PS Term Project Report

(EE309 & EE310)

Real-Time Estimation of Power System Frequency Using Nonlinear Least Squares Method

R. Chudamani, Krishna Vasudevan and C. S. Ramalingam

Submitted by:

Aryan Arora	(2022EEB1161)
Divyanshu Sharma	(2022EEB1169)
Garvit Bhalla	(2022EEB1310)

Real-Time Estimation of Power System Frequency Using Nonlinear Least Squares Method

GitHub Link of All executables: MATLAB Executables

Objective:

The goal of this project is to implement and validate a nonlinear least squares (NLS) algorithm for real-time estimation of power system's fundamental frequency. The focus is on dynamic response of estimator under various conditions, using MATLAB & Simulink for all simulations.

The main objectives are summarized below:

- Develop a MATLAB-based implementation of the NLS frequency estimation algorithm for a sinusoidal power signal.
- Evaluate the estimation performance under dynamic conditions, including abrupt frequency step changes and signal amplitude.
- Study the impact of signal harmonics by varying the number of harmonics included in the signal model.
- Verify accuracy by comparing the estimated frequency juxtapose the known reference value
- Attempt to reduce time complexity of code

METHODOLOGY

Non-Linear Least Square Approach:

- **Fourier-series model**

  Sample the voltage $v(t)$ at $t_n = nT_s$, and approximate it by

$$v[n] \approx \sum_{k=1}^{M} A_k \cos\left(k\omega\, nT_s + \phi_k\right).$$

- **Linear-in-parameters form**

  Stack the samples into $\mathbf{v} \in \mathbb{R}^N$ and define a parameter vector $\mathbf{a}$ of all A_k, ϕ_k. Then

$$\mathbf{v} = \mathbf{H}(\omega)\, \mathbf{a},$$

  where $\mathbf{H}(\omega)$ contains known sine/cosine columns at harmonics of ω.

In the paper (see Eq. (3)), the $N \times 2T$ regressor matrix $\mathbf{H}(\omega)$ is split into two $N \times T$ blocks—one for cosines and one for sines:

$$\mathbf{H}(\omega) = \left[\,\mathbf{H}_a(\omega)\ \ \mathbf{H}_b(\omega)\right],$$

where each block is defined entry-wise for $n = 0, \ldots, N-1$ (sample index) and $k = 1, \ldots, T$ (harmonic index):

- **Cosine block $\mathbf{H}_a(\omega)$:**

$$\left[\mathbf{H}_a(\omega)\right]_{n,k} = \cos\left(k\,\omega\,nT_s\right),$$

- **Sine block $\mathbf{H}_b(\omega)$:**

$$\left[\mathbf{H}_b(\omega)\right]_{n,k} = \sin\left(k\,\omega\,nT_s\right).$$

Written out in matrix form, with T_s the sampling period:

$$\mathbf{H}_a(\omega) = \begin{bmatrix} \cos(1\omega\,0T_s) & \cos(2\omega\,0T_s) & \cdots & \cos(T\omega\,0T_s) \\ \cos(1\omega\,1T_s) & \cos(2\omega\,1T_s) & \cdots & \cos(T\omega\,1T_s) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(1\omega\,(N-1)T_s) & \cos(2\omega\,(N-1)T_s) & \cdots & \cos(T\omega\,(N-1)T_s) \end{bmatrix},$$

$$\mathbf{H}_b(\omega) = \begin{bmatrix} \sin(1\omega\,0T_s) & \sin(2\omega\,0T_s) & \cdots & \sin(T\omega\,0T_s) \\ \sin(1\omega\,1T_s) & \sin(2\omega\,1T_s) & \cdots & \sin(T\omega\,1T_s) \\ \vdots & \vdots & \ddots & \vdots \\ \sin(1\omega\,(N-1)T_s) & \sin(2\omega\,(N-1)T_s) & \cdots & \sin(T\omega\,(N-1)T_s) \end{bmatrix}.$$

Thus the full linear model is

$$\mathbf{v} = \mathbf{H}(\omega)\,\mathbf{a} = \left[\mathbf{H}_a(\omega)\ \mathbf{H}_b(\omega)\right] \begin{bmatrix} \mathbf{a}_c \\ \mathbf{a}_s \end{bmatrix},$$

where $\mathbf{a}_c$ and $\mathbf{a}_s$ collect the cosine- and sine-mode amplitudes, respectively.

Least-squares for fixed ω

$$\hat{\mathbf{a}}(\omega) = \left[\mathbf{H}^T\mathbf{H}\right]^{-1}\mathbf{H}^T\mathbf{v}.$$

Nonlinear-least-squares cost

Eliminate $\mathbf{a}$ to get the residual

$$\mathbf{e}(\omega) = \mathbf{v} - \mathbf{H}(\omega)\,\hat{\mathbf{a}}(\omega), \quad J(\omega) = \|\mathbf{e}(\omega)\|^2.$$

1-D frequency search

$$\hat{\omega} = \arg \min_{\omega \in [\omega_{\min}, \omega_{\max}]} J(\omega),$$

found by a simple grid search over the allowed frequency range.

Here that search range is kept from 48.5 to 51.5 Hz

Sampling Constraints

$$v(t) = \sin\omega_0 t + 0.0856\sin 5\omega_0 t + 0.0428\sin 7\omega_0 t + 0.0306\sin 11\omega_0 t + 0.0183\sin 13\omega_0 t.$$

The fundamental frequency is 50 Hz, and the signal contains harmonics up to the 7th order in this expression. However, for a more general model accommodating up to the 13th harmonic, the highest frequency component present in v(t) would be:

$f_{max} = 13\times50 = 650$ Hz

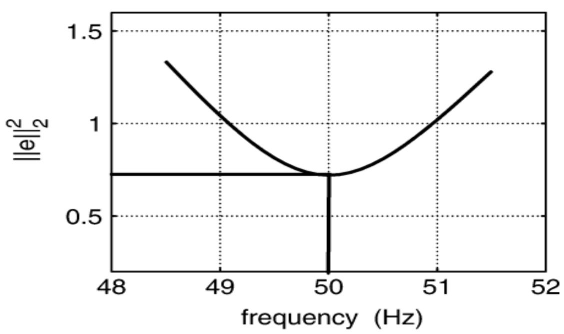
According to the Nyquist criterion, to avoid aliasing, the sampling frequency must be at least twice the highest frequency component:

$f_s \geq 2\times650 = 1300$ Hz

To ensure sufficient margin we took $f_s = 1.6$ KHz.

