COMBINING A REED-SOLOMON BLOCK CODE WITH A BLIND EQUALIZER: SYNCHRONIZATION AND BIT ERROR RATE PERFORMANCE

Alexandre Skrzypczak, Grégory Blanc, and Tangi Le Bournault

ZODIAC DATA SYSTEMS,
2 rue de Caen - F - 14740 Bretteville l'Orgueilleuse - France

{alexandre.skrzypczak, gregory.blanc, tangi.lebournault}@zodiacaerospace.com
http://www.zodiacaerospace.com

ABSTRACT

The performance of telemetry systems may be strongly affected by diverse sources of perturbations. Among them, multipath channels and transmission noise are the most critical. While the effects due to the multipath channels can be attenuated thanks to equalization, the effects of the noise are limited if forward error correction is used. This paper first proves that the combination of blind equalization and forward error correction can strongly improve bit error rates. The other objective of the paper is to show that reasonably powerful codes like Reed-Solomon codes are sufficient to enable quasi-error free transmissions in a large majority of propagation channel scenarios.

Keywords: Reed-Solomon code, equalization, PCM/FM modulation, channel modelling.

INTRODUCTION

In flight testing, and more generally speaking in telemetry, the data link is needed to be kept available during an entire mission. However, a mission is the succession of different aircraft manoeuvres. If one considers an airplane for instance, it is first motionless on its parking position, then slowly moves on to its take-off position (taxiing), then takes off and finally flights. An efficient telemetry is able to guarantee an available data link for each manoeuvre.

However, in terms of physical analysis, each of the four aforementioned phases corresponds to a very different signal transmission scenario (in terms of transmission channel). In a parking context, the emitters and receivers are close to each other, leading to a high signal-to-noise ratio (i.e. the noise level is very low) but the transmitted signal may be subject to several permanent and high-levelled reflections on buildings or on the floor. The channel is seen as a frequency-varying multipath channel. For taxiing, the signal-to-noise becomes lower, the reflections are more attenuated but as the aircraft has a given speed, a Doppler shift and a Doppler spread may affect the transmission quality. In such a case, we get a slowly time-varying channel. For the take-off, the signal-to-noise still decreases, the channel still remains frequency-varying but as the aircraft speeds up, the channel becomes quickly time-varying. In a far flight context, the signal-to-noise ratio is very low, but the power of the reflected paths can be neglected and the aircraft appears motionless from the receiver point of view: the channel can be seen as a pure Gaussian channel.

Channel coding is generally used if we want to limit the amount of transmission errors due to the additive Gaussian white noise (AWGN). Different coding strategies can be considered and they are described here below. This solution is very efficient for the far-flight channel scenario. However, if the signal is also altered by a multipath channel, the coding performance may be significantly degraded. Indeed, a large majority of coding processes are designed by considering a Gaussian perturbation while the Inter-Symbol Interference that is generated by a multipath channel may follow a different statistical distribution.

Parallel to this, in the presence of noiseless multipath channels, an equalization process is very efficient in order to limit the number of transmission errors. However, as equalization never perfectly inverts the channel in practical cases, a residual distortion may still affect the data symbols, leading to residual transmission errors. Equalization is very efficient for parking scenarios where the noise level is negligible.

The target of this study is to experimentally find conditions to guarantee a good transmission quality in terms of synchronization and bit error rate for all the transmission scenarios that were previously described. We suppose that by coupling a channel coding with reasonable performance (here a Reed-Solomon code) with the equalizer that was developed by Zodiac Data Systems, the residual errors induced by the equalization process can be corrected by the channel coding, even for reasonably low signal-to-noise ratios i.e. we can expect quasi-error free transmissions for all the channel scenarios.

The paper is organized as follows: after a brief presentation of the Reed-Solomon code that is considered for the experimentations and of the ZDS equalizer, we then describe the experiment set up in a first time and the experiment results in a second time. We finally propose some requirements and draw our conclusions.

