P1 - Processamento de Áudio e Imagem

Gabriel Tavares 10773801

Q1

a)

Forne¸ca os valores da TDF para k ≥ 60.

A TDF é uma função períodica em **k** com um período igual ao número de pontos usados para seu cálculo. Nesse exercício temos um número de pontos N = 120, portando a cada 120 pontos a função irá se repetir. Além disso também temos a propriedade de frequências espelhadas negativas tem o coeficiente conjulgado (X[2] = X[-2]*).

Portanto segue a progressão :

X[120] = X[0] | X[119] = X[1]* | X[118] = X[2]* | ... | X[60] = X[59]*

Então temos:

$$X[61] = X[62] = X[63] = ... = X[115] = 0$$

$$X[120] = -60$$

$$X[119] = 0$$

$$X[118] = 30e^{j\pi/3}$$

$$X[117] = 120j$$

$$X[116] = 40e^{-j\pi/4}$$

b)

Forne¸ca a expressão de x(t)

Usando a propriedade de antitransformada da TDF, temos:

$$\frac{-60}{120} + \frac{0cos(\omega_0 n)}{120} + \frac{2 \cdot 30cos(2\omega_0 n - \frac{\pi}{3})}{120} + \frac{2 \cdot -120cos(3\omega_0 n + \frac{\pi}{2})}{120} + \frac{2 \cdot 40cos(4\omega_0 n +}{120}$$

E sabendo que:

$$\omega_o = \frac{2\pi}{N} = \frac{\pi}{60}$$

Portanto temos:

$$x[n] = -0.5 + 0.5cos(\frac{2\pi}{60} - \pi/3) - 2cos(\frac{3\pi}{60} + \pi/2) + \frac{2}{3}cos(\frac{4\pi}{60} + \pi/4)$$

Para saber o sinal em tempo contínuo usamos $\Omega = \omega \cdot fa = \omega \cdot 40kHz$, então:

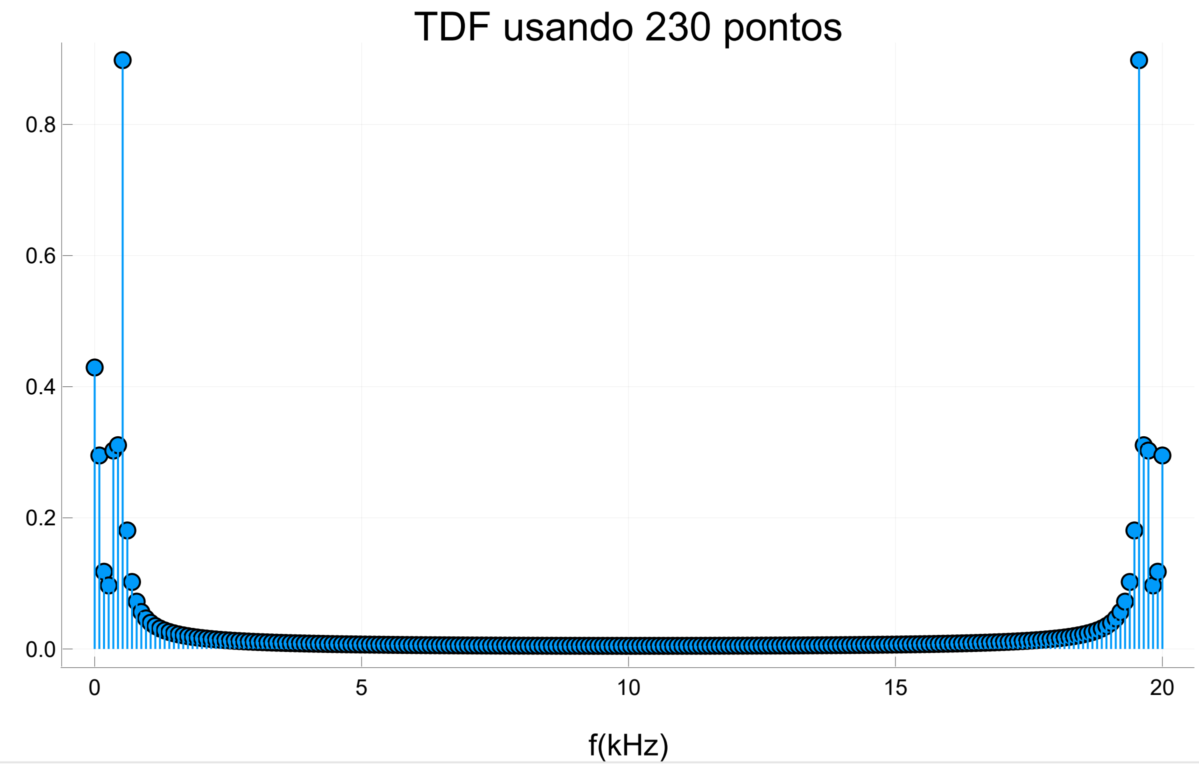
$$x(t) = -0.5 + 0.5cos(\frac{2\pi}{60}40000t - \pi/3) - 2cos(\frac{3\pi}{60}40000t + \pi/2) + \frac{2}{3}cos(\frac{4\pi}{60}40000t + \pi/$$

