados que geram dados aleatórios e permitem simular várias situações e obter soluções em cada condição.

3.13 Python

A linguagem Python foi criada em 1989 por Guido Van Rossum e atualmente é considerada a segunda linguagem mais utilizada no mundo segundo as publicações da *RedMonk*, uma empresa voltada para desenvolvimento de softwares (2020), ficando atrás somente da linguagem *JavaScript*. O *Python* é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos. (REDMONK, 2021)

A linguagem *Python* é dotada de uma versatilidade que permite ao usuário trabalhar em diversos ambientes de interação computacionais, e tal facilidade permite embutir outras bibliotecas matemáticas e ferramentas que agregam valor às investigações estatísticas nesse sentido. Dentre as bibliotecas disponíveis são o *Matplotlib*, *Numpy* e *Pandas*.

O *Matplotlib* é uma biblioteca de softwares para criação de gráficos e visualizações de dados em geral, feita para e da linguagem de programação Python e sua extensão de matemática *NumPy*, sendo uma plataforma cruzada de visualização de dados e plotagem gráfica. (STEPHENS,2021)

O *Pandas* é uma biblioteca para manipulação e análise de dados, escrita em *Python*. Essa é a biblioteca mais indicada para iniciar análises exploratórias de dados, pois permite efetuar leituras estatísticas, fora a manipulação e plotagem de dados de forma sucinta. Em relação ao presente trabalho, esta ferramenta é utilizada a fim de modificar estruturas e operações para manipular tabelas numéricas e séries temporais.

O *NumPy* é uma biblioteca *Python* usada principalmente para realizar cálculos em *arrays* multidimensionais. Esta biblioteca fornece um grande conjunto de funções e operações de que auxiliam programadores a executar cálculos numéricos. (STEPHENS,2021)

3.14 Sinais corpóreos

As emoções, que afetam o estado fisiológico e psicológico humano, desempenham um papel muito importante na vida humana. As emoções positivas ajudam a melhorar a saúde humana e a eficiência do trabalho, enquanto as emoções negativas podem causar problemas de saúde, e acúmulos dessas emoções a longo prazo, predis põem fatores para a depressão, que podem levar ao suicídio nos piores casos. (CESAR, 2021)

Em comparação com o humor que é um estado mental mais consciente, a emoção geralmente se refere a um estado mental que surge de modo espontâneo, e muitas vezes é acompanhado por esforços físicos e mudanças fisiológicas que são relevantes para os órgãos e tecidos humanos como cérebro, coração, pele, fluxo sanguíneo, músculos, expressões faciais, voz dentre outros.

A outra categoria está usando os sinais internos – os sinais fisiológicos, que incluem o eletroencefalograma (EEG), temperatura (T), eletrocardiograma (ECG), eletromiograma (EMG), resposta galvânica da pele (GSR), respiração (RSP) dentre outros. (CESAR, 2021)

A frequência cardíaca é um sinal que pode mostrar o bom ou mal funcionamento do coração. Dependendo das condições fisiológicas do indivíduo podem ser de alta ou baixa frequência. A frequência cardíaca possui um valor máximo e seus valores são tabelados conforme sua faixa etária. A Tabela 1 mostra a frequência cardíaca máxima de acordo com a idade. (ALMEIDA,2021)

Tabela 1: Tabela da Frequência máxima de acordo com a idade.

idade	Frequência máxima
20	200
30	191
35	185
40	180
45	175
50	171
55	165
60	160
65	155
70	151

Fonte: Elaboração própria.

Numa atividade de exercício de uma pessoa ativa, a frequência deve ficar dentro do limite de 50% e aos poucos elevando ao decorrer do tempo até alcançar 85% do máximo. (VEJA, 2019)

4 RESULTADOS

Inicialmente foram criados os geradores de sinais senoidais, ondas quadradas, ondas de pulsos, ondas de dente de serra e ruído branco em *Python*. Os sinais senoidais foram avaliados alterando o número de amostragem que permitisse uma boa identificação pela Transformada Rápida de Fourier e, posteriormente foram inseridos o ruído nas ondas senoidais para avaliar a capacidade de identificação quando o nível do ruído era aumentado.

Os resultados possibilitam a indicação no número de pontos para uma boa estimativa das frequências e com isso possibilita a criação da carta de controle para avaliar o simulador de batimento cardíaco. Neste caso foram criados simuladores com dois tipos de sinais (pulso e dente de serra) somados ao ruído branco para avaliar a sensibilidade da carta de controle.

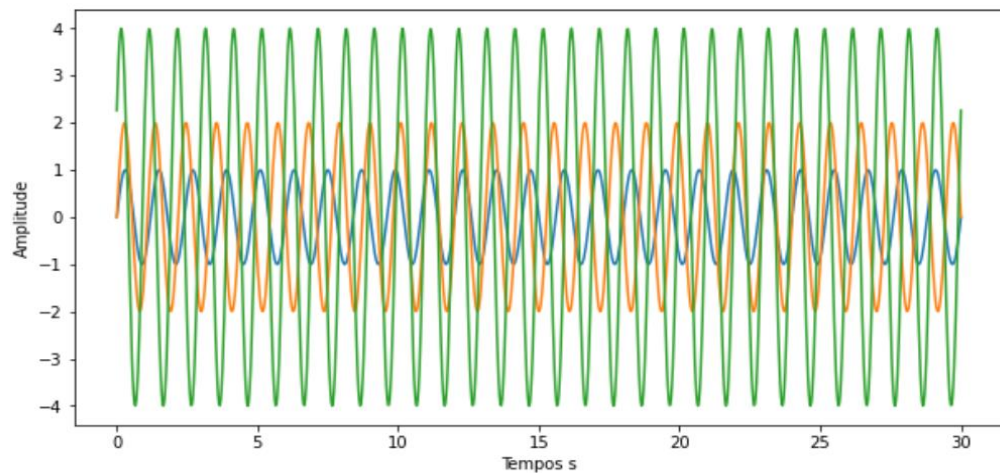
4.1 Simulador de Sinais

Para simular os sinais senoidais, quadrada, dente de serra e do “delta de Dirac” foram desenvolvidos os códigos em *Python* que estão disponíveis em ANEXO.

4.2 Ondas Senoidais

Através dos códigos criados em *Python*, as ondas senoidais foram criadas usando a biblioteca *Numpy* e os resultados da função senoidal com três frequências distintas estão ilustradas na Figura 8.

