Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: http://oatao.univ-toulouse.fr/
Eprints ID: 13116

To link to this article: DOI: 10.1109/ISTC.2014.6955110
URL: http://dx.doi.org/10.1109/ISTC.2014.6955110


Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr
On Linear Frequency Domain Turbo-Equalization Of Non Linear Volterra Channels

Bouchra Benammar*, Nathalie Thomas*, Charly Poulliat*, Marie-Laure Boucheret*, Mathieu Dervin†
*University of Toulouse, INPT-ENSEEIHT/IRIT, Toulouse France
†Thales Alenia Space, Toulouse, France
{bouchra.benammar, nathalie.thomas, charly.poulliat, marie-laure.boucheret}@enseeih.fr
{mathieu.dervin@thalesaleniaspace.com}

Abstract—This article deals with iterative Frequency Domain Equalization (FDE) for Single Carrier (SC) transmissions over Volterra non linear satellite channels. SC-FDE has gained much importance in recent research for its efficient implementation at the receiver and its interesting low Peak to Average Power Ratio (PAPR) at the transmitter. However, nearly saturated power amplifiers on board satellites generate linear and non linear Inter Symbol Interference (ISI) at the receiver. It is thus interesting to investigate the implementation of SC-FDE for non linear channels. To do so, a frequency domain equivalent satellite channel is derived based on the time domain Volterra series representation of the non linear channel. Then a Minimum Mean Square Error (MMSE)-based iterative frequency domain equalizer is designed. It is shown that the proposed equalizer consists of a Soft Interference Canceller (SIC) which subtracts both the linear and non-linear soft frequency symbols. The equalizer performance is then compared to the equivalent time domain implementation. Results show that a channel-memory independent efficient implementation is achieved at the price of a negligible spectral efficiency loss due to cyclic prefix insertion.

Keywords—iterative equalization, volterra series, non linear interference.

I. INTRODUCTION

The increasing demand for high data rates urges new communication systems to provide both high spectral and power efficiency. In recent satellite communications standards, this trade off can be achieved using multi-level amplitude constellations such as Amplitude and Phase Shift Keying (APSK). However, such modulations increase the signal fluctuations and thus, give rise to non linear distortions when being passed through a High Power Amplifier (HPA) operating near saturation. These distortions can be either mitigated at the transmitter using signal or symbol pre-distortion, or at the receiver using equalization. To do so, analytical expressions for the non linear channel have been widely investigated [1], [2] and [3]. [1] and [2] use a triple-effect representation of the non linear channel, consisting of additive noise, a scalar multiplication and warping. This approximation, although practical, is not accurate for higher order modulations since the additive noise can not be considered circular for outer ring symbols. Another category of channel models consists of a decomposition of the non linear channel by means of Taylor series leading to the so called Volterra representation. The Volterra series representation has been applied to different transmission channels, namely in [4] for the non linear magnetic channel and to differential impulse radio UWB systems in [5]. As far as satellite channels are concerned, a Volterra series expansion was derived in the leading work of [6]. Such a decomposition depends on the chain filters, the HPA model, the modulation type and the Input Back Off (IBO). Based on this Volterra series expansion, the satellite channel can be represented as a finite state machine, which allows for optimal symbol and sequence detection. Maximum A Posteriori (MAP) and Maximum Likelihood Sequence Detection MLSD. However, as the complexity of these optimal equalizers is exponential in the channel memory, low complexity equalizers have been investigated among which we will focus on iterative solutions. Indeed, new equalization trends are based on iterative receivers for their excellent Shannon-bound approaching abilities [7], [8], and [9]. In [10], we presented an iterative time domain MMSE-based equalizer to mitigate the channel linear and non linear interference. Different low complexity approximations were investigated. The performances showed that a No-Apriori approximation has minor performance degradation but far less complexity than the exact MMSE solution. In this paper, we are interested in further reducing the computational complexity by applying frequency domain equalization. Previous studies have derived non iterative frequency domain equalization for non linear channels [11] [12]. In [11] an adaptive block Least Mean Square (LMS) equalizer is proposed and a complexity comparison shows the advantage of using frequency rather than time domain equalization. In this work, we use the frequency domain Volterra channel model, to derive an iterative linear MMSE-FDE. As argued for the time domain solution in [10], time domain linear equalization is able to cope with both linear and non linear interference when used in an iterative way. Similarly, the proposed iterative FDE is able to cancel both linear and non linear soft symbols which asymptotically approaches the Additive White Gaussian Noise (AWGN) ISI-free performance. The remainder of this paper is organised as follows. First, we present the frequency domain Volterra channel model. In Section III, we derive the iterative linear MMSE-FDE based on the no-Apriori MMSE approximation. Finally, we investigate the performance of the MMSE equalizer implementations before ending with some conclusions.
II. USEFUL NOTATIONS AND RESULTS ON CIRCULANT MATRICES

