

# **EXPERIMENTS WITH LOCK-IN-AMPLIFIER USING EXPEYES**

## **(Calibration, Low Resistance and CV profiling)**

*Open Lab Report Submitted  
in Partial Fulfilment of the Requirements  
for the course*

**P442-Integrated Laboratory-II**

*by*

**ANANTHA PADMANABHAN M NAIR**

Course Instructors:

**Dr. Ashok Mohapatra**

**Dr. Gunda Santosh Babu**



*to the*

**School of Physical Sciences**

**National Institute of Science Education and Research**

**Bhubaneswar**

**March 4, 2024**

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Theory</b>	<b>1</b>
2.1	Fast Fourier Transform . . . . .	1
2.2	Calibration of the LIA . . . . .	2
2.3	Measurement of Low Resistance . . . . .	2
2.4	C-V Profiling of Solar Cell . . . . .	2
2.4.1	Autophasing . . . . .	3
<b>3</b>	<b>Experimental Setup and Procedure</b>	<b>3</b>
3.1	Building the Lock-in Amplifier . . . . .	3
3.2	Procedure . . . . .	4
3.2.1	Calibration . . . . .	4
3.2.2	Measurement of Low Resistance . . . . .	4
3.2.3	C-V Profiling of Solar Cell . . . . .	4
<b>4</b>	<b>Observations and Data</b>	<b>5</b>
4.1	Verification of working of the Lock in Amplifier . . . . .	5
4.2	Data for Calibration . . . . .	6
4.3	Data for Calculation of Low Resistance . . . . .	6
4.4	Data for C-V Profiling . . . . .	6
<b>5</b>	<b>Data analysis</b>	<b>6</b>
5.1	Calibration of the Lock in Amplifier . . . . .	6
5.2	Measurement of Low Resistance . . . . .	7
5.3	C-V Profiling of Solar Cell . . . . .	7
<b>6</b>	<b>Error Analysis</b>	<b>8</b>
6.1	Error in the Calibration . . . . .	8
6.2	Error in the Doping density and built in potential . . . . .	8
<b>7</b>	<b>Results</b>	<b>8</b>
<b>8</b>	<b>Conclusion and Discussions</b>	<b>9</b>
8.1	Conclusions . . . . .	9
8.2	Discussions . . . . .	9
8.3	Sources of error . . . . .	9
8.4	Future Prospects . . . . .	9

## List of Figures

1	Circuit Diagram for the Measurement of Low Resistance . . . . .	2
2	Circuit Diagram CV Profiling of Solar Cell . . . . .	2
3	Modified Circuit Diagram CV Profiling of Solar Cell . . . . .	5
4	Setup for the CV Profiling of Solar Cell (connected to calibration capacitor) .	5
5	Plot for verification of the working of the Lock in Amplifier function . . . . .	5
6	Plot for verification of the working of the Lock in Amplifier function with a real signal from SEELab . . . . .	6
7	Calibration curve of the Lock in amplifier . . . . .	6
8	CV Plot for polycrystalline solar cell . . . . .	7
9	Plot of $1/C^2$ vs $V_R$ for the polycrystalline solar cell . . . . .	8
10	CV Plot for monocrystalline solar cell . . . . .	8
11	Plot of $1/C^2$ vs $V_R$ for the monocrystalline solar cell . . . . .	8

## List of Tables

2	Data for the Calculation of Low resistance . . . . .	6
3	Results from the calibration curve . . . . .	6
1	Data for the calibration of the Lock In Amplifier . . . . .	7
4	Linear fitted parameters for mono and poly crystalline solar cells . . . . .	7

# EXPERIMENTS WITH LOCK-IN-AMPLIFIER USING EXPYES

## (Calibration, Low Resistance and CV profiling)

### Abstract

This experiment primarily centres on constructing a Lock-in amplifier using ExpEYES/SeeLAB, validating its functionality, and subsequently conducting various experiments employing the built apparatus. These experiments encompass calibration, low resistance measurement, and CV profiling of both monocrystalline and polycrystalline solar cells. The obtained results demonstrated close conformity with actual data. Furthermore, given the affordability and accessibility of the components utilized, this cost-effective approach to lock-in detection proves to be particularly suitable for instructional laboratories.

## 1 Introduction

A flexible open-source hardware and software framework called ExpEYES, which stands for "Experiments for Young Engineers and Scientists," is intended for use in physics, electronics, biology, and chemistry investigations. ExpEYES, created by the Inter-University Accelerator Centre in India, offers a practical and affordable approach to science teaching and experimentation. The lock-in amplifier operates on the basis of phase-sensitive detection, in which the target signal is efficiently isolated from noise by modulating it at a known frequency and then demodulating it at the same frequency.

## 2 Theory

A lock-in amplifier serves the purpose of detecting minute voltages amid noisy environments. This is accomplished through phase-sensitive detection, a technique employed by commercially accessible lock-in amplifiers. In order to amplify the signal from the noise, we must know the frequency of the signal that has to be separated.

In the lock in amplifier, we use a reference signal which has the same frequency as that of the noisy signal. We multiply the signal with the reference as well as a reference shifted by a phase of  $\pi/2$  and then we put a low pass filter to get the DC components and hence we get the amplitude of the corresponding signal. Let's see how that's implemented.