Ou simplificando:

$$x(t) = -0.5 + 0.5cos(\frac{2000}{3}2\pi t - \pi/3) - 2cos(1000 \cdot 2\pi t + \pi/2) + \frac{2}{3}cos(\frac{4000}{3}2\pi t + \pi/4)$$

c)

Use o sinal reconstru´ıdo do item anterior para calcular a TDF usando N = 230 pontos para o c ´alculo. Desenhe o gr´afico do m´odulo da TDF obtida. O eixo x do seu gr´afico deve ser dado em termos da frequ^encia em Hz correspondente a cada raia da TDF.




```
begin
    fa1 = 40_000
    Ta1 = 1/fa1
    N1c = 230
    t = range(0,length = N1c , step= Ta1)
    x1 = -0.5 .- 0.5*cos.(2000/3*2π*t .- π/3) .- 2*cos.(1000*2π*t .+ π/2) .+ 2/3*
cos.(4000/3*t .+ π/4)
    X1c = fft(x1)
    f1c = range(0,fa1/2, length = N1c)
    plot(f1c/1000,abs.(X1c)/N1c, line=:stem, marker=:circle)
    plot!(title = "TDF usando 230 pontos", legend = false, xlabel = "f(kHz)")
end
```

d)

Qual ´e o per´ıodo de x(t)?

O período do sinal vale o MMC entre os períodos dos 3 cossenos

$$T1 = 6/4000$$

$$T2 = 4/4000$$

$$T3 = 3/4000$$

Então aplicando mmc

$$T = 12/4000 = 3ms$$

3)

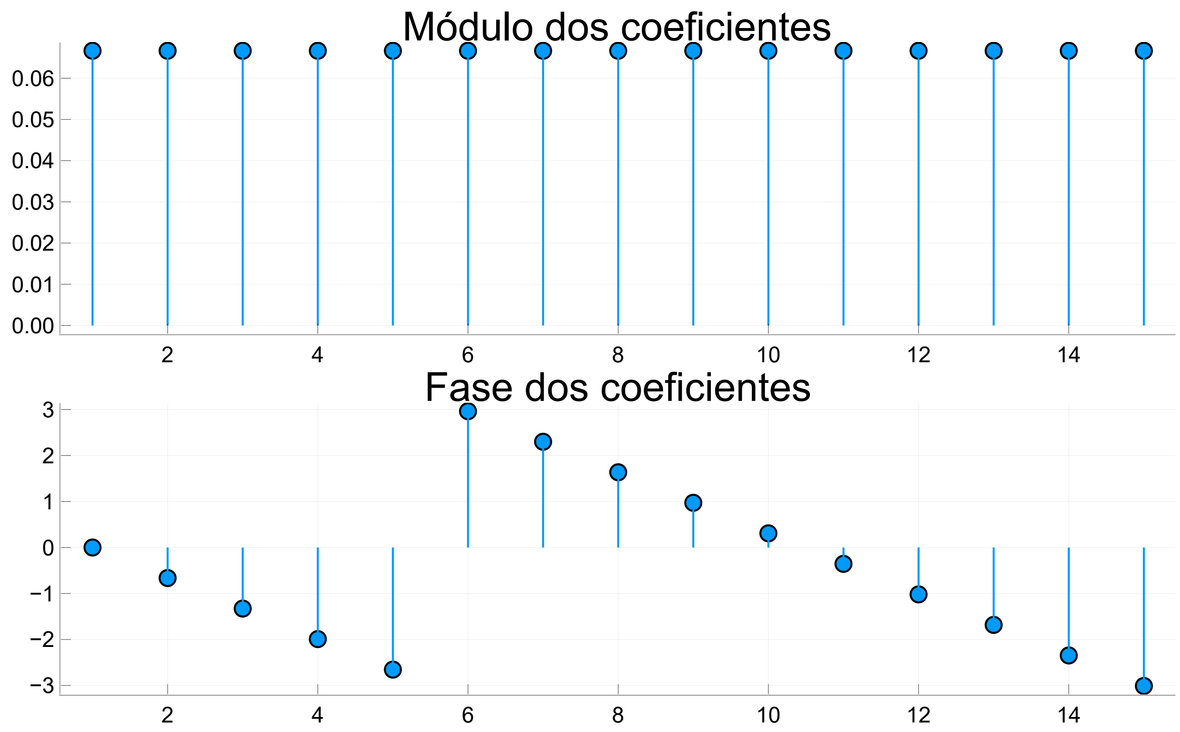
Parâmetros do arranjo:

```
-25
```

```
begin
    M3 = 15
    f03 = 100_000_000 #100MHz
    Ω03 = 2π*f03
    c = 3e8
    λ3 = c/f03
    d3 = λ3/4 #m
    θ0 = 25 #graus
    θ1 = -25
end
```

