

REAL-TIME CMA EQUALIZATION OF SOQPSK FOR AERONAUTICAL TELEMETRY

Arlene Cole-Rhodes, Henry Umuolo, Farzad Moazzami

**Morgan State University
Baltimore, MD 21251**

Michael Rice

**Brigham Young University
Provo, UT 84604**

ABSTRACT

This paper presents the results of using the constant modulus algorithm (CMA) to recover a shaped offset quadrature-phase shift keying (SOQPSK)-TG modulated signal, which has been transmitted over an aeronautical telemetry channel using the iNET data packet structure. The iNET-packet structure contains known data bits (the preamble and asynchronous marker (ASM) bits) within each data packet, which can be used to determine the minimum mean square error (MMSE) equalizer and is then used as an initializer for the CMA algorithm. A baseline analysis of the performance of the MMSE-initialized CMA equalizer has been previously done using average bit error rates (BER). In order to achieve real-time processing, in this implementation the CMA equalizer weights are updated using a new data packet for each adaptation of the equalizer during which the bit error analysis is being done. Two implementations of the block processing CMA are compared. The first achieves convergence of the equalizer in the startup phase by processing a single fixed packet and the second achieves convergence by processing the multiple packets of received data. Performance evaluation results based on bit error rates, are presented for these two methods and they are compared to a previously determined baseline performance. We note that with the proper choice of parameters these real-time methods can achieve the baseline performance.

INTRODUCTION

We investigate the effectiveness of the CMA equalizer in recovering an unknown data bit-stream that has been transmitted over an aeronautical telemetry channel with shaped offset QPSK, version 'TG' (SOQPSK-TG), using the iNET data packet structure. The CMA equalizer has been widely used and is the most popular blind adaptive equalizer in use today because of its relative simplicity, and also its good global convergence properties [1]. It is well studied, and it is known to be effective for signals and constellations that possess the constant modulus property.

Since SOQPSK-TG is a partial response continuous phase modulation (CPM), it should be a perfect candidate for this equalizer, but previous work by Law [2] had noted some difficulties with the use of CMA to equalize SOQPSK-TG over a telemetry channel. Previous work in [3] showed that a CMA equalizer with block processing is an effective equalizer for recovering SOQPSK-TG modulated signals in the the iNET data packet structure which have been transmitted over an aeronautical channel. This equalizer is especially effective when the known data bits in the form of the preamble and asynchronous

marker (ASM) bits contained in the iNET-packet structure are used to provide an alternative method of initializing the CMA equalizer. Since this decreases the convergence time of the adaptation and actually produces a lower final bit error count when compared to the usual method of center tap initialization. For real-time processing, the block processing CMA algorithm is applied to a data stream of multiple packets, where the bit error analysis is done after convergence of the equalizer is achieved, and we then continue to update the CMA weight vector using each new incoming data packet. The effect of the equalizing CMA radius on the BER performance is also investigated.

COMMUNICATION SYSTEM MODEL

The INET packet structure is made up of the known preamble of length, 128 bits and ASM is of length, 64 bits, together with the actual data of length, 6144 bits. The total packet length is 6336 bits and this packet structure is shown in Fig. 1. The transmitted signal is SOQPSK-TG, a partial-response continuous phase modulation (CPM) with a constrained ternary alphabet. The precoder, modulation index and frequency pulse are described in [5,6].



Fig. 1: INET Packet structure

The system communication model is shown in Fig. 2. A bit stream, formatted as shown in Fig. 1, is modulated using SOQPSK-TG. The SOQPSK-TG signal experiences multipath propagation, in the form of an LTI system with impulse response $h(n)$, with the addition of white Gaussian noise. The received signal is equalized by an FIR filter with impulse response $w(n)$. The equalizer output is input to a symbol-by-symbol SOQPSK-TG demodulator [6] to produce the bit estimates used to assess the performance of the equalizer. Because of the known preamble and ASM bits in the transmitted data packet, it is possible to compute the minimum mean square error (MMSE) equalizer, which is then used to initialize the CMA equalizer for efficiency. The CMA equalizer is updated by using each received data block after convergence of the adaptation, and bit error rates are determined.

