2º Trabalho Prático de
Programação de Computadores
ISEL / DEEA

1º Semestre de 2019/20 - Data de inicio 23/11/2019 - entrega 2/Jan/2020

Trabalho a realizar por grupos de 1 ou 2 alunos

1 - Introdução:

O aparecimento de microprocessadores e micro-controladores de muito baixo custo veio permitir a criação de aplicações informáticas que aplicam técnicas de processamento digital de sinais, com bastantes vantagens sobre soluções analógicas equivalentes, pois podem ser facilmente reconfiguradas bastando alterar o software que executam, com muitas aplicações na área do controlo e monitorização de sistemas, entre outras.

Para desempenhar essas tarefas, as aplicações necessitam recolher dados acerca do estado dos sistemas que estão a controlar ou monitorizar, que podem ser obtidos recorrendo a diversos tipos de sensores que convertem grandezas físicas em sinais elétricos. Por exemplo, temperaturas, pressões de gases ou líquidos, posição e velocidades de motores, tensões e correntes, etc.

Os valores produzidos pelos sensores são depois lidos periodicamente pelos micro-processadores/micro-controladores/DSPs que possuem pinos especiais de entrada que permitem ler valores digitais ou analógicos instantâneos.

Aos valores lidos nessas entradas dá-se o nome de amostras e o ritmo a que os valores são lidos denomina-se frequência de amostragem. A figura 1 apresenta uma sequencia de amostras extraída de um sinal sinusoidal com freq. 50Hz e 320V de amplitude. Este sinal foi amostrado com uma frequência de 3kHz, pelo que foram adquiridas 60 amostras por cada período da onda original.

O espaço de tempo entre amostras, chamado período de amostragem, é o inverso da frequência de amostragem. Neste exemplo, o período de amostragem h = 1.0s/3000 = 333.3 micro-segundos.

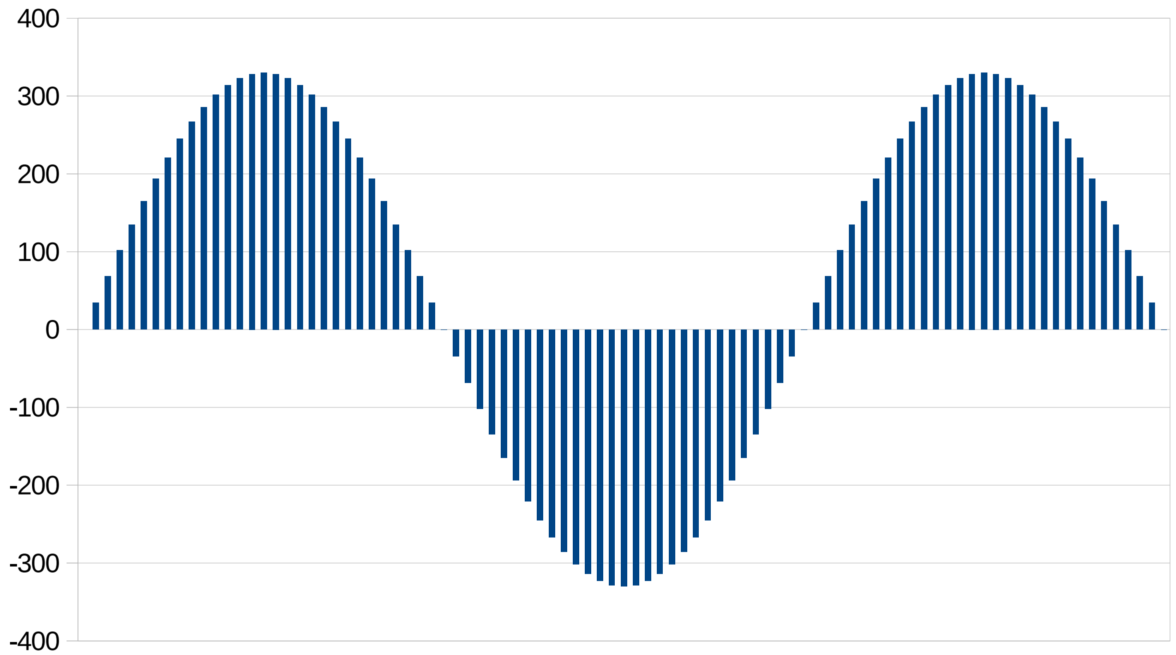


Fig 1: Amostras de um sinal sinusoidal (V)

