Virtual Digital Signal Processing

Using LabVIEW for application in AC Susceptibility measurements

JUNE 1 – JUNE 23, 2020

Inter-University Accelerator Centre
Authored by: R K Rupesh
IMSc Physics, School of Physics
University of Hyderabad, Hyderabad

Under the Guidance of
Dr Joby Antony & Rajesh Nirdoshi, Inter-University Accelerator Centre, Delhi

अंतर विश्वविद्यालय त्वरक केंद्र
Inter-University
Accelerator Centre - (IUAC)

Introduction

The project that I did in IUAC was a fantastic experience for me. I had a lot of learning in various aspects of signal analysis and graphical programming using LabVIEW. Under the guidance of my mentors, I successfully made a Virtual Instrument of a Lock-In amplifier, which has numerous applications in a wide range of fields for efficient signal processing.

Acknowledgements

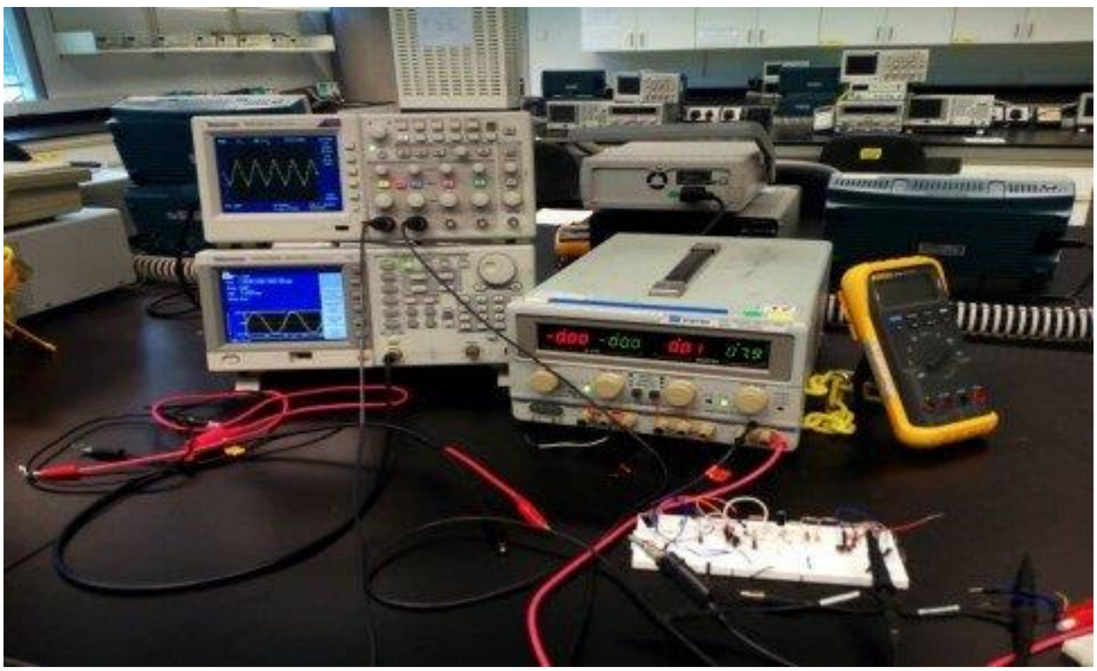
This internship would not have been possible without the contribution and support of others. My sincere gratitude to:

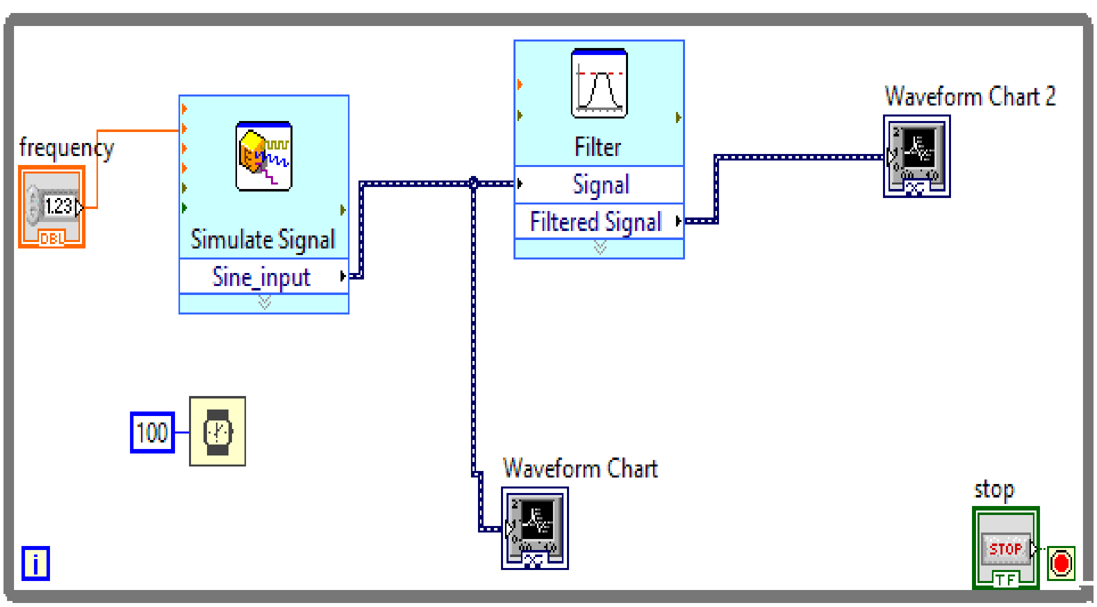
- Dr Joby Antony for endorsing my application and giving me an opportunity and for his constant supervision, support and able guidance that enabled me to complete the internship program successfully.
- Rajesh Nirdoshi sir for his constant supervision and support throughout the program.
- Dr Saif for meticulously coordinating all the events and programs.
- Dr S Gosh, Dr A Tripathi & Dr N Madhavan for their valuable seminars that gave an insightful perspective of the work at IUAC
- The Director, IUAC - Prof A C Pandey for being at the frontier for holding the summer program despite the odds of the global pandemic situation.
- To my parents for their support and motivation throughout my internship period

Contents

Virtual Instrumentation

A virtual instrument consists of an industry-standard computer or workstation equipped with powerful application software, cost-effective hardware such as plug-in boards, and driver software, which together perform the functions of traditional instruments. Virtual instruments represent a fundamental shift from conventional hardware-centred instrumentation systems to software-centred systems that exploit the computing power, productivity, display, and connectivity capabilities of desktop computers and workstations





