Q4

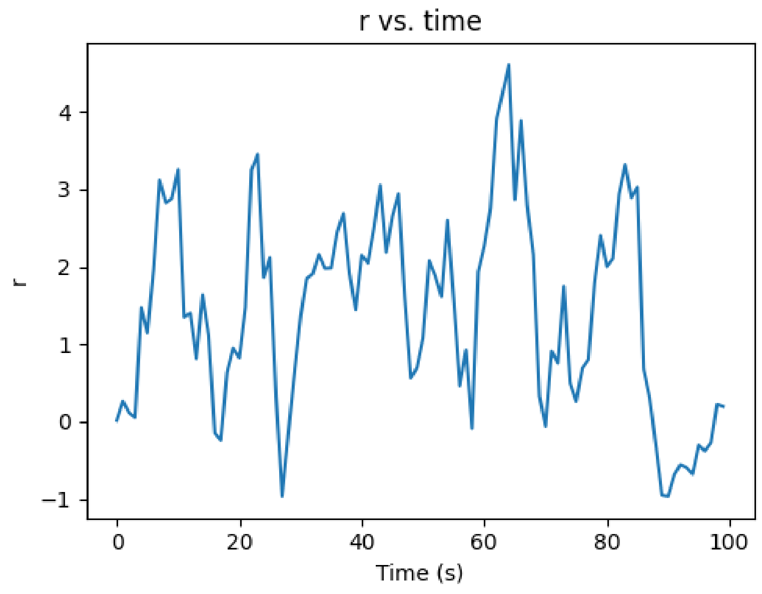
Deconvolution of the Haemodynamic Response. Neuronal activity causes local changes in deoxyhemoglobin concentration in the blood, which can be measured using magnetic resonance imaging (MRI). One drawback of this is that the haemodynamic response is both delayed and slower than the underlying neural responses. We can model the delay and spread of the measurements relative to the neural signals using a linear shift-invariant system:

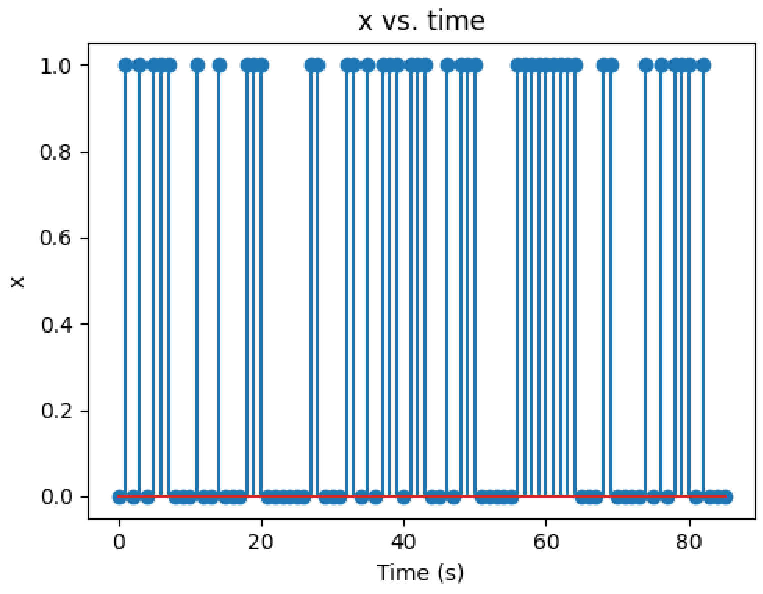
$$ r(n) = \sum_k x(n-k) h(k) $$

where x(n) is an input signal delivered over time (for example, a sequence of light intensities), h(k) is the haemodynamic response to a single light flash at time k = 0 (i.e., the impulse response of the MRI measurement), and r(n) is the MRI response to the full input signal. In the file hrfDeconv.mat, you will find a response vector r and an input vector x containing a sequence of impulses (indicating flashes of light). Your goal is to estimate the HRF, h, from the data. Each of these signals are sampled at 1 Hz. Plot vectors r and x versus time to get a sense for the data. (Use the stem command for x, and label the x-axis).

