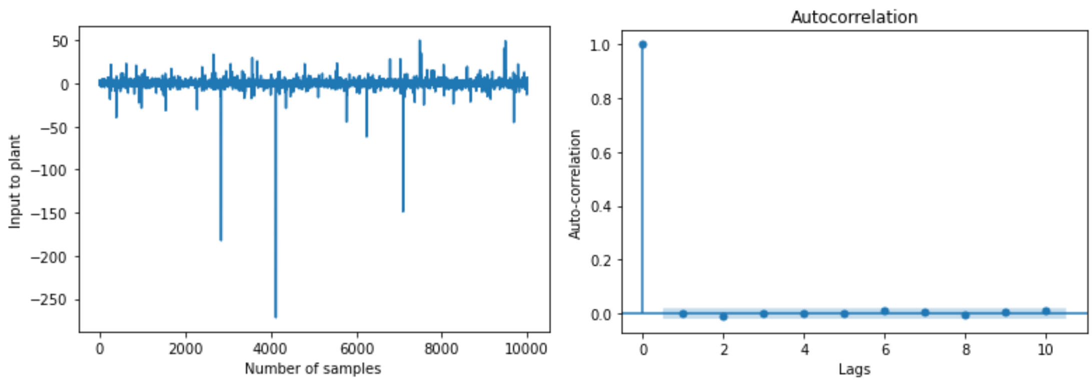
Homework 2

Kaifa Lu

Solution to Problem 1:

Step 1: **Input**. The input to the plant is alpha stable noise with $\alpha = 1.5$. Based on the characteristic function $\emptyset(t) = \exp(-|t|^{1.5})$, we can first generate 10,000 samples of the alpha stable noise using *filter_x = levy_stable.rvs(alpha = 1.5, beta = 0, size = 10000)* as the input $x(n)$ to the plant. Also, we plotted the autocorrelation to confirm the input to the plant.



(a) Plot of 10,000 samples (b) Autocorrelation

Fig. 1 Plots of input to the plant

Step 2: **Transfer function**. Simplify transfer function to establish the relationship between the input $x(n)$ and output $y(n)$:

$$H(z) = \frac{1 - z^{-10}}{1 - z^{-1}} = \sum_{i=0}^{9} z^{-i}$$

$$y(n) = x(n) \sum_{i=0}^{9} z^{-i}$$

$$y(n) = \sum_{i=0}^{9} x(n-i)$$

Therefore, the plant is a FIR system.

Step 3: **Noisy output**. The noisy output of the plant is obtained through adding the Gaussian noise of power $N = 0.1$ to the output of transfer function as follows:

$$y(n) = \sum_{i=0}^{9} x(n-i) + Gaussian\ Noise$$

Then we plotted Power Spectrum Density (PSD) to confirm the output with Gaussian Noise to the plant.

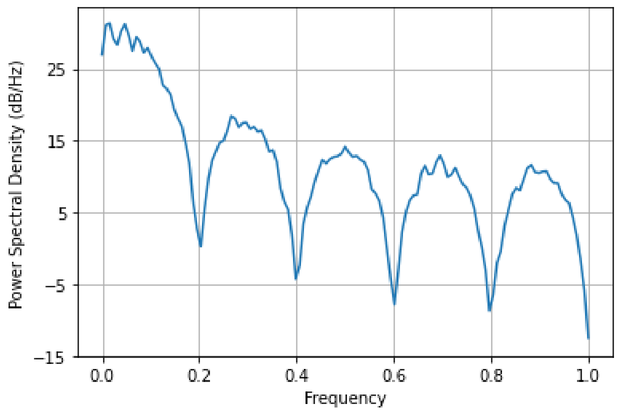


Fig. 2 Power Spectrum Density of the output with Gaussian Noise to the plant

Step 4: **RLS algorithm**. Once we have obtained the input and noisy output of the plant, we designed a linear filter updated with the RLS algorithm to identify the unknown plant transfer function. To evaluate the performance of the linear filter with RLS, (1) we computed the weighted error power $WSNR = 10\log\left(\frac{W^{*T}W^{*}}{(W^{*}-W(n))^{T}(W^{*}-W(n))}\right)$ and (2) plotted the evolution of updated weight vectors and normalized MSE regarding each new samples. Where $W^{*} = [1,1,1,1,1,1,1,1,1,1]^{T}$

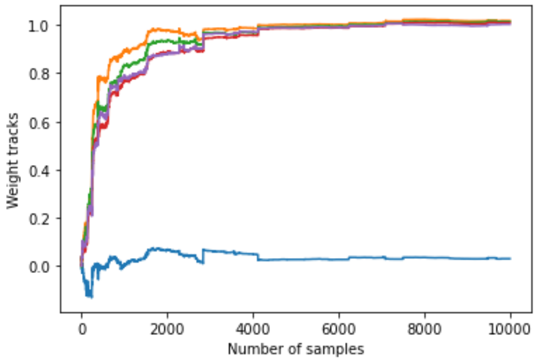
i. <u>Vary filter of order from 5, 15 to 30</u>

1) **Scenario 1**: filter of order = 5

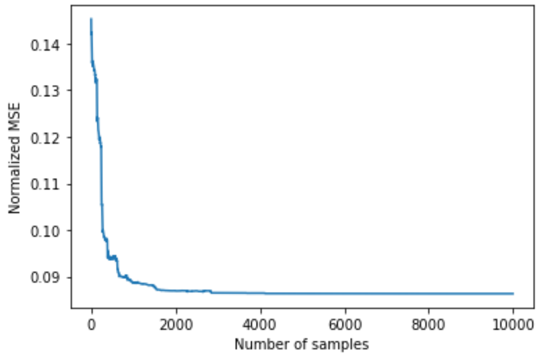
$$W(5) = [0.03,1.02,1.01,1.01,1.00]^{T}$$

$$W^{*}\_sized = [1,1,1,1,1]^{T}$$

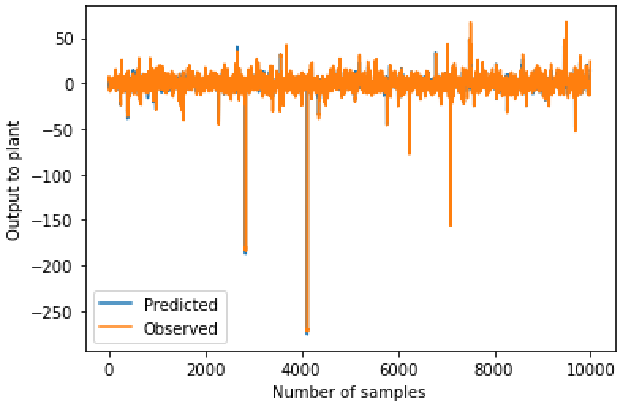
$$WSNR = 10\log 10\left(\frac{W^{*T}W^{*}}{(W^{*}-W(n))^{T}(W^{*}-W(n))}\right) = 7.2549$$



(1) Weight tracks



(2) Normalized MSE



(3) Comparison between the observed and predicted values

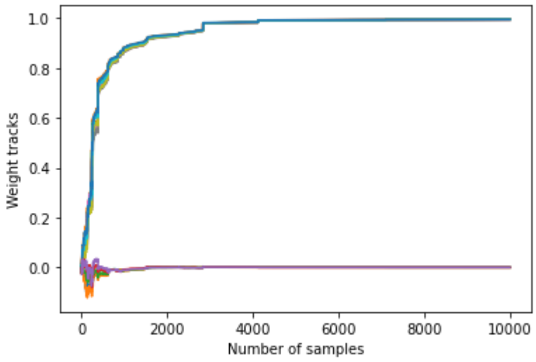
Fig. 3 The performances of the linear filter with RLS algorithm

