On the Design of Pilot Symbol Assisted Codes

Nicholas Bonello, Sheng Chen and Lajos Hanzo

School of ECS, University of Southampton, SO17 1BJ, United Kingdom.

Email: {nb06r,sqc,lh}@ecs.soton.ac.uk, http://www-mobile.ecs.soton.ac.uk

Abstract— We propose a novel technique, hereby referred to as pilot symbol assisted coding (PSAC), where a predetermined fraction of binary pilot symbols is interspersed with the channel-coded bits at the channel coding stage, instead of multiplexing the pilots with the data symbols at the modulation stage, as in classic pilot symbol assisted modulation (PSAM). We will subsequently demonstrate that the PSAC-aided scheme succeeds in gleaning more beneficial knowledge from the inserted pilots, because the pilot bits are not only useful for estimating the channel at the receiver, but they are also beneficial in terms of significantly reducing the computational complexity of the channel decoder. Our results suggest that in the specific application example considered the PSAC-aided scheme requires up to 45% fewer message-passing updates per decoded bit than the corresponding channel coded PSAM benchmarker scheme.

I. INTRODUCTION

There are two dominant approaches that are frequently employed to estimate the channel; namely that of either estimating the channel blindly or using reference/pilot symbols. Typically blind channel estimation techniques impose a high complexity and suffer from a performance degradation as well as from a slow rate of convergence. On the other hand, the insertion of known pilot symbols into the transmitted data stream using pilot symbol assisted modulation (PSAM) potentially achieves an improved bit error ratio (BER) performance, at the expense of an unavoidable reduction of the effective throughput due to the associated pilot overhead. PSAM was conceived by Moher and Lodge [1] as an alternative technique to the use of transparent tones-in-band (TTIB) [2]. Closed form expressions for the BER using binary phase shift keying (BPSK) and quadrature phase shift keying (QPSK) modulation schemes as well as tight upper bounds on the symbol error ratio (SER) for 16-quadrature amplitude modulation (QAM) were then provided by Cavers in [3].

In this paper, we will assume that the receiver estimates the channel's amplitude and phase using known pilots; however, instead of inserting pilots at the modulation stage as in classic PSAM, we are proposing a novel technique, hereby referred to as pilot symbol assisted coding (PSAC), where a predetermined fraction of binary pilot symbols is appropriately interspersed with the channel-coded bits at the channel coding stage. Recall that in a channel-coded PSAM scheme, the pilot bits are only beneficial for estimating the channel and so they are removed after channel estimation. On the contrary, we will demonstrate that the pilot bits in the proposed PSAC technique are also valuable for enhancing the performance of the decoder, hence they are retained and further processed by the decoder. The motivation behind using pilot symbol assisted codes is therefore that of gleaning more "profit" from the pilot overhead "investment", than just simply the capability of channel estimation such as in the PSAM technique. Our technique is generic, since it can be applied to any iterative decoding (ID)-aided channel coding scheme. In fact, we recently discovered that a somewhat similar technique was employed in the context of regular low-density parity-check (LDPC) codes [4]. In this paper, we generalize the technique to also include non-systematic codes. We believe that the benefits of the proposed technique are more pronounced in non-systematic codes, whose ID process typically requires the employment of code

doping for triggering their ID convergence. We will be referring to that channel code employing the proposed PSAC technique as a pilot symbol assisted (PSA) code.

The rest of this paper is organized as follows. Section II details the PSA encoder. Furthermore, we derive the lower bound on the achievable rate as well as the throughput and provide a graph-based description of a specific PSA code. The extrinsic information transfer (EXIT) functions of PSA codes are then derived in Section III. In Section IV, we have detailed the specific algorithm that was employed for optimizing the proposed PSA code; i.e. to find the specific check and variable node distributions that maximize the effective throughput. Finally, Section V presents our simulation results, followed by our final conclusions in Section VI.

