Towards Real-Time CMA Equalization by using FFT for Signal Blocks transmitted over an Aeronautical channel

Towards Real-Time CMA Equalization by using FFT for Signal Blocks transmitted over an Aeronautical channel

Peter Oluseyi Taiwo, Itie Serge Kone Dossongui

Faculty Advisors:
Dr. Arlene Cole-Rhodes and Dr. Farzad Moazzami

Department of Electrical and Computer Engineering
Morgan State University

***Abstract*— We consider the problem of equalizing data blocks of signals, which have been transmitted over an aeronautical channel using two different modulation schemes. The equalization is performed using the block-processing constant modulus algorithm (CMA), and in order to achieve real-time processing a Fast Fourier Transform (FFT) is used to compute the gradient of this cost function during equalization. The equalizer length is chosen to be five times of the channel length. For the first experiment, we present the result of equalizing a set of measured data, which was modulated and transmitted using the iNET packet structure with SOQPSK modulation. In this case, the CMA equalizer is first initialized using MMSE and the equalizer coefficients are then updated once, using each entire block (iNET packet). In the second experiment, we apply the FFT-based block processing equalizer to received data blocks of QPSK signals, which have been randomly generated and transmitted over an aeronautical channel. A modified constant modulus algorithm and alphabet matched algorithm (CMA + AMA) equalizer is used to recover these data blocks. For this case of QPSK signals, the equalizer performance is evaluated over 500 Monte Carlo runs, using the average symbol error rate (SER).**

***Index Terms*— Blind Equalization, Constant Modulus Algorithm, Alphabet Matched Algorithm, FFT, Channel Estimation, SOQPSK, iNET, Aeronautic channels, MMSE Equalizers.**

I. INTRODUCTION

In this work we focus on the recovery of a degraded received signal, which has been transmitted over a single-input/single-output frequency selective aeronautical telemetry channel, and corrupted by noise and multipath interference. We recover the transmitted signal by using a process of equalization, which reverses the effect of the multipath channel. In this work, we generate a set of random signals in each data packet and recover these using a blind equalizer. Thus the performance overhead is reduced since blind equalizers do not require the use of training signals.

In our first experiment the bit stream is transmitted using the iNET data packet structure, which has been modulated with shaped offset quadrature phase shift-keying (SOQPSK), version ‘TG’. SOQPSK-TG is a type of continuous phase modulation (CPM) which uses a constant envelope and has a constant radius property. Since previous tests on simulated SOQPSK-TG data, transmitted over aeronautical channels, have shown that CMA initialized by the MMSE equalizer provides a lower bit error rate ([4]) than the usual center-tap initialized CMA equalizer, we will continue to use an MMSE-initialized CMA equalizer to recover transmitted signal here. In order to achieve real-time performance of the equalizer for these aeronautical channels, we will implement our CMA equalizer using a Fast Fourier Transform (FFT) in order to compute the gradient of the CMA cost function and update the equalizer coefficients.

In the second experiment we generate random signals in data blocks, based on a similar packet structure to that specified in iNET. The data is modulated using the quadrature phase shift-keying (QPSK) instead before transmission over the aeronautical channels. For this experiment, the equalizer used is CMA+AMA (constant modulus algorithm + alphabet matched algorithm) with center-tap initialization.

II. Communication System Model

Figure 1 shows the system model used for our experiments. *s(n)* is the transmitted data packet which consists of randomly generated data bits, which are sent over an aeronautical channel specified by *h(n)*. The noisy received signal is specified by *x(n)*, which is then passed through an equalizer to produce the equalized output signal, *y(n)*. The equalizer used for both our experiments is based on CMA [1], and the equalized output signal should be the transmitted signal with a time delay, and a possible phase rotation due to the phase-blind nature of CMA.
The data is transmitted using a packet structure, which is based on that defined by the iNET standards. In Experiment 1, the data is modulated using SOQPSK-TG and the actual iNET data packet structure is used as shown in figure 1 and the specified equalizer, *w(n)* is a data-aided CMA ([4]). In Experiment 2, the data is modulated using QPSK, and the packet structure used is shown in figure 2 and the specified equalizer, *w(n)* is CMA+AMA ([3]).

