Research Article

Error Recovery Properties and Soft Decoding of Quasi-Arithmetic Codes

Simon Malinowski,[1] Hervé Jégou,[1] and Christine Guillemot[2]

[1] *IRISA/University of Rennes, Campus de Beaulieu, 35042 Rennes Cedex, France*
[2] *IRISA/INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France*

Correspondence should be addressed to Simon Malinowski, simon.malinowski@irisa.fr

This paper first introduces a new set of aggregated state models for soft-input decoding of quasi arithmetic (QA) codes with a termination constraint. The decoding complexity with these models is linear with the sequence length. The aggregation parameter controls the tradeoff between decoding performance and complexity. It is shown that close-to-optimal decoding performance can be obtained with low values of the aggregation parameter, that is, with a complexity which is significantly reduced with respect to optimal QA bit/symbol models. The choice of the aggregation parameter depends on the synchronization recovery properties of the QA codes. This paper thus describes a method to estimate the probability mass function (PMF) of the gain/loss of symbols following a single bit error (i.e., of the difference between the number of encoded and decoded symbols). The entropy of the gain/loss turns out to be the average amount of information conveyed by a length constraint on both the optimal and aggregated state models. This quantity allows us to choose the value of the aggregation parameter that will lead to close-to-optimal decoding performance. It is shown that the optimum position for the length constraint is not the last time instant of the decoding process. This observation leads to the introduction of a new technique for robust decoding of QA codes with redundancy which turns out to outperform techniques based on the concept of forbidden symbol.

1. INTRODUCTION

Variable length coding offers great performance in terms of compression, but it is however very sensitive to channel noise. Indeed, a single error in the bit-stream may result in the desynchronization of the decoder leading to dramatic symbol-error rates (SERs). Synchronization recovery properties of variable length codes (VLCs) have been first studied in [1]. A model to derive the error span E_s following a single bit error (i.e., the expected number of symbols on which the error propagates inside the decoded symbol stream) is proposed. It can be shown that VLCs strictly satisfying the Kraft inequality statistically resynchronize with a probability of 1. However, they do not always resynchronize in the strict sense (i.e., the symbol length of the decoded bit-stream may differ from the one of the encoded bit-stream). In [2], the authors have extended the model of [1] to calculate the probability mass function (PMF) of the so-called *gain/loss*. The *gain/loss* represents the difference between the number of encoded and

decoded symbols. It is shown in [3] that E_s reflects the performance of a VLC with hard decoding. In [4], the measure of the *gain/loss* is shown to reflect the performance of VLCs when soft-input decoding with a termination constraint on the number of encoded symbols is applied at the decoder side. In the following, the term termination constraint will denote a constraint that enforces the symbol length of the decoded sequence to be equal to a given number of symbols (known by the decoder). The term length constraint will refer to the same kind of constraint but at a given bit clock time instant of the decoding process (not necessarily at the end). The synchronization recovery properties of VLCs have been used in [4] to validate that optimal decoding performance can be reached with the aggregated model (of low complexity) proposed in [5] for soft-input decoding of VLCs.

Recently, arithmetic coding (AC) [6] has drawn the attention of many researchers because of its use in practical applications such as JPEG2000 or MPEG-4. The major interest of AC is that the representation of the information may

be arbitrarily close to the entropy. However, it is also very sensitive to channel noise, which has led many authors to design error-correcting schemes involving AC. For instance, parity-check bits are embedded in the encoding procedure in [7]. Alternative approaches consist in reserving space in the encoding interval for a forbidden symbol in order to detect and correct errors [8, 9]. To protect the bit-stream against the desynchronization phenomenon, the authors of [10] introduced soft information about the symbol clock inside the bit-stream.

Quasi arithmetic (QA) coding is a reduced precision version of AC. A QA coder operates on an integer interval $[0, N)$ rather than on the real interval $[0, 1)$ to avoid numerical precision problems. The parameter N controls the tradeoff between the number of states and the approximation of the source distribution. In [11], a method is proposed to represent a QA code with state machines. The number of states can be infinite. To ensure a finite number of states, Langdon proposed a method called bit-stuffing. An alternative method to keep the number of states finite is proposed in [12]. These methods induce a slight loss in terms of compression. In this paper, we will represent QA codes with two finite state machines (FSMs), one for the encoder and one for the decoder. If N is sufficiently small, the state transitions and the outputs can be precomputed and arithmetic operations can thus be replaced by table lookups. Recently, the authors of [12] have provided asymptotic error-correcting performance bounds for joint source channel-channel schemes based on arithmetic coding. These bounds can be derived for arithmetic and QA coding with or without a forbidden symbol.

This paper addresses the problem of optimal and reduced complexity soft-input decoding of QA codes with a termination constraint. An optimal state model for soft-decoding of QA codes, exploiting a termination constraint, has been presented in [10]. However, the decoding complexity with this model is very high and not tractable in practical applications. The set of aggregated state models introduced in [5] for VLC is first extended to QA codes. These models are parameterized by an integer T that controls the tradeoff between estimation accuracy and decoding complexity. Coupled with a BCJR or a Viterbi algorithm, decoding performance close to the optimal model performance [10] can be obtained with low values of the aggregation parameter T, that is, with a significantly reduced complexity. The choice of the parameter T which will lead to close to optimum decoding actually depends on the synchronization recovery properties of the QA code considered.

The paper then describes a method to analyze the synchronization recovery properties of QA codes. This method is based, as in [1] for VLC, on error state diagrams (ESDs). Using transfer functions on these ESDs, the approach computes the expected error span E_s and the *gain/loss* ΔS following a single bit error for QA coding. The computation of these quantities is then extended to the case where the bit-stream is transmitted over a BSC. The entropy of the *gain/loss* turns out to be a powerful tool to analyze the performance of soft-input decoding of QA codes with termination constraints. It allows us to quantify the amount of information conveyed by a length constraint on both the optimal and the aggregated models. For a given QA code and a given channel signal-to-noise ratio, the entropy of the *gain/loss* allows us to determine the value of the aggregation parameter for which close-to-optimum decoding performance will be obtained.

The positioning of the sequence length constraint leading to the best decoding performance is then studied. It is shown that the optimum position of the length constraint is not the last time instant of the decoding process. The best position actually depends on the channel signal-to-noise ratio and on the error recovery properties of the code. This observation led to the introduction of a new approach to add redundancy to QA codes in order to improve their decoding performance. The redundancy is added under the form of length constraints exploited at different time instants of the decoding process on the aggregate state model. In comparison with other methods based on side information such as markers which help the resynchronization of the decoding process, the strategy proposed here does not lead to any modification of the compressed bit-stream. The side information can be transmitted separately. The approach turns out to outperform widely used techniques based on the introduction of a forbidden symbol.

The rest of the paper is organized as follows. QA codes and their representation as FSMs are recalled in Section 2. The optimal model together with the proposed aggregated models for soft-input decoding of QA codes with termination constraints are described in Section 3. Section 4 presents the method to compute the PMF of the *gain/loss* following a single bit error, and when the source sequence is transmitted over a BSC channel. In Section 5, the decoding performance of QA codes is analyzed in terms of the aggregation parameter, showing the link between the entropy of the *gain/loss*, the aggregation parameter, and the decoding performance. Finally, Section 6 presents a new method to add redundancy to QA codes in order to further help the decoder resynchronization on aggregated state models.

