Detection of PAF using ECGs

JOSHUA PETER EBENEZER – 15EC10023

DIGITAL SIGNAL PROCESSING PROJECT

ABSTRACT

This report presents a scheme to differentiate between the ECG (Electrocardiographs) signals of patients having healthy hearts and patients showing symptoms of paroxysmal atrial fibrillation (PAF). This report improves upon the earlier report by reporting cross validation results, for a larger dataset, and by tuning some of the heuristic parameters. The product of the energy of the lowpass component of ECG signal (less than 12 Hz) and the autocorrelation continues to be used as the distinguishing feature. The maximum likelihood classifier gave a precision of 92% $\pm$ 8.366 and a recall of 70%$\pm$15.8114 for 5 fold cross validation over 25 normal and 25 abnormal signals. The larger dataset used in this module tested how well the model has generalized, but the presence of far outliers in a few rounds of validation might have affected results as well.

INTRODUCTION

The task is to find a feature or features that can differentiate between ECG (Electrocardiographs) signals of patients having healthy hearts and patients showing symptoms of paroxysmal atrial fibrillation (PAF). Atrial fibrillation occurs when the atria, or upper chambers of the heart, lose their normal rhythm and beat chaotically. As the population ages, the prevalence is expected to rise; currently approximately 6% of the US population over the age of 65 are diagnosed with this arrhythmia. The development of accurate predictors of the acute onset of PAF is clinically important because of the increasing possibility of electrically stabilizing and preventing the onset of atrial arrhythmias with different atrial pacing techniques.

DATABASE

The database of two-channel ECG recordings was created for use in the Computers in Cardiology Challenge 2001, and is available in PhysioNet. The learning set that was used by the author consists of 50 records. Each record has two signals, both taken of the same patient at the same time with two different electrodes. Each signal is of 5 minutes' duration and sampled at 128 Hz. 5 fold cross validation was performed with 20 normal and 20 abnormal signals used for training, and the remaining 10 signals used for testing.

CHARACTERISTICS OF THE DATA

The signals were first normalized with respect to total energy (so that all signals have unit total energy). A normal ECG signal, titled 'n01cm.mat' in the database, is shown in Fig. 1 (taken from training set), before normalization. An ECG of a patient with PAF, titled ‘p14cm.mat’ in the database, is shown in Fig. 2 before normalization.