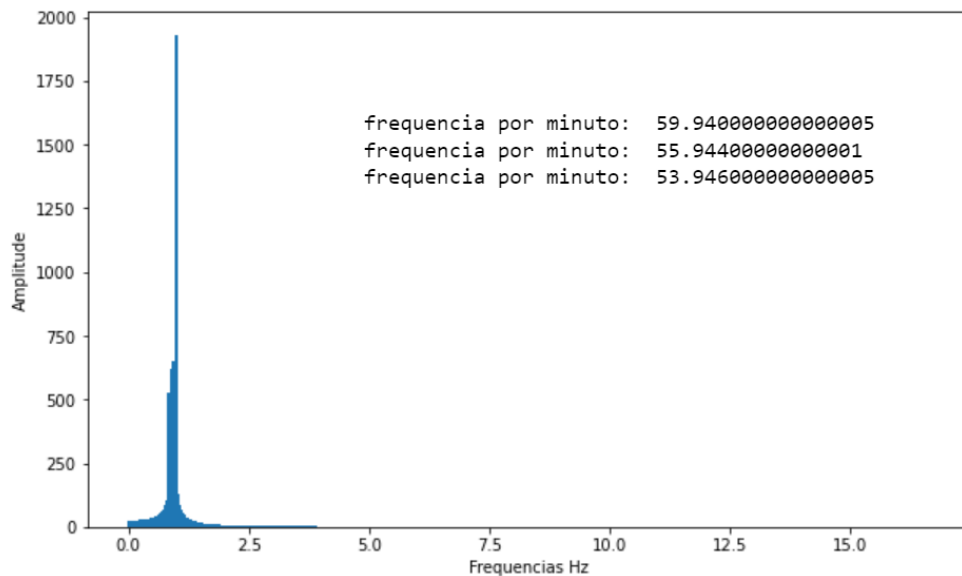
Figura 8. Senoidais com três frequências (50, 55 e 60 ondas por minuto).



Fonte: Elaboração própria.

Os valores das frequências obtidas nessas condições estão ilustrados na Figura 9. (O eixo da abcissa foi calculado em Hz e o valor apresentado transformado em ondas por minuto)

Figura 9. Frequência obtida pela FFT.

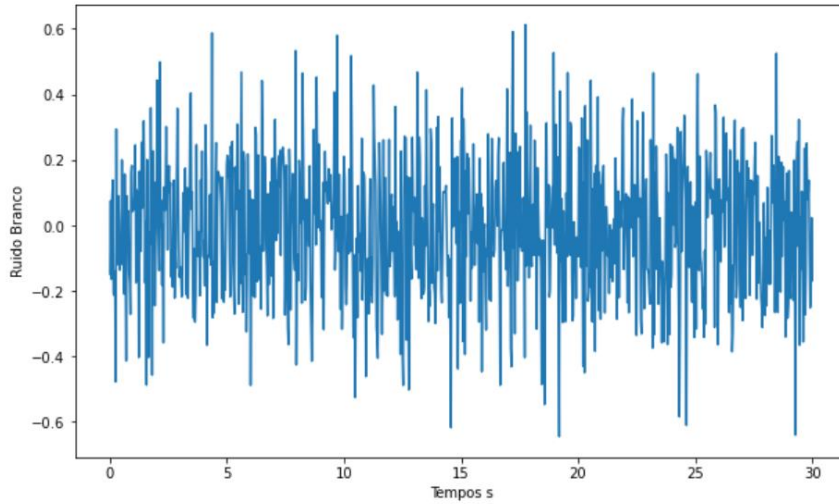


Fonte: Elaboração própria.

4.3 Ruído Branco

O ruído branco foi obtido pela simulação de valores gerados por uma distribuição normal com média zero e desvio padrão de 1,0. O resultado pode ser visualizado na Figura 10 e as frequências identificadas por FFT em duas simulações estão ilustradas na Figura 11.

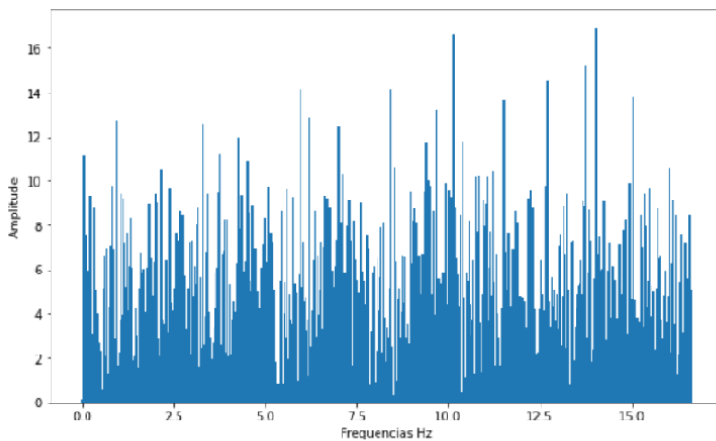
Figura 10. Sinal de Ruído branco com distribuição normal.



Fonte: Elaboração própria.

As frequências obtidas pela transformada de Fourier do sinal de ruído tem como resultado valores aleatórios uma vez que o sinal do ruído branco não possui periodicidade. Desta forma, os resultados obtidos saíram de acordo com o esperado.

Figura 11. Gráfico das amplitudes e frequências observadas por FFT.



Simulação 1

frequencia por minuto: 841.158
frequencia por minuto: 609.390
frequencia por minuto: 825.174

Simulação 2

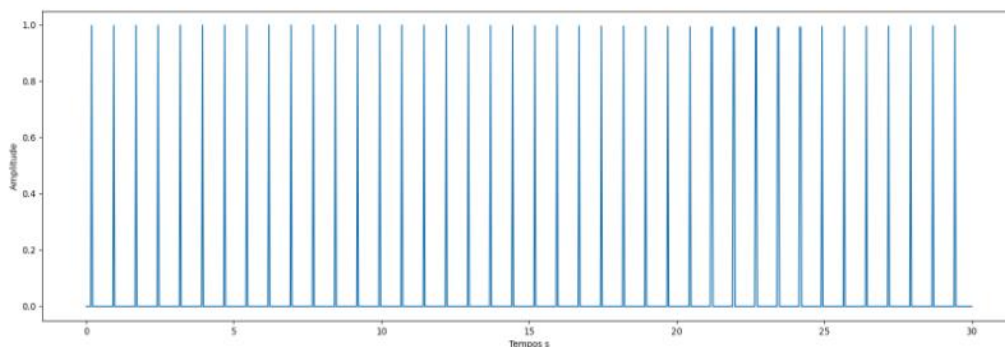
frequencia por minuto: 65.9340
frequencia por minuto: 995.004
frequencia por minuto: 903.096

Fonte: Elaboração própria.

4.4 “Delta de Dirac” ou de Pulso Unitário

As ondas de pulso unitário foram obtidas por meio da alteração de uma onda senoidal e uso de filtros, e o resultado obtida usando a frequência de 80 pulsos por minutos estão na Figura 12.