Let the signal that we want to measure be a sine wave of frequency  $\omega_{sig}$  and amplitude  $V_{sig}$ . Consider this is the signal that is embedded in the noise. This can be represented by:

$$V_{sig}(t) = V_{sig} \sin(\omega_{sig}t + \phi) \quad (1)$$

Let the reference signals be:

$$V_{ref}^0(t) = V_{ref} \sin(\omega_{ref}t) \quad (2)$$

$$V_{ref}^{90}(t) = V_{ref} \cos(\omega_{ref}t) \quad (3)$$

Multiplying equations (2) and (3) with (1), we get:

$$V_{sig}(t)V_{ref}^0(t) = V_{sig}V_{ref} \sin(\omega_{sig}t + \phi) \sin(\omega_{ref}t) \quad (4)$$

$$V_{sig}(t)V_{ref}^{90}(t) = V_{sig}V_{ref} \sin(\omega_{sig}t + \phi) \cos(\omega_{ref}t) \quad (5)$$

Let's call the former  $V^0(t)$ , the in-phase component and the latter one  $V^{90}(t)$ , as the quadrature component. So, after using trigonometric identities and setting  $\omega_{ref} = \omega_{sig}$  we get these components as:

$$V^0(t) = \frac{V_{sig}V_{ref}}{2} \cos(\phi) + \frac{V_{sig}V_{ref}}{2} \cos(2\omega_{ref}t + \phi) \quad (6)$$

$$V^{90}(t) = \frac{V_{sig}V_{ref}}{2} \sin(\phi) + \frac{V_{sig}V_{ref}}{2} \sin(2\omega_{ref}t + \phi) \quad (7)$$

Now if we set  $V_{ref} = 2V$ , we get the components as:

$$V^0(t) = V_{sig}(\cos(\phi) + \cos(2\omega_{ref}t + \phi)) \quad (8)$$

$$V^{90}(t) = V_{sig}(\sin(\phi) + \sin(2\omega_{ref}t + \phi)) \quad (9)$$

Now we can perform a low pass filter to get the DC components of each of these voltages which gives us the amplitude of the input signal.

The Low pass filtering is done by taking an FFT or Fast Fourier Transform of the in-phase and the Quadrature signals. Thus, by eliminating the other frequencies, we will get the required Amplitude of the input signal.

Lets assume that after we take the FFT of the in-phase and the quadrature signals, we get the DC components as  $V^0$  and  $V^{90}$  respectively. Then we can now clearly see that the magnitude of the amplitude is given by the following equation (after eliminating the higher frequency components):

$$V_{sig} = \sqrt{V^0{}^2 + V^{90}{}^2} \quad (10)$$

### 2.1 Fast Fourier Transform

The Fast Fourier Transform (FFT) is a powerful mathematical algorithm used to efficiently compute the discrete Fourier transform (DFT) and its inverse. By converting a signal from its time domain representation to its frequency domain representation, the FFT enables the analysis of signal components at different frequencies. This process helps in identifying dominant frequencies, filtering out noise, and extracting useful information from complex signals.

For our experiment, the Voltage values and the corresponding time values are stored in an array. The FFT of the array is then taken and plotted in the Frequency domain. As we know the Fourier transform of sine and cosine functions are delta functions. Since

in the FFT, unlike the analytical Fourier transform, we will not get an infinity at the specific frequency in which the delta function is infinity. The height of the peak is calculated from the computing algorithm to be  $V_{sig} \times \text{"The Length of the array"}$ . So, by dividing the peak value by the array length, we will get the  $V_{sig}$ .

Now since we are taking the FFT we now have the real and imaginary part. This is for each of the in phase and quadrature signals. So, the amplitude of the signal  $V_{sig}$  is given by:

$$V_{sig} = \sqrt{V^0{}^2 + V^{90}{}^2} \quad (11)$$

Where the  $V^0$  and  $V^{90}$  is obtained by dividing the peaks by the array length of the respective  $V^0(t)$  and  $V^{90}(t)$

## 2.2 Calibration of the LIA

The Lock in amplifier is calibrated using a known signal from the function generator. A plot of the  $V_{in}$  vs  $V_{out}$  is made where  $V_{in}$  is the input signal and the output signal is  $V_{out}$ . We can use either the peak-to-peak value of the voltage or the RMS value. The slope of the plot gives us the amplification factor of the Lock in amplifier. The calibration is mainly done when an external amplifier (Using IC741 or TL083) is connected before the LIA. This may be required if the input voltage is very low, the SEELab or ExpEYES may not be able to measure it properly. So, an amplification along with the calibration is required for the smooth working of the Lock in amplifier. From the slope of the Calibration curve, we can calculate the amplification factor  $\alpha$  ( $slope = \alpha$ ).

## 2.3 Measurement of Low Resistance

We can measure the low resistance using the Lock in amplifier. The circuit diagram for this measurement is shown in the Figure-1.

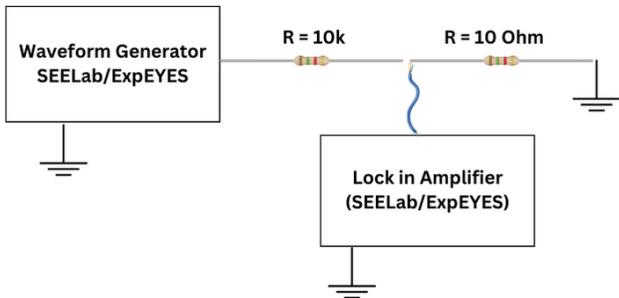


Figure 1: Circuit Diagram for the Measurement of Low Resistance

In the figure, we set the resistance  $R$  to be much greater than the small resistance  $r$ . The total current through the circuit is given by:

$$I = \frac{V}{R+r} \quad (12)$$

Since  $R \gg r$ , we can write  $I = V/R$ . As the resistances are connected in series, the current through them will be the same. So,

$$\frac{V_R}{R} = \frac{V_r}{r} = \frac{V}{R} \quad (13)$$

Also, we have the amplification factor  $\alpha$ , which was obtained from the calibration. Then  $V_{out} = V_r \times \alpha$ . So, we get:

$$r = \frac{RV_{out}}{\alpha V} \quad (14)$$

So by measuring the  $V_{out}$  using the LIA, we can measure the low resistance  $r$ .

