Spectral processing of COVID-19 data using digital filter designing Signal Processing - Assignment

P.A Dinushika Chithrani
Reg No: 19/ENG/009
Department of Electrical and Electronic engineering
Faculty of Engineering
University of Sri Jayewardenepura

***Abstract*—Since the beginning of 2020, COVID-19 was identified as the fourth leading cause of death globally, resulting for one in 20 deaths worldwide. To have a better understanding of the pandemic the presence of oscillations in total covid 19 data is processed with spectral analysis [1]. It is used to show additional properties of the data, and the oscillations are also observed using frequency analysis. Using the digital filter designing methods, it proceeds smoothing techniques and lowering trends. Various digital filter types are applied in this paper and the difference of the results are discussed. The behavior of the seven-day moving average is also discussed, especially the cause of its jaggedness and the phase error it introduces. These methods have applications that include modeling time-series data as well as identifying less obvious properties of the data.**

***Index Terms*—COVID 19, smoothing, moving average, oscillation, spectral analysis, filtering, frequency analysis.**

I. Introduction

CORONAVIRUS disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus. Since the start of 2020, based on official counts, COVID-19 has killed more than 6.7 million people, including more than 975,000 in the United States. There are many research papers on collecting data on COVID 19. For COVID-19, daily counts of both new cases and new deaths, many countries and regions show clear oscillations. Observing just a few days, the counts can experience large fluctuations. As an example, between July 6 and 7, the number of deaths in Arizona state in USA is varied from 1 to 117.

Some researchers have been proposed reasons to explain the oscillations. Where the oscillations can be caused by data acquisition practices, they concluded that the variations of data points can contain a large amount of error. Identifying the sources of error could give an understanding to more effective pandemic responses. Probably the reason could be an absence of any standard procedures for reporting and collecting COVID-19 data.

Previously many papers have looked at trend analysis and forecasting details. [2] [3] Those methods include alternative smoothing methods. This paper shows how to use the digital filtering techniques to improve data smoothing as well as the extraction of shorter-term fluctuations with the help of [4]. This is done with spectral analyzing and modelling techniques. In this study oscillations are attenuated rather than removed.

More people in the United States died of COVID-19 than in any other country. Although this is in part due to the large U.S. population, the country also has a high death rate due to COVID-19. As a source of aggregated COVID-19 data, the repository from [5] is used in this paper. This study has considered the daily deaths in U.S. due to COVID-19 from 1 March to 24 October, since the first death in the US was reported on 1 March 2020. The periodogram is used to observe the behavior of the pandemic. Instead of observing the number of deaths directly, the converted peak spectrum can simply indicate the strength and the tendency of the pandemic waves. In signal processing, a periodogram is an estimate of the spectral density of a signal.

This study proposes to perform reproducing time-series data oscillations, for the understanding of the weekly progression of the epidemic. Which gives the results to forecast minimums, maximums, in the new daily deaths and cases count. Therefore, it identifies varying data collecting practices, and it can detect irregularities in the time series data.

II. Methodology

As of this writing, only 238 days are considered in the COVID-19 time-series which are mathematically significant to this paper's analysis. For a signal processing work, it is an small number of a data count.

A. Deriving the Periodogram of the time series data

Before processing spectral analysis is performed in this study to get a better understanding about the oscillation. In signal processing, a periodogram is an estimate of the spectral density of a signal. For time series data, periodogram can be viewed as a sum of sine and cosine waves with varying amplitudes and frequencies. That is useful to observe important frequencies (or periods) in the time series. The periodogram function is the starting tool for doing this. The count of daily deaths in U.S. due to COVID-19 among 238 days is considered for the analysis. The time series signal is of length N = 238 data points taken daily for 238 days. So formally the sampling rate is Fs = 1 Hz.