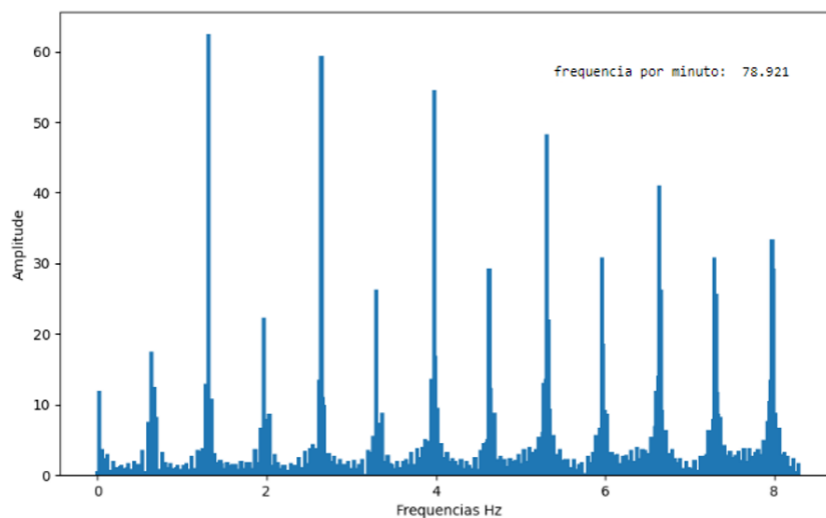
Figura 12. Ondas de Dirac.



Fonte: Elaboração própria

Os vários picos observados na Figura 13 correspondem a frequência fundamental (maior) e as outras correspondem às suas harmônicas. O resultado da frequência fundamental obtida foi de 78,921 (valor muito próximo da onda simulada com frequência 80).

Figura 13. Gráfico das amplitudes e frequências observadas por FFT.

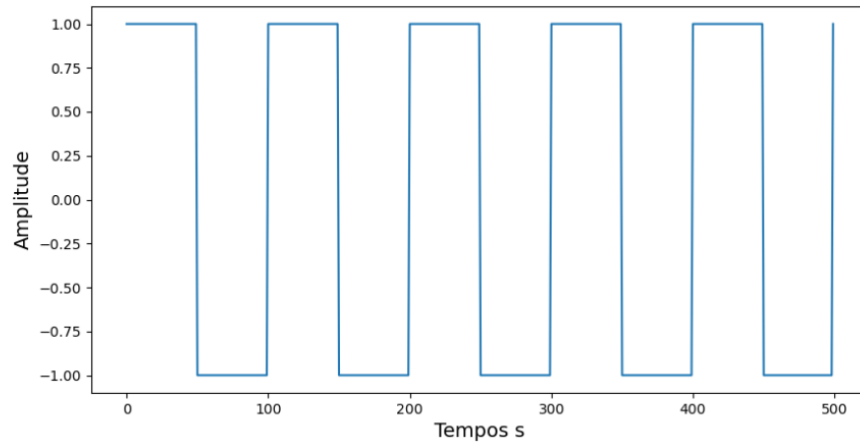


Fonte: Elaboração própria

4.5 Ondas quadradas

Para gerar uma onda quadrada foi utilizado a biblioteca *Scipy* usando a frequência de 80 ondas por minuto e descrito na Figura 14.

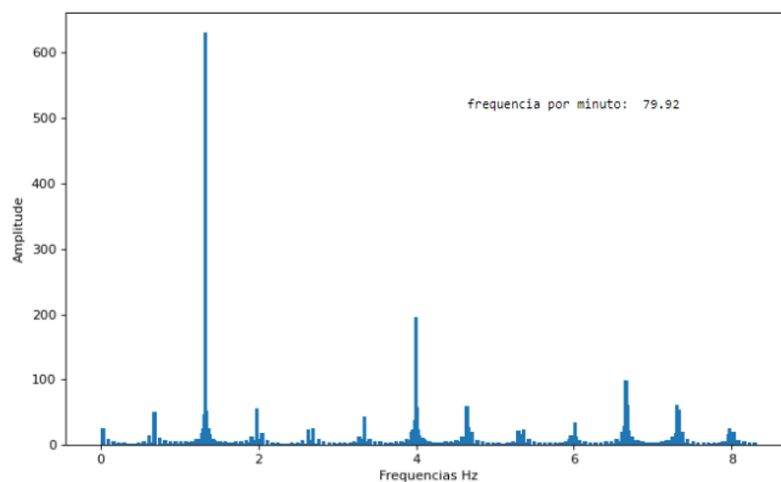
Figura 14. Onda quadrada com frequência de 80.



Fonte: Elaboração própria

Os vários picos observados na Figura 15 correspondem a frequência fundamental e as outras correspondem às suas harmônicas semelhantes ao observado na onda de pulso. O resultado da frequência fundamental obtida foi de 79,92 (valor muito próximo da onda simulada com frequência 80).

Figura 15. Gráfico das amplitudes e frequências observadas por FFT.

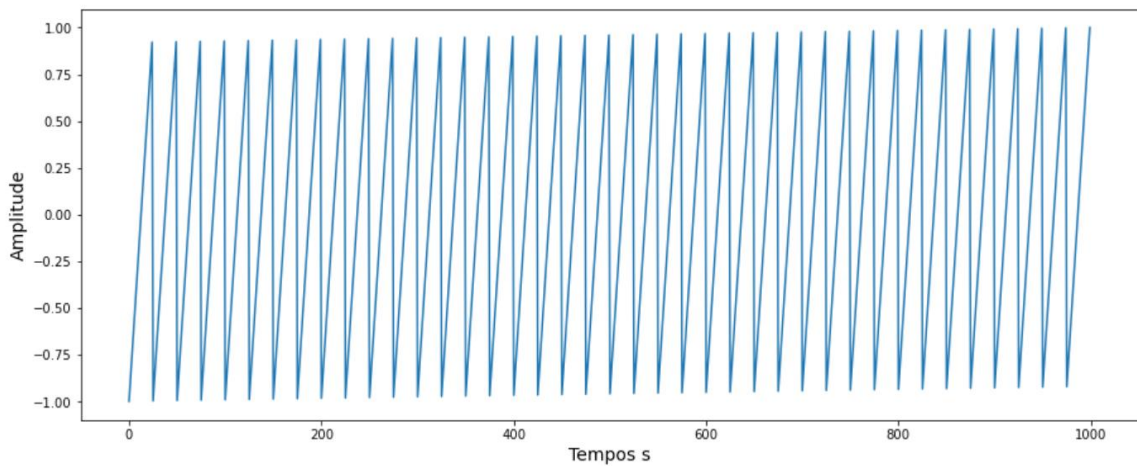


Fonte: Elaboração própria

4.6 Ondas de dente de serra

Para criar a onda na forma de dente de serra foi utilizado a biblioteca *Scipy* com frequência de 80 ondas por minuto e ilustrado na Figura 16 e sua transformada de Fourier com a frequência obtida na Figura 17.