2. QUASI ARITHMETIC CODES

In arithmetic coding, the real interval $[0, 1)$ is subdivided in subintervals. The length of the subintervals is proportional to the probabilities of the symbol sequence it represents. At the end of the encoding, the arithmetic encoder outputs enough bits to distinguish the final interval from all other possible intervals. One of the main drawbacks of AC is the coding delay and the numerical precision, as the probabilities and hence the length of the final interval quickly tends to be small. In practice, QA coding [13] is more often used. In QA coding, the initial interval is set to the integer interval $[0, N)$. Bits are output as soon as they are known to reduce the encoding delay and the interval is rescaled to avoid numerical precision problems. Here, we will consider QA codes as FSMs as in [14]. A QA code can be defined with two different FSMs: an encoding one and a decoding one. The number of states of these FSMs will be denoted N_e and N_d, respectively. The sets of states for the encoding and decoding FSM will be denoted $\mathcal{l}_e = \{\alpha_0, \alpha_1, \ldots, \alpha_{N_e-1}\}$ and $\mathcal{l}_d = \{\beta_0, \beta_1, \ldots, \beta_{N_d-1}\}$, respectively. The encoding FSM generates a variable number of bits which depends on the current state and on the source
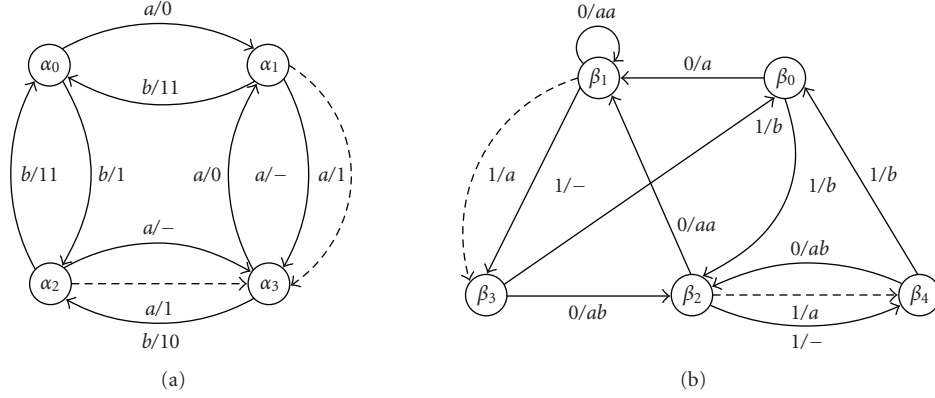
FIGURE 1: Encoding (a) and decoding (b) FSM associated with code $\mathcal{C}_7$.

symbol that has been sent. The decoding FSM retrieves the symbol stream from the received bit-stream. The number of decoded symbols for each received bit depends as well on the state of the decoding FSM and on the bit received. Let $\mathbf{S} = S_1, \ldots, S_{L(\mathbf{S})}$ be a source sequence of length $L(\mathbf{S})$ taking its value into the binary alphabet $\mathcal{A} = \{a, b\}$. The probability of the more probable symbol (MPS), $\mathbb{P}(a)$ will be denoted p in the following. This source sequence is encoded with a QA code, producing a bit-stream $\mathbf{X} = X_1, \ldots, X_{L(\mathbf{X})}$ of length $L(\mathbf{X})$. This bit-stream is sent over a noisy channel. A hard decision is taken from the received soft channel outputs. The corresponding bit-stream will be noted $\mathbf{Y} = Y_1, \ldots, Y_{L(\mathbf{X})}$.

Example 1. Let us consider the QA code for $N = 8$ proposed in [10]. This code, denoted $\mathcal{C}_7$ in the following, is adapted to a binary source with $\mathbb{P}(a) = 0.7$. The corresponding encoding and decoding FSMs are depicted on Figure 1. On the branches, are denoted the input/output bit(s)/symbol(s) associated with the FSMs. For this code, we have $N_e = 4$ and $N_d = 5$. The initial states of the encoding and decoding FSM are α_0 and β_0, respectively. To ensure that the encoded bit-stream is uniquely decodable, mute transitions (i.e., transitions that do not trigger any output bit) may not be used to encode the last symbol. For example, encoding symbol stream aa with the automata of Figure 1 leads to the bit-stream 0, which would also have been obtained to encode a. To avoid that, the mute transitions are changed so that they output bits when used to encode the last symbol. These transitions are depicted by dotted arrows in Figure 1. For example, the symbol stream $aabaa$ is encoded with the bit-stream 01001.

The representation of a QA code by two FSMs allows the computation of the expected description length (EDL) of the code (as in [15, page 32]). The expected length $\bar{l}_{\alpha_i}$ of the output from a state α_i ($\alpha_i \in \mathcal{I}_e$) of the encoding FSM is given by

$$\bar{l}_{\alpha_i} = p \times o_{\alpha_i,a} + (1-p) \times o_{\alpha_i,b}, \qquad (1)$$

where $o_{\alpha_i,a}$ and $o_{\alpha_i,b}$ represent the number of bits produced by the transitions from state α_i triggered by the symbols a and b, respectively. In addition, if the transition matrix correspond-

ing to the encoding FSM is denoted P_e, the eigenvector of P_e associated with the eigenvalue 1 gives the long-term state occupation probabilities $\mathbb{P}(\alpha_i)$ for each state of the FSM. Note that this holds if the considered automaton is ergodic. It is the case for most QA automata. Hence, the EDL of the code is given by

$$\bar{l} = \sum_{\alpha_i \in \mathcal{I}_e} \bar{l}_{\alpha_i} \mathbb{P}(\alpha_i). \qquad (2)$$

According to this method, the EDL of code $\mathcal{C}_7$ is equal to 0.905.

Let us now estimate the PMF of the bit length of the message. This PMF is approximated by a Gaussian PMF of mean $\bar{l} \times L(\mathbf{S})$. The variance of this distribution is estimated hereafter. Denoting

$$v_{\alpha_i}^2 \triangleq p \times (o_{\alpha_i,a} - \bar{l})^2 + (1-p) \times (o_{\alpha_i,b} - \bar{l})^2, \qquad (3)$$

the expected variance of the bit-length output for one input symbol is given by

$$v^2 = \sum_{\alpha_i \in \mathcal{I}_e} v_{\alpha_i}^2 \mathbb{P}(\alpha_i). \qquad (4)$$

The PMF of $L(\mathbf{X})$ is then approximated by a Gaussian PMF of mean $\bar{l} \times L(\mathbf{S})$ and variance $v \times L(\mathbf{S})$. This estimation will be used in Section 4.3.

3. REDUCED COMPLEXITY SOFT-INPUT DECODING OF QA CODES WITH TERMINATION CONSTRAINT

An optimal state model for trellis-based decoding of QA codes was proposed in [10]. This model integrates both the bit and symbol clocks to perform the estimation. It is called optimal since, coupled with a maximum a posteriori (MAP) estimation algorithm, it provides optimal decoding performance in terms of symbol-error rate (SER) or frame-error rate (FER). In this section, we propose a set of state models defined by an aggregation parameter T. The complexity of these models is linear with T. We will see that the optimal decoding performance (i.e., the one obtained with the model in [10]) is obtained for a significantly reduced complexity.

After having recalled the optimal state model, we will define our proposed set of state models and provide some simulation results.

3.1. Optimal state model

Let us assume that the number of transmitted symbols is perfectly known on the decoder side. To use this information as a termination constraint in the decoding process, the state model must keep track of the symbol clock (i.e., the number of decoded symbols). The optimal state model for trellis-based decoding of QA codes with termination constraint is defined in [10]. The states of this model are the tuples of random variable (N_k, T_k). The random variables N_k and T_k represents, respectively, the state of the decoder automaton and the possible symbol clock values at the bit clock instant k. The termination constraint on this model amounts to forcing the number of symbols of the estimated sequence to be equal to $L(\mathbf{S})$. The number of states of this model is a quadratic function of the sequence length (equivalently the bit-stream length). The resulting computational cost is thus not tractable for typical values of the sequence length $L(\mathbf{S})$. To fight against this complexity hurdle, most authors apply suboptimal estimation methods on this model as sequential decoding ([16], e.g.).

3.2. Aggregated model

Let us define the aggregated state model by the tuples (N_k, M_k), where M_k is the random variable equal to T_k modulo T, that is, the remainder of the Euclidean division of T_k by T. The transitions that correspond to the decoding of σ symbol(s) modify M_k as $M_k = (M_{k-1} + \sigma) \bmod T$. Hence, the transition probabilities on this model are given by

$$\forall (\beta_i, \beta_j) \in \mathcal{L}_d \times \mathcal{L}_d,$$

$$\mathbb{P}(N_k = \beta_i, M_k = m_k \mid N_{k-1} = \beta_j, M_{k-1} = m_{k-1})$$

$$= \begin{cases} \mathbb{P}(N_k = \beta_i \mid N_{k-1} = \beta_j) & \text{if } m_k = m_{k-1} + \sigma \bmod T, \\ 0 & \text{otherwise,} \end{cases}$$

$$(5)$$

where the probabilities $\mathbb{P}(N_k = n_k \mid N_{k-1} = n_{k-1})$ are deduced from the source statistics. Note that $T = L(\mathbf{S})$ amounts to considering the optimal model described above. The parameter T controls the tradeoff between estimation accuracy and decoding complexity. The aggregated model of parameter T keeps track of the symbol clock values modulo the parameter T, not of the entire symbol clock values as in the optimal model. The termination constraint on this model amounts to forcing the number of symbols modulo T of the estimated sequence to be equal to $m_{L(\mathbf{X})} = L(\mathbf{S}) \bmod T$. If $m_{L(\mathbf{X})}$ is not given by the syntax elements of the source coding system, this value has to be transmitted. The transmission cost of $m_{L(\mathbf{X})}$ is equal to $\lceil \log_2 T \rceil$ bits.