1) Matlab design methodology: This section describes the MATLAB code for implementing the periodogram for the number of daily deaths in US and world. The data is included in the world.csv file from github.

```
%reading the data example:
%Import columns as column vectors
Array = readmatrix('world.csv');

time = Array(:, 1); %number of days(238)
deaths = Array(:, 2);%number of deaths

fs =1; % one measurement per day

deathsN = deaths− mean(deaths);
[pxx,f] = periodogram(deathsN,[],[],fs);

%Original daily deaths
figure(1)
plot(time, deaths, 'Linewidth', 1);
xlim([0 238]);
title('New daily deaths in the U.S.')
xlabel('time (days)')
ylabel('Daily new deaths')
grid on;

%Periodogram for daily deaths
figure(2)
plot(f,pxx,'Linewidth', 1)
title('Spectogram of the daily counts of
    deaths in US')
xlabel('1/Days')
ylabel('Magnitude')
grid on;
```

2) Matlab output: Figure 1 shows the number of daily deaths in the US where '0' on the X-axis represents 1 March 2020, while '237' on the X-axis represents 24 October 2020. Figure 2 shows the result of spectrum analysis using 238 days.

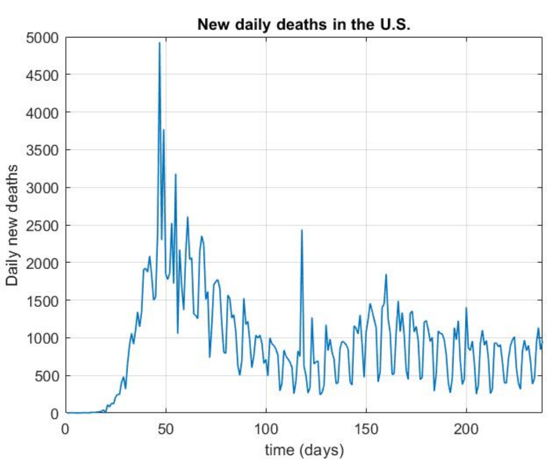

Fig. 1: Number of daily deaths in the US

The behavior of COVID-19 daily death count of US data can be better understood and better characterized by altering

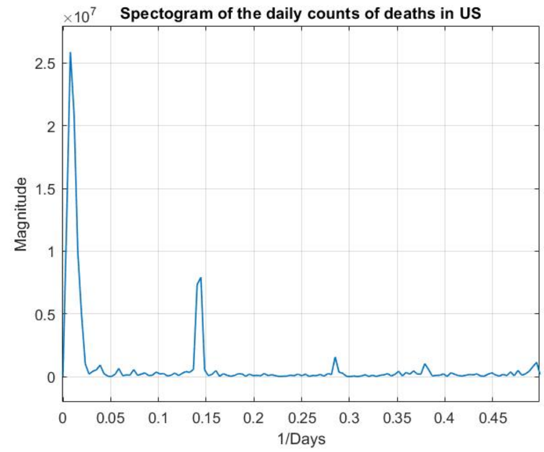

Fig. 2: Result of spectrum analysis using Periodogram

its spectral properties. [6] In this study, two methods were used for modifying the spectral content of U.S. time-series data.

1) The moving average filtering
2) Infinite impulse response (IIR) filtering

B. Seven day moving average filter

In the context of digital signal processing, the seven-day moving average is a finite impulse response (FIR) filter. Simply, the method of moving average filters is used to smooth the time series data and reveal underlying trends. Smoothing is the technique of reducing random changes in a plot of raw time series data.

The formula for a simple moving average is:

$$y'_t = \frac{y_t + y_{t-1} + ... + y_{t-n-1}}{n}$$

where y is the variable (count of new daily deaths in U.S.), t is the current time period (the current day), and n is the number of time periods in the average. In most research's, 3, 4, 5 moving averages were used (so that n = 3, 4 or 5), since larger the n, the series becomes smoother. In this case, n = 7, as seven day moving average is considered.

The data set is imported from [5]. To each data point, the filter is applied by constructing the coefficients of the filter. Such that each data point is equally weighted and contributes 1/7 to the total average. This gives us the average death over each 7 day.

