

# Experiência 01 - Beamforming

Gabriel Tavares - 10773801

Guilherme Reis - 10773700

Diego Hidek Caetano - 10336622

PlotlyBackend()

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

## Declaração da função de ganho espacial

Aplicação da fórmula vista em aula

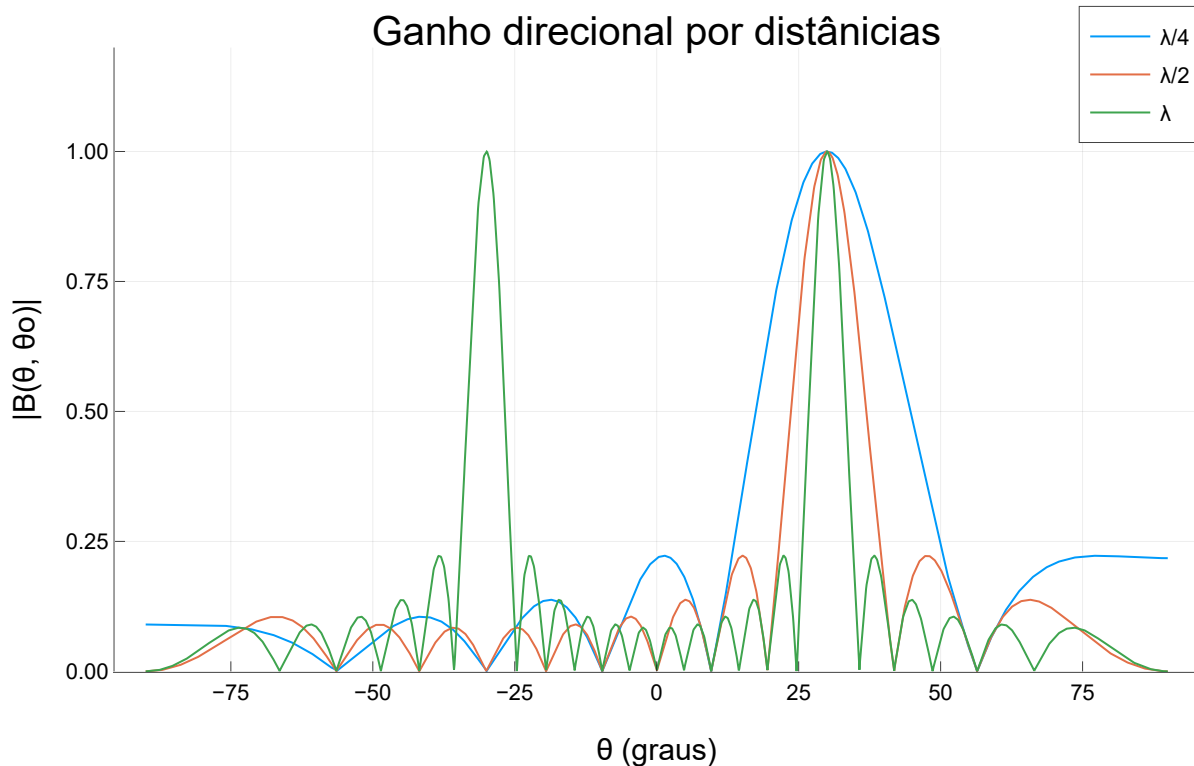
$$B(\theta, \theta_o) = e^{j\frac{M-1}{2} \frac{\Omega_o d}{c} (\sin(\theta) - \sin(\theta_o))} \frac{\text{sen}\left(\frac{M\Omega_o d}{2c} (\sin(\theta) - \sin(\theta_o))\right)}{\text{sen}\left(\frac{\Omega_o d}{2c} (\sin(\theta) - \sin(\theta_o))\right)}$$

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
```

## 1) Gráfico de ganho para M = 12

Nesta seção realizamos uma simulação de um arranjo de 12 antenas com diferentes espaçamentos em função da frequência de entrada na do sistema. O arranjo terá uma ganho apontado para  $\theta_o = 30^\circ$  e trabalha em uma frequência de 500 MHz.



