Signal Capture and Analysis

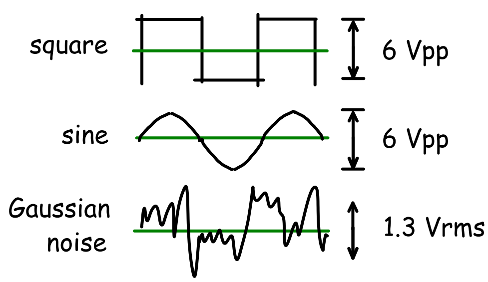
Introduction

One purpose of this lab is to capture waveforms with a digital oscilloscope and analyze the captured signals using numerical analysis software and spreadsheets.

Another is to measure a random signal. You're already familiar with measurements of predictable ("deterministic") signals such as sine and square waves. However, some signals are unpredictable ("random") and we cannot predict their future voltages or currents. Typical examples include noise or speech signals.

Although we cannot predict their values, we can often specify random signals' statistics, which are summaries of some relevant feature of the signal such as the fraction of time it has a certain voltage (the "probability density") or the amount of power the signal contains at certain frequencies (the "power spectrum").

In this lab you will measure some statistics of three waveforms that can be generated by the lab's waveform generator: a square wave, a sine wave, and Gaussian noise:



You will capture these waveforms with a digital oscilloscope. Then you will use Octave[1] to compute histograms that estimate the probability density functions of the waveforms. The histogram data will be imported into a spreadsheet that you will use to plot the histograms, compute the RMS powers from the probabilities and compare them to the powers measured using an RMS voltmeter and the oscilloscope's measurement function.

[1]You should also be able to use Matlab with some minor changes to the commands.

Software

Numerical Analysis Software

Numerical analysis software allows us to carry out a much wider range of analyses than can be done by an oscilloscope.

For this lab you will use Octave. Many other software packages, both commercial and open-source, are also available. Some of the most popular include Matlab (largely Octave-compatible), Mathematica, Maple, MathCAD, and R[2].

All of these programs have high-level language features such as support for working with complex vectors and matrices. These allow us to analyze signals using little or no programming.

Spreadsheets

You should be familiar with spreadsheets from previous courses. They are useful for manipulating and presenting limited amounts of data and for formatting numerical data for reports.

For this lab you can use Microsoft Excel or LibreOffice Calc.

CSV Files

Text files contain printable characters that form readable text. On the other hand, "binary" files contain data in formats that are only meant to be read and written by software.

A common text file format for exchanging numerical data between programs is called comma-separated values (CSV). Each line of the file corresponds to one row of a matrix or table and the values for each column are separated by commas.

Here's an example of a CSV file with data that could be used to fill a table consisting of four columns and three rows:

```
2.6,2.6, 1.3, .006
9.7, 1.734e-2, 1.1, 3
```

[2]See Wikipedia for a more complete list.

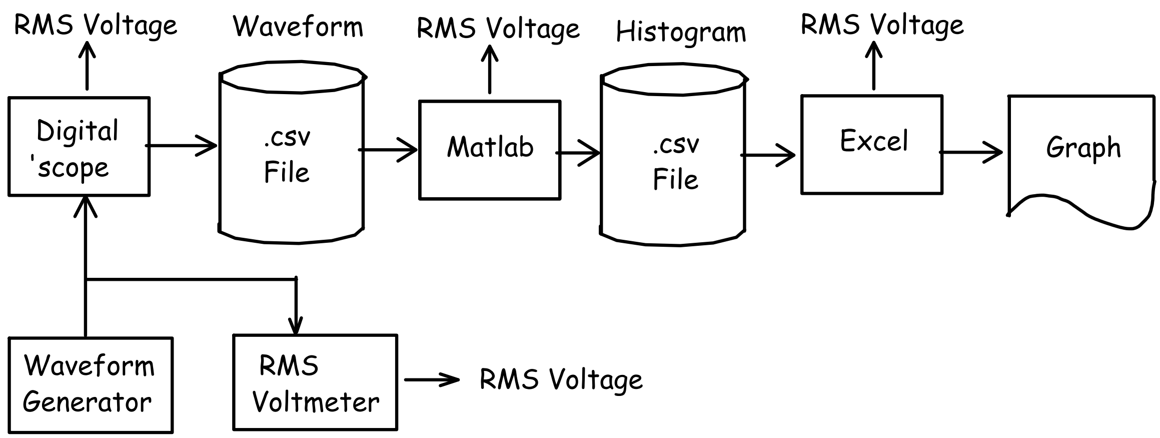


Figure 1: Signal and data flow.

```
-0.3, .87, 0.45, 22
```

Strings can be included in CSV files by quoting them but in this lab we will use files with numerical values.

