WIGGLES

EXPERIMENT #05

SEPTEMBER 2022

EXPERIMENT NAME

Implementing and verifying different properties of convolution namely, Commutative Property, Distributive Property and Associative Property.

1. COMMUTATIVE PROPERTY

Theory

1. The **commutative property** of convolution states that **the order** in which we convolve two signals does not change the result, **i.e.,**

Expression

$$x1(t) * x2(t) \ = \ x2(t) * x1(t)$$

Getting the environment ready

- **Python 3.10** is installed in the system and added to the system variables.
- The library is installed through pip i.e. through the command **“pip install wiggles.”**
- Here, **vs code** is used to code and test out the results.
- The code is written to best find the solution of the given problem and then is evaluated and displayed using the inbuilt **‘show()’** or the **‘compare()’** function in wiggles.

PROBLEM

- Implementing and Verifying commutative property of convolution.

PROGRAM CODE

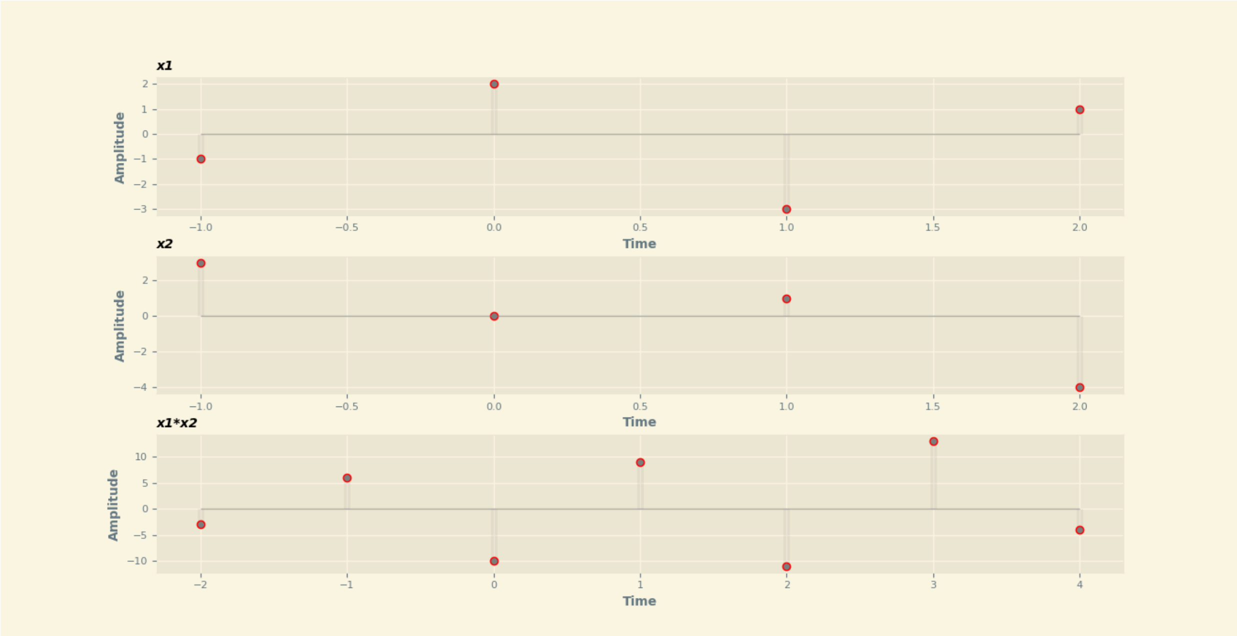
```
#from wiggles import signals as sp

#making two test signals
x1 = sp.discrete([-1,2,-3,1],-1)
x1.name="x1"
x2 = sp.discrete([3,0,1,-4],-3)
x2.name="x2"

'''
Commutative Property of Convolution:
The commutative property of convolution states that the order in which
we convolve two signals does not change the result,
i.e.,
x1(t)*x2(t)=x2(t)*x1(t)
'''

#Calculating LHS:
lhs = x1.convolve(x2)

#Calculating RHS:
rhs = x2.convolve(x1)

#LHS = RHS, Displaying both the signal
lhs.compare(rhs)
```

OUTPUT

Printed notation, LHS & RHS after the operation

```
x1*x2
[     -3     6     -10     9     -11     13     -4     ]
                                  ↑

x2*x1
[     -3     6     -10     9     -11     13     -4     ]
                                  ↑
```

Plotted graph, LHS & RHS after the operation



The comparison between LHS & RHS after the operation plotted in the discrete time domain using a user defined function. Represented through a stem graph.