2) **Scenario 2**: filter of order = 15

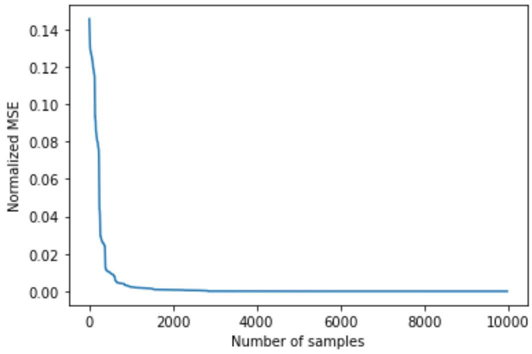
$$W(15) = \begin{bmatrix} 0.001,0.995,0.995,0.995,0.995,0.995,0.995,0.996,0.996,0.995, \\ 0.995,0.000,0.000,0.000,0.000 \end{bmatrix}^{T}$$

$$W^{*}\_sized = [1,1,1,1,1,1,1,1,1,1,0,0,0,0,0]^{T}$$

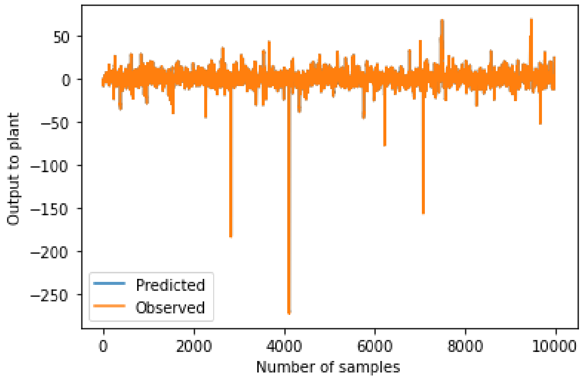
$$WSNR = 10\log 10\left(\frac{W^{*T}W^{*}}{(W^{*}-W(n))^{T}(W^{*}-W(n))}\right) = 7.0121$$



(1) Weight tracks



(2) Normalized MSE



(3) Comparison between the observed and predicted values

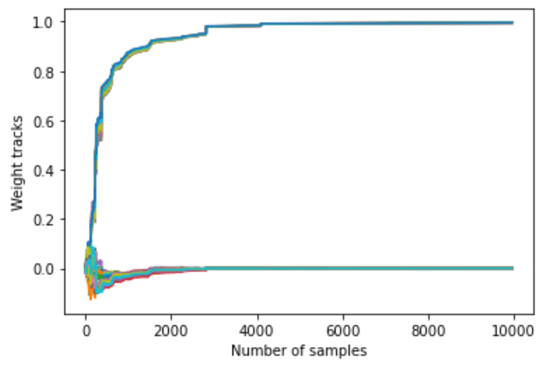
Fig. 4 The performances of the linear filter with RLS algorithm

3) **Scenario 3**: filter of order = 30

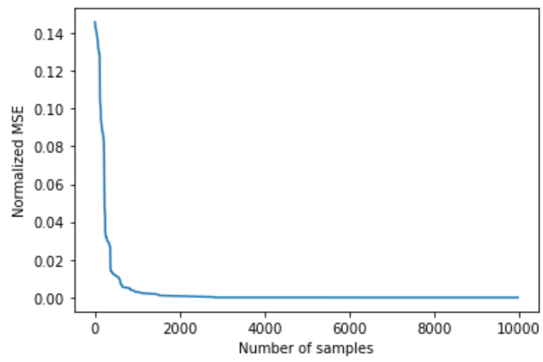
$$W(30) = \begin{bmatrix} 0.001,0.995,0.995,0.995,0.995,0.995,0.995,0.996,0.996,0.995,0.995 \\ 0.000,0.000,0.000,0.000,0.000,0.000,0.000,-0.000,0.000,0.000 \\ -0.000,-0.000,-0.001,-0.000,0.000,-0.000,0.000,0.001,0.000 \end{bmatrix}^T$$

$$W^{*}\_sized = [1,1,1,1,1,1,1,1,1,1,0,\cdots,0]^T$$

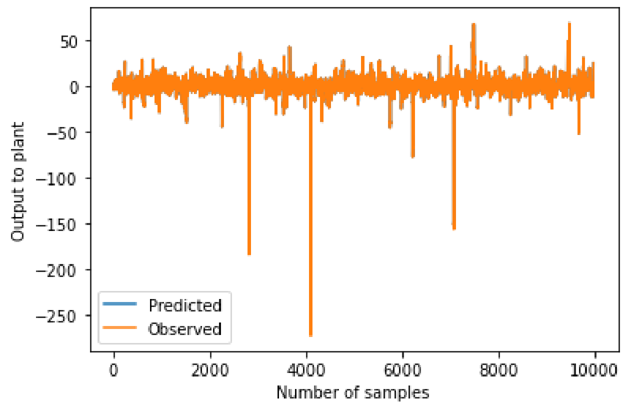
$$WSNR = 10\log 10\left(\frac{W^{*T}W^{*}}{(W^{*}-W(n))^T(W^{*}-W(n))}\right) = 7.0121$$



(1) Weight tracks



(2) Normalized MSE



(3) Comparison between the observed and predicted values

Fig. 5 The performances of the linear filter with RLS algorithm

<u>Main findings regarding different orders of filter from 5, 15 to 30</u>:

It is easily found from the transfer function that the output of the unknown plant at the time (n) is strongly associated with the inputs at the time $(n-9,\cdots,n)$. Therefore, the linear filter of order 5 presents larger errors between the filter output and the desired values than that of order 15 and 30. However, increasing the filter of order from 15 to 30 hardly improves the performances of the linear filter. It is worth noting that when the order of the filter is higher than 10, the weight vector shows faster convergence speed and more consistent weight tracking curves among different weight components. The similar changing trend of weight tracks can also be found for the normalized MSE.

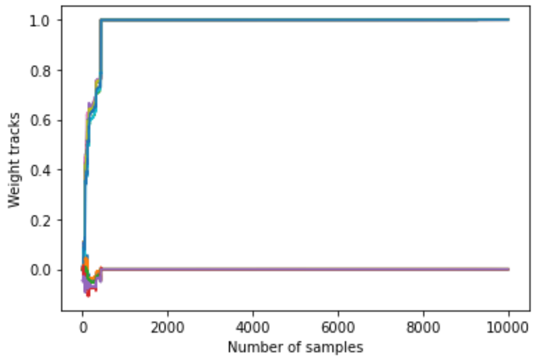
ii. Increase the noise N = 0.3, 1.5

1) **Scenario 1**: Noise N = 0.3, filter of order = 15

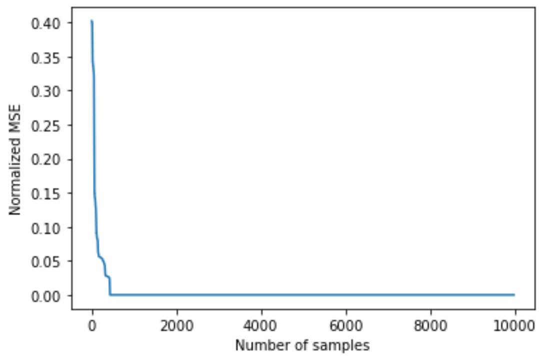
$$W(15) = \begin{bmatrix} -0.000,0.999,0.999,0.999,0.999,0.999,0.999,1.000,1.000,0.999, \\ 0.999,-0.000,0.000,-0.000,-0.000 \end{bmatrix}^T$$

