BIOEN 317 Lab 9-10 **Spring 2012**

LAB 9

Background:

Pulse Oximetry is a useful, non-invasive method to monitor a patient's percent oxigenation level (SpO_2) by analyzing the amount of light absorbed by the blood hemoglobin.
A sensor is placed on a thin part of the patient's body, usually a fingertip or earlobe and light with red wavelengths and light with infrared wavelengths is sequentially passed from one side to a photo-detector on the other side.

The measurement that makes this exercise possible, of course, is the light intensity detected by the photomer after transmission through the medium.

If light intensity is to be measured across the UV-visible-IR spectrum, then a spectrophotometer is the desired equipment. It does, however, take some time to scan through the spectrum so it is not optimal for measuring time-varying concentrations. If response time is important and the absorption at a single wavelength is adequate, then a two-wavelength photometer is a better tool.

A simple photometer can be made from a light-emitting diode (LED), a photodetector, and associated electronic circuitry. Because LEDs easily burn out if too much current is passed through them, a current source is often used to drive them (Lab 7 exercise II) and shown in Figure 1.

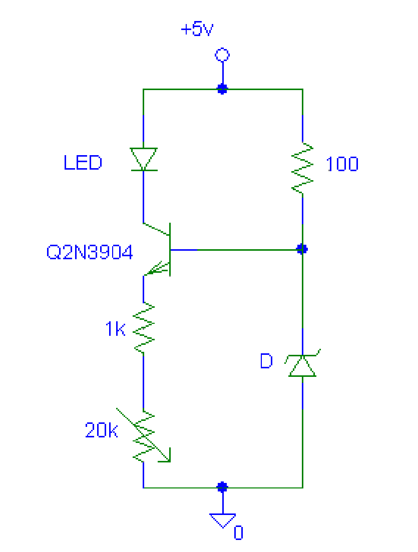

Figure 1. Transistor circuit for a resistance-controlled current source. The transistor's base voltage is set at 3.3 V by a 1N4728 Zener diode.
How it works: In this BJT transistor implementation a *zener voltage stabilizer* (R1=100Ω and D) drives an *emitter follower* (Q2N3904) loaded by an *emitter resistor* (R2=1kΩ+20KΩ potentiometer) sensing the load current. The external load (the LED) of this current source is connected in the collector so that almost the same current flows through it and the emitter resistor (they can be thought as connected in series). The transistor Q2N3904 adjusts the output (collector) current to keep up the voltage drop across the constant emitter resistor R2 almost equal to the relatively constant voltage drop across the zener diode D. As a result, the output current is almost constant even if the load resistance and/or voltage vary.

The color of the LED, the spectral sensitivity of the photodetector, and the absorption spectrum of the material should be matched to provide the best sensitivity. Unfortunately, photodetectors are typically made from semiconductors that are most sensitive to infrared radiation. This less-than-ideal sensitivity spectrum should not cause a problem as long as there is only one absorbing compound and ambient light is prevented from entering the test system.

Use red LEDs for your prototypes. The light will be detected by a broad-band phototransistor that has a sensitivity peak near 800 nm. The current through the phototransistor is converted to a voltage with the help of a resistor, and this voltage can be amplified using an op-amp circuit (today it will be just an amplifier circuit, next week will be a low-pass filter). Figure 2 shows the circuit schematic.

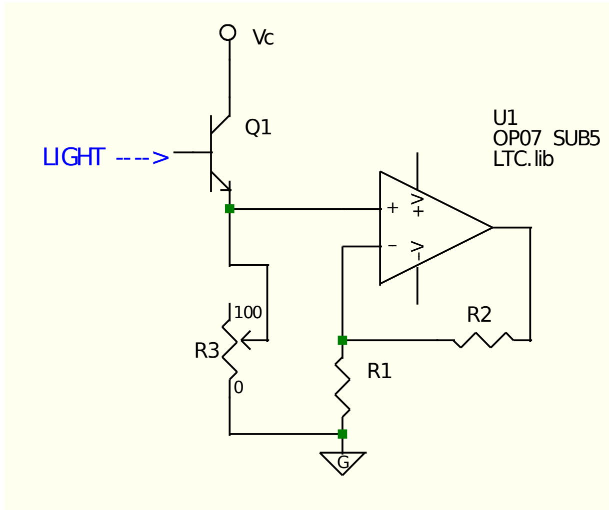

Figure 2. Photodetector circuit. A phototransistor is used as the upper half of a voltage divider, followed by a non-inverting amplifier. The amplifier is optional, but it will improve measurement resolution.

Lab exercise I:

1. Set up the LED current controller circuit shown in Figure 1 (same as lab 7).
2. Set up the photodetector circuit shown in Figure 2 using a BPV11 phototransistor (Q_1) and a potentiometer (R_3) of either 5kΩ or 10kΩ.
3. Choose resistors R_1 and R_2 to provide a gain of 5.
4. Connect the output of the op-amp to the oscilloscope.
5. Test the LED-photodetector pair by modulating the amount of current applied to the LED and looking for a change in the detector output. Note that the photometer works well with low light levels, so it will not be necessary to operate the LED at maximum brightness in your photometer.
6. Discuss your work and results

Lab exercise II: Data Acquisition with an Analog to Digital Converter

Hardware Setup:

1. Connect the op-amp output of the photodetector circuit (Figure 2) to the USB DAQ device. This lead should go to one of the analog input channels, such as AI0. Connect your circuit ground to one of the connections labeled "GND".

Taking Data:

Before we take data in MATLAB, we must perform several setup steps.
In the following, the MATLAB commands you need to use are in RED, in black it shows the output as you change parameters.

