Lock -In Amplifier based on Virtual Instrumentation

Author: Gemma Ballarà Madrid.
Advisor: Jose Miguel Asensi Lopez
Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.*

Abstract: When a slowly varying voltage is detected the main problem is the noise especially when it is several times greater than the signal itself. A lock-in amplifier is used to measure the small signals in presence of noise. The aim of this paper is to do a digital lock-in amplifier based on virtual instrument language *LabVIEW* and later on, it will replace a practice of a subject from the Physics' degree as a very useful teaching tool. This project is distributed with two experimental parts; the first one works by simulating the signals, reference and input signal, having some parameters to select, even the option if noise is wanted for the simulated input signal. The second one captures signals from external sources making it more realistic.

I. INTRODUCTION

In many areas of technology it is common to have to work with low-level signals. These signals are affected by many sources of noise, such as from the power network, the thermal noise, also, noise from sensors or amplifiers, … [1]. However, to remove equipment's turbulences a lock-in amplifier is used.

The main feature of the Lock-in amplifier is the Phase-Sensitive Detection (PSD). It consists of a multiplier, and then there is also a low pass filter which filtrates off signals that do not have the same frequency than the nominal one. Also it can be used to know the amplitude and phase of the output signal.

The experimental part is based on virtual instrument language *LabVIEW* [2] and it is separated into two programs, both programs use the theory of two-phase lock-in amplifier. The first one is based on a simulation in which the students can choose some parameters of the input signal and the reference signal, such as the amplitude, the frequency of each signal and the form and phase of the input signal. The other one is based on capturing both signals from an external source. However, the last program has generated many difficulties during the progress of the project and it could not have been completed on the expected time.

Also, there is a section where are shown some examples of exercises for the students based on the simulated program, for instance they can increase the input phase and they can explain why it is changing or another example is that they can understand the importance of the low-pass filter and which is the best constant time to get a DC component.

II. LOCK-IN AMPLIFIERS

The lock-in amplifiers can be divided into two groups depending on how the reference signal is used. On the one hand, there is the single-phase lock-in which works with one single PSD [3]; on the other hand, there is the two-phase lock-in which involves two PSDs, so the reference is shifted for the other PSD.

This paper is focused on the two-phase lock-in group. As shown below, its definition is given by using sinusoidal signals for both. It is also valid for other forms, as shown in section IV and the difference will be in the amplitude term and it will not change the operations done with PSD, but it has to be clear to keep always the same frequency between the two signals [4].

Therefore, having a sinusoidal signal defined as the input signal, it has the following form:

$$V_{\mathrm{i}}(t) = V_{\mathrm{i}} \sin(\omega_i t + \varphi_i) \tag{1}$$

where ω_i is the angular frequency, φ_i is the initial phase angle of the input signal and V_{i} is the input amplitude.

However, the amplifier needs a reference signal. This signal must have the same frequency as the input signal which is assumed later on. Apart from that, another reference signal is needed, this one is obtained by shifting the phase of the reference signal previously defined. So, these two signals have the following form:

$$V_{R1}(t) = V_{R1} \sin(\omega_R + \varphi_R) \tag{2}$$

$$V_{R2}(t) = V_{R2} \sin(\omega_R - \varphi_R + \pi/2) \tag{3}$$

Once all signals are defined, the main operation the lock-in amplifier does, using the PSD, is multiplication. Therefore, the output signal of the PSD, taking the first reference signal is:

$$V_{out1}(t) = V_i(t) \cdot V_{R1}(t) = V_{\mathrm{i}} \sin(\omega_i t + \varphi_i) \cdot V_{R1} \sin(\omega_R + \varphi_R) \tag{4}$$

Then, using basic trigonometrical functions, the output signal from PSD has this form:

$$V_{out1}(t) = \frac{1}{2} V_i V_{R1} [\cos([\omega_i - \omega_R]t + \varphi_i - \varphi_R) - \cos([\omega_i + \omega_R]t + \varphi_i + \varphi_R)] \tag{5}$$