II. THE PILOT SYMBOL ASSISTED ENCODER

We elected to consider the application of the proposed PSAC technique in the context of non-systematic repeat-accumulate (RA) codes [5]. Let the coding rate and the (irregular) check node degree distribution be represented by R and $\delta(x)$, respectively. The latter can be conveniently represented by means of a polynomial distribution defined by:

$$\delta(x) := \sum_{\forall d_c \in \mathbf{d}} \delta_{d_c} x^{d_c - 1},$$
$$= \delta_1 + \delta_2 x + \ldots + \delta_{d_c} x^{d_c - 1} + \ldots + \delta_{D_c} x^{D_c - 1}, \quad (1)$$

where the positive coefficients δ_{d_c}, $d_c \in \mathbf{d}$ denote the particular fraction of intermediate bits (or check nodes) of degree d_c and $D_c = \max(\mathbf{d})$ is the maximal check (left) degree. The vector $\mathbf{d}$ contains the range of (check) degree values of the degree distribution. In contrast to conventional (non-PSAC) RA codes [5], there is now two different categories of degree-one bits and as a result, the fraction δ_1 in (1) can be rewritten as:

$$\delta_1 = \delta_1^p + \delta_1^{\overline{p}}, \quad (2)$$

where δ_1^p and $\delta_1^{\overline{p}}$ denote the fraction of degree-one nodes corresponding to pilot bits and to information bits, respectively. The fraction δ_1^p that inserts K_p pilots would therefore represent the pilot overhead that is calculated in order to guarantee the effective sampling of the channel's complex-valued envelope at a rate that is higher than the Nyquist rate.

This PSA encoder, also illustrated in Figure 1, will then map a K-bit (input) information sequence represented by $\mathbf{a} = [a_1, a_2, \ldots, a_K]$ into a $(K' R^{-1})$-bit output sequence $\mathbf{c}$, where $K' = K + K_p$, by first attaching a predetermined[1] pilot-bit sequence $\mathbf{p} = [p_1, p_2, \ldots, p_{K_p}]$ to the beginning of the K-bit input stream $\mathbf{a}$, so that the modified K'-bit input sequence becomes $\mathbf{a}' = [\mathbf{p} \ \mathbf{a}]$. We assume a regular variable node distribution, defined by $v(x) := x^{d_v - 1}$, where d_v denotes the variable node degree. Consequently, each bit in $\mathbf{a}'$ is then repeated d_v times, as shown in Figure 1. The value of each bit in the intermediate (check) bit sequence represented by $\mathbf{b}$ is then calculated by first choosing a degree $d_c \in \mathbf{d}$ randomly from the degree distribution $\delta(x) - \delta_1^p$, then randomly selecting this d_c value of bits from the repeated $\mathbf{a}'$ bits, and finally combining these selected bits using modulo-2 addition.

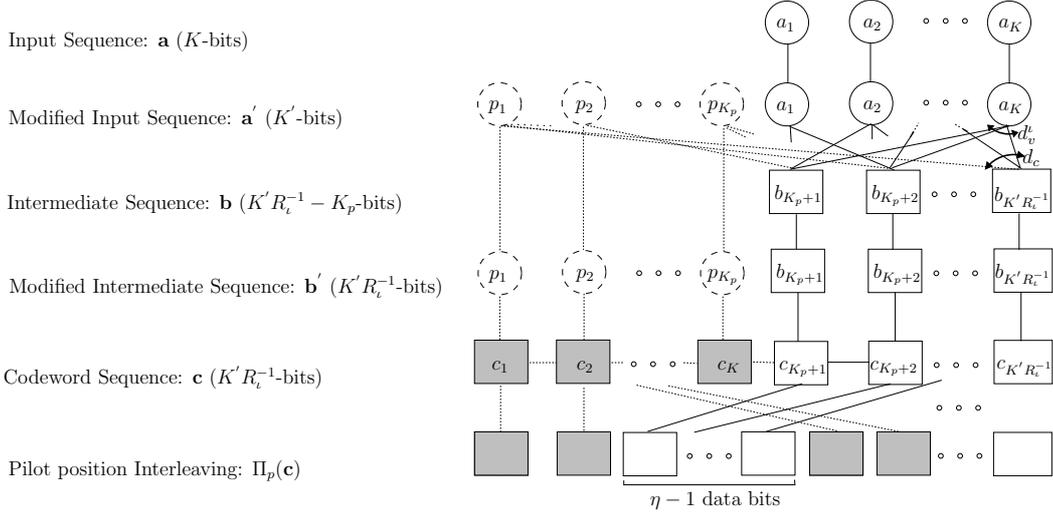[1]In our case, we have employed an alternating sequence of ones and zeros.

Fig. 1. The pilot symbol assisted encoder of rate R based on the encoder of a non-systematic RA code [5].

