Experiment 01

Aditya Tripathy - ee24btech11001, Durgi Swaraj Sharma - ee24btech11018

Abstract
In Experiment-01, we produced Lissajous figures on an oscilloscope, and also captured a one time event on it.

1 Part 1 - Lissajous figures

1.1 Objective

To produce and verify Lissajous figures using an oscilloscope, a function generator, and Python.

1.2 Apparatus

- Oscilloscope
- Two channel function generator
- Connectimng wires and probes
- A computer to verify the results on

1.3 Theory

Lissajous figures are complex patterns formed when two sinusoidal signals are applied to the x and y axis as inputs to an oscilloscope. Their shapes are hence dependent on the many parameters that make each sinusoidal signal.

1.4 Procedure

1. Connections
   - Connect the two channels of the oscilloscope with the channels of the function generator, using 4 cables and the oscilloscope probes.
2. Device Setup
   - Turn the signals ON on the function generator, and set up sinusoidal outputs for both the channels.
   - Ensure that the oscilloscope reads these signals, use the auto-scale feature if necessary.
   - To change the plotting from $Y-T$ to $X-Y$, press the $Main/Delayed$ button, and in the menu, change the Time Base from $Y-T$ to $X-Y$.
   - Align Phase on the function generator to remove any inaccuracies.

3. Modify the various parameters on the function generator, Align Phase each time, and capture the Lissajous figures on the oscilloscope screen.

The captured Lissajous figures are shown in the next segment.

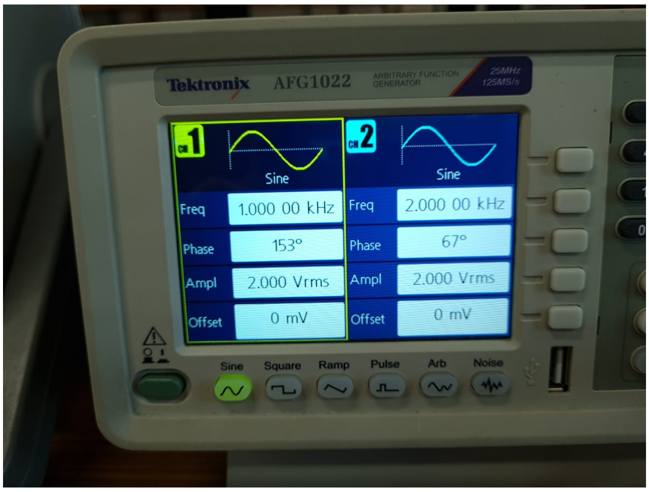
1.5 Justification

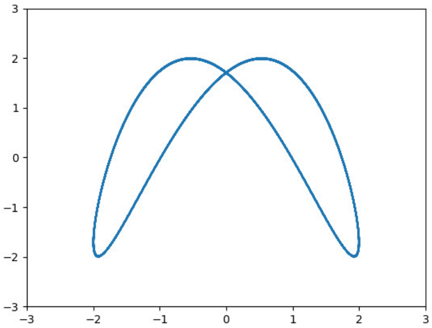
We will be using Python and the Matplotlib and Numpy libraries to verify our experiment. The following code plots out the theoretical Lissajous figures depending on the values of Voltage amplitudes, frequencies, and phase of the two channnels.

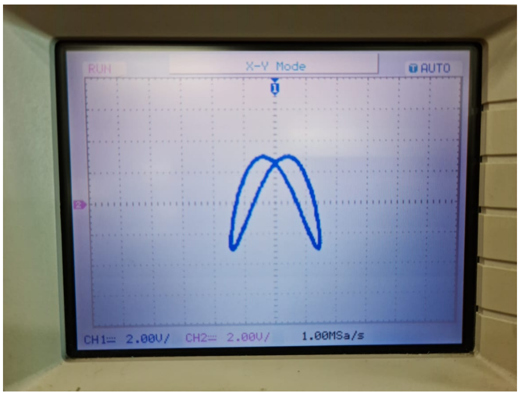
```
import numpy as np
import matplotlib.pyplot as plt
t = np.linspace(-3, 3, 1000)
A1 = {amplitude of signal 1}
f1 = {frequency of signal 1}
phi1 = {phase of signal 1 in radians}
 x = A1 * np.sin(2 * np.pi * f1 * t + phi1 )
A2 = {amplitude of signal 2}
f2 = {frequency of signal 2}
phi2 = {phase of signal 2 in radians}
y = A2 * np.sin(2 * np.pi * f2 * t + phi2 )
plt.plot(x, y)
plt.savefig("plot1.png")
plt.show()
```

By following the procediure for different values of the voltage parameters, we produced the following six Lissajous figures and verified them using the previously mentioned code. The parameters for the signals used are present in the pictures of the function generator menu.