Let us introduce some useful notations and results. Vectors are written in bold letters and matrices in capital letters. \( \mathbb{I}_m \) stands for the identity matrix of size \( m \) and \( \mathbb{1}_{1 \times m} \) for the all ones vector of size \( 1 \times m \). \((.\)\)\(_m\) refers to the modulo-
\( m \) operator. Let \( F \) be the normalised 1-Dimensional Discrete Fourier Transform (1D-DFT) matrix of size \( N \). The matrix element \( F_{i,j} \) is:

\[
F_{i,j} = \frac{1}{\sqrt{N}} W_{N}^{-ij} \quad (1)
\]

where \( W_{N}^{k} = e^{\frac{-j \pi k}{N}} \). We then have \( F^H = F^{-1} \). Let \( H \) and \( G \) be two circulant matrices of size \( (N \times N) \). A circulant matrix can be diagonalised using the DFT matrix \( F \). More precisely:

\[
H = F^H H_d F \quad (2)
\]

\[
G = F^H G_d F
\]

where the notation \( H_d \) and \( G_d \) stands for the diagonal matrix containing the eigenvalues of \( H \) and \( G \) respectively. \( H \) and \( G \) satisfy the following properties:

1) \( HG \) is a circulant matrix and \( HG = F^H H_d G_d F \).
2) \( H^H \) is a circulant matrix \( H^H = F^H H_d^H F \).
3) \( H^{-1} \) is a circulant matrix and \( H^{-1} = F^H H_d^{-1} F \).
4) \( \forall (n, m) \in \{0, \ldots , N - 1\}^2 \), \( u_m^T H u_m = u_n^T H u_n \)

where \( u_m = [0, 1, \mathbb{0}_{N-m-1}]^T \)

The N-3D normalised DFT of 3D symbols \( y_{m,n,l}^{(3)} \) for \( m, n, l \in \{0, \ldots , N - 1\} \) is:

\[
y_{p,q,r}^{(3)} = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \sum_{l=0}^{N-1} y_{m,n,l}^{(3)} W_{N}^{-mp} W_{N}^{-nq} W_{N}^{-lr} \quad (3)
\]

Similarly, the N-3D normalised IDFT of 3D frequency symbols \( y_{p,q,r}^{(3)} \) writes as follows:

\[
y_{m,n,l}^{(3)} = \frac{1}{\sqrt{N}} \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} \sum_{r=0}^{N-1} Y_{p,q,r}^{(3)} W_{N}^{mp} W_{N}^{nq} W_{N}^{lr} \quad (4)
\]

III. FREQUENCY DOMAIN NON-LINEAR VOLterra CHANNEL MODEL

A satellite High Power Amplifier (HPA) is a memoryless device characterized by two frequency independent functions [13] \( A(.) \) (AM/AM) and \( \Phi(.) \) (AM/PM) relating the power and phase of the output signal \( y \) to the amplitude of the input signal \( x \) as follows:

\[
y = c(|x|) \exp(j \Phi(x)) = A(|x|) \exp(j \Phi(|x|) + \varphi(x)) \quad (5)
\]

On-board reception resp. transmission filters introduce memory in the overall satellite transponder response which is expressed at the receiver by a Volterra odd-series decomposition of the received symbols \( z_n \) in the form [6]:

\[
z_n = \sum_{m=0}^{v} \sum_{n=0}^{\infty} \sum_{n_{m+1}} \ldots \sum_{n_{m+1}} \hat{x}_{n-n_{m+1}} \ldots \hat{x}_{n-n_{m+1}} + w_n \quad (6)
\]

where \( v \) defines the decomposition order of the Volterra series, \( h_{n_{m+1}, \ldots , n_{m+1}} \) are called the Volterra kernels and \( w_n \) is the filtered sampled additive noise with variance \( \sigma_n^2 \). The overall system can be described by Fig. 1. An example of Volterra decomposition of a satellite transponder with butterworth on-board filters is given in [6]. This shows the negligible contribution of orders higher than three. Hence, this has been selected as the order of the Volterra series decomposition in this paper.