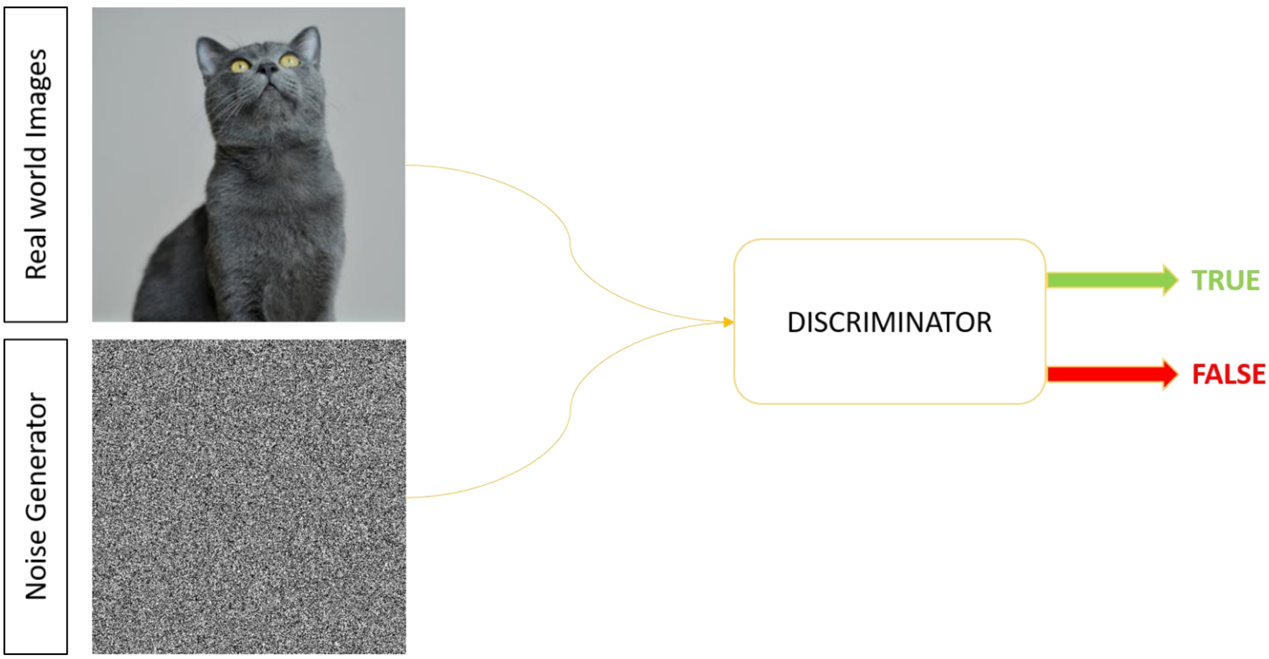
Introduction to LabVIEW

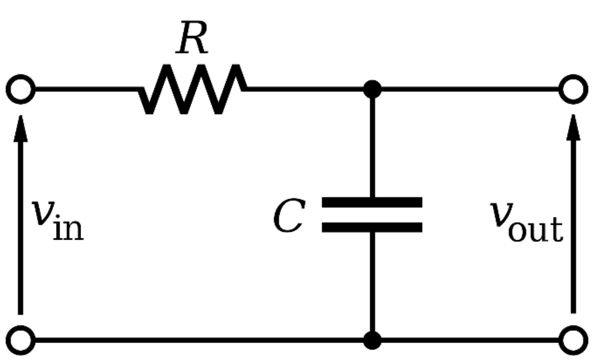
LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming environment which has become prevalent throughout research labs, academia and industry. It is a powerful and versatile analysis and instrumentation software system for measurement and automation. It's graphical programming language called G programming is performed using a graphical block diagram that compiles into machine code and eliminates a lot of the syntactical details. LabVIEW offers more flexibility than standard laboratory instruments because it is software-based. Using LabVIEW, the user can create precisely the type of virtual instrument needed, and programmers can easily view and modify data or control inputs. The popularity of the National Instruments LabVIEW graphical dataflow software for beginners and experienced programmers in so many different engineering applications and industries can be attributed to the software's intuitive graphical programming language used for automating measurement and control systems.

LabVIEW programs are called virtual instruments (VIs) because their appearance and operation imitate physical instruments like oscilloscopes. LabVIEW is designed to facilitate data collection and analysis, as well as offers numerous display options. With data collection, analysis and display combined in a flexible programming environment, the desktop computer functions as a dedicated measurement device. LabVIEW contains a comprehensive set of VIs and functions for acquiring, analysing, displaying and storing data, as well as tools to help you troubleshoot your code.

What is Noise?

Any measurement process involves a certain amount of noise or random fluctuations in the signal. Noises in measurements render the signals unusable for deducing any observation from the measurement. Hence removal of noise becomes a necessary process in any measurement.



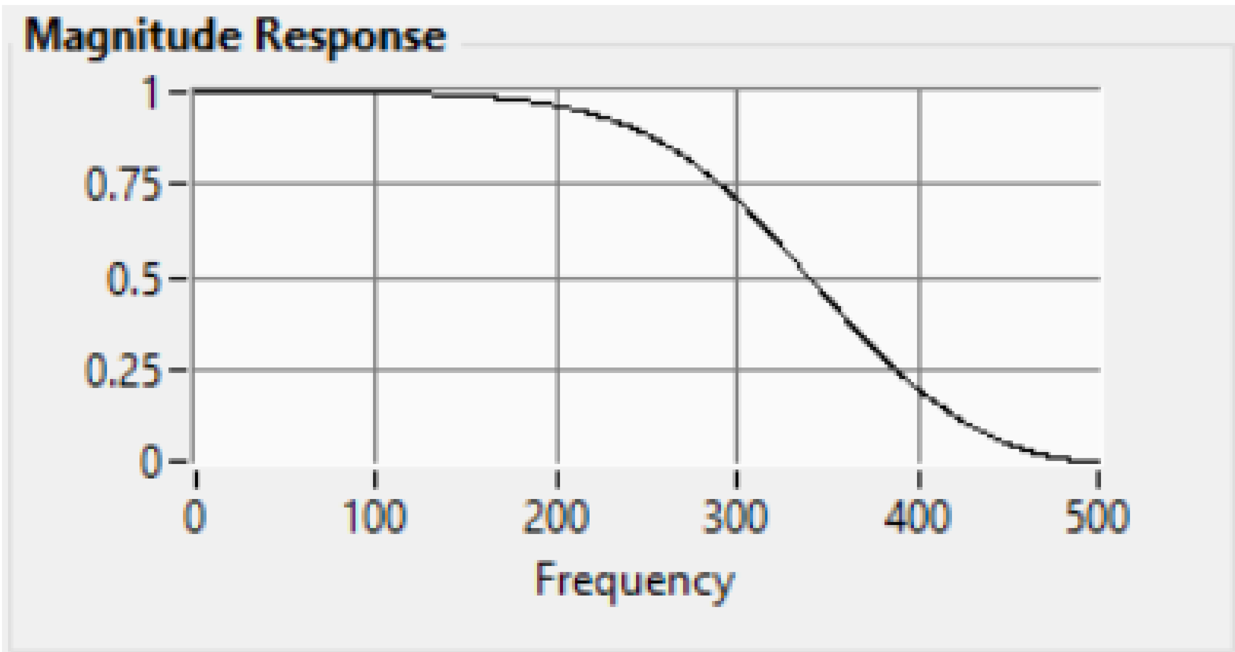


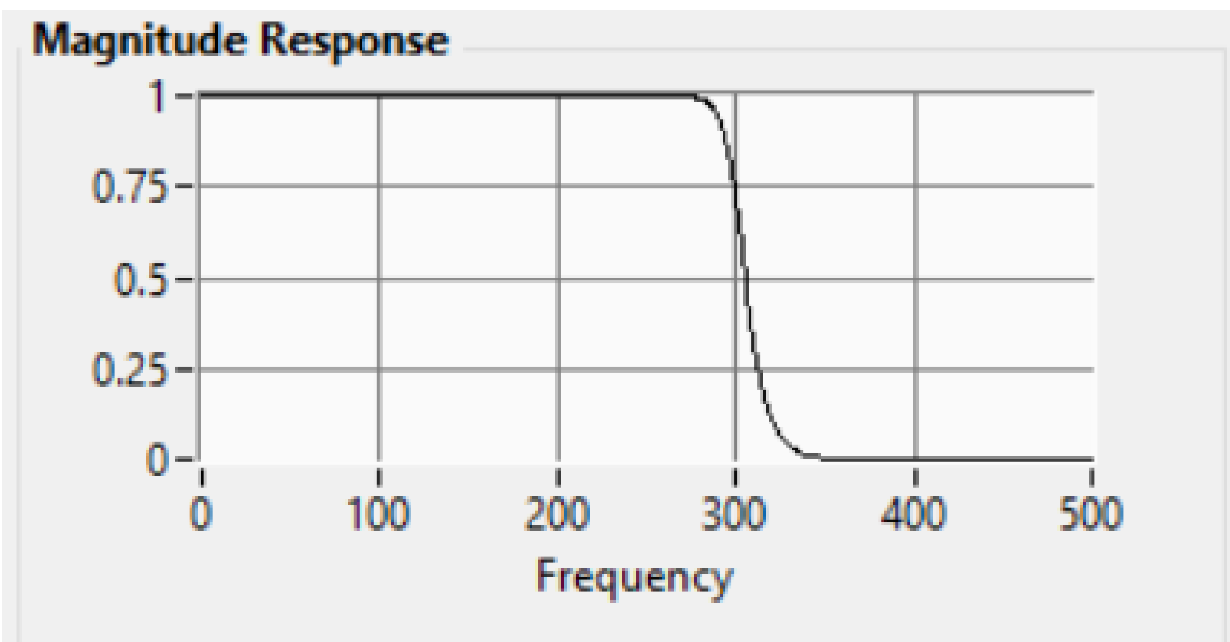
Simple analog filters such as RC filters are used for specific applications where the S/N ratio is significantly high. But this will not work for all the applications.

