Universidade de São Paulo

Escola Politécnica

Programa de Pós-Graduação em Engenharia Elétrica

PTC5005 – Processamento Digital de Sinais

Exercício Computacional – Filtro Notch

Aluno: Stéfano Albino Vilela Rezende

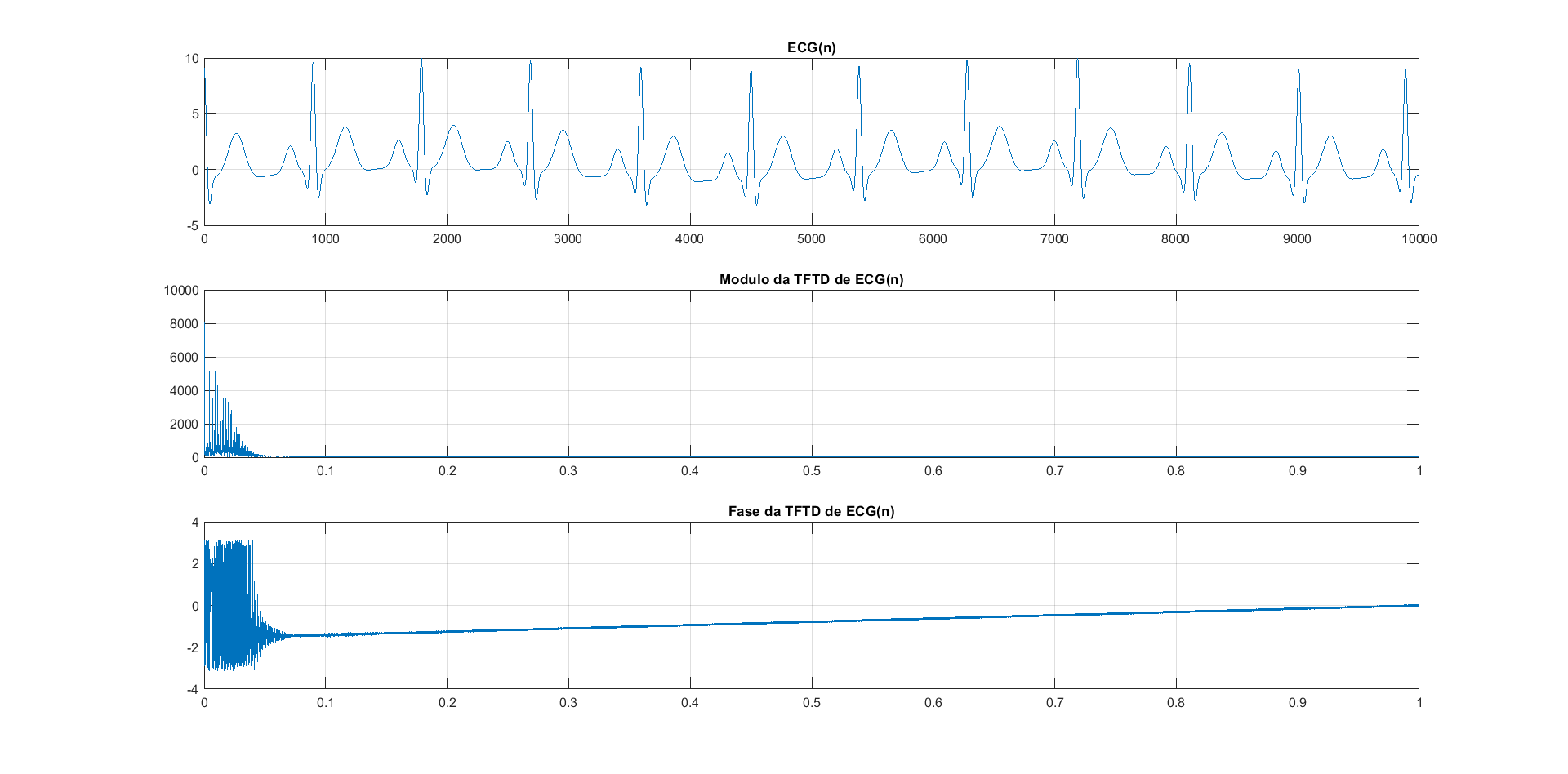
NUSP: “Aluno Ouvinte”

Professor: Maria D. Miranda

São Paulo

2019

**1)****a)**



%PTC 5005 - 2019

%Prof: Maria D. Miranda

%Aluno: Stéfano Albino Vilela Rezende (Ouvinte)

%Exercício Computacional - Filtro Notch

%1.a)

clear

load('dados05102012.mat')

N = length(ecg2);

n = 0 : N-1;

[omega,H] = TFTD(n, ecg2);

figure(1)

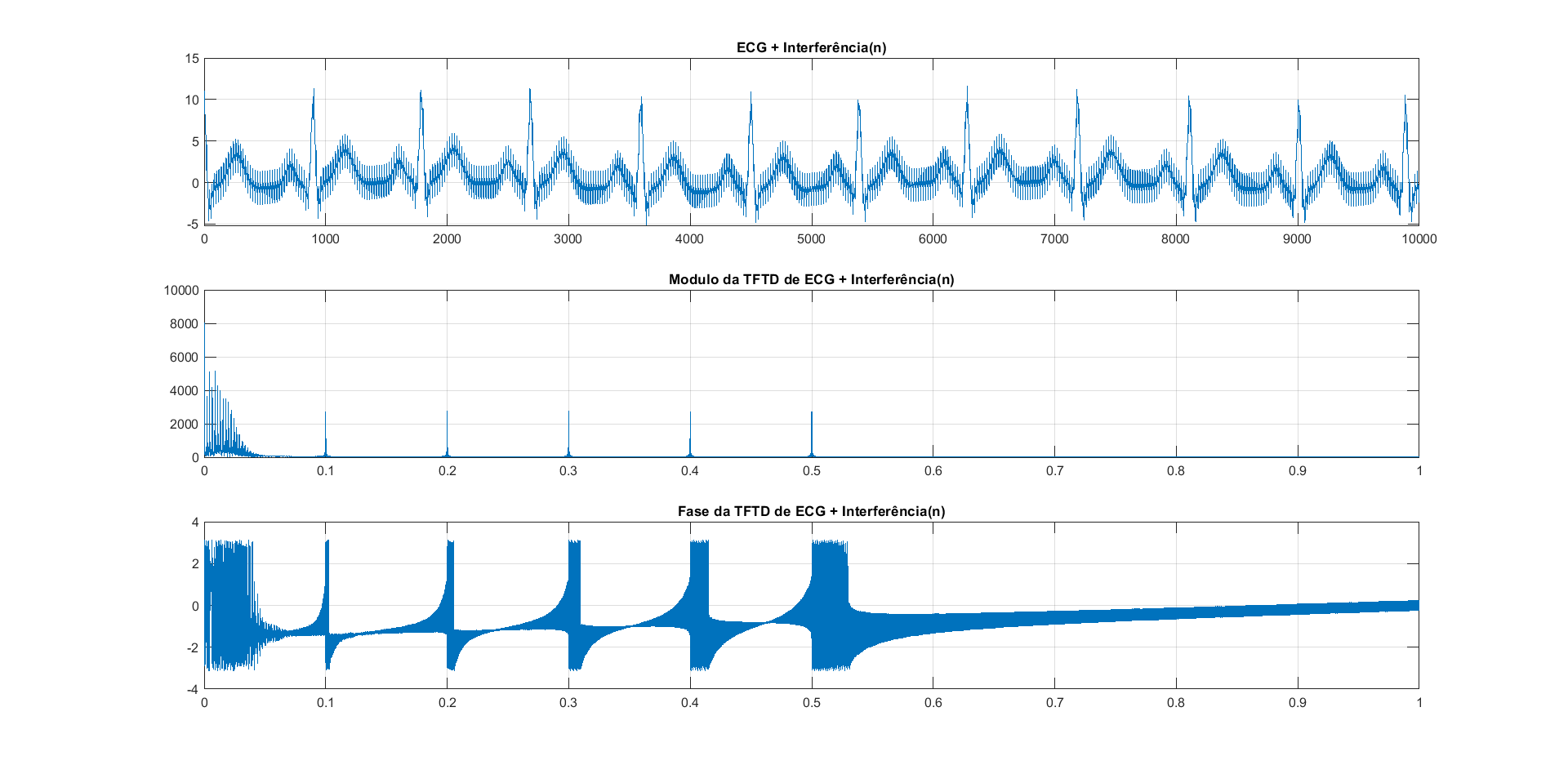
subplot(311); plot(n, ecg2); title('ECG(n)'); grid

