PHYS2641 – Laboratory Skills and Electronics

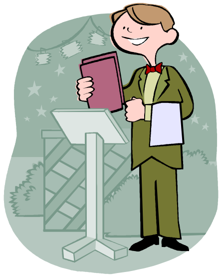
Electronics

Lecture 1

Dr Aidan Hindmarch
a.t.hindmarch@durham.ac.uk
November 2019

Durham University

Today's menu



Aims:

1. Course background and Jupyter Notebook system
2. RC filters
3. Transfer function and Bode Plots

Time at the end to discuss Skills reports

Durham University

Course aims – 'Research Skills'

Officially:

'to teach electronics as a theoretical and a practical subject'

In reality:

We want to teach basic electronics as a practical skill/tool for physics research...

- to be able to connect and use commonplace lab instrumentation
    - *Oscilloscope, Multimeter, Power Supply, Signal-Generator etc.*
- to understand simple circuit diagrams and be able to set up simple, useful, electronic circuits based on them

Ability to work through a practical problem in an ordered, logical, fashion – exactly as one would in experiment design, computing, data analysis, etc.

Electronics course overview

LECTURES MAP ONE-TO-ONE TO LAB PRACTICALS

Lectures		Labs
1. Intro & passive circuits	→	Passive Filters
2. Operational amplifiers	→	Operational amplifiers 1
3. Operational amplifiers (cont.)	→	Operational amplifiers 2
4. Modulation	Christmas vacation	Modulation
5. Phase-sensitive detection	→	Assessed practical (week 1)
		Assessed practical (week 2)

Durham University

Online Jupyter Notebook system will be used for all of the practical classes.

Benefits:

- Replaces printed laboratory scripts and (largely) laboratory notebooks
- NO python programming is required! (from you, I have done it all)
- You do NOT need to do battle with Excel (or matplotlib) to plot graphs!

Drawbacks:

- None! So long as you use the system sensibly…

Jupyter notebooks on shared PCs

Treat the Jupyter system like you online banking!

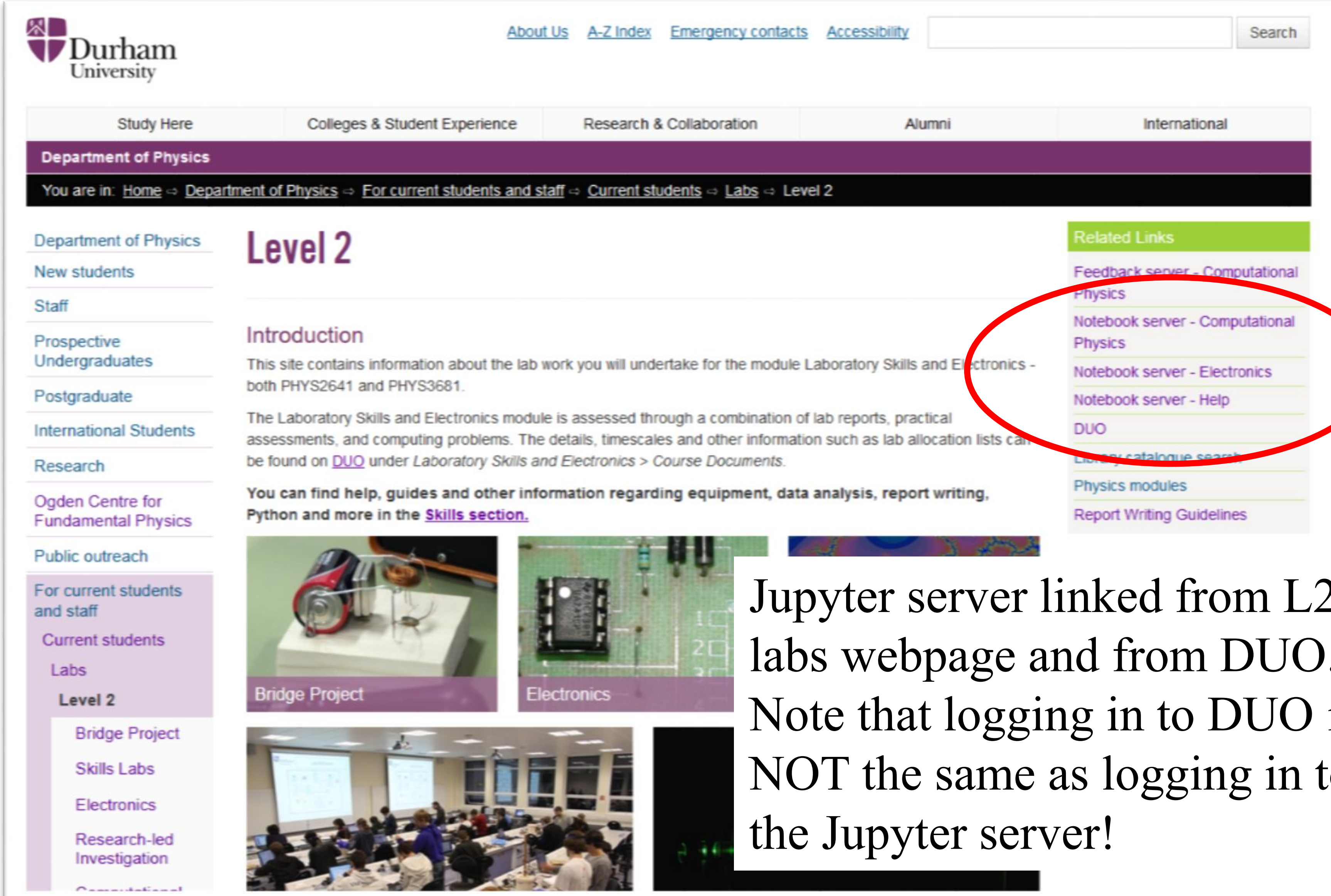
You need to ensure that you log in to the Jupyter system under **your own** user account: If you have NOT entered your CIS username and password at the Jupyter prompt, you are not logged in to your account.