```

• begin
•     c= 3e8
•     θ₀ = 30
•
•     θ = range(-90,90, length = 2000)
•     M = 12
•
•     f = 500e6
•     Ω₀ = 2π*f #
•     λ = c/f
•     d1 = [λ/4, λ/2, λ]
•
•     Bs=zeros(Complex,length(θ), 3)
•
•     for i in 1:3
•         Bs[:,i]=B.(θ, θ₀, M, d1[i], Ω₀)
•     end
•
•     plot(θ,abs.(Bs),label=["λ/4" "λ/2" "λ"])
•
•     plot!(title="Ganho direcional por distâncias", xlabel="θ (graus)",
•         ylabel="|B(θ, θ₀)|",ylim=(0,1.2) )
• end

```

## Conclusão

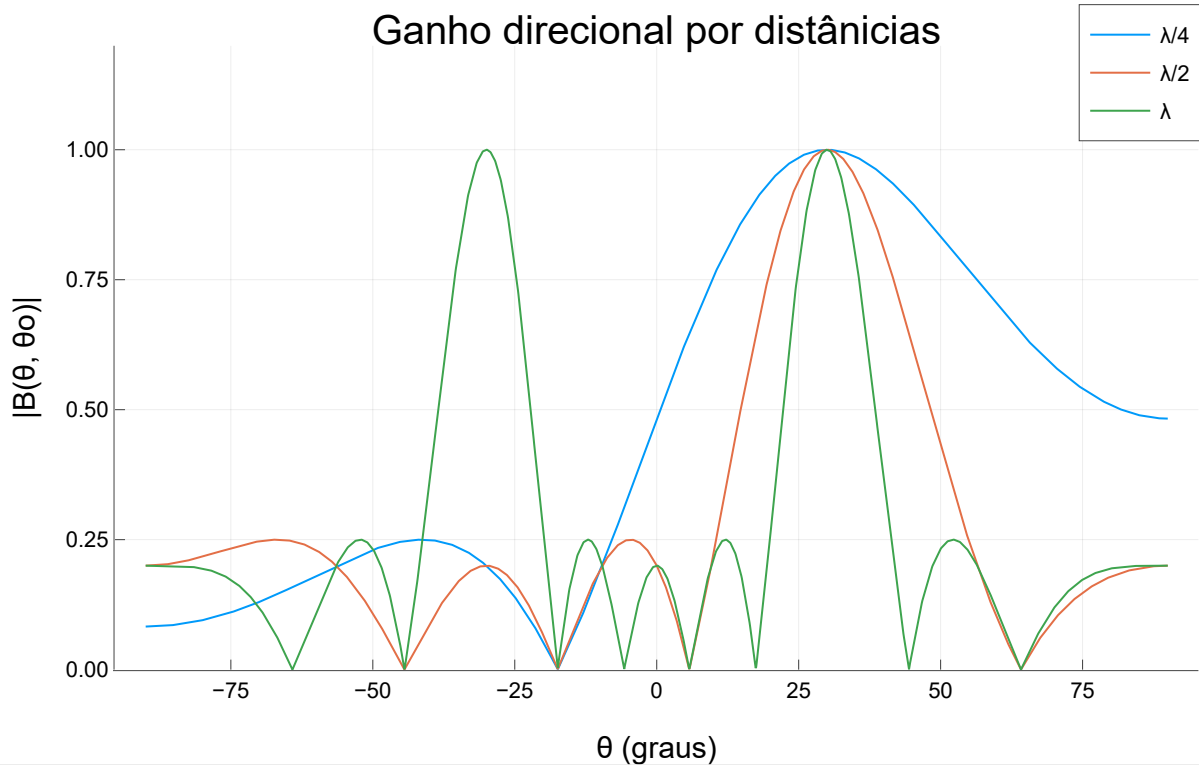
Nesta simulação vemos que é possível realizar um filtro espacial direcional com um arranjo simples de antenas.

Observa-se também que para distâncias de antenas maiores do que  $\frac{\lambda}{2}$  pode existir mais de um pico de ganho máximo dependendo da direção de foco do arranjo  $\theta_o$ .

## 2) Arranjo de 5 antenas

Nesta seção iremos repetir o processo anterior para um número menor de antenas.

Caso esteja usando o Pluto pode utilizar o fader abaixo para alterar o número de antenas



