Abstract—Over the last few years there has been growing interest in performing channel estimation via superimposed training (ST), where a training sequence is added to the information-bearing data, as opposed to being time-division multiplexed with it. Recent enhancements of ST are data-dependent ST (DDST), where an additional data-dependent training sequence is also added to the information-bearing signal, and semiblind approaches based on ST. In this paper, along with the channel estimation, we consider new algorithms for training sequence synchronization for both ST and DDST and block (or frame) synchronization (BS) for DDST (BS is not needed for ST). The synchronization algorithms are based on the structural properties of the vector containing the cyclic means of the channel output. In addition, we also consider removal of the unknown dc offset that can occur due to using first-order statistics with a non-ideal radio-frequency receiver. The subsequent bit error rate (BER) simulations (after equalization) show a performance not far removed from the ideal case of exact synchronization. While this is the first synchronization algorithm for DDST, our new approach for ST gives identical results to an existing ST synchronization method but with a reduced computational burden. In addition, we also present analysis of BER simulations for time-varying channels, different modulation schemes, and traditional time-division multiplexed training. Finally, the advantage of DDST over (conventional, non semi-blind) ST will reduce as the constellation size increases, and we also show that even without a BS algorithm, DDST is still superior to conventional ST. However, iterative semiblind schemes based upon ST outperform DDST but at the expense of greater complexity.

Index Terms—Block synchronization, dc-offset removal, implicit training, superimposed training channel estimation, training sequence synchronization.

I. INTRODUCTION

CHannel estimation is often a preliminary to channel equalization, which is crucial for communication systems operating in time-dispersive environments. Hence it is important to have a channel estimation technique that is both robust and accurate. Furthermore, reduced complexity techniques are desirable for practical implementation. The most popular approach to channel estimation is to use a training sequence that is allocated an empty time slot between information data packets at the channel input. This is known as time-division multiplexed (TDM) channel estimation. As these training sequence time slots now cannot contain any data, some useful bandwidth is inevitably lost.

An alternative implementation of the training sequence approach is superimposed training (ST) [1]–[11]—also known as hidden pilot [1] or implicit training [6]. In ST, the training sequence is added to the information sequence, and so no bandwidth is lost. However, the data power is now slightly decreased to accommodate the low-power training sequence. The ST sequence is generally chosen to be periodic with period $P$ [4], [6]–[8]. This periodicity induces cyclostationarity in the channel output’s first-order statistics. Knowledge of these cyclostationary statistics and the training sequence suffices, under certain conditions [6], to accurately estimate the channel. But two important points need consideration. First, because we are using first-order statistics, any possible dc-offset introduced by the nonideal radio-frequency (RF) receiver has to be included in the mathematical model [4]. Secondly, in order to unambiguously identify the channel, we need to be (symbol) synchronized at the receiver with the training sequence at the transmitter. We will refer to this kind of synchronization as training sequence synchronization (TSS).

References [4] and [8] solved the ST channel estimation problem, including a dc-offset, but with the prior (impractical) assumption of exact TSS. In [4], the approach adopted was to take the fast Fourier transform (FFT) of the cyclic mean, and discard the zero frequency component. In [8], a least squares (LS) approach was proposed. In addition, the error variance of the LS estimates of the channel and dc-offset was computed in [8], and the design of the training sequence that minimized this error variance was addressed.

The issue of unknown TSS was first addressed in [6] and later in [7] and [11]. The first two approaches also included the estimation (and removal) of the dc-offset, but while the former resorted to higher order statistics and polynomial rooting, the latter was based on first-order statistics and the FFT, and so
it had a reduced computational burden and, as shown by simulations in [7], a better performance. Although the computational burden reduction of [7] was significant, it still remained relatively high. The computational burden was further reduced without performance degradation in [11]. However, the method in [11] could not deal with a dc-offset. Also, it was restricted to work with training sequences with discrete Fourier transform (DFT) coefficients of equal magnitude, even when there was no proof that these kinds of training sequences were optimal for synchronization.

Once the dc-offset and TSS issues had been addressed, later research focused on improving the ST performance. In one approach, detailed in [9], the deterministic mean of each input data block was removed before transmission. This gave a significantly better channel estimation error, and the bit error rate (BER) (after equalization) was also reduced when compared to normal ST. But this approach was actually a special case of a more general development of ST that led to data-dependent ST (DDST). In DDST [10], we started with the realization that the \( N \) samples of information-bearing data in ST impair the channel estimation performance of the \( N \) samples of the \( P \)-periodic training sequence. By simply adding an additional (data-dependent) training sequence at the transmitter (but unknown to the receiver), we can significantly reduce the deleterious effect of this data. And if a cyclic prefix is also attached to each data block, then the data become “invisible” to the channel estimation algorithm. Viewed in the frequency domain, this simply means that the nonzero DFT bins of the training sequence now contain no contribution from the data. Note that [9] had implicitly forced this to happen only for the dc bin. Now, as the extra DDST training sequence is unique and different for each transmitted block, the DDST method requires block synchronization (BS), in addition to TSS. The issue of BS and DDST has not previously been addressed.

So the goals of this paper are as follows:

- to develop a TSS method for the (DD)ST method with the applicability of [6] and [7], but with a reduced computational burden;
- to develop a BS method for DDST that, combined with the TSS above, provides full synchronization for DDST;
- to design a training sequence for improving the performance of the proposed method—i.e., for improving synchronization;
- to provide an analytical expression for the variance of the channel estimate (for both ST and DDST) that includes the effect of the unknown dc-offset;
- to both analytically and experimentally investigate what effect lack of synchronization will have for both ST and DDST.

This paper is organized as follows. In Section II, a brief description of the setup for the ST and DDST methods is provided—in particular, the main assumptions that will be used in this paper will be stated. A new approach to the (DD)ST problem is introduced in Section III, from which the proposed new method for channel estimation—including TSS and dc-offset removal—is readily derived. The practical implementation of this new proposed method is then studied in Section IV, where BS is also introduced for the DDST method. In Section V, we analytically study the effect of lack of synchronization on both ST and DDST. The important topic of training sequence design to improve TSS using the proposed method is also considered. Simulations illustrate the performance of the proposed method in Section VI. Conclusions are drawn in Section VII.

Notation: Boldface uppercase and lowercase denote matrices and vectors, respectively. Vectors are assumed to be column vectors. The superscripts “T” and “H” denote, respectively, the transpose and conjugate transpose of a matrix or vector. For any matrix \( A \), we will define \( A^{[L]c} \) and \( A^{[L]k} \) to correspond, respectively, to its first and last \( L \) columns. Furthermore, \( A^{[L]c} \) and \( A^{[L]k} \), respectively, denote its first and last \( L \) rows. Considering a column vector of length \( L \) as an \( L \times 1 \) matrix, this notation can also be appropriately applied to any vector. \( \mathbf{1}_{L \times Q} \) and \( \mathbf{0}_{L \times Q} \) correspond, respectively, to \( L \times Q \) matrices of ones and zeros. \( \mathbf{I}_P \) is the \( P \times P \) identity matrix. Finally, \( || \cdot || \) represents the Euclidean norm of a vector.

II. GENERAL FRAMEWORK FOR BOTH ST AND DDST

Consider a complex, low-pass equivalent digital communications system as depicted in Fig. 1, configured for both ST [6] and DDST [10]. The channel’s input signal \( s(k) \) is a succession of data blocks, each block being \( N \)-samples long. Now

\[
s(k) = b(k) + c(k) + e(k)
\]

where \( b(k) \) is the information-bearing sequence; \( c(k) \) is the known periodic superimposed training sequence with period \( P \), mean \( \overline{c} = (1/P) \sum_{k=0}^{P-1} c(k) \), and power \( \sigma_c^2 = (1/P) \sum_{k=0}^{P-1} |c(k)|^2 \); and \( e(k) \) corresponds to a data-dependent training sequence term to be defined next. Assume then that the block of interest in (1) is indexed \( k = 0, 1, \ldots, N - 1 \), so that, within this block for ST \( e(k) = 0 \), while for DDST \( e(k) \) is periodic (period \( P \)) with the first period given by \( e(k) = -(1/N_P) \sum_{p=0}^{N_P-1} b(p + k), k = 0, \ldots, P - 1 \), where \( N_P = N/P \) [10].

