Digital Signal Processing | Lab 3

Consider a sample input signal **S**, **[Your Student ID]** and an impulse

response **H** = **[2, 0, 2, 1]**

If your student ID is 160041010, then **S** = **[1, 6, 0, 0, 4, 1, 0, 1, 0]**

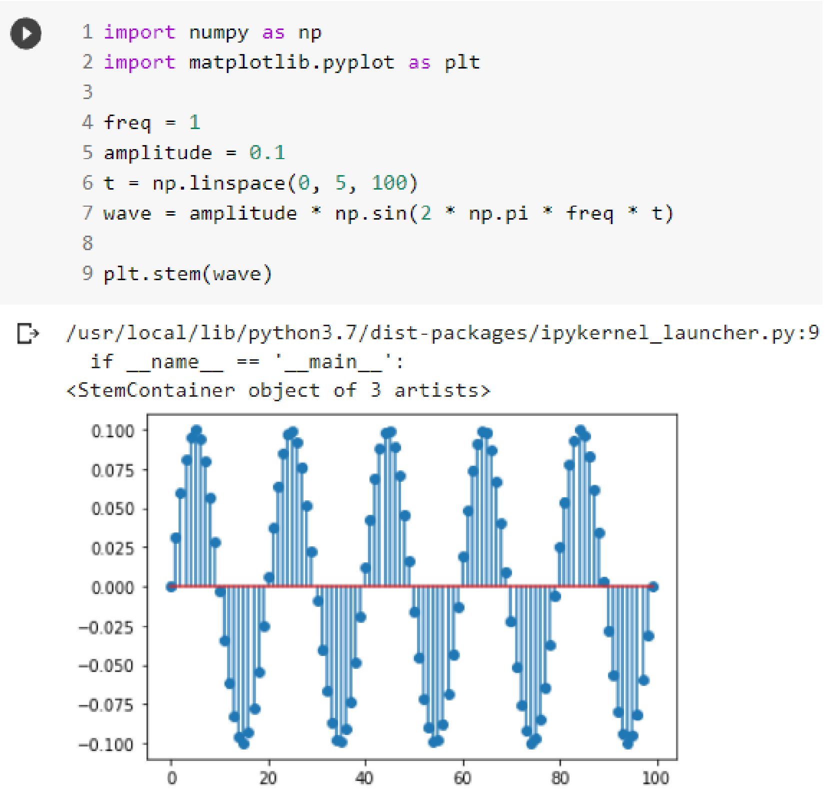
1. Use the built-in **np.convolve** function to convolve S with H. (Use 'same' for padding). Plot the output signal along with the original input signal and impulse response.