\[
z_n = \sum_{i=0}^{M-1} h_i \hat{x}_{n-i} + \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} h_{ijk} \hat{x}_{n-i-j} \hat{x}_{n-k} + w_n \quad (7)
\]

where \( M \) is the memory order. After removing the cyclic prefix, the received samples in (7) can be expressed as follows:

\[
z_n = \sum_{i=0}^{M-1} h_i x_{(n-i)M} + \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} h_{ijk} x_{(n-i)M} x_{(n-j)M} x_{(n-k)M} + w_n
\]

\[
z_n = z_n^{(1)} + z_n^{(3)} + w_n \quad (8)
\]

The RHS of (8) consists of a sum of both a circular convolution \( z_n^{(1)} \) and a third order circular convolution \( z_n^{(3)} \) expressed as follows:

\[
z_n^{(1)} = \sum_{i=0}^{M-1} h_i x_{(n-i)M} \quad (9)
\]

\[
z_n^{(3)} = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} h_{ijk} x_{(n-i)M} x_{(n-j)M} x_{(n-k)M} \quad (10)
\]

In the frequency domain, the 1D-DFT of the circular convolution in (9) translates into an element-wise multiplication of the 1D-DFT linear filter coefficients.
can be shown that the 3D-DFT of symbols can thus be written as follows:

\[ Z_{m,n,l}^{(1)} = H_d(m)X_m \]  

where \( 0 \leq m \leq N-1 \). For general values of \((m,n,r)\), it can be shown that the 3D-DFT of \( z_{m,n,l} \) is [14]:

\[ Z_{p,q,r}^{(3)} = \sqrt{N^3} H_{p,q,r}^{(3)} X_p X_q X_r \]  

where \( X \) is the N 1D-DFT of conjugate symbols \( x_n^* \) and \( H_{p,q,r}^{(3)} \) are the N-3D-DFT of triplets \( h_{i,j,k} \). Equivalently for \( m = n = l \), the time domain symbols \( z_{m,n,n}^{(3)} \) can be expressed as:

\[ z_{m,n,n}^{(3)} = \frac{1}{\sqrt{N^3}} \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} \sum_{r=0}^{N-1} H_{p,q,r}^{(3)} X_p X_q X_r \]  

where the delta-function modulo \( N \) is defined as follows:

\[ \delta_N(p) = 1 \text{ if } (m)_N = 0 \]  

Equation (13) shows that \( z_{m,n,n}^{(3)} \) is the mono-dimensional N-1D-IDFT of a combination of N-3D-DFT Volterra kernels. The non linear interference can thus be projected on a 1D-DFT instead of a 3D-DFT. The \( m \)th N-1D-DFT output of the received symbols can thus be written as follows:

\[ z_m = H_d(m)X_m + \sqrt{N} \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} \sum_{r=0}^{N-1} H_{p,q,r}^{(3)} X_p X_q X_r \]  

It should be noted, that triplets \((p,q,r)\) satisfying \( \delta_N(p+q+r-m) = 1 \) are disjoint for different frequency indexes \( m \).

IV. Linear Frequency Domain Volterra-MMSE Turbo Equalization

In this section, we derive the frequency domain MMSE equalizer. Let us define the following notations:

\[ \mathbf{z} = [z_0, \ldots, z_{N-1}]^T, \quad \mathbf{x} = [x_0, \ldots, x_{N-1}]^T, \quad \mathbf{w} = [w_0, \ldots, w_{N-1}]^T \]

It follows that:

\[ \mathbf{z} = \mathbf{Hx} + \sum_{i} \sum_{j} \sum_{k} H_{ijk} \mathbf{x}^{ijk} + \mathbf{w} \]

where \( \mathbf{H} \triangleq \text{Circ}([h_{0,0,1} \times_{N-1} h_{0,M-1} \ldots h_{1,1}]) \) is the circular convolution matrix having \([h_{0,0,1} \times_{N-1} h_{0,M-1} \ldots h_{1,1}]\) at its first row; \( H_{ijk} = h_{i,j,k} \) are the non linear ISI matrices and \( \mathbf{x}^{ijk} \) are the third order ISI terms:

\[ \mathbf{x}^{ijk} = \begin{bmatrix} x_{(0-i),m} x_{(0-j),m} x_{(0-k),m}^* \\ x_{(N-1-i),m} x_{(N-1-j),m} x_{(N-1-k),m}^* \end{bmatrix} \]

