POLITECNICO DI TORINO
Repository ISTITUZIONALE

Theoretical Analysis and Implementation of Effective Receivers for Telecommand Space Links

Original
Theoretical Analysis and Implementation of Effective Receivers for Telecommand Space Links / Baldi, M.; Prata, R.; Simone, L.; Urrutia, C.; Bertinelli, M.; Chiaraluce, F.; Silva, P. F. D.; Garello, R.; Maturo, N.; Navarro, M.; Palomo, J. M.; Paolini, E.. - ELETTRONICO. - (2019), pp. 1-8. (Intervento presentato al convegno 8th ESA International Workshop on Tracking, Telemetry and Command Systems for Space Applications, TTC 2019 tenutosi a Darmstadt, Germany nel 2019) [10.1109/TTC.2019.8895426].

Availability:
This version is available at: 11583/2836383 since: 2020-09-27T17:38:05Z

Publisher:
Institute of Electrical and Electronics Engineers Inc.

Published
DOI:10.1109/TTC.2019.8895426

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

12 October 2024

Theoretical Analysis and Implementation of Effective Receivers for Telecommand Space Links

Marco Baldi[1], Massimo Bertinelli[2], Franco Chiaraluce[1], Pedro Freire da Silva[3], Roberto Garello[4], Nicola Maturo[5], Monica Navarro[6], José Maria Palomo[7], Enrico Paolini[8], Ricardo Prata[3], Lorenzo Simone[9], Cristian Urrutia[10]

[1]*Università Politecnica delle Marche* and *CNIT*, Ancona, Italy – email: {m.baldi, f.chiaraluce}@univpm.it
[2]*ESA/ESTEC*, Noordwijk, The Netherlands – email: massimo.bertinelli@esa.int
[3]*Deimos Engenharia*, Lisbon, Portugal – email: {pedro.silva, ricardo.prata}@deimos.com.pt
[4]*Politecnico di Torino* and *CNIT*, Torino, Italy – email: roberto.garello@polito.it
[5]*University of Luxembourg*, Luxembourg – email: nicola.maturo@uni.lu
[6]*Centre Tecnològic de Telecomunicacions de Catalunya*, Barcelona, Spain – email: mnavarro@cttc.es
[7]*Deimos Space S.L.U.*, Madrid, Spain – email: jose-maria.palomo@deimos-space.com
[8]*Università di Bologna* and *CNIT*, Bologna, Italy – email: e.paolini@unibo.it
[9]*Thales Alenia Space*, Rome, Italy – email: Lorenzo.Simone@thalesaleniaspace.com
[10]*Deimos Space UK*, Harwell, United Kingdom – email: cristian.urrutia@deimos-space.com

***Abstract*—This paper deals with the feasibility of the new receiver scheme for telecommand space links, based on the adoption of short low-density parity-check codes recently introduced in the standard. Being able to reduce significantly the required signal-to-noise ratio, these codes have an impact on the acquisition, tracking and synchronization issues. All these aspects have been faced, both theoretically and practically through the realization of a breadboard that implements the core elements of the on-board receiver, and is able to demodulate and decode uplink signals employing the new codes. The breadboard incorporates innovative solutions to cope with the acquisition and synchronization problems, and a pioneering implementation of non-iterative decoders based on the most reliable basis algorithm.**

Keywords—frame synchronization, hybrid decoder, low-density parity-check codes, most reliable basis, signal acquisition, space links, telecommand, tracking.

I. Introduction

The introduction of new coding techniques able to improve the performance of TeleCommand (TC) in terms of supported data rate and maximum distance are key enabling factors for next generation Near Earth (NE) and Deep Space (DS) missions, as they allow maximization of TC data volume in point to point communication links between Earth stations and spacecrafts.

Space TCs must guarantee ultra reliability in conveying control information as well as software patches from Earth control centers to scientific payload instruments and engineering equipment On-Board (O/B) spacecraft. The success of a mission may be compromised because of an error corrupting a TC message: a detected error causing no execution or, as a potentially catastrophic event, an undetected error causing a wrong execution. This imposes strict constraints on the maximum acceptable detected and undetected error rates.

The European Space Agency (ESA) has funded a project titled Next Generation Uplink Coding Techniques (NEXCODE), aimed at research, design, development, and demonstration of a TC receiver chain for scientific missions, encompassing new Low-Density Parity-Check (LDPC) codes recently included in the Consultative Committee for Space Data Systems (CCSDS) recommendation [1]. The following outcomes have been reached at the end of the project:

- Study, evaluate and optimize advanced co/decoding techniques to improve significantly the uplink performance (in terms of data-rate and/or maximum distances) of NE and DS science missions, compared to the currently used code, targeting 64 ksps and 7.8125 sps for NE and DS, respectively.
- Evaluate and clarify the impact of the new LDPC codes in terms of: i) Required protocol modifications; ii) O/B receiver algorithms (acquisition and tracking of uplink signals at lower Signal-to-Noise Ratio (SNR), determined by higher coding gains); iii) O/B receiver architecture (to cope with extra complexity and new algorithms).
- Prototype the O/B receiver chain core elements, including the decoder for the advanced coding schemes, by means of Commercial Off-The-Shelf HardWare (COTS HW), such as Field Programmable Gate Array (FPGA) platforms, to help validating the approach and minimize the risk of adoption, bringing the technology readiness level 4.
- Evaluate the most relevant metrics, including the effective coding gains, and the performance/complexity trade-off.

The project featured an explicit demonstration of the feasibility of the new receivers, paving the way for a more effective management of TC links in future missions. For it, several testing tools were developed allowing to emulate the TC link at Intermediate Frequency (I/F) and digital level, as well as to generate performance statistics with the receiver results.

All the most critical blocks were designed, evaluated and implemented. Among them:

- Carrier acquisition and tracking: shown to be very critical

Study funded by European Space Agency under the contract 4000111690/14/NL/FE.

due to very low SNR values; Fast Fourier Transform (FFT)-based carrier acquisition techniques have been analyzed and designed (no longer need of on-ground carrier sweeping).

