plot

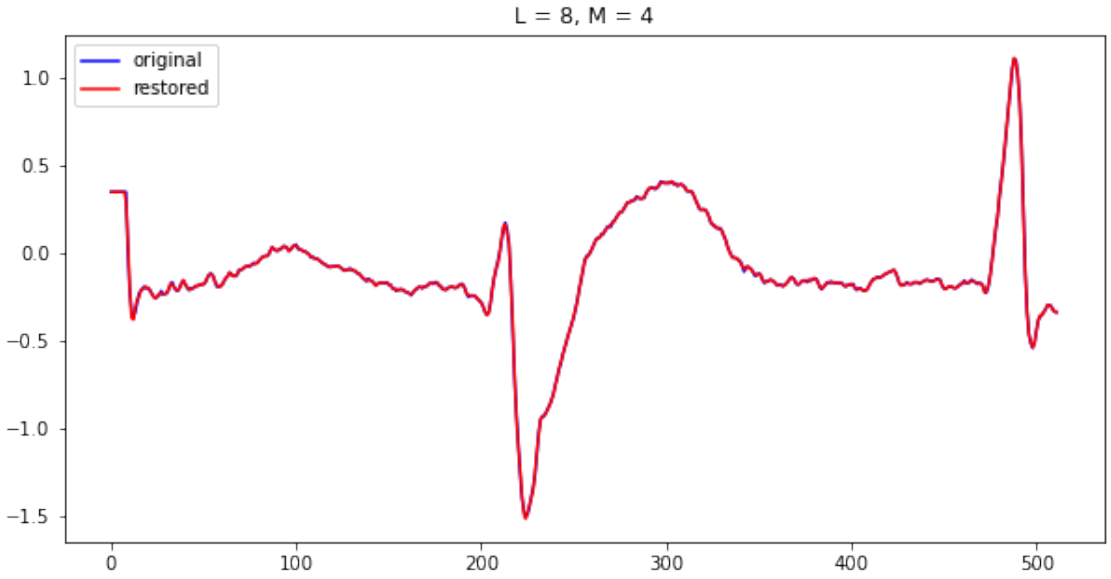
October 5, 2022

0.0.1 Malthe Tøttrup 201907882 au644177

[ ]:
```
import matplotlib.pyplot as plt
import numpy as np
```

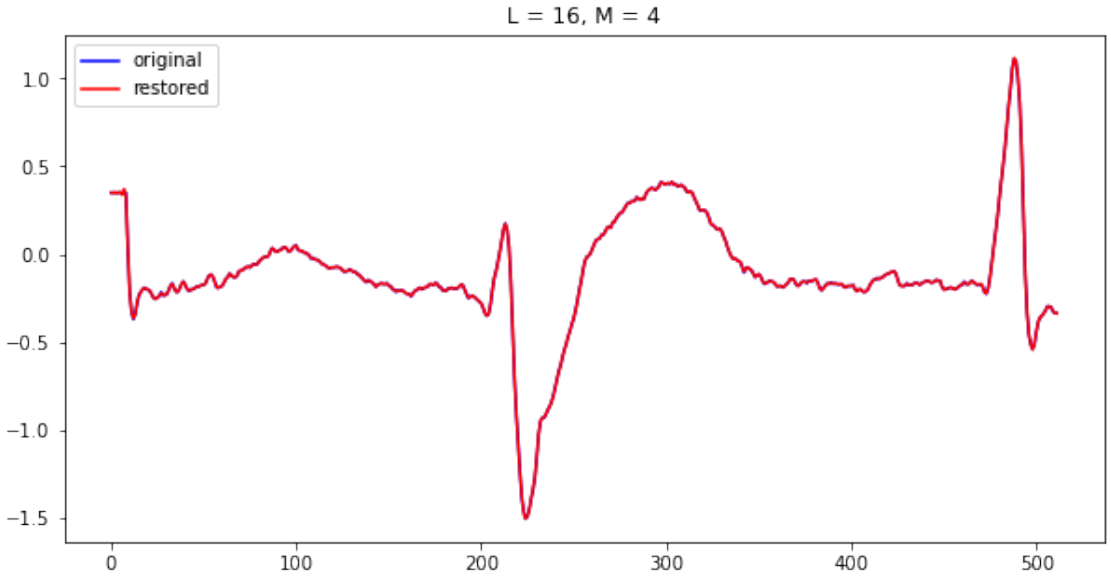
[5]:
```
"""TEST 1: L = 8, M = 4"""

with open('x_L8_M4.txt', 'r') as f:
    x = f.read().split("\n")


with open('ecgsig.txt', 'r') as f:
    ecgsig = f.read().split("\n")


plt.rcParams['figure.figsize'] = [10,5]
plt.title("L = 8, M = 4")
plt.plot(np.array(ecgsig).astype(float), 'b')
plt.plot(np.array(x).astype(float),  'r')
plt.legend(["original", "restored"])

print("Time for compression incl. H = 1765 [m/s]")
print("Time for compression W/O. H = 1203 [m/s]")
```

```
Time for compression incl. H = 1765 [m/s]
Time for compression W/O. H = 1203 [m/s]
```