As observed from Figure 1, K_p number of bits denoting the pilot-bit sequence $\mathbf{p}$ are left unselected and uncombined;[2] these are then attached to the beginning of $\mathbf{b}$, thus creating $\mathbf{b}' = [\mathbf{p}\ \mathbf{b}]$. Finally, the value of the each codeword bit $c_i \in \mathbf{c}$, $i = 1, \ldots, K'R^{-1}$ is determined by:

$$
\begin{aligned}
c_i &= b_i' & i = 1, \\
&= b_i' \oplus c_{i-1}' & i = 2, \ldots, K'R^{-1},
\end{aligned}
\tag{3}
$$

where $b_i' \in \mathbf{b}'$ and $\oplus$ represents the modulo-2 addition operation. The pilot bits in $\mathbf{c}$ correspond to the bits $c_i \in \mathbf{c}$ with $i = 1, \ldots, K_p$, whose value is now equal to one due to the accumulator process of (3). Figure 1 also depicts what we refer to as the the pilot position interleaver Π_p, which spaces a pair of pilots $(\eta - 1)$ data bits apart, where η denotes the pilot spacing. The data bits are separated by a pair of pilot bits instead of a single pilot, since we are going to consider a single-user multiple-input multiple-output (MIMO) system employing two transmit and two receive antennas. Hence a total of four channels have to be estimated using the two pilots. The interleaved codeword $\pi_p(\mathbf{c})$ is then modulated and re-encoded using a rate-one Alamouti space-time block code (STBC) [6].

A. Lower Bounds on the Realizable Rate and the Achievable Throughput

In the previous section, the number of bits in the modified input bit sequence $\mathbf{a}'$ was denoted by

$$
K' = \left|\mathbf{a}'\right| = K + K_p,
\tag{4}
$$

where we have $K = |\mathbf{a}|$, while K_p represents the number of pilot bits and $|\cdot|$ denotes the cardinality of a set. The number of pilot bits required is given by $K_p = \delta_1^p K' R^{-1}$. Therefore, we have a recursive process represented by

$$
K' = K + \delta_1^p K' R^{-1},
\tag{5}
$$

which can also be represented by means of a geometric series having a scale factor of K and a common ratio of δ_1^p / R, yielding

$$
K' = \sum_{i=0}^{\infty} K \left(\frac{\delta_1^p}{R}\right)^i.
\tag{6}
$$

[2]This occurs due to the fact that the degree d_c is chosen from the distribution $\delta(x) - \delta_1^p$ rather than from $\delta(x)$.

It may be readily shown that this series converges to

$$
K' = \frac{KR}{R - \delta_1^p},
\tag{7}
$$

if and only if we have $\frac{\delta_1^p}{R} < 1$, i.e. $R > \delta_1^p$. This implies that the rate of a PSA code must always be higher than the fraction of pilots in the code. At first glance this might appear to be a limitation, however we note that δ_1^p is selected according to the worst expected fading rate, and hence for slow-fading channels PSA codes can practically realize codes having any rate. Moreover, δ_1^p is never chosen to be excessively large, since this will considerably reduce the effective throughput $T_{\text{effective}}$, measured in bits/second/Hz, which is given by $T_{\text{effective}} = R - \delta_1^p$. The number of pilot symbols required according to the pre-determined pilot overhead δ_1^p is obtained by substituting (7) in (4), thus giving $K_p = K\delta_1^p / (R - \delta_1^p)$.

B. Graph-Based Analysis of Pilot Symbol Assisted Codes

A Tanner graph representation of a PSA code is provided in Figure 2, which shows an unbalanced tripartite graph $\mathbb{G}$ consisting of the finite set of vertices V and the finite set of edges E. The vertices set V can be further divided into three disjoint sets representing the variable nodes, the check nodes and the parity nodes. Following the notation introduced in Section II and Figure 1, the variable (information) nodes would then correspond to $\mathbf{a}'$, the check (intermediate) nodes are represented by $\mathbf{b}'$, whilst the parity nodes relate to the (pilot-bit-containing) codeword bits $\mathbf{c}$. Given the graph $\mathbb{G}$, let $\mathbb{G}(v)$ will then denote the set of vertices adjacent to the vertex $v \in V$. The degrees d_v and $d_c \in \mathbf{d}$ correspond to the discrete values assumed by the variable node distribution $\upsilon(x)$ and the check node distribution $\delta(x)$, respectively.