2. Write a custom function **InputSideConvolution** that implements convolution using the Input Side Algorithm.

3. Write another function **OutputSideConvolution** that uses the Output Side Algorithm.

4. Does all three produce the same result?

5. First create the signal **wave_plus_ramp** as shown in the given figures.

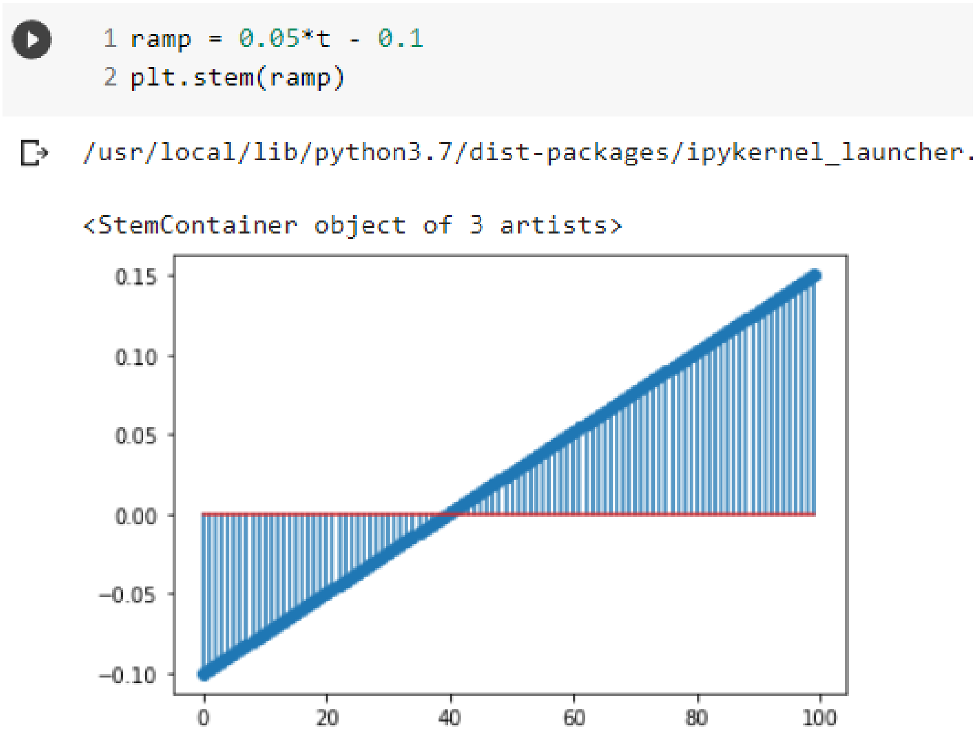


```python
import numpy as np
import matplotlib.pyplot as plt

freq = 1
amplitude = 0.1
t = np.linspace(0, 5, 100)
wave = amplitude * np.sin(2 * np.pi * freq * t)

plt.stem(wave)
```

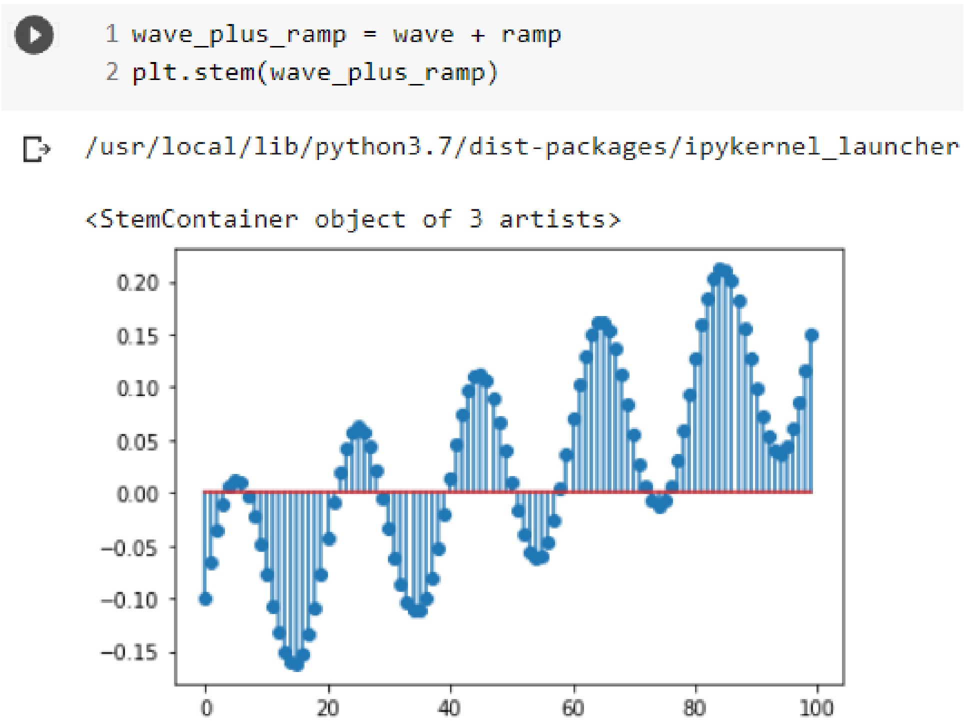
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9
  if __name__ == '__main__':
<StemContainer object of 3 artists>
```



```
ramp = 0.05*t - 0.1
plt.stem(ramp)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.