Then, the channel output is given by

\[
x(k) = \sum_{l=0}^{M-1} h(l) s(k - \tau_l - l) + n(k) + m
\]
where \( h(k) \) is the channel impulse response, \( n(k) \) is the noise, and \( m \) represents the unknown dc-offset (first introduced in [4]) and due to the nonideal RF receiver. Also, in (2), we have an additional \( \tau_s \) symbol offset between the transmitter and the receiver, where the subscript “s” refers to lack of synchronization (not shown in Fig. 1). Note that all terms can be complex valued.

As we will shortly show, for a proper operation of ST, \( \tau_s \) must be determined modulo-\( P \)—i.e., TSS must be achieved—while for DDST \( \tau_s \) must be determined modulo-\( N \)—i.e., BS must also be achieved. So the problem now becomes to establish TSS (and BS) for ST (and DDST) and then estimate the channel \( \{h(l)\}_{l=0}^{M-1} \), initially from \( \{x(k)\}_{k=0}^{N-1} \), where \( h(k), e(k), n(k), \) and \( m \) are all unknown. Ideally, \( \{x(k)\}_{k=0}^{N-1} \) should be used, but this is impossible since \( \tau_s \) is not known. The following assumptions will now be adopted.

A1) \( h(k) \) is a zero-mean independent identically distributed (i.i.d) random process. The same applies to \( n(k) \). Furthermore, \( h(k) \) and \( n(k) \) are mutually independent.

A2) The channel is of order \( M-1 \)—i.e., \( h(0) \neq 0 \) and \( h(M-1) \neq 0 \).

A3) The exact channel order \( (M-1) \) is known in advance.

A4) Matrix \( C = \text{circ}(c(0), c(P-1), c(P-2), \ldots, c(1)) \), where \( \text{circ}(\cdot) \) produces a circulant matrix. We will further constrain \( C \) to be full rank.

A5) \( P \geq 2M + 1 \). This will be hereafter known as the strong constraint.

A6) The channel taps are independent random variables, with \( \mathbb{E}[h(l)h(l)^H] = \delta[l] \).

A7) \( 0 \leq \tau_s \leq N-1 \) is the synchronization offset, where \( \tau_s := \tau_P P + \tau \), with \( 0 \leq \tau_P \leq N_P - 1 \) and \( 0 \leq \tau \leq P-1 \), and \( N_P = N/P \). So the pair \( (\tau_P, \tau) \) uniquely determines \( \tau_s \).

A8) We will assume that the channel is invariant over the duration of the received samples \( \{x(k)\}_{k=0}^{N-1} \).

Assumption A3) will later be relaxed when an upper bound for the channel order is required instead and the lower bound for \( P \) in A5) could be reduced if enough information about the channel structure is available (see Section III-C). Note that A6) is only used in the analysis of the proposed method and is not required for the method to work. Only A1), A4), and A8) are necessary to develop the proposed method. Note first that A2) does not impose any real constraint since \( M \) can take any value and secondly that the real symbol offset in the system could be greater than \( N-1 \). As we are only interested in synchronizing with the beginning of the currently incoming block, it suffices to consider the values \( 0 \leq \tau_s \leq N-1 \) as in A7). This is in agreement with a previous comment stating that \( \tau_s \) is required modulo-\( N \). Note that the decomposition of \( \tau_s \) in A7) is desirable since it reflects the \( P \)-periodic nature of the training sequence.

The subsequent decomposition of \( \tau_s \) shown in A7) will now be used. Under A1) and A7), the previous definition of \( e(k) \) and the periodicity of \( c(k) \), the output \( x(k) \) in (2), has the following period-\( P \) cyclic mean

\[ y(j) := \mathbb{E}[x(jP + j)] = \sum_{l=0}^{M-1} h(l)c(j - \tau - l) + m \tag{3} \]

where \( j = 0, 1, \ldots, P-1 \). Note that the expression for \( y(j) \) given in the right-hand side of (3) does not depend on whether \( x(k) \) in (3) follows the ST or DDST schemes. The differences between ST and DDST will influence the actual implementation of the method to be discussed later. To simplify future analysis, we choose to write (3) in matrix form as

\[ y = C_{\tau}^* h + m1_{P \times 1} \tag{4} \]

where \( h = [h(0), h(1), \ldots, h(M-1)]^T \) and \( y = [y(0), y(1), \ldots, y(P-1)]^T \). Matrix \( C_{\tau} = \text{circ}(c(-\tau), c(-\tau - 1), \ldots, c(-\tau - P + 1)) \), where \( c(k) = c(k + P) \), is obtained by cyclically permuting the columns or rows of \( C \), and so \( C_{\tau} \) is also full rank by A4). Note that all indexes of \( c(k) \) can be recalculated modulo-\( P \), since \( c(k) \) is periodic with period \( P \).

### III. PROPOSED METHOD FOR CHANNEL ESTIMATION FOR BOTH ST AND DDST

An equation similar to (4) was used in [4], [6], and [8], either in its present form or in its frequency-domain equivalent, to compute the channel \( h \). Nevertheless, from this equation, there is no direct way to approximate the values of \( \tau \) and \( m \), which are necessary to extract the channel \( h \), and the procedures currently proposed to compute them [6], [7] are rather cumbersome. In order to significantly simplify the computation of both \( \tau \) and \( m \), in this paper we propose to rewrite (4) in a different form

\[ y = C_{\tau} \left[ [h^T 0_{1 \times (P-M)}]^T + \hat{m}1_{P \times 1} \right], \quad \hat{m} = \frac{1}{P} \hat{m}. \tag{5} \]

The reasons for doing this will become clear in the following sections, where we will prove that (5) and knowledge of the training sequence alone are sufficient to obtain the channel coefficients, even in the more general case where both \( \tau \) and \( m \) are unknown. This is more easily achieved when (5) is rewritten as

\[ C_{\tau} y = [h^T 0_{1 \times (P-M)}]^T + \hat{m}1_{P \times 1}. \tag{6} \]

To begin with, in the next section we will first show how \( h \) can be obtained assuming perfect TSS.

### A. Channel Estimation Assuming Perfect Training Sequence Synchronization

Having perfect TSS implies that \( \tau \) is known, and so to extract \( h \) from (6), we need first \( \hat{m} \). Note that \( \hat{m} \) can be obtained from any of the last \( P-M \) elements in the right-hand side of (6), but in anticipation of later replacing \( y \) in (6) by its finite sample estimate (\( \hat{y} \)), we propose

\[ \hat{m} = \frac{1}{P-M} 1_{1 \times (P-M)} (C_{\tau}^* y)_{[P-M]P} \]. \tag{7} \]

Then, the channel coefficients can be computed from the first \( M \) elements of \( C_{\tau}^* y - \hat{m}1_{P \times 1} \) in (6) as follows:

\[ h = (C_{\tau}^* y - \hat{m}1_{P \times 1})^{[M]P}. \tag{8} \]
where \( \hat{m} \) comes from (7). For computational simplicity and for convenience in later developments, using (7), we can rewrite (8) as

\[
\hat{h} = \Gamma \hat{y}
\]  

(9)

where

\[
\Gamma = (C^{-1}\tau)^{[M]}r - \frac{1}{P-M} 1_{M \times (P-M)}^T (C^{-1}\tau)^{[P-M]}r.
\]  

(10)

As a special case, for a nonexistent dc-offset, we have

\[
\Gamma = (C^{-1}\tau)^{[M]}r.
\]  

(11)

Note that \( \Gamma \) can be computed once and for all knowing \( P, M, \tau, \) and \( \epsilon(k) \).

It is important to note that (8) and the least squares approach proposed in (9) give exactly the same value for the channel vector if the training sequence is such that all its DFT coefficients have the same modulus. This kind of training sequences play an important role in (DD)ST, as vouched for by the results in the following sections.

B. Channel Estimation Where Training Sequence Synchronization Has to be Implemented at the Receiver

With perfect TSS, the channel coefficients were computed from (8), which was in turn derived from (6). Obviously, \( \tau \) was needed in order to get (6) from (5). Now we will assume that \( \tau \) is not known. So, let us investigate what happens if we try to follow the same steps as above. Take any integer \( \tau' \) with \( 0 \leq \tau' \leq P-1 \) and multiply (5) by \( C^{-1}\tau \) to get

\[
C^{-1}\tau y = P_{\tau'}\left\{ \left[h^T 0_{1 \times (P-M)} \right]^T \hat{m} 1_{P \times 1} \right\}
\]  

(12)

where \( P_{\tau'} \) is an unknown \( P \times P \) cyclic permutation matrix such that \( C_{\tau'} = P_{\tau'} C_{\tau} \). and we have used the fact that any cyclic permutation matrix is a circulant matrix and circulant matrices commute [12]. Note that the right-hand side of (12) is an unknown cyclic permutation of the right-hand side of (6). Thus, the information about the channel coefficients is still contained in (12), but now the information has to be extracted in a different way—i.e., we need to cyclically permute (12) back to its original form in (6). We then have to find out the position that corresponds in (6) to at least one element of the vector in (12).

