SAGI Summer School 2024
IC/DAQ & Electronics Project

Presenters:

Duong Dang Le
Ngan Le
Nguyen Nguyen Duc
Tin Mai Nhu

Advisors: Sam Codon, Logan Foote, and Hien Nguyen

Outline

I. Introduction: Motivation and background

II. Filter design

III. Data acquisition

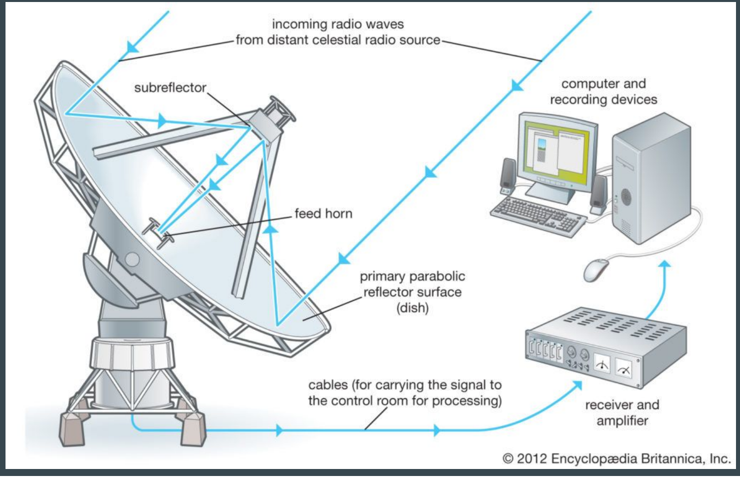
1. Overview of instruments
2. Softwares
3. Measurement procedures