The output is an AC signal with two different frequencies, the sum and the rest of the input and reference frequency. Assuming $\omega = \omega_i = \omega_R$, as it is said before, the output signal has a DC component.

$$V_{out1}(t) = \frac{1}{2} V_i V_{R1} [\cos(\varphi_i - \varphi_R) - \cos(2\omega t + \varphi_i + \varphi_R)] \tag{6}$$

Using a low-pass filter the output signal will have only the DC component by filtering the AC component. Finally, the output signal is obtained:

* Electronic address: gballama10@alumnes.ub.edu

$$V_{out1} = \frac{1}{2} V_i V_{R1} \cos\varphi \approx V_i \cos\varphi = X \qquad (7)$$

where $\varphi = \varphi_i - \varphi_R$.

Doing the same steps with the shifted reference signal, the second output signal is:

$$V_{out2} = \frac{1}{2} V_i V_{R2} \sin\varphi \approx V_i \sin\varphi = Y \qquad (8)$$

So, there are two outputs. The first output is called the synchronous in-phase component, X, and the other is called the quadrature, Y. These outputs permit to calculate the amplitude of the signal, it is determinate independently on the phase shift, and the phase shift:

$$V_i = \sqrt{X^2 + Y^2} \qquad (9)$$

$$\varphi = \tan^{-1}\left(\frac{Y}{X}\right) \qquad (10)$$

Therefore, knowing all the steps followed in order to obtain these results, it can be summarized in a block diagram as it is displayed in the following picture:

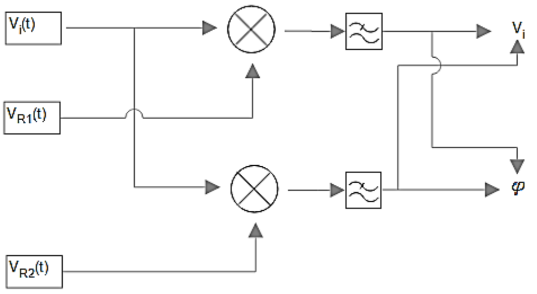

FIG. 1 Schematic diagram of a two-phase lock-in amplifier

However, the explanation of why it is assumed that the frequencies must be equal is that, if it is considered that the frequencies are quite different, then the output signal is going to have equally two AC components and using the low-pass filter, this signal is going to disappear.

III. OVERVIEW OF THE PROGRAM

Considering the previous theory, a digital two phase lock-in amplifier has been developed. It is based on virtual instrument language *LabVIEW* and it is divided into two programs, as it is said before, the first one is simulated [5] and the second is more realistic.

Therefore, this section explains in general terms how each program work. *LabVIEW* has two windows, first the front panel and then the block diagram windows. The front panel is the user interface window, where the program runs and shows the results. This window can be done like a physic instrument (that is why *LabVIEW* programs are called virtual instruments). The block diagram is where the user can add code using graphical representations in order to control the front panel objects.

A. Simulated program

The first program explained is the one that produces the signals by itself. The main purpose of it is to use it as an educational tool.

Regarding the front panel, the first steps that should be set are the necessary parameters of the input signal. These parameters are: the frequency, the amplitude, the form, the phase of the input signal and also the noise. In case that noise is wanted, the standard deviation can be set. Moreover, the front panel lets the reference signal frequency be customized, as it has been said, it is an educational tool but if it was not, the frequency should be the same one that the initial one, without being able to update it.

Once the characteristics of both signals are chosen, it is only needed to modify the time constant variable of the low-pass filter. These filters allow to choose the cut-off frequency and also, they allow to choose the order of the filter (order 1 or order 2). In order to do this, a Butterworth filter should be used.