What is special about the structure of the vector in the right-hand side of (6) is that its last \( P-M \) elements are all equal. So we have to find in the right-hand side of (12) a sequence of \( P-M \) equal elements, but because of matrix \( P_{\tau'} \) in (12), this is more easily visualized via a “circular” geometry. Fig. 2 illustrates this.

The next proposition summarizes the proposed method to establish TSS at the receiver. First, we define \( \hat{m}_{\tau'} \) to be as in (7) except that \( \hat{C}_{\tau} \) is now replaced by \( \hat{C}_{\tau'} \). Remember that \( \tau \) is the unknown (fixed) true TSS offset and \( \tau' \) is a variable.

**Proposition 1:** Let \( \tau' \) be an integer satisfying \( 0 \leq \tau' \leq P-1 \). Then, under A2)–A5), \( J(\tau') := \| (C^{-1}\tau y)_{[P-M]} r - \hat{m}_{\tau'} 1_{(P-M) \times 1} \| = 0 \) if \( \tau' = \tau \).

**Proof:** The necessary condition \( (J(\tau') = 0 \iff \tau' = \tau) \) is proved by realizing that \( \hat{m}_{\tau'} - \hat{m} \) and then using (6). The proof of sufficiency \( (J(\tau') = 0 \Rightarrow \tau' = \tau) \) begins by noting, from the definition of \( \hat{m}_{\tau'} \), that \( J(\tau') = \| V(C^{-1}\tau y)_{[P-M]} \| \), where \( V := I_{P-M} - (1/(P-M)) 1_{(P-M) \times (P-M)} \) acting on a vector produces the same vector with its mean removed from each element. So it follows from (12) that \( \| V(P_{\tau'} h^T 0_{1 \times (P-M)}^T) 1_{P-M} \| = 0 \) if the last \( P-M \) elements of \( P_{\tau'} h^T 0_{1 \times (P-M)}^T \) are all equal. Now, as letting \( \tau' = \tau \) is equivalent to setting \( P_{\tau'} = I_P \), then the last \( P-M \) elements of \( P_{\tau'} h^T 0_{1 \times (P-M)}^T \) should come from the vector \( 0_{1 \times (P-M)} \) i.e., \( 0_{1 \times (P-M)} \) should represent the only \( P-M \) equal elements in \( P_{\tau'} h^T 0_{1 \times (P-M)}^T \). And to ensure that \( P-M \) equal elements cannot occur in \( h \), we let \( P-M > M \) i.e., the previously stated assumption A5).

**Remark:** It is computationally more efficient to use \( J(\tau') = \| V(C^{-1}\tau y)_{[P-M]} \| \) instead of its original form at the beginning of Proposition 1, and so this new form will be used where convenient.

It turns out that a special case of Proposition 1—i.e., training sequences fulfilling \( C^H \mathbf{C} = J_A \) and no dc-offset—can be found in [11]. Note as well that the TSS problem visualization of [11] as a subspace projection problem is completely lost here. Thus, although Proposition 1 is a generalization of [11] (in the sense just described), both rely on completely different formalisms, are obtained differently, and should not be interpreted in the same way.

C. Comments on Assumptions A3)–A5) for Channel Estimation and Equalization

So far, our proposed channel estimation method can be summarized by the following three steps.

**Step 1)** Compute \( y \) according to (3).

**Step 2)** The true offset \( \tau \) in (3) is obtained via Proposition 1.

**Step 3)** The channel coefficients are then computed from (8).
1) **Comment on A4**: Now the reasoning behind A4 is clear. The synchronization method makes use of the inverse of $C$. It is important to remark that, from an LS approach to the (DD)ST estimation problem, full rank matrices $C$ do not provide the minimum variance channel and dc-offset estimates [8] if synchronization is assumed. So it would be desirable to design a new TSS method without this restriction.

2) **Comment on A5**: In light of the proof of Proposition 1, the strong constraint in A5 can be relaxed. Assume that the number of contiguous equal taps in $h$ is $G$; then the strong constraint can be replaced by a soft constraint like $P - M > G$. For $G = M$, the soft constraint becomes the strong constraint. It should be noticed that a channel with all its taps equal is very unlikely to happen, so the strong constraint could be relaxed anyway in practice. In existing TSS methods, the condition relating $P$ and $M$ for a proper TSS varies. In [6], $P = M$ was sufficient, while in [7], $P \geq 2M + 1$ was required. Although this last condition is the same as the one in this paper, this is the first time that an interpretation of this condition in terms of the channel structure is provided. Hence, according to this interpretation, the constraint can be relaxed if $G$ (above) is known.

3) **Comment on A3**: A3 may be the most restrictive assumption, given that in practical cases the exact number of taps ($M$) is often not known. Assume that instead what is available is just an upper bound on the number of taps. To develop this idea, assume (temporarily) that this upper bound on the number of taps is $M + 1$ and the training sequence period (as before) is equal to $P$. Consider again the proof of Proposition 1 and examine the term $P_{r}[h^T 0_{(P-M)}^T]_T$. This should now be restructured as the vector $P_{r}[h^T 0_{(P-M)}^T]_T$. Since we are looking for a $(P - M - 1)$-long sequence of equal values in this vector, we get two possibilities for the channel estimate—i.e., $[h^T 0]_T$ or $[0 h^T]_T$ (since $h(0) \neq 0$ and $h(M - 1) \neq 0$ by A2). If we choose incorrectly (i.e., the latter), then we simply introduce a single sample (identification) delay into the ideal equalizer. So returning to the initial assumption of upper bound $M + 1$, we can easily generalize for $M + 2$, $M + 3$, etc., where the only change will be an increased identification delay after equalization. Thus, if we are interested in equalization up to an unknown delay, A3 can be rewritten as follows.

A3) An upper bound for the channel order is known.

Note that with the practical implementation of the next section (i.e., with noise present), an identification delay can also be introduced simply by virtue of having very small coefficients either at the beginning or at the end of the channel vector.

IV. **PRACTICAL IMPLEMENTATION OF THE PROPOSED METHOD**

The first step in the practical implementation of the proposed method is to estimate (the theoretical) $\mathbf{y}$ as defined in (3). The elements of the cyclic mean vector $\mathbf{y}$ have to be estimated from $\{x(k)\}_{k=1}^{N-1}$. The proposed estimate is

$$\hat{y}(j) = \frac{1}{N_P} \sum_{i=0}^{N_P-1} x(iP + j), \quad j = 0, 1, \ldots, P - 1. \quad (13)$$

So for (2) with $\tau_s = 0$, then from (1)—(3)

$$\hat{y}(j) = y(j) + \sum_{l=0}^{M-1} h(l) \left[ \hat{b}(j-l) + e(j-l) \right] + \frac{1}{N_P} \sum_{i=0}^{N_P-1} n(iP + j) \quad (14)$$

where $\hat{b}(j) = (1/N_P) \sum_{i=0}^{N_P-1} \hat{h}(iP + j)$ with $j = 0, 1, \ldots, P - 1$. Clearly, from (14), the estimate consists of the true $y$ plus an estimation noise term. The estimation noise term for ST is simply $\sum_{l=0}^{M-1} h(l) \hat{b}(j-l) + (1/N_P) \sum_{i=0}^{N_P-1} n(iP + j)$. For DDST, given that we have explicitly set $e(j) = -\hat{b}(j)$ (see after (1), and also [10]), the estimation noise reduces to $(1/N_P) \sum_{i=0}^{N_P-1} n(iP + j)$ (i.e., there is no noise contribution from the data $b(k)$) and consequently the channel estimation is significantly improved [10]. This cancellation in (14) has been possible because for $\tau_s = 0$, $\{x(k)\}_{k=1}^{N-1}$ corresponds exactly with the output of a transmitted block—in this case $\{\hat{b}(k)\}_{k=1}^{N-1}$. In the case of a nonzero synchronization offset (i.e., $\tau_s \neq 0$), the $\hat{b}(j)$ in (14) is no longer derived from a single transmitted data block but from $N$ samples spread over two different (contiguous) blocks. In this case, $\hat{b}(j-l)$ and $e(j-l)$ will not cancel in (14). Remember that $\{e(k)\}_{k=1}^{N-1}$ is defined as the $P$ periodic means of the data sequence $b(k)$ within a given transmitted block. Thus, to benefit from the full potential of DDST, in addition to TSS, BS is now also needed. But note that BS is not needed for ST.