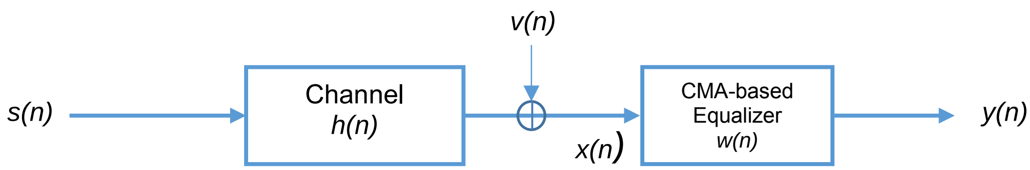

Figure 1: INET data packet structure for SOQPSK

The received signal is given by the convolution operation as

$$x(n) = \sum_{k=-\mathrm{K1}}^{\mathrm{K2}} h(n)s(n-k) + v(n) \tag{1}$$

where *K1* is the length of the non-causal, and *K2* is the length of the causal part of the impulse response of the channel, *h(n)* and *v(n)* is the additive white Gaussian noise (AWGN).
The CMA-based equalizer, *w(n)* is as specified above, and its output is computed using convolution

$$y(n) = w(n) * x(n) = \sum_{l=-L1}^{L2} w(n)x(n-l) \tag{2}$$

where * specifies the convolution operation above, *L1* is the length of the non-causal part, and *L2* is the length of the causal part impulse response of the equalizer, *w(n),* which is being updated during the CMA adaptation process. Note that the length of *w(n)* is (*L1* +*L2* +*1*).

Experiment 1: Packet Structure and Equalizer Description for SOQPSK-TG

In the first experiment the data is modulated using SOQPSK-TG and transmitted in a packet of length, *N* = 6336 bits. This packet structure, which is specified by the iNET standards, is made up of known data which are the preamble and asynchronous marker (ASM) bits, together with the actual (unknown) data. The length of the preamble is 128 bits; the length of the ASM is 64 bits, and actual data consists of 6144 bits. The known preamble and ASM can be used to identify the start of the packet, and also to determine any phase shift introduced by the CMA equalizer.

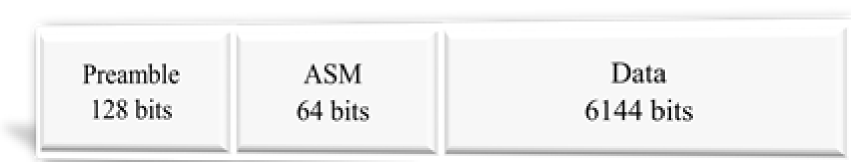

Figure 2: INET data packet structure for SOQPSK

Figure 3 shows an expanded view of the system model of figure 1, which includes the SOQPSK-TG modulator at the transmitter, and an SOPQSK-TG detector which is a demodulator that performs a symbol-by-symbol detection of the data bits at the receiver. (See [5] for details.) Thus we can assess the performance of our equalizer.

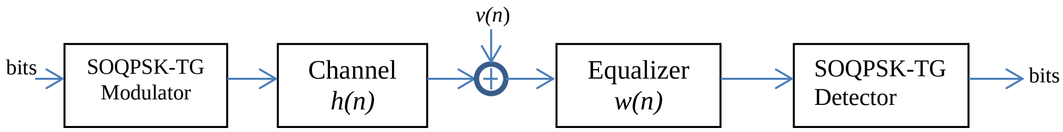

Figure 3: SOQPSK Communication Model

In this experiment we use an MMSE-initialized CMA equalizer. The MMSE equalizer is computed using the known preamble and ASM bits within the first received packet, and that filter is used to initialize the CMA equalizer. This is then followed by an update of the filter coefficients of the CMA equalizer using a single data packet. The equalizer coefficients then continue to be updated once using the CMA update equation (9) for each subsequent received data packet. The detection filter is applied to the equalized data, which is first down-sampled before the CMA phase correction is computed. The phase correction was implemented as follows. The phase lock loop (PLL) is applied to the entire received data packet, including the known preamble + ASM data bits. Assuming that the PLL has converged, the CMA phase mismatch is determined by comparing the last ASM symbol in the equalized iNET packet to its known transmitted value. The computed phase rotation for the last ASM symbol is then applied to the equalized data. Thus the preamble bits in transmitted packet allows the equalizer to do phase correction using data aided phase-lock loop (PLL). Note that with the CMA+AMA equalizer, a PLL is not required.

