SASP-DAAP Homework 1

PERRIN Paloma (ID 10873558) and MORO Alessandra (ID 10620673)

20 March 2022

This report is aimed to explain the main reasoning behind the implementation of the matlab code that estimates the Room Impulse Response (RIR) of a small reverberant environment.

1 Wiener Filters

First of all, we wanted to compute the Wiener-Hopf solution (1) in order to have a functional Wiener filter.

$$R.w_0 = p \Leftrightarrow w_0 = R^{-1} * p \tag{1}$$

To do this, we had to define all the parameters of the Wiener-Hopf equations which are:

- The autocorrelation matrix of the input signal: we computed it for any positive lags in order to have a matrix diagonal with only positive values, then we reduced it to a 4000 samples vector length and we used this vector to compute the Toeplitz matrix.
- The cross-correlation matrix between the input signal and the desired response : this time, we computed it for all negative lags to respect the formula and we reduced its length to M.

Once we did that, we started to compute the Steepest Descent Algorithm. To do that we had to respect certain properties:

- To respect the necessary and sufficient condition for stability, the step size parameter has to respect:

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{2}$$

- To respect the necessary and sufficient condition for convergence, the global time constant parameter has to respect:

$$\tau_k = \frac{1}{\mu.\lambda_{min}} \tag{3}$$

To apply those conditions we had to compute the Eigenvalue of the autocorrelation matrix R with the matlab function *eig(R)*.We then computed the maximum and minimum of this eigenvalue matrix.
We chose the value of the step size parameter between the defined range that ensured the best cost functions ratio. To compute the value of the MSE function we had to perform the iterative update of the filter taps for 2000 gradients. Then we computed the formula of the MSE function.
By respecting these conditions we obtained a ratio of 1.0064. We plotted the true RIR, the wiener filtered signal and the steepest descent filtered signal using a time vector defined in terms of the sample frequency in order to plot the amplitude being a function of time.

2 Overlap and Add

We performed the time-domain filtering by computing the convolution of the input signal with the filter taps vector.

To filter in frequency-domain, we had to zero-pad the input signal and the filter taps vector to the power of two closest to the length: $N_y = N_x + N_w - 1 = 18436$ which was 2^{15}. We then computed the FFT of both the input signal and the filter taps vector and multiplied them to perform the filtering. To transform back to the time domain, we used the IFFT.

Finally, we implemented the Overlap and Add algorithm by defining the : window length, hop size, tapered triangle window and frame number. With a loop, we first defined the number of iterations of the OLA. Then, the aim was to separate the filtered input signal into frames with the help of input indexes in order to, then, compute the FFT of all those samples. Finally, by defining output indexes aiming to reconstruct the signal in the time domain, we computed the IFFT of the triangle filtered signal, filtering it at the same time with the wiener filter. The loop then allowed us to reconstruct the signal using the output indexes. We plotted all the different filtered signals.