EXTERNAL FEC DESIGN TO INCREASE DATA TRANSFER RATES OVER BAMS RADIOS

Dr. Stéphane Pigeon - Maj. Bart Scheers - Prof. Patrick Verlinde
Royal Military Academy
CISS Laboratory
Renaissancelaan, 30
1000 Brussels, Belgium
E-mail: stephane.pigeon@rma.ac.be - bart.scheers@rma.ac.be - patrick.verlinde@rma.ac.be

ABSTRACT

This paper relates to a study undertaken by the Royal Military Academy aiming at increasing data rates between BAMS tactical radios. The underlying idea consists in using the highest data rate available in a BAMS - namely a 16 kbit/s synchronous transfer mode - and to rely on an external Forward Error Correction (FEC) to correct transmission errors. The paper will provide the reader with the simulated and measured performances of various Reed-Solomon-based codecs working around a user data rate of 9600 bit/s under additive white Gaussian noise and jamming interference.

INTRODUCTION

The BAMS tactical radios were adopted by the Belgian Army in 1993 and are currently in use as a voice and data communication system by the Belgian armed forces. Those transceivers operate between 30MHz and 108MHz with a channel spacing of 25kHz and offer various transmission modes including frequency hopping at rates as high as 250 hops per second without any restrictions whatsoever on the use of the 3,120 available frequencies.

In combination with ruggedised laptops, BAMS transceivers provide a reliable way of sharing information between computers by using an embedded Forward Error Correction (FEC) in order to cope with interferences and jamming.

Under synchronous operating mode and without any embedded error correction applied, transmission speeds up to 16 kbit/s are possible. However, once the build-in FEC is engaged, the highest rate available to the operator decreases to 2400 bit/s. Back in the time when BAMS were introduced, 2400 bit/s were sufficient for most applications under consideration, mostly short text messaging services. By today standards however, this constraint represents a serious bottleneck which slows down the development of future digital battlefield applications.

This paper relates to a study undertaken by the Royal Military Academy aiming at increasing data transfer rates between BAMS transceivers. The underlying idea consists in using the highest data rate available in a BAMS - namely the 16 kbit/s synchronous transfer mode - and to rely on external FEC software plug-ins in the laptop to correct transmission errors. Depending on the error rates one needs to address, various bit rates can be achieved, all above 2400 bit/s.

This paper details the various coding-decoding schemes that have been considered and their performance in correcting errors around a user data rate of 9600 bit/s.

EXPERIMENTAL SETUP

Our experimental setup consists of two BAMS transceivers, each being linked to a laptop by means of a 16 kbit/s serial synchronous connection. At the transmitter side, the laptop plays the role of the FEC encoder and outputs the encoded data to the BAMS emitter clocked at 16 kbit/s. At the remote side, the second BAMS works as a receiver and its attached laptop as a FEC

decoder. Such a setup is illustrated in *Figure 1*.

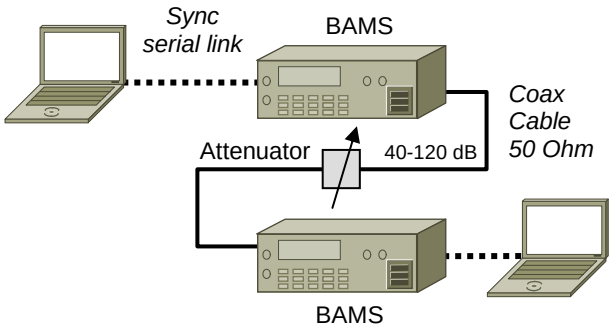

Figure 1 : Experimental setup

The data on the synchronous link between the BAMS transceivers and the laptops are encapsulated in HDLC frames. Basically, an HDLC frame consists of an 8-bit flag followed by data bits and a CRC-16 error detection code. These frames follow each other. When a transmission error occurs, the CRC-16 doesn't match the content of the received frame anymore. In the context of the particular serial interface that has been used[1] *such mismatch ends up in discarding the entire frame at the receiver side*, i.e. a single erroneous bit implies the loss of an entire frame.

REED-SOLOMON

Our study relies on Reed-Solomon-type of error detection and correction algorithms [1]. By adding M additional bytes to a data stream - the so-called parity bytes - Reed-Solomon codes are able to detect and correct up to M/2 erroneous bytes or up to M bytes when one knows where the erroneous bytes are located. This paper relies on shortened Reed-Solomon codes with 32 parity bytes (M=32), either a RS(80,48) for the codec introduced in the next section (EFEC1) or a RS(143,111) later on (EFEC2).