CHANNEL CODING AND EQUALIZATION

In this section, we first recall some generalities about channel coding. We then describe the channel code we considered for our experiments and we also briefly detail the blind equalizer that is developed by Zodiac Data Systems.

1. Channel Coding

In the literature, we can find different kinds of channel code. They are classically classified into three main families [1]: block codes, convolutional codes and iterative codes.

a. Block code

The block coding process consists in transforming a block (of bits or bytes) of size k into a block (of bits or bytes) of size $n > k$. Consequently, $n - k$ bits of redundancy are added at the end of the data block. The ratio $k/_n$ is named coding rate. They are usually decoded by the syndrome decoding algorithm or the Chase algorithm. Among the block code family, we can put the light on particular codes: cyclic codes with the longest minimal Hamming distance, also called Reed-Solomon codes. Their particularity lies on the fact that they were designed to ensure an optimal capability of error correction into a block code. And as they are cyclic codes, the decoding algorithms are based on polynomial calculation, which guarantees a reasonably low complexity.

b. Convolutional code

Convolutional codes are another well-known family of codes. They are based on the following idea: the $n - k$ redundancy bits are processed by a binary operation on a given set of bits. These bits are obtained from a sliding window of size k on the original data stream. The decoding of this convolutional code is made thanks to the well-known Viterbi algorithm. Its principle consists in finding the most probable path in the code treillis by minimizing the Hamming distance between the received binary sequence and the possible candidate sequences.

c. Iterative code

The last class of forward error coding is the iterative codes like the Turbo-Codes and the LDPC codes. Although the encoders for theses codes are very easy to implement (two recursive and systematic convolutional codes with an interleaving for the Turbo-Codes and a block code with a parity check matrix with a few number of zeros for the LDPC), the efficiency of this kind of codes lies in the iterative decoding. For Turbo-Codes, the BCJR algorithm calculates extrinsic information that iteratively feeds another BCJR decoder in order to tune the probability to decode a zero or a one in a finer way. LDPC decoding is made thanks to the sum-product algorithm whose principle is to iteratively send messages in the Tanner's graph of the code in order to get a precise evaluation of the probability to decode a zero or a one.

This class of code is known to get quasi-optimal performance with respect to the Shannon's limit. They are also now proposed in the latest release of the IRIG 106 standard [2] and, more generally speaking, in a large number of telecommunication standards.

d. Code choice

In this experiment, we decided to use a Reed-Solomon code (the one of the CCSDS) as we want to prove that in a large majority of channel scenarios, a reasonably performing channel coding is sufficient to get a quasi-error free transmission when combined with an equalizer.

2. Reed-Solomon code from CCSDS

In the following, we use the Reed-Solomon code that is proposed in the CCSDS standard. It is fully described in [3]. The basic features are the following ones:

- $k = 223$ bytes and $n = 255$ bytes. This is a systematic code.
- The raw code rate is then $r = {}^{k}/_{n} \approx 0.87$.
- This code is theoretically is able to correct $t = (n - k)/2 = 16$ false bytes into a given block.

- In the CCSDS, it is possible to introduce an interleaving of size I (with $I \in \{1, 2, 3, 4, 5, 8\}$) consisting in grouping I data blocks of size k in some way to compute the code redundancy.
- The beginning of the code block is detected by a binary sequence called ASM (Attached Sync Marker). Its good detection/synchronization is a prerequisite for a good decoding process. The synchronization on this ASM is also studied in this paper.
- In the following experiment, the 4-bytes ASM is the one of the CCSDS i.e.

  Ox 1A CF FC 1D.
- The data frame after coding is finally represented in Fig. 1.

