Discrete-Time Systems & Their Difference Equations

Arnav Goyal - 251244778

February 8, 2024

Algorithms

Here are the three methods (algortihms) presented in the lab report

$$v_{avg}(n) = \frac{1}{4}\left[v(n) + v(n-1) + v(n-2) + v(n-3)\right] \tag{1}$$

$$v_{avg}(n) = 0.3 \cdot v(n) + 0.7 \cdot v_{avg}(n-1) \tag{2}$$

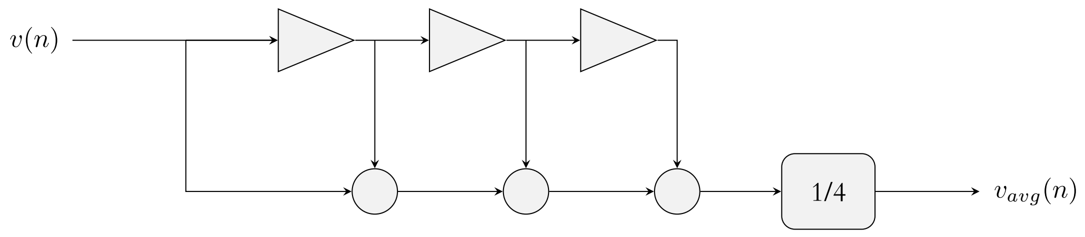
$$v_{avg}(n) = \frac{1}{4}\left[v(n) - v(n-4)\right] + v_{avg}(n-1) \tag{3}$$

Block Diagrams

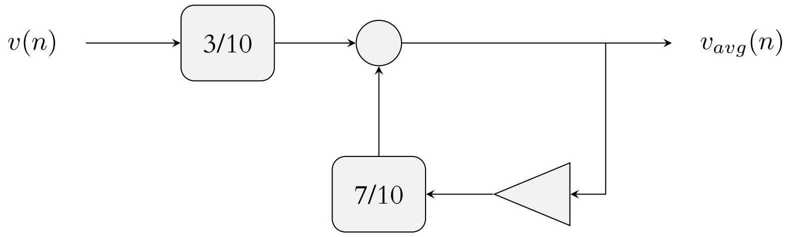
The block diagrams drawn below follow a pretty simple format:

- Rectangular blocks represent a multiplication (transfer function)
- Triangular shapes represent a time delay of one unit (unit delay)
- Circular nodes represent a summing junction
- They also look really nice (LaTeX moment)

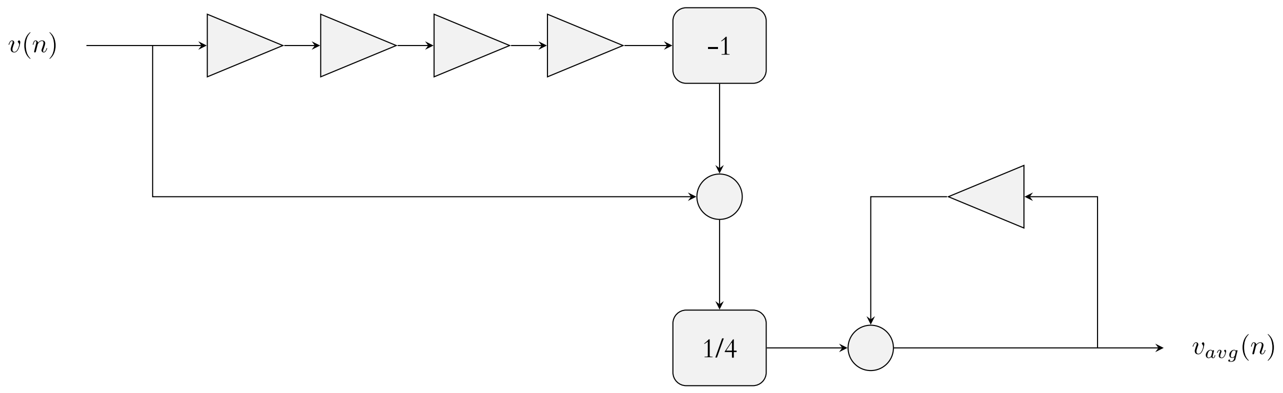
The block diagram for (1):



The block diagram for (2):



The block diagram for (3):