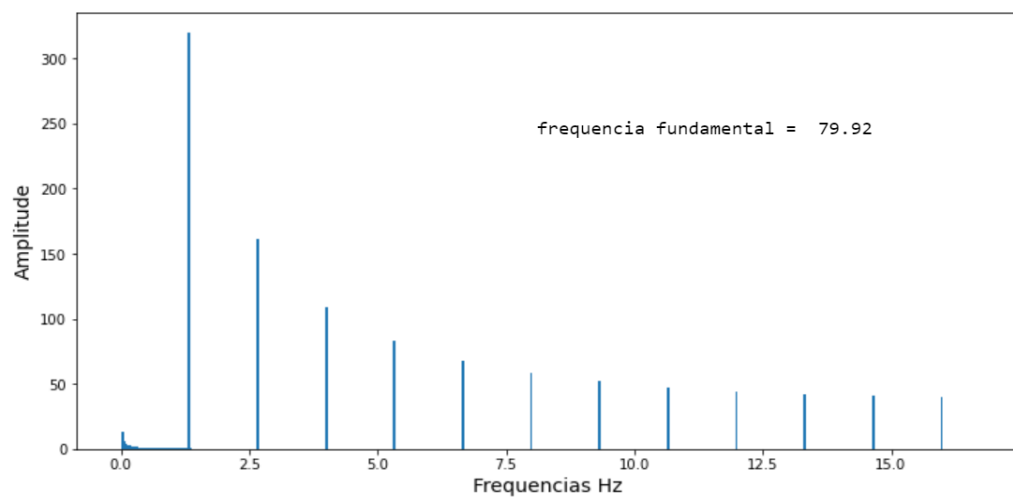
Figura 16. Onda dente de serra com frequência de 80.



Fonte: Elaboração própria.

O resultado da frequência fundamental obtida foi de 79,92 ondas por minuto que está muito próximo da onda simulada de frequência 80.

Figura 17. Gráfico das amplitudes e frequências observadas por FFT.

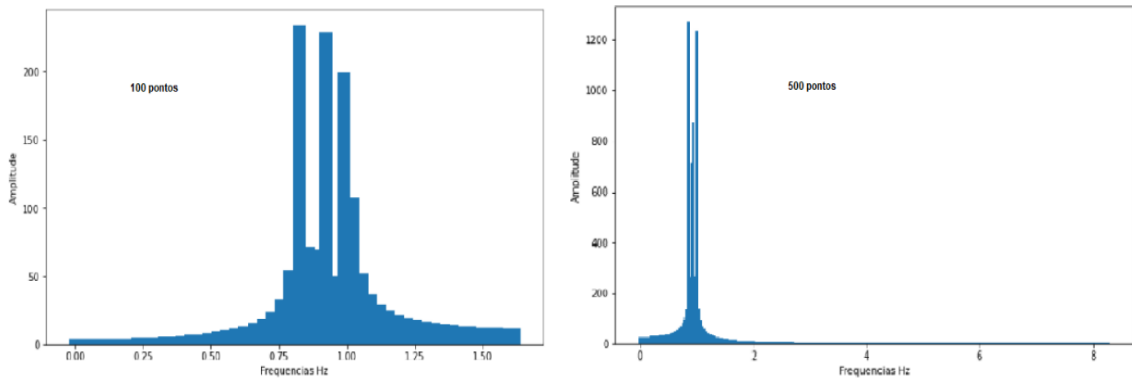


Fonte: Elaboração própria.

4.7 Análise do número de pontos para identificação das frequências

O teste consistiu em simular três ondas senoidais com frequências próximas (50, 55 e 50 bpm) e mudar a quantidade de pontos para sua identificação. A Figura 18 (a esquerda) mostra a frequências estimadas quando foi utilizado 100 pontos (baixa resolução). Ao utilizar 500 pontos a resolução para identificação é melhor conforme ilustra a Figura 18 (a direita).

Figura 18. Onda dente de serra com frequência de 80.



Fonte: Elaboração própria.

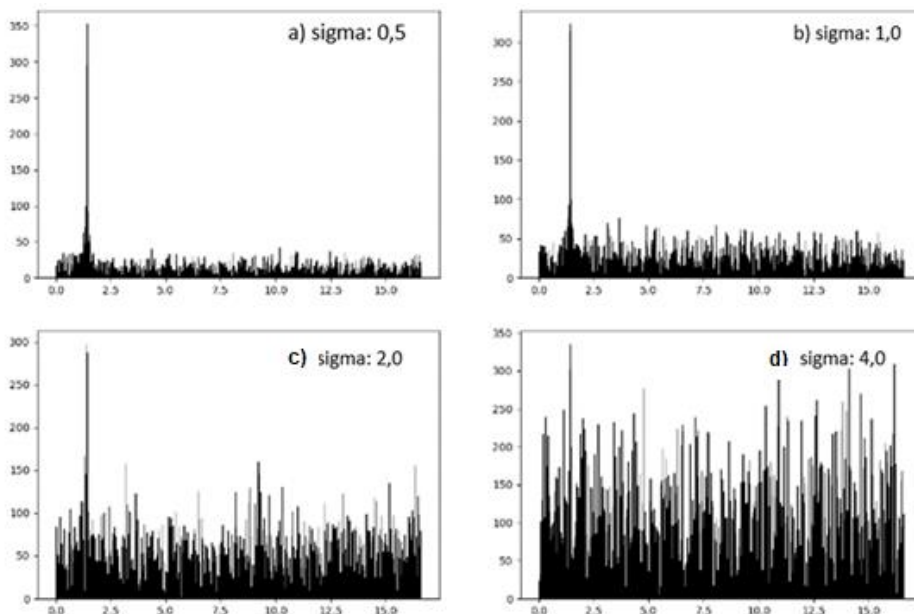
O resultado indica que o número de pontos sem problema de identificação das frequências a ser utilizada é de 1000 pontos.

4.8 Análise do sinal senoidal com ruído branco

Nesta parte do trabalho foram gerados sinais senoidais com frequência pré-estabelecido e incluído ruído branco com diferentes valores de desvio padrão. O sinal possui 1000 pontos e o valor da frequência estabelecida foi de 85 ondas por minuto.

O ruído branco tem distribuição normal com média igual a zero e desvio padrão variando de 0,5; 1,0; 2,0 e 4,0. Os resultados obtidos por FFT estão na Figura 19 (a) até 19 (d) e é visível que a estimativa da frequência piora com o aumento da variabilidade do ruído.

Figura 19. Transformada de Fourier dos sinais com ruído branco variando o valor do sigma.



Fonte: Elaboração própria

A Tabela 2 mostra a frequência fundamental identificada para outros valores do desvio padrão que vão de 0,5 a 8,0 e, para valores acima de 6,0 do desvio padrão não foi mais possível identificar a frequência simulada de 85.