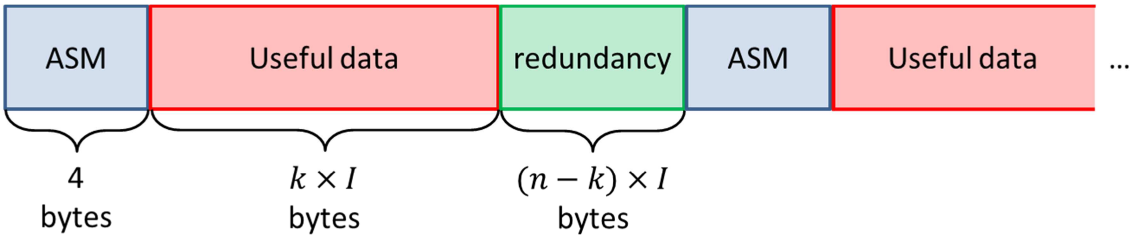


Figure 1: organization of the RS frame of the CCSDS standard

3. ZDS Equalizer

Zodiac Data Systems recently propose an equalizer as an option of its RTR (Radio Telemetry Receiver). The basic features are the following ones:

- This is a multi-criterion (including a basic CMA) and an iterative algorithm.
- For the moment, it only works with the classical PCM/FM modulation that is used for in instance in the IRIG106 standard.
- This is an adaptive algorithm i.e. it tracks the channel variations. In other words, it is able to correct the degradations brought by time-varying and frequency-varying channels.
- It is well-adapted for low bit rates (around 1 Mbps) but its performance does not degrade for bit rates up to 4 Mbps.

Additional information is given in [4] so as its theoretical performance. It is then shown that, in very different channel scenarios, this iterative equalizer offers a considerably better data link availability when compared to standard demodulation or to a CMA algorithm.

EXPERIMENT SETUP

In this section, we describe the experimental testbench we used, so as the equipment and their configurations. We also propose the channel models we considered.

1. Testbench

In order to fairly compare the gain that is brought by the RS code, the equalizer and the association of both, we choose to record PCM/FM signals that were distorted by different channels and noise power. This signal is then demodulated by 4 different methods: equalizer on/off and RS decoder on/off. Doing so, t the different demodulation processes run over the same noisy and distorted channel so that we can affirm that the different performance gain we might observe is independent of the distortion statistics.

To simulate a signal transmission that is close to real on-field situations, we first used a SMBV 100A from Rohde and Schwarz to simulate data transmission with a PCM/FM modulation. The useful data is a binary counter from 0 to 255 that loops back to 0 after it reaches the value 255. The redundancy is then derived from the coding techniques described in [CCSDS]. In order to study the influence of the coding interleaving, we chose 2 different interleaving depths I=1 and I=5. After ASM insertion, 2 final bit streams (for I=1 and I=5) are then loaded in the SMBV. This data stream is then applied as a PCM/FM modulation with an index 0.7. We also set 2 different bit rates: 1 Mbps and 4 Mbps.

We used an AMU 200 from Rohde and Schwarz as channel simulator and noise generator. The signal that is at the SMBV baseband output is then distorted by static or dynamic multipath channels and noise is also added by the AMU. After that, the resulting signal is re-injected into the SMBV for a RF transposition at 70 MHz. The resulting signal is then recorded and stored by a RSR (Radio Signal Recorder), which is a signal recorder developed by ZDS. The testbench that is used for signal generation is then shown in Fig. 2(a).

(a) testbench for signal generation and recording (b) testbench for estimation of BER and synchronization performance

Figure 2: used testbenches

The RSR is also able to replay a recorded signal. The replayed signal then feeds a RTR that demodulates the signal (with/without RS decoder – with/without equalization) and recovers the received binary frames. These binary frames are stored in files and these files are finally post-processed by Matlab in order to estimate ASM synchronization performance and bit error rates. This testbench is displayed on Fig. 2(b).