In many cases, these filters are not robust, and they can introduce new noise into the signal. The customizability is very limited. To overcome these issues, a **Digital Filter** is the obvious solution.

Digital Filters

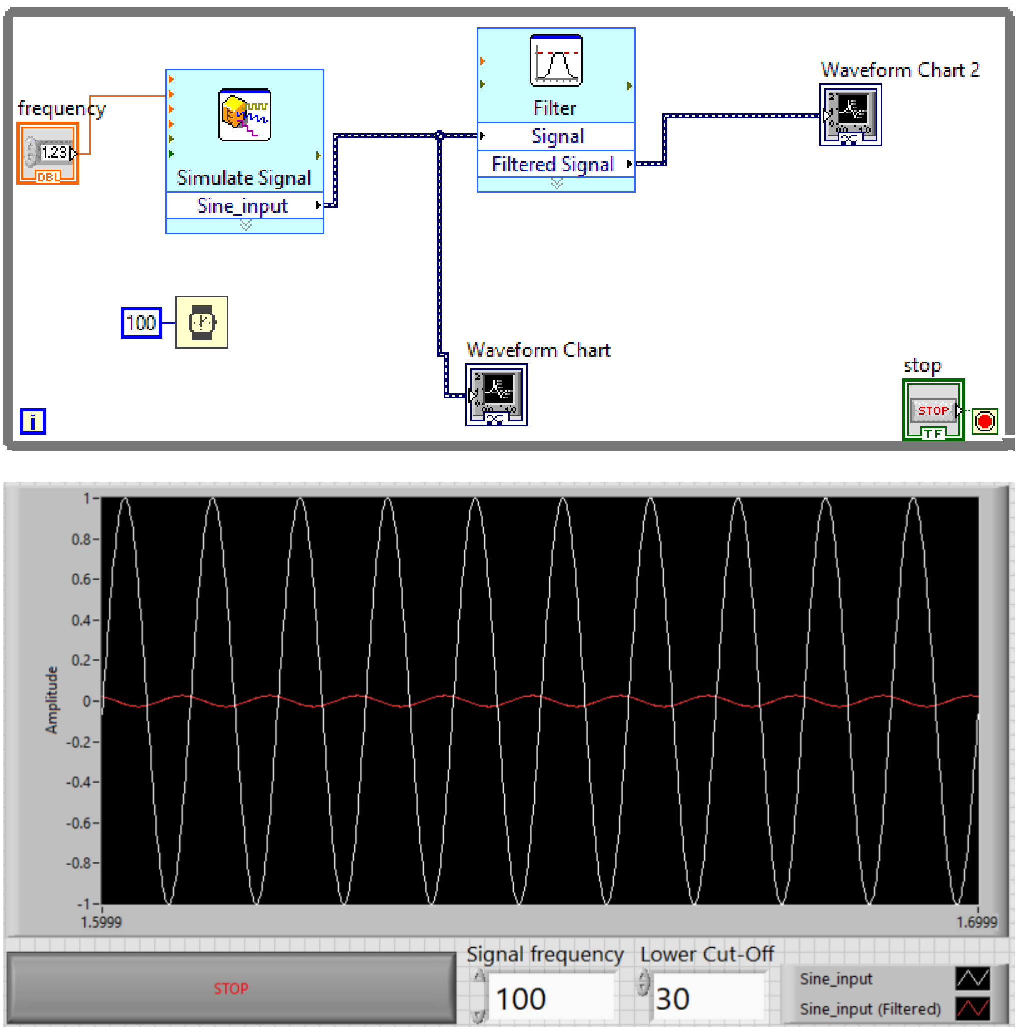
A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. The processor may be a general-purpose computer such as a PC, or a specialised DSP (Digital Signal Processor) chip. A digital filter has many advantages over an analog filter.





Advantages of a Digital Filter

- It is programmable and hence can be modified without affecting the entire circuit
- It is not susceptible to drift or temperature variations unlike analog filters
- Ability to handle complex combinations of filters in parallel or cascade making the hardware requirements relatively simple and compact
- A single filter can be replicated any number of times within a program without the need of any additional hardware, hence extremely cost-effective.
- They can be programmed to a have a steep frequency response to incredible precisions and dynamic reserves

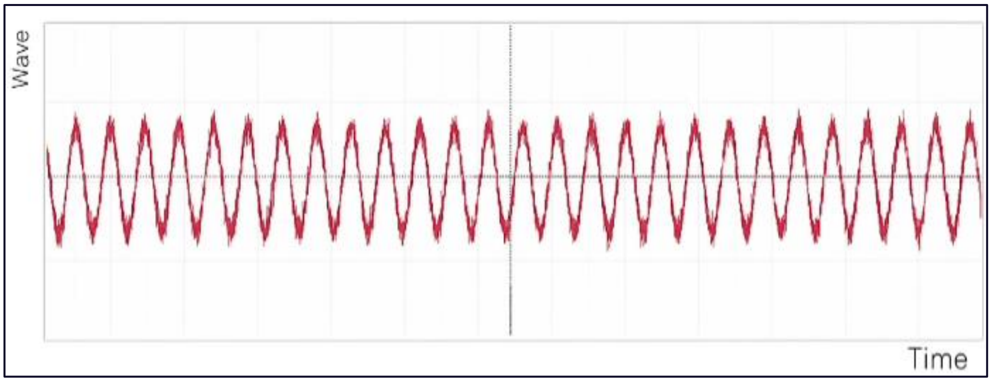


A Simple Low pass circuit that I designed in LabVIEW

Although a simple digital low pass filter may be sufficient for some applications, certain advanced filters are required when the noise is much higher than the signal

Filtering & the Fourier Transform

Noisy signal



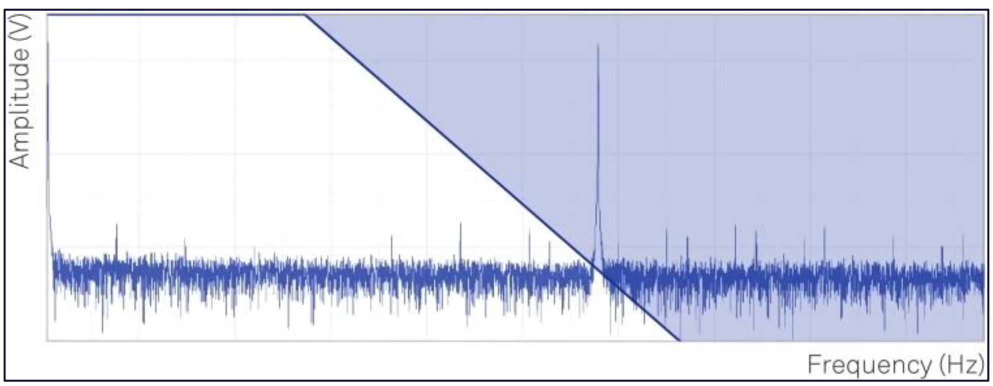
$$X_a(\omega) = \int_{-\infty}^{\infty} x_a(t)e^{-j\omega t}dt$$