Primeiramente calculei os coeficientes do filtro de atraso e soma pelas fórmulas usadas em sala.

$$w_k = \frac{1}{M} e^{j\Omega_0 \tau_k(\theta_0)}$$

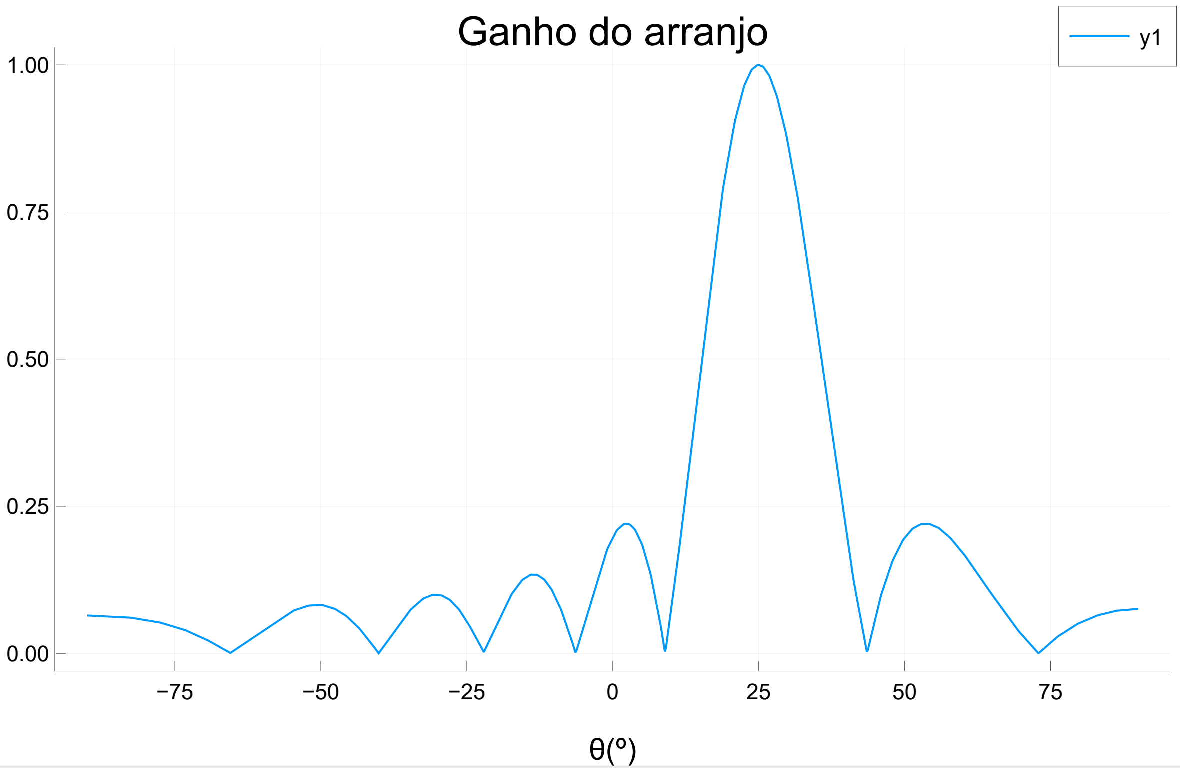



```
begin
    #coeficientes do filtro espacial DAS
    j = im
    w = ones(Complex, M3)
    for k in 0:M3-1
        w[k+1] = 1/M3*exp(-j*Ω03*tau(k,θ0,d3))
    end

    p3_abs = plot(abs.(w), line=:stem, marker=:circle, title="Módulo dos coeficientes")
    p3_angle = plot(angle.(w), line=:stem, marker=:circle, title ="Fase dos coeficientes")
    plot(p3_abs,p3_angle, layout = (2,1), legend = false)
end
```

