TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PURWANCHAL CAMPUS

DHARAN – 8

A lab report on DSP

Submitted By:

Name: Ram Chandra Ghimire

Faculty: BCT

Roll no: PUR078BCT062

Group: C

Lab Report: Lab 1 — Basic Signal Generation and Plotting using Python

Objective:

To implement and visualize basic discrete-time signals using Python and the `thinkdsp` module, and to understand their mathematical and practical significance.

1. Introduction (Theory)

In digital signal processing (DSP), signals are functions that convey information and are often represented as discrete-time sequences in computers. Basic signal types such as the unit impulse, unit step, ramp, exponential, sinusoid, and complex exponential form the foundation for understanding more complex signal behaviors and systems.

This lab uses Python and the `thinkdsp` module (from *Think DSP* by Allen B. Downey) to generate and visualize these elementary signals.

2. Tools and Libraries Used

- **Python** (v3.6+ recommended)
- **NumPy** for numerical operations
- **Matplotlib** for plotting signals
- **thinkdsp.py** module for DSP signal manipulation

3. Procedure

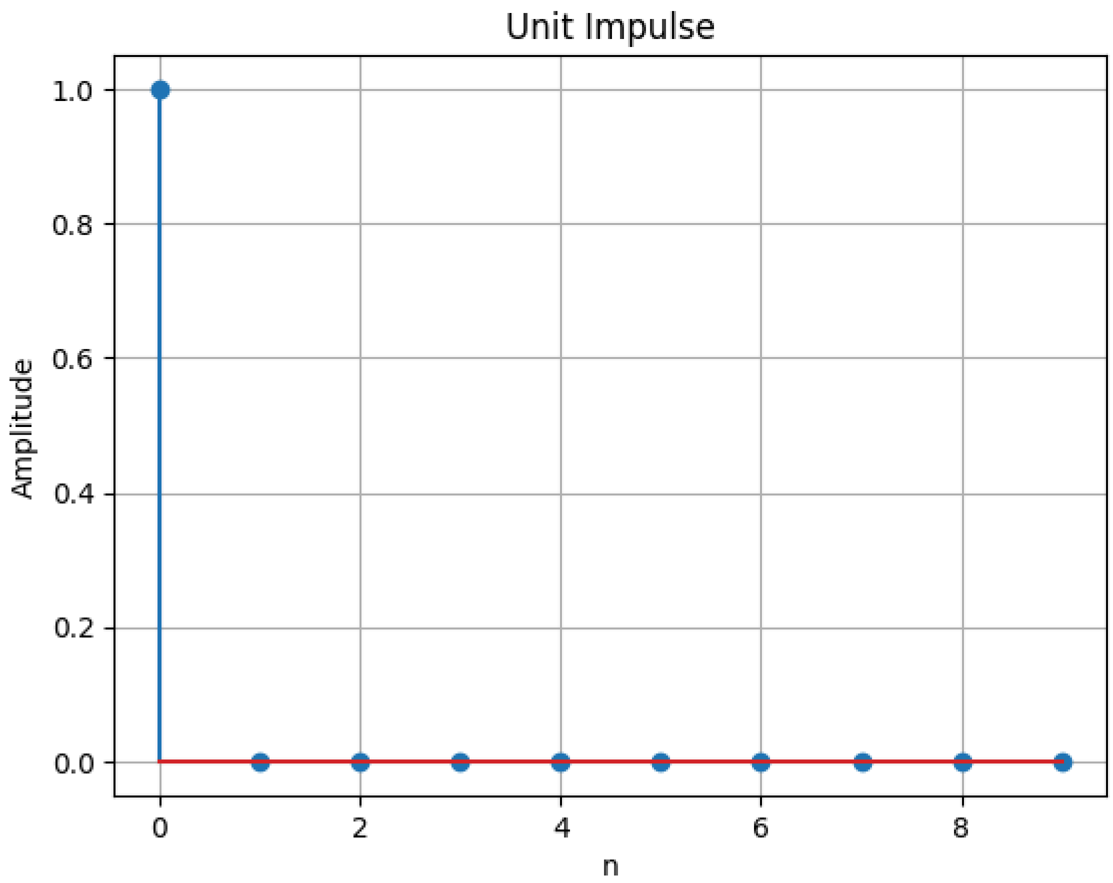
Step 1: Setup Python Environment

- Install Python
- Install necessary libraries.
    - `pip install numpy matplotlib`
    - Download `thinkdsp.py` from https://github.com/AllenDowney/ThinkDSP.

