

*Instituto Politécnico de Lisboa*

*Instituto Superior de Engenharia de Lisboa*

Relatório

1º Trabalho de COMUNICAÇÕES

***Docente: Artur Ferreira***

***Turma: LI31N***

Elaborado por:

Nº 38858 – Miguel Lourenço

Nº 38866 – Manuel Dias

Nº32734 – André Fonseca

Índice

Conteúdo

[Parte 1 3](#_Toc529826564)

[a) 3](#_Toc529826565)

[b) 3](#_Toc529826566)

[c) 3](#_Toc529826567)

[d) 3](#_Toc529826568)

[e) 4](#_Toc529826569)

[f) 5](#_Toc529826570)

[g) 7](#_Toc529826571)

[Parte 2 7](#_Toc529826572)

[a. 7](#_Toc529826573)

[b. 7](#_Toc529826574)

[c) 8](#_Toc529826575)

[Parte 3 9](#_Toc529826576)

[a. 9](#_Toc529826577)

[b. 9](#_Toc529826578)

[C. 11](#_Toc529826579)

[d) 14](#_Toc529826580)

[Parte 4 15](#_Toc529826581)

[a) 15](#_Toc529826582)

[b) 15](#_Toc529826583)

# Parte 1

# a)

Ao executar as instruções pedidas pelo guia, ocorreram 2 erros.

O primeiro erro teve origem ao tentar aceder a um índice do array que não era nem um valor inteiro positivo nem um valor lógico. Neste caso foi tentado aceder ao índice 0 do array.

O segundo erro foi parecido ao anterior, no entanto, nesta instrução foi tentado aceder ao índice x do array. Apesar de ser possível aceder aos índices de arrays através de uma variável, não é possível se esta variável for também um array.

# b)

O primeiro operador, .^2, eleva ao quadrado cada elemento da matriz e o segundo operador, .\*, multiplica cada elemento de uma matriz por cada elemento de outra matriz, se as matrizes tiverem o mesmo tamanho.

# c)

Uma maneira para calcular a energia de um sinal é somar o quadrado de cada ponto do sinal. O resultado obtido foi

# d)

A instrução B(1,:) devolve todos os valores da linha 1. O resultado obtido foi [1 2 3].  
A instrução B(:,2) devolve todos os valores da coluna 2. O resultado obtido foi [2;1;0;0].  
A instrução B(2:3) devolve o valor da linha 2 coluna 3. O resultado obtido foi 2.  
A instrução sum(B) devolve o resultado da soma de todos os valores de cada coluna. O resultado obtido foi [-6 3 -1].  
A instrução sum(sum(B)) devolve o resultado da soma de todos os valores da matriz. O resultado obtido foi -4.  
A instrução [NR, NC]=size(B) devolve o número de linhas no primeiro argumento e o número de colunas no segundo. O resultado obtido foi NR=4 e NC=3.  
A instrução length(B) devolve o número de linhas da matriz. O resultado obtido foi 4.  
A instrução M=A copia todos os valores da matriz A para a matriz M. O resultado obtido foi M=[1 2 3;0 1 2;-3 0 -5].  
A instrução M(4,:)=y copia todos os valores de y para a linha 4 da matriz M. O resultado obtido foi M=[1 2 3;0 1 2;-3 0 -5; -4 0 -1].  
A instrução M(:,3)=[] elimina a terceira coluna da matriz M. O resultado obtido foi M=[1 2;0 1;-3 0;-4 0].