In contrast to conventional RA codes, the previously described PSA codes also possess what we refer to as pilot nodes and pilot edges. Formally, we have the *pilot variable nodes*, $p_i \in \mathbf{a}'$, where $i = 1, \ldots, K_p$, of degree d_v, having a known value, which hence do not carry any information, as opposed to the remaining variable nodes. Then, the *pilot check nodes*, $p_i \in \mathbf{b}'$, where $i = 1, \ldots, K_p$, are the degree-one check nodes connected by a single edge to the pilot variable nodes. The output of the accumulator contains the *pilot parity nodes*, $c_i \in \mathbf{c}$, where $i = 1, \ldots, K_p$. Finally, we also have the *pilot edges*, seen in Figure 2, consisting of the edges emerging from the pilot variable nodes and those joining the pilot check nodes to the pilot parity nodes. There are a total of $K_p d_v$ pilot
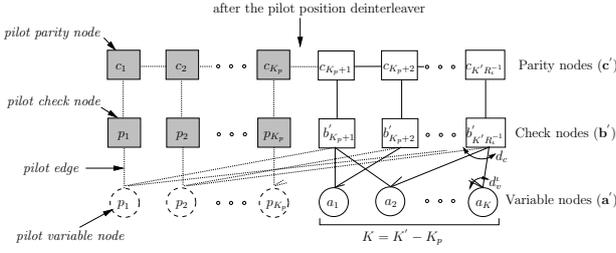
Fig. 2. A tripartite graph representation of a pilot symbol assisted code.

edges between the variable and check nodes, and a further $2K_p$ pilot edges between the check and the parity nodes. It is also important to note from Figure 2, that in order to ensure the initialization of the iterative decoding convergence, the pilot edges sprouting from the K_p pilot variable nodes are not only associated with the pilot check nodes, but are also involved in other parity-check equations containing higher-degree check nodes. The messages passed over the pilot edges are perfectly known, since they originate from nodes having predetermined values.

These three different pilot nodes as well as the pilot edges are explicitly marked in Figure 2. We further distinguish between what we refer to as *transparent* and *non-transparent* pilots. Transparent pilots are those which are neither transmitted nor possess any information content. An example of the transparent pilots are the pilot variable nodes corresponding to a sequence of bits that are known to both the transmitter as well as the receiver. On the other hand, the pilot parity nodes are considered to be non-transparent pilots, since they are transmitted over the channel and input to the channel estimator.

III. EXIT CHART FUNCTIONS OF PILOT SYMBOL ASSISTED CODES

The PSA decoder represented by the tripartite graph of Figure 2 is effectively constituted by the serial concatenation of two decoders separated by a uniform random interleaver. The inner decoder is the amalgam of a memory-one trellis decoder used for the accumulator (ACC) and of a check node decoder (CND), whilst the outer decoder is a variable node decoder (VND). The convergence behavior of this decoding process can then be analyzed in a similar manner to that used for other iterative decoding processes by means of observing the evolution of the input and output mutual information exchange between the inner and outer decoders in consecutive iterations, which is diagrammatically represented using the semi-analytical tool of EXIT charts [7], [8].

The combined EXIT function $I_{E,D\&A\&C}(\cdot)$ of the detector, accumulator and CND can be approximated as in [8] by:

$$I_{E,D\&A\&C}(I_A, I_E, \mathbf{d}, \psi_{\text{avg}}) \approx \sum_{\forall d_c \in \mathbf{d}} \Delta_{d_c}[1 - \quad (8)$$

$$J\left(\sqrt{(d_c - 1) \cdot [J^{-1}(1 - I_A)]^2 + [J^{-1}(1 - I_E)]^2}\right)],$$

where the function $J(\cdot)$ denotes the mutual information, $I_A := I_{A,CND} = I_{A,D\&A\&C}$ represents the *a-priori* information input of the CND and ψ_{avg} denotes the MIMO system's signal-to-noise ratio (SNR). The extrinsic information accumulator output is then defined by $I_E := I_{E,ACC}[I_{A,ACC}(I_{A,CND}, \mathbf{d}), I_{E,D}(\psi_{\text{avg}})]$, where $I_{A,ACC}$ denotes the *a-priori* accumulator information input and $I_{E,D}$ represents the extrinsic information detector output. The parameter Δ_{d_c} in (9) corresponds to the specific fraction of edges emanating from the intermediate bits (or check nodes) of degree