- Frame synchronization: the advantage related to longer start and tail sequences (the latter being optional for the shortest LDPC code) was demonstrated. Low-complexity, but near optimal, methods for pattern recognition have been investigated and implemented.
- LDPC decoders: The error rate performance of the best decoding algorithms for the new LDPC codes were properly addressed for both iterative (Sum Product, Min Sum (MS), Normalized Min Sum (NMS)) and non-iterative (Most Reliable Basis (MRB)); impact of quantization was assessed and residual Undetected Error Rate (UER) was evaluated; NMS implementation was optimized and, very remarkably, the first hardware realization of a hybrid (NMS + MRB) decoder was developed and validated in a System-on-Chip (SoC) proof-of-concept, with SoftWare + Digital Signal Processor (SW + DSP) implementation prototyping; the gain achieved by the hybrid decoder with respect to conventional iterative-only LDPC decoders was validated; limitations on maximum bit-rate achievable by hybrid decoder for DS and NMS for both scenarios were investigated.

A first series of results from the NEXCODE project has been presented in [2], [3]. In this paper we deepen the subject, with special emphasis on the implementation issues.

The organization of the paper is as follows. In Section II we introduce basic notation, limited to the aspects of interest for the present paper. In Section III we describe the breadboard design and the main steps for the development of the most critical elements of the study. In Section IV we present some results on the error rate performance by comparing theoretical results with measured ones through the breadboard. Finally, some conclusions are drawn in Section V.

II. Definitions and Notation

The NEXCODE project is focused on the TC receiver functionalities. Critical issues concern decoding, but also acquisition and tracking, and frame synchronization.

A. Decoding

Two binary LDPC(n, k) codes, where n is the length and k the dimension, have been recently included in the CCSDS recommendation for TC synchronization and channel coding [1]. Both they are characterized by coding rate $R_c = 1/2$, the shortest one having $k = 64$ and the longest one $k = 256$. The codes are systematic and are defined through their parity check matrices **H**. These are composed by $Q \times Q$ submatrices where $Q = k/4 = n/8$ (so, $Q = 16$ for the LDPC(128, 64) code and $Q = 64$ for the LDPC(512, 256) code). Matrix **H** is related to the generator matrix **G**, which is another way to specify the code and is used in the MRB algorithm.

The performance of the LDPC codes depends on the decoding algorithm adopted (the standards do not specify the decoding algorithms, whose choice is left to the Agencies). In the NEXCODE project we have focused on three different algorithm families:

- Iterative decoding algorithms. These are classical LDPC soft-decision iterative decoding algorithms, like the Sum-Product Algorithm (SPA), typically implemented using log-likelihood ratios (SPA-LLR), or its simplified versions, e.g., MS or NMS.
- Non-iterative MRB decoding algorithms. They exploit a soft-decision procedure, potentially able to achieve performance very close to that of the optimum Maximum Likelihood (ML) decoder. Their main drawback is complexity, which is an important issue in TC links, where decoding is performed O/B.
- Hybrid decoding algorithms. These decoders perform first a low complexity decoding attempt through an iterative algorithm and invoke MRB only when the iterative algorithm is not able to find any codeword (detected error).

The application of the non-iterative MRB algorithm (alone or inside the hybrid decoding) to LDPC codes has been recently investigated from a theoretical point of view [4], [5]. To the best of authors' knowledge, however, no practical implementation has been reported, until now, in the literature. So, the NEXCODE project gives very innovative elements in this sense.

In its basic version, MRB consists of the following steps:

1. Find the k most reliable received bits and collect them in a vector $\mathbf{v}^*$.
2. Perform Gauss-Jordan elimination on matrix **G**, with respect to the positions of the k bits, in such a way as to obtain a systematic generator matrix $\mathbf{G}^*$ corresponding to such bits. It is possible to verify that, if **G** is full rank, the same property holds for $\mathbf{G}^*$, too.
3. Encode $\mathbf{v}^*$ by $\mathbf{G}^*$ to obtain a candidate codeword $\mathbf{c}^* = \mathbf{v}^*\mathbf{G}^*$.
4. Choose the order i of the algorithm (we denote by MRB(i) an instance of the algorithm with order i).
5. Consider all (or an appropriate subset of) Test Error Patterns (TEPs) of length k and Hamming weight $w \le i$.
6. For each of them: add it to $\mathbf{v}^*$, encode by $\mathbf{G}^*$, verify if the Euclidean distance from the received vector is smaller than that of the previous candidate codeword and, if this is true, update the candidate.

At the end of the process, the algorithm always provides a codeword, which is the codeword at minimum distance from the received vector within the considered set. Clearly, if $i = k$ the set of all possible codewords is considered and the algorithm turns to be equivalent to the exact ML soft-decision decoding. Unfortunately, this requires considering $2^i = 2^k$ vectors, then it is unfeasible if the code is, relatively, long. A key item for the algorithm complexity is the chosen order i. In fact, the maximum number of TEPs to be tested, for each received word to be decoded, is equal to $N_{\mathrm{TEP}}^{\max} = \sum_{j=0}^{i} \binom{k}{j}$. Starting from this basic version, some techniques which allow reducing the algorithm complexity and properly generating the test error patterns are described in [4].

As mentioned above, an alternative consists in using MRB within a hybrid algorithm [6]. More details on this approach are given in Section III.

The performance of the decoding algorithms is measured by the SNR value required to achieve a target Codeword Error Rate (CER). For TC, the reference value is CER = 10^{-5}. The

performance achieved through both numerical simulations and practical tests on the breadboard is presented in Section IV.

B. Acquisition and Tracking

Since the adoption of more powerful codes potentially allows to operate at the very small SNRs, signal acquisition and tracking may prove extremely challenging.

Identification of receiver processing bottlenecks considers DS and NE missions. Whereas the former is representative of very low SNR scenarios, the latter is representative of high data-rates scenarios. In both cases, residual carrier modulation is typically used for the TC link. For NE missions, the transponder is configured to receive a Manchester encoded data phase modulated onto the carrier modulation, a scheme referred to as Pulse Code Modulation Shift Phase-Level (PCM SP-L). On the other hand, in DS missions the transponder is configured to receive Non-Return-to-Zero (NRZ) data modulated on a sinusoidal-wave subcarrier that is then phase modulated onto the carrier (PCM NRZ-L modulation).