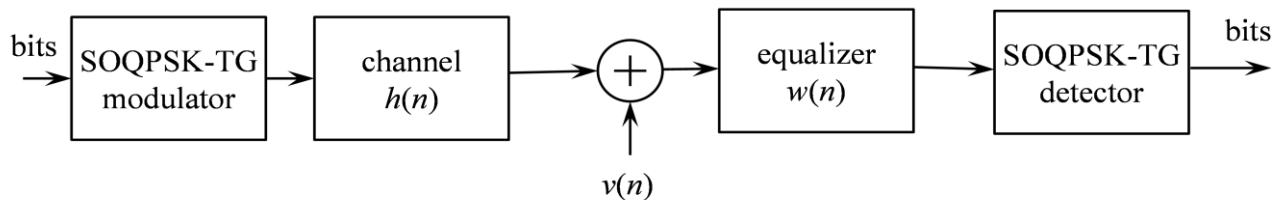


Fig. 2: SOQPSK Communication System Model

The channel is a single-input single-output (SISO) system characterized by the input/output relationship:

$$x(n) = \sum_{k=-K_1}^{K_2} h(k)s(n-k) + v(n), \quad (1)$$

where $h(n)$ is the channel impulse response which is non-causal and of length $(K_1 + K_2 + 1)$, $s(n)$ are samples of the complex-valued low-pass equivalent signal corresponding to the SOQPSK-TG modulated carrier within the packet structure, and $v(n)$ is AWGN noise. The sequence $x(n)$ comprises N samples. Assuming the received signal is sampled at a rate equivalent to 2 samples/bit, N is at least 12672. (The actual length depends on the length of the channel.) For our experiments the equalizer update is done using a single packet of transmitted databits, which was randomly generated, these are modulated and transmitted over a set of aeronautical test channels [4]. The equalizer output is given by

$$y(n) = \sum_{l=-L_1}^{L_2} w_n(l) x(n-l), \quad (2)$$

where $\underline{w}_n = [w_n(-L_1) \dots w_n(-1) w_n(0) \dots w_n(L_2)]^T$ is the equalizer weight vector at iteration, n of the CMA adaptation process.

CMA EQUALIZATION ALGORITHM

Because SOQPSK-TG is constant modulus, it is expected that it will be well-suited for equalization using CMA. The MMSE equalizer is the filter that minimizes the mean square error between the equalizer output $y(n)$ and the channel input $s(n)$ [7]. By initializing CMA with the MMSE equalizer, there will be no phase ambiguity for the CMA equalized symbols.

Description of CMA Equalization Algorithm

The constant modulus algorithm (CMA) is a blind equalization algorithm, which provides an adaptation based on a specific cost function. The cost function, $J_{CMA}(y(n))$, which is minimized, is a function of the distance of the equalizer output from a circle of known radius. This radius is determined from the modulation used for signal transmission. The CMA cost function is given by

$$J_{CMA}(y(n)) = E\left\{\left(|y(n)|^2 - R_2\right)^2\right\}, \quad (3)$$

where R_2 is the radius squared of the CMA circle on which the received samples, $x(n)$ lie. This cost function performs equalization by taking into account the distance between the equalizer output samples and the pre-specified radius of the desired signal. For SOQPSK-TG, the transmitted signal is CPM and of the form: $s(n) = \exp(j\phi(n))$, so the value of CMA radius is $R_2 = 1$. Note that because SOQPSK-TG is a CPM it has constant modulus, and the equalizer operates on samples of the complex-valued low-pass equivalent signal, which are the received samples from a transmitted signal that was constant modulus. This motivates the use of CMA.

The update of the equalizer weights for this cost function is based on a stochastic gradient descent rule given by

$$\underline{w}_{n+1} = \underline{w}_n - \mu \nabla_w J_{CMA}(y(n)) \quad (4)$$

where μ is the algorithm step-size, the equalizer output is a block of data determined by $y(n) = \underline{w}_n^T \underline{x}_n$, for equalizer weight vector \underline{w}_n , and a received data vector, $\underline{x}_n = [x(n + L_1) \dots x(n) \dots x(n - L_2)]^T$.