2. Synchronization and BER

The synchronization of the bit stream is made by a Frame Sync. Frames are identified thanks to a sync pattern and the Frame Sync parameters are the following ones:

PARAMETER	DEFINITION	VALUE
Bit Slip	Size variation of the search window where the sync word is expected	0
Sync threshold	Number of errors allowed in the sync word	3
Check to Lock	Number of consecutive frames correctly synchronized to be declared ***locked***	0
Lock to Search	Number of consecutive frames where the sync word is not found before restarting the search of the sync word	0

With the transmissions parameters (data rate, number of bits in the frame …), the number of transmitted frames can be easily estimated. The RTR can also give us the number of binary frames that are processed. As a consequence, the percentage of lost frames can be easily derived. The BER is then calculated on the frames that were correctly synchronized (the lost frames are not considered at all). In the following, we consider that we are in a quasi-error free transmission if BER<10^{-6}.

3. Channel models

a. Parking channel

As previously described in the introduction, transmitters and receivers are close to each other. Consequently, we can expect a high E_b/N_0 but the reception can be degraded by several high-powered signal reflexions. In addition, the transmitter is supposed to be motionless, the channel can consequently be considered as static. We then modelize this channel by the model proposed by M. Rice in [5], called AFB2 channel. Even if AFB2 was not proposed to represent such a transmission scenario, it fits with a lot of channel soundings we made in a parking context. All paths have a constant phase and no Doppler as the transmitter is motionless. The characteristics of the paths are given in the following table.

Path number	**Relative power (in dB)**	**Delay (in µs)**	**Doppler (in Hz)**	**Path type**
1	-16	0	0	Constant phase
2	0	0.05	0	Constant phase
3	-9	0.1	0	Constant phase
4	-9	0.49	0	Constant phase
5	-9	0.73	0	Constant phase
6	0	0.78	0	Constant phase
7	-16	0.87	0	Constant phase
8	-15	0.92	0	Constant phase

b. Taxiing channel

In a taxiing channel, the contribution of reflected paths is attenuated by the fact that the distance between transmitters and receivers increases. However, as the aircraft moves, the channel becomes time and frequency varying. We made some channel soundings at the Airbus' airport at Toulouse-Blagnac [6] and from these experiments, we have modelled a taxiing scenario that might be problematic for classical FM demodulation. The taxiing model is described in the following table. The main path follows a Rice distribution while the reflected ones follow a Rayleigh distribution. The channel is simulated for a transmission over the S-Band, which means that the 100 Hz Doppler corresponds to a transmitter speed equal to 50 km/h.

Path number	**Relative power (in dB)**	**Delay (in µs)**	**Doppler (in Hz)**	**Path type**
1	0	0	100	Rice
2	-8	2.5	100	Rayleigh
3	-27	8	100	Rayleigh

c. Take-Off channel

From the channel soundings at Airbus [6], we have derived some models for the take-off case. It can be found that the reflected paths are low-powered but have a wide Doppler spectrum. This scenario may be very problematical for multicarrier modulation like COFDM as a wide Doppler spectrum may generate important intercarrier interference. For monocarrier modulations like PCM/FM, the effect of Doppler spread is very limited. In addition, the power of the reflected paths is low enough so that the classical FM demodulation is not affected and may guarantee a quasi-error free transmission if the noise level is sufficiently low, which has been confirmed by Matlab simulations, not displayed here. As a consequence, no experiment on this channel has been proposed.

d. Far Flight channel

In such a context, we suppose that the aircraft is far enough so that the reflected paths are very negligible. However, as the distance between transmitter and receiver is important, the value of E_b/N_0 is very low. To simulate this channel, we use the AMU 200 as a noise generator by switching off the multipath fading.

e. Benchmark channel

In order to characterize the performance of our equalizer, we proposed a synthetized channel for which the FM demodulation outputs a lot of errors and for which the equalizer offers a quasi-error free transmission. In the frequency point of view, this channel is seen as a deep fading in the signal bandwidth and this fading periodically crosses the signal bandwidth in a short period of time. As a result, this signal is highly frequency selective (because of the deep fading) and highly time selective (because the channel characteristics quickly vary). This channel is modelled as described in the following table.