MATLAB gives a method to perform a moving average filter. [7] The MATLAB code to obtain the seven day moving average filter is listed below.

```
Array = readmatrix('world.csv'); %import data
time = Array(:, 1);
deaths = Array(:, 2);

B = 1/7*ones(7,1);
movfilt = filter(B,1,deaths); %apply filter
```

The frequency response for the seven-day moving average is shown in Fig 3 It is obtained by fvtool( ) function.

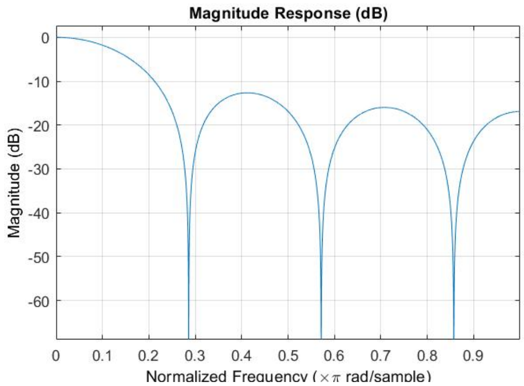

Fig. 3: The frequency response for the moving average filter

C. IIR filters

This section describes the main part of this study, which describes the performance of 4 types of digital filter designs.

The basic idea in IIR filters is that the output continues infinitely, in time even if the input time series is finite in length. It happens due the feedback part. The main advantage of IIR filters over FIR filters is that they offer high precision of frequency difference using fewer data points in the averaging.

Elliptic filters are a type of IIR filter. This filter type is characterized by the ripple in both pass-band and stop-band and also with the fastest transition between pass-band and ultimate roll-off. During the design, the levels of ripple in the pass-band and stop-band can be adjusted independently.

Elliptic type filters were chosen in this study because their steep frequency roll-off allows for strong frequency isolation. Signal processing filters frequently exhibit frequency-dependent phase alterations. However, if filtering is applied twice in opposing directions, these artifacts are eliminated. The signal processing toolbox in MATLAB was used to generate four elliptic filters. [8] To remove the phase difference, the data-set was filtered twice: first backwards, then forwards.

D. Filter performance

A filter decomposes a time series into three main components.Which are trend, cycle, and error components.

Mathematically if y is a time series then

$$y_t = T_t + C_t + e_t$$

Where 'T' is the trend component, 'C' is the cyclic component and 'e' is the error component.

To filter-out each components, by observing the frequency distribution in figure 2 the design parameters are were selected according to table I.

To compare the performance of each type, all 4 types of filters were used in this study. Filters were designed by MATLAB using the signal processing toolbox functions. After designing each filter, the frequency response plots for each filter was plot using fvtool() function. All plots are shown in the figures 4 to 8.

The following MATLAB code was used to design the filters.

1) Low pass filter: -

```
Array = readmatrix('world.csv');
time = Array(:, 1);  %number of days(238)
deaths = Array(:, 2);  %number of deaths

Fs = 1;  % Sampling Frequency
lpFilt = designfilt('lowpassiir',...
     'PassbandFrequency',1/9, ...
     'StopbandFrequency',1/8,...
     'PassbandRipple',0.01, ...
     'StopbandAttenuation',40,...
     'DesignMethod','ellip','SampleRate',Fs);

%Filtering twice
Filtered_deaths = filtfilt(lpFilt, deaths);
```

2) High pass filter: -

```
Array = readmatrix('world.csv');
time = Array(:, 1);  %number of days(238)
deaths = Array(:, 2); %number of deaths

Fs = 1;  % Sampling Frequency
hpFilt = designfilt('highpassiir',...
    'PassbandFrequency',1/7, ...
    'StopbandFrequency',1/8,...
    'PassbandRipple',0.01, ...
    'StopbandAttenuation',40,...
    'DesignMethod','ellip','SampleRate',Fs);

%Filtering twice
Filtered_deaths = filtfilt(hpFilt, deaths);
```