Step 2: Implement Signals in Python

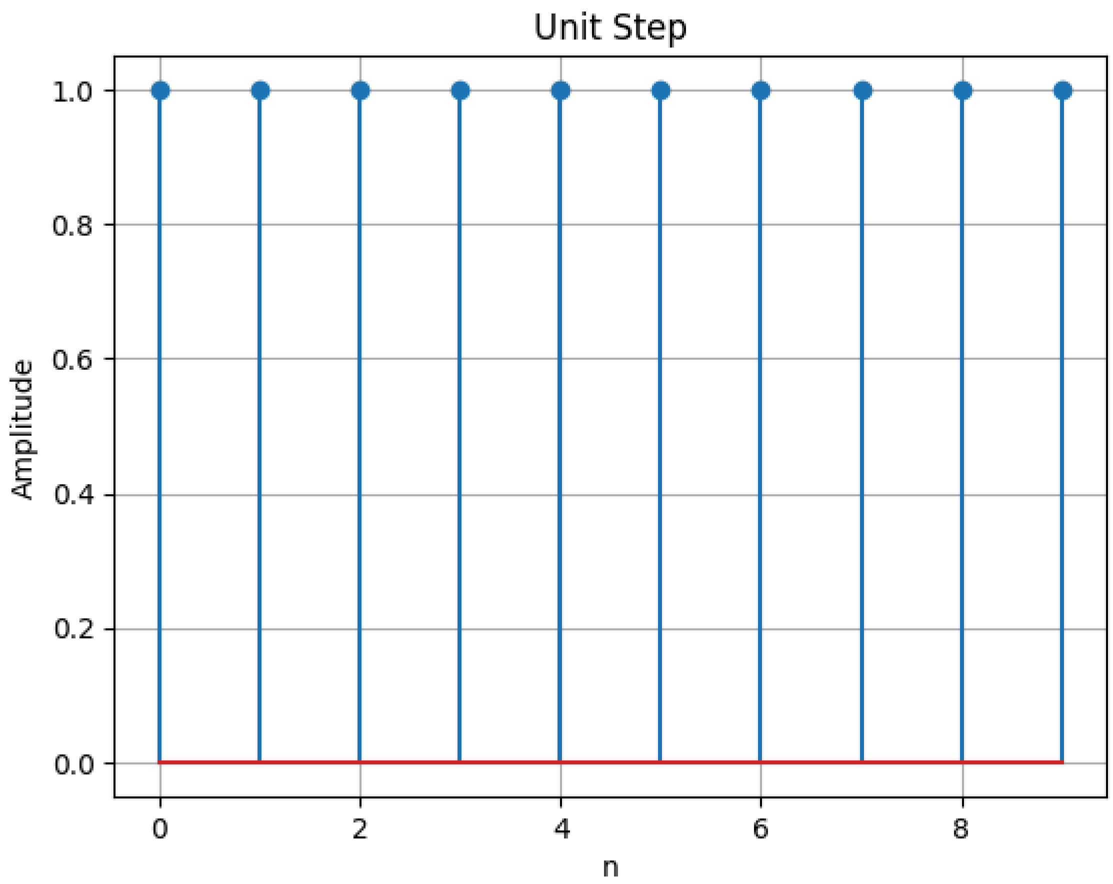
a) Unit Impulse Signal

```
impulse = np.zeros(10)
impulse[0] = 1
plt.stem(impulse)
plt.title("Unit Impulse Signal")
plt.show()
```



