POWER LINE INTERFERENCE CANCELLING ON REAL TIME MEASURING DATA

EMRE SENGİR

ABSTRACT

This report is about developing a pseudo-code, deictic demonstration of filtering assumed real time biomedical resistance measurement data by using cyclic voltammetry on, corrupted by the power line interference. Although we say real time measured data, we approached and developed the prototype of the algorithm on MATLAB due to the its simple syntax structure and easy access on observable data as graphs that is derived during the process. Lastly this report focuses only on the successful and approved algorithm by the supervisor engineer.

1) EXPECTED FORM OF INCOMING DATA

From the excel file that is given, we see the certain characteristics of completely measured data. Here is some sample values from excel file of experiment,

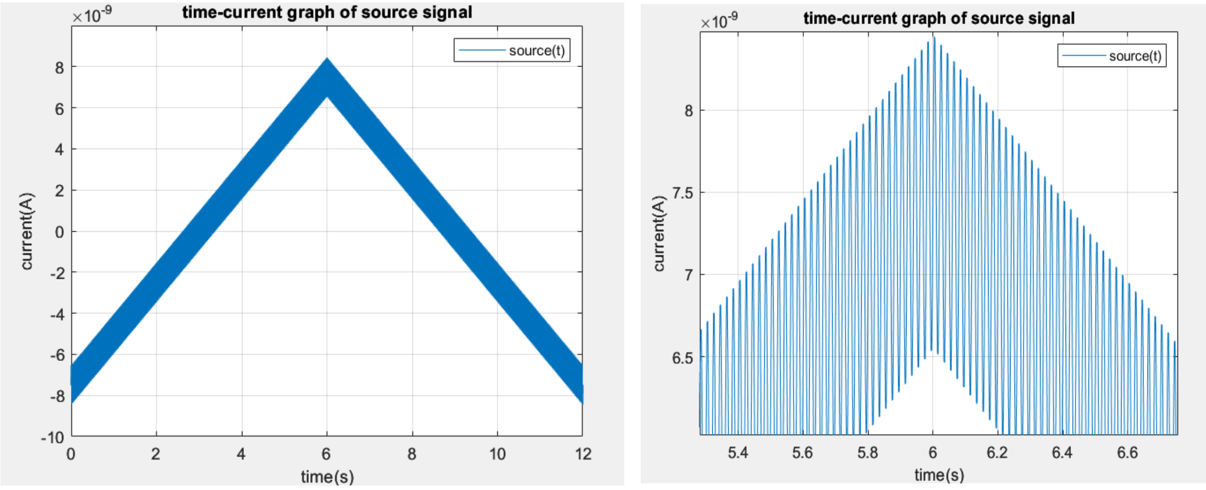
Applied			Measurement		
time(s)	Volt[mV]		time(s)	Volt[mV]	Ampere[μA
0	-1500		0	-1500	-0.15704
0.002	-1499		0.002	-1499	-0.08217
0.004	-1498		0.004	-1498	-0.08581
0.006	-1497		0.006	-1497	-0.07523
0.008	-1496		0.008	-1496	-0.06431

….

11.99	-1495		11.99	-1495	-0.07622
11.992	-1496		11.992	-1496	-0.07919
11.994	-1497		11.994	-1497	-0.08382
11.996	-1498		11.996	-1498	-0.07787
11.998	-1499		11.998	-1499	-0.07258

Table 1: Excel data of experiment

From table 1, one can determine that time interval between data points is equal to dt = 0.002, which makes sampling frequency $f_s = 500$, for duration d = 12 seconds resulting in having 6001 data points for voltage values cycling in mV from -1500 to +1500 and back, for resistance value of 220 Mohm.



(a) (b)

Figure 1: Source signal, (a) whole signal, (b) zoomed in to see pattern.

When we look into the frequency spectrum of the signal by using fast fourier transform, we observe the following,

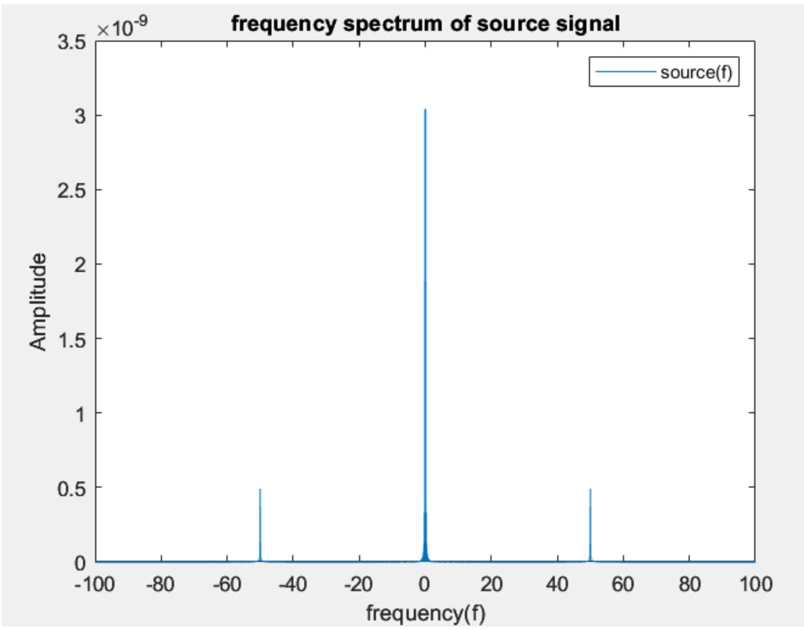


Figure 2: Frequency domain representation of source signal.

As one can expect spikes near 0 values hold the DC offset and linear characteristics information of the source signal, where the ones at +-50 Hz are component belonging to the noise.