3) Band pass filter: -

```
Array = readmatrix('world.csv');
time = Array(:, 1);  %number of days(238)
deaths = Array(:, 2); %number of deaths

Fs = 1;  % Sampling Frequency
bpFilt = designfilt('bandpassiir', '
    StopbandFrequency1', 1/9, ...
    'PassbandFrequency1', 1/8, ...
    'PassbandFrequency2', 1/6, ...
    'StopbandFrequency2', 1/5, ...
    'StopbandAttenuation1', 40, ...
    'PassbandRipple', 0.01, ...
    'StopbandAttenuation2', 40, ...
    'SampleRate', Fs, ...
    'DesignMethod', 'ellip');

Filtered_deaths = filtfilt(bpFilt, deaths);
```

The same MATLAB code in 1) and 3) was used for low pass filter type 2 and band pass filter type 2 respectively by changing the parameters.

TABLE I: PARAMETERS FOR ELLIPTIC FILTERS

filter name	pass-band frequency	stop-band frequency	pass-band ripple	stop-band attenuation
low-pass 1	1/9	1/8	0.01 dB	40 dB
low-pass 2	1/21	1/19	0.01 dB	40 dB
high-pass 1	1/7	1/8	0.01 dB	40 dB
band-pass 1	1/8 1/6	1/9 1/5	0.01 dB	40 dB
band-pass 2	1/19 1/9	1/21 1/8	0.01 dB	40 dB