$d_c \in \mathbf{d}$ and is given by

$$\Delta_{d_c} = \delta_{d_c} \cdot \frac{d_c}{d_{c,\text{avg}}}, \quad (9)$$

and the average check node degree $d_{c,\text{avg}}$ is defined by $d_{c,\text{avg}} := \sum_{\forall d_c \in \mathbf{d}} \delta_{d_c} \cdot d_c$. Then, by substituting $\delta_1 = \delta_1^p + \delta_1^{\overline{p}}$ into (9) for $d_c = 1$, the fraction of edges attributed to the degree-one pilot nodes as well as to the non-pilot check nodes is given by $\Delta_1 = \left(\delta_1^p + \delta_1^{\overline{p}}\right) / (d_{c,\text{avg}})$. We note that neither $I_{A,ACC}(I_{A,CND}, \mathbf{d})$ nor $I_{E,D}(\psi_{\text{avg}})$ can be explicitly represented in closed form, thus the functions are evaluated with the aid of Monte Carlo simulations. The functions $J(\cdot)$ and $J(\cdot)^{-1}$ are approximated according to the Appendix of [9].

For the particular case of the proposed PSA codes (and thus in contrast to [5], [8]), the inner decoder's EXIT function $I_{E,D\&A\&C}(\cdot)$ can be analyzed in terms of three separate components as follows

$$I_{E,D\&A\&C}(I_A, I_E, \mathbf{d}, \psi_{\text{avg}}) \approx$$
$$I_{E,D\&A\&C}^1(I_A, I_E, \psi_{\text{avg}}, d_c > 1)$$
$$+ I_{E,D\&A\&C}^2(I_A, I_E, \psi_{\text{avg}}, d_c = 1, \delta_1 = \delta_1^{\overline{p}})$$
$$+ I_{E,D\&A\&C}^3(d_c = 1, \delta_1 = \delta_1^p). \quad (10)$$

The first component of (10) represented by the function $I_{E,D\&A\&C}^1(\cdot)$ is determined by using (9) and by substituting $d_c \in \mathbf{d}$ for all the check nodes that are higher than one. It may be readily shown that the second and third constituent functions of (10) are then approximated by

$$I_{E,D\&A\&C}^2(\cdot) \approx \frac{\delta_1^{\overline{p}}}{d_{c,\text{avg}}}\left[1 - J\left(\sqrt{[J^{-1}(1 - I_E)]^2}\right)\right]$$
$$= \frac{\delta_1^{\overline{p}}}{d_{c,\text{avg}}} I_E, \quad (11)$$

whilst $I_{E,D\&A\&C}^3(\cdot)$ is determined by the multivariable limit formulated by

$$I_{E,D\&A\&C}^3(\cdot) \approx \lim_{(I_A, \psi_{\text{avg}}) \to (1, \infty)} \frac{\delta_1^p}{d_{c,\text{avg}}}\left[1 - J\left([J^{-1}(1 - I_E)]\right)\right]$$
$$= \frac{\delta_1^p}{d_{c,\text{avg}}}. \quad (12)$$

In (12), we are seeking the limit as $(I_A, \psi_{\text{avg}}) \to (1, \infty)$ since the fraction δ_1^p corresponds to the pilot check nodes (please refer to Figure 2), which receive perfect messages from both the pilot parity nodes as well as from the pilot variable nodes. Subsequently, we can substitute (10), (11) and (12) into (9), yielding

$$I_{E,D\&A\&C}(\cdot) \approx \frac{1}{d_{c,\text{avg}}}\left(\delta_1^p + \delta_1^{\overline{p}} I_E\right) + \sum_{d_c > 1} \Delta_{d_c}[1 -$$
$$J\left(\sqrt{(d_c - 1) \cdot [J^{-1}(1 - I_A)]^2 + [J^{-1}(1 - I_E)]^2}\right)]. \quad (13)$$

Given a variable node distribution $v(x)$, the outer decoder's EXIT function representing the extrinsic information output of the VND can then be formulated in a similar manner to that of a non-systematic RA code [8], namely as:

$$I_{E,VND}(I_{A,VND}, d_v) = J\left[\sqrt{(d_v - 1)} \cdot J^{-1}(I_{A,VND})\right], \quad (14)$$

where $I_{E,VND}(I_{A,VND}, d_v)$ represents the extrinsic information output of the VND as a function of the its *a-priori* information input $I_{A,VND}$ and its variable node degree d_v.

Against this background, we elucidate the following two points:

1) The contribution of the pilot variable nodes was ignored in (14), since they represent transparent pilots that are not

conveying any information. This is due to the fact that their actual value is predetermined and known to both the transmitter as well as receiver and thus does not have to be recovered. Consequently, there is no need for exchanging any messages along the pilot edges; both the accumulator, the CND and the VND know with absolute certainty what the messages should be.