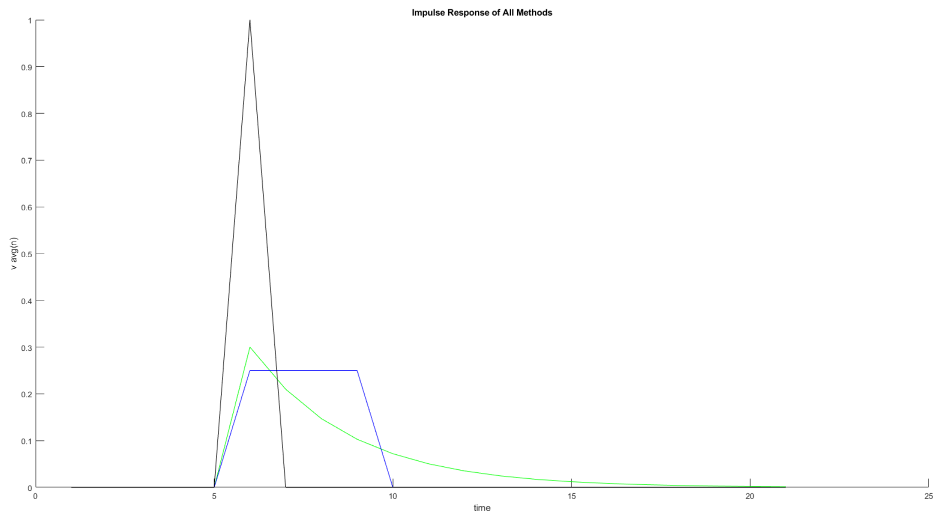
Impulse Response

To calculate impulse response, the following `MATLAB` script was created.

```
input = [0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
% any impulses

method1 = zeros(size(input));
method2 = zeros(size(input));
method3 = zeros(size(input));

% method 1
for i = 4:numel(input)
    method1(i) = mean(input(i-3:i));
end

% method 2
for i = 2:numel(input)
    method2(i) = (0.3 * input(i)) + (0.7*method2(i-1));
end

% method 3
for i = 5:numel(input)
    method3(i) = (0.25) * (input(i) - input(i-4)) + method3(i-1);
end

% plotting
figure;
hold on;
plot(method1, 'r');
plot(method2, 'g');
plot(method3, 'b');
plot(input, 'black');
title('Impulse Response of All Methods')
xlabel('time')
ylabel('v avg(n)')
```

This code provides the impulse response of (1), (2), and (3) plotted in red, green, and blue respectively. Additionally, the unit impulse was also plotted in black.

Note: The red plot of (1) and blue plot of (3) overlap completely, thus the figure only shows the blue as it was plotted last.



Comparison of Algorithms

Lets start by analytically finding the impulse responses of (1) and (3), To find the impulse response, we can essentially treat $v(n) = \delta(n)$, where $\delta(n)$ is the dirac delta function. Performing this substitution we get (4) and (5) for the impulse responses of (1) and (3) respectively . . .

$$v_{avg}(n) = \frac{1}{4}\left[\delta(n) + \delta(n-1) + \delta(n-2) + \delta(n-3)\right] \tag{4}$$

$$v_{avg}(n) = \frac{1}{4}\left[\delta(n) - \delta(n-4)\right] + v_{avg}(n-1) \tag{5}$$

The impulse response of (1), shown in (4), suggests that the dirac delta function is essentially *spread out* over 4 time units, this means that essentially the dirac delta will appear in one out of every four terms for 4 time periods, this leads to the step of 0.25 seen on the impulse response.