```

• begin
•     Bs2=zeros(Complex,length(θ), 3)
•     M2 = 5
•     for i in 1:3
•         Bs2[:,i]=B.(θ, θ₀, M2, d1[i], Ω₀)
•     end
•
•     plot(θ,abs.(Bs2),label=["λ/4" "λ/2" "λ"] )
•
•     plot!(title="Ganho direcional por distâncias", xlabel="θ (graus)",
•           ylabel="|B(θ, θ₀)|",ylim=(0,1.2) )
• end

```

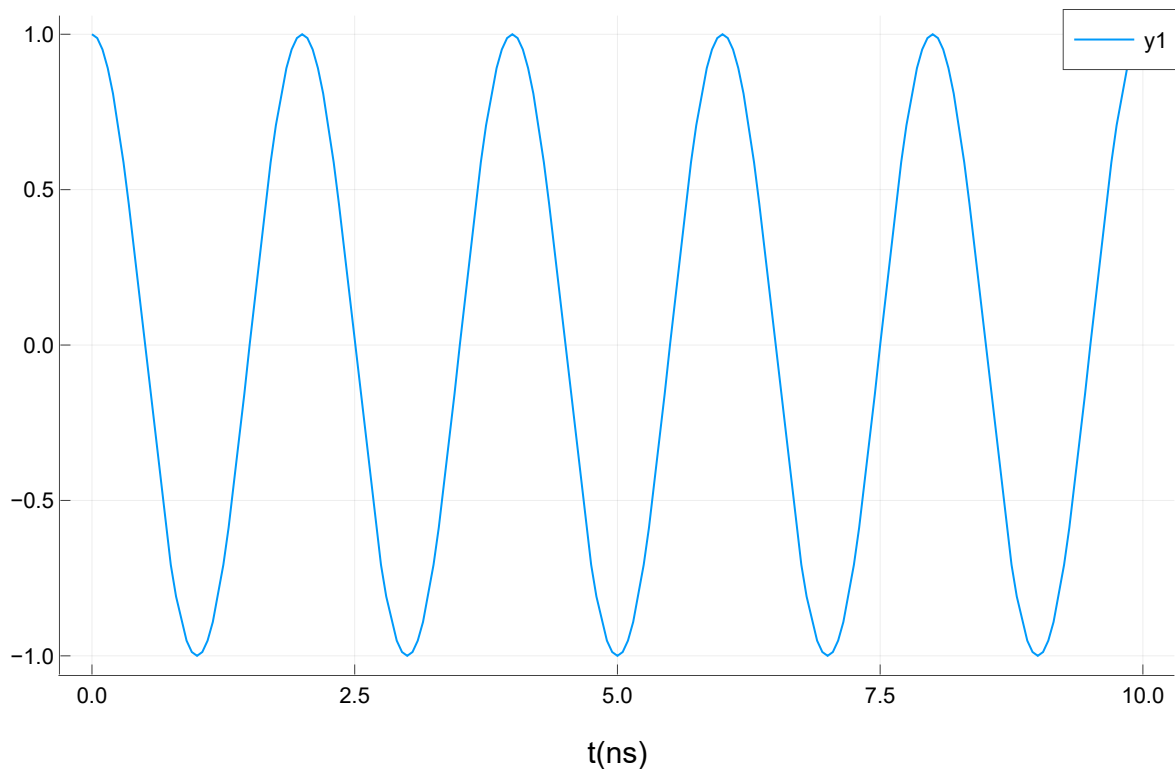
## Conclusão

Nesta simulação é possível observar que o **aumento do número de antenas** faz com que o sistema seja **mais diretivo** para uma mesma distância entre antenas.

## 3) Sinal ideal

Nesta etapa iremos observar teoricamente o sinal de um arranjo quando a antena recebe um sinal imaginário.

Considerando o sinal  $x_o(t) = e^{j\Omega t} + e^{j\Omega t}$ , sendo a primeira parcela vinda de um ângulo de  $30^\circ$  e a segunda de um ângulo de  $0^\circ$ .