## 2.4 C-V Profiling of Solar Cell

Lock in Amplifier can also be used in the C-V-Profilng of not only the solar cell but also for Schottky Diodes, MOS-FETs, etc. In Our case, we will be restricting ourselves to the case of Solar Cell only. We will be doing the profiling for the Monocrystalline as well as Polycrystalline solar cell. The circuit Diagram for the same is shown below in Figure 2.

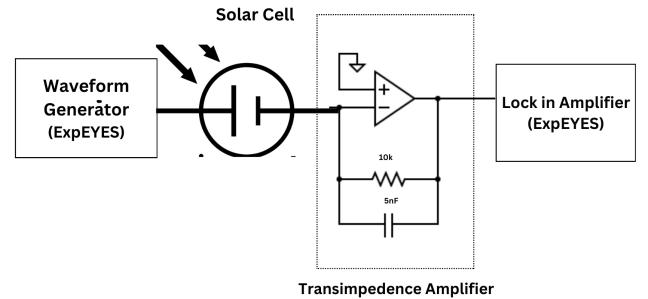


Figure 2: Circuit Diagram CV Profiling of Solar Cell

For the solar cell, if we apply a reverse bias voltage  $-V_R$ , the net potential at the depletion region will become:

$$-V_0 = -V_R + V_{bi} \quad (15)$$

Where  $V_{bi}$  is the built in potential of the silicon. Let the total charge on the surface be  $-Q$ . The depletion region will spread to a width  $W$ .as a result of semiconductor conduction electrons moving away from the interface. When the semiconductor reaches the depletion area, the electric field is zero and the semiconductor becomes

neutral, negating the impact of  $-Q$ .  $W$  is the screening length, and this effect is known as dielectric screening.

Firstly, for simplicity's sake, we'll assume a uniform doping density, where each dopant atom has a single charge  $+e$ , or  $\rho(x) = \rho$  (a constant). By symmetry, we can infer that the electric field  $\vec{E}$  in the depletion area points in the (negative) x-direction if the cross-sectional dimensions of the Depletion region are significantly larger than  $W$ . Then, using the rectangular Gaussian surface around the depletion region with one of its end caps in the neutral bulk region (where  $\vec{E} = 0$ ) and the other end cap located a distance  $x$  from the depletion region of n-p interface, The Gauss Law gives:

$$\epsilon EA = \frac{+e\rho A \times (W - x)}{\epsilon_0} \quad (16)$$

Where  $A$  is the area of the solar cell depletion region,  $\epsilon$  is the dielectric constant of the solar cell depletion region. Now to calculate the electric potential as a function of  $x$ , we impose the boundary condition:

$$V(0) = -(V_R + V_{bi}) \quad (17)$$

$$V(W) = 0 \quad (18)$$

now we have:

$$V(W) - V(0) = \int_0^W \vec{E} \cdot d\vec{x} \quad (19)$$

$$\text{where } \vec{E} = \frac{+e\rho(W - x)}{\epsilon_0} \hat{x} \quad (20)$$

on solving the above equation we get:

$$V_R + V_{bi} = \frac{e\rho W^2}{2\epsilon_0} \quad (21)$$

on solving for the depletion width  $W$  we get:

$$W = \sqrt{\frac{2\epsilon_0(V_R + V_{bi})}{e\rho}} \quad (22)$$

Now if we increase the reverse bias voltage by  $dV_R$ , the depletion width will increase by  $dW$  and the charge on the surface will increase by  $dQ$ . The extra space charge and the  $dQ$  is related as  $dQ = e\rho AdW$ . So, differentiating the equation-21 we get:

$$dV_R = \frac{e\rho AW dW}{\epsilon_0} = \frac{W dQ}{\epsilon_0 A} \quad (23)$$

Now we can relate the capacitance to the doping density as:

$$C = \frac{dQ}{dV_R} = \frac{\epsilon_0 A}{W} \quad (24)$$

Now using the equation-22 we can write the capacitance as:

$$\frac{1}{C^2} = \frac{2}{A^2 \epsilon \epsilon_0 \rho} (V_R + V_{bi}) \quad (25)$$

So for a Solar cell depletion region, the plot of  $1/C^2$  vs  $V_R$  is a straight line with slope  $\frac{2}{A^2 \epsilon \epsilon_0 \rho}$  and intercept  $V_{bi}$  from where we can calculate the doping density and the built in potential.[5]

#### 2.4.1 Autophasing

There could be more stray sources of capacitance in the circuit in addition to the capacitance of interest. We must remove unnecessary capacitance in order to improve the accuracy of our measurements. The method employed for this is autophasing. This technique's primary idea is to take a sort of dot product and project the device's capacitance onto an ideal capacitor.

Here, we replace our device with an ideal calibration capacitor of known value ( $C_0$ ). The in-phase ( $V_{x0}$ ) and the quadrature components ( $V_{y0}$ ) of the voltage obtained from this capacitor are measured through lock-in detection and are noted for all the biasing voltages. Using these values, the capacitance of the device (C) can be calculated as:

$$C = \frac{V_{0x} V_x + V_{0y} V_y}{V_{0x}^2 + V_{0y}^2} \times C_0 \quad (26)$$

where  $V_x$  and  $V_y$  are the in-phase and quadrature components obtained from the device.