You also NEED to explicitly LOG OUT at the end of the session – closing the browser and turning off the computer is insufficient, you remain logged in when the computer restarts.

The result of this is that if, say, a student in Monday labs doesn't log out, you can end up with a student on Tuesday labs overwriting all of their data.

The end result of this is that NEITHER student any longer has access to the data that they collected in the lab!

Durham University



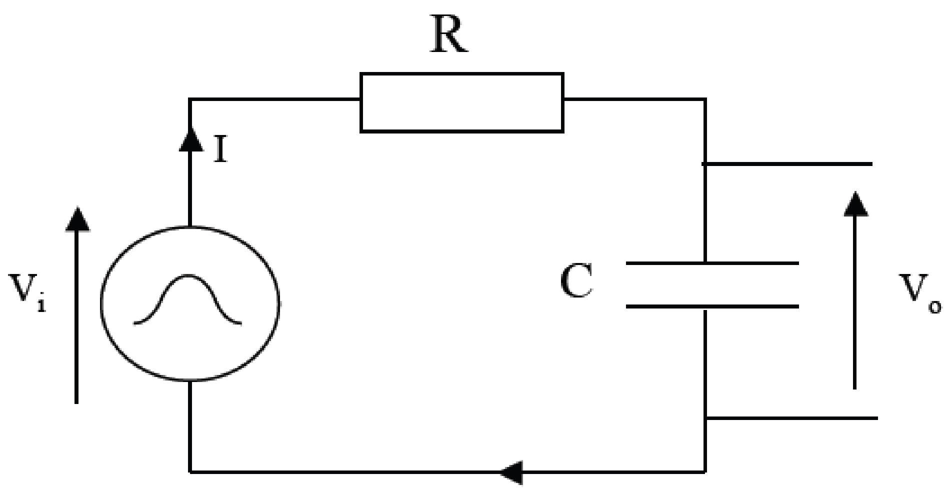
Jupyter server linked from L2 labs webpage and from DUO. Note that logging in to DUO is NOT the same as logging in to the Jupyter server!

Durham University

Now I’ll try to do a LIVE demo of how we will use Jupyter Notebooks in electronics labs…

Durham University

a.c. circuit analysis



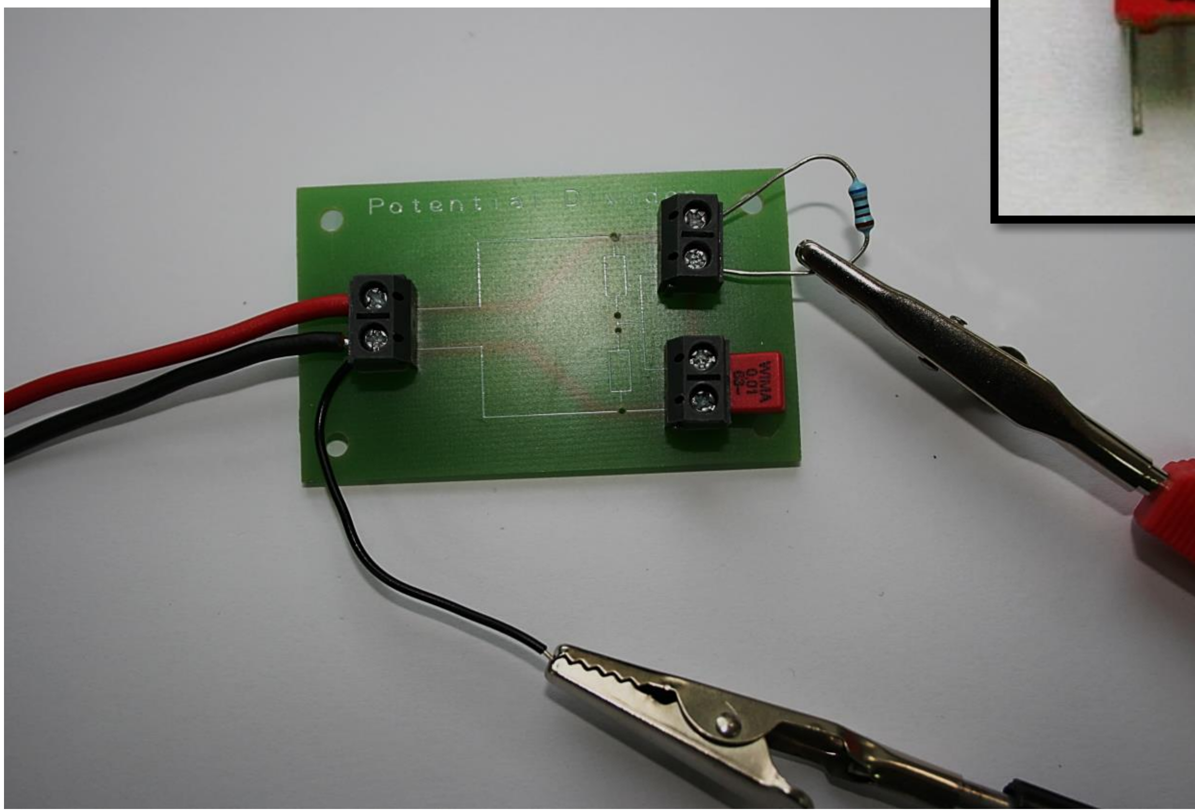
This is a potential divider...

