DAAP Homework #1

Marco Furio Colombo - Student ID: 10537094
Matteo Pettenò - Student ID: 10868930

***Abstract* - This report is intended to comment on the key steps in the MATLAB code and on the results obtained for the assigned homework.**

1 Wiener Filter

The first task is to implement the closed-form Wiener-Hopf ideal solution $w_o = R^{-1} \cdot p$. We started with computing the the autocorrelation matrix R through this code snippet:

```
maxlag = M−1;
[Rxx, Rxx_lags] = xcorr(x, maxlag, 'biased');
rpos = Rxx(Rxx_lags >= 0); % Rxx is symmetric
R = toeplitz(rpos);
```

Note that the autocorrelation vector is obtained using the function *xcorr* and is used to generate the corresponding diagonal constant matrix R, with a maximum shift of M taps. The cross correlation p is computed the same way and having both terms allowed us to get the ideal solution w_o. The chosen '*biased*' normalization option of the *xcorr* function scales the elements by one over the length of the x array factor.

In order to avoid the computational costs $[O(n^3)]$ for matrix inversion, we also determine an estimation of the Wiener filter using the Steepest Descent method. To ensure the convergence of the alghorithm the step-size μ is set to be between 0 and $2/\lambda_{max}$: we tried different values and we saw that best results are obtained if $\mu = 0.95 \cdot 2/\lambda_{max}$. Using μ and R's smallest eigenvalue (λ_{min}) we then find the global time constant for convergence $\tau_J \approx 7 \cdot 10^7$ iterations: a slow convergence to the optimum is exactly what we expected, since the condition number of R is pretty high ($\lambda_{max}/\lambda_{min} = 2.8 \cdot e^8$). The estimate w is then computed starting from a zero-valued initial array. The gradient vector is not explicitly visible in the loop, because we use its relation with p and R.

```
w = zeros(M, 1);
iterations = 2000 ;
for i = 1:iterations
    w = w + mu * (p − R * w);
end
```

Lastly, the theoretical minimum MSE J_{min}, the estimated one J_w and their ratio are computed. The following table shows the result that we obtained by testing the algorithm with 2000, 10000 and 100000 gradient updates.

#iterations	J_{min}	J_w	J_w / J_{min}
2000	0.001317896013452	0.001326901793865	1.00683345295946
10000	0.001317896013452	0.001319141601403	1.00094513371146
100000	0.001317896013452	0.001317976198578	1.00006084328723

2 Overlap-and-Add

In the second part of the homework it is required to apply the filter to the provided input signal. In particular here we want to focus on the implementation of the Overlap-and-Add (OLA) algorithm for real-time filtering in the frequency domain: we choose an *Hann* type tapered window and therefore to satisfy the COLA condition we set the *hop-size* to $w_{len}/2$ that corresponds to a 50% overlap (a 75% overlap with $hop = w_{len}/4$ is also a possible choice). Before filtering the input signal, we compute the total number of frames that will be processed from OLA and if needed we pad its size to be an exact multiple of the *hop-size*.

```
n_frames = ceil((length(x)−wlen)/hop + 1);
% pad x to match the total numbers of frames
last_frame_end = (n_frames+1)*hop;
x = padarray(x, [last_frame_end−N, 0], 'post');
```

Then the actual OLA loop follows: in each iteration, it performs the filtering in the frequency domain on the m_{th} windowed and padded frame (FFT), brings it back to the time domain (IFFT) and, finally, adds the result to the output filtered signal *y_hat_ola*, starting from the correct index $m \cdot hop + 1$.

3 Conclusions

We can compare the filter obtained in three different ways:

- **Ratio** - As we can see from the previous table the *MSE* cost function of the estimated filter J_w gets closer to the theoretical limit J_{min} ($ratio = 1$) as the number of iterations increases: although convergence is expected after τ_J iterations, we can say that with 10000 cycles we already get a great result with a 10^{-5} numeric precision.
- **Playback** - By playing back the filtered signal we observed that the difference with the real output signal y can actually be heard only with speakers or headphones with a good frequency response: with this devices the filtered signal results a little bit darker in frequency, a kind of low-passed version of the real one.
- **Visual** - Thanks to the provided real RIR, it is possible to carry out a visual comparison as well. As you can see below, the plots confirm what we stated in the previous points.