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

4.3 Filter operation

You are familiar with linear filters such as low-pass filters or high-pass filters. In this section, we will look at a non-linear filter which is commonly used for removing noise from signals or images. The median filter is one such type of filter. While linear filters compute the weighted average of neighboring samples, the median filter replaces each sample with the median value of its neighboring samples within a specified window or kernel size.

Let's try to understand how the median filter works:

Window or kernel: Define a window or kernel size that moves over the signal or image. This window typically takes the form of a square or rectangular region.

Neighborhood: For each sample in the signal or image, collect the values of neighboring samples within the defined window.

Median computation: Compute the median value of the collected samples. The median value is the middle value when the samples are sorted in ascending or descending order. If the number of samples is even, the median is usually taken as the average of the two middle values.

Replace: Replace the original sample with the computed median value.

When n is odd, y(k) is the median of x(k-(n-1)/2:k+(n-1)/2).

When n is even, y(k) is the median of x(k-n/2:k+(n/2)-1).

Example: If n = 11, then y(k) is the median of x(k-5:k+5).

Example: If n = 12, then y(k) is the median of x(k-6:k+5).

Let's see the filtering operation by creating a couple of signals and then adding noise to it.

Prepared by
Asif Newaz
Lecturer, EEE, IUT

```
% generating sine wave

fs = 1000
t = 0:1/fs:1
x = sin(2*pi*t*3)
x2 = sin(2*pi*t*3) + 0.15*sin(2*pi*t*40)
plot (t, x, t, x2)
```

The medfilt1 function in MATLAB applies a 1D median filter to the input vector. The 2nd parameter of the function defines the order/kernel size of the filter.

```
y = medfilt1(x,10)
plot (t, x2, t, y)
legend ('Original', 'Filtered')
legend ('boxoff')
```

From the output, you can see the median filter removes any spikes from the original signal.

Let's look at another example from the previous section.

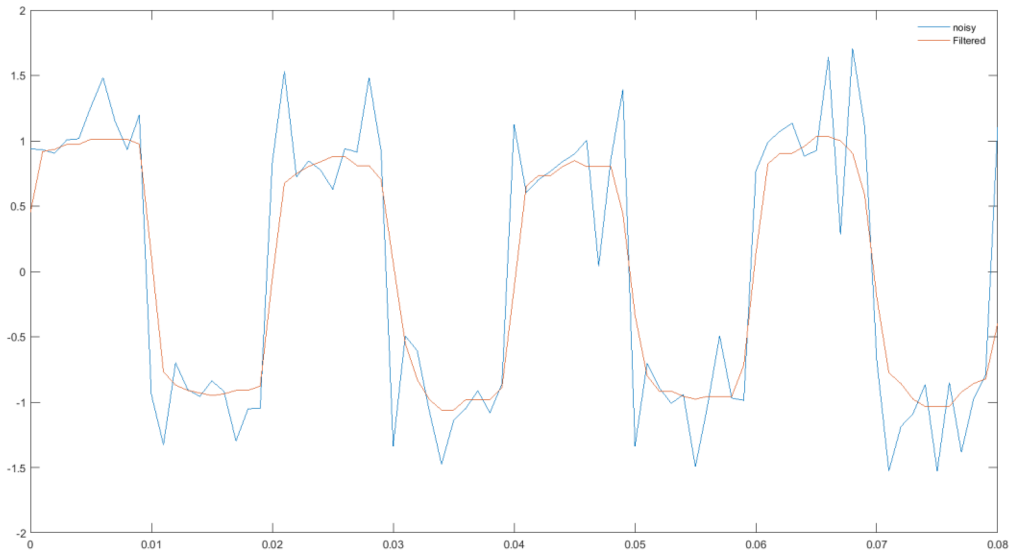
```
% adding white gaussian noise to square wave

x= 0:0.001:0.08
y= square(2*pi*50*x)
noise = awgn (y,10, 'measured')
plot (x, y, x, noise)
axis ([-0.01, 0.1, -2,2])
grid on
```

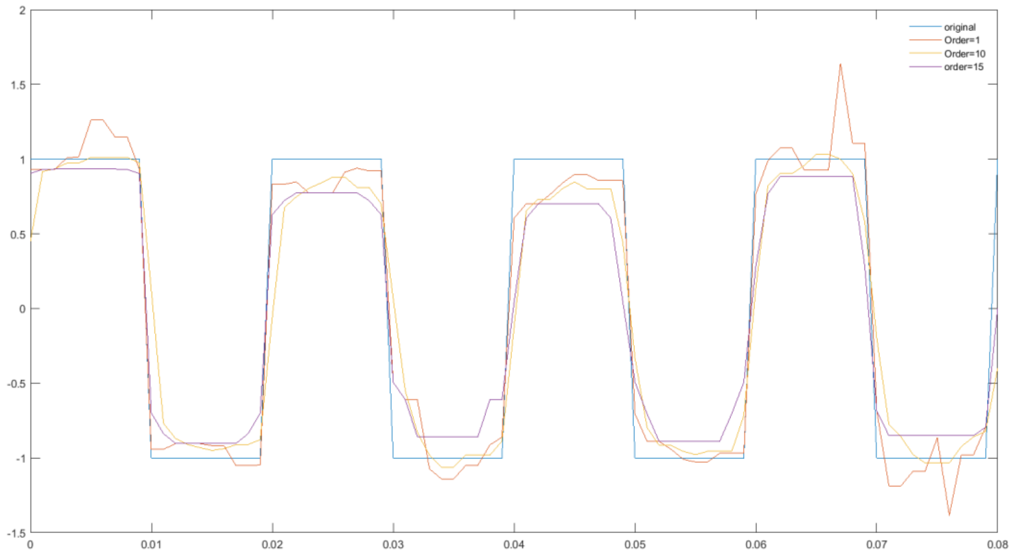
```
z= medfilt1(noise)
plot (x, noise, x, z)
legend ('noisy', 'Filtered')
legend ('boxoff')
```