$$V_0 = \frac{X_C}{R + X_C} V_i$$

To determine the output of a reactive circuit, we use the same method as for a resistive circuit, only using reactance (impedance) rather than resistance

But, both V_{i} and V_0 are now ***complex***!

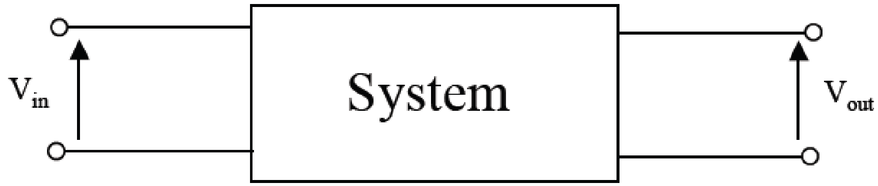
Real life



Durham
University

a.c. circuit analysis (2)

Transfer functions



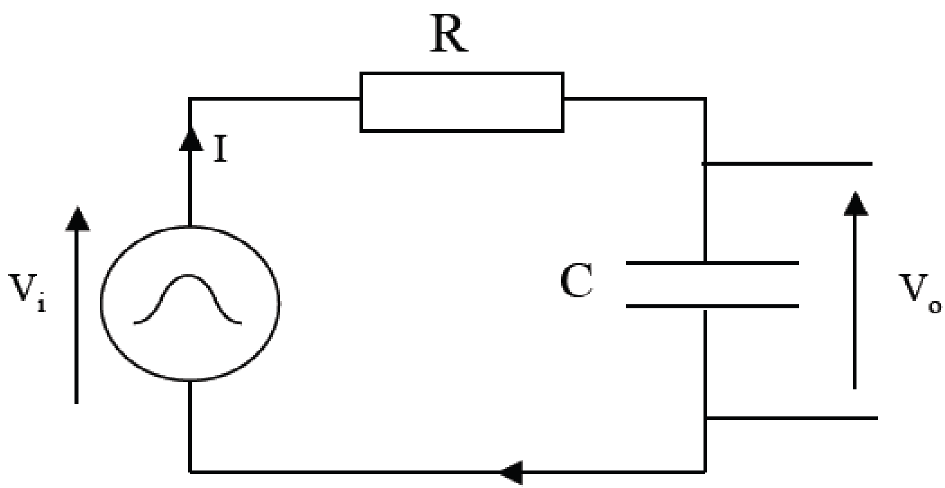
- The behaviour of a linear electronic system is often described by its ***Transfer Function***
- The transfer function, ***H(ω)****, describes the* ***Gain*** and the ***Phase shift*** of the system as a function of frequency, $\omega = 2\pi f$

  - The **gain** of a system is the magnitude ratio of the output to the input voltage
  - The **phase shift** of the system is the change in phase of the output voltage relative to the input voltage

$$\text{Gain} = \left|H(\omega)\right| = \left|\frac{V_{out}}{V_{in}}\right| \qquad \text{Phase shift} = \arg\left[H(\omega)\right] = \arg\left[\frac{V_{out}}{V_{in}}\right]$$

Durham University

RC circuit: transfer function



$$H(\omega) = \frac{V_0}{V_i} = \frac{X_C}{R + X_C}$$

One can show that $\boxed{|H(\omega)| = \frac{1}{\sqrt{1+\omega^2 C^2 R^2}}}$ and $\arg\left[H(\omega)\right] = \tan^{-1}(-\omega CR)$