2. DISTRIBUTIVE PROPERTY

Theory

1. The **distributive property of convolution** states that if there are three signals **x1(t),x2(t)and x3(t),** then the convolution of x1(t) is distributive over the **addition,**

Expression

$$x1(t)*[x2(t) + x3(t)] = [x1(t)*x2(t)] + [x1(t)*x3(t)]$$

Getting the environment ready

- **Python 3.10** is installed in the system and added to the system variables.
- The library is installed through pip i.e. through the command **“pip install wiggles.”**
- Here, **vs code** is used to code and test out the results.

- The code is written to best find the solution of the given problem and then is evaluated and displayed using the inbuilt **‘show()’** or the **‘compare()’** function in wiggles.

PROBLEM

- Implementing and Verifying distributive property of convolution.

PROGRAM CODE

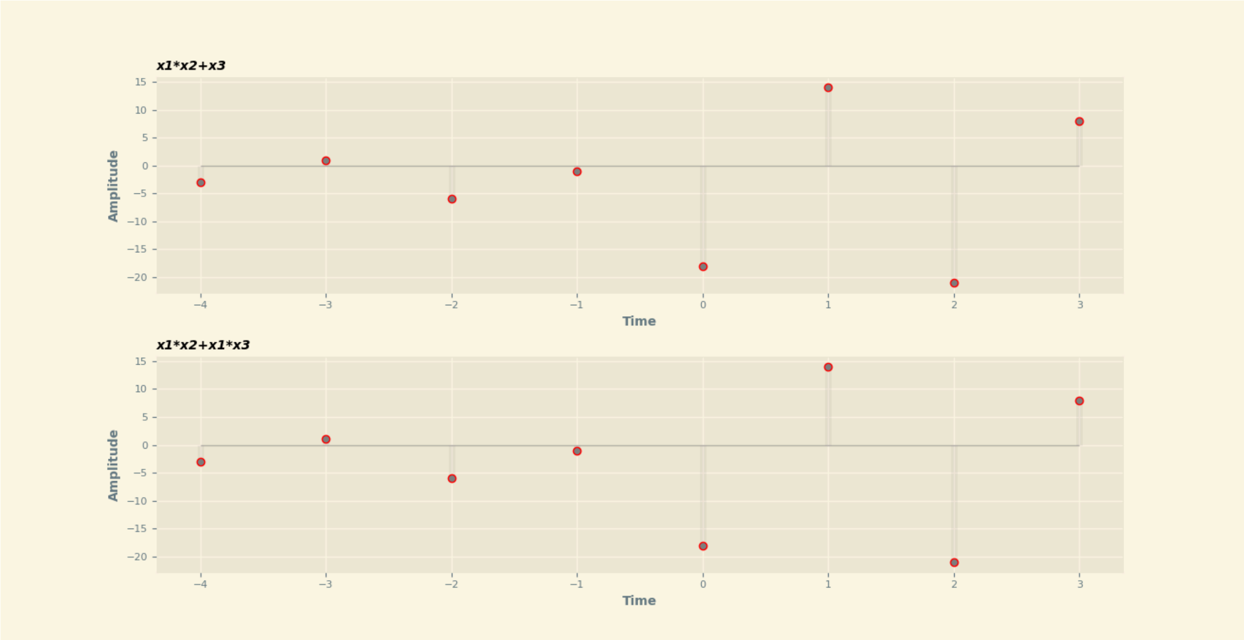
```
#from wiggles import signals as sp

#making two test signals
x1 = sp.discrete([-1,2,-3,1],-1)
x1.name="x1"
x2 = sp.discrete([3,0,1,-4],-3)
x2.name="x2"
x3 = sp.discrete([5,6,7,8],-2)
x3.name="x3"

'''
Distributive Property of Convolution:
The distributive property of convolution states that
if there are three signals x1(t),x2(t)and x3(t),
then the convolution of x1(t) is distributive over the addition,
x1(t)*[x2(t)+x3(t)] = [x1(t)*x2(t)]+[x1(t)*x3(t)]
'''

#Calculating LHS:
lhs = x1.convolve(x2+x3)

#Calculating RHS:
rhs = (x1.convolve(x2))+(x1.convolve(x3))

#LHS = RHS, Displaying both the signal
lhs.compare(rhs)
```

OUTPUT

Printed notation, LHS & RHS after the operation

```
x1*x2+x3
[      -3      1      -6      -1      -18      14      -21      8      ]
                                       ↑
x1*x2+x1*x3
[      -3      1      -6      -1      -18      14      -21      8      ]
                                       ↑
```

Plotted graph, LHS & RHS after the operation



The comparison between LHS & RHS after the operation plotted in the discrete time domain using a user defined function. Represented through a stem graph.

