Phase Offset Estimation using Enhanced Turbo Decoders

Bartosz Mielczarek and Arne Svensson
Communication Systems Group
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
PH: +46 31 772 1763 FAX: +46 31 772 1748

***Abstract* - This paper discusses a realistic turbo coding system with the signal phase which has not been perfectly estimated. We propose improved decoding algorithms for the situations when the residual phase error can be modelled by the Gaussian probability distribution and a Markov chain, a model which can be used in many actual phase estimators. It is shown that increasing the state space of the decoders can decrease the bit error probability.**

I. INTRODUCTION

One of the most important factors determining the efficiency of a wireless system is the power required for successful transmission of data. The required power needed to reliably transmit a signal can be reduced by using special coding techniques. One of the latest and the most prominent of such techniques is turbo coding with a performance which is almost able to reach the Shannon channel capacity limit [1].

The problem lies, however, in the practical applications of the turbo codes. Unfortunately, reducing the transmitted power makes it more difficult to estimate the channel and properly synchronize the phase of the incoming signal.

In our paper we propose algorithms aiming at improving performance of turbo-coded systems with non-perfect phase offset estimation.

II. PHASE SYNCHRONIZATION

In the majority of digital wireless communication systems, the incoming HF signal needs to be downconverted to lower frequency ([3]). The good frequency and phase synchronization is therefore essential for the reliability of wireless systems.

There exist a number of different techniques estimating carrier parameters (such as Phase Locked Loops, Costas loops etc.) but none of these algorithms succeed to provide perfect estimation of the carrier signal. One of the reasons for such performance loss is that synchronization of phase is done before the decoding process and cannot use the code properties to improve its accuracy. This is due to the fact that most decoders will not work without a proper phase estimation and must rely on some initial estimates of the signal phase. If, on the other hand, the phase estimator/decoder knows the structure of the data signal, it can use this knowledge in joint phase and data estimation and improve the system's performance. We will use this approach for the algorithms presented in this paper.

III. SYSTEM MODEL

The system analysed in this paper is presented in Fig. 1. A typical turbo encoder ([1],[2]) of rate 1/3 generates codewords consisting of systematic bits x_k^s and the parity bits $x_k^{p,1}$, $x_k^{p,2}$. The stream of code bits is BPSK-modulated and transmitted over an AWGN channel as real-valued signal samples c_k. The encoded signal suffers from a phase noise process θ_k (which can be a result of a fading process or oscillator instability) and is corrupted by the white, Gaussian noise. The incoming distorted signal is fed to the phase synchronizer, which produces estimates of the phase noise process $\hat{\theta}_k$. After adjusting the phase error, the signal is decoded (the decoded data sequence can then be used to refine the phase estimation process but this problem is not addressed in this paper).

Formally, the signal after the AWGN channel, phase estimation and receiver matched filtering (the timing recovery is assumed to be perfect) can be expressed as

$$y_k = e^{j\phi_k} c_k + n_k, \qquad (1)$$

where y_k is the complex received signal (which can be the systematic bit y_k^s, the first parity bit $y_k^{p,1}$ or the second parity bit $y_k^{p,2}$) and n_k is the white, additive, Gaussian complex noise with $E[|n_k|^2] = N_0$. ϕ_k is the residual phase offset estimation error $\phi_k = \hat{\theta}_k - \theta_k$; its statistics are discussed in the next section. Note that the amplitude of the signal is assumed to be constant, i.e. the fading is compensated by a perfect power control. The extension of the channel model to the non-compensated fading channels is relatively straightforward and will not be discussed in this paper.

IV. RESIDUAL PHASE ERROR MODELLING

The residual phase error ϕ_k can be modelled as Gaussian distributed, with known variance σ_ϕ^2 and zero mean for non-biased phase estimators (which are the most common solutions [3]). Such an assumption is rather popular in the existing literature and seems to be quite realistic since typical synchronizers produce an error distribution with a similar shape and known variance (for example, the Tikhonov distribution after the PLL loop, see [5]).

