Error Correction for Asynchronous Communication

Chen Yi and Jörg Kliewer
Helen and John C. Hartmann Department of Electrical and Computer Engineering
New Jersey Institute of Technology
Email: cy235@njit.edu, jkliewer@njit.edu

***Abstract*—We propose a forward error correction scheme for asynchronous sensor communication where the dominant errors consist of pulse deletions and insertions, and where encoding is required to take place in an instantaneous fashion. The presented scheme consists of a combination of a systematic convolutional code, an embedded marker code, and power-efficient frequency-shift keying (FSK) modulation at the sensor node. Decoding is first obtained via a maximum a-posteriori (MAP) decoder for the marker code which achieves synchronization for the insertion and deletion channel, followed by MAP decoding for the convolutional code. Besides investigating the rate trade-off between marker and convolutional codes, we also show that residual redundancy in the asynchronously sampled and quantized source signal can be successfully exploited in combination with redundancy only from a marker code. This provides a low complexity alternative for deletion and insertion error correction compared to using explicit redundancy.**

I. Introduction

In recent years there has been growing demand of low-power small-size sensors for short-range wireless sensing applications (see, e.g., [1–3]), such as environmental observation, biomedical, and health care monitoring. Typically, such applications require extremely power efficient integrated sensors capable of providing reliable wireless links under low circuit complexity, long battery life, and a small sensor footprint.

One potential approach to address the challenge of a low-complexity sensing operation is to replace classical Nyquist-based synchronous signal processing by an asynchronous sensing architecture combined with an asynchronous wireless interface. In fact, it has been shown that the Nyquist "sampling and quantization" approach is not the optimal solution for recording highly correlated analog waveform data or sparse signals [4], which are ubiquitous in most low-power sensing applications. Note that the power saving advantage of asynchronous sensing generally lies in the fact that if the input signal is inactive or no changes are detected, no sampling and transmission is made [5]. In contrast to the sensor node it is often assumed that the base station is wall powered and thus is able to run sufficiently complex algorithms.

While there has been significant activity related to understanding the theory and implementation of asynchronous sampling, only little work has been reported addressing the difficulty of *communicating* asynchronous samples in a reliable fashion over noisy channels. The drawback of uncoded asynchronous communication lies in the fact that the channel noise may lead to symbol insertions and deletions at the receiver, thus destroying the data synchronization [6].

On the other hand, a large amount of work has been devoted to insertion and/or deletion block error correcting codes for *synchronous* communication systems (see, e.g., [7]). Recently, concatenated codes have been considered, for example the concatenation of an outer forward error correction code and an inner nonlinear resynchronization code by inserting a specific number of marker bits which are exploited at the decoder to maintain synchronization of the bitstream (see, e.g., [8–11]). Also, convolutional codes for the insertion/deletion channel have been considered by either extending the state space of the code [12] or by modifying the path metric of the Viterbi decoder [13].

In this paper, we extend our previous results for the uncoded case in [6] to coded asynchronous communication. To the best of our knowledge, this is the first time that error correction has been addressed in the context of a communication system based on asynchronous sampling. Note that synchronous insertion/deletion error correcting schemes from above cannot directly be applied to the asynchronous setting. For example, the fact that in asynchronous communication the information about the underlying waveform signal is mostly contained in the timing information of the transmitted signal pulses only allows to embed code redundancy via extending the modulation alphabet and not by adding extra pulses. For the same reason, the code must be necessarily systematic.

These constraints are addressed in the following by a deletion/insertion correction scheme based on a combination of a systematic convolutional code, an embedded marker code, and power-efficient frequency-shift keying (FSK) modulation at the sensor node tailored to the asynchronous setting. Note that employing a convolutional code allows for encoding in a streaming fashion with low latency and only requires a buffer length of a few bits at the complexity constrained sensor node. We also show that residual redundancy in the asynchronously sampled and quantized source signal can be successfully exploited for synchronization in combination with a marker code, thus providing an extremely low complexity alternative to using explicit redundancy from a channel code.

II. Asynchronous Sampling

We first describe the asynchronous sample acquisition process carried out at the sensor node [6]. Fig. 1 shows a waveform quantized by using asynchronous delta modulation, where $S(t_k)$ denotes the amplitude of the waveform signal

This work was supported in part by U.S. National Science Foundation grant ECCS-1407910.