Later on, we have proposed a modification in this approach which reduced the operations involved significantly making it easier to deploy on dsp boards.

Simulation Results



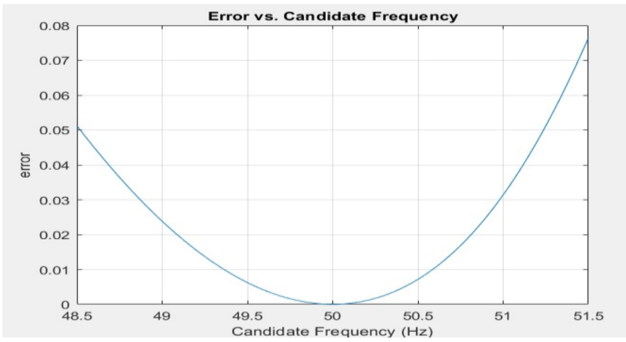


Fig. 1. Illustration of the variation of $||\mathbf{e}||_2^2$ as a function of f_0, when f_0 is varied from 48.5 Hz to 51.5 Hz in steps of 0.1 Hz.

From the figure:

- The norm of the error $||\mathbf{e}||_2^2$ varies as the assumed fundamental frequency f_0 is swept.
- There is a clear **minimum** in the error exactly at $f_0 = 50\,\mathrm{Hz}$, the true system frequency.
- Away from the true frequency, the error increases symmetrically.
- **At off-nominal frequencies (away from 50 Hz), the algorithm does not work precisely, as the error is significantly higher.**

EFFECT OF WINDOW LENGTH

Analysis window length refers to the time duration over which a signal segment is analyzed for processing, estimation, or transformation.

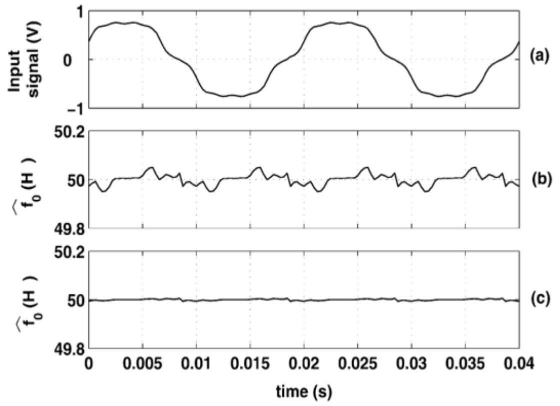
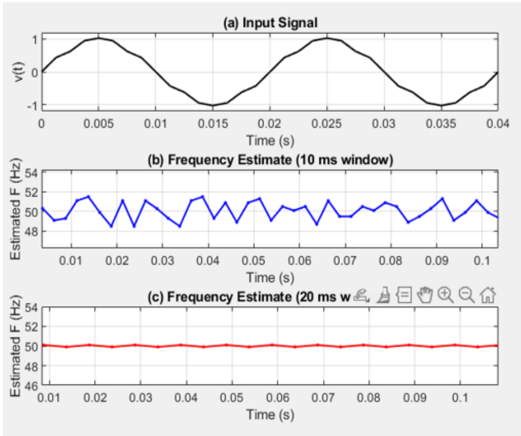


Fig. 2. Simulation results for $n_h = 13$ when the input signal's frequency is constant at 50 Hz. (a) Input signal. (b) Estimated frequency for $w_l = 10$ ms. (c) Estimated frequency for $w_l = 20$ ms.

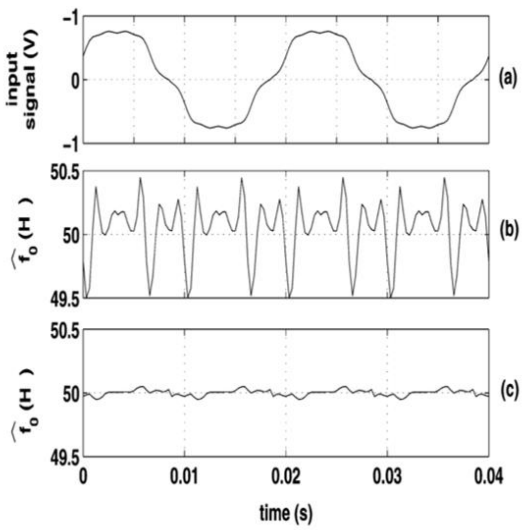


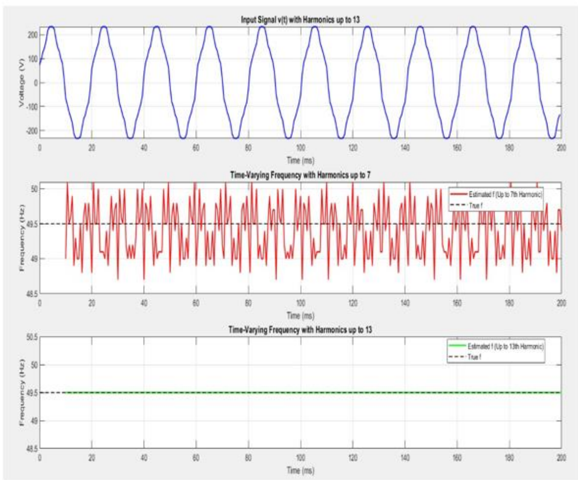
Test Software: **MATLAB**

Shorter Analysis Window: Fewer samples which means poorer frequency

Longer Analysis Window: More samples which means better frequency resolution and hence, the estimated frequency f is smoother and closer to the true value i.e. 50Hz.

Effect Of Number of Harmonics





(a) Input Signal Waveform for window length of 10ms when signal frequency is constant at 50hz.
(b) Estimated frequency for number of harmonics =7
(c) Estimated frequency for number of harmonics =13

The number of harmonics included in model was varied keeping window at 10ms.The results get better as more no of harmonics are considered.

Input **frequency** change **linearly** with time

In the “bath-tub” curve, you can see the estimation error is typically largest where the slope of the true frequency is steepest—i.e., where the change is happening most quickly.