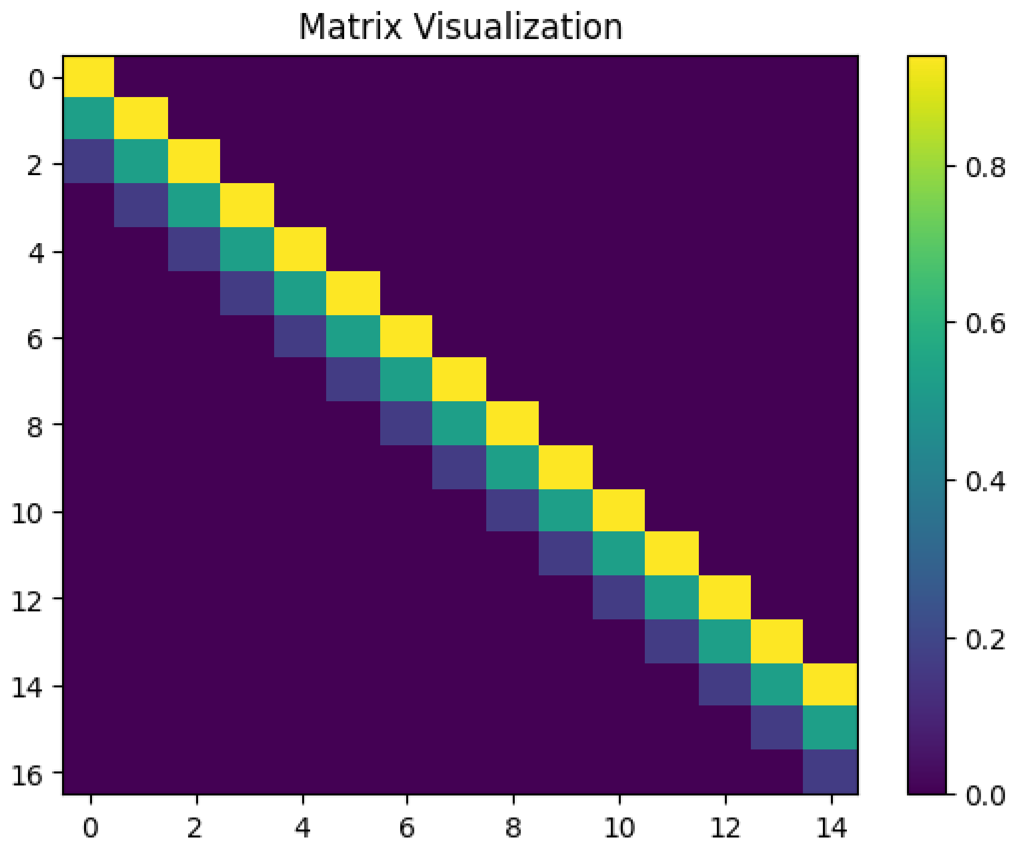
a)

Convolution is linear, and thus we can re-write the equation above as a matrix multiplication, r = Xh, where h is a vector of length M, and X is an N + M - 1 x M matrix (N is the length of the input x). Write a matlab function createConvMat, that takes as arguments an input vector x andM (the dimensionality of h) and generates a matrix X such that the response r = Xh is as defined in Eq. (1) for any h. Verify that the matrix generated by your function produces the same response as MatLab’s conv function when applied to a few random h vectors of length M = 15. Visualize the matrix X as an image (evaluate imagesc(X)), and describe its structure.