2) PARAMETER DETECTION

In an ordinary approach one can easily suppress unwanted frequencies like in our case 50 Hz with different general types of filters no matter that it is infinite impulse or finite impulse responses assuming they are ideal, the problem is their computational cost on real time filtering.

In order to perform such filtering we need certain approaches or systems that allows us to operate in smaller values of computational costs. For that reason we are going to introduce the operation principles and implementations of digital lock in amplifier, which is an useful component for weak signal analysis for such cases.

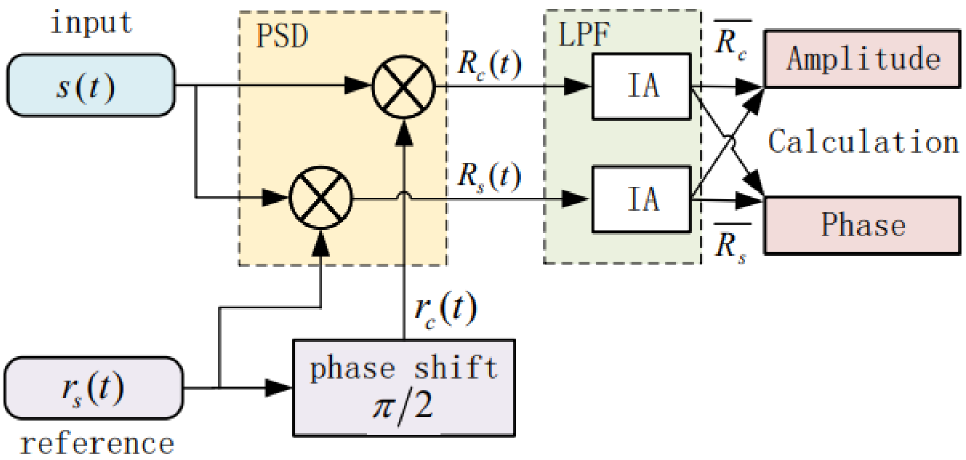
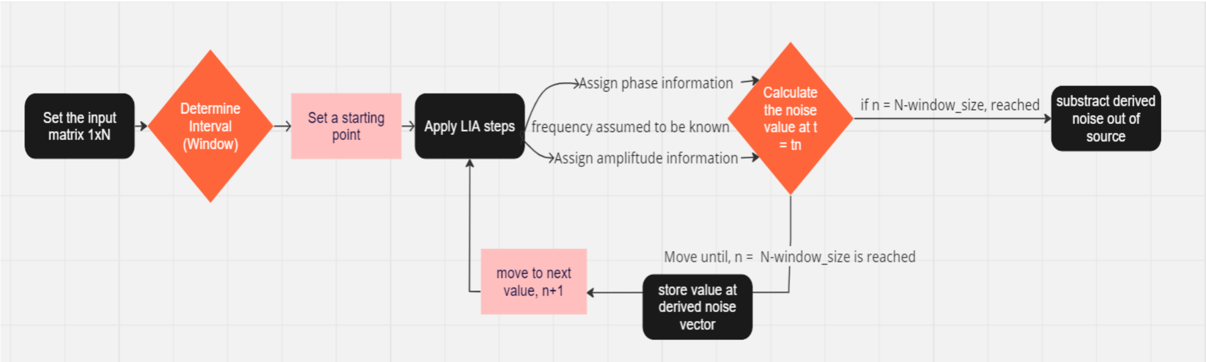


Figure 3: Block diagram of orthogonal lock in amplifier. [1]

On the above block diagram, we can easily see that s(t) represents our measured input data with respect to time, following by a multiplication of reference sine and cosine signals having same frequency with noise, after multiplication, resultant signals passed through a certain low-pass filter, and with basic trigonometric and inverse square law knowledge we determine the amplitude and phase informations of the incoming signals with respect to each data point that will be explained in later parts.

One of the important thing to care in this diagram is the given reference signals parameters. We know that the frequency is the same as inputs, this is because the analog to digital converter on measurement device has one empty pin that is useful for only to measure frequency but unable to perform measure phase and amplitude since there is a certain possibility that phase shift may occur between pins and amplitude of noise sharpened by the input.



Flowchart 1: Operation mechanism of filtering.

We may explain the operation mechanism of LIA and flowchart above as follows,

We first set the matrix representation that holds the time-data point values of the signal as 1x6001 matrix for duration 12 seconds and time interval as 0.002 seconds. After that, according to the filter's phase response we determine a window size that will move on the matrix one point at a time including "stepsize" points that will be multiplied and low-pass filtered in order to obtain in phase and quadrature components to derive phase and amplitude informations. Low pass filter that is been used in this implementation is an integral or moving average filter. This filter is a finite impulse response filter that has similar response to a butterworth filter but has narrow first transition band but response does not end with single transition it continues to bounce and decrease in frequency with respect to filter length n. With these informations, assuming frequency is known, we calculate the value of noise component in source signal that will be stored in a temporary array and subtracted from the source signal. During these operations some important properties is given as follows;

- As sampling frequency increases, phase accuracy, stepsize, that holds the number of data points, increases. Do not tempt by the increase in stepsize that is only increases because as sampling frequency increases, number of data points that made ups a period also increases results in a constant sampling frequency/stepsize ratio.
- Phase information with respect to time can be seen below for the given characteristics, it can be seen that there is a periodic shifting in phase values and as sampling frequency increases, or number of shifting, shifting frequency also increase. (It can be seen like that there is a frequency relation in shifting but in the end of report it can be seen that there is only a pattern)