TABLE 1: SER for soft-input decoding (Viterbi) with different values of the aggregation parameter T.

E_b/N_0	3	4	5	6	7
	Code $\mathcal{C}_7$				
$T = 1$	0.177725	0.119088	0.066019	0.0293660	0.0099844
$T = 2$	0.177660	0.119029	0.065977	0.0293483	0.0099790
$T = 3$	0.120391	0.058873	0.020400	0.0049425	0.0008687
$T = 4$	0.145964	0.080466	0.031784	0.0085192	0.0014814
$T = 5$	0.097285	0.047133	0.016785	0.0043368	0.0008219
$T = 6$	0.117477	0.057001	0.019686	0.0047921	0.0008442
$T = 7$	0.092465	0.045490	0.016468	0.0043100	*0.0008151*
$T = 8$	0.101290	0.048256	0.016978	0.0043348	*0.0008151*
$T = 9$	0.090876	0.045029	0.016414	*0.0042998*	—
$T = 13$	0.090095	0.044894	*0.016394*	—	—
$T = 15$	0.089984	*0.044882*	—	—	—
$T = 21$	*0.089963*	—	—	—	—
$T = 100$	0.089963	0.044882	0.016394	0.0042998	0.0008151
	Code $\mathcal{C}_9$				
$T = 1$	0.089340	0.056646	0.029854	0.0127630	0.0042457
$T = 2$	0.078041	0.045343	0.021652	0.0083708	0.0026533
$T = 3$	0.067114	0.032231	0.011263	0.0027026	0.0004697
$T = 4$	0.064223	0.032616	0.013335	0.0045597	0.0013501
$T = 5$	0.055957	0.023517	0.006744	0.0013292	0.0001833
$T = 6$	0.050188	0.020412	0.005796	0.0011347	0.0001552
$T = 7$	0.049695	0.020945	0.006313	0.0014643	0.0002736
$T = 8$	0.054915	0.027628	0.011621	0.0041529	0.0012898
$T = 9$	0.049107	0.020842	0.006527	0.0014290	0.0002403
$T = 15$	0.032811	0.011623	0.002845	0.0005035	*0.00005259*
$T = 20$	0.025392	0.009089	0.002322	*0.0004284*	—
$T = 35$	0.024457	0.008826	*0.0022843*	—	—
$T = 41$	0.023951	*0.008752*	—	—	—
$T = 46$	*0.023576*	—	—	—	—
$T = 100$	0.023576	0.008752	0.0022843	0.0004284	0.00005259

3.3. Decoding performance of the aggregated model

In this section, we use the aggregated model presented above for soft-input decoding of source sequences encoded with two QA codes $\mathcal{C}_7$ and $\mathcal{C}_9$. The code $\mathcal{C}_7$ is defined by the encoding and decoding automata of Figure 1. The code $\mathcal{C}_9$ is a 3-bit precision QA code obtained for a binary source with $P(a) = 0.9$. The EDL of these codes are, respectively, equal to 0.905 and 0.491. The decoding performance in terms of SER (Hamming distance between the sent and estimated symbol sequences) of these two codes with respect to the aggregation parameter T is presented in Table 1. This decoding performance is obtained by running a Viterbi algorithm on the aggregated state model. These results have been obtained for 10^5 sequences of $L(\mathbf{S}) = 100$ binary symbols sent through an additive white Gaussian noise (AWGN) channel characterized by its signal-to-noise ratio E_b/N_0. In this table, the best decoding performance for each code and at every E_b/N_0 value is written in *italics*. These values also correspond to the performance obtained on the optimal model (i.e., for $T = 100$).
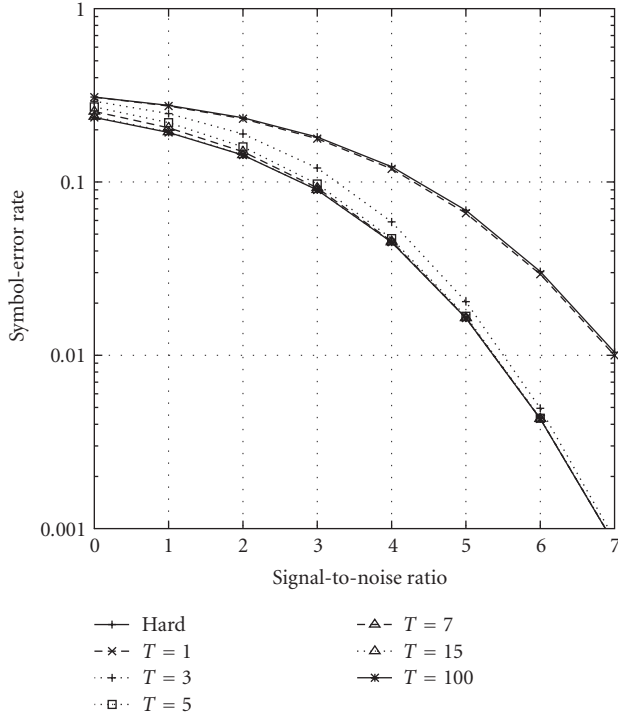
FIGURE 2: SER results versus signal-to-noise ratio for the code $\mathcal{C}_7$.

TABLE 2: Optimal parameters T and M for different values of the signal-to-noise ratio for code $\mathcal{C}_7$

E_b/N_0 (dB)	3	4	5	6
T_{opt}	21	15	13	9
M_{opt}	235	184	95	29

with the highest metric is selected. The values of M that yield the same decoding performance than the optimal model are depicted in Table 2. We observe that the optimal values of M are much higher than the optimal values of T for the same level of performance. Note that the number of branch metric computations for a Viterbi decoding on a trellis of parameter T is approximately equal to $2L(\mathbf{X})N_dT$, while it is approximately equal to $2L(\mathbf{X})M$ for the M-algorithm. Note also the Viterbi algorithm on an aggregated state model needs to store $3N_dT$ float values at each bit clock instant (state metric, last bit sent and previous state, for each state), while the M-algorithm needs to store $6M$ float values before sorting the $2M$ extended sequences (sequence, associated state of the decoding automaton and associated symbol length). We will show in Section 5 that the variations of the decoding performance with the aggregation parameter T actually depend the resynchronization properties of the codes. In order to establish this link, we first introduce in the next section mathematical tools used to analyze these properties.

4. SYNCHRONIZATION PROPERTIES OF QA CODES

The synchronization recovery properties of VLC has been studied in [2], where the authors propose a method to compute the PMF of the so-called *gain/loss* following a single bit error, denoted ΔS in the following. In [4], the PMF of ΔS is used to analyze the decoding performance behavior of VLC on the aggregated model. In this section, a method is proposed to compute the PMF of the *gain/loss* ΔS for QA codes. This method is based on calculating gain expressions on error state diagrams, as in [1, 2]. The error state diagrams proposed in [2] to compute the PMF of ΔS for VLC cannot be directly used for QA codes. We hence extend in this section the method of [2] to QA codes by defining a new type of error state diagram and computing the PMF of ΔS following a single bit error using these state diagrams. The computation of this PMF is then extended to the case where a source sequence is sent over a binary symmetrical channel (BSC) of given cross-over probability.