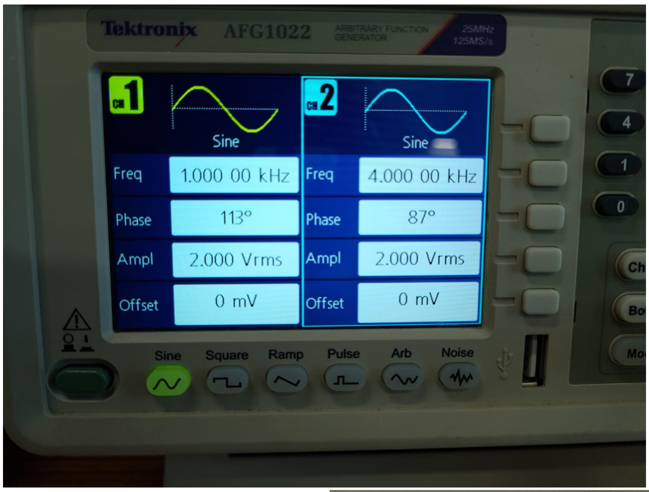
1.5.1 1.

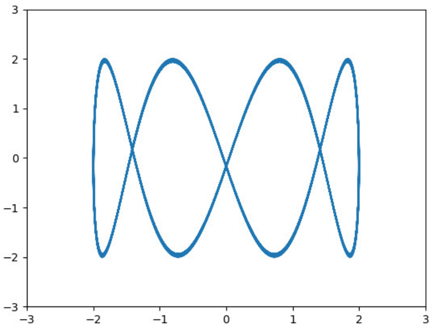


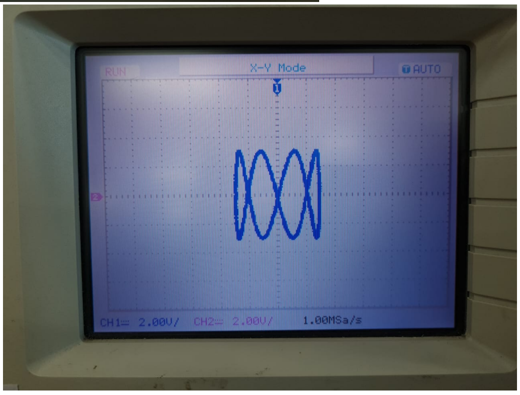

3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


X-Y Mode
AUTO
CH1= 2.00V/
CH2= 2.00V/
1.00MSa/s

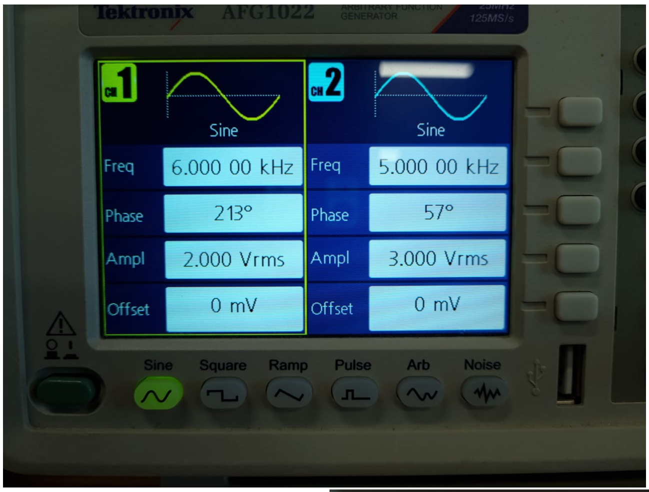
1.5.2 2.