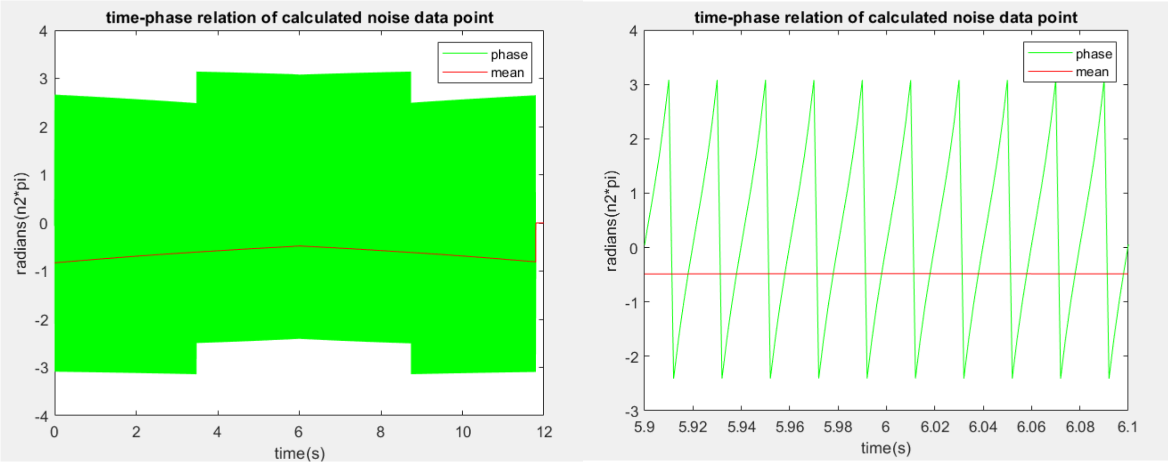


Figure 4: time(s)-phase(radians) relation for duration = 12s.

Mathematics that has been benefited during these steps is explained and given as follows;

- Production of multiplication with a reference signal,

For source signal,

$$s(t) = A\sin(2\pi f t + \phi) + n(t)$$

and reference signals,

$$r_c(t) = B\cos(2\pi f_1 t)$$

$$r_s(t) = B\sin(2\pi f_1 t)$$

Result of multiplication is,

$$R_s(t) = \frac{A \cdot B}{2}\{\cos(\phi) - \cos(4\pi f t + \phi)\} + n(t) r_s(t)$$

$$R_c(t) = \frac{A \cdot B}{2}\{\sin(\phi) + \sin(4\pi f t + \phi)\} + n(t) r_c(t)$$

$$\boxed{\begin{aligned} \sin\alpha\cos\beta &= \frac{\sin(\alpha+\beta)+\sin(\alpha-\beta)}{2} \\ \cos\alpha\cos\beta &= \frac{\cos(\alpha+\beta)+\cos(\alpha-\beta)}{2} \\ \sin\alpha\sin\beta &= \frac{\cos(\alpha-\beta)-\cos(\alpha+\beta)}{2} \end{aligned}}$$

Proof of the equation next to,

- Integral (moving) average filter,

And in the existence of an ideal integral average filter, our periodic high frequency component will be canceled out due to reason that periodic summation of a sinusoidal is equal to zero and we will be only left with sinusoidals that holds the phase information and a suppressed noise component (according to the performation on filtering).

- Drawing the informations,

As we explained in previous parts, in order to create a low computational cost filtering algorithm that can be applicable to real time measurements, we create a window on point (nth) of incoming data and evaluate this window to determine the noise component at that point of window and then shifting it to (n+1th) point to determine each point in time. In result of the evaluation of the windowed data points we obtain two components, one is in phase component and the other is quadrature.

$$\overline{R}_s = \frac{AB}{2}\cos(\phi) + n'(t)$$

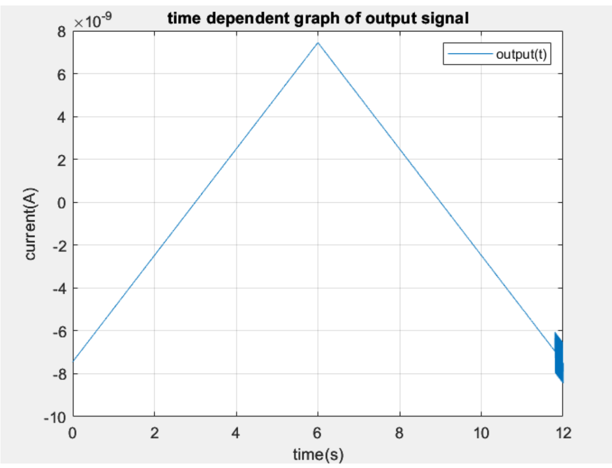
$$\overline{R}_c = \frac{AB}{2}\sin(\phi) + n'(t)$$

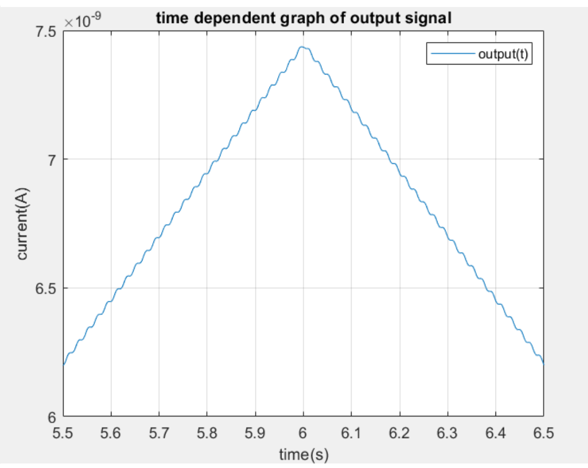
With this two values we are able to determine phase and amplitude parameters of the noise at that certain data point (n), by benefitting two components mentioned before as,