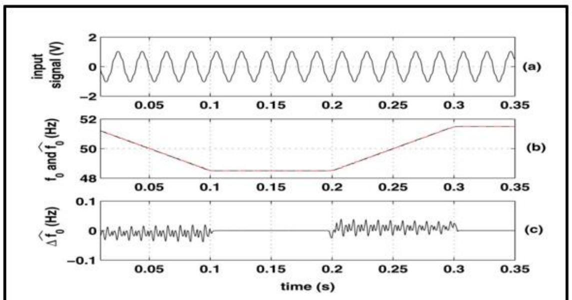
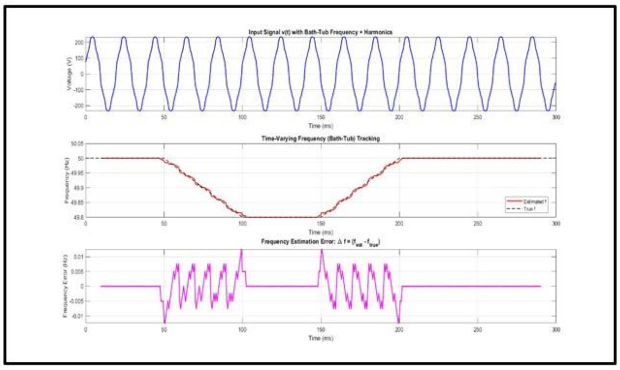


Fig. 10. Computer simulation results when the input frequency changes with time. (a) Input signal. (b) The true “bath-tub” frequency track and its estimate (where $\hat{f}_0$ has been assigned to the center of the analysis window). (c) Plot of $|f_0 - \hat{f}_0|$.



Observations:

When the frequency changes *during* the analysis window some part of the data in that window still corresponds to the old frequency while the rest corresponds to the new frequency!

Therefore, **change of buffer vector** causes error.

Frequency suddenly changes by **2Hz**

- In an electric power system, frequency fluctuation is a common phenomena
- Computer simulation results show that the estimated frequency goes through a transient phase before settling to the new value (**51 Hz** in our case)

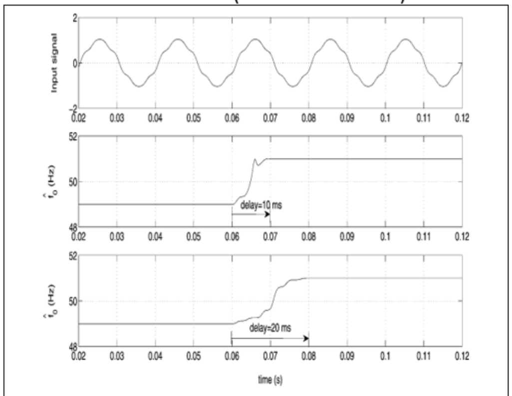
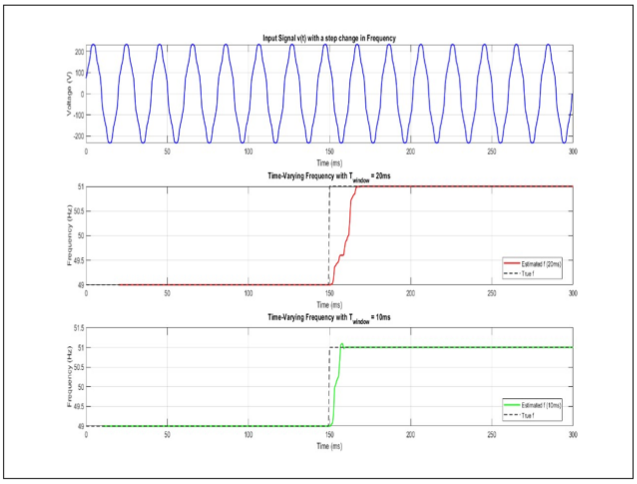


Fig. 7. Computer simulation results for $\hat{f}_0$ when the input signal frequency suddenly changes from 49 Hz to 51 Hz; $n_h = 13$. (a) Input signal. (b) Track of $\hat{f}_0$ for $w_l = 10$ ms. (c) Track of $\hat{f}_0$ for $w_l = 20$ ms.



- **Small window (10 ms):** Fast convergence (≈10 ms delay) but higher jitter/overshoot.
- **Large window (20 ms):** Slower convergence (≈20 ms delay) with smoother, more accurate estimates.

Load suddenly changes by +15%

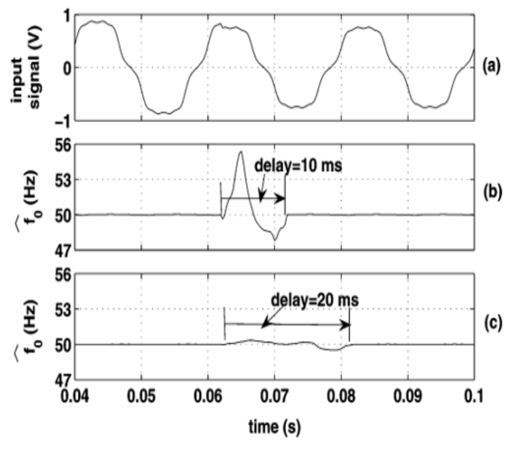
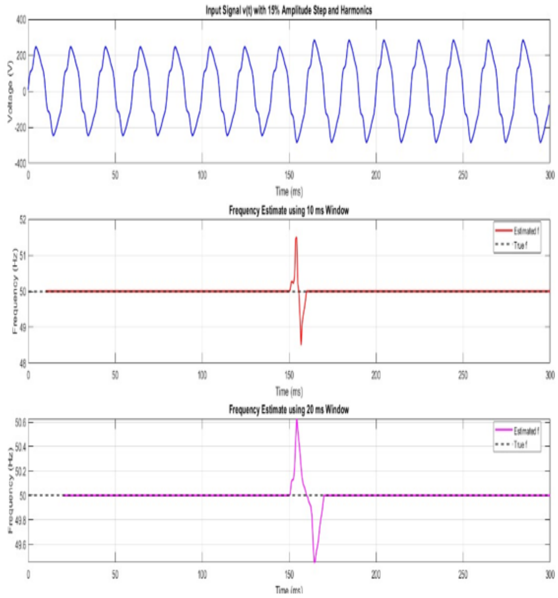