EFEC1

Since complete HDLC frames are discarded when an error occurs, parity bytes embedded within HDLC frames are of no use. This is the reason why, instead of working with parity bytes within frames, the first correction scheme proposed in this paper relies on adding parity *frames* and will be referred as EFEC1 (EFEC stands for External Forward Error Correction scheme).

EFEC1 works as follows:

□ At the emitter side:

- the initial data stream is divided into k_1-byte segments;
- k_2 of these segments are grouped into k_2-by-k_1-byte blocks;
- M parity bytes (Reed-Solomon) are added to the end of each of the k_1 columns found in a block. This process adds M additional rows to the initial block;
- an additional ordering byte (a row number) is added as a header to each row, including those newly added parity rows (k_2+M rows in total);
- each row is encapsulated in an HDLC frame then transmitted.

This process is summarized in Figure 2.

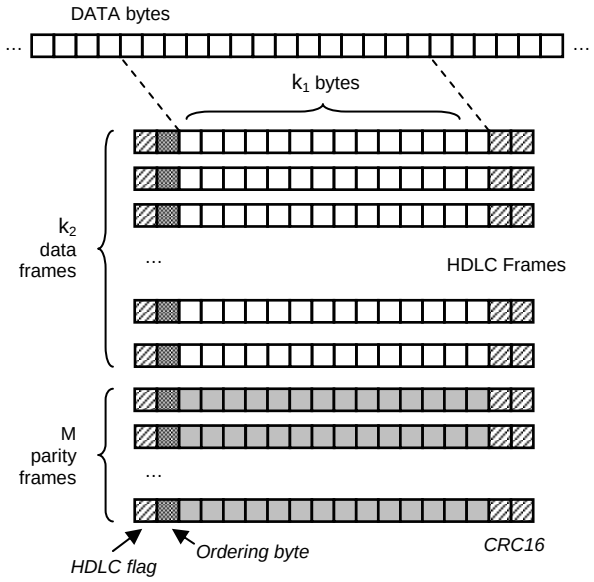

Figure 2 : Data partitioning

□ At the receiver side:

- k_1, k_2 and M are known;
- a given block is assumed to be correctly transmitted if no row is missing;
- if the number of rows missing is less or equal than M, the original block will be fully regenerated thanks to the Reed-Solomon algorithm (up to M missing bytes

[1] Quatech MPAP-100 (PCMCIA socket)

can be corrected since the positions of those bytes are precisely known thanks to the ordering bytes);
- if the number of rows missing is greater than M, the block will be considered as being lost.

As mentioned earlier, M equals 32. Values for k_1 and k_2 have been set with respect to the following compromises:
- the lower k_1, the higher the overhead caused by the additional ordering bytes and the CRC-16 within HDLC frame. On the other hand, when k_1 increases, the risk of losing a frame due to the presence of an erroneous byte increases as well;
- the lower k_2, the higher the protection of those k_2 frames against errors - a fixed number of parity frames are added whatever the value of k_2 - but the lower the user data rate.

Taking the ordering byte and the HDLC overhead (3 bytes) into account for k_1 and the 32 parity bytes for k_2, the user data rate D can be written as:

$$D_{EFEC1} = \frac{k_1 k_2}{(k_1+4)(k_2+32)} 16000$$

In order to reach an asymptotic user data rate of 9600 bit/s, k_2 has been set to 48. In the next sections, k_1 will be taken either equal to 20 or 40, ending up with user data rates equal to 8000 bit/s and 8727 bit/s respectively.

THEORETICAL PERFORMANCE

The theoretical performance related to EFEC1 can be easily computed in the following cases:
- when errors are uniformly distributed such as to each corrupt different HDLC frames: in the worst case, each error causes the loss of an entire frame (k_1 bytes);
- when errors are all contiguous to each other (i.e. occur in blocks) and corrupt as few frames as possible.

The fist case is representative of additive white Gaussian noise (AWGN) interfering with the data transmission. Blocks with 32 erroneous bits or less will be corrected. Error-free block decoding is guaranteed as long as the *Bit* Error Rate (BER) does not exceed the following threshold:

$$BER^*_{AWGN} = \frac{4}{(k_1+4)(k_2+32)}$$

The second scenario somehow relates to a transmitter operating in Frequency Hopping (FH) mode when given frequency bands are jammed. In such a case, blocks with up to $32(k_1+4)$ erroneous bytes will be corrected. Error-free decoding is guaranteed as long as the BER does not exceed:

$$BER^*_{JAMMING} = \frac{32}{(k_2+32)}$$

These two theoretical cases respectively provide the lower and upper bounds for the maximum correctable BER as particularized in the table below.