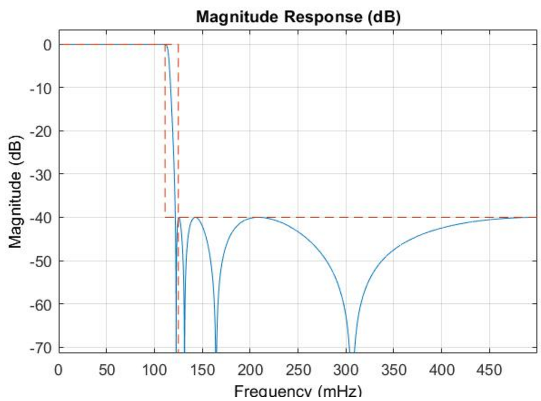

Fig. 4: The frequency response for the low pass 1 filter

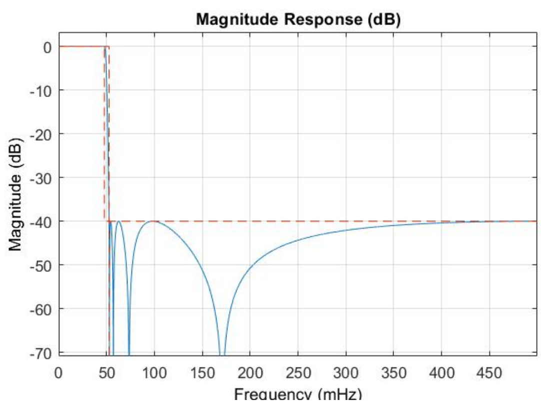

Fig. 5: The frequency response for the low pass 2 filter

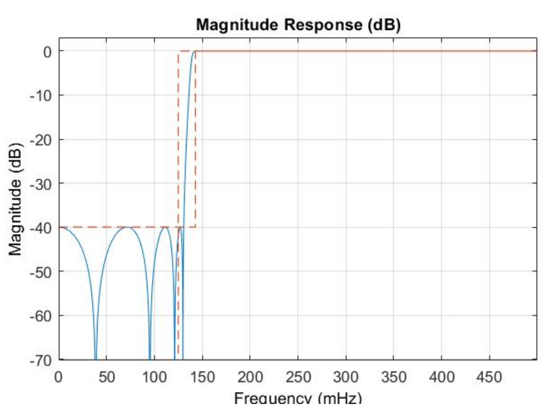

Fig. 6: The frequency response for the high pass filter

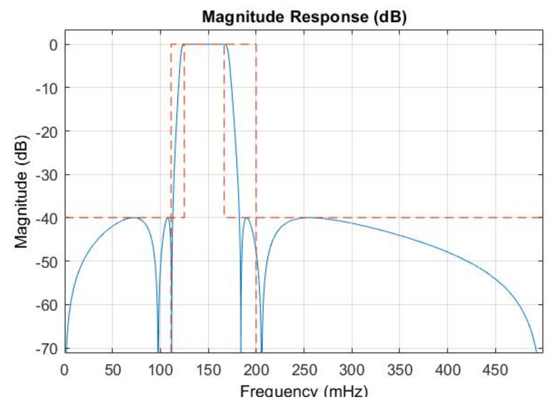

Fig. 7: The frequency response for the low pass 1 filter

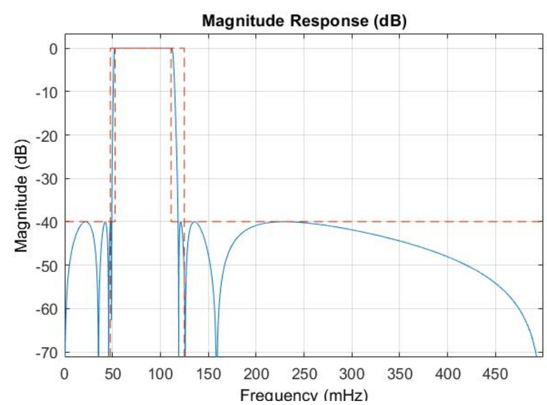

Fig. 8: The frequency response for the low pass 1 filter

III. Results and discussion

Using the above mentioned filters, the time series data of daily deaths in U.S was smoothed and extracted some fluctuations. The output was plot using the below MATLAB code.

```
% Compare outputs
plot(time, deaths,'LineWidth',1);
hold on;
title('New daily deaths in U.S. before and
    after processing')
plot(time, Filtered_deaths, 'LineWidth',1)
xlabel('time — days')
ylabel('Daily new death count')
xlim([0 238]);
grid on;
hold off;
legend('Daily cases','Filtered daily deaths');
```

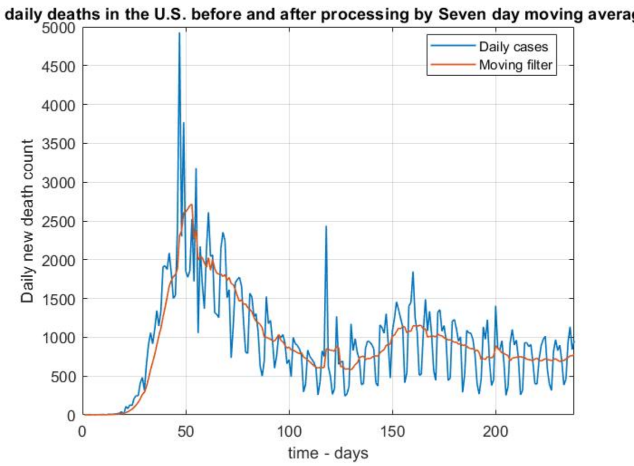

Fig. 9: New daily deaths in the U.S. before and after processing- Moving average filter

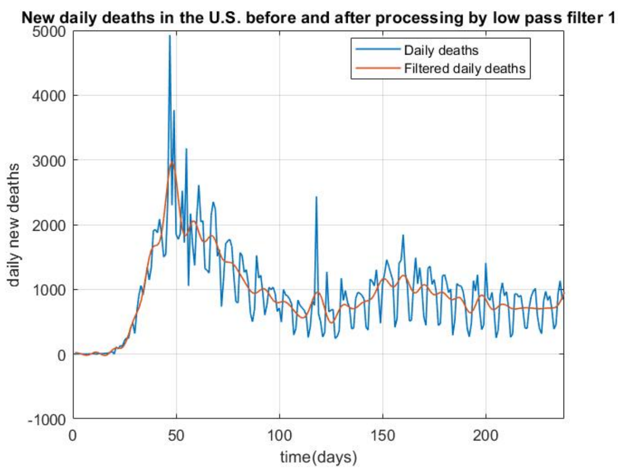

Fig. 10: New daily deaths in the U.S. before and after processing- Low-pass filter 1

