

## Anexo

A continuación, se muestran unos ejemplos de los *scripts* de Matlab utilizados para el tratamiento de los datos.

### 1) *Script* capítulo 3 Metodología, 3.3 Cálculo velocidad del sonido en el agua teórica:

```
% CÁLCULO ERROR VELOCIDAD TEÓRICA MEDWIN

clc
clear all
close all

% CONDICIONES AMBIENTALES: T=? (°C), S=0, Z=0
T = 20.2;
S = 0;
z = 0;

% FÓRMULA DE MEDWIN
c = 1449.2 + 4.6*T - 0.055*T^2 + 0.00029*T^3 + (1.34-0.010*T)*(S-35) +
0.016*z

%ERROR DE C SI SOLO SE CONSIDERA LA TEMPERATURA
ET = 0.1; % Error Temperatura

ErrorC = (4.6 - 2*0.055*T + 3*0.00029*T^2 + 0.35 - 0.01*S)*ET
```

### 2) *Script* capítulo 4 Calibración del sistema, 4.2 Montaje con dos transductores distintos:

```
% MEDIDAS REALIZADAS CON 2 TRANSDUCTORES DIFERENTES EL 090517
% CANAL 1: EMISOR
% CANAL 2: RECEPTOR
% VARIACIÓN EN LA DISTANCIA ENTRE TRANSDUCTORES CADA 0,5cm
% GENERADOR DE FUNCIONES: 120kHz, 5 CICLOS, 10Vpp, Burst

clear all
close all
clc

load ('090517_v1.mat')

%INTRODUCIR AQUÍ LAS DISTANCIAS DE LAS MEDIDAS/MUESTRAS
VectorDistancias=[0.03 0.035 0.04 0.045 0.05 0.055 0.06 0.065 0.07
0.075 0.08 0.085 0.09 0.095 0.1 0.105 0.11 0.115 0.12 0.125 0.13 0.135
0.14 0.145 0.15 0.155 0.16 0.165 0.17 0.175 0.18 0.185 0.19 0.195
0.2];

MatrizAyuda_auto=zeros(200,4999);
MatrizAyuda_cross=zeros(200,4999);
VectorDiferenciasAuto=zeros(200,1);
VectorDiferenciasCross=zeros(200,1);
VectorDiferenciasMaximos=zeros(200,1);
VectorX=zeros(40,1);
```

```

VectorY=zeros(40,1);

for k=30:5:200
    eval(['Y1_k= Y1_' num2str(k) ';' ]);
    eval(['Y2_k= Y2_' num2str(k) ';' ]);
    canal1_autocorr=xcorr(Y1_k,Y1_k); %autocorrelación canal 1
    canal2_crosscorr=xcorr(Y2_k,Y1_k); %correlación cruzada 2 sobre 1

    MatrizAyuda_auto(k-29,:)=canal1_autocorr; %en la matriz guardo
    todas las autocorrelaciones
    MatrizAyuda_cross(k-29,:)=canal2_crosscorr; %en la matriz guardo
    todas las correlaciones cruzadas

    %máximos de la AUTOCORRELACIÓN canal 1
    [FilaLocalAuto,
    ColumnaLocalAuto]=find(canal1_autocorr==max(max((canal1_autocorr))));
    %Fila y Columna del máximo en cada iteración
    DiferenciaLocalAuto=(ColumnaLocalAuto/1e7);
    VectorDiferenciasAuto(k-29)=DiferenciaLocalAuto;

    %máximos de la CORRELACIÓN CRUZADA
    [FilaLocalCross,
    ColumnaLocalCross]=find(canal2_crosscorr==max(max((canal2_crosscorr))
    )); %Fila y Columna del máximo en cada iteración
    DiferenciaLocalCross=(ColumnaLocalCross/1e7);
    VectorDiferenciasCross(k-29)=DiferenciaLocalCross;

    %diferencia de los MÁXIMOS de la correlación cruzada y la
    autocorrelación canal1
    VectorDiferenciasMaximos(k-29)=DiferenciaLocalCross-
    DiferenciaLocalAuto; %cruzada-autocorrelación1
end

VectorDiferenciasAuto(VectorDiferenciasAuto==0) = [];
VectorDiferenciasCross(VectorDiferenciasCross==0) = [];
VectorDiferenciasMaximos(VectorDiferenciasMaximos==0) = [];

%--REPRESENTACIÓN GRÁFICA ESPACIO/TIEMPO--

VectorX=VectorDiferenciasMaximos;
VectorY=VectorDistancias';

%--CÁLCULO BARRAS DE ERROR VERTICAL--
figure
ev = (1e-4)*ones(size(VectorX));
vertical=errorbar(VectorX, VectorY, ev, '.r');

%--CÁLCULO BARRAS DE ERROR HORIZONTAL--
hold on
eh = (1e-7)*ones(size(VectorY));
horizontal=herrorbar(VectorX, VectorY, eh, '.r');

scatter(VectorX, VectorY, '.k') %puntos de las medidas en negro
title('Velocidad del sonido en el medio')
ylabel('Espacio (m)')
xlabel('Tiempo (s)')

