This lecture deals with

- Repetition of the course Optimal signal processing (OSB)
- The method of the Steepest descent

Optimal Linear Filtering

The filter \( w = [w_0 \ w_1 \ w_2 \ldots]^T \) which minimizes the estimation error \( e(n) \), such that the output signal \( y(n) \) resembles the desired signal \( d(n) \) as much as possible is searched for.

In order to determine the optimal filter a cost function \( J \), which punish the deviation \( e(n) \), is introduced. The larger \( e(n) \), the higher cost.

From OSB you know some different strategies, e.g.,

- The total squared error (LS) Deterministic description of the signal.
  \[ J = \sum_{n_1}^{n_2} e^2(n) \]

- Mean squared error (MS) Stochastic description of the signal.
  \[ J = E\{|e(n)|^2\} \]

- Mean squared error with extra contraint
  \[ J = E\{|e(n)|^2\} + \lambda |u(n)|^2 \]
Optimal Linear Filtrering

The gradient operator yields
\[ \nabla J(w) = 2 \frac{\partial J(w)}{\partial w^*} = 2 \frac{\partial}{\partial w^*} (\sigma_d^2 w^H p - p^H w + w^H R w) \]
\[ = -2p + 2Rw \]

If the gradient vector is set to zero, the Wiener-Hopf equation system results
\[ Rw_o = p \quad \text{Wiener-Hopf} \]
which solution is the Wiener filter.
\[ w_o = R^{-1} p \quad \text{Wienerfilter} \]

In other words, the Wiener filter is optimal when the cost is controlled by $MSE$.

Adaptive Signal Processing 2011 Lecture 1

Steepest Descent

The method of the Steepest descent is a recursive method that leads to the Wiener-Hopfs equations. The statistics are known ($R$, $p$). The purpose is to avoid inversion of $R$ (saves computations)

- Set start values for the filter coefficients, $w(0)$ ($n = 0$)
- Determine the gradient $\nabla J(n)$ that points in the direction in which the cost function increases the most. $\nabla J(n) = -2p + 2Rw(n)$
- Adjust $w(n+1)$ in the opposite direction to the gradient, but weight down the adjustment with the stepsize parameter $\mu$
  \[ w(n+1) = w(n) + \frac{1}{\mu} [-\nabla J(n)] \]
- Repeat steps 2 and 3.
Exempel: Modellering/Identifiering

The influence of the stepsize parameter on the convergence can be seen when analyzing $J(w)$. The example below illustrates the convergence towards $J_{\text{min}}$, for different choices of $\mu$.

\[
p = \begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix} \]
\[
R = \begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \]
\[
w_0 = \begin{bmatrix} 0.8360 \\ -0.7853 \end{bmatrix} \]
\[
w(0) = \begin{bmatrix} 1 \\ 1.7 \end{bmatrix} \]

Adaptive Signal Processing 2011 Lecture 1

Convergence analysis

With the observation that $w(n) = Qv(n) + w_0$, the update of the cost function can be derived:

\[
w(n + 1) = w(n) + \mu[p - Rw(n)] \]
\[
Qv(n + 1) + w_0 = Qv(n) + w_0 + \mu[p - RQv(n) - Rw_0] \]
\[
v(n + 1) = v(n) - \mu Q \dot{R} Q v(n) = (1 - \mu \lambda) v(n) \]
\[
v_k(n + 1) = (1 - \mu \lambda_k) v_k(n) \quad (\text{Element } k \text{ in } v(n)) \]

The latter is a 1st order difference equation, with the solution

\[
v_k(n) = (1 - \mu \lambda_k)^n v_k(0) \]

For this equation to converge it is required that $|1 - \mu \lambda_k| < 1$, which leads to the stability criterion of the method of the Steepest Descent:

\[
0 < \mu < \frac{2}{\lambda_{\text{max}}} \]

Stabilitet, S.D.

Adaptive Signal Processing 2011 Lecture 1

Exempel: Ekosläckare II

Vid modellering/identifiering kopplas det adaptiva filtretparallellt med det undersöka systemet. Om det undersöka systemet endast har nollställen, så är det lämpligt att använda motsvarande längd på filtretparallellt med det undersöka systemet. Om det undersöka systemet endast har nollställen krävs i regel ett längt adpativt FIR-filter.