Table 1: Maximum correctable bit error rates

	BER^*_{AWGN}	$BER^*_{JAMMING}$
k_1=20, k_2=48	2.1E-3	0.4
k_1=40, k_2=48	1.1E-3	0.4

SIMULATION RESULTS

To complement those theoretical figures, simulations have been carried out under Matlab. AWGN has been used to test the first scenario and for the second one, errors have been introduced as 8-byte long random drops[2]. Simulation results are given in Figure 3 and were computed as an average over 100 blocs. This average explains why simulation results are actually (slightly) below the lowest theoretical limit in the case of the AWGN: although the *average* BER is below the limit, some blocks actually suffer from a local error rate higher than the average and get inevitably lost.

[2] 8 bytes do represent the amount of data transmitted during one frequency hop.

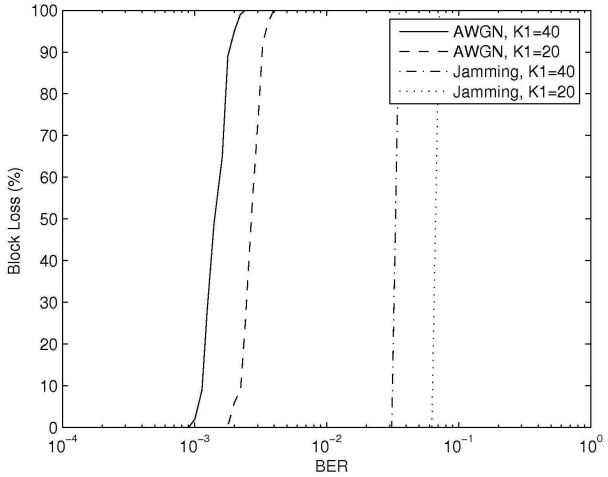

Figure 3: BAMS Performance under AWGN and Jamming Interferences - Simulation Results.

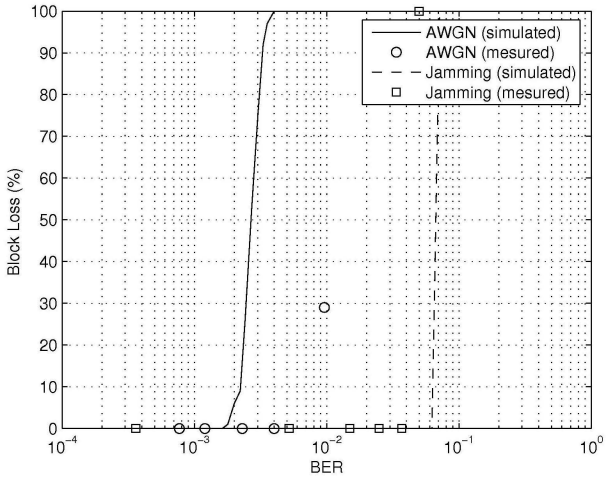

Figure 4: EFEC1 performance when k1=20. Actual measurements are depicted by discrete circles (AWGN) or squares (Jamming), earlier simulations by lines.

EXPERIMENTS

Experiments have been carried out in order to test the validity of our simulations. As an aerial transmission was difficult to setup in our (indoor) laboratory environment, two BAMS transceivers have been connected to each other by means of a coaxial cable coupled with a variable attenuator. By increasing the attenuation, one simulates the increase in distance between the two transceivers, resulting in a higher error rate. This setup offers a convenient way to test the efficiency of EFEC1 with respect to different noise levels interfering with the transmission. Furthermore, by setting up the two transceivers in frequency hopping mode and configuring the transmitter such as to use more or less frequency bands outside the receiver's active frequency range, jamming could be easily simulated. The results of these experiments are depicted in Figure 4 and Figure 5.

When k_1=20, EFEC1 is robust up to a BER of 2E-3 in case of AWGN or 3E-2 under jamming conditions.