A. **Block Synchronization**

We know from Proposition 1 that the correct $\tau$ ensures $\left\| \left( C \mathbf{y} \right)_{[P-M], -1} - \mathbf{n} \mathbf{1}_{(P-M) \times 1} \right\| = 0$. Now, if $\mathbf{y}$ and $\mathbf{n}$ are replaced by estimates, then clearly the better the estimates, the closer this term will be to zero. The BS technique—explained in the context of the implementation of the full proposed method for channel estimation—is as follows. The cyclic mean vector $(\mathbf{y})$ is estimated using time averages (13), and then an estimate of $\tau$ is obtained, via Proposition 1. The BS now starts by estimating $\tau_P$ [see A7)]. With $\tau$ already estimated (via Proposition 1), define $\hat{y}(j; \tau_P) = (1/N_P) \sum_{i=0}^{N_P-1} x(iP + \tau_P + \hat{\tau} + j)$, $j = 0, 1, \ldots, P - 1$. Then, we propose (based on the new definition of $\mathbf{F}(\tau')$ at the end of Proposition 1)

$$\hat{\tau}_P = \arg \min_{0 \leq \tau'_P \leq N_P-1} \left\| \mathbf{V} \left( C \mathbf{y} \right)_{[P-M], \tau'_P} \right\| \quad (15)$$

where $\mathbf{V}$ was defined for Proposition 1 and $\mathbf{y}(\tau'_P) = [\hat{y}(0; \tau'_P), \hat{y}(1; \tau'_P), \ldots, \hat{y}(P-1; \tau'_P)]^T$. The channel coefficients are finally computed using $\hat{y}(\tau_P)$ as an estimate of $\mathbf{y}$ in (9).

B. **Computational Burden**

Consider the TSS step via Proposition 1 and note that each of the vectors $\{C \mathbf{y} \}_{P-1}^{P-1}$ does not need to be explicitly computed from $\hat{y}$. We only need to cyclically permute $C^{-1} \hat{y}$. Also, the product $\mathbf{V} \mathbf{a}$, where $\mathbf{a}$ is any $(P - M) \times 1$ vector, does not require $(P - M)^2$ products. As we noted previously in Section III-B, $\mathbf{V}$ just subtracts the mean of all the elements of $\mathbf{a}$ from each element of $\mathbf{a}$, and this needs just two products.
This is a way of reducing the number of complex multiplications. Bearing this in mind, the computational burden of the proposed method, in terms of complex products/divisions, is given by \( P^3 + (1 - M)P^2 + 2P^{1-3} \), i.e., \( O(P^3) \) for the ST method, and an additional \( NP(P - M + 1)(P + 1) \), i.e., \( O(N(P - M)) \) if the DDST design is used.

There currently exist two TSS techniques for ST [6, 7] assuming a nonzero dc-offset. The technique in [7] is more efficient than that in [6] since it uses first-order statistics and the FFT, as opposed to higher order statistics and polynomial rooting. The method in [7] requires \( MNP + 2P^3 + (M + 1)P^2 - (M + 2)P + 1 \) products and divisions, or equivalently \( O(MNP) \). This is significantly higher than the \( O(P^3) \) computational burden of the proposed method for channel estimation within the ST framework, but is also still greater than the computational burden of the proposed method for channel estimation within the DDST framework.

As a comparison, take typical values of \( M = 3 \), \( P = 7 \), and \( N = 400 \) as reported in [7]. So, using the actual computational burden expressions of the previous paragraphs, we obtain 262, 2542, and 9248 complex products and divisions for ST, DDST (including BS), and [7]. Even including BS, our proposed method still presents a computational advantage over the approach in [7]. The differences in performance will be shown later via simulations in Section VI.

V. COMPARING ST AND DDST: SYNCHRONIZATION AND CHANNEL ESTIMATION

The estimate \( \hat{\mathbf{y}} \) in (13) of the true \( \mathbf{y} \) in (3) plays a central role in the proposed (DD)ST algorithm, and so the accuracy of \( \hat{\mathbf{y}} \) will have a significant impact on the performance of the (DD)ST algorithm. From (13), similar to what was shown in (14) for the special case of \( \tau_s = 0 \), it is easy to see that \( \hat{\mathbf{y}} = \mathbf{y} + \epsilon \), where \( \epsilon \) is approximately a zero-mean random Gaussian vector that accounts for the estimation noise. The Gaussianity is conferred by the central limit theorem when applied to any element of \( \epsilon \) that can be shown to be the sum of \( \mathcal{N} \) i.i.d. random variables. Thus, in this section, we will first study the variance of each element of \( \hat{\mathbf{y}} \) and will show that for any element of \( \hat{\mathbf{y}} \), its variance for DDST is always smaller than that for ST irrespective of synchronization. This will lead us to make the following two important claims (when TSS and BS also have to be estimated).

C1) DDST provides better synchronization than ST.
C2) DDST provides better channel estimates than ST.

Note that C2) is partially a consequence of C1) but that synchronization and channel estimation are different problems: the first deals with the estimation of \( \tau_s \) and the second is concerned with the estimation of \( \mathbf{h} \). On the other hand, C1) is an interesting claim on its own. Imagine that \( \mathbf{h} \) is known; then C1) implies that DDST should be used if we are now interested in synchronization.

Note that C1) and C2) apply to the ST algorithm as developed in [4] and [6], for example, where channel estimation and equalization are decoupled. New iterative semiblind approaches based on ST, such as that presented in [13] and [14], do not follow C1) and C2) since the initial ST setup is transformed (after a few iterations) into a fully trained scenario where \( \hat{\mathbf{y}} \) does not play any role. But for comparison with DDST, the method of [13] and [14] is later examined via simulations in Section VI-C.

So, let us now examine the variance of \( \hat{\mathbf{y}} \) and define

\[
\sigma^2_{\hat{\mathbf{y}}} = \mathbf{E}[(\hat{g}(j) - \mathbf{E}[\hat{g}(j)])^2] = (\mathbf{R}_g)_{jj} \tag{16}
\]

which in turn is the \( j \)th element of the diagonal of the covariance matrix \( \mathbf{R}_g = \mathbf{E}[(\hat{\mathbf{y}} - \mathbf{y})(\hat{\mathbf{y}} - \mathbf{y})^H] \), since \( \mathbf{y} = \mathbf{E}[\hat{\mathbf{y}}] \). Under A6), it is not difficult to show that \( \mathbf{R}_g \) is indeed diagonal, so the set \( \{\sigma^2_{\hat{\mathbf{y}}}(j)\}_{j=0}^{P-1} \) completely determines \( \mathbf{R}_g \).

The data-dependent training sequence \( e(k) \) employed in the DDST algorithm makes the output sequence \( x(k) \) in (2) different for ST and DDST. Then, the estimate \( \hat{\mathbf{y}} \) in (13) is different for ST and DDST, thus affecting the expression for \( \sigma^2_{\hat{\mathbf{y}}} \). So \( \sigma^2_{\hat{\mathbf{y}}} \) has to be specifically computed for each algorithm—i.e., ST and DDST. Additionally, the effect of the synchronization offset will also be considered.

An expression for \( \sigma^2_{\hat{\mathbf{y}}} \) in the ST setup can be derived from its definition in (16). The result does not depend on \( \tau_s \), and (16) becomes

\[
\sigma^2_{\hat{\mathbf{y}}}^{ST} = \frac{1}{NP} \left( \sigma^2_h + \sigma^2_n \right) \tag{17}
\]

where \( \sigma^2_h = \mathbf{E}[(\sum_{i=0}^{M-1} h(i)\mathbf{P})^2] \) is the mean energy of the random channel. The fact that \( \sigma^2_{\hat{\mathbf{y}}}^{ST} \) is independent with respect to \( \tau_s \) does not mean that ST does not require synchronization. It only means that the estimation of \( \mathbf{y} \) in (3) is neither worsened nor improved by \( \tau_s \). Note that \( \tau \) is taken into account when estimating \( \mathbf{h} \).