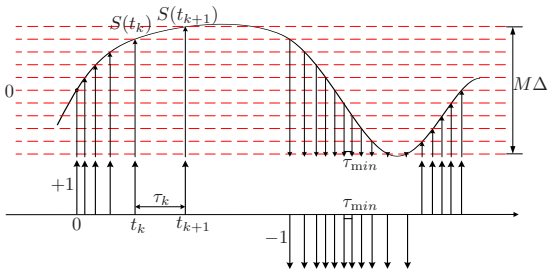

Fig. 1. Asynchronous sampling via asynchronous delta modulation [6].

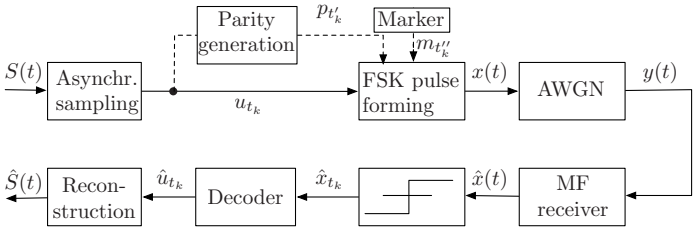

Fig. 2. System model

at time t_k. At each time the waveform is compared with M thresholds, with the quantization decision interval defined as $\Delta \triangleq \frac{2\max_t(|S(t)|)}{M}$. Whenever the waveform exceeds a decision threshold in the direction of increasing amplitude a "+1" sample is recorded at that specific time, otherwise a "-1" sample is placed; "+1" and "-1" are mapped to bits 1 and 0, respectively. In order to implement this scheme, no clock circuit is required at the sensor node compared to a traditional Nyquist based "sample and quantize" operation, which significantly reduces the power consumption of the sensor node.

III. System Model

The system model employed in this paper is shown in Fig. 2. First, the analog waveform signal $S(t)$ is asynchronously sampled into the sample bits u_{t_k} at time t_k, $k = 1, \ldots, T$, as described in Section II, where T denotes a prescribed number of transmitted bits. Then, along with generated parity bits $p_{t'_k}$ and marker bits $m_{t''_k}$ to ensure synchronization, Q-FSK pulses comprising orthogonal sinc waveforms are generated. FSK modulation is employed due to its power efficiency and its suitability for ultra wideband radio operation (see [6]). We restrict ourselves to $Q = 2$ for the uncoded and to $Q = 4$ for the coded case, respectively. Note that in order to preserve the timing information generated in the sampling process, redundancy can only be added by extending the symbol alphabet, which also extends the bandwidth of the transmission.

Fig. 3 addresses the *coded* case for $Q = 4$ and shows how asynchronously sampled information bits u_{t_k}, parity bits $p_{t'_k}$, and marker bits $m_{t''_k}$ are arranged to form Q-ary modulation symbols. Each of these symbols is transmitted at a specific time t_k determined by the timing information generated in the asynchronous sampling stage. During the first block of K symbols, only information bits are communicated, with the parity bits set to zero. At time t_K, a marker bit m_{t_K} is added, whose value is fixed and assumed to be known at the receiver.

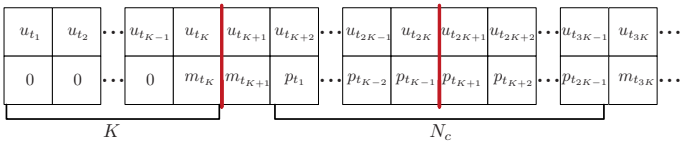

Fig. 3. Combination of information and redundant bits for FSK pulse forming. A 4-FSK symbol is obtained by combining the bits stacked on top of each other.

The receiver uses this marker information to resynchronize the received bit sequence from deletion and insertion errors. The next block in Fig. 3 contains information bits from time t_{K+1} to t_{2K} and another marker bit at time t_{K+1}. The redundancy in the rest of the block is constituted by parity bits generated from the information bits in the *previous* length-K block. This "chaining" approach for the parity bits is required due to the causality of the encoding process and the fact that the timing information t_k for each waveform sample must be preserved. The number of samples (or FSK pulses) between two blocks of marker bits is defined as N_c. As one important practical constraint, typically K is required to be quite small, since it is directly related to chip area and energy consumption of the sensor. For that reason we will employ parity bits $p_{t'_k}$ obtained from a systematic convolutional code with rate $R_c = K/N$. Also, let R_m denote the coderate for the marker code. Then, for $Q = 4$ we obtain the constraint $R_c R_m = 1/2$ for the tradeoff between the two coderates.