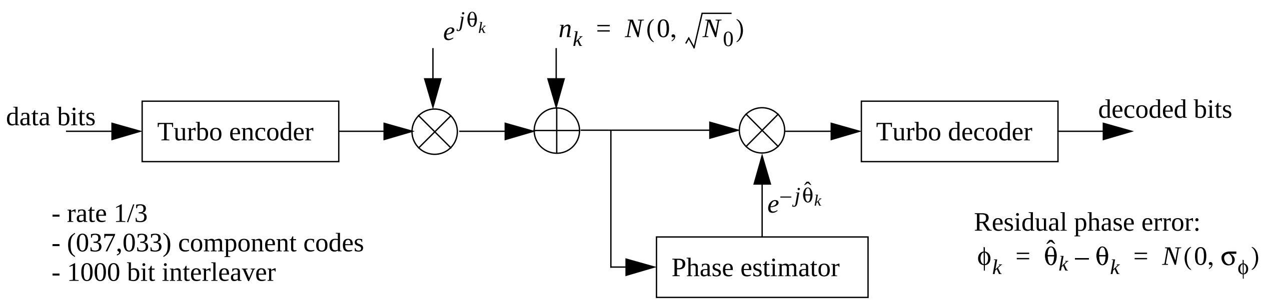

Fig. 1. System model

Moreover, since the phase estimation is usually an effect of some kind of non-perfect averaging of the initial signal, the residual phase error values may remain correlated to some extent (with a perfect phase estimation there would be no correlation between residual phase error). A new approach to solving this problem is to model the actual sign of the phase error (which tends to remain constant for a number of samples) as a simple Markov chain. This way we introduce a simple memory model to the channel which is relatively easy to incorporate into the turbo decoder.

By using the above unified framework (graphically shown in Fig. 2.) to model the phase error, our approach can be tailored to many existing phase synchronizers, just by applying the actual phase error variance and the crossover probability for the phase error sign change in the Markov model. The presence of a channel interleaver can be also included by setting the cross-over probability of the Markov model to 0.5, i.e., removing any statistical correlation between the consecutive phase errors. Even though more complicated models can be used, our approach achieves quite good results without increasing the decoding complexity too much.

V. Turbo Decoding Algorithm

Due to the presence of the interleaver in the decoder, the optimal decoding of turbo codes would be very complex. The practical sub-optimal implementations split the process into two separate processes, in which both component codes are decoded independently ([2]). The connection between the codes is implemented by exchanging soft extrinsic information and using a form of the MAP decoding (minimizing the probability of the bit error) for each component code.

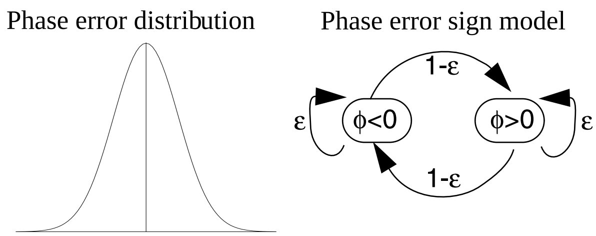

Fig. 2. Phase error modelling

The most commonly used decoding algorithm is the BCJR-MAP algorithm ([6]) which computes the soft bits using the log likelihood ratio (LLR) (see [2])

$$L(u_k|\boldsymbol{Y}) = \log\left(\frac{P(u_k = +1|\boldsymbol{Y})}{P(u_k = -1|\boldsymbol{Y})}\right), \tag{2}$$

where $\boldsymbol{Y}$ is the whole received code sequence. The LLR is calculated as

$$L(u_k|\boldsymbol{Y}) = \log\left(\frac{\sum_{S^+} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s)}{\sum_{S^-} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s)}\right), \tag{3}$$

where $\tilde{\alpha}_k(s')$ and $\tilde{\beta}_k(s)$ are the recursively calculated probabilities of arriving at state s' (computed from the start of the trellis) and state s (computed form the end of the trellis), respectively (for details see [2]). The term $\gamma_k(s',s)$ is the probability of the transition between states s' and s and is given by

$$\begin{aligned}\gamma_k(s',s) &= p(u_k)p(\mathrm{y}_k|u_k) \\ &\propto \exp\left(\frac{1}{2}u_k(L^e(u_k) + L_c y_k^s)\right)\exp\left(\frac{1}{2}L_c y_k^p x_k^p\right),\end{aligned} \tag{4}$$

where $L^e(u_k)$ and $L_c = 4E_c/\mathrm{N}_0$ are the extrinsic information about bit k (calculated by the first decoder, $L_{12}^e(u_k)$, or by the second decoder, $L_{21}^e(u_k)$) and the channel reliability factor of the decoder (E_c is the code bit energy), respectively. The numerator of (3) includes all the transitions which correspond to data bit $u_k = +1$ (S^+) and the denominator all the transitions which correspond to $u_k = -1$ (S^-).