Finally, when all the parameters are chosen, the program runs and a graphic is displayed with six signals. These signals are: the input, the reference, the real part (X), the imaginary part (Y), the amplitude and the phase shift. One option that can be set on the graphic is to be able to choose which signal is wanted to represent, disabling the other signals with boolean buttons that appear at the legend. A part from displaying the signals with the graphic, there are four indicators that give the result of the real part (X), the imaginary part (Y), the amplitude and the phase shift on each time and it makes reading the value from the graphic easier.

To sum up, the picture below shows the frontal panel with the parameters that can be changed and the graphic. It is wanted to look like an oscilloscope.

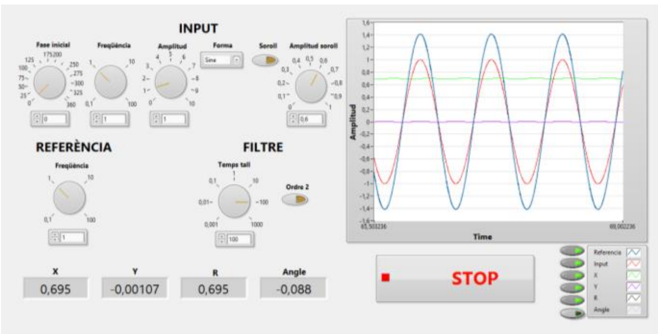

FIG. 2 The front panel of the simulated program

In order to develop the program, Express functions have been used in order to reduce the difficult level that represents the construction of a filter from the scratch or a signal (with random noise if it was wanted).

So, the block diagram of the simulated program is very easy. It consists in three Express functions designed to simulate each signal. But these functions only permits to modify some parameters. For this reason each function, or also called *subVI*, has been internally modified in order to update more of the permitted values. Next step has been done simply by using the multiplication function where the signals are multiplicated and then they pass through a low-pass filter, which is a Butterworth filter, also done with an Express function. On the one hand, out of the filter, all the signals are represented on a graphic and, using Boolean controls, it can be chosen which graphic can be displayed. However, some commands can be disabled, for instance the frequency, in order to avoid affecting the synchronism of the signals once the program is running, this has been done by using Property Node function. This function is used to read or write (in our case it is a write function) properties from local applications like a graphic. For last, there is a *STOP* button whose functionality is to stop the while loop and consequently the program.

In the future, if it is wanted to be improved, this simulation has not got a *PLAY* button to run the program when it is wanted; this button is very useful when the program is compiled into a runnable program. A possible problem could be the synchronism that must have with the *STOP* button and the other disabled commands.

Another improvement should be to introduce more steps to the reference signal, for example a PPL, and also to introduce more steps to the input signal, for instance an input amplifier and low-pass filter before the multiplication. Moreover, an important improvement needed for the simulated program is that the response time of the output signal could be improved, because now it takes several seconds to stabilize.

B. Program with signals from external sources

This case is slightly different of the first one. Both signals are catched by the outer sources, as generators or circuits, instead of internally generated (all the parameters cannot be chosen before).

Therefore, the front panel has been tried to get the same structure than the previous program but deleting the determined parameters. So, it can only be chosen the time constant of the low-pass filters and also, it can be selected, as before, which graphic can be used to represent the result.

Based on the block diagram, adding some extra blocks has been required in order to open the signals and being able to update them during the process.

However, this program has generated some difficulties and unexpected situations that have slowed the progress of this part. One of the main problems has been the synchronization of both signals, it is required a PLL (not used at the last program) in order to being able to enable the synchronization of the reference signal with the input one. Also, there have been more unexpected troubles, because both signals are not “perfect”, and the program cannot run because these errors could not be solved.

Even once obtained permissions from National Instruments to install the program, dedicated some time to understand the theme and using more time in doing simulations, the goals have not been reached. The difficulties of the non-perfect signals and the ignorance of some electronic themes makes impossible to assume the amount of work that represents that problem.

IV. EXPERIMENTAL RESULTS

A. Examples of practical lesson

This project is going to replace one practical lesson from Physics’ subject where the aim of this practice is to help the student to understand the lock-in amplifier by doing some exercises. So, these exercises that can be useful to do with the simulated program are the following ones.