# e)

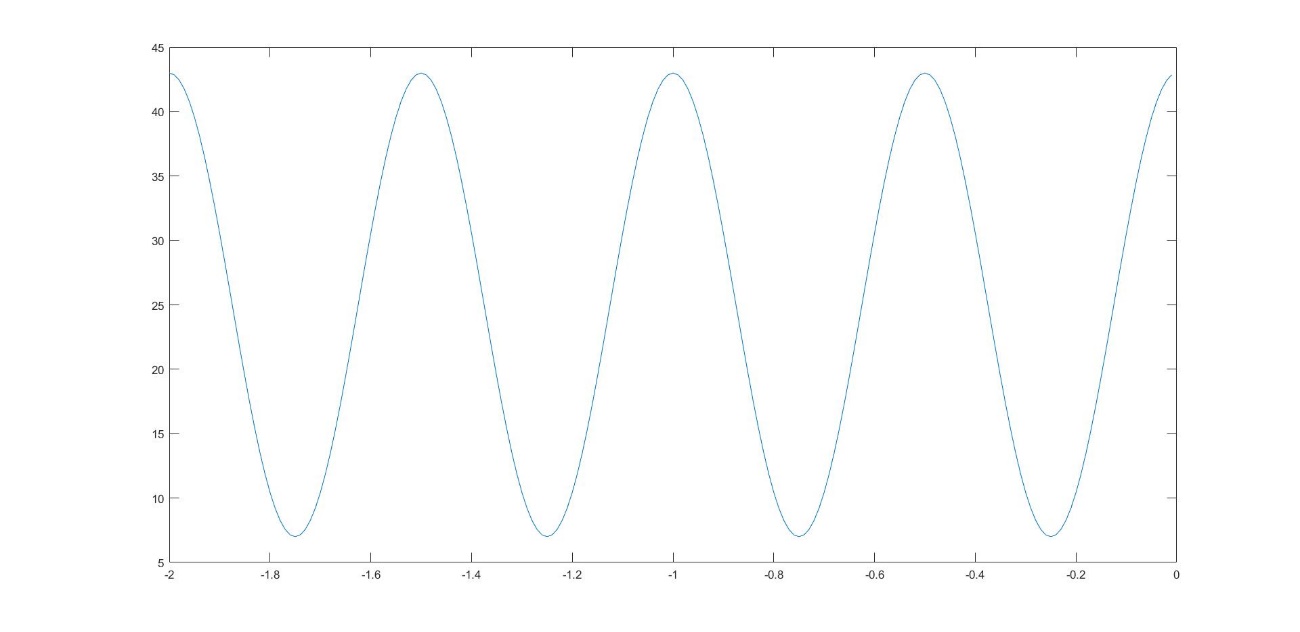


Figura Amostra de 4 periodos fundamentais do sinal w

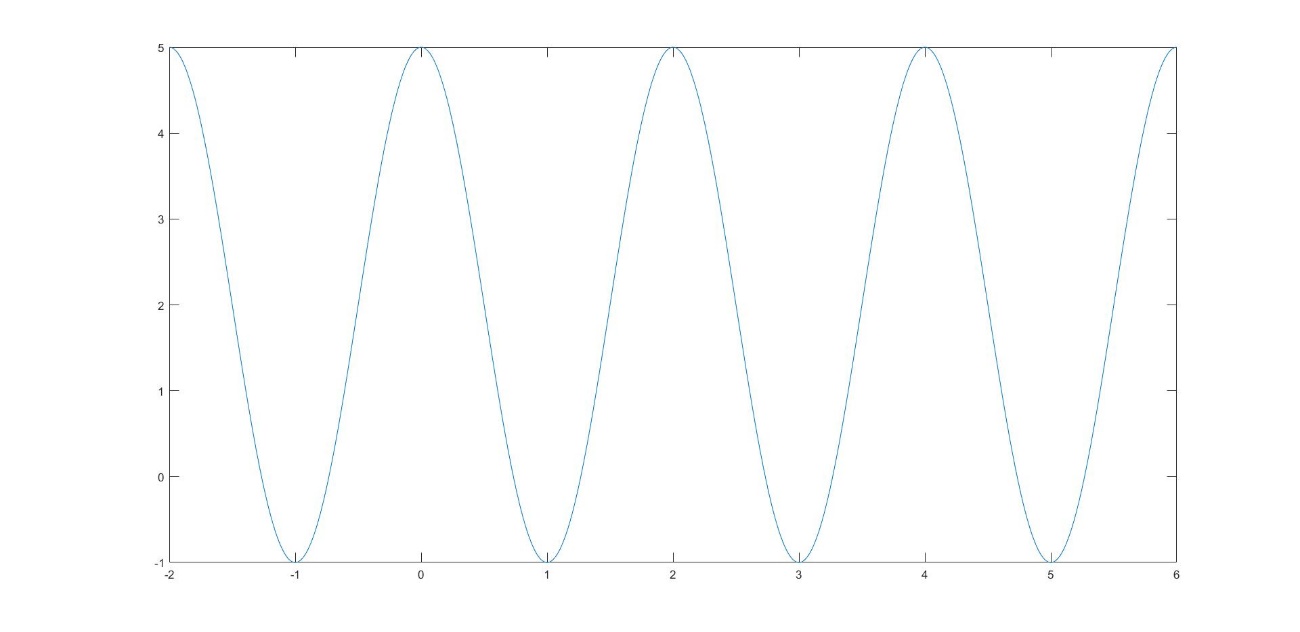


Figura Amostra de 4 periodos fundamentais do sinal z

# f)

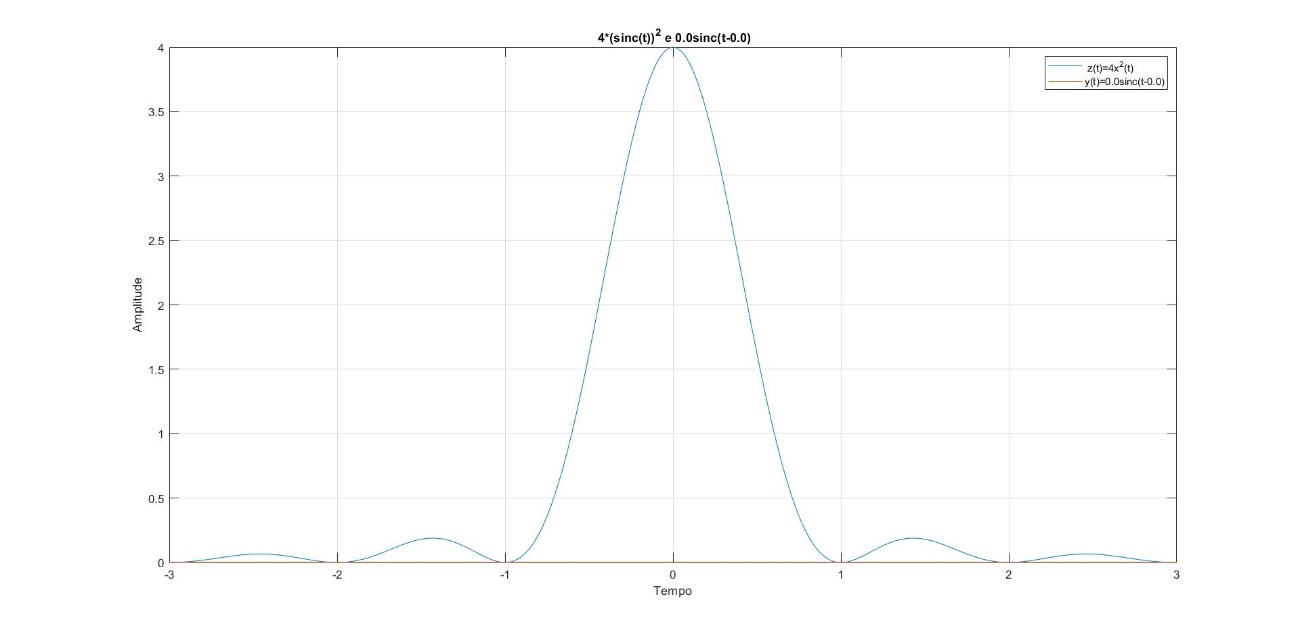


Figura Sinais z em azul e w a vermelho com a=0 e b=0

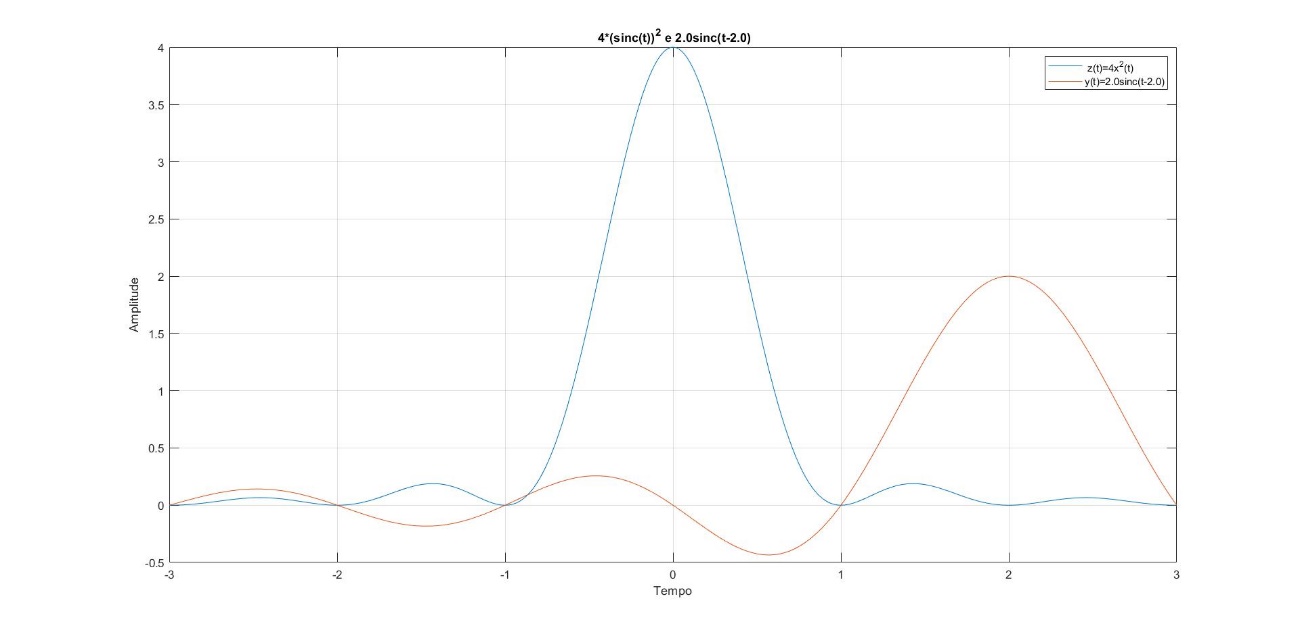


Figura Sinais z em azul e w a vermelho com a=2 e b=2

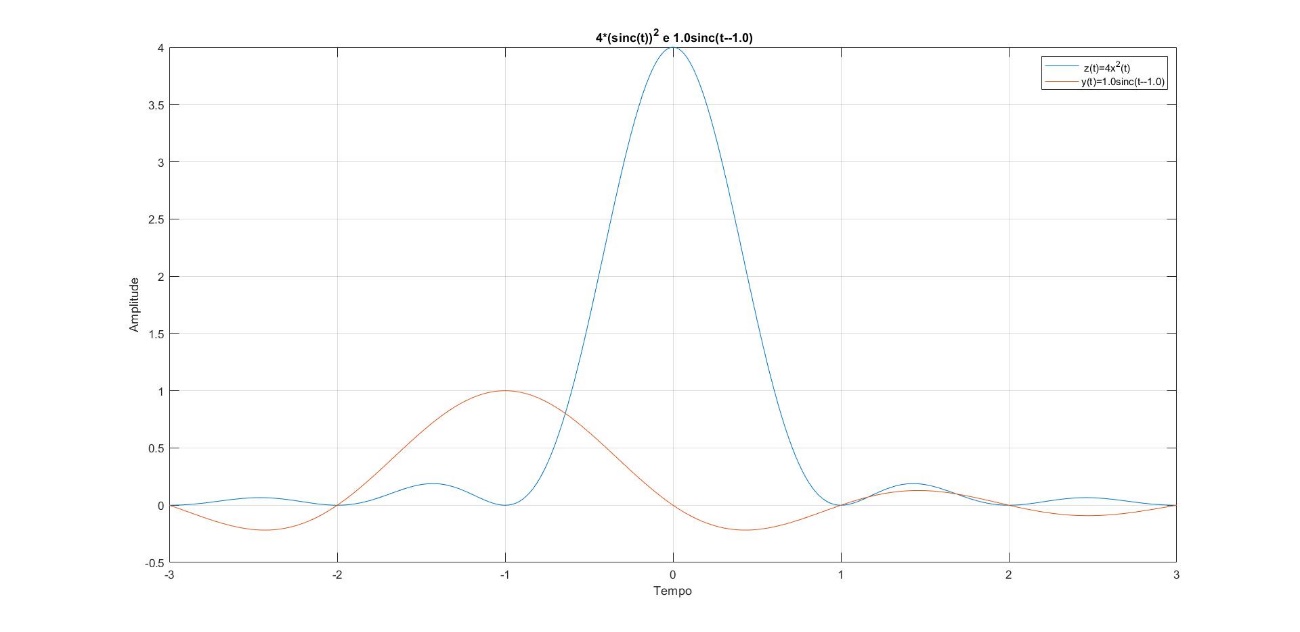


Figura Sinais z em azul e w a vermelho com a=1 e b=-1