$$A = \frac{2}{B}\sqrt{\overline{R}_s^2 + \overline{R}_c^2}$$

$$\phi = \arctan\left(\frac{\overline{R}_c}{\overline{R}_s}\right)$$

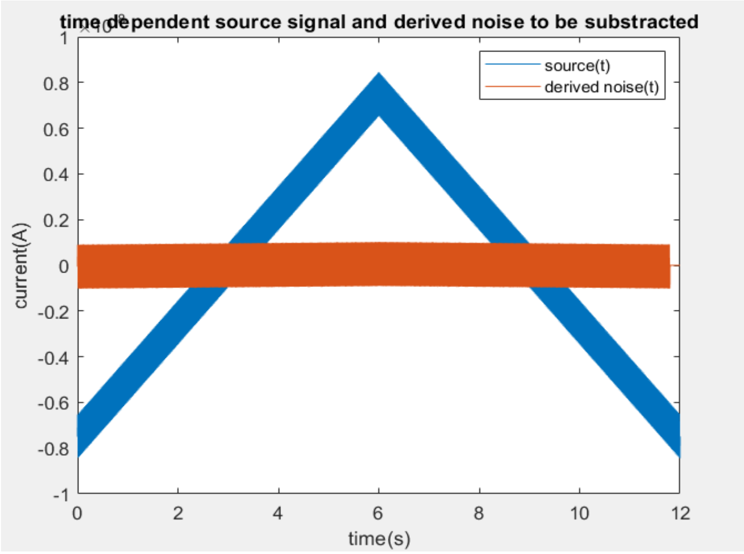
Therefore we obtain,





(a) (b)

And the components that is subtracted from each other are,

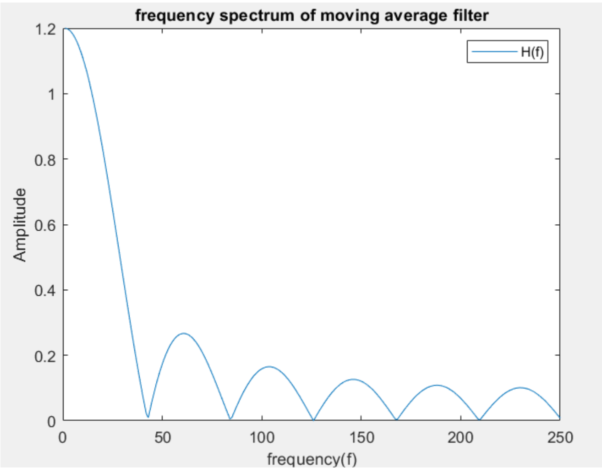


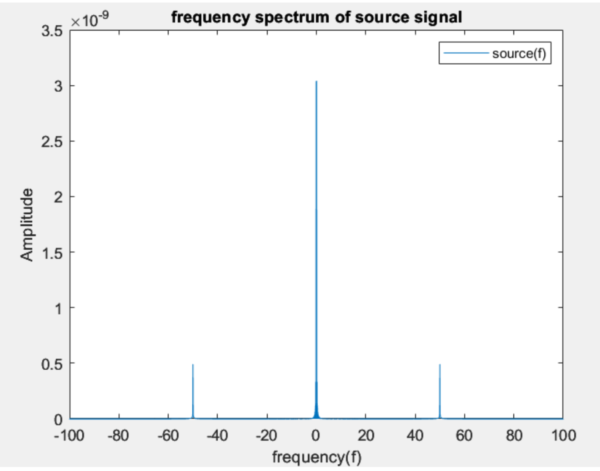
(c)

Figure 5: Output signal (a), zoomed in (b), source and derived noise under one (c).

3) EVALUATION

Frequency spectrum of filter, source signal and output is also given at once,

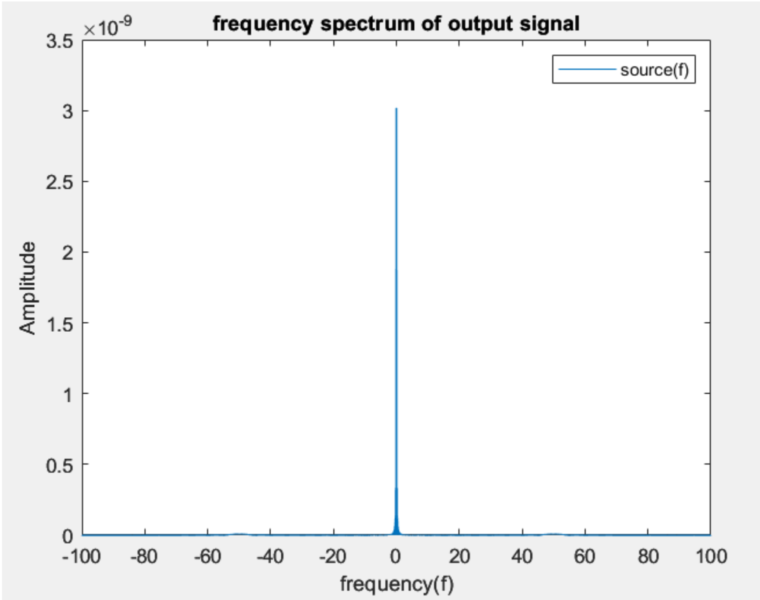




(a) (b)