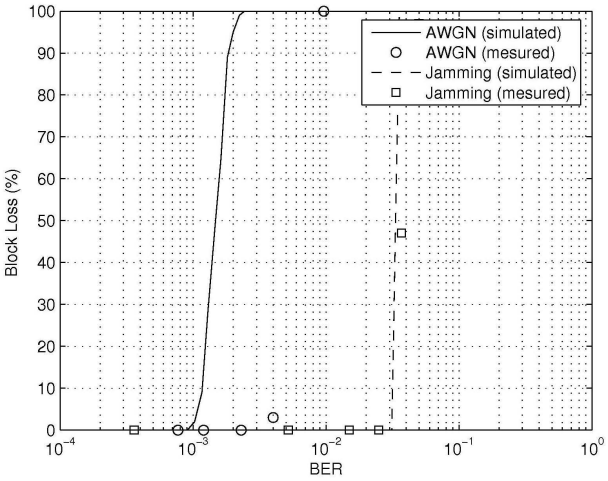

Figure 5: EFEC1 performance when k1=40. Actual measurements are depicted by discrete circles (AWGN) or squares (Jamming), earlier simulations by lines.

EFEC2

The performance of EFEC1 suffers from the fact that a single error causes the loss of an entire frame. Under jamming conditions, this drawback has little influence though: error bursts generally extend over the frame length, implying the loss of entire frames anyway. However, in the AWGN case, isolated errors are common, hence the poor error correction performance offered by EFEC1 in such a case.

Such an issue can be solved by implementing an error correction inside the HDLC frame itself, yet combined with the use of parity frames as described in EFEC1, leading to a concatenated coding strategy [2].

This coding strategy has been implemented in EFEC2 as follows:

□ At the emitter side:

- the data stream is divided into k_2-by-k_1-byte sized blocks;
- each row gets protected by 32 additional parity bytes, adding 32 columns to the block;
- each column, including the 32 additional ones, gets protected by 32 additional parity bytes, ending up in 32 additional rows of k_1+32-byte long;
- each of the k_2+32 rows is encapsulated in an HDLC frame, then transmitted.

□ The receiver proceeds to the decoding *iteratively*, as follows:

- correct each row impaired by less than 17 erroneous bytes (since the error positions are now unknown, the Reed-Solomon algorithm is only able to correct half the number of parity bytes);
- correct each column impaired by less than 17 erroneous bytes;
- successively apply the preceding steps until no other row or column can be corrected.

EFEC2 supersedes EFEC1 in two ways: first, EFEC2 is now able to correct errors within a frame; second, by correcting frames, EFEC2 is able to better correct the columns and vice versa. Such iterative process ends up in a performance that no single pass process can compete with. As a drawback however, EFEC2 has no clue anymore where the errors are located: while EFEC1 was able to correct up to 32 errors by adding 32 parity bytes, EFEC2 only corrects 16.

The effective data rate achieved by EFEC2 depends on the values of k_1 and k_2 and is given by:

$$D_{EFEC2} = \frac{k_1 k_2}{(k_1 + 33)(k_2 + 32)} 16000$$

and reaches 9600 bit/s when k_1=k_2=111.

The EFEC2 performance level, as simulated from Matlab, is depicted in Figure 6 for the AWGN case and for a user data rate of 9600 bit/s. For the sake of comparison, the same figure lists the performance achieved by former EFEC1 (dashed lines) when the bit rate was roughly similar (8727 bit/s, k_1=40). The increase in performance offered by EFEC2 is obvious, offering error-free transmissions up to BER=2E-2 instead of 1E-3.

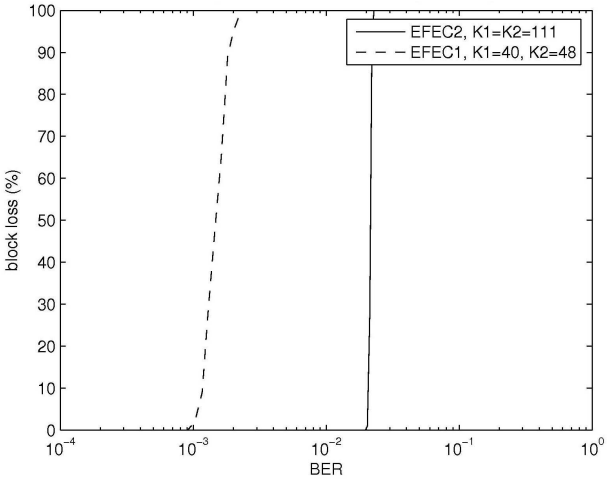

Figure 6 : E-FEC2 Performance under AWGN noise when k_1=k_2=111 (9600 bit/s)

Figure 7 refers to the performance achieved under jamming, i.e. when errors occur in 8-byte blocks. Under such a condition, EFEC2 performs better than EFEC1 again, increasing the robustness from BER=3E-2 to 7E-2

