



Subject: Programming With Python (01CT1309)

Aim: Analysis of Discrete-Time Signals Using Z-Transform

Experiment No: 17

Date:

Enrollment No:92400133037

Aim: Analysis of Discrete-Time Signals Using Z-Transform

IDE:

Install Library

pip install sympy

Example 1:

```
import sympy as sp
# Define symbols
n, z, a = sp.symbols('n z a')
# Define the signal x[n] = a^n * u[n]
x_n = a**n
# Compute the Z-transform
X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))
# Print the result
print("Z-transform of x[n] = a^n u[n]:")
sp.pprint(X_z, use_unicode=True)
```

```
lab17 > example1.py > ...
 1  import sympy as sp
 2  n,z,a=sp.symbols('n z a')
 3  x_n=a**n
 4  X_z=sp.summation(x_n*z**(-n),(n,0,sp.oo))
 5  print("Z-transform of x[n] = a^n u[n]:")
 6  sp.pprint(X_z, use_unicode=True)
 7

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
TERMINAL
```

PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\example1.py"
• Z-transform of x[n] = a^n u[n]:
$$\frac{1}{z - a}$$
 for $|z| > |a|$

$$\begin{cases} \infty & n = 0 \\ a^{-n} & n < 0 \\ a \cdot z^{-n} & \text{otherwise} \end{cases}$$



Subject: Programming With Python (01CT1309)

Aim: Analysis of Discrete-Time Signals Using Z-Transform

Experiment No: 17

Date:

Enrollment No:92400133037

Example 2:

```
# Define symbols
n, z, a = sp.symbols('n z a')
# Define the signal x[n] = a^n * u[n]
x_n = 2**n
# Compute the Z-transform
X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))
# Print the result
print("Z-transform of x[n] = a^n u[n]:")
sp.pprint(X_z, use_unicode=True)
```

The screenshot shows a Jupyter Notebook interface. The code cell contains the provided Python script. Below it, the terminal output shows the result of running the script, which is the Z-transform of the signal $x[n] = 2^n u[n]$. The output is displayed in a mathematical form using LaTeX-like syntax.

```
lab17 > example2.py > ...
1 import sympy as sp
2 n,z,a=sp.symbols('n z a')
3 x_n=2**n
4 X_z=sp.summation([x_n * z**(-n), (n, 0, sp.oo)])
5 print("Z-transform of x[n] = a^n u[n]:")
6 sp.pprint(X_z, use_unicode=True)
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

TERMINAL

- PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\example2.py"
Z-transform of x[n] = a^n u[n]:
$$\begin{cases} \frac{1}{z-2} & \text{for } |z| < 1/2 \\ \infty & \text{otherwise} \end{cases}$$

Example 3:

```
import sympy as sp
# Define symbols
n, z = sp.symbols('n z')
# Define the unit step signal u[n]
u_n = 1
```



Subject: Programming With Python (01CT1309)

Aim: Analysis of Discrete-Time Signals Using Z-Transform

Experiment No: 17

Date:

Enrollment No:92400133037

```
# Compute the Z-transform
U_z = sp.summation(u_n * z**(-n), (n, 0, sp.oo))

# Print the result
print("Z-transform of the unit step signal u[n]:")
sp.pprint(U_z, use_unicode=True)
```

```
lab17 > example3.py > ...
1 import sympy as sp
2 n,z=sp.symbols('n z')
3 u_n=1
4 U_z=sp.summation(u_n * z**(-n),(n,0,sp.oo))
5 print("Z-transform of the unit step signal u[n]:")
6 sp.pprint(U_z, use_unicode=True)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
TERMINAL
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\example3.py"
Z-transform of the unit step signal u[n]:

$$\frac{1}{1 - \frac{1}{z}}$$
 for  $|z| < 1$ 

$$1 - \frac{1}{z}$$


$$\left\{ \begin{array}{ll} \infty & \\ \sum_{n=0}^{\infty} z^{-n} & \text{otherwise} \end{array} \right.$$

```

Example 4:

```
import sympy as sp
# Define symbols
n, z, alpha = sp.symbols('n z alpha')
# Define the signal x[n] = exp(alpha * n) * u[n]
x_n = sp.exp(alpha * n)
# Compute the Z-transform
X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))
# Print the result
print("Z-transform of x[n] = exp(alpha * n) u[n]:")
sp.pprint(X_z, use_unicode=True)
```



Subject: Programming With Python (01CT1309)

Aim: Analysis of Discrete-Time Signals Using Z-Transform

Experiment No: 17

Date:

Enrollment No:92400133037

The screenshot shows a Jupyter Notebook interface with a code cell containing Python code to calculate the Z-transform of a sequence. The code uses the SymPy library to perform symbolic summation. The output cell shows the command run in the terminal and the resulting mathematical expression for the Z-transform.