Voltage Measurements

AC and DC Coupling

Most DMMs and 'scopes, have an "AC" (alternating current) setting that connects the input through a capacitor. This removes the average voltage and allows accurate measurements of small time-varying signals superimposed on a large constant voltage.

RMS and Average Voltage

An RMS voltage measurement squares the signal before averaging. This measures (the square root of) the power of the signal.

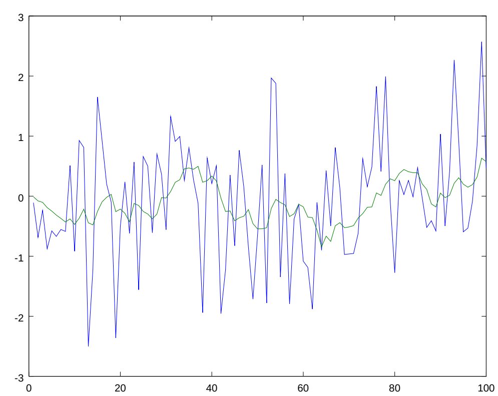
DMMs display the average voltage when DC coupling is used and the RMS voltage when AC coupling is used[3]. Some DMMs also have an "AC+DC" measurement that measures the RMS voltage with DC coupling.

Oscilloscopes can measure average and RMS voltages with AC or DC coupling as well as other types of voltages, such as peak-to-peak.

[3]If a DMM is not "true RMS," it still measures the average voltage and multiplies it by $^{2}/_{\pi\sqrt{2}} \approx {}^{0.707}/_{0.636} \approx 1.11$ which is the ratio of RMS to average voltage *for sine waves*. For other waveforms the RMS measurements will be incorrect.

Effect of Bandwidth

You will find that the power of the noise measured by the DMM does not match the power measured by the 'scope. This is because the noise signal's power is distributed over a frequency range (bandwidth) of tens of MHz while the DMM only responds to signals at low (audio) frequencies [4]. This means the DMM's input circuits "smooth out" the signal and the DMM measures a smaller voltage than is present at the input. For example, the following graph shows a noise signal before (blue) and after (green) low-pass filtering:



You can see that the low-pass filtered version has less variability (lower variance) and thus has less power than the unfiltered signal.

The bandwidth of the 'scope (also tens of MHz) is close to the bandwidth of the noise so the 'scope measurement is closer to the actual value.

[4]The DMM frequency response specifications are in the Specifications chapter in the manual available on the course web site.

Procedure

Remote Access

This year the lab will be done remotely. The equipment will have been connected up for you as described below. Go to `https://tcom.bcit.ca/workstation_status` to check workstation assignment and status and to `workspace.bcit.ca` to log in to your assigned workstation.

With File Explorer ( ), navigate to the `This PC > DATA (D:) > J_ELEX_3521`[5] folder, right-click on the file `fixperm.bat` and select "Run as administrator". This will fix some incorrect permissions that prevent use of Python scripts and change the Desktop background to make it more readable.

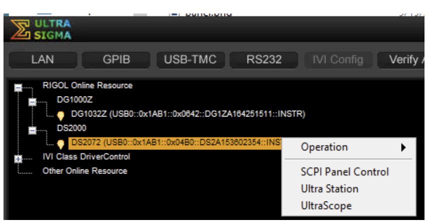
File Storage

You can upload and download files between the workstation and your computer using the menu items that appear when the mouse is near the top of the workstation display: . Note that files saved on the workstations will disappear each time they reboot; typically each evening. Store files that you want to keep on the `H:` (ShareFile) drive or download them to your computer before logging off.