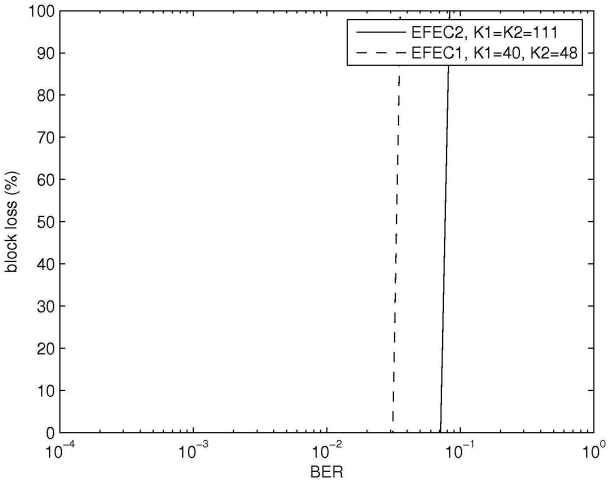

Figure 7: E-FEC2 Performance under Jamming noise when k_1=k_2=111 (9600 bit/s)

In our test case, jammed bytes did not extend over a frame period. When jamming corrupts more than k_1 bytes in a row, entire frames are lost. In such a situation, EFEC2 won't be able

to outperform EFEC1 anymore as EFEC2 is only able to recover 16 missing frames per block, while EFEC1 did 32.

EFEC2 clearly supersedes EFEC1 at the price of a higher decoding complexity. Furthermore, in order to achieve a 9600 bit/s user data rate, EFEC2 makes use of a 12kB block-size requiring a 10-second transmission time per block. This situation makes EFEC2 unpractical to use in the context of sending short text messages, but suitable for larger files like documents or images.

CONCLUSION

This paper highlighted the typical performance one could achieve using Reed-Solomon error correction codes in the context of speeding up data transmissions between BAMS transceivers above their nominal specification (2400 bit/s). Various coding schemes have been considered - all relying on shortened Reed-Solomon codes, either RS(80,48) or RS(143,111) depending on the codec in use - and their performances have been assessed against Gaussian noise and jamming interferences. In both cases, error-free transmissions up to 9600 bit/s were possible for bit error rates up to 2E-2 - or higher in case of jamming - assuming the selection of the right FEC scheme.

BIBLIOGRAPHY

[1] Bernard Sklar, *Digital Communications: Fundamentals and Applications*, 2nd Edition Prentice Hall, 2001.

[2] G. D. Forney, *Concatenated Codes*, M.I.T. Press, Cambridge, MA, 1966

BIOGRAPHY

Dr. Ir. **Stéphane Pigeon** was born in Brussels, Belgium, in December 1970 and got the degree of electrical engineer from the Université Catholique de Louvain in June 1994, with a specialization in Signal Processing. He worked at the Laboratoire de Télécommunications et Télédétection of this same university from January 1995 till December 1998. He first studied the relative merits of different scanning formats in the context of the future digital television (HAMLET European project) then worked in the field of multimodal biometric person authentication (M2VTS European project). He finalized a PhD thesis on this last topic in February 1999. In January 1999, he joined the Royal Military School as a researcher in the field of computer-assisted person identification and data fusion. His current interests are source and channel coding as well as speech processing.

Major **Bart Scheers**, Dr. Ir. was born in Rumst, Belgium, in November 1966 and got his degree of engineer, with a specialization in telecommunications, from the Royal Military Academy in 1991. After his studies he served as an officer in a territorial signal unit of the Belgian Army. In 1994 he was called back to the Royal Military Academy as an assistant in the field of signal processing. In 2001 he presented his PhD thesis on the use of ground penetrating radars in the field of humanitarian demining. From 2000 he works as a lecturer in the telecommunication department. His current domain of interest is networking.

Prof. Dr. Ir. **Patrick Verlinde** was born in Bruges (Belgium) in 1960 and he received his Master's degree in Engineering (Specializing in Telecommunications and Armament/Ballistics) from the Royal Military Academy in Brussels (Belgium) in 1983. In 1989 he received a Master's degree in Engineering (Specializing in Mechatronics) from the Katholieke Universiteit Leuven (Belgium), and in 1999 he obtained his PhD, on the use of decision fusion techniques in the field of the verification of the identity of a person, from the Ecole Nationale Supérieure de Télécommunications in Paris (France). Actually he is a full professor, heading the Telecommunication Chair at the CISS Department of the Royal Military Academy in Brussels, where he teaches courses on telecommunication systems and digital communications, while maintaining research activities in the field of information theory.