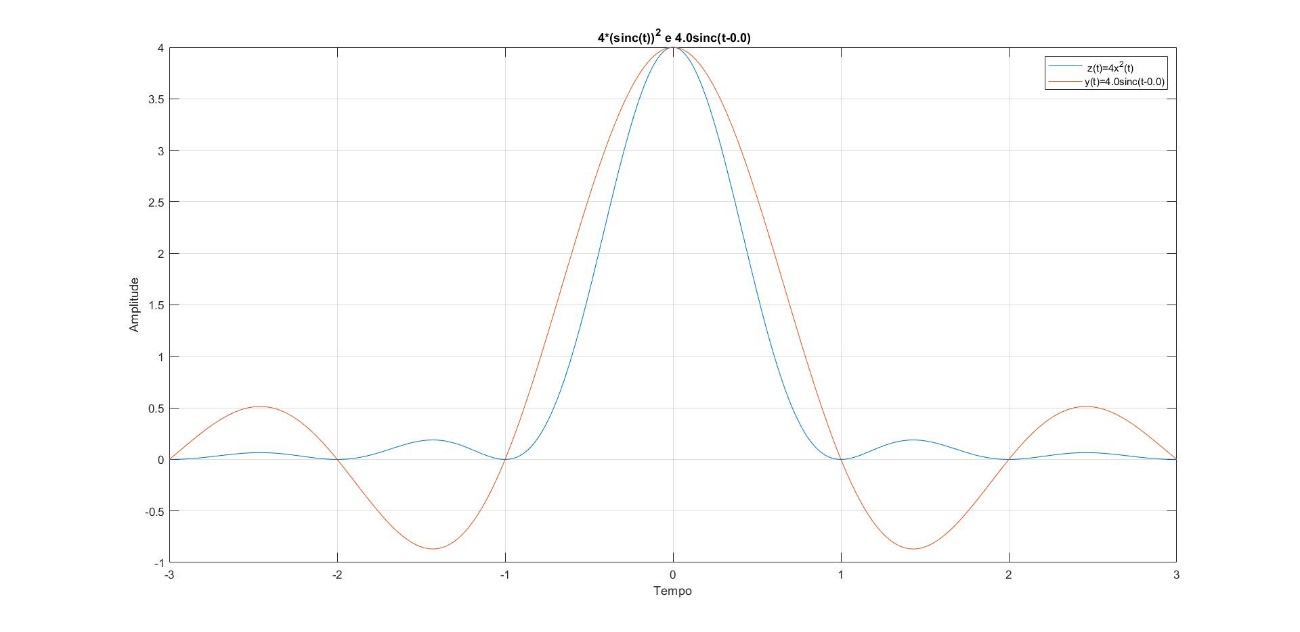


Figura Sinais z em azul e w a vermelho com a=4 e b=0

# g)

Para desenvolver a função *record\_audio\_to\_file*, esta recebe como parâmetros de entrada a frequência fundamental, a duração da gravação e o nome do ficheiro onde se vai guardar a gravação nos formatos .wav e .flac.

# Parte 2

# a.

b. m1(t)=m2(t)=1

a(t)= A\*cos() quanto à sua periodicidade é periódico, quanto à simetria é par , visto estarmos a falar de um co-seno.

b(t)= quanto à sua periodicidade é periódico, quanto à simetria é ímpar, visto estarmos a falar de um seno.

1. Com A e B =1

s(t) é periódico e sem simetria.

1. Com A = 0, B = 1

s(t) é periódico com simetria ímpar

1. Com A = 1, B= 0

s(t) é periódico com simetria par

1. Com A = 2 , B=4

S(t) é periódico sem simetria

# c)

Para o desenvolvimento da função *signal\_IQ*, esta recebe como parâmetros de entrada dois sinais e dois valores de amplificação. Como parâmetros de saída, foi decidido que a função devolvesse o sinal s e parte par e a parte ímpar dos sinais s, a e b, como é pedido no enunciado, e acrescentou-se a devolução do valor da frequência e a matriz de tempo caso seja necessário representar os sinais graficamente em função do tempo.

No entanto para realizar esta função foi necessário recorrer a uma função auxiliar, *createSignal*, que cria um sinal a partir de 3 parâmetros de entrada, um valor 0 ou 1, que define se o sinal é co-seno ou seno, um valor de frequência e uma matriz de tempo.

