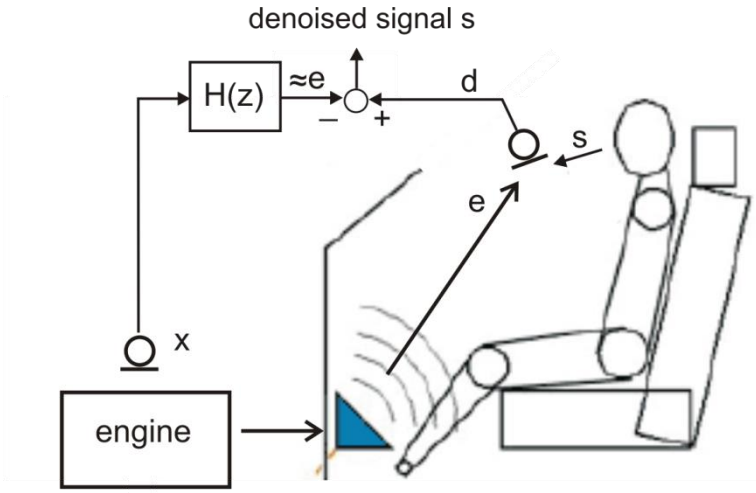
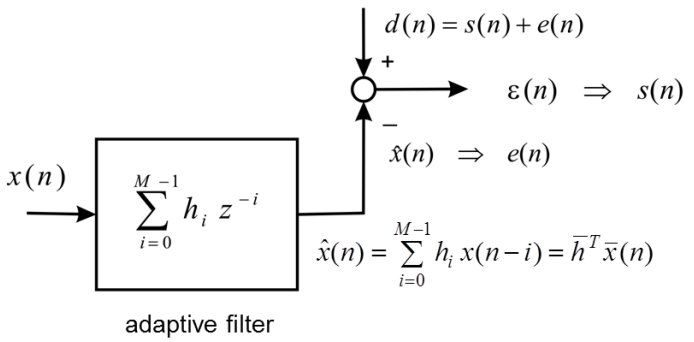
LABORATORY 3: ADAPTIVE FILTERING:

1. DENOISING





$s(n)$ - speech

$e(n)$ - noise

$d(n)$ - signal with noise (microphone 1)

$x(n)$ - noise recorded with the second microphone, correlated with $e(n)$, not correlated with $s(n)$

$\hat{x}(n)$ - filtered noise

$\varepsilon(n) = s(n) + e(n) - \hat{x}(n) \approx s(n)$ - signal with noise suppressed

Searched: impulse response of the adaptive filter: $\bar{h} = [h_0, h_1, ..., h_{M-1}]^T$ (*T*-transposition)

Criterion: minimum energy of the error signal: $\sum_n \varepsilon^2(n)$.

Solution: LMS (least mean square) algorithm: $\bar{h}(n+1) = \bar{h}(n) + \beta\,\varepsilon(n) \begin{pmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-M+1) \end{pmatrix}$

where β - filter adaptation speed, *n*-number of current sample..

Simulations:

Run MRCInstaller (if you have not done this before). This is a program delivered by Mathworks making possible to execute compiled Matlab code. Run it only once, it will stay in your computer. This operation is permitted by Mathworks, a company distributing Matlab.

Run adaptive.exe

Select the option `denoising`. Then select the name of audio file. The following wav files sampled at 44100 Hz are available: WIDEBAND_SPEECH, SONG, PIANO or VIOLIN. In this way you define the signal s(n).

The noise signal has been recorded simultaneously using two microphones:
Microphone 1: signal e(n)
Microphone 2: signal x(n).
Enter the signal-to-noise ratio in decibels: **SNR_in.** Signal e(n) will be amplified or attenuated so as to keep the required SNR value. E.g. SNR_in=0 dB yields the same power of signal s(n) and e(n). The input signal d(n) is a sum of s(n) and e(n).
Enter the adaptation speed β (see the formula describing the LMS algorithm).
Observe evolution of the impulse response of the adaptive filter (the impulse response contains M=500 samples).
At the end of simulation note the signal-to-noise ratio after denoising **SNR_out_dB**.
Display and listen to the following signals:

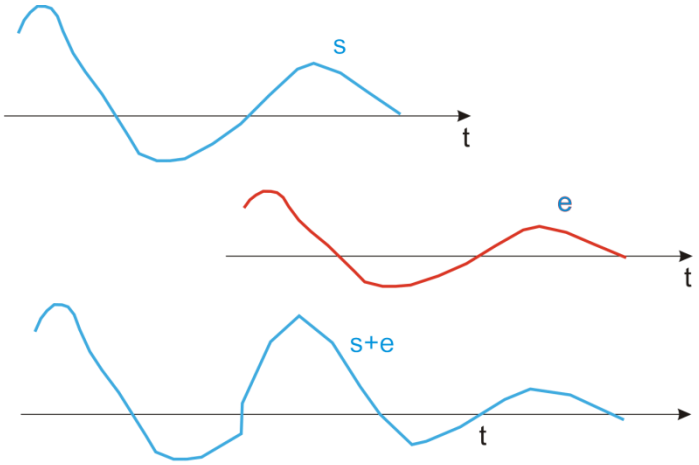
- **input_audio.wav** - speech or music s(n)
- **input_noise1.wav** - noise e(n) after amplification or attenuation
- **input.wav** - input signal d(n)=s(n)+e(n)
- **input_noise2.wav** - noise x(n) from the second microphone
- **denoised.wav –** output signal $\varepsilon(n)$
- **output_noise.wav** – signal $\varepsilon(n)-d(n)$

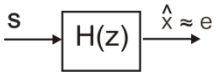
  The most important are signals **input.wav** and **denoised.wav**.

Tasks:

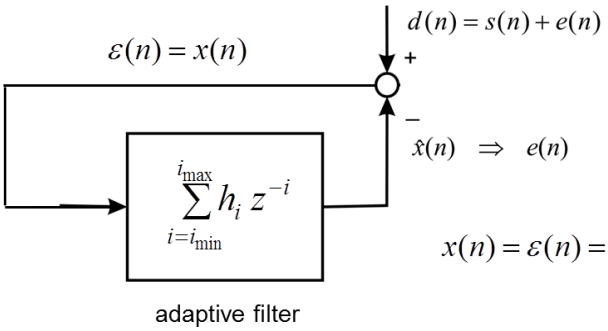
1. Select the audio file (only one) and set SNR_in = 0 dB. Note the improvement of SNR (i.e. SNR_out-SNR_in [dB]) due to adaptive filtering. Run simulations at different adaptation speed (e.g. β=0.001, 0.01, 0.1, 1, 10, 20). Describe the influence of the adaptation speed β on denoising process. What happens if the adaptation speed is too high? And too low?
2. Keep the best adaptation speed (task1) and note the improvement of SNR at different values of the input SNR (SNR_in = -5 dB, 0 dB, 5 dB, 10 dB, 20 dB). Comment on the results.

2. ECHO CANCELLATION IN AUDIO FILE





Signal s is unknown, so s+e-$\hat{x}$ is filtered



$s(n)$ - original speech or music

$e(n)$ - echo, e.g. $e(n)$ = a s(n-del)

$d(n)$- signal with echo (audio file)

$\hat{x}(n) \approx e(n)$ - approximation of echo signal

$x(n) = \varepsilon(n) = s(n) + e(n) - \hat{x}(n) \approx s(n)$ - audio signal with echo suppressed

Searched: impulse response of the adaptive filter: $\bar{h} = [h_{i\min}, \ldots, h_{i\max}]^T$ (the other coefficients are equal to zero). Criterion: minimum energy of the error signal: $\sum_n \varepsilon^2(n)$.

Solution: LMS (least mean square) algorithm: $\bar{h}(n+1) = \bar{h}(n) + \beta\,\varepsilon(n) \begin{pmatrix} x(n-i_{\min}) \\ x(n-i_{\min}-1) \\ \vdots \\ x(n-i_{\max}) \end{pmatrix}$

The input signal d(n) consists of an audio signal s(n) and a single echo (attenuation *a,* delay *del*):

$d(n) = s(n) + a\,s(n-del)$, which may be described using zed transform

$D(z) = S(z) + a\,S(z)\,z^{-del} = S(z)[1 + a\,z^{-del}]$.

The system presented above is described as follows:

$$X(z) = D(z) - X(z)\,H(z)$$
$$X(z)[1 + H(z)] = D(z)$$

where $H(z) = \sum_{i=i_{\min}}^{i_{\max}} h_i z^{-i}$. By comparing two formulas for D(z):

$D(z) = S(z)[1 + a\, z^{-del}] = X(z)[1 + H(z)]$ we come to the conclusion, that if $H(z) = a\, z^{-del}$, then X(z)=S(z) and x(n)=s(n). In this way we obtain the audio signal x(n) with suppressed echo.

Simulations:

Select option `echo`. Then select the name of audio file. The following wav files sampled at 8000 Hz are available: F1, F2, M1, FRANC, DIABOJ (longer phrases are preferred). In this way you define the signal s(n).

Enter echo delay `del` (up to 1 second) and echo amplitude `a` (i.e. echo/signal ratio, it should be less than 1). Program will produce the input signal $d(n) = s(n) + a\, s(n - del)$.

Enter the adaptation speed β (see the formula describing the LMS algorithm). Due to the feedback loop smaller values should be used, than in denoising problem.