Depois, calculei o ganho que teremos em cada angulo de chegada do sinal

$$B(\theta, \theta_o) = e^{j\frac{M-1}{2}\frac{\Omega_o d}{c}(sen(\theta)-sen(\theta_o))} \frac{sen(\frac{M\Omega_o d}{2c}(sen(\theta) - sen(\theta_o)))}{sen(\frac{\Omega_o d}{2c}(sen(\theta) - sen(\theta_o)))}$$




```
begin
    θ = range(-90,90,length = 1000)
    Bs = B.(θ, θ0, M3, d3, Ω03)

    plot(θ,abs.(Bs))
    plot!(title =  "Ganho do arranjo", xlabel = "θ(º)", )
end
```

Por fim calculei o ganho de um sinal chegando de -25º . $B(-25^o, 20^o) =$ **0.055** $\angle$ **0.1309**

B_θ1 = 0.054501 + 0.007176im

```
B_θ1 = round(B(θ1, θ0, M3, d3, Ω03), digits = 6)
```

Q4

a)

Quais s˜ao os fatores de convers˜oes de taxa intermedi´arios L e M?

Sabemos que $fa' = L/M fa$, então $L/M = 40k/25k = 8/5$.

Portanto **L=8** e **M=5**.

5

```
begin
    L4= 8
    M4 = 5
end
```

b)

No sinal de taxa elevada (interpolado por zeros), em que frequ^encias est˜ao centradas as imagens do espectro do sinal original? Qual a frequ^encia de corte e o ganho do filtro passa-baixa necess ´ario para remov^e-las? Haver´a perda de informa¸c˜ao na convers˜ao?

Os espectros do sinal original estão centrados em $\omega = 2\pi/L = 2\pi/8$.

A frequência de corte de um filtro ideal para a reamostragem é em $\omega_c = \pi/L = \pi/8$

A frequência máxima do sinal está em $f = \omega/2\pi \cdot fa = 3\pi/8\pi \cdot 25k = 9,375kHz$. Quando reamostrarmos o sinal a uma frequência maior, a máxima frequência possível no sinal será de 20kHz. Como a maior frequência do sinal original é menor do que a máxima frequência, não haverá nenhuma distorção.

c)

Agora voc^e vai projetar o filtro passa-baixa com janela de Kaiser. Determine os limites da banda de passagem e da banda de rejei¸c˜ao do filtro, e determine os par^ametros N e β da janela de Kaiser. Dados: distor¸c˜ao m´axima na banda-passante do sinal δp = ±0,01, oscila¸c˜ao m´axima na banda de rejei¸c˜ao δr = 0.001 (lembre que as oscila¸c˜oes δ consideradas no projeto da janela de Kaiser s˜ao relativos a um filtro de ganho unit´ario). Apresente um gr´afico com a resposta ao impulso e outro com a resposta em frequ^encia do filtro projetado

Determinamos a faixa de passagem e de rejeição do filtro a partir da frequência máxima do sinal original. Essa frequência máxima será achatada por um fator L, portanto $\omega_p = \omega_m/L$.

