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)

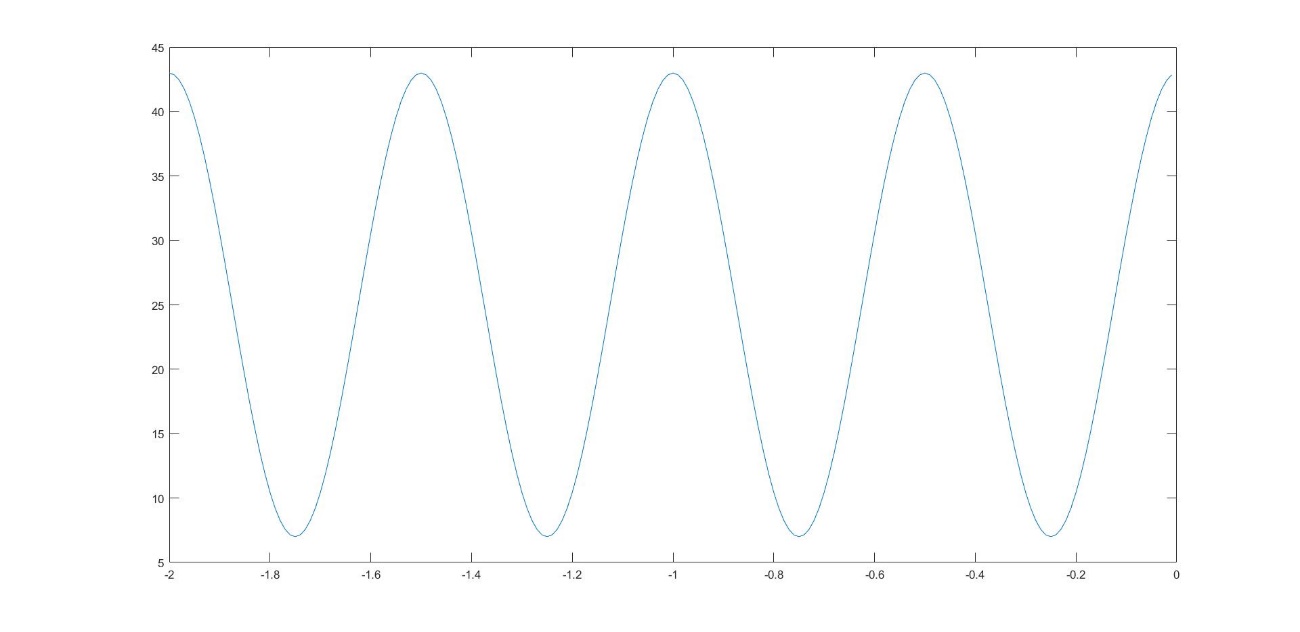


Figura Amostra de 4 periodos fundamentais do sinal w

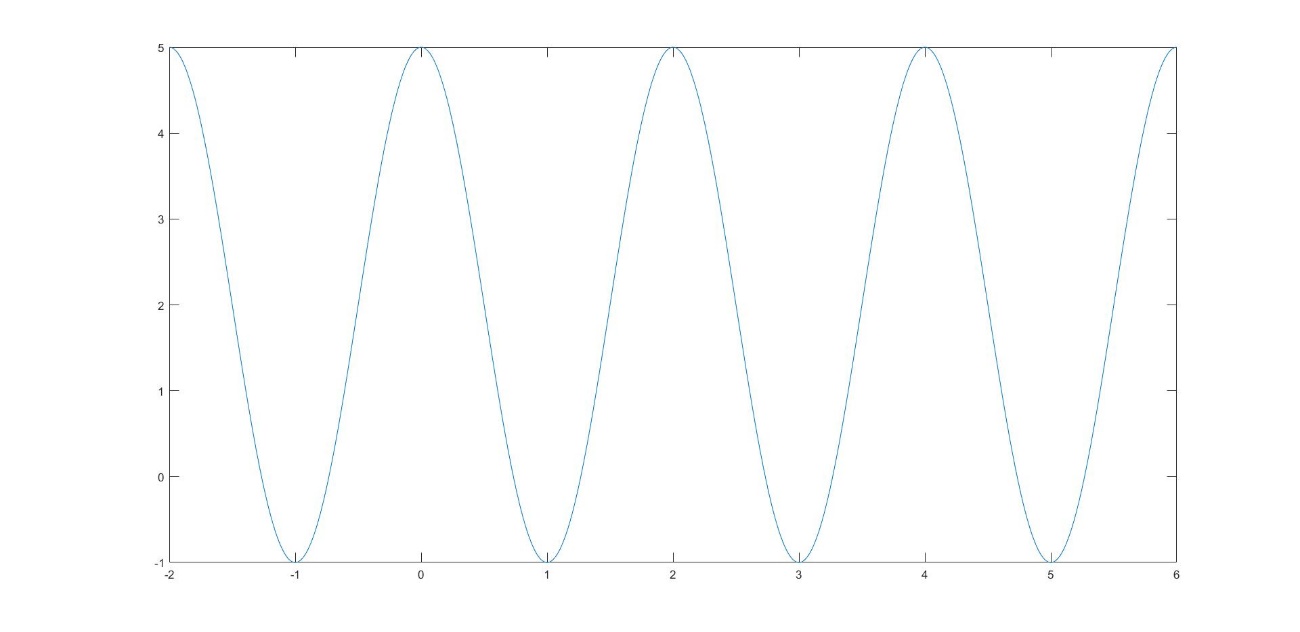


Figura Amostra de 4 periodos fundamentais do sinal z

f)