$$W^*\_sized = [1,1,1,1,1,1,1,1,1,1,0,0,0,0,0]^T$$

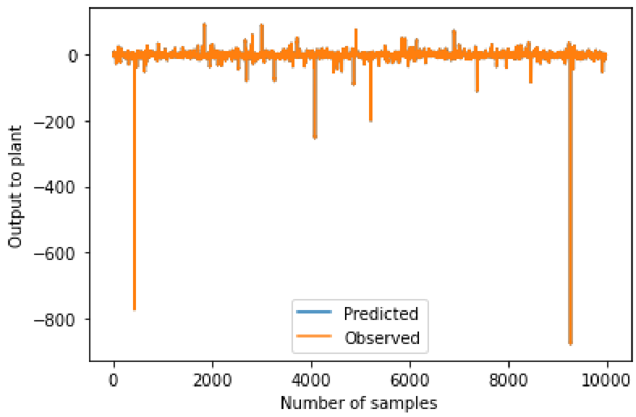
$$WSNR = 10\log10\left(\frac{W^{*T}W^*}{(W^* - W(n))^T(W^* - W(n))}\right) = 6.9906$$



(1) Weight tracks



(2) Normalized MSE



(3) Comparison between the observed and predicted values

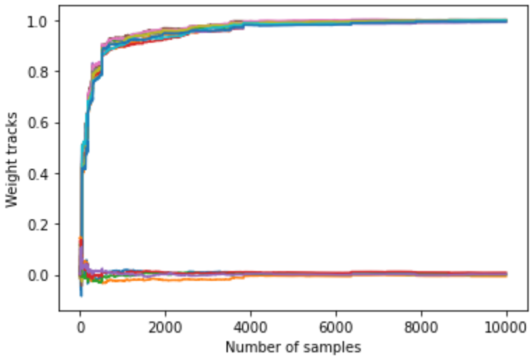
Fig. 6 The performances of the linear filter with RLS algorithm

2) **Scenario 2**: Noise N = 1.5, filter of order = 15

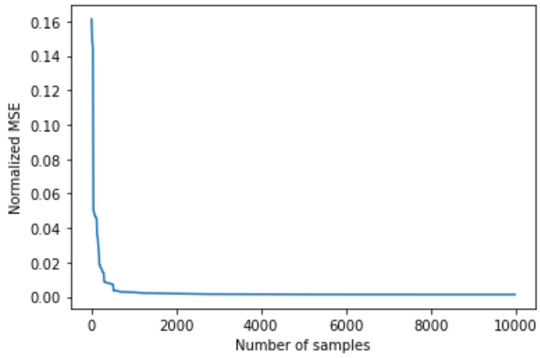
$$W(15) = \begin{bmatrix} -0.002,0.993,0.998,0.998,0.992,1.002,1.000,0.998,1.000,1.000, \\ 0.994,-0.007,0.000,0.008,0.001 \end{bmatrix}^T$$

$$W^*\_sized = [1,1,1,1,1,1,1,1,1,1,0,0,0,0,0]^T$$

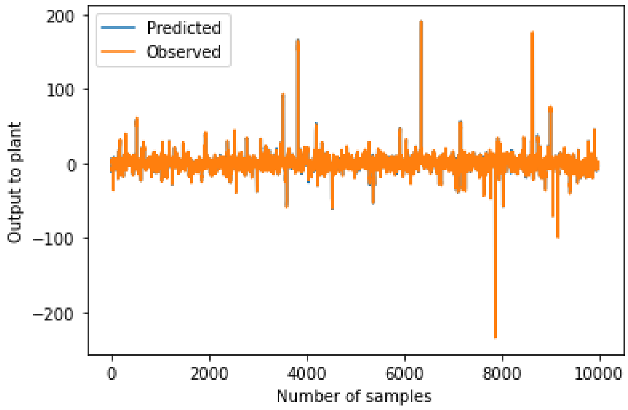
$$WSNR = 10\log 10\left(\frac{W^{*T}W^{*}}{(W^{*}-W(n))^{T}(W^{*}-W(n))}\right) = 7.0220$$



(1) Weight tracks



(2) Normalized MSE



(3) Comparison between the observed and predicted values

Fig. 7 The performances of the linear filter with RLS algorithm

Main findings regarding different noises rising from 0.1, 0.3 to 1.5:

Increasing the noise N from 0.1, 0.3 to 1.5 hardly affects the performances of the linear filter with the RLS algorithm. This can be reflected by the phenomena that the changing curves of weight vectors and the normalized MSE almost present the similar trend, whether the noise N is set to 0.1, 0.3 or 1.5. Additionally, when the noise N increases from 0.1, 0.3 to 1.5, the observed and predicted outputs of the plant all demonstrate a high consistency, further indicating that the RLS algorithm is very stable in fitting the transfer function of the unknown plant.

Step 5: **Comparison**. A follow-up question is to compare the performance of the linear filter with the RLS algorithm with the Wiener solution and the LMS employed in HW1. Specifically, for each scenario, we compared the weighted error power $WSNR$, the updated weight vectors and normalized MSE, as well as their convergence speed.

(1) The linear filters using whether Wiener solution, the LMS or RLS algorithms obtain almost the same optimal weight vectors and weighted error power $WSNR$, approximately equaling to 7.

(2) The Wiener solution requires relatively smaller number/size of samples than the LMS and RLS algorithms to achieve approximately identical performances. This means that more samples are needed for the linear filters when using the LMS and RLS algorithms.

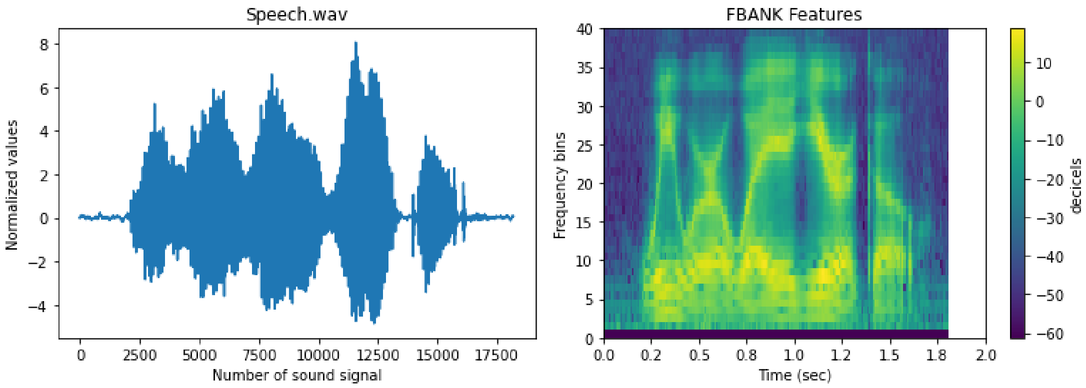
(3) The performances of the linear filter using the RLS algorithm are more stable that that using the LMS algorithm, in terms of weights tracks and normalized MSE. In other words, the changing curves of weight tracks and normalized MSE are less fluctuated and smoother for the RLS algorithm than the LMS algorithm. Furthermore, the optimal weight vector of the filter using the RLS algorithm presents faster convergence speed than that using the LMS algorithm.