After encoding, the modulated signal $x(t)$ is transmitted over an AWGN channel. The observation at the channel output $y(t)$ is applied to a matched filter (MF) receiver and the channel decoder which ensures resynchronization (i.e., insertion and deletion error correction) and the correction of substitution errors on the channel, followed by the waveform reconstruction stage. In the MF receiver, the output of the Q matched filters is compared with a threshold on a very fine grid (by running a high frequency local clock at the base station), and whenever the threshold is exceeded, the corresponding Q-ary symbol is reconstructed. A deletion error occurs when the energy waveform after the MF lies below a certain threshold γ. In contrast, an insertion takes place when channel noise during a silent phase triggers the threshold.

IV. Error Correction

In [10] a forward backward algorithm (FBA) for correcting insertion, deletion, and substitution errors for *synchronous* transmission was introduced based on ideas from [8]. We define $x_1^T = (x_1, x_2, ..., x_T)$ as the transmitted bit sequence of length T when conveyed through the deletion/insertion channel. Likewise, $y_1^R = (y_1, y_2, ..., y_R)$ is the received sequence of length R, where both R and T are assumed to be known at the receiver. The sequence x_1^T is composed of equal length information segments interrupted by periodic marker bits. $D_{k,n}$ is defined as the event that when the transmitted bit length is k, the received bits length is exactly n, where $k \in \{1, 2, 3, ..., T\}$, $n \in \{1, 2, 3, ..., R\}$. $D_{k-1,n-1} \rightarrow D_{k,n}$ represents a transmitted bit potentially suffering a substitution error, while $D_{k-1,n} \rightarrow D_{k,n}$ and $D_{k-1,n-2} \rightarrow D_{k,n}$ denote a bit being deleted and inserted during transmission, respectively.

As in the classical BCJR algorithm [14] we define $\alpha_{k,n} \triangleq P(D_{k,n}, y_1^n)$ and $\beta_{k,n} \triangleq P(y_{n+1}^R | D_{k,n})$. Then, forward and backward recursion are defined as [8]

$$\alpha_{k,n} = \frac{P_i}{4}\alpha_{k-1,n-2} + P_d\alpha_{k-1,n} + P_t\alpha_{k-1,n-1}\sum_{x_k} P(x_k)(1-P_s)^{\delta_{x_k,y_n}} P_s^{1-\delta_{x_k,y_n}}, \quad (1)$$

with $\delta_{x,y}$ denoting the Kronecker delta. Further, P_i and P_d denote insertion and deletion probabilities, resp., P_s is defined as the substitution (symbol) error probability, and $P_t \triangleq 1 - P_i - P_d$. The recursion for $\beta_{k,n}$ is defined similarly. Finally, we obtain

$$P(y_1^R|x_k) = \sum_{n=2}^{\min(2k,R)} \frac{P_i}{4}\alpha_{k-1,n-2}\beta_{k,n} + \sum_{n=0}^{\min(2k,R)} P_d\alpha_{k-1,n}\beta_{k,n} + \sum_{n=1}^{\min(2k,R)} P_t\alpha_{k-1,n-1}\beta_{k,n}(1-P_s)^{\delta_{x_k,y_n}} P_s^{1-\delta_{x_k,y_n}}. \quad (2)$$

A. Inner decoding of synchronization errors

We now employ the above algorithm with some modifications as inner decoder for synchronization errors in the asynchronous communication setup described in Section III. For the sake of brevity we denote timing information t_k as k, whenever there is no ambiguity. Fig. 3 indicates that the symbol alphabet is now defined over $\mathbb{F}_4$, which means that (1), the corresponding equation for $\beta_{k,n}$, and (2) must be modified for the resulting symbol based FBA. Specifically, the *a priori* probability $P_i/4$ must be replaced with $P_i/16$. Further, for $x_k, y_n \in \mathbb{F}_4$ if x_k contains a marker bit 0, i.e., $x_k \in \{[00],[10]\}$, we have $P(x_k = [00]) = P(x_k = [10]) = 0.5$. Likewise, if x_k contains a marker bit 1, i.e., $x_k \in \{[01],[11]\}$ we have $P(x_k = [01]) = P(x_k = [11]) = 0.5$, otherwise we obtain $P(x_k) = 0.25$. We can calculate the conditional marginal log-lokelihood ratio (LLR) for the information bit u_k at time t_k from the output of the symbol-based FBA as

$$L(u_k|y_1^R) = \log\left(\frac{P(x_k=[00]|y_1^R) + P(x_k=[01]|y_1^R)}{P(x_k=[10]|y_1^R) + P(x_k=[11]|y_1^R)}\right). \quad (3)$$