A. Turbo MMSE FDE

The linear MMSE equalizer consists of an affine estimation of the received symbols [15]:

\[ \hat{x}_n = a_n^H \mathbf{z} + b_n \]  

which computes estimates that minimize the mean square error with the transmitted symbols \( E[|x_n|^2] \). The time variant MMSE coefficients are given as follows:

\[ \begin{align*}
    a_n &= \text{Cov}(\mathbf{z}, \mathbf{z})^{-1} \text{Cov}(\mathbf{z}, \mathbf{x}_n) \\
    b_n &= E[\mathbf{x}_n] - a_n^H E[\mathbf{z}]
\end{align*} \]  

where \( \text{Cov}(\mathbf{x}, \mathbf{y}) \triangleq E[\mathbf{xy}^H] - E[\mathbf{x}] E[\mathbf{y}^H] \), leading to:

\[ \hat{x}_n = a_n^H (\mathbf{z} - E[\mathbf{z}]) + E[\mathbf{x}_n] \]

In order to cancel all the linear and non linear interference, the contribution of the symbol \( x_n \) is subtracted only for the linear terms. This allows for a perfect reconstruction of the non linear soft interference and thus its full cancellation at perfect priors. The estimated symbols can thus be expressed as:

\[ \hat{x}_n = a_n^H (\mathbf{z} - E[\mathbf{z}]) + b_n H \mathbf{u}_n E[\mathbf{x}_n] \]

As discussed in [10], the time varying exact MMSE implementation is computationally prohibitive. This encourages investigating some low complexity implementations, among which we select the No-Apriori implementation. Recalling the results in [10], the MMSE solution writes with the NA approximation as:

\[ C_{ZZ} \triangleq \text{Cov}(\mathbf{z}, \mathbf{z}) = \sigma_w^2 \mathbf{1}_N + HH^H + \sum_{i,j,k} |h_{ijk}|^2 \mathbf{1}_N \]

\[ C_{Zx_n} \triangleq \text{Cov}(\mathbf{z}, \mathbf{x}_n) = H \mathbf{u}_n \]

The non linear interference appears as additive white Gaussian noise with variance \( \sigma_w^2 = \sum_{i,j,k} |h_{ijk}|^2 \). Thus, by defining the overall noise variance, \( \sigma_w^2 = \sigma_t^2 + \sigma_n^2 \),

\[ C_{ZZ} = \sigma_w^2 \mathbf{1}_N + HH^H \]

The NA-MMSE solution can thus be written as:

\[ \hat{x}_n = u_n^T H^H C_{ZZ}^{-1} (\mathbf{z} - E[\mathbf{z}]) + u_n^T H^H C_{Zx_n} \mathbf{u}_n E[\mathbf{x}_n] \]

Using the results in Section II, the term \( C = u_n^T H^H C_{ZZ}^{-1} H \mathbf{u}_n \) is constant \( \forall n \in \{0, \ldots, N-1\} \), and thus,
the estimated symbols $\hat{x} = [\hat{x}_0, \ldots, \hat{x}_{N-1}]^T$ can be written in a compacted form as follows:
\[
\hat{x} = H^H C_{ZZ}^{-1}(x - E[x]) + CE[x]
\]
(21)

The computation of these filters can be done efficiently in the frequency domain. To do so, $C_{ZZ}^{-1}$ is computed as follows:
\[
C_{ZZ}^{-1} = (F^H (\sigma_w^2 I_N + H_d H_d^H) F)^{-1}
\]
\[
= F^H C_{ZZ,d}^{-1} F
\]
(22)

where $C_{ZZ,d} = \sigma_w^2 I_N + H_d H_d^H$ and the inverse is obtained using the results of Section II. It follows that the frequency estimated symbols are:
\[
\hat{x} \triangleq F \hat{x} = H_d^H C_{ZZ,d}^{-1}(Z - E[Z]) + CE[X]
\]
(23)

where $E[Z] = FE[x]$ and $E[X] = FE[x]$. The $i$th soft frequency symbol $E[Z_i]$ is expressed as follows:
\[
E[Z_i] = H_d(i)E[X_i] + \sqrt{\mathcal{N}} \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} \sum_{r=0}^{N-1} H_{p,q,r}^{(3)}
\]
\[
\times E[X_p x_q x_r] \delta(N(p + q + r - i))
\]
(24)