```

• begin
•     t3 = range(0,0.00000001,step=1/(40f))
•
•     j = im
•     d = d1[2]
•     x1 = exp.(j*Ω0*t3) #sinal vindo de 30°
•     θ1 = 30
•     x2 = exp.(j*Ω0*t3) #sinal vindo de 0°
•     θ2=0
•
•
•     y3 = x1 * B(θ1, θ0, M, d, Ω0) + x2*B(θ2,θ0,M,d,Ω0)
•     plot(1000000000*t3,real.(y3))
•     plot!(xlabel= "t(ns)")
• end

```

## 4)

33.035998057806054

```

• begin
•     db(x) = 20*log10(x)
•     SNR0 = db(abs(B(30, θ0, M, d, Ω0)/B(0, θ0, M, d, Ω0)))
•     SNR1 = db(abs(B(30, θ0, M, d, Ω0)/B(10, θ0, M, d, Ω0)))
• end

```

## Relção S/N na entrada

Na entrada do arranjo de antenas, o sistem recebe dois sinais vindo de direções diferentes. Como os dois sinais, neste caso, tem mesma amplitude, a relação Sinal/Ruído é o.

$$SNR = 20\log\left(\frac{x_1}{x_2}\right) = 0$$

Já na saída do arranjo, onde temos um processamento, a relação sinal ruído é dada por

$$SNR(\theta_1, \theta_2) = 20\log\left(\frac{x_1 B(\theta_1, \theta_0)}{x_2 B(\theta_2, \theta_0)}\right) = 20\log\left(\frac{B(\theta_1, \theta_0)}{B(\theta_2, \theta_0)}\right)$$

Numericamente temos:

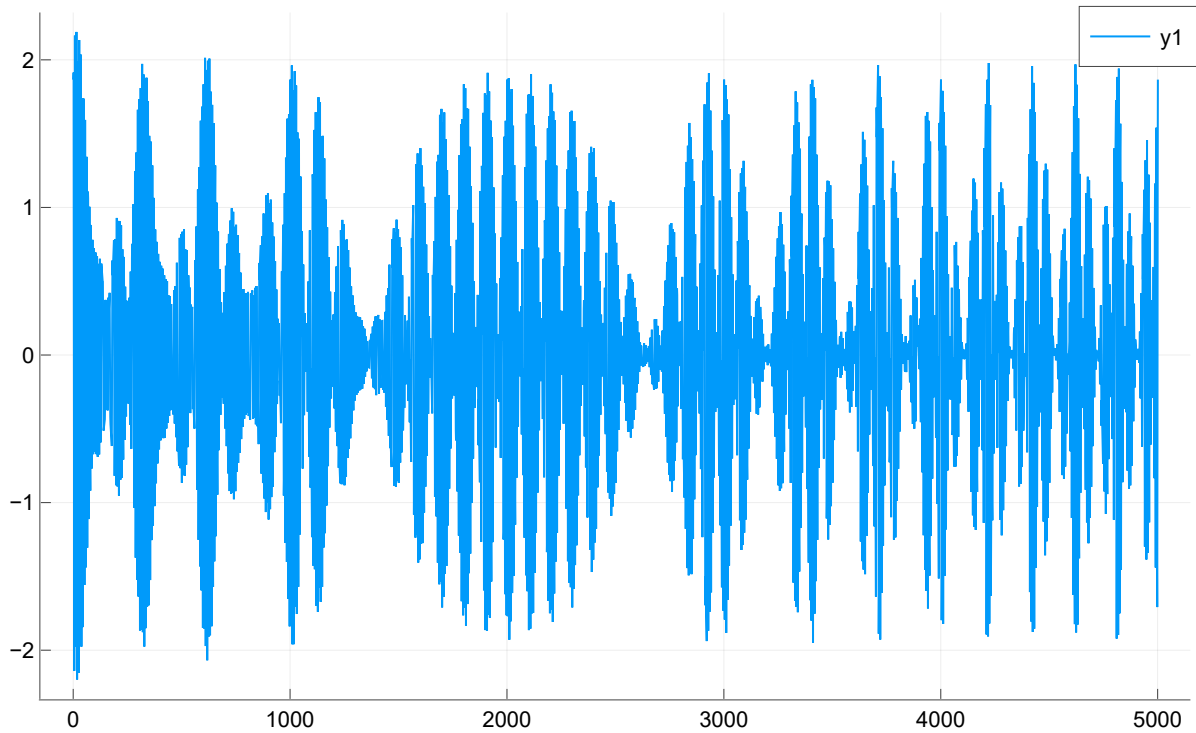
$$SNR(30^\circ, 0^\circ) = 306.71\text{dB}$$

$$SNR(30^\circ, 10^\circ) = 33.04\text{dB}$$

## 5

---

### Leitura do sinal