subplot(312); plot(omega/pi,abs(H)); grid; title('Modulo da TFTD de ECG(n)')

subplot(313); plot(omega/pi,angle(H)); grid; title('Fase da TFTD de ECG(n)')

*Figura 1 – Eletrocardiograma sem interferência representado no domínio do tempo e no domínio de frequência (Módulo e Fase).*

**b)**



%PTC 5005 - 2019

%Prof: Maria D. Miranda

%Aluno: Stéfano Albino Vilela Rezende (Ouvinte)

%Exercício Computacional - Filtro Notch

%1.b)

clear

load('dados05102012.mat')

N = length(ecg2);

n = 0 : N-1;

[omega,H] = TFTD(n, sinal);

figure(2)

subplot(311); plot(n, sinal); title('ECG + Interferência(n)'); grid

subplot(312); plot(omega/pi,abs(H)); grid; title('Modulo da TFTD de ECG + Interferência(n)')

subplot(313); plot(omega/pi,angle(H)); grid; title('Fase da TFTD de ECG + Interferência(n)')

*Figura 2 – Eletrocardiograma com interferência representado no domínio do tempo e no domínio de frequência (Módulo e Fase).*

**2)**

%PTC 5005 - 2019

%Prof: Maria D. Miranda

%Aluno: Stéfano Albino Vilela Rezende (Ouvinte)

%Exercício Computacional - Filtro Notch

load('dados05102012.mat')

%2.a)

fa=1200; % frequencia de amostragem

fint=60; % frequencia fundamental a ser eliminada

K=5; % fundamental mais K-1 de harm^onicas

Nt=length(sinal);

n=1:Nt-1;

omega0=2\*pi\*fint/fa;

z1=exp(1j\*omega0\*[1:K]); % zeros da fun¸c~ao de transfer^encia

% Precisamos ter zeros tamb´em nos complexos conjugados para que os coef. do

% filtro sejam reais

z2=conj(z1);

ind=find(real(z2)==-1);

z2(ind)=[]; % para evitar zero duplo em -1

zz=[z1 z2]; % o filtro n~ao elimina DC pois n~ao tem zeros em z=1

bfir=poly(zz); % numerador do filtro FIR

figure(1)

zplane(bfir,1); title('Zeros do FIR')

%b)

fator=input('Entre com o valor do fator do = ');

bfirfator=fator\*bfir;

[Hfir,ffir]=freqz(bfirfator,1,4096,fa);

figure(2)

subplot(211); plot(ffir,20\*log10(abs(Hfir))); ylabel('Módulo (dB)'); grid on; ylim([-30 100]);

title('Resposta em frequência do filtro FIR')

subplot(212); plot(ffir,angle(Hfir));

ylabel('Fase (rad)'); xlabel('f (Hz)'); grid on

%c)

% Filtro FIR

ecg\_fir=filter(bfirfator,1,sinal);

t=(0:length(sinal)-1)/fa; % tempo em segundos

figure(3)

t1=length(bfir);

t2=length(sinal);

subplot(211); plot(t(t1:t2), sinal(t1:t2)); title('ECG com interferência'); grid

subplot(212); plot(t(t1:t2), ecg\_fir(t1:t2));

xlabel('tempo (s)'); title('Saída do filtro FIR')

hold; plot(t, ecg2(1:end),'r'); hold off; grid

%d)

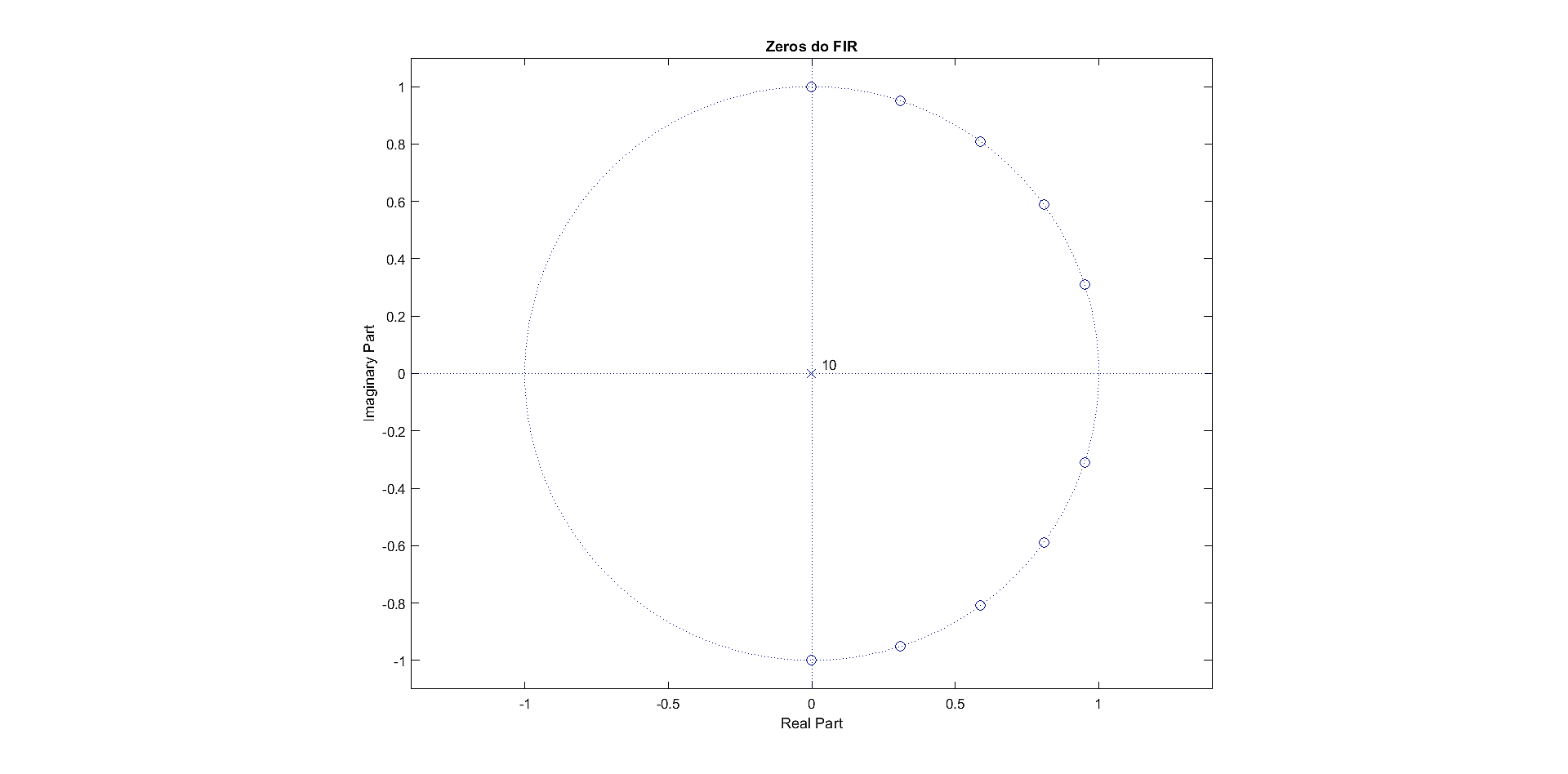
[omega, Hsinal] = TFTD (n,ecg\_fir)

figure(4)

subplot(211); plot(omega/pi,abs(Hsinal)); grid; title('Modulo da TFTD de ECG(n) Filtrado')