Fourier Transform

Going from time domain to frequency domain

Low-pass filtering

Remove any unwanted frequencies

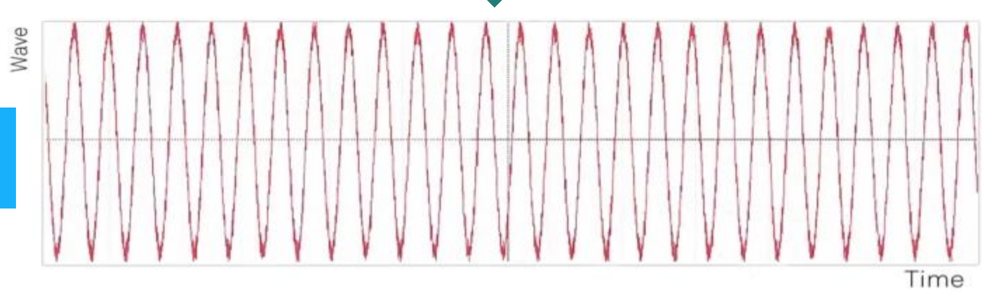


$$x_a(t) = \int_{-\infty}^{\infty} X_a(\omega)e^{j\omega t}d\omega$$

Inverse Fourier Transform

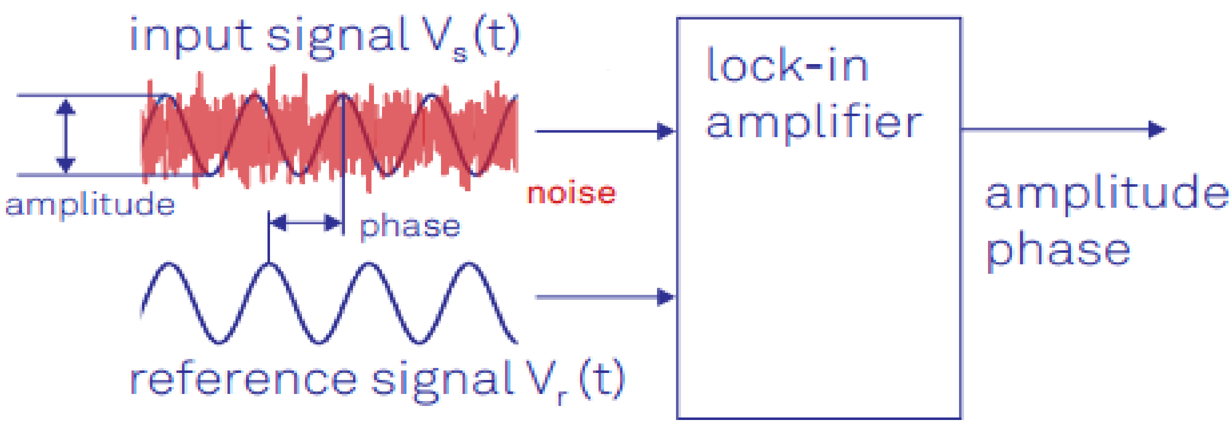
Going from frequency domain back to time domain

Filtered signal



Lock-in Amplifier – An Advanced Filter

Lock-in amplifiers were invented in the 1930s as electrical instruments capable of extracting signal amplitudes and phases in extremely noisy environments. Depending on the dynamic reserve of the instrument, signals up to 1 million times smaller than noise components, potentially reasonably close by in frequency, can still be reliably detected.



They employ a homodyne detection scheme and low-pass filtering to measure a signal's amplitude and phase relative to a periodic reference. A lock-in measurement extracts signals in a defined frequency band around the reference frequency, efficiently rejecting all other frequency components

Mathematically, a Lock-in amplifier can be written as

$$\int_0^T Asin(w_1 t) * Bsin(w_2 t)dt = 0 \qquad \forall\ w_1 \neq w_2$$

$$\int_0^T Asin(w_1 t) * Bsin(w_2 t)dt = \frac{1}{2} A * B \quad w_1 = w_2$$

They are capable of accurately measuring a signal in the presence of noise up to a million times higher in amplitude than the signal of interest that is an S/N of 10^{-6}.

Signal Mixing

An essential part of signal analysis is the mixing of two signals. The process of mixing two signals produces a signal with the difference and sum frequencies. Mathematically it can be shown as

$$X_1 = A_1 Cos(w_1 t + \emptyset_1) = \frac{A_1}{2}\left(e^{i(w_1 t+\emptyset_1)} + e^{-i(w_1 t+\emptyset_1)}\right)$$

$$X_2 = A_2 Cos(w_2 t + \emptyset_2) = \frac{A_2}{2}\left(e^{i(w_2 t+\emptyset_2)} + e^{-i(w_2 t+\emptyset_2)}\right)$$

$$X_1 X_2 = \frac{A_1 A_2}{4}\begin{pmatrix} e^{i((w_1+w_2)t+\emptyset_1+\emptyset_2)} + \\ e^{-i((w_1+w_2)t+\emptyset_1+\emptyset_2)} + \\ e^{i((w_1-w_2)t+\emptyset_1-\emptyset_2)} + \\ e^{-i((w_1-w_2)t+\emptyset_1-\emptyset_2)} \end{pmatrix} = \frac{A_1 A_2}{2}\begin{pmatrix} Cos((w_1 + w_2)t + \emptyset_1 - \emptyset_2) \\ + \\ Cos((w_1 - w_2)t + \emptyset_1 - \emptyset_2) \end{pmatrix}$$

Hence, after signal mixing the output signal has the sum and difference frequencies. Considering a special case where the reference signal is the same as the input signal,

$$\boldsymbol{For\ w_1 = w_2 = w}$$

$$\boldsymbol{X_1 X_2 = \frac{A_1 A_2}{2} Cos(\emptyset_1 - \emptyset_2) Cos(2wt + \emptyset_1 - \emptyset_2)}$$

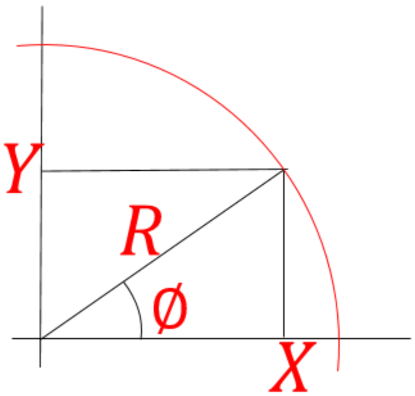
$$\boldsymbol{\langle X_1 X_2\rangle_t = \frac{A_1 A_2}{2} \langle Cos(\emptyset_1 - \emptyset_2)\rangle_t \, \langle Cos(2wt + \emptyset_1 - \emptyset_2)\rangle_t}$$

$$\boldsymbol{\langle X_1 X_2\rangle_t = \frac{A_1 A_2}{2} Cos(\emptyset_1 - \emptyset_2)}$$

After time averaging, we get an output which is proportional to the phase difference of the input signals. This forms the basis for phase-sensitive detection.

Principle of Phase-Sensitive Detection

Graphically, a phase-sensitive detection can be modelled as an input signal multiplied with a reference using a signal mixer and then passed through a low pass filter.

