ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 4602: Signals & Systems Lab

Lab – 04: Signal Processing

A signal carries some information. Signal processing (SP) refers to extracting knowledge from a signal through certain operations. It involves converting or transforming data in a way that allows us to see things in it that are not possible via direct observation. Different signal processing techniques allow us to analyze, modify, correct, or optimize signals. In this lab, you will learn about several SP techniques and how to utilize them in different applications.

4.1 Peak Analysis

In this section, we will try to answer the following questions –

- How to find the peaks in a signal?
- What are the global maxima/minima?
- How to measure the distance between peaks?
- How to measure the amplitude of peaks?
- How to capture the trend in a signal?

Let's start by loading a signal into MATLAB.

```
load("spots_num")
```

This will load two variables into MATLAB's workspace – avSpots and year. The data contains the average number of sunspots observed every year from 1749 to 2012. There are in total 264 records. Let's plot the data to visualize the pattern.

Prepared by
Asif Newaz
Lecturer, EEE, IUT

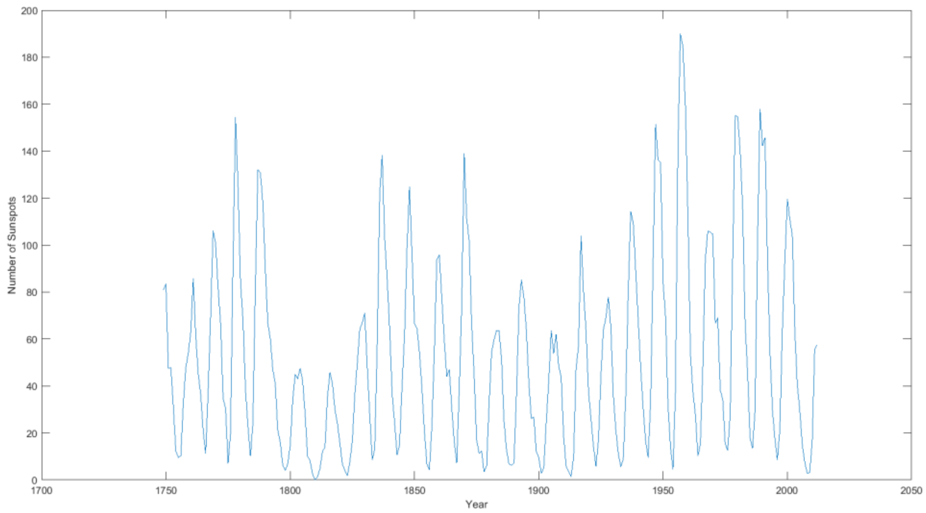

Fig. 1

Fig. 1 shows the avg number of sunspots observed every year. We can see a repeating pattern in the data. To understand the signal, let's start by identifying its peaks.

Finding Peaks

```
val= findpeaks(avSpots)
```

The "findpeaks" function returns a vector with the local maxima (peaks) of the input signal. A local peak is defined as a data sample that is larger than its two neighboring samples (check documentation). To also know the points (years) at which the peaks occur, use multiple output commands.

```
[val, loc] = findpeaks(avSpots)
```

Now, to mark the peaks in the signal, use the following code –

```
plot(year, avSpots, year(loc), val, "o" )
axis tight
```


Prepared by
Asif Newaz
Lecturer, EEE, IUT

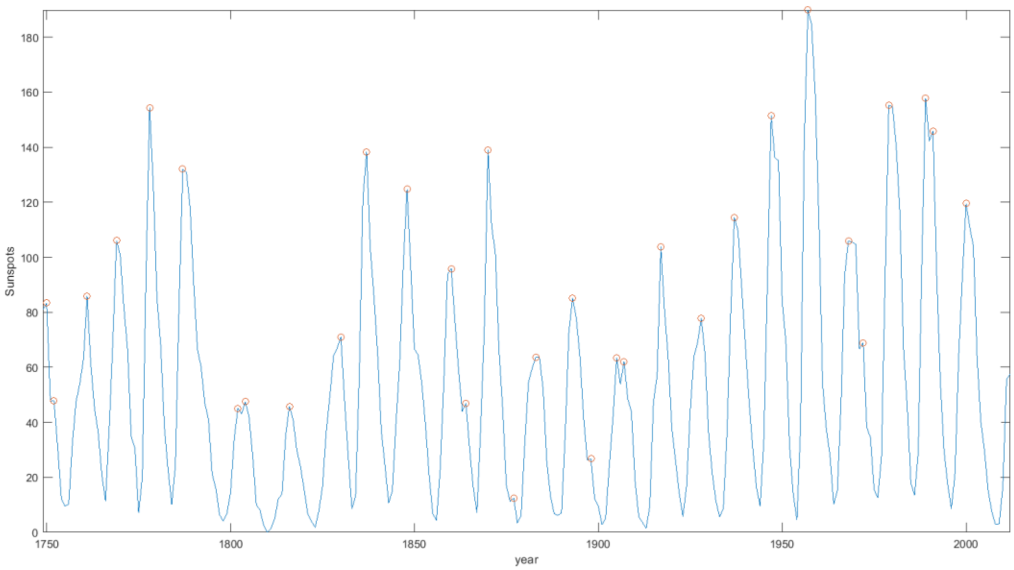

Fig. 2

Finding the Highest Peak

The global maxima and the year it occurred can be easily computed from the data.

```
[a, b] = max(avSpots)
year(b)
```

As we can see from the results, the highest peak (maximum number of sunspots observed) occurred in 1957.