```

3) *Script* capítulo 5 Caracterización acústica, 5.1 Medidas con caballa (*mackerel*), 5.1.2 Medidas con tejido óseo

```
%Fitxer a analitzar
clear
close all
filename='20170803_MACK1_MUESTRA1_M1_1CICLO';
fitxer=strcat(filename, '.mat')
load(fitxer);
Distancia=15.65; %mm
%Parametres1cicle

DistanciaCorreccio1cicle=12.55; %mm
TempsCorreccio1cicle= 1.3920e-005; %s

%busquem mínims del senyal emés X1,Y1. Hem de veure que la distància
entre
%mínims siga adequada, cosa que dependrà de la freqüència. Triem 100

lMinIndY1 = localMaximum(-Y1_m1,100,true);
ind=1;
valorllindar1=0.005;
for i=1:length(lMinIndY1)
    if (abs(Y1_m1(lMinIndY1(i)))>valorllindar1)

        MinY1(ind)=Y1_m1(lMinIndY1(i));
        IndexMinF(ind)=lMinIndY1(i);
        ind=ind+1;
    end
end
figure()
plot(X1_m1,Y1_m1,'r')
title('Emitted')
hold on
plot(X1_m1(IndexMinF), Y1_m1(IndexMinF),'+k')

%creuament pel 0 entre els primer i segon minim del senyal Y1

ind0_Y1=IndexMinF(1)+find(Y1_m1(IndexMinF(1):IndexMinF(2))==0)-1;

plot(X1_m1(ind0_Y1), Y1_m1(ind0_Y1),'.k')
hold off
%busquem mínims del senyal rebut X2,Y2. Hem de veure que la distància
entre
%mínims siga adequada, cosa que dependrà de la freqüència. Triem 100

lMinIndY2 = localMaximum(-Y2_m1,100,true);
ind=1;
valorllindar2=0.002;
for i=1:length(lMinIndY2)
    if (abs(Y2_m1(lMinIndY2(i)))>valorllindar2)

        MinY2(ind)=Y2_m1(lMinIndY2(i));
        IndexMinF2(ind)=lMinIndY2(i);
        ind=ind+1;
    end
end
figure()
```

```

plot(X2_m1,Y2_m1,'r')
title('Received')
hold on
plot( X2_m1(IndexMinF2), Y2_m1(IndexMinF2),'+k')

%creuament pel 0 entre els primer i segon minim del senyal rebut Y2

ind0_Y2=IndexMinF2(1)+find(Y2_m1(IndexMinF2(1):IndexMinF2(2))==0)-1;

plot( X2_m1(ind0_Y2), Y2_m1(ind0_Y2),'.k')
hold off

DeltaTemps=X2_m1(ind0_Y2(1))-X1_m1(ind0_Y1(1))
TempsCorregit_M1=DeltaTemps-TempsCorrecciocicle
DistanciaCorregida_M1=(Distancia-DistanciaCorrecciocicle)/1000; %la
paso a metros

VELOCITAT_M1=DistanciaCorregida_M1/TempsCorregit_M1

nomefich=strcat('VELOCIDAD_',filename);
save(nomefich,'VELOCITAT_M1');

nomefich=strcat('TIEMPO_',filename);
save(nomefich,'TempsCorregit_M1');

nomefich=strcat('DISTANCIA_',filename);
save(nomefich,'DistanciaCorregida_M1');