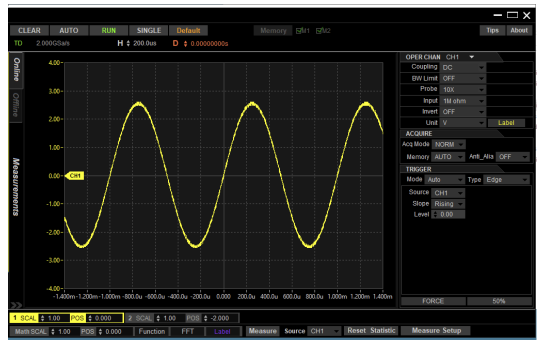
Instrument Control Software

Start up the remote interfaces for the three instruments you will be using:

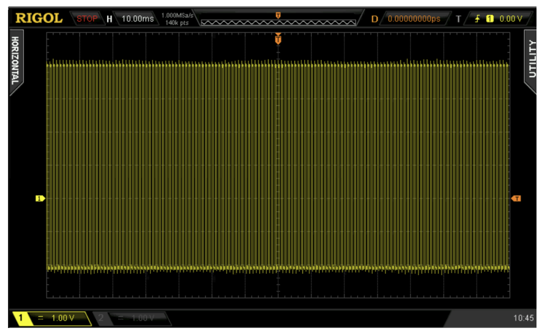
Rigol DS2072 Oscilloscope Run the Ultra Sigma utility from the shortcut on the desktop ( ). Right-click on the DS2072 instrument and select UltraScope:



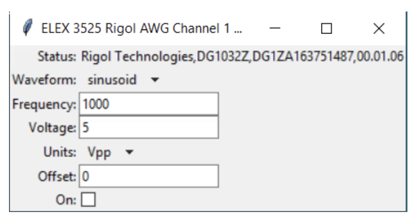
to start the remote interface to the Rigol DS2072 'scope:



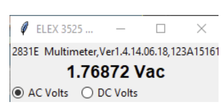
You can capture the instrument's actual display, rather than what is shown on your computer, by right-clicking on the waveform display and selecting "Print Instrument Screen". An example:



Rigol DG1032Z Arbitrary Waveform Generator Run the `awg.py` Python script. This will provide a simple interface that configures the AWG to output one of five waveforms with a configurable voltage and DC offset:



BK 2831E DMM Run the `dmm.py` script. This will display the DMM's AC or DC voltage readings:



The two scripts will open console windows that display error or progress messages. You can minimize these.

The software is not robust. You may need to restart the scripts, reset the instruments or verify the results by capturing the instrument display.

[5] If this link does not exist, go to `J:\ELEX\3521` directly.

Connections

As shown in Figure 1, The Channel 1 output of the waveform generator will have been connected to both Channel 1 of the 'scope and to the voltmeter. This lets you simultaneously view the waveform and measure its voltage on the DMM:

The 'scope will sample the input signal and digitize it (convert it to numbers). The RMS voltmeter will allow you to measure the signal's AC and DC voltages independently of the 'scope.

Square Wave

Set up the waveform generator for a square-wave output at a 1 kHz frequency with a 6 Vpp level and +1 V offset and the output turned on.

To capture samples for further processing:

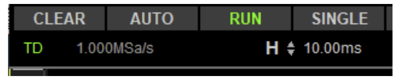
- set channel 1 on, channel 2 off by clicking on the numbers 1 and 2 along the bottom:

- In the **OPER CHAN** section, set up channel 1 for DC coupling with the appropriate probe scale factor (typically 10× — if you don't do this all of your measurements will be scaled by a factor of 1/10).

- set channel 1 **SCAL** to 1 V/div and **POS** to -0.5V or as appropriate for each waveform

- In the **ACQUIRE** section set Memory to 140k samples

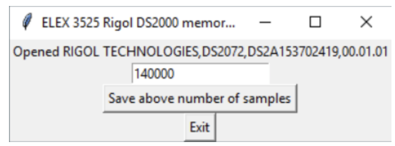
- On the status line at the top of the display, click on the up or down arrows beside the Horizontal scale control (**H**) until the sampling rate, shown to the left of **H**, is 1 MSa/s. This should correspond to 10ms/division:



- press the **RUN** button (it should change to **STOP**)

- Verify the instrument settings by getting a screen capture (right-click on the display and select **Print Instrument Screen**) before you proceed (see example above).

Run the `scopegrab.py` utility:



set the number of samples to download (use 140000 to match the memory depth setting above), and click on **Save above number of samples** to save the samples to a CSV file. You will be prompted for a file name[6].

Open the 'scope's measurement menu (using the tab on the left) click on **Frequency**, **Vamp** (amplitude), **Vavg** (average) and **Vrms** (RMS) to enable these measurements. The actual measurements are shown in a table when you click on the **Measure** button along the bottom.