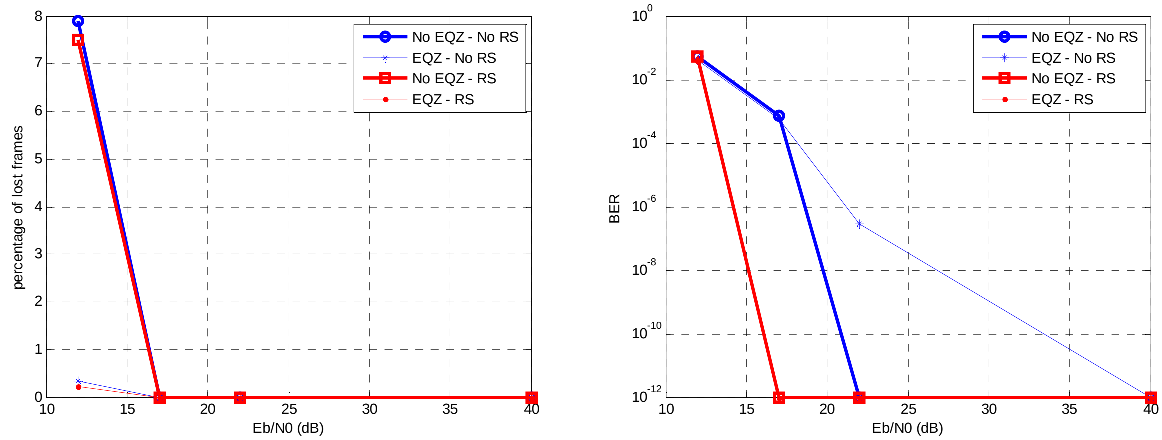
Path number	**Relative power (in dB)**	**Delay (in µs)**	**Doppler (in Hz)**	**Path type**
1	0	0	0	Static
2	-1.5	2	30	Constant phase

EXPERIMENTAL RESULTS

In this section, we show the results of our lab experiments using the testbenches displayed on Fig. 2 and 3. We chose to give the results only for a bit rate equal to *1 Mbps* and for *I=5* as we did not notice any major differences (except if mentioned) in the performance with a data rate of 4 Mbps and with I=1.

1. Far Flight scenario

The results of our experiments are given in Fig. 3.