### 3 Experimental Setup and Procedure

The Setup of the Experiment mainly involves the use of the python library of the ExpEYES/SEELab called the eyes17 library in python. The signal required for the processing is collected by the A1/A2 port of the ExpEYES. Both of the reference signals for the LIA are generated (it is because we assume that we already know the frequency of the signal that we need) within the software itself using the NumPy library.

#### 3.1 Building the Lock-in Amplifier

The first step is to import the python library eyes17. In order to access the device, we need the library. This is shown in the listing-1

```

1 import numpy as np
2 import math as m
3 from scipy.fft import fft, ifft, fftfreq
4 import eyes17.eyes
5 p=eyes17.eyes.open()
6 import time as tim
```

Listing 1: importing the libraries-[3]-[6]-[4]-[1]

For simplicity, we will create the Lock in amplifier as a python function. As discussed in the theory the signal and the references are multiplied and an FFT is taken. The FFT is taken from the SciPy library. And all the plots are made in Matplotlib. As the signal is collected in arrays of voltage and time values, our function will have to take the arguments as a list of voltage and time. Also, one of the other arguments is the frequency. It is important to note that the units of the time which should be in seconds before passing it to the LIA function. In the upcoming sections. The whole list has been divided by a factor of 1000 in order to maintain clarity in the units. The function is shown in the Listing-2.

```

1 def LIA(V_in,t_in,f):
2     V_in_sin=[]
3     V_in_cos=[]
4     for i in range(len(V_in)):
5         V_in_sin.append(V_in[i]*2*np.sin(2*np.pi
6             *f*t_in[i]))
6         V_in_cos.append(V_in[i]*2*np.cos(2*np.pi
7             *f*t_in[i]))
7     V_out_sin_fft=fft(V_in_sin)
8     V_out_cos_fft=fft(V_in_cos)
9     V_out= np.sqrt((V_out_sin_fft[0].real/len(
10    V_out_sin_fft))**2+(V_out_cos_fft[0].real/len(
10    V_out_cos_fft))**2)
10

```

Listing 2: Defining the Lock-In-Amplifier Function

The algorithm is exactly the same as that mentioned in the theory section. The outputs of the function LIA are " $V_{out}$ " as we can see in the code, is the amplitude of the signal with frequency  $f$  present in the noisy signal. For the CV profiling the function has been modified to return the in phase and the quadrature components that is required for the autophasing as shown in the listing-3.

```

1 V_out_sin_fft=fft(V_in_sin)
2 Vx=V_out_sin_fft[0].real/len(V_out_sin_fft)
3 V_out_cos_fft=fft(V_in_cos)
4 Vy=V_out_cos_fft[0].real/len(V_out_cos_fft)
5

```

Listing 3: Modified function for CV-Profilng

The complete code and calculations for this experiment can be accessed in our [GitHub repository](#).

The signals are taken as the inputs using the capture1 function in the eyes17 library by the following method:

```
1 t,v = p.capture1(stri,8192,2)
```

The 8192 is the total number of samples that is to be measured in a single run of the above line and 2 is the time gap between each sample whose units is in microseconds. The time obtained above is in milliseconds which we have to convert to seconds before feeding it to the LIA.

The value of number of samples is chosen as a power of 2 (Maximum capacity for the device is 10000 samples in one run) so that the FFT can be taken easily. The time gap is chosen as 2 microseconds so that the signal is sampled at a good rate of 50 MSPS.

## 3.2 Procedure

The procedure mainly includes the building of the external circuit. The output from the WG of the SEELab is very restricted, that is it can only give out 80mV, 1V or 3V depending on the arguments of the following command:

```

1 p.set_sine_amp(2)
2 p.set_sine(f)

```

Where in the former line, one can set it as 0 for 80mV, 1 for 1V and 2 for 3V. Since we require a continuous variation of the sinusoidal potential, we will be using a potentiometer to divide the voltage and get the required voltage amplitude.

### 3.2.1 Calibration

For the Calibration the output from the potentiometer is directly given to the IC741 with  $100k\Omega$  feedback resistor and input resistance of  $1k\Omega$  which together should give rise to an amplification of  $100\times$ . The software amplification been set to one for the sake of simplicity. For the calibration  $R_f = 100k\Omega$  is used as the feedback resistor in the inverting amplifier before LIA

### 3.2.2 Measurement of Low Resistance

For this part the, the circuit is the same as shown in Figure-1. The input of the potentiometer is given to the WG and the output to the high resistance  $R$ . the voltage across the small resistance  $r$  is measured using the built LIA. The resistances used are  $R = 10000\Omega$ .

### 3.2.3 C-V Profiling of Solar Cell

The circuit for the CV profiling is shown in the Figure-2. Here we have built an adder circuit using the IC741 in order to add the DC voltage and a small DC signal. The DC signal is generated from the pv1 of the ExpEYES and the sine wave is generated from the WG.

Since, we want small perturbations with the DC voltage we need to add these signals together. So, the modified circuit for the CV profiling is shown in Figure 3.

The resistances of the adder circuit are set in such a way that the net gain through the adder is 1. Unlike the Calibration the feedback resistance used in the transimpedance amplifier is  $10k$ . and a  $5nF$  capacitor is used. The calibration capacitor has a value of  $5.56nF$ . which is used for further calculations. All this is done and it is made sure that the solar cell is reverse biased. In the circuit show above, there is a potentiometer in between WG and the adder circuit. For simplicity, it's not shown in the circuit.