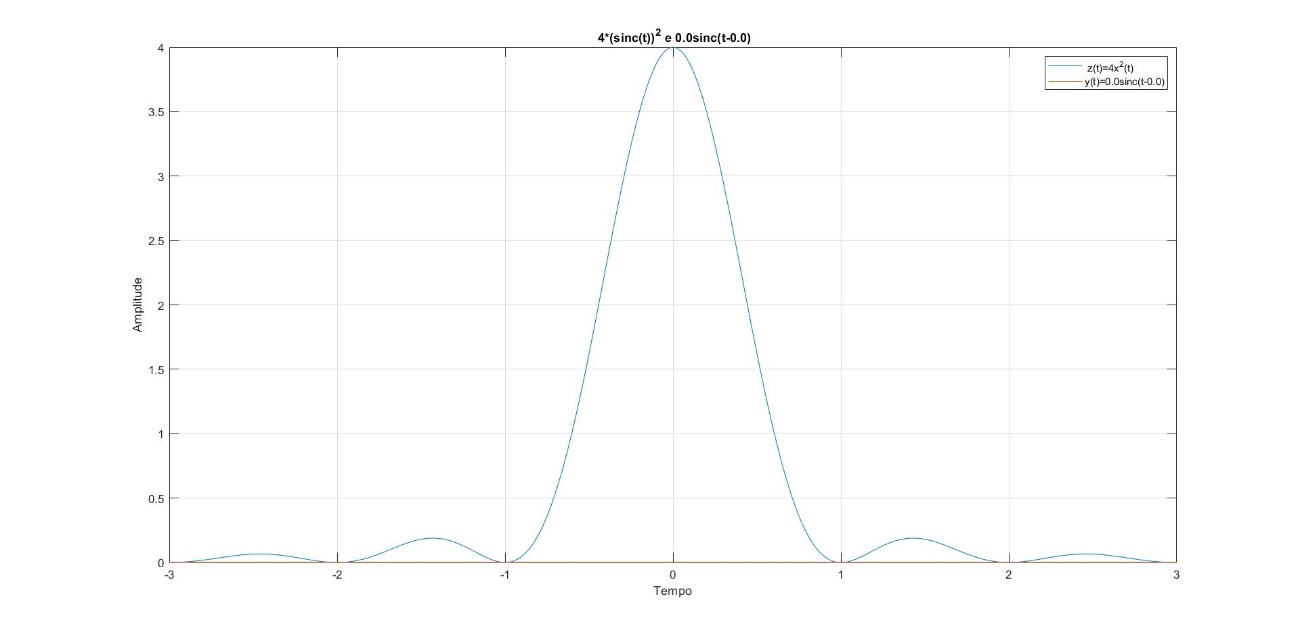


Figura Sinais z em azul e w a vermelho com a=0 e b=0

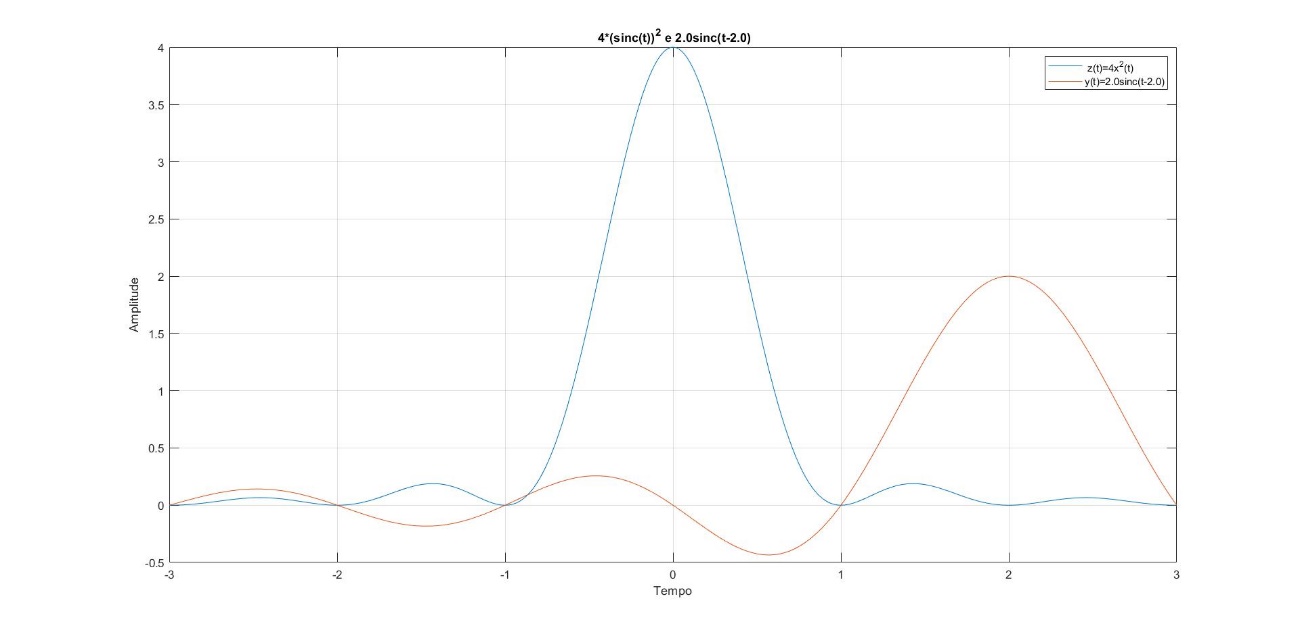


Figura Sinais z em azul e w a vermelho com a=2 e b=2

