ITERATIVE SYNCHRONISATION AND DC-OFFSET ESTIMATION USING SUPERIMPOSED TRAINING

S. M. A. Moosvi[1]*, D. C. McLernon*[1] *and E. Alameda-Hernandez*[2]

[1]School of Electronic and Electrical Eng.,The University of Leeds, Leeds LS2 9JT, UK.
[2]Electrical Engineering, Edificio Politecnico, Universidad de Granada, CP 18071, Granada, Spain.

ABSTRACT

In this paper, we propose a new iterative approach for superimposed training (ST) that improves synchronisation, DC-offset estimation and channel estimation. While synchronisation algorithms for ST have previously been proposed in [2],[4] and [5], due to interference from the data they performed sub-optimally, resulting in channel estimates with unknown delays. These delay ambiguities (also present in the equaliser) were estimated in previous papers in a non-practical manner. In this paper we avoid the need for estimation of this delay ambiguity by iteratively removing the effect of the data "noise". The result is a BER performance superior to all other ST algorithms that have not assumed a-priori synchronisation.

Index Terms— Synchronisation, Fading channels, Iterative methods, Estimation.

1. INTRODUCTION

In communications, the channel estimation problem is often solved by the inclusion of a training sequence. An alternative method is the superimposed training (ST) scheme, where a periodic training sequence is added to the data sequence [1]-[6], at the expense of small data power loss. In ST, it is important that the position within the received sequence, that corresponds to the start of a training sequence period, is known at the receiver. We will refer to this kind of synchronisation as training sequence synchronisation (TSS). TSS for ST was first studied in [2] in conjunction with DC-offset estimation. The TSS method presented in [2] was based on higher-order statistics (HOS) and polynomial rooting, and only required that the training sequence period (P) is no smaller that the number of channel taps (M), i.e. $P \geq M$. The use of HOS and polynomial rooting was avoided in the TSS method presented in [4], but required $P \geq 2M + 1$. A lower complexity synchronisation algorithm was proposed in [5] based on structural properties of the vector containing the cyclic means of channel output. But as we will show, it is outperformed by the algorithm presented here.

S. M. A. Moosvi is funded by the ORS Scheme and the University of Leeds.

Objectives and Contributions: Although synchronisation using ST was covered in [2],[4] and [5], it suffered from interference due to the transmitted information-bearing data. In this paper, we will remove the effect of the information data "noise" on the synchronisation process. This is done in an iterative manner, where the equalised symbols (obtained through traditional ST in a first step) are fed back to the proposed ST synchronisation algorithm so that the data "noise" can be successively reduced, and hence achieve better synchronisation, DC-offset estimation, channel estimation and symbol detection. This new method for TSS gives a much better performance than the existing methods for ST synchronisation in [2],[4] and [5] in terms of the MSE of the channel estimates and the BER. Finally note that while [6] also describes a form of iterative ST (but different from that proposed here), it assumes perfect synchronisation exists.

Notation: Superscript '†'and 'T' denote pseudo-inverse and transpose operator respectively. For matrix $\mathbf{A}$, define $\mathbf{A}^{[L]_c}$ and $\mathbf{A}_{[L]_c}$ to correspond respectively to its first and last L columns. Furthermore, $\mathbf{A}^{[L]_r}$ and $\mathbf{A}_{[L]_r}$ respectively denote its first and last L rows. Further, for any matrix $\mathbf{A}$, $\mathbf{A}_\tau$ is obtained by cyclically shifting the columns of $\mathbf{A}$ to the left by τ. $\mathbf{1}_{P\times Q}$ and $\mathbf{0}_{P\times Q}$ correspond respectively to $P \times Q$ matrices of ones and zeros. $\mathbf{I}_P$ is the $P \times P$ identity matrix. Finally, $\|.\|$ represents the Euclidean norm of a vector.

