PROJECT 3: INVERSION OF ROOM ACOUSTICS AND CREATION OF VIRTUAL SOUND SOURCES BY DIRECT ADAPTATION OF PRE-COMPENSATION FILTERS

Project supervisor:
Lars-Johan Brännmark
Signals & Systems
018-471 3071
073-572 7528
ljb@signal.uu.se
LECTURE OVERVIEW

• Overall project aims
• Scalar Filtered-X LMS
• Multichannel Filtered-X LMS
• Limitations of standard FIR approach
• Warping of the frequency axis
• A linear/warped hybrid scalar LMS algorithm
• Project details
• Audio I/O in MATLAB.
• Goal 1: Improve sound reproduction in a stereo system by using a scalar pre-compensation filter on each channel.

How? Scalar filters are obtained with a frequency-warped version of the Filtered-X LMS algorithm.
• **Goal 2**: Control the sound pressure at the ears of a listener, using two (closely spaced) loudspeakers, and microphones at the listener’s ears. Start with a two-loudspeaker version, then extend to four loudspeakers.

**How?** The desired FIR filter matrix is obtained using a multivariable version of the Filtered-X LMS algorithm.
Single-channel acoustic system $c(q^{-1})$:

Problem: Design signal $y(t)$ to minimize power of $e(t) = d(t) - \tilde{d}(t)$:

( In this case, $d(t) = x(t - \Delta)$ )
\[ y(t) = \sum_{i=0}^{I-1} h_i x(t - i) \quad ; \quad \hat{d}(t) = \sum_{j=0}^{J-1} c_j y(t - j) \]

\[ \hat{d}(t) = \sum_{j=0}^{J-1} c_j \sum_{i=0}^{I-1} h_i x(t - i - j) = \sum_{i=0}^{I-1} h_i \sum_{j=0}^{J-1} c_j x(t - i - j) \]

Suppose \( c_j \), or an estimate \( \hat{c}_j \), is available

Now let \( r(t) = \sum_{j=0}^{J-1} c_j x(t - j) \)

Then we get \( \hat{d}(t) = \sum_{i=0}^{I-1} h_i r(t - i) = r^T(t) h \)

\[ e(t) = d(t) - r^T(t) h \]

where \( r(t) = [r(t) \; r(t - 1) \; \ldots \; r(t - I + 1)]^T \)

\[ h = [h_0 \; h_1 \; \ldots \; h_{I-1}]^T \]
Minimize \( J = \mathbb{E}\{e^2(t)\} \) w.r.t. \( h \) (Mean Squared Error)

Gradient:
\[
\frac{\partial J}{\partial h_i} = 2\mathbb{E}\{e(t)\frac{\partial e(t)}{\partial h_i}\} ; \ i = 0, 1, \ldots, I - 1
\]
\[
\frac{\partial e(t)}{\partial h_i} = -r(t - i)
\]

LMS adaptation law: \( h_i(t + 1) = h_i(t) + \alpha r(t - i)e(t) \)
\[
\implies h(t + 1) = h(t) + \alpha r(t)e(t)
\]

The working principle of the algorithm can be illustrated as follows:

```
\text{"Filtered-X LMS"}
```
Loudspeaker input signals:

\[ y_1(t) = \sum_{i=0}^{I-1} h_{11}(i)x_1(t - i) + \sum_{i=0}^{I-1} h_{12}(i)x_2(t - i) \]

\[ y_2(t) = \sum_{i=0}^{I-1} h_{21}(i)x_1(t - i) + \sum_{i=0}^{I-1} h_{22}(i)x_2(t - i) \]

Microphone output signals:

\[ \hat{d}_1(t) = \sum_{j=0}^{J-1} c_{11}(j)y_1(t - j) + \sum_{j=0}^{J-1} c_{12}(j)y_2(t - j) \]

\[ = \sum_{j=0}^{J-1} c_{11}(j) \left[ \sum_{i=0}^{I-1} h_{11}(i)x_1(t - i - j) + \sum_{i=0}^{I-1} h_{12}(i)x_2(t - i - j) \right] \]

\[ + \sum_{j=0}^{J-1} c_{12}(j) \left[ \sum_{i=0}^{I-1} h_{21}(i)x_1(t - i - j) + \sum_{i=0}^{I-1} h_{22}(i)x_2(t - i - j) \right] \]