From a receiver perspective, the architecture is similar for both scenarios, with the main difference that PCM SP-L in NE does not require subcarrier tracking. In fact, carrier acquisition is common to both scenarios since it is performed under carrier modulation mode CMM1, which only contains the unmodulated carrier. As reported in [2], [3] and [10], replacing the current carrier sweeping scheme and the standard Phase Locked Loop (PLL) at the receiver with spectral estimation techniques overcomes the bottleneck for low symbol rate at the target $E_s/N_0 \approx 2$ dB, where E_s is the symbol energy and N_0 the one-side spectral density of the thermal noise (operative SNR reached with the new LDPC codes at CER = 10^{-5}), and can also relax the requirements on the carrier loop enforced by the on-ground sweeping procedure. Such an approach has several advantages: i) FFT-based carrier acquisition does not require on-ground sweeping, ii) allows faster acquisition, and iii) can reach lower estimation errors on carrier frequency.

However, despite the improvements introduced by FFT-based carrier acquisition, the carrier and subcarrier tracking persist as a bottleneck for some of the selected scenarios. A detailed list of the performance limits for the most promising techniques addressed in the project is reported in [2], [3] and [10]. To summarize, from the evaluated enhanced approaches, aimed at reducing the loop bandwidth, FFT-aided Costas subcarrier acquisition and tracking provides gains in the low SNR region, yet they do not meet the target E_s/N_0 at the lowest symbol rate scenarios. Allowing to modify the modulation index together with a side-band aiding carrier tracking can further stress the performance limits for DS scenarios at the lowest possible symbol rate of 7.8125 sps but still could not manage to reach the target $E_s/N_0 \approx 2$ dB. Advanced techniques which imply a higher complexity at the transponder side, such as Frequency Locked Loop (FLL)-assisted and (Kalman Filter) KF-based carrier tracking schemes did not cope either with the performance requirements at the lower SNR regime. Alternative solutions with implications to the transmission scheme were also explored but required implementing a coherent carrier and subcarrier generation.

As for symbol tracking, it is currently implemented by a 2nd order Data Transition Tracking Loop (DTTL). After careful adjustment, this technique is able to operate at the target SNR values, provided that carrier/subcarrier tracking is successful.

In summary, for NE scenarios the target operation point does not compromise correct baseline receiver behavior. On the other hand, DS scenarios with very low data-rates constitute the main challenges since the effective SNR at the input of the receiver is extremely low.

C. Frame Synchronization

Commands from Earth to space are sent in the payload of variable length TC Transfer Frames (TFs). A TF is divided into blocks and individually encoded. The codewords are then encapsulated between a start sequence and (optionally) a tail sequence to obtain a Communication Link Transmission Unit (CLTU). The role of the start sequence is to enable frame synchronization, so that the decoding process can be initiated. On the contrary, the goal of the tail sequence, when present, is to allow the decoder for implicitly detecting the end of the CLTU. Preceding the start sequence there is an acquisition sequence of alternating '0' and '1' (starting with either a '0' or a '1') of variable length that serves the purpose of signal acquisition and aids in locking timing synchronization loop. Idle sequences can also be found between CLTUs to help maintaining symbol synchronization.

Introduction of the new LDPC codes has implied lengthening of the start sequence, that is passed from 16 bits to 64 bits, in order to compensate the impact of the potential operation at a reduced SNR [2], [3]. The tail sequence is also affected by the new codes, and has been lengthened as well, passing from 64 bits to 128 bits (Fig. 1).

Fig. 1. CLTU structure for LDPC(128, 64) encoding

Frame synchronization takes place after symbol demodulation and consists in determining the correct position of the start sequence. Since frame synchronization has to be achieved before reception of the entire CLTU, whose length is variable and unknown a priori, the baseline frame synchronizer implements one-shot frame synchronization. That is, it compares, for each position of the observation window, the computed metric to a pre-defined threshold in order to decide if the current position corresponds or not to the start of the CLTU. The metric measures the similarity of the received symbols in the observation window, which has the same length as the start sequence, with the known start sequence. Such metric is based on standard soft and hard correlation, which results in a receiver bottleneck for the target SNR and the start sequence length. The proposed Simplified Likelihood Ratio Test (S-LRT) metric introduced in [2], [3], has been implemented in the breadboard and its performance contrasted with previous simulation results. This near-optimum metric accounts for the sign ambiguity inherent in the TC binary modulation scheme, which as a by-product, and provides the sign of the received symbols required for decoding without the need for differential modulation. Details of the implementation choice and performance are given in Section III.

After detecting the CLTU start, the decoder processes each block of n bits. To recognize the CLTU end, the receiver exploits the tail sequence. In principle, the "uncorrectable pattern" approach may be used: when a block is marked as incorrect, the receiver declares the end of the CLTU. Actually, if the code is much more powerful, as the new LDPC codes

are, it is difficult to find an uncorrectable pattern. Moreover, the approach fails for complete decoders (like those based on the MRB algorithm), which always return a codeword. The "natural" approach for CLTU termination is then the application of a detector, which looks for the tail sequence after each codeword.

Like for the start sequence, the optimal Likelihood Ratio Test (LRT) is characterized by a quite high complexity, which makes its application to high data-rate implementation difficult. Soft and hard correlations are then often used, although they are highly sub-optimal. In our study, we have focused on the simplified Massey detector which, given the pattern symbols and the received symbols, computes their correlation only where the symbol signs are different. The simplified Massey detector provides an excellent solution, with limited complexity and very good performance.