```
• begin
•     x = matread("arranjo.mat")["x"]
•
•     fi = 150e3 #Hz
•     Ωi = 2π*fi #rad/s
•     fa = 1e6 #Hz
•     Ta = 1/fa
•     θ5 = 30 #graus
•
•     plot(x[:,1])
•
• end
```

## Criação do filtro

Para essa etapa criamos um filtro com frequência de corte de 150kHz pois é a frequência da portadora do sinal quando amostrado.

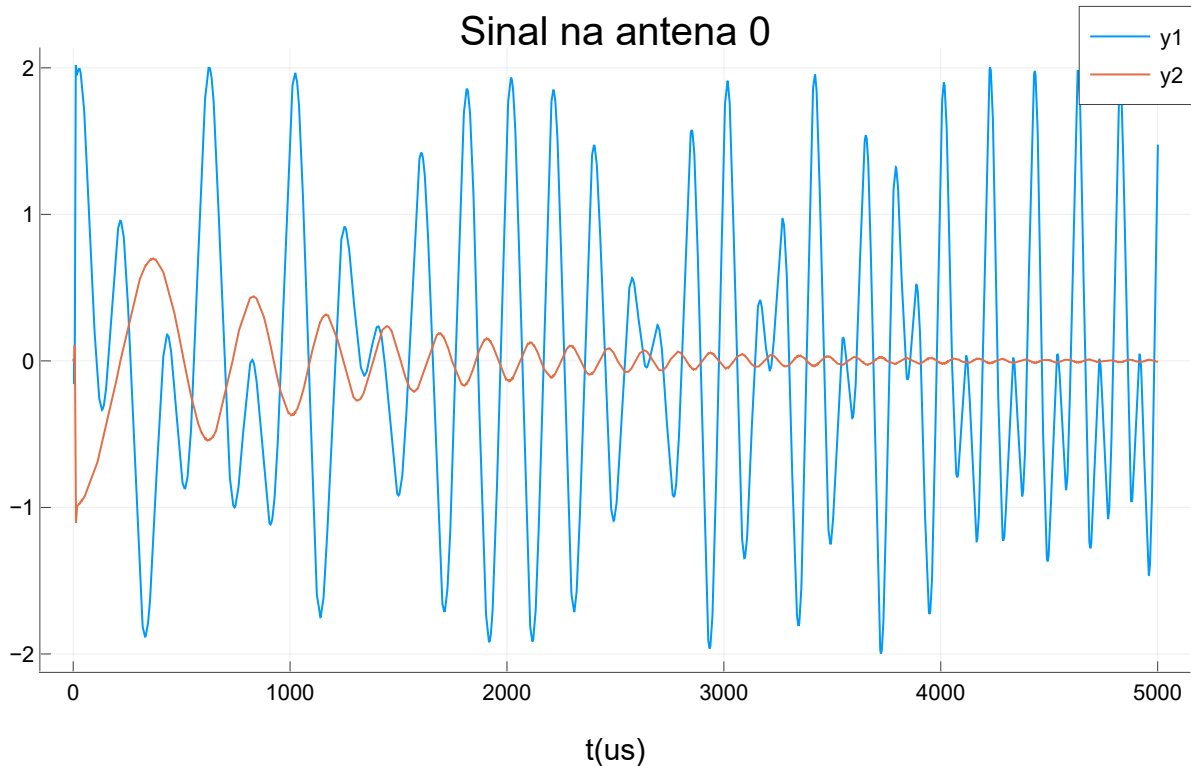


```
• begin
•     responsetype = Lowpass(fi, fs=1e6)
•     designmethod = FIRWindow(hanning(20))
•     PB = digitalfilter(responsetype, designmethod)
•     PBf = PolynomialRatio(PB,[1])
•
•     ω = range(0,π, length=500)
•     H =freqz(PBf, ω)
•     plot(ω/π, abs.(H))
•     plot!(title= "Resposta em frequência do filtro", legend = false, xlabel = "ω/π")
• end
```

## Filtragem do sinal

Nesta etapa o sinal é passado multiplicado por um cosseno e por um seno e depois passa pelo Passa-Baixas para termos a componente real e imaginária do sinal.

É possível observar que há informações na parte imaginária e real do sinal, mas as informações possuem interferência de sinais vindos de outras direções.