4.1. Error state diagrams

Let us first consider that the bit-streams $\mathbf{X}$ (sent) and $\mathbf{Y}$ (received) differ only by a single bit inversion at position k_p. Let the tuples $(N_k^{\mathbf{X}}, N_k^{\mathbf{Y}})$ denote the pair of random variables representing the state of the decoding FSM after having decoded k bits of $\mathbf{X}$ and $\mathbf{Y}$, respectively. The realizations of these tuples will be denoted $(n_k^{\mathbf{X}}, n_k^{\mathbf{Y}})$. We have

$$\forall k < k_p, \quad n_k^{\mathbf{X}} = n_k^{\mathbf{Y}}. \qquad (6)$$

The values of T that yield the best decoding performance depend on the code, and on the signal-to-noise ratio. For the set of parameters of Table 1, the optimal value of T is always lower than $L(\mathbf{S})$, which ensures that the complexity is reduced compared to the optimal model of [10], for the same level of performance. Figure 2 depicts the SER versus the signal-to-noise ratio at different values of T for code $\mathcal{C}_7$. The even values of T are not depicted as they are not appropriate for this code according to Table 1. We could expect the decoding performance of a code to increase with T. However, the results given in Table 1 show that the SER behavior is not monotonic with T. Indeed, for the code $\mathcal{C}_7$, the values $T = 2t - 1, t > 1$, yield lower SER than $T = 2t$, and this at every signal-to-noise ratio. The behavior of the decoding performance of code $\mathcal{C}_9$ with respect to T is also not monotonic: the SER for $T = 8$ is always higher than the one obtained for $T = 6$ and $T = 7$. In addition, for $E_b/N_0 \geq 4$ dB, the SER obtained for $T = 4$ is higher than the one for $T = 3$. The sequential decoding algorithm called M-algorithm [17] has been applied to code $\mathcal{C}_7$ in order to find the minimum values of M that yield the same decoding performance as the optimal decoding of Table 1 at different values of the signal-to-noise ratio. The basic principle of this algorithm is to store for each bit clock instant k the M most probable sequences of k bits. At the bit clock instant $k + 1$, these M sequences are extended into $2M$ sequences of $k + 1$ bits. The M most probable sequences amongst these $2M$ sequences are stored. When the last bit clock instant is reached, the most probable valid sequence (i.e., with the right number of symbols) is selected. If, at the last bit clock instant, none of the M sequences satisfy the constraint on the number of symbols, the sequence

Table 3: Computation of the transitions and transition probabilities for the error state diagram of initial state (β_0, β_0).

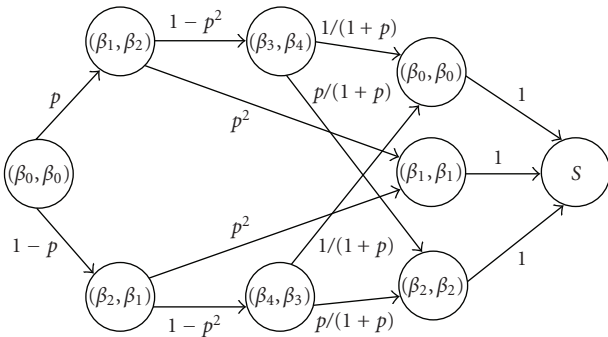| State	Bit sent	Bit received	Next state	Probability
(β_0, β_0)	0	1	(β_1, β_2)	p
(β_0, β_0)	1	0	(β_2, β_1)	$1 - p$
(β_1, β_2)	0	0	(β_1, β_1)	p^2
(β_1, β_2)	1	1	(β_3, β_4)	$1 - p^2$
(β_2, β_1)	0	0	(β_1, β_1)	p^2
(β_2, β_1)	1	1	(β_4, β_3)	$1 - p^2$
(β_3, β_4)	0	0	(β_2, β_2)	$p/(1 + p)$
(β_3, β_4)	1	1	(β_0, β_0)	$1/(1 + p)$
(β_4, β_3)	0	0	(β_2, β_2)	$p/(1 + p)$
(β_4, β_3)	1	1	(β_0, β_0)	$1/(1 + p)$

Figure 3: Error state diagram of initial state (β_0, β_0) for the code $\mathcal{C}_7$.

The bit inversion at position k_p in **Y** may lead to $n_{k_p}^{\mathbf{X}} \neq n_{k_p}^{\mathbf{Y}}$, which means that the decoder is desynchronized. The decoder will resynchronize at the bit index k_r such that

$$k_r = \min_{k_p \leq k \leq L(\mathbf{X})} k \mid n_k^{\mathbf{X}} = n_k^{\mathbf{Y}}. \tag{7}$$

An error state diagram depicts all possible tuples $(n_k^{\mathbf{X}}, n_k^{\mathbf{Y}})$ from the bit error (bit instant $k_p - 1$) to the resynchronization. Depending on the state of the decoder when the error occurs (i.e., $n_{k_p-1}^X$), the synchronization recovery behavior of the code will be different. Hence, N_d diagrams are drawn, one for each states of $\mathcal{I}_d$. The final states of these diagrams are the tuples (β_i, β_i), $\beta_i \in \mathcal{I}_d$. When one of these states is reached, the decoder is resynchronized. Table 3 depicts the states $(n_k^{\mathbf{X}}, n_k^{\mathbf{Y}})$ reached when the error occurs in the state β_0 of the decoding FSM. The corresponding error state diagram is given in Figure 3. The transition probabilities are denoted next to the branches.

4.2. Computation of the PMF of ΔS

To compute the PMF of ΔS, the branches of the error state diagrams are labeled with a variable z^l, where l represents the difference between the number of encoded and decoded symbols along the considered transition. Note that l can be negative if the decoded sequence has more symbols than the encoded one.

Example 2. Let us calculate the value of l associated with the transition between the states (β_0, β_0) and (β_1, β_2). The transition from β_0 to β_1 triggers the symbol a when the sequence **X** is decoded, whereas the transition from β_0 to β_2 triggers the symbol b when **Y** is decoded. Hence, $l = 1 - 1 = 0$. The label of this branch on the diagram is hence set to $p \times z^0 = p$.

Hence, the gain of the diagram from the initial state (β_j, β_j), $\beta_j \in \mathcal{I}_d$ to the synchronization state S is a formal Laurent series of the variable z:

$$G_{\beta_j}(z) \triangleq \sum_{k \in \mathbb{Z}} g_{\beta_j, k} z^k. \tag{8}$$

This gain series can be calculated with Mason's formula [18], or by inverting the matrix $(I - \overline{H})$, where I represents the identity matrix and $\overline{H}$ is the transition matrix associated with the error state diagram. This matricial expression is explained in more detail in [3]. The coefficient $g_{\beta_j, k}$ is equal to $\mathbb{P}(\Delta S = k \mid N_{i-1}^{\mathbf{X}} = \beta_j)$.

Let us define the overall formal Laurent series $G(z)$ as

$$G(z) \triangleq \sum_{k \in \mathbb{Z}} g_k z^k = \sum_{j \in \mathcal{I}_d} G_{\beta_j}(z) \mathbb{P}(N_{i-1}^{\mathbf{X}} = \beta_j), \tag{9}$$

where the long-term state occupation probabilities $\mathbb{P}(N_{i-1}^{\mathbf{X}} = \beta_j)$ are calculated as explained in Section 2. Hence, the coefficient g_k of z^k in the formal Laurent series $G(z)$ is equal to $\mathbb{P}(\Delta S = k)$.

Hence, we have shown how to compute the PMF of ΔS by adapting the branch labeling on a set of error state diagrams. Note that these quantities are valid for a single bit error.

The error span E_s can also be derived from the error state diagrams described above. The labels of the branch have to be replaced by $z^{l'}$, where l' represents the number of decoded symbol along the transition. The method described in [1] to compute E_s for VLC can be applied on the error state diagrams described above.

4.3. Extension to the BSC

We propose here to estimate the PMF of ΔS for a symbol sequence of length $L(\mathbf{S})$ that has been sent through a BSC of crossover probability π (equals to the bit-error rate). This analysis also applies to binary phase shift keying over AWGN channels with hard decisions at the channel output, which results in an equivalent BSC of cross-over probability $\pi = 1/2 \operatorname{erfc}(\sqrt{E_b/N_0})$.