You must adjust the horizontal scale to get accurate voltage measurements. The oscilloscope's voltage measurements are based on the displayed, rather than captured, samples. The 'scope's measurements will be incorrect if the horizontal scale is too small (because the display does not show an integral number of periods) or too large (because each displayed point combines adjacent samples).

Compare the 'scope readings to the AWG settings and the DC and AC voltages on the DMM.

Right-click on the screen and select "Print Instrument Screen" to get a screen capture of the instrument's actual screen. Use the "Save to File" button to save the screen capture.

As described above, these files will be lost if left on the `C:` drive.

Record the date and time, the type of waveform, the voltage shown on the DMM, the measured values from the 'scope and the names of the screen image (`.bmp`) and waveform capture (CSV) files. You will need this information for your report.

[6] For example, you could name it `square.csv`.

Sine Wave

Now configure the signal generator to a sine-wave output with (peak-to-peak) voltage of 6 V and zero DC offset. Repeat the steps above for the sine wave, saving both image and waveform capture files and recording the 'scope and DMM readings.

Noise Waveform

Repeat after configuring the waveform generator for (Gaussian) noise output with a level of 1.3 Vrms and zero offset. Again, save the image and waveform capture files and record the same information as before.

You should now have three sets of voltage readings from two instruments, three display files and three waveform files.

The rest of the lab can be done without the test equipment but I recommend you try to do as much of it as possible during the lab session in case you need to repeat some of your measurements.

Import Samples into Octave

Run Octave (GUI). Use the File Browser to change to the folder that contains your waveform capture CSV files. Right-click on a file and select **Open in Text Editor**. Your file should look something like:

```
0,-2.04
1e-06,-2.04
2e-06,-2.04
3e-06,-2.04
4e-06,-2.04
```

Right-click on the file and select **Load Data**. This will create a variable with the same name as the file (e.g. `square`) and dimensions 140000×2. Right-click on the variable name in the **Workspace** tab and select **Open in Variable Editor**. In the Variable Editor, right-click on the column header for the first column (the sample time values) and select **Delete Column** to get rid of the time data since we won't use it. In the variable tab the variable should now have dimensions 140000×1.

Switch to the **Command Window** tab and enter the following command:

```
plot(square(1:2000))
```

to plot the first 2000 samples of the waveform to make sure you captured and have read the waveform [7].

[7] Use the appropriate variable name instead of `square`.

Use the **File > Save As...** menu item to save the plot to an image file (e.g. `square.png`) so that you can include it your report.

In the command window, use the following commands to compute the mean and RMS values of the captured waveform and record the results for your report:

```
mean(square)
std(square)
```

Compute the Histogram

A histogram is a plot of the number of times a value appears in a set of values. For example, a histogram of BCIT student ages would show the number of 18-year old students as the y-value for the x-value 18.

If a random value is continuous we must group the values into ranges of values called "bins".

In our case the binning is actually done by the 'scope's 8-bit analog-to-digital (A/D) converter which quantizes the input signal into $2^8 = 256$ discrete voltage levels.

The voltage levels are separated by steps called the "quantization step size". For example if the step size was 0.08 volts, a particular measurement might include all voltages between 1.60 and 1.68 volts. An input voltage of 1.63 volts would be counted as falling in this range.

When computing the histogram of the signal we should use a bin width equal to the step size to make sure we include exactly one voltage range per bin.

We can find the step size by examining the voltage levels measured by the 'scope. The Matlab function `unique()` returns a sorted list of the unique values in the input. Assuming we have at least two values separated by the step size we can find the step size as the minimum difference between successive unique values. Enter the following in the command window to do this:

```
ssize=min(diff(unique(square)))
```

where the sequence of unique, diff and min operations computes the minimum difference between successive unique measured voltages.

The step size can then be used to compute the number of histogram bins we should use:

```
nbin=(max(square)-min(square))/ssize + 1
```

Now we can compute a histogram of the voltage waveform using the `hist()` function:

```
[n,v]=hist(square,nbin);
```

This will return a vector `n` with the number of samples values in each histogram bin and a vector `v` with the voltages at the midpoints of each bin.

We can then export the histogram to a CSV file by opening a file for writing:

```
fo=fopen('lab1square.csv','w')
```

and writing the values to the file as strings separated by commas, two per line:

```
fprintf(fo,'%f, %d\n',[v;n])
```

and closing the file:

```
fclose(fo)
```

Note that this histogram CSV file is different than the one containing the sample values; don't overwrite the data in the CSV file that contains your captured data!