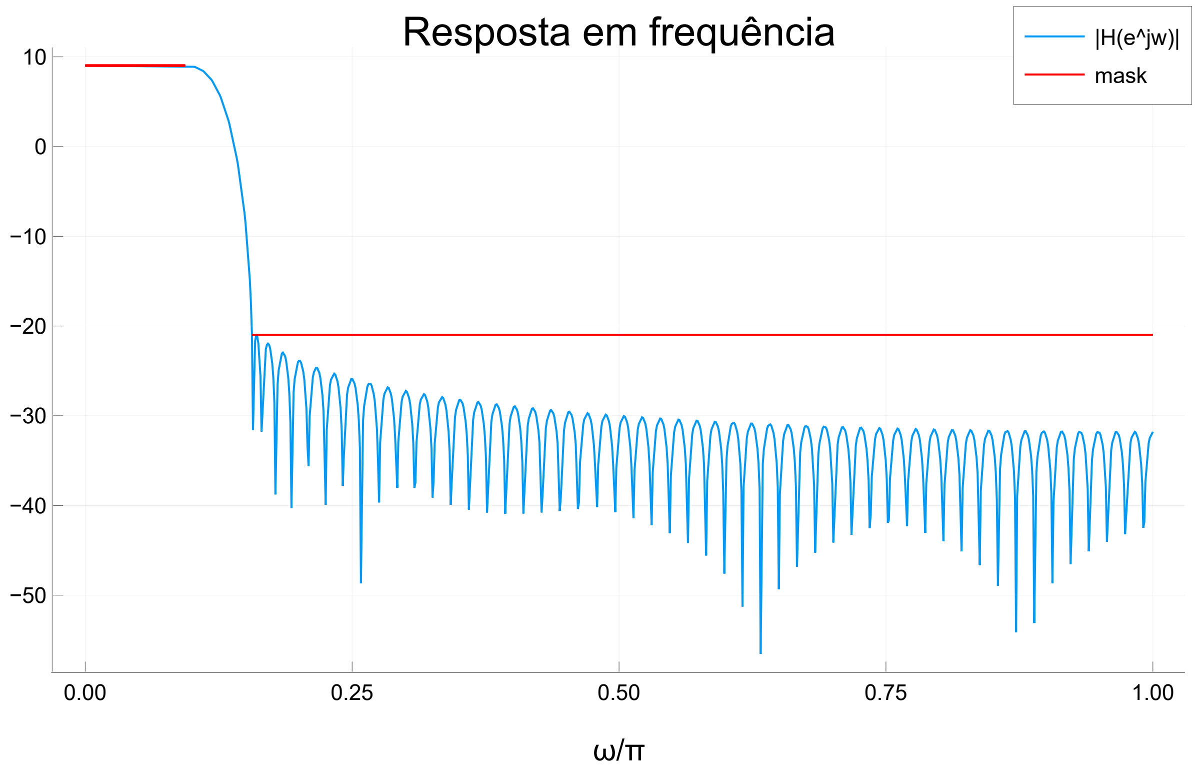
A faixa de rejeição é determinada pelas imagens do espectro que estão centras em $2\pi/L$. Queremos rejeitas estes espectros inteiros, portanto faremos $\omega_r = 2\pi/L - \omega_m/L$.

Além disso o filtro deve ter um ganho de L.

```
[-0.000632953, -0.000450468, 0.0, 0.000716145, 0.00161997, 0.0025537, 0.00329482, 0.0035
```

```
begin
    δp4 = 0.01
    δr4 = 0.001
    ω_max4 = 3π/4
    ωp4 = ω_max4/L4
    ωr4 = (2π - ω_max4)/L4
    pb4 = L4*kaiser_filter_lowpass(δp4, δr4, ωp4, ωr4)
end
```


```
begin
    plot(pb4, label = "h_kaiser")
    plot!(title="Resposta ao impulso")
end
```




```
begin
    ω4 = range(0,π, length = 1000)
    pb4_f = PolynomialRatio(pb4,[1])
    PB4 = freqz(pb4_f, ω4)

    plot(ω4/π, pow2db.(abs.(PB4)), label ="|H(e^jw)|")
    plot!([ωr4/π, 1],pow2db.([L4*δr4, L4*δr4]), color =:red, label ="mask")
    plot!([0, ωp4/π],pow2db.([L4+L4*δp4, L4+L4*δp4]), color =:red, label = missing)
    plot!([0, ωp4/π],pow2db.([L4-L4*δp4, L4-L4*δp4]), color =:red, label = missing)
    plot!(title = "Resposta em frequência", xlabel = "ω/π" )
end
```

Anexos

Funções e bibliotecas usadas no código

```
PlotlyBackend()
```

```
begin
    using DSP
    using FFTW
    using Plots
    using MAT
    plotly()
end
```

Funções para arranjo de antenas

```
tau (generic function with 1 method)
```

```
function tau(k, θin, d)
    θ = deg2rad(θin)
    c= 3e8
    return k*d*sin(θ)/c
end
```

