

## Lab 15: Lock-in amplifier (Version 1.4)

**WARNING:** Use electrical test equipment with care! Always double-check connections before applying power. Look for short circuits, which can quickly destroy expensive equipment.

### Lab preparation

An assigned LabView VI simulates lock-in detection in a noisy experimental environment. By specifying the amount of data points read in a given time, the integration and hence accuracy of a measurement can be controlled. This lab uses lock-in techniques to measure the output of a sinusoidally modulated LED with a photodiode. NOTE: You will be making a complicated LabView Block Diagram. Please follow the directions carefully.

### LED

Insert an LED into two open holes near a far edge of the Elvis breadboard. Add a current limiting resistor in the range 300--500 ohms in series, then verify the proper polarity of the diode by connecting it to a 5V DC bias. If the LED fails to illuminate, switch its polarity. Bend the diode so that it points to the other side of the Elvis board where the photodiode will be located. Disconnect the LED from 5V as it will be directly driven by the analog output. No transistor is used.

### Configure the VI

Start LabView, open your assigned VI that simulates a lock-in amplifier, and create a blank VI. On the Front Panel, copy your controls for frequency, amplitude, and offset. Add a new control for phase (degrees), a control for sampling rate, and a control for number of sample points. (Do not copy the existing cluster control) All these controls should be DBL representation. Also copy the three waveform graphs and the chart displaying lock-in output. Arrange the reference and signal (input) graphs vertically. This will help adjusting the phase.

On the Block Diagram, select the DAQmx palette and pin it open. Create two channels: one for analog voltage input and one for analog voltage output. Right-click on the **physical channels** terminal and create a constant for both. Select one of the 16 analog input channels and one of the two analog outputs (ao0 or ao1).

The two analog I/O channels must be synchronized for lock-in detection to work properly. Place two copies of DAQmx Timing on the Block Diagram. Create a constant on the **sample mode** terminal of each and set both for Continuous Samples.

The output channel will drive the LED with a continuous sine wave. This will be implemented *outside* the While Loop. Add DAQmx Write and DAQmx Start Task to the Block Diagram. Configure DAQmx Write for Analog: Single Channel: Multiple Samples: Waveform. Wire the **task/channels in** and **task out** terminals to force left-to-right data flow as follows: Create Channel, Timing, Write, and Start.

Add a second DAQmx Start and set the input channel sequence as: Create Channel, Timing, and Start. Do not wire the error clusters at this time. Arrange the two channels vertically, with the input channel on top.

Place the Sine Waveform VI on the Block Diagram (this is the same function used in the assignment) and connect the Front Panel controls (amplitude, frequency, offset, phase) to the corresponding terminals.

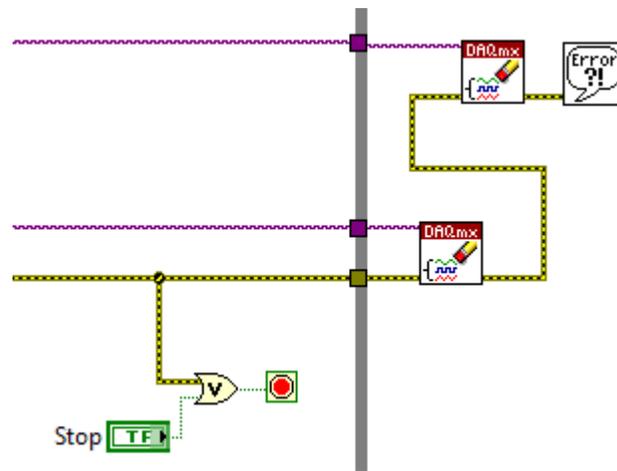
Use the cluster bundle operation to create a two-component cluster with the sampling rate control on the top terminal and sample points control on the lower terminal. Wire this cluster to the **sampling info** terminal of the sine function. Wire the **signal out** terminal to the **data** terminal of the DAQmx Write VI.

Now place a While Loop to the immediate right of the channel configuration code you have just written. Initially, there should be nothing in the While Loop. Add a Stop button control to the loop's conditional stop. Inside the While Loop, place the following two VIs from the DAQmx palette: DAQmx Read and DAQmx Advanced Task Options: Is Task Done. Connect the Read operation to the input chain and the Task Done query to the write chain. This is done by connecting the purple **task in/out** wiring through the While Loop creating two tunnels. The last step in the DAQmx setup is to clear both channels with two Clear Task VIs placed on the right outside of the While Loop. Connect the corresponding **task in/out** wires through two output tunnels. Do not wire any error clusters at this time.

Inside the While Loop, configure DAQmx Read for Analog: Single Channel: Multiple Samples: Waveform. Connect its **data** output terminal to the signal waveform graph. This will display the photodiode input signal. It will be processed with the lock-in algorithm. The input terminal labeled **number of samples per channel** must be connected to the sample points control that is outside the While Loop.

Outside the While Loop, the two DAQmx Timing VIs require a **rate** to be specified. Connect the sampling rate control to both terminals. The **source** for the analog input





Also add an OR gate to automatically stop the While Loop if an error is encountered.

The While Loop is timed the same way as in the assignment, except the controls are already located outside the loop. Write code to perform the division operation: **sample points/sampling rate**, multiply by 1000, and connect this to a “Wait Until next ms Multiple” VI inside the loop.