Moreover, as mentioned, the CCSDS has recently approved the use of a new 128-symbol tail sequence when the short LDPC(128, 64) code is used. One of the reasons is the solution of a peculiar problem: the data in a TF might happen to match the 64-symbol tail sequence, that is, the length used in the past, or approximately match it. The tail sequence, instead, is not present when using the LDPC(512, 256) code. This choice is motivated by the observation that this code must necessarily exploit the conventional LDPC decoder, for example based on the NMS algorithm; MRB, in fact, cannot be used, even in the hybrid form, because of its too high complexity. This kind of decoder will reliably fail to decode when it over-runs the end of a CLTU, and this property may be used to stop the decoding process without the need for a tail sequence.

III. Breadboard Design and Development

This section describes the BreadBoard (BB) design and the main steps for the development of the most critical elements of the study, in particular: channel decoding algorithms (NMS and MRB decoders) and frame synchronization algorithm (as start and tail sequence detectors).

Table I presents the reference scenarios considered in this study for NE and DS or planetary exploration missions, while Table II shows the performance requirements. In Table I, reference is done to some ESA missions (Lagrange/Exomars). In Table II, E_b represents the energy per bit.

TABLE I. Parameter specification of reference scenarios

Parameter	Notation	Near-Earth (e.g. Lagrange mission)	Deep-Space (e.g. Exomars Mission)
Modulation type	-	Remnant carrier	Remnant carrier
Modulation waveform	-	PCM SP-L Direct on carrier	PCM NZR-L sine-waveform subcarrier
Modulation index	mc	1.0 radians	1.2 radians
Nominal symbol rate	Rs	64 ksps	$4000/2^9$=7.8125 sps
Carrier frequency	Fc	X-band	X-band
Subcarrier frequency	Fsc	N/A	16 kHz & 8 KHz

A. Architecture design and specification

Figure 2 shows the high-level view of the BB architecture. More precisely, the diagram represents the key functions selected during the study specifications. The system architecture design supports both NEXCODE scenarios: NE and DS. Most of the TC processing components can be configured for the corresponding scenario such as Carrier PLL Tracking or Soft-Quantizer. There are other dedicated components especially for the DS scenario, where the synchronization includes the selected enhancements in the system requirements, which are highlighted in the figure, such as FFT-Aided Carrier Tracking or Hybrid Carrier Loop.

TABLE II. Performance requirements for both reference scenarios: NE and DS

Parameter	Notation	Value
Bit Error Rate	BER	$\leq 10^{-5}$
Codeword Error Rate	CER	$\leq 10^{-5}$
Frame Error Rate	FER	$\leq 10^{-3}$
Undetected Frame Error Rate	UFER	$\leq 10^{-9}$
Eb/N0	EbN0	≤5dB
Es/N0 (Rc = ½)	EsN0	≤2dB

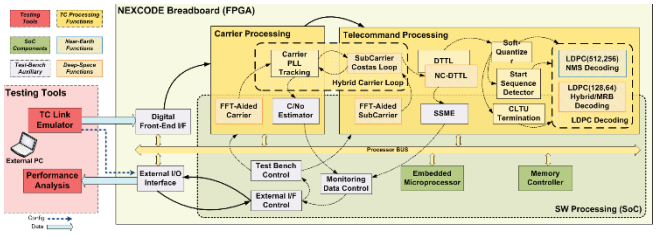

Fig. 2. High-level view of NEXCODE BB and test-bench architecture

Regarding LDPC decoding, each scenario has a different LDPC code length and decoding algorithm:

- Near-Earth: LDPC(512, 256) code with NMS Decoder.
- Deep-Space: LDPC(128, 64) code with hybrid algorithm, NMS + MRB.

NEXCODE BB is composed by a flexible architecture supported by a SoC approach. The HW components are connected to the processor bus that allows the microprocessor to control and configure every TC receiver processing unit, in order to set up the corresponding scenario or evaluate a specific novel algorithm. The SW components are also included in the TC processing.

The external elements to the BB that are required to complete the Test-Bench architecture are also shown in the figure. These are the testing tools required for system validation, that is, TC Link Emulator and Performance Analysis.

Additionally, there are components used for monitoring and communications. The Monitoring Data Control receives inputs from many components (decoded symbols, loop stress, etc.), although in the figure only the most relevant one has been highlighted. The Test Bench Control is the link between the NEXCODE BB environment and the validation approach. The C/N_0, being C the carrier power, and Split Symbol Moment Estimator (SSME) provides valuable power measurements at different signal processing stages, while the resting blocks serve as external interface between the Testing Tools and the BB. Further details on the SoC design, the SW architecture and the HW platform are reported in [9] and the other project technical reports, which are available on request to the consortium prime.

B. Breadboard implementation and test-bench

In this section we present the design and implementation of the most critical elements for the study, in particular the

channel decoding algorithms (NMS and MRB decoders) and the frame synchronization algorithm (as start and tail sequence detectors).

a) NMS decoder component description

The decisions on the architecture components were made based on the analysis of the parity-check matrix. In fact, the number of Central Processing Units (CPUs), Variable Processing Units (VPUs), and Memory Banks (MBs) are calculated based on the **H** matrix properties.

As mentioned in Section II, the circulant square submatrices dimensions are $Q = 16$ for the (128, 64) code and $Q = 64$ for the (512, 256) code, so that both horizontal step and vertical step are split into 16 sub-steps in the (128, 64) case and into 64 sub-steps in the (512, 256) case. Moreover, the partially parallel architecture is composed of $v = (n - k)/Q = 4$ CPUs and $t = n/Q = 8$ VPUs for each of the two codes. The total number of required MBs is therefore $v \cdot t = 32$ MBs for both codes. Both LDPC(128, 64) and LDPC(512, 128) codes share the same decoder architecture displayed. The only differences between the codes are:

- Different circulant sub-matrix size Q, likewise the MBs size, which is 16 cells for short and 64 cells for long code, as stated above.
- Different VPU offset for each sub-matrix, according each code parity-check matrix.

Table III summarizes the resource estimate, in terms of Look-Up Tables (LUTs), flip-flops and Random Access Memory (RAM) required. The estimate is based on the overall decoder architecture for the partially parallel architecture and also for the serial approach. It is also shown as reference the available resources in space devices from Microsemi: RTAX2000S, RTAX4000S and RT4G150. In particular, the main reference is the RTAX4000S which is used to calculate percentage of usage of the device (% column).