Now assume that the acoustic channel impulse responses \( c_{pq}(j) \) are known, and change summation order!
\[ \hat{d}_1(t) = \sum_{i=0}^{I-1} h_{11}(i) \sum_{j=0}^{J-1} c_{11}(j)x_1(t - i - j) + \sum_{i=0}^{I-1} h_{12}(i) \sum_{j=0}^{J-1} c_{11}(j)x_2(t - i - j) \]
\[ + \sum_{i=0}^{I-1} h_{21}(i) \sum_{j=0}^{J-1} c_{12}(j)x_1(t - i - j) + \sum_{i=0}^{I-1} h_{22}(i) \sum_{j=0}^{J-1} c_{12}(j)x_2(t - i - j) \]
\[ = \sum_{i=0}^{I-1} h_{11}(i)r_{111}(t - i) + \sum_{i=0}^{I-1} h_{12}(i)r_{112}(t - i) \]
\[ + \sum_{i=0}^{I-1} h_{21}(i)r_{121}(t - i) + \sum_{i=0}^{I-1} h_{22}(i)r_{122}(t - i) \]

Similarly,
\[ \hat{d}_2(t) = \sum_{i=0}^{I-1} h_{11}(i)r_{211}(t - i) + \sum_{i=0}^{I-1} h_{12}(i)r_{212}(t - i) \]
\[ + \sum_{i=0}^{I-1} h_{21}(i)r_{221}(t - i) + \sum_{i=0}^{I-1} h_{22}(i)r_{222}(t - i) \]

where \( r_{klm}(t) = \sum_{j=0}^{J-1} c_{kl}(j)x_m(t - j) \); \( k, l, m \in \{1, 2\} \)
In matrix form:

\[
\begin{bmatrix}
\hat{d}_1(t) \\
\hat{d}_2(t)
\end{bmatrix} =
\begin{bmatrix}
r_{111}(t) \cdots r_{111}(t-I+1) \\
r_{211}(t) \cdots r_{211}(t-I+1) \\
r_{122}(t) \cdots r_{122}(t-I+1) \\
r_{222}(t) \cdots r_{222}(t-I+1)
\end{bmatrix}
\begin{bmatrix}
h_{11}(0) \\
h_{11}(I-1) \\
h_{22}(0) \\
h_{22}(I-1)
\end{bmatrix}
\]

More compactly (notation as in Elliott/Nelson):

\[\hat{d}(t) = R(t)h\]

Now let \( e(t) = \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} = d(t) - \hat{d}(t) = d(t) - R(t)h \)

where

\[d(t) = \begin{bmatrix} d_1(t) \\ d_2(t) \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{K-1} d_{11}(k)x_1(t-k) + \sum_{k=0}^{K-1} d_{12}(k)x_2(t-k) \\ \sum_{k=0}^{K-1} d_{21}(k)x_1(t-k) + \sum_{k=0}^{K-1} d_{22}(k)x_2(t-k) \end{bmatrix}\]

Minimization of \( J = e_1^2(t) + e_2^2(t) = \mathbb{E}\{e^T(t)e(t)\} \)

yields the LMS tap weight vector update

\[h(t+1) = h(t) + \alpha R^T(t)e(t)\]

Correct step length \( \alpha \) important for convergence, see references…
Challenge: Extend the above 2-input-2-output approach to a 5-input-2 output i.e. create a virtual home cinema using only 2 loudspeakers...

• Choosing $d_{12}(k) = d_{21}(k) = 0$; $k = 0, \ldots, K - 1$ and $d_{11}(k) = d_{22}(k) = \delta(k - d)$; $k = 0, \ldots, K - 1$ for some delay $d$ gives a crosstalk canceller, or virtual headphones.

• “Virtual sources” are obtained if the desired responses are measured for loudspeakers at different positions, or by using the proper Head-Related Transfer Functions (HRTFs), associated with the desired virtual source directions.
LIMITATIONS OF STANDARD FIR FILTERS

- Good low-frequency performance requires very long FIR filters.
- LMS convergence is very slow for long FIR filters.
- In audio applications, a logarithmic frequency resolution is often desirable.
- General IIR filters \( \frac{B(q^{-1})}{A(q^{-1})} \) not well suited for adaptation. (Local minima, stability issues etc.)