2) The initialization of convergence for this iterative decoding process is guaranteed by the term $\frac{1}{d_{c,\text{avg}}}\left(\delta_1^p + \delta_1^{\overline{p}}I_E\right)$ in (13). However, we note that for the case of medium to high SNRs, the optimization technique we have employed (please refer to Section IV) yields $\delta_1^{\overline{p}} = 0$, and thus the initialization of convergence is totally dependent on the pilot nodes. The remaining fraction denoted by $\overline{\delta} := \delta_1^p/d_{c,\text{avg}}$ in (13) - which is contributed by the pilot bits - is independent of both I_A and I_E. Additionally, the pilot bits also contribute to the widening of the open EXIT tunnel between the $I_{E,D\&A\&C}(\cdot)$ curve and the corresponding outer decoder's EXIT function, thus potentially reducing the number of iterations and hence the computational complexity. This occurs due to the fact that this $\overline{\delta}$-fraction is positive and proportional to δ_1^p.

IV. EXIT CHART BASED OPTIMIZATION FOR PILOT SYMBOL ASSISTED CODES

This optimization problem is tackled by the following linear programming approach, with the primal problem formulated by

$$\max \sum_{\forall d_c \in \mathbf{d}} \frac{d_c}{\Delta_{d_c}} \tag{15}$$

subject to the *equality constraint*

$$\sum_{\forall d_c \in \mathbf{d}} \Delta_{d_c} = 1 \tag{16}$$

and to the *inequality constraints* given by

$$I_{E,D\&A\&C}(\mathcal{I}, \mathbf{d}, \psi_{\text{avg}}) > I_{A,VND}(\mathcal{I}, d_v) + \varsigma, \tag{17}$$

and

$$\Delta_{d_c}|_{\forall d_c \in \mathbf{d}} > 0, \tag{18}$$

where (16) and (18) ensures that the resultant Δ_{d_c} values are both valid and non-negative. The parameter $\mathcal{I}$ represents the discrete set of gradually increasing values in the interval $[0, 1]$ over which the functions $I_{E,D\&A\&C}(\cdot)$ and $I_{A,VND}(\cdot) = I_{E,VND}^{-1}(\cdot)$ (please refer to (13) and (14)) are calculated, whilst ς assumes values across $\mathcal{I}$, which determines the area of the tunnel between the two EXIT curves. This area has a direct relationship to the number of iterations required in order the reach the (1,1) point of the EXIT chart. Optimizing the objective function of (15) subject to the above-mentioned constrains, will determine the feasible set of candidate solutions having values of Δ_{d_c} (and consequently δ_{d_c}) corresponding to the specific check node degrees $d_c \in \mathbf{d}$ that substantiate that distribution $\delta(x)$, which maximizes the design rate, for a predefined d_v value.

We remark that the constraints represented in (16), (17) and (18) are sufficient to design conventional RA codes, they are however insufficient to guarantee that the proposed RA-code-based PSA code will provide a δ_1^p-fraction of pilot bits. For this particular reason, a stricter constraint than that of (18) must be introduced for the specific fraction of edges Δ_1 terminating in degree-one check nodes, which must also obey $\Delta_1 \geq \delta_1^p/d_{c,avg}$. The difficulty in satisfying the latter constraint arises from the dependence of Δ_1 on the average check node degree $d_{c,avg}$, where the latter is again

dependent on the value of $d_c \in \mathbf{d}$ as well as on the value of δ_{d_c}, both of which constitute part of the set of solutions for the optimization problem considered. This problem is circumvented by utilizing a search algorithm, similar to a binary search algorithm, which progressively finds better estimates of the required Δ_1 value that results in the required δ_1^p-fraction of pilot bits. We note that a conventional binary search algorithm cannot be directly applied in this scenario due to the continuous nature of Δ_1, which prevents its representation in a sorted array.

The first step of the design technique was that of solving the optimization problem of (15) satisfying the constraints of (16), (17) and (18), and temporarily setting δ_1^p to zero. This initial step is carried out in order to estimate the number of degree one check nodes that are available. The fraction of degree one nodes, δ_1, is then calculated according to (9) and using the Δ_1 value resulting from the first run of the linear program.