# Parte 3

a. A função *signal\_mixer* tem como parâmetros de entrada uma matriz de valores de amplitude, uma matriz de sinais, o número de sinais que são para somar e uma matriz de tempo para representar graficamente o sinal final ao longo do tempo. A função simplesmente junta todos os sinais que vêm como parâmetro e junta num só sinal, multiplicando pelo valor da amplitude correspondente.

# x1.jpgb.

Figura x1(t) = 5 cos(2π100t) + 10 cos(2π300t)

Amax= 15 V

Amin=-15 V

Amed= 0 V

Frequência fundamental=mdc(100,300)= 100 Hz

LB= 300-200 = 100 Hz

Pelos gráficos resultantes da utilização da função signal\_mixer, pode-se constatar que a amplitude máxima do sinal x1(t) é 15 V, que a amplitude mínima é -15 V e ainda que a amplitude média do sinal é 0 V. Pelo espetro de amplitude do sinal, consegue-se extrair a frequência máxima e a frequência mínima, conseguindo desta forma calcular a largura de banda existente. Pelo espectrograma do sinal consegue se ver que a gama de frequências concentra-se á volta dos 300 Hz e dos 100 Hz, resultantes da soma dos co-senos do sinal.

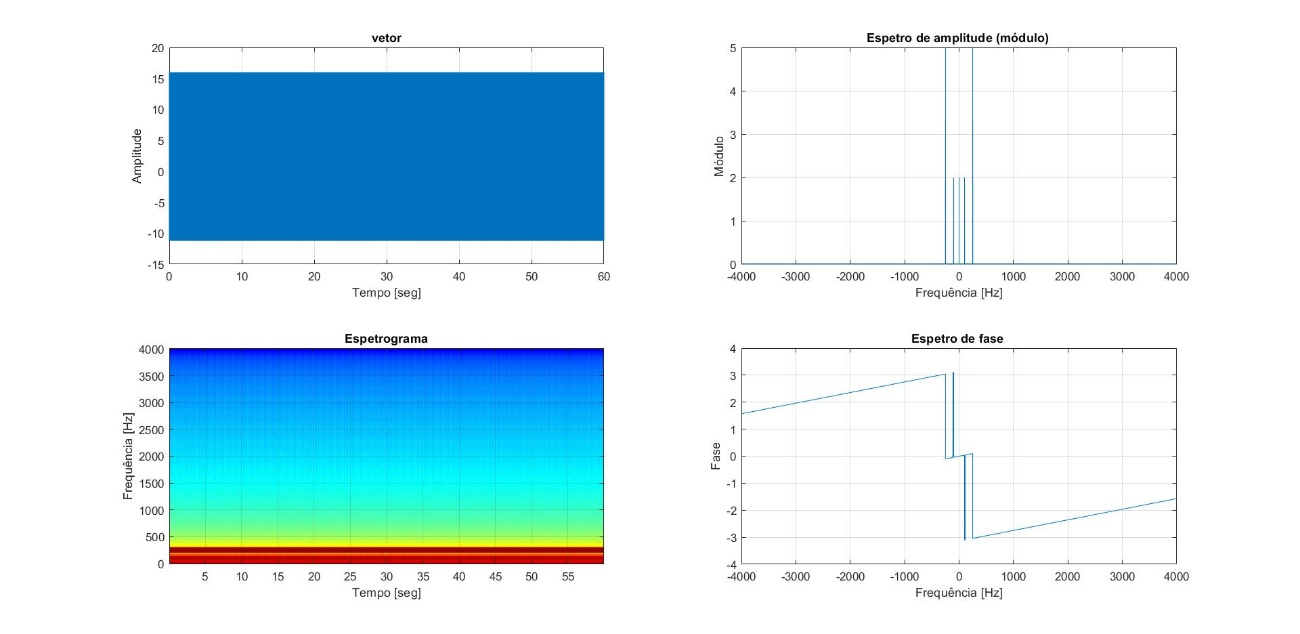


Figura x2(t) = 2 + 4 cos(2π100t) + 10 cos(2π250t)

Amax= 16 V

Amin=-12 V

Amed= 2 V

Frequência fundamental=mdc(100,250)= 50 Hz

LB= 250-100 = 150 Hz

Pelos gráficos resultantes da utilização da função signal\_mixer, pode-se constatar que a amplitude máxima do sinal x2(t) é 16 V, que a amplitude mínima é -12 V e ainda que a amplitude média do sinal é 2 V resultante da componente DC do sinal.