```

#### 4) Script capítulo 5 Caracterización acústica, 5.2 Medidas con atún listado (*skipjack tuna*), 5.2.1 Medidas con tejido muscular:

```

% MEDIDAS REALIZADAS CON ATÚN DESCONGELADO EL 290617 - PEZ 2
% CANAL 1: EMISOR
% CANAL 2: RECEPTOR
% GENERADOR DE FUNCIONES: 120kHz, 5 CICLOS, 10Vpp, Burst

clear all
close all
clc

load ('290617_AIRE.mat')

%INTRODUCIR AQUÍ LAS DISTANCIAS DE LAS MEDIDAS/MUESTRAS
VectorDistancias=[0.0198 0.0198 0.0198 0.0397 0.0397 0.0397 0.06225
0.06225 0.06225 0.08043 0.08043 0.08043 0.08089 0.08089 0.08089 0.0583
0.0583 0.0583 0.04797 0.04797 0.04797 0.0341 0.0341 0.0341 0.01481
0.01481 0.01481 0.01817 0.01817 0.01817 0.01915 0.01394 0.02765
0.04559 0.05766];

MatrizAyuda_auto=zeros(40,4999);
MatrizAyuda_cross=zeros(40,4999);
VectorDiferenciasAuto=zeros(40,1);
VectorDiferenciasCross=zeros(40,1);
VectorDiferenciaMaximos=zeros(40,1);
VectorDiferenciaMaximosCorregido=zeros(40,1);
VectorVelocidad=zeros(40,1);

```

```

VectorX=zeros(40,1);
VectorY=zeros(40,1);
for k=1:1:35
    eval(['Y1_mk= Y1_m' num2str(k) ';' ]);
    eval(['Y2_mk= Y2_m' num2str(k) ';' ]);
    canal1_autocorr=xcorr(Y1_mk,Y1_mk); %autocorrelación canal 1
    canal2_crosscorr=xcorr(Y2_mk,Y1_mk); %correlación cruzada 2 sobre
1

    MatrizAyuda_auto(k,:)=canal1_autocorr; %en la matriz guardo todas
las autocorrelaciones
    MatrizAyuda_cross(k,:)=canal2_crosscorr; %en la matriz guardo
todas las correlaciones cruzadas

    %máximos de la AUTOCORRELACIÓN canal 1
    [FilaLocalAuto
ColumnaLocalAuto]=find(canal1_autocorr==max(max((canal1_autocorr))));
%Fila y Columna del máximo en cada iteración
    DiferenciaLocalAuto=(ColumnaLocalAuto/1e7);
    VectorDiferenciasAuto(k)=DiferenciaLocalAuto;

    %máximos de la CORRELACIÓN CRUZADA
    [FilaLocalCross
ColumnaLocalCross]=find(canal2_crosscorr==max(max((canal2_crosscorr))
)); %Fila y Columna del máximo en cada iteración
    DiferenciaLocalCross=(ColumnaLocalCross/1e7);
    VectorDiferenciasCross(k)=DiferenciaLocalCross;

    %diferencia de los MÁXIMOS de la correlación cruzada y la
autocorrelación canal1
    VectorDiferenciaMaximos(k)=DiferenciaLocalCross-
DiferenciaLocalAuto; %cruzada-autocorrelación
    VectorDiferenciaMaximosCorregido(k)=VectorDiferenciaMaximos(k)-
(0.0299/1503.5); %se resta la corrección de la línea de ajuste que
hemos calculado

    %cálculo velocidad del sonido en el medio
    VectorVelocidad(k)=VectorDistancias(k)/VectorDiferenciaMaximosCorregid
o(k);
end

VectorDiferenciasCross(VectorDiferenciasCross==0) = [];
VectorDiferenciasAuto(VectorDiferenciasAuto==0) = [];
VectorDiferenciaMaximos(VectorDiferenciaMaximos==0) = [];
VectorDiferenciaMaximosCorregido(VectorDiferenciaMaximosCorregido==0)
= [];
VectorVelocidad(VectorVelocidad==0) = [];

%--CÁLCULO DEL PROMEDIO DE LAS 3 MEDIDAS DE CADA PUNTO, PARA GRAFICAR
SOLO 1 PUNTO POR MEDIDA--
for j=17:3:29
    VectorX(j)=(VectorDiferenciaMaximosCorregido(j-
1)+VectorDiferenciaMaximosCorregido(j)+VectorDiferenciaMaximosCorregid
o(j+1))/3;
    VectorY(j)=(VectorDistancias(j-
1)+VectorDistancias(j)+VectorDistancias(j+1))/3;
end

VectorX(VectorX==0) = [];
VectorY(VectorY==0) = [];