Tabela 2. Valores das frequências obtidos por FFT.

desvio padrão do Ruído Branco	Frequência pela FFT
0,50	83,91
1,00	83,91
2,00	83,91
4,00	83,91
6,00	83,91
8,00	47,95

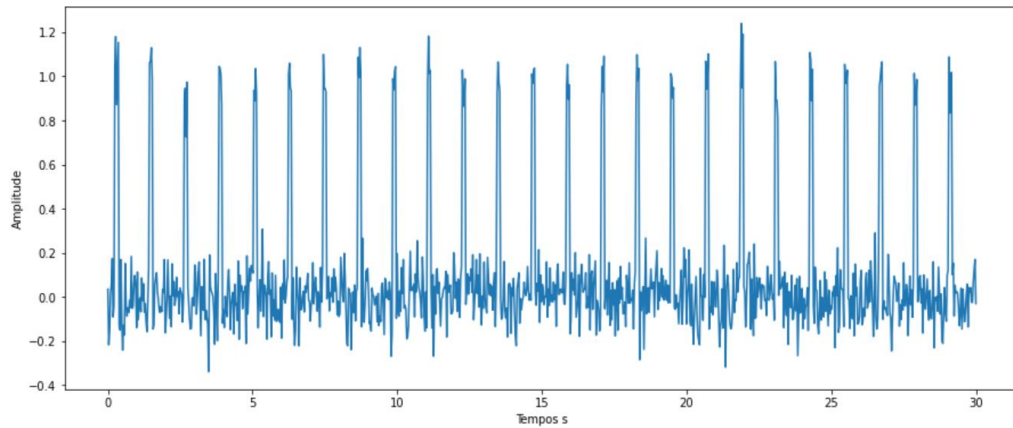
Fonte: Elaboração própria.

4.9 Simulação de frequência cardíaca e atuação da carta de controle

Dois simuladores foram desenvolvidos para sinais com uma única frequência de interesse como o batimento cardíaco. No primeiro foi utilizado pulso com frequência pré-determinado. O simulador de batimento cardíaco com

frequência de 60 bpm e com ruído branco normalmente distribuído com média zero e desvio padrão de 0,1 está ilustrado na Figura 20.

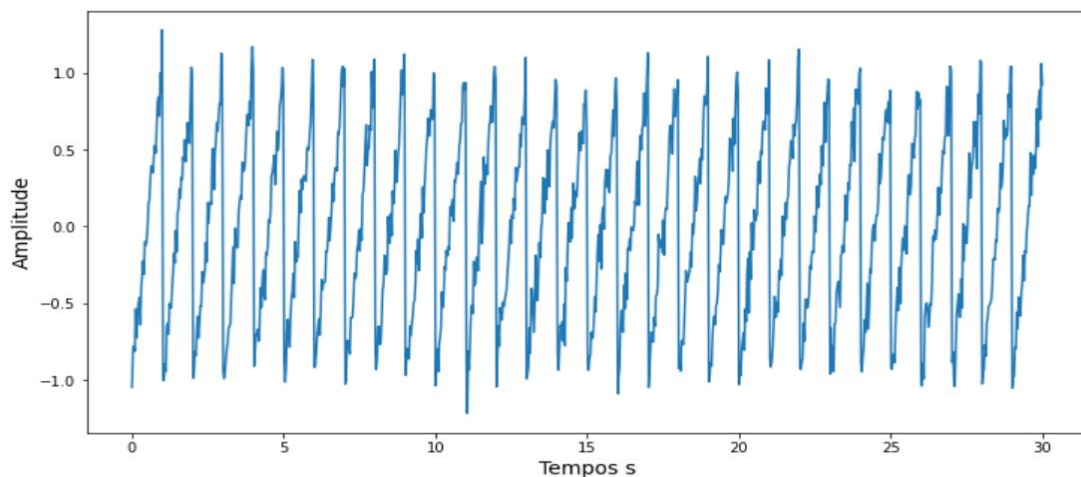
Figura 20. Simulação de batimento cardíaco com frequência de 60 bpm.



Fonte: Elaboração própria.

O segundo simulador foi obtido com onda de dente de serra mais o ruído branco normalmente distribuído com média zero e desvio padrão de 0,1. O simulador de batimento cardíaco com frequência de 60 bpm e com ruído branco está ilustrado na Figura 21.

Figura 21. Simulação de batimento cardíaco com frequência de 60 bpm (dente de serra).



Fonte: Elaboração própria.

Nos dois casos o FFT identificou-se a frequência de 59,94 bpm que está próximo ao 60 bpm simulado.

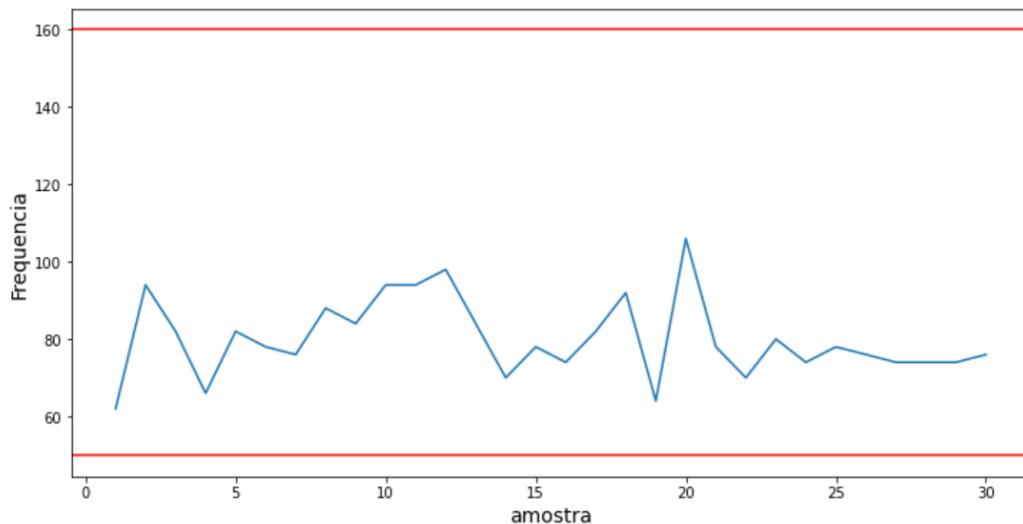
4.10 Carta de controle

As cartas de controle tradicionais são desenvolvidas supondo que exista uma condição de estabilidade do processo, isto é, quando os parâmetros são constantes ao longo do tempo e o cálculo dos limites de controle são estabelecidas nessa condição.

O presente trabalho não estuda a condição de estabilidade da frequência, mas analisa o nível de variação da frequência. Essa variação tende a indicar a anormalidade de algum produto ou órgão biológico. Nesse caso, supõe-se que a normalidade, sejam pequenas alterações na frequência e a anormalidade ocorrendo em dois casos: quando ocorre tendência nas frequências e quando a variabilidade fica acima do normal.

Este trabalho avaliou os dois casos usando as ondas de dente de serra e mudando as frequências com determinado desvio padrão e outra criando tendência nas frequências, mas mantendo a variância constante. Supondo que a normalidade da variação da frequência seja de 10% da frequência média. A carta de controle fica como ilustrado na Figura 22 quando no estado normal.