2. SYSTEM MODEL

The received data block in the ST method has the following form:

$$x(k) = \sum_{l=0}^{M-1} h(l)b(k-\tau_s-l) + \sum_{l=0}^{M-1} h(l)c(k-\tau_s-l) + n(k) + d \tag{1}$$

where $k = 0, 1, \dots, N-1$, $b(k)$ is the information bearing sequence, $h(k)$ is the channel impulse response, $n(k)$ is the noise, d represents an unknown DC-offset term due to using first-order statistics [2], [4], and the integer τ_s is the symbol synchronisation offset between the transmitter and the receiver. It is assumed that all terms in (1) are complex; that $b(k)$ and $n(k)$ are from independent, identically distributed (i.i.d) random zero-mean processes, with powers σ_b^2 and σ_c^2 respectively; and that the channel is of known order $M-1$, i.e. $h(0) \neq 0$ and $h(M-1) \neq 0$. Furthermore, $c(k)$ is the periodic superimposed training sequence (period $P \geq M$) with power $\sigma_c^2 = \frac{1}{P}\sum_{k=0}^{P-1}|c(k)|^2$.

The effect of τ_s in our set-up is as follows. For $\tau_s = 0$, (i.e., when there is perfect synchronisation), then $x(0)$ marks the start of the received block. But when $\tau_s \neq 0$, then $x(\tau_s)$ should mark the start of the received block– but of course, τ_s is unknown. As we will shortly show, for a proper operation of ST, τ_s must be determined modulo-P. So the problem now becomes to establish TSS for ST and then estimate the channel $\{h(m)\}_{m=0}^{M-1}$ from $\{x(k)\}_{k=0}^{N-1}$. Ideally of course $\{x(k)\}_{k=\tau_s}^{\tau_s+N-1}$ should be used, but this is impossible since τ_s is not known. Now since the transmitted and received blocks are N-samples long and $0 \leq \tau_s \leq N-1$, where $\tau_s = \tau_P P + \tau$, $0 \leq \tau_P \leq N_P - 1$, $0 \leq \tau \leq P-1$ and $N_P = N/P$, then we can estimate the cyclic mean of (1) as

$$\hat{y}(j) = \frac{1}{N_P}\sum_{i=0}^{N_P-1} x(iP+j) \tag{2}$$

with $j = 0,\ 1,\ \ldots,\ P-1$, where $y(j) \equiv E\{x(iP+j)\}$ is the true cyclic mean (period P). So from (1) and (2)

$$\begin{aligned}\hat{y}(j) = &\sum_{m=0}^{M-1} h(m)\tilde{b}(j-\tau-m) + \\ &+ \sum_{m=0}^{M-1} h(m)c(j-\tau-m) + \tilde{n}(j) + d\end{aligned} \tag{3}$$

where $\tilde{b}(k) = \frac{1}{N_P}\sum_{i=0}^{N_P-1} b(iP+k)$ with $k = 1-P,\ 2-P,\ \ldots,\ P-1$ and $\tilde{n}(j) = \frac{1}{N_P}\sum_{i=0}^{N_P-1} n(iP+j)$ with $j = 0,\ 1,\ \ldots,\ P-1$. So (3) can now be written as

$$\hat{\mathbf{y}} = (\mathbf{C}_\tau^{[M]_c} + \tilde{\mathbf{B}}_\tau^{[M]_c})\mathbf{h} + d\mathbf{1}_{P\times 1} + \tilde{\mathbf{n}} \tag{4}$$

where $\mathbf{h} = [h(0)\ \ldots\ h(M-1)]^T$ and $\hat{\mathbf{y}} = [\hat{y}(0)\ \ldots\ \hat{y}(P-1)]^T$, with a similar expression for $\tilde{\mathbf{n}}$. The matrices $\mathbf{C}_\tau$ and $\tilde{\mathbf{B}}_\tau$ are cyclically shifted versions of $\mathbf{C}$ and $\tilde{\mathbf{B}}$ respectively, with $(.)_\tau$ as defined in section 1. Note that $\mathbf{C}$ is a $P \times P$ circulant matrix with first column $[c(0)\ c(1)\ \ldots\ c(P-1)]^T$; and $\tilde{\mathbf{B}} = \tilde{\mathbf{B}}_1 + \tilde{\mathbf{B}}_2$, where $\tilde{\mathbf{B}}_1$ is $P \times P$ circulant with first column $[\tilde{b}(0)\ \tilde{b}(1)\ \ldots\ \tilde{b}(P-1)]^T$ and $\tilde{\mathbf{B}}_2$ is $P \times P$ upper triangular Toeplitz and $\frac{[b(-k)-b(N-k)]}{N_P}$ are the elements of the k-th ($k = 1,\ 2,\ \ldots,\ P-1$) upper diagonal. Due to the usual choice of relatively large N_P, we have $\tilde{\mathbf{B}}_2 \approx \mathbf{0}$. So (4) can be approximated as