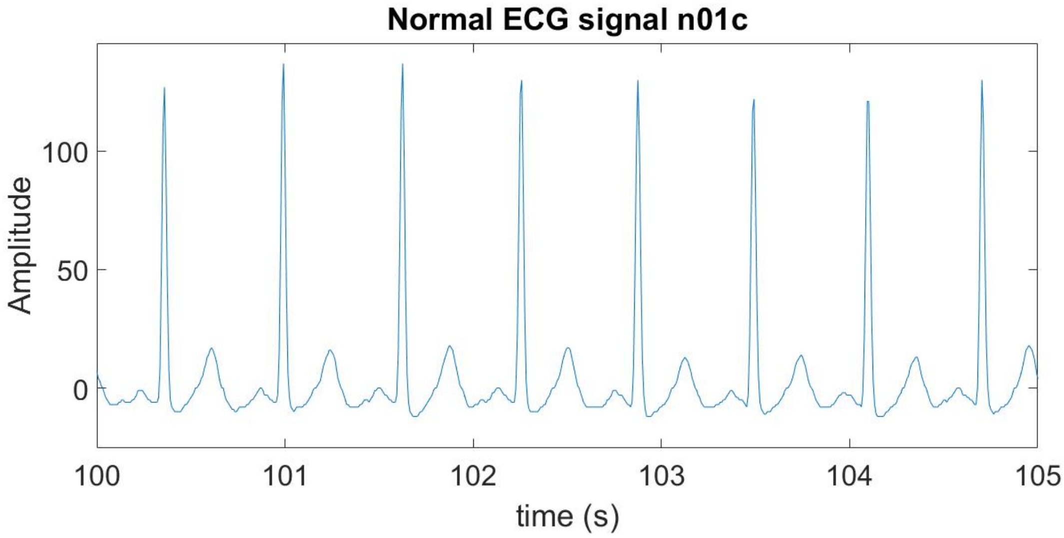

Fig. 1 *Normal ECG signal (time domain)*

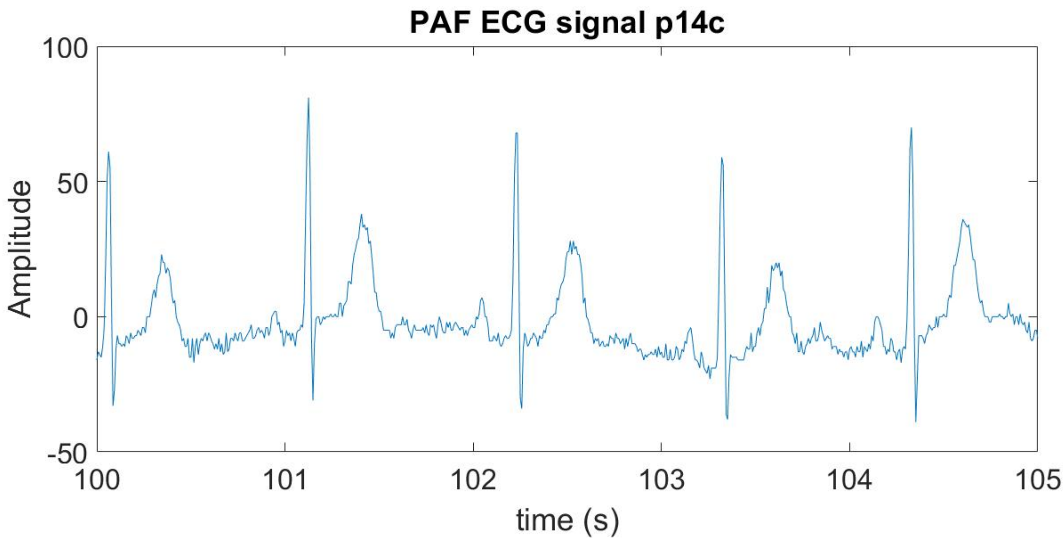

Fig. 2 *ECG signal of patient with PAF (p14c)*

The FFTs of a few normal signals (after energy normalization) are shown in Fig. 3. FFTs of a few abnormal signals (also after energy normalization) are shown in Fig. 4. The spread of the data might be misleading. Abnormal signals actually have a larger peak at 0 Hz than normal signals, hence the spectra near 0 Hz looks smaller in amplitude in comparison to that of normal signals, but it is actually greater. Hence, the summed energy of the lowpass portion of abnormal signals is greater than that of normal signals. This is because the fluctuations in PAF cause a larger, non-zero DC average. Also, while normal signals do not have an envelope, most PAF signals have a very low frequency slowly varying envelope.