```
lab17 > ℗ example4.py > ...
1 import sympy as sp
2 n,z,alpha=sp.symbols('n z alpha')
3 x_n=sp.exp(alpha*n)
4 X_z=sp.summation(x_n*z**(-n),(n,0,sp.oo))
5 print("Z-transform of x[n] = exp(alpha * n) u[n]:")
6 sp.pprint(X_z, use_unicode=True)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
▼ TERMINAL

● PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\example4.py"
Z-transform of x[n] = exp(alpha * n) u[n]:

$$\sum_{n=0}^{\infty} x_n z^{-n} = \frac{1}{z - e^{\alpha}}$$

```

Example 5:

```
import sympy as sp
# Define symbols
n, z = sp.symbols('n z')
# Define the finite sequence x[n] = {1, 2, 3}
x_n = [1, 2, 3]
# Compute the Z-transform manually
X_z = sum(x_n[i] * z**(-i) for i in range(len(x_n)))
# Print the result
print("Z-transform of the finite sequence {1, 2, 3}:")
sp.pprint(X_z, use_unicode=True)
```



Subject: Programming With Python (01CT1309)

Aim: Analysis of Discrete-Time Signals Using Z-Transform

Experiment No: 17

Date:

Enrollment No:92400133037

```
lab17 > ℗ example5.py > ...
1  import sympy as sp
2  n,z=sp.symbols('n z')
3  x_n=[1, 2, 3]
4  X_z=sum([x_n[i]*z**(-i)for i in range(len(x_n))])
5  print("Z-transform of the finite sequence {1, 2, 3}:")
6  sp.pprint(X_z, use_unicode=True)
7

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
▼ TERMINAL

● PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\example5.py"
Z-transform of the finite sequence {1, 2, 3}:

$$\frac{2}{z^2} + \frac{1}{z}$$

```

Example 6

```
import sympy as sp
# Define symbols
n, z, omega = sp.symbols('n z omega')
# Define the sinusoidal sequence x[n] = sin(omega * n) * u[n]
x_n = sp.sin(omega * n)
# Compute the Z-transform
X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))
# Print the result
print("Z-transform of x[n] = sin(omega * n) u[n]:")
sp.pprint(X_z, use_unicode=True)
```

```
lab17 > ℗ example6.py > ...
1  import sympy as sp|
2  n,z,omega=sp.symbols('n z omega')
3  x_n=sp.sin(omega*n)
4  X_z=sp.summation(x_n *z**(-n),(n,0,sp.oo))
5  print("Z-transform of x[n] = sin(omega * n) u[n]:")
6  sp.pprint(X_z, use_unicode=True)

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
▼ TERMINAL

● PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\example6.py"
Z-transform of x[n] = sin(omega * n) u[n]:

$$\sum_{n=0}^{\infty} z^{-n} \cdot \sin(n \cdot \omega)$$

```



Subject: Programming With Python (01CT1309)

Aim: Analysis of Discrete-Time Signals Using Z-Transform

Experiment No: 17

Date:

Enrollment No:92400133037

Post Lab Exercise:

- Using Python, compute the Z-transform of the sequence $x[n] = 3^n u[n]$.
- Using Python, compute the Z-transform of the sequence $x[n] = \cos(\omega n)u[n]$.

```
lab17 > postLab.py > ...
1 import sympy as sp
2 n,z=sp.symbols('n z')
3 x_n=3**n
4 X_z=sp.summation(x_n*z**(-n),(n,0,sp.oo))
5 print("Z-transform of x[n]=3^n u[n]:")
6 sp.pprint(X_z,use_unicode=True)
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
✓ TERMINAL
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\postLab.py"
• Z-transform of x[n]=3^n u[n]:

$$\begin{cases} \frac{1}{1 - \frac{3}{z}} & \text{for } |z| < 1/3 \\ \infty & \text{otherwise} \end{cases}$$

n = 0

8 #b
9 n,z,omega=sp.symbols('n z omega')
10 x_n=sp.cos(omega*n)
11 X_z=sp.summation(x_n*z**(-n),(n,0,sp.oo))
12 print("Z-transform of x[n] = \cos(\omega n) u[n]:")
13 sp.pprint(X_z,use_unicode=True)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
✓ TERMINAL
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\postLab.py"
Z-transform of x[n] = \cos(\omega n) u[n]:

$$\sum_{n=0}^{\infty} z^{-n} \cdot \cos(n \cdot \omega)$$

```

GITHUB LINK:

https://github.com/Heer972005/Python_Lab