The gradient vector of the cost function is given as:

$$\nabla_w J_{CMA}(y(n)) = E \left\{ 4 \left(|y(n)|^2 - R_2 \right) y(n) \underline{x}_n^* \right\}. \quad (5)$$

The * operator denotes the complex conjugate, and for these experiments an adaptive step-size, μ was used in (4) which is given (with $\alpha=0.4$) by

$$\mu = \alpha \frac{\|\underline{x}_n^T \underline{w}_n\|}{\|\underline{x}_n^T \nabla_w J(y(n))\|}. \quad (6)$$

In general the CMA equalizer weight vector is updated starting from an initial vector, \underline{w}_0 which is center-tapped. Previous work in [3] has shown the improvement in performance over the CMA equalizer with center-tap initialization of the CMA equalizer when it is initialized using the MMSE equalizer. Since the MMSE-initialized CMA equalizer does converge faster to achieve a lower bit error rate. For SOQPSK the value of the CMA radius to be used should be unity, but a non-unity radius of 2 is tested in our experiments below in order to equalize the received SOQPSK modulated signals and it is observed that when the value of R_2 in equation (3) is chosen to have the value of 2, instead of unity, the CMA equalization works much better.

AERONAUTICAL TELEMETRY CHANNELS

Four aeronautical test channels with impulse responses which have been previously measured from channel sounding experiments at Edwards AFB [4] were used in our experiments. Characteristics of these four channels used are provided in Table 1 in terms of total channel delay spread and total number of non-zero multi-paths. These channels represent snapshots of the flight path of an aircraft. Channels 1 and 3 were captured on a taxiway, channel 4 was captured during takeoff on runway 22 L, and channel 6 is airborne flight.

TABLE 1: CHANNEL INFORMATION

Channel #	1	3	4	6
Channel Length	9	24	19	4
No. of Non-zero taps	3	6	9	4

EXPERIMENTAL SETUP

Implementation of the CMA for PAQ

For this implementation the CMA equalizer weight vector, \underline{w}_0 is initialized using the minimum mean square (MMSE) equalizer, which is computed using the known (the preamble and ASM) databits

contained in the data packet. For this application CMA is to be evaluated on a real-time system which consists of data stream received as multiple packets. Thus for the case of real-time processing, there is an initialization phase during which the incoming data packets are being processed to achieve convergence of the CMA equalizer. This phase may take 50 to 100 data packets, after which the bit error analysis can be done to count total errors on the following data packets. Two real-time approaches are considered here as described in the simulation procedures below.

Simulation Procedure 1 (Multiple Packet - Adaptive startup)

This is referred to as the Block-by-Block (BxB) update procedure.

Step 1: Generate an iNET packet at 2 samples/bit. Send the packet through the test channel and add noise to create the received packet.

Step 2: Initialize the equalizer filter.

Step 3: Equalize, update the equalizer filter coefficients based on the received packet produced in *Step 1*. Go to step 1 until the equalizer has converged (50 packets), then proceed to *Step 4*.

Step 4: Generate a new iNET packet at 2 samples/bit. Send the packet through the test channel and add noise to create a received packet.

Step 5: Equalize, update the equalizer filter coefficients, and count the number of bit errors.

Step 6: Repeat *Step 4* and *Step 5*, 600 times.

Simulation Procedure 2 (Single Packet – Static startup)

Step 1: Generate an iNET packet at 2 samples/bit. Send the packet through the test channel and add noise to create the received packet.

Step 2: Initialize the equalizer filter.

Step 3: Iterate on the received packet produced in *Step 1*, updating the equalizer filter once per packet, until the equalizer has converged (50 iterations).

Step 4: Generate a new iNET packet at 2 samples/bit. Send the packet through the test channel and add noise to create a received packet.

Step 5: Equalize, update the equalizer filter coefficients, and count the number of bit errors.

Step 6: Repeat *Step 4* and *Step 5*, 600 times.

The main difference between these two procedures is as follows: For Simulation Procedure 1, the filter update equation (4) achieves convergence by updating from startup using multiple received data packets. For Simulation Procedure 2 at startup, a single fixed data packet is used for the update of equation (4) until convergence at 50 iterations, and only after this does its update continue with a new data packet being processed at each iteration.

In the section below we present results for the two real-time processing schemes outlined above, and we compare performance of this MMSE-initialized CMA equalizer for the communication system model which was described in Section II based on the test channels outlined in Table I. We consider the effect of changing some parameters in order to achieve the *best* performance indicated by the baseline bit error rates from [3], and in particular we consider the effect of using a CMA radius squared with a value of 2 for SOQPSK equalization. Note that the baseline performance is determined by running steps 1-3 of procedure 2, and averaging over a 500 Monte Carlo simulations. In addition we also compare the effect of using a different CMA radius squared value.