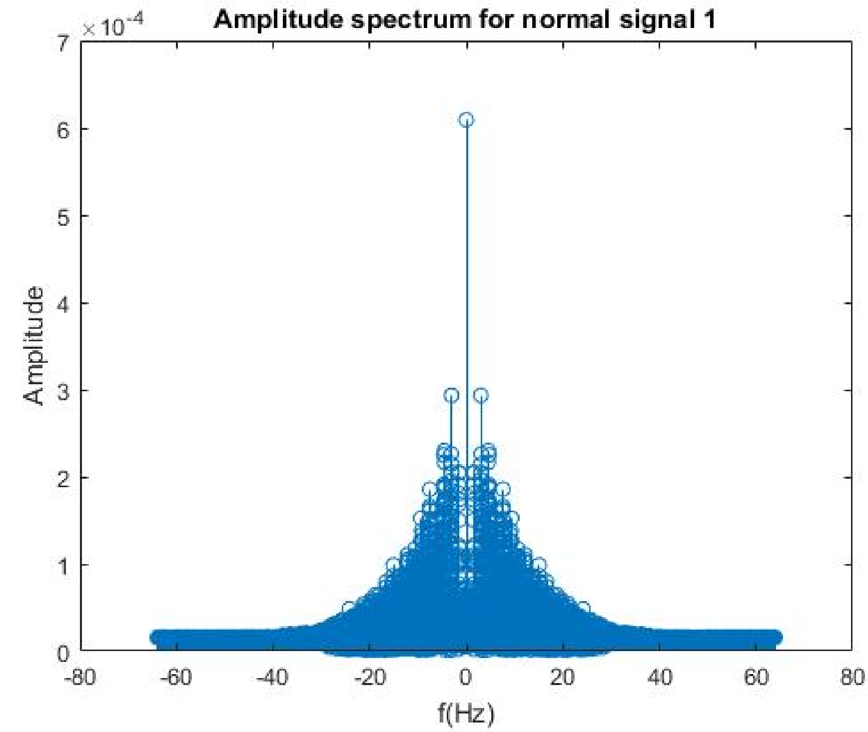

Fig. 3 *Spectrum of normal ECG signal*

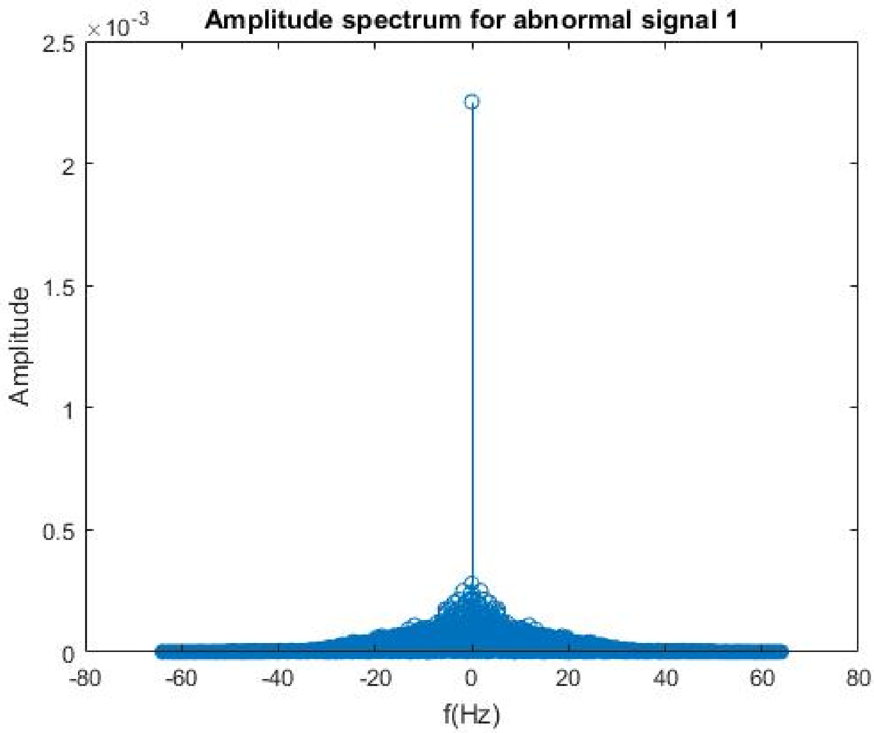

Fig. 4 *Spectrum of abnormal ECG signal*

The absolute auto-correlation of normal signals for a spacing of 1s (around the average duration of a heart beat) is lesser than that of PAF signals. This can also be attributed to the variations in PAF, fluctuating away from baseline with greater amplitudes. Table 1 and 2 show the parameters for data in the training set. Auto-correlation is displayed for signal 2 of each normal record, and signal 1 of each abnormal record. Energy is the average of the signals in each record (after filtering).

Table 1 *Absolute auto-correlation and lowpass (<12 Hz) energy for normal ECGs in training (validation round 5)*

RECORD LABEL	$R_{xx}(10)$	LOWPASS ENERGY
'N01CM.MAT'	0.000370941696483284	0.487346490806009
'N02CM.MAT'	-0.128461888617335	0.304140153339211
'N03CM.MAT'	0.0554480814673563	0.834781870013090
'N05CM.MAT'	0.0262571598116157	0.359736754758502
'N06CM.MAT'	0.0231031178333396	0.298289561869708
'N07CM.MAT'	0.117823661269687	0.421198918578131
'N08CM.MAT'	-0.0347722773589175	0.820996537901168
'N09CM.MAT'	-0.0552832473885434	0.941855495111795
'N10CM.MAT'	-0.182128829412965	0.879899108632147
'N11CM.MAT'	-0.00344861319128152	0.0811776482743613
'N12CM.MAT'	-0.107659773587440	0.556734437758162
'N13CM.MAT'	-0.0970345469044062	0.540446812197107
'N14CM.MAT'	-0.0278969729993582	0.762061876137204
'N15CM.MAT'	0.194807431237560	0.781305356238854
'N16CM.MAT'	0.00426915509837172	0.583427612076782
'N17CM.MAT'	0	0.360069481601244
'N18CM.MAT'	0.0967672960636768	0.678266697162478
'N19CM.MAT'	-0.0261559892432415	0.881254373057782
'N20CM.MAT'	0.0103850849099132	0.900710197953593
'N21CM.MAT'	0.254864255878032	0.825978535822514