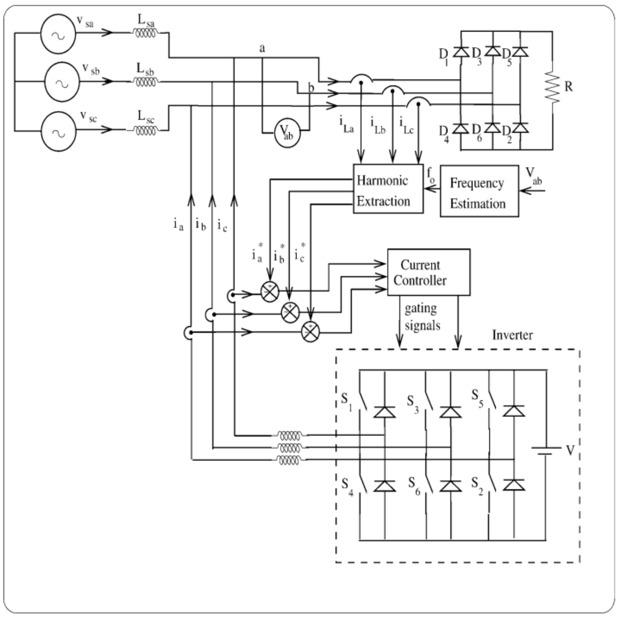
Fig. 12. Computer simulation results for $\hat{f}_0$ when there is a step change in the input's amplitude. (a) Input signal. (b) $\hat{f}_0$ for $w_l = 10$ ms. (c) $\hat{f}_0$ for $w_l = 20$ ms.



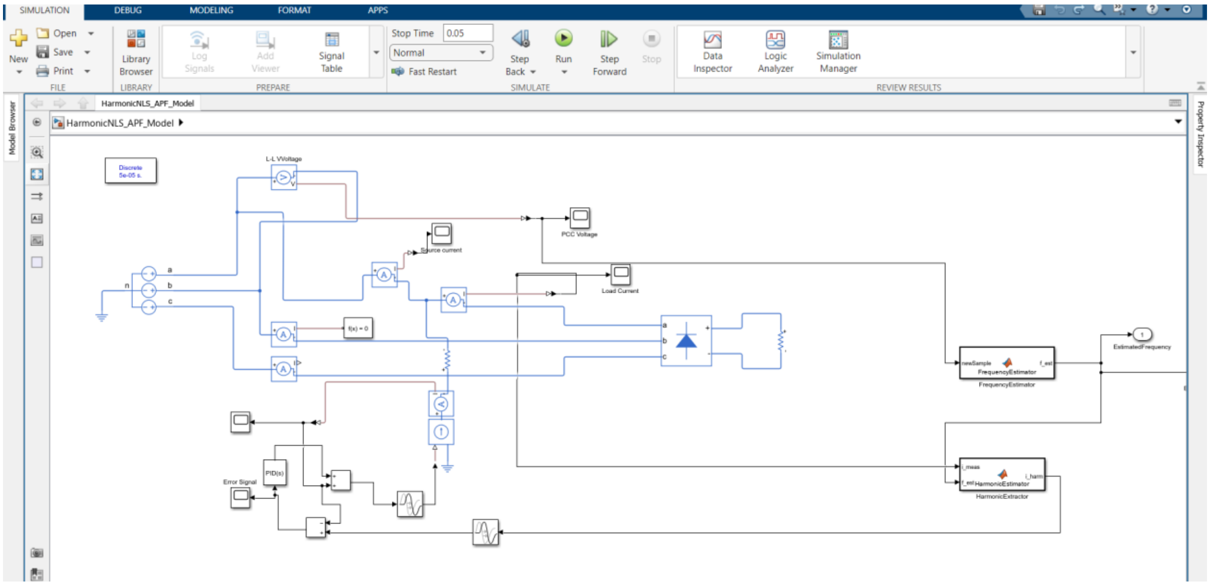
Observations:

For shorter window of 10ms we have larger overshoot of around 5 Hz and for larger window size of 20ms we have smaller overshoot of 0.62Hz.

A general Test Case for our algorithm:



Active Power Filter



Active Power Filter in SIMULINK

Here we have **used PI controller instead of inverter** due to unavailability of a customisable block which provides switching pulses to the controlled switches S_1 to S_6. We tuned **P = 0.86 and I = 0.1**. The controller then injects these harmonics into the system, thereby **relieving the grid from having to supply the harmonic components**.

<u>Research Paper Results of Active Power Filter:</u>

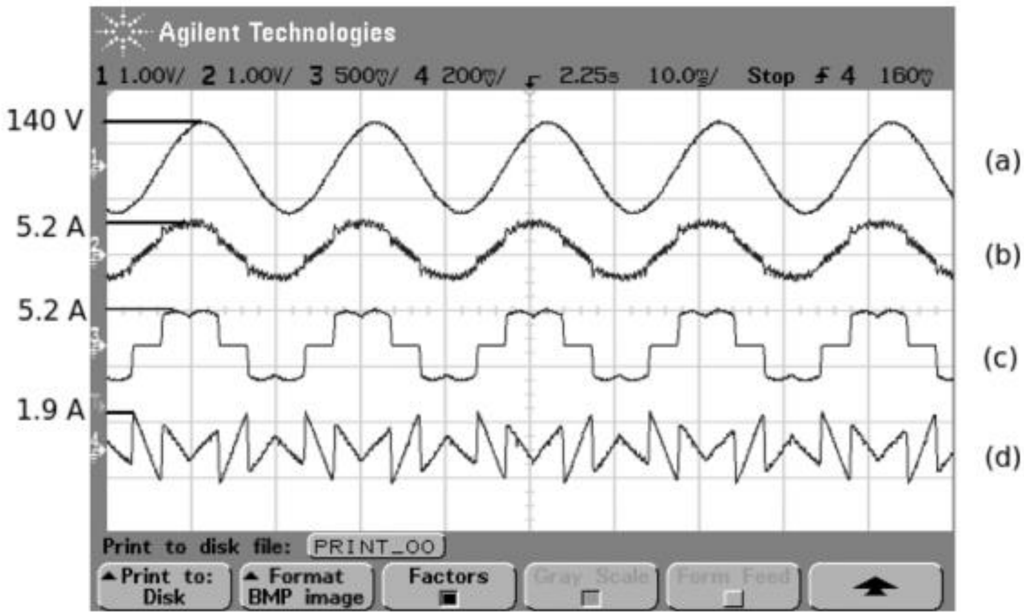
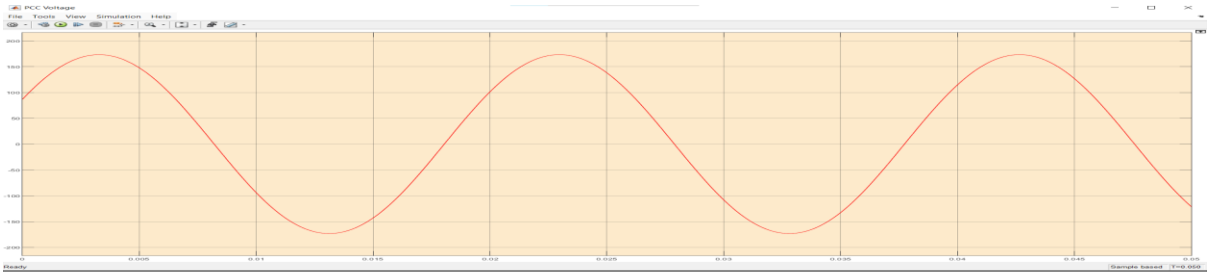


Fig. 18. Experimental results showing the compensation achieved by the active power filter using the proposed algorithm. (a) Voltage at the PCC. (b) Source current after compensation. (c) Load current. (d) Total harmonic components.