The algorithm for the collection of data is that the PV1 Is varied in a loop and corresponding  $V_x$  and  $V_y$  are written to a text file, 1st with the calibration capacitor and then with the solar cell. The code for this is given

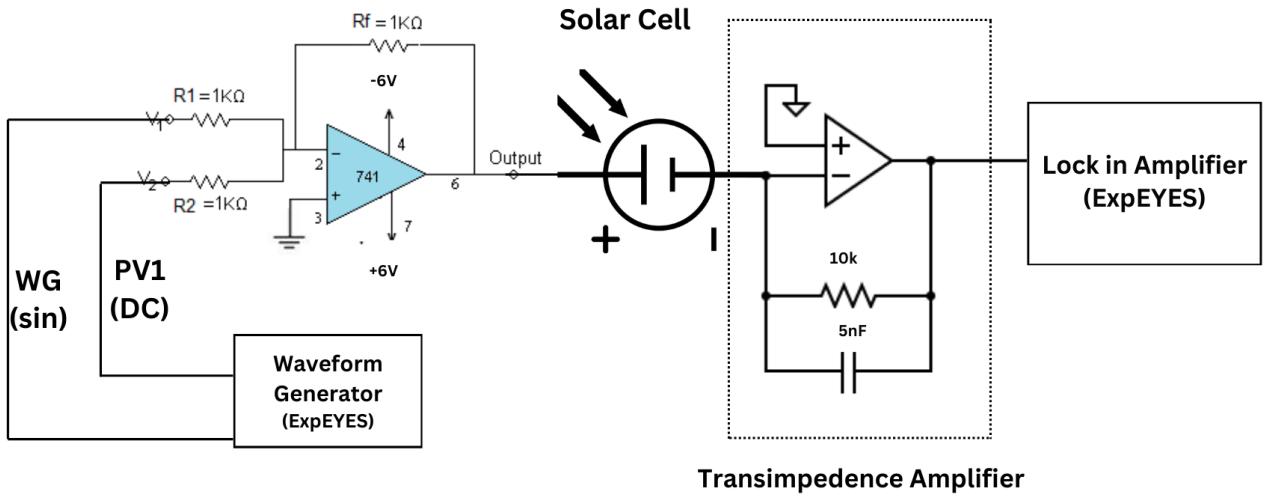


Figure 3: Modified Circuit Diagram CV Profiling of Solar Cell

below in listing 4. Then the collected data is then used for further analysis

```

1 vset =[0.2,0.3,0.4,0.5,0.6,0.7,0.8,
2      0.9,1.0,1.1,1.2,1.3,1.4,
3      1.5,1.6,1.7,1.8,1.9,2,2.1]
4 j=0
5 for i in vset:
6     p.set_pv1(i)
7     tim.sleep(3)
8     t,v=collect_signal('A1',N_sample,N_div,f)
9     tim.sleep(3)
10    Vx,Vy=LIA(v,t/1000,f)
11    print(str(vset[j])+'\t'+str(Vx)+'\t'+str(Vy)
12    )
13    with open('Data\\SS.txt','a') as file1:
14        file1.write(str(vset[j])+'\t'+str(Vx)+'
t'+str(Vy)+'\n')
j=j+1

```

Listing 4: Collecting the data as arrays for the CV Profiling[2]

The setup of the Cv-Profilng Experiment is shown in the Figure-4.

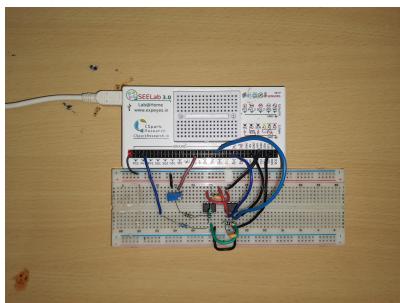


Figure 4: Setup for the CV Profiling of Solar Cell (connected to calibration capacitor)

## 4 Observations and Data

### 4.1 Verification of working of the Lock in Amplifier

The functioning of the Lock in amplifier was verified by using a known signal from generated in the python library with noise. The input signal and output were measured and plotted so as to verify the working of the LIA. The plot is shown in the Figure-5. Here the amplification is set to 10x virtually.

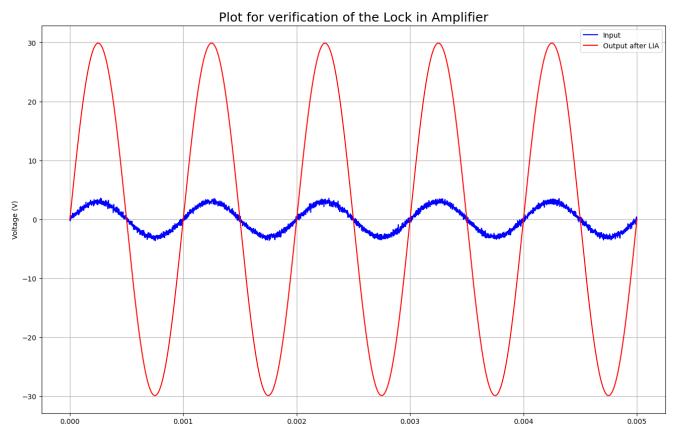


Figure 5: Plot for verification of the working of the Lock in Amplifier function

Now a real signal from the WG of the SEELab is given to the A1 and the output is measured. The plot is shown

in the Figure-6. Here the amplification is set back to unity.