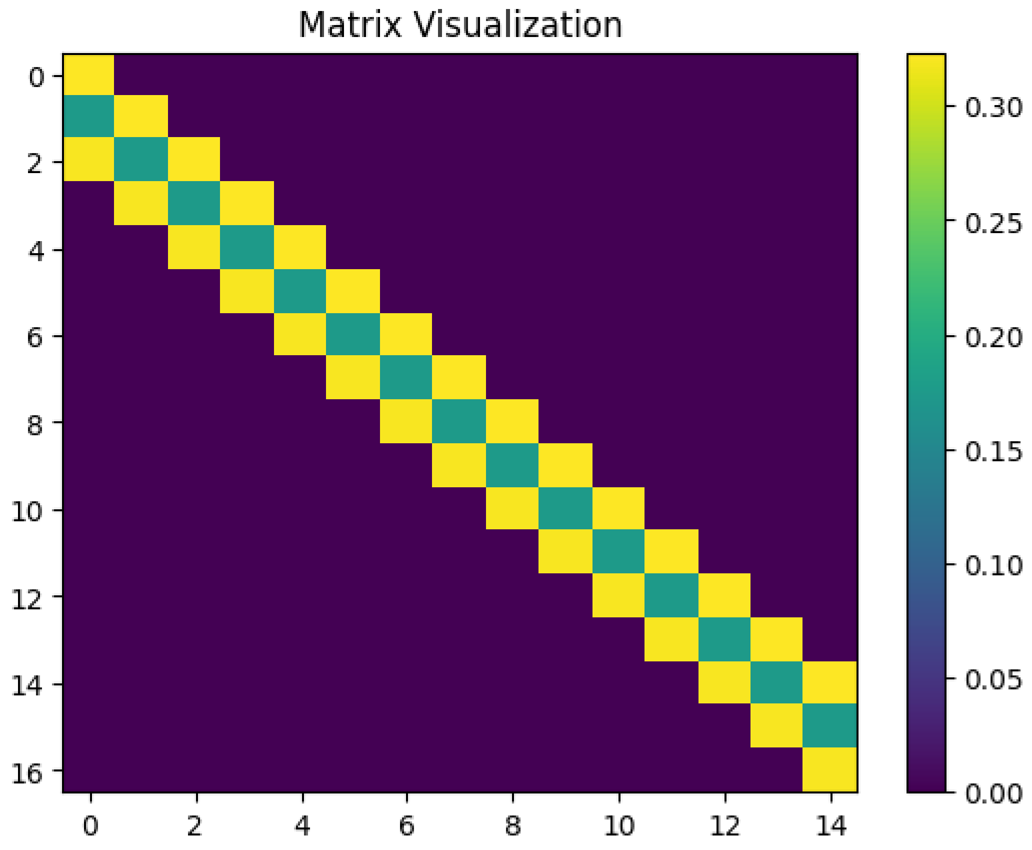




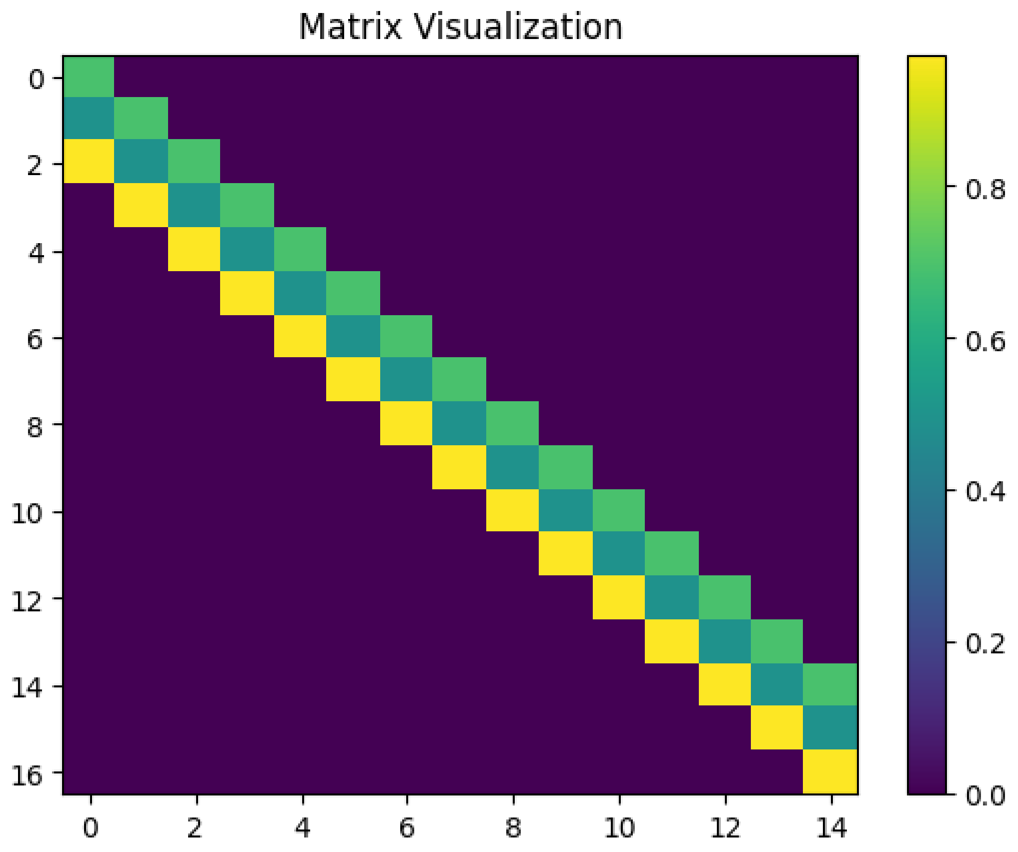
```
--- Iteration 1 ---
Convolution Matrix:
[[0.93957289 0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.52926266 0.93957289 0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.1653172  0.52926266 0.93957289 0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.1653172  0.52926266 0.93957289 0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.1653172  0.52926266 0.93957289 0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.1653172  0.52926266 0.93957289
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.1653172  0.52926266
  0.93957289 0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.1653172
  0.52926266 0.93957289 0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.1653172  0.52926266 0.93957289 0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.1653172  0.52926266 0.93957289 0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.1653172  0.52926266 0.93957289 0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.1653172  0.52926266 0.93957289
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.1653172  0.52926266
  0.93957289 0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.1653172
  0.52926266 0.93957289 0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.1653172  0.52926266 0.93957289]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.1653172  0.52926266]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.1653172 ]]
```



