WIGGLES

EXPERIMENT #07

OCTOBER 2022

EXPERIMENT NAME

Computing and displaying even and odd components of a signal.

1. EVEN COMPONENT

Theory

1. A signal is said to be an **even signal** if it is **symmetrical about the vertical axis or time origin.**
2. Every signal need not be either **purely even signal or purely odd signal**, but the signal can be expressed as **the sum of even and odd components**.
3. The even component of any signal can be calculated by,

Expression

$$xe(t) = 1/2\,[\,x(t) + x(-\,t)\,]$$

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

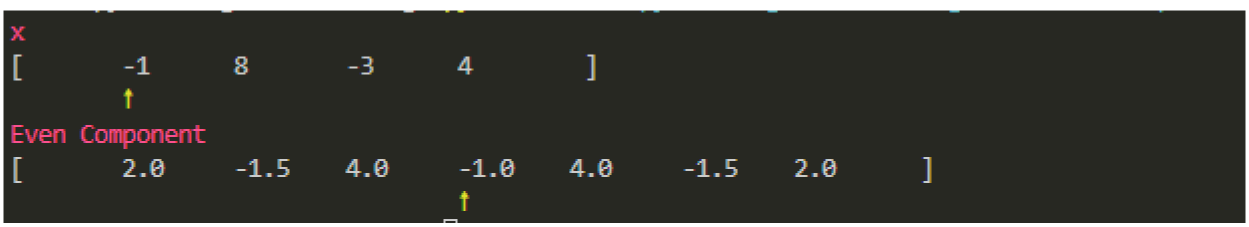
- Implementing and Verifying calculation of **even components** of a signal.

PROGRAM CODE

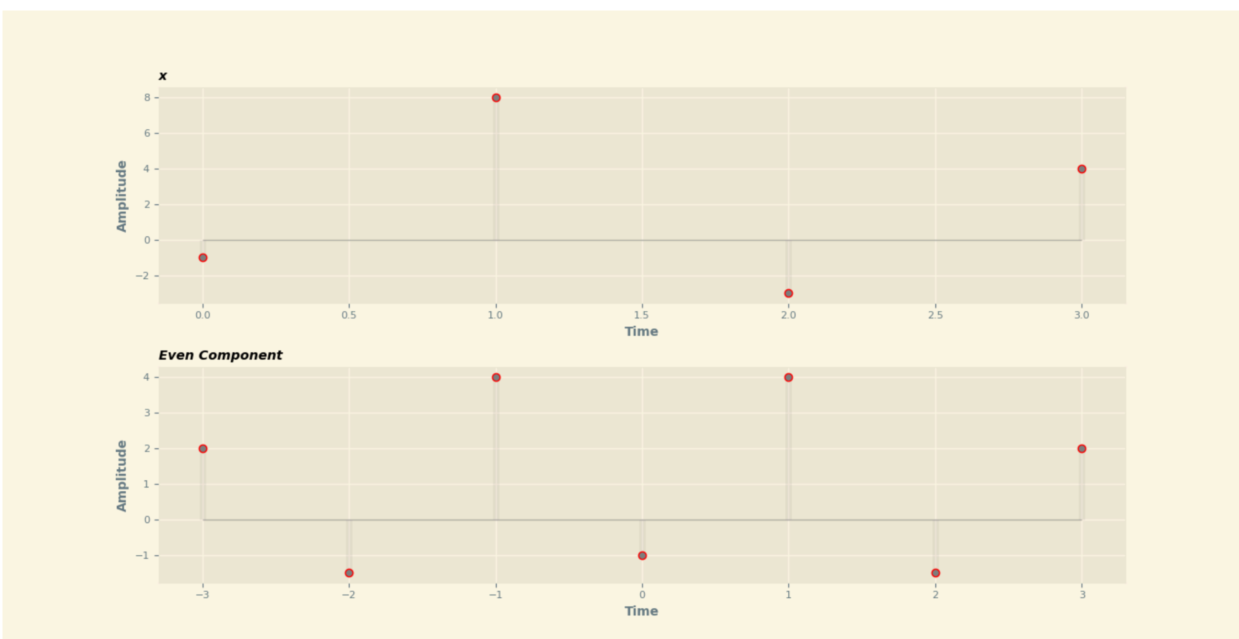
```
#from wiggles import signals as sp

#making a test signal
x = sp.discrete([-1,8,-3,4],0)
x.name="x"

#Finding the component and displaying the result
component = x.even_component()
component.name = "Even Component"
x.compare(component)
```

OUTPUT

Printed notation


```
x
[      -1      8      -3      4      ]
Even Component
[      2.0     -1.5    4.0     -1.0    4.0     -1.5    2.0     ]
```

Plotted graph



The comparison between the original and the computed component of the signal plotted in the discrete time domain using a user defined function. Represented through a stem graph.