Now, the value of \( \sigma^2_{\hat{\mathbf{y}}}^{DDST} \) for DDST is strongly dependent upon the synchronization offset. Although the analysis is not technically difficult, it has to take into account many different factors. This analysis is thus shown in the Appendix, but here as an example let \( \tau_s = 0 \) (or known \( \tau_s \)). So, from (14) and (16), it is not difficult to show that

\[
\sigma^2_{\hat{\mathbf{y}}}^{DDST} = \frac{1}{NP} \sigma^2_n \tag{18}
\]

where, as expected, the dependency on \( \sigma^2_h \) has disappeared, compared with (17). So the purpose of (18) is to illustrate that \( \sigma^2_{\hat{\mathbf{y}}}^{DDST} < \sigma^2_{\hat{\mathbf{y}}}^{ST} \) (and this holds for all \( \tau_s \)—see the Appendix). This supports C1) and C2).

Now, in the next sections (as a direct application of this section) we will look for the “optimum” (to be defined) training sequence for synchronization and will compute the channel estimate error power.

A. Training Sequence Design for Both ST and DDST

The training sequence should be designed so that the proposed algorithm has the best possible performance, in the sense now to be described. The proposed method uses the following sequence of estimates \( \{\hat{\mathbf{y}}, \hat{\tau}, \hat{\mathbf{y}}(\tau_{n}), \hat{\tau}_{n}, \hat{\mathbf{h}} \} \). Note that \( \hat{\mathbf{y}} \) is obtained from \( \{x(k)\}_{k=0}^{M-1} \) [see (14)], and so there is no way of improving this estimation. It is clear that the better the estimate \( \hat{\tau} \) is, the better will be \( \hat{\mathbf{y}}(\tau_{n}) \) and \( \hat{\tau}_{n} \), and consequently a
better estimate of \( \hat{y} \) should be obtained. So if TSS is not accomplished, then the proposed method will not provide any meaningful channel estimates. It is also clear that since the TSS is based on \( \mathbf{C}^{-1} \mathbf{y} \) and all its cyclic permutations, the better the estimate of \( \mathbf{C}^{-1} \mathbf{y} \) is, then the better TSS we should have.

We now propose to design the training sequence such that \( \mathbf{C}^{-1} \hat{\mathbf{y}} \) is “as close as possible” to the true value \( \mathbf{C}^{-1} \mathbf{y} \). Recall that \( \mathbf{y} \) is given by (3) or, equivalently, (4), and \( \hat{\mathbf{y}} \) is given by (14). So due to the Gaussian nature of \( \mathbf{y} \), and since \( \mathbb{E}[\hat{\mathbf{y}}] = \mathbf{y} \), we seek to minimize

\[
\sigma_{\mathbf{C}^{-1} \mathbf{y}}^2 = \mathbb{E} \left[ \| \mathbf{C}^{-1} \hat{\mathbf{y}} - \mathbb{E}[\mathbf{C}^{-1} \mathbf{y}] \|^2 \right].
\]  

Then after some manipulation we get

\[
\sigma_{\mathbf{C}^{-1} \mathbf{y}}^2 = \text{Tr} \{ \mathbf{C}^{-1} \mathbf{C} \mathbf{R}_{\mathbf{y}} \mathbf{C}^{-1} \mathbf{R}_{\mathbf{y}} \mathbf{C} \}
\]  

where the covariance matrix \( \mathbf{R}_{\mathbf{y}} = \mathbb{E} \{ \hat{\mathbf{y}} - \mathbf{y} \hat{\mathbf{y}} - \mathbf{y} \mathbf{y}^H \} \) is as previously defined for (16).

Now let the familiar DFT matrix \( \mathbf{F} \) have entries \( \mathbf{F}_{ij} = \exp(-j2\pi k(l/P)) \), and so define the following similarity transforms:

\[
\mathbf{A}_c = \mathbf{F}^{-1} \mathbf{C} \mathbf{F},
\]

\[
\hat{\mathbf{R}}_{\mathbf{y}} = \mathbf{F}^{-1} \mathbf{R}_{\mathbf{y}} \mathbf{F}.
\]

Note that because \( \mathbf{C} \) is circulant, then \( \mathbf{A}_c \) is diagonal with values equal to the DFT coefficients of the training sequence \( \{c(k)\}_{k=0}^{P-1} \).

So from (20) and (21), we can say that

\[
\sigma_{\mathbf{C}^{-1} \mathbf{y}}^2 = \text{Tr} \left\{ \mathbf{A}_c^{-1} \mathbf{A}_c^{-1} \hat{\mathbf{R}}_{\mathbf{y}} \right\}
\]

and the following lemma will help us minimize (22).

**Lemma 1:** Let \( \hat{\mathbf{R}}_{\mathbf{y}} \) be the previously defined covariance matrix of \( \hat{\mathbf{y}} \) and let \( \hat{\mathbf{R}}_{\mathbf{y}} \) in (21) be the result of a similarity transformation with the FFT matrix \( \mathbf{F} \). Then, under assumption A6), all the diagonal elements of \( \hat{\mathbf{R}}_{\mathbf{y}} \) equal \( \tau_y \), where \( \tau_y \) is defined as the mean of all the elements in \( \mathbf{R}_{\mathbf{y}} \).

**Proof:** Since \( \mathbf{F}^{-1} = (1/P)^{-1} \mathbf{F}^H \) and \( |(\mathbf{F})_{ij}|^2 = 1 \forall i, j \), it is easy to see from (21) that \( (\hat{\mathbf{R}}_{\mathbf{y}})_{ii} = (1/P) \sum_{k=1}^{P} (\mathbf{R}_{\mathbf{y}})_{ii} := \tau_y \) given that under A6), \( \mathbf{R}_{\mathbf{y}} \) is diagonal.

Now, since \( \mathbf{A}_c^{-1} \mathbf{A}_c^{-1} \) is a diagonal matrix, then from Lemma 1, we can write that

\[
\sigma_{\mathbf{C}^{-1} \mathbf{y}}^2 = \tau_y \text{Tr} \left\{ \mathbf{A}_c^{-1} \mathbf{A}_c^{-1} \right\}.
\]

Note that \( \tau_y \) does not depend on the training sequence, so we only need to minimize \( \text{Tr} \{ \mathbf{A}_c^{-1} \mathbf{A}_c^{-1} \} \), subject to the constraint of a fixed training sequence energy—i.e., \( \sum_{k=0}^{P-1} |c(k)|^2 \) is constant. This kind of training sequence has previously been used in the ST context [6], [8], but for different reasons. In [6], they were called optimum channel independent (OCI) sequences and were designed so as to make the MSE of the channel estimates independent of the channel characteristics. In [8], they were used to minimize the MSE of the channel estimates in the dc-offset free case. Now we have shown that these sequences are also optimal for training sequence synchronization, and we will use them throughout the rest of this paper. The fact that these OCI training sequences are now optimal for both channel estimation and synchronization intuitively rests upon the importance of the transformation \( \mathbf{C}^{-1} \mathbf{y} \) for all aspects of the algorithm. Since in practice, as we said before, \( \mathbf{y} = \hat{\mathbf{y}} - \mathbf{e} \), then we wish to choose \( \mathbf{C} \) to minimize \( \mathbb{E}[\|\mathbf{C}^{-1} \mathbf{e}\|^2] \), which leads to the minimization of \( \text{Tr} \{ \mathbf{C}^H \mathbf{C} \} \), which implies \( \mathbf{C}^H \mathbf{C} = \mathbf{P} \sigma_n^2 \mathbf{I} \)—i.e., OCI sequences.