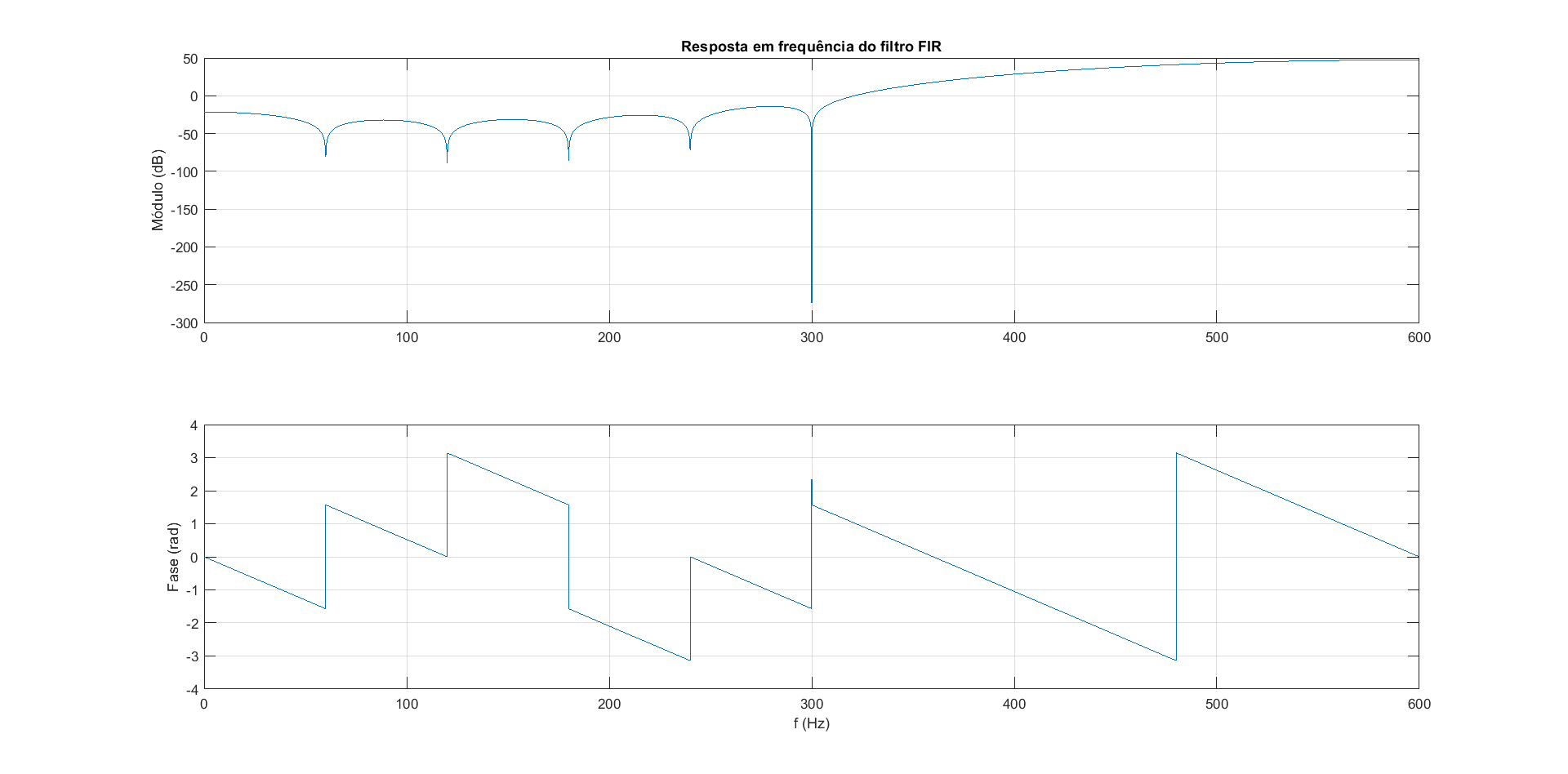
subplot(212); plot(omega/pi,angle(Hsinal)); grid; title('Fase da TFTD de ECG(n) Filtrado')

