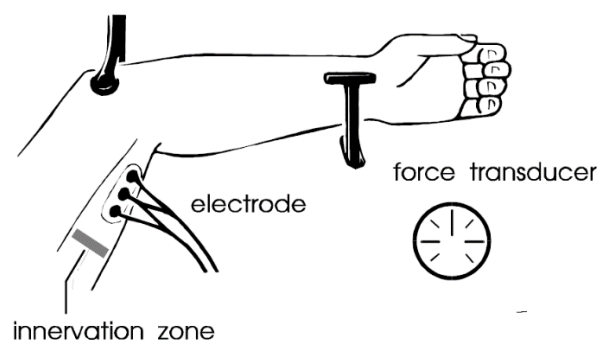


# Digital Biosignal Processing

## MATLAB Laboratory 1

Two EMG signals have been recorded from the biceps brachii muscle of a healthy subject during a voluntary contraction. The two EMG bipolar systems have been spaced by 24 mm and mounted between the innervation zone and the distal tendon. A segment of these signals is saved in the matlab data file Signals\_EMG.mat. The sampling frequency during the experimental recording was 2048 Hz. The aim of this exercise is to estimate with high accuracy the conduction velocity of the active muscle fibres by processing the two recorded signals. A scheme of the experiment is reported below.



Study the short Matlab script provided and understand the meaning of the parameters. Use the Matlab script to estimate conduction velocity from the signals provided. Change the parameters of the script to obtain estimates with different step sizes, downsampling factors, and number of steps. The script provides plot of the signals before and after time shift.

**In your report, please provide the following:**

- For the optimal alignment case, plot the two signals in time domain and report the set of parameters (step size, number of steps, downsampling factor) used for this plot in a table. Report the optimal delay and estimated conduction velocity. [30%]
- Plot the estimation error (variable MSE\_vect in the Matlab function) as a function of the delay, for downsampling factors  $M=1,2,4,8$  (please include all graphs for the different  $M$  in the same plot with different colors). For each  $M$ , report the optimal delay, estimated conduction velocity, and optimal MSE in a table. Comment on the estimates in the different cases of downsampling factors. [70%]

**PLEASE NOTICE:** The report is limited to 1 A4 page, including all graphs and comments.

```

close all
clear all
fclose('all')
load('Signals_EMG.mat'); % Loading the recorded EMGs (two channels)
n_step = 100; % Number of steps in the loop for optimal alignment
stepSize = 0.2; % Size of the step of non-integer delay applied to one of
the signals for alignment
ied=24; % Interelectrode distance of the recordings in mm
Fs = 2048; % Recording sampling frequency
Ts = 1/Fs; % Sampling interval

% Downsampling by an integer
M = 1; % Downsampling factor
channel1 = channel1(1:M:end); % Downsampling first channel
channel2 = channel2(1:M:end); % Downsampling second channel
Fs=Fs/M;
Ts=Ts*M;
% End downsampling

timeAxis=[1:length(channel1)].*Ts.*1000; % Definition of time axis in ms
freqAxis=fftshift([-0.5:1/(length(channel1)):0.5-1/(length(channel1))]); %
Definition of discrete frequency axis

figure(1);plot(timeAxis,channel1,'k');hold on; plot(timeAxis,channel2-
1000,'k'); % Plot of the two recordings after down-sampling
xlabel('Time (ms)')
ylabel('Signals (AU)')

channel1_ft = fft(channel1); % Fourier transform of the first channel

figure(2)
for uu = 1 : n_step
    channel1_dt = (channel1_ft).*exp(-i*2*pi*stepSize*uu*freqAxis); %
complex exponential multiplication (delay in frequency domain)
    channel1_dt = real(ifft((channel1_dt))); % inverse transform to go back
to the time domain
    plot(timeAxis,channel1_dt,'r'); % Plot of the time-shifted signal
    hold on
    plot(timeAxis,channel2,'k');
    uu
    MSE_vect(uu)= sum((channel1_dt - channel2).^ 2)./sum(channel2.^
2).*100; % normalized mean square error between aligned signals
    delay(uu) = stepSize*uu; % Imposed delay in samples
end;
xlabel('Time (ms)')
ylabel('Signal amplitude (AU)')

% Identification of the optimal delay (minimum mean square error)
[MSEopt, optDelay] = min(MSE_vect);

% Plot of optimal alignment
[Here please complete with plot instructions of the two
signals with optimal alignment. This means that channel1 has
to be plotted after shift by the optimal delay.]
% Delay and conduction velocity estimate
fprintf('The optimal delay is %2.2f ms \n',delay(optDelay)*Ts*1000);
fprintf('The estimated conduction velocity is %2.2f m/s
\n',ied/(delay(optDelay)*Ts*1000));
fprintf('Optimal MSE between signals: %2.2f %%\n',MSEopt);

```