Filter parameters,

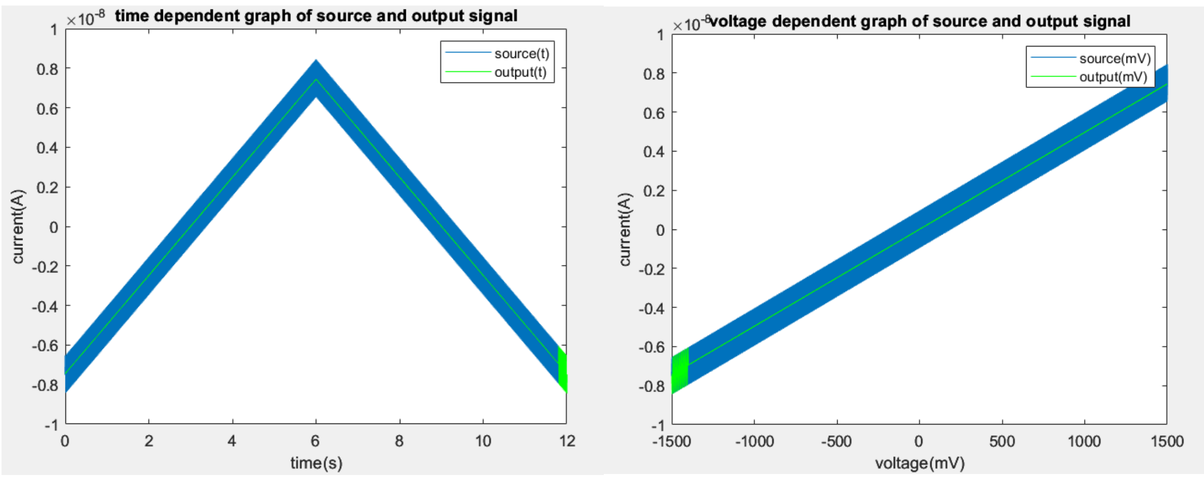
- Filter length (N) = 12,
- Cuttof frequency (w_c) = No cutoffs but first bounce at near 50 Hz,



(c)

Figure 6: Magnitude response of, (a) moving average filter, (b) source signal, (c) output signal.

In order to check if that any offset, shifting or misevaluated data points occurred we give input and output signal graphs at once depending on time and voltage,



(a) (b)

Figure 7: Source signal in blue and output is in green, (a) time dependent, (b) voltage dependent.

Exact overlap at both endings is a result of window filtering since number of filtered data points have to end at size_of_source_signal – window_size, for this case our window size was 100 and source signal constructed of 6001 data points, 5901 data points are filtered and last 100 points are not.

4) RESULTS OF OTHER IMPLEMENTATIONS

There are other trials we have conducted during the development, but we do not intend to dive in more details, so we are just sharing the responses of the filter types to the algorithm as source signal and output in time and frequency domain.

4.1) IIR Butterworth filtering

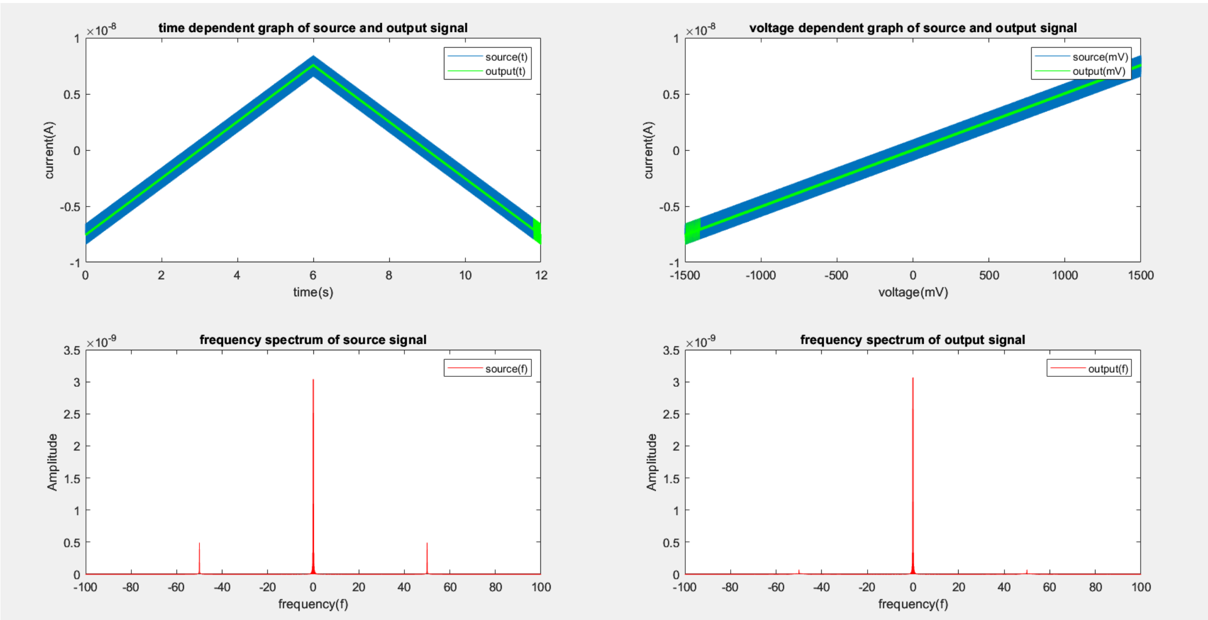


Figure 8: Butterworth filtering implementation results.