VI. Large Phase Error Correction

In general, the main performance loss is caused by large phase errors which result from long tails of the error distribution. Luckily, the large phase errors are easier to detect than small ones, which suggests that the synchronization algorithm can concentrate primarily on reducing them instead of trying to correct all phase errors ([7]).

Correcting phase errors is equal to rotating the received signal samples in the opposite direction of the actual phase error.

The two parameters which must be known prior to such correction are the size and the sign of the actual phase offset. Assuming that the size of the rotation is fixed to be the expected value of the absolute value of the residual phase error, the remaining uncertainty is the sign of the offset. One of the ways to detect it is to create two sets of samples $y_k(+)$ and $y_k(-)$ as

$$y_k(\pm) = y_k e^{\pm j\overline{|\phi|}}, \tag{5}$$

where $\overline{|\phi|} = E[|\phi|]$ is the rotation size, and use the decoder to compare their metrics.

VII. Turbo Decoder Modification

To detect the sign of the phase offset, (2) can be reformulated in the following form

$$L(\phi_k|\boldsymbol{Y}) = \log\left(\frac{P(\phi_k > 0|\boldsymbol{Y})}{P(\phi_k < 0|\boldsymbol{Y})}\right), \tag{6}$$

which is the LLR of the probability that the *k*th code bit had the positive phase error and the probability of the negative phase error. One of the ways to solve (6) is to use the turbo technique, i.e., use two decoders to iteratively improve estimation of the rotation signs.

The proposed decoder architecture is shown in Fig. 3. Two specially modified APP decoders are connected in a feedback loop, exchanging soft data information and soft phase error sign information (discussed later in the paper) for all symbols. The two streams of soft messages are properly interconnected using interleavers and deinterleavers.

In order to incorporate the phase rotation algorithm, the transition probability from (4) is redefined as

$$\begin{aligned}\gamma_k(s',s|\phi_k^s, \phi_k^p) &= p(u_k)p(\mathrm{y}_k|u_k, \phi_k^s, \phi_k^p)p(\phi_k^s, \phi_k^p)\\ &\propto p(\phi_k^s, \phi_k^p)\exp\left(\frac{1}{2}u_k L^e(u_k)\right) \quad , \\ &\cdot \exp\left(\frac{1}{2}L_c y_k^p(\phi_k^p)x_k^p\right)\exp\left(\frac{1}{2}u_k L_c y_k^s(\phi_k^s)\right)\end{aligned} \tag{7}$$

where the received samples y_k^s and y_k^p are conditioned on the offset sign events ϕ_k^s and ϕ_k^p, respectively, having the joint probability distribution $p(\phi_k^s, \phi_k^p)$. Such a modification suggests that additional states and transitions are needed in the decoder to detect the sign of the offsets.

VIII. Modified APP Decoder

In general the phase error sign can vary from one signal sample to the other. Even if it can be modelled as the Markov chain, the use of channel interleaver (a widely used solution aiming at combating the burst errors, common in wireless channels) may effectively remove the correlation between consecutive phase error signs.

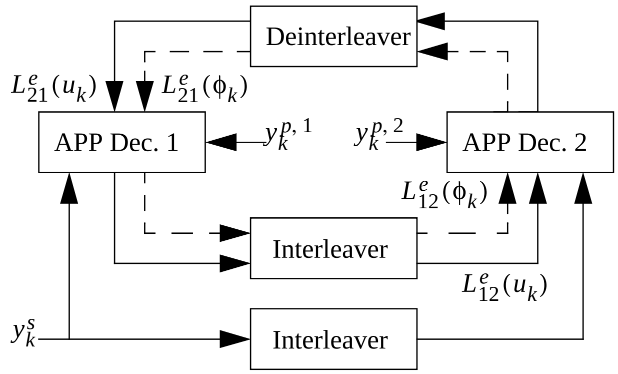

Fig. 3. Modified decoder architecture with additional soft phase information exchange

To fully represent such a situation, we have to split each original state of the code into four different states, one representing two positive shifts denoted henceforth as (+,+), one representing two negative transitions (-,-) and two with mixed error signs (+,-) and (-,+).