Try to understand the effect of the order/window size of the filter on the output. Change the 2nd parameter to see how the output varies. Figure 1 demonstrates how the median filter reduces noise from a signal. Figure 2 displays the effect of changing the order of the filter. Too small a kernel won't be able to reduce the noise properly. On the other hand, too large a kernel would distort the original signal.

Prepared by
Asif Newaz
Lecturer, EEE, IUT



1. Fig 1



2. Fig 2

The median filter is effective at removing impulse noise or "salt and pepper" noise, where individual pixels or samples are corrupted with random spikes or dips in intensity. Since the median filter replaces each sample with the median value of its neighbors, it tends to preserve edges and fine details better than linear filters, making it particularly useful for images.

Prepared by
Asif Newaz
Lecturer, EEE, IUT

However, the median filter may not be as effective for removing other types of noise, such as Gaussian noise, which is better handled by linear filters like Gaussian or mean filters.

Hampel Filter

The Hampel filter is a generalization of the median filter that considers a window or kernel and compares each data point to the median of that window. If a data point deviates from the median by more than a certain threshold (determined by a scale parameter), it is replaced with the median.

In the MATLAB implementation of the 'Hampel' filter, the window is composed of the sample and its six surrounding samples, three per side. It also estimates the standard deviation of each sample about its window median using the median absolute deviation (MAD). If a sample differs from the median by more than three standard deviations, it is replaced with the median.

These filters are particularly useful for removing outliers in time series data. The median filter is usually used when dealing with impulse noise or when simplicity and edge preservation are crucial. The Hampel filter is utilized when dealing with time series data containing outliers of varying magnitudes, or when more robust outlier detection is required. So, the choice between them depends on the specific characteristics of the data and the noise being addressed.

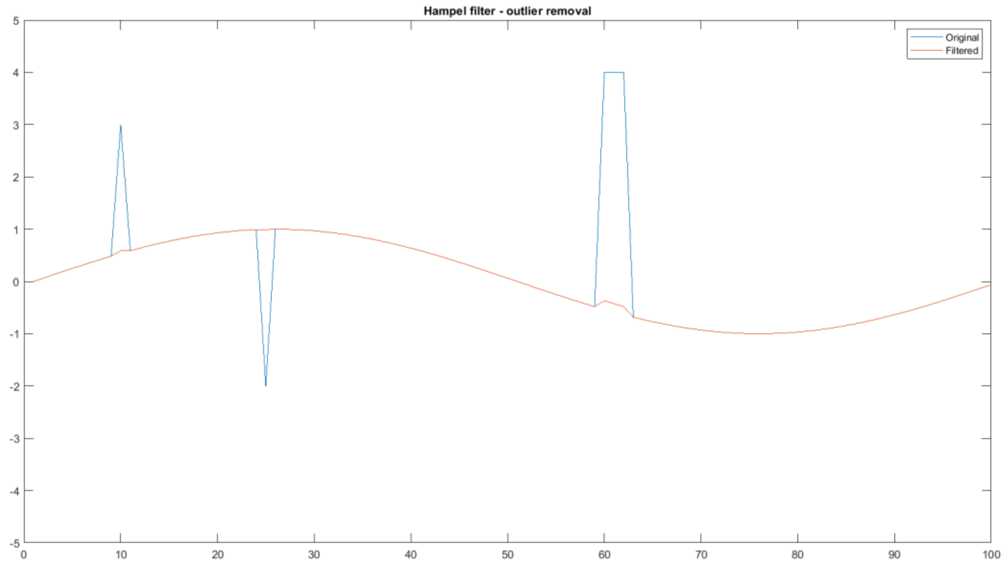
Let's add a few outliers/spikes in a sinusoidal wave.

```
x = sin(2*pi*(0:99)/100)
t = 1: length (x)
plot (t, x)
x (10) = 3
x (25) = -2
x (60:62) = 4
```

```
[y, i, xmedian, xsigma] = hampel(x)
% i locates the outlier points
```

See the output in Figure 3.

Prepared by
Asif Newaz
Lecturer, EEE, IUT



3. Fig 3

Code files

The associated live script file for this lab can be found here –

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

Prepared by
Asif Newaz
Lecturer, EEE, IUT