2 - Trabalho prático:

Para este trabalho, pretende-se criar um programa com capacidade para ler ficheiros de texto contendo amostras de sinais, executar processamento sobre esses sinais e voltar a criar ficheiros de texto com os sinais resultantes. Os ficheiros resultantes, devem obedecer ao formato CSV (comma-separeted-values), podendo ser facilmente lidos por folhas de cálculo (MS Excell, Libre-Office, etc.), que permitem visualizar os resultados de forma gráfica.

2.1 - Estruturas de dados

Crie uma estrutura de dados (struct) “sinal”, para guardar na memória do computador os dados de um sinal, incluindo os seguintes campos:

a) Frequência de amostragem (inteiro)
b) Quantidade de amostras recolhidas (inteiro)
c) Apontador para um “array” com valores das amostras (valores reais de Voltagem)

2.2 - Ficheiros de entrada

Crie uma função em linguagem C para ler dados de um sinal a partir de um ficheiro de texto, de acordo com as seguintes especificações:

a) A função recebe como parâmetro uma string com o nome/caminho do ficheiro a abrir (previamente preenchida pelo programa antes de executar a função)
b) A função retorna como resultado uma estrutura “sinal”, contendo a informação obtida do ficheiro
c) A função deve alocar dinamicamente um “array” para guardar as amostras lidas
d) O ficheiro de texto, tem uma valor por linha e obedece ao formato do exemplo seguinte:

sinal.txt

```
1000        → Frequência de amostragem
20          → Quantidade de amostras
0.0         → Primeira amostra (t = 0)
10.0        → Segunda amostra
11.3        → Terceira amostra
15.0
5.1
0.0
-5.0
-12
-15.3
-8.2
-3.1
0.1
5.33
10.25
16.2
8.1
3.2
-1.1
-8.0
-12.3           → Amostra N. 20 (ultima)
```

2.2 - Ficheiros de saída

Crie uma função em linguagem C para guardar os dados de dois sinais num ficheiro de texto em formato CSV.

a) A função recebe com parâmetros de entrada o nome/caminho do ficheiro a criar e dois apontadores para duas estruturas sinal, correspondendo a um sinal de entrada e outro de saída.
b) A função deve verificar se o número de amostras dos dois sinais é igual.
Se não acontecer, deve terminar imediatamente.
c) Deve criar um ficheiro de texto de acordo com o formato do seguinte exemplo:

resultado.csv

```
#Tempo     Entrada    Saída
0.000;     0.0;       0.0
0.001;     10.0;      20.0
0.002;     11.3;      22.6
0.003;     15.0;      30.0
0.004;     5.1;       10.2
0.005;     0.0;       0.0
0.006;     -5.0;      -10.0
0.007;     -12.0;     -24.0
0.008;     -15.3;     -30.6
0.009;     -8.2;      -16.4
0.010;     -3.1;      -6.2
0.011;     0.1;       0.2
0.012;     5.33;      10.66
0.013;     10.25;     20.5
0.014;     16.2;      32.4
0.015;     8.1;       16.2
0.016;     3.2;       6.4
0.017;     -1.1;      -2.2
0.018;     -8.0;      -16.0
0.019;     -12.3;     -24.6
```

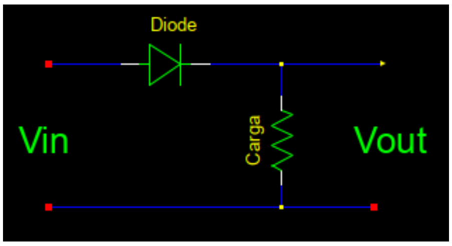
Em que a primeira linha contém uma comentário começado por “#” com a descrição de cada uma das colunas. Cada uma das linhas seguintes contém 3 valores separados por ponto-e-virgula “;”, correspondentes ao um valor de tempo e duas amostras correspondentes de cada sinal (entrada e saída).