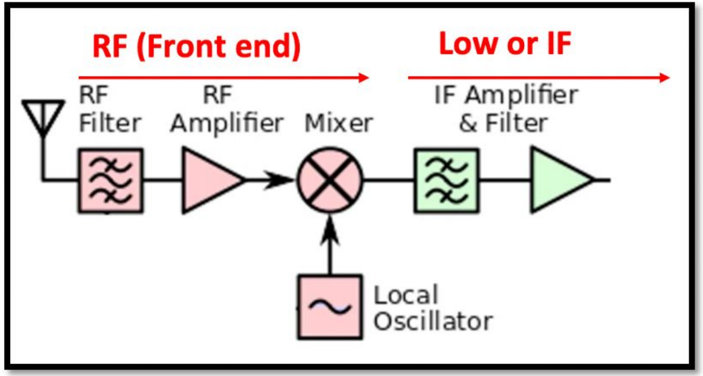
IV. Data analysis

1. FFT: Limitations
2. Transmission
3. Results:

V. Conclusions.

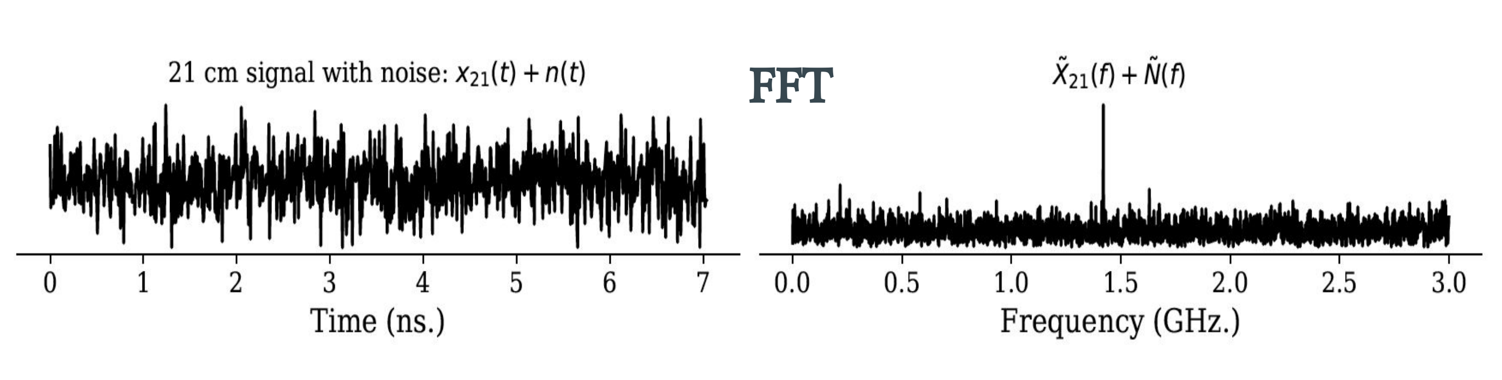
I. Introduction

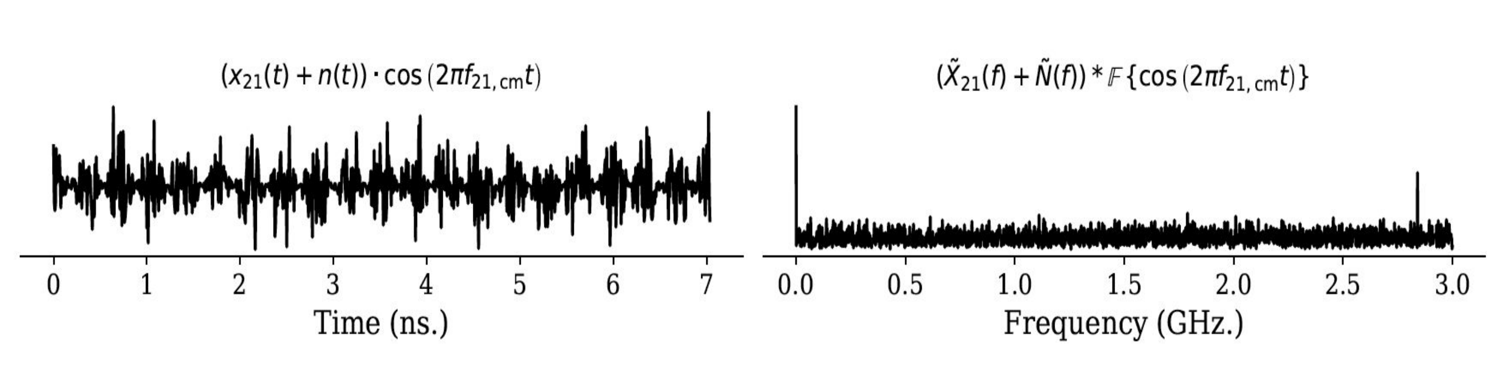




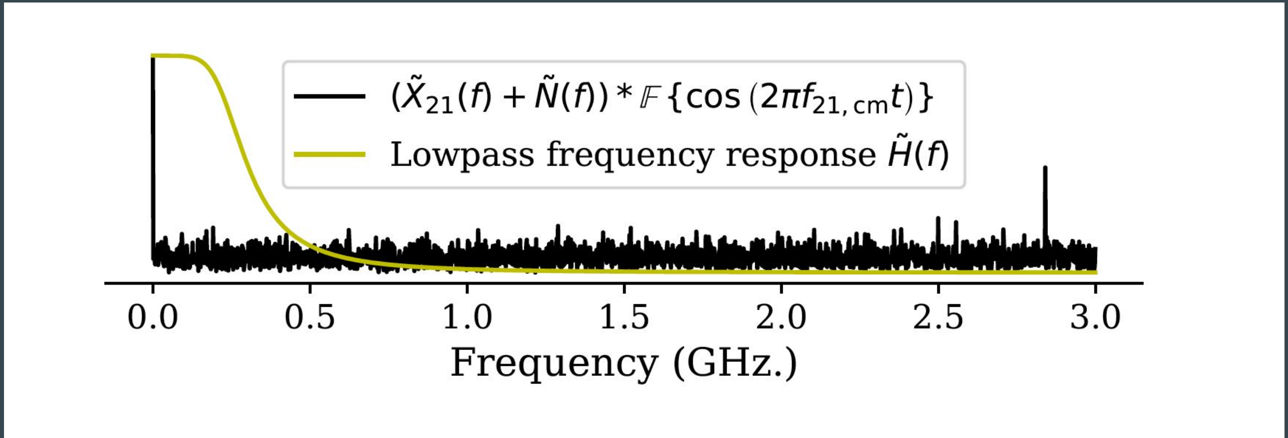
- Received digital signal is discrete → need to be sample to get continue one.
- Sometimes observed signal is at high frequency → need to low down the frequency by "*mixing* "
- Filtering is also important to extract the signal from noise.

