Analysis and Design of Radial Basis Function-Based Turbo Equalizers

Hasan Abdulkader, Bouchra Benammar, Charly Poulliat, Marie-Laure Boucheret and Nathalie Thomas

University of Toulouse, IRIT-ENSEEIHT-INPT, Toulouse, France

August 18-22, 2014, Bremen, Germany.
1. Problem statement and motivations
   - Discrete communication model
   - Equalizer description
   - Motivations

2. Equalizer parameter

3. Simulation settings
   - Simulation results

4. Open questions
   - Blind channel learning

5. Conclusion
Satellite communication system

Channel modelling

- Weiner-Hammerstein model [Bershad et al. 2000]
- Volterra series model [Benedetto et al. 1979]
- Hybrid model [Ibnkahla et al. 1997]
- ...
Discrete communication model

\[ y(k) = h_0 x(k) + \sum_{i=1}^{L-1} h_i x(k-i) + \sum_{i,j,l=0}^{L-1} h_{i,j,l} x(k-i)x(k-j)x^*(k-l) + n(k) \]

\( L \) is the channel depth
Neural network equalizers

- Almost all types of NN have been successfully used as SIHO equalizers [Burse et al. 2010]
- RBF has been implemented as SISO equalizer, it produced the Bayes rule [Chen et al. 1993]
- RBF has been implemented in Turbo equalization for linear channels [Yee et al. 2003]
- Here we apply RBF-TEQ in non linear communication channels.
Radial Basis Function equalizer

\[
\begin{align*}
\hat{x}(k-d) &= \sum_{\phi} w_v \phi(y, \bar{y}) \\
N_h &= M^{L+m-1} \\
\phi(y, \bar{y}) &= \exp \left( -\frac{\|y - \bar{y}\|^2}{2\sigma^2} \right)
\end{align*}
\]

Advantages of RBF equalizer

- Can be unified with the Bayes optimal classification rule
- Allows Distributed computation (Parallel computing)
- Allows a smart control of computational complexity
- Applies Symbol by symbol EQ. instead of sequence detection

\[ m\text{-dimensional input} \]
From Bayes to RBF

\[ p(\bar{y}|y) = p(y|\bar{y}) \cdot p(\bar{y}) \]

\[ p(x|y) = p(y|\bar{y}) \cdot p(x) \]

\[ p(x(k-d) = s_i|y) = p(s_i) \sum_{\tilde{x}} p(y|\bar{y}) \cdot p(\tilde{x}) \]

\[ \bar{y} = [\bar{y}(k), ..., \bar{y}(k - m + 1)]^T \]

\[ x = [x(k), ..., x(k - d), ..., x(k - L - m + 2)]^T \]

\[ \tilde{x} = x \setminus x(k - d) \]
From Bayes to RBF

\[
p(\vec{y}|\vec{y}) = p(y|\vec{y})p(\vec{y})
\]

\[
p(x|y) = p(y|\vec{y})p(x)
\]

\[
p(x(k - d)|y) = p(s_i) \sum_{\tilde{x}} p(y|\vec{y})p(\tilde{x})
\]

\[
\vec{y} = [\vec{y}(k), \ldots, \vec{y}(k - m + 1)]^T
\]

\[
x = [x(k), \ldots, x(k - d), \ldots, x(k - L - m + 2)]^T
\]

\[
\tilde{x} = x \setminus x(k - d)
\]

\[
y_{RBF} = \sum_j \phi(y, \vec{y}_j)w_j
\]

\[
\phi(y, \vec{y}) = e^{-\frac{||y - \vec{y}||^2}{2\sigma^2}}
\]
RBF as a SISO equalizer, Parallel implementation [Yee]

\[ y(k) \]
\[ Z^{-1} \]
\[ Z^{-1} \]
\[ \vdots \]
\[ Z^{-1} \]

\[ \phi(., \bar{y}_1) \]
\[ \phi(., \bar{y}_{n_h}) \]
\[ \phi(., \bar{y}_1) \]
\[ \phi(., \bar{y}_{n_h}) \]

\[ w_1 \]
\[ w_{n_h} \]
\[ w_1 \]
\[ w_{n_h} \]

\[ p(x|k-d=s_1|y) \]
\[ p(x|k-d=s_M|y) \]

\[ n_h = M^{L+m-2} \]

\[ s_1, \ldots, s_M \text{ symbols of the constellation} \]