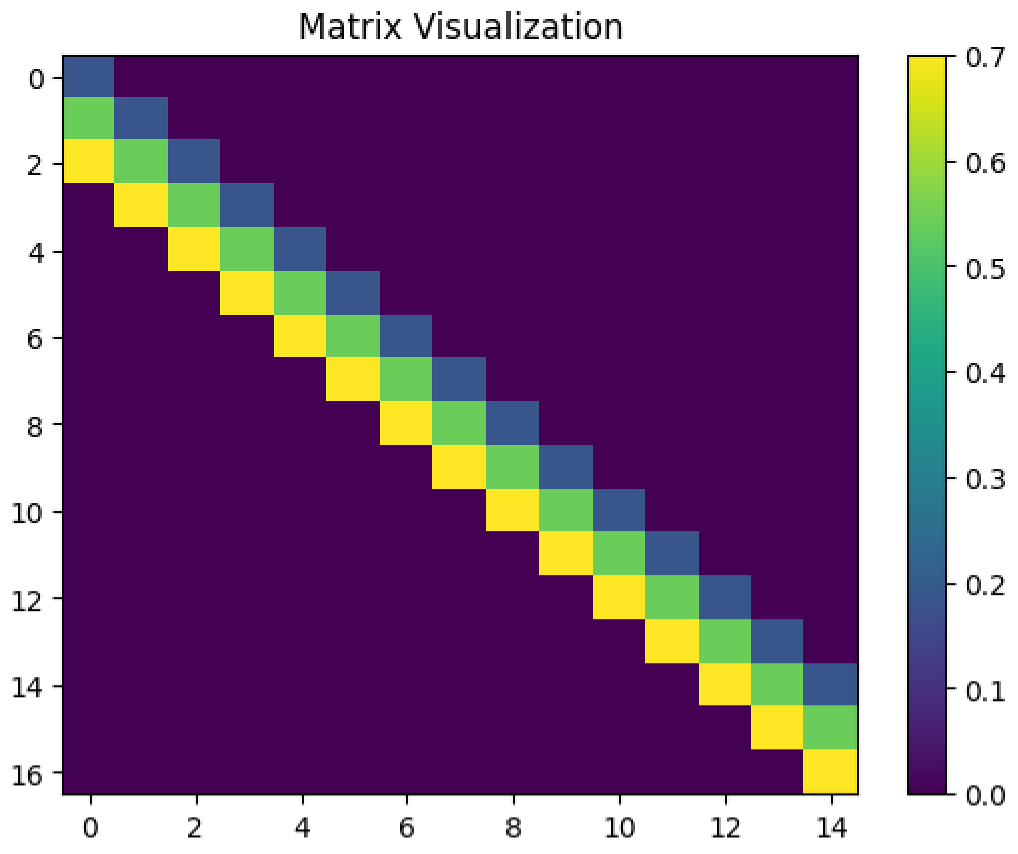
```
Result from matrix multiplication:
[0.31161484 0.72359915 1.216561   0.65144075 0.73738714 0.59197582
 0.30929994 0.69678066 0.50627793 0.66617625 0.87357342 1.33219339
 0.68058231 0.22035075 0.42043119 0.22745732 0.06968326]

Result from numpy's convolution:
[0.31161484 0.72359915 1.216561   0.65144075 0.73738714 0.59197582
 0.30929994 0.69678066 0.50627793 0.66617625 0.87357342 1.33219339
 0.68058231 0.22035075 0.42043119 0.22745732 0.06968326]

Difference between methods:
[0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.11022302e-16 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.11022302e-16 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00]

--- Iteration 2 ---
Convolution Matrix:
[[0.32332243 0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.17569569 0.32332243 0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.31981234 0.17569569 0.32332243 0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.31981234 0.17569569 0.32332243 0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.31981234 0.17569569 0.32332243 0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.31981234 0.17569569 0.32332243
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.31981234 0.17569569
  0.32332243 0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.31981234
  0.17569569 0.32332243 0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.31981234 0.17569569 0.32332243 0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.31981234 0.17569569 0.32332243 0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.31981234 0.17569569 0.32332243 0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.31981234 0.17569569 0.32332243
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.31981234 0.17569569
  0.32332243 0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.31981234
  0.17569569 0.32332243 0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.31981234 0.17569569 0.32332243]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.31981234 0.17569569]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.31981234]]
```