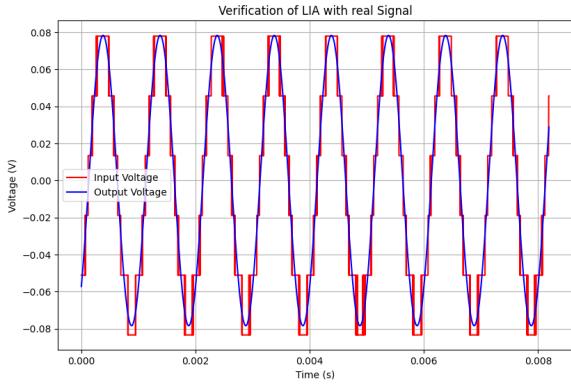


Figure 6: Plot for verification of the working of the Lock in Amplifier function with a real signal from SEELab

The input signal and the reference signals had a phase difference of  $\phi$ . Since we are now obtaining the  $V_x$  and  $V_y$ , this phase was removed following the amplification. The following formula is used to do this:

$$\phi = \tan^{-1} \frac{V_x}{V_y} \quad (27)$$

This phase is added to the output signal so as to match with the input signal.

## 4.2 Data for Calibration

Data for the calibration is shown in the Table-1. The data contains the input voltage, output voltage (both in RMS) for different frequencies of 500Hz, 1000Hz, and 2000Hz.

## 4.3 Data for Calculation of Low Resistance

The data for the Low resistance include the input voltages and the output voltages and the calculated values of the low resistance along with its standard deviation error are shown in Table-2.

$V_{in}$ (V)	$V_{out}$ (V)	r ( $\Omega$ )
0.15339623	0.00285270	1.58443460
0.17830225	0.00288968	1.36311397
0.19489926	0.00284919	1.24703544
0.20888439	0.00279952	1.16354451
0.23100520	0.00285050	1.05212476
<b>Error, <math>\Delta r</math></b>		<b>0.04076064</b>
<b>Avg r</b>		<b>1.07516899</b>

Table 2: Data for the Calculation of Low resistance

## 4.4 Data for C-V Profiling

The observations for the CV profiling of the monocrystalline and polycrystalline solar cells are too large to be accommodated in this report. These can be accessed via our [GitHub repository](#) in the "Folder -> Data -> CV" where you can find the file Data.md in which the CV data's are stored in tables. Or the more detailed Excel file can be found in the same folder which contain different sheets of data for poly and mono crystalline solar cells.

## 5 Data analysis

The datasets from the previous section were analyzed and the corresponding calculations is shown in the following subsections.

### 5.1 Calibration of the Lock in Amplifier

The plotting of the  $V_{out}$  vs  $V_{in}$  is done in python for different frequencies and the results are tabulated in the Table-3 below:

Frequency (Hz)	Slope/ $\alpha$	$\Delta\alpha$
500	116.2499844	12.30863049
1000	114.4791589	14.78432675
2000	105.7033615	12.7032036
<b>Average</b>	112.1441683	13.26538695

Table 3: Results from the calibration curve

From the above table, the net amplification factor is averaged out to be:

$$\alpha = 112.144 \quad (28)$$

The Corresponding plot of the Calibration is shown in the Figure-7.

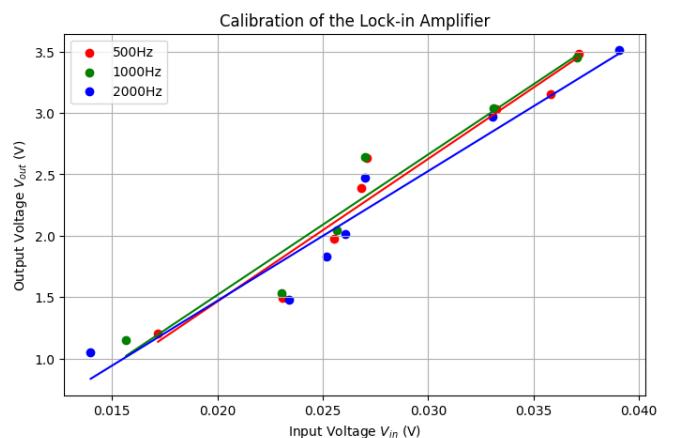


Figure 7: Calibration curve of the Lock in amplifier

Sl No	500Hz		1000Hz		2000Hz	
	$V_{in}$ (V)	$V_{out}$ (V)	$V_{in}$ (V)	$V_{out}$ (V)	$V_{in}$ (V)	$V_{out}$ (V)
1	0.017182018	1.203372531	0.015671727	1.149396809	0.013987165	1.047709881
2	0.023065578	1.492325673	0.023028788	1.528740499	0.023391511	1.479491873
3	0.025554896	1.978069124	0.025688889	2.04770376	0.025175565	1.828756137
4	0.026813747	2.387335743	0.027003734	2.641769499	0.026062813	2.016311599
5	0.027089012	2.636136896	0.03309075	3.039432545	0.027024287	2.475542671
6	0.033221258	3.031335845	0.037057903	3.451293706	0.033044768	2.974094531
7	0.035821276	3.153957316			0.039064961	3.511858117
8	0.037135761	3.484986789				

Table 1: Data for the calibration of the Lock In Amplifier

## 5.2 Measurement of Low Resistance

The calculations for the low resistance are done using the formula in 14. The results are shown in the Table 2 itself. The obtained value of the Low resistance r is:

$$r = 1.075\Omega \quad (29)$$

## 5.3 C-V Profiling of Solar Cell

The data shown in the GitHub Repository are plotted in python and the corresponding plots are made. The linear fitting of the  $1/C^2$  vs  $V_R$  is done. In this part the CV profiling is done in dark conditions for both of the solar cells. The data are linear fitted and the slopes and intercepts are tabulated in the Table-4. The further calculations are as follows. The plots are shown in the Figure-8 and Figure-9 for the polycrystalline one and in Figure-10 and Figure-11 for the monocrystalline one.