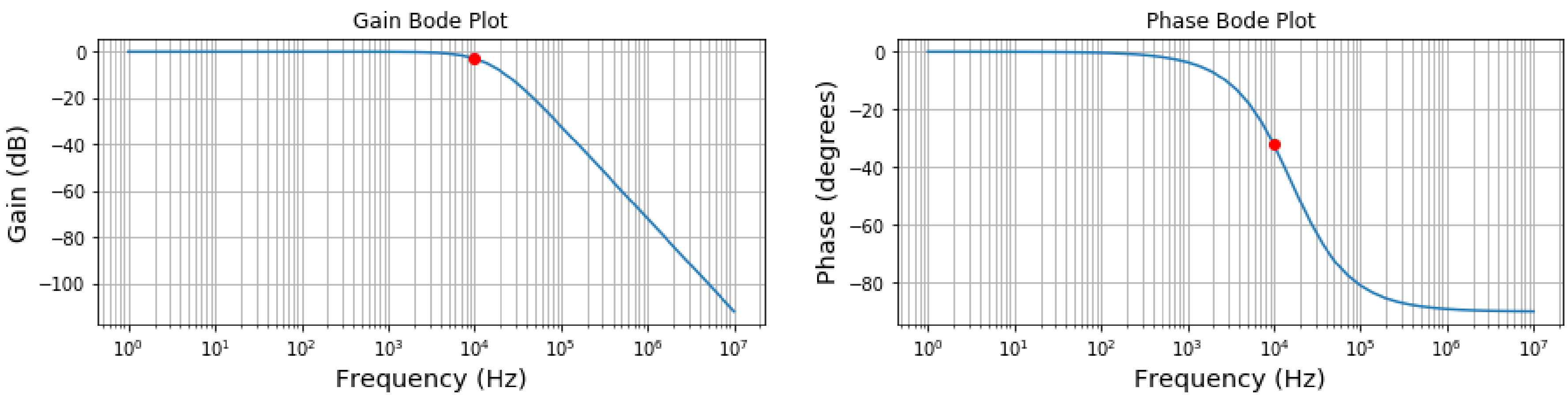
$|H(\omega)|$ decreases as frequency increases. High-frequency voltage signals are not transferred

This is a ***Low-pass filter***

Durham University

Bode plots

Transfer function is commonly represented graphically using ***Bode plots***.



Bode plots show the frequency dependence of the *gain* and *phase*: H(ω)

Broad frequency range – 7 decades of frequency in this case

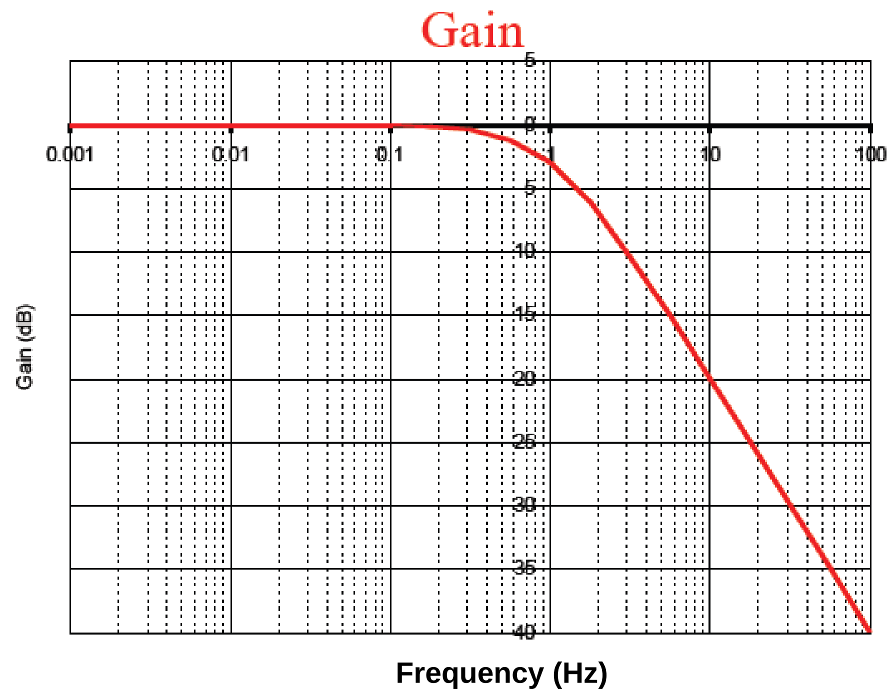
Durham University

Bode plots (2)