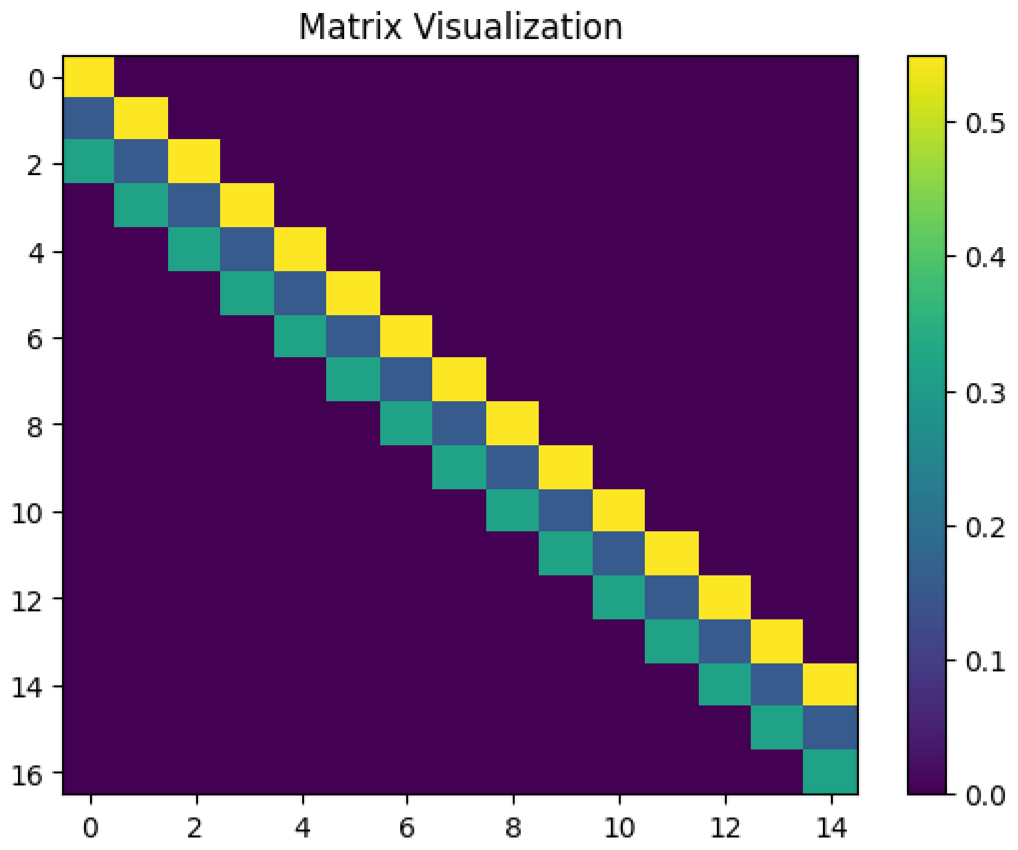
```
Result from matrix multiplication:
[0.14435434 0.39876425 0.6170684  0.58779037 0.66428325 0.52473317
 0.56884602 0.48964771 0.48596014 0.46888064 0.45863602 0.42647628
 0.31093511 0.3676672  0.25166036 0.16846291 0.10744128]

Result from numpy's convolution:
[0.14435434 0.39876425 0.6170684  0.58779037 0.66428325 0.52473317
 0.56884602 0.48964771 0.48596014 0.46888064 0.45863602 0.42647628
 0.31093511 0.3676672  0.25166036 0.16846291 0.10744128]

Difference between methods:
[ 0.00000000e+00  0.00000000e+00 -1.11022302e-16 -1.11022302e-16
  0.00000000e+00  1.11022302e-16  0.00000000e+00  0.00000000e+00
  0.00000000e+00  0.00000000e+00  0.00000000e+00 -5.55111512e-17
  0.00000000e+00 -5.55111512e-17  0.00000000e+00  0.00000000e+00
  0.00000000e+00]

--- Iteration 3 ---
Convolution Matrix:
[[0.69189475 0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.49752632 0.69189475 0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.97298361 0.49752632 0.69189475 0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.97298361 0.49752632 0.69189475 0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.97298361 0.49752632 0.69189475 0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.97298361 0.49752632 0.69189475
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.97298361 0.49752632
  0.69189475 0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.97298361
  0.49752632 0.69189475 0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.97298361 0.49752632 0.69189475 0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.97298361 0.49752632 0.69189475 0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.97298361 0.49752632 0.69189475 0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.97298361 0.49752632 0.69189475
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.97298361 0.49752632
  0.69189475 0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.97298361
  0.49752632 0.69189475 0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.97298361 0.49752632 0.69189475]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.97298361 0.49752632]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.97298361]]
```



```
Result from matrix multiplication:
[0.58184116 0.50215383 1.20298251 0.462446   1.22295889 0.79926541
 1.26131573 0.51724577 0.74687168 0.83102868 0.89990101 0.8390699
 0.5403871  0.81936244 1.03029782 0.92763384 0.82324185]

Result from numpy's convolution:
[0.58184116 0.50215383 1.20298251 0.462446   1.22295889 0.79926541
 1.26131573 0.51724577 0.74687168 0.83102868 0.89990101 0.8390699
 0.5403871  0.81936244 1.03029782 0.92763384 0.82324185]

Difference between methods:
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  5.55111512e-17
  0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
  0.00000000e+00 -1.11022302e-16  0.00000000e+00  0.00000000e+00
  0.00000000e+00  0.00000000e+00  2.22044605e-16  0.00000000e+00
  0.00000000e+00]

--- Iteration 4 ---
Convolution Matrix:
[[0.18663657 0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.5436644  0.18663657 0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.70018043 0.5436644  0.18663657 0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.70018043 0.5436644  0.18663657 0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.70018043 0.5436644  0.18663657 0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.70018043 0.5436644  0.18663657
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.70018043 0.5436644
  0.18663657 0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.70018043