Adaptive Signal Processing 2011 Bilaga, Föreläsning 1

Denna struktur är tillämplig på högtalartelefoner, videokonferenssystem och dylikt. Precis som vid telefonifallet hör Talare 1 sig själv i form av ett echo. Dock är denna effekt mer påtaglig här, eftersom mikrofonen fängar upp det som sänds ut av högtalaren. Adaptionen stoppas normalt när Talare 2 pratar, men filteringen sker under hela tiden.

Adaptive Signal Processing 2011 Bilaga, Föreläsning 1
Information i form av en periodisk signal störas av ett färgat brus som är korrelerat med sig själv inom en viss tidsram. Genom att fördjöja signalen så pass mycket att bruset (i $u(n)$ respektive $d(n)$) blir okorrelaterat, kan bruset tryckas ned genom linjär prediktion.

Adaptive Signal Processing 2011 Bilaga, Föreläsning 1

**Exempel: Kanalutjämnmare (Equalizer)**

En känd pseudo noise-sekvens används för att skatta en invers modell av kanalen ("tränings"). Därefter stoppas adaptionen, men filtret fortsätter att verka på den överförda signalen. Syftet är att ta bort kanalens inverkan på den överförda signalen.

Adaptive Signal Processing 2011 Bilaga, Föreläsning 1

**Lecture 2**

During this lecture you will learn about

- The Least Mean Squares algorithm (LMS)
- Convergence analysis of the LMS
- Equalizer (Kanalutjämnmare)

Adaptive Signal Processing 2011 Lecture 2

**The Least Mean Square (LMS) algorithm**

We want to create an algorithm that minimizes $E\{ |e(n)|^2 \}$, just like the SD, but based on unknown statistics.

A strategy that then can be used is to uses estimates of the autocorrelation matrix $\hat{R}$ and the cross correlationen vector $\hat{p}$. If instantaneous estimates are chosen,

$$\hat{R}(n) = u(n)u^H(n)$$
$$\hat{p}(n) = u(n)d^*(n)$$

the resulting method is the Least Mean Squares algorithm.

Adaptive Signal Processing 2011 Lecture 2
The Least Mean Square (LMS) algorithm

For the SD, the update of the filter weights is given by

$$w(n+1) = w(n) + \mu \nabla J(n)$$

where $$\nabla J(n) = -2p(n) + 2R(n)w(n)$$

In the LMS we use the estimates $$\hat{R}$$ och $$\hat{p}$$ to calculate $$\nabla J(n)$$. This, also the updated filter vector becomes an estimate. It is therefore denoted $$\hat{w}(n)$$:

$$\nabla J(n) = -2\hat{p}(n) + 2\hat{R}(n)\hat{w}(n)$$

Properties of the LMS

- Convergence for the same interval as the SD:
  $$0 < \mu < \frac{1}{\lambda_{\text{max}}}$$

- $$J_{\text{ex}}(\infty) = J_{\text{min}} + \frac{\mu}{2} \sum_{i=1}^{M} \lambda_i$$

- Misadjustment $$\mathcal{M} = \frac{J_{\text{ex}}(\infty)}{J_{\text{min}}}$$ is a measure of how close the the optimal solution the LMS (in mean-square sense).

Rules of thumb LMS

The eigenvalues of $$R$$ are rarely known. Therefore, a set of rules of thumbs is commonly used, that is based on the tap-input power,

$$\sum_{k=0}^{M-1} E[|u(n-k)|^2] = \text{tr}(R)$$

- The stepsize $$0 < \mu < \frac{1}{\sum_{k=0}^{M-1} E[|u(n-k)|^2]}$$

- Misadjustment $$\mathcal{M} = \sum_{k=0}^{M-1} E[|u(n-k)|^2]$$

- Average time constant $$\tau_{\text{mse,av}} \approx \frac{1}{2} \int_{0}^{\infty} \frac{d\tau}{\lambda_{\text{av}}}$$

- Approximate relationship $$\mathcal{M}$$ and $$\tau_{\text{mse,av}}$$:
  $$\mathcal{M} \approx \frac{M}{\tau_{\text{mse,av}}}$$

Here, $$\tau_{\text{mse,av}}$$ denotes the time it takes for $$J(n)$$ to decay a factor of $$e^{-1}$$. 
Example, cont: Learning Curve

The curves illustrate learning curves for two different $\mu$.