PERFORMANCE EVALUATION RESULTS

In this section we present results which show the performance of the MMSE-initialized CMA equalizer for the communication system model described above using four aeronautical telemetry channels. Since the CMA equalizer is initialized with the MMSE equalizer, the MMSE equalizer performance is included in all plots as well as the baseline performance from [3], and the theoretical lower bound curve for AWGN channel. We compare the bit error curves of the multiple packet startup procedure 1 to the single packet startup procedure 2, where for consistency 100 data packets are used for initialization to convergence of both procedures, and errors are then counted over 600 packets of received data.

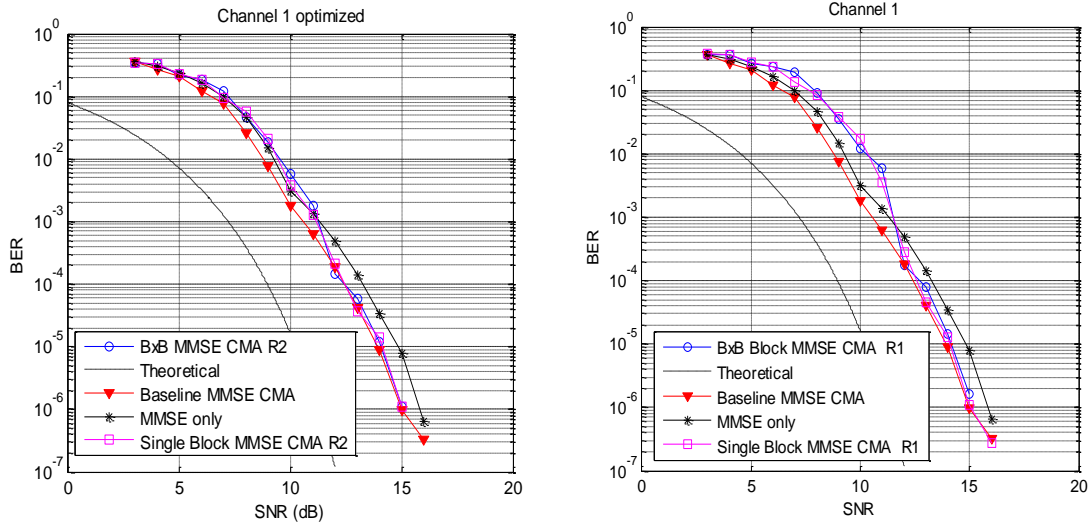


Fig. 3: Single Packet start-up and Block-by-Block CMA Equalizer for channel 1
 $R_2 = 2$ (left) versus $R_1 = 1$ (right)

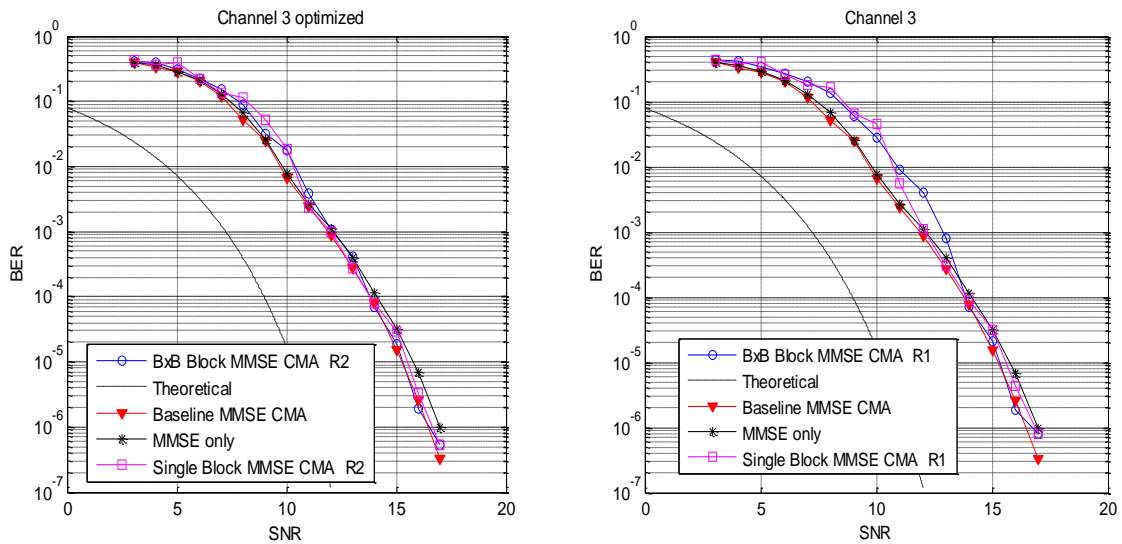


Fig. 4: Single Packet start-up and Block-by-Block CMA Equalizer for channel 3
 $R_2 = 2$ (left) versus $R_1 = 1$ (right)