  0.5436644  0.18663657 0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.70018043 0.5436644  0.18663657 0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.70018043 0.5436644  0.18663657 0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.70018043 0.5436644  0.18663657 0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.70018043 0.5436644  0.18663657
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.70018043 0.5436644
  0.18663657 0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.70018043
  0.5436644  0.18663657 0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.70018043 0.5436644  0.18663657]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.70018043 0.5436644 ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.70018043]]
```



```
Result from matrix multiplication:
[0.00909165 0.1257986  0.4240517  0.84457971 0.98380776 0.92598323
 0.49984719 0.67608692 1.14646019 0.79387851 0.32948729 0.61178598
 0.772695   1.02431397 0.97284232 0.66029784 0.46332998]

Result from numpy's convolution:
[0.00909165 0.1257986  0.4240517  0.84457971 0.98380776 0.92598323
 0.49984719 0.67608692 1.14646019 0.79387851 0.32948729 0.61178598
 0.772695   1.02431397 0.97284232 0.66029784 0.46332998]

Difference between methods:
[ 0.00000000e+00  0.00000000e+00  5.55111512e-17 -1.11022302e-16
  0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 -2.22044605e-16  0.00000000e+00  0.00000000e+00 -1.11022302e-16
  0.00000000e+00  2.22044605e-16  0.00000000e+00  0.00000000e+00
  0.00000000e+00]

--- Iteration 5 ---
Convolution Matrix:
[[0.54957809 0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.15813343 0.54957809 0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.31633154 0.15813343 0.54957809 0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.31633154 0.15813343 0.54957809 0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.31633154 0.15813343 0.54957809 0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.31633154 0.15813343 0.54957809
  0.         0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.31633154 0.15813343
  0.54957809 0.         0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.31633154
  0.15813343 0.54957809 0.         0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.31633154 0.15813343 0.54957809 0.         0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.31633154 0.15813343 0.54957809 0.         0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.31633154 0.15813343 0.54957809 0.
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.31633154 0.15813343 0.54957809
  0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.31633154 0.15813343
  0.54957809 0.         0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.31633154
  0.15813343 0.54957809 0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.31633154 0.15813343 0.54957809]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.31633154 0.15813343]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.31633154]]
```



```
Result from matrix multiplication:
[0.52205728 0.23947402 0.7875133  0.37827378 0.85800248 0.30688419
 0.85132309 0.50344336 0.91584525 0.79343061 0.88333215 0.48702742
 0.72130154 0.22622544 0.5488364  0.10992444 0.16462678]

Result from numpy's convolution:
[0.52205728 0.23947402 0.7875133  0.37827378 0.85800248 0.30688419
 0.85132309 0.50344336 0.91584525 0.79343061 0.88333215 0.48702742
 0.72130154 0.22622544 0.5488364  0.10992444 0.16462678]

Difference between methods:
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
  1.11022302e-16 -5.55111512e-17  0.00000000e+00  0.00000000e+00
  0.00000000e+00  0.00000000e+00  0.00000000e+00 -5.55111512e-17
  0.00000000e+00 -2.77555756e-17  0.00000000e+00  0.00000000e+00
  0.00000000e+00]
```