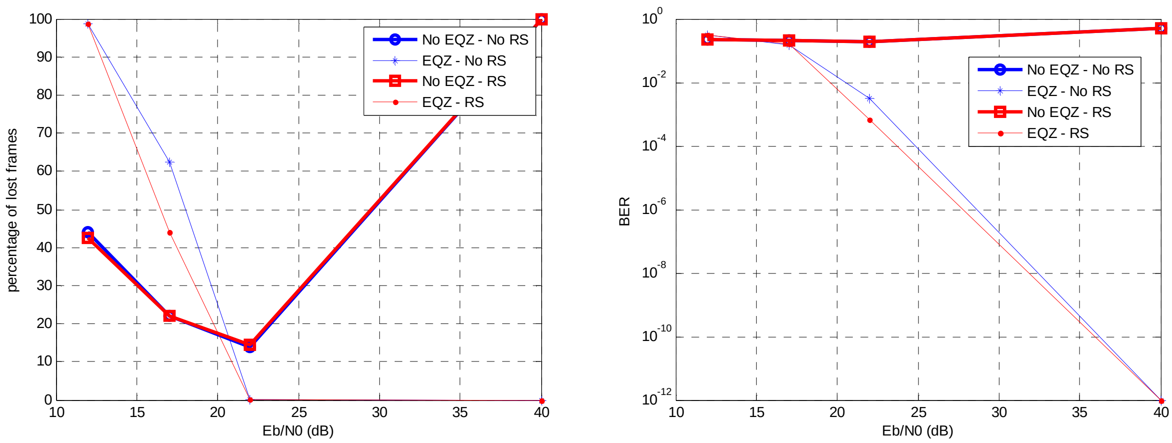
(a) Percentage of lost frames *(b) BER*

Figure 3: experiment results for the Far Flight channel scenario

These results show that equalization does not improve at all the results in a Gaussian scenario, as expected. This can be easily explained as the equalizer is designed to invert the convolutive effect of the channel and is not able to correct the additive noise which is the only source of degradation in a Gaussian channel. However, RS decoding strongly improves the BER, as expected. For low E_b/N_0 values, we observe a strong improvement of the synchronization performance (represented by the percentage of lost frames) by using blind equalization, i.e. we can retrieve more frames. For further investigations, it would be interesting to develop a way to detect the presence or absence of multipath in order to activate the equalizer when it is useful.

2. Parking scenario

The results of our experiments are given in Fig. 4.



(a) Percentage of lost frames *(b) BER*

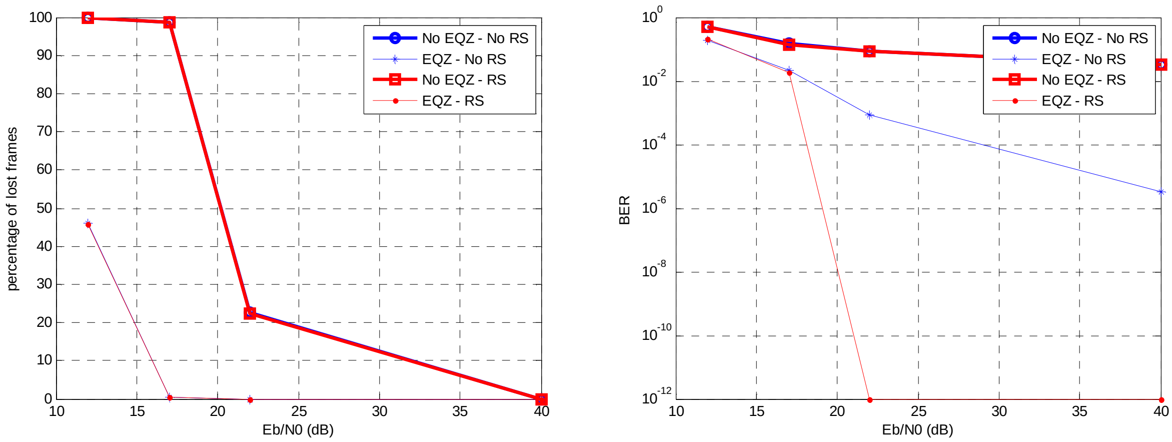
Figure 4: experiment results for the Parking channel scenario

In this scenario, we observe that without equalization, it is very difficult to lock the synchronization process. This can be explained by the fact that if the sync word is altered for one frame, it remains true for all the frames as it is a static channel. We also show that equalization drastically improves the synchronization and consequently the BER. We observe that in terms of BER, blind equalization allows a quasi-error free transmission for $E_b/N_0 >$

27 dB. However, we need to add more points in the graph to affirm that RS decoding allows the correction of the residual errors after equalization. It can be also observed that RS decoding without equalizer does not improve the BER results at all.

3. Benchmark scenario

The results of our experiments are given in Fig. 5.