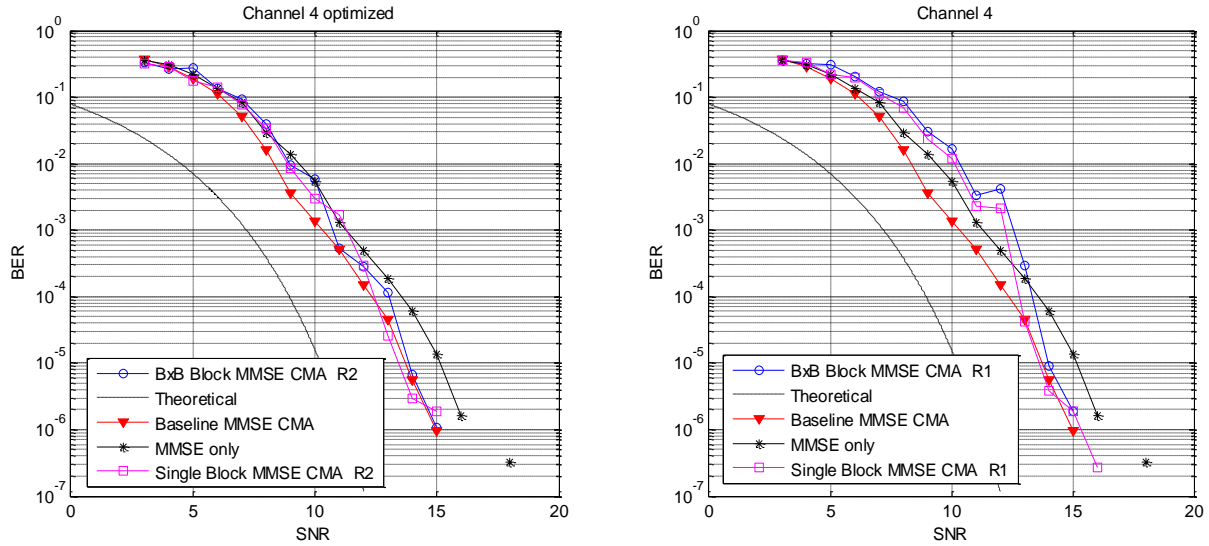


Fig. 5: Single Packet start-up and Block-by-Block CMA Equalizer for channel 4 $R_2 = 2$ (left) versus $R_1 = 1$ (right)

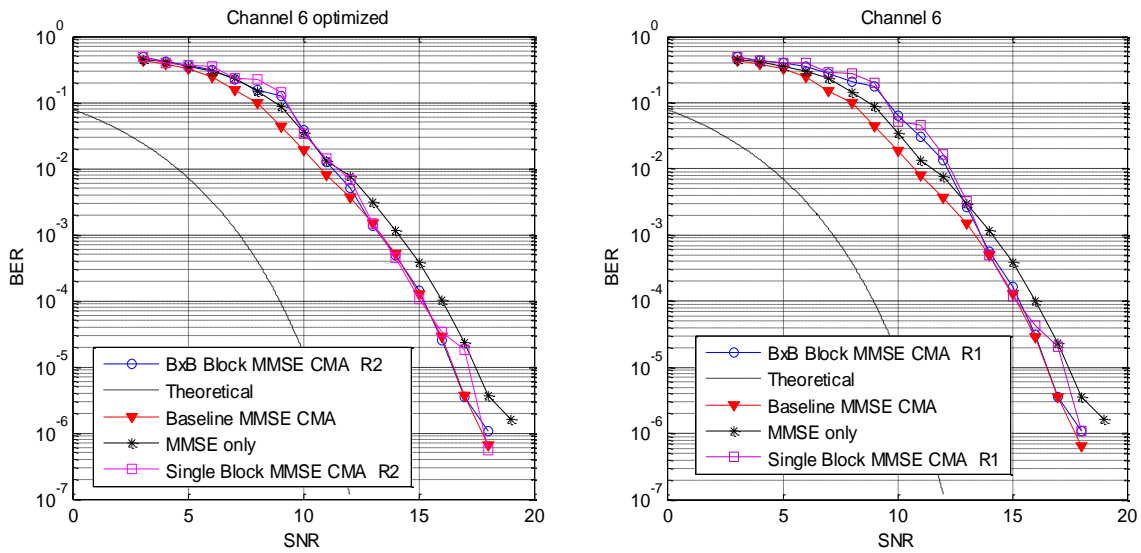


Fig. 6: Single Packet start-up and Block-by-Block CMA Equalizer for channel 6 $R_2 = 2$ (left) versus $R_1 = 1$ (right)