$$\hat{\mathbf{y}} \approx (\mathbf{C}_\tau^{[M]_c} + \tilde{\mathbf{B}}_{1,\tau}^{[M]_c})\mathbf{h} + d\mathbf{1}_{P\times 1} + \tilde{\mathbf{n}} \tag{5}$$

$$\begin{aligned}\Rightarrow \hat{\mathbf{y}} \approx &(\mathbf{C}_\tau + \tilde{\mathbf{B}}_{1,\tau})\Big\{[\mathbf{h}^T\ \mathbf{0}_{1\times(P-M)}]^T \\ &+ m\mathbf{1}_{P\times 1}\Big\} + \tilde{\mathbf{n}}\end{aligned} \tag{6}$$

where $m = d/(P(\overline{c} + \overline{\tilde{b}}))$, $\overline{c} = \frac{1}{P}\sum_{k=0}^{P-1} c(k)$ and $\overline{\tilde{b}} = \frac{1}{P}\sum_{k=0}^{P-1}\tilde{b}(k)$. So an estimate for $\mathbf{h}$ and m is

$$[\hat{\mathbf{h}}^T\ \mathbf{0}_{1\times(P-M)}]^T + \hat{\mathbf{m}} = (\mathbf{C}_\tau + \tilde{\mathbf{B}}_{1,\tau})^{-1}\hat{\mathbf{y}} \tag{7}$$

where $\hat{\mathbf{m}} \approx \hat{m}\mathbf{1}_{P\times 1}$. We will now show how all the information regarding the synchronisation and channel estimation can be extracted from the LHS of (7).

3. PROPOSED SYNCHRONISATION, DC-OFFSET AND CHANNEL ESTIMATION ALGORITHM

Now when there is perfect TSS, i.e., τ is assumed known, then the information about the channel coefficients can be extracted from (7). But we first need $\hat{m}$, which can be found (in practice) by averaging over all $P-M$ values as follows:

$$\hat{m} = \frac{1}{P-M}\mathbf{1}_{1\times(P-M)}\Big((\mathbf{C}_\tau + \tilde{\mathbf{B}}_{1,\tau})^{-1}\hat{\mathbf{y}}\Big)_{[P-M]_r}. \tag{8}$$

Now the estimated channel coefficients are

$$\hat{\mathbf{h}} = \Big((\mathbf{C}_\tau + \tilde{\mathbf{B}}_{1,\tau})^{-1}\hat{\mathbf{y}} - \hat{m}\mathbf{1}_{P\times 1}\Big)^{[M]_r}. \tag{9}$$

But in practice, the actual constant offset τ is unknown. So now replacing the fixed (but unknown) τ in the RHS of (7) with a variable τ', $0 \leq \tau' \leq P-1$, is equivalent to cyclically permuting the LHS of (7) to give

$$\mathbf{P}_{\tau'}\Big\{[\hat{\mathbf{h}}^T\ \mathbf{0}_{1\times(P-M)}]^T + \hat{m}\mathbf{1}_{P\times 1}\Big\} = (\mathbf{C}_{\tau'} + \tilde{\mathbf{B}}_{1,\tau'})^{-1}\hat{\mathbf{y}} \tag{10}$$

where $\mathbf{P}_{\tau'}$ is a permutation matrix formed from shifting $\mathbf{P} = \mathbf{I}$ by τ' columns to the left. Note that the RHS of (10) is an unknown cyclic permutation of the RHS of (7). Thus, the information about the channel coefficients is still contained in (10), but now the information has to be extracted in a different way, i.e., we need to cyclically permute (10) back to its original form in (7). In order to achieve the synchronisation we propose to exploit the special structure of the vector in the LHS of (7), which has its last $P-M$ elements of equal magnitude. It should be noted that although [5] also uses the special structure of the vector containing the cyclic means of the channel output, it suffers due to interference from data. But in this proposed method, the data will also be used as effective training in an iterative fashion. So we simply search the RHS of (10) (for different values of τ') until the last $P-M$ elements $\Big($i.e. $\big((\mathbf{C}_{\tau'} + \tilde{\mathbf{B}}_{1,\tau'})^{-1}\hat{\mathbf{y}}\big)_{[P-M]_r}\Big)$ are all equal— as would theoretically be the case for proper synchronisation as in (7). Hence we propose to obtain synchronisation by minimising the cost function