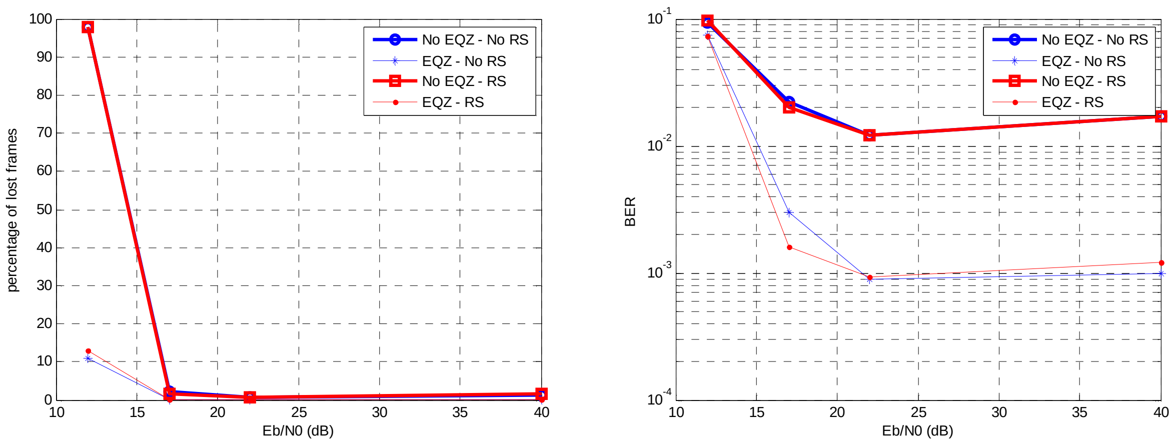
(a) Percentage of lost frames *(b) BER*

Figure 5: experiment results for the Benchmark channel scenario

The association of blind equalization and RS decoding allows quasi-error free transmissions for $E_b/N_0 > 20$ dB. We here confirm that equalization lowers the error floor and RS decoding is able to correct residual errors after equalization. As the channel is dynamic in this case, it proves the ability of the algorithm to well track the quick channel variations, even for reasonably high values of noise power. Note again that RS decoding without equalization is totally inefficient. Equalization also brings a great reduction of lost frames.

4. Taxiing scenario

The results of our experiments are given in Fig. 6.



(a) Percentage of lost frames *(b) BER*

Figure 6: experiment results for the Taxiing channel scenario

This experiment shows that the error floor for EQZ+RS takes place between 10^{-3} and 10^{-4} with no real gain brought by the RS decoding. This proves that the channel is basically hard to perfectly equalize. The equalization error is too high-powered so that the channel decoding may correct it. Using higher data rate also seems to degrade the performance as equalization algorithm is limited by the high computational resources needed.

In such a case, a solution is to increase the number of iteration in the equalization algorithm in order to lower the power of the equalization error or to use a more efficient channel code. We can guess that a concatenated code (RS + convolutional code with an interleaving between them) is sufficient. If not, a LDPC code could be able to reach a quasi-error free performance.

CONCLUSIONS

From this study, we can extract interesting information about equalization and channel decoding:

- Equalization always improves the synchronization process, i.e. we can retrieve more frames by using a blind equalizer,
- In the presence of a multipath channel, RS coding alone does not improve the BER performance at all,
- In the same context, blind equalization alone always improves the BER,
- For a large set of channel scenarios, the combination of blind equalization and RS decoding allow quasi error free transmissions for reasonable values of E_b/N_0,
- For some complex channel models, the residual distortion after blind equalization is too important so that the RS algorithm could correctly decode the information. More powerful codes would improve the system performance if they suit the residual distortion.

REFERENCES

[1] J. Proakis, Digital Communications. McGraw-Hill. 2001.

[2] IRIG 106, Transmitter and Receiver Systems, 2015.

[3] CCSDS 101.0-B-6, Telemetry Channel Coding, October 2002.

[4] A. Gueguen and D. Auvray, *“Multipath Mitigation on an Operational Telemetry Link”*, in ITC 2011, October 2011.

[5] M. Rice, *“Multipath Modeling and Mitigation Using Multiple Antennas (M4A)”*, Technical report, 2014.

[6] A. Skrzypczak, A. Thomas, and G. Duponchel, *“Paradigms Optimization for a C-Band COFDM Telemetry with High Bit Efficiency”*, in ITC 2013, October 2013.