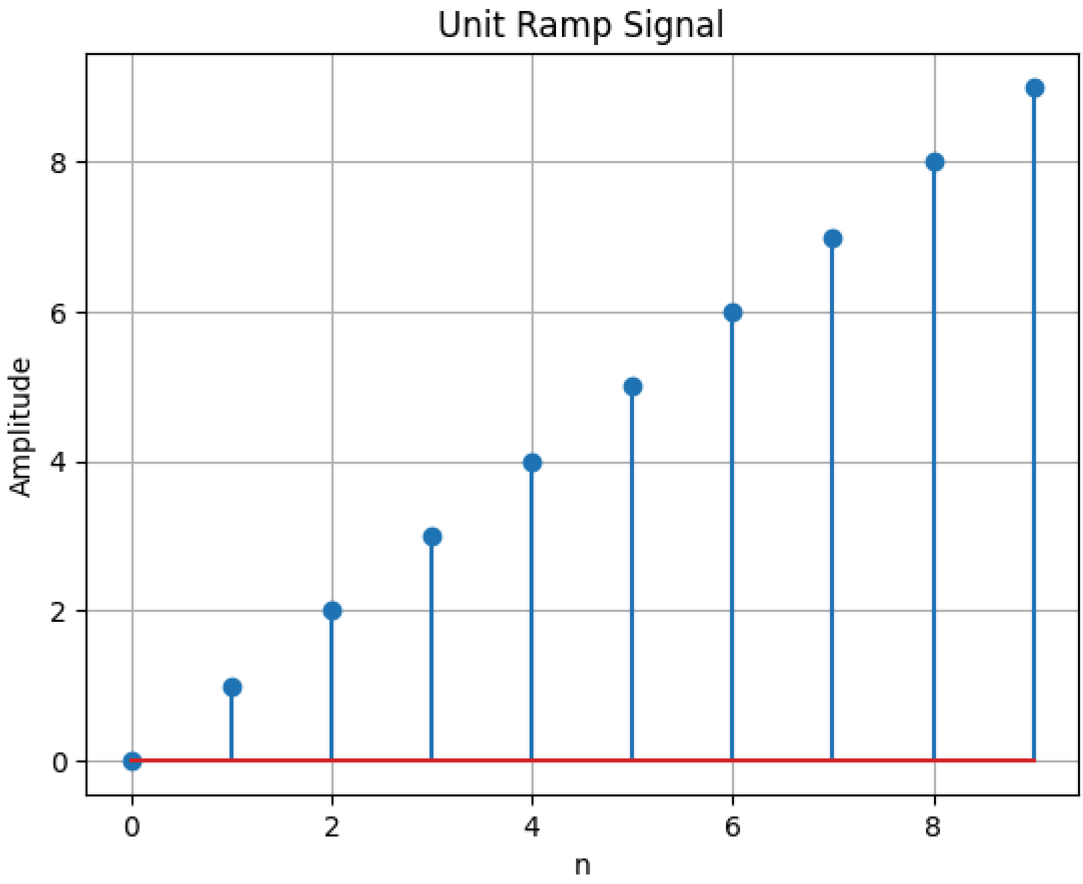
b) Unit Step Signal

```
step = np.ones(10)
plt.stem(step)
plt.title("Unit Step Signal")
plt.show()
```