Adaptive Signal Processing 2011 Lecture 2

Lecture 3

Lecture 3 includes the following:

- Eigenvalue spread of $R$ and its influence on the convergence speed for the LMS.
- Variants of the LMS:
  - The Normalized LMS
  - The Leaky LMS
  - The Sign LMS
- The Echo Canceller

Adaptive Signal Processing 2011 Lecture 3

The Normalized LMS

For the standard LMS which we have looked at earlier, the step is proportional to $u(n)$:

$$\hat{w}(n+1) = \hat{w}(n) + \mu u(n) e^*(n).$$

This means that the gradient noise is amplified when the signal is strong.

One solution to this problem is to normalize the update $\tilde{w}(n)$ med $||u(n)||^2 = u^H(n)u(n)$:

$$\tilde{w}(n+1) = \tilde{w}(n) + \frac{\bar{\mu}}{\alpha + ||u(n)||^2}u(n)e^*(n)$$

where $0 < \bar{\mu} < 2$, and $\alpha$ is a positive protection constant.
The Leaky LMS

As for the standard LMS we start with the method of the Steepest descent,
\[ w(n+1) = w(n) + \mu \left( -\nabla J(n) \right) , \]
but we now instead use the cost function
\[ J(n) = E\left\{ |e(n)|^2 \right\} + \alpha ||w(n)||^2 = E\{e^*(n)e(n)\} + \alpha w^H(n)w(n) . \]

The gradient becomes
\[ \nabla J(n) = -2p + 2Rw(n) + 2\alpha w(n) . \]

With estimated statistics we get
\[ \nabla J(n) = -2\hat{p}(n) + 2\hat{R}(n)\hat{w}(n) + 2\alpha \hat{w}(n) . \]

The Sign LMS

Again, starting from the method of the Steepest descent
\[ w(n+1) = w(n) + \mu \left( -\nabla J(n) \right) , \]
and the cost function
\[ J(n) = E\{ |e(n)| \} \]
the gradient becomes
\[ \nabla J(n) = -\text{sign}(e(n))u(n) . \]

With this gradient, the update equation becomes
\[ \hat{w}(n+1) = \hat{w}(n) + \alpha \cdot \text{sign}(e(n))u(n) . \]

The Sign LMS is used in applications with extremely high requirements on computational complexity.

Lecture 4

Lecture 4 contains descriptions of
- Block-based LMS (7.1) (edition 3: 10.1)
- Frequency Domain LMS (FDAF, 7.2-7.3) (edition 3: 10.2-10.3)
Summary of the Block-LMS

1. \( y(kL+i) = \mathbf{w}^T(k)\mathbf{u}(kL+i), \) L st i.e., \( y(k) = \mathbf{w}^T(k)\mathbf{u}(k). \)

2. \( e(kL+i) = d(kL+i) - y(kL+i) \)
i.e., \( e(k) = d(k) - y(k). \)

3. \( \mathbf{w}(k+1) = \mathbf{w}(k) + \mu \sum_{i=0}^{L-1} \mathbf{u}(kL+i)e(kL+i) \)
i.e., \( \mathbf{w}(k+1) = \mathbf{w}(k) + \mu \mathbf{u}(k)e(k). \)

In applications with long filters, e.g. echo cancellation, the complexity becomes very high. The equations that take time are the filtering in (1) and the cross correlation in (3).

FDAF

In order to increase the calculation speed of the LMS algorithm, the filtering (convolution) and the gradient estimation (crosscorrelation) can be done in the frequency domain instead of the time domain.

Strategy:
1. FFT of input and error signals
2. Convolution and crosscorrelation corresponds to multiplication in the frequency domain
3. IFFT

Two advantages of the FDAF:
1. Faster calculation
2. Independent coefficients

Problem

Multiplication in the frequency domain corresponds to circular convolution, but in order to maintain the properties of the LMS, linear convolution must be used.

- Filtering must be done with linear convolution.
- Gradient estimation should be done with linear convolution. Then the method is called Fast LMS. If linear convolution is not used here the method is called Unconstrained FDAF.

Properties of the Fast LMS, cont.

- The convergence speed for the Fast LMS can be optimized for each mode separately.
- The convergence speed for the \( i \)-th mode depends on \( \mu_i, \lambda_i \). A measure of \( \lambda_i \) is the average power in the frequency bin of the \( i \)-th mode,
\[
P_i = \left| U_i \right|^2.
\]
If \( \mu_i = \frac{\alpha}{P_i} \), all modes will converge equally fast (WSS).
- If the input signal is not WSS, \( P_i \) must be estimated recursively
\[
P_i(k) = \gamma P_i(k-1) + (1 - \gamma)\left| U_i(k) \right|^2
\]
- The stepsize parameter \( \mu \) is here substituted by a diagonal \( 2M \times 2M \) matrix \( \mu = \alpha \mathbf{D}(k) \), where \( \mathbf{D}(k) = \text{diag}(P_0^{-1}, P_1^{-1}, \ldots, P_{2M-1}^{-1}) \).
Fast LMS, Update equations

