A new iterative phase tracking scheme

C. Vanstraceele\textsuperscript{1}; J.M. Brossier\textsuperscript{2}; J.P. Barbot\textsuperscript{1}; B. Geller\textsuperscript{1,3}

\textsuperscript{1}SATIE ENS Cachan, 61 avenue du Président Wilson, 94235 Cachan Cedex, France. vanstraceele@satie.ens-cachan.fr
\textsuperscript{2}LIS/ENSIEG. BP 46. 38402 Saint-Martin d’Hères. France. jean-marc.brossier@lis.inpg.fr
\textsuperscript{3}LETIC-ESRR Université Paris12, 110 rue Paul Armandot, 94400 Vitry sur Seine, France. geller@satie.ens-cachan.fr

Abstract

This paper presents a new synchronizing scheme designed for block turbo coded systems\textsuperscript{1}. The phase estimation relies on the feedback of the extrinsic information. A near optimum version of the algorithm based on the block turbo decoder devised by Pyndiah allows easy implementation. We display various results showing that the proposed algorithm is able to track frequency shifts and that it works very well in the case of general Quadrature Amplitude Modulation (QAM) constellations. We also show that this turbo synchronizer works at low Signal to Noise Ratios (SNR) taking completely advantage of the turbo coding possibilities.

1 Introduction

Turbo decoding [1], based on the iterative Maximum a Posteriori (MAP) decoding of concatenated convolutional codes, allows near Shannon capacity performance. For higher rates, iterative decoding of block codes [2] also achieves such an outstanding performance and Pyndiah [3] proposed a near optimum low complexity decoding of product codes. Some further improvements [4, 5] to lower the complexity and the delay of the decoding of the Block Turbo Codes (BTC) are even possible.

Nevertheless, all these methods assume that a perfect synchronization is achieved, which is unrealistic at low SNR where these sophisticated decoding algorithms are efficient. For instance, figure 1 illustrates the effect of a phase mismatch on the Bit Error Rate (BER) of a BCH(32,26,4)\textsuperscript{2} iteratively decoded. It should be clear that to achieve the theoretical performance, the carrier phase offset must be estimated and compensated for.

Since a carrier phase offset reduces the mean extrinsic power at a turbo decoder output, [6] [7] proposed phase recovery algorithms tracking the mean extrinsic power extremum of convolutional coded systems; [8] studies asymptotic bounds of trellis coded schemes. The phase tracking algorithm that we investigate in this paper works jointly with a Block Turbo Code and allows a significant reduction in decoding latency. Furthermore, we also consider the synchronization problem in the case of various QAM constellations.

\textsuperscript{1}This work was supported by the MEDEA A106 program

The remainder of this paper is thus as follows. In section 2, we briefly recall the BTC decoding principle and present the system model. Section 3 details the turbo synchronization algorithm and displays various results. Finally, Section 4 opens a discussion about this algorithm.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{BER_vs_phase_offset}
\caption{BER versus carrier phase offset at the output of a BCH(32,26,4)\textsuperscript{2} decoder (SNR=9dB, QAM16)}
\end{figure}

2 System model

2.1 Block Turbo Codes

At the transmitter, a product code is obtained by encoding in two dimensions an array of $k_1 \times k_2$ information binary symbols; each column of the array is encoded using a $C_1(n_1, k_1, d_1)$ linear block code and each line is encoded with a $C_2(n_2, k_2, d_2)$ code. The resulting product code $[P] = C_1 \ast C_2$ is a $P(n_1 n_2, k_1 k_2, d_1 d_2)$ code. We consider the case of various BCH constituent codes, with $C_1 = C_2$ for simplicity considerations; one can readily see that a minimum distance of 16 can be obtained with simple extended Hamming codes.