Eliminating misidentified peaks

In Fig. 2, the local peaks are marked as a circle. If you look carefully, you will find some incorrectly predicted local maxima points (around year 1875 for instance). Moreover, there are some very closely placed local maxima points (around years 1810, and 1910 for instance). These points do not recur at regular intervals. So, they can be considered as noise.

To identify and remove such points, you can add an additional parameter to restrict the acceptable peak-to-peak separations. Use the following code for that –

```
[val2, loc2] = findpeaks(avSpots, "MinPeakDistance", 5)
```

Prepared by
Asif Newaz
Lecturer, EEE, IUT

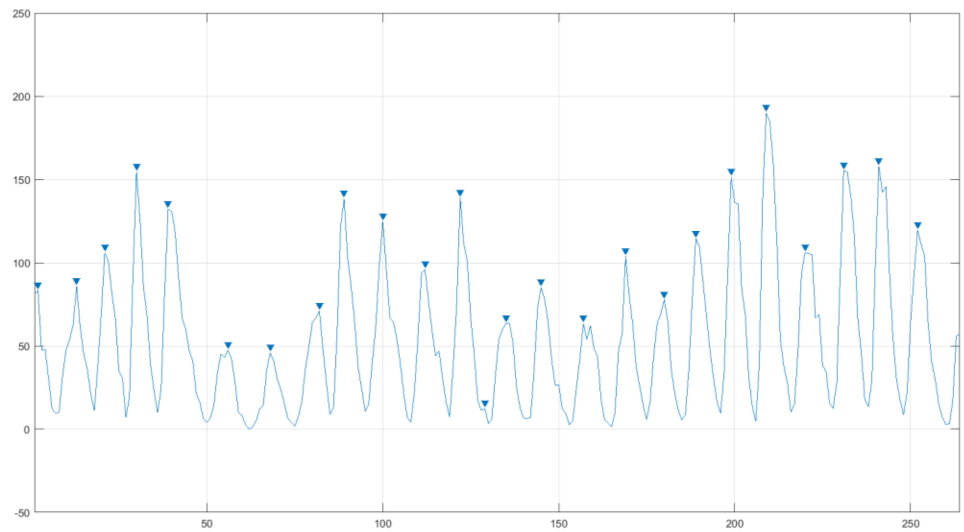

Fig. 3

Using peak-to-peak separation as 5 years, you get the output in Fig. 3. While some of the noisy points are removed, it still contains one misidentified optimum. So, what should be the appropriate value?

Use a ‘numeric slider’ to see the change in output and identify the most appropriate value. Go to “Live Editor >>> Control” to place the numeric slider in your live script. The final output would look like Fig. 4.

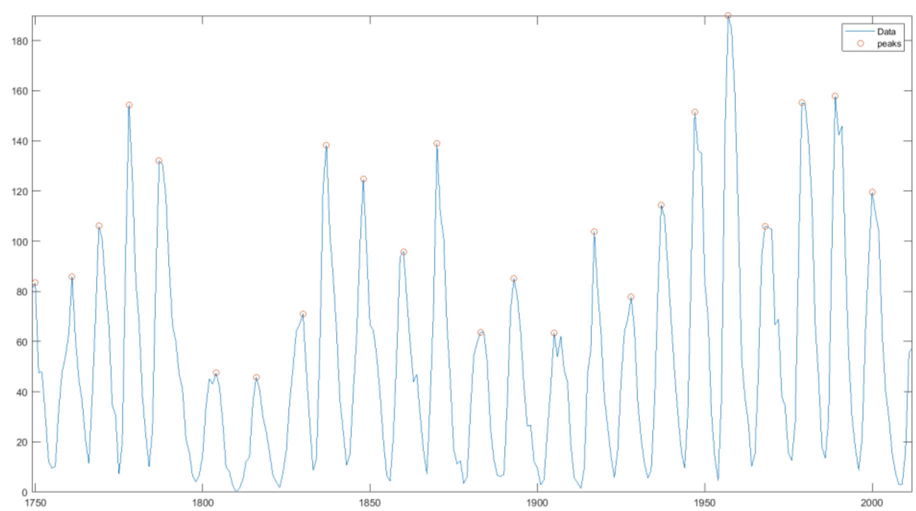

Fig. 4

Prepared by
Asif Newaz
Lecturer, EEE, IUT

Finding peak intervals

To find the average interval at which the peaks occur, use the following code –

```
Peak_distance = diff(loc)
Avg_peak_distance = mean (Peak_distance)
```

The first command gives you the distance (in year) between the peaks. The average is calculated as 10.8696 years. So, after roughly 11 years, the signal peaks are repeated.

Sunspots can change continuously and may last for only a few hours to days; or even months for the more intense groups. The total number of sunspots varies with an approximately 11-year repetition known as the solar cycle.