\[ U(k) = \text{diag}(\text{FFT}[u((k-1)M) \ldots u(kM-1), u(kM) \ldots u((k+1)M-1)])^T \]

\[ y(k) = \text{last } M \text{ elements of IFFT}[U(k)\tilde{W}(k)] \]

\[ d(k) = [d(kM) \ldots d(kM+1) \ldots d((k+1)M-1)]^T \]

\[ e(k) = d(k) - y(k) \]

\[ E(k) = \text{FFT} \begin{bmatrix} 0 \\ u(k) \end{bmatrix} \]

\[ P(k) = \gamma P(k-1) + (1-\gamma)U^H(k)U(k) \]

\[ D(k) = P^{-1}(k) = \text{diag}[P_0^{-1}(k), P_1^{-1}(k), \ldots, P_{2M-1}^{-1}(k)] \]

\[ \phi(k) = \text{first } M \text{ elements of IFFT}[D(k)U^H(k)E(k)] \]

\[ \tilde{W}(k+1) = \tilde{W}(k) + \alpha \text{FFT} \begin{bmatrix} \phi(k) \\ 0 \end{bmatrix} \]

Adaptive Signal Processing 2011 Lecture 4

Lecture 5 presents

- Optimal LS filter
- Windowing of data
- Weighted LS filter
- Recursive LS filter (RLS)

Adaptive Signal Processing 2011 Lecture 5

LS-filtret

The derivation of the normal equations for the LS filter is made in the same way as for the Wiener filter, but with the difference that the input signal now is known instead of that its statistics is known (Wiener filter).

We therefore choose to minimize the total error over a certain interval \( \{i_1, i_2\} \):

\[ E = \sum_{i=i_1}^{i_2} |e(n)|^2 \]

instead of the average error \( E(\bullet) \).

Adaptive Signal Processing 2011 Lecture 5

LS-filtret

When minimizing the total error \( \sum_{i=1}^{i_2} |e(n)|^2 \) terms of the following type instead appears

\[ \sum_{i=i_1}^{i_2} u(i-k)u^\ast(t-t) = \phi(t, k) \]

and

\[ \sum_{i=i_1}^{i_2} u(i-k)d^\ast(i) = z(-k) \]

\( \phi(t, k) \) is called time-averaged autocorrelation and \( z(-k) \) is called time-averaged crosscorrelation.

Note that these are definitions and not estimates. In OSB the same notations were used both for ensemble-averaged and time-averaged correlation \( \{r_s(k)\text{ and } r_d(k)\} \).
Time-averaged autocorrelation, matrix form

The corresponding time-averaged crosscorrelation vector (compare p) is

\[ z = [z(0) \quad z(1) \quad \ldots \quad z(-M + 1)]^T \]

When minimizing \( E = \sum_{i=1}^{M} |e(i)|^2 \) the normal equations for the LS filter will be:

\[ \Phi \hat{w} = z \quad (\text{jmf } Rw = p) \]

which solution is

\[ \hat{w} = \Phi^{-1} z \quad ( = (A^H A)^{-1} A^H d ) . \]

Weighted LS filter

The first step towards creating an adaptive LS filter is to introduce a weighting factor in the minimization criteria (prewindowing)

\[ E(n) = \sum_{i=1}^{n} \beta(n, i) |e(i)|^2 , \]

where \( \beta(n, i) \) determines how much the error for sample \( i \) shall be weighted into the cost function at time \( n \).

The weight must be in the interval

\[ 0 < \beta(n, i) \leq 1 \]

and the most common choice of \( \beta(n, i) \) is

\[ \beta(n, i) = \lambda^{n-i} . \]

\( \lambda \) is called forgetting factor and the method is called exponentially weighted LS.

In the same way as earlier the correlation matrix, \( \Phi \), and crosscorrelation vector, \( z \), can now be defined:

\[ \Phi(n) = \sum_{i=1}^{n} \lambda^{n-i} u(i) u^H(i) \]

and

\[ z(n) = \sum_{i=1}^{n} \lambda^{n-i} u(i)d^*(i) , \]

i.e., with exponential weighting and forgetting factor.