(4) The performances of the LMS adaptive filter are greatly affected by the learning rate or step size. This means that only when the learning rate is suitably selected

can the LMS adaptive filter bring a good prediction performance. However, the RLS algorithm would not encounter this situation, thus being more stable than the LMS algorithm.

Solution to Problem 2

Step 1: Load the sound signal file *speech.wav* and normalize the data by the power of the input. Then we plot the normalized input signal and sound spectrogram, as shown in Fig. 1.



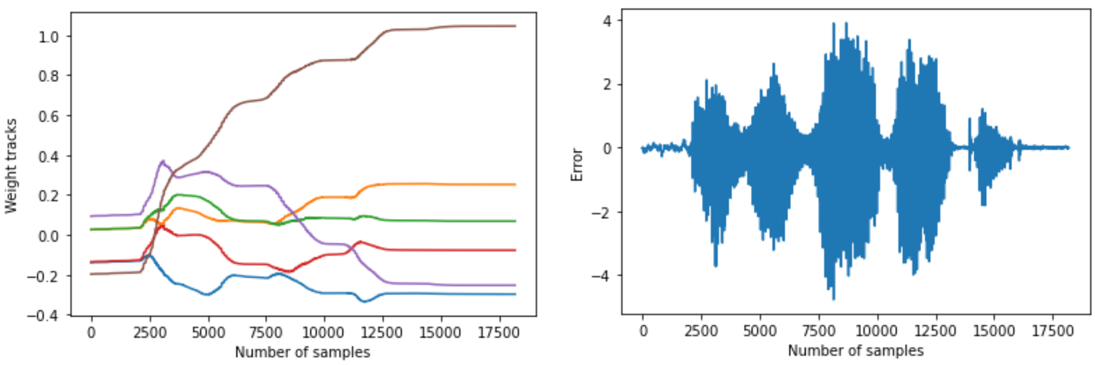
(1) Plots of sound signal (2) Spectrogram

Fig. 1 Illustration of the utterance data

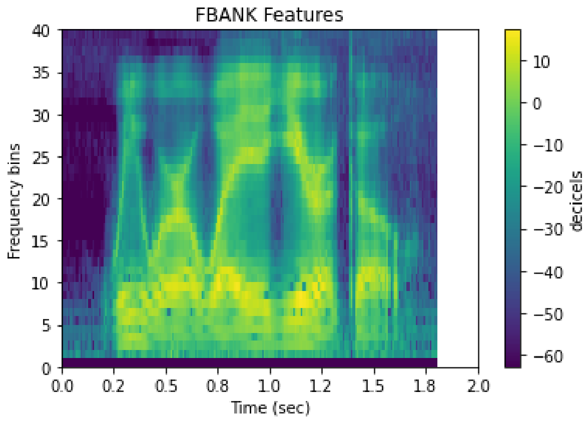
Step 2: **RLS algorithm**. Import the utterance data into the linear filter with RLS for denoising under different filter lengths and forgetting factors to compare the quality of prediction.

i. Vary filter of order from 6 to 15

1) **Scenario 1**: filter of order = 6 and forgetting factor = 1



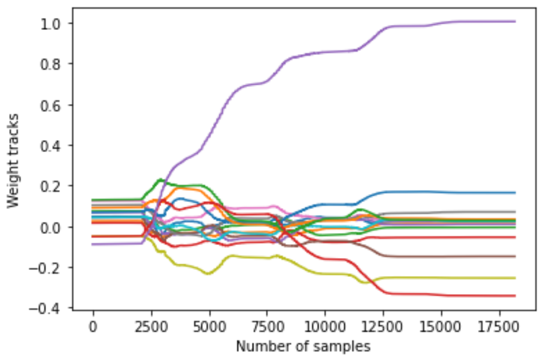
(a) Weight tracks (b) Error



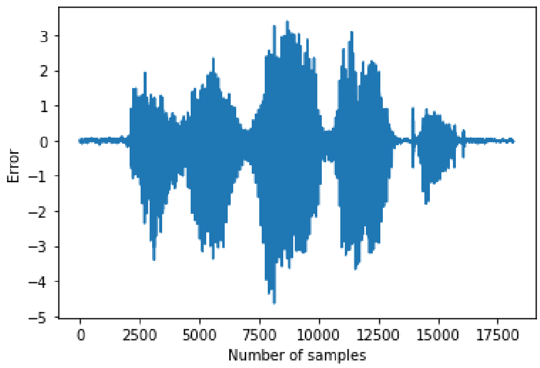
(c) Spectrogram

Fig. 2 Changes in the filter parameters over time and filtering outputs of the signal

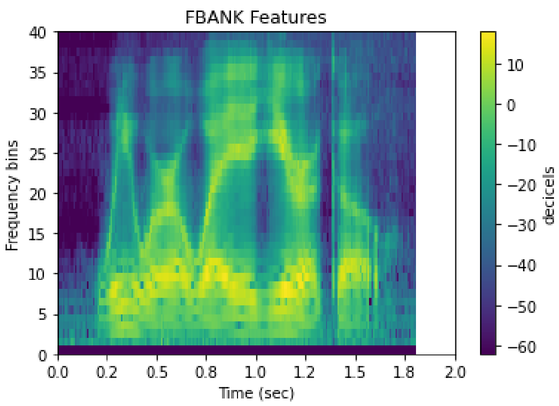
2) **Scenario 2**: filter of order = 15 and forgetting factor = 1



(a) Weight tracks



(b) Error



(c) Spectrogram

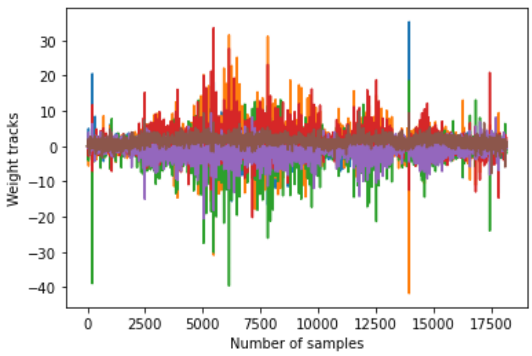
Fig. 3 Changes in the filter parameters over time and filtering outputs of the signal

<u>Major findings regarding different filter lengths:</u>

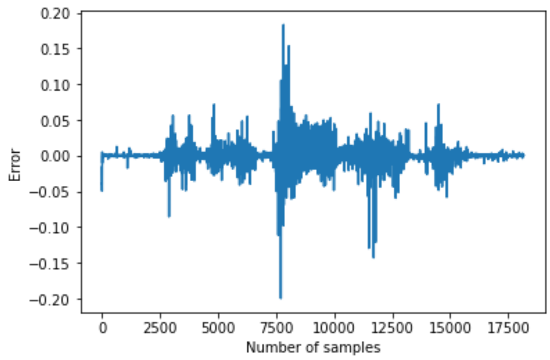
Increasing the filter length from 6 to 15 slightly improves the performances of the linear filter, accompanied by lower errors over all samples under the filter of order 15. The changing curves of weight tracks and errors and the denoised spectrogram present the similar trend under different orders (6 or 15) of the linear filter. It is worth noting that some weight components under the filter of order (15) eventually converge to zero, which means that some weight components hardly contribute to the linear filter.