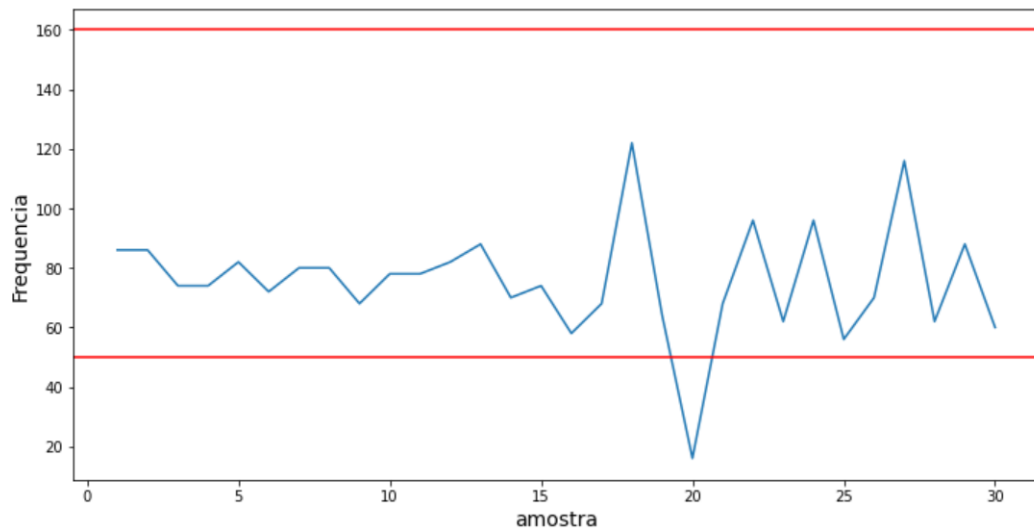
Figura 22. Carta de controle ao supor normalidade da variação da frequência em 10%.



Fonte: Elaboração própria.

A simulação do primeiro caso de anormalidade na variação da frequência em 30% da frequência média a partir do ponto 15 e carta de controle fica conforme a Figura 23.

Figura 23. Carta de controle com anormalidade da variação da frequência em 30%

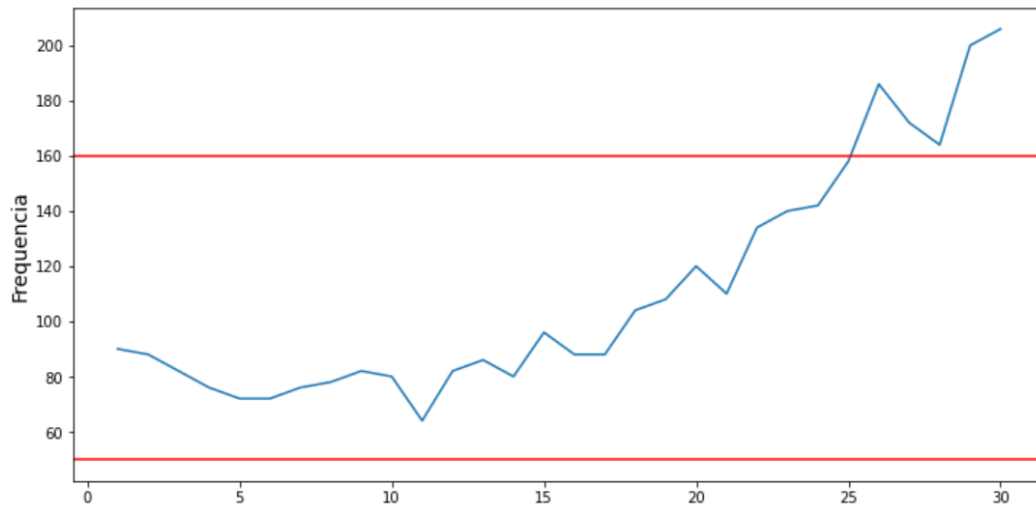


Fonte: Elaboração própria.

O gráfico da Figura 23 nos mostra que é possível identificar o efeito da variação devido ao ponto fora das linhas, mas em muitos casos continuam dentro dos limites pré-estabelecidos. Esse resultado indica que os limites devem ser mais bem planejados, pois é possível perceber que os pontos têm maior variabilidade e se os limites fossem mais estreitos seria possível identificar o problema.

A simulação do segundo caso de anormalidade é quando ocorre tendência no valor média. Essa situação foi simulada a partir da décima quinta amostra, aumentando em um desvio padrão a cada amostra e mantendo o desvio padrão em 10% da frequência média inicial. O resultado pode ser visto na Figura 24.

Figura 24. Carta de controle com anormalidade da variação da frequência em 10%



Fonte: Elaboração própria.

O gráfico da Figura 24 nos mostra que é possível de identificar o efeito da tendência (crescente) na frequência, mas novamente é necessário avaliar os limites para a identificação mais rápida o efeito da tendência. No caso em questão foi possível identificar o problema somente na vigésima quinta amostra, isto é, somente depois de 10 amostra após o início do efeito da tendência.

5 CONCLUSÕES

Os resultados obtidos neste trabalho mostram que o uso da linguagem *Python* em problemas de sinais sob efeito de periodicidade, é adequada pois atualmente existem muitas bibliotecas disponíveis para análise e que agilizam e flexibilizam o estudo de vários problemas.

As técnicas de simulação Monte Carlo foram úteis para o desenvolvimento do simulador e foi possível avaliar a capacidade e a limitação do FFT em obter as estimativas das frequências.

O simulador do batimento cardíaco e a carta de controle mostraram seu potencial no uso em aplicações reais.

6 TRABALHOS FUTUROS

A seguir estão as sugestões para os trabalhos futuros:

- Desenvolver sensores para obter os sinais de frequência cardíaca e a aquisição de seus dados e utilizar os modelos criados nesse trabalho.
- Melhorar os limites de controle para identificação de algum problema associado a frequência cardíaca como arritmia, por exemplo.
- Avaliar a frequência em condição de estresse como de ansiedade, pressão em tomada de decisão entre outras situações.

7 REFERÊNCIAS

1. SOUSA Renato; Como é feita a leitura do ritmo cardíaco na banda de peito? Disponível em: <https://vidademaratonista.com/leitura-ritmo-cardiaco-banda-e-relogio-de-pulso/> Acesso em: 12 maio 2022.
2. ALMEIDA Sandra; - Mundo da Automação - Disponível em: <https://www.mundodaaautomacao.net/2021/04/sensor-de-frequencia-cardiaca.html> Acesso em: 13 maio 2022.
3. IGOR SIMÃO; LUCAS CAVALCANTE MARCELO - Sistema de sensores e comunicação sem fio para monitoramento de sono e auto quantificação Engenharia Eletrica - UFPR - Universidade Federal do Paraná - Disponível em: <http://www.eletrica.ufpr.br> > arquivostccs Acesso em: 4 abril de 2022.
4. SILVEIRA; Cristiano Bertulucci - Carta de Controle - Disponível em: <https://www.citisystems.com.br/carta-de-controle/> 19 de fevereiro de 2013 Acesso em: 4 maio 2022.
5. ROGERS, D W O (2006). "Fifty years of Monte Carlo simulations for medical physics". *Physics in Medicine and Biology*.
6. ROSENBLUTH, Marshall, N.; Rosenbluth, Arianna, W. (1955). "Monte-Carlo calculations of the average extension of macromolecular chains".
7. CESAR da Silva, Julio & Fabiano Valias de Carvalho - Sensor de Batimentos Cardíacos Sensor de Batimentos Cardíacos – Inatel Disponível em: <https://www.inatel.br> > pos-seminarios > seceb > file Acesso em: 4 maio 2022.