The optimal filter at time \( n \), \( \hat{w}(n) \), based on data since sample 1 (exponentially weighted) can now be calculated from the time-dependent normal equation.

\[ \Phi(n) \hat{w}(n) = z(n) \]

as

\[ \hat{w}(n) = \Phi^{-1}(n) z(n) . \]

Problem: We cannot invert this big matrix for every new sample.
The last expression can be written

\[ w(n) = w(n-1) + k(n) \left( d(n) - w^H(n) u(n) \right) \]

The vector \( k(n) \) is often called the gain vector.

Convergence analysis, RLS

- The ensemble-averaged learning curve for the RLS converges in approximately 2\( M \) iterations. This is considerably faster than for the LMS.
- When the number of iterations goes to infinity, the RLS error converges to zero.
- No misadjustment.
- The convergence is independent of the eigenvalues of the correlation matrix of \( u(n) \).
## To read, Week 1

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1–4</td>
<td>General understanding</td>
</tr>
<tr>
<td>5</td>
<td>General understanding, (ej square-root, FastRLS)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Good understanding</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.1–2.7</td>
<td>Repetition, alt. Hayes</td>
</tr>
<tr>
<td>4</td>
<td>4.1–4.5</td>
<td>Good understanding of principles</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Exercises, computer exercises</td>
</tr>
</tbody>
</table>

## To read, Week 2

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5.1–5.2</td>
<td>Very important</td>
</tr>
<tr>
<td>5.3</td>
<td>Good understanding, (ej application 1, 3, 6)</td>
<td></td>
</tr>
<tr>
<td>5.4</td>
<td>The lecture substitutes pages 238–269. Pages 269–272 are important. Pages 273–278 is not included in the course</td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td>Very important except the part on small-step theory</td>
<td></td>
</tr>
<tr>
<td>5.6</td>
<td>Good understanding</td>
<td></td>
</tr>
<tr>
<td>5.7</td>
<td>Very important, the Equalizer example</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Exercises, computer exercises</td>
</tr>
</tbody>
</table>

## To read, Week 3

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5.9</td>
<td>Good understanding</td>
</tr>
<tr>
<td>5.13</td>
<td>Good understanding</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6.1</td>
<td>Understanding corresponding to the lecture. Extra material from lecture on the home page F3.</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Material in lecture notes which is not in the chapter like, Leaky-LMS, etc.</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Excitation of filter, hidden instability, see addition to F3.</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Echo canceller – general understanding and principles</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Exercises, computer exercises</td>
</tr>
</tbody>
</table>

## To read, Week 4

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Intro.</td>
<td>Very important</td>
</tr>
<tr>
<td>7.1</td>
<td>Important. Lecture notes cover the important part.</td>
<td></td>
</tr>
<tr>
<td>7.2</td>
<td>Important. Lecture notes are enough + the relationship between linear and circular convolution.</td>
<td></td>
</tr>
<tr>
<td>7.3</td>
<td>Good understanding, Lecture notes are enough.</td>
<td></td>
</tr>
<tr>
<td>7.4</td>
<td>Lecture notes are enough.</td>
<td></td>
</tr>
<tr>
<td>7.5</td>
<td>General understanding</td>
<td></td>
</tr>
<tr>
<td>7.7–7.8</td>
<td>General understanding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>All block diagrams, see addition to F4 on the web.</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Exercises, computer exercises, laboration 1 (parameter drift)</td>
</tr>
</tbody>
</table>

### To read, Week 5

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>8.1–8.4</td>
<td>Good understanding, jmf OSB</td>
</tr>
<tr>
<td></td>
<td>8.5</td>
<td>Important</td>
</tr>
<tr>
<td></td>
<td>8.6</td>
<td>Good understanding</td>
</tr>
<tr>
<td>9</td>
<td>9.1–9.3</td>
<td>Very important. Lecture covers the important part.</td>
</tr>
<tr>
<td></td>
<td>9.8</td>
<td>Very important</td>
</tr>
<tr>
<td></td>
<td>9.10</td>
<td>Overview</td>
</tr>
<tr>
<td>+</td>
<td></td>
<td>Exercises, computer exercises, laboratory 2</td>
</tr>
</tbody>
</table>

### To read, Week 6

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Inled. + 13.1–13.2</td>
<td>Overview, (ej parameterdrift). The important parts are in the lecture notes.</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>Inled. + 14.1</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Parameterdrift, see excitation of filters in the add to F3.</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Exercises, computer exercises</td>
</tr>
</tbody>
</table>