$$\mathcal{J}(\tau') = ||\mathbf{V}\big((\mathbf{C}_{\tau'} + \tilde{\mathbf{B}}_{1,\tau'})^{-1}\hat{\mathbf{y}}\big)_{[P-M]_r}|| \tag{11}$$

where $\mathbf{V} := \mathbf{I}_{(P-M)} - \frac{1}{P-M}\mathbf{1}_{(P-M)\times(P-M)}$ acting on a vector produces the same vector with its mean removed from each element. Hence $\mathcal{J}(\tau')=0$ iff the last $P-M$ elements of $\left((\mathbf{C}_{\tau'}+\tilde{\mathbf{B}}_{1,\tau'})^{-1}\hat{\mathbf{y}}\right)_{[P-M]_r}$ are all equal. So the synchronisation offset is estimated (in practice) using

$$\hat{\tau}=\arg\min_{0\le\tau'\le P-1}||\mathbf{V}\left((\mathbf{C}_{\tau'}+\tilde{\mathbf{B}}_{1,\tau'})^{-1}\hat{\mathbf{y}}\right)_{[P-M]_r}|| \quad (12)$$

and the channel estimate obtained with

$$\hat{\mathbf{h}}=\Big(\left((\mathbf{C}_{\hat{\tau}}+\tilde{\mathbf{B}}_{1,\hat{\tau}})^{-1}\right)^{[M]_r}- -\frac{1}{P-M}\mathbf{1}_{M\times(P-M)}\left((\mathbf{C}_{\hat{\tau}}+\tilde{\mathbf{B}}_{1,\hat{\tau}})^{-1}\right)_{[P-M]_r}\Big)\hat{\mathbf{y}} \quad (13)$$

where we have used the last $P-M$ elements of (10) to estimate the scaled DC-offset and then used it along with first M elements of (10) to obtain the channel estimates. But note that (at the receiver) we have no a-priori knowledge of $\tilde{\mathbf{B}}_{1,\tau'}$ in (11)-(13). So to overcome this problem we propose the following iterative algorithm:

Iterative Synchronisation, DC-Offset Estimation and Channel Estimation Algorithm

a) Evaluate $\hat{y}(j)$ in (2).

b) Estimate $\hat{\tau}$ using (12), by setting $\tilde{\mathbf{B}}_{1,\tau'}=\mathbf{0}$.

c) Estimate the channel using (13), setting $\tilde{\mathbf{B}}_{1,\hat{\tau}}=\mathbf{0}$.

d) Design a MMSE equaliser (from (c)) and filter $x(k)$ (in (1)) to estimate $b(k)$ and hence obtain the estimate $\hat{\tilde{\mathbf{B}}}_{1,\hat{\tau}}$.

e) Repeat steps (b) to (d), but now using estimated $\tilde{\mathbf{B}}_{1,\hat{\tau}}$. This is the end of iteration one.

f) Perform subsequent iterations until there is little appreciable change in $\hat{\tau}$.

(Note that for simulations purposes later on we will refer to operations (a) to (d) as "step 1"—i.e. using $\tilde{\mathbf{B}}_{1,\tau'}=\mathbf{0}$ and $\tilde{\mathbf{B}}_{1,\hat{\tau}}=\mathbf{0}$.) Finally, as our algorithm achieves correct synchronisation by permuting the LHS of (10) until (theoretically) the last $(P-M)$ elements are all equal (i.e., $\hat{m}$), so we must ensure that $\hat{\mathbf{h}}$ itself cannot have $(P-M)$ equal elements, or the algorithm will fail. Setting $P-M>M$ ensures this, as $\hat{\mathbf{h}}$ has only M elements.