TABLE III. SUMMARY OF FPGA RESOURCE ESTIMATE FOR NMS DECODER WITH PARTIALLY PARALLEL ARCHITECTURE

	Parameter	LUT		Flip-Flops		RAM bits	
		units	%	units	%	units	%
Reference	Microsemi RTAX2000S	21504	53%	10752	53%	288 k	53%
	Microsemi RTAX4000S	**40320**	**100%**	**20160**	**100%**	**540 k**	**100%**
	Microsemi RT4G150	151824	377%	151824	753%	5.3 M	1005%
Analysis	1 x CPU	1360	3%	112	1%	0	0%
	1 x VPU_a (to 5 MBs)	360	1%	90	0%	0	0%
	1 x VPU_b (to 3 MBs)	216	1%	64	0%	0	0%
	1 x MB	0	0%	0	0%	384	0%
	Controller	80	0%	20	0%	0	0%
	Syndrome Control	1536	4%	1280	6%	2048	0%
	Review Serial Architecture: 1 CPU + 1 VPU_a + 40 MBs	**3336**	**8%**	**1502**	**7%**	**17408**	**3%**
	Partially Parallel Architecture: 4 CPU + 4 VPU_a +4 VPU_b + 40 MBs	**9360**	**23%**	**2364**	**12%**	**17408**	**3%**

Obviously, the partially parallel architecture needs more resources than the serial version (about 3 times more in LUTs, with the advantage of higher symbol rate as it will be explained in the latency analysis below. Despite the more demanding complexity, the resources required by the partially parallel architecture fit very well in the reference space FPGA, with resource utilization from 44% in the smallest device to 6% in the largest device considering the combinatorial logic, or LUTs (mostly used kind of resource by the algorithm). This coarse estimate could be considered quite conservative, since the allocation assumes a LUT per each combinational logic operation, while several of these could be combined in a single LUT. The estimate for controller and the syndrome control units is also included for both architectures.

The main advantage of the partially parallel architecture is the higher performance, in terms of maximum symbol rate. For each iteration of the NMS decoder, the processing of vertical and horizontal steps is accelerated with respect to the serial approach due to the parallelization.

Simultaneous access to parallel MicroBlaze™'s arranged as in the partially parallel architecture is assumed. This parallel access adds very little routing cost, and, in fact, it is compensated with the simplification of the addressing logic using the quasi-cyclic properties of the matrix **H**. As regards the throughput of the short code LDPC(128, 64), the maximum symbol rate is nearly the same as for the long code. The latency estimate is summarized in Table IV for both code lengths and architectural approaches.

TABLE IV. SUMMARY OF PROCESSING LATENCY OF NMS DECODER

Decoding Latency	LDPC(512,256)		LDPC(128,64)	
	Serial	Parallel	Serial	Parallel
Horizontal Step [ticks]	1024	256	256	64
Vertical Step [ticks]	2560	320	640	80
Iteration [ticks]	3584	576	896	144
50 iterations [ticks]	179200	28800	44800	7200
Syndrome check [ticks]	528	528	144	144
Total Decoding [ticks]	179728	29328	44944	7344
Total [ms] @100 MHz	**1,80**	**0,29**	**0,45**	**0,07**
Max Symbol rate [Ksps]	**284,9**	**1745,8**	**284,8**	**1742,9**

Based on these and other results, here not shown for saving space, we can conclude that any considered architecture complies with large margin the requirements initially fixed for the study in terms of HW resources (compared with the space of reference FPGA) and maximum symbol rate (> 64 kbps). Nevertheless, the fully serial architecture was selected for HW implementation since it easily satisfies decoder performance requirements and requires less HW resources.

b) Hybrid decoder component description

The hybrid decoder is actually composed by two decoders: NMS and MRB. The working idea for this decoder is to start the decoding procedure with NMS (characterized by lower complexity) and if and only if NMS fails, i.e., it is not able to find a valid codeword, to use the MRB (characterized by higher complexity).

The MRB algorithm can be decomposed in two parts:

- MRB Part 1: operations that must be performed once for each decoded word. These operations are step 1 to 3 of MRB algorithm described in Section II. In particular, the Gauss-Jordan elimination on **G** is a quite complex algorithm with many control decisions and matrix transformation for which HW implementation is not well suited. So, this Part 1 has been implemented in SW.
- MRB Part 2: operations that must be performed several times (depending on the MRB order) for each decoded word. These are step 5 and 6 of the algorithm; and due to the high number of iterations, a parallel implementation is necessary to reduce latency. Therefore, this part is allocated to HW implementation.

MRB Part 2 is the most interesting one for the BB implementation. It can consist of many parallel TEP Evaluation Units (TEUs). This critical processing unit is

briefly described next. For Part 1 of the algorithm, instead, the SW used for simulation was migrated and optimized for the BB microprocessor.

The SW processing burden of the algorithm is quite critical in order to comply with latency requirements. A SW profiling refinement has been performed in the target platform, considering already the SoC and including some preliminary optimizations to improve computational performance. The results are presented in Table V, where the time execution needs of the two main parts of the algorithm are decomposed. The total amount shown in the last row of the table includes Part 1 and Part 2 of the algorithm, plus the reverse ordering of the chosen codeword before passing to the O/B computer at the end of the algorithm, which is negligible (less than 0.1% of the total) compared to the two main parts highlighted. For Part 2 of the algorithm, note that the results are presented for 1000 TEP evaluations, while the target for MRB performance is 400K TEPs.

Indeed, despite the several optimizations, the time needed for 400K TEP evaluations is estimated around 60 s, which is still clearly incompatible with the latency requirement (around 2 s). More SW optimizations could be added, but the residual improvement should be very small, and it will not be enough. Therefore, a HW implementation was required for the TEU.