ii. <u>Vary the forgetting factor from 0.5, 0.7, 0.9 to 0.99</u>

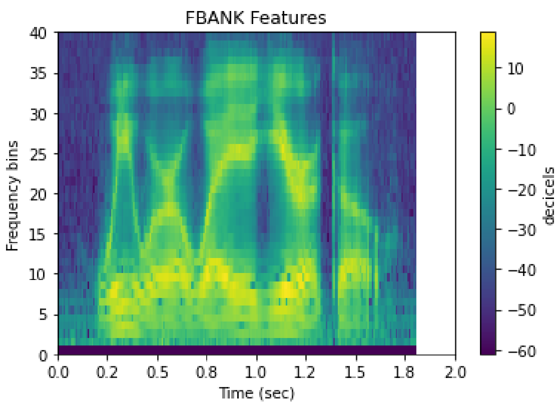
1) **Scenario 1**: filter of order = 6 and forgetting factor = 0.5



(a) Weight tracks



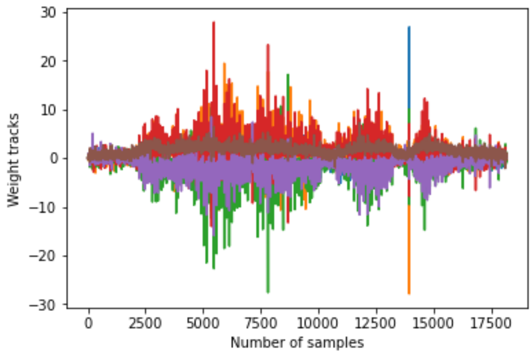
(b) Error



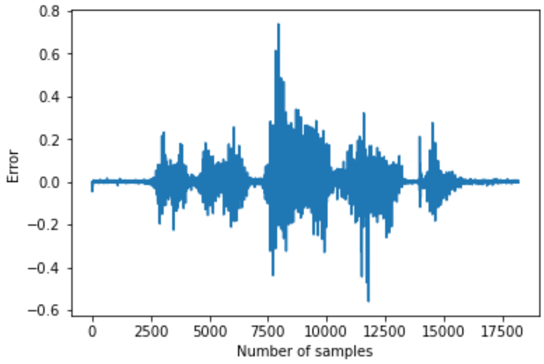
(c) Spectrogram

Fig. 4 Changes in the filter parameters over time and filtering outputs of the signal

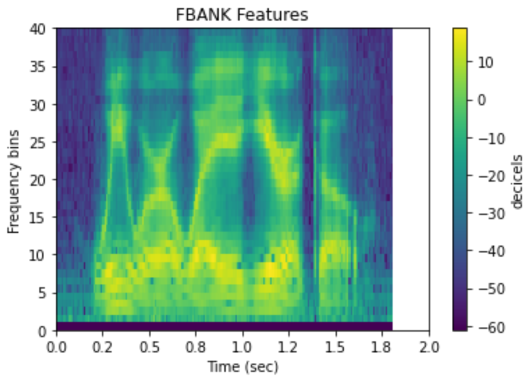
2) **Scenario 2**: filter of order = 6 and forgetting factor = 0.7



(a) Weight tracks



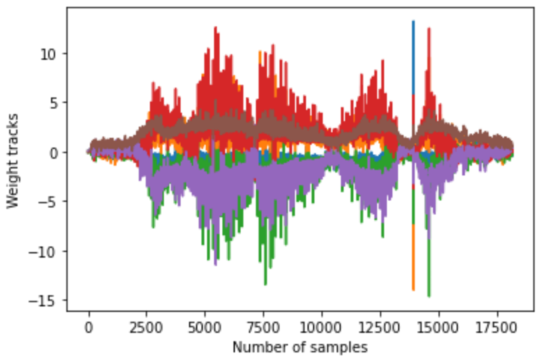
(b) Error



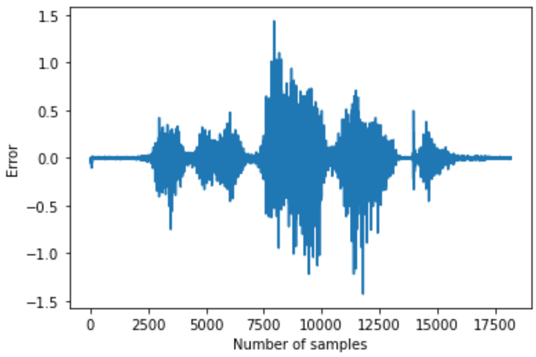
(c) Spectrogram

Fig. 5 Changes in the filter parameters over time and filtering outputs of the signal

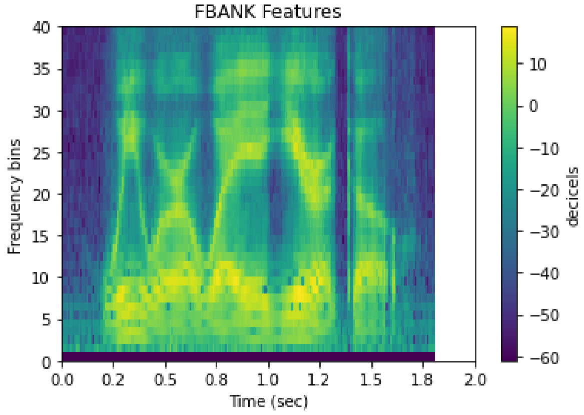
3) **Scenario 3**: filter of order = 6 and forgetting factor = 0.9



(a) Weight tracks



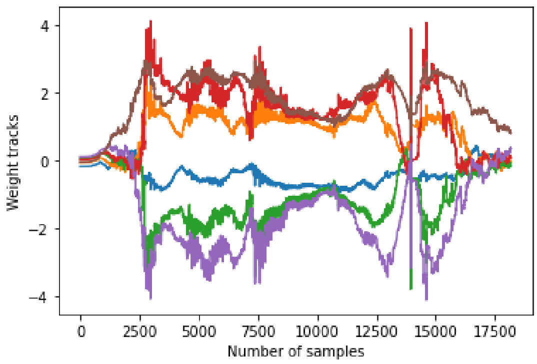
(b) Error



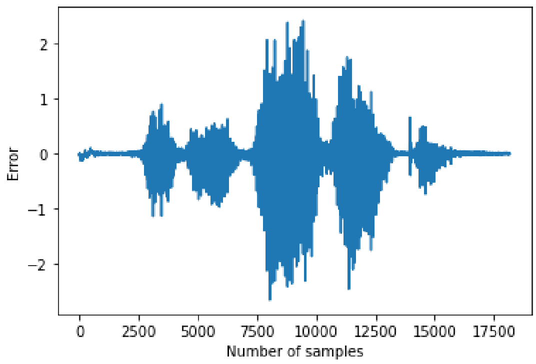
(c) Spectrogram

Fig. 6 Changes in the filter parameters over time and filtering outputs of the signal

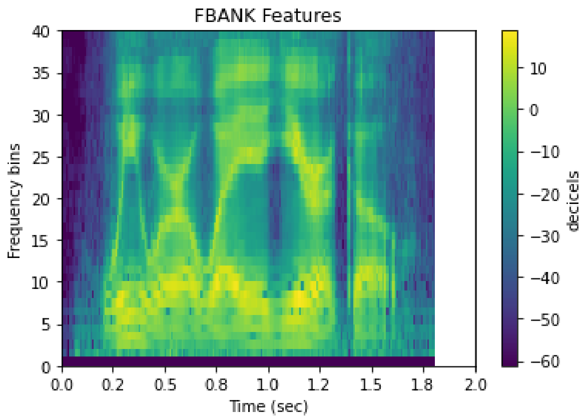
4) **Scenario 3**: filter of order = 6 and forgetting factor = 0.99



(a) Weight tracks



(b) Error



(c) Spectrogram

Fig. 7 Changes in the filter parameters over time and filtering outputs of the signal

Major findings regarding different forgetting factors:

When using different forgetting factors from 0.5, 0.7, 0.9 to 0.99 to filter the sound signal, lower forgetting factors are usually accompanied by lower errors between the original and filtered signals, while higher forgetting factors exhibit smoother weight tracks curves. This further indicates that the linear filter using the RLS algorithm with higher forgetting factors shows higher stability in filtering the sound signal, which is less affected by unexpected noises or samples. Therefore, based on the experiments, we can identify that the forgetting factor (0.9) outperforms other forgetting factors.