2. ODD COMPONENT

Theory

1. A signal is said to be an **odd signal** if it is **anti-symmetrical about the vertical axis.**
2. Every signal need not be either **purely even signal or purely odd signal**, but the signal can be expressed as **the sum of even and odd components**.
3. The even component of any signal can be calculated by,

Expression

$$xe(t) = 1/2\,[\,x(t)\ -\ x(-\,t)\,]$$

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 calculation of **odd components** of a signal.

PROGRAM CODE

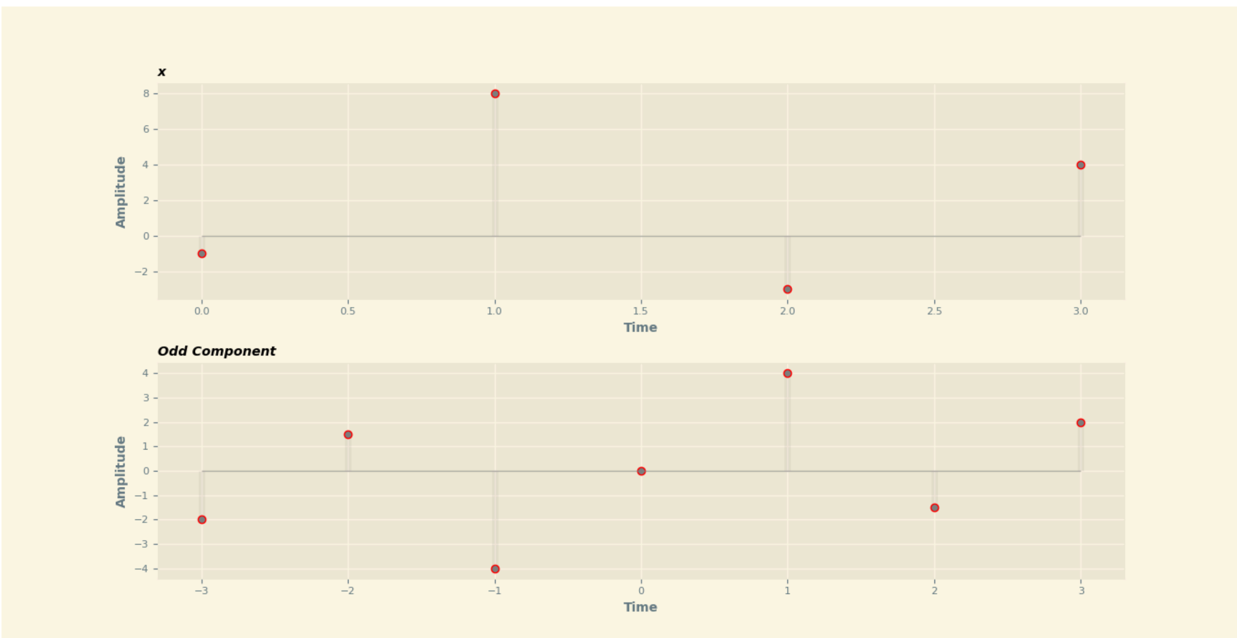
```
#from wiggles import signals as sp

#making a test signal
x = sp.discrete([-1,8,-3,4],0)
x.name="x"

#Finding the component and displaying the result
component = x.odd_component()
component.name = "Odd Component"
x.compare(component)
```

OUTPUT

Printed notation

```
x
[       -1      8       -3      4       ]
        ↑
Odd Component
[       -2.0    1.5     -4.0    0.0     4.0     -1.5    2.0     ]
                                ↑
```

Plotted graph



The comparison between the original and the computed component of the signal plotted in the discrete time domain using a user defined function. Represented through a stem graph.