**B. Channel Estimation Error Power Assuming Perfect Synchronization**

The channel estimation error power can be defined as \( \mathbb{E}[\|\mathbf{h} - \mathbf{h}\|^2] \), where for \( \mathbf{h} \) we will use (9) but with \( \mathbf{y} \) replaced by \( \hat{\mathbf{y}} \) and \( \tau \) replaced by \( \hat{\tau} \). To start with, assume that \( \tau_n = \tau P + \tau \) is known, and so \( \mathbf{h} = \mathbf{y} \mathbf{G} \), where \( \mathbf{G} \) is defined as in (9) and \( \hat{\mathbf{y}} \) is estimated from \( \{x(k)\}_{k=0}^{N-1} \). Now

\[
\mathbb{E} \left[ \| \mathbf{h} - \mathbf{h} \|^2 \right] = \mathbb{E} \left[ \| \mathbf{h} - \mathbb{E}[\mathbf{h}] \|^2 \right] = \sigma_h^2 = \text{Tr} \{ \mathbf{T}^H \mathbf{T} \mathbf{R}_{\mathbf{y}} \}
\]

where \( \mathbf{R}_{\mathbf{y}} = (1/N_P)(\mu \sigma_n^2 \epsilon_h + \sigma_n^2) \mathbf{I}_P \), with \( \mu = 1 \) for ST and 0 for DDST. This follows from (17) and (18).

\[
\sigma_h^2 = \frac{1}{N_P} (\mu \sigma_n^2 \epsilon_h + \sigma_n^2) \text{Tr} \{ \mathbf{T}^H \mathbf{T} \}
\]

where \( \epsilon_h \) is defined for (17). In addition, note that the variance of the channel estimate in (25) depends on \( \mathbf{G} \). Remember that \( \mathbf{G} \) is given by either (10) or (11), depending upon whether a dc-offset is present or not. So using the property that \( \mathbf{C}^{-1} = (1/P) \sigma_n^2 \mathbf{C}^H \), then the variance of the channel estimate for both cases can now be succinctly expressed as

\[
\sigma_h^2 = \frac{M}{N} (1 + a) \frac{\mu \sigma_n^2 \epsilon_h + \sigma_n^2}{\sigma_n^2}
\]

where \( a = 1/(P - M) \) in the first case (unknown dc-offset) and \( a = 0 \) otherwise (known or nonexistent dc-offset). Note that in the latter case, (26) does not depend on the ST sequence period \( P \). This was previously reported in [8], but via a different analysis. Note that (4) is a linear system of \( P \) equations and \( M + 1 \) unknowns, and so \( P - M \) gives an idea of how overdetermined the system is. As the overdetermination \( (P - M) \) increases, then \( a \rightarrow 0 \); i.e., for large \( P - M \), it does not matter if there is an unknown dc-offset. Finally, for \( a = 0 \), \( P = M \), and \( \mu = 1 \)—or, equivalently, for known dc-offset, a “square” system of equations in (4), and the ST implementation—we recover the expression for the channel estimation variance originally reported in (6).

If frame/training sequence synchronization is not provided but has to be estimated using the proposed technique, two situations may arise. First, the synchronization technique achieves perfect synchronization so \( \sigma_h^2 \) is computed as before. Secondly, the synchronization technique fails (\( \hat{\tau} \neq \tau \)), which implies the following.
1) The estimate of $y$ is not the best one in the DDST scheme given that samples of two contiguous received blocks are used. On the other hand, the variance of the estimate of $y$ does not worsen nor improve in the ST scheme when samples of two contiguous blocks are used. In this sense, with respect to the estimation of $y$, the ST scheme is not affected.

2) If $\hat{\tau} \neq \tau$ [see (A7)], then the dc-offset and channel estimates based, respectively, on (7) and (8) will be biased.

3) If $\hat{\tau} = \tau$ [see (A7)] but $\tau_y \neq \tau_p$, then (7) and (8) will give unbiased estimates but with higher variances than when block synchronization is also achieved, as shown previously in this section (and the Appendix). Again, this only affects DDST, as lack of BS does not affect the variance of ST estimates.

For a comprehensive analysis of $\sigma_y^2$ when $\tau_y = \tau_p P + \tau \neq 0$ is unknown, we really need the probability density functions for $\hat{\tau}$ and $\hat{\tau}_p$, and that is outside the remit of this paper.

C. About the Use of Existing Synchronization Methods

There already exist some general methods for TSS that make use of traditional TDM training sequences. These methods could have been applied to the ST scenario, but we contend that their success could be greatly impaired by the fact that as ST is not a TDM method, so the training sequence is severely corrupted by the addition of the information sequence (see Fig. 1). So a better performance is expected from the proposed method where additionally the periodicity of the training sequence is taken into account—i.e., this new method is specifically designed for the ST implementation.

For the particular case of DDST, synchronization methods that make use of the existence of a cyclic prefix could have been applied, but given the short length of the cyclic prefix many blocks would be needed to obtain a close estimate to the synchronization offset. The proposed method can easily achieve synchronization using just one block, given that it uses the redundancy available in the training sequence (i.e., its periodicity). Again, the advantage of the proposed method is that it is specifically designed for the DDST scheme.

VI. SIMULATIONS

The performance of the proposed method for synchronization, dc-offset removal and channel estimation is tested with a channel output $x(k)$ in (2) obtained via the following settings (taken from [7] for comparison). The channel $h(k)$ is a three-tap complex Rayleigh fading: both the real and imaginary parts of the channel taps follow a normal distribution, rescaled to achieve unit mean energy channels [in (17), $e_l = 1$]; $h(k) \in \{-1,+1\}$, each value with equal probability; dc-offset $m_0 = \sqrt{\frac{1}{P}}$, $P = 7$, $\sigma_c^2$, and $\sigma_{\tilde{x}}^2$ are chosen for ST and DDST independently such that $\sigma_{\tilde{x}}^2 + \sigma_c^2 = 1$ and $\sigma_{\tilde{x}}^2/\sigma_{\tilde{x}}^2 = 0.2$; $x(k)$ is constructed as explained in [6] such that its DFT bins have equal magnitude (see Section V-A) and unity peak to average power ratio; and $N = 300$. Following [10], a cyclic prefix of length $P$ is added at the beginning of each block for an optimum performance of the DDST method. To compute the BER, prior to the filtering through the equalizer, each $N$-sample-long record of received samples $x(k)$ has its periodic mean removed, i.e., for the $\{x(k)\}_{k=0}^{N-1}$ received, the input to the equalizer becomes $x'(iP + j) = x(iP + j) - (1/(N_P + 1)) \sum_{j=0}^{N_P} x(iP + j)$, where as usual $i = 0, 1, \ldots, N_P$ and $j = 0, 1, \ldots, P - 1$. We then use $12 \cdot 10^4$ samples to compute the BER. The channel estimate using the proposed method is used to design the minimum mean squared error (MMSE) equalizer of length 11. The MMSE equalizer operates using its optimum delay: for a given delay $d$, an MMSE equalizer is computed $\tilde{w}_d$, then the optimum delay $d_{opt}$ is that minimizing the MSE, where $\tilde{h}$ is assumed to be the true channel, $d_{opt} = \arg \min_d \{\sum_{k=-\infty}^\infty |b(k \bar{d} - \tilde{h} \ast \tilde{w}_d)(k)|^2\}$. Finally, unless otherwise stated, these parameters will be used for all following simulations.

The cyclic mean removal (previously mentioned) ensures that the input to the equalizer is free of $e(k)$ and $m_0$, as can be verified by subtracting (3) from (2). Then, the output of the equalizer (equalized signal) will ideally consist only of $b(k)$ and $e(k)$—i.e., $s(k)$ in (1) with no training sequence (already removed). Note then that $e(k)$ still corrupts the desired information sequence $b(k)$. We will refer to this corrupting effect of $e(k)$ as data-dependent distortion (DDD), and we will see later how to reduce its effect. Finally, 1000 different channel realizations are averaged to compute both the MSE and BER.

The effect of the “channel identification delay” (as commented upon in Section III-C and not to be mistaken with the known optimum delay ($d_{opt}$) in the design of the MMSE equalizer) has a similar effect as an unknown equalization delay. To provide a meaningful BER measurement, this unknown equalization delay has to be accounted for. The equalization delay is computed by comparing the equalized sequence, shifted by different delays, with the actually transmitted symbols $\{b(k)\}$ in order to compute the BER. Then the delay providing the smallest BER is the equalization delay used. This delay is then used to recompute the correct MSE of the channel estimates.

So first the channel estimate MSE obtained is shown in Fig. 3. The ST method in [7] and the method presented here (for the ST version) actually provide identical results, and both are very
close to the MSE obtained when we use ST with perfect synchronization and known dc-offset—i.e., (26) with $a = 0$ and $\mu = 1$. Nevertheless, as we have already commented, the computational burden of our proposed ST method is greatly reduced compared to the ST method in [7]. When it comes to the proposed method with the DDST implementation, the MSE is substantially improved, and again the MSE is not far from the situation with perfect TSS and BS and known dc-offset—i.e., (26) with $a = 0$ and $\mu = 0$. To show how important BS is in the performance of DDST, in Fig. 3 the MSE for DDST with only TSS (the BS step is skipped in the proposed method) is also shown. It is clear that lack of BS in DDST gives substantial reduction in performance. However, as we predicted theoretically in Section V (and the Appendix), even without BS, DDST still provides better channel estimates than conventional ST.