These LLRs can be used to directly determine the transmitted data via hard decision decoding or can be further utilized in the outer decoder for the convolutional code. For this we apply the standard BCLR algorithm [14], which is employed to recover the substitution errors remaining at the output of the inner decoder. Here, the information block length K in Fig. 3 is the number of information bits associated with a single state transition for the convolutional code. Even if we are able to recover the position of a deleted pulse as outlined below, its value is still unknown and thus can be seen as an erased symbol with respect to the outer convolutional code.

B. Location of deleted and inserted symbols

The obtained results from inner synchronization decoding are now employed to localize the position of deleted and inserted FSK pulses by visualizing the evolution of transmitted and received symbols on a two dimensional grid of dimension $T \times R$. Examples are shown in Fig. 4. A diagonal path with $n(k) = k$ means that no insertions or deletions have occured during transmission. In contrast, for synchronization errors the path deviates from the diagonal line, where deletion errors cause the path to move upwards, and insertion errors downwards, respectively. For example, in Fig. 4(a), there is a deletion at transmitted position 50, and Fig. 4(b) shows both a deletion and an insertion at transmitted positions 197 and 212, respectively.

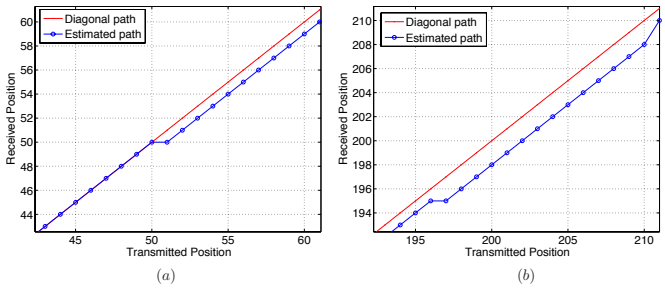

Fig. 4. Optimum path to locate deletions/insertions

By leveraging the results from the symbol-based FBA, this path can be estimated by obtaining the most likely grid point $\hat{n}(k)$ at position k belonging to this path as

$$\hat{n}(k) = \arg\max_n \left\{\frac{P_i}{16}\alpha_{k-1,n-2}\beta_{k,n},\ P_d\alpha_{k-1,n}\beta_{k,n},\ P_t\alpha_{k-1,n-1}\beta_{k.n}(1-P_s)^{\delta_{x_k,y_n}} P_s^{1-\delta_{x_k,y_n}}\right\}. \quad (4)$$

For insertions errors we can remove the extra pulses directly where they are located. However, the exact position of a deleted pulse after MF filtering between adjacent pulses at times t_k and t_{k+1} is still unknown and cannot be recovered at the receiver. As a workaround we have empirically obtained the distribution of the relative position of deleted pulses between its neighbors for a typical waveform signal in medical applications, a mouse heart beat signal of 10 ms duration, asynchronously quantized with $M = 63$ threshold levels. The resulting distribution for the normalized sample position is shown in Fig. 5. As we can see, the distribution is highly concentrated around the midpoint of the interval, such that placing a deleted pulse at this midpoint will only incur a small additional end-to-end distortion in the average.

C. Exploiting residual source redundancy

In the following we introduce a simple model for data bits obtained by asynchronously quantizing the waveform source signal. This allows us to exploit residual source redundancy in the outer decoder instead of explicit redundancy from a convolutional code. The proposed model makes use of the simple observation that for an asynchronously sampled waveform signal with moderate M the obtained data sequence contains alternating contiguous runs of the form "00..." and "11...". We consider the most conservative case by addressing only the two-bit "runs" "00" and "11", which leads to the four-state Markov chain depicted in Fig. 6. Based on the

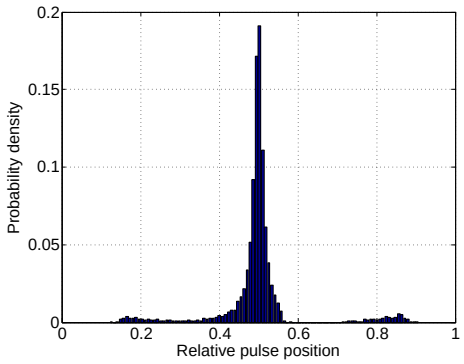

Fig. 5. Distribution of the relative position of deleted pulses between its neighbors for a typical waveform signal in medical applications (heart beat signal of a mouse).