TABLE V. SW PROFILING OF THE MRB DECODER ALGORITHM

MRB Algorithm Section	Original SW		Only Float and Data Cache		Only Integers	
	Time [s]	%	Time [s]	%	Time [s]	%
1) Symbols and Matrix Ordering	0,8	4%	0,045	5%	0,035	20%
2) TEP Evaluation (1K iter)	19,9	96%	0,818	95%	0,146	80%
TOTAL	20,8	100%	0,865	100%	0,182	100%

Another interesting conclusion is that Part 1 in SW is a constraint for the maximum throughput of the decoder. The symbols and matrix ordering performed in this part, in fact, takes around 35 ms, that limits the minimum latency of the overall hybrid decoder and the constraint has been used as input in the HW considerations for FPGA implementation during design. As mentioned, the TEU is the best suited processing element for parallelization in HW implementation. Considering also the algorithm description and the overall BB architecture, the MRB Decoder architecture is defined by the diagram in Fig. 3.

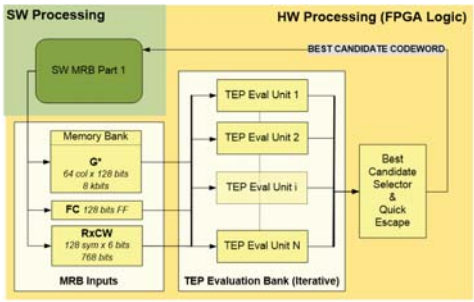

Fig. 3. MRB decoder architecture in the breadboard

For the TEU, and the parallel processing units, there are many degrees of freedom for design, balancing between maximum throughput and minimum resource usage. The TEP generator can be implemented with a set of cascading counters and can be shared by all the parallel TEUs in the bank. This can be accomplished by assigning 1 TEP pattern to each TEU, so the 1st order TEP counter is different for each TEU. For example, the [k + 1, X, Y, Z] is assigned to the 1st TEU, the [k + 2, X, Y, Z] to the 2nd TEU, and so on, until the [k + N, X, Y, Z] for the Nth TEU, where X, Y and Z can be any pattern index (rows of **G***).

The subsequent processing is the candidate codeword encoding. The operation uses the four patterns and the first candidate CodeWord (CW) to generate the resulting candidate CW as an XOR of 5 vectors of 128 bits. This operation contains redundancy between the parallel TEUs; hence an optimization is proposed, based on the fact that the TEP generator is shared by all the TEUs in the bank. The idea is displayed in Fig. 4 and it is based on pre-encoding the first candidate CW with the TEP index of order 2, 3 and 4. In the figure, the pattern of four numbers [k_i, X, Y, Z] identifies a TEP (where 0 means no error).

Once the candidate codeword is encoded in each TEU, the next step is the "Distance Calculator". This unit computes summation of soft-values, so it is expected to be the most expensive part of the TEU. The main block is an accumulator that receives the soft-values from the re-ordered Received Codeword (RxCW) buffer. The soft-values are accumulated whenever there is an encoded bit mismatch with the encoded candidate, i.e., if the XOR between the RxCW sign bit XOR with the candidate CW bit results true.

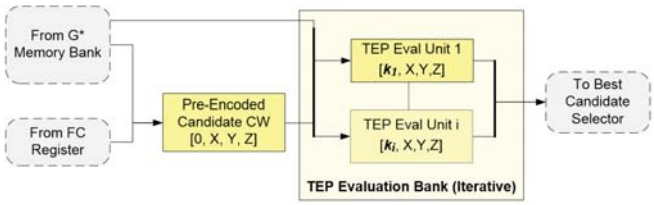

Fig. 4. MRB decoder: shared pre-encoded candidate CW for TEU

The structure described for the Distance Calculator can be accomplished by a partially parallel approach, by dividing the 128 bits CW vector in c sub-vectors of size m = 128/c, and reducing the processing latency c times. The same partially parallel architecture must be used to implement the candidate CW encoding in each TEU, thus performing serially first each bit decoding step and then the distance calculation step. The selection of c (and m) parameter depends on resources and timing constraints, being c = 1 and c = 128 the extreme options for serial and fully parallel architectures respectively.

The last processing element to implement in the HW domain the MRB algorithm is the "Best Candidate Selector and Quick Escape" block. It is just a selection of a minimum between the distances calculated by the TEUs and a comparison with the quick escape thresholds.

Table VI summarizes the FPGA resource estimate for each MRB processing element described above. The different architecture configurations demonstrate as the resources demands increase with number of TEUs, N, and the partially parallel level, parameter c. An interesting remark is that the parallelization at TEU bank level, i.e., N TEUs, has higher cost than parallelization inside the TEU, i.e., c sub-vectors in parallel. In terms of device occupation, the fully serial approach with 1 TEU would use a very small portion of the overall FPGA resources, while the occupation rises quickly with parallelization.

As mentioned, the selection of the number of TEUs, N, and the c (and m) parameter depends on resources and timing constraints. As regards algorithm throughput, the maximum is limited by the SW processing of MRB Part 1 that, as stated

above, is around 36 ms. Furthermore, it must be considered the cost of the first part of the hybrid algorithm, the NMS decoder, which would spend around 0.45 ms for the LDPC(128, 64) decoding with a maximum number of iterations I_{max} = 50. With these two initial costs, we can calculate the remaining processing time, T_{max}, with respect to the overall latency target. Table VII summarizes the results of these calculations for the maximum throughput and for the NEXCODE latency requirement (for 62.5 sps). The maximum symbol rate achievable would be around 3.5 ksps, but it needs to be adjusted to a valid figure of the standard symbol rate for TC; in this case the nearest normalized symbol rate is 2 ksps. With the calculated time budget for the MRB Part 2, T_{max}, the overall clock cycles for the algorithm can be obtained, as well as the number of cycles per each of the 400K TEP evaluation needs in the worst case, in order to obtain a first indication of the timing constrains as a function of the number of TEUs. At first view, it should be enough 1 TEU for the NEXCODE throughput target with 502 cycles/TEP, while it would be hard to reach maximum symbol rate with this simple architecture (only 6 cycles/TEP).