Remark 1: It should be noted that the method presented in [4] and [5] attempts synchronisation, but due to the interference from data and due to the presence of noise the TSS algorithm fails to correctly estimate the true synchronisation offset, which has the effect of delays/advances in the estimated channel impulse response and so this delay ambiguity also causes the equaliser output to be similarly delayed. Therefore [4] and [5] resort to estimating this (equalisation) delay by comparing the equalised sequence, shifted by different amounts, with the transmitted symbols $\{b(k)\}$ in order to compute the BER. So the delay providing the smallest BER is the actual true equalisation delay, and this can now be compensated for. In [5] this delay is then used to compute the correct MSE of the channel estimates. The problem is, of course, that in practice we cannot use the transmitted symbols to calculate this equalisation delay, via BER. In this paper, since we reduce the interference from data on synchronisation, we can avoid compensating for the equalisation delay, which cannot be obtained in practical applications. It should be noted that the only errors in synchronisation in our proposed method are due to the additive noise term $\tilde{\mathbf{n}}$ in (6) and the approximation $\tilde{\mathbf{B}}_2\approx\mathbf{0}$.

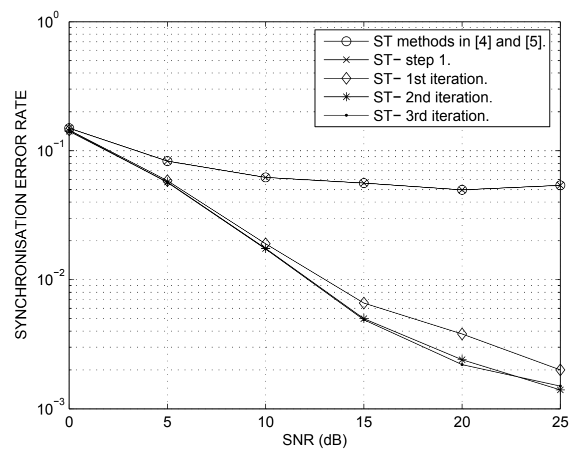

Fig. 1. Synchronisation error rate (see (14)) using the proposed iterative ST method. The results using the methods in [4] and[5] are also included for comparison.

4. SIMULATION RESULTS

We will now proceed with some simulations to test our synchronisation algorithm. The channel $h(k)$ was a three tap complex Rayleigh fading: both real and imaginary parts of the channel taps follow a normal distribution, rescaled to achieve unit mean energy channel $\left(\mathrm{E}\left[\sum_{m=0}^{M-1}|h(m)|^2\right]=1\right)$. The data was a BPSK sequence, to which a training sequence fulfilling $\mathbf{C}\mathbf{C}^{\mathrm{H}}=P\sigma_c^2\mathbf{I}$ as in [2] was added before transmission. The training to information power ratio $\left(\mathrm{TIR}=\frac{\sigma_c^2}{\sigma_b^2}\right)$ was set to 0.2 , $P=7$, $N=399$, DC-offset $d=\sqrt{0.1}$ and a linear MMSE equaliser of length $Q=11$ taps was used throughout as in [4]. The MMSE equaliser operates using its optimum delay– i.e., for a given delay α an MMSE equaliser ($\mathbf{w}_\alpha$) was computed, and $\alpha_{opt}=\arg\min_\alpha\left\{\sum_{k=-\infty}^{\infty}|\delta(k-\alpha)-(\hat{\mathbf{h}}*\mathbf{w}_\alpha)(k)|^2\right\}$, was used. In each Monte Carlo run, a random synchronisation offset between 0 and $N - 1$ was introduced between the transmitter and the receiver, so that we could be at any sample index within the first block. The synchronisation error rate (γ) is defined as

$$\gamma = \frac{\text{No. of errors in estimating the synchronisation offset}}{\text{Total no. of Monte Carlo runs}} \quad (14)$$

where an error is said to occur when $\hat{\tau} \neq \tau$. So Figure 1 shows the synchronisation error rate. It can be seen that the proposed method for synchronisation completely outperforms all existing conventional ST synchronisation schemes and we achieve maximum performance after only two iterations. As mentioned earlier, we can see that there is still some interference, due to the approximation of $\tilde{\mathbf{B}}_2 \approx \mathbf{0}$, which can be observed by the flattening of curve at high SNR. Now Figure 2 shows the MSE of the channel estimates using the proposed iterative method. We can also see that in two iterations we get the maximum performance. Again due to the approximation of $\tilde{\mathbf{B}}_2 \approx \mathbf{0}$, we see that the MSE of channel estimates does not decrease much for higher SNR. Finally, Figure 3 shows the BER using the proposed method and it completely outperforms all existing conventional ST schemes and maximum performance is again reached in two iterations.