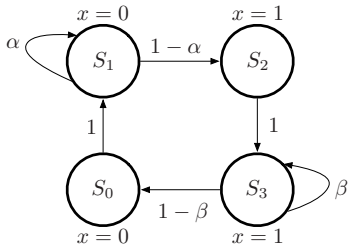

Fig. 6. Four-state Markov chain with states $S_0, \ldots, S_3$ for the data bits obtained by asynchronously quantizing the source signal. x indicates the output for each state.

transition probabilities α and β we can compute the steady state probabilities μ_i for state S_i, $i \in \mathcal{I}$, $\mathcal{I} = \{0, 1, 2, 3\}$, as $\mu_0 = \frac{\bar{\alpha}\bar{\beta}}{2\bar{\alpha}\beta+\bar{\alpha}+\bar{\beta}} = \mu_2$, $\mu_1 = \frac{\bar{\beta}}{2\bar{\alpha}\beta+\bar{\alpha}+\bar{\beta}}$, and $\mu_3 = \frac{\bar{\alpha}}{2\bar{\alpha}\beta+\bar{\alpha}+\bar{\beta}}$ with $\bar{\alpha} \triangleq 1-\alpha$ and $\bar{\beta} \triangleq 1-\beta$, respectively. For the above mentioned mouse heartbeat signal and $M = 63$ we obtain $\alpha = 0.9533$ and $\beta = 0.9537$. The entropy rate of this source can be computed as $H(\mathcal{X}) = -\sum_{i\in\mathcal{I}} \mu_i \sum_{j\in\mathcal{I}} p_{ij} \log_2 p_{ij} = 0.1798$ bits with $p_{01} = p_{23} = 1$, $p_{11} = \alpha$, $p_{12} = \bar{\alpha}$, $p_{33} = \beta$, $p_{34} = \bar{\beta}$, all other p_{ij} are zero.

We can now use a variant of the BCJR algorithm to compute the *a posteriori* probabilities $P(\Psi_k = p|y_1^R)$ for the state Ψ_k at time t_k, where $p, q \in \mathcal{I}$, as

$$P(\Psi_k = p|y_1^R) = \sum_{q\in\mathcal{I}_p} P(\Psi_k = p, \Psi_{k+1} = q|y_1^R) = \sum_{q\in\mathcal{I}_p} \alpha_k(p)\gamma_k(p,q)\beta_{k+1}(q), \quad (5)$$

where $\mathcal{I}_p$ represents the set of all states at time t_{k+1} which are connected to state $\Psi_k = p$. We further have

$$\alpha_{k+1}(q) = \sum_{p\in\mathcal{I}} \alpha_k(p)\gamma_k(p,q), \quad (6)$$

with a similar definition for the $\beta_k(p)$ term in (5). Finally, the $\gamma_k(p,q)$-term is given as

$$\gamma_k(p,q) = P(\Psi_{k+1} = q|\Psi_k = p) \cdot \underbrace{P(y_1^R|\Psi_k = p)}_{P(y_1^R|x_k)} \cdot \underbrace{P(y_1^R|\Psi_{k+1} = q)}_{P(y_1^R|x_{k+1})} \quad (7)$$

where $P(\Psi_{k+1} = q|\Psi_k = p)$ is the state transition probability of the Markov chain in Fig. 6 for $p, q \in \mathcal{I}$. Further, the probabilities $P(y_1^R|\Psi_k = p)$ and $P(y_1^R|\Psi_{k+1} = q)$ are determined by the output of the inner symbol-based FBA and constitute the *a priori* input for the source BCJR algorithm.

V. Simulation Results

In order to evaluate the proposed error correction strategy, we employ an 10 ms excerpt of the above mentioned recorded mouse heart beat signal, asynchronously quantized with $M = 63$ threshold levels. The marker bits are fixed as $(m_{t_{\ell K}}, m_{t_{\ell K+1}}) = (0, 1)$ with ℓ being an odd integer. Table I shows the employed feedforward convolutional codes for different values of N_c along with the corresponding mother coderates R, punctured rates R_c, and marker rates R_m. In order to preserve the timing information of the pulses only parity bits are punctured. Also, the puncturing patterns are selected such that the combined deletion, insertion, and substitution error probabilities are minimized.