**a)**

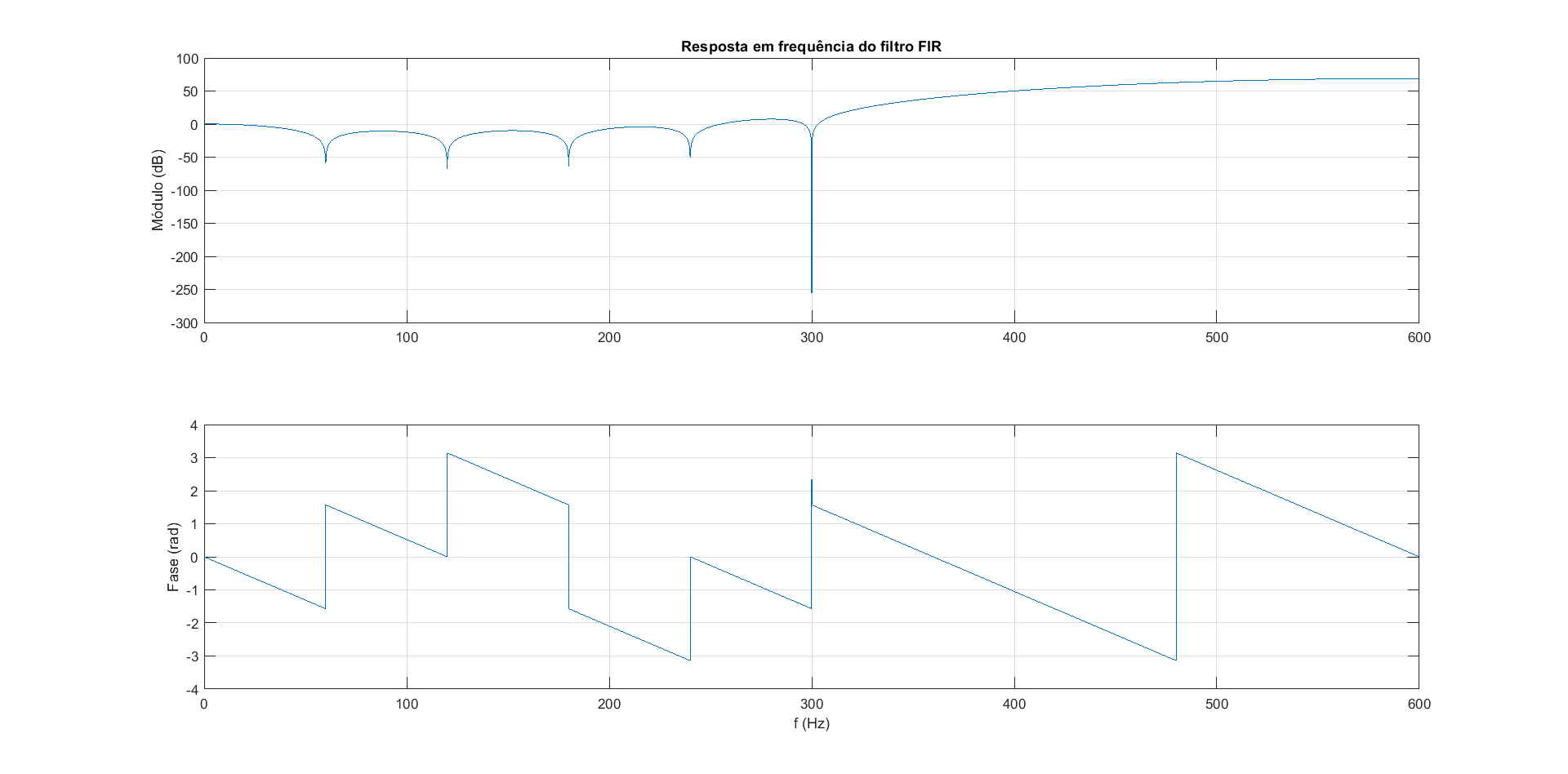
****

*Figura 3 – Diagrama de Polos e Zeros do filtro FIR projetado.*

**b)**

****

*Figura 4 – Resposta em Frequência do filtro FIR projetado para fator = 1.*

**

*Figura 5 – Resposta em Frequência do filtro FIR projetado para fator = 12.*

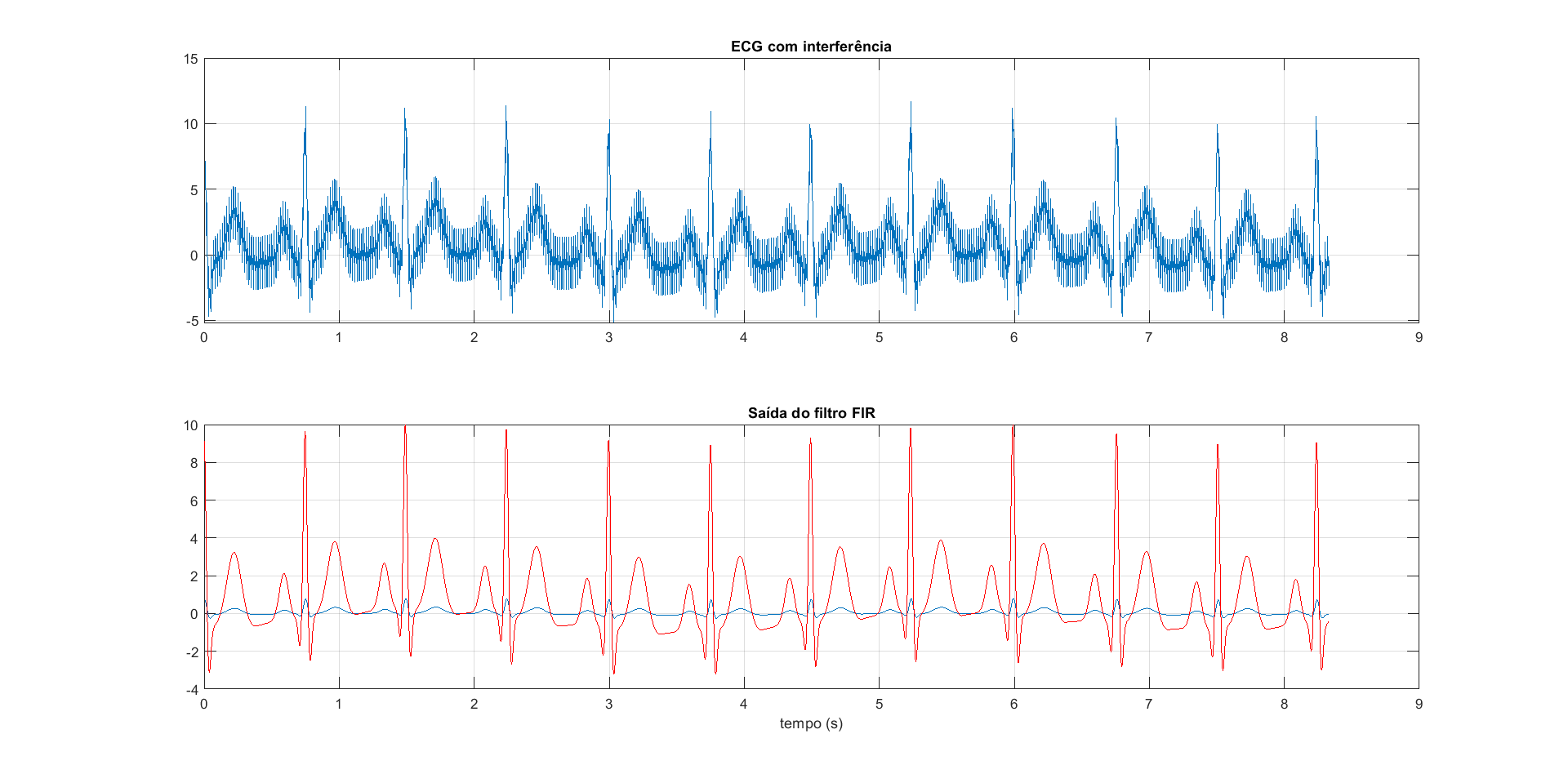
A *Tabela 1* apresenta os valores de atenuação na frequência de interferência e sua harmônicas, de acordo com o fator escolhido para o filtro.