Type ->	Monocrystalline	Polycrystalline
slope, m	5.743692249	53.21159361
err in slope, $\Delta m$	0.445068424	1.604544349
Intercept c	-2.880222429	-17.33620863
err in intercept $\Delta c$	0.5725663	2.064195013

Table 4: Linear fitted parameters for mono and poly crystalline solar cells

Let the slope of the  $1/C^2$  vs  $V$  plot be  $m$ . Then by equation-25, the doping density is given by:

$$\rho = \frac{2}{A^2 e \epsilon_0 m} \quad (30)$$

And the built in potential is given by:

$$V_{bi} = \frac{c}{m} \quad (31)$$

where  $c$  is the intercept of the plot.

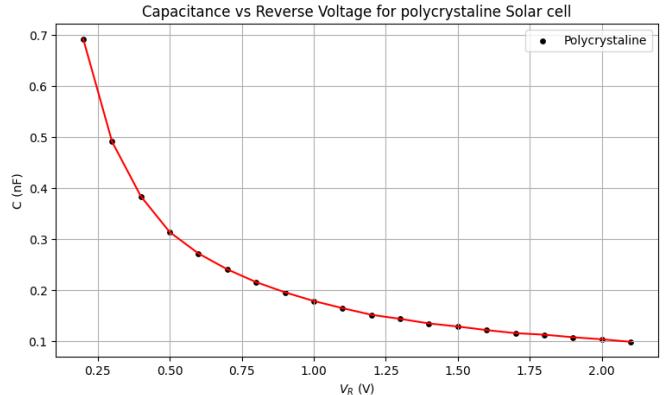


Figure 8: CV Plot for polycrystalline solar cell

So, for the polycrystalline solar cell, using the area as  $0.036m^2$ , and the dielectric constant  $\epsilon = 11.68$  for Silicon, the values are:

$$\rho^{poly} = 1.753 \times 10^{12} m^{-3} \quad (32)$$

$$V_{bi}^{poly} = -0.325V \quad (33)$$

And similarly for the monocrystalline one:

$$\rho^{mono} = 1.624 \times 10^{15} m^{-3} \quad (34)$$

$$V_{bi}^{mono} = -0.501V \quad (35)$$

For the mono crystalline one,  $A = 0.0036m^2$  and the dielectric constant is taken to be 11.68.

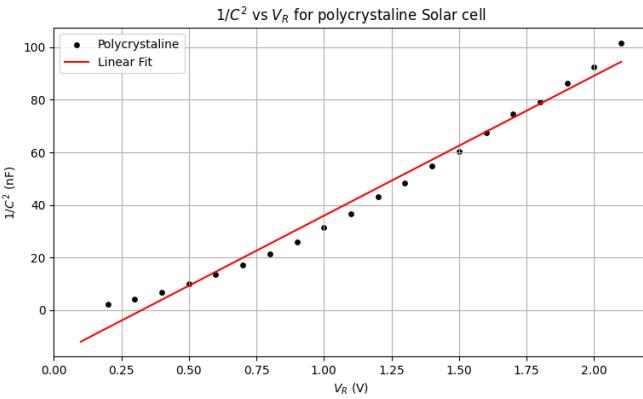


Figure 9: Plot of  $1/C^2$  vs  $V_R$  for the polycrystalline solar cell

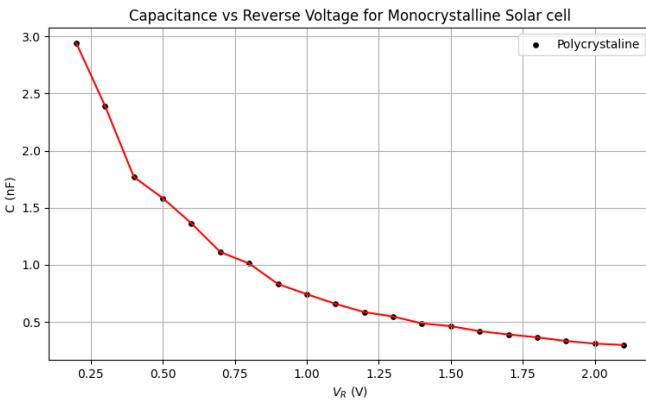


Figure 10: CV Plot for monocrystalline solar cell

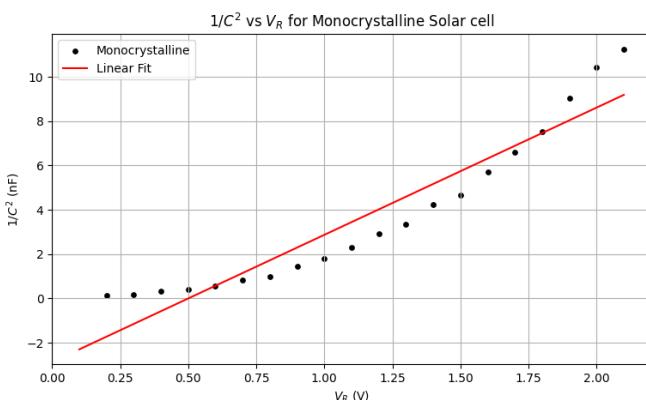


Figure 11: Plot of  $1/C^2$  vs  $V_R$  for the monocrystalline solar cell

## 6 Error Analysis

Let  $\Delta x$  be the error in the quantity  $x$ . if the error on  $x$  depends on some random errors in the quantities  $A, B, C\dots$  like  $x = A^a B^b C^c \dots$  etc then the error in the quantity  $x$  can be calculated as:

$$\frac{\Delta x}{x} = \sqrt{(a \frac{\Delta A}{A})^2 + (b \frac{\Delta B}{B})^2 + \dots} \quad (36)$$