Experiment 2: Packet Structure and Equalizer Description for QPSK

In the second experiment the data is modulated using QPSK and transmitted in a packet structure (similar to iNET packet structure in experiment 1). It consists of a total of length, N = 3168 bits, which have been modulated using QPSK as shown in figure 2. Note that the length of the preamble is 64 bits, the length of the ASM is 32 bits, and actual data consists of 3072 bits.

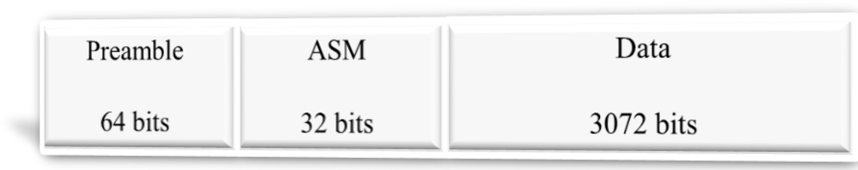

Figure 4: INET data packet structure for QPSK

The constant modulus algorithm (CMA) is effective when the transmission modulation scheme is one which lies a constant radius, such as QPSK. Note that CMA is phase blind, and this requires the use of a phase-correction mechanism. By applying the alphabet matched algorithm (AMA) together with the CMA equalizer, we can successfully equalize QPSK signals to within a phase offset of a multiple of $\pi/2$.

The CMA+AMA equalizer has been successfully applied to 16-QAM, whose signal constellation has both a property of multiple constant radii, together with a rectangular orientation of the signal constellation [3]. Note that AMA is a blind equalization scheme that exploits the knowledge of the known constellation alphabet, and it takes into consideration both the amplitude and phase of the signal. AMA is highly dependent on good initialization which will be provided by CMA when using the combination.

III. AERONAUTICAL TELEMETRY CHANNELS

In our experiments the data packets are transmitted over a sample of four different aeronautical channels from a whole set of 12 channels (0-11). These channels are modeled by a set of FIR filters which represent different phase of a test flight. Channels 1 and 2 represents taxiway before takeoff, channel 4 represents the takeoff phase, and channel 6 represents airborne flight. The impulse responses of these four aeronautical test channels were previously measured from channel sounding experiments, done at Edwards AFB. The characteristics of these channels are shown in Table 1, and the corresponding frequency response curves can be found in [5].

TABLE 1: CHANNEL INFORMATION

Channel #	**1**	**2**	**4**	**6**
Channel Length	9	20	19	4
No. of Non-zero taps	3	8	9	4

IV. EQUALIZER IMPLEMENTATION USING FFT

In this section we describe how the equalizers, for both the SOQPSK and the QPSK experiments, were implemented using the fast Fourier transform. Since the data packets described in section II are to be equalized, so the implementation of our equalizers is by using block processing. Our previous implementation for updating the equalizer coefficients used the Toeplitz matrix. In this work we apply the Fast Fourier transform (FFT) to speed up the computation for a real-time environment of aeronautical telemetry.

Note that the equalized signal as given in equation (2) is applicable to serial processing. Since the data is being transmitted using the data packets described above, we can re-write the equations using block processing as follows. *x(n)* is now the received signal block which is of length *N*, so (2) can be written in Toeplitz form as

$$y(n) = \boldsymbol{XX}\, w(n) \tag{3}$$

where $XX \in \mathbb{C}^{(L+N-1) \times L}$ is a Toeplitz matrix built from the elements of the received signal block *x(n)*, as shown below, and the length of *w(n)* is $L = L1 + L2 + 1$.

$$XX = \begin{bmatrix} x(1) & 0 & \dots & 0 & 0 \\ x(2) & x(1) & \dots & : & : \\ x(3) & x(2) & \dots & 0 & 0 \\ : & x(3) & \dots & 0 & 0 \\ : & : & : & x(1) & 0 \\ x(L) & \dots & \dots & x(2) & x(1) \\ : & : & : & : & : \\ x(N-1) & \dots & \dots & x(N-L+1) & x(N-L) \\ x(N) & x(N-1) & : & : & x(N-L+1) \\ 0 & x(N) & \dots & x(N-2) & : \\ 0 & 0 & \dots & x(N-1) & x(N-2) \\ : & : & : & x(N) & x(N-1) \\ 0 & 0 & 0 & \dots & x(N) \end{bmatrix} \tag{4}$$