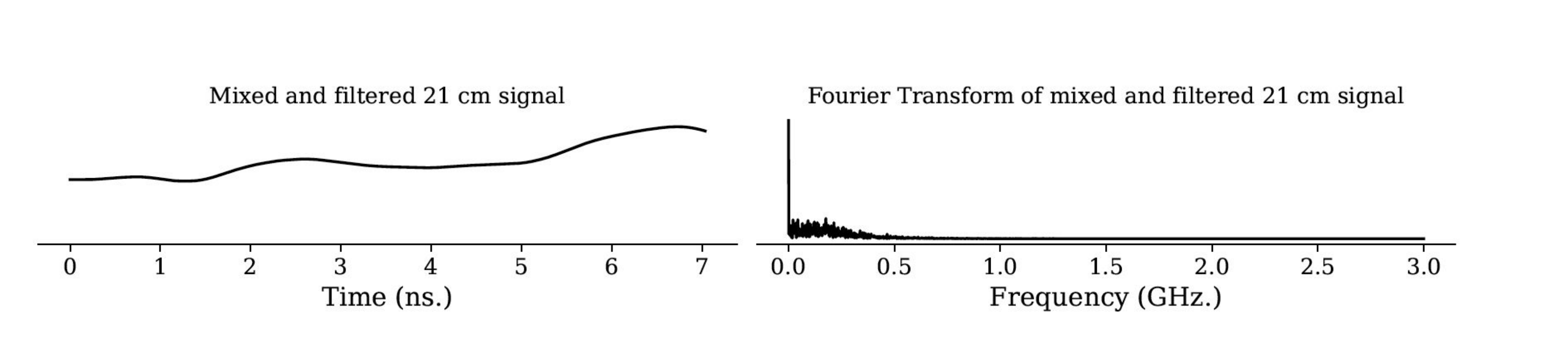
I. Introduction



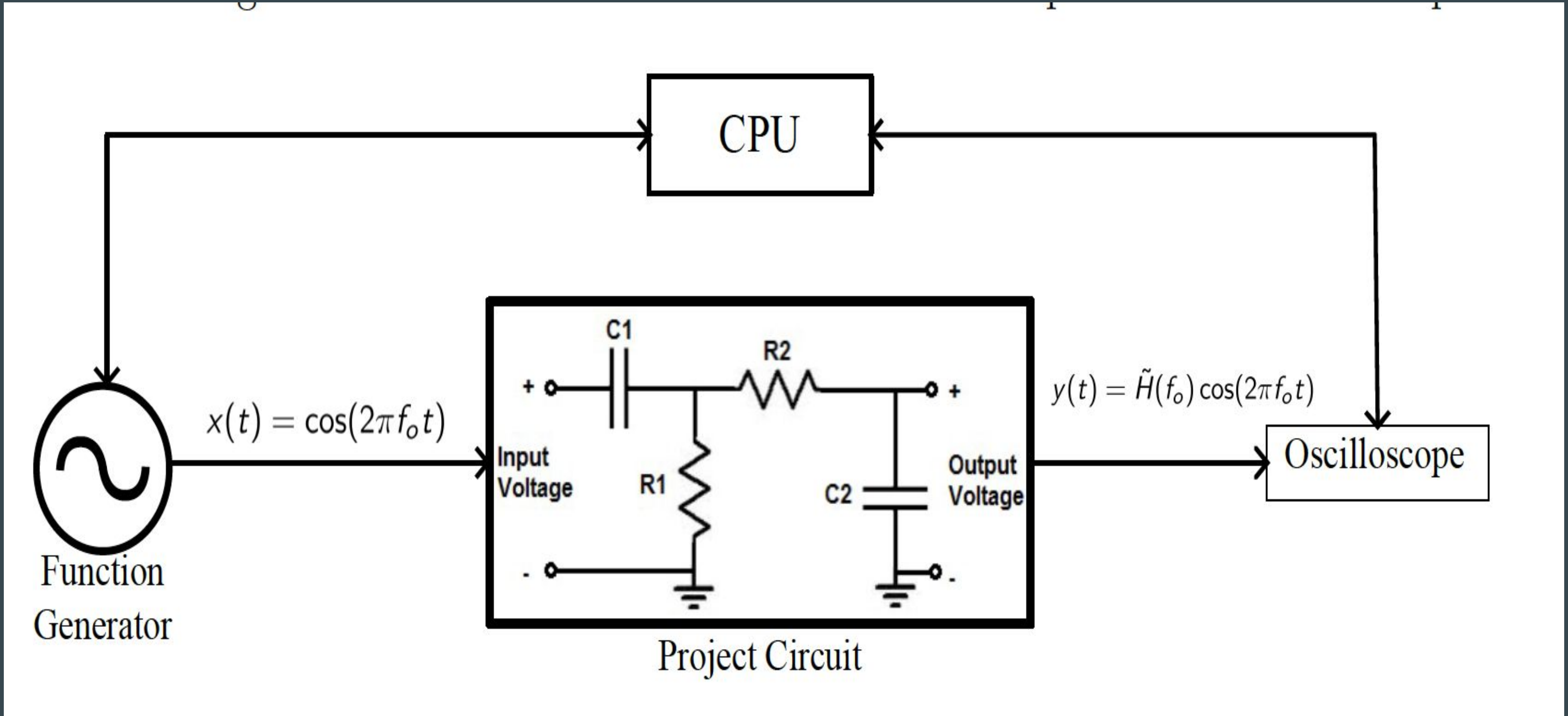


I. Introduction





I. Introduction



Measurement scheme

II. Filter design for IC/DAQ

Requirement:

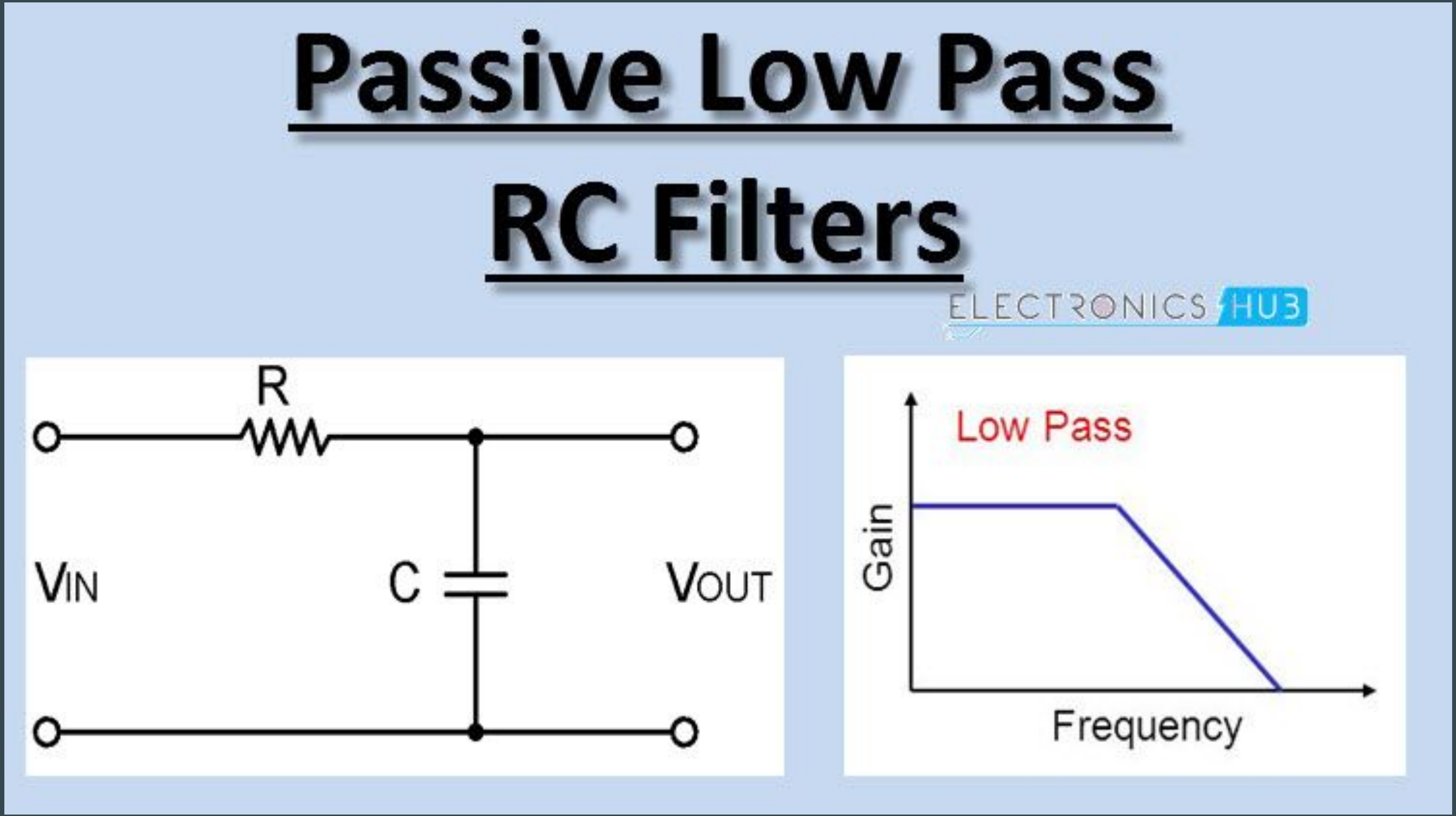
Bandpass filter: 3dB from 6 kHz to 65 kHz