*Tabela 1 – Resposta em Frequência do filtro FIR projetado para fator = 12.*

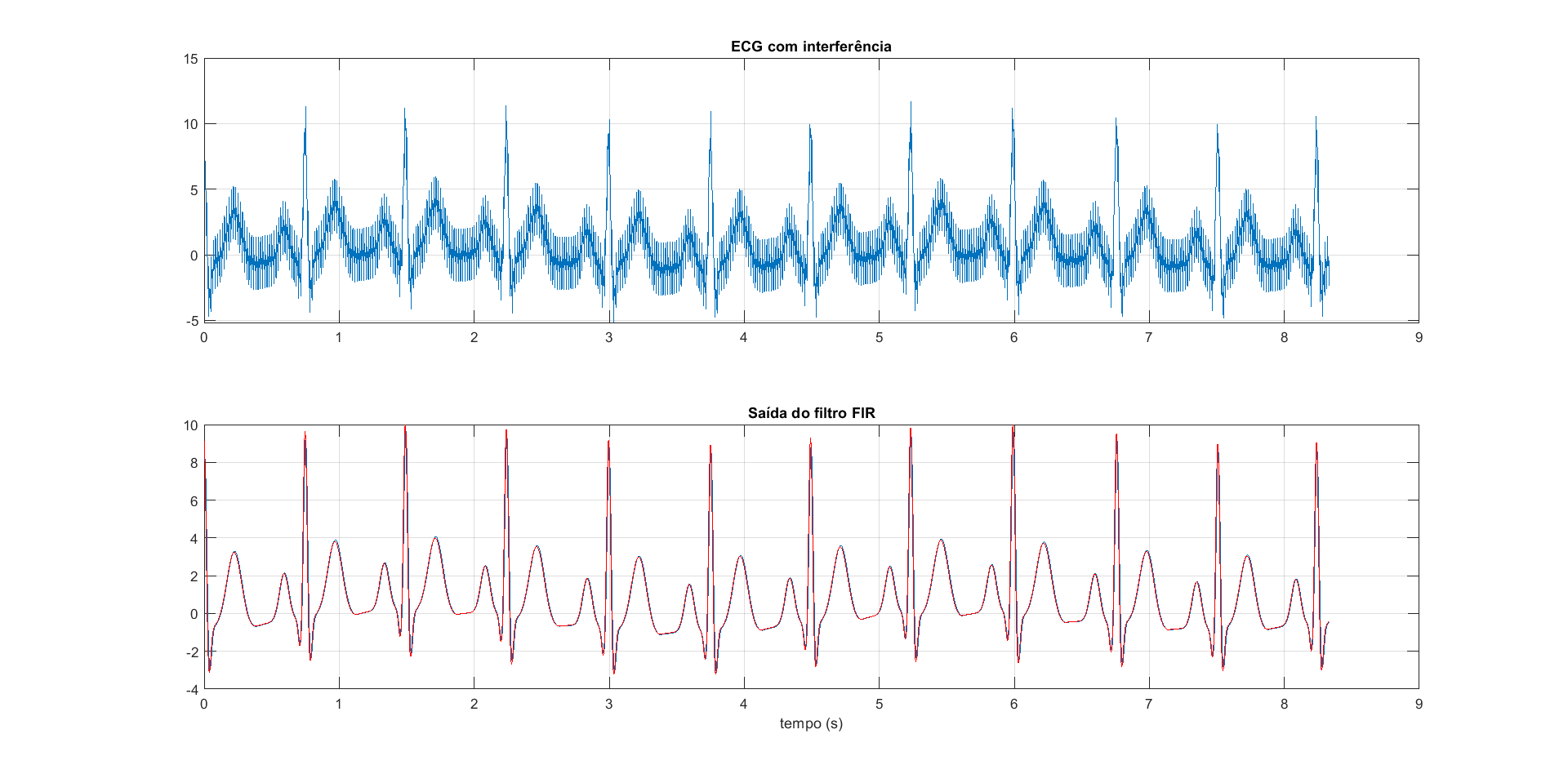
|  |  |  |
| --- | --- | --- |
| **Frequência (Hz)** | **Atenuação para fator=1 (dB)** | **Atenuação para fator=12 (dB)** |
| 60 | 76,86 | 55,27 |
| 120 | 88,71 | 67,12 |
| 180 | 73,59 | 52,01 |
| 240 | 71,77 | 50,18 |
| 300 | 273,93 | 255,87 |

Algo a ser ressalta é a resposta nas frequências de interesse para o caso em que o *fator = 1* temos uma atenuação de aproximadamente 21 dB, já no caso em que o *fator = 12* temos a resposta nessas frequências próximo a 0 dB.

**c)**

****

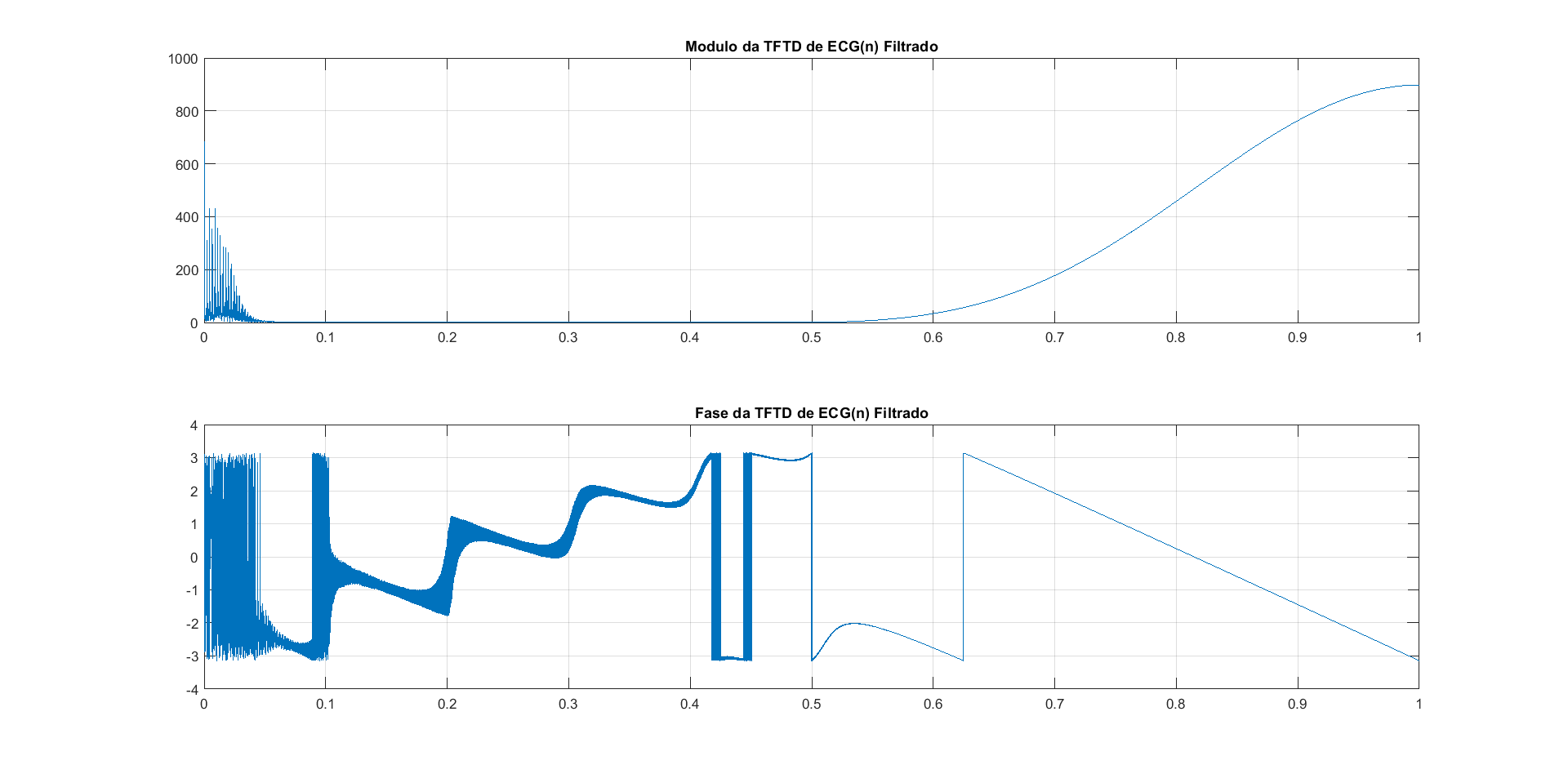
*Figura 6 – Resposta do sinal ECG em função do tempo com interferência, sem interferência (vermelho) e a saído do filtro FIR com fator=1.*