Lab Task – 01

I. Eliminate the peaks from the signal that are very closely placed using the 'MinPeakProminence' property. Check documentation to understand the parameter.

II. Find the peaks from the signal: clippedpeaks.mat

Finding Peaks in Saturated Signals

When a signal is saturated or clipped at the top, you may want to exclude the clipped portion. To do that, you need to use the 'threshold' property. It specifies a minimum excursion in the amplitude difference between a peak and its immediate neighbors.

```
load clippedpeaks.mat
findpeaks(saturatedData, 'threshold', ___ )
```

Use a numerical slider to find the appropriate threshold value for the signal. Figure 5 shows the results.

Prepared by
Asif Newaz
Lecturer, EEE, IUT

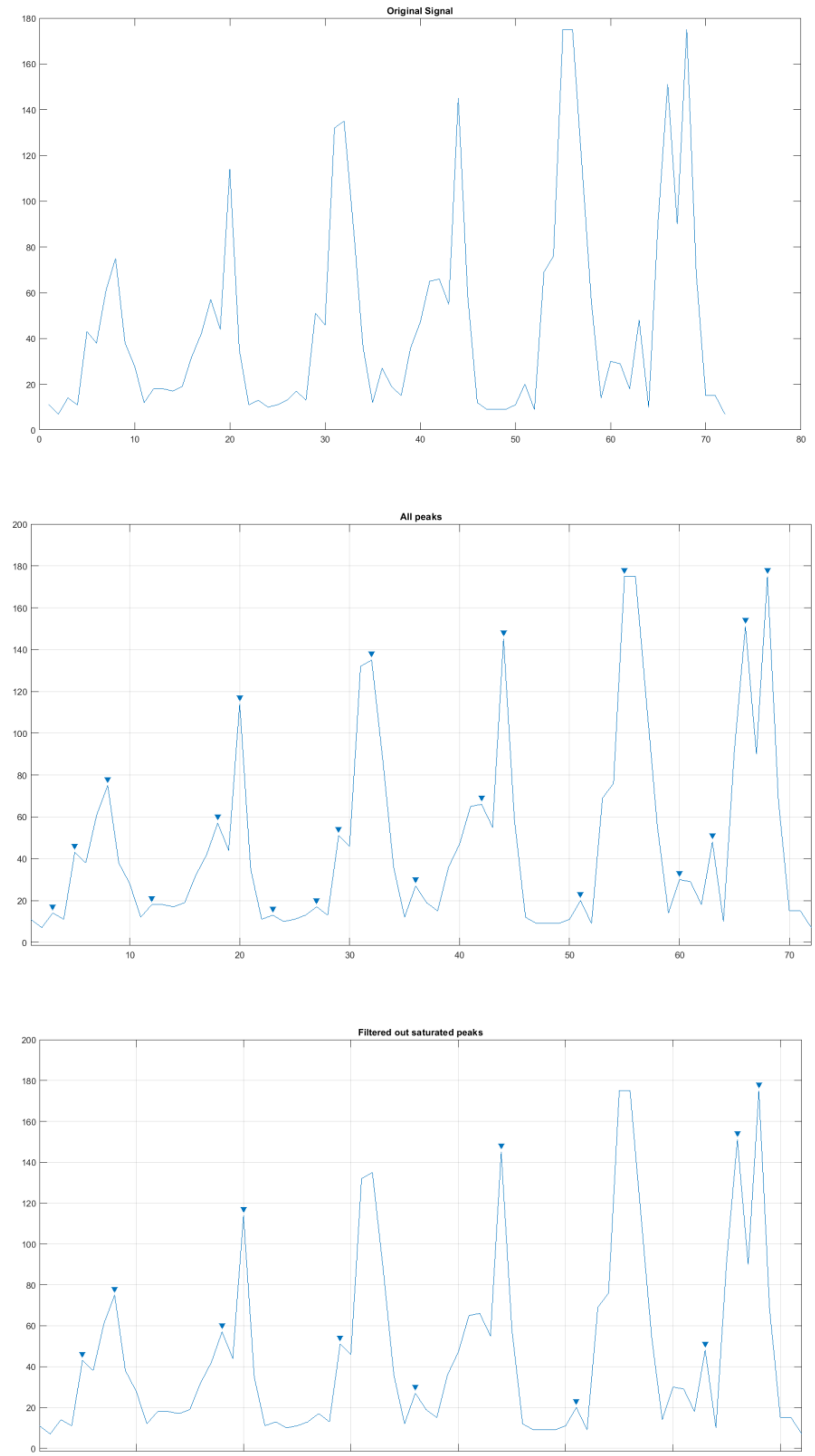

Fig. 5

Prepared by
Asif Newaz
Lecturer, EEE, IUT

Assignment

- Load “noisyecg.mat” signal in your workspace. Identify the peaks in the signal.
  Hint: try other parameters in the findpeaks function.

Code Files

All necessary files are available in this repository:

https://github.com/newaz-aa/Signal_Processing_Lab

Prepared by
Asif Newaz
Lecturer, EEE, IUT