8. MARTINS R.M.S. Desenvolvimento de um Sensor de Fotopletismografia para Monitorização Cardíaca para aplicação no Pulso, Coimbra, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, setembro de 2010.

9. TOKIO Higuti,Ricardo & Cláudio Kitano - Sinais e Sistemas - Faculdade de Engenharia de Ilha Solteira Departamento de Engenharia Elétrica Universidade Estadual Paulista Júlio de Mesquita Filho, Julho de 2003

- 10 GEDEANE Kenshima Aprenda a usar o Sensor de frequência cardíaca Disponível em: [.https://www.filipeflop.com/blog/aprenda-usar-o-sensor-de-frequencia-cardiaca/](https://www.filipeflop.com/blog/aprenda-usar-o-sensor-de-frequencia-cardiaca/) Acesso em: 4 maio 2022.

11. <https://www.Python.org/>; acesso em 22 abr. 2022, às 21:50;

12. **Matplotlib: Python plotting — Matplotlib 3.4.1 documentation.** Disponível em: <<https://matplotlib.org/>>. Acesso em: 9 abr. 2022

13. VEJA, revista- Qual é a frequência cardíaca ideal durante o exercício?<https://veja.abril.com.br/saude/qual-e-a-frequencia-cardiaca-ideal-durante-o-exercicio/> 19 set 2019, Acesso em: 13 fev. 2022

14. DIRAC, Paul (1930). Principles os Quantum Mechanics.

15. STEPHENS, Rachel- RedMonk Top 20 Languages Over Time: June 2021, 5 de Agosto de 2021- Reddith<https://redmonk.com/rstephens/2021/08/05/top-20-june-2021/> Acesso em: 15 mai. 2022

16. REDMONK, By Stephen O'Grady RedMonk Top 20 Languages Over Time, 5 de Agosto de 2021- <https://redmonk.com/sogradey/2022/03/28/language-rankings-1-22/>Acesso em: 15 mai. 2022.

8 ANEXO

8.1 Importação das bibliotecas:

```
from scipy import signal
import numpy as np
import matplotlib.pyplot as plt
```

8.2 Criação das Variáveis:

```
### Especificação das variáveis
fx_idade = np.array([20, 30, 35, 40, 45, 50, 55, 60, 65, 70])
freq_max = np.array([200, 191, 185, 180, 175, 171, 165, 160, 155, 151])
freq_min = np.array([40, 40, 40, 40, 40, 40, 40, 40, 40, 40])
freq_max = freq_max[0]
freq_min = freq_min[0]
#### batimento por minuto
freq_analise = 50
vet_freq_analise = np.array([freq_analise, 5+freq_analise, 10+freq_analise])
vet_freq_seg = vet_freq_analise/60
print('freq. por minuto', vet_freq_analise, 'freq. por segundo', vet_freq_seg)
pontos = 500
bloco_seg = 30 #####
```

8.3 Desenvolvimento das ondas senoidais:

```
##### Sinal senoidal #####
t = np.linspace(0, bloco_seg, pontos) # Define o número de pontos no intervalo supondo 30 seg
dt = t[1]-t[0] # Calcula o intervalo entre dois pontos

N = t.size # Conta o número de pontos real

R = np.array([5,5,5]) # Define as amplitudes de 3 ondas senoidais

fase = np.array([0.0, 0*dt, 0*dt]) # Define as fases das 3 ondas senoidais de acordo com dt

##### Obtenção das 3 ondas senoidais #####
s1 = R[0]*np.sin(vet_freq_seg[0]*2*np.pi*t+fase[0])
s2 = R[1]*np.sin(vet_freq_seg[1]*2*np.pi*t+fase[1])
s3 = R[2]*np.sin(vet_freq_seg[2]*2*np.pi*t+fase[2])
plt.figure(figsize=(10,5))
plt.plot(t,s1)
plt.plot(t,s2)
plt.plot(t,s3)
plt.ylabel("Amplitude")
plt.xlabel("Tempos s")
```

8.4 Transformada de Fourier das Ondas senoidais:

```
##### Transformada de Fourier
sinal = s1 + s2 + s3
fft = np.fft.fft(sinal)
T = t[1]-t[0]
N = sinal.size          # numero de dados da serie
f = np.fft.fftfreq(len(sinal),T)
frequencias = f[:N//2]
amplitudes = np.abs(fft)[:N//2]

plt.figure(figsize=(10,6))
plt.ylabel("Amplitude")
plt.xlabel("Frequencias Hz")
plt.bar(frequencias,amplitudes, width = 0.05)
```

8.5 Desenvolvimento do ruído branco

```
##### Inserção de Ruído Branco
##### Ruído como Gaussiana
mu=0.0
sigma=1
ruído = np.random.normal(mu, sigma, N)

plt.figure(figsize=(10,6))
plt.plot(t,ruído)
plt.ylabel("Ruído Branco")
plt.xlabel("Tempos s")
```

8.6 Transformada de Fourier ruído branco:

```
sinal = ruído
fft = np.fft.fft(sinal)
T = t[1]-t[0]
N = sinal.size          # numero de dados da serie
f = np.fft.fftfreq(len(sinal),T)
frequencias = f[:N//2]
amplitudes = np.abs(fft)[:N//2]
#amplitudes = np.abs(fft)[:N//2]*1/N
#amplitudes = np.abs(fft)[:N//2]
plt.figure(figsize=(10,6))
plt.ylabel("Amplitude")
plt.xlabel("Frequencias Hz")
plt.bar(frequencias,amplitudes, width = 0.05)
```

8.7 Transformada de Fourier da onda senoidal com ruído branco:

```
# Sinal com frequência de 100 batimento po minuto
# Ruído Branco com desvio padrão variando de 0,20 a 2,00
mu=0.0
sigma=0.5 # PEDRO alterar o desvio padrão em 0.5 1.0 2.0 e 4.0
ruído = np.random.normal(mu, sigma, N)
sinal = s2 + ruído
plt.figure(figsize=(10,6))
plt.plot(t,sinal)
plt.ylabel("Ruído Branco")
plt.xlabel("Tempos s")
```