[6]:
```
"""TEST 2: L = 16, M = 4"""

with open('x_L16_M4.txt', 'r') as f:
    x = f.read().split("\n")


with open('ecgsig.txt', 'r') as f:
    ecgsig = f.read().split("\n")


plt.rcParams['figure.figsize'] = [10,5]
plt.title("L = 16, M = 4")
plt.plot(np.array(ecgsig).astype(float), 'b')
plt.plot(np.array(x).astype(float),  'r')
plt.legend(["original", "restored"])

print("Time for compression incl. H = 6023 [m/s]")
print("Time for compression W/O. H = 3609 [m/s]")
```

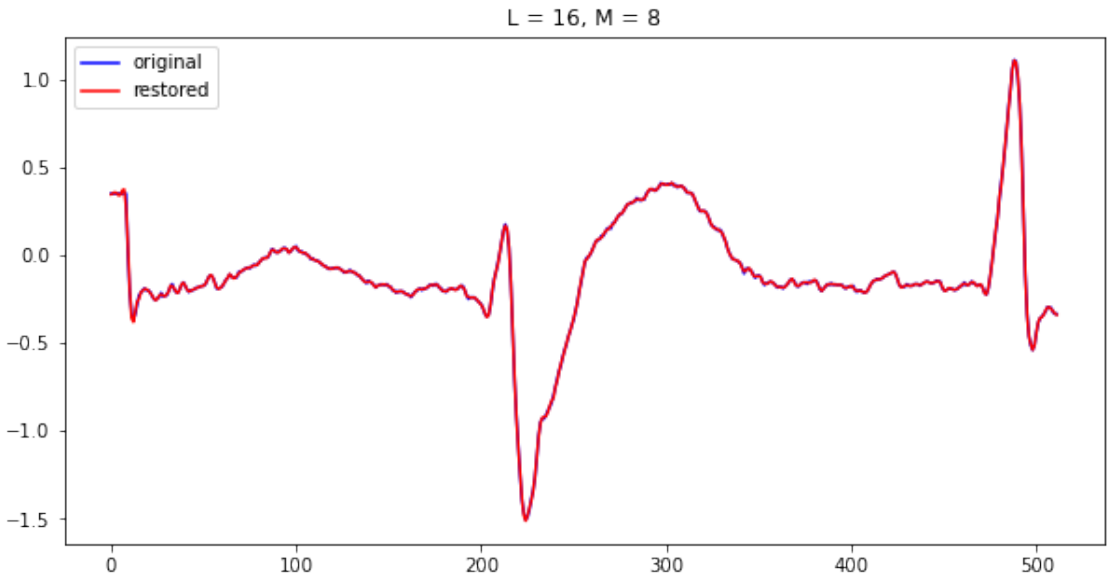
```
Time for compression incl. H = 6023 [m/s]
Time for compression W/O. H = 3609 [m/s]
```



As expected it takes longer to generate H and compress as we increase L.

[7]:
```
"""TEST 3: L = 16, M = 8"""

with open('x_L16_M8.txt', 'r') as f:
    x = f.read().split("\n")


with open('ecgsig.txt', 'r') as f:
    ecgsig = f.read().split("\n")


plt.rcParams['figure.figsize'] = [10,5]
plt.title("L = 16, M = 8")
plt.plot(np.array(ecgsig).astype(float), 'b')
plt.plot(np.array(x).astype(float),  'r')
plt.legend(["original", "restored"])

print("Time for compression incl. H = 4828 [m/s]")
print("Time for compression W/O. H = 2406 [m/s]")
```

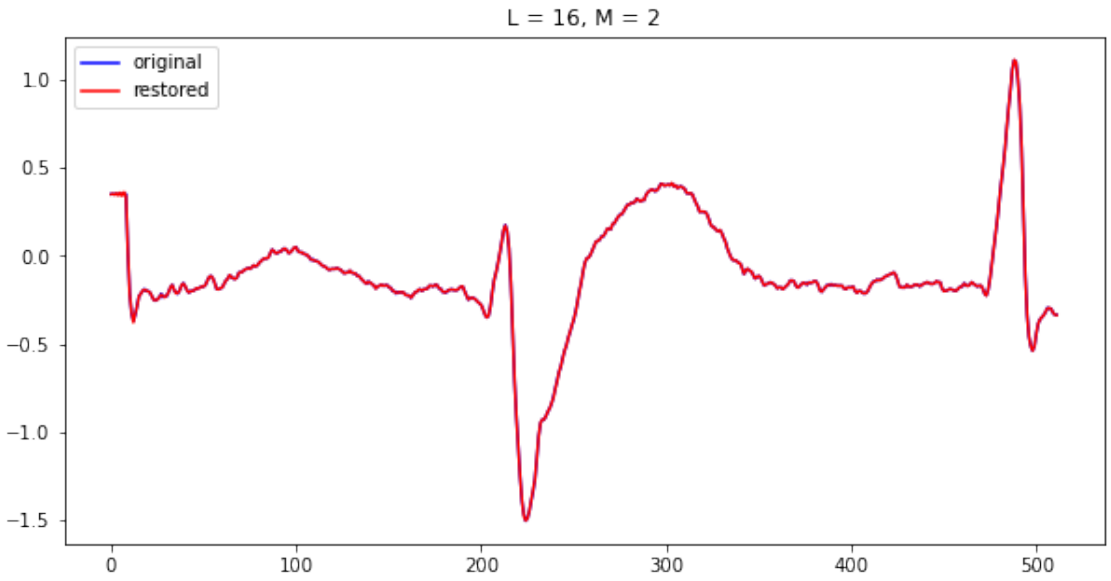
```
Time for compression incl. H = 4828 [m/s]
Time for compression W/O. H = 2406 [m/s]
```