- Solution: Linear regression in arbitrary functions of past data.

\[
y(t) = \sum_{i=0}^{I-1} h_i W_i(q^{-1}) x(t) \quad \text{instead of} \quad y(t) = \sum_{i=0}^{I-1} h_i q^{-i} x(t)
\]

where each \( W_i(q^{-1}) \) is a rational function in the delay operator \( q^{-1} \)

What is a good choice of basis functions?
FREQUENCY WARPING

- Standard FIR filter, or tapped delay line, \( W_i(q^{-1}) = q^{-i} \):

- Instead of delays, use, for example, all-pass elements:

\[
W_i(q^{-1}) = \left( \frac{\lambda + q^{-1}}{1 + \lambda q^{-1}} \right)^i = \left( \frac{\lambda + q^{-1}}{1 + \lambda q^{-1}} \right) W_{i-1}(q^{-1}) \quad ; \quad W_0(q^{-1}) = 1
\]

- The frequency axis (unit circle) is mapped onto itself (conformal mapping):
  - Delays become frequency-dependent
  - \(-1 < \lambda < 0\) yields an increased low-frequency resolution.
  - Generalizations: Laguerre and Kautz filters.
The warped scalar FXLMS

\[ y(t) = \sum_{i=0}^{I-1} h_i W_i(q^{-1}) x(t) \quad ; \quad \hat{d}(t) = \sum_{j=0}^{J-1} c_j y(t-j) \]

\[ \hat{d}(t) = \sum_{j=0}^{J-1} c_j \sum_{i=0}^{I-1} h_i W_i(q^{-1}) x(t-j) = \sum_{i=0}^{I-1} h_i W_i(q^{-1}) \sum_{j=0}^{J-1} c_j x(t-j) \]

\[ \Rightarrow \hat{d}(t) = \sum_{i=0}^{I-1} h_i W_i(q^{-1}) r(t) = r^T(t) h \]

\[ \text{where} \quad r(t) = \begin{bmatrix} r(t) & W_1(q^{-1}) r(t) & W_2(q^{-1}) r(t) & \cdots & W_{I-1}(q^{-1}) r(t) \end{bmatrix}^T \]

\[ h = \begin{bmatrix} h_0 & h_1 & \cdots & h_{I-1} \end{bmatrix}^T \]

As before, the LMS update of \( h \) is obtained as

\[ h(t+1) = h(t) + \alpha r(t) e(t) \]

\[ \text{where} \quad \alpha \text{ is the step size, and} \]

\[ e(t) = d(t) - \hat{d}(t) = d(t) - r^T(t) h(t) \]
NOTE 1: \( r(t) \) is now a more complicated function of past data, compared with the standard linear FIR case.

\[
\begin{align*}
\mathbf{r}(t) & = \begin{bmatrix} r(t) & W_1(q^{-1})r(t) & W_2(q^{-1})r(t) & \cdots & W_{I-1}(q^{-1})r(t) \end{bmatrix}^T \\
& = \begin{bmatrix} \rho_0(t) & \rho_1(t) & \cdots & \rho_{I-1}(t) \end{bmatrix}^T
\end{align*}
\]

where

\[
\begin{align*}
\rho_0(t) & = r(t) \\
\rho_1(t) & = -\lambda \rho_1(t - 1) + \lambda \rho_0(t) + \rho_0(t - 1) \\
\vdots & \\
\rho_i(t) & = -\lambda \rho_i(t - 1) + \lambda \rho_{i-1}(t) + \rho_{i-1}(t - 1) \\
\vdots & \\
\rho_{I-1}(t) & = -\lambda \rho_{I-1}(t - 1) + \lambda \rho_{I-2}(t) + \rho_{I-2}(t - 1)
\end{align*}
\]

NOTE 2: Using the coefficients in \( \mathbf{h} \), an “unwarped” impulse response \( g(t) \) of arbitrary length is readily obtained by sending a unit pulse \( \delta(t) \) into the warped structure and reading the output for as many samples as desired:

\[
\begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_I \end{bmatrix} = \begin{bmatrix} \delta(t) & W_1(q^{-1})\delta(t) & W_2(q^{-1})\delta(t) & \cdots & W_{I-1}(q^{-1})\delta(t) \end{bmatrix}
\]
A LINEAR / WARPED HYBRID APPROACH

• If the acoustic system $c(q^{-1})$ is not minimum phase, inversion requires a noncausal filter, which can be approximated by allowing a delay in the compensated system, $d(t) = q^{-\Delta}x(t)$.