Certain parameters that has been used during the implementation are,

- Filter order (n) = 1,
- Cuttof frequency (w_c) = 50 Hz,
- Filter length = 2,

4.2) IIR Chebyshev Filtering

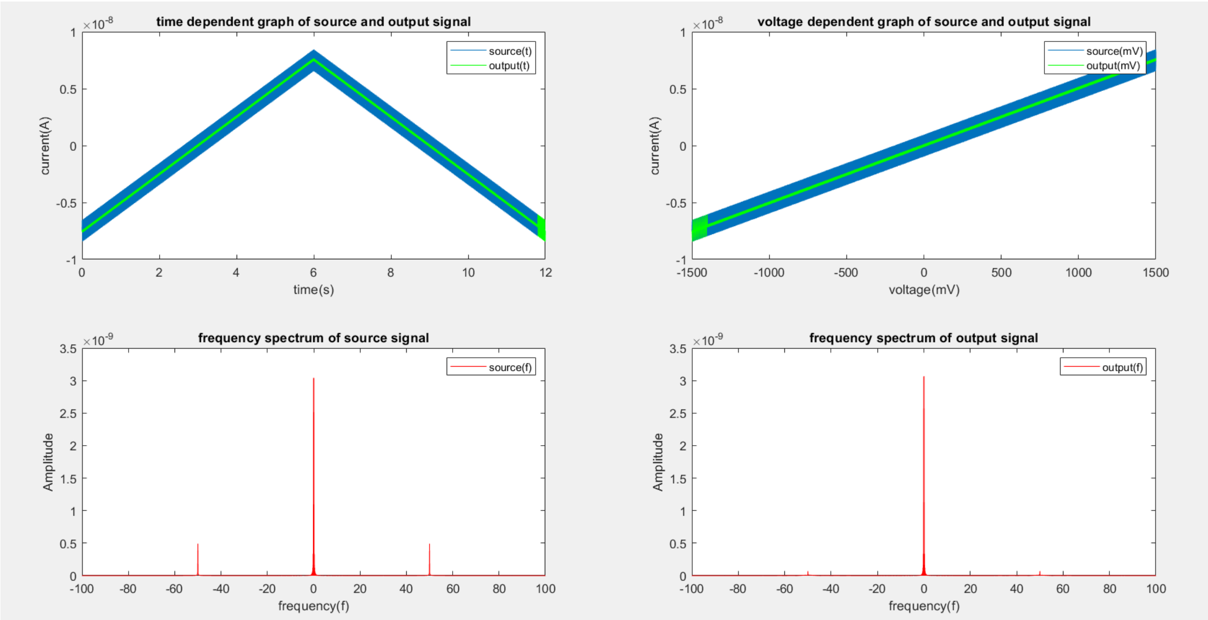
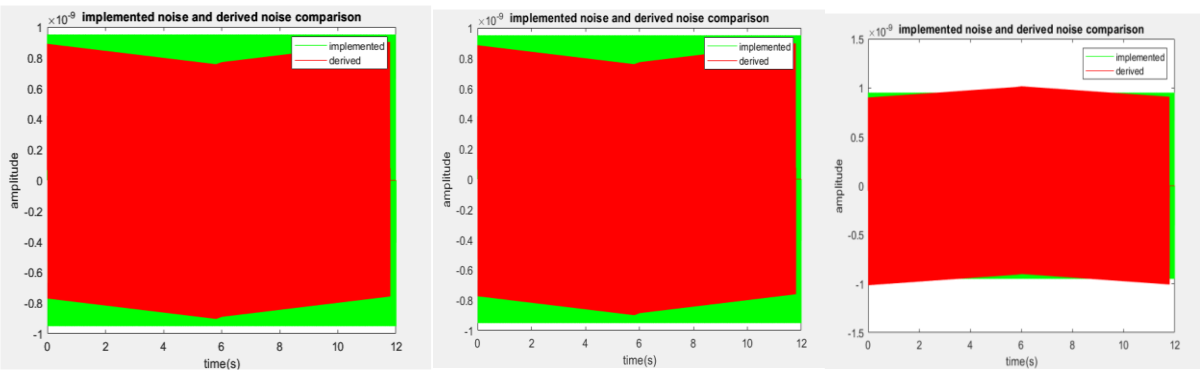


Figure 9: Chebyshev filtering implementation results.

- Filter order (n) = 1,
- Cuttof frequency (w_c) = 2 Hz,
- Filter length = 2.

For the fittest parameters for each filter implementation we found, difference in filter responses can be observable on the comparison between the implemented noise versus derived noise,



(a) (b) (c)

Figure 10: Implemented noise in green, derived noise in red for, (a) butterworth, (b) Chebyshev, (c) moving average.

4.3) Notch Filtering (Band-Reject), direct filtering:

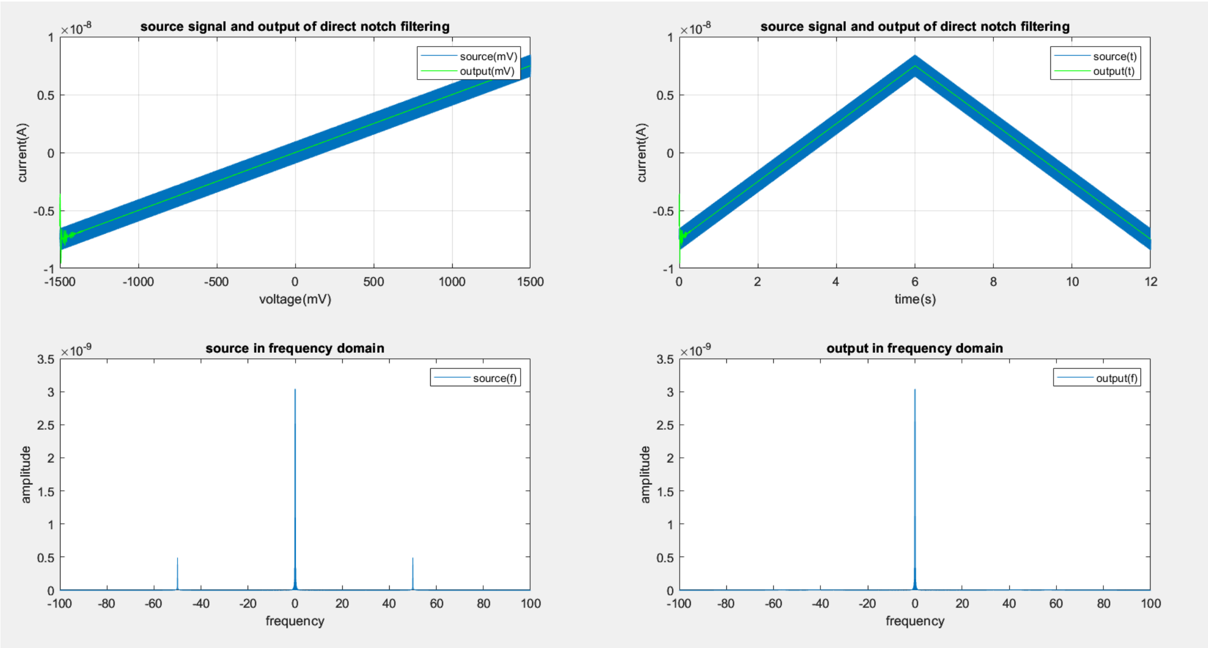


Figure 11: Results of Butterworth notch filtering applied to whole source data at once.

- Filter order = 4,
- Cutoff frequencies = [45 , 55],
- Filter length = 9

Since we apply filtering at once to whole data, there is an inevitable transient response in time domain which corrupts the initial values of the experiment and makes them uncertain. As data propagates values may correct even better than ours but for consistency in whole data proposed solution has more advantages.

Here is exact equivalent of proposed solution to compare with other type solutions,

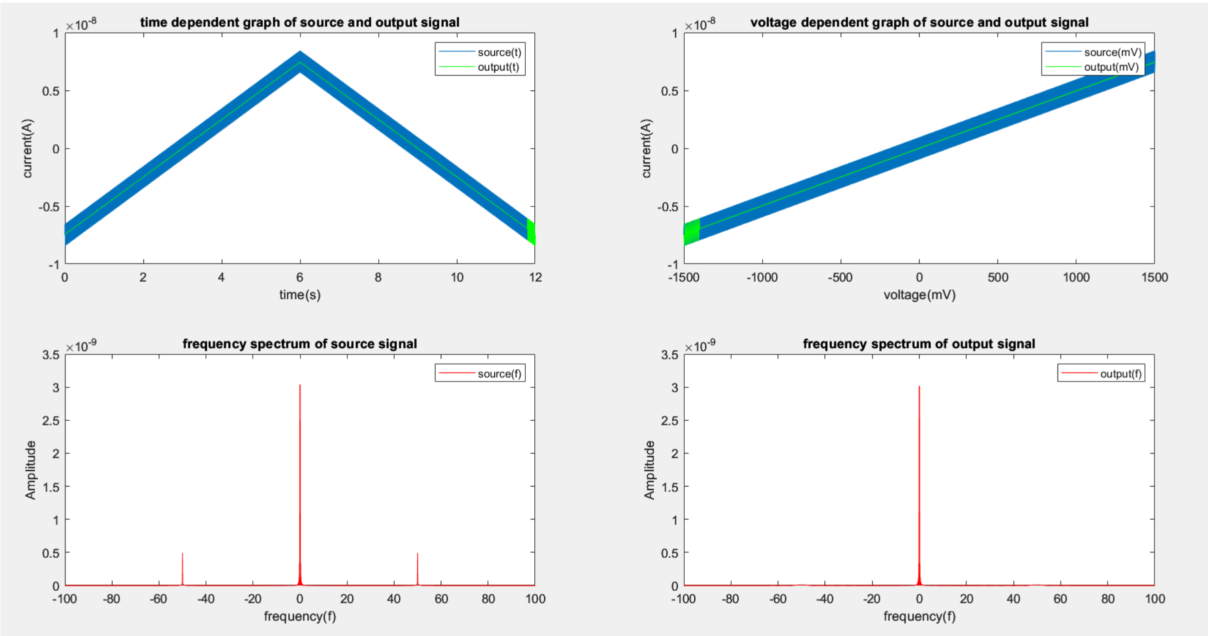


Figure 12: Results of proposed solution.

