WIGGLES

EXPERIMENT #06

OCTOBER 2022

EXPERIMENT NAME

Performing operations like addition, subtraction and multiplication between two different discrete signals.

1. ADDITION

Theory

1. The **sum** of **two discrete** time signals x**1[n] and** x**2[n]** can be obtained by **adding** their values at every instant of time**.**

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 the result after **addition** of two discrete signals .

PROGRAM CODE

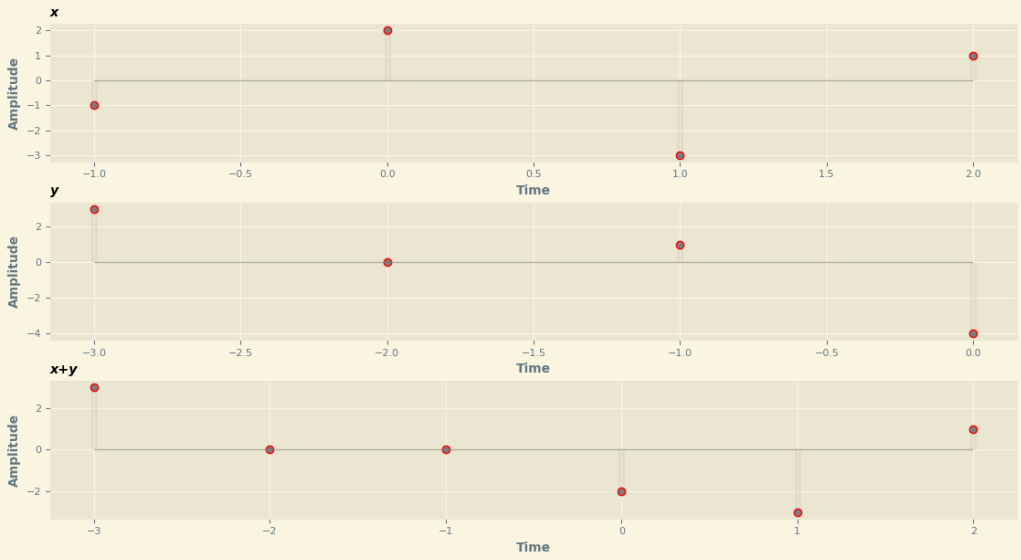
```
#from wiggles import signals as sp

#making two test signals (Starting from different index)
x = sp.discrete([-1,2,-3,1],-1)
x.name="x"
y = sp.discrete([3,0,1,-4],-3)
y.name="y"

#Performing operation and displaying the result
result = x+y
x.compare(y,result)
```

OUTPUT

Printed notation

```
x
[      -1      2       -3      1       ]
               ↑
y
[      3       0       1       -4      ]
                               ↑
x+y
[      3       0       0       -2      -3      1       ]
                               ↑
```

Plotted graph, before and after the operation



The comparison between the signals before and after the operation plotted in the discrete time domain using a user defined function. Represented through a stem graph.

2. SUBTRACTION

Theory

2. The **difference** of **two discrete** time signals x**1[n] and** x**2[n]** can be obtained by **subtracting** their values at every instant of time**.**

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 the result after **subtraction** of two discrete signals .

PROGRAM CODE

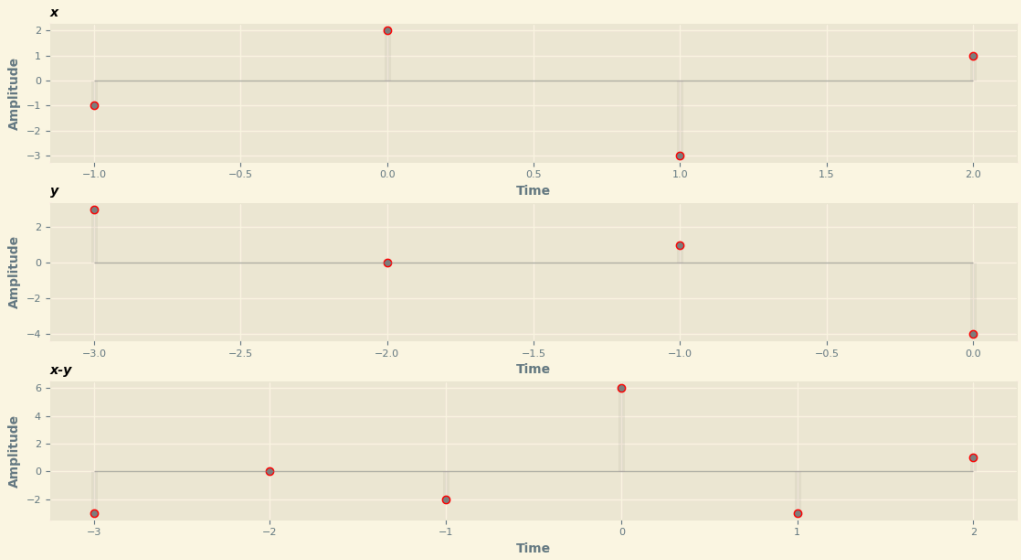
```
#from wiggles import signals as sp

#making two test signals (Starting from different index)
x = sp.discrete([-1,2,-3,1],-1)
x.name="x"
y = sp.discrete([3,0,1,-4],-3)
y.name="y"

#Performing operation and displaying the result
result = x-y
x.compare(y,result)
```

OUTPUT

Printed notation

```
x
[       -1      2       -3      1       ]
          ↑
y
[       3       0       1       -4      ]
                                  ↑
x-y
[       -3      0       -2      6       -3      1       ]
                                  ↑
```

Plotted graph, before and after the operation



The comparison between the signals before and after the operation plotted in the discrete time domain using a user defined function. Represented through a stem graph.