IC/DAQ: Sweep from 1 kHz to 200 kHZ to obtain response

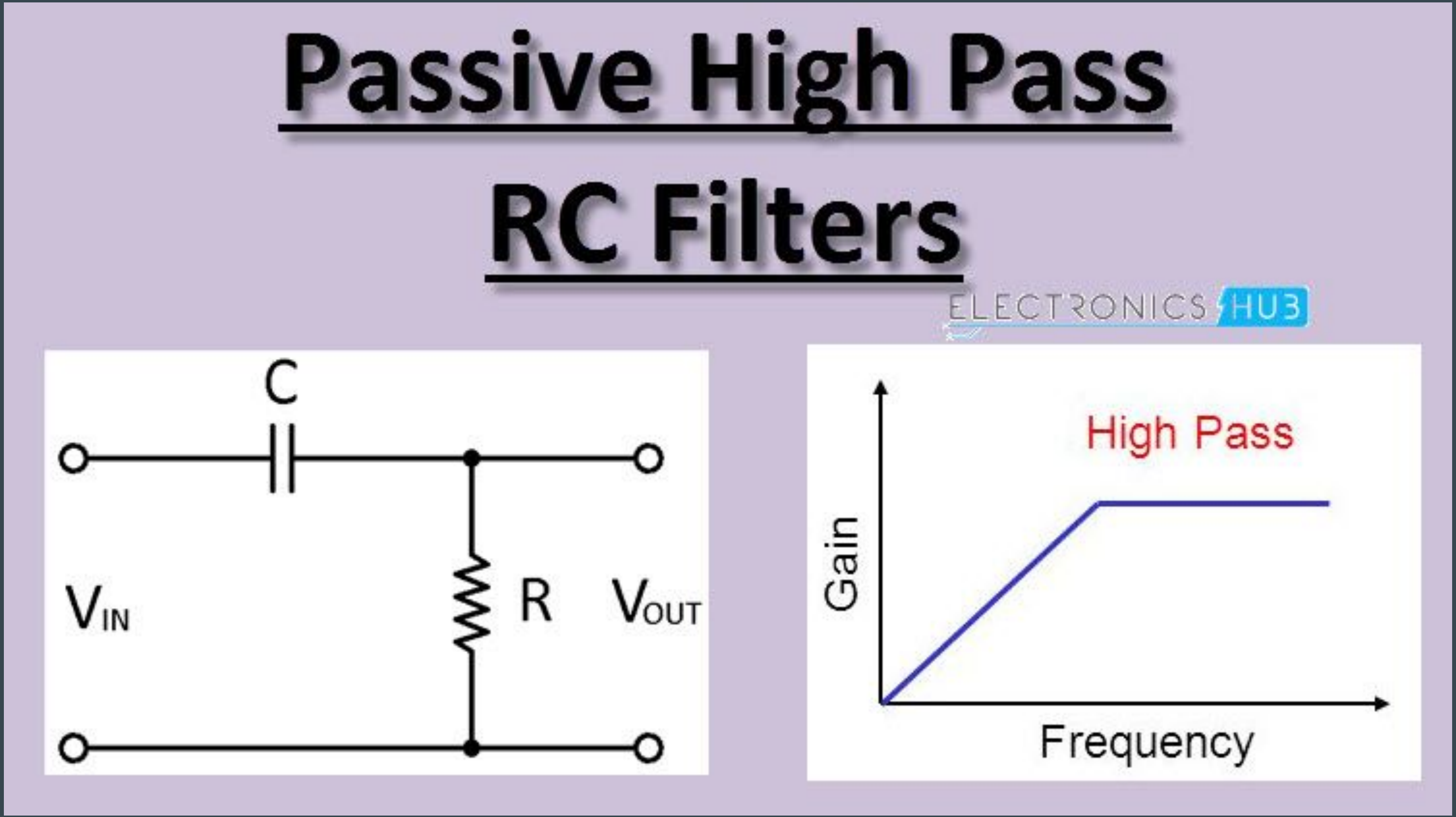
Signal generator: Maximum output 60Mhz

Oscilloscope: 100Mhz

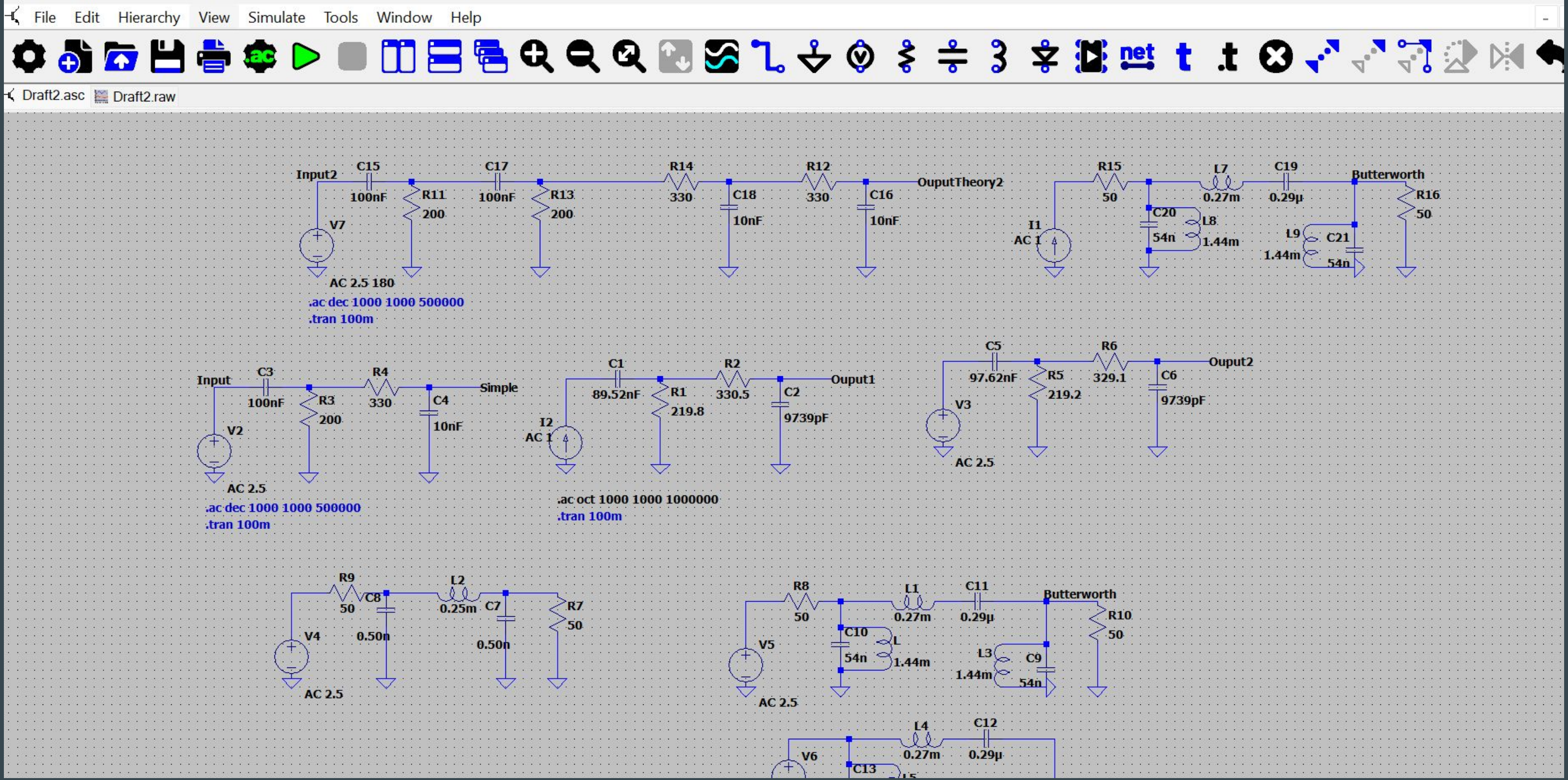
II. Filter design



II. Filter design

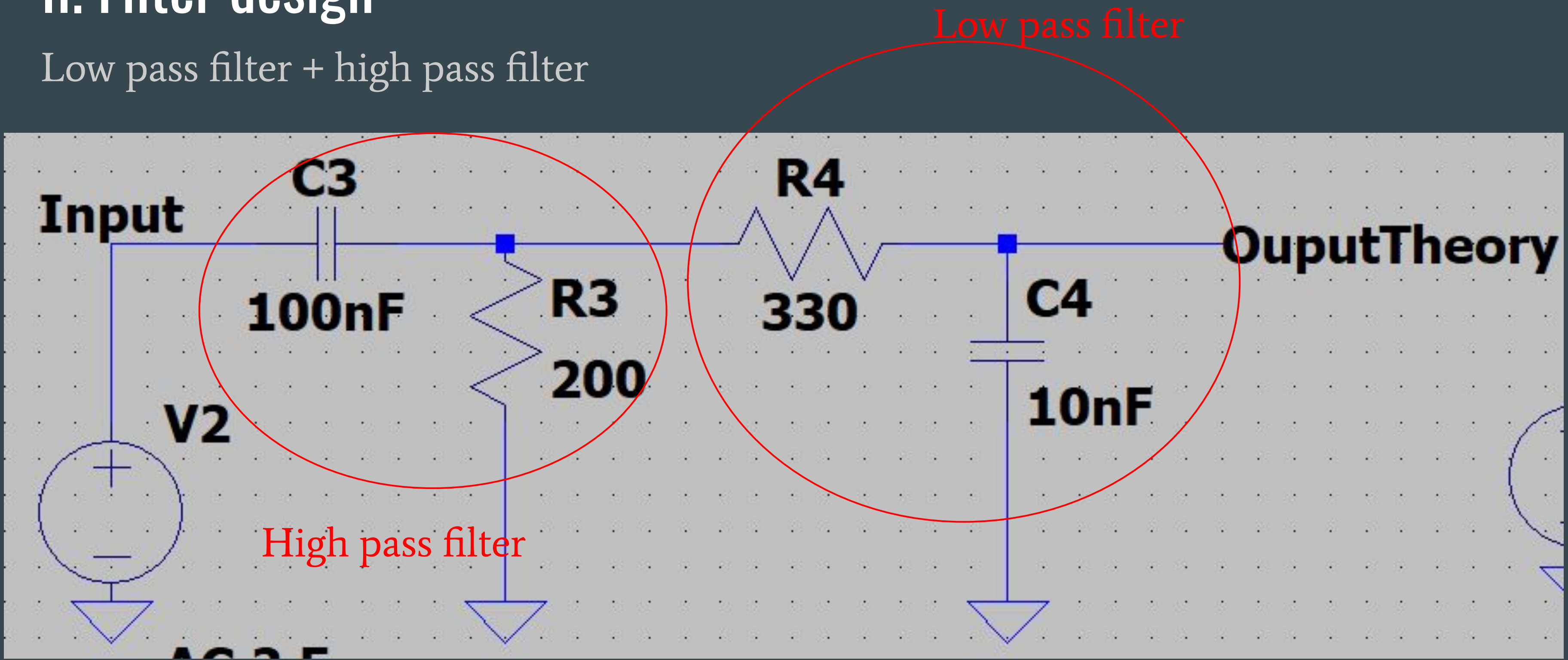


II. Filter design: Simulation using Ltspice



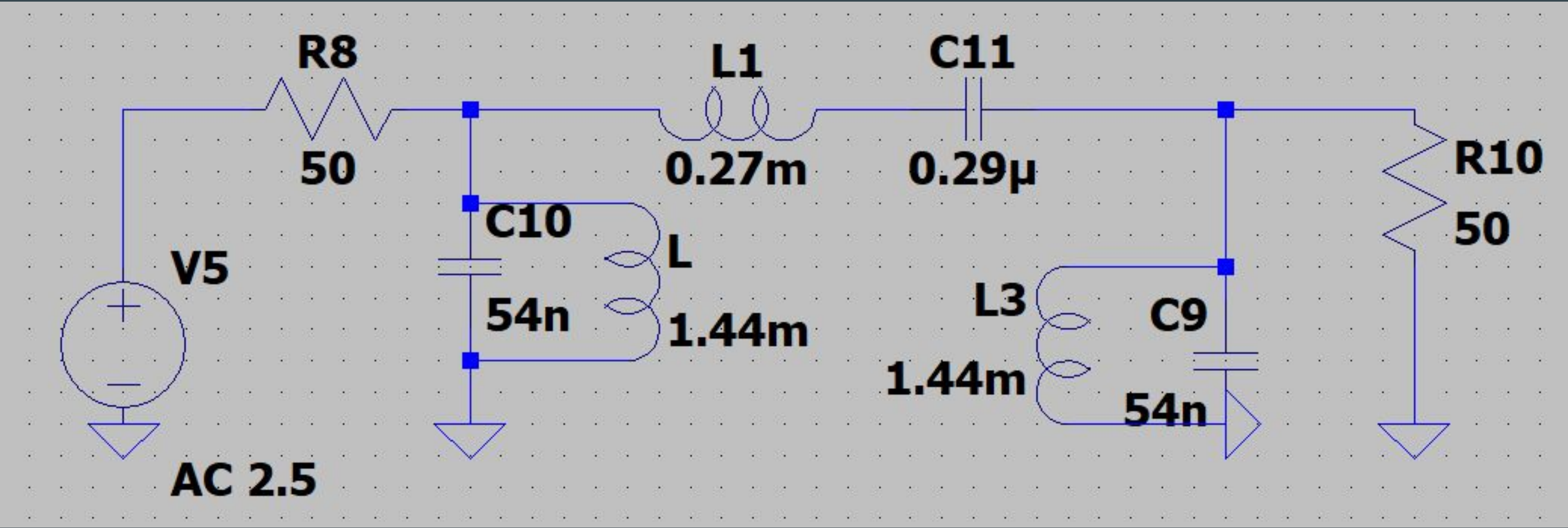
II. Filter design

Low pass filter + high pass filter

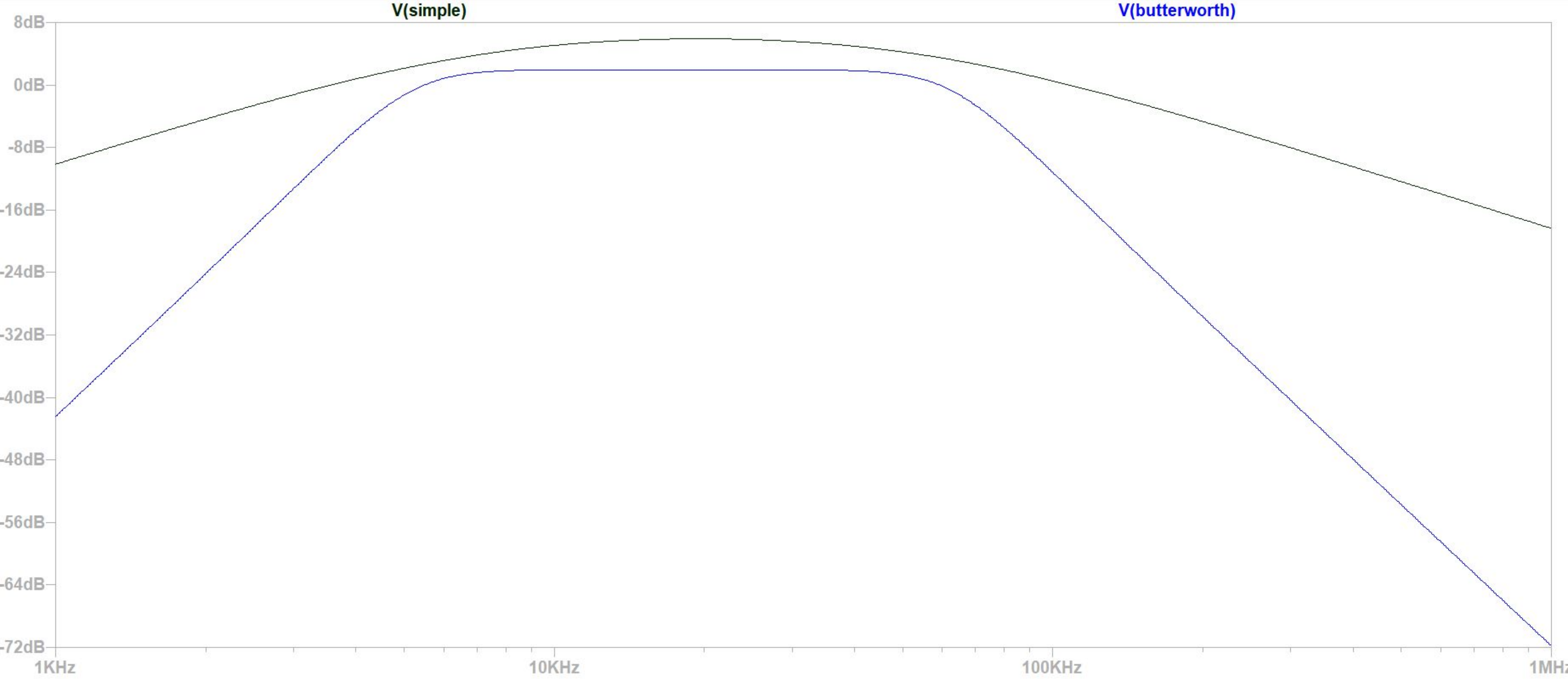


II. Filter design

Butterworth bandpass filter



II. Filter design



III. Data acquisition

1. Overview of instruments

- KKmoon FY6900 function generator (upper): provides various waveforms (sine, square, etc.) at different frequencies and amplitudes.
- Siglent SDS1104X-E oscilloscope (lower): used for observing the varying signal voltages, typically in the time domain.
- Others: PC,...

2. Softwares

- Python is the primary language for instrument control and data acquisition.