Frequency is *always plotted on a log. axis*

Gain is normally given in decibels (dB): dB units are a log form

Bode plot is essentially a log-log plot



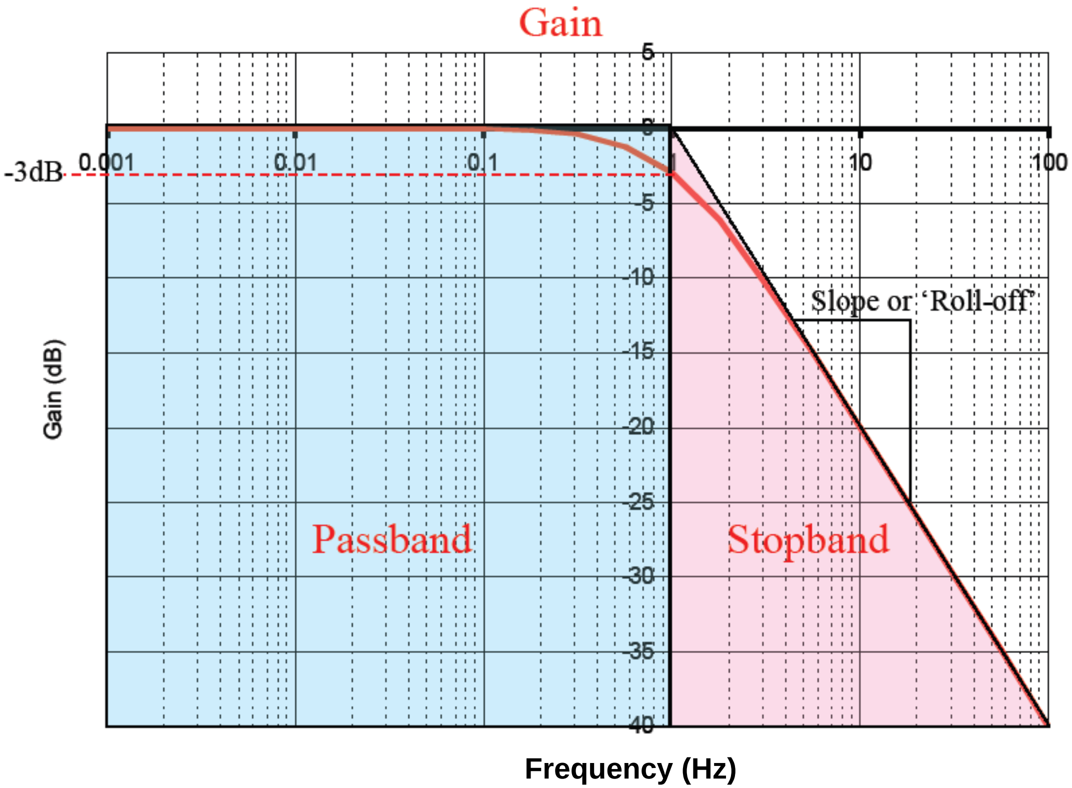
dB units are primarily used for power (intensity): results in **'20-log' rule** for voltage gain:

$$\text{Gain}_{\text{dB}} = 10\log_{10}\left(\frac{P_{out}}{P_{in}}\right) = 10\log_{10}\left(\frac{|V_{out}|^2 / R}{|V_{in}|^2 / R}\right) = 10\log_{10}\left(\frac{|V_{out}|^2}{|V_{in}|^2}\right)$$

$$= 20\log_{10}\left(\frac{|V_{out}|}{|V_{in}|}\right) = 20\log_{10}\left(|H(\omega)|\right)$$

Durham University

Bode plots (3)



A filter has one or more pass- and stop-bands

The transition between the two is defined by the '-3dB point' ('corner' or 'cut-off' frequency) – the point at which the gain drops by 3dB from the maximum value

Note: -3dB corresponds to

$$|H(\omega)| = \frac{1}{\sqrt{2}} \approx 0.707$$

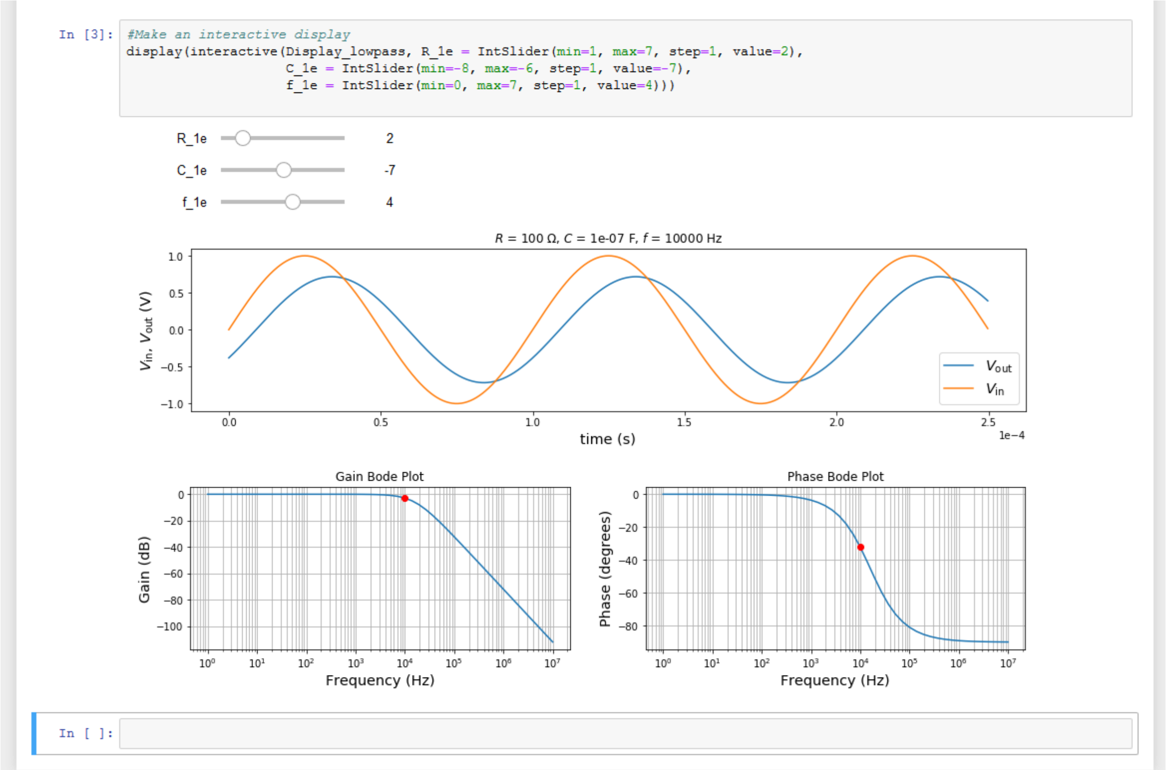
The stop-band is characterised by a slope, or 'Roll-off', normally quoted in dB/decade

This filter is characterised as: low-pass, F_{-3dB} = 1 Hz, roll-off = -20 dB/dec

Durham University

Jupyter Demo: Example_RC_filter.ipynb

Let’s try a live demonstration of RC filter response and Bode plots using an interactive Jupyter Notebook



The interactive Notebooks used throughout the course will be available on the Notebook server for you to try out.