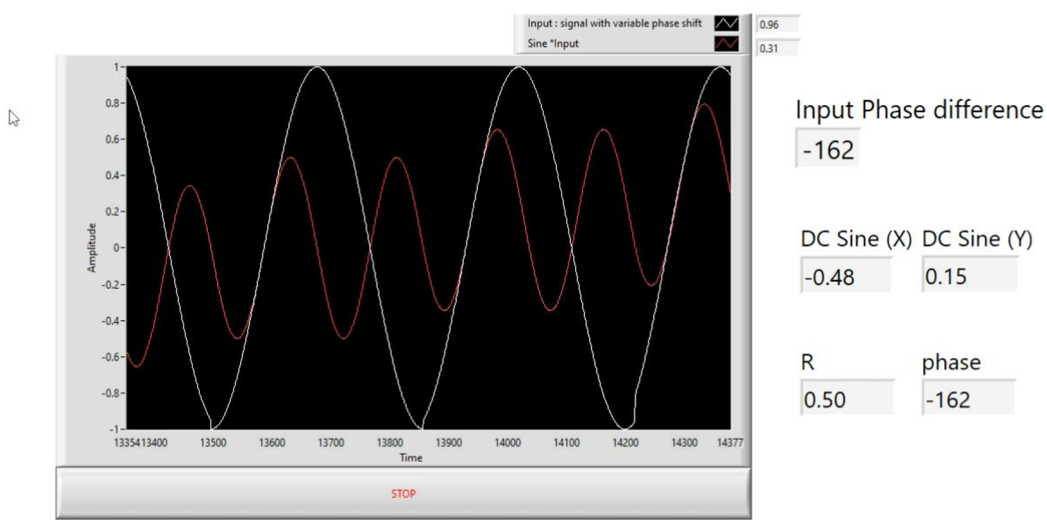


The concept of Phase-sensitive detection is better understood by considering the signal as a complex phasor with real and complex parts

- Using a cosine reference signal outputs the real part of the signal.
- Using a sine reference signal outputs the complex part of the signal.

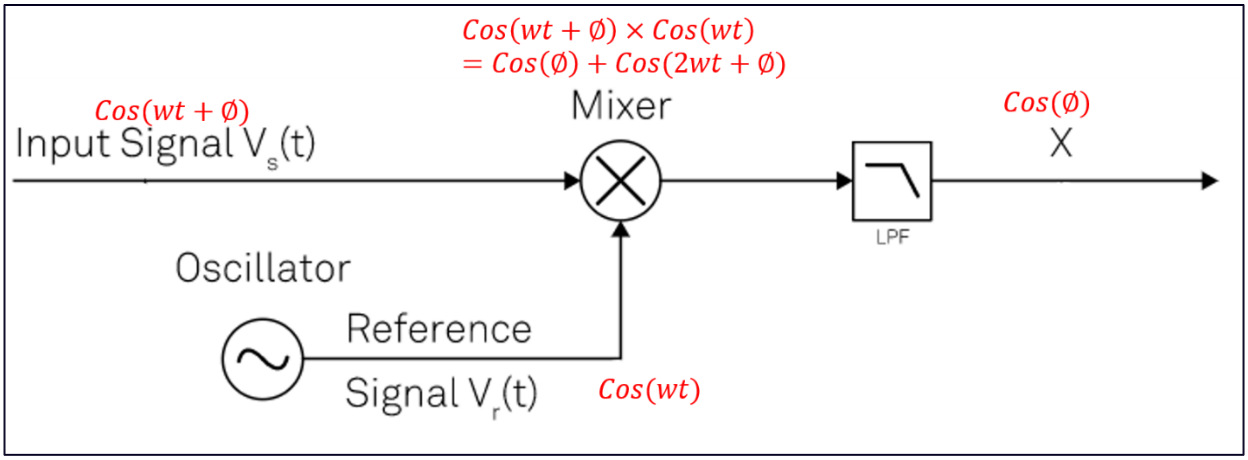
Signal multiplication of input with a reference signal is also sometimes called down-mixing or heterodyne detection.

Video:

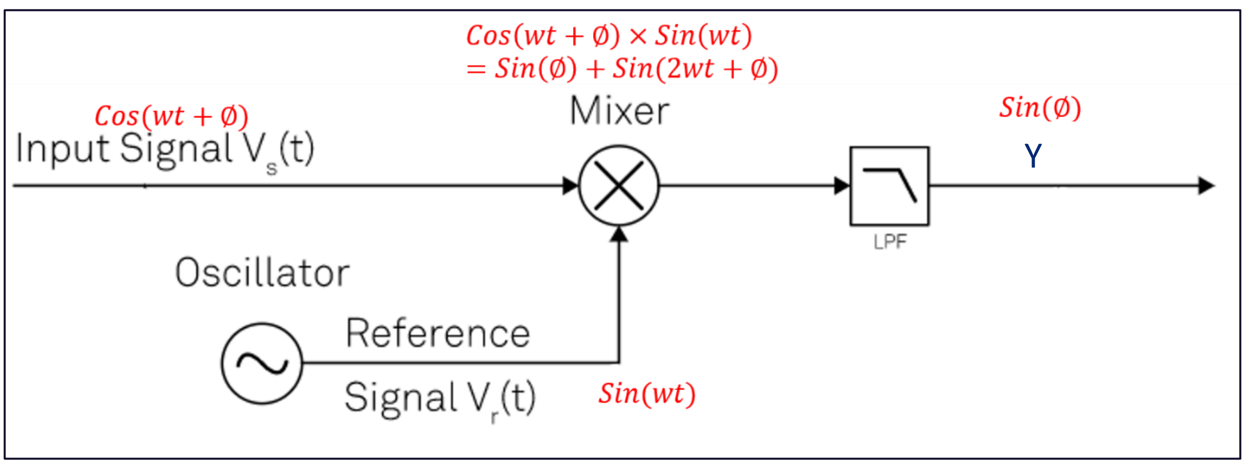


A LabVIEW program I designed which extracts the phase information from the input signal by heterodyne detection

For complete information about the signal, both real and complex (Quadrature) parts of the signal must be extracted. Hence both sine and cosine reference signals are used