**

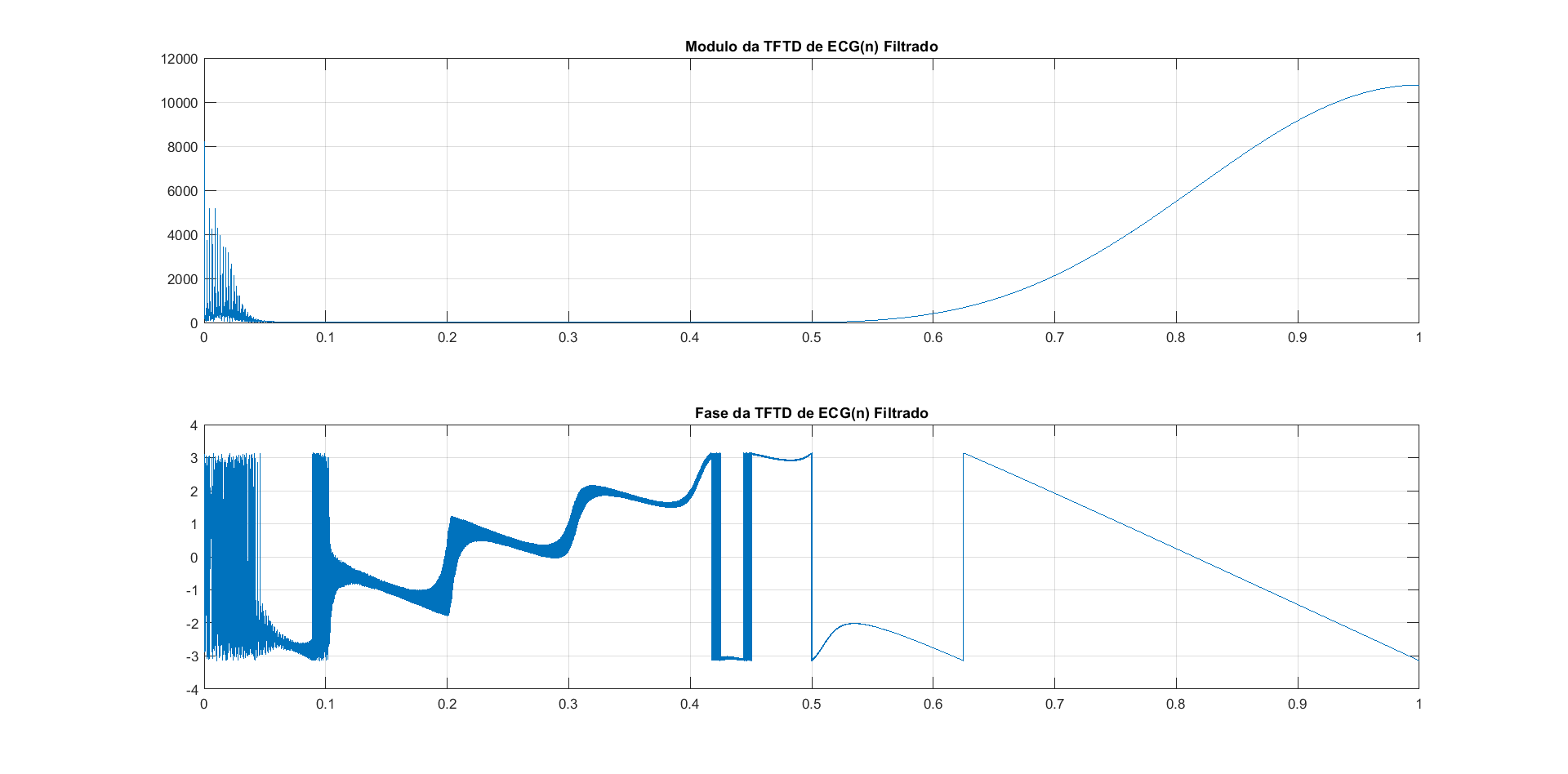
*Figura 7– Resposta do sinal ECG em função do tempo com interferência, sem interferência (vermelho) e a saído do filtro FIR com fator=12.*

O filtro eliminou as interferências conforme desejado. Mas utilizando o *fator =1* temos uma atenuação no sinal do ECG devido a resposta do filtro nas frequências desejadas. Já com o *fator = 12* temos um resultado satisfatório.

**d)**

****

*Figura 8 – Resposta em frequência do sinal ECG na saído do filtro FIR com fator=1.*

**

*Figura 9 – Resposta em frequência do sinal ECG na saído do* *filtro FIR com fator=12.*

A principal diferença percebida é o aparecimento de componentes em alta frequência, significativamente a partir de 360 Hz.

**3)**

%PTC 5005 - 2019

%Prof: Maria D. Miranda

%Aluno: Stéfano Albino Vilela Rezende (Ouvinte)

%Exercício Computacional - Filtro Notch

load('dados05102012.mat')

%3.a)

fa=1200; % frequencia de amostragem

fint=60; % frequencia fundamental a ser eliminada

K=5; % fundamental mais K-1 de harm^onicas

Nt=length(sinal);

n=1:Nt-1;

omega0=2\*pi\*fint/fa;

z1=exp(1j\*omega0\*[1:K]); % zeros da fun¸c~ao de transfer^encia

% Precisamos ter zeros também nos complexos conjugados para que os coef. do

% filtro sejam reais

z2=conj(z1);

ind=find(real(z2)==-1);

z2(ind)=[]; % para evitar zero duplo em -1

zz=[z1 z2]; % o filtro não elimina DC pois não tem zeros em z=1

r=0.95; % modulo dos polos do filtro notch

polos=r\*zz;

b64=real(poly(zz)); % numerador do filtro IIR

a64=real(poly(polos));% denominador do filtro IIR

figure(1)

zplane(b64,a64); title('Polos e Zeros do IIR')

%b)

figure(2)

[H64,f64]=freqz(b64,a64,4096,fa);

subplot(211); plot(f64,20\*log10(abs(H64))); ylabel('Módulo (dB)'); grid on

subplot(212); plot(f64,angle(H64)); ylabel('Fase (rad)'); xlabel('f (Hz)'); grid on

%c)

% Filtro IIR

ecg\_iir=filter(b64,a64,sinal);

t=(0:length(sinal)-1)/fa; % tempo em segundos

figure(3)

subplot(211); plot(t, sinal), title('ECG com interfência'); grid on

subplot(212); plot(t, ecg\_iir), xlabel('tempo (s)'); title('Saída do filtro IIR')

hold; plot(t, ecg2(1:end),'r'); hold off; grid on

%d)