Taking Fourier transforms (FFT) of (2), we obtain

$$y(n) = \mathrm{iFFT}\{FFT(w(n))FFT(x(n))\} \tag{5}$$

This implies the taking the FFT of the equalizer and the received signal separately in order to get a computation from the time domain to a corresponding frequency domain. The direct product of these can be done in the frequency domain and the find the inverse Fast Fourier transform (*iFFT*) to retrieve the result, which is the equalized signal, back in the time domain. We also apply this same FFT concept in computing the gradient of the cost function for both the CMA and the CMA+AMA equalizers.

In general, the cost function for the CMA is given by

$$J_{CMA}(\underline{w}) = (|y(n)|^2 - R_2)^2 \tag{6}$$

where R_2 is the radius-squared of the CMA circle, which is computed by $R_2 = \frac{E\{|c(i)|^4\}}{E\{|c(i)|^2\}}$ .Note that $c(i)$ for *i=1,2,….M* are the known constellation points (where *M* =4 for QPSK). For SOQPSK we set $R_2 = 1$, since the transmitted signal is CPM of the form, $e^{j\Omega}$ and it lies on the unit circle.

In the second experiment, our single stage CMA+AMA equalizer takes the received QPSK signal which has been trasmitted over an aeronautical channel and recovers it by updating equalizer coefficients until convergence is reached using the gradient of the joint cost function of the combined CMA and AMA.

The cost function for AMA is given by

$$J_{AMA}\left(\underline{w}\right) = E\left\{1 - \sum_{i=1}^{M} e^{-\frac{|y(n)-c(i)|^2}{2\sigma_{AMA}^2}}\right\} \tag{7}$$

where σ_{AMA} is the constant parameter which specifies the width of the nulls around each constellation point, with M = 4 for QPSK.

Combining CMA with AMA allows CMA to provide the required initialization for the AMA equalizer. The cost function which we use for CMA+AMA is given by

$$J(y) = J_{CMA}(y) + \beta\, J_{AMA}(y) \tag{8}$$

where the individual cost functions are given by (6) and (7).

Note that the equalizer implementation for SOQPSK which uses only the CMA equalizer will be applied using β=0; while the equalizer implementation for the QPSK which uses the CMA+AMA equalizer is done with β=1.

The update of the equalizer weights is based on minimizing the specified cost function. At each stage of equalization, iterative an update of the equalizer is computed until convergence to a final equalizer weight is achieved. This is done using the gradient of the cost functions which then determines an updated equalized output, *y(n)*. The steepest descent rule used for this adaptation process is given by

$$w_{k+1}(n) = w_k(n) - \mu_k\, \nabla J_{\,k}(y) \tag{9}$$

where for iteration k, $w_k(n)$ is the vector of equalizer weights, $\nabla J_{\,k}(y)$ is the block gradient of the cost function. For SOQPSK, we use a fixed step-size of μ_k = 5e-3, while for QPSK, μ_k is an adaptive step size which is computed using the received data block, x as follows:

$$\mu_k = \| x^H\, w_k \| \,/\, \| x^H\, \nabla J_{\,k}(y) \| \tag{10}$$

and $(.)^H$ is the Hermitian (complex-conjugate transpose operator), and the gradient of the CMA+AMA cost function, $\nabla J(y)$ is given by

$$\nabla J(y) = \nabla J_{CMA}(y) + \beta\, \nabla J_{AMA}(y) \tag{11}$$

Since using block processing, the entire received data block $\{x(1),\, x(2) \ldots\ldots x(N)\}$ is processed in a single iteration, thus the gradients are computed as shown in equation (12), which in effect takes an average over all the samples of the data block Thus the block gradient of the CMA cost function is

$$[\nabla J_{CMA}(y)]^T = \left\{4(|y(n)|^2 - R_2)\, y(n)^T \right\}(XX)^* \in \mathbb{C}^{L} \tag{12}$$