8.8 Transformada de Fourier do sinal senoidal com ruído branco alterando o valor do σ :

```
# gerar sinal com ruído normalmente distribuído com variância alternado de 0,50 a 2,00
i=1
mu = 0
sigma = 0.0
N = s2.size
T = t[1]-t[0]
plt.figure(figsize=(15,10))
ruído = np.random.normal(mu, 10, N)
sinal = s2 + ruído
fft = np.fft.fft(sinal)
f = np.fft.fftfreq(len(sinal),T)
frequencias = f[:N//2]
amplitudes = np.abs(fft)[:N//2]
plt.subplot(2, 2, 1) # gera uma matriz 2x2 de gráficos
plt.bar(frequencias, amplitudes, color='black', width = 0.05, label='sigma = 0,5')
teste = amplitudes
I = np.argsort(teste) #
print('frequencia por minuto: ',60*frequencias[I[len(I)- i]])
ruído = np.random.normal(mu, 8, N)
sinal = s2 + ruído
fft = np.fft.fft(sinal)
f = np.fft.fftfreq(len(sinal),T)
frequencias = f[:N//2]
amplitudes = np.abs(fft)[:N//2]
plt.subplot(2, 2, 2) # gera uma matriz 2x2 de gráficos
plt.bar(frequencias, amplitudes, color='black', width = 0.05, label='sigma = 1,0')
teste = amplitudes
I = np.argsort(teste) #
print('frequencia por minuto: ',60*frequencias[I[len(I)- i]])
ruído = np.random.normal(mu, 18, N)
sinal = s2 + ruído
fft = np.fft.fft(sinal)
f = np.fft.fftfreq(len(sinal),T)
frequencias = f[:N//2]
amplitudes = np.abs(fft)[:N//2]
plt.subplot(2, 2, 3) # gera uma matriz 2x2 de gráficos
plt.bar(frequencias, amplitudes, color='black', width = 0.05, label='sigma = 2,0')
teste = amplitudes
I = np.argsort(teste) #
print('frequencia por minuto: ',60*frequencias[I[len(I)- i]])
ruído = np.random.normal(mu, 6, N)
sinal = s2 + ruído
fft = np.fft.fft(sinal)
f = np.fft.fftfreq(len(sinal),T)
frequencias = f[:N//2]
amplitudes = np.abs(fft)[:N//2]
plt.subplot(2, 2, 4) # gera uma matriz 2x2 de gráficos
plt.bar(frequencias, amplitudes, color='black', width = 0.05, label='sigma = 4,0')
teste = amplitudes
I = np.argsort(teste) #
print('frequencia por minuto: ',60*frequencias[I[len(I)- i]])
```

8.9 Desenvolvimento onda Quadrada:

```
# Onda Quadrada
q1 = s1
q2 = np.ones(len(s1))
##### Criando MASCARA
mask1 = (q1 < 0)
q2[mask1]=0
quadrada = q2
plt.figure(figsize=(20,3))
plt.plot(t,quadrada)
plt.ylabel("Amplitude")
plt.xlabel("Tempos s")
```

8.10 Transformada de Fourier da onda quadrada:

```
fft = np.fft.fft(quadrada)
T = t[1]-t[0]
N = quadrada.size          # numero de dados da serie
f = np.fft.fftfreq(len(quadrada),T)
frequencias = f[:N//2]
amplitudes = np.abs(fft)[:N//2]
plt.figure(figsize=(20,3))
plt.ylabel("Amplitude")
plt.xlabel("Frequencias Hz")
plt.bar(frequencias,amplitudes, width = 0.05)
```

```
maiores = 5
count = 0
teste = amplitudes
#quantidade = len(teste)
I = np.argsort(teste) #
##### Comando para avaliar as maiores frequencias, no caso de senoidal não é necessário descarta o
primeiro
i=1
while i < 4:
    print('frequencia por minuto: ',60*frequencias[I[len(I)- i]])
    i+=1
#####
```

8.11 Desenvolvimento do “delta de Dirac”:

```
# DELTA DIRAC
q1 = s1
q2 = np.ones(len(s1))
##### Criando MASCARA
mask1 = (q1 < 0.97*R[0])
q2[mask1]=0
delta = q2
```

```
plt.figure(figsize=(20,3))
plt.plot(t,delta)
plt.ylabel("Amplitude")
plt.xlabel("Tempos s")
```

8.12 Transformada de Fourier da onda “delta de Dirac”:

```
sinal = delta
fft = np.fft.fft(sinal)
T = t[1]-t[0]
N = sinal.size          # numero de dados da serie
f = np.fft.fftfreq(len(sinal),T)
frequencias = f[:N//2]
amplitudes = np.abs(fft)[:N//2]
#amplitudes = np.abs(fft)[:N//2]*1/N
#amplitudes = np.abs(fft)[:N//2]
plt.figure(figsize=(10,6))
plt.ylabel("Amplitude")
plt.xlabel("Frequencias Hz")
plt.bar(frequencias,amplitudes, width = 0.05)
```

8.13 Desenvolvimento da onda dente de serra:

```
#t = np.linspace(0.0,0.5,500)
serra = signal.sawtooth(2*vet_freq_seg[0]*np.pi*t)
plt.figure(figsize=(12,6))
plt.plot(serra)
plt.ylabel("Amplitude", fontsize='14')
plt.xlabel("Tempos s", fontsize='14')
```

8.14 Desenvolvimento carta de controle:

```
##### Looping de num_pt_carta
num_pt_carta = 30
amostra = np.arange(1,num_pt_carta+1)
freq_fft = []
media_freq = 80
dp_freq = media_freq*0.1
media_freq = media_freq/60
dp_freq = dp_freq/60
media_ruido = 0
dp_ruido = 0.1
amplitude = 1
fase = 0
t = np.linspace(0, bloco_seg, pontos)
for i in range(0,num_pt_carta):
    if (i < 15):
```

```

    dp_freq = media_freq*0.1
    aleatorio = np.random.normal(0, dp_freq, 1)
else:
    dp_freq = media_freq*0.3
    aleatorio = np.random.normal(0, dp_freq, 1)
    freq = media_freq + aleatorio
    serra = signal.sawtooth(2*freq*np.pi*t)
    N = s.size
    ruido = np.random.normal(media_ruido, dp_ruido, N)
    sinal = serra + ruido
    fft = np.fft.fft(sinal)
    T = t[1] - t[0]
    f = np.fft.fftfreq(len(sinal),T)
    frequencias = f[:N//2]
    amplitudes = np.abs(fft)[:N//2]*1/N
    teste = amplitudes
    I = np.argsort(teste)
    freq_amostra = 60*frequencias[I[len(I)-1]]
    freq_fft.append(freq_amostra)

```

```

plt.figure(figsize=(12,6))
plt.ylabel("Frequencia", fontsize='14')
plt.xlabel("amostra", fontsize='14')
plt.plot(amostra,freq_fft)
plt.axhline(y = 160, color = 'r', linestyle = '-')
plt.axhline(y = 50, color = 'r', linestyle = '-')

```