3. VERIFICATION

Theory

1. Every signal need not be either **purely even signal or purely odd signal**, but the signal can be expressed as **the sum of even and odd components**.i.e.,

Expression

$$x(t) = xe(t) + xo(t)$$

- $xe\ (t)$ is the even component of the signal, and
- $xo\ (t)$ is the odd component of the signal.,

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

- To verify by **adding** the **even and odd components** of the signal in order to get back the **original signal.**

PROGRAM CODE

```
#from wiggles import signals as sp

#making a test signal
x = sp.discrete([-1,8,-3,4],0)
x.name="x"

#Finding the even component
even = x.even_component()
even.name = "Even Component"
```

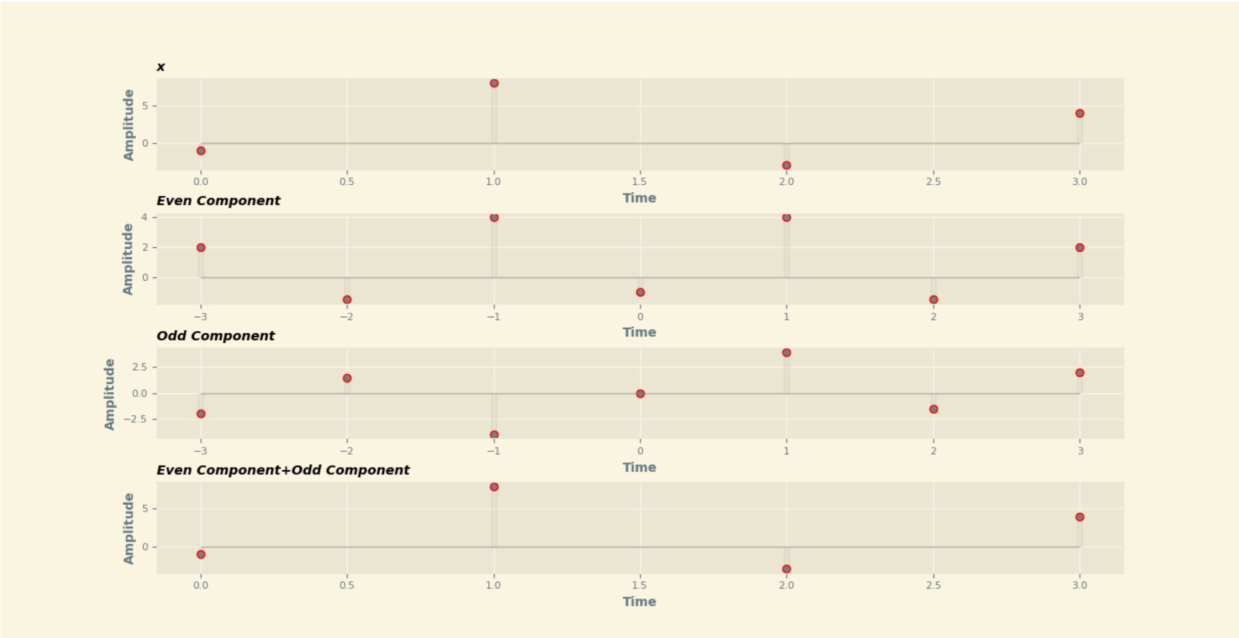
```
#Finding the odd component
odd = x.odd_component()
odd.name = "Odd Component"

#Adding two components
'''
since,
Every signal need not be either purely even signal or purely odd signal,
but the signal can be expressed as the sum of even and odd components.
x(t) = xe (t) + xo (t)
Where,
      xe (t) is the even component of the signal, and
      xo (t) is the odd component of the signal.
'''
verify = even + odd
verify.trim()

#displaying the results
x.compare(even,odd,verify)
```

OUTPUT

Printed notation

```
x
[       -1      8       -3      4       ]
        ↑
Even Component
[       2.0     -1.5    4.0     -1.0    4.0     -1.5    2.0     ]
                                ↑
Odd Component
[       -2.0    1.5     -4.0    0.0     4.0     -1.5    2.0     ]
                                ↑
Even Component+Odd Component
[       -1.0    8.0     -3.0    4.0     ]
        ↑
```

Plotted graph



The comparison between the original and the addition of both the computed components of the signal plotted in the discrete time domain using a user defined function. Represented through a stem graph.