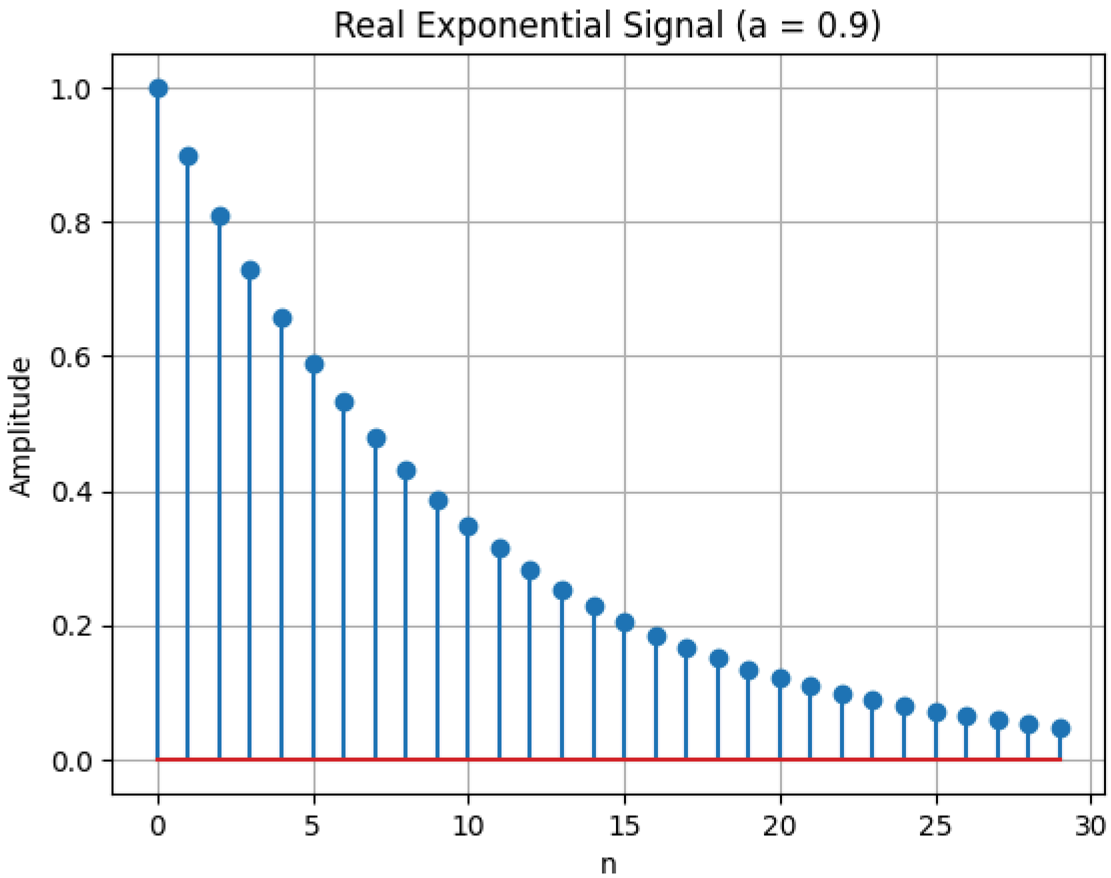
c) Unit Ramp Signal

```
n = np.arange(10)
ramp = n
plt.stem(n, ramp)
plt.title("Unit Ramp Signal")
plt.show()
```



d) Real Exponential Signal

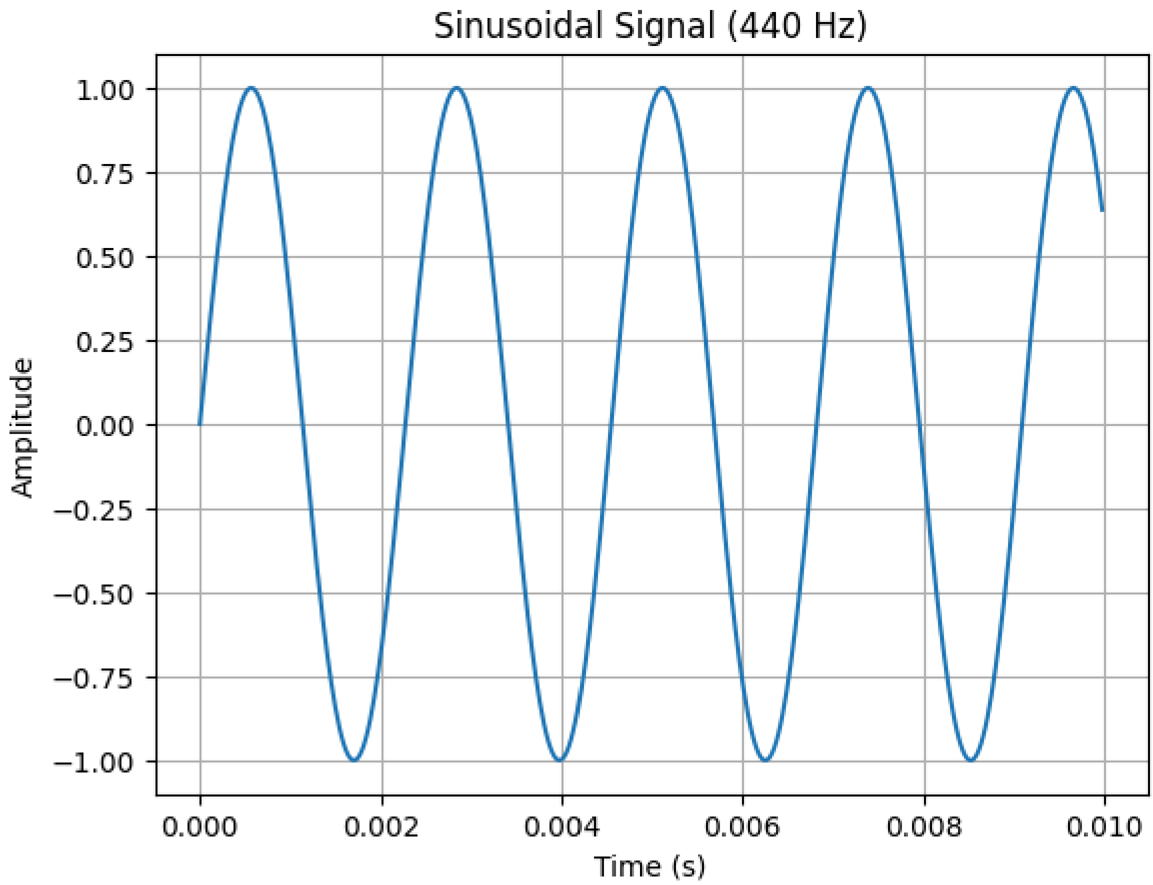
```
n = np.arange(30)
a = 0.9
real_exp = a ** n
plt.stem(n, real_exp)
plt.title("Real Exponential Signal (a=0.9)")
plt.show()
```