NOTA: Este formato de texto pode ser aberto por aplicações de folha de cálculo para apresentar gráficos mostrando os dois sinais, em que a primeira coluna deve ser usada para escala do eixo X. Em cada linha do ficheiro, tempo associado a cada amostra começa em zero e é calculado com base no período de amostragem, que neste exemplo corresponde 1/1000s.

2.3 - Processamento de sinais

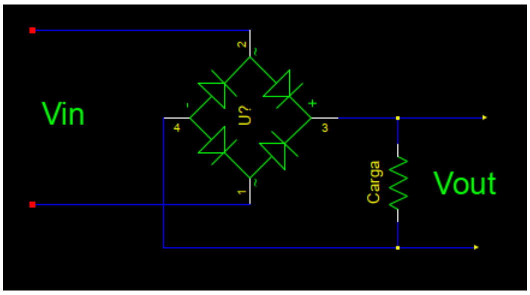
Crie várias funções em linguagem C que recebem como parâmetro um apontador para um sinal e produzem como resultado outro sinal com o mesmo número de amostras e frequência de amostragem. Cada função deve criar uma nova estrutura para o sinal de saída, alocar dinamicamente espaço para as amostras do resultado e calcular os valores do sinal de saida com base na amostra correspondente (e anteriores), do sinal de entrada.

a) Retificação de meia onda:



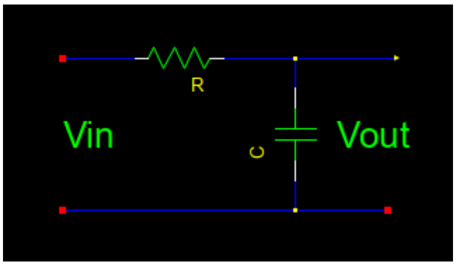
y(t) = x(t) quando x(t) > 0
y(t) = 0 quando x(t) < 0

b) Retificação de onda completa:



y(t) = x(t) quando x(t) > 0
y(t) = -x(t) quando x(t) < 0

c) Filtro RC:



Um circuito RC rege-se pela seguinte equação diferencial:

$$\frac{\partial Vc}{\partial t}=\frac{1}{RC}(Vin-Vc)$$

Que pode ser resolvida numericamente usando vários métodos, como por exemplo, o método de Euler-melhorado:

Dada a equação diferencial genérica: $\frac{\partial y}{\partial x}=f(x,y)$

O método de Euler melhorado calcula valores sucessivos de Y, partido de valores anteriores de X e Y, usando as seguintes formulas:

$$K1=f(x[i-1],y[i-1])$$
$$K2=f(x[i],y[i-1]+h*K1)$$
$$Y[i]=Y[i-1]+h*\frac{(K1+K2)}{2}$$

Notas:
1 - Assume-se que inicialmente o condensador C está descarregado (Vc = 0)
2 - Será usado um valor de «h» correspondente ao período de amostragem
3 – Para a eq. diferencial do circuito RC ficamos com: $f(x,y)=\frac{1.0}{RC}(Vin-y)$
4 - Começando com Y[0] = 0, a função deve aplicar as fórmulas do método de E.M. a todas as restantes amostras do sinal [1 a N-1].

2.4 - Programa principal

Elabore um programa principal que apresente ao utilizador um menu semelhante ao seguinte:

1 – Abrir ficheiro de amostras
2 – Aplicar retificação de meia onda
3 – Aplicar retificação de onda completa
4 – Aplicar filtro RC
5 – Guardar ficheiro de resultados
6 – Sair do programa

Em que:

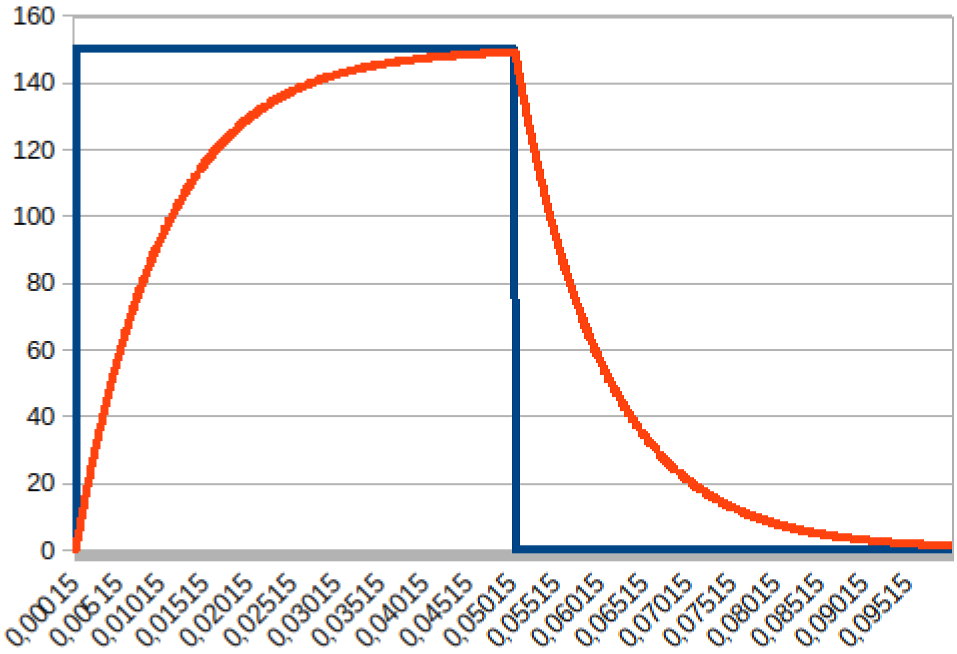
a) O programa deve correr em ciclo até o utilizador escolher a opção 6.
b) As opções 1 e 5 devem perguntar ao utilizador o nome/caminho do ficheiro a abrir
c) A opção 4 deve perguntar ao utilizador os valores de R (Ω) e C (uF).

Notas adicionais:

1 - Conforme as definições de língua/país do computador, o conjunto de caracteres usado na janela de terminal e o caracter decimal ('.' ou ','), poderá variar. Dessa forma podem ocorrer erros ao ler os ficheiros em anexo. Para evitar esses problemas a função **main** poderá começar por configurar o as definições de língua da seguinte forma:

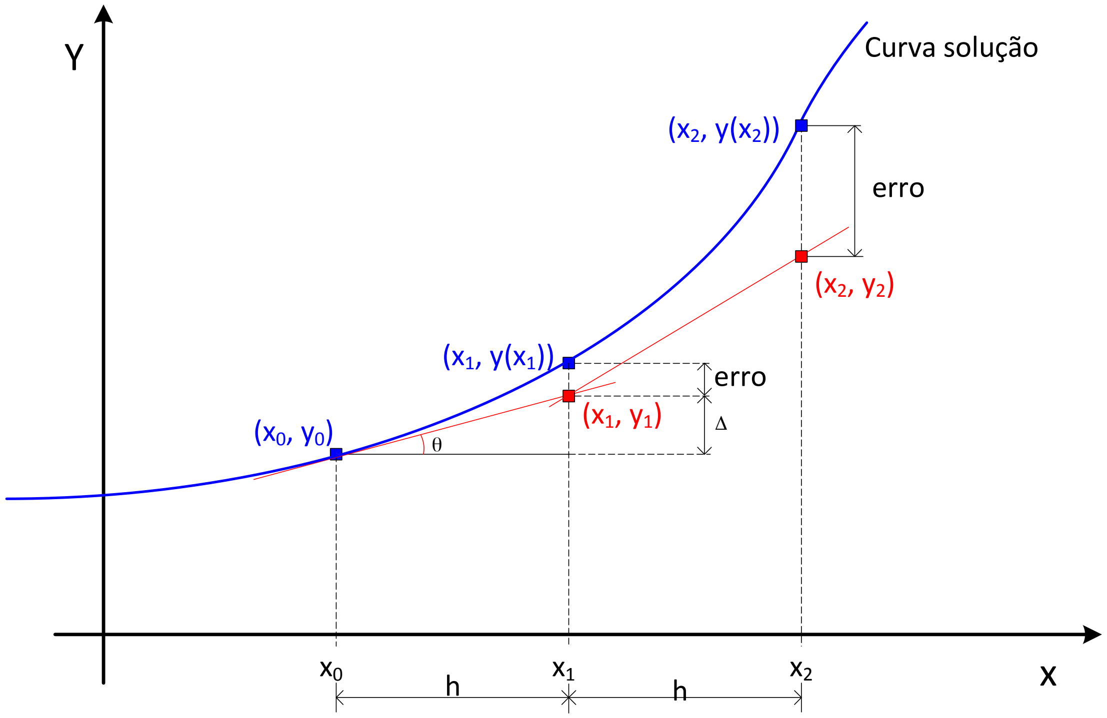
```
#include <locale.h>
.
.
.
setlocale( LC_ALL, "pt_PT" ); // Língua Portuguesa – janela terminal com caracteres nacionais
setlocale( LC_NUMERIC. "C" ); // Usar '.' como caracter decimal em vez de ','
```