### Flash the LED

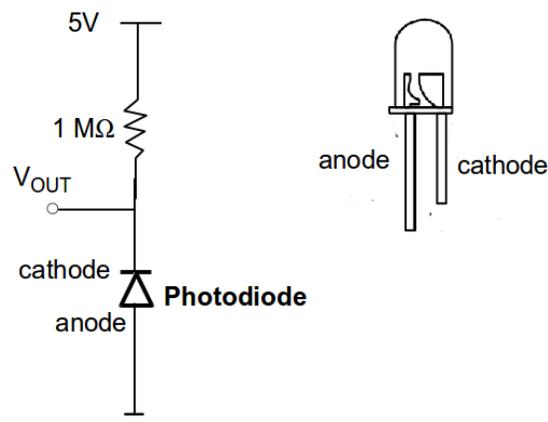
An intermediate test can be done at this point. (The two graphs and the RMS chart are still unwired, but this is OK) The program should be able to flash the LED. Initial control parameters are not critical. The Elvis board is capable of high sampling rate, so it can be set for 10,000 samples/sec, a frequency of 10 Hz, and 5000 sample points to get started. Leave the phase at 0 deg. Drive the LED with 1.5V amplitude and 2V offset that are set using the Front Panel controls. The LED control signal coming from the analog output channel can also be monitored with the analog input channel you have configured. Simply connect the corresponding terminals on the I/O channels. Don't forget that analog I/O requires a ground connection on the negative terminals or they will not work. When the program is run, you should see the LED clearly flash at 10 Hz and the sinusoidal drive signal displayed on the Front Panel. When the program is stopped, a DC voltage may remain on the analog output causing the LED to glow continuously, but this is OK. If the two analog I/O channels are not working, troubleshoot as necessary before going further.

NOTE: Enabling the Highlight Execution diagnostic mode here will cause an immediate error inside the While Loop. Data acquisition in this VI is timed by software; slowing

down the loop will cause the DAQmx read buffer to overflow.

## Photodiode

Insert a photodiode in two open holes on the far side of the board -- opposite the LED and facing it. There should be several cm of separation between the LED and photodiode; they can even be located at opposite corners of the Elvis board. Bend the photodiode to face the LED. The photodiode must be operated with a *reverse* bias, whereas the LED runs with a *forward* bias. It should be connected to 5V through a 1 M $\Omega$  resistor as shown in the schematic. Be sure to measure the resistor as a smaller value may lead to damage of the photodiode. By looking at the electrodes inside the clear plastic capsule, the anode/cathode can be easily recognized. Note that 5V/GND connections are conveniently located on both sides of the Elvis module.



Disconnect the LED drive signal from the analog input channel. Connect the photodiode voltage ( $V_{OUT}$ ) directly to the analog input and confirm that a 10 Hz signal is displayed on the Front Panel when the VI is running. This is direct detection (i.e. having poor sensitivity) as no reference signal multiplication has been done yet. The photodiode will also detect ambient room light so it may be helpful to shield the path. When the light path is completely blocked, however, the detected signal should disappear into the noise. Do not proceed until you are certain the photodiode is working properly. The appendix briefly describes the physics of photodiode operation.

## Reference signal and lock-in configuration

Stop the VI. A reference signal is needed to get the sensitivity inherent to lock-in operation. This is obtained by placing a second copy of Sine Waveform.vi *inside* the While Loop. (Tip: If you need to make space in your Block Diagram, hold down the CTRL

button, left-click the mouse, and drag) Connect the **sample info** cluster (same as used above) to its corresponding terminal. Wire the frequency control to the frequency terminal, but leave the other terminals at their default values. Connect the **signal out** to the Reference waveform graph, also inside the loop. The remainder of the setup is identical to the LabView homework assignment. Multiply the reference and input waveforms together, display the product on the lock-in signal graph, and route this to the low-pass filter. The Filter Express VI cannot be copied from an existing program. It must be placed new from the palette as was done in the LabView assignment. Set the cutoff frequency as done in the assignment and connect the filter RMS output to the chart and a numeric display. Display the cutoff and response times. You may also wish to limit the displayed points on your graphs with property nodes.

## Testing

Test the VI, starting with the same control parameters as above except increase the operating frequency to 100 Hz. Higher frequency generally leads to smaller noise, which scales as  $1/f$ . Note that the controls are outside the loop, so the VI must be stopped and re-started to implement any Front Panel changes. Observe how phase affects the lock-in output; maximum signal should be obtained when the phase is around 180 degrees. This is because the photo-diode is reverse biased. If the offset voltage is reduced to  $< 1.7V$  and the amplitude is set at 0.1 V, output light may no longer be visible. The lock-in, in contrast, can clearly detect LED photons at these levels, i.e. less than 100  $\mu V$ . It should remain nearly immune to room lights or ambient fluctuations. Only completely blocking the path will cause the signal to disappear.

Demonstrate that the VI can detect very weak LED output. There is no writeup.

## Appendix. Photodiode operation

The photodiode used in this lab consists of a silicon p-n junction. The clear plastic enclosure collects and focuses light on the junction. A semiconductor photodiode is similar to a solar cell. Both convert incident photons into an electrical current due to a quantum transition between distinct energy levels in the semiconductor. A solar cell operates without any bias and uses the photons to produce DC power similar to a battery. A photodiode uses a separate DC power source to reverse bias the junction, allowing detection of incident photons with high speed and sensitivity.