ADDITIONAL : Micromodeller DSP

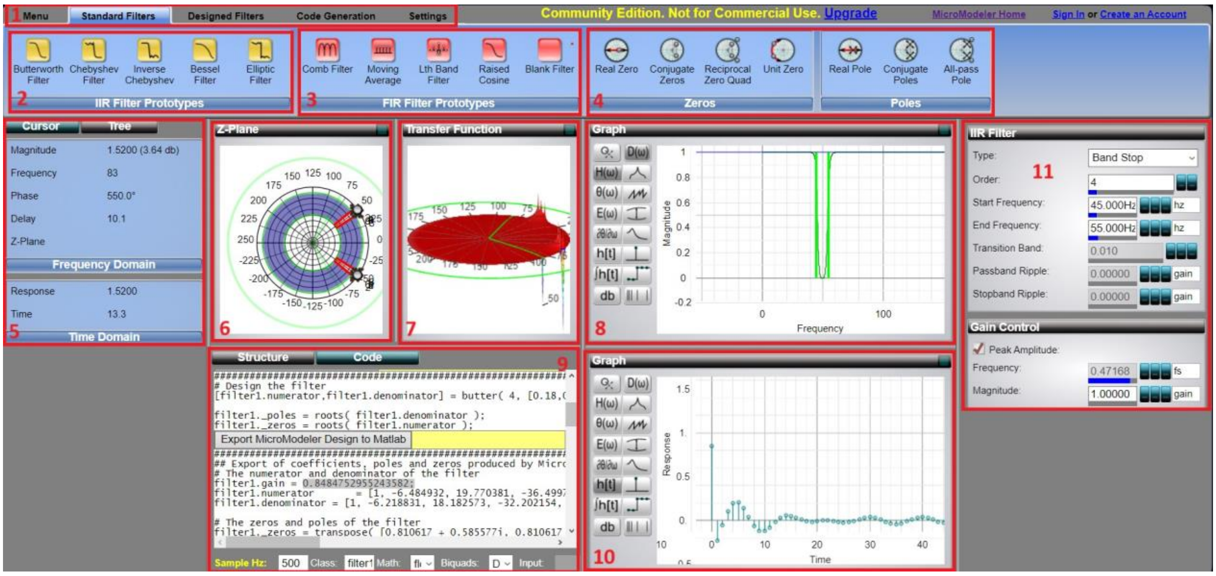
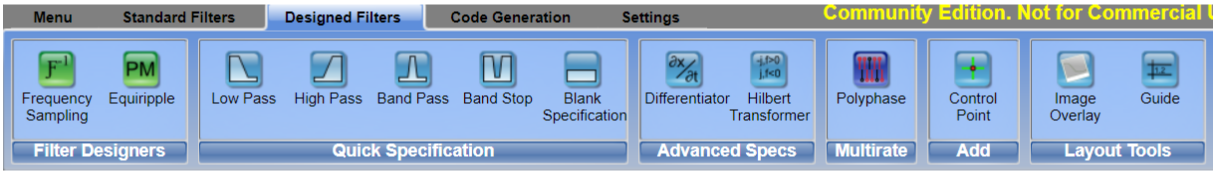


Figure 13: User interface of micromodeller DSP website, community edition.

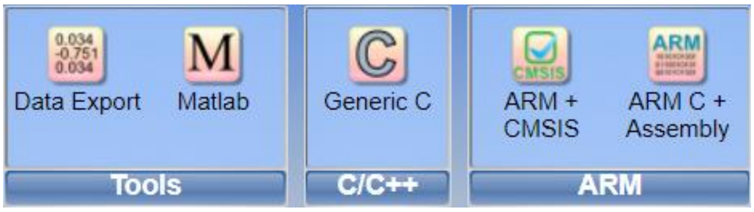
In the above figure you can see an useful tool as a website, micromodeller DSP. Website's fundamental purpose is to demonstrate certain graphs etc. that belongs to the filters to be designed in development phase for engineers and mathematicians for determined parameters and types from the user.

Features on the interface labelled as numbers and can be explained shortly as,

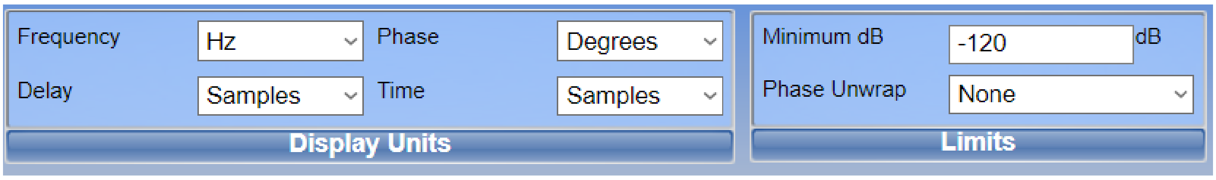
1) In menu user can shift between the windows that is customized for certain purposes for example in “Designed Filter” we can access,



Filters that are already designed and can be manipulated in the below section for parameters “11”, other mathematical tools like differentiator can be used for demonstrating purposes by the user.



From the code generation tab by dragging or clicking to the section “9” user can generate approximate codes, parameters, figures and etc. and use them in the proper IDE to observe and manipulate the results.



At last in settings tab user can change units of the section “9”.

2) Holds the choices of Infinite impulse response filter prototypes that can easily be applied to the graphs and observed the responses in just.

3) Holds the choices of Finite impulse response filter prototypes that can easily be applied to the graphs and observed the responses in just.

4) Explains the symbols that is used in section “6”, and for users that works with mathematical cases symbols can be added to the graphs and observe the changes in other demonstrations.

5) Gives exact details on graphs using cursor.

6) Z-plane graphs, can be useful for mathematical interpretations.

7) Transfer function demonstrated in 3D, magnitudes can be observed as a unit circle increases and decreases in area as cursor moves along the frequencies.

8) Demonstrates the graph that can be chose by the user for specific characteristic of filter.

9) “Structure” part shows the block diagram in z-domain and “Code” part shows approximate syntax of code in certain programming languages, might have false syntaxes but can be manipulated to a fit one.

10) Second graph for same purposes as “8”.

11) As in settings tab on ”1”, holds the usable and changeable characteristics of filters and filter prototypes if response is right (finite or infinite), can be adjust any want but it is good to move along with the “sample Hz” in settings and code parts.

REFERENCES:

- **[1]** Wang Y, Cheng Y, Chen K, Wang L, Wang H. A Software Digital Lock-In Amplifier Method with Automatic Frequency Estimation for Low SNR Multi-Frequency Signal. Applied Sciences. 2022; 12(13):6431. https://doi.org/10.3390/app12136431

- Bhattacharyya, Sabyasachi & Ahmed, Ragib & Purkayastha, Basab & Bhattacharyya, Kaustubh. (2016). Implementation of Digital Lock-in Amplifier. Journal of Physics: Conference Series. 759. 012096. 10.1088/1742-6596/759/1/012096.

- Yue Der Lin, Yu Hen Hu, Power-Line Interference Detection and Suppression in ECG Signal Processing. IEEE Transactions on Biomedical Engineering, vol. 55, no. 1, January 2008
- https://www.micromodeler.com/dsp/

EMRE SENGİR