The Real Part



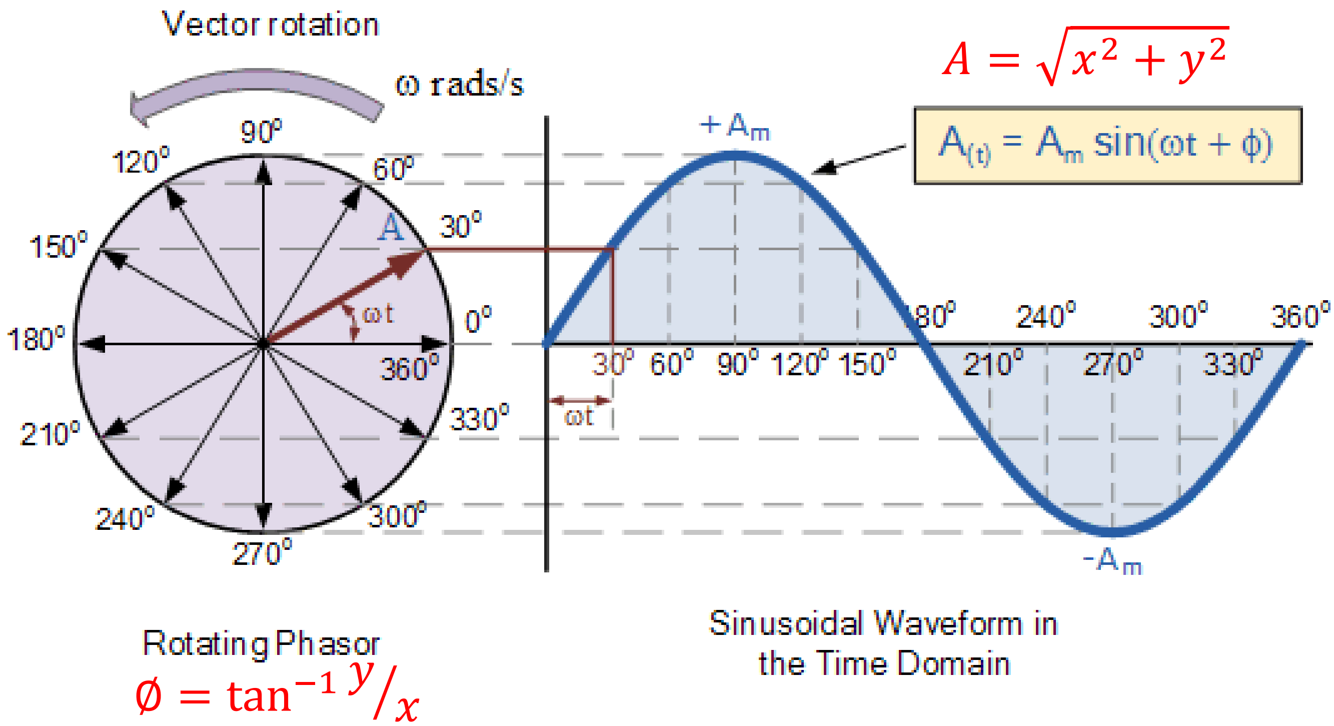
The Quadrature part

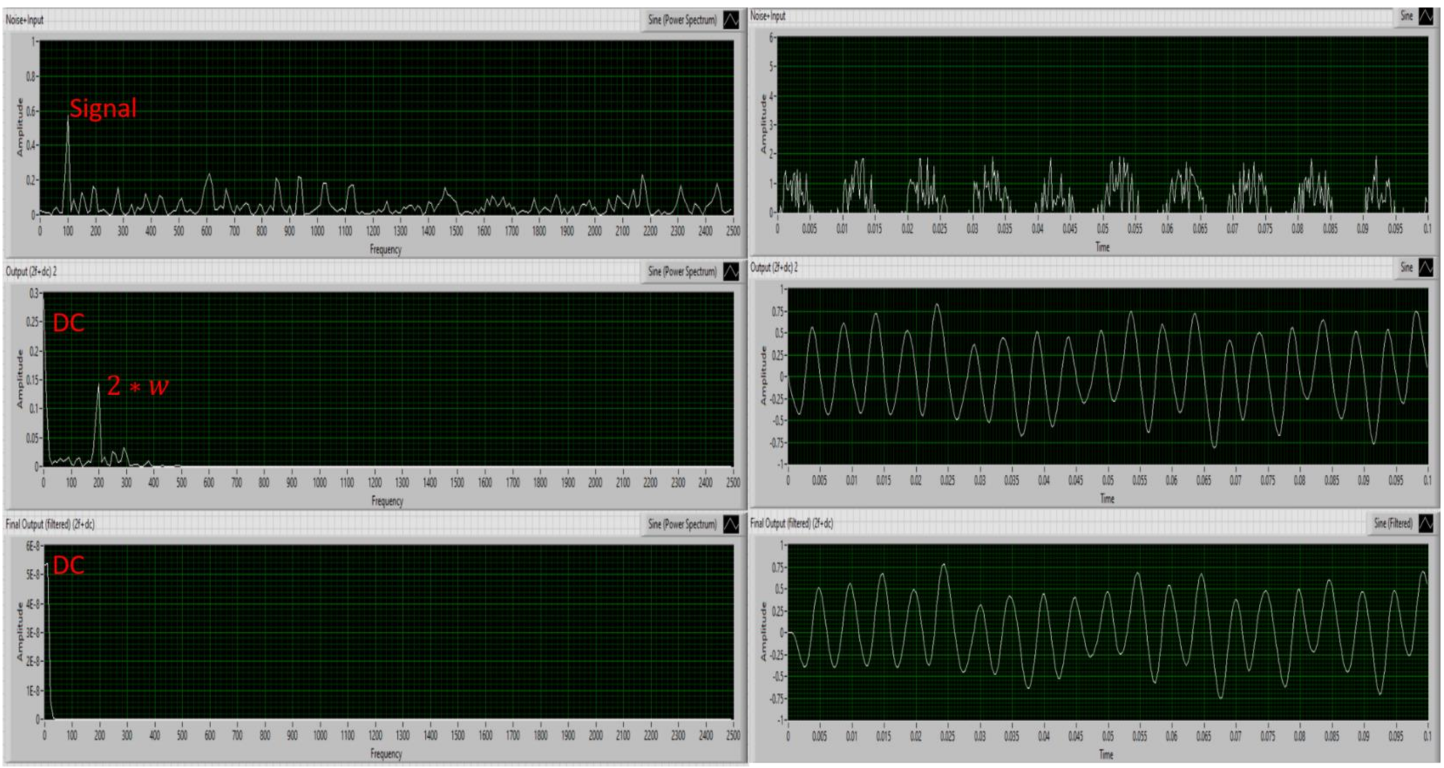
The real (X) and quadrature (Y) components thus obtained are used to reconstruct the noise-free signal by converting from the Cartesian argand plane to polar coordinates using the formula:

$$A = \sqrt{x^2 + y^2}$$

$$\emptyset = tan^{-1} {}^{y}/_{x}$$

Phasor Diagram

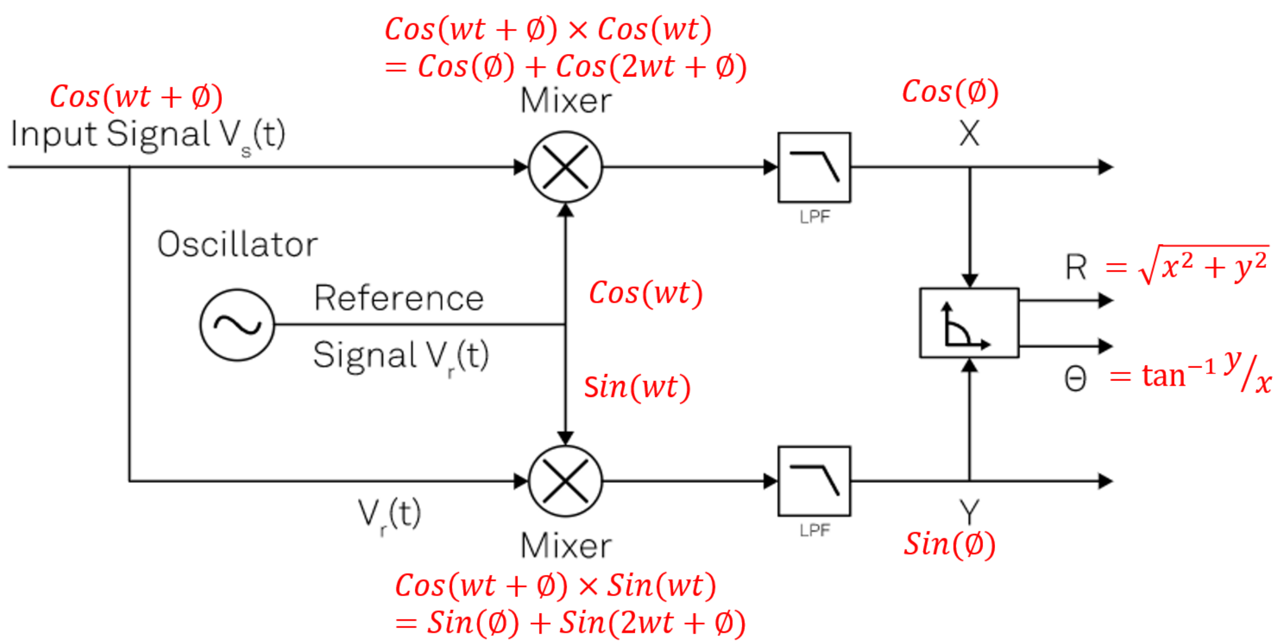


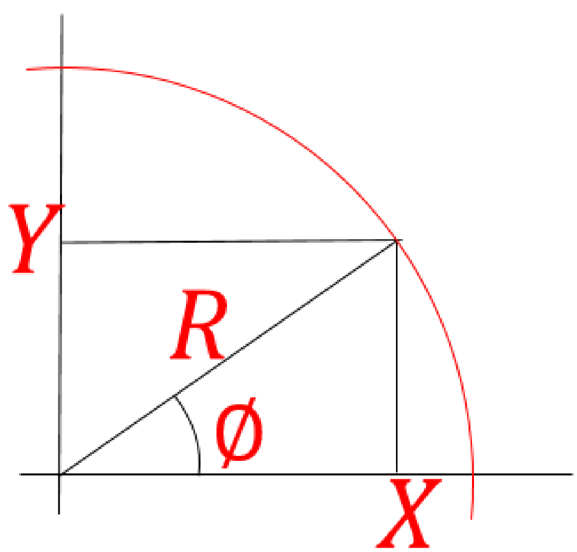


A LabVIEW program I designed to display the heterodyne detection process in time and frequency domain for a 100Hz signal with noise

Dual-Phase Lock-in Amplifier

The complete Lock-in detection involves simultaneous demodulation of both components and reconstruction in the polar coordinates as a complex phasor.



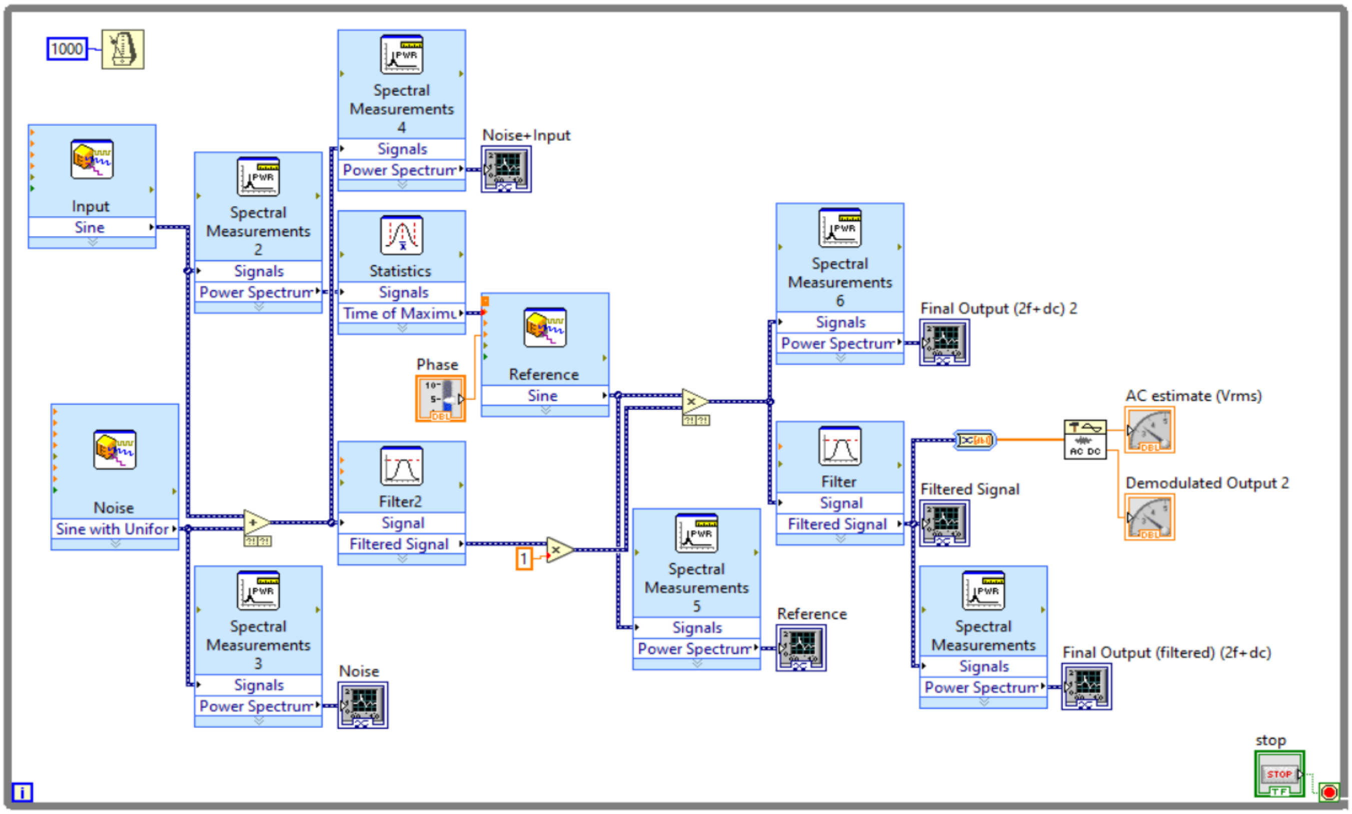


The input signal is multiplied by the reference signal and a 90° phase-shifted version of the reference signal.

The mixer outputs are low-pass filtered to reject the noise and the 2ω component, and finally converted into polar coordinates.

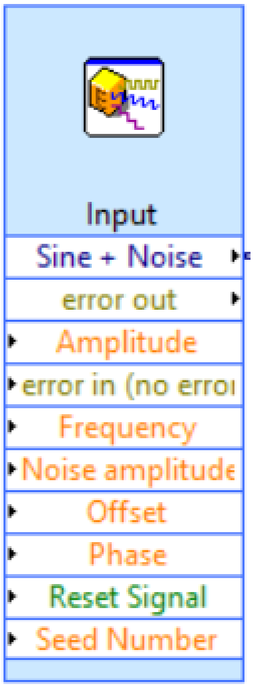
Digital Lock-in Amplifier

A digital version of the lock-in amplifier will have all the components in the above block diagram.



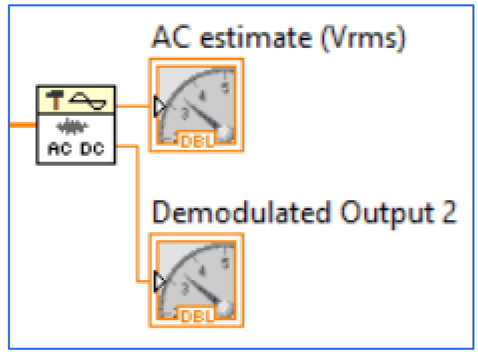
Lock-in Amplifier LabVIEW Program

- A sine signal is generated by the "Simulate Signal express vi" which is then added with some noise to create the input signal.
- This input is then mixed with a reference signal of variable phase, which produces a signal of sum and difference of the frequencies, that is the DC and 2w signals.
- A 3rd order Butterworth filter is then applied with a cut-off of 1Hz to filter out the 2w component leaving the DC component in the output.
- This signal is then demodulated to give the phase information of the input.
- The "Spectral Measurements express vi" along with a waveform graph is used to visualise the signal in the time and frequency domains during the process



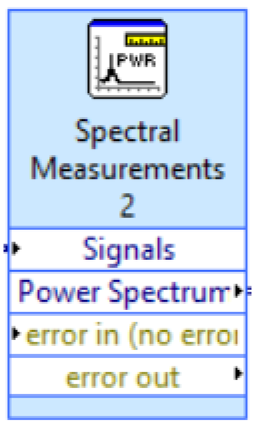
Simulate signal express VI

Generates different waveforms like sine, square, and triangle with or without noise at a particular frequency, amplitude, phase and offset.