Bode plots: Python/matplotlib

Example notebook on server: simple example data, also includes model uncertainty calculations, can be used in 'normal' python code also.

Data format is consistent (compatible) with the chi-squared notebook that you have previously used in Skills… and plots include error-bars!

Bode plots using Python

In [1]:
```
#Import required modules etc.
%matplotlib inline
import matplotlib.pyplot as pyplot
import numpy
```

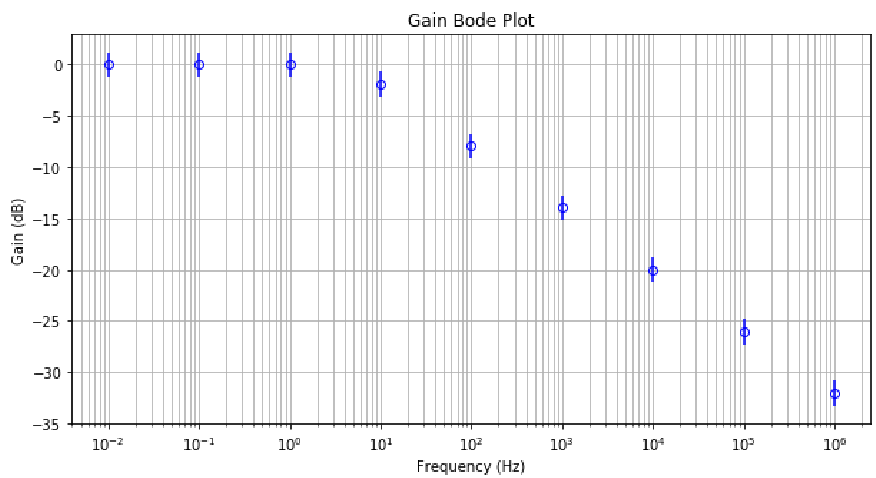
In [2]:
```
# Generate frequency data: take a measurement in steps of one decade of frequency.

Frequency = numpy.array([1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3, 1e4, 1e5, 1e6]) # [Hz]
```

Gain Bode Plots:

In [3]:
```
# Generate voltage data and errors.

V_in = 5.0 # [V] : Amplitude of input voltage, kept fixed

V_out = numpy.array([5.0, 5.0, 5.0, 4.0, 2.0, 1.0, 0.5, 0.25, 0.125]) # [V] : Example measured output voltages

Voltage_percentage_err = 10 # [%] : Let's assume that the voltage measurement error is 10%;
                            # this is a pretty poor measurement, but as an example shows up clearly on plot.

V_in_err = V_in * Voltage_percentage_err/100 # [V]
V_out_err = V_out * Voltage_percentage_err/100 # [V]
```

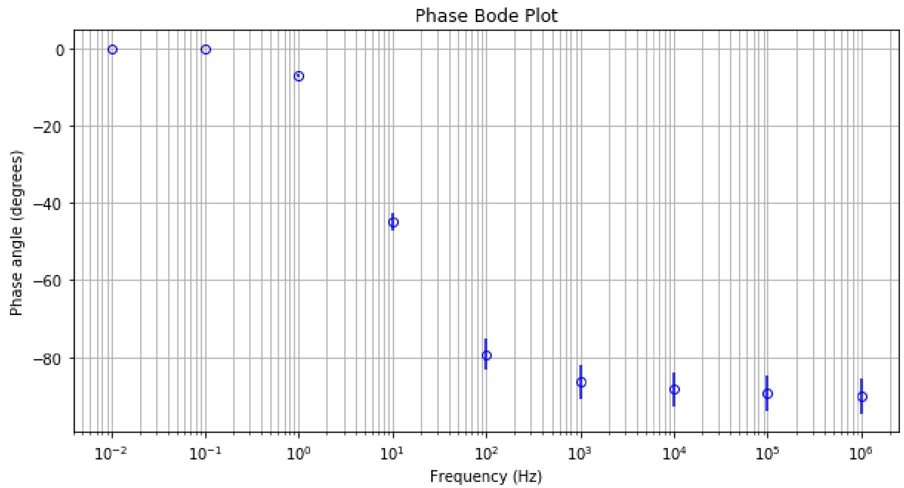
Durham University

In [8]:
```
# Plot a figure: Bode plot is dB gain as a function of frequency, with frequency plotted on a log scale.

pyplot.figure(figsize = (10,5))

pyplot.errorbar(Frequency,            # Errorbar plot: x axis is Frequency [Hz]
                dB_gain,              # y axis is dB_gain [dB]
                dB_gain_err,          # error-bars on y axis [dB]
                marker="o",           # round, unfilled markers
                fillstyle = "none",
                color="blue",         # markers and errorbars same colour for clarity
                ecolor = "blue",
                linestyle="none")     # Do NOT show connecting lines between datapoints!

pyplot.xscale('log', basex=10)  # log 10 scale for x axis
pyplot.yscale('linear')  # explicitly set (default) linear scale for y axis

pyplot.grid(which='both', #gridlines for both 'major' and 'minor' tick labels
            axis='both') #gridlines for both 'x' and 'y' axes

pyplot.title("Gain Bode Plot") # Title
pyplot.xlabel("Frequency (Hz)") # Axis labels [Hz]
pyplot.ylabel("Gain (dB)") # [dB]


pyplot.show()
```