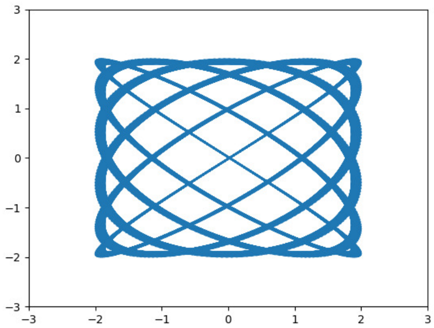

Tektronix
AFG1022
Sine
Freq
1.000 00 kHz
Phase
113°
Ampl
2.000 Vrms
Offset
0 mV
Sine
Freq
4.000 00 kHz
Phase
87°
Ampl
2.000 Vrms
Offset
0 mV
Sine
Square
Ramp
Pulse
Arb
Noise

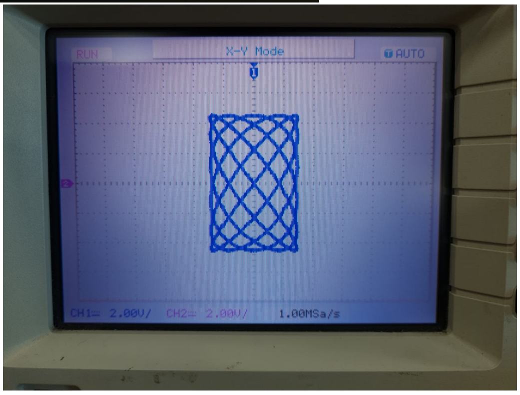

3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


X-Y Mode
AUTO
CH1= 2.00V/
CH2= 2.00V/
1.00MSa/s

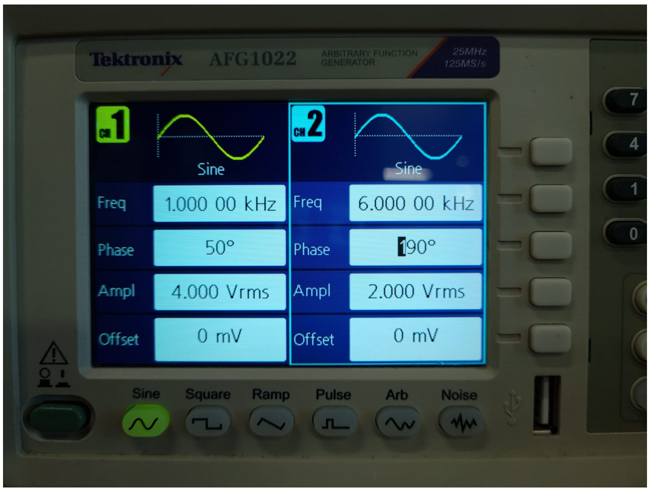
1.5.3 3.

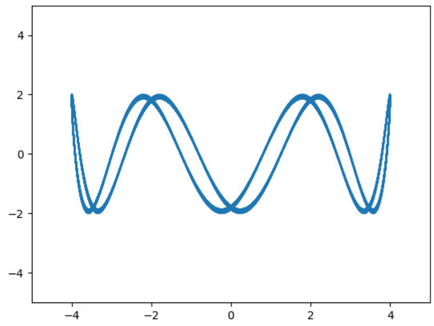

Tektronix AFG1022
Sine
Freq 6.000 00 kHz
Phase 213°
Ampl 2.000 Vrms
Offset 0 mV
Sine
Freq 5.000 00 kHz
Phase 57°
Ampl 3.000 Vrms
Offset 0 mV
Sine Square Ramp Pulse Arb Noise