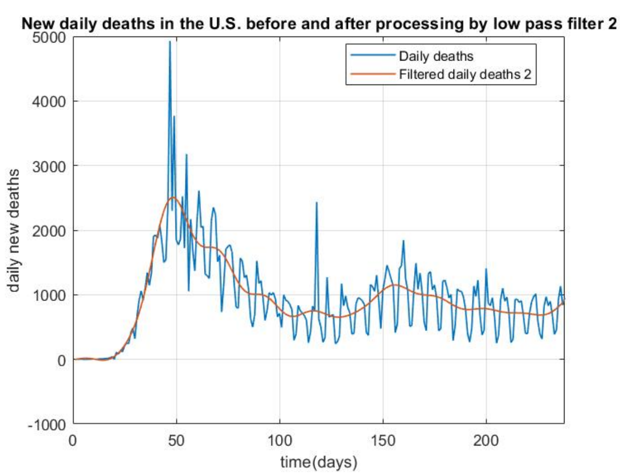

Fig. 11: New daily deaths in the U.S. before and after processing- Low-pass filter 2

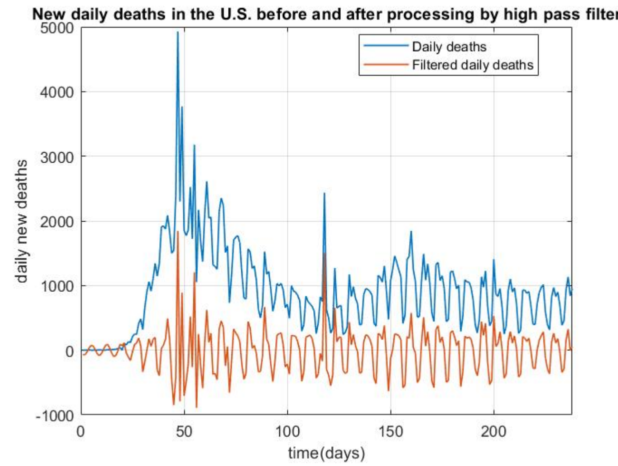

Fig. 12: New daily deaths in the U.S. before and after processing- High pass filter

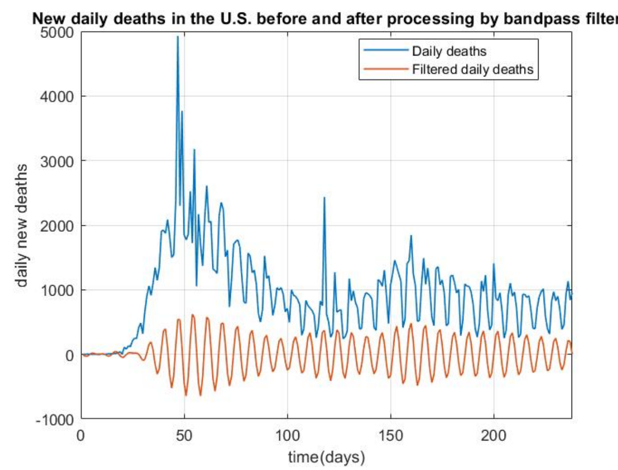

Fig. 13: New daily deaths in the U.S. before and after processing- Band-pass filter 1

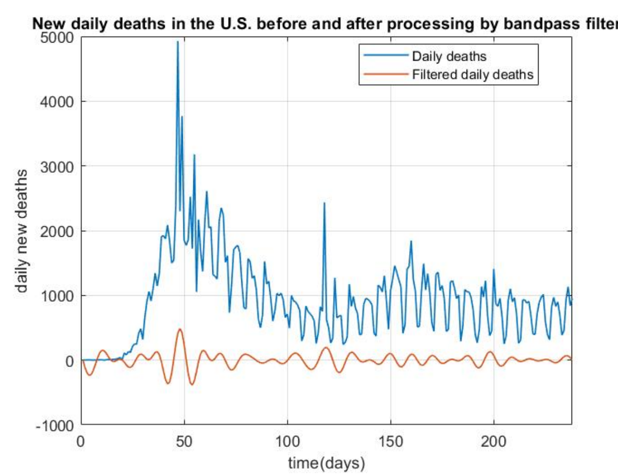

Fig. 14: New daily deaths in the U.S. before and after processing- Band-pass filter 2

.