For the sake of further explaining the procedure used, we will denote the fraction of edges and nodes calculated after the ith evaluation of the objective function of (15) by $\Delta_{1,i}$ and $\delta_{1,i}$, respectively. Following this, if the resultant initial value $\delta_{1,1}$ is smaller than the target value δ_1^p, the linear program is run again by introducing a fourth inequality constraint given by $\Delta_1 > 2\Delta_{1,1}$. In doing so, the value $\Delta_{1,1}$ is set to be the (temporarily) lowest value of the search interval Δ_1. After the second iteration, which provides the solution for both $\Delta_{1,2}$ and for the corresponding fraction $\delta_{1,2}$, a comparison is made again between $\delta_{1,2}$ and the target fraction of pilots. If the value of $\delta_{1,2}$ is found to be larger than δ_1^p, the value of $\Delta_{1,2}$ is set to be the (temporarily) highest value of the search interval. The search may then continue by solving the objective function of (15) for the third time, with the additional fourth constraint of $\Delta_1 > (\Delta_{1,2} - \Delta_{1,1})/2$. On the other hand, if the calculated value $\delta_{1,2}$ is again smaller than the target value, the value $\Delta_{1,2}$ becomes the new lowest value of our search interval and the additional fourth constraint is twice this lowest value; i.e. $\Delta_1 > 2\Delta_{1,2}$. Following this, every further run of the linear program will enable use to narrow our search interval by a factor of two, until the target value is found. The procedure used is shown summarized in Algorithm 1.

V. SIMULATION RESULTS

We considered binary phase shift keying (BPSK) modulated transmission of $K = 10,000$ bits over a (2×2) correlated Rayleigh MIMO channel. We assumed a mobile terminal of 100 mph. The data signaling rate and the carrier frequency were those from the 3G standard, and were set to 15 kbps and 2 GHz, respectively. The proposed PSAM scheme described above was compared to its RA-coded PSAM counterpart, i.e. an RA code dispensing with pilots (i.e. we set $\delta_1^p = 0$ at the encoding stage, which was previously described in Section II) but then insert the required number of pilots at the modulation stage. In this sense, we are therefore comparing PSAC with PSAM in an attempt to verify which of the two techniques offers a better performance for the same amount of pilot overhead.

Both schemes had an effective throughput of 0.82 bits/s/Hz and a pilot overhead of 10%. The distributions for the channel codes in both schemes were determined by employing the technique of EXIT chart matching as described in Section V, where the RA code in the RA-coded PSAM benchmarker scheme was designed by solving the optimization problem of (15), while satisfying the constraints of (16), (17) and (18). The PSA code was optimized using the aforementioned algorithm, summarized in Algorithm 1. The interleaving of the interconnections of the variable nodes

Algorithm 1: The EXIT chart based optimization of pilot symbol assisted codes

input : d_v, $\mathcal{I}$, ς, δ_1^p, ψ_{avg}

output: Δ_{d_c}, **d**

1   Initializations: $target\ value \leftarrow \delta_1^p$, (iteration) $i \leftarrow 0$

2   **while** $\delta_{1,i} < target\ value$ **do**

3       $i \leftarrow i + 1$

4   **if** $i = 1$ **then**

5   Solve the optimization problem of (15) satisfying the constraints of (16), (17) and (18), and temporarily setting δ_1^p to zero.

6          $\delta_{1,i} \leftarrow \delta_1$, $\Delta_{1,1} \leftarrow \Delta_1$. Set fourth constraint for iteration $i = 2$: $\Delta_1 > 2\Delta_{1,1}$.

7   **else**

8   Solve the optimization problem of (15) subject to the constraints of (16), (17), (18) and the additional fourth constraint set in iteration $i - 1$.

9          $\delta_{1,i} \leftarrow \delta_1$, $\Delta_{1,i} \leftarrow \Delta_1$.

10   **if** $\delta_{1,i} < target\ value$ **then**

11   Fourth constraint for iteration $i + 1$: $\Delta_1 > 2\Delta_{1,i}$.

12   **else if** $\delta_{1,i} > target\ value$ **then**

13   Fourth constraint for iteration $i + 1$: $\Delta_1 > 0.5(\Delta_{1,i} - \Delta_{1,i-1})$.

14   **else**

15   Target value has been reached. Return output parameters.

16   **end**

17   **end**

18   **end**

Fig. 3. Performance comparison in terms of the achievable BER and the computational complexity.

and check nodes was carried out pseudo-randomly, and no girth optimization techniques were employed.