The convolution matrix appears to be a diagonal matrix.

b)

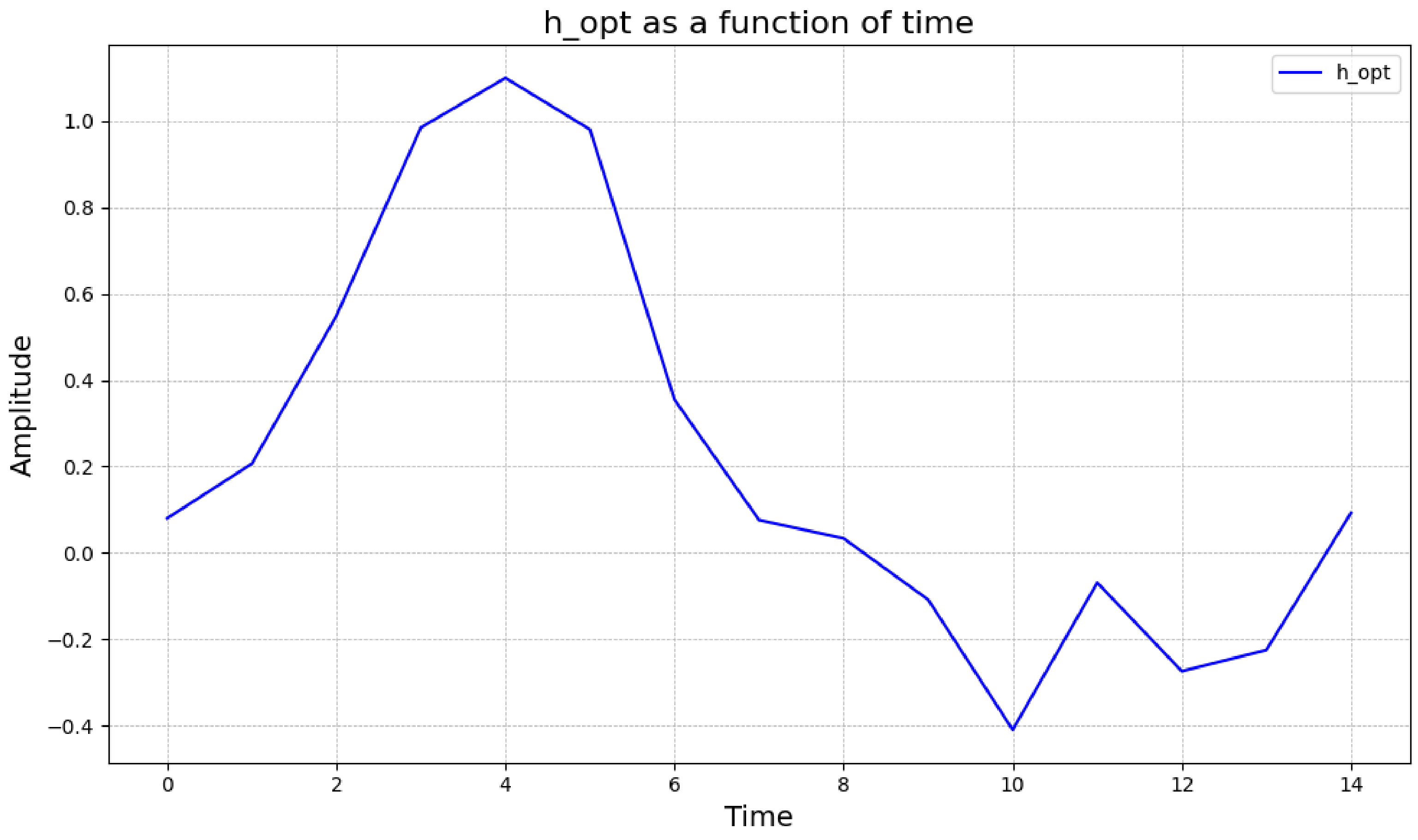
Now, given the X generated by your function for M = 15, solve for h by formulating a least-squares regression problem:

$$ h_{opt}=arg \min{||r-Xh||}^2 $$

Plot hopt as a function of time (label your x-axis, including units). How would you describe it? How long does it last?