The first exercise consists on changing the initial phase of the input signal, taking values of 0º, 90º, 180º and 270º and explaining each result. Also, it can be done with different signal forms, for instance triangle, square or sine.

The result of this first exercise using a triangle signal is the following:

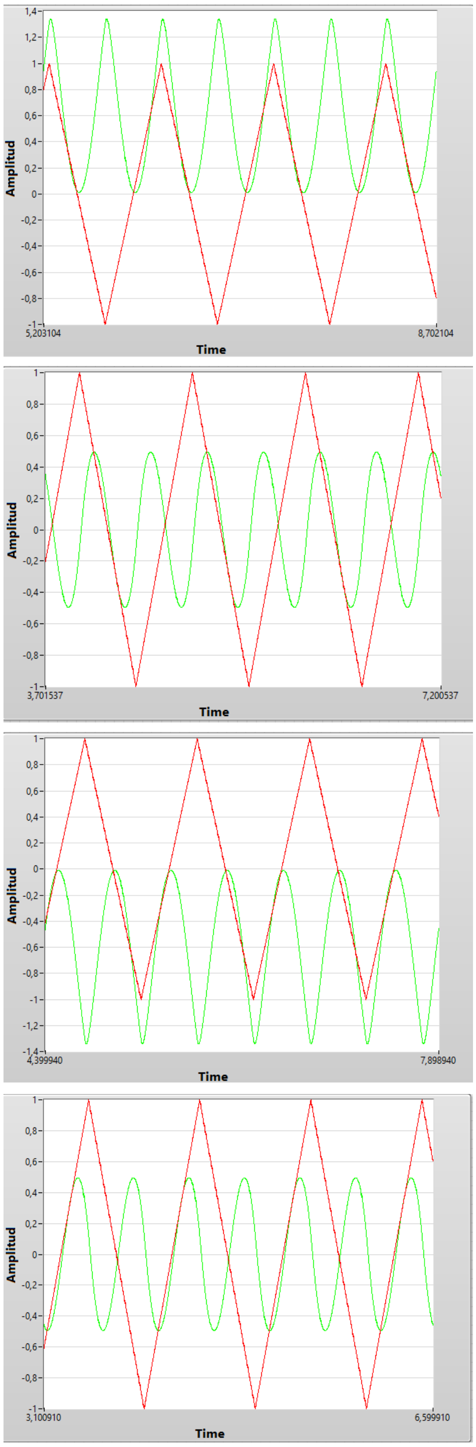

FIG. 3 Graphic from the front panel which shows V(t) for an initial phase of 0º, 90º, 180º and 270º respectively. In red the input signal and in green the output X signal.

The second exercise introduces the low-pass filter, so the student has to vary the time constant and then, he has to explain how a filter works. When the student increases the time constant, the output signal is more constant, so it is keeping its DC component and eliminating its AC component.

Now, the result is done by using a square signal and increasing in one order the time constant of the filter:

