
 <b>Marwadi University</b> Marwadi Chandarana Group 	<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Programming With Python (01CT1309)</b>	<b>Aim:</b> Analysis of Discrete-Time Signals Using Z-Transform	
<b>Experiment No: 17</b>	<b>Date:</b>	<b>Enrollment No: 92400133055</b>

### [GITHUB](#)

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

**IDE:** Visual Studio Code

Install Library `pip install sympy`

#Example 1: import sympy as sp

# Define symbols

`n, z, a = sp.symbols('n z a')`

# Define the signal  $x[n] = a^n \cdot u[n]$

`x_n = a**n`

# Compute the Z-transform

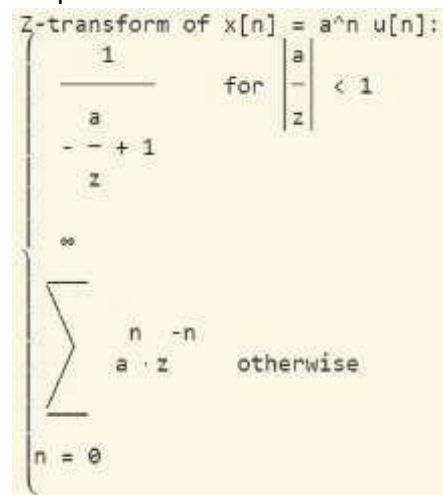
`X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))`

# Print the result

`print("Z-transform of  $x[n] = a^n u[n]$ :")`

`sp.pprint(X_z, use_unicode=True)`

Output:



$$\begin{aligned}
 &\text{Z-transform of } x[n] = a^n u[n]: \\
 &\sum_{n=0}^{\infty} a^n z^{-n} \quad \text{for } \left| \frac{a}{z} \right| < 1 \\
 &= \frac{1}{1 - \frac{a}{z}}
 \end{aligned}$$

#Example 2:

# Define symbols `n, z, a =`


`sp.symbols('n z a')` # Define the

signal  $x[n] = a^n \cdot u[n]$  `x_n = 2**n`

# Compute the Z-transform

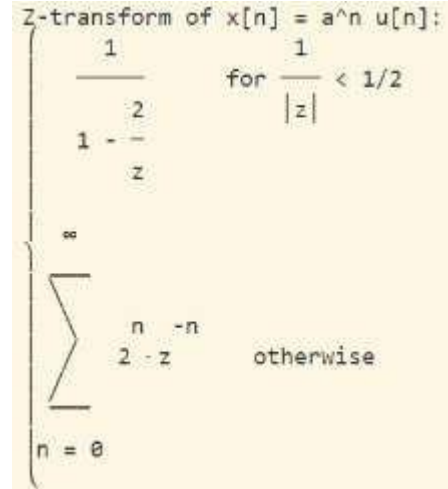
`X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))`

# Print the result

 <b>Marwadi University</b> Marwadi Chandarana Group	<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Programming With Python (01CT1309)</b>	<b>Aim:</b> Analysis of Discrete-Time Signals Using Z-Transform	
<b>Experiment No: 17</b>	<b>Date:</b>	<b>Enrollment No: 92400133055</b>

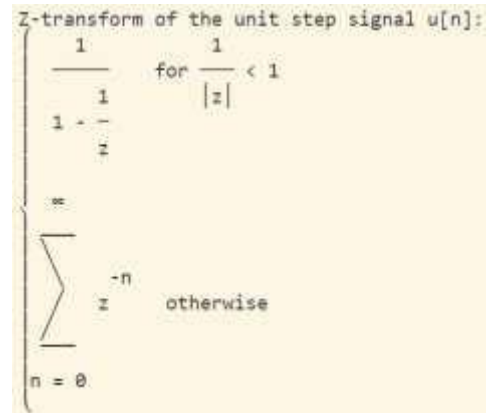
```
print("Z-transform of x[n] = a^n u[n]:")
```

```
sp.pprint(X_z, use_unicode=True) Output:
```