• The warped basis functions are not well suited for such “noncausal” filters, having long initial delays.

• Better to keep a high time-domain resolution in the early part of the filter’s impulse response.

The basis functions $W_i(q^{-1})$ are now defined as

$$W_i(q^{-1}) = q^{-i}$$

$$W_i(q^{-1}) = q^{-(I_0-1)} \left( \frac{\lambda + q^{-1}}{1 + \lambda q^{-1}} \right)^{i-(I_0-1)}$$

; $i = 0, 1, \ldots, I_0 - 1$

; $i = I_0, \ldots, I - 1$
SOME REMARKS ABOUT THE PROJECT WORK

• Online adaptation requires fast computations and very low input-output latency (computer sound card not sufficient).

• In this project, filter adaptation is performed off-line, using measured impulse responses $c(k), d(k), c_{pq}(k), d_{pq}(k)$ and noise signals $r(t), d(t), r_{klm}(t), d_1(t), d_2(t)$ generated from them.

• Desired impulse responses $d(k), d_{pq}(k)$ are either user-defined (as e.g., in crosstalk cancellation), or are obtained by system identification of loudspeakers at different positions (for creation of virtual sources).

• Impulse responses may need some processing (e.g. truncation, windowing) before the signals are generated.

• MATLAB functions for system identification, multichannel FIR filtering and audio playback will be provided on the course webpage.
• **GOAL 1:** Loudspeaker compensation with scalar filters, using LMS adaptation of the parameters in the hybrid linear/warped structure.

• Implement the scalar FXLMS using the hybrid warped structure.

• Investigate how the result depends on different parameters, for example: Number of coefficients $I$, warping parameter $\lambda$, hybrid structure “linear/warped tradeoff parameter” $I_0$, desired system delay $\Delta$, number of iterations, stepsize $\alpha$ etc.

• Use “unwarped” impulse responses for real-time filtering and plotting.

• Subjective results: Music listening in stereo.

• Spatial robustness: Impulse response of the compensated system at other spatial positions.
• **GOAL 2**: Virtual sound sources perceived within in a +/- 90° panorama, using the 2x2 MIMO LMS algorithm.

• Implement the MIMO FXLMS.

• How does the result depend on filter lengths, truncation/windowing of models, loudspeaker positioning etc.? Compare your results with available theory.

• To what extent is crosstalk cancellation possible? In what frequency regions? Measure the transfer functions after filtering with the crosstalk canceller matrix.

• What about position sensitivity?

• Extend the given algorithm to include four loudspeakers. Implement and evaluate the results. (This should be possible with the 8-channel Firepod soundcard).

• Further experimenting. Use your imagination!
AUDIO I/O IN MATLAB

• Important to have consistent input/output delays.
• Standard Windows sound drivers, supported by MATLAB’s Data Acquisition Toolbox, are unreliable. I/O latency may vary from time to time.
• ASIO drivers (developed by Steinberg) are better in this respect, but are not supported by MATLAB.
• Fortunately, an ASIO-based Matlab utility for audio playback and recording, called Playrec, has been developed at KTH. Available through the webpage: http://www.playrec.co.uk/
• This utility will be installed on all project workstations.
• Functions crucial for getting started will be provided on the course homepage.
MATLAB FUNCTIONS FOR GETTING STARTED

• Matlab code for audio recording, playback, real-time filtering and system identification will be provided on the course web pages.
  • Playrec functionality demonstration
    – A script which implements a simple real-time block-based data acquisition.
      • playrec_demoscript.m
  • System identification
    – A function which measures the impulse responses of a MIMO acoustical system.
      • ASIOImpRespMeasure.m
  • Filtering
    – Real-time FIR MIMO filtering and playback.
      • ASIOFilterPlayback.m
  • Plotting
    – Magnitude and relative phase of FIR systems
      • FIR_bode.m
• Other functions, applicable in other projects, are available on the course home page as well.
REFERENCES

Strongly recommended:


Other:


• The virtual acoustics project at the University of Southampton: http://www.isvr.soton.ac.uk/FDAG/VAP/