- Libraries used
  - PyVisa: communication with instruments via VISA.
  - NumPy: for mathematical operations, especially Fast Fourier Transform (FFT) for frequency domain analysis.
  - Matplotlib, Time,...

FY6900 Serial communication protocol FeelTech

Communication Protocol Summary

Writing Command	Command Line			Return	Reading Command	Command Line		Return	
	Code	Value	End Mark			Code	End Mark	Value	End Mark
Set waveform of main wave	WMW	xxxxxxxx	0x0a	0x0a	Read waveform of main wave	RMW	0x0a	xxxxxxxx	0x0a
Set frequency of main wave	WMF	xxxxxxxx	0x0a	0x0a	Read frequency of main wave	RMF	0x0a	xxxxxxxx	0x0a
Set amplitude of main wave	WMA	xxxxxxxx	0x0a	0x0a	Read amplitude of main wave	RMA	0x0a	xxxxxxxx	0x0a
Set offset of main wave	WMO	xxxxxxxx	0x0a	0x0a	Read offset of main wave	RMO	0x0a	xxxxxxxx	0x0a
Set duty cycle of main wave	WMD	xxxxxxxx	0x0a	0x0a	Read duty cycle of main wave	RMD	0x0a	xxxxxxxx	0x0a

2. Softwares

- Example: instruments initialising code

```
osc_inst = osc.connect('192.168.1.4')
fgen_inst = fgen.connect('192.168.1.2')

osc.initialize(osc_inst)
xscale = 'VARIABLE'
xoffset, yoffset = 0, 0
yscale = 1
amplification = 1
trigger_mode = 'AUTO'
osc.set_yscale(osc_inst, yscale)
inst_log = osc.create_log(xscale, xoffset, yscale, yoffset, amplification, trigger_mode) + '\n'

fgen.initialize(fgen_inst)
fgen.set_amplitude(fgen_inst, 1, 5)
channel = 1
wave_type = 'SINE'
freq = 'VARIABLE'
amp = 5
offset = 0
inst_log += fgen.create_log(channel, wave_type, freq, amp, offset) + '\n'
```

3. Measurement procedures: Instrument control

- First, ensure the instruments are properly connected and can be controlled through our software.
- Testing
  - Use the function generator to produce a signal of known frequency and amplitude.
  - Capture the signal with the oscilloscope.