C. Para realizar este exercício foi necessário criar uma função, *signal\_mixer\_sinc*, que crie os sinais e depois chame a função *signal\_mixer* para juntar os sinais criados e adicione uma amplitude .

x3(t) = A0 + dsinc(kd) cos(2πkfot)

(i). A0= 5 V , N=5 , d= 0.5 , fo= 500 Hz

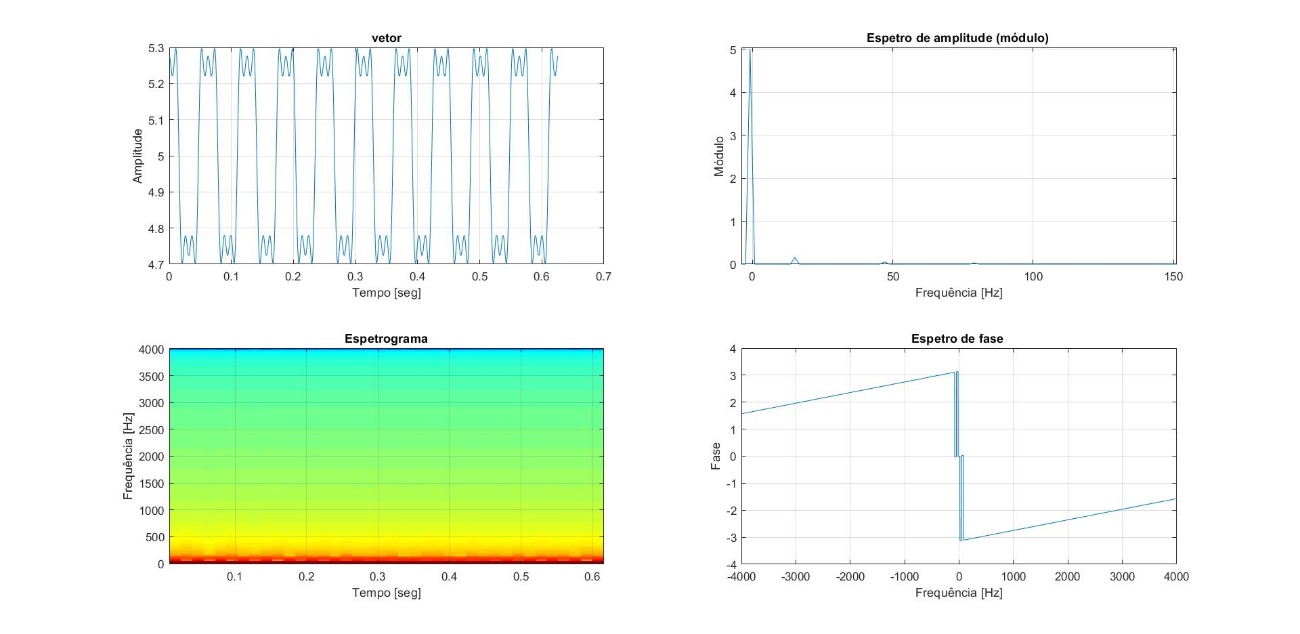
x3(t)=

Figura Sinal x3(t) com A0=5, N=5, d=a.5 e fo=500

Com A0=3, N=15, d=0.25, fo=1000

X3(t)=

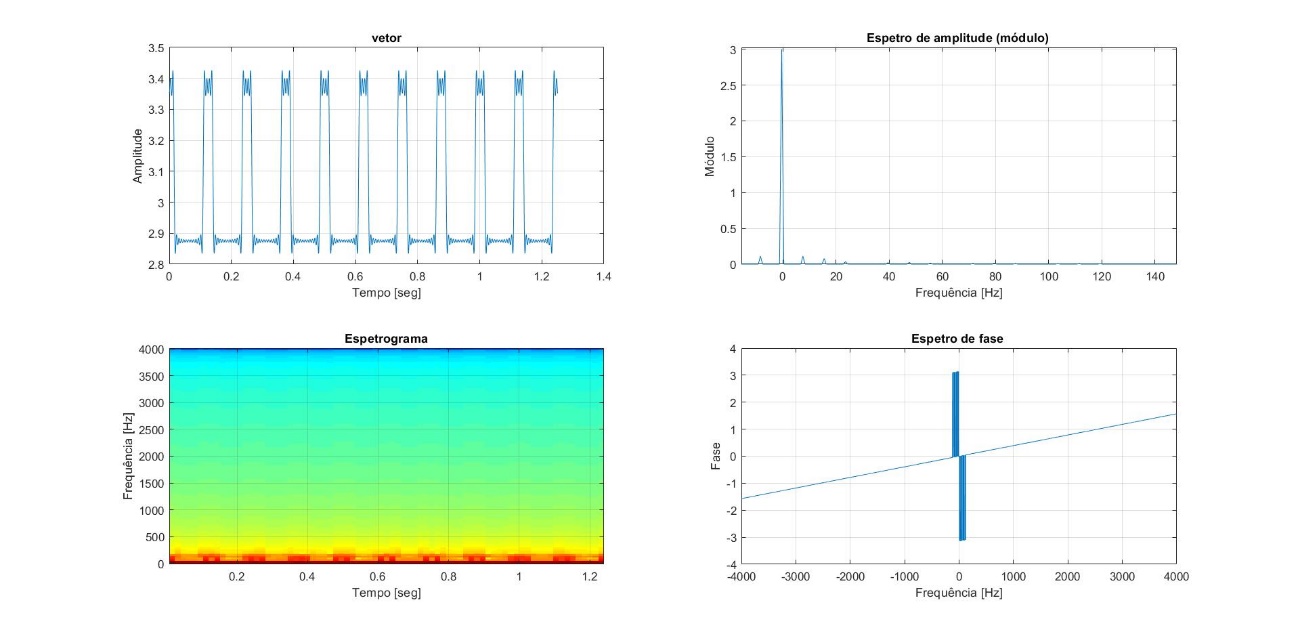


Figura Sinal x3(t) com A0=3, N=15, d=0.25, fo=1000

(ii). A evolução do valor médio do sinal, só está dependente do valor A0, se este aumenta, o valor médio aumenta, se este diminui, o valor médio também diminui consequentemente. Quaisquer que sejam as mudanças nos outros parâmetros do sinal, não influenciam em nada o seu valor, ou seja, quanto maior ou menor o N, A0 não altera.

Já a largura de banda, não depende do valor de A0, ou seja, a largura de banda não irá aumentar nem diminuir com o aumento ou diminuição do valor de A0. Com o aumento do N, o valor da largura de banda também aumenta visto que quanto mais amostras temos, maior será o seu valor máximo.

A potência total do sinal é a soma de todas as potências instantâneas , sendo assim, quanto mais amostras temos (N), maior será a potência final, assim como a variação do valor A0, se este aumentar, a potência aumenta, no outro caso, se A0 diminuir, o valor da potência diminui.