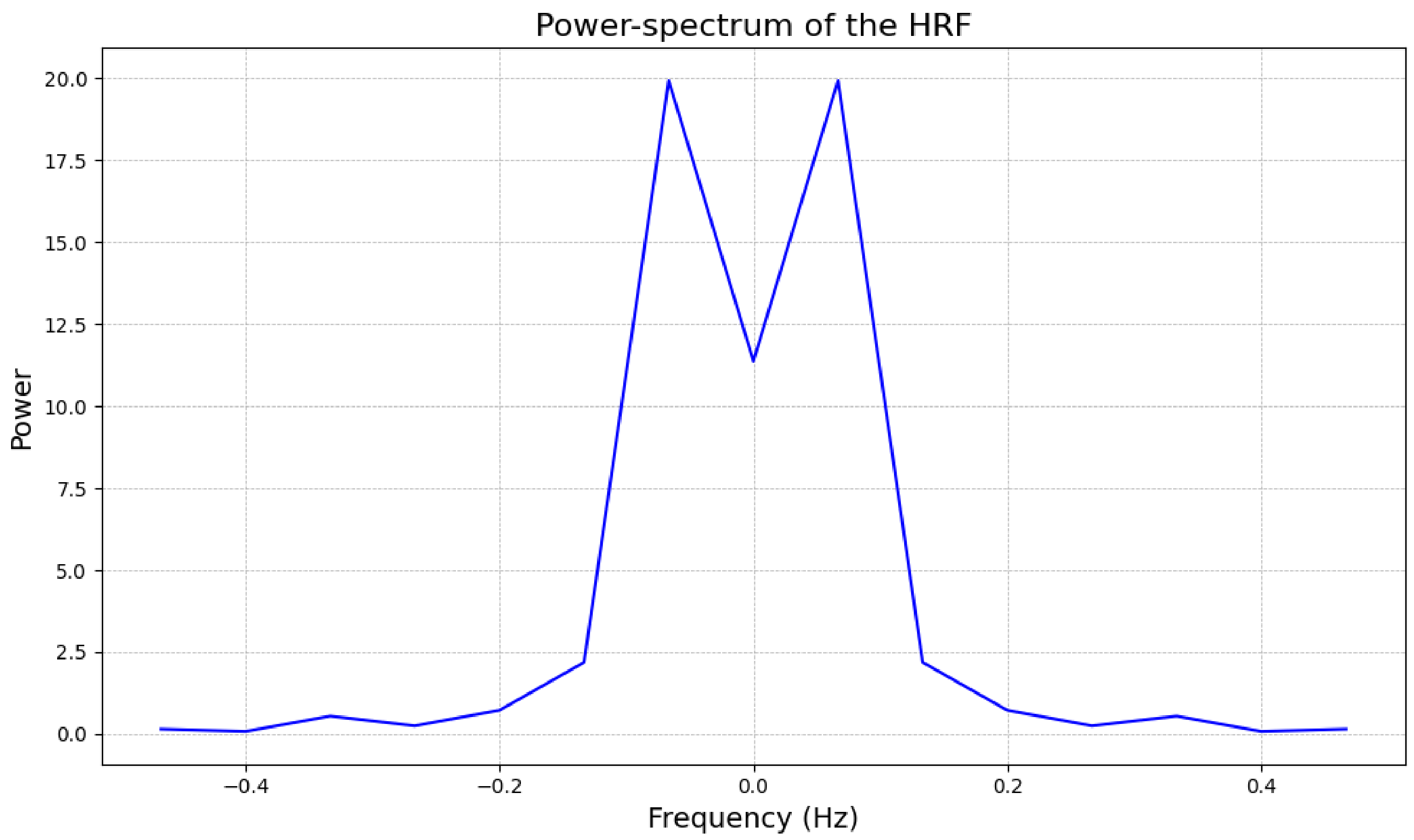
```
Shape of x_data: (86,)
Shape of response_data: (100,)
```

```
Optimal h using library function: [ 0.08038615  0.20651242  0.54851395  0.98458329  1.09930568  0.98008
733
  0.35467779  0.07621158  0.03380402 -0.10810732 -0.40934496 -0.0687829
 -0.27365116 -0.22476736  0.09256217]

Optimal h using Gaussian elimination: [ 0.08038615  0.20651242  0.54851395  0.98458329  1.09930568  0.9
8008733
  0.35467779  0.07621158  0.03380402 -0.10810732 -0.40934496 -0.0687829
 -0.27365116 -0.22476736  0.09256217]

Using Gaussian elimination h_opt_gaussian and library function h_opt are approximately the same.
```



c)

It’s often easier to understand an LSI system by viewing it in the frequency domain. Plot the power-spectrum of the HRF (i.e. jF(h)j2, where F(h) is the Fourier transform of the HRF). Plot this with the zero frequency (DC) in the middle, and label the x axis, in Hz. Based on this plot, what kind of filter is the HRF? Specifically, which frequencies does it allow to pass, and which does it block?



When we look at the power spectrum of the HRF, we're basically seeing which frequencies the most power in the HRF. Think of the HRF as a filter: it's more slow changes (low frequencies) and pass down the quicker, quick changes (high frequencies).