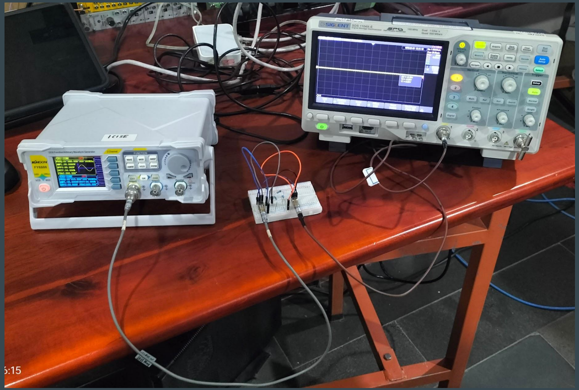
3. Measurement procedures: Frequency sweep measurement

- Purpose: to measure the transmission characteristics of the circuit across a range of frequencies.
- Steps:
  - Set up the initial parameters for the function generator and oscilloscope.
  - Automate the function generator to sweep through a range of frequencies.
  - Collect data at each frequency point using the oscilloscope.
  - Perform FFT on the captured data to obtain the frequency response.

```python
fgen.set_channel_state(fgen_inst, 1, 'ON')

out_dir = 'out/'
for index, frequency in enumerate(frequencies):
    set_freq(fgen_inst, osc_inst, frequency)
    tsample, time, voltage = osc.capture_data(osc_inst)
    np.save(out_dir + f'freqsweep_{index:03d}.npy', [time, voltage])

fgen.set_channel_state(fgen_inst, 1, 'OFF')
```

3. Measurement procedures: Filter transmission

- Purpose: to measure the transmission characteristics of the circuit across a range of frequencies with a bandpass filter.

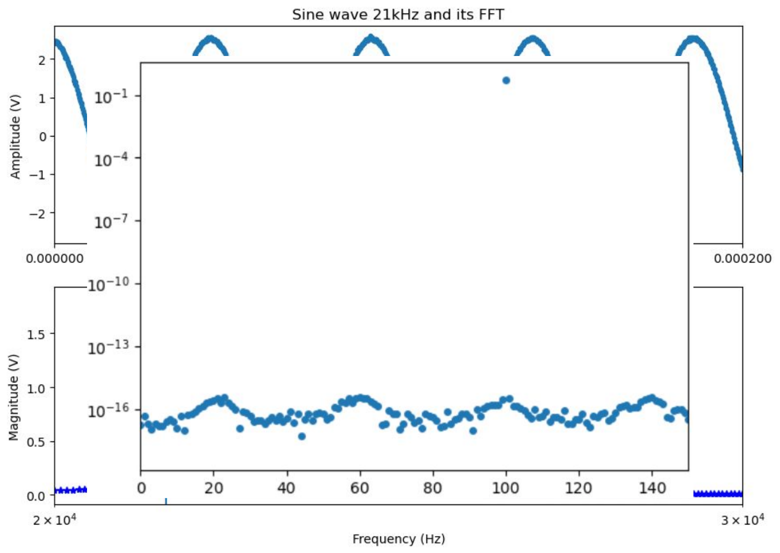


```python
def record(fil_name):
    osc_inst = osc.connect('192.168.1.4')
    fgen_inst = fgen.connect('192.168.1.2')

    out_directory = f'data/sweep_{fil_name}/'
    os.makedirs(out_directory, exist_ok = True)
    start_freq = 1e3
    end_freq = 200e3
    npoints = 200
    yscale = 1
    capture_freq_sweep(out_directory, osc_inst, fgen_inst, start_freq, end_freq, npoints, yscale)

    frequencies, voltage = import_freq_sweep(out_directory)
    transmission = 20*np.log10(voltage/2.5)

    out_ = np.array([frequencies,voltage,transmission])
    out_ =np.transpose(out_)

    np.savetxt(f'freq_volt_trans_{fil_name}.txt', X=out_,
    delimiter=' ', newline='\n', header='freq (Hz) volt (V) trans (dB)')

    fig, ax = plt.subplots(1,1, figsize = [12,6])
    ax.plot(frequencies, transmission, 'b-', marker = 'x', drawstyle = 'steps-mid', lw =2)
    ax.set(xscale = 'log')
    ax.set_xlabel('frequency (Hz)')
    ax.set_ylabel('transmission (dB)')
```

IV. Results

1. Methods

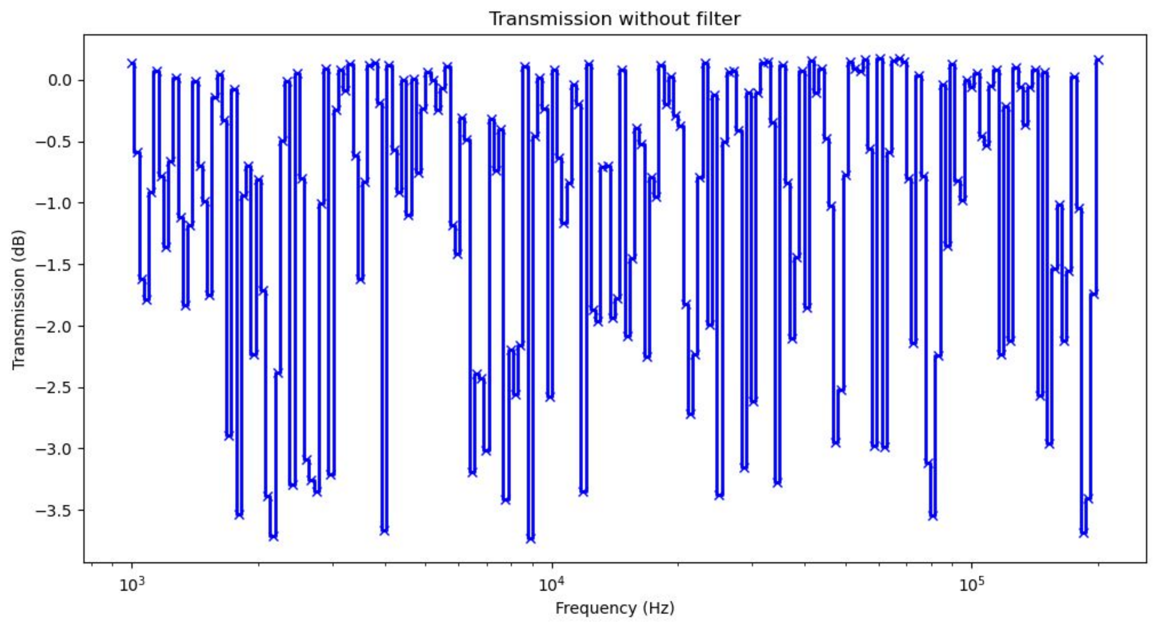
- Swept out the signal with frequency ranges from: 1 kHz to 200 kHz.
- Extract the frequency and amplitude of the signal (Using FFT).
    - Signal in frequency space is convoluted with a sinc function.
    - Sampling rate: 10 MHz -> above Nyquist
    - Cannot fully recover the voltage -> systematic errors