Step 3: **Examine how the filter parameters change over time**.

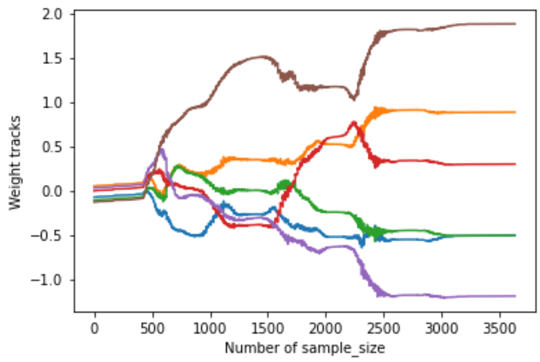
(1) **Weight tracks**: As shown in Fig. 2-3, when the forgetting factor = 1, the weight vector was updated by a very small step over each new sample, which means that the weight tracks curve is relatively smooth, and gradually converges to the optimal weight vector until the 12,500th samples. As illustrated in Fig. 4-7, when the forgetting factor is less than 1, the changing curve of weight tracks presents a high similarity with the variations of the sound signal. This indicates that the update of the weight vector is largely affected by the sound signal itself.

(2) **Error between the original and filtered signal**. Similarly, the changes in errors between the original and filtered sound signals over each new sample exhibit almost the same fluctuating trend as the sound signal itself. This means that the samples fluctuating abruptly are still difficult to be filtered.

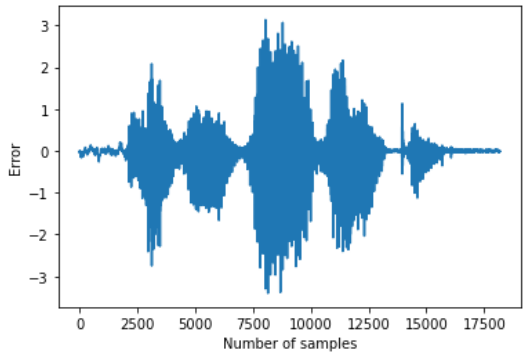
Step 4: **APA 1 – Using the least square error and gradient descent method**. Import the utterance data into the APA 1 filter for denoising under different filter lengths and number of samples to repeat the above process and compare the quality of APA 1 predictor in this sound signal.

i. Vary filter of order from 6 to 15

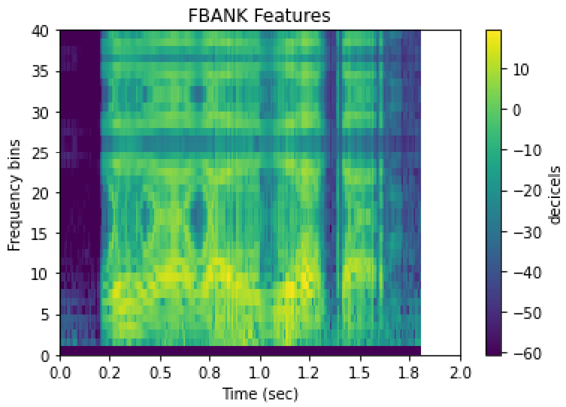
1) **Scenario 1**: filter of order = 6, step size = 0.005, and sample size = 5



(a) Weight tracks



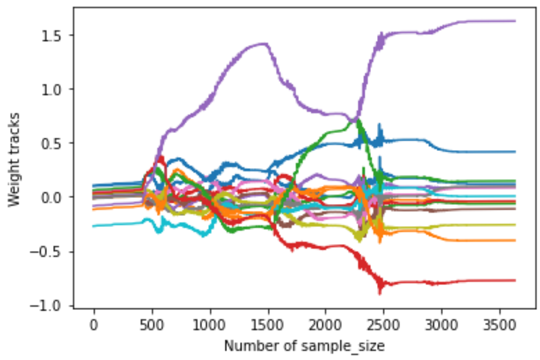
(b) Error



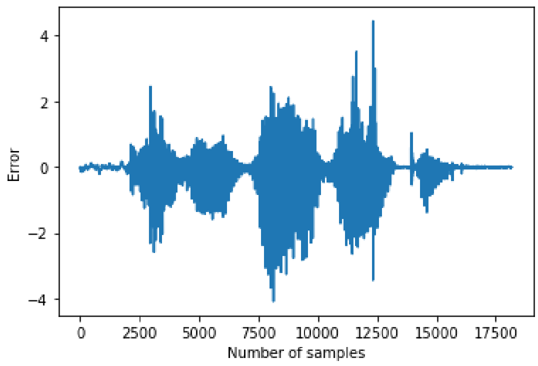
(c) Spectrogram

Fig. 8 Changes in the filter parameters over time and filtering outputs of the signal

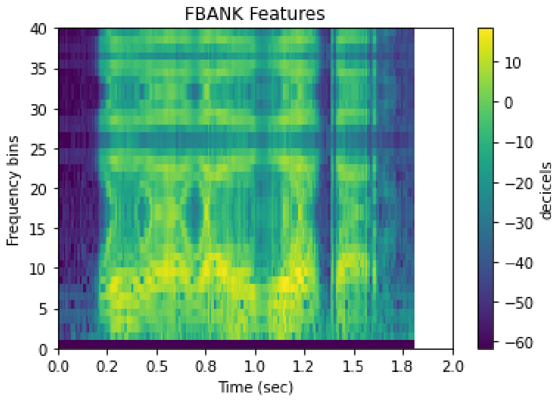
2) **Scenario 2**: filter of order = 15, step size = 0.05, and sample size = 5



(a) Weight tracks



(b) Error

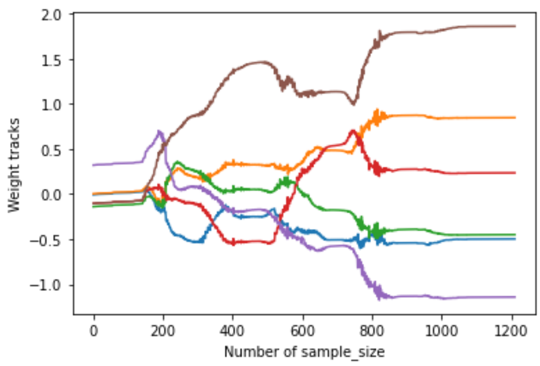


(c) Spectrogram

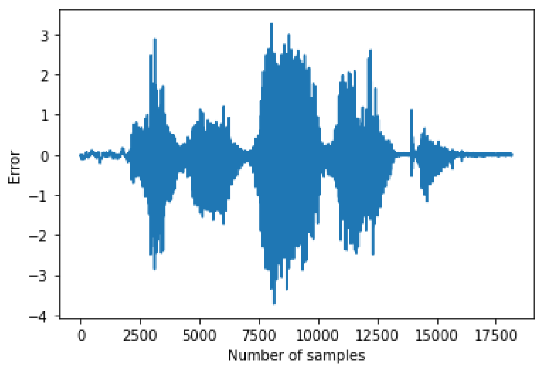
Fig. 9 Changes in the filter parameters over time and filtering outputs of the signal

ii. <u>Vary the number of samples from 5, 15 to 30</u>

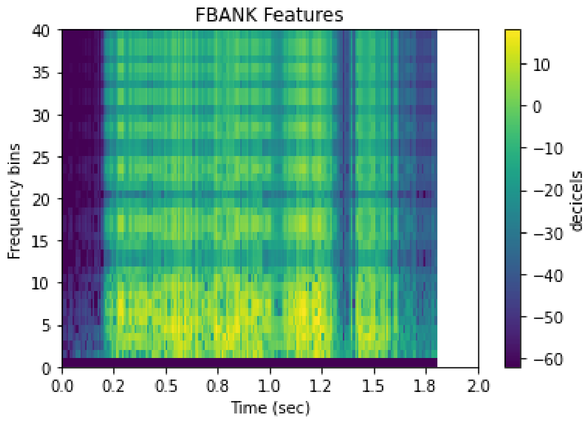
1) **Scenario 1**: filter of order = 5, step size = 0.05, and sample size = 15