2 – Para validar as funções criadas, poderá visualizar graficamente os ficheiros de saída numa folha de cálculo, como mostra o próximo exemplo:



ANEXO

A1 - Método de Euler



A curva a azul representa a solução da equação diferencial

$$\frac{dy}{dx} = f(x, y)$$

Em que se conhece a condição inicial

$$y(x_0) = y_0$$

O método de Euler consiste no cálculo de sucessivos pontos (x_i, y_i) (representados a vermelho), considerando um valor *h* entre cada 2 pontos sucessivos.
Conhecendo o valor de y_0

$$y_1 = y_0 + \Delta$$

Sendo

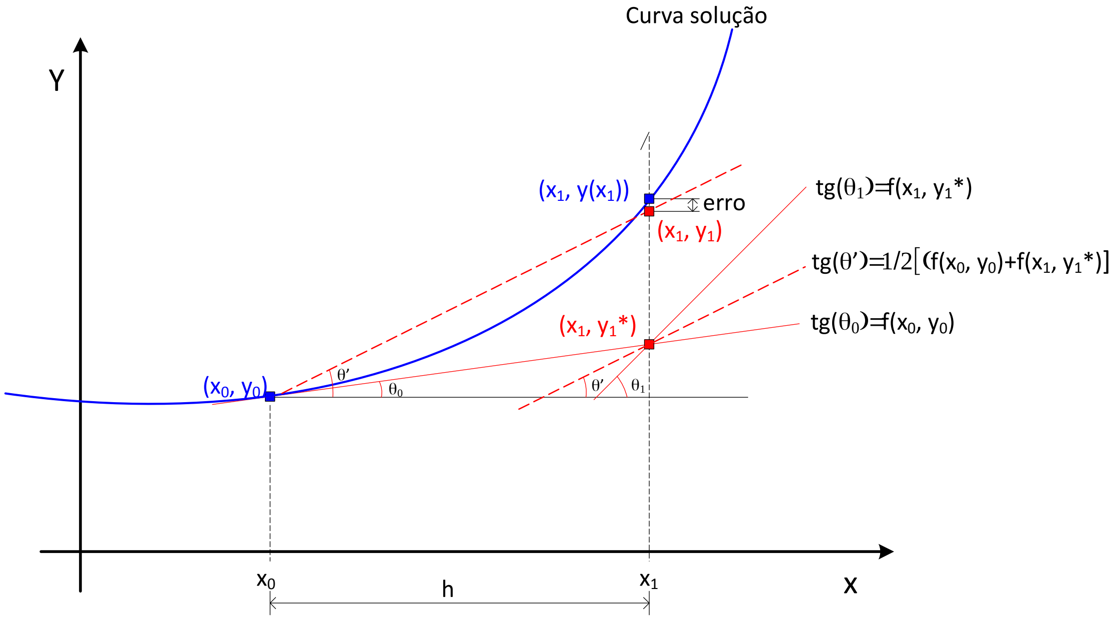
$$tg(\theta) = f(x_0, y_0)$$
$$\Delta = f(x_0, y_0) \cdot h$$

Assim, na iteração *i*, temos:

$$\begin{cases} x_i = x_{i-1} + h \\ y_i = y_{i-1} + f(x_{i-1}, y_{i-1}) \cdot h \end{cases}$$

A2 - Método de Euler melhorado

O método de Euler melhorado consiste no cálculo do valor de *y*, com base numa aproximação, *y**, calculada através do método de Euler simples.