and the block gradient of the AMA cost function is

$$[\nabla J_{AMA}(y)]^T = \mathrm{E}\left\{\sum_{i=1}^{M} \frac{1}{\sigma_{AMA}^2}(y(n) - c(i))(XX)^*\, e^{-\frac{|y(n)-c(i)|^2}{2\sigma_{AMA}^2}}\right\} \tag{13}$$

where $(XX)^*$ is the complex conjugate of the matrix specified in (4), and σ_{AMA} is the constant parameter which specifies the width of the nulls around each constellation point for minimization
Define the vector

$$e(n) = E\{\, 4(|y(n)|^2 - R_2)y(n) \in \mathbb{C}^{N} \tag{14}$$

and

$$d(n) = \mathrm{E}\left\{\sum_{i=1}^{M} \frac{1}{\sigma_{AMA}^2} e^{-\frac{|y(n)-c(i)|^2}{2\sigma_{AMA}^2}} (y(n) - c(i))\right\} \in \mathbb{C}^{\mathrm{N}} \tag{15}$$

then we can re-write the block gradient of the CMA cost function in Toeplitz form equivalent to (12) as

$$\nabla J_{CMA}(y) = (\mathrm{XX})^{\mathrm{H}} e(n) = x^*(n) * e(n) \tag{16}$$

and we can also re-write the block gradient of the AMA cost function in Toeplitz form equivalent to (13) as

$$\nabla J_{AMA}(\mathrm{y}) = (\mathrm{XX})\mathrm{H}\, e(n) = x^*(n) * \mathrm{d(n)} \tag{17}$$

where $(\mathrm{XX})^{\mathrm{H}} \in \mathbb{C}^{\mathrm{L\,x\,(L+N-1)}}$ is the Hermitian of the Toeplitz matrix, XX .
Thus based on the convolution operation, we can apply Fourier transforms to compute the gradient of the CMA \ and the AMA cost functions as follows:

$$\nabla J_{CMA}(y) = iFFT\{\mathrm{FFT}[x^*(n)]\ \mathrm{FFT}[e(n)]\} \tag{18}$$

$$\nabla J_{AMA}(y) = iFFT\{\mathrm{FFT}[x^*(n)]\ \mathrm{FFT}[d(n)]\} \tag{19}$$

Thus the gradient of the CMA+AMA cost function, $\nabla \mathrm{J(y)}$ given in (11) can be written in FFT form as:

$$\nabla \mathrm{J(y)} = iFFT\{[\mathrm{FFT}(x^*(n))]\ [\mathrm{FFT}(e(n)) + \beta\ \mathrm{FFT}(d(n))]\} \tag{20}$$

where $e(n)$ is given by (14), and $d(n)$ is given by (15) and the equalizer weights are updated using (9).

V. RESULTS

Figure 5 illustrates the equalizer output of the CMA equalization process which recovers the transmitted iNET packet over QPSK signal. It is an iterative process during which the equalizer weight vector is updated to produce the final equalized output as shown.

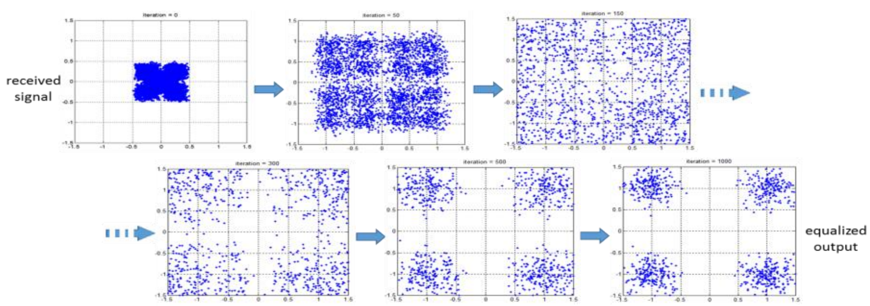

Figure 5: Equalization process to recover QPSK iNET packet

For both the SOQPSK and the QPSK experiments, simulations were done over four aeronautical channels, and the equalizer performance is evaluated over a range of SNR from 3 – 20 dB for 500 Monte Carlo runs. The four aeronautical channels are described in section III, and the respective channel lengths are provided in Table 1. For each channel the equalizer used is of length, L equal to 5 times the channel length.