### 6.1 Error in the Calibration

The error in the calibration is just the error in the slopes of the calibration curve which were linear fitted. The value of the amplification factor is calculated out to be:

$$\Delta\alpha = 13.265 \quad (37)$$

The error in the Low resistance, as we can see from Table-2, we use the standard deviation error. So, we get the value of the low resistance  $r$  as:

$$\Delta r = 0.041\Omega \quad (38)$$

### 6.2 Error in the Doping density and built in potential

From the Equation-25, we can see that the error in the doping density depends on only on the slope of the  $1/C^2$  vs  $V$  curve, because other parameters are constants and are already known values. So, the percentage error in the doping density will be same as that of the percentage error in slope. So, on calculating the error using linear regression, the final value of the doping density can be written as:

$$\rho^{mono} = 0.126 \times 10^{15} m^{-3} \quad (39)$$

$$\rho^{poly} = 0.052 \times 10^{12} m^{-3} \quad (40)$$

Now for the built in potential, we know its value depends on both of the intercept and slope, so, we will be using the equation-36 for the same, so we will get:

$$\Delta V_{bi}^{mono} = 0.106V \quad (41)$$

$$\Delta V_{bi}^{poly} = 0.123V \quad (42)$$

## 7 Results

The results obtained from this experiment are as follows.

$$\alpha = 112.144 \pm 13.265 \quad (43)$$

$$r = 1.075 \pm 0.041\Omega \quad (44)$$

$$\rho^{mono} = (1.624 \pm 0.126) \times 10^{15} m^{-3} \quad (45)$$

$$\rho^{poly} = (1.753 \pm 0.052) \times 10^{12} m^{-3} \quad (46)$$

$$V_{bi}^{mono} = (-0.501 \pm 0.106)V \quad (47)$$

$$V_{bi}^{poly} = (-0.326 \pm 0.123)V \quad (48)$$

## 8 Conclusion and Discussions

### 8.1 Conclusions

Using the SEELab and some low cost IC's and resistors, we were able to build a lock in amplifier. The calibration of the lock in amplifier was done and the amplification factor was calculated. The low resistance was measured and the CV profiling of the solar cells was done. The doping density and the built in potential of the solar cells were calculated. The results are in good agreement with the literature values. The error in the measurements is also calculated and is found to be within acceptable limits. It is evident from the data and the calculations that this low-cost method of building a lock in amplifier using SEELab along with Python, both which has an open-source library is very efficient and successful.

### 8.2 Discussions

1. It's possible that the circuit's extremely high feedback resistance is the cause of the high DC offset voltages seen with the TL082 IC. A significant voltage drop resulting from the input bias current might be caused by a high feedback resistance. The high input base current of the TL082 IC is 20 nA. The output has a DC offset as a result of this voltage drop. In a network with numerous integrated circuits connected, each one may contribute to the DC offset; as a result, we will obtain a significant offset voltage.
2. The phase shift in the output signal is due to the fact that the input signal and the reference signal are not in phase. This phase shift is removed by adding the phase shift to the output signal. However this can be avoided by using a phase shifter circuit or triggering of the SEELab. We didn't use

the inbuilt triggering since it was not required for the experiments that we did.

3. There was also a problem with the SEELab's Sampling as the sampling was not uniform. For Example. Lets consider 20 samples out of 8192 that we took. Let the first sample has a voltage 0.5 and the 20th has a voltage of 0.6. Lets say an intermediate voltage of 0.55 is also collected. The output array will contain 0.5,0.55,0.5,0.55,0.5...0.6. That is the output oscillated between 0.5 and 0.55 for a few samples before reaching the value of 0.6. This is a problem with the SEELab and not with the lock in amplifier. Otherwise the results could have been more accurate.

### 8.3 Sources of error

1. Loose connections and stray capacitance in the circuit can cause errors in the measurements.
2. Setting the sampling of the SEELab to a perfect value.(not over and under sampling)

### 8.4 Future Prospects

1. The Lock in amplifier can be modified to obtain live signals by setting proper triggering and phase shifting circuits.
2. The SEELab can be modified to have a better sampling rate and a better triggering system.
3. This Lock in amplifier can be used for other experiments such as measurement of Mutual inductance of two coils, measurement of the phase difference between two signals etc.
4. The lock-in algorithm can be made more computationally efficient by modifying the python code.

## References

- [1] csparkresearch. Seelablet. <https://github.com/csparkresearch/SEELablet.git>, Year. Accessed: March 3, 2024.
- [2] Python Software Foundation. Python programming language, 1991–present.
- [3] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [4] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [5] Neal D Reynolds, Cristian D Panda, and John M Essick. Capacitance-voltage profiling: Research-grade approach versus low-cost alternatives. *American Journal of Physics*, 82(3):196–205, 2014.

- [6] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.