### To read, Week 1

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intro.</td>
<td>1–3</td>
<td>General understanding</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>General understanding, (ej square-root, FastRLS)</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Good understanding, (ej lattice, blind eq., beamforming)</td>
</tr>
<tr>
<td>5</td>
<td>5.1–5.7</td>
<td>Repetition, alt. Hayes</td>
</tr>
<tr>
<td>8</td>
<td>8.1–8.5</td>
<td>Good principiell understanding</td>
</tr>
<tr>
<td>+</td>
<td></td>
<td>Exercises, computer exercises</td>
</tr>
</tbody>
</table>

### To read, Week 2

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>9.1–9.2</td>
<td>Mycket viktigt</td>
</tr>
<tr>
<td></td>
<td>9.3</td>
<td>Good understanding, (ej exempel 1, 3, 6)</td>
</tr>
<tr>
<td></td>
<td>9.5</td>
<td>Mycket viktigt</td>
</tr>
<tr>
<td></td>
<td>9.6</td>
<td>Good understanding</td>
</tr>
<tr>
<td></td>
<td>9.7</td>
<td>Mycket viktigt, Equalizer-exemplet</td>
</tr>
<tr>
<td>+</td>
<td></td>
<td>Exercises, computer exercises</td>
</tr>
</tbody>
</table>
### To read, Week 3

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>9.9</td>
<td>Good understanding</td>
</tr>
<tr>
<td></td>
<td>9.11</td>
<td>understanding motsvarande genomgång på föreläsningen. Finns på nätet som tillägg till F3.</td>
</tr>
<tr>
<td></td>
<td>9.12</td>
<td>Viktigt</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Material i föreläsningsanteckningarna som ej finns i ovan nämnda Chapter, Leaky-LMS, etc.</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Excitering av filter, dold instabilitet, se tillägg till F3.</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Ekosläckare – allmän understanding och princip</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Exercises, computer exercises</td>
</tr>
</tbody>
</table>

### To read, Week 4

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Inled.</td>
<td>Mycket viktigt</td>
</tr>
<tr>
<td></td>
<td>10.1</td>
<td>Viktigt, lite röligt: M skall vara L i ekv. (10.9) och (10.11). Fel i ekv. (10.20). Föreläsningsanteckningarna räcker.</td>
</tr>
<tr>
<td></td>
<td>10.2</td>
<td>Viktigt. Föreläsningsanteckningarna räcker – linjär resp. cirkulär faltning, samt overlap-save.</td>
</tr>
<tr>
<td></td>
<td>10.3</td>
<td>Good understanding, föreläsningsanteckningarna räcker.</td>
</tr>
<tr>
<td></td>
<td>10.4</td>
<td>Kursivt, föreläsningsanteckningarna räcker.</td>
</tr>
<tr>
<td></td>
<td>10.5-10.6</td>
<td>General understanding</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Olika strukturer, se tillägg till F4 på nätet.</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Exercises, computer exercises, laboration 1 (parameterdrift)</td>
</tr>
</tbody>
</table>

### To read, Week 5

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>11.1–11.4</td>
<td>General understanding, jmf OSB</td>
</tr>
<tr>
<td></td>
<td>11.5</td>
<td>Viktigt</td>
</tr>
<tr>
<td></td>
<td>11.6</td>
<td>General understanding</td>
</tr>
<tr>
<td></td>
<td>13.7</td>
<td>Mycket viktigt</td>
</tr>
<tr>
<td></td>
<td>13.9</td>
<td>Generalt</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Exercises, computer exercises, laboration 2</td>
</tr>
</tbody>
</table>

### To read, Week 6

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Inled.</td>
<td>Föreläsningsanteckningarna räcker för övrigt.</td>
</tr>
<tr>
<td>17</td>
<td>17.1–17.2</td>
<td>Generalt, (ej parameterdrift). Det viktiga finns i föreläsningsanteckningarna.</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Parameterdrift, se excitering av filter i tillägg till F3.</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>Exercises, computer exercises</td>
</tr>
</tbody>
</table>
Exam

Time and place:

Thursday 16/12 14.00-19.00, in room Sparta C.

The following is allowed to bring:

- Course book (Haykin). Dock ej anteckningar med lösningar till övningsuppgifter.
- Course book from OSB (Hayes). Dock ej anteckningar med lösningar till övningsuppgifter.
- Formulas (typ TeFyMa och Beta).
- Formulas from basic course.
- Calculator.

Adaptive Signal Processing 2011 Exam information. Lecture 7