Fig. 4. shows the construction of the trellis for one bit transition. The probability of the transitions are defined as

$$\begin{aligned}p_k^{(+,+)} &= p(\phi_k^s > 0, \phi_k^p > 0)\\ p_k^{(+,-)} &= p(\phi_k^s > 0, \phi_k^p < 0)\\ p_k^{(-,+)} &= p(\phi_k^s < 0, \phi_k^p > 0)\\ p_k^{(-,-)} &= p(\phi_k^s < 0, \phi_k^p < 0)\end{aligned} \quad . \tag{8}$$

These probabilities can be initially set to values reflecting the a-priori information (when the Markov model is used) or be set equal (if no initial phase correlation is expected when, for example, a channel interleaver was used).

The enhanced APP decoder works in a usual way using the BCJR algorithm except that four times as many states have to be included in (3). Moreover, the transition probabilities are calculated using (7) and (8). In this way, the extrinsic information about phase offsets can be calculated after a decoding pass, hopefully with the majority of the rotations correctly detected.

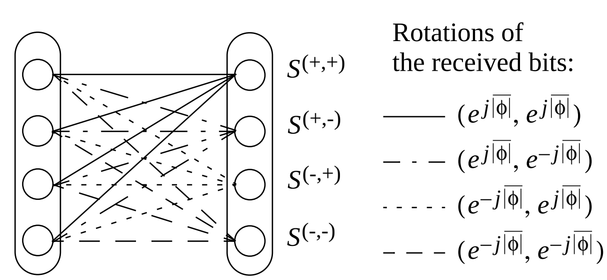

Fig. 4. General modification of the trellis transition

After the decoding pass, each component decoder produces soft information about the information bits and, in addition to that, phase information about systematic bits as

$$L^{e,s}(\phi_k) = \log\left(\frac{\sum_{S^{(+,*)}} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s)}{\sum_{S^{(-,*)}} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s)}\right), \qquad (9)$$

where $S^{(+,*)}$ are transitions between all the $S^{(+,+)}$ and $S^{(+,-)}$ states and $S^{(-,*)}$ are transitions between the $S^{(-,+)}$ and $S^{(-,-)}$ states. The soft phase information about the parity bits is produced as

$$L^{e,p}(\phi_k) = \log\left(\frac{\sum_{S^{(*,+)}} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s)}{\sum_{S^{(*,-)}} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s)}\right), \qquad (10)$$

where $S^{(*,+)}$ are transitions between all the $S^{(+,+)}$ and $S^{(-,+)}$ states and $S^{(*,-)}$ are transitions between the $S^{(+,-)}$ and $S^{(-,-)}$ states.

The transformations between the actual probabilities and soft information can be derived from (6) (see [8]) and, assuming that the phase offsets are independent (i.e., only the systematic phase offsets are common for both component codes), are given as

$$p_k^{(+,+)} = p_k^{(+,-)} = \frac{\exp(L^{e,s}(\phi_k))}{1 + \exp(L^{e,s}(\phi_k))}, \qquad (11)$$

$$p_k^{(-,+)} = p_k^{(-,-)} = 1 - p_k^{(+,+)}, \qquad (12)$$

these probabilities are then passed to the second decoder (after proper interleaving) along with the soft information bits and the same procedure is repeated.

IX. Simplified APP Decoder

If no channel interleaver is used, there is a large probability that both systematic and parity bits (fed directly to the first APP decoder, see Fig. 3.) will experience phase errors with the same signs. This fact can be used to simplify the construction of the first APP decoder and improve the performance of the system (the second decoder cannot be simplified, due to the interleaving of the systematic bits). With the above assumption the number of additional states in the first APP decoder can be reduced to only twice as many as the original one.