```

```

%--REPRESENTACIÓN GRÁFICA ESPACIO/TIEMPO--

%--CÁLCULO BARRAS DE ERROR VERTICAL--
figure
ev = (1e-4)*ones(size(VectorX));
vertical=errorbar(VectorX, VectorY, ev, '.r');

%--CÁLCULO BARRAS DE ERROR HORIZONTAL--
hold on
eh = (1e-7)*ones(size(VectorY));
horizontal=herrorbar(VectorX, VectorY, eh, '.r');

scatter(VectorX, VectorY, '.k') %puntos de las medidas en negro
title('Velocidad carne SKJ')
ylabel('Espacio (m)')
xlabel('Tiempo (s)')

```

5) *Script* capítulo 5 Caracterización acústica, 5.2 Medidas con atún listado (*skipjack tuna*), 5.2.2 Medidas con tejido óseo:

```

%cálculo velocidad promedio HUESO SKJ MUESTRA 3

clc
clear all
close all

Velocidad=zeros(10,1);

load('VELOCIDAD_20170803_SKJ2_MUESTRA3_M1_1CICLO.mat');
load('VELOCIDAD_20170803_SKJ2_MUESTRA3_M2_1CICLO.mat');
load('VELOCIDAD_20170803_SKJ2_MUESTRA3_M3_1CICLO.mat');
load('VELOCIDAD_20170803_SKJ2_MUESTRA3_M4_1CICLO.mat');
load('VELOCIDAD_20170803_SKJ2_MUESTRA3_M5_1CICLO.mat');
load('VELOCIDAD_20170803_SKJ2_MUESTRA3_M6_1CICLO.mat');

for k=1:1:6
    eval(['VELOCITAT_Mk= VELOCITAT_M' num2str(k) ';' ]);
    Velocidad(k)=VELOCITAT_Mk;
end

Velocidad(Velocidad==0) = [];

%--VELOCIDAD MEDIA--
VelocidadMedia = mean(Velocidad)

savefile = 'Velocidad_Media_Muestra3.mat';
save(savefile, 'VelocidadMedia');

%--DISPERSIÓN--

vMAX=max(Velocidad);
vMIN=min(Velocidad);

Dispersion = (abs(vMAX - vMIN)/VelocidadMedia)*100;

```

```
disp('Dispersión (%) = ')
disp(Dispersion)

savefile = 'Dispersion_Muestra3.mat';
save(savefile, 'Dispersion');

%--ERROR--

VectorAyuda=zeros(length(Velocidad),1);

for i=1:length(Velocidad)
    VectorAyuda(i)=(VelocidadMedia-Velocidad(i))^2;
end

Suma = sum(VectorAyuda);
Error = (sqrt(Suma))/(length(Velocidad) - 1)

savefile = 'Error_Muestra3.mat';
save(savefile, 'Error');
```