The constant $C$ can be computed using $u_0$ as:
\[
C = u_0^T H^H C_{ZZ}^{-1} H u_0 = u_0^T F^H H_d^H C_{ZZ,d}^{-1} H_d F u_0
\]
\[
= \frac{1}{N} \sum_{i=0}^{N-1} \frac{|H_d(i)|^2}{\sigma_w^2 + |H_d(i)|^2}
\]
(25)

where the factor $\frac{1}{N}$ comes from $u_0^T H^H = \frac{1}{N} 1_{1 \times N}$. Thus, the computation of the frequency domain equalizer yields the following estimates:
\[
\hat{x}_i = \frac{H_d(i)}{\sigma_w^2 + |H_d(i)|^2} Z_i + \left( C - \frac{|H_d(i)|^2}{\sigma_w^2 + |H_d(i)|^2} \right) E[X_i]
\]
\[
- \frac{H_d(i)^*}{\sigma_w^2 + |H_d(i)|^2} \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} \sum_{r=0}^{N-1} H_{p,q,r}^{(3)} E[X_p x_q x_r]
\]
\[
\times \sqrt{\mathcal{N}} \delta(N(p + q + r - i))
\]
(26)

We recognise in (26) and Fig. 2 the structure of a soft interference canceller where linear ISI $E[X_i]$ and non linear ISI $E[X_p x_q x_r]$ terms appearing in $E[Z_i]$ are cancelled. To avoid the complexity of computing third order non linear interference associated with $E[X_p x_q x_r]$, we can compute the non linear interference in the time domain and by means of DFT, compute the non linear equivalent frequency interference. More specifically, let:
\[
E[Z^{(3)}] = \sqrt{\mathcal{N}} \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} \sum_{r=0}^{N-1} H_{p,q,r}^{(3)} E[X_p x_q x_r]
\]
\[
\Delta_N(p + q + r) = F \sum_k \sum_j H_{ijk} E[x^{(jk)}]
\]
(27)

where we define $\Delta_N(p + q + r) = [\delta_N(p + q + r - 0), \ldots, \delta_N(p + q + r - N - 1)]^T$. Computing soft non linear symbols is then less computationally complex.

To obtain the MMSE coefficients one needs to compute expectations of $E[x^{(jk)}]$ and $E[x]$ i.e. expectation of products of three symbols and symbol conjugates at different time instants as mentioned in [6]. Due to the presence of an interleaver between the decoder and the equalizer, symbols can be considered mutually independent. Thus, we can write the average of a product of $p$ symbols and $q-p$ symbol conjugates as follows:
\[
E \left[ x_{n-i} x_{n-i-1} \ldots x_{n-i-p} x_{n-i-p+1} \ldots x_{n-i-q} \right]
\]
\[
= \prod_j \sum_{m=1}^{M} s_{n_i}^{(j)} s_{m}^{(j)} P \left[ x_{n-i} = s_{m} \right]
\]
(28)

where $v_j$ ($v_j^*$) represents the number of occurrences of symbol $x_{n-i}$ ($x_{n-i}^*$) in the product average, and $s_m$ $m \in [1 \ldots M]$ is the $m$th constellation symbol with probability computed as:
\[
P \left( x_{n-i} = s_{m} \right) = \frac{\log_2(M)}{i=1} P \left( e_{n,i} = s_{m,i} \right)
\]

where $c_{n,i}$ is the $i$th coded bit of the symbol $x_n$ and $s_{m,i}$ is the $i$th bit of constellation symbol $s_m$.

The coded bit probabilities can be computed from the input Log-Likelihood Ratios (LLRs) $L_n$ from the decoder as: $L_n (e_{n,i}) = \ln \frac{P(\kappa_{n,i}=0)}{P(\kappa_{n,i}=1)}$.