```

• begin
•     t = range(0, length(x[:,1])*Ta - Ta, length=length(x[:,1]))
•
•     xa_real = copy(x)
•     xa_imag = copy(x)
•     for k in 1:M
•         xa_real[:,k] = filt(PB,x[:,k].*2cos.(Qi*t))
•     end
•
•     for k in 1:M
•         xa_imag[:,k] = filt(PB,x[:,k].*-2sin.(Qi*t))
•     end
•
•     xa = xa_real+j*xa_imag
•     plot(real.(xa[:,1]))
•     plot!(imag.(xa[:,1]))
•     plot!(title="Sinal na antena 0", xlabel = "t(us)")
• end

```

tau (generic function with 1 method)

```

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

```

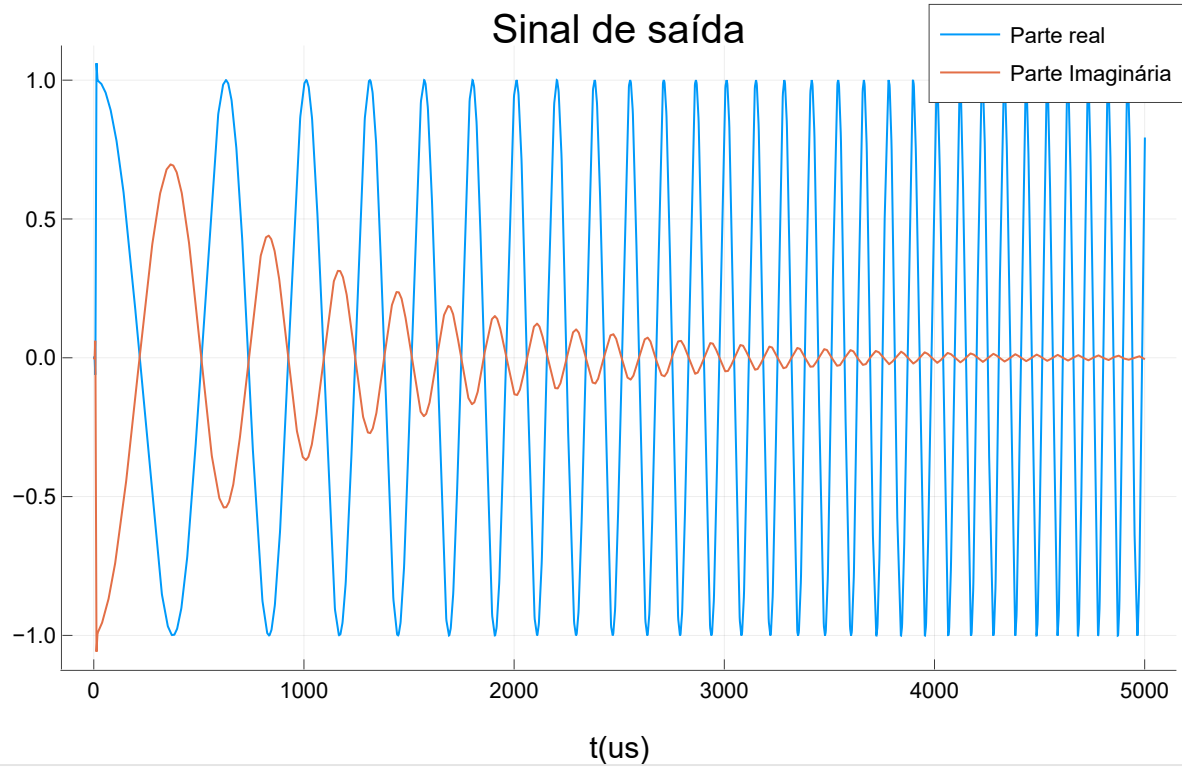
## Soma do sinal das antenas

Nesta etapa ocorre o filtro espacial. O sinal de cada antena é multiplicado por um coeficiente de **w** e todos os sinais serão somados.

Dessa forma os sinais vindo de uma direção diferentes de 30° serão filtrados.

No gráfico abaixo é possível ver o sinal desejado. A parte real com um cosseno de frequência variando; e uma senoide de amplitude variando na parte imaginária.