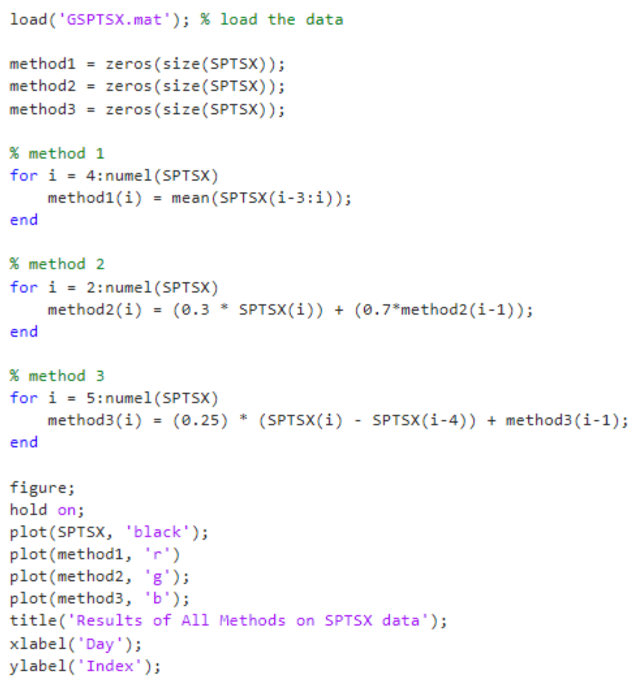
The impulse response of (3), shown in (5), suggests that the impulse response would be different due to the formula being different, however if we unbox the recursion we can start to see the similarities. Lets assume that the impulse appears at $t = 0$, as well as 0 initial conditions.

$$\begin{aligned}
v_{avg}(0) &= \frac{1}{4}(1-0)+0 &&= 0.25\\
v_{avg}(1) &= \frac{1}{4}(0-0)+v_{avg}(0) &&= 0.25\\
v_{avg}(2) &= \frac{1}{4}(0-0)+v_{avg}(1) &&= 0.25\\
v_{avg}(3) &= \frac{1}{4}(0-0)+v_{avg}(2) &&= 0.25\\
v_{avg}(4) &= \frac{1}{4}(0-1)+v_{avg}(3) &&= 0.00
\end{aligned}$$

After unboxing the recursion, we can see that given 0 initial conditions, both (1) and (3) display the exact same finite impulse response. This essentially means that they process the signal the same way. (1) is a simple moving average filter, and (3) is some sort of (subtractive?) average with a recursion/feedback term that behaves the same. This similarity in processing can be seen in the included plot in the next section.

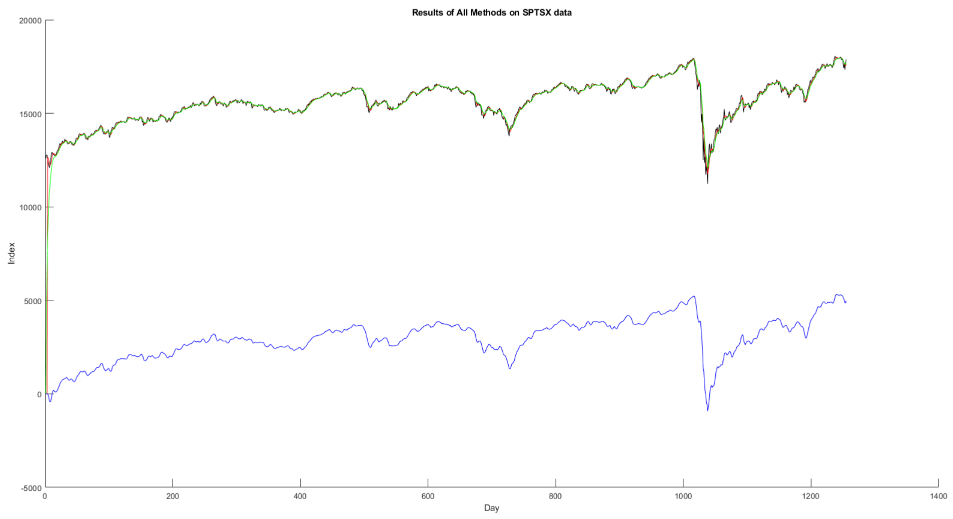
Data Processing

In order to process the data according to (1), (2), and (3), the following `MATLAB` script was developed