TABLE VI. MRB PART 2 FPGA RESOURCE ESTIMATE FOR DIFFERENT IMPLEMENTATION ALTERNATIVES

Parameter			LUT units	LUT %	Flip-Flops units	Flip-Flops %
Reference		Microsemi RTAX2000S	21504	53%	10752	53%
		Microsemi RTAX4000S	**40320**	**100%**	**20160**	**100%**
		Microsemi RT4G150	151824	377%	151824	753%
Analysis		TEP Generator	42	0,1%	3	0,01%
		Pre-Encode CW	512	1%	256	1%
		CW encoder / TEU	258	1%	128	1%
		Distance Calculator / TEU	910	2%	7	0%
		Best Candidate Selector / TEU	146	0%	134	1%
	1 TEU	**Fully Serial (c=1)**	**1486**	**4%**	**528**	**3%**
		Partially Parallel (c=8)	**3180**	**8%**	**577**	**3%**
		Partially Parallel (c=64)	**18496**	**46%**	**969**	**5%**
		Fully Parallel (c=128)	**35772**	**89%**	**1289**	**6%**
	8 TEUs	**Fully Serial (c=1)**	**6515**	**16%**	**1473**	**7%**
		Partially Parallel (c=16)	**38315**	**95%**	**2313**	**11%**
	32 TEUs	**Fully Serial (c=1)**	**28213**	**70%**	**4713**	**23%**
		Partially Parallel (c=4)	**51349**	**127%**	**5385**	**27%**

As a summary of the FPGA implementation analysis, we can cross resources and timing estimate. Table VIII shows the trade-offs between HW complexity and latency obtained. The HW needs are shown as resources units and percentage of the reference device, RTAX4000S. The latency results are presented in ms and as percentage of the remaining processing time, T_{max}, in Table VII, for the target latencies, i.e., for NEXCODE requirement (62.5 sps) and also for maximum throughput (2 ksps). Obviously, a latency percentage higher than 100% means that the latency constraint is not met, and the same applies to HW resources.

According to these estimates, two alternative architectures have been proposed at the end of the study:

- Only one TEU with fully serial architecture (first line in the table): this uses the minimum HW resources and complies with latency requirement for 62.5 sps. It is proposed as the lowest size HW implementation to meet system requirements.
- Three TEUs with partially parallel architecture (c = 8) (sixth line in the table): this uses the minimum HW resources to comply with maximum throughput target of 2 ksps. It is proposed as the highest throughput architecture with moderate HW resources requirements. This option was selected for BB implementation since it allowed to speed-up the decoder testing campaign.

TABLE VII. TIMING CONSTRAINTS FOR THE PARALLEL TEU DEPENDING ON LATENCY TARGET

Parameter		Max Throughput	Latency Req.
Hybrid Algorithm Initial Cost	MRB Part 1 [ms]	35,69	
	Tmax NMS [ms]	0,45	
	Initial Cost [ms]	36,14	
	Max. Symbol Rate (sps)	3542	62,5
	Norm. Symbol Rate (sps)	2000	62,5
	Latency [ms]	64	2048
MRB Part 2	T_{max} [ms]	27,9	2011,9
	T_{max} [cycles] @100 MHz	2,79E+06	2,01E+08
Cycles/TEP for 400K	Serially (1 TEU)	6	502
	Parallel (4 TEUs)	27	2011
	Parallel (16 TEUs)	111	8047
	Parallel (32 TEUs)	222	16094

TABLE VIII. MRB PART 2 HW IMPLEMENTATION: RESOURCES VS TIMING TRADE-OFFS

MRB Part 2 HW Architecture		LUT units	LUT %	Flip-Flops units	Flip-Flops %	Latency ms	Latency % at 62.5 sps	Latency % at 2 Ksps
1 TEU	**Fully Serial (c=1)**	1486	4%	528	3%	**512**	**25%**	1838%
	Partially Parallel (c=32)	9732	24%	745	4%	16	1%	57%
2 TEUs	**Fully Serial (c=1)**	2169	5%	663	3%	256	13%	919%
	Partially Parallel (c=16)	10029	25%	873	4%	16	1%	57%
3TEUs	**Fully Serial (c=1)**	2864	7%	798	4%	171	8%	613%
	Partially Parallel (c=8)	8170	20%	945	5%	21	1%	**77%**
4TEUs	**Fully Serial (c=1)**	3570	9%	933	5%	128	6%	459%
	Partially Parallel (c=8)	10682	26%	1129	6%	16	1%	57%

c) HW breadboard implementation for frame synchronization

The frame synchronization has been implemented using the S-LRT as mentioned in Section II. This metric can be expressed as

$$\Lambda_{\text{S-LRT}}(n) = \left| \sum_{k=1}^{N_S} y_{n+k-1} s_k \right| - \sum_{k=1}^{N_S} \left| y_{n+k-1} \right| .$$

In this expression, N_S is the length of the start (or tail) sequence and y_k and s_k are, respectively, the kth symbol of the input signal and the start (or tail) sequence. Both start ant tail sequences are stored in the HW component.

A fully HW implementation has been used. As both sequences do not have to be checked simultaneously, the same HW component is used to obtain their metric computations. Two modes, which are chosen by the SW are available:

- For the start sequence, as its initial position is unknown, the metric is obtained once for each new symbol. In each computation it compares the last N_S = 64 symbols of the received signal against the stored start sequence.
- For the tail sequence, the computation is performed once at the end of each CW. This is done once each n symbols. In this case, when used, it compares the first 128 symbols following each CW.

Other details on HW implementation can be found in [9].

d) Carrier/Subcarrier tracking in deep-space scenario

Carrier and subcarrier acquisition, recovery and tracking

are critical bottleneck in the DS scenario. With such low symbol rate (7.8125 sps), the target SNR implies carrier to noise power in the carrier loops below 10 dB, which is the minimum recommended ratio for correct carrier acquisition and tracking. The enhanced approaches evaluated until now provide marginal gains in the low SNR region, and yet they do not meet the target E_s/N_0 in the DS scenario.

IV. Summary of Performance Results

This section reports a subset of the results obtained through the wide campaign of simulations done for testing the efficiency and functionalities of the breadboard. For the sake of brevity, we limit to discuss the CER performance.