e) Sinusoidal Signal using `thinkdsp`

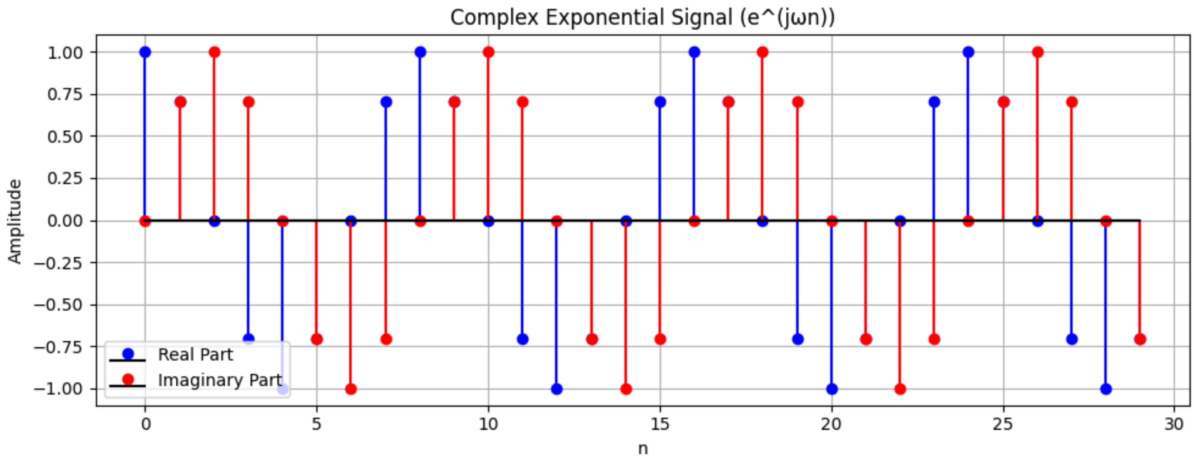
```
import thinkdsp

sin_sig = thinkdsp.SinSignal(freq=440, amp=1.0)
wave = sin_sig.make_wave(duration=0.01, framerate=44100)
wave.plot()
plt.title("Sinusoidal Signal (440 Hz)")
plt.show()
```



f) Complex Exponential Signal

```
n = np.arange(30)
omega = np.pi / 4
complex_exp = np.exp(1j * omega * n)
plt.stem(n, complex_exp.real, label='Real')
plt.stem(n, complex_exp.imag, label='Imag')
plt.title("Complex Exponential Signal")
plt.legend()
plt.show()
```



4. Observations

Signal Type	Key Features
Unit Impulse	Non-zero only at `n=0`, used for system testing
Unit Step	Constant value from `n=0` onward
Unit Ramp	Linearly increasing sequence
Real Exponential	Shows decay or growth based on base `a`
Sinusoidal	Periodic oscillation, characterized by frequency & amplitude
Complex Exponential	Rotating phasor, represents basis of DFT

---

5. Conclusion

This lab introduced foundational signal types using Python and helped visualize their discrete behavior. These signals are instrumental in analyzing linear systems and understanding frequency-domain behavior in future labs.