O ponto (x_1, y_1) é calculado com base numa reta cujo declive é a média dos declives da reta tangente à curva no ponto (x_0, y_0) e uma aproximação ao declive da reta tangente à função no ponto (x_1, y_1*).

Os declives destas retas são:

$$k_1=f(x_0, y_0)$$
$$k_2=f(x_1, py_1)$$

Com

$$py_1=y_0+k_1\cdot h$$

O valor de y_1 é obtido por:

$$y_1=y_0+\frac{k_1+k_2}{2}\cdot h$$

Generalizando, para a iteração *i*,

$$\begin{cases} k_1=f(x_{i-1}, y_{i-1}) \\ k_2=f(x_i, y_{i-1}+k_1\cdot h) \\ y_i=y_{i-1}+\dfrac{k_1+k_2}{2}\cdot h \end{cases}$$

A3 – Exemplo de resolução:

A3.1 – Dados do problema:

C = 10 μF = 0.00001 F
R = 1 kOhm = 1000 Ohm

Sinal de entrada Vin:

1000	Frequência de amostragem 1kHz
5	5 Amostras
0.000	Amostra N.0 (Volt)
15.000	N.1
25.000	N.2
30.000	N.3
23.500	Amostra N.4

Período de amostragem: h = 1.0/1kHz = 0.001s (1ms)

Da eq. do sistema: $\frac{\partial Vc}{\partial t} = \frac{1}{RC}(Vin - Vc)$

Obtemos a função: **f(x, y) = 1.0/(R*C) * (x – y)**

A3.2 - Calculo das amostras do sinal de saída Y (Vout):

Amostra inicial, i = 0:
Y[0] = 0V (Condições iniciais – o condensador está descarregado)

Amostra seguinte, i = 1:
k1 = f(Vin[0], Y[0]) = f(0, 0) = 1.0/(R*C) * (Vin[0] – Y[0]) = 1.0/(R*C) * (0-0) = 0
k2 = f(Vin[1], Y[0] + k1 * h) = f(15.0, 0 + 0 * 0.001) = 1.0/(R*C) * (15.0 – 0) = 1500
Y[1] = Y[0] + h * (k1 + k2) / 2 = 0 + 0.001 * (0 + 1500) / 2 = 0.75V

Amostra i=2:
k1 = f(Vin[1], Y[1]) = f(15.0, 0.75) = 1.0/(R*C) * (Vin[1] – Y[1]) = 1425
k2 = f(Vin[2], Y[1] + k1 * h) = f(25.0, 0.75 + 1425 * 0.001) =
f(25.0, 2.175) = 1.0/(R*C) * (25 - 2.175) = 2282.5
Y[2] = Y[1] + h * (k1 + k2) / 2 = 0.75 + 0.001 * (1425 + 2282.5) / 2 = 2.60375V

Amostra i=3:
k1 = f(Vin[2], Y[2]) = f(25.0, 2.60375) = 1.0/(R*C) * (Vin[2] – Y[2]) = 2239.625
k2 = f(Vin[3], Y[2] + k1 * h) = f(30.0, 2.60375 + 2239.625 * 0.001) =
f(30.0, 4.843375) = 1.0/(R*C) * (30 - 4.843375) = 2515.6625
Y[3] = Y[2] + h * (k1 + k2) / 2 = 2.60375+ 0.001 * (2239.625 + 2515.662) / 2 = 4.9813935V

Amostra i=4:
k1 = f(Vin[3], Y[3]) = f(30.0, 4.9813935) = 1.0/(R*C) * (Vin[3] – Y[3]) = 2501.86065
k2 = f(Vin[4], Y[3] + k1 * h) = f(23.5, 4.9813935 + 2501.86065*0.001) = f(23.5, 7.48325415)
= 1.0/(R*C) * (23.5 - 7.48325415) = 1601.674585
Y[4] = Y[3] + h * (k1 + k2)/2 = 4.9813935 + 0.001*(2501.86065 + 1601.674585)/2 = 7.0331611V