A. Moving average filter: (Fig 9)

This is simply a low pass filter. The Fig 9 shows the one-sided moving averages with a length of 7 days for the time series data. Consider that how the seasonal pattern of the US daily death count data is flattened, and the underlying trend is visible. The daily average of the past seven days is represented by each moving average point.

B. Low-Pass Filter: (Fig 10 and Fig 11)

In this case, high-frequency values are attenuated by low-pass filtering in frequency domain. With the presence of the low pass filter type 1 and 2 the data plots visually more smoothed. Also, it gives more weight on longer-term trends than on fine details. Compared to the moving average line, in both low pass filter variations it is visibly less spiky. When comparing both type 1 and type 2 low pass filters, there are fewer lower-frequency oscillations in type 2 due to the small value of band pass and stop band frequencies.

C. High-Pass Filter: (Fig 12)

As shown in the fig 1 the signal drifts are slowly rising and falling, and after applying the high pass filter these drifts are removed. As the name suggests, the filter lets high frequencies pass, but removes low frequencies, which are the signal drifts or trends in time domain. Therefore, as shown high passing removes long term trends effectively.

D. Band-Pass Filter: (Fig 13 and Fig 14)

A filter which allows it to pass frequencies within a pre-defined range and rejects the frequencies going out of the range are known as band pass filters. The band pass filter produces a new series that does not contain fluctuations more than the certain frequencies. As shown, in type 1 band pass filtering isolates the oscillations between 5 and 9 days. In type 2 band pass filter, it isolates oscillations with periods between 8 and 21 days according to the predefined ranges. The resultant wave-forms are much more sinusoidal.

IV. Conclusion

Significant spectral differences were observed among U.S daily death count. These differences are frequently seen in time-series data. But there isn't a biological explanation, this must be the result of a huge difference in the way data is collected, reported, and aggregated.

Those filters were implemented in spectral processing to remove larger fluctuations in the time series, which can be caused by unused changes in any time series, or to smooth the time series since any forecasting model can accurately estimate or forecast values. It was concluded that high trends can be deleted to create a nicer-looking signal for visual inspection or publication, especially to observe long-term behavior over short-term oscillations.

Smoothing techniques reduce data series volatility, allowing analysts to identify major varying trends. Although the moving average technique is a straightforward way to smooth data, it may distort recent trends because it uses data from previous time periods. Smoothing techniques are time series forecasting methods that apply the weighted average of previous observations to forecast future values or predict new values. These methods work effectively with time-series data that has fewer variances over time.

References

[1] "timeseriesbook/spectral-analysis - search," https://bookdown.org/rdpeng/timeseriesbook/spectral-analysis.html.

[2] Y. Takefuji, "Fourier analysis using the number of covid-19 daily deaths in the us," *Epidemiology & Infection*, vol. 149, 2021.

[3] A. Bergman, Y. Sella, P. Agre, and A. Casadevall, "Oscillations in us covid-19 incidence and mortality data reflect diagnostic and reporting factors," *Msystems*, vol. 5, no. 4, pp. e00 544–20, 2020.

[4] S. McGovern, "Spectral processing of covid-19 time-series data," *arXiv preprint arXiv:2008.08039*, 2020.

[5] "https://github.com/ytakefuji/fourier/blob/main/d.csv - search," https://github.com/ytakefuji/fourier/blob/main/d.csv.

[6] "Spectram analysis of various time series data - search," http://web.stanford.edu/class/earthsys214/notes/series.html#dengue-example.

[7] "Signal smoothing - matlab & simulink example - mathworks india," https://in.mathworks.com/help/signal/ug/signal-smoothing.html.

[8] K. Talkie, "Matlab beginners tutorial low pass filter design in matlab - search," https://youtu.be/aHo1tWHVw9o.