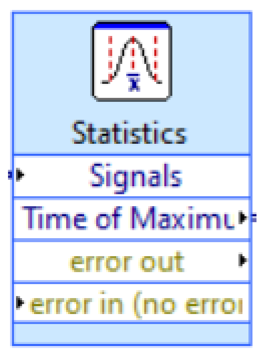
AC DC demodulator

Demodulates the input waveform and gives the corresponding AC and DC values



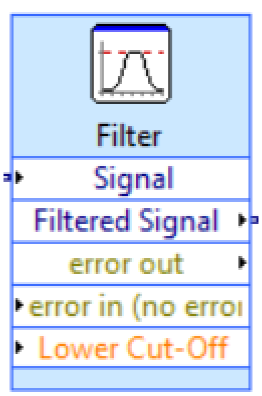
Spectral Measurements

Does a Fourier transform of the input signal and gives an amplitude vs frequency plot



Statistics

Provides various attributes of the input waves like the maximum, minimum, range, the corresponding time values, RMS etc.



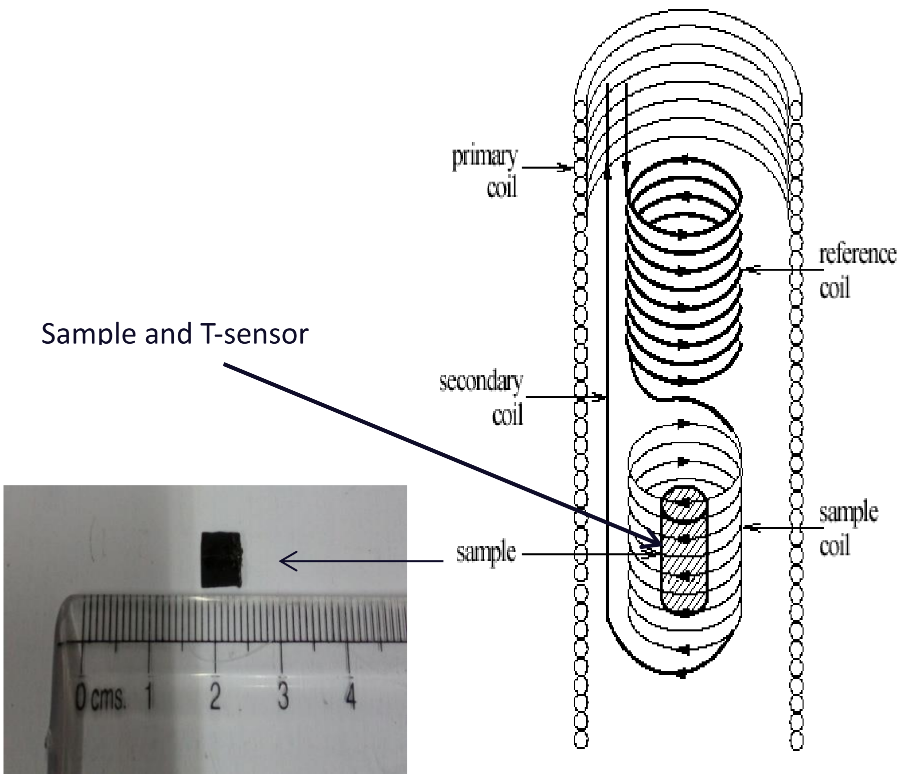
Filter VI

Applies an IIR (infinite impulse response) 3rd Butterworth low-pass filter to the input signal at the given cut-off frequency

Applications – AC Susceptometer

A Lock-in amplifier has various applications in almost any field since acquiring and processing signals is a common task in any measurement.

In the project, I have made a case study for the application of a lock-in amplifier for an AC susceptometer. AC Susceptometer is used to measure changes in the magnetic field due to the presence of a magnetic sample. A schematic of the instrument is shown here. The primary coil induces a voltage in the secondary coil. But since the secondary coil is



a series connection of a clockwise wound and anticlockwise wound copper coil, the net voltage is zero, and hence the net magnetic flux in the secondary coil is zero. But when a small magnetic sample is introduced in one of the coils, the net flux changes. This change is flux is usually buried in noise and needs an instrument like a lock-in amplifier to extract the signal.

Theory of AC Susceptometer

$$B_a = \mu_0 H_a$$

$$B = \mu_0(H_a + M)$$

$$M = \chi H_a$$

$$B = \mu_0(1+\chi)H_a$$

paramagnetic materials, $\chi > 0$

diamagnetic materials, $\chi < 0$

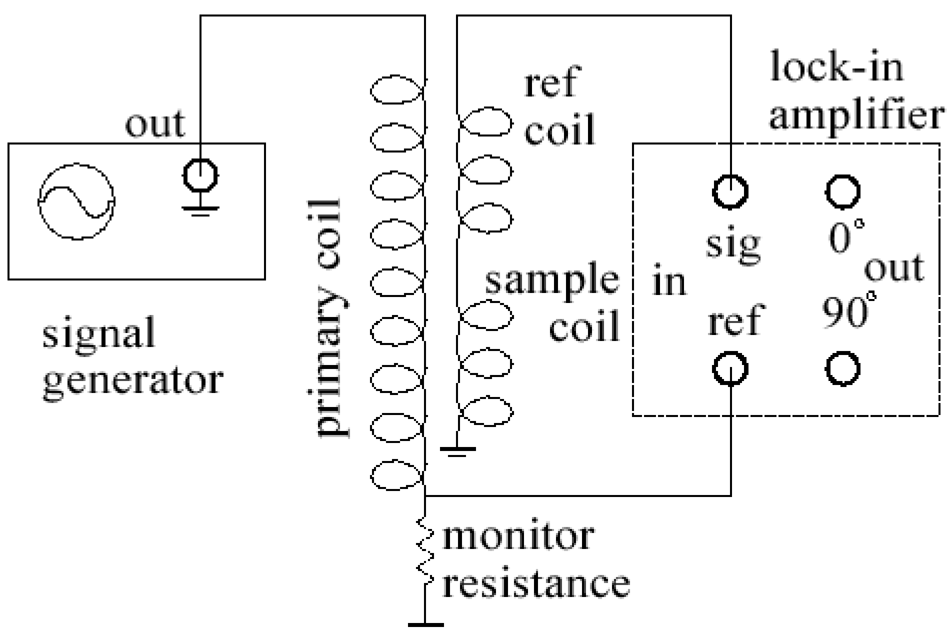
$$H_a = H_0 \cos \omega t$$

$$M = H_0(\chi' \cos \omega t + \chi'' \sin \omega t)$$

$$e^{i\omega t} = \cos\omega t + i \sin\omega t$$

$$\chi = \chi' - i\chi''$$

Since the current in the primary is changing sinusoidally, the magnetisation of the sample also varies sinusoidally. The frequency of this is the reference signal.



Tasks Achieved

- Learning LabVIEW programming
- Simulating a sine wave in LabVIEW
- Designing a program for observing signal mixing
- Mathematics of signal mixing
- Extracting phase information from signal mixing
- Mathematics of different filters
- Simulating a low-pass filter in LabVIEW and generating a bode plot
- Mathematics of lock-in amplifiers
- Case study of AC susceptometer as an application