Observe evolution of the impulse response of the adaptive filter (the impulse response contains 100 nonzero samples).

At the end of simulation note the signal-to-echo ratio before and after denoising **SNR_in_dB, SNR_out_dB**. They are measured for the second half of the file, so as to skip errors of the initial adaptation phase.

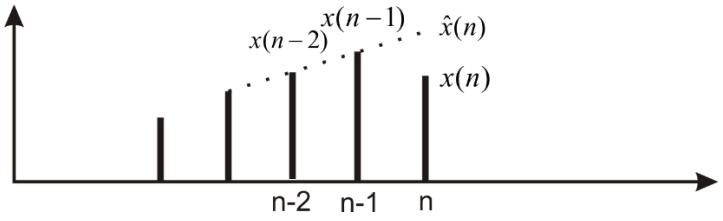
Display and listen to the following signals:

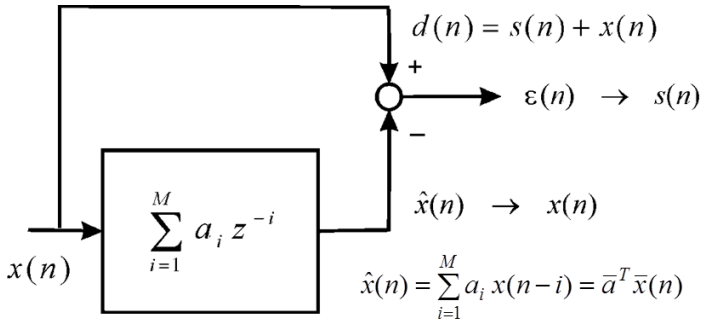
- **input_wave.wav** - signal d(n) (audio s(n) plus echo)
- **output_wave.wav** signal $\varepsilon(n) = x(n)$
- **echo_input.wav** signal e(n)=d(n)-s(n)
- **echo_output.wav** signal $\varepsilon(n) - s(n)$

Tasks:

1. Select the audio file (only one) and set echo amplitude a=0.5. Note the improvement of signal to echo ratio (i.e. SNR_out-SNR_in [dB]) due to adaptive filtering. Find the best adaptation speed β .
2. Keep the best adaptation speed (task1) and note the improvement of signal to echo ratio at different values of echo amplitude (a). Comment on the results.

3. PREDICTION





$x(n)$ - current signal sample

$s(n)$ - measurement noise or zero

$\hat{x}(n)$ -prediction of $x(n)$

$\varepsilon(n) = s(n) + x(n) - \hat{x}(n)$ - prediction error

Searched: prediction coefficients $\bar{a} = [a_1, ..., a_M]^T$.

Criterion: minimum energy of prediction error signal: $\sum_n \varepsilon^2(n)$.

Solution: LMS (least mean square) algorithm: $\bar{a}(n+1) = \bar{a}(n) + \beta\,\varepsilon(n) \begin{pmatrix} x(n-1) \\ x(n-2) \\ \vdots \\ x(n-M) \end{pmatrix}$

where β - filter adaptation speed (called also the step size).

Simulations:

Select option `prediction`. Then select the name of audio file. The following speech files sampled at 8000 Hz are available: F1, F2, M1, DIABOJ, FRANC . Sine signal of frequency equal to 444 Hz is also prepared (SIN444). In this way you define the signal d(n)=x(n).

Prediction of current sample is a linear combination of previous M samples. Prediction coefficients are adjusted for every signal sample, using the LMS algorithm.

Enter the number of prediction coefficients M, i.e. number of previous samples used for prediction of the current sample. Then enter the adaptation speed (step size) β.

Observe the evolution of prediction coefficients (impulse response of the predictor). At the end observe waveforms of the input signal x(n) and prediction error signal $\varepsilon(n)$. Note the prediction gain **Gain_dB**, i.e. ratio of input signal power to prediction error power, expressed in decibels.

Display and listen to the following signals:

- **input.wav** - input audio signal d(n)=x(n)
- **prediction.wav** signal $\hat{x}(n)$
- **pred_error.wav** signal $\varepsilon(n)$

Tasks:

1. Select the speech file (only one), then input the number of prediction coefficients M=10. Run simulations at different adaptation speed (e.g. β=0.001, 0.01, 0.1, 1, 10, 20). Describe the influence of the adaptation speed β on prediction gain.
2. Keep the optimum value of adaptation speed β and test the influence of the number of coefficients M on prediction gain. Choose M=1, 2, 5, 10, 20, 40.
3. Select the sine signal of frequency 444 Hz (SIN444.wav) and test the predictor of 1, 2 and 4 coefficients (adjust properly the adaptation speed β). What is the difference between results obtained for speech signal and sine signal?