In Figures 3 - 6 we compare the performance of the CMA equalizer using the multiple packet startup (Simulation Procedure 1) to the single packet startup (Simulation Procedure 2) for two different values of the CMA radius squared R_2 . Radius squared $R_2 = 1$ is on the right, while $R_2 = 2$ is the plot on the left of each numbered figure. We note that for both of the procedures tested, the bit error rate performance is generally better for $R_2 = 2$ than for $R_2 = 1$, especially at the higher SNRs. In fact at low SNR, the performance of the MMSE-initialized CMA is much worse than the MMSE equalizer when the CMA radius squared $R_2 = 1$ is used. It would appear that neither of the two simulation procedures has a bit error rate performance advantage over the other, even though both procedures do approach the baseline performance at high SNR.

CONCLUSIONS

In this work, we have presented two block-processing CMA equalization schemes for SOQPSK-TG modulated data, which has been transmitted over an aeronautical channel. Since the transmitted data packets contain known data for the telemetry application of interest (iNET data packet structure) the MMSE equalizer has been used as an initializer for CMA in this application. We have implemented, tested and compared the use of a multiple packet startup (Simulation Procedure 1) and a single packet startup (Simulation Procedure 2) for the CMA algorithm. In general using a CMA radius squared $R_2 = 2$, the MMSE-initialized CMA equalizer achieves better performance than the use of radius 1 for the two procedures introduced here. Considering real-time implementation issues it is observed that for both procedures, the bit error rate performance is also more stable at a lower error rate when the radius squared value of 2 is used, versus a value of unity. It is unclear exactly why the non-obvious choice of CMA radius 2 provides lower bit error rates, and this continues to be a topic of further investigation.

ACKNOWLEDGEMENTS

This work was funded by the Test Resource Management Center (TRMC) Test and Evaluation Science and Technology (T&E/S&T) Program through the U.S. Army Program Executive Office for Simulation, Training and Instrumentation (PEO STRI) under contract W900KK_13-C-0026 (PAQ). Approved for public release; distribution is unlimited. 412TW-PA-14299.

REFERENCES

- [1] R. Johnson, et. al. "Blind Equalization using the Constant Modulus criterion: A Review", Proceedings of IEEE, Vol.86, No.10, pp. 1927-1950, 1998.
- [2] E. Law, "How well does a blind adaptive CMA equalizer work in a simulated telemetry multipath environment", Proceedings of the International Telemetry Conference (ITC) 2004, San Diego, CA.
- [3] A.Cole-Rhodes, H. Umuolo, F. Moazzami, M. Rice, "A Block Processing Approach to CMA Equalization of SOQPSK for Aeronautical Telemetry", submitted to MILCOM 2014.
- [4] M. Rice, "Phase 1 Report: Preamble Assisted Equalization for aeronautical telemetry (PAQ)," Brigham Young University, Technical Report, 2014, submitted to the Spectrum Efficient Technologies (SET) Office of the Science & Technology, Test & Evaluation (S&T/T&E) Program, Test Resource Management Center (TRMC). Also available on-line at <http://hdl.lib.byu.edu/1877/3242>.
- [5] T. Nelson, E. Perrins, M. Rice, "Near optimal common detection techniques for shaped offset QPSK and Feher's QPSK," IEEE Transactions on Communications, vol. 56, no. 5, pp. 724-735, May 2008.
- [6] E. Perrins, "FEC systems for aeronautical telemetry," IEEE Transactions on Aerospace and Electronic Systems, vol. 49, no. 4, pp. 2340-2352, October 2013.
- [7] M. Rice, M. Saquib, M.S. Afran, A.Cole-Rhodes, F. Moazzami, "A Comparison of Three Equalization Techniques for iNET-formatted SOQPSK-TG", submitted to ITC 2014.