Step 1. Discover hardware devices.

```
d  = daq.getDevices

d =

Data acquisition devices:

index Vendor Device ID         Description
----- ------ --------- ---------------------------
1     ni     cDAQ1Mod1 National Instruments NI 9205
2     ni     cDAQ1Mod2 National Instruments NI 9263
3     ni     cDAQ1Mod3 National Instruments NI 9234
4     ni     cDAQ1Mod4 National Instruments NI 9201
5     ni     cDAQ1Mod5 National Instruments NI 9402
6     ni     cDAQ1Mod6 National Instruments NI 9213
7     ni     cDAQ1Mod7 National Instruments NI 9219
8     ni     cDAQ1Mod8 National Instruments NI 9265
```

(this is an example, you may not get all of these Devices, probably only one)

Step 2. Get detailed device information.

Shows the details of the first Device

```
d(1)

ans =

ni: National Instruments NI 9205 (Device ID: 'cDAQ1Mod1')
   Analog input subsystem supports:
      4 ranges supported
      Rates from 0.1 to 250000.0 scans/sec
      32 channels ('ai0' - 'ai31')
      'Voltage' measurement type

This module is in slot 1 of the 'cDAQ-9178' chassis with the name 'cDAQ1'.
```

Step 3. Create a data acquisition session.

```
 s = daq.createSession('ni')

s =

Data acquisition session using National Instruments hardware:
   Will run for 1 second (1000 scans) at 1000 scans/second.
   No channels have been added.
```

Step 4. Configure session properties.

Change the scan rate to 100 scans (Hz) per second and the duration to 120 seconds.

```
s.Rate = 100;
s.DurationInSeconds = 120;


s

s =
```

```
Data acquisition session using National Instruments hardware:
   Will run for 120 seconds (12000 scans) at 100 scans/seconds

No channels have been added.
```

Step 5. Add channels to the session.

Add an analog input channel to the session:

```
s.addAnalogInputChannel('cDAQ1Mod1','ai0', 'Voltage')
```

NOTE: here 'cDAQ1Mod1' should be the Devide ID you get from Step 1

```
ans =

Data acquisition session using National Instruments hardware:
   Will run for 120 seconds (12000 scans) at 100 scans/seconds
   Number of channels: 1
      index Type  Device    Channel MeasurementType      Range       Name
      ----- ---- --------- ------- --------------- ---------------- ----
      1     ai   cDAQ1Mod1 ai0     Voltage (Diff)  -10 to +10 Volts
```

Step 6. Change channel properties.

```
s.Channels.InputType='SingleEnded'

s =

Data acquisition session using National Instruments hardware:
   Will run for 120 seconds (12000 scans) at 100 scans/seconds
   Number of channels: 1
      index Type  Device    Channel   MeasurementType        Range       Name
      ----- ---- --------- ------- ------------------- ---------------- ----
      1     ai   cDAQ1Mod1 ai0     Voltage (SingleEnd) -10 to +10 Volts
```

Step 7. Start Acquisition and Plot the Data

```
[data,time] = s.startForeground;
plot(time,data)
```

The “s.startForeground” command starts the data acquisition engine for our input.
Every time you want to take a new set of data, you must re-run the “s.startForeground” command

Today's Lab exercise II:

1. Place your finger between the red LED of Figure 1 and the photodetector of Figure 2. Cover the system with a screen (or similar) to isolate it from the ambient light.

2. Practice collecting data from your fingers and displaying them in Matlab. Use a 100 Hz sampling frequency and collect data for approximately 120 seconds.

NOTE: This lab and next lab will have a single lab report. Here we practice and debug the full system,

next lab we will collect the real data and perform signal processing on the dataset.
SUGGESTION: Use this week to write up the majority of the lab report, such that you can easily incorporate next week results and data processing quickly.

LAB 10

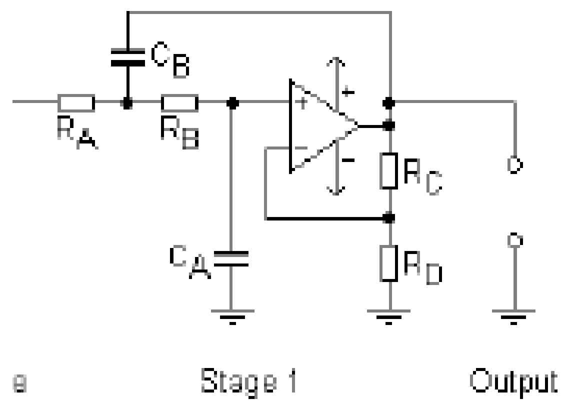

Figure 3. 2nd order 3dB ripple Chebychev low pass filter.

Exercise I

1. Replace the op-amp stage in Figure 2 with the filter shown in Figure 3. Use values as close as possible to: C_A=2.2u, C_B=1u, R_A=8.2kΩ, R_B= 12kΩ. Choose R_C and R_D to obtain a gain of 5. (The input to this filter is the emitter lead of Figure 2. With the above components, this filter has approximately a 15 Hz cutoff frequency).
2. Place your finger between the red LED of Figure 1 and the photodetector of Figure 2. Cover the system with a screen (or similar) to isolate it from the ambient light.
3. Using the same procedure as in Lab 9, use the NI DAQ and MATLAB to digitize the output signal and import it into MATLAB.
4. Create in MATLAB a 5^{th} order Butterworth high-pass filter with frequency cutoff of 1 Hz and filter the acquired data (use function butter).
5. Plot the data and compare with the RED signal of the Finger data from subject1a.mat from Lab 5. Discuss your results.