**B. Soft demapper**

In order to map the output of the equalizer to code LLRs, we define the residual equalizer output error $\epsilon_n = \hat{x}_n - \kappa_n x_n$. Computing the distribution of the estimation error instead of the distribution of $\hat{x}_n$ given $x_n$ turns out to be a practical choice, since one needs not track occurrences of $x_n$ in third order covariances. For practical considerations, this error is assumed to be Gaussian. More precisely:
\[
\kappa_n = \text{Cov}(\hat{x}_n, x_n)
\]
\[
= u_0^T F^H H_d^H C_{ZZ,d}^{-1} H_d F u_0
\]
\[
= C
\]
(29)

It can be shown that the expectation of the residual error is $E[\epsilon_n] = 0$. The variance of the residual error writes as follows:
\[
\sigma_n^2 = \text{Cov}(\epsilon_n, \epsilon_n) = |C|^2
\]
\[
= u_0^T F^H H_d^H C_{ZZ,d}^{-1} H_d F \left( V + (1 - v_n)u_n u_n^T \right)
\]
\[
+ F^H H_d^H + \sigma_v^2 I_N C_{ZZ,d}^{-1} H_d F u_n - |C|^2
\]

where $V = \text{dia}(v_0, \ldots, v_{N-1})$ and $v_i$ is the covariance of the $i$th symbol. The term $(1 - v_n)u_n u_n^T$ is added due to the subtraction of the contribution of symbol $x_n$ which translates to $v_n = 1$. In the computation of the $\text{Cov}(Z, Z)$, the covariance of non linear ISI terms was approximated by $\sigma_v^2$. This approximation was referenced as implementation-b in [10] and was previously proposed in [16]. The equalizer output $\hat{x}_n \sim N(Cx_n, \epsilon_n)$ and the output extrinsic LLR $L_c$
h3 = 0.0503 + 0.0433i

<table>
<thead>
<tr>
<th>1st order kernels</th>
<th>3rd order kernels</th>
</tr>
</thead>
<tbody>
<tr>
<td>h0 = 0.8529 + 0.4502i</td>
<td>h000 = 0.1091 − 0.0613i</td>
</tr>
<tr>
<td>h1 = 0.0881 − 0.0014i</td>
<td>h010 = 0.0503 − 0.0503i</td>
</tr>
<tr>
<td>h2 = −0.0336 − 0.0106i</td>
<td>h001 = 0.9979 − 0.9979i</td>
</tr>
<tr>
<td>h3 = 0.0503 + 0.0433i</td>
<td>h110 = −0.0280 − 0.0475i</td>
</tr>
</tbody>
</table>

Fig. 3. BER comparison for different turbo equalizers

can write as:

\[ L_{ce}(c_{n,i} | \hat{x}_n) = \ln \frac{P(c_{n,i} = 0 | \hat{x}_n)}{P(c_{n,i} = 1 | \hat{x}_n)} - L_{a}(c_{n,i}) \]

\[ = \ln \frac{\sum_{s_j:s_j\neq 0} \left( \exp \left( \frac{-|x_n - Cs_j|^2}{\sigma_n^2} \right) \prod_{k \neq i} P(c_{n,k} = s_j,k) \right)}{\sum_{s_j:s_j\neq 1} \left( \exp \left( \frac{-|x_n - Cs_j|^2}{\sigma_n^2} \right) \prod_{k \neq i} P(c_{n,k} = s_j,k) \right)} \]

V. SIMULATION RESULTS

Independent and identically distributed bits are encoded using a 1/2 rate (7,5) non-recursive convolutional encoder. The codewords are then interleaved and mapped into 8PSK symbols using a gray mapping. A cyclic prefix of length 3 symbols is appended to each block of N = 512 symbols. The overall baud-rate equivalent channel is represented by the Volterra Kernels in Table. I proposed in [16] which have been extracted from [6] but with a stronger non linear ISI. Fig. 3 plots the BER performance for four turbo-iterations for both the time and frequency domain MMSE equalizers: the exact MMSE linear equalizer, the time domain MMSE equalizer (MMSE-TDE) and the frequency domain MMSE equalizer (MMSE-FDE). The exact MMSE refers to the full complexity time domain equalizer, whereas the MMSE-TDE refers to the time domain MMSE implementation-b in [10]. All time domain equalizers were simulated without CP, and hence the MMSE-FDE is shifted by 0.025 dB to account for the loss of spectral efficiency. Performance of the frequency domain equalizer joins that of the time domain equalizer.

VI. CONCLUSION

This paper has proposed an iterative frequency domain MMSE equalizer for non linear Volterra channels. The proposed equalizer has the advantage of a channel-memory implementation, as well as a simplified multiplicative equalization for the linear interference. The MMSE-FDE is equivalent to its time domain realisation at the exception of an additional cyclic prefix which is negligible for large block lengths and small channel memory.

REFERENCES