The same analysis of the MSE performance is valid for the BER performance shown in Fig. 4. Note that channel MSE performance is secondary, and the key metric is BER. Additionally, note that the BER for DDST can be further improved by using the iterative method to remove DDD as explained next (as presented in [10]). The sequence $e(k)$ in (1) can be estimated after the received block is filtered through the equalizer and a hard-decision step is made to give $h_{\text{hd}}^{(0)}(k)$. So the estimate of $e(k)$ is $e^{(0)}(j) = -(1/N_P) \sum_{i=0}^{N_P-1} b_{\text{hd}}^{(0)}(iP+j)$. This value of $e^{(0)}(j)$ is then subtracted from the equalized received block before proceeding with a new hard-decision process to give $h_{\text{hd}}^{(1)}(k)$, which allows for a new iteration of the technique if desired. It should be noted that nearly all the gain is obtained after the first step, so no more than one iteration is usually required. This improvement is clear in Fig. 4.

Furthermore, Fig. 5 shows the symbol error rate (SER) performance of the proposed methods when the input to the channel is now a quadrature phase-shift keying (QPSK) sequence. As expected, we can notice an increase in the SER for all methods. For higher order modulations, as the constellation points move closer together, the SER will continue to increase. In addition, the advantage of DDST over conventional ST will reduce, due to the increasing influence of the $e(k)$ DDST term on SER performance when using closer spaced constellation points. Finally, the channel MSE is of course not affected by the type of modulation of the input, as was noted in [6].

A. Time-Varying Channels

Real-life channels are time-varying, but in this paper, the methods have been developed assuming approximately stationary environments over $N$ sample blocks. Now we desire to test how these methods behave for time-varying channels. So, we consider binary phase-shift keying (BPSK) input sequences in the same scenario as before but now the channel is time-varying following the Jakes’ model. The carrier frequency is set to $f_c = 900$ MHz and the sampling rate to $f_s = 400$ kbps (assuming a bandwidth of an individual channel in GSM of 200 kHz). The BER obtained with the proposed methods when the relative speed between the transmitter and the receiver is $v = 40$ Km/h and $v = 160$ Km/h is shown in Fig. 6. As expected, the degradation in the BER increases with the speed $v$. The difference in performance between ST and DDST reduces as well, since the underlying DDST principle does not hold for a nonstationary channel. It is important to remark that the method in [7] gives the same performance as the ST method proposed here, and so it is not shown in Fig. 6.

In summary, it is for time-varying channels that ST (and, more importantly, DDST) methods can be real contenders to TDM methods. Note that for stationary channels, in general any channel estimation needs to be done only once at the beginning of the transmission, and so the loss of bandwidth associated with TDM is only suffered once. On the other hand, (DD)ST methods are always transferring part of the input signal power to the training during the whole transmission process, but this is not needed if working with stationary channels.
B. Comparison With Time-Division Multiplexed Training

ST methods have been proposed in this paper for channel estimation. Traditionally, the training sequence was time-division multiplexed with the information sequence. Here we compare both training schemes in a simplified scenario of perfect synchronization and known dc-offset. So, the first question that arises is how to make a fair comparison. We have chosen to force DDST and TDM to provide the same channel estimation error, and then to compare the BERs of both methods. Now, it can easily be shown that the channel estimation error for TDM is

\[ \sigma_h^2(\text{TDM}) = \frac{M \sigma_n^2}{N_t \sigma_i^2}, \]  

(27)

Comparing (26) with (27), then DDST and TDM will have the same channel estimation error if \( N_t \sigma_n^2 = N_t \mu = \alpha = 0 \) in (26), \( N_t + M - 1 \) is the length of the TDM training sequence with power \( \sigma_i^2 \), and \( \sigma_h^2 = 1 \) (as the total transmit power is set to unity).

Note that to estimate the channel under the TDM scheme, the memory of the channel must already be full and so an \( N_t + M - 1 \)-length training sequence is required with \( N_t \geq M \). Finally, note that for DDST, in addition to \( N_t \) samples, we require an \( (M-1) \)-length cyclic prefix.

Fig. 7 shows the BER versus the SNR for DDST and TDM for the same scenario as before, with TDM variables chosen as explained above. We can see that for DDST, there is almost no significant BER loss in performance compared to TDM. But note that the bandwidth loss for DDST is also minimal (just \( M - 1 \) samples compared to \( N_t + M - 1 \) for TDM). Furthermore, in this scenario, the computational burden is reduced from \( N_t(M^2 + M) + M^2 \) required by TDM down to \( M^2 + M \) required by DDST.

C. Semiblind Channel Estimation and Detection Using ST

Finally, as we remarked at the beginning of Section V, the iterative ST method of [13] is a recent development. So how does its performance compare with conventional ST or DDST? The method in [13] feeds the equalized symbols obtained from the ST method into a traditional least squares (LS) approach, where now both data and training are assumed known. The symbols equalized after the LS approach can be fed back into the LS approach in an iterative manner. Two iterations are usually required to reach convergence. Fig. 8 shows the channel estimation MSE for the conventional ST and DDST and the ST scheme in [13]. For reference, the MSE when all transmitted power is devoted to training is also shown. As expected, the ST scheme in [13] converges to a fully trained system for high SNR, thus clearly outperforming both conventional ST and DDST.

The corresponding BER is displayed in Fig. 9. Again, the ST scheme of [13] is compared with conventional ST and DDST. For reference, the case of a known channel with all the transmitted power assigned to the data is included. Although the ST scheme of [13] outperforms both the conventional ST and DDST, as the gap left for improvement between DDST and the
when, then the estimate, as was used in Section V. In will depend with re-
— is the number of samples that enter in the
for mathematical simplicity, and all new blocks will
if before the transmis-
was assumed. While a cyclic pre
instead of
of channel taps. We showed that these
estimates of the first-order cyclostationary statistics of the
channel output are obtained under the DDST scheme, even if
synchronization is not provided, compared to those obtained
under the conventional ST scheme. This means 1) better syn-
chronization and 2) better channel estimates for DDST than
for plain ST, even if DDST has no BS. Semiblind approaches
based on ST require a different analysis not carried out here.

Simulations showed that the proposed method for ST
synchronization (which only requires TSS) had a behavior
comparable to that of existing algorithms but with a signi-
ficantly reduced computational burden. We also found that the
DDST synchronization method (which requires both TSS and
BS) for channel estimation and equalization proposed here out-
performs all existing conventional ST schemes and still retains
a smaller computational burden when compared with existing
ST methods. The advantages of DDST over ST reduce as the
modulation order and/or vehicle speed increases—as expected.

Finally, we also showed that the semiblind approach of [13]
based on iterative ST outperforms DDST but at the expense of an
LS-like computational burden. In addition, DDST gave a sim-
ilar performance as TDM channel estimation, but with a much
smaller loss in bandwidth.

APPENDIX

This Appendix is concerned with calculating the variance of each element of \( \hat{\mathbf{y}} \)—i.e., \( \sigma^2_{\hat{y}(j)} \) as was used in Section V. In particular, we are concerned with the synchronization offset \( \tau_s \) [see (2)] and its effect on \( \sigma^2_{\hat{y}(j)} \). Thus we will explain why DDST (even without synchronization) can be expected to outperform
traditional ST (see claims C1 and C2) at the start of Section V).

Before computing the variance of \( \hat{\mathbf{y}} \) in the DDST setup, we
need to know which samples enter into the estimation of \( \hat{\mathbf{y}} \) when we have a synchronization offset. By design, the data-depen-
dent training sequence \( \ell(k) \) makes the DDST estimates in (14)
independent of the data sequence \( h(k) \) if before the transmis-
sion of each block in (1) a cyclic prefix is sent [10]. Recall that in (14) \( \tau_s = 0 \) was assumed. While a cyclic prefix of length
\( M - 1 \) would be enough to achieve independence of \( \hat{\mathbf{y}} \) with re-
spect to \( h(k) \), in this Appendix the cyclic prefix is assumed to be
of length \( P \) for mathematical simplicity, and all new blocks will
now include a cyclic prefix. Note that to keep the notation con-
sistent, \( N \) is now the number of samples in a block (excluding
the cyclic prefix), so now we have \( N_P + 1 \) periods in total in a
block (including the cyclic prefix), where as usual \( N_P = N/P \).

Note as well that \( N \) is the number of samples that enter in the
estimation of \( \mathbf{y} \). If \( \tau_s \neq 0 \), then the estimate \( \hat{y}(j) \) will depend
on \( h(k) \), and this will affect the variance of the estimate.