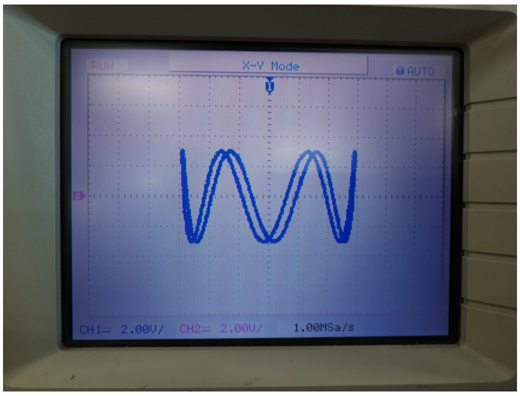

3
2
1
0
−1
−2
−3
−3 −2 −1 0 1 2 3


X-Y Mode
AUTO
CH1= 2.00V/ CH2= 2.00V/ 1.00MSa/s

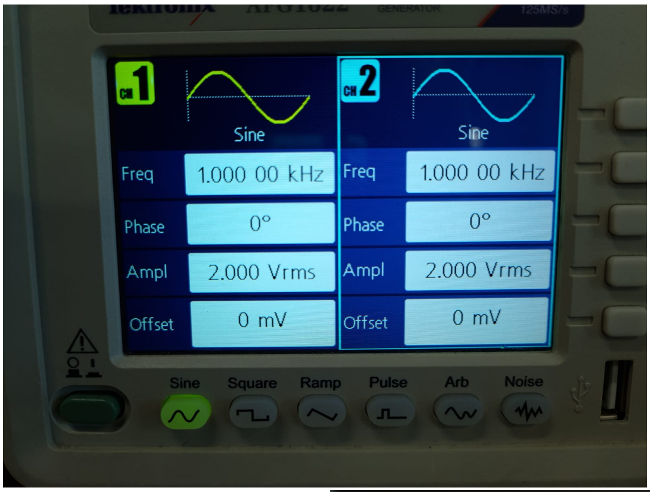
1.5.4 4.

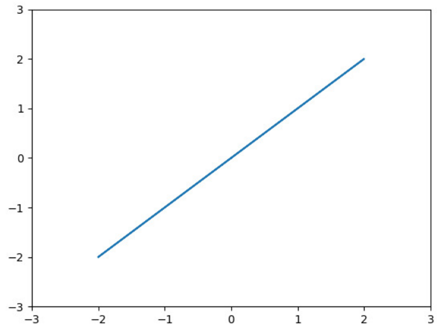

Tektronix AFG1022
Sine
Freq 1.000 00 kHz
Phase 50°
Ampl 4.000 Vrms
Offset 0 mV
Sine
Freq 6.000 00 kHz
Phase 190°
Ampl 2.000 Vrms
Offset 0 mV
Sine Square Ramp Pulse Arb Noise

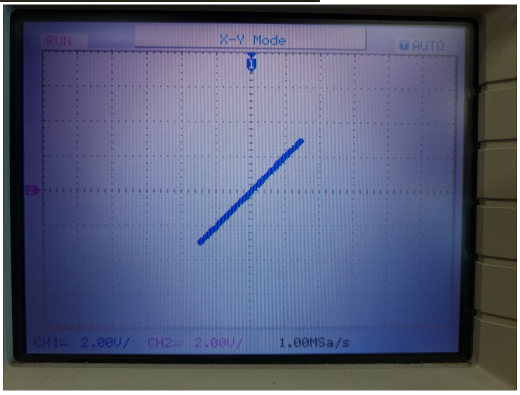




1.5.5 5.

