In this assignment you are asked to implement FIR causal Wiener filters in the frequency domain, from a short-time Fourier transform (STFT) implemented with a Fast Fourier Transform applied to consecutive overlapping signal segments. The main related equations are:

$$e(k,n') = d(k,n') - \left[w_0(k) \cdots w_{N-1}(k) \right] \begin{bmatrix} x(k,n') \\ \vdots \\ x(k,n'-N+1) \end{bmatrix}$$

$$\boldsymbol{x}(k,n') = \left[x(k,n') \cdots x(k,n'-N+1) \right]^T \quad \boldsymbol{w}(k) = \left[w_0^*(k) \cdots w_{N-1}^*(k) \right]^T$$

$$\boldsymbol{p}(k) = \underset{n'}{E}\left[\boldsymbol{x}(k,n') d^H(k,n') \right] \quad \boldsymbol{R}(k) = \underset{n'}{E}\left[\boldsymbol{x}(k,n') \boldsymbol{x}^H(k,n') \right]$$

$$\boldsymbol{w}_{opt}(k) = \boldsymbol{R}^{-1}(k) \boldsymbol{p}(k)$$

where k is the FFT bin index and n' is the frame/segment index.

The setup corresponds to a system identification setup, where the unknown system to be identified is a FIR filter of 100 coefficients.

- For the filter input signal $x(n)$ and the desired signal $d(n)$, use the provided .wav files. These files correspond to 500 segments or frames of signal samples, with the segment length and overlap specified below.
- Use an FFT size of 1024 ;
- Use a segment or frame size identical to the FFT size (1024 samples) and a frame shift of 1024/8=128 samples;
- Apply a non-rectangular window to each frame (e.g. Hann window) before computing the FFT.

Some additional background:

- What is the difference between this frequency domain FIR causal Wiener solution with N coeffs. per frequency, and the previously seen frequency domain interpretation of the unconstrained Wiener solution?
  - In terms of cost function, they are not exactly the same: $\underset{n'}{E}\left[\left| e(k,n') \right|^2 \right]$ for the frequency domain FIR causal Wiener solution (where k represents a frequency bin), and $E\left[\left| e(n) \right|^2 \right]$ for the unconstrained Wiener solution. One key difference is that $e(k,n')$ is computed based on frames of finite length. Only for infinite frame lengths does the frequency domain FIR causal Wiener solution with $N=1$ become equivalent with the unconstrained Wiener solution. One consequence is that the frame size (or the equivalent length of all the overlapping frames considered if $N>1$) needs to be longer than the time domain impulse response to model, to avoid significant block processing effects.
  - Furthermore, in practice when the PSDs in the frequency domain unconstrained Wiener solution ($\Phi_{xx}(k), \Phi_{dx}(k)$) are estimated using a Welch method based on finite segment lengths as in assignment #1, they become the same as the PSDs in $\boldsymbol{p}(k) = \underset{n'}{E}\left[\boldsymbol{x}(k,n') d^H(k,n') \right]$ $\boldsymbol{R}(k) = \underset{n'}{E}\left[\boldsymbol{x}(k,n') \boldsymbol{x}^H(k,n') \right]$ estimated in the frequency domain FIR causal Wiener solution for the case $N=1$.
- Having a FIR causal filter of size N applied to the "time" sequence of each frequency bin signal $x(k,n')$, means that a frequency dependent response can be obtained within each bin or frequency "band", as opposed to a flat constant response assumed in the band. So it is similar to increasing the frequency resolution of the overall Wiener filter. It also allows to use FFTs $x(k,n')$ from past frames, and to model unknown systems with longer impulse responses.

- A quick discussion about complexity: If L_h represents the impulse response length for the unknown system to be modeled, the complexity for Wiener solution computation in the time domain would be of order $O\{L_h^3\}$ (with some likely ill-conditioning in the $L_h \times L_h$ $\boldsymbol{R}$ matrix). For the frequency domain computation, it would be order $O\{N_{fft} \times N^3\}$ (or $O\left\{\frac{N_{fft}}{2} \times N^3\right\}$ if we consider symmetry in the frequency domain coefficients). With $L_h = 100$ and $N_{fft} = 1024$, we have $O\{1{,}000{,}000\}$ for time domain computation and $O\{64{,}000\}$ for frequency domain computation, so we see that the frequency domain computation is more efficient. However, a more detailed assessment of complexity would need to evaluate the number of samples and the time required to estimate the correlation functions in each domain, with the cost of this estimation process. The cost of the FFTs or subband filtering should also be considered.

Tasks to do:

1. For $N = 1$ and $N = 5$ filter coefficients in each frequency bin, compute the $\boldsymbol{w}_{opt}(k) = \boldsymbol{R}^{-1}(k)\boldsymbol{p}(k)$ optimal solution in each frequency bin, as well as the resulting theoretical $MMSE(k) = \underset{n'}{E}\left[d(k,n')d^H(k,n')\right] - \boldsymbol{p}^H(k)\boldsymbol{w}_{opt}(k)$ using estimated values for $\underset{n'}{E}\left[d(k,n')d^H(k,n')\right]$ and $\boldsymbol{p}(k)$. Then normalize $MMSE(k)$ by $\underset{n'}{E}\left[d(k,n')d^H(k,n')\right]$, to have a normalized MMSE: $MMSE(k)\Big/\underset{n'}{E}\left[d(k,n')d^H(k,n')\right]$, and convert in dB. Plot the resulting normalized $MMSE(k)$ (in dB, as a function of *k*) and compare the performance for $N = 1$ and $N = 5$.

2. Next estimate $MMSE(k)$ by actually producing an error signal in each bin using the optimal filters found: $e(k,n') = d(k,n') - \left[w_{opt,0}(k) \cdots w_{opt,N-1}(k)\right]\begin{bmatrix} x(k,n') \\ \vdots \\ x(k,n'-N+1) \end{bmatrix}$, and measuring the power $\underset{n'}{E}\left[\left|e(k,n')\right|^2\right]$ in each bin. Again, normalize $MMSE(k)$ by $\underset{n'}{E}\left[d(k,n')d^H(k,n')\right]$ and convert it in dB, and compare the performance for $N = 1$ and $N = 5$.

3. For each segment n', it is possible to collect the filter outputs from all bins *k*: $\left[y(0,n'), y(1,n'), \cdots, y(k,n'), \cdots, y(N_{fft}-1,n')\right]$, with $y(k,n') = \left[w_0(k) \cdots w_{N-1}(k)\right]\begin{bmatrix} x(k,n') \\ \vdots \\ x(k,n'-N+1) \end{bmatrix}$, and to apply an IFFT function to produce an output time domain signal $y(n,n')$ for segment n'. The final output time signal $y(n)$ will be built by the superposition/addition of the overlapping output time domain segments $y(n,n')$ obtained from the IFFTs.

   Before computing the time domain error signal, you should divide the final output signal $y(n)$ by 4:

   $$e(n) = d(n) - \frac{y(n)}{4}.$$

   This is because the use of a non-rectangular window combined with a frame shift of 1024/8 (and not 1024/2) leads to an output signal that has a gain 4 times larger than its "normal" level.

   For $N = 1$ and $N = 5$, compute an estimated MMSE $E\left[\left|e(n)\right|^2\right]$ from $e(n)$, and normalize it by $E\left[\left|d(n)\right|^2\right]$, then convert in dB . When computing $E\left[\left|e(n)\right|^2\right]$, you should discard the first " 7 x frame_shift" samples (896 samples) and the last " 7 x frame_shift" samples, because these include transient effects from the segment by segment processing, as can easily be seen if you plot $e(n)$.