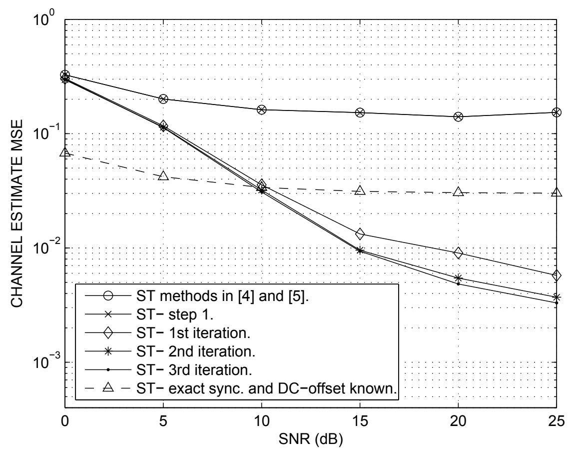

Fig. 2. MSE of channel estimates using the proposed iterative ST method. The results using the methods in [4] and [5], together with the results assuming known DC-offset and perfect TSS for ST, are also included for comparison.

5. CONCLUSION

In this paper, we have presented a new iterative synchronisation and DC-offset estimation algorithm for channel estimation using superimposed training (ST). No a-priori training sequence synchronisation (TSS) was assumed and a DC-offset could be present at the output. The proposed method of synchronisation is based on the particular structure of the

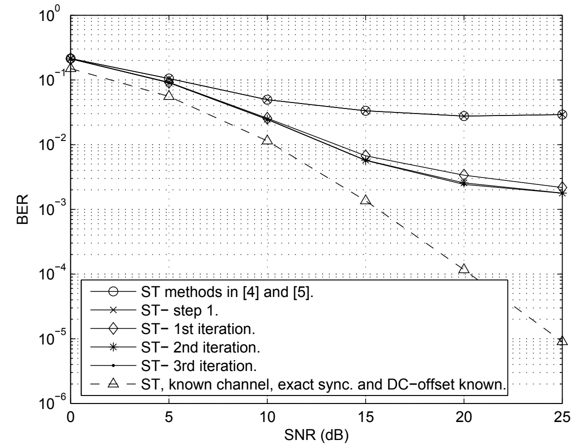

Fig. 3. BER using the proposed iterative ST method. The results using the methods in [4] and [5], together with the results assuming known DC-offset and perfect TSS for ST when the channel is completely known, are also included.

channel output's cyclic mean vector. Since the proposed method reduces the interference due to the data, it does not require any knowledge of equalisation delay (which cannot be obtained in practical applications). The simulations show that the proposed method completely outperforms the existing ST-based methods of [2],[4]-[5] and maximum performance is achieved in two iterations.

6. REFERENCES

[1] B. Farhang-Boroujeny, "Pilot-based channel identification: proposal for semi-blind identification of communication channels", *Elect. Lett.*, vol. 31, pp. 1044-46, 1995.

[2] A. G. Orozco-Lugo, M. M. Lara and D. C. McLernon, "Channel estimation using implicit training", *IEEE Trans. on Signal Processing*, vol. 52, pp. 240-254, 2004.

[3] J. K. Tugnait and W. Luo, "On channel estimation using superimposed training and first-order statistics", *IEEE Commun. Lett.*, vol. 7, pp. 413-415, 2003.

[4] J. K. Tugnait and X. Meng, "Synchronisation of superimposed training for channel estimation", *Int. Conf. Acoust, Speech, Signal Processing*, vol. 4, pp. 853-856, 2004.

[5] E. Alameda-Hernandez, D.C. McLernon, A.G. Orozco-Lugo, M. Lara and M. Ghogho, "Frame/training sequence synchronisation and DC-offset removal for (data-dependent) superimposed training based channel estimation.", *to appear in IEEE Trans. on Signal Processing*.

[6] X. Meng and J. K. Tugnait, "Semi-blind channel estimation and detection using superimposed training", *Int. Conf. Acoust, Speech, Signal Processing*, vol. 4, 2004.