The channel estimates were calculated from the known pilots, up-sampled and interpolated using a low-pass interpolator. Armed with this MIMO channel estimate, the received signal was then detected using a soft-input soft-output maximum a-posteriori probability (MAP) detector. The detected signal was de-interleaved using Π_p and then passed to the decoder, which employed the classic belief propagation algorithm in order to estimate the original information bit sequence. The decoder was limited to 100 iterations.

Figure 3 compares the performance of the two schemes in terms of the achievable BER and the computational complexity CC, where the latter was evaluated in terms of the number of message-passing updates per decoded bit, given by $CC = I_{\mathrm{avg}}|E|/K$, where I_{avg} represents the average number of iterations required for finding a legitimate codeword and $|E|$ represents the number of edges in the Tanner graph. It can be observed from Figure 3 that the proposed PSAC scheme exhibits small but noticeable BER performance gains.[3] Besides this advantage, the computational complexity of PSAC is significantly lower, requiring 45% fewer message-passing updates per decoded bit than the benchmarker scheme at $E_b/N_0 = 2$ dB. We note that the lower CC offered by PSA codes can be attributed to the fact that (a) they require a lower number of I_{avg} and (b) they demand a lower number of edges $|E|$ in order to realize the code.

VI. CONCLUSION

It can therefore be concluded that it is more beneficial to appropriately intersperse pilots at the channel code stage as in the

[3]In this context, it is worth mentioning that the benchmarker system has also been optimized by the conventional EXIT chart matching technique.
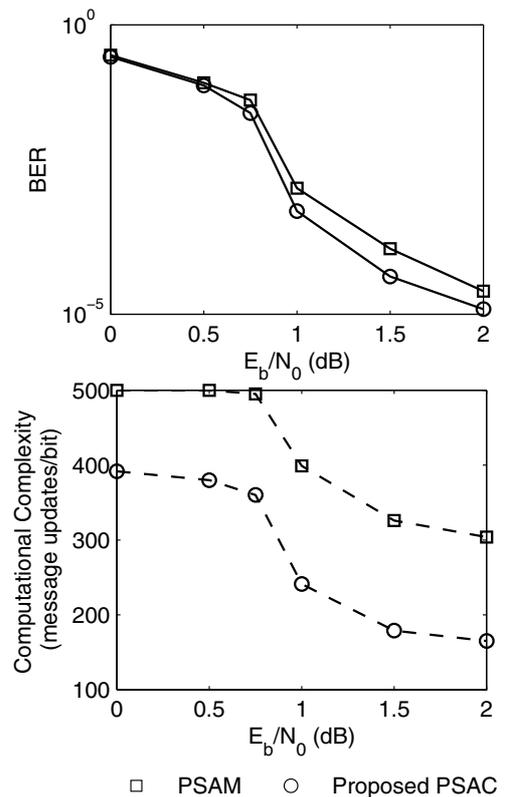
proposed PSAC rather than at the modulation stage as in classic PSAM, since the pilot bits are not only useful for channel estimation but also for enhancing the decoder's performance. From another point-of-view, we can also regard pilot symbol assisted codes, as a family of codes, which are specifically designed for systems that require pilot-aided channel estimation.

REFERENCES

[1] M. L. Moher and J. H. Lodge, "TCMP-a modulation and coding strategy for Rician fading channels," *IEEE Journal on Selected Areas on Communications*, pp. 1347–1355, Dec. 1989.

[2] J. P. McGeehan and A. J. Bateman, "Phase locked transparent tone-in-band (TTIB): A new spectrum configuration particularly suited to the transmission of data over SSB mobile radio networks," *IEEE Transactions on Communications*, pp. 81–87, Jan. 1984.

[3] J. K. Cavers, "An analysis of pilot symbol assisted modulation for Rayleigh faded channels," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 686–693, Nov. 1991.

[4] M.-K. Oh, H. M. Kwon, D.-J. Park, and Y. H. Lee, "Iterative channel estimation and LDPC decoding with encoded pilots," *IEEE Transactions on Vehicular Technology*, vol. 57, pp. 273–285, Jan. 2008.

[5] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proceedings* 2nd *International Symposium on Turbo Codes and Related Topics*, (Brest, France), pp. 1–8, Sept. 2000.

[6] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 1451–1458, Oct. 1998.

[7] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, pp. 1727–1737, Oct. 2001.

[8] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2764–2772, Nov. 2003.

[9] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Transactions on Communications*, vol. 52, pp. 670–678, Apr. 2004.