3. ASSOCIATIVE PROPERTY

Theory

1. **The associative property** of convolution states that the way in which the signals are **grouped** in a convolution does not change the result, **i.e.,**

Expression

$$x1(t) * [\, x2(t) * x3(t)\,] = [\, x1(t) * x2(t\,)] * x3(t)$$

Getting the environment ready

- **Python 3.10** is installed in the system and added to the system variables.
- The library is installed through pip i.e. through the command **“pip install wiggles.”**
- Here, **vs code** is used to code and test out the results.

- The code is written to best find the solution of the given problem and then is evaluated and displayed using the inbuilt **'show()'** or the **'compare()'** function in wiggles.

PROBLEM

- Implementing and Verifying associative property of convolution.

PROGRAM CODE

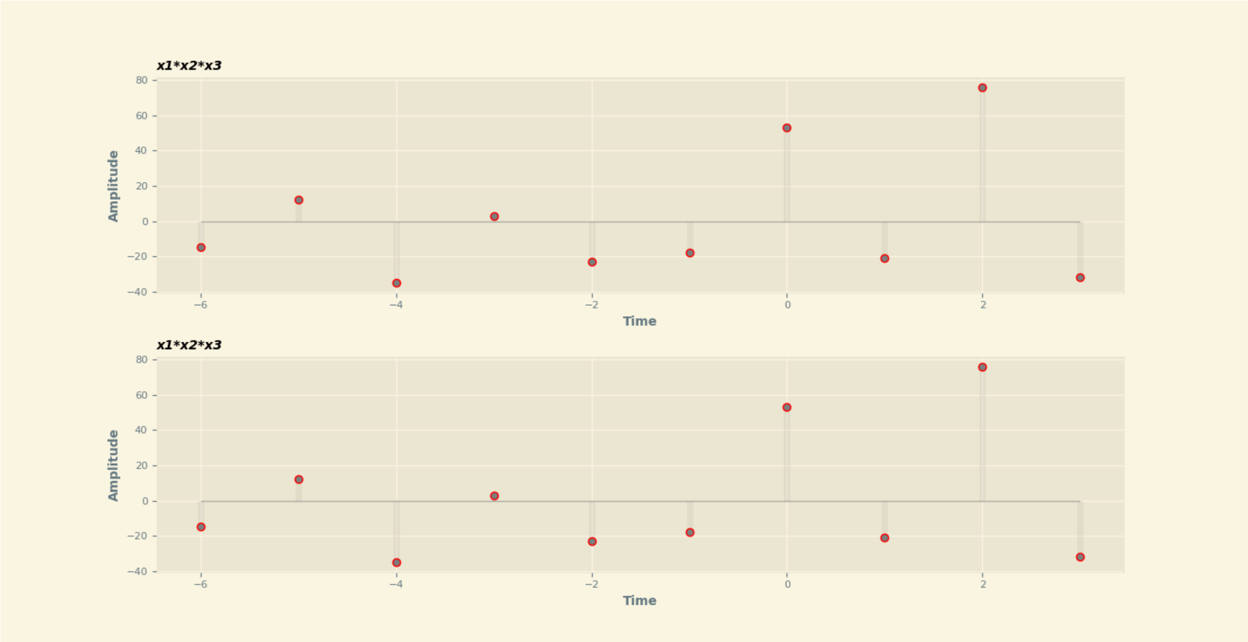
```
#from wiggles import signals as sp

#making two test signals
x1 = sp.discrete([-1,2,-3,1],-1)
x1.name="x1"
x2 = sp.discrete([3,0,1,-4],-3)
x2.name="x2"
x3 = sp.discrete([5,6,7,8],-2)
x3.name="x3"

'''
Associative Property of Convolution:
The associative property of convolution states that the way in which
the signals are grouped in a convolution does not change the result,
i.e.,
x1(t)*[x2(t)*x3(t)] = [x1(t)*x2(t)]*x3(t)
'''

#Calculating LHS:
lhs = x1.convolve(x2.convolve(x3))

#Calculating RHS:
rhs = (x1.convolve(x2)).convolve(x3)

#LHS = RHS, Displaying both the signal
lhs.compare(rhs)
```

OUTPUT

Printed notation, LHS & RHS after the operation

```
x1*x2*x3
[     -15     12     -35     3     -23     -18     53     -21     76     -32     ]
                                                  ↑
x1*x2*x3
[     -15     12     -35     3     -23     -18     53     -21     76     -32     ]
                                                  ↑
```

Plotted graph, LHS & RHS after the operation



The comparison between LHS & RHS after the operation plotted in the discrete time domain using a user defined function. Represented through a stem graph.