The assumption of the correlated phase error signs can be extended to the second parity bit as well, which will slightly change the exchange of the phase information between the APP decoders. Since, after the first APP decoding, the soft phase error sign information has to be interleaved for the systematic bits, the transformation between the actual probabilities and soft information passed to the second decoder will be given as

(13)

$$p_k^{(+,+)} = \frac{\exp(L_{12}^e(\phi_l))}{1 + \exp(L_{12}^e(\phi_l))}\frac{\exp(L_{12}^e(\phi_k))}{1 + \exp(L_{12}^e(\phi_k))}, \qquad (14)$$

$$p_k^{(+,-)} = \frac{\exp(L_{12}^e(\phi_l))}{1 + \exp(L_{12}^e(\phi_l))}\frac{1}{1 + \exp(L_{12}^e(\phi_k))}, \qquad (15)$$

$$p_k^{(-,-)} = \frac{p_k^{(+,+)}}{\exp(L_{12}^e(\phi_l) + L_{12}^e(\phi_k))}, \qquad (16)$$

$$p_k^{(-,+)} = 1 - p_k^{(-,+)} - p_k^{(+,+)} - p_k^{(-,-)} \qquad (17)$$

where $L_{12}^e(\phi_l)$ is the soft phase value of the systematic bit l, corresponding to the systematic bit k after interleaving as $L_{12}^e(\phi_k) = \Pi(L_{12}^e(\phi_l))$, where Π is the interleaver function.

Also the feedback phase information message from the second APP decoder to the first APP decoder has to be modified. This is done by combining the information about the systematic and parity offsets ([8]) to obtain the final soft phase value $L_{21}^e(\phi_k)$ as

$$L_{21}^e(\phi_k) = \log\frac{1 + e^{L_{21}^{e,s}(\phi_l) + L_{21}^{e,p}(\phi_k)}}{e^{L_{21}^{e,s}(\phi_l)} + e^{L_{21}^{e,p}(\phi_k)}}, \qquad (18)$$

where $L_{21}^{e,s}(\phi_l)$ is the soft phase value of the systematic bit l, corresponding to the systematic bit k after deinterleaving as $L_{21}^{e,s}(\phi_k) = \Pi^{-1}(L_{21}^{e,s}(\phi_l))$, where Π^{-1} is the inverse interleaver function.

Finally, the transition probabilities are calculated as

$$p_k^{(+,+)} = \frac{\exp(L_{21}^e(\phi_k))}{1 + \exp(L_{21}^e(\phi_k))}, \; p_k^{(-,-)} = 1 - p_k^{(+,+)} . \qquad (19)$$

Using the modified first APP decoder, the complexity of the enhanced turbo decoder can be reduced by 25% (the second decoder must have quadrupled number of states).

X. Algorithm Performance

To test the performance gains and discuss the properties of the algorithm we chose to employ a system experiencing a very severe phase noise. The synchronizer uses a residual carrier which is Wiener-filtered to track the rapid changes of the phase. Such systems have been proposed for deep-space communications ([9]) and their properties are similar to the very fast fading wireless channels.

The phase noise process is generated by a program reflecting actual phase noise characteristics of a deep-space communication link (oscillator instability). The parameter ε of the Markov channel is empirically calculated to be around 0.6. The simulations were conducted with the CCSDS (Consultative Committee for Space Data Systems) (023,033) turbo code ([10]) and an interleaver size of N=1000. The number of decoding iterations was set to 10. All BER calculations were

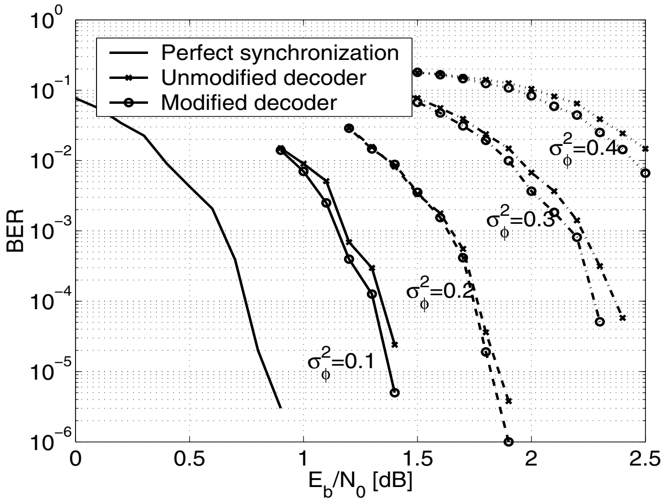

Fig. 5. BER performance of the improved decoder (circled line), classical decoder (crossed line) and the system with ideal synchronization (solid line) for different phase error variances.