Durham University

In [12]:
```python
# Plot a figure: Bode plot is phase angle as a function of frequency, with frequency again plotted on a log scale.

pyplot.figure(figsize = (10,5))

pyplot.errorbar(Frequency,            # Errorbar plot: x axis is Frequency [Hz]
                phase_shift,              # y axis is Phase shift [degrees]
                phase_shift_err,          # error-bars on y axis [degrees]
                marker="o",           # round, unfilled markers
                fillstyle = "none",
                color="blue",         # markers and errorbars same colour for clarity
                ecolor = "blue",
                linestyle="none")     # Do NOT show connecting lines between datapoints!

pyplot.xscale('log', basex=10)  # log 10 scale for x axis
pyplot.yscale('linear')  # explicitly set (default) linear scale for y axis

pyplot.grid(which='both', #gridlines for both 'major' and 'minor' tick labels
            axis='both') #gridlines for both axes

pyplot.title("Phase Bode Plot") # Title
pyplot.xlabel("Frequency (Hz)") # Axis labels [Hz]
pyplot.ylabel("Phase angle (degrees)") # [degrees]

pyplot.show()
```



Bode plots: Lab Jupyter Notebooks

Throughout the labs course Bode plots will be generated ‘automatically’ from data that you input: no python programming required!

Remove last line

Frequency (Hz)	Frequency Error (Hz)	V_{in} (mV)	V_{out} (mV)	Time Shift (µs)	Time Shift Error (µs)
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Add line to table

Save data

Precision of V measurement (mV): 0

The 'Generate Bode Plots' button below produces Bode plots for gain and phase-shift from your tabulated data (also ensuring the data is saved first).

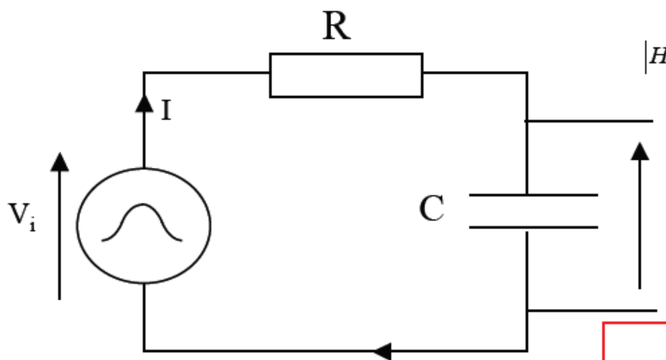
The 'Print gain and phase data' button which then appears allows you to view the numpy arrays containing your data; you can copy and paste these into e.g. another Notebook if you want to make your own bode plots - for example to try to fit the frequency dependence using chi-squared analysis.

Generate Bode Plots

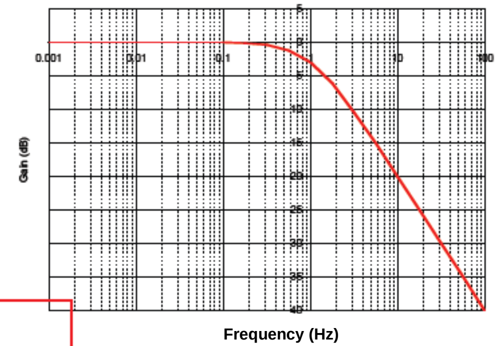
Let’s try a ‘live’ (and ‘online’) example using this week’s lab notebook!

Durham University

RC filters

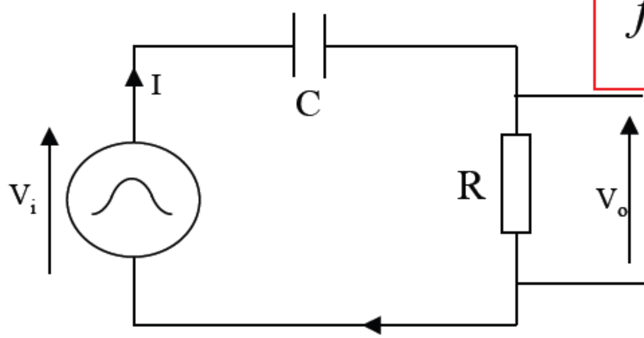


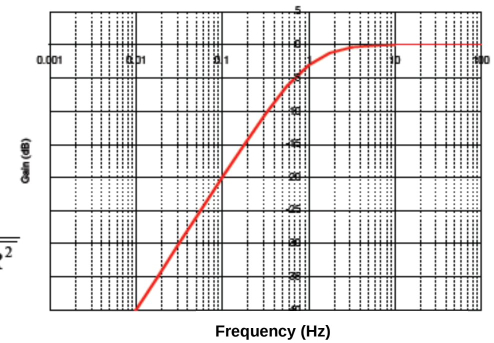
$$|H(\omega)| = \frac{1}{\sqrt{1+\omega^2 C^2 R^2}}$$