Repeat for the sine wave and Gaussian noise capture files, saving them to differently-named CSV files and recording your results for the mean and standard deviation (rms voltage). You can use the command history window to select previous commands and edit them to reduce typing.

Compute RMS Voltage

The DC-coupled RMS (root mean square) voltage of a signal is defined as the square root of the mean (average) of the square of the voltage. If all voltages had the same probability we could just add up the squares and divide by the number of voltages.

However, in this case different voltages have different probabilities so we must scale each squared voltage (normalized power) by the probability of that power and then compute the sum. For example, if 20% of the values had a power of 0.3W and 80% had a power of 0.2W then the average (normalized) power would be $0.2 \times 0.3 + 0.8 \times 0.2 = 0.06 + 0.16 = 0.22$.

We can use a spreadsheet and the histograms stored in the CSV files to compute the RMS values of the three waveforms. This is a another check on the values measured by the instruments.

Run Excel or LibreOffice Calc. Open the CSV file (**File > Open**) which will result in a spreadsheet with two columns, the first with the bin counts and the second with the bin voltages.

> *Note: Immediately save the file in `.xlsx` format or you will lose all your work when you exit the spreadsheet software!*

Enter a formula at the bottom of the first column to compute the total number of samples (this should be 140,000). On each row add a cell with a formula that divides the count by this value. This converts the bin counts to probabilities[8]. The probabilities should sum to 1. Then add cells on each row that compute the square of each bin voltage multiplied by the probability of that voltage. Compute the sum of these scaled squared voltages. This is the mean square voltage. Finally, compute the square root of the mean square voltage. This is the RMS voltage of the (DC-coupled) waveform.

Here is an example showing the spreadsheet columns containing bin voltages, bin counts, bin probabilities and scaled squared voltages. The bottom row contains sums of the bin counts, probabilities, and the mean-square and RMS voltages.

203	3.90066	5	3.5714E-005	0.000543404	
204	3.940485	1	7.1429E-006	0.00011091	
205	3.980291	4	2.8571E-005	0.000452649	
206	4.020097	4	2.8571E-005	0.000461748	
207		140000	1	1.793218566	1.33911111

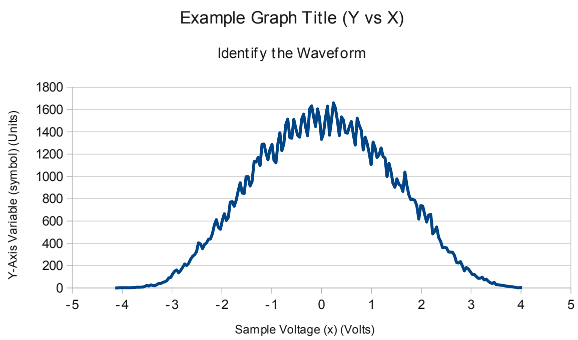
Repeat using the histogram bin counts for the sine and Gaussian noise waveforms, importing their data into different sheets of the same document (**Data > From Text/CSV** or **Import > Sheet from File**). Compare your results to the values read from the 'scope and DMM.

Plot Histograms

Create three line graphs (called "charts" in spreadsheets) with curves showing the values of the three histograms. Use **Insert > Chart**, set the chart type to an XY Line graph and use the voltages as the X values and the histogram sample counts as the Y values.

[8]To insert a reference to an absolute rather than relative cell, prefix the column or row with a dollar sign ($). For example, A101.

Add an accurate title, sub-title and X and Y axis labels. Adjust the chart properties so it is formatted as shown below:



Lab Report

Upload three files to the dropbox folder for this lab:

1. the histogram CSV file for the sine wave input in CSV (`.csv`) format (only the sine wave file)
2. the Excel (or LibreOffice Calc) spreadsheet file in Excel (`.xls` or `.xlsx`) format (any version) containing the three sheets and three charts
3. your report in PDF (`.pdf`) format

Your report should include:

- identification information as described in the course information handout,
- plots of the first 2000 samples of each of the three sampled waveforms,
- 'scope screen captures for the three waveforms including the measurement data,
- the histograms for the three waveforms created by your spreadsheet (if possible, use copy and paste, not a screen capture)
- a table comparing the RMS voltages measured by the DMM, the 'scope, Octave (or Matlab) and computed by your spreadsheet from the histogram. Include one row for each waveform.
- a brief explanation for any discrepancies between the results for the different types of measurements.