TABLE I
Code rates and puncturing schemes for the employed systematic convolutional codes

R	Generators (octal)	Puncturing matrix	R_c	N_c	R_m
$\frac{2}{3}$	$\mathbf{g}^{(1)} = 3$, $\mathbf{g}^{(2)} = 7$	—	$\frac{2}{3}$	2	$\frac{3}{4}$
$\frac{1}{2}$	$\mathbf{g}^{(1)} = 15$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$	$\frac{3}{5}$	4	$\frac{5}{6}$
$\frac{1}{2}$	$\mathbf{g}^{(1)} = 15$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$	$\frac{5}{9}$	8	$\frac{9}{10}$

Fig. 7 displays the achieved end-to-end mean-squared-error (MSE) distortion for the code from the first row of Table I ($R_c = 2/3$) under different MF thresholds γ. For high values of γ deletion errors dominate, whereas for low values of γ insertion errors are the dominant ones. Note that we introduce additional distortion by the potentially erroneous location of inserted and deleted pulses and by approximating the timing information of a deleted pulse by its expected value.

In Fig. 8 we compare the total symbol error probability, i.e., insertion, deletion, and substitution errors for different amounts of marker redundancy N_c and the codes displayed in Table I for a threshold of $\gamma = 0.6$. Despite this figure ignores errors due to incorrect localization of the position of deleted and inserted pulses, it still provides a reasonable assessment for the performance of these schemes. Fig. 8 also shows the performance for a scheme in which all explicit redundancy is constituted by a marker code (i.e., $R_m = 1/2$, $R_c = 1$) and where source redundancy instead of explicit redundancy is used to fix the value of the information bits after deletion/insertion inner decoding. We observe from Fig. 8 that increasing marker redundancy, i.e., smaller values of N_c, also leads to a smaller total error probability if a convolutional code is used to clean up the residual errors after insertion/deletion decoding. However, the scheme based on source redundancy suffers from a higher error probability even it uses $N_c = 0$.

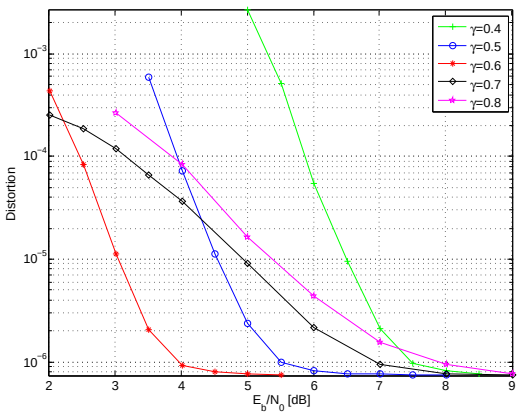

Fig. 7. MSE distortion for the code from the first row of Table I (with $R_c = 2/3$) under different MF threshold values γ.

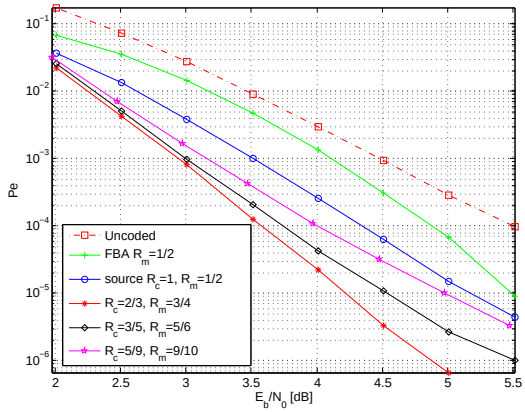

Fig. 8. Total error probability for the codes displayed in Table I and a MF threshold of $\gamma = 0.6$. The scheme with $R_c = 1$ uses residual source redundancy and explicit redundancy solely from a marker code.

This is due to the fact that despite the entropy rate induced by the source redundancy is low, this does not provide any advantage in minimum distance compared to the uncoded case (i.e., we have $d_{\min} = 1$). This is also supported by the observation that for $R_c < 1$ *additionally* exploiting source redundancy after channel decoding does not provide a significant further gain in performance. We can also see from Fig. 8 that solely employing source redundancy provides a significant gain compared to just using the inner marker code and a hard decision after the output of the inner FBA decoder.