In Section 2, we have seen how to estimate the PMF of the length of the bit-stream $L(\mathbf{X})$ corresponding to the encoding of a message of $L(\mathbf{S})$ symbols. Let E denote the random variable corresponding to the number of errors in the received bit-stream **Y**. Its probability can be expressed as

$$\mathbb{P}(E = e) = \sum_{i \in \mathbb{N}} \mathbb{P}(E = e \mid L(\mathbf{X}) = i) \mathbb{P}(L(\mathbf{X}) = i). \tag{10}$$

Note that $\mathbb{P}(E = e \mid L(\mathbf{X}) = i)$ only depends on π and is equal to

$$\mathbb{P}(E = e \mid L(\mathbf{X}) = i) = \begin{cases} \binom{i}{e}\pi^e(1-\pi)^{i-e} & \text{if } e \leq i, \\ 0 & \text{if } e > i. \end{cases} \tag{11}$$

Let us now assume that the decoder has already recovered from previous errors when another error occurs. This assumption requires that the probability that an error occurs when the decoder has not resynchronized yet is low. Lower is E_s and lower is π, the more accurate is this approximation. Under this assumption, the quantity ΔS is independently impacted by multiple errors. Hence, the coefficients $a_{i,e}$ of the formal Laurent series $G_e(z)$ defined by

$$G_e(z) = \sum_{i \in \mathbb{Z}} a_{i,e} z^i \triangleq (G(z))^e \tag{12}$$

satisfy

$$a_{i,e} = \mathbb{P}(\Delta S = i \mid E = e). \tag{13}$$

Note that the Laurent series $G_{e\mid e=1}$ corresponds to the gain series of (9) (single bit error case).

With (10), the resulting gain series for this cross-over probability is expressed as

$$\widetilde{G}(y) = \sum_{e \in \mathbb{N}} G_e(y)\mathbb{P}(E = e), \tag{14}$$

where only the quantity $\mathbb{P}(E = e)$ depends on π. The coefficients $\widetilde{g}_i$ of $\widetilde{G}$ satisfy

$$\begin{aligned} \widetilde{g}_i &= \sum_{e \in \mathbb{N}} a_{i,e}\mathbb{P}(E = e) \\ &= \sum_{e \in \mathbb{N}} \mathbb{P}(\Delta S = i \mid E = e)\mathbb{P}(E = e) \\ &= \mathbb{P}(\Delta S = i). \end{aligned} \tag{15}$$

This method has been used to compute the PMF of ΔS for the code $\mathcal{C}_7$ on an AWGN channel. This PMF is depicted in Figure 4, together with the simulated PMF for $E_b/N_0 = 4$ dB. These values have been computed for a 100-symbol message. Note that for this code, the theoretical probabilities that ΔS is odd are equal to zero. In the simulated values of ΔS, these probabilities are not all equal to zero. ΔS can indeed be odd if a bit error is located at the end of the bit-stream, hence leading the decoder not to resynchronize before the end of the message.

5. LINK BETWEEN ΔS AND THE DECODING PERFORMANCE ON THE AGGREGATED MODEL

A trellis-based decoder with a termination constraint exploits two kinds of information to help the estimation: the residual redundancy of the code and the information conveyed by the termination constraint. For a given code, the
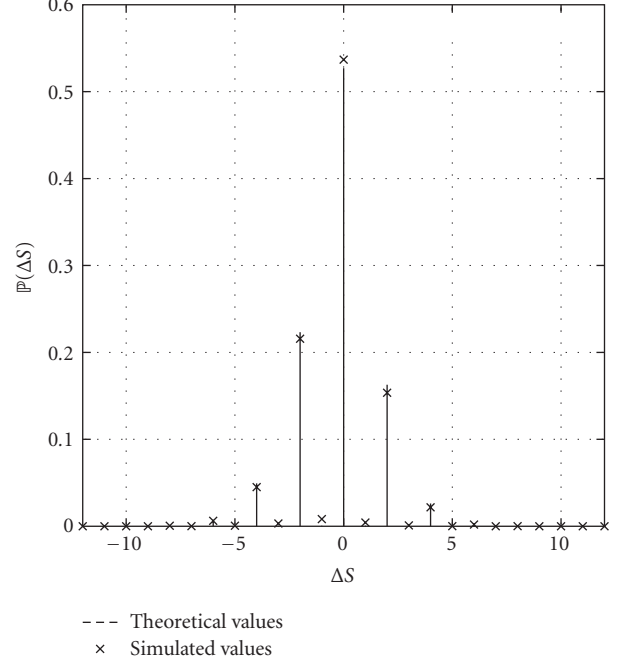

FIGURE 4: Theoretical and simulated PMFs of ΔS: code $\mathcal{C}_7$, $E_b/N_0 = 4$ dB.

residual redundancy per source symbol is equal to the difference between the EDL of the code and the entropy of the source. As QA codes are entropy codes, their residual redundancy is very low. Hence, on both optimal and aggregated state models, the main quantity of information available at the decoder is the one conveyed by the termination constraints. On the optimal model, this information is quantified by the entropy of the random variable ΔS, denoted $H(\Delta S)$. Indeed, the termination constraint on the optimal model discards the sequences that do not have the right number of symbols, that is, that do not satisfy $\Delta S = 0$. On the aggregated model, the termination constraint discards the sequences that do not satisfy $\Delta S \bmod T = 0$. The information conveyed by this constraint is hence quantified by the entropy of the random variable $\Delta S \bmod T$, denoted $H(\Delta S \bmod T)$. To compare the decoding performance of a QA code on the aggregated model with the performance on the optimal model, we hence need to compare $H(\Delta S)$ with $H(\Delta S \bmod T)$. If these two quantities are close, we can expect the decoding performance of a QA code on both models to be close.

The quantities $H(\Delta S)$ and $H(\Delta S \bmod T)$ have been computed for different values of T and signal-to-noise ratio. These quantities have been obtained according to the method described in the previous section for source sequences of $L(\mathbf{S}) = 100$ binary symbols (as for the decoding performance presented in Section 3.3). Figure 5 depicts these two quantities for the code $\mathcal{C}_7$, $E_b/N_0 = 6$ dB and for different values of T. We can observe that for $T = 2t$, $t > 1$, $H(\Delta S \bmod T)$ is lower than $H(\Delta S \bmod (T - 1))$. This unexpected behavior of $H(\Delta S \bmod T)$ stems from the synchronization properties of $\mathcal{C}_7$ explained in the previous section. Indeed, we have seen
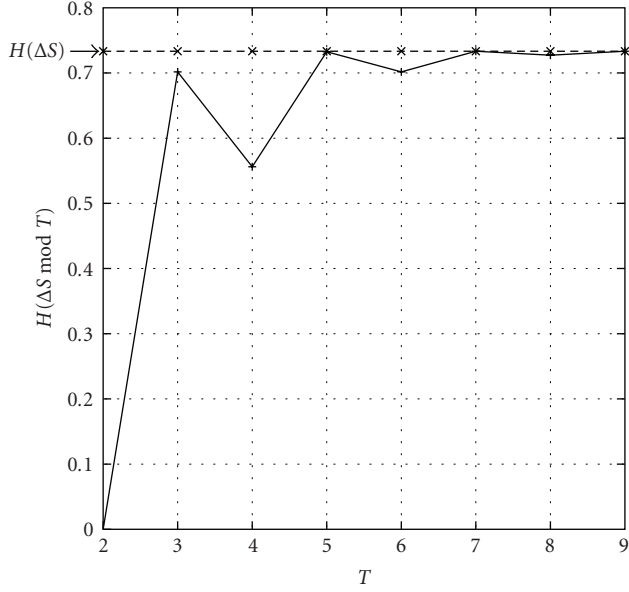
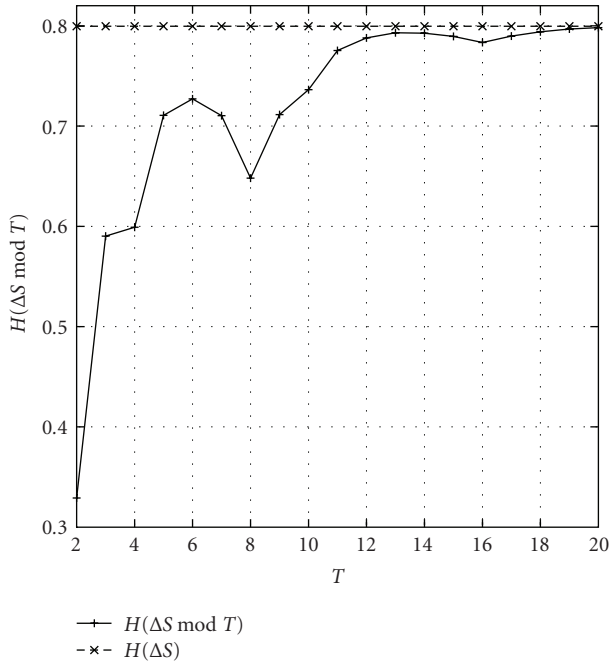FIGURE 5: Entropy of $\Delta S \bmod T$ versus T for code $\mathcal{C}_7$ and $E_b/N_0 = 6$ dB.