(a) Weight tracks



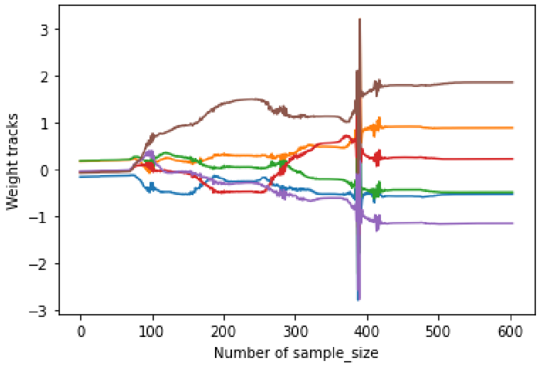
(b) Error



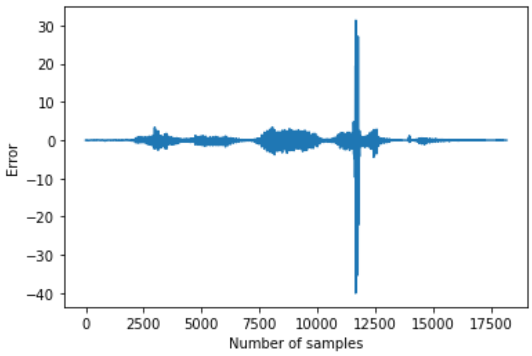
(c) Spectrogram

Fig. 10 Changes in the filter parameters over time and filtering outputs of the signal

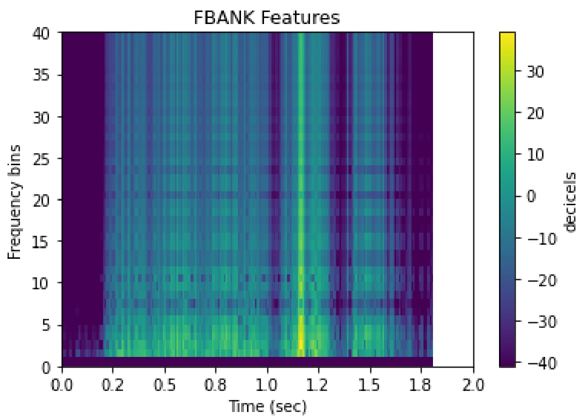
2) **Scenario 2**: filter of order = 5, step size = 0.05, and sample size = 30



(a) Weight tracks



(b) Error



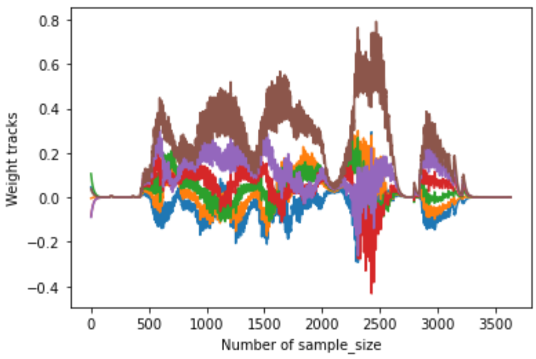
(c) Spectrogram

Fig. 11 Changes in the filter parameters over time and filtering outputs of the signal

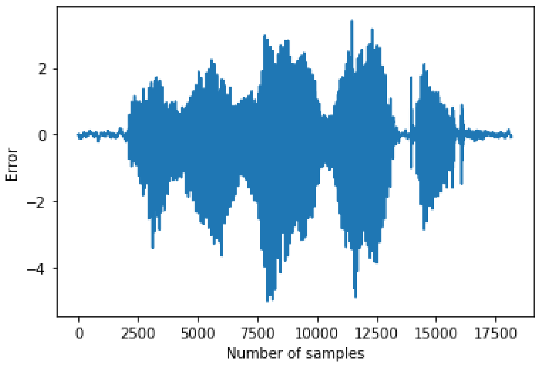
Step 5: **APA 3 – Using the least square error with regularization and gradient descent method**. Import the utterance data into the APA 3 filter for denoising under different filter lengths and number of samples to repeat the above process and compare the quality of APA 3 predictor in this sound signal.

i. Vary filter of order from 6 to 15

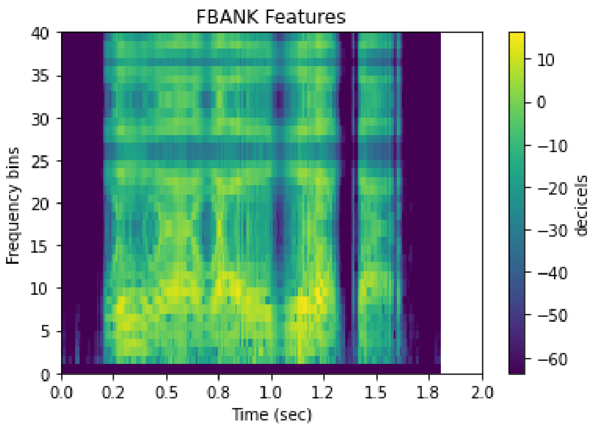
1) **Scenario 1**: filter of order = 6, step size = 0.05, lambda = 10, sample size = 5



(a) Weight tracks



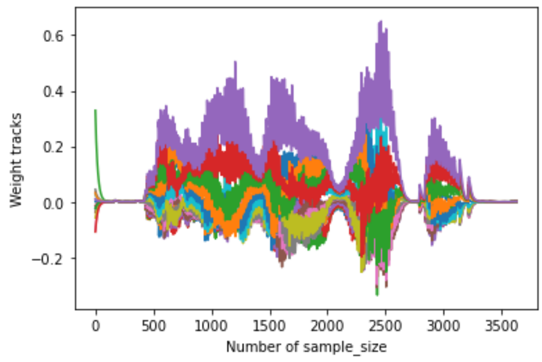
(b) Error



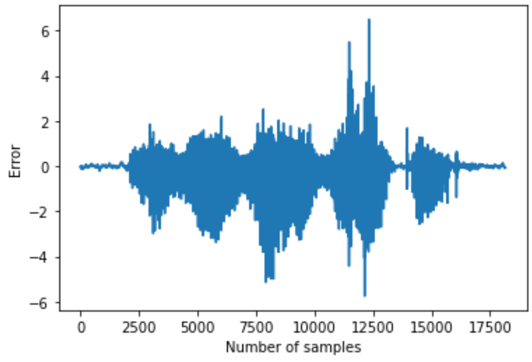
(c) Spectrogram

Fig. 12 Changes in the filter parameters over time and filtering outputs of the signal