3. MULTIPLICATION

Theory

3. The **multiplication** of **two discrete** time signals x**1[n] and** x**2[n]** can be obtained by **multiplying** their values at every instant of time**.**

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 the result after **multiplication** of two discrete signals .

PROGRAM CODE

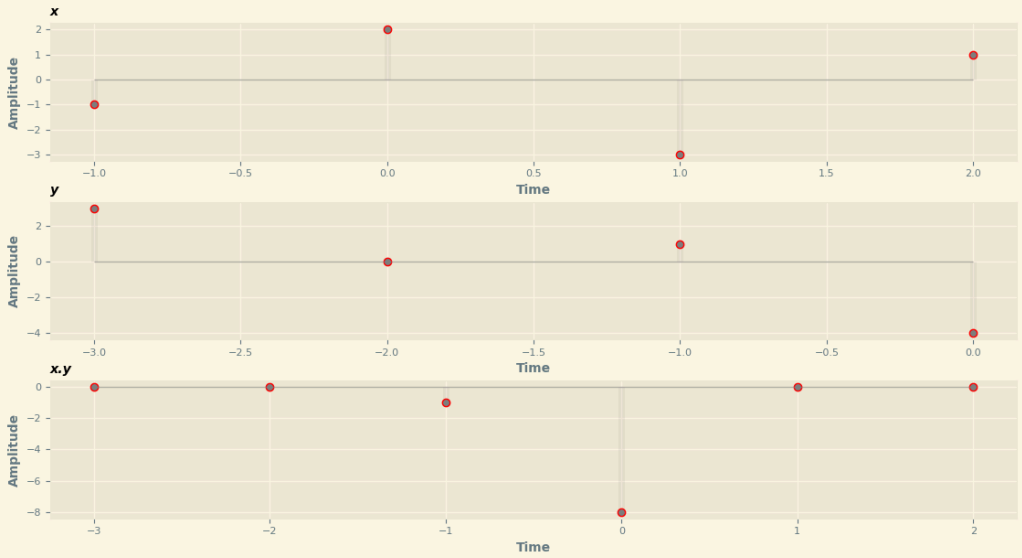
```
#from wiggles import signals as sp

#making two test signals (Starting from different index)
x = sp.discrete([-1,2,-3,1],-1)
x.name="x"
y = sp.discrete([3,0,1,-4],-3)
y.name="y"

#Performing operation and displaying the result
result = x*y
x.compare(y,result)
```

OUTPUT

Printed notation

```
x
[       -1      2       -3      1       ]
                ↑
y
[       3       0       1       -4      ]
                                ↑
x.y
[       0       0       -1      -8      0       0       ]
                                ↑
```

Plotted graph, before and after the operation



The comparison between the signals before and after the operation plotted in the discrete time domain using a user defined function. Represented through a stem graph.

4. DIVISION

Theory

4. The **division** of **two discrete** time signals x**1[n] and** x**2[n]** can be obtained by **dividing** their values at every instant of time**.**

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 the result after **division** of two discrete signals .

PROGRAM CODE

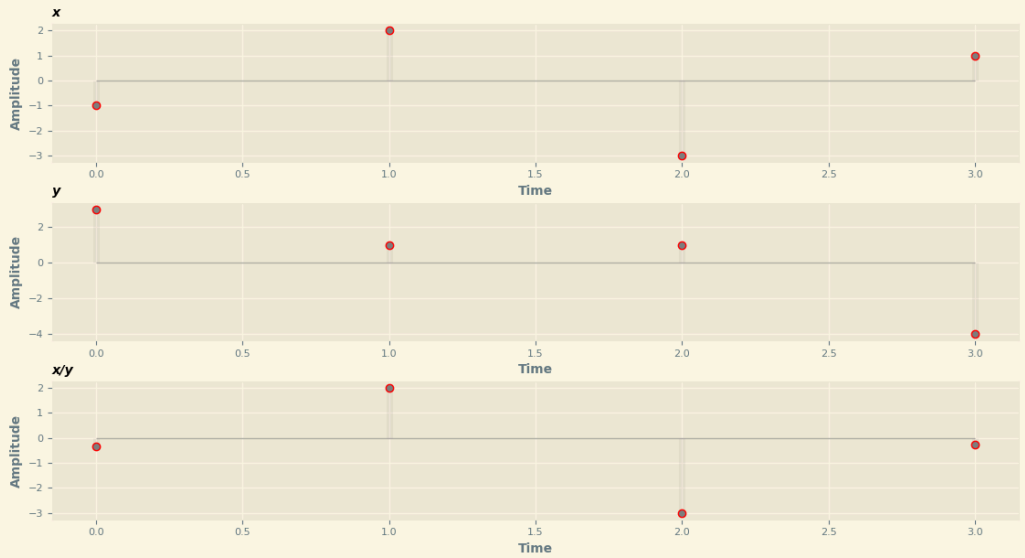
```
#from wiggles import signals as sp

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

#Performing operation and displaying the result
result = x/y
x.compare(y,result)
```

OUTPUT

Printed notation

```
x
[       -1      2       -3      1       ]
        ↑
y
[       3       1       1       -4      ]
        ↑
x/y
[       -0.3333333333333333     2.0     -3.0    -0.25   ]
        ↑
```

Plotted graph, before and after the operation



The comparison between the signals before and after the operation plotted in the discrete time domain using a user defined function. Represented through a stem graph.