Table 2 *Auto-correlation and lowpass (<12 Hz) energy for abnormal ECGs in training (validation round 5)*

RECORD LABEL	$R_{xx}(10)$	LOWPASS ENERGY
'P02CM.MAT'	0.567233309011075	0.764464980632774
'P04CM.MAT'	0.0575637457012276	0.677421025531250
'P06CM.MAT'	0.380670171847789	0.894291870117648
'P08CM.MAT'	0.397532703549358	0.803188709724122
'P10CM.MAT'	0.774279473335052	0.936788339545517
'P12CM.MAT'	0.383656819319209	0.770354147579664
'P14CM.MAT'	0.612175851293025	0.907363911271414
'P16CM.MAT'	0.00999296613949006	0.849103082219564
'P18CM.MAT'	0.553005552553372	0.974035505606387
'P20CM.MAT'	0.0351482782748674	0.958116072377058
'P22CM.MAT'	0.334380459072233	0.756488694364192
'P24CM.MAT'	0.360541346407814	0.977873484677905
'P26CM.MAT'	0.384238273640854	0.693284676517855
'P28CM.MAT'	0.493582895563993	0.961301382705734

'P30CM.MAT'	0.396307086673159	0.767073908176921
'P32CM.MAT'	0.489228522635578	0.952489695241126
'P34CM.MAT'	0.565738656710293	0.802438469647754
'P36CM.MAT'	-0.176584056893377	0.972970299071777
'P38CM.MAT'	0.570062174134492	0.765936000495025
'P40CM.MAT'	0.344587069382144	0.787620577357480

For validation round 5, the mean absolute auto-correlation for normal signals in training is **0.0018**, and for abnormal signals is **0.3078**. The energy of the filtered normal signals is **0.3925**, and that of filtered abnormal signals is **0.7652**. The results are similar for other rounds as well. Hence the product of these two averages for normal signals is much lower than that of abnormal signals.

FILTER USED

The data in the training set was used to find thresholds for energy and auto-correlation. The signals were passed through a Butterworth filter with lowpass cutoff frequency 12 Hz (0.1875π in digital domain), and an order of 4. The frequency response of the filter is shown in Fig. 5.