B (generic function with 1 method)

```
function B(θin, θ0in, M, d, Ω0)
    θ = deg2rad(θin)
    θ0 = deg2rad(θ0in)

    c = 3e8
    exponencial = exp( im*(M-1)/2 * Ω0*d/c *( sin(θ)-sin(θ0) ) )

    if  (sin(θ)-sin(θ0))==0.0
            senos = M
    else
        senos = sin(M*Ω0*d/(2c) * (sin(θ)-sin(θ0)) ) / ( sin( Ω0*d/(2c) * (sin(θ)-
sin(θ0) )) )
    end
    valor =exponencial *1/M* senos

    return valor

end
```

Funções para projeto de filtro kaiser

kaiser_filter_lowpass (generic function with 1 method)

```
function kaiser_filter_lowpass(δp, δr, ωp, ωr)
    #retorna um filtro passa baixas apenas
    Δω = ωr - ωp
    A= -20log10(min(δp, δr))
    Nk = ceil(Int,(A - 8) / (2.285 * Δω) + 1)
    β = kaiserbeta(δp, δr, Δω)
    nk = 0:Nk-1
    kaiser_window = kaiser(Nk,β/π)

    Lk = (Nk -1)÷2
    ωc = (ωr+ωp)/2
    h = ωc/π * sinc.(ωc/π .* (nk.-Lk))
    hk = h.*kaiser_window
    return hk
end
```

kaiserbeta (generic function with 1 method)

```
function kaiserbeta(δp, δr, Δω)
     δ = min(δp, δr)
     A = -20log10(δ)
     if A < 21
        return 0.0
     elseif A ≤ 50
        return 0.5842(A-21)^0.4 + 0.07886(A-21)
     else
        return 0.1102(A-8.7)
     end
end
```