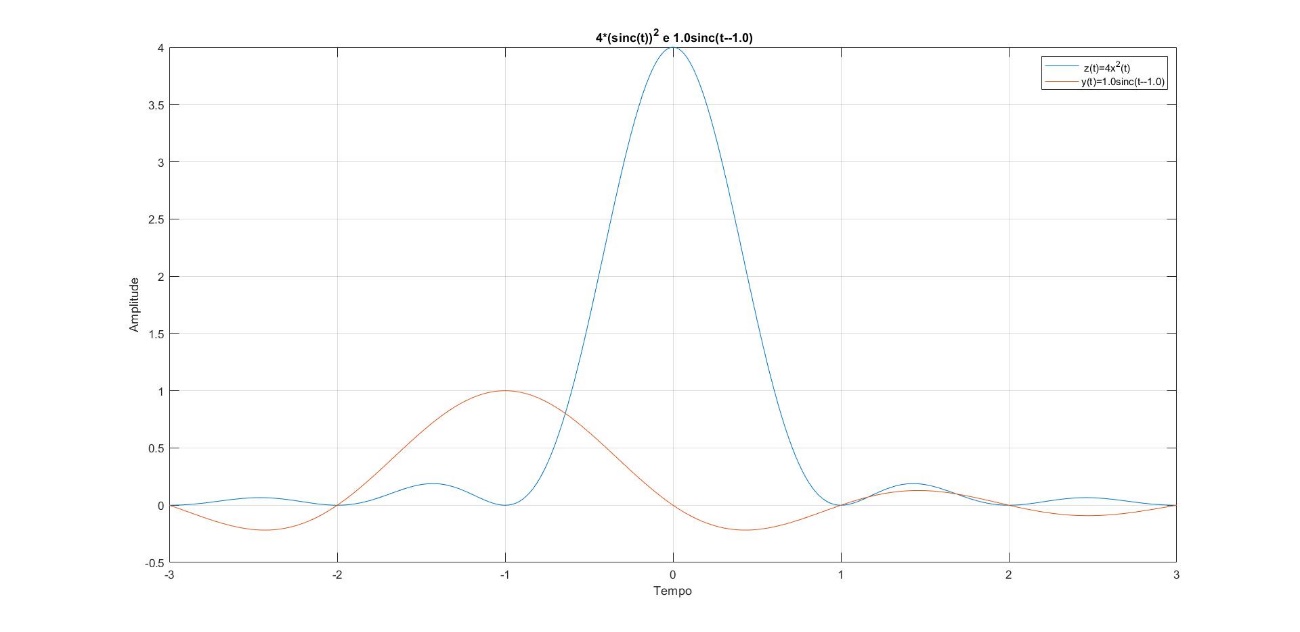


Figura Sinais z em azul e w a vermelho com a=1 e b=-1

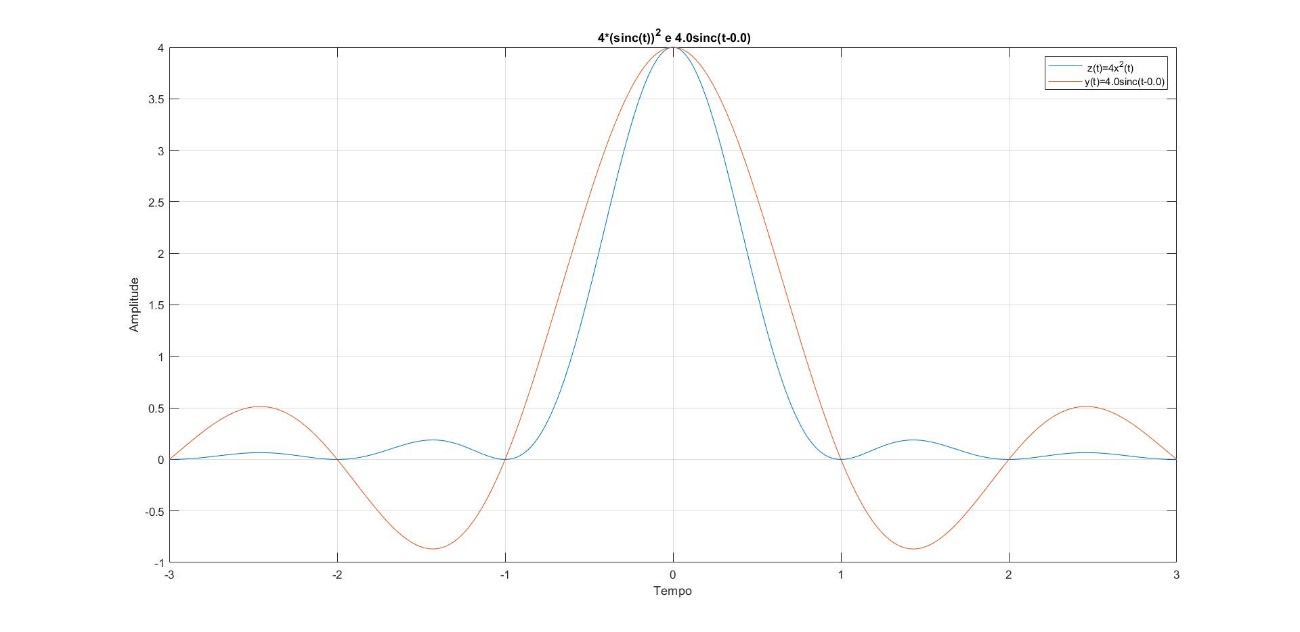


Figura Sinais z em azul e w a vermelho com a=4 e b=0

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

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.

c) i)

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 .

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.