- \( \bar{y} \) channel state vector is the center of the activation function
- \( w_i = \prod_{j=0}^{L+m-2} p(x_i(k-j)) \)
Soft Mapping and Demapping

Mapping

\[ L_{a1}(c_i) \rightarrow p(c_i = 0) \& p(c_i = 1) \]

\[ p(x(k)) = \prod_{u=0}^{n_m-1} p(c_i (n_m.k - u)); \text{where } n_m = \log_2 M \]
### Soft Mapping and Demapping

#### Mapping

\[ L_{a1}(c_i) \rightarrow p(c_i = 0)\& p(c_i = 1) \]

\[ p(x(k)) = \prod_{u=0}^{n_m-1} p(c_i(n_m.k - u)) \text{; where } n_m = \log_2 M \]

#### Demapping

\[ L_{e1}(c_i) = \ln p(c_i = 0|y) - \ln p(c_i = 1|y) - L_{a1}(c_i) \]

\[ p(c_i = 0|y) = \sum_{j=1}^{n_m} p(x(n - d) = s_j|y) \delta_i^j \]

Or **generalized Jacobian alg.** [Yee et al. 2003], \( J(\ldots, J(\theta_3, J(\theta_2, \theta_1))) \)

with \( J(\theta_2, \theta_1) = \ln \left( e^{\theta_2} + e^{\theta_1} \right) = \max(\theta_2, \theta_1) + \ln \left( 1 + e^{-|\theta_2 - \theta_1|} \right) \)

\[ \ln(p(c_i = 0|y) = \ln \left( \sum_{j=1}^{n_m} \left( \sum_{l=1}^{n_h} e^{-\frac{||y - y_l||^2}{2\sigma^2}} e^{\ln(p(x_l))} \right) \delta_i^j \right) \]
Equalizer parameters: \( m, d \) and \( FB \)

\( FB \): Nber of estimated symbols used as feedback

\[
\bar{y}_v(k) = f(h, x_v)
\]

\[
\begin{align*}
x_v &= [x(k), \ldots, x(k - d), \ldots, x(k - L - m + 2)]_v^T \\
\end{align*}
\]

Estimated symbols at the equalizer output form a vector:

\[
\hat{x}_{FB} = [\hat{x}(k - d - 1), \ldots, \hat{x}(k - L - m + 2)]^T
\]
Equalizer parameters: \( m, d \) and \( FB \)

**FB**: Number of estimated symbols used as feedback

\[
\bar{y}_v(k) = f(h, x_v)
\]

\[
x_v = [x(k), ..., x(k - d), ..., x(k - L - m + 2)]^T
\]

Estimated symbols at the equalizer output form a vector:

\[
\hat{x}_{FB} = [\hat{x}(k - d - 1), ..., \hat{x}(k - L - m + 2)]^T
\]

A subset of the RBF centers is defined by the FedBack symbols

\[
\bar{y}_{sub-v}(k) = f\left(h, [x(k), ..., x(k - d - 1), ..., ..., x(k - L - m + 2)]_v \right)
\]

Symbols identified by FB

\[\text{maxFB} = L + m - d - 2\]
Equalizer parameters: \( m, d \) and \( FB \)

**FB**: Number of estimated symbols used as feedback

\[
\bar{y}_v(k) = f(h, x_v)
\]

\[
x_v = [x(k), ..., x(k - d), ..., x(k - L - m + 2)]^T_v
\]

Estimated symbols at the equalizer output form a vector:

\[
\hat{x}_{FB} = [\hat{x}(k - d - 1), ..., \hat{x}(k - L - m + 2)]^T
\]