<StemContainer object of 3 artists>
```



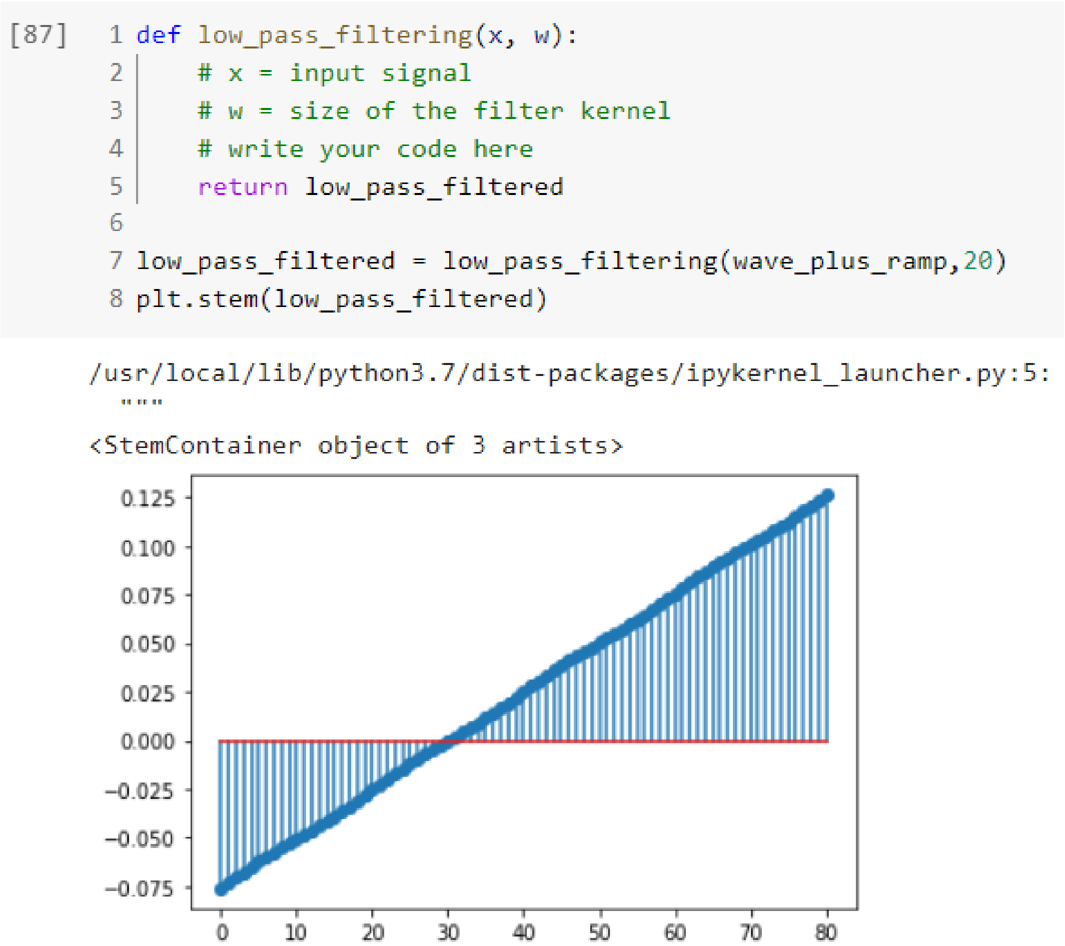
```
wave_plus_ramp = wave + ramp
plt.stem(wave_plus_ramp)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher

<StemContainer object of 3 artists>
```

Now, create a low pass filter kernel with size 20. Hint: make a moving average filter kernel where all of these 20 samples are nonzero. Convolve it with wave_plus_ramp using np.convolve function. (Use ‘valid’ for padding)

Output signal should retain the ramp and discard the wave. Can you tell why?



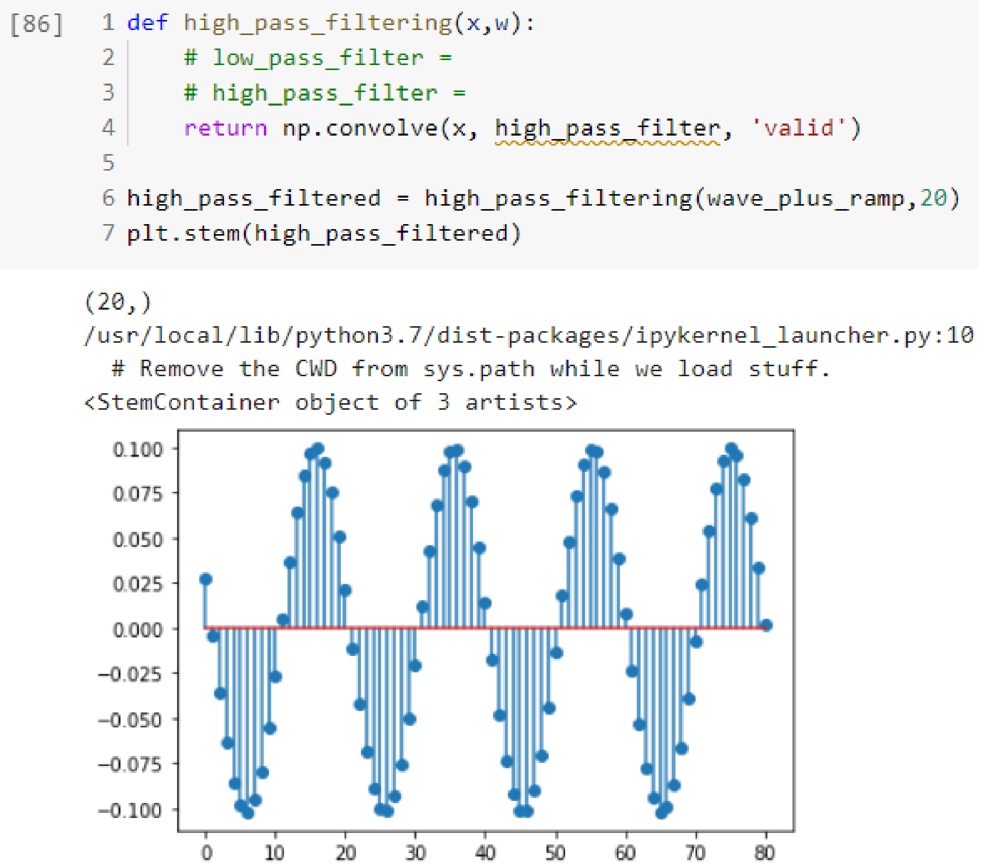
```
[87]
def low_pass_filtering(x, w):
    # x = input signal
    # w = size of the filter kernel
    # write your code here
    return low_pass_filtered

low_pass_filtered = low_pass_filtering(wave_plus_ramp,20)
plt.stem(low_pass_filtered)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5:
  """
<StemContainer object of 3 artists>
```

6. Now, take the low_pass_filter kernel and change every sample's sign. Now add 1 to the sample in the middle of the kernel. This will produce a high_pass filter. Now convolve wave_plus_ramp with high_pass filter kernel using np.convolve. (Use 'valid' for padding)

This will discard the ramp and retain the wave. Can you tell why?



[86]

```
def high_pass_filtering(x,w):
    # low_pass_filter =
    # high_pass_filter =
    return np.convolve(x, high_pass_filter, 'valid')

high_pass_filtered = high_pass_filtering(wave_plus_ramp,20)
plt.stem(high_pass_filtered)
```

```
(20,)
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:10
  # Remove the CWD from sys.path while we load stuff.
<StemContainer object of 3 artists>
```