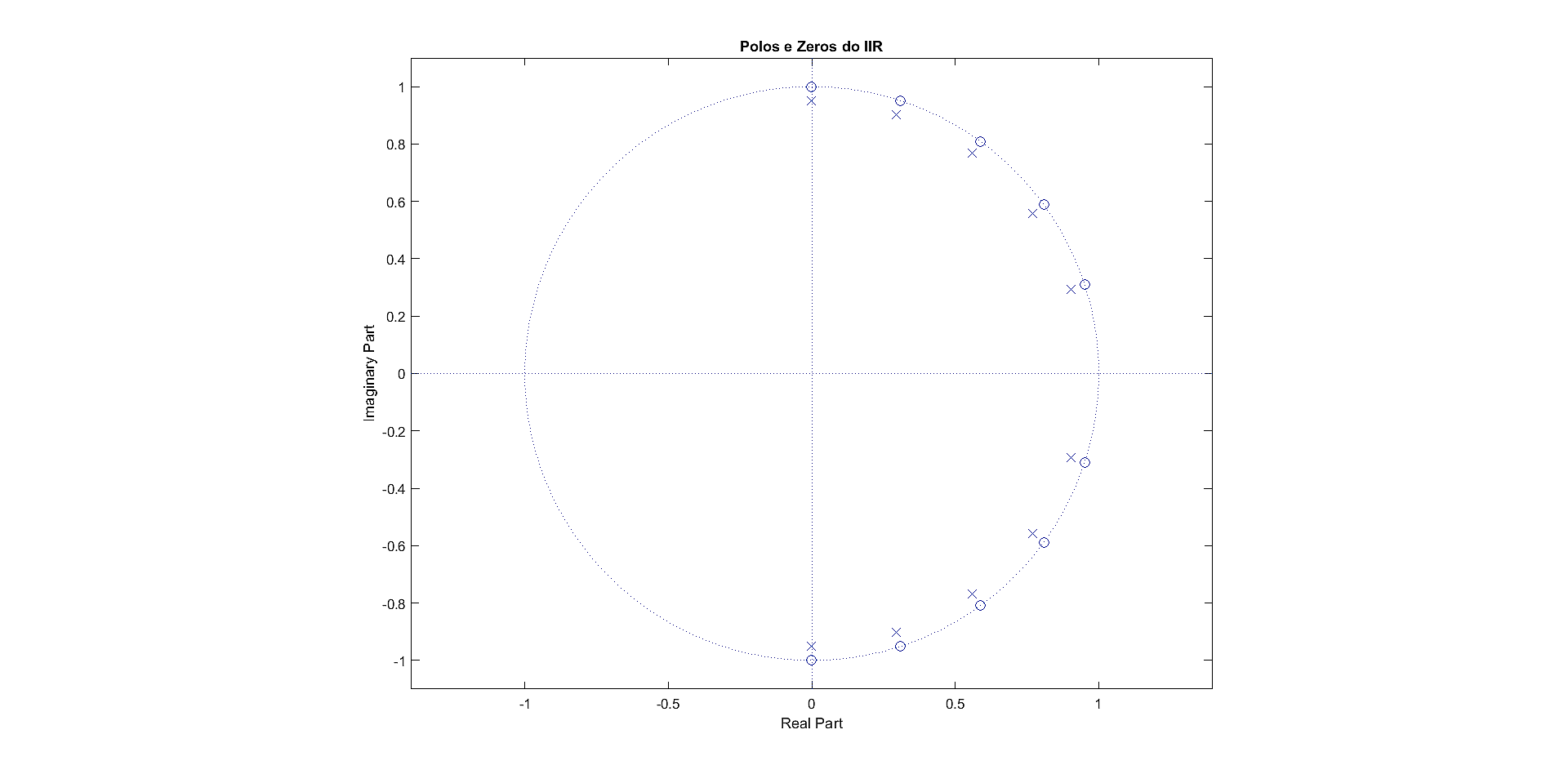
[omega, Hsinal] = TFTD (n,ecg\_iir)

figure(4)

subplot(211); plot(omega/pi,abs(Hsinal)); grid; title('Modulo da TFTD de ECG(n) Filtrado')

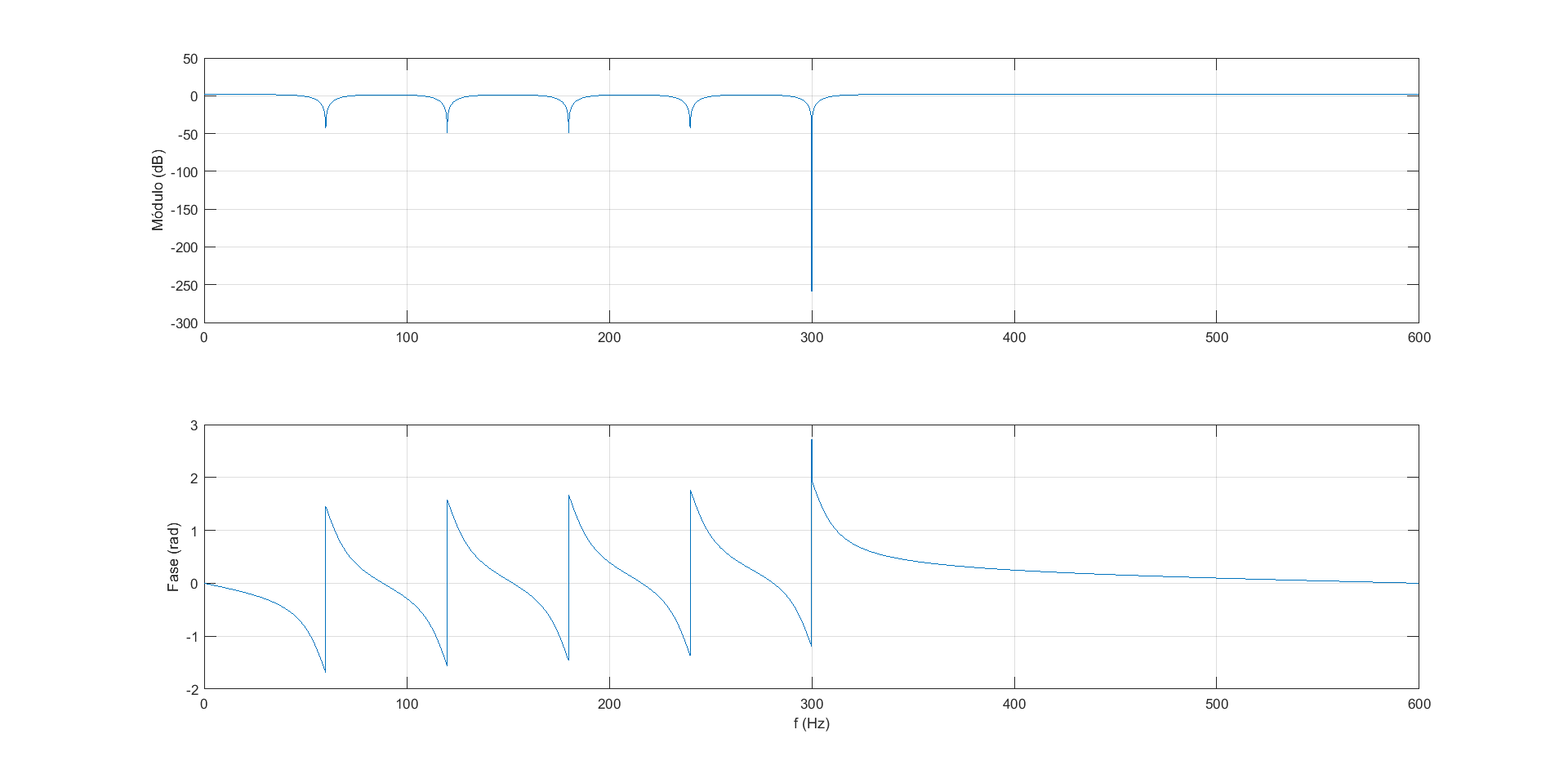
subplot(212); plot(omega/pi,angle(Hsinal)); grid; title('Fase da TFTD de ECG(n) Filtrado')