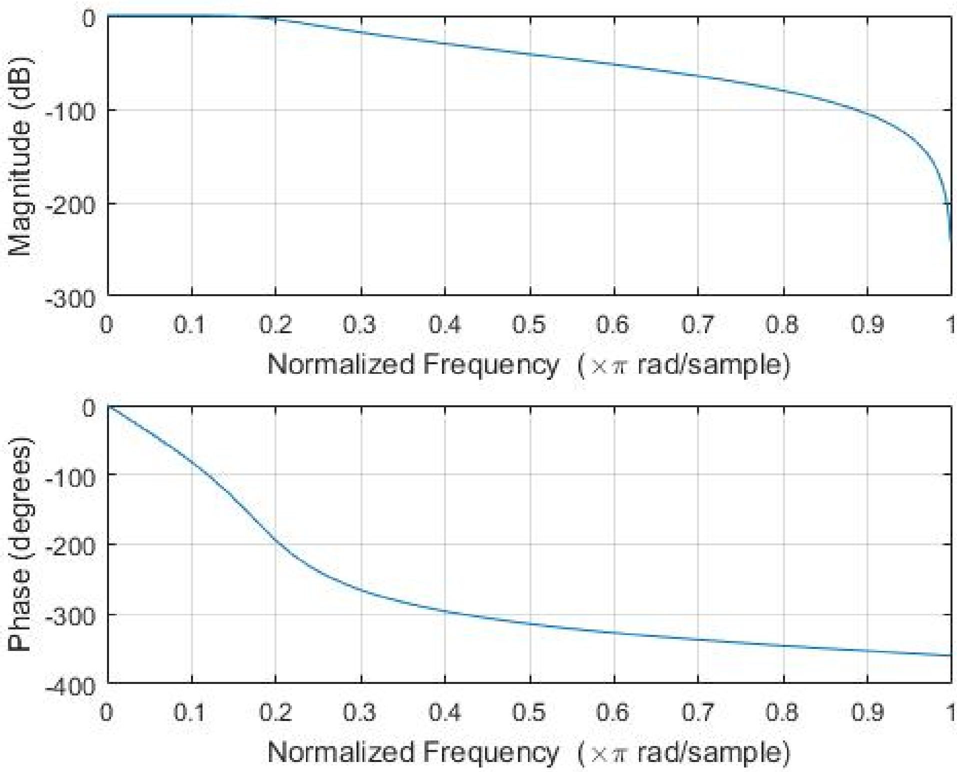

Fig. 5 *Frequency response (magnitude and phase) of the filter used*

Hence, the filtered signal has energy content mostly only in the lowpass region. The energies of the filtered signal were different for normal and abnormal signals, and are tabulated in Tables 1 and 2.

VALIDATION

In the first round of validation, records 1-20 of each set (normal and abnormal) were used for training, and 20-25 were used for validation. In the second set, 5-25 were used for training, and so on.

Precision for the 5-fold cross-validation was [100,90,90,80,100] and recall was [60,70,80,90,50] in percentage. Mean precision is 92% with a standard deviation of 8.366, and mean recall is 70% with a standard deviation of15.8114. The poor recall in the 1st and last rounds might be due to the presence of outliers. Since only 5 signals are used for validation, presence of a few outliers can cause significant variation in results. A larger dataset might provide more accurate results, with lesser deviations.

CONCLUSION

The report shows that the product of the energy and the autocorrelation may be an important consideration while differentiating between PAF and normal signals.