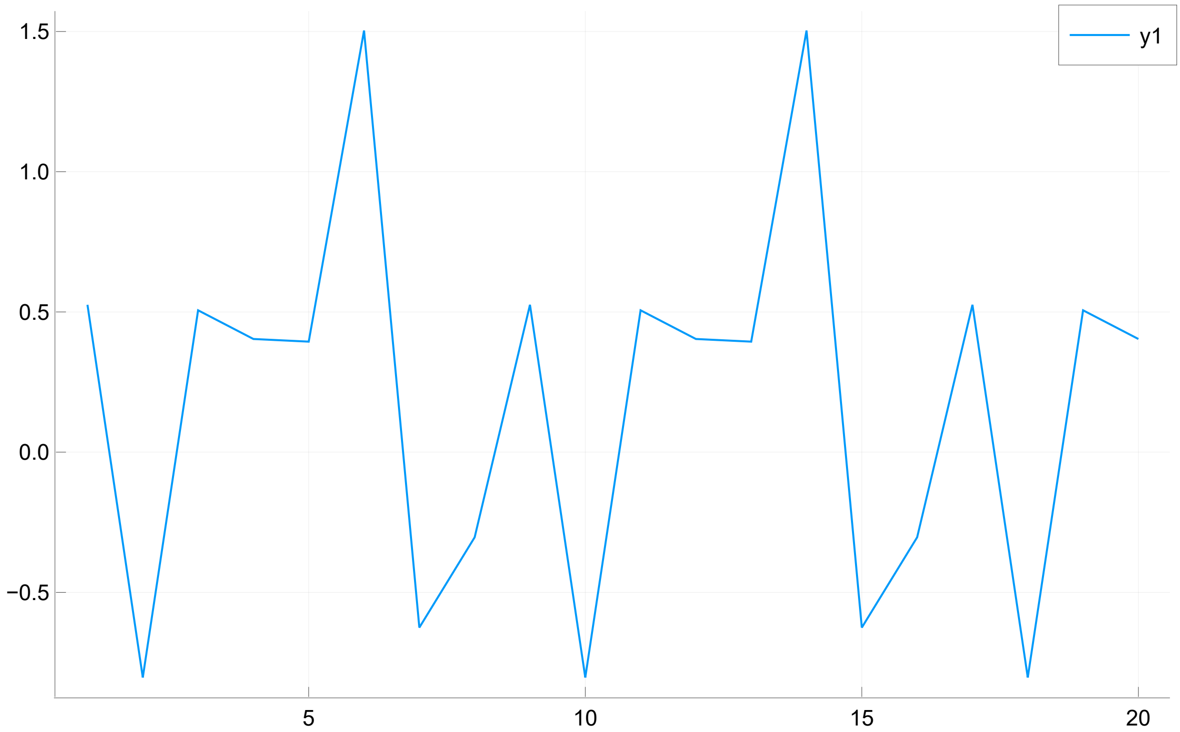
2 -> DESCONSIDERAR

Fiz essa questão e me arrependi, use a correção dela apenas caso ache pertinente

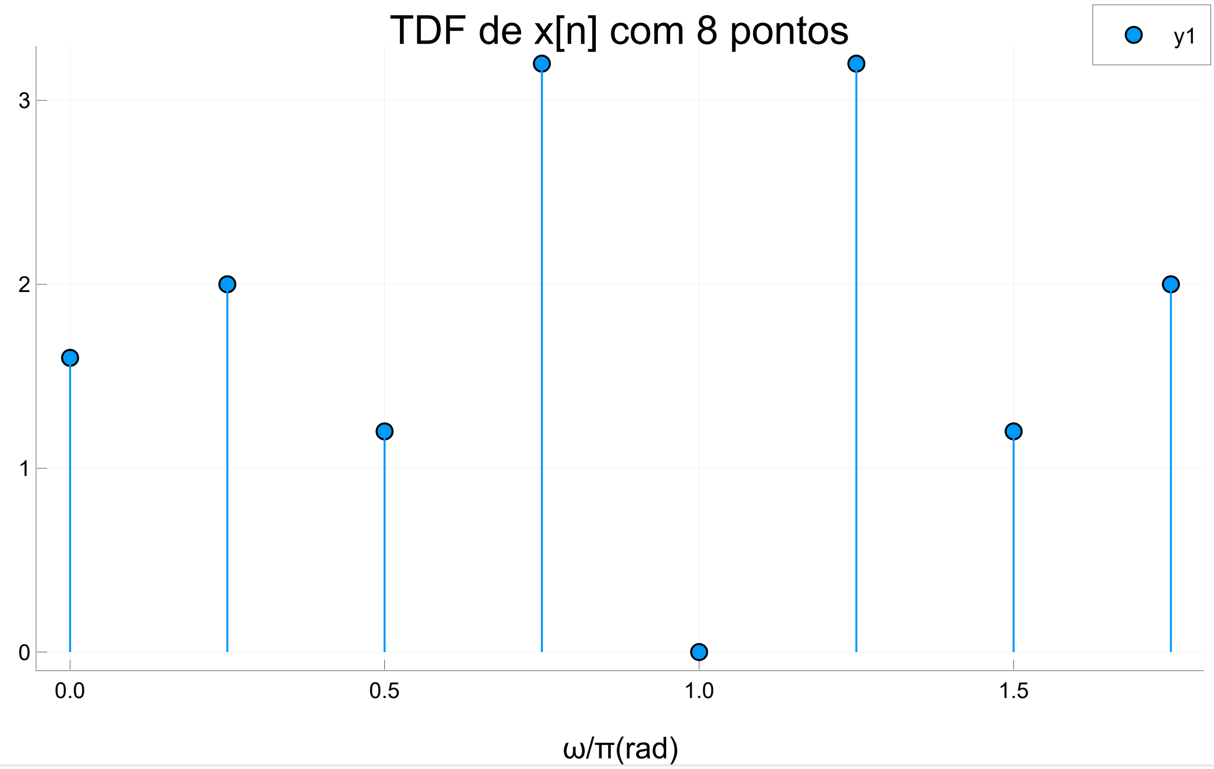
Para este sinal temos fo = 500Hz, e portanto ωo = π/4 rad.

Sendo assim, usamos N = 2π/ωo = 8.

Essee valor pode ser verificado no gráfico abaixo.




```
begin
    fa2 = 4_000
    Ta2 = 1/fa2
    var = matread("sinal.mat")
    x2 = var["x"]
    plot(x2[1:20])
end
```




```
begin
    N2 = 8
    X2_TDF = fft(x2[1:N2])
    ω02 = 2π/N2
    ω2 = range(0, step= ω02, length = N2)
    plot(ω2/π , abs.(X2_TDF), line=:stem, marker=:circle)
    plot!(title="TDF de x[n] com 8 pontos", xlabel = "ω/π(rad)")
end
```

Podemos ter os coeficientes da Série de Fourier a partir do coeficientes da TDF apenas multiplicando os coeficientes por Ta