Figures 5 and 6 show the NMS CER for LDPC(128, 64) and LDPC(512, 256) decoding, respectively. Both figures present software simulator and BB results for 3- and 6-bit of quantization. Comparing the achieved results with the theoretical values, reported in [3], it is verified that the NMS performance is close to the expected results, with a CER difference with respect to the SW simulator of about 0.1 dB for both the short and the long codes. The target CER $\leq 10^{-5}$ is estimated for E_s/N_0 around 2.5 dB and 2.7 dB, for the short code, and around 0.8 dB and 0.9 dB for the long code, by assuming 6-bits and 3-bits quantization, respectively, in the HW breadboard implementation with NMS only decoder.

Figure 7 presents the CER with NMS and hybrid (NMS + MRB) decoding for the LDPC(128, 64) code by assuming 6-bit quantization, both for software and hardware. The comparison with the theoretical values, reported in [3] shows a slight difference between the software and the breadboard results for the hybrid algorithm, where a maximum difference of 0.15 dB is observed. The target CER $\leqslant 10^{-5}$ is estimated for E_s/N_0 around 0.7 dB in the HW breadboard implementation with hybrid decoder.

V. Conclusions

The assessment of the code impact on the receiver functionalities and identification of bottlenecks in the transponder synchronization schemes, indicate that both acquisition and tracking loops must be upgraded to fully exploit the potential gains that more powerful coding schemes bring to telecommand in space communications. Current standard configurations will not be able to operate at the lower link budget for the lowest rate operational modes. Both NMS and hybrid decoders were implemented and validated in the breadboard being also observed losses of about 0.1 dB for most of the test cases. This breadboard and the associated test-bench allows the validation of the novel decoding techniques proposed in the new CCSDS standard, thus minimizing the risk of adoption of the new code(s) for future missions.

References

[1] CCSDS, "TC Synchronization and Channel Coding," Blue Book, CCSDS 231.0-B-3 (Sep. 2017).

[2] M. Baldi, M. Bertinelli, F. Chiaraluce, P. Closas, R. Garello, N. Maturo, M. Navarro, J. M. Palomo, E. Paolini, S. Pfletschinger, P. F. Silva, L. Simone, and J. Vilà-Valls, "NEXCODE: Next generation uplink coding techniques," Proc. TTC 2016 International Workshop on Tracking, Telemetry and Command Systems for Space Applications, Noordwijk, The Netherlands, 13-16 Sep. 2016.

[3] M. Baldi, M. Bertinelli, F. Chiaraluce, P. Closas, P. Dhakal, R. Garello, N. Maturo, M. Navarro, J. M. Palomo, E. Paolini, S. Pfletschinger, P. F. Silva, L. Simone, and J. Vilà-Valls. "State-of-the-art space mission telecommand receivers," IEEE Aerospace and Electronic Systems Magazine, vol. 32, no. 6, pp. 4-15, Jun. 2017.

[4] M. Baldi, N. Maturo, E. Paolini, and F. Chiaraluce. "On the use of ordered statistics decoders for low-density parity-check codes in space telecommand links," EURASIP Journal on Wireless Communications and Networking, 2016:272, 15 pages, 2016.

[5] M. Baldi, F. Chiaraluce, N. Maturo, G. Liva, and E. Paolini. "A hybrid decoding scheme for short non-binary LDPC codes," IEEE Communications Letters, vol. 18, no. 12, pp. 2093-2096, Dec. 2014.

[6] M. P. C. Fossorier, "Iterative reliability-based decoding of low-density parity check codes," IEEE Journal Selected Areas in Communications, vol. 19, no. 5, pp. 908–917, May 2001.

[7] M. Chiani and M. G. Martini, "On sequential frame synchronization in AWGN channels," IEEE Transactions on Communications, vol. 54, no. 2, pp. 339-348, Feb. 2006.

[8] J. L. Massey, "Optimum frame synchronization," Transactions on Communications, vol. COM-20, no. 4, pp. 115–119, Apr. 1972.

[9] Next Generation Uplink Coding Techniques – Final Report, 28 Mar. 2019.

[10] J. Vilà-Valls, M. Navarro, P. Closas and M. Bertinelli, "Synchronization challenges in deep space communications," IEEE Aerospace and Electronic Systems Magazine, vol. 34, no. 1, pp. 16-27, Jan. 2019.

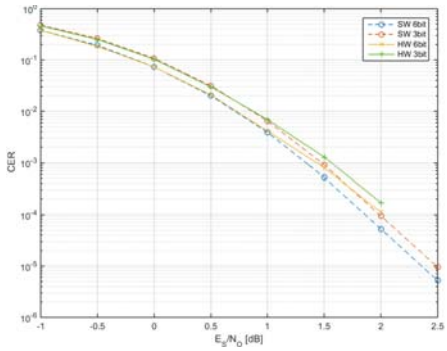

Fig. 5. LDPC(128, 64) NMS decoder CER performance

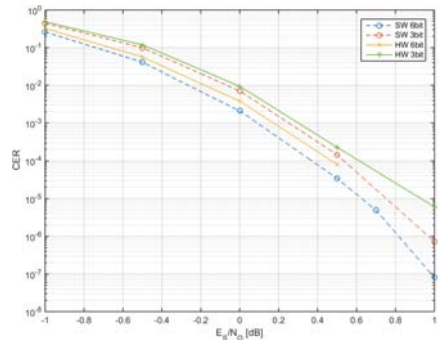

Fig. 6. LDPC(512, 256) NMS decoder CER performance

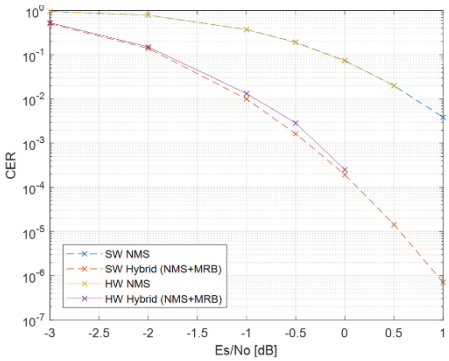

Fig. 7. LDPC(128, 64) hybrid (NMS + MRB) decoder CER performance