FIGURE 6: Entropy of $\Delta S \bmod T$ versus T for code $\mathcal{C}_9$ and $E_b/N_0 = 6$ dB.

that for $\mathcal{C}_7$, $\mathbb{P}(\Delta S = 2\delta), \delta \in \mathbb{Z}$ is very low (actually theoretically equal to zero for infinite sequences), resulting in low values of $H(\Delta S \bmod 2\delta)$. This result explains why the decoding performance of code $\mathcal{C}_7$ is worse on an aggregated model of parameter $T = 2t$ than on a one of parameter $T = 2t - 1$ (as depicted on Table 1). For the parameters of Figure 5, there exists a finite value of T such that $H(\Delta S) = H(\Delta S \bmod T)$. Indeed, for $T = 9$, these two quantities are equal. They are very close for $T = 5$ and $T = 7$. We can hence expect
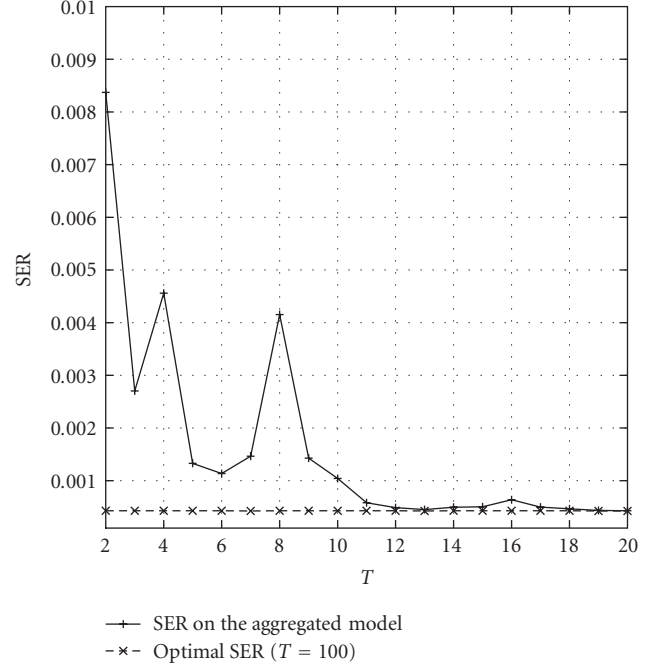

FIGURE 7: Symbol-error rate versus T for code $\mathcal{C}_9$ and $E_b/N_0 = 6$ dB.

the decoding performance of code $\mathcal{C}_7$ to be similar on both models for these values of T. Table 1 confirms that the decoding performance of code $\mathcal{C}_7$ on the trellis corresponding to $T = 9$ is the same as the performance on the optimal trellis at $E_b/N_0 = 6$ dB.

In Figure 6, $H(\Delta S \bmod T)$ is depicted versus T for code $\mathcal{C}_9$ and $E_b/N_0 = 6$ dB. We observe that for $T = 7$ and $T = 8$, $H(\Delta S \bmod T)$ is lower than for $T = 6$. The decoding performance of $\mathcal{C}_9$ follows the variations of $H(\Delta S \bmod T)$ as it can be seen on Figure 7: the higher $H(\Delta S \bmod T)$, the lower the associated SER. We can also observe that for code $\mathcal{C}_9$, the convergence of $H(\Delta S \bmod T)$ is slower than for $\mathcal{C}_7$. According to Table 1, the convergence in terms of SER of code $\mathcal{C}_9$ is also slower.

We have seen in this section that the PMF of ΔS is a mathematical tool that allows us to analyze the decoding performance of QA codes on the aggregated model presented in this paper. In particular, for a given QA code and signal-to-noise ratio, the behavior of the decoding performance of the code is strongly related to the behavior of $H(\Delta S \bmod T)$. This allows us to determine how the decoding performance of a QA code on the aggregated model converges with respect to T towards the performance of the same code on the optimal model.

6. SOFT-INPUT DECODING WITH SIDE INFORMATION

We have seen in the previous section that the termination constraint on a trellis-based soft-input decoding scheme improves the decoding performance of a QA code. We propose in this section a method to further improve the robustness of QA codes by exploiting length constraints at different time instants of the decoding process. Let us recall that the term length constraint refers to a constraint on the number of
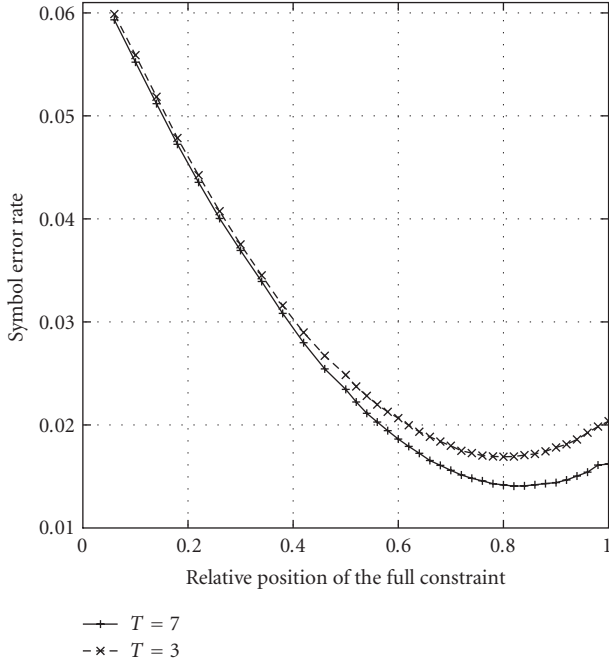
FIGURE 8: SER performance of code $\mathcal{C}_7$ versus the relative position of a full constraint for $E_b/N_0 = 5$ dB.

decoded symbols at a given bit clock instant, and the term termination constraint refers to a length constraint at the last bit clock instant. The termination constraint enforces some synchronization at both ends of the sequence, but not in the middle, as depicted in [19].

In this section, we first study the impact of the position of the length constraint on the decoding performance. Then, we propose a strategy to replace this unique constraint into weaker but a high number of constraints placed at different time instants on the aggregated state model. These constraints increase the likelihood of synchronized paths, leading to enhanced SER performance. Finally, this strategy is extended to introduce extra redundancy in the form of side information for robust decoding of QA codes. This approach is an alternative to the forbidden symbol strategy. Note that the method described presents the advantage that the redundancy is not inserted in the bit-stream. The redundancy can hence be more flexibly adapted to varying transmission conditions.

6.1. Impact of the position of the length constraint

A length constraint on the aggregated state model invalidates the sequences for which the number of decoded symbols modulo T (M_k) at a bit clock instant k differs from a given value m_k. All sequences which do not pass in states of the trellis such that $M_k = m_k$ are discarded. This constraint can be exploited at the end of the decoding process as a termination constraint. In this case, it becomes $M_{L(\mathbf{X})} = L(\mathbf{S}) \bmod T$. The amount of information required to signal this so-called *modulo* constraint to the decoder is of the order of $\nu = H(L(\mathbf{S}) \bmod T) \approx \log_2 T$ bits. We assume in the

following that $\nu \in \mathbb{N}^*$. The rest of this subsection is dedicated to optimize the use of these ν bits, which can be seen as bits of side information.

Length constraints are usually considered at the last bit clock time instant $L(\mathbf{X})$ of the decoding process, but they need not be. The modulo constraint can in particular be used at any intermediate time instants $k = 1, \ldots, L(\mathbf{X})$, thus helping the decoder to resynchronize at these instants. The SER decoding performance actually depends on the relative position k of the modulo constraint. Figure 8 shows the SER performance of code $\mathcal{C}_7$ against the position k of the constraint on M_k. The optimal position depends on the code considered, on the signal-to-noise ratio and on the parameter T of the trellis. At $E_b/N_0 = 5$ dB and $T = 3$, the optimal relative position of the constraint is equal to 0.8 and the corresponding SER is equal to 0.016911, which outperforms the value SER = 0.020366 obtained when the constraint is placed at the end of the bit-stream. For $T = 7$, the best SER is obtained for a relative position of the constraint equal to 0.82. The corresponding SER is equal to 0.014058, while the SER obtained when the constraint is placed at the end of the bit-stream is equal to 0.01620.