The received $n_1 \ast n_2$ array $[R]$ is decoded in each dimension with a Chase algorithm [9]; this means that for each row and each column of the array, test patterns are formed by perturbing the $p$ least reliable received positions. After decoding of the
Then the reliability information for each bit is expressed in terms of a log-likelihood ratio (LLR):

\[ A(d_j) = \log \frac{\Pr(e_j = +1/R)}{\Pr(e_j = -1/R)} \]  

(1)

where \( D = (d_1 \cdots d_j \cdots d_n) \) is the decided codeword after Chase decoding, \( R = (r_1 \cdots r_j \cdots r_n) \) is the received noisy sequence and \( E = (e_1 \cdots e_j \cdots e_n) \) is the transmitted codeword. Pyndiah [3] obtains the following approximation of the normalized LLR \( r'_j \): \( r'_j = r_j + \alpha w_j \), where \( \alpha \) is a weight factor and the extrinsic information \( w_j \) depends on \( C^{+1(j)} = \left(c_1^{+1(j)} \cdots c_n^{+1(j)}\right) \) and \( C^{-1(j)} = \left(c_1^{-1(j)} \cdots c_n^{-1(j)}\right) \) the codewords at minimum Euclidean distance from \( R \) with respectively a value +1 and −1 at the bit position \( j \). [3] then shows that:

\[ w_j = \sum_{k=1, k \neq j}^{n} r_k c_k^{+1(j)} p_k \]  

(2)

with \( p_k = 0 \) if \( c_k^{+1(j)} = c_k^{-1(j)} \) and \( p_k = 1 \) if \( c_k^{+1(j)} \neq c_k^{-1(j)} \). When all the competing codewords of the Chase decoding have the same value at position \( j \), the normalized LLR simply reduces to: \( r'_j = \beta d_j \), where the reliability factor \( \beta \) is fixed to 1.

The previous updating of the normalized LLR is performed on all the bits of the array code:

\[ [R'] = [R] + \alpha [W] \]  

(3)

A full iteration of the BTC decoder is made out of the serial updating of the normalized LLR working on the row words followed by the updating of the normalized LLR working on the column words. Four iterations are usually sufficient to insure the algorithm convergence.

### 2.2 Baseband turbo synchronizer scheme

In figure 2, the previous turbo decoder is included on the left side of the general baseband diagram of the turbo synchronizer. On the right side of figure 2, the noisy received demodulated QAM samples \( Y \) are multiplied by the term \( \exp(-j\hat{\theta}) \), where \( \hat{\theta} \) is the estimated residual carrier phase offset, in order to correct a possible phase mismatch; as detailed in section 3, this phase offset is driven by the feedback loop coming out from the turbo decoder output. The resulting stream of data \( Y' \) is then soft-demapped so that the turbo decoder proceeds soft value representing binary digits.

### 3 Phase estimation

A carrier phase offset at the receiver side involves a power reduction at the input of the turbo decoder which in turn reduces test patterns, the most likely among the generated codewords is chosen to be the decided codeword.

The phase tracking thus consist in searching the maximum of phases in figure 2; then the phase corresponding to the largest extrinsic value at the decoder output. This is illustrated for various QAM constellations by figure 3 where \( M \) is a measure of the extrinsic values of one BCH(16,11,4)2 codeword; more precisely, if \( M^{(i)} \) designates the lowest extrinsic values mean at the end of the decoding of the \( 2^{nd} \) codeword, one has the following expression:

\[ M^{(i)} = \frac{1}{m} \sum_{k=1}^{m} w_k^{(l)} \]  

(4)

where \( m \) is an integer smaller than the code length and \( w_k^{(l)} \) is one of the \( m \) smallest extrinsic value found within the \( l^{th} \) codeword ; if \( m \) is chosen too small, the noise will generate large fluctuations over consecutive codewords; on the opposite side, if \( m \) is chosen too large, the largest extrinsic values will perturb a possible phase drift tracking; this is due to the fact that more confident extrinsic values are less sensitive to possible perturbations. For instance we display results of an extended BCH(16,11,4)2 code, with \( m = 10 \).

As it could be expected, the smaller is the QAM constellation size, the larger is the "main lobe" of \( M \) displayed in figure 3. The phase tracking thus consist in searching the maximum of \( M \).