Assume (as before) that the synchronization offset is \( \tau_s =
\tau_P P + \tau \), where now \( 0 \leq \tau_P \leq N_P \) instead of \( 0 \leq \tau_P \leq
N_P - 1 \), because we have included a cyclic prefix. This means
that one of the received blocks starts with \( x(\tau_s) \). Let us call the

\( \tau_s \)
received block comprising \( \{x(k)\}^{T_x+N+P-1}_{k=0} \) received block 1." For the sake of clarity in the nomenclature, let us propose the following one-to-one mapping, in general, for the \( r \)th received block:

\[
\{x(k)\}^{T_x+r(N+P)-1}_{k=0} \Rightarrow \{x_r(iP+j)\}_{i=0}^{N+P-1}
\]

This means that, in order to distinguish between different received blocks, a subscript \( r \) (denoting "received") is added to each variable as needed. So, for example, \( x_r(iP+j) \) is the \((iP+j)\)th element of the \( r \)th received block. From now on, we will slightly alter the nomenclature to accommodate the cyclic prefix, which we will designate as period \( j = 0 \) within the transmitted block, while period \( i = 0 \) within a received block will be referred to as well as the cyclic prefix period. Ideally, \( y \) should be computed from

\[
\hat{y}(j) = \frac{1}{N_P} \sum_{i=1}^{N_P} x_r(iP+j)
\]  

(28)

for any \( r \), so that DDST benefits from the "data" noise cancellation, as previously explained. But actually \( \hat{y}(j) \) is computed in practice from the samples of \( x(k) \) included in (13) and depicted in Fig. 10.

Now the expression for the variance of \( \hat{y}(j) \) for DDST—i.e., \( \sigma^2_{\text{DDST}} \)—is complicated and can be shown to depend on \( j, M, \) and \( \tau_x \) or, equivalently, on \( j, M, i_\tau, \) and \( j_\tau \) where \( i_\tau \) and \( j_\tau \) are defined in Fig. 10. The expression for \( \sigma^2_{\text{DDST}} \) for specific values of \( j, M, \) and \( \tau_x \) is easily computed but is not shown here due to lack of space. Nevertheless, to give the whole picture and as an illustration, we will plot \( \sigma^2_{\text{DDST}} \) versus \( i_\tau \) for one particular case. This is because \( \sigma^2_{\text{DDST}} \) as a function of \( i_\tau \) gives useful information about how including samples of two consecutive received blocks affects the estimation of \( y(j) \). It can be shown as well that there are four different forms for \( \sigma^2_{\text{DDST}} \) as a function of \( i_\tau \). One of these forms can be found in Fig. 11, where we require the following parameters in order to plot it:

\[ N_P = 57, P = 7, M = 3, \sigma^2_{h} = 0.848, \sigma^2_{f}(l) = E[h(l)|f] = 1/3, \] 

and \( \sigma^2_{n} = 10^{-3.5} \). The variances \( \sigma^2_{\text{DDST}} \) (for both ST and DDST) shown in Fig. 11 are the theoretical expressions; \( \sigma^2_{\text{DDST}} \) comes from (17), \( \sigma^2_{\text{DDST}} \) (for known \( \tau_x \)) comes from (18), and \( \sigma^2_{\text{DDST}} \) (for unknown \( \tau_x \)) can easily be derived (but is not shown here due to lack of space). Note that we have not included simulations, since they lie exactly (as predicted) on the theoretical curves.

Recall that (17) and (18) have already been commented upon in Section V, but now in Fig. 11 we can graphically see the advantage of DDST (for known \( \tau_x \)) over ST. The third function plotted in Fig. 11 is \( \sigma^2_{\text{DDST}} \) (for unknown \( \tau_x \)) as a function of \( i_\tau \). The shape for \( \sigma^2_{\text{DDST}} \) in Fig. 11 is quite similar to the rest of the other three forms. When \( i_\tau \) is close to zero, we use nearly only one received block to estimate \( y \) (see Fig. 10) and so \( \sigma^2_{\text{DDST}} \) (for unknown \( \tau_x \)) behaves as \( \sigma^2_{\text{DDST}} \) (for known \( \tau_x \)). The same reasoning can be applied to the point \( i_\tau = N_P \) (see Fig. 11). For intermediate values of \( i_\tau \), the increase in the

Fig. 10. Structure of two consecutive received blocks (as defined in the Appendix)—in particular, received block 0 and 1. Each received block can be divided, similarly to a transmitted block, into \( N_p + 1 \) periods. So, the periods labelled as \( j = 0 \) can be viewed as "received" cyclic prefixes. Ideally, \( y \) should be computed using periods \( i = 1, 2, \ldots , N_P \) within any received block. Due to the offset \( \tau_x = \tau_x P + \tau \) in (2), \( y \) in (13) is actually computed from an \( N \)-sample-long record starting at sample \( j_r = P - \tau \) of the \( \tau_x = N_P + 1 - \tau_x P \) period within received block 0.
mixing between received blocks worsens $\sigma^2_{\text{ST},(j)}$, pushing it towards the $\sigma^2_{\text{ST}}$ value, but it still never exceeds it. This supports claims (C1) and (C2) given at the beginning of Section V: that even without proper synchronization, DDST outperforms ST.

ACKNOWLEDGMENT

The authors would like to acknowledge the reviewers for their helpful suggestions, which they believe improved the quality of this paper.

REFERENCES


Enrique Alameda-Hernandez was born in Granada, Spain, in 1975. He received the B.Sc. and Ph.D. degrees in physics from the University of Granada, Spain, in 1998 and 2003, respectively.

In December 2003, he joined the University of Leeds, Leeds, U.K., as a Postdoctoral Fellow for a two-year period working on signal processing for communications. His research interests include adaptive signal processing, channel estimation, and equalization.

Des C. McLernon (M’94) was born in Downpatrick, Northern Ireland. He received the B.Sc. degree in electronic and electrical engineering and the M.Sc. degree in electronics from Queen’s University of Belfast, Northern Ireland, in 1975 and 1976, respectively, and the Ph.D. degree in two-dimensional signal processing from Imperial College, University of London, London, U.K., in 1982.

He was then with Ferranti Ltd., Edinburgh, Scotland, U.K. After lecturing at South Bank University, London, he joined the School of Electronic and Electrical Engineering, University of Leeds, U.K. His research interests (in which he has over 150 publications) are broadly within the field of signal processing for communications.

Aldo G. Orozco-Lugo (S’96–M’00) was born in Guadalajara City, Mexico, in 1970. He received the B.Sc. degree in electronics and communications from the University of Guadalajara, Guadalajara, Jalisco, Mexico, in 1993, the M.Sc. degree in electrical engineering (specialized in communications) from the Center for Research and Advanced Studies, IPN, Mexico City, Mexico, in 1997, and the Ph.D. degree in electrical engineering (specialized in digital signal processing for communications) from the University of Leeds, Leeds, U.K., in 2000.

He is currently a Principal Member of Research Staff with the Center for Research and Advanced Studies, IPN. His research interests include space-time signal processing, wide-band channel modeling, analog and digital communication systems, antenna array technology, and radar.

M. Mauricio Lara (M’90) received the B.Sc. degree from the National Autonomous University of Mexico, Mexico City, Mexico, in 1986, the M.Sc. degree from the Center for Research and Advanced Studies, IPN, Mexico City, Mexico, in 1986, and the Ph.D. degree from the University of Leeds, Leeds, U.K., in 1990, all in electrical engineering.

From 1990 to 1993, he was a Research Fellow with the University of Leeds. During 2001, he was a Visiting Researcher with Turin Politecnico, Italy. Since 1993, he has been with the Center for Research and Advanced Studies as a Researcher. His areas of interest include wireless systems and signal processing for communications.

Mounir Ghogho (M’98) received the Ph.D. degree from National Polytechnic Institute of Toulouse, France, in 1997.

He was a Research Fellow with the University of Strathclyde from 1997 to 2001. He joined the University of Leeds, U.K., in 2002, where he is currently an Associate Professor (Reader). His research interests are in signal processing for communications and networking.

Dr. Ghogho was an Associate Editor of IEEE SIGNAL PROCESSING LETTERS from 2001 to 2004. He is currently an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING and a member of the IEEE SPSCOM and SPTM Technical Committees. He received the Royal Academy of Engineering Research Fellowship in September 2000.