# d)

Para desenvolver a função *mobile\_ring\_tones*, foram usadas as funções *record\_audio\_to\_file* e *signal\_mixer.*   
A função *record\_audio\_to\_file* foi ligeiramente alterada que retorna os valores do áudio gravado.  
Ao executar a função *mobile\_ring\_tones*, primeiro existe um momento de preparação de variáveis, depois grava-se 3 ficheiros de áudio no formato .wav e .flac, de seguida junta-se os valores do áudio gerados e por último grava a junção das 3 gravações em ficheiros no formato .wav e .flac.

# Parte 4

# a)

A alteração feita na função *notes*, foi a adição do sentido em que é reproduzido a oitava. O modo ascendente já estava implementado, por isso foi necessário criar uma condição para reproduzir a oitava em modo ascendente ou descendente. Para implementar o modo descendente foi preciso primeiro calcular a frequência digital final e para cada nota tocada, a frequência foi diminuindo, ao contrário do modo ascendente.

# b)

Para desenvolver o GUI a nível gráfico foi tomado em consideração o exemplo que os professores deram em anexo para a realização deste trabalho.

A nível algorítmico, a função *notes* é chamada sempre que o botão Up Scale ou Down Scale é carregado. No entanto, quando o utilizador premir um destes botões, ambos ficam inativos enquanto a função *notes* é executada.

Para desenvolver a funcionalidade do botão de Exit, é forçado fechar todas as figuras.