A timing test was done using the SOQPSK modulated signal on channel 1, using 500 MC runs and 500 iterations of the CMA equalizer, to determine how much faster the equalizer implementation using FFT is over that using Toeplitz matrix. The machine timing from Matlab for the Toeplitz code is 33928 seconds, while that for the FFT implementation was 10988 seconds, which is an initial speed-up of at least 3 times.

For the SOQPSK experiment, the CMA equalizer is initialized using the MMSE equalizer, and the length of the transmitted signal block is N = 6336, and the CMA equalizer is updated over 500 iterations until convergence. For the QPSK experiment, we use the center-tap initialized CMA+AMA equalizer, and the length of the transmitted signal block is N = 3168. The AMA equalizer is implemented using a value of $\sigma_{AMA} = 0.5381$, and the combined CMA+AMA equalizer is updated over 500 iterations until convergence.

The performance of the MMSE-initialized CMA equalizer on the SOQPSK-modulated data packet is shown in Figure 6. The average bit error rate (BER) is plotted versus SNR (dB) for each of the four different channels specified in section III.

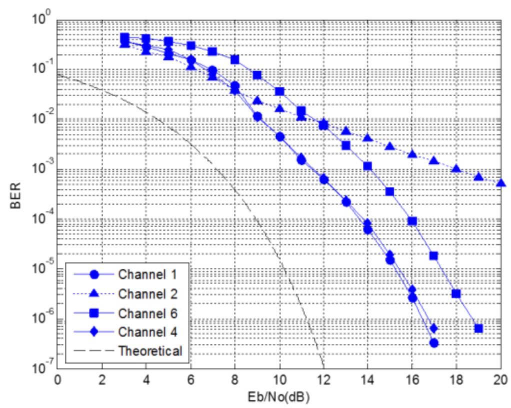

Figure 6: Average BER vs. E_b/N_0 for SOQPSK-modulated data packets

Note that the CMA equalization performs best on channel 1 (on the taxiway) and channel 4 (at takeoff) with zero BER at an SNR of 18 dB and above. Simlarly for channel 6 the equalizer produces zero BER at an SNR of 20 dB (which is airborne flight). The equalizer performance is worst for channel 2, which is also on the taxiway and it is longer channel with additional multipath, and the error rates are above 1e-3 even for high SNR.

The performance of the CMA+AMA equalizer is also being evaluated on the QPSK-modulated data packets for same four channels specified in section III. Initial tests indicate that the CMA-based equalizer for QPSK packets requires a higher SNR range, than the SOQPSK packets in order to successfully recover the data.

VI. CONCLUSIONS

In this work we have shown that a CMA-based equalizer can be used to equalize transmitted packets, in which the data has been modulated using either SOQPSK or QPSK. Since the SOQPSK symbols lie on a circle, the CMA equalizer is able to successfully recover the transmitted data with MMSE initialization. For QPSK, a modified CMA+AMA equalizer is tested to successfully equalize those data packets. The CMA-based block-processing equalizers described above have been implemented using the fast Fourier transform (FFT), and this has provided a speed-up over the regular Toeplitz implementation of a factor of at least three.

VII. ACKNOWLEDGEMENTS

This work is being done in collaboration with Dr. Michael Rice of Brigham Young University. It is 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).

REFERENCES

1. R. Johnson, et. al. “Blind Equalization using the Constant Modulus criterion: A Review”, Proc. 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”, International Telemetry Conference (ITC) 2004.

3. A. Beasley, A. Cole-Rhodes, “Performance of an Adaptive Blind Equalizer for QAM Signals”, Military Communications Conference (MILCOM), Atlantic City, NJ, October 2005.

4. A. Cole-Rhodes, H. Umuolo, F. Moazzami, M. Rice, “A Block Processing Approach to CMA Equalization of SOQPSK for Aeronautical Telemetry”, in Proceedings of the IEEE Military Communications Conference (MILCOM), Baltimore, MD, October 2014.

5. M. Rice, M. Saquib, A. Cole-Rhodes, and F. Moazzami, “Phase 1 Final 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.