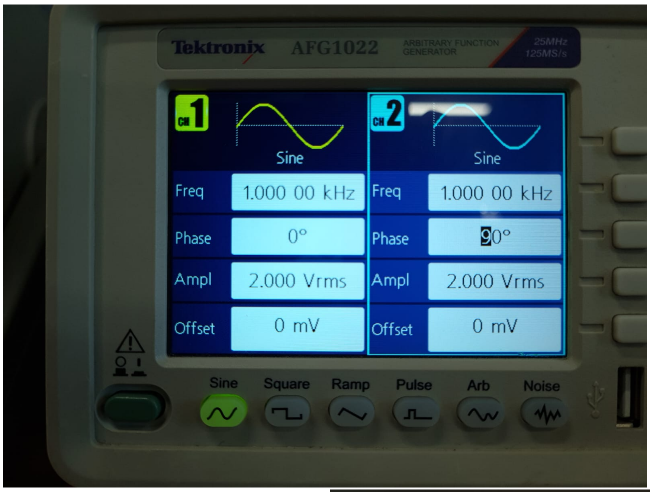


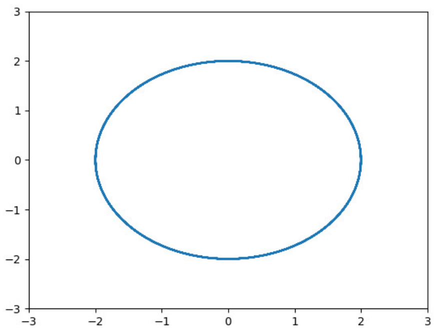


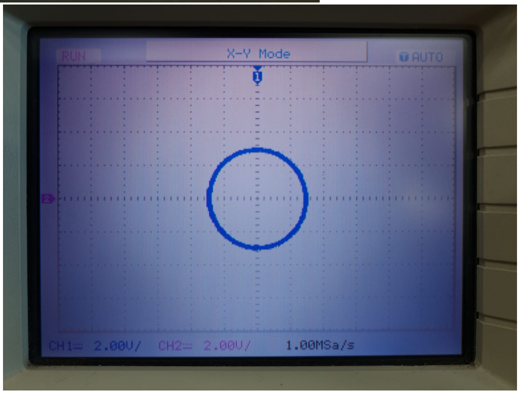


When all parameters for both the signals are equal, in the Lissajous figures essentially $x = y$, resulting in a straight line.

1.5.6 6.







When the phase difference is $\frac{\pi}{2}$ with all other parameters equal, we end up with a circle because a circle can be paramerterized as $x = a\cos\theta, y = a\sin\theta$, which is how are signals are.

1.6 Conclusion

We generated simple sinusoidal signals and made Lissajous figures with them, observing patterns emerging from the frequency ratios, symmetry, phase differences; and verified them using code.

2 Capturing a one-time event on the CRO

2.1 Objective

To capture a one-time event on an oscilloscope, by producing it on a function generator.

2.2 Apparatus

- Oscilloscope
- Function Generator
- Connecting probes and wires

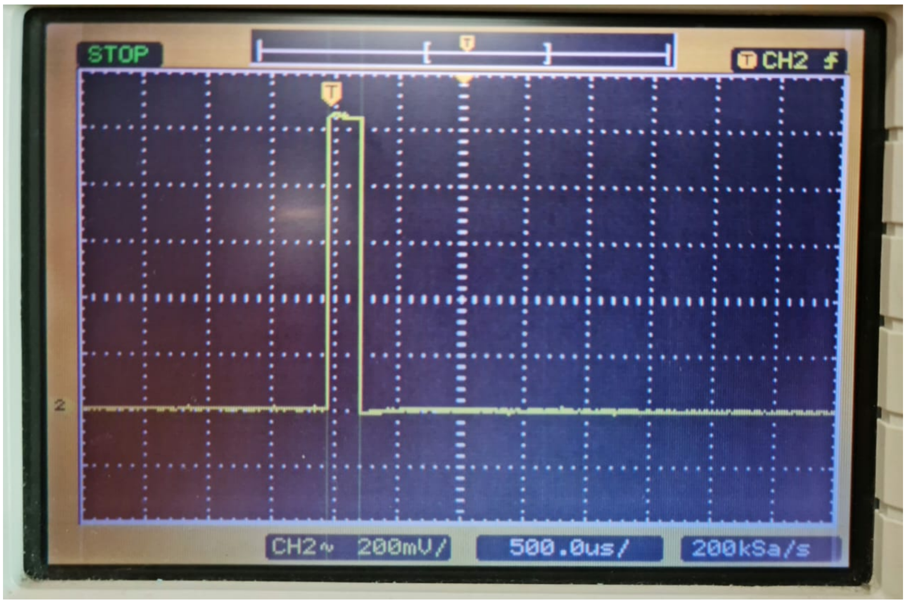
2.3 Theory

An oscilloscope's triggering capability lets us capture a one-time event. Triggering ensures the display starts when the event occurs.
Burst mode on the Fuction Generator helps us generate the one-time event.

2.4 Procedure

1. Set up the oscilloscope and the function generator by connecting the wire and the probe.
2. Set up the function generator by going to Burst mode, and choosing an appropriate one-time event with a manual trigger.
3. On the oscilloscope's Mode Coupling settings, ensure that the right channel is being monitored, and the oscilloscope looks for an increasing Edge. Ensure that the Sweep is Normal and the Coupling is DC.
4. Set an appropriate Trigger level on the oscilloscope.
5. Select Single mode, the oscilloscope will now start waiting for the one-time event.
6. Trigger the event on the function generator. Observe on the oscilloscope how we have captured the signal.

2.5 Demonstration's results



The one-time event we have captured.

2.6 Conclusion

We successfully captured a one-time event. The ability to capture a one-time event on an oscilloscope is of immense value, with numerous applications.