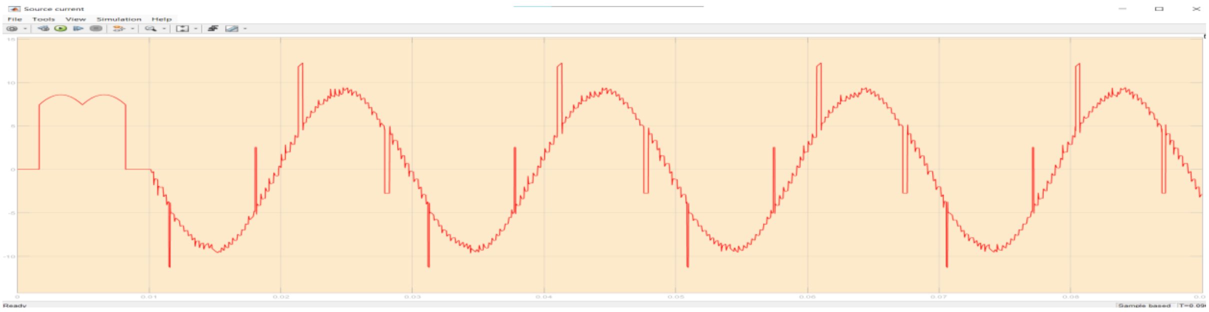
Our simulation Results for Active Power Filter

Note: Kindly ignore the results from 0 to 10ms as the algorithm itself takes one window length (here, 10ms) to begin estimation.

Voltage at PCC (evidently free from harmonics) :

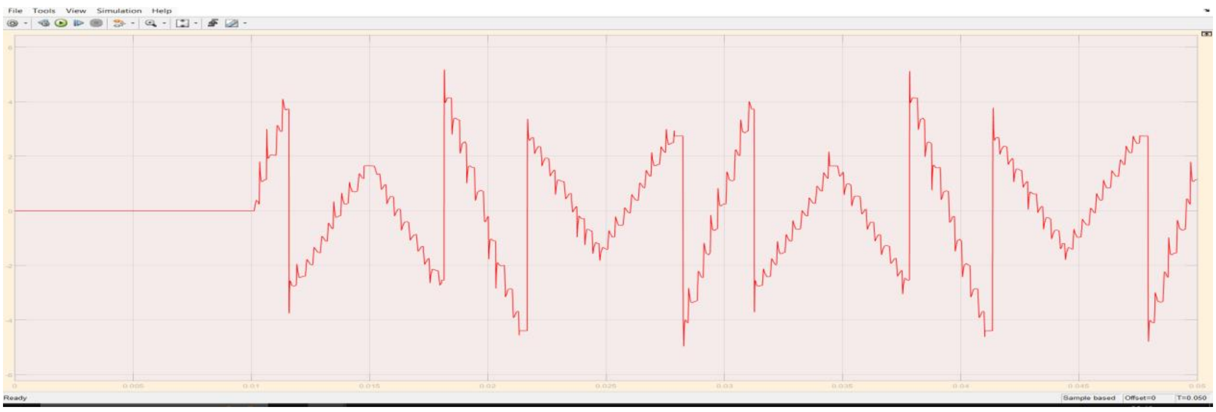


Source current (almost sinusoidal) :



Load Current (as per the demand of non-linear load i.e. 3-ph rectifier)**:**

Total Harmonics Components:



Discussion:

a) **Real-Time Frequency Tracking:** The NLS estimator reliably follows abrupt frequency steps within a single 10 ms window, offering sub-millisecond responsiveness crucial for live power system monitoring and control. Incorporating multiple harmonics into the NLS signal model reduces estimation deviations under waveform distortion.

b) **Analysis Window Trade-Off:** Short windows yield faster updates but larger overshoot (≈5 Hz), whereas longer windows (20 ms) produce smoother estimates with minimal overshoot (0.62 Hz), highlighting the balance between speed and accuracy.

c) **Search Optimization and DSP Computational Efficiency:** Replacing exhaustive linear search with a targeted minimum-norm selection halved operations from 31 to 12 iterations. This cut operations by over 60%, enabling practical embedded DSP deployment.

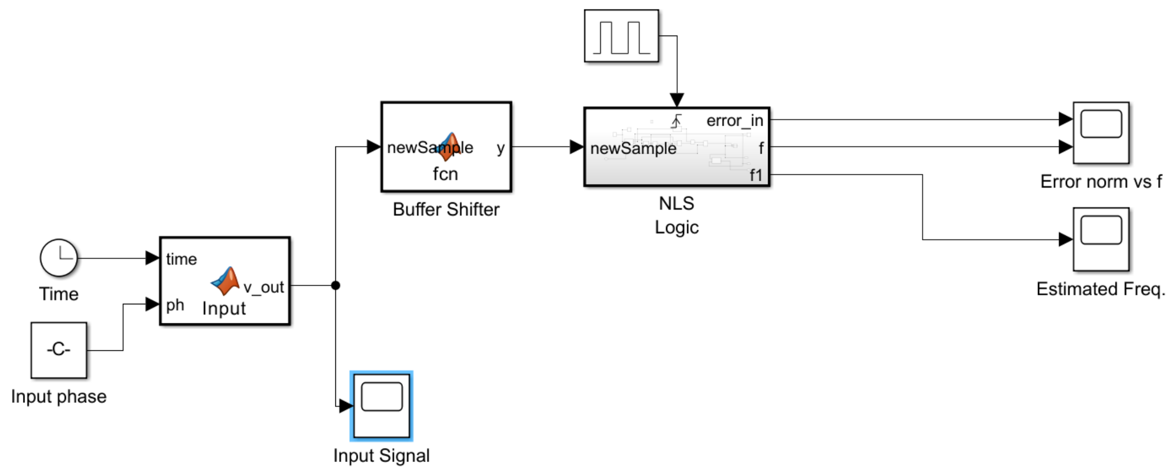
d) **Database Integration and Visualization:** Real-time frequency outputs are streamed to a MongoDB server and visualized via a web interface, validating end-to-end data acquisition, transmission, and remote supervisory control readiness for smart grid applications.

e) **Active Power Filter Performance and Smart Grid Potential:** Simulated APF with PI control (P=0.86, I=0.1) effectively suppresses load harmonics, restoring near-sinusoidal source currents, underscoring applicability for adaptive harmonic compensation in ***modern smart grids***.