A subset of the RBF centers is defined by the FedBack symbols

\[
\bar{y}_{sub-v}(k) = f \left( h, \left[ x(k), ..., x(k - d - 1), ..., ..., x(k - L - m + 2) \right]_v \right)
\]

This allows a complexity reduction, where the subset size \( n_s \) is:

\[
n_s = M^{L+m-FB-2} \leq n_h = M^{L+m-2}
\]
Motivation

- $d = 2$
- Full complexity $FB = 0$
- Reduced complexity $FB = 2$
- MFB Matched filter bound
- Reduced complexity performs better ...!! (UNEXPECTED)
Motivation

LITERATURE: FACTS and COUNTERFACTS

- NNs may suffer from an **OVERFITTING** phenomenon ...

- NNs trained by Bayes rule should not show an overfitting ...
Motivation

Objectives

1. Explain the RBF behaviour and prove the overfitting
2. Find the optimal operation point (parameters) of the equalizer
System description:

ECC $CC(7, 5)$, Mapper $8-PSK$, Equalizer memory $m = 3$ and Volterra channel with $L = 4$ [Ampeliotis et al. 2008]

Decoder: BCJR algorithm

TEQ: generalized Jacobian algorithm
Simulation parameters

**System description:**

ECC \( CC(7, 5) \), Mapper \( 8 - PSK \), Equalizer memory \( m = 3 \) and Volterra channel with \( L = 4 \) [Ampeliotis et al. 2008]

Decoder: BCJR algorithm

TEQ: generalized Jacobian algorithm

**Variable parameters**

- \( FB \) varies between 0 and \( maxFB \)
- \( d \) varies between 0 and \( L + m - 2 \)

**Methodology**

- EXIT chart [Ten Brink 2001] and achievable rate are computed for all cases
EXIT chart

Performance versus $FB$ when $d = 0$

![Graph showing performance versus $FB$ for $d = 0$.](image1)

Performance versus $FB$ when $d = 2$

![Graph showing performance versus $FB$ for $d = 2$.](image2)
Influence of $FB$ (for $d = 0$ and $E_b/N_0 = 3dB$)

Optimal performance when $FB = 3$
Influence of $d$ ($Eb/N0 = 6dB$ and $I_A = 0.5$)

Performance deterioration when $d > 2$
Achievable rate

\[ \text{Ach} - \text{Rate} = \int_{0}^{1} I_{E} \cdot dI_{A} \]

\( d \geq 2 \) leads to performance deterioration
Achievable rate function of $FB$ (for $d = 0$)

Optimal performance when $FB = 3$
Open questions

How should the channel model be accurate?

1. Reduced complexity allowed better performance
2. Overfitting depends on RBF size and the channel

- Can a sub-optimal estimation of the CIR be sufficient in TEQ?
- Is there a better criterion to measure the model accuracy other than MSE?
- Is it possible to estimate the best CIR inside the receiver?
Blind channel learning

- Problem statement
- Equalizer parameters
- Simulation
- Corollary
- Conclusion
- References

![Diagram of blind channel learning](image)

- MLP
- Soft symbol detection
- Equalizer
- Decoder
- $\Pi$
- $L_{e1}$
- $L_{e2}$
Blind equalization \((d = 2, \, FB = 2)\)

Blind RBF-TEQ achieves the MFB within a few iterations
Conclusions

- A framework to design and optimize RBF-Based turbo equalizers is presented.
- Computational complexity exponentially depends:
  - in direct proportion on channel depth and equalizer memory
  - in inverse proportion on feedback parameter
- Delay $d$ and feedback $FB$ influence the performance.
- Overfitting phenomenon is observed.
-Controlling $d$ and $FB$ allows to overcome the overfitting and to optimize the performance.
- The model of the channel should take into account the overfitting.
- RBF equalizer can be implemented in blind mode.
<table>
<thead>
<tr>
<th>Authors</th>
<th>Title</th>
</tr>
</thead>
</table>
Thank you for your attention