If we keep the ratio between L and M the same as test 1, we see that it is rougly eqivalet to multiplying the compression time with factor 2. Although is takes more time if we include the generation of H.

[8]:
```
"""TEST 4: L = 16, M = 2"""

with open('x_L16_M2.txt', 'r') as f:
    x = f.read().split("\n")


with open('ecgsig.txt', 'r') as f:
    ecgsig = f.read().split("\n")


plt.rcParams['figure.figsize'] = [10,5]
plt.title("L = 16, M = 2")
plt.plot(np.array(ecgsig).astype(float), 'b')
plt.plot(np.array(x).astype(float),  'r')
plt.legend(["original", "restored"])

print("Time for compression incl. H = 6648 [m/s]")
print("Time for compression W/O. H = 4226 [m/s]")
```

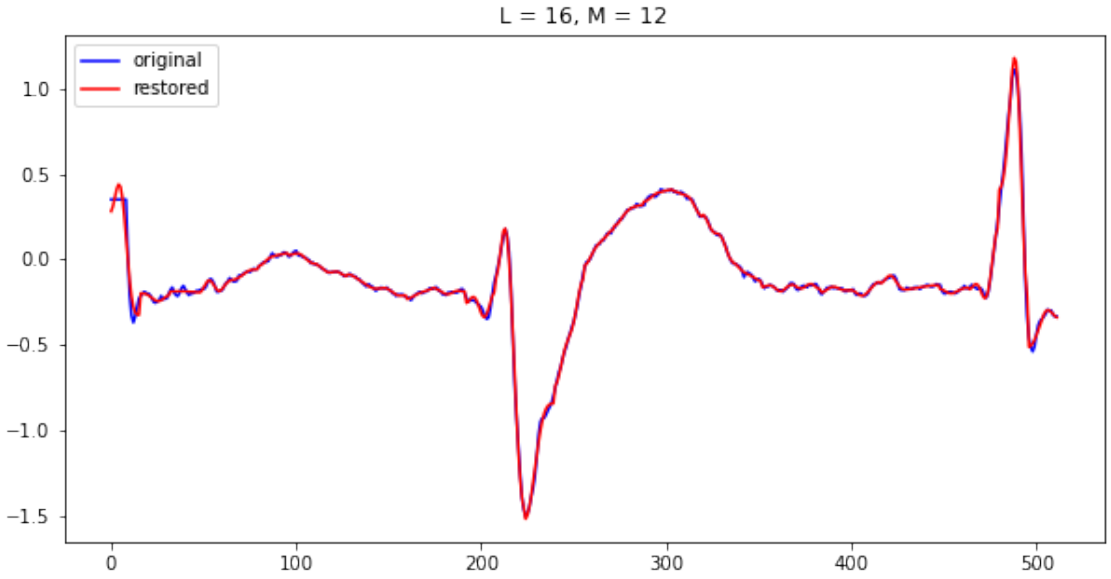
```
Time for compression incl. H = 6648 [m/s]
Time for compression W/O. H = 4226 [m/s]
```



Significant increase in compression time if we take a small M.

[9]:
```
"""TEST 5: L = 16, M = 12"""

with open('x_L16_M12.txt', 'r') as f:
    x = f.read().split("\n")


with open('ecgsig.txt', 'r') as f:
    ecgsig = f.read().split("\n")


plt.rcParams['figure.figsize'] = [10,5]
plt.title("L = 16, M = 12")
plt.plot(np.array(ecgsig).astype(float), 'b')
plt.plot(np.array(x).astype(float),  'r')
plt.legend(["original", "restored"])

print("Time for compression incl. H = 3609 [m/s]")
print("Time for compression W/O. H = 1187 [m/s]")
```

```
Time for compression incl. H = 3609 [m/s]
Time for compression W/O. H = 1187 [m/s]
```



Significant decrease in compression time if we take a large M. However, the reconstructed singal is not as good.