The resulting end-to-end MSE distortion is shown in Fig. 9 which also includes errors due to inaccurate localization of deleted and inserted transmit pulses. We observe that all three schemes using explicit redundancy have similar performance, whereas the scheme using implicit source redundancy suffers from a performance penalty of around 0.5 dB in SNR for small to moderate channel SNRs. Also, solely employing a marker code without any additional explicit or implicit redundancy incurs an additional penalty of 0.5 dB for moderate SNRs.

VI. Conclusion

We have presented a concatenated error correction scheme for asynchronous communication comprising a combination of a systematic convolutional code, an embedded marker code, and efficient FSK modulation at the sensor node. Simulation results have shown that if explicit redundancy from a convolutional code is employed, a higher ratio of marker bits provides a lower end-to-end distortion. This observation makes residual source redundancy schemes an attractive low complexity alternative with only a small loss in performance compared to using explicit redundancy from a convolutional code as it allows to use the maximal rate for the marker bits.

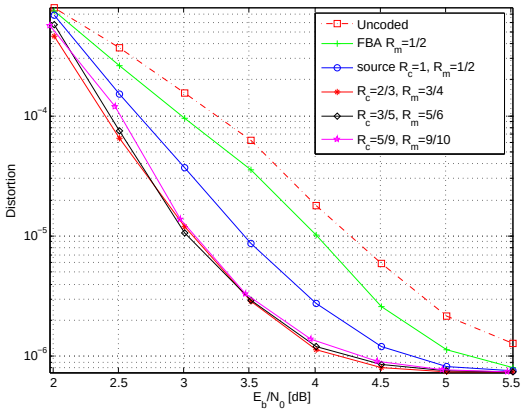

Fig. 9. MSE distortion for $\gamma = 0.6$ and the codes from Fig. 8.

References

[1] J. Foerster, E. Green, S. Somayazulu, and D. Leeper, "Ultra-wideband technology for short- or medium-range wireless communications," *Intel Technology Journal*, vol. 5, no. 2, pp. 1–11, May 2001.

[2] A. Willig, "Recent and emerging topics in wireless industrial communications: A selection," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 2, pp. 102 –124, May 2008.

[3] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Trans. on Industrial Electronics*, vol. 56, no. 10, pp. 4258 –4265, Oct. 2009.

[4] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, "Beyond Nyquist: Efficient sampling of sparse bandlimited signals," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 520–544, 2010.

[5] B. Schell and Y. Tsividis, "A continuous-time ADC/DSP/DAC system with no clock and with activity-dependent power dissipation," *IEEE Journ. of Solid-State Circ.*, vol. 43, no. 11, pp. 2472–2481, Nov. 2008.

[6] Q. Hu, C. Yi, J. Kliewer, and W. Tang, "Asynchonous communication for wireless sensors using ultra wideband impulse radio," in *IEEE 58th International Midwest Symposium on Circuits and Systems*, Fort Collins, CO, Aug. 2015, pp. 1–4.

[7] N. J. A. Sloane, "On single-deletion-correction codes," in *Codes and Designs*. Berlin: Walter de Gruyter, May 2000, pp. 273–291.

[8] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 687–698, Feb. 2001.

[9] J. Chen, M. Mitzenmacher, C. Ng, and N. Varnica, "Concatenated codes for deletion channels," in *Proc. IEEE Int. Sympos. on Inform. Theory*, Yokohama, Japan, Jun. 2003, pp. 218–218.

[10] F. Wang, D. Fertonani, and T. M. Duman, "Symbol-level synchronization and LDPC code design for insertion/deletion channels," *IEEE Trans. Comm.*, vol. 59, no. 5, pp. 1287–1297, May 2011.

[11] F. Wang, D. Aktas, and T. M. Duman, "On capacity and coding for segmented deletion channels," in *Proc. 49th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sep. 2011, pp. 1408–1413.

[12] M. F. Mansour and A. H. Tewfik, "Convolutional decoding in the presence of synchronization errors," *IEEE J. Sel. Areas in Commun.*, vol. 28, no. 3, pp. 218–227, Feb. 2010.

[13] H. Mercier and V. K. Bhargava, "Convolutional codes for channels with deletion errors," in *Proc. 11th Canadian Workshop on Information Theory, 2009*, May 2009, pp. 136–139.

[14] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.