6.2. Spreading a length constraint into partial constraints

The modulo constraint can be further spread along the decoding process. A constraint is called *partial* if the decoder is not given the entire binary value of m_k, but only a subset of ω bits among the ν bits. This partial constraint actually partitions the set $\mathcal{I}_T = \{0, \ldots, T - 1\}$ of all possible modulo values into 2^ω subsets of cardinal $2^{\nu - \omega}$. The partitions of $\mathcal{I}_{T=8}$ corresponding to the first, second, or last bit of m_k are, respectively, given by

$$
\begin{aligned}
\mathcal{P}_1 &= \{\{0, 1, 2, 3\}, \{4, 5, 6, 7\}\}, \\
\mathcal{P}_2 &= \{\{0, 1, 4, 5\}, \{2, 3, 6, 7\}\}, \\
\mathcal{P}_3 &= \{\{0, 2, 4, 6\}, \{1, 3, 5, 7\}\}.
\end{aligned}
\tag{16}
$$

For a fixed rate of side information, the frequency of the partial constraints will be increased, which protects the synchronization of the decoded sequence all along the decoding process.

The question to be addressed is thus the type of constraints to be used, their frequency and the instants at which they should be used to have the lowest SER. Such a best configuration actually depends on E_b/N_0, T, and on the code considered, but it may be found for each set of parameters using a simulated annealing algorithm. Table 4 depicts the effect of 3 partial constraints of 1 bit on the SER for $E_b/N_0 = 6$ dB, code $\mathcal{C}_7$ and gives the best solution (last line). The best configuration leads to SER = 0.002903, hence outperforming the value SER = 0.003969 obtained with the best full constraint position for the same amount of side information (and all the more the value SER = 0.004335 obtained with the usual configuration, i.e., the termination constraint placed at the last bit clock instant).

TABLE 4: Positioning configurations of 3 partial constraints of 1 bit and corresponding SER results for code $\mathcal{C}_7$, $E_b/N_0 = 6$ dB, and $T = 8$.

| Relative positions of the partial constraints	Partitions	SER
0.40 0.64 0.88	$\mathcal{P}_3 \; \mathcal{P}_3 \; \mathcal{P}_3$	0.0138598
0.40 0.64 0.88	$\mathcal{P}_3 \; \mathcal{P}_3 \; \mathcal{P}_2$	0.0085099
0.40 0.64 0.88	$\mathcal{P}_2 \; \mathcal{P}_3 \; \mathcal{P}_2$	0.0057394
0.40 0.64 0.88	$\mathcal{P}_3 \; \mathcal{P}_2 \; \mathcal{P}_1$	0.0054139
0.40 0.64 0.88	$\mathcal{P}_3 \; \mathcal{P}_2 \; \mathcal{P}_2$	0.0053022
0.42 0.66 0.84	$\mathcal{P}_3 \; \mathcal{P}_3 \; \mathcal{P}_3$	0.0139033
0.42 0.66 0.84	$\mathcal{P}_3 \; \mathcal{P}_3 \; \mathcal{P}_2$	0.0057883
0.42 0.66 0.84	$\mathcal{P}_2 \; \mathcal{P}_3 \; \mathcal{P}_2$	0.0043527
0.42 0.66 0.84	$\mathcal{P}_3 \; \mathcal{P}_2 \; \mathcal{P}_1$	0.0055276
0.42 0.66 0.84	$\mathcal{P}_3 \; \mathcal{P}_2 \; \mathcal{P}_2$	0.0045336
0.44 0.66 0.82	$\mathcal{P}_3 \; \mathcal{P}_3 \; \mathcal{P}_3$	0.0144611
0.44 0.66 0.82	$\mathcal{P}_3 \; \mathcal{P}_3 \; \mathcal{P}_2$	0.0056628
0.44 0.66 0.82	$\mathcal{P}_2 \; \mathcal{P}_3 \; \mathcal{P}_2$	0.0045738
0.44 0.66 0.82	$\mathcal{P}_3 \; \mathcal{P}_2 \; \mathcal{P}_1$	0.0054556
0.44 0.66 0.82	$\mathcal{P}_3 \; \mathcal{P}_2 \; \mathcal{P}_2$	0.0047592
0.40 0.71 0.99	$\mathcal{P}_2 \; \mathcal{P}_3 \; \mathcal{P}_2$	0.0029031
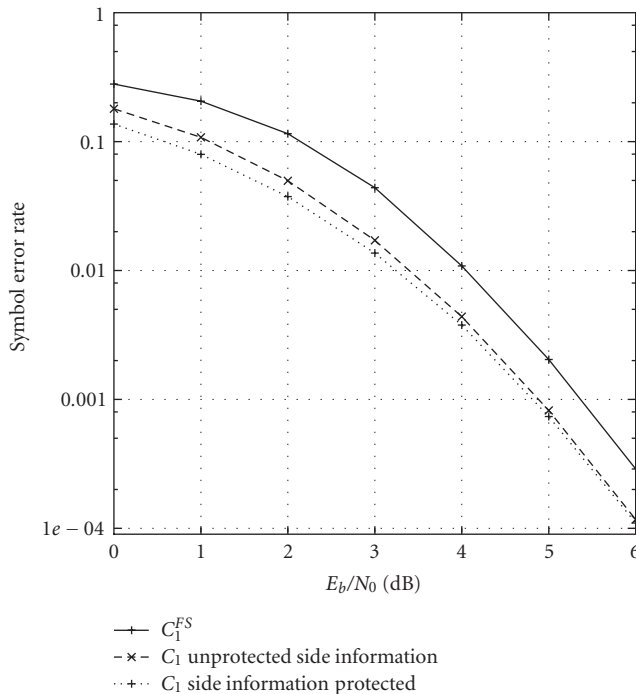
FIGURE 9: SER versus signal-to-noise ratio for $\mathcal{C}_1$ with side information and $\mathcal{C}_1^{FS}$. The transmission rate is equal to 1.074 bits/symbol.

6.3. Robust decoding of QA codes with side information

6.3.1. Description of the decoding scheme

To improve the robustness of QA codes, many techniques have been proposed as the introduction of a forbidden symbol in the encoding process to detect and correct errors [8] or the introduction of synchronization markers inside the bit-stream [10]. These techniques are based on adding redundancy in the encoded bit-stream. In this section, we propose a new technique to provide redundancy in the bit-stream based on the introduction of length constraints all along the decoding process. In the previous part, we have seen that exploiting more partial constraints leads to better decoding performance, provided that the kind of constraints and their positions are appropriately chosen. Let us consider the transmission of a source sequence $\mathbf{S}$ of $L(\mathbf{S})$ binary symbols. This sequence is encoded by a QA code of compression rate R_q. The bit length of the encoded message is denoted $L(\mathbf{X})$. To achieve a desired bit transmission rate R_t ($R_t > R_q$), $\nu_b = \lfloor (R_t - R_q) \times L(\mathbf{S}) \rfloor$ bits of side information are used to signal length constraints at the decoder side. According to the results of the first parts of this section, ν_b partial constraints of 1 bit will be used. This ν_b constraints are given at the uniformly spaced bit positions $\{\lfloor k \times L(\mathbf{X})/\nu_b \rfloor, 1 \leq k \leq \nu_b\}$. The bits of side information corresponding to the length constraints are sent separately from the bit-stream and can hence be chosen to be protected against channel noise or not.

6.3.2. Performance comparison with the forbidden symbol technique [8]

This method to provide additional redundancy to QA codes has been implemented and compared to the technique based on introducing a forbidden symbol in the encoding process. We have considered two 3-bit precision QA codes, $\mathcal{C}_1$, and $\mathcal{C}_2$, respectively, adapted to the source probabilities $\mathbb{P}(a) = 0.7$ and $\mathbb{P}(a) = 0.8$. We have also considered the associated QA codes $\mathcal{C}_1^{FS}$ and $\mathcal{C}_2^{FS}$ constructed with a forbidden symbol of probability 0.1. The automata of these codes have been computed according to the method given in [12]. The EDLs of codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are equal to $edl_1 = 0.890$ and $edl_2 = 0.733$, respectively. The ones of codes $\mathcal{C}_1^{FS}$ and $\mathcal{C}_2^{FS}$ are equal to $edl_1^{FS} = 1.074$ and $edl_2^{FS} = 1.045$, respectively.