MATLAB® CODE

```
clear;
clc;
close all;

fnames=dir('train/*.mat');
normal = zeros(50,2,38400);
arrhy = zeros(50,2,38400);

test_normal = zeros(5,2,38400);
test_arrhy = zeros(5,2,38400);

Fs = 128;
% number of sampling instances
N = 38400;
% frequency base
f = (-N/2:N/2-1)*Fs/N;
% time base
t = 0:1/Fs:5*60-1/Fs;

data_size = length(fnames); % no of samples (+ve and -ve)
val_index = 1:5;    % 5 fold cross validation
data_indices = 1:data_size;    % indices of names of files in database
normal_indices = 1:data_size/2;
abnormal_indices = data_size/2+1:data_size;

%% Butterworth lowpass filter
n = 4;
Wn =12/(Fs/2);
% Zero-Pole-Gain design
[z,p,k] = butter(n,Wn,'low');
sos = zp2sos(z,p,k);
```

```
precision = zeros(1,5);
recall = zeros(1,5);

for val_index = 1:5     % cross validation folds
    test_indices_normal = (val_index-1)*5+1:val_index*5;        % test
indices
    test_indices_abnormal = data_size/2 + (val_index-1)*5+1:data_size/2 +
val_index*5;
    ntest_indices_bool = ismember(normal_indices,test_indices_normal);
    antest_indices_bool = ismember(abnormal_indices,test_indices_abnormal);
    train_indices_normal = normal_indices(~ntest_indices_bool);    % train
indices
    train_indices_abnormal = abnormal_indices(~antest_indices_bool);

    index = 0;
    norm_energy =zeros(data_size/2-5,2,1);
    abnorm_energy = zeros(data_size/2-5,2,1);

    norm_autocorr = zeros(data_size/2-5,2,1);
    abnorm_autocorr = zeros(data_size/2-5,2,1);
    %% Training of normal instances
    for ntrain_index = train_indices_normal
        % Read the data and normalize it
        index = index+1;
        file=fullfile('train',fnames(ntrain_index).name);
        s = load(file);
        normal(index,:,:) = L2normalize(s);

        % Find the energy of the filtered signals
        % Find auto-correlations of original signals
        % Define the product(s) as features for each signal
        [norm_energy(index,:),norm_autocorr(index,:)] =
feature_extract(normal(index,:,:),sos);
    end
    index = 0;

    %% Training of abnormal samples
    for antrain_index = train_indices_abnormal
        index = index+1;
        % Read the data and normalize it
        file=fullfile('train',fnames(antrain_index).name);
        s = load(file);
        arrhy(index,:,:) = L2normalize(s);

        % Find the energy of the filtered signals
        % Find auto-correlations of original signals
        [abnorm_energy(index,:),abnorm_autocorr(index,:)] =
feature_extract(arrhy(index,:,:),sos);
    end

    %% Definition of thresholds
    norm_measure1 = (mean(norm_energy(:,1))*(mean(abs(norm_autocorr(:,1)))));
```

```
    norm_measure2 = (mean(norm_energy(:,2))*(mean(abs(norm_autocorr(:,2)))));
    abnorm_measure1 =
(mean(abnorm_energy(:,1))*mean(abs(abnorm_autocorr(:,1))));
    abnorm_measure2 =
(mean(abnorm_energy(:,2))*mean(abs(abnorm_autocorr(:,2))));

%     norm_measure1 = (mean(abs(norm_autocorr(1))));
%     norm_measure2 =(mean(abs(norm_autocorr(2))));
%     abnorm_measure1 =mean(abs(abnorm_autocorr(1)));
%     abnorm_measure2 =mean(abs(abnorm_autocorr(2)));
    fin_norm_thresh = (norm_measure1+norm_measure2)/2;
    fin_arrhy_thresh = (abnorm_measure1+abnorm_measure2)/2;

    %% -------------------------------- Validation ------------------------
- %%
    tp = 0;
    tn = 0;
    fp=0;
    fn=0;

    nindex = 0;     % index for normal test cases

    %% Testing for normal test cases
    for ntest_index=test_indices_normal
        file=fullfile('train',fnames(ntest_index).name);
        s = load(file);

        nindex = nindex+1;

        test_normal(nindex,:,:) = L2normalize(s);
        % Find energy of the signal and normalize

        [measure1,measure2]=feature_extract(test_normal(nindex,:,:),sos);
        m1 = measure1(1)*abs(measure2(1));
        m2 = measure1(2) * abs(measure2(2));
        norm_true1 = abs(m1-fin_norm_thresh)<abs(m1-fin_arrhy_thresh);
        norm_true2 = abs(m2-fin_norm_thresh)<abs(m2-fin_arrhy_thresh);
        if (norm_true1)
            tp=tp+1;
        else
            fn=fn+1;
        end
        if (norm_true2)
            tp=tp+1;
        else
            fn=fn+1;
        end
    end
    aindex = 0;     % index for abnormal test cases

    %% Testing for abnormal test cases
    for atest_index = test_indices_abnormal

        file=fullfile('train',fnames(atest_index).name);
        s = load(file);
```

```
        aindex = aindex+1;

        test_arrhy(aindex,:,:)=L2normalize(s);

        [measure1,measure2]=feature_extract(test_arrhy(aindex,:,:),sos);
        m1 = measure1(1)*abs(measure2(1));
        m2 = measure1(2)*abs(measure2(2));

        abnorm_true1 = abs(m1-fin_norm_thresh)>abs(m1-fin_arrhy_thresh);
        abnorm_true2 = abs(m2-fin_norm_thresh)>abs(m2-fin_arrhy_thresh);

        if (abnorm_true1)
            tn=tn+1;
        else
            fp=fp+1;
        end
        if (abnorm_true2)
            tn=tn+1;
        else
            fp=fp+1;
        end
    end
    precision(val_index) =tp/(tp+fn)*100;
    recall(val_index) =tn/(tn+fp)*100;

    fprintf('%f is the precision for val
round %d\n',precision(val_index),val_index);
    fprintf('%f is the recall for val
round %d\n',recall(val_index),val_index);


end


    fprintf('%f is the mean precision\n',mean(precision));
    fprintf('%f is the mean recall\n',mean(recall));
function [normal] = L2normalize(s)
normal=s.val;
% Find energy of the signal and normalize
energy1 = sum(normal(1,:).^2);
energy2 = sum(normal(2,:).^2);
normal(1,:)=normal(1,:)/sqrt(energy1);
normal(2,:)=normal(2,:)/sqrt(energy2);
end


function [energy,auto_corr] = feature_extract(signal,sos)

s1=signal(1,1,:);
s2=signal(1,2,:);

% filter the signals
y1 = sosfilt(sos,s1);
```

```
y2 = sosfilt(sos,s2);
energy(1) = sum(y1.^2);
energy(2) = sum(y2.^2);

% Auto-correlation of original signals
% 128 samples corresponds to 10s
auto_corr(1) = sum(s1(1,1,1:end-128).*s1(1,1,129:end));
auto_corr(2) = sum(s2(1,1,1:end-128).*s2(1,1,129:end));


end
```