```

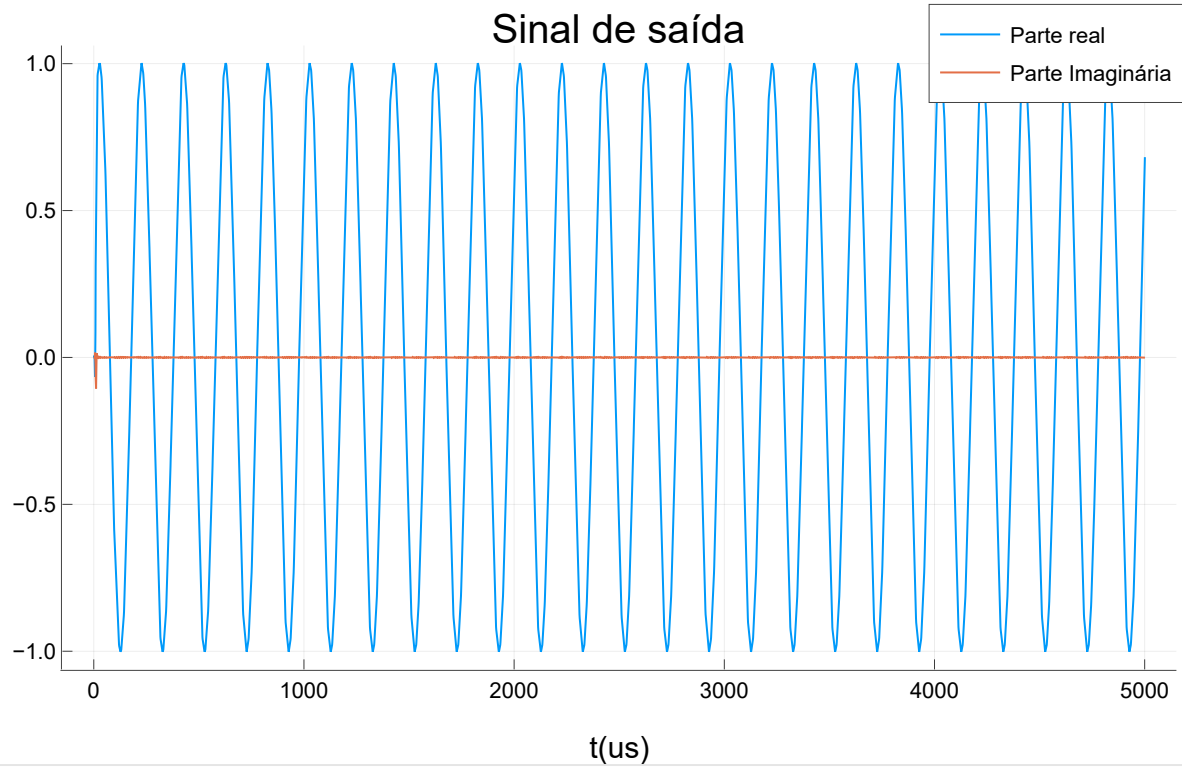
• begin
•     w = ones{Complex, 12}
•     for k in 0:M-1
•         w[k+1] = 1/M*exp(-j*Ω0*tau(k,θ5,d))
•     end
•
•     y = xa*w
•     plot(real.(y), label = "Parte real")
•     plot!(imag.(y), label = "Parte Imaginária")
•     plot!(title= "Sinal de saída", xlabel = "t(us)")
• end
•

```

## Extra

No sinal a cima também é possível filtrar o sinal vindo da direção de  $0^\circ$  apenas mudando o valor dos coeficientes de **w**.

Vemos no gráfico abaixo que o há uma interferência de um cosseno vindo da direção  $0^\circ$



```
• begin
•   θ_extra = 0 #graus
•   w_extra = ones(Complex, 12)
•   for k in 0:M-1
•       w_extra[k+1] = 1/M*exp(-j*Ω0*tau(k,θ_extra,d))
•   end
•   y_extra = xa*w_extra
•   plot(real.(y_extra), label = "Parte real")
•   plot!(imag.(y_extra), label = "Parte Imaginária")
•   plot!(title= "Sinal de saída", xlabel = "t(us)")
• end
```