Low-pass filter

$$\boxed{f_{-3dB} = \frac{1}{2\pi RC}}$$





$$|H(\omega)| = \frac{\omega CR}{\sqrt{1+\omega^2 C^2 R^2}}$$

High-pass filter

Filter types

Low frequencies removed	➡	High-pass filter
High frequencies removed	➡	Low-pass filter
Frequencies in middle of spectrum removed	➡	Band-stop filter
Frequencies at both ends of spectrum removed	➡	Band-pass filter

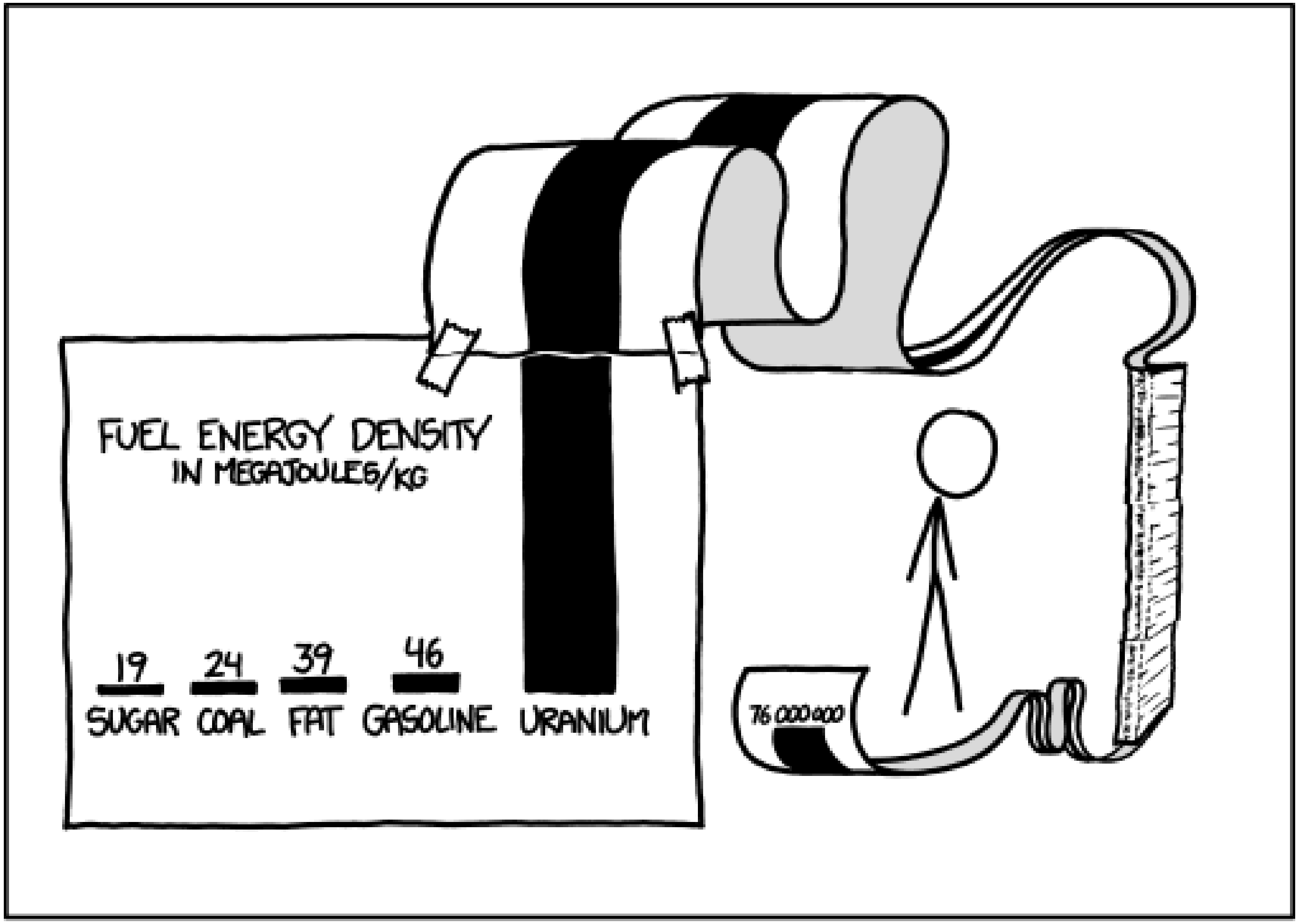
Durham University

Summary

- Jupyter Notebook system used for electronics labs
- Potential divider circuits & passive filters
- Transfer function, Bode plots: Electronics labs Jupyter Notebooks & underlying Python/matplotlib code

Next week: Control systems & operational amplifiers

Durham University



SCIENCE TIP: LOG SCALES ARE FOR QUITTERS WHO CAN'T FIND ENOUGH PAPER TO MAKE THEIR POINT *PROPERLY*.

https://xkcd.com/1162/