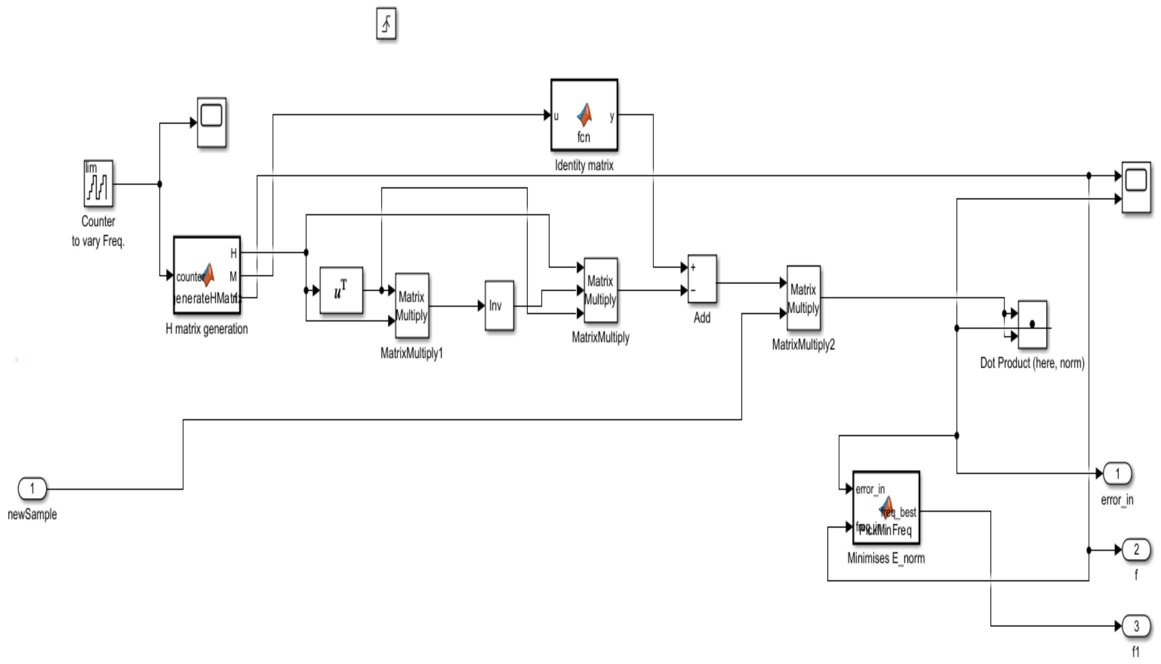
f) **Response to Abrupt Frequency an Amplitude Step Changes:** In a 1 Hz step change from 51 Hz to 50 Hz, the estimator converged within one analysis window, with overshoot inversely proportional to window length, demonstrating robust transient tracking. Under ±15 % signal amplitude fluctuations, frequency remained effectively unchanged, confirming the NLS estimator's robustness against amplitude perturbations.

g) **Effect of Number of Harmonics Considered:** Increasing the number of modelled harmonics progressively improves estimation accuracy, though beyond a certain count returns diminish, guiding optimal harmonic selection for balanced performance.

<u>Running Algorithm in Simulink:</u>

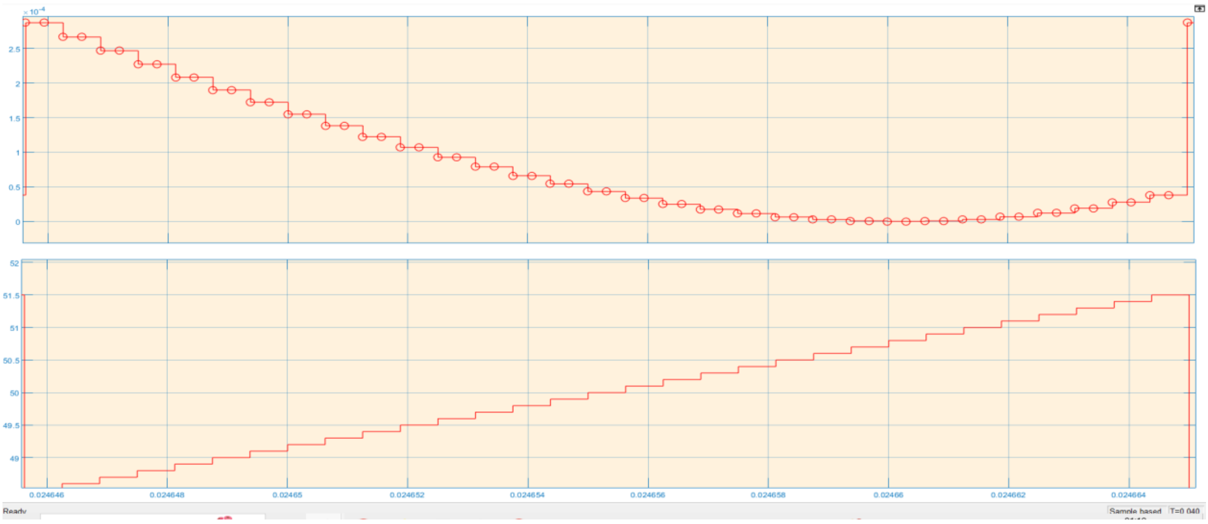


Here input phase is 2*pi*50.5 and buffer shifter shifts one sample in each iteration i.e. removes a past sample to add a new sample. The NLS Logic block is shown below:



The counter used is to vary frequeny from 48.5 to 51.5 Hz . At each iteration counter clocks and increases frequency by 0.1Hz. In total we will need 31 states so we used a **31 Mod counter** which can be physically formed **easily from 5 Flip Flops**

On magnifying the output of norm of error vs frequency we see that minima occurs at 50.5Hz which is indeed the input frequency.