If it is assumed that at the starting point, the receiver has no knowledge of the initial phase, one can compute simultaneously several \( M^{init}(i) \) corresponding to several equally spaced \( \hat{\theta} \) trial phases in figure 2; then the phase corresponding to the largest value \( M^{init} = \max_i[M^{init}(i)] \) is chosen as the initial phase for the first codeword; one can see in figure 3 that for instance, 8,
16 and 32 initial phase trials are sufficient to find at least one trial initial phase in the main lobe of $M_{\text{init}}$ in the respective cases of the 8-QAM, 64-QAM and 1024-QAM constellations. If insuring a very low acquisition failure probability is the key issue, one can raise the calculus burden; for instance, during the acquisition period, the Chase decoder can tilt $p = 6$ test bits, or $M$ can be averaged over $N = 10$ consecutive words. As seen shortly, this smooths completely the variations of $M$ with the phase offset.

Once the initial phase acquisition successfully completed, the maximum value of $M$ can be reached for instance with a stochastic gradient. In this case, the algorithm to track the carrier phase is:

$$\hat{\varphi}_l = \hat{\varphi}_{l-1} + \mu (M^{(l)} - M^{(l-1)}) \left[ \text{sign}(\hat{\varphi}_{l-2} - \hat{\varphi}_{l-1}) \right]$$

(5)

where classically the stepsize $\mu$ is chosen to fulfill a tradeoff between the asymptotic Mean Square Error (MSE) and the convergence rate.

In practical situations there is often a frequency shift between the transmitter’s clock and the receiver’s clock; this classical synchronization problem [10] may be due for instance to Doppler shifts and the baseband discrete time residual phase can be modeled by a Brownian motion with a linear drift [11]. Figure 4 is a typical illustration of the variations of $M$ not only as a function of a constant phase offset, but also as a function of a constant linear drift. This surface is plotted in the case of a 64-QAM constellation with an extended BCH$(16, 11, 4)$ product code; the considered SNR=12 dB would give a 0.05 symbol error rate in the case of an uncoded 64-QAM constellation where perfect synchronization is assumed. One can recover the curve corresponding to a 64-QAM constellation in figure 3 by setting the linear drift to zero in figure 4. The main central lobe shows that the previous definition of $M$ allows to detect linear drift with a $10^{-2}$ rad/64 QAM symbol magnitude. As displayed in figure 5, when $M$ is averaged over $N = 10$ consecutive words (2560 bits), the local variations that could be seen in figure 4 are smoothed away. Figure 6 illustrates in the time domain the phase tracking ability of the turbo synchronizer in the case of a $2\pi \cdot 10^{-4}$ rad/symb linear drift (64 QAM) in the time domain (SNR=12 dB).

Finally, we compare the proposed algorithm with a fourth power phase estimator [12]. This algorithm is often used to recover QAM constellations. It is updated as follows; the observation is $y_k = a_k e^{i \varphi_k} + n_k$, where $a_k$ are i.i.d. random variables drawn from a QAM constellation, $n_k$ is a complex gaussian noise and $\varphi_k$ is the phase to be estimated. For a constant phase $\varphi_k = \hat{\varphi}$, the fourth power estimator is based on the equality $\varphi = \frac{1}{4} \arg \left( E(y_k^4) \right) + \frac{\pi}{4}$. The adaptive implementation:

$$\hat{\varphi}_k = \hat{\varphi}_{k-1} + \gamma \Im \left( y_k e^{-i4\hat{\varphi}_{k-1}} \right)$$

(6)

converges to the fourth-power solution (with a $\pi/2$ shift ambiguity) when estimating a constant phase; in (6), $\Im(z)$ stands for the imaginary part of a complex number $z$ and the stepsize $\gamma$ is
set to its optimum value \[11\]. Figure 7 illustrates that for example in the case of a 2048 QAM the turbo synchronizer achieves a smaller MSE which in turns leads to a smaller Bit Error Rate.

4 Conclusion

In this paper, we have presented a new phase synchronization algorithm. Taking advantage of a BTC extrinsic information feedback, this algorithm is able to cope with low SNR as well as frequency offsets. Block Turbo Codes have attractive features \[3]\[13\] for high data rate applications where large constellations sizes are useful; once again, this phase algorithm tackles very well the cases of large constellations. More over, this turbo phase estimator does not suffer from the traditional drawbacks of the fourth power estimator \[11\]: self created noise, cycle slip and a \(\pi/2\) ambiguity which may also influence the design of a Trellis Coded Modulation. This algorithm could thus be useful to Block Turbo Coded applications.

References