2. Results

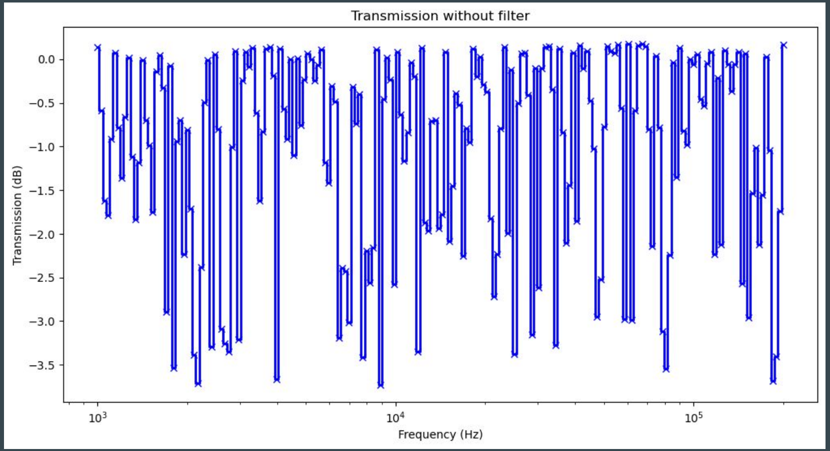
- Transmission:
- Difference responses between frequencies.
- Up to 30% loss (-3dB).
- The losses may due to the transmission lines, instruments or FFT.

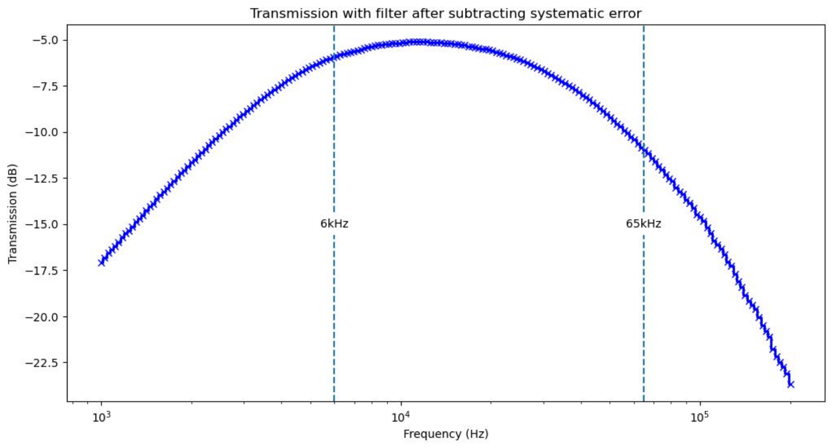
$$dB = 20 \times \log\left(\frac{V_{\text{out}}}{V_{\text{in}}}\right) \rightarrow dB = 20 \times \log\left(\frac{V_{\text{out}}}{2.5\ V}\right)$$

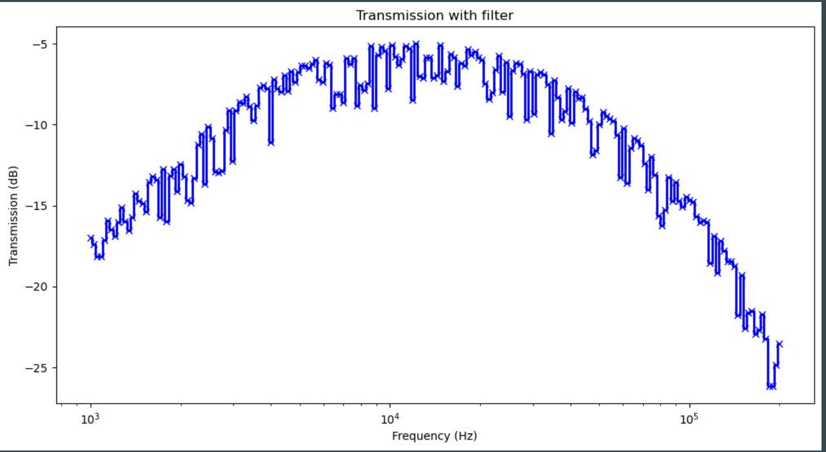


2. Results

- Performance of the band-pass filter from 6 kHz to 65 kHz (1 kHz to 200 kHz in total).
- Subtract the systematic errors from the baseline in order to get a smooth response.

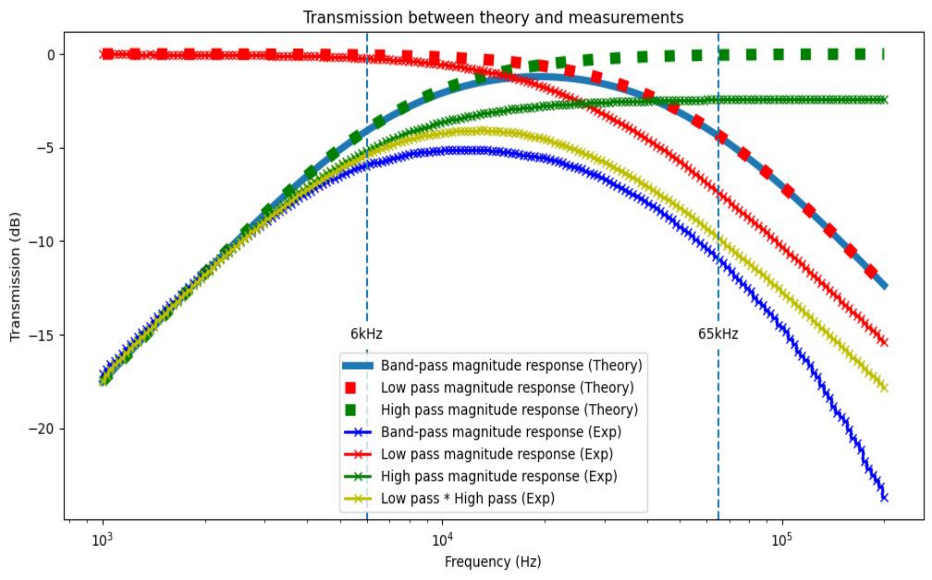






2. Results

- The mismatch between theory calculation and measurements.
- Start to split from 6 kHz.
- Huge transmission loss at high frequency domain.
- Impedance mismatch.
- Poor connection between instruments.



V. Conclusions

THANKS FOR YOUR LISTENING!