**a)**

****

*Figura 10 – Diagrama de Polos e Zeros do filtro IIR projetado.*

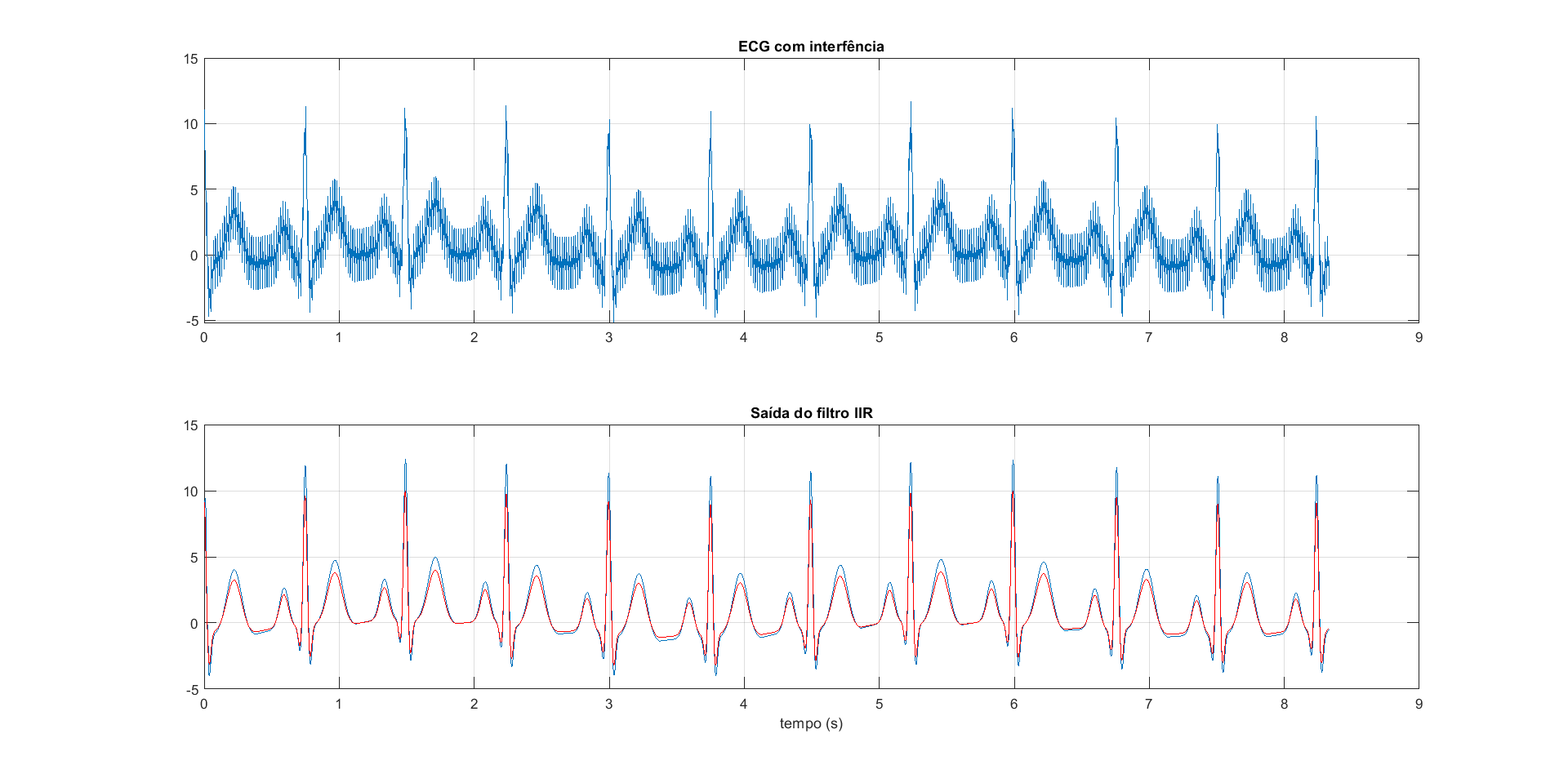
**b)**

****

*Figura 11 – Resposta em Frequência do filtro IIR projetado.*

A principal vantagem do filtro IIR sobre o FIR é a planicidade da resposta nas frequências de interesse e um decaimento abrupto nas frequências de corte, isso garante uma resposta uniforme.

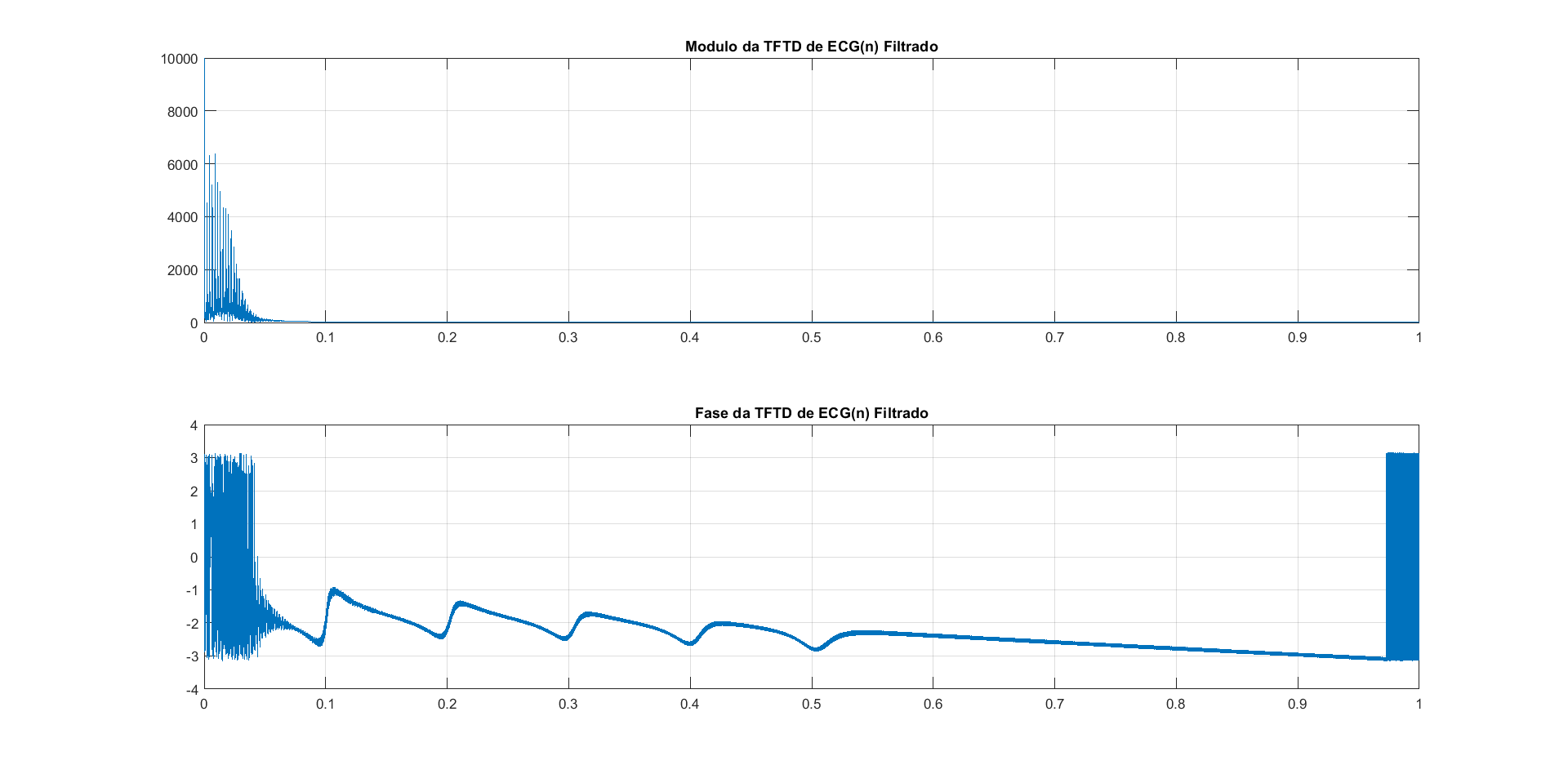
**c)**

****

*Figura 12– Resposta do sinal ECG em função do tempo com interferência, sem interferência (vermelho) e a saído do filtro IIR.*

Devido a planicidade da resposta em frequência do filtro, vemos o sinal reconstituído.

**d)**

****

*Figura 13 – Resposta em frequência do sinal ECG na saído do filtro IIR.*

É percebido que as frequências de interesse houveram um ganho em relação ao sinal filtrado pelo FIR. E o comportamento da fase é mais estável.