Modification:

Decrement in Number of Operations

For deploying the code on a dsp board like F2837xD C2000 MCU the number of operations must be as small as possible. Here**,** we have used **Fibonacci Search Method** instead of **linear search** on the elements of **error's norm** vector to find the *minimum* of all those.

Comparison of Number of Operations per second
Earlier: MAC Operations: 16*16 (multiplication of 16 X 16 matrix)
Samples: 1600 (as f_s =1.6KHz)
Operations to find Minima of error norms: 31
Total = 16*16*1600*31= 1,26,97,600
Now: MAC Operations: 16*16
Samples: 1600
Operations to find Minima of error norms: 12
Total = 16*16*1600*31= 49,15,200

This way we used 12 operations instead of 31 operations ***reducing the total no of operations by 61.29%*** and significant **computation burden**.

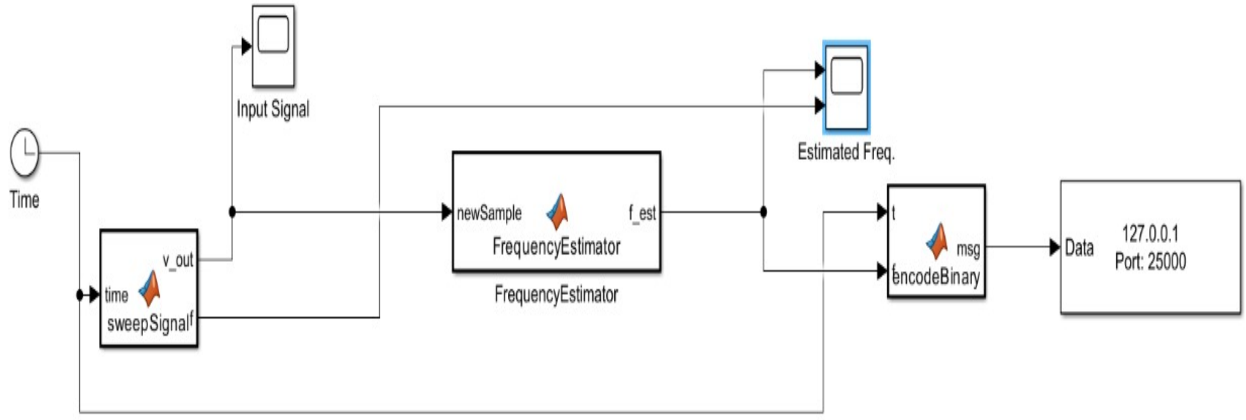
Extension In Work

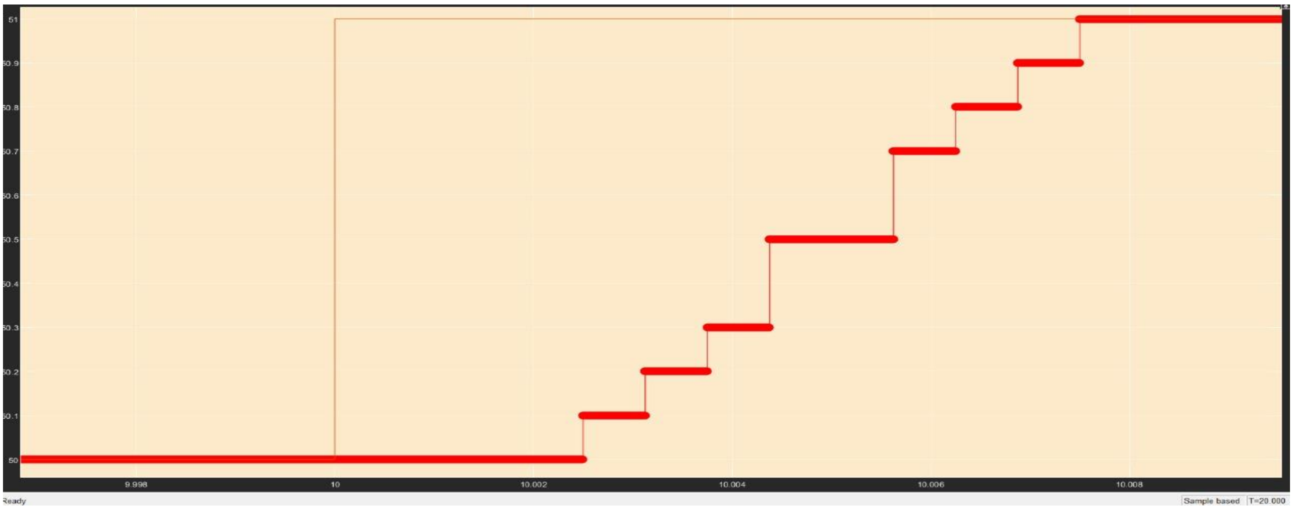
Real-time integration with a MongoDB Web interface

We have uploaded our data on MongoDB server and served it to clients via a website.

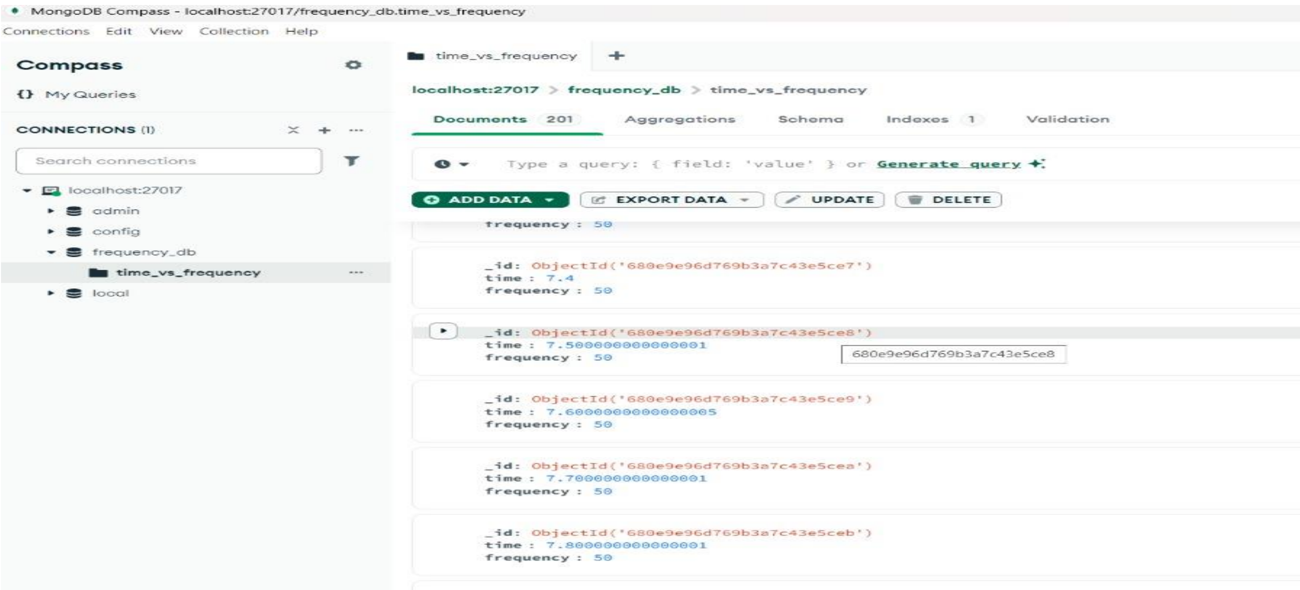
This has been tested as shown below:

The input frequency is given **step change from 50 Hz to 51 Hz** and then kept constant at 51 Hz The algorithm successfully tracks the frequency change and updates the data on server in **Real Time** as the code runs!

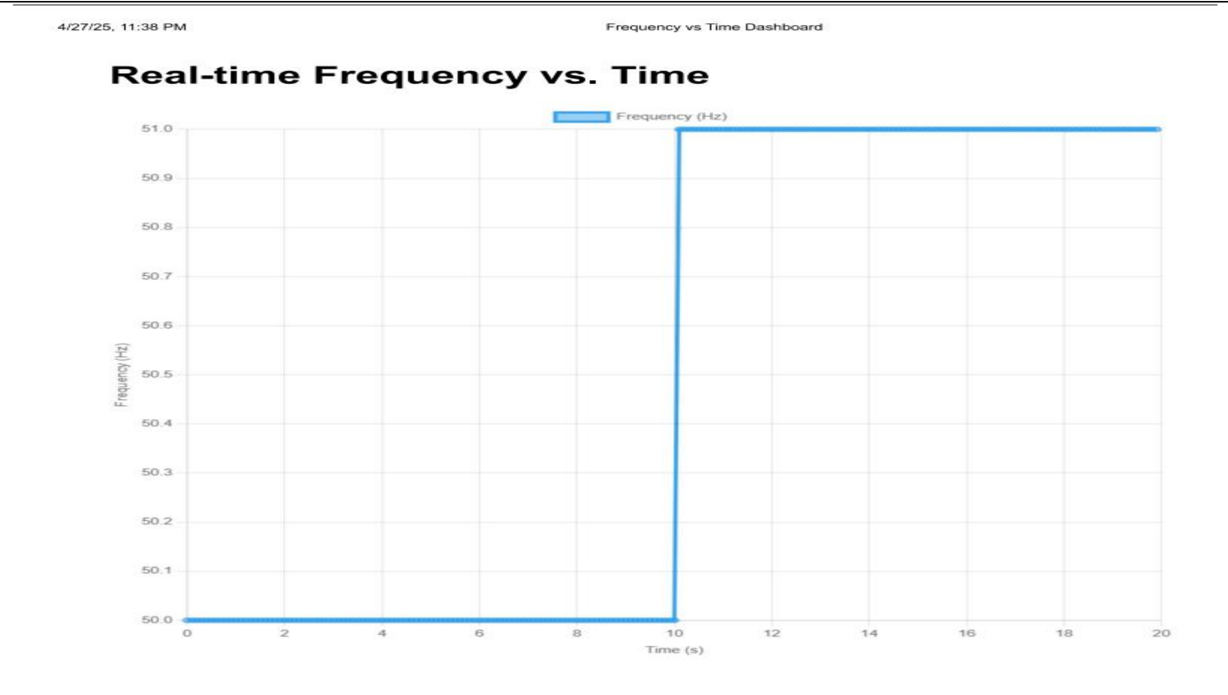




From the above simulation we see that the algo does step change in less than 8ms (almost 1 window length) and the same is sent on MongoDB servers



Each packet consists of an ID, a timestamp, and an estimated frequency corresponding to the window.



Plot of frequency vs time on MongoDB for a step change

From above we can see that **MATLAB Real-Time** library is used to send the samples to the server-based Database system of MongoDB which simultaneously plots the values of freq. vs time graph.

CONCLUSION

1. **Real-Time Frequency Tracking**: The NLS estimator reliably tracks abrupt frequency changes within a single 10 ms window, achieving sub-millisecond responsiveness essential for real-time monitoring and control of power systems. Incorporating multiple harmonics reduces estimation error under waveform distortion, confirming algorithm effectiveness.
2. **Analysis Window Trade-Off**: A shorter 10 ms window delivers rapid frequency updates but incurs larger overshoot (~5 Hz), whereas extending to 20 ms produces smoother estimates with minimal overshoot (~0.62 Hz), illustrating the balance between estimation speed and accuracy in NLS-based detection.
3. The Nonlinear Least Squares (NLS) method excels in handling both **frequency step changes** and gradual variations with minimal delay. Unlike PLLs, which require multiple cycles (around 5 as mentioned in research paper) to lock, NLS swiftly estimates frequency within a single analysis window (10ms or 20ms).
4. **Search Optimization & DSP Efficiency**: Replacing exhaustive linear search with a targeted Fibonacci (minimum-norm) search reduces iterations from 31 to 12, cutting total operations by over 60% and enabling practical embedded deployment on DSP platforms without sacrificing estimation performance.
5. **Database Integration & Smart Grid Readiness**: Real-time frequency outputs are streamed to a MongoDB server and visualized via a web interface, validating end-to-end data acquisition, transmission, and remote supervisory control, and demonstrating the solution's viability for smart grid monitoring and adaptive management.
6. **Robustness to Signal Variations**: The NLS estimator maintains accurate frequency estimates under ±15% amplitude fluctuations and benefits from modelling additional harmonics up to the point of diminishing returns, ensuring reliable operation under dynamic waveform conditions and guiding optimal harmonic selection.

References:

1) https://ieeexplore.ieee.org/abstract/document/5109872

2) APF Technological Review (CITESEERX)
D. Singh, "Harmonics Mitigation Using Active Power Filter: A Technological Review," 2005. Tutorial-style survey of shunt, series, and hybrid APF topologies

3) https://www.geeksforgeeks.org/how-to-install-mongodb-on-windows/

4) https://en.wikipedia.org/wiki/Non-linear_least_squares