The image shows the output of a SymPy Z-transform calculation for  $x[n] = a^n u[n]$ . The output is displayed in a large, stylized font. It shows the Z-transform  $X(z) = \frac{1}{1 - az^{-1}}$  for  $|z| > |a|$ , and the summation  $\sum_{n=0}^{\infty} a^n z^{-n}$  otherwise.

```
#Example 3: import sympy as sp
# Define symbols n, z =
sp.symbols('n z') # Define the
unit step signal u[n] u_n = 1 #
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) Output:
```



The image shows the output of a SymPy Z-transform calculation for the unit step signal  $u[n]$ . The output is displayed in a large, stylized font. It shows the Z-transform  $U(z) = \frac{1}{1 - z^{-1}}$  for  $|z| > 1$ , and the summation  $\sum_{n=0}^{\infty} z^{-n}$  otherwise.

 <b>Marwadi University</b> Marwadi Chandarana Group 	<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Programming With Python (01CT1309)</b>	<b>Aim:</b> Analysis of Discrete-Time Signals Using Z-Transform	
<b>Experiment No: 17</b>	<b>Date:</b>	<b>Enrollment No: 92400133055</b>

```
#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) Output:
```

$$\text{Z-transform of } x[n] = \exp(\alpha n) u[n]:$$

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



$$n = 0$$

```
#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) Output:
```

$$\text{Z-transform of } x[n] = \exp(\alpha n) u[n]:$$

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

$$n = 0$$

 <b>Marwadi University</b> Marwadi Chandarana Group 	<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Programming With Python (01CT1309)</b>	<b>Aim:</b> Analysis of Discrete-Time Signals Using Z-Transform	
<b>Experiment No: 17</b>	<b>Date:</b>	<b>Enrollment No: 92400133055</b>

```
#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) Output:
```

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



### Post Lab Exercise:

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

Code:

```
import sympy as sp n, z, a =
sp.symbols('n z a') x_n =
a**n
X_z = sp.summation(x_n * z**(-n), (n, 0,
sp.oo)) print("Z-transform of x[n] = a^n
u[n]:") sp.pprint(X_z, use_unicode=True) X_z_3
= X_z.subs(a, 3) print("Z-transform of x[n] =
3^n u[n]:") sp.pprint(X_z_3, use_unicode=True)
```

Output:



 <b>Marwadi University</b> Marwadi Chandarana Group 	<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Programming With Python (01CT1309)</b>	<b>Aim:</b> Analysis of Discrete-Time Signals Using Z-Transform	
<b>Experiment No: 17</b>	<b>Date:</b>	<b>Enrollment No: 92400133055</b>

$$\begin{aligned} &\text{Z-transform of } x[n] = a^n u[n]: \\ &\left\{ \begin{array}{ll} \frac{1}{1 - \frac{a}{z}} & \text{for } \left| \frac{a}{z} \right| < 1 \\ \sum_{n=0}^{\infty} a^n z^{-n} & \text{otherwise} \end{array} \right. \\ &\text{Z-transform of } x[n] = 3^n u[n]: \\ &\left\{ \begin{array}{ll} \frac{1}{1 - \frac{3}{z}} & \text{for } \left| \frac{3}{z} \right| < 1/3 \\ \sum_{n=0}^{\infty} 3^n z^{-n} & \text{otherwise} \end{array} \right. \end{aligned}$$

- Using Python, compute the Z-transform of the sequence  $x[n] = \cos(wn)u[n]$ . Code:

```
import sympy as sp
import math
n, z, w = sp.symbols('n z w')
x_n = sp.cos(w*n)
X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))
print("Z-transform of x[n] = cos(wn)u[n]:")
sp.pprint(X_z, use_unicode=True)
```

Output:

 <b>Marwadi University</b> <small>Marwadi Chandarana Group</small> 	<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: Programming With Python (01CT1309)</b>	<b>Aim:</b> Analysis of Discrete-Time Signals Using Z-Transform	
<b>Experiment No: 17</b>	<b>Date:</b>	<b>Enrollment No: 92400133055</b>

Z-transform of  $x[n] = \cos(\omega n)u[n]$ :

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

$n = 0$