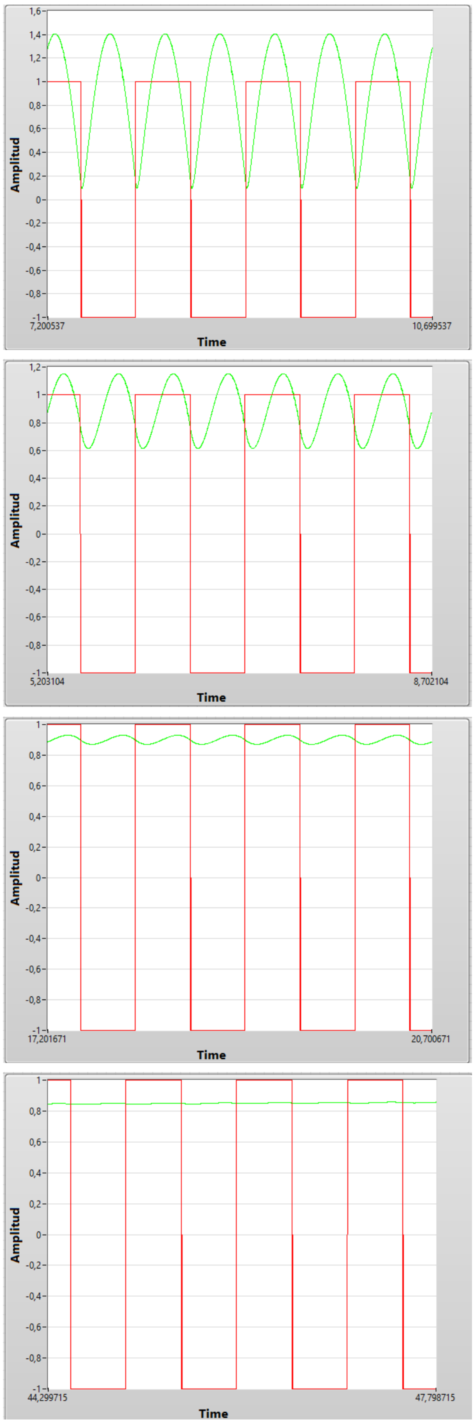

FIG. 4 Graphic from the front panel which shows V(t) for a constant time of 0.1, 1, 10, 100. In red the input signal and in green the output X signal.

To know exactly what the most useful time constant is, it is better to do a Bode diagram.

Another exercise introduces the noise, so the student can choose the standard deviation of input noise. To see that the lock-in amplifier is working well, the student can compare the input noise signal with the input signal without noise.

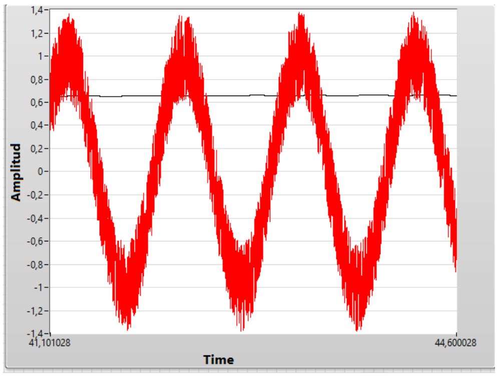

FIG. 5 Graphic from the front panel that shows input signal with noise, 0.35 standard deviation, and R value for 0º phase shift.

B. Testing exercises

To examine the system results, there are several tests to try to prove the theoretical behavior. Following are two tests to study the dependence of the output signal.

The first one consists in proving the dependence of the DC-output voltage on the phase shift. As it is shown in section II, output voltage X behavior is a cosine and output voltage Y behavior is a sine. So, the result of the digital lock-in amplifier is the following:

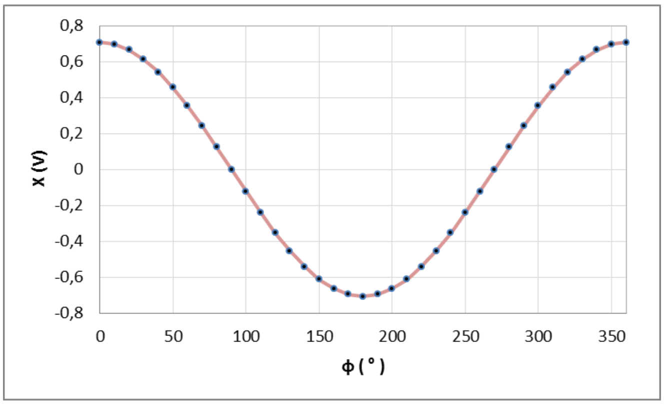

FIG. 6 Output voltage X versus phase shift

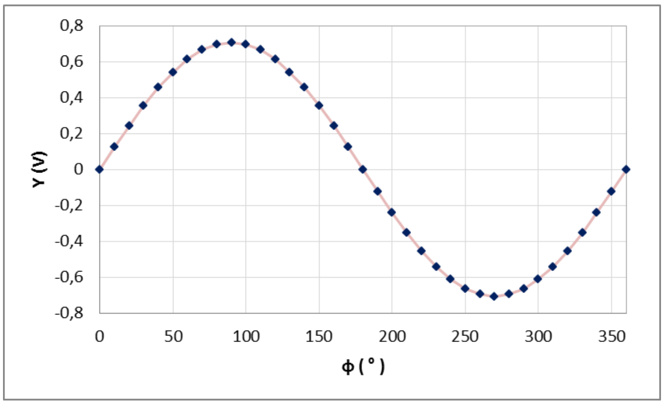

FIG. 7 Output voltage Y versus phase shift

Thus, observing the results it can be said that the result is satisfactory because the signals have the expected behavior.

Another test involves looking at how the output signal varies with increasing noise of the input signal. However, the time constant of the filter must be constant, in this case it has a value of 100s.

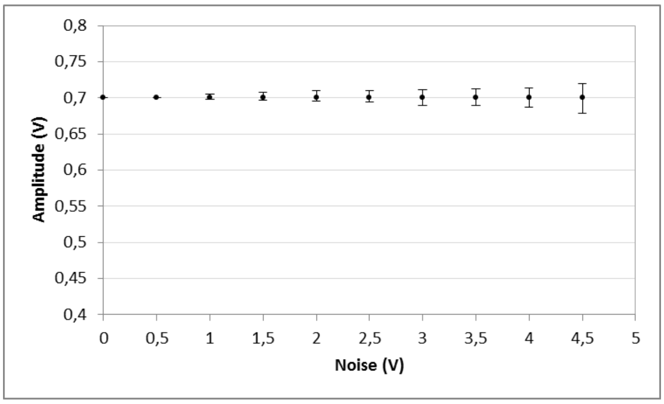

FIG. 8 Amplitude range value of the output signal when noise increases.

Analysing the result obtained on the graph, it can be seen that increasing noise amplitude increases imprecision. However, the observed error is reasonable and this can be improved by increasing the time constant of the filter. Also, it shows that the designed digital lock-in amplifier has a good anti-noise capability.

V. CONCLUSIONS

In this paper the theory of a two-phase lock-in amplifier has been studied and also, how to implement it with virtual instrument language *LabVIEW*. To know if the results are correct, there are two examples of evaluating the output signal. Also, there are some examples for using it as a teaching tool.

Therefore, using *LabVIEW* is very useful to evaluate the signals at every moment to see really how it is working.

Instead of one-phase lock-in amplifier a two-phase one has been implemented because the theory and the program are not as much difficult as it would seem and the result gives more information about the signal.

In future, it could be important to improve, on one hand, the response time of the simulated program and on the other hand, the PLL and other electronic difficulties of the other program.

Acknowledgments

I especially would like to thank my advisor Jose Miguel Asensi Lopez, who always had time to help me and also a lot of new ideas for the project. I would also like to show my gratitude to my friend J. O., who answered me some questions about the program and he always was ready to help me. Therefore, I want to thank my family for their unconditional support, especially to my brother.

[1] E. R. Gutiez, “Implementación software de un amplificador Lock-in.”, 2015

[2] P. Kromer, R. Robinett, R. Bengtson, and C. Hays, “PC-based digital lock-in detection of small signals in the presence of noise,” *AAPT Appar.* 1999.

[3] J. H. Scolfield, “Frequency-Domain Description of a Lock-in amplifier”, *American Journal of Physics* 62.2 (1994): 129-132 .

[4] P. Q. Trieu and N. A. Duc, “Implementation of the digital phase-sensitive system for low signal measurement,” *VNU J. Sci. Math. - Phys.*, vol. 24, pp. 239–244, 2008.

[5] B. Ye, F. Chen, and M. Li, “The Digital Lock-in Amplifier for Detecting the Power Traveling Wave Signal,” vol. 8, no. 4, pp. 361–374, 2015.