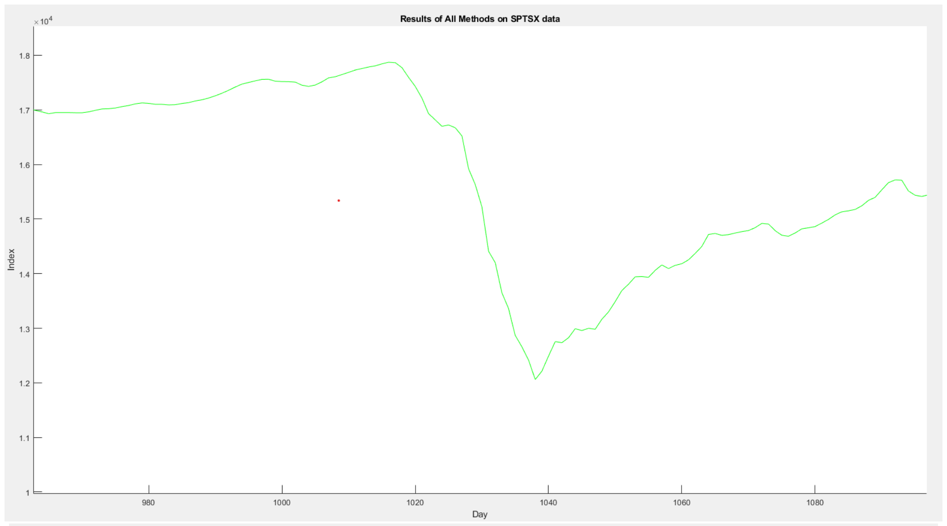
```
load('GSPTSX.mat'); % load the data

method1 = zeros(size(SPTSX));
method2 = zeros(size(SPTSX));
method3 = zeros(size(SPTSX));

% method 1
for i = 4:numel(SPTSX)
    method1(i) = mean(SPTSX(i-3:i));
end

% method 2
for i = 2:numel(SPTSX)
    method2(i) = (0.3 * SPTSX(i)) + (0.7*method2(i-1));
end

% method 3
for i = 5:numel(SPTSX)
    method3(i) = (0.25) * (SPTSX(i) - SPTSX(i-4)) + method3(i-1);
end

figure;
hold on;
plot(SPTSX, 'black');
plot(method1, 'r')
plot(method2, 'g');
plot(method3, 'b');
title('Results of All Methods on SPTSX data');
xlabel('Day');
ylabel('Index');
```

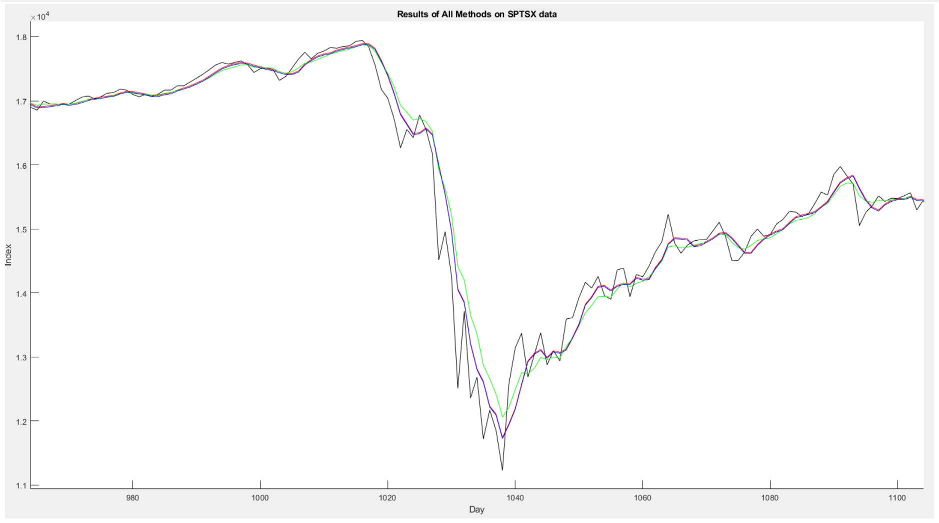
This script outputs a plot of the original data in black, and the processed data with algorithms (1), (2), and (3) plotted in red, green, and blue respectively.



From a quick look at the graph, we can easily see that the green trace seems to be smoother than the other extremely sharp traces, thus I think that the algorithm in (2) worked the best at smoothing out the data. I have provided a more zoomed in picture contrasting both plots on the next page, to help you see how much smoother the green trace is than the other ones. I think this makes sense because when an impulse (daily fluctuation) appears in (2) its impulse response suggests a smoothed out (diminishing) response which is smoother than the step on and off from (1) and (3).

Note: In the below images the blue trace has been shifted up by 12650 y-units to appear alongside the other traces. Also note the equivalency in fluctuations of the red and blue traces.


Results of All Methods on SPTSX data
×10⁴
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
Index
980
1000
1020
1040
1060
1080
Day


Results of All Methods on SPTSX data
×10⁴
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
Index
980
1000
1020
1040
1060
1080
1100
Day