based on using data from 10000 transmitted blocks with independent, randomly generated parameters. Fig. 5. presents the behaviour of the simplified phase offset correction algorithm for different phase error variances. It improves the BER performance of the traditional scheme, introducing a gain of approximately 0.1-0.2dB. The gain is particularly visible for low quality phase estimation and in the region of large SNRs.

Initially, the simulations of the system showed that repeated exchange of the soft phase information leads to convergence to the incorrect solution. The reason for it is that the correlation between the samples (high crossover probability) is rather small and the successive iterations do not contribute a lot of new information to the soft phase information. After noticing this, the algorithm was slightly modified by turning of the feedback from the second APP decoder to the first APP decode, which eliminated the convergence problem.

Moreover, increasing number of iterations proved to further improve the performance of the enhanced system. Unfortunately, due to the time and computer capacity constraints, these results could not be presented here and will be discussed in future papers.

XI. Conclusions

Phase synchronization is currently one of the biggest problems when implementing turbo codes in real wireless systems. In this paper, we introduced a general method of modelling the phase estimation errors, which can be used with different phase synchronizers. This method was then used to modify the classical turbo decoder structure and proved to improve the performance, even in a very severely distorted system.

The analysis of the proposed solution is still incomplete and further research is still necessary. It is not quite clear how the iterative process will proceed with different types of phase errors, preliminary results suggest that the soft phase information exchange must be terminated when the correlation between phase samples is low.

Moreover, the discussion of the specific system parameters such as the rotation value must follow since there exists an optimal rotation value, unique for every set of phase error parameters. It is, however, relatively safe to say that a lot is to be gained by incorporating phase synchronization into the turbo decoder. Such solutions may be the only way to combat heavily distorted channels.

Acknowledgements

I would like to thank Dr. Dariush Divsalar and Dr. Jon Hamkins at the Jet Propulsion Laboratory for providing me with an idea for this work and the phase noise generator. The work was supported by the Swedish Foundation for Strategic Research under the Personal Computing and Communication grant, a scholarship of Telefonaktiebolaget LM Ericsson's Stiftelsen för främjande av elektroteknisk forskning and a scholarship of Stiftelsen för internationalisering av högre utbildning och forskning (STINT).

References

[1] Berrou, C., Glavieux, A., Thitimajshima, P., "Near Shannon limit error-correcting coding and decoding: turbo codes," *ICC 1993*, pp. 1064-1070.

[2] Barbulescu, S.A., *Iterative Decoding of Turbo Codes and Other Concatenated Codes*, PhD Dissertation, University of South Australia, 1996.

[3] Meyr, H., Moeneclaey, M., Fechtel, S.A., *Digital Communication Receivers*, Wiley 1998

[4] Mengali, U., D'Andrea, A.N., *Synchronization Techniques for Digital Receivers*, Plenum Press 1997

[5] Proakis, J.G., Digital Communications, McGraw-Hill 1995

[6] L. Bahl, J. Cocke, F. Jelinek, and J.Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans. Inf. Theory, pp. 284-287, Mar. 1974

[7] B. Mielczarek, *Synchronization in Turbo Coded Systems*, Licentiate thesis, Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, 2000

[8] J. Hagenauer, "The Turbo Principle: Tutorial and State of the Art," Proceedings International Symposium on Turbo Codes and Related Topics, Brest, France, pp. 1-11, Sept. 1997

[9] J. Hamkins, "A Joint Receiver-Decoder for Convolutionally Coded Binary Phase-Shift Keying (BPSK)," TMO Progress Report 42-139, November 1999.

[10] Consultative Committee for Space Data Systems, 101.0-B-4: Telemetry Channel Coding, Blue Book. May 1999