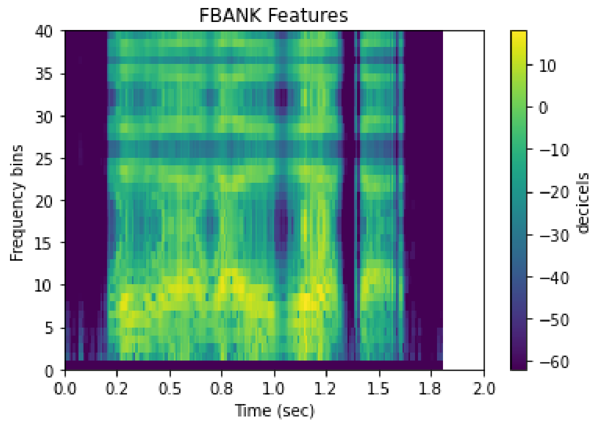
2) **Scenario 2**: filter of order = 15, step size = 0.05, lambda = 10, sample size = 5



(a) Weight tracks



(b) Error

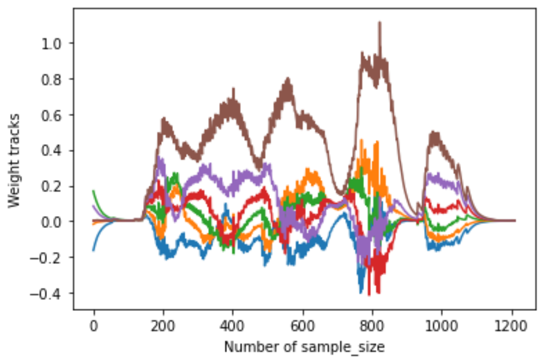


(c) Spectrogram

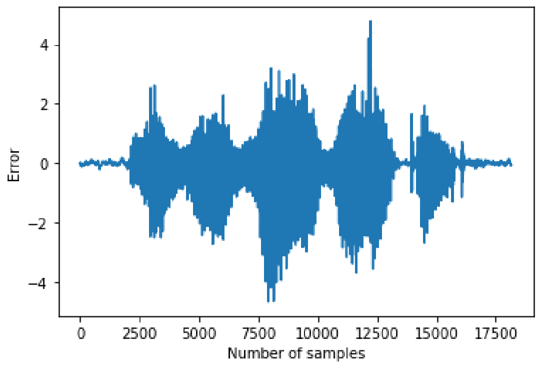
Fig. 13 Changes in the filter parameters over time and filtering outputs of the signal

ii. <u>Vary the number of samples from 5, 15 to 30</u>

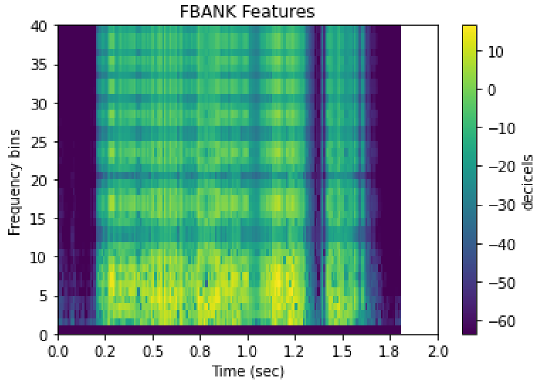
1) **Scenario 1**: filter of order = 5, step size = 0.05, lambda = 10, sample size = 15



(a) Weight tracks



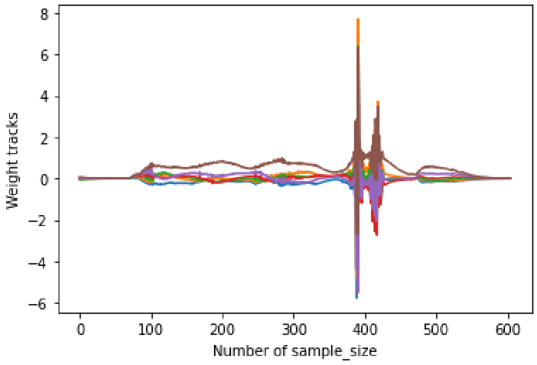
(b) Error



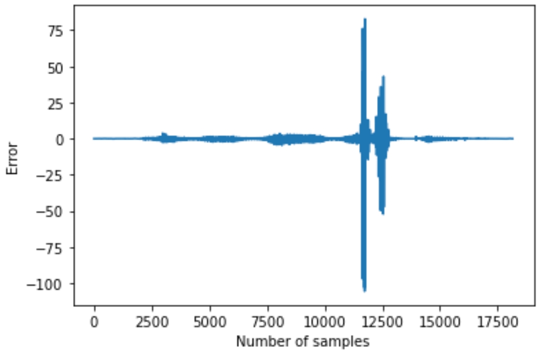
(c) Spectrogram

Fig. 14 Changes in the filter parameters over time and filtering outputs of the signal

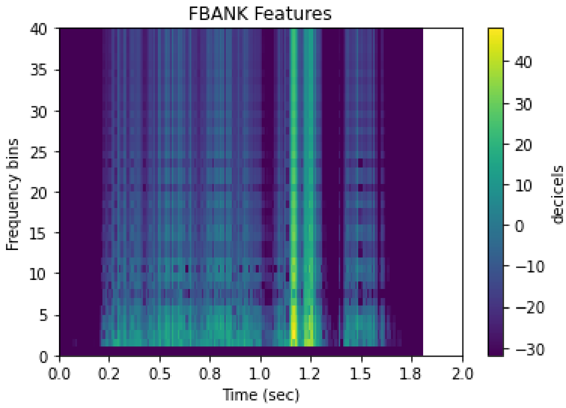
2) **Scenario 2**: filter of order = 5, step size = 0.05, lambda = 10, sample size = 30



(a) Weight tracks



(b) Error



(c) Spectrogram

Fig. 15 Changes in the filter parameters over time and filtering outputs of the signal

Major findings regarding different APA models and effects of the number of samples:

(1) **Selection of APA1 and APA3 models**. Considering that the RLS algorithm only uses one new sample for each time and the APA models can use more than one samples to estimate the weight vector of the linear filter, APAs appear as intermediate complexity/performance algorithms between the LMS and RLS algorithms. Here, we select two APA models: APA1 using the least square error and gradient descent method, and APA3 using the least square error with regularization and gradient descent method. The two APA models could be efficiently used to estimate the optimal weight vector of the linear filter over multiple samples for each update.

(2) **Effects of filter lengths on APA models**. Increasing the filter length from 6 to 15 can improve the performances of the linear filter, albeit not significantly. The changing curves of weight tracks, errors and the denoised spectrogram present the similar trend under different orders (6 or 15) of the linear filter. It is worth noting that the changing curves of the APA1 model are smoother than those of the APA3 model, and the latter is easily affected by the sound signal itself.

(3) **Effects of the number of samples on APA models**. Increasing the number of samples from 5, 15 to 30 makes the performances of filter worse. This means that when the number of sample (sample size) is equal to 5 for each update of weight vector, the linear filter using the APA models achieve the best prediction performances than that with other sample sizes.

(4) **Comparison between RLS and APA models**. The APA models are introduced

to improve the convergence rate of the adaptive filtering algorithms than LMS, and at the same time, the APA models show less complexities than the RLS algorithm. However, it suffers from an inherent drawback that imposes a trade-off between high initial convergence rate and low steady-state error or weight tracks, particularly for the APA3 model.