Hence, in order to have a fair comparison between the decoding performance obtained with this strategy with respect to the one obtained with the forbidden symbol, that is between $\mathcal{C}_1$ with $\mathcal{C}_1^{FS}$, we have transmitted a total number of $\lfloor (edl_1^{FS} - edl_1) \times L(\mathbf{S}) \rfloor$ extra bits used as partial constraints in the decoding process on the aggregated model. Similarly, to compare $\mathcal{C}_2$ with $\mathcal{C}_2^{FS}$, we have transmitted $\lfloor (edl_2^{FS} - edl_2) \times L(\mathbf{S}) \rfloor$ extra bits. We have also considered both cases where the side information is assumed to be protected by an error correcting code and the case where it is impacted by the same channel noise as the compressed stream.

The SER performance of the two strategies ($\mathcal{C}_1$ and $\mathcal{C}_1^{FS}$) are shown in Figure 9 and similarly in Figure 10 for $\mathcal{C}_2$ with $\mathcal{C}_2^{FS}$. These curves have been obtained for 10^5 sequences of 100 symbols. The SER for both strategies has been obtained by running a Viterbi algorithm on an aggregated model of parameter $T = 5$. It can be observed that for both codes the proposed technique outperforms the strategy based on a forbidden symbol, even in the case where the side information is not protected, hence affected by the same channel noise as
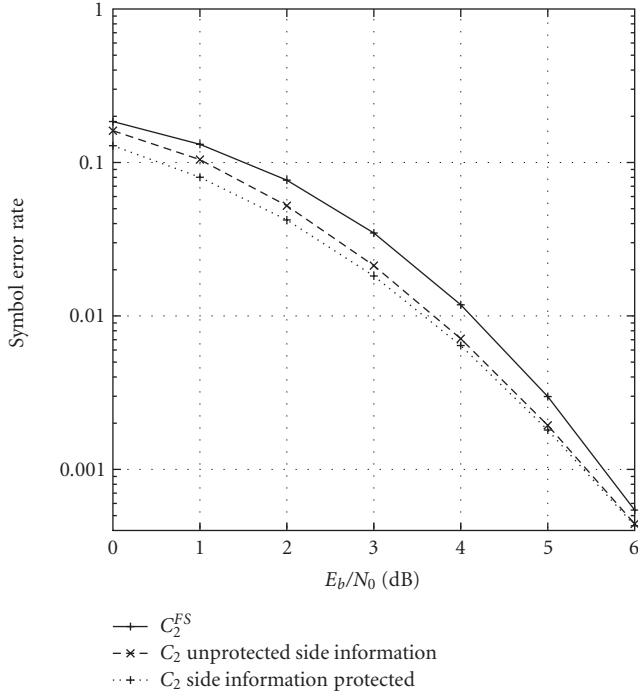
FIGURE 10: SER versus signal-to-noise ratio for $\mathcal{C}_2$ with side information and $\mathcal{C}_2^{\text{FS}}$. The transmission rate is equal to 1.045 bits/symbol.

the compressed data. Note that the possibility to protect it is extra flexibility that does not offer the method based on the forbidden symbol.

7. CONCLUSION

In this paper, a set of aggregated state models for soft-input decoding of QA codes is presented. These models are configured by an aggregation parameter T that controls the trade-off between estimation accuracy and the complexity of the decoding. Soft-input decoding performance of QA codes on these models are presented. These models yield very close decoding performance compared to the optimal decoding performance using a complete state model. Then, the method proposed in [1] to analyze error recovery properties of VLC is extended to QA codes. This method allows the calculation of the PMF of the *gain/loss* of symbols caused by channel errors. The entropy of the *gain/loss* turns out to be an appropriate tool to analyze the decoding performance on the aggregated model and compare it to the one on the optimal model. By analyzing the impact of the position of the length constraint relatively to the bit-stream, it turns out that the optimum position of this constraint is not at the last bit clock instant, as it usually considered. Indeed, a constraint only protects the bits in the neighborhood of its position. This result led us to propose a new approach to add redundancy to QA codes under the form of length constraints all along the decoding process. This approach improves the decoding of QA codes and outperforms the approach based on QA codes using a forbidden symbol in the encoding process. Note that the proposed

approach presents the advantage that the redundancy is not need inserted in the bit-stream. The redundancy can hence be more flexibly adapted to varying transmission conditions.

REFERENCES

[1] J. Maxted and J. Robinson, "Error recovery for variable length codes," *IEEE Transactions on Information Theory*, vol. 31, no. 6, pp. 794–801, 1985.

[2] P. F. Swaszek and P. DiCicco, "More on the error recovery for variable-length codes," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 2064–2071, 1995.

[3] G. Zhou and Z. Zhang, "Synchronization recovery of variable-length codes," *IEEE Transactions on Information Theory*, vol. 48, no. 1, pp. 219–227, 2002.

[4] S. Malinowski, H. Jégou, and C. Guillemot, "Synchronization recovery and state model reduction for soft decoding of variable length codes," *IEEE Transactions on Information Theory*, vol. 53, no. 1, pp. 368–377, 2007.

[5] H. Jégou, S. Malinowski, and C. Guillemot, "Trellis state aggregation for soft decoding of variable length codes," in *Proceedings of IEEE Workshop on Signal Processing Systems Design and Implementation (SiPS '05)*, pp. 603–608, Athens, Greece, November 2005.

[6] J. J. Rissanen, "Generalized kraft inequality and arithmetic coding," *IBM Journal of Research and Development*, vol. 20, no. 3, pp. 198–203, 1976.

[7] G. F. Elmasry, "Arithmetic coding algorithm with embedded channel coding," *Electronics Letters*, vol. 33, no. 20, pp. 1687–1688, 1997.

[8] C. Boyd, J. G. Cleary, S. A. Irvine, I. Rinsma-Melchert, and I. H. Witten, "Integrating error detection into arithmetic coding," *IEEE Transactions on Communications*, vol. 45, no. 1, pp. 1–3, 1997.

[9] I. Kozintsev, J. Chou, and K. Ramchandran, "Image transmission using arithmetic coding based continuous error detection," in *Proceedings of the Data Compression Conference (DCC '98)*, pp. 339–348, Snowbird, Utah, USA, March 1998.

[10] T. Guionnet and C. Guillemot, "Soft and joint source-channel decoding of quasi-arithmetic codes," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 3, pp. 393–411, 2004.

[11] G. G. Langdon Jr., "An introduction to arithmetic coding," *IBM Journal of Research and Development*, vol. 28, no. 2, pp. 135–149, 1984.

[12] S. Ben-Jamaa, C. Weidmann, and M. Kieffer, "Asymptotic error-correcting performance of joint source-channel schemes based on arithmetic coding," in *Proceedings of IEEE 8th Workshop on Multimedia Signal Processing (MMSP '06)*, pp. 262–266, Victoria, BC, Canada, October 2006.

[13] P. Howard and J. Vitter, "Practical implementations of arithmetic coding," in *Image and Text Compression*, J. A. Storer, Ed., pp. 85–112, Kluwer Academic Publishers, Boston, Mass, USA, 1992.

[14] M. J. Gormish and J. D. Allen, "Finite state machine binary entropy coding," in *Proceedings of the Data Compression Conference (DCC '93)*, pp. 449–459, Snowbird, Utah, USA, March-April 1993.

[15] M. J. Gormish, "Source coding with channel, distorsion, and complexity constraints," Ph.D. dissertation, Stanford University, Stanford, Calif, USA, 1994.

[16] M. Grangetto, P. Cosman, and G. Olmo, "Joint source/channel coding and MAP decoding of arithmetic codes," *IEEE Transactions on Communications*, vol. 53, no. 6, pp. 1007–1016, 2005.

[17] J. Anderson and C.-W. Ho, "Architecture and construction of a hardware sequential encoder for speech," *IEEE Transactions on Communications*, vol. 25, no. 7, pp. 703–707, 1977.

[18] S. J. Mason, "Feedback theory-further properties of signal flow graphs," *Proceedings of the Institute of Radio Engineers*, vol. 44, no. 7, pp. 920–926, 1956.

[19] J. Kliewer and R. Thobaben, "Iterative joint source-channel decoding of variable-length codes using residual source redundancy," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 919–929, 2005.